STRUKTUR CPU

Arsitektur Komputer

Tujuan

- Mengerti struktur dan fungsi CPU yaitu dapat melakukan Fetch instruksi, interpreter instruksi , Fetch data, eksekusi, dan menyimpan kembali.

  serta struktur dari register, macam-macam register dan fungsinya.
- Mengerti aliran data pada siklus pengambilan, siklus tak langsung, siklus interupt.
- Mengerti pipelining, dan mengerti teknik-teknik menangani percabangan pada pipelining.

Materi

- Bagian ini membahas aspek-aspek struktur dan fungsi CPU untuk dasar pembahasan berikutnya, yaitu RISC.
- Fokus bab struktur dan fungsi CPU adalah organisasi prosesor dan register, siklus instruksi dan strategi dalam metode pipelining.

CPU

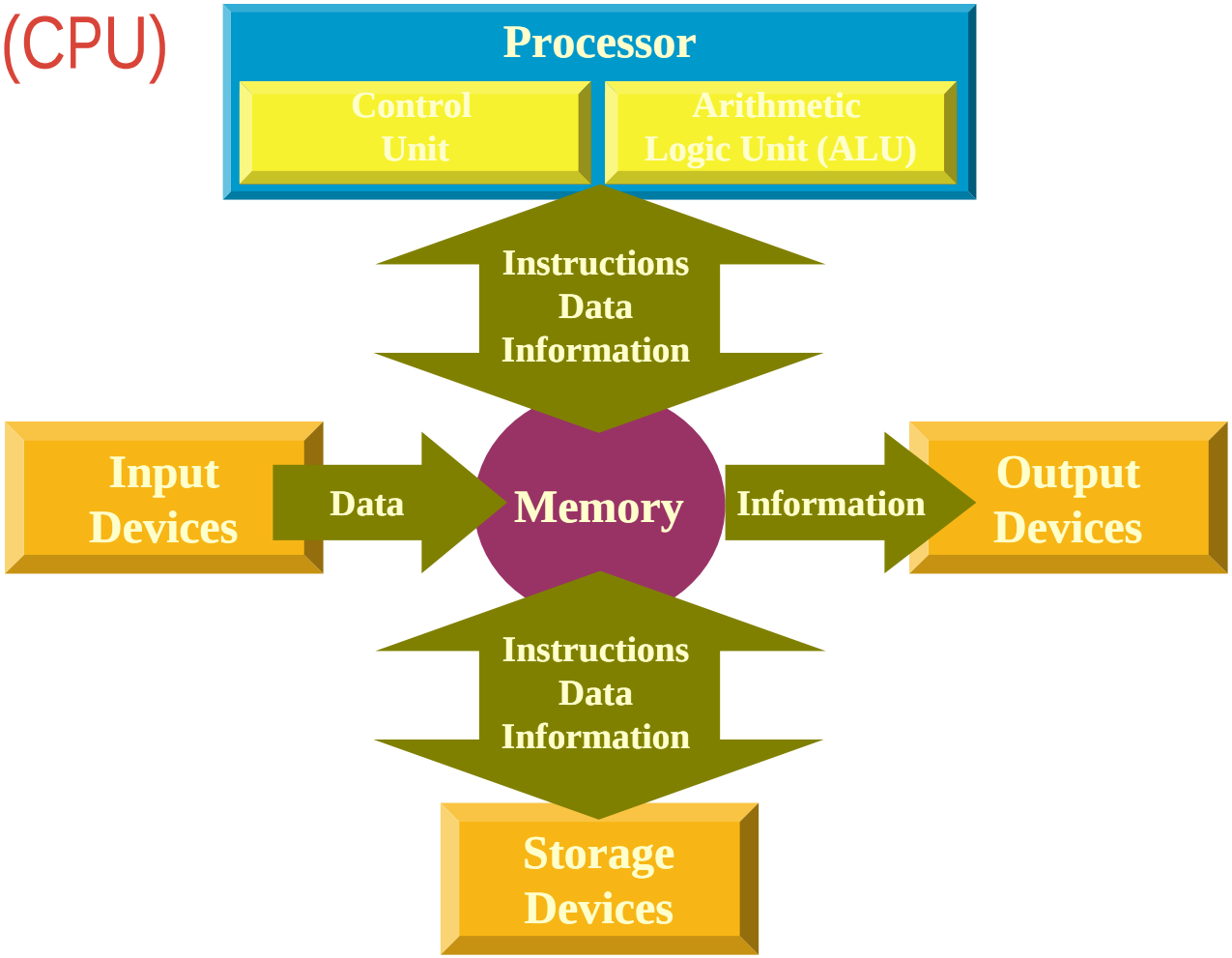
- Central Processing Unit
- Merupakan komponen terpenting dari sistem komputer
- komponen pengolah data berdasarkan instruksi yang diberikan kepadanya
- Dalam mewujudkan fungsi dan tugasnya, CPU tersusun atas beberapa komponen

Komponen Utama CPU

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

Prosesor

- *Central Processing Unit* (CPU)
  - Menginterpretasikan dan mengerjakan instruksi-instruksi dasar operasi komputer
  - Disebut juga prosesor



Arithmetic and Logic Unit (ALU)

- Bertugas membentuk fungsi – fungsi pengolahan data komputer.
- ALU sering disebut bahasa mesin (machine language) karena bagian ini mengerjakan instruksi – instruksi bahasa mesin yang diberikan padanya.
- ALU terdiri dari dua bagian, yaitu unit arithmetika 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 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.

Organisasi Register

- Sistem komputer menggunakan hirarki memori
- Pada tingkatan yang lebih atas, memori yang lebih cepat, lebih kecil dan lebih mahal.
- Didalam CPU terdapat sekumpulan register yang tingkatan memorinya berada diatas hirarki memori utama dan cache.
- Apa fungsi register pada CPU?

Fungsi Register CPU

- User Visible Register
  - Register ini memungkinkan pemogram bahasa mesin dan bahasa assembler meminimalkan referensi main memory dengan cara mengoptimasi penggunaan register
- Control and Status Register
  - Register ini digunakan oleh *Control Unit* untuk mengontrol operasi CPU dan oleh program sistem operasi untuk mengontrol eksekusi program

- Tidak terdapat pemisahan yang jelas diantara kedua jenis register diatas

- Jumlah register sangat menentukan kinerja suatu prosesor
- Jumlah register juga berpengaruh pada rancangan set instruksi karena register yang lebih banyak akan memerlukan bits operand specifier yang lebih banyak pula
- Register yang berukuran antara 8 hingga 32 bit dapat dikatakan optimum

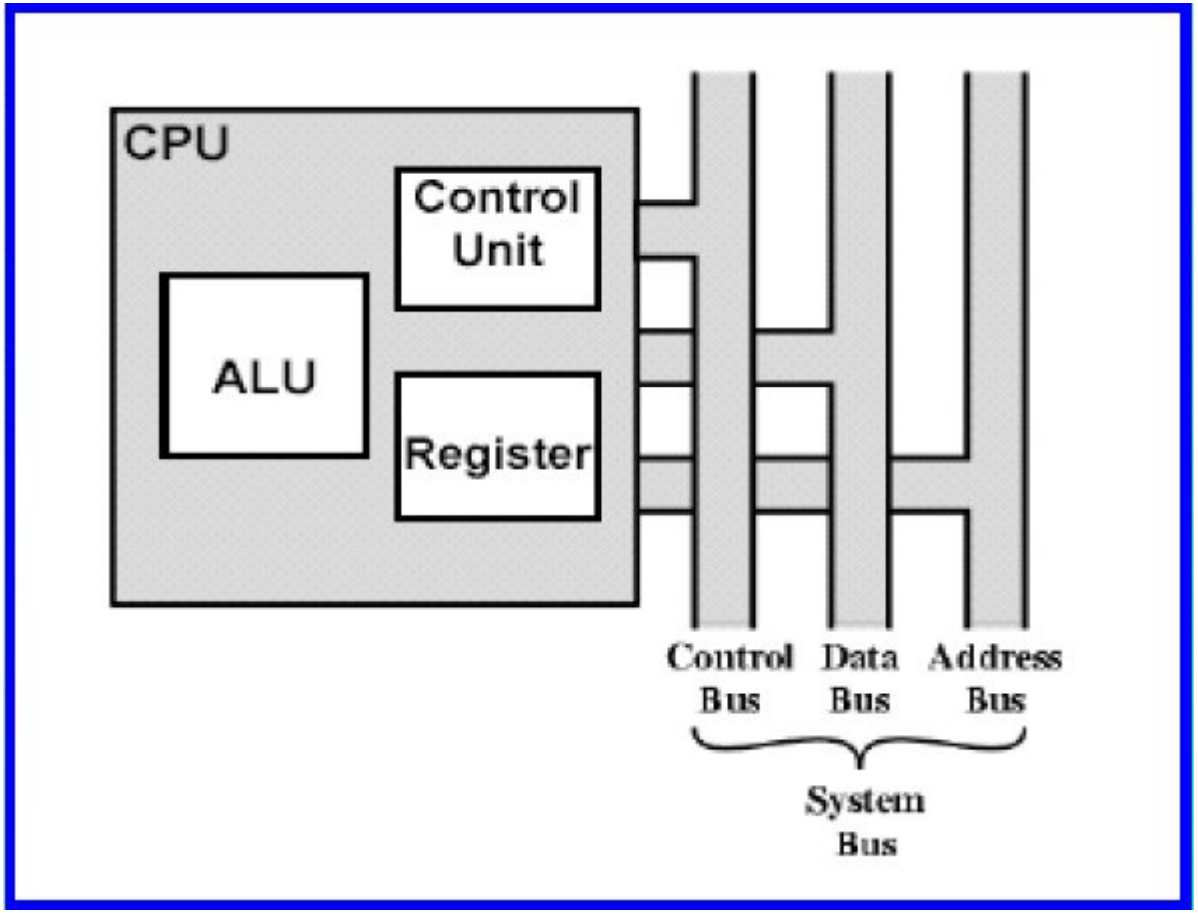
Register yang penting bagi eksekusi instruksi

- Program Counter (PC) atau pencacah program
  - berisi alamat instruksi yang akan diambil
- Instruction Register (IR)
  - berisi instruksi yang terakhir diambil
- Memori Address Register (MAR)
  - berisi alamat sebuah lokasi didalam memori
- Memori Buffer Register (MBR)
  - berisi sebuah word data yang akan dituliskan kedalam memori atau word yang terakhir dibaca

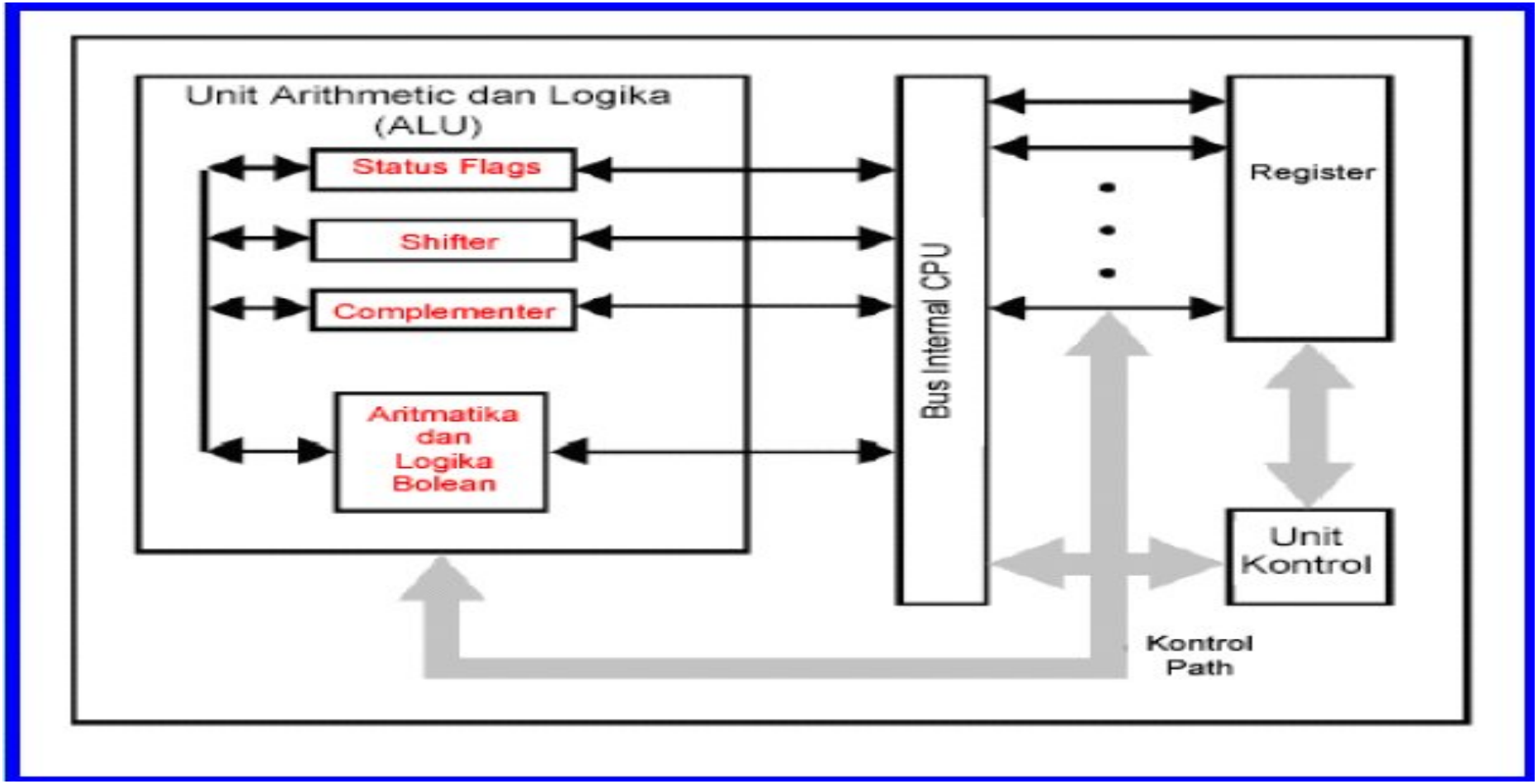
CPU Interconnections

- 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 *main* memori, piranti masukan/keluaran (*I/O*)

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)

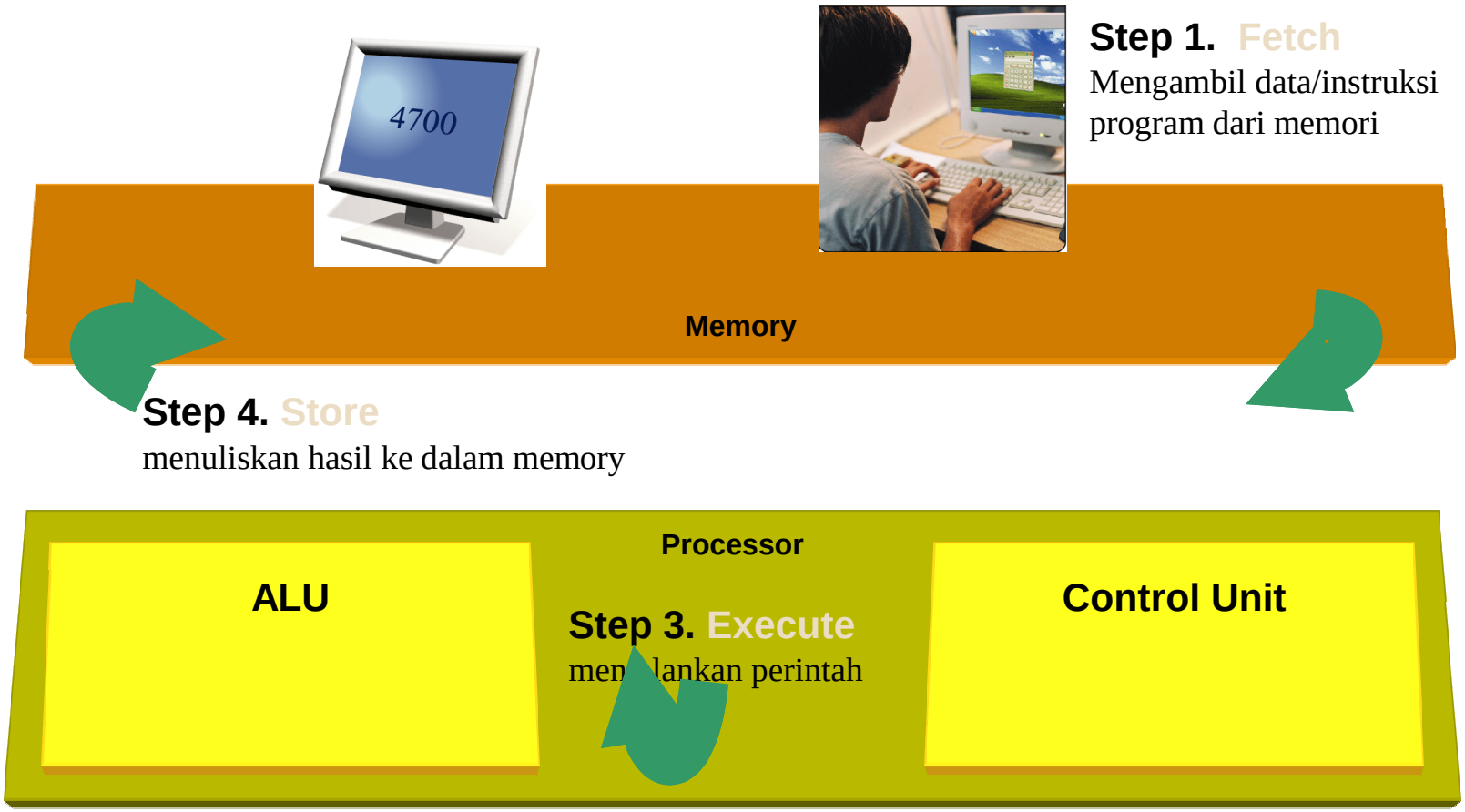
Aksi CPU

- *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.

Prosesor

Siklus mesin (*machine cycle*)

- **Dalam satu siklus mesin terdapat empat operasi dalam CPU**



Siklus Instruksi

- Ada beberapa sub-siklus
- Apa saja??

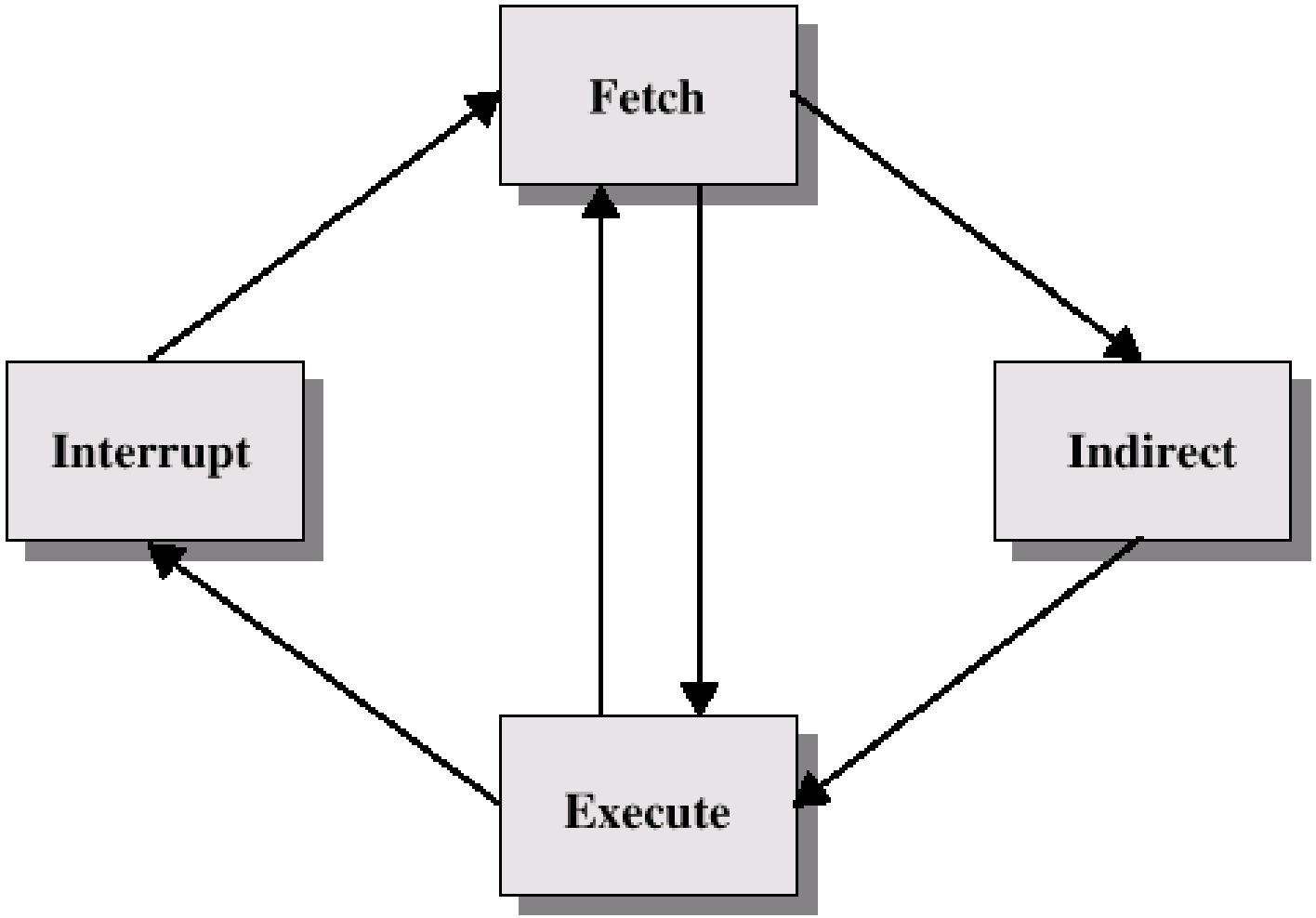
Sub Siklus Instruksi

- Fetch adalah siklus pengambilan data ke memori atau register
- Execute: menginterpretasikan opcode dan melakukan operasi yang diindikasikan
- Interupt: apabila interupt diaktifkan dan interupt telah terjadi, simpan status proses saat itu dan layani interupsi

Siklus tidak langsung

- Eksekusi sebuah instruksi melibatkan sebuah operand atau lebih didalam memori , yang masing-masing operand memerlukan akses memori
- Pengambilan alamat-alamat tak langsung dianggap sebagai sebuah subsiklus instruksi atau lebih

Siklus instruksi



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

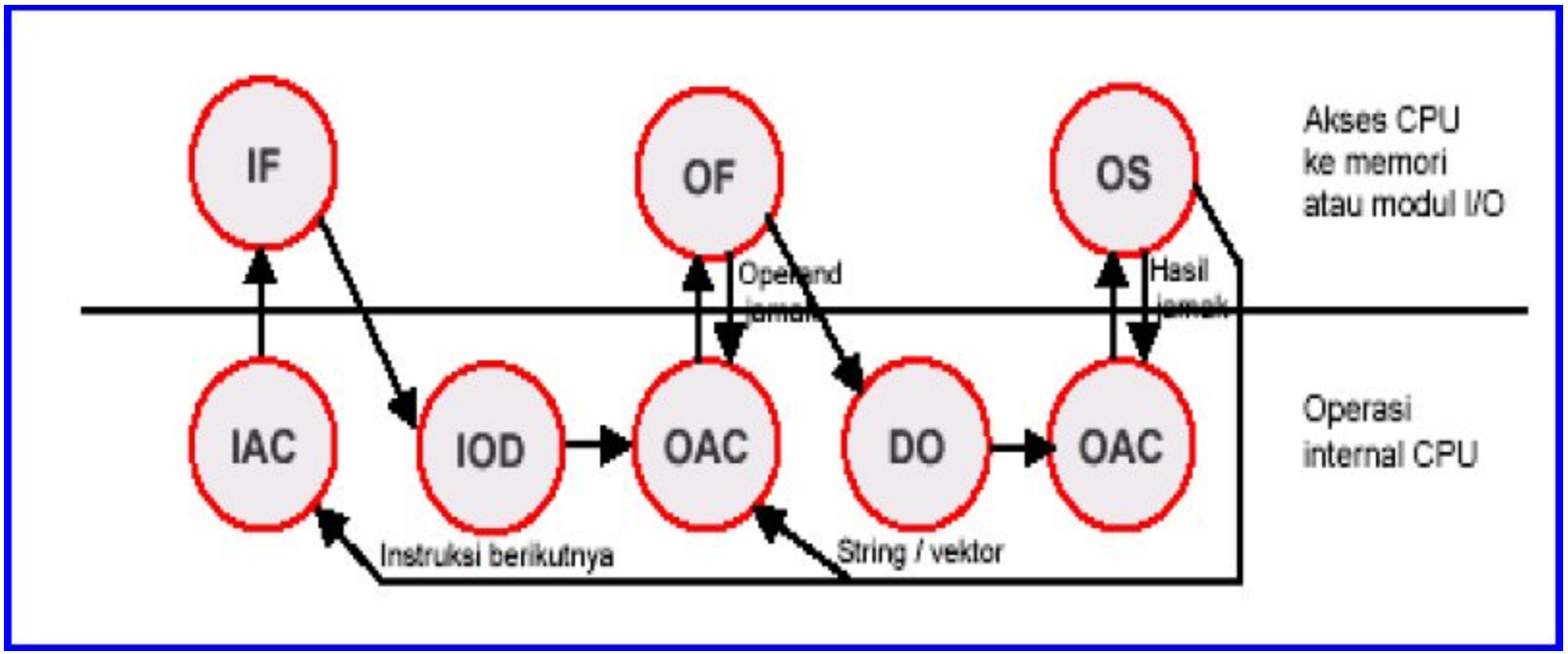
Siklus Fetch - Eksekusi

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

Siklus Eksekusi

- *Instruction Addess 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 pengambil 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

Diagram siklus instruksi



Fungsi Interupsi

- 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 menejemen 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.
- Dapat sebagai sinkronisasi kerja antar modul

Kelas sinyal interupsi

- *Program*, yaitu interupsi yang dibangkitkan dengan beberapa kondisi yang terjadi pada hasil eksekusi program. Contohnya: arimatika overflow, pembagian nol, oparasi 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.

Proses Interupsi

- 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

Proses Interupsi

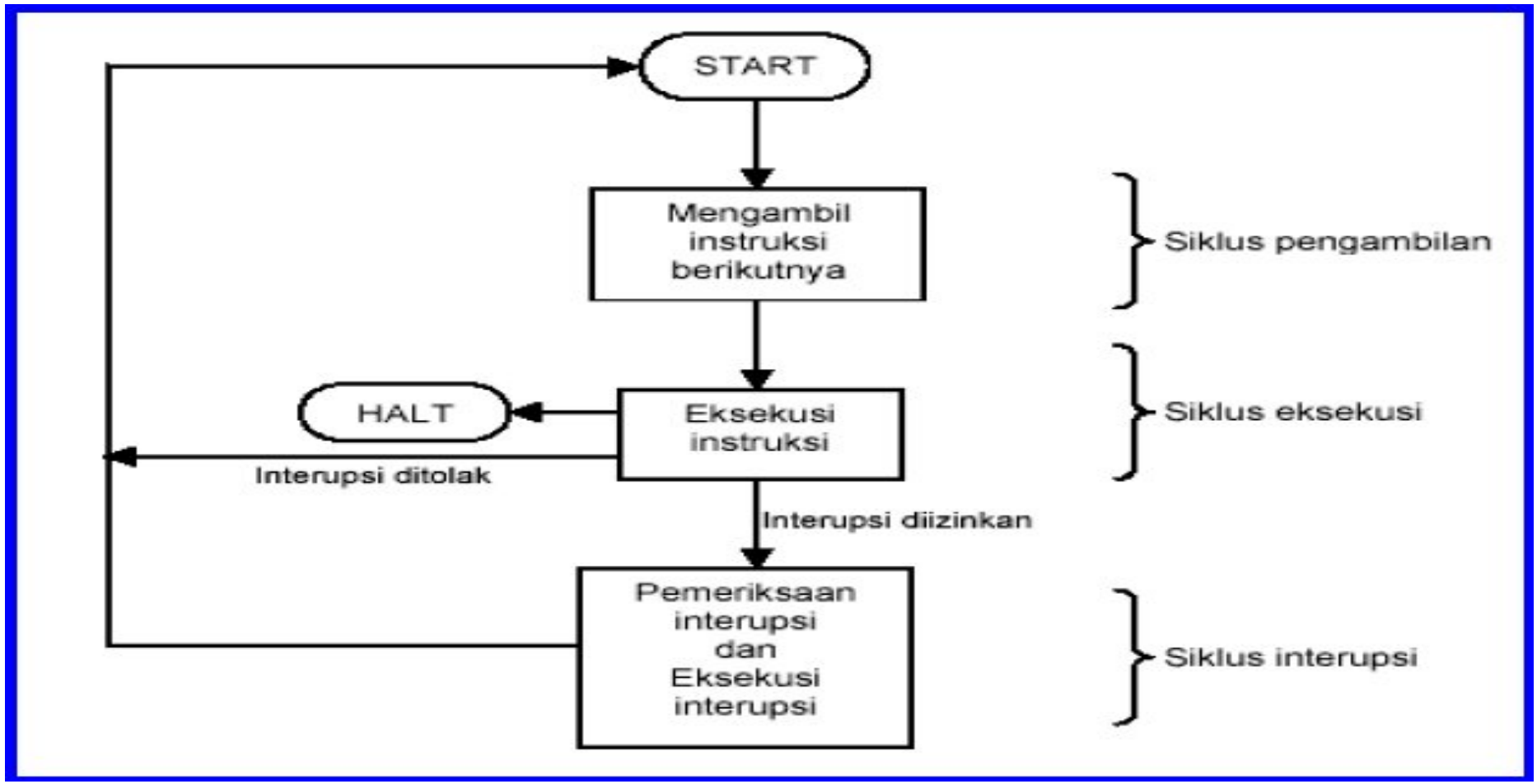
- Kemudian prosesor akan menghentikan eksekusi yang dijalankannya untuk menghandel routine interupsi.
- Setelah program interupsi selesai maka prosesor akan melanjutkan eksekusi programnya kembali.
- Saat sinyal interupsi diterima prosesor ada dua kemungkinan tindakan, yaitu interupsi diterima/ditangguhkan dan interupsi ditolak

Interupsi Ditangguhkan

Apa yang dilakukan Prosessor ?

- Prosesor menangguhkan eksekusi program yang dijalankan dan menyimpan konteksnya. Tindakan ini adalah menyimpan alamat instruksi berikutnya yang akan dieksekusi dan data lain yang relevan.
- Prosesor menyetel program counter (PC) ke alamat awal routine *interrupt handler*.

Siklus eksekusi oleh prosesor dengan adanya fungsi interupsi



Sistem operasi kompleks

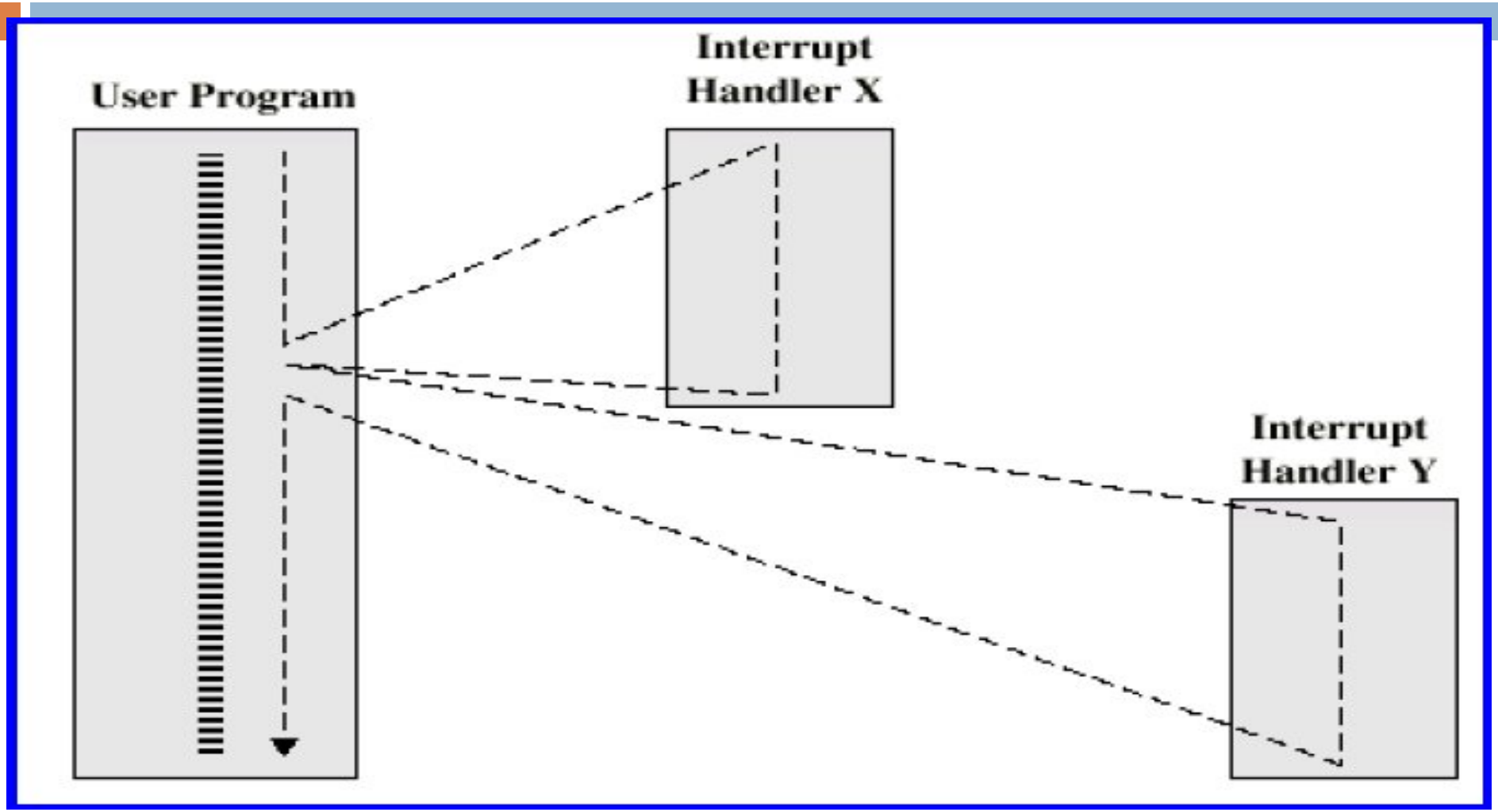
- Interupsi ganda (multiple interrupt).
  - Misalnya suatu komputer akan menerima permintaan interupsi saat proses pencetakan dengan printer selesai, disamping itu dimungkinkan dari saluran komunikasi akan mengirimkan permintaan interupsi setiap kali data tiba.
- Dapat diambil dua buah pendekatan untuk menangani interupsi ganda ini

Pendekatan Interupsi ganda

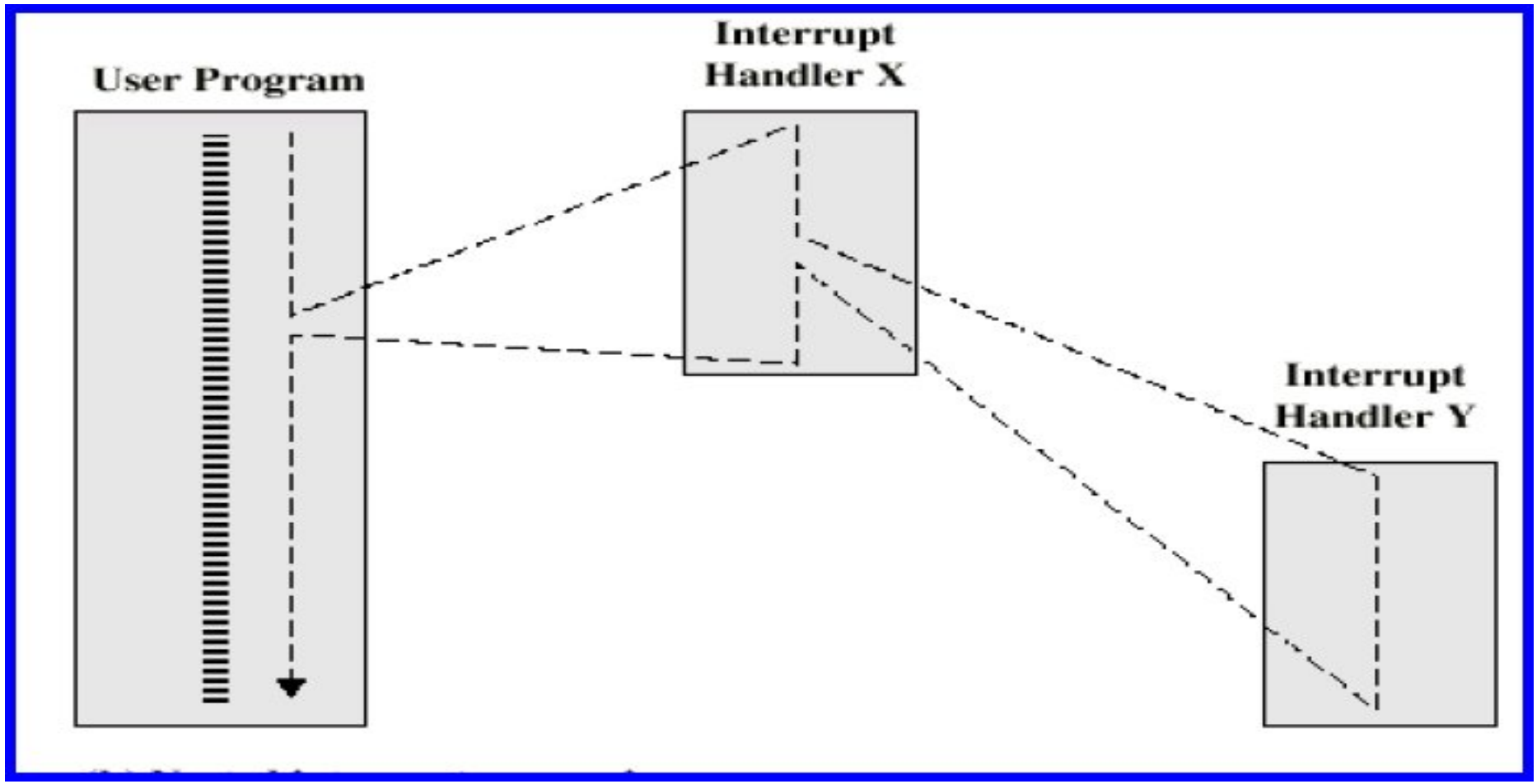
Ada 2 Pendekatan :

- Pendekatan ini disebut pengolahan interupsi berurutan / sekuensial
  - Menolak atau tidak mengizinkan interupsi lain saat suatu interupsi ditangani prosesor.
  - Setelah prosesor selesai menangani suatu interupsi maka interupsi lain baru di tangani.
- Pengolahan interupsi bersarang yaitu mendefinisikan prioritas bagi interupsi
  - Interrupt handler mengizinkan interupsi berprioritas lebih tinggi ditangani terlebih dahulu

Multiple Interrupts - Sequential



Multiple Interrupts - Nested



Contoh Kasus

- Suatu sistem memiliki tiga perangkat I/O: printer, disk, dan saluran komunikasi, masing – masing prioritasnya 2, 4 dan 5. Bagaimana proses interupsinya ?

Contoh Kasus

- 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 pengeksekusian 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