

INPUT/OUTPUT

Fungsi : Memindahkan informasi antara CPU atau memori utama dengan dunia luar

I/O terdiri :

- Piranti I/O (*peripheral*)
- Pengendali I/O (*device controller*)
- Perangkat lunak

Proses transfer informasi antara CPU dengan sebuah peripheral :

- Memilih I/O dan mengujinya.
- Menginisialisasi transfer dan mengkoordinasikan pengaturan waktu operasi I/O.
- Mentransfer informasi.
- Menghentikan proses transfer.

Klasifikasi piranti I/O terdiri 3 kelompok:

- Kelompok yang memasukkan informasi (input), contoh : keyboard, ADC, scanner
- Kelompok yang menampilkan informasi (output), contoh : VDU (monitor), printer
- Kelompok yang melayani input dan output, contoh : Floppy disk

Pengaksesan I/O terdiri dari 2 cara :

1. *Memory mapped I/O*

Piranti I/O dihubungkan sebagai lokasi memori virtual dimana port I/O tergantung memori utama.

Karakteristik:

- Port I/O dihubungkan ke bus alamat.
- Piranti input sebagai bagian memori yang memberikan data ke bus data. Piranti output sebagai bagian memori yang memiliki data yang tersimpan di dalamnya.
- Port I/O menempati lokasi tertentu pada ruang alamat dan diakses seolah-olah adalah lokasi memori.

2. *I/O mapped I/O (I/O isolated)*

Piranti I/O dihubungkan sebagai lokasi terpisah dengan lokasi memori, dimana port I/O tidak tergantung pada memori utama.

Karakteristik:

- Port I/O tidak tergantung memori utama.
- Transfer informasi dilakukan di bawah kendali sinyal kontrol yang menggunakan instruksi INPUT dan OUTPUT.
- Operasi I/O tergantung sinyal kendali dari CPU.
- Instruksi I/O mengaktifkan baris kendali read/write pada port I/O, sedangkan instruksi memori akan mengaktifkan baris kendali read/write pada memori.
- Ruang memori dan ruang alamat I/O menyatu, sehingga dapat memiliki alamat yang sama.

Kelebihan dan kekurangan:

- I/O mapped I/O lebih cepat dan efisien, karena lokasi I/O terpisah dengan lokasi memori.
- I/O mapped I/O mempunyai keterbatasan jumlah instruksi yang dapat digunakan untuk operasi I/O.

Operasi I/O terbagi menjadi 3 metode :

1. I/O terprogram

Metode di mana CPU mengendalikan operasi I/O secara keseluruhan dengan menjalankan serangkaian instruksi I/O dengan sebuah program.

Karakteristik:

- Program tersebut digunakan untuk memulai, mengarahkan dan menghentikan operasi-operasi I/O.
- Membutuhkan sejumlah perangkat keras (register) yaitu:
 - Register status, berisi status piranti I/O dan data yang akan dikirimkan.
 - Register buffer, menyimpan data sementara sampai CPU siap menerimanya
 - Pointer buffer, menunjuk ke lokasi memori di mana sebuah karakter harus ditulis atau dan mana karakter tersebut harus dibaca.
 - Counter data, tempat penyimpanan jumlah karakter dan akan berkurang nilainya jika karakter ditransfer.
- Membutuhkan waktu proses yang lama dan tidak efisien dalam pemanfaatan CPU.

2. I/O interupsi

Metode di mana CPU akan bereaksi ketika suatu piranti mengeluarkan permintaan untuk pelayanan.

Karakteristik:

- Lebih efisien dalam pemanfaatan CPU, karena tidak harus menguji status dari piranti.

- Interupsi dapat berasal dari piranti I/O, interupsi perangkat keras misalnya : timer, memori, power supply, dan Interupsi perangkat lunak misalnya : overflow, opcode/data yang ilegal, pembagian dengan nol.

Ada 2 jenis interupsi:

1. Interupsi maskable

Interupsi yang dapat didisable (dimatikan) untuk sementara dengan sebuah instruksi *disable interupsi khusus*.

2. Interupsi nonmaskable

Interupsi yang tidak dapat didisable dengan instruksi perangkat lunak.

Dalam sistem komputer terdapat lebih dari satu piranti yang memerlukan pelayanan interupsi, dapat digunakan metode:

- *Polling/pollled interupt*

Berdasarkan urutan prioritas yang telah ditentukan sebelum piranti memerlukan interupsi.

Misal: piranti A dan B mempunyai urutan prioritas A lebih dulu dari B, maka jika A dan B secara bersamaan memerlukan pelayanan interupsi, maka piranti A akan didahulukan.

- *Vector Interupt*

Peralatan yang berinterupsi diidentifikasi secara langsung dan dihubungkan routine pelayanan vector interupt.

INTR = Sinyal yang dikeluarkan oleh peralatan.

INTA = Sinyal kendali yang digunakan CPU untuk menyiapkan pelayanan interupt

Cara yang biasa digunakan dengan metode daisy chain dan encoder prioritas

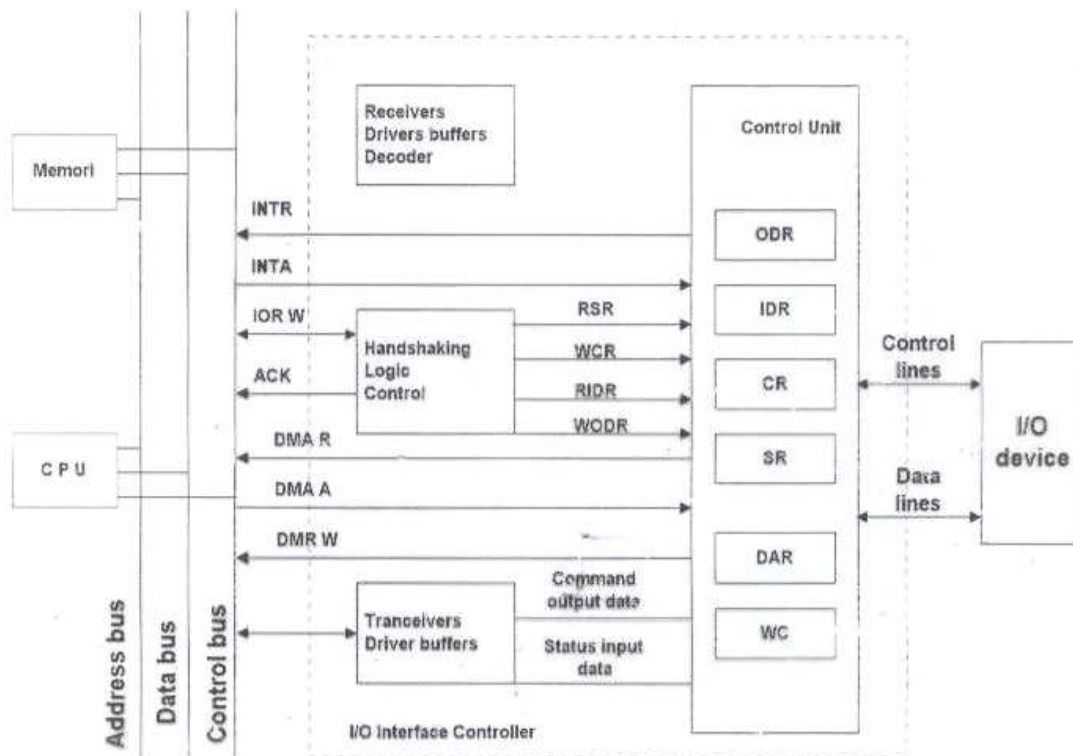
3. Direct Memory Access (DMA)

Metode transfer data secara langsung antara memori dengan piranti tanpa pengawasan dan pengendalian CPU.

- Skema transfer blok DMA dual port
CPU dan DMA controller mengakses memori utama melalui MAR dan MBR dengan menggunakan sebuah memori utama dual port (2 port).
Port I ----> melayani CPU
Port II ----> melayani DMA controller
- Skema transfer blok DMA *cycle stealing* (pencurian siklus)
Hanya memerlukan sebuah memori port tunggal di mana CPU dan piranti I/O beradu cepat pada basis asinkron, prioritas utama akan diberikan pada piranti I/O.

Interfacing

Adalah peralatan yang digunakan untuk menghubungkan suatu piranti dengan CPU melalui bus.



Gambar 1. diagram blok sebuah unit interface I/O secara skematis

Keterangan:

- Register kendali (CR) digunakan untuk mencatat berbagai perintah dan informasi lainnya dalam peripheral.
- Register status (SR) digunakan untuk menyimpan status piranti dan memberitahukan pesan-pesan kesalahan .
- Register data input (IDR) dan register data output (ODR) masing-masing berfungsi sebagai bufer data untuk operasi input dan output.
- Urutan operasi interface:
 - Unit logika *handshaking* memasok unit kendali dengan empat sinyal.
 - Dua sinyal, register kendali penulisan (WCR atau write control register) dan register status

pembacaan (RSR atau *read status register*), masing-masing berhubungan dengan CR dan SR.

- Sedangkan dua sinyal lainnya adalah register pembacaan data input (RIDR atau *read input data register*) dan register penulisan data output (WODR atau *write output data register*), masing-masing mengendalikan IDP dan ODR.

Transfer data

Format transfer

- Paralel : semua bit pada karakter (word dengan panjang tertentu) dikirim secara bersamaan dalam batas waktu yang diberikan.
- Serial : Data dikirim secara berurutan dalam satu saluran.

Transfer data secara paralel lebih cepat daripada secara serial karena saluran transmisinya banyak, kelemahannya kalau terlalu panjang akan terjadi interferensi antar saluran.

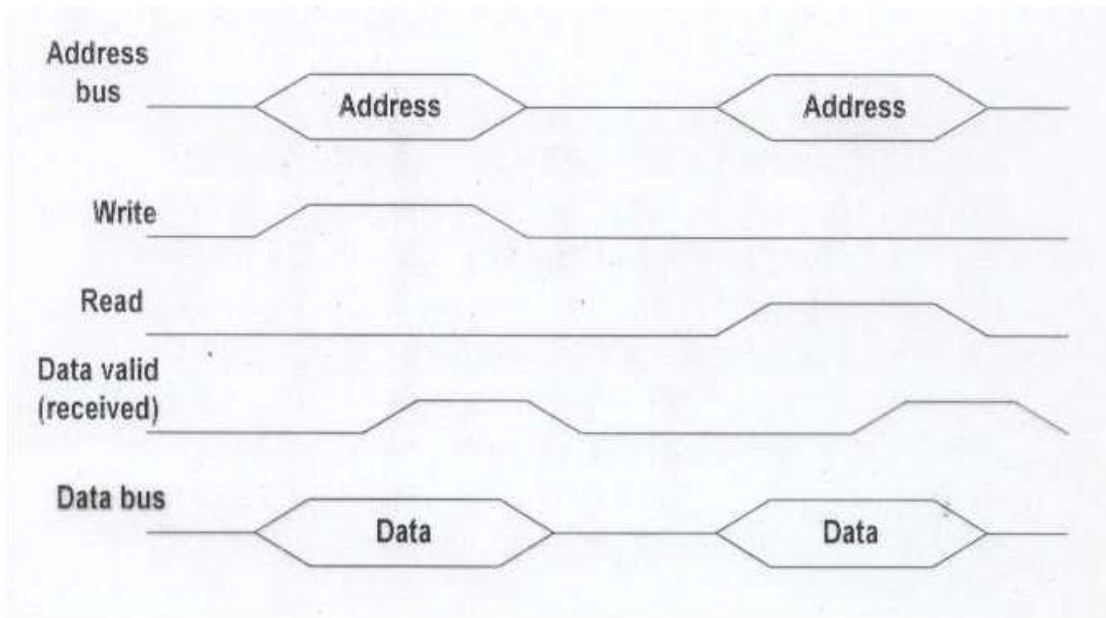
Mode transfer data

1. Synchronous mode

Baris kendali digunakan untuk mengsinkronkan waktu pada semua kejadian yang terjadi selama periode waktu tertentu.

Kelemahan:

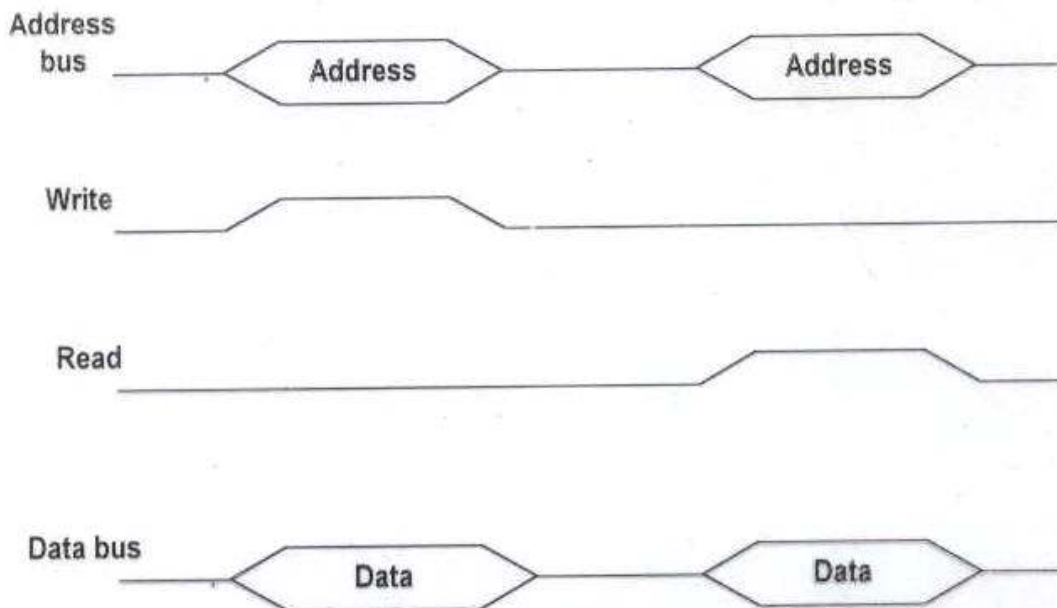
- Tiap piranti I/O berbeda-beda kecepatan operasinya, sehingga harus diturunkan pada kecepatan yang paling rendah.



Gambar 2. Sinyal Pengaturan waktu mode sinkron

2. Asynchronous mode

Menggunakan teknik jabat tangan (*hand shaking*) untuk menyakinkan transfer data antara pengirim dan penerima tidak ada kesalahan (data valid)



Gambar 3. Sinyal Pengaturan waktu mode asinkron

Kelemahan :

- memerlukan lebih banyak kendali
- kecepatan transfer lebih rendah dari yang sebenarnya.

Kelebihan :

- memungkinkan penggunaan piranti I/O yang memiliki berbagai variasi kecepatan dalam system yang sama.

Sistem Prosesor I/O

Saluran I/O:

Merupakan sebuah prosesor khusus dengan kemampuan terbatas yang disusun untuk interface beberapa piranti I/O ke memori.

- Saluran I/O dapat melakukan pendeteksian dan pembetulan kesalahan dan beroperasi dalam basis *cycle stealing*.
- Saluran I/O berkomunikasi dengan CPU sebagai suatu fasilitas DMA dan berkomunikasi dengan piranti I/O seolah-olah sebuah CPU.

Karena piranti I/O mempunyai kecepatan transfer yang berbeda-beda, maka saluran dibagi menjadi 3 pelayanan, yaitu:

- Saluran Multiplexer

Digunakan untuk menghubungkan piranti yang berkecepatan rendah dan sedang serta serta mengoperasikannya secara bersamaan dengan multiplexing.

- Saluran Selektor

Digunakan untuk menghubungkan piranti I/O yang

berkecepatan tinggi tanpa multiplexing.
Contoh: pita magnetis, disk

-Saluran Multiplexer Blok Merupakan kombinasi dari dua pelayanan diatas.

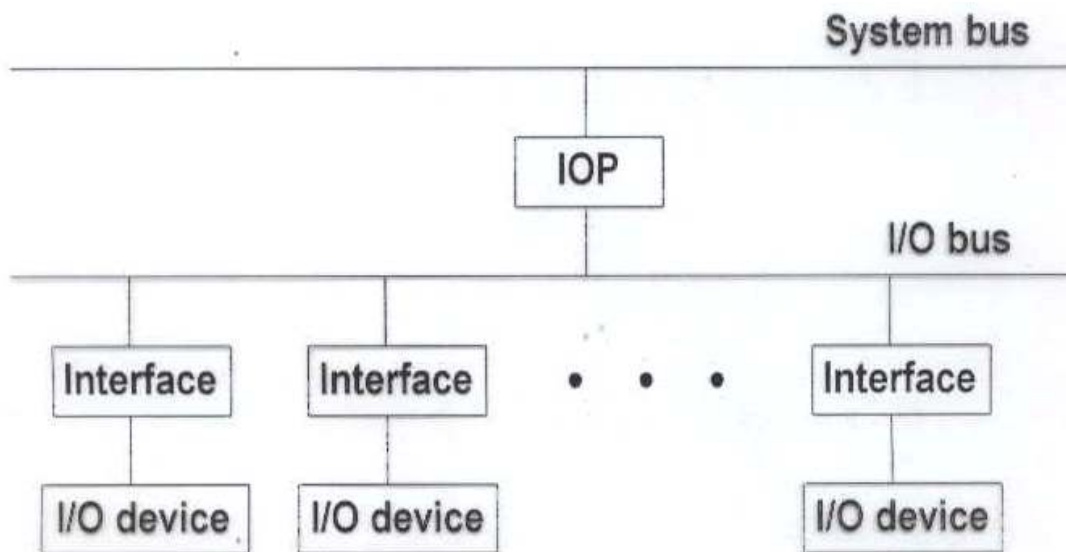
Prosesor I/O (IOP)

Merupakan komputer umum yang berkomunikasi dengan memori utama melalui fasilitas DMA system bus dan dengan piranti I/O atas satu atau lebih bus I/O.

Ada 2 mode yaitu :

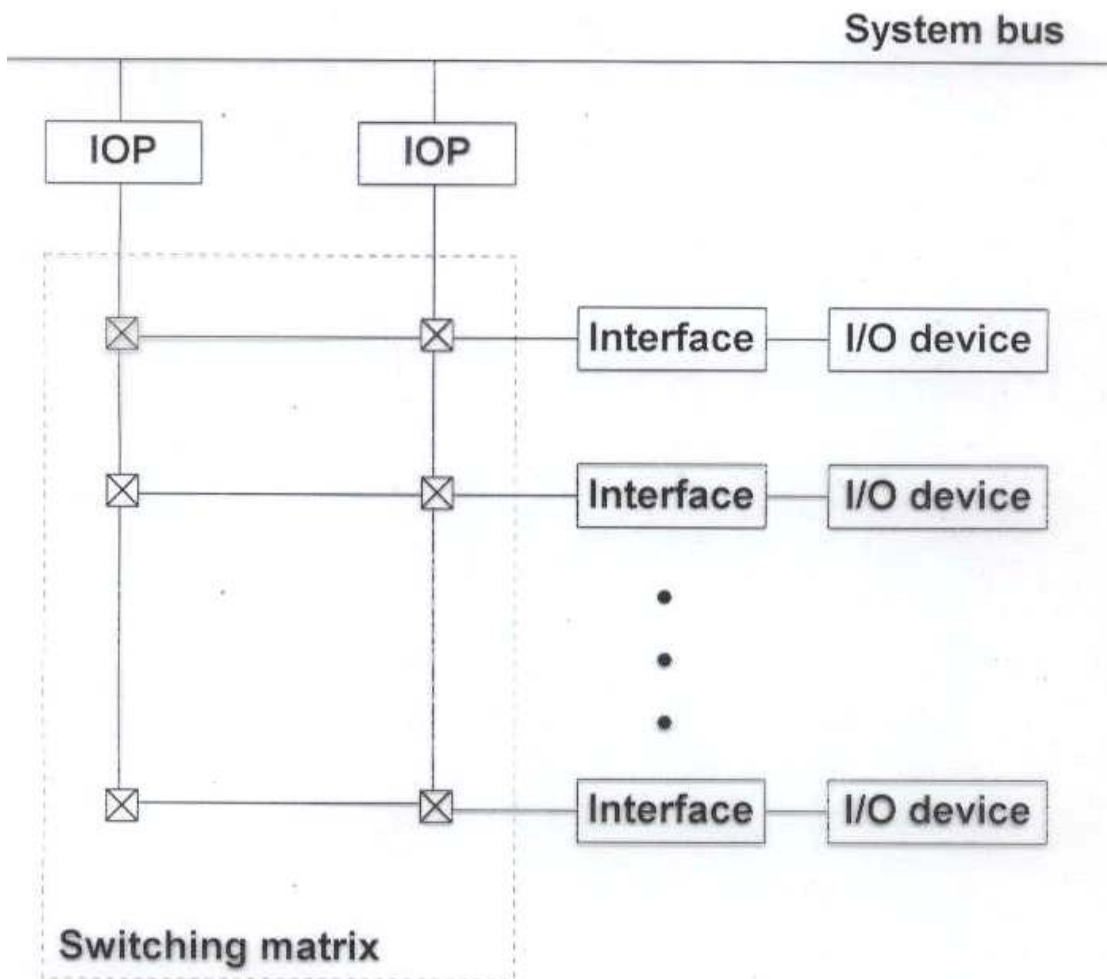
1. Single Shared bus

Setiap IOP mengendalikan sejumlah piranti I/O tertentu yang tetap.



Gambar 4. Model Single Shared bus

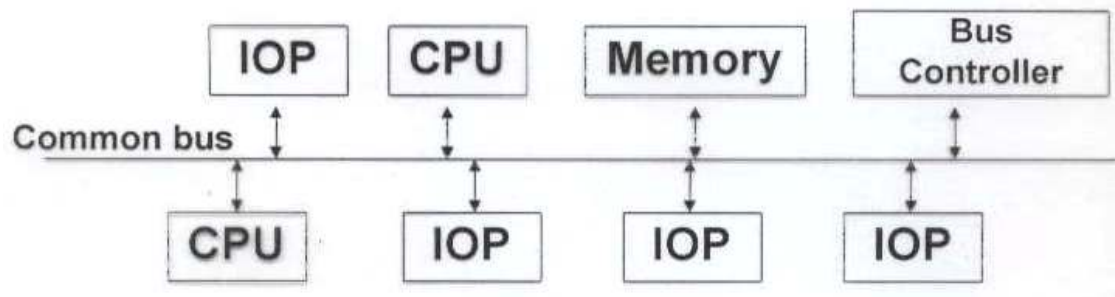
2. Switching matriks bus
Setiap IOP mengendalikan satu piranti I/O



Gambar 5. Model Switching matriks bus

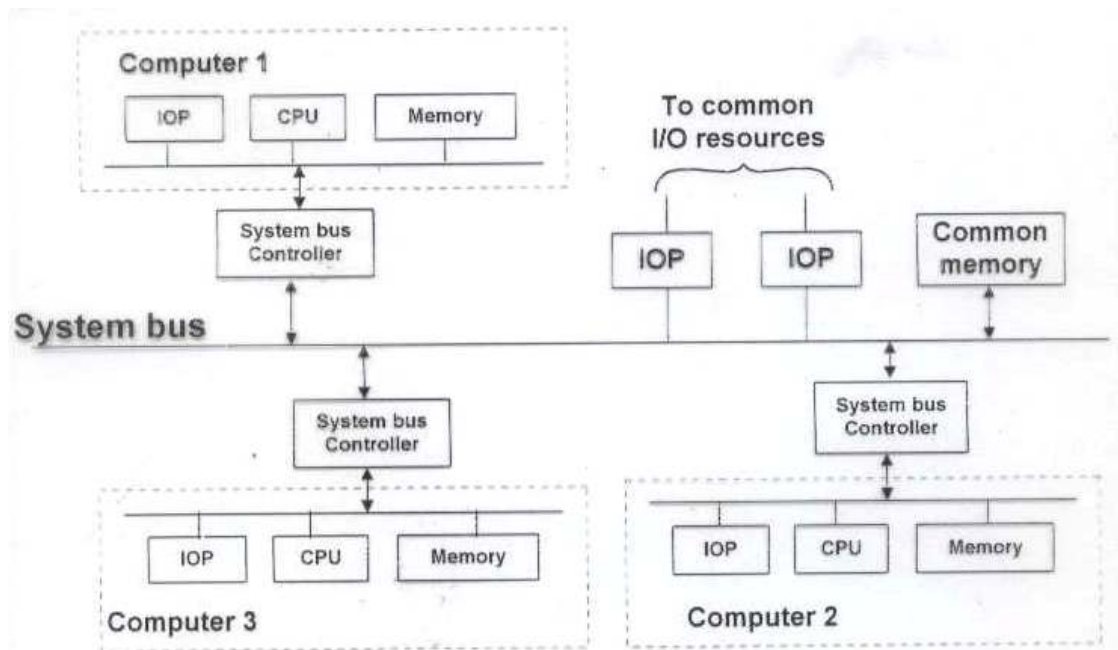
Konfigurasi Multiprosesor

Di dalam satu komputer seakan-akan terdapat beberapa mikroprosesor, meskipun sebenarnya mikroprosesor utamanya hanya satu, sedangkan yang lainnya berupa prosesor I/O (IOP). Hubungan yang paling sederhana menggunakan common bus.



Gambar 6. Model common bus pada multiprosesor

- Bus umum bersifat membagi waktu (*time shared*) oleh semua prosesor dan hanya satu prosesor yang dapat mengakses memori pada waktu tertentu. Tetapi dapat juga menggunakan bus umum ke dalam organisasi multiprosesor dual bus.
- Setiap komputer dihubungkan suatu pengendali sistem ke bus umum
- Komunikasi interkomputer ini dilakukan pada sistem bus melalui memori umum.



Gambar 7. Model system bus pada multiprosesor