
Arsitektur & Organisasi Komputer



STRUKTUR FUNGSI CPU

Tujuan

- ❑ Menjelaskan tentang komponen utama CPU dan Fungsi CPU
- ❑ Membahas struktur dan fungsi internal prosesor, organisasi ALU, control unit dan register
- ❑ Menjelaskan fungsi prosesor dalam menjalankan instruksi-instruksi mesin

CPU

- ❑ Central Processing Unit
- ❑ Merupakan komponen terpenting dari sistem komputer
- ❑ Komponen pengolah data berdasarkan instruksi yang diberikan kepadanya
- ❑ Dalam mewujudkan fungsi dan tugasnya, CPU tersusun atas beberapa komponen

Komponen Utama CPU

- ❑ *Arithmetic and Logic Unit (ALU)*
- ❑ *Control Unit*
- ❑ *Registers*
- ❑ *CPU Interconnections*

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 – fungsinya.
- ❑ Termasuk dalam tanggung jawab unit kontrol adalah mengambil instruksi – instruksi dari memori utama dan menentukan jenis instruksi tersebut.

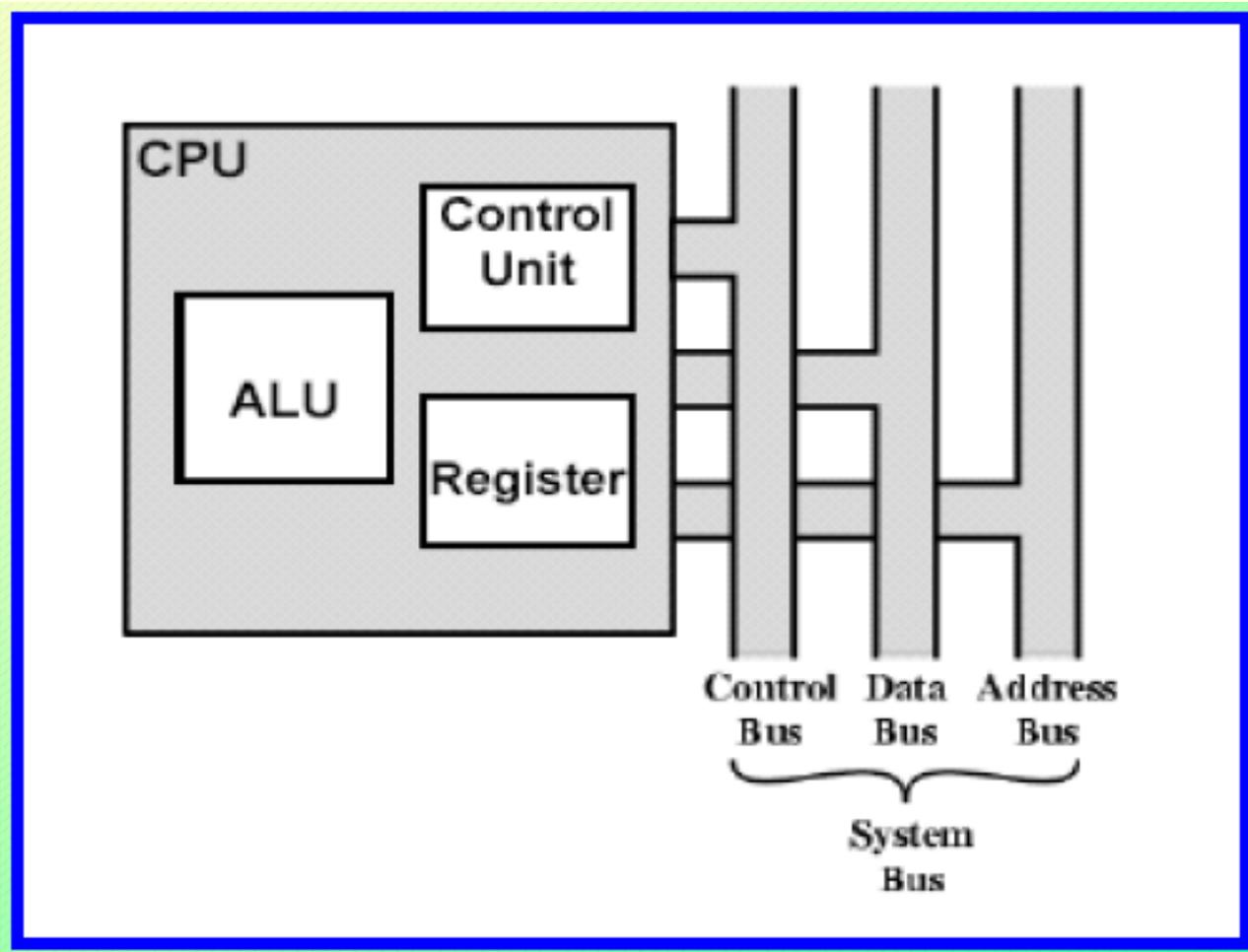
Registers

- ❑ Media penyimpan internal CPU yang digunakan saat proses pengolahan data.
- ❑ Memori ini bersifat sementara, biasanya digunakan untuk menyimpan data saat diolah ataupun data untuk pengolahan selanjutnya.

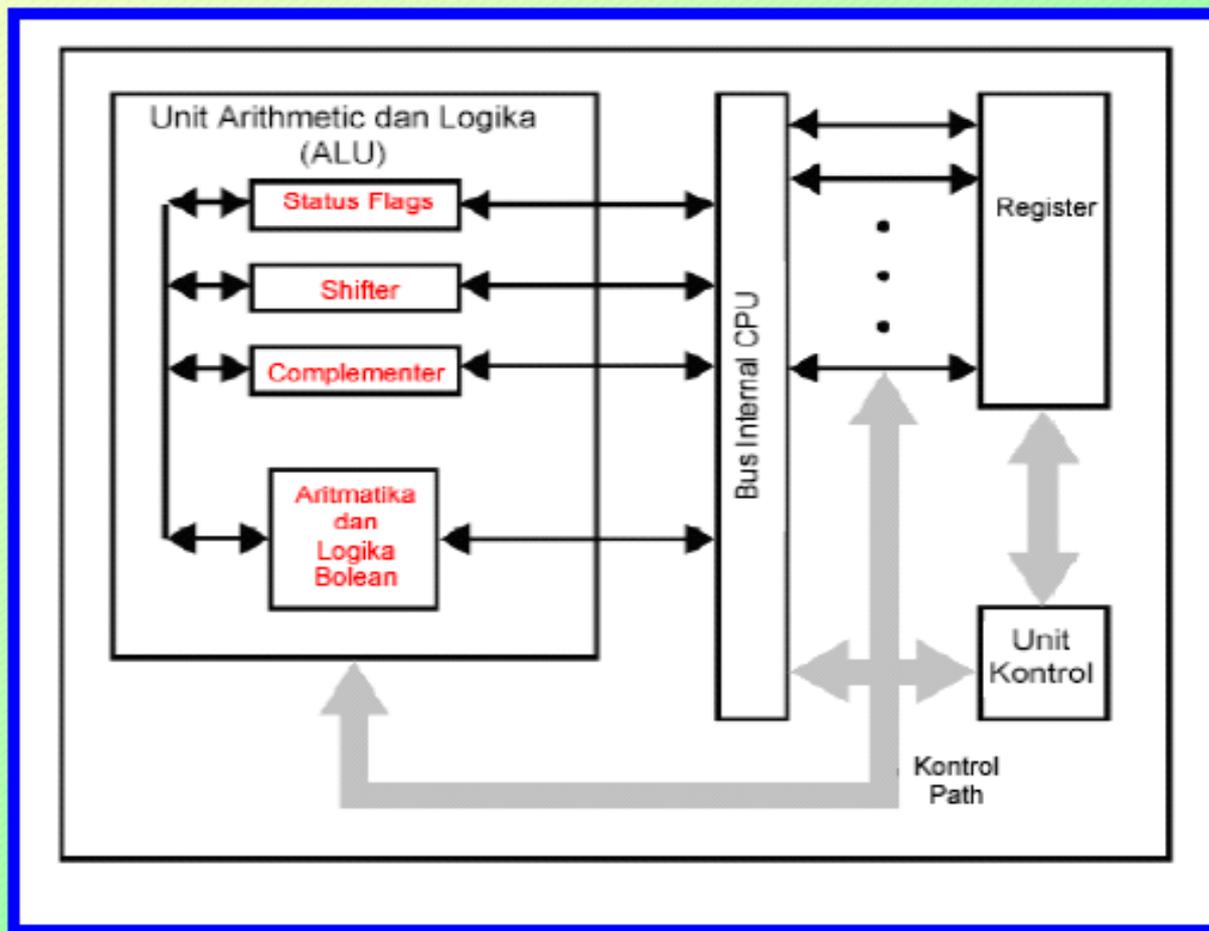
CPU Interconnections

- ❑ Sistem koneksi dan bus yang menghubungkan komponen internal dan bus – bus eksternal CPU
- ❑ Komponen internal CPU yaitu ALU, unit kontrol dan register – register.
- ❑ Komponen eksternal CPU : sistem lainnya, seperti memori utama, piranti masukan/keluaran

Komponen internal CPU



Struktur detail internal CPU



Fungsi CPU

- ❑ Menjalankan program – program yang disimpan dalam memori utama dengan cara mengambil instruksi – instruksi, menguji instruksi tersebut dan mengeksekusinya satu persatu sesuai alur perintah.
- ❑ 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)

Kenapa ada Program?

- Hardwired system (sistem yang instruksinya dikendalikan oleh kombinasi hardware) → tidak fleksibel
Kabel-kabel, jumper dsb → Rumit
- Dibuatlah general purpose hardware yang bisa menerima **control signal**.
- Jenis control signal menentukan instruksi yang akan dijalankan

Apakah Program itu?

- Serangkaian langkah-langkah (a sequence of steps)
- Pada tiap langkah, dilakukan sebuah operasi logika atau aritmetik
- Untuk tiap-tiap operasi, disediakan sebuah kode instruksi yang unik.
- Satu kode instruksi mengandung satu set **control signal** yang dapat dimengerti oleh hardware

Eksekusi Program → Siapa saja yang terlibat?

- Control Unit di CPU → mengatur aliran program
- Memory → menyimpan baris-baris program yang akan dieksekusi dan hasil outputnya
- Arithmetic and Logic Unit di CPU → untuk melakukan operasi aritmetik dan logika jika diperintahkan oleh program

Sekilas – Bagaimana Program disimpan di memory

- Kira-kira beginilah format penyimpanan program di memory:

Memory

Alamat	Data
0000	1003
0001	3004
0002	5070
0003	4000

n-bit alamat

m-bit instruksi

: x – bit kode instruksi

y – bit data

Data : bisa data angka atau alamat

0000 1 003

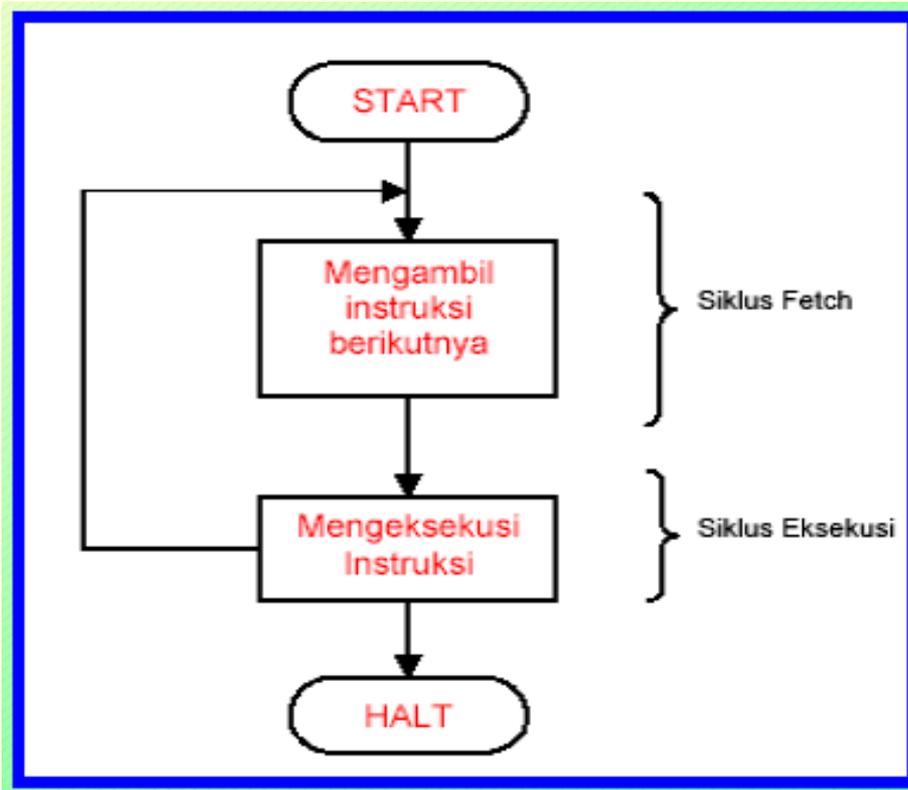
4 Bit alamat, 1 bit kode instruksi, 3 bit data

Sekilas Kedua – Apa sih isinya Control Unit?

- PC: Program Counter
Berisi alamat memory untuk mengambil instruksi selanjutnya
- IR: Instruction Register
Menampung instruksi yang diambil dari memory
- AC: Accumulator
Penampung sementara untuk data hasil eksekusi instruksi

Siklus Instruksi

- Terdiri dari siklus fetch dan siklus eksekusi



Siklus Instruksi

Siklus instruksi merupakan proses yang terjadi didalam CPU yaitu sebuah siklus yang diawali dengan pembacaan instruksi dari memori. Setiap kali terjadi pembacaan memori maka Program Counter (PC) akan menambah satu hitungan. Selanjutnya Instruksi-instruksi tersebut akan diisikan didalam register instruksi (IR) yang kemudian akan diterjemahkan oleh CPU untuk menjalankan proses yang diperintahkan.

Siklus Instruksi

Proses yang diperintahkan tersebut dikelompokan :

1. CPU – memori, perpindahan data dari CPU ke memori dan sebaliknya
2. CPU – I/O, perpindahan data dari CPU ke modul I/O dan sebaliknya.
3. Pengolahan data, CPU membentuk sejumlah operasi aritmatika dan logika terhadap data.
4. Kontrol, merupakan instruksi untuk melakukan kontrol fungsi atau kerja. Siklus eksekusi untuk suatu instruksi dapat melibatkan lebih dari sebuah referensi ke memori.

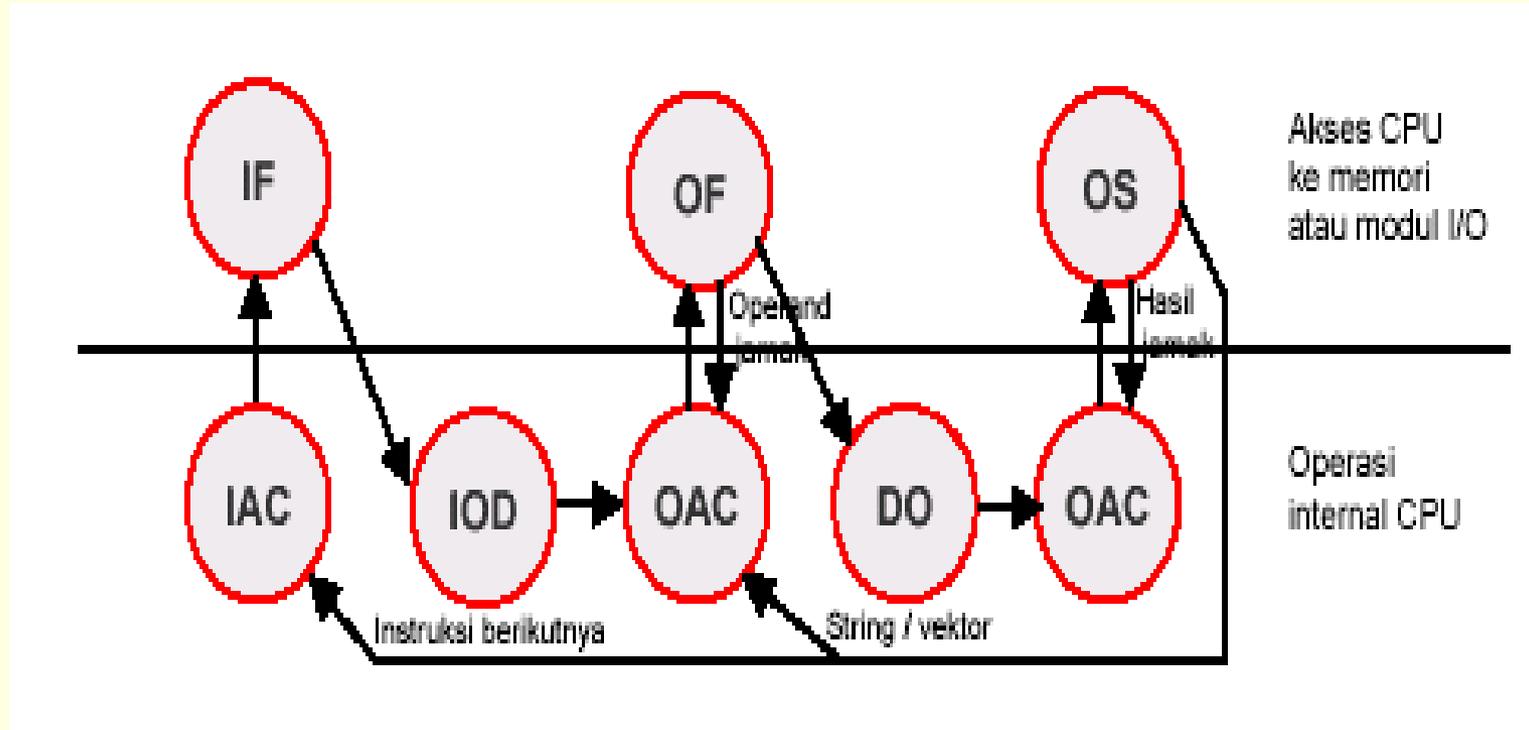
Siklus Instruksi

5. Instruction Address Calculation (IAS) menentukan alamat instruksi berikutnya yang akan dieksekusi.
6. Instruction Fetch (IF) membaca atau mengambil instruksi dari lokasi memori ke CPU
7. Instruction Operation Decoding (IOD) menganalisa intruksi untuk menentukan jenis operasi yang akan dibentuk dan operand yang akan digunakan.
8. Operand Address Calculation (OAC) menentukan alamat operand apabila melibatkan referensi operand pada memori.

Siklus Instruksi

9. Operand Fetch (OF) mengambil operand dari memori atau dari modul I/O
10. Data Operation (DO) membentuk operasi yang diperintahkan dalam instruksi
11. Operand Store (OS) menyimpan hasil eksekusi ke dalam memori.

Siklus Instruksi



Gambar Diagram Siklus Instruksi

Fetch Cycle

- Program Counter (PC) berisi alamat instruksi yang akan dibaca
- Processor membaca instruksi dari alamat memory yang ditunjuk PC
- Increment PC ($PC = PC + 1$) - kecuali ada perintah lain
- Instruksi di-load ke Instruction Register (IR)
- Processor menterjemahkan isi instruksi dan memerintahkan aksi yang sesuai dg isi instruksi

Execute Cycle

- Transfer data antara CPU dengan main memory
- Transfer data antara CPU dengan modul I/O
- Data processing (operasi aritmetik / logika terhadap data oleh CPU)
- Control
 - Perubahan urutan program (merubah isi PC) dg instruksi JUMP
 - Menghentikan program di tengah jalan dg insruksi HALT
- Kombinasi dari hal-hal diatas

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

Siklus Fetch - Eksekusi

- ❑ Instruksi – instruksi yang dibaca akan dimuat dalam register instruksi (IR).
- ❑ Instruksi – instruksi ini dalam bentuk kode – kode biner yang dapat diinterpretasikan oleh CPU kemudian dilakukan aksi yang diperlukan

Aksi CPU

- ❑ *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.

Elemen Siklus Eksekusi

- ❑ **Instruction Address Calculation (IAC)**, yaitu mengkalkulasi atau menentukan alamat instruksi berikutnya yang akan dieksekusi. Biasanya melibatkan penambahan bilangan tetap ke alamat 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 eksekusi ke dalam memori

Interrupt – Apakah itu?

- Suatu instruksi dalam program kadang hasilnya tidak sesuai yang diinginkan
- Ada hambatan yang berasal dari:
 - Kekeliruan matematis (overflow, division by zero)
 - Proses I/O (menulis, membaca data)
 - Kegagalan hardware
 - Timer CPU → Penjadwalan

Interrupt – No Problem?

- Kalau hambatan muncul seketika, bisa langsung dihandle lewat program itu sendiri, misal dengan menyiapkan percabangan berdasarkan kondisi
- Bagaimana kalau eksekusi sebuah instruksi itu butuh waktu? Apakah harus menunggu hasilnya, error atau tidak?

Interrupt – terus bagaimana ya?

- CPU membiarkan sebuah instruksi dieksekusi
- CPU melanjutkan membaca instruksi berikutnya
- TETAPI sebelum membaca instruksi berikutnya, CPU memeriksa, adakah sinyal interrupt yang masuk dari instruksi yang tadi?

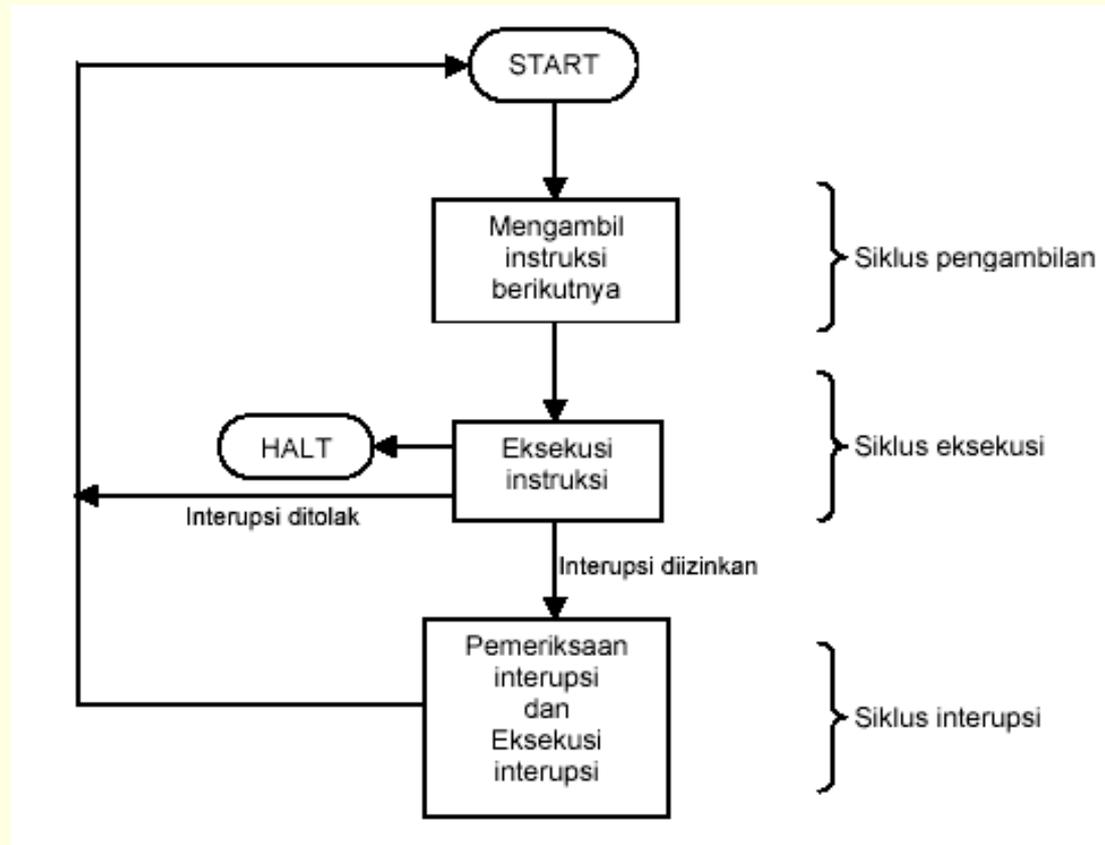
Fungsi Interrupt

- Fungsi interupsi adalah mekanisme penghentian atau pengalihan pengolahan instruksi dalam CPU kepada routine interupsi.
- 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 overflow, 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.

Siklus instruksi dengan interrupt



Interrupt Handling - Details

- Sebelum membaca instruksi berikutnya, CPU memeriksa, ada sinyal interrupt atau tidak
- Jika tidak ada interrupt, baca instruksi berikutnya
- Jika ada interrupt, arahkan PC ke alamat instruksi Interrupt Handler, dan hentikan program yg sedang berjalan untuk sementara.
- Jika handling sudah selesai, arahkan kembali PC ke alamat instruksi berikutnya yang seharusnya dibaca.

Apabila interupsi ditanggguhkan, maka

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) ke alamat awal routine *interrupt handler*.

Interupsi Ganda

1. Menolak atau tidak mengizinkan interupsi lain saat suatu interupsi ditangani prosesor. Kemudian setelah prosesor selesai menangani suatu interupsi maka interupsi lain baru ditangani.

Disebut *pengolahan interupsi berurutan / sekuensial*.

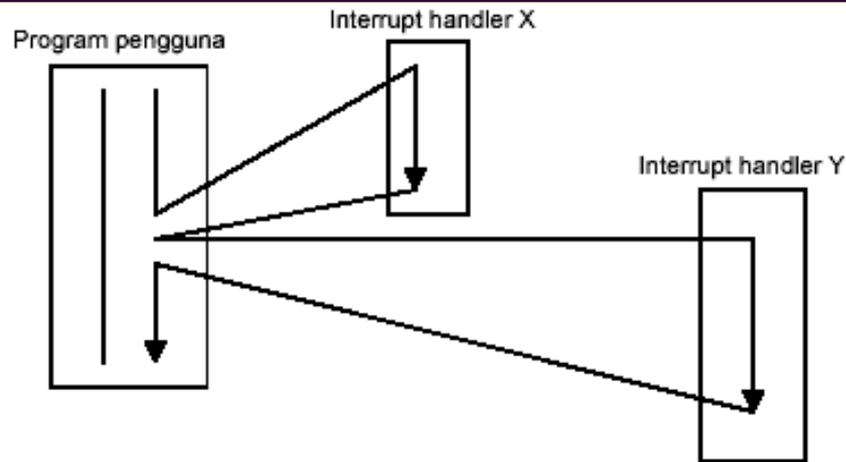
Pendekatan ini cukup baik dan sederhana karena interupsi ditangani dalam urutan yang cukup ketat.

Kelemahan → tidak memperhitungkan prioritas interupsi. Diperlihatkan pada gambar 3.6a.

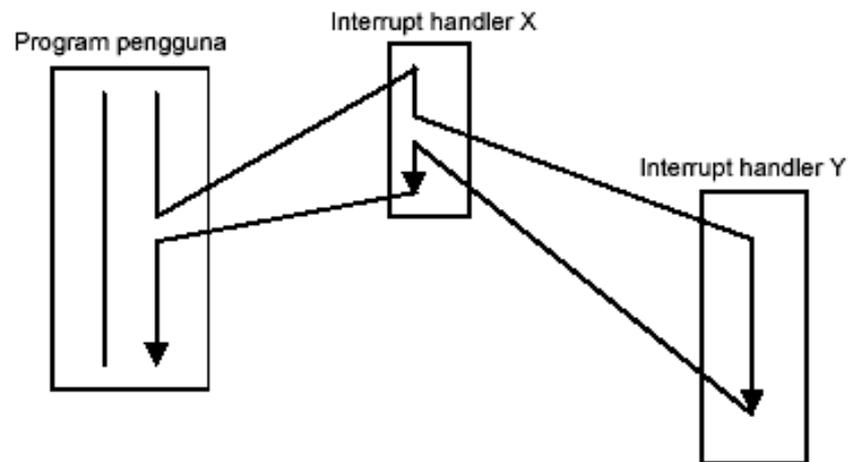
2. Mendefinisikan prioritas bagi interupsi dan *interrupt handler* mengizinkan interupsi berprioritas lebih tinggi ditangani terlebih dahulu.

Disebut *pengolahan interupsi bersarang*, diperlihatkan pada gambar 3.6b.

Penanganan Interupsi Ganda



(a) Pengolahan interupsi sekuensial



(b) Pengolahan interupsi bersarang

Contoh Pendekatan Bersarang

- Suatu sistem memiliki tiga perangkat I/O: printer, disk, dan saluran komunikasi, dengan prioritas masing – masing 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.

Jika Interrupt yang muncul banyak?

- Disable interrupts
 - Processor will ignore further interrupts while processing one interrupt
 - Interrupts remain pending and are checked after first interrupt has been processed
 - Interrupts handled in sequence as they occur
- Define priorities
 - Low priority interrupts can be interrupted by higher priority interrupts
 - When higher priority interrupt has been processed, processor returns to previous interrupt



TERIMAKASIH

Kuis Struktur Fungsi CPU – Pertemuan ke-13

1. Gambarkan komponen Internal CPU dan Jelaskan?
2. Jelaskan apa yang dimaksud dengan RISC dan CISC?

Note: Kerjakan di selembar kertas/buku catatan mata kuliah Arsikom kalian!