

BUKU DARAS

ORGANISASI & ARSITEKTUR KOMPUTER



Disusun oleh :

Faisal, S.T, M.T

JURUSAN TEKNIK INFORMATIKA
UIN ALAUDDIN MAKASSAR
2015

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah swt, atas segala rahmat dan hidayahNya sehingga penulis mampu menyelesaikan buku yang sederhana ini dengan segala kekurangannya. Buku ini disusun dalam rangka mengimplementasikan integrasi keilmuan antara disiplin ilmu komputasi yang dikaitkan dengan kajian-kajian ayat Al-Qur'an yang berhubungan dengannya, sehingga kita sebagai manusia semakin menyadari bahwa ilmu yang kita miliki hanyalah setetes air ditengah lautan ilmu yang dimiliki Allah swt.

Buku ini merupakan bahan kuliah Organisasi dan Arsitektur Komputer selama satu semester dengan bobot teori 2 SKS ditambah dengan praktikum 1 SKS, dan merupakan mata kuliah wajib di jurusan Teknik Informatika UIN Alauddin Makassar. Buku ini berisi 11 satuan bahasan yang masing-masing membahas tentang : Pengantar Organisasi dan Arsitektur Komputer; Evolusi dan Kinerja Komputer; Sistem Bilangan; Gerbang Logika Dasar dan Aljabar Boole; Rangkaian Logika Kombinasi; Rangkaian Logika Sekuensial; Central Processing Unit (CPU); Memori Utama; Memori Sekunder; Peralatan Input/Output; dan Sistem Bus.

Dengan membaca buku ini, diharapkan mahasiswa dan para pembaca dapat memahami konsep-konsep dari organisasi dan arsitektur komputer itu sendiri, dan dalam prakteknya mahasiswa mampu mengenali berbagai jenis komponen dan fungsinya sebagai pembentuk dari perangkat komputer. Betapa besar kekuasaan Allah swt telah menciptakan manusia sebagai makhluk yang paling mulia dan dengan ilmu yang dimilikinya telah menciptakan perangkat komputer yang sangat membantu aktifitas manusia sehari-hari.

Pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada semua pihak baik langsung maupun tidak langsung, yang telah banyak memberikan bantuan, saran dan kritikan, sehingga penulisan buku ini dapat terwujud. Ucapan terima kasih penulis sampaikan kepada Rektor UIN Alauddin Makassar, Kepala LP2M, Dekan Fakultas Sains & Teknologi, rekan-rekan dosen, dan segenap pihak yang tidak sempat disebutkan satu per satu, serta terkhusus kepada keluarga tercinta atas segala dukungan dan bantuannya mulai dari tahap awal hingga rampungnya buku ini. Semoga Allah swt menerima kebaikan semuanya dan menjadikan sebagai catatan amaliah disisiNya.

Pada akhirnya penulis menyadari akan keterbatasan ilmu dan pengetahuan yang dimiliki sehingga apa yang tersaji dalam buku ini masih jauh dari kesempurnaan. Tak ada gading yang tak retak. Oleh karena itu, segala kritik, saran dan masukan yang membangun, sangat penulis harapkan demi perbaikan buku ini untuk penerbitan selanjutnya.

Semoga buku ini ada manfaatnya untuk kepentingan akademik, bangsa dan agama, serta mampu menjadi pelita dunia ilmiah di tanah air. Aamiinn ya Rabbal aalamiin.

Makassar, Agustus 2015

PENULIS

DAFTAR ISI

| | |
|--|------------|
| Sambutan Rektor UIN Alauddin Makassar | iii |
| Kata Pengantar | iv |
| Daftar Isi | vi |
| Daftar Gambar..... | x |
| Daftar Tabel..... | xii |

Satuan Bahasan 1 :

| | |
|--|----------|
| Pengantar Organisasi dan Arsitektur Komputer..... | 1 |
| 1.1. Komputer | 2 |
| 1.2. Pengertian Organisasi dan Arsitektur Komputer..... | 4 |
| 1.3. Struktur dan Fungsi Utama Komputer..... | 4 |
| 1.3.1. Struktur Komputer | 4 |
| 1.3.2. Fungsi Komputer | 5 |
| Rangkuman..... | 7 |
| Tes Formatif | 8 |

Satuan Bahasan 2 :

| | |
|---|----------|
| Evolusi dan Kinerja Komputer..... | 9 |
| 2.1. Sejarah Perkembangan Komputer | 10 |
| 2.1.1. Komputer Primitif | 10 |
| 2.1.2. Generasi Pertama: Tabung Vakum (1945-1955)..... | 11 |
| 2.1.3. Generasi Kedua: Transistor(1955-1965)..... | 15 |
| 2.1.4. Generasi Ketiga: Integrated Circuit(1965-1980) | 17 |
| 2.1.5. Generasi Keempat: Very Large Scale Integrated (1980-sekarang) | 18 |
| 2.2. Perancangan Kinerja..... | 19 |
| 2.3. Contoh Evolusi Komputer | 21 |
| Rangkuman..... | 23 |
| Tes Formatif | 24 |

Satuan Bahasan 3 :

| | |
|--|-----------|
| Sistem Bilangan..... | 25 |
| 3.1. Jenis Sistem Bilangan | 26 |
| 3.2. Sistem Bilangan Desimal | 26 |
| 3.3. Sistem Bilangan Binari (Biner)..... | 28 |
| 3.4. Sistem Bilangan Oktal..... | 28 |
| 3.5. Sistem Bilangan Hexadesimal..... | 29 |
| 3.6. Konversi Sistem Bilangan..... | 30 |
| 3.6.1. Konversi Sistem Bilangan Desimal..... | 30 |
| 3.6.2. Konversi Sistem Bilangan Binari | 32 |

| | |
|---|----|
| 3.6.3. Konversi Sistem Bilangan Oktal | 34 |
| 3.6.4. Konversi Sistem Bilangan Hexadesimal..... | 35 |
| 3.7. Sistem Kode | 36 |
| 3.7.1. Kode BCD (<i>Binary-Coded Decimal</i>) | 36 |
| 3.7.2. Kode Excess-3 | 37 |
| 3.7.3. Kode Gray | 38 |
| 3.7.4. Kode 7-Segment Display..... | 38 |
| 3.7.5. Kode ASCII | 39 |
| Rangkuman..... | 40 |
| Tes Formatif | 40 |

Satuan Bahasan 4 :

| | |
|---|-----------|
| Gerbang Logika Dasar & Aljabar Boole | 41 |
| 4.1. Tabel Kebenaran | 42 |
| 4.2. Gerbang Logika Dasar | 44 |
| 4.2.1. AND Gate | 45 |
| 4.2.2. OR Gate | 45 |
| 4.2.3. NOT Gate | 46 |
| 4.2.4. NAND Gate | 47 |
| 4.2.5. NOR Gate | 47 |
| 4.2.6. EX-OR Gate | 48 |
| 4.2.7. EX-NOR Gate | 48 |
| 4.3. Teorema-teorema Aljabar Boole | 48 |
| 4.3.1. Teorema Variabel Tunggal | 49 |
| 4.3.2. Teorema Variabel Jamak | 49 |
| Rangkuman..... | 50 |
| Tes Formatif | 51 |

Satuan Bahasan 5 :

| | |
|--|-----------|
| Rangkaian Logika Kombinasi | 52 |
| 5.1. Pengertian Logika Kombinasi | 53 |
| 5.2. Bentuk-bentuk Persamaan Logika | 53 |
| 5.2.1. Bentuk <i>Sum Of Product (SOP)</i> | 54 |
| 5.2.2. Bentuk <i>Product Of Sum (POS)</i> | 57 |
| 5.2.3. Mengubah Fungsi Bentuk Tak Standar Menjadi Bentuk Standar | 60 |
| 5.3. Penyederhanaan Secara Aljabar | 60 |
| Rangkuman..... | 73 |
| Tes Formatif. | 73 |

Satuan Bahasan 6 :

| | |
|--|-----------|
| Rangkaian Logika Sekuensial | 74 |
|--|-----------|

| | |
|--|----|
| 6.1. Flip Flop | 75 |
| 6.1.1. Rangkaian Flip Flop Dasar | 75 |
| 6.2. Encoder dan Decoder | 85 |
| 6.3. Multiplexer dan Demultiplexer | 94 |
| Rangkuman..... | 98 |
| Tes Formatif..... | 99 |

Satuan Bahasan 7 :

| | |
|--------------------------------------|------------|
| Central Processing Unit | 100 |
| 7.1. Komponen Utama CPU..... | 101 |
| 7.2. Fungsi CPU | 103 |
| 7.2.1. Siklus Fetch-Eksekusi | 103 |
| 7.2.2. Fungsi Interrupt | 105 |
| Rangkuman..... | 108 |
| Tes Formatif | 108 |

Satuan Bahasan 8 :

| | |
|--|------------|
| Memori Utama | 109 |
| 8.1. Operasi Sel Memori..... | 110 |
| 8.2. Karakteristik Sistem Memori | 111 |
| 8.3. Keandalan Memori | 113 |
| 8.4. Satuan Memori | 115 |
| 8.5. Memori Utama Semikonduktor | 115 |
| 8.5.1. Jenis Memori Random Akses | 115 |
| 8.5.2. Pengemasan (<i>Packaging</i>) | 117 |
| 8.5.3. Koreksi Error | 118 |
| 8.6. Cache Memori | 122 |
| 8.7. Elemen Rancangan | 124 |
| 8.7.1. Kapasitas Cache | 124 |
| 8.7.2. Ukuran Blok | 125 |
| 8.7.3. Fungsi Pemetaan (Mapping) | 125 |
| 8.7.4. Algoritma Penggantian | 127 |
| 8.7.5. <i>Write Policy</i> | 128 |
| 8.7.6. Jumlah Cache | 129 |
| Rangkuman..... | 129 |
| Tes Formatif | 130 |

Satuan Bahasan 9 :

| | |
|------------------------------|------------|
| Memori Sekunder | 131 |
|------------------------------|------------|

| | |
|--------------------------|-----|
| 9.1. Magnetik Disk | 131 |
| 9.2. RAID | 138 |
| 9.3. Optical Disk | 141 |
| 9.4. Pita Magnetik | 146 |
| Rangkuman..... | 147 |
| Tes formatif | 148 |

Satuan Bahasan 10 :

| | |
|--|------------|
| Peralatan Input/Output | 149 |
| 10.1. Sistem Masukan dan Keluaran Komputer | 150 |
| 10.1.1. Fungsi Modul I/O | 151 |
| 10.1.2. Struktur Modul I/O | 153 |
| 10.2. Teknik Masukan/Keluaran | 154 |
| 10.2.1. I/O Terprogram | 154 |
| 10.2.2. Interrpt-Driven I/O | 156 |
| 10.2.3. <i>Direct Memory Access</i> (DMA) | 160 |
| 10.3. Perangkat Eksternal | 162 |
| Rangkuman | 163 |
| Tes Formatif | 163 |

Satuan Bahasan 11 :

| | |
|------------------------------------|------------|
| Sistem BUS | 164 |
| 11.1. Struktur Interkoneksi | 165 |
| 11.2. Interkoneksi Bus | 167 |
| 11.3. Elemen Perancangan Bus | 170 |
| 11.4. Contoh Bus | 173 |
| 11.4.1. Bus ISA | 173 |
| 11.4.2. Bus PCI | 174 |
| 11.4.3. Bus USB..... | 174 |
| 11.4.4. Bus SCSI..... | 175 |
| 11.4.5. Bus P1394/FireWire..... | 176 |
| Rangkuman..... | 176 |
| Tes Formatif..... | 177 |

| | |
|-----------------------------|------------|
| Daftar Pustaka | 178 |
|-----------------------------|------------|

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 1.1. Unit Fungsional Dasar pada Komputer..... | 3 |
| Gambar 1.2. Struktur Dasar Komputer..... | 5 |
| Gambar 1.3. Fungsi Dasar Komputer..... | 6 |
| Gambar 1.4. Operasi-operasi Komputer..... | 7 |
| Gambar 2.1. John von Neumann..... | 12 |
| Gambar 2.2. Struktur Komputer IAS..... | 13 |
| Gambar 2.3. Format Memori IAS..... | 13 |
| Gambar 2.4. Struktur Detail Komputer IAS..... | 14 |
| Gambar 2.5. Konfigurasi IBM 7094..... | 16 |
| Gambar 2.6. Struktur Bus PDP-8..... | 18 |
| Gambar 2.7. Grafik Jumlah Transistor dalam Chip Pentium.... | 19 |
| Gambar 2.8. Grafik Perbandingan Kecepatan Mikroprosesor dan Memori | 20 |
| Gambar 3.1. Bobot Nibble | 29 |
| Gambar 3.2. Peraga 7-Segmen | 39 |
| Gambar 4.1. Diagram Blok Rangkaian Logika Satu Variabel Input | 43 |
| Gambar 4.2. Diagram Blok Rangkaian Logika Dua Variabel Input | 43 |
| Gambar 4.3. Diagram Blok Rangkaian Logika Tiga Variabel Input | 44 |
| Gambar 4.4. Simbol Gerbang AND | 45 |
| Gambar 4.5. Simbol Gerbang OR | 46 |
| Gambar 4.6. Simbol Gerbang NOT | 46 |
| Gambar 4.7. Simbol Gerbang NAND | 47 |
| Gambar 4.8. Simbol Gerbang NOR | 47 |
| Gambar 4.9. Simbol Gerbang EX-OR..... | 48 |
| Gambar 4.10. Simbol Gerbang EX-NOR | 48 |
| Gambar 5.1. Diagram Blok Rangkaian Logika Kombinasi..... | 53 |
| Gambar 5.2. Rangkaian Logika Sebelum dan Sesudah Diminimisasi | 61 |
| Gambar 5.3. Diagram Venn | 64 |
| Gambar 5.4. Diagram Venn untuk irisan dua variabel | 64 |
| Gambar 5.5. Diagram Venn untuk irisan tiga variabel..... | 64 |
| Gambar 5.6. Karnaugh Map untuk 2 variabel input | 65 |
| Gambar 5.7. Karnaugh Map untuk 3 variabel input | 65 |
| Gambar 5.8. Karnaugh Map untuk 4 variabel input | 66 |
| Gambar 5.9. Karnaugh Map untuk 6 variabel input | 66 |
| Gambar 5.10. Contoh input data dari <i>truth table</i> ke K-Map..... | 66 |
| Gambar 5.11. Input Nilai Logic 1 dalam K-Map..... | 67 |
| Gambar 5.12. K-Mapmembentuk persamaan logika sederhana.. | 67 |

| | |
|---|-----|
| Gambar 6.1. Rangkaian Flip Flop Dasar dengan Gerbang NOR | 75 |
| Gambar 6.2. Rangkaian Flip Flop Dasar dengan Gerbang NAND | 77 |
| Gambar 6.3. Daigram Logika Flip Flop RS | 79 |
| Gambar 6.4. Lambang Tanpa dan Dengan Pulsa Waktu | 79 |
| | 81 |
| Gambar 6.5. Persamaan Karakteristik | 82 |
| Gambar 6.6. Diagram Logika Flip Flop D..... | 83 |
| Gambar 6.7. Lambang dan Persamaan Karakteristik | 83 |
| Gambar 6.8. Diagram Logika Flip Flop JK..... | 84 |
| Gambar 6.9. Lambang dan Persamaan Karakteristik | 85 |
| Gambar 6.10 Diagram Logika Flip Flop T | 85 |
| Gambar 6.11. Lambang dan Persamaan Karakteristik | 86 |
| Gambar 6.12. Encoder dan Decoder | 87 |
| Gambar 6.13. Rangkaian Encoder Oktal ke Biner | 88 |
| Gambar 6.14. Rangkaian Decoder Biner ke Oktal..... | 89 |
| Gambar 6.15. Seven Segment | 89 |
| Gambar 6.16. Sirkuit untuk menyalakan LED | |
| Gambar 6.17. Azas Menyalakan LED | 90 |
| Gambar 6.18. Angka-angka yang dapat ditampilkan oleh 7-Segment..... | 92 |
| Gambar 6.19. Decoder BCD ke Desimal | 93 |
| Gambar 6.20. Rangkaian Decoder BCD to 7-Segmen common anoda | 95 |
| Gambar 6.21. Diagram Multiplexer | 102 |
| Gambar 6.22. Diagram Multiplexer 8 kanal 1 bit | 102 |
| Gambar 7.1. Komponen Internal CPU..... | 103 |
| Gambar 7.2. Struktur Detail Internal CPU | 105 |
| Gambar 7.3. Siklus Instruksi Dasar..... | 106 |
| Gambar 7.4. Diagram Siklus Insstruksi | 107 |
| Gambar 7.5. Siklus eksekusi Instruksi dengan Interupsi | 110 |
| Gambar 7.6. Transfer Pengendalian pada interupsi Ganda..... | 114 |
| Gambar 8.1. Operasi Sel Memori | 118 |
| Gambar 8.2. Hirarki Memori | 118 |
| Gambar 8.3. Pin dan Sinyal Kemasan Memori | 119 |
| Gambar 8.4. Packaging SIMM | 123 |
| Gambar 8.5. Koreksi Kesalahan dengan Kode Hamming | 123 |
| Gambar 8.6. Hubungan Cache Memori | 126 |
| Gambar 8.7. Organisasi Cache Memori | 127 |
| Gambar 8.8. Organisasi <i>Cache</i> Pemetaan Langsung | 132 |
| Gambar 8.9. Organisasi <i>Cache</i> Pemetaan Asosiatif | 133 |
| Gambar 9.1. Layout Data Disk | 134 |
| Gambar 9.2. Metode Layout Data Disk | 135 |
| Gambar 9.3. Macam Disk Berdasarkan Gerakan Head | 136 |

| | |
|---|-----|
| Gambar 9.4. Disk Piringan Banyak (<i>Multiple Platter Disk</i>) | 143 |
| Gambar 9.5. Floppy Disk | 144 |
| Gambar 9.6. Layout Disk CLV..... | 147 |
| Gambar 9.7. Format Blok CD-ROM | 151 |
| Gambar 9.8. Format Fisik Pita Magnetik | 153 |
| Gambar 10.1. Model Generik dari suatu Modul I/O | 154 |
| Gambar 10.2. Skema Suatu Perangkat Peripheral | |
| Gambar 10.3. Blok Diagram Struktur Modul I/O..... | 158 |
| Gambar 10.4. Pemakaian Pengontrol Interupsi 8559A pada 8086 | 159 |
| Gambar 10.5. Modul I/O 8255A | 160 |
| Gambar 10.6. Interface Keyboard dan Display dengan Intel 8255A | 161 |
| Gambar 10.7. Blok Diagram DMA | 166 |
| Gambar 10.8. Konfigurasi Modul DMA | 167 |
| Gambar 11.1. Modul-modul Komputer | 170 |
| Gambar 11.2. Pola Interkoneksi Bus | 170 |
| Gambar 11.3. Arsitektur Bus Jamak Tradisional | 172 |
| Gambar 11.4. Arsitektur Bus Jamak Berkinerja Tinggi | 172 |
| Gambar 11.5. Contoh Pewaktuan Sinkron | |
| Gambar 11.6. Contoh Pewaktuan Asinkron | |

DAFTAR TABEL

| | |
|--|------------|
| Tabel 2.1. Karakteristik Penting Kelompok Sistem/360..... | 17 |
| Tabel 3.1. Posisi Value..... | 27 |
| Tabel 3.2. Posisi Bilangan Binari..... | 28 |
| Tabel 3.3. Tabel Posisi Bilangan Oktal..... | 29 |
| Tabel 3.4. Posisi Bilangan Hexadesimal | 30 |
| Tabel 3.5. Digit Oktal | 32 |
| Tabel 3.6. Digit Hexadesimal | 33 |
| Tabel 4.1. Tabel Kebenaran Rangkaian Logika Satu Variabel Input | 43 |
| Tabel 4.2. Tabel Kebenaran Rangkaian Logika Dua Variabel Input | 43 |
| Tabel 4.3. Tabel Kebenaran Rangkaian Logika Tiga Variabel Input | 43 |
| Tabel 4.4. Tabel Kebenaran Gerbang AND 2 Input..... | 44 |
| Tabel 4.5. Tabel Kebenaran Gerbang OR 2 Input..... | 45 |
| Tabel 4.6. Tabel Kebenaran Gerbang NOT 2 Input..... | 46 |
| Tabel 4.7. Teorema-teorema Aljabar Boole untuk Variabel Tunggal | 47 |
| Tabel 4.8. Teorema-teorema Aljabar Boole untuk Variabel Jamak | 49 |
| Tabel 5.1. Tabel Kebenaran Fungsi X | 49 |
| Tabel 5.2. Tabel Kebenaran Fungsi R | 56 |
| Tabel 5.3. Kelompok Jumlah Logika 1..... | 59 |
| Tabel 6.1. Tabel Kebenaran Flip Flop Dasar dengan Gerbang NOR | 68 |
| Tabel 6.2. Tabel Kebenaran Flip Flop Dasar dengan Gerbang NAND..... | 76 |
| Tabel 6.3. Tabel Karakteristik Flip Flop RS Menurut Waktu.... | 78 |
| Tabel 6.4. Tabel Karakteristik Flip Flop D | 80 |
| Tabel 6.5. Tabel Karakteristik Flip Flop JK..... | 82 |
| Tabel 6.6. Tabel Karakteristik Flip Flop T | 84 |
| Tabel 6.7. Tabel Kebenaran Encoder Oktal ke Biner | 85 |
| Tabel 6.8. Tabel Kebenaran Decoder Biner ke Oktal | 86 |
| Tabel 6.9. Tabel Kebenaran Decoder BCD ke Desimal | 88 |
| Tabel 6.10. Tabel Kebenaran Decoder common anoda..... | 91 |
| Tabel 6.11. Tabel Kebenaran Decoder common katoda..... | 93 |
| Tabel 8.1. Karakteristik Penting Sistem Memori Komputer | 94 |
| Tabel 8.2. Tabel Spesifikasi Memori | 111 |
| Tabel 8.3. Tingkatan Satuan Memori | 115 |
| Tabel 8.4. Tipe-tipe Memori Semikonduktor | 115 |
| Tabel 8.5. Penambahan Bit Cek Paritas untuk koreksi kode Hamming..... | 117 120 |

| | |
|--|-----|
| Tabel 8.6. Unsur-unsur Rancangan Cache Memori | 124 |
| Tabel 9.1. Karakteristik Magnetik Disk | 133 |
| Tabel 9.2. Karakteristik Berbagai Macam Disket | 137 |
| Tabel 9.3. Versi Disk SCSI | 138 |

SATUAN BAHASAN 1

PENGANTAR ORGANISASI DAN ARSITEKTUR KOMPUTER

1. Gambaran Singkat Mengenai Materi Kuliah

Materi kuliah dari satuan bahasan yang pertama ini berisikan tentang pengertian dari komputer, perbedaan antara organisasi dengan arsitektur komputer. Juga membahas tentang fungsi dan struktur dasar dari komputer.

2. Pedoman Mempelajari Materi

Baca dengan cermat dan teliti materi pada satuan bahasan 1, kemudian pahami setiap teori yang akan membantu pemahaman anda tentang organisasi dan arsitektur komputer itu sendiri. Pada akhir satuan bahasan terdapat tes formatif yang perlu anda kerjakan sebagai sarana untuk mengukur seberapa jauh tingkat pemahaman anda terhadap materi ini.

3. Tujuan Pembelajaran

Setelah mempelajari satuan bahasan 1 ini, mahasiswa diharapkan dapat :

- menjelaskan pengertian komputer dan bagian-bagian utamanya.
- menjelaskan perbedaan antara organisasi komputer dengan arsitektur komputer beserta dengan contohnya masing-masing
- menjelaskan struktur utama komputer beserta fungsinya
- menjelaskan struktur utama dari CPU beserta fungsinya
- engetahui fungsi-fungsi operasi komputer

1.1. Komputer

Kemajuan ilmu pengetahuan sebagai bagian dari peradaban manusia, telah menciptakan perubahan besar dalam kehidupan manusia. Perkembangan teknologi telah menghasilkan suatu perangkat alat hitung yang dulunya masih menggunakan mesin sederhana, kini telah menjelma menjadi perangkat teknologi yang luar biasa.

Disinilah peranan ilmu pengetahuan yang senantiasa wajib dikembangkan demi kemaslahatan umat manusia. Allah mewajibkan setiap umat manusia untuk terus belajar dan menggali potensi ilmunya, sebagaimana dijelaskan dalam QS.Al-Alaq/96:1-5.

أَقْرَأْ بِأَسْمِ رَبِّكَ الَّذِي خَلَقَ ۝ خَلَقَ الْإِنْسَانَ مِنْ عَلَقٍ ۝ أَلَمْ يَكُنْ أَقْرَأَ ۝ وَرَبُّكَ
الْأَكْرَمُ ۝ الَّذِي عَلَّمَ بِالْقَلَمِ ۝ عَلَّمَ الْإِنْسَانَ مَا لَمْ يَعْلَمْ ۝

Terjemahnya: Bacalah dengan (menyebut) nama Tuhanmu yang Menciptakan. Dia telah menciptakan manusia dari segumpal darah. Bacalah, dan Tuhanmulah yang Maha pemurah. Yang mengajar (manusia) dengan perantaran kalam. Dia mengajar kepada manusia apa yang tidak diketahuinya. (Departemen Agama, 2005)

Ayat tersebut menggambarkan bahwa manusia awalnya tidak punya apa-apa, kecuali Allah mengajarkan melalui perantaraan kalam sehingga manusia mengenal teknologi.

Salah satu produk teknologi yang dihasilkan dari kemajuan ilmu pengetahuan adalah komputer. Komputer adalah sebuah mesin hitung elektronik yang secara cepat menerima informasi masukan digital dan mengolah informasi tersebut menurut seperangkat instruksi yang tersimpan dalam komputer tersebut dan menghasilkan keluaran informasi yang dihasilkan setelah diolah. Daftar perintah tersebut dinamakan program komputer dan unit penyimpanannya adalah memori komputer.

Menurut Hartono (1998), ada beberapa definisi tentang komputer yang disajikan oleh beberapa pakar komputer seperti:

Menurut buku *Computer Annual* (Robert H.Blissmer) : komputer adalah suatu alat elektronik yang mampu melakukan beberapa tugas berikut : menerima input; memproses input sesuai dengan programnya, menyimpan perintah-perintah dan hasil dari pengolahan; dan menyediakan output dalam bentuk informasi.

Menurut buku *Computer Organization* (V.C.Hamacher, Z.G.Vranezic, S.G.Zaky): Komputer adalah mesin penghitung elektronik yang cepat, dapat menerima informasi input digital, memprosesnya sesuai dengan suatu program yang tersimpan di memorinya (*stored program*), dan menghasilkan output informasi.

Menurut buku *Introduction to Computer* (Gordon B.Davis): Komputer adalah tipe khusus alat penghitung yang mempunyai sifat tertentu yang pasti.

Menurut buku *Computer Today* (Donald H.Sanders): Komputer adalah sistem elektronik untuk memanipulasi data yang cepat dan tepat serta dirancang dan diorganisasikan supaya secara otomatis menerima dan menyimpan data input, memprosesnya dan menghasilkan output dibawah pengawasan suatu langkah-langkah instruksi-instruksi program yang tersimpan di memori (*stored program*).

Dalam bentuk yang paling sederhana komputer terdiri dari lima bagian utama yang mempunyai fungsi sendiri-sendiri. Unit-unit tersebut adalah : masukan, memori, aritmetika dan logika, keluaran dan kontrol seperti pada gambar 1.1.



Gambar 1.1. Unit fungsional dasar pada komputer

Unit masukan menerima informasi yang dikodekan dari operator manusia lewat alat-alat elektromekanik seperti papan ketik pada suatu terminal video, atau dari komputer-komputer lain lewat

jalur komunikasi digital. Informasi yang diterima dan disimpan dalam memori untuk dipergunakan kelak, atau langsung diolah oleh rangkaian aritmetika dan logika untuk melaksanakan operasi yang diinginkan. Langkah-langkah pengolahan ditentukan oleh program yang disimpan dalam memori. Akhirnya hasil-hasil yang diperoleh dikirimkan kembali keluar melalui unit keluaran. Seluruh kegiatan ini dikoordinasi oleh unit kontrol.

1.2. Pengertian Organisasi dan Arsitektur Komputer

Organisasi Komputer adalah bagian yang terkait erat dengan unit-unit operasional dan interkoneksi antar komponen penyusun sistem komputer dalam merealisasikan aspek arsitekturalnya. Contoh aspek organisasional adalah teknologi hardware, perangkat antarmuka, teknologi memori, sistem memori, dan sinyal-sinyal kontrol.

Arsitektur Komputer lebih cenderung pada kajian atribut-atribut sistem komputer yang terkait dengan seorang programmer. Contohnya, set instruksi, aritmetika yang digunakan, teknik pengalamatan, mekanisme I/O.

Sebagai contoh apakah suatu komputer perlu memiliki instruksi pengalamatan tanpa memori merupakan masalah rancangan arsitektural. Apakah instruksi pengalamatan tersebut akan diimplementasikan secara langsung ataukah melalui mekanisme *cache* adalah kajian organisasional.

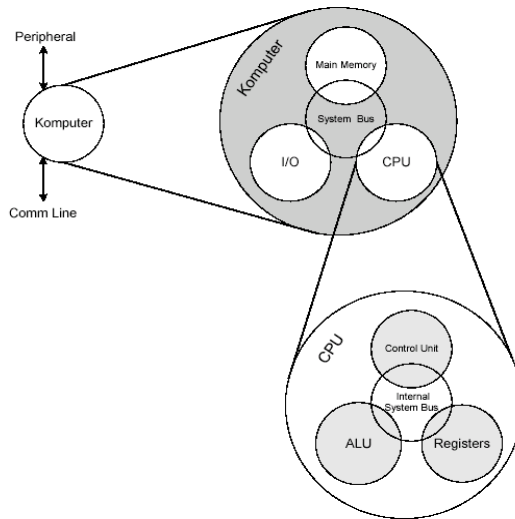
1.3. Struktur dan Fungsi Utama Komputer

1.3.1. Struktur Komputer

Komputer adalah sebuah sistem yang berinteraksi dengan cara tertentu dengan dunia luar. Interaksi dengan dunia luar dilakukan melalui perangkat peripheral dan saluran komunikasi. Dalam buku ini akan banyak dikaji seputar struktur internal komputer. Perhatikan gambar 1.2, terdapat empat struktur utama :

- *Central Processing Unit* (CPU), berfungsi mengendalikan kerja komputer, pemrosesan data dan pusat pengolahan fungsi – fungsi komputer. CPU sering juga disebut sebagai *processor* (prosesor).
- *Memori Utama*, berfungsi sebagai penyimpan data yang akan atau baru saja diproses.
- *I/O (Input/Output)*, berfungsi memindahkan data antara komputer ke lingkungan luar atau perangkat lainnya.

- *System Interconnection*, yang lebih dikenal dengan istilah *bus* merupakan mekanisme yang menghubungkan komunikasi internal antara CPU, memori utama dan I/O.



Gambar 1.2. Struktur Dasar Komputer

Salah satu komponen yang paling kompleks dan merupakan ‘jantung’nya komputer adalah CPU. Struktur CPU terlihat pada gambar 1.2 dengan struktur utamanya adalah :

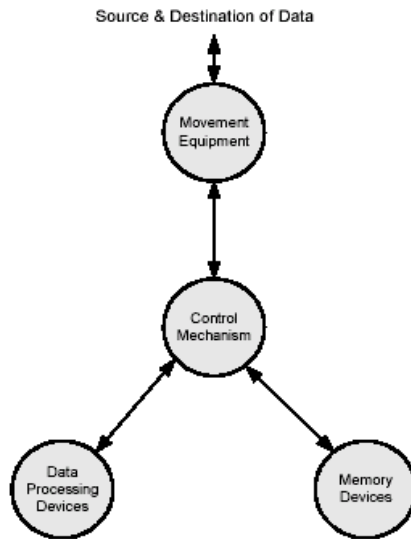
- *Control Unit*, berfungsi untuk mengontrol operasi CPU dan mengontrol komputer secara keseluruhan.
 - *Arithmetic And Logic Unit (ALU)*, berfungsi untuk membentuk fungsi– fungsi pengolahan data komputer.
 - *Register*, berfungsi sebagai penyimpan internal bagi CPU.
- CPU Interconnection*, berfungsi menghubungkan seluruh bagian dari CPU.

1.3.2. Fungsi Komputer

Fungsi dasar sistem komputer adalah sederhana seperti terlihat pada gambar 1.3. Pada prinsipnya terdapat empat buah fungsi operasi, yaitu :

- a. Fungsi Operasi Pengolahan Data, seperti penjumlahan, pengurangan, konversi terhadap data, dan lain sebagainya,

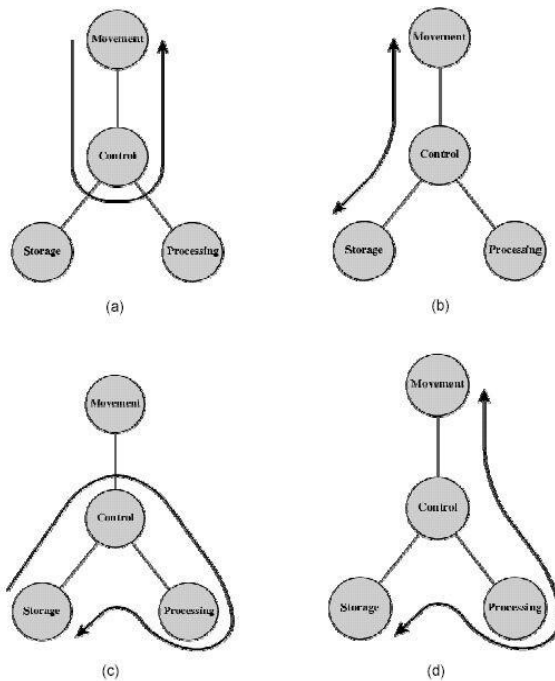
- b. Fungsi Operasi Penyimpanan Data, seperti penyimpanan data hasil pemrosesan, pengambilan data yang telah disimpan
- c. Fungsi Operasi Pemindahan Data, seperti pergerakan data internal dan pergerakan data eksternal,
- d. Fungsi Operasi Kontrol, berfungsi mengendalikan ketiga fungsi diatas.



Gambar 1.3. Fungsi Dasar Komputer

Komputer harus dapat *memproses data*. Representasi data disini bermacam–macam, akan tetapi nantinya data harus disesuaikan dengan mesin pemrosesnya. Dalam pengolahan data, komputer memerlukan unit penyimpanan sehingga diperlukan suatu mekanisme *penyimpanan data*. Walaupun hasil komputer digunakan saat itu, setidaknya komputer memerlukan media penyimpanan untuk data prosesnya. Dalam interaksi dengan dunia luar sebagai fungsi *pemindahan data* diperlukan antarmuka (*interface*), proses ini dilakukan oleh unit *Input/Output (I/O)* dan perangkatnya disebut *peripheral*. Saat interaksi dengan perpindahan data yang jauh atau dari remote device, komputer melakukan proses *komunikasi data*. Gambar 1.4 mengilustrasikan operasi–operasi komputer. Gambar 1.4a adalah operasi pemindahan data, gambar 1.4b adalah operasi penyimpanan

data, gambar 1.4c adalah operasi pemrosesan data dan gambar 1.4d adalah operasi pengolahan data.



Gambar 1.4. Operasi-operasi komputer

Rangkuman

1. Dalam bentuk yang paling sederhana komputer terdiri dari lima bagian utama yang mempunyai fungsi sendiri-sendiri. Unit-unit tersebut adalah : masukan, memori, aritmetika dan logika, keluaran dan kontrol
2. *Organisasi Komputer* adalah bagian yang terkait erat dengan unit-unit operasional dan interkoneksi antar komponen penyusun sistem komputer dalam merealisasikan aspek arsitekturalnya. Sedangkan

Arsitektur Komputer lebih cenderung pada kajian atribut–atribut sistem komputer yang terkait dengan seorang programmer.

3. Terdapat 4 struktur utama dalam komputer yaitu : CPU, memori utama, Input/Output, dan System Interkoneksi (sistem bus)
4. Terdapat 4 struktur utama dalam CPU yaitu : Control Unit, ALU, Register, dan System Interkoneksi (sistem bus).
5. Fungsi dasar komputer meliputi fungsi operasi pengolahan data, penyimpanan data, pemindahan data, dan fungsi control.

Tes Formatif

1. Jelaskan perbedaan antara pengertian organisasi komputer dengan arsitektur komputer !
2. Jelaskan struktur utama dari komputer beserta fungsinya masing-masing !
3. Jelaskan pula struktur utama dari CPU beserta fungsinya masing-masing !
4. Sebutkan fungsi-fungsi operasi komputer dan berikan contohnya masing-masing !

SATUAN BAHASAN 2

EVOLUSI DAN KINERJA KOMPUTER

1. Gambaran Singkat Mengenai Materi Kuliah

Dalam materi kuliah pada satuan bahkomputerasan 2 ini mula-mula akan dibahas tentang sejarah singkat komputer yang terbagi dalam beberapa generasi karena beberapa diantaranya merupakan dasar dari perkembangan komputer saat ini. Kemudian dibahas mengenai bermacam-macam teknik dan strategi yang digunakan untuk mencapai unjuk kerja yang seimbang dan efisien. Terakhir akan dibahas evolusi dua sistem yang sangat penting pada komputer saat ini yaitu Pentium dan Power PC.

2. Pedoman Mempelajari Materi

Baca dengan cermat dan teliti materi pada satuan bahasan 2, kemudian pahami setiap bagian dari evolusi komputer. Hal ini akan membantu anda dalam memahami sejarah perkembangan komputer beserta kinerjanya setiap generasi. Pada akhir satuan bahasan terdapat tes formatif yang perlu anda kerjakan sebagai sarana untuk mengukur seberapa jauh tingkat pemahaman anda terhadap materi ini.

3. Tujuan Pembelajaran

Setelah mempelajari satuan bahasan 2 ini, mahasiswa diharapkan dapat :

- mengetahui sejarah perkembangan komputer di setiap generasinya
- mengetahui jenis-jenis komputer yang dirancang di setiap generasinya
- menjelaskan beberapa teknik yang dikembangkan dalam perancangan kinerja komputer
- menjelaskan evolusi komputer pada kelompok Intel Pentium dan Power PC

2.1. Sejarah Perkembangan Komputer

Sebagai bagian dari perkembangan teknologi informasi dewasa ini, komputer yang digunakan saat ini tentunya melalui proses evolusi yang panjang. Komputer yang dulunya adalah alat hitung yang sangat sederhana kini menjadi suatu perangkat teknologi yang saat ini memudahkan segala aktifitas manusia. Bahkan dalam kurun waktu lebih dari sepuluh tahun ini, menjelma menjadi perangkat yang mampu menggabungkan berbagai teknologi dalam bentuk laptop, gadget yang sifatnya *mobile* dan lebih dinamis.

Sejarah perkembangan komputer dikelompokkan dalam beberapa generasi. Hal ini bertujuan untuk memudahkan pemahaman tentang perubahan teknologi yang digunakan setiap generasinya. Perubahan teknologi tidak serta merta terjadi begitu saja. Manusia diberi akal dan ilmu untuk terus mengembangkan pengetahuannya dan menghasilkan penemuan-penemuan yang luar biasa. Allah swt telah memberikan ilmu kepada manusia dan melebihkannya dibanding makhluk-makhluk ciptaan lainnya sebagaimana dijelaskan dalam QS.An-Naml/27:15.

وَلَقَدْ آتَيْنَا دَاوُدَ وَسُلَيْمَانَ عِلْمًا وَقَالَا الْحَمْدُ لِلَّهِ الَّذِي فَضَّلَنَا عَلَى كَثِيرٍ مِّنْ

عِبَادِهِ الْمُؤْمِنِينَ ﴿١٥﴾

Terjemahnya: “dan Sesungguhnya Kami telah memberi ilmu kepada Daud dan Sulaiman; dan keduanya mengucapkan: "Segala puji bagi Allah yang melebihkan Kami dari kebanyakan hamba-hambanya yang beriman".

Ayat tersebut menggambarkan bahwa Allah memberikan ilmu pengetahuan kepada manusia untuk dimanfaatkan dan dikembangkan, sehingga dapat menghasilkan sesuatu yang dapat bermanfaat bagi umat manusia, tentu saja diantaranya teknologi komputer yang sudah kita rasakan manfaatnya saat ini.

2.1.1 Komputer Primitif

Pada awalnya cikal bakal komputer dimulai pada tahun 1642 oleh Blaise Pascal yang menciptakan mesin hitung pertama yang

diberi nama Kalkulator Pascal. Kemudian di tahun 1672 ilmuwan Leibniz menambahkan beberapa fungsi lain dan menghasilkan Kalkulator Leibniz.

Sekitar 150 tahun kemudian, seorang professor Matematika dari Universitas Cambridge, Charles Babbage (1791-1871) merancang suatu mesin yang dikenal dengan nama Difference Engine di tahun 1823. Mesin yang dibuat ini masih sangat sederhana dengan menggunakan bahan-bahan mekanik. Mesin ini mampu menghitung tabel matematis.

Sepuluh tahun setelah mesin pertamanya dirancang, tahun 1833 Babbage membuat konsep mesin baru yaitu Analytic Engine. Mesin ini dapat dianggap sebagai konsep awal sebuah komputer serbaguna yang dapat digunakan untuk berbagai tujuan. (Abdurohman, 2014).

Satu dekade berikutnya adalah masa peralihan menuju awal generasi komputer. Perkembangan komputer waktu itu tidak hanya untuk keperluan penelitian, tetapi juga digunakan untuk berbagai kepentingan dalam perang dunia II.

John Atanasoff seorang ilmuwan dari Iowa State University, tahun 1930-an merancang sebuah mesin *Linear Equation Solver* (pemecah persamaan linear). Tidak kurang dari 300 buah tabung digunakannya sebagai teknologi dasar pembangun rancangannya ini. Mesin ini mampu menyelesaikan persamaan linear dan integral diferensial.

Atanasoff-Berry Computer (1937-1942) adalah komputer digital pertama yang mengimplementasikan perhitungan biner untuk menyelesaikan persamaan linear. Hampir bersamaan, Konrad Zuse, ilmuwan Jerman merancang Z3 (1941), sebuah komputer elektrik-mekanik serbaguna yang dapat diprogram. Di Inggris, Tommy Flowers merancang Colossus (1944), komputer elektronik yang dapat diprogram. Harvard Mark I (1944) dirancang oleh Howard Aiken bekerjasama dengan IBM sejak 1939. Harvard Mark I dikenal juga sebagai IBM ASCC (*Automatic Sequence Controlled Calculator*), merupakan komputer digital otomatis pertama di Amerika. (Abdurohman, 2014).

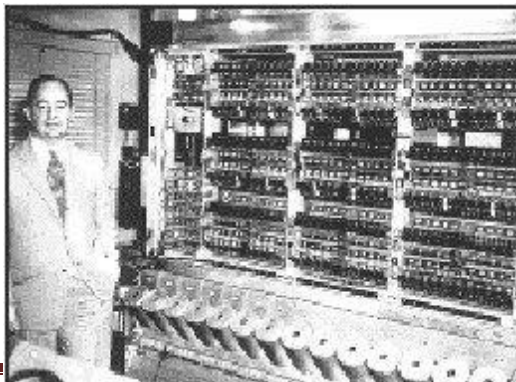
2.1.2 Generasi Pertama : Tabung Vakum (1945 – 1955)

ENIAC

ENIAC (*Electronic Numerical Integrator And Computer*), pada tahun 1946 dirancang dan dibuat oleh John Mauchly dan John Presper Eckert di Universitas Pennsylvania merupakan komputer digital elektronik untuk kebutuhan umum pertama di dunia. ENIAC dibuat dibawah lembaga *Army's Ballistics Research Laboratory* (BRL). Sebuah badan yang bertanggungjawab dalam pembuatan jarak dan table lintasan peluru kendali senjata baru. Sebelumnya tugas ini dilakukan oleh kurang lebih 200 personil dengan menggunakan kalkulator untuk menyelesaikan persamaan matematis peluru kendali yang memakan waktu lama.

ENIAC mempunyai berat 30 ton, bervolume 15.000 kaki persegi, dan berisi lebih dari Tabung vakum. Daya listrik yang dibutuhkan sebesar 140KW. Kecepatan operasi mencapai 5.000 operasi penambahan perdetik. ENIAC masih merupakan mesin desimal, representasi data bilangan dalam bentuk decimal dan arimetiknya dibuat dalam bentuk desimal. Memorinya terdiri atas 20 akumulator, yang masing – masing akumulatornya mampu menampung 10 digit desimal. Setiap digit direpresentasikan oleh cincin yang terdiri atas 10 buah tabung vakum. Kekurangan utama mesin ini adalah masih manual pemrogramannya, yaitu dengan menyetel switch–switch, memasang dan menanggalkan kabel–kabelnya. ENIAC selesai pada tahun 1946 sejak proposal diajukan tahun 1943, sehingga tahun 1946 merupakan gerbang bagi zaman baru komputer elektronik.

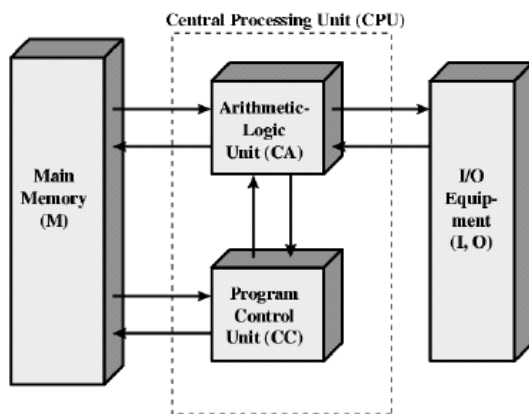
John Van Neumann seorang ahli matematika yang merupakan konsultan pembuatan ENIAC pada tahun 1945 mencoba memperbaiki kelemahan ENIAC dengan rancangan komputer barunya, bernama **EDVAC** (*Electronic Discrete Variable Computer*) dengan konsep program tersimpan (*stored-program concept*).



Gambar 2.1. John von Neumann

Tahun 1946 komputer dengan *stored-program concept* dipublikasikan, yang kemudian dikenal dengan Komputer IAS (*Computer of Institute for Advanced Studies*). Struktur komputer IAS terlihat pada gambar 2.2. Komputer ini terdiri :

- *Memori Utama*, untuk menyimpan data maupun instruksi.
- *Arithmetic Logic Unit (ALU)*, untuk mengolah data biner.
- *Control Unit*, untuk melakukan interpretasi instruksi-instruksi di dalam memori sehingga adanya eksekusi instruksi tersebut.
- *I/O*, untuk berinteraksi dengan lingkungan luar.



Gambar 2.2. Struktur Komputer IAS

Memori IAS terdiri atas 1.000 lokasi penyimpanan yang disebut *word*. Word terdiri atas 40 *binary digit* (bit). Data maupun instruksi disimpan dalam memori ini, sehingga data maupun instruksi harus dikodekan dalam bentuk biner. Format memori terlihat pada gambar 2.2. Setiap bilangan terdiri atas sebuah bit tanda dan 39 bit nilai. Sebuah word terdiri atas 20 bit instruksi dengan masing– masing 8 bit kode operasi (*op code*) dan 12 bit alamat.

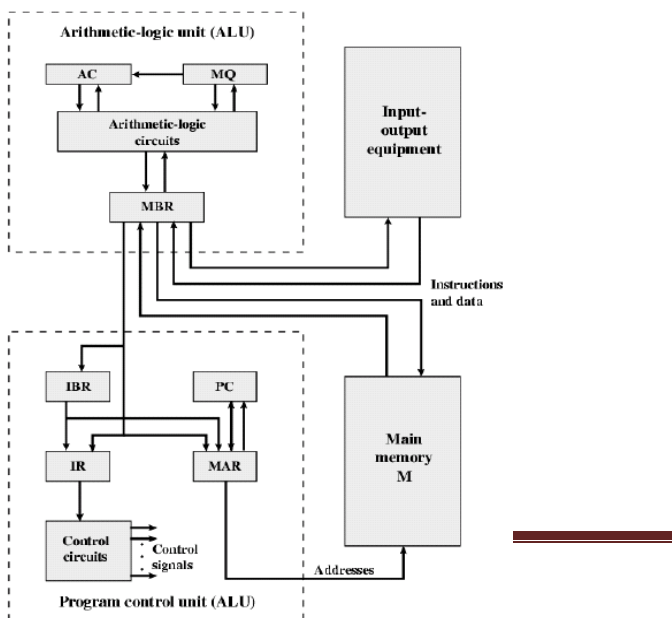
| Instruksi kiri | Instruksi kanan |
|----------------------------------|-----------------|
| Organisasi & Arsitektur Komputer | 13 |

| | | | |
|--------|--------|---------|--------|
| OpCode | Alamat | Op Code | Alamat |
|--------|--------|---------|--------|

Gambar 2.3 Format memori IAS

Struktur detail komputer IAS disajikan dalam gambar 2.3. Gambar ini menjelaskan bahwa baik unit kontrol maupun ALU berisi lokasi-lokasi penyimpanan, yang disebut *register*, yaitu :

- *Memory Buffer Register (MBR)*, berisi sebuah word yang akan disimpan didalam memori atau digunakan untuk menerima word dari memori.
- *Memory Address Register (MAR)*, untuk menentukan alamat word dimemori untuk dituliskan dari MBR atau dibaca oleh MBR.
- *Instruction Register (IR)*, berisi instruksi 8 bit kode operasi yang akan dieksekusi.
- *Instruction Buffer Register (IBR)*, digunakan untuk penyimpanan sementara instruksi sebelah kanan word didalam memori.
- *Program Counter (PC)*, berisi alamat pasangan instruksi berikutnya yang akan diambil dari memori.
- *Accumulator (AC)* dan *Multiplier Quotient (MQ)*, digunakan untuk penyimpanan sementara operand dan hasil ALU. Misalnya, hasil perkalian 2 buah bilangan 40 bit adalah sebuah bilangan 80 bit; 40 bit yang paling berarti (*most significant bit*) disimpan dalam AC dan 40 bit lainnya (*least significant bit*) disimpan dalam MQ.



Gambar 2.4 Struktur detail komputer IAS

IAS beroperasi secara berulang membentuk *siklus instruksi*. Komputer IAS memiliki 21 instruksi, yang dapat dikelompokkan seperti berikut ini :

- *Data transfer*, memindahkan data diantara memori dengan register–register ALU atau antara dua register ALU sendiri.
- *Unconditional branch*, perintah – perintah eksekusi percabangan tanpa syarat tertentu.
- *Conditional branch*, perintah–perintah eksekusi percabangan yang memerlukan syarat tertentu agar dihasilkan suatu nilai dari percabangan tersebut.
- *Arithmetic*, kumpulan operasi– operasi yang dibentuk oleh ALU.
- *Address Modify*, instruksi– instruksi yang memungkinkan pengubahan alamat saat dikomputasi sehingga memungkinkan fleksibilitas alamat yang tinggi pada program.

Komputer Komersial

Tahun 1950 dianggap sebagai tahun kelahiran industri komputer dengan munculnya 2 buah perusahaan yang saat itu mendominasi pasar, yaitu Sperry dan IBM.

Tahun 1947, Eckert dan Mauchly mendirikan Eckert-Mauchly Computer Corporation untuk memproduksi komputer secara komersial. Komputer pertama yang mereka hasilkan adalah UNIVAC I (*Universal Automatic Computer*). UNIVAC I menjadi tulang punggung penghitungan sensus tahun 1950 di USA.

UNIVAC II yang memiliki kapasitas memori lebih besar dan kinerja yang lebih baik diluncurkan tahun 1950. Mulai saat itu perusahaan telah mengembangkan produk–produk baru yang kompatibel dengan produk sebelumnya sehingga pangsa pasar konsumen mereka tetap terjaga menggunakan produknya.

IBM pun tidak mau kalah dengan mengeluarkan produk mereka yang akhirnya mendominasi pangsa pasar bisnis saat ini. Seri IBM pertama adalah seri 701 tahun 1953 dan terus berkembang menjadi lebih baik hingga sekarang.

2.1.3 Generasi Kedua : Transistor (1955 – 1965)

Sejak pesatnya teknologi semi konduktor hingga menghasilkan komponen transistor membawa perubahan besar pada dunia

komputer. Komputer era ini tidak lagi menggunakan tabung vakum yang memerlukan daya operasional besar, tabung-tabung itu digantikan komponen kecil bernama transistor. Konsumsi daya listrik amat kecil dan bentuknya pun relatif kecil.

Transistor ditemukan di Bell Labs pada tahun 1947 dan tahun 1950 telah meluncurkan revolusi elektronika modern. IBM sebagai perusahaan pertama yang meluncurkan produk komputer dengan transistor sehingga tetap mendominasi pangsa pasar komputer. NCR dan RCA adalah perusahaan yang mengembangkan komputer berukuran kecil saat itu, kemudian diikuti IBM dengan mengeluarkan seri 7000-nya.

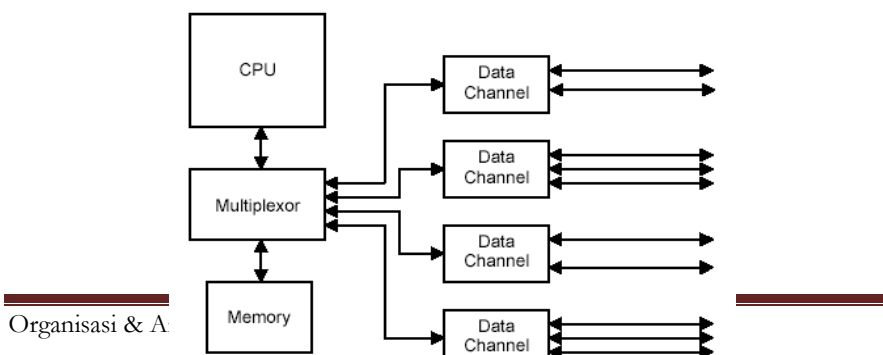
Dengan adanya transistor membuat hardware komputer saat itu makin cepat prosesnya. Memori makin besar kapasitasnya namun makin kecil bentuknya. Generasi dua ini juga terdapat perubahan perkembangan pada ALU yang makin kompleks, lahirnya bahasa pemrograman tingkat tinggi maupun tersedianya software sistem operasi.

Generasi kedua juga ditandai munculnya *Digital Equipment Corporation* (DEC) tahun 1957 dan meluncurkan komputer pertamanya, yaitu PDP-1. Komputer ini sangat penting bagi perkembangan komputer generasi ketiga.

IBM 7094

Komputer ini diluncurkan tahun 1962. Kemajuan IBM 7094 adalah adanya *Instruction Backup Register* (IBR) yang berfungsi membeffer instruksi berikutnya, efeknya komputer akan lebih cepat prosesnya. Unit control mengambil dua word yang berdampingan dari memori untuk sebuah pengambilan instruksi, kecuali bila terjadi percabangan.

Kemajuan IBM 7094 lainnya adalah adanya *multiplexor* untuk memultiplex *data channel* (saluran data). Multiplexor berfungsi sebagai sentral switch data yang akan diproses dalam CPU. Gambar 2.5 merupakan konfigurasi IMB 7094.



Gambar 2.5 Konfigurasi IBM 7094

2.1.4 Generasi Ketiga : Integrated Circuits (1965 – 1980)

Pada tahun 1958 terjadi revolusi elektronika kembali, yaitu ditemukannya *integrated circuit* (IC) yang merupakan penggabungan komponen–komponen elektronika dalam suatu paket. Dengan ditemukan IC ini semakin mempercepat proses komputer, kapasitas memori makin besar dan bentuknya semakin kecil.

IBM System/360

Tahun 1964 dikeluarkan IBM System/360 yang telah menggunakan teknologi IC. Dalam satu decade IBM menguasai 70% pasaran komputer.

Sistem 360 merupakan kelompok komputer pertama yang terencana. Banyak model dalam arsitektur 360 ini dan saling kompatibel. Hal ini sangat menguntungkan konsumen, karena konsumen dapat menyesuaikan dengan kebutuhan maupun harganya. Pengembangan (*upgrading*) dimungkinkan dalam komputer ini. Karakteristik komputer kelompok ini adalah :

- *Set Instruksi Mirip atau Identik*, dalam kelompok komputer ini berbagai model yang dikeluarkan menggunakan setinstruksi yang sama sehingga mendukung komabilitas sistem maupun perangkat kerasnya.
- *Sistem Operasi Mirip atau Identik*, ini merupakan feature yang menguntungkan konsumen sehingga apabila kebutuhan menuntut penggantian komputer tidak kesulitan dalam sistem operasinya karena sama.
- *Kecepatan yang meningkat*, model–model yang ditawarkan mulai dari kecepatan rendah sampai kecepatan tinggi untuk penggunaan yang dapat disesuaikan konsumen sendiri.
- *Ukuran Memori yang lebih besar*, semakin tinggi modelnya akan diperoleh semakin besar memori yang digunakan.
- *Harga yang meningkat*, semakin tinggi modelnya maka harganya semakin mahal.

Tabel 2.1 Karakteristik Penting Kelompok Sistem /360

| Karakteristik | Model | Model | Model | Model | Model |
|-------------------|-------|-------|-------|-------|-------|
| Ukuran memori(Kb) | 64 | 256 | 256 | 512 | 512 |

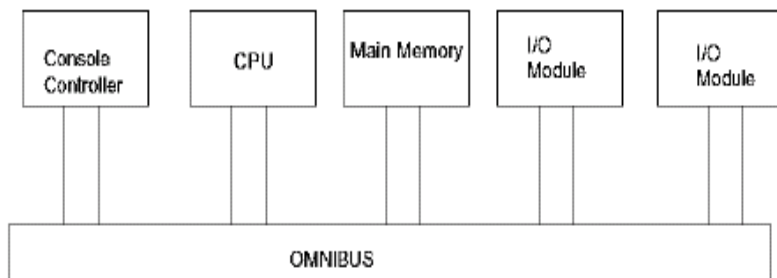
| | | | | | |
|-----------------------------|-----|-------|-----|------|------|
| Laju data dari memori | 0.5 | 0.8 | 2.0 | 8.0 | 16.0 |
| Prosesor cycle time(µdetik) | 1.0 | 0.625 | 0.5 | 0.25 | 0.2 |
| Jumlah maksimum data | 3 | 3 | 4 | 6 | 6 |
| Data maks per | 250 | 400 | 800 | 1250 | 1250 |

DEC PDP-8

Pada tahun yang sama saat IBM mengeluarkan System /360, DEC meluncurkan DEC PDP-8. Komputer ini memiliki keunggulan bentuknya yang kecil sehingga sangat fleksibel digunakan.

PDP-8 juga memiliki varian–varian yang modelnya sama dengan IBM System/360 untuk menyesuaikan kebutuhan pelanggannya. Dengan hadirnya PDP-8 ini membawa DEC sebagai perusahaan menyuplai komputer mini terbesar membawa DEC sebagai pabrik komputer terbesar kedua setelah IBM.

Arsitektur PDP-8 sangat berbeda dengan IBM terutama bagian sistem *bus*. Pada komputer ini menggunakan *omni bus system*. Sistem ini terdiri atas 96 buah lintasan sinyal yang terpisah, yang digunakan untuk membawa sinyal–sinyal kontrol, alamat maupun data. Karena semua komponen menggunakan jalur bus ini maka penggunaannya dikontrol oleh CPU. Arsitektur bus seperti PDP-8 ini nantinya digunakan oleh komputer–komputer modern selanjutnya. Struktur bus PDP-8 terlihat pada gambar 2.6.



Gambar 2.6. Struktur Bus PDP-8

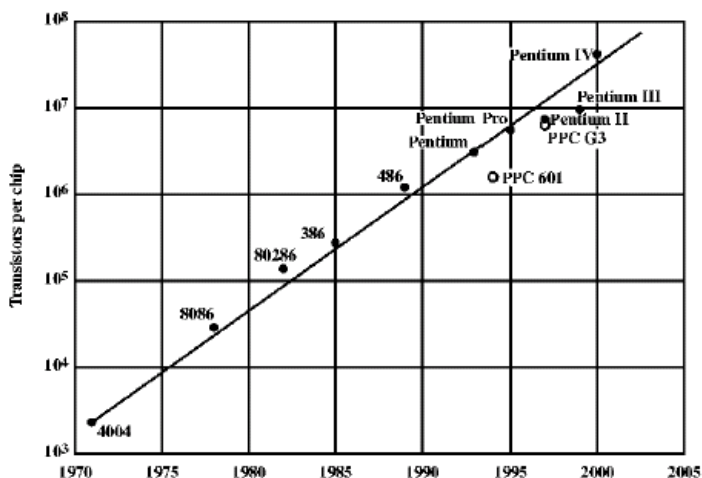
2.1.5 Generasi Keempat : Very Large Scale Integration (1980 - sekarang)

Era keempat perkembangan generasi komputer ditandai adanya VLSI. Paket VLSI dapat menampung 10.000 komponen lebih perkepingnya dengan kecepatan operasi mencapai 100 juta operasi

perdetiknya. Gambar 2.7 mengilustrasikan perkembangan mikroprosesor Pentium terhadap jumlah transistor per kepingnya. Masa-masa ini diawali peluncuran mikroprosesor Intel seri 4004.

Mikroprosesor 4004 dapat menambahkan dua bilangan 4 bit dan hanya dapat mengalikan dengan cara pengulangan penambahan. Memang masih primitif, namun mikroprosesor ini tonggak perkembangan mikroprosesor–mikroprosesor canggih saat ini. Tidak ada ukuran pasti dalam melihat mikroprosesor, namun ukuran terbaik adalah *lebarbus data*: jumlah bit data yang dapat dikirim–diterima mikroprosesor. Ukuran lain adalah *jumlah bit dalam register*.

Tahun 1972 diperkenalkan dengan mikroprosesor 8008 yang merupakan mikroprosesor 8 bit. Mikroprosesor ini lebih kompleks instruksinya tetapi lebih cepat prosesnya dari pendahulunya. Kemudian Bells dan HP menciptakan mikroprosesor 32 bit pada 1981, sedangkan Intel baru mengeluarkan tahun 1985 dengan mikroprosesor 80386.



Gambar 2.7 Grafik jumlah transistor dalam chips Pentium

2.2 Perancangan Kinerja

Kinerja sebuah sistem komputer merupakan hasil proses dari seluruh komponen komputer, yang melibatkan CPU, memori utama, memori sekunder, bus, peripheral. Dari segi perkembangan program

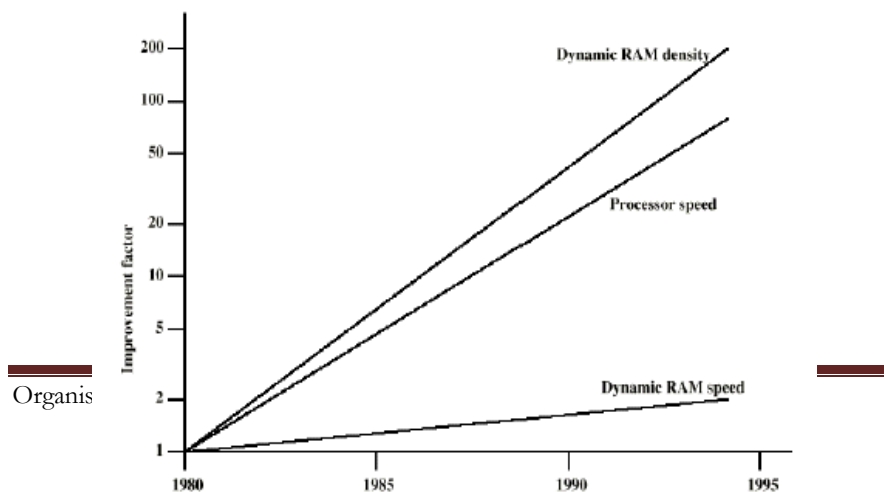
aplikasi pun sangat menakjubkan. Aplikasi desktop yang hampir dimiliki semua sistem komputer saat ini meliputi :

- Pengolahan citra
- Pengenalan voice atau pembicaraan
- Video conference
- Multimedia
- Transfer data

Yang menakjubkan lagi adalah dari sudut pandang organisasi dan arsitektur komputer saat ini adalah mirip dengan komputer IAS yang dibuat sekitar 50 tahun lalu, namun perkembangan dan kecanggihannya dapat kita rasakan sekarang ini. Peningkatan kinerja mikroprosesor ini terus berlanjut tidak kenal henti dengan berbagai teknik yang telah dikembangkan, diantaranya :

- *Branch Prediction*, teknik dimana prosesor memungkinkan mengamati terlebih dahulu didalam software dan melakukan prediksi percabangan atau kelompok instruksi yang akan dieksekusi berikutnya.
- *Data Flow Analysis*, prosesor akan menganalisa instruksi- instruksi yang tidak tergantung pada hasil atau data lainnya untuk membuat penjadwalan yang optimum dalam eksekusi.
- *Speculative Execution*, dengan modal prediksi cabang dan analisis data, maka prosesor dapat melakukan eksekusi spekulatif terlebih dahulu sebelum waktunya.

Perkembangan mikroprosesor, dilihat dari kapasitas operasi dan kecepatannya sangatlah pesat. Perkembangan mikroprosesor ini sulit diimbangi oleh komponen lainnya semisal memori. Hal ini menimbulkan masalah kesenjangan dan kurang sinkronnya operasi antar komponen. Perhatikan laju perkembangan prosesor dibandingkan memori utama seperti terlihat pada gambar 2.8. Organisasi dan arsitektur komputer yang handal sangat diperlukan untuk mengatasi persoalan seperti ini.



Gambar 2.8 Grafik perbandingan kecepatan mikroprosesor dan memori

Terdapat beberapa metode untuk mengatasi masalah perbedaan kecepatan operasi antara mikroprosesor dengan komponen lainnya, diantaranya :

- Meningkatkan jumlah bit yang dicari pada suatu saat tertentu dengan melebarkan DRAM dan melebarkan lintasa system busnya.
- Mengubah antar muka DRAM sehingga lebih efisien dengan menggunakan teknik *cache* atau pola *buffer* lainnya pada keeping DRAM.
- Meningkatkan bandwidth inter koneksi prosesor dan memori dengan menggunakan hierarki bus-bus yang lebih cepat untuk *buffering* dan membuat struktur aliran data.

Bidang lain yang menjadi fokus kajian peningkatan kinerja sistem komputer adalah penanganan perangkat-perangkat I/O. Masalah yang terjadi hampir sama dengan memori. Teknik penyelesaian yang digunakan umumnya adalah teknik *buffering* dan *caching*.

Target yang ingin dicapai dalam peningkatan kinerja adalah tercapainya keseimbangan proses operasi antar komponen-komponen penyusun komputer sehingga menghasilkan kinerja komputer yang tinggi.

2.3 Contoh Evolusi Komputer

Evolusi komputer yang akan dijelaskan adalah kelompok komputer Pentium Intel dan Power PC. Alasannya adalah komputer Pentium Intel mampu mendominasi pasaran dan secara teknologi menggunakan rancangan CISC (*complex instruction set computers*) dalam arsitekturnya. Sedangkan Power PC merupakan kelompok komputer yang menerapkan teknologi RISC (*reduced instruction set computers*).

Pentium

Pentium merupakan produk Intel yang mampu mendominasi pasaran prosesor hingga saat ini. Generasi demi generasi diluncurkan ke pasaran dengan kenaikan unjuk kerja yang menakjubkan dalam memenuhi kebutuhannya.

Berikut evolusi prosesor keluaran Intel dari prosesor sederhana sampai prosesor keluaran saat ini :

- Intel 4004, yang dikeluarkan pada tahun 1971 merupakan mikroprosesor pertama di dunia. Lebar bus data 4 bit, no interrupt dan jumlah pin 16 pin DIP.
- Intel 4040, dengan lebar bus data 4 bit, no interrupt dan jumlah pin 24 pin DIP.
- Intel 8008, dikeluarkan tahun 1972 merupakan mikroprosesor 8-bit pertama yang mampu melaksanakan 48 instruksi dengan ukuran memori 16 KB. Adanya instruksi tambahan menyebabkan prosesor ini dapat diaplikasikan dalam sejumlah aplikasi yang lebih maju.
- Intel 8080, keluar tahun 1973 merupakan mikroprosesor modern pertama keluaran Intel dengan mesin 8 bit dan bus data ke memori juga 8bit. Jumlah instruksinya 66 instruksi dengan kemampuan pengalamatan 16KB. Dapat melaksanakan instruksi 10 kali lebih cepat dari 8008.
- 8086, dikenalkan tahun 1974 adalah mikroprosesor 16 bit dengan teknologi cache instruksi. Jumlah instruksi mencapai 111 dan kemampuan pengalamatan ke memori 64KB.
- 80286, keluar tahun 1982 merupakan pengembangan dari 8086, kemampuan pengalamatan mencapai 1MB dengan 133 instruksi.
- 80386, keluar tahun 1985 dengan mesin 32 bit. Sudah mendukung sistem multi tasking. Dengan mesin 32 bitnya, produk ini mampu menjadi terunggul pada masa itu.
- 80486, dikenalkan tahun 1989. Kemajuannya pada teknologi *cache* memori dan *pipelining* instruksi. Sudah dilengkapi dengan *math co-processor*.
- Pentium, dikeluarkan tahun 1993, menggunakan teknologi *superscalar* sehingga memungkinkan eksekusi instruksi secara paralel.
- Pentium Pro, keluar tahun 1995. Kemajuannya pada peningkatan organisasi *superscalar* untuk proses paralel, ditemukan sistem prediksi cabang, analisa aliran data dan sistem cache memori yang makin canggih.
- Pentium II, keluar sekitar tahun 1997 dengan teknologi MMX sehingga mampu menangani kebutuhan multimedia. Mulai Pentium II telah menggunakan teknologi RISC.
- Pentium III, terdapat kemampuan instruksi *floating point* untuk menangani grafis 3D.
- Pentium IV, diproduksi tahun 2000 merupakan produk Intel yang kecepatan prosesnya mampu menembus kecepatan hingga 3,06 GHz.

- Itanium, memiliki kemampuan 2 unit floating point, 4 unit integer, 3 unit pencabangan, internet streaming, 128 integer register.
- Intel Core 2 Duo dibangun di beberapa fasilitas manufaktur bervolume tinggi dan canggih di dunia menggunakan proses berteknologi silikon 64-nanometer dari Intel. Intel Corporation meluncurkan prosesor Intel Core 2 Duo yang ditujukan bagi PC dan workstation desktop dan laptop consumer dan bisnis – prosesor dengan teknologi yang dapat menghasilkan kinerja lebih, konsumsi daya lebih kecil, serta keleluasaan pemakaian bagi para penggunanya.
- Intel Core i3, i5, dan i7, merupakan seri prosesor yang dikeluarkan oleh Intel dan saat ini banyak digunakan pada laptop maupun desktop.

PowerPC

Proyek sistem RISC diawali tahun 1975 oleh IBM pada komputer mini seri 801. Seri pertama ini hanyalah prototipe, seri komersialnya adalah PCRT yang dikenalkan tahun 1986. Tahun 1990 IBM mengeluarkan generasi berikutnya yaitu IBM RISC System /6000 yang merupakan mesin RISC super scalar work station. Setelah ini arsitektur IBM lebih dikenal sebagai arsitektur POWER.

IBM menjalin kerjasama dengan Motorola menghasilkan mikroprosesor seri 6800, kemudian Apple menggunakan keeping Motorola dalam Macintoshnya. Saat ini terdapat 4 kelompok Power PC, yaitu :

- 601, adalah mesin 32 bit merupakan produksi awal arsitektur Power PC untuk lebih dikenal masyarakat.
- 603, merupakan komputer desktop dan komputer portabel. Kelompok ini sama dengan seri 601 namun lebih murah untuk keperluan efisien.
- 604, seri komputer Power PC untuk kegunaan komputer *low-end server* dan komputer desktop.
- 620, ditujukan untuk penggunaan *high-end server*. Mesin dengan arsitektur 64 bit.
- 740/750, seri dengan cache L2.
- G4, seperti seri 750 tetapi lebih cepat dan menggunakan 8 instruksi paralel

Rangkuman

1. Sejarah perkembangan komputer dapat dikelompokkan dalam beberapa generasi berdasarkan teknologi yang digunakan.

2. ENIAC merupakan komputer digital elektronik untuk kebutuhan umum pertama di dunia.
3. Tahun 1946 komputer dengan konsep *stored program concept* dipublikasikan yang kemudian dikenal dengan komputer IAS (*Computer of Institute for Advance Studies*). Merupakan komputer yang menjadi pondasi sistem komputer modern yang digunakan saat ini.

Tes Formatif

1. Jelaskan secara umum perkembangan komputer ENIAC !
2. Mengapa John von Neumann dianggap sebagai peletak dasar sistem komputer modern !
3. Jelaskan struktur utama dari komputer IAS !
4. Jelaskan teknik-teknik yang dikembangkan dalam perancangan kinerja komputer !

SATUAN BAHASAN 4

GERBANG LOGIKA DASAR & ALJABAR BOOLEAN

1. Gambaran Singkat Mengenai Materi Kuliah

Materi kuliah dari satuan bahasan yang pertama ini berisikan tentang pengertian dari komputer, perbedaan antara organisasi dengan arsitektur komputer. Juga membahas tentang fungsi dan struktur dasar dari komputer.

2. Pedoman Mempelajari Materi

Baca dengan cermat dan teliti materi pada satuan bahasan 4, kemudian pahami setiap teori yang akan membantu pemahaman anda tentang organisasi dan arsitektur komputer itu sendiri. Pada akhir satuan bahasan terdapat tes formatif yang perlu anda kerjakan sebagai sarana untuk mengukur seberapa jauh tingkat pemahaman anda terhadap materi ini.

3. Tujuan Pembelajaran

Setelah mempelajari satuan bahasan 4 ini, mahasiswa diharapkan dapat :

- menjelaskan pengertian tabel kebenaran dan
- menjelaskan sifat dan karakteristik macam-macam gerbang logika
- mengetahui teorema-teorema Aljabar Boole, baik teorema variabel tunggal maupun teorema variabel jamak
- menghubungkan keterkaitan antara gerbang logika dengan tabel kebenaran.
- mengimplementasikan persamaan logika dalam bentuk rangkaian logika

4.1. Tabel Kebenaran (Truth Table)

Tabel kebenaran merupakan tabel yang menunjukkan pengaruh pemberian level logika pada input suatu rangkaian logika terhadap keadaan level logika outputnya. Melalui tabel kebenaran dapat diketahui watak atau karakteristik suatu rangkaian logika. Oleh karena itu, tabel kebenaran mencerminkan watak atau karakteristik suatu rangkaian logika. Sebagaimana manusia harus memiliki watak dan karakteristik untuk mempergunakan ilmu pengetahuannya ke jalan yang diridhai oleh Allah swt.

Berbicara mengenai ilmu, Allah swt dalam QS. Al ‘Ankabut/29:43 :

وَتِلْكَ الْأَمْثَلُ نَضْرِبُهَا لِلنَّاسِ وَمَا يَعْقِلُهَا إِلَّا الْعَالِمُونَ ﴿٤٣﴾

Terjemahannya : “Dan perumpamaan-perumpamaan ini Kami buat untuk manusia; dan tiada yang memahaminya kecuali orang-orang yang berilmu” (Departemen Agama RI,2005).

Makna ayat tersebut diatas adalah, Allah swt mengumpamakan sesuatu perumpamaan bagi manusia. Hanya orang berakal yang dapat memikirkan perumpamaan tersebut. Allah swt mengambil laba-laba sebagai perumpamaan, untuk memudahkan manusia untuk memahaminya. Selain dari itu, juga dimaksudkan untuk menerangkan segala keraguan mereka selama ini. Bagi orang yang selalu menggunakan hati dan pikirannya dan bagi ahli-ahli ilmu pengetahuan, pasti dapat memahami perumpamaan seperti tersebut dan akan semakin banyak mengetahui rahasia-rahasia Allah swt yang terkandung dalam ayat-ayat-Nya.

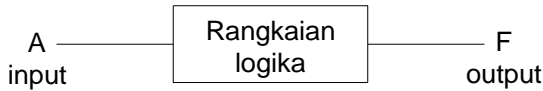
Dengan adanya ilmu, manusia dapat menemukan kebenaran dan rahasia alam semesta yang diciptakan oleh Allah swt. Seperti yang telah ditulis di atas, bahwa terdapat banyak jalan untuk menemukan kebenaran. Untuk menentukan proses penelusuran kebenaran tadi digunakanlah metodologi.

Tabel kebenaran harus memuat seluruh kemungkinan keadaan input tergantung pada jumlah variabel input atau jumlah saluran input dari suatu rangkaian logika, dan mengikuti rumus :

Jumlah seluruh kemungkinan input = 2^n , dengan n merupakan jumlah variabel atau saluran input rangkaian .

Contoh :

1. Rangkaian logika dengan 1 variabel input, maka jumlah seluruh kemungkinan input = $2^1 = 2$



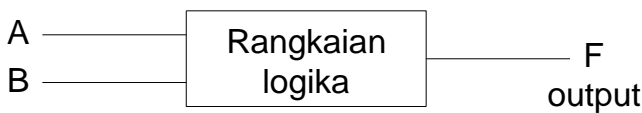
Gambar 4.1. Diagram blok rangkaian logika satu variabel input

Tabel kebenaran:

Tabel 4.1. Tabel kebenaran rangkaian logika satu variabel input

| Input (A) | Output (F) |
|-----------|------------|
| 0 | |
| 1 | |

2. Rangkaian logika dengan 2 variabel input, maka jumlah seluruh kemungkinan input = $2^2 = 4$



Gambar 4.2. Diagram blok rangkaian logika dua variabel input

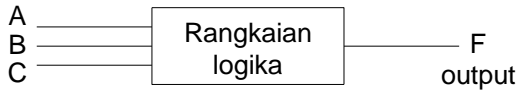
Tabel kebenaran:

Tabel 4.2. Tabel kebenaran rangkaian logika dua variabel input

| Input | | Output |
|-------|---|--------|
| A | B | F |
| 0 | 0 | |

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

3. Rangkaian logika dengan 3 variabel input, maka jumlah seluruh kemungkinan input = $2^3 = 8$



Gambar 4.3. Diagram blok rangkaian logika tiga variabel input

Tabel kebenaran:

Tabel 4.3. Tabel kebenaran rangkaian logika tiga variabel input

| Input | | | Output |
|-------|---|---|--------|
| A | B | C | F |
| 0 | 0 | 0 | |
| 0 | 0 | 1 | |
| 0 | 1 | 0 | |
| 0 | 1 | 1 | |
| 1 | 0 | 0 | |
| 1 | 0 | 1 | |
| 1 | 1 | 0 | |
| 1 | 1 | 1 | |

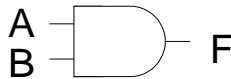
4.2. Gerbang Logika Dasar

Gerbang-gerbang dasar logika merupakan elemen rangkaian digital dan rangkaian digital merupakan kesatuan dari gerbang-gerbang logika dasar yang membentuk fungsi pemrosesan sinyal digital. Gerbang dasar logika terdiri dari 3 gerbang utama, yaitu AND Gate,

OR Gate, dan NOT Gate. Gerbang lainnya seperti NAND Gate, NOR Gate, EX-OR Gate dan EX-NOR Gate merupakan kombinasi dari 3 gerbang logika utama tersebut.

4.2.1. AND Gate

Gerbang AND merupakan salah satu gerbang logika dasar yang memiliki 2 buah saluran masukan (input) atau lebih dan sebuah saluran keluaran (output). Suatu gerbang AND akan menghasilkan sebuah keluaran biner tergantung dari kondisi masukan dan fungsinya. Prinsip kerja dari gerbang AND adalah kondisi keluaran (output) akan berlogic 1 bila semua saluran masukan (input) berlogic 1. Selain itu output akan berlogic 0. Simbol gerbang logika AND 2 input :



Gambar 4.4. Simbol Gerbang AND

dengan persamaan Boolean fungsi AND adalah $F = A.B$ (dibaca $F = A$ AND B).

Tabel kebenaran:

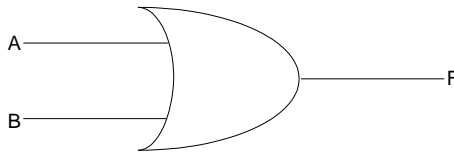
Tabel 4.4. Tabel kebenaran gerbang AND 2 input

| input | | Output |
|-------|---|--------|
| A | B | F |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

4.2.2. OR Gate

Gerbang OR merupakan salah satu gerbang logika dasar yang memiliki 2 buah saluran masukan (input) atau lebih dan sebuah saluran

keluaran (output). Berapapun jumlah saluran masukan yang dimiliki oleh sebuah gerbang OR, maka tetap memiliki prinsip kerja yang sama dimana kondisi keluarannya akan berlogik 1 bila salah satu atau semua saluran masukannya berlogik 1. Selain itu output berlogik 0. Simbol gerbang logika OR 2 input :



Gambar 4.5. Simbol Gerbang OR

dengan persamaan Boolean fungsi OR adalah $F = A+B$ (dibaca $F = A$ OR B).

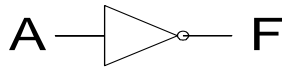
Tabel kebenaran:

Tabel 4.5. Tabel kebenaran Gerbang OR 2 input

| input | | Output |
|-------|---|--------|
| A | B | F |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

4.2.3. NOT Gate

Gerbang NOT sering disebut dengan gerbang inverter. Gerbang ini merupakan gerbang logika yang paling mudah diingat. Gerbang NOT memiliki 1 buah saluran masukan (input) dan 1 buah saluran keluaran (output). Gerbang NOT akan selalu menghasilkan nilai logika yang berlawanan dengan kondisi logika pada saluran masukannya. Bila pada saluran masukannya berlogik 1 maka pada saluran keluarannya akan berlogik 0 dan sebaliknya. Simbol gerbang logika NOT :



Gambar 4.6. Simbol Gerbang NOT

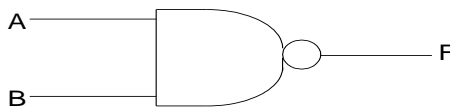
Tabel kebenaran:

Tabel 4.6. Tabel kebenaran Gerbang NOT 2 input

| Input (A) | Output (F) |
|-----------|------------|
| 0 | 1 |
| 1 | 0 |

4.2.4. NAND Gate

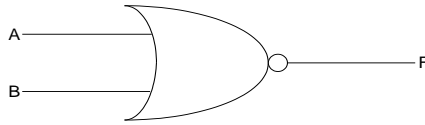
Gerbang NAND merupakan kombinasi dari gerbang AND dengan gerbang NOT dimana keluaran gerbang AND dihubungkan ke saluran masukan dari gerbang NOT. Karena keluaran dari gerbang AND di”NOT”kan maka prinsip kerja dari gerbang NAND merupakan kebalikan dari gerbang AND. Outputnya merupakan komplemen atau kebalikan dari gerbang AND, yakni memberikan keadaan level logic 0 pada outputnya jika dan hanya jika keadaan semua inputnya berlogika 1. Simbol gerbang logika NAND 2 input :



Gambar 4.7. Simbol Gerbang NAND

4.2.5. NOR Gate

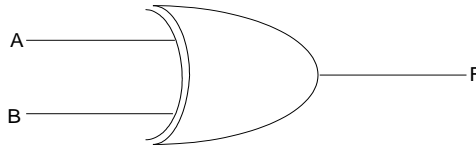
Sama halnya dengan NAND Gate, gerbang NOR merupakan kombinasi dari gerbang OR dengan gerbang NOT dimana keluaran gerbang OR dihubungkan ke saluran masukan dari gerbang NOT. Karena keluaran dari gerbang OR di”NOT”kan maka prinsip kerja dari gerbang NOR merupakan kebalikan dari gerbang OR. Outputnya merupakan komplemen atau kebalikan dari gerbang OR, yakni memberikan keadaan level logic 0 pada outputnya jika salah satu atau lebih inputnya berlogika 1. Simbol gerbang logika NOR 2 input :



Gambar 4.8. Simbol Gerbang NOR

4.2.6. EX-OR Gate

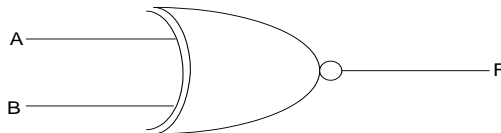
EX-OR singkatan dari Exclusive OR dimana jika input berlogic sama maka output akan berlogic 0 dan sebaliknya jika input berlogic beda maka output akan berlogic 1. Simbol gerbang logika EX-OR 2 input :



Gambar 4.9. Simbol Gerbang EXOR

4.2.7. EX-NOR

EX-NOR gate adalah kebalikan dari EX-OR gate dimana jika input berlogic sama maka output akan berlogic 1 dan sebaliknya jika input berlogic beda maka output akan berlogic 0. Simbol gerbang logika EX-NOR 2 input :



Gambar 4.10. Simbol Gerbang EX-NOR

4.3. Teorema-teorema Aljabar Boole

Aljabar Boole sangat penting peranannya di dalam proses perancangan maupun analisis rangkaian logika. Untuk memperoleh hasil rancangan yang berupa persamaan logika yang siap diimplementasikan, diperlukan tahap pemberlakuan kaidah-kaidah perancangan. Salah satu alat yang diperlukan di dalam tahap pemberlakuan kaidah-kaidah perancangan adalah aljabar Boole. Demikian juga untuk proses analisis rangkaian logika, aljabar Boole juga diperlukan sebagai alat untuk mengevaluasi output rangkaian

logika. Terdapat dua jenis teorema dalam aljabar Boole yaitu teorema variabel tunggal dan teorema variabel jamak. Masing-masing teorema tersebut memiliki teorema rangkapnya.

4.3.1. Teorema Variabel Tunggal

Teorema variabel tunggal aljabar Boole diturunkan dari operasi logika dasar OR, AND dan NOT. Teorema pada operasi AND dapat diperoleh melalui teorema pada operasi OR dan sebaliknya. Untuk memperoleh suatu teorema dari teorema yang diketahui, dapat dilakukan dengan cara :

- a. mengubah tanda + menjadi dot (.) atau sebaliknya
- b. mengubah 1 menjadi 0 atau sebaliknya

Berdasarkan dengan cara tersebut diatas, dapat diambil contoh misalnya terdapat teorema (1) : $A+1 = 1$, maka menghasilkan teorema (2) : $A.0 = 0$.

Dari penjelasan diatas dapat dituliskan teorema-teorema aljabar Boole untuk variabel tunggal seperti tersaji pada tabel 4.7 berikut ini.

Tabel 4.7. Teorema-teorema Aljabar Boole untuk variabel tunggal

| Teorema | Ekspresi | Sifat Rangkap |
|--------------|--|---------------------------------|
| Satu dan Nol | Teorema (1): $A+1=1$ | Teorema (2): $A.0=0$ |
| Identitas | Teorema (3): $A+0=A$ | Teorema (4): $A.1=A$ |
| Idempoten | Teorema (5): $A+A=A$ | Teorema (6): $A.A=A$ |
| Komplemen | Teorema (7): $A+\overline{A}=1$ | Teorema (8): $A.\overline{A}=0$ |
| Involusi | Teorema (9): $\overline{\overline{A}} = A$ | |

4.3.2. Teorema Variabel Jamak

Teorema-teorema variabel jamak aljabar Boole umumnya sama dengan teorema-teorema pada aljabar biasa. Seperti pada aljabar biasa, pada aljabar Boole juga terdapat teori komutatif, asosiatif, dan distributif. Tabel 4.8 berikut menunjukkan teorema-teorema variabel jamak pada aljabar Boole.

Tabel 4.8. Teorema-teorema Aljabar Boole untuk variabel jamak

| Teorema | Ekspresi | Sifat Rangkap |
|---------|----------|---------------|
|---------|----------|---------------|

| | | |
|-------------|---|---|
| Komutatif | Teorema (10): $A+B=B+A$ | Teorema (11): $AB=BA$ |
| Asosiatif | Teorema (12): $A+(B+C)=(A+B)+C$ | Teorema (13): $A(BC)=(AB)C$ |
| Distributif | Teorema (14): $A+BC=(A+B)(A+C)$ | Teorema (15): $A(B+C)=AB+AC$ |
| Absorpsi | Teorema (16): $A+AB=A$ Teorema (18): $A+\overline{A}B=A+B$ | Teorema (17): $A(A+B)=A$ Teorema (19): $A\overline{(A+B)}=\overline{A}B$ |
| De Morgan | Teorema (20): $\overline{A+B+\dots} = \overline{A} \cdot \overline{B} \cdot \dots$ | Teorema (21): $\overline{\overline{A} \cdot \overline{B} \cdot \dots} = A + B + \dots$ |

Rangkuman

1. Tabel kebenaran merupakan tabel yang menunjukkan pengaruh pemberian level logika pada input suatu rangkaian logika terhadap keadaan level logika outputnya. Melalui tabel kebenaran dapat diketahui watak atau karakteristik suatu rangkaian logika.
2. Ada 3 gerbang logika dasar, yaitu gerbang AND, gerbang OR dan gerbang NOT.
3. Gerbang AND (AND gate) didefinisikan sebagai gerbang logika yang memberikan keadaan level logika 1 (tinggi) pada outputnya, jika dan hanya jika semua keadaan inputnya berlevel logika 1 (tinggi).
4. Gerbang OR (OR gate) didefinisikan sebagai gerbang logika yang memberikan keadaan level logika 1 (tinggi) pada outputnya, keadaan salah satu atau lebih inputnya berlogika 1 (tinggi).
5. Gerbang NOT (NOT gate) merupakan gerbang logika yang memberikan keadaan level logika 1 (tinggi) pada outputnya, jika keadaan inputnya berlogika 1 (rendah) atau sebaliknya gerbang ini akan memberikan keadaan level logika 0 (rendah) pada outputnya jika keadaan inputnya berlevel 1 (tinggi).
6. Aljabar Boole sangat penting peranannya di dalam proses perancangan maupun analisis rangkaian logika. Terdapat dua jenis teorema dalam aljabar Boole yaitu teorema variabel tunggal dan teorema variabel jamak.

Tes Formatif

1. Susun tabel kebenaran rangkaian logika dengan tiga input yakni A(MSB), B, dan C(LSB) dan satu output yakni Y yang memberikan keadaan output bernilai tinggi jika:
 - a. Input A dan input B berbeda
 - b. Input A dan C sama
 - c. Input B dan C berbeda
2. Implementasikan persamaan berikut ini ke dalam bentuk rangkaian logika !
 - a. $X = (A+B)(A+\overline{C})\overline{B}$
 - b. $Y = \overline{ABC} + \overline{ABC} + \overline{BC}$
 - c. $\overline{\overline{Z}} = (A+D).(B+C)$

SATUAN BAHASAN 5

RANGKAIAN LOGIKA KOMBINASI

1. Gambaran Singkat Mengenai Materi Kuliah

Materi kuliah dari satuan bahasan yang kelima ini berisikan tentang pengertian logika kombinasi, bentuk-bentuk persamaan logika (SOP dan POS), dan bagaimana menyederhanakan suatu persamaan logika secara aljabar dengan menggunakan beberapa metode.

2. Pedoman Mempelajari Materi

Baca dengan cermat dan teliti materi pada satuan bahasan 5, kemudian pahami setiap teori yang akan membantu pemahaman anda tentang organisasi dan arsitektur komputer itu sendiri. Pada akhir satuan bahasan terdapat tes formatif yang perlu anda kerjakan sebagai sarana untuk mengukur seberapa jauh tingkat pemahaman anda terhadap materi ini.

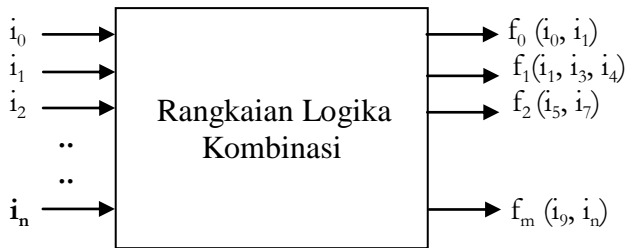
3. Tujuan Pembelajaran

Setelah mempelajari satuan bahasan 5 ini, mahasiswa diharapkan dapat :

- menjelaskan pengertian logika kombinasi.
- mengetahui cara mengubah suatu persamaan logika dalam bentuk *Sum of Product* dan *Product of Sum*.
- mengubah fungsi persamaan logika bentuk tidak standar menjadi bentuk standar
- melakukan penyederhanaan persamaan logika secara aljabar menggunakan metode Aljabar Boolean, Diagram Venn, Karnaugh Map, dan Quinne-Mc.Cluskey

5.1. Pengertian Logika Kombinasi

Logika kombinasi merupakan rangkaian logika yang outputnya hanya tergantung pada kombinasi input-inputnya saja, dan tidak tergantung pada keadaan output sebelumnya. Karena kondisi tersebut diatas, logika kombinasi disebut juga rangkaian logika yang outputnya tidak tergantung pada waktu.



Gambar 5.1. Diagram blok rangkaian logika kombinasi

Pada gambar diatas diperlihatkan output rangkaian logika kombinasi merupakan fungsi langsung dari input-inputnya. Output f_0 merupakan fungsi dari input i_0 dan i_1 , dalam hal ini f_0 hanya dipengaruhi oleh kombinasi dari i_0 dan i_1 saja. Demikian pula dengan output-output yang lain.

Begitu pula halnya manusia dalam melakukan berbagai aktifitasnya, tentunya dalam bertindak lebih menekankan kepada logikanya, sehingga akan menghasilkan output berupa hasil pekerjaannya atau hasil aktivitasnya apakah memberi manfaat atau tidak. Sebagaimana dijelaskan dalam QS.Al-Baqarah/2:32.

قَالُوا سُبْحَانَكَ لَا عِلْمَ لَنَا إِلَّا مَا عَلَّمْتَنَا إِنَّكَ أَنْتَ الْعَلِيمُ الْحَكِيمُ ﴿٣٢﴾

Terjemahnya : Mereka menjawab: "Maha suci Engkau, tidak ada yang Kami ketahui selain dari apa yang telah Engkau ajarkan kepada kami; Sesungguhnya Engkau lah yang Maha mengetahui lagi Maha Bijaksana." (Departemen Agama, 2005)

5.2. Bentuk-bentuk Persamaan Logika

Selain menggunakan simbol elemen logika, deskripsi rangkaian logika kombinasi dapat dilakukan dengan menggunakan persamaan logika yang telah dipelajari sebelumnya. Secara umum persamaan logika diklasifikasikan ke dalam dua bentuk yakni *Sum Of Product (SOP)* dan *Product Of Sum (POS)*. Dari masing-masing bentuk persamaan tersebut dapat diklasifikasikan lagi menjadi bentuk standar dan bentuk tak standar.

5.2.1. Bentuk *Sum Of Product (SOP)*

SOP merupakan persamaan logika yang mengekspresikan operasi OR dari suku-suku berbentuk operasi AND. Secara sederhana dapat dikatakan bahwa SOP adalah bentuk persamaan yang melakukan operasi OR terhadap AND. Bentuk SOP ini terdiri dari 2 macam, yaitu SOP standar dan SOP tidak standar. SOP standar adalah persamaan logika SOP yang setiap sukunya mengandung semua variabel input yang ada, sedangkan SOP tidak standar merupakan persamaan logika SOP yang tidak setiap sukunya mengandung semua variabel input. Pada bentuk SOP standar, setiap sukunya dinamakan *minterm*, disingkat dengan m (huruf kecil). *Minterm* bersifat unik, yakni untuk semua kombinasi input yang ada hanya terdapat satu kombinasi saja yang menyebabkan suatu *minterm* bernilai 1. Dengan kata lain, suatu persamaan logika dalam bentuk SOP, dapat dilihat dari outputnya yang berlogik 1. Tanda sigma (Σ) digunakan sebagai pengganti operator-operator penjumlahan (operasi logika OR).

Contoh :

$$X = \bar{A} \bar{B} C + A \bar{B} \bar{C} + A B \bar{C} + A B C \quad \text{pers.(1)}$$

$$Y = A \bar{B} + \bar{A} B \quad \text{pers.(2)}$$

$$P = A \bar{B} C + \bar{A} B + B \bar{C} + \bar{A} \quad \text{pers.(3)}$$

$$Q = \bar{A} B + \bar{B} C + \bar{A} \bar{C} \quad \text{pers.(4)}$$

Persamaan (1) dan (2) merupakan contoh SOP bentuk standar, sedangkan persamaan (3) dan (4) merupakan contoh SOP bentuk tak standar. SOP standar adalah persamaan logika SOP yang setiap sukunya mengandung semua variabel input yang ada, sedangkan SOP tak standar merupakan persamaan logika SOP yang tidak setiap sukunya mengandung semua variabel yang ada.

Pada bentuk SOP standar, setiap sukunya dinamakan *minterm*, disingkat dengan m (huruf kecil). Perhatikan fungsi X berikut :

$$X = \underbrace{\bar{A}\bar{B}C}_{\text{I}} + \underbrace{\bar{A}B\bar{C}}_{\text{II}} + \underbrace{A\bar{B}\bar{C}}_{\text{III}} + \underbrace{ABC}_{\text{IV}}$$

minterm

Minterm bersifat unik, yakni untuk semua kombinasi input yang ada hanya terdapat satu kombinasi saja yang menyebabkan suatu minterm bernilai 1.

Contoh :

Untuk $A = 0$, $B = 0$, $C = 1$, nilai fungsi X adalah :

$$\begin{aligned} X &= \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC \\ &= \bar{0}\bar{0}.1 + \bar{0}.0\bar{1} + 0.0\bar{1} + 0.0.1 \\ &= 1.1.1 + 0.1.0 + 0.0.0 + 0.0.1 \\ &= 1 + 0 + 0 + 0 = 1 \end{aligned}$$

Untuk $A = 1$, $B = 0$, $C = 0$, nilai fungsi X adalah :

$$\begin{aligned} X &= \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC \\ &= \bar{1}\bar{0}.0 + \bar{1}.0\bar{0} + 1.0\bar{0} + 1.0.0 \\ &= 0.1.0 + 1.1.1 + 1.0.1 + 1.0.0 \\ &= 0 + 1 + 0 + 0 = 1 \end{aligned}$$

Untuk $A = 1$, $B = 1$, $C = 0$, nilai fungsi X adalah :

$$\begin{aligned} X &= \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC \\ &= \bar{1}\bar{1}.0 + \bar{1}.1\bar{0} + 1.1\bar{0} + 1.1.0 \\ &= 0.0.0 + 1.0.1 + 1.1.1 + 1.1.0 \\ &= 0 + 0 + 1 + 0 = 1 \end{aligned}$$

Untuk $A = 1$, $B = 1$, $C = 1$, nilai fungsi X adalah :

$$\begin{aligned}
 X &= \bar{A} \bar{B} C + A \bar{B} \bar{C} + A B \bar{C} + A B C \\
 &= \bar{1} \bar{1} 1 + 1 \bar{1} \bar{1} + 1 1 \bar{1} + 1 1 1 \\
 &= 0.0.1 + 1.0.0 + 1.1.0 + 1.1.1 \\
 &= 0 + 0 + 0 + 1 = 1
 \end{aligned}$$

Oleh karena pada fungsi X diatas terdapat empat buah *minterm*, maka akan terdapat empat input yang akan memberikan nilai 1 pada fungsi X, yaitu pertama (A=0, B=0, C=1), kedua (A=1, B=0, C=0), ketiga (A=1, B=1, C=0), dan keempat (A=1, B=1, C=1). Selain input-input tersebut fungsi X bernilai 0.

Perhatikan bahwa input pertama bernilai desimal 1, input kedua bernilai desimal 4, input ketiga bernilai desimal 6, dan input keempat bernilai desimal 7. Oleh karena itu dapat dikatakan bahwa fungsi X mengandung minterm m1, m4, m6 dan m7. Jadi fungsi X dapat ditulis :

$$X = \underbrace{\bar{A} \bar{B} C}_{m1} + \underbrace{A \bar{B} \bar{C}}_{m4} + \underbrace{A B \bar{C}}_{m6} + \underbrace{A B C}_{m7} \rightarrow \text{cara penulisan I}$$

$$X = m1 + m4 + m6 + m7$$

$$X(A,B,C) = \sum m(1,4,6,7) \rightarrow \text{cara penulisan II}$$

Tanda \sum merupakan symbol persamaan bentuk SOP, sehingga cara penulisan II dapat dibaca X adalah fungsi dari input A, B dan C, yang merupakan persamaan logika berbentuk SOP standar terdiri atas minterm m1, m4, m6, dan m7. Dengan demikian tabel kebenaran untuk fungsi X adalah :

Tabel 5.1. Tabel kebenaran fungsi $X = \bar{A} \bar{B} C + A \bar{B} \bar{C} + A B \bar{C} + A B C$

| INPUT | | | OUTPUT |
|-------|---|---|--------|
| A | B | C | X |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |

$m_1 = \bar{A} \bar{B} C$
 $m_4 = A \bar{B} \bar{C}$
 $m_6 = A B \bar{C}$
 $m_7 = A B C$

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 0 | 1 | → |
| 1 | 1 | 1 | 1 | → |

5.2.2. Bentuk *Product Of Sum (POS)*

POS merupakan suatu persamaan logika yang mengekspresikan operasi AND dari suku-suku berbentuk operasi OR atau dengan kata lain POS adalah bentuk persamaan yang meakukan operasi AND terhadap OR. Bentuk POS ini terdiri dari 2 macam, yaitu POS standar dan POS tidak standar. POS standar adalah persamaan logika POS yang setiap sukunya mengandung semua variabel input yang ada, sedangkan POS standar merupakan persamaan logika POS yang tidak setiap sukunya mengandung semua variabel input. Pada bentuk POS standar, setiap sukunya dinamakan *maxterm*, disingkat dengan M (huruf besar). Sama halnya dengan *minterm*, *maxterm* juga bersifat unik, yakni untuk semua kombinasi input yang ada hanya terdapat satu kombinasi saja yang menyebabkan suatu *maxterm* bernilai 0. Dengan kata lain, suatu persamaan logika dalam bentuk POS, dapat dilihat dari outputnya yang berlogik 0. Tanda phi (Π) digunakan sebagai pengganti operator-operator perkalian (operasi logika AND).

Contoh :

$$R = (\bar{A} + \bar{B} + C) \cdot (A + B + \bar{C}) \cdot (A + \bar{B} + C) \cdot (\bar{A} + B + C) \quad \text{pers.(5)}$$

$$S = (A + B) \cdot (A + B) \quad \text{pers.(6)}$$

$$T = (\bar{A} + \bar{B} + C) \cdot (\bar{A} + B + C) \cdot (\bar{A} + C) \quad \text{pers.(7)}$$

$$U = (A + B) \cdot (B + \bar{C}) \cdot (\bar{A} + C) \quad \text{pers.(8)}$$

Persamaan (5) dan (6) merupakan contoh POS bentuk standar, sedangkan persamaan (7) dan (8) merupakan contoh POS bentuk tak standar. POS standar adalah persamaan logika POS yang setiap sukunya mengandung semua variabel input yang ada, sedangkan POS tak standar merupakan persamaan logika POS yang tidak setiap sukunya mengandung semua variabel yang ada.

Pada bentuk POS standar, setiap sukunya dinamakan Maxterm, disingkat dengan M (huruf besar). Perhatikan fungsi R berikut :

$$R = \underbrace{(\bar{A} + \bar{B} + C)}_I \cdot \underbrace{(A + \bar{B} + C)}_II \cdot \underbrace{(\bar{A} + B + C)}_III \cdot \underbrace{(A + B + C)}_IV$$

Rangkaian Logika Kombinasi



Maxterm bersifat unik, yakni untuk semua kombinasi input yang ada hanya terdapat satu kombinasi saja yang menyebabkan suatu maxterm bernilai 0.

Contoh :

Untuk $A = 1$, $B = 1$, $C = 0$, nilai fungsi R adalah :

$$\begin{aligned} R &= (\bar{A} + \bar{B} + C) (A + B + \bar{C}) (A + \bar{B} + C) (\bar{A} + B + C) \\ &= (\bar{1} + \bar{1} + 0) (1 + 1 + \bar{0}) (1 + \bar{1} + 0) (\bar{1} + 1 + 0) \\ &= (0 + 0 + 0) (1 + 1 + 1) (1 + 0 + 0) (0 + 1 + 0) \\ &= 0 \cdot 1 \cdot 1 \cdot 1 = 0 \end{aligned}$$

Untuk $A = 0$, $B = 0$, $C = 1$, nilai fungsi X adalah :

$$\begin{aligned} R &= (\bar{A} + \bar{B} + C) (A + B + \bar{C}) (A + \bar{B} + C) (\bar{A} + B + C) \\ &= (\bar{0} + \bar{0} + 1) (0 + 0 + \bar{1}) (0 + \bar{0} + 1) (\bar{0} + 0 + 1) \\ &= (1 + 1 + 1) (0 + 0 + 0) (0 + 1 + 1) (1 + 0 + 1) \\ &= 1 \cdot 0 \cdot 1 \cdot 1 = 0 \end{aligned}$$

Untuk $A = 0$, $B = 1$, $C = 0$, nilai fungsi X adalah :

$$\begin{aligned} R &= (\bar{A} + \bar{B} + C) (A + B + \bar{C}) (A + \bar{B} + C) (\bar{A} + B + C) \\ &= (\bar{0} + \bar{1} + 0) (0 + 1 + \bar{0}) (0 + \bar{1} + 0) (\bar{0} + 1 + 0) \\ &= (1 + 0 + 0) (0 + 1 + 1) (0 + 0 + 0) (1 + 1 + 0) \\ &= 1 \cdot 1 \cdot 0 \cdot 1 = 0 \end{aligned}$$

Untuk $A = 1$, $B = 0$, $C = 0$, nilai fungsi X adalah :

$$\begin{aligned} R &= (\bar{A} + \bar{B} + C) (A + B + \bar{C}) (A + \bar{B} + C) (\bar{A} + B + C) \\ &= (\bar{1} + \bar{0} + 0) (1 + 0 + \bar{0}) (1 + \bar{1} + 0) (\bar{1} + 1 + 0) \\ &= (0 + 1 + 0) (1 + 0 + 1) (1 + 1 + 0) (0 + 0 + 0) \\ &= 1 \cdot 1 \cdot 1 \cdot 0 = 0 \end{aligned}$$

Oleh karena pada fungsi R diatas terdapat empat buah *Maxterm*, maka akan terdapat empat input yang akan memberikan nilai 0 pada fungsi R, yaitu pertama (A=1, B=1, C=0), kedua (A=0, B=0, C=1), ketiga (A=0, B=1, C=0), dan keempat (A=1, B=0, C=0). Selain input-input tersebut fungsi R bernilai 1.

Perhatikan bahwa input pertama bernilai desimal 6, input kedua bernilai desimal 1, input ketiga bernilai desimal 2, dan input keempat bernilai desimal 4. Oleh karena itu dapat dikatakan bahwa fungsi R mengandung Maxterm M6, M1, M2 dan M4. Jadi fungsi R dapat ditulis :

$$R = \underbrace{(\bar{A}+\bar{B}+C)}_{M6} \underbrace{(A+B+\bar{C})}_{M1} \underbrace{(\bar{A}+B+\bar{C})}_{M2} \underbrace{(A+B+C)}_{M4} \rightarrow \text{cara penulisan I}$$

$$R = M1 + M2 + M4 + M6 \quad \text{atau}$$

$$R = M6 + M1 + M2 + M4$$

$$R(A,B,C) = \prod M(1,2,4,6) \rightarrow \text{cara penulisan II}$$

Tanda \prod merupakan symbol persamaan bentuk POS, sehingga persamaan diatas dapat dibaca R adalah fungsi dari input A, B dan C, yang merupakan persamaan logika berbentuk POS standar terdiri atas maxterm M1, M2, M4, dan M6. Dengan demikian tabel kebenaran untuk fungsi R adalah :

Tabel 5.2. Tabel kebenaran fungsi R =

$$(\bar{A}+\bar{B}+C) (A+B+\bar{C}) (A+\bar{B}+C) (\bar{A}+B+C)$$

| INPUT | | | OUTPUT | |
|-------|---|---|--------|-----------------------------|
| A | B | C | X | |
| 0 | 0 | 0 | 1 | |
| 0 | 0 | 1 | 0 | → $M_1 = A+B+\bar{C}$ |
| 0 | 1 | 0 | 0 | |
| 0 | 1 | 1 | 1 | → $M_2 = A+B+C$ |
| 1 | 0 | 0 | 0 | |
| 1 | 0 | 1 | 1 | → $M_4 = \bar{A}+B+C$ |
| 1 | 1 | 0 | 0 | |
| 1 | 1 | 1 | 1 | → $M_6 = \bar{A}+\bar{B}+C$ |

5.2.3. Mengubah Fungsi Bentuk Tak Standar Menjadi Bentuk Standar

Jika suatu persamaan logika bentuk SOP tak standar ingin diubah menjadi bentuk SOP standar, maka dapat dilakukan dengan cara seperti contoh berikut ini :

Ubahlah fungsi berikut menjadi bentuk standar :

a. $Y = A \bar{B} \bar{C} + \bar{B} C$

b. $Y = A B + \bar{C}$

Jawab :

a. $Y = A \bar{B} \bar{C} + \bar{B} C$

$$Y = A \bar{B} \bar{C} + \bar{B} C (A + \bar{A}) \text{ Ingat : } A + \bar{A} = 1$$

$$Y = A \bar{B} \bar{C} + A \bar{B} C + \bar{A} \bar{B} C$$

$$Y(A,B,C) = \Sigma m (1,4,5)$$

b. $Y = A B + \bar{C}$

$$Y = A B (C + \bar{C}) + \bar{C} (B + \bar{B}) (A + \bar{A})$$

$$Y = A B C + A B \bar{C} + (\bar{B}C + \bar{B}\bar{C}) (A + \bar{A})$$

$$Y = A B C + A B \bar{C} + A \bar{B} C + \bar{A} B C + A \bar{B} \bar{C} + \bar{A} \bar{B} \bar{C}$$

$$Y = A B C + A B \bar{C} + \bar{A} B C + A \bar{B} \bar{C} + \bar{A} \bar{B} \bar{C}$$

$$Y(A,B,C) = \Sigma m (7,6,2,4,0)$$

5.3. Penyederhanaan Secara Aljabar

Bentuk suatu persamaan logika baik dalam bentuk SOP maupun POS yang diperoleh dari tabel kebenaran umumnya jika diimplementasikan ternyata merupakan bentuk implemementasi yang tidak efisien. Oleh karena itu, setiap persamaan logika yang akan diimplementasikan ke dalam bentuk rangkaian logika pada dasarnya

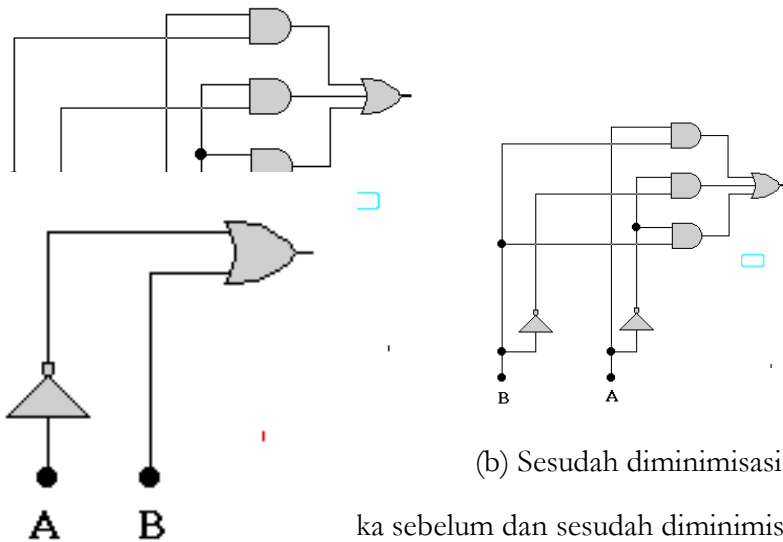
dapat dilakukan jika persamaan logika tersebut sudah dalam bentuk minimum, yaitu dengan tahap minimisasi. Tahap minimisasi merupakan suatu cara untuk memanipulasi atau menyederhanakan suatu persamaan logika dengan menggunakan teorema aljabar Boolean, diagram venn, karnaugh map, dan sebagainya.

Teknik minimisasi dalam ilmu digital adalah suatu teknik yang digunakan untuk menyederhanakan suatu persamaan logika. Mengapa suatu persamaan logika perlu disederhanakan?

Suatu persamaan logika perlu disederhanakan agar jika persamaan logika itu kita buat menjadi sebuah rangkaian logika kita bisa:

- ▶ Mengurangi jumlah komponen yang digunakan
- ▶ Mengurangi jumlah biaya yang diperlukan
- ▶ Mempersingkat waktu untuk merangkai
- ▶ Menghasilkan respon rangkaian lebih cepat karena delay rangkaian berkurang
- ▶ Memperkecil dimensi fisik rangkaian
- ▶ Menganalisa rangkaian dengan mudah

Berikut adalah contoh rangkaian yang belum diminimisasi dan rangkaian yang sudah diminimisasi.



ka sebelum dan sesudah diminimisasi

Bagaimanakah cara menyederhanakan persamaan logika? Berikut beberapa metode untuk menyederhanakan persamaan suatu logika diantaranya ;

- Aljabar Boolean
- Diagram Venn
- Karnaugh Map
- Quinne -Mc.Cluskey

1. Teorema Aljabar Boolean

Aljabar Boolean sangat penting peranannya di dalam proses perancangan maupun analisis rangkaian logika. Untuk memperoleh hasil rancangan yang berupa suatu persamaan logika yang siap diimplementasikan, diperlukan tahap pemberlakuan kaidah-kaidah perancangan. Salah satunya adalah aljabar Boolean. Aljabar Boolean merupakan aljabar yang diberlakukan pada variabel yang bersifat diskrit, dan oleh karena itu, aljabar ini cocok diberlakukan pada variabel yang ada pada rangkaian logika. Terdapat 2 jenis teorema aljabar Boolean yakni teorema variabel tunggal dan teorema variabel jamak. Setiap teorema baik yang bersifat tunggal maupun jamak selalu memiliki teorema rangkapnya.

a. Sifat Idempoten (sama)

$$- X + X = X$$

$$- X \bullet X = X$$

b. Sifat Absorpsi (menghilangkan)

$$- X + (X \bullet Y) = X$$

$$- X \bullet (X + Y) = X$$

c. Teorema Identitas

$$- X + Y = Y$$

$$- X \bullet Y = Y$$

$$(Jika X = Y)$$

d. Teorema Komplemen

$$- Jika X + Y = 1, atau$$

$$- Jika X \bullet Y = 0,$$

$$Maka X = \bar{Y}$$

e. Teorema Involution

$$- X = \overline{\overline{X}}$$

f. Teorema Van De Morgan

$$- \overline{X + Y + Z} = \overline{X} \bullet \overline{Y} \bullet \overline{Z}$$

$$- \overline{X \bullet Y \bullet Z} = \overline{X} + \overline{Y} + \overline{Z}$$

2. Postulate Huntington

a. Postulate 1

$$- X + 0 = X \rightarrow X \bullet 1 = X$$

$$- X \bullet 0 = 0 \rightarrow X + 1 = 1$$

b. Postulate 2

$$- X + Y = Y + X$$

$$- X \bullet Y = Y \bullet X$$

c. Postulate 3

$$- X \bullet (Y + Z) = (X \bullet Y) + (X \bullet Z)$$

$$- X + (Y \bullet Z) = (X + Y) \bullet (X + Z)$$

d. Postulate 4

$$- X + (Y + Z) = (X + Y) + Z$$

$$- X \bullet (Y \bullet Z) = (X \bullet Y) \bullet Z$$

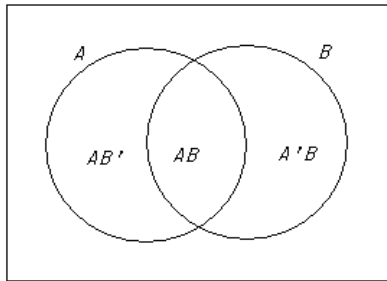
e. Postulate 5

$$- \overline{\overline{X}} + X = 1$$

$$- \overline{X} \bullet X = 0$$

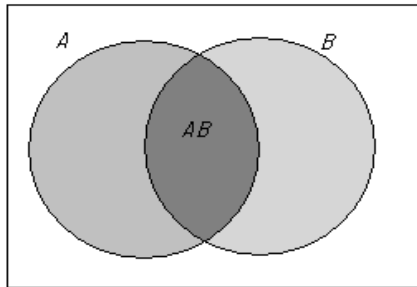
3. Diagram Venn

Salah satu cara untuk memudahkan untuk melukiskan hubungan antara variable dalam aljabar boolean adalah dengan menggunakan diagram venn. Diagram ini terdiri dari sebuah segi empat yang didalamnya dilukis lingkaran-lingkaran yang mewakili variabelnya, satu lingkaran untuk setiap variabelnya. Masing-masing lingkaran itu diberi nama menurut variable yang diwakilinya. Ditetapkan bahwa semua titik diluar lingkaran itu tidak dimiliki oleh variable tersebut. Misalnya lingkaran dengan nama A , jika dalam lingkaran itu dikatakan bernilai 1, maka diluar A dikatakan bernilai 0. Untuk dua lingkaran yang bertumpang tindih, terdapat empat daerah dalam segiempat tersebut.



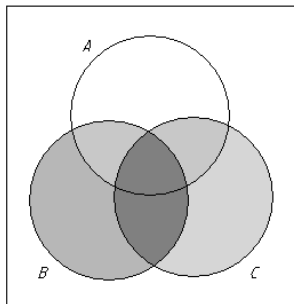
Gambar 5.3. Diagram Venn

Diagram venn dapat digunakan untuk melukiskan postulate aljabar boole atau untuk membuktikan berlakunya aljabar Boolean. Gambar berikut menunjukan bahwa daerah yang dimiliki oleh AB terletak dalam lingkaran A sehingga $A+AB = A$.



Gambar 5.4. Diagram Venn untuk irisan dua variabel

Gambar berikut menunjukan hukum distributive $A(B+C) = AB+AC$



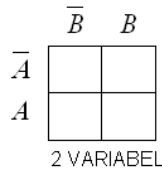
Gambar 5.5. Diagram Venn untuk irisan tiga variabel

Dalam lingkaran itu tampak tiga lingkaran yang bertumpang tindih, satu untuk masing-masing variable A, B dan C. dengan demikian dimungkinkan untuk membedakan delapan daerah yang terpisah dalam diagram venn dengan variable itu. Dalam hal ini hukum distributiv dibuktikan dengan menunjukan bahwa daerah yang memotong lingkaran A dengan daerah yang meliputi B atau C adalah daerah yang sama yang dimiliki oleh AB atau A .

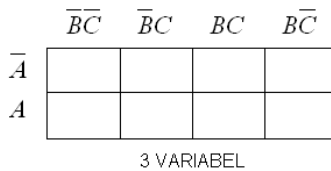
4. Karnaugh Map

Selain menggunakan teorema aljabar Boolean, agar suatu persamaan logika dengan cepat dapat diketahui sudah dalam bentuk minimum atau masih perlu diminimumkan dapat digunakan metode Karnaugh Map. Keuntungan yang diperoleh dari penyederhanaan persamaan logika dengan menggunakan K-map ditinjau dari persamaan akhir yang dihasilkan selalu merupakan persamaan yang tersederhana. Aturan penyederhanaan persamaan logika dengan K-map :

- a. Untuk persamaan logika yang terdiri dari n variable diperlukan K-map dengan 2^n kotak. Penomoran kotak berurutan berdasarkan kode gray.



Gambar 5.6. Karnaugh Map untuk 2 variabel input



Gambar 5.7. Karnaugh Map untuk 3 variabel input

Rangkaian Logika Kombinasi

| | | | | |
|----------------------------|----------------------------|-----------------|------|-----------------|
| | $\overline{C}\overline{D}$ | $\overline{C}D$ | CD | $C\overline{D}$ |
| $\overline{A}\overline{B}$ | | | | |
| $\overline{A}B$ | | | | |
| AB | | | | |
| $A\overline{B}$ | | | | |

4 VARIABEL

Gambar 5.8. Karnaugh Map untuk 4 variabel input

| | | | | | | | | |
|--|--|-----------------------------|-----------------------------|------------------|-----------------------------|------------------|------------------|-------|
| | $\overline{D}\overline{E}\overline{F}$ | $\overline{D}\overline{E}F$ | $\overline{D}E\overline{F}$ | $\overline{D}EF$ | $D\overline{E}\overline{F}$ | $D\overline{E}F$ | $DE\overline{F}$ | DEF |
| $\overline{A}\overline{B}\overline{C}$ | | | | | | | | |
| $\overline{A}\overline{B}C$ | | | | | | | | |
| $\overline{A}B\overline{C}$ | | | | | | | | |
| $\overline{A}BC$ | | | | | | | | |
| $AB\overline{C}$ | | | | | | | | |
| ABC | | | | | | | | |
| $A\overline{B}\overline{C}$ | | | | | | | | |
| $A\overline{B}C$ | | | | | | | | |
| $AB\overline{C}$ | | | | | | | | |

6 VARIABEL

Gambar 5.9. Karnaugh Map untuk 6 variabel input

b. Memasukkan data dari truth table ke dalam K-map

| A | B | C | D | F |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |

| | | | | |
|----------------------------|----------------------------|-----------------|------|-----------------|
| | $\overline{C}\overline{D}$ | $\overline{C}D$ | CD | $C\overline{D}$ |
| $\overline{A}\overline{B}$ | 0 | 0 | 1 | 0 |
| $\overline{A}B$ | 1 | 1 | 1 | 1 |
| AB | 1 | 1 | 0 | 0 |
| $A\overline{B}$ | 0 | 0 | 0 | 0 |

Gambar 5.10. Contoh input data dari truth table ke K-Map

- c. Penyederhanaan dilakukan dengan menggabungkan kotak-kotak yang bersebelahan dengan anggota sebanyak 2^m kotak dan formasi kotak membentuk segi empat ($0 \leq m \leq n$).
- d. Setiap kelompok dalam K-map akan membentuk satu suku dalam persamaan hasil penyederhanaan, dan jumlah variabel yang terkandung dalam suatu suku tergantung kepada jumlah kotak/daerah dalam suatu kelompok
- e. Dalam K-map dengan n variabel, suatu kelompok yang memiliki 2^m kotak merupakan suatu suku dengan $(n-m)$ variabel.
- f. Jumlah kelompok (group) dalam suatu K-map harus dibuat seminimal mungkin.
- g. Jumlah anggota (kotak) dalam suatu kelompok harus dibuat semaksimal mungkin

| | $\bar{C}\bar{D}$ | $\bar{C}D$ | CD | $C\bar{D}$ |
|------------------|------------------|------------|------|------------|
| $\bar{A}\bar{B}$ | 0 | 0 | 1 | 0 |
| $\bar{A}B$ | 1 | 1 | 1 | 1 |
| AB | 1 | 1 | 0 | 0 |
| $A\bar{B}$ | 0 | 0 | 0 | 0 |

Gambar 5.11. Input nilai logic 1 dalam K-map

- h. Proses pengelompokan dilakukan sampai seluruh kotak yang berlogik 1 tergabung dalam pengelompokan.

| | $\bar{C}\bar{D}$ | $\bar{C}D$ | CD | $C\bar{D}$ |
|------------------|------------------|------------|------|------------|
| $\bar{A}\bar{B}$ | 0 | 0 | 1 | 0 |
| $\bar{A}B$ | 1 | 1 | 1 | 1 |
| AB | 1 | 1 | 0 | 0 |
| $A\bar{B}$ | 0 | 0 | 0 | 0 |

$F = \bar{A}B + B\bar{C} + \bar{A}CD$

Gambar 5.12. K-map membentuk persamaan logika sederhana

5. Metoda Quine - Mc. Cluskey

Untuk menyederhanakan suatu persamaan logika empat variable, K-map memang metode yang paling efektif. Akan tetapi jika persamaan itu lebih dari empat variable metode ini akan mengalami kesulitan. Metode Quine Mc. Cluskey adalah salah satu cara yang memungkinkan untuk menyederhanakan suatu persamaan logika lebih dari empat variable.

Berikut langkah-langkahnya ;

Bila diberikan persamaan logika $F = \Sigma(0,3,7,8,9,13)$

a. Nyatakan masing-masing unsur minterm kedalam kode biner

$$0 = 0000$$

$$3 = 0011$$

$$7 = 0111$$

$$8 = 1000$$

$$9 = 1001$$

$$13 = 1011$$

b. Tentukan jumlah logik 1 dalam suatu angka biner sebagai indeks dari angka. Kumpulkan semua angka biner yang berindeks sama menjadi satu kelompok pada tabel.

$$0 = 0000 \rightarrow \text{jumlah logik 1} = 0$$

$$3 = 0011 \rightarrow \text{jumlah logik 1} = 2$$

$$7 = 0111 \rightarrow \text{jumlah logik 1} = 3$$

$$8 = 1000 \rightarrow \text{jumlah logik 1} = 1$$

$$9 = 1001 \rightarrow \text{jumlah logik 1} = 2$$

$$13 = 1011 \rightarrow \text{jumlah logik 1} = 3$$

Tabel 5.3. Kelompok jumlah logika 1

| | ABCD |
|---|---------|
| 0 | 0 |
| 1 | 8 |
| 2 | 3 9 |
| 3 | 7 13 |

c. Bandingkan antara tiap unsur mulai dari indeks terkecil dengan tiap unsur dari indeks sesudahnya. Nilai unsur dari indeks pertama harus lebih kecil dari nilai unsur indeks sesudahnya. Apabila

terdapat selisih 2^n maka boleh digabung. Langkah ini akan menghasilkan kelompok baru.

- d. Lakukan kembali langkah c sampai tidak ada lagi selisih 2^n .
- e. Tiap kelompok diberi nama.

Tabel 5.4. Sinkronisasi tabel logika 1

| | ABCD |
|---|------|
| 0 | 0 |
| 1 | 8 |
| 2 | 3 |
| | 9 |
| 3 | 7 |
| | 13 |

| | | |
|------|---|---|
| 0,8 | 8 | a |
| 8,9 | 1 | b |
| 3,7 | 4 | c |
| 9,13 | 4 | d |

| | 0 | 3 | 7 | 8 | 9 | 13 |
|------|-----|-----|-----|-------|-------|-----|
| 0,8 | ✓ | | | ✓ | | |
| 8,9 | | | | ✓ | ✓ | |
| 3,7 | | ✓ | ✓ | | | |
| 9,13 | | | | | ✓ | ✓ |
| | (a) | (c) | (c) | (a,b) | (b,d) | (d) |

- f. Untuk penyelesaian, kita ambil satu nama yang mewakili tiap angka (a, b, c atau d). Pengambilan nama harus seminimal mungkin. Sehingga akan didapat:

$$\begin{aligned}
 F &= a + c + d \\
 &= 0000 + 0011 + 1001 \\
 &= 1000 + 0111 + 1011 \\
 &= -000 + 0-11 + 10-1 \\
 &= \overline{BCD} + \overline{ACD} + \overline{ABD}
 \end{aligned}$$

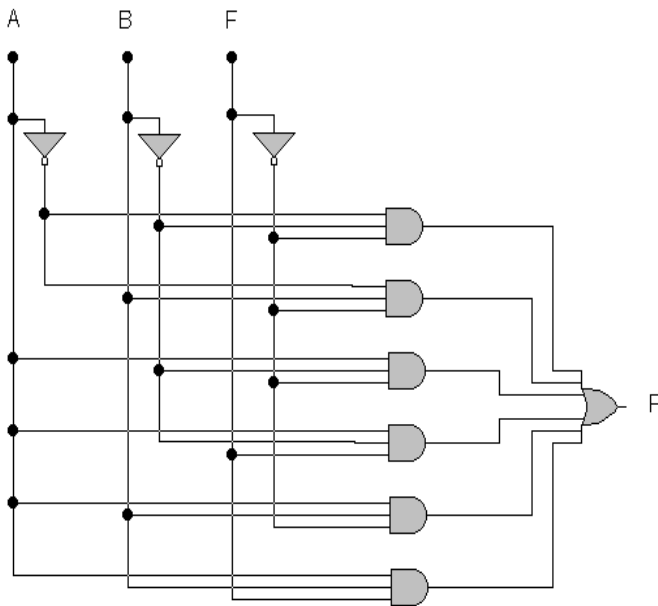
Sebagai contoh latihan, sederhanakan persamaan logika pada tabel kebenaran dibawah ini.

| A | B | C | F |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

$$F = \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + A\overline{B}\overline{C} + A\overline{B}C + A\overline{B}C + ABC$$

Jawab :

Maka rangkaian logikanya adalah



Persamaan diatas dapat disederhanakan dengan beberapa metode yang telah dijelaskan diatas.

- o Dengan aljabar Boolean

$$\begin{aligned}
 F &= \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + A\overline{B}\overline{C} + A\overline{B}C + AB\overline{C} + ABC \\
 &= \overline{A}\overline{C}(\overline{B} + B) + A(\overline{B}\overline{C} + \overline{B}C + B\overline{C} + BC) \\
 &= \overline{A}\overline{C}(1) + A(1) \\
 &= \overline{A}\overline{C} + A \\
 &= A + \overline{C}
 \end{aligned}$$

- o Dengan K-map

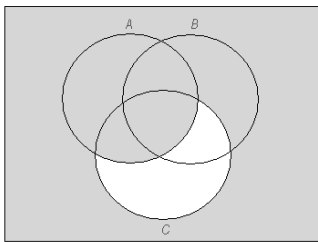
| A | B | C | F |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

| | $\bar{B}\bar{C}$ | $\bar{B}C$ | BC | $B\bar{C}$ |
|-----------|------------------|------------|------|------------|
| \bar{A} | 1 | 0 | 0 | 1 |
| A | 1 | 1 | 1 | 1 |

$$F = A + \bar{C}$$

- o Dengan diagram venn

$$F = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$$



Dari gambar diatas kita bisa lihat lingkaran A terisi oleh arsiran sedangkan lingkaran C tidak terisi oleh arsiran hanya sebagian yang terisi dan itu pun sudah terwakili oleh lingkaran A . Jadi $F = A + \bar{C}$.

- o Dengan Quine Mc-Cluskey

$$F = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$$

000 010 100 101 110 111 → Biner

0 2 4 5 6 7 → Desimal

Rangkaian Logika Kombinasi

| Jumlah logik 1 | ABCD |
|----------------|------|
| 0 | 0 |
| 1 | 2 |
| | 4 |
| 2 | 5 |
| | 6 |
| 3 | 7 |

| | |
|-----|---|
| 0,2 | 2 |
| 0,4 | 4 |
| 2,6 | 4 |
| 4,5 | 1 |
| 4,6 | 2 |
| 5,7 | 2 |
| 6,7 | 1 |

| | |
|---------|---|
| 0,2,4,6 | 4 |
| 0,4,2,6 | 2 |
| 4,5,6,7 | 2 |
| 4,6,5,7 | 1 |

| | |
|---------|---|
| 0,2,4,6 | a |
| 4,5,6,7 | b |

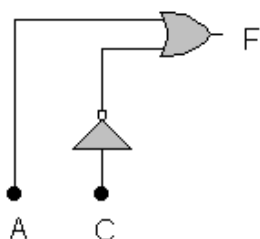
| 0 | 2 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|
| ✓ | ✓ | ✓ | | ✓ | |
| | | ✓ | ✓ | ✓ | ✓ |
| a | a | a | b | a | b |
| | | b | | b | |

$$F = a + b \rightarrow (0,2,4,6) + (4,5,6,7)$$

| ABC | ABC |
|---------------|------------|
| 000 | 100 |
| 010 | 101 |
| 100 | 110 |
| <u>110</u> | <u>111</u> |
| 0 = \bar{C} | 1 = A |

$$F = A + \bar{C}$$

Jadi penyederhanaan persamaan logika diatas dapat diimplementasikan dalam rangkaian sebagai berikut :



$$F = A + \bar{C}$$

Rangkuman

1. Logika kombinasi merupakan rangkaian logika yang outputnya hanya tergantung pada kombinasi input-inputnya saja, dan tidak tergantung pada keadaan output sebelumnya.
2. Secara umum persamaan logika diklasifikasikan ke dalam dua bentuk yakni *Sum Of Product (SOP)* dan *Product Of Sum (POS)*.
3. *Sum Of Product (SOP)* merupakan persamaan logika yang mengekspresikan operasi OR dari suku-suku berbentuk operasi AND. Sedangkan *Product Of Sum (POS)* merupakan persamaan logika yang mengekspresikan operasi AND dari suku-suku berbentuk operasi OR.
4. Teknik minimisasi dalam ilmu digital adalah suatu teknik yang digunakan untuk menyederhanakan suatu persamaan logika.
5. Tahap minimisasi merupakan suatu cara untuk memanipulasi atau menyederhanakan suatu persamaan logika dengan menggunakan teorema aljabar Boolean, diagram venn, karnaugh map, dan sebagainya.

Tes Formatif

1. Jelaskan pengertian *Sum Of Product*, *Product Of Sum*, *Minterm* dan *Maxterm* !
2. Tuliskan persamaan SOP standar an POS standar yang dihasilkan dari tabel kebenaran suatu rangkaian yang memberikan output 1 jika inputnya merupakan bilangan prima 3-bit !
3. Ubahlah bentuk persamaan logika berikut ini menjadi bentuk standar:
 - a. $X = A + BC$
 - b. $Y = A'B'C' + A'BC' + B'C'$
 - c. $Z = (A + B')(A + C)$
4. Sederhanakan persamaan $X = A'B'C + A'BC + ABC + AB'C' + AC$ dengan peta Karnaugh !

SATUAN BAHASAN 6

RANGKAIAN LOGIKA SEKUENSIAL

1. Gambaran Singkat Mengenai Materi Kuliah

Materi kuliah dari satuan bahasan yang pertama ini berisikan tentang prinsip kerja dari beberapa jenis rangkaian Flip Flop, rangkaian Encoder dan Decoder, serta rangkaian Multiplexer dan Demultiplexer.

2. Pedoman Mempelajari Materi

Baca dengan cermat dan teliti materi pada satuan bahasan 6, kemudian pahami setiap teori yang akan membantu pemahaman anda tentang organisasi dan arsitektur komputer itu sendiri. Pada akhir satuan bahasan terdapat tes formatif yang perlu anda kerjakan sebagai sarana untuk mengukur seberapa jauh tingkat pemahaman anda terhadap materi ini.

3. Tujuan Pembelajaran

Setelah mempelajari satuan bahasan 6 ini, mahasiswa diharapkan dapat :

- menjelaskan pengertian dan jenis-jenis Flip flop
- mengetahui prinsip kerja Flip flop RS, flip flop D, flip flop JK, dan flip flop T.
- menjelaskan perbedaan Encoder dan Decoder
- mengetahui penerapan Encoder dan Decoder
- menjelaskan perbedaan Multiplexer dan Demultiplexer
- mengetahui penerapan Multiplexer dan Demultiplexer
- dapat merancang Multiplexer dan Demultiplexer dalam suatu

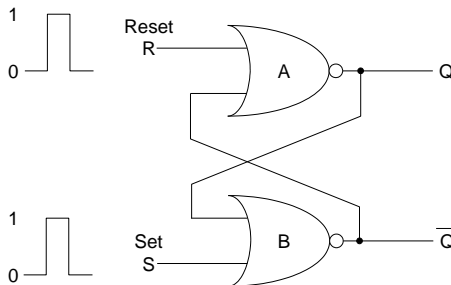
rangkaian

6.1. FLIP-FLOP

Rangkaian flip-flop dapat mempertahankan suatu keadaan biner dalam waktu yang tak terbatas sampai suatu sinyal masukan baru datang untuk mengubah keadaan itu. Perbedaan utama diantara berbagai jenis flip-flop itu adalah banyaknya masukan yang dimiliki dan perilaku bagaimana masukan itu mempengaruhi keadaan biner dalam flip-flop tersebut.

6.1.1 Rangkaian Flip-flop Dasar

Suatu rangkaian flip-flop dapat disusun dengan dua gerbang NOR atau dua gerbang NAND. Susunan itu ditunjukkan pada gambar 6.1 dan gambar 6.2. Masing-masing rangkaian itu membentuk suatu flip-flop dasar yang merupakan dasar pengembangan bagi jenis-jenis flip-flop yang lain. Hubungan silang dari salah satu gerbang ke masukan gerbang yang lain merupakan suatu jalur umpan-balik. Dengan alasan itu rangkaian tersebut dapat digolongkan kepada rangkaian urutan tak-serempak. Masing-masing flip-flop itu mempunyai dua keluaran, Q dan \bar{Q} dan dua masukan, set dan reset. Masukan set membuat flip-flop menjadi dalam keadaan set atau bernilai logik 1 pada keluaran normalnya (Q), dan masukan reset membuat flip-flop menjadi dalam keadaan bebas (*clear*) atau mempunyai nilai logik 0 pada keluaran normalnya. Jenis flip-flop ini sering dikenal sebagai flip-flop RS gandengan langsung (*direct coupled RS flip-flop*), R dan S merupakan huruf pertama nama masukannya.



Gambar 6.1. Rangkaian flip-flop dasar dengan gerbang NOR

Tabel 6.1 Tabel kebenaran flip-flop dasar dengan gerbang NOR

| S | R | Q | \bar{Q} |
|---|---|---|-----------|
| 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 |

(setelah S = 1, R = 0)
(setelah S = 0, R = 1)

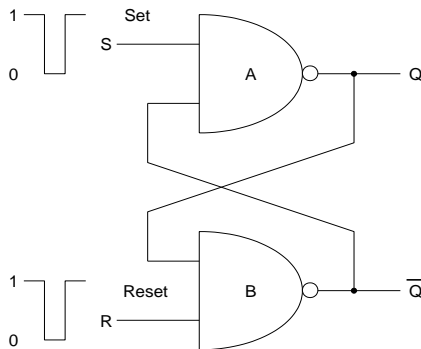
Untuk menganalisis rangkaian pada gambar 6.1, harus diingat bahwa keluaran suatu gerbang NOR adalah 0 jika salah satu masukannya sama dengan 1 dan keluaran gerbang NOR adalah 1 hanya jika semua masukannya sama dengan 0. Sebagai titik awal, misalnya masukan set adalah 1 dan masukan reset sama dengan 0. Karena gerbang B mempunyai sebuah masukan 1, keluaran Q' harus sama dengan 0 yang mengakibatkan kedua masukan ke gerbang A itu sama dengan 0 dan keluarannya Q sama dengan 1. Bila masukan set dikembalikan ke 0, keluarannya tetap sama. Hal itu adalah karena keluaran Q tetap 1 sehingga masih ada sebuah masukan 1 pada gerbang B, yang selanjutnya membuat keluaran Q' tetap 0. Akibatnya kedua masukan ke gerbang A sama dengan 0 dan keluaran Q tetap sama dengan 1. Dengan cara yang sama dapat dibuktikan bahwa suatu 1 pada masukan reset akan mengubah keluaran Q menjadi 0 dan Q' menjadi 1. Bila masukan reset itu dikembalikan ke 0, keluarannya tidak berubah.

Bila sebuah 1 diberikan bersama-sama ke masukan set dan reset, kedua keluarannya Q dan Q' menjadi 0. Dalam praktek keadaan semacam itu harus dihindari. Suatu flip-flop mempunyai dua keadaan stabil. Bila $Q = 1$ dan $Q' = 0$ dikatakan flip-flop itu dalam keadaan set (atau keadaan 1). Dan $Q = 0$ dan $Q' = 1$ merupakan keadaan bebas (atau keadaan 0). Keluaran Q dan Q' merupakan komplemen antara yang satu dengan yang lain dan dikatakan sebagai keluaran normal dan komplemen flip-flop tersebut. Keadaan biner suatu flip-flop diambil dari nilai keluaran normalnya.

Dalam operasi normal, kedua masukan suatu flip-flop akan tetap 0 kecuali bila keadaan flip-flop itu akan diubah. Pengenaan 1

sesaat ke masukan set menyebabkan flip-flop itu menjadi dalam keadaan set. Masukan set itu harus kembali ke 0 sebelum suatu 1 diberikan ke masukan resetnya. Pengenaan 1 sesaat ke masukan reset menyebabkan flip-flop tersebut menjadi dalam keadaan bebas kembali. Bila kedua masukannya itu mula-mula sama dengan 0, dan bila suatu 1 dikenakan ke masukan set sedangkan flip-flop itu dalam keadaan set atau bila sebuah 1 yang diberikan ke masukan reset sedangkan flip-flop itu dalam keadaan bebas, maka keadaan keluarannya tidak akan berubah. Bila sebuah 1 dikenakan sekaligus ke masukan set dan reset, kedua keluarannya akan sama dengan 0. Keadaan itu tidak terdefinisi dan biasanya dihindari. Jika kedua masukan itu menjadi 0 kembali, keadaan flip-flop menjadi tak tentu dan tergantung pada masukan mana yang menerima 1 lebih lama sebelum kembali ke 0.

Rangkaian flip-flop dasar NAND pada gambar 6.2 bekerja dengan kedua masukannya dalam keadaan normal sama dengan 1 kecuali bila keadaan flip-flop itu akan diubah. Pengenaan 0 sesaat ke masukan set menyebabkan keluaran Q menjadi 1 dan Q menjadi 0 membuat flip-flop menjadi dalam keadaan set. Setelah masukan set itu kembali ke 1, 0 sesaat pada masukan reset akan menyebabkan keadaan flip-flop menjadi bebas. Bila kedua masuka itu menjadi 0 bersama-sama, kedua keluaran pada flip-flop itu sama dengan 1, suatu keadaan yang harus dihindari dalam praktek.



Gambar 6.2. Rangkaian flip-flop dasar dengan gerbang NAND

Tabel 6.2 Tabel kebenaran flip-flop dasar dengan gerbang NAND

| S | R | Q | \bar{Q} |
|---|---|---|-----------|
| 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 |

(setelah S = 1, R = 0)

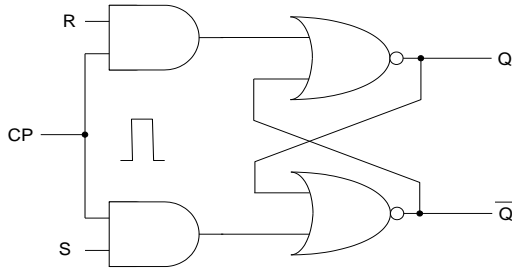
(setelah S = 0, R = 1)

A. Flip-flop RS

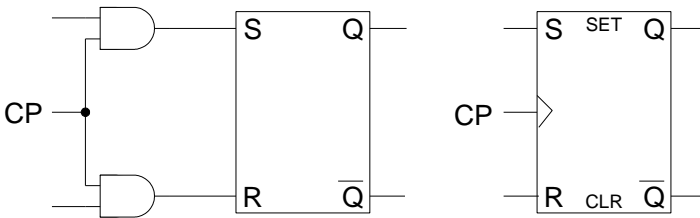
Suatu flip-flop dasar pada dasarnya adalah suatu rangkaian urutan tak serempak. Dengan menambah suatu gerbang ke setiap masukan rangkaian dasar itu, flip-flop tersebut dapat diubah untuk menanggapi masukan selama adanya suatu pulsa waktu. Flip-flop RS menurut waktu yang ditunjukkan pada gambar a itu terdiri dari flip-flop NOR dasar dengan dua gerbang AND. Keluaran kedua gerbang AND tersebut tetap 0 selama pulsa waktu (yang diberi lambang CP - *clock pulse*) sama dengan 0, tanpa memandang nilai masukan S dan R nya. Bila pulsa waktu itu menjadi 1, informasi dari masukan S dan R diijinkan untuk masuk ke flip-flop dasar tersebut. Keadaan set tercapai dengan S = 1, R = 0, dan CP = 1. Untuk mengubahnya menjadi keadaan bebas, masukan S harus 0, R = 1, dan CP = 1. Dengan masukan R dan S yang keduanya sama dengan 1, adanya pulsa waktu akan menyebabkan kedua keluaran flip-flop itu sesaat sama dengan 0. Bila pulsa waktu itu hilang, keadaannya menjadi tak tentu, dapat dalam keadaan set atau bebas, tergantung apakah masukan set atau reset yang lebih lama sama dengan 1 sebelum berubah menjadi 0 pada akhir pulsa waktu tersebut.

Tanggapan flip-flop menurut waktu merupakan praktek yang umum dijumpai dalam sistem digital karena perubahan dalam sistem itu umumnya diinginkan terjadi serentak menurut kendali sumber waktu. Oleh sebab itu, flip-flop menurut waktu disebut sebagai suatu rangkaian urutan serempak.

Dua lambang untuk flip-flop RS ditunjukkan pada gambar 6.3. Gerbang AND dengan masukan pulsa waktu dapat dilukis diluar lambang tersebut, atau suatu lambang dengan tanda CP digunakan untuk menunjukkan bahwa keluaran flip-flop tersebut tidak akan terpengaruh kecuali bila ada pulsa waktu pada masukan yang bertanda CP itu.



Gambar 6.3. Diagram logika



Gambar 6.4. Lambang tanpa dan dengan pulsa waktu

| SR | | Q | | | |
|----|---|----|----|----|----|
| | | 00 | 01 | 11 | 10 |
| Q | 0 | | | X | 1 |
| | 1 | 1 | | X | 1 |

$$Q(t + 1) = S + \bar{R}Q$$

$$SR = 0$$

Gambar 6.5 Persamaan karakteristik

Tabel 6.3. Tabel karakteristik flip-flop RS menurut waktu

| Q | S | R | Q(t + 1) |
|---|---|---|-----------|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | tak tentu |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | tak tentu |

Dalam praktek flip-flop menurut waktu itu seringkali diinginkan untuk membuat flip-flop tersebut dalam keadaan set atau bebas tanpa harus menunggu datangnya pulsa waktu. Untuk itu umumnya flip-flop menurut waktu selalu dilengkapi dengan masukan set atau reset langsung. Masukan langsung itu sering diberi label SET atau CLR (*clear*-bebas) untuk membedakannya dengan masukan S (set) dan R (reset) yang bekerja menurut waktu seperti yang ditunjukkan pada gambar 6.4.

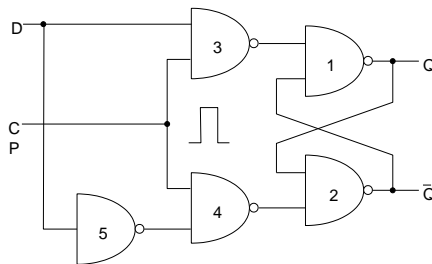
Pada awal penggunaan suatu flip-flop sering tidak dapat diramal perilakunya, dalam hal semacam itu masukan SET dan CLR berguna untuk mengawali operasi suatu sistem dengan keadaan flip-flop yang terdefinisi. Persamaan karakteristik flip-flop itu diturunkan dari gambar c. Persamaan itu memberikan nilai keadaan berikutnya sebagai fungsi keadaan sekarang dan masukan-masukannya. Persamaan karakteristik itu adalah pernyataan aljabar untuk informasi biner pada tabel karakteristiknya. Dua keadaan tak tentu pada flip-flop itu ditandai dengan x dalam peta itu karena dapat bernilai 1 atau 0. akan tetapi hubungan $SR = 0$ harus dimasukkan sebagai bagian persamaan karakteristik itu untuk menunjukkan bahwa S dan R tidak dapat sama dengan 1 secara serentak.

Tabel karakteristik flip-flop tersebut ditunjukkan pada tabel 6.3. Tabel itu merupakan ringkasan operasi flip-flop dalam bentuk tabel. Q adalah keadaan biner flip-flop pada suatu waktu yang diketahui (yang dikatakan sebagai keadaan sekarang), kolom R dan S memberikan nilai yang dapat terjadi untuk masukannya dan Q (t + 1) adalah keadaan flip-flop setelah timbulnya suatu pulsa waktu (dikatakan sebagai keadaan berikutnya).

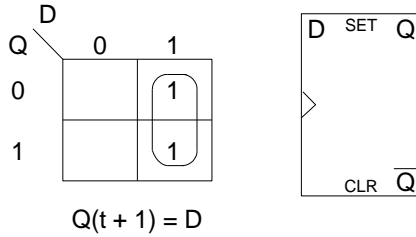
B. Flip-flop D

Flip-flop D yang ditunjukkan pada gambar a merupakan modifikasi flip-flop RS menurut waktu. Gerbang NAND 1 dan 2 membentuk suatu flip-flop dasar. Gerbang 3 dan 4 mengubahnya menjadi suatu flip-flop menurut waktu. Masukan D langsung diberikan ke masukan S dan komplementennya melalui gerbang 5, dikenakan ke masukan R. Selama masukan pulsa waktu 0, gerbang 3 dan 4 mempunyai nilai 1 pada keluarannya, tanpa memandang nilai masukannya yang lain. Hal itu sesuai dengan persyaratan bahwa kedua masukan flip-flop NAND dasar tersebut (gambar b) pada awalnya mempunyai nilai logika 1. Masukan D dicacah (*sampled*) selama adanya pulsa waktu. Jadi pada saat masukan D sama dengan 1, keluaran gerbang 3 menjadi 0 sehingga mengakibatkan flip-flop itu menjadi dalam keadaan set (kecuali bila flip-flop itu telah berada dalam keadaan set sebelumnya). Jika masukan D itu sama dengan 0, keluaran gerbang 4 menjadi 0 yang mengubah flip-flop tersebut menjadi dalam keadaan bebas.

Flip-flop D itu mendapatkan namanya karena kemampuannya memindahkan 'data' ke dalam flip-flop. Rangkaian flip-flop itu pada dasarnya adalah rangkaian flip-flop RS dengan sebuah pembalik di masukan R nya. Adanya penambahan pembalik itu mengurangi banyaknya masukan dari dua menjadi satu. Disamping itu karena keluaran Q tidak akan menerima masukan D sampai datangnya suatu pulsa waktu, bentuk itu sering juga disebut sebagai flip-flop tertunda (*delay flip-flop*).



Gambar 6.6. Diagram logika Flip Flop D



Gambar 6.7. Lambang dan persamaan karakteristik

Lambang untuk flip-flop D menurut waktu itu diberikan pada gambar b. Seperti halnya dengan setiap flip-flop menurut waktu, flip-flop D juga dilengkapi dengan masukan SET dan CLR. Persamaan karakteristiknya diturunkan dengan peta karnaugh di (c) dan tabel karakteristik flip-flop D itu diberikan oleh tabel 6.2. Persamaan karakteristik itu membuktikan bahwa keadaan berikutnya pada flip-flop tersebut sama seperti masukan D dan tidak tergantung pada nilai keadaan sekarang.

Tabel 6.4. Tabel karakteristik flip-flop D

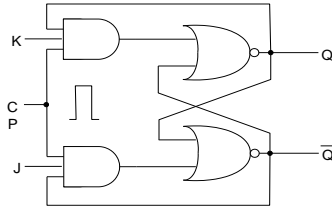
| Q | D | Q(t + 1) |
|---|---|----------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

C. Flip-flop JK

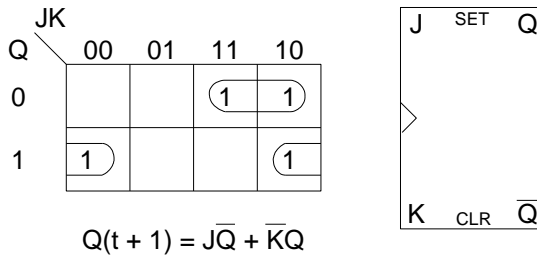
Flip-flop ini merupakan perbaikan dari flip-flop RS sehingga keadaan tak tentu pada jenis RS menjadi terdefinisi untuk jenis JK

tersebut. Masukan J dan K berlaku seperti masukan R dan S (perhatikan bahwa untuk suatu flip-flop JK, huruf J adalah untuk set dan huruf K untuk bebas). Bila masukan J dan K diberikan secara serentak, nilai flip-flop itu berubah menjadi komplementnya, yaitu jika mula-mula $Q = 1$, akan berubah menjadi $Q = 0$ dan sebaliknya.

Suatu flip-flop JK menurut waktu ditunjukkan pada gambar a. Keluaran diAND-kan dengan masukan K dan CP sehingga flip-flop itu dibebaskan selama suatu pulsa waktu hanya jika Q sebelumnya sama dengan 1.



Gambar 6.8. Diagram logika Flip Flop JK



Gambar 6.9. Lambang dan persamaan karakteristik

Demikian pula keluaran Q flip-flop tersebut diAND-kan dengan masukan J dan CP sehingga flip-flop itu dapat diset dengan pulsa waktu hanya jika Q sebelumnya sama dengan 1. bila baik J maupun K sama dengan 1, keadaan Q akan selalu berubah tanpa memandang bagaimana keadaan Q tersebut sebelum pulsa waktu diberikan. Jadi jika Q sama dengan 1, keluaran gerbang AND yang diatas menjadi 1 dan flip-flop itu dibebaskan. Tampak bahwa jika sinyal CP itu tetap 1 setelah keluarannya dikomplemenkan, flip-flop itu akan berubah menjadi suatu keadaan yang baru.

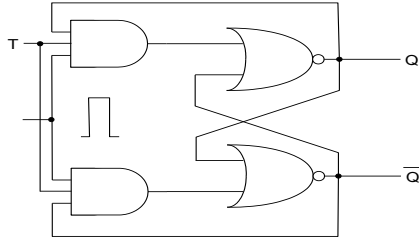
Lambang dan persamaan karakteristik flip-flop JK itu diberikan pada gambar b dan c. Tabel karakteristik flip-flop itu diberikan oleh tabel 6.5.

Tabel 6.5. Tabel karakteristik flip-flop JK

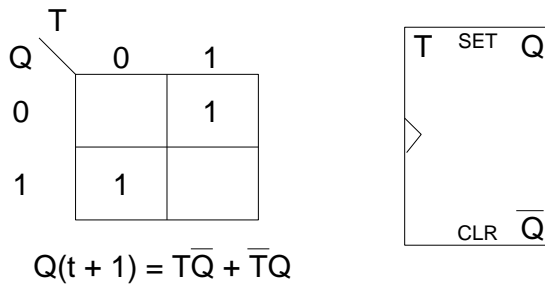
| Q | J | K | Q(t + 1) |
|---|---|---|----------|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

D. Flip-flop T

Flip-flop ini adalah flip-flop JK dengan masukan tunggal. Seperti yang tampak pada gambar a, flip-flop T itu didapatkan dari jenis JK jika kedua masukannya dijadikan satu. Nama "T" (*toggle*-artinya saklar pengalih dua keadaan) itu diperoleh karena kemampuan flip-flop itu untuk mengubah keadaannya. Apapun keadaan sekarang flip-flop T itu akan berubah menjadi komplementnya setiap kali pulsa waktu diberikan pada saat masukan T itu bernilai 1. Lambang dan persamaan karakteristik flip-flop T itu ditunjukkan pada gambar b dan c. Tabel karakteristiknya diberikan oleh tabel 1. Keempat jenis flip-flop yang diperkenalkan diatas dapat tersedia dalam keadaan tanpa pengaturan waktu. Flip-flop tanpa masukan waktu tersebut berguna untuk operasi tak-serempak. Keempat jenis itu merupakan jenis yang umum dijumpai dalam rangkaian digital dan tersedia di pasaran.



Gambar 6.10. Diagram logika Flip Flop T



Gambar 6.11. Lambang dan karakteristik

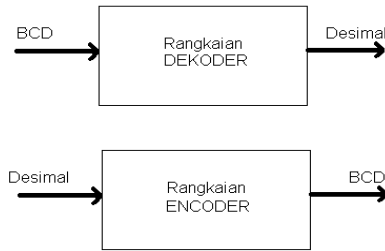
Tabel 6.6. Tabel karakteristik flip-flop T

| Q | T | Q(t + 1) |
|---|---|----------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

6.2. Encoder dan Decoder

Encoder adalah suatu rangkaian logika yang berfungsi untuk mengkonversikan kode yang lebih dikenal oleh manusia ke dalam kode yang kurang dikenal manusia. Decoder adalah suatu rangkaian logika yang berfungsi untuk mengkonversikan kode yang kurang dikenal manusia kedalam kode yang lebih dikenal manusia.

Contoh



Gambar 6.12. Encoder dan Decoder

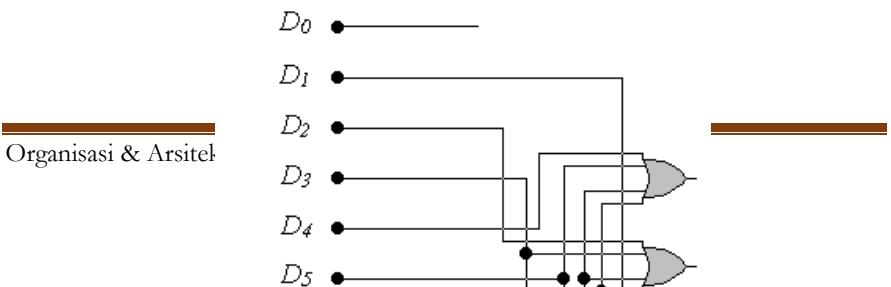
A. Encoder Oktal ke Biner

Encoder oktal ke biner ini terdiri dari delapan input, satu untuk masing-masing dari delapan angka itu, dan tiga output yang menghasilkan bilangan binernya yang sesuai. Rangkaian itu terdiri dari gerbang OR. Berikut tabel kebenarannya.

Tabel 6.7. Tabel Kebenaran Encoder Oktal ke Biner

| Input | | | | | | | | Output | | |
|-------|-------|-------|-------|-------|-------|-------|-------|--------|-----|-----|
| D_0 | D_1 | D_2 | D_3 | D_4 | D_5 | D_6 | D_7 | A | B | C |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

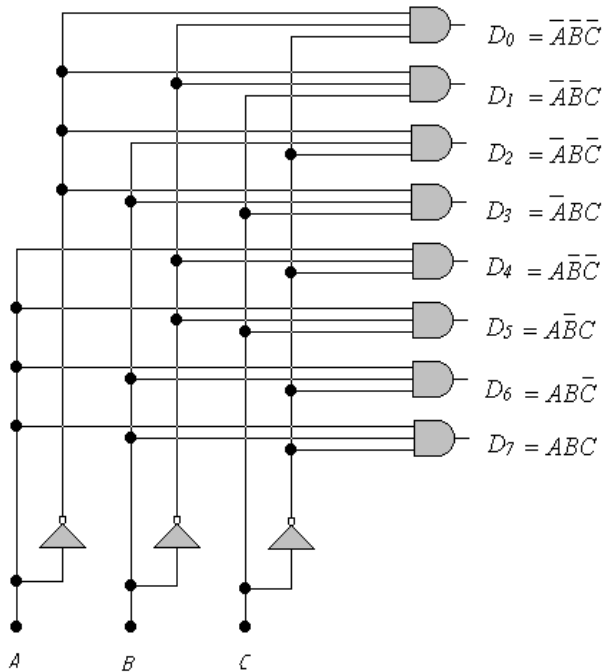
Diandaikan hanya ada satu saluran input dengan logik 1 untuk setiap kalinya, selain dari itu input tersebut tidak mempunyai arti. Tampak bahwa rangkaian itu mempunyai delapan input yang dapat memberikan 2^8 kemungkinan kombinasi, tetapi hanya delapan kombinasi yang mempunyai arti.



Gambar 6.13. Rangkaian encoder oktal ke biner

B. Decoder Biner ke Octal

Pada decoder dari biner ke oktal ini terdapat tiga input yaitu A , B dan C yang mewakili suatu bilangan biner tiga bit dan delapan output yang yaitu D_0 sampai dengan D_7 yang mewakili angka oktal dari 0 sampai dengan 1.



Gambar 6.14. Rangkaian decoder biner ke oktal

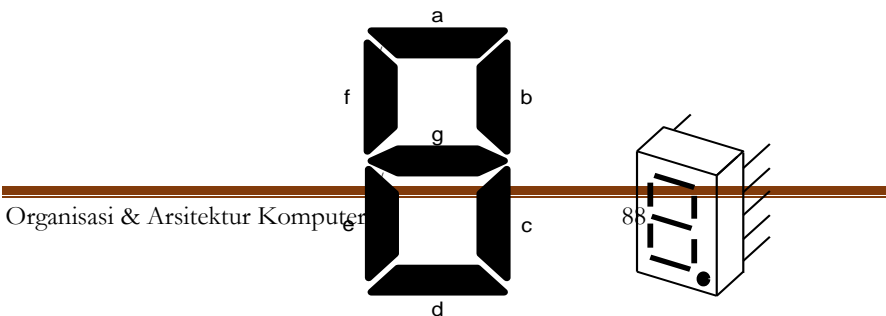
Dalam hal ini unsur informasinya adalah delapan angka oktal. Sandi untuk informasi diskrit ini terdiri dari bilangan biner yang diwakili oleh tiga bit. Kerja dekoder ini dapat lebih jelas tampak dari hubungan input dan output yang ditunjukkan pada tabel kebenaran dibawah ini. Tampak bahwa variabel outputnya itu hanya dapat mempunyai sebuah logk 1 ntuk setiap kombinasi inputnya. Saluran output yang nilainya sama dengan 1 mewakili angka oktal yang setara dengan bilangan biner pada saluran inputnya.

Tabel 6.8. Tabel Kebenaran Dekoder Biner ke Oktal

| Input | | | Output | | | | | | | |
|----------|----------|----------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| <i>A</i> | <i>B</i> | <i>C</i> | <i>D</i> ₀ | <i>D</i> ₁ | <i>D</i> ₂ | <i>D</i> ₃ | <i>D</i> ₄ | <i>D</i> ₅ | <i>D</i> ₆ | <i>D</i> ₇ |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

C. Peraga 7 segmen

Untuk menampilkan bilangan yang dikeluarkan oleh decoder akan dapat dipakai sebuah penampil 7-segmen (*seven segment display*). Penampil ini terdiri dari 7-segmen yang tersusun membentuk angka-angka, ditunjukkan pada gambar 6.12.



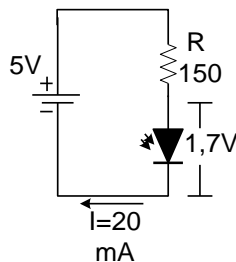
Gambar 6.15. Seven Segment

Cara mengidentifikasi segmen dalam penampil 7-segmen

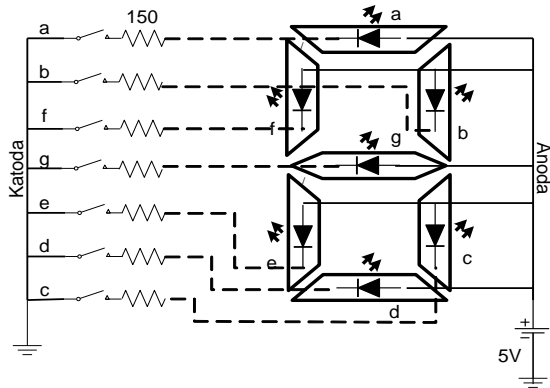
Segmen-segmen ditandai dengan huruf-huruf a, b, c, d, e, f dan g. setiap segmen dapat diisi sebuah filamen yang akan berpijar apabila diaktifkan. Jenis penampil semacam ini disebut penampil pijar (*incandescent display*). Cara memijarkan tidak beda dengan lampu-lampu pijar biasa.

Jenis penampil lain adalah yang segmen-segmennya mengandung tabung gas (*gas discharge tube*), yang beroperasi dengan tegangan tinggi. Penampil ini berpendar dengan warna jingga. Ada pula penampil pendaran (*fluorescent tube*) yang mengeluarkan cahaya kehijauan, dan beroperasi dengan tegangan rendah.

Penampil yang banyak dipakai adalah yang menerapkan LED (*Light Emitting Diode*). Untuk menyalakan LED diterapkanlah sirkit seperti pada gambar 6.13. $R=150\Omega$ berfungsi untuk membatasi arus agar bertahan pada 20mA. Tanpa R, LED akan terbakar. Pada LED akan terdapat tegangan kira-kira 1,7V.



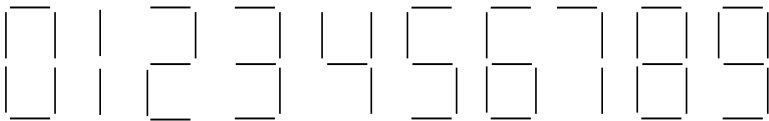
Gambar 6.16. Sirkit untuk menyalakan LED



Gambar 6.17. Azas Menyalakan LED

LED yang dibumikan (lewat $R=150\ \Omega$) akan menyala

Setiap segmen didalam penampil pada Gb.C1 berisi satu LED. Adapun dasarnya hubungan LED ditunjukkan dalam Gb.C3, yaitu anoda-anoda disatukan dan diberi potensial $+V_{cc}$ (5V). katodalah yang diberi logik 0 atau 1 dari dekoder lewat $R=150\ \Omega$. Apabila saklar ditutup, maka katoda yang bersangkutan memperoleh logik 0 dan LED itupun menyala, sebab sirkit baterai tertutup. Pada Gb.C4 ditunjukkan angka-angka yang akan dapat ditampilkan oleh tujuh segmen.



Gambar 6.18. Angka-angka yang dapat ditampilkan oleh 7-segmen

Sebagai contoh, untuk menyalakan atau menampilkan angka 6, maka saklar a, c, d, e, f, dan g harus ditutup, sehingga segmen-segmen a, c, d, e, f, dan g pun menyala. Dalam pelaksanaan praktek, segmen-segmen a hingga g dikoneksikan langsung pada keluaran a hingga g pada dekoder. Keluaran yang aktif akan meng-*ground*-kan segmen yang berkoneksi padanya, sehingga segmen tersebut menyala. Contoh, keluaran pada dekoder (a, b, c) aktif, maka output-output itu masing-masing meng-*ground*-kan katodanya LED yang ada di segmen a, b, dan c, sehingga tampilah 7.

D. Decoder BCD ke Desimal

Rangkaian Dekoder BCD ke desimal ditunjukkan pada gambar D2. Unsur informasi dalam hal ini adalah sepuluh angka desimal yang diwakili oleh sandi BCD. Masing-masing keluarannya sama dengan 1 hanya bila variabel masukannya membentuk suatu kondisi bit yang sesuai dengan angka desimal yang diwakili oleh sandi BCD itu. Tabel D2 menunjukkan hubungan masukan dan keluaran dekoder tersebut. Hanya sepuluh kombinasi masukan pertama yang berlaku untuk penentuan sandi itu, enam berikutnya tidak digunakan dan menurut definisi, merupakan keadaan tak acuh. Jelas keadaan tak acuh itu pada perencanaannya digunakan untuk menyederhanakan fungsi keluarannya, jika tidak setiap gerbang akan memerlukan empat masukan. Untuk kelengkapan analisis tabel D2 memberikan semua keluaran termasuk enam kombinasi yang tidak terpakai dalam sandi BCD itu; tetapi jelas keenam kombinasi tersebut tidak mempunyai arti apa-apa dalam rangkaian itu.

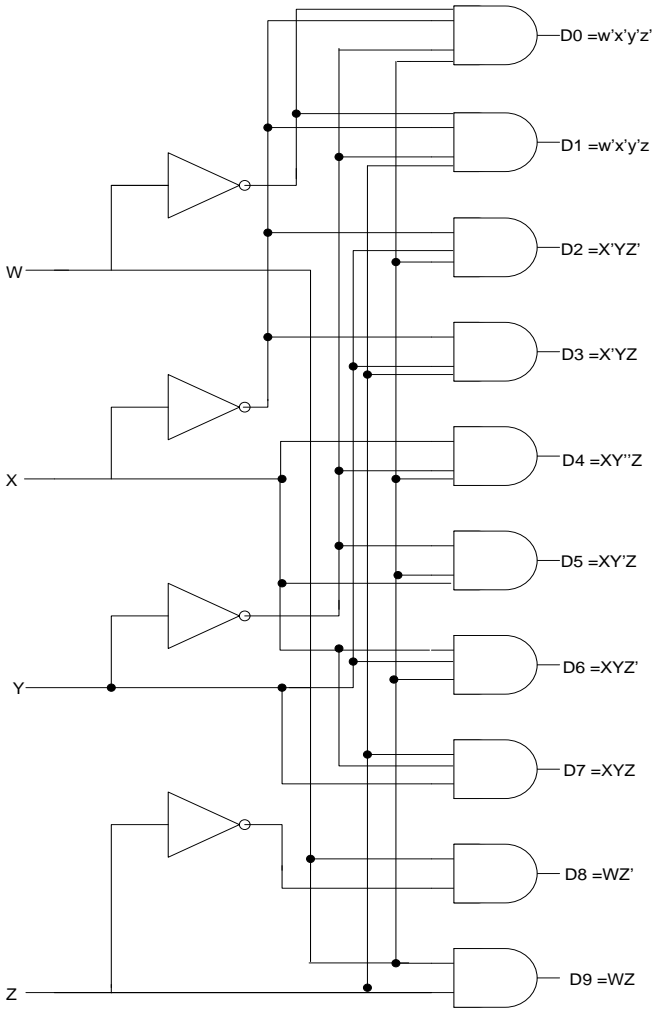
Dekoder dan enkoder itu banyak sekali dipakai dalam sistem digital. Dekoder tersebut berguna untuk memperagakan unsur informasi diskret yang tersimpan dalam register. Misalnya suatu angka desimal yang disandikan dalam BCD dan tersimpan dalam register empat sel dapat diperagakan dengan pertolongan rangkaian dekoder BCD ke desimal dimana keluaran keempat sel biner tersebut diubah sehingga menyalakan 10 lampu penunjuk. Lampu penunjuk itu dapat berupa angka peraga (*display digit*), sehingga suatu angka desimal akan menyala bila keluaran dekoder yang sesuai adalah logika 1. Rangkaian dekoder juga berguna untuk menentukan isi register dalam proses pengambilan keputusan. Pemakaiannya yang lain adalah untuk membangkitkan sinyal waktu dan sinyal urutan untuk keperluan pengaturan.

Tabel 6.9. Tabel kebenaran decoder BCD ke desimal

| Masukan | | | | Keluaran | | | | | | | | | |
|---------|---|---|---|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| w | x | y | z | D ₀ | D ₁ | D ₂ | D ₃ | D ₄ | D ₅ | D ₆ | D ₇ | D ₈ | D ₉ |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

Rangkaian Logika Sekuensial

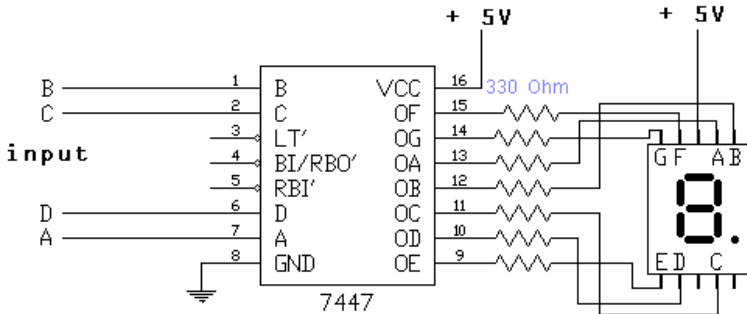
| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0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



Gambar 6.19. Dekoder BCD ke decimal

Decoder BCD ini ada 2 macam yaitu yang outputnya aktif level tinggi dan yang outputnya aktif rendah sehingga membutuhkan 7 segmen yang berbeda. Untuk aktif level tinggi menggunakan 7 segmen kommon katoda, sedangkan untuk aktif level rendah menggunakan 7 segmen kommon anoda.

Contoh rangkaian Decoder BCD to 7 segmen kommon anoda



Gambar 6.20. Rangkaian Decoder BCD to 7 segmen kommon anoda

Tabel 6.10. Tabel Kebenaran Decoder common anoda

kommon anoda

| D | C | B | A | g | f | e | d | c | b | a |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Tabel 6.11. Tabel Kebenaran Decoder common katoda

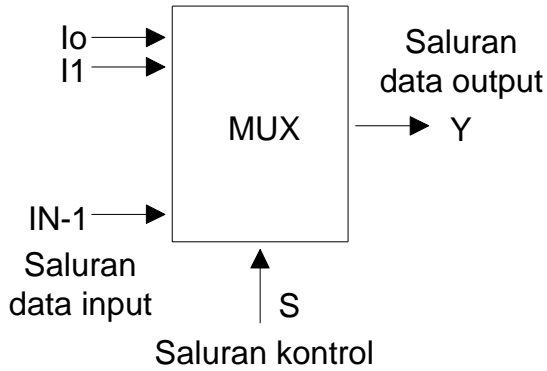
kommon katoda

| D | C | B | A | g | f | e | d | c | b | a | |
|---|---|---|---|---|---|---|---|---|---|---|-----|
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | → 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | → 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | → 2 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | → 3 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | → 4 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | → 5 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | → 6 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | → 7 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | → 8 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | → 9 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | → E |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | → 5 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | → 4 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | → 6 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | → E |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | → B |

Dengan demikian untuk peraga 7 segmen jenis common cathode memerlukan decoder dengan output jenis active high untuk menyalakan setiap segmennya, sedangkan untuk peraga 7 segmen jenis common anode memerlukan decoder dengan output jenis active low.

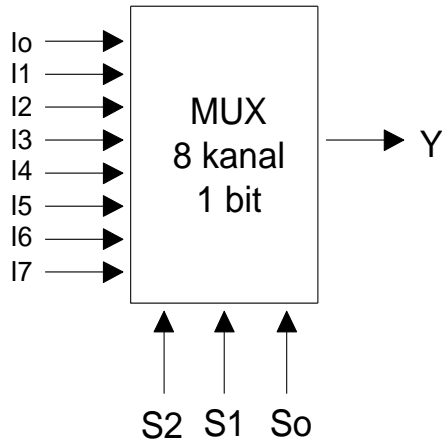
6.3. Multiplexer dan Demultiplexer

Multiplexer dapat didefinisikan sebagai suatu rangkaian logika yang dapat menerima beberapa saluran data input yang terdiri dari 1 bit/lebih secara paralel dan pada outputnya hanya dilewatkan salah satu saluran data yang terpilih. Saluran data input yang terpilih dikontrol oleh beberapa saluran control yang sering disebut sebagai saluran pemilih (input select). Jumlah saluran control berkaitan erat dengan jumlah saluran data input yang akan dikontrol. Multiplexer sering juga disebut dengan selector data. Diagram sebuah multiplexer secara umum :



Gambar 6.21. Diagram Multiplexer

Contoh multiplexer 8 kanal 1 bit :

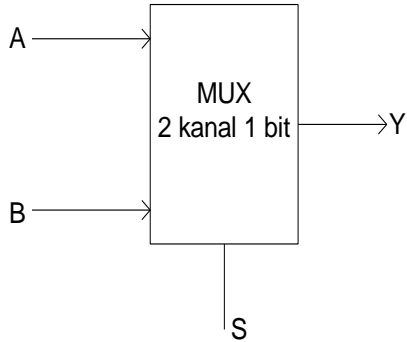


Gambar 6.22. Diagram Multiplexer 8 kanal 1 bit

Contoh soal :

1. Rancanglah sebuah MUX 2 kanal 1 bit.

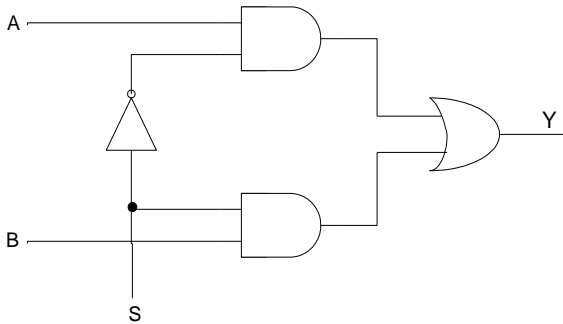
Jawab :



Tabel kebenaran MUX 2 kanal 1 bit

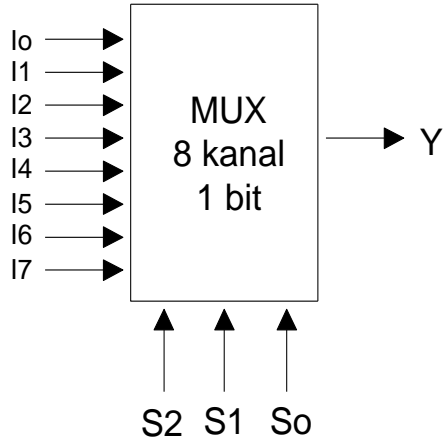
| Selector (S) | Output (Y) |
|--------------|------------|
| 0 | $S'A$ |
| 1 | SB |

Rangkaian dalam MUX 2 kanal 1 bit



2. Rancanglah MUX 8 kanal 1 bit.

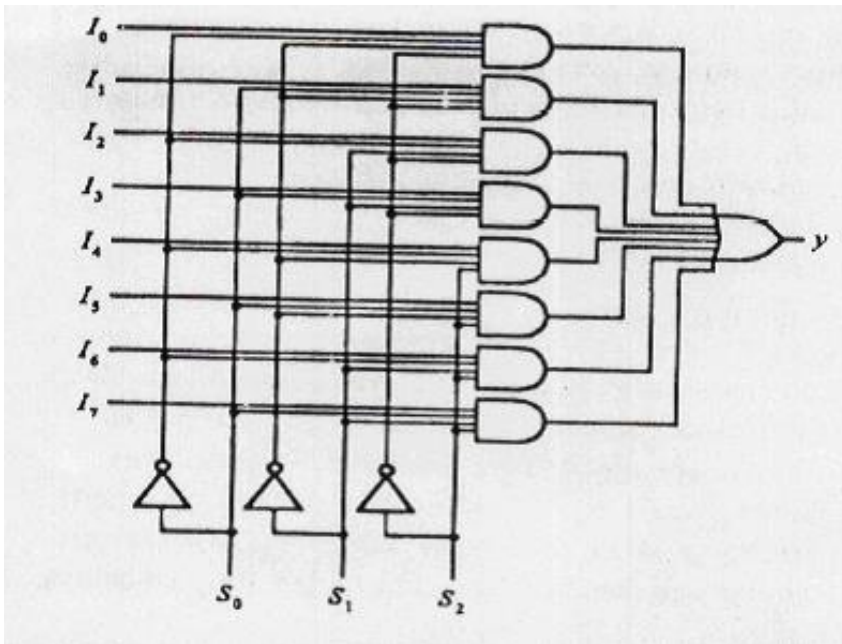
Jawab :



Tabel kebenaran MUX 8 kanal 1 bit

| S_2 | S_1 | S_0 | Output (Y) |
|-------|-------|-------|------------|
| 0 | 0 | 0 | I_0 |
| 0 | 0 | 1 | I_1 |
| 0 | 1 | 0 | I_2 |
| 0 | 1 | 1 | I_3 |
| 1 | 0 | 0 | I_4 |
| 1 | 0 | 1 | I_5 |
| 1 | 1 | 0 | I_6 |
| 1 | 1 | 1 | I_7 |

Rangkaian dalam MUX 8 kanal 1 bit



Rangkuman

1. Rangkaian Logika Sekuensial adalah rangkaian logika yang outputnya tergantung pada keadaan input-inputnya juga tergantung pada keadaan output sebelumnya.
2. Suatu rangkaian flip-flop dapat disusun dengan dua gerbang NOR atau dua gerbang NAND.
3. Ada beberapa jenis Flip-Flop yaitu Flip-flop RS, Flip-flop D, Flip-flop T, dan Flip-flop JK.
4. Encoder adalah suatu rangkaian logika yang berfungsi untuk mengkonversikan kode yang lebih dikenal oleh manusia ke dalam kode yang kurang dikenal manusia. Decoder adalah suatu rangkaian logika yang berfungsi untuk mengkonversikan kode yang kurang dikenal manusia kedalam kode yang lebih dikenal manusia.
5. *Multiplexer* dapat didefinisikan sebagai suatu rangkaian logika yang dapat menerima beberapa saluran data input yang terdiri dari 1 bit/lebih secara paralel dan pada outputnya hanya dilewatkan salah satu saluran data yang terpilih.

Tes Formatif

1. Suatu rangkaian logika sekuensi memiliki sebuah input R satu bit dan output Y empat bit.
 - a. Berapa jumlah flip flop yang diperlukan sebagai unit penyimpan rangkaian tersebut ?
 - b. Jika unit penyimpanannya menggunakan flip flop D, gambarkan diagram blok rangkaian dengan menampilkan semua flip flop yang diperlukan.
2. Implementasikan persamaan berikut ini ke dalam dua bentuk yakni menggunakan multiplexer 8 ke 1 dan decoder 3 ke 8 !
 - a. $Y = A'B'C + A'BC' + AB'C' + ABC'$
 - b. $Y = A + BC'$
 - c. $Y = (A' + B) (A + C')$

SATUAN BAHASAN 7

CENTRAL PROCESSING UNIT (CPU)

1. Gambaran Singkat Mengenai Materi Kuliah

Pada satuan bahasan yang ke 7 ini, akan dibahas tentang salah satu komponen penting dalam sistem komputer yaitu CPU (*Central Processing Unit*). Seperti telah dijelaskan pada bagian pengantar, bahwa komputer digital terdiri dari sistem prosesor atau sering disebut CPU, memori–memori, dan piranti masukan/keluaran yang saling berhubungan dan saling dukung mewujudkan fungsi operasi komputer secara keseluruhan.

CPU merupakan komponen yang mengolah data berdasarkan instruksi-instruksi yang diberikan padanya. Di bagian ini akan dipelajari mengenai komponen utama CPU, fungsi utama CPU yang meliputi siklus *fetch and execution*, dan fungsi interrupt.

2. Pedoman Mempelajari Materi

Baca dengan cermat dan teliti materi pada satuan bahasan ini, kemudian pahami bagian-bagian dari CPU. Hal ini akan membantu anda dalam memahami pentingnya fungsi CPU dalam berjalannya sistem komputer. Pada akhir satuan bahasan terdapat tes formatif yang perlu anda kerjakan sebagai sarana untuk mengukur seberapa jauh tingkat pemahaman anda terhadap materi ini.

3. Tujuan Pembelajaran

Setelah mempelajari satuan bahasan 7 ini, mahasiswa diharapkan dapat :

- menjelaskan komponen-komponen utama dalam CPU beserta fungsinya
- menjelaskan fungsi utama CPU
- menjelaskan siklus instruksi pada CPU yang terdiri atas siklus *fetch* dan siklus *execution*
- menjelaskan fungsi dan tujuan interupsi
- menjelaskan macam-macam kelas sinyal interupsi

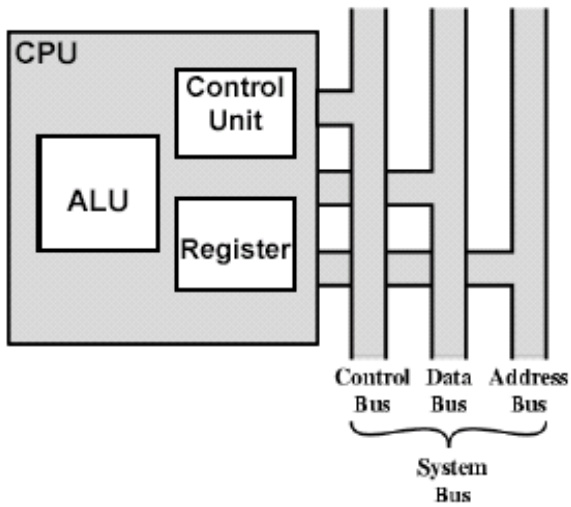
- engetahui hal-hal yang dilakukan prosesor jika interupsi ditangguhkan
- menjelaskan dua metode pendekatan dalam penanganan interupsi

3.1 Komponen Utama CPU

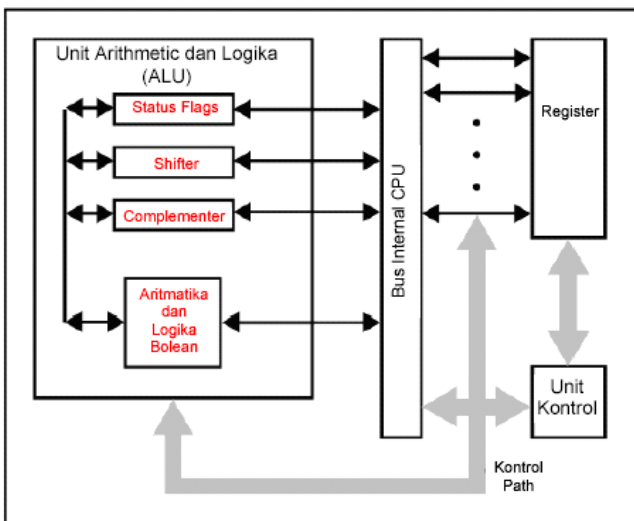
CPU merupakan komponen terpenting dari sistem komputer. CPU berfungsi mengolah data berdasarkan instruksi–instruksi yang diberikan kepadanya.

Dalam mewujudkan fungsi dan tugasnya, CPU tersusun atas beberapa komponen sebagai bagian dari struktur CPU, seperti terlihat pada gambar 7.1 dan struktur detail internal CPU terlihat pada gambar 7.2. CPU tersusun atas beberapa komponen,yaitu :

- *Arithmetic and Logic Unit (ALU)*, bertugas membentuk fungsi–fungsi pengolahan data komputer. ALU sering disebut *mesin bahasa (machine language)* karena bagian ini mengerjakan instruksi–instruksi bahasa mesin yang diberikan padanya. Seperti istilahnya, ALU terdiri dari dua bagian, yaitu unit aritmetika dan unit logika boolean, yang masing–masing memiliki spesifikasi tugas tersendiri.
- *Control Unit*, bertugas mengontrol operasi CPU dan secara keseluruhan mengontrol komputer sehingga terjadi sinkronisasi kerja antar komponen dalam menjalankan fungsi–fungsi operasinya. Termasuk dalam tanggung jawab unit control adalah mengambil instruksi–instruksi dari memori utama dan menentukan jenis instruksi tersebut.
- *Registers*, adalah media penyimpanan internal CPU yang digunakan saat proses pengolahan data. Memori ini bersifat sementara, biasanya digunakan untuk menyimpan data saat diolah atau pun data untuk pengolahan selanjutnya.
- *CPU Interconnections*, adalah sistem koneksi dan bus yang menghubungkan komponen internal CPU, yaitu ALU, unit control dan register–register dan juga dengan bus–bus eksternal CPU yang menghubungkan dengan sistem lainnya, seperti memori utama, piranti masukan/keluaran.



Gambar 7.1 Komponen internal CPU

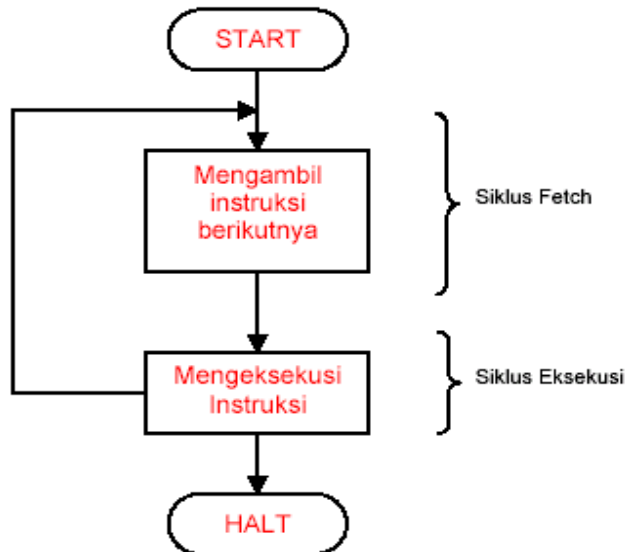


Gambar 7.2 Struktur detail internal CPU

7.1 Fungsi CPU

Fungsi CPU adalah menjalankan program-program yang disimpan dalam memori utama dengan cara mengambil instruksi-instruksi, menguji instruksi tersebut dan mengeksekusinya satu persatu sesuai alur perintah.

Untuk memahami fungsi CPU dan caranya berinteraksi dengan komponen lain, perlu kita tinjau lebih jauh proses eksekusi program. Pandangan paling sederhana proses eksekusi program adalah dengan mengambil pengolahan instruksi yang terdiri dari dua langkah, yaitu: operasi pembacaan instruksi (*fetch*) dan operasi pelaksanaan instruksi (*execute*). *Siklus instruksi* yang terdiri dari siklus fetch dan siklus eksekusi diperlihatkan pada gambar 7.3 berikut.



Gambar 7.3 Siklus instruksi dasar

7.1.1 Siklus Fetch - Eksekusi

Pada setiap siklus instruksi, CPU awalnya akan membaca instruksi dari memori. Terdapat register dalam CPU yang berfungsi mengawasi dan menghitung instruksi selanjutnya, yang disebut *Program Counter* (PC). PC akan menambah satu hitungannya setiap kali CPU membaca instruksi.

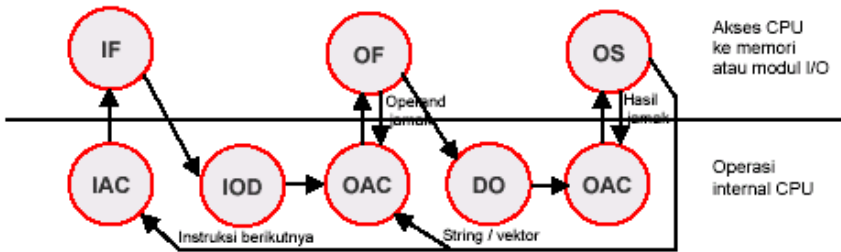
Instruksi–instruksi yang dibaca akan dibuat dalam register instruksi (IR). Instruksi–instruksi ini dalam bentuk kode–kode biner yang dapat diinterpretasikan oleh CPU kemudian dilakukan aksi yang diperlukan. Aksi– aksi ini dikelompokkan menjadi empat kategori, yaitu :

- *CPU – Memori*, perpindahan data dari CPU ke memori dan sebaliknya.
- *CPU – I/O*, perpindahan data dari CPU ke modul I/O dan sebaliknya.
- *Pengolahan Data*, CPU membentuk sejumlah operasi aritmatika dan logika terhadap data.
- *Kontrol*, merupakan instruksi untuk pengontrolan fungsi atau kerja. Misalnya instruksi pengubahan urusan eksekusi.

Perlu diketahui bahwa siklus eksekusi untuk suatu instruksi dapat melibatkan lebih dari sebuah referensi ke memori. Disamping itu juga, suatu instruksi dapat menentukan suatu operasi I/O. Perhatikan gambar 7.4 yang merupakan detail siklus operasi pada gambar 7.3, yaitu :

- *Instruction Address Calculation (IAC)*, yaitu mengkalkulasi atau menentukan alamat instruksi berikutnya yang akan dieksekusi. Biasanya melibatkan penambahan bilangan tetap kealamat instruksi sebelumnya. Misalnya, bila panjang setiap instruksi 16 bit padahal memori memiliki panjang 8 bit, maka tambahkan 2 ke alamat sebelumnya.
- *Instruction Fetch (IF)*, yaitu membaca atau mengambil instruksi dari lokasi memorinya ke CPU.
- *Instruction Operation Decoding (IOD)*, yaitu menganalisa instruksi untuk menentukan jenis operasi yang akan dibentuk dan operand yang akan digunakan.
- *Operand Address Calculation (OAC)*, yaitu menentukan alamat operand, hal ini dilakukan apabila melibatkan referensi operand pada memori.
- *Operand Fetch (OF)*, adalah mengambil operand dari memori atau dari modul I/O.

- *Data Operation (DO)*, yaitu membentuk operasi yang diperintahkan dalam instruksi.
- *Operand store (OS)*, yaitu menyimpan hasil eksekusike dalam memori.



Gambar 7.4 Diagram siklus instruksi

7.1.2 Fungsi Interrupt

Fungsi interupsi adalah mekanisme penghentian atau pengalihan pengolahan instruksi dalam CPU kepada routine interupsi. Hampir semua modul (memori dan I/O) memiliki mekanisme yang dapat menginterupsi kerja CPU.

Tujuan interupsi secara umum untuk manajemen pengeksekusian routine instruksi agar efektif dan efisien antar CPU dan modul–modul I/O maupun memori. Setiap komponen komputer dapat menjalankan tugasnya secara bersamaan, tetapi kendali terletak pada CPU disamping itu kecepatan eksekusi masing–masing modul berbeda sehingga dengan adanya fungsi interupsi ini dapat sebagai sinkronisasi kerja antar modul. Macam–macam kelas sinyal interupsi:

- *Program*, yaitu interupsi yang dibangkitkan dengan beberapa kondisi yang terjadi pada hasil eksekusi program. Contohnya: arimatika over flow, pembagian nol, operasi ilegal.
- *Timer*, adalah interupsi yang dibangkitkan pewaktuan dalam prosesor. Sinyal ini memungkinkan sistem operasi menjalankan fungsi tertentu secara reguler.

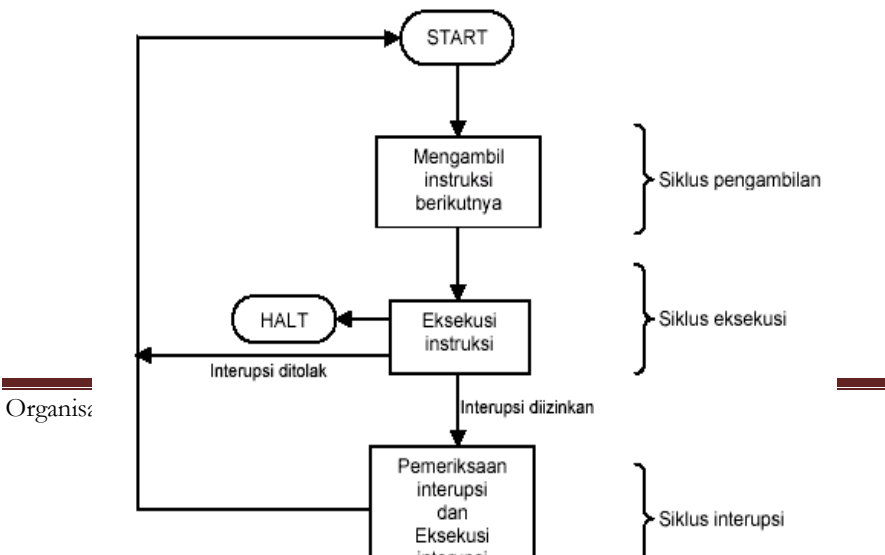
- *I/O*, sinyal interupsi yang dibangkitkan oleh modul I/O sehubungan pemberitahuan kondisi error dan penyelesaian suatu operasi.
- *Hardware failure*, adalah interupsi yang dibangkitkan oleh kegagalan daya atau kesalahan paritas memori.

Dengan adanya mekanisme interupsi, prosesor dapat digunakan untuk mengeksekusi instruksi– instruksi lain. Saat suatu modul telah selesai menjalankan tugasnya dan siap menerima tugas berikutnya maka modul ini akan mengirimkan permintaan interupsi ke prosesor. Kemudian prosesor akan menghentikan eksekusi yang dijalankannya untuk menghandel routine interupsi. Setelah program interupsi selesai maka prosesor akan melanjutkan eksekusi programnya kembali. Saat sinyal interupsi diterima prosesor ada dua kemungkinan tindakan, yaitu interupsi diterima/ditanggguhkan dan interupsi ditolak. Apabila interupsi ditanggguhkan, prosesor akan melakukan hal– hal dibawah ini:

1. Prosesor menanggguhkan eksekusi program yang dijalankan dan menyimpan konteksnya. Tindakan ini adalah menyimpan alamat instruksi berikutnya yang akan dieksekusi dan data lain yang relevan.
2. Prosesor menyetel program counter (PC) kealamat awal routine *interrupthandler*.

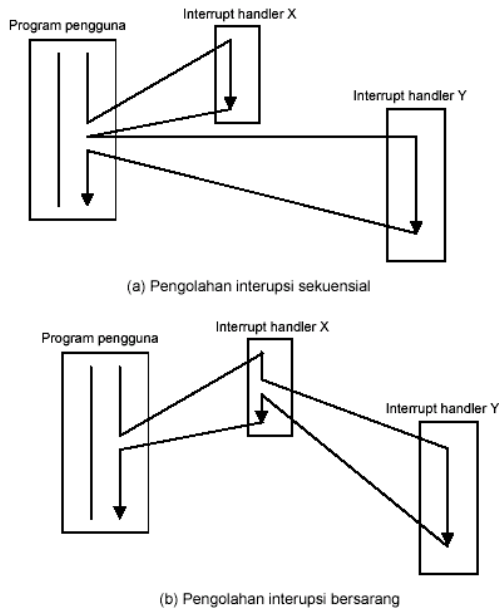
Gambar 7.5 berikut menjelaskan siklus eksekusi oleh prosesor dengan adanya fungsi interupsi.

Untuk sistem operasi yang kompleks sangat dimungkinkan adanya interupsi ganda (*multiple interrupt*). Misalnya suatu komputer akan menerima permintaan interupsi saat proses pencetakan dengan printer selesai, disamping itu dimungkinkan dari saluran komunikasi akan mengirimkan permintaan interupsi setiap kali data tiba. Dalam hal ini prosesor harus menangani interupsi ganda.



Gambar 7.5 Siklus eksekusi instruksi dengan interrupt

Dapat diambil dua buah pendekatan untuk menangani interupsi ganda ini. Pertama adalah menolak atau tidak mengizinkan interupsi lain saat suatu interupsi ditangani prosesor. Kemudian setelah prosesor selesai menangani suatu interupsi maka interupsi lain baru ditangani. Pendekatan ini disebut *pengolahan interupsi berurutan/sekuensial*. Pendekatan ini cukup baik dan sederhana karena interupsi ditangani dalam urutan yang cukup ketat. Kelemahan pendekatan ini adalah metode ini tidak memperhitungkan prioritas interupsi. Pendekatan ini diperlihatkan pada gambar 7.6a.



Gambar 7.6 Transfer pengendalian pada interupsi ganda

Pendekatan kedua adalah dengan mendefinisikan prioritas bagi interupsi dan *interrupt handler* mengizinkan interupsi berprioritas lebih tinggi ditangani terlebih dahulu. Pendekatan ini disebut *pengolahan interupsi bersarang*. Metode ini digambarkan pada gambar 7.6b.

Sebagai contoh untuk pendekatan bersarang, misalnya suatu sistem memiliki tiga perangkat I/O: printer, disk, dan saluran komunikasi, masing-masing prioritasnya 2, 4 dan 5. Pada awal sistem melakukan pencetakan dengan printer, saat itu terdapat pengiriman data pada saluran komunikasi sehingga modul komunikasi meminta interupsi. Proses selanjutnya adalah pengalihan eksekusi interupsi modul komunikasi, sedangkan interupsi printer ditangguhkan. Saat pengekseskuan modul komunikasi terjadi interupsi disk, namun karena prioritasnya lebih rendah maka interupsi disk ditangguhkan. Setelah interupsi modul komunikasi selesai akan dilanjutkan interupsi yang memiliki prioritas lebih tinggi, yaitu disk. Bila interupsi disk selesai dilanjutkan eksekusi interupsi printer. Selanjutnya dilanjutkan eksekusi program utama.

Rangkuman

1. CPU (*Central Processing Unit*) merupakan komponen terpenting dari sistem computer.
2. CPU terdiri atas beberapa komponen utama yaitu : ALU (*Arithmetic and Logic Unit*), *Control Unit*, *Register*, dan *CPU Interconnection*.
3. Fungsi CPU adalah menjalankan program-program yang disimpan dalam memori utama dengan cara mengambil instruksi-instruksi, menguji instruksi tersebut dan mengeksekusinya satu per satu sesuai alur perintah.
4. Siklus instruksi dalam proses eksekusi program terdiri dari siklus fetch dan siklus eksekusi.
5. Fungsi interrupt (interupsi) adalah mekanisme penghentian atau pengalihan pengolahan instruksi dalam CPU kepada routine interupsi.
6. Tujuan interupsi secara umum untuk manajemen pengekseskuan routine instruksi agar efektif dan efisien antar CPU dan modul-modul I/O maupun memori.
7. Macam-macam kelas sinyal interupsi adalah : program, timer, I/O dan hardware failure.
8. Jelaskan hal-hal yang dilakukan prosesor jika interupsi ditangguhkan!

Tes Formatif

1. Jelaskan komponen utama CPU beserta fungsinya masing-masing !
2. Jelaskan fungsi dan tujuan dari CPU !
3. Jelaskan mekanisme siklus fetch dan eksekusi dalam proses eksekusi program !
4. Jelaskan fungsi dan tujuan interupsi !
5. Jelaskan macam-macam kelas sinyal interupsi !
6. Jelaskan dua model pendekatan dalam menangani interupsi ganda !

SATUAN BAHASAN 8

MEMORI UTAMA

1. Gambaran Singkat Mengenai Materi Kuliah

Memori adalah bagian dari komputer tempat program-program dan data-data disimpan. Tanpa sebuah memori sebagai tempat untuk mendapatkan informasi guna dibaca dan ditulis oleh prosesor maka tidak akan ada komputer-komputer digital dengan sistem penyimpanan program. Memori dikelompokkan atas memori internal atau memori utama, dan memori eksternal atau memori sekunder.

Dalam satuan bahasan ini akan dipelajari mengenai memori internal yang meliputi operasi sel memori, karakteristik sistem memori, keandalan memori, satuan memori, memori utama semikonduktor, dan cache memori.

2. Pedoman Mempelajari Materi

Baca dengan cermat dan teliti materi pada satuan bahasan berikut, kemudian pahami setiap bagian dari memori internal. Hal ini akan membantu anda dalam karakteristik sistem memori. Pada akhir satuan bahasan terdapat tes formatif yang perlu anda kerjakan sebagai sarana untuk mengukur seberapa jauh tingkat pemahaman anda terhadap materi ini.

3. Tujuan Pembelajaran

Setelah mempelajari satuan bahasan 8 ini, mahasiswa diharapkan dapat :

- menjelaskan perbedaan antara memori internal dengan metode eksternal.
- menjelaskan sifat-sifat tertentu dari seluruh sel memori.
- menjelaskan karakteristik penting sistem manajer komputer
- elakukan mekanisme perbaikan kesalahan dengan menggunakan kode Hamming
- menjelaskan fungsi cache memori

menjelaskan unsur-unsur rancangan cache memori

8.1 Operasi Sel Memori

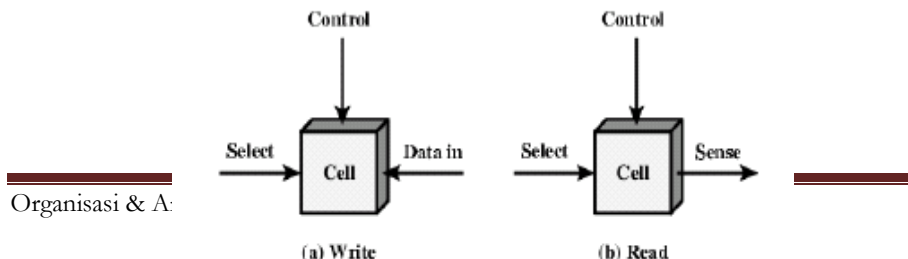
Memori adalah bagian dari komputer tempat program-program dan data-data disimpan. Beberapa pakar komputer (terutama dari Inggris) menggunakan istilah *store* atau *storage* untuk memori, meskipun kata *storage* sering digunakan untuk menunjuk ke penyimpanan disket.

Walaupun konsepnya sederhana, memori komputer memiliki aneka ragam jenis, teknologi, organisasi, unjuk kerja dan harganya. Memori dapat dikelompokkan atas memori internal atau memori utama, dan memori eksternal atau memori sekunder. *Memori internal* adalah memori yang dapat diakses langsung oleh prosesor. Sebenarnya terdapat beberapa macam memori internal, yaitu register yang terdapat didalam prosesor, cache memori dan memori utama berada diluar prosesor. Sedangkan *memori eksternal* adalah memori yang diakses prosesor melalui piranti I/O, seperti disket dan hardisk.

Elemen dasar memori adalah sel memori. Walaupun digunakan sejumlah teknologi elektronik, seluruh sel memori memiliki sifat-sifat tertentu :

- Sel memori memiliki dua keadaan stabil (atau semi-stabil), yang dapat digunakan untuk merepresentasikan bilangan biner 1 atau 0.
- Sel memori mempunyai kemampuan untuk ditulisi (sedikitnya satu kali).
- Sel memori mempunyai kemampuan untuk dibaca.

Gambar 8.1 menjelaskan operasi sel memori. Umumnya sel memori mempunyai tiga terminal fungsi yang mampu membawa sinyal listrik. Terminal *select* berfungsi memilih operasi tulis atau baca. Untuk penulisan, terminal lainnya menyediakan sinyal listrik yang men-set keadaan sel bernilai 1 atau 0, sedangkan untuk operasi pembacaan, terminal ini digunakan sebagai keluaran.



Gambar 8.1. Operasi Sel Memori

8.2 Karakteristik Sistem Memori

Untuk mempelajari sistem memori secara keseluruhan, harus mengetahui karakteristik–karakteristik kuncinya. Karakteristik penting sistem memori disajikan dalam tabel 8.1 berikut:

Tabel 8.1 Karakteristik penting sistem memori komputer

| Karakteristik | Macam/Keterangan |
|---------------------|---|
| Lokasi | 1. CPU 2. Internal(main) 3. External(secondary) |
| Kapasitas | 1. Ukuran word 2. Jumlah word |
| Satuan transfer | 1. Word 2. Block |
| Metode akses | 1. Sequential access 2. Direct access 3. Random access 4. Associative access |
| Kinerja | 1. Accesstime 2. Cycletime 3. Transferrate |
| Tipe fisik | 1. Semi konduktor 2. Magnetik |
| Karakteristik fisik | 1. Volatile/nonvolatile 2. Erasable/nonerasable |

Dilihat dari *lokasi*, memori dibedakan menjadi beberapa jenis, yaitu register, memori internal dan memori eksternal. Register berada di dalam chipprosesor, memori ini diakses langsung oleh prosesor dalam menjalankan operasinya. Register digunakan sebagai memori sementara dalam perhitungan maupun pengolahan data dalam prosesor. Memori internal adalah memori yang berada diluar chip prosesor namun mengaksesnya langsung oleh prosesor. Memori internal dibedakan

menjadi memori utama dan cache memori. Memori eksternal dapat diakses oleh prosesor melalui piranti I/O, memori ini dapat berupa disk maupun pita.

Karakteristik lainnya adalah *kapasitas*. Kapasitas memori internal maupun eksternal biasanya dinyatakan dalam bentuk *byte* (1 byte = 8 bit) atau *word*. Panjang word umumnya 8, 16, 32 bit. Memori eksternal biasanya lebih besar kapasitasnya dari pada memori internal, hal ini disebabkan karena teknologi dan sifat penggunaannya yang berbeda.

Karakteristik berikutnya adalah *satuan transfer*. Bagi memori internal, satuan transfer sama dengan jumlah saluran data yang masuk ke dalam dan keluar dari modul memori. Jumlah saluran ini seringkali sama dengan panjang word, tapi dimungkinkan juga tidak sama. Tiga konsep yang berhubungan dengan satuan transfer:

- *Word*, merupakan satuan “alami” organisasi memori. Ukuran word biasanya sama dengan jumlah bit yang digunakan untuk representasi bilangan dan panjang instruksi.
- *Addressable system units*, pada sejumlah sistem, addressable units adalah word. Namun terdapat sistem dengan pengalaman tanpa dating kata byte. Pada semua kasus hubungan antara panjang A suatu alamat dan jumlah addressable unit adalah $2^A = N$.
- *Unit of transfer*, adalah jumlah bit yang dibaca atau dituliskan kedalam memori pada suatu saat. Pada memori eksternal, transfer data biasanya lebih besar dari suatu word, yang disebut dengan *block*.

Perbedaan tajam yang terdapat pada sejumlah jenis memori adalah *metode akses*-nya. Terdapat empat macam metode :

- *Sequential access*, memori diorganisasi menjadi unit-unit data yang disebut *record*. Akses harus dibuat dalam bentuk urutan linier yang spesifik. Informasi mengalamatkan yang disimpan dipakai untuk memisahkan record-record dan untuk membantu proses pencarian. Terdapat *shared read/write mechanism* untuk penulisan/pembacaan memorinya. Pita magnetic merupakan memori yang menggunakan metode sequential access.
- *Direct access*, sama sequential access terdapat *shared read/write mechanism*. Setiap blok dan record memiliki alamat unik berdasarkan lokasi fisiknya. Akses dilakukan langsung pada alamat memori. Disk adalah memori direct access.
- *Random access*, setiap lokasi memori dipilih secara random dan diakses serta dialamati secara langsung. Contohnya adalah memori utama.
- *Associative access*, merupakan jenis random akses yang memungkinkan perbandingan lokasi bit yang diinginkan untuk pencocokan. Jadi data dicari berdasarkan isinya bukan alamatnya

dalam memori. Contoh memori ini adalah cache memori yang akan dibahas diakhir bab ini.

Berdasarkan karakteristik unjuk kerja, memiliki tiga parameter utama pengukuran unjuk kerja, yaitu :

- *Access time*, bagi random access memory, waktu akses adalah waktu yang dibutuhkan untuk melakukan operasi baca atau tulis. Sedangkan untuk memori non-random akses merupakan waktu yang dibutuhkan dalam melakukan mekanisme baca atau tulis pada lokasi tertentu.
- *Memory cycle time*, konsep ini digunakan pada random access memory dan terdiri dari access time ditambah dengan waktu yang diperlukan transient agar hilang pada saluran sinyal.
- *Transfer rate*, adalah kecepatan data transfer keunit memori atau dari unit memori. Pada random access memory sama dengan $1 / (\text{cycle time})$. Sedangkan untuk non-random access memory dengan perumusan :

$$T_N = T_A + \frac{N}{R}$$

- T_N = waktu rata – rata untuk membaca atau menulis
N bit
 T_A = waktu akses rata – rata
 N = jumlah bit
 R = kecepatan transfer dalam bit perdetik (bps)

Jenis *tipe fisik* memori yang digunakan saat ini adalah memori semi konduktor dengan teknologi VLSI dan memori permukaan magnetic seperti yang digunakan pada disk dan pita magnetik.

Berdasarkan *karakteristik fisik*, media penyimpanan dibedakan menjadi *volatile* dan *non-volatile*, serta *erasable* dan *non erasable*. Pada *volatile memory*, informasi akan hilang apabila daya listriknya dimatikan, sedangkan *non-volatile memory* tidak hilang walau daya listriknya hilang. Memori permukaan magnetic adalah contoh *no-nvolatile memory*, sedangkan semi konduktor ada yang *volatile* dan *non-volatile*. Ada jenis memori semi konduktor yang tidak bisa dihapus kecuali dengan menghancurkan unit *storage*-nya, memori ini dikenal dengan ROM (*Read Only Memory*).

8.3 Keandalan Memori

Untuk memperoleh keandalan sistem ada tiga pertanyaan yang diajukan: Berapa banyak ? Berapa cepat ? Berapa mahal?

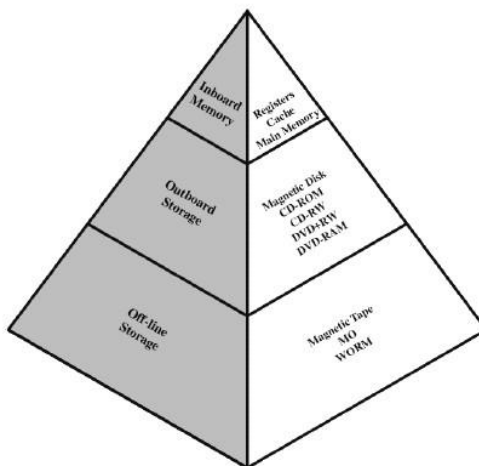
Pertanyaan berapa banyak adalah sesuatu yang sulit dijawab, karena berapa pun kapasitas memori tentu aplikasi akan menggunakannya. Jawaban pertanyaan berapa cepat adalah memori harus mampu mengikuti kecepatan CPU sehingga terjadi sinkronisasi kerja antar CPU dan memori tanpa adanya waktu tunggu karena komponen lain belum selesai prosesnya. Mengenai harga, sangatlah relatif. Bagi produsen selalu mencari harga produksi paling murah tanpa mengorbankan kualitasnya untuk memiliki daya saing dipasaran.

Hubungan harga, kapasitas dan waktu akses adalah :

- Semakin kecil waktu akses, semakin besar harga perbitnya.
- Semakin besar kapasitas, semakin kecil harga perbitnya.
- Semakin besar kapasitas, semakin besar waktu aksesnya.

Dilema yang dihadapi para perancang adalah keinginan menerapkan teknologi untuk kapasitas memori yang besar karena harga perbit yang murah namun hal itu dibatasi oleh teknologi dalam memperoleh waktu akses yang cepat. Salah satu pengorganisasian masalah ini adalah menggunakan *hirarki memori*. Seperti terlihat pada gambar 8.2, bahwa semakin menurunnya hirarki maka hal berikut akan terjadi:

- Penurunan harga/bit
- Peningkatan kapasitas
- Peningkatan waktu akses
- Penurunan frekuensi akses memori oleh CPU



Gambar 8.2 Hirarki memori

| Tipe memori | Teknologi | Ukuran | Waktu akses |
|---------------|-------------------|--------------|----------------------|
| Cache Memory | Semikonduktor RAM | 128 – 512 KB | 10 ns |
| Memori Utama | Semikonduktor RAM | 4 – 128 MB | 50 ns |
| Disk magnetik | Hard Disk | Gigabyte | 10 ms, 10MB/det |
| Disk Optik | CD-ROM | Gigabyte | 300ms, 600KB/det |
| Pita magnetik | Tape | 100 MB | Det- met 10MB/met |

Kunci keberhasilan hirarki ini pada penurunan frekuensi aksesnya. Semakin lambat memori maka keperluan CPU untuk mengaksesnya semakin sedikit. Secara keseluruhan sistem komputer akan tetap cepat namun kebutuhan kapasitas memori besar terpenuhi.

Tabel 8.2 Tabel spesifikasi memori

8.4 Satuan Memori

Satuan pokok memori adalah digit biner, yang disebut *bit*. Suatu bit dapat berisi sebuah angka 0 atau 1. Ini adalah satuan yang paling sederhana. Memori juga dinyatakan dalam byte (1byte=8bit). Kumpulan byte dinyatakan dalam *word*. Panjang word yang umum adalah 8,16,dan 32 bit.

Tabel 8.3 Tingkatan satuan memori

| Symbol | | Numberofbytes | |
|-----------|----|---------------|-----------------------|
| Kilobytes | Kb | 2e10 | 1024 |
| Megabyte | Mb | 2e20 | 1,048,576 |
| Gigabyte | Gb | 2e30 | 1,073,741,824 |
| Terabyte | Tb | 2e40 | 1,099,511,627,7 76 |

8.5 Memori Utama Semikonduktor

Pada komputer lama, bentuk umum random access memory untuk memori utama adalah sebuah piringan ferro magnetic berlubang yang dikenal sebagai *core*, istilah yang tetap dipertahankan hingga saat ini.

8.5.1 Jenis Memori Random Akses

Semua jenis memori yang dibahas pada bagian ini adalah berjenis random akses, yaitu data secara langsung diakses melalui logic pengalamatan *wired-in*. Tabel 8.4 adalah daftar jenis memori semikonduktor utama.

Hal yang membedakan karakteristik RAM (*Random Access Memory*) adalah dimungkinkannya pembacaan dan penulisan data kememori secara cepat dan mudah. Aspek lain adalah RAM bersifat *volatile*, sehingga RAM hanya menyimpan data sementara. Teknologi yang berkembang saat ini adalah statik dan dinamik. *RAM dinamik* disusun oleh sel–sel yang menyimpan data sebagai muatan listrik pada kapasitor. Karena kapasitor memiliki kecenderungan alami untuk mengosongkan muatan, maka RAM dinamik memerlukan pengisian muatan listrik secara periodik untuk memelihara penyimpanan data. Pada *RAM statik*, nilai biner disimpan dengan menggunakan konfigurasi gate logika flip flop tradisional. RAM statikan menyimpan data selama ada daya listriknya.

RAM statik maupun dinamik adalah *volatile*, tetapi RAM dinamik lebih sederhana dan rapat sehingga lebih murah. RAM dinamik lebih cocok untuk kapasitas memori besar, namun RAM statik umumnya lebih cepat.

Read only memory (ROM) sangat berbeda dengan RAM, seperti namanya, ROM berisi pola data permanen yang tidak dapat diubah. Data yang tidak bias diubah menimbulkan keuntungan dan juga kerugian. Keuntungannya untuk data yang permanen dan sering digunakan pada sistem operasi maupun sistem perangkat ke rasakan aman diletakkan dalam ROM. Kerugiannya apabila ada kesalahan data atau adanya perubahan data sehingga perlu penyisipan–penyisipan.

Kerugian tersebut bisa diantisipasi dengan jenis *programmable ROM*, disingkat PROM. ROM dan PROM bersifat *non-volatile*. Proses penulisan PROM secara elektrik dengan peralatan khusus.

Variasi ROM lainnya adalah *read mostly memory*, yang sangat berguna untuk aplikasi operasi pembacaan jauh lebih sering daripada operasi penulisan. Terdapat tiga macam jenis, yaitu : EPROM, EEPROM dan *flash memory*.

EEPROM (*electrically erasable programmable read only memory*) merupakan memori yang dapat ditulisi kapan saja tanpa menghapus isi

sebelumnya. EEPROM menggabungkan kelebihan *non-volatile* dengan fleksibilitas dapat di-update.

Bentuk memori semikonduktor terbaru adalah *flash memory*. Memori ini dikenalkan tahun 1980-an dengan keunggulan pada kecepatan penulisan programnya. *Flash memory* menggunakan teknologi penghapusan dan penulisan elektrik. Seperti halnya EPROM, *flash Memory* hanya membutuhkan sebuah transistor per byte sehingga dapat diperoleh kepadatan tinggi.

4.5.2 Pengemasan (Packging)

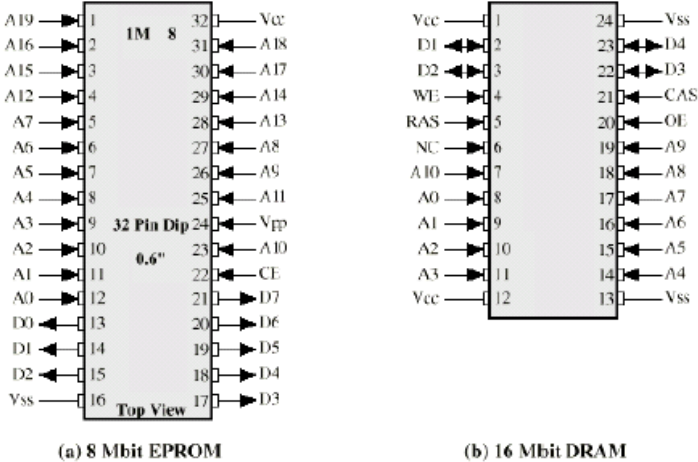
Gambar 8.3a menunjukkan sebuah contoh kemasan EPROM, yang merupakan keeping 8Mbit yang diorganisasi sebagai 1Mx8. Dalam kasus ini, organisasi dianggap sebagai kemasan satu word perkeeping. Kemasan terdiri dari 32 pin, yang merupakan salah satu ukuran kemasan keeping standar. Pin – pin tersebut mendukung saluran – saluran sinyal berikut ini :

- Alamat word yang sedang diakses. Untuk 1 Mword, diperlukan sejumlah 20 buah ($2^{20}= 1M$).
- Data yang akan dibaca, terdiri dari 8 saluran (D0–D7)
- Catu daya keeping adalah Vcc
- Pin grounding Vss
- Pinchipenable(CE). Karena mungkin terdapat lebih dari satu keeping memori yang terhubung pada bus yang sama maka pinCE digunakan untuk mengindikasikan valid atau tidaknya

pinini. PinCE diaktifkan oleh logic yang terhubung dengan bit berorde tinggi bus alamat(diatas A19)

- Tegangan program (Vpp).

Konfigurasi pin DRAM yang umum ditunjukkan gambar 8.3b, untuk keping 16 Mbit yang diorganisasikan sebagai 4Mx4. Terdapat sejumlah perbedaan dengan keping ROM, karena ada operasi tulis maka pin-pin data merupakan input/output yang dikendalikan oleh WE (*wriiteenable*) dan OE (*outputenable*).



Gambar 8.3 Pin dan sinyal kemasan memori



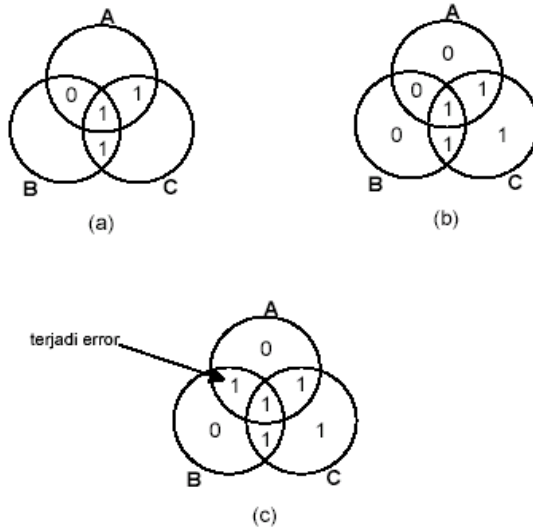
Gambar 8.4 Packging SIMM

4.5.3 Koreksi Error

Dalam melaksanakan fungsi penyimpanan, memori semikonduktor dimungkinkan mengalami kesalahan. Baik kesalahan berat yang biasanya merupakan kerusakan fisik memori maupun kesalahan ringan yang berhubungan data yang disimpan. Kesalahan ringan dapat dikoreksi kembali. Untuk mengadakan koreksi kesalahan

data yang disimpan diperlukan dua mekanisme, yaitu *mekanisme pendeteksian kesalahan* dan *mekanisme perbaikan kesalahan*.

Mekanisme pendeteksian kesalahan dengan menambahkan data word (D) dengan suatu kode, biasanya bit cek paritas (C). Sehingga data yang disimpan memiliki panjang D+C. Kesalahan akan diketahui dengan menganalisa data dan bit paritas tersebut. Mekanisme perbaikan kesalahan yang paling sederhana adalah *kode Hamming*. Metode ini diciptakan Richard Hamming di Bell Lab pada tahun 1950.



Gambar 8.5 Koreksi kesalahan dengan kode Hamming

Perhatikan gambar 8.5, disajikan tiga lingkaran Venn (A,B,C) saling berpotongan sehingga terdapat 7 ruang. Metode diatas adalah koreksi kesalahan untuk word data 4bit (D=4). Gambar 8.5a adalah data aslinya. Kemudian setiap lingkaran harus diset bit logika 1 berjumlah genap sehingga harus ditambah bit-bit paritas pada ruang yang kosong seperti gambar 8.5b. Apabila ada kesalahan penulisan bit pada data seperti gambar 8.5c akan dapat diketahui karena lingkaran A dan B memiliki logika 1 berjumlah ganjil.

Lalu bagaimana dengan word lebih dari 4 bit ? Ada cara yang mudah yang akan diterangkan berikut. Sebelumnya perlu diketahui jumlah bit paritas yang harus ditambahkan untuk sejumlah bit word. Contoh sebelumnya adalah koreksi kesalahan untuk kesalahan tunggal yang sering disebut *single error correcting* (SEC). Jumlah bit paritas yang harus ditambahkan lain pada *double error correcting* (DEC). Tabel 8.5

menyajikan jumlah bit paritas yang harus ditambahkan dalam sistem kode Hamming.

Tabel 8.5 Penambahan bit cek paritas untuk koreksi kode Hamming

| # Data Bits | # BitParitas SEC | # BitParitas DEC |
|-------------|------------------|------------------|
| 8 | 4 | 5 |
| 16 | 5 | 6 |
| 32 | 6 | 7 |
| 64 | 7 | 8 |
| 128 | 8 | 9 |
| 512 | 9 | 10 |

Mekanisme koreksi kesalahan akan meningkatkan realibitas bagi memori tetapi resikonya adalah menambah kompleksitas pengolahan data. Disamping itu mekanisme koreksi kesalahan akan menambah kapasitas memori karena adanya penambahan bit-bit cek paritas. Jadi ukuran memori akan lebih besar beberapa persen atau dengan kata lain kapasitas penyimpanan akan berkurang karena beberapa lokasi digunakan untuk mekanisme koreksi kesalahan.

Contoh koreksi kode Hamming 8 bit data :

Dari tabel 8.5 untuk 8 bit data diperlukan 4 bit tambahan sehingga panjang seluruhnya adalah 12 bit. Layout bit disajikan dibawah ini:

| Posisi Bit | Posisi Angka | Cek Bit | Data Bit |
|------------|-----------------|---------|----------|
| 12 | 1 1 0 0 | | D8 |
| 11 | 1 0 1 1 | | D7 |
| 10 | 1 0 1 0 | | D6 |
| 9 | 1 0 0 1 | | D5 |
| 8 | 1 0 0 0 | C4 | |
| 7 | 0 1 1 1 | | D4 |
| 6 | 0 1 1 0 | | D3 |
| 5 | 0 1 0 1 | | D2 |
| 4 | 0 1 0 0 | C3 | |
| 3 | 0 0 1 1 | | D1 |

Bit cek paritas ditempatkan dengan perumusan 2^N dimana $N = 0,1,2,\dots$, sedangkan bit data adalah sisanya. Kemudian dengan exclusive-OR dijumlahkan sebagai berikut:

$$C1 = D1 \oplus D2 \oplus D4 \oplus D5 \oplus D7$$

$$C2 = D1 \oplus D3 \oplus D4 \oplus D6 \oplus D7$$

$$C4 = D2 \oplus D3 \oplus D4 \oplus D8$$

$$C8 = D5 \oplus D6 \oplus D7 \oplus D8$$

Setiap cek bit (C) beroperasi pada setiap posisi bit data yang nomor posisinya berisi bilangan 1 pada kolomnya.

Sekarang ambil contoh suatu data, misalnya masukkan data: 00111001 kemudian ganti bit data ke 3 dari 0 menjadi 1 sebagai error-nya. Bagaimanakah cara mendapatkan bit data ke 3 sebagai bit yang terdapat error ?

Jawab:

Masukkan data pada perumusan cek bit paritas :

$$C1 = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1$$

$$C2 = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1$$

$$C4 = 0 \oplus 0 \oplus 1 \oplus 0 = 1$$

$$C8 = 1 \oplus 1 \oplus 0 \oplus 0 = 0$$

Sekarang bit 3 mengalami kesalahan sehingga data menjadi : 00111101

| | | |
|--------------|--|--|
| | $C1 = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1$ | |
| Organisasi & | $C2 = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 0 = 0$ | |
| | $C4 = 0 \oplus 1 \oplus 1 \oplus 0 = 0$ | |
| | $C8 = 1 \oplus 1 \oplus 0 \oplus 0 = 0$ | |

Apabila bit-bit cek dibandingkan antara yang lama dan baru maka terbentuk *syndrom word* :

| | | | | | |
|---|----|----|----|----|-----|
| | C8 | C4 | C2 | C1 | |
| | 0 | 1 | 1 | 1 | |
| ⊕ | 0 | 0 | 0 | 1 | |
| | | | | | |
| | 0 | 1 | 1 | 0 | = 6 |

Sekarang kita lihat posisi bit ke-6 adalah data ke-3.

Mekanisme koreksi kesalahan akan meningkatkan realibilitas bagi memori tetapi resikonya adalah menambah kompleksitas pengolahan data. Disamping itu mekanisme koreksi kesalahan akan menambah kapasitas memori karena adanya penambahan bit-bit cek paritas. Jadi ukuran memori akan lebih besar beberapa persen atau dengan kata lain kapasitas penyimpanan akan berkurang karena beberapa lokasi digunakan untuk mekanisme koreksi kesalahan.

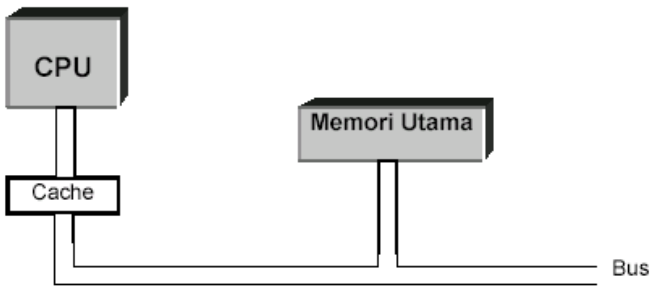
4.6 Cache Memori

Cache memori difungsikan mempercepat kerja memori sehingga mendekati kecepatan prosesor. Konsepnya dijelaskan pada gambar 8.6 dan gambar 8.7. Dalam organisasi komputer, memori utama lebih besar kapasitasnya namun lambat operasinya, sedangkan cache memori berukuran kecil namun lebih cepat. Cache memori berisi salinan memori utama.

Pada saat CPU membaca sebuah word memori, maka dilakukan pemeriksaan untuk mengetahui apakah word tersebut berada dalam cache memori. Bila ada dalam cache memori maka dilakukan pengiriman ke CPU, bila tidak dijumpai maka dicari dalam

Memori Utama

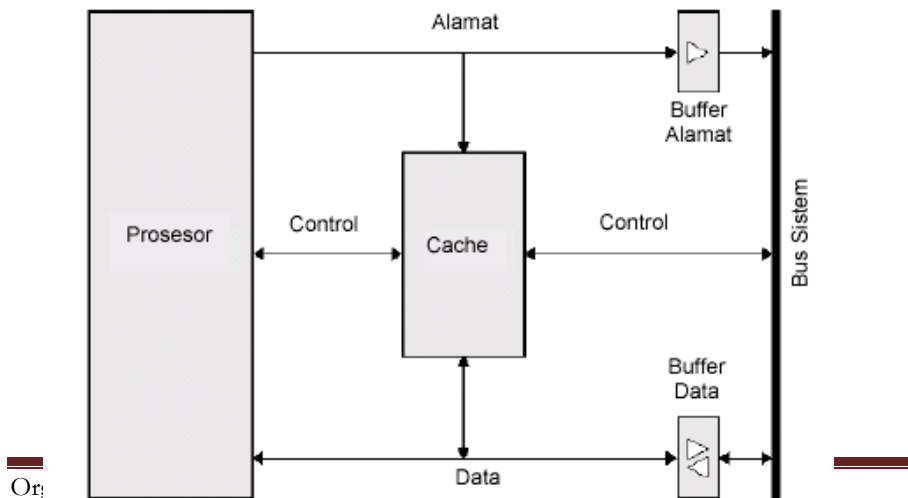
memori utama, selanjutnya blok yang berisi sejumlah word tersebut dikirim ke cache memori dan word yang diminta CPU dikirimkan ke CPU dari cache memori. Karena fenomena lokalitas referensi, ketika blok data diberikan kedalam cache memori, terdapat kemungkinan bahwa word-word berikutnya yang berada dalam satu blok akan diakses oleh CPU. Konsep ini menjadikan kinerja memori lebih baik.



Gambar 8.6 Hubungan cache memori

Sehingga dapat disimpulkan bahwa kerja cache adalah antisipasi terhadap permintaan data memori yang akan digunakan CPU. Apabila data diambil langsung dari memori utama bahkan memori eksternal akan memakan waktu lama yang menyebabkan status tunggu pada prosesor.

Ukuran cache memori adalah kecil, semakin besar kapasitasnya maka akan memperlambat proses operasi cache memori itu sendiri, disamping harga cache memori yang sangat mahal.



Gambar 8.7 Organisasi cache memori

4.7 Elemen Rancangan

Walaupun terdapat banyak implementasi cache, namun dari sisi organisasi maupun arsitekturnya tidak banyak macamnya.

Tabel 8.6 Unsur– unsur rancangan cache memori

| Unsur | Macam |
|---------------------|--|
| Kapasitas | - |
| Ukuran blok | - |
| Mapping | <ol style="list-style-type: none"> 1. <i>Direct Mapping</i> 2. <i>Assosiative Mapping</i> 3. <i>Set Assosiative Mapping</i> |
| Algoritma pengganti | <ol style="list-style-type: none"> 1. <i>Leastrecently used (LRU)</i> 2. <i>Firstin firstout (FIFO)</i> 3. <i>Leastf requently used (LFU)</i> 4. <i>Random</i> |
| Write Policy | <ol style="list-style-type: none"> 1. Write Through 2. Write Back 3. Write Once |
| Jumlah Cache | <ol style="list-style-type: none"> 1. Singe atau dua level 2. Unified atau split |

4.7.1 Kapasitas Cache

Menentukan ukuran memori cache sangatlah penting untuk mendongkrak kinerja komputer. Dari segi harga cache sangatlah mahal tidak seperti memori utama. Semakin besar kapasitas cache tidak berarti semakin cepat prosesnya, dengan ukuran besar akan terlalu banyak gate pengalamatannya sehingga akan memperlambat proses.

Kita bisa melihat beberapa mereka prosesor dipasaran beberapa waktu lalu. AMD mengeluarkan prosesor K5 dan K6 dengan cache yang besar (1MB) tetapi kinerjanya tidak bagus. Kemudian Intel pernah mengeluarkan prosesor tanpa cache untuk alasan harga yang murah, yaitu seri Intel Celeron pada tahun 1998-an hasil kinerjanya sangat buruk terutama untuk operasi data besar, floating point, 3D. Intel Celeron versi berikutnya sudah ditambah cache sekitar 128 KB.

Lalu berapa idealnya kapasitas cache ? Sejumlah penelitian telah menganjurkan bahwa ukuran cache antara 1KB dan 512KB akan lebih optimum.

4.7.2 Ukuran Blok

Elemen rancangan yang harus diperhatikan lagi adalah ukuran blok. Telah dijelaskan adanya sifat lokalitas referensi maka nilai ukuran blok sangatlah penting. Apabila blok berukuran besar ditransfer ke cache akan menyebabkan hit ratio mengalami penurunan karena banyaknya data yang dikirim disekitar referensi. Tetapi apabila terlalu kecil, dimungkinkan memori yang akan dibutuhkan CPU tidak tercakup. Apabila blok berukuran besar ditransfer ke cache, maka akan terjadi:

1. Blok-blok yang berukuran lebih besar mengurangi jumlah blok yang menempati cache. Karena isi cache sebelumnya akan ditindih.
2. Dengan meningkatnya ukuran blok maka jarak setiap word tambahan menjadi lebih jauh dari word yang diminta, sehingga menjadi lebih kecil kemungkinannya digunakan cepat.

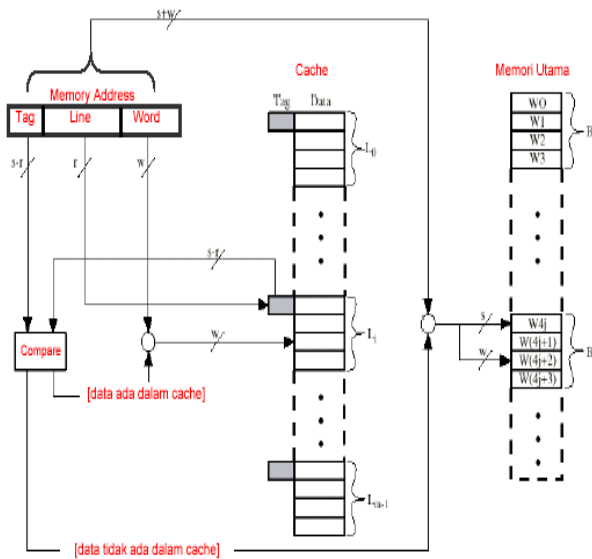
Hubungan antara ukuran blok dan hit ratio sangat rumit untuk dirumuskan, tergantung pada karakteristik lokalitas programnya dan tidak terdapat nilai optimum yang pasti telah ditemukan. Ukuran antara 4 hingga 8 satuan yang dapat dialamati (word atau byte) cukup beralasan untuk mendekati nilai optimum.

4.7.3 Fungsi Pemetaan (Mapping)

Telah kita ketahui bahwa cache mempunyai kapasitas yang kecil dibandingkan memori utama. Sehingga diperlukan aturan blok-blok mana yang diletakkan dalam cache. Terdapat tiga metode, yaitu pemetaan langsung, pemetaan asosiatif, dan pemetaan asosiatif set.

Pemetaan Langsung

Pemetaan langsung adalah teknik yang paling sederhana, yaitu teknik ini memetakan blok memori utama hanya ke sebuah saluran cache saja. Gambar 8.8 menjelaskan mekanisme pemetaan langsung.



Gambar 8.8. Organisasi *Cache* pemetaan langsung

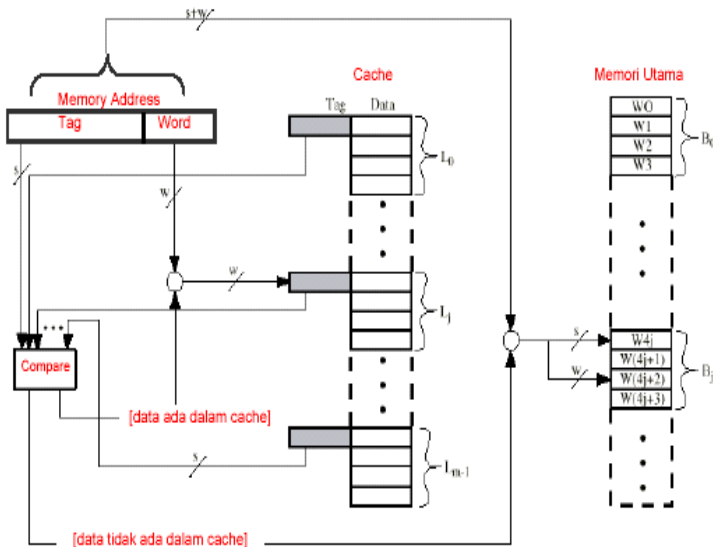
Jadi dalam metode ini pemetaan adalah bagian alamat blok memori utama sebagai nomor saluran cache. Ketika suatu blok data sedang diakses atau dibaca terhadap saluran yang diberikan, maka perlu memberikan tag bagi data untuk membedakannya dengan blok-blok lain yang dapat sesuai dengan saluran tersebut.

Teknik pemetaan ini sederhana dan mudah diimplementasikan, namun kelemahannya adalah terdapat lokasi cache yang tetap bagi sembarang blok-blok yang diketahui. Dengan demikian, apabila suatu program berulang-ulang melakukan word referensi dari dua blok yang berbeda memetakan saluran yang sama maka blok-blok itu secara terus-menerus akan di-swapke dalam cache sehingga hitrasinya akan rendah.

Pemetaan Asosiatif

Pemetaan asosiatif mengatasi kekurangan pemetaan langsung dengan cara setiap blok memori utama dapat dimuat kesembarang saluran cache. Alamat memori utama diinterpretasikan dalam field tag dan field word oleh control logika cache. Tag secara unik mengidentifikasi sebuah blok memori utama.

Mekanisme untuk mengetahui suatu blok dalam cache dengan memeriksa setiap tag saluran cache oleh control logika cache. Dengan pemetaan ini didapat fleksibilitas dalam penggantian blok baru yang ditempatkan dalam cache. Algoritma penggantian dirancang untuk memaksimalkan hit ratio, yang pada pemetaan langsung terdapat kelemahan dalam bagian ini. Kekurangan pemetaan asosiatif adalah kompleksitas rangkaian sehingga mahal secara ekonomi.



Gambar 8.9. Organisasi Cache dengan pemetaan asosiatif

8.7.4 Algoritma Penggantian

Algoritma Penggantian adalah suatu mekanisme pergantian blok-blok dalam memori cache yang lama dengan data baru. Dalam pemetaan langsung tidak diperlukan algoritma ini, namun dalam pemetaan asosiatif dan asosiatif set, algoritma ini mempunyai peranan penting untuk meningkatkan kinerja cache memori.

Banyak algoritma penggantian yang telah dikembangkan, namun dalam buku ini akan dijelaskan algoritma yang umum digunakan saja. Algoritma yang paling efektif adalah *Least Recently Used* (LRU), yaitu mengganti blok data yang terlama berada dalam cache dan tidak memiliki referensi. Algoritma lainnya adalah *First In First Out* (FIFO), yaitu mengganti blok data yang awal masuk. Kemudian *Least Frequently Used* (LFU) adalah mengganti blok data yang mempunyai referensi paling sedikit. Teknik lain adalah algoritma *Random*, yaitu penggantian tidak berdasarkan pemakaian datanya, melainkan berdasar slot dari beberapa slot kandidat secara acak.

8.7.5. Write Policy

Apabila suatu data telah diletakkan pada cache maka sebelum ada penggantian harus dicek apakah data tersebut telah mengalami perubahan. Apabila telah berubah maka data pada memori utama harus di-update. Masalah penulisan ini sangat kompleks, apalagi memori utama. Dapat diakses langsung oleh modul I/O, yang memungkinkan data pada memori utama berubah, lalu bagaimana dengan data yang telah dikirim pada cache? Tentunya perbedaan ini menjadikan data tidak valid.

Teknik yang dikenalkan diantaranya, *writethrough*, yaitu operasi penulisan melibatkan data pada memori utama dan sekaligus pada cache memori sehingga data selalu valid. Kekurangan teknik ini adalah menjadikan lalulintas data ke memori utama dan cache sangat tinggi sehingga mengurangi kinerja sistem, bahkan bisa terjadi hang.

Teknik lainnya adalah *writeback*, yaitu teknik meminimasi penulisan dengan cara penulisan pada cache saja. Pada saat akan terjadi penggantian blok data cache maka baru di adakan penulisan pada memori utama. Masalah yang timbul adalah mana kala data dimemori utama belum di-update telah diakses modul I/O sehingga data dimemori utama tidak valid.

Penggunaan multi cache terutama untuk multiprosesora dan menjumpai masalah yang lebih kompleks. Masalah validasi data tidak hanya antara cache dan memori utama saja, namun antar cache juga

harus diperhatikan. Pendekatan penyelesaian masalah yang dapat dilakukan adalah dengan :

- *Bus Watchingwith Write Through*, yaitu setiap cache controller akan memonitoring bus alamat untuk mendeteksi adanya operasi tulis. Apabila ada operasi tulis dialamat yang datanya digunakan bersama maka cache controller akan menginvalidasi data cache-nya.
- *Hardware Transparency*, yaitu adanya perangkat keras tambahan yang menjamin semua updating data memori utama melalui cache direfleksikan pada seluruh cache yang ada.
- *Non Cacheable Memory*, yaitu hanya bagian memori utama tertentu yang digunakan secara bersama. Apabila ada mengakses data yang tidak dishare merupakan kegagalan cache.

8.7.6 Jumlah Cache

Terdapat dua macam letak cache. Berada dalam keeping prosesor yang disebut *onchipcache* atau cache internal. Kemudian berada diluar chipprosesor yang disebut *offchipcache* atau cache eksternal.

Cache internal diletakkan dalam prosesor sehingga tidak memerlukan bus eksternal, akibatnya waktu aksesnya akan cepat sekali, apalagi panjang lintasan internal bus prosesor sangat pendek untuk mengakses cache internal. Cache internal selanjutnya disebut *cache tingkat 1 (L1)*.

Cache eksternal berada diluar keeping chipprosesor yang diakses melalui bus eksternal. Pertanyaannya, apakah masih diperlukan cache eksternal apabila telah ada cache internal ? Dari pengalaman, masih diperlukan untuk mengantisipasi permintaan akses alamat yang belum tercakup dalam cache internal. Cache eksternal selanjutnya disebut *cache tingkat 2 (L2)*.

Selanjutnya terdapat perkembangan untuk memisah cache data dan cache instruksi yang disebut *unified cache*. Keuntungan *unified cache* adalah :

- *Unified cache* memiliki hit rate yang tinggi karena telah dibedakan antara informasi data dan informasi instruksi.
- Hanya sebuah cache saja yang perlu dirancang dan diimplementasikan.

Namun terdapat kecenderungan untuk menggunakan *split cache*, terutama pada mesin–mesin super scalar seperti Pentium dan Power PC yang menekankan pada parallel proses dan perkiraan – perkiraan eksekusi yang akan terjadi. Kelebihan utama *splitcache* adalah mengurangi persaingan antara prosesor instruksi dan unit eksekusi untuk

mendapatkan cache, yang mana hal ini sangat utama bagi perancangan prosesor– prosesor *pipelining*.

Rangkuman

1. Memori internal adalah memori yang dapat diakses langsung oleh prosesor, contohnya : register, cache memori, RAM, ROM. Sedangkan memori eksternal adalah memori yang diakses prosesor melalui piranti I/O, seperti flashdisk, harddisk, dan sebagainya.
2. Sel memori memiliki sifat-sifat tertentu yaitu : sel memori memiliki dua keadaan stabil, mempunyai kemampuan untuk ditulisi, dan mempunyai kemampuan untuk dibaca.
3. Ada beberapa macam karakteristik penting sistem memori computer, yaitu : berdasarkan lokasi, kapasitas, satuan transfer, metode akses, kinerja, tipe fisik, karakteristik fisik.
4. Cache memori berfungsi mempercepat kerja memori sehingga mendekati kecepatan prosesor. Ukuran cache memori adalah kecil, semakin besar kapasitasnya maka akan memperlambat proses operasi cache memori itu sendiri.
5. Terdapat dua macam letak cache, yaitu cache internal diletakkan dalam prosesor sehingga tidak memerlukan bus eksternal, dan cache eksternal yang berada diluar keeping chip prosesor yang diakses melalui bus eksternal.
6. Elemen rancangan cache memori meliputi unsur-unsur : *mapping* (pemetaan), algoritma pengganti, *write policy*, dan jumlah cache.

Tes Formatif

1. Jelaskan perbedaan antara memori internal dengan memori eksternal!
2. Jelaskan sifat-sifat tertentu dari sel memori !
3. Jelaskan karakteristik sistem memori berdasarkan metode Direct Access !
4. Jelaskan karakteristik sistem memori berdasarkan unjuk kerjanya !
5. Jelaskan perbedaan antara cache internal dengan cache eksternal !

SATUAN BAHASAN 9

MEMORI SEKUNDER

1. Gambaran Singkat Mengenai Materi Kuliah

Materi kuliah ini akan membahas mengenai memori sekunder atau memori eksternal. Dalam materi ini akan dijelaskan tentang jenis-jenis memori eksternal berdasarkan kelompoknya seperti magnetic disk, RAID, optical disk, dan pita magnetik.

2. Pedoman Mempelajari Materi

Baca dengan cermat dan teliti materi pada satuan bahasan 9, kemudian pahami setiap bagian dari memori sekunder. Hal ini akan membantu anda dalam memahami karakteristik berbagai macam memori sekunder. Pada bagian akhir terdapat tes formatif yang perlu anda kerjakan sebagai sarana untuk mengukur seberapa jauh tingkat pemahaman anda terhadap materi ini.

3. Tujuan Pembelajaran

Setelah mempelajari satuan bahasan 9 ini, mahasiswa diharapkan dapat :

- engetahui jenis-jenis magnetic disk
- engetahui istilah-istilah dalam magnetic disk
- enjelaskan mekanisme pembacaan dan penulisan pada disk
- enjelaskan berbagai karakter magnetic disk
- enjelaskan mengenai RAID
- enjelaskan jenis-jenis optical disk
- enjelaskan mengenai pita magnetik

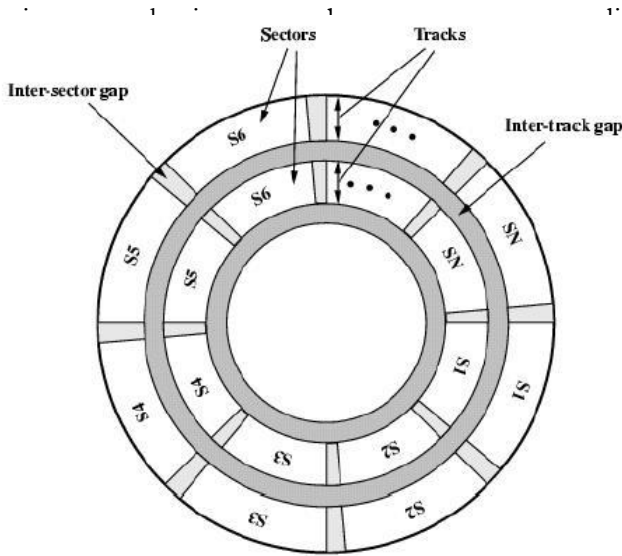
9.1 Magnetik Disk

Disk adalah piringan bundar yang terbuat dari bahan tertentu (logam atau plastik) dengan permukaan dilapisi bahan yang dapat dimagnetisasi. Mekanisme membaca/tulis menggunakan kepala baca atau tulis yang disebut *head*, merupakan kompartemen pengkonduksi (*conductingcoil*). Desain fisiknya, head bersifat stasioner sedangkan piringan disk berputar sesuai kontrolnya.

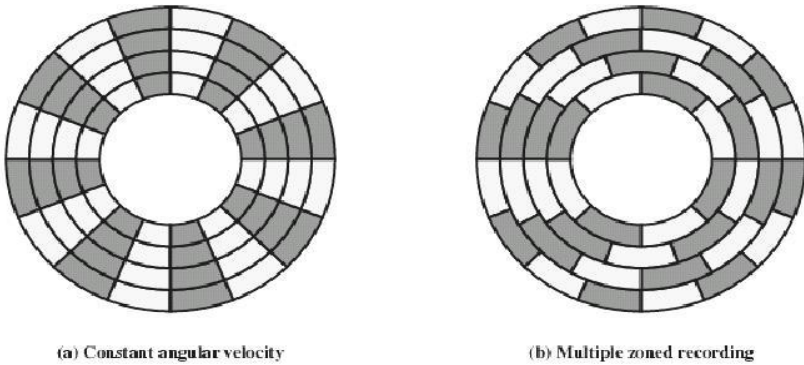
Layout data pada disk diperlihatkan pada gambar 9.1 dan gambar 9.2. Terdapat dua metode *layout* data pada disk, yaitu *constant angular velocity* dan *multiple zoned recording*. Disk diorganisasi dalam bentuk cincin-cincin konsentris yang disebut *track*. Tiap *track* pada disk dipisahkan oleh *gap*. Fungsi *gap* untuk mencegah atau mengurangi kesalahan pembacaan maupun penulisan yang disebabkan melesetnya head atau karena interferensi magnet.

Sejumlah bit yang sama akan menempati *track-track* yang tersedia. Semakin kedalam disk maka kerapatan (*density*) disk akan bertambah besar. Data dikirim ke memori ini dalam bentuk blok, umumnya blok lebih kecil kapasitasnya dari pada *track*. Blok-blok data disimpan dalam disk yang berukuran blok, yang disebut *sector*. Sehingga *track* biasanya terisi beberapa *sector*, umumnya 10 hingga 100 *sector* tiap *track*-nya.

disk? ... pada
sector ... posisi
data ... reader
data. ... suatu
drive s ... 1 disk



Gambar 9.1 *Layout* data disk

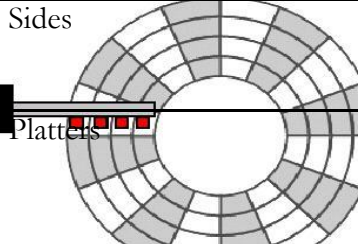
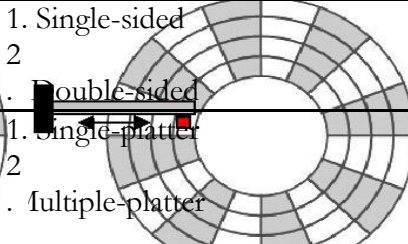


Gambar 9.2. Metode *Layout* data disk

Karakteristik Magnetik Disk

Ada berbagai macam karakteristik magnetic disk. Tabel 9.1 menyajikan daftar karakteristik utama dari magnetic disk.

Tabel 9.1. Karakteristik Magnetic Disc

| Karakteristik | Macam |
|--|--|
| Gerakan head | 1. Fixed head (satu per track) 2. Movable head (satu per surface) |
| Portabilitas disk | 1. Nonremovable disk 2. Removable disk |
| Sides  Platters (a) Fixed head |  Double-sided 1. Single-platter 2. Multiple-platter |
| Mekanisme head | 1. Contact (floppy) 2. Fixed gap 3. Aerodynamic gap (Winchester) |

Berdasarkan gerakan head, terdapat dua macam jenis yaitu head tetap (*fixed head*) dan head bergerak (*movable head*) seperti terlihat pada gambar 9.3. Pada head tetap setiap track memiliki kepala head sendiri, sedangkan pada head bergerak, satu kepala head digunakan untuk beberapa track dalam satu muka disk. Mekanisme dalam head bergerak adalah lengan head bergerak menuju track yang diinginkan berdasarkan perintah dari disk drive-nya.

Gambar 9.3. Macam disk berdasarkan gerakan head

Karakteristik disk berdasar portabilitasnya dibagi menjadi disk yang tetap (*non-removable disk*) dan disk yang dapat dipindah (*removable disk*). Keuntungan disk yang dapat dipindah atau diganti – ganti adalah tidak terbatas dengan kapasitas disk dan lebih fleksibel.

Karakteristik lainnya berdasar *sides* atau muka sisinya adalah satu sisi disk (*single sides*) dan dua muka disk (*double sides*). Kemudian berdasarkan jumlah piringannya (*platters*), dibagi menjadi satu piringan (*single platter*) dan banyak piringan (*multiple platter*). Gambar disk dengan multiple platters tersaji dalam gambar 9.4.

Terakhir, mekanisme head membagi disk menjadi tiga macam, yaitu head yang menyentuh disk (*contact*) seperti pada floppy disk, head yang mempunyai celah utara tetap maupun yang tidak tetap tergantung medan magnetnya. Celah atau jarak head dengan disk tergantung kepadatan datanya, semakin padat datanya dibutuhkan jarak head yang semakin dekat, namun semakin dekat head maka faktor resikonya semakin besar, yaitu terjadinya kesalahan baca. Teknologi Winchester dari IBM mengantisipasi masalah celah head diatas dengan model head aerodinamik. Head berbentuk lembaran timah yang berada dipermukaan disk apabila tidak bergerak, seiring perputaran disk maka disk akan mengangkat headnya.

Istilah Winchester dikenalkan IBM pada model disk 3340-nya. Model ini merupakan removable disk pack dengan head yang dibungkus di dalam pack. Sekarang istilah Winchester digunakan oleh sembarang disk drive yang dibungkus pack dan memakai rancangan head aerodinamis.

Gambar 9.4 Disk piringan banyak (*multiple platters disk*)

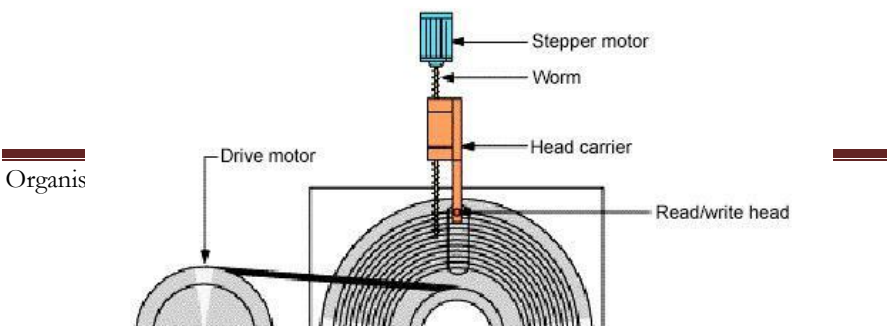
Disk drive beroperasi dengan kecepatan konstan. Untuk dapat membaca dan menulis, head harus berada pada track yang diinginkan dan pada awal sektornya. Diperlukan waktu untuk mencapai track yang diinginkan, waktu yang diperlukan disebut sebagai *seek time*. Apabila track sudah didapatkan maka diperlukan waktu sampai sector yang bersangkutan berputar sesuai dengan headnya, yang disebut *rotational latency*. Jumlah seek time dan rotational latency disebut dengan *access time*. Dengan kata lain, access time adalah waktu yang diperlukan disk untuk berada pada posisi siap membaca atau menulis.

Berikutnya akan dijelaskan memori eksternal yang termasuk magnetik disk, yaitu floppy disk (disket), harddisk model IDE dan harddisk model SCSI.

Floppy Disk (Disket)

Dengan berkembangnya komputer pribadi maka diperlukan media untuk mendistribusikan software maupun pertukaran data. Solusinya ditemukannya *disket* atau *floppy disk* oleh IBM.

Karakteristik disket adalah head menyentuh permukaan disk saat membaca ataupun menulis. Hal ini menyebabkan disket tidak tahan lama dan sering rusak. Untuk mengurangi kerusakan atau aus pada disket, dibuat mekanisme penarikan head dan menghentikan rotasi disk ketika head tidak melakukan operasi baca dan tulis. Namun akibatnya waktu akses disket cukup lama. Gambar 9.5. memperlihatkan bentuk floppy disk.



Gambar 9.5. Floppy Disk

Tabel 9.2 Karakteristik berbagai macam disket

Ada dua ukuran disket yang tersedia, yaitu 5,25 inchi dan 3,5

| Parameter | LD 5,25" | HD 5,25" | LD 3,5" | HD 3,5" |
|------------------|----------|----------|---------|---------|
| Ukuran (inchies) | 5,25 | 5,25 | 3,5 | 3,5 |
| Kapasitas (byte) | 360K | 1,2M | 720K | 1,44M |
| Tracks | 40 | 80 | 80 | 80 |
| Sectors/track | 9 | 15 | 9 | 18 |
| Heads | 2 | 2 | 2 | 2 |
| Rotasi/min | 300 | 500 | 300 | 300 |
| Data rate (kbps) | 250 | 500 | 250 | 500 |
| Type | flexible | flexible | rigid | rigid |

inchi dengan masing – masing memiliki versi *low density* (LD) dan *high density* (HD). Saat ini disket sudah jarang digunakan sebagai memori sekunder. Perhatikan karakteristik model disket yang beredar saat ini pada tabel 9.2.

Integrated Drive Electronic Disk (Harddisk)

Saat IBM mengembangkan PC XT, menggunakan sebuah hardisk Seagate 10 MB untuk menyimpan program maupun data. Harddisk ini memiliki 4 head, 306 silinder dan 17 sektor per track, dicontrol oleh pengontrol disk Xebec pada sebuah kartu *plug-in*.

Teknologi yang berkembang pesat menjadikan pengontrol disk yang sebelumnya terpisah menjadi satu paket terintegrasi, diawali dengan teknologi drive IDE (*Integrated Drive Electronics*) pada tengah tahun 1980. Teknologi saat itu IDE hanya mampu menangani disk berkapasitas maksimal 528 MB dan mengontrol 2 disk.

Seiring kebutuhan memori, berkembang teknologi yang mampu menangani disk berkapasitas besar. IDE berkembang menjadi EIDE (*Extended Integrated Drive Electronics*) yang mampu menangani harddisk lebih dari 528 MB dan mendukung pengalamatan LBA (*Logical Block Addressing*), yaitu metode pangalamatan yang hanya memberi nomer pada sektor – sektor mulai dari 0 hingga maksimal 224-1. Metode ini mengharuskan pengontrol mampu mengkonversi alamat – alamat LBA menjadi alamat head, sektor dan silinder.

Peningkatan kinerja lainnya adalah kecepatan tranfer yang lebih tinggi, mampu mengontrol 4 disk, mampu mengontrol drive CD-ROM.

SCSI Disk (Harddisk)

Disk SCSI (*Small Computer System Interface*) mirip dengan IDE dalam hal organisasi pengalamatannya. Perbedaannya pada piranti antarmukanya yang mampu mentransfer data dalam kecepatan tinggi. Versi disk SCSI terlihat pada tabel 5.3.

Karena kecepatan transfernya tinggi, disk ini merupakan standar bagi komputer UNIX dari Sun Microsystem, HP, SGI, Machintos, Intel terutama komputer – komputer server jaringan, dan vendor – vendor lainnya.

SCSI sebenarnya lebih dari sekedar piranti antarmuka harddisk. SCSI adalah sebuah bus karena SCSI mampu sebagai pengontrol hingga 7 peralatan seperti: harddisk, CD ROM, rekorder CD, scanner dan peralatan lainnya. Masing – masing peralatan memiliki ID unik sebagai media pengenalan oleh SCSI.

Tabel 9.3 Versi disk SCSI

| Nama | Data bits | Bus MHz | MB/det |
|-------------------|-----------|---------|--------|
| SCSI-1 | 8 | 5 | 5 |
| Fast SCSI | 8 | 10 | 10 |
| Wide Fast SCSI | 16 | 10 | 20 |
| Ultra SCSI | 8 | 20 | 20 |
| Wide Ultra SCSI | 16 | 20 | 40 |
| Ultra-2 SCSI | 8 | 40 | 40 |
| Wide Ultra-2 SCSI | 16 | 40 | 80 |

5.2 Redundancy Array of Independent Disk (RAID)

Telah dijelaskan diawal bahwa masalah utama sistem memori adalah mengimbangi laju kecepatan CPU. Beberapa teknologi dicoba dan dikembangkan, diantaranya menggunakan konsep akses paralel pada disk.

RAID (*Redundancy Array of Independent Disk*) merupakan organisasi disk memori yang mampu menangani beberapa disk dengan sistem akses paralel dan redudansi ditambahkan untuk meningkatkan reliabilitas. Karena kerja paralel inilah dihasilkan resultan kecepatan

disk yang lebih cepat. Teknologi database sangatlah penting dalam model disk ini karena pengontrol disk harus mendistribusikan data pada sejumlah disk dan juga membacaan kembali. Karakteristik umum disk RAID :

- RAID adalah sekumpulan disk drive yang dianggap sebagai sistem tunggal disk.
- Data didistribusikan ke drive fisik array.
- Kapasitas redundan disk digunakan untuk menyimpan informasi paritas, yang menjamin recoveribility data ketika terjadi masalah atau kegagalan disk.

Jadi RAID merupakan salah satu jawaban masalah kesenjangan kecepatan disk memori dengan CPU dengan cara menggantikan disk berkapasitas besar dengan sejumlah disk – disk berkapasitas kecil dan mendistribusikan data pada disk – disk tersebut sedemikian rupa sehingga nantinya dapat dibaca kembali.

RAID tingkat 0

Sebenarnya bukan RAID karena tidak menggunakan redundansi dalam meningkatkan kinerjanya. Data didistribusikan pada seluruh disk secara array merupakan keuntungan daripada menggunakan satu disk berkapasitas besar.

Sejalan perkembangan RAID – 0 menjadi model data strip pada disk dengan suatu management tertentu hingga data sistem data dianggap tersimpan pada suatu disk logik. Mekanisme tranfer data dalam satu sektor sekaligus sehingga hanya baik untuk menangani tranfer data besar.

RAID tingkat 1

Pada RAID - 1, redundansi diperoleh dengan cara menduplikasi seluruh data pada disk *mirror*-nya. Seperti halnya RAID – 0, pada tingkat 1 juga menggunakan teknologi *stripping*, perbedaannya adalah dalam tingkat 1 setiap strip logik dipetakan ke dua disk yang secara logika terpisah sehingga setiap disk pada array akan memiliki *mirror disk* yang berisi data sama. Hal ini menjadikan RAID – 1 mahal. Keuntungan RAID – 1:

- Permintaan pembacaan dapat dilayani oleh salah satu disk karena terdapat dua disk berisi data sama, tergantung waktu akses yang tercepat.
- Permintaan penyimpanan atau penulisan dilakukan pada 2 disk secara paralel.
- Terdapat back-up data, yaitu dalam disk *mirror*-nya.

RAID – 1 mempunyai peningkatan kinerja sekitar dua kali lipat dibandingkan RAID – 0 pada operasi baca, namun untuk operasi tulis tidak secara signifikan terjadi peningkatan. Cocok digunakan untuk menangani data yang sering mengalami kegagalan dalam proses pembacaan. RAID – 1 masih bekerja berdasarkan sektor – sektornya.

RAID tingkat 2

RAID – 2 menggunakan teknik akses paralel untuk semua disk. Dalam proses operasinya, seluruh disk berpartisipasi dan mengeksekusi setiap permintaan sehingga terdapat mekanisme sinkronisasi perputaran disk dan headnya.

Teknologi *stripping* juga digunakan dalam tingkat ini, hanya stripnya berukuran kecil, sering kali dalam ukuran *word* atau *byte*. Koreksi kesalahan menggunakan sistem bit paritas dengan kode Hamming. Cocok digunakan untuk menangani sistem yang kerap mengalami kesalahan disk.

RAID tingkat 3

Diorganisasikan mirip dengan RAID – 2, perbedaannya pada RAID – 3 hanya membutuhkan disk redundan tunggal, tidak tergantung jumlah array disknya. Bit paritas dikomputasikan untuk setiap data word dan ditulis pada disk paritas khusus. Saat terjadi kegagalan drive, data disusun kembali dari sisa data yang masih baik dan dari informasi paritasnya.

RAID – 3 menggunakan akses paralel dengan data didistribusikan dalam bentuk strip – strip kecil. Kinerjanya menghasilkan transfer berkecepatan tinggi, namun hanya dapat mengeksekusi sebuah permintaan I/O saja sehingga kalau digunakan pada lingkungan transaksi data tinggi terjadi penurunan kinerja.

RAID tingkat 4

RAID – 4 menggunakan teknik akses yang independen untuk setiap disknya sehingga permintaan baca atau tulis dilayani secara paralel. RAID ini cocok untuk menangani sistem dengan kelajuan tranfer data yang tinggi. Tidak memerlukan sinkronisasi disk karena setiap disknya beroperasi secara independen. Stripping data dalam ukuran yang besar.

Strip paritas bit per bit dihitung ke seluruh strip yang berkaitan pada setiap disk data. Paritas disimpan pada disk paritas khusus. Saat operasi penulisan, array management software tidak hanya meng-update data tetapi juga paritas yang terkait. Keuntungannya dengan disk paritas yang khusus menjadikan keamanan data lebih terjamin, namun dengan disk paritas yang terpisah akan memperlambat kinerjanya.

RAID tingkat 5

Mempunyai kemiripan dengan RAID – 4 dalam organisasinya, perbedaannya adalah strip – strip paritas didistribusikan pada seluruh disk. Untuk keamanan, strip paritas suatu disk disimpan pada disk lainnya. RAID – 4 merupakan perbaikan dari RAID – 4 dalam hal peningkatan kinerjanya. Disk ini biasanya digunakan dalam server jaringan.

RAID tingkat 6

Merupakan teknologi RAID terbaru. Menggunakan metode penghitungan dua paritas untuk alasan keakuratan dan antisipasi terhadap koreksi kesalahan. Seperti halnya RAID – 5, paritas tersimpan pada disk lainnya. Memiliki kecepatan transfer yang tinggi.

5.3 Optical Disk

Pada tahun 1980, Philips dan Sony mengembangkan CD (*Compact Disk*). Detail teknis produk ini dipublikasikan dalam *international standard* resmi pada tahun 1983 yang populer disebut *red book*. CD merupakan disk yang tidak dapat dihapus, mampu menyimpan memori kurang lebih 60 menit informasi audio pada salah satu sisinya. Keberhasilan secara komersial CD yang mampu menyimpan data dalam jumlah yang besar, menjadikannya media

penyimpan yang fleksibel digunakan di berbagai peralatan seperti komputer, kamera video, MP3 player, dan lain-lain. Sejak dipublikasikan sampai dengan saat ini, terdapat bermacam-macam variasi sesuai dengan penggunaan dan teknologinya. Berikut tabel diantara produk-produk optical disk :

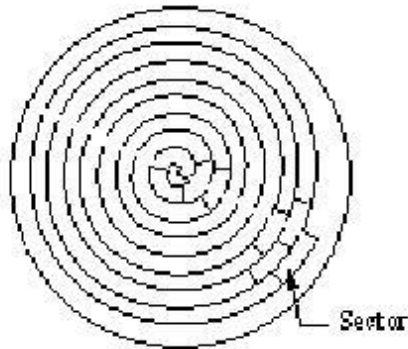
CD ROM

(*Compact Disk – Read Only Memory*). Merupakan generasi CD yang diaplikasikan sebagai media penyimpanan data komputer. Dikenalkan pertama kali oleh Phillips dan Sony tahun 1984 dalam publikasinya, yang dikenal dengan *Yellow Book*.

Perbedaan utama dengan CD adalah CD ROM player lebih kasar dan memiliki perangkat pengoreksi kesalahan, untuk menjamin keakuratan tranfer data ke komputer. Secara fisik keduanya dibuat dengan cara yang sama, yaitu terbuat dari resin, contohnya *polycarbonate*, dan dilapisi dengan permukaan yang sangat reflektif seperti aluminium.

Penulisan dengan cara membuat lubang mikroskopik sebagai representasi data dengan laser berintensitas tinggi. Pembacaan menggunakan laser berintensitas rendah untuk menterjemahkan lubang mikroskopik ke dalam bentuk data yang dapat dikenali komputer. Saat mengenai lubang mikroskopik, intensitas sinar laser akan berubah – ubah. Perubahan intensitas ini dideteksi oleh fotosensor dan dikonversi dalam bentuk sinyal digital.

Karena disk berbentuk lingkaran, terdapat masalah dalam mekanisme baca dan tulis, yaitu masalah kecepatan. Saat disk membaca data dibagian dekat pusat disk diperlukan putaran rendah karena padatnya informasi data, sedangkan apabila data berada di bagian luar disk diperlukan kecepatan yang lebih tinggi. Ada beberapa metode mengatasi masalah kecepatan ini, diantaranya dengan sistem *constant angular velocity (CAV)*, yaitu bit – bit informasi direkam dengan kerapatan yang bervariasi sehingga didapatkan putaran disk yang sama. Metode ini biasa diterapkan dalam disk magnetik, kelemahannya adalah kapasitas disk menjadi berkurang. Metode lain, yang biasa diterapkan pada disk optik adalah *constant linier velocity (CLV)*, yaitu dalam mengantisipasi kerapatan data pada disk dengan menyesuaikan kecepatan putaran disk yang dikontrol oleh *disk drive*-nya. Keuntungannya adalah kapasitas disk besar, namun waktu akses secara keseluruhan lebih lambat dibandingkan metode CAV. Layout disk CLV terlihat pada gambar 9.6.



Gambar 9.6 Layout disk CLV

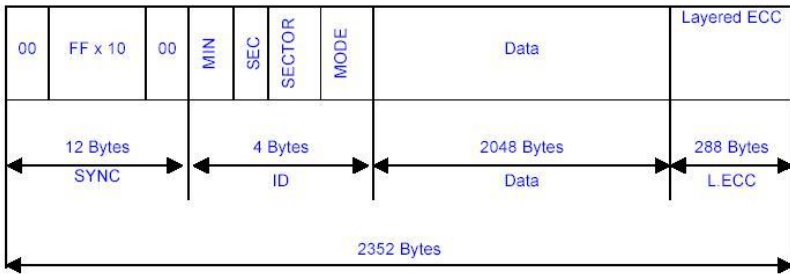
Data pada CD-ROM diorganisasikan sebagai sebuah rangkaian blok-blok. Formasi blok yang umum ditunjukkan pada gambar 9.7. Format ini terdiri dari field-field sebagai berikut :

- *Sync* : *Field sync* mengidentifikasi awal sebuah blok. *Field* ini terdiri dari sebuah byte yang seluruhnya nol, 10 byte yang seluruhnya satu, dan sebuah byte akhir yang seluruhnya nol.
- *Header* : Header terdiri dari alamat blok dan byte mode. Mode nol menandakan suatu field data blanko; mode satu menandakan penggunaan kode error-correcting dan 2048 byte data; mode dua menandakan 2336 byte data pengguna tanpa kode error-correcting.
- *Data* : Data pengguna
- *Auxiliary* : Data pengguna tambahan dalam mode dua. Pada mode satu, data ini merupakan kode error-correcting 288 byte.

Untuk dapat digunakan diberbagai sistem operasi, perlu adanya sistem file CD-ROM yang standar. Diadakan pertemuan antar produsen CD untuk membahas standar ini di High Sierras (perbatasan California – Nevada) sehingga standar sistem file CD-ROM dikenal dengan sebutan *High Sierra* (IS 9660). Standar ini meliputi 3 level. Level 1 diantaranya berisi :

- Nama – nama file maksimum 8 karakter, yang secara opsional diikuti dengan nama ekstensi maksimal 3 karakter. (Menyesuaikan sistem operasi MS-DOS. Untuk level 2 mencapai 32 karakter.
- Nama – nama file hanya dapat memuat huruf – huruf besar, digit, dan karakter tambahan tertentu saja.

Direktori dapat dibuat hingga mencapai 8 tingkat tanpa memuat karakter ekstensi.



Gambar 9.7 Format blok CD-ROM

CD – R

(Compact Disk Recordables) Secara fisik CD-R merupakan CD polikarbonat kosong berdiameter 120 mm sama seperti CD ROM. Perbedaannya adanya alur – alur untuk mengarahkan laser saat penulisan. Awalnya CD-R dilapisi emas sebagai media refleksinya.

Permukaan reflektif pada lapisan emas tidak memiliki depresi atau lekukan – lekukan fisik seperti halnya pada lapisan aluminium sehingga harus dibuat tiruan lekukan antara pit dan land-nya. Caranya dengan menambahkan lapisan pewarna di antara pilikarbonat dan lapisan emas. Jenis pewarna yang sering digunakan adalah *cyanine* yang berwarna hijau dan *pthalocynine* yang berwarna oranye kekuning-kuningan. Pewarna ini sama seperti yang digunakan dalam film fotografi sehingga menjadikan Kodak dan Fuji produsen utama CD-R.

Sebelum digunakan pewarna bersifat transparan sehingga sinar laser berdaya tinggi dapat menembus sampai ke lapisan emas saat proses penulisan. Saat sinar laser mengenai titik pewarna, sinar ini memanasnya sehingga pewarna terurai melepaskan ikatan kimianya membentuk suatu noda. Noda – noda inilah sebagai representasi data yang nantinya dapat dikenali oleh foto-detektor apabila disinari dengan laser berdaya rendah saat proses pembacaan.

Seperti halnya jenis CD lainnya, CD-R dipublikasikan dalam buku tersendiri yang memuat spesifikasi teknisnya yang dikenal dengan *Orange Book*. Buku ini dipublikasikan tahun 1989.

Terdapat format pengembangan, yaitu ditemukannya seri CD-ROM XA yang memungkinkan penulisan CD-R secara inkremental sehingga menambah fleksibilitas produk ini. Kenapa hal ini bisa dilakukan, karena sistem ini memiliki multitrack dan setiap track memiliki VOTC (*volume table of content*) tersendiri. Berbeda dengan model CD-ROM sebelumnya yang hanya memiliki VOTC tunggal pada permulaan saja.

CD – RW

(*Compact Disk Rewritables*) Jenis CD ini memungkinkan penulisan berulang kali sehingga jenis ini memiliki nilai kompetitif dibandingkan jenis lain. Namun CD-RW belum banyak dipasarkan karena masih relatif mahal.

Karena proses penulisan berulang kali maka secara fisik berbeda dengan CD-R. CD-RW tidak menggunakan lapisan pewarna, namun menggunakan logam paduan antara perak, indium, antimon dan tellurium.

CD-RW drive menggunakan laser dalam 3 daya berbeda. Laser berdaya tinggi bertugas melelehkan paduan logam untuk mengubah kondisi stabil kristalin reflektivitas tinggi menjadi kondisi stabil amorf reflektivitas rendah agar menyerupai sebiah pit. Laser berdaya sedang menjadikan logam paduan meleleh dan berubah menjadi kondisi kristalin alamiah sebagai representasi land. Sedangkan laser berdaya rendah digunakan dalam proses pembacaan saja.

Saat ini CD-RW belum mampu menggeser penggunaan CD-R karena disamping harganya masih relatif mahal dibandingkan CD-R, juga karena CD-R yang tidak dapat dihapus merupakan backup data terbaik saat ini.

DVD

(*Digital Versatile Disc*, awalnya *Digital Video Disk*) Merupakan pengembangan CD untuk memenuhi kebutuhan pasar dalam penyimpanan memori besar.

Desain DVD sama dengan CD biasa, terbuat dari polikarbonat 1,2 mm yang berisi pit dan land, disinari dioda laser dan dibaca oleh foto-detektor. Hal yang baru adalah :

- Pit – pit lebih kecil (0,4 mikron, atau setengahnya CD biasa)
- Spiral lebih rapat (0,74 mikron, sedangkan pada CD biasa 1,6 mikron)
- Menggunakan teknologi laser merah dengan ukuran 0,65 mikron, sedangkan pada CD biasa 0,78 mikron.

Hal baru diatas menjadikan DVD lebih besar kapasitasnya, yaitu untuk sisi tunggal dan berlapis tunggal 4,7 GB, sedangkan untuk berlapis ganda ataupun bersisi ganda akan lebih besar lagi.

Tranfer data pada DVD drive sekitar 1,4 MB/det, sedangkan CD biasa hanya 150 KB/det. Kecepatan, teknologi laser yang berbeda menimbulkan sedikit masalah untuk kompatibilitas dengan teknologi CD maupun CD-ROM. Akan tetapi, saat ini beberapa produsen telah mengantisipasi dengan diada laser ganda ataupun teknologi lain yang memungkinkan saling kompatibel. Saat ini berkembang 4 format DVD, yaitu :

1. Bersisi tunggal dengan lapisan tunggal (kapasitas 4,7 GB)
2. Bersisi tunggal dengan lapisan ganda (kapasitas 8,5 GB)
3. Bersisi ganda dengan lapisan tunggal (kapasitas 9,4 GB)
4. Bersisi ganda dengan lapisan ganda (kapasitas 17 GB)

Piringan berlapis ganda memiliki satu lapisan reflektif pada bagian bawah, yang ditutup dengan lapisan semireflektif. Lapisan bawah memiliki pit dan land yang lebih lebar agar akurat dalam pembacaan sehingga lapisan bawah berkapasitas lebih kecil daripada lapisan atasnya. Pada piringan bersisi ganda dibuat dengan melekatkan dua sisi disk.

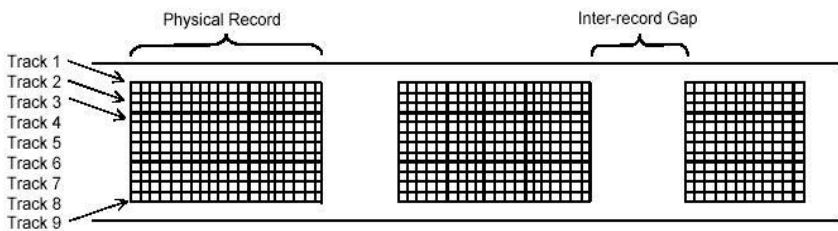
5.4 Pita Magnetik

Sistem pita magnetik menggunakan teknik pembacaan dan penulisan yang identik dengan sistem disk magnetik.

Medium pita magnetik berbentuk track – track paralel, sistem pita lama berjumlah 9 buah track sehingga memungkinkan penyimpanan satu byte sekali simpan dengan satu bit paritas pada track sisanya. Sistem pita baru menggunakan 18 atau 36 track sebagai penyesuaian terhadap lebar word dalam format digital.

Seperti pada disk, pita magnetik dibaca dan ditulisi dalam bentuk blok – blok yang bersambungan (kontinyu) yang disebut

physical record. Blok – blok tersebut dipisahkan oleh gap yang disebut *inter-record gap*. Gambar 9.8 menyajikan format fisik pita magnetik.



Gambar 9.8 Format fisik pita magnetik

Head pita magnetik merupakan perangkat *sequential access*. Head harus menyesuaikan letak record yang akan dibaca ataupun akan ditulis. Apabila head berada di tempat lebih atas dari record yang diinginkan maka pita perlu dimundurkan dahulu, baru dilakukan pembacaan dengan arah maju. Hal ini sangat berbeda pada teknologi disk yang menggunakan teknik *direct access*. Kecepatan putaran pita magnetik adalah rendah sehingga transfer data menjadi lambat, saat ini pita magnetik mulai ditinggalkan digantikan oleh jenis – jenis produk CD.

Rangkuman

1. Disk adalah piringan bundar yang terbuat dari bahan tertentu (logam atau plastik) dengan permukaan dilapisi bahan yang dapat imagnetisasi. Mekanis membaca/tulis menggunakan kepala baca atau tulis yang disebut *head*.
2. Disk diorganisasi dalam bentuk cincin–cincin konsentris yang disebut *track*. Tiap track pada disk dipisahkan oleh *gap*. Fungsi gap untuk mencegah atau mengurangi kesalahan pembacaan maupun

penulisan yang disebabkan melesetnya head atau karena interferensi dan magnet.

3. Ada berbagai macam karakteristik utama dari berbagai jenis disk, yaitu gerakan head, portabilitas disk, sides, platters, dan mekanisme head.
4. CD merupakan disk yang tidak dapat dihapus, mampu menyimpan memori kurang lebih 60 menit informasi audio pada salah satu sisinya. Keberhasilan secara komersial CD yang mampu menyimpan data dalam jumlah yang besar, menjadikannya media penyimpan yang fleksibel digunakan di berbagai peralatan seperti komputer, kamera video, MP3 player, dan lain-lain.

Tes Formatif

1. Jelaskan pengertian istilah-istilah berikut : Disk, Track, Gap, Sektor.
2. Bagaimana mekanisme pembacaan maupun penulisan pada disk ?
3. Jelaskan karakteristik disk berdasarkan gerakannya !
4. Jelaskan pula karakteristik disk berdasarkan mekanisme headnya!
5. Jelaskan keunggulan optical disk dibandingkan dengan magnetic disk !
6. Jelaskan apa yang anda pahami tentang pita magnetic.

SATUAN BAHASAN 10

PERALATAN INPUT/OUTPUT

1. Gambaran Singkat Mengenai Materi Kuliah

Dalam materi kuliah ini akan membahas mengenai bagaimana peran modul input/output sebagai piranti antarmuka dengan peralatan peripheral lainnya. Cakupan materi meliputi system masukan dan keluaran, fungsi dan struktur modul I/O, teknik masukan/keluaran, dan perangkat eksternal.

2. Pedoman Mempelajari Materi

Baca dengan cermat dan teliti materi pada satuan bahasan 10 ini, kemudian pahami mekanisme modul I/O dalam menjalankan fungsinya. Hal ini akan membantu anda dalam memahami peran sentral modul I/O dalam sistem komputer. Pada bagian akhir terdapat tes formatif yang perlu anda kerjakan sebagai sarana untuk mengukur seberapa jauh tingkat pemahaman anda terhadap materi ini.

3. Tujuan Pembelajaran

Setelah mempelajari satuan bahasan 10 ini, mahasiswa diharapkan dapat :

- engetahui alasan mengapa piranti-piranti tidak langsung dihubungkan dengan bus system komputer
- enjelaskan fungsi utama dari modul I/O
- engetahui struktur dan fungsi modul I/O
- enjelaskan teknik-teknik masukan dalam operasi I/O
- engetahui pengklasifikasian perangkat eksternal

10.1. Sistem Masukan & Keluaran Komputer

Sistem komputer memiliki tiga komponen utama, yaitu : CPU, memori (primer dan sekunder), dan peralatan masukan/keluaran (*I/O devices*) seperti printer, monitor, keyboard, mouse, dan modem. Beberapa bab sebelumnya telah membahas CPU dan memori, sekarang akan kita jelaskan tentang peralatan atau modul I/O pada bab ini.

Modul I/O merupakan peralatan antarmuka (*interface*) bagi sistem bus atau switch sentral dan mengontrol satu atau lebih perangkat peripheral. Modul I/O tidak hanya sekedar modul penghubung, tetapi sebuah piranti yang berisi logika dalam melakukan fungsi komunikasi antara peripheral dan bus komputer.

Ada beberapa alasan kenapa piranti – piranti tidak langsung dihubungkan dengan bus sistem komputer, yaitu :

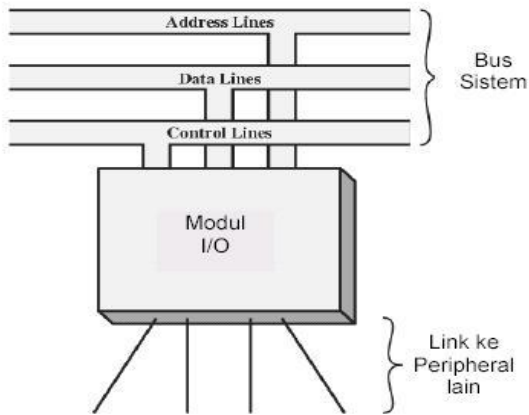
- Bervariasinya metode operasi piranti peripheral, sehingga tidak praktis apabila sistem komputer harus menangani berbagai macam sistem operasi piranti peripheral tersebut.
- Kecepatan transfer data piranti peripheral umumnya lebih lambat dari pada laju transfer data pada CPU maupun memori.
- Format data dan panjang data pada piranti peripheral seringkali berbeda dengan CPU, sehingga perlu modul untuk menselaraskannya.

Dari beberapa alasan diatas, modul I/O memiliki dua buah fungsi utama, yaitu :

1. Sebagai piranti antarmuka ke CPU dan memori melalui bus sistem.
2. Sebagai piranti antarmuka dengan peralatan peripheral lainnya dengan menggunakan link data tertentu.

Bagaimana modul I/O dapat menjalankan tugasnya, yaitu menjembatani CPU dan memori dengan dunia luar merupakan hal yang terpenting untuk kita ketahui. Inti mempelajari sistem I/O suatu komputer adalah mengetahui fungsi dan struktur modul I/O. Perhatikan gambar 10.1 yang menyajikan model generik modul I/O.

Peralatan Input/Output



Gambar 10.1 Model generik dari suatu modul I/O

10.1.1 Fungsi Modul I/O

Modul I/O adalah suatu komponen dalam sistem komputer yang bertanggung jawab atas pengontrolan sebuah perangkat luar atau lebih dan bertanggung jawab pula dalam pertukaran data antara perangkat luar tersebut dengan memori utama ataupun dengan register – register CPU. Dalam mewujudkan hal ini, diperlukan antarmuka internal dengan komputer (CPU dan memori utama) dan antarmuka dengan perangkat eksternalnya untuk menjalankan fungsi – fungsi pengontrolan.

Fungsi dalam menjalankan tugas bagi modul I/O dapat dibagi menjadi beberapa kategori, yaitu:

- Kontrol dan pewaktuan.
- Komunikasi CPU.
- Komunikasi perangkat eksternal.
- Pem-buffer-an data.
- Deteksi kesalahan.

Fungsi kontrol dan pewaktuan (*control & timing*) merupakan hal yang penting untuk mensinkronkan kerja masing – masing komponen penyusun komputer. Dalam sekali waktu CPU berkomunikasi dengan satu atau lebih perangkat dengan pola tidak menentu dan kecepatan transfer komunikasi data yang beragam, baik dengan perangkat internal seperti register – register, memori utama, memori sekunder, perangkat peripheral. Proses tersebut bisa berjalan apabila ada fungsi kontrol dan pewaktuan yang mengatur sistem secara keseluruhan. Contoh kontrol pemindahan data dari peripheral ke CPU melalui sebuah modul I/O dapat meliputi langkah – langkah berikut ini :

1. Permintaan dan pemeriksaan status perangkat dari CPU ke modul I/O.
2. Modul I/O memberi jawaban atas permintaan CPU.
3. Apabila perangkat eksternal telah siap untuk transfer data, maka CPU akan mengirimkan perintah ke modul I/O.
4. Modul I/O akan menerima paket data dengan panjang tertentu dari peripheral.
5. Selanjutnya data dikirim ke CPU setelah diadakan sinkronisasi panjang data dan kecepatan transfer oleh modul I/O sehingga paket – paket data dapat diterima CPU dengan baik.

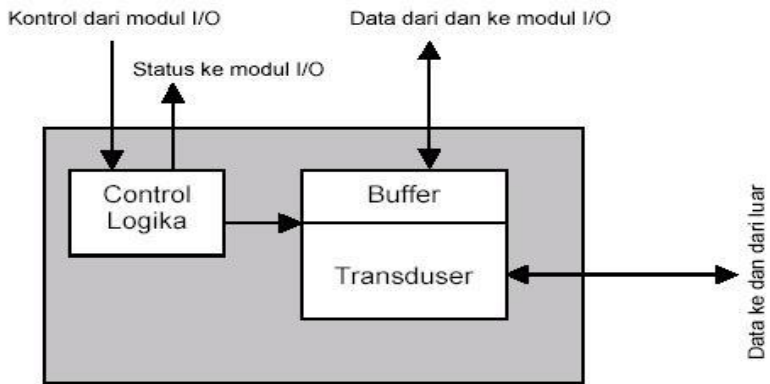
Transfer data tidak akan lepas dari penggunaan sistem bus, maka interaksi CPU dan modul I/O akan melibatkan kontrol dan pewaktuan sebuah arbitrase bus atau lebih.

Adapun fungsi komunikasi antara CPU dan modul I/O meliputi proses – proses berikut :

- *Command Decoding*, yaitu modul I/O menerima perintah – perintah dari CPU yang dikirimkan sebagai sinyal bagi bus kontrol. Misalnya, sebuah modul I/O untuk disk dapat menerima perintah: Read sector, Scan record ID, Format disk.
- *Data*, pertukaran data antara CPU dan modul I/O melalui bus data.
- *Status Reporting*, yaitu pelaporan kondisi status modul I/O maupun perangkat peripheral, umumnya berupa status kondisi *Busy* atau *Ready*. Juga status bermacam – macam kondisi kesalahan (*error*).
- *Address Recognition*, bahwa peralatan atau komponen penyusun komputer dapat dihubungi atau dipanggil maka

harus memiliki alamat yang unik, begitu pula pada perangkat peripheral, sehingga setiap modul I/O harus mengetahui alamat peripheral yang dikontrolnya.

Pada sisi modul I/O ke perangkat peripheral juga terdapat komunikasi yang meliputi komunikasi data, kontrol maupun status. Perhatikan gambar 10.2 berikut.



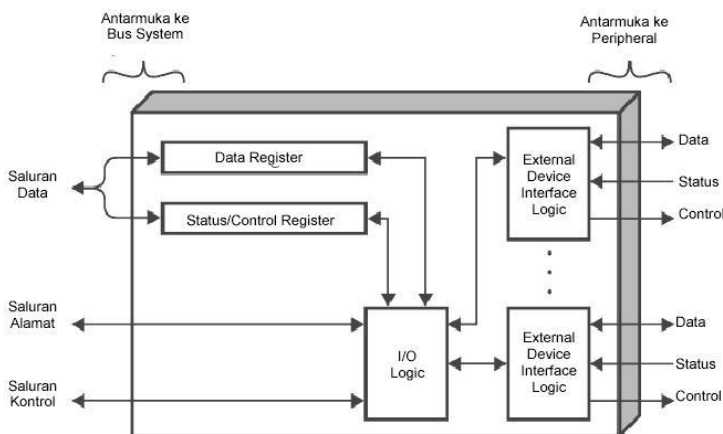
Gambar 10.2 Skema suatu perangkat peripheral

Fungsi selanjutnya adalah *buffering*. Tujuan utama *buffering* adalah mendapatkan penyesuaian data sehubungan perbedaan laju transfer data dari perangkat peripheral dengan kecepatan pengolahan pada CPU. Umumnya laju transfer data dari perangkat peripheral lebih lambat dari kecepatan CPU maupun media penyimpanan.

Fungsi terakhir adalah deteksi kesalahan. Apabila pada perangkat peripheral terdapat masalah sehingga proses tidak dapat dijalankan, maka modul I/O akan melaporkan kesalahan tersebut. Misal informasi kesalahan pada peripheral printer seperti: kertas tergulung, tinta habis, kertas habis, dan lain – lain. Teknik yang umum untuk deteksi kesalahan adalah penggunaan bit paritas.

10.1.2 Struktur Modul I/O

Terdapat berbagai macam modul I/O seiring perkembangan komputer itu sendiri, contoh yang sederhana dan fleksibel adalah Intel 8255A yang sering disebut PPI (*Programmable Peripheral Interface*). Bagaimanapun kompleksitas suatu modul I/O, terdapat kemiripan struktur, seperti terlihat pada gambar 10.3.



Gambar 10.3 Blok diagram struktur modul I/O

Antarmuka modul I/O ke CPU melalui bus sistem komputer terdapat tiga saluran, yaitu saluran data, saluran alamat dan saluran kontrol. Bagian terpenting adalah blok logika I/O yang berhubungan dengan semua peralatan antarmuka peripheral, terdapat fungsi pengaturan dan switching pada blok ini.

10.2 Teknik Masukan/Keluaran

Terdapat tiga buah teknik dalam operasi I/O, yaitu: I/O terprogram, *interrupt – driven*, I/O, dan DMA (*Direct Memory Access*). Ketiganya memiliki keunggulan maupun kelemahan, yang penggunaannya disesuaikan sesuai unjuk kerja masing – masing teknik.

10.2.1 I/O Terprogram

Pada I/O terprogram, data saling dipertukarkan antara CPU dan modul I/O. CPU mengeksekusi program yang memberikan operasi I/O kepada CPU secara langsung, seperti pemindahan data, pengiriman perintah baca maupun tulis, dan monitoring perangkat.

Kelemahan teknik ini adalah CPU akan menunggu sampai operasi I/O selesai dilakukan modul I/O sehingga akan membuang waktu, apalagi CPU lebih cepat proses operasinya. Dalam teknik ini, modul I/O tidak dapat melakukan interupsi kepada CPU terhadap proses – proses yang diinteruksikan padanya. Seluruh proses merupakan tanggung jawab CPU sampai operasi lengkap dilaksanakan.

Untuk melaksanakan perintah – perintah I/O, CPU akan mengeluarkan sebuah alamat bagi modul I/O dan perangkat peripheralnya sehingga terspesifikasi secara khusus dan sebuah perintah I/O yang akan dilakukan. Terdapat empat klasifikasi perintah I/O, yaitu:

1. Perintah *control*.

Perintah ini digunakan untuk mengaktivasi perangkat peripheral dan memberitahukan tugas yang diperintahkan padanya.

2. Perintah *test*.

Perintah ini digunakan CPU untuk menguji berbagai kondisi status modul I/O dan peripheralnya. CPU perlu mengetahui perangkat peripheralnya dalam keadaan aktif dan siap digunakan, juga untuk mengetahui operasi – operasi I/O yang dijalankan serta mendeteksi kesalahannya.

3. Perintah *read*.

Perintah pada modul I/O untuk mengambil suatu paket data kemudian menaruh dalam buffer internal. Proses selanjutnya paket data dikirim melalui bus data setelah terjadi sinkronisasi data maupun kecepatan transfernya.

4. Perintah *write*.

Perintah ini kebalikan dari *read*. CPU memerintahkan modul I/O untuk mengambil data dari bus data untuk diberikan pada perangkat peripheral tujuan data tersebut.

Dalam teknik I/O terprogram, terdapat dua macam implementasi perintah I/O yang tertuang dalam instruksi I/O, yaitu: *memory-mapped I/O* dan *isolated I/O*.

Dalam *memory-mapped I/O*, terdapat ruang tunggal untuk lokasi memori dan perangkat I/O. CPU memperlakukan register status dan register data modul I/O sebagai lokasi memori dan menggunakan instruksi mesin yang sama untuk mengakses baik memori maupun perangkat I/O. Konsekuensinya adalah diperlukan saluran tunggal untuk pembacaan dan saluran tunggal untuk penulisan. Keuntungan *memory-mapped I/O* adalah efisien dalam pemrograman, namun memakan banyak ruang memori alamat.

Dalam teknik *isolated I/O*, dilakukan pemisahan ruang pengalamatan bagi memori dan ruang pengalamatan bagi I/O. Dengan teknik ini

diperlukan bus yang dilengkapi dengan saluran pembacaan dan penulisan memori ditambah saluran perintah output. Keuntungan *isolated I/O* adalah sedikitnya instruksi I/O.

10.2.2 Interrupt – Driven I/O

Teknik interrupt – driven I/O memungkinkan proses tidak membuang – buang waktu. Prosesnya adalah CPU mengeluarkan perintah I/O pada modul I/O, bersamaan perintah I/O dijalankan modul I/O maka CPU akan melakukan eksekusi perintah – perintah lainnya. Apabila modul I/O telah selesai menjalankan instruksi yang diberikan padanya akan melakukan interupsi pada CPU bahwa tugasnya telah selesai.

Dalam teknik ini kendali perintah masih menjadi tanggung jawab CPU, baik pengambilan perintah dari memori maupun pelaksanaan isi perintah tersebut. Terdapat selangkah kemajuan dari teknik sebelumnya, yaitu CPU melakukan *multitasking* beberapa perintah sekaligus sehingga tidak ada waktu tunggu bagi CPU.

Cara kerja teknik interupsi di sisi modul I/O adalah modul I/O menerima perintah, misal *read*. Kemudian modul I/O melaksanakan perintah pembacaan dari peripheral dan meletakkan paket data ke register data modul I/O, selanjutnya modul mengeluarkan sinyal interupsi ke CPU melalui saluran kontrol. Kemudian modul menunggu datanya diminta CPU. Saat permintaan terjadi, modul meletakkan data pada bus data dan modul siap menerima perintah selanjutnya.

Pengolahan interupsi saat perangkat I/O telah menyelesaikan sebuah operasi I/O adalah sebagai berikut :

1. Perangkat I/O akan mengirimkan sinyal interupsi ke CPU.
2. CPU menyelesaikan operasi yang sedang dijalankannya kemudian merespon interupsi.
3. CPU memeriksa interupsi tersebut, kalau valid maka CPU akan mengirimkan sinyal *acknowledgment* ke perangkat I/O untuk menghentikan interupsinya.
4. CPU mempersiapkan pengontrolan transfer ke routine interupsi. Hal yang dilakukan adalah menyimpan informasi yang diperlukan untuk melanjutkan operasi yang tadi dijalankan sebelum adanya interupsi. Informasi yang diperlukan berupa:

- a. Status prosesor, berisi register yang dipanggil PSW (*program status word*).
- b. Lokasi intruksi berikutnya yang akan dieksekusi.

Informasi tersebut kemudian disimpan dalam stack pengontrol sistem.

5. Kemudian CPU akan menyimpan PC (*program counter*) eksekusi sebelum interupsi ke stack pengontrol bersama informasi PSW. Selanjutnya mempersiapkan PC untuk penanganan interupsi.
6. Selanjutnya CPU memproses interupsi sampai selesai.
7. Apabila pengolahan interupsi selesai, CPU akan memanggil kembali informasi yang telah disimpan pada stack pengontrol untuk meneruskan operasi sebelum interupsi.

Terdapat bermacam teknik yang digunakan CPU dalam menangani program interupsi ini, diantaranya :

- *Multiple Interrupt Lines*.
- *Software poll*.
- *Daisy Chain*.
- *Arbitrasi bus*.

Teknik yang paling sederhana adalah menggunakan saluran interupsi berjumlah banyak (*Multiple Interrupt Lines*) antara CPU dan modul – modul I/O. Namun tidak praktis untuk menggunakan sejumlah saluran bus atau pin CPU ke seluruh saluran interupsi modul – modul I/O.

Alternatif lainnya adalah menggunakan *software poll*. Prosesnya, apabila CPU mengetahui adanya sebuah interupsi, maka CPU akan menuju ke routine layanan interupsi yang tugasnya melakukan poll seluruh modul I/O untuk menentukan modul yang melakukan interupsi. Kerugian *software poll* adalah memerlukan waktu yang lama karena harus mengidentifikasi seluruh modul untuk mengetahui modul I/O yang melakukan interupsi.

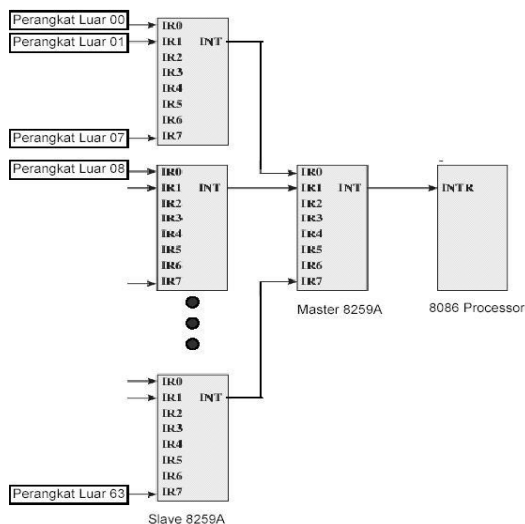
Teknik yang lebih efisien adalah *daisy chain*, yang menggunakan *hardware poll*. Seluruh modul I/O tersambung dalam saluran interupsi CPU secara melingkar (*chain*). Apabila ada permintaan interupsi, maka CPU akan menjalankan sinyal *acknowledge* yang berjalan pada saluran interupsi sampai menjumpai modul I/O yang mengirimkan interupsi.

Teknik berikutnya adalah *arbitrasi bus*. Dalam metode ini, pertama – tama modul I/O memperoleh control bus sebelum modul ini menggunakan saluran permintaan interupsi. Dengan demikian hanya akan terdapat sebuah modul I/O yang dapat melakukan interupsi.

Pengontrol Interrupt Intel 8259A

Intel mengeluarkan chips 8259A yang dikonfigurasi sebagai *interrupt arbiter* pada mikroprosesor Intel 8086. Intel 8259A melakukan manajemen interupsi modul - modul I/O yang tersambung padanya. Chips ini dapat diprogram untuk menentukan prioritas modul I/O yang lebih dulu ditangani CPU apabila ada permintaan interupsi yang bersamaan. Gambar 10.4 menggambarkan pemakaian pengontrol interupsi 8259A. Berikut mode-mode interupsi yang mungkin terjadi :

- *Fully Nested*: permintaan interupsi dengan prioritas mulai 0 (IR0) hingga 7 (IR7).
- *Rotating*: bila sebuah modul telah dilayani interupsinya akan menempati prioritas terendah.
- *Special Mask*: prioritas diprogram untuk modul I/O tertentu secara spesial.

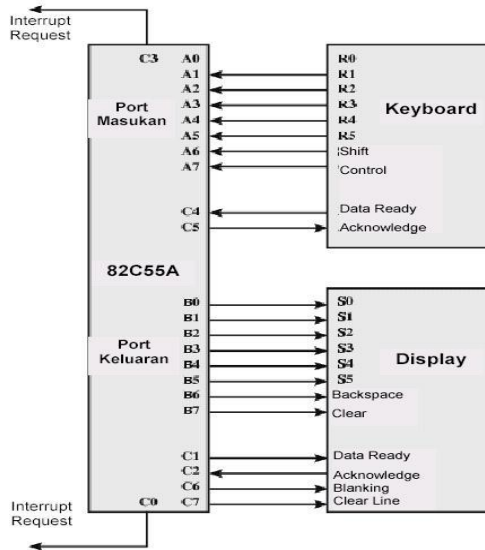


Gambar 10.4 Pemakaian pengontrol interupsi 8259A pada 8086

Programmable Peripheral Interface Intel 8255A

Contoh modul I/O yang menggunakan I/O terprogram dan interrupt driven I/O adalah Intel 8255A Programmable Peripheral Interface (PPI). Intel 8255A dirancang untuk keperluan mikroprosesor 8086. Gambar 10.5 menunjukkan blok diagram Intel 8255A dan pin layout-nya.

Peralatan Input/Output



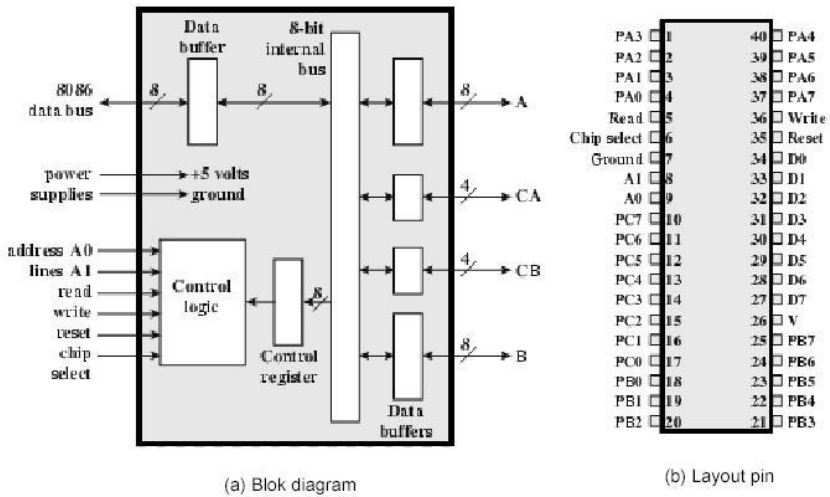
Gambar 10.5 Modul I/O 8255A

Bagian kanan dari blok diagram Intel 8255A adalah 24 saluran antarmuka luar, terdiri atas 8 bit port A, 8 bit port B, 4 bit port CA dan 4 bit port CB. Saluran tersebut dapat diprogram dari mikroprosesor 8086 dengan menggunakan register kontrol untuk menentukan bermacam – macam mode operasi dan konfigurasi. Bagian kiri blok diagram merupakan interface internal dengan mikroprosesor 8086. Saluran ini terdiri atas 8 bus data dua arah (D0 – D7), bus alamat, dan bus kontrol yang terdiri atas saluran CHIP SELECT, READ, WRITE, dan RESET.

Pengaturan mode operasi pada register kontrol dilakukan oleh mikroprosesor. Pada Mode 0, ketiga port berfungsi sebagai tiga port

I/O 8 bit. Pada mode lain dapat port A dan port B sebagai port I/O 8 bit, sedangkan port C sebagai pengontrol saluran port A dan B.

PPI Intel 8255A dapat diprogram untuk mengontrol berbagai peripheral sederhana. Gambar 10.6 memperlihatkan contoh penggunaan 8255A untuk modul I/O Keyboard dan display.



Gambar 10.6 Interface keyboard dan display dengan Intel 8255A

10.2.3 Direct Memory Access (DMA)

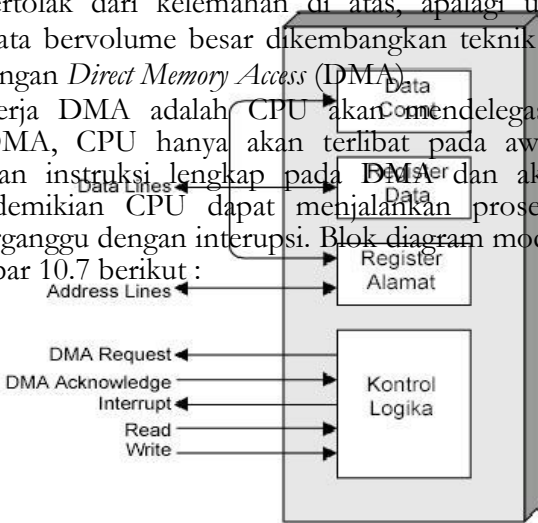
Teknik yang dijelaskan sebelumnya yaitu I/O terprogram dan Interrupt-Driven I/O memiliki kelemahan, yaitu proses yang terjadi pada modul I/O masih melibatkan CPU secara langsung. Hal ini berimplikasi pada :

- Kelajuan transfer I/O yang tergantung pada kecepatan operasi CPU.
- Kerja CPU terganggu karena adanya interupsi secara langsung.

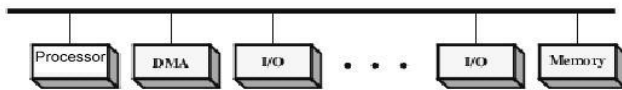
Peralatan Input/Output

Bertolak dari kelemahan di atas, apalagi untuk menangani transfer data bervolume besar dikembangkan teknik yang lebih baik, dikenal dengan *Direct Memory Access* (DMA).

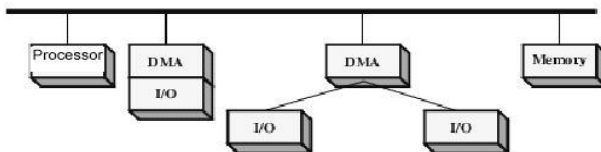
Prinsip kerja DMA adalah CPU akan mendelegasikan kerja I/O kepada DMA, CPU hanya akan terlibat pada awal proses untuk memberikan instruksi lengkap pada DMA dan akhir proses saja. Dengan demikian CPU dapat menjalankan proses lainnya tanpa banyak terganggu dengan interupsi. Blok diagram modul DMA terlihat pada gambar 10.7 berikut :



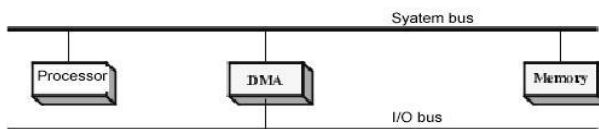
Gambar 10.7 Blok diagram DMA



(a) Single bus



(b) Single bus, Integrated DMA-I/O



Gambar 10.8 Konfigurasi modul DMA

Dalam melaksanakan transfer data secara mandiri, DMA memerlukan pengambilalihan kontrol bus dari CPU. Untuk itu DMA akan menggunakan bus bila CPU tidak menggunakannya atau DMA memaksa CPU untuk menghentikan sementara penggunaan bus. Teknik terakhir lebih umum digunakan, sering disebut *cycle-stealing*, karena modul DMA mengambil alih siklus bus. Penghentian sementara penggunaan bus bukanlah bentuk interupsi, melainkan hanyalah penghentian proses sesaat yang berimplikasi hanya pada kelambatan eksekusi CPU saja. Terdapat tiga buah konfigurasi modul DMA seperti yang terlihat pada gambar 10.8.

10.3 Perangkat Eksternal

Mesin komputer akan memiliki nilai apabila bisa berinteraksi dengan dunia luar. Lebih dari itu, komputer tidak akan berfungsi apabila tidak dapat berinteraksi dengan dunia luar. Ambil contoh saja, bagaimana kita bisa menginstruksikan CPU untuk melakukan suatu operasi apabila tidak ada keyboard. Bagaimana kita melihat hasil kerja sistem komputer bila tidak ada monitor. Keyboard dan monitor tergolong dalam perangkat eksternal komputer.

Perangkat eksternal atau lebih umum disebut *peripheral* tersambung dalam sistem CPU melalui perangkat pengendalinya, yaitu modul I/O seperti telah dijelaskan sebelumnya. Lihat kembali gambar 10.2. Secara umum perangkat eksternal diklasifikasikan menjadi 3 kategori:

- *Human Readable*, yaitu perangkat yang berhubungan dengan manusia sebagai pengguna komputer. Contohnya: monitor, keyboard, mouse, printer, joystick, disk drive.
- *Machine readable*, yaitu perangkat yang berhubungan dengan peralatan. Biasanya berupa modul sensor dan transduser untuk monitoring dan kontrol suatu peralatan atau sistem.

- *Communication*, yaitu perangkat yang berhubungan dengan komunikasi jarak jauh. Misalnya: NIC dan modem.

Pengklasifikasian juga bisa berdasarkan arah datanya, yaitu perangkat output, perangkat input dan kombinasi output-input. Contoh perangkat output: monitor, proyektor dan printer. Perangkat input misalnya: keyboard, mouse, joystick, scanner, mark reader, bar code reader.

Rangkuman

1. Modul I/O adalah suatu komponen dalam sistem komputer yang bertanggung jawab atas pengontrolan sebuah perangkat luar atau lebih dan bertanggung jawab pula dalam pertukaran data antara perangkat luar tersebut dengan memori utama ataupun dengan register – register CPU.
2. Modul I/O memiliki dua buah fungsi utama, yaitu : sebagai piranti antarmuka ke CPU dan memori melalui bus sistem, dan sebagai piranti antarmuka dengan peralatan peripheral lainnya dengan menggunakan link data tertentu.
3. Piranti-piranti I/O tidak langsung dihubungkan dengan bus sistem komputer, karena : Bervariasinya metode operasi piranti peripheral, kecepatan transfer data piranti peripheral umumnya lebih lambat dari pada laju transfer data pada CPU maupun memori, dan format data dan panjang data pada piranti peripheral seringkali berbeda dengan CPU.
4. Terdapat tiga buah teknik dalam operasi I/O, yaitu: I/O terprogram, *interrupt – driven*, I/O, dan DMA (*Direct Memory Access*). Ketiganya memiliki keunggulan maupun kelemahan, yang penggunaannya disesuaikan sesuai unjuk kerja masing – masing teknik.
5. Secara umum perangkat eksternal diklasifikasikan menjadi 3 kategori yaitu : *Human Readable*, *Machine readable*, dan *Communication*

Tes Normatif

1. Mengapa piranti-piranti I/O tidak langsung dihubungkan dengan bus sistem komputer ?
2. Jelaskan fungsi utama dari modul I/O !
3. Jelaskan prinsip kerja I/O terprogram !

4. Secara umum perangkat eksternal diklasifikasikan menjadi 3 kategori. Jelaskan !

SATUAN BAHASAN 11

SISTEM BUS

1. Gambaran Singkat Mengenai Materi Kuliah

Materi kuliah ini akan membahas tentang bagaimana interkoneksi komponen sistem komputer dalam menjalankan fungsinya, interkoneksi bus dan juga pertimbangan-pertimbangan perancangan bus. Selain itu juga akan disajikan contoh-contoh bus yang berkembang saat ini.

2. Pedoman Mempelajari Materi

Baca dengan cermat dan teliti materi pada satuan bahasan 11 ini, kemudian pahami setiap bagian dari sistem bus. Hal ini akan membantu anda dalam memahami fungsi bus sebagai penghubung bagi keseluruhan komponen computer. Pada akhir satuan bahasan terdapat tes formatif yang perlu anda kerjakan sebagai sarana untuk mengukur seberapa jauh tingkat pemahaman anda terhadap materi ini.

3. Tujuan Pembelajaran

Setelah mempelajari satuan bahasan 11 ini, mahasiswa diharapkan dapat :

- menjelaskan pengertian dan fungsi dari sistem bus
- mengetahui jenis-jenis pertukaran data yang diperlukan oleh modul-modul penyusun komputer
- mengetahui jenis-jenis perpindahan data dalam struktur interkoneksi
- menjelaskan pengertian dari bus data, bus alamat dan bus control
- mengetahui prinsip operasi bus
- menjelaskan hierarki multiple bus
- menjelaskan elemen-elemen perancangan bus

menjelaskan jenis-jenis bus yang banyak digunakan saat ini

11.1 Struktur Interkoneksi

Komputer tersusun atas beberapa komponen penting seperti CPU, memori, perangkat I/O. Setiap komponen saling berhubungan membentuk kesatuan fungsi. Sistem bus adalah penghubung bagi keseluruhan komponen komputer dalam menjalankan tugasnya. Transfer data antar komponen komputer sangatlah mendominasi kerja suatu komputer. Data atau program yang tersimpan dalam memori dapat diakses dan dieksekusi CPU melalui perantara bus, begitu juga kita dapat melihat hasil eksekusi melalui monitor juga menggunakan sistem bus.

Era saat ini memerlukan saluran data atau bus yang handal. Kecepatan komponen penyusun komputer tidak akan berarti kalau tidak diimbangi kecepatan dan manajemen bus yang baik. Trend mikroprosesor saat ini adalah melakukan pekerjaan secara paralel dan program dijalankan secara multitasking menuntut sistem bus tidak hanya lebar tapi juga cepat.

Komputer tersusun atas komponen – komponen atau modul – modul (CPU, memori dan I/O) yang saling berkomunikasi. Kumpulan lintasan atau saluran berbagai modul disebut *struktur interkoneksi*. Rancangan struktur interkoneksi sangat bergantung pada jenis dan karakteristik pertukaran datanya.

Gambar 11.1 menyajikan jenis pertukaran data yang diperlukan oleh modul – modul penyusun komputer :

- Memori :

Memori umumnya terdiri atas N word memori dengan panjang yang sama. Masing – masing word diberi alamat numerik yang unik (0, 1, 2, ... $N-1$). Word dapat dibaca maupun ditulis pada memori dengan kontrol Read dan Write. Lokasi bagi operasi dispesifikasikan oleh sebuah alamat.

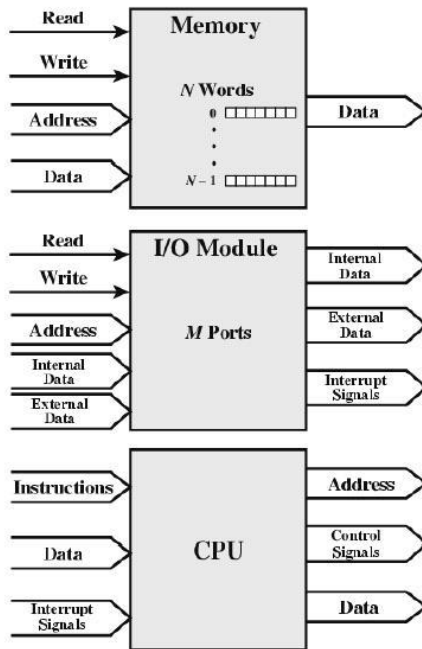
- Modul I/O :

Operasi modul I/O adalah pertukaran data dari dan ke dalam komputer. Berdasarkan pandangan internal, modul I/O dipandang

sebagai sebuah memori dengan operasi pembacaan dan penulisan. Seperti telah dijelaskan pada bab sebelumnya bahwa modul I/O dapat mengontrol lebih dari sebuah perangkat peripheral. Modul I/O juga dapat mengirimkan sinyal interrupt.

• CPU :

CPU berfungsi sebagai pusat pengolahan dan eksekusi data berdasarkan routine – routine program yang diberikan padanya. CPU mengendalikan seluruh sistem komputer sehingga sebagai konsekuensinya memiliki koneksi ke seluruh modul yang menjadi bagian sistem komputer.



Gambar 11.1 Modul – modul komputer

Dari jenis pertukaran data yang diperlukan modul – modul komputer, maka struktur interkoneksi harus mendukung perpindahan data berikut :

- *Memori ke CPU* : CPU melakukan pembacaan data maupun instruksi dari memori.

- *CPU ke Memori* : CPU melakukan penyimpanan atau penulisan data ke memori.
- *I/O ke CPU* : CPU membaca data dari peripheral melalui modul I/O.
- *CPU ke I/O* : CPU mengirimkan data ke perangkat peripheral melalui modul I/O.
- *I/O ke Memori atau dari Memori* : digunakan pada sistem DMA.

Sampai saat ini terjadi perkembangan struktur interkoneksi, namun yang banyak digunakan saat ini adalah sistem bus. Sistem bus ada yang digunakan secara tunggal dan ada secara jamak, tergantung karakteristik sistemnya.

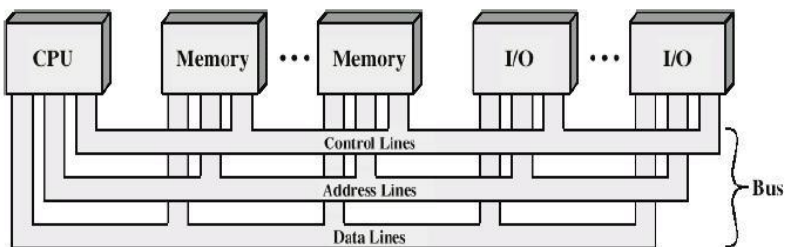
11.2 Interkoneksi Bus

Bus merupakan lintasan komunikasi yang menghubungkan dua atau lebih komponen komputer. Sifat penting dan merupakan syarat utama adalah bus adalah media transmisi yang dapat digunakan bersama oleh sejumlah perangkat yang terhubung padanya.

Karena digunakan bersama, diperlukan aturan main agar tidak terjadi tabrakan data atau kerusakan data yang ditransmisikan. Walaupun digunakan bersama namun dalam satu waktu hanya ada sebuah perangkat yang dapat menggunakan bus.

Struktur Bus

Sebuah bus biasanya terdiri atas beberapa saluran. Sebagai contoh bus data terdiri atas 8 saluran sehingga dalam satu waktu dapat mentransfer data 8 bit. Secara umum fungsi saluran bus dikategorikan dalam tiga bagian, yaitu saluran data, saluran alamat dan saluran kontrol, seperti terlihat pada gambar 11.2.



Gambar 11.2 Pola interkoneksi bus

Saluran data (data bus) adalah lintasan bagi perpindahan data antar modul. Secara kolektif lintasan ini disebut *bus data*. Umumnya jumlah saluran terkait dengan panjang word, misalnya 8, 16, 32 saluran dengan tujuan agar mentransfer word dalam sekali waktu. Jumlah saluran dalam bus data dikatakan *lebar bus*, dengan satuan bit, misal lebar bus 16 bit.

Saluran alamat (address bus) digunakan untuk menspesifikasi sumber dan tujuan data pada bus data. Saluran ini digunakan untuk mengirim alamat word pada memori yang akan diakses CPU. Juga digunakan untuk saluran alamat perangkat modul komputer saat CPU mengakses suatu modul. Perlu diketahui, semua peralatan yang terhubung dengan sistem komputer, agar dapat diakses harus memiliki alamat. Semisal mengakses port I/O, maka port I/O harus memiliki alamat hardware-nya.

Saluran kontrol (control bus) digunakan untuk mengontrol bus data, bus alamat dan seluruh modul yang ada. Karena bus data dan bus alamat digunakan oleh semua komponen maka diperlukan suatu mekanisme kerja yang dikontrol melalui bus kontrol ini. Sinyal – sinyal kontrol terdiri atas sinyal pewaktuan dan sinyal – sinyal perintah. Sinyal pewaktuan menandakan validitas data dan alamat, sedangkan sinyal perintah berfungsi membentuk suatu operasi. Secara umum saluran kontrol meliputi :

- *Memory Write*, memerintahkan data pada bus akan dituliskan ke dalam lokasi alamat.
 - *Memory Read* memerintahkan data dari lokasi alamat ditempatkan pada bus data.
 - *I/O Write*, memerintahkan data pada bus dikirim ke lokasi port I/O.
 - *I/O Read*, memerintahkan data dari port I/O ditempatkan pada bus data.
 - *Transfer ACK*, menunjukkan data telah diterima dari bus atau data telah ditempatkan pada bus.
 - *Bus Request*, menunjukkan bahwa modul memerlukan kontrol bus.
 - *Bus Grant*, menunjukkan modul yang melakukan request telah diberi hak mengontrol bus.
 - *Interrupt Request*, menandakan adanya penangguhan interupsi dari modul.
 - *Interrupt ACK*, menunjukkan penangguhan interupsi telah diketahui CPU.
 - *Clock*, kontrol untuk sinkronisasi operasi antar modul.
- Reset*, digunakan untuk menginisialisasi seluruh modul.

Secara fisik bus adalah konduktor listrik paralel yang menghubungkan modul – modul. Konduktor ini biasanya adalah saluran utama pada PCB motherboard dengan layout tertentu sehingga didapat fleksibilitas penggunaan. Untuk modul I/O biasanya dibuat slot bus yang mudah dipasang dan dilepas, seperti slot PCI dan ISA. Sedangkan untuk chips akan terhubung melalui pinnya.

Prinsip operasi bus adalah sebagai berikut. Operasi pengiriman data ke modul lainnya :

1. Meminta penggunaan bus.
2. Apabila telah disetujui, modul akan memindahkan data yang diinginkan ke modul yang dituju.

Operasi meminta data dari modul lainnya :

1. Meminta penggunaan bus.
2. Mengirim *request* ke modul yang dituju melalui saluran kontrol dan alamat yang sesuai.
3. Menunggu modul yang dituju mengirimkan data yang diinginkan.

Hierarki Multiple Bus

Bila terlalu banyak modul atau perangkat dihubungkan pada bus maka akan terjadi penurunan kinerja, yang disebabkan oleh :

- Semakin besar delay propagasi untuk mengkoordinasikan penggunaan bus.
- Antrian penggunaan bus semakin panjang.
- Dimungkinkan habisnya kapasitas transfer bus sehingga memperlambat data.

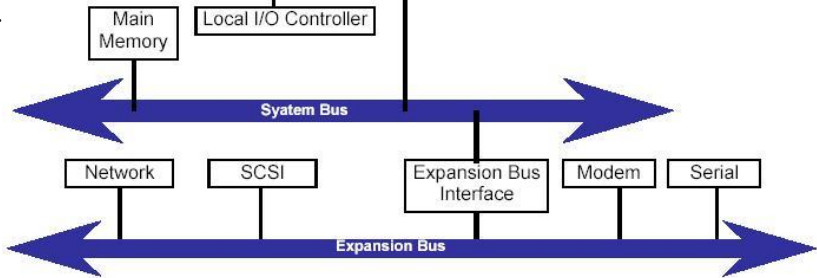
Antisipasi dan solusi persoalan di atas adalah penggunaan bus jamak yang hierarkis.

Modul – modul diklasifikasikan berdasarkan kebutuhan terhadap lebar dan kecepatan bus. Bus biasanya terdiri atas bus lokal, bus sistem, dan bus ekspansi. Gambar 11.3 dan gambar 11.4 berikut menyajikan contoh hierarki penggunaan bus jamak.

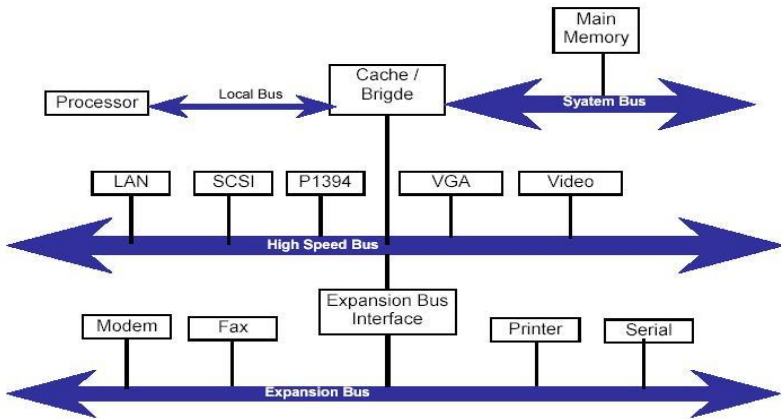
Terlihat pada gambar 11.3 dan gambar 11.4, Prosesor, cache memori dan memori utama terletak pada bus tersendiri pada level tertinggi karena modul – modul tersebut memiliki karakteristik pertukaran data yang tinggi. Pada arsitektur berkinerja tinggi, modul – modul I/O diklasifikasikan menjadi dua, yaitu yang memerlukan transfer data berkecepatan tinggi dan berkecepatan rendah. Modul dengan transfer data berkecepatan tinggi disambungkan dengan bus berkecepatan tinggi pula, sedangkan modul yang tidak memerlukan transfer data cepat disambungkan pada bus ekspansi.

Sistem Bus

Keuntungan hierarki bus jamak kinerja tinggi pada gambar 11.4 adalah bus berkecepatan tinggi lebih terintegrasi dengan prosesor. Perubahan pada arsitektur prosesor tidak begitu mempengaruhi kinerja bus.



Gambar 11.3 Arsitektur bus jamak tradisional



Gambar 11.4 Arsitektur bus jamak kinerja tinggi

11.3 Elemen Perancangan Bus

Saat ini terdapat banyak implementasi sistem bus, tetapi parameter dasar perancangan bus dapat diklasifikasikan berdasarkan jenis (*dedicated* dan *multiplexed*), metode arbitrase (tersentralisasi dan terdistribusi), timing (sinkron dan tak sinkron), lebar bus (lebar address dan lebar data) dan jenis transfer datanya (*read*, *write*, *read-modify-write*, *read-alter-write*, *block*).

Tujuan yang hendak dicapai dalam perancangan adalah bagaimana bus dapat cepat menghantarkan data dan efisiensinya tinggi. Intinya karakteristik pertukaran data dan modul yang terkait merupakan pertimbangan utama dalam perancangan bus.

Jenis Bus

Berdasar jenis busnya, bus dibedakan menjadi bus yang khusus menyalurkan data tertentu, misalnya paket data saja, atau alamat saja, jenis ini disebut *dedicated bus*. Namun apabila bus dilalukan informasi yang berbeda baik data, alamat maupun sinyal kontrol dengan metode multiplex data maka bus ini disebut *multiplexed bus*.

Keuntungan *multiplexed bus* adalah hanya memerlukan saluran sedikit sehingga dapat menghemat tempat, namun kerugiannya adalah kecepatan transfer data menurun dan diperlukan mekanisme yang kompleks untuk mengurai data yang telah dimultiplex.

Saat ini yang umum, bus didedikasikan untuk tiga macam, yaitu bus data, bus alamat dan bus kontrol.

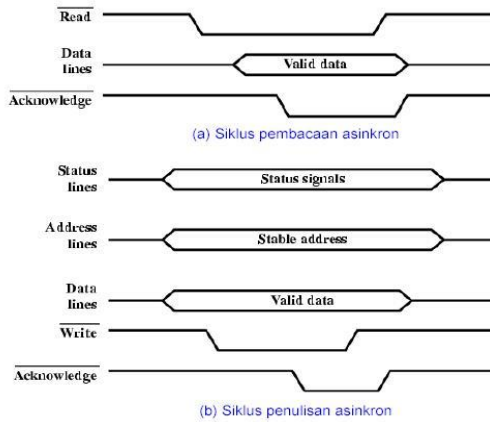
Metode Arbitrase

Terdapat dua macam metode arbitrase, yaitu *tersentral* dan *terdistribusi*. Pada metode tersentral diperlukan pengontrol bus sentral atau *arbiter* yang bertugas mengatur penggunaan bus oleh modul. Arbiter bisa suatu modul atau bagian fungsi CPU. Sedangkan dalam metode terdistribusi, setiap modul memiliki logika pengontrol akses (*access control logic*) yang berfungsi mengatur pertukaran data melalui bus. Kedua metode arbitrase intinya menugaskan suatu perangkat bisa modul I/O ataupun CPU bertindak sebagai master kontrol pertukaran.

Timing

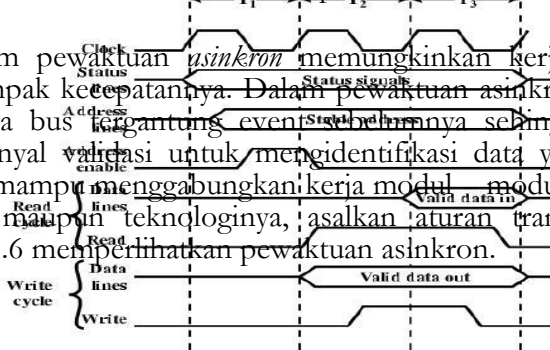
Metode pewaktuan *sinkron* terjadinya event pada bus ditentukan oleh sebuah pewaktu (*clock*). Sebuah transmisi 1 – 0 disebut siklus waktu atau siklus bus dan menentukan besarnya slot waktu. Semua perangkat modul pada bus dapat membaca atau mengetahui siklus clock. Biasanya satu siklus untuk satu event. Model ini mudah

diimplementasikan dan cepat namun kurang fleksibel menangani peralatan yang beda kecepatan operasinya. Biasanya digunakan untuk modul – modul tertentu yang sudah jelas karakteristiknya. Contoh pewaktuan sinkron disajikan pada gambar 11.5.



Gambar 11.5 Contoh pewaktuan sinkron

Dalam pewaktuan *asinkron* memungkinkan kerja modul yang tidak serempak kecepatannya. Dalam pewaktuan asinkron, event yang terjadi pada bus bergantung event sebelumnya sehingga diperlukan sinyal – sinyal validasi untuk mengidentifikasi data yang ditransfer. Sistem ini mampu menggabungkan kerja modul – modul yang berbeda kecepatan maupun teknologinya, asalkan aturan transfernya sama. Gambar 11.6 memperlihatkan pewaktuan asinkron.



Gambar 11.6 Contoh pewaktuan asinkron

Lebar Bus

Lebar bus sangat mempengaruhi kinerja sistem komputer. Semakin lebar bus maka semakin besar data yang dapat ditransfer sekali waktu. Semakin besar bus alamat, akan semakin banyak range lokasi yang dapat direfensikan.

Jenis Transfer Data

Dalam sistem komputer, operasi transfer data adalah pertukaran data antar modul sebagai tindak lanjut atau pendukung operasi yang sedang dilakukan. Saat operasi baca (*read*), terjadi pengambilan data dari memori ke CPU, begitu juga sebaliknya pada operasi penulisan maupun operasi – operasi kombinasi. Bus harus mampu menyediakan layanan saluran bagi semua operasi komputer.

11.4 Contoh Bus

Banyak perusahaan yang mengembangkan bus – bus antarmuka terutama untuk perangkat peripheral. Diantara jenis bus yang beredar di pasaran saat ini adalah PCI, ISA, USB, SCSI, FuturaBus+, FireWire, dan lain – lain. Semua memiliki keunggulan, kelemahan, harga dan teknologi yang berbeda sehingga akan mempengaruhi jenis – jenis penggunaannya.

11.4.1 Bus ISA

Ketika IBM memperkenalkan PC/AT yang berbasiskan CPU 80286, perusahaan ini menghadapi masalah besar. Jika IBM telah memulai sejak awal dan merancang sebuah bus 16 bit yang seluruhnya baru, banyak konsumen potensial akan bergegas membeli mesin tersebut karena tidak ada satupun dari begitu banyak papan plug-in PC yang disediakan oleh para vendor pihak ketiga dapat bekerja dengan

menggunakan mesin baru tersebut. Di sisi lain, dengan tetap berpegang pada bus PC dan 20 jalur alamatnya serta 8 jalur data tidak akan memperoleh manfaat dari keunggulan CPU 80286 untuk mengalamatkan 16 M memori dan mentransfer word 16 bit.

Solusi yang dipilih adalah mengembangkan PC. Kartu-kartu plug-in PC memiliki sebuah konektor sisi dengan 62 kontak, tetapi operasi konektor sisi ini tidak menjangkau seluruh papan ini. Solusi PC/AT adalah menempatkan sebuah konektor sisi kedua pada bagian dasar papan tersebut, dekat dengan konektor sisi utama, dan merancang sirkuit AT untuk beroperasi dengan kedua jenis papan ini.

Konektor kedua pada bus PC/AT memiliki 36 jalur. Dari ke-36 jalur ini, 31 disediakan untuk jalur-jalur alamat tambahan, jalur-jalur data tambahan, jalur-jalur interupsi tambahan, serta untuk daya dan ground. Sisanya digunakan untuk mengatasi perbedaan-perbedaan antara transfer 8 bit dan 16 bit.

Industri komputer personal lainnya merespon perkembangan ini dengan mengadopsi standarnya sendiri, bus ISA (*Industry Standar Architecture*), yang pada dasarnya adalah bus PC/AT yang beroperasi pada 8,33 MHz. Keuntungannya adalah bahwa pendekatan ini tetap mempertahankan kompatibilitas dengan mesin-mesin dan kartu-kartu yang ada. Pendekatan ini juga didasarkan pada sebuah bus yang telah dilisensikan secara bebas oleh IBM kepada banyak perusahaan dalam rangka untuk menjamin bahwa sebanyak mungkin pihak ketiga dapat memproduksi kartu-kartu untuk PC pertama, sesuatu yang kembali menghantui IBM. Setiap PC yang berbasis Intel masih menggunakan bus jenis ini, meskipun biasanya juga disertai dengan satu atau lebih bus lain.

11.4.2 Bus PCI

Peripheral Component Interconnect (PCI) adalah bus yang tidak tergantung prosesor dan berfungsi sebagai bus mezzanine atau bus peripheral. PCI memiliki kinerja tinggi untuk sistem I/O berkecepatan tinggi seperti : video adaptor, NIC, disk controller, sound card, dan lain-lain.

Standard PCI adalah 64 saluran data pada kecepatan 33 MHz, laju transfer data 264 MB per detik atau 2,112 Gbps. Keunggulan PCI tidak hanya pada kecepatannya saja tetapi murah dengan keping yang sedikit.

Intel mulai menerapkan PCI pada tahun 1990 untuk sistem pentiumnya. Untuk mempercepat penggunaan PCI, Intel mempatenkan PCI bagi domain publik sehingga vendor dapat mengeluarkan produk dengan PCI tanpa royalti.

11.4.3 Bus USB

Semua perangkat peripheral tidak efektif apabila dipasang pada bus berkecepatan tinggi PCI, sedangkan banyak peralatan yang memiliki kecepatan rendah seperti keyboard, mouse, dan printer. Sebagai solusinya tujuh vendor komputer (Compaq, DEC, IBM, Intel, Microsoft, NEC, dan Northern Telecom) bersama-sama merancang bus untuk peralatan I/O berkecepatan rendah.

Standard yang dihasilkan dinamakan Universal Standard Bus (USB). Keuntungan yang didapatkan dan tujuan dari penerapan USB adalah sebagai berikut :

1. Pemakai tidak harus memasang tombol atau jumper pada PCB atau peralatan.
2. Pemakai tidak harus membuka casing untuk memasang peralatan I/O baru.
3. Hanya satu jenis kabel yang diperlukan sebagai penghubung.
4. Dapat mensuplai daya pada peralatan-peralatan I/O.
5. Memudahkan pemasangan peralatan-peralatan yang hanya sementara dipasang pada komputer.
6. Tidak diperlukan reboot pada pemasangan peralatan baru dengan USB.
7. Harganya murah

Bandwidth total USB adalah 1,5 MB per detik. Bandwidth itu sudah mencukupi peralatan I/O berkecepatan rendah seperti keyboard, mouse, scanner, telepon digital, printer, dan sebagainya. Kabel pada bus terdiri dari 4 kawat, 2 untuk data, 1 untuk power (+5 volt), dan 1 untuk ground. Sistem pensinyalan mentransmisikan sebuah bilangan nol sebagai transisi tegangan dan sebuah bilangan satu bila tidak ada transmisi tegangan.

11.4.4 Bus SCSI

Small Computer System Interface (SCSI) adalah perangkat peripheral eksternal yang dipopulerkan oleh macintosh pada tahun 1984. SCSI merupakan interface standard untuk drive CD-ROM, peralatan audio, hard disk, dan perangkat penyimpanan eksternal berukuran besar. SCSI menggunakan interface paralel dengan 8, 16, atau 32 saluran data.

Konfigurasi SCSI umumnya berkaitan dengan bus, walaupun pada kenyataannya perangkat-perangkat tersebut dihubungkan secara

daisy-chain. Perangkat SCSI memiliki dua buah konektor, yaitu konektor input dan konektor output. Seluruh perangkat berfungsi secara independen dan dapat saling bertukar data misalnya hard disk dapat mem-back up diri ke tape drive tanpa melibatkan prosesor.

Terdapat beberapa macam versi SCSI. SCSI-1 dibuat tahun 1980 memiliki 8 saluran data, dan beroperasi pada kecepatan 5 MHz. Versi ini memungkinkan sampai 7 perangkat dihubungkan secara daisy-chain. SCSI-2 diperkenalkan tahun 1992 dengan spesifikasi 16 atau 32 saluran data pada kecepatan 10 MHz. SCSI-3 yang mendukung kecepatan yang lebih tinggi sampai saat ini masih dalam tahap penelitian.

11.4.5 Bus P1394 / Fire Wire

Semakin pesatnya kebutuhan bus I/O berkecepatan tinggi dan semakin cepatnya prosesor saat ini yang mencapai 1 GHz, maka perlu diimbangi dengan bus berkecepatan tinggi juga. Bus SCSI dan PCI tidak dapat mencukupi kebutuhan saat ini. Sehingga dikembangkan bus performance tinggi yang dikenal dengan *Fire Wire* (P1394 standard IEEE).

P1394 memiliki kelebihan dibandingkan dengan interface I/O lainnya, yaitu sangat cepat, murah, dan mudah untuk diimplementasikan. Pada kenyataannya P1394 tidak hanya populer pada sistem komputer, namun juga pada peralatan elektronik seperti pada kamera digital, VCR, dan televisi. Kelebihan lain adalah penggunaan transmisi serial sehingga tidak memerlukan banyak kabel.

Rangkuman

1. Sistem bus adalah penghubung bagi keseluruhan komponen komputer dalam menjalankan tugasnya.
2. Kumpulan lintasan atau saluran berbagai modul disebut *struktur interkoneksi*. Rancangan struktur interkoneksi sangat bergantung pada jenis dan karakteristik pertukaran datanya.
3. Struktur interkoneksi harus mendukung perpindahan data dari Memori ke CPU, CPU ke Memori, I/O ke CPU, CPU ke I/O dan I/O ke Memori atau dari Memori.
4. *Saluran data (data bus)* adalah lintasan bagi perpindahan data antar modul. *Saluran alamat (address bus)* digunakan untuk menspesifikasi sumber dan tujuan data pada bus data. *Saluran kontrol (control bus)* digunakan untuk mengontrol bus data, bus alamat dan seluruh modul yang ada.

5. Elemen perancangan bus berdasarkan jenis (*dedicated* dan *multiplexed*), metode arbitrase (tersentralisasi dan terdistribusi), timing (sinkron dan tak sinkron), lebar bus (lebar address dan lebar data) dan jenis transfer datanya (*read, write, read-modify-write, read-alter-write, block*).

Tes Normatif

1. Jelaskan pengertian Bus !
2. Sebutkan jenis-jenis pertukaran data yang diperlukan oleh modul penyusun computer !
3. Jelaskan jenis-jenis perpindahan data dalam struktur interkoneksi !
4. Jelaskan fungsi utama dari Bus Data, Bus Alamat dan Bus Kontrol !
5. Jelaskan elemen perancangan bus berdasarkan metode arbitrasinya !

DAFTAR PUSTAKA

- Abdurohman, Maman, *Organisasi & Arsitektur Komputer*, Penerbit Informatika, Bandung, 2014.
- Ariyanto, Endro, *Computer Organization and Architecture*, <http://EndroAriyanto.stttelkom.ac.id>, diakses tanggal 10 Juni 2015.
- Departemen Agama RI, Al-Quran al-Karim dan Terjemahannya, Semarang: CV. Toha Putra, 2005.
- Faisal, *Handout Organisasi Sistem Komputer*, Teknik Informatika Fakultas Sains dan Teknologi UIN Alauddin Makassar, 2011.
- Hartono, Jogiyanto, *Pengenalan Komputer*, Penerbit Andi, 1988.
- Kadir, Abdul. Triwahyuni, Terra CH., *Pengenalan Teknologi Informasi*, Penerbit Andi, 2003.
- Malvino, Albert Paul. On, Tjia May, *Elektronika Komputer Digital: Pengantar Mikrokomputer, Edisi Kedua*, Penerbit Erlangga, 1991.
- Minartiningtyas, Brigida Arie, Artikel Teknik Infotmatika dan Sistem Informasi, 14 Desember 2012, diakses pada <http://informatika.web.id/category/data-warehouse> tanggal 1 Agustus 2014.
- Muchlas, *Rangkaian Digital*, Penerbit Gava Media, 2005.
- Nugraha, Deni Wiria, Dunia Elektro, Komputer dan Informatika, Mei 2009 diakses pada <http://denywirianugraha.blogspot.com/2009/05/algorithm-prim.html> tanggal 1 Agustus 2014.
- Sicard, E. and Xi, C. *Dsch2 Commands*, <http://intrade.insatles.fr/~etienne>
- Stalling, Williams, *Computer Organization and Architecture, 7th edition*, McGraw Hill, 2010.

Sistem Bus

Stalling, Williams, *Computer Organization and Architecture*, 7th edition,
<http://WilliamStallings.com/COA/COA7e.html>, diakses
tanggal 15 Juli 2015

Willa, Lukas, *Teknik Digital, Mikroprosesor dan Mikrokomputer*, Penerbit
Informatika, 2010.