

Arsitektur dan Organisasi Komputer

COM 60011

Topik #7 – Input / Output



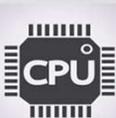
Tujuan

CPMK : Mahasiswa mampu menjabarkan antarmuka dari input-output (IO) dan peripheral-periperel lainyang ada maupun terhubung pada suatu komputer.

Sub CPMK : Mahasiswa mampu menjelaskan tentang antarmuka dari input-output maupun peripheral yang ada maupun terhubung pada komputer.

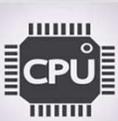
Materi Terkait:

- **Sistem i/o**
- **Sungsi modul i/o**
- **Struktur modul i/o**
- **interupt driven 1/o**
- **DMA**
- **Perangkat eksternal**





Input / Output





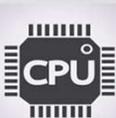
Pendahuluan (1)

Input / Output (I/O) adalah salah satu elemen penting pada sebuah komputer, selain CPU dan sejumlah modul memori.

Modul I/O bukan hanya merupakan konektor mekanis sederhana yang menghubungkan suatu perangkat dengan bus sistem.

Akan tetapi modul I/O berisi sesuatu yang “cerdas” yaitu berisi logika untuk melakukan fungsi komunikasi antara peripheral dengan bus.

Unit Input/Output (I/O) adalah bagian dari sistem mikroprosesor yang digunakan oleh mikroprosesor itu untuk berhubungan dengan dunia luar / lingkungan operasi komputer.





Pendahuluan (2)

Unit input adalah unit luar yang digunakan untuk memasukkan data dari luar ke dalam CPU, contohnya data yang berasal dari keyboard atau mouse.

Sementara unit output biasanya digunakan untuk menampilkan atau menyajikan data.

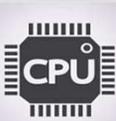
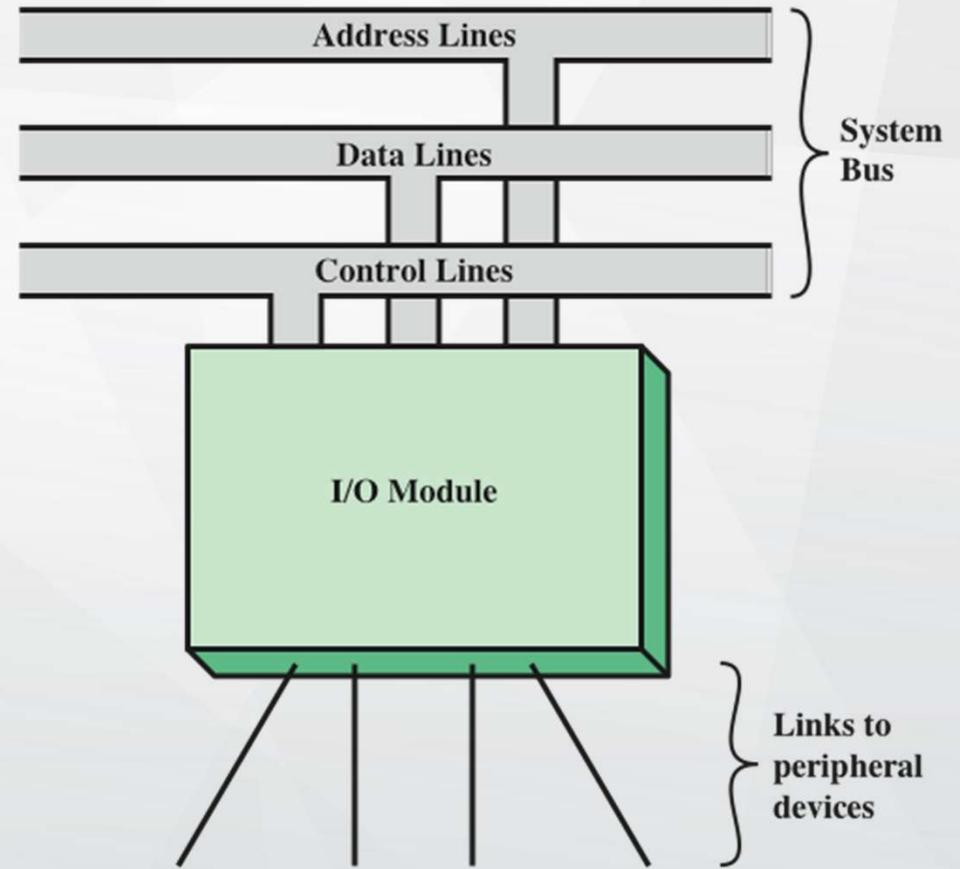
Ada beberapa alasan kenapa perangkat I/O tidak langsung dihubungkan dengan bus sistem komputer, yaitu :

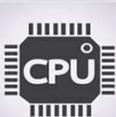
- Bervariasinya metode operasi piranti peripheral, sehingga tidak praktis apabila system komputer harus menangani berbagai macam sistem operasi piranti peripheral tersebut.
- Kecepatan transfer data piranti peripheral umumnya lebih lambat dari pada laju transfer data pada CPU maupun memori.
- Format data dan panjang data pada piranti peripheral seringkali berbeda dengan CPU, sehingga perlu modul untuk menselaraskannya.

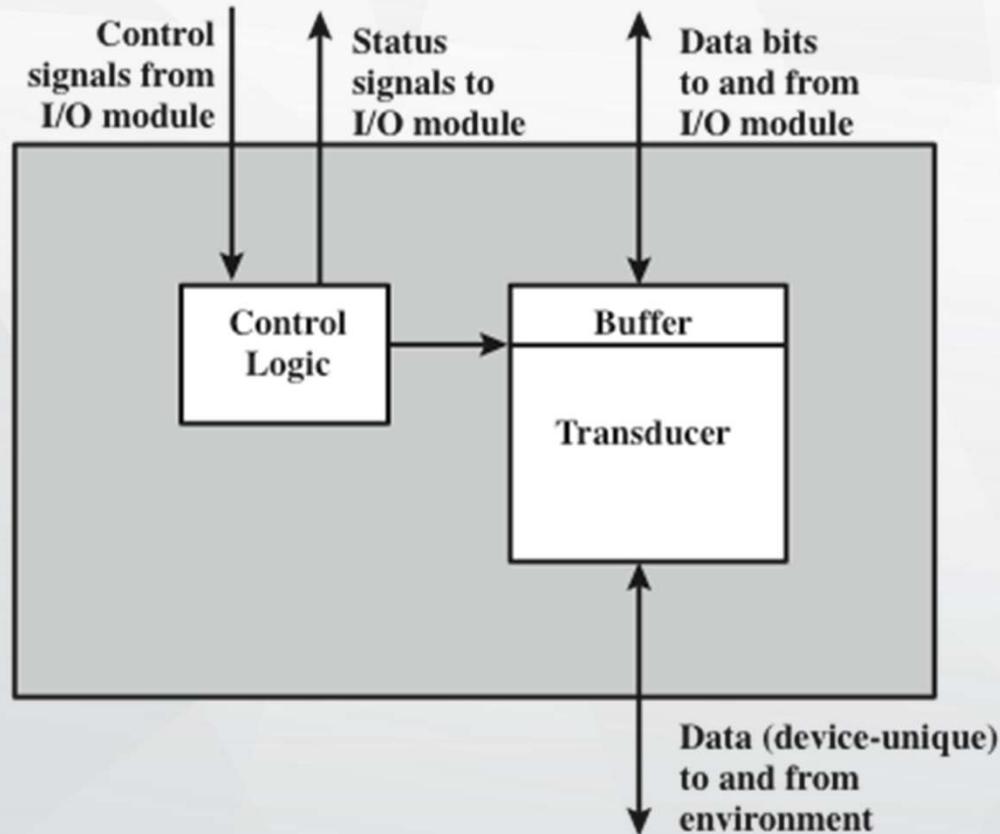




Generic Model of an I/O Module







External Device Block Diagram





Penjelasan External Device Block Diagram (1)

Fungsi kontrol dan pewaktuan (control & timing) merupakan hal yang penting untuk mensinkronkan kerja masing – masing komponen penyusun komputer.

1. Permintaan dan pemeriksaan status perangkat dari CPU ke modul I/O.
2. Modul I/O memberi jawaban atas permintaan CPU.
3. Apabila perangkat eksternal telah siap untuk transfer data, maka CPU akan mengirimkan perintah ke modul I/O.
4. Modul I/O akan menerima paket data dengan panjang tertentu dari peripheral.
5. Selanjutnya data dikirim ke CPU setelah diadakan sinkronisasi panjang data dan kecepatan transfer oleh modul I/O sehingga paket – paket data dapat diterima CPU dengan baik.



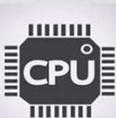


Penjelasan External Device Block Diagram (2)



Transfer data tidak akan lepas dari penggunaan sistem bus, maka interaksi CPU dan modul I/O akan melibatkan kontrol dan pewaktuan sebuah arbitrase bus atau lebih.

1. Command Decoding, yaitu modul I/O menerima perintah – perintah dari CPU yang dikirimkan sebagai sinyal bagi bus kontrol. Misalnya, sebuah modul I/O untuk disk dapat menerima perintah: Read sector, Scan record ID, Format disk.
2. Data, pertukaran data antara CPU dan modul I/O melalui bus data.
3. Status Reporting, yaitu pelaporan kondisi status modul I/O maupun perangkat peripheral, umumnya berupa status kondisi Busy atau Ready. Juga status bermacam – macam kondisi kesalahan (error).
4. Address Recognition, bahwa peralatan atau komponen penyusun komputer dapat dihubungi atau dipanggil maka harus memiliki alamat yang unik, begitu pula pada perangkat peripheral, sehingga setiap modul I/O harus mengetahui alamat peripheral yang dikontrolnya.



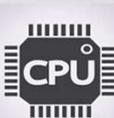


Penjelasan External Device Block Diagram (3)



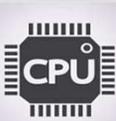
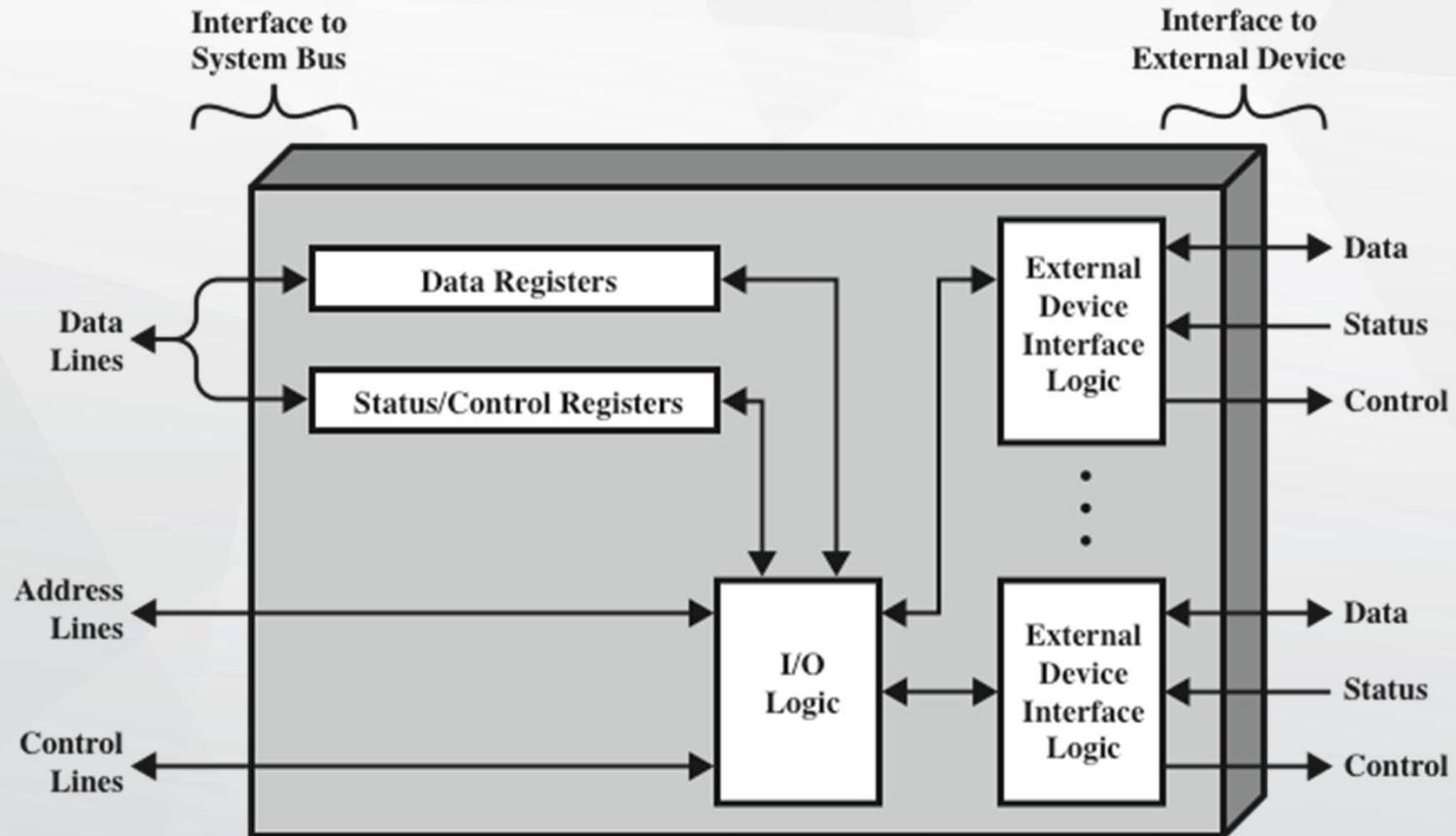
Fungsi selanjutnya adalah buffering. Tujuan utama buffering adalah mendapatkan penyesuaian data sehubungan perbedaan laju transfer data dari perangkat peripheral dengan kecepatan pengolahan pada CPU.

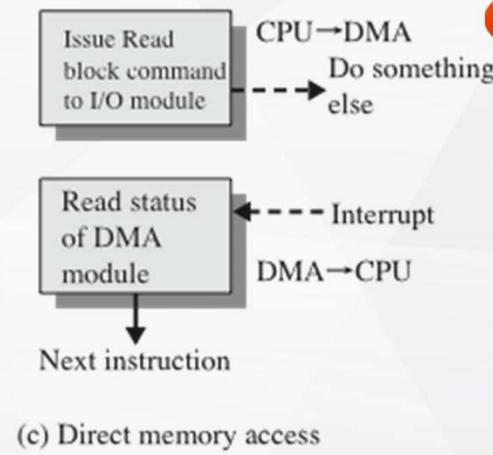
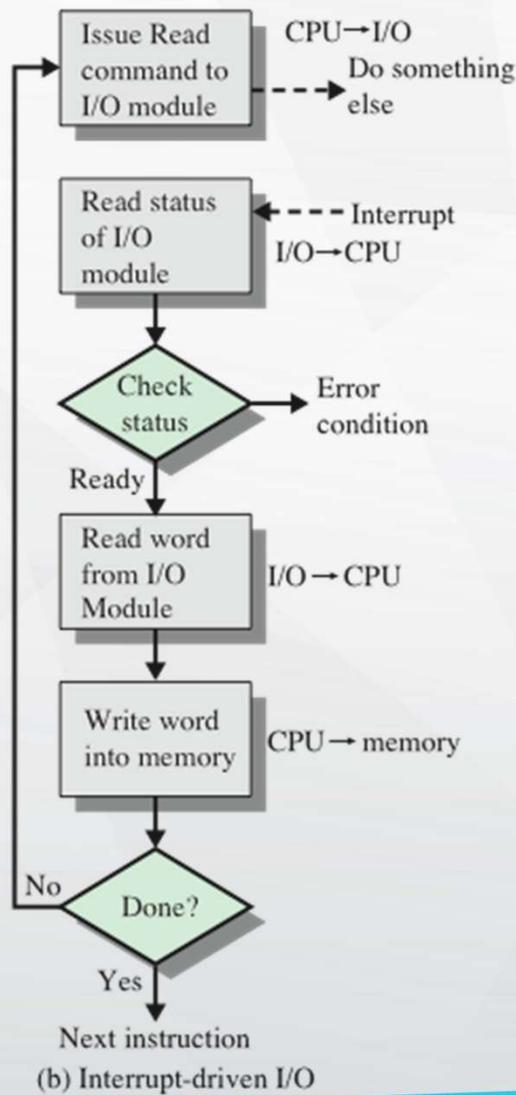
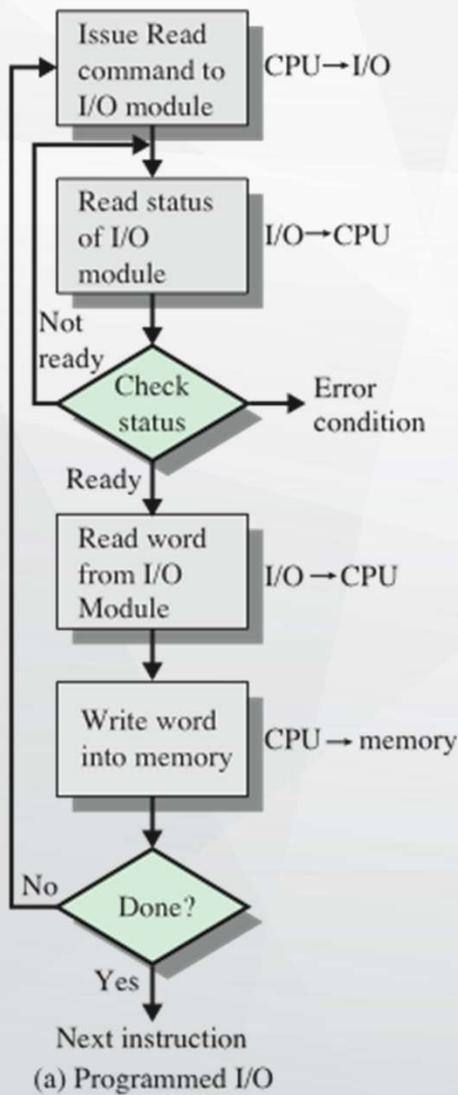
Fungsi terakhir adalah deteksi kesalahan. Apabila pada perangkat peripheral terdapat masalah sehingga proses tidak dapat dijalankan, maka modul I/O akan melaporkan kesalahan tersebut.





I/O Module Structure





Three Techniques for Input of a Block of Data

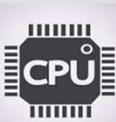




Aplikasi Antarmuka I/O

Ragam dari antarmuka I/O

- Character-stream atau block,
- Sequential atau random-access,
- Synchronous atau asynchronous,
- Sharable atau dedicated,
- Speed atau operation,
- Read-write, read only, write only.





I/O Technique

Programmed I/O

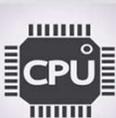
data saling dipertukarkan antara CPU dan modul I/O. CPU mengeksekusi program yang memberikan operasi I/O kepada CPU secara langsung, seperti pemindahan data, pengiriman perintah baca maupun tulis, dan monitoring perangkat.

Interrupt Driven I/O

Prosesnya adalah CPU mengeluarkan perintah I/O pada modul I/O, bersamaan perintah I/O dijalankan modul I/O maka CPU akan melakukan eksekusi perintah – perintah lainnya.

Direct Memory Access

untuk menangani transfer data bervolume besar serta privilege yang lebih tinggi





Programmed I/O

Untuk melaksanakan perintah – perintah I/O, CPU akan mengeluarkan sebuah alamat bagi modul I/O dan perangkat peripheralnya. empat klasifikasi perintah I/O, yaitu:

1. Perintah control : Perintah ini digunakan untuk mengaktivasi perangkat peripheral dan memberitahukan tugas yang diperintahkan padanya.
2. Perintah test : Perintah ini digunakan CPU untuk menguji berbagai kondisi status modul I/O dan peripheralnya. CPU perlu mengetahui perangkat peripheralnya dalam keadaan aktif dan siap digunakan, untuk mengetahui operasi yang bisa dilakukan serta mendeteksi kesalahannya.
3. Perintah read : Perintah pada modul I/O untuk mengambil suatu paket data kemudian menaruh dalam buffer internal.
4. Perintah write.
Perintah ini kebalikan dari read. CPU memerintahkan modul I/O untuk mengambil data dari bus data untuk diberikan pada perangkat peripheral tujuan data tersebut.



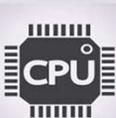


I/O Commands

Dalam teknik Programmed I/O, terdapat dua implementasi perintah I/O, antara lain :

Memory-mapped I/O, terdapat ruang tunggal untuk lokasi memori dan perangkat I/O. CPU memperlakukan register status dan register data modul I/O sebagai lokasi memori dan menggunakan instruksi mesin yang sama untuk mengakses baik memori maupun perangkat I/O. Konsekuensinya adalah diperlukan saluran tunggal untuk pembacaan dan saluran tunggal untuk penulisan. Keuntungan memory-mapped I/O adalah efisien dalam pemrograman, namun memakan banyak ruang memori alamat.

Isolated I/O, dilakukan pemisahan ruang pengalamatan bagi memori dan ruang pengalamatan bagi I/O. Dengan teknik ini diperlukan bus yang dilengkapi dengan saluran pembacaan dan penulisan memori ditambah saluran perintah output. Keuntungan isolated I/O adalah sedikitnya instruksi I/O.





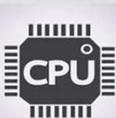
Interrupt-Driven I/O

Masalah dengan Programmed I/O adalah prosesor harus menunggu lama hingga modul I/O siap untuk penerimaan atau transmisi data.

Alternatifnya adalah prosesor mengeluarkan perintah ke modul I/O dan kemudian melanjutkan untuk melakukan pekerjaan lainnya

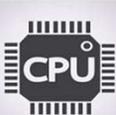
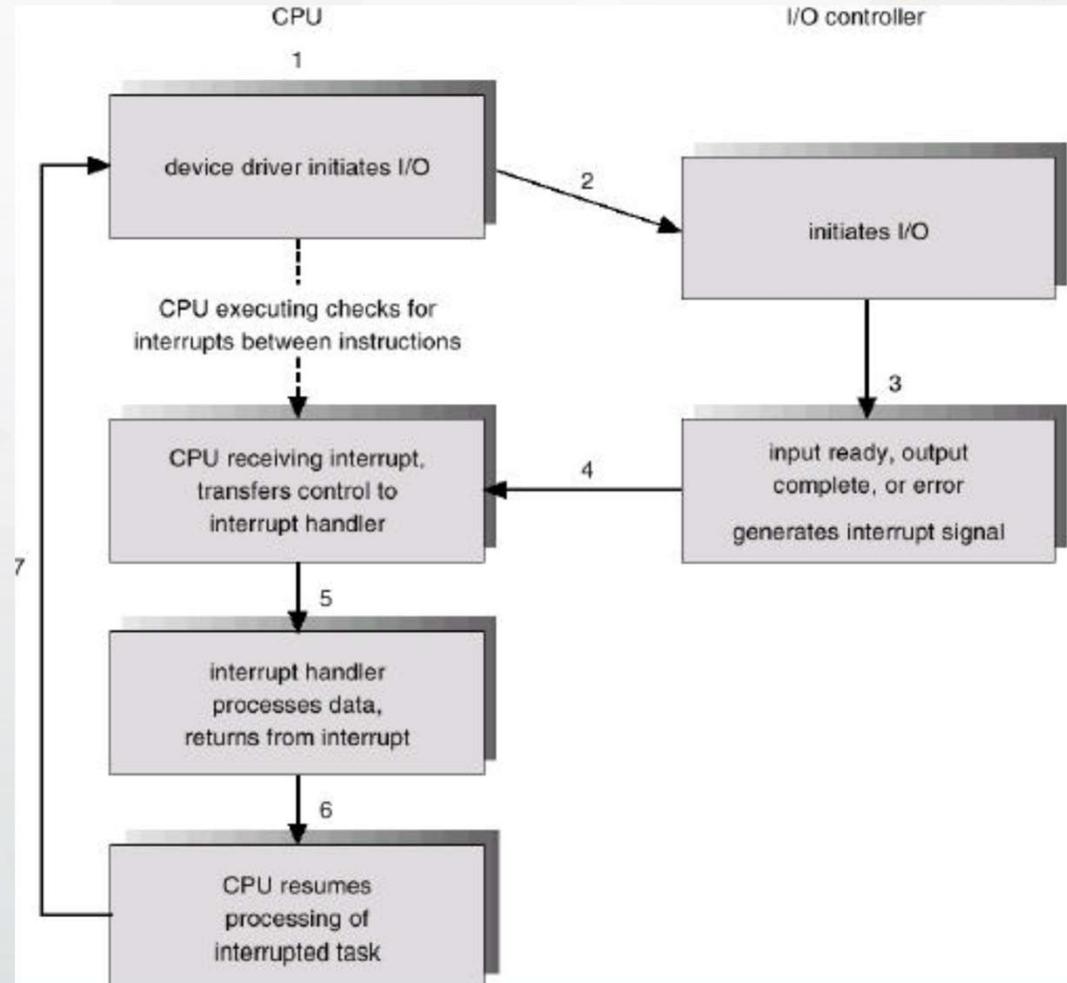
Modul I/O kemudian akan menginterupsi prosesor untuk meminta layanan ketika siap untuk bertukar data dengan prosesor

Prosesor menjalankan transfer data dan melanjutkan proses sebelumnya





Interrupt-Driven I/O Cycle





Interrupt Technique

Multiple Interrupt Lines antara CPU dan modul – modul I/O. Namun tidak praktis untuk menggunakan sejumlah saluran bus atau pin CPU ke seluruh saluran interupsi modul I/O.

Software Poll. Prosesnya, apabila CPU mengetahui adanya sebuah interupsi, maka CPU akan menuju ke routine layanan interupsi yang tugasnya melakukan poll seluruh modul I/O untuk menentukan modul yang melakukan interupsi.

Kerugian software poll adalah memerlukan waktu yang lama karena harus mengidentifikasi seluruh modul untuk mengetahui modul I/O yang melakukan interupsi.

Deasy Chain, yang menggunakan hardware poll. Seluruh modul I/O tersambung dalam saluran interupsi CPU secara melingkar (chain). Apabila ada permintaan interupsi, maka CPU akan menjalankan sinyal acknowledge yang berjalan pada saluran interupsi sampai menemui modul I/O yang mengirimkan interupsi.

Bus Arbitration. Dalam metode ini, pertama – tama modul I/O memperoleh kontrol bus sebelum modul ini menggunakan saluran permintaan interupsi. Dengan demikian hanya akan terdapat sebuah modul I/O yang dapat melakukan interupsi.



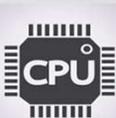


Direct Memory Access (DMA)

Prinsip kerja DMA adalah CPU akan mendelegasikan kerja I/O kepada DMA, CPU hanya akan terlibat pada awal proses untuk memberikan instruksi lengkap pada DMA dan akhir proses saja.

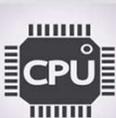
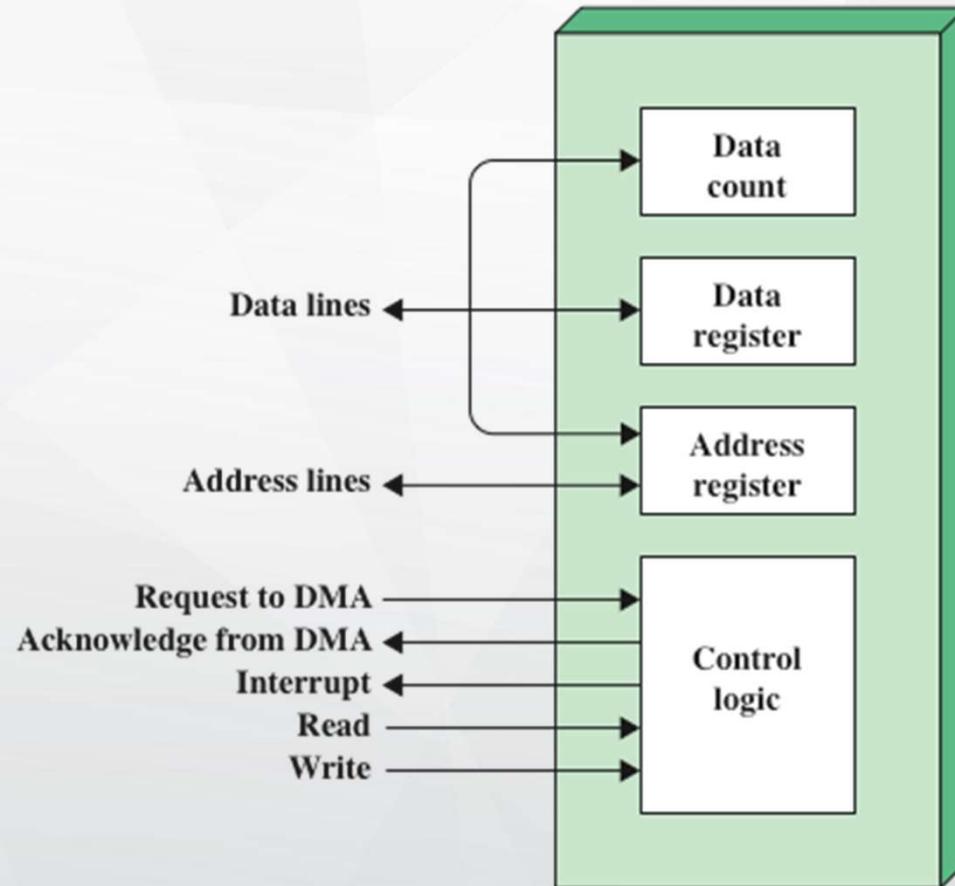
Dengan demikian CPU dapat menjalankan proses lainnya tanpa banyak terganggu dengan interupsi.

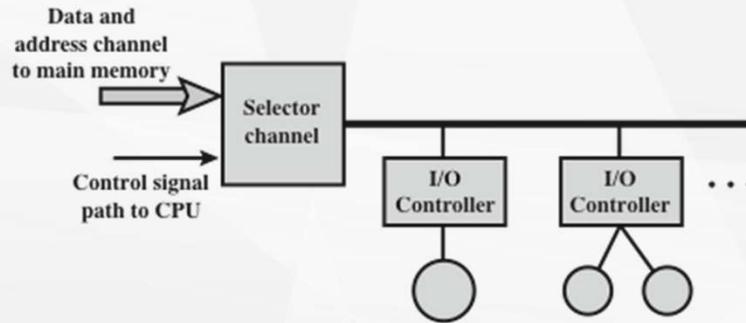
Dalam melaksanakan transfer data secara mandiri, DMA memerlukan pengambilalihan kontrol bus dari CPU. Untuk itu DMA akan menggunakan bus bila CPU tidak menggunakannya atau DMA memaksa CPU untuk menghentikan sementara penggunaan bus.



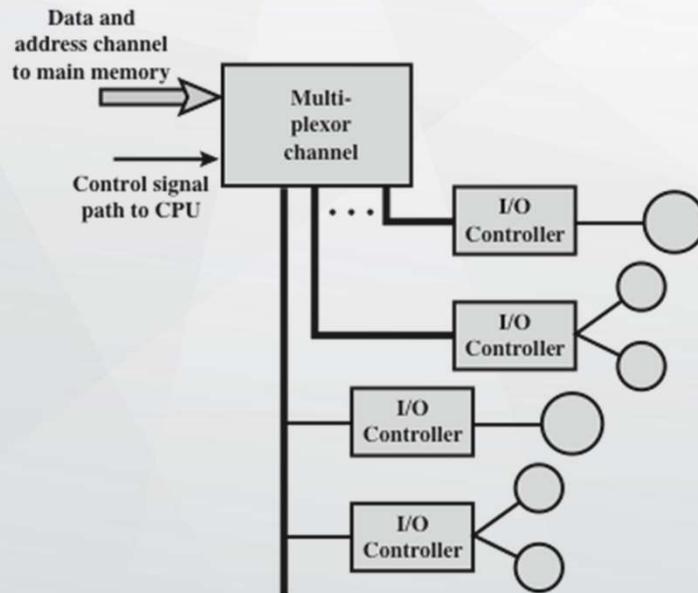


Typical DMA Module Diagram



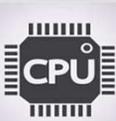


(a) Selector



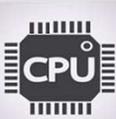
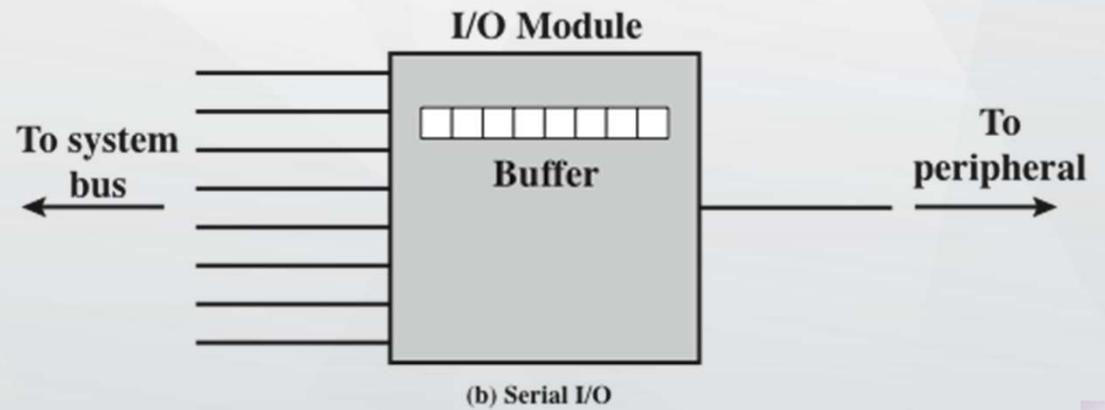
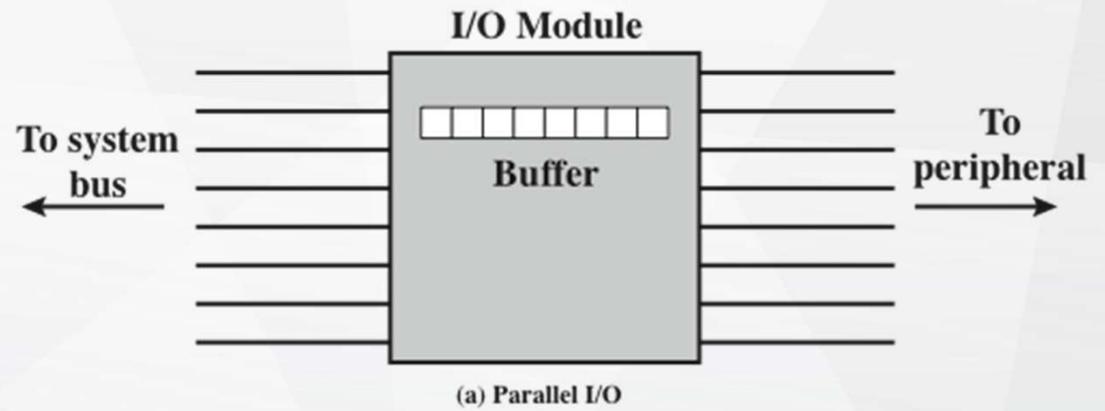
(b) Multiplexor

I/O Channel Architecture





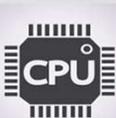
Parallel and Serial I/O





Evolution of the I/O Function

- CPU secara langsung mengontrol perangkat perifer.
- Control Unit pada modul I/O ditambahkan. CPU menggunakan programmed I/O tanpa interupsi.
- Konfigurasi yang sama seperti pada poin 2 digunakan, tetapi sekarang interupsi digunakan. CPU tidak perlu menghabiskan waktu menunggu operasi I/O dilakukan, sehingga meningkatkan efisiensi.
- Modul I/O diberikan akses langsung ke memori melalui DMA. Sekarang dapat memindahkan satu blok data ke atau dari memori tanpa melibatkan CPU, kecuali di awal dan akhir transfer.
- Modul I/O ditambahkan CPU sendiri, dengan set instruksi khusus yang disesuaikan untuk I/O
- Modul I/O memiliki memori lokalnya sendiri dan, pada kenyataannya, merupakan komputer tersendiri.





Terima Kasih

Pustaka : William Stallings, “Computer Organization and Architecture Designing for Performance Eighth Edition”, Prentice Hall, 2019

