Organisasi dan Arsitektur Komputer : Perancangan Kinerja (William Stallings)

Chapter 3

Bus Sistem

Konsep Program

- Pemrograman (hardware) merupakan proses penghubungan berbagai komponen logik pada konfigurasi yang diinginkan untuk membentuk operasi aritmatik dan logik pada data tertentu
- *Hardwired program* tidak flexibel
- *General purpose hardware* dapat mengerjakan berbagai macam tugas tergantung sinyal kendali yang diberikan
- Daripada melakukan *re-wiring*, Lebih baik menambahkan sinyal-sinyal kendali yang baru

Program ?

- Adalah suatu deretan langkah-langkah
- Pada setiap langkah, dikerjakan suatu operasi arithmetic atau logical
- Pada setiap operasi, diperlukan sejumlah sinyal kendali tertentu

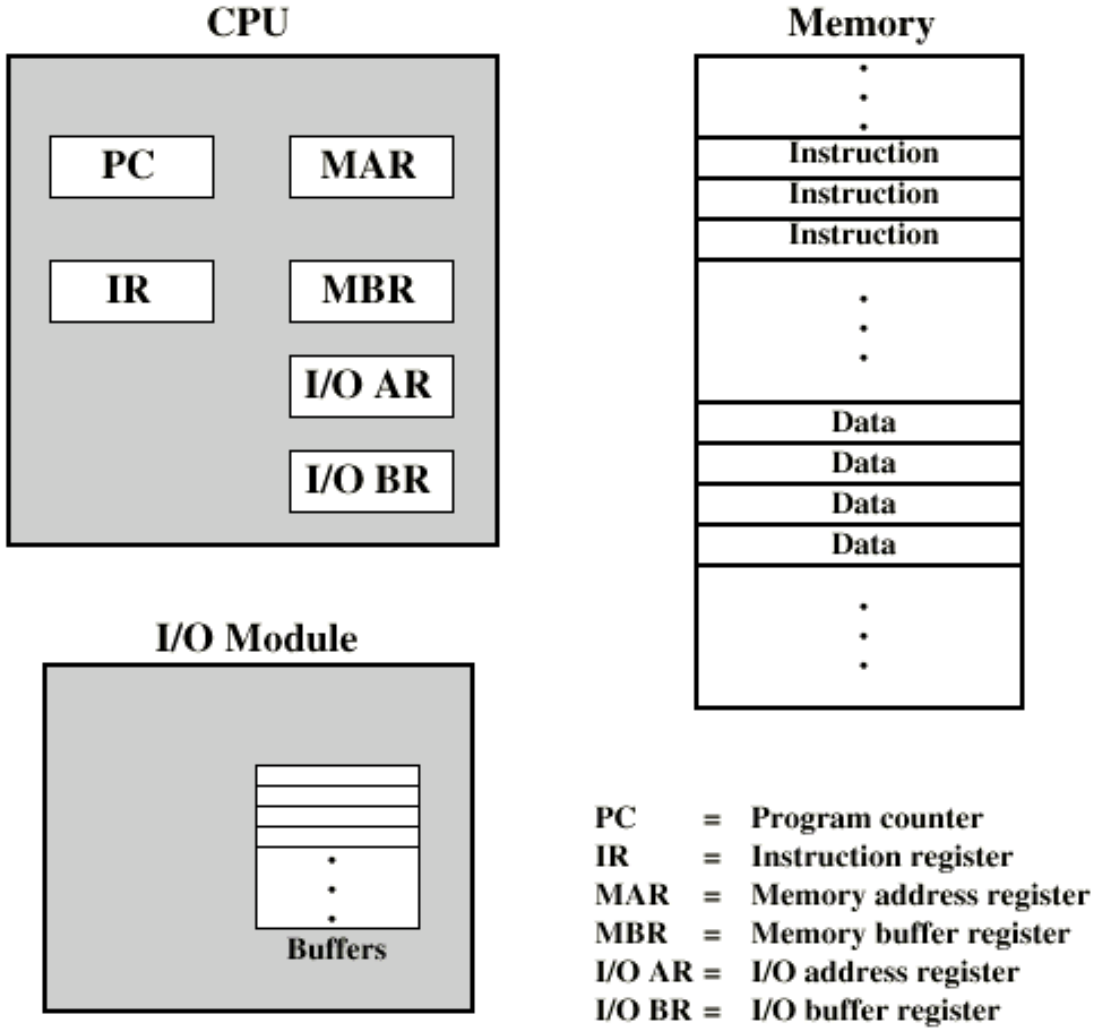
Fungsi Control Unit

- Untuk setiap operasi disediakan kode yang unik
  - Contoh: ADD, MOVE
- Bagian hardware tertentu menerima kode tersebut kemudian menghasilkan sinyal-sinyal kendali
- Jadilah komputer!

Komponen yang diperlukan

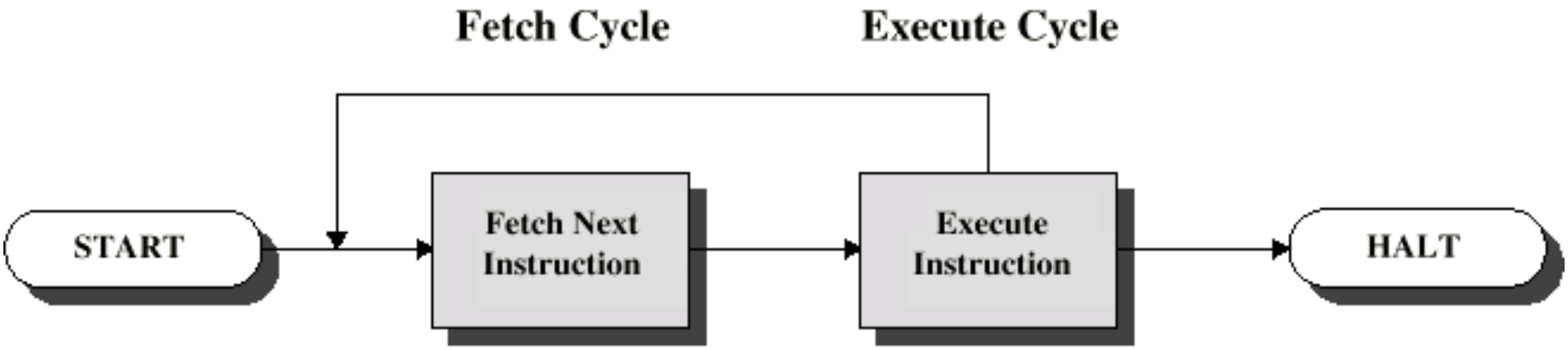
- Control Unit (CU) dan Arithmetic and Logic Unit (ALU) membentuk Central Processing Unit (CPU)
- Data dan instruksi harus diberikan ke sistem dan dikeluarkan dari sistem
  - Input/output
- Diperlukan tempat untuk menyimpan sementara kode instruksi dan hasil operasi.
  - Main memory

Komponen Komputer: Top Level View



Siklus Instruksi

- Two steps:
  - Fetch
  - Execute



Fetch Cycle

- Program Counter (PC) berisi address instruksi berikutnya yang akan diambil
- Processor mengambil instruksi dari memory pada lokasi yang ditunjuk oleh PC
- Naikkan PC
  - Kecuali ada perintah tertentu
- Instruksi dimasukkan ke Instruction Register (IR)
- Processor meng-interpret dan melakukan tindakan yang diperlukan

Execute Cycle

- Processor-memory
  - Transfer data antara CPU dengan main memory
- Processor I/O
  - Transfer data antara CPU dengan I/O module
- Data processing
  - Operasi arithmetic dan logical pada data tertentu
- Control
  - Mengubah urutan operasi
  - Contoh: jump
- Kombinasi diatas

Contoh Eksekusi Program

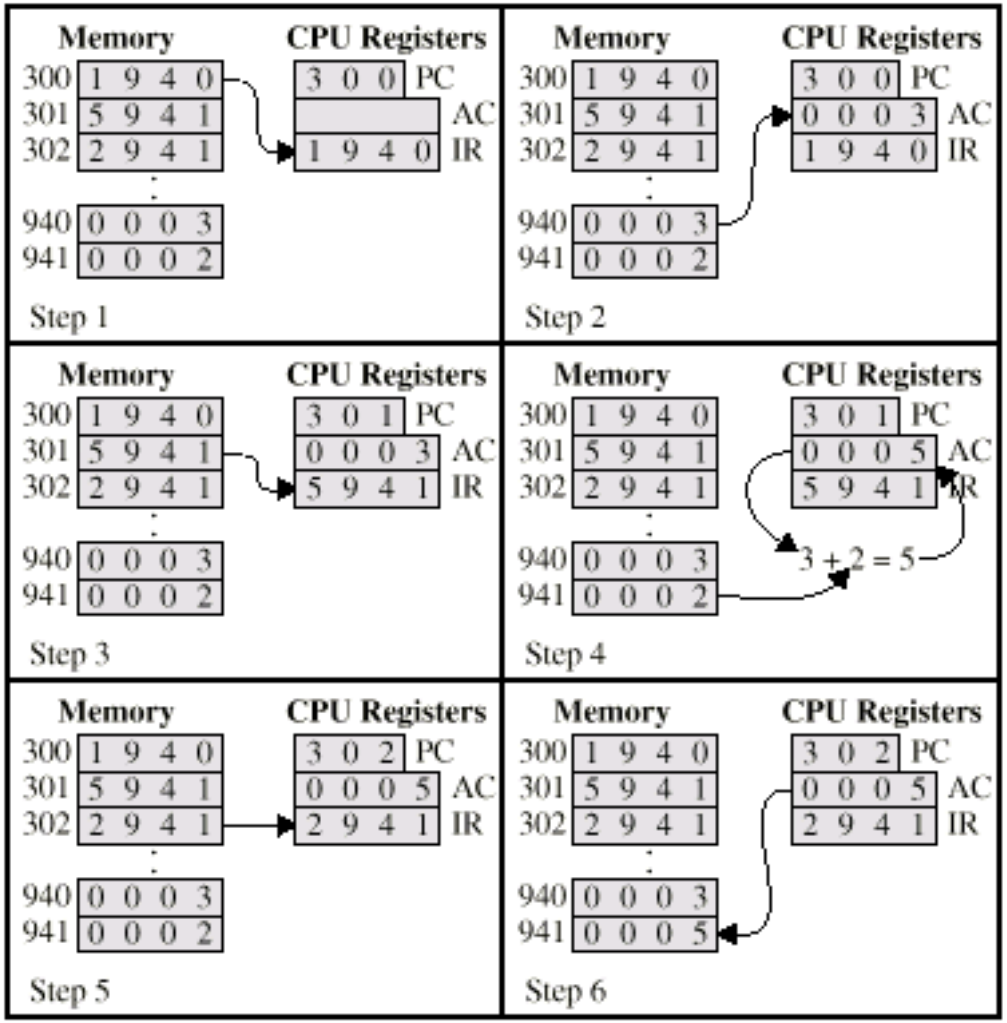
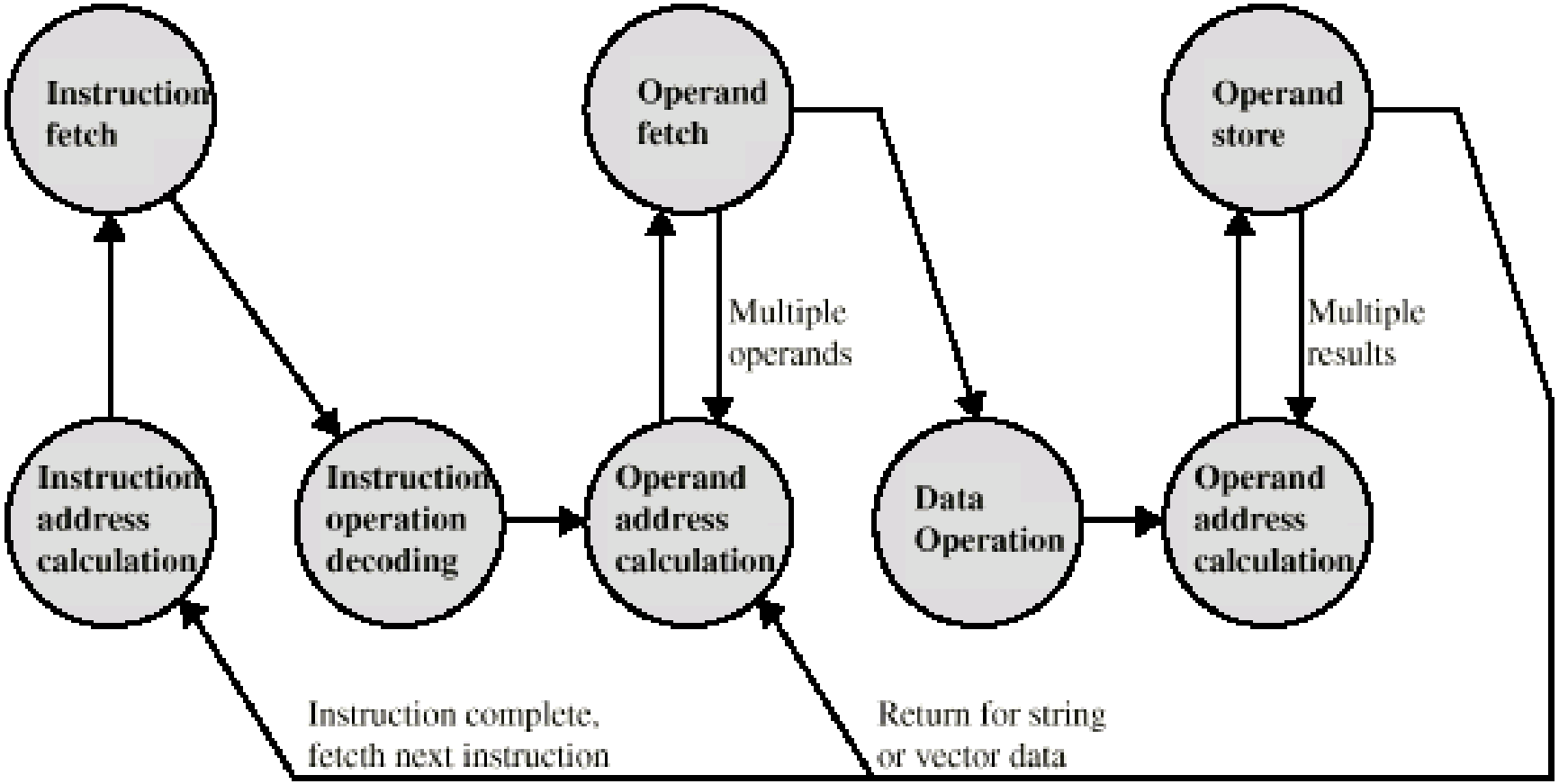


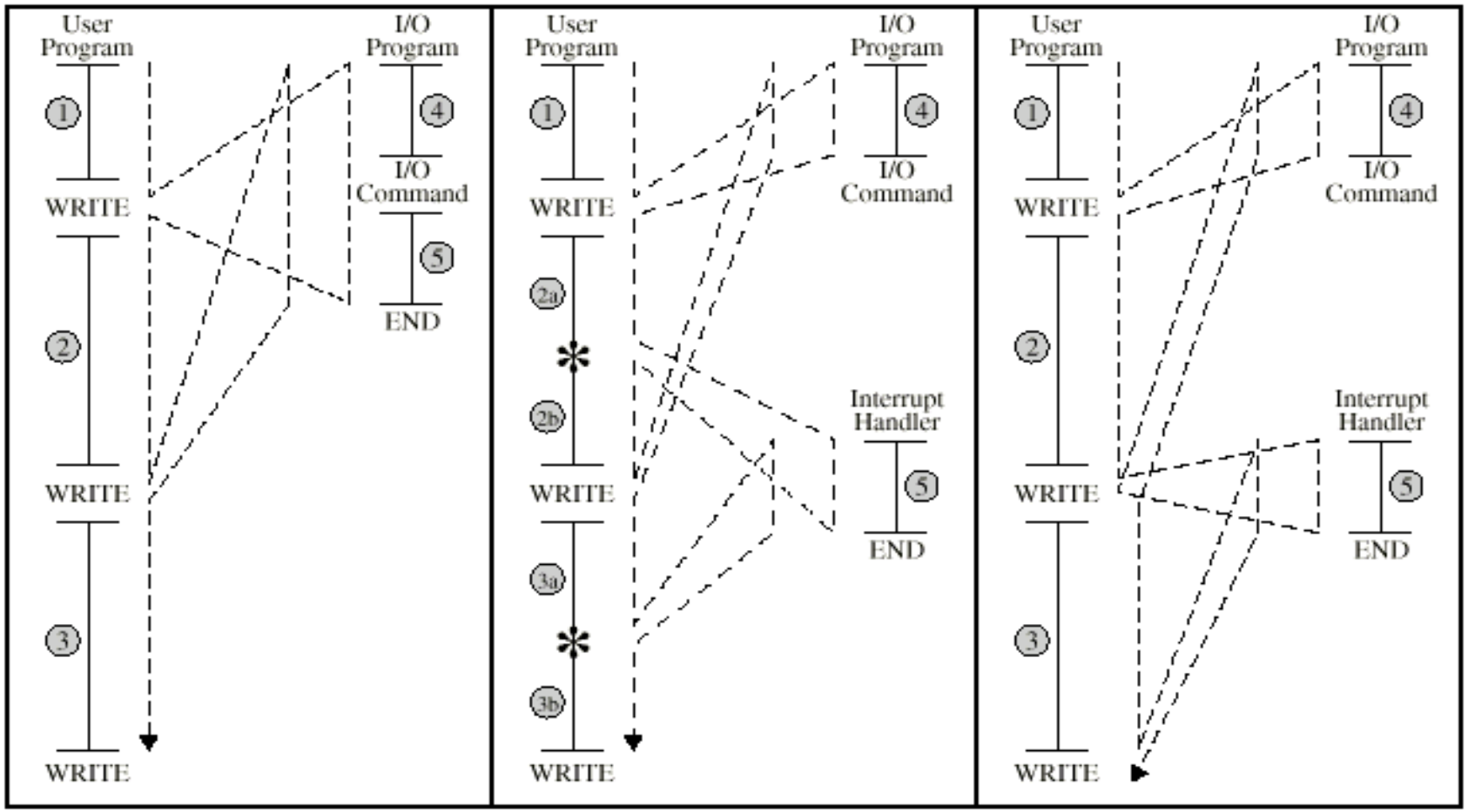
Diagram Keadaan Siklus Instruksi



Interrupt

- Suatu mekanisme yang disediakan bagi modul-modul lain (mis. I/O) untuk dapat meng-interupsi operasi normal CPU
- Program
  - Misal: overflow, division by zero
- Timer
  - Dihasilkan oleh internal processor timer
  - Digunakan dalam pre-emptive multi-tasking
- I/O
  - dari I/O controller
- Hardware failure
  - Misal: memory parity error

Program Flow Control



(a) No interrupts

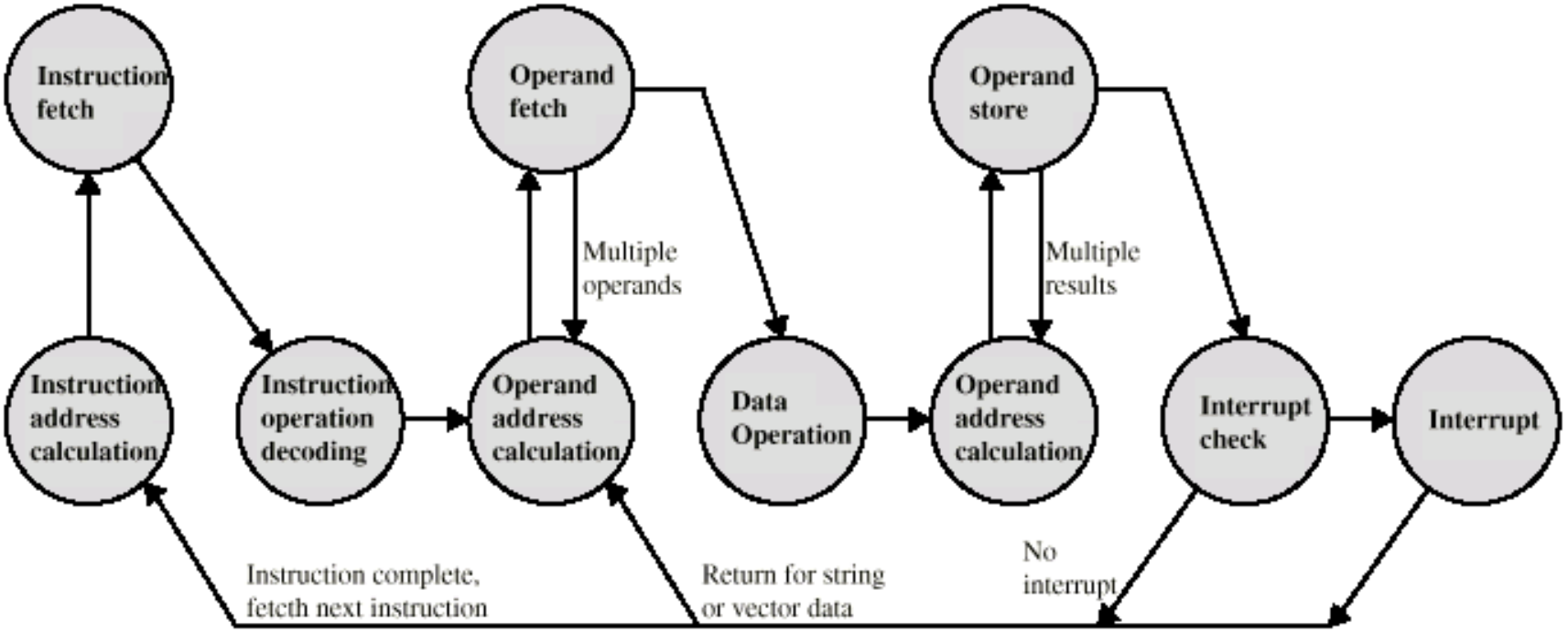
(b) Interrupts; short I/O wait

(c) Interrupts; long I/O wait

Siklus Interupsi

- Ditambahkan ke instruction cycle
- Processor memeriksa adanya interrupt
  - Diberitahukan lewat interrupt signal
- Jika tidak ada interrupt, fetch next instruction
- Jika ada interrupt:
  - Tunda eksekusi dari program saat itu
  - Simpan *context*
  - Set PC ke awal address dari routine interrupt handler
  - Proses interrupt
  - Kembalikan *context* dan lanjutkan program yang terhenti.

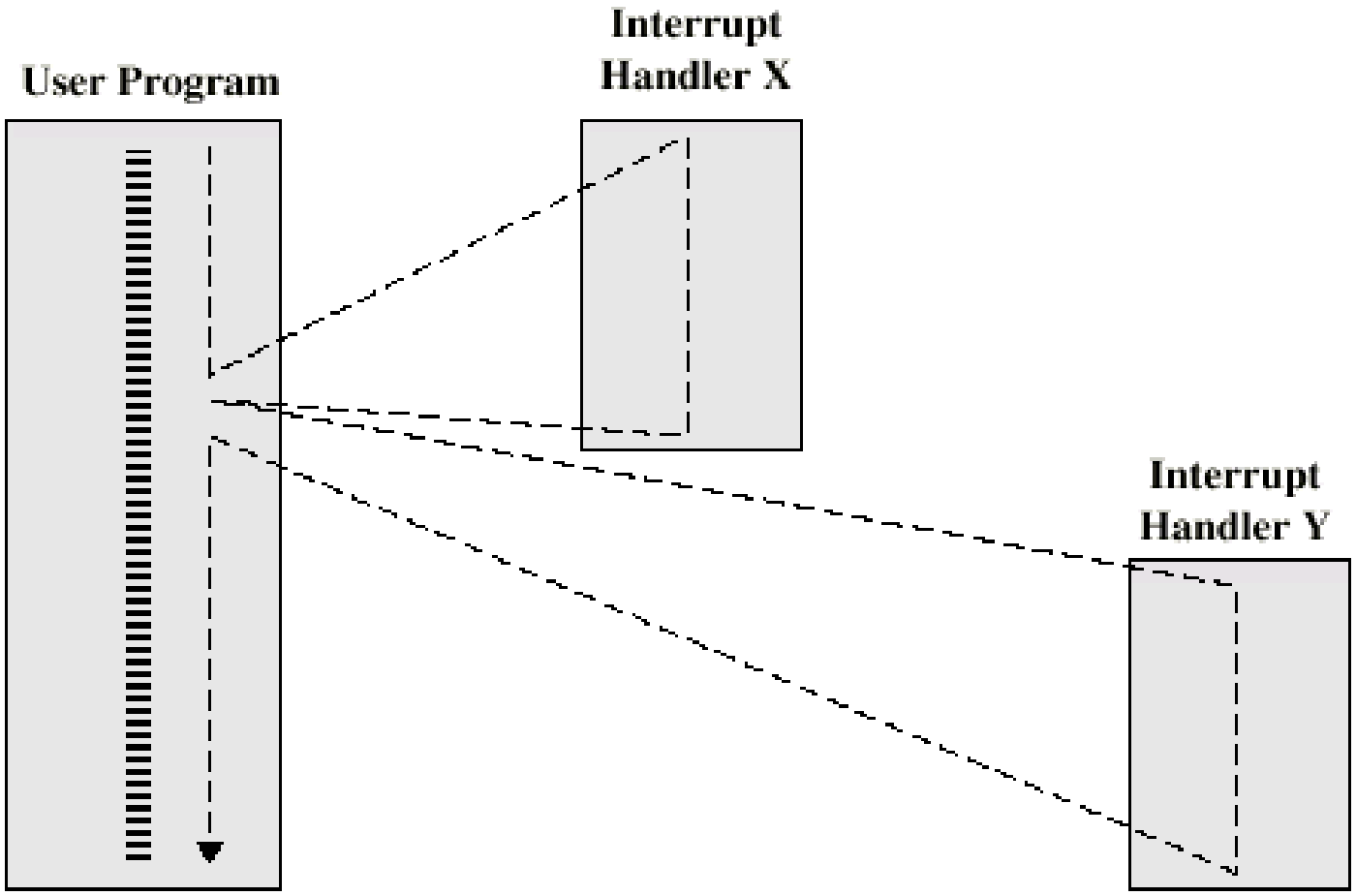
Diagram keadaan Siklus Instruksi dengan Interrupt



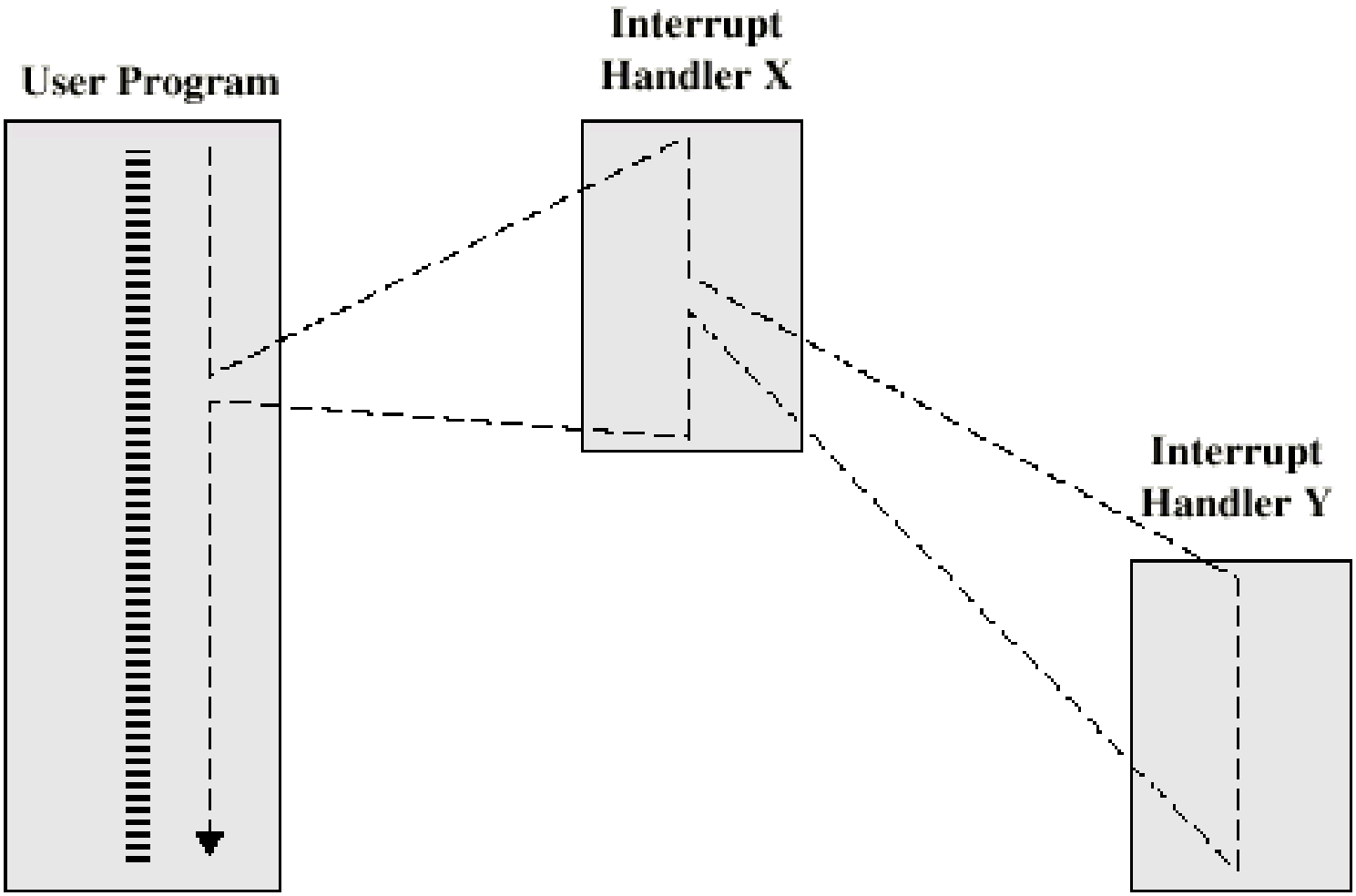
Multiple Interrupts

- Disable interrupts
  - Processor akan mengabaikan interrupt berikutnya
  - Interrupts tetap akan diperiksa setelah interrupt ynag pertama selesai dilayani
  - Interrupts ditangani dalam urutan sesuai datangnya
- Define priorities
  - Low priority interrupts dapat di interrupt oleh higher priority interrupts
  - Setelah higher priority interrupt selesai dilayani, akan kembali ke interrupt sebelumnya.

Multiple Interrupts - Sequential



Multiple Interrupts - Nested



Sambungan

- Semua unit harus tersambung
- Unit yang beda memiliki sambungan yang beda
  - Memory
  - Input/Output
  - CPU

Sambungan Memori

- Menerima dan mengirim data
- Menerima addresses
- Menerima sinyal kendali
  - Read
  - Write
  - Timing

Sambungan Input/Output

- Serupa dengan sambungan memori
- Output
  - Menerima data dari computer
  - Mengirimkan data ke peripheral
- Input
  - Menerima data dari peripheral
  - Mengirimkan data ke computer

Sambungan Input/Output

- Menerima sinyal kendali dari computer
- Mengirimkan sinyal kendali ke peripherals
  - Contoh: spin disk
- Menerima address dari computer
  - Contoh: nomor port
- Mengirimkan sinyal interrupt

CPU Connection

- Membaca instruksi dan data
- Menuliskan data (setelah diproses)
- Mengirimkan sinyal kendali ke unit-unit lain
- Menerima (& menanggapi) interrupt

Bus

- Ada beberapa kemungkinan interkoneksi sistem
- Yang biasa dipakai: Single Bus dan multiple BUS
- PC: Control/Address/Data bus
- DEC-PDP: Unibus

What is a Bus?

- Jalur komunikasi yang menghubungkan beberapa device
- Biasanya menggunakan cara broadcast
- Seringkali dikelompokkan
  - Satu bus berisi sejumlah kanal (jalur)
  - Contoh bus data 32-bit berisi 32 jalur
- Jalur sumber tegangan biasanya tidak diperlihatkan

Data Bus

- Membawa data
  - Tidak dibedakan antara “data” dan “instruksi”
- Lebar jalur menentukan performance
  - 8, 16, 32, 64 bit

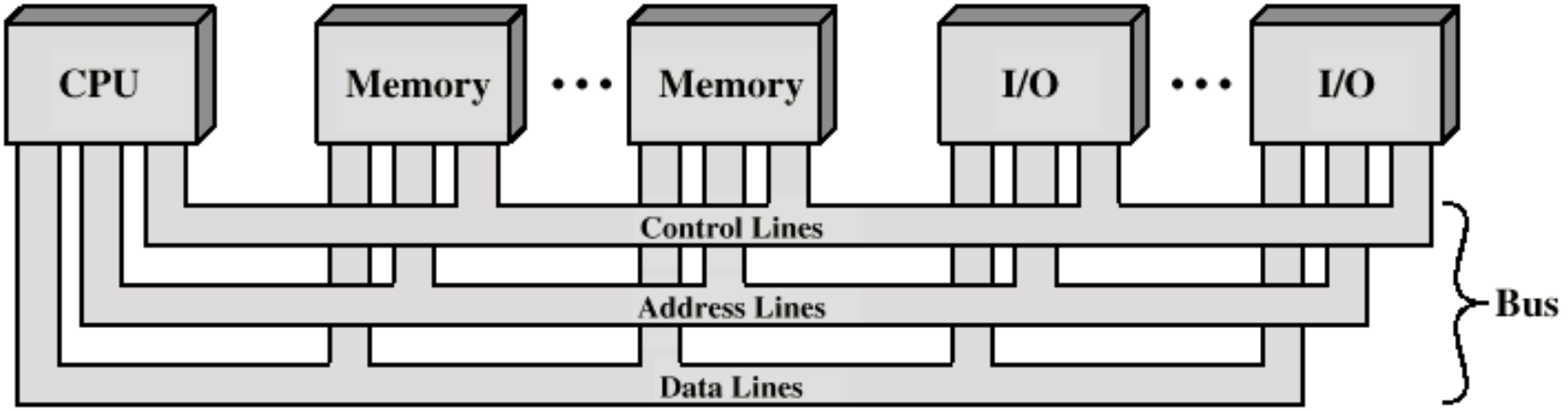
Address bus

- Menentukan asal atau tujuan dari data
- Misalkan CPU perlu membaca instruksi (data) dari memori pada lokasi tertentu
- Lebar jalur menentukan kapasitas memori maksimum dari sistem
  - Contoh 8080 memiliki 16 bit address bus maka ruang memori maksimum adalah 64k

Control Bus

- Informasi kendali dan timing
  - Sinyal read/write memory (MRD/MWR)
  - Interrupt request (IRQ)
  - Clock signals (CK)

Skema Interkoneksi Bus



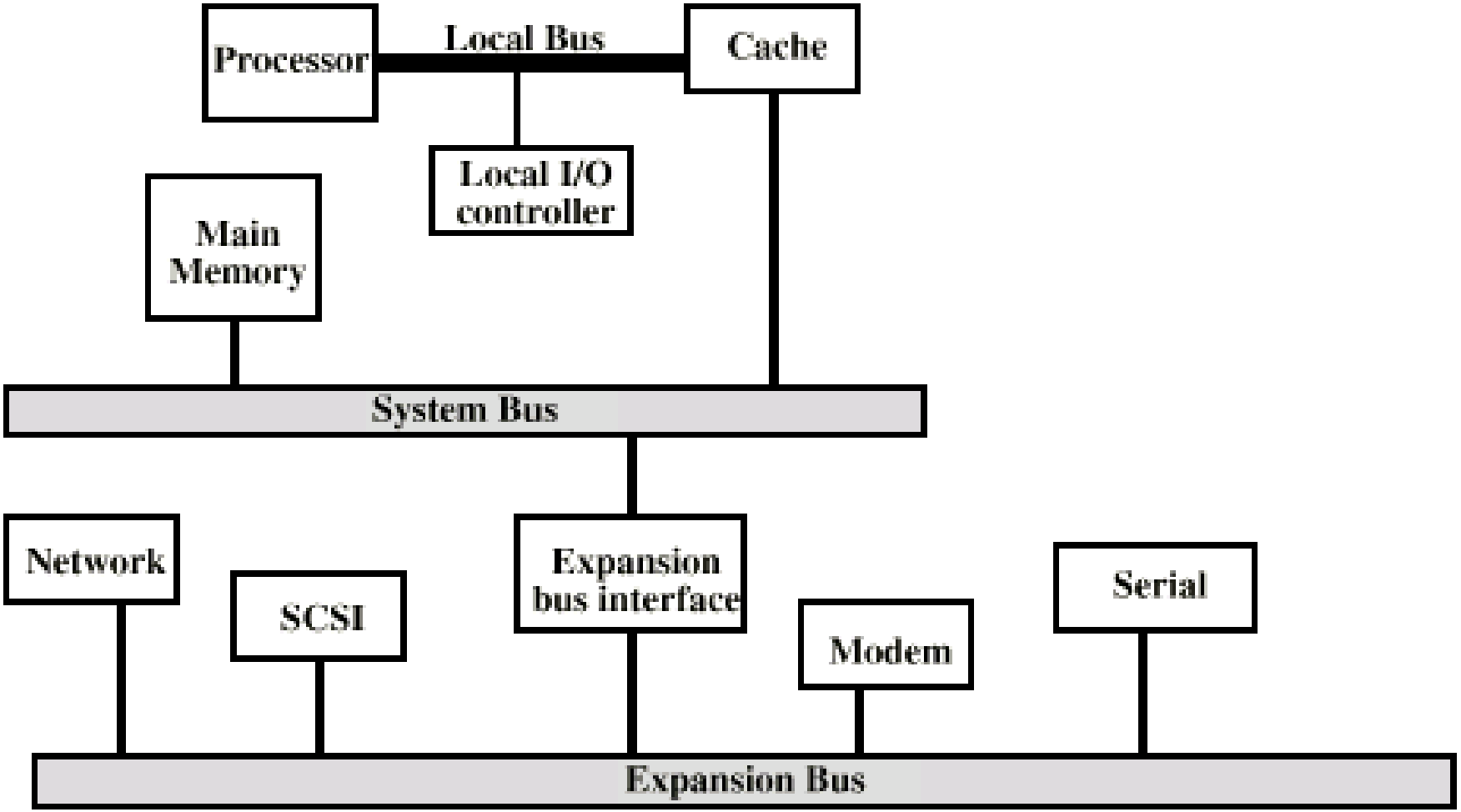
Bentuk Fisik

- Bagaimana bentuk fisik bus?
  - Jalur-jalur parallel PCB
  - Ribbon cables
  - Strip connectors pada mother boards
    - contoh PCI
  - Kumpulan kabel

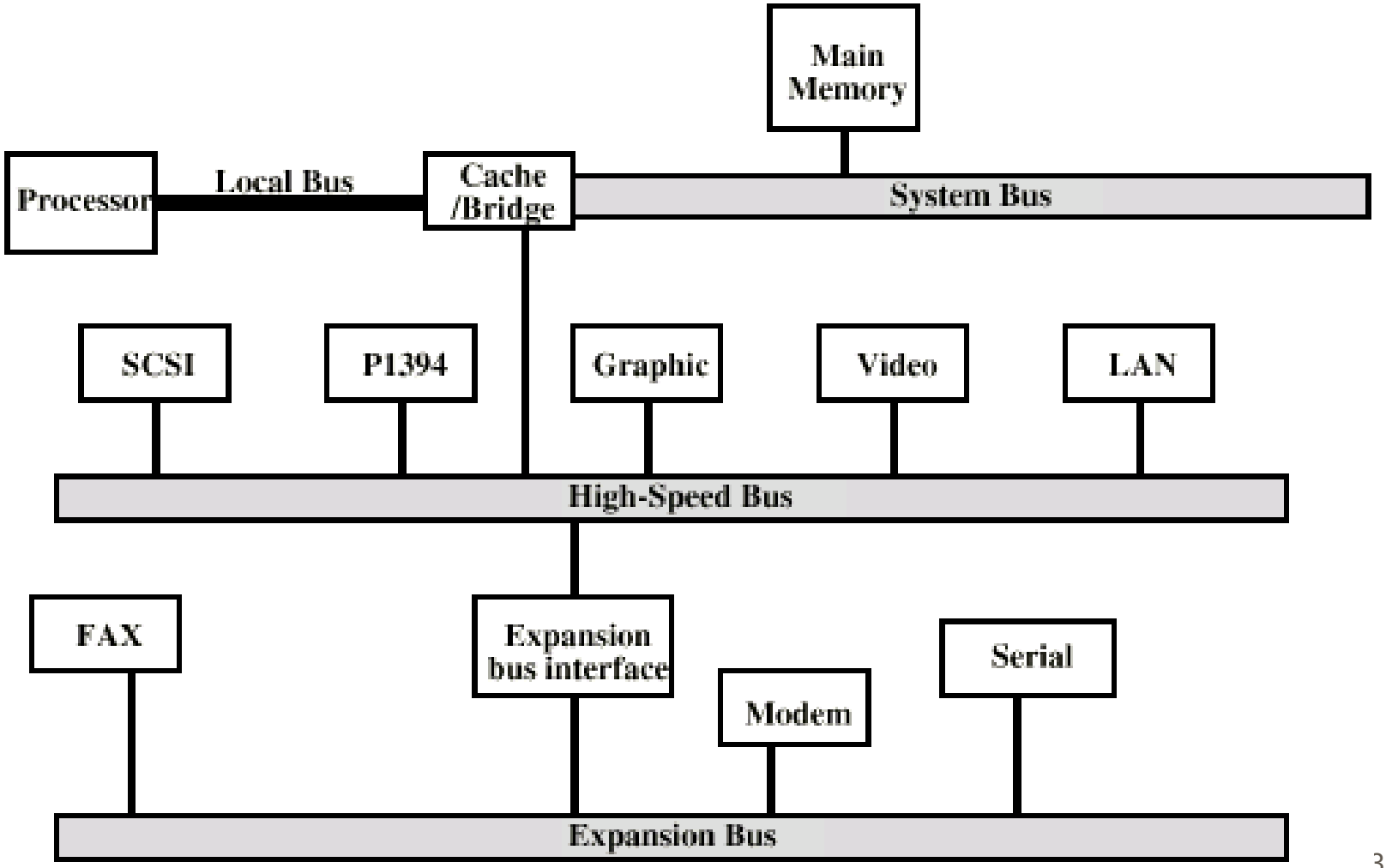
Problem pada Single Bus

- Banyak devices pada bus tunggal menyebabkan:
  - Propagation delays
    - Jalur data yg panjang berarti memerlukan koordinasi pemkaian shg berpengaruh pada performance
    - If aggregate data transfer approaches bus capacity
- Kebanyakan sistem menggunakan multiple bus

Bus Traditional (ISA) (menggunakan cache)



High Performance Bus



Jenis Bus

- Dedicated
  - Jalur data & address terpisah
- Multiplexed
  - Jalur bersama
  - Address dan data pada saat yg beda
  - Keuntungan – jalur sedikit
  - Kerugian
    - Kendali lebih komplek
    - Mempengaruhi performance

Arbitrasi Bus

- Beberapa modul mengendalikan bus
- contoh CPU dan DMA controller
- Setiap saat hanya satu modul yg mengendalikan
- Arbitrasi bisa secara centralised atau distributed

Arbitrasi Centralised

- Ada satu hardware device yg mengendalikan akses bus
  - Bus Controller
  - Arbitrer
- Bisa berupa bagian dari CPU atau terpisah

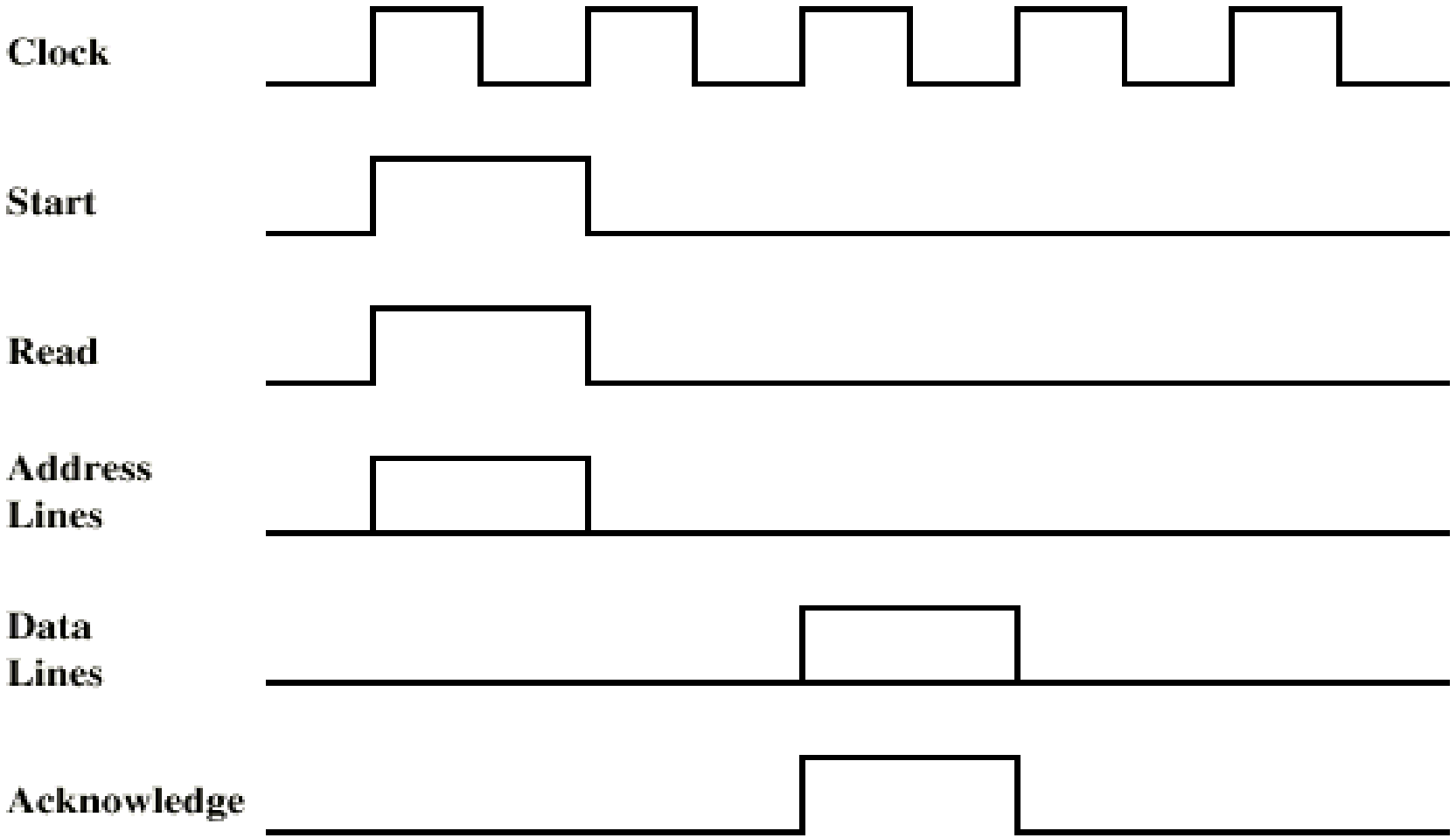
Arbitrasi Distributed

- Setiap module dapat meng-klaim bus
- Setiap modules memiliki Control logic

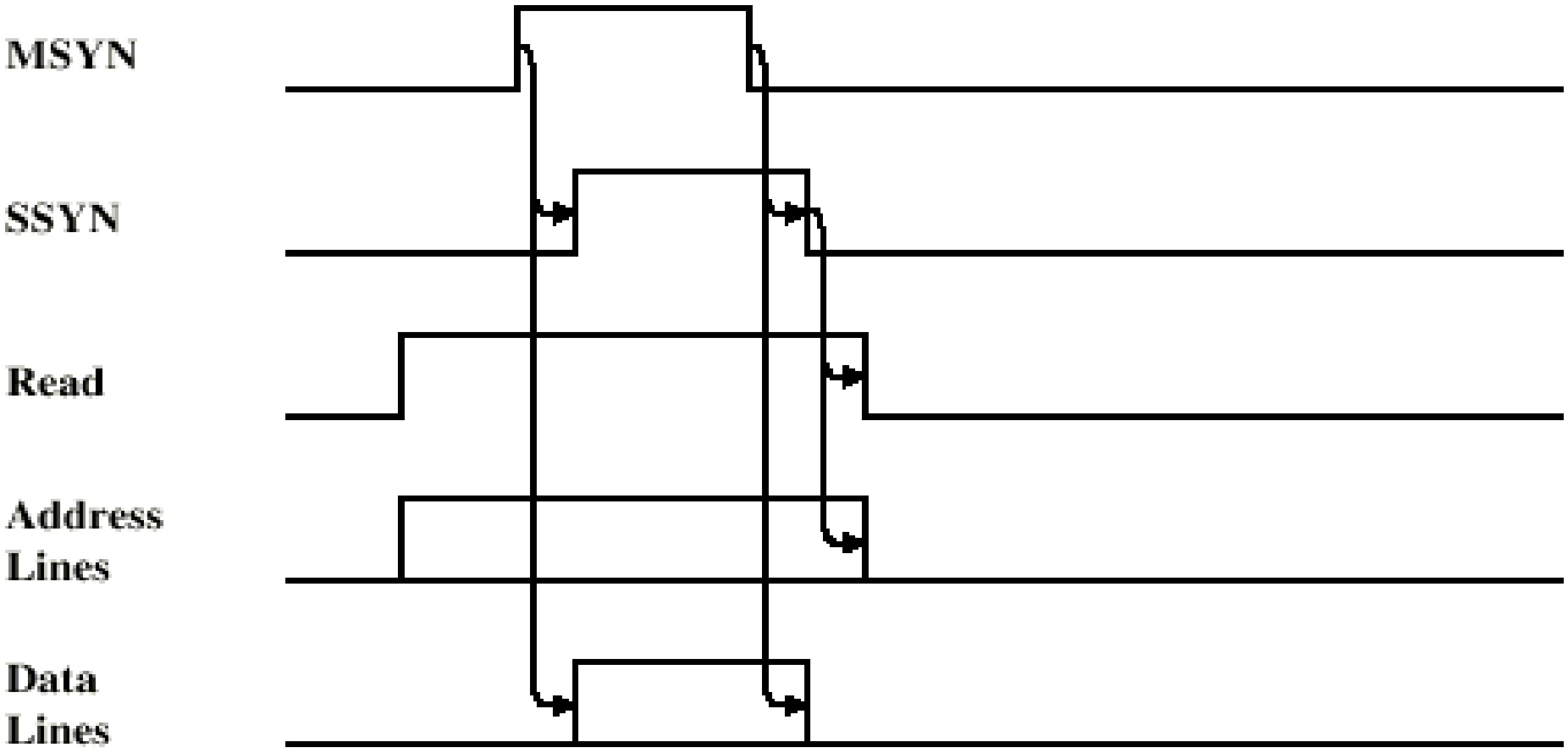
Timing

- Koordinasi event pada bus
- Synchronous
  - Event ditentukan oleh sinyal clock
  - Control Bus termasuk jalur clock
  - Siklus bus (bus cycle) transmisi 1 ke 0
  - Semua devices dpt membaca jakur clock
  - Biasanya sinkronisasi terjadi pada tepi naik (leading edge)
  - Suatu event biasanya dimualai pada awal siklus

Synchronous Timing Diagram



Asynchronous Timing Diagram



Bus PCI

- Peripheral Component Interconnection
- Dikeluarkan oleh Intel sebagai public domain
- 32 atau 64 bit
- 50 Jalur

Jalur pada Bus PCI (yg harus)

- Jalur System
  - clock and reset
- Address & Data
  - 32 jalur multiplex address/data
  - Jalur validasi
- Interface Control
- Arbitrasi
  - Not shared
  - Direct connection to PCI bus arbiter
- Error lines

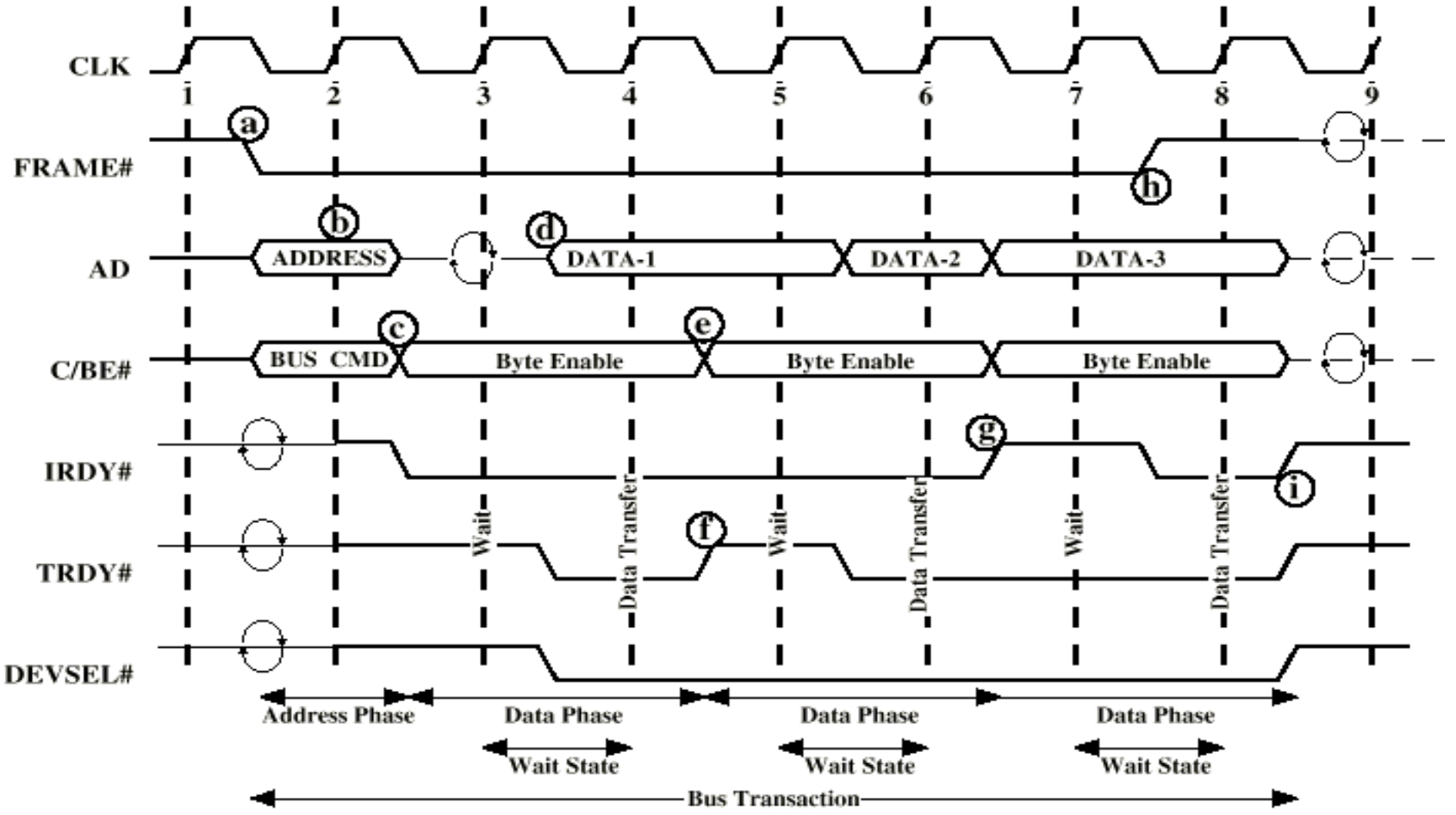
Jalur Bus PCI (Optional)

- Interrupt lines
  - Not shared
- Cache support
- 64-bit Bus Extension
  - Additional 32 lines
  - Time multiplexed
  - 2 lines to enable devices to agree to use 64-bit transfer
- JTAG/Boundary Scan
  - For testing procedures

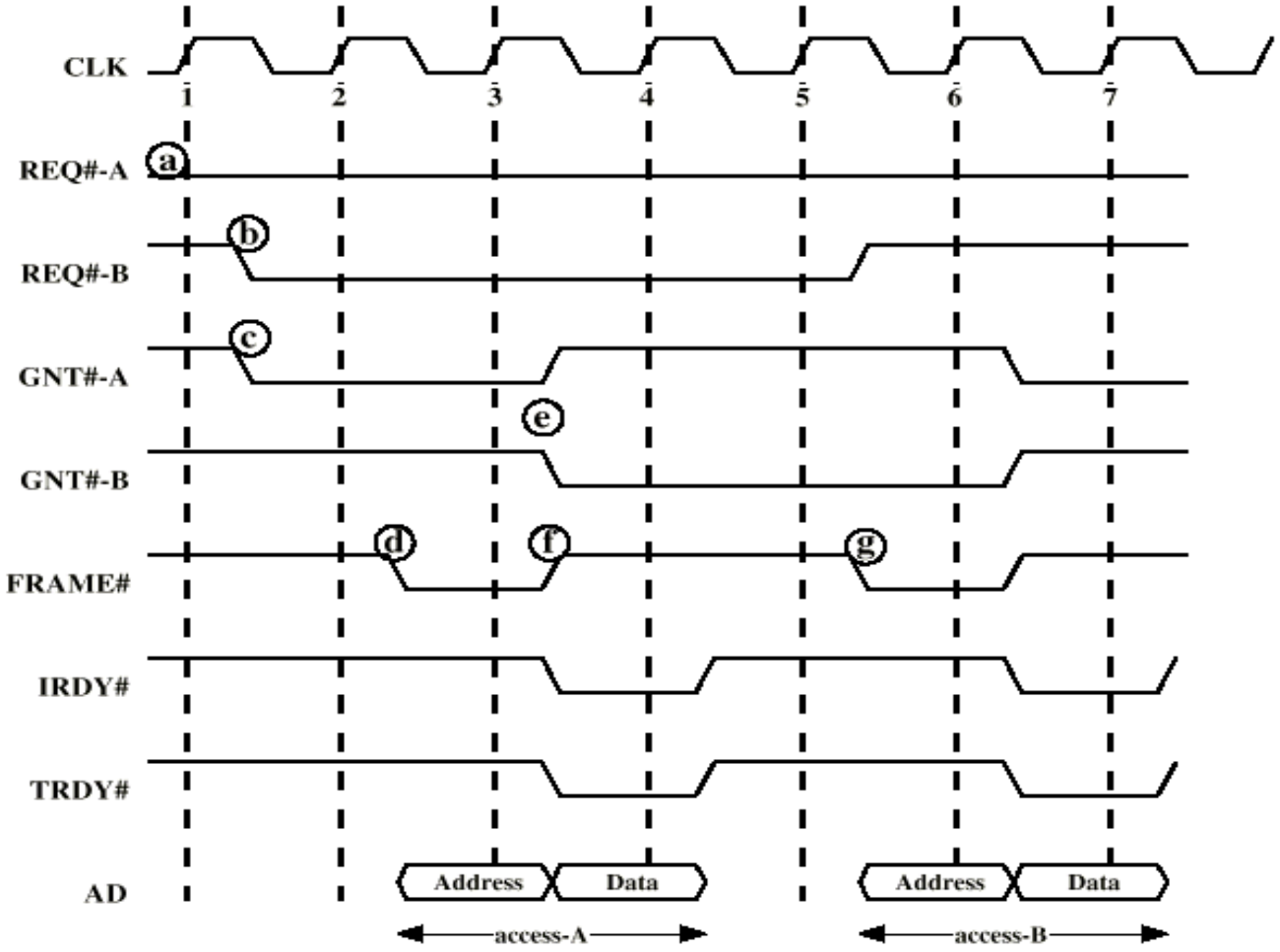
Command pada PCI

- Transaksi antara initiator (master) dg target
- Master pegang kendali bus
- Master menentukan jenis transaksi
  - Misal I/O read/write
- Fase Address
- Fase Data

PCI Read Timing Diagram



PCI Bus Arbitration



Internet Resource

- www.pcguide.com/ref/mbsys/buses/
- www.pcguide.com/