

Arsitektur dan Organisasi Komputer

COM 60011

Topik #2 – Performance Issues and Top Level View of Computer Function



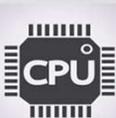
Tujuan

CPMK : Mahasiswa mampu menjabarkan arsitektur dan organisasi dari prosesor (CPU) pada suatu komputer

Sub CPMK : Mahasiswa mampu menjelaskan hubungan antara arsitektur dan organisasi pada suatu komputer dengan performansinya.

Materi Terkait:

- **Computer Components**
- **Basic Instruction Cycle**
- **I/O Function**
- **Bus Interconnection Scheme**
- **Perkembangan sistem komputer**





A Top-Level View of Computer Function and Interconnection





Pendahuluan

- Komputer dibangun oleh 2 macam perangkat yaitu perangkat lunak (software) dan perangkat keras (hardware).
- Hardware komputer memiliki beberapa komponen yang memiliki fungsi dan spesifikasi tersendiri.
- Komponen utamanya antara lain: I/O module, CPU, dan Memory.
- Komponen-komponen tersebut juga saling berinteraksi yang dihubungkan oleh sistem bus, sehingga antar komponen dapat berinteraksi satu sama lain sesuai dengan instruksi yang ada.





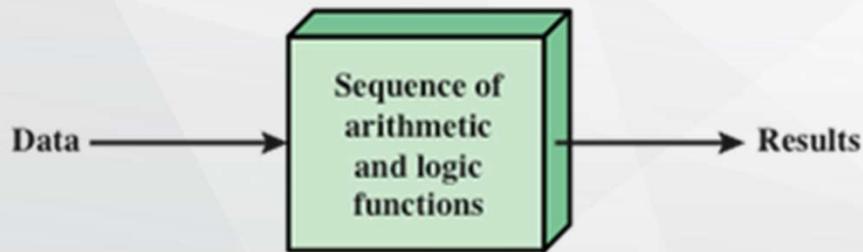
Komponen Komputer

- Menurut Von Neuman Komponen utama pada hardware komputer ada 3:
 - Central Processing Unit (CPU),
 - Memory,
 - I/O module.
- Fungsi komponen tersebut pada CPU berfungsi sebagai pusat proses
- Dalam CPU itu terdapat
 - Control unit yang berfungsi untuk memberikan arahan/kendali/ kontrol
 - ALU berfungsi untuk melakukan operasi hitungan aritmatika dan logika.
 - I/O yang berfungsi untuk memasukkan data dan instruksi ke dalam sistem dan untuk menunjukkan hasil dari pengolahan data
 - Memory merupakan tempat penyimpanan data sementara kode ataupun hasil yang pada saat itu di butuhkan.
- Hardwired Program : Pada CPU, seluruh program dapat dilakukan konfigurasi sesuai keinginan

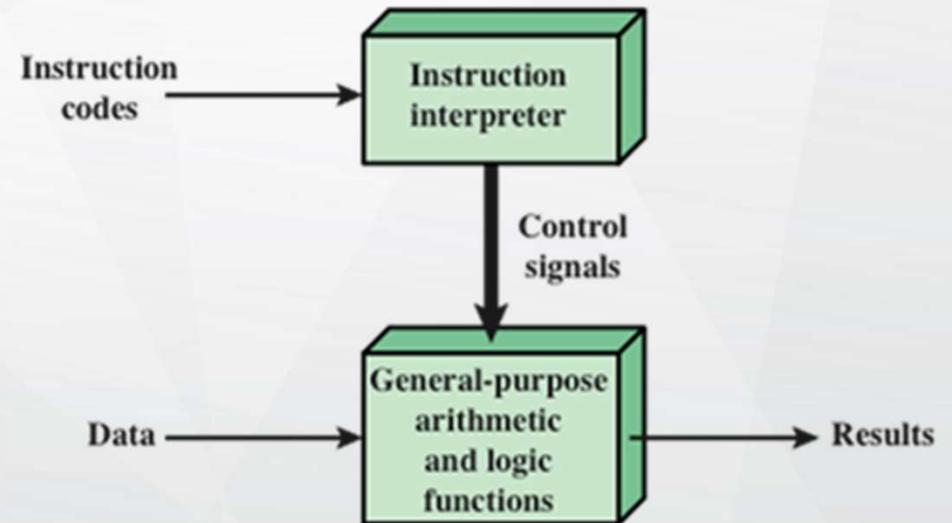




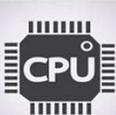
Pendekatan Programming pada Hardware dan Software



(a) Programming in hardware

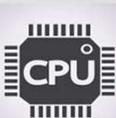
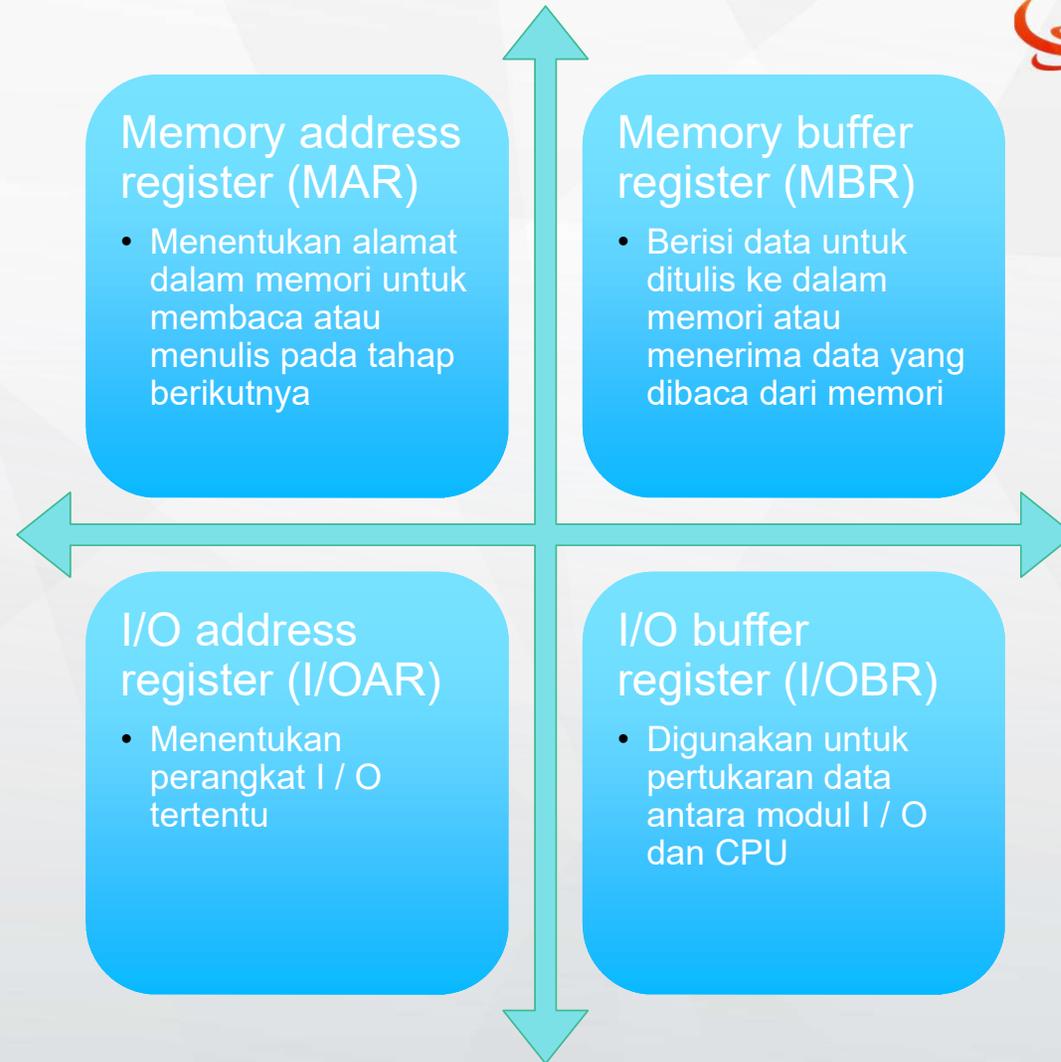


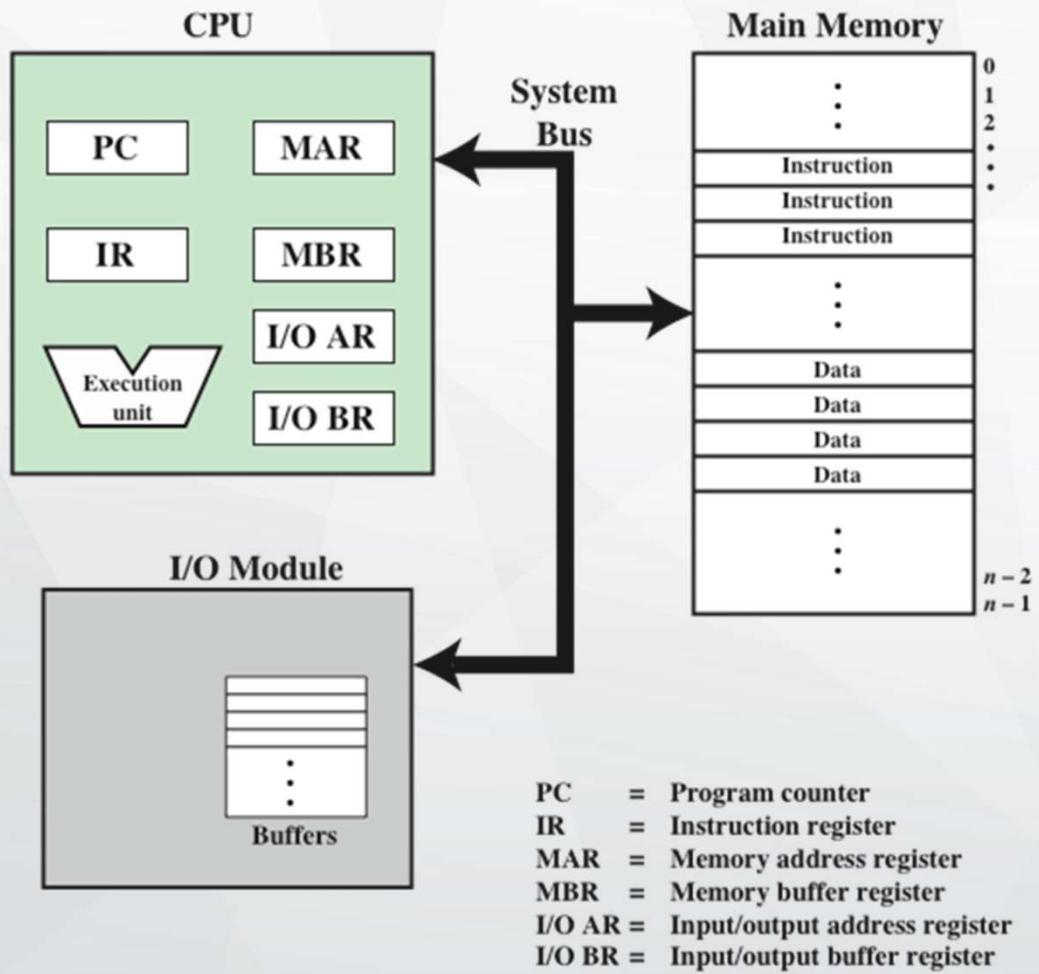
(b) Programming in software





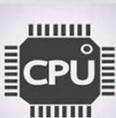
Memori pada CPU





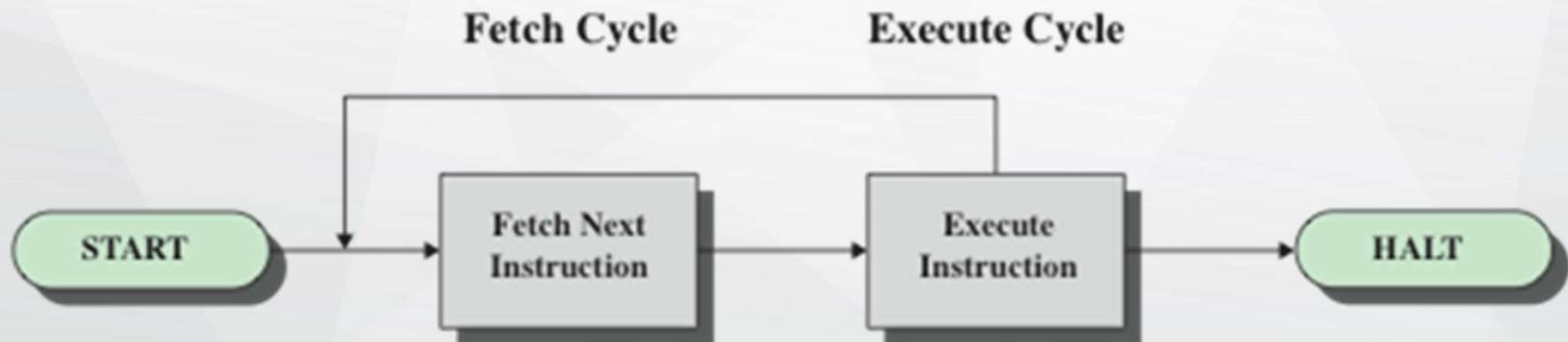
Computer Components: Top Level View

Figure 3.2 Computer Components: Top-Level View

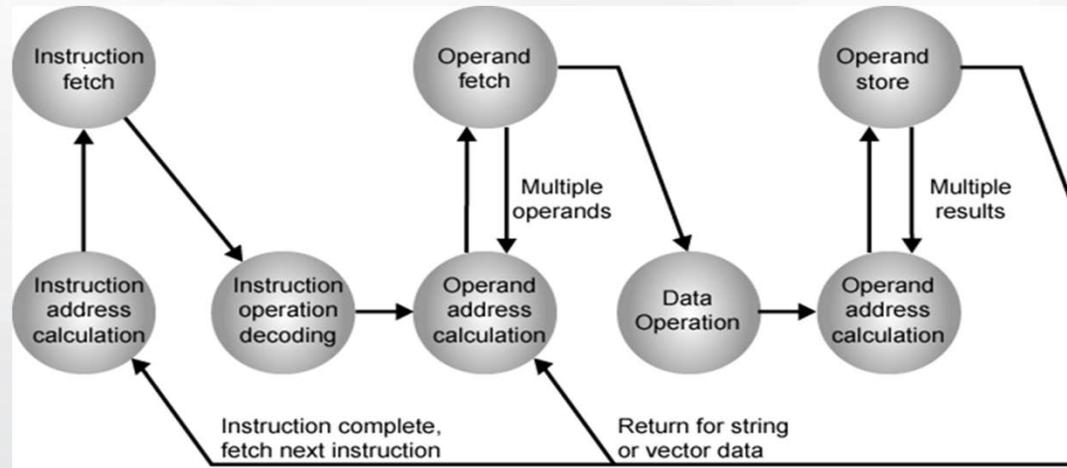




Siklus Dasar Instruksi



Siklus Eksekusi secara Utuh



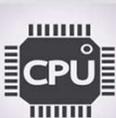
- Instruction address calculation : menentukan alamat instruksi berikutnya yang akan dieksekusi.
- Instruction fetch : membaca instruksi dari lokasi memori kedalam prosesor.
- Instruction operation decoding : menganalisis instruksi untuk menentukan jenis operasi yang akan dilakukan dan operand yang akan digunakan.
- Operand address calculation : jika operasi melibatkan referensi ke operand di dalam memori atau tersedia via I/O, kemudian menentukan alamat operand.
- Operand fetch : mengambil operand dari memori atau membaca operand itu dari I/O
- Data operation : bentuk operasi yang ditandai didalam instruksi.
- Operand store : menulis hasilnya ke dalam memori atau keluar untuk I/O.



Interrupt

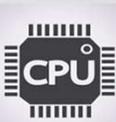
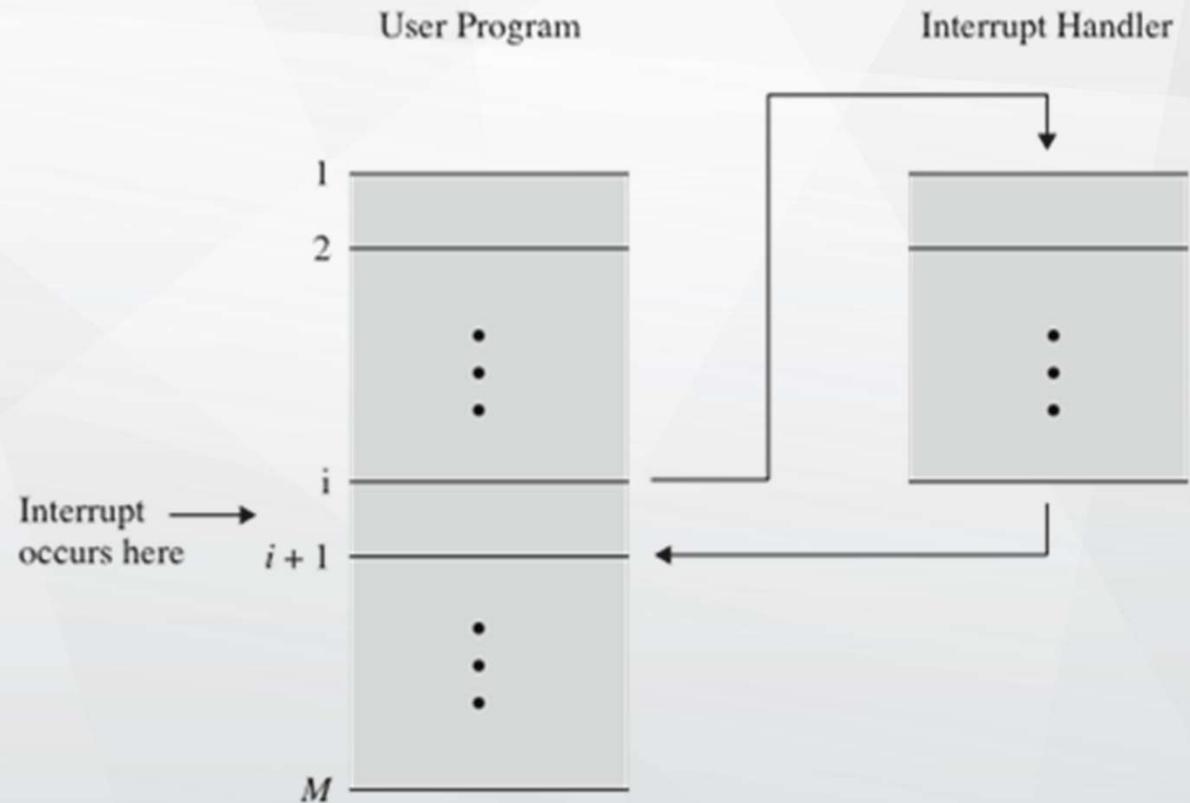
- Secara virtual semua computer menyediakan mekanisme yang membuat modul-modul lainnya (I/O, memori) dapat menginterupsi pengolahan normal prosesor.

| | |
|-------------------------|--|
| Program | Generated by some condition that occurs as a result of an instruction execution, such as arithmetic overflow, division by zero, attempt to execute an illegal machine instruction, or reference outside a user's allowed memory space. |
| Timer | Generated by a timer within the processor. This allows the operating system to perform certain functions on a regular basis. |
| I/O | Generated by an I/O controller, to signal normal completion of an operation, request service from the processor, or to signal a variety of error conditions. |
| Hardware failure | Generated by a failure such as power failure or memory parity error. |



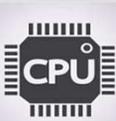
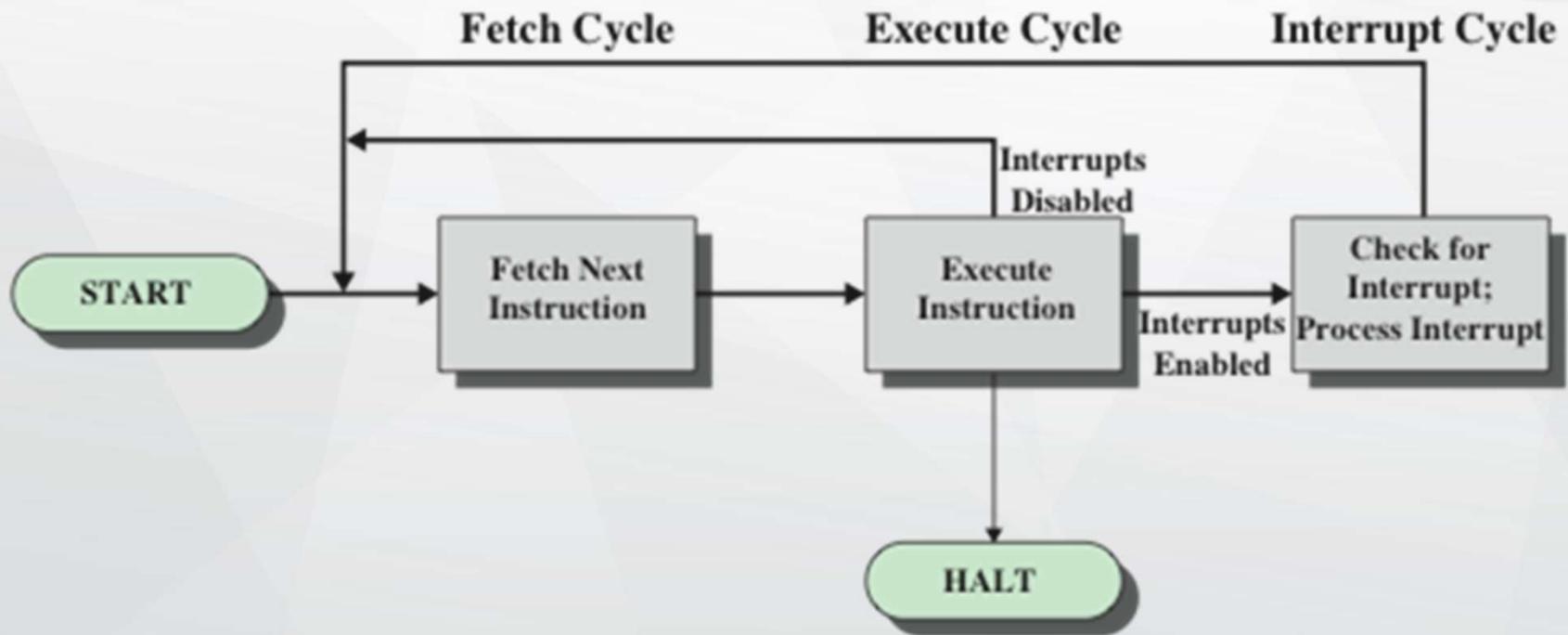


Instruction Cycle with Interrupts

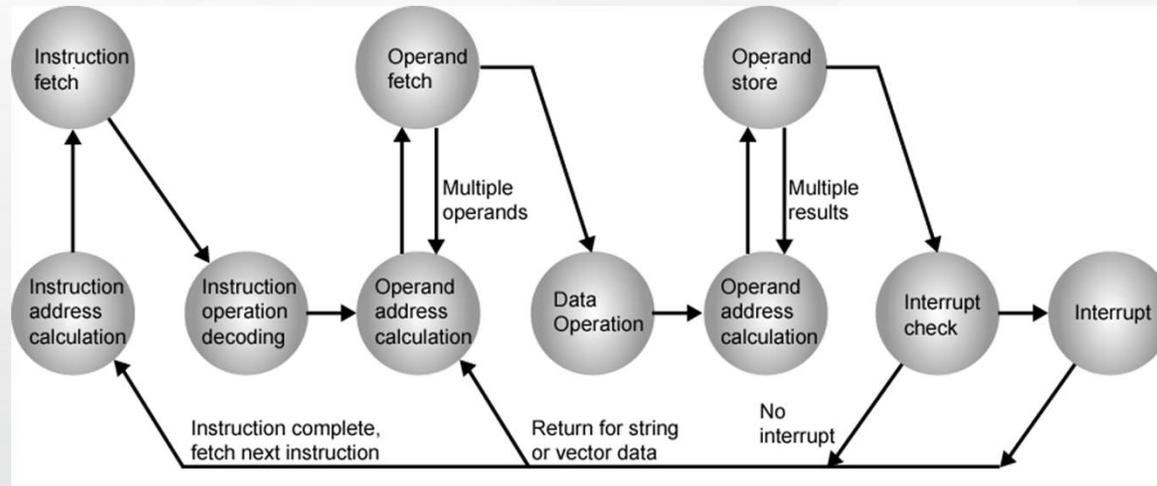




Interrupt Instruction Cycle



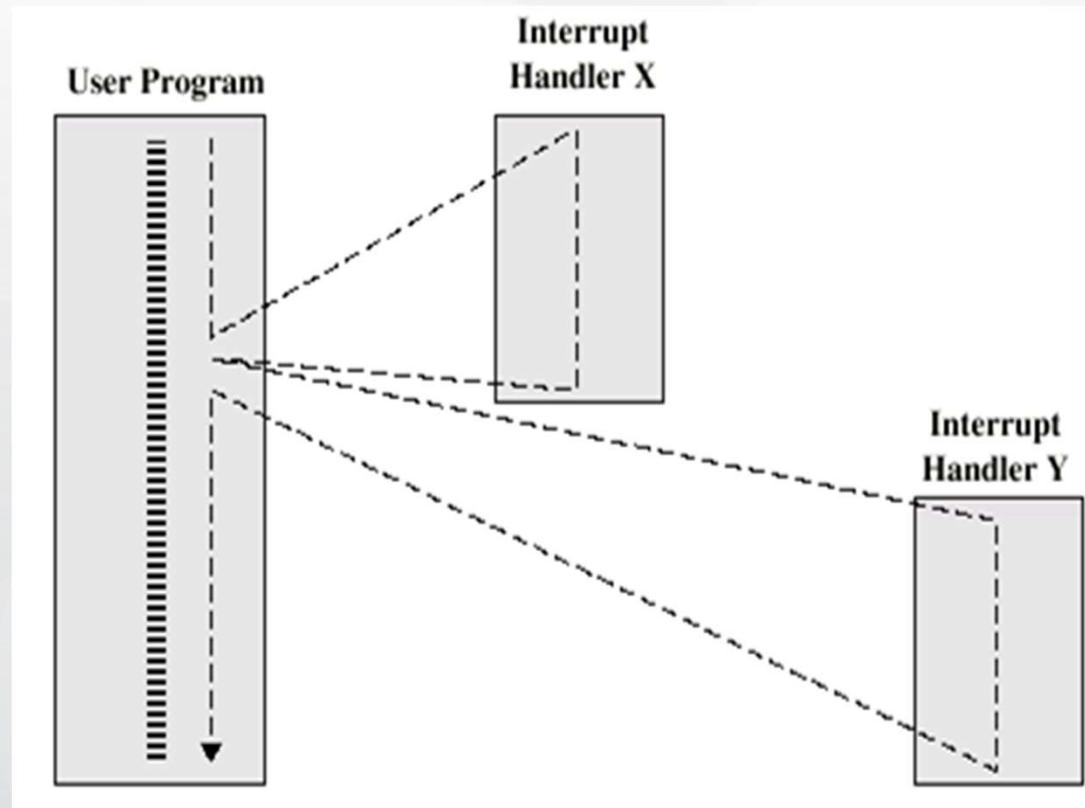
Interrupt Instruction Cycle State Diagram



- Prosesnya yaitu prosesor memproses siklus fetch dan mengambil instruksi pertama dalam program interrupt handler.
- Program interrupt handler merupakan bagian umum dari sistem operasi.
- Program ini menentukan sifat interrupt dan melakukan tindakan yang dibutuhkan atau diperlukan.
- Sebagai contoh, handler menentukan modul I/O yang dihasilkan interrupt, dan mungkin membagi program yang akan me write data lebih banyak untuk modul I/O tersebut.
- Ketika rutin interrupt handler selesai, prosesor dapat melanjutkan eksekusi program pada titik/point interupsi.



Multiple Interrupts





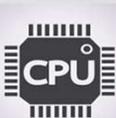
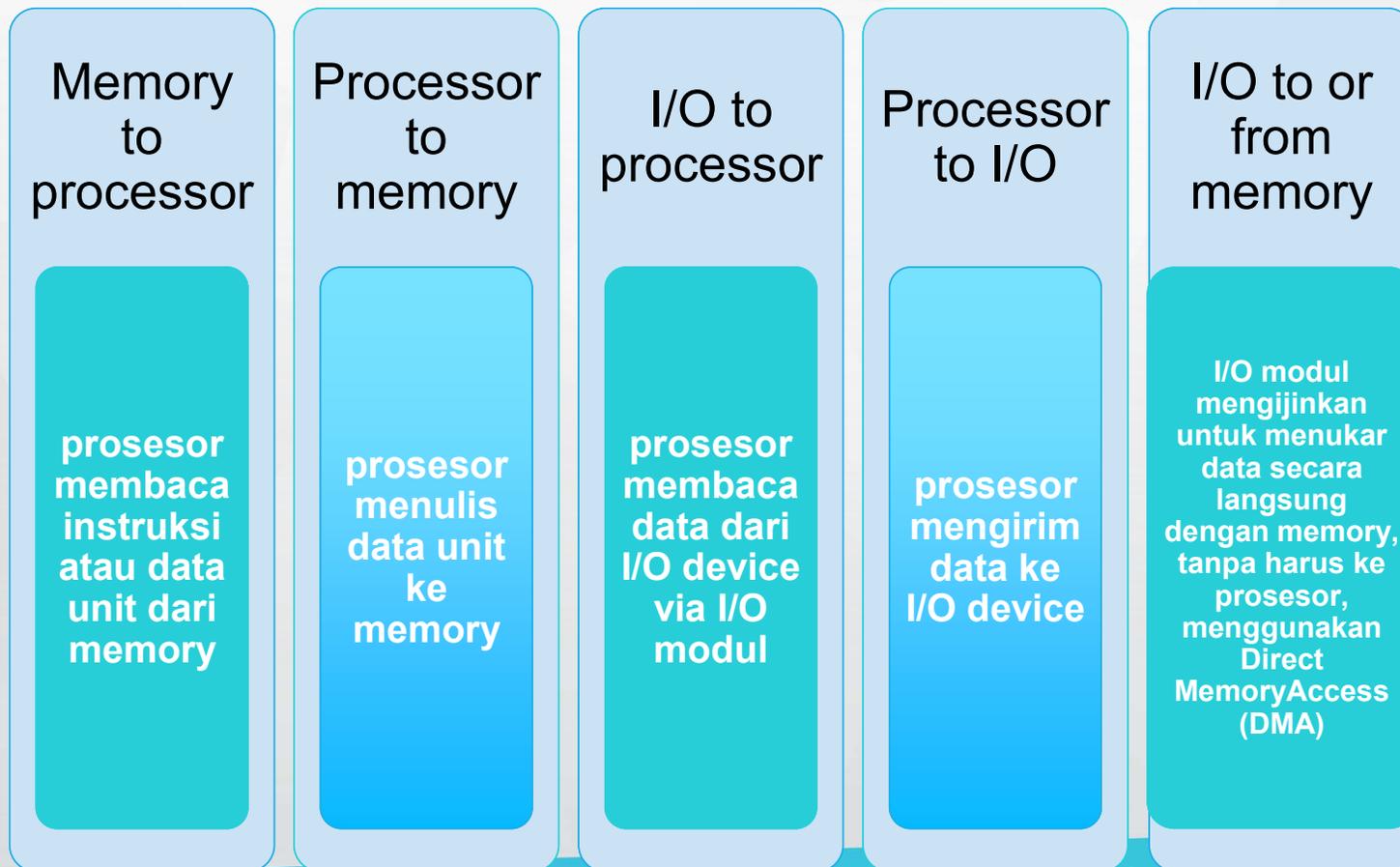
Struktur Interkoneksi

- Komputer terdiri dari sebuah set komponen atau modul-modul dari 3 tipe dasar yaitu prosesor, memory, I/O yang saling berkomunikasi satu sama lain.
- Sehingga komputer adalah jaringan dari dasar modul. Jadi harus ada jalan untuk menghubungkan modul-modul tersebut.
- Jadi jalur-jalur yang menghubungkan berbagai modul tersebut dinamakan struktur interkoneksi.
- Desain struktur ini akan tergantung dari pertukaran yang harus dilakukan antar modul.





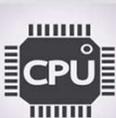
Struktur Transfer Interkoneksi yang harus dimiliki oleh CPU





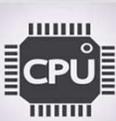
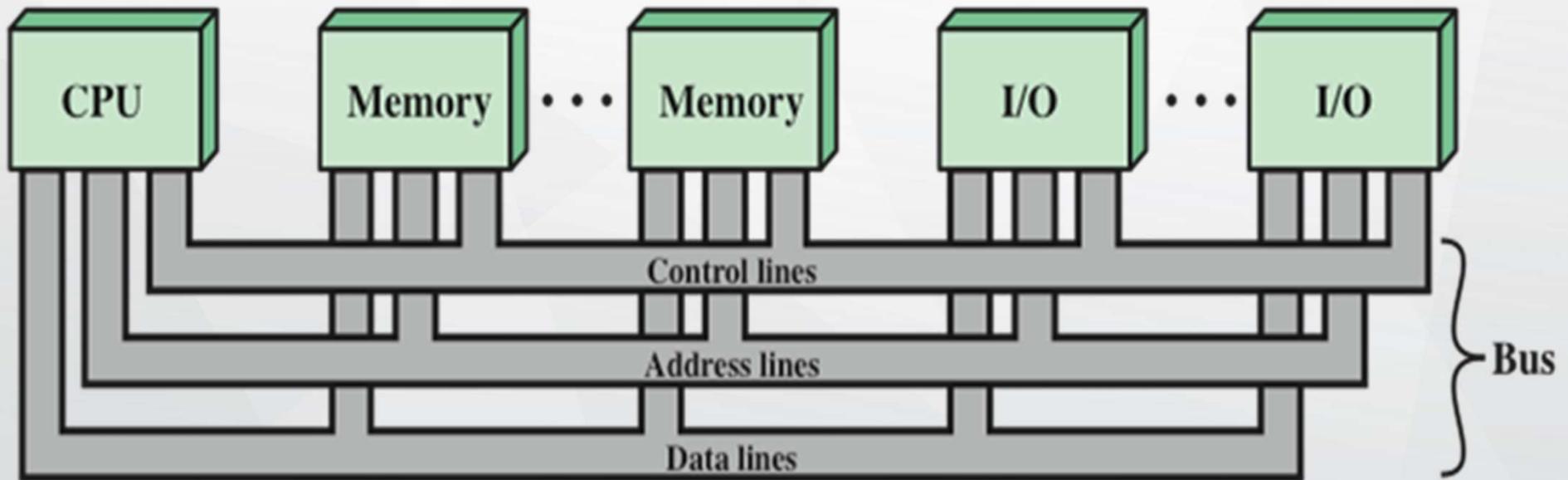
Bus Interconnection Scheme

- Sistem komputer mengandung sejumlah bus yang berbeda yang menyediakan jalurantara komponen pada berbagai tingkat hirarki sistem komputer.
- Sebuah bus adalah jalur komunikasi yang menghubungkan dua atau lebih perangkat.
- Karakteristik kunci bus adalah bahwa itu adalah media transmisi bersama. Beberapa perangkat terhubung ke bus, dan sinyal ditransmisikan oleh setiap perangkat yang satu ini tersedia untuk penerimaan oleh semua perangkat lain yang melekat pada bus.
- Jika dua perangkat mengirimkan selama periode waktu yang sama, sinyal mereka akan tumpang tindih dan menjadi kacau.





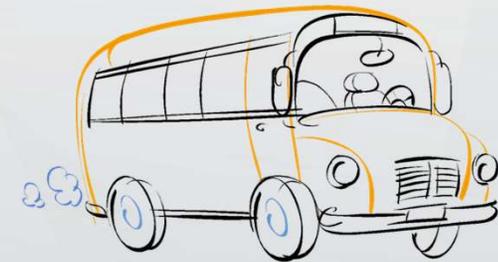
Bus Interconnection Scheme





Data Bus

- Jalur data Bus menyediakan jalur untuk memindahkan data di antara sistem
- Mungkin terdiri dari 32, 64, 128, atau lebih baris terpisah
- Jumlah baris disebut sebagai lebar bus data
- Jumlah baris menentukan berapa banyak bit yang dapat ditransfer dalam satu waktu
- Lebar bus data adalah faktor kunci dalam menentukan secara keseluruhan kinerja sistem





Address Bus

- Digunakan untuk menentukan sumber atau tujuan Data Bus
- Jika prosesor ingin membaca sebuah data dari data dari memori, ia meletakkan alamat dari data yang diinginkan pada baris alamat
- Lebar Address Bus menentukan kemungkinan kapasitas memori maksimum dari sistem
- Juga digunakan untuk menangani port I / O
- Bit urutan yang lebih tinggi digunakan untuk memilih modul tertentu di bus dan bit urutan yang lebih rendah memilih lokasi memori atau port I / O dalam modul





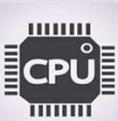
Control Bus

- Digunakan untuk mengontrol akses dan penggunaan Data dan Address Bus
- Karena Data dan Address Bus digunakan bersama oleh semua komponen, pasti ada cara untuk mengontrol penggunaannya.
- Sinyal kontrol mengirimkan informasi perintah dan waktu di antara modul sistem.
- Sinyal waktu menunjukkan validitas data dan informasi alamat.
- Sinyal perintah menentukan operasi yang akan dilakukan.





Performance Issue





Moore's Law

1965; Gordon Moore – co-founder Intel

Jumlah transistor yang diamati yang dapat diletakkan pada satu chip meningkat dua kali lipat setiap tahun

Kecepatannya melambat menjadi dua kali lipat setiap 18 bulan pada tahun 1970-an, tetapi terus berlanjut hingga saat ini

Konsekuensi dari Moore's law:

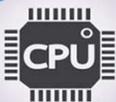
Biaya logika komputer dan sirkuit memori telah turun drastis

Panjang jalur listrik dipersingkat, meningkatkan kecepatan operasi

Komputer menjadi lebih kecil dan lebih nyaman digunakan di berbagai lingkungan

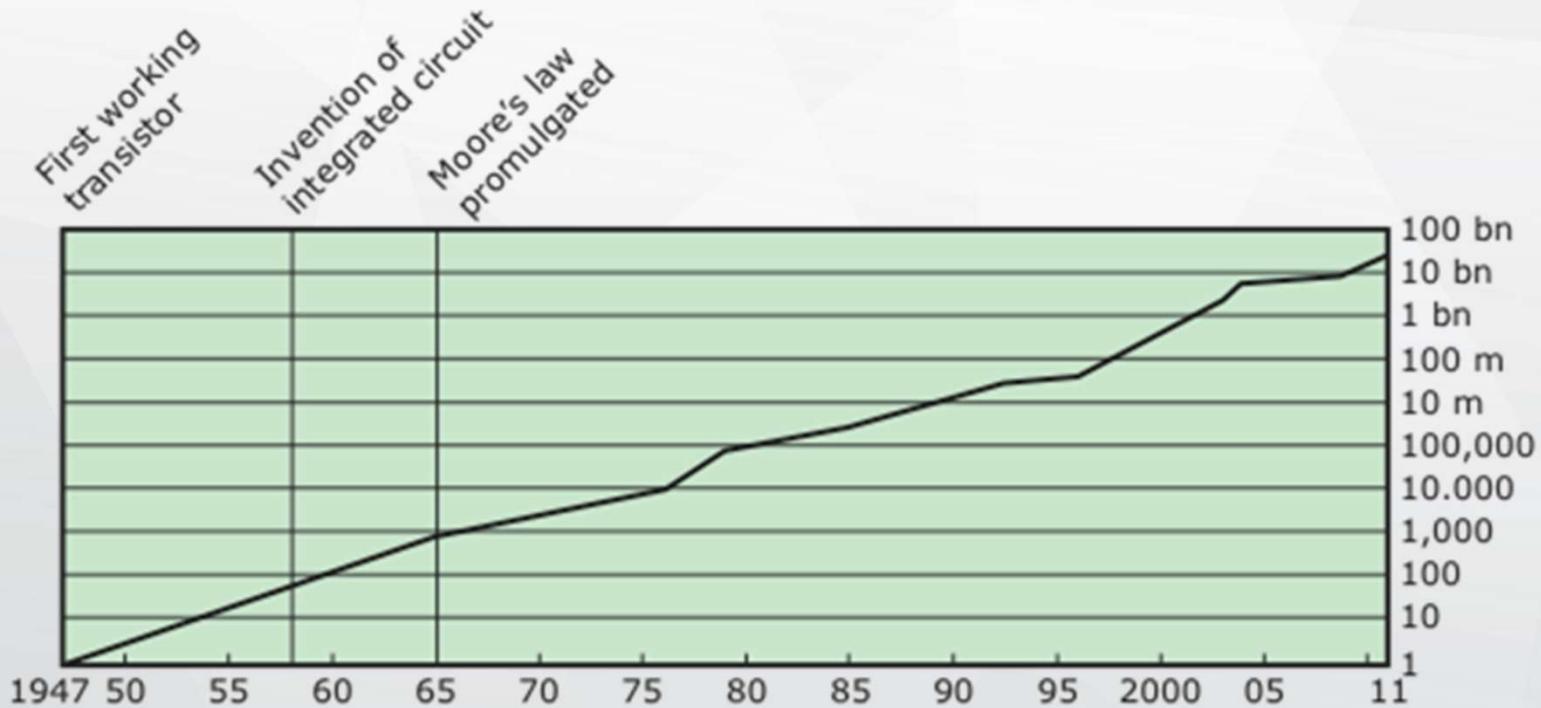
Pengurangan kebutuhan daya dan pendinginan

Lebih sedikit interchip connections

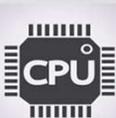




Pertumbuhan Chip pada Komputer



Pertumbuhan transistor dihitung dari IC yang dipergunakan





Later Generations

LSI : Large Scale Integration
VLSI : Very Large Scale Integration
ULSI : Ultra Large Scale Integration

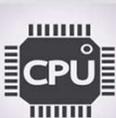




Microprocessor Speed

Teknik yang dibangun ke dalam prosesor kontemporer meliputi:

| | |
|-----------------------|--|
| Pipelining | <ul style="list-style-type: none">• Prosesor memindahkan data atau instruksi ke dalam pipa konseptual dengan semua tahapan pemrosesan secara bersamaan |
| Branch prediction | <ul style="list-style-type: none">• Prosesor melihat ke depan dalam kode instruksi yang diambil dari memori dan memprediksi kelompok instruksi mana yang kemungkinan besar akan diproses selanjutnya |
| Data flow analysis | <ul style="list-style-type: none">• Prosesor menganalisis instruksi mana yang bergantung pada hasil satu sama lain, atau data, untuk membuat jadwal instruksi yang dioptimalkan |
| Speculative execution | <ul style="list-style-type: none">• Dengan menggunakan prediksi cabang dan analisis aliran data, beberapa prosesor secara spekulatif menjalankan instruksi sebelum kemunculannya yang sebenarnya dalam eksekusi program, menahan hasilnya di lokasi sementara, menjaga mesin eksekusi sesibuk mungkin. |
| | |
| | |
| | |





Terima Kasih

Pustaka : William Stallings, “Computer Organization and Architecture Designing for Performance Eighth Edition”, Prentice Hall, 2019

