

Register Transfer dan Microoperation

PERTEMUAN 3

Komponen Utama CPU

- ❖ **Arithmetic and Logic Unit (ALU)**, bertugas membentuk fungsi – fungsi pengolahan data komputer
- ❖ **Control Unit**, bertugas mengontrol operasi CPU dan secara keseluruhan mengontrol komputer sehingga terjadi sinkronisasi kerja antar komponen dalam menjalankan fungsi – fungsinya
- ❖ **Registers**, adalah media penyimpan internal CPU yang digunakan saat proses pengolahan data
- ❖ **CPU Interconnections**, adalah sistem koneksi dan bus yang menghubungkan komponen internal CPU, yaitu ALU, unit kontrol dan register – register dan juga dengan bus – bus eksternal CPU yang menghubungkan dengan sistem lainnya

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
- ❖ **Proses Eksekusi Program** adalah dengan mengambil pengolahan instruksi yang terdiri dari dua langkah, yaitu : operasi pembacaan instruksi (fetch) dan operasi pelaksanaan instruksi (execute)

Siklus Fetch–Eksekusi

- ❖ 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)

Aksi-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

Siklus Instruksi

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

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

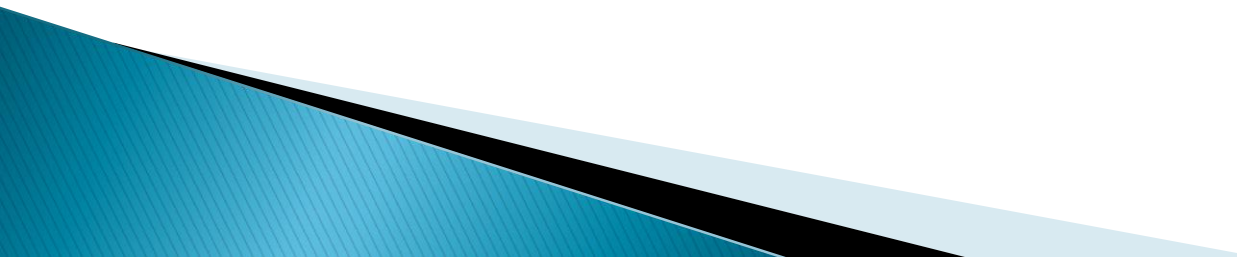
Sinyal Interupsi

- ❖ **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
- ❖ **//0**, 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

Mekanisme Interupsi

- ❖ Saat suatu modul telah selesai menjalankan tugasnya dan siap menerima tugas berikutnya maka modul ini akan mengirimkan permintaan interupsi ke prosesor
- ❖ 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/ditanggihkan dan interupsi ditolak

Iterupsi Ditanggguhkan

- ❖ Prosesor menanggguhkan 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
- 

Iterupsi Ganda

- ❖ Menolak atau tidak mengizinkan iterupsi lain saat suatu iterupsi ditangani prosesor. Kemudian setelah prosesor selesai menangani suatu iterupsi maka iterupsi lain baru di tangani. Pendekatan ini disebut *pengolahan iterupsi berurutan / sekuensial*
- ❖ Prioritas bagi iterupsi dan *interrupt handler* mengizinkan iterupsi berprioritas lebih tinggi ditangani terlebih dahulu. Pendekatan ini disebut *pengolahan iterupsi bersarang*

Interupsi Bersarang

- ❖ Sistem memiliki tiga perangkat I/O: printer, disk, dan saluran komunikasi
- ❖ 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