Dr. Ir. Agus Wibowo, M.Kom, M.Si, MM.

# Desain VLSI Hemat Daya dengan AI



YAYASAN PRIMA AGUS TEKNIK

Dr. Ir. Agus Wibowo, M.Kom, M.Si, MM.

# Desain VLSI Hemat Daya dengan AI




YAYASAN PRIMA AGUS TEKNIK

PENERBIT :
YAYASAN PRIMA AGUS TEKNIK
Jl. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

ISBN 978-634-7695-45-1 (PDF)
9 786347 695451

**Desain VLSI Hemat Daya dengan AI**

**Penulis :**
Dr. Ir. Agus Wibowo, M.Kom, M.Si, MM.

**ISBN : 978-634-7695-45-1 (PDF)**

**Editor :**
Dr. Joseph Teguh Santoso, S.Kom., M.Kom.

**Penyunting :**
Dr. Mars Caroline Wibowo. S.T., M.Mm.Tech

**Desain Sampul dan Tata Letak :**
Irdha Yunianto, S.Ds., M.Kom

**Penebit** :
Yayasan Prima Agus Teknik Bekerja sama dengan
Universitas Sains & Teknologi Komputer (Universitas STEKOM)

**Anggota IKAPI No:** 279 / ALB / JTE / 2023

**Redaksi :**
Jl. Majapahit no 605 Semarang
Telp. 08122925000
Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

**Distributor Tunggal :**
**Universitas STEKOM**
Jl. Majapahit no 605 Semarang
Telp. 08122925000
Fax. 024-6710144
Email : info@stekom.ac.id

Hak cipta dilindungi undang-undang
Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara apapun tanpa ijin dari penulis

# KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya buku yang berjudul **"***Desain VLSI Hemat Daya dengan AI***"** ini dapat disusun dan diselesaikan dengan baik. Buku ini hadir sebagai upaya untuk memberikan pemahaman yang komprehensif mengenai perkembangan desain *Very Large Scale Integration* (VLSI) berdaya rendah (*low-power*), serta peran strategis kecerdasan buatan (*Artificial Intelligence*) dan pembelajaran mesin (*Machine Learning*) dalam mendorong efisiensi, keandalan, dan kinerja sistem elektronik modern.

Perkembangan teknologi semikonduktor yang sangat pesat dalam beberapa dekade terakhir telah membawa perubahan besar dalam dunia elektronika, komputasi, dan sistem tertanam (*embedded systems*). Ketika skala transistor terus menyusut hingga tingkat *deep sub-micron*, tantangan fisik seperti arus bocor (*leakage current*), disipasi panas berlebih, dan keterbatasan daya baterai menjadi hambatan utama dalam desain sirkuit terpadu. Di tengah tuntutan perangkat yang semakin kecil, cepat, cerdas, dan hemat energi, desain VLSI tidak lagi hanya berfokus pada peningkatan performa frekuensi *clock*, melainkan juga pada pengurangan konsumsi daya statis maupun dinamis, optimalisasi luas area *chip*, peningkatan efisiensi termal, serta ketahanan terhadap variasi proses manufaktur. Dalam konteks inilah, pendekatan berbasis AI dan ML menjadi instrumen yang sangat relevan untuk membantu otomatisasi alur desain, optimasi ruang parameter yang kompleks, serta verifikasi rangkaian terpadu secara lebih cepat, adaptif, dan presisi.

Buku ini disusun secara sistematis untuk memandu pembaca dari fondasi dasar hingga penerapan tingkat lanjut. Pembahasan materi dirancang mencakup empat pilar utama: Fondasi Desain VLSI Daya Rendah dan Tantangan Fisik**:** Mengulas komponen konsumsi daya statis dan dinamis, efek saluran pendek (*short-channel effects*), batas-batas fisik pada MOSFET modern, serta tantangan efisiensi daya pada arsitektur memori. Bagian ini juga membedah teknik pengendalian daya konvensional seperti *clock gating*, *power gating*, hingga *multi-voltage scaling*.

Integrasi AI dalam Alur Desain EDA (*Electronic Design Automation*)**:** Menganalisis transformasi alur desain semikonduktor berkat intervensi kecerdasan buatan. Pembahasan mencakup sintesis logika adaptif, optimasi level RTL, estimasi daya berbasis ML pada tahap awal perancangan (*early-stage estimation*), pencarian arsitektur saraf (*Neural Architecture Search/NAS*), hingga otomatisasi *placement and routing* pada desain fisik yang secara dramatis memangkas waktu siklus pengembangan (*design turnaround time*).

Arsitektur Tingkat Lanjut dan Komputasi Perkiraan (*Approximate Computing*)**:** Membedah penerapan komputasi perkiraan pada arsitektur pembelajaran mesin, komputasi neuromorfik yang meniru cara kerja otak manusia, serta perancangan pengganda perkiraan (*approximate multipliers*) yang toleran terhadap kesalahan demi mencapai efisiensi energi yang ekstrem pada aplikasi *fault-tolerant*.

Sirkuit Analog/Sinyal Campuran Berbantuan AI dan Ekosistem Perangkat Keras**:** Menjangkau domain sirkuit terpadu analog dan sinyal campuran (*mixed-signal*), mencakup optimasi AI pada *Voltage-Controlled Oscillator* (VCO) cincin, manajemen daya adaptif, pemodelan transistor berbasis regresi, integritas pad I/O pada *chip* neuromorfik, hingga implementasi OP-AMP dan DAC hemat daya. Tidak hanya itu, buku ini juga mengulas ekosistem *hardware* AI masa kini yang melampaui arsitektur GPU dan TPU konvensional, termasuk pemanfaatan *Field-Programmable Gate Array* (FPGA) untuk percepatan pembelajaran mendalam (*deep learning*).

Dengan cakupan yang komprehensif dan berorientasi pada masa depan ini, pembaca tidak hanya memahami AI sebagai beban kerja (*workload*) komputasi, melainkan juga sebagai alat (*tool*) revolusioner yang mentransformasi cara kita merancang semikonduktor masa depan. Buku ini diharapkan mampu menjadi referensi yang bermanfaat bagi mahasiswa, dosen, peneliti, maupun praktisi yang bergerak di bidang teknik elektro, semikonduktor, sistem tertanam, dan kecerdasan buatan.

Penyusunan buku ini tentu tidak terlepas dari berbagai keterbatasan dan tantangan. Penulis menyadari bahwa di tengah dinamika teknologi yang terus bergerak cepat, masih terdapat kekurangan di beberapa bagian, baik dari segi penyajian materi, kedalaman pembahasan teknis, maupun kelengkapan contoh aplikasi praktis. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi penyempurnaan karya ini pada edisi-edisi berikutnya.

Akhir kata, penulis berharap buku ini dapat memberikan manfaat yang luas, memperkaya wawasan akademis dan praktis, serta menjadi kontribusi nyata dalam pengembangan ilmu pengetahuan dan teknologi, khususnya pada bidang desain VLSI hemat daya berbasis kecerdasan buatan. Semoga kehadiran buku ini dapat menjadi sumber inspirasi dan referensi yang berguna bagi pengembangan riset maupun penerapan teknologi di masa depan.

Semarang, Juli 2026
Penulis

Dr. Ir. Agus Wibowo, M.Kom, M.Si, MM.

# DAFTAR ISI


**KATA PENGANTAR** ..... **i**
**DAFTAR ISI** ..... **iii**
**BAB 1 DASAR-DASAR VLSI DAYA RENDAH** ..... **1**
1.1 Pendahuluan ..... 1
1.2 Teknologi Memori Yang Berkembang ..... 3
1.3 Sistem Komputasi Elektronik ..... 5
1.4 Jenis Memori Berdasarkan Fungsionalitas ..... 9
1.5 Teknologi Memori Yang Berkembang ..... 12
1.6 Tantangan Pengurangan Ukuran MOSFET ..... 14
1.7 Efek Saluran Pendek ..... 16
1.8 Teknik Yang Digunakan Untuk Mengontrol Efek Kaluran Pendek ..... 17
1.9 Evolusi Dan Masa Depan Teknologi VLSI Berdaya Rendah ..... 19
**BAB 2 MEREVOLUSI DESAIN VLSI DALAM KECERDASAN BUATAN** ..... **20**
2.1 Integrasi Kecerdasan (AI) ..... 20
2.2 Dasar-Dasar Desain VLSI Daya Rendah ..... 23
2.3 Integrasi Kecerdasan Buatan Dalam Desain VLSI ..... 27
2.4 AI Dalam Sintesis Logika Dan Optimasi RTL ..... 32
2.5 Aplikasi AI Dalam Desain Fisik ..... 36
2.6 Estimasi Dan Pemodelan Daya Yang Ditingkatkan AI ..... 38
2.7 Pencarian Arsitektur Neural Yang Berorientasi Perangkat Keras (NAS) ..... 39
2.8 Peningkatan Alat EDA Menggunakan AI ..... 41
2.9 Tantangan Implementasi AI dalam Desain VLSI ..... 43
2.10 Studi Kasus Dan Implementasi Dunia Nyata ..... 44
2.11 Tren Dan Arah Pembahasan Masa Depan ..... 45
2.12 Transformasi AI Dalam Desain VLSI ..... 47
**BAB 3 KOMPUTASI PERKIRAAN PADA ARSITEKTUR PEMBELAJARAN MESIN** ..... **49**
3.1 Tantangan Komputasi Dan Era Silikon Gelap ..... 49
3.2 Klasifikasi Teknik Komputasi Aproksimasi (AxC) ..... 51
3.3 Teknik Pendekatan Perangkat Keras ..... 52
3.4 Strategi Tingkat Arsitektur ..... 54
3.5 Pendekatan Tingkat Perangkat Lunak ..... 57
3.6 Tantangan Dan Arah Masa Depan ..... 64
3.7 Aproksimasi Terkontrol Dalam ML ..... 65
**BAB 4 DESAIN SIRKUIT TERPADU HEMAT DAYA BERBASIS AI DAN ML** ..... **67**
4.1 Era Baru Desain VLSI Berdaya Rendah Berbasis AI ..... 67
4.2 Dasar-Dasar Konsumsi Daya Dalam VLSI ..... 69
4.3 Trade-Off Daya, Luas, Dan Kinerja (PAP) ..... 73
4.4 Contoh Skenario Kompromi Spesifik Industri ..... 75

4.5 Paradigma AI Dan ML Dalam Optimasi Daya VLSI ..... 76
4.6 Desain Tingkat Sistem Dan Eksplorasi Arsitektural ..... 83
4.7 Studi Kasus Dan Aplikasi Industri ..... 87
4.8 Tantangan Dan Isu Pembahasan Terbuka ..... 89
4.9 Interpretabilitas Dan Kepercayaan Model ..... 90
4.10 Beban Komputasi ..... 90
4.11 Generalisasi Di Seluruh Node Teknologi ..... 90
4.12 Integrasi Dengan Alat CAD Yang Ada ..... 91
4.13 Keamanan Dan Ketangguhan ..... 91
4.15 Komputasi Neuromorfik Untuk Arsitektur Berdaya Sangat Rendah ..... 92
4.16 Pembelajaran Terfederasi Dalam Desain Semikonduktor ..... 92
4.17 AI Terjelaskan (XAI) Untuk Otomasi Desain Terpercaya ..... 93
4.18 Pembelajaran Penguatan Untuk Alur Desain Adaptif ..... 93
4.19 AI Yang Terinspirasi Kuantum Untuk Optimasi VLSI ..... 93
4.20 AI Berkelanjutan Dan Ramah Lingkungan Untuk Desain Daya Rendah ..... 93
4.21 Era Baru VLSI Berdaya Rendah Berbasis AI ..... 94
**BAB 5 KOMPUTASI NEUROMORFIK VLSI CMOS RENDAH DAYA ..... 96**
5.1 Komputasi Neuromorfik Dan Arsitektur Terinspirasi Otak ..... 96
5.2 Prinsip-Prinsip Komputasi Neuromorfik Dalam VLSI CMOS Daya Rendah ..... 97
5.3 Blok Bangunan Tingkat Transistor CMOS ..... 99
5.4 Desain Sirkuit Dan Strategi Arsitektural ..... 101
5.5 Pemantauan Arsitektur Dan Profil Daya SNN ..... 103
5.6 Studi Kasus Dan Aplikasi ..... 105
5.7 Arah Masa Depan Dan Kesimpulan ..... 106
**BAB 6 PENGGANDA PERKIRAAN BERDAYA RENDAH BERTOLERANSI KESALAHAN .... 108**
6.1 Komputasi Aproksimasi Dalam Pencitraan Dan Akselerator AI ..... 108
6.2 Pengali Perkiraan Yang Diusulkan ..... 113
6.3 Pengali Aproksimasi Berdaya Rendah Untuk Pemrosesan Gambar ..... 124
**BAB 7 OPTIMASI AI PADA VCO CINCIN ..... 125**
7.1 Optimasi RVCO Berbasis AI ..... 125
7.2 Metodologi Desain Yang Dibantu AI ..... 128
7.3 Paradigma Optimasi Berbasis AI ..... 130
7.4 Tantangan Dan Keterbatasan Optimasi RVCO Berbasis AI ..... 135
7.5 Pendekatan AI untuk Optimasi dan Adaptasi VCO Cincin ..... 137
7.6 Pengali Aproksimasi Berdaya Rendah Untuk Pemrosesan Gambar ..... 139
**BAB 8 MANAJEMEN DAYA ADAPTIF BERBASIS AI DALAM DESAIN VLSI ..... 140**
8.1 Manajemen Daya VLSI Berbasis AI ..... 140
8.2 Persyaratan Desain VLSI Daya Rendah ..... 141
8.3 Teknik Manajemen Daya Tradisional Dalam Desain VLSI ..... 145
8.4 Integrasi AI Dan ML Dalam Optimasi Daya VLSI ..... 153
8.5 Teknik Manajemen Daya Cerdas Berbasis AI ..... 155

8.6 Dampak Praktis Dan Tren Yang Muncul ..... 162
8.7 Manajemen Daya Cerdas Berbasis AI Dalam Desain VLSI ..... 164
**BAB 9 PEMODELAN VLSI EFISIEN MELALUI PREDIKSI MOSFET BERBASIS REGRESI ... 166**
9.1 Pemodelan Transistor Berbasis Machine Learning ..... 166
9.2 Perkembangan Pemodelan Transistor Berbasis Pembelajaran Mesin ..... 167
9.3 Deskripsi Dataset ..... 168
9.4 Evaluasi Algoritma Regresi Untuk Pemodelan MOSFET ..... 183
**BAB 10 INTEGRITAS PAD I/O PADA CHIP NEUROMORFIK HEMAT ENERGI ..... 184**
10.1 Integritas IOPAD Dalam Sistem Neuromorfik ..... 184
10.2 Pendahuluan Tentang Jaringan Neural Spiking ..... 184
10.3 Evaluasi Model Neuron Spiking Dalam Komputasi ..... 195
10.4 Integritas I/O Chip Neuromorfik ..... 210
**BAB 11 OP-AMP 3-TAHAP RNMC DAYA RENDAH UNTUK BIOMEDIS 45 NM ..... 211**
11.1 Optimasi AI/ML Pada Desain OP-AMP 3-Tahap RNMC Daya Rendah 45 nm ..... 211
11.2 Perancangan OTA Tiga Tahap Dengan Kompensasi RNMC ..... 213
11.3 Hasil Simulasi Dan Analisis Kinerja OTS 45 nm ..... 217
11.4 Desain OP-AMP Biomedis 45 nm Ultra-Daya Rendah Berbasis RNMC ..... 222
**BAB 12 KOMPUTASI PERKIRAAN UNTUK PEMBELAJARAN MESIN ..... 224**
12.1 Efisiensi Perangkat Keras AI Melalui Komputasi Aproksimasi ..... 224
12.2 Teknik Komputasi Aproksimasi ..... 226
12.3 Arsitektur ML ..... 227
12.4 Jenis-Jenis Pengganda Perkiraan ..... 231
12.5 Kesalahan Dan Sensitivitas Sebagai Metrik Kinerja ..... 234
12.6 Pertukaran Desain Dalam Komputasi Aproksimasi ..... 236
12.7 Tantangan Dan Keterbatasan Komputasi Aproksimasi ..... 237
12.8 Arah Pengembangan Komputasi Aproksimasi ..... 238
12.9 Revolusi Perangkat Keras AI Berbasis Komputasi Aproksimasi ..... 239
**BAB 13 EKOSISTEM HARDWARE AI MELAMPAUI GPU DAN TPU ..... 240**
13.1 Aplikasi Kecerdasan Buatan Dan Kebutuhan Perangkat Keras ..... 240
13.2 Peran CPU Dalam Menggerakkan AI Dan ML ..... 241
13.3 Evolusi Dan Dampak GPU Dalam AI ..... 243
13.4 TPU Untuk AI Dan ML ..... 245
13.5 Akselerasi AI Dan ML Dengan FPGA ..... 248
13.6 Komputasi Dalam Memori Dan Memristor ..... 253
13.7 AI Di Tepi Perangkat ..... 255
13.8 Teknologi Dan Perkembangan Perangkat Keras AI ..... 256
13.9 Dinamika Dan Tren Masa Depan Perangkat Keras AI ..... 258
**BAB 14 DAC 8-BIT MODE ARUS DAYA RENDAH BERBANTUAN AI UNTUK VLSI ..... 260**
14.1 Integrasi AI Pada Konverter Digital-Ke-Analog Mode Arus ..... 260
14.2 Perancangan DAC Berbasis OTRA CMOS ..... 262
14.3 Hasil Simulasi Dan Pembahasan OTRA ..... 268

14.4 Penyetelan Adaptif Berbasis AI Untuk Kinerja Yang Tangguh ..... 274
14.5 Koreksi Kinerja Waktu Nyata DAC Mode Arus Melalui Optimasi AI ..... 280
**BAB 15 DESAIN DAN IMPLEMENTASI FPGA PADA PEMBELAJARAN MENDALAM ..... 282**
15.1 Akselerasi Pembelajaran Mendalam Menggunakan Arsitektur FPGA ..... 282
15.2 Arsitektur FPGA ..... 286
15.3 Desain Berbasis HDL Konvensional ..... 291
15.4 Sintesis Tingkat Tinggi Modern (HLS) ..... 295
15.5 DL Vs. HLS: Tinjauan Komparatif ..... 298
15.6 Aplikasi Real-Time Dari Deep Learning ..... 299
15.7 Kesimpulan ..... 300
**DAFTAR PUSTAKA ..... 301**

# BAB 1
# DASAR-DASAR VLSI DAYA RENDAH

## 1.1 PENDAHULUAN

Selama beberapa dekade terakhir, telah terjadi revolusi luar biasa dalam bidang elektronik, dan miniaturisasi perangkat, penemuan material baru, dan meningkatnya permintaan akan kepadatan integrasi tidak akan pernah berhenti. Keinginan umat manusia untuk kinerja dengan konsumsi daya rendah bertanggung jawab atas penggantian tabung vakum besar yang mengonsumsi energi dalam jumlah besar, oleh transistor persimpangan bipolar kecil, kemudian FET unipolar, dan akhirnya CMOS. Dalam beberapa dekade terakhir, revolusi eksplosif dalam bidang elektronik (penyusutan dimensi perangkat dan munculnya material baru) bersamaan dengan peningkatan tanpa henti dalam kepadatan integrasi chip telah disaksikan.

Dorongan tak terelakkan menuju rasio kinerja terhadap daya yang lebih baik pada umat manusia diwakili dengan baik oleh penggantian tabung katup penyerap energi yang besar dengan BJT solid-state yang kecil, kemudian FET, dan kemudian CMOS. Terlepas dari semua perkembangan ini, bidang ini berada di persimpangan jalan dengan efisiensi energi dalam sistem VLSI menjadi perhatian besar untuk mendesain sistem. Penggunaannya telah didorong oleh kemajuan teknologi VLSI, dan sekarang digunakan dalam banyak aplikasi, mulai dari komputasi mobile hingga elektronik konsumen, termasuk sistem otomotif, dan kekayaan *Internet of Things* (IoT) yang berkembang pesat.

Masing-masing bidang ini membahas bagian-bagian yang berbeda dari sirkuit terpadu, tetapi semuanya memiliki tujuan penting yang sama: menggunakan lebih sedikit energi. Solusi desain hemat energi sangat penting untuk elektronik konsumen karena mereka membutuhkan perangkat kecil dan ringan dengan masa pakai baterai yang lama. Pusat data dengan banyak modul memori dan unit pemrosesan juga menggunakan banyak energi, yang menghabiskan banyak uang dan merusak lingkungan. Arah utama adalah peningkatan teknologi CMOS yang konstan. Prinsip penskalaan Dennard menyatakan bahwa pengurangan jumlah transistor telah menyebabkan peningkatan eksponensial jumlah elemen elektronik pada substrat silikon. Hal ini telah meningkatkan kinerja dan menurunkan biaya.

Namun, penyusutan desain perangkat yang terus-menerus menciptakan masalah baru, terutama dalam hal konsumsi daya. Seiring perkembangan, hal-hal seperti efek kanal pendek, efek kebocoran, dan ketidakpastian menjadi lebih jelas. Hal ini mempersulit untuk mempertahankan atau meningkatkan efisiensi energi sesuai dengan Hukum Moore. Perubahan dramatis dalam tujuan desain telah memungkinkan revolusi teknologi ini. Pada masa sebelumnya, kecepatan dan kinerja adalah hal terpenting dalam VLSI.

Namun sekarang, hal terpenting adalah konsumsi daya yang rendah. Dua hal utama yang memungkinkan transformasi ini adalah batasan fisik dan teknis dari perluasan lebih lanjut dan meningkatnya kebutuhan akan perangkat seluler dan nirkabel yang harus bekerja dengan daya yang sangat sedikit. Contoh dari masalah ini adalah Static Random Access Memory

(SRAM) *on-chip*, yang sebelumnya populer karena cepat dan bekerja dengan baik dengan operasi logika. Namun, seiring semakin banyaknya perangkat yang menggunakannya, arus bocor yang besar yang menyertai geometri submikron yang sangat kecil telah menjadikannya faktor penentu utama dalam seberapa besar daya yang dibutuhkan sistem. Kebutuhan aplikasi baru, seperti IoT dan edge computing, membuat pencarian solusi VLSI berdaya rendah menjadi semakin sulit.

Orang biasanya menggunakan baterai kecil dalam teknologi pengiriman yang harus menggunakan daya sangat sedikit dan menjaga keamanan data. Memori volatil biasa (SRAM dan DRAM) menggunakan banyak daya saat tidak digunakan karena menyimpan muatan dalam kapasitor. Ini berarti bahwa mereka tidak dapat menggunakan baterai yang digunakan dalam aplikasi IoT, yang lebarnya sekitar satu milimeter. Karena batasan tegangan ambang dan fakta bahwa proses sulit diprediksi, sulit untuk mengoptimalkan penggunaan daya. Ini merupakan kebutuhan sekaligus tantangan bagi node teknologi yang canggih. Untuk mengatasi masalah ini, kita perlu melihat material dan struktur memori baru. Alternatif untuk memori volatil tradisional atau bahkan sebagai tambahan termasuk teknologi memori non-volatil, seperti NOR Flash, *Phase Change RAM* (PCRAM), *Ferroelectric RAM* (FeRAM), *Resistive RAM* (RRAM), dan *Magnetic RAM* (MRAM). Solusi-solusi tersebut dicirikan oleh berbagai kompromi dalam hal kecepatan, daya tahan, skalabilitas, dan yang terpenting, efisiensi daya. Perkembangan seperti *Spin-Transfer Torque MRAM* (STT-MRAM) menawarkan daya tahan, retensi, dan tegangan operasi rendah yang lebih baik, sehingga meningkatkan kemungkinan aplikasi perangkat berdaya rendah. Meskipun demikian, terdapat inovasi tingkat sirkuit dan sistem konservatif yang harus mengatasi masalah teknologi, seperti margin penginderaan yang buruk atau tegangan tinggi dari solusi yang ada. Penskalaan perangkat sehubungan dengan disipasi daya, baik dalam hal disipasi daya statis maupun dinamis, merupakan masalah berkelanjutan dengan kepadatan integrasi.

Daya dinamis dihasilkan ketika beban kapasitif diisi dan dikosongkan dalam sirkuit dan terutama disebabkan oleh switching, yang secara tradisional diminimalkan dengan menurunkan tegangan suplai dan bertujuan untuk meminimalkan node kapasitif. Namun, pada MOSFET modern, umumnya tidak mungkin untuk menurunkan tegangan suplai karena tegangan ambang membatasi hal ini, dan konduksi subambang dimulai. Sebaliknya, ancaman melampaui daya dinamis sebagai kontributor utama dalam total penggunaan energi adalah daya statis atau kebocoran, yang tumbuh secara eksponensial seiring dengan berkurangnya ukuran peralatan dan meningkatnya suhu. Ini merupakan pergeseran paradigma dalam desain VLSI dan telah mendorong pengembangan teknologi multigate, seperti FinFET dan FET gate-all-around, serta desain rekayasa kanal dan source/drain untuk menurunkan kebocoran dan meningkatkan kinerja perangkat yang tangguh. Desain VLSI daya rendah juga menghadapi kompleksitas kebutuhan untuk mengatasi kendala kinerja tinggi, keandalan, dan biaya. Aktivitas modern menggunakan besaran teknologi, perencanaan spasial optimal, dan kecanggihan bangunan untuk melakukan trade-off antara wilayah, kecepatan, energi, dan output produksi. Hal ini dicapai melalui partisi sirkuit dan tata letak lantai yang halus, serta algoritma optimasi yang terinspirasi dari biologi dengan hasil pengurangan panjang koneksi

dan ukuran chip, sehingga meningkatkan efisiensi energi dan throughput sistem. Penekanan bergeser ke pendekatan desain holistik yang memiliki inovasi tingkat perangkat dan optimasi tingkat sistem, tidak hanya topologi transistor baru tetapi juga skema penugasan dan alokasi sumber daya yang cerdas.

Aspek terpenting adalah membangun VLSI berdaya rendah generasi berikutnya dengan mengembangkan model teoritis, analitis, dan berbasis simulasi yang akan memiliki kemampuan untuk memandu perilaku perangkat dalam kondisi operasi yang lebih canggih. Penemuan transistor planar gerbang tunggal menjadi transistor multigate dengan berbagai jenis: transistor gerbang ganda, gerbang tiga, dan tanpa sambungan merupakan ekspresi dari upaya industri untuk meningkatkan dampak saluran pendek dan mencapai ukuran yang tidak pernah dibayangkan oleh fisika klasik. Selain itu, proses penyeimbangan antara kinerja dan proses konservasi energi biasanya umum dalam hal proses rekayasa oksida gerbang, dielektrik high-k dan teknologi substrat, serta optimasi proses. Kendala VLSI berdaya rendah bukan hanya dimensi perangkat yang lebih kecil, tetapi juga melibatkan pemikiran ulang tentang material, desain sirkuit, sistem memori, dan desain sistem. Bidang ini terletak di persimpangan antara inovasi berbasis aplikasi, otomatisasi desain elektronik, fisika dan ilmu material, serta otomatisasi desain elektronik.

## 1.2 TEKNOLOGI MEMORI YANG BERKEMBANG

Sejarah teknologi semikonduktor selama empat dekade terakhir didorong oleh upaya tanpa henti untuk mengurangi ukuran peralatan elektronik dan menciptakan sistem VLSI yang sangat canggih yang menggunakan transistor yang sangat kuat dan hemat energi. Tren penskalaan teknologi di mana ukuran geometris transistor MOSFET secara sistematis diskalakan telah menjadi katalis bagi kemajuan dan pertumbuhan pesat sirkuit terpadu di setiap sektor aplikasi dalam elektronik konsumen dan komputasi berkinerja tinggi. Penskalaan, selain mendorong kemampuan chip yang lebih tinggi dan memungkinkan aplikasi kompleks melalui kepadatan perangkat yang tinggi, yang difasilitasi pada silikon, secara teoritis juga mengurangi biaya manufaktur dan kebutuhan energi. Penskalaan pertama kali dijelaskan oleh Dennard, di mana ia berpendapat bahwa pengurangan ukuran fisik, khususnya panjang gerbang, lebar saluran, dan ketebalan oksida, dan secara proporsional mengurangi tegangan, akan meningkatkan kinerja tanpa menyebabkan peningkatan konsumsi daya yang signifikan. Hal ini secara perlahan akan membentuk dasar dari Hukum Moore yang terkenal, yang mencatat bahwa jumlah transistor pada sirkuit terpadu berlipat ganda setiap sekitar 18–24 bulan. Dengan mengikuti prinsip-prinsip penskalaan, industri semikonduktor telah mencapai kepadatan integrasi yang luar biasa, kecepatan komputasi, dan penyusutan ukuran perangkat dari waktu ke waktu.

Di antara keuntungan utama penskalaan perangkat MOSFET adalah peningkatan kemampuan untuk menggerakkan arus dan penurunan kapasitansi input. Ini tidak hanya meningkatkan kecepatan switching tetapi juga mengurangi pengisian dan pengosongan beban kapasitif dalam sirkuit, sehingga secara positif memengaruhi kecepatan dan disipasi daya dinamis. Penskalaan menambahkan proses yang lebih kompleks ke area tertentu melalui

minimisasi kapasitansi dan resistansi parasit; oleh karena itu, penskalaan meningkatkan kecepatan operasional transistor dan kemampuan untuk melakukan fungsi yang lebih kompleks. Namun, skalabilitas teknologi memiliki keterbatasan. Pelemahan bertahap yang terus-menerus pada sifat-sifat perangkat juga dikaitkan dengan sejumlah masalah, seperti perkembangan efek kanal pendek, yang menyulitkan kontrol elektrostatik konvensional pada desain MOSFET. Pada ukuran perangkat submikron dan nanometer yang sangat kecil, penurunan tegangan ambang, penurunan penghalang yang diinduksi oleh drain (DIBL), efek pembawa panas, kebocoran drain yang diinduksi oleh gerbang (GIDL), dan variasi doping acak menjadi jelas dan merupakan sumber ketidakandalan dan pemborosan energi pada perangkat.

Penskalaan tingkat lanjut sangat menurunkan tegangan ambang, mengakibatkan peningkatan geometris pada arus kebocoran subambang, dan memperburuk parameter kinerja ION/IOFF, yang penting untuk operasi daya rendah. Pada perangkat besar, penyempitan kanal antara daerah penipisan sumber/drain dan kanal meningkat, dan oleh karena itu, gerbang memiliki kontrol yang lebih sedikit atas pembawa kanal. Hal ini, secara tidak langsung, meningkatkan arus kebocoran di dalam perangkat saat tidak beroperasi dan secara signifikan meningkatkan jumlah daya yang disipasi saat tidak beroperasi. Telah diketahui bahwa tren modern dalam penskalaan CMOS dapat menyebabkan situasi di mana kehilangan daya statis melebihi disipasi dinamis, yang menimbulkan kekhawatiran bagi sistem VLSI dengan keterbatasan daya.

Masalahnya adalah perlunya menemukan keseimbangan antara kecepatan perangkat, konsumsi energi, dan stabilitas. Pemendekan panjang gerbang dan ukuran bukaan sumber/drainase pada perangkat dapat menyebabkan gangguan fisik dan listrik. Contoh efek samping dari penskalaan agresif adalah modulasi panjang kanal, saturasi kecepatan, variasi doping, dan sensitivitas terhadap efek pembawa panas. Medan listrik yang meningkat di sisi perangkat kanal pendek berkontribusi pada efek pembawa panas dan menyebabkan variasi negatif pada tegangan ambang, dan dapat mengganggu sifat isolasi oksida gerbang, yang dapat menyebabkan keausan dini pada perangkat. Posisi gerbang dan drain yang lebih tinggi menciptakan kebocoran pada drain dan merupakan penyebab peningkatan kebocoran pada penerowongan pita-ke-pita. Untuk mengurangi tantangan yang menyertai penskalaan, beberapa peningkatan telah diusulkan di bidang rekayasa perangkat VLSI.

Desain multigate, MOSFET dua gerbang dan tiga gerbang menggabungkan arsitektur desain gate-all-around (GAA), yang telah mengubah desain transistor. Dengan arsitektur tersebut, tingkat kontrol elektrostatik ditingkatkan dan penurunan besar efek kanal pendek direalisasikan, mempertahankan arus ON-state yang besar dan kebocoran OFF-state yang rendah, melalui penambahan gerbang tambahan untuk mengatur kanal. Arsitektur gerbang ganda sangat stabil terkait ambang batas dan dapat mencapai kontrol yang halus terhadap sifat-sifat perangkat melalui perubahan fungsi kerja gerbang. Perangkat trigate, yang ditandai dengan arsitektur silikon tipis dan housing 3-gate, meminimalkan jalur kebocoran dan meningkatkan kontrol gerbang, sehingga memiliki skalabilitas dan kinerja yang lebih tinggi dalam rangkaian yang kompleks.

MOSFET dikenal memiliki efek kanal pendek yang tinggi, terutama pada ukuran kecil, yang dapat dikurangi melalui penggunaan MOSFET GAA yang mencakup struktur nanowire silikon, yang telah dicirikan oleh sifat transpor kuasi-balistiknya. Perangkat ini memiliki metode oksidasi yang unggul dalam pembuatannya, dan telah dibuat kompatibel dengan CMOS saat ini dan karenanya dapat dengan mudah diintegrasikan ke dalam node teknologi saat ini dan masa depan. Jenis inovasi lainnya termasuk transistor tanpa sambungan (junctionless transistor), yang berupaya menghilangkan kebutuhan akan sambungan ultra-dangkal, meningkatkan kelayakan penskalaan sub-nanometer, dan meningkatkan ketahanan elektrostatik sirkuit memori, analog, dan komputasi.

Selain geometri perangkat, peningkatan rekayasa tambahan mencakup ekstensi sumber/drainase, termasuk struktur drain yang didoping ringan (LDD), yang bertujuan untuk mengurangi medan listrik tinggi di ujung drain, mengurangi kemungkinan ionisasi dampak dan efek DIBL, dan meningkatkan kinerja perangkat pada frekuensi tinggi. Oksida gerbang kurang dari satu nanometer dianggap sebagai langkah besar, dan desain high-k serta susunan dielektrik multi-lapisan penting dalam mengendalikan kebocoran dan memastikan kapasitansi gerbang yang kuat, yang berbanding lurus dengan kinerja pada sirkuit VLSI canggih.

Masalah penskalaan juga dapat diatasi dengan rekayasa substrat, seperti penerapan semikonduktor senyawa. Pengurangan tegangan suplai sebagai teknik konvensional untuk mengurangi daya dinamis juga diperparah oleh keterbatasan pada peningkatan tegangan ambang yang memengaruhi tidak hanya kecepatan switching tetapi juga kinerja kebocoran. Doping saluran, zat dielektrik, dan arsitektur perangkat adalah area pembahasan saat ini untuk membantu penskalaan teknologi, keberlanjutan energi, dan keandalan fungsional sirkuit terpadu. Sebagai sistem yang menjelaskan bagaimana parameter saling berhubungan, peraturan dan model penskalaan saat ini, seperti teori penskalaan Dennard yang terkenal, mendefinisikan hubungan antara parameter, termasuk panjang kanal, ketebalan oksida gerbang, tegangan suplai, disipasi daya, dan frekuensi, sebagai peta jalan menuju miniaturisasi sektor semikonduktor saat ini dan yang akan datang.

Pengembangan VLSI berdaya rendah di masa depan tidak hanya akan didasarkan pada pembahasan ulang inovatif tentang geometri dan material perangkat, tetapi juga paradigma yang sepenuhnya baru, seperti penerowongan kuantum, nanowire, transistor berbasis material 2D, dan optimasi tingkat sirkuit yang komprehensif. Penskalaan teknologi merupakan peluang sekaligus tantangan bagi VLSI berdaya rendah dan membutuhkan keseimbangan yang kompleks antara kreativitas individu dalam desain perangkat, ilmu material, dan teknologi manufaktur untuk memperkenalkan sistem semikonduktor berdensitas tinggi, hemat energi, dan andal. Penggunaan teknologi VLSI di masa depan bergantung pada keberhasilan pengelolaan konsekuensi penskalaan melalui solusi canggih yang menjadi perhatian baik dari segi fisika dasar perangkat, maupun implementasi praktis desain sirkuit terpadu.

## 1.3 SISTEM KOMPUTASI ELEKTRONIK

Dunia modern sangat berubah dengan penggunaan teknologi elektronik yang telah menyebar ke hampir setiap bidang kehidupan manusia. Munculnya dan adopsi dunia

elektronik, termasuk perangkat primitif seperti jam tangan, hingga yang canggih seperti superkomputer dan kota pintar, telah mengubah cara orang hidup dan berperilaku di berbagai lingkungan. Kemajuan teknik elektro tidak hanya mengurangi hambatan terhadap inovasi teknologi, tetapi juga meningkatkan laju pengembangan dan komersialisasi teknologi baru. Dalam hal ini, sistem komputer elektronik dapat dilihat sebagai pilar peradaban modern, karena merupakan elemen kunci bagi gaya hidup modern dan industri modern.

Baik dalam otomatisasi industri maupun sistem komputer pribadi, sistem elektronik terintegrasi memanfaatkan interaksi antara komponen perangkat keras (termasuk prosesor, unit memori, dan mekanisme input/output) untuk mengumpulkan, memproses, dan mentransfer informasi. Elektronika telah menyatu dengan bidang lain seperti ilmu komputer dan teknologi informasi untuk menghasilkan penggunaan baru, seperti bangunan cerdas, perangkat IoT, dan perangkat yang dapat dikenakan, yang digunakan di berbagai bidang seperti perawatan kesehatan, pertanian, eksplorasi ruang angkasa, dan pertahanan. Koneksi elektronika dengan teknik mesin akan menyebabkan peningkatan jumlah robot industri dan peningkatan produktivitas serta efisiensinya, serta pengurangan ketergantungan pada tenaga kerja manusia. Teknologi yang terkait dengan bidang diagnostik, platform telemedisin, instrumen bedah robotik, dan lainnya (khususnya elektronika dan kedokteran) mengubah pemahaman tentang perawatan.

Sistem komputer listrik memiliki dua lapisan dasar: komponen analog dan digital. Lapisan-lapisan tersebut juga berkomunikasi melalui topologi yang kompleks, dengan pemrosesan logis dilakukan oleh sirkuit digital dan koneksi ke dunia fisik dilakukan oleh sirkuit analog. Semua peralatan elektronik menggunakan subsistem komputasi sebagai dasar yang dapat melakukan kombinasi fungsi aritmatika, logika, pengalamatan, alokasi, dan pemrosesan sinyal dalam berbagai bentuk seperti modul komputasi otomatis, sistem yang dapat dikonfigurasi ulang, dan unit pemrosesan data waktu nyata.

Dengan meningkatnya permintaan akan perangkat cerdas, kecil, dan hemat energi, kemampuan prosesor, kepadatan memori, dan kemampuannya untuk beroperasi dalam jangka waktu yang lebih lama menjadi isu yang semakin penting. Pengembangan teknologi pemrosesan dan penyimpanan telah memainkan peran penting dalam elektronik berkinerja tinggi untuk memenuhi kebutuhan kecepatan dan kapasitas penyimpanan yang terus meningkat. Pada tahun 1965, lahirnya Hukum Moore memberikan kontribusi besar bagi masa depan seluruh bisnis elektronik. Penggandaan jumlah transistor per generasi sirkuit terpadu memungkinkan penskalaan perangkat buatan manusia; yang menyebabkan produksi barang yang lebih cepat dan lebih fungsional dengan setiap pembuatan selanjutnya. Namun, ini adalah jalan yang penuh dengan rintangan yang membutuhkan inovasi setiap saat.

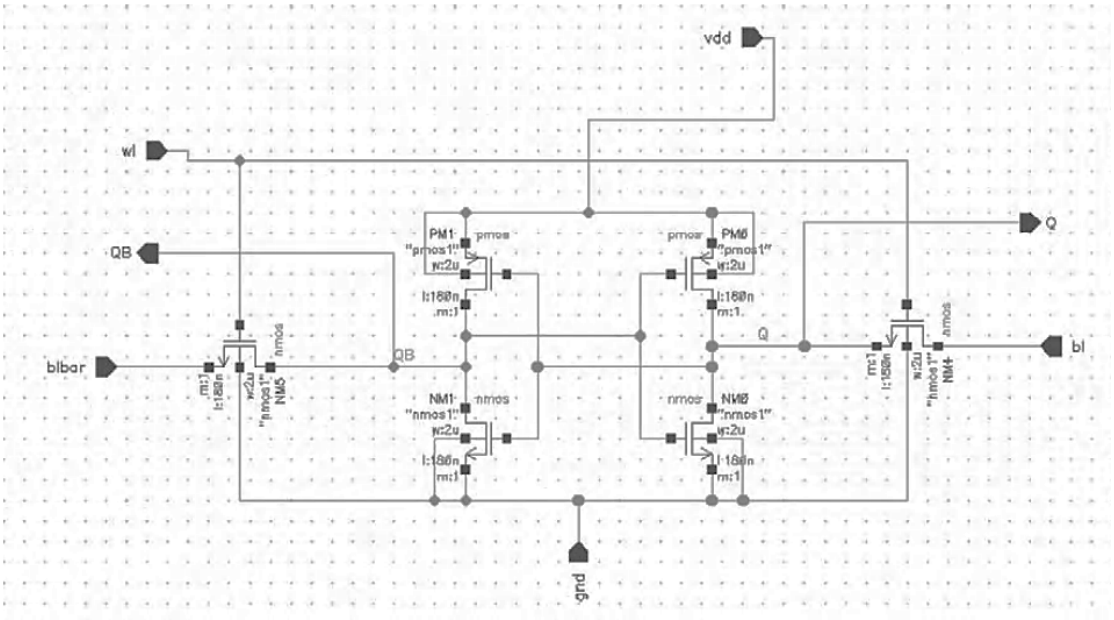
Elemen paling penting dari sistem komputer elektronik saat ini adalah teknologi memori, yang memiliki fitur memori volatil (sementara) dan non-volatil (jangka panjang). Memori volatil, misalnya SRAM dan DRAM, adalah satu-satunya jenis yang cepat dan harus terus-menerus diberi daya, atau informasi yang tersimpan akan hilang. Kemampuan untuk menyimpan data tanpa daya pada memori non-volatil, termasuk Flash, *Magnetic RAM* (MRAM), *Phase Change RAM* (PCRAM), dan *Resistive RAM* (RRAM), juga berarti bahwa hal itu

tidak lagi terbatas pada skala arsitektur tradisional dan dapat digunakan untuk menciptakan arsitektur berdaya rendah dan berdensitas tinggi. SRAM adalah memori in-chip dominan dalam desain VLSI saat ini yang berfungsi dalam cache dan register karena memiliki kecepatan yang sangat tinggi dan memiliki interoperabilitas yang mudah dengan proses logika SRAM. Ini didasarkan pada kontrol tertanam atau prosesor berkecepatan tinggi.

Namun, arus bocor meningkat dengan node teknologi yang lebih kecil di mana SRAM digunakan, dan lebih jauh lagi, terdapat peningkatan laju arus bocor yang tinggi dengan suhu, serta variasi dalam proses. Kebocoran dan kebutuhan untuk beroperasi terus-menerus menjadikan SRAM sebagai penyebab utama konsumsi daya total, terutama berkaitan dengan perangkat portabel kecil bertenaga baterai dan pusat data skala besar. Untuk mengurangi konsumsi daya, trennya adalah mengurangi tegangan suplai; penurunan tegangan lebih sulit karena penurunan tegangan ambang batas teknologi CMOS yang dikembangkan. Selain itu, margin kebisingan dan keandalan menjadi masalah begitu tegangan suplai diturunkan, dan menjadi sulit untuk menurunkannya lebih jauh lagi.

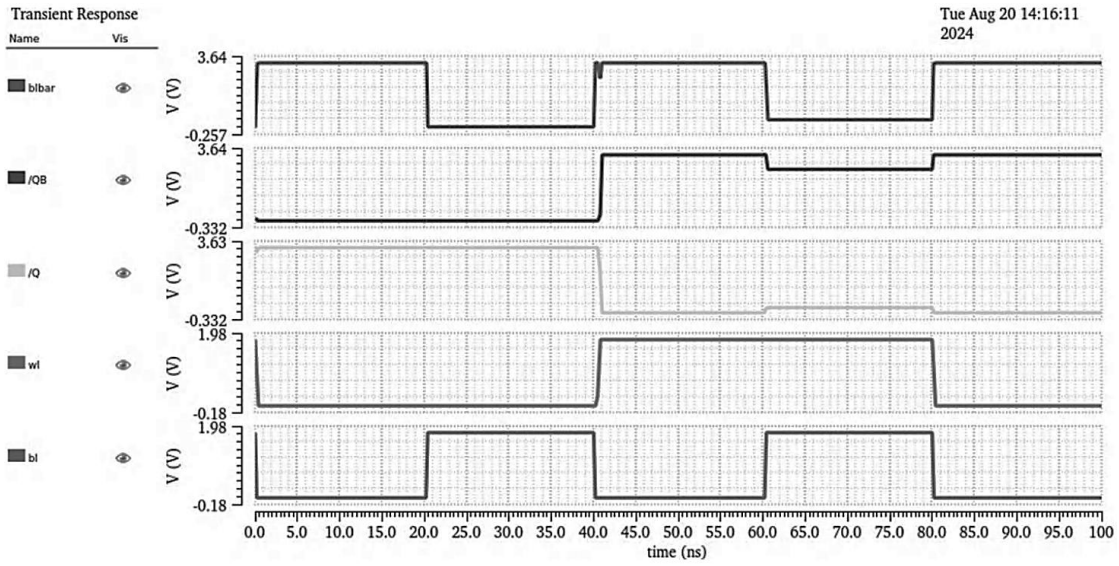
Sistem komputer elektronik telah ditingkatkan oleh teknologi memori non-volatil yang dapat mencakup NOR Flash, PCRAM, FeRAM, RRAM, dan MRAM, dan membantu meminimalkan penggunaan daya siaga dan menyimpan data selama pemadaman listrik. Teknologi ini memiliki kelebihan dan kekurangan yang unik, terkait dengan daya tahan, waktu akses, skalabilitas, dan kompatibilitas dengan metode manufaktur tradisional. Contohnya adalah NOR Flash, yang umum karena proses manufaktur yang telah dikembangkan, dan integrasi NOR Flash yang lancar dengan sirkuit CMOS; namun, teknik manufaktur yang sama membuat perangkat tersebut mengkonsumsi banyak daya selama pemrograman dan penghapusan. Teknologi baru seperti STT-MRAM menjanjikan karena aksesibilitas yang cepat, tegangan operasional rendah, daya tahan tinggi, dan integrasi kepadatan tinggi.

Selain itu, saat ini terdapat perdebatan mengenai batasan tingkat sirkuit, termasuk optimasi margin penginderaan. Sistem komputer elektronik dirancang berdasarkan konsep memori utama dan sekunder. Memori utama biasanya terdiri dari RAM dan ROM, di mana RAM cepat dan mudah berubah, dan digunakan dalam komputasi. Informasi disimpan secara permanen di memori sekunder untuk memudahkan proses pengambilan data dan memastikan kontinuitas dalam alur kerja. Komponen bistabil SRAM digunakan sebagai penyimpanan berkecepatan tinggi sementara; komponen kapasitif DRAM digunakan sebagai penyimpanan berkecepatan tinggi periodik; instruksi operasional yang diperlukan dalam ROM; versi yang dapat diprogram ulang dan dihapus secara elektrik, termasuk EEPROM dan Flash, merupakan bagian dari kategori ini. Gambar 1.1 menunjukkan diagram skematik SRAM 6T Konvensional. Kompleksitas hubungan antara kecepatan, kapasitas, volatilitas, dan biaya menentukan jenis memori mana yang tepat untuk sistem komputasi tertentu. Perkembangan teknologi masih didorong oleh keinginan yang tak pernah puas untuk menciptakan sistem komputer yang lebih canggih menggunakan elektronika.




**Gambar 1.1** Diagram skematik SRAM 6T Konvensional.

Pemendekan lebih lanjut panjang kanal transistor, seperti yang diramalkan oleh Hukum Moore, telah membuat sirkuit terpadu mencapai kepadatan yang sebelumnya tidak mungkin dicapai; namun, hal ini telah menciptakan masalah seperti kebocoran, penurunan integritas sinyal, dan penurunan keandalan. Para insinyur elektronik juga berupaya mengurangi dampak tersebut melalui optimasi tegangan suplai, pengembangan desain perangkat yang lebih baik, dan penggabungan perangkat switching alternatif, termasuk spintronik, memristor, dan FET terowongan. Inovasi memori non-sukarela juga mengesankan dalam hal perubahan. MRAM, yang menggunakan sambungan terowongan magnetik, memiliki kecepatan baca/tulis yang tinggi dan secara teoritis masa pakai yang tidak terbatas, artinya dapat digunakan sebagai modul memori universal. Manfaat desain memristor adalah dapat menggabungkan memori dan logika, yang mendukung komputasi dalam memori dan rekayasa neuromorfik, yang merupakan masa depan kecerdasan buatan dan pembelajaran mesin. Gambar 1.2 menunjukkan bentuk gelombang transien SRAM 6T konvensional.




**Gambar 1.2** Bentuk gelombang transien SRAM 6T Konvensional.

Konvergensi yang terlihat dalam komputasi elektronik tidak terbatas pada perkembangan individu; hal ini ditandai dengan integrasi ide-ide dalam sejumlah disiplin ilmu. Elektronik juga menawarkan sistem komunikasi hemat energi berkualitas tinggi, sistem diagnostik medis generasi berikutnya, sistem transportasi adaptif, dan pertanian presisi. Dengan pergeseran perangkat menuju sistem yang lebih bersifat siber-fisik, subsistem komputasi menjadi semakin relevan dalam pengelolaan pemrosesan sinyal yang kompleks, integrasi data, dan kontrol waktu nyata.

Kunci masa depan sistem komputer elektronik terletak pada kompromi antara efisiensi daya, kinerja, dan keandalan. Permasalahan terkait penskalaan perangkat, integrasi memori volatil dan non-volatil, serta arsitektur heterogen tetap menjadi pendorong utama di balik upaya berkelanjutan untuk mengeksplorasi material baru, topologi sirkuit, dan metode desain bersama secara keseluruhan. Dengan kemajuan bidang kecerdasan terintegrasi, komputasi elektronik akan menjadi sorotan, dan akan terus berkembang dengan peningkatan berkelanjutan serta menciptakan inovasi di setiap bidang.

## 1.4 JENIS MEMORI BERDASARKAN FUNGSIONALITAS

Memori adalah blok bangunan dari setiap sistem digital dan komputer; memori dibutuhkan tidak hanya untuk menyimpan, mengambil, dan mengakses informasi, tetapi juga untuk mendukung berbagai perangkat, seperti sensor bergerak dan pusat data besar. Perbedaan teknologi memori saat ini dapat dikaitkan dengan sifat spesifikasi arsitektur desain sistem elektronik yang bervariasi dan kemajuan teknologi dalam sistem teknologi. Untuk menentukan bagaimana kebutuhan penyimpanan yang berubah dapat dipenuhi, perangkat memori dibagi berdasarkan prinsip fisik yang mendasarinya, sifat retensi data, dan peran aplikasi fungsional dalam hierarki sistem.

Perbedaan utama dalam perangkat memori terjadi baik dalam inti fungsional maupun antara perangkat volatil dan non-volatil. Teknologi memori, yang dapat berupa memori volatil,

seperti SRAM dan *Dynamic Random Access Memory* (DRAM), didefinisikan sebagai retensi data hanya selama perangkat memiliki daya. Ketika daya dimatikan, semua informasi yang tersimpan di dalamnya hilang. Aplikasi memori volatil sebagian besar melibatkan perangkat elektronik portabel, sistem tertanam, dan struktur cache prosesor untuk mendukung akses dan pemrosesan yang cepat. Prosesor menggunakan SRAM sebagai memori cache karena kecepatan baca-tulisnya dan fakta bahwa ia dapat dibuat menggunakan CMOS. Namun demikian, perawatannya mahal dan merupakan titik kebocoran utama seiring perkembangan teknologi. Telah terjadi peningkatan kebutuhan akan jenis memori non-volatil yang diperlukan untuk memenuhi persyaratan ketahanan data, bahkan jika terjadi gangguan daya. Data disimpan dengan aman dalam memori non-volatil (NVM) dan tidak dapat diubah atau dihapus. Pembagian fungsional lain membagi NVM menjadi subkelas berdasarkan mekanisme kerjanya: memori berbasis muatan, berbasis polaritas, dan berbasis resistansi.

Memori semacam itu telah merevolusi solusi penyimpanan pada perangkat portabel dengan ratusan megabit penyimpanan yang andal dan ekonomis di ponsel, SSD, dan sistem tertanam. Namun, memori berbasis muatan tidak dapat diterapkan pada aplikasi yang membutuhkan frekuensi penulisan tinggi karena masalah daya tahan dan penggunaan daya yang tinggi selama pemrograman dan penghapusan. Memori material polarisasi pembagi yang bergantung pada tegangan seperti RAM feroelektrik (FRAM atau FeRAM) dapat menyimpan data di mana keadaan polarisasi dikendalikan oleh muatan listrik, yang menyebabkan waktu akses yang tinggi dan kinerja tinggi dalam aplikasi penulisan data [6]. FeRAM dapat digunakan untuk menawarkan alternatif dielektrik dengan material feroelektrik dalam sel penyimpanan; ia non-volatil tetapi mirip dengan kinerja DRAM, dan oleh karena itu, teknologi ini dapat ditempatkan dalam persyaratan yang dibatasi daya di lingkungan di mana pembaruan data yang sering dibutuhkan.

Sebagian besar pembahasan telah terkonsentrasi pada teknologi memori berbasis resistansi yang menawarkan potensi tinggi untuk mengatasi kendala teknologi berbasis muatan tradisional. Teknologi tersebut adalah *Magnetoresistive RAM* (MRAM), *Phase Change Memory* (PCM atau PCRAM), dan *Resistive RAM* (RRAM). Para peneliti RAM Magnetoresistif (MRAM) menyimpan data dengan menyesuaikan sifat biner orientasi magnetik di dalam lapisan antarmuka Magnetic Tunnel Junction (MTJ) dan mengambil data dengan mengukur perubahan oposisi. MRAM menawarkan kecepatan baca/tulis yang sangat tinggi, masa pakai tulis balik yang lama yang memungkinkannya untuk mendukung penulisan balik berulang, dan yang terakhir, bersifat non-volatil. Versi yang lebih halus adalah *Spin-Transfer Torque MRAM* (STT-MRAM), yang sangat skalabel dan mengkonsumsi daya lebih sedikit untuk memanfaatkan sifat kuantum putaran elektron, yang menjadikannya pesaing prospektif aplikasi memori universal, baik secara mandiri maupun dalam sistem tertanam. *Phase Change RAM* (PCRAM) memanfaatkan transformasi kimia reversibel dari material kalkogenida antara keadaan amorf dan kristal, dengan masing-masing memiliki resistansi karakteristik. Metode ini mendorong berbagai desain sel, kepadatan tinggi, dan skala besar. Namun, metode ini dibatasi oleh daya tahan, pergeseran selama resistansi, dan panas yang dihasilkan oleh pemrograman ke sel penyimpanan lain di node canggih.

RAM resistif (RRAM), juga dikenal sebagai ReRAM, bergantung pada satu konsep: filamen konduktif di dalam material dielektrik dibentuk atau dihilangkan oleh pulsa tegangan, sehingga beralih antara dua keadaan konduktif. Arsitektur ini mampu mendukung desain baru dari array cross-bar berdensitas tinggi, yang dapat digunakan baik dalam model komputasi in-memory maupun neuromorfik di masa depan, selain penggunaan saat ini dalam penyimpanan konvensional. Penggunaan di lapangan perlu mempertimbangkan variabilitas, skalabilitas, dan kemampuan manufaktur perangkat. RAM berbasis memristor dianggap sebagai teknologi masa depan yang memecah kesenjangan logika dan memori. Teknologi ini menawarkan jejak yang kecil, skalabilitas yang luar biasa, dan dapat digunakan dalam komputasi in-memory, dan NAND Flash mungkin akan digantikan. Muatan terakumulasi yang telah melewati memristor terhubung dengan resistansi mesin, dan perangkat dapat mempertahankan keadaannya bahkan ketika catu daya dimatikan. Mereka telah menunjukkan potensi besar di bidang sistem neuromorfik, komputasi analog, dan sistem berbasis crossbar karena efisiensi dan kecepatannya.

Tingkat pertumbuhan NVM (*Non-Value Memory*) tinggi, dan didasarkan pada faktor praktis dan strategis. NVM tertanam dengan tampilan tunggal baru sedang dikembangkan sebagai unit fundamental dalam sistem digital, elektronik kendaraan, dan aplikasi IoT yang menuntut efisiensi energi, perlindungan informasi, dan miniaturisasi karena masalah penskalaan SRAM on-chip terkait dengan penggunaan daya dan kebocoran serta variasi proses. Jenis NVM baru seperti MRAM, PCRAM, RRAM, dan memristor tidak hanya akan menjadi solusi penyimpanan industri tetapi juga sebagai bagian dari model logika, model komputasi dalam memori, dan model perangkat keras yang aman.

Memori primer dan sekunder merupakan bagian dari aktivitas fungsional memori. Volatilitas dan kecepatan adalah kualitas yang membedakan memori utama dan memori sekunder karena yang terakhir menyimpan informasi dan tidak berubah dengan operasi sistem. Drive mekanis, solid-state drive, dan media optik, seperti CD dan DVD, juga merupakan perangkat penyimpanan sekunder yang tidak memiliki padanan dalam hal kapasitas penyimpanan, dalam hal pelestarian data, dan aplikasi data bervolume tinggi. Desain elektronik modern menggabungkan manfaat dari kedua jenis memori dengan mendesain memori secara hierarki dan menetapkan keseimbangan antara kecepatan, kapasitas penyimpanan, dan konsumsi daya.

Tingkat pertama hierarki memori adalah cache, yang sebagian besar menggunakan SRAM karena kecepatan aksesnya yang tinggi, yang kemudian diikuti oleh DRAM sebagai memori utama dan perangkat penyimpanan non-volatil sekunder. Piramida memori mampu memenuhi kebutuhan kinerja tinggi dari volume data besar seperti analisis big data, kecerdasan buatan, dan penginderaan tertanam, serta kebutuhan penyimpanan besar dari inti berkinerja tinggi. Hal ini mendorong pergeseran ke arsitektur memori yang digerakkan oleh aplikasi dan pengembangan arsitektur memori hibrida dan desain memori bertumpuk melalui sel multi-level, integrasi 3D, dan platform komputasi heterogen. Upaya sedang dilakukan untuk meningkatkan daya dan keandalan dengan pembentukan sirkuit kontrol yang canggih,

pendekatan penyegaran selektif, dan pengontrol cerdas, yang spesifik untuk jenis memori fungsional yang akan digunakan.

Implementasi dan pendekatan berbasis alam dari memori berbasis fungsionalitas diklasifikasikan dan diterapkan sedemikian rupa sehingga akan memengaruhi sistem VLSI berdaya rendah dengan tingkat detail yang halus. Setiap teknologi memori, baik volatil maupun non-volatil, berbasis muatan maupun resistif, telah diberi nilai tambah ke dalam ceruk fungsional tertentu dengan pembahasan berkelanjutan untuk mengurangi kompromi dan berinovasi dalam hal kinerja, masa pakai, dan integrasi yang mutakhir. Kebutuhan akan memori yang lebih berorientasi pada fungsionalitas akan terus meningkat seiring dengan perkembangan sistem digital dalam hal kinerja dan fleksibilitas, dan akan berpuncak pada pengembangan generasi baru elektronik cerdas dan hemat energi.

## 1.5 TEKNOLOGI MEMORI YANG BERKEMBANG

Teknologi memori baru merupakan salah satu perkembangan utama dalam upaya menciptakan komputer VLSI berdaya rendah, berdensitas tinggi, dan berkinerja tinggi. SRAM, DRAM, dan NOR Flash tradisional memiliki keterbatasan tinggi dalam hal skalabilitas, kebocoran, dan daya tahan karena kebutuhan daya pemrosesan, efisiensi energi, dan kapasitas penyimpanan yang terus meningkat di pusat data, perangkat seluler, mobil, dan IoT. Pada bagian ini, kami telah memberikan pembahasan rinci tentang perangkat memori baru yang mentransformasi bidang VLSI dan operasinya, manfaatnya, serta masalah integrasi yang ada saat ini.

**Memori Non-Volatil yang Berkembang**

Kebutuhan akan kepadatan yang lebih tinggi, kecepatan yang lebih tinggi, serta konsumsi daya yang lebih rendah telah mendorong peningkatan desain VLSI. Embedded SRAM, meskipun sebagian besar diimplementasikan sebagai memori cache dan solusi memori akses cepat, karena memiliki kecepatan baca/tulis yang sangat tinggi, merupakan konsumen arus bocor yang besar, terutama ketika ukurannya sesuai karena membutuhkan daya untuk menyimpan data yang tersimpan. Kebutuhan daya yang besar dari embedded SRAM menimbulkan masalah bagi aplikasi yang menggunakan daya baterai, serta pusat data besar. Flash dan media penyimpanan non-volatil lainnya, bersama dengan SRAM, juga telah dikembangkan sepenuhnya, kompatibel dengan manufaktur, memiliki ukuran sel yang kecil, dan juga andal. Namun, memori flash membutuhkan tegangan pemrograman dan penghapusan yang besar; oleh karena itu, mereka membutuhkan daya untuk menulis, yang merupakan masalah signifikan dalam desain sistem yang dibatasi energi.

**Magnetoresistive RAM (MRAM) dan Teknologi Spintronik**

Produk lain yang telah diusulkan di sektor pembahasan memori adalah *Magneto-Resistant RAM* (MRAM) dan *spin-transfer torque* (STT-MRAM), yang didefinisikan oleh kecepatan, non-destruktif, dan skalabel. Informasi dalam MRAM disimpan menggunakan orientasi magnetik dari dua lapisan feromagnetik yang terdapat di antara lapisan tipis yang disebut MTJ. Berdasarkan penyelarasan paralel atau antiparalel, MTJ menunjukkan berbagai tingkat resistansi logika biner. STT-MRAM merupakan peningkatan dari arsitektur ini; arus

terpolarisasi spin digunakan untuk memodifikasi polaritas lapisan magnetik bebas; oleh karena itu, dapat digunakan untuk melakukan penulisan yang lebih cepat, menggunakan daya yang lebih rendah. Teknologi MRAM memiliki sejumlah sifat yang diinginkan: kecepatan baca/tulis tinggi, masa pakai miliaran siklus, dan non-volatilitas, yang berarti data disimpan bahkan tanpa daya.

Karakteristik ini menjadikan MRAM sangat menarik dalam aplikasi memori universal, di mana diusulkan untuk menggabungkan manfaat DRAM, SRAM, dan Flash, dengan bonus tambahan berupa konsumsi daya siaga yang rendah. Perangkat spintronik, yang menyimpan informasi pada spin elektron daripada pada muatannya, telah meningkatkan masa depan pengembangan di luar batasan CMOS planar yang diskalakan, dan mungkin memperluas Hukum Moore lebih jauh.

**Memori Perubahan Fase (PCM)**

Prinsip kerja PCM adalah bahwa bentuk amorf dan kristal dari material kalkogenida memiliki resistansi listrik yang berbeda. Pulsa arus menyebabkan pemanasan lokal, yang menyebabkan transisi fase ke keadaan kristal resistansi rendah atau keadaan amorf resistansi tinggi, yang sesuai dengan penyimpanan data biner. Fitur unik PCM meliputi penskalaan, kecepatan switching tinggi, daya tahan tinggi, dan daya saing tinggi dalam penggunaan memori utama. Terlepas dari efek yang jelas menguntungkan dari PCM pada kepadatan dan retensi data, teknologi ini mengandung sejumlah masalah teknologi, yang sebagian besar terkait dengan gangguan penulisan dan pergeseran resistansi. Teknologi manajemen kesalahan yang kuat, serta teknologi peningkatan daya tahan, telah dieksplorasi untuk memberikan beberapa solusi, dan hibrida antara PCM dan sistem berbasis Flash atau DRAM sedang dikembangkan untuk memanfaatkan atribut komplementer dari masing-masing (daya tahan, kecepatan, dan efektivitas biaya).

**Memori Akses Acak Resistif (RRAM)**

RRAM adalah teknologi NVM berbasis resistansi pilihan pertama. Ini adalah karya yang didasarkan pada material oksida yang terletak di antara konduktor logam di mana tegangan bias diterapkan untuk membentuk atau memutus filamen konduktif di dalam oksida dan karenanya, mengubah resistansi sel total untuk mengkodekan informasi. Fitur-fitur utama RRAM meliputi arsitektur sel yang sederhana, tegangan operasi tinggi, skalabilitas tinggi, dan implementasi array crossbar. RRAM memiliki kepadatan tinggi dan dapat ditumpuk dalam 3D, dan karenanya merupakan teknologi masa depan yang layak pada memori on-chip ultra-padat. Perubahan status yang cepat dan siaga daya hampir nol disebabkan oleh metode pengalihan penyimpanan non-muatan, yang karakteristiknya adalah mobilitas ion. Namun, masalah seperti perbedaan ambang batas pengalihan, kinerja terbatas karena siklus program/hapus yang berulang, dan kompleksitas dalam integrasi masih perlu diteliti.

**Sistem Memori yang Ditingkatkan dengan Memristor**

Memristor, sakelar resistif dua terminal, pertama kali ditemukan pada tahun 1971 dan diwujudkan pada akhir tahun 2000-an, yang mencirikan keadaan terprogramnya sebagai nilai tegangan dan arus historis. Hal ini membawa kemampuan untuk menggabungkan pemrosesan data dan penyimpanan data sedemikian rupa sehingga lebih menarik bagi komputasi

neuromorfik dan sistem komputasi dalam memori. Array memristor mampu melakukan pemrosesan logika dalam memori, membatasi aliran data dan juga dapat menghemat banyak energi. Memristor bersifat non-volatil dengan kecepatan switching yang lebih tinggi dan dimensi yang lebih kecil. Arsitektur crossbar memristor menawarkan paralelisme ekstrem, yang dibutuhkan oleh kecerdasan buatan dan komputasi tepi. Saat ini, sedang dilakukan upaya untuk meningkatkan homogenitas perangkat, mengurangi arus sneakpath, dan menyederhanakan proses manufaktur untuk memungkinkan daya saing penyimpanan berbasis memristor dalam aplikasi massal.

**Dampak yang Lebih Luas dan Lintasan Masa Depan**

Permintaan yang meningkat untuk operasi daya rendah, akses bandwidth tinggi, dan integrasi sistem merupakan kekuatan pendorong di balik tren baru teknologi memori. Mengingat kemajuan dalam subsistem memori hibrida, penting untuk disebutkan bahwa subsistem tersebut tidak saling eksklusif, dan dibuat untuk meniru sifat MRAM yang instan dan berdaya tahan tinggi, kepadatan arsitektur RRAM dan PCM, serta paradigma logika dalam memori yang diusulkan oleh memristor. Dalam arsitektur komputer papan tunggal, keempat domain tersebut harus ditingkatkan untuk mengatasi hambatan memori ketika kecepatan prosesor lebih cepat daripada laju transfer data dan ketika disipasi daya dalam memori volatil tidak lagi layak dalam sistem VLSI dengan penempatan antarmuka yang padat. Perkembangan ini akan membantu generasi mendatang dari sistem elektronik yang hemat energi, beroperasi terus-menerus, dan cerdas untuk memenuhi tuntutan besar peralatan yang dapat bergerak sendiri, sensor cerdas, dan program yang membutuhkan banyak data melalui pembahasan yang berkelanjutan dan pematangan teknologi tersebut secara konsisten.

## 1.6 TANTANGAN PENGURANGAN UKURAN MOSFET

Miniaturisasi teknologi MOSFET, yang menjadi dasar pengembangan sirkuit terintegrasi, telah menjadi tantangan besar, yang dipicu oleh peningkatan efek kanal pendek dan kendala fisik lainnya pada perangkat. Karena pengurangan panjang gerbang dan dimensi kritis lainnya seperti yang ditentukan oleh Hukum Moore, lebih banyak fenomena yang terkait dengan perangkat skala nano secara negatif memengaruhi kinerja perangkat. Dampak masalah ini pada sifat listrik, keandalan, dan disipasi daya MOSFET sangat luas; oleh karena itu, hal ini membuat penskalaan CMOS lebih lanjut ke aplikasi VLSI daya rendah menjadi lebih rumit. Penskalaan CMOS didasarkan pada ide dasar untuk mengurangi ukuran perangkat, yaitu panjang garis gerbang, ketebalan oksida gerbang, dan tegangan suplai untuk mencapai kepadatan transistor yang tinggi, kecepatan tinggi, dan pengurangan daya per operasi.

Ini adalah skala yang sudah dikenal yang bergantung pada prinsip penskalaan yang dikembangkan oleh Dennard, yang memprediksi pengurangan dimensi dan tegangan yang proporsional satu sama lain, dan membutuhkan peningkatan linier dalam kinerja dan disipasi daya pada perangkat. Karena MOSFET telah direduksi ke skala panjang gerbang 100 nm dan lebih kecil, penskalaan klasik tidak lagi berlaku karena efek kanal pendek (SCE), yang menyebabkan kontrol perangkat yang buruk dan arus bocor yang tinggi.

Dampak lain yang relatif dikenal dari SCE (*Short-Channel Execution*) adalah penurunan tegangan ambang, di mana, seiring dengan dipersingkatnya panjang kanal MOSFET, tegangan ambang perangkat tersebut menurun. Sumber dan drain keluar dari kanal, menyebabkan berkurangnya kontrol elektrostatik sumber dan drain pada pembawa muatan kanal. Hilangnya tegangan ambang berarti perangkat akan mengalami kebocoran arus lebih banyak saat mati, dan akan menyala pada tegangan gerbang yang lebih rendah, sehingga menyebabkan arus bocor meningkat dan, akibatnya, disipasi daya statis meningkat. Ini adalah masalah pada desain daya rendah karena arus bocor subambang yang tumbuh secara eksponensial dengan tegangan ambang dapat menjadi sumber daya utama, dan manfaat miniaturisasi perangkat menjadi hilang. Efek berbahaya kedua dari pengecilan ukuran adalah DIBL (*Drain-Induced Bladder Loss*).

Medan listrik yang disebabkan oleh drain pada fabrikasi kanal pendek (berpotensi) memengaruhi penghalang potensial di sisi sumber secara signifikan, sehingga menurunkannya dan mendorong inversi pembawa muatan di kanal, meskipun tegangan gerbang rendah. Efek yang dihasilkan dari proses ini adalah peningkatan arus bocor dan kinerja keluaran yang buruk. Hal ini juga digunakan dalam DIBL untuk mengurangi tegangan ambang dan mengganggu kemampuan perangkat untuk mati secara tidak sengaja, sehingga memengaruhi rasio arus hidup terhadap arus mati (I ON/I OFF), parameter penting dalam logika digital. Daerah-daerah ini menghasilkan pembawa muatan panas, yang dapat merusak oksida gerbang atau terperangkap, menyebabkan variasi tegangan ambang dan keandalan perangkat jangka panjang.

Hal ini mengurangi umur perangkat dan menciptakan ketidakkonsistenan dalam perilaku operasinya; oleh karena itu, membuat prediksi dan konsistensi rangkaian menjadi sulit. Elemen kebocoran lain yang ditingkatkan melalui penskalaan adalah GIDL. Semakin kecil tumpang tindih antara drain dan gerbang, semakin tinggi medan listrik yang muncul di tepi drain ketika perangkat tidak beroperasi, dan hal ini dengan mudah memungkinkan penerowongan pita-ke-pita pembawa muatan, yang menyebabkan arus bocor. Orientasi kebocoran berkontribusi pada konsumsi energi stasioner, dan solusi untuk desain, seperti profil doping drain, diperlukan.

Berbagai pendekatan rekayasa telah diajukan dan dipraktikkan untuk mengatasi masalah pengecilan ukuran MOSFET. Jenis perangkat ini lebih efektif dalam mengurangi efek SCE (Short-Cell Error) dan juga meningkatkan kemiringan sub-threshold, sehingga sangat cocok untuk dimensi yang lebih kecil. Teknik lain, misalnya rekayasa LDD (*Low-Domain Depthreshold*), digunakan untuk mengurangi jumlah medan listrik di sekitar drain untuk mengurangi efek hot-carrier dan arus bocor. Profil yang dicapai dengan rekayasa kanal dari material gerbang logam ganda atau multi-logam dengan fungsi kerja yang berbeda dapat menawarkan profil medan listrik yang mengurangi DIBL (*Drain-Induced Bleeding Layer*) dan efek SCE lainnya.

Peningkatan oksida gerbang terbatas pada batas fisik, dan pengurangan kebocoran gerbang telah dicapai dengan menggunakan oksida ultra-tipis dan pengenalan material dielektrik high-k, yang memungkinkan pengendalian arus bocor dengan menggunakan oksida

kapasitif yang lebih tebal secara fisik untuk mengontrol kapasitansi. Struktur oksida heterogen dan struktur oksida gerbang bertumpuk digunakan untuk mengoptimalkan keandalan isolasi gerbang dan mengurangi disipasi daya. Terlepas dari pencapaian ini, masalah penskalaan MOSFET yang lebih kecil dari beberapa nanometer menghadirkan masalah variabilitas, karena variasi dalam kepadatan dopan, ketebalan lapisan oksida, dan kekasaran ujung garis secara negatif memengaruhi keseragaman perangkat dan hasil sirkuit. Selain itu, efek mekanika kuantum seperti penerowongan dan pengurungan pembawa muatan menempatkan batasan intrinsik pada kinerja perangkat dan membutuhkan model dan proses desain baru.

Saluran kecil, arus bocor yang lebih tinggi, dan masalah keandalan, termasuk degradasi pembawa muatan panas dan penerowongan oksida gerbang serta ketidakpastian pada skala nano, telah terbukti membuat pengurangan MOSFET menjadi upaya yang menantang. Tantangan ini telah menyebabkan pengembangan desain perangkat yang canggih, teknik doping, desain saluran, dan material baru. Inovasi lebih lanjut di bidang ini diperlukan untuk melanjutkan miniaturisasi perangkat sesuai dengan persyaratan sistem VLSI berdaya rendah dan berkinerja tinggi.

Tantangan-tantangan ini menggambarkan bagaimana desain MOSFET planar secara bertahap digantikan oleh desain transistor tiga dimensi, multigate, dan desain transistor baru, yang akan mengatasi keterbatasan model penskalaan konvensional. Perspektif internasional dari permasalahan yang sedang dipertimbangkan ini merupakan titik awal untuk membahas desain perangkat yang lebih baik dan teknologi baru yang dibutuhkan oleh desain VLSI berdaya rendah di masa mendatang.

## 1.7 EFEK SALURAN PENDEK

Efek saluran pendek (SCE) adalah sekelompok proses fisik yang muncul pada MOSFET dengan panjang saluran skala nanometer yang disebabkan oleh kemajuan tak terhindarkan dari Hukum Moore dan tujuan peningkatan kepadatan integrasi, kinerja yang lebih tinggi, dan penggunaan daya yang lebih rendah dalam sistem VLSI. Perangkat saluran panjang sepenuhnya mengontrol saluran dengan medan elektrostatik, dan pergerakan pembawa muatan antara sumber dan drain dikontrol oleh tegangan gerbang, yang digunakan untuk mengontrol tegangan ambang aktivasi atau deaktivasi perangkat. Ide mendasar ini gagal bekerja secara efektif ketika panjang saluran menjadi lebih dekat dengan lebar area deplesi sumber dan drain.

Sumber dan drain didekatkan satu sama lain, menyebabkan interaksi langsung dengan penghalang saluran; potensial drain lebih kritis pada penghalang daripada potensial gerbang, dan karenanya gerbang memiliki kemampuan terbatas untuk mengontrol elektrostatik. Kekurangan kinerja seperti penurunan tegangan ambang, kebocoran subambang yang besar, penipisan penghalang yang diinduksi oleh drain (DIBL), efek pembawa panas, tembusan, hamburan permukaan, dan saturasi mobilitas pembawa dapat digunakan untuk mendefinisikan SCE. Efek buruk lainnya adalah DIBL, di mana peningkatan tegangan drain akan memungkinkan penghalang saluran terbuka untuk memungkinkan arus mengalir dari sumber ke drain dan karenanya meningkatkan kebocoran serta mengurangi keandalan perangkat dan

margin kebisingan. Medan listrik lateral yang tinggi pada drain dapat digunakan untuk mempercepat pembawa muatan ke energi tinggi, menyebabkan terciptanya elektron panas yang masuk ke oksida dan menyebabkan pengisian daya, yang mengakibatkan masalah keandalan jangka panjang berupa pergeseran tegangan ambang dan degradasi oksida. Efek terburuk dan paling tidak diinginkan adalah tembusan (*punch-through*), ketika area deplesi sumber dan drain tumpang tindih membentuk sumber bocor, terlepas dari nilai tegangan gerbang.

Kombinasi kekurangan ini mengakibatkan penurunan daya statis yang sebagian besar ditentukan oleh arus subambang dan kebocoran gerbang. Arus ini diketahui menjadi perhatian mendasar bagi node VLSI tercanggih, sama pentingnya dengan energi switching dinamis. Untuk mencapai solusi desain dan material yang dibutuhkan, SCE memerlukan adopsi arsitektur perangkat baru, seperti FET gerbang ganda, gerbang tiga, dan GAA, di mana satu atau beberapa elektroda gerbang digunakan untuk mengisolasi saluran, memungkinkan saluran tersebut dikontrol secara elektrostatik dan mode kebocoran yang tidak diinginkan diminimalkan, menggunakan teknologi modern. Kanal dan daerah sumber/drainase yang didoping ringan atau kanal dan daerah sumber/drainase yang diimplan halo adalah teknik utama yang digunakan dalam mendesain mitigasi efek pembawa panas dan tembusan. Selain itu, ketebalan oksida dielektrik dapat memiliki nilai besar bersamaan dengan dielektrik high-k dan gerbang logam dalam tumpukan gerbang untuk meminimalkan kebocoran terowongan langsung, sambil tetap memberikan kontrol gerbang, yang pada gilirannya secara substansial mengurangi kebocoran gerbang dan juga memungkinkan pengurangan progresif ketebalan oksida ekuivalen pada node yang semakin maju.

Oleh karena itu, para ilmuwan telah berkonsentrasi pada peningkatan arsitektur perangkat dan metode produksi untuk mengurangi SCE, dengan menggunakan kanal yang tidak didoping atau didoping ringan dan perangkat tanpa sambungan, serta pada material yang secara alami menguntungkan untuk perilaku transportasi pembawa terbaik. SCE pada MOSFET menciptakan ketidaknyamanan yang signifikan dalam desain VLSI dalam teknologi modern; studi tentangnya merupakan penekanan utama dalam desain arsitektur dan dalam manufaktur. Desain arsitektur sinergis di masa depan dengan material baru, profil doping yang dioptimalkan, dan litografi canggih akan menghindari efek kanal pendek (SCE), memungkinkan penskalaan VLSI lebih lanjut, meningkatkan efisiensi energi, dan mendukung komputasi berkinerja tinggi dan IoT yang meluas.

## 1.8 TEKNIK YANG DIGUNAKAN UNTUK MENGONTROL EFEK KALURAN PENDEK

Salah satu masalah utama dari miniaturisasi teknologi MOSFET yang terus meningkat dalam teknologi VLSI canggih modern adalah mitigasi SCE: efek ini sangat menghambat kinerja perangkat, meningkatkan arus bocor, dan memperburuk integritas sirkuit. Akibatnya, banyak solusi telah muncul dalam upaya untuk meningkatkan tingkat integritas elektrostatik, material, dan bentuk peralatan. Banyak topologi berbeda (misalnya, gerbang ganda, gerbang tiga (FinFET), dan MOSFET GAA) telah terbukti memiliki kopling gerbang-saluran yang lebih unggul daripada MOSFET planar konvensional, di mana elektroda gerbang sejajar dengan satu

permukaan kristal. Interaksi yang lebih besar antara gerbang dan saluran mengurangi pembagian muatan antara sumber dan drain, sehingga menghilangkan penurunan tegangan ambang, DIBL, dan kebocoran subambang. MOSFET, termasuk yang memiliki dua gerbang, memiliki gerbang bawah dan atas dan disandingkan dengan perangkat FinFET yang memiliki gerbang pembungkus tiga sisi di sekitar saluran untuk meningkatkan mobilitas pembawa dan fungsi keseluruhan.

Pemberantasan SCE filamen memerlukan rekayasa sumber-drain, yang mengurangi tingkat maksimum medan listrik pada sambungan, pemicu efek pembawa panas dan tembusan. Struktur LDD melunakkan profil doping sumber dan drain, menghilangkan lonjakan medan listrik, memengaruhi ionisasi, dan berpartisipasi dalam degradasi pembawa panas padat. Dalam pendekatan doping yang telah dikembangkan, seperti halo-/pocket-doping, wilayah yang luas di sekitar tepi kanal didoping secara kuat untuk memungkinkan sebagian besar distribusi medan listrik, dan juga menghilangkan efek DIBL dan punch-through.

Rekayasa kanal adalah salah satu metode di mana sejumlah besar material gerbang dengan berbagai fungsi kerja ditumpangkan pada satu elektroda gerbang. Ini disebut sebagai dual-metal atau triple-metal gating, yang meningkatkan variasi longitudinal penggerak kerja kanal, mengurangi SCE dengan membatasi gangguan medan listrik melalui modulasi drain. Kedekatan tingkat fungsi kerja gerbang dengan drain menekan medan listrik lokal, sehingga mengurangi arus bocor dan meningkatkan integritas perangkat.

Rekayasa substrat juga telah ditingkatkan ke arah peningkatan elektrostatika dan untuk menghilangkan kebocoran substrat. Substrat SOI mengisolasi bagian aktif substrat, sehingga mengurangi kapasitansi parasit dan arus bocor. MOSFET berbasis SOI memiliki sifat yang sepenuhnya terkuras dan menunjukkan sifat yang lebih tajam di wilayah subambang, dan mampu menekan SCE (*Short-Cell Exchange*). Selain itu, saluran terkait material lain seperti III–V, material silikon yang mengalami regangan juga telah diusulkan untuk mengoptimalkan transportasi pembawa muatan dan meminimalkan hamburan untuk membatasi degradasi karena SCE. Integrasi Dielektrik, desain gerbang logam dan integrasi dielektrik high-K meningkatkan optimasi sistem gerbang-dielektrik. Teknologi yang berbeda, berdasarkan substitusi silikon dioksida konvensional dengan material dielektrik tinggi, memungkinkan penggunaan oksida yang lebih tebal secara fisik tanpa menurunkan kapasitansi gerbang, dan menghilangkan kebocoran terowongan langsung.

Gradien dielektrik merupakan dielektrik gerbang yang dikompromikan, dielektrik gerbang bertumpuk atau komposit. Peningkatan simultan oksida gerbang dan arsitektur antarmuka cacat rendah menyebabkan peningkatan kontrol saluran dan pengurangan kebocoran tanpa keruntuhan tegangan ambang. Bahkan teori-teori baru tentang perangkat ini, seperti transistor tanpa sambungan (*junctionless transistor*), tidak mengandung sambungan sumber-drain yang tiba-tiba, melainkan memiliki saluran yang sepenuhnya didoping yang dialirkan oleh gerbang, bukan oleh penghalang inheren.

Paradigma desain ini membantu dalam pembuatan instrumen (produksi) serta dalam peningkatan kinerja saluran pendek (pengurangan fluktuasi doping dan kontrol elektrostatik). Pendekatan yang kompleks diperlukan dalam mengurangi efek saluran pendek (SCE), yang

melibatkan desain multigate canggih, desain gating canggih, optimasi doping gating dan substrat, distribusi fungsi kerja yang disesuaikan dalam desain gerbang, dan dielektrik gerbang berkualitas tinggi. Semua teknik tersebut meningkatkan kontrol saluran, mengurangi efek medan listrik yang tidak diinginkan, mengurangi arus bocor, dan juga memungkinkan penskalaan lebih lanjut dari sistem VLSI saat ini yang menggunakan MOSFET dalam skala nanometer tanpa mengorbankan kinerja, daya, atau keandalan.

## 1.9 EVOLUSI DAN MASA DEPAN TEKNOLOGI VLSI BERDAYA RENDAH

Pengecilan ukuran arsitektur MOSFET yang terus-menerus dan peningkatan yang terkait telah memfasilitasi pengembangan teknologi VLSI berdaya rendah, terutama karena keinginan untuk mencapai peningkatan kepadatan perangkat, peningkatan kinerja operasional, dan, pada saat yang sama, untuk mencapai konsumsi daya yang lebih rendah dalam sirkuit terpadu. Tren ini menghadapi tantangan fisik dan teknis yang serius, terutama ketika memperpanjang peralatan hingga skala nano yang sangat kecil, efek skala kecil (SCE), kebocoran, ketidakpastian, dan keandalan semuanya menjadi topik hangat.

Namun, telah ada kemajuan signifikan dalam hal menekan efek ini dan mempertahankan laju Hukum Moore dengan pengenalan arsitektur perangkat baru, rekayasa material, dan proses manufaktur. Desain multigate (termasuk FinFET, trigate, dan desain GAA) telah secara signifikan meningkatkan kontrol elektrostatik, sehingga menghasilkan kebocoran yang lebih rendah dan arus penggerak yang lebih tinggi, dan pada akhirnya menurunkan disipasi daya total dalam keadaan stabil dan memungkinkan penskalaan tegangan yang lebih agresif. Demikian pula, pendekatan doping tingkat lanjut dan metode rekayasa sumber/drainase mengurangi medan listrik puncak, sehingga mengurangi degradasi pembawa panas dan mengurangi SCE.

Kombinasi dielektrik gerbang high-k dan elektroda gerbang logam memungkinkan penggunaan oksida gerbang yang lebih tebal secara fisik, tanpa memengaruhi kapasitansi gerbang secara negatif, memungkinkan pengurangan besar pada arus bocor gerbang, dan peningkatan stabilitas perangkat. Perangkat NVM tersebut menurunkan penggunaan daya serta memungkinkan implementasi arsitektur komputer baru, termasuk komputasi dalam memori dan neuromorfik. Namun, masih ada masalah serius terkait manajemen variabilitas, jaminan keandalan, dan linearitas dalam hal penggabungan material baru dan desain perangkat baru ke dalam platform CMOS yang sudah mapan.

Inovasi dalam VLSI di masa depan kemungkinan akan berpusat pada pengembangan teknologi multigate, eksplorasi material kanal baru, dan manufaktur sinergis teknologi baru dalam memori dan logika, dengan tujuan untuk menciptakan efisiensi daya dan kemampuan komputasi baru yang belum pernah ada sebelumnya. Sifat multidomain dari inovasi-inovasi ini, termasuk fisika perangkat, desain sirkuit, dan arsitektur, akan memainkan peran sentral dalam pengembangan sistem VLSI berdaya rendah di masa depan, sehingga memungkinkan implementasi elektronik portabel di pusat data skala besar dan sistem kecerdasan buatan.

# BAB 2
# MEREVOLUSI DESAIN VLSI DALAM KECERDASAN BUATAN

## 2.1 INTEGRASI KECERDASAN (AI)

Integrasi Kecerdasan Buatan (AI) ke dalam alur kerja desain chip menyebabkan evolusi transformatif di sektor semikonduktor. Kerumitan sirkuit terpadu kontemporer yang semakin meningkat menghadirkan tantangan yang semakin besar bagi metodologi desain VLSI tradisional, yang secara tradisional bergantung pada optimasi yang berpusat pada ahli dan heuristik manual. Permintaan akan chip yang sangat efisien dan hemat daya telah meningkat pesat karena perkembangan teknologi yang dapat dikenakan, komputasi mobile, sistem otonom, dan *Internet of Things* (IoT). AI memperkenalkan paradigma yang terukur, adaptif, dan cerdas yang mengungguli teknologi tradisional dengan memungkinkan perancang untuk memproses sejumlah besar informasi, menemukan pola yang rumit, dan memperoleh wawasan prediktif dengan akurasi dan efisiensi yang lebih besar.

Konvergensi AI dan desain VLSI ini mewakili pergeseran paradigma, bukan hanya peningkatan. AI memberdayakan para desainer untuk mencapai iterasi desain yang dipercepat, menggunakan metodologi adaptif waktu nyata yang sebelumnya tidak praktis, dan secara bersamaan mengoptimalkan daya, kinerja, dan luas area (PPA). AI memfasilitasi otomatisasi, analitik prediktif, dan pengambilan keputusan cerdas di seluruh proses desain VLSI dengan mengintegrasikan algoritma canggih dengan metodologi yang sadar perangkat keras. Selain itu, integrasi ini mengurangi laju penyesuaian manual, mempercepat inovasi, dan membantu memenuhi persyaratan energi yang ketat dari sistem elektronik kontemporer. Bab ini dimulai dengan membahas tantangan inti yang terkait dengan perancangan sirkuit VLSI yang hemat daya.

Masalah-masalah ini meliputi disipasi daya karena aktivitas switching, kebocoran sub-threshold, arus hubung singkat, dan inefisiensi dalam distribusi clock. Seiring dengan peningkatan skala teknologi di luar node 10 nm, efek kuantum, variasi proses, dan peningkatan latensi interkoneksi semakin memperburuk masalah yang berkaitan dengan daya. Dengan memanfaatkan kemampuannya dalam pembelajaran mesin, pengenalan pola, dan pengambilan keputusan cerdas, AI memperkenalkan solusi baru dengan menganalisis data desain, mengenali pola tersembunyi, dan mengoptimalkan parameter yang sulit disesuaikan secara manual. Dalam sintesis logika dan optimasi *Register Transfer Level* (RTL), teknik AI seperti pembelajaran terawasi dan pembelajaran penguatan digunakan untuk menyempurnakan parameter guna mencapai keseimbangan daya, luas, dan penundaan yang optimal. Misalnya, agen pembelajaran penguatan dapat dilatih untuk melakukan transformasi sintesis sesuai dengan tujuan desain tertentu, sehingga menghasilkan konvergensi desain yang lebih baik dan ditingkatkan.

Dari perspektif desain fisik—khususnya dalam mencakup penempatan, perutean, dan perencanaan tata letak—model AI, termasuk jaringan saraf konvolusional (CNN) dan jaringan saraf grafik (GNN), digunakan untuk mengantisipasi kemacetan perutean, ketidakaturan

termal, dan jalur kritis dengan tepat. Prediksi ini memungkinkan perancang untuk melakukan penyesuaian awal yang mengurangi pemborosan daya dan meningkatkan manajemen termal. Aplikasi utama AI dalam VLSI adalah dalam estimasi dan pemodelan daya. Alat estimasi daya tradisional membutuhkan banyak komputasi, memerlukan banyak sumber daya, dan beroperasi sangat lambat. Model regresi berbasis AI, yang dilatih pada kumpulan data besar dari desain sebelumnya, dapat memprediksi penggunaan konsumsi daya dengan presisi tinggi dan biaya komputasi minimal. Model-model ini memungkinkan perancang untuk membuat pilihan yang hemat daya sejak dini dalam fase desain tanpa harus melakukan simulasi skala penuh, sehingga menghemat waktu dan fasilitas.

Bab ini juga membahas gagasan pencarian arsitektur neural yang sadar perangkat keras (NAS), di mana AI digunakan tidak hanya untuk mengembangkan model neural tetapi juga untuk mengoptimalkan kinerjanya berdasarkan keterbatasan perangkat keras tertentu, seperti efisiensi energi dan latensi. Pendekatan ini sangat penting dalam desain Sirkuit Terpadu Khusus Aplikasi (ASIC) dan chip AI tepi, di mana mencapai konsumsi daya yang sangat rendah sangat penting.

Dengan mengintegrasikan AI ke dalam desain algoritmik dan pengembangan perangkat keras, kerangka kerja desain bersama ini memastikan bahwa produk akhir efektif secara fungsional dan hemat energi. Lebih lanjut, bab ini menyoroti kemajuan alat Otomasi Desain Elektronik (EDA) dengan menggunakan teknik berbasis AI, yang mencakup bagaimana algoritma cerdas dapat memberikan arahan pada alur kerja desain, meningkatkan presisi, dan meminimalkan beban komputasi. Alat EDA kontemporer semakin mengintegrasikan modul AI untuk mengotomatiskan tugas-tugas kompleks seperti verifikasi, sintesis tata letak, dan analisis waktu. Integrasi AI secara signifikan mempercepat penutupan desain dan mengurangi ketergantungan pada intervensi manual, sehingga meningkatkan produktivitas dan mengurangi siklus waktu pemasaran.

Terlepas dari poin-poin positif ini, adopsi AI memfasilitasi deteksi anomali cerdas di antara fase pengujian dan verifikasi, meningkatkan hasil dan keandalan. Terlepas dari keuntungan-keuntungan ini, integrasi AI dalam desain Integrasi Skala Sangat Besar (VLSI) memperkenalkan serangkaian kompleksitas teknis dan tantangan implementasi yang memerlukan pertimbangan dan minimalisasi yang cermat. Kekhawatiran utama meliputi kemampuan menjelaskan model AI, kebutuhan akan sejumlah besar data pelatihan berkualitas tinggi, kemampuan untuk melakukan generalisasi di berbagai node teknologi, dan risiko overfitting.

Selain itu, model AI harus menunjukkan ketahanan terhadap input yang merugikan, terutama dalam aplikasi yang sensitif terhadap keamanan. Bab ini juga menyajikan aspek etika dan praktis seperti keterbukaan, kerahasiaan data, dan konsumsi energi yang terlibat dalam pelatihan model AI besar. Untuk memberikan perspektif praktis, bab ini menyertakan studi kasus ilustratif dari akademisi dan industri yang menunjukkan bagaimana AI telah menghasilkan kemajuan signifikan dalam efisiensi daya dan otomatisasi desain.

Ilustrasi mencakup penerapan teknik pembelajaran penguatan (*reinforcement learning*) oleh Google untuk perencanaan tata letak (*floorplanning*), inisiatif akademis yang

menerapkan AI untuk prediksi kebocoran, dan platform EDA komersial yang menyematkan modul pembelajaran mesin untuk optimasi daya. Setiap contoh mengilustrasikan manfaat nyata dari pengintegrasian AI dengan metodologi VLSI tradisional. Singkatnya, bab ini menyatakan bahwa AI sedang bertransformasi dari instrumen tambahan dalam desain VLSI menjadi komponen fundamental inti dari inovasi chip masa depan. Dengan menyematkan kecerdasan dalam alur kerja desain, AI mendorong iterasi yang lebih cepat, pengambilan keputusan yang tepat, dan hasil yang lebih efisien.

Khususnya dalam aplikasi berdaya rendah, AI memberdayakan perancang untuk mencapai kinerja maksimum sambil tetap mematuhi batasan energi. Seiring algoritma AI terus menjadi lebih kuat dan terintegrasi, dan seiring desain chip menjadi semakin berorientasi pada aplikasi dan kompleks, kolaborasi antara AI dan VLSI akan menjadi sangat penting untuk mengembangkan generasi sistem cerdas dan hemat energi berikutnya. Bab ini membahas aplikasi praktis AI dalam berbagai tahapan desain VLSI, termasuk sintesis logika, optimasi tata letak fisik, estimasi daya, dan peningkatan alat EDA, untuk mengilustrasikan bagaimana AI merevolusi desain VLSI berdaya rendah. Melalui contoh-contoh terpilih dari akademisi dan industri, yang menyoroti peningkatan efisiensi daya, otomatisasi desain, dan keandalan sistem secara keseluruhan.

**Munculnya AI dalam Desain Semikonduktor**

Integrasi AI ke dalam desain semikonduktor menandai pergeseran besar dari praktik desain konvensional. Secara tradisional, insinyur VLSI menyeimbangkan trade-off antara PPA melalui penyetelan manual, penyesuaian berbasis aturan, dan strategi heuristik. Karena chip modern menjadi semakin kompleks dan ukuran fiturnya menyusut di bawah 10 nm, metode tradisional ini kesulitan untuk mempertahankan efisiensi, skalabilitas, dan keandalan. Pendekatan berbasis AI mengatasi tantangan ini dengan belajar dari data desain masa lalu, mengenali pola kinerja, dan menyarankan konfigurasi desain yang dioptimalkan secara lebih efektif daripada teknik yang digerakkan manusia.

**Evolusi Desain VLSI Daya Rendah**

Selama beberapa dekade terakhir, prinsip-prinsip desain daya rendah telah berkembang pesat. Pada tahap awalnya, strategi desain terutama berfokus pada pengurangan aktivitas switching, pengelolaan arus bocor, dan penerapan metode penskalaan tegangan yang sederhana. Seiring kemajuan teknologi semikonduktor, masalah seperti penundaan interkoneksi, arus hubung singkat, dan kebocoran sub-ambang menjadi lebih menonjol. Desain VLSI daya rendah modern kini bertujuan tidak hanya untuk mengurangi konsumsi energi tetapi juga untuk memastikan stabilitas termal, keandalan, dan kinerja yang konsisten di seluruh perangkat. Dengan mengintegrasikan AI ke dalam proses ini, perancang dapat menerapkan teknik optimasi adaptif berbasis data untuk mencapai efisiensi daya yang lebih tinggi, bahkan dalam arsitektur *System-on-Chip* (SoC) yang kompleks yang beroperasi di bawah anggaran energi yang ketat.

**Peran AI dalam Sistem Elektronik Modern**

AI telah muncul sebagai komponen penting dalam desain dan optimasi sistem elektronik modern. Teknik seperti pembelajaran mesin, pembelajaran penguatan, dan

pembelajaran mendalam kini diterapkan di berbagai tahapan alur desain VLSI, termasuk sintesis logika, optimasi RTL, penempatan, perutean, dan perencanaan tata letak. Sebelum fabrikasi, model berbasis AI mampu memprediksi potensi hambatan perutean, titik panas termal, dan inefisiensi daya, memungkinkan para insinyur untuk melakukan penyesuaian awal yang meningkatkan kinerja dan efisiensi energi. Dalam praktiknya, alat EDA yang dibantu AI juga mengurangi intervensi manual, meningkatkan hasil desain, dan mempersingkat waktu pemasaran untuk produk semikonduktor yang semakin kompleks.

**Ruang Lingkup dan Tujuan Bab Ini**

Tujuan utama bab ini adalah untuk memberikan analisis komprehensif tentang pendekatan berbasis AI untuk desain VLSI berdaya rendah dan untuk meneliti bagaimana teknologi ini mentransformasi industri semikonduktor. Bab ini berfokus pada beberapa tujuan utama:

- Menganalisis tantangan yang terkait dengan perancangan sirkuit terpadu hemat energi dalam node teknologi canggih.
- Mendemonstrasikan peran AI dalam meningkatkan alat EDA, mengoptimalkan proses desain logika dan fisik, dan memungkinkan pemodelan daya prediktif.
- Menyajikan studi kasus yang melaporkan peningkatan terukur dalam otomatisasi desain, keandalan, dan efisiensi daya.
- Mendiskusikan keterbatasan adopsi AI, khususnya mengenai ketersediaan data, interpretasi model, dan generalisasi di berbagai teknologi fabrikasi.
- Mengatasi pertimbangan praktis dan etis, termasuk privasi data, transparansi, dan implikasi energi dari pelatihan model AI skala besar.
- Dengan mengeksplorasi dimensi-dimensi ini, bab ini memposisikan AI sebagai kekuatan strategis dan transformatif dalam alur kerja VLSI modern dan masa depan, terutama dalam konteks aplikasi desain hemat daya dan hemat energi.

## 2.2 DASAR-DASAR DESAIN VLSI DAYA RENDAH

Komponen penting dalam pengembangan semikonduktor kontemporer adalah desain VLSI. Mengurangi penggunaan energi sambil mempertahankan kinerja dan keandalan sangat penting karena perangkat menjadi lebih kecil dan lebih kaya fitur. Elemen dasar konsumsi daya, pendekatan desain konvensional, kesulitan yang dihadirkan oleh node teknologi sub-10 nm, dan pentingnya desain hemat daya dalam SoC dan perangkat edge kontemporer semuanya dibahas dalam bagian ini.

**Komponen Konsumsi Daya: Switching, Kebocoran, Hubung Singkat**

Tiga komponen utama dapat digunakan untuk mengklasifikasikan secara luas disipasi daya dalam sirkuit VLSI:

- **Perubahan Daya**: Ketika transistor CMOS berganti antara keadaan logika, daya switching dihasilkan. Ini memiliki hubungan langsung dengan kuadrat faktor aktivitas, kapasitansi beban, dan tegangan suplai. Komponen ini menyumbang sebagian besar konsumsi daya dinamis dalam sirkuit aktif.

- **Kebocoran Daya**: Konduksi di bawah ambang batas, kebocoran gerbang, dan kebocoran sambungan pada transistor merupakan sumber daya kebocoran. Kebocoran menjadi masalah serius seiring kemajuan teknologi dan penurunan tegangan ambang batas, terutama pada node sub-10 nm yang canggih.
- **Daya Korsleting**: Ketika transistor NMOS dan PMOS melakukan konduksi secara bersamaan selama transisi sinyal, daya korsleting dihasilkan. Pada sistem frekuensi tinggi, hal ini tidak dapat diabaikan, karena kontribusinya lebih kecil daripada daya switching dan kebocoran.

Secara keseluruhan, daya switching secara historis mendominasi node teknologi sebelumnya, tetapi penskalaan agresif dan tegangan ambang batas yang lebih rendah pada perangkat sub-10 nm telah menjadikan daya kebocoran sebagai perhatian yang signifikan. Tabel 2.1 menunjukkan komponen disipasi daya dalam VLSI, di mana daya switching mendominasi sirkuit aktif, daya kebocoran menjadi kritis dalam mode skala dan idle, dan daya korsleting, meskipun kecil, berdampak pada desain frekuensi tinggi.

**Tabel 2.1** Komponen Disipasi Daya dalam VLSI

| Komponen daya | Menyebabkan | Faktor kunci | Dampak pada desain |
|---|---|---|---|
| Daya switching | Transisi keadaan CMOS | Faktor aktivitas, kapasitansi, tegangan suplai | Dominan pada sirkuit aktif, hal ini dapat dikurangi dengan clock gating. |
| Daya bocor | Kebocoran di bawah ambang batas dan gerbang | Tegangan ambang, ukuran transistor, suhu | Meningkat seiring dengan penskalaan, sangat penting dalam mode idle/standby. |
| Daya hubung singkat | Konduksi simultan NMOS dan PMOS | Waktu naik/turun, tegangan suplai | Kontribusi kecil, memengaruhi sirkuit frekuensi tinggi |

**Teknik Desain Tradisional dan Keterbatasannya**

Untuk menurunkan konsumsi daya, proses desain VLSI sebelumnya sebagian besar bergantung pada teknik heuristik dan berbasis aturan. Metode umum meliputi:

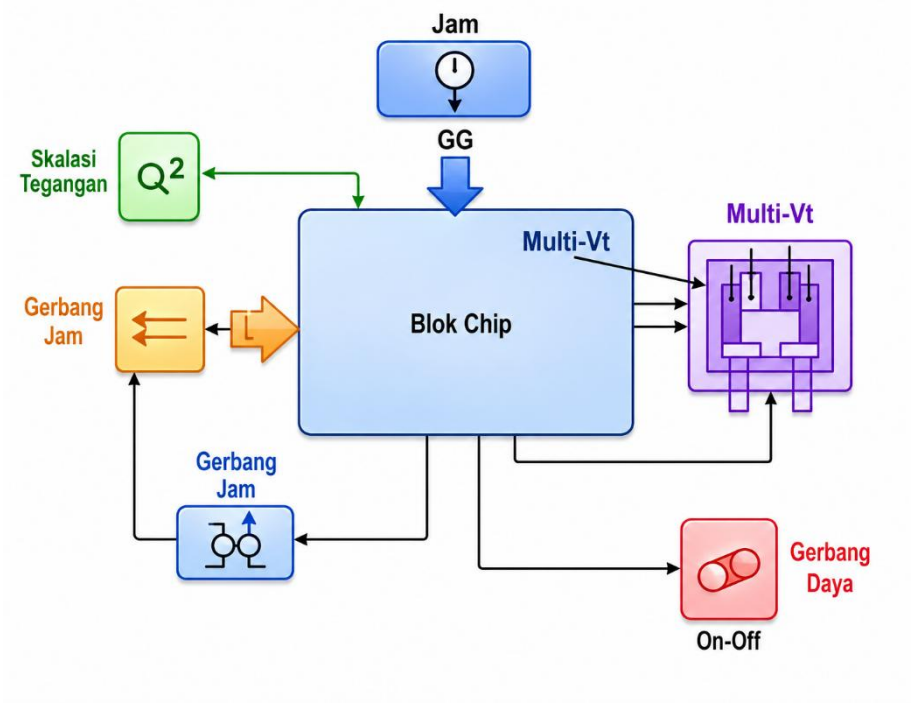
- **Penskalaan Tegangan**: Daya dinamis dikurangi secara kuadratik dengan menurunkan tegangan suplai, tetapi kinerja seringkali memburuk.
- **CMOS Multi-Threshold (MTCMOS)**: Untuk mengurangi daya bocor, transistor Vt tinggi digunakan pada jalur non-kritis.
- **Pengaktifan Jam**: Untuk menghemat daya dinamis, blok fungsional yang tidak aktif diputus dari jam.
- **Pengaktifan Daya**: Untuk mencegah kebocoran saat siaga, semua blok yang tidak digunakan dimatikan.

**Kekurangan Metode Tradisional**

- Optimasi manual membutuhkan banyak pekerjaan dan rentan terhadap kesalahan.

- Metode-metode ini kehilangan efektivitasnya pada SoC yang sangat kompleks dan pada node yang lebih kecil dari 10 nm.
- Metode-metode ini tidak mampu menyesuaikan diri dengan perubahan beban kerja dan keterbatasan daya secara real-time.

Gambar 2.1 mengilustrasikan beberapa teknik tradisional yang digunakan untuk mengurangi konsumsi daya pada sirkuit VLSI. Gambar ini menunjukkan bagaimana metode seperti Voltage Scaling, Clock Gating, Multi-Vt, dan Power Gating diterapkan pada "Chip Block" untuk mengelola dan meminimalkan disipasi daya.




**Gambar 2.1** Teknik hemat daya tradisional dalam VLSI.

**Tantangan dalam Teknologi Node Sub-10 Nm**

Beberapa hambatan signifikan muncul akibat peningkatan skala teknologi node sub-10 nm:

- **Peningkatan Arus Bocor**: Tegangan ambang yang lebih rendah berpengaruh pada penggunaan energi siaga dengan meningkatkan daya bocor secara dramatis.
- **Variabilitas Proses**: Ketidaksempurnaan dalam fabrikasi menyebabkan variasi kinerja transistor, yang menurunkan hasil produksi dan menimbulkan ketidakpastian waktu.
- **Efek Saluran Pendek dan Kuantum**: Penerowongan dan penurunan penghalang yang diinduksi oleh drain (DIBL) adalah dua fenomena yang mengurangi keandalan perangkat.
- **Penundaan Interkoneksi**: Transmisi sinyal dalam interkoneksi sering kali diperlambat oleh efek resistansi-kapasitansi (RC), yang pada gilirannya berkontribusi pada penundaan waktu dan konsumsi energi yang lebih tinggi.

- **Titik Panas Termal**: Konsentrasi daya yang berlebihan menghasilkan titik panas termal yang menurunkan kinerja perangkat dan mengganggu stabilitas operasional dari waktu ke waktu.

Tabel 2.2 merangkum tantangan utama yang dihadapi dalam teknologi VLSI sub-10 nm, seperti peningkatan kebocoran, variabilitas proses, dan efek kanal pendek, penundaan interkoneksi, dan titik panas termal, yang semuanya memengaruhi efisiensi daya, hasil produksi, dan keandalan secara keseluruhan.

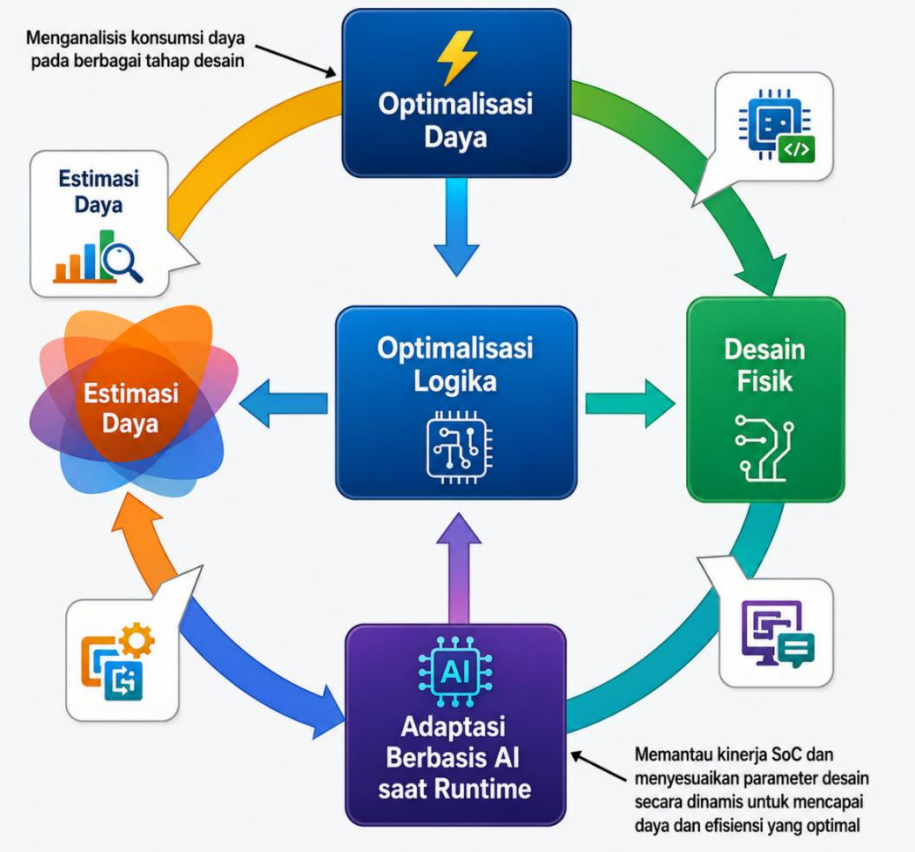
**Tabel 2.2** Tantangan utama dalam VLSI sub-10 nm

| Tantangan | Penyebab | Dampak |
|---|---|---|
| Peningkatan kebocoran | Vt rendah, oksida gerbang tipis | Daya siaga lebih tinggi |
| Variabilitas proses | Ketidaksempurnaan fabrikasi | Pengurangan hasil dan ketidakpastian waktu |
| Efek saluran pendek | Penskalaan perangkat | Pergeseran tegangan ambang, peningkatan kebocoran |
| Penundaan interkoneksi | Efek resistansi-kapasitansi | Perambatan sinyal yang lebih lambat |
| Titik panas termal | Peningkatan kepadatan daya | Penurunan keandalan dan kinerja |

**Pentingnya Desain Hemat Daya pada SoC dan Perangkat Edge**

Desain hemat daya telah menjadi sangat penting untuk SoC dan platform edge canggih, di mana para perancang harus menyeimbangkan kinerja, efisiensi, dan keandalan dalam batasan rekayasa yang ketat. Untuk mencegah penurunan kinerja dan mempertahankan operasi yang konsisten dalam tata letak chip yang padat, manajemen termal yang tepat sangat penting. Demikian pula, menurunkan konsumsi daya secara langsung memperpanjang masa pakai baterai pada sistem portabel. Dengan keberlanjutan yang kini menjadi prioritas desain utama, meminimalkan total penggunaan energi juga telah menjadi tujuan lingkungan dan ekonomi. Mencapai hasil ini di bawah anggaran daya yang ketat membutuhkan optimasi yang cermat di berbagai tingkat abstraksi dalam hierarki desain.

Kerangka kerja desain modern yang dibantu AI melengkapi metode heuristik tradisional dengan menerapkan penempatan dan perutean cerdas, penjadwalan beban kerja adaptif, dan model prediktif untuk titik panas daya dan suhu. Strategi berbasis AI ini memungkinkan pertukaran daya-kinerja-luas (PPA) yang lebih efektif, membantu SoC dan sistem edge modern mencapai keseimbangan yang diinginkan antara efisiensi dan kinerja yang dapat diskalakan. Gambar 2.2 menyajikan alur desain SoC yang menyematkan manajemen daya di seluruh siklus hidupnya—dari konsep awal hingga operasi runtime—menekankan lingkaran umpan balik antara fase desain dan adaptasi sistem secara real-time.




**Gambar 2.2** Alur desain hemat daya untuk SoC modern.

## 2.3 INTEGRASI KECERDASAN BUATAN DALAM DESAIN VLSI

Dalam beberapa tahun terakhir, desain sirkuit VLSI menjadi semakin kompleks, dengan para insinyur menghadapi persyaratan daya-kinerja-luas (PPA) yang lebih ketat daripada sebelumnya. Tantangan yang semakin meningkat ini telah menyebabkan integrasi AI secara bertahap ke hampir setiap tahap proses desain. Teknik heuristik tradisional, meskipun efektif untuk sistem skala kecil, kesulitan untuk mengimbangi tuntutan arsitektur modern. Pendekatan berbasis AI, di sisi lain, membawa otomatisasi, wawasan ke depan, dan pengambilan keputusan adaptif ke dalam alur kerja. Dengan memanfaatkan data desain yang ekstensif dan belajar dari pola optimasi masa lalu, AI memungkinkan perancang untuk menangani keterbatasan penskalaan secara lebih efisien, mempersingkat siklus pengembangan, dan mencapai kualitas desain keseluruhan yang lebih baik. Bab ini membahas metodologi AI utama yang sekarang diterapkan pada desain VLSI, relevansinya terhadap model pembelajaran mesin dan pembelajaran mendalam, isu-isu penting dalam persiapan dan pelatihan data, dan bagaimana praktik berorientasi AI berbeda dari metode desain konvensional dalam hal kemampuan adaptasi dan kinerja.

**Gambaran Umum Teknik AI dalam VLSI**

Seiring dengan peningkatan integrasi dan kompleksitas sirkuit VLSI, alur kerja EDA semakin mengadopsi metode berbasis AI untuk mengatasi kendala daya-kinerja-luas (PPA) yang ketat. Saat ini, teknik AI beroperasi di hampir setiap lapisan abstraksi—dari pemodelan tingkat perangkat hingga optimasi sistem secara keseluruhan—menawarkan otomatisasi,

adaptabilitas, dan akurasi prediksi yang tidak mudah diberikan oleh pendekatan heuristik konvensional.

(i) **Pembelajaran Terawasi**: Dalam pembelajaran terawasi, data desain historis dengan hasil yang diketahui digunakan untuk melatih model yang mampu memprediksi indikator kinerja utama seperti daya bocor, penundaan, hasil, dan keandalan. Algoritma klasifikasi dapat menandai potensi pelanggaran waktu, sedangkan model regresi memprediksi arus bocor di bawah variasi proses yang berbeda. Namun, keberhasilan metode ini sangat bergantung pada ketersediaan dan kualitas dataset besar yang diberi label dengan baik.

(ii) **Pembelajaran Tak Terawasi**: Ketika data berlabel langka atau tidak tersedia, pembelajaran tak terawasi membantu mengungkap pola yang tersembunyi dalam informasi desain yang kompleks. Teknik pengelompokan dapat mengelompokkan modul logika dengan karakteristik daya atau latensi yang serupa, sementara model deteksi anomali mengidentifikasi perilaku perutean yang tidak teratur yang nantinya dapat menyebabkan kemacetan. Metode pengurangan dimensi, termasuk Analisis Komponen Utama (PCA) dan autoencoder, lebih lanjut menyederhanakan data desain berdimensi tinggi, meningkatkan visualisasi dan kinerja tahap optimasi selanjutnya.

(iii) **Pembelajaran Penguatan (RL)**: Pembelajaran penguatan telah mendapatkan perhatian dalam tugas desain yang berat optimasi di mana agen belajar melalui percobaan dan umpan balik berulang. Dalam perencanaan tata letak dan penempatan, misalnya, agen RL bertujuan untuk meminimalkan kemacetan, panjang kawat, dan pelanggaran waktu secara bersamaan. Demonstrasi yang terkenal adalah perencanaan tata letak chip berbasis RL Google, yang mencapai tata letak yang lebih unggul daripada yang dihasilkan oleh para ahli manusia. Terlepas dari janjinya, RL tetap mahal secara komputasi dan membutuhkan rekayasa penghargaan yang cermat untuk memastikan konvergensi yang stabil.

(iv) **Pembelajaran Mendalam (DL)**: Pembelajaran mendalam memperluas potensi AI dengan memodelkan hubungan nonlinier yang kompleks dalam data desain skala besar. CNN membantu dalam analisis citra tata letak dan prediksi kemacetan, sementara GNN secara efektif merepresentasikan konektivitas sirkuit untuk memodelkan hubungan daya dan penundaan. Selain itu, model pengganti mendalam meniru simulasi fisik terperinci, secara signifikan mengurangi waktu komputasi dan memungkinkan eksplorasi alternatif desain dengan cepat. Namun, kebutuhan akan kumpulan data yang ekstensif dan sumber daya komputasi berkinerja tinggi terus membatasi penerapan industri skala besar.

Bersama-sama, paradigma AI ini membentuk kembali lanskap desain VLSI. Dengan menggabungkan otomatisasi dengan kecerdasan prediktif, mereka memungkinkan para perancang untuk mengelola tantangan yang ditimbulkan oleh node teknologi di bawah 10 nm. Tabel 2.3 merangkum peran teknik-teknik ini: pembelajaran terawasi dan tak terawasi membantu dalam prediksi dan penemuan pola, RL menyempurnakan perencanaan tata letak dan perutean, dan DL membantu mengurangi kemacetan dan hambatan termal—masing-

masing memberikan kekuatan unik sambil menghadirkan tantangan data dan komputasi tersendiri.

**Tabel 2.3** Teknik dan aplikasi AI dalam VLSI

| Teknik AI | Aplikasi dalam VLSI | Keuntungan | Tantangan |
|---|---|---|---|
| Pembelajaran terawasi | Estimasi daya, prediksi kebocoran | Prediksi akurat untuk data yang diketahui | Membutuhkan kumpulan data berlabel yang besar |
| Pembelajaran tanpa pengawasan | Pengelompokan desain, deteksi anomali | Mengidentifikasi pola tersembunyi | Sulit untuk menafsirkan hasil |
| Pembelajaran penguatan | Perencanaan tata ruang, optimasi perutean | Mempelajari strategi optimal secara mandiri | Membutuhkan komputasi intensif, memerlukan desain imbalan |
| Pembelajaran mendalam | Prediksi kemacetan, titik panas termal | Menangani data berdimensi tinggi | Membutuhkan kumpulan data besar dan sumber daya komputasi tinggi |
| Pembelajaran terawasi | Estimasi daya, prediksi kebocoran | Prediksi akurat untuk data yang diketahui | Membutuhkan kumpulan data berlabel besar |

**Peran Pembelajaran Mesin dan Pembelajaran Mendalam**

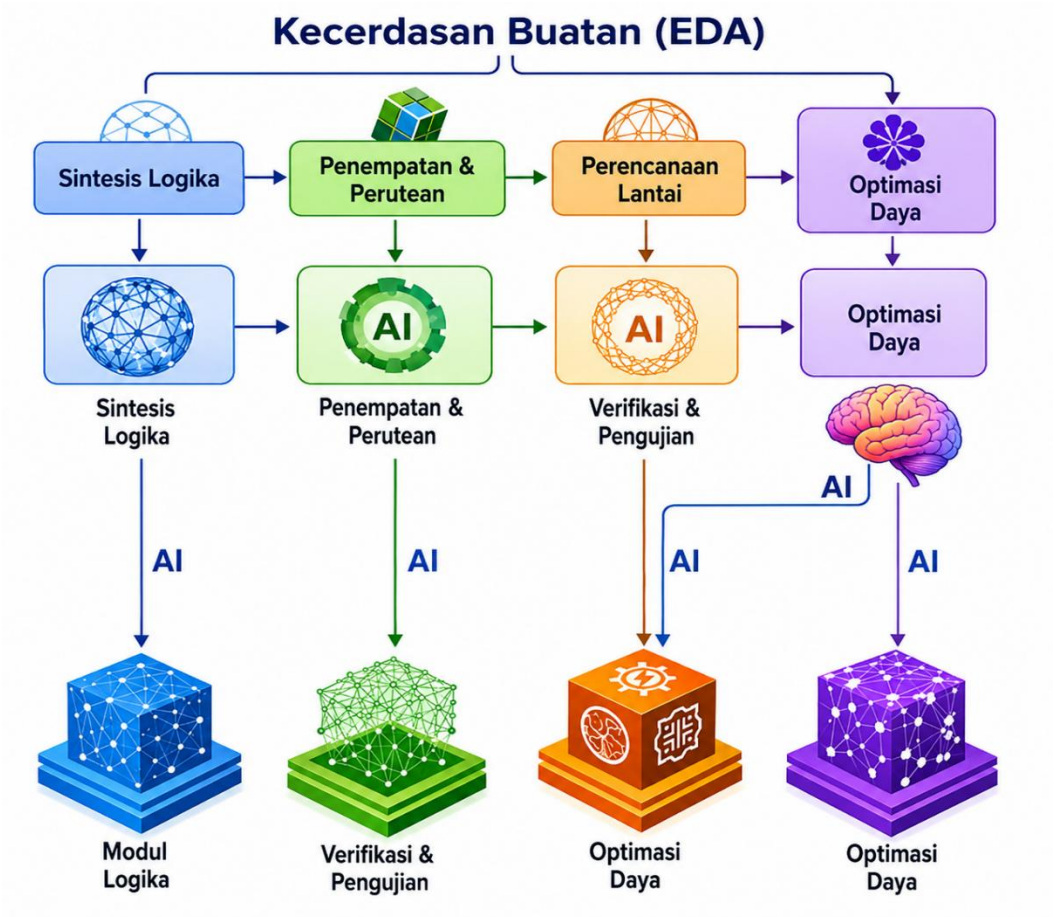
Desain VLSI modern semakin bergantung pada pembelajaran mesin (ML) dan pembelajaran mendalam (DL) untuk memungkinkan otomatisasi, optimasi adaptif, dan analitik prediktif di seluruh alur kerja desain. Salah satu aplikasi paling awal dan paling berdampak terletak pada pemodelan daya prediktif, di mana algoritma ML berbasis regresi memperkirakan daya switching, kebocoran, dan hubung singkat jauh sebelum simulasi sirkuit terperinci dilakukan. Estimasi awal ini secara signifikan mengurangi ketergantungan pada simulasi yang intensif secara komputasi.

Dalam optimasi logika dan RTL, model ML juga memandu penyetelan parameter sintesis untuk mencapai keseimbangan yang efektif antara latensi, area, dan konsumsi daya. Selama desain fisik, pendekatan DL—khususnya CNN dan GNN—mendukung deteksi proaktif pelanggaran waktu, kemacetan perutean, dan titik panas termal, memungkinkan koreksi lebih awal dan kualitas tata letak yang lebih tinggi.

Integrasi ML dan DL ke dalam alat EDA telah mengubah eksplorasi dan implementasi ruang desain. Platform Cerebrus dari Cadence menggunakan RL untuk menyempurnakan alur sintesis dan implementasi, sementara DSO.ai dari Synopsys mengintegrasikan modul ML untuk mengotomatiskan eksplorasi dan optimasi. Demikian pula, pekerjaan Google dalam RL mendalam untuk perencanaan tata letak chip telah menunjukkan hasil yang setara atau bahkan melampaui desain manusia ahli, baik dalam waktu penyelesaian maupun kualitas.

Secara kolektif, aplikasi dunia nyata ini menyoroti bagaimana strategi berbasis AI mendefinisikan ulang praktik EDA tradisional. Dengan menggabungkan akurasi prediktif dengan pembelajaran adaptif, mereka meningkatkan skalabilitas, mempercepat penutupan

desain, dan memberikan hasil daya-kinerja-luas (PPA) yang lebih baik dalam sistem VLSI canggih. Gambar 2.3 mengilustrasikan peran AI di berbagai tahapan proses EDA, menekankan kontribusinya terhadap peningkatan efisiensi, pengurangan siklus desain, dan metrik sistem yang dioptimalkan.




**Gambar 2.3** Aplikasi AI di seluruh alur desain VLSI.

**Persyaratan Data dan Pertimbangan Pelatihan**

Akurasi model ML dan DL dalam desain VLSI sangat bergantung pada keragaman, kelengkapan, dan keterwakilan data yang digunakan untuk pelatihan. Akibatnya, pembuatan dataset berkualitas tinggi membentuk dasar untuk aplikasi AI yang andal di domain ini. Informasi desain historis—seperti tata letak tata letak, tata letak routing, pengukuran daya, dan metrik kinerja—memberikan wawasan berharga tentang pola struktural dan listrik yang berulang dalam pengembangan SoC dan ASIC yang kompleks. Selain data desain yang diarsipkan, dataset berbasis simulasi yang dihasilkan dari model RTL atau SPICE sering digunakan untuk tugas pembelajaran terawasi, memungkinkan estimasi awal pelanggaran waktu, aktivitas switching, dan daya kebocoran tanpa perlu simulasi chip lengkap yang menyeluruh.

Persiapan data yang efektif sangat penting untuk memastikan kekokohan dan generalisasi. Teknik seperti normalisasi, ekstraksi fitur, pengurangan dimensi, dan penyaringan noise membantu menekankan atribut penting desain sekaligus meningkatkan skalabilitas

model. Melatih model AI untuk VLSI juga membutuhkan pertimbangan yang cermat terhadap interpretasi untuk menjaga transparansi selama verifikasi dan validasi desain. Selain itu, meminimalkan overfitting sangat penting untuk portabilitas di berbagai teknologi proses—misalnya, ketika transisi dari node 7 ke 3 nm—sambil menyeimbangkan biaya komputasi dengan peningkatan desain yang terukur.

Dataset benchmark terbaru semakin membentuk praktik pembahasan dan evaluasi. Sumber daya yang tersedia untuk umum seperti dataset kontes ISPD (untuk penempatan dan perutean) dan benchmark OpenROAD Flow (untuk optimasi desain fisik), bersama dengan dataset kepemilikan semi-terbuka seperti dataset floorplanning TPU Google, menyediakan testbed realistis yang mempersempit kesenjangan antara eksperimen akademis dan aplikasi industri. Dataset ini mendorong reproduksibilitas dan perbandingan kinerja yang adil, yang keduanya sangat penting untuk kemajuan dalam desain berbasis AI. Ketika kualitas data, pra-pemrosesan, dan prosedur pelatihan dikelola secara sistematis, alur kerja yang diaktifkan AI dapat mencapai peningkatan substansial dalam optimasi PPA. Tabel 2.4 merangkum pertimbangan data dan pelatihan utama untuk AI dalam VLSI—menyoroti skala dataset, pemilihan fitur, pengendalian overfitting, dan persyaratan komputasi sebagai penentu utama akurasi, efisiensi, dan skalabilitas model.

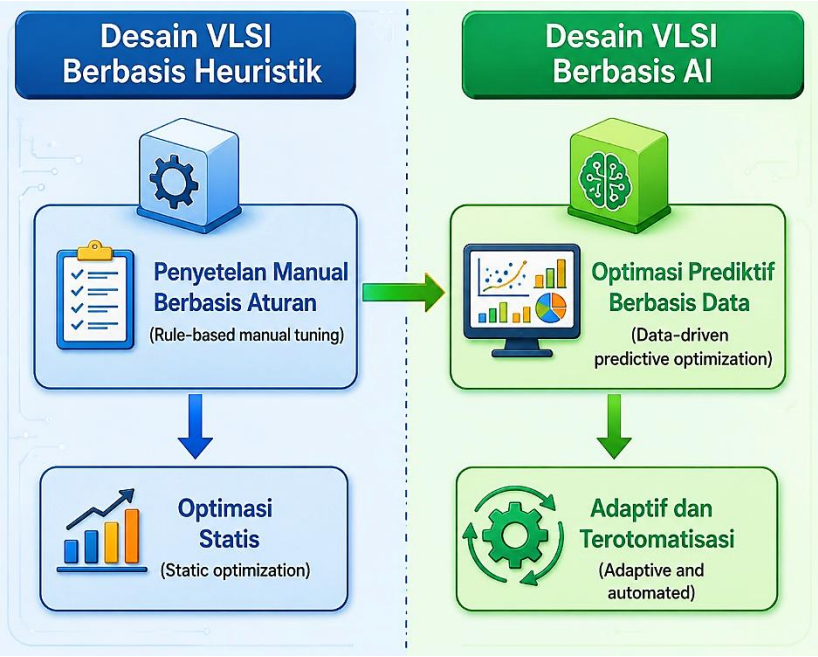
**Tabel 2.4** Pertimbangan data dan pelatihan untuk AI dalam VLSI

| Pertimbangan | Detail | Dampak pada Model AI |
|---|---|---|
| **Ukuran Dataset** | Dataset historis yang besar dan beragam | Meningkatkan akurasi serta kemampuan generalisasi model |
| **Pemilihan Fitur** | Parameter penting seperti kepadatan sel (*cell density*) dan aktivitas perpindahan (*switching activity*) | Mengurangi kompleksitas pelatihan dan meningkatkan efisiensi |
| **Overfitting** | Dicegah melalui regularisasi (*regularization*) atau validasi silang (*cross-validation*) | Memastikan model mampu bekerja dengan baik pada desain baru |
| **Sumber Daya Komputasi** | GPU atau akselerator khusus | Mempercepat proses pelatihan untuk dataset berukuran besar |

**AI Vs Heuristik Tradisional Dalam EDA**

Berbeda dengan pendekatan berbasis heuristik tradisional yang terutama bergantung pada aturan desain tetap dan intuisi manusia, AI memperkenalkan pengambilan keputusan adaptif dan berpusat pada data ke dalam desain VLSI. Adopsi AI menawarkan beberapa keuntungan yang jelas. Analisis prediktif memungkinkan identifikasi awal titik panas daya, termal, dan kemacetan, sehingga meminimalkan revisi desain yang mahal pada tahap selanjutnya. Perilaku pembelajaran adaptifnya meningkatkan ketahanan di berbagai beban kerja dan node teknologi, sementara optimasi otomatis dengan upaya manual yang terbatas mempercepat penutupan desain secara keseluruhan.

Seiring arsitektur SoC modern terus tumbuh dalam kompleksitas, heuristik klasik kesulitan untuk mengatasi parameter desain yang luas dan saling bergantung. Algoritma berbasis AI, yang dilatih pada data desain yang ekstensif, secara unik diposisikan untuk mengelola ruang desain multidimensi tersebut secara efisien. Dengan memanfaatkan pola yang dipelajari dan melakukan generalisasi di berbagai skenario, AI mencapai skalabilitas, presisi, dan efisiensi komputasi yang lebih besar daripada teknik berbasis heuristik konvensional. Gambar 2.4 membandingkan dua metodologi fundamental dalam desain VLSI. Alur kerja heuristik tradisional bergantung pada proses berbasis aturan dan pengawasan desainer yang signifikan; sedangkan kerangka kerja berbasis AI menggunakan pemodelan prediktif dan optimasi otomatis untuk secara cerdas memandu seluruh siklus hidup desain.




**Gambar 2.4** Perbandingan antara metodologi desain VLSI berbasis AI dan heuristik.

## 2.4 AI DALAM SINTESIS LOGIKA DAN OPTIMASI RTL

Dalam desain VLSI modern, sintesis logika dan optimasi RTL berfungsi sebagai tahapan penting untuk memenuhi tujuan PPA yang ketat. Seiring dengan meningkatnya kerumitan arsitektur SoC canggih, strategi desain berbasis heuristik dan pengalaman konvensional menghadapi keterbatasan dalam skalabilitas dan efisiensi. Integrasi AI memperkenalkan solusi cerdas, adaptif, dan berpusat pada data yang memungkinkan penyetelan parameter sintesis otomatis, eksplorasi ruang desain dinamis, dan optimasi sistematis di berbagai tujuan. Dengan memanfaatkan pemodelan prediktif, RL, dan eksplorasi yang dipandu optimasi, AI secara signifikan meningkatkan throughput dan kualitas desain. Bagian ini mengeksplorasi pendekatan berbasis AI untuk penyetelan parameter, kerangka kerja optimasi, dan implementasi dunia nyata yang menunjukkan peningkatan terukur dalam alur kerja sintesis RTL.

**AI Untuk Penyetelan dan Optimasi Parameter**

Metodologi berbasis AI meminimalkan intervensi manual dan meningkatkan hasil desain secara keseluruhan dengan menyetel parameter sintesis dan optimasi RTL secara otomatis. Implementasi utama meliputi:

- **Penentuan Ukuran Gerbang dan Penempatan Buffer**: Model AI memprediksi dimensi gerbang optimal dan titik penyisipan buffer untuk mengurangi penundaan propagasi dan daya switching dinamis di seluruh jalur sinyal kritis.
- **Adaptasi Tegangan dan Ambang Batas**: Algoritma ML secara dinamis mengatur tegangan suplai dan tingkat ambang batas transistor untuk mencapai efisiensi energi sambil mempertahankan penutupan waktu.

**Tabel 2.5** Aplikasi AI untuk penyetelan parameter dalam sintesis logika

| Target Optimasi | Pendekatan Tradisional | Pendekatan Berbasis AI | Manfaat |
|---|---|---|---|
| Penentuan Ukuran Gerbang (Gate Sizing) | Percobaan manual (*trial-and-error*) | Model regresi *Machine Learning* (ML) memprediksi ukuran gerbang yang optimal | Konvergensi lebih cepat dan kompromi PPA (Power, Performance, Area) yang lebih baik |
| Penyisipan Buffer (Buffer Insertion) | Aturan heuristik | Penyisipan buffer prediktif berbasis AI | Mengurangi pelanggaran waktu (*timing violations*) dan kebocoran daya (*leakage*) |
| Optimasi Tegangan/Ambang Batas (Voltage/Threshold) | Aturan tetap | Optimasi adaptif berbasis *Machine Learning* | Desain yang lebih hemat energi pada berbagai beban kerja |
| Eksplorasi Ruang Desain (Design Space Exploration) | Pencarian menyeluruh (*exhaustive search*) atau pencarian terpandu | *Reinforcement Learning* atau algoritma evolusioner | Menemukan solusi optimal secara global dengan lebih efisien |
| Penentuan Ukuran Gerbang (Gate Sizing) *(duplikasi dari tabel asli)* | Percobaan manual (*trial-and-error*) | Model regresi *Machine Learning* (ML) memprediksi ukuran gerbang yang optimal | Konvergensi lebih cepat dan kompromi PPA (Power, Performance, Area) yang lebih baik |

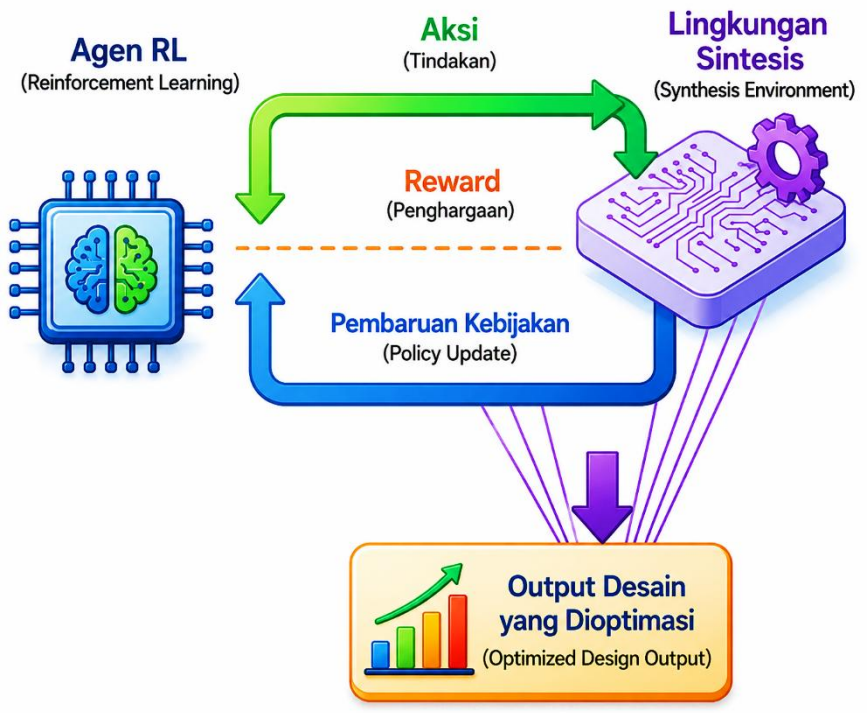
- **Eksplorasi Ruang Desain (DSE)**: Kerangka kerja eksplorasi berbasis AI menemukan solusi optimal Pareto yang secara simultan memenuhi banyak kendala PPA, menavigasi ruang parameter yang luas dan kompleks secara signifikan lebih efektif daripada pencarian menyeluruh.

Tabel 2.5 menunjukkan aplikasi AI untuk penyetelan parameter dalam sintesis logika, di mana pendekatan berbasis AI untuk penentuan ukuran gerbang, penyisipan buffer, penyesuaian tegangan/ambang batas, dan eksplorasi ruang desain mengungguli metode tradisional dengan

memungkinkan konvergensi yang lebih cepat, pengurangan pelanggaran, dan desain yang lebih hemat energi dan optimal secara global.

**Pembelajaran Penguatan untuk Transformasi Sintesis**

Dengan mempelajari urutan tindakan terbaik untuk meningkatkan kualitas desain, RL telah menjadi metode ampuh untuk mengotomatiskan transformasi sintesis. Agen RL terlibat dengan lingkungan simulasi yang mewakili desain dalam paradigma ini, di mana setiap tindakan dikaitkan dengan perubahan sintesis tertentu, termasuk pengaturan waktu ulang, penyisipan buffer, atau restrukturisasi logika. Efek daya, area, dan penundaan dari aktivitas ini dinilai oleh lingkungan dan dicatat oleh fungsi penghargaan yang dirancang dengan baik. Agen memperoleh pengetahuan tentang kebijakan yang secara bertahap meningkatkan efisiensi desain sambil menyeimbangkan pertukaran antara tujuan desain yang saling bertentangan selama beberapa episode pelatihan. RL selalu berevolusi, menemukan urutan transformasi baru yang mungkin terlewatkan oleh perancang manusia, berbeda dengan proses heuristik statis. Karena itu, RL sangat berguna untuk menskalakan SoC yang rumit dan menyelidiki jalur optimasi non-trivial yang menyimpang dari teknik berbasis aturan standar. Gambar 2.5 mengilustrasikan sistem loop tertutup dari agen RL yang diterapkan pada optimasi RTL dari desain VLSI.



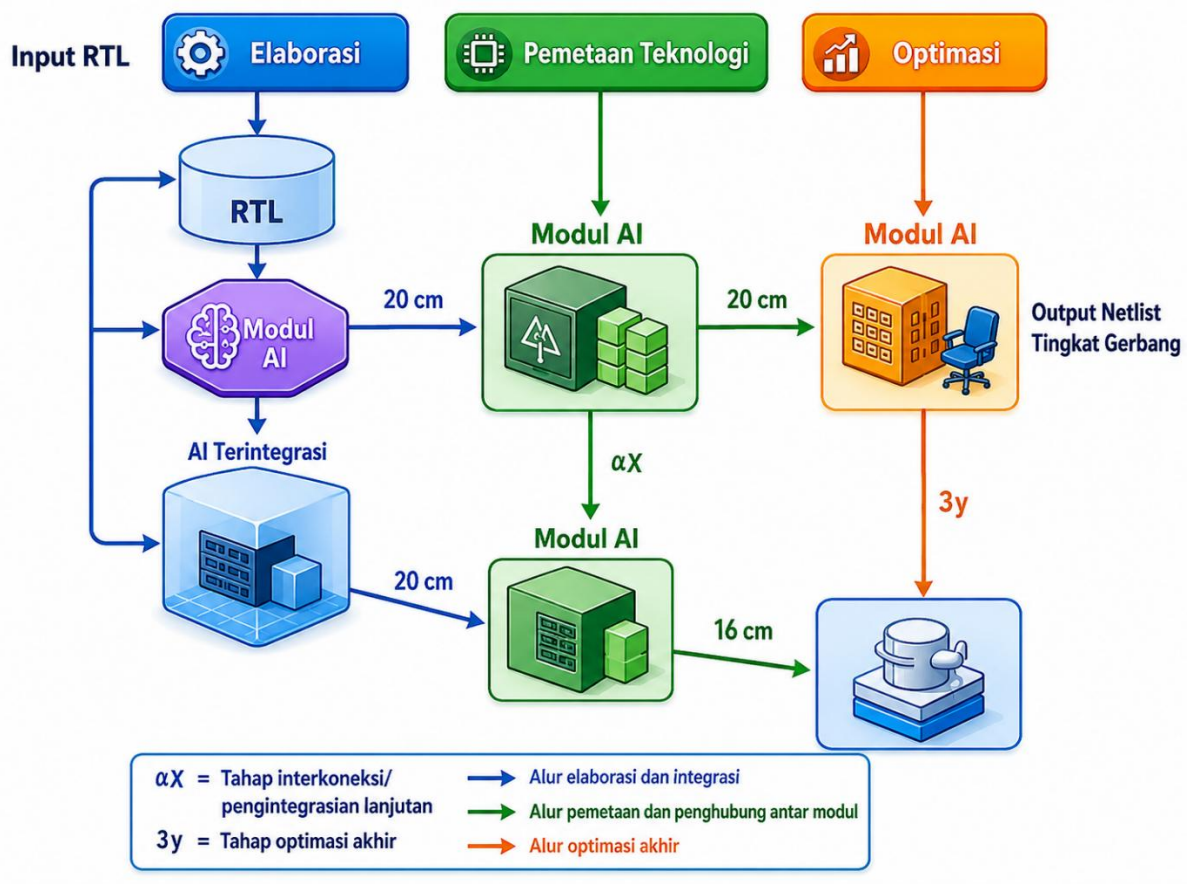

**Gambar 2.5** Alur kerja pembelajaran penguatan untuk optimasi RTL.

**Contoh Kasus dari Alat Industri**

Alur kerja desain VLSI tradisional dengan cepat diubah oleh penggunaan AI dalam optimasi RTL dan sintesis logika. Banyak inisiatif dari sektor akademis dan industri menunjukkan bagaimana metode AI meningkatkan produktivitas, menurunkan konsumsi daya, dan meningkatkan kualitas desain:

- Proyek Google AI Floorplanning mengoptimalkan penempatan dan konfigurasi SoC yang rumit dengan memanfaatkan RL. Proyek ini mengungguli teknik floorplanning tradisional dengan mencapai pengurangan yang signifikan dalam penggunaan daya dan kemacetan routing melalui eksplorasi cerdas dari pilihan penempatan.
- **Modul Pembelajaran Mesin dari Synopsys dan Cadence**: Penyedia EDA terkemuka semakin banyak memasukkan model ML ke dalam produk mereka untuk membantu prediksi daya, pengurangan kemacetan, dan ukuran gerbang. Dengan memungkinkan sintesis RTL yang lebih cepat dan lebih tepat, modul berbasis ML ini mengurangi kebutuhan akan iterasi coba-coba manusia.
- **Prototipe Akademik**: Pembahasan universitas telah menunjukkan bahwa dalam optimasi multi-objektif, metode sintesis berbasis RL secara rutin mengungguli metode berbasis heuristik konvensional. Prototipe ini menawarkan wawasan tentang teknik desain generasi berikutnya dengan mendemonstrasikan kemampuan AI untuk mengelola daya, luas area, dan penundaan sekaligus.

Kesimpulannya, alat-alat berbasis AI mengubah lingkungan desain VLSI baik di industri maupun akademisi dengan memungkinkan para desainer untuk menghasilkan perangkat berkinerja tinggi, hemat daya, dan efisien dalam hal luas area dengan kecepatan dan presisi yang belum pernah terjadi sebelumnya. Gambar 2.6 mengilustrasikan bagaimana AI diintegrasikan ke dalam alur sintesis RTL konvensional untuk meningkatkan hasil desain akhir. Modul AI ditempatkan secara strategis pada tahap-tahap kritis untuk mengotomatisasi dan mengoptimalkan keputusan yang secara tradisional dilakukan secara manual dan berdasarkan heuristik.




**Gambar 2.6** Contoh industri—alur sintesis yang ditingkatkan dengan AI.

## 2.5 APLIKASI AI DALAM DESAIN FISIK

Dalam desain fisik VLSI, penempatan, perutean, perencanaan tata letak, dan verifikasi memiliki dampak signifikan pada PPA (*Placement and Performance Automation*). Seiring meningkatnya kompleksitas chip, metodologi desain tradisional menghadapi tantangan seperti kepadatan interkoneksi, kemacetan perutean, dan inefisiensi termal. Teknik AI—terutama CNN dan GNN—memungkinkan optimasi adaptif berbasis data yang mengotomatiskan keputusan desain untuk meningkatkan efisiensi dan kinerja. Bagian ini mengeksplorasi bagaimana model-model ini meningkatkan regulasi termal, integritas sinyal, perencanaan tata letak, estimasi kemacetan, deteksi titik panas daya, dan optimasi penempatan dan perutean secara keseluruhan.

**CNN dan GNN untuk Penempatan dan Perutean**

CNN dan GNN sangat efektif untuk pemodelan prediktif dalam desain fisik, yang secara inheren menunjukkan karakteristik spasial dan relasional. CNN memproses grid penempatan sebagai peta fitur spasial, menangkap korelasi antara wilayah tata letak lokal dan global untuk memperkirakan kepadatan perutean, distribusi penundaan, dan risiko kemacetan. Sebaliknya, GNN merepresentasikan netlist sirkuit sebagai grafik—di mana node sesuai dengan elemen logika dan edge menunjukkan interkoneksi—memungkinkan prediksi masalah integritas sinyal, pelanggaran waktu, dan kemacetan struktural berdasarkan ketergantungan topologi.

Bersama-sama, kerangka kerja pembelajaran ini memungkinkan identifikasi dan mitigasi proaktif terhadap tantangan penempatan dan perutean jauh sebelum finalisasi desain.

**Perencanaan Tata Letak dan Prediksi Kemacetan Berbasis AI**

AI meningkatkan perencanaan tata letak dengan memungkinkan penempatan blok fungsional dan interkoneksi yang didorong oleh data dan antisipatif. Algoritma ML dapat memprediksi potensi wilayah kemacetan sebelum tahap perutean, sedangkan kerangka kerja RL secara iteratif menentukan strategi penempatan optimal yang meminimalkan kompleksitas pengkabelan dan panjang interkoneksi. Pandangan ke depan seperti itu membantu perancang dalam mengatasi masalah waktu dan kemampuan perutean sejak dini dalam proses. Dengan mengintegrasikan pemodelan prediktif dengan optimasi multi-objektif untuk efisiensi daya dan pemanfaatan kabel, AI mendorong perencanaan tata letak yang seimbang dan berorientasi pada kinerja yang secara signifikan menyederhanakan tahapan desain selanjutnya.

**Peramalan Titik Panas Daya**

Pembentukan titik panas daya secara signifikan memengaruhi stabilitas termal, keandalan, dan kinerja sirkuit terpadu. Kerangka kerja prediktif yang dibantu AI mengevaluasi tata letak desain pada tahap awal untuk menemukan wilayah yang rentan terhadap disipasi daya dinamis atau kebocoran yang tinggi. Dengan menggabungkan wawasan ini selama penempatan dan perutean, perancang dapat secara strategis memodifikasi atau mengubah ukuran modul untuk mencapai distribusi termal yang lebih baik dan meminimalkan pemanasan lokal. Dalam arsitektur SoC yang padat, manajemen termal prediktif tersebut menghasilkan peningkatan efisiensi energi, pengurangan risiko panas berlebih, dan peningkatan keandalan operasional.

**Dampak pada Manajemen Termal dan Integritas Sinyal**

Kemajuan yang didorong oleh AI secara signifikan meningkatkan integritas sinyal dan pengaturan termal dalam desain VLSI modern. Algoritma prediktif menunjukkan zona daya tinggi, memungkinkan penyeimbangan beban kerja dan penyesuaian struktural yang mempertahankan keseragaman suhu dan mencegah kehilangan kinerja akibat tekanan termal. Demikian pula, sistem AI mendeteksi potensi masalah seperti interferensi silang, penurunan tegangan, dan penyimpangan waktu, sehingga memfasilitasi tindakan perbaikan atau perlindungan yang tepat waktu untuk menjaga keandalan listrik. Melalui analisis proaktif tersebut, AI memastikan kinerja yang konsisten dan tangguh dalam batasan daya dan termal yang ketat.

Dengan menyematkan kecerdasan prediktif ke dalam tahap perencanaan tata letak, penempatan, dan perutean, AI secara fundamental mengubah alur kerja desain fisik konvensional. Perancang dapat mengantisipasi dan mengurangi kemacetan, titik panas termal, dan komplikasi terkait integritas, sehingga mencapai metrik daya-kinerja-luas (PPA) yang unggul. Pendekatan ini memastikan bahwa SoC canggih dengan kepadatan tinggi dan arsitektur edge beroperasi secara efisien, andal, dan dalam anggaran termal dan daya yang ditentukan.

## 2.6 ESTIMASI DAN PEMODELAN DAYA YANG DITINGKATKAN AI

Dalam sistem VLSI modern, estimasi daya memainkan peran penting dalam mencapai desain chip yang hemat energi dan berkinerja tinggi, khususnya pada SoC yang kompak dan hemat daya. Teknik estimasi tradisional sangat bergantung pada simulasi yang menyeluruh yang, meskipun akurat, membutuhkan upaya komputasi dan waktu desain yang ekstensif. Integrasi AI memperkenalkan pendekatan prediktif berbasis data yang memungkinkan analisis daya yang lebih cepat dan lebih tepat, memungkinkan perancang untuk membuat pilihan yang tepat sejak awal siklus pengembangan. Kerangka estimasi cerdas ini meningkatkan skalabilitas dan selaras dengan metodologi desain yang ada, mengubah praktik optimasi daya konvensional menjadi proses yang lebih adaptif dan efisien.

**Metode Estimasi Tradisional Vs Model AI**

Estimasi tingkat RTL, yang menggunakan model analitik untuk memperkirakan daya dinamis dan kebocoran, dan simulasi SPICE, yang menawarkan wawasan komprehensif tingkat transistor tetapi sangat lambat untuk desain besar, adalah contoh metodologi estimasi daya konvensional. Teknik berbasis heuristik sering digunakan oleh alat EDA untuk mempercepat proses, meskipun akurasi dikorbankan dalam proses tersebut. Sistem skala besar, arsitektur yang sangat rumit, dan siklus desain iteratif semuanya sulit ditangani oleh pendekatan konvensional ini. Dengan menggunakan pendekatan regresi atau ML untuk belajar dari data desain historis, model berbasis AI melampaui batasan ini dan memberikan proyeksi daya tahap awal tanpa perlu simulasi lengkap. Model-model ini memungkinkan para perancang untuk membuat keputusan yang tepat dan sadar akan daya pada tahap yang jauh lebih awal dalam proses desain karena model-model ini dapat digeneralisasikan di banyak iterasi desain dan node teknologi.

**Model Regresi untuk Prediksi Daya Awal**

Untuk memprediksi daya dinamis, kebocoran, dan hubung singkat, model regresi sering digunakan. Regresi polinomial menangkap korelasi nonlinier antara faktor desain dan daya, sedangkan regresi linier sederhana menawarkan perkiraan yang cepat dan mudah untuk eksplorasi desain awal. Regresi berbasis jaringan saraf memberikan akurasi yang lebih tinggi untuk data berdimensi tinggi dan rumit, memungkinkan perilaku desain yang kompleks. Untuk kinerja daya yang optimal, model-model ini sangat berguna dalam mengidentifikasi blok-blok kritis daya, mengarahkan eksplorasi ruang desain tahap awal, dan memengaruhi desain fisik dan sintesis logika.

**Manfaat Kecepatan, Akurasi, dan Skalabilitas**

Dibandingkan dengan teknik konvensional, perkiraan daya yang ditingkatkan AI memiliki beberapa keunggulan. Sementara model yang terlatih dengan baik mempertahankan akurasi dalam beberapa poin persentase dari simulasi terperinci, prediksi dihasilkan jauh lebih cepat daripada simulasi SPICE. Estimasi daya berbasis AI sangat cocok untuk arsitektur skala besar kontemporer karena model ini efektif untuk SoC yang sangat besar dengan jutaan gerbang dan interkoneksi.

**Integrasi dengan Alur Desain**

Beberapa fase siklus desain VLSI dapat dengan mudah menggabungkan estimasi daya berbasis AI. Estimasi ini menggunakan perkiraan daya awal untuk mengarahkan sintesis logika pada tingkat RTL. Perkiraan titik panas membantu memandu penempatan dan pilihan perutean selama desain fisik untuk mengurangi penggunaan daya. Perancang dapat membuat keputusan yang hemat daya tanpa menjalankan simulasi yang panjang berkat model regresi yang disertakan dalam alat EDA. Akurasi model ditingkatkan lebih lanjut untuk desain selanjutnya melalui umpan balik berkelanjutan dari pengukuran pasca-silikon, menciptakan lingkaran pembelajaran yang secara bertahap meningkatkan keandalan yang diprediksi.

Estimasi daya berbasis AI mengubah alur kerja VLSI konvensional dengan menawarkan proyeksi yang cepat, tepat, dan terukur. Prediksi daya berbasis regresi memungkinkan para perancang untuk mempercepat pengembangan SoC yang hemat energi dengan mengurangi beban simulasi, meningkatkan hasil daya-kinerja-luas (PPA), dan mendukung keputusan desain yang memperhatikan daya pada tahap awal yang mempersingkat waktu pemasaran secara keseluruhan.

## 2.7 PENCARIAN ARSITEKTUR NEURAL YANG BERORIENTASI PERANGKAT KERAS (NAS)

NAS merupakan kerangka kerja berbasis AI canggih yang mengotomatiskan desain jaringan neural sekaligus mengoptimalkannya untuk kendala perangkat keras tertentu. Pendekatan ini sangat penting untuk platform edge-AI dan ASIC berdaya sangat rendah, di mana efisiensi, latensi, dan kapasitas komputasi yang terbatas membatasi desain yang ketat. Dengan memungkinkan optimasi bersama model neural dan perangkat keras yang mendasarinya, NAS yang berorientasi perangkat keras memastikan bahwa arsitektur tetap efisien secara komputasi dan dapat diterapkan secara praktis pada sistem dengan keterbatasan sumber daya. Bagian ini mengeksplorasi metodologi NAS utama, prinsip desain bersama perangkat keras-algoritma, dan implementasi dunia nyata yang menunjukkan bagaimana model AI hemat energi dapat dicapai melalui optimasi arsitektur sistematis.

**Gambaran Umum Teknik NAS**

NAS mengeksplorasi berbagai strategi cerdas untuk secara otomatis menemukan struktur jaringan neural yang efektif. Dalam pendekatan berbasis RL, agen otonom menghasilkan dan mengevaluasi model kandidat, secara bertahap menyempurnakan kebijakannya untuk mencapai desain optimal. Algoritma evolusioner beroperasi melalui generasi-generasi berturut-turut dengan memilih, memutasi, dan menggabungkan kembali arsitektur berkualitas tinggi, memastikan keragaman dalam kumpulan desain. Teknik NAS berorientasi gradien, seperti DARTS, menerapkan optimasi diferensiabel untuk menyesuaikan parameter jaringan secara efisien, mencapai konvergensi cepat dengan upaya komputasi yang berkurang.

Ketika teknik-teknik ini digunakan bersama, mereka memungkinkan eksplorasi sistematis dari kemungkinan arsitektur, memungkinkan pembuatan jaringan saraf yang selaras dengan tujuan algoritmik dan kemampuan perangkat keras. Tabel 2.6 menyajikan perbandingan singkat kategori NAS utama, menunjukkan bahwa RL mencapai hasil yang adaptif dan berkinerja tinggi tetapi membutuhkan komputasi yang tinggi, algoritma

evolusioner memastikan keragaman model tetapi berevolusi lambat, dan NAS berbasis gradien menawarkan efisiensi dengan rentang eksplorasi yang lebih sempit.

**Tabel 2.6** Teknik dan karakteristik utama NAS

| Teknik NAS (Neural Architecture Search) | Strategi Pencarian | Kelebihan | Kekurangan |
|---|---|---|---|
| Reinforcement Learning (Pembelajaran Penguatan) | Eksplorasi berbasis agen | Menghasilkan arsitektur berkualitas tinggi dan sangat fleksibel | Membutuhkan biaya komputasi yang tinggi |
| Algoritma Evolusioner | Evolusi populasi | Pencarian yang tangguh (*robust*) dengan keragaman solusi yang tinggi | Konvergensi relatif lambat |
| NAS Berbasis Gradien (Gradient-based NAS) | Optimasi yang dapat diturunkan (*Differentiable Optimization*) | Cepat dan efisien | Ruang pencarian (*search space*) terbatas |
| Reinforcement Learning (Pembelajaran Penguatan) *(duplikasi dari tabel asli)* | Eksplorasi berbasis agen | Menghasilkan arsitektur berkualitas tinggi dan sangat fleksibel | Membutuhkan biaya komputasi yang tinggi |
| Algoritma Evolusioner *(duplikasi dari tabel asli)* | Evolusi populasi | Pencarian yang tangguh (*robust*) dengan keragaman solusi yang tinggi | Konvergensi relatif lambat |

**Desain Bersama Algoritma dan Perangkat Keras**

Untuk memastikan bahwa model AI beroperasi secara efisien pada platform penyebaran tertentu, NAS yang sadar perangkat keras menekankan optimasi bersama struktur jaringan dan kendala perangkat keras. Dengan mengintegrasikan pertimbangan latensi dan sumber daya secara langsung ke dalam proses pencarian, NAS mencapai keseimbangan antara kecepatan inferensi, konsumsi energi, dan pemanfaatan memori. Pendekatan desain bersama yang terkoordinasi ini tidak hanya memungkinkan pemrosesan waktu nyata tetapi juga meningkatkan efisiensi tingkat sistem di seluruh lingkungan komputasi heterogen. Manfaat utamanya meliputi:

- **Efisiensi Energi:** Menurunkan penggunaan daya pada ASIC berdaya sangat rendah dan perangkat edge.
- **Jejak Memori:** Mengurangi jumlah bandwidth dan penyimpanan yang dibutuhkan.
- **Throughput:** Mengoptimalkan paralelisme akselerator perangkat keras untuk pemrosesan yang lebih cepat.

Dengan mengoptimalkan jaringan bersamaan dengan perangkat keras, metode desain bersama ini menjamin kinerja yang baik bahkan di bawah keterbatasan sumber daya yang parah.

**Aplikasi dalam Edge-AI dan ASIC Berdaya Sangat Rendah**

NAS yang sadar perangkat keras memainkan peran transformatif dalam memungkinkan penerapan AI yang efisien di berbagai platform yang dibatasi oleh daya dan kapasitas komputasi yang terbatas. Aplikasi yang paling menonjol terlihat di lingkungan yang membutuhkan kecerdasan yang hemat energi dan waktu respons minimal. Area implementasi utama meliputi:

- Perangkat Edge: Digunakan dalam ponsel pintar, perangkat wearable, dan sensor berbasis IoT yang membutuhkan latensi sangat rendah dan pengoperasian baterai yang lama.
- ASIC Berdaya Sangat Rendah: Akselerator AI khusus yang dirancang untuk sistem otonom, instrumentasi perawatan kesehatan, dan aplikasi IoT industri.
- Arsitektur AI yang Dioptimalkan: NAS merancang model neural yang mempertahankan akurasi prediksi tinggi sambil memenuhi batasan ketat pada energi, jejak memori, dan penundaan inferensi.

**Contoh Kasus: NAS dalam Model AI Hemat Energi**

Nilai praktis dari NAS yang sadar perangkat keras jelas tercermin dalam kemampuannya untuk menghasilkan model AI yang dioptimalkan energi untuk perangkat dengan sumber daya terbatas. Beberapa contoh dunia nyata menyoroti kemampuan ini:

- Edge TPU Google dengan NASNet menunjukkan bahwa arsitektur neural spesifik perangkat keras dapat mencapai akurasi prediksi yang tinggi sekaligus secara signifikan mengurangi latensi komputasi dan permintaan energi secara keseluruhan.
- Pembahasan tentang jaringan konvolusional kompak untuk elektronik yang dapat dikenakan lebih lanjut menunjukkan bahwa optimasi berbasis NAS dapat memenuhi batasan desain yang ketat dari ASIC daya ultra-rendah tanpa mengorbankan kinerja analitis.
- Studi perbandingan secara konsisten melaporkan bahwa arsitektur yang dihasilkan NAS memberikan efisiensi energi yang unggul, inferensi yang lebih cepat, dan pengurangan penggunaan memori dibandingkan dengan desain yang direkayasa secara manual, menempatkannya sebagai solusi ideal untuk komputasi tepi dan platform AI tertanam.

## 2.8 PENINGKATAN ALAT EDA MENGGUNAKAN AI

Alat EDA membentuk dasar pengembangan VLSI saat ini, mendukung tugas-tugas seperti tata letak sirkuit, verifikasi, dan analisis waktu. Namun, seiring arsitektur chip menjadi lebih kompleks dan persyaratan daya-kinerja-luas (PPA) semakin ketat, sistem EDA tradisional sering kali menghadapi keterbatasan dalam skalabilitas dan efisiensi komputasi. Integrasi AI ke dalam alur kerja EDA telah mengatasi tantangan ini dengan memperkenalkan otomatisasi, pemodelan prediktif, dan optimasi adaptif. Melalui kemampuan ini, lingkungan EDA yang ditingkatkan AI memungkinkan sintesis tata letak yang lebih cepat, konvergensi waktu yang

lebih baik, dan deteksi anomali desain yang cerdas—memungkinkan para insinyur untuk mencapai target kinerja yang lebih tinggi dengan upaya desain yang lebih sedikit.

**Otomatisasi dalam Tata Letak, Verifikasi, dan Analisis**

AI mentransformasi EDA dengan mengotomatiskan beberapa proses desain inti. Dalam desain tata letak, sistem AI memprediksi konfigurasi perutean dan penempatan yang efisien, secara signifikan mengurangi ketergantungan pada penyesuaian manual. Selama verifikasi dan Pemeriksaan Aturan Desain (DRC), model ML meningkatkan akurasi dengan mengidentifikasi wilayah yang rentan terhadap kesalahan dan menandai potensi pelanggaran aturan sejak dini. Demikian pula, teknik AI prediktif mempercepat Analisis Waktu Statis (STA) dengan cepat mengenali inkonsistensi waktu yang mungkin menunda penutupan. Melalui otomatisasi ini, AI tidak hanya meminimalkan intervensi manusia tetapi juga memperkuat presisi dan produktivitas desain secara keseluruhan.

**Penutupan Waktu Cerdas dan Optimasi Hasil Produksi**

Mencapai penutupan waktu sangat penting untuk memenuhi target kinerja dan memastikan integritas fungsional desain VLSI. AI meningkatkan proses ini melalui analisis waktu prediktif, yang mengantisipasi potensi pelanggaran dan mengidentifikasi jalur sinyal kritis. Dengan menggunakan strategi optimasi otomatis seperti pengubahan ukuran gerbang, penyisipan buffer, dan restrukturisasi jalur, AI dapat mengusulkan tindakan korektif sebelumnya. Selain itu, model cerdas dapat memprediksi variasi selama fabrikasi dan menyempurnakan pola desain untuk meningkatkan hasil produksi, memastikan bahwa chip akhir berkinerja tinggi dan dapat diproduksi dalam skala besar.

**Deteksi Anomali dalam Validasi Desain**

AI memperkuat validasi desain dengan mengungkap ketidakberaturan yang mungkin gagal dideteksi oleh teknik konvensional. Melalui pengenalan pola tingkat lanjut, model ML mengidentifikasi perilaku yang tidak biasa dalam struktur logika, jalur perutean, dan konfigurasi tata letak. Deteksi outlier lebih lanjut menyoroti variasi yang tidak terduga dalam pengaturan waktu, daya, atau integritas sinyal. Dengan merekomendasikan tindakan korektif sebelum fabrikasi, AI meminimalkan kemungkinan kesalahan desain dan meningkatkan keandalan sistem secara keseluruhan.

**Peningkatan Produktivitas dan Pengurangan Waktu Pemasaran**

Integrasi AI ke dalam alat EDA secara signifikan mempercepat waktu pemasaran dan meningkatkan produktivitas secara keseluruhan. Manfaat utamanya meliputi:

- **Iterasi Lebih Cepat:** Optimasi otomatis mempersingkat siklus desain manual dan mempercepat siklus umpan balik.
- **Pengurangan Kesalahan:** Deteksi dini pelanggaran aturan desain dan anomali meminimalkan pengerjaan ulang dan meningkatkan akurasi.
- **Skalabilitas:** Sistem AI dapat secara efisien menangani SoC skala besar dan arsitektur jutaan gerbang.
- **Optimalisasi Sumber Daya**: Dengan berkurangnya upaya komputasi dan manual, para perancang dapat lebih fokus pada inovasi dan kreativitas sistem.

Dengan menggabungkan otomatisasi dengan kecerdasan prediktif dan pengambilan keputusan berbasis data, alat EDA yang didukung AI mendefinisikan ulang alur kerja desain modern. Dalam industri semikonduktor yang kompetitif, kemajuan ini memungkinkan proses desain yang lebih cepat, lebih andal, dan sangat terukur—penting untuk mewujudkan SoC generasi berikutnya yang hemat daya dan berkinerja tinggi.

## 2.9 TANTANGAN IMPLEMENTASI AI DALAM DESAIN VLSI

Meskipun AI telah membawa kemajuan luar biasa pada desain VLSI modern, integrasinya disertai dengan serangkaian tantangan yang harus ditangani dengan cermat untuk memastikan keandalan, skalabilitas, dan keberlanjutan. Salah satu masalah yang paling mendesak melibatkan interpretasi model AI. Arsitektur DL sering berfungsi sebagai "kotak hitam," sehingga sulit bagi para insinyur untuk memahami dasar keputusan otomatis mereka. Kurangnya transparansi ini dapat mengurangi kepercayaan pada alur kerja yang dibantu AI, terutama ketika hasil optimasi bertentangan dengan batasan desain atau menyebabkan trade-off yang tidak terduga.

Ketersediaan dan generalisasi data semakin menambah kompleksitas. Melatih model AI yang akurat membutuhkan kumpulan data berlabel yang ekstensif dan berkualitas tinggi; namun, data tersebut dari desain masa lalu biasanya terbatas atau tidak dapat diakses. Selain itu, model yang dilatih pada satu node teknologi mungkin tidak berkinerja efektif ketika diterapkan pada node lain, sehingga memerlukan pembuatan kumpulan data tambahan atau sintetis untuk menjembatani kesenjangan ini. Tanpa manajemen yang cermat, keterbatasan tersebut dapat melemahkan kinerja prediksi model dan keandalan secara keseluruhan.

Kekhawatiran lain terletak pada overfitting dan kerentanan terhadap serangan. Model yang overfitting mungkin berkinerja baik selama pelatihan tetapi gagal untuk digeneralisasi ke skenario desain baru, sementara input serangan—baik yang tidak disengaja maupun disengaja—dapat mendistorsi prediksi AI, menimbulkan risiko terhadap keselamatan dan fungsionalitas. Lebih lanjut, pelatihan AI itu sendiri mengkonsumsi energi komputasi yang substansial, yang kontras dengan tujuan daya rendah dari desain VLSI. Jika metode pelatihan hemat energi atau akselerator perangkat keras tidak diadopsi, biaya komputasi pelatihan mungkin melebihi manfaat optimasi berbasis AI.

Pertimbangan privasi dan etika juga menuntut perhatian. Penggunaan data desain yang bersifat rahasia atau sensitif memerlukan perlindungan ketat terhadap kekayaan intelektual, dan akuntabilitas harus dijaga melalui pengambilan keputusan AI yang transparan. Selain itu, keberlanjutan harus dimasukkan ke dalam alur pengembangan AI, dengan menekankan komputasi yang bertanggung jawab terhadap lingkungan. Terlepas dari tantangan-tantangan ini, AI terus mendefinisikan kembali ruang lingkup otomatisasi dan efisiensi dalam sistem VLSI. Mengatasi masalah interpretasi, kelangkaan data, ketahanan, penggunaan energi, dan tanggung jawab etika sangat penting. Melalui kerangka kerja AI yang dapat dijelaskan, pelatihan model yang tangguh, dan praktik komputasi yang berkelanjutan, generasi berikutnya dari desain VLSI berbasis AI dapat dibuat aman, andal, dan sadar lingkungan.

## 2.10 STUDI KASUS DAN IMPLEMENTASI DUNIA NYATA

Peran transformatif AI dalam desain VLSI hemat daya menjadi jelas melalui berbagai implementasi dunia nyata yang diambil dari pembahasan akademis dan praktik industri. Studi-studi ini menggambarkan bagaimana metodologi berbasis AI meningkatkan alur kerja EDA, menyempurnakan strategi floorplanning, dan meningkatkan akurasi estimasi daya bocor. Hasilnya adalah kemajuan terukur dalam indikator kinerja utama seperti efisiensi daya, area desain, penundaan, dan produktivitas keseluruhan. Jika dibandingkan dengan teknik desain konvensional, pendekatan yang dibantu AI ini menyoroti tidak hanya keunggulan teknisnya tetapi juga kelayakan praktisnya, menggarisbawahi keuntungan nyata dari pengintegrasian algoritma cerdas ke dalam proses desain VLSI modern.

**Pembelajaran Penguatan Google untuk Floorplanning**

Google mempelopori penggunaan RL dalam floorplanning chip untuk mencapai penempatan blok yang optimal, penggunaan daya yang lebih rendah, panjang interkoneksi yang lebih pendek, dan pengurangan kemacetan routing. Dalam kerangka kerja ini, agen RL berinteraksi secara dinamis dengan lingkungan tata letak simulasi, menerima umpan balik berdasarkan kualitas desain dan secara progresif menyempurnakan strategi penempatannya. Hasilnya menunjukkan peningkatan signifikan dalam penutupan waktu, efisiensi daya, dan manajemen kemacetan jika dibandingkan dengan tata letak yang dirancang secara manual. Pendekatan ini menyoroti kemampuan RL untuk melampaui metodologi konvensional yang digerakkan manusia dengan secara otonom menavigasi dan mengoptimalkan ruang desain kompleks yang melekat pada sistem VLSI modern.

**AI untuk Prediksi Daya Kebocoran di Lingkungan Akademis**

Dalam pembahasan akademis, model ML semakin banyak digunakan untuk memprediksi daya kebocoran pada node teknologi submikron dan nanoskala. Dengan memanfaatkan catatan desain historis dan parameter terkait proses, para sarjana telah menggunakan teknik regresi, arsitektur neural, dan metode pembelajaran ensemble untuk menghasilkan perkiraan kebocoran tahap awal yang akurat. Pendekatan ini secara signifikan mengurangi ketergantungan pada simulasi SPICE yang intensif secara komputasi dan memberdayakan perancang untuk membuat modifikasi desain yang terinformasi yang mengurangi kebocoran, meningkatkan pemanfaatan energi, dan mempercepat keseluruhan jadwal pengembangan. Investigasi ini menggarisbawahi bagaimana AI dapat mengubah analisis kompleks berbasis data menjadi alur kerja desain yang terstruktur, mudah dikelola, dan sangat mudah diprediksi.

**Modul ML dalam Perangkat Lunak EDA Komersial**

Perusahaan EDA terkemuka seperti Mentor Graphics, Cadence, dan Synopsys telah mulai mengintegrasikan modul AI dan ML untuk meningkatkan kerangka kerja desain komersial mereka. Modul cerdas ini memungkinkan estimasi daya yang akurat, identifikasi dini titik panas termal dan kemacetan, optimasi penutupan waktu, dan analisis prediktif tantangan perutean. Bahkan dalam arsitektur SoC skala besar dengan jutaan gerbang, adopsi AI secara signifikan mempercepat siklus desain, meningkatkan tingkat keberhasilan percobaan pertama,

dan meningkatkan produktivitas secara keseluruhan. Dengan mengotomatiskan tugas analitis yang berulang dan memandu keputusan desain penting melalui wawasan berbasis data, AI telah menjadi elemen dasar untuk mencapai skalabilitas dan efisiensi dalam lingkungan desain VLSI modern.

**Analisis Komparatif dan Metrik Kinerja**

Analisis komparatif secara konsisten menunjukkan bahwa metodologi berbasis AI mengungguli pendekatan desain tradisional di berbagai parameter evaluasi. Model prediktif berbasis RL dan pembelajaran mesin berkontribusi pada siklus pengembangan yang lebih pendek, pengurangan disipasi daya, peningkatan pemanfaatan area, dan peningkatan akurasi waktu. Sistem cerdas ini memungkinkan perancang untuk secara efisien mengelola kompleksitas arsitektur SoC modern yang terus berkembang tanpa peningkatan upaya manual yang proporsional, sehingga meningkatkan hasil produksi tahap pertama. Secara kolektif, temuan tersebut menetapkan AI sebagai penggerak transformatif dalam desain VLSI saat ini, memberikan keuntungan nyata dalam presisi, efisiensi, dan pemanfaatan sumber daya.

## 2.11 TREN DAN ARAH PEMBAHASAN MASA DEPAN

Integrasi AI ke dalam desain VLSI berkembang pesat, menandai transformasi penting dalam pengembangan semikonduktor. Seiring dengan terus menyusutnya ukuran node teknologi dan meningkatnya kompleksitas desain, metodologi berbasis AI yang muncul diharapkan dapat mendefinisikan ulang bagaimana prosesor berdaya rendah, berkinerja tinggi, dan aman direkayasa. Ke depan, fokus akan bergeser ke arah pencapaian adaptabilitas yang lebih besar, integrasi AI yang mulus, peningkatan efisiensi energi, dan mekanisme keamanan tertanam untuk memastikan ketahanan dan keberlanjutan arsitektur VLSI generasi berikutnya.

**Model Adaptif Mandiri dan Pembelajaran Berkelanjutan**

Arah masa depan yang signifikan dalam desain VLSI melibatkan adopsi model adaptif mandiri dan pembelajaran berkelanjutan. Tidak seperti pendekatan konvensional yang memerlukan pelatihan ulang lengkap ketika kendala desain atau pola data berkembang, model ini secara dinamis menyesuaikan diri sepanjang proses desain. Kemampuan ini menjadi sangat berharga pada node teknologi di bawah 10 nm, di mana variasi proses sering terjadi dan sulit diprediksi. Dengan belajar secara bertahap dari data waktu nyata, sistem tersebut dapat secara iteratif menyempurnakan parameter desain, yang mengarah pada konvergensi yang lebih cepat dan hasil kinerja yang lebih konsisten. Selain itu, ketergantungan yang lebih rendah pada kumpulan data statis memungkinkan generalisasi yang lebih luas di berbagai metodologi desain dan proses fabrikasi. Kemampuan adaptasi ini memungkinkan alur kerja VLSI untuk berkembang seiring dengan inovasi semikonduktor yang cepat tanpa menimbulkan biaya komputasi yang besar terkait dengan pelatihan ulang model skala besar.

**AI untuk IC 3D dan Integrasi Heterogen**

Seiring dengan kemajuan fabrikasi chip menuju sirkuit terpadu 3D (IC 3D), di mana beberapa lapisan silikon ditumpuk secara vertikal untuk mencapai kepadatan perangkat yang lebih tinggi, para perancang menghadapi tantangan baru terkait dengan manajemen termal

dan penundaan interkoneksi. AI menyediakan strategi efektif untuk mengurangi masalah ini. Model prediksi termal berbasis pembelajaran mesin dapat mengidentifikasi potensi titik panas di awal fase desain dan menyarankan tindakan mitigasi yang ditargetkan, memastikan keandalan operasional jangka panjang. Selain itu, optimasi berbasis AI ditingkatkan melalui perencanaan via silikon (TSV) dan penempatan blok vertikal, mengurangi kemacetan dan latensi sinyal dalam arsitektur bertumpuk.

Di luar IC 3D, integrasi heterogen—menggabungkan prosesor, unit memori, FPGA, dan ASIC dalam satu paket—menghadirkan potensi besar untuk peningkatan kinerja. Dalam konteks ini, AI memfasilitasi desain bersama yang cerdas dengan mengelola aliran data, alokasi sumber daya, dan komunikasi antar komponen secara dinamis. Kontrol adaptif berbasis pembelajaran penguatan lebih lanjut memastikan bahwa sumber daya sistem dimanfaatkan secara efisien sambil meminimalkan hambatan kinerja. Secara kolektif, strategi desain yang dibantu AI ini mengatasi tantangan utama dalam integrasi, perutean, dan pembuangan panas, menjadikan sistem chip yang kompleks, berdensitas tinggi, dan heterogen menjadi layak dan andal.

**Kerangka Kerja AI yang Hemat Energi**

Arah baru yang penting dalam desain VLSI adalah pengembangan kerangka kerja AI hemat energi yang mengintegrasikan keberlanjutan ke dalam proses desain itu sendiri. Dengan meningkatnya ketergantungan pada AI untuk optimasi chip, jejak energi komputasi AI telah menjadi perhatian yang semakin besar. NAS yang sadar perangkat keras menyediakan pendekatan sistematis untuk membangun model neural yang beroperasi secara efisien pada silikon sambil menghormati batasan daya perangkat keras. Ketika dilengkapi dengan metode kompresi seperti pemangkasan, kuantisasi, dan aritmatika presisi bit yang dikurangi, arsitektur ini secara signifikan mengurangi konsumsi daya sambil mempertahankan akurasi prediksi yang tinggi.

Lebih lanjut, strategi penyebaran yang berpusat pada edge berkontribusi untuk menurunkan permintaan energi secara keseluruhan dengan mengeksekusi komputasi di dekat sumber data, sehingga mengurangi kebutuhan transfer data yang ekstensif ke server terpusat. Keseimbangan antara kompleksitas algoritma dan efisiensi perangkat keras ini tidak hanya memajukan tujuan desain daya rendah tetapi juga selaras dengan tujuan keberlanjutan yang lebih luas dalam rekayasa semikonduktor. Kemajuan di masa depan diharapkan akan menekankan alur kerja yang digerakkan oleh AI yang secara inheren sadar energi, memastikan bahwa inovasi dalam desain VLSI tetap berkinerja tinggi dan bertanggung jawab terhadap lingkungan.

**AI untuk Desain VLSI yang Sadar Keamanan**

Seiring arsitektur VLSI menjadi semakin kompleks dan saling terhubung, arsitektur tersebut menghadapi peningkatan paparan terhadap berbagai ancaman keamanan, seperti penyisipan Trojan perangkat keras, eksploitasi saluran samping, dan pelanggaran kekayaan intelektual. Tindakan pencegahan tradisional—seperti verifikasi manual atau inspeksi pasca-fabrikasi—tidak lagi cukup untuk melindungi dari kerentanan yang canggih tersebut. AI memperkenalkan pergeseran paradigma dengan menanamkan kecerdasan keamanan

langsung ke dalam alur kerja desain, memungkinkan mitigasi ancaman proaktif daripada pertahanan reaktif.

Algoritma ML dapat menganalisis pola aktivitas sirkuit selama tahap desain awal untuk mendeteksi potensi anomali yang mengindikasikan Trojan perangkat keras sebelum fabrikasi. Demikian pula, model AI prediktif dapat menilai kemungkinan kerentanan saluran samping dan memandu penerapan strategi desain protektif jauh sebelum penyebaran. Lebih lanjut, sistem verifikasi yang dibantu AI membantu menjaga kerahasiaan data dan memastikan integritas kekayaan intelektual dengan terus memantau modifikasi atau kebocoran yang tidak sah. Melalui integrasi yang mulus dari mekanisme cerdas ini, AI memperkuat kepercayaan, mengurangi risiko keamanan, dan menjaga keandalan chip sepanjang masa pakainya.

Ke depan, lintasan inovasi VLSI menggarisbawahi peran penting AI dalam membentuk arsitektur komputasi generasi berikutnya. Metodologi berbasis AI seperti sirkuit terpadu 3D (3D IC) dan integrasi heterogen diharapkan dapat mengatasi tantangan yang terkait dengan kepadatan komponen, pembuangan panas, dan optimasi interkoneksi. Bersamaan dengan itu, model pembelajaran adaptif akan memungkinkan evolusi desain secara real-time, memungkinkan sirkuit untuk menyesuaikan diri secara otomatis untuk kinerja, efisiensi, dan toleransi kesalahan. Kerangka kerja AI yang hemat energi akan semakin mendorong inovasi chip yang berkelanjutan, sementara langkah-langkah keamanan berbasis AI yang tertanam akan memperkuat sistem terhadap ancaman yang muncul. Secara kolektif, kemajuan ini menjanjikan realisasi sistem VLSI yang tangguh, cerdas, dan ramah lingkungan yang mampu mendukung era berikutnya dari elektronik berkinerja tinggi dan berdaya sangat rendah.

## 2.12 TRANSFORMASI AI DALAM DESAIN VLSI

Dalam elektronika kontemporer, konvergensi desain VLSI dan AI mewakili pergeseran paradigma mendasar. Melalui metodologi cerdas berbasis data, perancang dapat melampaui hambatan tradisional dalam optimasi PPA—kemampuan yang sangat penting dalam aplikasi daya ultra-rendah. Bab ini telah menunjukkan bagaimana AI merevolusi setiap tahap desain VLSI, dari tata letak fisik dan sintesis logika hingga estimasi daya, NAS yang sadar perangkat keras, dan pengembangan alat EDA, yang pada akhirnya mendefinisikan ulang bagaimana chip dirancang dan direkayasa.

Salah satu kontribusi AI yang paling transformatif terletak pada kemampuannya untuk mengotomatisasi dan memprediksi operasi desain kompleks yang dulunya membutuhkan intervensi manual yang ekstensif. Algoritma berbasis AI sekarang menyederhanakan proses-proses penting seperti perencanaan tata letak, perutean, dan penutupan waktu, secara substansial mengurangi siklus desain sekaligus meningkatkan akurasi dan produktivitas. Pada saat yang sama, model ML dan RL memungkinkan prediksi daya tahap awal, pengurangan kebocoran, dan optimasi energi dinamis—memastikan bahwa sistem VLSI mencapai kinerja superior tanpa mengorbankan batasan daya yang ketat.

Sama pentingnya adalah integrasi AI dalam kerangka kerja desain bersama yang sadar perangkat keras. Dengan menyelaraskan desain NAS dan tingkat sirkuit dengan keterbatasan perangkat keras praktis, AI memastikan pencapaian kinerja tinggi dan konsumsi energi minimal

secara bersamaan. Implementasi dunia nyata—yang mencakup domain industri dan akademis, termasuk yang dipimpin oleh Google dan penyedia alat EDA utama—membuktikan peningkatan yang terukur dalam efisiensi desain, penutupan waktu, pengurangan kemacetan, dan peningkatan hasil produksi tahap pertama di seluruh arsitektur chip yang kompleks.

Namun, di samping potensi transformatifnya, AI memperkenalkan tantangan baru dan pertimbangan etis yang harus ditangani. Interpretasi model, ketergantungan pada kumpulan data besar, risiko overfitting, dan jejak energi yang tinggi dari pelatihan model AI tetap menjadi perhatian pembahasan yang berkelanjutan. Selain itu, imperatif etis—seperti keadilan, transparansi, dan pemanfaatan AI yang bertanggung jawab—semakin menjadi pusat wacana seputar desain berbasis AI.

Ke depan, masa depan desain VLSI yang dibantu AI siap untuk maju melalui model pembelajaran mandiri dan berkelanjutan, integrasi yang lebih dalam ke dalam sistem heterogen dan terintegrasi 3D, kerangka kerja AI yang hemat energi, dan metodologi desain yang berorientasi pada keamanan. Kemajuan tersebut akan memungkinkan terciptanya chip yang cerdas, hemat energi, dan aman yang disesuaikan untuk domain yang sedang berkembang, termasuk perangkat yang dapat dikenakan, sistem otonom, komputasi seluler, dan IoT.

Singkatnya, AI telah berevolusi dari alat pelengkap menjadi penggerak utama desain VLSI generasi berikutnya. Integrasinya memastikan iterasi desain yang dipercepat, peningkatan kinerja, dan pemanfaatan energi yang berkelanjutan—menandai era baru inovasi sistem yang cerdas, hemat daya, dan berkinerja tinggi.

# BAB 3
# KOMPUTASI PERKIRAAN PADA ARSITEKTUR PEMBELAJARAN MESIN

## 3.1 TANTANGAN KOMPUTASI DAN ERA SILIKON GELAP

Perkembangan pesat teknologi disruptif, seperti Kecerdasan Buatan (AI), Pembelajaran Mesin (ML), Pemrosesan Sinyal Digital (DSP), analitik big data, komputasi awan, dan IoT, telah menciptakan pertumbuhan yang belum pernah terjadi sebelumnya dalam kebutuhan komputasi dan penyimpanan. *International Data Corporation* (IDC) memperkirakan bahwa volume data di seluruh dunia akan tumbuh dari 33 zettabyte (ZB) pada tahun 2018 menjadi 175 ZB pada tahun 2025, dengan Tingkat Pertumbuhan Tahunan Majemuk (CAGR) yang luar biasa sebesar 61%. Pertumbuhan yang belum pernah terjadi sebelumnya ini meningkatkan permintaan akan kerangka kerja komputasi yang terukur dan hemat energi, terlebih lagi di lingkungan komputasi yang terbatas seperti platform seluler dan perangkat yang dapat dikenakan. Selama lebih dari 40 tahun, kita telah membangun sistem komputasi berdasarkan dua prinsip fundamental: Hukum Mooreyang terus mendorong kepadatan transistor, dan Skala Dennard, yang sebagian besar gagal selama dekade terakhir. Karena ketidakmampuan untuk menskalakan tegangan suplai dengan dimensi transistor, kita melihat peningkatan kepadatan daya. Bersama dengan keterbatasan termal dan kendala fisik silikon, kegagalan penskalaan ini telah mengantarkan era "Silikon Gelap", di mana efisiensi energi menjadi aspek dominan dan kendala desain baik pada perangkat edge maupun aplikasi cloud.

i) **Kebutuhan Komputasi Daya/Energi Rendah**: Seiring dengan kemajuan dimensi transistor ke rezim nanometer yang dalam, daya dan energi telah menjadi kendala desain utama dalam sistem komputasi modern. Para peneliti memproyeksikan bahwa jika keadaan terus seperti ini, pada tahun 2040, permintaan global untuk komputasi mungkin akan lebih besar daripada total kapasitas planet untuk memenuhi permintaan energi ini. Tantangan ini semakin diperparah oleh proliferasi besar-besaran perangkat IoT (Internet of Things), peningkatan pesat volume data, terutama dari media kaya seperti video dan gambar, dan pertumbuhan opsi infrastruktur siber yang penting, termasuk pusat data skala besar. Tantangan energi hadir di semua platform komputasi, dari perangkat edge berdaya sangat rendah hingga infrastruktur cloud berkinerja tinggi.

   Untuk sistem seluler dan edge, khususnya ketika beroperasi dalam konteks IoT, komputasi hemat energi sangat penting untuk mempertahankan operasi baterai dan memperpanjang masa pakai baterai. Demikian pula, dari sudut pandang pusat data, efisiensi energi sangat penting untuk mengurangi biaya operasional yang terkait dengan listrik dan pendinginan serta untuk memastikan keandalan operasi. Dalam aplikasi yang sangat penting, termasuk mengemudi otonom, konsumsi daya yang berlebihan dapat meningkatkan suhu chip, yang dapat mengganggu keandalan dan menyebabkan kegagalan fatal. Berkaitan dengan hal itu, krisis konsumsi energi diperburuk oleh meningkatnya permintaan akan beban kerja AI, ML, dan DSP serta

jumlah dan kompleksitas data yang mereka butuhkan. Akibatnya, solusinya akan membutuhkan pergeseran signifikan menuju metodologi desain daya rendah untuk memberikan keandalan dan kinerja dengan keberlanjutan yang wajar di berbagai komputasi.

ii) **Kebutuhan akan Komputasi yang Dipercepat**: Pemrosesan aliran data tepat waktu diperlukan dalam banyak aplikasi (misalnya, kendaraan otonom dan robotika, dan misi luar angkasa), tetapi algoritma canggih semakin kompleks, sehingga implementasi perangkat keras menjadi sulit. Misalnya, penerapan teknik AI dan ML (khususnya *Deep Neural Networks* (DNN)) sangat umum, di mana DNN canggih memiliki jutaan parameter. Menjalankan model ini membutuhkan komputasi yang signifikan dalam bentuk operasi aritmatika (penjumlahan dan perkalian), aktivasi nonlinier, dan banyak akses memori, yang dapat menyebabkan latensi yang sangat besar dalam pemrosesan sehingga mencegah penanganan data yang masuk tepat waktu. Penundaan dalam pemrosesan data secara real-time dapat mengurangi akurasi output suatu sistem dan tidak dapat ditoleransi ketika sistem berjalan di edge dengan sumber daya terbatas dan kendala real-time yang ketat.

Komputasi perkiraan telah muncul sebagai kerangka kerja yang menjanjikan dalam ruang desain digital saat ini, menarik perhatian para peneliti dan praktisi industri. Perusahaan-perusahaan besar yang mengembangkan pembahasan dan pekerjaan komersial pada komputasi perkiraan adalah IBM, Google, Intel, dan ARM. Contoh pekerjaan Google adalah Tensor Processing Units (TPU), yang memanfaatkan pengurangan presisi untuk mengurangi konsumsi energi secara signifikan. Terdapat beberapa paradigma lain, seperti komputasi stokastik, komputasi neuromorfik, dan komputasi kuantum, yang telah mendapatkan minat yang kuat di komunitas pembahasan. Tabel 3.1 merinci keempat kerangka kerja ini dan memberikan perbandingan dari kelebihan dan kekurangannya. Berkaitan dengan kelebihan dan kekurangan tersebut, komputasi aproksimasi mendukung tingkat yang dapat diterima dibandingkan dengan kecepatan, akurasi, daya, dan keandalan. Misalnya, komputasi aproksimasi cepat dengan mengorbankan presisi, sedangkan komputasi stokastik hemat daya, tetapi juga lambat dan mungkin tidak akurat.

Manfaat utama komputasi aproksimasi adalah kemampuannya untuk mencapai manfaat energi dan kinerja yang signifikan ketika mentoleransi akurasi numerik yang ketat dalam aplikasi yang toleran terhadap kesalahan. Dalam hal ini, metrik kesalahan dapat disesuaikan sebagai parameter desain untuk menyeimbangkan antara penggunaan energi atau throughput pemrosesan. Meskipun kesalahan biasanya tidak diinginkan, beberapa aplikasi sangat toleran terhadap kesalahan, berdasarkan.

i) Mereka mungkin memproses data yang bising dan berlebihan, yang sering kali berasal dari sensor heterogen.
ii) Mereka menggunakan algoritma probabilistik atau iteratif.
iii) Mereka menghasilkan keluaran di mana sejumlah kecil kesalahan dapat diterima karena keterbatasan resolusi indera manusia.

**Tabel 3.1** Perbandingan fitur-fitur paradigma komputasi yang ada

| Pendekatan Komputasi | Eksak (Exact) | Aproksimasi (Approximate) | Stokastik (Stochastic) | Neuromorfik (Neuromorphic) | Kuantum (Quantum) |
|---|---|---|---|---|---|
| Latensi | Rendah | Rendah hingga sedang | Tinggi | Rendah | Rendah *(dibandingkan dengan kompleksitas permasalahan)* |
| Akurasi | Tinggi | Sedang hingga tinggi | Rendah hingga sedang | Sedang hingga tinggi | Tinggi *(secara teoretis)* |
| Konsumsi Daya | Tinggi | Rendah hingga sedang | Rendah | Rendah | Tinggi *(karena kebutuhan pendinginan)* |
| Keandalan | Tinggi | Sedang hingga tinggi | Sedang | Tinggi | Rendah hingga sedang *(karena ketidakstabilan qubit)* |
| Penggunaan Memori | Tinggi | Rendah hingga sedang | Rendah | Sedang hingga tinggi | Rendah *(karena superposisi qubit)* |

Domain umum komputasi aproksimasi adalah analisis big data, pemrosesan gambar dan sinyal, ML, dan penambangan data. Karena sifat parametrik paradigma ini, ada banyak pertimbangan yang perlu diperhatikan karena dapat diterapkan pada berbagai tingkat abstraksi—dari modifikasi transistor di dalam perangkat keras itu sendiri hingga optimasi yang dilakukan pada tingkat perangkat lunak. Jenis implementasi akan berdampak berbeda pada keandalan perangkat keras dan akurasi output.

Sebagai contoh, mengizinkan aproksimasi hingga 10% kesalahan untuk dijalankan melalui algoritma pemrosesan gambar dapat meningkatkan efisiensi energi dan kecepatan secara signifikan. Demikian pula, mengubah kecepatan penyegaran memori atau hanya menerapkan strategi yang memungkinkan presisi data bervariasi, akan memungkinkan peningkatan kinerja. Namun, komputasi aproksimasi tidak cocok untuk aplikasi yang sangat penting, seperti pengaturan medis atau pertahanan, di mana keandalan dan akurasi sangat penting. Komputasi aproksimasi berfungsi pada empat lapisan dasar: data, perangkat lunak, arsitektur, dan sirkuit (perangkat keras). Salah satu tantangan utama dengan aproksimasi adalah efeknya sangat bervariasi di berbagai aplikasi, yang menghalangi pendekatan universal tunggal. Sebaliknya, pilihan desain harus dibuat berdasarkan konteks, untuk menyeimbangkan efisiensi dan akurasi berdasarkan persyaratan aplikasi spesifik.

## 3.2 KLASIFIKASI TEKNIK KOMPUTASI APROKSIMASI (AXC)

Teknik AxC dapat dibagi menjadi tiga kategori berdasarkan tingkat implementasinya: perangkat lunak, arsitektur, dan perangkat keras. Ada beberapa teknik yang dapat diimplementasikan pada berbagai tingkat ini (ditunjukkan pada Gambar 3.1). Misalnya, dimungkinkan untuk mengaproksimasi pembacaan memori dalam perangkat lunak atau melalui unit kontrol memori. Gambar 3.1 menggambarkan sejumlah aproksimasi yang umum

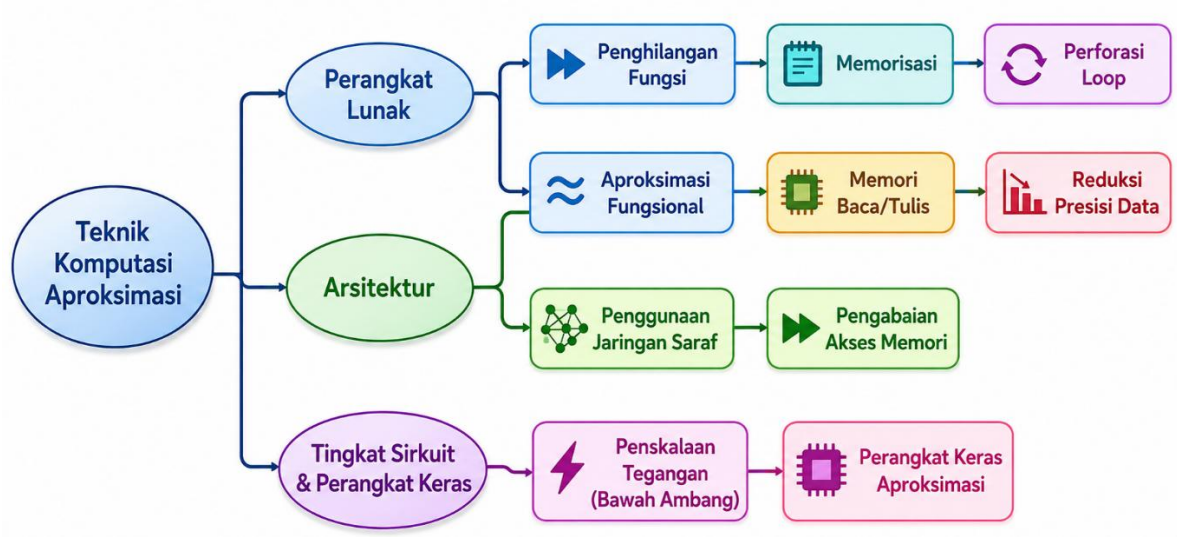
digunakan dalam literatur, tetapi penting untuk disebutkan bahwa ada banyak aproksimasi yang dapat dibuat untuk suatu aplikasi, dan apa yang sebenarnya merupakan aproksimasi adalah ide yang diperdebatkan.

i) **Tingkat Data**: Strategi tingkat data penting untuk daya yang lebih rendah dan kinerja yang lebih baik. Teknik-teknik, termasuk pengambilan sampel, kuantisasi, dan kompresi, memungkinkan kita untuk melakukan trade-off antara kualitas data dan penurunan efisiensi pemrosesan melalui representasi yang disederhanakan atau diperkecil.

ii) **Tingkat Perangkat Lunak:** Strategi komputasi perkiraan berbasis perangkat lunak, seperti perforasi loop, fungsi perkiraan, dan sinkronisasi yang lebih longgar, bertujuan untuk meningkatkan kinerja implementasi kode melalui peningkatan efisiensi dengan mengorbankan tingkat kualitas.

iii) **Tingkat Arsitektur:** Peningkatan efisiensi dan manajemen daya pada tingkat arsitektur merupakan intervensi yang lebih kompleks yang melibatkan desain ulang unit pemrosesan perkiraan khusus dan sistem memori. Desain set instruksi dan adaptasi format presisi variabel juga berkontribusi pada komputasi yang hemat energi.

iv) **Tingkat Sirkuit**: Teknik perkiraan digunakan pada tingkat sirkuit untuk mencapai desain yang lebih hemat daya. Teknik-teknik ini memetakan ulang gerbang logika, menyesuaikan operasi transistor, dan merekayasa ulang unit komputasi agar bekerja di bawah akurasi yang lebih rendah. Semua strategi ini bertujuan untuk menghemat daya dan energi, sekaligus memperkenalkan berbagai tingkat ketidakakuratan dalam komputasi.




**Gambar 3.1** Klasifikasi teknik AxC.

## 3.3 TEKNIK PENDEKATAN PERANGKAT KERAS

Bagian ini kemungkinan besar mengklasifikasikan dan memperkenalkan metode perkiraan tingkat perangkat keras, yang menempati lapisan terbawah dari tingkat hierarki abstraksi desain. Metode ini biasanya diadopsi untuk meningkatkan metrik tingkat sirkuit

seperti area, daya, dan kinerja, yang merupakan metrik penting untuk kinerja prosesor, akselerator, dan platform komputasi. Metode pendekatan perangkat keras secara garis besar dibagi menjadi tiga kelompok: (i) Pendekatan Fungsional Sirkuit (CFA), (ii) Penskalaan Berlebih Tegangan (VOS), dan (iii) Over-Clocking (OC). Dalam kedua kasus tersebut, ada dua jenis kesalahan yang berbeda: kesalahan fungsional dari CFA dan kesalahan waktu dari VOS dan OC. Desain perangkat keras ini, berdasarkan perkiraan definisi komputasi, berfungsi dengan pertimbangan akurasi yang longgar dalam implementasi sirkuit untuk mencapai efisiensi. Bagian ini memberikan gambaran umum tentang metode perkiraan tingkat perangkat keras, yang diklasifikasikan menjadi tiga area unik: Gambar 3.2 perkiraan fungsional, perkiraan data, dan perkiraan pada perangkat.




**Gambar 3.2** Klasifikasi teknik aproksimasi perangkat keras.

i) **Aproksimasi Fungsional**: Teknik ini menggantikan operasi yang rumit (misalnya, perkalian atau fungsi transendental) dengan alternatif yang lebih murah berdasarkan penggunaan tabel pencarian, model linier piecewise, atau pengurangan sirkuit logika, sehingga menghasilkan penghematan daya dan area pada implementasi perangkat keras sambil tetap mencapai akurasi yang dapat diterima dari proses aproksimasi.

ii) **Aproksimasi Data:** Teknik ini menurunkan presisi atau mengurangi data yang diproses melalui algoritma kuantisasi, kompresi, atau pemangkasan. Hal ini menghasilkan pengurangan kebutuhan memori dan bandwidth dan, sebagai hasilnya, meningkatkan kecepatan dan efisiensi energi untuk sistem.

iii) **Aproksimasi Pada Perangkat**: Teknik ini menggabungkan teknik aproksimasi langsung di dalam perangkat edge atau embedded, memungkinkan komputasi dan pemrosesan real-time dengan sumber daya terbatas. Dengan potensi memberikan kinerja yang dapat diterima pada tugas tertentu, pendekatan ini memberikan potensi penghematan

energi dan/atau efisiensi, khususnya untuk aplikasi inferensi, penginderaan, atau kontrol seperti yang diimplementasikan pada platform seluler atau perangkat IoT.

## 3.4 STRATEGI TINGKAT ARSITEKTUR

Desain arsitektur yang memungkinkan pertukaran yang disengaja antara kebenaran, konsumsi sumber daya, dan penyimpanan terkadang dapat mencapai peningkatan kinerja dan efisiensi dibandingkan dengan tidak adanya kesalahan. Secara praktis, keuntungan ini diwujudkan oleh prosesor dan sistem memori aproksimasi, di mana beberapa komputasi atau operasi penyimpanan dilakukan oleh unit khusus yang dirancang untuk aproksimasi. Hal ini menghasilkan komputasi yang diproses dalam waktu yang lebih singkat, tetapi dengan pengurangan akurasi yang diketahui. Perlu dicatat bahwa biasanya, penggunaan material ini dilakukan pada tingkat abstraksi yang lebih tinggi, dengan merestrukturisasi modul fungsional atau "mengoptimalkan" memori daripada mengubah komponen fisik seperti yang dilakukan dalam artikel ini.

**Akselerator Aproksimasi (ASIC, FPGA)**

Dari sisi arsitektur, komputasi aproksimasi mencakup perancangan ulang unit pemrosesan, sistem memori, dan arsitektur set instruksi untuk memungkinkan sistem melakukan pekerjaan dengan efisiensi yang lebih besar. Tujuan komputasi aproksimasi adalah untuk sengaja mengorbankan sedikit akurasi sebagai imbalan atas peningkatan besar dalam efisiensi daya, luas perangkat keras, dan kecepatan komputasi. Berikut adalah beberapa teknik yang banyak digunakan untuk mewujudkan pertukaran ini:

- **Unit Pemrosesan Khusus**: Para peneliti menciptakan komponen perangkat keras perkiraan khusus (pengali, penambah, atau akselerator) yang meningkatkan kecepatan operasi sambil menggunakan lebih sedikit sumber daya.
- **Perluasan Set Instruksi**: Program juga dapat memperluas/memodifikasi set instruksi untuk memungkinkan program meminta operasi perkiraan di bagian eksekusi yang tidak kritis dan memberi pengembang kendali yang lebih akurat atas pertukaran antara akurasi dan penghematan energi.
- **Sistem Memori Perkiraan**: Mengurangi presisi data, mengurangi kecepatan penyegaran memori, atau menggunakan kompresi lossy dapat mengurangi penggunaan daya dan kebutuhan bandwidth sambil tetap memberikan kinerja yang sesuai pada aplikasi target.
- **Arsitektur Heterogen**: Arsitektur ini akan menggabungkan inti yang tepat dan perkiraan. Tugas-tugas kritis akan ditugaskan ke perangkat keras yang tepat, dan tugas-tugas yang dapat mentolerir kesalahan akan dialihkan ke unit perkiraan. Arsitektur heterogen juga dapat disesuaikan untuk memenuhi kebutuhan akan keandalan dengan peningkatan efisiensi.

Pendekatan seperti ini sangat menguntungkan dalam aplikasi seperti pemrosesan gambar, pemrosesan sinyal, ML, dan analitik data bervolume besar, di mana kesalahan yang kecil biasanya tidak mengubah hasil keseluruhan secara signifikan.

Dengan mengimplementasikan konsep arsitektur yang diusulkan pada platform seperti ASIC dan FPGA, akselerator aproksimasi menyediakan realisasi perangkat keras dari konsep-konsep ini. Masing-masing platform ini memiliki keunggulannya sendiri, misalnya, ASIC lebih baik dalam hal kinerja dan konsumsi energi, sedangkan FPGA memiliki prototipe yang lebih cepat dan fleksibilitas.

i) **Aproksimasi Berbasis ASIC**: Akselerator ASIC biasanya difasilitasi dengan blok aritmatika yang lebih sederhana, yaitu, pengali aproksimasi 2 × 2 atau 4 × 4 yang menghilangkan kompleksitas redundan saat menghasilkan produk parsial. Meskipun jenis desain ini menghemat area dan penundaan, komputasi dikorbankan paling banyak untuk tingkat kesalahan yang kecil. Mereka sangat cocok untuk sistem fungsi tetap skala besar di mana efisiensi energi dan kinerja yang lebih tinggi lebih diinginkan daripada presisi yang tepat.

ii) **Aproksimasi Berbasis FPGA**: FPGA memungkinkan implementasi akselerator aproksimasi yang dapat dikonfigurasi ulang. Perancang dapat menghasilkan pengali atau unit fungsional yang ditentukan pengguna menggunakan sumber daya logika (LUT) ini untuk membuat unit khusus. Misalnya, pada, penulis mengusulkan pengali aproksimasi pada FPGA, menghasilkan pertukaran area, latensi, dan energi dalam batas yang dibahas dalam karya lain dari perspektif ASIC; sebenarnya, penyesuaian umum pada pengali aproksimasi ini meningkatkan inspirasi ASIC. Dengan fleksibilitas ini, perancang dapat mengelola akurasi relatif suatu desain dengan lebih baik dan mendukung sub-kondisi operasional yang berbeda.

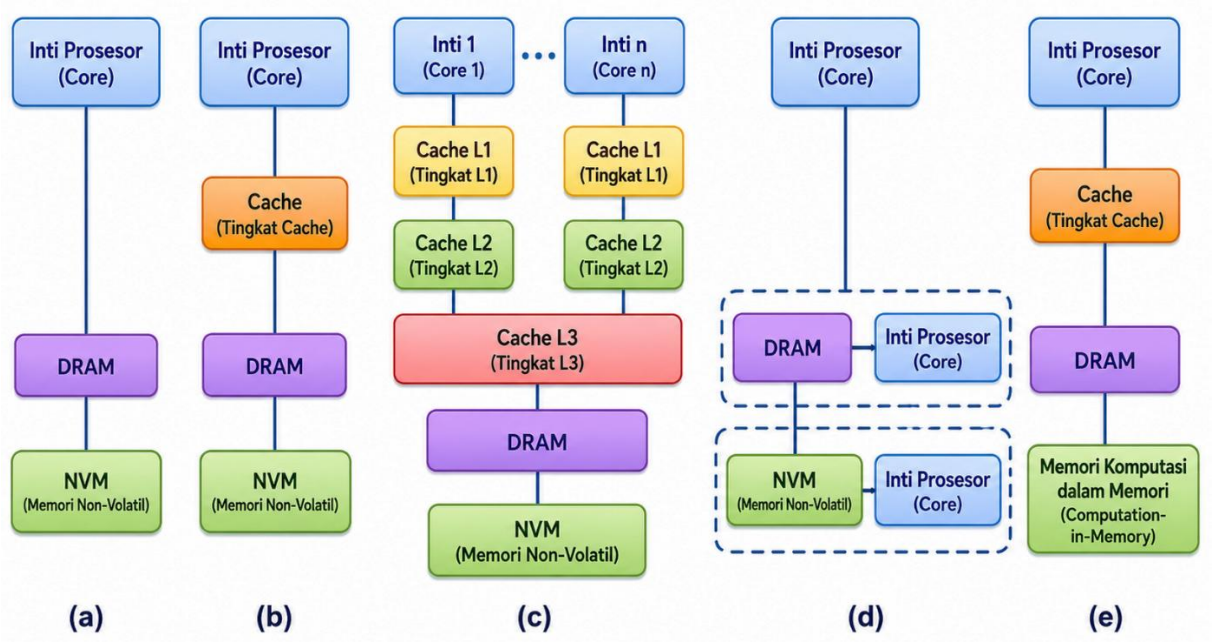
iii) **Kerangka Kerja Otomatis**: Iterasi terbaru dari kerangka kerja telah mengambil jalur Xel-FPGA, menyematkan model statistik dan ML ke dalam alur desain. Kerangka kerja otomatisasi ini memindai ruang desain yang besar dan dengan cepat menemukan akselerator berbasis FPGA yang efisien yang menyeimbangkan sumber daya yang diketahui dengan batas deteksi kesalahan.

Jika digabungkan, akselerator ASIC dan FPGA mewakili dua jalur yang saling melengkapi; ASIC menawarkan efisiensi fungsi tetap yang sangat optimal, sedangkan FPGA menawarkan fleksibilitas dan kecepatan eksplorasi, yang bermanfaat dalam aplikasi yang toleran terhadap kesalahan.

**Hierarki Memori dan Cache Perkiraan**

Secara luas diketahui bahwa subsistem memori adalah salah satu hambatan paling menonjol dalam kinerja dan efisiensi energi sistem komputasi kontemporer. Situasi ini sebagian besar terjadi karena apa yang biasa disebut sebagai "dinding memori", yaitu peningkatan perbedaan di antara prosesor dan waktu akses memori. Komputasi perkiraan adalah arah pembahasan yang layak untuk mengurangi penyebab mendasar dari dinding memori dengan mengurangi presisi di area komputasi yang dapat mentolerir kesalahan terbatas, seperti pemrosesan multimedia, ML, atau analitik data skala besar. Gambar 3.3 menunjukkan klasifikasi sistem komputasi untuk mengilustrasikan di mana data diproses. Terlepas dari klasifikasi ini, mengurangi konsumsi energi memori dalam sistem komputasi masih menghadirkan tantangan, khususnya dengan aplikasi baru, seperti ML, yang menuntut

banyak memori. Hal ini menekankan perlunya solusi baru yang dapat meningkatkan efisiensi energi dan kinerja baik di tingkat konsumen maupun perusahaan.




**Gambar 3.3** Klasifikasi sistem komputer menurut lokasi pemrosesan data: (a, b, dan c) desain konvensional berpusat pada CPU di mana data ditransfer dari memori ke prosesor, (d) model pemrosesan dekat memori dengan pemrosesan terjadi lebih dekat ke memori, dan (e) model komputasi dalam memori dengan pemrosesan dilakukan di dalam memori (misalnya, memori perubahan fase, memristor).

**Memori Perkiraan dalam Sistem Hierarkis**

Untuk mengatasi keterbatasan memori, para peneliti telah mengeksplorasi pendekatan hierarkis, seperti cache on-chip yang lebih besar, integrasi 3D, dan pemrosesan dekat memori. Struktur hierarkis ini memanfaatkan teknik memori perkiraan untuk meningkatkan efisiensi, konsumsi daya, dan latensi akses memori sambil mempertahankan tingkat kualitas output yang dapat diterima. Misalnya, SRAM dapat mengurangi tegangan suplai, DRAM dapat mengurangi kecepatan penyegaran, dan kompresi lossy dapat diterapkan pada data yang disimpan. Jika kita membatasi perkiraan pada area memori yang tidak kritis, sistem masih dapat memastikan akurasi komputasi kritis sambil memberikan peningkatan efisiensi di wilayah yang toleran terhadap kesalahan.

**Cache Perkiraan**

Cache memainkan peran penting dalam hierarki memori, dan komputasi perkiraan telah digunakan untuk desain cache perkiraan, yang lebih hemat energi karena melonggarkan jaminan keandalan yang ketat. Aproksimasi ini dapat terjadi pada tingkat bit, misalnya, mengabaikan kesalahan pada bit paling tidak signifikan (LSB) atau mengurangi mekanisme koreksi kesalahan pada cache tingkat bawah.

Teknologi memori non-volatil baru, seperti STT-MRAM, juga telah digunakan dalam desain cache aproksimasi. Misalnya, QuARK menghadirkan cache yang dapat dikonfigurasi

kualitasnya yang memungkinkan keandalan disesuaikan untuk menukar energi dan akurasi, sementara CAST menghadirkan manajemen cache yang sadar konten yang bertujuan untuk mendukung berbagai tingkat aproksimasi. Sistem seperti itu memungkinkan pertukaran akurasi dan energi dilakukan secara dinamis berdasarkan tuntutan aplikasi.

**Aproksimasi yang Dapat Dikonfigurasi Ulang dan Presisi Adaptif**

Dilema signifikan dalam komputasi aproksimasi adalah bahwa aplikasi dapat memerlukan berbagai tingkat akurasi dan efisiensi. Karena semua metode aproksimasi memengaruhi kinerja dan akurasi secara berbeda, tidak ada satu metode pun yang dapat mencakup semua aplikasi. Akibatnya, pekerjaan di masa mendatang mengarah pada sistem adaptif yang dapat memvariasikan tingkat aproksimasi pada saat runtime. Tujuan dari sistem adaptif adalah untuk mencapai keseimbangan yang dapat diterima antara kesalahan dan energi, serta peningkatan kinerja.

Salah satu jalur pembahasan populer di bidang ini adalah sirkuit perkiraan yang dapat dikonfigurasi ulang, misalnya, adder adaptif, yang mampu mengubah konfigurasinya berdasarkan data masukan dan sinyal pembawa. Sirkuit ini dapat mengenali konteks di mana operasi aritmatika dapat menyebabkan kesalahan besar dan mengkonfigurasi ulang ke mode yang lebih tepat selama konteks ini, sambil mempertahankan perkiraan yang kurang tepat untuk komputasi yang kurang sensitif. Melalui penyesuaian dinamis ini, seseorang dapat mencapai efisiensi energi dan optimasi kinerja dengan mempertahankan akurasi yang dibutuhkan untuk aplikasi yang sangat presisi. Lebih baik lagi, sirkuit adaptif dapat melakukan keduanya, yaitu beradaptasi dengan karakteristik masukan dan, pada saat yang sama, dengan karakteristik komputasi untuk memberikan keseimbangan yang wajar antara peningkatan keandalan atau peningkatan kinerja. Fleksibilitas ini juga berkaitan dengan desain bersama lintas lapisan, di mana perkiraan terjadi dengan dua modul atau lebih, dan pada beberapa tingkat tumpukan komputasi.

Dalam beban kerja ML, misalnya, strategi aproksimasi dapat diterapkan secara bersamaan pada tingkat perangkat keras (misalnya, unit aritmatika aproksimasi), pada tingkat arsitektur (misalnya, hierarki memori), dan pada tingkat perangkat lunak (misalnya, kompresi dan kuantisasi model). Lebih jauh lagi, telah ada studi yang meneliti sirkuit tercetak untuk ML, di mana aproksimasi memberikan penghematan energi dan desain yang disederhanakan. Secara keseluruhan, aproksimasi yang dapat dikonfigurasi ulang dan aliran lintas lapisan menandakan desain arsitektur ML yang fleksibel dan sadar konteks yang dapat dioptimalkan untuk berbagai kasus penggunaan aplikasi.

## 3.5 PENDEKATAN TINGKAT PERANGKAT LUNAK

Komputasi perkiraan telah muncul sebagai paradigma baru dalam rekayasa komputer yang bertujuan untuk mengurangi kompleksitas komputasi dan konsumsi energi dengan sengaja melonggarkan persyaratan akurasi yang ketat dalam komputasi. Pendekatan ini sangat bermanfaat dalam kasus di mana hasil yang akurat tidak penting, atau ketika biaya komputasi yang tepat menjadi terlalu kompleks, boros sumber daya, dan/atau memakan waktu sehingga tidak sepadan, dan diinginkan untuk persyaratan yang lebih lunak. Seiring aplikasi modern

menjadi lebih canggih, aplikasi tersebut diharapkan untuk terus beradaptasi dengan tuntutan yang kompleks; menyediakan layanan baru; dan memproses sejumlah besar data. Dengan meningkatnya kompleksitas ini, hal ini akan berkontribusi pada peningkatan biaya peningkatan dan kebutuhan daya, eksekusi, dan memori.

Pada tingkat perangkat lunak, komputasi perkiraan berfokus pada kompleksitas kegagalan pemrosesan aplikasi, yang dicirikan dalam hal jumlah operasi yang diselesaikan dan/atau jumlah referensi memori. Dengan merasionalisasi faktor-faktor ini, total biaya implementasi sistem perangkat lunak dapat diturunkan secara signifikan. Telah banyak pembahasan yang mengeksplorasi aproksimasi tingkat perangkat lunak yang hanya memperkenalkan ketidakakuratan (atau aproksimasi) pada komponen komputasi tertentu. Tujuannya adalah mencapai peningkatan kompleksitas dan kinerja nyata dengan kualitas yang dapat diterima.

Salah satu tantangan utama di bidang ini adalah mengidentifikasi komputasi yang tidak secara signifikan merusak akurasi dan keandalan hasil akhir, setelah kita memilih untuk melakukan aproksimasi. Jadi, pendekatan aproksimasi tingkat perangkat lunak adalah pendekatan sistematis yang berguna untuk menghasilkan, menganalisis, dan memelihara sistem perangkat lunak tanpa mengganggu fungsionalitas dasar. Tujuan dari metode ini adalah untuk mengurangi pertukaran antara penyederhanaan dan tingkat kinerja yang dapat diterima. Perlu dicatat, pendekatan aproksimasi tingkat perangkat lunak dapat diterapkan pada berbagai tahap dalam siklus hidup pengembangan perangkat lunak, khususnya desain konseptual, implementasi, dan pengujian program. Oleh karena itu, komputasi aproksimasi tingkat perangkat lunak menyediakan kerangka kerja yang fleksibel untuk memperkenalkan peningkatan efisiensi yang mungkin sambil memastikan kinerja sistem cukup kuat untuk aplikasi praktis.

Tabel 3.2 mengkategorikan dan membahas teknik aproksimasi yang diimplementasikan pada tingkat perangkat lunak, yang sesuai dengan tingkatan atas hierarki abstraksi desain. Tujuan utama komputasi aproksimasi berbasis perangkat lunak adalah untuk meningkatkan kecepatan eksekusi program dan/atau mengurangi konsumsi energi sistem.

**Tabel 3.2** Teknik aproksimasi pada tingkat perangkat lunak

| **Teknik** | **Manfaat** | **Tantangan** |
|---|---|---|
| Melewati Komputasi (Computation Skipping) | Mengurangi waktu eksekusi dan konsumsi energi dengan melewati komputasi yang tidak penting. | Pemilihan komputasi yang akan dilewati harus dilakukan secara cermat agar tidak menyebabkan penurunan akurasi keluaran yang signifikan. |
| Perforasi Perulangan (Loop Perforation) | Mengurangi waktu eksekusi secara signifikan pada algoritma berbasis perulangan. | Menentukan jumlah iterasi yang optimal untuk dilewati tanpa mengurangi akurasi hasil merupakan tantangan yang cukup besar. |
| Pembukaan Perulangan Aproksimasi | Mempercepat eksekusi dan menghemat energi pada | Akurasi hasil interpolasi harus dikendalikan dengan baik agar kualitas keluaran tetap dapat diterima. |

| | | |
|---|---|---|
| (Approximate Loop Unrolling) | algoritma berbasis perulangan. | |
| Agregasi Perulangan (Loop Aggregation) | Mengurangi kompleksitas dan waktu eksekusi pada model simulasi. | Pemilihan tingkat agregasi yang tepat sangat penting agar peningkatan kinerja tidak mengorbankan akurasi. |
| Melewati Tugas (Task Skipping) | Mempercepat komputasi paralel dengan melewati tugas-tugas tertentu. | Tugas-tugas yang bersifat kritis harus diidentifikasi dan tidak boleh dilewati agar kebenaran aplikasi tetap terjaga. |
| Sinkronisasi Longgar (Relaxed Synchronization) | Mengurangi *cache miss* dan menghemat energi pada program paralel. | Berpotensi menyebabkan kegagalan program (*crash*) apabila bagian kritis yang diperkirakan tidak ditangani dengan benar. |
| Memorisasi (Memorization) | Mempercepat eksekusi dengan menggunakan kembali hasil fungsi yang telah dihitung sebelumnya. | Diperlukan prediksi yang akurat terhadap nilai yang sering digunakan agar tidak menimbulkan kesalahan keluaran yang signifikan. |
| Aproksimasi Fungsi (Function Approximation) | Menghemat energi dengan mengaproksimasi fungsi-fungsi yang kurang kritis. | Pemilihan fungsi yang akan diaproksimasi harus dilakukan secara hati-hati, karena keputusan yang kurang tepat dapat menyebabkan akurasi keluaran menjadi tidak dapat diterima. |
| Implementasi Jaringan Saraf (Neural Network Implementation) | Mempercepat eksekusi bagian-bagian kode yang menggunakan pendekatan aproksimasi melalui jaringan saraf. | Mengembangkan model jaringan saraf yang akurat sehingga mampu menghasilkan hasil yang sebanding dengan tingkat kesalahan yang masih dapat diterima merupakan tantangan utama. |
| Aproksimasi Nilai Muatan (Load Value Approximation) | Mengurangi latensi *cache miss* dan konsumsi energi. | Melatih model aproksimasi nilai muatan agar mencapai tingkat akurasi yang dapat diterima merupakan proses yang menantang. |
| Aproksimasi Memori Baca/Tulis (Read/Write Memory Approximation) | Menghemat energi dan penggunaan memori pada aplikasi pemrosesan citra dan video. | Menyeimbangkan antara pengurangan penggunaan memori dan kualitas keluaran yang tetap dapat diterima merupakan hal yang kompleks. |
| Pengurangan Presisi Data (Data Precision Reduction) | Mengurangi jejak penggunaan memori (*memory footprint*) dan konsumsi energi. | Menentukan keseimbangan yang tepat antara pengurangan presisi data dan kualitas keluaran yang masih dapat diterima menjadi tantangan utama. |

**Algoritma dan Pustaka Perkiraan**

Tujuan utama teknik perkiraan tingkat algoritma adalah untuk mengetahui seberapa banyak data dan komputasi aplikasi dapat disederhanakan tanpa terlalu memengaruhi output

akhir. Hal ini membantu mengurangi kebutuhan pemrosesan dan memori. Dalam kebanyakan kasus, perkiraan ini bekerja paling baik ketika algoritma dirancang bersama dengan perangkat keras yang mendasarinya, meskipun beberapa masih dapat diterapkan bahkan tanpa dukungan perangkat keras khusus.

Salah satu metode perkiraan yang paling banyak digunakan dalam AI dan ML adalah kuantisasi, juga dikenal sebagai penskalaan presisi. Pendekatan ini diterapkan baik dalam pelatihan model maupun inferensi dan telah menjadi standar di industri. Sederhananya, kuantisasi mengurangi jumlah bit yang digunakan untuk merepresentasikan bobot dan aktivasi dalam sebuah model. Menurunkan presisi bit membawa banyak manfaat: hal ini secara signifikan mengurangi biaya komputasi (misalnya, pengali 16-bit bisa sekitar empat kali lebih murah daripada pengali 32-bit) dan juga mengurangi penggunaan memori dan kebutuhan transfer data. Pada saat yang sama, ia mempertahankan pola komputasi reguler yang umum dalam model AI/ML, yang membuatnya cocok untuk prosesor tujuan umum dan akselerator perangkat keras khusus.

Kekurangan utama dengan kuantisasi adalah memastikan bahwa pengurangan presisi numerik tidak merusak akurasi model. Untuk pelatihan, para peneliti telah menguji beberapa format floating-point 16-bit (FP16), seperti IEEE-FP16 (1-5-10), B Float (1-8-7), dan DL Float (1-6-9). Baru-baru ini, format Hybrid-FP8 (HFP8) diperkenalkan, yang menggabungkan dua format FP8: (1,4,3) untuk bobot dan aktivasi, dan (1,5,2) untuk kesalahan, untuk lebih mengurangi biaya pelatihan.

Untuk inferensi, representasi fixed-point (FxP) lebih umum. Dua strategi utama digunakan di sini:

i) **Kuantisasi Pasca-Pelatihan (PTQ):** Mengonversi model floating-point langsung ke bentuk fixed-point. Meskipun sederhana, metode ini sering mengurangi akurasi, terutama di bawah delapan bit, meskipun kalibrasi ulang statistik normalisasi batch dapat membantu.

ii) **Pelatihan Sadar Kuantisasi (QAT):** Melatih ulang model sambil mensimulasikan efek kuantisasi, memungkinkan bobot untuk beradaptasi. Metode ini mencapai akurasi yang jauh lebih baik, dengan pendekatan seperti PACT (Parameterized Activation Clipping) dan SAWB (*Statistics-Aware Weight Binning*) mencapai hasil yang mendekati state-of-the-art bahkan dengan presisi INT4, dan hanya menunjukkan sedikit penurunan akurasi pada INT2. Perlu juga dicatat bahwa kuantisasi umumnya hanya diterapkan pada bagian model AI/ML yang paling berat komputasinya, yaitu operasi konvolusi (CONV) dan perkalian matriks (GEMM). Jika memungkinkan, lebih baik untuk mempertahankan operasi lain seperti aktivasi, pooling, dan normalisasi dalam FP16. Lapisan dengan tingkat presisi berbeda juga dapat digunakan dalam model yang sama. Untuk menjalankan model AI/ML secara efisien pada perangkat dengan keterbatasan sumber daya (misalnya, ponsel, perangkat koleksi, atau sistem IoT), model kompak yang mampu memperoleh akurasi yang akan dihasilkan oleh jaringan yang lebih besar sangat dibutuhkan. Lapisan seperti lapisan bottleneck terbalik, konvolusi terpisah kedalaman, dan konvolusi grup membantu mengurangi ukuran model sambil

meminimalkan komputasi. Bersamaan dengan prinsip desain manual lainnya, teknik seperti distilasi pengetahuan atau Pencarian Arsitektur Neural (NAS) otomatis menjadi praktik yang lebih umum untuk menghasilkan model yang kompak tetapi berkinerja tinggi.

iii) **Pemangkasan/Sparsifikasi dan Kompresi:** Salah satu teknik yang paling umum digunakan untuk mengurangi ukuran model adalah dengan memangkas, yaitu, mengatur bobot tertentu dari jaringan saraf menjadi nol dan menghapus koneksi neuron yang terkait. Tensor bobot menjadi jarang, yang berarti pada akhirnya dapat disimpan lebih kompak (contoh: hanya pasangan indeks-nilai yang dibutuhkan). Kendala utama dalam pemangkasan adalah bahwa hal itu menyebabkan pola akses memori yang tidak teratur untuk bobot, yang menyebabkan akselerasi perangkat keras yang kurang efektif. Untuk mengatasi hal ini, metode pemangkasan terstruktur yang lebih baru, seperti sparsitas blok, di mana tensor dibagi menjadi blok berukuran tetap dengan elemen non-nol yang terkontrol, membantu menjaga keteraturan. Mirip dengan kuantisasi, pemangkasan biasanya dilakukan selama pelatihan sehingga kehilangan akurasi dapat dipulihkan. Dalam pelatihan terdistribusi, salah satu hambatan utama adalah komunikasi gradien bobot di luar chip di beberapa pembelajar. Untuk mengurangi biaya ini, algoritma kompresi gradien digunakan, mencapai tingkat kompresi lebih dari 100 × tanpa kehilangan akurasi yang signifikan.

iv) **Pengambilan Sampel dan Sub-set Data**: Pengambilan sampel data hanya menggunakan sebagian dari tensor untuk komputasi, menurunkan penggunaan memori dan mengurangi kebutuhan bandwidth. Karena nilai data yang berdekatan (misalnya, piksel dalam gambar) seringkali serupa, pengambilan sampel bekerja secara efektif dalam praktiknya. Beberapa komputasi redundan yang diperkenalkan oleh pengambilan sampel dapat dihilangkan kemudian untuk efisiensi.

v) **Sinkronisasi Longgar**: Dalam pelatihan terdistribusi, pembelajar biasanya melakukan sinkronisasi setelah setiap minibatch sehingga bobot dapat diperbarui secara konsisten. Sinkronisasi longgar memungkinkan pembelajar untuk melanjutkan secara semi-sinkron atau asinkron, artinya pembelajar yang lebih cepat tidak perlu menunggu pembelajar yang lebih lambat. Komprominya adalah untuk memastikan bahwa hal ini tidak meningkatkan jumlah epoch yang diperlukan agar model konvergen.

**Optimasi Tingkat Kompiler**

Optimasi tingkat kompiler memainkan peran penting dalam memungkinkan komputasi aproksimasi dengan secara otomatis menerapkan trade-off akurasi-efisiensi selama kompilasi program. Kompiler dengan kemampuan aproksimasi memeriksa kode aplikasi untuk mengidentifikasi area non-kritis di mana kesalahan kecil memiliki sedikit pengaruh pada kualitas produk akhir.

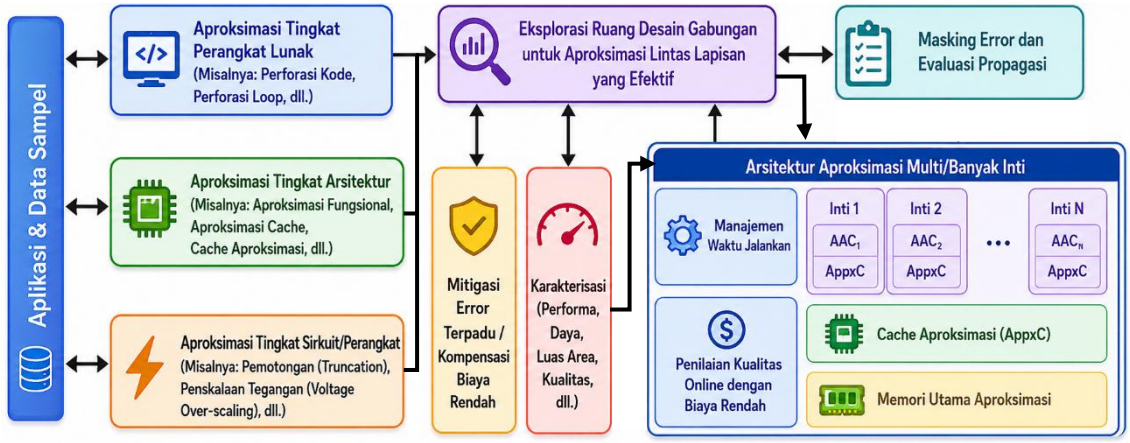
Ketika aproksimasi diidentifikasi, kompiler akan menyajikan berbagai teknik aproksimasi untuk mengatasi situasi tersebut (misalnya, sinkronisasi yang dilonggarkan, melewati tugas, aproksimasi operator, dan perforasi loop (alias, melewati iterasi)). Kompiler-runtime hibrida mengadopsi aproksimasi ini secara statis (pada waktu kompilasi) atau adaptif

(pada waktu eksekusi). Misalnya, ACCEPT menggabungkan analisis statis dan dinamis untuk mengidentifikasi bagian-bagian program yang dapat mentolerir aproksimasi.

Hal ini diperluas lebih lanjut oleh alat-alat canggih seperti ApproxTuner dan mARGOt, yang secara dinamis melakukan auto-tuning aplikasi untuk mempertahankan batasan kualitas sambil memaksimalkan waktu eksekusi dan penggunaan energi. Tugas tersulit untuk aproksimasi tingkat kompiler adalah menemukan titik keseimbangan antara kehilangan akurasi yang dapat diterima dan peningkatan kinerja yang signifikan. Untuk mencegah aproksimasi yang menghasilkan kinerja yang tidak dapat diterima, kompiler harus menggunakan heuristik, model penyebaran kesalahan, dan pembuatan profil khusus aplikasi. Secara umum, teknik aproksimasi semacam ini akan secara signifikan mengurangi beban pengembang sekaligus memungkinkan penerapan komputasi aproksimasi yang terukur di berbagai domain perangkat lunak. Memang benar bahwa optimasi tingkat kompiler akan menurunkan biaya pengembangan masalah sampai batas tertentu.

**Alur Desain Lintas Lapisan yang Mengintegrasikan Perangkat Keras dan Perangkat Lunak**

Dari kerangka kerja perangkat lunak dan algoritma hingga sirkuit perangkat keras, alur desain lintas lapisan menggabungkan teknik aproksimasi di seluruh tumpukan komputasi (Gambar 3.4). Di bawah batasan kualitas yang ditentukan pengguna, metodologi ini memaksimalkan efisiensi energi dan kinerja dengan mengoordinasikan trade-off di seluruh perangkat lunak, arsitektur, memori, dan perangkat keras, daripada membatasi aproksimasi pada satu tingkat saja.




**Gambar 3.4** Integrasi lintas lapisan strategi aproksimasi di seluruh tumpukan komputasi.

Studi pembahasan telah menunjukkan bahwa peningkatan efisiensi yang signifikan dapat dicapai dengan menggabungkan optimasi tingkat perangkat keras, seperti unit aritmatika perkiraan, hierarki memori, atau penskalaan tegangan, dengan metode tingkat perangkat lunak seperti kuantisasi, pemangkasan, atau kompresi model. Pada FPGA, perangkat IoT, akselerator pembelajaran mendalam, dan sistem lain yang terbatas sumber daya, sinergi ini meningkatkan pemanfaatan sumber daya.

Contoh kerangka kerja lintas lapisan meliputi:

- **Metodologi Plastisin**: Kerangka kerja untuk unit pengali-pembagi SIMD perkiraan yang mencapai throughput tinggi dengan ketidakakuratan yang terkontrol.
- **Xel-FPGA:** Kerangka kerja desain ujung-ke-ujung otomatis yang mempercepat eksplorasi akselerator perkiraan berbasis FPGA menggunakan metode statistik dan ML.
- **Akselerator AI Komersial (misalnya, Google TPU, IBM AI Core):** Menggunakan aritmatika presisi rendah dan alur perkiraan perangkat keras-perangkat lunak untuk menyeimbangkan kecepatan, akurasi, dan efisiensi daya.

Pendekatan sistem penuh juga telah diterapkan pada sistem kamera pintar dan mesin inferensi DNN. Dalam sistem ini, perkiraan tingkat sensor seperti subsampling piksel mengurangi konsumsi energi dan beban komputasi, perkiraan tingkat memori seperti peningkatan interval penyegaran DRAM menghemat energi sambil mempertahankan data penting, dan teknik tingkat komputasi seperti perforasi loop atau penghentian dini meminimalkan perhitungan yang tidak perlu. Dengan menyetel semua parameter aproksimasi secara sistematis, mengoptimalkan resolusi sensor, parameter memori, dan strategi komputasi, penghematan energi yang signifikan dapat dicapai dengan kehilangan kualitas minimal.

Sistem aproksimasi terdistribusi memperluas prinsip-prinsip ini ke beberapa perangkat. Aproksimasi terkoordinasi memungkinkan inferensi kolaboratif DNN yang hemat energi di edge. Mereka menggunakan pemangkasan dan penyetelan dinamis parameter aproksimasi di seluruh subsistem untuk memenuhi standar kualitas [30]. Semakin banyak teknologi AI baru menggabungkan aproksimasi di seluruh algoritma, perangkat keras, dan lapisan kompilator. Beberapa metode tersebut adalah struktur CNN dinamis, pengurangan kernel, kuantisasi lapisan demi lapisan, pemangkasan aktivasi, penskalaan tegangan, dan penskalaan presisi. Menambahkan aproksimasi ke proses pelatihan dapat membantu mengimbangi kehilangan akurasi yang disebabkan oleh aproksimasi tingkat perangkat keras dan membuat seluruh proses lebih hemat energi.

Komputasi aproksimasi sangat berguna ketika sumber daya terbatas, seperti ketika perangkat ML berjalan dengan daya rendah atau dicetak. Klasifikasi cetak lebih murah daripada desain CMOS tradisional, dan Anda dapat membuatnya sesuai kebutuhan Anda dengan menggunakan arsitektur khusus. Kerangka kerja otomatis menggunakan aproksimasi pada tingkat model dan sirkuit untuk membuat sirkuit yang menggunakan lebih sedikit energi dan hampir sempurna. Algoritma pelatihan ulang, bersama dengan aproksimasi tingkat perangkat keras yang menghilangkan informasi yang kurang penting dalam neuron, membuat efisiensi area dan daya menjadi lebih baik. Hal ini memungkinkan perangkat bertenaga baterai untuk menggunakan pengklasifikasi yang rumit.

Alur desain lintas lapisan meminimalkan fragmentasi desain, mengurangi waktu eksplorasi, dan menciptakan pertukaran yang kuat antara akurasi, latensi, dan efisiensi energi dengan mengoptimalkan beberapa lapisan sekaligus. Ketika algoritma dapat menangani perhitungan perkiraan, teknik ini bekerja dengan baik dengan optimasi tingkat perangkat keras, dan teknik ini memiliki potensi besar dalam ML dan bidang lain di mana kesalahan dianggap wajar.

## 3.6 TANTANGAN DAN ARAH MASA DEPAN

Sejumlah masalah penting perlu diselesaikan jika komputasi perkiraan (AC) ingin digunakan secara lebih luas di dunia nyata.

i) **Aproksimasi Lintas Lapisan dan Tingkat Sistem**: Meskipun sebagian besar pekerjaan sebelumnya berfokus pada pembuatan aproksimasi dalam satu lapisan tumpukan komputasi, seperti aproksimasi dalam algoritma, memori, dan/atau perangkat keras, manfaat sebenarnya berasal dari fokus pada aproksimasi di berbagai lapisan sistem yang menggunakan aproksimasi dalam pendekatan terkoordinasi. Dengan kata lain, menggunakan aproksimasi dalam interaksi berbagai lapisan yang mencakup perangkat lunak, arsitektur, memori, dan perangkat keras untuk mendapatkan "titik optimal" untuk kualitas dan efisiensi keluaran. Nilai sebenarnya dari aproksimasi terjadi ketika memilih aproksimasi yang tepat pada lapisan yang tepat untuk suatu sistem. Meskipun konsep ini telah disentuh dalam beberapa studi sebelumnya, alat otomatis yang secara metodis akan mengungkap trade-off di seluruh sistem masih belum ada hingga saat ini. Salah satu tantangannya adalah karena kita belum memiliki metodologi yang dapat mengukur secara andal bagaimana kombinasi beberapa aproksimasi, dalam setiap lapisan tumpukan, memperkenalkan perubahan akurasi dan keandalan keseluruhan untuk aplikasi.

ii) **Pemodelan Perangkat Keras**: Kekurangan lain berkaitan dengan tidak adanya model perangkat keras yang tervalidasi. Model perangkat keras yang tervalidasi sangat berharga dalam menavigasi ruang desain sistem dan mempertimbangkan masalah tingkat aproksimasi mana yang akan diterapkan pada setiap item perangkat keras. Seringkali, model menggunakan asumsi yang agak terlalu disederhanakan mengenai distribusi kesalahan atau presisi (misalnya, memori berbasis VOS dan ReRAM, dll.). Dengan demikian, hasil studi yang ada mungkin tidak dapat dibandingkan secara langsung satu sama lain karena model yang berbeda. Untuk tujuan perbandingan yang adil, dan untuk memastikan penggunaan yang valid dalam desain lintas lapisan, kita membutuhkan model standar dan tervalidasi yang mungkin menggunakan beberapa pemodelan intensif simulasi yang divalidasi melalui pemodelan.

iii) **Tolok Ukur**: Tantangan lain adalah tolok ukur yang andal. Perilaku metode aproksimasi jelas dimotivasi oleh kumpulan data dan aplikasi. Oleh karena itu, memiliki serangkaian tolok ukur representatif yang komprehensif sangat penting. Rangkaian yang ada, AxBench, misalnya, hanya menguji beberapa skenario. Di luar kriteria yang disepakati komunitas untuk memilih kasus uji, diperlukan skema kerja benchmark, dataset, dan metode evaluasi dengan lebih banyak skenario dan keragaman. Dan dengan itu, evolusi beban kerja yang cepat dari kemajuan teknologi dari waktu ke waktu akan membantu memberikan skala evaluasi yang tidak memihak dan lebih adil pada aplikasi.

iv) **Metrik Evaluasi**: Metrik menginformasikan proses desain dan eksplorasi dengan menguraikan pertimbangan antara kinerja, efisiensi, dan kualitas. Metrik yang dipilih secara tidak tepat dapat mengarahkan sistem untuk memilih desain yang kurang optimal dan pada akhirnya mengganggu pengalaman pengguna. Oleh karena itu, setiap

domain aplikasi membutuhkan metrik penilaian yang terdefinisi dengan baik dan dapat dipercaya yang dapat dievaluasi terhadap aspek fungsionalitas dan keandalan yang paling signifikan.

v) **Pengujian Pasca-Fabrikasi**: Pada perangkat skala nano, variasi dan cacat manufaktur sudah menjadi masalah utama. Pengujian menjadi lebih menantang karena adanya perkiraan, terutama jika metode deteksi kesalahan tradisional mungkin tidak dapat mengakomodasi kesalahan yang disengaja. Pengujian perangkat keras perkiraan hanya dipelajari secara terbatas; namun, kita harus mempertimbangkan beberapa paradigma pengujian yang akan memastikan bahwa perangkat dapat dievaluasi tanpa kehilangan hasil yang berlebihan, mengingat bahwa perangkat yang biasanya tidak dapat diterima dengan persyaratan penerimaan yang lebih ketat dapat diselamatkan menggunakan teknik seperti pengelompokan yang sadar akan perkiraan.

vi) **Reproduksibilitas dan Pembahasan Terbuka:** Terakhir, reproduksibilitas masih menjadi masalah penting. Banyak artikel yang diterbitkan tidak mendokumentasikan metode mereka secara memadai dan oleh karena itu tidak memberikan informasi yang cukup bagi orang lain untuk mereproduksi hasilnya secara akurat. "Defisit pengetahuan" ini menciptakan hambatan bagi orang lain untuk membuat skema, membandingkan, atau membangun pembahasan sebelumnya. Selain memperkuat transparansi, peningkatan penggunaan alat, kumpulan data, dan kerangka kerja sumber terbuka akan menciptakan kapasitas untuk adaptasi kolaboratif di berbagai bidang pengetahuan, dengan pengetahuan yang diciptakan bersama dengan cepat.

## 3.7 APROKSIMASI TERKONTROL DALAM ML

Gagasan tentang aproksimasi terkontrol (AC) telah secara luas dianggap sebagai strategi yang baik yang mampu memberikan solusi yang memuaskan terhadap peningkatan kebutuhan komputasi dan energi yang ditimbulkan oleh model ML terkini. Studi telah menunjukkan bahwa banyak aplikasi ML, khususnya DNN, dapat mentolerir tingkat kesalahan yang terbatas. Toleransi ini memungkinkan perancang untuk membuat aproksimasi terkontrol tanpa menurunkan akurasi secara signifikan. Dimungkinkan untuk menurunkan penggunaan energi, luas perangkat keras, dan penundaan pemrosesan dengan hati-hati menurunkan presisi. Semua ini merupakan faktor penting dalam pelatihan dan inferensi. Untuk mendapatkan semua manfaat ini, aproksimasi harus digunakan di banyak tingkatan tumpukan ML, mulai dari desain sirkuit dan manajemen memori hingga arsitektur sistem dan kerangka kerja perangkat lunak.

Meskipun terdapat pekerjaan yang menggembirakan dalam AC di ML, masih ada tantangan signifikan yang dihadapi dalam mengimplementasikannya ke dalam alur kerja ML arus utama. Bidang ini tidak hanya kekurangan konsensus seputar pemodelan perangkat keras, tetapi masalah benchmarking, metodologi evaluasi, dan kesesuaian bidang juga mempersulit jalan menuju kepercayaan portabel dari tugas ML ke tugas ML lainnya. Ketidakmampuan untuk melakukan benchmarking secara menyeluruh di berbagai tugas dan menyederhanakan perbandingan berbagai desain AC, karena tugas ML sering dibangun di atas tugas evaluasi yang

berbeda, masih menghambat implementasi AC. Ekosistem desain analog belum sepenuhnya mengembangkan alat otomatis yang menggunakan teknik AC dari desain hingga metodologi implementasi, yang berarti bahkan skema pengujian otomatis yang dipikirkan dengan baik pun memiliki teknik pengujian yang lebih realistis melalui pengaturan pengujian pasca-fabrikasi. Agar bidang ini dapat maju, diperlukan kerangka kerja dan kumpulan data yang mudah diakses, terdefinisi dengan baik, dan terstandarisasi dalam lingkungan sumber terbuka, sehingga setiap tugas dapat di-benchmark secara terpisah dan bermanfaat di sepanjang desain multi-tugas berdasarkan prosesor yang dirancang.

# BAB 4
# DESAIN SIRKUIT TERPADU HEMAT DAYA BERBASIS AI DAN ML

## 4.1 ERA BARU DESAIN VLSI BERDAYA RENDAH BERBASIS AI

Abad ke-21 ditandai dengan meningkatnya ketergantungan pada sistem elektronik yang kecil, portabel, dan cerdas. Harapan yang diterapkan pada sirkuit terpadu (IC) sejauh menyangkut ponsel pintar dan perangkat yang dapat dikenakan, begitu pula kendaraan otonom dan platform komputasi tepi. Konsumen juga membutuhkan tidak hanya kinerja dan konektivitas yang tinggi, tetapi juga masa pakai baterai yang lama dan pengoperasian yang berkelanjutan. Pola prioritas desain yang umum sebelumnya di bidang desain Integrasi Skala Sangat Besar (VLSI) adalah bertujuan untuk meningkatkan kinerja dan mengurangi luas area desain karena Hukum Moore tentang kinerja dan penskalaan Dennard tentang luas area. Ukuran node teknologi CMOS yang semakin kecil, dari mikrometer, nanometer, dll., memungkinkan peningkatan kecepatan transistor, peningkatan kepadatan integrasi, dan pengurangan biaya per fungsi yang signifikan. Dalam beberapa dekade terakhir, paradigma penskalaan ini cukup untuk mempertahankan peningkatan kinerja komputasi secara eksponensial. Namun, dengan ukuran perangkat yang semakin kecil hingga skala sub-mikron dan nanometer, konsumsi daya menjadi kendala utama untuk pengembangan lebih lanjut.

Secara tradisional, desain sirkuit berfokus pada memaksimalkan ukuran kinerja seperti kecepatan clock, throughput, dan efisiensi area. Meskipun demikian, dengan pengurangan ukuran perangkat semikonduktor hingga skala sub-mikron dan nanometer, konsumsi energi telah menjadi persyaratan desain utama. Daya rendah bukan lagi pilihan; sebaliknya, daya rendah menjadi batasan utama yang menentukan keberhasilan, kegunaan, dan kelayakan suatu produk saat ini. Munculnya *Internet of Things* (IoT), komunikasi antar mesin, dan perangkat wearable merupakan bukti perubahan ini. Miliaran perangkat yang terhubung ke jaringan perlu memiliki batasan daya yang ketat dan dapat ditenagai oleh baterai kecil atau skema pengumpulan energi.

Demikian pula, di pusat data dan sistem komputasi berkinerja tinggi (HPC), daya yang dikonsumsi berbanding lurus dengan biaya operasional, kebutuhan pendinginan, dan jejak karbon. Laporan di industri menunjukkan bahwa hampir 8% dari konsumsi listrik global mungkin akan dialokasikan untuk penggunaan energi oleh pusat data sebelum dekade ini berakhir, tanpa memprioritaskan efisiensi pusat data. Hal ini semata-mata karena sistem tersebut belum ditempatkan dalam arus utama efisiensi, tetapi disajikan sebagai kesimpulan utama tentang kelayakannya. Pada perangkat seluler, kepuasan pengguna dan penerimaan pasar ditentukan oleh masa pakai baterai. Biaya energi merupakan komponen signifikan dari biaya kepemilikan keseluruhan (TCO) di HPC dan pusat data, yang mungkin lebih besar daripada biaya pembelian perangkat keras. Hal ini diperparah oleh masalah lingkungan, menurut proyek pusat data, yang diperkirakan akan menggunakan hingga 8% dari listrik global pada tahun 2030 kecuali strategi konservasi energi diterapkan secara drastis.

Hal ini telah menyebabkan penggunaan istilah yang disebut silikon gelap, yang berarti area chip yang harus tetap tidak aktif karena keterbatasan daya, bahkan ketika secara teoritis dapat beroperasi. Teknik pengurangan daya yang lebih tradisional, termasuk clock gating, *multi-threshold CMOS* (MTCMOS), *dynamic voltage and frequency scaling* (DVFS), dan power gating, sebenarnya telah menjadi inti dari generasi desain IC sebelumnya.

Namun, metode-metode ini telah mengalami penurunan efektivitas seiring dengan semakin heterogennya sistem itu sendiri, dan beban kerja menjadi semakin tidak terduga. Pergeseran ke komputasi di mana-mana (IoT, perangkat yang dapat dikenakan, perangkat edge) dan kecerdasan edge-of-computing (akselerator AI/ML yang tertanam dalam SoC) membutuhkan metode desain dan sistem baru yang memungkinkan optimasi dinamis atas berbagai tujuan seperti daya rendah, kinerja tinggi, keandalan, dan efektivitas biaya. Metode desain hemat daya tradisional—pengaturan clock, pengaturan daya, dan penskalaan tegangan—telah memainkan peran penting dalam meminimalkan jumlah energi yang hilang. Namun, batasan strategi ini semakin terbebani seiring dengan semakin rumitnya jumlah rangkaian dan semakin dinamisnya permintaan akan sistem.

Metode klasik sangat heuristik dan berbasis aturan, serta didasarkan pada eksplorasi manual ruang desain dan optimasi deterministik. Menyeimbangkan beberapa tujuan desain yang mungkin saling bertentangan, termasuk mengurangi energi, meningkatkan throughput, dan keandalan secara bersamaan, merupakan tantangan. Dalam konteks inilah Kecerdasan Buatan (AI) dan Pembelajaran Mesin (ML) menjadi paradigma yang disruptif. Dengan kemampuan untuk memasukkan kecerdasan berbasis data ke dalam proses desain, AI dan ML memberikan peluang untuk mengelola kompleksitas, menemukan pola, dan mengotomatiskan aktivitas optimasi. Alih-alih perancang yang dirancang manusia dan aturan desain heuristik tetap, teknik-teknik tersebut telah memungkinkan penggunaan algoritma adaptif, prediktif, dan terukur dalam proses otomatisasi desain elektronik (EDA). Metode AI/ML tersebut efektif dalam ruang optimasi multi-objektif berdimensi tinggi (berbeda dengan berbasis aturan). Metode ini mampu menemukan korelasi yang tidak dapat dilihat oleh perancang manusia, memprediksi solusi desain tanpa simulasi yang akan memakan waktu lama, dan merespons variasi proses dan beban kerja secara real-time.

Metode berbasis kecerdasan buatan sangat membantu karena desain VLSI kontemporer bersifat multi-dimensi. Sebagai contoh, kompromi dalam daya kebocoran dapat berdampak negatif pada kecepatan switching, dan kompromi dalam panjang kawat dapat menciptakan masalah termal atau keandalan baru. Algoritma ML dirancang dengan baik untuk bergerak di sekitar ruang optimasi multi-objektif tersebut. Algoritma ini akan mampu menggunakan kumpulan besar desain historis, memprediksi bagaimana parameter yang berbeda akan memengaruhi desain, dan mengusulkan peningkatan desain yang dapat menghasilkan kompromi yang mendekati optimal.

Selain itu, AI juga tidak terbatas pada satu langkah alur desain VLSI. Penggunaannya meluas ke pemodelan arsitektur tingkat tinggi, hingga implementasi dan pengujian fisik. Pembelajaran remedial (RemL), misalnya, telah dimanfaatkan secara efisien dalam program perencanaan tata letak dan penempatan, sedangkan penalaran mendalam telah dimanfaatkan

secara efektif dalam pemodelan arus bocor dan fenomena pewaktuan dengan presisi tinggi. Implikasi dari kemajuan ini adalah bahwa versi masa depan dari proses desain IC dapat diotomatisasi, secara radikal mempersingkat waktu pemasaran, dan juga mencapai efisiensi daya yang superior. Karena penskalaan CMOS telah memfasilitasi kemajuan signifikan dalam beberapa dekade terakhir, daya bocor, efek saluran pendek, dan kompleksitas desain telah menjadi masalah, yang menyebabkan adopsi solusi baru.

Bab ini memberikan analisis mendalam tentang bagaimana AI dan ML mengganggu desain VLSI daya rendah. Untuk menempatkan masalah ini dalam perspektif, kita mulai dengan diskusi tentang sumber-sumber disipasi daya dalam sirkuit digital secara sangat rinci. Kemudian, kita memeriksa trade-off antara daya dan kinerja area (PAP) yang dapat mewakili salah satu tantangan desain mendasar dan dapat memerlukan pembuatan trade-off yang sulit. Kemudian kita melihat metode desain hemat daya tradisional, keberhasilannya, dan kelemahannya. Berdasarkan hal ini, kami menyajikan praktik AI dan ML dalam desain VLSI, mengevaluasi pendekatan ini dalam keseluruhan alur desain, dan menganalisis proyek contoh industri dunia nyata.

Terakhir, kami mengomentari hambatan yang menghalangi adopsi yang luas, dan kami menyajikan arah pembahasan di masa depan yang mencakup desain neuromorfik dan pembelajaran federasi dalam EDA.

## 4.2 DASAR-DASAR KONSUMSI DAYA DALAM VLSI

Desain VLSI tidak dapat dirancang dengan efisiensi energi yang berarti. Pengetahuan tentang konsumsi daya sirkuit digital. Disipasi daya tidak hanya memengaruhi masa pakai baterai sistem portabel tetapi juga memengaruhi keandalan chip, stabilitas termal, dan biaya operasional sistem. Oleh karena itu, perancang harus menganalisis berbagai jenis konsumsi daya dan karakteristiknya. Disipasi daya keseluruhan. Pada sirkuit berbasis CMOS, yang sekarang merupakan VLSI yang paling banyak digunakan, kinerja ditentukan oleh penskalaan oleh dua komponen sirkuit yang dikenal: HC dan VCE. Secara umum (Ptotal) dibagi menjadi empat kategori.

$$P_{total} = P_{dynamic} + P_{short-circuit} + P_{leakage} + P_{static}$$

Keputusan desain, teknologi fabrikasi, dan hal-hal lain berdampak pada kondisi operasi setiap komponen.

**Disipasi Daya Dinamis**

Sumber konsumsi utama pada sebagian besar sirkuit digital adalah aktivitas pengalihan daya dinamis. Ketika terjadi pengosongan dan pengisian daya, kapasitansi muncul. Hal ini terkait dengan gerbang logika dan konektor. Bentuk paling sederhana dari daya dinamis adalah sebagai berikut:

$$P_{dynamic} = \alpha \cdot C_L \cdot V_{dd}^2 \cdot f$$

di mana

$\alpha$ adalah faktor aktivitas switching (probabilitas transisi per siklus clock)

$C_L$ adalah kapasitansi beban

$V_{dd}$ adalah tegangan suplai

$f$ adalah frekuensi operasi

**Ketergantungan kuadratik pada tegangan** ($V_{dd}^2$) – menjadikan penskalaan tegangan sebagai kontrol paling efektif untuk mengurangi daya dinamis.

**Ketergantungan linier pada frekuensi** $(f)$ – mendorong skema DVFS yang mengurangi frekuensi clock pada aktivitas beban kerja rendah.

**Sensitivitas faktor aktivitas** $(\alpha)$ – harus menjalani optimasi tingkat arsitektur seperti clock gating dan isolasi operand.

Berdasarkan persamaan ini, jelas bahwa daya dinamis adalah hubungan kuadratik dari tegangan suplai dan bergantung secara linier pada frekuensi. Inilah alasan mengapa DVFS merupakan metode yang efektif untuk mengurangi energi. Namun, hal ini juga menyoroti keterbatasan inti: mengurangi tegangan dapat secara signifikan menurunkan kinerja atau memerlukan mekanisme koreksi kesalahan yang lebih kuat. Daya dinamis dalam FinFET SRAM muncul dari pengisian/pengosongan workline dan bitline yang sering terjadi selama akses memori.

**Dampak Penskalaan Teknologi**

Seiring dengan penyusutan transistor CMOS, kapasitansi per perangkat menurun, tetapi integrasi miliaran transistor menyebabkan ledakan daya switching agregat. Pada SoC modern, kapasitansi interkoneksi lebih berkontribusi pada daya dinamis daripada kapasitansi gerbang transistor, karena kabel panjang pada lapisan routing global.

**SRAM FinFET**

Daya dinamis pada SRAM berbasis FinFET muncul dari pengisian/pengosongan wordline dan bitline yang sering terjadi. Para peneliti menunjukkan bahwa pengoptimalan desain penguat sense dan driver wordline dapat mengurangi energi switching secara signifikan, faktor penting dalam SRAM biomedis berdaya rendah.

**Daya Hubung Singkat**

Daya hubung singkat terjadi selama transisi ketika transistor NMOS dan PMOS dalam gerbang CMOS secara bersamaan menghantarkan arus dalam waktu singkat. Hal ini menciptakan jalur langsung antara catu daya dan ground.

Besarnya arus hubung singkat bergantung pada:

- Waktu naik/turun input
- Tegangan suplai relatif terhadap tegangan ambang
- Kondisi beban output

Meskipun biasanya lebih kecil daripada daya dinamis, daya hubung singkat dapat mencapai 10%–20% dari total konsumsi pada sirkuit berkecepatan tinggi. Penentuan ukuran transistor dan kontrol tegangan yang cermat diperlukan untuk mengurangi dampaknya.

Kami menggunakan beberapa Strategi Mitigasi untuk daya hubung singkat, seperti penentuan ukuran transistor, untuk mengurangi tumpang tindih periode konduksi. Kontrol laju

perubahan tegangan (*slew rate*) untuk meminimalkan konduksi simultan. Bias bodi adaptif pada FinFET untuk mengurangi tumpang tindih hubung singkat.

**Daya Bocor**

Seiring perkembangan teknologi ke rezim nanometer, daya bocor telah menjadi perhatian utama. Bahkan ketika transistor "mati," arus bocor kecil mengalir, menyebabkan kehilangan energi statis. Daya bocor, terutama dalam aplikasi biomedis dan IoT yang membutuhkan operasi "selalu aktif," sangat penting. Struktur FinFET mengurangi kebocoran dibandingkan dengan CMOS planar tetapi tetap rentan terhadap kebocoran gerbang dan penurunan penghalang yang diinduksi oleh drain (DIBL).

Beberapa jenis arus kebocoran ada:

- **Kebocoran subambang** – karena konduksi inversi lemah di bawah tegangan ambang.
- **Kebocoran terowongan oksida gerbang** – disebabkan oleh terowongan kuantum melalui oksida gerbang ultra tipis.
- **Kebocoran sambungan** – timbul dari sambungan dioda yang diberi bias terbalik.

Ekspresi untuk arus kebocoran subambang (Isub) sering diberikan sebagai:

$$I_{sub} \approx I_0 e^{\frac{(V_{gs}-V_{th})}{nV_T}} \left(1 - e^{\frac{-V_{ds}}{V_T}}\right)$$

di mana $I_0$ adalah konstanta yang bergantung pada proses, $V_T$ adalah tegangan termal, dan n adalah faktor kemiringan subambang.

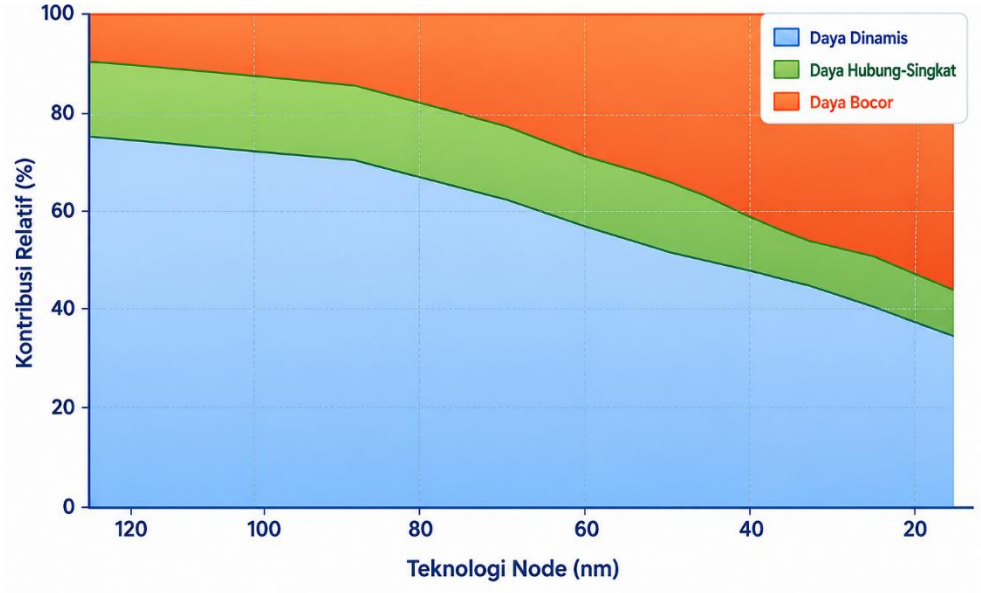
Kebocoran daya telah menjadi sangat penting seiring dengan perpindahan dari node 90 nm ke 7 nm. Pada beberapa prosesor mobile canggih, kebocoran daya mencapai hingga 40% dari total daya chip dalam kondisi siaga. Pada 7 nm dan di bawahnya, kebocoran daya dapat mencapai hingga 40% dari daya siaga pada prosesor mobile. Untuk node IoT yang selalu aktif, kebocoran daya mendominasi total anggaran energi, sehingga memerlukan inovasi seperti power gating, sel multi-threshold, dan topologi FinFET.

**Daya Statis**

Daya statis dikonsumsi ketika suatu sirkuit dalam keadaan idle tetapi tidak sepenuhnya dimatikan. Ini termasuk kebocoran daya, arus latar belakang, dan operasi penyegaran dalam memori. Meskipun kecil pada teknologi lama, sekarang ini signifikan pada node ultra-skala, terutama pada array SRAM besar yang mendominasi area SoC. Daya statis mengacu pada daya yang dikonsumsi ketika sirkuit dalam keadaan idle, tanpa switching aktif. Ini termasuk daya kebocoran tetapi juga dapat mencakup arus bias, operasi penyegaran memori, dan aktivitas latar belakang lainnya. Meskipun daya statis relatif kecil dibandingkan daya dinamis pada node yang lebih lama, daya statis semakin signifikan pada proses ultra-skala. Analisis daya hubung singkat pada SRAM 7T menunjukkan pengurangan yang nyata karena ukuran transistor dan topologi sel yang dioptimalkan.

**Analisis komparatif sumber daya**

Kontribusi relatif daya dinamis, hubung singkat, dan daya bocor sangat bergantung pada node teknologi dan lingkungan operasi, seperti yang ditunjukkan pada Gambar 4.1.




**Gambar 4.1** Kontribusi relatif sumber daya di berbagai node teknologi.

- Pada 130 nm dan di atasnya, daya dinamis merupakan komponen dominan (>70%).
- Pada 45 nm dan di bawahnya, daya bocor menjadi sebanding dengan daya dinamis.
- Pada 7 nm dan seterusnya, daya dinamis dan daya bocor sama-sama penting, dengan teknologi pengemasan dan pendinginan canggih yang diperlukan untuk mempertahankan operasi.

**Implikasi untuk Desain Daya Rendah**

Pengetahuan tentang sumber daya konsumsi tersebut meletakkan dasar untuk metodologi desain. Setiap upaya untuk mengurangi total konsumsi energi harus mempertimbangkan pengurangan daya dinamis menggunakan penskalaan tegangan yang mengurangi daya dinamis tetapi dengan mengorbankan kinerja. Transisi cepat dan penskalaan transistor untuk pengurangan hubung singkat, Power gating, dan pengurangan kebocoran dengan menggunakan MTCMOS, Manajemen pada tingkat sistem, seperti DVFS, prediksi beban kerja, dan clock gating.

Meskipun demikian, diamati bahwa pada sirkuit yang lebih kompleks, hanya solusi berbasis aturan yang kuat atau deterministik yang dapat diterapkan. Karena kesulitan dalam menemukan trade-off terbaik, strategi trade-off terbaik dapat dieksplorasi secara manual; Namun, pada kompleksitas SoC saat ini, hal ini tidak memungkinkan. Di sinilah AI dan ML berperan, menyediakan cara cerdas untuk memodelkan dan mengoptimalkan komponen daya yang saling berinteraksi di seluruh hierarki desain. Hal ini mensyaratkan optimasi berdasarkan AI/ML, yang dapat mensimulasikan interaksi antara berbagai sumber daya ini dan memberikan solusi trade-off yang cerdas.

## 4.3 TRADE-OFF DAYA, LUAS, DAN KINERJA (PAP)

Desain IC tidak pernah hanya berfokus pada optimasi satu ukuran saja. Sebaliknya, ini adalah trade-off antara tujuan yang saling bertentangan, terutama konsumsi daya dan pemanfaatan luas, dan kinerja. Salah satunya adalah trade-off PAP, yang juga dapat disebut sebagai salah satu masalah paling mendasar dalam desain VLSI dimensi yang berinteraksi untuk memberikan perspektif mengapa pendekatan tradisional gagal dan mengapa strategi berbasis AI/ML menjadi sangat diperlukan.

**Mendefinisikan Metrik**

Daya (P): Mewakili konsumsi energi per satuan waktu. Meliputi komponen dinamis, korsleting, kebocoran, dan statis seperti yang dibahas sebelumnya. Penting untuk perangkat portabel (masa pakai baterai), chip berdensitas tinggi (manajemen termal), dan komputasi berkelanjutan (biaya energi). Daya ultra-rendah sangat penting untuk implan biomedis dan perangkat edge IoT.

Daya mencerminkan laju konsumsi energi dalam suatu rangkaian, yang dinyatakan sebagai:

$$P = E/T$$

di mana E adalah energi yang dikonsumsi dan t adalah waktu.

**Signifikansi Praktis**

Untuk SoC seluler, daya secara langsung menentukan masa pakai baterai. Untuk server, daya menentukan biaya operasional dan anggaran termal. Untuk implan biomedis, daya ultra-rendah merupakan kendala yang sangat penting.

**Area (A):** Area mengacu pada luas fisik silikon yang ditempati oleh desain, diukur dalam mm2. Area yang lebih besar meningkatkan probabilitas cacat, menurunkan hasil produksi, dan meningkatkan biaya per chip. Pada SoC modern, cache dan buffer SRAM mengkonsumsi hingga 60%–70% dari luas die, sehingga topologi SRAM menjadi sangat penting. Area yang lebih besar memperburuk panjang kawat, parasit, dan secara tidak langsung memengaruhi daya. Luas silikon yang dibutuhkan untuk mengimplementasikan suatu desain. Memengaruhi hasil produksi dan biaya manufaktur, area yang lebih kecil biasanya menurunkan biaya produksi. Mempengaruhi kemacetan routing, panjang interkoneksi, dan, secara tidak langsung, daya. SRAM mendominasi area SoC, sehingga topologi sel memori yang efisien (6T, 7T, 8T) secara langsung memengaruhi kepadatan chip.

**Kinerja (Perf):** Diukur dalam hal frekuensi clock, throughput, latensi, atau instruksi per siklus (IPC). Kinerja yang lebih tinggi biasanya membutuhkan pipeline yang lebih dalam, jalur data yang lebih lebar, atau frekuensi operasi yang lebih tinggi, yang semuanya dapat meningkatkan daya atau area. Kecepatan operasi baca/tulis tidak boleh dikorbankan, terutama pada akselerator AI real-time. Kinerja didefinisikan oleh throughput komputasi yang dapat dicapai dalam anggaran daya dan area. Ini dapat diukur dengan: Frekuensi clock (f max), IPC, Latensi (waktu untuk menyelesaikan tugas), Throughput

(tugas per detik), dan contoh dunia nyata adalah CPU menekankan kinerja single-thread (sensitif terhadap latensi), GPU memprioritaskan throughput paralel (sensitif terhadap bandwidth), node IoT sering mengorbankan kinerja untuk efisiensi daya dan area.

**Sifat dari Trade-Off**

Mengoptimalkan satu metrik seringkali mengorbankan metrik lainnya:

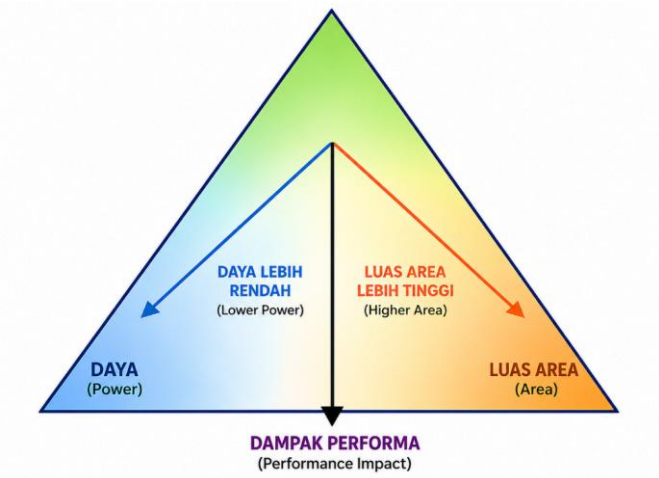
**Mengurangi daya** mungkin memerlukan penurunan tegangan suplai (Vdd), tetapi ini mengurangi frekuensi maksimum, sehingga menurunkan kinerja. Meningkatkan kinerja dengan menambahkan tahapan pipeline atau unit paralel biasanya meningkatkan luas area dan aktivitas switching, yang menyebabkan daya dinamis yang lebih tinggi. Minimisasi dengan pengurangan luas area dengan meminimalkan transistor atau struktur memori dapat merugikan atau menambah kinerja (cache terbatas) atau menambah daya (lebih banyak akses di luar chip). Eksperimen SRAM FinFET 7T menunjukkan bahwa peningkatan minimal pada luas area (transistor tambahan lokal) dapat memberikan penghematan yang signifikan dalam daya statis, yang mengarah pada efisiensi energi keseluruhan yang lebih baik per operasi. Ini menggambarkan segitiga PAP, di mana mengoptimalkan satu sudut (mengurangi daya) biasanya memerlukan trade-off dengan parameter lain (misalnya, peningkatan luas area). Oleh karena itu, perancang sirkuit dituntut untuk melakukan kompromi dengan masalah optimasi multi-objektif, di mana tidak ada solusi absolut yang dianggap terbaik, yang bergantung pada tujuan desain dan area kerja.

**Representasi Grafis dari Kompromi**

Ketiga dimensi tersebut dapat digambarkan dalam segitiga kompromi (Gambar 4.2).

Setiap titik sudut mewakili optimasi satu metrik dengan mengorbankan metrik lainnya. Bagian dalamnya mewakili titik desain yang layak. Titik optimal bergantung pada aplikasi:

- Perangkat seluler → prioritaskan *daya*.
- Pusat data → seimbangkan *kinerja* dan *daya*.
- Elektronik konsumen → tekankan *luas* (biaya).




**Gambar 4.2** Segitiga kompromi daya-luas-kinerja (PAP).

Kompromi PAP dapat digambarkan dalam bentuk segitiga di mana: Setiap sudutnya sesuai dengan memaksimalkan satu metrik (Daya, Luas, atau Kinerja). Desain yang sebenarnya terletak pada keberadaan bagian dalamnya. Optimasi spesifik aplikasi bergantung pada aplikasinya.

## 4.4 CONTOH SKENARIO KOMPROMI SPESIFIK INDUSTRI

**Smartphone:** Kapasitas baterai terbatas (Kapasitas baterai ≈ 4000–6000 mAh). Produsen biasanya memilih untuk memiliki kinerja yang kurang optimal, selama itu memungkinkan masa pakai baterai yang lebih lama. Di sini, daya telah menjadi hegemoni kompromi; kompromi telah mempromosikan efisiensi daya daripada kinerja absolut. Contoh – Arsitektur big.LITTLE ARM secara dinamis beralih antara inti daya tinggi dan daya rendah.

**Server Kinerja Tinggi:** Penyedia cloud menekankan pada throughput, mereka bersedia menerima area yang lebih besar (chip mahal) selama kinerja per watt tetap menguntungkan. Baik performa maupun daya lebih penting, sedangkan luas area kurang kritis. Performa per watt adalah kuncinya. Pusat data menerima chip area besar jika menghasilkan throughput yang lebih tinggi dengan watt per operasi yang lebih rendah. Contoh – akselerator TPU Google, dioptimalkan untuk efisiensi perkalian matriks.

**Perangkat IoT:** Perangkat seperti sensor atau tag pintar memiliki anggaran area dan daya yang ketat. Performa seringkali menjadi hal sekunder, karena tugas mungkin melibatkan akuisisi dan komunikasi data sederhana. Daya dan area mendominasi → mikrokontroler kecil, seringkali di bawah 50 MHz. Performa dapat diterima selama konektivitas dan beban kerja penginderaan didukung. Contoh: SRAM 7T daya ultra rendah untuk sensor biomedis.

**Akselerator AI (GPU/TPU):** Performa diprioritaskan tetapi dalam anggaran daya. GPU NVIDIA menggunakan eksplorasi ruang desain berbasis ML untuk mengidentifikasi arsitektur mikro yang memaksimalkan perf/W.

**Perumusan Analitis dari Trade-Off**

Para perancang sering mencoba untuk mengkuantifikasi trade-off ini. Misalnya, produk energi-tunda (EDP) dan produk energi-tunda-luas (EDAP) adalah metrik komposit yang banyak digunakan:

$$EDP = Energy \times Delay$$
$$EDAP = Energy \times Delay \times Area$$

Meminimalkan EDP mendukung desain dengan konsumsi daya rendah dan kinerja tinggi.

Meminimalkan EDAP lebih lanjut memperhitungkan biaya silikon.

Formulasi ini berguna dalam mengevaluasi keputusan trade-off secara kuantitatif, terutama dalam pembahasan yang membandingkan teknik hemat daya baru.

**Pendekatan tradisional untuk trade-off PAP**

Secara historis, perancang telah mengandalkan:

**Eksplorasi manual:** Insinyur menyesuaikan parameter seperti ukuran transistor, kedalaman pipeline, dan ukuran cache.

**Algoritma heuristik:** Metode berbasis aturan yang tertanam dalam alat EDA (misalnya, sintesis berbasis waktu dan penempatan dan perutean berbasis area).

**Analisis batas Pareto:** Mengidentifikasi solusi non-dominan di mana tidak ada metrik yang dapat ditingkatkan tanpa menurunkan metrik lainnya.

Meskipun metode ini berfungsi untuk ukuran desain yang moderat, metode ini menjadi semakin tidak praktis untuk SoC modern, yang berisi miliaran transistor dan komponen yang sangat heterogen.

**Keterbatasan optimasi konvensional**

Pertukaran PAP dalam teknologi canggih menghadirkan beberapa tantangan:

- **Ruang Desain yang Meledak**: Dengan banyak pulau tegangan, domain clock, dan integrasi 3D, ruang desain tumbuh secara eksponensial. Penyetelan manual tidak dapat mengimbanginya.
- **Ketergantungan Kompleks**: PAP tidak independen. Misalnya, mengurangi panjang interkoneksi dapat menurunkan daya dinamis tetapi meningkatkan kemacetan routing, yang meningkatkan kebocoran.
- **Kondisi Operasi Dinamis**: Chip berfungsi pada beban kerja dan suhu yang berbeda. Optimasi statis mungkin tidak sesuai dengan fakta kasus.
- **Skalabilitas Teknologi**: Variabilitas pada node canggih (misalnya, 5 nm, 3 nm) membuat prediksi kebocoran dan timing sangat tidak pasti.

Kekurangan inilah yang mendorong penggunaan metode AI dan ML, yang mampu mensimulasikan ketergantungan hibrida dan merespons lingkungan.

**Peran AI/ML yang Muncul dalam Penanganan Kompromi**

Optimasi PAP dilengkapi dengan dua keunggulan oleh metode AI dan ML:

**Pemodelan Prediktif**: Model ML mampu memproyeksikan ukuran daya dan kinerja pada tahap awal desain, mengurangi banyak simulasi yang mahal. Sebagai contoh, daya kebocoran di sudut-sudut teknologi diprediksi oleh model regresi.

**Eksplorasi Ruang Desain (DSE):** Pertukaran (trade-off) dapat dicari oleh agen pembelajaran penguatan melalui eksperimen dengan ide desain dan pembelajaran dari hasilnya. Jaringan saraf mampu memodelkan batas Pareto dengan akurat. Dengan cara ini, masalah pertukaran yang dulunya ditangani melalui coba-coba harus dilakukan oleh masalah optimasi AI/ML berdasarkan data.

## 4.5 PARADIGMA AI DAN ML DALAM OPTIMASI DAYA VLSI

Integrasi AI dan ML ke dalam desain VLSI, proses perancangan Sirkuit Skala Sangat Besar (VLSI), mengubah cara para insinyur bekerja menuju optimasi daya rendah. Praktik

konvensional bergantung pada heuristik deterministik berbasis aturan untuk melakukan pertukaran kinerja, luas area, dan konsumsi daya. Metode tersebut mungkin gagal pada skala desain yang lebih besar dan tidak berhasil pada skala desain yang lebih kecil ketika menangani ruang pencarian yang besar dan multidimensi yang merupakan bagian dari teknologi semikonduktor modern.

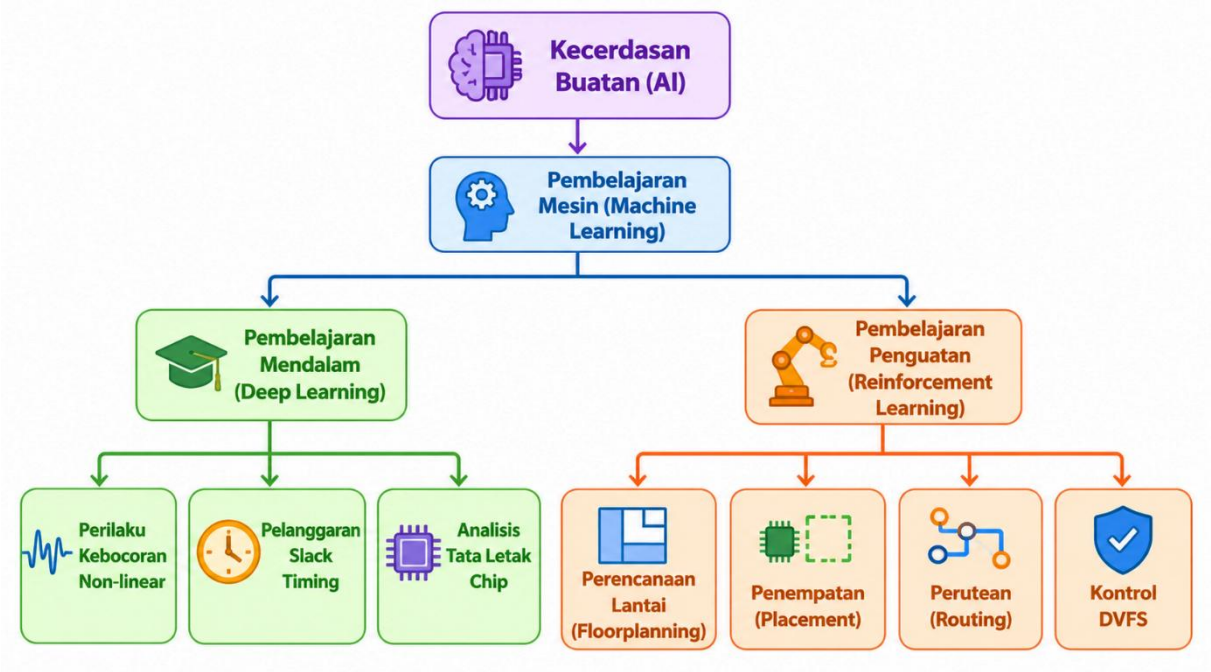
Sebaliknya, AI dan ML menghadirkan metode berbasis data adaptif yang memiliki kemampuan untuk membangun asosiasi tersembunyi, memprediksi desain, perilaku, dan optimasi trade-off secara real-time. Pengenalan AI dan ML ke dalam VLSI telah mengubah cara konvensional dalam mendesain perangkat berdaya rendah. Secara tradisional, proses desain VLSI bersifat heuristik dan berbasis aturan, menekankan penggunaan metode deterministik dan pencarian ruang desain secara manual.

Metode ini bekerja cukup baik pada skala menengah tetapi tidak bekerja dengan baik pada teknologi nanometer di mana ruang desainnya sangat besar secara eksponensial, efek antar parameter desain sangat non-linier, dan keterbatasan daya. Tidak seperti AI dan ML, AI dan ML secara adaptif mengoptimalkan pola berdasarkan data multi-tujuan. Kemampuan untuk menemukan asosiasi laten dan kemampuan generalisasi ini sekarang diperlukan untuk desain otomatisasi VLSI, terutama untuk IC yang kekurangan daya.

**Membedakan AI, ML, Pembelajaran Mendalam, dan Pembelajaran Penguatan**

Pertama-tama, perlu dibedakan antara berbagai paradigma kecerdasan komputasional dan fungsi-fungsi khasnya dalam optimasi VLSI seperti yang ditunjukkan pada Gambar 4.3.

**Kecerdasan Buatan (AI):** Bidang umum yang berkaitan dengan mereplikasi kecerdasan seperti manusia pada mesin. AI menawarkan struktur dalam kasus VLSI. AI diterapkan dalam memecahkan masalah, merepresentasikan pengetahuan, dan mempelajari penalaran otomatis, disiplin payung yang terlibat dalam meniru kecerdasan pada mesin yang disamakan dengan kecerdasan manusia. Dalam VLSI, AI memberikan dasar konseptual untuk penalaran otomatis, pengambilan keputusan alur desain, dan pemecahan masalah simbolik. Misalnya, pada awal tahun 1980-an, alat sintesis logika berbasis aturan berdasarkan AI dibuat.




**Gambar 4.3** Model ML untuk tugas VLSI.

**Pembelajaran Mesin (ML):** Salah satu arah AI berkaitan dengan algoritma yang kinerjanya meningkat seiring dengan paparan data yang lebih banyak. Teknik ML mampu mempelajari perilaku sirkuit berdasarkan desain dan data masa lalu, serta memiliki kemampuan prediksi dan optimasi. Penggunaan ML dalam desain HDL dan VLSI mencakup pemodelan prediktif (misalnya, estimasi daya kebocoran) dengan parameter transistor, pengenalan pola (misalnya, jaringan kritis yang kemungkinan menyebabkan) dan eksplorasi ruang desain. Berbeda dengan algoritma heuristik, ML dapat disesuaikan dan belajar, sehingga menghilangkan ketergantungan pada coba-coba manusia.

**Pembelajaran Mendalam (DL):** Ini adalah subbidang ML berdasarkan jaringan saraf multi-lapisan (ANN). DL dikenal berkinerja baik dalam menyediakan fitur berdimensi tinggi dalam kumpulan data yang besar dan kompleks, memungkinkan ruang desain dimodelkan secara rinci dan memprediksi kinerja sirkuit. Teknik DL dapat digunakan untuk menangkap perilaku kebocoran non-linier pada perangkat skala nano dalam desain VLSI dan untuk memprediksi kapan pelanggaran slack akan terjadi pada tata letak yang kompleks. Menganalisis tata letak chip sebagai gambar menggunakan jaringan saraf konvolusional (CNN) untuk optimasi penempatan dan perutean.

**Pembelajaran Penguatan (*Reinforcement Learning*/RL):** Sebuah paradigma ML di mana agen belajar dengan berinteraksi dengan lingkungan dan menerima umpan balik dalam bentuk hadiah. RL sangat cocok untuk proses pengambilan keputusan berurutan seperti perencanaan tata letak (floorplanning), penempatan, dan perutean serta kontrol DVFS. Aplikasi penting Google tentang RL untuk perencanaan tata letak menunjukkan bahwa agen AI dapat mengungguli pakar manusia dalam optimasi tata letak chip, memangkas waktu desain dari berbulan-bulan menjadi beberapa jam.

**Taksonomi Teknik ML dalam VLSI**

Teknik ML yang berbeda memiliki peran unik dalam alur desain VLSI:

**Pembelajaran Terawasi**: Berguna untuk tugas prediksi daya dan kinerja. Model seperti regresi, pohon keputusan, dan mesin vektor pendukung dapat belajar dari kumpulan data berlabel dari desain masa lalu untuk memprediksi hasil untuk sirkuit baru. Untuk pemodelan daya di FinFET SRAM, model regresi memprediksi daya kebocoran berdasarkan parameter perangkat. Model pembelajaran terawasi dilatih pada kumpulan data berlabel. Dalam VLSI, contohnya meliputi: model regresi yang memprediksi daya kebocoran berdasarkan ukuran gerbang, tegangan ambang, dan suhu. Pohon keputusan yang memperkirakan dampak ukuran transistor pada daya dinamis dan hubung singkat. Mesin Vektor Pendukung (SVM) yang mengklasifikasikan sel standar dengan kebocoran tinggi vs. kebocoran rendah. Metode terawasi sangat ampuh dalam prediksi daya/kinerja tahap awal, di mana simulasi menyeluruh tidak praktis.

**Pembelajaran Tak Terawasi:** Efektif untuk pengelompokan dan deteksi anomali. Misalnya, outlier daya kebocoran dalam desain sub-10 nm dapat diidentifikasi tanpa kumpulan data berlabel. Untuk pengelompokan desain dan mengidentifikasi kondisi kebocoran tipikal di seluruh ruang desain yang besar. Metode tanpa pengawasan menemukan struktur tersembunyi dalam data yang tidak berlabel. Aplikasinya meliputi: pengelompokan varian desain untuk mengidentifikasi trade-off Pareto-optimal dalam ruang PAP. Deteksi anomali untuk menemukan pola kebocoran yang tidak biasa atau hotspot penurunan IR. Pengurangan pustaka desain, di mana sel standar yang berlebihan atau tidak efisien secara otomatis dihilangkan.

**Pohon Keputusan dan Hutan Acak**: Dapat memberikan model yang mudah dipahami untuk mengevaluasi trade-off di antara kendala desain. Mereka memungkinkan para insinyur untuk memahami bagaimana variasi parametrik desain memengaruhi konsumsi daya.

**Mesin Vektor Pendukung (SVM):** Ini sangat cocok dalam klasifikasi, seperti dalam mengidentifikasi pustaka sel dengan kebocoran tinggi dan rendah saat melakukan pemilihan sel.

**Jaringan Saraf Konvolusional (CNN):** CNN sangat efektif dalam kasus masalah yang terstruktur secara spasial, seperti penempatan dan perutean, di mana data tata letak menyerupai gambar dengan fitur dan pola.

**Pembelajaran Penguatan (RL):** Metode ini digunakan dalam tugas desain iteratif yang membutuhkan aplikasi yang berjalan di garis tipis antara berbagai tujuan yang berlawanan. Contoh terbaik adalah sistem floorplanning berbasis RL milik Google, yang mengurangi waktu layout chip dari beberapa minggu menjadi beberapa jam.

**Keunggulan AI/ML Dibandingkan Heuristik Tradisional**

**Skalabilitas**: AI/ML mampu bekerja dengan ruang desain yang sangat besar dalam teknologi nanometer, yang mampu mengeksplorasi teknologi yang kompleks yang

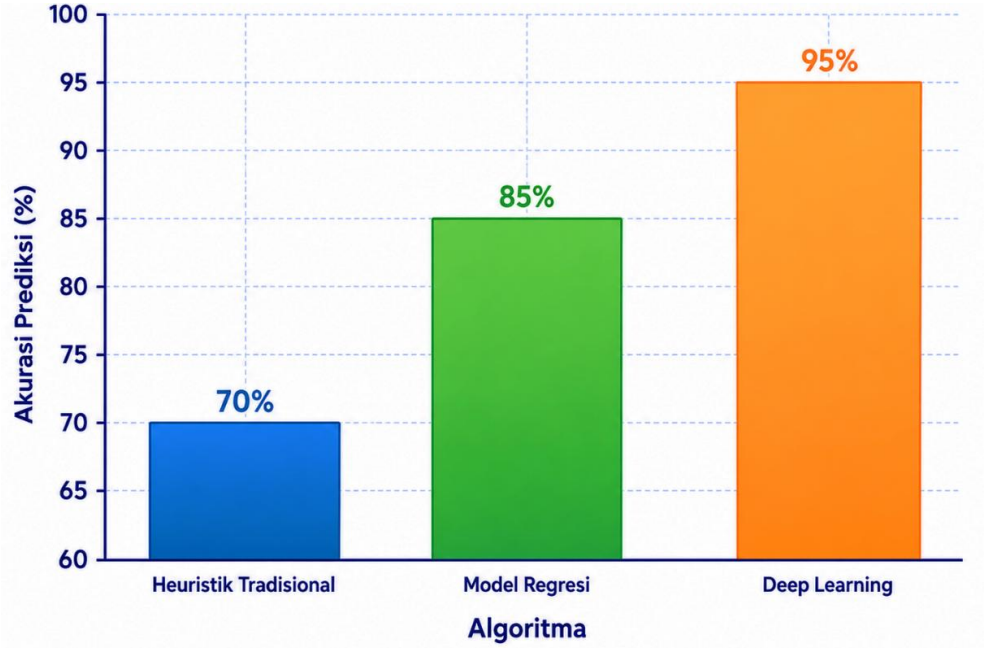
gagal dilakukan oleh heuristik dalam ruang desain yang sangat besar. Contoh - floorplanning RL mencari miliaran konfigurasi dengan cepat.

**Otomatisasi**: Banyak langkah dalam alur desain dapat diotomatisasi, mengurangi ketergantungan pada intervensi manual dan mempersingkat waktu pemasaran. Mengurangi ketergantungan pada penyetelan parameter manual. Mempersingkat waktu pemasaran, yang sangat penting dalam daya saing semikonduktor. Alur desain memiliki sejumlah langkah yang dapat diotomatisasi, yang membuatnya semakin sedikit bergantung pada intervensi manual dan mengurangi waktu pemasaran. Mengurangi penggunaan penyetelan parameter manual. Mengurangi waktu pemasaran, yang sangat penting untuk daya saing semikonduktor.

**Pemodelan Prediktif**: Alih-alih mensimulasikan setiap konfigurasi yang mungkin, model ML dapat memprediksi kinerja sirkuit dengan akurasi tinggi. Model ML pengganti mendekati simulasi SPICE yang mahal. Model regresi memprediksi kebocoran dalam margin kesalahan 5%–10% seperti yang diprediksi pada Gambar 4.4. Model ML tidak harus mensimulasikan semua konfigurasi yang mungkin, tetapi hanya beberapa saja yang dapat membuat prediksi akurat tentang kinerja sirkuit. Dalam model ML pengganti, kebocoran diperkirakan menggunakan model regresi dengan margin kesalahan 5%–10%.

**Kemampuan Adaptasi**: Teknik AI/ML dapat beradaptasi secara dinamis terhadap kendala yang terus berkembang, seperti perubahan pada node teknologi atau persyaratan sistem. AI/ML beradaptasi dengan node teknologi, beban kerja, dan kendala baru tanpa perlu menulis ulang heuristik. Metode AI/ML dapat menyesuaikan diri secara dinamis terhadap kendala yang berubah, termasuk modifikasi node teknologi atau persyaratan sistem. AI/ML menyesuaikan diri dengan hal-hal baru tanpa perlu menulis ulang heuristik, node teknologi, beban, dan kendala.

**Generalisasi Lintas Desain**: Dengan data pelatihan yang memadai, model AI/ML dapat menggeneralisasi wawasan dari satu proyek desain ke proyek desain lainnya, sebuah fitur yang tidak dapat ditawarkan oleh pendekatan heuristik. CNN yang dilatih pada satu set floorplan dapat menggeneralisasi ke desain yang belum pernah dilihat sebelumnya, sehingga mengurangi waktu siklus desain. Model AI/ML mampu, dengan data pelatihan yang cukup, mengekstrapolasi pengetahuan yang diperoleh dari satu proyek desain ke proyek desain lainnya, yang tidak dapat dilakukan oleh heuristik, dan tidak dapat digeneralisasi ke CNN yang dilatih pada set floorplan yang berbeda, sehingga mempersingkat waktu siklus desain.




**Gambar 4.4** Perbandingan akurasi prediksi untuk estimasi daya.

**Titik Integrasi di Seluruh Alur Desain VLSI**

**Pemodelan Tingkat Sistem**: Model daya berbasis ML dapat dengan cepat memperkirakan konsumsi, memandu pilihan arsitektur tingkat tinggi seperti desain hierarki memori atau strategi integrasi SoC. Model daya ML cepat dalam memperkirakan konsumsi yang memengaruhi keputusan desain tingkat atas seperti desain hierarki memori atau strategi integrasi SoC.

**Sintesis Logika**: AI dapat mengoptimalkan pemetaan teknologi untuk mengurangi aktivitas switching dan kedalaman logika, sehingga meminimalkan daya dinamis. SVM memandu pemetaan teknologi dengan memprediksi kebocoran/kinerja pustaka sel. AI mampu mengoptimalkan pemetaan teknologi untuk meminimalkan aktivitas switching, kedalaman logika, dan pengurangan daya dinamis yang dihasilkan. SVM adalah pemetaan arah teknologi dengan memprediksi kebocoran/kinerja pustaka sel.

**Desain Fisik**: Penempatan, perencanaan tata letak, dan perutean mendapat manfaat dari RL dan CNN untuk meminimalkan panjang kabel, mengurangi titik panas, dan mendistribusikan daya secara merata di seluruh chip. Agen RL mengoptimalkan penempatan/perencanaan tata letak, CNN mendeteksi kemacetan perutean, ML memprediksi titik panas. RL dan CNN bermanfaat dalam desain fisik, yaitu penempatan, perencanaan tata letak, dan perutean untuk menjaga panjang kabel sependek mungkin, menghilangkan titik panas, dan menyebarkan kepadatan daya secara seragam di seluruh chip. Agen RL merencanakan/merencanakan tata letak, CNN menemukan kemacetan perutean, ML memprediksi titik panas.

**Pengujian dan Verifikasi**: Algoritma ML membantu mengidentifikasi anomali, memprediksi tingkat kegagalan, dan memperkirakan disipasi daya selama skenario pengujian, memastikan sirkuit yang kuat dan hemat energi. Pengelompokan tanpa pengawasan mendeteksi aktivitas peralihan anomali. Generasi pola uji berbasis ML

mengurangi daya puncak selama pengujian pemindaian. Algoritma ML digunakan untuk mendeteksi anomali dan memperkirakan tingkat kegagalan, dan menentukan kehilangan daya dalam kasus uji, sehingga menjamin kondisi sirkuit hemat energi yang baik dan sehat. Pengelompokan tanpa pengawasan mengidentifikasi aktivitas peralihan abnormal. Generasi pola uji berbasis ML meminimalkan daya puncak selama pengujian pemindaian.

**Tantangan dalam Mengadopsi Paradigma AI/ML**

Meskipun ini adalah bidang yang menjanjikan, terdapat beberapa kendala dalam penerapan AI/ML:

**Ketersediaan Data**: Terdapat keterbatasan dataset berlabel berkualitas tinggi karena alasan kepemilikan. Federated learning menjanjikan peluang di industri semikonduktor.

**Interpretabilitas**: Model DL juga dapat bertindak sebagai kotak hitam, sehingga sulit untuk menjelaskan keputusan desain yang dibuat oleh para insinyur. Explainable AI (XAI) dimaksudkan untuk mengatasi hal ini.

**Masalah Generalisasi**: Generalisasi setelah model dilatih pada satu node teknologi mungkin bukan transisi yang mudah untuk diidentifikasi dengan orang lain tanpa akomodasi. Sebagai contoh, model yang dilatih pada 28 nm mungkin tidak dapat diskalakan langsung ke node 5 nm. Transfer learning adalah solusi potensial.

**Overhead Komputasi**: Model yang kompleks dapat dilatih dan diterapkan, menimbulkan komputasi yang berat, yang mengimbangi kemungkinan penghematan energi. Membangun Deep Networks itu mahal, dan Light-weight ML dan TinyML sedang dipertimbangkan.

**Terhubung dengan alat CAD yang sudah ada**: Alur kerja saat ini (Synopsys, Cadence, Mentor) sangat bergantung pada heuristik. Desain alur kerja yang menggabungkan heuristik dan prediksi ML sedang muncul (Tabel 4.1).

**Tabel 4.1** Ringkasan paradigma AI/ML

| No. | Paradigma | Peran Utama dalam VLSI | Contoh Aplikasi |
|---|---|---|---|
| **1** | Pembelajaran Terawasi (*Supervised Learning*) | Pemodelan prediktif | Model regresi untuk memprediksi daya bocor (*leakage power*) |
| **2** | Pembelajaran Tak Terawasi (*Unsupervised Learning*) | Klasterisasi (*clustering*) dan deteksi anomali | Mengidentifikasi titik panas (*hotspots*) **IR-drop** |
| **3** | Pembelajaran Penguatan (*Reinforcement Learning*) | Optimasi berurutan (*sequential optimization*) | *Floorplanning* berbasis **Reinforcement Learning** yang dikembangkan Google |
| **4** | Pembelajaran Mendalam (*Deep Learning*) | Pemodelan berdimensi tinggi | *Convolutional Neural Networks* (**CNN**) untuk penempatan (*placement*) dan perutean (*routing*) |

| | | | |
|---|---|---|---|
| **5** | Pembelajaran Hibrida/Semi-Terawasi (*Hybrid/Semi-Supervised Learning*) | Lingkungan dengan data yang terbatas | *Transfer Learning* antar node teknologi |

**Aplikasi AI dan ML di Seluruh Alur Desain VLSI**

Alur desain VLSI adalah proses dalam banyak fase yang akan dimulai dari desain tingkat sistem bangunan hingga uji coba dan kesalahan chip tiruan. Setiap tahap menawarkan tantangan dan kemungkinan baru dalam optimalisasi efisiensi daya. Secara tradisional, para insinyur menggunakan pemodelan analitis, heuristik, dan simulasi ekstensif di setiap langkah. Namun demikian, dengan menggunakan AI dan ML, terdapat area prediksi baru, alur kerja dimodifikasi menjadi otomatisasi yang lebih cerdas, dan metodologi pengambilan keputusan yang cerdas, adaptif, terukur, dan hemat daya. Alur desain VLSI umumnya kompleks, dengan abstraksi multilevel, mulai dari pemodelan arsitektur tingkat sistem hingga pengujian pasca-silikon. Terdapat berbagai tahapan yang menghasilkan hal-hal berbeda, persaingan dalam optimasi daya, tujuan kinerja, dan kendala manufaktur. Secara konvensional, para insinyur menggunakan pemodelan analitik, pencarian heuristik, dan simulasi menyeluruh secara bertahap. Meskipun efektif di masa lalu dengan pendekatan ini, node teknologi saat ini telah menjadi SoC dengan miliaran transistor yang secara komputasi tidak layak. AI dan ML kini telah menjadi pendukung lintas sektor, menerapkan model prediktif, otomatisasi, dan optimasi adaptif pada kedua tahapan alur desain. Peran mereka dibahas secara sistematis di subbagian selanjutnya.

## 4.6 DESAIN TINGKAT SISTEM DAN EKSPLORASI ARSITEKTURAL

Pemodelan tingkat sistem adalah definisi arsitektur pada tingkat pertama desain, cetak biru Norris dari sebuah sirkuit terpadu. Ini adalah pilihan seperti mikroarsitektur, topologi interkoneksi, dan hierarki memori yang sangat berpengaruh pada perilaku daya hilir. Desain tingkat sistem adalah deskripsi rencana sebuah sirkuit terpadu. Pilihan seperti mikroarsitektur prosesor, hierarki cache, topologi interkoneksi, efisiensi daya hilir berdasarkan topologi, dan domain daya dipertimbangkan.

**Model Prediktif pada Kecerdasan Buatan**: Algoritma ML mampu memberikan perkiraan nilai PAP yang cepat dan tepat, memungkinkan pra-pemangkasan simulasi desain yang tidak lengkap. Model ML di bawah pengawasan memprediksi daya/kinerja yang diberikan sebagai parameter sistem seperti jumlah inti, ukuran cache, atau jumlah memori. Contoh: regresi bandwidth dan jaringan saraf. Jaringan saraf yang memprediksi daya pada tingkat sistem. Penggunaan kurang dari 5% kesalahan dibandingkan dengan simulasi akurat siklus.

**Eksplorasi Ruang Desain (DSE):** Agen pembelajaran penguatan mampu secara otonom mencari melalui ruang konfigurasi yang sangat besar, memilih parameter SoC (misalnya, ukuran cache dan kedalaman pipa) yang menyeimbangkan antara konsumsi energi dan laju komputasi.

Tingkat ruang desain ini mungkin sangat besar. Agen RL mencari pengaturan parameter tanpa intervensi manusia, dan mereka memangkas solusi yang tidak produktif. Google mempresentasikan RL dalam eksplorasi mikroarsitektur TPU, membagi ke dalam radius yang relatif kecil lebih lambat daripada insinyur manusia.

**Kasus Penggunaan**: Telah ditunjukkan bahwa jaringan saraf dalam dapat digunakan untuk memprediksi daya suatu sistem pada tingkat sistem dengan kesalahan kurang dari 5% relatif terhadap simulasi yang lebih detail, sehingga mengurangi eksplorasi hingga faktor 5. Model ML yang dilatih pada 500 varian SoC mengurangi waktu eksplorasi sebesar 80% sambil mengidentifikasi titik Pareto-optimal dalam ruang daya-kinerja-luas.

**Sintesis Logika dan Pemetaan Teknologi**

Sintesis logika mengubah deskripsi perilaku tingkat tinggi (misalnya, kode HDL) menjadi implementasi tingkat gerbang. Langkah ini sangat penting untuk mengurangi daya dinamis, yang sangat dipengaruhi oleh aktivitas peralihan dan kedalaman logika. Optimasi daya pada tahap ini sangat penting, karena aktivitas peralihan dan kedalaman logika mendorong konsumsi daya dinamis.

Sintesis logika bekerja dengan deskripsi perilaku tingkat tinggi (misalnya, kode HDL) dan menghasilkan kode untuk implementasi tingkat gerbang. Langkah ini penting dalam meminimalkan daya dinamis, yang sangat dipengaruhi oleh aktivitas peralihan dan kedalaman logika. Optimasi daya pada tahap ini sangat penting; peralihan aktivitas/kedalaman logika menimbulkan konsumsi daya dinamis.

**Prediksi Aktivitas Peralihan**: Klasifikasi ML mampu mendeteksi titik-titik panas di mana pengalihan sinyal terlalu banyak dan menyarankan teknik pemetaan ulang untuk meminimalkan transisi. ML mendeteksi sinyal dengan tingkat pengalihan tinggi oleh klasifikasi untuk mendorong strategi pemetaan ulang. Contoh – Model berbasis CNN yang menggambarkan netlist sebagai grafik untuk memprediksi titik-titik panas aktivitas.

**Optimasi Pemetaan Gerbang**: Pohon keputusan dan model regresi dapat mengusulkan sel pustaka optimal, yang mengurangi produk energi-tunda (EDP). Contoh Aplikasi: Sebuah contoh menggunakan pendekatan SVM diberikan untuk meningkatkan akurasi estimasi daya kebocoran, mengarahkan pilihan pustaka MTCMOS yang lebih baik.

**Praktik Industri**: Praktik industri berbasis ML dalam EDA sekarang tersedia di Cadence dan Synopsys, yang mengantisipasi kekuatan dan jadwal dalam sintesis logika, sangat mengurangi waktu eksekusi.

**Desain Fisik: Penempatan, Perencanaan Tata Letak, dan Perutean**

Salah satu tahap yang paling intensif secara komputasi adalah desain fisik di masa lalu. Ini melibatkan sistematisasi sel standar, makro, dan interkoneksi untuk memenuhi batasan waktu dan daya.

**Pembelajaran Penguatan untuk Perencanaan Tata Letak**: Sistem RL yang digunakan oleh Google dilatih pada desain Chip dengan ribuan desain chip, yang telah terbukti

menghasilkan hasil berkualitas untuk perencanaan tata letak per jam, bukan perencanaan tata letak mingguan pada Gambar 4.5.




**Gambar 4.5** Perbandingan waktu desain – tata letak tradisional vs. berbasis RL.

**CNN untuk Optimasi Penempatan**: CNN mengklasifikasikan gambar tata letak chip, dengan CNN kemacetan, panjang kawat, dan kepadatan daya dapat diprediksi, memungkinkan tindakan korektif preventif.

**Optimasi Perutean**: Heuristik perutean berbasis ML mampu memprediksi risiko elektromigrasi dan menemukan metode perutean yang meminimalkan penurunan IR, titik panas daya.

**Analisis Waktu, Estimasi Daya, dan Verifikasi**

Setelah tata letak desain, validasi desain dilakukan dengan estimasi daya dan analisis waktu statis (STA), untuk menentukan bagaimana desain mencapai tujuan kinerja dan energinya.

**Model Pengganti Estimasi Daya**: Alih-alih simulasi tingkat SPICE yang lambat, model DL dan regresi dapat digunakan untuk memberikan estimasi konsumsi daya yang mendekati waktu nyata dalam metode estimasi daya komersial.

**Lokalisasi Kesalahan**: Model ML memiliki kemampuan untuk mempelajari pola kegagalan simulasi atau log pengujian silikon, dengan cepat menemukan penyebab kebocoran atau pengalihan yang tidak diinginkan.

**Deteksi Anomali dalam Verifikasi**: Ini dapat dilakukan menggunakan pengelompokan tanpa pengawasan untuk menunjukkan perilaku sinyal yang aneh yang mengindikasikan ketidaksempurnaan desain.

**Pengujian Manufaktur dan Validasi Pasca-Silikon**

Biaya pengembangan chip mencakup aspek yang cukup besar dari pengujian dan validasi. Pengenalan AI dan ML dapat mendukung pengurangan biaya tersebut dan meningkatkan keandalan.

**Penyediaan Pengujian Hemat Daya**: Klasifikasi ML dapat memodifikasi pola pengujian untuk melakukannya secara adaptif, meminimalkan daya puncak, meminimalkan penurunan IR (*Induced Response*) dari pengujian pemindaian.

Pemeliharaan Prediktif: Sistem ML dapat dipasok dengan data pemantauan pasca-silikon untuk memprediksi indikator kuat dalam mendeteksi terjadinya kebocoran, gangguan, atau cacat lainnya dan mengantisipasi kegagalan prematur atau keandalan.

**Prediksi dan Optimasi Hasil**: Perkiraan kehilangan hasil regresi digunakan untuk memprediksi kehilangan hasil sebagai akibat dari variabilitas kebocoran FinFET. Tindakan korektif proaktif meningkatkan total hasil.

**Studi Kasus**: Model ML diimplementasikan untuk memprediksi kerugian dalam hal hasil produksi yang disebabkan oleh perbedaan kebocoran pada teknologi FinFET, yang memungkinkan respons perbaikan preventif (Tabel 4.2).

**Tabel 4.2** Ringkasan aplikasi di berbagai tahapan desain

| No. | Tahap Desain | Aplikasi AI/ML | Contoh Hasil |
|---|---|---|---|
| 1 | Desain Tingkat Sistem (*System-Level Design*) | Pemodelan prediktif daya dan kinerja, **Reinforcement Learning (RL)** untuk eksplorasi ruang desain (*Design Space Exploration*) | Tingkat kesalahan **< 5%**, eksplorasi desain **10× lebih cepat** |
| 2 | Sintesis Logika (*Logic Synthesis*) | Prediksi aktivitas pensaklaran (*switching activity*), pemetaan gerbang (*gate mapping*), dan estimasi daya bocor berbasis **Support Vector Machine (SVM)** | Estimasi daya bocor (*leakage power*) yang lebih akurat |
| 3 | Desain Fisik (*Physical Design*) | **Reinforcement Learning** untuk *floorplanning*, **Convolutional Neural Networks (CNN)** untuk *placement*, serta optimasi *routing* berbasis **Machine Learning (ML)** | Waktu desain berkurang dari **minggu menjadi jam**, serta penurunan konsumsi daya dinamis |
| 4 | Verifikasi dan Analisis (*Verification and Analysis*) | Model daya pengganti (*surrogate power models*), deteksi anomali, serta **Machine Learning** untuk analisis *timing* dan lokalisasi kesalahan (*fault localization*) | Tingkat kesalahan sekitar **8% dibandingkan SPICE**, serta peningkatan kemampuan isolasi kesalahan |
| 5 | Pengujian dan Pascasilikon (*Testing and Post-Silicon*) | Pemilihan pola pengujian berbasis kesadaran daya (*power-aware test pattern selection*), pemeliharaan prediktif (*predictive maintenance*), dan optimasi *yield* | Konsumsi daya pada **scan-test** berkurang sekitar **25%**, serta peningkatan *yield* produksi |

## 4.7 STUDI KASUS DAN APLIKASI INDUSTRI

Contoh terbaik adalah keberhasilan tinggi dalam pengenalan AI dan ML dalam desain VLSI berdaya rendah dengan menggunakan studi kasus aktual. Ini adalah contoh yang menggambarkan tidak hanya kelayakan teknis, tetapi juga kelayakan komersial dari aliran desain yang dihasilkan AI/ML. Mereka mengungkapkan fakta bahwa heuristik tradisional di industri dan di antara para peneliti telah dikesampingkan, sehingga model cerdas dapat menghasilkan chip yang lebih kecil, lebih cepat, lebih penting, dan jauh lebih hemat daya.

**Pembelajaran Penguatan Google untuk Perencanaan Tata Letak (Floorplanning)**

Aplikasi pembelajaran penguatan Google untuk perencanaan tata letak (floorplanning) mungkin dapat dianggap sebagai studi kasus yang paling terkenal seperti yang ditunjukkan oleh Google Research [6, 10]. Kemajuan luar biasa dalam riset Google telah menyediakan desain VLSI yang didukung AI, yang didasarkan pada penguatan masalah tradisional perencanaan tata letak chip dengan pembelajaran penguatan (RL). Perencanaan tata letak tradisional dapat memakan waktu berbulan-bulan dari waktu rekayasa profesional, menempatkan blok logika secara optimal, dengan mempertimbangkan panjang kabel, kepadatan daya, dan kinerja. Model RL yang dikembangkan oleh Google, berdasarkan ribuan desain sebelumnya, membuktikan kemampuannya untuk menghasilkan tata letak yang paling efisien dalam hitungan jam dengan hasil yang kompetitif atau lebih baik daripada para ahli manusia.

**Pengaruh pada Daya**: Terdapat pengurangan interkoneksi panjang dalam optimasi berbasis RL, mengurangi disipasi daya dinamis karena pengurangan kapasitansi switching.

**Relevansi Industri**: Ini telah diimplementasikan di TPU Google. Kegunaan akselerator (*Tensor Processing Unit*) terbukti bermanfaat di dunia nyata.

**Arsitektur GPU yang Ditingkatkan AI oleh NVIDIA**

NVIDIA telah menggunakan eksplorasi arsitektur GPU-nya dengan alat AI/ML. Insinyur dapat memprediksi kinerja per watt dengan menggunakan model pembelajaran terawasi dalam pemilihan varian desain. ML mempercepat penemuan ruang desain melalui eliminasi kandidat yang tidak optimal, dan memungkinkan insinyur untuk mencapai efisiensi yang lebih besar dalam waktu yang jauh lebih singkat.

- **Dampak pada Daya**: Keputusan mikro-arsitektur yang lebih baik menghasilkan peningkatan efisiensi energi untuk beban kerja AI.
- **Pengaruh pada Daya**: Peningkatan energi dicapai jika pilihan mikro-arsitektur dan efisiensi untuk beban kerja AI ditingkatkan.
- **Relevansi Industri**: Kartu grafis partisi Ampere dan Hopper oleh NVIDIA. generasi.

**Pembahasan Neuromorfik dan Daya Rendah Intel**

Chip neuromorfik Loihi Intel merupakan langkah menjauh dari komputasi berbasis peristiwa yang terinspirasi oleh otak manusia. Loihi adalah desain yang mampu menjalankan beban kerja AI berat pada jaringan saraf spiking berdaya rendah, yang menggunakan energi jauh lebih sedikit daripada arsitektur hierarki komputasi. Praktik desain berbasis AI

mengoptimalkan interkoneksi dan power gating untuk beban kerja asinkron. Menunjukkan bagaimana AI bukan hanya alat desain, tetapi juga target desain perangkat keras, yaitu mengoptimalkan perangkat lunak, arsitektur, dan silikon secara bersamaan.

**Arsitektur Big.LITTLE ARM dan Adaptasi AI Saat Runtime**

ARM memperkenalkan arsitektur big.LITTLE pada prosesor mobile untuk menyeimbangkan kinerja dan daya. Core "big" yang besar menangani tugas-tugas komputasi intensif, sementara core "LITTLE" yang lebih kecil menangani beban kerja latar belakang. Klasifikasi beban kerja berbasis AI saat runtime memprediksi perilaku tugas dan beralih core secara dinamis. Hal ini menghasilkan penghematan daya hingga 75% pada beban kerja smartphone pada umumnya. Pelajaran kuncinya adalah bahwa adaptasi saat runtime berbasis AI melengkapi desain daya rendah statis dengan memberikan efisiensi energi yang peka terhadap konteks.

**Optimasi SoC Berbasis ML Apple**

SoC kustom Apple (prosesor M1 dan seri A) menggunakan ML baik pada waktu desain maupun saat runtime. Alat EDA yang dibantu ML mengoptimalkan tata letak dan pengaturan waktu. Model ML saat runtime mengelola kebijakan DVFS, memprediksi kapan harus menurunkan tegangan/frekuensi tanpa kehilangan kinerja yang dirasakan pengguna. Mencapai rasio perf/W terdepan di industri pada prosesor mobile dan laptop.

**Studi Akademis: Prediksi Kebocoran dan DVFS**

Selain industri, studi ilmiah telah menunjukkan penerapan model ML dalam prediksi daya dan penskalaan adaptif:

**Model berbasis ML untuk prediksi kebocoran** telah menunjukkan akurasi lebih dari 90% dalam memperkirakan arus kebocoran primer antar node teknologi, yang lebih baik daripada model berbasis aturan. Teknik pembelajaran penguatan telah diterapkan pada DVFS, memungkinkan penyesuaian pengaturan tegangan/frekuensi secara real-time untuk prosesor, meminimalkan energi sambil memenuhi target kinerja.

**Inovasi Perangkat Berbasis FinFET**: Inovasi - FinFET tiga gerbang 10 nm. Telah ditunjukkan bahwa perangkat berbasis FinFET dapat menggunakan FinFET tiga gerbang, kemiringan subambang yang ditingkatkan, dan kontrol kebocoran. Ini adalah optimasi tingkat perangkat dalam AI yang dapat dikombinasikan dengan optimasi tingkat sirkuit.

**Arsitektur SRAM untuk Aplikasi Biomedis**: Arsitektur sel SRAM 7T dalam aplikasi Biomedis Dioptimalkan menggunakan sel SRAM 7T. Simulasi berbasis ML meminimalkan kebocoran siaga, yang sangat penting dalam sensor biomedis bertenaga baterai (Tabel 4.3.

**Tabel 4.3** Perbandingan studi kasus yang berbeda

| No. | Studi Kasus | Teknik AI/ML | Tahap yang Ditargetkan | Dampak terhadap Efisiensi Daya |
|---|---|---|---|---|
| 1 | Google TPU – Floorplanning | Reinforcement Learning | Desain fisik (*Physical Design*) | Panjang jalur interkoneksi (*wirelength*) menurun, |

| | | | | |
|---|---|---|---|---|
| | | (Pembelajaran Penguatan) | | konsumsi daya dinamis menurun |
| 2 | GPU NVIDIA | Machine Learning Terawasi (Supervised ML) | Eksplorasi arsitektur (*Architectural Exploration*) | Rasio performa terhadap daya (*Performance/Watt*) meningkat, waktu eksplorasi berkurang |
| 3 | Intel Loihi | Optimasi berbantuan AI | Perangkat keras neuromorfik (*Neuromorphic Hardware*) | Mendukung jaringan *spiking neural network* dengan konsumsi daya sangat rendah |
| 4 | ARM big.LITTLE | Pengklasifikasi *Machine Learning* saat *runtime* | DVFS (Dynamic Voltage and Frequency Scaling) dan penjadwalan inti (*Core Scheduling*) | Penghematan energi hingga 75% |
| 5 | Apple M1 | *Machine Learning* pada tahap desain dan saat *runtime* | System-on-Chip (SoC) dan DVFS | Efisiensi performa per watt (*Performance/Watt*) terbaik di kelas industri |
| 6 | Academic FinFET | Model Regresi / *Machine Learning* | Perangkat dan proses fabrikasi (*Device/Process*) | Daya bocor (*Leakage Power*) berkurang, karakteristik Subthreshold Slope meningkat |
| 7 | Academic 7T SRAM | Optimasi berbasis *Machine Learning* | Arsitektur memori (*Memory Architecture*) | Daya bocor saat siaga (*Standby Leakage*) menurun, siap untuk aplikasi Biomedical IoT |

## 4.8 TANTANGAN DAN ISU PEMBAHASAN TERBUKA

Integrasi AI dan ML ke dalam desain VLSI berdaya rendah telah menghasilkan kemajuan yang luar biasa, namun beberapa tantangan teknis dan metodologis masih belum terselesaikan. Tantangan-tantangan ini menentukan arah pembahasan saat ini dan di masa mendatang, dan pemahaman tentangnya sangat penting bagi praktisi dan akademisi. Meskipun AI dan ML telah menunjukkan potensi yang luar biasa dalam desain VLSI berdaya rendah, penerapannya tidak lepas dari hambatan yang signifikan. Tantangan-tantangan ini mencakup ketersediaan data, kepercayaan model, biaya komputasi, penskalaan teknologi, dan risiko keamanan. Mengatasinya sangat penting untuk memastikan AI/ML dapat berkembang dari kisah sukses eksperimental menjadi praktik industri arus utama.

Persyaratan dan Masalah Pembahasan Terbuka. Pengenalan AI dan ML ke dalam desain pengembangan VLSI berdaya rendah telah menghasilkan hasil yang fenomenal. Namun, beberapa masalah teknis dan metodologis masih belum terselesaikan. Masalah-masalah

tersebut menentukan alur pembahasan yang ada dan di masa mendatang, dan penting untuk memahaminya baik oleh akademisi maupun praktisi. Penerapannya telah menunjukkan bahwa ML memiliki potensi luar biasa dengan desain VLSI berdaya rendah, namun tidak ada jalan tanpa kendala serius. Masalah-masalah ini mencakup ketersediaan data, kepercayaan pada model, biaya komputasi, penskalaan teknologi, dan ancaman keamanan. Mengatasinya penting untuk memungkinkan AI/ML berkembang melampaui kisah sukses eksperimental dan praktik industri umum.

## 4.9 INTERPRETABILITAS DAN KEPERCAYAAN MODEL

- **Masalah**: Model DL bertindak sebagai "kotak hitam," sehingga sulit untuk menjelaskan mengapa keputusan desain tertentu diambil.
- **Isu:** Model DL adalah kotak hitam dan karenanya tidak mudah dipahami mengapa pilihan desain tertentu dibuat.
- **Dampak:** Perancang enggan mempercayai alat yang sepenuhnya otomatis untuk langkah-langkah desain kritis.
- **Dampak**: Perancang tidak ingin memasukkan langkah-langkah desain penting ke dalam alat yang sepenuhnya otomatis.
- **Area Pembahasan Terbuka**: Memajukan AI yang dapat dijelaskan (XAI) yang disesuaikan untuk desain VLSI untuk memberikan pengambilan keputusan yang transparan dan dapat dilacak.
- **Area Pembahasan Terbuka**: Mengembangkan AI yang dapat dijelaskan (XAI) khusus untuk desain VLSI. memberikan pengambilan keputusan yang transparan dan dapat dilacak.

## 4.10 BEBAN KOMPUTASI

- **Masalah:** Melatih model deep learning untuk floorplanning, prediksi daya, atau optimasi DVFS membutuhkan komputasi yang intensif.
- **Isu:** Pelatihan model deep learning: floorplanning, prediksi daya, atau pelatihan DVFS secara langsung. Optimasi adalah proses yang intensif informasi.
- **Dampak:** Hal ini menciptakan paradoks – mengonsumsi energi tinggi untuk merancang sistem berdaya rendah.
- **Dampak:** Ini membentuk paradoks, yaitu, menggunakan energi tinggi untuk mengembangkan sistem berdaya rendah.
- **Area Pembahasan Terbuka:** Mengembangkan algoritma ML ringan yang dioptimalkan untuk alat CAD dan penerapan edge computing.

## 4.11 GENERALISASI DI SELURUH NODE TEKNOLOGI

- **Masalah:** Model yang dilatih pada 28 nm mungkin tidak dapat digeneralisasi dengan baik ke 7 nm atau node 3 nm di masa mendatang karena penskalaan kebocoran dan variabilitas.
- **Dampak:** Membutuhkan pelatihan ulang, yang mahal dan memakan waktu.

- **Area Pembahasan Terbuka:** Teknik transfer learning dan adaptasi domain untuk generalisasi lintas node.

## 4.12 INTEGRASI DENGAN ALAT CAD YANG ADA

- **Masalah**: Alat standar industri (Cadence, Synopsys, Mentor Graphics) didasarkan pada algoritma heuristik yang telah teruji dengan baik.
- **Dampak**: Mengintegrasikan AI/ML tanpa mengganggu alur kerja bukanlah hal yang mudah.
- **Area Pembahasan Terbuka**: Menciptakan kerangka kerja CAD hibrida di mana ML menambah – bukan menggantikan – alat EDA yang ada.

## 4.13 KEAMANAN DAN KETANGGUHAN

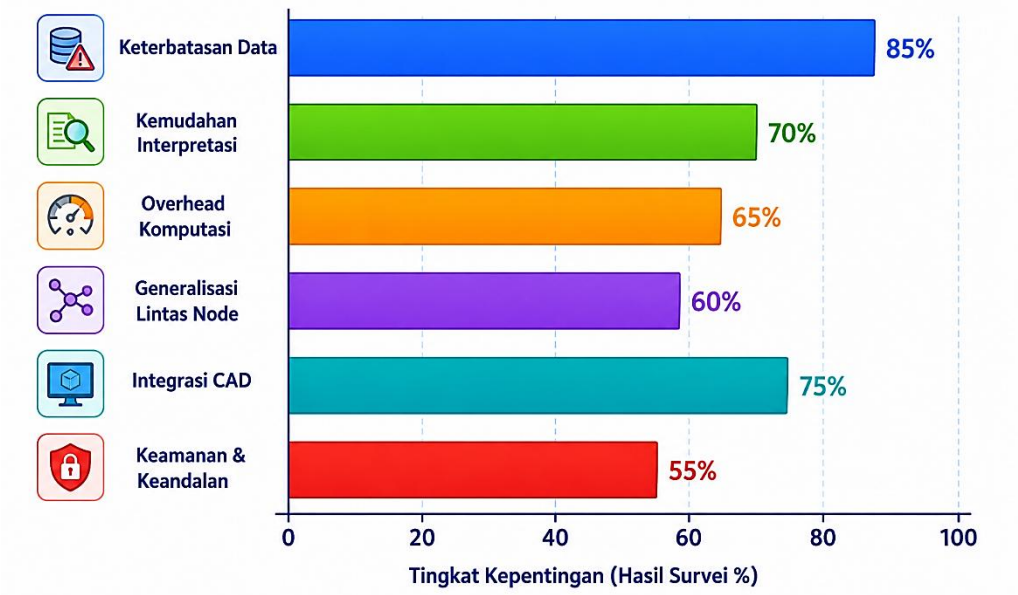
- **Masalah:** Model AI mungkin rentan terhadap serangan adversarial, di mana gangguan kecil pada data input menyesatkan model.
- **Dampak:** Dapat menyebabkan desain chip yang cacat dengan kerentanan tersembunyi.
- **Area Pembahasan Terbuka:** Mengembangkan model AI yang tangguh dan tahan terhadap manipulasi adversarial di domain VLSI (Tabel 4.4).

**Tabel 4.4** Tantangan utama dan arah pembahasan dalam desain VLSI hemat daya berbasis AI

| No. | Tantangan | Dampak | Arah Pembahasan |
|---|---|---|---|
| 1 | Keterbatasan Data *(Data Scarcity)* | Keterbatasan data untuk proses pelatihan (*training*) dan validasi model | Pengembangan **data sintetis (synthetic data)** dan **Federated Learning** |
| 2 | Interpretabilitas Model *(Model Interpretability)* | Kurangnya kepercayaan perancang (*designer*) terhadap hasil yang diberikan model AI | Pengembangan **Explainable Artificial Intelligence (XAI)** atau AI yang dapat dijelaskan |
| 3 | Overhead Komputasi *(Computational Overhead)* | Tingginya konsumsi energi dan biaya komputasi selama pelatihan model AI | Pengembangan model AI ringan (*lightweight models*) dan **Edge Machine Learning (Edge ML)** |
| 4 | Generalisasi Lintas Node Teknologi *(Cross-Node Generalization)* | Model AI perlu dilatih ulang (retraining) untuk setiap node teknologi baru | Transfer Learning dan Adaptasi Domain (Domain Adaptation) |
| 5 | Integrasi dengan Perangkat Lunak CAD *(CAD Tool Integration)* | Gangguan pada alur kerja (workflow disruptions) akibat integrasi AI dengan perangkat lunak CAD | Pengembangan kerangka kerja Hybrid Machine Learning + Heuristic CAD Frameworks |
| 6 | Kerentanan Keamanan *(Security Vulnerabilities)* | Risiko manipulasi desain melalui serangan adversarial (adversarial design manipulation) | Pengembangan Machine Learning yang tangguh (Robust ML) dan metode deteksi anomali (Anomaly Detection) |

## 4.14 ARAH MASA DEPAN DAN TREN YANG MUNCUL

Implementasi AI dan ML dalam desain VLSI berdaya rendah masih dalam tahap awal, namun terdapat banyak arah menjanjikan yang dapat diikuti seiring perkembangan bidang ini dalam beberapa tahun mendatang. Bagian ini membahas tren penting di masa depan, yang kemungkinan besar akan memengaruhi pembahasan, industri, dan inovasi semikonduktor (Gambar 4.6).




**Gambar 4.6** Pentingnya tantangan AI/ML dalam VLSI berdaya rendah menurut persepsi industri.

## 4.15 KOMPUTASI NEUROMORFIK UNTUK ARSITEKTUR BERDAYA SANGAT RENDAH

Sistem neuromorfik dirancang untuk mereplikasi struktur otak, dengan bantuan implementasi jaringan saraf spiking (SNN) pada perangkat keras. Tidak seperti perangkat keras neuromorfik tradisional, perangkat keras ini diwujudkan oleh arsitektur Neumann yang mengonsumsi lebih sedikit energi secara default melalui bantuan komputasi berbasis peristiwa. Perusahaan seperti Intel (Loihi) dan kelompok riset di seluruh dunia sedang mengembangkan chip neuromorfik yang memiliki kemampuan untuk mengubah definisi dunia, IoT, implan biomedis, dan aplikasi edge. Beberapa aplikasi umum yang selalu aktif dan hemat daya meliputi IoT, implan biomedis, dan aplikasi edge. Alur VLSI masa depan dapat didasarkan pada alat desain berbasis AI yang disesuaikan untuk optimasi sirkuit neuromorfik, yang memungkinkan penggunaan arsitektur yang terinspirasi secara biologis untuk meningkatkan efisiensi pembelajaran federasi dalam desain semikonduktor.

## 4.16 PEMBELAJARAN TERFEDERASI DALAM DESAIN SEMIKONDUKTOR

Model ML klasik membutuhkan data besar dan terpusat. Namun, informasi desain semikonduktor biasanya tersebar di berbagai organisasi dan dibatasi oleh perlindungan hak kekayaan intelektual (IP). Pembelajaran terfederasi (FL) memungkinkan model AI untuk dilatih

secara real-time tanpa pertukaran data mentah, dan memastikan privasi data dan perlindungan IP. Hal ini memungkinkan perusahaan semikonduktor untuk melatih model secara bersama-sama untuk mendesain pesawat ruang angkasa, perencanaan kegagalan, dan optimasi proses, serta menyimpan data kepemilikan secara lokal. Pembahasan lain dalam FL juga menyelidiki pengurangan overhead komunikasi dan ketahanan terhadap serangan adversarial, yang keduanya sangat penting untuk alur kerja CAD.

## 4.17 AI TERJELASKAN (XAI) UNTUK OTOMASI DESAIN TERPERCAYA

Sifat "kotak hitam" dari ML adalah salah satu kendala terbesar ML dalam desain VLSI model deep learning. Metode Kecerdasan Buatan yang Dapat Dijelaskan (XAI) mampu mengungkap alasan mengapa suatu model memberikan rekomendasi desain tertentu, dan ini memiliki pengaruh positif pada adopsi oleh para desainer. Dengan menggunakan metrik interpretasi, para insinyur dapat yakin bahwa platform CAD Proksimal berbasis ML dapat memberikan saran di masa mendatang dalam perencanaan tata ruang, prediksi daya, dan perutean. Platform CAD di masa mendatang mungkin akan mengintegrasikan dasbor XAI, memungkinkan perbandingan berdampingan antara keputusan desain heuristik dan berbasis AI.

## 4.18 PEMBELAJARAN PENGUATAN UNTUK ALUR DESAIN ADAPTIF

Aplikasi RL saat ini dalam perencanaan tata ruang dan penempatan menunjukkan potensi yang luar biasa, tetapi arah masa depan melibatkan alur berbasis RL ujung-ke-ujung yang dapat menyesuaikan diri dengan kendala. Alih-alih hanya mengoptimalkan satu tahap, agen RL dapat secara dinamis menyeimbangkan pertimbangan di seluruh tingkat sistem, sintesis logika, desain fisik, dan pengujian. Agen RL multi-objektif semacam itu akan mampu belajar terus menerus seiring dengan menyusutnya node teknologi dan berkembangnya kendala. Kemampuan adaptasi ini memastikan peningkatan kinerja per watt yang berkelanjutan selama generasi chip berikutnya.

## 4.19 AI YANG TERINSPIRASI KUANTUM UNTUK OPTIMASI VLSI

Seiring kemajuan komputasi kuantum, pendekatan hibrida yang menggabungkan algoritma yang terinspirasi kuantum dengan ML dapat mengatasi masalah NP-hard dalam desain VLSI (misalnya, penempatan, perutean, dan penjadwalan). Meskipun alat CAD kuantum sepenuhnya masih dalam pengembangan, metaheuristik yang terinspirasi kuantum (misalnya, Q-learning dengan konsep annealing kuantum) sudah dapat memberikan manfaat. Metode tersebut dapat secara drastis mengurangi waktu penutupan desain untuk sirkuit kompleks.

## 4.20 AI BERKELANJUTAN DAN RAMAH LINGKUNGAN UNTUK DESAIN DAYA RENDAH

Pelatihan AI itu sendiri mungkin mahal secara komputasi dan energi. Fokus studi lebih lanjut adalah pada AI ramah lingkungan, yang dioptimalkan tidak hanya dalam desain chip VLSI tetapi juga dalam desain algoritma ML. Beberapa model ringan (TinyML, pemangkasan, kuantisasi) akan sangat penting dalam mengurangi jumlah energi yang dikonsumsi oleh

perangkat lunak CAD berbasis AI saat diterapkan. Hal ini sejalan dengan tren internasional dalam pengembangan semikonduktor secara berkelanjutan.

## 4.21 ERA BARU VLSI BERDAYA RENDAH BERBASIS AI

Desain VLSI berdaya tinggi sedang direvolusi oleh adopsi AI dan ML. Seperti yang telah ditunjukkan, efisiensi energi adalah tujuan dan bukan lagi kebutuhan dasar di seluruh aplikasi seperti komputasi tepi dan pakaian pintar hingga mobil otonom dan IoT. Metode desain tradisional, yang cukup mapan, kesulitan untuk mengimbangi pertumbuhan, kompleksitas PAP, sirkuit yang rumit, dan kebutuhan akan fleksibilitas waktu nyata dari sistem kontemporer. Dalam inovasi perangkat, FinFET (MOSFET tanpa sambungan), otomatisasi desain berbasis AI/ML. Kontribusinya membuktikan topologi SRAM berdaya rendah dan transistor baru. Struktur AI modern dapat membantu mengevaluasi, memodelkan, dan mengoptimalkan desain.

AI dan ML memiliki solusi dengan kekuatan besar yang memperkenalkan pemodelan prediktif, optimasi adaptif, dan otomatisasi cerdas di semua tingkat alur desain VLSI. Dari eksplorasi arsitektur tingkat sistem hingga sintesis logika, desain fisik, dan pengujian/verifikasi, teknik-teknik ini dapat digunakan untuk dengan cepat melintasi area luas di antara para desainer, menciptakan ruang dalam desain yang lebih akurat dan efisien dibandingkan dengan heuristik tradisional.

Dalam studi kasus, perencanaan tata ruang berbasis pembelajaran penguatan Google atau optimasi arsitektur berbasis AI dari NVIDIA, dll., menunjukkan bagaimana para pemimpin industri telah membentuk praktik desain. Melalui sumber konsumsi daya, pertimbangan PAP, AI, penggunaan teknik, aplikasi alur desain, dan tren baru, kita menemukan bagaimana pembahasannya tidak hanya berkontribusi tetapi juga mendukung kisah AI/ML sebagai pilar VLSI berdaya rendah.

Pada saat yang sama, masih ada tantangan serius yang tersisa, seperti kelangkaan data, interpretasi, beban komputasi, generalisasi lintas node, dan integrasi alat CAD, yang merupakan hambatan nyata untuk adopsi yang luas. Jika kita tidak menyelesaikan masalah ini, tidak akan cukup, bukan hanya kemajuan teknis, tetapi juga struktur organisasi di bidang akademisi dan industri. Secara khusus, munculnya AI yang dapat dijelaskan (XAI) dan FL dapat berfokus pada isu-isu terkait kepercayaan, transparansi, dan perlindungan kekayaan intelektual dalam desain semikonduktor.

Di masa depan, sejumlah tren baru memiliki potensi besar untuk mengarah pada masa depan VLSI cerdas hemat daya:

Komputasi neuromorfik memiliki FL ultra-hemat daya yang khas secara biologis. FL adalah arsitektur yang memungkinkan pengembangan model kolaboratif yang menjaga privasi. AI yang dapat dijelaskan akan meningkatkan kepercayaan dalam adopsi perancang, dan pembelajaran penguatan dapat dikembangkan menjadi desain adaptif ujung-ke-ujung. Go Green, AI, yang terinspirasi kuantum, dan otomatisasi akan mendorong inovasi berkelanjutan lebih jauh lagi. Artinya, AI dan ML bukan lagi perangkat periferal; mereka menjadi kemampuan sentral dari

pendekatan desain VLSI hemat daya generasi berikutnya. Peran mereka dalam mengembangkan sirkuit yang terukur, hemat daya, dan cerdas merupakan bahan penting dalam masa depan elektronika. Pasar semikonduktor masih beroperasi di bawah kendala daya dan kompleksitas desain yang meningkat, dan alur desain berbasis AI akan menjadi fondasi untuk membangun perangkat keras dan perangkat lunak yang lebih cepat, lebih cerdas, dan lebih ramah lingkungan.

# BAB 5
# KOMPUTASI NEUROMORFIK VLSI CMOS RENDAH DAYA

## 5.1 KOMPUTASI NEUROMORFIK DAN ARSITEKTUR TERINSPIRASI OTAK

Komputasi neuromorfik mewakili perubahan mendasar dari paradigma komputasi tradisional; khususnya, ia berasal dari jaringan saraf biologis otak manusia dan bertujuan untuk mereplikasi banyak prinsip arsitektur utama yang ditunjukkan oleh otak; ini termasuk, di antara banyak lainnya, pemrosesan paralel tinggi, integrasi memori dan komputasi, dan komunikasi berbasis peristiwa. Komputer tradisional memproses instruksi secara sekuensial yang ketat, sedangkan sistem neuromorfik terdiri dari neuron dan sinapsis buatan yang berkomunikasi menggunakan "lonjakan" (atau pulsa) listrik. Model jaringan saraf spiking (SNN), seperti yang diketahui, menyerupai pelepasan impuls neuron biologis dalam arti bahwa informasi hanya disampaikan setelah ambang batas pelepasan impuls tertentu terpenuhi, dan tidak seperti komputer tradisional, setiap neuron biologis berkomunikasi melalui jalur komunikasi terpisah.

Prosesor standar menggunakan aliran data yang kontinu dan sinkron yang menggambarkan pemrosesan dan tindakan yang tepat untuk menghasilkan hasil yang telah ditentukan – yaitu, T = 0, 1, 2, 3; suatu tindakan terjadi pada T = 3: kirim “node A” ATAU proses input dan output dan lakukan tindakan. Motivasi utama di balik paradigma ini adalah upaya untuk menciptakan implementasi perangkat keras yang merupakan prosesor yang lebih baik, atau prosesor yang lebih efisien, daripada kalkulator; prosesor yang mendekati perilaku cerdas, seperti perilaku yang terlihat dalam pola dan pemrosesan emosi sensorik dan tindakan organisme individu, dan tugas-tugas yang secara inheren sulit bagi komputer standar (yaitu, pengenalan, pemrosesan sensorik, pembelajaran/percobaan, dll.) yang sangat baik diproses oleh otak. Hal ini akan dicapai dengan membangun sistem yang dicirikan oleh fungsi otak asinkron dan sifat paralel masif dari desain arsitektur otak, berbeda dengan fungsi P = 1 dari arsitektur komputer standar, yang sangat tidak memadai untuk mencapai kecepatan eksekusi yang signifikan dan implementasi perangkat keras yang hemat energi untuk domain aplikasi tertentu.

Arsitektur von Neumann tradisional, yang telah menjadi inti komputasi selama beberapa dekade, memiliki pembagian yang jelas antara Unit Pemroses Sentral (CPU) dan memori. Desain fundamental ini telah terbukti berhasil dalam melakukan berbagai macam tugas; namun, ada keterbatasan utama yang melekat pada jenis desain ini. Ini dikenal sebagai hambatan von Neumann atau dinding memori. Hambatan ini disebabkan oleh pergerakan data yang konstan antara prosesor dan memori, yang memakan banyak waktu dan energi. Karena pembelajaran mendalam dan pembelajaran mesin membutuhkan banyak data, hambatan von Neumann telah menjadi hambatan kinerja dan energi utama.

Kumpulan data dan model saat ini jauh lebih besar daripada sebelumnya dan membutuhkan banyak pergerakan data yang jauh melebihi arsitektur von Neumann. Selain itu, meningkatnya jumlah perangkat komputasi mobile dan edge yang memiliki batasan yang lebih

ketat pada konsumsi daya dan ruang fisik akan membuat bottleneck von Neumann menjadi kekurangan yang kritis. Arsitektur von Neumann harus ditinggalkan untuk arsitektur baru agar generasi sistem komputasi berkinerja tinggi dan hemat energi berikutnya dapat muncul. Kekuatan komputasi neuromorfik terletak pada kemampuannya untuk mengatasi tantangan yang terkait dengan arsitektur von Neumann, seperti pemisahan antara pemrosesan dan memori, yang membutuhkan komunikasi melalui koneksi pada sirkuit (atau silikon) dan menimbulkan latensi dan konsumsi energi.

Chip neuromorfik memfasilitasi komputasi dalam memori, yang tidak hanya mengurangi energi dan waktu yang dibutuhkan sinyal data untuk bergerak, tetapi juga memungkinkannya terjadi pada kecepatan sinyal data merambat di sepanjang koneksi dalam silikon. Sifat berbasis peristiwa dari SNN berarti daya hanya dikonsumsi saat neuron memproses (mengirim atau menerima informasi). Ini berpotensi menjadi efisiensi energi terbaik, dan pertimbangan daya sangat penting untuk aplikasi seperti robotika mobile, kendaraan otonom, perangkat yang dapat dikenakan, dll.

Selain itu, sistem neuromorfik secara alami cocok untuk aplikasi real-time, yang berkaitan erat dengan pembelajaran online adaptif. Arsitektur sistem memungkinkan paralelisme untuk memproses sejumlah besar data secara bersamaan, karena berkaitan dengan data sensor real-time dari kamera, mikrofon, atau data EEG dalam antarmuka otak-komputer. Chip dan sistem neuromorfik dapat memungkinkan pembelajaran on-chip tanpa memerlukan program pelatihan eksternal sebelum operasi normal dalam agen otonom.

Pada akhirnya, komputasi neuromorfik menghadirkan peluang untuk membangun sistem yang menggunakan energi jauh lebih sedikit sambil tetap melakukan tugas prosesor digital canggih yang dibutuhkan dalam sistem AI dan ML. Ini bukan sekadar peningkatan bertahap tetapi pemikiran ulang dari bawah ke atas tentang bagaimana komputasi dapat dilakukan untuk menghasilkan solusi hemat daya yang terinspirasi oleh otak untuk berbagai macam aplikasi.

## 5.2 PRINSIP-PRINSIP KOMPUTASI NEUROMORFIK DALAM VLSI CMOS DAYA RENDAH

Tugas ini melibatkan transfer proses biologis yang sangat hati-hati ke perangkat keras silikon untuk mengimplementasikan prinsip-prinsip neuromorfik dalam sirkuit VLSI CMOS daya rendah. Tujuan utama dari tugas ini adalah untuk mengimplementasikan emulasi neuron dan sinapsis biologis menggunakan metode analog dan sinyal campuran dalam desain sirkuit. Sirkuit neuromorfik cenderung menggunakan komponen analog untuk melakukan komputasi, berbeda dengan gerbang logika digital tradisional dan operasi sekuensial. Misalnya, integrasi arus dapat mensimulasikan potensial membran neuron biologis, dan perubahan konduktansi serangkaian transistor secara terkoordinasi dapat memodelkan bobot sinaptik. Beberapa karakteristik unik neuron biologis yang dapat dipetakan dengan tepat ke kondisi daya rendah meliputi sifat komputasi asinkron dan berbasis peristiwa. Sirkuit neuromorfik beroperasi pada daya rendah hingga "spike" dihasilkan atau diproses, dan sebaliknya memiliki keadaan diam dengan daya rendah (berbeda dengan contoh komputasi digital standar, yang diatur oleh clock dan mengkonsumsi daya terlepas dari hasilnya atau kegunaannya.

Operasi transistor telah menjadi semakin kompleks dan beragam selama beberapa dekade. Wilayah sub-threshold, di mana perangkat diberi bias pada tegangan di bawah ambang operasi normal, menjadi sangat penting untuk dipertimbangkan sekarang. Operasi pada tegangan ini dapat memberikan pengurangan daya yang substansial bagi perancang tetapi menimbulkan kompromi, seperti menjadi lebih lambat atau lebih sensitif terhadap noise. Keterbatasan ini dapat ditoleransi untuk aplikasi yang terinspirasi otak yang tidak memerlukan pemrosesan ultra-cepat.

**Tantangan Desain: Meminimalkan Konsumsi Daya dan Meningkatkan Kinerja**

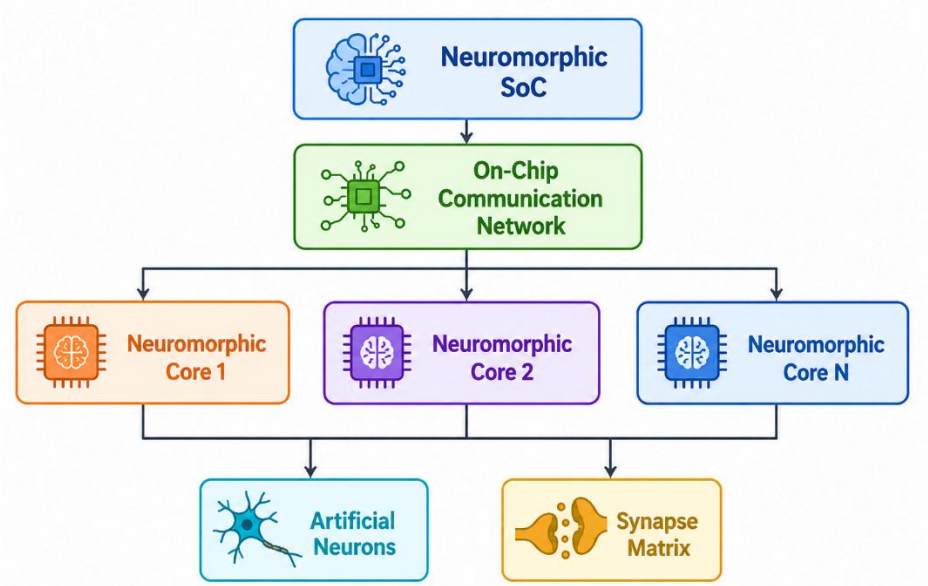
Membangun arsitektur neuromorfik dalam CMOS memiliki tantangan desain unik yang menyeimbangkan kinerja, daya, dan akurasi. Misalnya, daya rendah adalah persyaratan utama bagi pengguna akhir. Daya rendah penting untuk AI edge dan sistem tertanam, tetapi seringkali berarti mengorbankan kinerja. Meminimalkan konsumsi daya berarti keringkasan di seluruh desain sirkuit, pilihan arsitektur, dan tujuan teknologi proses. Misalnya, mendesain sirkuit di wilayah sub-threshold dapat sangat hemat daya tetapi dapat menyebabkan variabilitas dan kebisingan yang signifikan yang perlu diatasi untuk mencapai kinerja yang diinginkan dari desain.

Sistem neuromorfik tidak dibandingkan pada metrik tradisional seperti kecepatan clock; mereka dibandingkan pada kejadian per detik atau laju pembaruan sinapsis. Jadi, meningkatkan kinerja suatu sistem berarti mengoptimalkan interaksi antara neuron dan sinapsis dan mengurangi latensi spike untuk memastikan propagasi cepat di seluruh chip kita. Saat kita mendesain sistem neuromorfik, kita harus berurusan dengan trade-off analog vs. digital. Sirkuit analog dapat sangat hemat energi saat melakukan komputasi tetapi juga dapat rentan terhadap kebisingan, sangat sensitif terhadap variasi proses, dan memerlukan kalibrasi. Sebaliknya, sirkuit digital memiliki tingkat ketahanan yang lebih tinggi, tetapi seringkali kurang hemat daya. Sebagian besar sistem neuromorfik praktis saat ini menggunakan pendekatan sinyal campuran untuk memanfaatkan kedua jenis sirkuit.

**Skalabilitas dan Pertimbangan Arsitektur**

Aspek penting dari desain sirkuit neuromorfik adalah skalabilitas harus dipertimbangkan. Otak manusia terdiri dari miliaran neuron dan triliunan sinapsis, skala yang saat ini tidak dapat dicapai dalam satu chip silikon. Rute paling umum menuju skalabilitas adalah arsitektur modular atau ubin. Dalam hal ini, chip dibagi menjadi beberapa "inti" atau "ubin" identik, dengan setiap ubin terdiri dari kumpulan lokal neuron dan sinapsis buatan, seperti yang diilustrasikan pada Gambar 5.1. Selain itu, ubin-ubin tersebut saling terhubung dengan jaringan on-chip untuk memungkinkan komunikasi dan perutean spike ke berbagai bagian chip. Modularitas desain ini menyederhanakan proses desain dan memungkinkan sistem untuk diskalakan dengan menambahkan lebih banyak ubin, meskipun masalah baru topologi jaringan, protokol perutean, dan kemacetan sekarang harus dipecahkan. Kinerja jaringan komunikasi antar-ubin tidak boleh kurang efisien daripada kinerja sirkuit neuron dan sinapsis lokal. Pertimbangan desain struktural juga mencakup pilihan komunikasi, baik berbasis peristiwa atau berbasis waktu.

Seperti yang telah dibahas sebelumnya, sistem berbasis peristiwa hanya mengirimkan data ketika neuron aktif, yang menyebabkan komunikasi yang jarang dan daya yang rendah. Sistem berbasis waktu cenderung memiliki komunikasi yang lebih kuat, dengan mengorbankan efisiensi. Pada akhirnya, desain arsitektur terbaik akan sangat bergantung pada target aplikasi dan persyaratan kinerja untuk daya, latensi, throughput, dll.




**Gambar 5.1** Diagram blok yang menggambarkan arsitektur sistem-on-chip (SoC) neuromorfik tipikal.

## 5.3 BLOK BANGUNAN TINGKAT TRANSISTOR CMOS

### Neuron dan Sinapsis Buatan: Implementasi Tingkat Sirkuit

Neuron dan sinapsis buatan yang dimaksudkan untuk meniru neuron dan sinapsis biologis adalah elemen pembangun penting dari sistem neuromorfik. Dalam implementasi CMOS standar, mereka dibuat dari kombinasi diskrit transistor, kapasitor, dan resistor. Neuron buatan adalah unit dasar pemrosesan dan, untuk tujuan implementasi generik, dapat dimodelkan sebagai neuron "integrasi dan tembak", yang menggunakan sirkuit yang memodelkan neuron dengan menggunakan kapasitor untuk meniru potensial membran neuron (yaitu, kapasitor membangun muatan dari input sinaptik atau eksternal yang masuk).

Ketika tegangan kapasitor melebihi ambang batas penembakan yang telah ditentukan, neuron "buatan" menembak dengan menghasilkan pulsa/lonjakan listrik pendek. Transistor digunakan sebagai sakelar untuk mengatur ulang tegangan kapasitor, sekali lagi meniru periode refraktori yang ditunjukkan oleh neuron biologis. Untuk konsumsi daya rendah, sirkuit ini sering menggunakan ruang solusi untuk membiaskan transistor di bawah ambang batas, yang memungkinkan arus bocor yang berkurang secara substansial, sehingga menghasilkan kehilangan daya statis minimal. Sinapsis buatan menghubungkan neuron dan menyimpan "bobot," atau kekuatan koneksi, biasanya sebagai konduktansi variabel. Konduktansi mengontrol muatan yang dapat disuntikkan ke neuron postsynaptic untuk setiap lonjakan yang masuk, memvariasikan kekuatan, atau sinyal. Selain melakukan operasi ini, sirkuit sinaptik ini

juga harus menyimpan bobot, yang seringkali membutuhkan elemen memori, yang menghadirkan hambatan signifikan dalam hal efisiensi daya dan area dari CMOS tradisional.

**Teknologi Memori Baru untuk Sirkuit Neuromorfik**

Teknologi CMOS sangat bagus untuk membangun elemen logika dan switching di neuron, tetapi kurang dioptimalkan untuk menyediakan penyimpanan bobot sinaptik non-volatil dengan kepadatan tinggi. Solusi memori yang ada, seperti SRAM dan DRAM, bersifat volatil dan mengkonsumsi daya yang berlebihan. Oleh karena itu, solusi memori yang ada tidak akan berfungsi untuk sistem neuromorfik skala besar dan hemat energi. Sebagai solusinya, terdapat pembahasan tentang teknologi memori baru yang telah muncul, khususnya memristor dan material perubahan fasa (PCM).

Memristor adalah perangkat dua terminal pasif, juga dikenal sebagai "resistor memori," yang tingkat resistansinya bergantung pada riwayat arus listrik. Meskipun memristor anorganik terdiri dari perangkat solid-state, mereka tetap mensimulasikan bobot sinaptik dan dapat mencapai memori non-volatil karena rentang penskalaannya. Memristor dapat mengingat keadaannya tanpa input daya dan dapat dianggap sebagai resistor yang dapat diprogram. Array crossbar memristor adalah salah satu metode terpadat dan paling efisien dalam hal luas area untuk menggunakan matriks sinaptik besar yang memungkinkan perhitungan paralel tinggi dalam memori.

PCM biasanya membutuhkan pemanasan yang signifikan untuk transisi cepat dari keadaan kristal ke amorf (resistansi lebih tinggi). Hal ini berpotensi digunakan untuk mencapai fungsi yang sama seperti memristor berdasarkan keadaan resistansi, sementara membutuhkan energi termal. Fase material dan resistansi selanjutnya dapat dikontrol secara akurat dengan pulsa listrik, menjadikannya kandidat yang baik untuk elemen sinaptik non-volatil. Kombinasi kuat antara kepadatan dan kemampuan pemrograman in-situ dari teknologi ini merupakan kemajuan signifikan dalam desain perangkat keras neuromorfik.

**Integrasi Teknologi Baru Ke Dalam Proses CMOS**

Potensi sebenarnya dari teknologi memori baru ini terjadi ketika teknologi tersebut dapat diintegrasikan dengan proses CMOS standar. Biasanya, ini akan menjadi proses hibrida dengan perangkat memristor atau PCM yang dibuat di atas IC atau wafer CMOS yang sudah jadi (*back-end-of-line* (BEOL)). Dengan memungkinkan perancang untuk mempertahankan daya dan ketahanan CMOS untuk implementasi sirkuit neuron inti dan logika kontrol periferal, dan menggunakan material semikonduktor baru untuk mengimplementasikan susunan sinaptik non-volatil yang padat, ko-integrasi ini dapat dicapai. Integrasi pada node yang sama sulit, tetapi teknologi optimal dapat dikembangkan yang menggabungkan manfaat dari kedua struktur, fleksibilitas dan efisiensi komputasi CMOS, dan memori padat berdaya rendah dari perangkat yang sedang berkembang. Misalnya, susunan crossbar memristor dapat dibuat langsung di atas tahap terakhir CMOS dari IC untuk penguat neuron atau sirkuit generator spik. Ini akan memiliki panjang kabel minimum dan kapasitansi parasit dari sinapsis ke neuron, memastikan propagasi energi spike yang cepat dan efisien. Mengintegrasikan berbagai teknologi baru akan membuka potensi sistem komputasi yang benar-benar terinspirasi otak dan dapat diskalakan, yang dapat diproduksi dengan jalur fabrikasi semikonduktor yang ada

atau sedikit dimodifikasi. Tabel 5.1 menunjukkan sinapsis transistor CMOS dan hibrida (memristor atau PCM).

**Tabel 5.1** Sinapsis berbasis transistor CMOS sederhana versus sinapsis hibrida

| Karakteristik | Transistor CMOS | Hibrida (Memristor/PCM) |
|---|---|---|
| Luas Area (Area) | Membutuhkan area yang lebih besar | Implementasi lebih ringkas (*compact*) |
| Konsumsi Daya (Statis) | Konsumsi daya statis lebih tinggi | Konsumsi daya statis lebih rendah |
| Konsumsi Daya (Dinamis) | Memiliki kebutuhan daya dinamis tertentu | Memiliki karakteristik kebutuhan daya dinamis yang berbeda |
| Volatilitas | Penyimpanan data bersifat non-volatil (*non-volatile*) | Penyimpanan data bersifat volatil (*volatile*) |
| Kompatibilitas CMOS | Sepenuhnya kompatibel dengan teknologi CMOS | Memerlukan integrasi hibrida dengan CMOS |

## 5.4 DESAIN SIRKUIT DAN STRATEGI ARSITEKTURAL

### Tata Letak Sirkuit Baru untuk Pembelajaran dan Pemrosesan Sinyal

Kendaraan yang sukses untuk komputasi neuromorfik dalam sirkuit CMOS VLSI dibangun di atas tata letak sirkuit unik yang berbeda dari metodologi desain digital tradisional. Sementara prosesor tradisional telah dirancang untuk dioptimalkan untuk logika Boolean dan operasi aritmatika, sirkuit neuromorfik dirancang untuk belajar dan melakukan pemrosesan sinyal secara efisien. Tata letak analog sirkuit bukan hanya area; itu adalah bagian dari fungsionalitas. Jika aturan pembelajaran seperti *Spike-Timing-Dependent Plasticity* (STDP) akan diimplementasikan, maka tata letak sirkuit harus mendukung secara spasial hubungan antara spike prasinaptik dan postsinaptik.

Dalam kapasitas ini, ini berarti Anda biasanya menginginkan memori Anda (misalnya, memristor) diposisikan dekat tidak hanya dengan output neuron, tetapi juga dengan jalur yang dimasuki spike. Tata letak analog dan sinyal campuran juga harus cukup mencegah kebisingan dan kopling, yang jauh lebih bermasalah untuk desain sub-threshold. Tujuannya adalah untuk mendesain sirkuit yang kompak, hemat daya, dan tahan terhadap variasi proses dan kebisingan di lingkungan. Seringkali, ketika mendesain tata letak ini, penting untuk mengambil pendekatan sistem keseluruhan di mana kita mempertimbangkan sifat analog dan neuron, bersama dengan sifat memori-dalam-piksel dari sinapsis, dan mengoptimalkan kedua variabel tersebut bersama-sama. Secara umum, ini seringkali mengarah pada sirkuit yang sangat paralel dan tidak seragam dengan cara yang meniru interrelasi jaringan saraf biologis, daripada mengikuti pengaturan yang sangat teratur dari chip logika digital.

### Sistem Komputasi Terdistribusi Fungsional

Pendekatan desain penting untuk mencapai skalabilitas dan toleransi kesalahan dalam sistem neuromorfik adalah dengan menggunakan paradigma komputasi terdistribusi fungsional. Pendekatan ini menghindari titik pemrosesan tunggal dan sumber memori tunggal

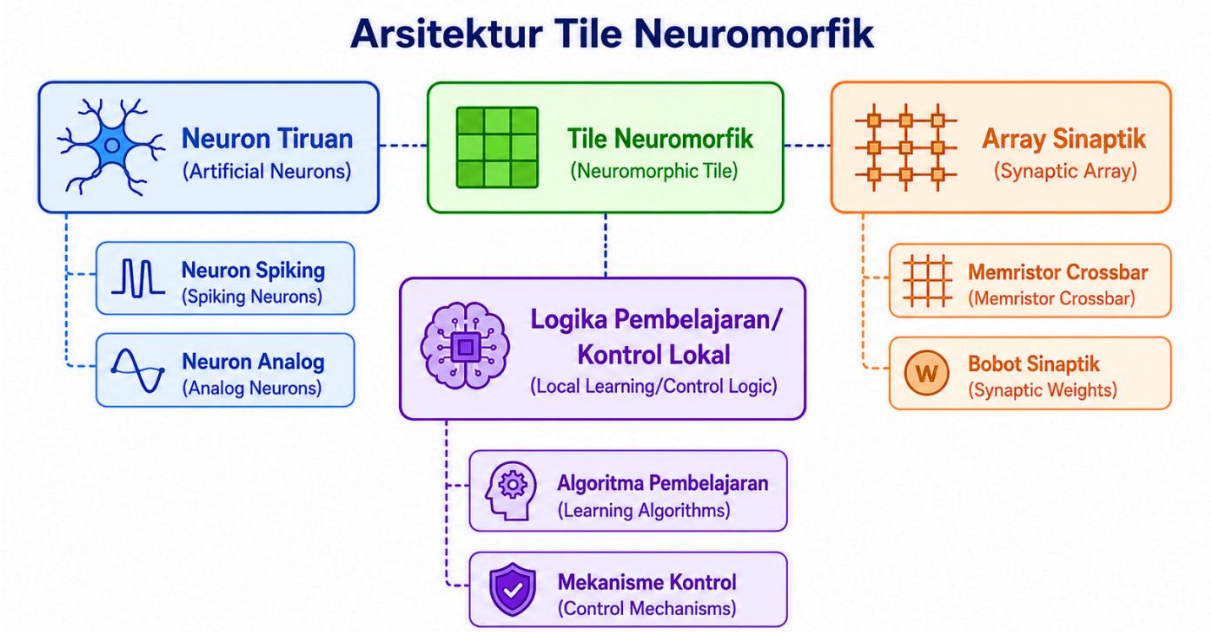
dari arsitektur von Neumann dengan menggantinya dengan model terdistribusi, dengan modul yang terhubung secara longgar. Sebuah chip neuromorfik biasanya terdiri dari banyak "ubin" atau "inti" yang mencakup populasi lokal neuron buatan, susunan sinaptik padat lokal, logika pembelajaran dan kontrol lokal.

Arsitektur ini memungkinkan alokasi semua tugas pemrosesan dan memori ke seluruh chip untuk beroperasi secara paralel masif, dan untuk menghilangkan biaya transfer data jarak jauh. Setiap lonjakan yang terbentuk di satu ubin dapat dialihkan ke sejumlah sinapsis di ubin lain oleh jaringan on-chip yang mendukung topologi jaringan yang lebih kompleks. Manfaat yang mungkin diperoleh dari komputasi terdistribusi sangat banyak. Jika terjadi kerusakan pada sebuah ubin, kerusakan tersebut dapat menyebabkan penurunan kualitas pengalaman secara bertahap. Sistem skala besar juga dapat dirancang dan diverifikasi lebih mudah dengan pendekatan terdistribusi, dan kompleks buatan sendiri. Toleransi kesalahannya diwujudkan dengan melakukan komputasi secara lokal dan berkomunikasi hanya jika benar-benar diperlukan, serta menghemat energi, karena hambatan pada sistem komputasi tradisional adalah pergerakan data.

**Perspektif Efisiensi: Pembelajaran dan Pemrosesan Sinyal**

Keberhasilan utama komputasi neuromorfik terletak pada kemampuan sistem untuk memberikan peningkatan efisiensi yang signifikan dibandingkan dengan sistem konvensional dalam hal daya dan kinerja. Untuk konstruksi efisiensi, sangat penting untuk membedakan efisiensi pembelajaran (adaptasi) dari efisiensi pemrosesan sinyal (inferensi). Sistem neuromorfik sangat efisien dari perspektif pemrosesan sinyal, mengingat bahwa suatu peristiwa (spike) hanya diproses ketika ada peristiwa untuk diproses, yang sangat baik untuk aliran data yang jarang. Ini sangat ampuh untuk aplikasi yang menggunakan sensor yang selalu aktif atau pemrosesan sensor waktu nyata. Lebih lanjut, komputasi yang terjadi di memori (dengan memristor atau array PCM) juga meningkatkan efisiensi dengan tidak perlu mentransfer semua data bolak-balik antara memori dan prosesor terpisah (transfer data yang mengkonsumsi energi yang signifikan).

Dalam hal pembelajaran, tujuan efisiensi utama adalah untuk dapat melakukan pembaruan bobot (biasanya pengurangan energi 100 ×) yang dibutuhkan, baik di dalam chip untuk pembaruan bobot seperti pembelajaran tanpa pengawasan dan pembaruan bobot pembelajaran dengan pengawasan dengan energi yang sangat sedikit. Konfigurasi banyak desain tata letak sirkuit baru untuk aturan pembelajaran unik seperti STDP memungkinkan bobot sinaptik diperbarui secara lokal dalam waktu nyata tanpa memerlukan prosesor off-chip terpisah atau data pelatihan yang melimpah. Arsitektur terdistribusi fungsional memungkinkan hal ini karena memungkinkan pemrosesan lokal. Gabungan dari jenis sirkuit baru ini dan arsitektur terdistribusi membentuk dasar untuk merancang sistem yang tidak hanya mampu memproses informasi, tetapi juga mampu belajar dan beradaptasi dengan daya dan otonomi yang sebelumnya dianggap sangat efisien. Gambar 5.2 menunjukkan arsitektur ubin neuromorfik terdistribusi.




**Gambar 5.2** Diagram blok dari satu ubin neuromorfik.

## 5.5 PEMANTAUAN ARSITEKTUR DAN PROFIL DAYA SNN

### Gambaran Umum Jaringan Saraf Spiking (SNN)

SNN, generasi ketiga dari model komputasi saraf, dikembangkan sebagai model yang mencakup karakteristik temporal neuron biologis. SNN tidak menggunakan aktivasi bernilai kontinu seperti Jaringan Saraf Buatan (ANN) konvensional, tetapi bergantung pada lonjakan listrik diskrit, memungkinkan komputasi asinkron dan berbasis peristiwa. Arsitektur ini memungkinkan komputasi dilakukan hanya ketika terjadi lonjakan yang berguna, dan oleh karena itu, mengkonsumsi daya dinamis minimal. Oleh karena itu, SNN menawarkan platform yang menarik untuk sistem neuromorfik hemat energi, terutama ketika melakukan tugas-tugas kecerdasan tepi dan pemrosesan sensorik berdaya rendah.

### Komponen Arsitektur Inti

SNN terdiri dari tiga komponen utama model neuron, sinapsis, dan topologi jaringan. Model neuron meliputi model potensial membran neuron dan penembakan Leaky Integrate-and-Fire (LIF), Izhikevich, dan Hodgkin-Helmholtz. Sinapsis adalah interaksi berbobot yang mentransmisikan sinyal dan plastisitas, dalam banyak kasus dikendalikan oleh hukum pembelajaran STDP. Baik transfer informasi maupun kompleksitas komputasi sistem ditentukan oleh topologi jaringan, yang dapat berupa feedforward, konvolusional, atau rekuren. Atribut struktural ini bersama-sama menentukan akurasi jaringan serta profil daya.

### Implementasi Perangkat Keras Neuromorfik

Biasanya, SNN diimplementasikan dalam perangkat keras pada arsitektur digital atau sinyal campuran berbasis peristiwa. Sistem neuromorfik berbasis teknologi CMOS, seperti TrueNorth IBM, dan Loihi Intel, menggunakan sirkuit asinkron untuk komputasi berbasis spike berdaya rendah. Selain perangkat CMOS, perangkat generasi berikutnya seperti memristor, FET ferroelektrik (FeFET), dan FET terowongan (TFET) sedang dikembangkan untuk akselerator SNN. Perangkat ini dapat memungkinkan bobot sinaptik non-volatil, beroperasi pada tegangan

rendah, dan mengkonsumsi sedikit energi switching, sehingga memungkinkan sistem yang dapat diskalakan yang menjembatani fisika tingkat perangkat ke komputasi neuromorfik tingkat tinggi.

**Konsumsi Daya dan Metrik Efisiensi**

Konsumsi daya dari SNN akan bergantung pada laju lonjakan dan bobot sinaptik yang digunakan, serta latensi komunikasi dan laju akses memori. Karena peristiwa lonjakan terjadi pada laju yang cukup rendah, pada satu waktu hanya sebagian kecil dari total jaringan neuron dan sinapsis yang mengonsumsi daya, sehingga laju daya paket dapat dikurangi secara substansial, dibandingkan dengan jaringan ANN berbasis frame, sementara masih mampu membuat tingkat inferensi yang sama. Sirkuit berbasis peristiwa juga membantu menghindari, sampai batas tertentu, aktivitas komputasi yang tidak perlu, sehingga menghasilkan daya statis dan dinamis yang lebih rendah. Metrik kinerja efisiensi umum dari sistem SNN meliputi energi/lonjakan (E/lonjakan), produk penundaan energi (EDP), dan *throughput/watt*. Hasil menunjukkan bahwa akselerator SNN yang dioptimalkan mengonsumsi satu hingga dua orde besaran lebih sedikit daripada akselerator SNN ANN yang setara dengan kinerja serupa yang diukur dalam akurasi.

**Tantangan Desain dan Strategi Optimasi**

Meskipun terdapat manfaat peningkatan efisiensi, masalah yang masih dihadapi oleh desain SNN kembali ditantang oleh ko-optimasi algoritma-perangkat keras, yang umum terjadi pada semua pembelajaran mesin. Pelatihan akurat dengan spike masih menjadi masalah terbuka karena 1) spike tidak dapat didiferensiasi dan 2) tergantung pada definisi backpropagation, kompatibilitas aturan pembelajaran terbatas. Selain itu, ketidaksempurnaan dalam perangkat keras (variabilitas perangkat, kesalahan kuantisasi, noise, dll.) berdampak pada stabilitas dalam sistem. Karya terbaru dalam SNN telah menyoroti optimasi lintas lapisan sebagai bagian dari mengatasi masalah ini yang terjadi pada tingkat algoritma, perangkat, dan sirkuit.

Arsitektur perangkat hibrida (yaitu, inti neuromorfik berbasis TFET atau FeFET) menyediakan ruang baru untuk fisika perangkat daya ultra-rendah dan mekanisme pembelajaran yang terinspirasi secara biologis untuk memenuhi tantangan SNN yang lebih dekat dengan sistem AI waktu nyata, skala besar, dan berkelanjutan. Tabel 5.2 menunjukkan fitur arsitektur penting dari SNN dan fitur konsumsi daya terkaitnya. Setiap tingkat desain (pemodelan neuron, implementasi sirkuit, dll.) dapat memengaruhi efisiensi secara berbeda. Lebih spesifiknya, tabel tersebut menunjukkan bagaimana sparsitas komputasi, pilihan perangkat, dan topologi arsitektur dapat berinteraksi untuk memengaruhi energi per lonjakan dan throughput, memberikan ringkasan yang bagus tentang pertimbangan desain sistem secara keseluruhan untuk komputasi neuromorfik berdaya rendah.

**Tabel 5.2** Ringkasan arsitektur dan profil daya Spiking Neural Network (SNN)

| Tingkat Desain | Fitur Penting | Mekanisme Penghematan Daya |
|---|---|---|
| Model Neuron | Lonjakan Diskrit (*Discrete Spikes*) | Daya hanya digunakan ketika terjadi peristiwa lonjakan (*spike*), bukan secara terus-menerus. |

| | | |
|---|---|---|
| | Sinapsis dan Pembelajaran Lokal (STDP) | Proses pembelajaran terjadi secara lokal pada sinapsis, sehingga mengurangi perpindahan data. |
| Struktur Jaringan | Konektivitas Lokal (*Local Connectivity*) | Energi yang digunakan untuk komunikasi jarak jauh berkurang karena pemrosesan dilakukan di dekat sumber data. |
| Perangkat Keras (Hardware) | Rangkaian Berbasis Peristiwa (*Event-Driven Circuits*) | Rangkaian tetap tidak aktif hingga terjadi lonjakan (*spike*), sehingga menghilangkan pemborosan daya saat kondisi siaga (*idle*). |
| | Perangkat Baru (*Novel Devices*, misalnya FeFET) | Memungkinkan operasi pada tegangan sangat rendah (*ultra-low voltage*), sehingga mengurangi konsumsi energi aktif maupun energi akibat kebocoran (*leakage energy*). |

## 5.6 STUDI KASUS DAN APLIKASI

**Implementasi CMOS yang Efektif untuk Sistem Neuromorfik**

Prinsip-prinsip teoritis komputasi neuromorfik telah diintegrasikan ke dalam beberapa implementasi CMOS utama untuk memberikan contoh konkret dari teknologi tersebut. TrueNorth IBM dan chip riset Loihi Intel adalah dua contoh yang paling terkenal. *TrueNorth* IBM, yang dirancang sebagai bagian dari proyek DARPA SyNAPSE, adalah chip berbasis peristiwa yang sangat skalabel yang berisi 1 juta neuron digital (yang diketahui terhubung secara independen) dan 256 juta sinapsis (koneksi potensial dari neuron ke neuron). Memiliki 4.096 inti yang tersusun dalam bentuk ubin, setiap inti merupakan unit komputasi lengkap yang memiliki neuron, sinapsis, logika perutean, dll. Ini adalah contoh komputasi neuromorfik yang efisien, karena merupakan teknologi digital yang menggunakan puluhan miliwatt sambil melakukan tugas kognitif kompleks seperti deteksi objek video waktu nyata.

Namun, Loihi Intel memiliki tingkat fleksibilitas yang lebih tinggi, karena inti neurologisnya memiliki inti SNN dan CPU x86 tradisional, yang diproduksi pada die yang sama. Loihi juga memiliki fabrik berbasis peristiwa yang sepenuhnya asinkron yang dapat mendukung berbagai jenis aturan pembelajaran on-chip, bahkan pembelajaran tanpa pengawasan. Berbeda dengan TrueNorth, yang lebih spesifik domain, Loihi adalah platform pembahasan yang dapat diprogram, memungkinkan eksperimen dengan berbagai algoritma. Secara keseluruhan, kedua chip tersebut merupakan pencapaian penting dalam menunjukkan bahwa arsitektur hemat daya yang terinspirasi otak dapat dibuat dengan proses CMOS VLSI canggih.

**Komputasi Neuromorfik Spesifik Aplikasi**

Fitur khas sirkuit neuromorfik membuatnya sangat menarik untuk berbagai aplikasi yang membutuhkan daya rendah dan operasi waktu nyata. Pemrosesan sensorik adalah area contoh di mana sistem ini dapat unggul, memproses aliran data yang diambil sampelnya secara terus menerus dan dipicu secara asinkron dari sensor seperti kamera dan mikrofon. Misalnya, dengan menggunakan perangkat seluler chip visi neuromorfik, informasi visual dapat diproses secara berbasis peristiwa, dan hanya ketika beberapa perubahan terdeteksi di bidang visual,

dan sebagai hasilnya pemrosesan lebih efisien daripada hanya memproses setiap piksel dari bingkai video setiap bingkai.

Sejauh sistem neuromorfik dibutuhkan untuk mendukung domain aplikasi, seperti pengenalan pola, sistem ini secara alami cocok untuk melakukan pengenalan suara dan pengenalan wajah, karena arsitektur paralel masifnya akan mendukung pencocokan cepat data sensorik yang masuk terhadap data pola yang dipelajari untuk inferensi dan navigasi. Dan karena komputasi waktu nyata dan hemat daya merupakan kasus penggunaan utama untuk sistem komputasi neuromorfik, sistem ini menawarkan platform ideal untuk menyediakan opsi ringan dan hemat daya bagi aktor otonom seperti robot dan drone. Misalnya, prosesor neuromorfik pada drone dapat berinteraksi dengan banyak sensor, membuat keputusan secara waktu nyata dengan latensi sangat rendah untuk menavigasi ruang angkasa dan menghindari rintangan menggunakan daya minimal, sehingga memperpanjang waktu terbang dan otonomi. Tabel 5.3 di bawah ini menunjukkan perbedaan antara prosesor neuromorfik dan prosesor konvensional.

**Tabel 5.3** Perbandingan prosesor neuromorfik vs. Konvensional

| **Fitur** | **Prosesor Neuromorfik** | **Prosesor Konvensional** |
|---|---|---|
| Model Arsitektur | **Spiking Neural Network (SNN)**, komputasi di dalam memori (*in-memory computing*) | Arsitektur **Von Neumann**, pemrosesan dan memori terpisah |
| Komunikasi | Asinkron, berbasis peristiwa (*event-driven*) menggunakan **spike** | Sinkron, berbasis **clock** menggunakan bus data (*data bus*) |
| Konsumsi Daya | Sangat rendah (*ultra-low*), berkisar dari miliwatt hingga watt | Tinggi, berkisar dari puluhan hingga ratusan watt |
| Paralelisme | Paralelisme sangat tinggi (ribuan inti pemrosesan) | Paralelisme tinggi (ribuan *shader cores* pada GPU) |
| Aplikasi Utama | AI waktu nyata (*real-time AI*), pemrosesan sensor, robotika | AI tujuan umum, grafika komputer, dan komputasi ilmiah |

## 5.7 ARAH MASA DEPAN DAN KESIMPULAN

### Arah Pembahasan dan Dampak yang Diantisipasi

Bidang komputasi neuromorfik masih relatif muda, dan tidak diragukan lagi, masih banyak pembahasan yang perlu dilakukan. Area fokus utama adalah mengembangkan algoritma pembelajaran on-chip yang lebih canggih dan andal yang memanfaatkan sifat unik perangkat keras neuromorfik. Hal ini akan mencakup, antara lain, penyelidikan aturan pembelajaran yang lebih masuk akal secara biologis, serta pengembangan pemetaan yang lebih efisien dari jaringan saraf dalam yang kompleks ke perangkat keras spiking. Selain mengembangkan algoritma baru, ada juga sejumlah perkembangan menarik dalam ilmu material dan eksplorasi berkelanjutan dari teknologi memori baru yang muncul di luar memristor dan PCM yang mungkin memberikan kepadatan dan kinerja yang lebih tinggi. Dalam area pertumbuhan ini, tantangan baru juga akan muncul untuk menggabungkan

material ini dengan proses fabrikasi CMOS untuk menghasilkan perangkat dan metodologi baru. Terakhir, konsekuensi dari pembahasan ini sangat menarik. Sebagai ilustrasi, terdapat kemungkinan untuk mempelajari perangkat edge yang otonom dan mampu belajar sendiri secara efektif serta sistem neuromorfik multi-chip skala besar yang mampu bersaing dengan perangkat keras, kemampuan, dan efisiensi otak manusia.

**Dampak pada Paradigma Komputasi Generasi Berikutnya**

Komputasi neuromorfik akan secara signifikan merevolusi masa depan komputasi. Ini akan menawarkan alternatif yang valid untuk paradigma arsitektur von Neumann dan mempopulerkan kategori baru yang disebut koprosesor neuromorfik yang dapat digunakan secara paralel dengan CPU dan GPU konvensional untuk melakukan beberapa tugas yang tidak mereka kuasai. Ini akan mengembangkan sistem komputasi hibrida yang mengambil yang terbaik dari kedua dunia: komputasi tujuan umum yang tersertifikasi, di satu sisi, dan efisiensi daya ekstrem serta pernyataan komputasi waktu nyata dari perangkat keras neuromorfik, di sisi lain. Ini akan menjadi pergeseran paradigma radikal dengan generasi baru akselerator AI yang terinspirasi otak yang bertujuan untuk menyediakan layanan AI waktu nyata di masa depan dan hadir dalam aplikasi smartphone dan IoT hingga pusat data raksasa.

**Masa Depan Solusi Komputasi Hemat Daya yang Terinspirasi Otak**

Bab ini telah membahas teori komputasi neuromorfik hingga praktik dalam implementasi sirkuit CMOS VLSI hemat daya, dan juga telah membahas blok bangunan paling sederhana dari neuron dan sinapsis buatan, serta telah memperkenalkan teknologi memori baru yang dapat digunakan untuk menghasilkan sistem praktis dan terukur. Kami juga telah menggambarkan pentingnya rencana arsitektur komputasi terdistribusi yang sangat penting untuk mengatasi kendala sumber daya yang berkorelasi dengan paradigma komputasi tradisional. Solusi yang terinspirasi otak secara inheren terkait dengan masa depan komputasi hemat daya. Dengan menjembatani kesenjangan saat ini antara proposal komputasi von Neumann klasik dan proposal komputasi yang lebih terinspirasi secara biologis, seseorang akan membawa komputasi ke tingkat sistem komputasi baru, yang pada kenyataannya dapat jauh lebih cepat, lebih kuat, dan akhirnya, jauh lebih hemat energi, cerdas, dan mampu belajar. Inovasi-inovasi ini akan menciptakan peluang baru bagi kecerdasan buatan dan membawa kita lebih dekat ke masa depan mesin yang dapat belajar dengan keanggunan dan kemampuan beradaptasi otak manusia.

# BAB 6
# PENGGANDA PERKIRAAN BERDAYA RENDAH BERTOLERANSI KESALAHAN

## 6.1 KOMPUTASI APROKSIMASI DALAM PENCITRAAN DAN AKSELERATOR AI

Laju inovasi dalam teknologi pencitraan digital telah merevolusi pengambilan, pemrosesan, dan analisis data visual. Perangkat keras pemrosesan gambar canggih diperlukan untuk menghadirkan aplikasi seperti diagnostik medis, citra satelit, jaringan pengawasan, realitas tertambah, streaming video waktu nyata, dan fotografi seluler, yang semuanya bergantung pada pemrosesan sejumlah besar data dalam waktu singkat dengan batasan daya yang sangat ketat. Hal ini sulit bagi para perancang karena mereka harus menyeimbangkan antara efisiensi dan daya komputasi. Dengan algoritma yang menjadi lebih canggih dan resolusi gambar yang meningkat, permintaan perangkat keras yang dibutuhkan meningkat secara signifikan. Pengenalan objek, segmentasi, dan deteksi objek mengandung banyak operasi aritmatika, terutama perkalian, yang sebagian besar menentukan biaya energi dan kompleksitas komputasi. Pengganda eksak, meskipun akurat, memiliki daya, penundaan, dan biaya area silikon yang berlebihan. Dengan pengurangan ukuran teknologi yang terus berlanjut hingga node nanometer yang sangat kecil, arus bocor dan aksi switching ditemukan memainkan peran yang semakin penting dalam kehilangan energi lebih lanjut, sehingga perangkat keras aritmatika yang presisi tidak lagi diperlukan dalam aplikasi pemrosesan gambar berdaya rendah.

Hal ini paling terlihat pada sistem pencitraan medis genggam yang menggunakan pengali halus untuk melakukan transformasi kompleks, yang menyebabkan konsumsi daya yang sangat besar, sehingga menyebabkan pendinginan termal yang berlebihan dan memperpendek masa pakai baterai perangkat bertenaga baterai. Akibatnya, kebutuhan akan struktur pemrosesan gambar yang efektif dan kompak telah diuraikan dengan eksplorasi paradigma desain baru yang konsekuen. Telah ada upaya untuk menemukan cara-cara di mana seseorang dapat secara selektif mengurangi akurasi tanpa menurunkan kualitas persepsi. Akibatnya, komputasi perkiraan telah muncul sebagai salah satu metode desain yang menjanjikan untuk memenuhi tantangan tersebut. Melalui pertukaran persyaratan presisi di area komputasi yang tidak kritis, desain sirkuit aritmatika perkiraan menunjukkan penghematan daya yang besar dan pertukaran kompleksitas perangkat keras, tanpa banyak kehilangan kualitas output.

Eksperimen pada pemrosesan gambar, termasuk perkalian, penghalusan, penajaman, dan penyaringan, membuktikan bahwa hasil visualnya dapat diterima. Terakhir, penilaian kualitas berdasarkan Mean Structural Similarity Index Measurement (MSSIM) menunjukkan bahwa kualitas persepsi gambar yang diproses tetap tinggi bahkan untuk aproksimasi. Hasil ini menunjukkan bahwa sedikit perbedaan karena desain yang tidak tepat biasanya tidak terlihat secara visual dan karenanya dapat berguna untuk aplikasi yang toleran terhadap kesalahan.

Selain pemrosesan gambar, pengali yang disajikan juga menunjukkan potensi besar untuk akselerator jaringan saraf. Jutaan operasi perkalian dan penjumlahan terdapat dalam jaringan saraf, yang mendominasi daya dan penundaan saat melakukan pelatihan dan inferensi. Karena sifat statistik dan redundan dari jaringan saraf, jaringan saraf dapat tetap kuat terhadap beberapa kesalahan komputasi kecil, dan pengali aproksimasi dapat dengan mudah diintegrasikan. Mengganti pengali tepat dengan pengali aproksimasi yang diusulkan berpotensi memberikan peningkatan revolusioner dalam efisiensi energi dan kecepatan dengan sedikit pengorbanan dalam akurasi.

Singkatnya, karya ini memberikan tiga kontribusi signifikan:

- Menyajikan pengali perkiraan berukuran kecil dan hemat daya yang diturunkan dari kompresor 4:2 yang dimodifikasi dan kompresi produk parsial hibrida.
- Menyajikan evaluasi menyeluruh terhadap tugas pemrosesan gambar dan menunjukkan bahwa kualitas persepsi hampir tidak dapat dibedakan dari aslinya di bawah perkiraan.
- Mendorong penerapan desain yang lebih luas pada perangkat keras jaringan saraf, di mana hal itu dapat memfasilitasi pembangunan akselerator pembelajaran mesin yang lebih cepat dan hemat energi.

Secara keseluruhan, bab ini menunjukkan bahwa komputasi perkiraan adalah pendekatan yang layak untuk mengoptimalkan sirkuit aritmatika dalam sistem digital kontemporer, terutama ketika presisi ideal tidak diperlukan.

**Peran Pengali Perkiraan dalam Pemrosesan Gambar yang Efisien**

Perkalian adalah operasi aritmatika yang paling umum dalam sejumlah besar pemrosesan gambar dalam proses peningkatan, penyaringan, dan pengeditan gambar. Untuk memaksimalkan kinerja operasi yang membutuhkan banyak komputasi ini, para peneliti telah menyelidiki banyak desain pengali perkiraan, yang dirancang agar cepat dan kurang intensif energi daripada desain eksak tradisional. Semua pengali ini secara khusus memperkenalkan kesalahan perhitungan kecil yang tidak dapat dilihat oleh mata pada gambar yang dihasilkan karena sistem penglihatan manusia juga dapat menangani kesalahan kecil tanpa dianggap memiliki kualitas yang dirasakan secara signifikan berkurang. Oleh karena itu, pengali perkiraan secara khusus diadaptasi untuk aplikasi yang perlu tahan terhadap kesalahan, termasuk pemrosesan model gambar dan video, di mana presisi dalam angka tidak sepenting kualitas visual.

Salah satu teknik yang paling umum digunakan untuk mendesain pengali ini adalah penggunaan kompresor perkiraan, yang lebih mudah dikembangkan dan kurang rumit daripada tugas mengakumulasi parsial produk. Logika perkiraan sering kali diterapkan secara selektif pada bit paling tidak signifikan (LSB) dari sebuah array bit dalam arsitektur hibrida, di mana efek visual dari kesalahan komputasi paling kecil, tetapi dipertahankan pada bit paling signifikan (MSB), di mana akurasi yang dibutuhkan harus konsisten. Desain hibrida ini menawarkan kompromi terbaik antara akurasi dan efisiensi, yang memungkinkan penghematan daya dan area yang cukup besar tanpa mengurangi kualitas gambar.

Telah diamati dalam studi simulasi dan implementasi perangkat keras bahwa pengali perkiraan dalam banyak hal jauh lebih cepat, lebih hemat energi, dan mengkonsumsi area silikon yang lebih sedikit daripada rekan-rekannya. Hasil yang dilaporkan menunjukkan pengurangan yang signifikan dalam parameter sirkuit, dan karenanya cocok untuk diintegrasikan ke dalam sistem kecil dan hemat daya. Efisiensi energi seperti ini lebih penting, terutama dalam sistem pencitraan tertanam, prosesor seluler, dan produk elektronik genggam, di mana masa pakai baterai, pembuangan panas, dan stabilitas sistem secara langsung dipengaruhi oleh efisiensi energi perangkat ini. Secara umum, penggunaan pengali perkiraan merupakan tren yang menjanjikan dalam desain perangkat keras digital saat ini, dan memungkinkan komputasi yang efisien dan berkecepatan tinggi untuk tugas pemrosesan gambar dan video secara real-time.

**Integrasi dengan Arsitektur Jaringan Saraf Baru**

Kegunaan pengali perkiraan kini telah jauh melampaui aplikasi pemrosesan gambar tradisional dan merupakan fasilitator utama dalam berbagai arsitektur komputasi yang lebih canggih. Secara historis, pengali merupakan elemen tradisional dari sistem digital, terutama di mana aplikasi sinyal dan gambar perlu melakukan sejumlah besar komputasi pada tingkat piksel. Namun demikian, munculnya teknologi Kecerdasan Buatan (AI) dan Pembelajaran Mesin (ML), dan khususnya pada perangkat berbasis gambar seperti pengenalan objek, klasifikasi gambar, dan visi komputer, telah meningkatkan kebutuhan komputasi hingga satu tingkat besaran. Jaringan saraf paling umum yang membentuk inti sistem AI menggunakan operasi tipe perkalian dan penjumlahan (MAC) yang dieksekusi melalui beberapa lapisan yang saling terhubung. Ketika dikodekan dengan pengali tepat, operasi ini membutuhkan banyak daya dan sumber daya perangkat keras, dan ini telah menjadi masalah dalam hal efisiensi daya, luas silikon, dan kecepatan pemrosesan, terutama pada perangkat edge dan portabel.

Dalam upaya untuk mengatasi tantangan tersebut, komputasi aproksimasi telah menjadi bidang pembahasan yang lebih disukai, dengan para peneliti jelas mengorbankan sedikit akurasi komputasi untuk mencapai kinerja dan pengurangan energi yang besar. Di bawah paradigma ini, pengali aproksimasi telah muncul sebagai salah satu solusi yang paling menjanjikan. Pengali ini memberikan ketidakakuratan yang disengaja dalam komputasi aritmatika, dan ini dapat diterima mengingat banyaknya algoritma AI dan pemrosesan gambar, di mana akurasi yang ketat tidak selalu menjadi persyaratan. Sumber kedua dari jaringan saraf, khususnya, adalah bahwa mereka toleran terhadap kesalahan karena dilatih untuk belajar, dan mereka berbasis statistik. Kesalahan kecil dalam operasi MAC tidak akan memiliki peran serius dalam akurasi akhir inferensi karena model dapat menyesuaikan dan mengkompensasi selama periode pembelajaran. Kualitas inilah yang membuat jaringan saraf sangat tepat dalam proses mengadopsi pengali aproksimasi, karena ada komputasi yang efisien, meskipun kualitas output dan akurasi peramalan tidak menurun secara signifikan.

Penggunaan pengali aproksimasi dalam akselerator saraf telah terbukti memiliki sejumlah keuntungan yang signifikan. Penurunan konsumsi energi adalah salah satu manfaat yang paling menonjol. Karena akselerator jaringan saraf mengalikan dan membagi data miliaran atau bahkan jutaan kali setiap detik, optimasi suatu operasi dapat menghasilkan

pengurangan daya total yang signifikan. Hal ini dilakukan oleh pengali perkiraan, yang menyederhanakan pembangkitan produk parsial, pohon kompresor, dan jalur propagasi carry, yang pada gilirannya mengurangi aktivitas switching dan jumlah transistor. Mereka juga digunakan untuk mencapai peningkatan throughput komputasi, yang mengurangi jalur kritis serta meminimalkan penundaan propagasi. Akibatnya, individu yang merancang arsitektur menggunakan pengali perkiraan dapat mencapai frekuensi clock yang lebih tinggi atau tegangan suplai yang lebih rendah, yang berarti mereka menghasilkan peningkatan efisiensi kinerja per watt – pertimbangan penting lainnya saat menerapkan AI edge.

Keuntungan signifikan lainnya adalah area yang lebih kecil, sehingga lebih sedikit silikon yang dibutuhkan. Dengan persyaratan ketebalan yang longgar, pengali perkiraan mampu menghilangkan banyak logika redundan dan mengurangi kompleksitas sirkuit. Ini tidak hanya mengurangi biaya manufaktur tetapi juga memungkinkan sumber daya perangkat keras tambahan untuk dimasukkan dalam area chip yang sama, memungkinkan lebih banyak fungsi untuk disertakan. Hal ini memungkinkan perancang sistem untuk mendapatkan lebih banyak paralelisme dan throughput yang lebih baik dengan batasan anggaran daya dan area. Keuntungan ini sangat berguna dalam sistem kecil seperti ponsel pintar, drone terbang otonom, sensor kesehatan yang dapat dikenakan, dan perangkat edge IoT, di mana ukuran kecil dan konsumsi daya akan diperlukan.

Pengali perkiraan telah terbukti secara eksperimental dapat diterapkan dalam berbagai sistem AI berbasis gambar. Pengali perkiraan digunakan dalam lapisan konvolusional dan terhubung penuh dalam jaringan saraf konvolusional (CNN), yang merupakan yang paling umum dan banyak digunakan untuk melakukan pengenalan visual. Pembahasan telah menunjukkan bahwa akurasi klasifikasi hampir sama dengan pengali eksak ketika aritmatika tidak tepat digunakan. Perbedaan persepsi antara keluaran visual yang diperoleh dalam tugas restorasi gambar, deteksi objek, dan segmentasi dengan komputasi langsung telah diamati tidak berbeda. Fakta ini membuktikan kesesuaian aritmatika perkiraan untuk sistem visual yang digerakkan oleh AI. Pengali perkiraan dapat digunakan dalam kendaraan otonom dan robotika untuk memberikan waktu respons yang lebih cepat dengan kebutuhan energi yang rendah, karena pemrosesan data gambar dan sensor secara real-time diperlukan. Komputasi yang dihasilkan dengan konsumsi energi yang berkurang memungkinkan masa pakai baterai yang lebih lama dan bahkan keandalan yang lebih besar dari sistem pengambilan keputusan otonom. Dalam hal yang sama, ketika menyangkut perangkat seluler dan perangkat yang dapat dikenakan, kombinasi pengali perkiraan memungkinkan implementasi kemampuan AI seperti pengenalan wajah, pengenalan gerakan, dan realitas tertambah pada tingkat konsumsi daya yang rendah. Keunggulan praktis tersebut telah menjadikan pengali perkiraan sebagai blok perangkat keras yang penting dari platform cerdas INT generasi berikutnya.

Selain itu, pengenalan pengali perkiraan telah mengungkap arah pembahasan baru dalam pengoptimalan bersama perangkat keras-perangkat lunak. Para perancang saat ini sedang mempertimbangkan sistem presisi hibrida yang dapat merespons secara dinamis dalam hal tingkat perkiraan sebagai respons terhadap sensitivitas beban kerja. Sebagai contoh, bagian dari jaringan saraf yang kritis dan harus memengaruhi presisi keluaran mungkin

menggunakan nilai pengali yang hampir tepat, sedangkan komponen yang tidak kritis mungkin menggunakan nilai pengali yang sangat tidak tepat. Hal ini menjadikan pendekatan adaptif sebagai keseimbangan antara akurasi dan efisiensi energi yang cukup fleksibel untuk meningkatkan kinerja pada tingkat sistem. Secara keseluruhan, penggunaan pengali perkiraan dalam akselerator AI dan ML merupakan pergeseran paradigma dalam desain perangkat keras, yaitu, tidak lagi berfokus pada akurasi dalam desain perangkat keras tetapi berfokus pada efisiensinya.

Konsumsi daya yang mendekati, kemampuan untuk melakukan komputasi dengan kecepatan tinggi dengan daya dan area yang paling sedikit menjadikannya penting untuk jenis aplikasi yang dibatasi daya saat ini yang membutuhkan throughput tinggi. Tergantung pada apakah kita berurusan dengan pencitraan cerdas, analitik video nyata, atau sistem AI tingkat tertanam, pengali perkiraan merupakan inovasi radikal. Mereka mengurangi redundansi komputasi dan mempertahankan kualitas output peralatan, mengantarkan generasi berikutnya dari arsitektur perangkat keras cerdas dan hemat energi yang digunakan dalam teknologi berbasis AI. Mereka diprediksi akan terus menyebar di smartphone, sistem otonom, dan perangkat wearable, yang hanya akan mendukung tujuan mereka sebagai dasar untuk pengembangan platform komputasi yang efisien dan terukur.

**Metodologi Desain dan Aplikasi Dunia Nyata**

Upaya terkini dalam desain pengali aproksimasi bertujuan untuk meningkatkan operasi pemrosesan dalam pengolahan citra sehubungan dengan kemampuannya untuk mengeksekusi fungsi lebih cepat dengan efisiensi energi yang lebih baik. Pendekatan presisi adaptif tipe kompresor dan metode telah diadopsi untuk mengembangkan kompromi optimal antara akurasi dan kinerja. Dalam metode ini, kompleksitas perangkat keras yang digunakan telah diminimalkan secara signifikan dengan peningkatan kecepatan perhitungan yang besar. Ketika diproyeksikan ke dunia nyata pengolahan citra, pengali aproksimasi telah ditemukan berkinerja lebih baik dalam hal efek visual yang tidak dapat dibedakan dari yang diperoleh oleh operasi aritmatika paling murni.

Ukuran evaluasi populer, termasuk Rasio Sinyal-ke-Derau Puncak (PSNR) dan Indeks Kesamaan Struktural Rata-rata (MSSIM), selalu memvalidasi pemeliharaan kualitas citra visual dalam gambar yang diproses. Selain itu, kontrol presisi adaptif telah dimasukkan dalam beberapa arsitektur canggih, memungkinkan tingkat kontrol yang akurat tergantung pada tuntutan dinamis aplikasi. Fleksibilitas ini akan mencapai penghematan energi tanpa mengorbankan fidelitas keluaran yang kritis. Penerapan pengali perkiraan berkinerja tinggi telah meluas ke banyak bidang, seperti pengenalan gambar, deteksi objek, peningkatan video, dan keamanan multimedia. Berdasarkan efisiensi, kekompakan, dan ketahanannya, desain tersebut diperkirakan menjadi pusat pengembangan sistem elektronik berkinerja tinggi dan berdaya rendah generasi berikutnya serta sistem pencitraan tertanam.

**Berlayar Menuju Cakrawala Masa Depan**

Dalam prospeknya, pengali perkiraan berbutir kasar diharapkan dapat mencapai peningkatan luar biasa dalam kinerja, skalabilitas, dan efisiensi daya. Aktivitas pembahasan saat ini diarahkan pada pengembangan arsitektur kompresor yang lebih canggih, dan

mekanisme koreksi kesalahan tingkat tinggi juga diintegrasikan untuk lebih meningkatkan akurasi tanpa mengorbankan penghematan energi. Pada saat yang sama, banyak kemajuan dalam komputasi neuromorfik, di mana model didasarkan pada aturan otak manusia, dimanfaatkan untuk menciptakan unit pemrosesan gambar berdaya sangat rendah generasi berikutnya yang dapat beradaptasi secara cerdas dan belajar sendiri. Terobosan terbaru dalam metode fabrikasi baru, terutama dalam desain tumpukan 3D, desain tegangan mendekati ambang batas, dan desain chip yang dikurangi, juga akan memainkan peran penting dalam peningkatan kinerja dan pengurangan biaya desain. Telah diajukan usulan untuk memperkenalkan kerangka kerja benchmarking standar yang memungkinkan kemampuan untuk menilai berbagai arsitektur pengali perkiraan secara setara, membandingkannya dengan cara yang sama, sehingga memastikan evaluasi yang adil, dan menemukan peluang optimasi desain.

Selain itu, teknik eksplorasi dan optimasi desain berbasis AI kemungkinan akan memiliki pengaruh signifikan pada pengembangan pengali perkiraan di masa depan. Dengan menggunakan sintesis sirkuit yang dibantu AI dan analisis trade-off akurasi-daya otomatis, arsitektur yang lebih efisien dan cerdas akan dapat dicapai, dengan pengali perkiraan menjadi komponen desain yang tak tergantikan dalam sistem komputasi hemat energi berkinerja tinggi yang baru.

## 6.2 PENGALI PERKIRAAN YANG DIUSULKAN

Persyaratan yang semakin meningkat dari paradigma komputasi berkinerja tinggi berdaya rendah telah dicatat untuk mendorong perhatian yang cukup besar dalam pembahasan di bidang pencapaian implementasi aritmatika perkiraan. Toleransi kesalahan yang kecil biasanya diperbolehkan dalam sistem multimedia, seperti aplikasi pemrosesan gambar atau lingkungan ML. Rangkaian presisi berulang, misalnya, adder dan kompresor, dikenal menghasilkan hasil yang presisi, tetapi memiliki kompleksitas perangkat keras dan konsumsi daya yang tinggi. Untuk mengatasi kekurangan ini, komputasi aproksimasi telah ditunjukkan sebagai metode desain yang efektif, di mana sedikit toleransi dalam akurasi ditoleransi untuk mendapatkan peningkatan yang signifikan dalam luas area, daya, dan penundaan.

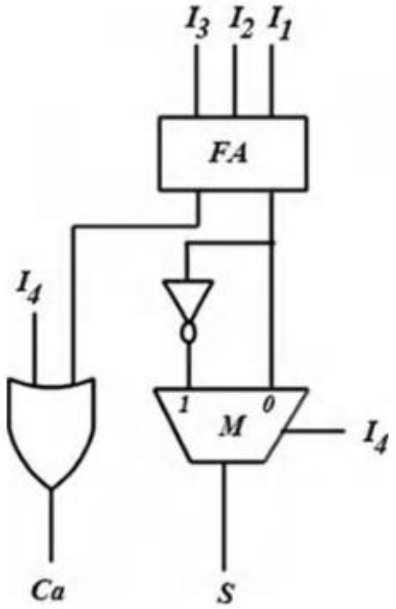
Blok bangunan pengali aproksimasi:

- Kompresor 4:2 yang dimodifikasi
- Penambah Penuh Aproksimasi (AFA)
- Unit Pembangkitan Jumlah (SG)

**Desain Kompresor 4:2 yang Dimodifikasi**

Rangkaian ini telah berupaya menyederhanakan ekspresi Boolean dan mengurangi jumlah gerbang dan overhead transistor, dengan hanya kesalahan komputasi kecil yang meluas ke dalamnya. Bit produk yang kurang penting dihitung dengan operasi logika aproksimasi, sedangkan bit yang paling penting dihitung dengan operasi logika presisi, sehingga menciptakan keseimbangan antara efisiensi energi dan akurasi komputasi dengan trade-off optimal. Pendekatan jenis ini telah terbukti sangat tepat untuk pencitraan secara real-time

dengan toleransi kesalahan tertentu. Kompresor 4:2 yang dimodifikasi telah diperkenalkan untuk meningkatkan efisiensi pengali, seperti yang ditunjukkan pada Gambar 6.1. Tabel kebenaran dari desain khusus tersebut ditabulasi dalam Tabel 6.1. Desain arsitektur sangat berfokus pada pengurangan kedalaman logika, daya, dan penundaan propagasi dalam rentang akurasi yang dapat diterima. Kompresor 4:2 yang dimodifikasi dikembangkan berdasarkan adder penuh (FA), di mana tiga bit input diproses dengan bantuan multipleks untuk memilih antara jumlah FA dan komplemennya, sehingga memungkinkan overhead perangkat keras minimal dan memberikan pemrosesan yang lebih cepat.




**Gambar 6.1** Desain kompresor 4:2 yang dimodifikasi.

**Tabel 6.1** Tabel kebenaran desain 4:2 yang dimodifikasi

| **Masukan** | | | | **Keluaran** | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | *Presisi* | | | *Tidak Presisi* | | |
| $I_4$ | $I_3$ | $I_2$ | $I_1$ | $Co$ | $Ca$ | $S$ | $Ca$ | $S$ | $E$ |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 2 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | -2 |

**Manfaat utama:**

- Pengeluaran logika minimum, yang juga mengurangi disipasi daya. Ekspresi logika kompresor yang ada dan yang diusulkan disorot dalam Persamaan (6.1–6.5).
- Desain yang berjalan lebih cepat karena overhead perangkat keras yang lebih rendah.
- Akurasi yang wajar, yang sesuai untuk realisasi pengali perkiraan dalam aplikasi tahan kesalahan seperti pemrosesan gambar.
- Dengan blok bangunan aritmatika perkiraan dan pedoman desain ini, para peneliti dapat merancang sirkuit aritmatika perkiraan berkinerja tinggi dan andal untuk berbagai aplikasi.

Ekspresi logika kompresor 4:2 yang tepat

$$Sum = X_1 \oplus X_2 \oplus X_3 \oplus X_4 \oplus C_{in} \quad (6.1)$$

$$C_{out} = (X_1 \oplus X_2) \cdot X_3 + \overline{(X_1 \oplus X_2)} \cdot X_1 \quad (6.2)$$

$$Carry = (X_1 \oplus X_2 \oplus X_3 \oplus X_4) \cdot X_{in} + \overline{(X_1 \oplus X_2 \oplus X_3 \oplus X_4)} \cdot X_4 \quad (6.3)$$

Ekspresi logis dari kompresor 4:2 yang dimodifikasi

$$S = (I_2 \odot I_1)(I_4 \oplus I_3) + (I_2 \oplus I_1)(I_4 \odot I_3) \quad (6.4)$$

$$Ca = I_4 + I_3 I_2 + I_1(I_3 + I_2) \quad (6.5)$$

Berdasarkan persamaan tersebut, dapat diamati dengan jelas bahwa desain yang dimodifikasi membutuhkan lebih sedikit komponen. Akibatnya, kompleksitas perangkat keras dan konsumsi daya berkurang.

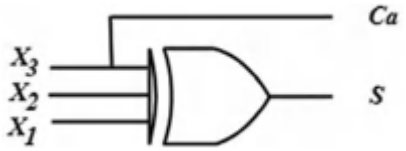
Kompresor 4:2 juga telah dioptimalkan agar sesuai dengan struktur pengali dengan mengonsumsi lebih sedikit daya dan penundaan, tetapi tetap menghasilkan hasil yang wajar. FA dalam kompresor 4:2 yang dimodifikasi bekerja dengan tiga bit input dengan bit input keempat ($I_4$) yang disertakan dalam jalur pembangkitan carry dan sum untuk menyederhanakan desain. Output carry dihasilkan dari operasi OR untuk memberikan propagasi yang kedap. Multiplexer digunakan dalam desain ini, dan jalur seleksinya adalah I4. Aktivitas switching telah dikurangi, dan telah diamati bahwa logika disederhanakan dibandingkan dengan kompresor tipikal (tepat). Akibatnya, kompresor 4:2 dalam proposal ini menawarkan persyaratan perangkat keras dan kecepatan yang lebih sedikit, dan dapat digunakan dalam desain pengali perkiraan yang dibutuhkan oleh komputasi seperti pemrosesan gambar, di mana toleransi kesalahan kecil dimungkinkan. Desain Pengali 8 × 8

Perkiraan dapat dioptimalkan lebih lanjut menggunakan teknologi CMOS mutakhir dan material baru. Peningkatan besar dalam efisiensi daya, serta kinerja.

Fleksibilitas arsitektur ini membuatnya dapat diterapkan pada aplikasi yang lebih luas untuk digunakan dalam kompresi data, ML, dan enkripsi. Desain kompresor 4:2 yang ditingkatkan merupakan jawaban yang layak untuk perangkat seluler hemat energi dan pusat data berkinerja tinggi, berkat skalabilitas dan fleksibilitasnya. Membangun sirkuit berbasis aritmatika perkiraan sebagai kompresor 4:2 berdasarkan konstruksi yang diusulkan telah dikuantifikasi sebagai salah satu pendorong adopsi massal komputasi perkiraan. Sirkuit ini telah memainkan peran penting dalam memfasilitasi komputasi hemat energi dengan menawarkan pemahaman tentang pertimbangan desain dan juga menawarkan kemungkinan penggunaan.

**Desain Penambah Penuh Perkiraan**

Gambar 6.2, diagram blok Penambah Penuh Perkiraan (AFA) adalah blok logika yang digunakan dalam proses perkiraan perhitungan target, yang merupakan blok logika kombinasi. Tabel kebenaran dari desain khusus tersebut ditabulasi dalam Tabel 6.2. Kombinasi tiga variabel input $(X_1, X_2, \text{dan } X_3)$ digunakan dalam jaringan logika yang disederhanakan yang menghasilkan dua nilai output, yaitu S (output jumlah) dan Ca (output tambahan).




**Gambar 6.2** Desain AFA.

**Tabel 6.2** Tabel kebenaran desain AFA

| $X_3$ | $X_2$ | $X_1$ | $C_a$ | $S$ | $C_a$ | $S$ | $E$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | -2 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 2 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

Output tambahan $Ca$ berbanding lurus dengan salah satu input ($X_3$), sedangkan $S$ diperoleh sebagai hasil perkalian ketiga input. Desain AFA ini memungkinkan fungsinya sebagai penambah parsial, yang memberikan perkiraan jumlah dan sinyal carry atau kontrol ke tahap lain. AFA menghemat area, aktivitas switching, dan disipasi daya yang cukup besar dengan meminimalkan kompleksitas logika dan mempertahankan tingkat kesalahan komputasi yang dapat diterima. Kompleksitas yang dikurangi telah dikembangkan untuk meningkatkan kinerja

sistem berdaya rendah dan telah diidentifikasi sebagai sesuai untuk pemrosesan gambar dan aplikasi ML, di mana mesin tidak perlu sangat akurat. AFA juga dapat dikombinasikan dengan inti komputasi perkiraan lainnya, termasuk kompresor 4:2 yang dimodifikasi atau metode penghematan daya, termasuk penskalaan volume.

Dengan struktur pengali yang besar, ia juga dapat digunakan untuk memungkinkan struktur campuran, dengan logika kasar digunakan untuk mengeksekusi bit dengan signifikansi lebih rendah dan logika halus digunakan untuk mengeksekusi bit dengan signifikansi lebih tinggi. Inilah tingkat fleksibilitas yang memungkinkan AFA digunakan dalam aplikasi seperti pemrosesan gambar, akselerator jaringan saraf, kompresi data, enkripsi, dan pemrosesan sinyal waktu nyata.

Penjelasan Tabel Kebenaran:

- Tabel Kebenaran untuk AFA dengan input $X_3, X_2, X_1$, dan output $Ca, S,$ dan $E$. Dua kelompok kolom untuk $Ca$ dan $S$ mungkin terkait dengan dua operasi berbeda atau dua langkah berbeda untuk mendapatkan output $Ca$ dan $S$.
- Input: Kombinasi $X_3, X_2, X_1$ (masing-masing 0 atau 1).
- Output:
  - $Ca$: Output carry.
  - $S$: Output sum.
  - $E$: Output error, dengan nilai 0, -2, atau 2 dalam kasus tertentu.

Mengamati pola:

- E bukan nol hanya untuk kombinasi input $X_3 = 1, X_2 = 0\ (E = 2)$ atau $X_3 = 0, X_2 = 1, X_1 = 1\ (E = -2)$.
- Rangkaian kolom $Ca$ dan S kedua muncul untuk menyoroti nilai keluaran kunci (seperti 1 atau 0) untuk kombinasi masukan tertentu.

**Unit pembangkit penjumlahan**

Pengoperasian unit HA dan SG menghasilkan penjumlahan dari dua variabel masukan ($X_1$ dan $X_2$). Dalam tiga dari empat kemungkinan kombinasi masukan (00, 01, dan 10), unit SG dapat bertindak dengan cara yang sama seperti HA tradisional, yang menghasilkan hasil tanpa kesalahan. Struktur SG yang disederhanakan telah menghilangkan kebutuhan untuk menghasilkan logika carry; oleh karena itu, hal ini mengurangi kompleksitas perangkat keras dan hanya menimbulkan kesalahan kecil yang dapat dikelola.

- Pengecualian terjadi untuk $X_1X_2$ = 11:
  - HA: Output jumlah = 0, Output carry = 1
  - Unit SG: Output jumlah = 1, Tidak ada output carry (Faktor kesalahan $E = -1$)

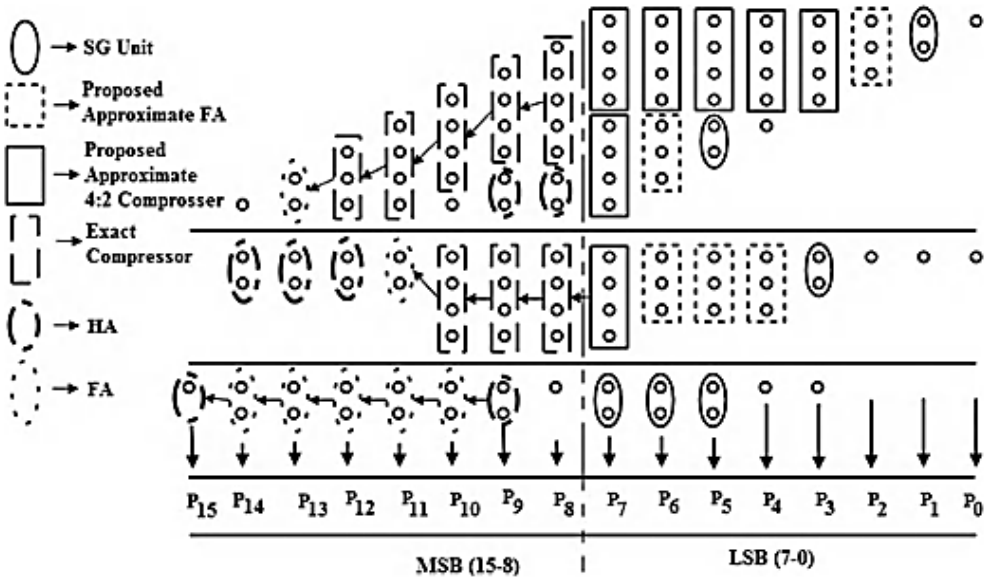
Penggantian HA dalam kasus unit SG (Gambar 6.3) telah terbukti memudahkan rangkaian, mengurangi konsumsi daya, dan mengurangi penundaan propagasi. Penghapusan jalur pembangkitan carry mengurangi aktivitas switching, terutama dalam aplikasi daya rendah. Mengingat bahwa pemrosesan gambar dan aplikasi jaringan saraf dapat menangani kesalahan kecil, unit SG dapat berhasil digunakan untuk menggantikan HA tradisional, memberikan keuntungan signifikan dalam kesederhanaan, efisiensi, dan kecepatan.




**Gambar 6.3** Bagian SG.

**Arsitektur Pengali 8 × 8 yang Diusulkan**

Pengali 8 × 8 yang diusulkan, dan perhitungan berdasarkan Gambar 6.4, menggunakan metode komputasi perkiraan yang mengurangi konsumsi daya serta penundaan dan kompleksitas perangkat keras, tetapi dengan kesalahan yang dapat diterima. Pengganda 8 × 8 yang diusulkan (lihat Gambar 6.4) menggabungkan beberapa jenis blok, termasuk SG, AFA, kompresor 4:2 yang dimodifikasi, dan blok "tepat".




**Gambar 6.4** Desain pengali 8 × 8 yang diusulkan.

Penerapan blok dalam arsitektur hibrida ini mengikuti prinsip berdasarkan prioritas relatif yang diberikan pada nilai yang dihitung. Secara khusus, bit prioritas rendah $P_0 - P_7$ diproses melalui unit perkiraan untuk meminimalkan konsumsi daya dan kebutuhan area silikon, sedangkan bit prioritas tinggi $P_8 - P_{15}$ diproses melalui unit akurat untuk memberikan tingkat presisi yang cukup pada hasil perhitungan. Prinsip ini berlaku untuk berbagai aplikasi, termasuk pemrosesan gambar, ML, pemrosesan multimedia, dll., yang dapat mentolerir beberapa tingkat kesalahan dalam hasil komputasi yang dilakukan. Secara keseluruhan, arsitektur pengali 8 × 8 yang diusulkan memungkinkan keseimbangan antara efisiensi dan akurasi komputasi yang dilakukan dan mengurangi jumlah sakelar dan kompleksitas logika yang dibutuhkan. Gambar 6.4 mengilustrasikan dengan baik kompromi yang dicapai dalam hal

konsumsi daya, area silikon, dan waktu komputasi untuk setiap komponen dari arsitektur pengali 8 × 8 yang diusulkan.

Dengan demikian, arsitektur pengali 8 × 8 yang diusulkan sangat cocok untuk perangkat medis genggam, sistem visi tertanam, prosesor seluler, dan desain VLSI lainnya yang memiliki anggaran daya terbatas. Bab ini membahas desain dan evaluasi rangkaian aritmatika perkiraan, di mana AFA, kompresor 4:2 yang dimodifikasi, dan unit pembangkitan penjumlahan (SG) telah dipertimbangkan secara khusus dan diterapkan pada pengali perkiraan 8 × 8 yang dikembangkan. Hasil yang diperoleh dari tabel kebenaran menunjukkan bahwa desain AFA dan 4:2 yang dimodifikasi memberikan hasil yang memang sangat dekat dengan nilai yang benar untuk semua kondisi input.

Kesalahan kecil yang dihasilkan, kemudian, dalam rangkaian jenis perkiraan ini, dapat diterima untuk sistem apa pun yang mampu mentolerir sedikit ketidakakuratan, tetapi keuntungan besar yang akan dihasilkan dalam hal penguatan, penghematan daya, luas area, dll., dalam desain tersebut. Jenis desain perkiraan yang digunakan adalah menggunakan sirkuit perkiraan untuk bagian-bagian yang kurang penting, dan bagian-bagian yang akurat untuk bagian-bagian yang lebih penting, sehingga diperoleh keseimbangan yang baik antara efisiensi tertentu dan desain yang akurat. Hal ini sangat cocok untuk sistem yang tahan terhadap kesalahan.

**Hasil Implementasi**

Pengganda perkiraan 8 × 8 yang diusulkan diimplementasikan menggunakan Verilog HDL dan disintesis dengan Synopsys Design Compiler berdasarkan Teknologi CMOS 45 nm. Arsitektur terdiri dari AFA, Kompresor 4:2 yang dimodifikasi dan unit SG, dan logika periferal bersifat hibrida di beberapa tempat yang paling signifikan, sehingga menggunakan logika perkiraan, dan di sebagian besar tempat yang signifikan menggunakan logika tepat. Hasil pasca-sintesis menunjukkan peningkatan efisiensi perangkat keras yang luar biasa dibandingkan dengan pengganda tepat konvensional, sebesar 29,32% dalam luas area, 53,96% dalam penundaan, dan 45,73% dalam konsumsi daya. Desain ini menunjukkan produk daya-penundaan (PDP) yang lebih rendah dan kompleksitas desain sirkuit karena sirkuit logika perkiraan dengan minimisasi sirkuit yang diperoleh dengan menerapkan teknik komputasi perkiraan dalam sirkuit VLSI. Pengganda yang diusulkan memberikan solusi yang dirancang dengan sangat baik untuk sistem digital berdaya rendah dan berkinerja tinggi, dengan syarat adanya kompromi yang baik terkait sumber daya perangkat keras yang tersedia dan kemampuan komputasi yang dibutuhkan darinya.

Perbandingan kinerja berbagai pengganda, termasuk desain perkiraan yang diusulkan, sehubungan dengan luas area, penundaan, daya, dan produk daya-penundaan diberikan pada Tabel 6.3. Terlihat bahwa pengganda yang diusulkan lebih baik daripada desain, dengan peningkatan luas area sebesar 15,09%, penundaan sebesar 27,5%, dan daya sebesar 16,62% untuk kinerja akurasi yang wajar. Perbandingan menunjukkan kompromi antara akurasi, daya, dan kecepatan, memberikan efisiensi dan peningkatan kinerja yang baik pada desain yang diusulkan.

Sejumlah desain, misalnya, yang ada di (10, 5 desain 1 dan desain 2), memberikan hasil yang sedikit lebih baik mengenai beberapa faktor untuk sirkuit ini, tetapi mengalami kesalahan yang jauh lebih tinggi dalam metrik kesalahan terkait. Oleh karena itu, desain yang diusulkan memberikan desain menyeluruh yang baik, memperoleh peningkatan kinerja menyeluruh dalam efisiensi perangkat keras tanpa mengorbankan kinerja pada produk sinyal.

**Tabel 6.3** Hasil implementasi berbagai pengali

| Referensi | Area ($\mu m^2$) | Delay ($ns$) | Power ($\mu W$) | PDP ($ns \times \mu W$) |
|---|---|---|---|---|
| Accurate | 3,979 | 0.630 | 101 | 63.630 |
| [4] Design 1 | 3,261 | 0.410 | 64.850 | 26.590 |
| [4] Design 2 | 3,092 | 0.370 | 59.890 | 22.080 |
| [5] Design 1 | 682.2 | 0.290 | 33.530 | 9.724 |
| [5] Design 2 | 713.8 | 0.280 | 34.510 | 9.662 |
| [5] Design 3 | 2,618 | 0.340 | 53.920 | 18.330 |
| [5] Design 4 | 3,070 | 0.420 | 61.470 | 25.820 |
| [6] | 3,624 | 0.500 | 79.240 | 39.620 |
| [7] | 3,502 | 0.480 | 74.330 | 35.680 |
| [8] | 3,422 | 0.450 | 72.660 | 32.700 |
| [9] | 3,715 | 0.490 | 80.920 | 39.290 |
| [10] | 834 | 0.310 | 39.220 | 12.160 |
| [11] | 3,683 | 0.470 | 80.60 | 37.880 |
| [12] | 3,312 | 0.400 | 65.740 | 26.300 |
| Proposed | 2.812 | 0.290 | 54.810 | 15.894 |

Selain itu, sejauh menyangkut hasil implementasi, menunjukkan bahwa pengali yang diusulkan sangat skalabel dan cocok untuk aplikasi di mana daya merupakan sumber daya yang berharga, dengan ukuran silikon yang lebih kecil dan aktivitas switching yang jauh lebih rendah sambil memperoleh throughput yang lebih tinggi daripada desain konvensional. Hasil ini menunjukkan bahwa sistem kami sebenarnya dapat bekerja dengan baik untuk sistem VLSI yang tidak menggunakan banyak daya tetapi tetap perlu berkinerja baik. Hal ini sangat benar ketika Anda dapat mengorbankan sedikit ketepatan untuk penggunaan sumber daya yang tersedia dengan lebih baik.

**Tabel 6.4** Analisis matriks kesalahan dari berbagai pengali

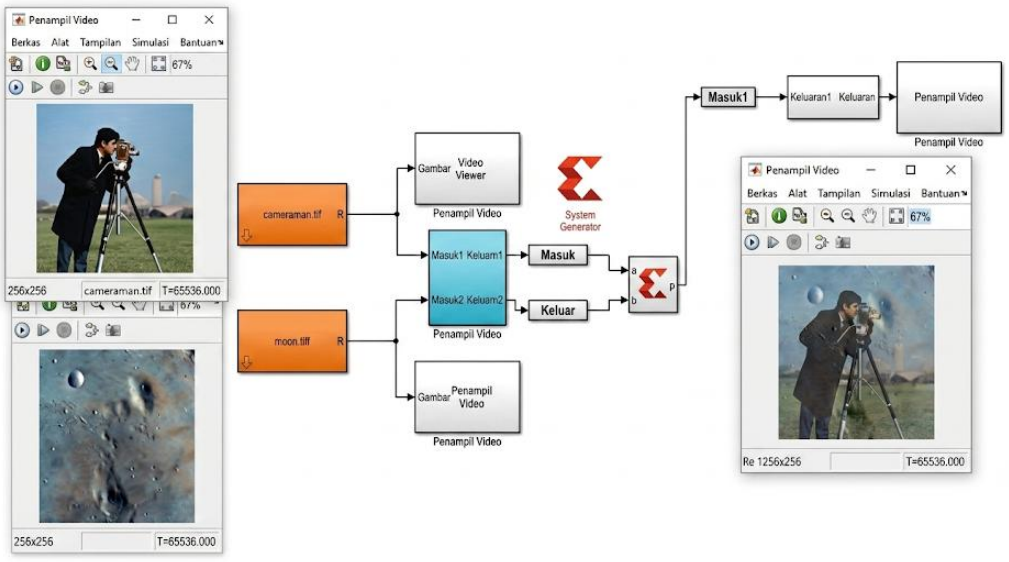
| Reference | MED | MRED | NED | Exact Outcome | PDP x MRED |
|---|---|---|---|---|---|
| Accurate | – | – | – | – | – |
| [4] Design 1 | 3,873.80 | 4.5483 | 0.0596 | 103 | 120.9 |
| [4] Design 2 | 3,508.80 | 4.2843 | 0.054 | 503 | 94.6 |
| [5] Design 1 | 4,376 | 0.4424 | 0.0673 | 690 | 4.3 |
| [5] Design 2 | 2,787.80 | 0.3135 | 0.0429 | 624 | 3.0 |
| [5] Design 3 | 3,095.70 | 0.3907 | 0.0476 | 1,885 | 7.2 |
| [5] Design 4 | 1,376.60 | 0.0854 | 0.0212 | 11,052 | 2.2 |

| | | | | | |
|---|---|---|---|---|---|
| [6] | 25.212 | 0.0326 | 0.0023 | 15,863 | 1.3 |
| [7] | 635.27 | 0.0292 | 0.0114 | 8,344 | 1.0 |
| [8] | 1,584.20 | 0.8264 | 0.0387 | 1,216 | 27.0 |
| [9] | 6,476 | 1.2000 | 0.0537 | 8,562 | 47.1 |
| [10] | 2,888 | 0.3002 | 0.0444 | 2,313 | 3.7 |
| [11] | 1,245 | 0.8107 | 0.0386 | 3,130 | 30.7 |
| [12] | 502.9 | 0.0728 | 0.0077 | 19,527 | 1.91 |
| Proposed | 137.69 | 0.0511 | 0.0021 | 10,787 | 0.81 |

Jika Anda melihat Tabel 6.4, Anda dapat dengan mudah memeriksa seberapa akurat berbagai pengali tersebut. Tabel tersebut mencantumkan desain yang di samping pengali baru kami. Tabel tersebut juga berisi beberapa nilai kesalahan utama. Efisiensi kesalahan energi, yang diukur melalui PDP x MRED (fJ), menunjukkan keunggulan desain yang diusulkan dibandingkan dengan pengali yang ada. Ukuran kinerja yang dicapai, yaitu MED sebesar 137,69, MRED sebesar 0,0511, NED sebesar 0,0021, dan 10.787 keluaran yang benar, jauh melampaui hasil desain sebelumnya. Arsitektur yang disarankan memiliki variasi yang sangat kecil dalam hal mendapatkan nilai aritmatika yang tepat dan PDP x MRED terendah (0,81 fJ), menunjukkan rasio energi-akurasi yang luar biasa.

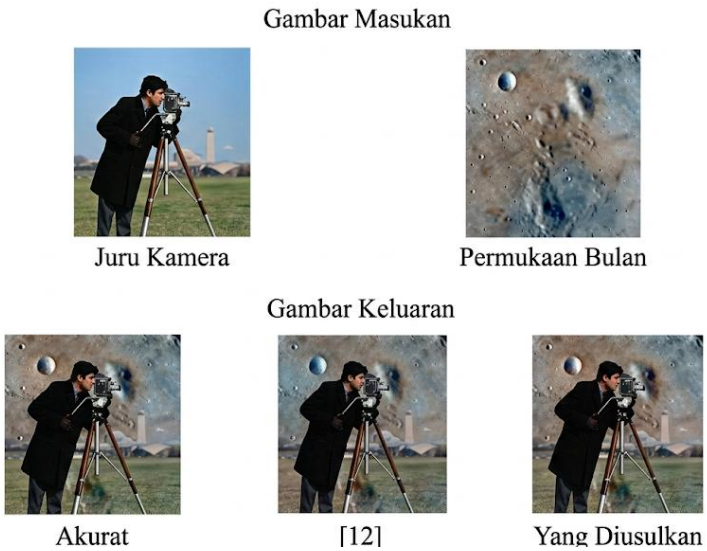
**Evaluasi Pemrosesan Citra**

Kinerja pengali perkiraan yang diusulkan diestimasi melalui banyak eksperimen pada dataset citra umum, termasuk Kameraman, Bulan, Jam, dan Pesawat Terbang, yang memiliki resolusi 256 × 256 piksel. Eksperimen dilakukan dalam Sistem MATLAB Simulink R2017b (64-bit) untuk menjamin penyediaan simulasi yang tepat dari operasi aritmatika dalam pipeline pemrosesan citra. Gambar 6.5 menunjukkan pengaturan eksperimental perkalian citra. Gambar 6.6 menyoroti hasil perkalian citra dengan desain pengali yang berbeda dan menunjukkan peningkatan kualitas visual melalui desain yang diusulkan.




**Gambar 6.5** Pengaturan Simulink sistem MATLAB untuk perkalian gambar.

Tabel 6.5, yang menggunakan *Mean Structural Similarity Index* (MSSIM) untuk melakukan analisis kuantitatif, menegaskan bahwa pengali yang diusulkan menghasilkan citra berkualitas tinggi, berbeda dengan desain yang saat ini digunakan. Gambar standar 512 × 512 (seperti Lena, Airplane16, dan Lakecolor) digunakan untuk melakukan eksperimen lebih lanjut guna menilai kinerja pada tugas penghalusan dan penajaman gambar. Gambar 6.7 dari hasil pemrosesan menunjukkan bahwa struktur gambar dapat dipertahankan secara efektif, artinya pengali yang diusulkan ini kuat dan akurat ketika diterapkan dalam berbagai kondisi.




**Gambar 6.6** Keluaran perkalian gambar.

**Tabel 6.5** Analisis MSSIM perkalian gambar menggunakan berbagai pengali

| Reference | Cameraman x moon surface | Clock x Airplane |
|---|---|---|
| [4] Design 1 | 0.6500 | 0.6642 |
| [4] Design 2 | 0.6800 | 0.6971 |
| [5] Design 1 | 0.6000 | 0.6153 |
| [5] Design 2 | 0.6500 | 0.6653 |
| [5] Design 3 | 0.6400 | 0.6580 |
| [5] Design 4 | 0.8100 | 0.8306 |
| [6] | 0.9300 | 0.9526 |
| [7] | 0.8700 | 0.8927 |
| [8] | 0.7600 | 0.7794 |
| [9] | 0.8600 | 0.8824 |
| [10] | 0.6300 | 0.6439 |
| [11] | 0.8400 | 0.8635 |
| [12] | 0.9200 | 0.9438 |
| Proposed | 0.9971 | 0.9979 |

Angka-angka numerik dalam Tabel 6.6 dan 6.7 menunjukkan bahwa desain ini memiliki nilai MSSIM yang lebih baik, yang mengindikasikan bahwa desain ini mempertahankan informasi

gambar yang lebih halus dan meningkatkan fidelitas gambar dalam proses penghalusan dan penajaman. Data tersebut menunjukkan bahwa pengali yang diusulkan dapat berhasil digunakan dalam peningkatan gambar secara real-time.




**Gambar 6.7** Hasil penghalusan gambar.

**Tabel 6.6** Analisis MSSIM penghalusan gambar menggunakan berbagai pengali

| Reference | Lena | Lake color |
|---|---|---|
| [4] Design 1 | 0.23 | 0.2609 |
| [4] Design 2 | 0.23 | 0.2697 |
| [5] Design 1 | 0.27 | 0.3024 |
| [5] Design 2 | 0.28 | 0.3153 |
| [5] Design 3 | 0.57 | 0.6284 |
| [5] Design 4 | 0.64 | 0.6931 |
| [6] | 0.72 | 0.7864 |
| [7] | 0.69 | 0.7529 |
| [8] | 0.6 | 0.6632 |
| [9] | 0.69 | 0.7582 |
| [10] | 0.29 | 0.3264 |
| [11] | 0.73 | 0.7941 |
| [12] | 0.74 | 0.8312 |
| Proposed | 0.8361 | 0.8601 |

**Tabel 6.7** Analisis MSSIM penajaman gambar menggunakan berbagai pengali

| Referensi | Lena | Permukaan Bulan (Moon Surface) |
|---|---|---|
| [5] Desain 1 | 0.60 | 0.6088 |
| [5] Desain 2 | 0.62 | 0.6297 |
| [5] Desain 3 | 0.69 | 0.7062 |
| [5] Desain 4 | 0.75 | 0.7614 |
| [6] | 0.87 | 0.8825 |
| [7] | 0.81 | 0.8216 |
| [8] | 0.71 | 0.7208 |
| [12] | 0.89 | 0.9033 |
| Metode yang Diusulkan (Proposed) | 0.9801 | 0.9896 |

**Sorotan dari Pekerjaan Ini**

- Desain pengali perkiraan berdaya rendah diusulkan untuk mengurangi konsumsi daya, luas area, dan penundaan komputasi.
- Kinerja desain yang diusulkan divalidasi melalui aplikasi pemrosesan gambar, termasuk perkalian gambar, penghalusan, dan penajaman.
- Semua eksperimen dilakukan menggunakan MATLAB System Simulink R2017b (64-bit), dan kinerja perangkat keras dianalisis melalui simulasi dalam teknologi 45 nm.
- Gambar uji standar seperti Cameraman, Lena, Permukaan Bulan, Pesawat Terbang, Jam, dan warna Danau digunakan untuk mengevaluasi kualitas keluaran yang diproses.
- Kualitas gambar dinilai menggunakan Mean SSIM, memastikan perbandingan kuantitatif antara desain perkiraan dan akurat.
- Hasil eksperimen mengkonfirmasi bahwa desain yang diusulkan mencapai penghematan daya yang signifikan sambil mempertahankan kualitas gambar yang dapat diterima.

## 6.3 PENGALI APROKSIMASI BERDAYA RENDAH UNTUK PEMROSESAN GAMBAR

Sebuah pengali aproksimasi berdaya rendah telah dirancang dan diimplementasikan secara khusus yang cocok untuk aplikasi pemrosesan gambar. Metrik MSSIM dari evaluasi kinerja menunjukkan pengurangan signifikan dalam penggunaan daya dan kompleksitas area dengan kinerja yang wajar dalam hal output. Hasil simulasi dan sintesis menunjukkan bahwa pengali yang diusulkan mengkonsumsi area 29,32% lebih sedikit, penundaan 53,96% lebih singkat, dan penghematan daya 45,73% dibandingkan dengan pengali eksak. Temuan ini membuktikan bahwa komputasi aproksimasi adalah metode yang sangat layak untuk menyederhanakan perangkat keras dengan mengorbankan akurasi yang dapat diterima dalam aplikasi yang toleran terhadap kesalahan, seperti pemrosesan gambar.

Desain yang disarankan menawarkan keseimbangan ideal antara akurasi komputasi, efektivitas daya, dan kecepatan pemrosesan, yang paling cocok untuk sistem tertanam dengan keterbatasan sumber daya dalam waktu nyata. Ini juga merupakan komponen yang menjanjikan dari akselerator jaringan saraf karena mampu berintegrasi dengan operasi perkalian-akumulasi. Studi lanjutan akan berupaya menggabungkan arsitektur tersebut ke dalam jaringan saraf untuk menentukan pengaruhnya terhadap akurasi klasifikasi, waktu yang dibutuhkan untuk inferensi, serta penggunaan dayanya terkait dengan model benchmark seperti MNIST dan ImageNet. Kinerja praktisnya juga akan dicari dengan mengimplementasikan perangkat keras. Arsitektur ini akan bermanfaat dalam pembuatan citra komputasional dan penskalaan akselerator AI serta perangkat keras pemrosesan citra masa depan dengan optimasi yang lebih baik.

# BAB 7
# OPTIMASI AI PADA VCO CINCIN

## 7.1 OPTIMASI RVCO BERBASIS AI

Komunikasi modern membutuhkan sinyal jam frekuensi tinggi. Pemancar dan penerima menggunakan sinyal jam ini untuk mengirim dan menerima sinyal. Sistem komunikasi ini membutuhkan sinyal jam yang bersih untuk meningkatkan kinerja sistem. Karakteristik utama sinyal jam adalah noise fase di domain frekuensi dan jitter di domain waktu. Loop pengunci fase (PLL) umumnya menghasilkan sinyal jam berkualitas tinggi. PLL terdiri dari VCO sebagai generator jam dan blok lainnya, seperti detektor frekuensi fase, pompa muatan, filter loop, dan pembagi, yang membantu meningkatkan noise fase jam keluaran. VCO dikategorikan menjadi VCO LC dan VCO cincin. Pada LC-VCO, area yang besar diperlukan, tetapi noise fase lebih rendah, sedangkan pada ring VCO, area yang lebih kecil dibutuhkan tetapi memiliki noise fase yang lebih tinggi pada sinyal. Mengoptimalkan trade-off kinerja area dan noise fase. Aplikasi menentukan topologi dan kategori yang tepat. Pendekatan desain konvensional didasarkan pada simulasi iteratif untuk pemodelan tingkat perangkat, dan penyetelan bias heuristik seringkali terbukti tidak cukup untuk pendekatan desain multi-dimensi (Gambar 7.1).

AI adalah alat yang ampuh untuk mengatasi masalah optimasi sirkuit yang kompleks, termasuk RVCO. Ada beberapa teknik, seperti pembelajaran mesin (ML), pembelajaran mendalam (DL), jaringan saraf yang diinformasikan fisika (PINN), dan pembelajaran penguatan (RL), yang berpotensi mempercepat desain RVCO untuk mengeksplorasi, mengoptimalkan keadaan bias, dan kalibrasi on-chip adaptif. Model yang digerakkan AI menangkap non-linearitas dalam dinamika sel tunda dan dapat memprediksi noise fase dengan lebih sedikit simulasi, yang juga berlaku untuk lintas teknologi.

Selain itu, arsitektur adaptif yang didukung oleh AI juga membantu osilator pemulihan diri untuk pergeseran PVT waktu nyata. Penggunaan AI dalam desain RVCO juga menghadirkan tantangan [6] berupa kelangkaan data berlabel berkualitas tinggi, risiko model kotak hitam yang kurang interoperabilitas, biaya integrasi untuk blok desain sinyal campuran yang berbeda dan alur desainnya, serta kesenjangan antara pemodelan dan batasan tingkat sirkuit. Pendekatan yang berbeda dikombinasikan dengan wawasan tingkat sirkuit, dengan metode berbasis data atau model pengganti yang terukur untuk desain tingkat sistem dan kerangka kerja lintas teknologi untuk mengatasi tantangan ini. Sebagian besar teknik berbasis AI ini dapat diterapkan pada modul sinyal campuran analog lainnya. Peran optimasi berbasis AI untuk RVCO adalah untuk meningkatkan kinerja osilator dan membentuk kembali metode desain untuk synthesizer frekuensi generasi berikutnya. Pembahasan tambahan dapat memungkinkan solusi osilator yang tahan terhadap noise, hemat energi, dan portabel secara teknologi untuk sistem komunikasi berkecepatan tinggi.



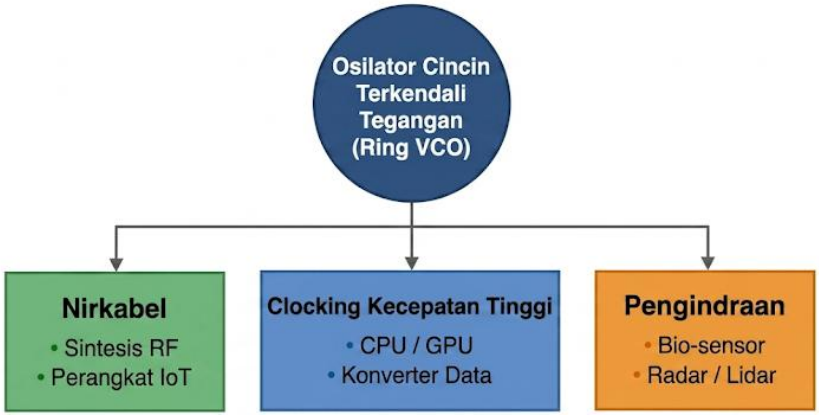

**Gambar 7.1** Gambaran umum teknik optimasi berbasis AI untuk arsitektur RVCO.

**Peran RVCO dalam Sistem Komunikasi Modern**

Karena kesederhanaan struktural, kompatibilitas CMOS, dan kemampuan rentang penyetelan yang luas, RVCO digunakan untuk sistem pembangkitan frekuensi. LC-VCO menggunakan induktor dan kapasitor besar untuk membangun resonansi, yang meningkatkan luas area. Pada saat yang sama, RVCO menggabungkan elemen penundaan area kecil dalam loop umpan balik, sehingga menarik untuk aplikasi sistem-on-chip yang intensif digital. Tujuan utama RVCO adalah untuk menyediakan clock yang bersih dengan kualitas sinyal yang lebih tinggi ke sistem secara keseluruhan. Clock yang bersih ini mengoperasikan sistem lengkap seperti mikroprosesor, FPGA, sistem komputasi klasik atau kuantum, dan sistem komputasi neuromorfik.

Sinyal clock yang bersih adalah prioritas utama untuk kinerja yang lebih baik pada tingkat sirkuit dan sistem. Desain VCO cincin juga harus kompatibel dengan proses CMOS dan digital standar. Hal ini membuatnya mudah diintegrasikan dengan sirkuit sinyal campuran lainnya, menawarkan rentang penyetelan yang luas yang penting untuk sistem komunikasi multi-standar. Peningkatan VCO cincin adalah masalah optimasi parameter multi-level, yang meliputi noise fase, konsumsi daya, area, rentang penyetelan, dan ketahanan start-up.

VCO cincin juga memainkan peran penting dalam sistem clocking dan timing digital. Kemampuan mereka untuk menghasilkan frekuensi dalam rentang multi-GHz dengan sel tunda yang dapat diskalakan membuat mereka ideal untuk distribusi clock chip, penggandaan frekuensi, dan osilator yang dikontrol secara digital di semua PLL digital. Demikian pula, AI dan sistem komputasi tepi yang sedang berkembang bergantung pada osilator adaptif yang harus mampu melakukan kalibrasi sendiri di bawah variasi PVT yang berbeda (Gambar 7.2).




**Gambar 7.2** Peran VCO cincin dalam sistem modern.

Meskipun desain RVCO memiliki beberapa keunggulan, mereka menghadapi keterbatasan dalam noise fase, jitter, dan sensitivitas catu daya ketika digunakan di lingkungan noise ultra-rendah untuk front end RF dan referensi waktu presisi. Namun, pembahasan yang sedang berlangsung yang mencakup teknik sirkuit yang sadar noise, perluasan bandwidth menggunakan saluran tunda yang dikontrol tegangan, dan teknik optimasi yang sadar AI meningkatkan kinerja RVCO, yang secara konvensional didominasi oleh LC VCO.

**Keterbatasan Alur Desain Konvensional**

Metode desain konvensional untuk RVCO meliputi analisis manual, penentuan ukuran transistor, dan simulasi iteratif serta optimasinya. Metode ini cocok untuk studi kelayakan awal tetapi menghadapi masalah signifikan ketika diterapkan untuk analisis tingkat sistem lengkap. Iterasi ini, yang dibentuk untuk optimasi desain RVCO, mencakup berbagai ukuran transistor dan arus bias dengan beban yang berbeda, yang diverifikasi oleh simulasi SPICE. Karena efek kanal pendek dan masalah lainnya, simulasi SPICE secara tidak akurat memodelkan perangkat pada node teknologi canggih, sehingga kebisingan menjadi signifikan. Desain juga dipengaruhi oleh variasi PVT dan kelayakan simulasi yang ekstensif. Tantangan dalam alur desain konvensional adalah sebagai berikut:

**Kompromi Multidimensi**

Terdapat saling ketergantungan antara derau fasa, jitter, rentang penyetelan, konsumsi daya, dan luas area, yang memengaruhi penyetelan parameter secara manual dalam ruang desain yang kompleks, sehingga menghasilkan solusi yang kurang optimal.

**Beban Komputasi**

Untuk mengoptimalkan RVCO untuk simulasi derau dan transien pada kondisi PVT, diperlukan simulasi tingkat SPICE yang ekstensif, yang meningkatkan beban secara eksponensial dan semakin meningkat dengan simulasi lintas teknologi.

**Akurasi Prediktif Terbatas**

Karena model derau fasa yang disederhanakan dan dinamika tingkat perangkat, akurasi yang terbatas dicapai pada proses CMOS canggih di mana efek saluran pendek dan arus bocor mendominasi.

**Skalabilitas Buruk di Berbagai Teknologi**

Dalam metode konvensional, data simulasi yang dihasilkan dalam satu teknologi tidak dapat langsung digunakan untuk node teknologi yang berbeda. Hal ini menyebabkan skalabilitas yang buruk di berbagai teknologi. Setiap migrasi dari satu teknologi ke teknologi lain memerlukan optimasi bias berulang dan validasi parameter lainnya.

**Penanganan Variabilitas yang Tidak Memadai**

Fluktuasi akibat PVT, penuaan, dan efek ketidaksesuaian mengakibatkan guard banding yang berlebihan dan peningkatan konsumsi daya, yang merupakan tantangan signifikan dalam metode konvensional.

**Hambatan Integrasi**

VCO cincin jarang digunakan secara terpisah. VCO cincin merupakan bagian dari PLL yang lebih besar atau synthesizer frekuensi. Alur desain tradisional kurang memiliki metodologi ko-simulasi yang efisien yang secara bersamaan mengoptimalkan VCO dengan filter loop, pembagi, dan logika kontrol digital. Bagian berikut membahas motivasi di balik metodologi desain yang dibantu AI dan manfaatnya.

## 7.2 METODOLOGI DESAIN YANG DIBANTU AI

Secara konvensional, perancang menggunakan proses manual dan iteratif dalam mendesain VCO cincin dan blok sinyal campuran analog lainnya. Perancang berpengalaman dapat memilih dan mencoba pendekatan berbasis kesalahan. Teknik-teknik ini telah bekerja dengan baik di masa lalu. Namun, seiring dengan penggunaan teknologi node yang lebih canggih, efektivitas pendekatan ini menurun, yang memotivasi para perancang untuk menggunakan metode berbasis AI untuk merancang sirkuit yang lebih baik dengan waktu desain yang lebih singkat dan hasil yang lebih akurat (Tabel 7.1).

**Tabel 7.1** Perbandingan alur desain konvensional dan yang dibantu AI

| Aspek | Alur Desain Tradisional | Alur Desain Berbantuan AI |
|---|---|---|
| Proses Desain | Penyetelan manual (*manual tuning*), simulasi SPICE secara iteratif, dan penyapuan parameter (*parametric sweeps*). | Eksplorasi otomatis menggunakan algoritma AI, seperti **Optimasi Bayesian (Bayesian Optimization)**, **Algoritma Genetika (Genetic Algorithms)**, dan **Deep Learning**. |
| Efisiensi | Memerlukan waktu yang lama dan sering kali membutuhkan ribuan simulasi. | Mengurangi waktu siklus desain sebesar **30%–70%** dengan jumlah simulasi yang lebih sedikit. |
| Optimasi | Berfokus pada satu metrik pada satu waktu, misalnya **phase noise** atau konsumsi daya. | Optimasi multiobjektif yang secara bersamaan mempertimbangkan **noise**, konsumsi daya, linearitas, dan keandalan (*robustness*). |
| Skalabilitas | Mengalami kesulitan pada node CMOS tingkat lanjut akibat efek | Mampu menangani ruang desain berdimensi tinggi dan nonlinier secara efisien. |

| | | |
|---|---|---|
| | **saluran pendek (short-channel effects)** dan variasi proses. | |
| Prediktabilitas | Sangat bergantung pada intuisi perancang dan eksplorasi manual. | Mempelajari hubungan antara parameter dan performa dari data sehingga memungkinkan optimasi prediktif. |
| Ketahanan terhadap Variasi PVT (Process, Voltage, Temperature) | Memerlukan simulasi **corner** secara terpisah; hasil yang diperoleh sering kali tidak konsisten. | Memasukkan data **PVT corner** dan simulasi **Monte Carlo** ke dalam proses pelatihan sehingga menghasilkan desain yang lebih andal. |
| Adaptabilitas | Desain bersifat statis dan memiliki ruang yang terbatas untuk penyesuaian setelah fabrikasi (*post-fabrication*). | **Reinforcement Learning** memungkinkan penyesuaian adaptif setelah fabrikasi (*post-fabrication tuning*), termasuk konsep **self-healing circuits**. |
| Biaya Simulasi | Sangat bergantung pada simulasi **SPICE** berakurasi tinggi yang mahal. | Menggunakan **surrogate models** atau **multi-fidelity models** untuk menyaring kandidat desain sehingga mengurangi biaya komputasi. |
| Transfer Pengetahuan | Setiap node teknologi baru memerlukan siklus desain baru dari awal. | **Transfer Learning** mempercepat proses adaptasi antar node teknologi. |
| Interpretabilitas | Keputusan desain bersifat transparan dan didasarkan pada pengalaman serta pertimbangan insinyur. | Sifat **black-box** pada AI menimbulkan tantangan terkait kepercayaan (*trust*) dan proses validasi. |

**Motivasi Penggunaan AI dalam Desain Analog/RF**

Motivasi utama untuk memasukkan AI dalam proses desain VCO cincin adalah saling ketergantungan dari berbagai parameter. Derau fasa, luas area, konsumsi daya, dan frekuensi operasi adalah parameter utama yang digunakan dalam proses desain. Semua parameter ini saling bergantung. AI membantu mengoptimalkan parameter VCO cincin dalam waktu singkat. Teknik AI seperti RL dapat membantu penyetelan pasca-fabrikasi dengan menyediakan penyesuaian arus bias secara real-time dan penyetelan perangkat untuk operasi yang andal.

**Manfaat Optimasi Berbasis AI**

Pendekatan berbasis AI memberikan banyak manfaat pada VCO cincin jika diterapkan secara sistematis. Ini termasuk peningkatan pemilihan parameter, optimasi multi-objektif, pengurangan waktu desain menggunakan algoritma AI seperti algoritma Bayesian dan Genetika, ketahanan yang lebih baik terhadap variasi, penyetelan pasca-fabrikasi, peningkatan waktu pemasaran, dan transfer pengetahuan.

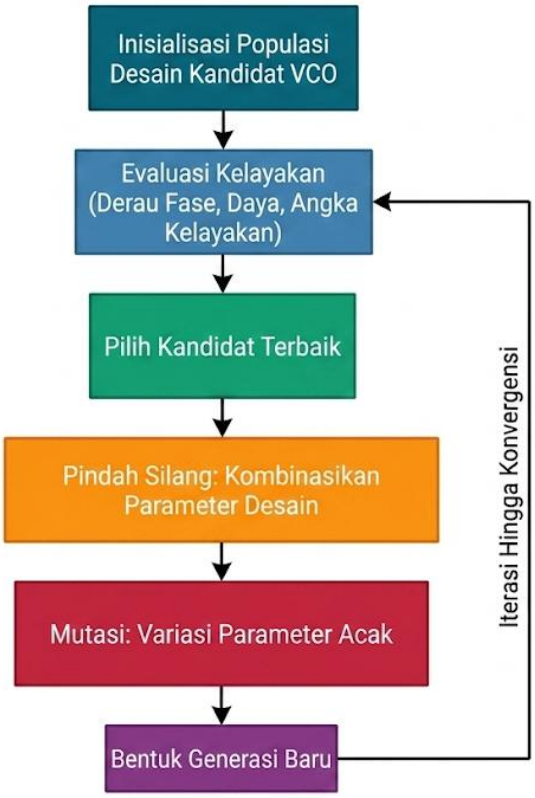
## 7.3 PARADIGMA OPTIMASI BERBASIS AI

Optimasi berbasis AI dapat dilakukan menggunakan algoritma AI karena VCO cincin membutuhkan trade-off multi-objektif. Optimasi berbasis AI membantu dalam simulasi yang sangat kompleks untuk lingkungan ruang desain multi-dimensi dengan menyediakan alternatif. Dalam literatur, beberapa pendekatan menggunakan AI, dan ini dibahas sebagai berikut.

**Algoritma Evolusi (EA)**

Kemajuan berkelanjutan dalam IC CMOS telah memungkinkan realisasi sistem elektronik yang semakin kompleks yang mendominasi lanskap teknologi saat ini. Evolusi ini mendorong permintaan akan solusi yang sangat miniatur dengan konsumsi daya rendah dan fungsionalitas yang ditingkatkan. Namun, desain sirkuit menjadi lebih menantang seiring dengan skala dimensi transistor, yang menyebabkan kerentanan yang lebih tinggi terhadap kesalahan desain dan siklus desain-ke-tapeout yang lebih lama. Untuk mengurangi masalah ini, alat perangkat lunak yang meningkatkan produktivitas desain menghadirkan solusi yang menjanjikan. Algoritma ini terinspirasi oleh seleksi alam berdasarkan prinsip.

Sebagai sebuah kromosom, ia mengkodekan parameter rangkaian, seperti arus bias, ukuran transistor, dan beban. Pertama-tama, ia menghasilkan populasi awal desain dan mengevaluasi desain menggunakan fungsi kebugaran yang disebut Figure of Merit (FoM). Kemudian, operator genetik seperti persilangan dan mutasi diterapkan untuk meningkatkan desain, dan iterasi dilakukan hingga konvergensi (Gambar 7.3).

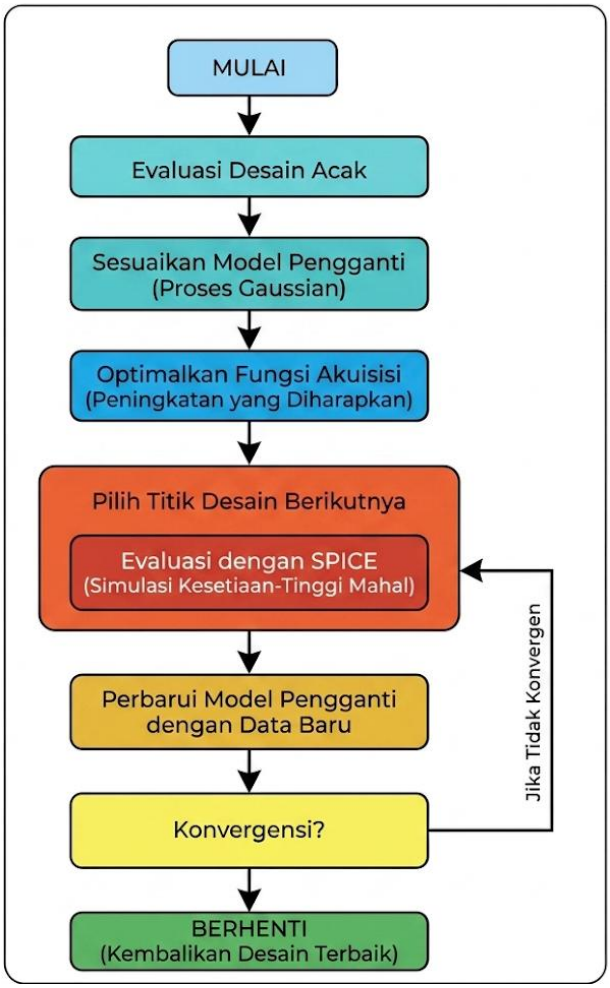



**Gambar 7.3** Diagram alir algoritma evolusi dalam desain VCO cincin.

Algoritma ini cocok untuk masalah yang sangat nonlinier. Konvergensi mungkin lambat dan membutuhkan banyak iterasi untuk menemukan solusi yang baik; namun, algoritma ini mencapai peningkatan 15%–25% pada FoM dan pengurangan hingga 60% dalam persyaratan simulasi dibandingkan dengan metode konvensional. Meskipun model analitik untuk noise fase, seperti model LTV dan FoM, dapat berfungsi sebagai pedoman desain yang dapat diterapkan, menjelajahi ruang desain lengkap tetap sangat menantang, bahkan untuk perancang yang berpengalaman.

**Optimasi Bayesian (BO)**

Optimasi Bayesian (BO) memperlakukan evaluasi kinerja sirkuit sebagai kotak blok. Ia menggunakan pendekatan pemodelan pengganti probabilistik, umumnya proses Gaussian. Ia dimulai dengan evaluasi desain acak dan menyesuaikan model pengganti untuk memprediksi kinerjanya. Ia juga menggunakan fungsi akuisisi untuk mengevaluasi peningkatan yang diharapkan (Gambar 7.4).




**Gambar 7.4** Alur kerja Optimasi Bayesian (BO).

Kemudian, proses berlanjut ke titik desain berikutnya, dievaluasi dengan SPICE, dan model diperbarui. Keunggulan algoritma ini adalah membutuhkan lebih sedikit simulasi daripada algoritma evolusi dan juga cocok untuk evaluasi berbasis SPICE yang mahal. Keterbatasan

algoritma ini adalah akurasi model pengganti menurun jika jumlah variabel lebih dari 20. Namun, algoritma ini dapat mencapai trade-off noise-power yang kompetitif tanpa perlu melakukan sweep yang intensif.

**Pendekatan Deep Learning**

Dalam pendekatan DL, jaringan saraf digunakan untuk mempelajari pemetaan dari parameter perangkat dan metrik kinerja (seperti frekuensi, noise fase). Setelah sistem dilatih, sistem dapat memberikan prediksi kinerja secara instan. Sistem ini dapat digunakan dalam VCO cincin untuk memprediksi spektrum noise fase dan kurva penyetelan. Setelah pelatihan, evaluasi menjadi hampir instan dan dapat digeneralisasikan di berbagai rentang parameter. Keterbatasan pendekatan ini termasuk persyaratan dataset pelatihan yang besar dan beragam. Pendekatan DL mencakup dua metode utama:

**Multilayer Perceptron (MLP)**

MLP adalah jaringan saraf feed-forward yang terdiri dari beberapa lapisan neuron yang saling terhubung. Setiap neuron menerapkan jumlah tertimbang dari inputnya, diikuti oleh fungsi aktivasi nonlinier. Ini menangkap hubungan nonlinier antara parameter sirkuit dan metrik output. Penumpukan beberapa lapisan tersembunyi dapat mendekati pemetaan nonlinier kompleks antara variabel input (parameter rangkaian) dan variabel output (metrik kinerja). Dalam kasus VCO cincin, MLP dilatih menggunakan data yang diperoleh dari simulasi SPICE atau rangkaian terukur, dan dapat memprediksi kinerja rangkaian hampir secara instan. Keuntungan dari pendekatan ini termasuk pemodelan nonlinier dan prediksi cepat. Namun, pada saat yang sama, dibutuhkan data yang ekstensif, dan mungkin gagal memprediksi secara akurat jika diterapkan pada node teknologi yang berbeda atau sudut PVT ekstrem.

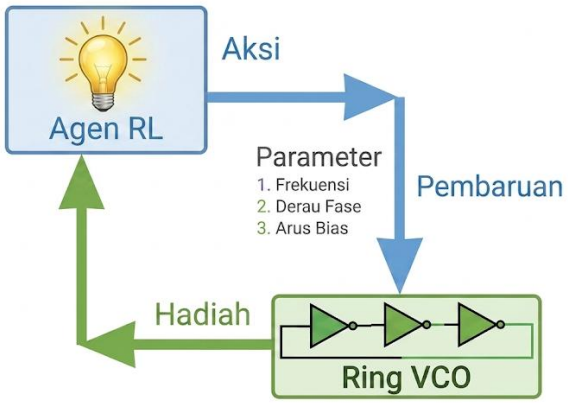
**Jaringan saraf konvolusional (CNN)**

CNN adalah jenis jaringan saraf khusus yang dirancang untuk korelasi spasial atau struktural dalam data. CNN menggunakan filter konvolusional, yang tidak memiliki lapisan yang terhubung sepenuhnya tetapi meluncur di atas input untuk mengekstrak pola dan fitur lokal. Hal ini membuat CNN sangat efektif dalam belajar dari data terstruktur seperti spektrum dan tata letak VCO. CNN dapat menganalisis tata letak dengan geometri perangkat, parasit perutean, jalur kebocoran, atau noise yang digabungkan. Ia juga dapat memproses spektrum derau fase dan histogram jitter. CNN mengungguli MLP ketika input memiliki korelasi struktural yang melekat dan membutuhkan lebih sedikit sampel pelatihan daripada MLP karena penggunaan kembali filter konvolusi. Namun, ia menjadi kompleks secara komputasi selama pelatihan.

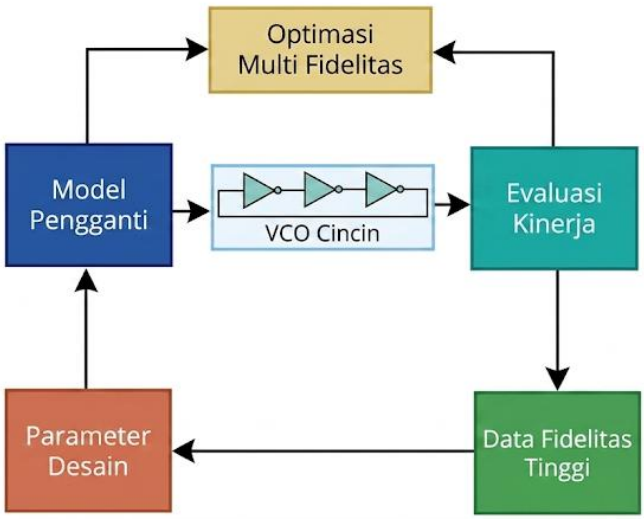
**Pembelajaran Penguatan (*Reinforcement Learning/RL*)**

Optimasi dilakukan dalam RL sebagai masalah pengambilan keputusan berurutan. Agen RL menyesuaikan parameter secara iteratif berdasarkan umpan balik, yang disebut hadiah, berdasarkan peningkatan kinerja sirkuit. Dalam RL, lingkungan diatur terlebih dahulu, seperti simulator sirkuit atau pengaturan perangkat keras, dan kemudian keadaan didefinisikan, seperti kumpulan parameter saat ini. Kemudian hadiah didefinisikan sebagai pengurangan noise fase dan peningkatan FoM (Gambar 7.5).

Terakhir, agen dilatih untuk memaksimalkan hadiah kumulatif. Ini lebih berguna untuk VCO dalam cincin dalam penyetelan pasca-fabrikasi dan optimasi perangkat keras dalam loop. Pelatihan dapat memakan biaya komputasi yang mahal, dan stabilitas juga bergantung pada fungsi hadiah dalam RL. Kerangka kerja RL meningkatkan efisiensi daya VCO cincin dengan menyetel bias dan parameter ukuran secara adaptif. Sebelum RL, desain mengkonsumsi daya statis yang lebih tinggi karena keterbatasan optimasi manual. Setelah RL, daya total berkurang sekitar 15%–25% dengan distribusi tahap yang seimbang dan stabilitas yang lebih baik di bawah variasi PVT, mencapai trade-off kinerja-daya yang superior (Gambar 7.6).




**Gambar 7.5** Alur kerja pembelajaran penguatan (*reinforcement learning*/RL).




**Gambar 7.6** Alur kerja pendekatan berbasis pemodelan pengganti (*surrogate modeling*).

**Pemodelan Surrogate dan Optimasi Multi-Fidelitas**

Prinsip utama pendekatan ini adalah bahwa tidak semua simulasi perlu sangat akurat. Pendekatan ini menggunakan model fidelitas rendah seperti persamaan yang disederhanakan dan model perilaku, tetapi hanya menyempurnakan kandidat yang menjanjikan dengan simulasi fidelitas tinggi. Hal ini mengurangi biaya komputasi sambil mempertahankan akurasi dan dapat dikombinasikan dengan model berbasis fisika dengan AO untuk meningkatkan

keandalan. Pendekatan ini membutuhkan kalibrasi yang cermat antara tingkat fidelitas yang berbeda, tetapi menghemat waktu eksekusi simulasi secara signifikan (Tabel 7.2).

**Tabel 7.2** Perbandingan berbagai algoritma dan aplikasinya pada VCO cincin

| Pendekatan | Aplikasi pada Ring VCO | Keunggulan untuk Ring VCO | Keterbatasan pada Ring VCO | Hasil yang Dilaporkan/Diharapkan |
|---|---|---|---|---|
| Algoritma Evolusioner (Evolutionary Algorithms/EA) *(misalnya Algoritma Genetika)* | Mengodekan ukuran transistor, arus bias, dan nilai beban sebagai solusi kandidat, kemudian mengembangkannya untuk memperoleh **phase noise** dan **Figure of Merit (FoM)** yang lebih baik. | Mampu menangani kompromi (*trade-off*) nonlinier, seperti **phase noise** versus konsumsi daya; mampu menghindari solusi optimum lokal; fleksibel untuk berbagai parameter desain (*design knobs*). | Memerlukan ratusan simulasi **SPICE** pada setiap generasi; konvergensi lebih lambat untuk ruang parameter yang besar. | Peningkatan **FoM** sebesar **15%–25%** dan pengurangan jumlah simulasi sekitar **60%** dibandingkan penyapuan parameter (*manual sweeps*). |
| Optimasi Bayesian (Bayesian Optimization/BO) | Memodelkan **FoM** dan **phase noise** sebagai fungsi *black-box*; menggunakan **surrogate model** (misalnya **Gaussian Process**) untuk memandu pemilihan parameter. | Mengoptimalkan kompromi antara **phase noise** dan konsumsi daya secara efisien; membutuhkan lebih sedikit simulasi SPICE berakurasi tinggi; sangat baik untuk penyetelan arus bias dan ukuran transistor. | Akurasi *surrogate model* menurun ketika jumlah variabel desain sangat banyak (>20); memerlukan pemilihan **kernel** dan **acquisition function** yang tepat. | Menemukan desain yang mendekati optimum dalam kurang dari **50 simulasi**; menghasilkan kompromi **noise–daya** yang kompetitif. |
| Multilayer Perceptron (MLP) | Mempelajari hubungan antara rasio **W/L**, arus bias, **VDD**, frekuensi, **phase noise**, konsumsi daya, dan kurva penyetelan (*tuning curve*). | Mampu mempelajari hubungan nonlinier antara parameter dan performa; setelah dilatih dapat memprediksi metrik VCO secara instan; mempercepat proses pencarian desain. | Membutuhkan dataset pelatihan yang besar pada berbagai kondisi **PVT**; kurang mampu melakukan generalisasi pada node teknologi baru; bersifat **black-box**. | Prediksi spektrum **phase noise** dan **FoM** yang akurat; kecepatan prediksi beberapa orde magnitudo lebih cepat dibandingkan simulasi SPICE. |
| Convolutional Neural Networks (CNN) | Belajar dari data terstruktur seperti gambar tata letak | Mampu menangkap efek parasitik akibat tata letak | Membutuhkan representasi data yang terstruktur | Memprediksi penurunan performa akibat tata letak; |

| | | | | |
|---|---|---|---|---|
| | (*layout images*), prediksi **noise** yang dipengaruhi parasitik, serta data spektral **phase noise/jitter**. | (routing dan kopling); mempelajari korelasi spektral pada **phase noise**; lebih hemat data dibandingkan MLP untuk data terstruktur. | (layout atau spektrum); proses pelatihan lebih kompleks; interpretabilitas masih terbatas. | mengidentifikasi **hotspot** pada spektrum untuk mendukung perbaikan desain. |
| Reinforcement Learning (RL) | Agen AI menyesuaikan parameter VCO secara bertahap (arus bias, elemen penunda, tegangan kontrol), dengan **reward** berupa peningkatan **FoM**. | Mendukung penyetelan adaptif setelah fabrikasi (*post-fabrication tuning*); mengompensasi variasi **PVT** dan penuaan (*aging*); cocok untuk **self-healing VCO**. | Pelatihan membutuhkan komputasi yang besar; perancangan **reward function** cukup sulit; mungkin memerlukan pengujian perangkat keras (*hardware-in-the-loop*). | Menjanjikan untuk VCO adaptif secara waktu nyata; mampu mempertahankan **FoM** pada berbagai kondisi proses (*process corners*). |
| Surrogate Model dan Multi-Fidelity Model | Menggunakan model perilaku (*behavioral model*) atau model analitik untuk pencarian awal, kemudian disempurnakan menggunakan simulasi **SPICE** berakurasi tinggi. | Mengurangi biaya simulasi secara signifikan; memungkinkan eksplorasi ruang desain berskala besar; dapat dikombinasikan dengan **physics-informed models**. | Risiko model dengan fidelitas rendah memberikan arah pencarian yang keliru; memerlukan proses kalibrasi antar model. | Memungkinkan eksplorasi parameter VCO dalam skala besar; waktu komputasi jauh lebih singkat dibandingkan simulasi SPICE secara menyeluruh (*brute-force*). |

## 7.4 TANTANGAN DAN KETERBATASAN OPTIMASI RVCO BERBASIS AI

Pada bagian ini, tantangan dan keterbatasan strategi optimasi berbasis AI untuk VCO cincin dibahas, yang meliputi hal-hal berikut.

**Generalisasi ke Node Teknologi Baru**

Semua model AI yang dibahas di atas biasanya dilatih pada data simulasi atau pengukuran dari node teknologi CMOS tertentu. Perilaku transistor, pemodelan parasit, dan mekanisme kebisingan berbeda secara signifikan pada node teknologi yang lebih rendah. Kebisingan fasa dan rentang penyetelan VCO cincin sangat bergantung pada gm perangkat dan sensitivitas catu daya. Model yang dilatih pada satu node mungkin berkinerja buruk atau salah prediksi pada node lain. Jadi, pelatihan ulang atau pembelajaran transfer merupakan persyaratan penting untuk setiap teknologi baru, yang meningkatkan biaya waktu dan data.

**Biaya Pembangkitan Data**

Pelatihan model AI secara akurat membutuhkan kumpulan data besar dari nilai frekuensi, kebisingan fasa, daya, dan jitter yang diperoleh dari simulasi SPICE atau pengukuran silikon. Simulasi noise fase (PSS/Pnoise) membutuhkan biaya komputasi yang tinggi, dan eksplorasi berbagai sudut PVT dan Monte Carlo menambah beban data yang besar. Pengumpulan data dapat memakan waktu lebih lama daripada desain manual kecuali jika digunakan pemodelan multi-fidelity atau metode yang efisien data.

**Integrasi dengan Toolchain EDA**

Pengoptimal AI tidak terintegrasi secara native ke dalam alur kerja standar industri seperti Cadence Virtuoso dan Synopsys. Jadi dalam hal ini, perancang perlu mengekspor hasil simulasi secara manual, memasukkannya ke dalam kerangka kerja AI eksternal seperti Python atau PyTorch, dan mengulanginya. Hal ini menghambat adopsi alur kerja yang lancar. Oleh karena itu, alur kerja yang dibantu AI tetap berorientasi pada pembahasan dan tidak diadopsi secara merata di tim desain IC komersial.

**Masalah Interpretasi dan Kepercayaan**

Desain yang dibantu AI seringkali mengalami hambatan signifikan untuk adopsi yang luas: kurangnya interpretasi solusi yang diberikan. Berbagai model AI beroperasi sebagai kotak hitam, seperti pendekatan DL, yang memetakan parameter perangkat ke hasil kinerja tanpa wawasan fisik apa pun. Ketidakjelasan mengenai wawasan sirkuit ini memengaruhi keandalan dan validitas desain. Selama optimasi berbasis AI, harus dijelaskan dengan jelas mengapa ukuran dan kondisi bias tertentu dipilih, yang membuat sulit untuk mempercayai atau memverifikasi desain. Untuk RVCO, optimasi yang dibantu AI harus menjelaskan istilah-istilah fisika osilator, seperti kriteria Barkhausen.

Jika tidak demikian, hal ini dapat menimbulkan tantangan dalam aplikasi regulasi. Tantangan ini menjadi lebih kritis dalam aplikasi keselamatan atau medis di mana pilihan parameter dapat menyebabkan kegagalan yang tidak terduga. Perancang mungkin enggan mengadopsi alur berbasis AI tanpa interpretasi, karena mereka tidak dapat dengan mudah memvalidasi atau men-debug lintasan optimasi. Mengatasi masalah ini membutuhkan integrasi teknik AI yang dapat dijelaskan ke dalam kerangka kerja optimasi sirkuit.

**AI On-Chip untuk Penyetelan Waktu Nyata**

Untuk RVCO yang sepenuhnya adaptif, kecerdasan buatan on-chip waktu nyata dapat menjadi solusi yang menjanjikan. Metode desain RVCO konvensional membutuhkan margin konservatif untuk ketahanan terhadap PVT. Pada saat yang sama, pendekatan yang dibantu AI pada chip membantu menyesuaikan bias dan parameter perangkat secara dinamis sesuai dengan kondisi operasi seperti noise suplai, pergeseran suhu, dan penuaan perangkat. Mesin RO dan jaringan saraf digunakan untuk ini karena mereka dapat berinteraksi terus menerus dengan lingkungan sirkuit untuk mengoptimalkan metrik RVCO.

Namun, AI pada chip dengan silikon tetap menantang karena alasan berikut. Implementasi perangkat keras harus sangat efisien dalam hal luas area dan daya untuk menghindari pengurangan manfaat penyetelan adaptif. Keandalan dan stabilitas algoritma pembelajaran di bawah kendala waktu nyata juga merupakan perhatian kritis, terutama dalam

sistem yang kritis terhadap keselamatan atau sistem kelas komunikasi. Lebih lanjut, perancangan bersama mesin pembelajaran dan front end analog sangat penting untuk memastikan kecepatan konvergensi dan ketahanan tanpa mengganggu operasi sirkuit reguler.

Terlepas dari tantangan ini, para peneliti telah mulai mengeksplorasi penanaman mesin AI ringan, seperti jaringan saraf, langsung pada chip untuk ring-VCO yang dapat menyetel sendiri. Penyetelan pasca-fabrikasi sangat penting untuk mengkompensasi variasi PVT, noise suplai, dan efek penuaan. AI dapat menyesuaikan tegangan kontrol, arus bias, atau elemen penundaan secara adaptif. Namun, hal ini meningkatkan beban perangkat keras dan mengurangi skalabilitas. AI pada chip masih berupa pembuktian konsep dengan kekhawatiran terkait biaya, daya, dan ketahanan.

## 7.5 PENDEKATAN AI UNTUK OPTIMASI DAN ADAPTASI VCO CINCIN

Pendekatan berbasis AI untuk desain VCO cincin memiliki kepentingan yang sangat besar. Hal ini dapat diklasifikasikan ke dalam subbagian berikut.

**Kerangka Kerja Desain Bersama AI-Manusia Hibrida**

Korelasi silang antara kecerdasan manusia dan AI dapat meningkatkan proses desain VCO cincin. Perancang mengetahui wawasan arsitektur, seperti pilihan sel tunda, skema arus bias, dan isolasi catu daya, dan AI dapat membantu dalam penyetelan lokal dengan kesadaran PVT/Montecarlo. Lingkaran kasar-halus ini dapat dibentuk, membuat proses desain menjadi lancar. Penyertaan kendala seperti margin awal, gm/c, headroom, Barkhau-sen, slew maksimum, dan DRC tata letak langsung ke dalam tujuan atau penalti dapat meningkatkan solusi yang diusulkan AI. Kerangka kerja desain bersama AI-manusia hibrida dapat ditingkatkan dengan memasukkan lingkaran dalam dan luar untuk pembuatan prototipe. Lingkaran dalam berisi BO atas ukuran transistor, kode DAC bias, dan kapasitor beban dengan target multi-tujuan. Namun, lingkaran luar terdiri dari suntingan manusia dalam topologi atau pilihan bias, seperti pemilihan sel tunda, penggunaan kembali arus, dan catu daya yang diatur.

**Pembelajaran Transfer Lintas Teknologi**

Pembelajaran transfer lintas teknologi untuk VCO cincin secara langsung mengatasi kenyataan bahwa setiap node CMOS baru menggeser dinamika transistor, perilaku sel tunda, dan karakteristik kebisingan. Perancang seringkali perlu mengkarakterisasi ulang VCO dari awal dengan simulasi SPICE yang berat. Menggunakan kembali model yang dipelajari dari satu node teknologi ke node teknologi lainnya akan mempercepat optimasi. Deskriptor yang tidak bergantung pada teknologi seperti GM/ID dan fT dapat digunakan sebagai pengganti metode penentuan ukuran W/L mentah. Model terlatih ini dapat menjadi tulang punggung untuk pendekatan pemodelan multi-node yang akan mengurangi waktu kalibrasi dengan node teknologi baru. Model pengganti dapat dilatih pada data SPICE 65 nm, disempurnakan dengan kumpulan data kecil dari simulasi 28 nm, dan menggunakan model ini untuk mempelajari hukum penskalaan. Pembelajaran transfer dapat dilakukan dari simulasi ke simulasi, pengukuran ke simulasi, dan lintas topologi. Topologi lintas-teknologi berarti bahwa jika Anda melatih model Anda pada RVCO 3 tahap dalam 65 nm, Anda dapat mentransfernya ke RVCO 5

tahap dalam 28 nm. Untuk ini, cukup lakukan fine-tuning pada bagian yang menangani penskalaan frekuensi osilasi.

**Jaringan Saraf Berbasis Fisika (PINN)**

PINN adalah jaringan saraf yang dilatih tidak hanya pada data, tetapi juga pada persamaan fisika yang mengaturnya (persamaan diferensial, kondisi batas, hukum konservasi). Pendekatan pemodelan jaringan saraf berbasis fisika dapat mengkodekan kendala keadaan tunak. Ini juga dapat digunakan untuk makromodel fase untuk evaluasi noise fase, kendala kemiringan, dan kondisi batas. VCO cincin sangat nonlinier, dan kinerjanya bergantung pada dinamika penundaan (inverter atau tahap diferensial), sumber noise (termal, flicker, kopling suplai), nonlinieritas tegangan kontrol terhadap frekuensi (KVCO), noise fase, dan variasi PVT.

Metode pemodelan konvensional, seperti SPICE, Verilog-A perilaku, atau persamaan derau fasa stokastik, terlalu lambat atau terlalu disederhanakan. PINN dapat menjembatani kesenjangan ini dengan menyematkan persamaan fisika sel tunda, memberlakukan kondisi periodisitas osilator, dan mempelajari pemetaan nonlinier KVCO dari simulasi atau pengukuran yang lebih sedikit. Secara keseluruhan, PINN dapat berfungsi sebagai model pengganti yang ringkas dan menghormati fisika untuk VCO cincin yang memprediksi dinamika osilasi, derau fasa, dan perilaku KVCO dengan simulasi yang jauh lebih sedikit sambil memungkinkan penyetelan adaptif yang digerakkan oleh AI secara real-time.

**Arsitektur VCO Adaptif Loop Tertutup**

VCO adalah osilator loop terbuka, sehingga setiap variasi dalam proses, catu daya, atau suhu secara langsung menggeser frekuensi dan noise fase. Pada PLL, pergeseran ini memengaruhi rentang penguncian, stabilitas loop (karena KVCO bervariasi, ia mengubah penguatan loop), dan noise fase serta jitter. Desain VCO konvensional memiliki rentang penyetelan yang lebih luas dan guard banding, yang memboroskan daya dan meningkatkan noise. Arsitektur VCO loop tertutup adaptif dapat mengoptimalkan silikon sendiri untuk mengatasi PVT, noise catu daya, dan penuaan. Misalnya, jitter/noise fase dapat dideteksi melalui sampler D-FF sederhana dan konverter waktu ke digital. Sensor riak pada catu daya dapat digunakan untuk pemantauan termal, dan sensor headroom arus ekor untuk area operasi yang aman. Jenis arsitektur VCO adaptif loop tertutup adalah sebagai berikut.

**VCO Adaptif Terkunci Frekuensi**

Detektor frekuensi daya rendah membandingkan frekuensi VCO dengan referensi dalam VCO adaptif terkunci frekuensi. Sinyal kesalahan menyetel arus bias atau kapasitansi beban hingga frekuensi target tercapai. Sekarang, jaga agar VCO tetap berada di tengah rentang penyetelan untuk variasi KVCO minimal.

**VCO Adaptif Sadar Noise Fasa**

Dalam VCO adaptif sadar noise fasa, sensor noise fasa on-chip digunakan, yang secara adaptif menyesuaikan bias, headroom suplai, atau arus ekor untuk meminimalkan noise fasa. Ini dapat secara dinamis menyeimbangkan daya dan noise.

**Bias Adaptif (VCO Bias Sendiri)**

Pada tipe ini, rangkaian bias merasakan amplitudo/frekuensi osilasi dan mengatur arus ekor. Loop tertutup memastikan ayunan konstan dan stabilisasi frekuensi meskipun terjadi drift PVT, seperti loop bias replika dan kekuatan inverter adaptif pada VCO cincin.

**Penyetelan Adaptif dengan Loop Kalibrasi**

Kalibrasi dapat dilakukan menggunakan penyetelan kasar dan halus. Pada penyetelan kasar, loop kalibrasi latar belakang digital mengunci frekuensi VCO di dekat pita target. Pada penyetelan halus, tegangan kontrol (dari PLL) menangani penyesuaian kecil. Kalibrasi mengurangi nonlinieritas KVCO dan meningkatkan linearitas kurva penyetelan, menstabilkan dinamika PLL.

**VCO Adaptif Suplai**

Fluktuasi pada suplai dideteksi, yang memodulasi bias untuk mempertahankan penundaan konstan per sel. Ini meningkatkan penekanan jitter yang disebabkan oleh suplai PSRR. Ini juga dapat diatur dengan menanamkannya dalam loop atau menggunakan konverter on-chip.

**Loop Adaptif yang dibantu Pembelajaran Mesin**

Dalam loop adaptif yang dibantu ML, model pengganti digunakan untuk memprediksi bias yang benar, yang dapat dipelajari oleh model untuk pemetaan ke kondisi PVT yang berbeda.

## 7.6 PENGALI APROKSIMASI BERDAYA RENDAH UNTUK PEMROSESAN GAMBAR

RVCO merupakan komponen penting dalam synthesizer frekuensi modern, ADC, dan sistem komunikasi lainnya karena ukurannya yang terukur, ukurannya yang kompak, dan memiliki rentang penyetelan yang lebih tinggi. Namun, trade-off desain yang saling terkait pada RVCO membuat proses optimasi menjadi manual dan tidak praktis dalam skenario teknologi canggih dan lintas teknologi. Sebagai pengganti pendekatan konvensional, metode berbasis AI menyediakan ruang yang menawarkan optimasi simultan dari berbagai parameter RVCO. Berbagai algoritma dibahas, seperti EA, BO, DL, RL, dan pendekatan pemodelan pengganti. Metode-metode ini secara signifikan meningkatkan waktu siklus desain, noise fase, dan FoM dibandingkan dengan metode konvensional. Beberapa tantangan dalam pendekatan berbasis AI telah dibahas; namun, pendekatan ini membantu memajukan integrasi alur kerja, pembangkitan data, dan generalisasi. Pendekatan desain bersama hibrida AI-Manusia diusulkan sebagai arah masa depan, yang dapat membantu dalam transfer learning dan pemodelan berbasis fisika dari pendekatan berbasis AI untuk desain RVCO yang tangguh. Secara keseluruhan, pendekatan berbasis AI untuk optimasi menghadirkan langkah transformatif untuk metode desain yang efisien, terukur, dan tangguh untuk sistem analog mixed signal dan RF generasi berikutnya.

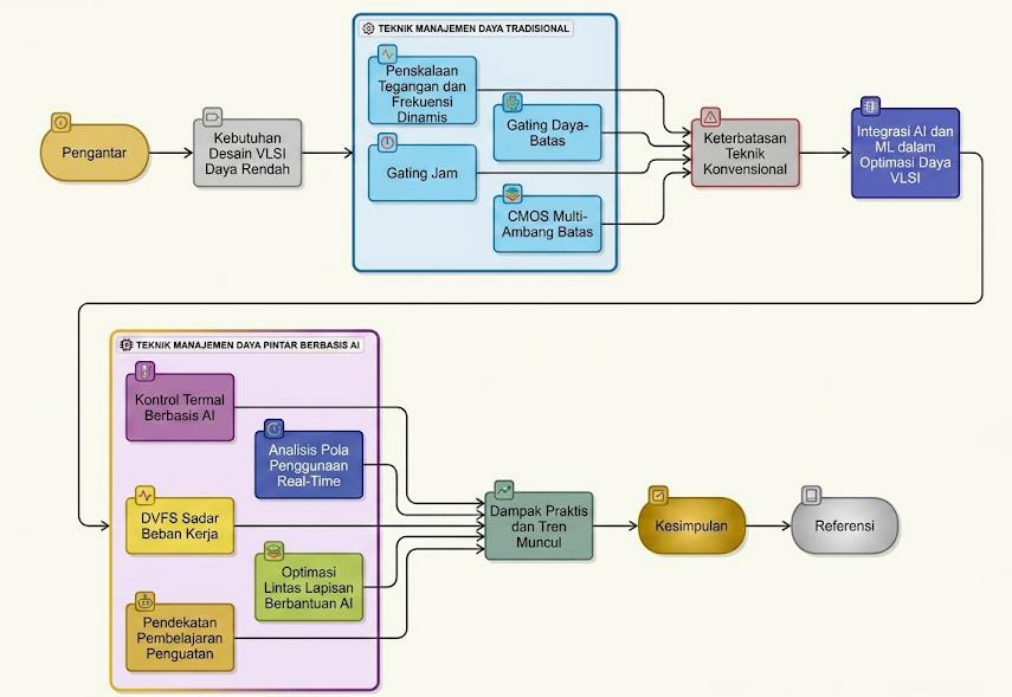
# BAB 8
# MANAJEMEN DAYA ADAPTIF BERBASIS AI DALAM DESAIN VLSI

## 8.1 MANAJEMEN DAYA VLSI BERBASIS AI

Abad ke-21 telah menyaksikan proliferasi teknologi digital yang belum pernah terjadi sebelumnya, dan ini sebagian besar didasarkan pada desain Integrasi Skala Sangat Besar (VLSI). Di dunia saat ini, baik itu dari ponsel pintar dan perangkat yang dapat dikenakan serta kendaraan otonom hingga komputasi awan dan penerapan Internet of Things (IoT), semuanya dimungkinkan oleh sirkuit VLSI yang memuat miliaran transistor per chip. Teknologi VLSI merupakan fondasi industri semikonduktor, inti dari segmen bisnis bernilai triliunan dolar dalam ekonomi global, dan bertanggung jawab atas terobosan dalam kecerdasan buatan, komunikasi 5G, realitas tertambah, dan antarmuka ke komputasi kuantum. Dengan meningkatnya kompleksitas komputasi, disipasi daya telah muncul sebagai salah satu kendala desain kritis untuk sirkuit VLSI. Persyaratan daya rendah dipengaruhi oleh permintaan untuk mencapai masa pakai baterai optimal untuk sistem portabel, oleh tantangan dalam mengelola masalah termal dengan lingkungan komputasi berkinerja tinggi, dan oleh tuntutan global untuk mengurangi konsumsi energi oleh sistem elektronik. Faktanya, pusat data menyumbang sebagian besar listrik di seluruh dunia dan oleh karena itu desain VLSI yang hemat energi merupakan tantangan teknik dan sekaligus suatu keharusan karena alasan lingkungan.

Dalam upaya untuk melampaui keterbatasan ini, penerapan Kecerdasan Buatan (AI) dan Pembelajaran Mesin (ML) untuk manajemen daya VLSI telah menjadi pilihan yang semakin relevan. Teknik berbasis AI dapat mencapai optimasi daya yang adaptif, cerdas, dan real-time serta mengatasi batasan yang diberlakukan oleh teknik konvensional. Sebagian besar teknik berbasis AI memanfaatkan pemodelan prediktif, pembelajaran penguatan, dan optimasi lintas lapisan dan dapat memungkinkan AI untuk mengelola daya secara dinamis pada tingkat sistem, arsitektur, dan sirkuit. Hal ini membuat manajemen daya menjadi kurang tetap dan malah menjadikannya sistem yang aktif dan mengoptimalkan diri sendiri dengan kemampuan untuk memenuhi berbagai kebutuhan aplikasi yang beragam secara tak terduga.

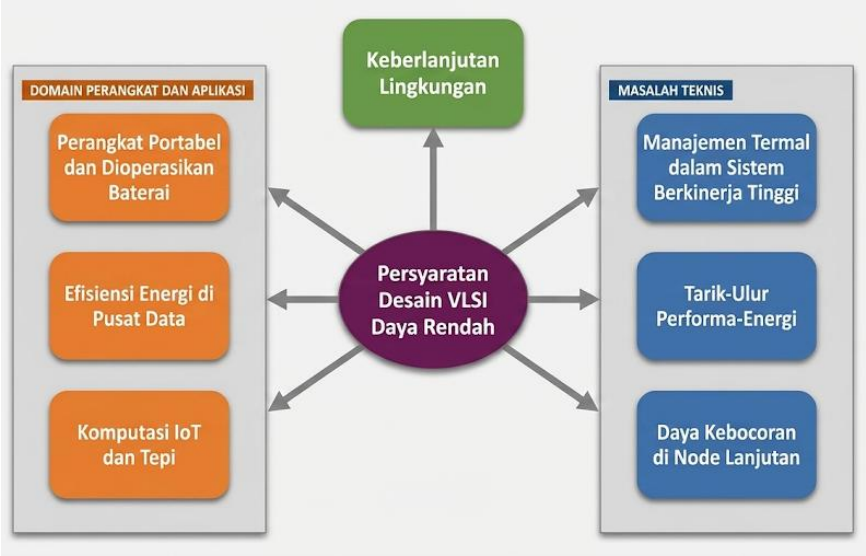
Bab ini mengkaji perkembangan manajemen daya dalam desain VLSI, dimulai dengan motivasi dan kebutuhan operasi daya rendah; menjelaskan pendekatan optimasi daya tradisional, dan selanjutnya, keterbatasan pendekatan tersebut; diikuti dengan fokus pada kerangka kerja manajemen daya adaptif yang didukung AI, menjelaskan prinsip kerja, aplikasi, keunggulan, dan tantangannya. Bab ini, yang menyoroti evolusi pendekatan manajemen daya, terutama menunjukkan bagaimana pemanfaatan pendekatan yang didukung AI akan menghasilkan pergeseran domain menuju masa depan sistem berbasis VLSI berkinerja tinggi dan berkelanjutan, yang didukung oleh otomatisasi jangka panjang, yang diperkenalkan di dunia komputasi yang terus berkembang (Gambar 8.1).




**Gambar 8.1** Garis Besar Bab.

## 8.2 PERSYARATAN DESAIN VLSI DAYA RENDAH

Dengan perkembangan pesat teknologi semikonduktor, kita membutuhkan desain VLSI daya rendah saat ini dan bahkan lebih lagi di masa depan karena meningkatnya kebutuhan akan efisiensi energi serta kinerja komputasi. Dengan setiap generasi baru penskalaan perangkat, jumlah transistor telah tumbuh secara eksponensial dan menghasilkan kepadatan daya yang lebih besar, dengan manajemen energi menjadi bagian penting dari masalah desain. Jadi, untuk sistem saat ini yang membutuhkan desain daya rendah, daya rendah bukan lagi tujuan optimasi desain, tetapi merupakan persyaratan untuk penerapan operasi aktual di berbagai domain aplikasi (Gambar 8.2).




**Gambar 8.2** Persyaratan desain VLSI daya rendah.

**Perangkat Portabel dan Bertenaga Baterai**

Ledakan perangkat portabel, termasuk ponsel pintar, laptop, elektronik yang dapat dikenakan, dan implan biomedis, telah menghasilkan tren penting menuju desain VLSI daya rendah. Karena perangkat portabel ditenagai baterai, sangat berguna untuk mengurangi konsumsi energi untuk mencapai masa pakai baterai yang lebih baik, dan dengan demikian, pengalaman pengguna yang lebih baik. Teknologi baterai yang memungkinkan kemampuan waktu penggunaan yang lebih lama hampir tidak meningkat dengan peningkatan kepadatan penyimpanan, karena kepadatan energi baterai hanya meningkat sedikit dibandingkan dengan peningkatan eksponensial dalam pemrosesan. Dengan demikian, sirkuit terpadu yang hemat energi lebih dibutuhkan.

Aplikasi yang membutuhkan komputasi berkinerja tinggi dalam ukuran kecil meliputi game seluler, streaming video waktu nyata, dan pemantauan status kesehatan secara terus menerus. Tanpa mempertimbangkan daya, banyak fitur akan dengan cepat menguras baterai, sehingga membuat perangkat portabel menjadi tidak praktis. Oleh karena itu, penting untuk menggunakan metode tingkat sirkuit yang berbeda, yaitu penskalaan tegangan, pembatasan clock, dan perubahan arsitektur. Karena kita dapat mencapai waktu penggunaan yang lebih lama dari perangkat bertenaga baterai, hal ini menambah portabilitas, kenyamanan, dan kegunaan.

**Iot dan Komputasi Tepi**

IoT mengharapkan miliaran node yang tersedia untuk diintegrasikan ke dalam kehidupan sehari-hari (perangkat pintar, sensor yang dapat dikenakan, pemantauan lingkungan, dan otomatisasi industri). Sejumlah besar perangkat ini ditempatkan di lokasi terpencil dan tidak terjangkau, dan karena menggunakan teknologi baterai, penggantian baterai tidak dapat dilakukan sesering mungkin, sehingga pengoperasian daya ultra-rendah menjadi suatu keharusan untuk implementasinya.

Untuk memenuhi persyaratan ini, desain VLSI untuk node IoT mengejar metode untuk pemanenan energi dan pengoperasian di bawah ambang batas. Banyak perangkat tepi menggunakan sumber ambien seperti matahari, RF, atau getaran, yang hanya dapat menyediakan sejumlah kecil energi secara berkala. Sirkuit yang hemat energi dan beban kerja adaptif membuat desain daya ultra-rendah diperlukan untuk pengoperasian jangka panjang. Oleh karena itu, desain daya ultra-rendah memungkinkan penerapan IoT dalam proporsi yang sangat besar sekaligus memungkinkan penskalaan tanpa perawatan yang berlebihan.

**Manajemen Termal dalam Sistem Berkinerja Tinggi**

Dorongan tanpa henti untuk prosesor, GPU, dan akselerator AI kelas atas guna mencapai kepadatan transistor dan frekuensi operasi tertinggi juga menyebabkan kepadatan daya yang lebih tinggi, yang mengakibatkan titik panas dan peningkatan pembuangan panas. Masalah termal tidak hanya akan memengaruhi kinerja karena pembatasan kinerja, tetapi juga akan mengurangi keandalan jangka panjang karena mekanisme keausan akibat tekanan pada perangkat semikonduktor dipercepat. Desain daya rendah sangat penting dalam mengatasi masalah ini melalui pengurangan daya dinamis dan kebocoran untuk membatasi

pembangkitan panas di sumbernya. Bersama dengan pengembangan sirkuit yang berorientasi daya, strategi manajemen termal yang inovatif akan memungkinkan solusi termal yang meningkatkan pembuangan panas daripada strategi termal konsumsi energi tinggi. Hal ini sangat penting untuk aplikasi kaya data seperti komputasi awan, game, dan sistem penggerak otonom, yang bergantung pada kinerja berkelanjutan di bawah anggaran termal yang terbatas.

**Daya Bocor pada Node Canggih**

Seiring dengan penyusutan node proses hingga dimensi nanometer dan sub-10 nm, arus bocor menjadi bagian yang lebih besar dari konsumsi daya keseluruhan. Saat transistor diperkecil dimensinya, tegangan ambang berkurang, dan ketebalan oksida juga berkurang secara bersamaan, yang meningkatkan kebocoran arus subambang dan gerbang di seluruh terminal. Semua ini menimbulkan tantangan signifikan untuk meminimalkan daya siaga rendah pada sirkuit terpadu kontemporer. Teknik desain daya rendah, seperti Multi-Threshold CMOS (MTCMOS), body biasing, dan power gating, umumnya digunakan untuk mengurangi daya bocor. Opsi ini memungkinkan sirkuit untuk tetap berjalan sambil memastikan tidak ada aliran arus saat dalam mode idle. Mengurangi daya bocor sangat penting untuk perangkat yang selalu terhubung, seperti prosesor seluler dan node IoT, karena perangkat tersebut paling sering berada dalam mode siaga untuk jangka waktu yang lama. Oleh karena itu, ada peningkatan permintaan untuk kontrol daya bocor yang maksimal dalam desain VLSI.

**Efisiensi Energi di Pusat Data**

Pusat data modern merupakan fondasi layanan cloud, AI, dan aplikasi komputasi berkinerja tinggi. Pusat data memiliki kemampuan komputasi yang sangat besar dengan konsumsi daya yang sama besarnya, seringkali mencapai gigawatt di fasilitas besar. Sebagian besar energi ini hilang sebagai panas, yang membutuhkan sistem pendingin tambahan, yang menambah penghematan operasional dan dampak lingkungan. Desain VLSI berdaya rendah dapat membantu meningkatkan kinerja per watt dari prosesor, akselerator, dan perangkat keras jaringan di pusat data. Bahkan peningkatan kecil dalam efisiensi daya chip dapat menghasilkan pengurangan besar dalam konsumsi listrik di infrastruktur pusat data. Selain itu, pengurangan Pengeluaran Operasional (OPEX) bersama dengan infrastruktur komputasi yang lebih berkelanjutan memungkinkan ekosistem cloud dan AI on-demand untuk tumbuh tanpa mengorbankan komitmen energi global kita.

**Keberlanjutan Lingkungan**

Ekonomi digital kini menjadi salah satu kontributor terbesar terhadap penggunaan listrik global dan emisi karbon. Dengan miliaran perangkat elektronik aktif, dan peningkatan lalu lintas data yang pesat yang belum pernah terjadi sebelumnya dalam sejarah, jejak lingkungan industri semikonduktor tumbuh dengan cepat. Dengan demikian, desain VLSI hemat daya tidak lagi hanya dapat didefinisikan sebagai tantangan teknis—kini telah menjadi tanggung jawab sosial dan lingkungan. Dengan mengurangi disipasi daya pada tingkat perangkat desain VLSI, perancang VLSI berkontribusi terhadap komputasi hijau dan tujuan keberlanjutan internasional. Ketika konsumsi daya berkurang, emisi gas rumah kaca berkurang, permintaan pada jaringan listrik berkurang, dan material yang dibutuhkan untuk infrastruktur pendinginan berkurang. Komitmen netralitas karbon baik dari perusahaan

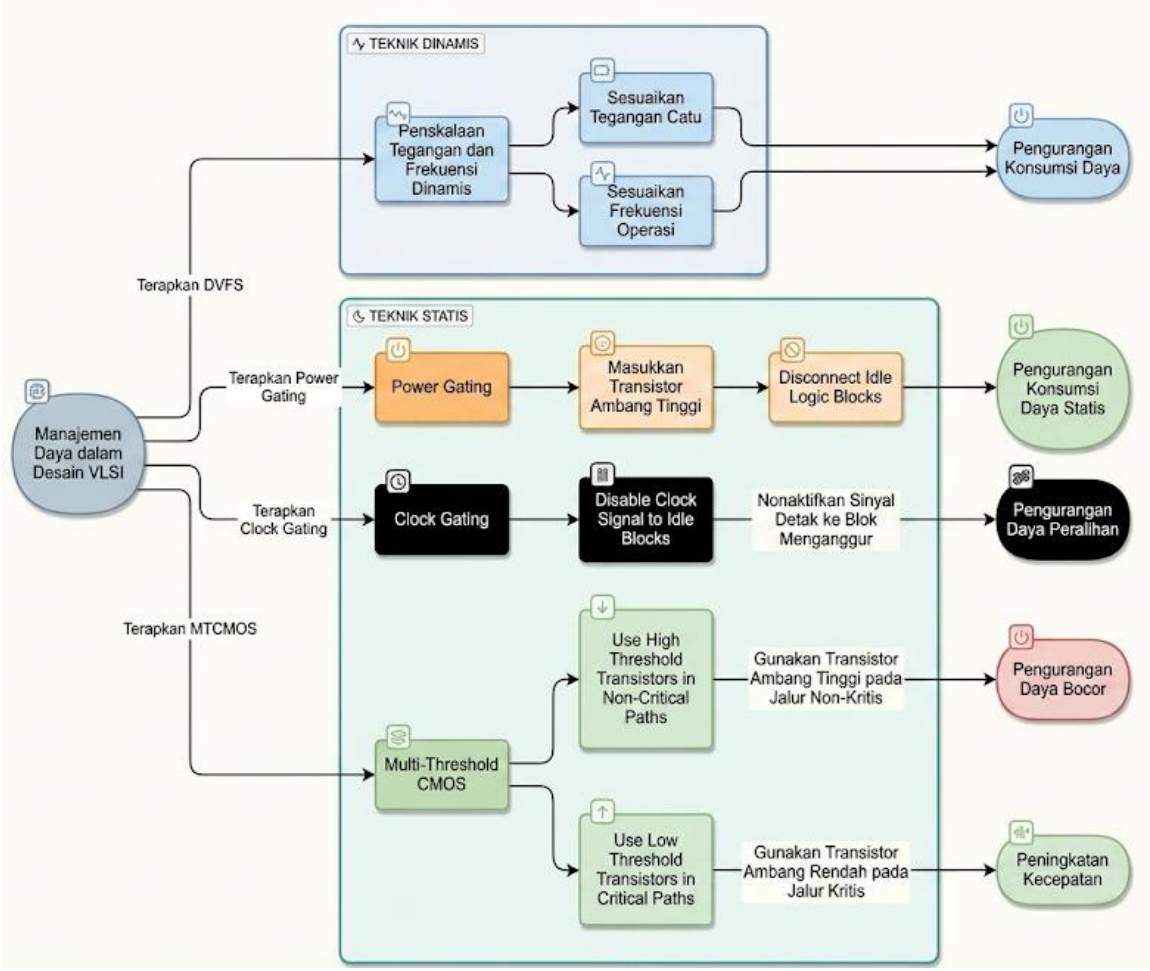
maupun pemerintah saat ini dilakukan dalam skala yang belum pernah terjadi sebelumnya dalam sejarah, dan desain sirkuit hemat energi akan menjadi pendorong utama masa depan digital yang berkelanjutan.

**Kompromi Kinerja-Energi untuk Aplikasi Generasi Berikutnya**

Domain aplikasi baru seperti AI, realitas tertambah (AR), realitas virtual (VR), mengemudi otonom, dan komunikasi 5G/6G membutuhkan kinerja komputasi yang belum pernah ada sebelumnya, tetapi perlu beroperasi dalam anggaran daya yang sangat terbatas, khususnya ketika diterapkan di edge atau untuk platform portabel. Hal ini menciptakan kompromi kompleks antara kinerja, presisi, dan eksekusi yang hemat energi. Desain VLSI daya rendah memungkinkan kemajuan yang belum pernah terjadi sebelumnya dari serangkaian akselerator khusus, unit pemrosesan neuromorfik, dan akselerator inferensi AI yang dapat dirancang untuk throughput dengan konsumsi daya minimum. Melalui tren ini, sistem generasi berikutnya dapat dirancang dengan cara yang tidak terbatas dan terbebas dari batasan suhu dan biaya energi. Akibatnya, desain sirkuit yang hemat daya akan menjadi mesin inovasi di berbagai pasar.

Di seluruh cakupan pertimbangan yang luas yang menciptakan kebutuhan akan desain VLSI berdaya rendah, penggunaan jangka panjang perangkat portabel, fungsionalitas jangka panjang node IoT, manajemen panas, dan pengurangan kebocoran, hingga keberlanjutan pusat data, itulah semua pertimbangan teknologi. Selain dimensi teknologi, ada tujuan keberlanjutan yang murah dan jangka panjang untuk planet kita secara keseluruhan, dan daya rendah akan menjadi sangat penting untuk keberhasilan implementasi aplikasi generasi berikutnya: misalnya, AI, AR/VR, dan komunikasi tanpa 5G/6G, masing-masing contoh ini perlu diseimbangkan antara efektivitas kinerja dan efektivitas daya agar dapat berfungsi.

Masing-masing aplikasi ini menunjukkan kredibilitas pemodelan dan desain VLSI berdaya rendah sebagai komponen penting dari semikonduktor modern dan menunjukkan perlunya teknik optimasi canggih generasi berikutnya yang melampaui metode tradisional. Aplikasi baru seperti AI, AR, VR, mobil otonom, dan komunikasi 5G/6G semuanya membutuhkan daya komputasi yang belum pernah terjadi sebelumnya dalam sejarah dan perlu dijalankan dengan anggaran daya yang sangat ketat, terutama jika dijalankan di perangkat tepi atau perangkat portabel (Gambar 8.3).




**Gambar 8.3** Berbagai teknik manajemen daya tradisional.

## 8.3 TEKNIK MANAJEMEN DAYA TRADISIONAL DALAM DESAIN VLSI

### Penskalasian Tegangan dan Frekuensi Dinamis (DVFS)

Penskalasian Tegangan dan Frekuensi Dinamis (DVFS) adalah teknik manajemen daya dinamis yang banyak digunakan dalam sistem VLSI yang menyesuaikan tegangan operasi dan frekuensi clock dari suatu sirkuit atau prosesor secara real-time sesuai dengan tuntutan beban kerja. Dengan DVFS, sistem beroperasi pada frekuensi dan tegangan tertinggi selama periode permintaan komputasi tinggi untuk mempertahankan kinerja. Pada periode aktivitas rendah atau keadaan idle, sirkuit akan menurunkan tegangan dan frekuensi, sehingga secara substansial mengurangi daya karena daya dinamis sebanding dengan $V^2 f$.

$$P_{dinamik} = \alpha C V^2 f$$

di mana α adalah faktor aktivitas switching, $C$ adalah kapasitansi switching, $V$ adalah tegangan suplai, dan $f$ adalah frekuensi clock. Jelas bahwa konsumsi daya sebanding dengan kuadrat tegangan suplai dan bergantung secara linier pada frekuensi. Oleh karena itu, mengurangi tegangan dan frekuensi selama beban kerja ringan menghasilkan penghematan energi yang substansial.

**Prinsip Operasi**

a) **Pemantauan Beban Kerja:**
   I. Sistem terus memantau pemanfaatan prosesor, antrian instruksi, atau kebutuhan tugas waktu nyata.
   II. Prediktor beban kerja atau penghitung kinerja digunakan untuk memperkirakan permintaan komputasi di masa mendatang.

b) **Keputusan Penskalaan Kinerja:**
   I. Jika beban kerja tinggi (misalnya, menjalankan algoritma AI yang kompleks), sistem meningkatkan frekuensi dan tegangan untuk mempertahankan kinerja.
   II. Jika beban kerja ringan (misalnya, tugas latar belakang yang tidak aktif atau sederhana), frekuensi dan tegangan dikurangi untuk menghemat daya.

c) **Penyesuaian Tegangan dan Frekuensi:**
   I. Frekuensi clock dimodifikasi menggunakan phase-locked loop (PLL) atau pembagi frekuensi.
   II. Tegangan suplai disesuaikan menggunakan regulator tegangan on-chip atau IC manajemen daya dinamis.
   III. Penskalaan frekuensi harus disinkronkan dengan penskalaan tegangan:
      - **Penurunan Skala**: Frekuensi dikurangi terlebih dahulu, kemudian tegangan.
      - **Peningkatan Tegangan**: Tegangan ditingkatkan terlebih dahulu, kemudian frekuensi.

   Urutan ini mencegah kesalahan waktu karena tegangan yang tidak mencukupi pada frekuensi yang lebih tinggi.

d) **Stabilisasi Sistem:**
   I. Setelah penyesuaian, sistem kembali beroperasi normal pada tingkat tegangan dan frekuensi baru.
   II. Transisi diulang secara adaptif seiring fluktuasi beban kerja.

**Mode Operasi DVFS**

- **Peningkatan Tegangan Mode Aktif**: Menyesuaikan kinerja secara terus menerus selama eksekusi program untuk menyeimbangkan kecepatan dan energi.
- **Peningkatan Tegangan Mode Idle/Standby**: Menurunkan frekuensi dan tegangan secara drastis ketika sistem menunggu input atau dalam keadaan idle.
- **Peningkatan Tegangan Prediktif**: Menggunakan algoritma ML atau prediksi beban kerja untuk mengantisipasi kebutuhan kinerja sebelumnya.

**Keunggulan**

- Meningkatkan tegangan dan frekuensi suplai berdasarkan beban kerja, memberikan fleksibilitas untuk mengorbankan kinerja demi penghematan daya secara real-time.
- Mengurangi daya dinamis secara drastis, mengingat daya berbanding lurus dengan V2f, sehingga sangat menguntungkan untuk perangkat seluler dan tertanam.
- Meningkatkan masa pakai baterai dengan mengurangi sumber daya selama periode idle atau permintaan rendah tanpa memengaruhi pengalaman pengguna secara signifikan.

- Membantu manajemen termal dengan mengurangi panas yang dihasilkan selama beban kerja yang lebih ringan, sehingga menghindari pembatasan termal.
- Sangat serbaguna dan banyak digunakan untuk ponsel pintar, laptop, prosesor tertanam, dan sistem komputasi awan skala besar.

**Keterbatasan**

- Peralihan antara keadaan tegangan dan frekuensi menimbulkan latensi yang dapat berdampak buruk pada aplikasi waktu nyata dan aplikasi penting.
- Membutuhkan prediksi beban kerja yang tepat; perkiraan yang tidak akurat menyebabkan penskalaan yang tidak efisien dan pemborosan daya.
- Menjadi tidak efektif pada node teknologi canggih di mana daya kebocoran menjadi lebih dominan daripada daya dinamis.
- Peningkatan tegangan yang berlebihan dapat menyebabkan pelanggaran waktu dan masalah keandalan sirkuit, sehingga membatasi rentang operasi yang aman.

**Power Gating**

Tujuan power gating adalah untuk meminimalkan konsumsi daya statis (kebocoran), yang menjadi mode konsumsi daya dominan dalam teknologi sub-mikron. Power gating diimplementasikan dengan transistor tidur ambang tinggi yang digunakan untuk memutus blok logika yang tidak digunakan dari catu daya ketika blok logika ini tidak digunakan. Ketika blok sirkuit ini dimatikan, hal ini menyebabkan arus kebocoran yang diminimalkan. Power gating bekerja sangat baik dalam desain *system-on-chip* (SoC), di mana unit fungsional tidak beroperasi pada waktu yang bersamaan. Namun, pengawasan yang tepat diperlukan untuk memastikan bahwa penundaan bangun tidak merugikan kinerja.

**Prinsip Operasi**

**a. Kebocoran Arus pada Blok Idle:**

I. Bahkan ketika sirkuit dalam keadaan idle, transistor terus mengalami kebocoran arus karena konduksi subambang dan kebocoran oksida gerbang.

II. Seiring dengan penurunan skala teknologi di bawah 90 nm, kebocoran menjadi sebanding atau lebih besar daripada daya dinamis, sehingga menjadi perhatian utama.

**b) Penyisipan Transistor Tidur:**

I. Transistor besar, yang dikenal sebagai transistor tidur, disisipkan di antara blok sirkuit dan rel daya/ground.

II. Dua konfigurasi umum adalah:

- **Sakelar Header (PMOS):** Terhubung antara Vdd dan blok.
- **Sakelar Footer (NMOS):** Terhubung antara blok dan ground.

**c) Mode Aktif vs. Mode Tidur:**

I. **Mode Aktif**: Transistor tidur dihidupkan, menyediakan jalur konduktif antara rel daya dan sirkuit, memungkinkan operasi normal.

II. **Mode Tidur**: Transistor tidur dimatikan, memutus jalur ke catu daya/ground, sehingga mencegah kebocoran arus pada blok idle.

**d) Retensi Status:**

I. Karena mematikan blok akan menghapus data yang tersimpan, sel retensi status (flip-flop atau SRAM) sering digunakan untuk menyimpan informasi penting selama mode tidur.

II. Ketika blok bangun, data dipulihkan dengan cepat tanpa inisialisasi ulang.

**e) Proses Bangun:**

I. Untuk mengembalikan blok ke operasi, transistor tidur dihidupkan kembali, mengembalikan daya penuh.

II. Desain harus hati-hati menangani arus lonjakan selama bangun untuk menghindari penurunan tegangan atau gangguan suplai.

**Mode Power Gating**

- Power Gating Kasar: Seluruh unit fungsional atau inti dimatikan. Misalnya, mematikan inti GPU yang tidak aktif di ponsel pintar.
- Power Gating Halus: Blok kecil (misalnya, file register, bank memori, unit aritmatika) dimatikan secara selektif. Ini menawarkan penghematan daya yang lebih tinggi tetapi menambah kompleksitas desain.

**Keunggulan**

- Mengontrol daya ke blok sirkuit yang tidak dibutuhkan, secara signifikan mengurangi daya bocor, yang penting dalam desain submikron yang dalam.
- Memberikan penghematan energi di domain dengan aktivitas idle yang lama, seperti IoT atau perangkat wearable bertenaga baterai.
- Efektif di semua tingkat desain, dari blok fungsional kecil hingga inti prosesor penuh, memberikan pilihan desain.
- Bekerja sama dengan DVFS dan clock gating dan menyediakan metodologi pengurangan daya yang lebih lengkap untuk SoC modern.

**Keterbatasan**

- Membangunkan blok yang di-gated memiliki latensi, yang dapat menyebabkan penurunan kinerja dalam aplikasi yang kritis waktu seperti sistem real-time.
- Membutuhkan flip-flop atau sel retensi untuk mempertahankan keadaan sistem, yang meningkatkan overhead area dan biaya desain.
- Transistor tidur dapat membuat tata letak, perutean, dan verifikasi lebih kompleks, serta meningkatkan kesulitan desain fisik.
- Penanganan transisi memerlukan desain yang cermat untuk menghindari arus lonjakan dan kesalahan fungsional saat penyalaan daya.

**Pengaturan Gerbang Jam**

Sinyal jam merupakan salah satu kontributor terbesar terhadap konsumsi daya dan energi dinamis dalam rangkaian VLSI, karena jam akan terus beralih meskipun logika yang menerima jam tersebut mungkin sedang tidak aktif. Pengaturan gerbang jam mengurangi aktivitas peralihan yang tidak perlu dengan mematikan sinyal jam ke bagian rangkaian yang tidak aktif melakukan komputasi. Dengan demikian, daya peralihan berkurang, sementara fungsionalitas rangkaian tetap terjaga. Pengaturan gerbang jam dapat digunakan pada

berbagai tingkatan, dari flip-flop individual hingga seluruh subsistem, dan sering digunakan dalam mikroprosesor dan prosesor sinyal digital (DSP).

**Prinsip Operasi**

**a) Daya Dinamis dalam Pohon Jam:**

I. Daya dinamis yang dikonsumsi dalam rangkaian CMOS diberikan oleh:

II. $P_{dynamic} = \alpha C V^2 f$

III. Jaringan jam memiliki aktivitas peralihan yang sangat tinggi (α ≈ 1) karena terus beralih, terlepas dari beban kerja.

IV. Menonaktifkan clock di bagian yang tidak aktif menghilangkan pengalihan yang berlebihan, mengurangi f (frekuensi efektif) untuk blok tersebut.

**b) Mekanisme Pengaktifan:**

I. Sel pengaktifan clock (gerbang logika atau sirkuit khusus) dimasukkan ke dalam jalur clock suatu blok.

II. Biasanya diimplementasikan sebagai:

- Gerbang AND atau gerbang OR dengan sinyal pengaktifan.
- Sel pengaktifan clock terintegrasi (ICG), yang menghindari gangguan dan dioptimalkan untuk overhead rendah.

III. Contoh:

- Jika Enable = 1, clock diteruskan ke sirkuit (mode aktif).
- Jika Enable = 0, clock diblokir, dan sirkuit tetap idle tanpa pengalihan yang tidak perlu.

**c) Keadaan Aktif vs. Idle:**

I. Keadaan Aktif: Blok menerima sinyal clock dan beroperasi normal.

II. Keadaan Idle: Blok terputus dari clock, menghilangkan daya switching sambil mempertahankan data yang tersimpan dalam flip-flop.

**d) Granularitas Aplikasi:**

I. Pengaturan Waktu Halus (*Fine-Grain Clock Gating*): Diterapkan pada unit kecil seperti register, flip-flop, atau unit fungsional.

II. Pengaturan Waktu Kasar (*Coarse-Grain Clock Gating*): Diterapkan pada blok yang lebih besar, subsistem, atau seluruh inti prosesor.

**Mode Pengaturan Waktu (Clock Gating)**

- Pengaturan Waktu Kombinasional (*Combinational Clock Gating*): Menggunakan logika kombinasional (AND/OR) untuk mengatur waktu (clock) dengan sinyal pengaktifan (enable). Implementasinya sederhana tetapi dapat menyebabkan gangguan (glitches) jika perubahan diaktifkan saat waktu (clock) aktif.
- Pengaturan Waktu Sekuensial (*Sequential Clock Gating*): Waktu (clock) diatur berdasarkan apakah nilai register diharapkan berubah pada siklus berikutnya. Ini memberikan penghematan yang lebih tinggi dengan melewatkan pembaruan yang berlebihan, tetapi meningkatkan kompleksitas desain.

- Pengaturan Waktu Halus (*Fine-Grain Clock Gating*): Diterapkan pada tingkat yang sangat kecil seperti flip-flop, register, atau unit fungsional kecil. Ini dapat mencapai penghematan maksimum tetapi menambah area dan overhead kontrol yang signifikan.
- Pengaktifan Jam Berbutir Kasar: Menonaktifkan jam untuk seluruh blok atau subsistem seperti ALU, cache, atau inti prosesor. Lebih mudah diimplementasikan, tetapi kurang presisi, karena logika internal yang tidak aktif masih dapat mengonsumsi daya.

**Keuntungan**

- Sederhana, efektif, dan memiliki dukungan yang cukup besar dari alat sintesis.
- Mengurangi konsumsi daya dinamis sambil mempertahankan kinerja selama eksekusi aktif.
- Dapat digunakan pada tingkat granularitas halus (flip-flop, unit fungsional) atau granularitas kasar (seluruh blok).
- Memiliki overhead area yang rendah, dibandingkan dengan pengaktifan daya atau MTCMOS.

**Keterbatasan**

- Pengaktifan yang tidak tepat atau dirancang dengan buruk dapat menimbulkan gangguan, yang dapat menyebabkan kesalahan fungsional.
- Menambah kompleksitas dalam verifikasi waktu, karena jam yang diaktifkan perlu memenuhi persyaratan setup/hold.
- Peningkatan daya hanya terbatas pada daya dinamis, dan tidak berpengaruh pada daya bocor.
- Membutuhkan optimasi yang cermat dengan teknik lain (misalnya, DVFS) untuk memastikan efisiensi keseluruhan.

**CMOS Multi-threshold (MTCMOS)**

Teknik MTCMOS memanfaatkan transistor dengan tegangan ambang (*Vth*) yang berbeda untuk memberikan keseimbangan terbaik antara kinerja sirkuit dan daya bocor. Bagian sirkuit yang bukan jalur kritis akan berisi transistor dengan nilai Vth tinggi untuk meminimalkan daya bocor. Sebaliknya, bagian sirkuit yang merupakan jalur kritis akan berisi transistor dengan nilai *Vth* rendah untuk memaksimalkan kecepatan sirkuit. Oleh karena itu, dengan ambang ganda (*Vth*), sirkuit mampu mencapai daya bocor rendah serta kinerja tinggi. MTCMOS dapat sangat menguntungkan dalam mendesain memori daya rendah dan aplikasi digital besar di mana daya bocor mungkin menjadi faktor dominan.

**Prinsip Operasi**

a) **Pemilihan Tegangan Ambang:**
   I. **Transistor *Vth* Rendah**:
      - Beralih lebih cepat, mengurangi penundaan.
      - Digunakan pada jalur kritis kinerja di mana kecepatan sangat penting.
      - Kelemahan: Arus bocor lebih tinggi saat idle.
   II. **Transistor *Vth* Tinggi**:
      - Beralih lebih lambat tetapi kebocoran arus jauh lebih sedikit.
      - Digunakan pada jalur non-kritis di mana kelonggaran waktu tersedia.

- Kelemahan: Menambah sedikit penundaan.

Dengan mencampur kedua jenis tersebut, perancang mencapai kompromi antara kebocoran dan kinerja.

**b) Transistor Tidur di MTCMOS:**

I. Dalam banyak implementasi MTCMOS, transistor tidur *Vth* tinggi (baik header PMOS atau footer NMOS) dimasukkan antara rel daya dan rangkaian logika.
   - Mode Aktif: Transistor tidur dihidupkan, menghubungkan rangkaian logika ke Vdd atau ground. Transistor *Vth* rendah di dalam logika memberikan kinerja tinggi.
   - Mode Tidur: Transistor tidur dimatikan, memutus jalur kebocoran dan secara signifikan mengurangi daya siaga.

**c) Retensi Data dan Bangun:**

I. Karena sirkuit internal kehilangan daya dalam mode tidur, elemen retensi status (seperti flip-flop retensi atau bank SRAM kecil yang selalu aktif) digunakan untuk menjaga data penting.
II. Saat bangun, transistor tidur diaktifkan kembali, memulihkan operasi normal.

**Mode MTCMOS**

- **MTCMOS Butir Halus**: Setiap sel standar atau blok logika kecil memiliki transistor tidurnya sendiri untuk kontrol kebocoran. Ini memberikan pengurangan kebocoran yang tinggi tetapi menambah kompleksitas area, perutean, dan verifikasi.
- **MTCMOS Butir Kasar**: Sekelompok blok logika berbagi transistor tidur yang lebih besar untuk power gating. Ini mengurangi overhead desain dan lebih efisien dalam hal area tetapi menawarkan kontrol kebocoran yang kurang granular.

**Keunggulan**

- Menciptakan keseimbangan yang solid antara kinerja dan pengurangan kebocoran.
- Sangat mudah diskalakan di berbagai node teknologi.
- Dapat digunakan sebagai teknik pendamping dengan power gating untuk penghematan daya yang lebih besar.
- Membantu dalam optimasi desain untuk konsumsi daya aktif dan siaga, tidak seperti DVFS yang terutama berfokus pada daya aktif.

**Kekurangan**

- Membutuhkan pemilihan perangkat yang cermat mulai dari ambang batas rendah hingga tinggi.
- Meningkatkan kompleksitas biaya manufaktur dan fabrikasi karena implantasi ambang batas ganda.
- Dapat menciptakan hambatan kinerja dalam desain jika terlalu banyak perangkat Vth tinggi digunakan pada jalur kritis.
- Membutuhkan alat EDA canggih untuk mencapai optimasi waktu dan kebocoran (Gambar 8.4).




**Gambar 8.4** Keterbatasan teknik manajemen daya konvensional.

**Keterbatasan Teknik Manajemen Daya Konvensional**

Meskipun teknik desain VLSI populer seperti DVFS, power gating, clock gating, dan MTCMOS sangat populer, ada sejumlah kekurangan yang diamati dalam teknik ini dan menghambat efisiensinya dalam teknologi canggih dan aplikasi yang baru muncul. Kekurangan-kekurangan ini dapat diringkas sebagai berikut:

a) **Ketidakefektifan pada Node Tingkat Lanjut**: Ketika perangkat ditingkatkan skalanya ke domain sub-nanometer dan bahkan nanometer, arus bocor mendominasi konsumsi daya total. Clock gating dan DVFS terutama berkaitan dengan daya dinamis, dengan daya bocor sebagian besar tetap tidak tersentuh dengan cara konvensional. Selain itu, variasi tegangan ambang untuk node tingkat lanjut membuat teknik konvensional untuk daya ultra-rendah menjadi kurang efektif.

a) **Overhead Transisi**: Metode seperti penskalaan tegangan atau frekuensi dan power gating semuanya memiliki waktu peralihan yang lebih lama antara mode operasi. Power gating adalah contoh kasus di mana latensi bangun dan arus masuk menyebabkan gangguan pada operasi normal. Gangguan serupa pada operasi normal muncul dengan DVFS melalui stabilisasi tegangan dan frekuensi suplai. Overhead ini membatasi aplikasi pada sistem real-time yang kritis waktu dan ketat dengan persyaratan ketat untuk respons kinerja segera.

b) **Peningkatan Kompleksitas Desain**: Mekanisme daya rendah seringkali memerlukan transistor tambahan, sel retensi, atau rangkaian kontrol, yang berarti peningkatan kompleksitas desain. Kompleksitas tambahan ini juga terkait dengan peningkatan upaya verifikasi, yang tentu saja menyebabkan peningkatan upaya pengujian dan biaya area. Selain peningkatan verifikasi dan pengujian, kemampuan manufaktur juga akan terpengaruh, dan potensi biaya desain dapat meningkat dengan metodologi daya rendah, yang sangat relevan dalam pekerjaan komersial.

c) **Masalah Skalabilitas**: Penganggaran daya di berbagai modul dalam desain SoC besar dengan miliaran transistor merupakan tantangan yang besar. Misalnya, power gating

membutuhkan transistor sleep yang besar dan membutuhkan retensi status yang tidak sebanding dengan ukuran desain. Hal yang sama berlaku untuk perangkat MTCMOS, yang memerlukan penerapan perangkat ambang batas yang berbeda yang mempersulit fabrikasi dan tata letak. Oleh karena itu, skalabilitas akan tetap menjadi tantangan inheren untuk penerapan teknik ini secara efektif dalam sistem yang sangat terintegrasi.

d) **Keterbatasan Adaptabilitas**: Pendekatan tradisional sebagian besar menggunakan kebijakan kontrol statis atau yang telah ditentukan sebelumnya, yang tidak dapat beradaptasi secara otomatis terhadap beban kerja yang berubah dengan cepat. Misalnya, aplikasi multimedia, ML, dan IoT biasanya tidak dapat diprediksi penggunaannya. Teknik daya rendah tetap dapat membuang energi atau merusak kinerja. Oleh karena itu, ketidakmampuan untuk beradaptasi dengan beban kerja yang berubah mengganggu operasi optimal lingkungan komputasi heterogen saat ini.

e) **Kompromi Keandalan dan Kinerja**: Penerapan agresif pendekatan tradisional seringkali menyebabkan kompromi yang memengaruhi kinerja dan keandalan jangka panjang suatu sirkuit. Misalnya, mengurangi tegangan suplai dalam DVFS dapat meningkatkan sensitivitas terhadap variasi proses dan pelanggaran waktu. Penggunaan transistor suplai ambang tinggi yang tidak efisien dalam MTCMOS ambang ganda dapat meningkatkan penundaan propagasi di jalur kritis, sementara pengalihan bias ambang ganda dari suplai dalam power gating menciptakan kebisingan yang dapat mengikis keandalan. Bias ini dapat membuat kinerja tampak tidak konsisten atau nominal dalam berbagai kondisi operasi.

f) **Tuntutan Lingkungan dan Energi**: Karena adopsi layanan cloud, pusat data, dan penerapan IoT yang cepat, konsumsi energi global oleh elektronik telah menjadi masalah penting. Langkah-langkah tradisional saja tidak akan memberikan efisiensi energi dan keberlanjutan yang dibutuhkan, terutama mengingat seruan untuk komputasi hijau dan pengurangan jejak karbon. Diperlukan langkah-langkah yang lebih canggih dan sesuai konteks untuk melengkapi langkah-langkah tradisional, guna memenuhi kebutuhan energi dan lingkungan di masa depan.

## 8.4 INTEGRASI AI DAN ML DALAM OPTIMASI DAYA VLSI

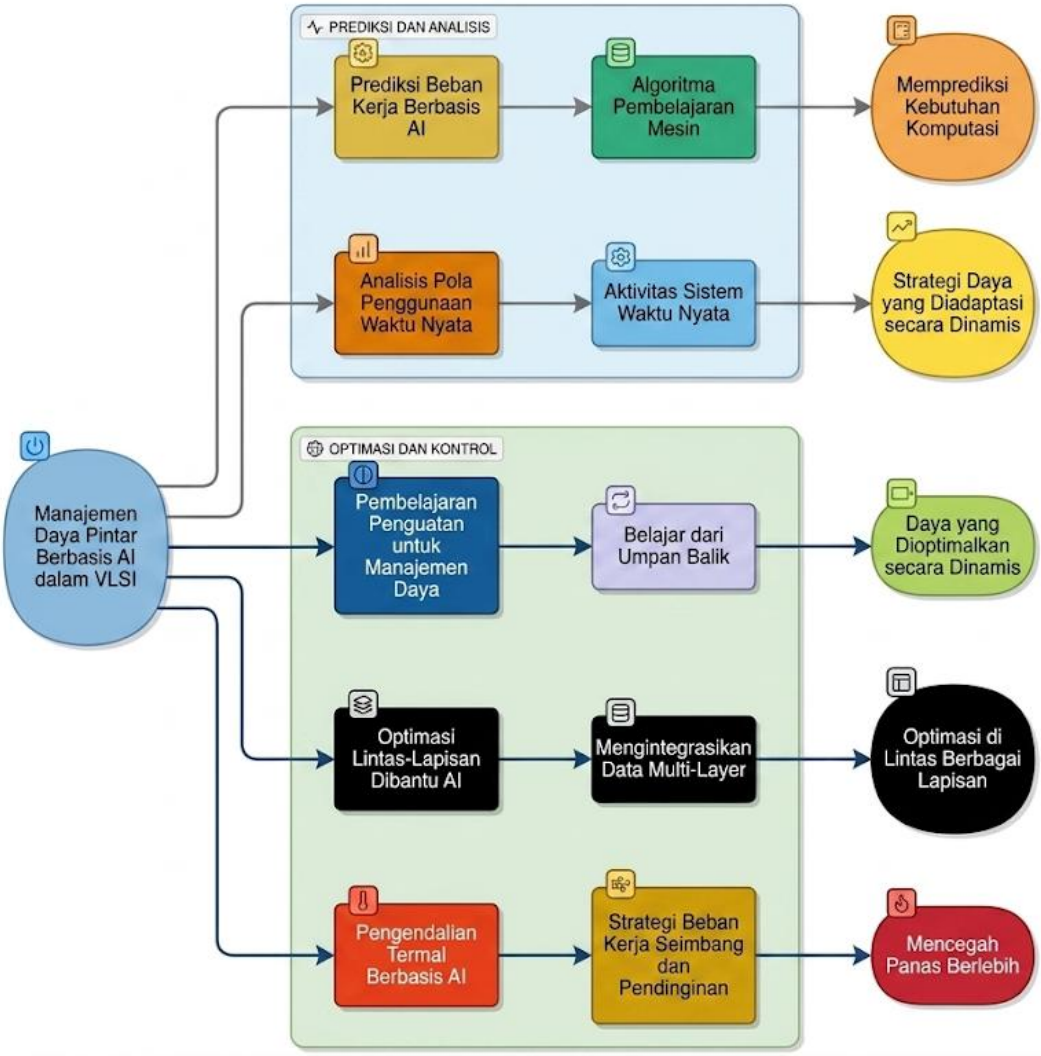
Keterbatasan metode manajemen daya kontemporer jelas menunjukkan kesenjangan yang semakin besar antara apa yang dapat dicapai oleh metode tradisional dan persyaratan yang lebih ketat dari sistem VLSI terbaru. Seiring kemajuan teknologi semikonduktor ke skala sub-mikron dan nano-meter yang sangat kecil, faktor dominan berupa daya bocor, bersamaan dengan peningkatan kompleksitas desain, dan beban kerja yang semakin stokastik, telah mengungkap keterbatasan metode optimasi statis atau yang telah ditentukan sebelumnya. Meskipun pendekatan seperti DVFS, power gating, clock gating, dan MTCMOS sangat efektif di masa lalu, pendekatan tersebut bergantung pada model statis yang gagal merespons secara dinamis sifat heterogen dari beban kerja komputasi modern.

Salah satu tantangan utama adalah nondeterminisme beban kerja. Sebagai contoh, beban kerja baru yang muncul saat ini meliputi pemrosesan multimedia waktu nyata, inferensi AI, mobil otonom, dan augmented reality atau VR, yang semuanya memiliki trade-off daya-kinerja yang sangat berbeda dari komputasi tujuan umum tradisional. Metode statis tidak cukup fleksibel untuk menangani beban kerja ini dengan baik. Selain itu, skala sistem VLSI baru yang mencakup miliaran transistor dan elemen pemrosesan heterogen membuat pengelolaan daya melalui penugasan atau metode berbasis aturan menjadi tidak layak.

Masalah lain adalah meningkatnya pentingnya keberlanjutan dan efisiensi energi global secara keseluruhan. Karena pusat data, infrastruktur IoT, dan perangkat seluler mewakili proporsi yang lebih besar dari konsumsi energi di seluruh dunia, teknik tradisional tidak memberikan optimasi skala luas dan sadar konteks yang dibutuhkan untuk memfasilitasi komputasi berkelanjutan. Clock gating dan power gating mengurangi daya dalam konteks lokal, tetapi teknik ini tidak dapat melakukan optimasi di berbagai lapisan arsitektur, beban kerja, dan konteks penggunaan secara bersamaan. Hal ini menunjukkan perlunya strategi manajemen daya yang lebih cerdas, adaptif, dan sadar konteks, yang dapat mengoptimalkan perilaku sistem secara real-time. Teknik AI dan ML mendasari sistem yang akan menutup kesenjangan ini dengan memungkinkan sistem untuk belajar dari beban kerja historis, beradaptasi secara dinamis terhadap kondisi operasi saat ini, dan memprediksi kebutuhan di masa mendatang. Berbeda dengan teknik tradisional, solusi berbasis AI secara inheren fleksibel, dan dapat mengoptimalkan pada beberapa lapisan abstraksi—sirkuit, arsitektur, dan sistem—sekaligus.

Teknik manajemen daya yang didukung AI tidak hanya memberikan prediksi beban kerja dengan akurasi yang lebih tinggi, tetapi juga memfasilitasi pengambilan keputusan yang lebih proaktif. Misalnya, pembelajaran penguatan terus meningkatkan strategi melalui interaksinya dengan lingkungan sistem dengan mempelajari kebijakan optimal dari waktu ke waktu. DVFS yang sadar beban kerja dan optimasi lintas lapisan berbasis AI memiliki karakteristik yang serupa, di mana keduanya memungkinkan sistem untuk mengoptimalkan kinerja atau daya sesuai dengan kondisi dunia nyata alih-alih bergantung pada asumsi terburuk secara teoritis. Karakteristik ini menjadikan manajemen daya efisien dan dapat diterapkan sesuai permintaan, dengan kepastian bahwa kinerja dan keandalan tidak pernah terganggu dan konsumsi energi yang boros dihindari.

Dengan kata lain, transisi dari teknik tradisional ke teknik berbasis AI merupakan pergeseran paradigma karena teknik tradisional dapat diimplementasikan tetapi kurang cerdas dan tidak fleksibel untuk diterapkan pada VLSI saat ini. Metode AI mengatasi keterbatasan tersebut, menanamkan kecerdasan pada teknik tradisional, dan mengubah manajemen daya menjadi proses yang cerdas, responsif, dan ramah lingkungan melalui pembelajaran, prediksi, dan optimasi mandiri. Secara harfiah dan metaforis, integrasi AI bukan sekadar perluasan teknik yang tersedia; ini adalah evolusi yang diperlukan untuk mengejar sistem komputasi generasi berikutnya (Gambar 8.5).




**Gambar 8.5** Teknik manajemen daya cerdas berbasis AI.

## 8.5 TEKNIK MANAJEMEN DAYA CERDAS BERBASIS AI

### Prediksi Beban Kerja Berbasis AI

Prediksi beban kerja berbasis AI adalah salah satu teknik terbaik untuk kontrol daya adaptif pada sistem VLSI saat ini. Dorongannya adalah untuk melatih algoritma ML agar dapat memprediksi kebutuhan komputasi suatu sistem jauh di masa depan, sehingga teknik kontrol daya proaktif, bukan reaktif, seperti DVFS, power gating, dan penjadwalan tugas, dapat diterapkan.

### Prinsip Operasi

a) **Pengumpulan Data**: Sistem mengamati karakteristik runtime seperti penggunaan CPU, urutan akses memori, cache hit/miss, campuran instruksi, lalu lintas jaringan, dan suhu. Karakteristik runtime ini dicatat secara terus menerus dan selanjutnya berfungsi sebagai data masukan untuk melatih model AI/ML.

a) **Identifikasi Fitur**: Karakteristik beban kerja diidentifikasi dari data yang dikumpulkan. Untuk prosesor seluler, ini bisa berupa frame rate, aktivitas sensor, atau penggunaan

aplikasi latar belakang; Untuk pusat data, hal tersebut dapat berupa intensitas kueri, permintaan I/O, atau perilaku tingkat kedatangan pekerjaan.

b) **Pelatihan Model**: Algoritma AI/ML dilatih untuk memprediksi intensitas beban kerja di masa mendatang, dan berbagai algoritma pembelajaran mesin dapat digunakan, misalnya, model regresi, pohon keputusan, jaringan saraf berulang (RNN), atau jaringan memori jangka pendek panjang (LSTM) tertutup. Pelatihan dapat dilakukan secara offline menggunakan dataset historis atau online, terus menerus meningkatkan dan beradaptasi.

c) **Prediksi Beban Kerja**: Selama waktu berjalannya beban kerja, model yang terlatih memprediksi permintaan beban kerja jangka pendek dan menengah (misalnya, beban komputasi rendah, menengah, atau tinggi pada jendela waktu berikutnya). Prediksi ini memungkinkan sistem untuk membuat keputusan proaktif tentang berapa banyak daya dan sumber daya kinerja yang akan disediakan.

d) **Manajemen Daya Adaptif**: Prediksi tersebut digunakan dalam modul optimasi daya tingkat tinggi seperti pengontrol DVFS, logika gerbang jam, dan penjadwal tugas. Misalnya, jika beban kerja diprediksi akan meningkat tajam, tegangan/frekuensi akan ditingkatkan terlebih dahulu; jika beban kerja diprediksi akan tetap idle, power gating dapat diaktifkan.

**Keuntungan**

- Menyesuaikan status daya lebih tepat dengan kebutuhan beban kerja dibandingkan dengan DVFS statis.
- Menskalakan penggunaan sumber daya berdasarkan permintaan yang diantisipasi, bukan menanggapi perubahan setelah terjadi.
- Sangat bermanfaat bagi perangkat seluler dan IoT dengan batasan energi yang ketat.
- Menyembunyikan overhead kinerja yang terkait dengan pengelolaan kinerja dan efisiensi—di luar jeda sekitar satu detik, pengguna tidak akan mengalami dampak throughput dan produktivitas yang jelas.

**Keterbatasan**

- Membutuhkan model prediksi beban kerja yang akurat, yang mungkin bervariasi di berbagai aplikasi.
- Inferensi waktu nyata oleh model ML dapat menimbulkan biaya komputasi.
- Beban kerja yang sangat dinamis atau tidak teratur masih dapat menyebabkan kesalahan prediksi sesekali.
- Implementasi DVFS yang sadar beban kerja pada sistem heterogen besar (CPU + GPU + akselerator) menambah kompleksitas desain.

**Pembelajaran Penguatan untuk Manajemen Daya**

Pembelajaran Penguatan (Reinforcement Learning/RL) adalah subset dari ML di mana agen belajar untuk melakukan tindakan dalam suatu lingkungan untuk memaksimalkan imbalan kumulatif. Dalam kasus manajemen daya VLSI, lingkungannya adalah sistem itu sendiri, dan agennya adalah pengontrol daya. Tidak seperti heuristik tetap, pengontrol

berbasis RL akan mempelajari kebijakan manajemen daya optimal di dalam perangkat keras, beserta beban kerjanya.

**Prinsip Operasi**

a) **Menyiapkan Lingkungan dan Agen**: Lingkungan dalam hal ini adalah sistem VLSI, yang mencakup prosesor, cache, memori, dan periferal. Agen dalam lingkungan ini adalah pengontrol manajemen daya (model AI), yang bertanggung jawab atas tindakan terkait daya seperti penskalaan tegangan/frekuensi, pengaktifan/penonaktifan blok, dan pengendalian pendinginan.

a) Representasi Keadaan: Keadaan sistem mencakup berbagai statistik: pemanfaatan CPU, tingkat kedatangan tugas, panjang antrian, suhu, tegangan, dan jenis beban kerja yang sedang dieksekusi. Pada setiap titik keputusan, keadaan ini menjadi masukan bagi agen RL.

b) **Tindakan**: Agen dapat mengambil berbagai tindakan. Dalam kasus paling sederhana, agen dapat memilih tingkat frekuensi (DVFS) untuk diterapkan pada beban kerja. Agen juga dapat mengambil tindakan untuk memicu power gating pada blok idle, menerapkan clock gating, atau secara aktif menetapkan ulang beban kerja untuk menyeimbangkan aktivitas. Setiap tindakan yang diambil agen akan mengubah kondisi operasi sistem, yang kemudian tercermin dalam daya dan kinerja.

c) **Fungsi Imbalan**: Fungsi imbalan dirancang untuk menyeimbangkan penghematan daya dan persyaratan kinerja. Misalnya, fungsi imbalan dapat menghukum pemborosan energi, pelanggaran termal, dan tenggat waktu yang terlewat, sementara memberikan imbalan untuk penghematan daya dan memenuhi batasan kualitas layanan (QoS). Tujuannya adalah untuk melakukan hal yang benar berdasarkan penghematan daya di seluruh waktu operasional, bukan hal yang mudah berdasarkan tujuan jangka pendek.

d) **Mempelajari Representasi**: Agen RL akan mencoba berbagai tindakan dan mengamati implikasi kinerja yang dihasilkan dalam hal daya dan kinerja, kemudian mempertimbangkan keadaannya. Seiring waktu, ia akan mempelajari kebijakan (pemetaan keadaan ke tindakan yang diharapkan) yang memaksimalkan imbalan kumulatif yang diharapkan. Algoritma umum yang digunakan meliputi Q-learning, Deep Q-Networks (DQN), atau metode Policy Gradient.

e) **Adaptasi**: Karena beban kerja dan kondisi operasi bervariasi, RL mampu beradaptasi secara dinamis. Misalnya, pada prosesor seluler, agen dapat mempelajari bahwa pemutaran video membutuhkan tingkat kinerja sedang dengan daya sedang, sementara permainan membutuhkan kinerja tinggi, yang akan memerlukan lebih banyak daya untuk skalabilitas.

**Keuntungan**

- Mengadaptasi kebijakan daya secara efektif dari waktu ke waktu dibandingkan dengan heuristik statis dan yang ditentukan pengguna.
- Dapat menyesuaikan diri dengan beban kerja yang berubah dan tidak dapat diprediksi, yang menjadikan komputasi heterogen paling cocok.

- Secara bersamaan menyeimbangkan berbagai tujuan (daya, kinerja, dan stabilitas termal) dengan memungkinkan fungsi penghargaan yang beragam.
- Menerapkan intervensi manusia yang lebih sedikit karena aspek pembelajaran mandiri dari kebijakan yang berkembang berdasarkan umpan balik sistem.

**Kerugian**

- Melibatkan eksplorasi dan keterlibatan yang cukup besar dengan sistem, yang mungkin menghasilkan keputusan yang buruk di awal.
- Sistem VLSI modern cenderung melibatkan banyak variabel (beban kerja, suhu, dan tegangan), sehingga sulit untuk dipelajari tanpa menyediakan teknik pengurangan keadaan yang efisien.
- Selain itu, untuk mengimplementasikan agen RL, mungkin diperlukan komputasi dan memori tambahan, yang dengan caranya sendiri akan mengkonsumsi energi.
- Perhatian utama kami adalah apakah agen tersebut belajar cukup cepat dari data yang dikumpulkannya untuk memungkinkan perangkat beradaptasi dan bekerja secara real-time.

**Optimasi Lintas Lapisan yang Dibantu AI**

Optimasi lintas lapisan berarti mengoptimalkan semua tingkat abstraksi dari sistem VLSI – mulai dari tingkat sirkuit (transistor, sel), hingga tingkat arsitektur (prosesor, cache, interkoneksi), dan sampai ke tingkat sistem/aplikasi (beban kerja, sistem operasi, interaksi pengguna). Secara tradisional, setiap lapisan ini telah dioptimalkan secara independen, yang seringkali menyebabkan trade-off yang suboptimal. Optimasi lintas lapisan yang dibantu AI memungkinkan koordinasi semua lapisan ini, menggunakan algoritma pembelajaran untuk memodelkan interaksi antar lapisan dan menemukan strategi manajemen daya optimal global.

**Prinsip Operasi**

a) Pengumpulan Data di Berbagai Tingkat: Metrik dikumpulkan di berbagai tingkat.
   I. **Pada Tingkat Sirkuit**: arus bocor, aktivitas switching, dan tegangan ambang.
   II. **Pada Tingkat Arsitektur**: cache miss, stall dalam pipeline, dan utilisasi inti.
   III. **Pada Tingkat Sistem/Aplikasi**: tenggat waktu penyelesaian tugas, persyaratan QoS yang dapat diterima, dan pola interaksi pengguna.

b) **Fitur dan Pemodelan**: AI akan mengambil fitur-fitur ini di berbagai tingkatan untuk menginformasikan model seperti jaringan saraf dalam (DNN), kerangka kerja RL, atau model jaringan Bayesian. Model tersebut memperhitungkan saling ketergantungan seperti hasil penurunan tegangan suplai (sirkuit), pengaruhnya terhadap perilaku cache (arsitektur), dan latensi aplikasi (sistem).

c) **Pengambilan Keputusan di Berbagai Tingkat**: Setelah model dilatih, AI memberikan saran untuk optimasi daya di berbagai tingkatan sebagai fungsi satu sama lain. Misalnya, model tersebut tidak hanya dapat menyarankan penurunan tegangan, tetapi juga dapat menjadwalkan tugas (tingkat OS) dalam urutan yang berbeda atau memodifikasi kebijakan cache sambil memberikan kinerja yang sama, tetapi dengan daya yang lebih rendah.

d) **Optimasi Perulangan**: Sistem akan terus menerus memberikan data runtime kembali ke model AI. Seiring waktu, AI akan beradaptasi dengan pola beban kerja, kondisi termal, dan proses penuaan perangkat sambil secara dinamis menyesuaikan kebijakannya tentang optimasi.

**Keunggulan**

- Memberikan optimasi global alih-alih peningkatan lokal pada lapisan individual, yang memberikan tingkat efisiensi yang lebih luas.
- Berubah secara modular, oleh karena itu dapat menyesuaikan diri dengan beban kerja heterogen, konteks pengguna, dan perubahan lingkungan.
- Akibatnya, ia dapat memaksimalkan ketergantungan (misalnya, penjadwalan tugas + DVFS) yang tidak dapat ditangkap oleh pendekatan silo konvensional.
- Sistem ini dapat diskalakan ke SoC yang kompleks dan platform heterogen, sehingga memungkinkan penggunaan akselerator AI/ML modern dan perangkat edge.

**Keterbatasan**

- Mensimulasikan interaksi antar lapisan dapat memakan banyak sumber daya dan membutuhkan sejumlah besar data.
- Menjalankan model AI secara real-time dapat memperparah penggunaan sumber daya perangkat keras dan perangkat lunak.
- Kebijakan lintas lapisan bahkan lebih sulit untuk diverifikasi karena perubahan pada satu lapisan dapat memiliki konsekuensi yang tidak terduga pada lapisan lain.
- Memperluas optimasi ke sistem yang sangat besar (misalnya, pusat data dan modul multi-chip) yang mungkin heterogen masih merupakan tantangan.

**Analisis Pola Penggunaan Waktu Nyata**

Analisis pola penggunaan waktu nyata didefinisikan sebagai penggunaan model AI/ML yang memantau penggunaan sistem elektronik yang sedang berlangsung (aplikasi yang dijalankan, interaksi pengguna dengan elemen GUI, dll., serta perangkat keras yang mungkin aktif) dan membuat kebijakan waktu nyata untuk memfasilitasi manajemen daya ([38–40]). Analisis pola penggunaan waktu nyata berbeda dari mode penghematan daya statis yang berjalan selama aktivitas pengguna, dan akan merespons secara waktu nyata terhadap perubahan skenario pengguna tanpa memerlukan konfigurasi pengguna individual. Analisis pola penggunaan waktu nyata mendukung efisiensi energi dalam elektronik tanpa mengorbankan pengalaman pengguna.

**Prinsip Operasi**

a) **Pengumpulan Data Waktu Nyata**: Berbagai komponen sistem menghasilkan data yang sesuai dengan penggunaan waktu nyata, penggunaan CPU/GPU, paralelisme tingkat instruksi, aktivitas akses memori, tingkat hit/miss cache, aktivitas sensor pada perangkat IoT, data aktivitas pengguna (interaksi sentuh, interaksi suara, atau idle). Data perilaku tersebut dimasukkan ke unit pemantauan sehingga dapat diproses, idealnya secara waktu nyata.

a) **Deteksi Pola dengan AI/ML**: Algoritma berbasis pembelajaran mesin (misalnya, pengelompokan, pembelajaran mendalam, dan model deret waktu) dapat secara

otomatis mendeteksi pola penggunaan yang sedang berlangsung. Misalnya, sistem mempelajari bahwa saat pengguna sedang menjelajahi web, beban kerja ringan dan dapat beroperasi pada frekuensi yang lebih rendah, sedangkan saat bermain video game, beban kerja berat.

b) **Penyesuaian Daya Dinamis**: Dengan menggunakan pola yang terdeteksi, model AI akan menyesuaikan tingkat daya secara real-time. Selama aktivitas rendah, frekuensi clock dan tegangan diturunkan. Selama beban kerja berat, tingkatkan inti berkinerja tinggi. Selama beban kerja idle, power gating atau deep sleep.

c) **Loop Umpan Balik Adaptif**: Sistem menilai apakah penyesuaian mencapai tujuan kinerja dan energi. Jika tidak, model AI akan menyesuaikan parameternya, meningkatkan akurasi dengan setiap penggunaan.

**Keunggulan**

- Beradaptasi segera dengan kondisi penggunaan dunia nyata.
- Mempelajari perilaku spesifik pengguna untuk penghematan energi yang disesuaikan.
- Mengurangi pemborosan penyediaan energi yang berlebihan.
- Bekerja di berbagai perangkat, dari ponsel pintar hingga pusat data.

**Keterbatasan**

- Jika tidak dikelola dengan benar, pemantauan penggunaan dapat mengakibatkan tantangan privasi dan keamanan.
- Analisis waktu nyata akan membutuhkan pengembangan model yang efisien dalam penggunaan daya.
- Umpan balik cepat dari pola penggunaan yang tidak terduga dan sementara dapat membuat analitik waktu nyata kurang efektif untuk jangka waktu terbatas.
- Integrasi di berbagai tingkat perangkat keras dan perangkat lunak tidaklah mudah.

**Pengendalian Termal Berbasis AI**

Respon terhadap manajemen termal telah menjadi perhatian utama bagi perancang VLSI karena kepadatan transistor yang tinggi dan tuntutan kinerja menyebabkan lebih banyak panas. Penghilangan panas yang tidak memadai dapat menurunkan tidak hanya kinerja perangkat tetapi juga mempercepat mekanisme keausan seperti elektromigrasi, ketidakstabilan suhu bias negatif (NBTI), dan kerusakan oksida. Teknik manajemen termal yang ada (yaitu, teknik pendinginan statis atau pembatasan termal tingkat tetap) seringkali konservatif dan reaktif, menyebabkan kinerja menurun meskipun kinerja keseluruhan dapat setara atau lebih unggul tanpa batasan. Kontrol termal berbasis AI menyediakan manajemen termal sistem yang secara adaptif dan proaktif mengelola panas berdasarkan kecerdasan prediktif, sambil memaksimalkan kinerja energi, keandalan, dan efisiensi.

**Prinsip Operasi**

a) **Pengukuran Suhu Waktu Nyata**: Sensor suhu yang terintegrasi ke dalam chiplet menyampaikan informasi terkini tentang keadaan termal titik panas, gradien termal, dan perubahan suhu di permukaan chip. Sistem ini juga mempertimbangkan perilaku sistem tipikal seperti daya, pengalihan, dan penggunaan.

b) **Model Prediksi Termal:** Model AI (misalnya, model jaringan saraf, model regresi, atau agen RL) digunakan untuk memprediksi termal chip berdasarkan beban kerja saat ini, konsumsi daya historis, dan/atau kondisi lingkungan beban kerja perangkat lunak. Misalnya, sebuah model dapat memprediksi bahwa beban kerja saat ini akan mendorong suhu titik panas CPU di atas kondisi operasi yang aman dalam beberapa milidetik.

c) **Tindakan Kontrol Termal Adaptif:** Setelah prediksi dibuat, sistem secara mulus memicu tindakan kontrol termal preventif, termasuk:

I. **Penskalaan Tegangan dan Frekuensi Dinamis**: Mengurangi frekuensi (dan karenanya daya/tegangan) inti tertentu untuk mengurangi pemanasannya.

II. **Migrasi Tugas**: Sistem memindahkan beban kerja dari titik panas ke titik yang lebih dingin.

III. **Kontrol Pendinginan Dinamis**: Sistem juga dapat secara dinamis menyesuaikan kecepatan kipas, intensitas pendinginan cairan, atau bahkan material antarmuka termal dalam sistem yang lebih besar.

IV. **Pembatasan Kecepatan/Daya**: Sistem juga dapat langsung menonaktifkan inti atau unit CPU lain yang tidak digunakan untuk beban kerja untuk mencegah unit tersebut berkontribusi pada pemanasan titik panas.

d) **Umpan Balik dan Pembelajaran Berkelanjutan**: Sistem AI memverifikasi apakah tindakan kontrol termal berhasil menjaga chip yang dibebani perangkat lunak di bawah batas termal target. Seiring waktu, sistem AI akan mempelajari korelasi antara beban kerja dan tindakan kontrol termal yang digunakan untuk mencegah batas termal. Dengan demikian, pengurangan inti idle dan pembatasan thread individual diperlukan untuk mengontrol keadaan termal chip.

**Keunggulan**

- Memprediksi titik panas sebelum terjadi untuk mencegah pembatasan mendadak.
- Mengurangi tekanan termal jangka panjang dan memperpanjang umur chip.
- Mengurangi biaya pendinginan yang tidak perlu (kipas, pendingin cair).
- Menyeimbangkan keamanan termal versus penurunan kinerja termal.

**Keterbatasan**

- Diperlukan pelatihan sistem AI pada berbagai beban kerja dan kondisi termal untuk prediksi yang akurat.
- Kepadatan sensor termal yang lebih besar akan menambah biaya dan luas area pada desain.
- Setiap model AI harus cukup cepat untuk mencegah panas berlebih akibat keputusan yang buruk.
- Karakteristik terkait profil termal dalam sistem heterogen besar (CPU, GPU, akselerator) tetap menjadi tantangan.

Tabel 8.1 menunjukkan perbandingan parameter seperti Akurasi Optimasi Daya, Kemampuan Adaptasi terhadap Beban Kerja, Kompleksitas Implementasi,

**Tabel 8.1** Perbandingan parameter berbagai teknik manajemen daya berbasis AI

| Teknik | Akurasi Optimasi Daya | Kemampuan Beradaptasi terhadap Beban Kerja | Kompleksitas Implementasi | Skalabilitas untuk Sistem Besar | Potensi Penghematan Energi | Latensi/ Waktu Respons | Aplikasi |
|---|---|---|---|---|---|---|---|
| Prediksi Beban Kerja Berbasis AI (AI-Based Workload Prediction) | Bergantung pada akurasi model | Bekerja lebih baik untuk beban kerja yang dapat diprediksi | Memerlukan proses **profiling** beban kerja dan pelatihan model | Dapat diskalakan ke sistem multi-core dan komputasi awan (*cloud*) | Tinggi | Dapat diterapkan secara **real-time** jika model cukup ringan | Smartphone, IoT, Server Cloud, Akselerator AI |
| Reinforcement Learning (RL) | Semakin meningkat seiring proses pelatihan | Mampu beradaptasi terhadap perubahan yang dinamis | Pelatihan dan validasi kebijakan (*policy validation*) cukup kompleks | Skalabilitas semakin sulit pada sistem multi-agen | Tinggi | Memerlukan waktu eksplorasi sebelum mencapai solusi optimal | Mobile SoC, Pusat Data (*Data Centers*), Perangkat Edge |
| Optimasi Lintas Lapisan (Cross-Layer Optimization) | Melakukan optimasi global pada seluruh lapisan sistem | Mencakup interaksi antar berbagai lapisan (*multi-level interactions*) | Membutuhkan dataset yang besar dan upaya desain yang tinggi | Sulit diterapkan pada sistem heterogen berskala besar | Sangat Tinggi | Terjadi penundaan akibat koordinasi antar lapisan | Sistem HPC, Kendaraan Otonom, SoC Heterogen |
| Analisis Pola Penggunaan Secara Real-Time (Real-Time Usage Pattern Analysis) | Bergantung pada tingkat keterprediksian perilaku pengguna | Bereaksi secara langsung terhadap perilaku pengguna | Membutuhkan sistem pemantauan (*monitoring*) yang efisien dan model ML | Model yang ringan memiliki skalabilitas yang baik | Sedang | Penyesuaian dapat dilakukan dengan cepat | Smartphone, Perangkat Wearable, Sensor IoT, Edge AI |
| Pengendalian Termal Berbasis AI (AI-Based Thermal Controlling) | Mampu memprediksi **hotspot** secara efektif | Beradaptasi terhadap variasi beban termal | Memerlukan sensor dan model AI | Skalabilitas termal pada SoC berukuran besar cukup kompleks | Tinggi | Proses prediksi dan pengendalian menambah sedikit latensi | Pusat Data, HPC, Sistem Otomotif, Prosesor Mobile |

Skalabilitas ke Sistem Besar, Potensi Penghematan Energi, Latensi/Waktu Respons, dan Aplikasi untuk berbagai teknik manajemen daya berbasis AI yang dibahas di atas.

## 8.6 DAMPAK PRAKTIS DAN TREN YANG MUNCUL

AI telah beralih dari teori ke implementasi aktual dalam mengelola penggunaan daya VLSI, dengan bukti konsep dalam praktik. Tren untuk sirkuit terpadu berkinerja tinggi dan

hemat energi terus meningkat pesat karena komputasi mobile, IoT, prosesor berkinerja tinggi, dan pusat data. Praktik manajemen daya berbasis AI memodifikasi sifat sistem semikonduktor modern untuk mengelola sumber daya energi dengan cara yang cerdas, adaptif, dan prediktif.

**Dampak Praktis Manajemen Daya Berbasis AI**

1. **Perangkat Mobile dan Wearable**: Peramalan beban kerja yang dibantu AI dan analitik tren konsumsi waktu nyata memperpanjang masa pakai baterai dengan menyesuaikan kinerja sistem pada saat itu sesuai dengan tindakan dan lokasi pengguna. Misalnya, ponsel pintar dapat memproyeksikan kebutuhan aplikasi yang digunakan (misalnya, bermain game, mencari informasi, atau menonton video) dan menyesuaikan jumlah energi yang dapat digunakan. Perangkat wearable juga mendapat manfaat dari mode daya sangat rendah yang secara mulus mengoptimalkan kemampuan untuk melanjutkan pemantauan kesehatan dalam jangka waktu lama tanpa perlu mengisi daya baterai secara teratur.
2. **Pusat Data dan Infrastruktur Cloud**: Pusat data skala besar mengonsumsi listrik dalam jumlah besar, dengan pendinginan menyumbang hampir 40% dari penggunaan energi. Kontrol termal berbasis AI dan optimasi lintas lapisan mengurangi kebutuhan pendinginan, menyeimbangkan beban kerja server, dan memastikan kepatuhan terhadap Perjanjian Tingkat Layanan (SLA), sehingga menghasilkan pengurangan biaya operasional dan jejak karbon yang signifikan.
3. **Komputasi Edge dan Sistem IoT**: Manajemen daya menggunakan AI merupakan faktor penting dalam perangkat IoT yang terbatas sumber daya. Perangkat di lokasi terpencil atau sulit dijangkau dapat bertahan selama bertahun-tahun dengan baterai dengan memprediksi pola beban kerja dan mengoptimalkan kondisi daya rendah. Prosesor edge yang memungkinkan inferensi AI pada perangkat memanfaatkan teknik ini untuk mencapai keseimbangan antara efisiensi daya dan kinerja.
4. **Aplikasi Otomotif dan Aplikasi yang Sangat Penting**: Untuk aplikasi otomotif, manajemen energi berbasis AI memungkinkan tercapainya keamanan optimal setiap saat dan pengeluaran energi minimal. Penggabungan sensor dan perencanaan jalur dioperasikan tanpa menggunakan sensor, sementara modul lain dimatikan dan memasuki mode daya rendah. Hal ini diperlukan untuk menjamin pengoperasian sistem yang andal, sekaligus membantu memfasilitasi peningkatan jangkauan, atau efisiensi Kendaraan Listrik Baterai (BEV) untuk mengurangi total konsumsi daya komputasi.
5. **Komputasi Kinerja Tinggi (HPC) dan Akselerator AI**: Dalam klaster HPC dan prosesor AI, GPU, TPU, atau banyak NPU, kita dapat memanfaatkan RL dengan DVFS yang sadar beban kerja sesuai dengan variasi permintaan komputasi. Hal ini menjamin throughput maksimal dan membantu mengurangi biaya konsumsi energi per operasi hingga jumlah yang wajar untuk melatih model pembelajaran mendalam skala besar tanpa biaya energi yang ekstrem.

**Tren yang Muncul dalam Manajemen Daya Berbasis AI**

1. **Komputasi Neuromorfik untuk AI Daya Ultra Rendah**: Arsitektur neuromorfik mengoptimalkan penggunaan energi berdasarkan desain, yang diciptakan dari otak manusia. Manajemen daya berbasis AI dengan chip neuromorfik juga memungkinkan konsumsi daya yang jauh lebih rendah daripada teknologi CMOS tradisional, yang sangat berharga dalam AI edge dan perangkat medis implan.
2. **Pembelajaran Terfederasi dan di Perangkat untuk Manajemen Daya**: Alih-alih pelatihan cloud penuh, semakin banyak perangkat yang menggunakan pembelajaran terfederasi dan AI edge untuk membiasakan diri dengan total konsumsi daya di lokasi. Hal ini memfasilitasi individualisasi yang lebih besar dan mengurangi latensi sambil menjaga privasi data dan mengurangi biaya energi yang terkait dengan komunikasi.
3. **Optimasi Lintas Disiplin (Desain Bersama Listrik-Termal-AI):** Metode baru mengusulkan optimasi ganda kebijakan daya berbasis AI bersama dengan kebijakan listrik dan termal. Chip yang akan datang akan menghadirkan solusi menyeluruh dan skala penuh untuk mencapai efisiensi energi melalui penskalaan tegangan secara bersamaan dan pengaturan beban kerja serta suhu.
4. **Desain Bersama Perangkat Keras-Perangkat Lunak yang Didukung AI:** AI semakin populer selama tahap desain awal sirkuit VLSI untuk mengeksplorasi arsitektur perangkat keras yang efisien dan optimasi perangkat lunak. Dengan mengintegrasikan keduanya, kerangka kerja desain bersama meminimalkan waktu pemasaran sekaligus memastikan pengurangan energi optimal di semua tingkatan dalam tumpukan.
5. **Integrasi dengan Subsistem yang Dijalankan oleh Energi Terbarukan**: Karena keberlanjutan secara universal dianggap penting, manajemen daya yang didukung AI akan membentuk inti komputasi hijau, dengan kemungkinan bagi chip untuk secara dinamis mengubah pengaturannya saat berjalan dengan energi terbarukan yang tidak stabil (node IoT bertenaga surya).

## 8.7 MANAJEMEN DAYA CERDAS BERBASIS AI DALAM DESAIN VLSI

Kebutuhan akan desain VLSI berdaya rendah sangat penting saat ini dengan meningkatnya permintaan akan peralatan seluler, aplikasi IoT, sistem otonom, dan pusat data besar. Teknik pengurangan daya tradisional seperti DVFS, power and clock gating, dan MTCMOS telah teruji oleh waktu dan bermanfaat, tetapi kehilangan daya tariknya dengan beban kerja baru, persyaratan faktor bentuk yang lebih kecil, dan mandat keberlanjutan. AI menandai pergeseran dari pengalaman desain pasif dan statis untuk mengelola daya dalam desain VLSI ke pendekatan cerdas, aktif, dan adaptif. Teknik desain AI untuk manajemen daya berfokus pada prediksi beban kerja dengan teknik canggih seperti RL, bekerja dengan beban kerja optimal di setiap tingkat abstraksi, menggunakan pola beban kerja waktu nyata, dan mengelola suhu melalui AI untuk membantu mengoptimalkan efisiensi, lebih dari yang mungkin dilakukan sebelumnya.

Teknik desain AI ini dapat membantu meningkatkan efisiensi daya sekaligus memastikan perangkat beroperasi secara efektif, memiliki masa pakai yang lebih lama, dan

berkelanjutan. Manajemen daya dengan AI telah memengaruhi beberapa teknologi, seperti elektronik otomotif, pusat data, sistem komputasi kelas atas, perangkat yang dapat dikenakan, dan ponsel pintar, dan beberapa konsep baru lainnya seperti desain bersama perangkat keras-perangkat lunak, pembelajaran federasi, dan arsitektur neuromorfik menunjukkan arah masa depan dengan konsumsi daya cerdas yang terintegrasi di semua aspek sistem komputasi. Pada skala yang lebih besar, desain VLSI yang dikombinasikan dengan manajemen daya adaptif AI merupakan kunci untuk sistem digital hemat energi dengan kinerja tinggi dan keberlanjutan. Seiring dengan semakin meluasnya batasan integrasi informasi dan kapasitas komputasi di industri semikonduktor, manajemen daya cerdas akan terus menjadi penentu utama apakah pertumbuhan industri secara teknologi layak dan secara ekologis berkelanjutan.

# BAB 9
# PEMODELAN VLSI EFISIEN MELALUI PREDIKSI MOSFET BERBASIS REGRESI

## 9.1 PEMODELAN TRANSISTOR BERBASIS MACHINE LEARNING

Transistor merupakan komponen sentral dalam elektronika modern, mulai dari sakelar sederhana hingga prosesor canggih, dan pemodelan akurat karakteristik arus-tegangan (I-V) sangat penting untuk desain sirkuit yang andal. Model berbasis fisika tradisional, yang berasal dari teori semikonduktor, memberikan wawasan berharga tetapi seringkali menjadi kompleks secara matematis dan kurang andal seiring dengan pengecilan ukuran perangkat dan peningkatan variasi di dunia nyata. Meningkatnya ketersediaan data terukur, bersamaan dengan kemajuan dalam ML, telah menciptakan peluang baru untuk pemodelan transistor berbasis data. Tidak seperti pendekatan berbasis persamaan, teknik ML dapat secara langsung mempelajari pemetaan nonlinier antara tegangan input dan arus drain dari data empiris, mengurangi ketergantungan pada asumsi fisik yang rinci.

Meskipun minat terhadap pemodelan perangkat berbasis ML semakin meningkat, sebagian besar studi berfokus pada satu algoritma atau menggunakan dataset yang dikurasi atau disimulasikan. Hanya sedikit yang memberikan perbandingan sistematis dari beberapa teknik regresi yang diterapkan pada dataset terukur yang sama di bawah kondisi pra-pemrosesan dan evaluasi yang identik. Kesenjangan ini membatasi panduan bagi para peneliti dan praktisi yang ingin memilih strategi pemodelan yang tepat.

Studi ini mengatasi kesenjangan tersebut dengan membandingkan empat algoritma regresi untuk prediksi arus drain transistor: RF, GBR, XGBoost, dan SVR. Metode-metode ini dipilih karena kemampuannya yang telah terbukti untuk menangkap pola nonlinier dan kekuatan komplementer: model pohon ensemble (RF, GBR, XGBoost) kuat dan dapat diinterpretasikan, sementara SVR memberikan perkiraan fungsi halus ketika dikonfigurasi dengan benar. Tujuan kami adalah untuk mengevaluasi seberapa baik setiap model dapat mempelajari hubungan antara tegangan drain-ke-sumber ($V_{DS}$), tegangan gerbang-ke-sumber ($V_{GS}$), dan arus drain ($I_D$) menggunakan dataset terstruktur dari pengukuran arus searah di seluruh rentang bias yang luas. Semua model dilatih dan diuji di bawah kerangka kerja yang seragam untuk memastikan keadilan dan perbandingan.

Cakupan studi ini terbatas pada karakteristik I–V DC statis. Kami tidak membahas perilaku sinyal kecil atau frekuensi tinggi, maupun ekstraksi model kompak untuk simulasi SPICE. Sebaliknya, fokusnya adalah pada regresi akurat data I–V terukur, yang secara langsung relevan untuk desain sirkuit tahap awal, abstraksi perilaku, dan pemodelan pengganti untuk percepatan simulasi. Karya ini berkontribusi dengan: (1) menyajikan perbandingan praktis dari empat metode regresi pada data transistor terukur; (2) membandingkan akurasi prediktif dan generalisasinya dalam kondisi yang identik; dan (3) membahas pertimbangan antara akurasi,

kompleksitas, dan efisiensi. Studi ini dimaksudkan agar dapat diakses oleh insinyur listrik dan peneliti dengan latar belakang ML.

## 9.2 PERKEMBANGAN PEMODELAN TRANSISTOR BERBASIS PEMBELAJARAN MESIN

Pemodelan perilaku transistor telah lama menjadi tema sentral dalam elektronika, dengan karya awal yang didasarkan pada fisika perangkat dan formulasi analitis. Pendekatan klasik, termasuk persamaan Shockley untuk MOSFET dan model Gummel–Poon untuk BJT, memberikan wawasan penting dan membentuk dasar simulator sirkuit. Namun, model-model ini diturunkan berdasarkan asumsi ideal dan kehilangan akurasi ketika perangkat menunjukkan respons nonlinier, penyimpangan yang bergantung pada proses, atau perilaku yang sensitif terhadap suhu.

Seiring dengan semakin kecilnya ukuran sirkuit terpadu hingga dimensi sub-mikron, keterbatasan model analitis tersebut menjadi semakin jelas. Efek kanal pendek, komponen parasit, dan variasi yang disebabkan oleh fabrikasi sulit untuk ditangkap dalam persamaan bentuk tertutup. Tantangan-tantangan ini telah memotivasi pencarian metode berbasis data yang mampu beradaptasi secara fleksibel terhadap perilaku perangkat yang terukur. Pembelajaran mesin (ML) telah muncul sebagai alternatif yang menjanjikan. Aplikasi awal yang menggunakan jaringan saraf tiruan menunjukkan kelayakan pembelajaran pemetaan nonlinier antara tegangan dan arus, meskipun model tersebut seringkali membutuhkan kumpulan data yang besar dan kurang interpretabilitas, sehingga membatasi penggunaan rutin.

Baru-baru ini, metode ensemble seperti RF, GBR, dan XGBoost telah menunjukkan kekokohan dan kinerja yang kuat pada kumpulan data terstruktur. RF diterapkan pada pemodelan I–V MOSFET dan dilaporkan meningkatkan akurasi dibandingkan persamaan empiris. GBR digunakan untuk estimasi parameter perangkat analog, menunjukkan generalisasi yang lebih baik dalam kondisi bising. XGBoost terbukti secara akurat memprediksi perilaku transistor di berbagai kondisi bias dan beban, mengungguli regressor konvensional dalam hal kecepatan dan akurasi.

SVR juga telah diteliti untuk pemodelan kompak karena kemampuannya untuk menyesuaikan fungsi halus dan kontinu dari kumpulan data yang relatif kecil. Ditemukan efektif bila dikombinasikan dengan metode kernel, meskipun ketergantungannya pada penskalaan fitur dan sensitivitas terhadap hiperparameter dapat membatasi kemudahan penerapannya. Meskipun masing-masing metode ini memiliki kekuatan yang berbeda, sebagian besar studi sebelumnya mengevaluasinya secara terpisah atau dalam kondisi yang berbeda, sehingga perbandingan langsung menjadi sulit.

Selain itu, sebagian besar literatur bergantung pada dataset simulasi atau yang telah dikurasi dengan cermat, yang mungkin tidak mencerminkan kompleksitas pengukuran di dunia nyata. Studi ini mengatasi kesenjangan tersebut dengan melakukan perbandingan sistematis antara RF, GBR, XGBoost, dan SVR pada dataset DC terukur yang sama, di bawah protokol pra-pemrosesan dan evaluasi yang konsisten. Analisis ini mempertimbangkan tidak hanya akurasi prediksi dan generalisasi, tetapi juga aspek komputasi seperti waktu pelatihan, ukuran model, dan efisiensi, memberikan panduan praktis untuk alur kerja pemodelan transistor.

## 9.3 DESKRIPSI DATASET

Dataset yang digunakan dalam studi ini terdiri dari pengukuran arus searah (DC) yang diperoleh dari MOSFET dalam berbagai kondisi bias listrik. Setiap sampel mencatat sepasang tegangan input (VDS dan VGS) bersama dengan ID yang dihasilkan. Untuk setiap nilai VGS yang tetap, tegangan drain-ke-sumber diubah-ubah dalam rentang yang ditentukan, dan arus drain yang sesuai diukur. Proses ini menghasilkan grid titik operasi terstruktur yang secara komprehensif menangkap respons listrik perangkat. Variabel input (VDS, VGS) mencakup rentang kontinu yang meliputi wilayah operasi linier dan saturasi. Arus keluaran, ID, bervariasi secara nonlinier terhadap input ini, sehingga dataset ini sangat cocok untuk menguji model regresi yang dirancang untuk menangkap ketergantungan yang kompleks. Karena data dikumpulkan khusus untuk tujuan pemodelan, data tersebut konsisten dan tidak memerlukan pembersihan atau pra-pemrosesan tambahan. Dataset terstruktur ini menyediakan dasar untuk semua eksperimen pemodelan selanjutnya, memastikan perbandingan yang adil dan sistematis dari teknik regresi yang dievaluasi dalam pekerjaan ini.

**Persiapan Data**

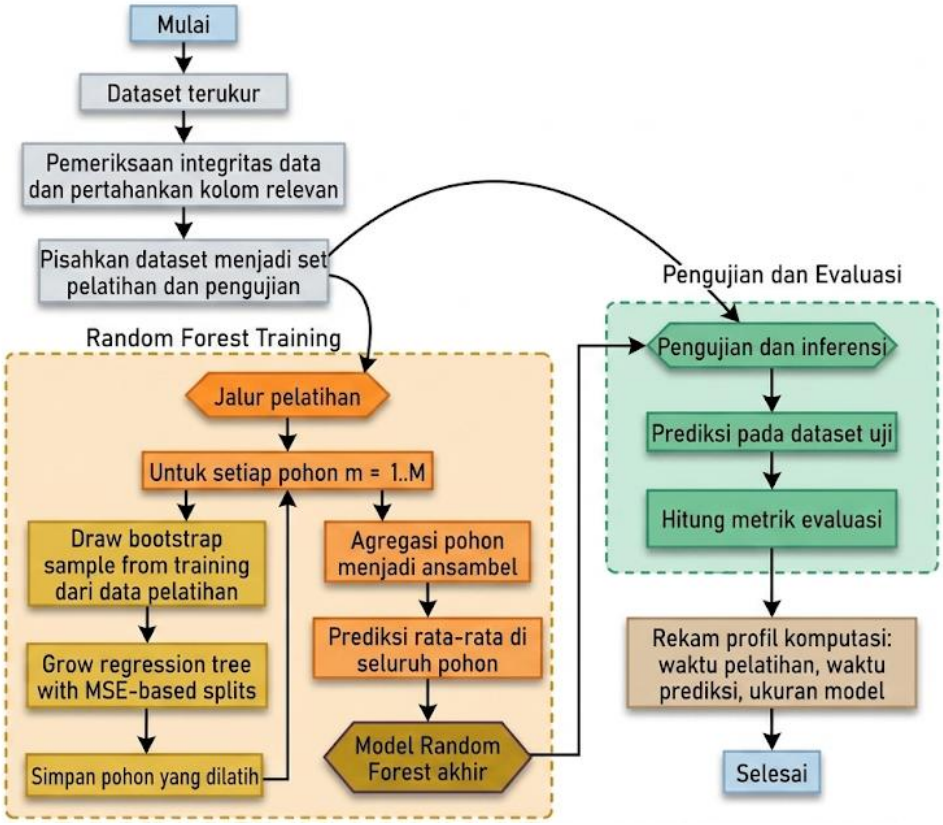
Dataset terdiri dari nilai $I_D$ terukur yang sesuai dengan kombinasi $V_{DS}$ dan $V_{GS}$. Setiap titik data sesuai dengan kondisi operasi yang berbeda, dan tujuan dari tugas pemodelan adalah untuk mempelajari hubungan nonlinier yang memetakan tegangan input ini ke arus yang dihasilkan. Sebelum pemodelan, dataset diperiksa konsistensinya, dan hanya kolom-kolom penting, $V_{DS}$, $V_{GS}$, dan $I_D$, yang dipertahankan. Data kemudian dibagi secara acak menjadi subset pelatihan dan pengujian menggunakan pembagian 80:20, memastikan bahwa model divalidasi pada sampel yang belum pernah dilihat sebelumnya. Karena tugas ini pada dasarnya adalah masalah regresi, variabel output ($I_D$) diperlakukan sebagai variabel kontinu. Untuk regressor berbasis pohon, RF, Gradient Boosting (GB), dan XGBoost, tidak diperlukan penskalaan fitur, karena model-model ini bergantung pada pembagian berbasis ambang batas dan tidak sensitif terhadap besarnya fitur input. Sebaliknya, SVR bergantung pada jarak geometris dalam ruang fitur, sehingga normalisasi menjadi sangat penting. Oleh karena itu, baik $V_{DS}$ maupun VGS distandarisasi menggunakan StandardScaler untuk mencapai rata-rata nol dan varians unit sebelum pelatihan SVR. Langkah ini memastikan konvergensi yang stabil dan kinerja kernel yang efektif.

**Regresi Random Forest untuk Pemodelan Arus Drain Transistor**

Seperti yang diilustrasikan pada Gambar 9.1, diagram alir dari pipeline regresi Random Forest (RF) yang digunakan dalam pembahasan ini mencakup persiapan dataset, pengambilan sampel bootstrap, konstruksi pohon, dan perataan ensemble. Untuk menangkap ketergantungan nonlinier arus drain $I_D$ pada tegangan terminal $V_{DS}$ dan $V_{GS}$, pembahasan ini menggunakan algoritma regresi RF [17]. Tujuannya adalah untuk mendekati pemetaan $f : \mathbb{R}^2 \longrightarrow \mathbb{R}$ sedemikian sehingga $f\ (V_{DS}, V_{GS}) \approx I_D$.

RF membangun sebuah ansambel pohon regresi $\{T_1, T_2, \dots, T_M\}$, yang masing-masing dilatih pada sebuah subset *bootstrap* dari himpunan data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$, di mana $x_i = \left[V_{DS}^{(i)}, V_{GS}^{(i)}\right]$, dan $y_i = I_D^{(i)}$. Di dalam setiap pohon, ruang input dipartisi secara rekursif menggunakan pemisahan biner. Pada setiap simpul, sebuah subset acak dari fitur $F_m \subseteq$

$\{V_{DS}, V_{GS}\}$ dipertimbangkan, dan pemisahan yang meminimalkan kesalahan kuadrat rata-rata (MSE) dipilih (18) seperti yang ditunjukkan pada Gambar. (9.1).




**Gambar 9.1** Diagram alir dari alur regresi Random Forest untuk pemodelan arus drain transistor.

$$MSE_{split} = \frac{1}{|\mathcal{D}_{left}|} \sum_{(x_i, y_i) \in \mathcal{D}_{left}} \left(y_j - \bar{y}_{left}\right)^2 + \frac{1}{|\mathcal{D}_{right}|} \sum_{(x_i, y_i) \in \mathcal{D}_{right}} \left(y_j - \bar{y}_{right}\right)^2 \tag{9.1}$$

di mana $\bar{y}_{left}$ dan $\bar{y}_{right}$ menyatakan rata-rata arus *drain* pada simpul anak. Pemisahan dilanjutkan hingga kondisi berhenti tercapai (misalnya, kedalaman maksimum atau ukuran sampel minimum).

Setiap pohon menghasilkan prediksi lokal $T_m(x)$. Output RF akhir adalah rata-rata dari seluruh pohon yang diberikan pada Gambar. (9.2).

$$\hat{y} = RF(x) = \frac{1}{M} \sum_{m=1}^{M} T_m(x) \tag{9.2}$$

**Langkah-langkah Algoritma (Kode Semu): RF**

- *Input:* himpunan data $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$, jumlah pohon $M$.
- *Output:* Fungsi regresi $RF(x)$.

1. Untuk setiap pohon $m = 1, \dots, M$:
   a. Ambil sampel *bootstrap* $\mathcal{D}_m$.
   b. Bangun pohon regresi $T_m$:
      - Pilih subset fitur secara acak $\mathcal{F}_m$.
      - Pilih pemisahan yang meminimalkan MSE.
      - Lanjutkan hingga kondisi berhenti.
   c. Simpan pohon yang telah dilatih $T_m$.
2. Untuk input baru $x$: hitung $\hat{y} = \frac{1}{M}\sum_{m=1}^{M} T_m\,(x)$

**Pertimbangan Kompleksitas: RF**

*Pelatihan:* $O(M \cdot n \cdot \log n)$, dengan $M$ pohon dan $n$ sampel. Sangat dapat diparalelkan karena setiap pohon bersifat independen.
*Inferensi:* $O(M \cdot d)$, di mana $d = 2$ (sangat kecil), memungkinkan prediksi yang efisien.

**Detail Implementasi: RF**

Studi ini menggunakan $M = 100$ pohon. Pemilihan fitur pada setiap simpul berlangsung secara otomatis ($d' = d$), dan pemisahan MSE dari scikit-learn diterapkan. Karena RF tidak sensitif terhadap transformasi monotonik, penskalaan fitur tidak diperlukan. Pelatihan/pengujian mengikuti pembagian 80:20 yang dijelaskan pada Bagian (Persiapan Data) untuk memastikan keterbandingan di seluruh model.

**Regresi Vektor Pendukung (SVR) untuk Pemodelan Arus Drain Transistor**

Seperti yang diilustrasikan pada Gambar 9.2, diagram alir dari pipeline Regresi Vektor Pendukung, menunjukkan standardisasi fitur, pemetaan kernel, optimasi, dan prediksi. Regresi Vektor Pendukung (SVR) digunakan untuk menangkap ketergantungan nonlinier arus drain $I_D$ pada tegangan terminal $V_{DS}$ dan $V_{GS}$. SVR merumuskan regresi sebagai masalah optimasi cembung di mana tujuannya adalah untuk mendekati fungsi yang menyimpang dari arus terukur paling banyak dengan toleransi ε, sambil mempertahankan kompleksitas model minimal. Fungsi regresi diberikan pada Persamaan (9.3)

$$f(x) = \langle w, \varphi(x) \rangle + b \tag{9.3}$$

di mana w adalah vektor bobot, b adalah bias, dan ϕ(x) memetakan input ke ruang fitur berdimensi lebih tinggi. Masalah optimasi ditunjukkan pada Persamaan (9.4)

$$\min_{w,b,\xi_i,\xi_i^*} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \tag{9.4}$$

tunduk pada:

$$\begin{cases} y_i - \langle w, \varphi(x)\rangle - b \le \varepsilon + \xi_i, \\ \langle w, \varphi(x)\rangle + b - y_i \le \varepsilon + \xi_i^*, \\ \xi_i, \xi_i^* \ge 0. \end{cases}$$

di mana $\xi_i, \xi_i^*$ adalah variabel kendur, dan C > 0 adalah parameter regularisasi yang mengontrol keseimbangan antara kerataan dan toleransi terhadap penyimpangan.

Untuk menangkap perilaku transistor nonlinier, kernel Radial Basis Function (RBF) diadopsi seperti yang disajikan pada Persamaan (9.6).

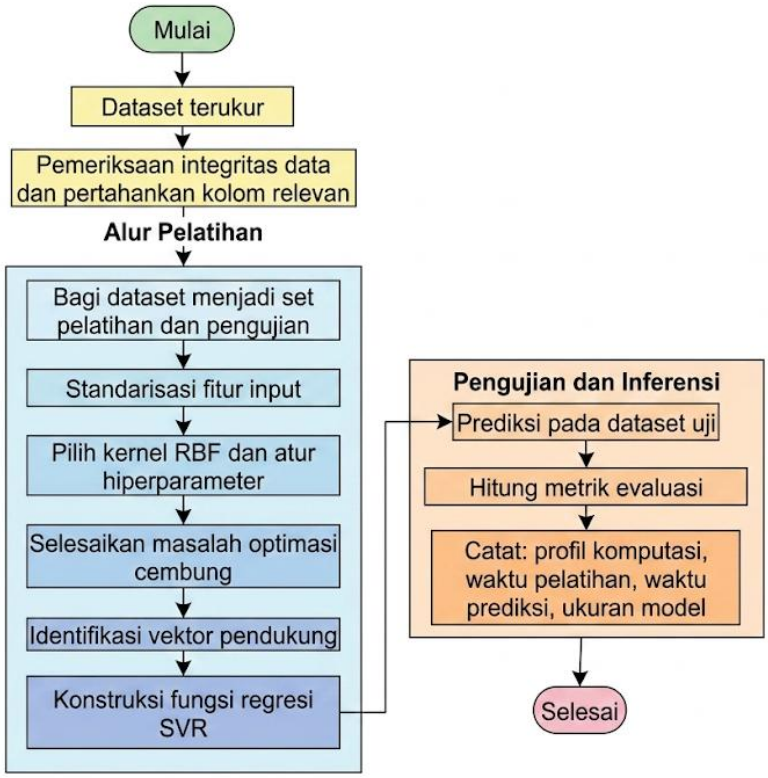
$$K(x_i, x_j) = \exp\left(-\gamma\|x_i - x_j\|^2\right) \tag{9.6}$$

Fungsi regresi akhir ditunjukkan pada Persamaan (9.7).

$$f(x) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\,K(x_i, x) + b \tag{9.7}$$

di mana hanya vektor pendukung (titik dengan pengali bukan nol αi, α∗) yang berkontribusi pada prediksi.

Karena SVR bergantung pada jarak geometris dalam ruang fitur, fitur input distandarisasi menjadi rata-rata nol dan varians unit sebelum pelatihan. Model diimplementasikan menggunakan scikit-learn dengan parameter C = 100, ε = 0,01, dan Y = *skala*.




**Gambar 9.2** Diagram alir dari alur kerja Support Vector Regression untuk pemodelan arus drain transistor.

**Langkah-langkah Algoritma (Kode Pseudo): SVR**

- *Input*: Dataset pelatihan $D = \{(x_i, y_i)\}_{i=1}^{n}$
- *Output*: Fungsi regresi $f(x)$

1. Standarisasi fitur input menjadi rata-rata nol dan varians satuan.
2. Tentukan fungsi kernel: RBF dengan $\gamma = scale$.
3. Tetapkan *hyperparameter*: $C = 100, \varepsilon = 0.01$.
4. Selesaikan masalah optimasi cembung untuk mendapatkan $(\alpha_i, \alpha_i^*, b)$.
5. Identifikasi vektor pendukung (titik data dengan pengali bukan nol).
6. Bangun fungsi regresi $f(x)$.
7. Gunakan $f(x)$ untuk memprediksi arus *drain* untuk input yang belum pernah dilihat.

**Pertimbangan Kompleksitas: SVR**

- *Pelatihan*: $O(n^2 \cdot d)$ hingga $O(n^3)$, tergantung pada ukuran matriks kernel ($n =$ jumlah sampel, $d =$ fitur).
- *Inferensi*: $O(s \cdot d)$, di mana $s =$ jumlah vektor pendukung (biasanya lebih kecil dari $n$).
- *Memori*: Berskala dengan jumlah vektor pendukung alih-alih total sampel.

**Detail Implementasi: SVR**

Studi ini mengimplementasikan SVR menggunakan scikit-learn dengan kernel RBF. Parameter ditetapkan sebagai $C = 100$, $\varepsilon = 0.01$, dan $\gamma = scale$. Karena SVR sensitif terhadap besaran fitur, baik $V_{DS}$ maupun $V_{GS}$ distandarisasi menjadi rata-rata nol dan varians

satuan menggunakan *StandardScaler*. Pelatihan dan pengujian mengikuti pembagian 80:20 yang dijelaskan di Bagian (Persiapan Data) untuk memastikan konsistensi di semua model.

**Regresi XGBoost untuk pemodelan arus drain transistor**

Seperti yang diilustrasikan pada Gambar 9.3, diagram alir dari alur regresi XGBoost, yang menguraikan putaran peningkatan berurutan, komputasi gradien/Hessian, penyesuaian pohon, dan pembaruan aditif. XGBoost digunakan untuk memodelkan ketergantungan nonlinier ID pada tegangan terminal $V_{DS}$ dan $V_{GS}$. XGBoost memperluas ide GB dengan menggabungkan optimasi orde kedua, regularisasi, dan pencarian pemisahan yang efisien, yang membuatnya kuat secara komputasi dan tangguh terhadap overfitting.

Fungsi prediktif, yang dinyatakan sebagai ensemble aditif dari M pohon regresi, ditunjukkan pada Persamaan (9.8)

$$\hat{y}_i = f(x_i) = \sum_{m=1}^{M} f_m\,(x_i), f_m \in \mathcal{F} \tag{9.8}$$

di mana $F$ menunjukkan ruang pohon regresi. Tidak seperti RF, yang menumbuhkan pohon secara independen, XGBoost membangunnya secara berurutan, dengan setiap pohon baru mengoreksi kesalahan residual dari ensemble sebelumnya.

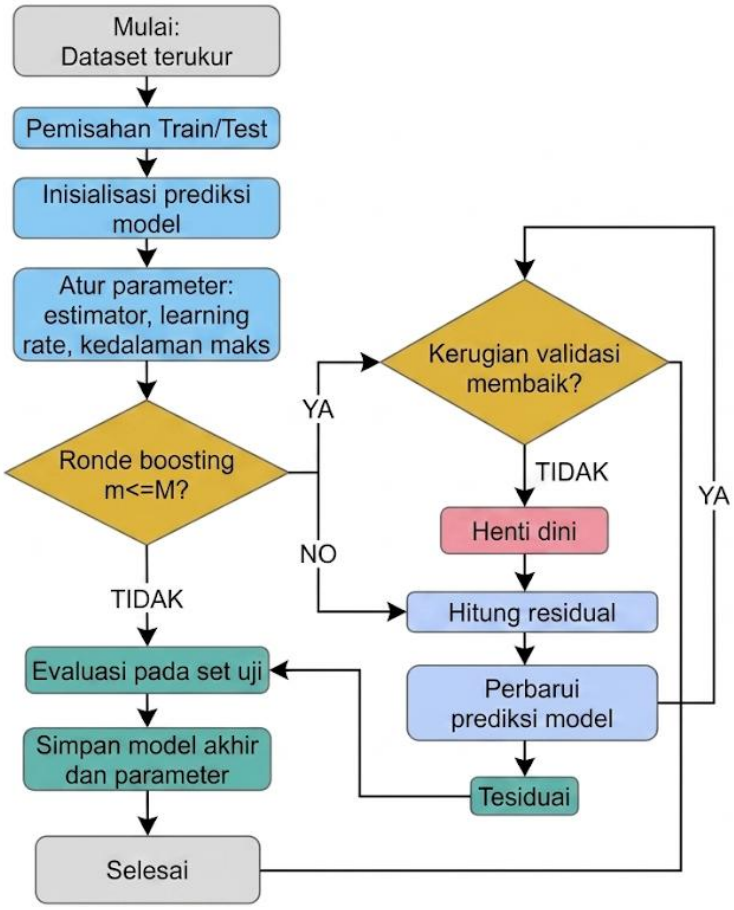
Tujuan optimasi menggabungkan fungsi kerugian cembung yang dapat didiferensiasi (kesalahan kuadrat dalam pembahasan ini) dengan suku regularisasi yang diberikan dalam Persamaan (9.9).

$$\mathcal{L} = \sum_{i=1}^{n} l\,(y_i, \hat{y}_i) + \sum_{m=1}^{M} \Omega\,(f_m) \tag{9.9}$$

di mana l$(y_i, y^i)$ adalah kerugian kesalahan kuadrat, dan $\Omega(f)$ memberikan penalti pada kompleksitas model berdasarkan jumlah daun dan bobotnya. Parameter regularisasi $(\gamma, \lambda)$ meningkatkan generalisasi dan mengurangi overfitting. Pada setiap putaran boosting, gradien dan Hessian dari fungsi kerugian dihitung untuk semua sampel pelatihan. Pembagian ditentukan dengan memaksimalkan gain, memastikan bahwa setiap pohon mengurangi kesalahan secara efektif. Setelah struktur ditetapkan, bobot daun optimal dihitung, dan prediksi diperbarui secara aditif.

**Regresi penguatan gradien untuk pemodelan arus drain transistor**

Seperti yang diilustrasikan pada Gambar 9.4, diagram alir dari pipeline regresi GB, menggambarkan komputasi residual, pelatihan pembelajar lemah, pembaruan iteratif, dan prediksi ensemble akhir. GBR digunakan untuk memodelkan hubungan nonlinier antara ID dan tegangan terminal, $V_{DS}$ dan $V_{GS}$. GB adalah metode ensemble yang secara berurutan membangun pohon regresi, dengan setiap pohon baru mengoreksi kesalahan residual dari ensemble yang telah dibangun sejauh ini, sehingga secara progresif meningkatkan akurasi prediksi.




**Gambar 9.4** Diagram alir pipeline regresi Gradient Boosting untuk drain transistor.

Misalkan dataset dinotasikan sebagai $= \{(x_i, y_i)\}_{i=1}^{n}$, di mana $x_i = \left[V_{DS}^{(i)}, V_{GS}^{(i)}\right]$ dan $y_i = I_D^{(i)}$. Fungsi prediktif dinyatakan seperti yang diberikan pada Persamaan (10).

$$\hat{y}_i = f(x_i) = \sum_{m=1}^{M} f_m(x_i), f_m \in \mathcal{F} \tag{9.10}$$

di mana 7 menunjukkan ruang pohon regresi. Pelatihan meminimalkan fungsi kerugian kuadrat kesalahan yang dikombinasikan dengan suku regularisasi ditunjukkan pada Persamaan (9.11).

$$\mathcal{L} = \sum_{i=1}^{n} \ell(y_i, \hat{y}_i) + \sum_{m=1}^{n} \Omega(f_m) \tag{9.11}$$

di mana $\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$ memberikan penalti terhadap kompleksitas model. Di sini, $T$ adalah jumlah daun, $w_j$ adalah bobot daun, dan $\gamma, \lambda$ adalah parameter regularisasi.

Pada putaran *boosting* $m$, pseudo-residual dihitung seperti yang ditunjukkan pada Pers. (9.12).

$$g_i^{(m)} = y_i - \hat{y}_i^{(m-1)} \tag{9.12}$$

Pohon regresi $f_m(x)$ kemudian dicocokkan pada dataset $\left\{\left(x_i, g_i^{(m)}\right)\right\}_{i=1}^{n}$. Nilai-nilai daun diperbarui dengan meminimalkan *loss* yang diregularisasi, dan prediksi ansambel diperbarui seperti yang diberikan pada Pers. (9.13).

$$\hat{y}_i^{(m)} = \hat{y}_i^{(m-1)} + \eta f_m(x_i) \tag{9.13}$$

di mana η adalah laju pembelajaran. Setelah M iterasi, model akhir adalah jumlah dari semua pohon.

**Langkah-langkah Algoritma (Kode Pseudo): GBR**

- *Input*: Dataset pelatihan $= \{(x_i, y_i)\}_{i=1}^{n}$, jumlah putaran *boosting* $M$.
- *Output*: Fungsi regresi $f(x)$.

1. Inisialisasi prediksi: $\hat{y}_i^{(0)} = 0$
2. Untuk setiap putaran *boosting* $m = 1, \ldots, M$:
   - Hitung pseudo-residual $g_i^{(m)} = y_i - \hat{y}_i^{(m-1)}$
   - Cocokkan pohon regresi $f_m(x)$ pada residual.
   - Hitung bobot daun optimal.
   - Perbarui prediksi: $\hat{y}_i^{(m)} = \hat{y}_i^{(m-1)} + \eta f_m(x_i)$
3. Output model akhir: $f(x) = \sum_{m=1}^{M} f_m(x)$

**Pertimbangan Kompleksitas: GBR**

- *Pelatihan (Training)*: $\mathcal{O}(M \cdot d \cdot \log n)$, di mana $M$ adalah jumlah putaran *boosting* dan $n$ adalah jumlah sampel pelatihan. Kompleksitas muncul dari pencocokan pohon secara berurutan (sekuensial).
- *Inferensi (Inference)*: $\mathcal{O}(M \cdot d)$, dengan $d$ adalah kedalaman pohon (kecil, di sini $\leq 4$), membuat prediksi menjadi efisien.
- *Memori (Memory)*: Lebih ringkas dibandingkan dengan RF karena pohon-pohonnya dangkal dan ditambahkan secara berurutan.

**Detail Implementasi: GBR**

Dalam pembahasan ini, GB diimplementasikan menggunakan *GradientBoostingRegressor* dari scikit-learn. Konfigurasinya mencakup 200 putaran *boosting*, tingkat pembelajaran (*learning rate*) $\eta = 0.05$, dan kedalaman pohon maksimum 4. Karena metode ini berbasis pohon, tidak ada penskalaan fitur input yang diperlukan. Pelatihan/pengujian mengikuti pembagian 80:20 yang dijelaskan pada Bagian (Persiapan Data) untuk menjaga konsistensi di antara semua model.

**Hasil dan Diskusi**

Tujuan utama pembahasan ini adalah untuk mengembangkan model yang akurat dan andal untuk memprediksi arus drain (ID) MOSFET dari tegangan terminalnya ($V_{DS}$ dan $V_{GS}$).

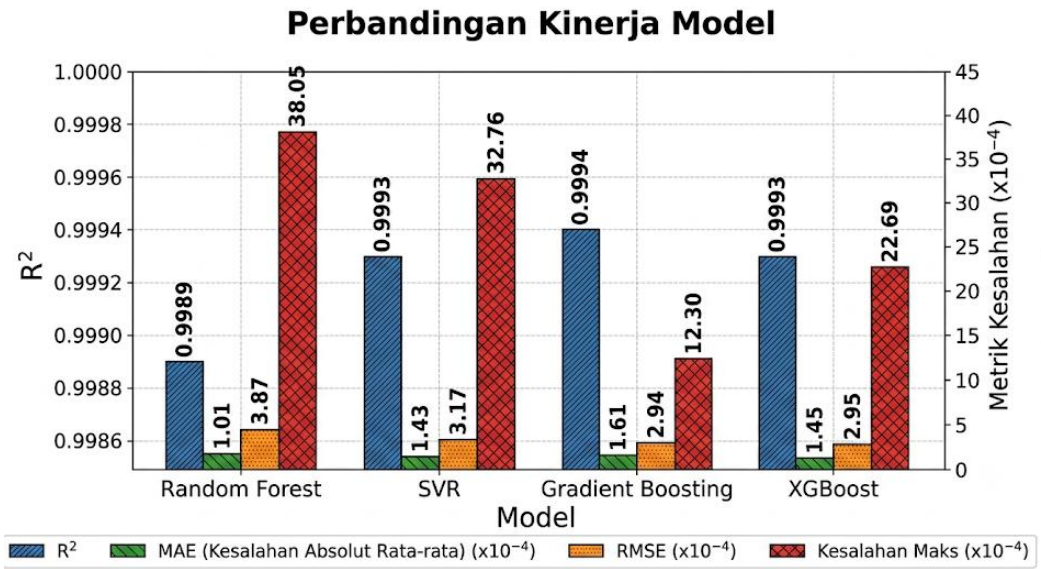
Berdasarkan pendekatan regresi yang dijelaskan sebelumnya, kami berupaya untuk menangkap pemetaan nonlinier antara tegangan input ini dan arus drain yang dihasilkan. Pipeline persiapan dan evaluasi data yang seragam memastikan perbandingan yang adil antar model.

Tabel 9.1 dan Gambar 9.5 merangkum kinerja prediksi dalam hal MAE, RMSE, R2, dan Kesalahan Maksimum. Keempat regressor mencapai akurasi yang sangat baik, dengan nilai R2 di atas 0,998, yang menegaskan bahwa lebih dari 99,8% varians dalam ID terukur dijelaskan. Kesalahan berada pada orde $10^{-4}$ A, masih dalam kisaran yang dianggap dapat diabaikan untuk desain dan simulasi MOSFET. GB mencapai RMSE terendah (2,94 × $10^{-4}$ A), menunjukkan presisi tinggi dalam mereproduksi kurva I–V terukur, sementara RF mencapai MAE terendah (1,01 × $10^{-4}$ A), mencerminkan konsistensi yang kuat. XGBoost menawarkan keseimbangan antara akurasi dan biaya komputasi, sedangkan SVR mempertahankan akurasi yang sangat baik sambil tetap menjadi yang paling ringkas dan tercepat untuk dijalankan.

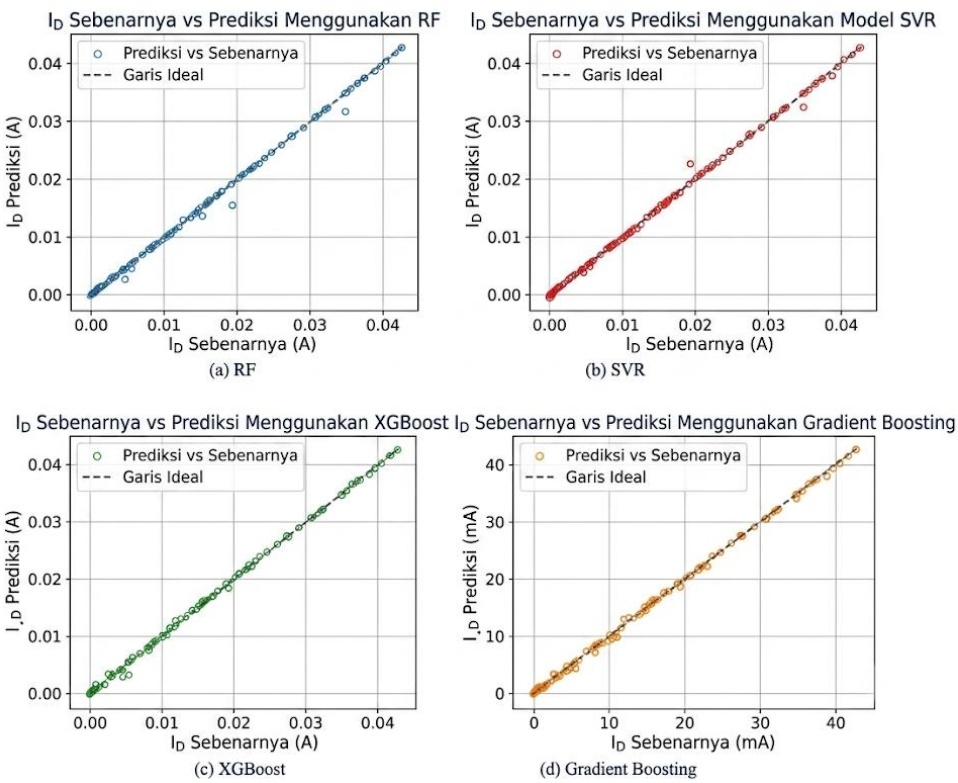
Gambar 9.6 menunjukkan ID aktual versus prediksi di semua model. Titik prediksi sejajar dengan garis ideal putus-putus, dengan GB dan XGBoost menunjukkan kecocokan paling ketat, konsisten dengan skor $R^2$ mereka yang sedikit lebih tinggi.

**Tabel 9.1** Kinerja prediksi dari empat model regresi untuk arus drain MOSFET

| Model | $MAE$ ($\times 10^{-4}$) | $RMSE$ ($\times 10^{-4}$) | $R^2$ | $Kesalahan\ Maks$ ($\times 10^{-4}$) |
|---|---|---|---|---|
| Random Forest | 1.01 | 3.87 | 0.9989 | 38.05 |
| SVR | 1.43 | 3.17 | 0.9993 | 32.76 |
| Gradient Boosting | 1.61 | 2.94 | 0.9994 | 12.30 |
| XGBoost | 1.45 | 2.95 | 0.9993 | 22.69 |

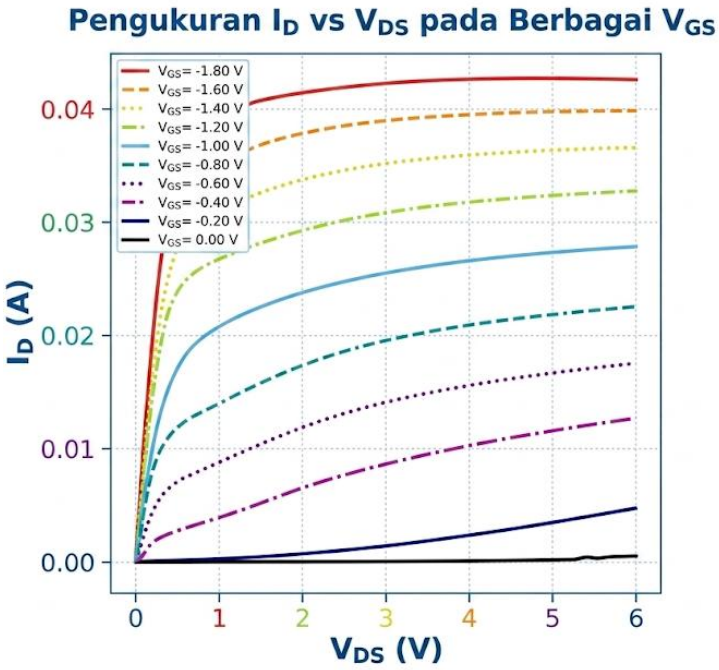



**Gambar 9.5** Kinerja prediksi dari empat model regresi untuk arus drain MOSFET.

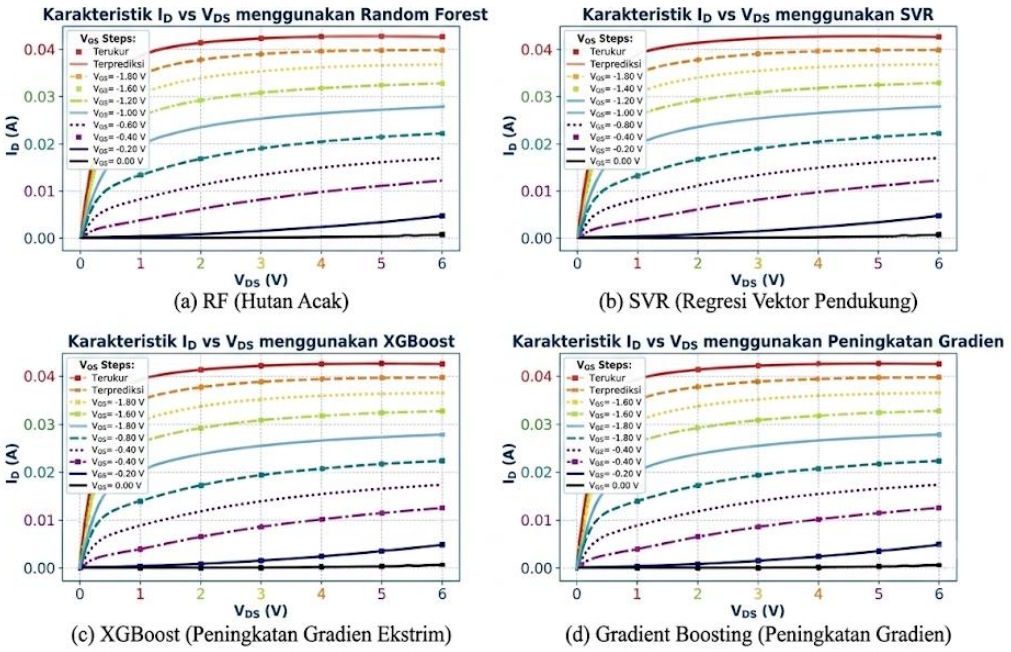



**Gambar 9.6** Arus drain aktual vs. prediksi ($I_D$) untuk keempat model regresi. Semua model selaras dengan garis ideal, dengan GB dan XGBoost menunjukkan kesesuaian terdekat. (a) RF, (b) SVR, (c) XGBoost, (d) Gradient Boosting.

Gambar 9.7 memplot kurva $I_D - V_{DS}$ terukur pada tingkat $V_{GS}$ tetap, sedangkan Gambar 9.8 membandingkan nilai terukur dan prediksi. Keempat model tersebut menangkap tren nonlinier secara efektif, dengan GB dan XGBoost menunjukkan kesesuaian yang lebih baik pada kondisi bias ekstrem. Analisis residual pada Gambar 9.9 menyoroti distribusi kesalahan prediksi. Di keempat model, residual berpusat di sekitar nol dengan varians kecil, sementara GB dan XGBoost menunjukkan sebaran yang paling sempit. Gambar 9.10 menyajikan permukaan 3D sebenarnya dari arus drain terukur sebagai fungsi $V_{DS}$ dan $V_{GS}$. Gambar 9.11 menunjukkan permukaan 3D yang diprediksi, di mana semua model mereplikasi bentuk keseluruhan secara akurat, meskipun GB dan XGBoost menghasilkan perkiraan yang lebih halus.



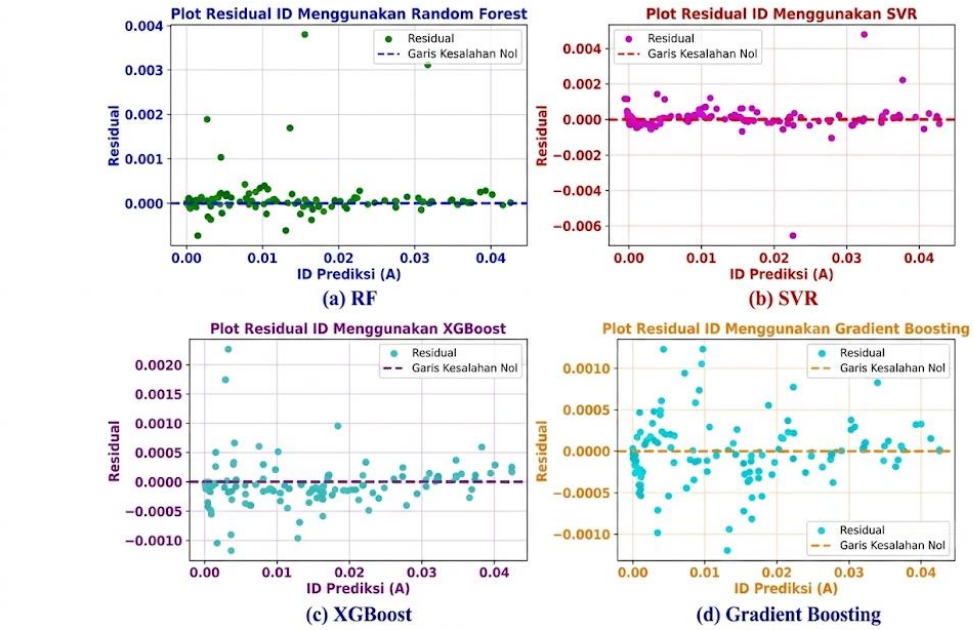

**Gambar 9.7** Kurva $I_D$ vs. $V_{DS}$ terukur pada tingkat $V_{GS}$ tetap.



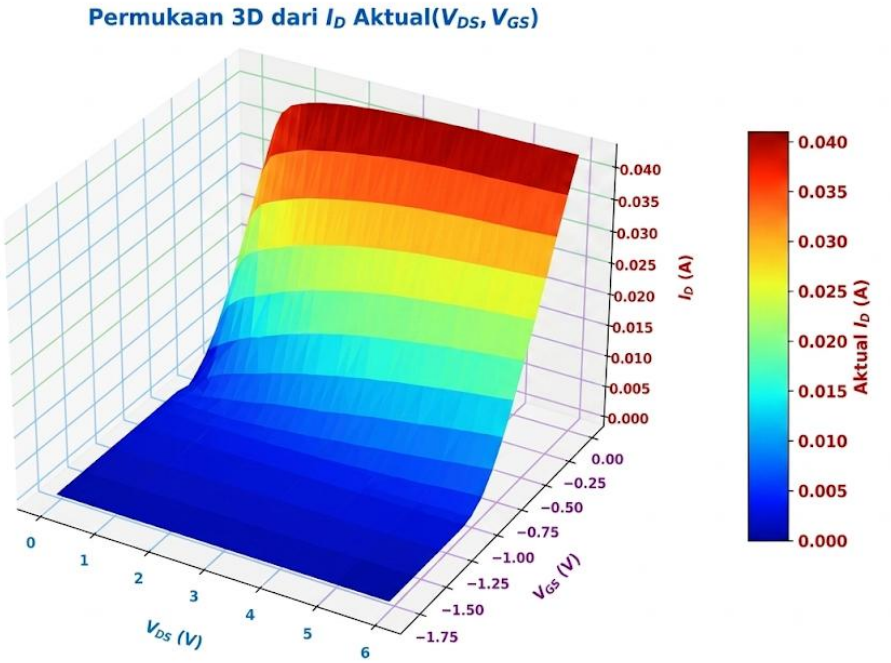

(a) RF (Hutan Acak) (b) SVR (Regresi Vektor Pendukung)

(c) XGBoost (Peningkatan Gradien Ekstrim) (d) Gradient Boosting (Peningkatan Gradien)

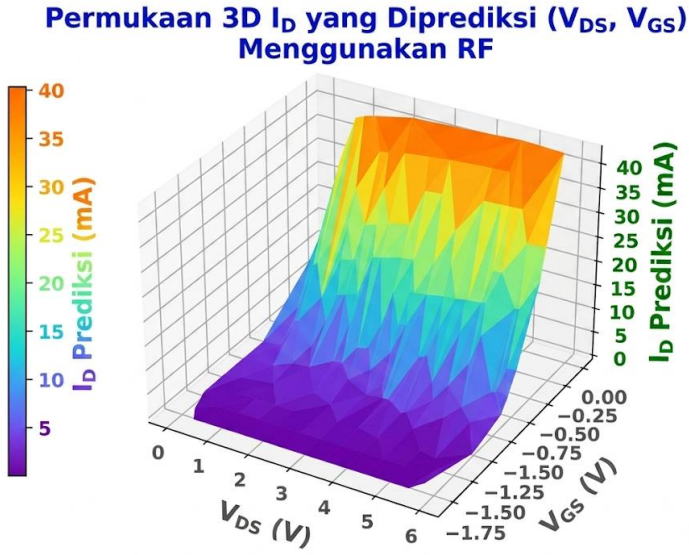
**Gambar 9.8** Perbandingan kurva $I_D - V_{DS}$ terukur dan prediksi pada tingkat $V_{GS}$ tetap. (a) RF, (b) SVR, (c) XGBoost, (d) Gradient Boosting.



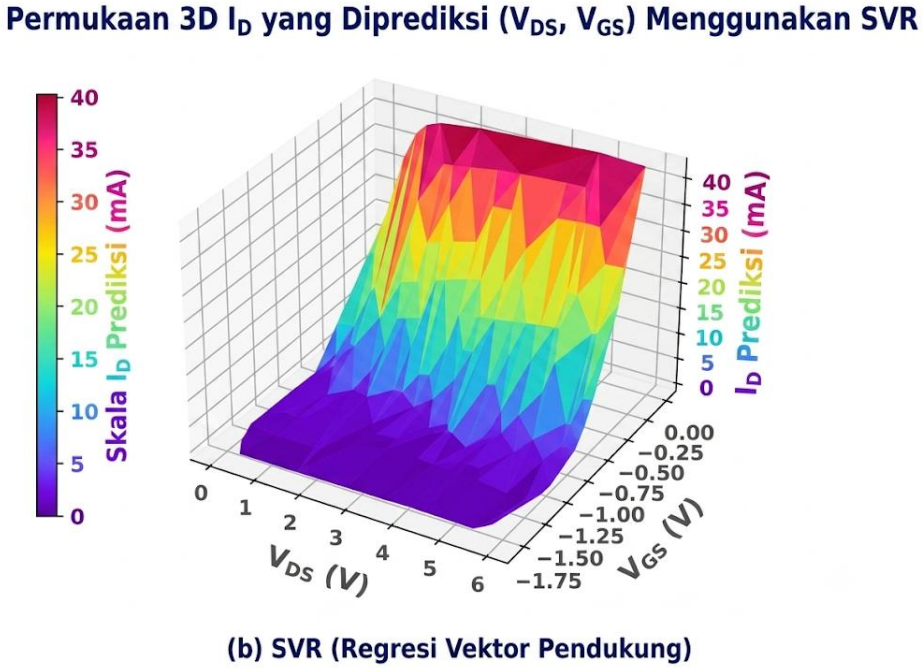

**Gambar 9.9** Plot residual arus drain yang diprediksi untuk semua model. (a) RF, (b) SVR, (c) XGBoost, (d) Gradient Boosting.




**Gambar 9.10** Permukaan 3D sebenarnya dari $I_D$ terukur sebagai fungsi $V_{DS}$ dan $V_{GS}$.



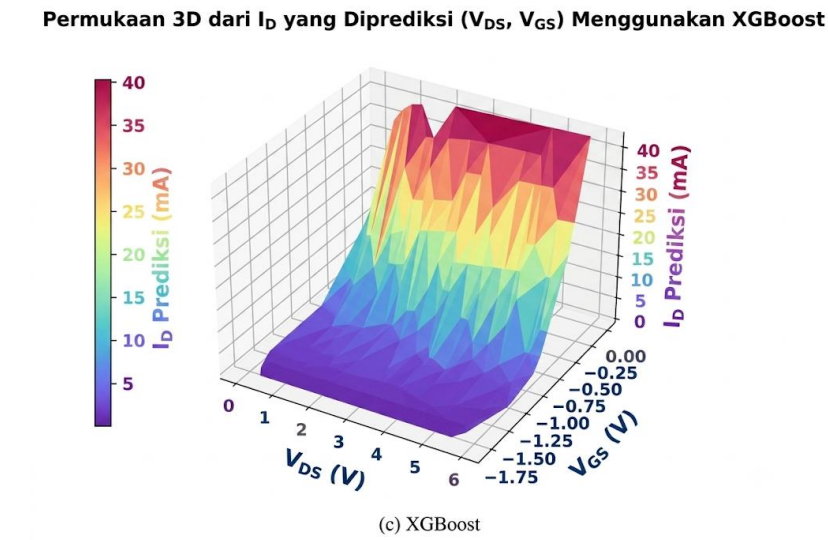

(a) RF (Hutan Acak)



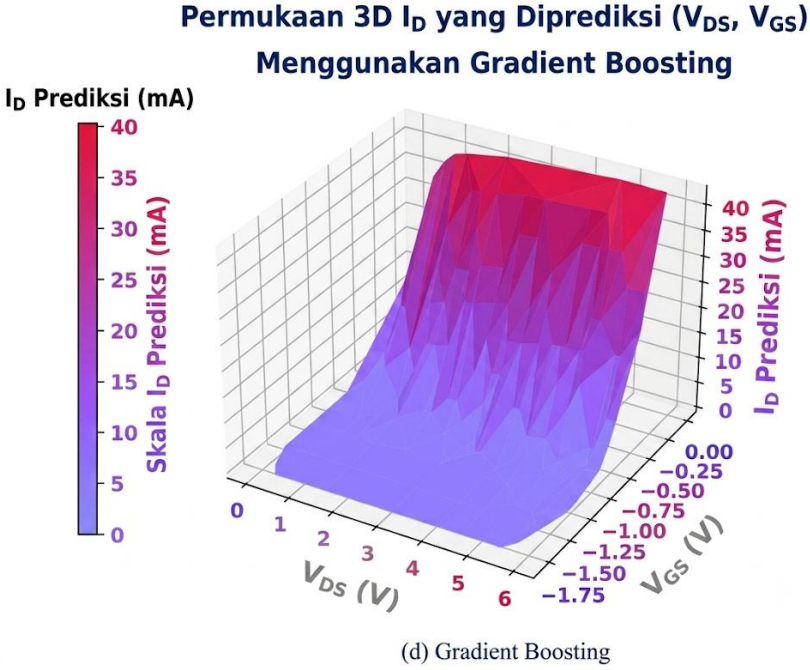

**(b) SVR (Regresi Vektor Pendukung)**

**Gambar 9.11** Permukaan 3D $I_D$ yang diprediksi dari empat model regresi. (a) RF, (b) SVR, (Bersambung)

Terakhir, Gambar 9.12 menyajikan peta panas kesalahan prediksi absolut. Kesalahan minimal di seluruh ruang bias, dengan GB dan XGBoost mempertahankan distribusi kesalahan yang paling seragam.



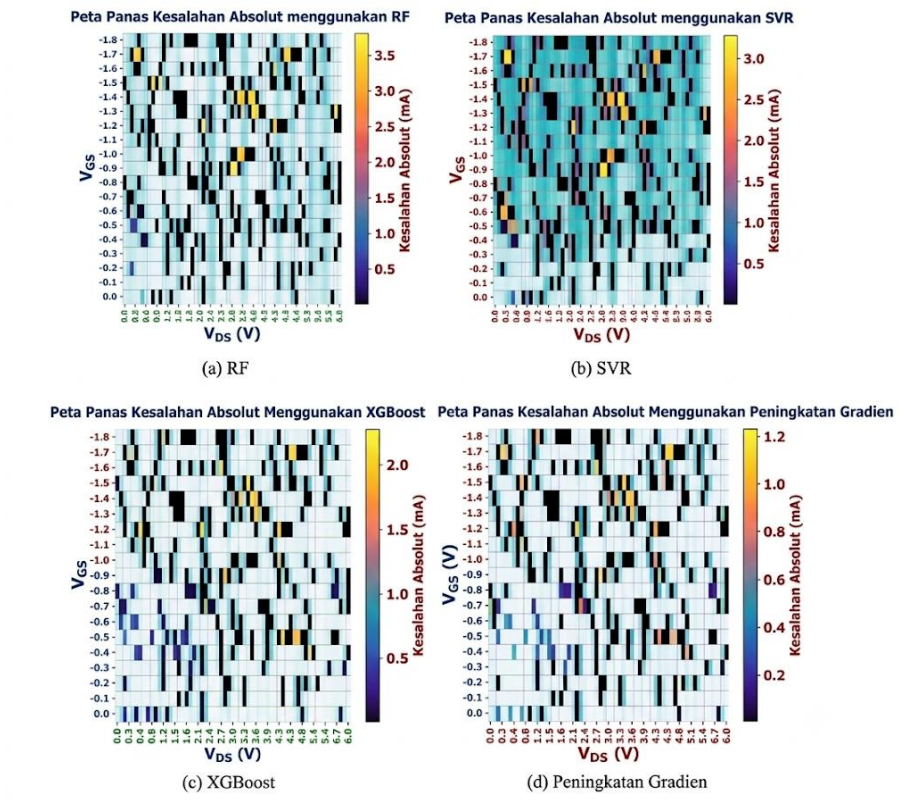

(c) XGBoost




(d) Gradient Boosting

**Gambar 9.11** (Lanjutan) (c) XGBoost, (d) Gradient Boosting.

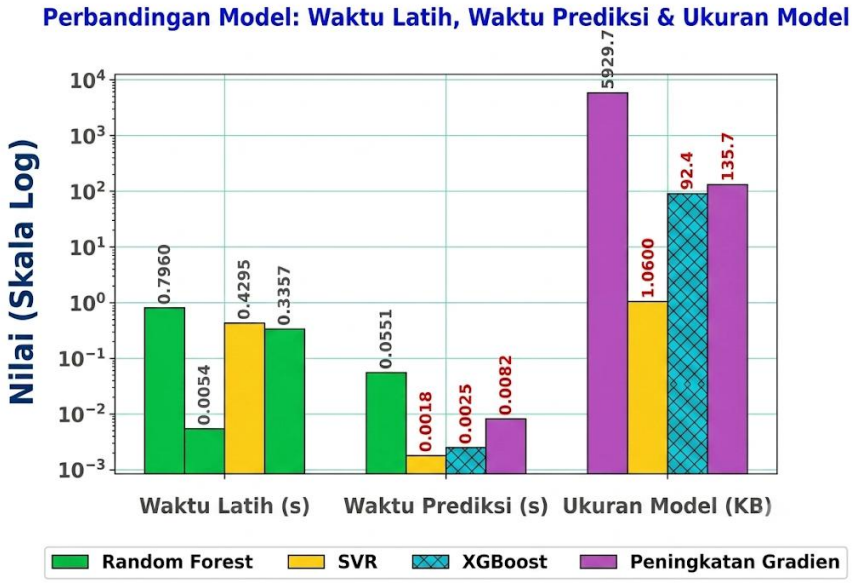
Tabel 9.2 dan Gambar 9.13 merangkum aspek komputasi: waktu pelatihan, waktu prediksi, dan ukuran model. SVR adalah yang tercepat untuk dilatih dan diprediksi serta memiliki jejak terkecil (~1 kB). GB dan XGBoost juga efisien, membutuhkan waktu pelatihan yang moderat dan menghasilkan model yang relatif kompak. RF, sebaliknya, lebih lambat, membutuhkan lebih banyak memori (>5 MB), dan menghasilkan ukuran model yang lebih besar.




**Gambar 9.12** Peta panas kesalahan prediksi absolut di seluruh ruang bias. (a) RF, (b) SVR, (c) XGBoost, (d) Gradient Boosting.

**Tabel 9.2** Efisiensi komputasi dari keempat model regresi

| ***Model*** | ***Waktu pelatihan (s)*** | ***Waktu prediksi (s)*** | ***Ukuran model (kB)*** |
|---|---|---|---|
| Random Forest | 0.7960 | 0.055114 | 5929.70 |
| SVR | 0.0054 | 0.001814 | 1.06 |
| XGBoost | 0.4295 | 0.002516 | 92.36 |
| Gradient Boosting | 0.3357 | 0.008207 | 135.70 |




**Gambar 9.13** Efisiensi komputasi dari empat model regresi untuk arus drain MOSFET.

Secara keseluruhan, hasil ini menunjukkan bahwa keempat regressor dapat berfungsi sebagai pengganti yang efektif untuk pemodelan I–V transistor. GB adalah yang paling presisi, RF adalah yang paling konsisten, XGBoost memberikan keseimbangan terbaik antara akurasi dan efisiensi komputasi, dan SVR adalah yang paling ringan. Tergantung pada prioritas desain, baik akurasi, kecepatan, atau efisiensi memori, model yang berbeda dapat dipilih, menjadikannya alat praktis untuk pemodelan perilaku yang ringkas dan percepatan simulasi.

## 9.4 EVALUASI ALGORITMA REGRESI UNTUK PEMODELAN MOSFET

Studi ini membandingkan empat algoritma regresi: RF, GB, XGBoost, dan Support Vector Regression untuk memodelkan perilaku I–V statis MOSFET menggunakan data DC terukur. Semua model mencapai akurasi yang luar biasa ($R^2$ > 0,998, kesalahan pada orde $10^{-4}$ A), yang menegaskan kelayakan metode berbasis data untuk pemodelan transistor. Setiap algoritma menunjukkan keunggulan yang berbeda. GB memberikan RMSE terendah, RF memberikan MAE terendah, XGBoost memberikan keseimbangan terbaik antara akurasi dan efisiensi, dan SVR memberikan implementasi tercepat dan paling ringkas. Perbedaan ini memberikan fleksibilitas dalam memilih model berdasarkan prioritas desain seperti presisi, konsistensi, biaya komputasi, atau penggunaan memori.

Secara keseluruhan, hasil menunjukkan bahwa regressor pembelajaran mesin (ML) standar dapat berfungsi sebagai pengganti yang akurat dan praktis untuk pemodelan I–V transistor, dengan relevansi langsung pada eksplorasi desain awal, abstraksi perilaku, dan percepatan simulasi. Perluasan di masa mendatang akan membahas perilaku perangkat dinamis dan frekuensi tinggi, pendekatan hibrida yang diinformasikan fisika, dan validasi yang lebih luas di berbagai teknologi, menuju peran yang lebih terintegrasi untuk ML dalam otomatisasi desain elektronik.

# BAB 10
# INTEGRITAS PAD I/O PADA CHIP NEUROMORFIK HEMAT ENERGI

## 10.1 INTEGRITAS IOPAD DALAM SISTEM NEUROMORFIK

Dalam bab ini, kita akan meninjau kembali integritas IOPAD, menyoroti peran vitalnya dalam memungkinkan komputasi neuromorfik berdaya rendah dan berkeandalan tinggi. Tujuan kami adalah menghubungkan pengembangan algoritma neuromorfik dengan implementasi praktisnya dalam teknologi VLSI. Bab ini membahas secara mendalam integritas pad input/output dalam ranah desain VLSI neuromorfik berdaya rendah. Meskipun sebagian besar fokus di kalangan akademisi dan industri tertuju pada SNN dan desain logika inti berdaya sangat rendah, integritas listrik dan fungsional antarmuka I/O tetap menjadi tantangan yang agak terabaikan. Aspek ini sangat penting untuk membangun sistem perangkat keras yang mengonsumsi lebih sedikit energi. Sepanjang bab ini, kita akan mengeksplorasi desain IOPAD, yang merupakan komponen kunci perangkat keras neuromorfik yang secara langsung memengaruhi fidelitas sinyal dan konsumsi daya secara keseluruhan. Bab ini dibagi menjadi dua bagian utama. Pada bagian pertama, kita akan memperkenalkan konsep inti SNN dan relevansinya terhadap komputasi neuromorfik. Selanjutnya, bagian kedua bab ini membahas alur desain perangkat keras digital dasar, dari deskripsi perilaku hingga tata letak fisik.

Fase implementasi fisik kemudian dibahas, dengan menekankan proses CMOS Sky Water 130 nm sebagai platform yang hemat biaya dan layak secara industri untuk pembuatan prototipe dan investigasi neuromorfik. Bab ini berfokus pada integritas dan keandalan IOPAD, mengelola antarmuka antara logika internal dan dunia eksternal. Kompromi berdasarkan aspek daya dan kinerja yang terkait dengan pemilihan pustaka IOPAD, pendekatan desain cincin pad, dan konfigurasi pengikatan akan dibahas. Contoh dunia nyata akan disajikan terkait dengan prinsip-prinsip teori yang mendasarinya. Investigasi ini dibingkai dalam konteks yang lebih luas dari sistem neuromorfik hemat energi, menyoroti bagaimana perencanaan I/O tahap pertama dapat mencegah desain ulang yang mahal dan kehilangan hasil produksi pada fase pengembangan selanjutnya.

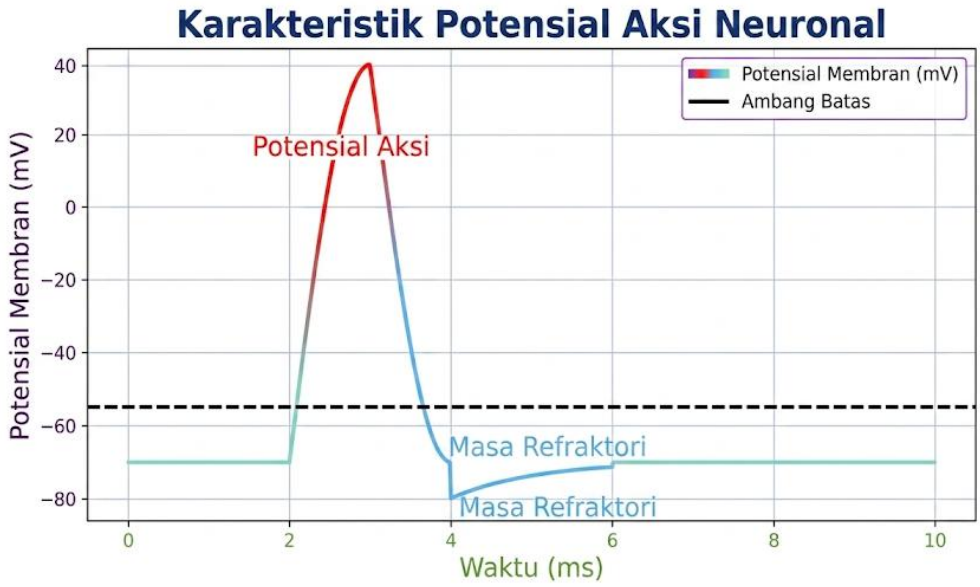
## 10.2 PENDAHULUAN TENTANG JARINGAN NEURAL SPIKING

### Relevansi Biologis

Bukti eksperimental telah menemukan bahwa jaringan neural spiking (SNN) yang masuk akal secara biologis dapat menggunakan waktu yang tepat dari potensial aksi untuk pemrosesan informasi, di mana kemungkinan representasi informasi dapat terletak pada waktu spike yang tepat. Dalam SNN, penting untuk dicatat bahwa sangat penting untuk mengetahui bagaimana data diproses. Produksi potensial aksi kecil atau spike mewakili komunikasi antar neuron. Potensial aksi memiliki amplitudo sekitar 100mV dan berlangsung selama 1–2 ms. Rangkaian potensial aksi yang terjadi disebut rangkaian spike, yang dapat terjadi secara deterministik atau acak. Setelah spike terjadi, tidak mungkin untuk segera

memicu spike lain. Periode waktu di mana tidak mungkin untuk memicu spike disebut 'periode refraktori'. Simulasi untuk potensial aksi diberikan pada Gambar 10.1.

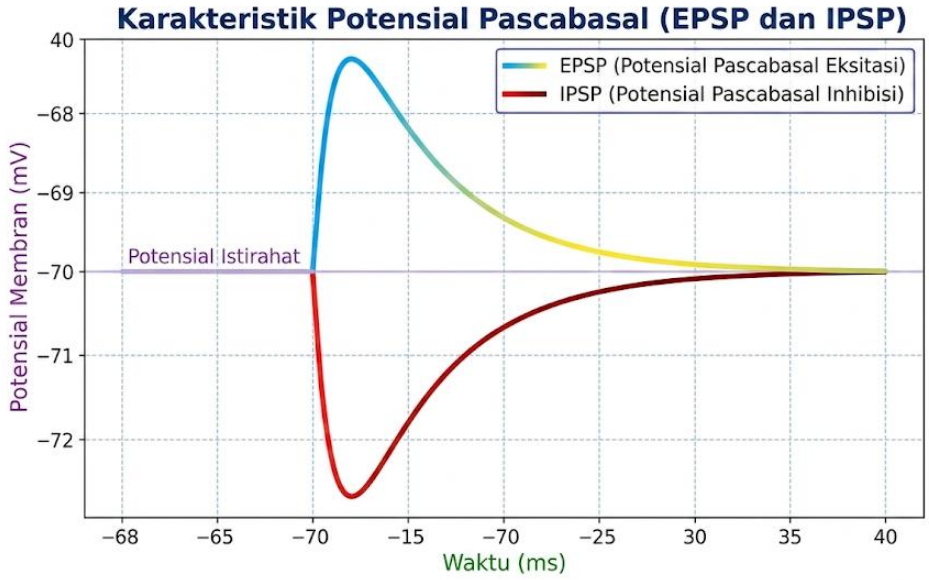
Proses yang terjadi di dalam sinaps dimulai ketika potensial aksi mencapai terminal presinaptik, yang memulai pelepasan molekul neurotransmiter. Sinaps rata-rata terdiri dari dua wilayah sel yang berbeda: membran presinaptik dan membran postsinaptik, dipisahkan oleh ruang ekstraseluler sempit yang disebut celah sinaptik. Di terminal presinaptik, neurotransmiter disimpan dalam vesikel sinaptik, masing-masing berdiameter sekitar 30–40 nm. Pembawa pesan kimia tersebut memainkan peran penting dalam menyampaikan sinyal melalui sinaps.




**Gambar 10.1** Potensial aksi.

Pengikatan neurotransmitter yang dilepaskan ke reseptor membran pascasinaps dapat menyebabkan terbukanya saluran ion, yang menyebabkan perubahan potensial membran pascasinaps. Hal ini dapat mengakibatkan depolarisasi (pergeseran positif pada potensial membran) atau hiperpolarisasi (pergeseran negatif) pada membran pascasinaps, bergantung pada apakah sinyal tersebut merupakan sinyal rangsang atau penghambat. Jika hasil bersih dari akumulasi sinyal melebihi tingkat ambang batas tertentu, maka potensial aksi akan dimulai di neuron pascasinaps dan kemudian merambat melalui aksonnya untuk mencapai neuron lain. Dua tipe utama EPSPs/IPSPs, yang juga diekspresikan dalam otak manusia melalui potensi postsinaptik rangsang dan penghambatan, menunjukkan dua tipe dasar respon sinaptik, dimana membran postsinaptik mempunyai kemungkinan terjadinya potensial aksi yang lebih besar atau peluang yang lebih rendah. Gambar 10.2 mengilustrasikan EPSP/IPSP, menggambarkan bagaimana neuron kortikal yang berbeda menampilkan pola potensial aksi dan integrasi sinyal yang kompleks, diselidiki secara lebih rinci oleh pembahasan yang disajikan oleh. Banyak jenis model neuron yang ada dalam literatur, masing-masing berbeda satu sama lain dalam hal tingkat akurasi biologis dan kompleksitas simulasi. Jenis model yang dipilih

biasanya bergantung pada tingkat abstraksi yang diperlukan untuk tugas simulasi saraf tertentu. Bagian selanjutnya ini akan memperkenalkan beberapa jenis model neuron spiking, beserta kekuatan, keterbatasan, dan kompleksitasnya.

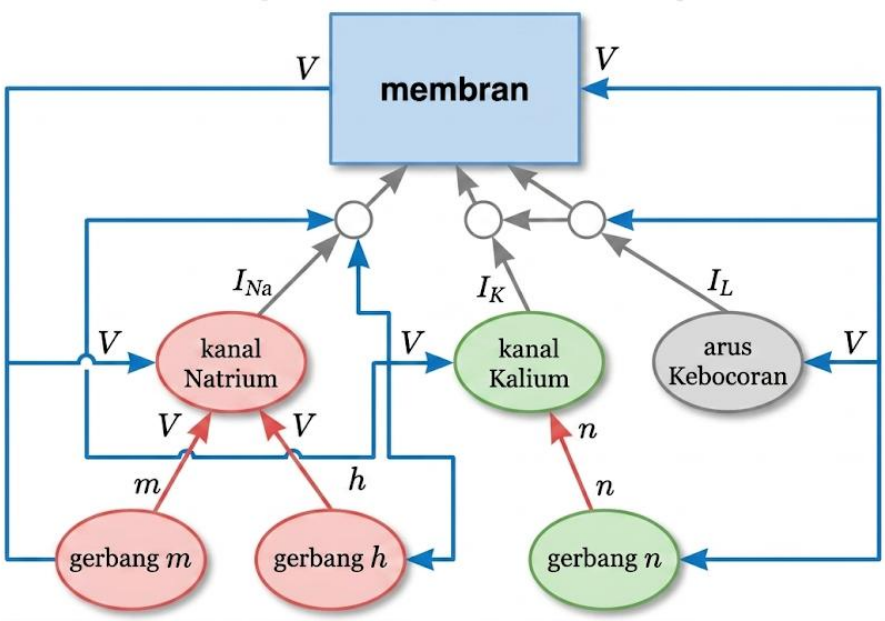



**Gambar 10.2** Potensi postsinaptik rangsang (positif) dan penghambatan (negatif).

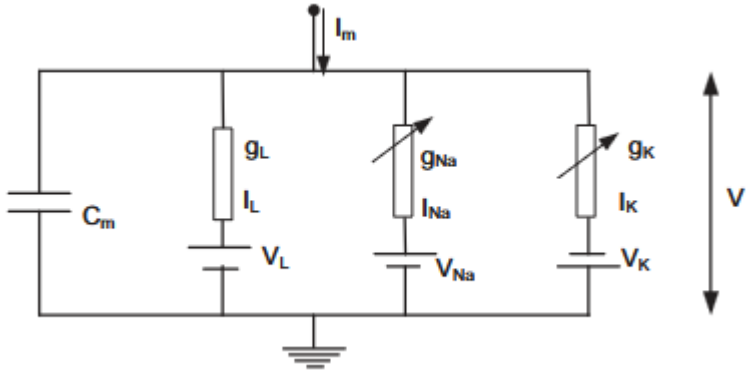
**Model *Hodgkin-Huxley* (HH)**

Salah satu model fundamental dalam bidang ilmu saraf komputasional adalah model *Hodgkin-Huxley* (HH). Dalam sebuah studi penting yang dilakukan pada tahun 1952, para ilmuwan Hodgkin dan A.F. Huxley meneliti aktivitas arus listrik pada akson raksasa cumi-cumi, memberikan wawasan berharga tentang mekanisme aktivitas pada neuron. Model ini secara proaktif berinteraksi melalui sistem empat dimensi, ditandai dengan adanya empat persamaan diferensial yang saling terkait. Contoh yang menggambarkan susunan struktural untuk model HH disajikan pada Gambar 10.3.

Berdasarkan hasil eksperimen, para penulis menurunkan model matematika yang dapat menjelaskan aktivitas listrik membran neuron selama transmisi sinyal dan eksitasi pada akson. Untuk memodelkan potensial aksi pada saraf cumi-cumi, Hodgkin dan Huxley memperkenalkan total tiga arus ionik, yaitu arus kalium ($I_K$), arus natrium ($I_{Na}$), dan arus bocor ($I_L$), yang kemudian diidentifikasi terutama dengan ion klorida. Kapasitansi membran disebut $C_m$, sedangkan $V_{Na}$, $V_K$, dan $V_L$ adalah potensial pembalikan untuk arus natrium, kalium, dan bocor, masing-masing. Ekspresi untuk arus membran total, Im, diberikan oleh persamaan (10.1) dan diilustrasikan secara grafis pada Gambar 10.4.




**Gambar 10.3** Deskripsi model HH.




**Gambar 10.4** Rangkaian listrik ekivalen HH.

$$I_m = C_m \frac{dV}{dt} + I_{Na} + I_K + I_L \tag{10.1}$$

Rumusan untuk arus ion diuraikan di bawah ini.

$$I_K = G_K n^4 (U - E_K) \tag{10.2}$$

$$I_{Na} = G_{Na} \mu^3 \theta (U - E_{Na}) \tag{10.3}$$

$$I_L = G_L (U - E_L) \tag{10.4}$$

di mana, $G_K, G_{Na}$, dan $G_L$ masing-masing menunjukkan konduktansi maksimal untuk saluran kalium, natrium, dan kebocoran. Istilah $E_K$, $E_{Na}$, dan $E_L$ mewakili potensial pembalikan (kesetimbangan) yang sesuai, sedangkan U menunjukkan tegangan transmembran relatif

terhadap tingkat referensi istirahat. Tabel 10.1 merangkum nilai parameter tipikal, yang diadaptasi dari kerangka kerja HH, dengan asumsi titik referensi di mana potensial istirahat diatur ke nol.

Dinamika temporal variabel gerbang ditangkap menggunakan persamaan kinetik orde pertama, di mana laju transisi alfa dan beta maju merupakan fungsi tegangan dan awalnya disesuaikan oleh Hodgkin dan Huxley untuk mencocokkan data akson cumi-cumi eksperimental.

**Tabel 10.1** Parameter model HH

| Saluran | $E_x$ $(mV)$ | $G_x$ $(mS/cm^2)$ |
|---|---|---|
| Na | 115 | 120 |
| K | -12 | 36 |
| L | 10.6 | 0.3 |

$$d\mu/dt = \alpha_\mu(U)(1-\mu) - \beta_\mu(U)\mu \tag{10.5}$$

$$d\eta/dt = \alpha_\eta(U)(1-\eta) - \beta_\eta(U)\eta \tag{10.6}$$

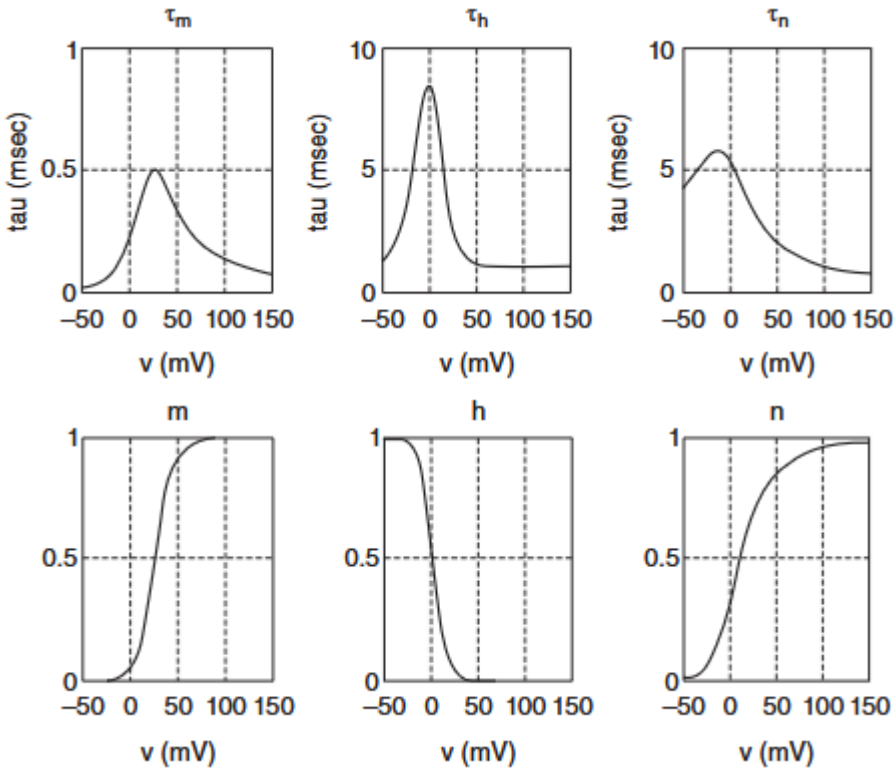
$$d\theta/dt = \alpha_\theta(U)(1-\theta) - \beta_\theta(U)\theta \tag{10.7}$$

Fungsi laju yang bergantung pada tegangan ($\alpha$ dan β) untuk setiap variabel gerbang tercantum dalam Tabel 10.2, di mana suku eksponensial dan ekspresi rasional menangkap sifat probabilistik dari pembukaan dan penutupan saluran.

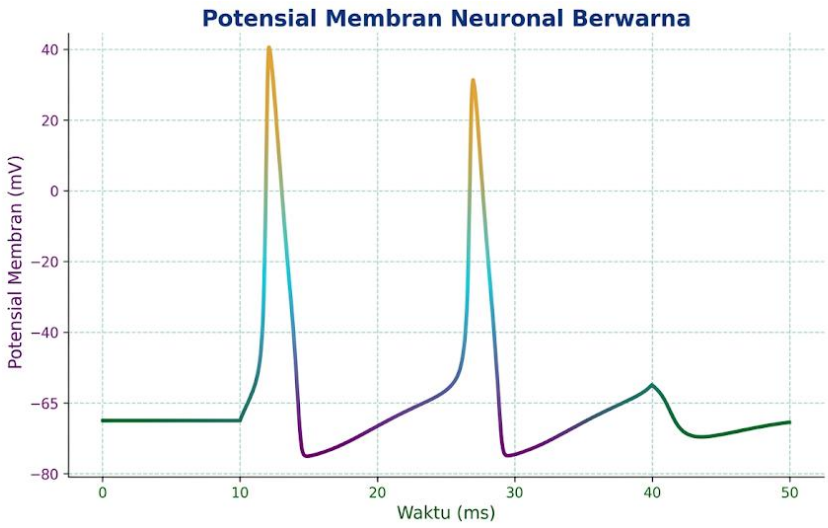
**Tabel 10.2** Fungsi laju untuk variabel gerbang

| Variabel | $\alpha_x(U)$ | $B_x(U)$ |
|---|---|---|
| $\eta$ (K) | $(0.1-0.01U)/[\exp(1-0.1U)-1]$ | $0.125\exp(-U/80)$ |
| $\mu$ (Aktivasi Na) | $(2.5-0.1U)/[\exp(2.5-0.1U)-1]$ | $4\exp(-U/18)$ |
| $\theta$ (Inaktivasi Na) | $0.07\exp(-U/20)$ | $1/\exp(3-0.1U)+1$ |

Output Matlab, yang disajikan pada Gambar 10.5, menunjukkan bahwa peningkatan tegangan membran $'v'$ menyebabkan penurunan variabel $'h'$, bersamaan dengan peningkatan $'m'$ dan $'n'$. Ini menyiratkan bahwa ketika stimulasi dari luar meningkat, membran sel $'m'$ juga meningkat, sehingga menyebabkan ion natrium mengalir ke dalam sel. Jika depolarisasi tersebut mencapai tingkat ambang batas, hal itu menyebabkan potensial aksi. Output juga menggambarkan bahwa konduktansi natrium pada akhirnya akan jenuh untuk nilai $'v'$ yang lebih tinggi karena $'h'$ berkurang, di mana konstanta waktu untuk $'m'$ lebih kecil daripada konstanta waktu untuk $'h'$. Karena konstanta waktu untuk m lebih kecil daripada konstanta waktu untuk $h$, $m$ merespons perubahan tegangan lebih cepat, sementara $h$ tertinggal. Selain itu, arus kalium mulai mengalir keluar pada skala waktu yang sebanding. Gambar 10.6 menyajikan inisiasi spike dan integrasi numeriknya menggunakan metode Euler dengan langkah waktu 0,01.




**Gambar 10.5** Model HH dan dampak konstanta waktu.



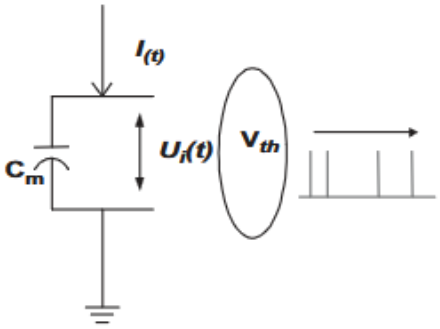

**Gambar 10.6** Integrasi model HH dengan integrasi Euler.

Model berbasis konduktansi sangat cocok untuk menggambarkan fenomena saraf seperti potensial aksi, pembangkitan spike, ambang batas, perilaku bursting, efek sinaptik, dan adaptasi spike. Namun, kelemahan utama dari model-model ini adalah kompleksitasnya, karena banyaknya parameter yang terlibat. Model-model ini juga membutuhkan komputasi yang besar dan berkinerja buruk ketika dijalankan pada sistem komputasi sekuensial tradisional. Pada bagian-bagian selanjutnya, beberapa model neuron yang disederhanakan

atau abstrak akan dibahas. Alternatif ini lebih efisien secara komputasi daripada model tipe Hodgkin-Huxley, sementara tetap menangkap banyak fitur penting dari neuron nyata.

**Model Integrasi dan Tembakan (I&F)**

Gambar 10.7 mengilustrasikan struktur dasar yang digunakan untuk memodelkan model Integrasi dan Tembakan (I&F). Kapasitor dalam model mengakumulasi arus masuk $I(t)$. Ketika mencapai nilai ambang batas $\theta$, neuron menembakkan lonjakan, dan tegangan kapasitor langsung diatur ulang menjadi nol. Ekspresi matematika untuk model I&F diberikan di bawah ini.

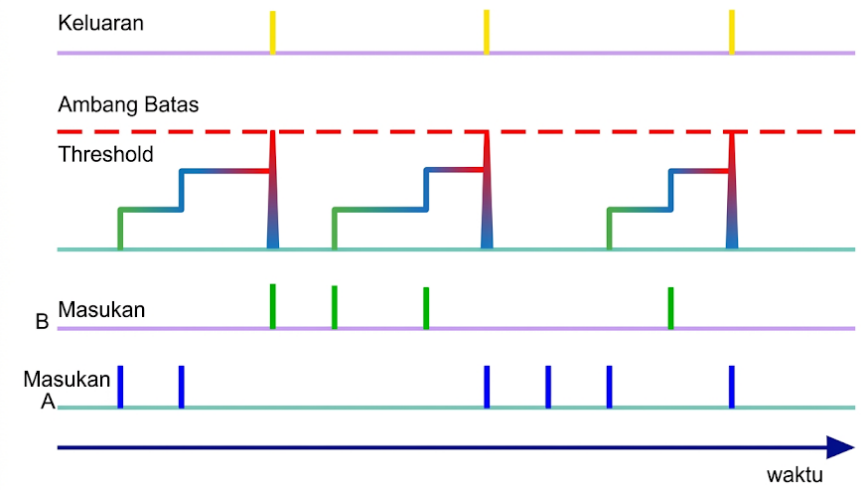



**Gambar 10.7** Model integrasi dan kebakaran tanpa kebocoran.

$$I(t) = C_m \frac{du_i(t)}{dt} \tag{10.8}$$

Dalam model matematika ini, '$C_m$' adalah kapasitansi membran, '$U_i(t)$' adalah potensial membran, dan '$I(t)$' adalah arus masukan. Model ini menggambarkan bagaimana neuron mengintegrasikan rangsangan dan melepaskan impuls ketika potensial membran melebihi tingkat ambang tertentu.

Namun, terlepas dari kesederhanaannya, model ini tidak realistis dari perspektif biologis, karena neuron biologis sebenarnya tidak dapat mengintegrasikan rangsangan dalam waktu nol. Kedua, model I&F tidak memperhitungkan keberadaan arus bocor, yang merupakan parameter kritis ketika menggambarkan perilaku pengisian dan pengosongan membran yang bergantung pada waktu. Pola aktivitas neuron I&F disajikan pada Gambar 10.8.



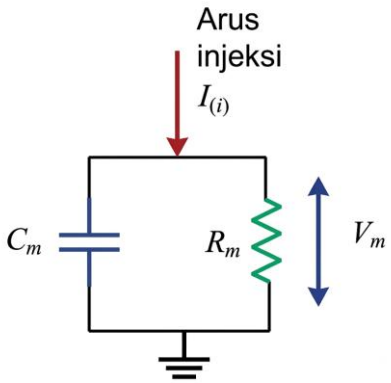

**Gambar 10.8** Perilaku neuron integrate and fire.

**Model Integrate and Fire (LIF) dengan kebocoran**

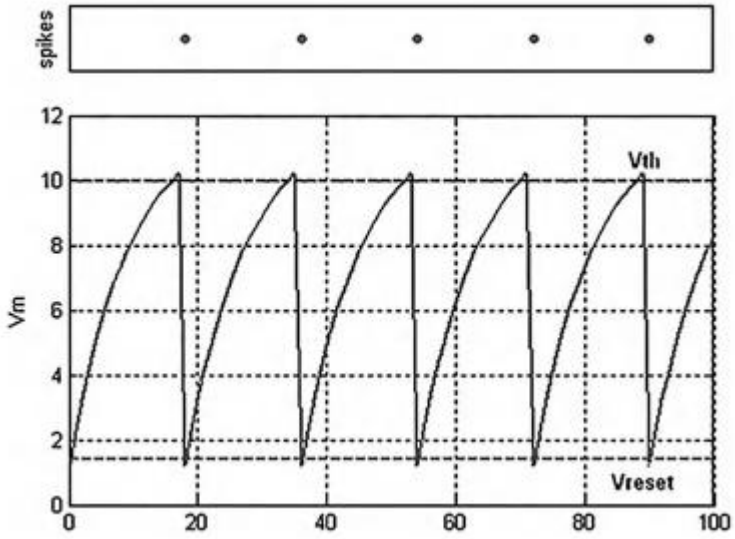
Model *leaky integrate-and-fire* (LIF) umumnya digunakan untuk menggambarkan perilaku firing neuron berdasarkan karakteristik utama seperti konstanta waktu membran, ambang firing, dan periode refraktori. Salah satu penyederhanaan utama dalam model ini adalah mengabaikan bentuk detail potensial aksi, hanya berfokus pada waktu yang tepat dari spike. Berbeda dengan model I&F yang diidealkan, neuron yang masuk akal secara biologis menunjukkan perilaku integrator bocor yang secara bertahap mengembalikan potensial membran ke keadaan istirahatnya ketika tidak ada input. Efek kebocoran ini dimodelkan dengan memasukkan resistansi membran $R_m$, seperti yang diilustrasikan pada Gambar 10.9. Representasi matematis dari model ini ditunjukkan sebagai berikut:

$$I(t) = C_m \cdot \frac{dV_m}{dt} + \frac{V_m}{R_m} \tag{10.9}$$

Gambar 10.10 menunjukkan hasil simulasi Matlab untuk neuron LIF. Dalam model ini, sinyal yang masuk diintegrasikan dari waktu ke waktu, dan ketika potensial membran melampaui ambang batas yang telah ditentukan $V_{th}$, ia menghasilkan lonjakan dan segera kembali ke tingkat istirahat $V_{reset}$. Meskipun neuron LIF dapat mereplikasi beberapa fitur neurokomputasi inti—khususnya, integrasi sinaptik, perilaku reset, dan ambang batas membran—kemampuannya terbatas dibandingkan dengan model yang lebih detail. Seperti yang dicatat oleh, model LIF hanya memperhitungkan 3 dari 20 sifat kunci yang diamati pada neuron biologis. Sebagai representasi satu dimensi, ia kekurangan banyak karakteristik penting dari perilaku lonjakan kortikal. Meskipun demikian, ia tetap berguna untuk mensimulasikan aktivitas neokortikal skala besar dalam kasus di mana fidelitas biologis yang tepat tidak diperlukan. Karena model neuron yang disederhanakan meningkatkan kejelasan konseptual dan kemudahan analisis, model LIF telah banyak diadopsi dalam berbagai studi teoretis dan komputasi. Kemudahan implementasinya juga menjadikannya pilihan populer dalam berbagai aplikasi teknik.




**Gambar 10.9** Model neuron integrate-and-fire dengan kebocoran.




**Gambar 10.10** Simulasi Matlab dari neuron LIF.

**Model Respons Lonjakan (SRM)**

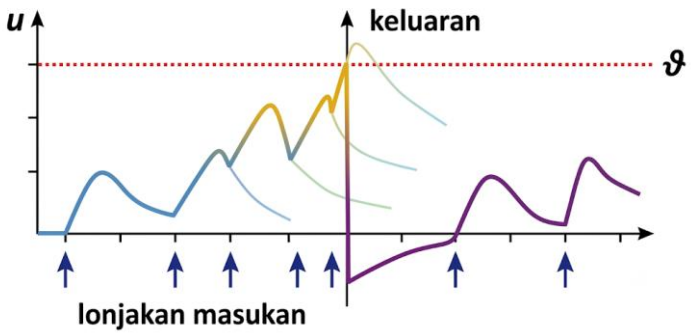
Model respons lonjakan (SRM) merupakan perluasan dari model LIF, yang menawarkan fleksibilitas lebih dalam proses pemodelannya terkait perilaku neuron. SRM juga berbeda dari model LIF, baik non-linier maupun lainnya, dengan menggunakan waktu yang telah berlalu sejak lonjakan terakhir alih-alih fungsi yang terkait dengan potensial membran. Perbedaan utama lainnya terletak pada bentuk yang digunakan untuk memodelkan elemen secara matematis, di mana model I&F memerlukan penggunaan persamaan diferensial, sedangkan persamaan SRM menyertakan potensial membran pada waktu t dengan integral yang berkaitan dengan aktivitas masa lalu.

Representasi matematis dari SRM diberikan di bawah ini.

$$u_i(t) = \sum_{t_i^{(f)} \in F_i} \eta_i\left(t - t_i^{(f)}\right) + \sum_{j \in \Gamma_i} \sum_{t_j^{(f)} \in F_j} w_{ij}\, \epsilon_{ij}\left(t - t_i^{(f)}\right) \qquad (10.10)$$

Simbol $\eta_i(s)$ memodelkan perilaku refraktori dari neuron. Ini digunakan untuk mengatur ulang neuron dan secara bertahap meluruh menjadi nol seiring berjalannya waktu. Untuk setiap waktu $s < 0$, $\eta_i(s) = 0$. $\epsilon_{ij}(s)$ disebut sebagai bentuk dari potensial pascasinapsis (PSP). Hal ini bergantung pada aktivitas lonjakan dari neuron prasinapsis dan memengaruhi potensial membran dari neuron $i$. Nilainya adalah nol untuk $s < 0$. Keadaan setiap neuron pada waktu $t$ dinotasikan oleh $u_i(t)$. Kekuatan, atau bobot, $w_{ij}$, memodulasi amplitudo dari PSP. Kumpulan waktu penembakan untuk neuron $j$ dinotasikan oleh $F_j$, dan ini didefinisikan untuk himpunan neuron prasinapsis terhadap sebuah neuron $i$, yang dinotasikan oleh $\Gamma_i$.

$U_i(t)$ tidak berubah dari tingkat istirahatnya ketika tidak ada spike yang masuk, menurut model tersebut. Potensial membran diubah oleh kontribusi yang ditunjukkan oleh ϵij untuk setiap lonjakan yang masuk, memodelkan bagaimana neuron merespons lonjakan yang masuk. Potensial membran meningkat melampaui ambang batas yang ditunjukkan oleh θ, menyebabkan neuron menghasilkan lonjakan keluaran, setelah itu potensial membran kembali ke tingkat istirahatnya. Untuk memodelkan fenomena refraktori, parameter tambahan η ditambahkan selama proses pengaturan ulang ini, memastikan bahwa tidak ada masukan baru selama periode tersebut.




**Gambar 10.11** Model respons lonjakan.

Gambar 10.11 di atas mengilustrasikan proses yang terjadi dalam model respons lonjakan (SRM). Lonjakan yang masuk ditambahkan pada interval waktu yang berbeda, sehingga berkontribusi pada potensial membran. Garis putus-putus pada gambar menunjukkan ambang batas, di mana, ketika total potensial terlampaui, PSP keluaran dihasilkan. Periode diam neuron, tergantung pada η, terjadi setelah produksi keluaran lonjakan.

**Model Neuron Izhikevich**

Model neuron Izhikevich adalah model dua dimensi yang realistis secara biologis yang dapat dijelaskan menggunakan persamaan diferensial berpasangan. Model ini mampu mereproduksi pola penembakan yang diamati pada semua jenis neuron kortikal yang diketahui. Dinamika model ini diatur oleh dua persamaan diferensial yang saling terkait:

$$\frac{du}{dt} = 0.04u^2(t) + 5u(t) + 140 - w(t) + I(t) \qquad (10.11)$$

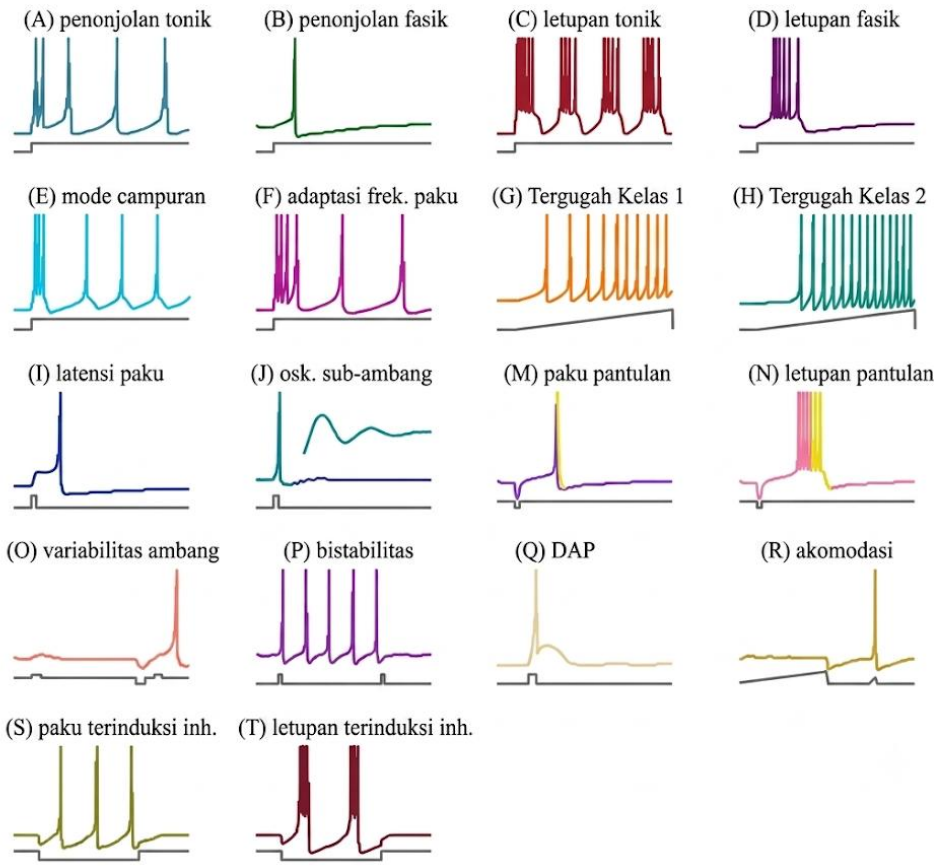
$$\frac{dw}{dt} = a\big(bu(t) - w(t)\big) \qquad (10.12)$$

Pengaturan ulang lonjakan (*spike resetting*) digambarkan seperti yang ditunjukkan di bawah ini:

$$\text{If } u \geq \theta \text{ then } u \leftarrow c \tag{10.13}$$

$$\text{and } w \leftarrow w + d \tag{10.14}$$

Model ini didefinisikan menggunakan variabel u, yang menunjukkan potensial membran, dan $w$, yang menunjukkan variabel pemulihan. Parameter $a, b, c$, dan $d$ juga tidak berdimensi. Untuk menjamin bahwa $u$ dalam milivolt (mV) dan waktu dalam milidetik (ms), ekspresi yang dipilih adalah $0{,}04u^2(t) + 5u(t) + 140$. Dengan nilai yang berbeda untuk parameter yang dipilih, dimungkinkan untuk membuat pola yang berbeda menggunakan model, diilustrasikan pada Gambar 10.12 dan dibahas oleh.




**Gambar 10.12** Model neuron Izhikevich. (a) lonjakan tonik, (b) lonjakan fasik, (c) ledakan tonik, (d) ledakan fasik, (e) mode campuran, (f) adaptasi frekuensi lonjakan, (g) kelas 1 yang dapat dirangsang, (h) kelas 2 yang dapat dirangsang, (f) latensi lonjakan, (j) osilasi subambang, (m) lonjakan pantulan, (n) ledakan pantulan, (o) variabilitas ambang batas, (p) bistabilitas, (q) DAP, (r) akomodasi, (s) penghambatan lonjakan terinduksi, (t) penghambatan ledakan terinduksi.

## 10.3 EVALUASI MODEL NEURON SPIKING DALAM KOMPUTASI

Berbagai jenis model spiking untuk neuron telah dibahas di bagian bab buku ini. Jenis yang dipilih bergantung pada seberapa besar akurasi biologis yang dibutuhkan, dibandingkan dengan seberapa efisien komputasinya. Jenis yang memiliki banyak akurasi biologis, seperti yang ditawarkan oleh model HH atau model serupa, memiliki akurasi biologis yang terlalu tinggi untuk sesuatu seperti komputasi saraf, di mana ia harus cepat dan efisien secara komputasi. Selain itu, model-model tersebut terlalu mahal secara komputasi, membutuhkan terlalu banyak daya pemrosesan untuk dihitung.

Dengan demikian, model tersebut tidak begitu berguna ketika mencoba menghitung jaringan saraf buatan untuk digunakan dalam pembelajaran mesin atau aplikasi teknik pada unit pemrosesan komputer standar, karena akan membutuhkan terlalu banyak daya pemrosesan, sehingga menjadi mahal secara komputasi. Model lain yang kurang mahal secara komputasi, seperti model I&F, terlalu sederhana untuk dihitung dan kurang memiliki akurasi biologis. Model ini hanya menghitung perbedaan waktu untuk penembakan neuron, sehingga mudah dihitung karena tidak menghitung terlalu banyak faktor biologis yang ada pada neuron sebenarnya. Model ini membutuhkan daya pemrosesan yang rendah, sehingga memudahkan komputasi jaringan saraf tiruan untuk digunakan dalam pembelajaran mesin atau aplikasi teknik.

Pilihan antara akurasi biologis yang terlalu tinggi dan akurasi biologis yang terlalu rendah, tergantung pada apakah itu untuk pembelajaran mesin atau aplikasi teknik, mungkin tidak sesederhana itu. Jika tujuannya untuk mempelajari proses biologis yang ada di balik cara kerja neuron sebenarnya, maka model seperti model HH akan menjadi pilihan yang lebih baik karena memiliki tingkat akurasi biologis yang lebih tinggi, memberikan model dan representasi yang lebih akurat. Tetapi untuk pembelajaran mesin atau aplikasi teknik, jaringan saraf tiruan yang membutuhkan daya pemrosesan lebih rendah, seperti model I&F, akan memberikan hasil yang lebih baik karena hanya membutuhkan daya pemrosesan yang lebih sedikit untuk menghitung jaringan saraf tiruan untuk digunakan dalam pembelajaran mesin atau aplikasi teknik.

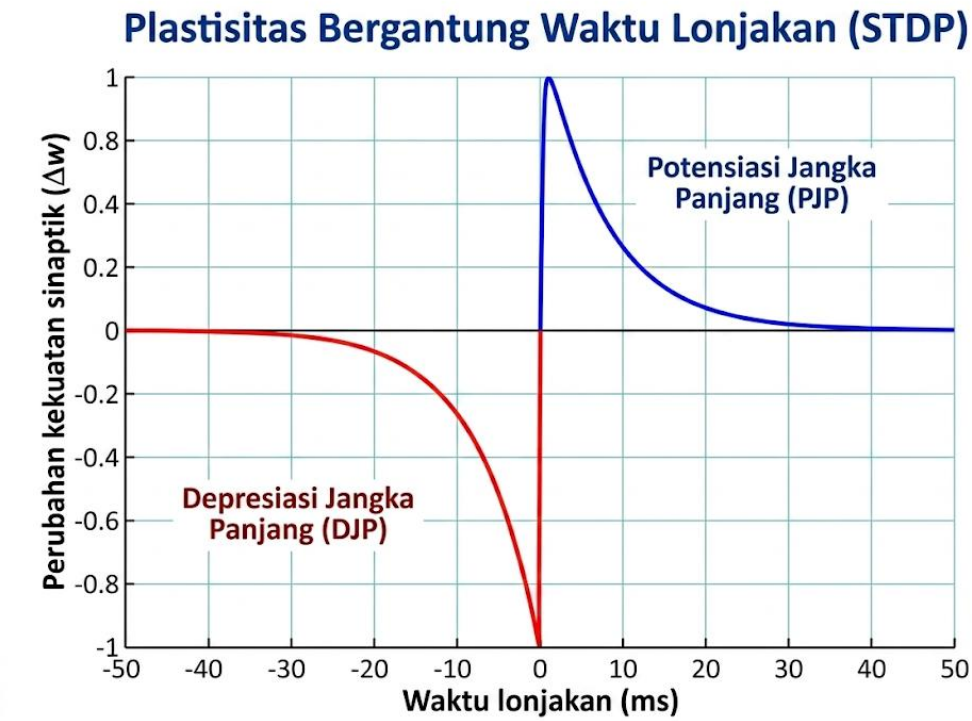
Oleh karena itu, kekuatan sejati dalam jaringan saraf berasal dari kemampuan belajarnya. Baik itu belajar dengan membedakan pola dalam data atau dengan membuat keputusan melalui pengalaman masa lalu, pembelajaranlah yang memungkinkan jaringan saraf menjadi sangat berharga. Algoritma pembelajaran juga memungkinkan jaringan saraf menjadi lebih baik dari waktu ke waktu, dan penting bagi algoritma pembelajaran dalam jaringan saraf untuk dipahami sepenuhnya untuk aplikasi yang membutuhkan jaringan saraf untuk melakukan tugas yang lebih kompleks. Pada bagian selanjutnya, kita akan mengeksplorasi bagaimana pembelajaran terjadi di SNN.

**Pembelajaran dalam Jaringan Saraf Spiking**

Hipotesis plastisitas sinaptik, yang asal-usulnya berasal dari pembahasan asli Donald Hebb (1949), telah lama menjadi model dasar pembelajaran dalam jaringan saraf buatan. Plastisitas Hebbian, yang secara khusus didefinisikan melalui potensiasi jangka panjang (LTP)

dan depresi, adalah dasar bagi sebagian besar model pembelajaran. Plastisitas Bergantung Waktu Spike (STDP) adalah salah satu mekanisme terpenting dalam paradigma ini, plastisitas Hebbian yang terkunci waktu pada spike dalam aktivitas saraf. Dalam STDP, jika spike presinaptik diikuti oleh spike postsinaptik, maka sinapsis diperkuat, dan jika diikuti oleh spike postsinaptik, maka sinapsis mengalami depresi.

Daya tahan LTP, atau memori sinaptik, adalah fungsi waktu yang menurun secara eksponensial seiring waktu, ditentukan oleh konstanta waktu membran. Ini berarti bahwa hanya lonjakan presinaptik yang tiba dalam jendela waktu tertentu relatif terhadap lonjakan postsinaptik yang berkontribusi pada potensiasi sinaptik. Sebaliknya, jika lonjakan presinaptik tiba sesaat setelah potensial aksi postsinaptik, sinapsis mengalami depresi. Ketika waktu lonjakan presinaptik dan postsinaptik terlalu jauh, tidak ada perubahan pada bobot sinaptik. Jendela waktu yang paling umum digunakan untuk mengatur bobot sinaptik diperoleh dari eksperimen yang melibatkan neuron hipokampus tikus yang dikultur, seperti yang dilaporkan oleh. Saya melakukan simulasi untuk mereplikasi perilaku STDP, dan hasilnya ditunjukkan pada Gambar 10.13. Pada gambar tersebut, bagian kanan atas menggambarkan LTP, sedangkan bagian kiri bawah mewakili Depresi Jangka Panjang (LTD). Simulasi ini berharga karena memberikan wawasan tentang kemungkinan merancang sirkuit berdasarkan model sinapsis ini untuk aplikasi pembelajaran on-chip.




**Gambar 10.13** Simulasi Matlab dari plastisitas berbasis STDP.

STDP dapat dimodelkan dengan memperkirakan kontribusi lonjakan prasinaptik dan pascasinaptik terhadap perubahan sinaptik. Modifikasi sinaptik seiring waktu dapat diekspresikan menggunakan fungsi eksponensial:

Untuk $t < 0$, perubahan diberikan oleh $A + exp\ (t/\tau +)$, di mana $A +$ lebih besar dari 0.

Untuk $t \geq 0$, perubahan mengikuti $A - exp\ (-t/\tau -)$, dengan $A -$ kurang dari 0.

Parameter $\tau +$ dan $\tau -$ mendefinisikan jendela waktu di mana perubahan sinaptik dapat terjadi, tergantung pada waktu lonjakan prasinaptik dan pascasinaptik. Konstanta $A +$ dan $A -$ menetapkan perubahan maksimum yang mungkin terjadi pada kekuatan sinaptik, dengan perubahan paling signifikan terjadi ketika lonjakan prasinaptik dan pascasinaptik terjadi berdekatan. Pada Gambar 10.13, waktu lonjakan diwakili oleh perbedaan waktu penembakan antara neuron prasinaptik dan pascasinaptik. Ketika perbedaan waktu $\Delta t$ lebih besar dari 0 dan mendekati nol, bobot sinaptik meningkat. Perubahan sinaptik ($\Delta w$) diwakili sebagai persentase. Berbagai bentuk STDP telah dibahas dalam literatur, dan perbedaan utama di antara model-model ini terletak pada simetri antara LTP dan LTD, serta diskontinuitas yang diamati dalam fungsi $\Delta w$.

**Pembelajaran Hebbian**

Salah satu prinsip pembelajaran yang paling terkenal diusulkan oleh ahli neuropsikologi Kanada Donald O. Hebb pada tahun 1949. Karyanya memperkenalkan konsep yang menjelaskan bagaimana pembelajaran terjadi melalui koneksi sinaptik antar neuron. Aturan Hebb menyatakan:

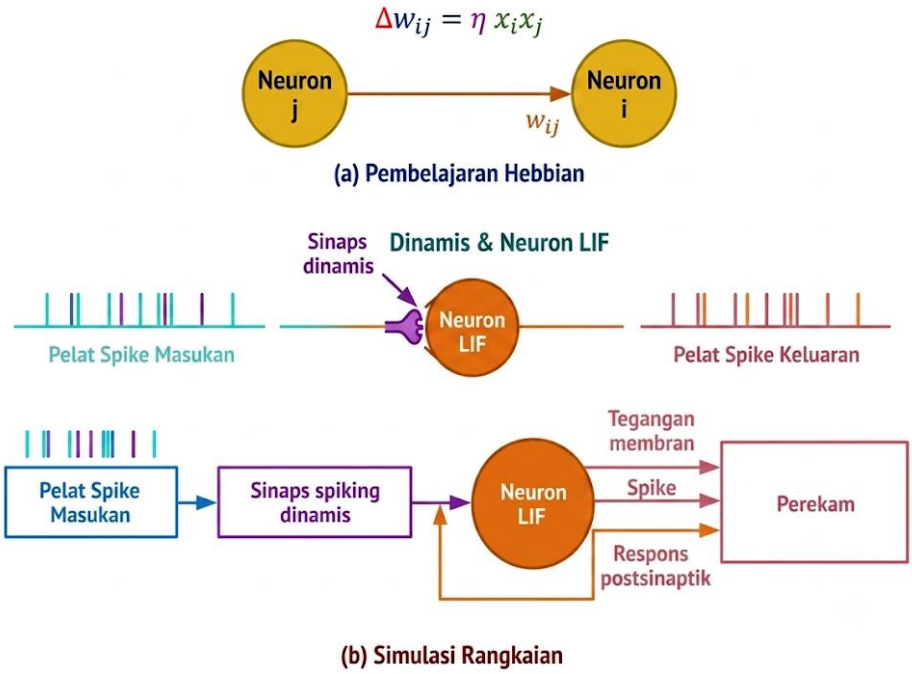
> Ketika akson sel A cukup dekat untuk merangsang sel B dan secara konsisten berpartisipasi dalam mengaktifkannya, beberapa perubahan pertumbuhan atau metabolisme terjadi pada satu atau kedua sel, sehingga efektivitas A dalam mengaktifkan B meningkat.

Secara sederhana, ketika dua neuron yang terhubung oleh sinapsis diaktifkan dalam jendela waktu tertentu, kekuatan sinapsis tersebut meningkat. Namun, jika neuron diaktifkan di luar jendela waktu ini, kekuatan sinapsis melemah, dan proses pembelajaran berdasarkan prinsip ini dikenal sebagai pembelajaran Hebbian. Pembelajaran Hebbian memiliki beberapa karakteristik utama, seperti tidak diawasi dan terjadi secara lokal, yang menambah plausibilitas biologisnya. Ini adalah konsep yang dapat berguna ketika melakukan benchmarking atau mendesain sirkuit untuk implementasi perangkat keras dan perangkat lunak.

Secara matematis, aturan Hebb untuk menyesuaikan bobot $W_{ij}$ dari neuron $j$ ke neuron $i$, dengan laju pembelajaran $\eta$, dapat dinyatakan sebagai berikut dan ditunjukkan pada Gambar 10.14a:

$$\Delta w_{ij} = n x_i x_j$$

Output dari neuron ke-$i$ diberikan oleh $X_i = F\ W_{ij}\ X_j$, di mana F adalah fungsi linier atau non-linier untuk mengubah input menjadi output. Perubahan bobot sinaptik, $\Delta w_{ij}$, diberikan sebagai $n x_i x_j$, di mana $\eta$ adalah laju pembelajaran. Aturan pembelajaran Hebbian bersifat asosiatif karena neuron prasinaptik dan pascasinaptik perlu diaktifkan dalam periode waktu terbatas untuk mengubah bobot sinaptik. Salah satu keterbatasan aturan ini adalah tidak adanya batasan pada bobot koneksi, yang mengakibatkan pertumbuhan tak terbatas. Namun, masalah ini dapat diperbaiki dengan menyertakan fungsi peluruhan pada setiap koneksi sehingga bobot tetap berada dalam batasan yang wajar.




**Gambar 10.**14 (a) Pembelajaran Hebbian. (b) Simulasi sirkuit.

**Pertimbangan Desain VLSI Untuk Jaringan Saraf Spiking**

Tujuan bagian ini adalah untuk membahas realisasi platform perangkat keras hemat daya yang dapat meniru karakteristik neuron spiking dengan akurasi yang sangat baik. Sirkuit VLSI yang terinspirasi neuro didasarkan pada studi sistem saraf biologis dan mencoba meniru kemampuan jaringan saraf pada tingkat perangkat keras. Konsumsi energi chip ini diminimalkan karena dirancang untuk beroperasi pada daya rendah, dan mereka melakukan komputasi dengan menggunakan paralelisme dan dinamika temporal neuron spiking dengan cara yang hemat energi.

Desain sirkuit untuk VLSI neuromorfik menggabungkan berbagai aspek, seperti STDP, perubahan bobot sinaptik, dan pembangkitan spike secara dinamis. Dengan mendedikasikan perangkat keras untuk simulasi proses-proses ini, chip VLSI neuromorfik memberikan tingkat kinerja dan efisiensi energi yang jauh lebih unggul daripada prosesor digital klasik untuk aplikasi dalam pemrosesan sensorik, pembelajaran, dan pengenalan pola.

**Neuron LIF dan Simulasi Sirkuit**

Untuk menunjukkan fungsionalitas SNN pada tingkat sirkuit, kami memeriksa perilaku SNN perangkat keras yang dibuat dengan neuron LIF menggunakan bahasa deskripsi perangkat keras Verilog. Fokus utama kami adalah pada perilaku perangkat keras sirkuit dan bagaimana perilaku tersebut dapat dioptimalkan untuk implementasi yang hemat energi. Neuron input spiking dihubungkan ke neuron LIF melalui model sinapsis dinamis yang mensimulasikan dinamika temporal dan sinaptik dengan cara yang mirip dengan biologi. Pengaturan yang digambarkan pada Gambar 10.14b dieksekusi dengan kode Verilog, yang menghasilkan simulasi bentuk gelombang terperinci yang membantu kami memahami perilaku temporal dan respons jaringan.

Simulasi ini menangkap berbagai jejak penting, seperti tegangan membran neuron LIF, PSP yang dihasilkan pada sinapsis, dan lonjakan keluaran yang dihasilkan oleh neuron sebagai reaksi terhadap input yang masuk. Potensial membran neuron LIF mengumpulkan respons postsynaptic dari waktu ke waktu, dan ketika potensial ini melampaui ambang batas yang telah ditetapkan sebelumnya, potensial aksi terjadi. Lonjakan ini, pada gilirannya, menunjukkan perilaku penembakan neuron, yang sangat penting untuk perhitungan jaringan saraf.

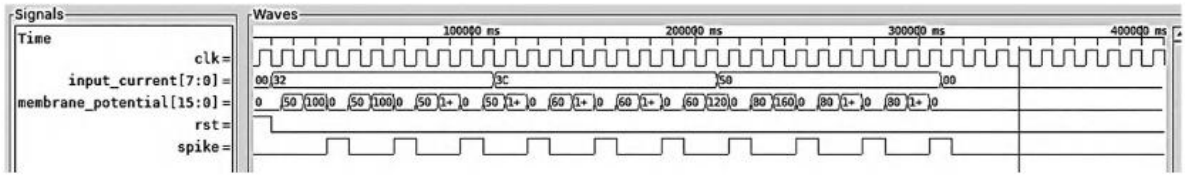
Bentuk gelombang yang diilustrasikan pada Gambar 10.15 menunjukkan evolusi waktu potensial membran neuron LIF, sehingga menggambarkan integrasi bertahap sinyal input dan penembakan lonjakan berikutnya. Tegangan membran dipengaruhi oleh lonjakan input presinaptik baik dari segi waktu dan kekuatannya, serta sifat koneksi sinaptik, yaitu, bobot sinaptik dan konstanta waktu sinapsis dinamis.

**Integritas Pad I/O dan Efisiensi Daya**

Dalam proses pengembangan chip VLSI neuromorfik, sangat penting untuk menjaga integritas pad I/O untuk memastikan kinerja chip, terutama ketika merupakan bagian dari sistem yang lebih besar dan kompleks. Pad I/O ini termasuk antarmuka utama chip dan lingkungan eksternalnya. Pad I/O harus mentransfer sinyal sementara daya dan keandalan transmisi sinyal tetap terjaga. Masalah ini diperburuk dalam kasus chip neuromorfik hemat energi, di mana tujuannya adalah untuk mendapatkan yang terbaik dari kedua dunia. Desain cincin pad I/O adalah salah satu faktor fundamental yang menentukan seberapa efisien chip akan bekerja secara keseluruhan. Alur desain RTL ke GDSII kami di bagian bab ini pertama-tama membuat tata letak chip untuk model SNN LIF sederhana dan kemudian beralih ke cincin pad I/O, yang tidak hanya memastikan kehilangan sinyal minimum tetapi juga efisiensi daya.

Desain pad I/O dibahas dari berbagai sudut pandang, dan sirkuit proteksi ESD dan buffer dijelaskan yang mengontrol aliran sinyal masuk dan keluar dari chip. Sirkuit ini sangat penting untuk menjaga level tegangan, tanpanya transfer data akan tidak akurat, dan degradasi sinyal akan terjadi. Faktor penting lainnya dalam desain pad I/O adalah pengurangan refleksi sinyal dan mitigasi noise, karena keduanya dapat merusak integritas sinyal dan akibatnya, kinerja keseluruhan chip. Oleh karena itu, kami mempertimbangkan berbagai tata letak pad I/O dalam simulasi chip neuromorfik kami untuk mengurangi hal ini. Dengan cara ini, kami dapat memantau pengaruhnya terhadap kualitas sinyal dan penggunaan daya.

Pembahasan menunjukkan bahwa cincin pad I/O yang dioptimalkan dengan baik dapat memastikan transmisi sinyal yang andal dan, pada saat yang sama, sangat mengurangi disipasi daya dengan memangkas penguatan sinyal yang tidak perlu. Secara keseluruhan, desain yang cermat dari pad I/O dan sirkuit terkaitnya menjamin bahwa chip neuromorfik mempertahankan efisiensi tinggi, menyediakan komunikasi yang akurat dan cepat dengan konsumsi energi rendah, yang pada gilirannya merupakan karakteristik penting untuk mencapai target kinerja sistem neuromorfik kontemporer.




**Gambar 10.15** Simulasi neuron pemicu sederhana yang menangkap dinamikanya.

**Hasil Simulasi Sirkuit VLSI**

Simulasi sirkuit neuromorfik sangat penting dalam menentukan baik keterbatasan maupun manfaat penggunaan neuron spiking perangkat keras. Terutama, kami menggunakan kode Verilog untuk mensimulasikan neuron LIF dengan interaksi sinaptiknya, memungkinkan kami untuk menggambarkan bentuk gelombang yang sangat halus yang sesuai dengan potensial membran, respons pasca-sinaptik, dan lonjakan keluaran, seperti yang diilustrasikan pada Gambar 10.14. Bentuk gelombang ini sangat penting untuk mengkonfirmasi operasi model sinapsis dinamis dan memberikan pemahaman tentang bagaimana sistem akan bekerja dalam praktiknya.

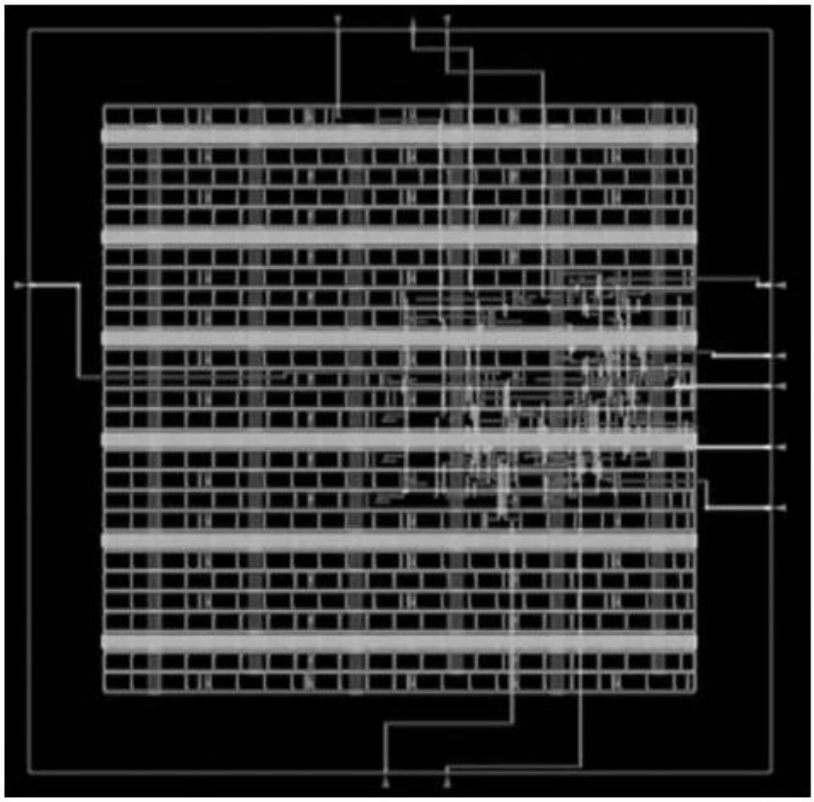
Simulasi juga menghasilkan perbandingan model sinaptik yang berbeda, menunjukkan cara yang paling sesuai untuk mensimulasikan plastisitas sinaptik dalam skenario perangkat keras. Interpretasi ini sangat penting untuk penyetelan kinerja chip neuromorfik, karena mendorong penyesuaian parameter kritis, seperti konstanta waktu sinaptik, laju pembelajaran, dan waktu kejadian lonjakan. Selain itu, penyempurnaan parameter-parameter ini berdasarkan data simulasi memastikan bahwa chip tersebut tidak hanya menjadi yang paling efisien tetapi juga mereplikasi dinamika jaringan saraf biologis secara akurat. Kemudian kami menggunakan hasil simulasi untuk menganalisis dan melihat apakah hasilnya sesuai dengan perilaku yang diharapkan dari model neuron LIF. Membran neuron menjumlahkan lonjakan input yang masuk, dan segera setelah potensial membran melampaui ambang batas, lonjakan output dihasilkan.

Sirkuit tersebut mampu meniru proses saraf fundamental dari pembangkitan lonjakan, plastisitas sinaptik, dan integrasi temporal, semuanya melalui penggunaan teknik yang sama yang memberikan efisiensi komputasi dan plausibilitas biologis yang cukup. Sebuah file GDSII dihasilkan bersamaan dengan Sky130 nm PDK setelah simulasi berhasil diverifikasi, dan tata letak dengan konfigurasi pin ditampilkan pada Gambar 10.16. Tata letak chip memiliki pin I/O di sisi Timur, Barat, Utara, dan Selatan, dan pin-pin ini diperlukan untuk menguji fungsionalitas chip: sinyal clk, potensial membran, lonjakan input, output lonjakan, dan sinyal reset.

Sangat penting untuk mengevaluasi statistik sintesis desain secara menyeluruh sebelum menambahkan padding I/O ke file layout. Langkah selanjutnya berisi statistik dari laporan sintesis desain (Tabel 10.3).

Dalam konteks desain VLSI neuromorfik, eksekusi model kompleks seperti neuron spiking LIF di Verilog memerlukan evaluasi desain melalui laporan sintesis RTL (Register-Transfer Level) dan CMOS sebagai langkah untuk memastikan kebenaran fungsional dan optimasi kinerja. Selain itu, metode seperti Cadence, Synopsys, dan Yosys sangat membantu dalam menghasilkan laporan sintesis yang dapat menjadi ukuran untuk menilai efisiensi desain

pada tingkat logika dan fisik. Laporan tersebut mencakup berbagai metrik seperti jumlah gerbang, penundaan waktu, penggunaan daya, dan okupansi area, sehingga memungkinkan untuk melihat dengan jelas sejauh mana desain telah sesuai dengan spesifikasi yang dimaksudkan. Dalam proses meninjau laporan sintesis RTL, kita dapat melihat bahwa kode Verilog dikonversi menjadi jumlah gerbang logika yang dioptimalkan secara tepat, sehingga mempertahankan esensi fungsionalitas neuron spiking LIF.



**Gambar 10.16** Inti GDSII.

Selain itu, laporan sintesis tingkat CMOS yang dihasilkan dari Yosys menunjukkan area di mana kinerja paling tertinggal. Ini bisa berupa pelanggaran waktu atau konsumsi daya yang lebih besar dari yang diperlukan, yang keduanya akan memengaruhi efisiensi keseluruhan chip neuromorfik. Karena tata letak GDSII telah dibuat dan statistik desain telah diperiksa, laporan sintesis ini berfungsi untuk mendukung langkah selanjutnya dalam proses desain, seperti langkah penting pembuatan pad ring, yang menjadi tugas selanjutnya. Fase ini menandai peralihan dari verifikasi dan optimasi ke implementasi fisik, yang memastikan bahwa chip siap untuk tape-out akhir dan dapat beroperasi dengan andal dalam kondisi dunia nyata. Untuk mendukung kebutuhan konfirmasi akhir hasil sintesis pada fase desain ini, hal ini sangat penting karena hasil ini menunjukkan perubahan terakhir yang harus dilakukan pada sistem untuk mencapai akurasi komputasi dan efisiensi perangkat keras pada sistem neuromorfik. Statistik tingkat RTL dan CMOS disediakan masing-masing pada Tabel 10.4 dan 10.5.

**Tabel 10.3** Laporan sintesis desain

| **Parameter** | **Pengukuran** |
|---|---|
| Luas total | 100 µm × 100 µm |
| Luas sel (µm²) | 935.898 |
| Luas total (µm²) | 10,000.000 |
| Luas sel standar (µm²) | 935.898 |
| Utilisasi (%) | 14.909 |
| Daya puncak (mW) | 0.120 |
| Daya kebocoran (mW) | 0.000 |
| IRdrop | 0.120 |
| Frekuensi maksimum, Fmax (Hz) | 117.839 M |
| Jumlah pin | 13 |
| Jumlah sel (Cells) | 152 |
| Jumlah register | 17 |
| Jumlah buffer | 3 |

Laporan sintesis RTL yang digambarkan dalam Tabel 10.4 menyampaikan fitur-fitur desain neuron spiking LIF pada tingkat transfer register. Desain ini menggabungkan empat port dengan total 11 bit port, yang sesuai dengan spesifikasi antarmuka yang diharapkan untuk input dan output neuron. Karena tidak ada memori yang terlibat, ini menunjukkan bahwa desain tersebut bergantung pada logika kombinasi dan register tanpa blok RAM tertanam. Jumlah total sel sebanyak 138 sebagian besar terdiri dari gerbang logika dasar seperti AND, NAND, NOR, dan flip-flop. Gerbang AND/NOT menunjukkan bahwa alat sintesis telah mengoptimalkan logika dengan menggunakan primitif gerbang kompleks untuk mengurangi area keseluruhan atau konsumsi daya.

Untuk laporan sintesis tingkat CMOS yang disajikan dalam Tabel 10.5, dapat diamati bahwa pemetaan teknologi dilakukan menggunakan pustaka sel standar sky130_fd_sc_hd. Di sini, jumlah wire dan bit wire meningkat, mencerminkan sumber daya routing dan interkoneksi tambahan yang dibutuhkan pada tingkat fisik. Jumlah total sel berkurang menjadi 74, yang menunjukkan bahwa alat sintesis secara efisien memetakan logika RTL ke dalam sel standar yang lebih kompleks seperti gerbang AOI (*AND-OR-INVERT*), pembalik clock, flip-flop, dan gerbang mayoritas. Pengurangan jumlah sel ini diharapkan karena sintesis fisik bertujuan untuk mengoptimalkan desain dengan menggabungkan gerbang dan meminimalkan jumlah elemen logika diskrit.

**Tabel 10.4** Laporan sintesis RTL

| **Parameter** | **Pengukuran** |
|---|---|
| Jumlah port | 4 |
| Jumlah bit port | 11 |
| Jumlah memori | 0 |

| | |
|---|---|
| Jumlah bit memori | 0 |
| Jumlah proses | 0 |
| Jumlah sel (cell) | 138 |
| $*ANDNOT* | 51 |
| $*AND* | 4 |
| $*DFF_PP0* | 17 |
| $*NAND* | 11 |
| $*NOR* | 9 |
| $*NOT* | 3 |
| $*ORNOT* | 9 |

**Tabel 10.5** Laporan sintesis CMOS

| **Parameter** | **Pengukuran** |
|---|---|
| Jumlah jalur sinyal (wire) | 217 |
| Jumlah bit jalur sinyal | 254 |
| Jumlah jalur sinyal publik | 5 |
| Jumlah bit jalur sinyal publik | 27 |
| Jumlah port | 4 |
| Jumlah bit port | 11 |
| Jumlah memori | 0 |
| Jumlah bit memori | 0 |
| Jumlah proses | 0 |
| Jumlah sel (cell) | 74 |
| sky130_fd_sc_hd__a21oi_1 | 5 |
| sky130_fd_sc_hd__a31oi_1 | 1 |
| sky130_fd_sc_hd__clkinv_1 | 18 |
| sky130_fd_sc_hd__dfrtp_1 | 17 |
| sky130_fd_sc_hd__maj3_1 | 5 |
| sky130_fd_sc_hd__nand2_1 | 4 |
| sky130_fd_sc_hd__nand4_1 | 1 |
| sky130_fd_sc_hd__nor2_1 | 4 |
| sky130_fd_sc_hd__nor3_1 | 1 |
| sky130_fd_sc_hd__nor3b_1 | 2 |

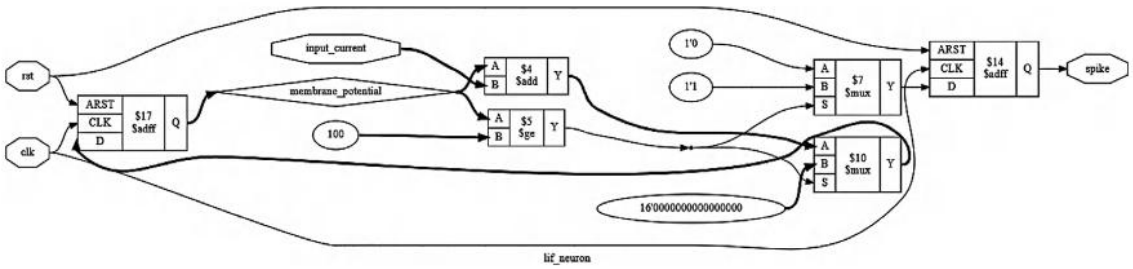
Kesamaan dalam ketiadaan memori di kedua laporan sintesis memverifikasi bahwa desain tersebut sebagian besar terdiri dari bagian kombinasi dengan bagian sekuensial, tetapi tanpa blok penyimpanan tertanam. Keberadaan komponen terkait jam dalam sintesis CMOS, seperti inverter jam dan flip-flop, menggarisbawahi aspek sinkron dari desain, yang sangat penting untuk pengaturan waktu yang akurat dari lonjakan dan pembaruan sinaptik dalam sistem neuromorfik. Penilaian komprehensif ini merupakan prasyarat untuk melanjutkan ke

produksi pad ring, karena menjamin bahwa desain tersebut sesuai dengan persyaratan kinerja, area, dan daya yang sangat penting untuk penerapan perangkat keras neuromorfik yang efektif.

Pengembangan sintesis model neuron lonjakan LIF terdiri dari beberapa tahapan penting, termasuk pembuatan diagram aktivitas yang menunjukkan desain secara visual pada berbagai fase. Mesin Keadaan Terbatas (FSM) RTL, misalnya, menawarkan pandangan dasar tentang logika dan aliran kontrol neuron, seperti yang digambarkan pada Gambar 10.18. Diagram ini menunjukkan status mana yang diaktifkan, bagaimana transisi terjadi, sehingga memfasilitasi verifikasi kebenaran tingkat transfer register dari perilaku neuron. Diagram aliran data berbutir kasar memberikan pandangan yang lebih abstrak tentang desain.

Diagram semacam ini sangat berguna dalam menemukan area peningkatan kinerja atau efisiensi potensial melalui penyorotan proses yang lebih lambat atau hambatan. Saat desain mendekati implementasi fisik yang sebenarnya, skema tingkat gerbang CMOS dikembangkan, yang menunjukkan desain yang sesuai dengan sel standar. Diagram ini menjadi perlu karena merupakan representasi bagaimana desain akan dibuat pada silikon, sehingga memberikan gambaran umum tentang kekuatan implementasi logika pada tingkat transistor. Ini memastikan bahwa desain berfungsi secara fisik, tidak hanya secara logis, dan dapat menunjukkan kemungkinan kesulitan dengan area, daya, dan waktu.

Diagram dan laporan ini mewakili tonggak penting dalam proses desain. Diagram ini memungkinkan verifikasi fungsionalitas desain di setiap tahap, dari logika hingga implementasi fisik, dan juga menunjukkan bagian-bagian yang perlu ditingkatkan. Selain itu, mereka memfasilitasi komunikasi dan diskusi di antara tim desain, sehingga desain akan lebih berkembang sebelum beralih ke proses akhir, seperti pembuatan pad ring dan pembuatan tata letak fisik. Diagram sintesis RTL FSM dan CMOS gate-level ditunjukkan pada Gambar 10.17 dan 10.18, masing-masing.



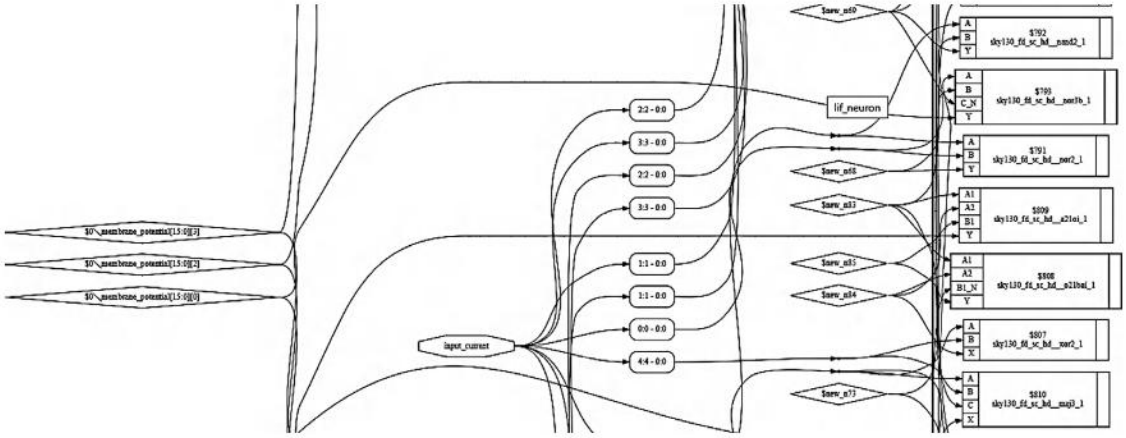

**Gambar 10.17** Mesin Keadaan Terbatas tingkat RTL dan diagram aktivitas.

**Bingkai Pad Input/Output**

Pada bagian ini, Bagian 10.2 akan menguraikan langkah-langkah desain yang diperlukan untuk mengembangkan bingkai pad I/O yang dapat memastikan integritas sinyal. Setelah logika inti dari chip neuromorfik disintesis ke dalam GDSII, mungkin tampak bahwa desain sudah lengkap. Namun, tanpa bingkai pad input/output (IO) yang direkayasa dengan benar, sirkuit tidak dapat diproduksi atau dioperasikan dalam silikon nyata. Bingkai pad

mendefinisikan antarmuka antara arsitektur neuromorfik on-chip dan dunia luar, memastikan bahwa sinyal, daya, dan ground dapat dikirimkan dengan andal dan kuat. Ini menyediakan struktur fisik di sekitar inti yang disintesis, mengintegrasikan bantalan I/O, bantalan sudut, dan bantalan khusus untuk daya dan ground.

Dalam sistem neuromorfik hemat energi, hal ini menjadi sangat penting karena SNN memerlukan konektivitas fan-in/fan-out yang tinggi serta front-end analog yang sensitif. Rangka pad yang dirancang dengan buruk dapat menimbulkan masalah integritas sinyal, kerentanan ESD, atau ketidakstabilan daya, yang semuanya akan menghambat keunggulan arsitektural yang dicapai pada tingkat algoritmik dan sirkuit. Oleh karena itu, rangka bantalan merupakan lapisan dasar desain fisik yang harus disertakan bersama inti GDSII yang disintesis agar desain dapat berfungsi.




**Gambar 10.18** Diagram aktivitas snapshot CMOS untuk model neuron LIF.

**Pertimbangan Pelepasan Muatan Listrik Statis (ESD)**

Salah satu peran utama bantalan I/O adalah memberikan perlindungan ESD. Setiap pin yang terbuka pada sebuah chip rentan terhadap peristiwa elektrostatik yang dapat secara instan menghasilkan tegangan beberapa kilovolt. Rangka bantalan mengintegrasikan perangkat perlindungan yang mencakup dioda dan klem yang mengalihkan arus berlebih dengan aman ke listrik atau rel ground. Model sederhana untuk perlindungan ESD dapat ditulis sebagai:

$$I_{ESD} = \frac{V_{event}}{R_{damp}}$$

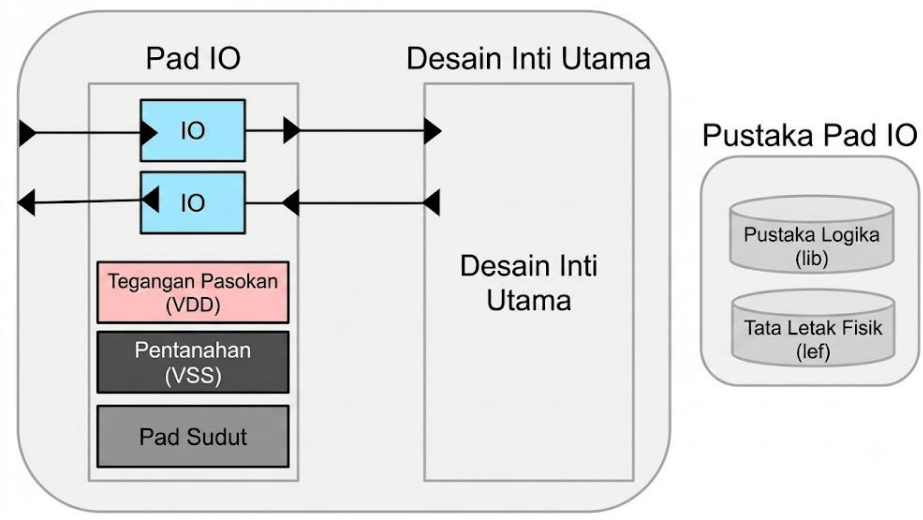
di mana Vevent adalah tegangan elektrostatik yang diterapkan dan Rclamp adalah resistansi ekivalen dari jalur proteksi. Pertimbangan desainnya adalah menjaga Rclamp cukup rendah untuk mengalihkan arus transien besar tanpa memengaruhi perilaku I/O normal. Dalam sistem neuromorfik di mana sirkuit sinaptik dan neuronal bergantung pada presisi sinyal kecil, penjepit ESD yang kuat mencegah kegagalan fatal selama pengemasan dan pengujian. Untuk

desain neuron LIF spesifik ini, sebuah padring dikembangkan dengan mempertimbangkan aspek-aspek berikut:

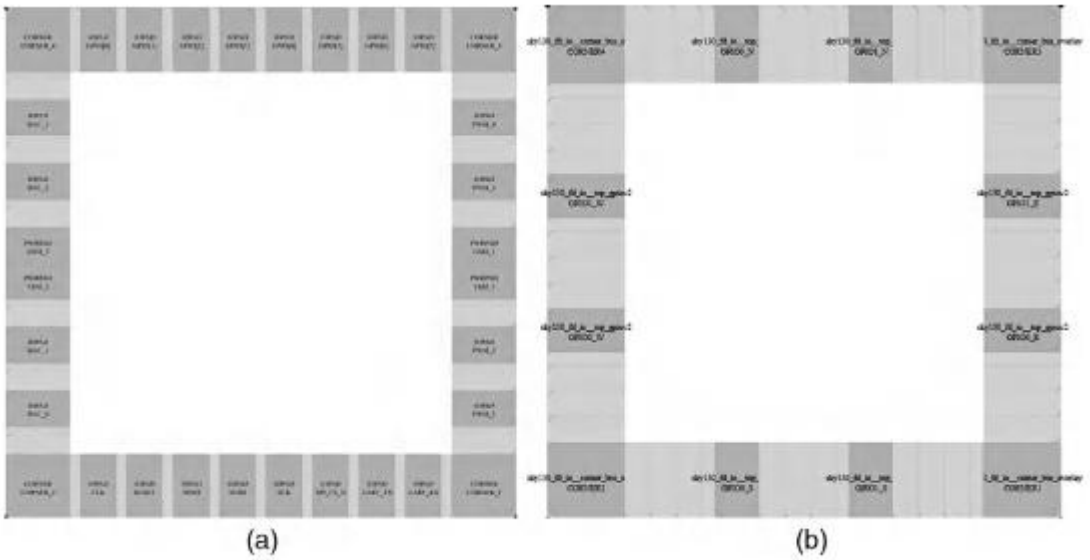
- **Sebuah padframe tingkat atas** yang menghubungkan pin chip eksternal ke sinyal internal melalui sel pad I/O yang diinstansiasi. Bingkai ini juga harus menggabungkan pad daya khusus ($V_{DD}$ dan $V_{SS}$) serta pad sudut untuk memastikan penutupan listrik dan mekanis.
- **Spesifikasi I/O tingkat chip** yang mendefinisikan urutan spasial dan pengaturan fisik sel pad di sekitar tepi die.

Gambaran umum dari padframe tersebut ditunjukkan pada Gambar 10.19. Sebelum mengembangkan padring, beberapa desain dianalisis dengan pad dummy untuk memvisualisasikan padframe seperti yang ditunjukkan pada Gambar 10.20a dan b.

Alur silikon open-source modern memungkinkan pembuatan padring I/O dummy semi-otomatis, termasuk sel sudut dan pad khusus. Padring ini sangat penting untuk memastikan penutupan mekanis batas chip dan membangun pengiriman daya/ground yang konsisten di sekitar tepi. Padring yang dihasilkan menegakkan simetri listrik di keempat tepi chip dan menyediakan kerangka kerja untuk memasukkan pad analog atau sinyal campuran tambahan. Mahasiswa dan peneliti dapat menghasilkan padring uji otomatis dengan alat seperti Cadence dan Synopsys; namun, dalam produksi, ini dioptimalkan dengan cermat untuk impedansi, jarak antar pad, dan kompatibilitas paket.




**Gambar 10.19** Antarmuka desain tingkat atas dengan pad daya, GND, dan I/O khusus untuk pustaka pad I/O yang disediakan oleh PDK.

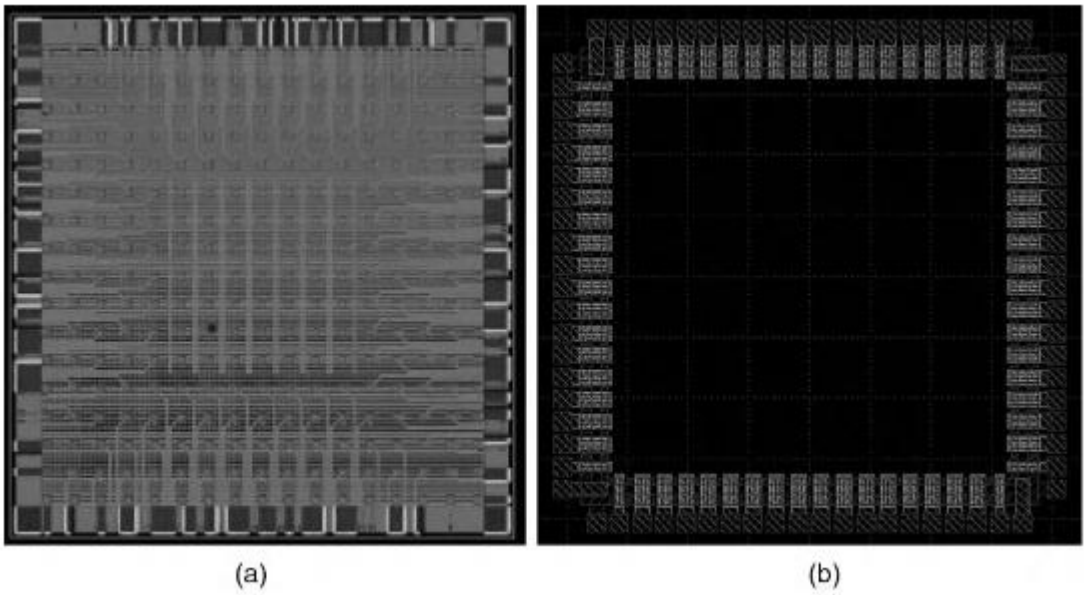



**Gambar 10.20** (a) dan (b) Evaluasi desain yang berbeda dengan padring dummy.

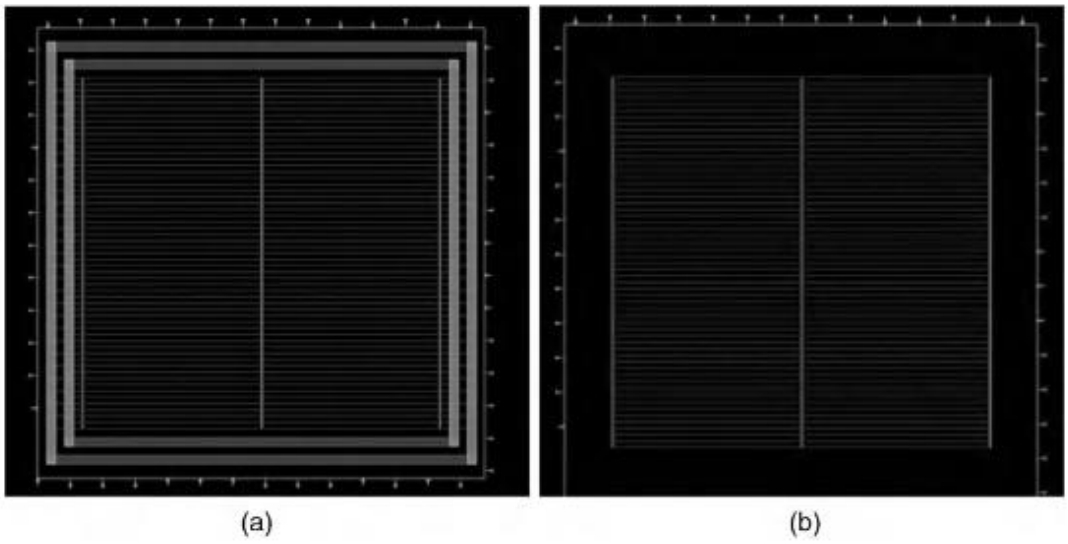
Cincin bantalan dengan alat-alat seperti Cadence dan Synopsys; namun, dalam produksi, ini dioptimalkan secara cermat untuk impedansi, jarak bantalan, dan kompatibilitas paket. Integritas daya adalah fungsi kritis kedua dari bingkai bantalan. Inti neuromorfik mungkin memiliki aktivitas peralihan yang digerakkan oleh peristiwa dan jam yang direkayasa secara cermat, tetapi tanpa pasokan impedansi rendah, aktivitas tersebut akan runtuh menjadi operasi yang tidak stabil. Lebar kawat yang diperlukan untuk penyaluran daya dapat dihitung menggunakan hukum Ohm sebagai $\Delta V = I_{load} \times R_{wire}$, di mana $\Delta V$ adalah penurunan tegangan di sepanjang rel pasokan, $I_{load}$ adalah arus puncak yang ditarik oleh inti neuromorfik, dan $R_{wire}$ adalah resistansi dari jejak pasokan. Untuk menjaga stabilitas tegangan, sangat penting untuk menahan penurunan tegangan ini dalam rentang 5%-10% dari tegangan pasokan nominal $V_{DD}$. Ini memberlakukan batasan: $\Delta V \leq \alpha \times V_{DD}$, di mana $\alpha$ adalah nilai antara 0,05 dan 0,10. Mensubstitusikan hukum Ohm ke dalam pertidaksamaan ini menghasilkan: $I_{load} \times R_{wire} \leq \alpha \times V_{DD}$, yang dapat disusun ulang untuk menentukan resistansi kawat maksimum yang diizinkan: $R_{wire} \leq (\alpha \times V_{DD})/I_{load}$. Resistansi kawat juga dapat ditentukan dari dimensi fisiknya dengan rumus: $R_{wire} = \rho \times \big(L/(W \times t)\big)$, di mana $\rho$ adalah resistivitas logam interkoneksi, $L$ adalah panjang jalur penyaluran daya, $W$ adalah lebar kawat, dan $t$ adalah ketebalan logam.

Rumus-rumus ini menjelaskan peran bentuk kawat dalam resistansi dan penurunan tegangan, yang karenanya menuntut penerapan cincin daya dalam cincin bantalan cip. Cincin-cincin ini bertindak sebagai penghalang ke jalur pasokan dan mencegah area di sekitarnya mengalami penurunan tegangan. Dalam hal desain cip, dapat dikatakan bahwa ada dua metode utama untuk membuat koneksi antara cip dan komponen luar. Yang paling umum adalah ikatan kawat, yaitu menghubungkan bantalan kontak pada cip dengan yang ada di paket melalui kawat logam tipis. Dengan metode ini, bantalan ditempatkan di tepi luar cip. Metode lain yaitu ikatan cip terbalik, adalah metode yang lebih modern, dan dapat memiliki kepadatan interkoneksi yang bahkan lebih tinggi. Dalam pendekatan ini, cip dipasang menghadap ke

bawah ke atas substrat paket menggunakan susunan tonjolan konduktif kecil. Ini memungkinkan penempatan bantalan koneksi di seluruh permukaan cip daripada membatasinya hanya di bagian tepi. Gambar 10.21 menunjukkan bingkai bantalan khas yang dikonfigurasi untuk baik ikatan kawat Gambar 10.21b maupun cip terbalik Gambar 10.21a. Daerah tengah cip berisi logika inti bersama dengan jaringan distribusi daya, yang dikelilingi oleh cincin sel bantalan input-output dan sel sudut. Fungsi utama sel bantalan adalah untuk memperkuat sinyal arus rendah yang dihasilkan di dalam cip sehingga dapat ditransmisikan secara efektif ke dunia luar. Setiap sel bantalan juga menyediakan lokasi fisik untuk menempelkan kawat dalam desain ikatan kawat, atau tonjolan solder dalam konfigurasi cip terbalik.




**Gambar 10.21** Gambaran umum kerangka pad I/O untuk integrasi inti (a) flip-chip (b) wire bonding.




**Gambar 10.22** (a) Inti dengan cincin (b) inti tanpa cincin.

Selain penguatan sinyal, sel pad melakukan pergeseran level tegangan ketika level tegangan yang digunakan untuk input dan output eksternal berbeda dari tegangan inti internal, dan penggeser level tersebut harus diintegrasikan dengan pad frame I/O. Mereka juga menyertakan perlindungan pelepasan elektrostatik, yang biasanya diimplementasikan menggunakan dioda bias terbalik antara pad dan rel ground ($V_{SS}$) dan daya ($V_{DD}$), untuk melindungi sirkuit internal dari lonjakan tegangan.

Sel sudut disertakan pada persimpangan cincin pad untuk memastikan bahwa jaringan distribusi daya kontinu dan tertutup dengan benar. Beberapa pad daya, baik $V_{DD}$ maupun $V_{SS}$, didistribusikan di sekitar chip. Koneksi daya redundan ini membantu memastikan bahwa daya didistribusikan secara merata di seluruh die dan bahwa semua wilayah mempertahankan tegangan suplai yang stabil, seperti yang ditunjukkan pada Gambar 10.22a dan b.

## 10.4 INTEGRITAS I/O CHIP NEUROMORFIK

Bab ini membahas bagian penting namun sering diabaikan, yaitu integritas pada I/O dalam konteks perancangan chip neuromorfik hemat energi. Dalam bidang komputasi neuromorfik, fokus biasanya tertuju pada inovasi algoritma dan logika pusat berdaya sangat rendah, sementara pembahasan ini menempatkan antarmuka I/O sebagai faktor utama yang memengaruhi kinerja sistem secara keseluruhan dan efisiensi daya. Bab ini, dengan menempatkan desain pad I/O sebagai faktor taktis dalam pengembangan sistem VLSI neuromorfik, menyelidiki pengaruh keandalan sinyal, integritas daya, dan teknik pengikatan pada fungsionalitas perangkat keras secara keseluruhan. Ide dasar SNN dan alur desain VLSI digital disajikan untuk memberikan kerangka kerja. Pengetahuan praktis diperoleh dari keterlibatan dengan proses CMOS SkyWater 130 nm.

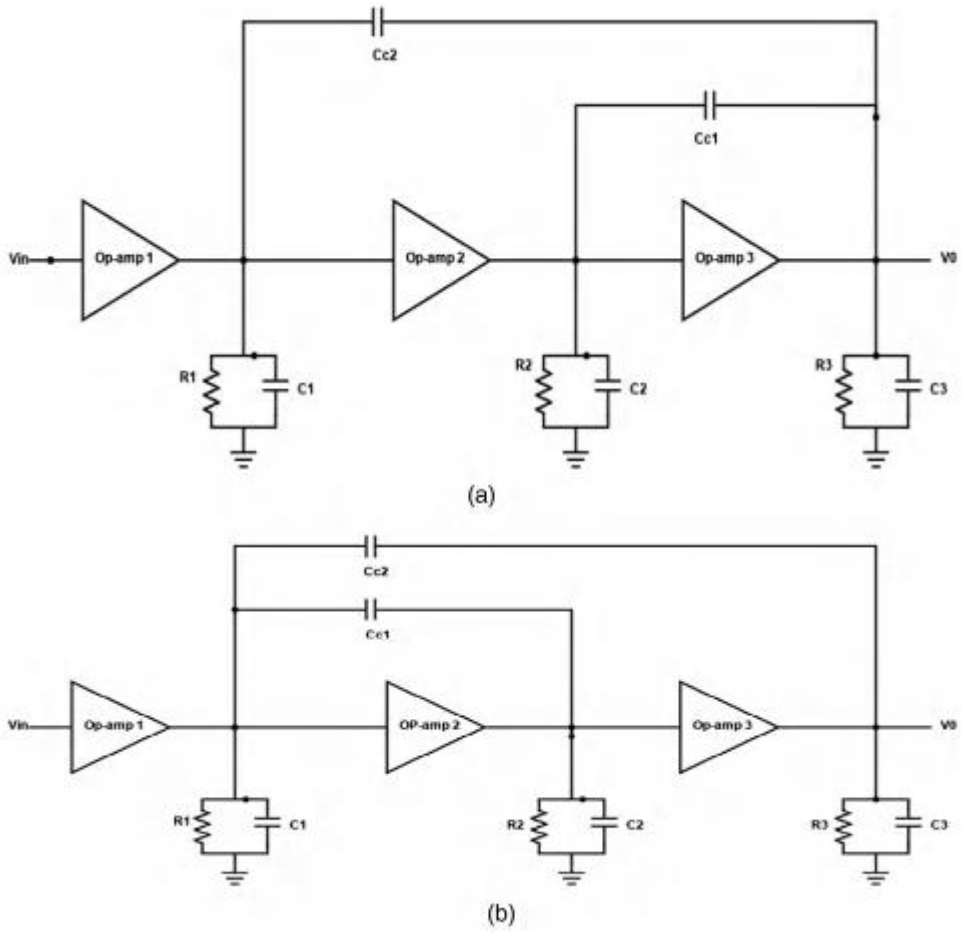
Topik utama meliputi pemilihan pustaka pad I/O, arsitektur cincin pad, perlindungan ESD, dan metode pengikatan seperti pengikatan kawat dan flip-chip. Melalui analisis dan contoh praktis, bab ini menunjukkan bagaimana perencanaan I/O tahap awal dapat secara signifikan menurunkan risiko desain, mencegah desain ulang yang mahal, dan mendorong pengembangan sistem neuromorfik yang tangguh. Konsep dasar SNN dan alur desain VLSI digital diperkenalkan untuk mengkontekstualisasikan diskusi. Implementasi praktis model SNN ditunjukkan dengan proses CMOS SkyWater 130 nm. Melalui analisis teoretis dan contoh praktis, bab ini menunjukkan bagaimana perencanaan I/O tahap awal dapat secara signifikan mengurangi risiko desain, menghindari siklus desain ulang yang mahal, dan mendukung pengembangan sistem neuromorfik yang tangguh.

# BAB 11
# OP-AMP 3-TAHAP RNMC DAYA RENDAH UNTUK BIOMEDIS 45 NM

## 11.1 OPTIMASI AI/ML PADA DESAIN OP-AMP 3-TAHAP RNMC DAYA RENDAH 45 NM

Penguat operasional (*Op-amp*) adalah komponen umum dalam sirkuit mode campuran dan analog. Penguatan intrinsik MOSFET berkurang karena peningkatan teknologi CMOS dan pengurangan ukuran komponen, sedangkan tren industri mengarah ke frekuensi dan resolusi yang lebih tinggi. Oleh karena itu, kebutuhan akan OTA dengan penguatan tinggi dan kecepatan tinggi tidak dapat dihindari. Akibatnya, pembahasan tentang masalah desain dengan penguat multi-tahap masih berlangsung. Untuk mengatasi masalah ini, dua strategi utama disarankan. Penguat operasional dengan penguatan tinggi dan kecepatan tinggi dimaksudkan untuk diimplementasikan menggunakan pengaturan kaskade dan kaskade. Pendekatan kaskade cocok untuk tegangan suplai tinggi karena transistor kaskade membatasi ayunan keluaran.

Namun, penguat operasional kaskade dan multi-tahap berfungsi lebih baik dalam aplikasi tegangan rendah. Karena banyaknya kutub dan nol, pendekatan ini membutuhkan disipasi daya tambahan dan harus disesuaikan. Untuk mencegah masalah stabilitas dan mencapai margin fase yang memadai, penekanan utama dalam desain penguat multistage adalah pada penerapan kompensasi frekuensi yang tepat. Pada penguat multistage, skema kompensasi frekuensi yang rumit diperlukan untuk kriteria stabilitas loop tertutup. Kompensasi Miller bersarang (NMC) dan kompensasi Miller bersarang terbalik (RNMC), yang diwakili pada Gambar 11.1 a dan b, dibahas dalam metode konvensional untuk kompensasi frekuensi penguat tiga tahap. Teknik kompensasi NMC dan berbagai variasinya membuat penguat menjadi lambat dan konsumsi daya tinggi, dan biasanya, ada persyaratan untuk kapasitor kompensasi yang sebanding dengan beban, sehingga struktur tersebut tidak cocok untuk aplikasi yang dimaksud. Baik kapasitor beban maupun kapasitor kompensasi membebani node keluaran dalam struktur NMC, menghasilkan kutub yang lebih dominan sebagai hasilnya. Mengenai GBW, masalah ini mengurangi respons frekuensi. Metode kompensasi populer lainnya untuk penguat multi-tahap dikenal sebagai RNMC. Dengan menggunakan kapasitansi Miller, kutub non-dominan dapat digeser lebih jauh dari frekuensi penguatan satuan, sehingga secara efektif meningkatkan pemisahannya dari kutub dominan.




**Gambar 11.1** (a) Skema kompensasi Miller bersarang. (b) Skema kompensasi Miller bersarang terbalik.

Meskipun RNMC memiliki stabilitas yang sangat baik, mereka menggunakan lebih banyak daya dan memiliki bandwidth yang sangat kecil. RNMC pada dasarnya dikembangkan untuk mengatasi masalah kapasitor beban keluaran yang mengganggu pendekatan Miller bersarang. Dengan menggunakan kapasitansi Miller, kutub non-dominan dapat digeser lebih jauh dari frekuensi gain satuan, secara efektif meningkatkan pemisahannya dari kutub dominan.

Selain itu, seiring dengan pengecilan transistor oleh teknologi CMOS, kepadatan area chip meningkat secara eksponensial, yang menyebabkan arus bocor yang lebih tinggi, efek parasit, dan kesalahan yang disebabkan oleh variasi parameter karena panjang saluran yang lebih pendek, yang membuat teknik coba-coba konvensional menjadi tidak efisien. Kecerdasan Buatan (AI) dan Pembelajaran Mesin (ML) telah dikenal sebagai solusi kunci untuk eksekusi otomatis desain analog dan teknik optimasi dengan tujuan untuk mengatasi kendala tersebut.

Untuk meningkatkan gain-bandwidth (GBW), mengurangi konsumsi daya, dan meningkatkan stabilitas fasa, metode seperti Algoritma Genetika (GA), Optimasi Swarm Partikel (PSO), dan Simulated Annealing (SA) digunakan untuk memodifikasi dimensi transistor, arus bias, dan nilai kapasitor kompensasi ($C_{C1}$ dan $C_{C2}$ ). Regresi proses Gaussian (GPR) mengurangi jumlah iterasi simulasi dengan bantuan optimasi Bayesian (BO). BO juga memaksimalkan ruang desain. BO menentukan nilai optimal transkonduktansi

($gm_1, gm_2, \text{dan } gm_6$) dan nilai kapasitor pada Op-amp berbasis RNMC yang memastikan stabilitas fasa di bawah kondisi beban yang berbeda dengan menyediakan agen untuk berinteraksi dengan pengaturan sirkuit yang mengukur parameter Op-amp seperti daya rendah atau gain tinggi dan mempelajari metodologi bias dan ukuran yang paling efektif, di mana teknik pembelajaran penguatan (RL) memberikan optimasi adaptif. Pemodelan pengganti berbasis pembelajaran mendalam meningkatkan teknik ini dengan mengukur respons frekuensi atau transien dengan memungkinkan siklus optimasi yang lebih cepat dan teknik otomatisasi berbasis data.

Memanfaatkan kemajuan terkini, platform Otomatisasi Desain Elektronik (EDA) kontemporer kini menggabungkan metodologi AI/ML untuk mengembangkan ekosistem desain yang cerdas, adaptif, dan berpusat pada data. Modul untuk mendesain DSE eksplorasi ruang angkasa yang menggunakan ML dalam dua jaringan saraf sederhana ini, GPR dan Support Vector Machine (SVM), untuk memprediksi matriks kinerja sirkuit seperti gain, bandwidth, dan margin fase, menghilangkan kebutuhan akan simulasi yang menyeluruh. Mesin optimasi yang memanfaatkan AI dalam alat EDA menggunakan algoritma GA, BO, dan RL untuk secara otomatis menyesuaikan parameter sirkuit, mengurangi waktu desain dan meningkatkan akurasi.

Dalam kerangka Op-amp RNMC Tiga Tahap Daya Rendah, modul tersebut mampu secara otomatis mengoptimalkan transkonduktansi, kapasitor kompensasi, dan arus bias untuk memastikan kinerja tinggi yang konsisten di tengah variasi proses. Selain optimasi desain, AI/ML secara substansial meningkatkan proses verifikasi dan analisis pasca-layout. Model pembelajaran mendalam yang terdiri dari Jaringan Saraf Konvolusional (CNN) secara efektif menunjukkan risiko elektromigrasi hotspot layout dan wilayah penurunan IR pada tahap awal alur kerja.

Selain itu, pengklasifikasi berbasis pembelajaran mesin berkontribusi pada prediksi parasit dan deteksi kesalahan. Asisten desain bertenaga AI memanfaatkan Pemrosesan Bahasa Alami (NLP) dan RL untuk meningkatkan proses desain dengan menyarankan rentang parameter, memfasilitasi eksplorasi desain, dan mengotomatiskan pengambilan keputusan arsitektur. Lebih lanjut, sistem EDA berbasis cloud memanfaatkan AI/ML untuk mengelola simulasi terdistribusi dan mengoptimalkan alokasi sumber daya, yang mengarah pada konvergensi yang lebih cepat dan peningkatan efisiensi komputasi.

Artikel ini disusun sebagai berikut, merinci desain penguat operasional tiga tahap yang diimplementasikan dalam teknologi 45 nm. Bagian 11.2 menyajikan metodologi desain penguat operasional, Bagian 11.3 menampilkan hasil eksperimen untuk rangkaian Op-amp tiga tahap, dan akhirnya, Bagian 11.4 memberikan kesimpulan.

## 11.2 PERANCANGAN OTA TIGA TAHAP DENGAN KOMPENSASI RNMC

Langkah awal dalam mengembangkan jaringan kompensasi melibatkan analisis fungsi transfer loop terbuka dari topologi yang diusulkan. Analisis ini dilakukan dengan mempelajari rangkaian sinyal kecil ekivalen, di mana parameter $G_{mi}, R_{oi}, \text{dan } C_{oi}$ masing-masing mewakili transkonduktansi, resistansi, dan kapasitansi keluaran ekivalen tahap ke-i, sedangkan $C_L$

adalah kapasitansi beban. Fungsi transfer akan dilakukan dengan cara berikut, dengan asumsi bahwa $C_L, C_{C1}$, dan $C_{C2}$ jauh lebih besar daripada $C_{oi}$ dan penguatan DC dari setiap tahap, $A_{vi} = G_{mi}\, R_{oi}$, jauh lebih besar daripada satu untuk mempertahankan akurasi sambil menyederhanakan ekspresi. Sebagai akibat dari kapasitansi parasit, kita akan mengabaikan kutub frekuensi tinggi.

Loop umpan balik kapasitif menstabilkan penguat dalam RNMC dengan memberikan umpan balik negatif. Namun, umpan balik kapasitif menunjukkan beban internal di dalam penguat. Arus umpan maju melalui kapasitor kompensasi ini tidak sefase dengan arus yang masuk ke kapasitor beban, sehingga menghasilkan efek umpan maju yang menghasilkan nol bidang setengah kanan (RHP). Menanggapi temuan ini, kami menyertakan resistor penyeimbang. Resistor penyeimbang memblokir jalur umpan maju. Akibatnya, resistor ini dapat mencegah kapasitor kompensasi menurunkan impedansi keluaran tahap pertama dengan cepat. Akibatnya, resistor penyeimbang mengurangi penguatan tahap pertama pada frekuensi yang lebih tinggi, yang pada gilirannya meningkatkan margin fase.

Meskipun dalam artikel khusus ini struktur RNMC yang diadopsi secara konseptual sama dengan yang disarankan dalam dengan fungsi transfer loop terbuka dan topologi penguat yang serupa, terdapat perbedaan teknologi, dan diimplementasikan dengan menetapkan nilai resistor penyeimbang pada RC = 0. Struktur RNMC ini dirancang untuk tegangan suplai rendah, konsumsi daya lebih rendah, dan ayunan tegangan maksimum yang diizinkan.

$$\omega_{p1} \cong \frac{1}{C_{c1} r_{01} g_{m2} r_{02} g_{m3} r_{03}} \tag{11.1}$$

$$A_0 = -g_{m1} r_{01} g_{m2} r_{02} g_{m3} r_{03} \tag{11.2}$$

di mana $A_0$ adalah penguatan tegangan DC
Oleh karena itu, hasil kali penguatan dan bandwidth adalah, seperti biasa

$$\omega_{GWB} = |A_0|\, \omega_{wp1} = \frac{g_{m2}}{C_{c1}} \tag{11.3}$$

$$g_{mvf} = g_{m8} \tag{11.4}$$

Penurunan tahap keluaran push-pull simetris menghasilkan bentuk sederhana Avs seperti yang disajikan dalam dengan mengecualikan efek resistor nol RC.

$$\mathrm{A}A_{VZ} = \frac{A_0}{\left(1+\frac{S}{\omega_{pl}}\right)} \frac{1+S\left[\frac{C_{c2}}{g_{m6}}\left(\frac{g_{mvf}}{g_{m8}}-1\right)+\frac{C_{c1}}{g_{m6}\, g_{m8}\, R_{02}}\right]-S^2\frac{C_{c3}C_{c2}}{g_{m6}\, g_{m8}}}{1+S\left[\frac{C_L\, C_{c2}}{g_{m8}\, C_{Cc3}}+\frac{C_{c2}}{g_{m8}}+\frac{C_{c2}}{g_{m6}}\frac{(g_{mvf}-1)}{g_{m8}}\right]+S^2\frac{C_{c2}C_L}{g_{m6}g_{m8}}} \tag{11.5a}$$

Sekarang substitusikan (11.4) ke dalam (11.5a)

$$A_{VS} = \frac{A_0}{(1+\frac{s}{\omega_{p1})}} \frac{1 + S\frac{C_{c3}}{g_{m6}g_{m8}R_{02}} - S^2\frac{C_{c3}C_{c2}}{g_{m6}g_{m8}}}{1 + S\frac{C_L C_{c2}}{g_{m8}C_{c3}} + S^2\frac{C_{c2}C_L}{g_{m6}g_{m8}}} \quad (11.5b)$$

Proses desain dimulai dengan menentukan nilai transkonduktansi tahap pertama, $gm_1$ dan $gm_2$ , yang dipilih berdasarkan kinerja noise yang dibutuhkan atau deviasi standar yang terkait dengan offset acak yang disebabkan oleh ketidaksesuaian perangkat. Setelah parameter ini ditentukan, kapasitor kompensasi $C_{C1}$ dipilih untuk mencapai produk gain-bandwidth yang diinginkan. Kapasitor kompensasi kedua, $C_{C2}$, kemudian disesuaikan untuk mencegah puncak pada respons frekuensi. Aproksimasi paling kanan berlaku dalam kondisi $C_L \gg C_{C1}$.

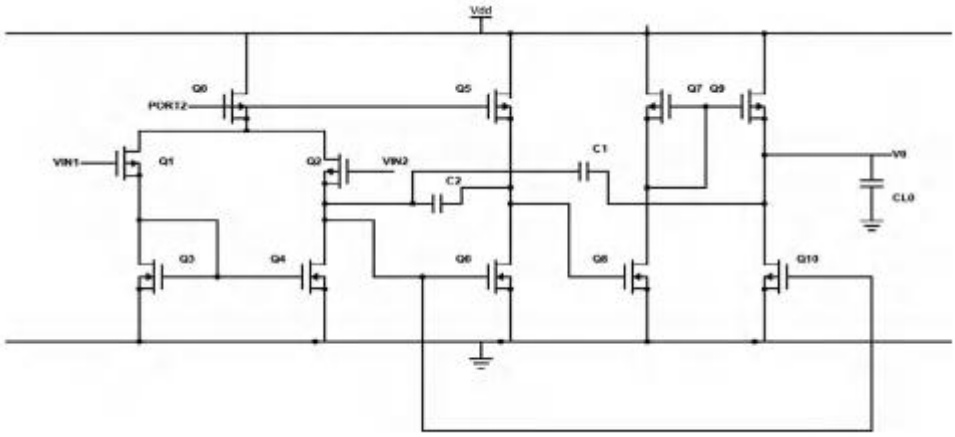
Penting untuk dicatat bahwa, tidak seperti pendekatan yang dijelaskan dalam, di mana $C_{C2}$ berbanding lurus dengan kapasitansi beban $C_L$, dalam desain ini $C_{C2}$ mencapai nilai yang lebih kecil bahkan dengan beban kapasitif yang besar. Karakteristik ini menyebabkan pengurangan yang signifikan pada luas silikon. Selain itu, berbeda dengan arsitektur NMC konvensional, di mana peningkatan gm6 meningkatkan kinerja amplifier, transkonduktansi tahap akhir dipilih hanya untuk mencapai nilai praktis CC2. Untuk kesederhanaan dan efisiensi implementasi, salah satu konfigurasi yang layak yang diusulkan oleh diadopsi di sini.

$$\emptyset = \tan^{-1}\frac{g_{nm2}C_{c1}^2 - g_{nm1}^2 C_L C_{c2}}{g_{nm1}g_{nm2}C_L C_{c2}} \quad (11.6)$$

$$\text{dimana } g_{nm1} = \frac{g_{m2}}{g_{m8}} \text{ dan } g_{nm2} = \frac{g_{m6}}{g_{m8}} \quad (11.7)$$

Transkonduktansi tahap amplifikasi kedua harus lebih besar daripada tahap pertama untuk menjamin stabilitas asimtotik sistem. Perlu juga disebutkan bahwa, tidak seperti pada arsitektur NMC konvensional, parameter $g_{m8}$ dalam konfigurasi ini tidak dibatasi pada nilai tetap, meskipun umumnya lebih disukai jika gm8 lebih besar, sistem masih dapat berfungsi dengan benar bahkan ketika nilainya lebih rendah daripada $g_{m2}$.
Selanjutnya,

$$FOM_s = \frac{\omega_{GBW}C_L}{power} \quad (11.7)$$




**Gambar 11.2** Penguat operasional tiga tahap dengan kompensasi frekuensi bersarang terbalik.

$$IFOM_s = \frac{\omega_{GBW} C_L}{I_{DD}} \tag{11.8}$$

Skema sederhana pada Gambar 11.2 dari OP-amp tiga tahap direalisasikan dan diimplementasikan dalam artikel ini menggunakan lingkungan Cadence Virtuoso 45 nm. Artikel ini merupakan adaptasi dari. OTA tiga tahap disimulasikan di bawah lingkungan Cadence Virtuoso dengan menetapkan level output pada VDD/2 sehingga memberikan ayunan tegangan maksimum yang diizinkan, dengan melakukan hal tersebut, kita mendapatkan respons keseluruhan yang lebih baik, untuk kompensasi frekuensi digunakan metodologi RNMC dengan menggunakan dua kapasitor kompensasi, yaitu $C_{C1}$ dan $C_{C2}, C_L \gg C_{C2}$ dan sebanding dengan $C_{C1}$.

Tahap pertama adalah pasangan diferensial PMOS $Q_{1-2}$ dan beban cermin arus $Q_{3-4}$. Tahap kedua adalah tahap pembalik, yang direalisasikan oleh $Q_{5-6}$ sedangkan tahap pembalik terakhir terdiri dari $Q_{7-10}$. Tahap umpan-maju $g_{mvf}$ diimplementasikan dengan memanfaatkan transistor beban aktif $Q_{10}$ pada tahap akhir, yang gerbangnya dikendalikan oleh keluaran dari tahap pertama. Dengan konfigurasi ini, transistor $Q_9$ dan $Q_{10}$ berfungsi sebagai tahap keluaran *kelas-AB semu*, yang secara efektif menyediakan kemampuan untuk menggerakkan kapasitor beban $C_L$.

Set parameter untuk Op-Amp tipe-p dalam lingkungan 45 nm untuk $V_{DD}$ sebesar 500 mV dan $I_D$ sebesar 200 nA untuk $g_{m1} = g_{m2} = 2.77\ \mu$A/V dan $C_{C1} = 1.5$ pF. Kemudian, untuk mendapatkan *phase margin* sekitar 60° kita menetapkan $g_{m6} = 13.86\ \mu$A/V. Kita menetapkan $g_{m6} = 13.86\ \mu$A/V, $g_{m8} = 15.48\ \mu$A/V, $g_{m10} = 14.3\ \mu$A/V. Kapasitor kompensasi $C_{C2}$ awalnya diatur ke 0.4 pF. Melalui simulasi berbantuan komputer, nilai kapasitor kompensasi $C_{C1}$ dan $C_{C2}$ disetel halus menjadi 1.5 dan 0.4 pF, secara berurutan. Arus drain dari transistor $Q_1$ dan $Q_2$ dikonfigurasikan ke 100 nA, sedangkan arus untuk tahap kedua dan ketiga disetel ke 400 nA, menghasilkan total konsumsi arus sebesar 1.4 $\mu$A. Dimensi transistor dirangkum dalam tabel yang sesuai. Respons frekuensi *open-loop* yang disimulasikan dari penguat diilustrasikan

pada gambar, menunjukkan penguatan DC sebesar 77.9 dB, produk *gain-bandwidth* sebesar 610 kHz, dan *phase margin* sebesar $61.56^{\circ}$.

## 11.3 HASIL SIMULASI DAN ANALISIS KINERJA OTA 45 NM

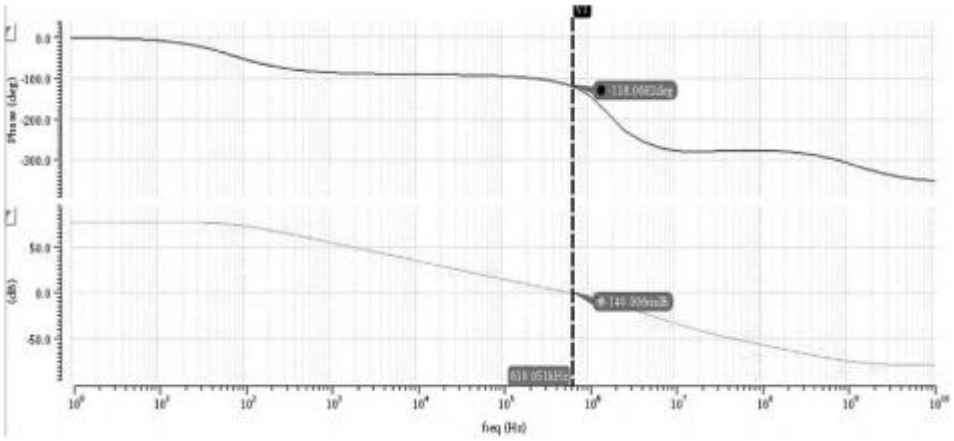
Transkonduktansi semua transistor untuk kedua teknologi disebutkan dalam Tabel 11.1. Semua hasil eksperimen yang dilakukan di lingkungan Cadence Virtuoso 45 nm untuk kedua teknologi disebutkan dalam Tabel 11.2, yang menunjukkan parameter kinerja yang berbeda untuk kedua teknologi tersebut. Untuk nilai tegangan suplai yang lebih rendah, yaitu 500 mV, diperoleh 77,9 dB, di mana kapasitor kompensasi (CC) 2 dan 2,3 pF digunakan untuk mencapai margin fasa 61,56° untuk kedua teknologi, yang merupakan margin fasa yang kami butuhkan untuk nilai kapasitor kompensasi yang lebih rendah ini. Nilai kapasitor kompensasi selanjutnya dikurangi dengan meningkatkan arus tahap kedua dan ketiga, tetapi rangkaian akan mengkonsumsi daya lebih banyak, sehingga ada kompromi antara kapasitor kompensasi (margin fasa) dan konsumsi daya. Konsumsi daya adalah 857,62 nW.

**Tabel 11.1** Transkonduktansi transistor

| TRANSKONDUKTANSI | 45 NM |
|---|---|
| $Gm_0$ | 5.33 |
| $Gm_1$, $Gm_2$ | 2.77 |
| $Gm_3$, $Gm_4$ | 2.89 |
| $Gm_5$ | 12.89 |
| $Gm_6$ | 13.86 |
| $Gm_7$, $Gm_9$ | 14.27 |
| $Gm_8$ | 15.48 |
| $Gm_{10}$ | 14.3 |

**Tabel 11.2** Parameter kinerja

| Parameter | Teknologi 45 Nm |
|---|---|
| Teknologi | 45 Nm |
| Tegangan Catu Daya | 500 Mv |
| Penguatan (Gain) | 77.9 Db |
| Ugb (Unity Gain Bandwidth) | 0.610 Mhz |
| Margin Fasa | 61.56° |
| Kapasitor Kompensasi | 2 Pf |
| Arus | 200 Na |
| Disipasi Daya | 857.621 Nw |
| Cmrr | 56 Db |
| Psrr | 69.24 Db |
| Kesalahan Tegangan Offset | 17.86 Mv |
| Derau Masukan (*Input Noise*) | 1.005 µv/√Hz |
| Fom (*Figure Of Merit*) | 1422 Mhz·Pf/µw |




**Gambar 11.3** Analisis respons frekuensi dari Op-amp tiga tahap.

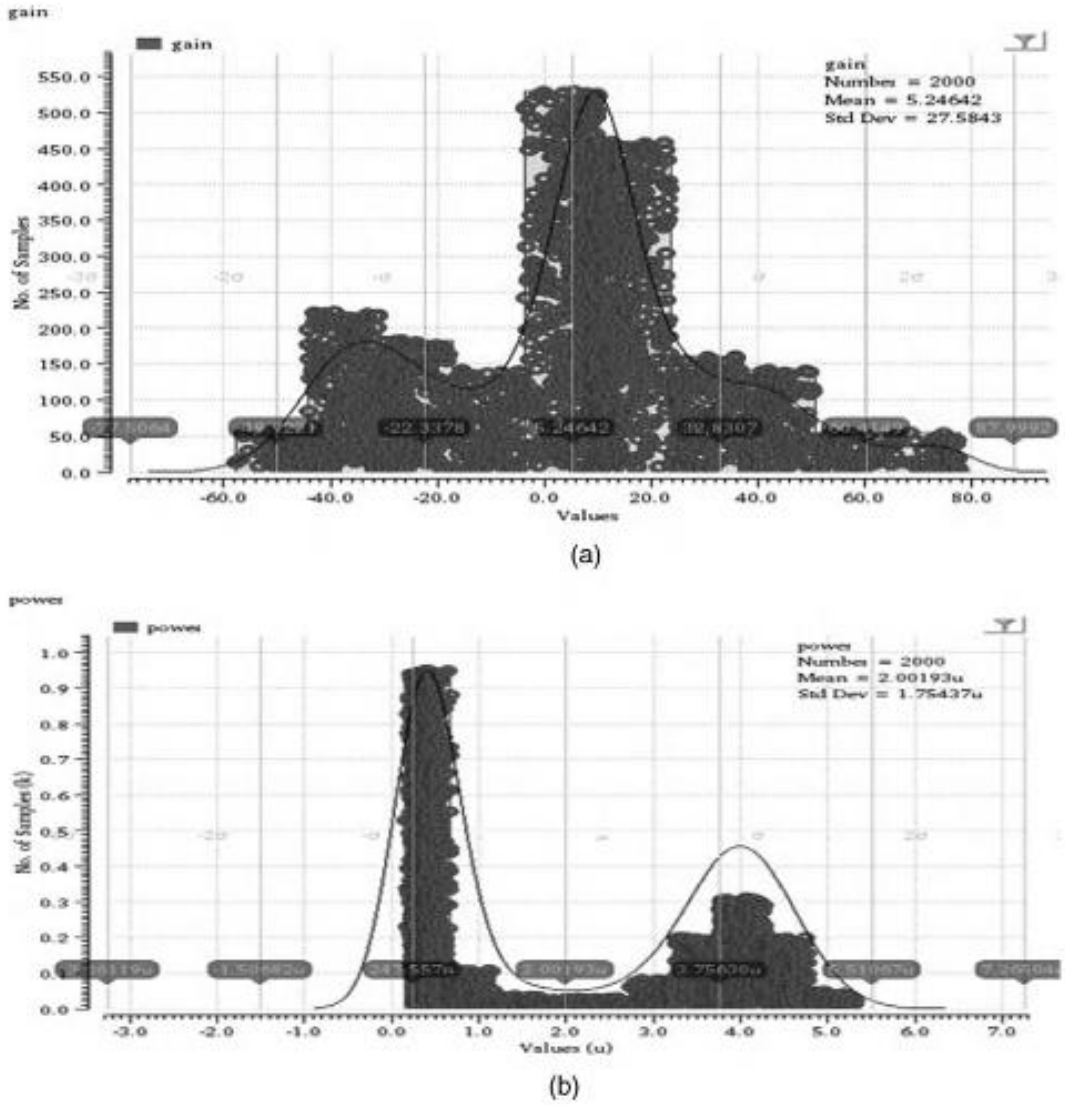
Gambar 11.3 menunjukkan penguatan DC sebesar 77,9 dB, yang tetap konstan hingga penurunan pada laju yang diharapkan sebesar -20 dB/dekade, dengan pergeseran fasa sebesar -118,07°, menghasilkan margin fasa sekitar 61,93°. Produk penguatan-bandwidth kira-kira 610 kHz pada desain 45 nm. Dengan cara yang serupa, CMRR frekuensi rendah sekitar 56 dB memastikan respons yang hampir stabil di seluruh rentang baseband dan PSRR sebesar -70 dB, yang menggarisbawahi kemampuan yang kuat untuk menahan noise catu daya frekuensi rendah. Pada frekuensi rendah (1–10 Hz), noise yang dirujuk ke input adalah 10 µV/√Hz, yang sesuai dengan noise flicker yang meningkat yang diamati pada teknologi CMOS sub-mikron. Kebisingan berkurang secara progresif di atas 100 Hz dan bergerak ke domain kebisingan termal, di mana ia stabil pada sekitar 2,5 µV/√Hz untuk frekuensi yang melebihi 10 kHz. Kebisingan keluaran menunjukkan pola spesifik, memvalidasi sumber kebisingan utama dan respons rangkaian. Penguatan kebisingan stabil pada sekitar 0,9 V/V di seluruh bandwidth penguat, dengan penurunan yang terlihat setelah sekitar 1 MHz, menunjukkan penurunan penguatan loop dan pengurangan amplifikasi kebisingan pada frekuensi yang lebih tinggi. OTA 45 nm mewakili offset mendekati nol hingga sekitar 17,86 mV.

Simulasi Monte Carlo komprehensif dengan 2000 sampel telah dilakukan. Simulasi ini memperhitungkan variasi dan ketidaksesuaian proses. Parameter utama yang dianalisis meliputi penguatan dan konsumsi daya total, CMRR, dan PSRR, yang sangat penting dalam menentukan efisiensi amplifikasi dan kesesuaian untuk aplikasi daya ultra-rendah. Berdasarkan distribusi statistik, Monte Carlo mensimulasikan ketidaksesuaian dan variasi proses secara realistis. Setiap parameter dihitung secara acak untuk setiap simulasi menggunakan model distribusi statistik. Dengan bantuan studi ini, kita dapat menentukan wilayah di mana sirkuit kita akan beroperasi paling sering. Rata-rata, median, dan deviasi standar yang diperoleh untuk parameter tersebut digambarkan pada Tabel 11.3, sedangkan Gambar 11.3a–d menunjukkan representasi histogram dari berbagai parameter.

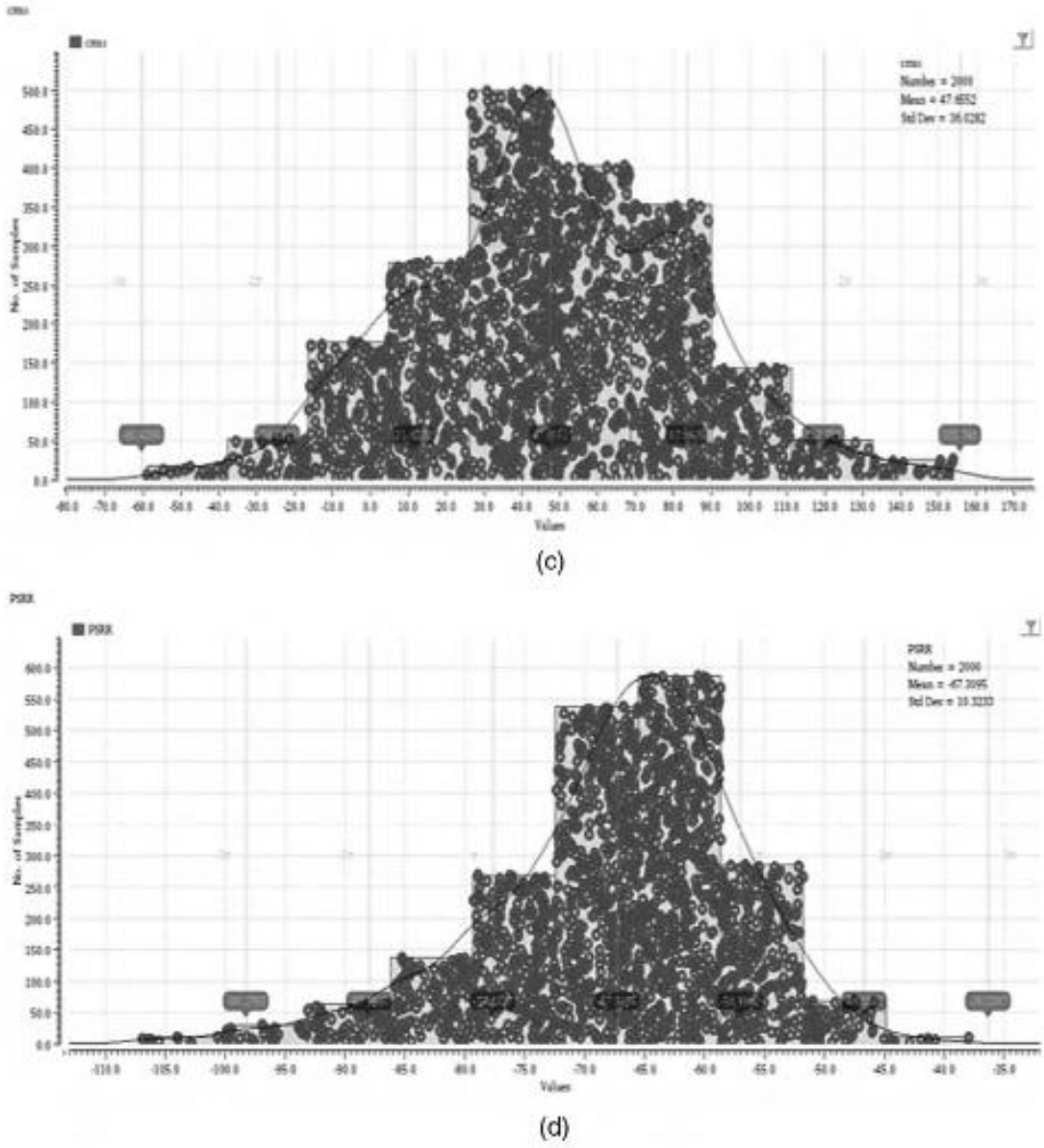
**Tabel 11.3** Hasil simulasi Monte Carlo

| MONTE CARLO 45 NM | RATA-RATA (MEAN) | MEDIAN | STANDAR DEVIASI |
|---|---|---|---|
| Penguatan (dB) | 5.24 | 8.384 | 27.58 |
| Cmrr (dB) | 51.98 | 53.5 | 5.067 |
| Psrr (dB) | -67.31 | -65.98 | 10.32 |
| Konsumsi Daya | 2.002 µW | 823.7 nW | 1.754 µW |

Histogram penguatan pada Gambar 11.4a menunjukkan penguatan rata-rata sebesar 5,25 dB, ditambah dengan deviasi standar sebesar 27,58 dB. Berbagai mode penguatan stabil hadir sebagai akibat dari ketidaksesuaian dan efek operasi di bawah ambang batas. Sejumlah sampel juga ditemukan dalam rentang 5–10 dB.




**Gambar 11.4** (a) Histogram penguatan. (b) Histogram konsumsi daya.
(Bersambung)




**Gambar 11.4** (Lanjutan) (c) Histogram CMRR. (d) Histogram PSRR.

Pada Gambar 11.4b terdapat deviasi standar sebesar 1,75 µW dan konsumsi daya rata-rata sekitar 2,0019 µW. Dengan puncak pada 0,25 dan 4–5 µW, histogram tersebut bimodal, menunjukkan operasi daya ultra-rendah yang konsisten dan seragam dalam kondisi ketidaksesuaian. Gambar 11.4c menunjukkan CMRR rata-rata sebesar 47,66 dB dan deviasi standar sebesar 36,03 dB, dengan variasi yang lebih besar antara -60 dan 155 dB. Gambar 11.4d menunjukkan PSRR rata-rata sebesar -67,31 dB, dengan deviasi standar sebesar 10,32 dB. Histogram menunjukkan distribusi unimodal dengan sedikit kemiringan, di mana sebagian besar sampel terkonsentrasi antara -75 dan -55 dB, menunjukkan kemampuan penolakan kebisingan yang lebih akurat.

**Tabel 11.4** Hasil simulasi sudut PVT.

| **Analisis Corner 45 Nm** | **Minimum** | **Maksimum** | **Rata-Rata** | **Median** | **Standar Deviasi** |
|---|---|---|---|---|---|
| Penguatan (DB) | 32.15 | 77.44 | 55.18 | 53.21 | 10.97 |
| Margin Fasa (°) | 33.4 | 82.9 | 66.38 | 69.09 | 11.02 |
| CMRR (DB) | 38.67 | 59.07 | 51.98 | 53.5 | 5.067 |
| PSRR (DB) | -88.06 | -30.07 | -51.75 | -50.08 | 13.37 |
| Konsumsi Daya | 11.04 nW | 22.02 µW | 4.906 µW | 857.6 nW | 6.789 µW |

Penguatan, fase, konsumsi daya, CMRR, dan PSRR dari penguat diferensial yang dirancang dianalisis secara komprehensif melalui analisis sudut Proses, Tegangan, dan Suhu (PVT) yang ekstensif menggunakan lingkungan Cadence Virtuoso ADE XL di seluruh teknologi CMOS 45 nm. Hasil Simulasi Sudut PVT digambarkan pada Tabel 11.4. Sebanyak 405 simulasi dilakukan dalam setiap kasus, dengan memvariasikan parameter kunci seperti kapasitor kompensasi (CC1 = 1,2 — 2,4 pF, $C_{C2}$ = 0,4 — 0,6 pF), tegangan suplai mulai dari 400 hingga 600 mV, dan suhu dari 0°C hingga 80°C, di atas sudut proses standar termasuk sudut TT, FF, SS, SF, FS, dan GPDK 45 (C0 — C4).

Analisis sudut teknologi node: CMOS 45 nm
Titik suhu: 0°C, 27°C, 80°C
Tegangan suplai ($V_{DD}$): 400, 500, 600 mV
Kapasitor kompensasi 1 ($C_{C1}$): 1, 1,5, 2 pF
Kapasitor kompensasi 2 ($C_{C2}$): 0,4, 0,5, 0,6 pF

Sirkuit ini menunjukkan beberapa keunggulan, termasuk penguatan maksimum 77,9 dB. Desain ini memastikan operasi yang stabil dalam rentang suhu dari 0°C hingga 80°C. Kehadiran kapasitor kompensasi ($C_{C1}$ dan $C_{C2}$) memungkinkan optimasi penguatan yang efisien dalam berbagai kondisi tanpa mengorbankan kinerja keseluruhan. Margin fasa berkisar dari 60,68° hingga 113,9°, bahkan dalam kasus terburuk, seperti suhu tinggi dan tegangan suplai rendah, margin tetap jauh di atas 60°. Hal ini memastikan stabilitas sirkuit yang andal dalam kondisi operasi yang menantang. Batas atas 113,9° mencerminkan kemampuan peredaman yang sangat baik, secara efektif menekan osilasi dan meminimalkan overshoot.

Konsumsi daya berkisar dari 11,04 hingga 22,02 µW, dengan rata-rata sekitar 4,906 µW dan deviasi standar 857,6 nW. CMRR mencapai minimum 38,67 dB, maksimum 59,07 dB, rata-rata 51,98 dB, dan deviasi standar 5,07 dB. Penurunan CMRR akibat suhu diamati dari 56,23 dB pada 0°C menjadi 50,29 dB pada 60°C, yang dapat dikaitkan dengan berkurangnya mobilitas pembawa dan ketidaksesuaian ambang batas pada suhu yang lebih tinggi. PSRR berkisar dari -88,06 hingga -30,07 dB, dengan rata-rata -51,75 dB dan deviasi standar 13,37 dB, menunjukkan variabilitas sedang. Hasil pengujian suhu menunjukkan peningkatan PSRR pada suhu yang lebih rendah dan sedikit penurunan pada suhu yang lebih tinggi, menekankan sensitivitas termal.

## 11.4 DESAIN OP-AMP BIOMEDIS 45 NM ULTRA-DAYA RENDAH BERBASIS RNMC

Topologi RNMC dasar disajikan untuk teknologi 45 nm. Gain, fase, CMRR, dan PSRR yang diinginkan dicapai dengan tegangan suplai yang lebih rendah, arus bias yang lebih rendah, dan konsumsi daya yang lebih rendah. Teknik kompensasi yang diusulkan menawarkan proses desain dan implementasi yang mudah, karena hanya bergantung pada komponen pasif. Secara khusus, hanya dua kapasitor kompensasi bernilai rendah yang diperlukan untuk mencapai kinerja optimal. Perbandingan analitis di antara metode yang diusulkan juga telah dilakukan. Dimensi dan kondisi bias transistor pada tahap kedua dan ketiga dipilih sedemikian rupa sehingga nilai kapasitor kompensasi tetap sekitar 2 pF.

Resistor nol diperkenalkan karena secara efektif memblokir jalur umpan maju, mencegah impedansi keluaran tahap pertama berkurang oleh kapasitor kompensasi pada frekuensi yang lebih tinggi. Akibatnya, penyertaan ini meningkatkan margin fase dengan meminimalkan degradasi penguatan tahap pertama pada frekuensi tinggi.

Meskipun karya khusus yang disajikan di sini dieksploitasi dengan membuat resistor nol RC=0. Analisis Monte Carlo lebih lanjut dan representasi histogram, yang hasilnya disebutkan dalam Tabel 11.3 dan Gambar 11.3a–d. Analisis sudut, yang hasilnya disebutkan dalam Tabel 11.4 untuk variabel yang berbeda, dilakukan. Tabel 11.5 mewakili studi perbandingan parameter kinerja untuk tahun-tahun dan pekerjaan yang dilakukan dalam artikel ini. Penguatan 77,9 dB untuk catu daya 500 mV dicapai dalam teknologi 45 nm. Penguat ini diberi bias untuk konsumsi daya arus 200 nA. Untuk rangkaian ini, konsumsi dayanya adalah 857,62 nW, yang merupakan terendah di antara semua karya yang disebutkan.

**Tabel 11.5** Studi perbandingan parameter kinerja

| Penulis | Grasso et al. [14] | Chong and Chan [35] | Mojarrd and Yavan [22] | Grasso et al. [24] | Amirzadah and Dasht [28] | Xie and Zhu [34] | Golabi et al. [25] | Qin et al. [25] | Centurelli et al. [31] | Centurelli et al. [31] | Amirzadah and Dashti [33] | Dong and Zhu [27] | Liu et al. [29] | Zaherfek and Biabonford [36] | Centurelli et al. [31] | Riad et al. [5] | Alizadeh Arand and Yavan [16] | Karya ini |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Tahun Publikasi | 2007 | 2012 | 2013 | 2014 | 2015 | 2015 | 2015 | 2016 | 2016 | 2018 | 2018 | 2018 | 2019 | 2019 | 2020 | 2020 | 2021 | 2023 |
| Teknologi (nm) | 500 | 65 | 90 | 65 | 90 | 65 | 90 | 65 | 65 | 65 | 90 | 180 | 65 | 180 | 130 | 130 | 90 | 45 |
| Tegangan Catu (V) | 3 | 1.2 | 0.9 | 1 | 1.2 | 1.2 | 1.2 | 1.2 | 1 | 1 | 1.2 | 1.2 | 1.2 | 1.8 | 1 | 1 | 1.2 | 0.5 |
| Penguatan | 109 | 100 | 92.1 | 91 | 100 | 101.1 | 72 | 100 | — | 99.5 | 110 | 100 | 72.9 | 112 | 120 | 83.46 | 81.6 | 77.9 |
| Daya (dB) | 0.255 mW | 0.0204 mW | 55.2 µW | 0.08 mW | 80 µW | 1.56 mW | 2.5 mW | 0.04 MW | 90 µW | 19 MW | — | 23 µW | — | 0.285 mW | 16.7 µW | 185 nW | 3.25 mW | 857.61 nW |
| GBW (MHz) | 2.4 | 2 | 15.1 | 19.06 | 1 | 330 | 121 | .62 | 14.66 | 1.35 | 4.65 | 6.5 | 2.410 | 14 | 24.8 | 1.7 | 193.6 | 0.610 |
| Laju Slew (V/µs) | 1.95/1.8 | 0.65 | 1.28 | 1.45 | 0.19 | 266.3 | 487.9 | 0.19 | — | 0.85/ 0.97 | 1.71 | 0.83 | 17.25 | 0.8 | 10 | 0.17 | 327.3 | — |
| Marjin Fase margin (°) | 58 | — | 75 | 59 | — | 58 | 63.8 | 58 | 67.3 | 72.8 | 57 | 48.2 | 82.6 | 86 | 73 | 82.36 | 61.8 | 61.56 |
| CMRR (dB) | 79 | — | — | — | — | | — | — | — | — | — | — | — | — | — | — | | 56 |
| Waktu Pemulihan | 497/ 560 | — | 870/ 433 ns @ 1% | — | — | 9.3 ns | 75 ns @ 0.02% | — | 30.1/ 27.6 µs @ 1% | 431/ 433 100 mV (ns) | 0.52/0.45 µs @ 0.1% | — | 1.4 ns 1% | — | 117/ 105 ns | 0.89 µs @ 1% | 9.1 µs @ 1% | — |

**Tabel 11.5** (Lanjutan)

| Penulis | Grasso et al. [14] | Chang and Chen [35] | Mojarad and Yavari [22] | Grasso et al. [24] | Amintadeh and Dashti [28] | Xie and Zhu [34] | Golabi et al. [25] | Qin et al. [20] | Centurelli et al. [31] | Centurelli et al. [31] | Amintadeh and Dasha [32] | Dong and Zhu [27] | Liu et al. [29] | Zaherfekr and Bizbankard [36] | Centurelli et al. [31] | Riad et al. [31] | Alizodeh Arand and Yayari | Karya ini |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Arus | 85 µA | 17 µA | 61 µA | 80 µA | 66 µA | — | — | 50 µA | 10 µA | 19.1 | 17 | 10.6 mA | — | — | 112 | — | 200 nA | |
| Derau tegangan input | — | — | — | — | — | — | — | — | — | — | | — | — | — | — | — | — | 1.005 µV/√Hz |
| FOM ANGKA KEUNG-GULAN (FOM) | 4706 FOMS MHz pF/ mW | 490,20 FOMS MHz pF/ mW | 273,551 FOMS MHz pF/ mW | 238,250 FOMS MHz pF/ mW | 125,00 FOMS MHz pF/ mW | 1057.7 FOMS MHz pF/ mW | 96.8 FOMS | 88,571 FOMS MHz pF/ mW | — | — | 113,970 FOMS MHz pF/mW | 158,700 FOMS MHz pF/ mW | 382.5 FOMS | 10.76 FOMS | 1485 FOMS MHz pF/ mW | 4.63 GB. CL./(IDC area) MHzpF / mA µm² | 125.32 FOMS MHz pF/ mW | 1422 MHz pFµW |
| FOM ANGKA KEUNG-GULAN (FOM) | 3529 FOML V/µs pF/ mW | 15,931 FOML V/µs pF/mV | 23,188 FOML V/µs pF/ mW | 18,125 FOMS V/µs pF/ mW | 23,750 FOML V/µs pF/mV | 855.3 FOML V/µs pF/mV | 390.3 FOML | — | — | — | 41,912 FOMS MHz pF/mW | 20,176 FOMS MHz pF/ mW | 273.3 FOML | 0.61 FOML | 598 FOMS nHz/ pF/ mW | 0.87 SR.CL/ (IDC. area) V/µs pF/mA µm² | 211.4 FOML V/µs pF/mV | — |
| PSSR (dB) | 75 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | | — | 69.24 |
| Luas estimasi (mm²) | — | — | — | — | — | — | — | 0.008 | — | — | — | 0.006 | — | — | — | 0.006 | — | — |

Selanjutnya, oleh sebuah studi perbandingan ditunjukkan untuk NMC dan RNMC dengan buffer tegangan sebagai jalur umpan balik dengan resistor nol dan tanpa resistor nol, di mana peningkatan yang cukup besar dalam parameter keseluruhan dicantumkan dan memberikan ruang untuk pekerjaan di masa mendatang untuk OTA yang disajikan dalam karya ini, yang akan diimplementasikan dengan nilai resistor nol RC yang dihitung seperti yang disebutkan dan memberikan analisis perbandingan untuk peningkatan dalam parameter keseluruhan.

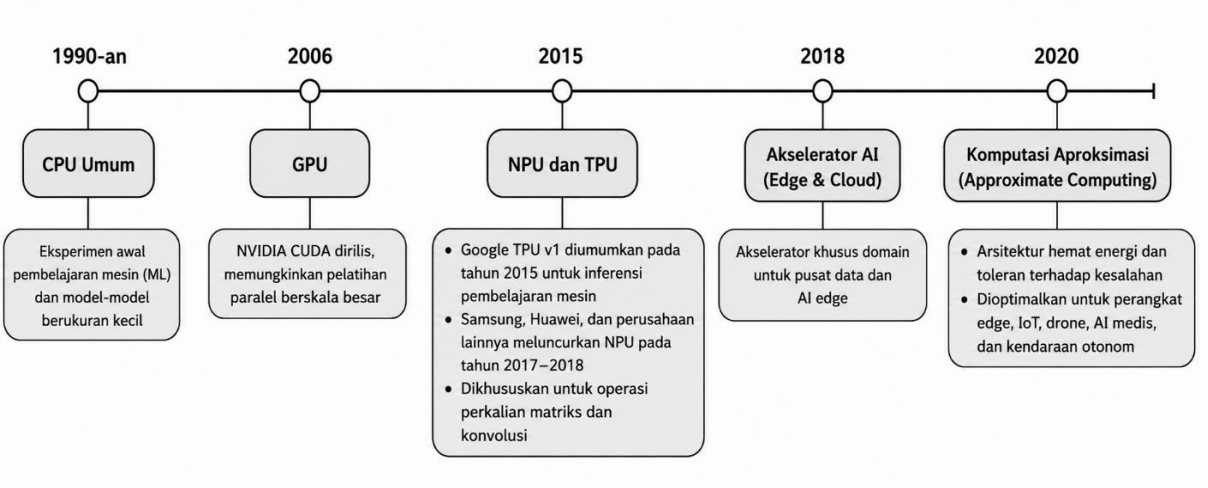
Seiring dengan perkembangan teknologi CMOS ke tingkat submikron yang sangat kecil seperti 45 nm, desain menjadi lebih kompleks, dan variasi dalam metode desain analog tradisional menjadi tidak efisien. AI dan ML memungkinkan optimasi otomatis berbasis data dari teknik sirkuit analog dan sinyal campuran seperti GA, PSO, BO, dan RL, secara cerdas menyempurnakan ukuran transistor, bias, dan parameter kompensasi untuk mencapai gain-bandwidth tinggi, daya rendah, dan operasi yang stabil. Terintegrasi ke dalam alat EDA modern, metode ini meningkatkan akurasi, adaptabilitas, dan kecepatan. Untuk penguat operasional RNMC tiga tahap berdaya rendah yang diusulkan, desain yang dibantu AI/ML, yang merupakan cakupan masa depan artikel ini, memastikan kinerja yang kuat dan efisien, menandai langkah kunci menuju sistem VLSI cerdas dan otonom untuk aplikasi IoT biomedis dan AI tepi.

# BAB 12
# KOMPUTASI PERKIRAAN UNTUK PEMBELAJARAN MESIN

## 12.1 EFISIENSI PERANGKAT KERAS AI MELALUI KOMPUTASI APROKSIMASI

Pembelajaran Mesin (*Machine Learning/ML*) telah merevolusi teknologi ke tingkat yang berbeda dalam beberapa tahun terakhir, menemukan aplikasinya di semua domain – dari perawatan kesehatan dan keuangan hingga sistem otonom dan ilmu roket. Seiring kemajuan ML, ia menjadi disiplin ilmu yang sangat menuntut komputasi, meningkatkan permintaan akan sumber daya perangkat keras untuk melatih dan menerapkan model kompleks pada kumpulan data yang besar.

Kebutuhan akan solusi ML yang lebih akurat, efisien, dan terukur meningkatkan kebutuhan akan sumber daya perangkat keras komputasi berkinerja tinggi, menjadikan perangkat keras yang kuat sebagai tulang punggung ML. Kebutuhan yang cepat akan perangkat keras seperti NPU, TPU, dan Akselerator AI meningkatkan kebutuhan untuk mengoptimalkan, meningkatkan, dan menciptakan arsitektur dan algoritma, mendorong batas menuju penerapan berbasis tepi, integrasi cloud, dan infrastruktur pengoptimalan biaya untuk mencapai keseimbangan antara kendala teknologi dan ekonomi (Gambar 12.1).




**Gambar 12.1** merepresentasikan garis waktu komponen perangkat keras ML.

### Komputasi Aproksimasi

Komputasi aproksimasi adalah teknik yang dapat digunakan dalam ML untuk mengurangi aspek energi, area, dan latensi suatu sistem, dengan akurasi hasil sebagai imbalannya. Strategi ini akan sangat bermanfaat untuk aplikasi yang dapat mentolerir ketidakakuratan sampai batas tertentu, sehingga tidak akan memberikan pengaruh signifikan pada hasil keseluruhan sistem. Munculnya komputasi aproksimasi dimulai dengan kebutuhannya dalam berbagai aplikasi seperti aritmatika komputer, unit aritmatika (seperti penjumlahan dan perkalian), perancangan filter, dan algoritma sistem. Namun, sangat penting untuk mengimplementasikan teknik komputasi aproksimasi dengan hati-hati pada lapisan

yang dibutuhkan dengan analisis dan pemahaman sistem yang tepat, karena kesalahan interpretasi apa pun dapat menyebabkan anomali drastis pada hasilnya.

Dalloo telah membahas langkah-langkah yang harus diikuti untuk mengimplementasikan komputasi aproksimasi. Hal ini melibatkan identifikasi unit komputasi non-kritis di mana penerapan aproksimasi tidak dapat memengaruhi kualitas output, diikuti dengan perancangan aproksimasi untuk meningkatkan kinerja. Sebelum menerapkan komputasi aproksimasi secara real-time, terutama dalam aplikasi kritis, hal tersebut harus dianalisis, dipantau, dinilai, dan disesuaikan dengan benar. Tujuan utama dari setiap teknik komputasi aproksimasi adalah untuk meminimalkan ketidakakuratan, bersamaan dengan efisiensi sumber daya. Menurut Zervakis dkk., kendala lain yang diinginkan untuk menerapkan komputasi aproksimasi dengan efisiensi yang baik adalah konfigurasi kesalahan, bias kesalahan yang rendah, dan pendekatan sistematis untuk aproksimasi yang diperkenalkan.

**Manfaat Komputasi Aproksimasi**

Pembahasan terbaru sebagian besar berfokus pada bagaimana komputasi aproksimasi menawarkan efisiensi energi, terutama dalam arsitektur ML. Survei oleh Leon dkk. melaporkan pengurangan daya yang luar biasa sebesar 22%–54% dalam aplikasi ML dengan komputasi aproksimasi dibandingkan dengan implementasi yang tepat, dengan kehilangan akurasi nol dalam beberapa kasus. Tidak hanya meningkatkan efisiensi energi, tetapi juga meningkatkan efisiensi dan kecepatan pelatihan dengan menyederhanakan sumber daya komputasi. Pengurangan bit dapat membantu mengurangi ukuran unit komputasi, terutama unit MAC, yang merupakan tulang punggung ML di sisi perangkat keras. Misalnya, lebar bit dapat dikurangi dari 16 bit menjadi 2 bit dengan pemetaan yang tepat dengan nilai presisi dan tingkat kuantisasi. Selain itu, nilai presisi rendah juga mengurangi kebutuhan bandwidth dan memori on-chip. Teknik aproksimasi lintas lapisan, yang akan dibahas di Bagian 1.4.5 bab ini, dilaporkan mencapai penghematan energi 2 × dengan hanya kehilangan akurasi sekitar 2,5% di beberapa inti oleh Leon et al.

**Ketahanan Terhadap Kesalahan dalam Komputasi Aproksimasi**

Ketahanan terhadap kesalahan adalah sifat bawaan dari aplikasi tertentu yang dimanfaatkan oleh teknik komputasi aproksimasi. Dallo dkk. secara eksplisit menyatakan bahwa komputasi aproksimasi bergantung pada ketahanan terhadap kesalahan dalam aplikasi target, dan Leon dkk. menyebutkan sifat ketahanan terhadap kesalahan yang melekat pada beban kerja ML dan pemrosesan sinyal digital, menjadikannya domain yang cocok untuk teknik aproksimasi.

**Komputasi Aproksimasi dalam Pembelajaran Mesin**

Konvergensi komputasi aproksimasi dengan pembelajaran mesin dapat membuka banyak kemungkinan kemajuan jika dibandingkan dengan implementasi ML tradisional, menjadikannya domain aktif untuk pembahasan dan eksperimen di berbagai tingkatan. Hal ini juga dapat dianggap sebagai langkah yang diperlukan untuk meningkatkan sistem ML dalam hal luas area dan konsumsi daya. Perlu dicatat bahwa banyak GPU, TPU, FPGA, dan NPU komersial digunakan dengan teknik komputasi aproksimasi yang terintegrasi di dalamnya.

Prosesor dan akselerator aproksimasi merupakan persyaratan utama untuk implementasi ML, yang terutama dibangun menggunakan unit *Multiply-and-Accumulate* (MAC). Dalam jaringan saraf, komputasi paling utama dan terpenting dilakukan oleh unit MAC, terutama pengali. Pilar komputasi utama dari *Deep Neural Networks* (DNN) adalah konvolusi dan perkalian matriks, menjadikan unit MAC sebagai persyaratan paling penting untuk operasi aritmatika. Pengali adalah komponen unit MAC yang paling banyak mengkonsumsi sumber daya dan daya, yang bertanggung jawab atas konsumsi energi utama dalam jaringan saraf dan DNN. Perlu dicatat, 99% energi dalam DNN dikonsumsi oleh pengali; oleh karena itu, kita dapat mengatakan bahwa pengali adalah target yang sangat penting untuk aproksimasi.

## 12.2 TEKNIK KOMPUTASI APROKSIMASI

Bidang komputasi aproksimasi mencakup beragam strategi, mulai dari algoritma tingkat tinggi hingga penyederhanaan tingkat transistor. Memahami strategi ini dan mengetahui cara mengimplementasikannya pada posisi yang tepat sangat penting untuk mendesain sistem, sehingga sesuai untuk aplikasi komputasi modern sambil mempertahankan kualitas layanan yang dapat diterima.

**Teknik Aproksimasi Perangkat Lunak**

Teknik aproksimasi tingkat perangkat lunak terutama berfokus pada modifikasi tingkat algoritma dan program untuk mengurangi kompleksitas komputasi, memanfaatkan toleransi kesalahan yang melekat dalam banyak aplikasi. Perforasi Loop adalah salah satu teknik aproksimasi perangkat lunak yang paling intuitif, di mana iterasi loop tertentu dilewati selama eksekusi program. Prinsip dasarnya adalah bahwa banyak algoritma iteratif menunjukkan sifat konvergensi di mana iterasi langsung hanya berkontribusi sedikit pada hasilnya. Teknik ini sangat efektif dalam aplikasi pemrosesan multimedia.

Memorisasi perkiraan adalah teknik yang menyimpan hasil komputasi sebelumnya dan menggunakannya kembali untuk pola input yang kurang lebih serupa. Pendekatan ini mempertahankan tabel pencarian yang berisi pasangan input-output dari komputasi sebelumnya, dan ketika input baru tiba, sistem mencari input serupa yang telah diproses sebelumnya.

Penskalasian Presisi digunakan untuk mengurangi presisi representasi data dan komputasi secara numerik selama eksekusi program. Ini terutama berlaku dalam aplikasi yang dapat mentolerir pengurangan presisi dalam hasil antara atau akhir tanpa penurunan kualitas. Metodologi ini sering menggunakan metode analisis kesalahan formal untuk memastikan kehilangan presisi tetap dalam batas yang dapat diterima sambil memaksimalkan manfaat energi dan kinerja.

Pengambilan Sampel Data mengatasi tantangan komputasi yang terkait dengan aplikasi big data dengan memproses subset representatif dari kumpulan data lengkap. Teknik ini menggunakan teori pengambilan sampel sehingga mengidentifikasi dan mempertahankan karakteristik penting dari seluruh kumpulan data. Pendekatan ini sangat berlaku dalam aplikasi berbasis pembelajaran mesin untuk pelatihan kumpulan data lengkap dengan pengurangan komputasi yang signifikan.

**Teknik Aproksimasi Perangkat Keras (atau) Sirkuit**

Aproksimasi tingkat perangkat keras menargetkan modifikasi sirkuit, terutama blok fundamental seperti adder dan multiplier, untuk mencapai efisiensi sumber daya. Hal ini dapat dilakukan dengan melonggarkan batasan waktu sirkuit yang ada atau merancang implementasi sirkuit yang secara fungsional mendekati. Penskalasian Tegangan Berlebih adalah teknik di mana tegangan suplai sengaja dikurangi di bawah spesifikasi desain dan termasuk kesalahan waktu, serta mengurangi konsumsi daya. Tantangan utama dalam metode ini adalah mengelola kesalahan waktu dalam kondisi tegangan yang dikurangi.

Pendekatan Fungsional Sirkuit melibatkan perancangan unit aritmatika dan logika yang sengaja menghasilkan hasil perkiraan sambil mengurangi kompleksitas perangkat keras. Pendekatan ini mencakup multiplier perkiraan, yang akan kita fokuskan lebih lanjut dalam bab ini. Ini terutama berlaku dalam aplikasi toleransi kesalahan komputasi untuk pengurangan substansial dalam luas area, penundaan, dan konsumsi daya. Overclocking adalah teknik yang membuat sirkuit beroperasi pada frekuensi yang lebih tinggi daripada spesifikasi desain untuk sengaja menimbulkan pelanggaran waktu guna meningkatkan throughput.

**Teknik Aproksimasi Spesifik Aplikasi**

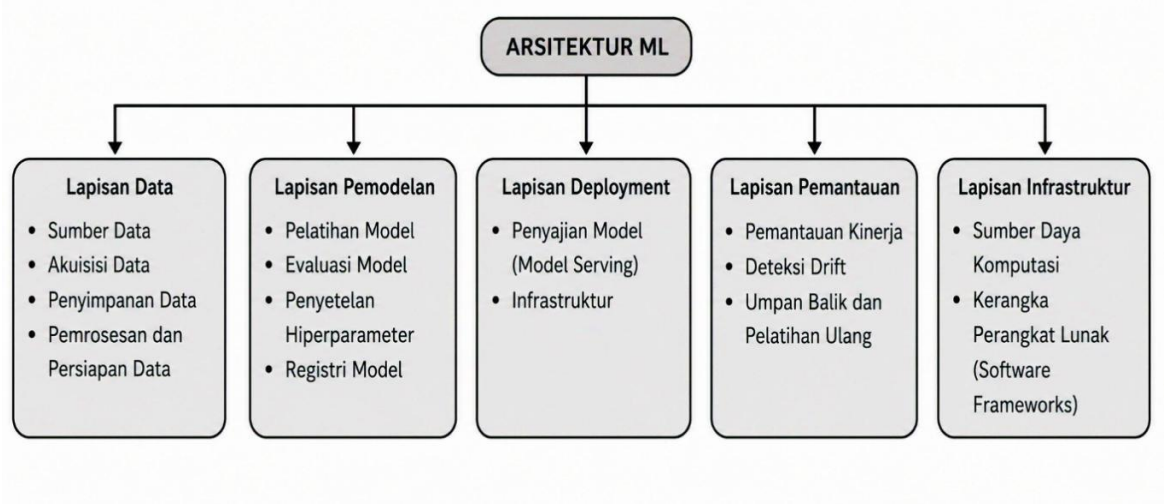
Teknik aproksimasi spesifik aplikasi adalah metode yang dirancang khusus untuk aplikasi yang membutuhkan komputasi intensif seperti pembelajaran mendalam, kecerdasan buatan, dan pemrosesan sinyal digital untuk pengurangan komputasi strategis. Pendekatan ini dapat beroperasi di seluruh domain perangkat lunak, perangkat keras, dan lintas lapisan.

**Teknik Pendekatan Arsitektur**

Teknik aproksimasi arsitektur beroperasi pada tingkat prosesor dan sistem memori. Pendekatan ini berkaitan dengan infrastruktur sistem untuk mendukung komputasi aproksimasi dengan bantuan arsitektur perangkat keras tertentu sesuai dengan persyaratan. Prosesor aproksimasi menggabungkan kemampuan aproksimasi langsung ke dalam pipeline, sehingga dapat beroperasi pada berbagai tingkat akurasi, sementara penyimpanan data aproksimasi memanfaatkan hierarki konsumsi daya yang signifikan dengan menerapkan berbagai teknik dalam sistem DRAMM dan SRAM.

## 12.3 ARSITEKTUR ML

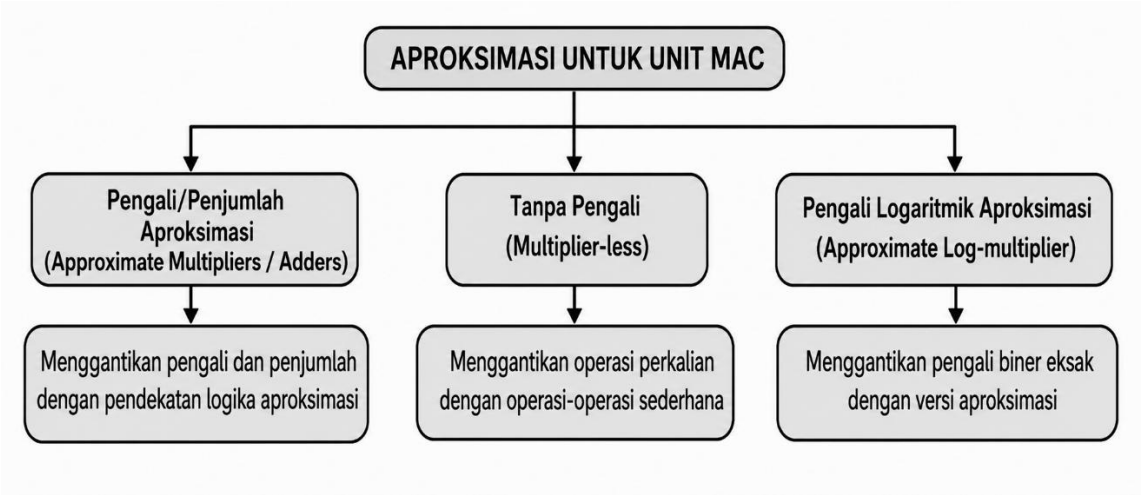
Arsitektur ML adalah kerangka kerja yang terdiri dari komponen-komponen seperti unit memori, ALU, unit MAC, dan antrian yang saling terhubung untuk membangun, melatih, menyebarkan, dan memelihara model ML. Umumnya, pendekatan berbasis pipeline digunakan untuk memindahkan dan memproses data. Setiap arsitektur ML unik, tergantung pada persyaratan aplikasinya. Komponen utama dari arsitektur ML adalah penyerapan data, penyimpanan data, kontrol versi data, penilaian model, penyebaran model, pemantauan model, pelatihan model, dan pelatihan ulang model (Gambar 12.2).




**Gambar 12.2** Merepresentasikan arsitektur ML sebagai kerangka lapisan.

**Peran Unit MAC dalam Akselerator ML**

Seperti yang telah kita bahas sebelumnya, sebagian besar operasi komputasi dalam jaringan saraf, khususnya pada DNN dan CNN, adalah operasi MAC. Armenia-kos dkk. menyatakan bahwa miliaran MAC akan dibutuhkan untuk melakukan operasi jaringan saraf, terutama CNN dan DNN. Dari sini, kita dapat menyimpulkan bahwa kompleksitas dataset dan kebutuhan komputasi berbanding lurus dengan jumlah unit MAC yang dibutuhkan. Untuk mempercepat jaringan saraf, operasi paralel unit MAC dapat meningkatkan kinerja. Perlu juga dicatat bahwa Google TPU memiliki sekitar 64.000 unit MAC yang terintegrasi di dalamnya, dan NPU Samsung memiliki sekitar 6.000 (Gambar 12.3).




**Gambar 12.3** Teknik aproksimasi yang dapat digunakan dalam unit MAC.

**PALARON**

PALARON adalah singkatan dari *Parallel Architecture-Level Approximate-Computing-Based Accelerator for Recurrent Neural Networks*, yaitu akselerator perangkat keras yang dirancang khusus untuk meningkatkan kinerja dan efisiensi energi dari Jaringan Neural Berulang (RNN) menggunakan komponen perangkat keras perkiraan. Ia menggunakan

arsitektur paralel dengan unit pemrosesannya yang bekerja secara simultan untuk mempercepat proses perkalian yang merupakan tulang punggung RNN. Mesin MAC diberi nama PARV-CE (*Precision-Aware Runtime-adaptive Vector Compute Element*), yang merupakan tulang punggung akselerator ini. Dilaporkan juga bahwa desain PALARON mengkonsumsi area dan daya yang lebih sedikit saat beroperasi pada frekuensi tinggi dibandingkan dengan unit MAC perkiraan lainnya. Ia dirancang khusus dengan arsitektur pipeline lima tahap yang dirancang untuk latensi rendah dan efisiensi area.

**GAMA**

GAMA adalah desain kerangka kerja yang digunakan untuk mempercepat Perkalian Matriks-Matriks Umum (GEMM), yang merupakan tulang punggung sebagian besar operasi ML. Desain komprehensif ini secara khusus ditingkatkan untuk berjalan pada arsitektur Versal AI Engine (AIE-ML) generasi kedua AMD, yang sangat disesuaikan untuk beban kerja ML. GEMM terdiri dari sejumlah besar operasi MAC (Tabel 12.1).

**Tabel 12.1** Perbandingan PALARON dan GAMA

| ASPEK | PALARON | GAMA |
|---|---|---|
| Tujuan Utama | Efisiensi Energi | Performa Tinggi |
| Teknik Inti | Komputasi Aproksimasi | Pemetaan Berperforma Tinggi |
| Implementasi | Desain Chip Baru | Kerangka Kerja Kompilator Dan Pemetaan |
| Inovasi Utama | Pemrosesan Paralel Dan Komputasi Aproksimasi | Kerangka Kerja Yang Dapat Diskalakan Dengan Teknik-Teknik Baru Seperti Penempatan Kernel Berjenjang Dan Manajemen Buffer Khusus |
| Tingkat Abstraksi | Desain Arsitektur Perangkat Keras Tingkat Rendah | Kerangka Kerja Perangkat Lunak Dan Desain Tingkat Tinggi |

**Peran Pengali dalam Unit MAC untuk Akselerasi ML**

Unit MAC terdiri dari adder dan multiplier, tetapi karena adder lebih sederhana dan kurang kompleks dibandingkan dengan multiplier, akan sangat berdampak jika kita fokus pada aproksimasi multiplier. Oleh karena itu, kita dapat menyimpulkan bahwa aproksimasi multiplier dapat menyebabkan transisi besar dalam kinerja, konsumsi energi, dan ukuran perangkat. Seiring meningkatnya kompleksitas komputasi, multiplier dari unit MAC akan secara fundamental menentukan kinerja, kebutuhan energi, dan skalabilitas akselerator ML. TPU generasi pertama Google memiliki 65.536 unit MAC, di mana unit MAC disebut unit perkalian dan penjumlahan untuk menangani operasi bilangan bulat 8-bit untuk matriks.

Peran multiplier melampaui peningkatan kecepatan aritmatika. Ia berfungsi sebagai jangkar arsitektur yang menentukan seluruh pola pergerakan data dalam akselerator. Penempatan pengali dan optimasi jalur aliran data merupakan elemen penting dalam perancangan akselerator ML. Pengali pada unit MAC merupakan elemen pemrosesan (PE) fundamental dari akselerator pembelajaran mendalam. Setiap pengali menangani pola

komputasi yang berbeda seperti konvolusi, perkalian matriks, dan komputasi produk titik. Pada pengali tingkat lanjut, arsitektur paralelisme subkata dapat diimplementasikan untuk mengoperasikan beberapa operasi presisi rendah secara bersamaan. Misalnya, satu pengali 16-bit dapat dikonfigurasi untuk satu perkalian 16 × 16 bit atau dua perkalian 8 × 16 bit atau empat perkalian 4 × 16 bit. Fleksibilitas ini membuka jalan menuju pemanfaatan perangkat keras maksimum.

**Kompleksitas Pengali untuk ML**

Kompleksitas pengali tradisional adalah O($n^2$), tetapi untuk pengali yang dirancang khusus untuk operasi ML, kompleksitasnya sangat bervariasi berdasarkan presisi dan arsitektur. Pemilihan algoritma ML yang digunakan akan menentukan kompleksitas sistem. Kompleksitas pengali bergantung pada persyaratan presisi aplikasi target. Pengali eksak seperti pengali pohon Wallace atau pengali Dadda membutuhkan produk parsial, dan seiring bertambahnya jumlah bit, kompleksitasnya juga akan meningkat.

Sedangkan pada pengali aproksimasi, kompleksitas berkurang dalam berbagai tahapan dengan strategi yang berbeda. Misalnya, pengali terpotong mengurangi jumlah produk parsial melalui pemotongan strategis, pengali susunan terputus mengurangi jumlah unit komputasi yang dibutuhkan, bersama dengan akurasi yang dapat diterima, dan desain toleran kesalahan menyederhanakan gerbang logika dan mengurangi luas area dibandingkan dengan implementasi eksak.

**Disipasi Energi pada Pengali**

Seperti yang dibahas sebelumnya, pengali unit MAC mengkonsumsi lebih banyak energi dibandingkan dengan adder. Memahami disipasi energi pengali unit MAC sangat penting untuk optimasi daya, khususnya ketika aplikasi meluas ke perangkat portabel atau yang dioperasikan dengan baterai. Pada pengali, pembangkitan dan pengurangan produk parsial memberikan kontribusi signifikan terhadap disipasi energi.

Optimasi dapat dilakukan pada berbagai tingkatan – sirkuit, arsitektur, dan algoritma, untuk mengurangi konsumsi energi sesuai dengan kebutuhan aplikasi. Jaswal dkk. mendemonstrasikan evaluasi perangkat keras, yang menyatakan bahwa hingga 30,24% energi dihemat dengan penggunaan pengali aproksimasi dibandingkan dengan pengali yang ada. Bao dkk. juga melaporkan pengurangan energi sebesar 47% dengan implementasi desain berbasis kuantisasi dari unit MAC. Helal dkk. mengusulkan arsitektur VUSA yang mendemonstrasikan akselerator modern yang memanfaatkan sparsitas inheren dalam jaringan saraf, yang mengklaim efisiensi daya sekitar 68%.

**Arsitektur Pengali Tingkat Lanjut**

Arsitektur pengali tingkat lanjut dapat mengoptimalkan area, daya, dan kecepatan dengan melakukan modifikasi pada representasi data dan pola komputasi. Mari kita bahas arsitektur perkalian berbasis pengkodean dan arsitektur serial multi-unit pada bagian ini.

**Perkalian Berbasis Pengkodean**

Arsitektur perkalian berbasis pengkodean adalah arsitektur yang membayangkan kembali operasi perkalian tradisional dengan memetakan input ke representasi bit lebar menggunakan gerbang logika sederhana, menetapkan bobot posisi yang dapat dipelajari ke

bit output individual, dan melakukan akumulasi melalui penjumlahan berbobot bit-wise. Liu dkk. telah menyatakan bahwa pengali berbasis pengkodean dapat mencapai pengurangan area sebesar 79,63% dan menghemat daya sebesar 70,18% dibandingkan dengan desain pengali konvensional.

**Arsitektur Serial Multi-Bit**

Arsitektur serial multi-bit merupakan perluasan dari arsitektur bit-serial, yang memungkinkan presisi bit variabel penuh tanpa batasan arsitektur, mengurangi kompleksitas perangkat keras melalui kontrol logika yang disederhanakan, operasi hemat energi untuk komputasi presisi rendah, dan penskalaan presisi yang mulus selama runtime. Arsitektur serial multi-bit memproses beberapa bit per siklus dan menyeimbangkan fleksibilitas operasi bit-serial.

## 12.4 JENIS-JENIS PENGGANDA PERKIRAAN

Berbagai jenis pengali perkiraan diusulkan oleh para peneliti dan diimplementasikan. Kita akan menyelidiki beberapa pengali perkiraan penting yang menemukan ruang lingkupnya di domain ML.

**Pengali Toleransi Kesalahan (ETM)**

Pengali toleransi kesalahan (ETM) mewakili pendekatan yang ampuh dan pragmatis untuk desain sistem berkinerja tinggi dan hemat daya. Pengali ini disederhanakan dan menggunakan lebih sedikit unit logika, sehingga mengurangi luas dan konsumsi daya karena pengurangan kompleksitas jalur data dan penundaan sinyal. Ide inti dari metode ini adalah untuk memperkenalkan dan mengelola kesalahan dengan hati-hati, sehingga menghasilkan dampak minimum pada output akhir, sehingga memberikan keuntungan yang signifikan dalam metrik PPA.

Pengali standar umumnya dibagi menjadi tiga tahap: pembangkitan produk parsial, akumulasi produk parsial, dan *adder carry-propa-gate* akhir untuk mendapatkan hasil akhir. Tahap pertama, yaitu, mengaproksimasi produk parsial, dapat dilakukan dengan banyak metode, salah satunya adalah algoritma pengkodean Booth yang dimodifikasi (MBE). Liu dkk. mengeksplorasi teknik MBE radix-4, yang dapat mengurangi jumlah produk parsial hingga setengahnya. Encoder ini menyederhanakan logika dengan memilih kelipatan dari pengali, menawarkan penghematan daya dan area. Tahap selanjutnya adalah akumulasi produk parsial menjadi dua baris akhir, yang merupakan tahap paling kompleks dan memakan waktu. Beberapa metode dapat digunakan untuk menyederhanakan, beberapa di antaranya adalah pemotongan, perforasi, dan kompresor perkiraan. Pemotongan (*Truncation*) adalah metode yang mudah dan sederhana di mana kolom-kolom yang paling tidak signifikan dari matriks produk parsial diabaikan, tetapi dapat menimbulkan kesalahan besar. Perforasi adalah pendekatan yang lebih terarah yang melibatkan secara selektif melewatkan seluruh baris dari matriks produk parsial, tetapi semakin banyak baris yang dihilangkan, semakin besar potensi kesalahan. Teknik yang sangat efektif dan banyak digunakan adalah kompresor perkiraan (approximate compressor), yang mengganti komponen eksak dengan komponen yang disederhanakan. Tingkat akurasi bergantung pada pemilihan algoritma perkiraan yang tepat

pada tingkat yang tepat. Keuntungan ETM meliputi peningkatan efisiensi perangkat keras, pengurangan kompleksitas, dan fleksibilitas desain. Efisiensi perangkat keras berarti penurunan konsumsi daya, membuat area sirkuit lebih kecil, dan meningkatkan kecepatan. Kinerja ditingkatkan dengan mengurangi penundaan dan meningkatkan kecepatan clock.

Di sisi lain, terlepas dari keuntungannya, hilangnya akurasi merupakan kelemahan utama dari perkiraan. Beberapa metode perkiraan menghasilkan pola kesalahan yang tidak menguntungkan, yang mengakibatkan kesalahan acak dan tidak bias. Untuk mengatasi kesalahan besar, beberapa desain menggabungkan sirkuit kompensasi kesalahan untuk menyeimbangkan dan mengoreksi perkiraan, yang akan meningkatkan luas dan konsumsi daya.

**Pengali Terpotong**

Perkalian terpotong adalah teknik optimasi perangkat keras yang efisien yang mengorbankan sedikit akurasi dengan cara menghilangkan sebagian kecil beban komputasi yang paling sedikit berkontribusi pada hasil akhir. Biasanya, bit paling tidak signifikan (LSB) diabaikan. Alih-alih menghitung seluruh bit, hanya bagian-bagian signifikan dari hasil perkalian yang dipilih dengan cermat atau dibulatkan sebelum komputasi. Ini umumnya digunakan sebagai unit pengali paralel.

Kunci operasi pengali ini adalah bagaimana kolom paling tidak signifikan dari matriks hasil perkalian parsial ditangani dan bagaimana propagasi kesalahan dikelola. Untuk pengali n-bit yang hasil perkaliannya akan memiliki 2n bit, mari kita asumsikan bahwa pengali terpotong memotong 't' bit paling tidak signifikan dari hasil perkalian. Schulte dkk. Diperkirakan bahwa hal ini dapat menghemat t(t - 1)/2 gerbang AND, (t - 1)(t - 2)/2 adder penuh dan (t - 1) adder setengah dalam kasus arsitektur pengali array. Karena output akhir akan kecil, dengan penggunaan adder yang lebih cepat, kecepatan komputasi keseluruhan berkurang. Namun, penghilangan produk parsial menimbulkan kesalahan reduksi, yang merupakan jumlah bit produk parsial positif yang dibuang dikurangi dengan hasil sebenarnya.

Teknik yang banyak digunakan untuk mengkompensasi kesalahan ini adalah koreksi konstan dan koreksi variabel. Dalam metode koreksi konstan, nilai yang telah dihitung sebelumnya yang ditentukan oleh kesalahan reduksi dan kesalahan pembulatan ditambahkan ke matriks produk parsial. Sedangkan dalam koreksi variabel, kompensasi kesalahan yang dinamis dan akurat dilakukan, yaitu, koreksi variabel digunakan sebagai pengganti nilai konstan. Memilih antara kedua metode ini bergantung pada batasan desain pengali sesuai dengan kebutuhan.

Tujuan utama dari pengali ini adalah untuk mengurangi disipasi daya, luas area, dan penundaan. Schulte dkk. melakukan simulasi dengan operan 16-bit dan 32-bit untuk pengali paralel terpotong dan menyatakan pengamatan bahwa mereka masing-masing mengurangi disipasi daya sebesar 20% dan 40% dibandingkan dengan rekan-rekan standarnya. Pengurangan jumlah gerbang AND dan adder karena berkurangnya jumlah produk parsial menyebabkan pengurangan ukuran fisik rangkaian, dan pengurangan kompleksitas perangkat keras ini juga menyebabkan pengurangan penundaan, sehingga meningkatkan kinerja sistem secara keseluruhan.

Namun terlepas dari efisiensinya, ia memiliki serangkaian kekurangan yang cukup signifikan, termasuk pengenalan kesalahan yang signifikan yang mengganggu akurasi. Bahkan dengan skema kompensasi kesalahan yang dapat digunakan untuk meningkatkan akurasi, skema tersebut memiliki kekurangan tambahan. Koreksi variabel akan membutuhkan perangkat keras tambahan, dan koreksi konstan akan sedikit mengimbangi penghematan area yang diperoleh dari pemotongan.

**Pengali Logaritmik: Algoritma Mitchell**

Algoritma untuk pengali logaritmik yang diusulkan oleh John Mitchell pada tahun 1962 menawarkan alternatif yang menarik untuk pengali tradisional dengan mengubah operasi perkalian yang kompleks menjadi operasi penjumlahan logaritmik yang sederhana. Prinsip dasar metode ini adalah bahwa hasil perkalian dua bilangan dapat dihitung dengan menambahkan logaritmanya. Hasil perkalian tersebut dapat direpresentasikan sebagai

$$P = A \times B = 2^{\log 2A + \log 2B}$$

Meskipun konversi logaritmik dan anti-logaritmik yang tepat itu kompleks, algoritma Mitchell memperkenalkan logika digital sederhana untuk melakukan konversi tersebut dengan sedikit perkiraan. Implementasi algoritma Mitchell mengikuti prinsip tiga langkah: konversi logaritma perkiraan, penjumlahan, dan konversi anti-logaritma. Proses ini menggantikan pengali tradisional dengan kombinasi detektor satu terdepan (LOD) / pengkode prioritas, penambah, dan penggeser, menghasilkan rangkaian yang lebih sederhana dan efisien. Kim dkk. menyatakan bahwa dengan menggunakan operan 32-bit, pengali logaritmik menghemat hingga 88% energi dibandingkan dengan pengali tepat.

Algoritma ini memungkinkan skalabilitas dan kustomisasi yang mudah untuk lebar bit operan yang berbeda. Algoritma Mitchell memiliki kompromi mendasar antara akurasi inheren, bias kesalahan, dan kesulitan dengan bilangan bertanda. Algoritma ini menghasilkan kesalahan komputasi yang relatif tinggi karena perkiraan linier selalu kurang dari atau sama dengan nilai sebenarnya. Pilipovi´c et al. menyatakan bahwa kesalahan relatif terburuk adalah 11,1%, tetapi terlalu tinggi untuk implementasi. Karena algoritma ini dirancang untuk bilangan tak bertanda, maka tidak berlaku untuk bilangan bertanda. Namun terlepas dari semua kekurangannya, algoritma Michelle tetap menjadi kandidat yang sangat baik untuk sistem yang dibatasi energi dan toleran terhadap kesalahan.

**Pengali Tak Berbias Rentang Dinamis**

Pengali Tak Berbias Rentang Dinamis (DRUM) adalah pendekatan yang dirancang untuk mengontrol sifat kesalahan aproksimasi, sehingga sangat cocok untuk aplikasi seperti pemrosesan sinyal digital, jaringan saraf, dan otomatisasi komputasi. Tujuan utamanya adalah untuk menciptakan pengali aproksimasi dengan kesalahan rata-rata nol. Keuntungan mendasar dari pendekatan ini adalah bahwa, ketika satu komputasi sedang berlangsung, kesalahan dalam komputasi sebelumnya akan diperbaiki.

Tingkat aproksimasi dapat disesuaikan sesuai dengan kebutuhan aplikasi, sehingga keseimbangan antara akurasi dan aspek perangkat keras dapat diputuskan. Metode ini

menghilangkan pohon penjumlahan yang kompleks untuk kolom yang paling tidak signifikan, yang akan secara drastis mengurangi jumlah komponen, sehingga menghasilkan propagasi penundaan dan konsumsi daya yang lebih rendah. Arsitektur DRUM memberikan banyak keuntungan, seperti kesalahan yang tidak bias, pengurangan penundaan propagasi, efisiensi energi, pengurangan luas area, skalabilitas, dan fleksibilitas.

**Tabel 12.2** Menyajikan perbandingan pengali perkiraan yang telah dibahas di atas, yaitu pengali yang toleran terhadap kesalahan, pengali terpotong, dan pengali logaritmik.

| Jenis Pengali | Performa/Kecepatan | Delay | Luas Area | Aproksimasi/Akurasi | Referensi |
|---|---|---|---|---|---|
| Pengali toleran kesalahan | Sangat tinggi | Paling rendah | Paling kecil | Tingkat kesalahan tinggi sehingga dapat digunakan untuk aplikasi yang toleran terhadap kesalahan | [20, 21, 27] |
| Pengali terpotong | Sedang | Rendah | Berkurang | Akurasi lebih rendah dibandingkan pengali eksak, tetapi lebih baik daripada EMT | [22, 23, 28] |
| Pengali logaritmik (Algoritma Mitchell) | Tinggi untuk DSP dan ML | Sangat rendah | Kecil | Umumnya memiliki tingkat kesalahan sekitar **5%–10%** | [24, 25, 29] |

Pengurangan pohon penjumlahan memungkinkan pengoperasian pengali pada kecepatan clock yang lebih tinggi, sehingga menghasilkan kinerja sistem yang lebih cepat secara keseluruhan. Memastikan kesalahan rata-rata nol menjadikannya pilihan yang andal. Arsitektur DRUM menyediakan solusi canggih dalam lanskap komputasi perkiraan, yang terbukti sangat bermanfaat dalam aplikasi yang berfokus pada ML dan kecerdasan buatan (Tabel 12.2).

## 12.5 KESALAHAN DAN SENSITIVITAS SEBAGAI METRIK KINERJA

Pendekatan yang berhasil dalam arsitektur ML hanya dapat dicapai dengan analisis yang cermat terhadap kesalahan dan sensitivitas sistem. Dengan analisis dan pendekatan yang tepat, dimungkinkan untuk menyeimbangkan akurasi, efisiensi energi, kecepatan, dan luas area, membuka jalan bagi perancangan sistem AI generasi berikutnya yang berteknologi tinggi.

**Memahami Kesalahan**

Komputasi perkiraan dapat dievaluasi menggunakan berbagai metrik kesalahan yang dicirikan oleh sifat dan besarnya ketidakakuratan. Mari kita bahas Tingkat Kesalahan (ER), Jarak Kesalahan (ED), Jarak Kesalahan Rata-rata (MED), dan Kesalahan Kuadrat Rata-rata (MSE) sebagai metrik utama dalam bab ini. ER adalah probabilitas menghasilkan kesalahan dalam output, dan ED adalah perbedaan aritmatika antara output yang tepat dan output yang diperkirakan. Demikian pula, MED dan MSE akan menjadi rata-rata besaran kesalahan di semua kemungkinan input. Sebagaimana akurasi bergantung pada desain, akurasi juga sangat bergantung pada data masukan. Dalam banyak kasus, kesalahan terjadi karena kurangnya hubungan yang dapat diprediksi atau jelas antara masukan dan keluaran (Tabel 12.3).

**Tabel 12.3** membahas metrik kesalahan berdasarkan aplikasi ML untuk berbagai teknik pengali perkiraan.

| Teknik Pengali | Tingkat Kesalahan (%) | Rata-rata Jarak Kesalahan | Kesesuaian Model ML |
|---|---|---|---|
| Pengali toleran kesalahan | Tinggi | Sedang | Dapat diterapkan karena kesalahan lebih seimbang |
| Pengali terpotong – arsitektur Wallace | Tinggi | Bervariasi, tergantung pada konfigurasi | Tidak cocok untuk model ML |
| Pengali terpotong – arsitektur array | Tinggi | Rendah pada beberapa konfigurasi | Sebaiknya diterapkan secara bersyarat |
| Pengali Booth terpotong | Sangat tinggi | Tinggi | Tidak cocok untuk model ML |
| Pengali logaritmik | Sangat tinggi | Sangat tinggi | Bergantung pada aplikasi model ML |
| Pengali tanpa bias rentang dinamis | Tinggi | Rendah | Sangat cocok untuk CNN dan DNN |

**Strategi untuk Menerapkan Aproksimasi Secara Selektif**

Menerapkan strategi yang tepat pada lapisan jaringan saraf yang tepat untuk aproksimasi adalah strategi paling mendasar untuk menerapkan aproksimasi. Misalnya, level yang berbeda dari DNN akan melakukan fungsi yang berbeda, dan karenanya mereka akan memiliki tingkat toleransi yang berbeda terhadap ketidaktepatan. Beberapa pembahasan telah menunjukkan bahwa lapisan awal jaringan saraf akan lebih sensitif terhadap kesalahan, dan seiring kita masuk ke lapisan yang lebih dalam, sensitivitasnya menjadi berkurang. Pengamatan ini mengarah pada strategi aproksimasi selektif per level yang memfasilitasi perancang untuk mendapatkan output yang sangat akurat.

Pengali presisi penuh akan diimplementasikan pada lapisan awal, diikuti oleh pengali aproksimasi yang efisien dan sesuai lapisan pada lapisan yang toleran terhadap kesalahan. Misalnya, suatu sistem dapat menggunakan pengali Booth Radix-4 untuk beberapa lapisan pertama dan kemudian beralih ke pengali perkiraan terpotong atau logaritmik sesuai dengan toleransi lapisan untuk lapisan klasifikasi akhir guna menghemat daya dan area serta meningkatkan efisiensi perangkat keras.

Strategi ampuh berikutnya adalah perkiraan sadar pelatihan yang memungkinkan kompensasi kesalahan selama proses pelatihan. Proses ini melibatkan kuantisasi bobot model dan bekerja dengan data presisi rendah selama pelatihan. Teknik canggih dapat digunakan sehingga pola kesalahan dapat dikenali dan diminimalkan melalui backpropagation, dan menghasilkan output akhir yang efektif. Proses pelatihan ulang ini dapat ditemukan lebih efektif jika diterapkan bersama dengan model yang telah dilatih sebelumnya.

Strategi canggih lainnya yang dapat digunakan adalah menerapkan perkiraan berdasarkan data input runtime atau kondisi operasi lainnya. Perkiraan semacam ini akan sangat bermanfaat dalam aplikasi AI edge seperti kendaraan otonom dan drone. Sistem yang

umumnya menerapkan strategi ini menggunakan unit MAC presisi tunggal yang dapat disesuaikan atau beralih secara dinamis antara pengali yang berbeda. Unit MAC khusus dapat digunakan yang akan melewatkan perkalian jika nol terdeteksi pada input, sehingga mengurangi penundaan dan mempercepat proses.

## 12.6 PERTUKARAN DESAIN DALAM KOMPUTASI APROKSIMASI

Tujuan utama penggunaan komputasi aproksimasi adalah untuk secara sengaja mengorbankan presisi demi peningkatan metrik PPA, yaitu Daya, Kinerja, dan Luas. Aproksimasi yang sedikit dapat menciptakan penyimpangan drastis dalam kualitas output, sementara aproksimasi yang agresif dapat memberikan angka yang sangat baik dalam metrik PPA.

**Akurasi Vs Energi, Luas, dan Latensi**

Semua perangkat AI edge modern melibatkan miliaran parameter, sehingga pelatihan dan penerapannya sangat intensif energi, lambat, dan mahal. Bidang-bidang yang membebani seperti kendaraan otonom, sensor IoT, dan monitor perawatan kesehatan yang dapat dikenakan harus melakukan tugas-tugas kompleks yang dikondisikan di bawah batasan daya dan kendala latensi. Dalam kasus ini, mengorbankan sejumlah kecil presisi sebagai imbalan untuk peningkatan kecepatan pemrosesan, luas chip, dan efisiensi energi dapat meningkatkan akselerasi perangkat keras ke tingkat yang dibutuhkan sesuai dengan aplikasi melalui komputasi aproksimasi. Hal ini membawa kita pada pertanyaan sentral yang telah kita bahas sepanjang bab ini: ‘seberapa banyak akurasi yang harus kita korbankan agar sistem kita bekerja lebih cepat, lebih kecil ukurannya, dan lebih hemat daya?’

Karena unit MAC adalah jantung dari jaringan saraf, kita dapat mengatakan bahwa unit ini adalah inti dari pertukaran ini. Penyederhanaan logika unit komputasi dengan mengurangi jumlah transistor dan aktivitas switching dapat menurunkan daya per operasi, memberikan keuntungan yang mengubah permainan dalam aplikasi yang membutuhkan banyak energi seperti aplikasi berbasis ruang angkasa atau drone, di mana pembelajaran mendalam diimplementasikan. Menggunakan strategi paralelisme dan mengemas banyak unit MAC ke dalam satu chip mengurangi ukuran sirkuit dan dengan demikian mengurangi biaya produksi. Latensi adalah ciri utama yang harus diperhatikan dalam aplikasi, khususnya dalam aplikasi seperti kendaraan otonom, di mana penundaan sepersekian detik pun dapat menyebabkan dampak besar. Penundaan ini dapat dipersingkat dengan mempercepat kecepatan clock dan mengurangi waktu komputasi.

**Optimasi Spesifik Aplikasi**

Penerapan komputasi aproksimasi yang sukses bukanlah penerapan akurasi secara seragam, tetapi membuka potensi sebenarnya dari komputasi aproksimasi melalui optimasi spesifik beban kerja yang tepat dengan menyesuaikan derajat dan jenis aproksimasi sesuai dengan model ML dan komputasi yang mendasarinya. Seperti yang kita ketahui, DNN terdiri dari operasi kompleks seperti perkalian matriks dan konvolusi, analisis rinci dari rangkaian dapat mengungkapkan beban kerjanya di berbagai lapisan dan tugas komputasi. Perlu dicatat, komputasi kompleks dalam DNN lebih sensitif pada adder daripada pada multiplier; oleh

karena itu, adder juga harus diperlakukan dengan sangat hati-hati untuk menjaga integritas komputasi secara keseluruhan. Hal ini membawa kita pada kesimpulan bahwa strategi komputasi aproksimasi tidak dapat bersifat generik; strategi tersebut harus berkaitan dengan arsitektur sistem yang dianalisis dan perspektif aplikasi. Pendekatan ini menggabungkan analisis offline dengan manajemen runtime dinamis untuk mengoptimalkan kinerja terkait dengan persyaratan kualitas aplikasi, menggunakan sistem SAGE (*Self-tuning Approximation for Graphics Engine*), yang awalnya dirancang untuk grafis.

Kerangka kerja ini beroperasi dalam dua fase utama: pembangkitan kernel offline dan manajemen kualitas runtime. Pembangkitan kernel offline bertujuan untuk secara otomatis menghasilkan operasi rutin komputasi inti seperti perkalian matriks dan konvolusi, dan manajemen kualitas runtime memilih kernel yang dibutuhkan dari pustaka yang telah dibuat sebelumnya. Menggunakan teknik ini akan membuat aproksimasi menjadi proses transparan dalam ML, membantu dalam desain perangkat keras aproksimasi.

**Dampak Penskalaan Teknologi dan Variabilitas Perangkat**

Seiring dengan menyusutnya ukuran transistor hingga skala nanometer, efek kuantum akan berperan bersama dengan ketidaksempurnaan manufaktur, sehingga membuatnya tidak dapat diandalkan. Secara tradisional, hal ini diatasi dengan implementasi guardband waktu pesimistis yang memaksa chip untuk berjalan pada kecepatan clock paling lambat, yaitu, dengan skenario terburuk transistor. Pendekatan ini mungkin menjamin keandalan tetapi disertai dengan kerugian yang signifikan dalam kinerja dan energi. Dalam kasus ini, kita dapat melihat komputasi perkiraan sebagai solusi yang memanfaatkan akurasi dan mencari peningkatan daya, kinerja, dan luas area. Komputasi perkiraan mengubah dampak penskalaan teknologi sebagai krisis keandalan menjadi batas baru untuk optimasi. Efek negatif dari pendekatan ini dapat diatasi dengan mengelola secara cermat pertukaran antara akurasi dan efisiensi, memungkinkan teknologi yang lebih cepat, kuat, dan hemat energi meskipun ada keterbatasan silikon.

## 12.7 TANTANGAN DAN KETERBATASAN KOMPUTASI APROKSIMASI

Tantangan paling signifikan dari komputasi aproksimasi adalah ketidakpastian dan sensitivitasnya terhadap kesalahan komputasi, karena tingkat yang berbeda memiliki sifat toleransi kesalahan yang berbeda. Seperti yang telah kita bahas, lapisan awal akan selalu menunjukkan sensitivitas yang lebih tinggi daripada lapisan yang lebih dalam. Selain itu, sirkuit aproksimasi memperkenalkan bias kesalahan sistematis yang sangat mengganggu operasi kompleks seperti perkalian matriks dan konvolusi.

Bahkan kesalahan kecil dapat memengaruhi banyak operasi dan menghasilkan penyimpangan yang signifikan dari keluaran akhir yang diharapkan. Untuk memprediksi kualitas tingkat aplikasi untuk aproksimasi dan implementasi perangkat keras, diperlukan simulasi yang ekstensif dan memakan waktu bahkan untuk aproksimasi tingkat rendah. Komputasi aproksimasi untuk ML sangat bergantung pada desain bersama perangkat keras-perangkat lunak, terutama dalam teknik seperti aproksimasi sadar pelatihan. Dalam model ini, kesalahan yang disebabkan oleh perangkat keras diidentifikasi dan diperbaiki oleh perangkat

lunak. Akselerator AI yang dirancang untuk satu aplikasi spesifik tidak dapat bekerja dengan efisiensi yang sama di aplikasi lain.

Evaluasi desain komputer perkiraan tidak berakhir dengan analisis metrik PPA; hal itu bergantung pada kualitas output sesuai dengan persyaratan aplikasi. Karena tidak ada metrik atau tolok ukur standar untuk analisis kinerja tugasnya, sulit untuk mengevaluasi dan membandingkan berbagai teknik komputasi perkiraan. Tingkat perkiraan yang dapat diterima bervariasi dengan batasan aplikasi. Misalnya, toleransi kesalahan untuk kendaraan otonom atau pencitraan medis harus hampir nol, oleh karena itu perkiraan tidak dapat agresif.

Oleh karena itu, kita dapat mengatakan bahwa tidak ada solusi universal, karena setiap arsitektur ML perkiraan akan disesuaikan secara eksklusif untuk setiap aplikasi. Meskipun komputasi perkiraan memiliki kemungkinan yang sangat besar untuk masa depan desain perangkat keras ML, mengatasi tantangannya, yaitu sensitivitas kesalahan, kebutuhan akan otomatisasi yang kompleks, beban desain bersama perangkat keras dan perangkat lunak, dan persyaratan kualitas spesifik aplikasi, sangat penting untuk membuka potensi penuh perkiraan.

## 12.8 ARAH PENGEMBANGAN KOMPUTASI APROKSIMASI

Seiring dengan semakin matangnya bidang komputasi aproksimasi dan berkembangnya melampaui tahap pembuktian konsep menuju sistem yang dapat diterapkan, fokusnya akan lebih pada peningkatan efisiensi sistem secara keseluruhan daripada pengali aproksimasi. Penciptaan sistem yang sadar diri dan adaptif dapat secara dinamis mengelola tingkat aproksimasi selama runtime berdasarkan data input atau lingkungan operasi, sehingga menghasilkan implementasi aproksimasi yang tepat melalui fleksibilitas perangkat lunak untuk properti tetap perangkat keras. Penggunaan model ML sekunder yang didedikasikan untuk memprediksi dan mengoreksi kesalahan yang disebabkan oleh perangkat keras aproksimasi utama merupakan area pembahasan yang menarik.

Perangkat keras yang dapat dikonfigurasi saat runtime adalah pengembangan unit MAC yang dapat diskalakan presisi yang mengubah akurasi sistem ML dengan beralih antara presisi tinggi ke presisi rendah dan mode daya tinggi ke mode daya rendah sesuai dengan sensitivitas operasi. Selain itu, adaptasi berbasis data optimal yang bekerja bergantung pada analisis data input. Hal ini paling dibutuhkan dalam aplikasi seperti kendaraan otonom, di mana presisi penuh akan dibutuhkan di lingkungan perkotaan, dan aproksimasi tinggi dapat digunakan di lingkungan jalan raya yang kosong.

Masa depan komputasi aproksimasi bukan hanya tentang otomatisasi dan adaptasi tetapi juga tentang merancang teknik aproksimasi untuk menyesuaikan karakteristik beban kerja. Merancang model ML aproksimasi, terutama berfokus pada bias dan besaran kesalahan . Teknik seperti pengali logaritmik dan pemotongan merupakan jalan yang menjanjikan untuk mengeksplorasi dan mengoptimalkan desain. Jadi, unit MAC akan menjadi kombinasi terintegrasi dari aproksimasi dan optimasi dengan menggunakan pendekatan baru seperti skema pengkodean data dan sparsitas untuk mengurangi kompleksitas komputasi.

## 12.9 REVOLUSI PERANGKAT KERAS AI BERBASIS KOMPUTASI APROKSIMASI

Komputasi perkiraan merupakan landasan bagi masa depan akselerasi ML karena kemampuannya menawarkan solusi ampuh untuk tantangan konsumsi energi, luas area, dan biaya yang semakin meningkat. Unit MAC, sebagai jantung perangkat keras ML, dipahami sebagai tempat paling strategis untuk mengimplementasikan perkiraan. Kami telah mengeksplorasi berbagai teknik dalam bab ini, mulai dari strategi perangkat lunak tingkat tinggi hingga inovasi perangkat keras tingkat rendah. Metode-metode ini bukan hanya pembahasan teoretis; kami juga telah membahas implementasinya dalam TPU dan NPU modern, mencapai kinerja yang belum pernah terjadi sebelumnya.

Lebih lanjut, komputasi perkiraan mengurangi hambatan komputasi dalam penggunaan ML dengan memfasilitasi penerapannya di lingkungan yang terbatas sumber daya. Ini melibatkan penggunaan aplikasi mulai dari diagnostik medis pada perangkat AI edge hingga memungkinkan pertanian presisi di daerah dengan infrastruktur yang buruk. Komputasi perkiraan bukannya tanpa kekurangan. Tanpa pendekatan yang tepat, sistematis, dan analitis, kesalahan penerapan perkiraan dapat menyebabkan penurunan akurasi model yang sangat buruk. Oleh karena itu, identifikasi yang cermat terhadap unit dan subunit komputasi non-kritis memainkan peran penting dalam mengimplementasikan dan menerapkan model ML berkualitas apa pun.

Sebagai kesimpulan, komputasi aproksimasi bukan hanya pendekatan optimasi teknis; ini mewakili pergeseran mendasar dalam filosofi desain perangkat keras. Ini memungkinkan penskalaan dan keberlanjutan ML, menyediakan kerangka kerja penting untuk penerapan kecerdasan yang lebih besar dengan kendala praktis dunia fisik.

# BAB 13
# EKOSISTEM HARDWARE AI MELAMPAUI GPU DAN TPU

## 13.1 APLIKASI KECERDASAN BUATAN DAN KEBUTUHAN PERANGKAT KERAS

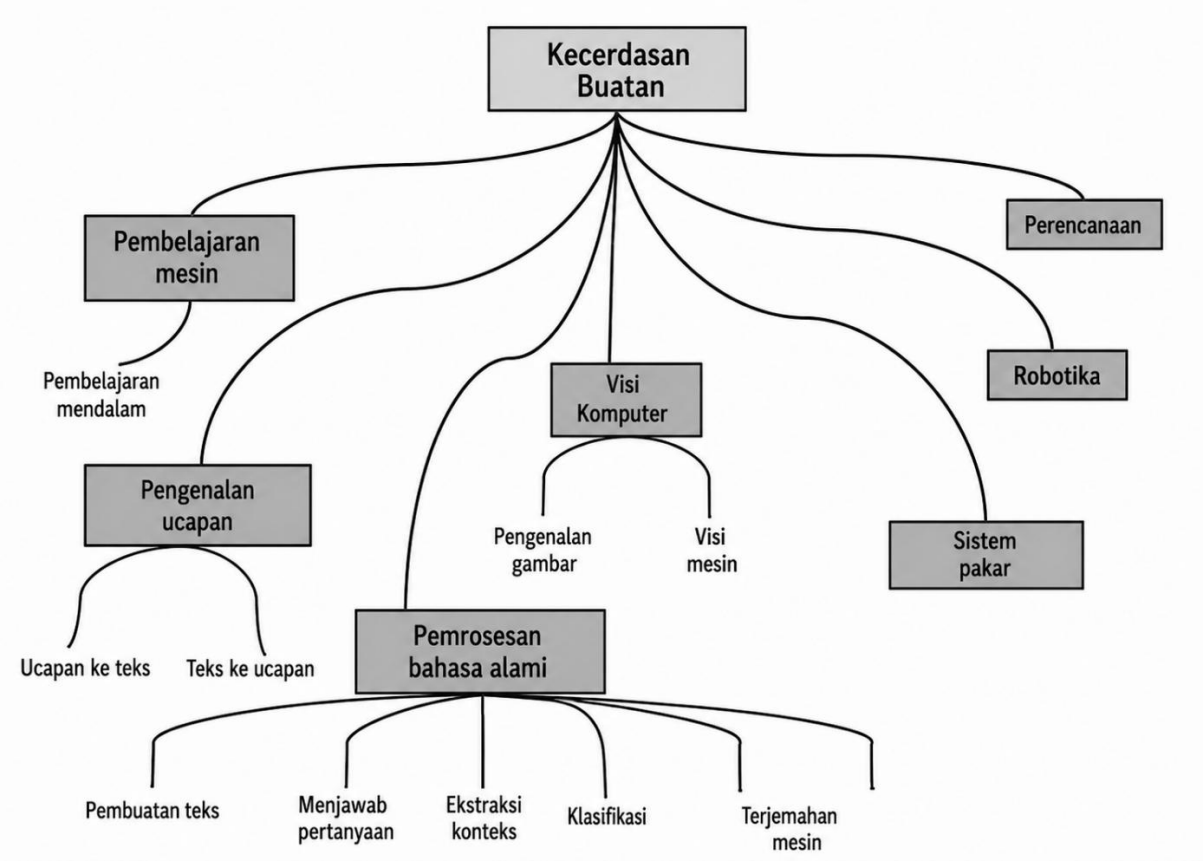
Kecerdasan Buatan (AI) dan Pembelajaran Mesin (ML) menjadi sangat penting di dunia saat ini dengan mengubah cara manusia berinteraksi dengan teknologi. AI adalah teknologi yang memungkinkan sistem komputer untuk melakukan tugas-tugas yang umumnya terkait dengan kecerdasan manusia. ML mengembangkan algoritma untuk sistem komputer yang belajar dari data dan meningkatkan kinerjanya tanpa pemrograman eksplisit.

Algoritma AI banyak digunakan dalam industri perawatan kesehatan untuk meningkatkan kecepatan dan akurasi pencitraan dan diagnostik medis, atau untuk menyarankan rencana perawatan yang dipersonalisasi. Chatbot digunakan dalam perawatan kesehatan untuk membantu pasien dalam memesan janji temu atau melakukan tindak lanjut. Penggunaan otomatisasi bertenaga AI di sektor keuangan mengurangi kesalahan dan beban kerja manual. Model ML digunakan untuk mendeteksi aktivitas penipuan dan untuk menganalisis serta mengelola risiko. Industri hiburan memanfaatkan teknologi AI melalui rekomendasi yang dipersonalisasi dan pembuatan konten.

Dengan menggunakan ML tingkat lanjut, mobil otonom siap mengubah transportasi dengan membuatnya lebih aman dan efisien. Teknologi AI dan ML meningkatkan proses yang ada dan menciptakan peluang serta aplikasi baru yang dulunya dianggap tidak mungkin. Gambar 13.1 menunjukkan ekosistem AI, yang mencantumkan berbagai aplikasi AI. Kinerja aplikasi AI dan ML sangat bergantung pada efisiensi perangkat keras yang mendasarinya. Dengan algoritma AI yang kompleks dan kumpulan data yang luas, daya komputasi juga meningkat. Hal ini telah memotivasi desain dan pengembangan prosesor khusus, sistem memori, dan interkoneksi yang mampu menangani beban kerja AI. Unit Pemrosesan Grafis (GPU) dan Unit Pemrosesan Tensor (TPU) adalah beberapa prosesor khusus yang menjadi penting untuk mempercepat tugas ML. NVIDIA, pemimpin dalam GPU yang dioptimalkan untuk AI, mengumumkan pengiriman NVIDIA® GH200 Grace Hopper Superchip untuk beban kerja AI kompleks pada kuartal kedua tahun 2023. NVIDIA merilis GPU NVIDIA® L40S baru pada tahun 2023, yang merupakan prosesor pusat data universal yang dirancang untuk mempercepat aplikasi yang membutuhkan komputasi intensif.

Sirkuit Terpadu Khusus Aplikasi (ASIC) dan *Field-Programmable Gate Array* (FPGA) sedang dikustomisasi untuk tugas-tugas AI, menawarkan kinerja tinggi dan efisiensi energi. Pasar FPGA bernilai Rp 121 triliun pada tahun 2024 dan diperkirakan akan berkembang menjadi Rp 258 triliun pada tahun 2029, dengan Tingkat Pertumbuhan Tahunan Majemuk (CAGR) yang mengejutkan sebesar 16,4%. Ekspansi pasar FPGA telah dipengaruhi oleh adopsi AI dalam aplikasi waktu nyata. Arsitektur perangkat keras baru dikembangkan untuk meningkatkan kinerja sistem AI dan juga untuk menerapkan model dan aplikasi AI canggih. Misalnya, mengintegrasikan chip AI ke dalam perangkat komputasi tepi memungkinkan kemampuan pengambilan keputusan cerdas pada ponsel pintar, perangkat *Internet of Things*

(IoT), dan sistem tertanam. WSE-2 dari Cerebras Systems, Tensor Streaming Processor dari Groq, Envise dari Lightmatter, Grace dari NVIDIA, ARM Cortex M55, Qualcomm Cloud AI 100 adalah beberapa chip AI yang diperkenalkan oleh produsen chip utama. Tabel 13.1 merinci ukuran berbagai jenis perangkat keras di pasar.




**Gambar 13.1** Ekosistem AI: bidang dan aplikasi utama.

**Tabel 13.1** Pasar AI dalam perangkat keras berdasarkan jenis, 2020–2023

| ***Jenis*** | ***2020*** | ***2021*** | ***2022*** | ***2023*** |
|---|---|---|---|---|
| Prosesor | 15.74 | 19.54 | 24.24 | 30.06 |
| Memori | 3.36 | 4.18 | 5.21 | 6.47 |
| Jaringan | 5.56 | 6.98 | 8.75 | 10.98 |
| Penyimpanan | 2.96 | 3.80 | 4.86 | 6.21 |

## 13.2 PERAN CPU DALAM MENGGERAKKAN AI DAN ML

Unit pemrosesan pusat (CPU) adalah jantung komputasi, yang mengimplementasikan berbagai aplikasi dengan mengeksekusi instruksi dan melakukan tugas-tugas tujuan umum. Sebagai mesin komputasi utama di sebagian besar sistem komputer, CPU mengeksekusi algoritma dan perhitungan kompleks untuk mengimplementasikan aplikasi AI dan ML. CPU sangat penting untuk tugas-tugas yang membutuhkan pemrosesan sekuensial, operasi kondisional, dan pengambilan keputusan yang kompleks, yang umum dalam banyak algoritma AI dan ML seperti pohon keputusan, regresi linier, atau mesin vektor pendukung. CPU juga

sangat cocok untuk pra-pemrosesan data, ekstraksi fitur, dan pelatihan model pada dataset yang lebih kecil.

Selain pra-pemrosesan dan pelatihan, CPU juga diperlukan dalam inferensi AI. Inferensi AI mengacu pada pembuatan prediksi pada input baru dengan model AI yang terlatih. Pelatihan AI adalah tugas yang intensif secara komputasi yang biasanya membutuhkan kemampuan pemrosesan paralel yang masif, tetapi inferensi AI dapat dijalankan secara efisien pada CPU. Manfaat menggunakan CPU untuk inferensi AI adalah:

- CPU lebih hemat biaya.
- CPU tersedia di semua lingkungan komputasi, seperti komputer pribadi, ponsel pintar, pusat data, server cloud, dll.
- Pustaka perangkat lunak canggih yang tersedia di CPU modern meningkatkan kemampuan CPU untuk aplikasi AI. Misalnya, CPU Intel menunjukkan kinerja tinggi dengan oneDNN dan toolkit OpenVino.

Permintaan akan CPU berkinerja tinggi juga meningkat seiring dengan meningkatnya kompleksitas beban kerja AI dan ML. CPU yang mampu menjalankan banyak thread dan CPU multi-core diperkenalkan untuk kinerja yang lebih unggul. Multithreading memungkinkan CPU untuk mengelola tugas secara efisien seperti pemrosesan data paralel, eksekusi simultan dari berbagai tahapan algoritma, atau menjalankan beberapa model secara bersamaan. CPU modern memiliki banyak core yang memungkinkan eksekusi thread secara paralel.

Multithreading menguntungkan dalam aplikasi intensif data, di mana kumpulan data besar diproses untuk ML. Hyper-threading adalah teknologi yang tersedia di CPU modern yang memungkinkan setiap core fisik untuk mengeksekusi banyak thread secara bersamaan. CPU modern menggunakan instruksi dan optimasi khusus AI dan ML. Untuk akses memori yang lebih cepat, CPU modern menggunakan kecepatan clock yang tinggi dan juga cache yang besar. Akses memori berkecepatan tinggi membuat CPU modern cocok untuk aplikasi yang melibatkan pengambilan keputusan cepat. Fitur CPU untuk AI dibandingkan dengan CPU tujuan umum dan dirinci dalam Tabel 13.2.

Kelemahan utama CPU dalam komputasi jaringan saraf adalah tuntutan pemrosesan paralel dari jaringan ini. Komputasi jaringan saraf melibatkan perkalian matriks besar, konvolusi, dan operasi aljabar linier lainnya yang mungkin memerlukan pemrosesan paralel. Meskipun CPU modern mampu menangani banyak thread dan memiliki banyak core, kemampuannya untuk melakukan pemrosesan paralel terbatas. CPU dibangun terutama untuk komputasi tujuan umum yang dapat menangani berbagai instruksi dan karenanya membuatnya kurang efisien untuk tugas-tugas khusus seperti pemrosesan paralel yang dibutuhkan dalam pelatihan jaringan saraf. Batasan lain dari CPU dalam pemrosesan jaringan saraf adalah bandwidth memori dan latensinya. Jaringan saraf membutuhkan transfer data yang signifikan antara memori dan unit pemrosesan. CPU tidak menunjukkan kinerja transfer data yang efisien dan memperlambat pelatihan jaringan saraf. CPU menunjukkan masalah konsumsi daya saat menangani dataset yang lebih besar dan tugas-tugas yang sangat paralel.

**Tabel 13.2** Perbandingan CPU untuk komputasi umum dan komputasi AI

| Atribut | CPU Khusus AI | CPU Serbaguna |
|---|---|---|
| Ekstensi Vektor | Vektor 512-bit | Vektor 256-bit |
| Presisi | Mendukung float16 dan int8 | Floating point 32-bit dan 64-bit |
| Bandwidth Memori | >500 GB/s dengan HBM | <100 GB/s dengan Double Data Rate (DDR) |
| Inti Khusus AI | Akselerator AI untuk komputasi tensor | Tidak ada |
| Interkoneksi | Kepadatan bandwidth tinggi pada chip (on-die) | Bandwidth sedang pada chip (on-die) |
| Set Instruksi | Instruksi AI khusus untuk jaringan saraf | Arsitektur Set Instruksi Serbaguna (Instruction Set Architecture/ISA) |
| Kemampuan Rekonfigurasi | Jalur data dapat dikonfigurasi | Pipeline tetap |
| Caching | Memori cache privat berkapasitas besar | Sistem cache bersama dengan kapasitas lebih kecil |
| Pemrograman | Pustaka yang dioptimalkan untuk mendukung framework AI | Bahasa pemrograman tradisional untuk tujuan umum |

Namun, ketika menangani tugas yang tidak melibatkan paralelisme tinggi, CPU memainkan peran utama dalam mengelola alur kerja keseluruhan mulai dari pra-pemrosesan data hingga penerapan model. Dalam sistem hibrida, di mana CPU digunakan bersama dengan GPU atau akselerator lainnya, kekuatan masing-masing jenis prosesor dimanfaatkan untuk mengoptimalkan kinerja. Beberapa opsi CPU untuk AI/ML adalah:

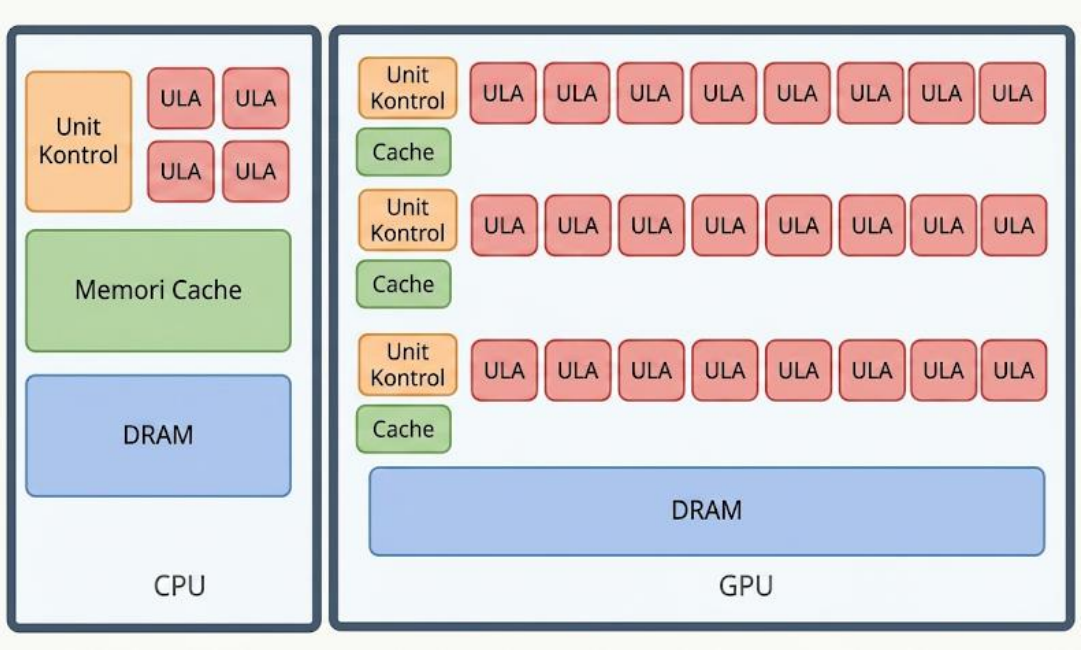
1. Intel Core i9-13900K: Dengan node proses Intel 7 10 nm, 8 core performa tinggi dan 16 core efisien, CPU ini menunjukkan kemampuan pemrosesan AI yang sangat baik. Frekuensi clock dasar adalah 3,0 GHz untuk core performa tinggi dan 2,2 GHz untuk core efisien.
2. AMD Ryzen 9 7950X3D: CPU ini cocok untuk pengembang AI/ML dan memiliki 16 core, serta dikenal dengan teknologi 3D V-Cache-nya. Frekuensi clock dasar adalah 4,2 GHz, dan node prosesnya adalah TSMC 5 nm.
3. Intel Core i7-14700K: CPU yang cocok untuk gamer dan pembuat konten, dan tidak terlalu boros energi, beroperasi dengan frekuensi clock dasar 3,4 GHz untuk core berkinerja tinggi dan 2,5 GHz untuk 12 core efisien.
4. AMD Ryzen 9 7900X: CPU ini memiliki 12 core dan beroperasi pada frekuensi clock dasar 4,7 GHz.

## 13.3 EVOLUSI DAN DAMPAK GPU DALAM AI

CPU dioptimalkan untuk melakukan tugas secara berurutan dengan latensi serendah mungkin. GPU adalah perangkat yang terdiri dari beberapa cluster prosesor dengan beberapa *Streaming Multiprocessor* (SM). GPU dirancang untuk memaksimalkan throughput data dan menangani komputasi paralel secara efisien. GPU dirancang untuk memproses sebanyak

mungkin tugas sekaligus. GPU muncul pada tahun 1980-an sebagai perangkat keras khusus, yang dirancang untuk mempercepat tugas rendering grafis. Gambar 13.2 mengilustrasikan perbedaan arsitektur antara CPU dan GPU.

GPU dirancang sebagai arsitektur paralel masif yang mengintegrasikan banyak unit pemrosesan bersama dengan sistem memori hierarkis. Misalnya, NVIDIA A100 menggabungkan 108 SM, cache level-2 40 MB, dan memberikan bandwidth hingga 2039 GB/s menggunakan memori High Bandwidth Memory-2 (HBM2) 80 GB. Setiap SM memiliki penjadwal instruksi dan pipeline eksekusi sendiri untuk menangani operasi aritmatika dan kontrol, sementara data program dan instruksi diambil dari *Dynamic Random Access Memory* (DRAM) melalui cache L2. Core CUDA adalah unit pemrosesan fundamental dalam GPU NVIDIA, yang bertanggung jawab untuk tugas komputasi paralel tujuan umum, seperti rendering grafis dan melakukan perhitungan kompleks dalam aplikasi ilmiah dan teknik. Core tensor adalah unit pemrosesan khusus yang dirancang untuk pembelajaran mendalam dan komputasi AI. Core ini mempercepat perkalian matriks dan operasi tensor lainnya yang umum digunakan dalam komputasi jaringan saraf. Pada GPU AMD, unit komputasi inti disebut Stream Processor.




**Gambar 13.2** Perbedaan arsitektur antara CPU dan GPU.

GPU untuk AI dirancang untuk unggul dalam kemampuan pemrosesan paralel dan operasi floating-point yang diperlukan untuk komputasi AI. GPU NVIDIA H200, yang dikembangkan pada arsitektur NVIDIA Hopper, untuk mempercepat beban kerja AI, dapat memberikan kinerja FP32 hingga 67 TFLOPS. Peneliti dan pengembang menggunakan kerangka kerja seperti TensorFlow dan PyTorch untuk akselerasi GPU dan untuk dengan mudah mengimplementasikan aplikasi pembelajaran mendalam.

Kinerja GPU juga bergantung pada hierarki memorinya. Sementara CPU menyimpan data yang sering diakses pada cache besar, GPU menggunakan kombinasi memori global, bersama, dan lokal untuk mengelola data secara efisien. Memori global adalah memori terbesar dan paling fleksibel, tetapi juga yang paling lambat. Ini digunakan untuk menyimpan kumpulan data besar yang perlu diproses oleh GPU. Memori bersama lebih cepat tetapi lebih kecil dan digunakan untuk data yang perlu diakses dengan cepat oleh beberapa inti dalam blok pemrosesan yang sama. Register atau memori lokal adalah yang tercepat dan digunakan untuk penyimpanan sementara variabel dalam satu thread.

GPU menunjukkan kemampuan pemrosesan paralel dengan menjalankan beberapa thread secara bersamaan. Warp (NVIDIA) atau wavefront (AMD) adalah sekelompok thread, biasanya 32, yang dieksekusi secara serentak, artinya mereka menjalankan instruksi yang sama secara bersamaan pada data yang berbeda. Pada GPU, fungsi dieksekusi dengan mengelompokkan thread ke dalam blok thread dan kemudian meluncurkan blok-blok ini secara bertahap. Ketika satu thread menunggu instruksi yang bergantung, GPU mengalihkan eksekusi ke thread lain, sehingga mengelola latensi.

Oleh karena itu, untuk mempertahankan utilisasi yang tinggi, lebih banyak thread harus diluncurkan daripada jumlah core yang tersedia. Semua thread dalam satu blok berkomunikasi dan melakukan sinkronisasi ketika setiap blok ditugaskan ke SM selama runtime. Untuk memanfaatkan GPU dengan banyak SM secara maksimal, beberapa blok thread dikirim. Sebuah SM mampu mengeksekusi beberapa blok secara bersamaan, dan oleh karena itu, jumlah blok thread harus melebihi jumlah SM. Metode ini meminimalkan efek "ekor", di mana hanya beberapa blok yang tetap aktif menjelang akhir eksekusi, yang menyebabkan utilisasi GPU yang kurang optimal. GPU menggunakan set instruksi khusus yang dirancang untuk pemrosesan paralel. NVIDIA menggunakan set instruksi CUDA, sedangkan AMD menggunakan arsitektur GCN (*Graphics Core Next*) atau RDNA (*Radeon DNA*).

GPU digunakan dalam komputasi AI untuk pemrosesan simultan dari beberapa operasi karena arsitektur paralelnya yang tinggi. GPU memungkinkan pengembangan algoritma dan model canggih yang mendukung berbagai aplikasi AI. GPU mempercepat pelatihan jaringan saraf. Perkalian matriks terlibat dalam komputasi setiap lapisan jaringan saraf, yang dapat diparalelkan di ribuan inti dalam GPU.

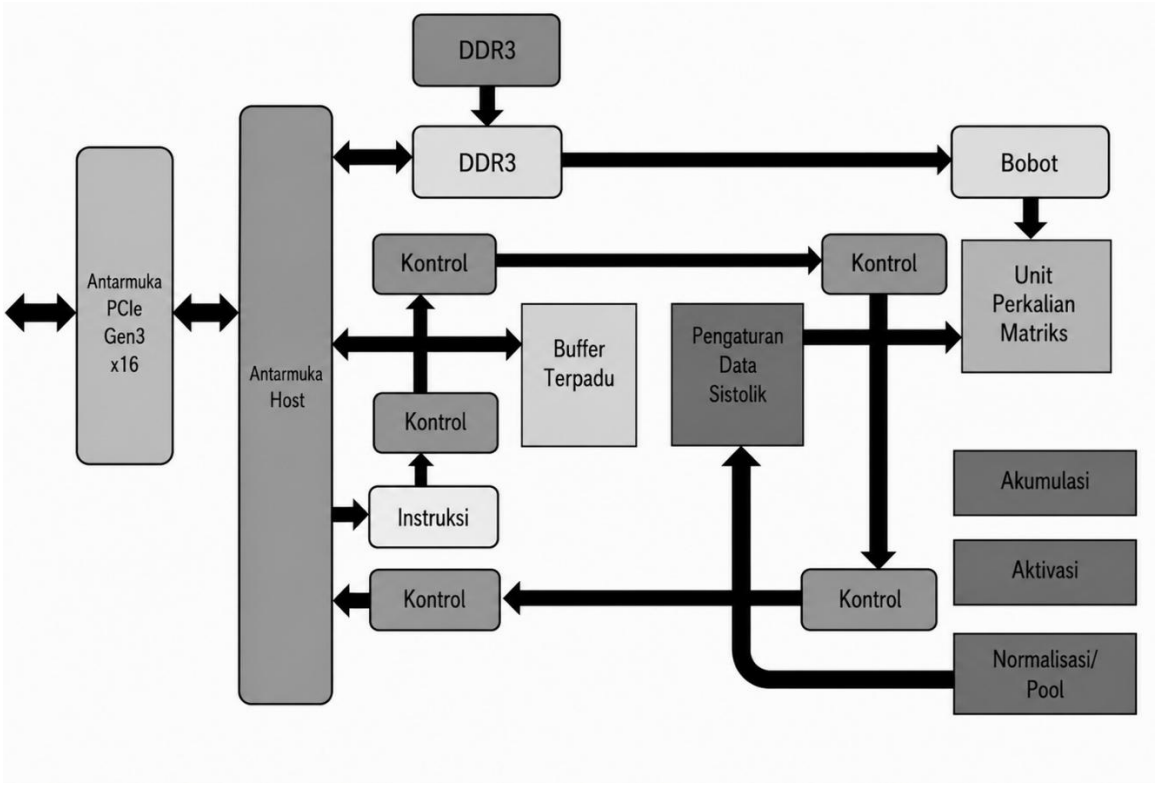
Selama proses maju (*forward pass*) jaringan saraf, GPU menghitung aktivasi untuk setiap lapisan secara simultan, memproses kumpulan input secara paralel. Selama propagasi balik (*back-propagation*), GPU menghitung gradien untuk bobot secara paralel, sehingga mempercepat proses pelatihan dibandingkan dengan komputasi hanya CPU. Misalnya, melatih jaringan saraf konvolusional (CNN) pada dataset gambar lebih cepat pada GPU karena setiap operasi konvolusi dapat dilakukan secara paralel untuk bagian gambar yang berbeda.

## 13.4 TPU UNTUK AI DAN ML

Kemampuan pemrosesan paralel GPU membuatnya menguntungkan dalam komputasi jaringan saraf, tetapi tidak selalu sangat efisien. Oleh karena itu, Google mengembangkan TPU, yang merupakan sirkuit terpadu akselerator AI. TPU mempercepat beban kerja ML, khususnya

yang melibatkan TensorFlow, sebuah pustaka sumber terbuka yang dikembangkan untuk pemrograman ML dan aliran data. TPU menggunakan arsitektur array sistolik untuk operasi matriks berkinerja tinggi dan merupakan pilihan yang lebih baik untuk tugas AI dan ML. Arsitektur fisik TPU ditunjukkan pada Gambar 13.3.

Inti komputasi utama TPU adalah Unit Pengali Matriks (MXU), bersama dengan unit lain seperti Buffer Terpadu (UB) dan Unit Aktivasi (AU). MXU mencakup 65.536 unit perkalian dan penjumlahan 8-bit untuk operasi matriks. Produk 16-bit disimpan dalam 4 mebibyte (MiB) akumulator 32-bit di bawah unit matriks. Keempat MiB ini mewakili 4096 akumulator, masing-masing mampu menampung 256 elemen 32 bit. Dalam setiap siklus, unit matriks menghasilkan jumlah parsial yang terdiri dari 256 elemen. UB adalah buffer on-chip 24 MiB yang menyimpan hasil sementara untuk dimasukkan ke MXU. AU adalah fungsi aktivasi yang terpasang secara permanen. Pengontrol Direct Memory Access (DMA) yang dapat diprogram memungkinkan transfer data antara memori host CPU dan UB. Perangkat keras deteksi dan koreksi kesalahan digunakan dalam memori internal dan eksternal.




**Gambar 13.3** Arsitektur fisik TPU.

Instruksi tingkat tinggi utama yang digunakan dalam pemrograman TPU adalah:

- Read_Host_Memory: Membaca data dari memori.
- Read_Weights: Membaca bobot dari memori.
- Matrix_Multiply: Mengalikan data dan bobot.
- Write_Host_Memory: Menulis hasil ke memori.

TPU beroperasi dengan pertama-tama memuat parameter dari HBM ke MXU. Data kemudian diambil dari HBM. Perkalian kemudian dilakukan, dan setelah setiap proses perkalian, hasilnya diteruskan ke pengali berikutnya sambil secara bersamaan melakukan penjumlahan.

Output akhir dihasilkan sebagai jumlah dari semua perkalian. Instruksi untuk TPU diambil dari host ke buffer instruksi menggunakan komponen periferal interconnect express, bus PCIe Gen3 × 16.

TPU menawarkan beberapa keuntungan yang meliputi:

1. Peningkatan kecepatan komputasi dan efisiensi energi.
2. Pengurangan waktu pelatihan dan inferensi untuk model ML.
3. Kapasitas memori yang ditingkatkan.
4. Dirancang agar mudah diskalakan.

TPU memiliki beberapa kekurangan:

1. TPU sangat mahal dibandingkan dengan CPU dan GPU.
2. TPU hanya mendukung TensorFlow dan tidak mendukung berbagai jenis operasi seperti GPU.

Tabel 13.3 membahas perbedaan antara TPU dan CPU, dan perbandingan TPU dengan GPU diberikan pada Tabel 13.4.

**Tabel 13.3** Perbandingan antara TPU dan CPU

| **TPU** | **CPU** |
|---|---|
| TPU adalah akselerator yang dioptimalkan | CPU adalah prosesor serbaguna |
| Lebih cepat | Lebih lambat |
| TPU mengeksekusi proyek yang dikembangkan menggunakan kerangka kerja TensorFlow | CPU menangani seluruh proses masukan/keluaran (input/output) komputer serta perhitungan logika |
| TPU mempercepat operasi perkalian matriks | CPU memastikan eksekusi program berjalan dengan lancar |

**Tabel 13.4** Perbandingan antara TPU dan GPU

| **TPU** | **GPU** |
|---|---|
| Bekerja lebih cepat dengan menggunakan sumber daya yang lebih sedikit | Bekerja dengan membagi permasalahan yang kompleks menjadi beberapa tugas terpisah |
| TPU memiliki dua inti (core) yang dilengkapi MXU (*Matrix Multiply Unit*) untuk melakukan perhitungan matriks berdensitas tinggi | GPU memiliki ratusan hingga ribuan inti (core) untuk melakukan berbagai perhitungan |
| TPU mampu melakukan sekitar empat triliun operasi per detik dengan konsumsi daya hanya sekitar 2 W | GPU mengonsumsi daya sekitar 350 W |
| TPU tersedia melalui Google Cloud Platform | GPU tersedia secara luas dari berbagai produsen, seperti NVIDIA, AMD, dan Intel |

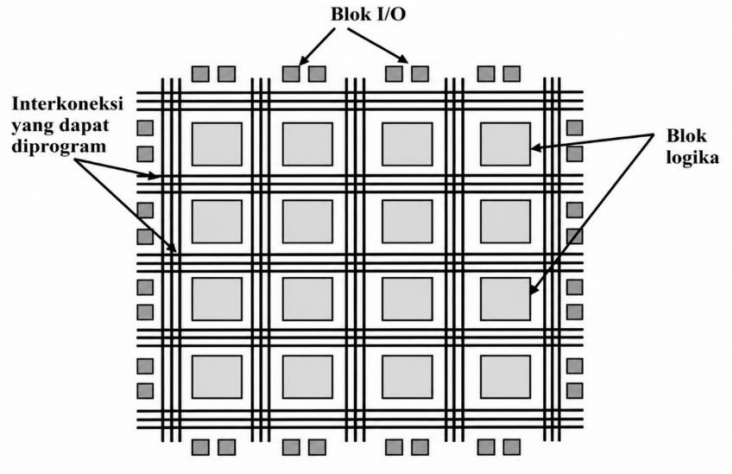
| | |
|---|---|
| TPU terintegrasi dengan kerangka kerja TensorFlow | GPU didukung oleh banyak kerangka kerja seperti TensorFlow, Keras, Caffe, dan lain-lain |

## 13.5 AKSELERASI AI DAN ML DENGAN FPGA

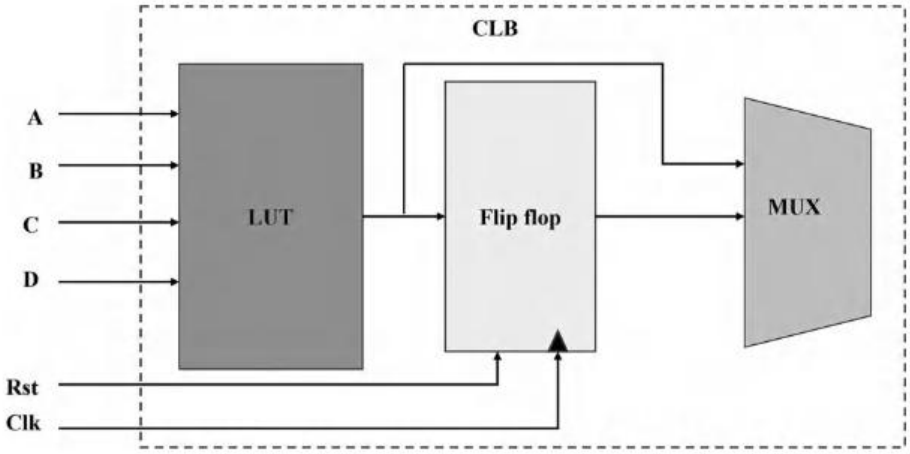
Meningkatnya permintaan akan aplikasi AI dan ML membutuhkan perangkat keras yang efisien untuk aplikasi ini. FPGA semakin populer dalam implementasi AI karena dapat menangani kumpulan data besar dan algoritma kompleks secara real-time dengan latensi rendah dan konsumsi daya yang lebih rendah. FPGA juga menawarkan kemampuan pemrosesan paralel. Fitur terpenting dari FPGA adalah kemampuannya untuk dikonfigurasi ulang. Pengguna dapat menggunakan Bahasa Deskripsi Perangkat Keras (HDL) seperti Verilog atau VHDL untuk mendeskripsikan fungsi perangkat keras yang diinginkan dan memprogram ulang FPGA sesuai kebutuhan, bahkan setelah penerapan. Keluarga FPGA populer yang dikenal karena konsumsi daya rendah dan kepadatan desain yang sederhana meliputi seri ECP3 dan ECP5 dari Lattice Semiconductor, seri Artix-7 dan Kintex-7 dari Xilinx, dan seri Stratix dari Intel.

FPGA, sebuah chip silikon yang telah difabrikasi dan dapat dikonfigurasi ulang, terdiri dari blok logika yang dapat diprogram yang terhubung melalui interkoneksi yang dapat diprogram. FPGA juga memiliki Blok Input/Output (IOB) yang terhubung ke blok logika, menyediakan koneksi di luar chip. Gambar 13.4 menyajikan arsitektur keseluruhan FPGA.
Komponen inti FPGA dirinci di bawah ini:

- Blok Logika yang Dapat Dikonfigurasi (CLB): Blok logika yang dapat diprogram atau Blok Logika yang Dapat Dikonfigurasi, mengimplementasikan logika pengguna dan terdiri dari Multiplexer (MUX), flip-flop D, dan tabel pencarian (LUT). Fungsi logika kombinasi diimplementasikan dengan LUT, yang outputnya disimpan dalam flip-flop D, dan MUX digunakan untuk logika seleksi. Jumlah input ke LUT dapat bervariasi dari 3 hingga 6. FPGA saat ini dengan LUT adaptif dapat menghasilkan dua output dari satu LUT.




**Gambar 13.4** Arsitektur FPGA.




**Gambar 13.5** Arsitektur internal CLB.

Terdapat ratusan hingga ribuan CLB dalam FPGA yang tersusun dalam susunan dua dimensi. Gambar 13.5 menggambarkan arsitektur internal CLB. LUT mengimplementasikan fungsi Boolean dengan menyimpan tabel kebenarannya dalam sel *Static Random Access Memory* (SRAM). Sirkuit sekuensial didukung oleh CLB dengan menghubungkan setiap LUT ke MUX dan flip-flop. Oleh karena itu, CLB memungkinkan implementasi fungsi kompleks.

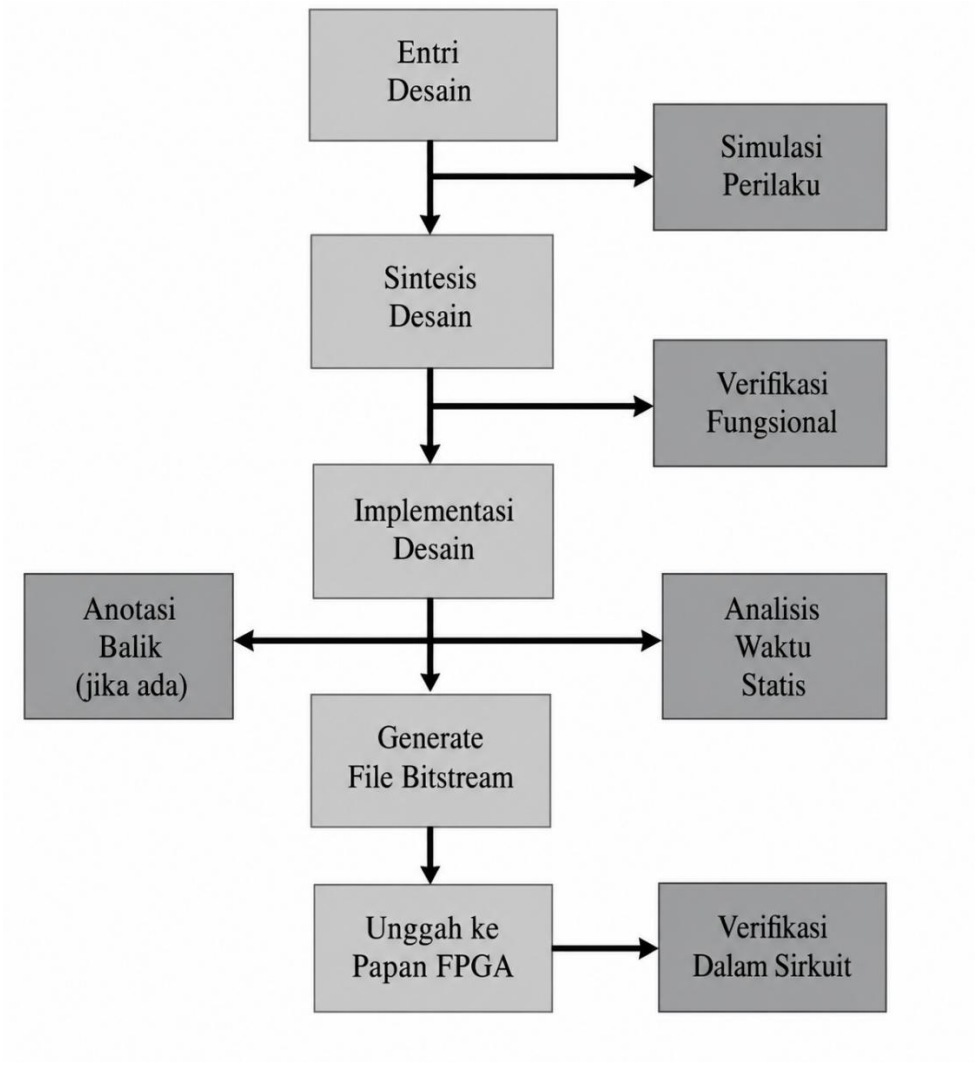
- Interkoneksi: Ini adalah jalur perutean yang dapat diprogram yang menghubungkan CLB dan komponen lain di FPGA. Jaringan interkoneksi memungkinkan pengguna untuk memprogram koneksi spesifik antara berbagai bagian FPGA, memungkinkan perangkat untuk melakukan tugas komputasi tertentu.
- IOB: IOB memungkinkan komunikasi antara logika internal FPGA dan lingkungan eksternal, memungkinkan FPGA untuk berkomunikasi dengan perangkat, sensor, atau prosesor lain.
- Irisan Pemrosesan Sinyal Digital (DSP): FPGA tertentu, seperti Xilinx Virtex-5, memiliki irisan DSP yang dapat melakukan perhitungan numerik berkecepatan tinggi. Berbagai operasi aritmatika yang dilakukan oleh irisan DSP pada FPGA meliputi perkalian-akumulasi, perkalian-penjumlahan, dan penghitung satu atau n langkah.
- Manajemen Jam: FPGA mencakup sirkuit manajemen jam, seperti loop terkunci fase (PLL) dan jaringan distribusi jam, untuk mengontrol waktu operasi.

**Alur Desain FPGA**

Alur desain untuk FPGA terdiri dari beberapa tahapan di mana spesifikasi desain tingkat tinggi diterjemahkan ke dalam implementasi perangkat keras fungsional. Alur desain FPGA ditunjukkan pada Gambar 13.6.

- Spesifikasi Desain dan Pengumpulan Persyaratan: Alur desain FPGA dimulai dengan mendefinisikan persyaratan dan spesifikasi fungsional untuk sistem berbasis FPGA. Ini termasuk memahami aplikasi, metrik kinerja, batasan daya, persyaratan waktu, dan antarmuka I/O yang diperlukan.

- Entri Desain: Fungsionalitas sistem dijelaskan dalam fase ini menggunakan HDL atau Sintesis Tingkat Tinggi (HLS).
- Simulasi Perilaku: Simulasi fungsional sistem yang dikembangkan dalam HDL dilakukan untuk memverifikasi perilaku logis desain. Alat simulasi seperti ModelSim, Xilinx Vivado, dll., digunakan untuk simulasi perilaku. Sebuah testbench dikembangkan untuk mensimulasikan stimulus input dan output yang diharapkan, dan dengan demikian perilaku fungsional desain diverifikasi.
- Sintesis Desain: Sintesis adalah proses mengubah desain HDL menjadi netlist tingkat gerbang. Netlist menggambarkan rangkaian dalam hal gerbang logika, flip-flop, dan interkoneksi yang dibutuhkan untuk mengimplementasikan desain. Alat sintesis seperti Xilinx Vivado dan Intel Quartus memetakan desain logika tingkat tinggi ke perangkat FPGA tertentu. Alat-alat ini selanjutnya melakukan optimasi untuk luas area, daya, dan kecepatan, serta menghasilkan laporan tentang penggunaan sumber daya. Verifikasi fungsional memastikan desain RTL sesuai dengan spesifikasi.




**Gambar 13.6** Alur desain FPGA.

- Implementasi Desain: Fase implementasi melibatkan beberapa langkah seperti translasi (netlist diterjemahkan ke dalam representasi perantara), pemetaan (netlist dipetakan ke blok logika aktual FPGA), penempatan dan perutean (menempatkan elemen logika yang dipetakan ke lokasi fisik aktual pada FPGA dan merute interkoneksi di antara mereka), analisis waktu (memeriksa apakah desain memenuhi persyaratan frekuensi clock).
- Analisis Waktu: Analisis waktu dilakukan untuk memeriksa apakah desain yang diimplementasikan memenuhi batasan waktu yang diberikan oleh pengguna. Analisis waktu memberikan informasi mengenai frekuensi clock, waktu setup dan hold, worst negative slack, dll.
- Pembuatan Bitstream: Tahap terakhir dalam alur desain FPGA adalah pembuatan bitstream, yaitu file biner yang mengkonfigurasi FPGA dengan desain. File bitstream berisi semua informasi yang diperlukan untuk mengkonfigurasi FPGA, termasuk penempatan logika, interkoneksi, dan penugasan pin I/O.
- Desain FPGA memungkinkan implementasi arsitektur perangkat keras yang disesuaikan untuk inferensi jaringan saraf, di mana perkalian dan penjumlahan adalah operasi inti. FPGA menunjukkan keunggulan seperti:
- Dapat diprogram ulang dan dikonfigurasi ulang sesuai dengan beban kerja AI yang berubah.
- Menawarkan kinerja tinggi melalui pemrosesan paralel dan memberikan latensi rendah.
- Desain FPGA dapat dioptimalkan untuk daya rendah, sehingga cocok untuk AI di edge computing.
- FPGA menyediakan fitur keamanan tingkat perangkat keras, seperti enkripsi bitstream dan mekanisme boot aman, melindungi model dan data AI dari perusakan dan akses tidak sah.

FPGA juga menghadapi beberapa kekurangan dalam mengimplementasikan aplikasi AI dan ML, yang meliputi kebutuhan akan keterampilan pemrograman VHDL atau Verilog, biaya unit yang tinggi, throughput yang lebih rendah dibandingkan dengan GPU, dll.

**Memori Bandwidth Tinggi**

Memori Bandwidth Tinggi (HBM) adalah teknologi memori berkinerja tinggi yang dikembangkan untuk meningkatkan kecepatan, efisiensi, dan kemampuan penanganan data dari sistem komputasi modern. Memori HBM memainkan peran penting dalam pelatihan AI dengan mempercepat pengembangan model dan inferensi pada skala yang tidak mungkin dilakukan dengan memori *Double Data Rate* (DDR) atau *Graphics Double Data Rate* (GDDR) konvensional. Misalnya, GPU H100 dari NVIDIA mengintegrasikan modul HBM3E, menawarkan bandwidth memori hingga 3,35 TB/s, yang memfasilitasi pelatihan real-time model bahasa skala besar yang melebihi 100 miliar parameter. Arsitektur HBM ditunjukkan pada Gambar 13.7. Arsitektur dasar HBM mencakup die logika dasar di bagian bawah, di atasnya ditumpuk beberapa die DRAM, biasanya 4, 8, atau 12 lapisan atau lebih. Interkoneksi disediakan menggunakan *Through-silicon vias* (TSV) atau microbump. Die DRAM inti dalam struktur HBM2

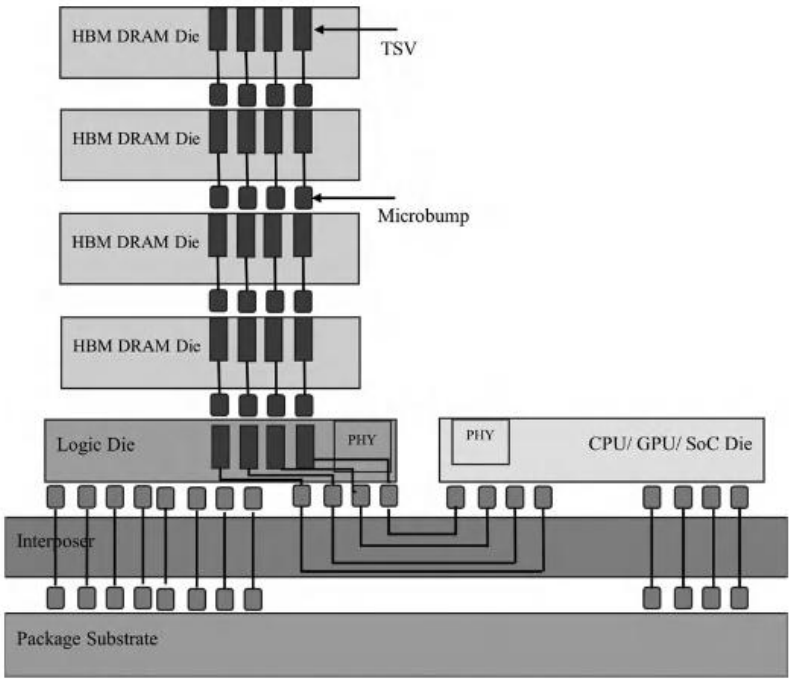
mirip dengan DRAM konvensional dengan susunan sel memori dan sirkuit periferal. Setiap die berisi beberapa saluran dan bank independen, yang terhubung melalui TSV untuk sinyal alamat, data, dan kontrol, dan memungkinkan operasi saluran paralel dan independen.

Die logika dasar terdiri dari PHY, koneksi utama antara DRAM HBM dan pengontrol memori ASIC host, TSV, logika DFT, dan akses langsung. Chip HBM ditempatkan dekat dengan prosesor, sehingga mengurangi latensi dan meningkatkan bandwidth. Desain ini dipadukan dengan antarmuka bus yang lebar yang memungkinkan lebih banyak data ditransfer secara bersamaan, sehingga disebut bandwidth tinggi. HBM diintegrasikan menggunakan teknologi pengemasan 2.5D atau 3D, yang selanjutnya meminimalkan panjang interkoneksi antara memori dan prosesor, serta meningkatkan kinerja dan efisiensi. Oleh karena itu, HBM memanfaatkan DRAM bertumpuk, TSV, micro-bump, dan pengemasan 2.5D untuk memberikan kapasitas yang lebih tinggi, bandwidth yang lebih besar, dan efisiensi daya yang lebih baik daripada DRAM tradisional.

Keunggulan HBM dibandingkan dengan arsitektur memori tradisional adalah:

- Bandwidth lebih tinggi dan konsumsi daya lebih rendah.
- Penempatan HBM dekat dengan prosesor mengurangi latensi dalam mengakses data.
- HBM menempati ruang yang jauh lebih sedikit di papan dibandingkan dengan arsitektur memori konvensional.

Desain dan pembuatan HBM kompleks dan mahal karena melibatkan pengemasan khusus, proses penumpukan 3D, dan teknologi interkoneksi yang kompleks, seperti interposer silikon atau TSV. Faktor-faktor ini membatasi penggunaan HBM dalam aplikasi AI. Namun, teknologi pengemasan canggih, seperti interkoneksi chiplet, menurunkan biaya dan kompleksitas integrasi HBM ke dalam sistem komputasi.




**Gambar 13.7** Memori bandwidth tinggi.

## 13.6 KOMPUTASI DALAM MEMORI DAN MEMRISTOR

Desain perangkat keras AI dibatasi oleh hambatan Von Neumann, di mana bandwidth memori yang tersedia tidak cukup untuk menangani data yang sangat besar dari chip AI modern. Sistem komputasi tradisional dengan unit memori dan pemrosesan terpisah tidak efisien dalam mentransfer data antara memori dan prosesor. Hal ini membutuhkan lebih banyak waktu dan energi untuk transfer data dalam sistem komputer tradisional. Seiring dengan semakin kompleksnya aplikasi AI, semakin banyak dataset yang digunakan, dan latensi dalam transfer data telah menjadi perhatian utama dalam komputasi intensif data. Komputasi Dalam Memori (*In-Memory Computing*/IMC) adalah paradigma komputasi yang memungkinkan pemrosesan data terjadi di tempat data disimpan, sehingga mengurangi kebutuhan akan transfer data yang ekstensif. IMC mengintegrasikan kemampuan komputasi langsung ke dalam perangkat memori. Alih-alih mengandalkan prosesor pusat untuk melakukan semua komputasi, arsitektur IMC memungkinkan unit memori untuk melakukan jenis perhitungan tertentu pada data yang mereka simpan. Metodologi ini sangat berguna untuk operasi seperti perkalian matriks, penjumlahan vektor, dan operasi logika yang umum dalam aplikasi AI/ML. Sistem IMC dapat diimplementasikan menggunakan berbagai jenis teknologi memori, termasuk:

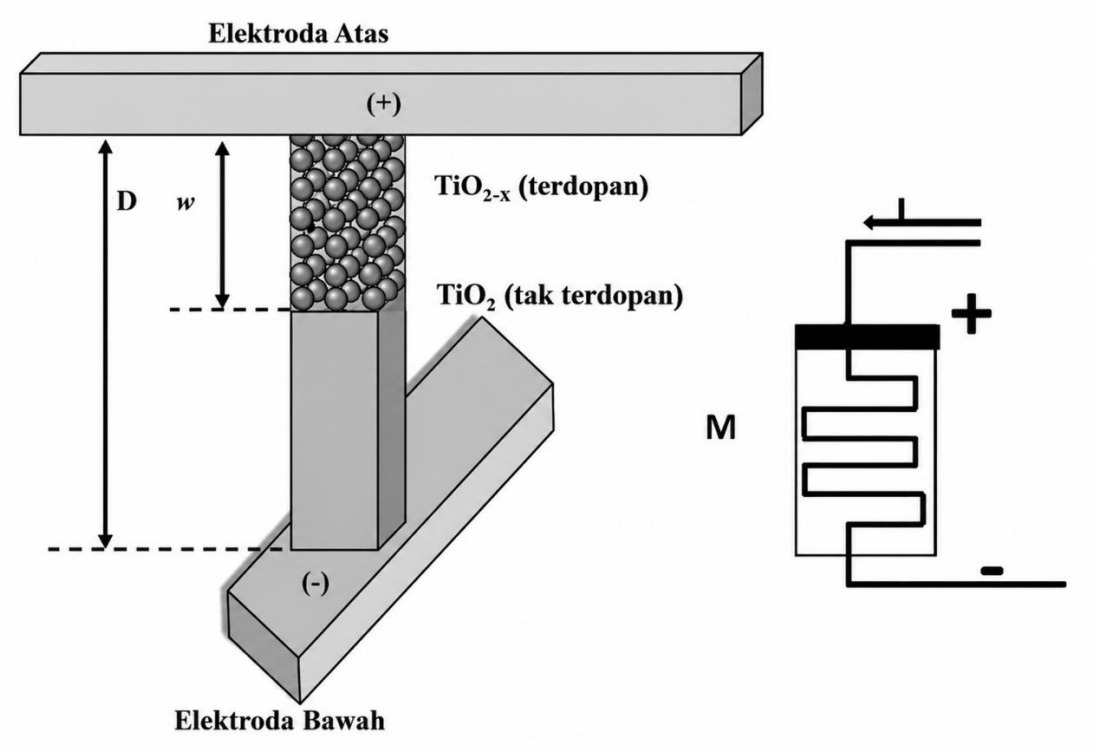
1. SRAM dan DRAM: Jenis memori tradisional ini memungkinkan operasi aritmatika dan logika sederhana dilakukan langsung di dalam memori dengan bantuan sirkuit tambahan. Namun, sifat volatil DRAM membuatnya kurang ideal untuk aplikasi AI dan ML, di mana efisiensi energi merupakan parameter utama yang perlu dipertimbangkan.
2. Memori Non-Volatil (NVM): Teknologi seperti RAM resistif (RRAM) dan memori perubahan fasa (PCM) lebih cocok untuk IMC karena kemampuannya untuk mempertahankan data bahkan ketika daya dimatikan. Perangkat memori ini tidak hanya menyimpan data tetapi juga dapat melakukan komputasi seperti perkalian matriks dengan menggunakan sifat analognya, sehingga cocok untuk aplikasi AI/ML.

**Memristor**

Memristor (resistor memori) atau memori pengalihan resistansi adalah jenis perangkat NVM yang mengintegrasikan penyimpanan dan komputasi di lokasi yang sama untuk mengurangi ukuran perangkat dan konsumsi daya. Ini adalah komponen elektronik dua terminal non-linier yang mempertahankan keadaannya tergantung pada riwayat muatan listrik yang telah mengalir melaluinya. Oleh karena itu, memristor mengingat resistansi terakhir yang dipegangnya bahkan setelah daya dimatikan. Memristor ditemukan secara teoritis oleh Leon Chua pada tahun 1971 dan secara eksperimental pada tahun 2008 oleh para peneliti di HP Labs. Memristor menawarkan kemampuan pemrosesan paralel yang hemat energi.

Gambar 13.8 mengilustrasikan struktur memristor dan simbolnya. Memristor terdiri dari struktur logam-isolator-logam (MIM) dengan dua lapisan elektroda dan lapisan resistif pusat yang terbuat dari semikonduktor atau isolator, yang pada dasarnya memberi mereka kemampuan memori. Elektroda memristor membawa arus dan juga terlibat dalam reaksi resistif. Material elektroda yang umum digunakan adalah material berbasis karbon, nanotube

karbon, nitrida, dll. Material pengalihan resistif dapat berupa material organik atau anorganik seperti oksida biner, perovskit, material 2D, dll.




**Gambar 13.8** Struktur memristor dan simbolnya.

Keunggulan memristor dalam mengimplementasikan aplikasi AI dan ML terletak pada kemampuannya untuk menyimpan data sebagai resistansi, mirip dengan sinapsis otak manusia. Memristor mengubah resistansinya tergantung pada muatan yang melewatinya. Keadaan resistansi ini dipertahankan bahkan tanpa adanya daya, sehingga membuat memristor ideal untuk tugas-tugas yang membutuhkan memori persisten. Resistansi material berubah ketika tegangan diterapkan di antara elektroda karena pergerakan ion atau cacat di dalam material. Resistansi memristor dapat diprogram berdasarkan tegangan atau arus yang diterapkan, sehingga cocok untuk memproses nilai analog yang dibutuhkan dalam tugas AI dan ML.

Memristor menawarkan beberapa keunggulan dibandingkan dengan arsitektur memori konvensional. Memristor memiliki ukuran perangkat yang lebih kecil, pengurangan pembangkitan panas, kecepatan lebih tinggi, dan biaya lebih rendah. Mereka dapat mendukung fungsi analog dan digital serta beroperasi pada tegangan yang lebih rendah, sehingga mengurangi konsumsi daya secara keseluruhan. Namun, ketersediaan komersialnya yang terbatas dan pengembangannya yang relatif baru sebagai elemen sirkuit membuat integrasinya dengan kerangka kerja perangkat keras dan perangkat lunak AI yang ada menjadi sulit.

## 13.7 AI DI TEPI PERANGKAT

Pemrosesan tugas AI secara langsung pada perangkat tepi seperti ponsel pintar, tablet, dll., disebut sebagai AI di tepi. AI di tepi memenuhi permintaan pengambilan keputusan secara real-time, respons latensi rendah, dan privasi data. Dengan memproses data secara lokal, bukan di server cloud, AI tepi dapat menghasilkan respons langsung dan mengurangi ketergantungan pada konektivitas internet yang konstan. Sebagai contoh, perangkat rumah pintar menggunakan AI di tepi untuk menganalisis perilaku pengguna dan memberikan pengalaman yang lebih personal. Konsumsi daya rendah dan latensi minimal adalah persyaratan utama untuk mendesain perangkat AI tepi. AI tepi menghadapi beberapa tantangan perangkat keras dan keamanan, seperti yang dijelaskan di bawah ini:

1. Keterbatasan Daya Pemrosesan: Perangkat seluler dan tepi biasanya dibatasi oleh daya komputasinya dibandingkan dengan server cloud terpusat. Algoritma AI membutuhkan sumber daya komputasi yang signifikan untuk berjalan secara efisien. Untuk menjalankan CNN, model transformer, dll., yang intensif secara komputasi di tepi, produsen perangkat keras telah mengembangkan prosesor khusus, seperti *System-on-Chips* (SoC) khusus AI.
2. Efisiensi Energi: Perangkat seluler bertenaga baterai membatasi jumlah pemrosesan data yang dapat dilakukan secara lokal. Tugas AI membutuhkan pemrosesan berkelanjutan yang dapat menguras daya baterai dengan cepat. Teknik implementasi daya rendah seperti penskalaan tegangan dinamis dan arsitektur berbasis peristiwa digunakan untuk mengoptimalkan efisiensi energi untuk tugas AI di edge.
3. Keterbatasan Memori: Perangkat seluler dan sensor IoT memiliki memori terbatas dibandingkan dengan sistem berbasis cloud. Model pembelajaran mendalam membutuhkan sejumlah besar memori untuk menyimpan bobot, aktivasi, dan parameternya. Hal ini dapat menimbulkan masalah ketika model besar dijalankan pada perangkat dengan RAM dan kapasitas penyimpanan yang terbatas. Teknik kuantisasi dan pemangkasan digunakan untuk mengompres model pembelajaran mendalam sambil mempertahankan akurasi yang sebanding.
4. Disipasi Panas: Komputasi AI dalam prosesor dapat menghasilkan panas berlebih yang mengurangi kinerja prosesor dan akhirnya mengakibatkan kegagalan perangkat keras. Teknik pendinginan seperti pipa panas, ruang uap, dan pasta termal diimplementasikan untuk melindungi sistem dari panas berlebih.
5. Integritas Model dan Privasi Data: Model ML yang berjalan pada perangkat edge rentan terhadap input berbahaya yang dapat mengubah perilaku model dan menyebabkan hasil yang merugikan. Sistem AI edge dapat dimanipulasi oleh serangan adversarial, di mana data yang dirancang dengan cermat dibuat untuk mengelabui model AI, menyebabkan prediksi yang salah. Oleh karena itu, untuk melindungi privasi data pada perangkat edge, protokol enkripsi dan teknik pelestarian privasi seperti privasi diferensial dan pembelajaran federasi harus diimplementasikan.
6. Keamanan Tingkat Perangkat: Untuk melindungi perangkat edge dari akses dan modifikasi yang tidak sah, mekanisme boot aman, lingkungan eksekusi tepercaya (TEE),

dan modul keamanan berbasis perangkat keras dapat digunakan. Boot aman memastikan bahwa perangkat memulai hanya menggunakan perangkat lunak tepercaya. TEE menyediakan area aman di dalam prosesor yang mengisolasi komputasi dan data sensitif dari bagian sistem lainnya.

## 13.8 TEKNOLOGI DAN PERKEMBANGAN PERANGKAT KERAS AI

### *Sistem-On-Chip* (Soc) yang Mendukung AI

Untuk mengeksekusi algoritma AI yang kompleks, akselerator perangkat keras khusus diintegrasikan langsung ke dalam satu chip. Beberapa komponen SoC khusus AI adalah Unit Pemrosesan Neural (NPU), GPU, dan akselerator AI khusus. NPU adalah akselerator perangkat keras khusus yang dirancang untuk mempercepat aplikasi AI. Unit-unit ini sangat paralel dan dapat mengeksekusi tugas AI jauh lebih cepat daripada prosesor tradisional. SoC yang mendukung AI seringkali mencakup inti hemat energi yang dirancang untuk menangani tugas AI sambil meminimalkan konsumsi daya.

### Biokomputer

Biokomputer menggunakan molekul dan sistem biologis untuk memproses informasi. Biomolekul ini melakukan komputasi dengan berinteraksi satu sama lain dengan cara yang mirip dengan cara kerja transistor silikon dalam sistem komputer tradisional. DNA atau protein dapat digunakan untuk mengkodekan dan memproses informasi. Biokomputer sangat paralel, yaitu, mereka dapat melakukan banyak komputasi secara bersamaan, sehingga cocok untuk aplikasi AI, di mana sejumlah besar data perlu diproses dengan cepat. Biokomputer hemat energi, karena dapat melakukan komputasi pada tingkat daya yang jauh lebih rendah daripada sistem berbasis silikon tradisional.

Biokomputer dapat berupa biokimia, biomekanik, atau bioelektronik, berdasarkan media operasi dan sinyal keluarannya. Komputer biokimia menggunakan reaksi kimia sebagai masukan dan memberikan konsentrasi kimia spesifik sebagai keluaran. Komputer biomekanik melakukan komputasi melalui proses mekanis dengan keluaran yang ditentukan oleh perubahan fisik. Komputer bioelektronik, di sisi lain, menjalankan komputasi menggunakan komponen biologis untuk memengaruhi konduktivitas listrik. Perubahan konduktivitas diinterpretasikan sebagai keluaran komputasi dalam komputer bioelektronik.

### Faktor Geopolitik yang Membentuk Industri Perangkat Keras AI

Industri perangkat keras AI dipengaruhi oleh beberapa faktor geopolitik seperti kebijakan perdagangan, aliansi teknologi, kontrol ekspor, dan persaingan atas sumber daya strategis. Industri perangkat keras AI bergantung pada rantai pasokan global untuk semikonduktor, komponen memori, dan bahan baku.

Negara-negara membentuk aliansi dan kolaborasi teknologi untuk meningkatkan kemampuan perangkat keras AI mereka. Aliansi Chip 4 menyatukan empat negara anggota, Amerika Serikat, Jepang, Korea Selatan, dan Taiwan, untuk mengamankan rantai pasokan semikonduktor global dan mengurangi ketergantungan pada Tiongkok.

AS dan Tiongkok berada dalam persaingan teknologi, menganggap AI sebagai teknologi utama untuk pertumbuhan ekonomi dan keamanan nasional. Hal ini akhirnya menyebabkan

persaingan yang ketat dalam pengembangan perangkat keras, semikonduktor, dan infrastruktur komputasi khusus AI. Perang di Ukraina telah menunjukkan pentingnya perjuangan ini dengan penggunaan drone yang ditingkatkan AI dalam peperangan. AS telah memberlakukan kontrol ekspor untuk membatasi akses Tiongkok ke sumber daya semikonduktor. Oleh karena itu, Tiongkok telah meningkatkan upayanya untuk membangun industri semikonduktor yang kuat.

Taiwan, Korea Selatan, dan Jepang adalah produsen utama semikonduktor canggih dan komponen elektronik yang penting untuk perangkat keras AI. *Taiwan Semiconductor Manufacturing Company* (TSMC) adalah salah satu pemasok terkemuka chip mutakhir yang digunakan dalam akselerator AI, GPU, dan SoC. Pandemi COVID-19, perang dagang AS-Tiongkok, dan kekurangan semikonduktor telah menuntut kebutuhan akan rantai pasokan yang lebih kuat untuk perangkat keras AI. Negara-negara mengembangkan pusat produksi regional mereka sendiri dan menciptakan aliansi teknologi untuk memastikan pasokan komponen penting yang stabil.

Produksi perangkat keras AI bergantung pada akses ke logam tanah jarang, mineral, dan bahan penting lainnya. Pada tahun 2024, Tiongkok memiliki cadangan unsur tanah jarang yang signifikan, yang sangat penting untuk pembuatan chip AI, sensor, dan komponen elektronik dalam jumlah 44 juta metrik ton, diikuti oleh AS. Negara-negara lain yang berkontribusi terhadap logam tanah jarang adalah Myanmar, Nigeria, Thailand, Australia, India, dan Rusia. Seiring meningkatnya aplikasi AI, negara-negara berupaya meningkatkan akses mereka terhadap logam tanah jarang dan mengurangi ketergantungan pada wilayah tertentu. Hal ini telah menyebabkan investasi dalam sumber alternatif bahan-bahan penting, serta pengembangan teknologi daur ulang baru untuk memulihkan unsur tanah jarang dari limbah elektronik.

**Faktor Ekonomi yang Membentuk Industri Perangkat Keras AI**

Beberapa faktor ekonomi, seperti tren investasi, pengeluaran yang dilakukan untuk pembahasan dan pengembangan, persaingan pasar, kebutuhan industri yang mengintegrasikan teknologi AI, dll., membentuk industri perangkat keras AI. Perusahaan teknologi terkemuka seperti NVIDIA, Intel, Google, dan AMD, produsen semikonduktor, dan pemerintah menginvestasikan sumber daya yang besar ke dalam pembahasan dan pengembangan untuk mengembangkan prosesor AI generasi berikutnya, akselerator, dan teknologi memori. Pemerintah AS meluncurkan Inisiatif AI Amerika pada tahun 2019, yang mencakup investasi dalam pembahasan perangkat keras AI, memungkinkan sistem yang lebih efisien dan terukur untuk memenuhi permintaan yang terus meningkat untuk aplikasi AI di berbagai industri.

Perusahaan yang dapat meningkatkan produksi perangkat keras khusus AI dengan biaya lebih rendah akan mendapatkan keuntungan di pasar. Persaingan yang semakin ketat antar produsen perangkat keras untuk menghadirkan akselerator AI, GPU, dan teknologi memori terbaru ke pasar menyebabkan inovasi yang lebih cepat. Persaingan ini diharapkan dapat menurunkan biaya perangkat keras AI dari waktu ke waktu, sehingga lebih mudah diakses oleh berbagai industri dan aplikasi. Teknologi AI banyak digunakan di berbagai industri

termasuk perawatan kesehatan, keuangan, dan otomotif. AI digunakan di industri perawatan kesehatan untuk tugas-tugas seperti pencitraan medis, penemuan obat, dan pengobatan personal. Perangkat keras AI sangat penting di industri otomotif untuk mengembangkan kendaraan otonom. Kendaraan otonom ini bergantung pada algoritma AI untuk pengambilan keputusan secara real-time dan pemrosesan sensor. Sektor keuangan memanfaatkan AI untuk deteksi penipuan, perdagangan algoritmik, dan manajemen risiko. Oleh karena itu, sektor keuangan memiliki permintaan yang meningkat untuk perangkat keras AI yang dapat menangani pemrosesan data frekuensi tinggi.

Model AI membutuhkan daya komputasi dan energi yang sangat besar untuk melatih dan menerapkannya. Industri perangkat keras AI sedang mengembangkan solusi komputasi yang lebih hemat energi, seperti prosesor neuromorfik, IMC, dan akselerator khusus AI untuk mengatasi kekhawatiran ini. Perangkat keras AI juga dikembangkan dengan fokus pada meminimalkan limbah elektronik.

**Arah Masa Depan Industri Perangkat Keras AI**

Dengan model AI yang kompleks dan kumpulan data yang besar, masa depan perangkat keras AI akan ditentukan oleh teknologi yang muncul, inovasi arsitektur, dan sumber daya baru yang dapat memenuhi kebutuhan komputasi, memori, dan energi dari sistem AI canggih.

1. Komputasi Heterogen: Arsitektur komputasi heterogen adalah salah satu arah masa depan utama yang dapat diadopsi untuk mengoptimalkan beban kerja AI. Komputasi heterogen melibatkan pengintegrasian berbagai jenis prosesor, seperti CPU, GPU, TPU, dan NPU, ke dalam satu sistem. Komputasi heterogen memungkinkan penanganan tugas komputasi yang berbeda secara efisien dan meningkatkan kinerja serta efisiensi energi sistem AI.
2. Komputasi Neuromorfik dan Kuantum: Komputasi neuromorfik dan komputasi kuantum adalah dua teknologi yang paling menjanjikan untuk meningkatkan efisiensi perangkat keras AI. Komputasi neuromorfik menggunakan neuron spiking untuk meniru struktur dan fungsi otak manusia. Neuron yang menghasilkan impuls (spiking neurons) digunakan untuk mengimplementasikan prosesor hemat energi dan berbasis peristiwa yang sangat cocok untuk tugas-tugas AI. Hal ini memungkinkan konsumsi daya rendah, pemrosesan waktu nyata, dan kemampuan adaptasi untuk aplikasi AI. Komputasi kuantum mampu memecahkan masalah optimasi yang kompleks secara komputasi yang tidak dapat dipecahkan oleh komputer klasik. Komputer kuantum menyediakan solusi canggih dalam ilmu material, kriptografi, dll., dengan pelatihan jaringan saraf skala besar yang lebih cepat dan pemrosesan data yang efisien.

## 13.9 DINAMIKA DAN TREN MASA DEPAN PERANGKAT KERAS AI

Bab ini membahas berbagai arsitektur perangkat keras yang mendukung AI dan ML, dengan fokus pada CPU, GPU, TPU, dan FPGA. CPU menawarkan keserbagunaan tujuan umum tetapi kurang paralel dibandingkan dengan perangkat keras khusus. GPU, dengan kemampuan pemrosesan paralelnya, telah menjadi inti dari pelatihan dan inferensi pembelajaran mendalam. TPU, yang dirancang khusus untuk operasi tensor, memberikan kinerja yang

dioptimalkan untuk tugas-tugas AI skala besar. FPGA menawarkan solusi yang dapat disesuaikan yang memberikan inferensi latensi rendah dan hemat daya untuk aplikasi khusus. Perbandingan arsitektur perangkat keras ini disajikan dalam Tabel 13.5. Teknologi memori yang sedang berkembang seperti HBM dan IMC mendukung kebutuhan bandwidth memori yang terus meningkat dari beban kerja AI dengan meminimalkan latensi dan meningkatkan throughput data.

Signifikansi AI di edge computing dan pengenalan SoC yang mendukung AI dan biokomputer juga diulas dalam bab ini. Bab ini juga membahas faktor geopolitik yang membentuk teknologi perangkat keras AI. Negara-negara semakin fokus pada kemampuan produksi dalam negeri, hak kekayaan intelektual, dan kemitraan untuk mengamankan posisi mereka dalam persaingan AI global. Masa depan perangkat keras AI dan ML akan bergantung pada kemajuan dalam desain arsitektur dan teknologi memori, dengan keseimbangan yang cermat antara kinerja, efisiensi, dan biaya.

**Tabel 13.5** Perbandingan berbagai perangkat keras untuk AI/ML

| Metrik | CPU | GPU | TPU | FPGA |
|---|---|---|---|---|
| Arsitektur | Serbaguna | Inti yang sangat paralel, dioptimalkan untuk tugas data berskala besar | Dikhususkan untuk operasi tensor | Gerbang logika yang dapat dikonfigurasi |
| Paralelisme | Rendah | Tinggi | Tinggi | Dapat dikustomisasi |
| Performa AI/ML | Sedang | Tinggi (sangat baik untuk pelatihan DNN) | Tinggi (dioptimalkan untuk TensorFlow) | Baik (untuk aplikasi inferensi khusus) |
| Konsumsi daya | Tinggi | Tinggi | Rendah hingga sedang | Sedang |
| Biaya | Rendah | Tinggi (NVIDIA Tesla, A100) | Berbasis layanan cloud (Google TPU) | Tinggi (biaya awal tinggi, tetapi hemat biaya untuk aplikasi khusus) |
| Kasus penggunaan | Beban kerja ML serbaguna | Pelatihan dan inferensi untuk DNN berukuran besar | Beban kerja DNN berbasis TensorFlow | Inferensi AI waktu nyata, Edge AI |

# BAB 14
# DAC 8-BIT MODE ARUS DAYA RENDAH BERBANTUAN AI UNTUK VLSI

## 14.1 INTEGRASI AI PADA KONVERTER DIGITAL-KE-ANALOG MODE ARUS

Permintaan akan perangkat elektronik portabel dan cepat telah meningkat secara substansial dalam beberapa tahun terakhir. Mengembangkan komponen analog dan sinyal campuran (AMS) yang hemat energi, berkecepatan tinggi, dan portabel yang memenuhi persyaratan saat ini adalah fokus utama pembahasan modern. Dalam berkomunikasi dengan dunia analog, ada kebutuhan untuk menerjemahkan sinyal analog kontinu, misalnya, suhu, tekanan, suara, dan gambar, ke dalam bentuk digital untuk diproses.

Demikian pula, menerjemahkan sinyal digital yang diproses ke dalam bentuk analog juga sama pentingnya dalam aplikasi yang mencakup instrumentasi medis, identifikasi frekuensi radio (RFID), antena nirkabel, dan antarmuka sensor. Dalam sistem tersebut, konverter digital-ke-analog (DAC) sangat penting untuk memfasilitasi antarmuka yang mulus antara prosesor digital dan keluaran analog dunia nyata. Sebagai ilustrasi, DAC dalam peralatan medis mengubah sampel pulsa diskrit menjadi bentuk gelombang analog kontinu, sementara aktuator daya rendah dan sistem RFID mengandalkan DAC terintegrasi untuk mencapai kontrol sinyal yang efektif.

Munculnya DAC dengan kinerja tinggi, daya rendah, dan area chip kecil telah menarik lebih banyak perhatian dari para peneliti karena semakin populer dalam kehidupan sehari-hari. Karena banyak arsitektur telah dikembangkan untuk memenuhi kebutuhan ini, jenis yang paling umum adalah DAC Pengarah Arus, DAC Berbobot Biner, DAC tangga R-2R, dan DAC Rangkaian Resistor. Masing-masing desain menunjukkan kompromi sendiri dalam resolusi tinggi, linearitas, efisiensi daya, kecepatan switching, dan ukuran area chip. Misalnya, DAC Pengarah Arus cocok untuk sistem komunikasi kecepatan tinggi. Namun, tegangan keluaran desain ini rendah, dan mekanisme switching-nya juga sangat kompleks.

Demikian pula, DAC Pengarah Arus Berkode Termometer ideal untuk aplikasi tingkat lanjut, karena menawarkan linearitas yang lebih baik dengan gangguan yang lebih rendah saat mengubah keadaan energi. Namun, ukuran area chipnya lebih besar, dan membutuhkan hampir dua kali lipat sumber arus untuk setiap bit resolusi. Mengenai DAC Berbobot Biner, desainnya tampak jauh lebih sederhana, tetapi menghasilkan gangguan yang lebih tinggi dan kesalahan DNL yang besar untuk resolusi bit tinggi. DAC Rangkaian Resistor bagus untuk aplikasi resolusi tinggi dengan kecepatan tinggi, tetapi area fabrikasi chipnya besar karena membutuhkan 2n resistor untuk mendapatkan resolusi N-bit.

Untuk mengatasi keterbatasan ini, beberapa pendekatan telah diusulkan oleh para peneliti. Misalnya, DAC pengarah arus tersegmentasi telah dikembangkan oleh Sarkar dkk, yang menggabungkan dekode termometer dengan sumber arus berbobot biner. Desain ini mencapai keseimbangan yang baik antara linearitas dan resolusi. Demikian pula, DAC pengarah arus 10-bit, 300 MS/s telah dikembangkan oleh Mehrjoo dan Buckwalter dalam

dengan buffer FET bertumpuk untuk meningkatkan resolusi yang lebih tinggi, meskipun hal ini meningkatkan konsumsi daya. Mercer menyajikan DAC 14-bit berdaya rendah yang menawarkan resolusi tinggi dengan konsumsi daya rendah. Fitrio dkk. menggunakan dekode termometer untuk MSB untuk mencapai kecepatan pengambilan sampel yang lebih tinggi. Deveugele dan Steyaert merancang DAC pemisah arus berbasis MOS 4-bit yang mengkonsumsi daya lebih rendah tetapi menunjukkan linearitas yang terbatas. Leñero-Bardallo dkk. memperkenalkan sumber arus rentang nanoampere menggunakan transistor NMOS yang dapat disesuaikan secara digital. Linares-Barranco et al. mengusulkan mini-DAC yang dioptimalkan yang beroperasi di wilayah inversi lemah untuk mencapai konsumsi daya ultra-rendah.

Berbeda dengan desain di atas, DAC tangga R-2R lebih efisien dalam hal luas area dan dapat diskalakan. Sirkuit ini hanya menggabungkan dua resistor (R dan 2R) dan oleh karena itu cocok untuk integrasi CMOS. DAC R-2R tidak memerlukan pencocokan arus yang tepat dan tidak menarik arus kontinu, tidak seperti DAC Pengarah Arus. Mereka mengkonsumsi daya statis yang lebih sedikit dan memiliki tata letak desain yang lebih sederhana.

Dibandingkan dengan DAC Rangkaian Resistor, desain R-2R lebih dapat diskalakan dan menghilangkan jumlah resistor. Meskipun kecepatannya dibatasi sampai batas tertentu oleh penundaan RC parasit, desain R-2R menghadirkan keseimbangan yang menarik antara efisiensi daya, linearitas, dan kesederhanaan. Esmaili dan Babazadeh merancang DAC 14-bit, 125 MHz menggunakan logika R-2R, tetapi mereka menghadapi masalah dengan ketidakcocokan sumber arus. Kristianti dkk.menyajikan DAC 8-bit berdasarkan tangga R-2R, penguat operasional, dan sakelar MOS untuk mewujudkan kinerja komunikasi kecepatan tinggi.

Saat mendesain sirkuit VLSI daya rendah, analisis Monte Carlo, analisis parasitik, dan analisis Proses, Tegangan, dan Suhu (PVT) adalah parameter paling penting yang digunakan untuk memverifikasi ketahanan. Penggabungan Kecerdasan Buatan (AI) dalam desain VLSI modern memungkinkan optimasi area yang lebih baik, efisiensi energi, dan kinerja keseluruhan. Sebagai contoh, Fayazi dkk. Berfokus terutama pada pemodelan ukuran dan kinerja sirkuit AMS, memeriksa dan mengintegrasikan alat AI dari *Support Vector Machines* (SVM), *Artificial Neural Networks* (ANN), *Deep Neural Networks* (DNN), *Convolutional Neural Networks* (CNN), dan *Recurrent Neural Networks* (RNN). Sebaliknya, Rapp dkk. Telah memberikan cakupan yang lebih luas dari aplikasi Pembelajaran Mesin (ML) dalam desain berbantuan komputer (CAD) sirkuit terpadu (IC). Mereka membahas isu-isu seperti eksplorasi desain tingkat sistem, sintesis tingkat tinggi, sintesis logika, desain tata letak fisik dan analog, litografi, manufaktur, dan pengujian. Demikian pula, desain IC analog dan RF menggunakan teknik ML telah dipresentasikan oleh Afacan dkk. Di mana pemodelan perangkat, sintesis front-end, desain tata letak, dan pengujian kesalahan dibahas. Survei terperinci tentang metode otomatisasi desain elektronik (EDA) berbasis AI/ML untuk desain IC yang sadar keamanan telah dipresentasikan oleh Koblah dkk. Mereka menekankan bagaimana AI dapat menambahkan batasan terkait keamanan, mengurangi tanggung jawab, dan melindungi sistem dari berbagai wabah. Golzan dkk. Menunjukkan Perspektif Kecerdasan Buatan dalam pemodelan dan optimasi IC AMS.

Dalam penggunaan dunia nyata, peningkatan VLSI yang berfokus pada AI ini memberikan manfaat nyata; perangkat edge dan IoT dapat melakukan inferensi AI kompleks secara lokal dengan latensi yang lebih rendah dan efisiensi energi yang lebih baik; Robot dan kendaraan otonom memperoleh persepsi dan kontrol yang cepat melalui akselerator berkinerja tinggi dan hemat daya yang mendukung visi dan pengambilan keputusan secara real-time; dan NLP serta aplikasi cerdas lainnya memanfaatkan inti VLSI khusus untuk menangani model besar secara efisien.

Dengan menyelaraskan strategi desain VLSI—mulai dari optimasi memori dan aliran data hingga akselerator khusus, fabrik yang dapat dikonfigurasi ulang, dan manajemen daya—dengan tuntutan beban kerja AI, para perancang mengembangkan platform perangkat keras yang jauh lebih cepat, lebih hemat energi dan luas, serta sangat cocok untuk aplikasi modern dalam komputasi edge, robotika, sistem otonom, dan lainnya.

Berdasarkan fakta-fakta ini, bab ini mengusulkan DAC tipe tangga mode arus yang memanfaatkan kekuatan arsitektur tangga R-2R sambil beroperasi dalam mode arus untuk meningkatkan kemampuan penggerak dan kinerja kecepatan tinggi. Karya ini juga mengusulkan agar tangga R-2R diimplementasikan menggunakan transistor MOS alih-alih resistor. Desain yang diusulkan menggunakan Penguat Transresistansi Operasional (OTRA) sebagai komponen aktif tunggal dengan sakelar MOS untuk melakukan konverter digital-ke-analog mode arus yang efisien. Selain itu, dengan mengintegrasikan teknik AI/ML canggih, DAC mencapai stabilitas dan akurasi yang lebih tinggi, bahkan dalam kondisi lingkungan yang bervariasi.

Bagian-bagian selanjutnya dari bab ini disusun sebagai berikut. Bagian 14.2 menyajikan rangkaian DAC, termasuk OTRA, sakelar digital berbasis MOSFET, dan jaringan tangga MOSFET. Bagian 14.3 menjelaskan arsitektur DAC yang diusulkan secara detail. Bagian 14.4 menyajikan hasil simulasi dan analisis kinerja. Pada Bagian 14.5, metode penyetelan berbasis AI adaptif untuk mencapai kinerja DAC yang tangguh dibahas dan dianalisis.

## 14.2 PERANCANGAN DAC BERBASIS OTRA CMOS

### Penguat Transresistansi Operasional

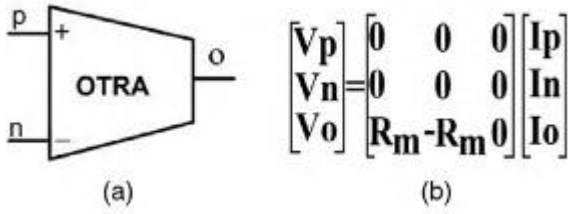
OTRA CMOS merupakan salah satu blok mode arus paling ampuh yang mengatasi beberapa keterbatasan penguat operasional mode tegangan tradisional sekaligus menawarkan kesesuaian untuk implementasi VLSI berkinerja tinggi. Tidak seperti penguat operasional, yang menderita keterbatasan produk gain-bandwidth dan laju perubahan tegangan, OTRA menawarkan bandwidth gain yang tinggi, sehingga sangat menarik untuk aplikasi frekuensi tinggi. Gambar 14.1a dan b menunjukkan diagram blok dan matriks port OTRA.

Di sini, arus input yang diterapkan pada terminal positif dan negatif masing-masing adalah Ip dan In, sedangkan Vo adalah tegangan output pada terminal impedansi rendah. Istilah Rm mewakili gain transresistansi DC dari penguat, idealnya tak terbatas. Dari matriks ini, terlihat jelas bahwa terminal input berada pada ground virtual $V_p = V_n = 0$. Selanjutnya, tegangan output OTRA dapat ditulis sebagai:

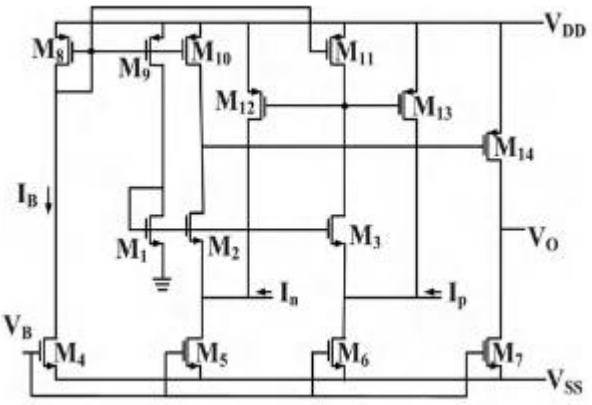
$$V_0 = (I_p - I_n) \tag{14.1}$$

Hubungan fundamental persamaan (14.1) menggambarkan bahwa karena impedansi input yang sangat rendah, OTRA sangat toleran terhadap kapasitansi liar dan parasitik perangkat, sehingga cocok untuk desain frekuensi tinggi. Berbagai topologi sirkuit telah diadopsi untuk realisasi OTRA berkinerja tinggi, seperti yang dijelaskan dalam referensi . Salah satu realisasi khusus OTRA CMOS berkinerja tinggi, seperti yang digambarkan pada Gambar 14.2, disajikan dalam referensi. Arsitektur CMOS internal dari OTRA dapat dipahami sebagai koneksi kaskade dari dua sub-blok: (i) tahap konveyor arus diferensial, yang merasakan dan mengurangi arus input, dan (ii) tahap penguatan tegangan common-source, yang mengubah perbedaan arus yang dihasilkan menjadi tegangan dengan penguatan transresistansi yang besar.

Dalam desain CMOS praktis, tahap input diferensial direalisasikan dengan konfigurasi cermin arus, memastikan perbedaan arus yang tepat antara Ip dan In. Tahap kedua menggunakan penguat MOSFET bertingkat dengan penguatan tinggi untuk mencapai nilai Rm yang besar, seringkali dalam orde ratusan kilo-ohm atau lebih tinggi, tergantung pada bias. Arus yang mengalir melalui transistor NMOS dapat dinyatakan dengan asumsi bahwa semuanya beroperasi di wilayah saturasi sebagai berikut. Tabel 14.1 menampilkan rasio aspek MOSFET yang digunakan dalam OTRA yang diusulkan.




**Gambar 14.1** (a) Diagram blok OTRA. (b) Matriks port OTRA.




**Gambar 14.2** Implementasi CMOS dari OTRA.

$$I_D = \frac{K}{2}(V_{GS} - V_T)^2$$

Dimana

$$K = \mu_n C_{ox} \frac{W}{L}$$

$$V_T = V_{T0} + \gamma(\sqrt{V_{SB} + 2_{jf}} - \sqrt{2_{jf}}$$

Salah satu fitur paling menarik dari CMOS OTRA adalah kemampuannya untuk menghilangkan nonlinieritas transistor. Dalam operasi transistor MOS konvensional, perangkat yang bekerja di wilayah ohmik atau trioda menimbulkan nonlinieritas karena ketergantungan kuadrat arus drain pada tegangan gerbang dan drain. Namun, karena terminal input OTRA dipaksa untuk tetap berada pada ground virtual, variasi tegangan sinyal kecil pada node ini dapat diabaikan. Kondisi ini menekan ketergantungan tegangan nonlinier dari MOSFET input, memastikan bahwa karakteristik transfer antara input arus dan tegangan output tetap linier. Selain itu, karena OTRA bergantung pada perbedaan arus dan bukan pada impedansi input tinggi, efek ketidaksesuaian dan modulasi panjang kanal berkurang secara signifikan. Akibatnya, banyak nonlinieritas yang melekat pada perangkat MOS secara efektif dihilangkan dalam konfigurasi OTRA, menghasilkan pemrosesan sinyal linier yang akurat bahkan pada frekuensi tinggi.

**Tabel 14.1** Rasio W/L MOS dalam OTRA

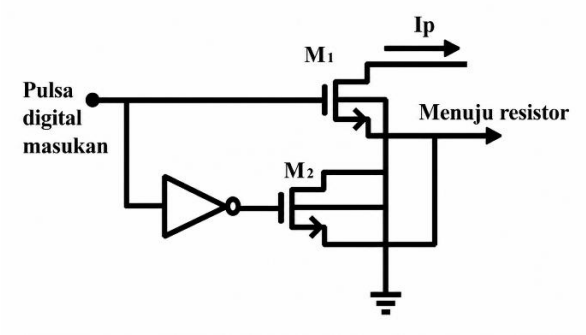
| TRANSISTOR | W (µM) | L (µM) |
|---|---|---|
| M8, M9, M10, M11, M12, M13, M14 | 4 | 0,18 |
| M1, M2, M3, M4, M5, M6, M7 | 12 | 0,18 |

**MOSFET sebagai Sakelar Digital**

MOSFET umumnya digunakan sebagai sakelar digital dalam rangkaian digital. Tergantung pada tegangan yang diterapkan pada terminal gerbang, ia dapat digunakan di wilayah cut-off atau wilayah saturasi. MOSFET di wilayah cut-off dimatikan, dan tidak menghantarkan arus apa pun antara terminal drain dan source. Sedangkan di wilayah saturasi, MOSFET menghantarkan arus, dan arus dialirkan melalui terminal source dan drain. Rangkaian yang ditunjukkan pada Gambar 14.3 bekerja sebagai sakelar pengarah arus komplementer di mana MOSFET M1 dan M2 beroperasi di wilayah pulsa yang berbeda. Segera setelah input menjadi tinggi, transistor M1 menerima tegangan gerbang yang lebih besar dari ambangnya dan menyala. Karena drain-nya terhubung ke sumber arus Ip dan node source dihubungkan ke resistor, M1 beroperasi di wilayah saturasi (ketika tegangan node output cukup rendah) atau di wilayah trioda (ketika tegangan node naik mendekati tegangan gerbang).

Dalam kondisi ini, M2 tetap dalam keadaan terputus (*cut-off*), karena gerbangnya digerakkan ke rendah oleh inverter, memastikan bahwa tidak ada arus yang dibuang ke tanah. Dengan demikian, seluruh arus Ip mengalir melalui M1 menuju resistor. Sebaliknya, ketika

input berlogika rendah, M1 digerakkan ke keadaan terputus (*cut-off*), memblokir jalur arus, sementara inverter menggerakkan gerbang M2 ke tinggi, memaksa M2 ke wilayah linier (triode). Dalam keadaan ini, M2 menyediakan jalur resistansi rendah ke tanah, melepaskan muatan node keluaran dengan cepat dan menahannya pada nol volt. Aksi pengalihan komplementer ini menjamin keadaan ON dan OFF yang terdefinisi, di mana arus node keluaran dialirkan ke resistor atau diatur ulang ke nol. Ringkasan operasinya disajikan dalam Tabel 14.2.




**Gambar 14.3** MOSFET sebagai sakelar digital.

**Tabel 14.2** Tabel kebenaran operasi MOSFET sebagai sakelar digital

| Pulsa Masukan | Daerah M1 | Daerah M2 | Perilaku Keluaran |
|---|---|---|---|
| Logika 1 | Saturasi/Trioda | Cut-off (Mati) | **Ip** mengalir menuju resistor |
| Logika 0 | Cut-off (Mati) | Trioda (ON, pelepasan muatan) | Node keluaran dijepit ke ground |

**Rangkaian tangga**

Transistor NMOS menggantikan resistor pada rangkaian tangga R-2R tradisional, pada rangkaian yang diusulkan dengan menggunakan persamaan berikut,

$$R = \frac{1}{\mu_n X_{ox} W (V_{gs} - V_{th})/L} \tag{14.5}$$

Resistansi transistor NMOS di daerah linier dapat didekati dengan parameter transkonduktansi (gm) dibagi dengan tegangan drain-source (VDS). Resistansi ini sering disebut sebagai resistansi sinyal kecil atau resistansi inkremental, karena mewakili resistansi transistor MOS terhadap perubahan kecil pada tegangan atau arus. Ketidaksesuaian interkorelasi hanya pada MOSFET DAC disebabkan oleh susunan tangga transistor dan akibatnya, menghasilkan ketidaksesuaian arus IREF/4 yang terkait dengan ketidaksesuaian transistor M2 dengan ukuran W/L. Demikian pula, IREF/8 dihasilkan dari ketidaksesuaian M2 dan M6, sedangkan IREF/16 berasal dari ketidaksesuaian pada M2, M6, dan M10, dan seterusnya. Akibatnya, kesalahan ketidaksesuaian transistor akhir cukup besar, karena mengakumulasi kesalahan ketidaksesuaian transistor sebelumnya.

Untuk meminimalkan ketidaksesuaian interkorelasi pada MOSFET, metode pertama melibatkan perataan arus keluaran DAC dengan beberapa replika DAC serupa untuk mengurangi dampak efek ketidaksesuaian acak dan meningkatkan monotonisitas. Namun,

teknik ini terhambat oleh kerugian disipasi daya yang tinggi karena arus keluaran empat kali lebih besar daripada pada DAC tunggal. Teknik lain digunakan untuk mengurangi dampak ketidaksesuaian. Dalam teknik ini, arus yang mengalir melalui DAC diminimalkan. Minimisasi ini dilakukan dengan menghubungkan transistor secara seri ke ground. Arus juga dapat diminimalkan dengan mengurangi arus bias, atau dapat dikurangi dengan meminimalkan rasio transistor (W/L).

**Rangkaian DAC berbasis OTRA yang diusulkan**

Rangkaian yang diusulkan pada Gambar 14.4 mewakili DAC mode arus 8-bit yang menggunakan jaringan tangga R–2R berbasis MOSFET, tahap pengalihan digital, dan OTRA untuk melakukan konversi digital-ke-analog. Desainnya kompak, efisien untuk integrasi CMOS, dan sangat cocok untuk aplikasi kecepatan tinggi karena arsitektur pengarah arusnya. Untuk memahami pengoperasian rangkaian yang diusulkan, penting untuk menganalisis rangkaian langkah demi langkah. Langkah awal dimulai dengan cara kerja jaringan switching digital, diikuti dengan pemeriksaan jaringan ladder MOS. Terakhir, jelaskan peran OTRA dan resistor umpan balik dalam menghasilkan output analog.

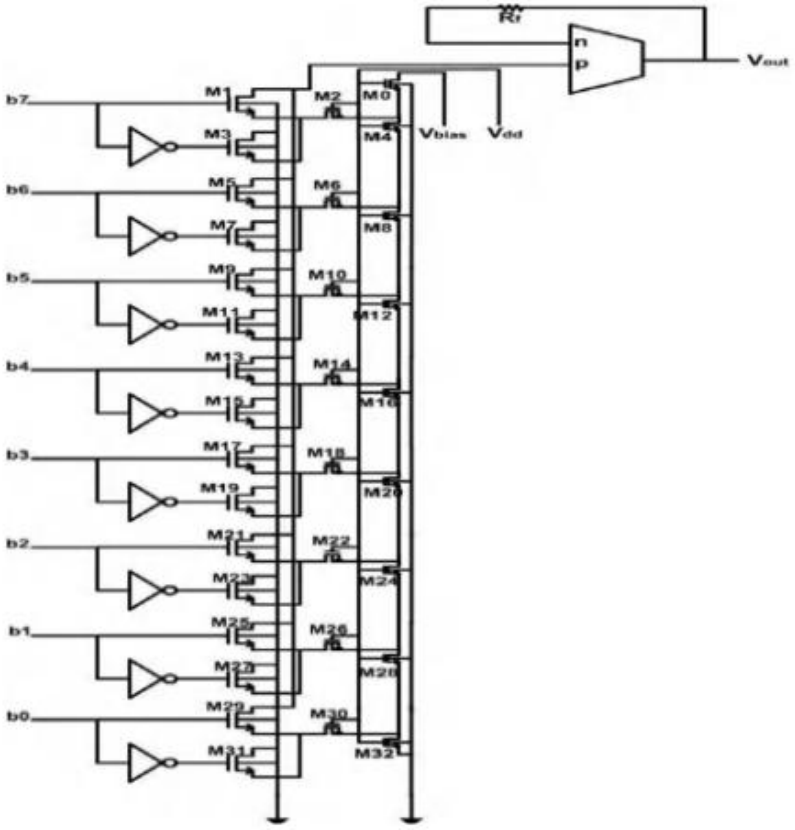
Pada awalnya, sebuah kata biner 8-bit dengan LSB (b0) hingga bit paling signifikan (b7) dianggap sebagai input digital dari jaringan switching digital. Bit input dilewatkan melalui inverter/buffer CMOS. Fungsi utama inverter/buffer ini adalah sebagai pemulih level logika. Buffer membersihkan sinyal yang terdegradasi atau bising dan memberikan sinyal penggerak output ke gerbang MOSFET, sedangkan inverter memberikan output komplementer yang dibutuhkan untuk switching. Output buffer/inverter ini mengontrol pasangan transistor NMOS, sehingga bagaimana pasangan NMOS membentuk sakelar yang dikontrol secara digital.

Sebuah sakelar menentukan apakah arus suatu bit mengalir ke node penjumlahan atau ke ground/beban dummy. Pengalihan ke ground/beban dummy menghilangkan kontribusi bit sambil menjaga beban ladder tetap konstan. Ketika suatu bit berlogika tinggi, satu transistor dalam pasangan tersebut menyala dan mengarahkan arus ke output. Ketika suatu bit berlogika rendah, transistor lainnya menyala dan mengalihkan arus dari output. Mekanisme pengalihan ini mengubah kata input biner menjadi kontribusi arus yang sesuai untuk tahap selanjutnya. Tahap kedua adalah jaringan ladder R-2R berbasis MOSFET, yang menggantikan ladder resistor konvensional dengan memanfaatkan transistor MOS yang beroperasi di wilayah trioda.

Dalam ladder R-2R berbasis resistor tradisional, resistor presisi digunakan untuk menghasilkan pembagian tegangan atau arus berbobot biner. Namun, dalam CMOS VLSI, resistor membutuhkan area chip yang besar dan tidak mudah diskalakan. Sebaliknya, MOSFET dapat dikonfigurasi untuk berperilaku sebagai resistor setara ketika diberi bias dengan benar di wilayah trioda, secara signifikan mengurangi luas silikon sambil mempertahankan fungsi ladder yang sama. Dalam desain ini, jaringan ladder membentuk dasar pembobotan biner DAC.

Setiap cabang ladder memberikan sebagian kecil arus referensi yang sesuai dengan posisi bitnya. Misalnya, cabang MSB membawa setengah dari tegangan atau arus yang mengalir melalui ladder, bit berikutnya membawa seperempat, dan seterusnya, hingga cabang LSB, yang membawa 1/128 dari kontribusi referensi. Distribusi pembobotan biner ini sangat

penting untuk konversi digital-ke-analog yang akurat. Ini memastikan bahwa output DAC secara tepat merepresentasikan jumlah tertimbang dari kata input digital. Alih-alih menggunakan resistor yang tersedia secara komersial, MOSFET memungkinkan pencocokan arus yang tepat dan peningkatan linearitas serta monotonisitas DAC.




**Gambar 14.4** DAC 8-bit yang diusulkan.

Sekarang, saat membahas operasi gabungan jaringan tangga dan sakelar digital, setiap bit digital yang dikendalikan oleh pasangan sakelar NMOS menentukan apakah arus berbobot dari cabang tangganya ditambahkan ke simpul penjumlahan atau ke ground/beban dummy. Ketika sebuah bit bernilai logika tinggi, sakelar menghubungkan arus berbobotnya ke simpul penjumlahan, berkontribusi pada output.

Di sisi lain, jika bit bernilai logika rendah (0), arus diarahkan ke ground, menghilangkan kontribusinya. Dengan demikian, berdasarkan input digital, simpul penjumlahan mengumpulkan semua komponen arus aktif dan membentuk sistem terdistribusi biner berbobot total. Secara matematis, hubungan ini dapat dinyatakan sebagai

$$I = V_{R1}[b_7 + b_6 + b_5 + b_4 + b_3 + b_2 + b_1 + b_0] \tag{14.6}$$
$$P \rightarrow 4R \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 8 \rightarrow 16 \rightarrow 32 \rightarrow 64 \rightarrow 128$$

di mana $b_0$ hingga $b_7$ sesuai dengan bobot biner bit input.

Tahap akhir dan krusial dari DAC yang diusulkan adalah koneksi OTRA dengan sakelar digital dan jaringan ladder. OTRA mengubah arus total dari ladder menjadi tegangan analog operasi. Tidak seperti penguat operasional (OPAMP) tradisional, OTRA beroperasi dengan ground virtual dan memastikan bahwa node penjumlahan tetap pada potensial yang stabil. Sesuai dengan matriks port pada Gambar 14.1b, kita dapat mengatakan bahwa:

$$I_p = I_n \tag{14.7}$$

Persamaan di atas menyatakan bahwa arus pada terminal positif dan negatif adalah sama. Sifat ini digunakan untuk mempertahankan arus input konstan dalam konfigurasi DAC. Arus berbobot biner dari rangkaian tangga masuk ke input OTRA, dan karena secara virtual terhubung ke ground, tidak terjadi penurunan tegangan pada node input. Sebaliknya, seluruh arus melewati resistor umpan balik (RF) seperti yang ditunjukkan pada Gambar 14.4. Jadi persamaan (14.1) dapat ditulis ulang sebagai:

$$V_0 = I_n R_F \tag{14.8}$$

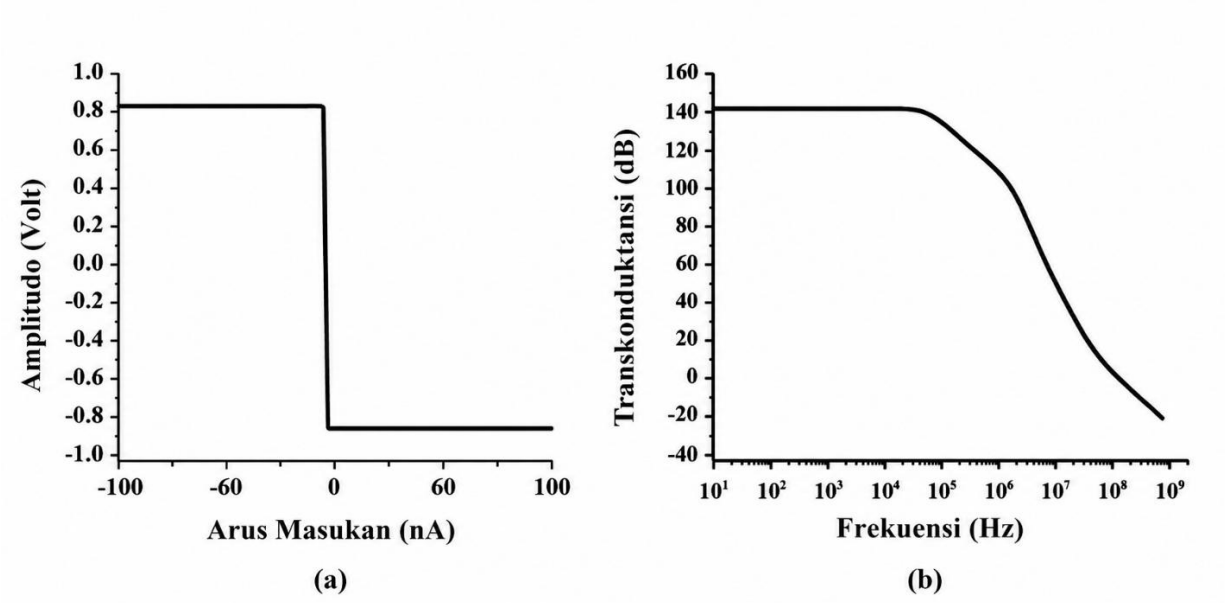
Impedansi input yang sangat rendah dan ground virtual yang stabil meminimalkan kesalahan tegangan dan efek pembebanan OTRA. Selain itu, resistor umpan balik tidak hanya mengontrol faktor penskalaan tegangan tetapi juga menstabilkan output konverter. Dengan menggabungkan akurasi jaringan ladder, switching kecepatan tinggi transistor NMOS, dan konversi arus-ke-tegangan yang presisi dari OTRA, DAC mencapai resolusi tinggi dengan linearitas yang baik dan kinerja CMOS yang efisien.

## 14.3 HASIL SIMULASI DAN PEMBAHASAN OTRA

Untuk analisis grafis dari DAC yang diusulkan, pada awalnya, OTRA telah diimplementasikan menggunakan parameter proses CMOS 180 nm dan disimulasikan dalam perangkat lunak PSPICE. Dalam simulasi tersebut, tegangan catu daya $V_{DD} = -V_{SS} = 0.9$ V, dengan $V_B = -0.2$ V. Tegangan bias ini ($V_B$) memastikan bahwa seluruh transistor MOS dari DAC beroperasi pada wilayah operasi yang benar dan stabil. Disipasi daya keseluruhan dari rangkaian adalah $553\ \mu$W, yang menunjukkan keseimbangan yang efektif antara penguatan tinggi, kinerja pita lebar (*wideband*), dan operasi daya rendah. Karakteristik keluaran dari OTRA memberikan wawasan penting ke dalam perilaku masukan-keluaran, stabilitas, dan kinerja keseluruhannya.

Gambar 14.5a menunjukkan grafik arus input ($I_n$) vs. tegangan, dari -100 hingga +100 µA. Dapat dilihat bahwa tegangan keluaran dipertahankan pada tegangan tinggi yang relatif stabil sekitar +0,88 V untuk sebagian besar nilai arus input negatif. Ini berarti bahwa OTRA beroperasi di wilayah liniernya dengan distorsi yang sangat kecil dan kemampuan tinggi untuk menolak perubahan di domain arus negatif. Ketika arus input mendekati nol, ada perubahan mendadak yang tinggi pada tegangan keluaran yang turun dari wilayah saturasi positif (+0,88 V) ke wilayah saturasi negatif (-0,88 V) dengan sangat cepat. Transisi yang curam ini dihasilkan

dari penguatan transresistansi tinggi pada OTRA, yang mengubah perubahan input kecil di sekitar arus nol menjadi perubahan tegangan besar pada output.




**Gambar 14.5** (a) Analisis arus-tegangan DC dari OTRA. (b) Penguatan transresistansi dari OTRA.

Di luar titik transisi ini, untuk nilai arus masukan positif, tegangan keluaran stabil pada -0,88 V dan tetap hampir konstan hingga +100 µA. Karakteristik saturasi ini di area positif dan negatif menunjukkan bahwa OTRA memang berfungsi sebagai konverter arus-ke-tegangan penguatan tinggi dengan batas saturasi yang jelas yang ditentukan oleh rel suplai ±0,9 V. Perbedaan minimal antara tingkat saturasi (+0,88 V dan –0,88 V) dan rel suplai optimal menunjukkan penurunan tegangan realistis yang disebabkan oleh ambang batas perangkat MOS internal dan kondisi bias.

Grafik tegangan-arus pada Gambar 14.5b menunjukkan bahwa respons simetris arus masukan aman, dan arus masukan dapat digunakan dalam pemrosesan sinyal analog, seperti pada filter, osilator, konverter, dan DAC. Sebaliknya, transisi peralihan yang tajam pada titik perpotongan nol menyiratkan bahwa ia sangat sensitif terhadap perubahan kecil pada arus masukan. Sifat-sifat ini menjadikan OTRA sangat baik dalam aplikasi sirkuit presisi tinggi dan sensitivitas tinggi. Secara keseluruhan, temuan membuktikan bahwa kinerja dan efisiensi OTRA menjadikannya kandidat yang baik untuk mengimplementasikan DAC tegangan rendah dan daya rendah.

Penguatan transresistansi (dalam dBΩ), terhadap frekuensi (dalam Hz), sebagai grafik dari OTRA, ditunjukkan pada Gambar 14.5b. Berdasarkan grafik tersebut, disimpulkan bahwa OTRA mampu mempertahankan penguatan transresistansi yang sangat tinggi dan konstan sekitar $150$ dBΩ pada frekuensi rendah, menghasilkan penguatan lingkaran terbuka (*open-loop*) DC sekitar $147$ dBΩ (sekitar $23$ MΩ). Informasi ini menunjukkan kualitas OTRA dalam memperkuat frekuensi rendah dengan pelemahan sinyal yang kecil. Selain itu, seiring dengan

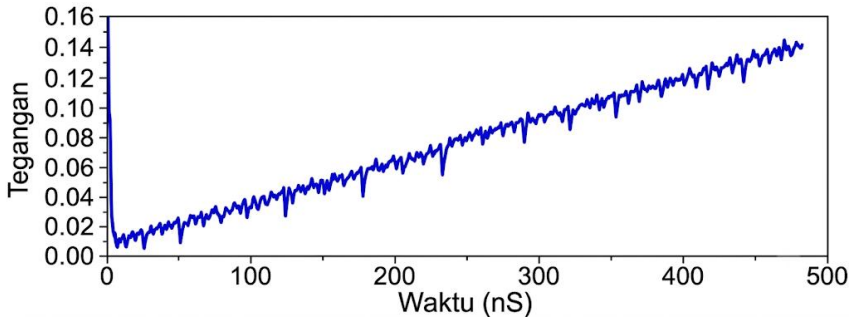
meningkatnya frekuensi secara bertahap pada rentang frekuensi menengah ($10^6 - 10^7$ Hz), sebagai akibat dari kapasitansi parasit dan efek kompensasi internal, penguatan mulai menurun secara perlahan. Pada wilayah frekuensi yang lebih tinggi ($10^8 - 10^9$ Hz), penurunan (*roll-off*) menjadi lebih nyata karena lebar pita (*bandwidth*) terbatas dari penguat mendominasi responsnya. Penguatan transresistansi turun di bawah $0$ dBΩ begitu frekuensi mencapai (di atas $10^{10}$ Hz). Dari hasil simulasi, disimpulkan bahwa resistansi masukan dari OTRA adalah $17.5\ \Omega$, dan produk penguatan-lebar pita (*gain-bandwidth product* atau GBW) dihitung sebesar $2300$ GHzΩ. Hasil ini menyoroti kemampuan penguat untuk mempertahankan penguatan tinggi pada rentang frekuensi yang luas. Secara keseluruhan, analisis grafis ini menunjukkan rentang operasi praktis dari DAC berbasis OTRA yang diusulkan untuk aplikasi pemrosesan sinyal analog.

**Hasil Simulasi DAC yang Diusulkan**

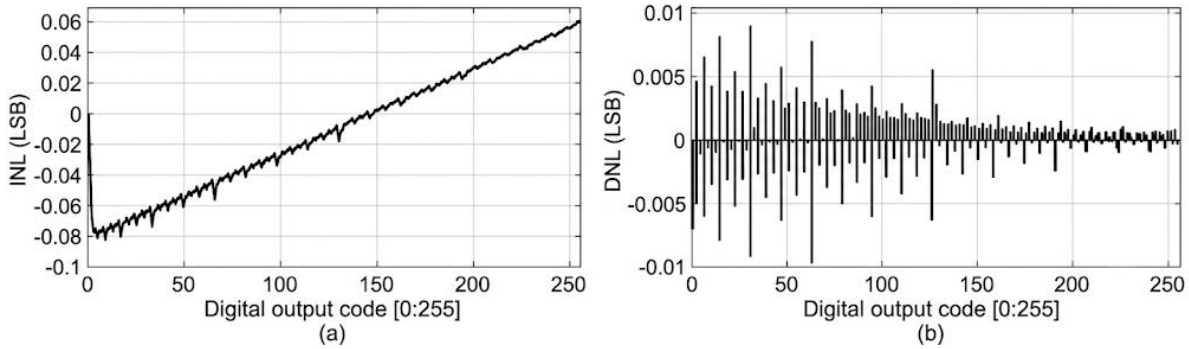
Hasil simulasi DAC 8-bit yang diusulkan, yang dirancang menggunakan teknologi CMOS 180 nm, ditunjukkan pada subbagian berikut, di mana semua simulasi dilakukan dalam perangkat lunak Cadence PSPICE dengan tegangan suplai 1,8 V. Untuk mengevaluasi kinerja kecepatan tinggi DAC yang diusulkan, simulasi dilakukan dengan urutan input digital frekuensi operasi 500 MHz. Gambar 14.6 menunjukkan bentuk gelombang keluaran tangga yang diharapkan dari DAC 8-bit, di mana setiap langkah sesuai dengan konversi kode input digital inkremental menjadi tegangan analog yang setara. Ayunan tegangan keluaran maksimum mencapai sekitar 0,14 V, yang sesuai dengan nilai teoritis yang diperoleh dari jaringan resistor untuk $R_F$ = 5 kΩ.

Selain itu, pada frekuensi input tinggi ini, karena peralihan simultan dari beberapa transisi, terjadi gangguan kecil pada batas kode utama. Namun, akurasi keluaran keseluruhan tetap sama karena semua gangguan tetap berada dalam batas yang dapat ditoleransi. Meskipun demikian, output DAC menjadi monoton, di mana dengan kode input digital yang lebih tinggi, ia selalu menghasilkan tegangan output yang lebih tinggi tanpa kehilangan level input apa pun. Grafik juga menunjukkan waktu penyelesaian yang cepat dan penundaan propagasi yang rendah, yang menegaskan kemampuannya untuk menangani urutan input digital frekuensi tinggi. Lebih lanjut, hasil tersebut membuat DAC cocok untuk konversi data kecepatan tinggi dan aplikasi sinyal campuran yang berorientasi komunikasi secara konsisten pada frekuensi operasi 500 MHz.

Selanjutnya, kinerja statis DAC telah dievaluasi dalam hal Non-Linearitas Integral (INL) dan Non-Linearitas Diferensial (DNL). Simulasi dilakukan dalam kondisi yang sama, menggunakan Cadence PSPICE dengan catu daya 1,8 V, dengan rentang frekuensi input yang diperluas hingga 500 MHz. Untuk mengevaluasi akurasi dan stabilitas output DAC di berbagai laju transisi input, dan memverifikasi kesesuaiannya untuk aplikasi pemrosesan sinyal analog kecepatan tinggi dan presisi, INL/DNL telah dilakukan.




**Gambar 14.6** Keluaran DAC berbasis OTRA 8-bit pada 500 MHz.




**Gambar 14.7** (a) Plot INL pada 500 MHz. (b) Plot DNL pada 500 MHz.

Delapan kata masukan dari 00000000 hingga 11111111 digunakan sebagai stimulus uji untuk mengukur INL. Karakteristik transfer linier ideal dibandingkan dengan respons keluaran. Dapat dilihat dari Gambar 14.7a bahwa INL terburuk pada frekuensi masukan maksimum 500 MHz dibatasi dalam ±0,06 LSB, yang jauh di bawah batas toleransi yang dapat diterima untuk DAC 8-bit. Ini menunjukkan bahwa DAC memiliki tingkat monotonisitas dan presisi transfer linier yang tinggi di seluruh rentang dinamisnya.

DNL diukur dengan cara yang sama seperti menghitung perbedaan antara arus keluaran untuk dua kata masukan berurutan, dikurangi satu arus LSB ideal, dan kemudian dinormalisasi oleh LSB. Seperti yang terlihat pada Gambar 14.7b, nilai DNL juga dijaga sangat dekat dengan ±0,01 LSB untuk semua kata masukan, bahkan pada laju masukan tertinggi 500 MHz. Kesimpulan ini membuktikan bahwa DAC bebas dari kode yang hilang dan memiliki ukuran langkah konstan di seluruh rentang konversinya.Analisis kinerja DAC yang diusulkan telah disajikan dalam Tabel 14.3.

**Tabel 14.3** Evolusi kinerja DAC 8-bit yang diusulkan

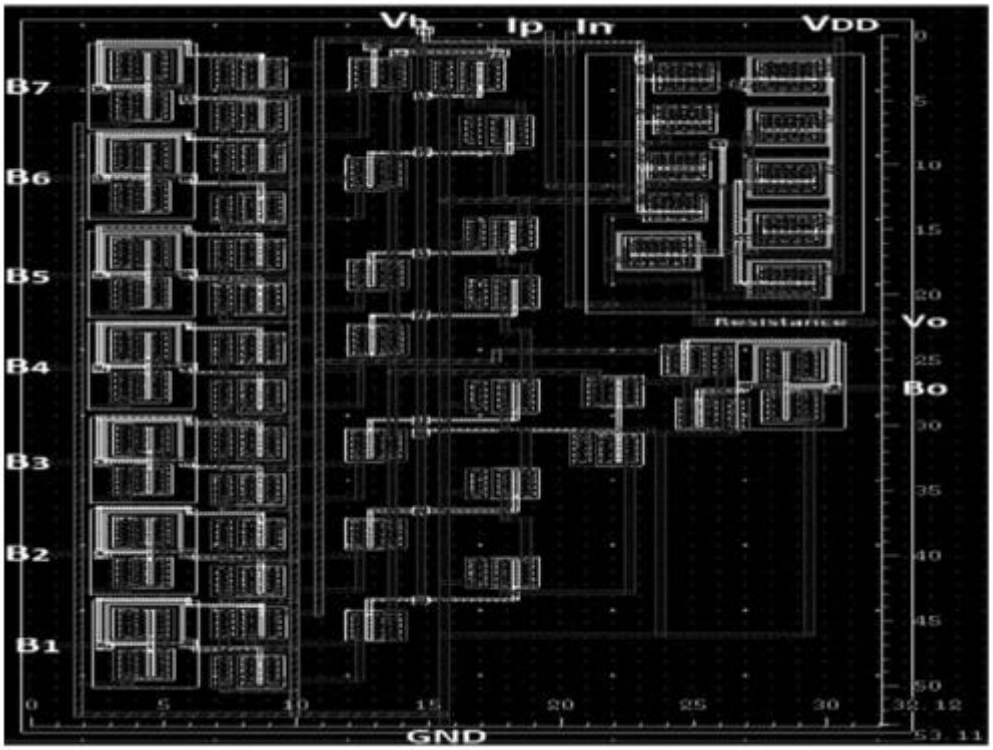
| **Parameter** | **Nilai** |
|---|---|
| Resolusi | 8 bit |
| Laju pembaruan | 500 MHz |
| INL | ±0,06 LSB |
| DNL | ±0,01 LSB |
| Tegangan catu daya | 1,8 V |

| | |
|---|---|
| Teknologi | GPDK 180 nm |

Desain yang diusulkan menawarkan resolusi data 8-bit, sehingga cocok untuk aplikasi presisi menengah hingga tinggi melalui keseimbangan yang baik antara kompleksitas perangkat keras dan akurasi keluaran. Sirkuit ini mendukung bandwidth hingga 500 MHz, yang menunjukkan kemampuannya untuk beroperasi secara efisien dalam kondisi masukan kecepatan tinggi, yang penting untuk sistem komunikasi dan pemrosesan sinyal modern. Lebih lanjut, kinerja linearitas statis telah diverifikasi melalui karakteristik INL dan DNL yang diukur. INL dan DNL masing-masing adalah ±0,06 dan ±0,01 LSB, menunjukkan linearitas global yang sangat baik dengan deviasi minimal dari kurva transfer ideal, yang mengkonfirmasi keseragaman langkah yang akurat tanpa kode yang hilang selama konversi. Bersama-sama, parameter-parameter ini menetapkan akurasi tinggi dan perilaku monotonik DAC di seluruh rentang input.

Tata letak DAC 8-bit yang diusulkan telah diimplementasikan menggunakan teknologi CMOS GPDK 180 nm, seperti yang ditunjukkan pada Gambar 14.8. Desain ini secara ketat mematuhi semua pemeriksaan aturan desain (DRC) untuk persyaratan lebar, jarak, dan penutup. Dalam tata letak ini, kami merepresentasikan Metal-1 dan Metal-2, yang terutama digunakan untuk mencegah tumpang tindih atau koneksi silang langsung di antara interkoneksi dan menjaga perutean sinyal yang tepat tanpa korsleting.

Susunan transistor dan resistor telah dilakukan dengan memperhatikan simetri sehingga efek pencocokan dan gradien diminimalkan, yang penting untuk linearitas dalam DAC. Penekanan kebisingan substrat dan peningkatan keandalan telah dicapai dengan menggunakan kontak sumur dan cincin pelindung. Dari penggaris tata letak, ukuran inti kira-kira 53,11 µm (L) × 32,12 µm (T), yaitu, ~$1{,}71 \times 10^3$ µm² (≈0,0027 mm²), tidak termasuk bantalan. Penempatan, arah perutean (M1 sebagian besar horizontal secara lokal, M2 vertikal untuk trunk), dan penataan ubin yang teratur dipilih untuk meminimalkan ketidaksesuaian yang disebabkan oleh gradien dan parasit, sehingga mendukung laju pembaruan 500 MHz yang terukur dan linearitas yang terverifikasi (INL ±0,06 LSB, DNL ±0,01 LSB).




**Gambar 14.8** Desain tata letak DAC yang diusulkan.

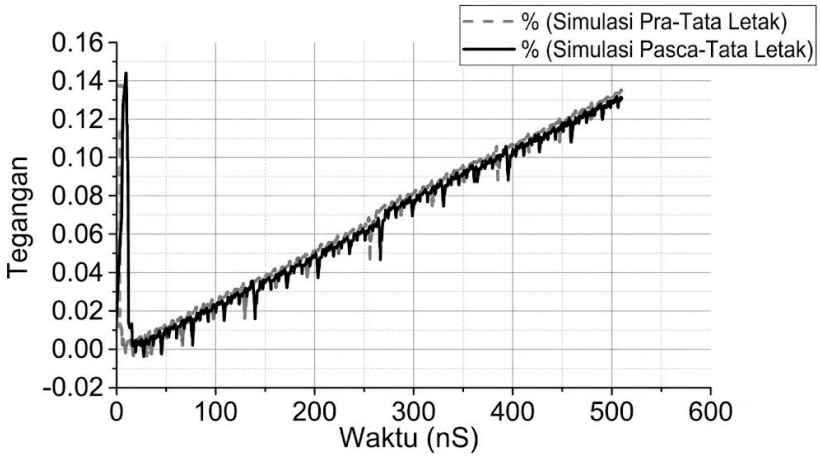
Semua bentuk memenuhi DRC 180 nm yang digunakan dalam GPDK180, termasuk aturan (khas) yang diterapkan adalah, panjang gerbang minimum = 0,18 µm; lebar poli ≥0,22 µm dan poli di atas penutup aktif; Lebar dan jarak minimum M1/M2 ≥0,28 µm; ukuran kontak/VIA1 ~0,26 µm dengan penutup ≥0,06–0,07 µm di lapisan masing-masing; jarak difusi/sumur dan penutup sesuai aturan dek di sekitar keran dan cincin pelindung; kepadatan logam dihormati melalui pengisian jika diperlukan.

Selanjutnya, verifikasi tata letak versus skema (LVS) mengkonfirmasi bahwa implementasi fisik tata letak DAC secara elektrik identik dengan netlist skema asli. Langkah ini memastikan bahwa tidak ada perangkat yang hilang, terhubung secara tidak benar, atau tidak sesuai dalam hal orientasi, dimensi, atau parameterisasi. Sebelum proses fabrikasi, untuk memastikan desain tata letak yang dimaksud sama dengan desain skema, verifikasi LVS sangat penting. Hal ini telah dilakukan untuk memastikan bahwa tata letak fisik akan mencapai hasil yang diharapkan karena tidak ada korsleting yang tidak disengaja, koneksi terbuka, atau ketidaksesuaian perangkat.

Selanjutnya, untuk memeriksa konektivitas daya, koneksi yang benar dari substrat dan kontak sumur telah diperiksa dengan Pemeriksaan Aturan Kelistrikan (ERC). Proses ini memastikan bahwa tidak ada jaringan mengambang, input yang tidak terkait, dan kondisi bias yang salah. ERC juga berguna dalam mengidentifikasi masalah keandalan, yang mungkin termasuk isolasi sumur yang salah, kontak bulk yang hilang, atau domain bias tumpang tindih, yang dapat menyebabkan kebocoran, latch-up, atau kinerja sirkuit yang buruk. Setelah pemeriksaan LVS dan ERC selesai, layout menjalani ekstraksi parasitik atau RC. Untuk menjelaskan efek resistansi dan kapasitansi yang diperkenalkan oleh lapisan interkoneksi

(Metal 1, Metal 2, VIA) dan perutean perangkat, proses ekstraksi ini telah dilakukan. Parasitik yang dihasilkan ini dibaca ulang ke dalam netlist untuk digunakan dalam simulasi pasca-layout. Penambahan elemen RC ini memperpanjang konstanta waktu jaringan switching dan menambahkan beberapa kopling kapasitif kecil antara interkoneksi yang berdekatan, yang memengaruhi waktu penyelesaian, energi glitch, dan penundaan transisi DAC.

Gambar 14.9 mengilustrasikan respons transien dari simulasi pra-layout dan simulasi pasca-layout. Bentuk gelombang hampir seluruhnya tumpang tindih, dan parasitik yang diekstrak tidak terbukti mengganggu fungsionalitas keseluruhan DAC. Respons pasca-layout ditandai dengan penundaan kecil dan distorsi langkah pada transisi kode, yang sesuai dengan yang diharapkan berdasarkan pembebanan RC pada jalur kritis. Linearitas tangga dan kemiringan keluaran keseluruhan sebagian besar tetap terjaga. Lebih lanjut, hasil ini menegaskan kekokohan desain DAC yang diusulkan setelah implementasi layout.




**Gambar 14.9** Variasi pasca-layout DAC.

## 14.4 PENYETELAN ADAPTIF BERBASIS AI UNTUK KINERJA YANG TANGGUH

Selanjutnya, mekanisme penyetelan adaptif berbantuan AI telah diintegrasikan ke dalam DAC untuk memastikan pengoperasian yang stabil dan akurat dalam kondisi PVT yang bervariasi. Dengan menambahkan AI ke sistem, sistem ini terus memantau semua indikator kinerja utama, seperti INL, DNL, pergeseran suhu, dan secara dinamis menyesuaikan parameter bias dan kontrol internal. Mekanisme kalibrasi cerdas ini mengkompensasi semua kinerja yang menurun, atau fluktuasi lingkungan. Pada subbagian berikut, pekerjaan ini dibahas secara detail.

**Arsitektur Kalibrasi Berbasis AI**

- Sistem kalibrasi berbasis AI yang diusulkan dikembangkan dengan tiga unit berikut:
- Unit ekstraksi fitur menghitung ukuran deviasi, seperti offset, berbagai bentuk kesalahan kontrol, dan indeks nonlinier, sebagai input fitur ke model AI. Unit ini beroperasi pada karakteristik/data input waktu nyata, dan data node tegangan/arus internal yang relevan lainnya dimasukkan ke unit berikutnya.

- Ketika informasi dari unit ekstraksi mencapai mesin inferensi yang berbasis Jaringan Saraf Dangkal (*Shallow Neural Network*/SNN), mesin tersebut melakukan estimasi real-time aktual dari parameter kompensasi yang paling tepat. Model SNN dilatih secara offline di bawah strategi pembelajaran terawasi berdasarkan basis data yang luas yang dibuat oleh banyak simulasi DAC dalam berbagai skenario PVT.
- Data yang diproses akhirnya mencapai unit Blok Kontrol Penyetelan. Unit kontrol penyetelan menggunakan teknik kompensasi yang diestimasi untuk melatih parameter sirkuit guna mencapai kinerja terbaik. Setiap sampel pelatihan adalah kumpulan data (INL, DNL, dan suhu) dan parameter penyetelan optimal masing-masing sebagai hasil optimasi tingkat sirkuit. Setelah dilatih, model ini memungkinkan DAC untuk secara otomatis memperbaiki perilakunya secara real-time tanpa kalibrasi ulang eksternal.

Kode sampel pelatihan yang digunakan untuk membuat dataset pelatihan ada di bawah ini:

```
Inputs (X) : [Process, Voltage, Temperature, Raw_INL, Raw_DNL]
Outputs (Y) : [Corrected_Bias, Corrected_Control, Target_INL,
Target_DNL]
```

**Metodologi Pelatihan**

Kasus simulasi dilakukan dengan kondisi PVT acak pada 10.000 kasus untuk membangun kumpulan data lengkap dengan nilai target kompensasi yang sesuai. Data yang dikumpulkan dinormalisasi secara standar, dan dimensinya dikurangi dengan menggunakan metode normalisasi dan pengurangan dimensi standar untuk meningkatkan efisiensi dan daya generalisasi model yang digunakan dengan melatihnya pada data ini. Arsitektur yang digunakan untuk memprediksi bias adaptif dengan bantuan SNN adalah dua lapisan tersembunyi dengan 16 dan 8 neuron, dan masing-masing dilengkapi dengan fungsi aktivasi *Rectified Linear Unit* (ReLU) untuk meningkatkan nonlinieritas dan mempercepat kecepatan konvergensi. Fungsi kerugian *Mean Squared Error* (MSE) digunakan untuk melatih jaringan sedemikian rupa sehingga meminimalkan bias antara prediksi dan data sebenarnya.

Model ini membutuhkan waktu pelatihan dan validasi yang lama untuk mencapai kesalahan regresi kurang dari 0,5% untuk memprediksi nilai koreksi bias terbaik, untuk memverifikasi akurasi yang tinggi dan stabilitas yang sangat baik dalam kondisi PVT apa pun. Selain itu, model SNN yang terlatih dapat diimplementasikan pada chip dengan konsumsi daya rendah karena memiliki struktur yang ringan dan dapat disimpan dalam ruang memori kecil, tidak memerlukan perangkat keras atau sumber daya daya khusus untuk menjalankan kalibrasi model terlatih secara real-time tanpa menimbulkan beban yang sangat besar pada perangkat keras.

**Integrasi Perangkat Keras**

Pengontrol kalibrasi pada chip mencakup modulator bias, yang secara dinamis menyesuaikan bias gerbang transistor berdasarkan output model AI. Loop ini beroperasi secara asinkron di latar belakang, dengan interval kalibrasi ulang periodik (misalnya, setiap 50 ms). Sistem ini tahan terhadap noise digital kecil dan fluktuasi transien karena filter rata-rata geser yang diterapkan pada fitur input. Mesin inferensi AI diimplementasikan menggunakan logika digital berbasis sel standar untuk meminimalkan area dan daya. Pengaturan clock gating

dan penskalaan tegangan digunakan untuk lebih mengoptimalkan efisiensi daya. Arsitektur ini mendukung antarmuka SPI/I2C untuk penggantian eksternal atau pembaruan firmware.

**Hasil dan Dampak Kinerja**

Setelah integrasi model AI, hasil simulasi menunjukkan peningkatan signifikan pada nilai INL dan DNL, serta stabilitas suhu. Dibandingkan dengan kalibrasi statis, penyetelan yang dibantu AI mengurangi variasi INL hingga 6 kali lipat. Sistem yang diperkenalkan mengkonsumsi daya tambahan kurang dari 10 µW dan hanya 2% area silikon tambahan. Arsitektur ini juga dapat diskalakan, memungkinkan penambahan fitur di masa mendatang seperti noise suplai atau kompensasi penuaan pada model. Pembelajaran penguatan untuk penyetelan umpan balik adaptif dalam operasi jangka panjang dapat menjadi topik pembahasan di masa mendatang.

**Visualisasi Berbasis Diagram Alir dari Proses Kalibrasi AI**

Untuk memberikan wawasan yang lebih baik tentang mekanisme kalibrasi yang dibantu AI, dua diagram alir ditunjukkan yang menggambarkan urutan fungsional dan keterlibatan pengambilan keputusan dalam mekanisme penyetelan adaptif dari DAC yang dirancang. Gambar 14.10 menunjukkan urutan keseluruhan tindakan dalam sistem kalibrasi AI. Awalnya, proses dimulai dengan pengoperasian DAC dalam kondisi nominal, diikuti dengan mengekstrak parameter utama seperti INL, DNL, PVT, dll. Parameter ini kemudian dimasukkan ke dalam unit ekstraksi fitur, yang memproses dan mengatur kumpulan data untuk inferensi AI.

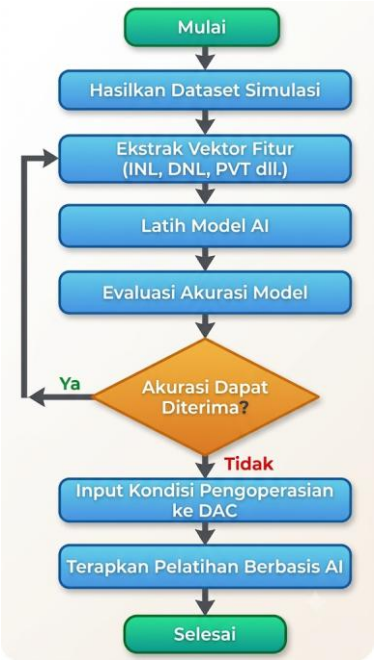
Vektor fitur yang diproses dimasukkan ke dalam SNN yang telah dilatih. Tergantung pada kondisi operasi pada saat itu, model memperkirakan nilai koreksi bias yang sesuai atau penyesuaian parameter kontrol. Nilai-nilai ini dimasukkan ke sirkuit DAC melalui modul penyetelan bias, yang memvariasikan tegangan gerbang, cermin arus, atau arus referensi sesuai kebutuhan. Sistem memasuki siklus periodik, memastikan kalibrasi ulang pada interval yang ditentukan (setiap 50 ms), sehingga meningkatkan ketahanan terhadap berbagai kondisi lingkungan (Gambar 14.11).




**Gambar 14.10** Alur kerja sistem kalibrasi adaptif.

Diagram alir kedua menguraikan proses penalaran mesin inferensi SNN. Setelah memasukkan vektor fitur, sistem pertama-tama melakukan uji pengurangan dimensi untuk menghilangkan fitur yang bising atau berlebihan. Jaringan saraf kemudian memprediksi nilai deviasi kesalahan berdasarkan keadaan saat ini dibandingkan dengan perilaku optimal yang dipelajari.

Jika kesalahan berada dalam toleransi yang dapat diterima, tidak ada koreksi yang dilakukan untuk memastikan efisiensi energi. Namun, jika kesalahan melampaui ambang batas yang ditentukan, mesin AI memulai blok modulasi bias. Penyesuaian hanya diterapkan ketika riwayat koreksi sebelumnya menunjukkan bahwa penyesuaian akan meningkatkan kinerja untuk menghindari osilasi atau ketidakstabilan yang tidak perlu. Diagram alir pada gambar ini menjamin bahwa pembelajaran adaptif didasarkan pada stabilitas, dengan penekanan pada efisiensi energi, pemrosesan latensi rendah, dan perlindungan sirkuit.




**Gambar 14.11** Alur pengambilan keputusan model AI.

**Kode Semu untuk Pelatihan dan Inferensi ANN dalam Penyetelan Adaptif DAC**

Deskripsi tingkat tinggi formal dari program menggunakan campuran bahasa alami yang menyederhanakan konvensi pemrograman untuk menguraikan logika, alih-alih sintaks bahasa pemrograman khusus untuk DAC yang diusulkan, ditunjukkan di bawah ini.

```
Python
#-------------------------------------------------
# Step 1: Dataset Preparation #
#-------------------------------------------------
# Each sample contains input features [INL, DNL, Temp, Vdd]
# and target correction values [Bias_adjust, Current_ref, Gate_
tune] dataset = load_simulation_data("DAC_PVT_dataset.csv")
```

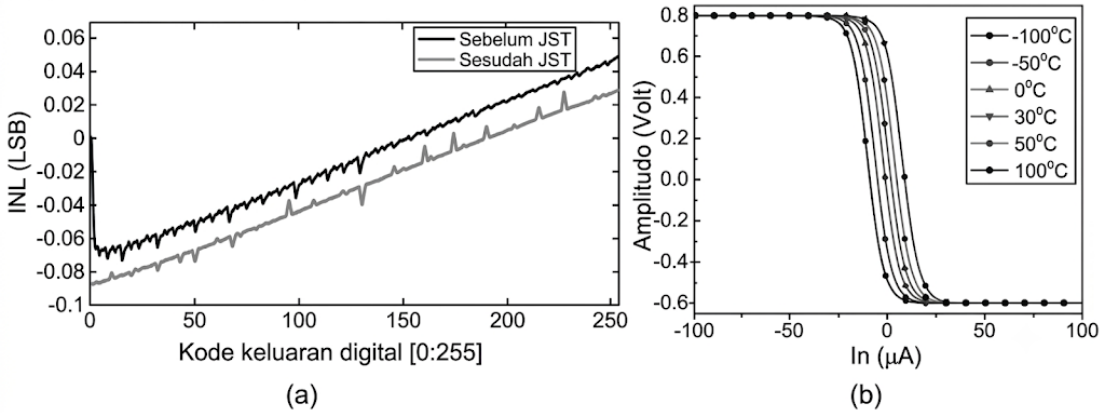
```
 X = dataset.features      # Input features
 Y = dataset.targets       # Optimal compensation values
 # Normalize features for better convergence
 X_norm = normalize(X)
 # Split dataset into training (80%) and Validation (20%)
 X_train, X_val, Y_train, Y_val = split (X_norm, Y, ratio=0.8)
 #-------------------------------------------------
 # Step 2: ANN Model Definition
 #-------------------------------------------------
 # Shallow Neural Network with 2 hidden layers model = ANN ()
 model.add (dense (16, Activation="ReLU", input_dim=4))
 model.add (dense (8, Activation="ReLU"))
 model.add (dense (3, Activation="Linear"))
 # Define optimizer and loss
 model.compile  (optimizer="Adam", learning_rate=0.001,
 loss="MSE")

#-------------------------------------------------
# Step 3: Training #
#-------------------------------------------------
history = model.train(X_train, Y_train,
                      validation_data=(X_val, Y_val),
                      epochs=200, batch_size=32, early_stopping=True)
#-------------------------------------------------
# Step 4: Model Evaluation #
#-------------------------------------------------
error = model.evaluate(X_val, Y_val)
print("Validation Regression Error:", error)
#-------------------------------------------------
# Step 5: AI-Assisted Compensation
#-------------------------------------------------

for test_sample in new_DAC_measurements:
    features = extract_features(test_sample)   # [INL, DNL,
Temp, Vdd]
    features_norm = normalize(features)
    correction = model.predict(features_norm)  # [Bias_adjust,
Current_ref, Gate_tune]
    # Apply correction to DAC
    DAC.apply_correction(correction)
    # Store improved output for comparative plot
    corrected_output = DAC.measure_output()
    save_to_results(corrected_output)
#-------------------------------------------------
# Step 6: Graph Generation
#-------------------------------------------------
plot(before_AI_outputs, label="Before AI Compensation")
```

```
plot(after_AI_outputs, label="After AI Compensation")
#-------------------------------------------------
```

Pertama, kumpulan data variasi PVT dengan koreksi bias optimal yang sesuai disiapkan dan dinormalisasi. ANN, yang dilatih menggunakan fungsi kerugian MSE, belajar memprediksi parameter kompensasi yang meminimalkan INL, DNL, dan drift. Selama inferensi, model terlatih memprediksi koreksi waktu nyata, yang diterapkan pada DAC, menghasilkan kurva keluaran setelah AI yang lebih baik dibandingkan dengan kasus sebelum AI yang tidak dikompensasi.




**Gambar 14.12** (a) Plot INL sebelum dan sesudah ANN pada 500 MHz. (b) Variasi suhu DAC setelah menggunakan ANN.

Analisis komparatif grafik INL yang ditunjukkan pada Gambar 14.12a dengan jelas menunjukkan peningkatan tersebut. Kurva sebelum ANN menunjukkan fluktuasi yang tiba-tiba, sedangkan kurva setelah ANN lebih halus, dengan nilai kesalahan yang dipertahankan dengan baik dalam toleransi target ±0,01 LSB. Selanjutnya, analisis variasi suhu blok OTRA dari -100°C hingga 100°C telah dilakukan untuk memeriksa ketahanan DAC yang diusulkan. Pada Gambar 14.12b, variasi kecil diamati; Namun, penggabungan mekanisme kalibrasi berbantuan AI memastikan kinerja tetap berada dalam rentang toleransi yang ditentukan.

Tabel 14.4 menunjukkan bahwa DAC yang diusulkan lebih unggul daripada desain canggih yang tersedia saat ini. Desain yang diusulkan dibuat dengan proses CMOS 180 nm, sehingga selain, karya-karya tersebut dibuat dengan parameter proses yang sama, yang merupakan dasar yang baik untuk perbandingan. Tidak seperti desain sebelumnya, yang menggunakan tegangan pemasok yang lebih besar (3,3 V dalam) atau menggunakan daya yang sangat besar untuk beroperasi, karya yang diusulkan memiliki disipasi daya yang sangat rendah yaitu 0,543 mW, lebih dari 14 kali lipat dibandingkan dan hampir 50 kali lipat lebih rendah dibandingkan. Kehilangan daya ini dilakukan tanpa mengorbankan kecepatan, dengan DAC mempertahankan laju konversi 500 MHz, yang sebanding dengan dan jauh lebih cepat daripada.

Meskipun memiliki laju konversi yang lebih baik yaitu 1250 MHz, hal itu dicapai dengan mengorbankan resolusi yang lebih buruk (12-bit), dan kinerja linearitasnya lebih buruk

daripada karya ini. Mengenai akurasi, metrik linearitas, yaitu INL dan DNL, menunjukkan peningkatan yang jelas. Desain yang diusulkan menunjukkan INL ±0,01 LSB dan DNL ±0,01 LSB, yang merupakan salah satu keluaran paling konsisten dan tepat dari semua karya yang telah dibandingkan. Namun, desain lain yang digunakan lebih menyimpang (nilai deviasi maksimum adalah 0,33 LSB pada INL). Desain yang diusulkan memiliki area lantai terkecil dari semua desain lain yang dilaporkan. Tingkat kesalahan linearitas yang tinggi dalam karya ini dapat dijelaskan oleh fakta bahwa sistem kalibrasi adaptif yang didukung dengan AI terus mengamati perubahan kinerja dalam kondisi PVT dan memperbaruinya secara real-time.

**Tabel 14.4** Analisis perbandingan DAC yang diusulkan dengan desain serupa.

| | | | | **REFERENSI** |
|---|---|---|---|---|
| Parameter | **[6]** | **[34]** | **[35]** | **Pembahasan Ini** |
| Teknologi (nm) | 180 | 350 | 180 | 180 |
| Tegangan catu daya (V) | 1,8 | 3,3 | 3,3 | 1,8 |
| Resolusi | 8 | 10 | 8 | 8 |
| Laju konversi (MHz) | 500 | 500 | 100 | 500 |
| INL (LSB) | 0,33 | 0,10 | 0,27 | ±0,01 |
| DNL (LSB) | 0,30 | 0,20 | 0,32 | ±0,01 |
| Daya yang dikonsumsi (mW) | 7,88 | 125 | 27,6 | 0,543 |
| Luas tata letak ($mm^2$) | — | 0,6 | 0,024 | 0,0027 |

## 14.5 KOREKSI KINERJA WAKTU NYATA DAC MODE ARUS MELALUI OPTIMASI AI

Artikel ini telah menyajikan desain baru dari DAC mode arus berkecepatan tinggi dan berdaya rendah yang menggunakan algoritma penyetelan adaptif berbantuan AI untuk menawarkan respons yang kuat terhadap perubahan PVT. Topologi tangga R–2R berbasis op-amp juga populer sebagai desain DAC tradisional, tetapi desain ini dikenal terbatas dalam bandwidth, disipasi daya tinggi, ayunan tegangan keluaran rendah, karakteristik keluaran non-linier, dan sensitivitas kebisingan. Desain ini merupakan penanggulangan untuk mengatasi keterbatasan tersebut; desain dalam bab ini beroperasi dalam mode arus, menawarkan bandwidth yang lebih tinggi, kemampuan untuk menggerakkan sirkuit dalam impedansi rendah, dan resolusi tinggi dibandingkan dengan sirkuit mode tegangan standar.

Orisinalitas karya ini terletak pada kombinasi struktur mode arus dan OTRA serta jaringan switching MOS, sehingga memungkinkan pengoperasian dalam pita lebar tanpa mengorbankan linearitas hasil. OTRA memiliki efisiensi dan stabilitas konversi arus-tegangan yang baik dan oleh karena itu, mengatasi kekurangan DAC mode tegangan konvensional. Kombinasi tersebut dapat menghasilkan peningkatan kinerja umum yang luar biasa dan menentukan kebaruan desain yang diusulkan.

Selain itu, mekanisme koreksi INL, DNL, dan pergeseran suhu selama waktu nyata dapat ditambahkan dengan penggabungan metodologi penyetelan adaptif menggunakan ANN. ANN dapat dilatih secara offline dengan basis data PVT yang besar dan disematkan pada chip dalam desain yang ringkas dan efisien untuk memfasilitasi skalabilitas. Dengan menyesuaikan

parameter pengaturan bias internal secara dinamis, hal ini memastikan bahwa kondisi dapat bervariasi selama fabrikasi dan faktor lingkungan berubah dengan parameter eksternal. Dengan memanfaatkan hasil integrasi tunggal dari desain tersebut, INL jauh lebih unggul, dan pengurangan INL yang signifikan tercapai (INL ± 0,01 LSB, DNL ± 0,01 LSB), bersamaan dengan pengoperasian daya ultra-rendah pada 0,543 mW dan dengan kemampuan konversi kecepatan tinggi pada 500 MHz.

# BAB 15
# DESAIN DAN IMPLEMENTASI FPGA PADA PEMBELAJARAN MENDALAM

## 15.1 AKSELERASI PEMBELAJARAN MENDALAM MENGGUNAKAN ARSITEKTUR FPGA

Pembelajaran mendalam menggunakan jaringan saraf multi-lapisan untuk menangani masalah yang menantang seperti pemrosesan suara, identifikasi gambar, dan analitik prediktif. Baik pelatihan maupun inferensi membutuhkan daya pemrosesan yang sangat besar. Karena menawarkan logika yang dapat disesuaikan, komputasi paralel, kemampuan adaptasi waktu nyata, dan efisiensi energi yang tidak tertandingi oleh CPU dan GPU konvensional, *Field-Programmable Gate Arrays* (FPGA) menjadi semakin populer dalam akselerasi perangkat keras pembelajaran mendalam.

Arsitekturnya, yang mencakup irisan Pemrosesan Sinyal Digital (DSP), blok logika yang dapat diprogram, dan memori khusus, memungkinkan optimasi jaringan saraf yang disesuaikan, yang membuatnya sangat cocok untuk aplikasi latensi rendah dan throughput tinggi yang unik dalam sistem tertanam dan kecerdasan buatan. Peneliti dan insinyur secara efektif memetakan primitif pembelajaran mendalam ke perangkat keras berkat perkembangan terkini dalam teknik desain FPGA, seperti HDL dan alat sintesis tingkat tinggi (HLS). Dasar untuk analisis menyeluruh arsitektur FPGA, fitur desain, dan alur kerja yang mendukung efektivitasnya sebagai platform pembelajaran mendalam.

Kecerdasan Buatan (AI) berada di garis depan inovasi teknologi, mendorong kemajuan di berbagai industri mulai dari perawatan kesehatan hingga otomotif. Namun, daya komputasi yang sangat besar yang dibutuhkan untuk aplikasi AI menghadirkan tantangan yang signifikan. Perangkat keras tradisional seperti CPU dan GPU, meskipun kuat, seringkali kesulitan untuk memenuhi tuntutan ini secara efisien. Dalam bab ini, kita akan mengeksplorasi peran transformatif FPGA dalam akselerasi AI.

**FPGA untuk Algoritma Pembelajaran**

Perangkat keras yang dapat diprogram, paralelisme, dan efisiensi energi teknologi FPGA menjadikannya ideal untuk pembelajaran mendalam. Dengan memodifikasi komponen perangkat keras seperti elemen logika, pengali, dan blok memori, FPGA dapat diprogram dan dimodifikasi untuk mempercepat arsitektur jaringan saraf tertentu. Yin dkk. berkontribusi pada peningkatan kinerja dan pengurangan konsumsi daya dibandingkan dengan CPU dan GPU konvensional.

(a) Perangkat Keras yang Dapat Dikonfigurasi Ulang: Seiring evolusi model, FPGA dapat menerapkan akselerator pembelajaran mendalam khusus berkat pemrograman ulang dinamis yang dimungkinkan oleh memori konfigurasi.

(b) Metodologi untuk Konfigurasi Ulang: Bitstream FPGA dapat dengan cepat diperbarui menggunakan antarmuka konfigurasi, memungkinkan penyempurnaan untuk berbagai tugas jaringan saraf dan pembelajaran mendalam.

(c) Paralelisme: Untuk operasi pembelajaran mendalam seperti konvolusi dan perkalian matriks, paralelisme spasial memungkinkan FPGA untuk menangani beberapa komputasi secara bersamaan.

**Arsitektur dan Sumber Daya Khusus**

FPGA terdiri dari interkoneksi yang dapat diprogram, blok I/O, dan blok logika yang dapat dikonfigurasi (CLB) dengan komponen logika (LUT, flip-flop)

(a) Operasi perkalian-akumulasi, yang penting untuk lapisan jaringan saraf seperti konvolusi dan perkalian matriks, didukung oleh slice DSP.
(b) Peta fitur dan bobot disimpan dalam blok memori on-chip (BRAM, UltraRAM) untuk akses lokal yang cepat.
(c) Beban kerja pembelajaran mendalam yang dipipelin, manajemen clock (PLL, MMCM), memberikan pengaturan waktu yang akurat.

**Alur Pengembangan dan Sintesis**

HLS mengotomatiskan pembuatan RTL dan menyederhanakan proses desain dengan langsung mengkonversi C/C++ dan spesifikasi tingkat tinggi lainnya ke bahasa deskripsi perangkat keras (HDL). HLS meningkatkan output perancang dan mempercepat peninjauan dan pembuatan prototipe. Meskipun HLS memungkinkan pembuatan yang lebih cepat dan fleksibilitas dengan mengorbankan penyempurnaan, HDL menawarkan kontrol terbaik dan penggunaan sumber daya yang efisien tetapi membutuhkan banyak tenaga kerja. Filter FIR, mesin perkalian matriks, dan contoh praktis pemetaan langsung lapisan jaringan saraf konvolusional ke sumber daya FPGA adalah contoh aplikasinya. Melalui kemampuan optimasi yang disesuaikan, FPGA telah menunjukkan keunggulan dalam inferensi latensi rendah dan penerapan di edge.

**Perangkat Keras FPGA yang Dapat Dikonfigurasi Ulang**

FPGA sebagai Perangkat Keras yang Dapat Dikonfigurasi Ulang. Di dunia perangkat keras digital, sebagian besar chip dibuat dengan tujuan tetap dan tidak dapat diubah. Misalnya, Sirkuit Terpadu Khusus Aplikasi (ASIC) dirancang secara permanen untuk melakukan satu pekerjaan dan tidak pernah dapat diubah. Inilah kemampuan unik yang mendefinisikan FPGA. Kekuatan intinya adalah kemampuannya untuk dikonfigurasi ulang, memungkinkan perancang untuk memodifikasi logika internalnya agar sesuai dengan persyaratan baru, memperbaiki bug, atau mengoptimalkan kinerja, bahkan lama setelah diterapkan dalam suatu sistem.

**Memori Konfigurasi**

Inti dari kemampuan adaptasi FPGA adalah memori konfigurasinya. Ini adalah cetak biru yang memberi tahu berbagai blok logika dan jalur perutean chip bagaimana cara terhubung dan beroperasi. Cetak biru ini dimuat ke dalam chip sebagai bitstream, yaitu urutan angka satu dan nol yang mewakili desain sirkuit digital lengkap. Pada banyak perangkat populer, seperti FPGA Seri 7, memori ini terdiri dari ribuan sakelar kecil yang disebut CMOS Configuration Latches (CCL). Saat FPGA dinyalakan, pada dasarnya ia seperti papan tulis kosong. Ia harus memuat bitstream dari sumber eksternal, biasanya chip memori non-volatil seperti flash atau EEPROM. Atau, prosesor dapat secara aktif memprogram FPGA melalui antarmuka khusus. Setelah dimuat, bitstream ini mengkonfigurasi setiap elemen fungsional,

dari *Lookup Tables* (LUT) dan flip-flop hingga pengelola clock, menghidupkan desain perangkat keras khusus.

**Metode Konfigurasi Ulang**

Pemrograman FPGA bukan hanya peristiwa sekali saja saat startup. Ada dua pendekatan utama yang ditunjukkan dalam yang dapat dikonfigurasi, tergantung pada kebutuhan aplikasi.

(a) Konfigurasi Ulang Statis: Seluruh perangkat dikonfigurasi saat startup dan tetap tidak berubah selama operasi. Ini adalah pendekatan yang paling mudah dan cocok untuk aplikasi dengan fungsionalitas tetap.

(b) Konfigurasi Ulang Dinamis (Konfigurasi Ulang Parsial): Sebagian dari FPGA dapat diperbarui sementara bagian sistem lainnya tetap beroperasi. Hal ini memungkinkan aplikasi untuk beradaptasi secara real-time—misalnya, mengubah frekuensi clock, mengubah antarmuka memori, atau menerapkan pemulihan kesalahan tanpa mengganggu operasi keseluruhan.

**Antarmuka Konfigurasi**

FPGA mendukung beberapa antarmuka untuk memuat atau memperbarui memori konfigurasi:

- JTAG (*Joint Test Action Group*): Antarmuka standar yang digunakan untuk debugging dan pemrograman.
- SelectMAP: Antarmuka konfigurasi paralel yang menyediakan transfer data lebih cepat.
- SPI (*Serial Peripheral Interface*): Antarmuka serial yang banyak digunakan untuk konfigurasi dan komunikasi.
- ICAPE2: Port akses konfigurasi internal yang digunakan untuk konfigurasi ulang sebagian dari dalam FPGA itu sendiri.

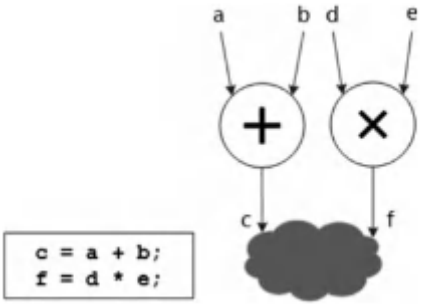
**FPGA untuk Pembelajaran Mendalam**

Tidak seperti GPU yang mengandalkan instruksi vektor, FPGA mencapai paralelisme data melalui arsitektur spasialnya. Kompilator secara otomatis mengidentifikasi operasi independen dan mengimplementasikannya sebagai jalur perangkat keras terpisah dan simultan. Hal ini memungkinkan FPGA untuk membuat jalur data perangkat keras khusus untuk setiap set instruksi, yang mengarah pada penggunaan silikon yang lebih efisien dibandingkan dengan unit fungsional tetap pada CPU dan GPU.

**Paralelisme: Pendekatan Spasial**

Prosesor tradisional seperti CPU dan bahkan GPU mencapai paralelisme dengan mengeksekusi instruksi yang sama pada beberapa bagian data, seperti yang disajikan pada Gambar 15.1. Ini adalah pendekatan yang efektif namun kaku. Alih-alih mengambil dan mengeksekusi instruksi, FPGA dikonfigurasi untuk menjadi sirkuit itu sendiri. Jika model pembelajaran mendalam Anda membutuhkan sepuluh operasi berbeda yang dapat berjalan secara independen, kompiler dapat memetakan operasi-operasi ini ke sepuluh sirkuit perangkat keras fisik yang berbeda pada fabrik FPGA yang semuanya berjalan secara bersamaan. Ini adalah paralelisme perangkat keras sejati. Gambar berikut mengilustrasikan

contoh adder dan multiplier, yang dijadwalkan untuk dieksekusi secara bersamaan sambil beroperasi pada input yang terpisah.




**Gambar 15.1** Eksekusi tugas secara paralel.

**Efisiensi Energi**

Dalam dunia AI, terutama untuk perangkat yang digunakan di "ujung" (seperti kamera pintar atau sensor industri), konsumsi daya merupakan masalah yang sangat penting. Prosesor standar membuang banyak energi untuk mengambil, mendekode, dan menjadwalkan instruksi. FPGA menghilangkan overhead ini. Logika dikonfigurasi langsung ke dalam silikon, menciptakan jalur data khusus yang dioptimalkan untuk satu tujuan, seperti operasi perkalian dan penjumlahan yang membentuk tulang punggung jaringan saraf. Hal ini menjadikannya pilihan ideal untuk aplikasi yang menggunakan daya baterai atau aplikasi dengan keterbatasan termal.

**Kustomisasi: Perangkat Keras yang Disesuaikan dengan Model Anda**

Mungkin fitur paling ampuh dari FPGA adalah kustomisasinya yang tak terbatas. Tidak seperti CPU atau GPU, yang memiliki set instruksi tetap dan arsitektur yang kaku, FPGA adalah kanvas kosong. Hal ini memungkinkan para perancang untuk menciptakan solusi perangkat keras yang disesuaikan secara sempurna dengan kebutuhan spesifik model pembelajaran mendalam. Kustomisasi ini mencakup beberapa area:

(a) Presisi Data Kustom: Jika jaringan saraf dapat mencapai akurasi tinggi menggunakan bilangan bulat 8-bit atau 4-bit alih-alih bilangan floating-point 32-bit standar, Anda dapat membangun logika pada FPGA yang menggunakan tipe data yang lebih kecil ini.

(b) Struktur Memori yang Dioptimalkan: Anda dapat merancang hierarki memori dan pola akses kustom.

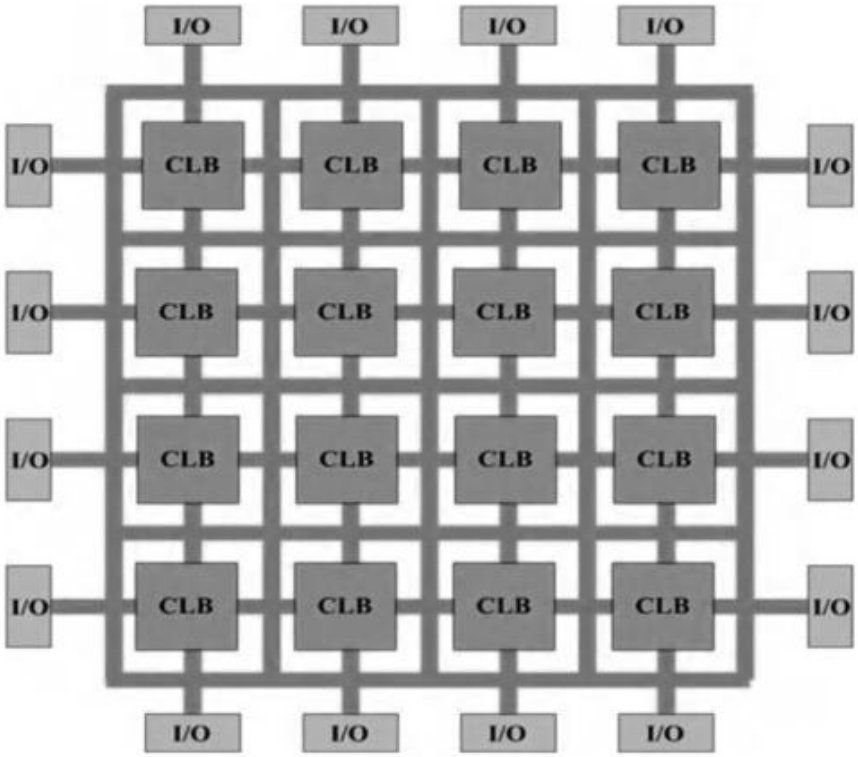
(c) Operator Khusus: Jika model Anda menggunakan operasi matematika yang unik atau tidak standar, Anda dapat merancang blok perangkat keras khusus pada FPGA untuk mempercepatnya, sesuatu yang tidak mungkin dilakukan pada chip arsitektur tetap.

Bab ini dimulai dengan mengeksplorasi arsitektur fundamental FPGA, menyoroti CLB, interkoneksi, dan sumber daya khusus yang membedakannya dari prosesor fungsi tetap. Memahami fitur arsitektur ini sangat penting, karena secara langsung memengaruhi bagaimana algoritma dipetakan ke perangkat keras dan bagaimana kinerja, daya, dan pemanfaatan sumber daya dikelola.

(a) HDL, seperti VHDL dan Verilog, menyediakan kontrol yang lebih rinci atas sumber daya perangkat keras dan telah menjadi metode tradisional untuk pengembangan FPGA.
(b) HLS, metodologi yang lebih baru yang memungkinkan perangkat keras untuk dijelaskan menggunakan bahasa pemrograman tingkat tinggi seperti C/C++ atau OpenCL, sehingga mempercepat pengembangan dan membuat desain FPGA lebih mudah diakses oleh para insinyur perangkat lunak.

## 15.2 ARSITEKTUR FPGA

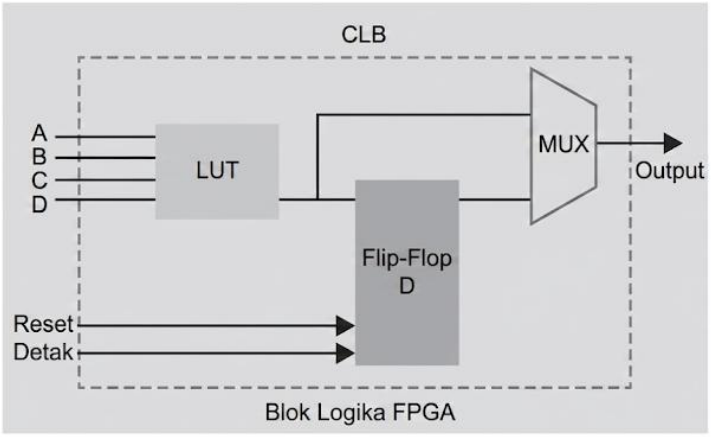
Arsitektur FPGA umum terdiri dari CLB, interkoneksi yang dapat diprogram, dan Blok I/O pada Gambar 15.2. Komponen-komponen ini disusun dalam susunan dua dimensi yang dapat diprogram untuk membuat sirkuit digital khusus.




**Gambar 15.2** Arsitektur FPGA.

**Blok Logika yang Dapat Dikonfigurasi (CLB)**

Setiap CLB berisi LUT, Flip-Flop (FF), dan multiplekser, seperti yang ditunjukkan pada Gambar 15.3. LUT mengimplementasikan fungsi logika kombinasi, FF menyediakan penyimpanan untuk logika sekuensial, dan multiplekser mengarahkan sinyal di dalam blok.




**Gambar 15.3** Arsitektur CLB.

**Elemen logika (LUT, flip-flop, MUX)**

Tabel Pencarian (LUT) adalah komponen yang bertanggung jawab untuk mengimplementasikan semua logika kombinasi (logika yang hanya bergantung pada input saat ini, seperti AND, OR, dan XOR). Anda dapat membayangkan LUT sebagai memori kecil dan cepat yang menyimpan tabel kebenaran yang telah ditentukan sebelumnya. Input ke LUT bertindak sebagai "alamat" ke memori ini, dan outputnya adalah satu bit data yang disimpan di alamat tersebut. Karena isi memori ini (tabel kebenaran) dapat diprogram, LUT dapat dikonfigurasi untuk mengimplementasikan fungsi Boolean apa pun dari inputnya. Misalnya, baris Verilog sederhana seperti ini menggambarkan fungsi yang murni kombinasi:

```
// This logic function's truth table will be programmed into a
LUT.
assign Y = (A & B) | (~C);
```

Alat sintesis akan menganalisis persamaan ini dan memprogram tabel kebenaran LUT untuk menghasilkan output yang benar “Y” untuk setiap kemungkinan kombinasi input “A,” “B,” dan “C.” Flip-flop adalah komponen yang bertanggung jawab untuk logika sekuensial—logika yang memiliki memori dan bergantung pada keadaan dari siklus clock sebelumnya. Ini adalah elemen penyimpanan fundamental yang dapat menyimpan satu bit data. Fitur utamanya adalah hanya memperbarui nilai yang tersimpan pada peristiwa tertentu, biasanya tepi naik sinyal clock. FF sangat penting untuk membuat mesin keadaan, penghitung, register, dan pipeline. Dalam HDL, flip-flop dijelaskan menggunakan proses berdetak:

```
// This code describes a flip-flop with a synchronous reset
// It will be synthesized into a physical flip-flop within a CLB
always @(posedge clk or posedge reset) begin
    if (reset)
        Q <= 0;
    else
        Q <= D;
```

end

**Interkoneksi dan Sumber Daya Perutean yang Dapat Diprogram**

Meskipun CLB bertindak sebagai otak kecil FPGA, mereka tidak berguna secara terpisah. Agar dapat bekerja bersama, mereka membutuhkan cara untuk berkomunikasi. Komunikasi ini ditangani oleh interkoneksi yang dapat diprogram. Sakelar dan kabel yang dapat diprogram yang menghubungkan CLB, blok I/O, BRAM, dan DSP. Menyediakan konektivitas sinyal antar blok.

```
assign Y = A & B; // route A,B -> AND -> Y
```

**Gerbang menuju dunia: Blok I/O**

Blok Input/Output (IOB) bertindak sebagai antarmuka antara logika internal FPGA dan dunia luar. Blok ini dapat dikonfigurasi sebagai port input, output, atau bidirectional dan mendukung berbagai standar kelistrikan seperti LVCMOS, LVTTL, dan LVDS. Selain itu, IOB modern seringkali mencakup fitur-fitur seperti kekuatan penggerak yang dapat diprogram, pensinyalan diferensial, dan serialisasi. Meskipun dalam HDL Anda hanya mendeklarasikan input dan output, karakteristik kelistrikan sebenarnya didefinisikan melalui file batasan yang memetakan pin dan standar.

```
module io_example(input wire clk,wire data_in, output reg
data_out);
```

Jangan tentukan pin fisik mana pada chip yang terhubung ke data_in, atau tegangan apa yang harus digunakan. Informasi fisik ini disediakan dalam file batasan terpisah (misalnya, file .XDC untuk FPGA Xilinx/AMD).

```
# In the constraint file, map the logic to the physical chip.
# Assign the logical port “data_in” to physical pin W5 on the
package. set_property PACKAGE_PIN W5 [get_ports data_in]
# Tell pin W5 to operate using the 3.3V LVCMOS electrical
standard.   set_property   IOSTANDARD   LVCMOS33 [get_ports
data_in]
```

**Sumber Daya Khusus**

**Irisan DSP (Operasi Perkalian-Akumulasi)**

Blok DSP FPGA adalah unit perangkat keras khusus di dalam FPGA, yang dirancang untuk mempercepat tugas pemrosesan sinyal seperti FFT dan filter FIR dengan menyediakan pengali, akumulator, dan ALU (*Arithmetic Logic Unit*) berkinerja tinggi. Blok-blok ini menawarkan keunggulan kinerja dan sumber daya yang signifikan dibandingkan dengan mengimplementasikan fungsi-fungsi tersebut dalam logika tujuan umum, memungkinkan algoritma kompleks dengan kecepatan tinggi, yang sangat penting untuk aplikasi di bidang

video, telekomunikasi, dan AI. Paralelisme masif dimungkinkan dengan mengimplementasikan ratusan atau ribuan unit MAC (Multiply-Accumulate Unit) secara paralel di seluruh fabrik FPGA. Latensi yang sangat rendah dan deterministik. Operasi diprogram secara langsung, menghindari overhead variabel dari penjadwalan dan cache sistem operasi.

**Block RAM (BRAM), ultraRAM, distributed RAM**

Salah satu kekuatan utama FPGA terletak pada sumber daya memori khusus mereka, yang memberikan fleksibilitas dan kinerja kepada perancang yang biasanya tidak tersedia pada CPU atau GPU standar. Alih-alih hanya mengandalkan memori eksternal, FPGA menyematkan beberapa jenis memori on-chip yang dioptimalkan untuk berbagai kasus penggunaan. BRAM adalah blok memori khusus berukuran tetap yang disematkan di seluruh fabrik FPGA. BRAM dapat dikonfigurasi sebagai RAM port tunggal atau port ganda, FIFO, atau ROM, sehingga ideal untuk buffering, LUT, dan penyimpanan data yang dekat dengan logika. Tidak seperti cache di CPU, BRAM sepenuhnya berada di bawah kendali perancang, memastikan pola akses deterministik untuk aplikasi real-time.

Tersedia di FPGA kelas atas, UltraRAM menawarkan kapasitas yang jauh lebih besar daripada BRAM sambil mempertahankan kecepatan on-chip. UltraRAM memungkinkan perancang untuk membangun buffer data yang lebih dalam dan menyimpan model besar atau peta fitur langsung di dalam FPGA, mengurangi ketergantungan pada memori DDR off-chip yang lebih lambat.

Kemampuan ini memberikan keunggulan pada FPGA untuk akselerator AI dan aplikasi big data, di mana bandwidth memori seringkali menjadi hambatan. Tidak seperti BRAM dan UltraRAM, RAM terdistribusi memanfaatkan LUT itu sendiri untuk mengimplementasikan struktur memori yang kecil dan fleksibel. Ini sangat berguna untuk kebutuhan penyimpanan yang sangat terlokalisasi seperti tabel koefisien, register geser, atau FIFO kecil. RAM terdistribusi terhubung erat dengan logika, memungkinkan akses dengan latensi sangat rendah.

**Manajemen clock (PLL, MMCM, jaringan distribusi clock)**

Dalam desain FPGA—terutama saat mempercepat beban kerja AI—clock sangatlah penting. Kinerja, daya, dan keandalan sangat bergantung pada bagaimana Anda menghasilkan, memanipulasi, dan mendistribusikan clock. Sumber daya khusus berikut dalam FPGA modern memberi Anda kendali yang baik atas clocking.

(a) *Phase-Locked Loop* (PLL): PLL mengambil jam referensi eksternal atau internal, membandingkan fase/frekuensinya dengan sinyal umpan balik, dan menyesuaikan osilator terkontrol tegangan (VCO) untuk "mengunci" output sehingga fase/frekuensi selaras. Ini memungkinkan perkalian atau pembagian frekuensi, pergeseran fase, penyesuaian siklus kerja, dan terkadang penyaringan jitter. PLL digunakan untuk menghasilkan jam pada frekuensi yang berbeda yang dibutuhkan dari satu jam input.

(b) *Mixed-Mode Clock Manager* (MMCM): MMCM adalah blok manajemen jam yang lebih canggih yang ditemukan dalam arsitektur FPGA modern (Xilinx UltraScale). Mereka menawarkan granularitas yang lebih halus: perkalian/pembagian fraksional, pergeseran fase halus, konfigurasi ulang dinamis, kontrol siklus kerja, dan terkadang

fitur penyelarasan/deskewing jam yang lebih baik. Pada dasarnya mereka dibangun di atas apa yang dilakukan PLL tetapi memberikan lebih banyak fleksibilitas.

(c) Jaringan Distribusi Jam & Manajemen Skew: Setelah jam Anda dihasilkan (melalui PLL/MMCM), jam tersebut perlu dikirim ke semua FF, RAM, blok DSP, dll., dengan skew minimal (perbedaan waktu kedatangan) dan jitter yang dapat diterima. FPGA menyediakan perutean jam hierarkis: jaringan global (untuk fan-out tinggi, jam di seluruh chip), jaringan regional/lokal (untuk bagian-bagian fabrik), dan jalur jam I/O. Menggunakan buffer yang tepat (buffer jam global, buffer regional) dan sumber daya perutean khusus sangat penting. Untuk mempertahankan hubungan fase dan deskew, output dari MMCM/PLL sering menggunakan loop umpan balik yang mengukur penundaan melalui buffer (umpan balik melalui pin umpan balik jam, mencocokkan jenis buffer) sehingga alat dapat mengkompensasi (umpan balik internal vs eksternal).

Selain itu, banyak vendor FPGA membagi chip menjadi beberapa wilayah jam. Jam yang melintasi batas wilayah atau menggunakan perutean yang kurang khusus dapat mengalami skew yang lebih tinggi. Memilih tipe buffer (global vs regional) dan memahami topologi wilayah clock perangkat sangat penting.

**Arsitektur FPGA Heterogen**

Satu jenis unit pemrosesan seringkali tidak cukup. Sebagai respons, FPGA modern telah berevolusi dari sekadar papan kosong berisi logika yang dapat diprogram menjadi sistem heterogen yang kompleks yang menggabungkan berbagai jenis perangkat keras khusus pada satu chip. Pendekatan "*System on Chip*" ini memungkinkan perancang untuk menggunakan alat yang tepat untuk pekerjaan yang tepat, mencapai tingkat kinerja dan efisiensi yang sebelumnya tidak mungkin.

(a) Prosesor Tertanam (Inti ARM dalam FPGA SoC): Salah satu kemajuan paling signifikan adalah integrasi inti mikroprosesor yang terpasang langsung ke die FPGA. Perangkat ini sering disebut FPGA System on Chip (SoC). Biasanya, perangkat ini memiliki sistem prosesor yang kuat, seperti seri ARM Cortex-A multi-inti, yang terhubung ke fabrik logika yang dapat diprogram.

- Fleksibilitas Perangkat Lunak: Prosesor dapat menjalankan sistem operasi lengkap seperti Linux, menangani pengambilan keputusan yang kompleks, komunikasi jaringan, dan tugas manajemen sistem yang ditulis dalam bahasa yang familiar seperti C++ atau Python.
- Akselerasi Perangkat Keras: Logika yang dapat diprogram dapat digunakan untuk membuat akselerator perangkat keras khusus yang sangat paralel untuk bagian algoritma yang paling intensif secara komputasi, seperti perkalian matriks inti dalam model pembelajaran mendalam.

(b) Transceiver Kecepatan Tinggi: Aplikasi AI modern bergantung pada pemindahan sejumlah besar data dengan sangat cepat. Untuk memenuhi kebutuhan ini, FPGA dilengkapi dengan blok khusus yang disebut transceiver kecepatan tinggi. Ini adalah sirkuit khusus yang dirancang untuk mengirim dan menerima data secara serial dengan kecepatan luar biasa, seringkali mencapai puluhan atau bahkan ratusan gigabit per

detik (Gbps). Transceiver ini adalah fondasi perangkat keras untuk protokol komunikasi kecepatan tinggi umum seperti:

- PCI Express (PCIe): Untuk menghubungkan FPGA ke motherboard komputer host.
- Ethernet Gigabit 10/40/100: Untuk koneksi langsung ke jaringan berkecepatan tinggi.
- Memori Bandwidth Tinggi (HBM): Untuk akses ultra cepat ke tumpukan memori eksternal.

(c) Blok Akselerasi AI/ML: Menyadari meningkatnya kecerdasan buatan, produsen FPGA telah mulai mengintegrasikan blok pemrosesan yang sangat khusus yang dirancang untuk AI dan pembelajaran mesin. Ini melangkah lebih jauh dari kemampuan aritmatika slice DSP. Contohnya termasuk Xilinx/AMD AI Engine atau Intel FPGA AI Tensor Blocks. Ini bukan hanya pengali tunggal; ini adalah susunan prosesor canggih yang dirancang untuk mengeksekusi operasi vektor dan matriks yang mendominasi beban kerja AI secara efisien. Mereka dapat memproses seluruh lapisan jaringan saraf dengan efisiensi ekstrem dalam hal kinerja dan konsumsi daya. Kehadiran blok-blok ini menjadikan FPGA sebagai platform yang benar-benar berpusat pada AI, menggabungkan kemampuan adaptasi logika yang dapat diprogram dengan kekuatan mentah silikon AI khusus.

**Konfigurasi FPGA**

Terdapat tiga teknologi utama yang digunakan untuk menyimpan konfigurasi di dalam FPGA.

(a) Berbasis SRAM: Ini adalah teknologi yang paling umum digunakan dalam FPGA modern. Konfigurasi disimpan dalam sel Static RAM (SRAM) standar. Karena SRAM bersifat volatil, FPGA merupakan lembaran kosong setiap kali dihidupkan dan harus dikonfigurasi dari sumber eksternal, biasanya chip memori flash non-volatil. Keuntungan utama dari teknologi ini adalah kemampuan pemrograman ulang yang tak terbatas, yang ideal untuk pengembangan dan untuk sistem yang perlu beradaptasi di lapangan.

(b) Berbasis Flash: Pada FPGA ini, konfigurasi disimpan dalam memori flash on-chip. Karena flash bersifat non-volatil, perangkat mempertahankan konfigurasinya bahkan tanpa daya. Ini berarti FPGA "aktif" dan beroperasi saat dihidupkan, yang dapat menjadi keuntungan dalam sistem tertentu.

(c) Anti-fuse: Ini adalah teknologi yang dapat diprogram sekali (OTP). Selama konfigurasi, koneksi fisik permanen ("anti-fuse") dibuat. Setelah diprogram, logika perangkat tidak dapat diubah lagi. Meskipun kurang umum saat ini, teknologi ini dihargai di lingkungan dengan keamanan tinggi dan radiasi tinggi karena konfigurasinya tidak dapat diubah atau dirusak.

## 15.3 DESAIN BERBASIS HDL KONVENSIONAL

Konsep HDL bermula pada awal tahun 1980-an. VHDL diperkenalkan pada tahun 1981, awalnya didanai oleh Departemen Pertahanan AS sebagai cara untuk secara formal mendeskripsikan dan mendokumentasikan perangkat keras digital. Tak lama kemudian, Verilog

dirilis pada tahun 1984 oleh Gateway Design Automation dengan tujuan serupa. Awalnya, kedua bahasa tersebut terutama digunakan untuk simulasi dan verifikasi fungsional daripada implementasi perangkat keras langsung.

Pada tahun 1990-an dan awal 2000-an, kemajuan dalam alat sintesis memungkinkan HDL digunakan tidak hanya untuk simulasi tetapi juga untuk sintesis logika, memungkinkan perancang untuk beralih langsung dari deskripsi perilaku ke implementasi FPGA fisik. Pergeseran ini menandai revolusi dalam desain digital, menggantikan pengambilan skema manual dengan metodologi pengkodean yang jauh lebih terukur dan dapat digunakan kembali. Bahasa! Verilog, VHDL, System Verilog

**Karakteristik Desain**

Desain berbasis HDL menggambarkan perangkat keras pada tingkat transfer register (RTL), menentukan bagaimana sinyal berpindah antar register dan bagaimana fungsi logika diterapkan dalam setiap siklus clock. Tidak seperti bahasa pemrograman perangkat lunak, HDL secara inheren paralel, artinya beberapa proses dapat dieksekusi secara bersamaan. Hal ini membuatnya sangat cocok untuk mendesain sirkuit yang memanfaatkan konkurensi, seperti pipeline dan jalur data paralel. Pendekatan RTL memberi perancang kendali yang lebih detail atas pengaturan waktu, penggunaan sumber daya, dan perilaku, itulah sebabnya HDL tetap menjadi standar emas untuk aplikasi yang kritis terhadap kinerja.

(a) Kontrol eksplisit atas sumber daya perangkat keras: Salah satu keuntungan terbesar HDL adalah memberikan perancang kendali langsung atas sumber daya perangkat keras. Setiap register, multiplekser, memori, atau blok aritmatika dapat dijelaskan secara eksplisit dalam RTL, memastikan bahwa sirkuit yang dihasilkan menggunakan fabrik FPGA persis seperti yang dimaksudkan. Kontrol yang detail ini sangat berharga ketika mengoptimalkan area, daya, atau pengaturan waktu. Misalnya, seorang perancang dapat memutuskan apakah memori harus diimplementasikan dalam Block RAM (BRAM) atau didistribusikan di seluruh LUT, atau apakah operasi aritmatika harus menggunakan slice DSP atau logika umum.

(b) Deskripsi Logika Akurat Siklus: HDL memungkinkan deskripsi siklus demi siklus dari logika digital, yang berarti perilaku sinyal dapat didefinisikan secara tepat untuk setiap tepi clock. Tidak seperti abstraksi tingkat tinggi, desain RTL memungkinkan untuk mengetahui secara tepat kapan data ditangkap, disebarkan, atau dimodifikasi. Pemodelan akurat siklus ini sangat penting dalam sistem berkinerja tinggi seperti protokol komunikasi atau pemrosesan sinyal waktu nyata, di mana persyaratan waktu yang ketat harus dipenuhi. Dengan bekerja pada tingkat ini, perancang dapat membuat sirkuit deterministik dengan latensi dan throughput yang dapat diprediksi.

(c) Fleksibilitas Tinggi, Optimasi Tingkat Rendah Dimungkinkan: Karena HDL mengekspos tingkat terendah dari perilaku sirkuit, ia memberikan fleksibilitas yang tak tertandingi untuk optimasi. Para perancang dapat menyusun ulang alur pemrosesan, menyeimbangkan level logika secara manual, dan mengontrol penempatan register untuk mencapai frekuensi clock maksimum. Mereka juga dapat menyempurnakan kompromi antara latensi, throughput, dan luas area dengan memutuskan bagaimana

komputasi dijadwalkan di seluruh siklus. Kemampuan untuk mengoptimalkan pada tingkat yang sangat rendah ini membuat HDL sangat diperlukan untuk aplikasi di mana setiap nanodetik sangat penting, seperti akselerator berbasis FPGA untuk jaringan atau pembelajaran mendalam.

**Alur Desain**

$$Design\ Entry \rightarrow simulation \rightarrow synthesis \rightarrow place\&route \rightarrow bitstream\ generation$$

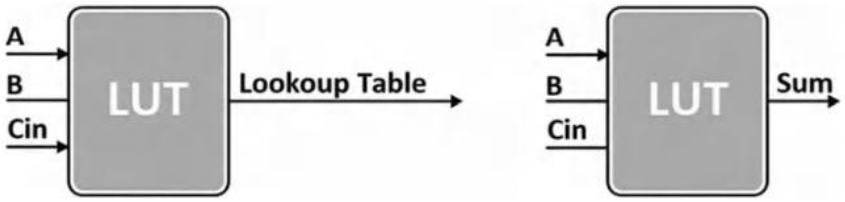
(a) Langkah pertama dalam mengimplementasikan desain digital disebut “entri desain”. Entri desain melibatkan penulisan desain digital yang diinginkan menggunakan HDL, seperti VHDL atau Verilog, dalam editor teks. Pilihan editor teks tidak terlalu penting; Anda dapat menggunakan editor yang paling nyaman bagi Anda. Fase ‘entri desain’ ini seringkali merupakan bagian yang paling memakan waktu dalam proses implementasi.

(b) Langkah kedua adalah “simulasi” desain. Setelah Anda mendeskripsikan desain Anda menggunakan HDL, Anda perlu memastikan desain tersebut berfungsi dengan benar, dan Anda dapat melakukan ini melalui simulasi. Dalam simulasi, Anda biasanya menerapkan input spesifik ke sirkuit Anda dan mengharapkan output tertentu. Dengan membandingkan output aktual dari sirkuit yang Anda rancang dengan output yang Anda harapkan, Anda dapat menentukan apakah Anda telah mengimplementasikan sirkuit dengan benar atau apakah mungkin ada kesalahan logika dalam desain.

(c) Langkah ketiga dalam mengimplementasikan sirkuit digital pada FPGA adalah salah satu fase yang paling vital. Ini disebut "sintesis desain." Sebelum kita membahas konsep sintesis desain untuk FPGA, mari kita bahas secara singkat gagasan umum sintesis dalam desain dan implementasi digital. Secara sederhana, sintesis dalam implementasi digital mengacu pada konversi dari deskripsi perangkat keras tingkat tinggi ke deskripsi tingkat rendah yang lebih detail. Tahap pertama sintesis melibatkan perpindahan dari merepresentasikan fungsi logika ke merepresentasikannya dengan gerbang logika. Tahap kedua sintesis membawa kita dari deskripsi tingkat gerbang ke deskripsi tingkat transistor.

Dalam FPGA, sintesis melibatkan implementasi sirkuit digital yang dijelaskan pada Gambar 15.4 menggunakan HDL dengan sumber daya perangkat keras yang tersedia di dalam FPGA itu sendiri. Sebuah adder penuh menggunakan bahasa seperti VHDL.

```
assign Sum = A^B^Cin;
assign Cout = (A&B) | (B&Cin) | (Cin&A);
```

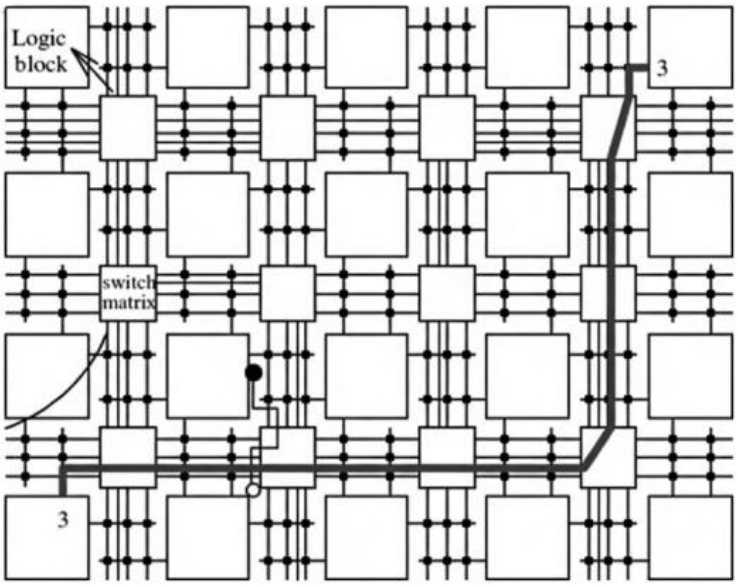



**Gambar 15.4** Implementasi adder penuh dengan LUT.

Pada bagian arsitektur, kita menggunakan dua pernyataan penugasan untuk menjelaskan cara kerja adder. Penugasan ini menggunakan gerbang logika seperti AND, OR, dan XOR untuk merepresentasikan adder penuh. Namun, ketika mengimplementasikan adder ini pada FPGA, kita perlu menggunakan sumber daya digital yang disediakan oleh FPGA itu sendiri, yaitu Lookup Table.

(d) Penempatan: Setelah alat sintesis menerjemahkan kode HDL abstrak kita menjadi kumpulan elemen logika konkret—seperti LUT, FF, dan slice DSP—kita dihadapkan pada tantangan baru. Kita memiliki semua komponen yang diperlukan untuk desain kita, tetapi komponen tersebut hanya ada sebagai daftar koneksi logis, sebuah netlist. Langkah penting untuk menugaskan setiap bagian logika ke lokasi fisik tertentu pada FPGA disebut penempatan. Ini bukan proses acak; ini adalah teka-teki optimasi yang sangat kompleks yang dikelola oleh alat CAD.

(e) Perutean: FPGA dilengkapi dengan sejumlah besar interkoneksi. Untuk melengkapi rangkaian yang diinginkan, diperlukan penyambungan kabel yang tepat, menghubungkannya ke LUT atau komponen lain pada FPGA. Pada Gambar 15.5, terlihat koneksi yang telah dibuat antara dua LUT yang kami gunakan untuk implementasi adder menggunakan kabel yang sudah ada di dalam FPGA.




**Gambar 15.5** Perutean sinyal melalui sumber daya FPGA.

(f) Pembuatan bitstream adalah tahap akhir dalam alur desain FPGA. Tahap ini mengambil desain yang telah sepenuhnya ditempatkan dan dirutekan, lalu mengubahnya menjadi file biner, yang dikenal sebagai bitstream (dengan ekstensi file seperti .bit atau .rbf). File ini adalah "cetak biru" akhir yang dapat dimuat, yang berisi semua data konfigurasi yang dibutuhkan untuk memprogram sel SRAM internal FPGA, memberi tahu setiap LUT, flip-flop, dan switch persis bagaimana harus berperilaku.

**Keunggulan**

Kekuatan utama desain berbasis HDL adalah kendali yang diberikannya kepada perancang. Karena setiap register, transisi keadaan, dan jalur data dapat dijelaskan secara eksplisit, HDL memungkinkan implementasi yang sangat optimal yang memanfaatkan sumber daya FPGA secara efisien. Hal ini sangat penting dalam domain yang kritis terhadap kinerja seperti jaringan, pemrosesan sinyal berkecepatan tinggi, dan akselerator pembelajaran mendalam khusus. Selain itu, ekosistem alat simulasi, verifikasi, dan sintesis yang matang menjadikan HDL sebagai metodologi yang kuat dan terbukti di industri.

- Optimasi tingkat detail (luas, kecepatan, daya)
- Transparansi dalam pemetaan perangkat keras

**Keterbatasan**

Meskipun memiliki kekuatan yang besar, desain berbasis HDL bukannya tanpa tantangan. Menulis dan men-debug kode RTL adalah proses yang memakan waktu, mengharuskan perancang untuk berpikir dalam hal siklus, sinyal, dan hubungan waktu. Kurva pembelajarannya curam, terutama bagi pengembang dengan latar belakang perangkat lunak. Desain yang besar juga dapat menjadi sulit dikelola, membutuhkan upaya signifikan dalam modularisasi, verifikasi, dan penutupan waktu. Keterbatasan ini membuka jalan bagi pendekatan alternatif seperti HLS, yang mengabstraksikan banyak detail tingkat rendah ini untuk mempercepat produktivitas desain.

- Kurva pembelajaran yang curam
- Memakan waktu untuk algoritma kompleks seperti CNN atau LSTM.

## 15.4 SINTESIS TINGKAT TINGGI MODERN (HLS)

HLS muncul sebagai respons terhadap kompleksitas dan persyaratan waktu desain RTL. Alih-alih mendeskripsikan siklus perangkat keras per siklus, HLS memungkinkan perancang untuk menulis fungsionalitas dalam bahasa tingkat tinggi seperti C, C++, OpenCL, atau System C, yang kemudian secara otomatis dikompilasi menjadi RTL oleh alat sintesis. Hal ini membuat proses lebih familiar bagi pengembang perangkat lunak dan secara signifikan mengurangi waktu yang dibutuhkan untuk membuat prototipe sistem berbasis FPGA.

- Bahasa/Framework: C, C++, OpenCL, Python (melalui framework), Vitis HLS, Intel HLS Compiler.

**Karakteristik Desain**

Tidak seperti HDL, di mana perancang menentukan perangkat keras pada RTL, HLS memungkinkan sistem untuk dijelaskan pada tingkat algoritmik. Fokusnya adalah pada apa

yang seharusnya dilakukan oleh desain, bukan pada bagaimana implementasinya dalam hal FF, status, atau siklus. Misalnya, alih-alih menulis pipeline terperinci untuk konvolusi, perancang cukup mengkodekan operasi matematika dalam C atau C++, dan alat sintesis menentukan cara memetakannya ke FPGA. Abstraksi ini membuat pengembangan jauh lebih cepat dan lebih intuitif, terutama bagi para insinyur yang berorientasi pada perangkat lunak.

Dalam HLS, pekerjaan berat mengubah kode tingkat tinggi menjadi perangkat keras ditangani oleh kompiler HLS. Kompiler menganalisis loop, ketergantungan data, dan pola akses memori, lalu menghasilkan deskripsi RTL yang setara dalam Verilog atau VHDL. Dari sana, alur sintesis FPGA standar mengambil alih. Meskipun perancang dapat memandu proses menggunakan arahan optimasi seperti pipeline atau unroll, sebagian besar pembuatan sirkuit terperinci diotomatiskan. Otomatisasi ini mengurangi waktu desain tetapi juga berarti bahwa beberapa kontrol tingkat rendah hilang dibandingkan dengan penulisan RTL secara langsung.

**Alur Desain**

$$\boldsymbol{Algoritma\ dalam\ C/C++ \rightarrow Sintesis\ HLS \rightarrow Netlist\ RTL \rightarrow Alur\ Toolchain\ FPGA}$$

(a) Desain Algoritma dalam C/C++: Proses dimulai dengan pembuatan model algoritma dalam bahasa pemrograman tingkat tinggi seperti C, C++, atau SystemC. Pada tahap ini, fokus sepenuhnya pada kebenaran fungsional, bukan detail spesifik perangkat keras. Perancang mengembangkan algoritma persis seperti yang akan ditulis untuk eksekusi perangkat lunak, menggunakan konstruksi umum seperti perulangan, array, dan fungsi.

```
// A simple C++ function for HLS
void matrix_multiply(int A[N] [N], int B[N] [N], int C[N] [N]) {
        // This loop structure is what HLS will convert to
hardware
        for (int i = 0; i < N; i++) {
                for (int j = 0; j < N; j++) {
                        int sum = 0;
                        for (int k = 0; k < N; k++) {
                                sum += A[i] [k] * B[k] [j];
                        }
                        C[i] [j] = sum;
                }
        }
}
```

(b) Sintesis Tingkat Tinggi (HLS): Ini adalah inti dari proses HLS. Kode C/C++ dimasukkan ke dalam alat sintesis HLS khusus. Alat ini bertindak sebagai kompiler canggih yang, alih-alih menghasilkan kode mesin untuk CPU, menghasilkan deskripsi perangkat keras. Alat ini melakukan beberapa tugas penting:

- Analisis Perilaku: Menginterpretasikan struktur algoritma dan ketergantungan kontrol/data.
- Penjadwalan dan Alokasi: Menentukan siklus clock dan sumber daya perangkat keras yang optimal untuk setiap operasi.
- Optimasi: Di sinilah perancang memandu alat. Dengan menggunakan instruksi khusus yang disebut pragma atau arahan langsung dalam kode C/C++, perancang menginstruksikan alat HLS tentang cara mengoptimalkan perangkat keras.

(c) Pembuatan dan Verifikasi Netlist RTL: Setelah sintesis dan optimasi HLS selesai, output akhir alat adalah netlist RTL. Ini adalah deskripsi perangkat keras dari algoritma, yang ditulis dalam HDL standar seperti VHDL atau Verilog. Kode RTL ini adalah deskripsi sirkuit digital yang sepenuhnya sesuai dan dapat disintesis yang berperilaku persis seperti algoritma C/C++ asli.

(d) Alur Rantai Alat FPGA Standar: Mulai dari titik ini, proses desain menyatu dengan mulus dengan alur FPGA tradisional yang telah kita bahas.

- Sintesis: Memetakan logika RTL ke dalam primitif FPGA.
- Penempatan dan Perutean (P&R): Mengatur blok logika secara fisik dan merute interkoneksi.
- Generasi Bitstream: Membuat file konfigurasi yang memprogram perangkat FPGA.

**Keuntungan**

(a) Pembuatan Prototipe Lebih Cepat, Produktivitas Lebih Tinggi: Daya tarik utama HLS terletak pada kesederhanaan dan abstraksinya. Alih-alih berfokus pada register individual, FF, atau detail waktu, seorang perancang dapat mendeskripsikan algoritma itu sendiri—misalnya, perkalian matriks atau kernel konvolusi dan membiarkan alat tersebut memutuskan cara terbaik untuk memetakannya ke FPGA. Hal ini tidak hanya mempercepat proses desain tetapi juga membuat pengembangan FPGA lebih mudah diakses oleh para insinyur yang mungkin tidak memiliki latar belakang yang kuat dalam desain perangkat keras digital.

(b) Integrasi yang Lebih Mudah dengan Kerangka Kerja Perangkat Lunak: Alat HLS juga menyediakan arahan (pragmas) yang memandu kompiler dalam mengoptimalkan perangkat keras yang dihasilkan. Misalnya, sebuah loop dapat dipipelin, diurai, atau diparalelkan hanya dengan menambahkan arahan, alih-alih menyusun ulang deskripsi perangkat keras secara manual. Hal ini memungkinkan eksplorasi cepat dari berbagai kompromi desain tanpa menulis ulang sebagian besar kode. Dalam praktiknya, perusahaan melaporkan pengurangan waktu pemasaran yang signifikan ketika menggunakan HLS dibandingkan dengan RTL.

**Keterbatasan**

Meskipun memiliki manfaat, HLS masih belum diadopsi secara luas seperti HDL tradisional untuk desain komersial yang kritis. Alasan utamanya adalah HLS tidak memberikan desainer tingkat kendali yang sama atas sumber daya perangkat keras. Meskipun dapat

menghasilkan sirkuit yang efisien, ada batasan seberapa banyak perangkat keras yang mendasarinya dapat disesuaikan.

Tidak seperti HDL, di mana setiap kabel, register, dan transisi keadaan dapat dikelola secara eksplisit, HLS menyerahkan banyak keputusan ini kepada kompiler. Kompromi ini berarti bahwa meskipun HLS sangat baik untuk produktivitas dan pembuatan prototipe cepat, HLS tidak selalu menjadi pilihan utama untuk desain FPGA yang kritis terhadap kinerja atau sangat disesuaikan.

## 15.5 DL VS. HLS: TINJAUAN KOMPARATIF

**Tabel 15.1** Studi perbandingan HDL dan HLS

| **Fitur** | **Hardware Description Languages (HDL)** | **High-Level Synthesis (HLS)** |
|---|---|---|
| Bahasa | Verilog, VHDL, SystemVerilog | C, C++, OpenCL, MATLAB |
| Tingkat abstraksi | Tingkat rendah. Mendeskripsikan perangkat keras pada tingkat **Register-Transfer Level (RTL)**. Anda secara eksplisit menentukan setiap siklus clock dan jalur data (data path). | Tingkat tinggi. Mendeskripsikan perilaku algoritmik. Alat (tool) secara otomatis menyimpulkan arsitektur perangkat keras. |
| Keahlian yang dibutuhkan desainer | Memerlukan pemahaman mendalam tentang logika digital, desain rangkaian, dan arsitektur perangkat keras. Umumnya digunakan oleh insinyur perangkat keras. | Memanfaatkan keterampilan pemrograman perangkat lunak. Mudah diakses oleh pengembang perangkat lunak yang memiliki pengetahuan dasar mengenai konsep perangkat keras. |
| Waktu pengembangan | Lebih lama. Desain manual memakan waktu dan rentan terhadap kesalahan. Memerlukan simulasi dan verifikasi yang ekstensif. | Lebih singkat. Pembuatan prototipe dan iterasi dapat dilakukan dengan cepat karena tingkat abstraksi yang lebih tinggi. Kode lebih ringkas. |
| Portabilitas kode | Rendah. Kode HDL sering kali bergantung pada pustaka vendor atau gaya desain tertentu, sehingga sulit digunakan kembali pada berbagai keluarga FPGA atau vendor yang berbeda. | Lebih tinggi. Kode algoritmik C/C++ sebagian besar bersifat portabel. Direktif HLS dapat disesuaikan untuk berbagai platform, tetapi logika inti tetap sama. |
| Performa dan efisiensi | Potensi tertinggi. Seorang desainer yang berpengalaman dapat membuat desain yang sangat dioptimalkan untuk memenuhi batasan waktu (timing), luas area, dan konsumsi daya tertentu. | Umumnya lebih rendah. Proses otomatis dari alat mungkin tidak mencapai tingkat optimasi yang sama seperti desainer berpengalaman. Pemanfaatan sumber daya terkadang dapat lebih tinggi. |
| Verifikasi dan debugging | Lebih kompleks dan lebih lambat. Simulasi RTL memerlukan waktu yang | Lebih cepat. Simulasi C/C++ berlangsung cepat. Anda dapat |

| | | |
|---|---|---|
| | cukup lama, dan proses debugging mengharuskan pemahaman terhadap bentuk gelombang (waveform) serta waktu sinyal (signal timing). | memverifikasi fungsionalitas algoritma di lingkungan perangkat lunak bahkan sebelum mensintesisnya menjadi perangkat keras. |

**Tabel 15.1** (Bersambung)

| **Fitur** | **Hardware Description Languages (HDL)** | **High-Level Synthesis (HLS)** |
|---|---|---|
| Penerapan untuk akselerator AI | Ideal untuk komponen yang sangat kritis terhadap performa. Digunakan untuk jalur data (data path) yang sangat dioptimalkan, antarmuka memori khusus, dan logika kontrol yang kompleks (misalnya mesin konvolusi atau pengendali memori), di mana setiap sumber daya harus dikendalikan secara presisi. | Lebih cepat. Simulasi C/C++ berlangsung cepat. Anda dapat memverifikasi fungsionalitas algoritma dalam lingkungan perangkat lunak bahkan sebelum mensintesisnya menjadi perangkat keras. |
| Penggunaan umum pada proyek | Digunakan untuk membuat **IP Core** khusus yang sangat dioptimalkan dan membutuhkan performa maksimum atau konsumsi daya yang rendah. | Digunakan untuk pembuatan prototipe secara cepat, mengeksplorasi berbagai kompromi arsitektur (*architectural trade-offs*), serta mengembangkan sebagian besar logika algoritma AI. Selain itu, dapat digunakan untuk menghasilkan desain akhir bagi komponen yang tidak terlalu kritis. |
| Contoh | Menulis sebuah array **Multiply-Accumulate (MAC)** yang dipipeline dari awal menggunakan Verilog untuk mencapai throughput dan latensi tertentu. | Menulis sebuah fungsi C++ untuk lapisan konvolusi (*convolution layer*) dan menggunakan pragma HLS untuk melakukan *pipeline* serta *loop unrolling* agar memenuhi target performa. |
| Perangkat utama | Vivado, Quartus Prime, Intel Quartus Pro, dan lain-lain. | Vitis HLS, Intel FPGA HLS, Xilinx Vivado HLS, dan lain-lain. |

## 15.6 APLIKASI REAL-TIME DARI DEEP LEARNING

Akselerator berbasis FPGA menjadi semakin penting untuk inferensi dan pelatihan deep learning, menawarkan keseimbangan antara efisiensi perangkat keras khusus dan kemampuan pemrograman. HDL yang digunakan untuk desain akselerator CNN yang hemat

memori: HDL untuk Akselerator Berkinerja Tinggi dan Hemat Memori: Ketika tujuannya adalah untuk menciptakan akselerator canggih dengan kinerja terbaik dan konsumsi daya terendah, perancang menggunakan HDL tradisional (Verilog/VHDL). Pendekatan ini menawarkan kontrol yang sangat detail atas setiap kabel dan register, memungkinkan mereka untuk merancang skema akses memori yang sangat canggih dan jalur data khusus yang memaksimalkan setiap tetes kinerja dari silikon.

Inilah jalan untuk menciptakan akselerator AI kelas terbaik dan siap produksi. HLS untuk Penyebaran Cepat Akselerator Inferensi - Dalam banyak situasi, tujuan utamanya bukanlah kinerja terbaik, tetapi waktu tercepat untuk memasarkan produk. Di sinilah HLS unggul. Seorang ahli algoritma AI, yang mungkin bukan ahli desain perangkat keras, dapat mengambil model yang ditulis dalam C/C++ dan menggunakan alat HLS untuk secara otomatis menghasilkan akselerator perangkat keras berkualitas baik dalam waktu yang jauh lebih singkat. Ini sangat cocok untuk pembuatan prototipe cepat dan penerapan akselerator untuk aplikasi di mana mendapatkan solusi yang cepat dan efisien untuk diterapkan dengan cepat adalah prioritas utama.

## 15.7 KESIMPULAN

FPGA telah terbukti menjadi platform yang efektif untuk mempercepat inferensi pembelajaran mendalam, khususnya dalam situasi di mana latensi rendah dan penghematan energi diperlukan. Arsitektur yang dapat dikonfigurasi ulang memungkinkan pembuatan solusi perangkat keras yang disesuaikan dan fleksibel yang dapat beradaptasi dengan beban kerja AI yang berkembang dan memaksimalkan komputasi jaringan saraf. Dibandingkan dengan CPU dan GPU konvensional, FPGA memberikan keuntungan besar dalam hal fleksibilitas, paralelisme, dan penggunaan sumber daya; namun demikian, ada juga kekurangan, seperti keterbatasan bandwidth memori dan kompleksitas pengembangan yang lebih tinggi untuk sistem yang sangat disesuaikan. Namun, kemajuan terbaru dalam perangkat keras FPGA dan teknik sintesis, seperti HLS, membuka pintu bagi penerapan AI yang lebih baik di seluruh aplikasi edge dan embedded dengan membuat perangkat ini mudah tersedia dan sesuai untuk tugas-tugas deep learning di dunia nyata.

# DAFTAR PUSTAKA


Attaoui, Y., Chentouf, M., Ismaili, Z. E. A. A., & Elmourabit, A. (2024). Machine Learning In VLSI Design: A Comprehensive Review. *Journal Of Integrated Circuits And Systems*, *19*(2), 1-14.

Cao, Y., Chen, Y., Fan, X., Fu, H., & Xu, B. (2026). Advanced Design For High-Performance And AI Chips. *Nano-Micro Letters*, *18*(1), 13.

Chêne, M. (2025). *Distributed Artificial Intelligence Integrated Circuits For Ultra-Low-Power Smart Sensors* (Doctoral Dissertation, Centrale Lille Institut).

Deng, C., Ho, C. T., & Ren, H. (2026). Artificial Intelligence-Assisted IC Design: Large Language Models And Agentic Systems For Modern Chip Design. *IEEE Solid-State Circuits Magazine*, *18*(2), 41-51.

Dev, A., Fouda, M. M., & Chiu, S. (2025, May). The Role Of Artificial Intelligence In Transforming VLSI Design, Testing, And Manufacturing. In *2025 Intermountain Engineering, Technology And Computing (IETC)* (Pp. 1-6). IEEE.

Dhivya, V., Al-Qerem, A., Shifani, S. A., Geetha, R., & Venkatraman, A. (2026, April). Thermal Aware Intelligent Scheduling Framework For High Density Artificial Intelligence VLSI Chips. In *2026 3rd International Conference On Research Methodologies In Knowledge Management, Artificial Intelligence And Telecommunication Engineering (RMKMATE)* (Pp. 1-6). IEEE.

Dutta, S., Goswami, K., & Mondal, H. (2026). Design And Analysis Of An AI-Enhanced Low-Power Current-Mode 8-Bit DAC For VLSI Circuits. In *Chips And Intelligence* (Pp. 311-335). CRC Press.

Gorde, K. S., Sonavane, S. M., Hutke, S., & Hutke, A. (2025). Analog Artificial Intelligence Hardware For Neural Networks: Design Trends And Considerations. *Bulletin Of Electrical Engineering And Informatics*, *14*(6), 4399-4410.

Karpagam, P., Arulanandan, S., Pandi, V. S., Al-Safarini, M. Y., Satyavati, Y. A., & Dinesh, M. (2025, November). Constructing AI-Driven Microprocessors For Smart Devices: Integrating On-Chip Intelligence To Boost Cutting-Edge Decision-Making Performance. In *2025 IEEE 7th International Conference On Computing, Communication And Automation (ICCCA)* (Pp. 1-7). IEEE.

Khan, I. U., Mittal, N., & Ansari, M. A. (2023). Applications Of VLSI Design In Artificial Intelligence And Machine Learning. *Machine Learning For VLSI Chip Design*, 1-17.

Khetarpal, V., Gupta, L., Dhand, R., & Sharma, P. (2023, December). Machine Learning Techniques For VLSI Circuit Design: A Review. In *International Conference On Intelligent Systems Design And Applications* (Pp. 191-199). Cham: Springer Nature Switzerland.

Kim, Y., Patel, S., Kim, H., Yadav, N., & Choi, K. K. (2021). Ultra-Low Power And High-Throughput SRAM Design To Enhance AI Computing Ability In Autonomous Vehicles. *Electronics*, *10*(3), 256.

Kumar, R., Nigmatov, D., Mohammed, M. A., Sheet, S. A. W., Mahmoud, M. I., & Vijayalakshmi, B. A. (2025, August). Innovative VLSI System Design And Embedded Architectures Empowered By AI And Machine Learning Advancements. In *2025 12th International Conference On Emerging Trends In Engineering & Technology-Signal And Information Processing (ICETET-SIP)* (Pp. 1-6). IEEE.

Lee, K. J., Lee, J., Choi, S., & Yoo, H. J. (2020). The Development Of Silicon For AI: Different Design Approaches. *IEEE Transactions On Circuits And Systems I: Regular Papers*, *67*(12), 4719-4732.

Maheswari, S., Gokilam, G. G., Navaneethan, S., Amer, A., Sunheriya, N., & Parthi, P. S. (2025, March). AI-Powered VLSI Design Automation For Faster Chip Production In Emerging Technologies. In *2025 3rd International Conference On Device Intelligence, Computing And Communication Technologies (DICCT)* (Pp. 445-450). IEEE.

Mishra, N. K. (2026). AI And ML-Based Design Methodologies For Power-Constrained Integrated Circuits. In *Chips And Intelligence* (Pp. 73-104). CRC Press.

Misra, A., Saikia, R., & Sarma, M. P. (2027). AI-Driven Adaptive Power Management In VLSI Design. In *Chips And Intelligence* (Pp. 164-194). CRC Press.

Mohamedyaseen, A., Vasantharaj, A., & Kumar, E. S. (2023). VLSI Design Of Low-Power Edge AI Processors For IOT Devices. *ICTACT J. Microelectron*, *9*(3), 1634-1639.

Morsy, M. M. (2026). AI-On-Chip Systems: A Cross-Layer Review Of Architectures, Interconnects, Design Automation, And Embedded Intelligence. *Electronics*, *15*(12), 2645.

Nagar, P., Boruah, S., Bhoi, A. K., Patel, A., Sarda, J., & Darjij, P. (2024, January). Emerging VLSI Technologies For High Performance AI And ML Applications. In *2024 International Conference On Advancements In Smart, Secure And Intelligent Computing (ASSIC)* (Pp. 1-5). IEEE.

Raj, B., Tripathi, S. L., Chaudhary, T., Rao, K. S., & Singh, M. (Eds.). (2024). *Integrated Devices For Artificial Intelligence And VLSI: VLSI Design, Simulation And Applications*. John Wiley & Sons.

Rajput, H. (2027). Revolutionizing VLSI Design Through Artificial Intelligence For Low-Power Applications. In *Chips And Intelligence* (Pp. 22-52). CRC Press.

Sathish, K., Sathya, A., Pattunnarajam, P., Rao, A. S., & Lakshmisridevi, S. (2024, August). Innovative Approaches In Advanced VLSI Design For High-Performance Computing Applications. In *2024 Second International Conference On Intelligent Cyber Physical Systems And Internet Of Things (Icoici)* (Pp. 643-648). IEEE.

Shah, O. A., & Tiwari, D. (2024). From Chips To Systems: Exploring Disruptive VLSI Ecosystems. In *A Practitioner's Approach To Problem-Solving Using AI* (Pp. 268-281). Bentham Science Publishers.

Shah, P. P., Munjani, J., Vyas, D., & Sharma, A. (2025, August). Review Of Cutting-Edge Latest Trends And Innovations In Low-Power VLSI Physical Design Techniques For Next-Gen AI/ML Hardware. In *2025 5th Asian Conference On Innovation In Technology (ASIANCON)* (Pp. 1-8). IEEE.

Shreyanth, S., Harshitha, D. S., & Niveditha, S. (2023). Implementation Of Machine Learning In VLSI Integrated Circuit Design. *SN Computer Science*, *4*(2), 137.

Shrirao, N. M. (2024). Intelligent Design Automation In VLSI Systems: Leveraging AI For Future Electronics Applications. *Electronics, Communications, And Computing Summit*, *2*(3), 28-39.

Singh, A. P., Mishra, V. K., & Akhter, S. (2025). Application Of Artificial Intelligence/Machine Learning In VLSI Design. In *Advancing VLSI Through Machine Learning* (Pp. 184-197). CRC Press.

Sumalatha, A., Sabri, M. O., Kumar, S., Vijayanandh, T., Lalitha, J., & Shanmugam, N. (2025, October). Automated Very Large-Scale Integration (VLSI) Design Using Artificial Intelligence: Energyconstrained Internet Of Things (Iot) Edge Devices. In *2025 IEEE International Conference On Blockchain And Distributed Systems Security (ICBDS)* (Pp. 1-6). IEEE.

Tanjila, H. R., & Rana, M. S. (2026). A Comprehensive Review Of Recent Advances In VLSI Design And Technology: Architectures, Challenges, And Future Trend. *Scientia. Technology, Science And Society*, *3*(2), 52-62.

Thakur, G., & Jain, S. (2025). Role Of Artificial Intelligence In VLSI Design: A Review. *Recent Advances In Computer Science And Communications*, *18*(1), E250424229315.

Vaibhav, V. G. (2025). A Neuromorphic-Inspired, Low-Power VLSI Architecture For Edge AI In Iot Sensor Nodes. *Journal Of Microelectronics And Solid State Devices*, *12*(2), 41-47p.

Yang, Y. (2025, October). Research On Specialized Chips For Artificial Intelligence. In *Proceedings Of The 2025 2nd International Conference On Electrical Engineering And Intelligent Control (EEIC 2025)* (P. 21). Springer Nature.

Ye, L., Wang, Z., Jia, T., Ma, Y., Shen, L., Zhang, Y., ... & Huang, R. (2023). Research Progress On Low-Power Artificial Intelligence Of Things (Aiot) Chip Design. *Science China Information Sciences*, *66*(10), 200407.

Zhang, Q., Deng, H., & Song, K. (2024). Latest VLSI Techniques For 3nm Technology For Building Efficient AI Chips. *Fusion Of Multidisciplinary Research, An International Journal*, *5*(2), 654-670.

# Desain VLSI Hemat Daya dengan AI

**Dr. Ir. Agus Wibowo, M.Kom, M.Si, MM.**

## BIO DATA PENULIS



Penulis memiliki berbagai disiplin ilmu yang diperoleh dari Universitas Diponegoro (UNDIP) Semarang. dan dari Universitas Kristen Satya Wacana (UKSW) Salatiga. Disiplin ilmu itu antara lain teknik elektro, komputer, manajemen, ilmu sosiologi dan ilmu hukum. Penulis memiliki pengalaman kerja pada industri elektronik dan sertifikasi keahlian dalam bidang Jaringan Internet, Telekomunikasi, Artificial Intelligence, Internet Of Things (IoT), Augmented Reality (AR), Technopreneurship, Internet Marketing dan bidang pengolahan dan analisa data (komputer statistik), Ilmu Perpajakan.

Penulis adalah pendiri dari Universitas Sains dan Teknologi Komputer (Universitas STEKOM ) dan juga seorang dosen yang memiliki Jabatan Fungsional Akademik Lektor Kepala (Associate Professor) yang telah menghasilkan puluhan Buku Ajar ber ISBN, HAKI dari beberapa karya cipta dan Hak Paten pada produk IPTEK. Sejak tahun 2023 penulis tercatat sebagai Dosen luar biasa di Fakultas Ekonomi & Bisnis (FEB) Universitas Diponegoro Semarang. Penulis juga terlibat dalam berbagai organisasi profesi dan industri yang terkait dengan dunia usaha dan industri, khususnya dalam pengembangan sumber daya manusia yang unggul untuk memenuhi kebutuhan dunia kerja secara nyata.

YAYASAN PRIMA AGUS TEKNIK

**PENERBIT :**
YAYASAN PRIMA AGUS TEKNIK
Jl. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

ISBN 978-634-7695-45-1 (PDF)
9 786347 695451