

ORGANISASI DAN ARSITEKTUR KOMPUTER (SESI 01)

FUNGSI DAN INTERKONEKSI KOMPUTER SECARA UMUM

(Sumber: Computer Organization & Architecture 10th Chapter 3 – William Stalling)



**Program Studi Teknik Informatika
Fakultas Ilmu Komputer
Universitas Esa Unggul
Jakarta
2018**

1. Latar Belakang

Komputer berkembang dengan pesat pada era digital seperti sekarang ini. Spesifikasi komputer juga semakin meningkat seiring dengan semakin banyaknya kebutuhan manusia akan kapasitas, kualitas dan kecepatan komputer. Komputer sudah menjadi aset wajib yang harus dimiliki dalam sebuah organisasi atau perusahaan, untuk dapat menyelesaikan pekerjaan dan permasalahan yang dialami. Komputer memiliki kemampuan yang lebih konstan dan stabil dalam memproses atau menghitung data, dibandingkan dengan manusia yang mengerjakan.

Contohnya dalam penggajian karyawan. Pada jaman dahulu sebelum komputer aktif dioperasikan pada perusahaan-perusahaan, penggajian karyawan dihitung secara manual. Misalnya ada 100 karyawan dalam perusahaan tersebut. Maka tim finansial harus menghitung satu per satu sebanyak 100 kali. Hal ini sangat tidak efektif dan sangat memakan waktu. Dengan adanya komputer, masalah ini dapat diatasi dengan mudah. Hanya cukup dengan memasukkan ID karyawan dan gaji pokok, pajak yang harus dibayarkan oleh karyawan tersebut dapat langsung dikalkulasi oleh aplikasi komputer dan langsung menampilkan jumlah gaji yang harus dibayarkan perusahaan kepada karyawan.

Komputer mempermudah kinerja manusia. Namun pengetahuan manusia terhadap komputer saat ini sebagian besar hanya sebatas pemakaiannya saja. Struktur organisasi dan arsitektur komputer serta interkoneksinya kurang dipahami. Sehingga apabila terjadi sedikit masalah dengan komputer, manusia cenderung untuk berhenti menggunakan komputer dan beralih ke cara manual kembali, tanpa berusaha untuk memperbaiki komputernya karena kurangnya pengetahuan tentang komputer tersebut.

Berdasarkan uraian di atas, penulis membuat makalah berjudul **“Fungsi dan Interkoneksi Komputer Secara Umum”** yang bersumber dari buku *Computer Organization and Architecture 10th Edition (William Stallings)*. Makalah ini diharapkan dapat menambah wawasan para pembaca terkait dengan fungsi komputer dan interkoneksi komputer secara umum.

2. Tujuan

Tujuan dari makalah ini adalah sebagai berikut:

1. Pembaca dapat menambah wawasan mengenai komponen dan cara kerja komputer.
2. Pembaca dapat mengetahui fungsi-fungsi komputer berdasarkan komponennya.
3. Pembaca dapat mengetahui cara kerja interkoneksi komputer.

3. Manfaat

Adapun manfaat penulisan makalah ini adalah:

Pembaca dapat meningkatkan wawasan pengetahuan mengenai fungsi dan interkoneksi komputer. Sehingga pembaca dapat mengetahui sumber masalah pada

fungsi dan interkoneksi komputernya dan lebih lanjut dapat menyelesaikan masalah tersebut.

4. Pembahasan

4.1. Komponen CPU

CPU adalah singkatan dari *Central Processing Unit*. CPU merupakan komponen terpenting dari sebuah sistem komputer. CPU merupakan tempat untuk melakukan pengolahan data berdasarkan instruksi yang diberikan. CPU sendiri terdiri dari beberapa komponen utama yaitu:

1. *Arithmetic and Logic Unit (ALU)*

ALU adalah salah satu komponen CPU yang berfungsi untuk melakukan operasi hitungan aritmatika dan logika. Contoh operasi aritmatika adalah penjumlahan, pengurangan, perkalian dan pembagian. Sedangkan contoh logika adalah AND dan OR. Dalam melakukan operasi aritmatika, ALU menggunakan operasi dasar penjumlahan. Sehingga pengurangan, perkalian dan pembagian dilakukan juga dengan dasar penjumlahan.

2. *Control Unit*

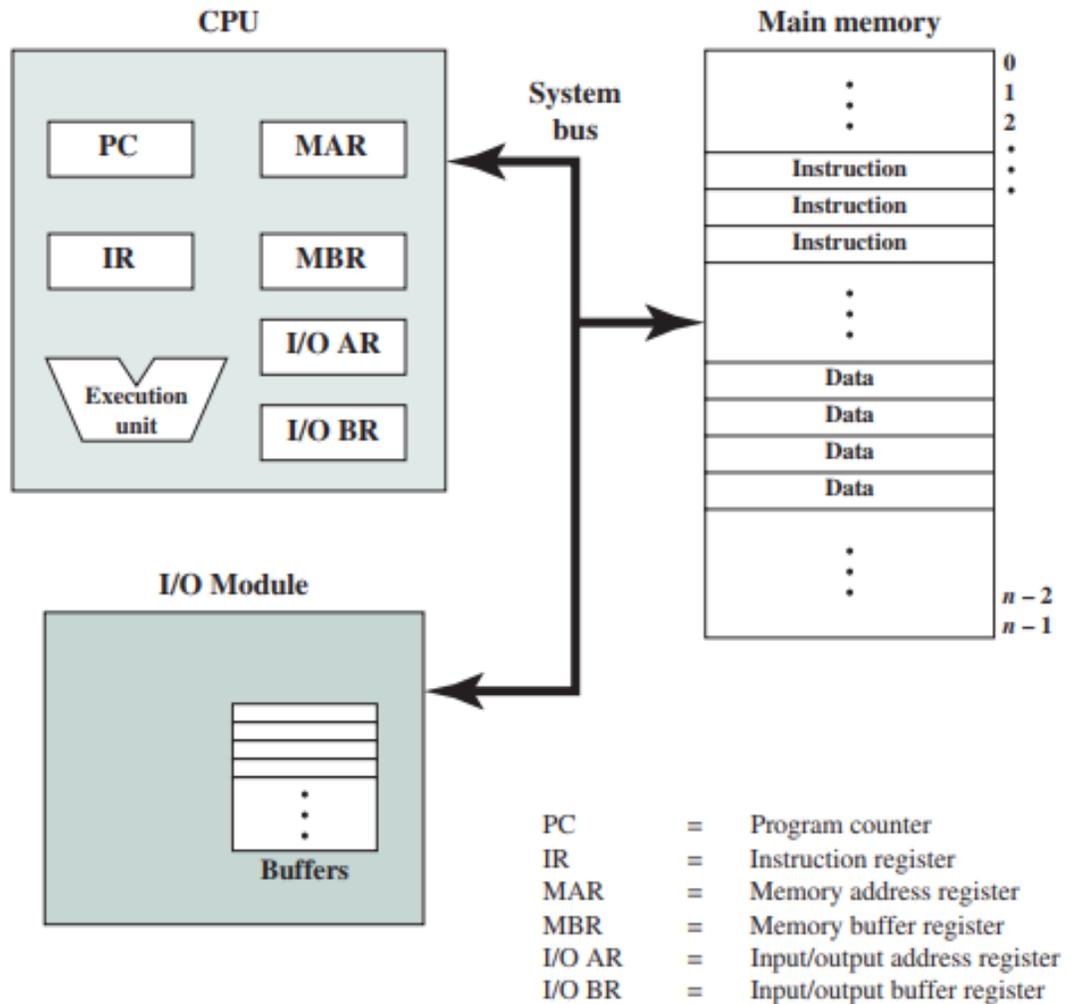
Merupakan salah satu bagian dari CPU yang bertugas untuk mengendalikan kontrol terhadap operasi yang dilaksanakan pada bagian ALU. *Control Unit* mengatur dan mengendalikan I/O, mengambil instruksi dari memori utama, mengirim instruksi ke ALU bila ada perhitungan aritmatika atau perbandingan logika dan menyimpan hasil proses ke memori utama.

3. *Register*

Merupakan alat penyimpanan kecil yang mempunyai kecepatan akses cukup tinggi, yang digunakan untuk menyimpan data dan instruksi yang sedang diproses, sementara data dan instruksi lainnya yang menunggu giliran untuk diproses masih disimpan di dalam memori utama. Setiap *register* dapat menyimpan satu bilangan hingga mencapai jumlah maksimum tertentu tergantung pada ukurannya. *Register* dapat dibaca dan ditulis dengan kecepatan tinggi karena berada pada CPU.

4. *CPU Interconnections*

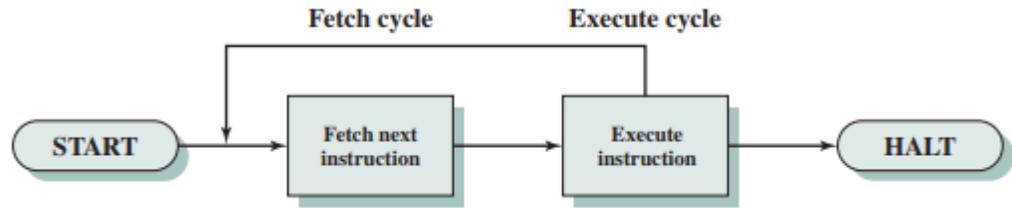
Merupakan sistem koneksi pada CPU yang menghubungkan komponen internal CPU lainnya, yaitu ALU, *Control Unit* dan *Register*. Selain itu *CPU Interconnections* juga menghubungkan CPU dengan sistem lainnya seperti memori utama dan *device I/O*.



Gambar 1. Komponen CPU dan interkoneksinya dengan memori utama dan device I/O(top level view)

4.2.Fungsi Komputer

Fungsi dasar dari CPU adalah eksekusi program. Program adalah set instruksi yang tersimpan dalam memori. CPU akan menjalankan program-program yang disimpan dalam memori utama dengan cara mengambil instruksi-instruksi, menguji instruksi tersebut dan mengeksekusinya satu per satu sesuai dengan urutan perintahnya. Operasi tersebut membutuhkan *instruction cycle* atau siklus instruksi. Ada 2 *basicinstruction cycle* dalam CPU yaitu *fetch cycle* dan *execute cycle*(Stallings, 2016). Selain itu komputer juga dapat menjalankan operasi aritmatika dan memiliki konsep *interrupts*.



Gambar 2. *Basic Instruction Cycle*

4.2.1. *Basic Instruction Cycle*

Fetch cycle terdiri dari beberapa proses berikut ini:

1. CPU awalnya akan membaca instruksi dari memori.
2. *Register* dalam CPU berfungsi mengawasi dan menghitung instruksi selanjutnya yang disebut *Program Counter (PC)*.
3. PC akan menambah satu hitungannya setiap kali CPU membaca instruksi.
4. Instruksi-instruksi yang dibaca akan dimasukkan ke dalam *Instruction Register (IR)*.
5. Instruksi-instruksi ini dalam bentuk kode biner yang dapat diinterpretasikan oleh CPU kemudian dilakukan *action* yang diperlukan sebagaimana penjelasan berikutnya.

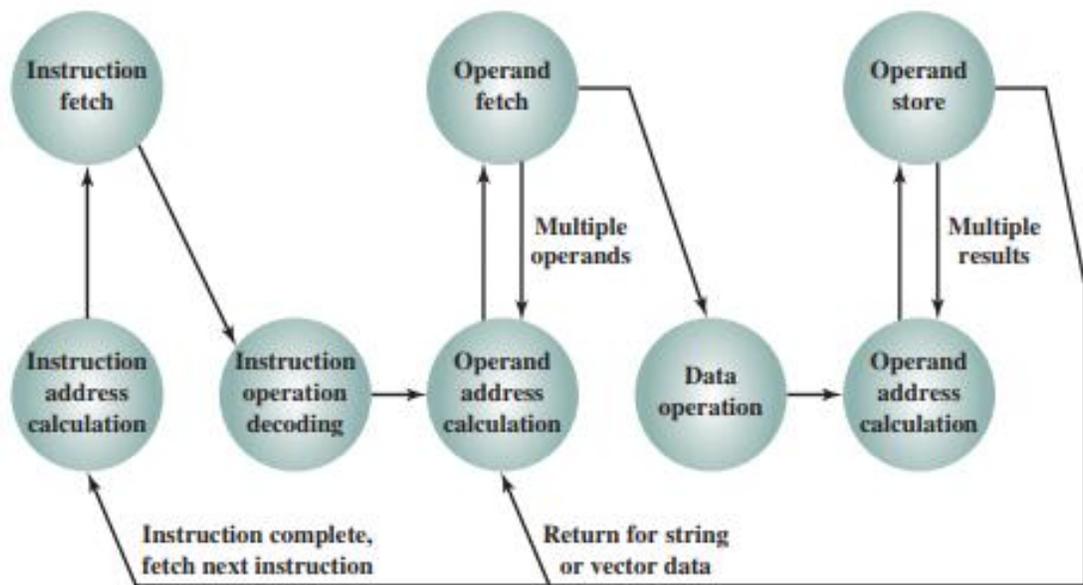
Action dari CPU tersebut dibagi menjadi 4 bagian yaitu:

1. CPU – Memori
Adalah perpindahan data dari CPU ke memori dan sebaliknya.
2. CPU – I/O
Adalah perpindahan data dari CPU ke I/O dan sebaliknya.
3. Pengolahan Data
Bagian dimana CPU melakukan sejumlah operasi aritmatika dan logika terhadap data.
4. Kontrol
Merupakan instruksi untuk melakukan pengontrolan fungsi atau kerja. Misalnya melakukan pemindahan alamat atau pengubahan urutan eksekusi *action*.

Execute cycle merupakan instruksi yang terdiri atas beberapa proses sebagai berikut:

1. *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.
2. *Instruction Fetch (IF)*, yaitu membaca atau mengambil instruksi dari lokasi memorinya ke CPU.

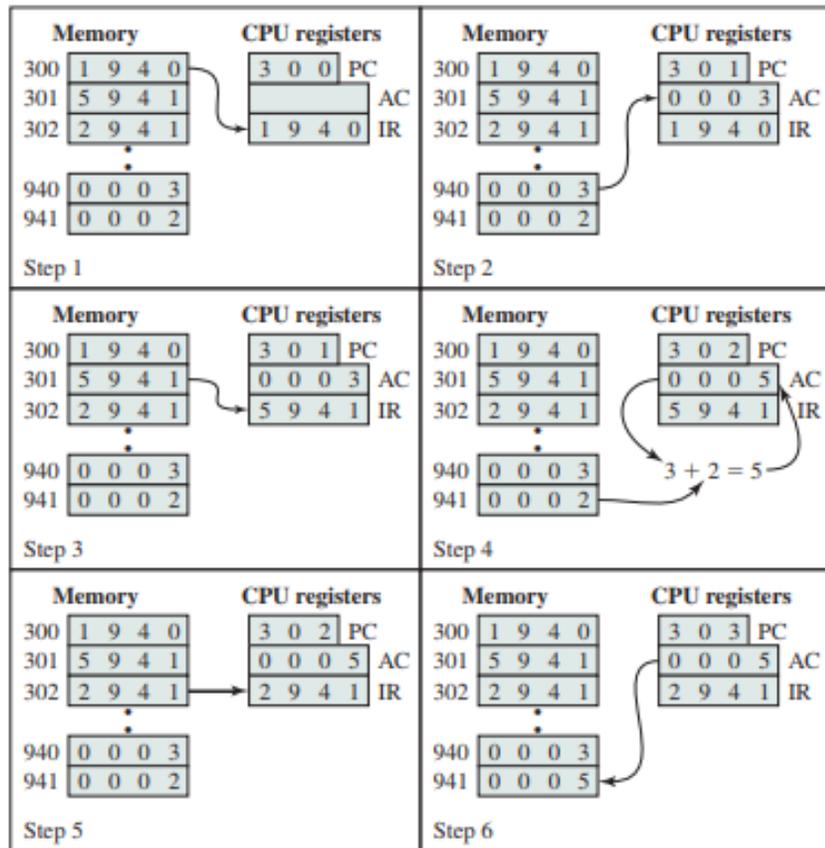
3. *Instruction Operation Decoding (IOD)*, yaitu menganalisa instruksi untuk menentukan jenis operasi yang akan dibentuk dan operand yang akan digunakan.
4. *Operand Address Calculation (OAC)*, yaitu menentukan alamat operand, hal ini dilakukan apabila melibatkan referensi operand pada memori.
5. *Operand Fetch (OF)*, adalah mengambil operand dari memori atau dari modul I/O.
6. *Data Operation (DO)*, yaitu membentuk operasi yang diperintahkan dalam instruksi.
7. *Operand store (OS)*, yaitu menyimpan hasil eksekusi ke dalam memori.



Gambar 3. Diagram State Instruction Cycle

4.2.2. Operasi Aritmatika

Operasi aritmatika pada komputer menggunakan operasi dasar penjumlahan dengan kode bilangan biner. Pengurangan, pembagian dan perkalian semua dilakukan dengan dasar penjumlahan. Contohnya untuk pengurangan. Maka bilangan yang menjadi pengurang akan diubah menjadi bilangan negatif terlebih dahulu melalui kode bilangan biner. Baru setelah itu dijumlahkan dengan bilangan lainnya.



Gambar 4. Operasi aritmatika pada CPU

Gambar di atas merupakan contoh eksekusi program parsial yang menunjukkan bagian yang berhubungan dari register memori dan prosesor. Dapat dilihat bahwa pada alamat 940 dan 941 merupakan dua bilangan yang dijumlahkan yaitu 3 (940) dan 2 (941). Maka untuk langkah-langkah *fetch cycle* dan *execute cycle* diperlukan:

1. PC dengan alamat 300 sebagai alamat instruksi pertama yang berisi 1940 dalam heksadesimal. Instruksi ini dimasukkan ke dalam IR. Sehingga PC bertambah menjadi 301 (dapat dilihat pada kotak paling atas sebelah kiri).
2. 4 bit pertamapada IR menunjukkan bahwa AC harus dimuat. 3 digit heksadesimal menentukan alamat 940 dari mana data akan dimuat. Maka dimasukkan 0003 ke dalam AC (kotak paling atas sebelah kanan).
3. Lihat ke PC dengan alamat 301 yang berisi instruksi 5941. Kemudian dimasukkan ke IR sehingga PC bertambah menjadi 302 (kotak tengah sebelah kiri).

4. Menambahkan isi lokasi 941 (2) dengan AC (3), maka pada AC sudah terisi 0005 hasil dari 2 + 3. Hasilnya disimpan di AC (kotak tengah sebelah kanan).
5. Instruksi berikutnya yaitu 2941 dimasukkan ke IR, sehingga PC bertambah menjadi 302 (kotak paling bawah sebelah kiri).
6. Isi AC disimpan di lokasi 941. PC bertambah menjadi 303 (kotak paling bawah sebelah kanan).

Pada contoh di atas, dibutuhkan masing-masing tiga *fetch cycle* dan *execute cycle* untuk menambahkan isi pada alamat 940 dan alamat 941. Maka dapat dilihat dengan semakin besarnya memori pada CPU akan semakin mempercepat proses operasi aritmatika yang banyak dan kompleks.

4.2.3. *Interrupts*

4.2.3.1. Pengertian dan Pembagian Kelas *Interrupts*

Interrupts atau interupsi adalah suatu mekanisme dimana ada komponen proses dari sistem komputer dihentikan karena terjadi kondisi yang lebih atau sangat penting. Interupsi merupakan mekanisme penghentian atau pengalihan pengolahan instruksi dalam CPU kepada interupsi rutin. Hampir semua modul memiliki mekanisme yang dapat menginterupsi kerja CPU.

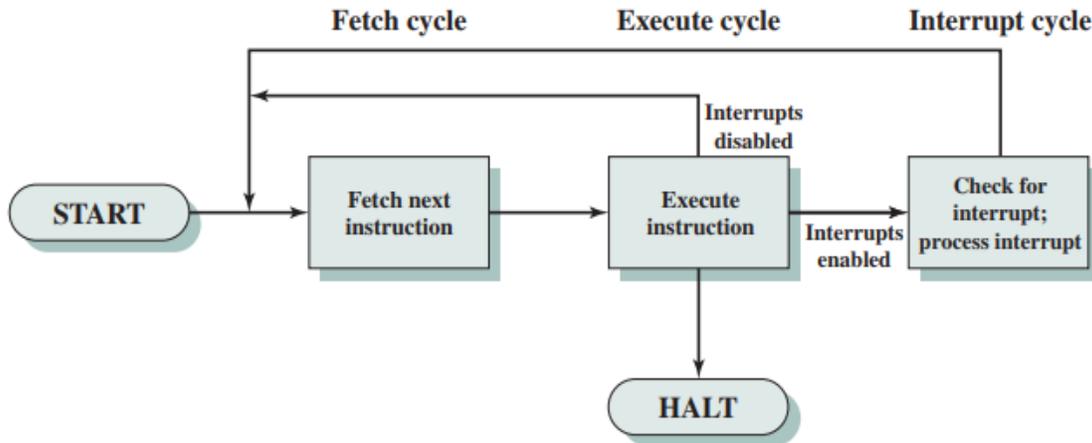
Berikut adalah pembagian kelas dari *interrupts*:

1. Program
Interupsi yang dijalankan dengan beberapa kondisi yang terjadi pada hasil eksekusi program. Contohnya: pembagian nol dan operasi ilegal.
2. *Timer*
Interupsi yang dijalankan karena *schedule* internal prosesor. Sinyal ini memungkinkan sistem operasi menjalankan fungsi tertentu secara reguler.
3. I/O
Merupakan sinyal interupsi yang dibangkitkan oleh modul I/O sehubungan dengan pemberitahuan kondisi error dan penyelesaian suatu operasi.
4. *Hardware Failure*
Interupsi yang dijalankan akibat dari kegagalan atau kesalahan pada perangkat keras memori.

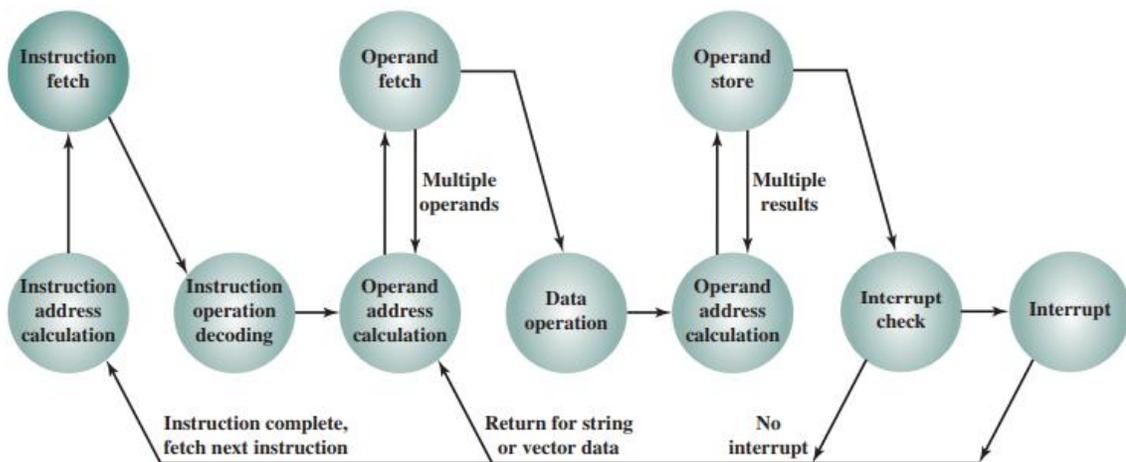
4.2.3.2. Proses *Interrupts*

Proses interupsi memungkinkan prosesor dapat digunakan untuk mengeksekusi intruksi lainnya. Saat suatu modul telah selesai menjalankan tugasnya dan siap menerima tugas berikutnya, maka modul ini akan mengirimkan permintaan interupsi ke prosesor. Kemudian proses akan menghentikan eksekusi yang dijalankannya untuk menjalankan interupsi rutin. Setelah interupsi selesai, prosesor melanjutkan eksekusi programnya kembali. Dan saat sinyal interupsi

diterima akan ada dua kemungkinan *action*, yaitu interupsi diterima atau ditanggihkan, dan interupsi ditolak.



Gambar 5. *Interrupt cycle*



Gambar 6. Diagram *State Instruction Cycle* dengan *Interrupts*

Jika suatu interupsi ditanggihkan maka prosesor akan menunda eksekusi program yang dijalankan dan menyimpan konteksnya. Tindakan ini adalah menyimpan alamat instruksi berikutnya yang akan dieksekusi dan data lain yang relevan. Prosesor mengatur agar PC kembali ke alamat awal penanganan *scheduled interrupts*.

4.2.3.3. Jenis-Jenis *Interrupts*

Secara garis besar *interrupts* dibagi menjadi dua yaitu:

1. *Single Interrupts*

Merupakan interupsi yang hanya menangani penghentian satu eksekusi program saja dan tidak bisa lebih dari satu eksekusi program.

2. Multiple Interrupts

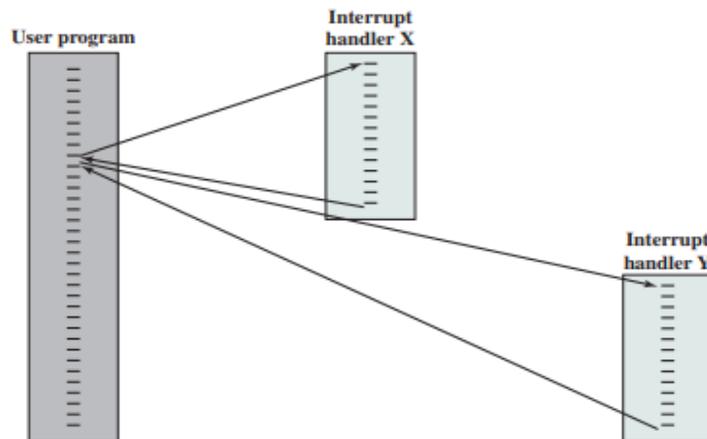
Merupakan interupsi yang menangani penghentian lebih dari satu eksekusi program. *Multiple interrupts* dapat dibagi menjadi dua yaitu:

a) *Sequential Interrupts*

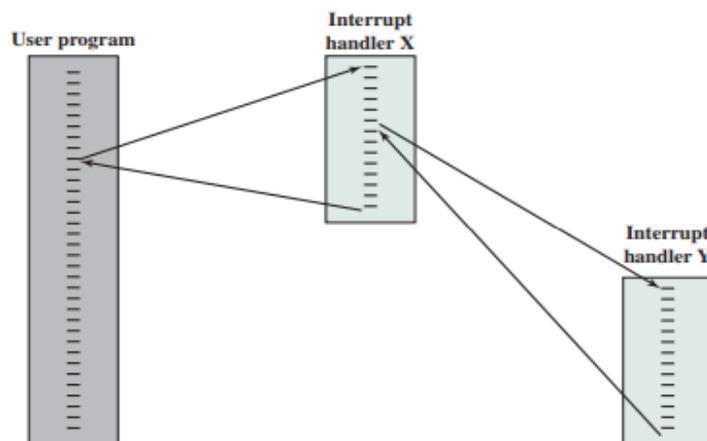
Pada proses interupsi sekuensial ini, CPU tidak mengizinkan adanya interupsi lain saat suatu interupsi sedang ditangani proses. Setelah proses selesai menangani suatu interupsi, maka interupsi lain baru dapat dijalankan.

b) *Nested Interrupts*

Pada proses interupsi *nested* ini, CPU mendefinisikan prioritas bagi interupsi. Interupsi dengan prioritas yang lebih tinggi akan ditangani terlebih dahulu. Sedangkan interupsi dengan prioritas rendah akan ditangani belakangan.



(a) Sequential interrupt processing



(b) Nested interrupt processing

Gambar 7. *Multiple Interrupts*

4.3. Interkoneksi Komputer

4.3.1. Struktur Interkoneksi Komputer

Komputer terdiri dari sekumpulan komponen atau modul dari tiga tipe dasar (prosesor, memori, I / O) yang berkomunikasi satu sama lain. Akibatnya, komputer adalah jaringan modul dasar. Dengan demikian, harus ada jalur untuk menghubungkan modul. Kumpulan jalur yang menghubungkan berbagai modul disebut interkoneksi struktur. Rancangan struktur ini akan bergantung pada pertukaran itu harus dibuat di antara modul.

Gambar berikut menyarankan jenis pertukaran yang dibutuhkan dengan menunjukkan bentuk utama dari input dan output untuk setiap tipe modul:

1. **Memori:** Biasanya, modul memori akan terdiri dari kata-kata N dengan panjang yang sama. Setiap kata diberi alamat numerik yang unik ($0, 1, \dots, N-1$). Sepatah kata dari data dapat dibaca dari atau ditulis ke dalam memori. Sifat operasi

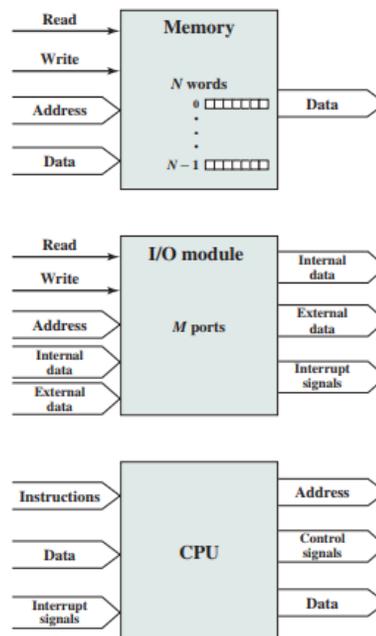


Figure 3.15 Computer Modules

²The wide arrows represent multiple signal lines carrying multiple bits of information in parallel. Each narrow arrow represents a single signal line.

Gambar 8. Memori, Modul I/O dan CPU

2. **Modul I / O:** Dari sudut pandang internal (ke sistem komputer), I / O adalah secara fungsional mirip dengan memori. Ada dua operasi; Baca dan tulis. Lebih lanjut, modul I / O dapat mengontrol lebih dari satu perangkat eksternal. Kami bisa merujuk ke masing-masing antarmuka ke perangkat eksternal sebagai port dan

memberikan masing-masing yang unikalamat (misalnya, 0, 1, c, M-1). Selain itu, ada jalur data eksternal untuk input dan output data dengan perangkat eksternal. Akhirnya, modul I / O mungkin dapat mengirim sinyal interrupt ke prosesor.

3. **Prosesor:** Prosesor membaca instruksi dan data, menulis data setelahnya memproses, dan menggunakan sinyal kontrol untuk mengontrol keseluruhan operasi sistem. Ia juga menerima sinyal interupsi. Daftar sebelumnya menentukan data yang akan dipertukarkan. Interkoneksi struktur harus mendukung jenis transfer berikut:
 - a. Memori ke prosesor: Prosesor membaca instruksi atau unit data dari ingatan.
 - b. Prosesor ke memori: Prosesor menulis satu unit data ke memori. I / O ke prosesor: Prosesor membaca data dari perangkat I / O melalui I / O modul
 - c. Prosesor ke I / O: Prosesor mengirim data ke perangkat I / O
 - d. I / O ke atau dari memori: Untuk dua kasus ini, modul I / O dibolehkan bertukar data langsung dengan memori, tanpa melalui prosesor, menggunakan akses memori langsung. Selama bertahun-tahun, sejumlah struktur interkoneksi telah dicoba. Sejauh ini yang paling umum adalah (1) bus dan berbagai struktur multi-bus, dan (2) point-to-point struktur interkoneksi dengan transfer data terkemas. Kami mengabdikan sisa bab ini untuk diskusi tentang struktur ini.

6.3.2 Interkoneksi *Bus*

Selama beberapa dekade. Untuk komputer tujuan umum, secara bertahap telah diberikan berbagai cara struktur interkoneksi point-to-point, yang sekarang mendominasi sistem komputer Desain. Namun, struktur bus masih umum digunakan untuk sistem embedded, khususnya mikrokontroler. Pada bagian ini, kami memberikan gambaran singkat tentang struktur bus. Bus adalah jalur komunikasi yang menghubungkan dua perangkat atau lebih. Kunci karakteristik bus adalah bahwa itu adalah medium transmisi bersama. Beberapa perangkat terhubung ke bus, dan sinyal yang dikirimkan oleh perangkat apa pun tersedia untuk penerimaan oleh semua perangkat lain yang terhubung ke bus. Jika dua perangkat mengirimkan selama yang sama periode waktu, sinyal mereka akan tumpang tindih dan menjadi kacau. Jadi, hanya satu perangkat di waktu dapat berhasil mengirimkan. Biasanya, bus terdiri dari beberapa jalur komunikasi, atau jalur. Setiap garis mampu

mentransmisikan sinyal yang mewakili biner 1 dan biner 0. Seiring waktu, urutan digit biner dapat ditransmisikan melalui satu baris. Diambil bersama-sama, beberapa baris bus dapat digunakan untuk mengirimkan digit biner secara bersamaan (secara paralel). Sebagai contoh, unit data 8-bit dapat ditransmisikan lebih dari delapan jalur bus. Sistem komputer mengandung sejumlah bus berbeda yang menyediakan jalur antara komponen pada berbagai tingkat hirarki sistem komputer. Sebuah bus itu menghubungkan komponen komputer utama (prosesor, memori, I / O) disebut system bus. Struktur interkoneksi komputer yang paling umum didasarkan pada penggunaan satu atau lebih bus sistem.

Sebuah bus sistem biasanya terdiri dari sekitar lima puluh hingga ratusan orang yang terpisah garis. Setiap baris diberi makna atau fungsi tertentu. Meski ada banyak desain bus yang berbeda, di setiap bus garis dapat diklasifikasikan menjadi tiga fungsional kelompok: data, alamat, dan garis kontrol. Selain itu, mungkin ada jalur distribusi daya yang memasok listrik ke modul-modul yang terpasang. Jalur data menyediakan jalur untuk memindahkan data antar modul sistem. Ingaris, secara kolektif, disebut bus data. Bus data dapat terdiri dari 32, 64, 128, atau bahkan garis yang lebih terpisah, jumlah garis yang disebut sebagai lebar bus data. Karena setiap baris hanya dapat membawa satu bit pada satu waktu, jumlah garis menentukan berapa banyak bit yang dapat ditransfer pada suatu waktu. Lebar data bus adalah faktor kunci dalam menentukan kinerja sistem secara keseluruhan. Misalnya, jika bus data 32 bit lebar dan setiap instruksi 64 bit panjang, maka prosesor harus akses modul memori dua kali selama setiap siklus instruksi. Baris alamat digunakan untuk menunjuk sumber atau tujuan dari data bus. Misalnya, jika prosesor ingin membaca kata (8, 16, atau 32 bit) data dari memori, menempatkan alamat kata yang diinginkan pada baris alamat. Jelas, lebar bus alamat menentukan kapasitas memori maksimum yang mungkin dari sistem. Selanjutnya, garis alamat umumnya juga digunakan untuk alamat I / O port. Biasanya, bit orde tinggi digunakan untuk memilih modul tertentu pada bus, dan bit lower-order memilih lokasi memori atau port I / O dalam modul. Misalnya, pada bus alamat 8-bit, alamat 01111111 dan di bawah ini mungkin referensi lokasi dalam modul memori (modul 0) dengan 128 kata memori, dan alamat 10000000 dan di atas merujuk ke perangkat yang terpasang ke modul I / O (modul 1).

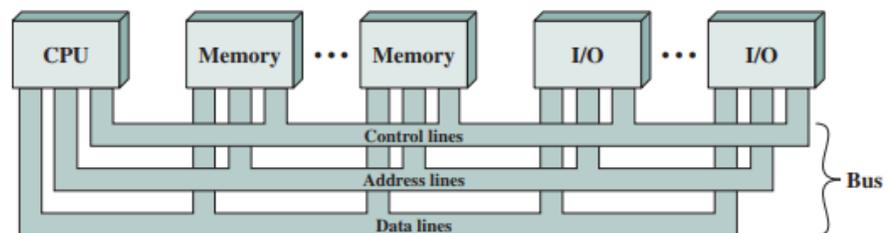


Figure 3.16 Bus Interconnection Scheme

Gambar 9. Skema Interkoneksi Bus

Jalur kontrol digunakan untuk mengontrol akses ke dan penggunaan data dangaris alamat. Karena data dan garis alamat dibagi oleh semua komponen,harus ada cara untuk mengontrol penggunaannya. Sinyal control mengirimkan kedua perintah dan informasi waktu antar modul sistem. Sinyal waktu menunjukkan validitas data dan informasi alamat. Sinyal perintah menentukan operasi yang harus dilakukan dilakukan. Jalur kontrol yang umum termasuk:

- *Memory write*: menyebabkan data pada bus yang akan ditulis ke lokasi yang dialamatkan.
- *Memory read*: menyebabkan data dari lokasi yang dituju untuk ditempatkan di bus.
- *I / O write*: menyebabkan data pada bus menjadi output ke port I / O yang dialamatkan.
- *I / O read*: menyebabkan data dari port I / O yang dialamatkan untuk ditempatkan di bus.
- *Transfer ACK*: menunjukkan bahwa data telah diterima dari atau ditempatkan padabis.
- *Bus request*: menunjukkan bahwa modul perlu menguasai bus.
- *Bus grant*: menunjukkan bahwa modul yang meminta telah diberikan kendali atasbis.
- *Interrupt request*: menunjukkan bahwa interupsi sedang tertunda.
- *Interrupt ACK*: mengakui bahwa interupsi yang tertunda telah diakui.
- *Clock*: digunakan untuk menyinkronkan operasi.
- *Reset*: menginisialisasi semua modul.

Pengoperasian bus adalah sebagai berikut. Jika satu modul ingin mengirim data kelain, itu harus melakukan dua hal: (1) mendapatkan penggunaan bus, dan (2) transfer datamelalui bus. Jika satu modul ingin meminta data dari modul lain, itu harus(1) dapatkan penggunaan bus, dan (2) transfer permintaan ke modul lain di ataskontrol dan garis alamat yang tepat. Kemudian harus menunggu modul kedua itukirim datanya

6.3.3. Interkoneksi *Point-to-Point*

Arsitektur *bus* adalah pendekatan standar untuk interkoneksi antara prosesor dengan memori, I/O dan komponen lainnya selama beberapa dekade. Tetapi saat ini sistem semakin bergantung kepada interkoneksi *point-to-point*. Alasan utamanya adalah kendala listrik yang dihadapi arsitektur *bus*. Pada tingkat data yang semakin tinggi, menjadi semakin sulit untuk melakukan sinkronisasi dan arbitrase secara tepat waktu.

Berikut adalah karakter dari skema interkoneksi *point-to-point*:

1. *Multiple Direct Connections*

Beberapa komponen dalam sistem melakukan koneksi berpasangan langsung ke komponen lain. Ini menghilangkan kebutuhan untuk arbitrase yang ditemukan dalam sistem *shared transmission*.

2. *Layered Protocol Architecture*

Interkoneksi tingkat prosesor ini lebih menggunakan arsitektur protocol yang berlapis dibandingkan dengan penggunaan sinyal sederhana yang ditemukan dalam sistem *shared bus*.

3. *Packetized Data Transfer*

Data tidak dikirim sebagai aliran bit yang mentah dan tidak dapat dikonsumsi. Sebaliknya, data dikirim sebagai urutan paket yang masing-masing mencakup kontrol *header* dan kontrol *error*.

6.4. *PCI Express*

PCI Express secara teknis merupakan kepanjangan dari *Peripheral Component Interconnect Express* namun sering disingkat PCIe atau PCI-E. Ini merupakan tipe standar koneksi untuk perangkat internal di komputer. Umumnya, *PCI Express* mengacu pada slot ekspansi aktual pada *motherboard* yang menerima kartu ekspansi berbasis PCIe dan jenis kartu ekspansi itu sendiri (jadi bukan hanya GPU yang bisa anda pasang).

PCIe ini menggantikan slot AGP dan PCI, dimana keduanya menggantikan tipe koneksi tertua yang sering digunakan dan biasa disebut ISA. Sementara komputer mungkin biasanya berisi campuran berbagai jenis slot ekspansi, *PCI Express* dianggap sebagai interface internal standar. Bahkan, banyak *motherboard* komputer saat ini diproduksi hanya dengan slot *PCI Express* saja.

Nomor setelah x biasanya menunjukkan ukuran fisik kartu PCIe atau slotnya, dimana x16 merupakan yang terbesar dan x1 adalah yang terkecil. Maksud dari ukuran tersebut biasanya mengacu pada jumlah pin, diantaranya :

- *PCI Express x1*: 18 pin (25 mm)
- *PCI Express x4*: 32 pin (39 mm)
- *PCI Express x8*: 49 pin (56 mm)
- *PCI Express x16*: 82 pin (89 mm)

Tidak peduli ukuran slot atau kartu PCIe apa, takik utamanya (*key notch*), ruang kecil di kartu atau slot itu, selalu ada jumlah 11 pin pada bagian pertama. Dengan kata lain, ini adalah panjang Pin yang terus bertambah saat anda berpindah dari PCIe x1. ke PCIe x16. Hal ini memungkinkan beberapa fleksibilitas untuk menggunakan satu kartu dengan slot yang lain. Secara umum, kartu *PCI Express* atau slot yang lebih besar biasanya mendukung kinerja yang lebih baik, dengan asumsi dua kartu atau slot yang anda bandingkan mendukung versi PCIe yang sama.

Setiap nomor setelah PCIe yang Anda temukan pada produk atau motherboard juga biasanya menunjukkan nomor versi terbaru dari spesifikasi *PCI Express* yang didukung. Begini caranya melihat versi *PCI Express* jika dibandingkan dengan performa kecepatan yang didukungnya :

- *PCI Express 1.0*: 2 Gbit / s per lajur (32 Gbit / s pada slot x16)

- *PCI Express 2.0*: 4 Gbit / s per lajur (64 Gbit / s pada slot x16)
- *PCI Express 3.0*: 7.877 Gbit / s per lajur (126,032 Gbit / s pada slot x16)

Semua versi *PCI Express* kompatibel baik itu pada bagian belakang atau depan, yang berarti tidak peduli versi apa kartu PCIe atau *motherboard* Anda dukung, mereka biasanya bisa bekerja sama, setidaknya pada tingkat minimum. *Update* utama pada standar PCIe secara drastis meningkatkan *bandwidth* yang tersedia setiap saat, dan ini akan meningkatkan potensi dari apa yang dapat dilakukan oleh perangkat keras yang terhubung. Perbaikan versi juga biasanya memperbaiki *bug*, menambahkan fitur, dan meningkatkan manajemen daya, namun peningkatan *bandwidth* merupakan perubahan yang terpenting yang dan perlu anda perhatikan dari versi ke versi.

7. Kesimpulan

Komputer adalah suatu perangkat elektronik unik yang dapat menerima *input*, melakukan proses dan menghasilkan *output*. Komputer memiliki arsitektur komponen yang saling bekerjasama untuk membangun sebuah sistem, dimana komponen-komponen tersebut membuat komputer memiliki fungsi untuk menjalankan siklus, menjalankan operasi aritmatika dan menjalankan interupsi. Interkoneksi komputer adalah sistem yang menghubungkan komponen-komponen komputer supaya pemrosesan di dalam komputer tetap berjalan dengan lancar dalam proses *input* dan menghasilkan *output*.

Dalam melaksanakan prosesnya, komputer memiliki beberapa alur kerja yang dimana alur kerja tersebut diatur sendiri oleh sistem komputer urutan prioritasnya. Sehingga proses kerja komputer dapat berjalan dengan tepat, walaupun tidak selamanya lancar. Pengguna komputer diharapkan dapat mengetahui proses kerja komputer ini. Karena walaupun komputer memiliki sistem otomatisasi, mereka tetap membutuhkan proses dan waktu dalam menjalankan sistemnya.

8. DAFTAR PUSTAKA

William Stallings, 2016, *Computer Organization and Architecture Designing for Performance*, 10th Edition, Pearson Education, New Jersey, USA.