



# Arsitektur Komputer

Prodi Teknologi Rekayasa Internet

# SYSTEM KOMPUTER 80

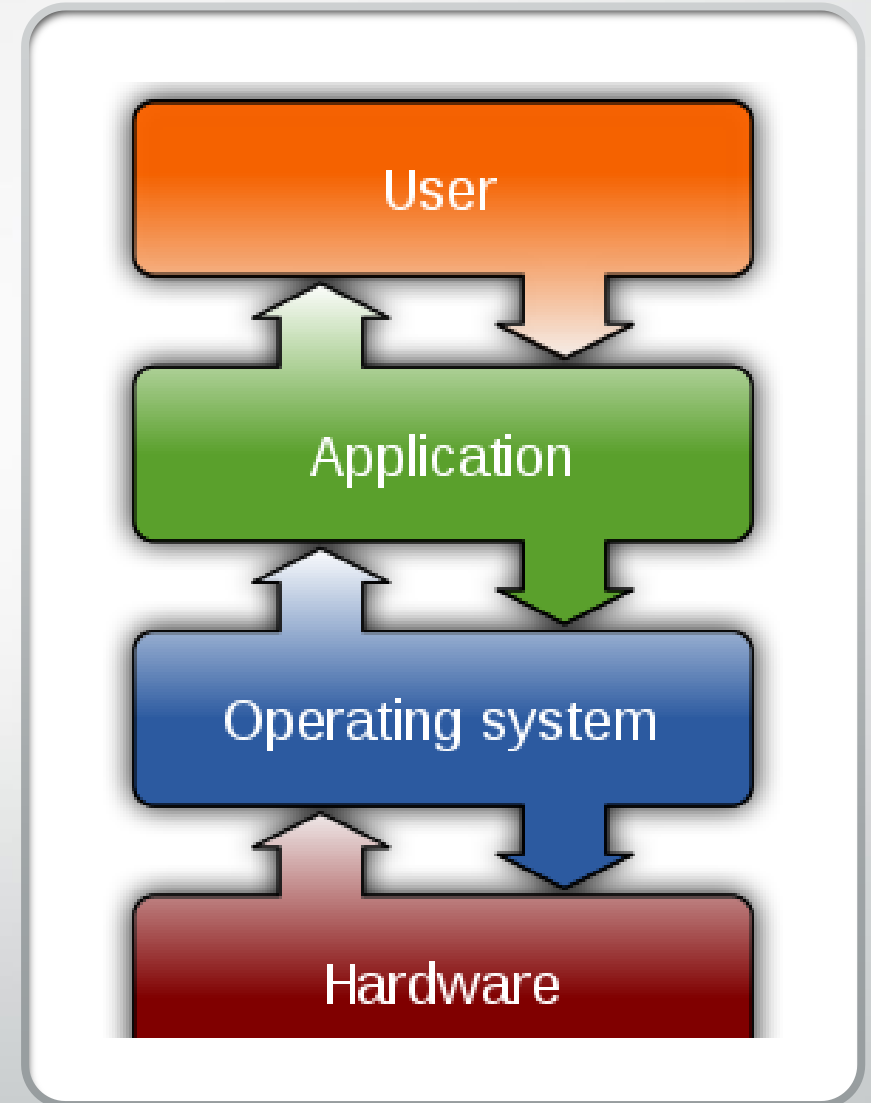
## *Capaian Pembelajaran :*

Setelah menyelesaikan bab ini, diharapkan dapat:

- Memahami elemen dasar dari loop instruksi dan peran interupsi.
- Mendeskripsikan konsep interaksi dalam sistem komputer.
- Mengevaluasi keuntungan relatif dari koneksi point-to-point versus koneksi bus.
- Memberikan gambaran umum tentang QPI.
- Berikan gambaran umum tentang PCIe.

# Interrupt/ Interupsi

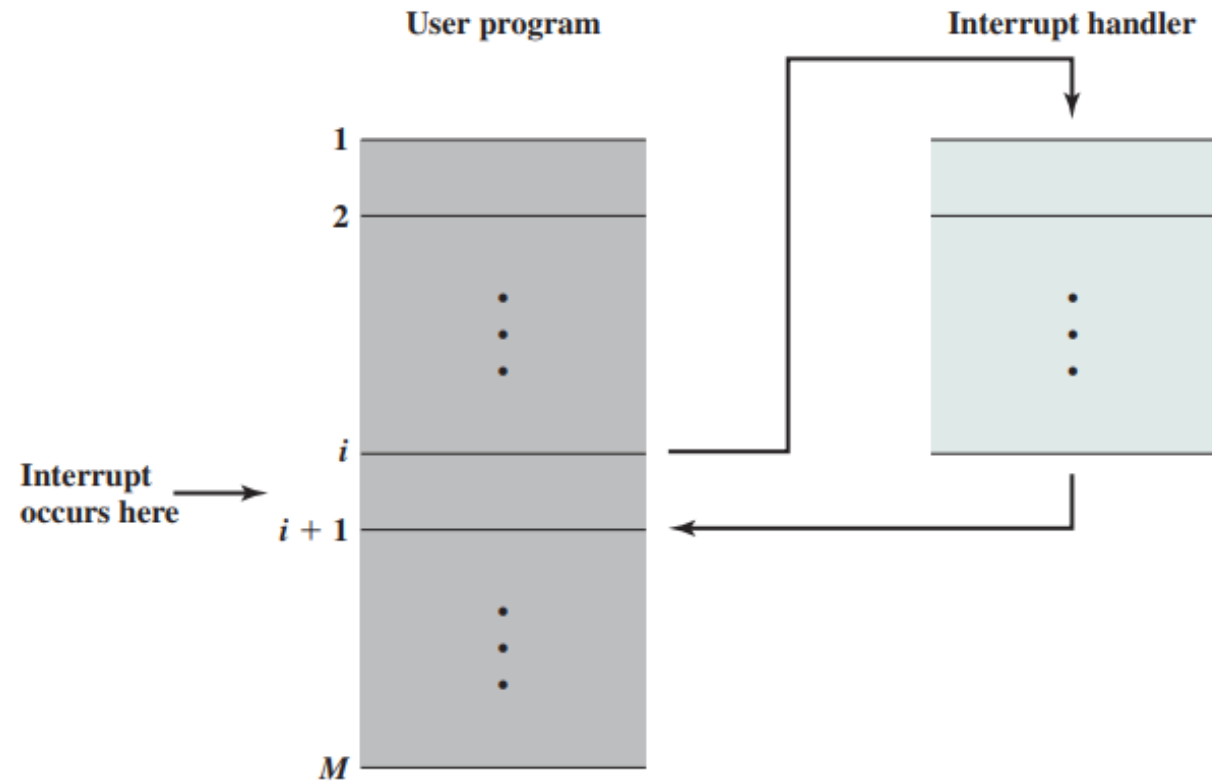
Aliran *Proses* pada  
sebuah computer



# Interupsi

Interupsi adalah respons prosesor terhadap suatu proses yang memerlukan perhatian dari perangkat lunak

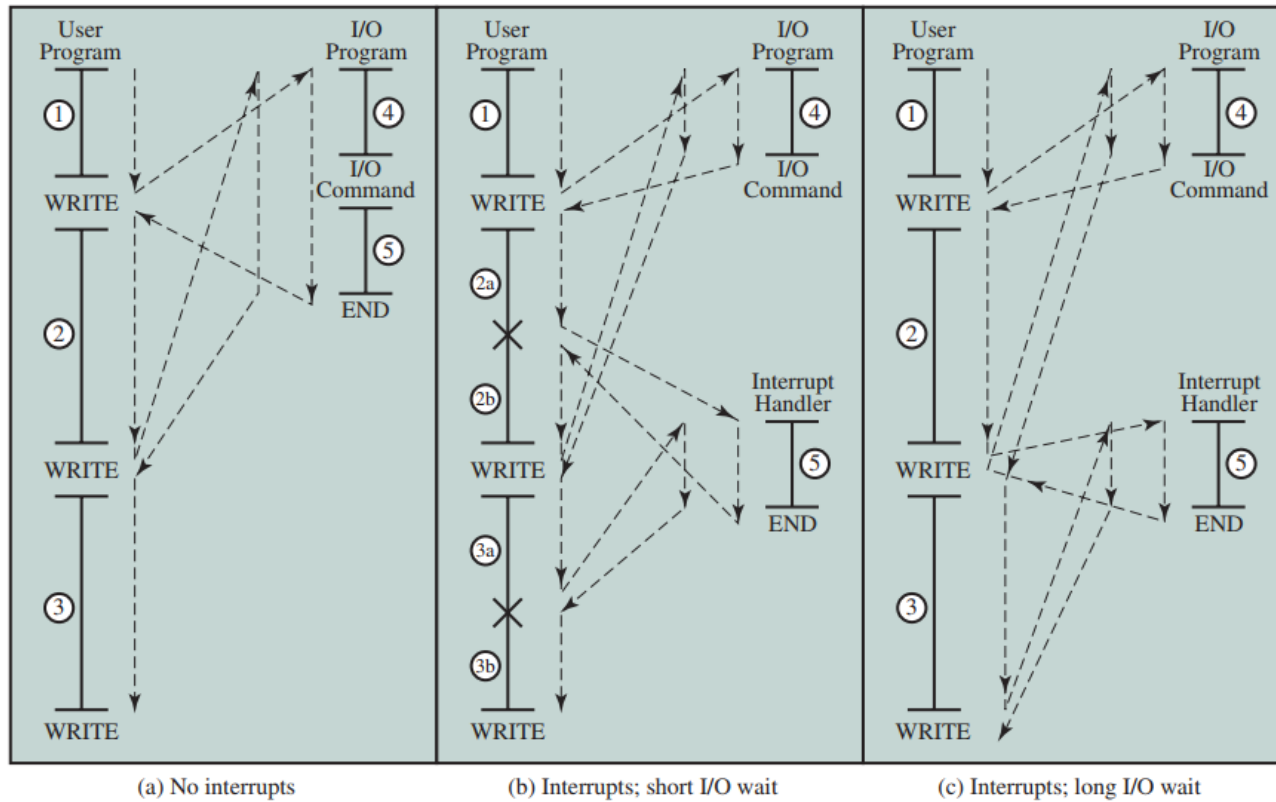
Kondisi interupsi memperingatkan prosesor dan berfungsi sebagai permintaan prosesor untuk menghentikan sementara apa yang sedang dikerjakan dan menjalankan apa yang telah diinterupsi dan jika sudah selesai maka prosesor akan melanjutkan proses yang sedang dia kerjakan.



**Figure 3.8** Transfer of Control via Interrupts

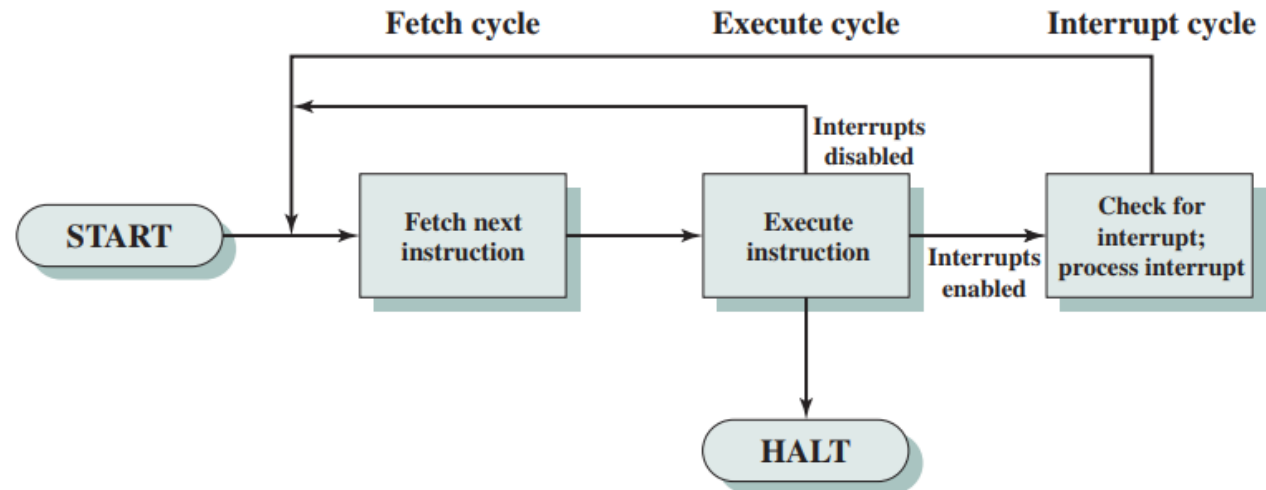
# Mekanisme Interupsi

# Interupsi dan Tanpa Interupsi



# Penambahan Interupsi

- Untuk mengakomodasi/ memberikan kesempatan interupsi maka ditambahkan siklus interupsi ke dalam set instuction



# Kelas 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.

**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.



# 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.

Saat sinyal interupsi diterima prosesor ada dua kemungkinan tindakan, yaitu interupsi diterima/**ditanggguhkan** dan interupsi ditolak.

- Saat ditanggguhkan Prosesor melakukan hal berikut:
  - Menjeda eksekusi program dan menyimpan konteksnya. Ini berarti menyimpan alamat instruksi berikutnya yang akan dieksekusi. (konten penghitung program saat jeda) dan data lain apa pun yang terkait dengan aktivitas prosesor saat itu.
  - Mengatur penghitung program ke alamat awal subrutin pengendali interupsi.

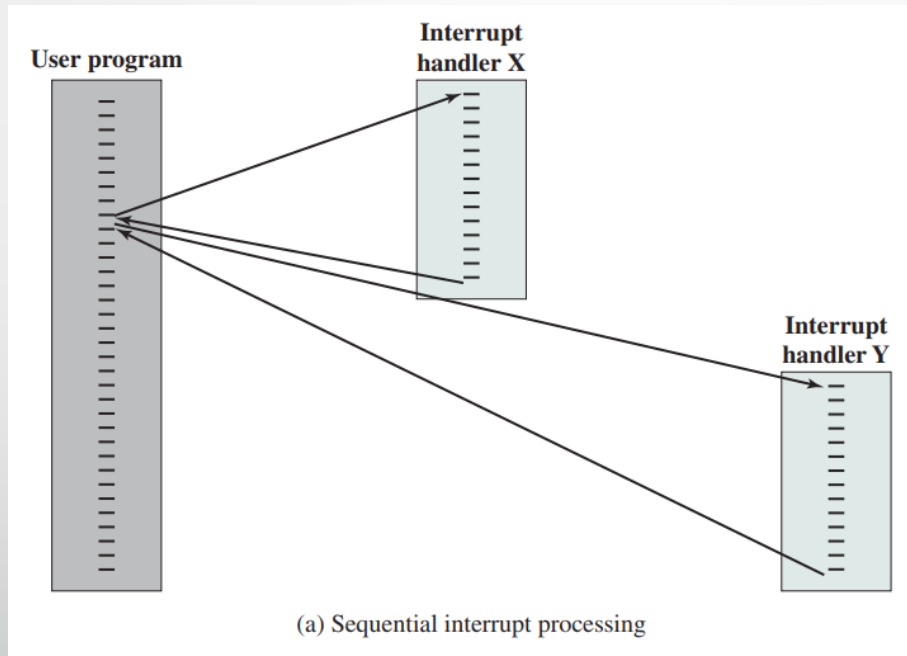
# Multiple Interrupts

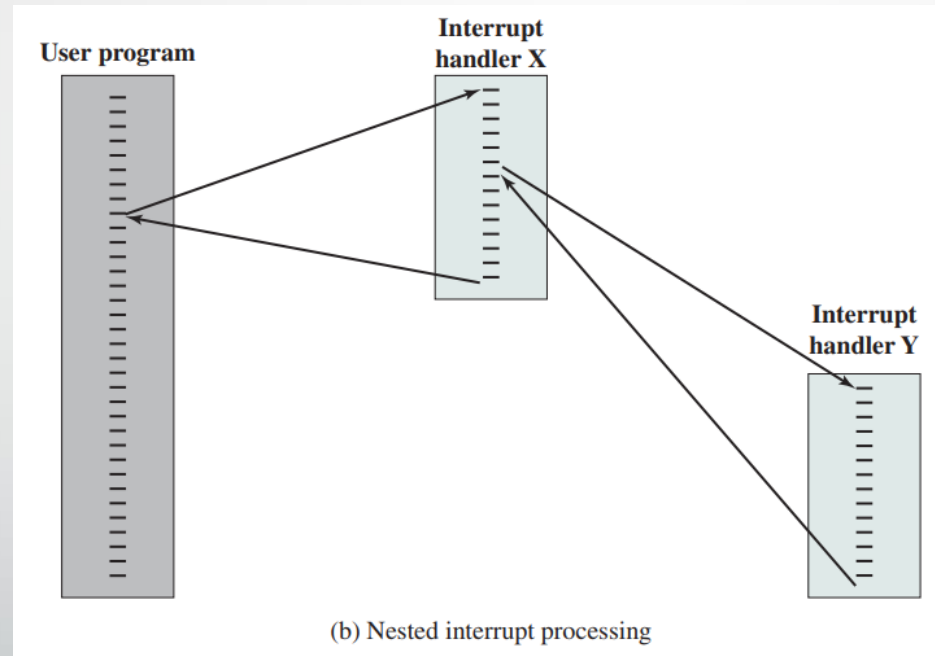
Interupsi ganda (multiple interrupt). Misalnya suatu komputer akan menerima permintaan interupsi saat proses pencetakan dengan printer selesai, disamping itu dimungkinkan dari saluran komunikasi akan mengirimkan permintaan interupsi setiap kali data tiba.

Dapat diambil dua buah pendekatan untuk menangani interupsi ganda ini

## 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





# Contoh Kasus

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



# Bus Interconnection

- Bus adalah jalur komunikasi yang menghubungkan dua atau lebih perangkat. Karakteristik utama dari bus adalah bahwa itu adalah **media transmisi bersama**. Beberapa perangkat terhubung ke bus dan sinyal yang ditransmisikan oleh satu perangkat tersedia untuk diterima oleh semua perangkat lain yang terhubung ke bus. Jika dua perangkat mentransmisikan selama jumlah waktu yang sama, sinyal mereka akan tumpang tindih dan kacau. Dengan demikian, **hanya satu perangkat pada satu waktu yang dapat berhasil mentransmisikan**.

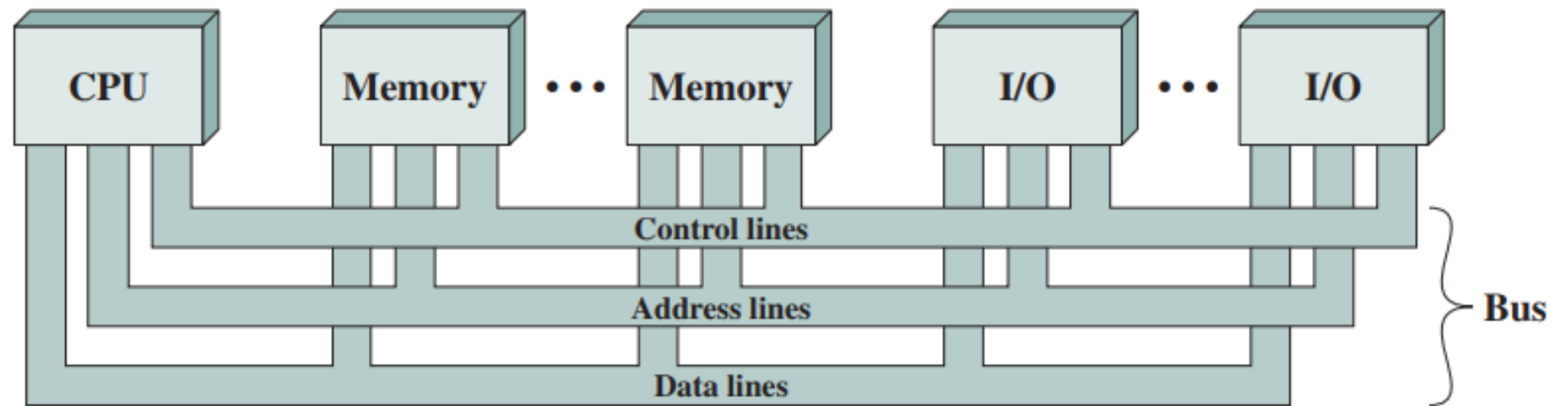


# Bus Interconnection

- Biasanya, sebuah bus terdiri dari beberapa jalur komunikasi, atau jalur. Setiap baris mampu mentransmisikan sinyal yang mewakili biner 1 dan biner 0. Seiring waktu, urutan digit biner dapat ditransmisikan pada satu baris. Bersama-sama, beberapa jalur pada bus dapat digunakan untuk mengirimkan digit biner secara bersamaan (secara paralel). Misalnya, unit data 8-bit dapat ditransmisikan pada delapan jalur bus.

# System Bus

- Adakah sebuah bus yang menghubungkan komponen utama komputer (prosesor, memori, I/O)
- Biasanya terdiri dari sekitar lima puluh hingga ratusan jalur terpisah. Setiap baris diberi arti atau fungsi tertentu.
- Diklasifikasikan menjadi tiga kelompok fungsional :
  - jalur data, alamat, dan kontrol serta jalur distribusi daya yang memasok daya ke modul yang terhubung



# Beberapa Perintah di Control Lines

- **Memory Write:** Menyebabkan data bus ditulis ke lokasi yang dialamatkan.
- **Memory Read:** Menyebabkan data lokasi alamat ditempatkan pada bus.
- **Write I/O:** Menyebabkan data pada bus dikirim ke port I/O yang dituju.
- **Read I/O:** menyebabkan data dari port I/O yang dialamatkan ditempatkan pada bus.
- **Transfer ACK:** menunjukkan bahwa data telah diterima atau ditempatkan di bus.
- **Bus Request:** Menunjukkan bahwa modul perlu mendapatkan kendali atas bus.
- **Bus Grant:** Menunjukkan bahwa modul yang meminta telah menguasai bus.
- **Interrupt Request :** Menunjukkan bahwa interupsi tertunda.
- **Interrupt ACK:** Mengonfirmasi bahwa interupsi yang tertunda telah diakui.
- **Clock:** digunakan untuk menyinkronkan operasi.
- **Restart:** Menginisialisasi semua modul.



# Terima Kasih