

# PERTEMUAN

# 14

## Control Unit Operation

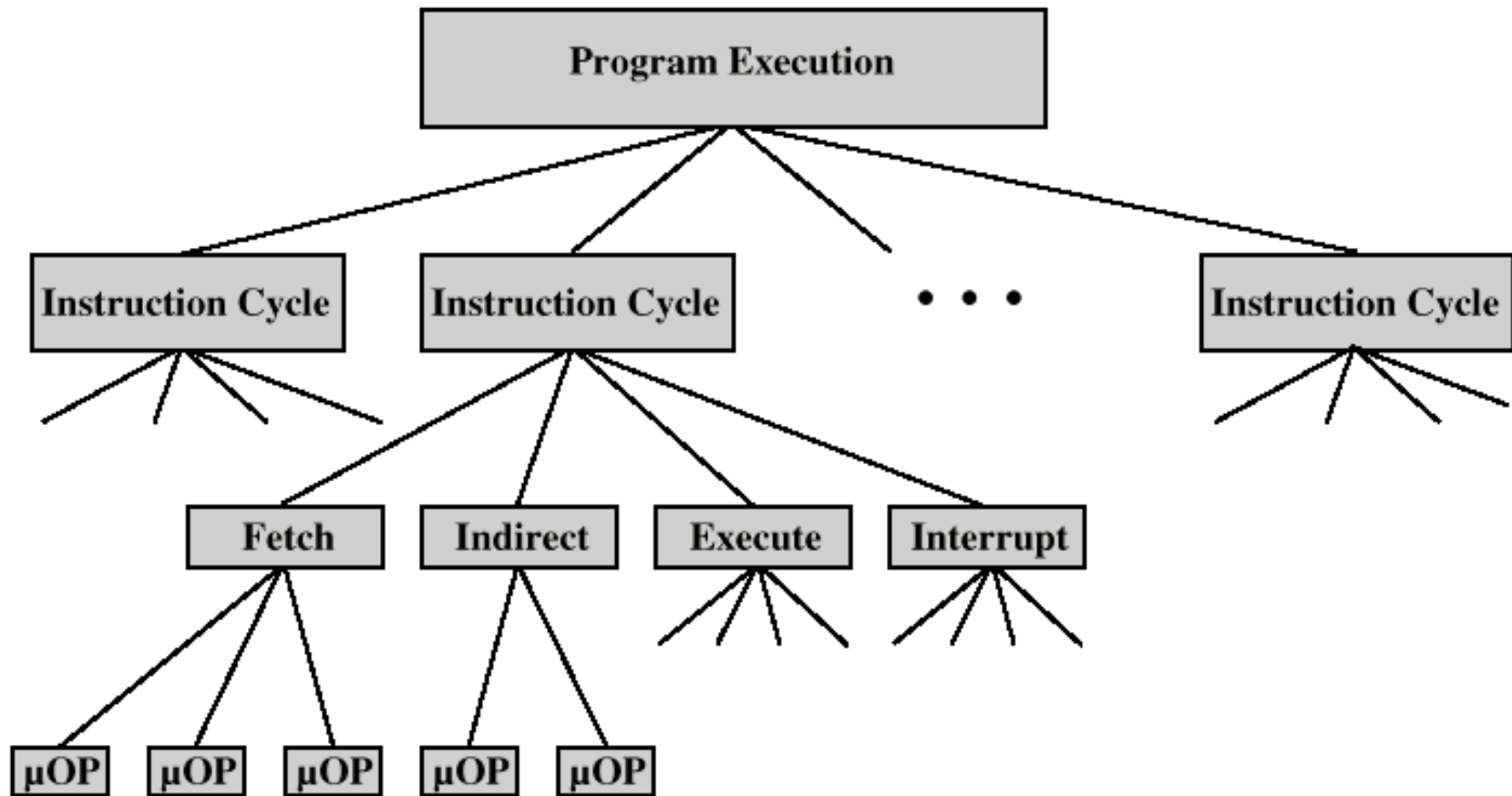
# Micro-Operations

Micro operation merupakan operasi atomic dari CPU (Atomic operation of CPU). Micro operation adalah kerja atau eksekusi terhadap data yang tersimpan pada register dan merupakan cara kerjanya dalam satu pulsa clock. Atau pengertian lainnya micro operation adalah suatu operasi mikro dimana suatu computer menjalankan suatu program dan melakukan siklus proses memasukkan dan mengambil data atau melakukan eksekusi (Fetch/execute cycle)

# Micro-Operations

- Digunakan Komputer untuk menjalankan program
- Fetch/execute cycle
- Memiliki beberapa langkah dalam setiap siklus/cycle
  - Pipelining
- Pemanggilan micro-operations
- Waktu dan langkah yang singkat
- Atomic operation CPU

# Elemen Konstituen dari Eksekusi Program



# Fetch - 4 Registers

- Memory Address Register (MAR)
  - Terhubung ke bus alamat
  - Menentukan alamat untuk operasi baca/tulis
- Memory Buffer Register (MBR)
  - Terhubung ke bus data
  - Menyimpan data untuk ditulis atau data terakhir terbaca
- Program Counter (PC)
  - Menyimpan alamat instruksi berikutnya yang akan diambil
- Instruction Register (IR)
  - Menandai instruksi terakhir yang diambil

# Fetch Sequence

- Alamat instruksi dan selanjutnya ada di PC
- Alamat (MAR) ditempatkan di bus alamat
- Unit kontrol mengeluarkan perintah READ
- Hasil (data dari memori) muncul di bus data
- Data dari bus data disalin ke MBR
- Jika pada PC instruksi bertambah 1 (sejajar dengan pengambilan data dari memori)
- Data (instruksi) dipindahkan dari MBR ke IR
- MBR sekarang bebas untuk pengambilan data lebih lanjut

# Fetch Sequence (simbol)

tx1:  $MAR \leftarrow (PC)$

tx2:  $MBR \leftarrow (\text{memory})$

$PC \leftarrow (PC) + 1$

tx3:  $IR \leftarrow (MBR)$

(tx = time unit/clock cycle)

Atau

tx1:  $MAR \leftarrow (PC)$

tx2:  $MBR \leftarrow (\text{memory})$

tx3:  $PC \leftarrow (PC) + 1$

$IR \leftarrow (MBR)$

# Rules pengelompokan Clock Cycle

- Sesuai urutan yang benar
  - $MAR \leftarrow (PC)$  mendahului  $MBR \leftarrow (\text{memory})$
- Masalah harus dihindari
  - Tidak boleh baca & tulis register yang sama pada waktu yang sama
  - $MBR \leftarrow (\text{memory})$  &  $IR \leftarrow (MBR)$  tidak boleh berada dalam siklus yang sama
- Perhatikan:  $PC \leftarrow (PC) + 1$  melibatkan penambahan
  - Menggunakan ALU
  - Mungkin perlu operasi mikro tambahan



# Indirect Cycle

- $MAR \leftarrow (IR_{\text{address}})$  - alamat bidang IR
- $MBR \leftarrow (\text{memory})$
- $IR_{\text{address}} \leftarrow (MBR_{\text{address}})$
- MBR berisi alamat
- IR dalam keadaan yang sama maka pengalamatan langsung telah digunakan

# Interrupt Cycle

t1: MBR  $\leftarrow$  (PC)

t2: MAR  $\leftarrow$  save-address

PC  $\leftarrow$  routine-address

t3: memory  $\leftarrow$  (MBR)

Hal minimal:

- Memerlukan operasi mikro tambahan untuk mendapatkan alamat
- menyimpan konteks dilakukan oleh penanganan interupsi routine, not micro-ops

# Execute Cycle (ADD)

- Berbeda untuk setiap instruksi  
e.g. ADD R1,X - tambahkan konten lokasi X to Register 1 , result in R1  
t1:  $MAR \leftarrow (IR_{address})$   
t2:  $MBR \leftarrow (memory)$   
t3:  $R1 \leftarrow R1 + (MBR)$
- Perhatikan, tidak ada operasi mikro yang tumpang tindih

# Execute Cycle (ISZ)

- ISZ X - naikan dan lewati jika nol
  - t1:  $MAR \leftarrow (IR_{address})$
  - t2:  $MBR \leftarrow (memory)$
  - t3:  $MBR \leftarrow (MBR) + 1$
  - t4:  $memory \leftarrow (MBR)$
  - if  $(MBR) == 0$  then  $PC \leftarrow (PC) + 1$
- Notes:
  - jika operasi mikro tunggal
  - Operasi mikro dilakukan selama t4

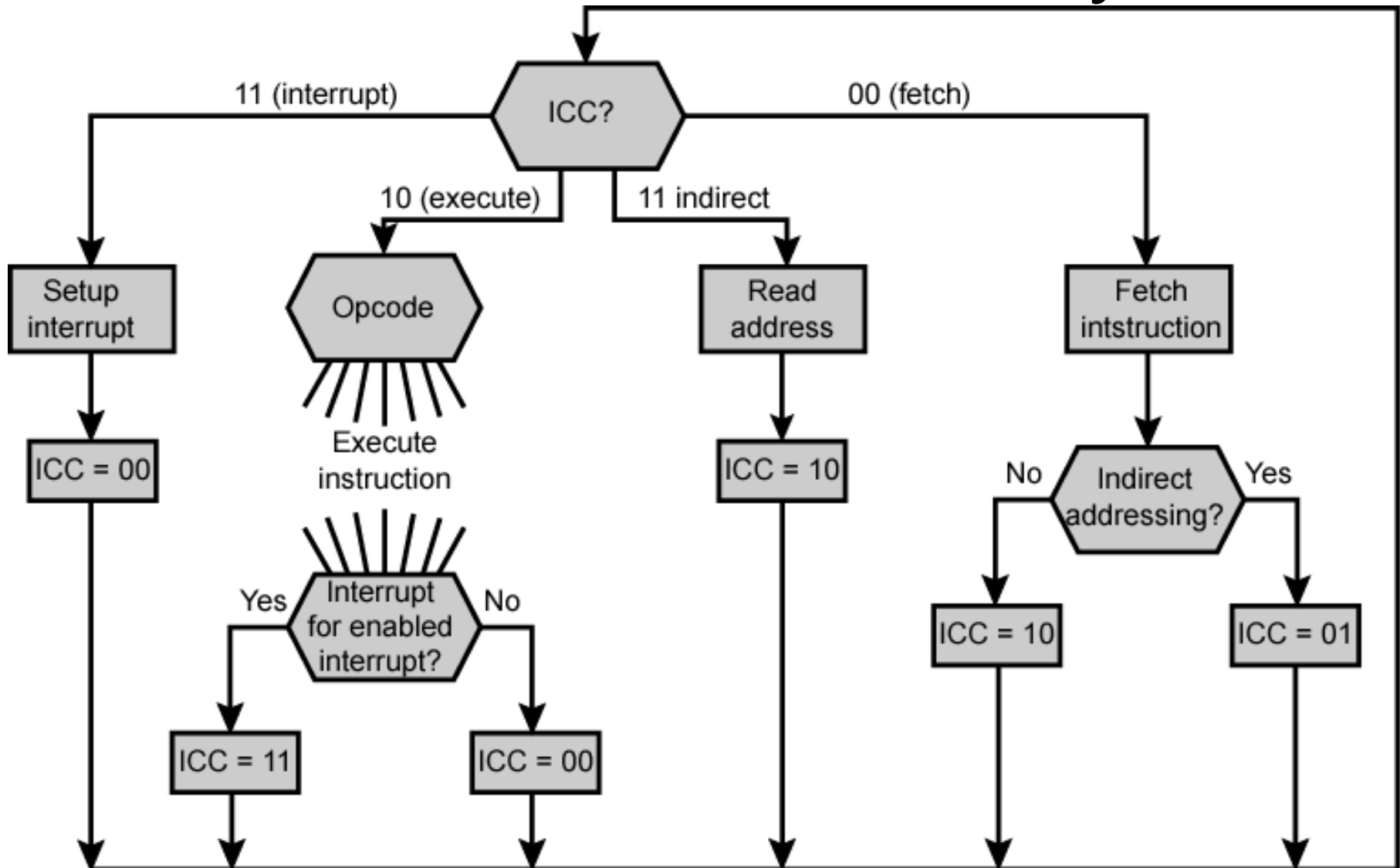
# Execute Cycle (BSA)

- **BSA X - Branch and Save Address**
    - Alamat instruksi setelah BSA disimpan di X
    - Eksekusi berlanjut dari X+1
- t1:     $MAR \leftarrow (IR_{address})$   
          $MBR \leftarrow (PC)$
- t2:     $PC \leftarrow (IR_{address})$   
          $memory \leftarrow (MBR)$
- t3:     $PC \leftarrow (PC) + 1$

# Instruction Cycle

- Setiap fase didekomposisi menjadi urutan operasi mikro dasar
  - E.g. fetch, indirect, and interrupt cycles
- Execute cycle
  - Satu urutan operasi mikro untuk setiap opcode
- Perlu untuk mengikat urutan bersama-sama
- Asumsikan register 2-bit baru
  - Kode siklus instruksi (ICC) menunjukkan bagian mana dari prosesor siklus yang berada di
    - 00: Fetch
    - 01: Indirect
    - 10: Execute
    - 11: Interrupt

# Flowchart Instruction Cycle



# Persyaratan Fungsional

- Tentukan elemen dasar prosesor
- Menjelaskan kinerja prosesor mikro-operasi
- Menentukan fungsi yang harus dilakukan unit kontrol control



# Basic Elements of Processor

- ALU
- Registers
- Internal data paths
- External data paths
- Control Unit

# Tipe Micro-operation

- Transfer data antar register register
- Transfer data dari register ke eksternal
- Transfer data dari eksternal ke register
- Lakukan operasi aritmatika atau logika

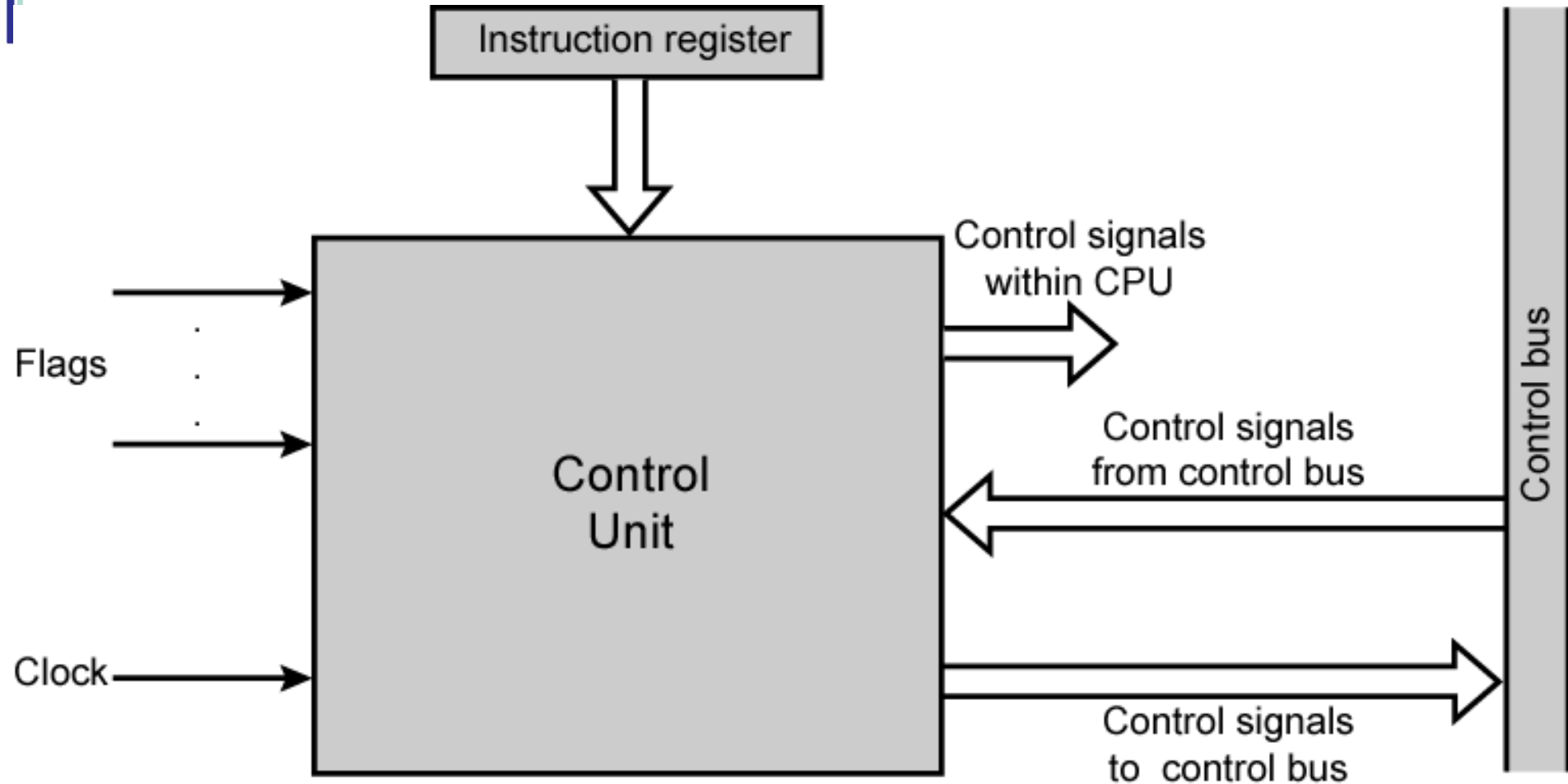
# Functions of Control Unit

- Sequential/pengurutan
  - Menyebabkan CPU melakukan serangkaian operasi mikro mikro
- Execution
  - Menyebabkan kinerja setiap operasi mikro
- Dilakukan dengan menggunakan Sinyal Kontrol

# Control Signals

- Clock
  - Satu instruksi mikro (atau serangkaian instruksi mikro paralel) per clock cycle
- Instruction register
  - Op-code untuk instruksi saat ini
  - Menentukan mikro-instruksi yang dilakukan
- Flags
  - Keadaan CPU
  - Hasil operasi sebelumnya
- Control bus
  - Interrupts
  - Acknowledgements

# Model of Control Unit



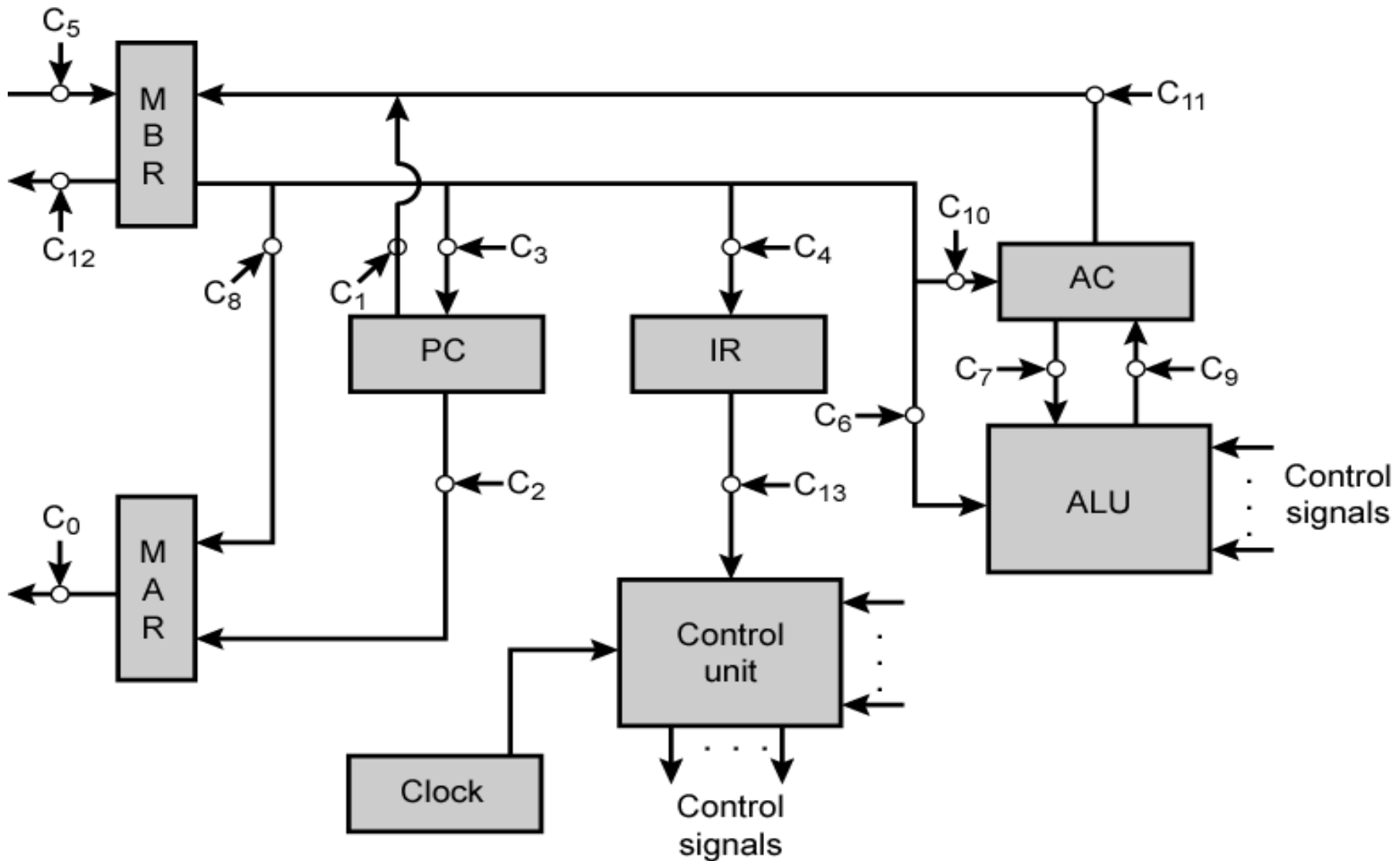
# Control Signals - output

- Dalam CPU
  - Penyebab perpindahan data data
  - Mengaktifkan fungsi tertentu
- Via control bus
  - untuk memory
  - untuk I/O modules

# Contoh Control Signal Sequence - Fetch

- MAR  $\leftarrow$  (PC)
  - Unit kontrol mengaktifkan sinyal untuk membuka gerbang antara PC dan MAR
- MBR  $\leftarrow$  (memory)
  - Buka gerbang antara MAR dan bus alamat
  - Sinyal kontrol pembacaan memori
  - Buka gerbang antara bus data dan MBR

# Data Paths and Control Signals

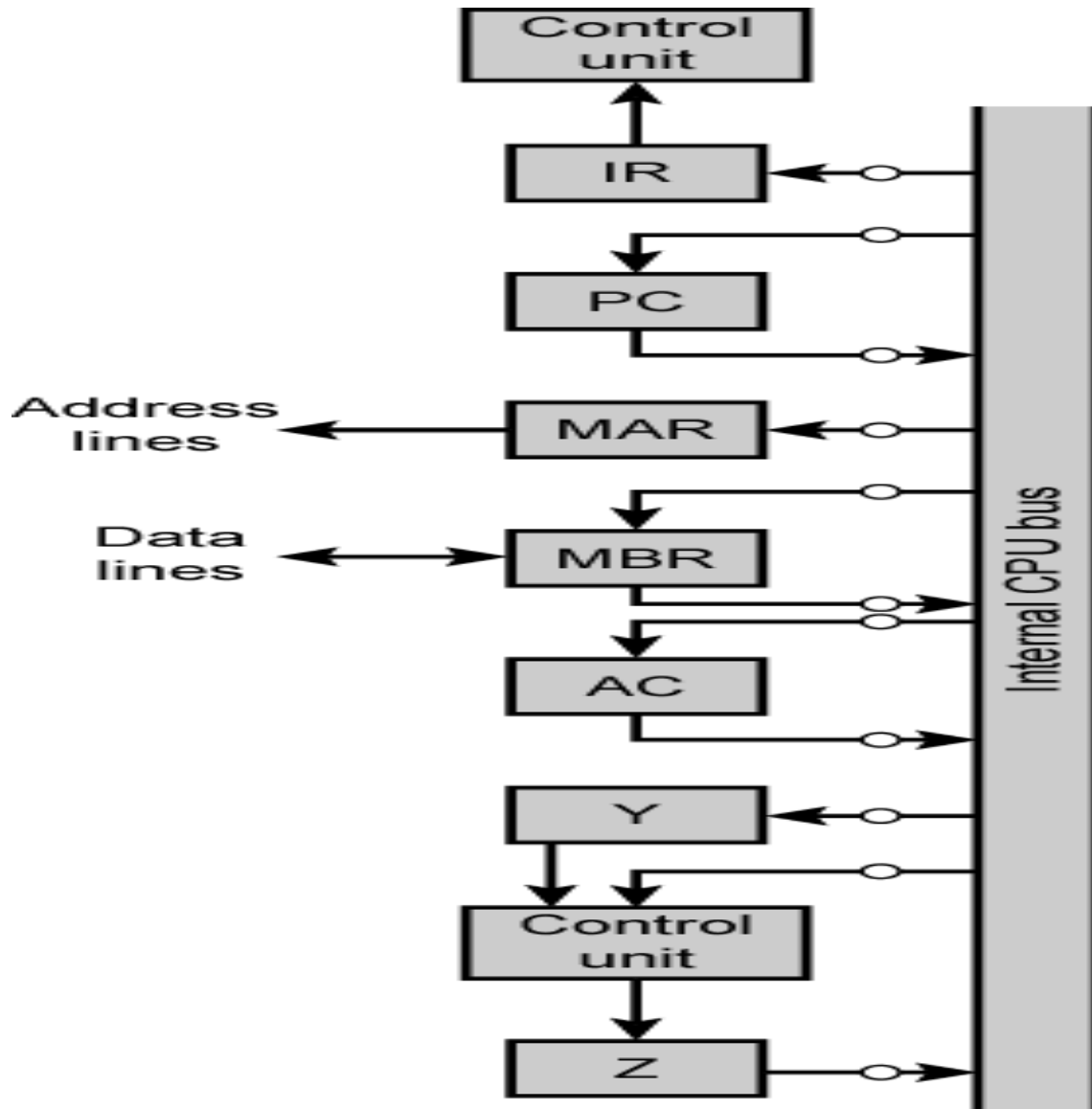




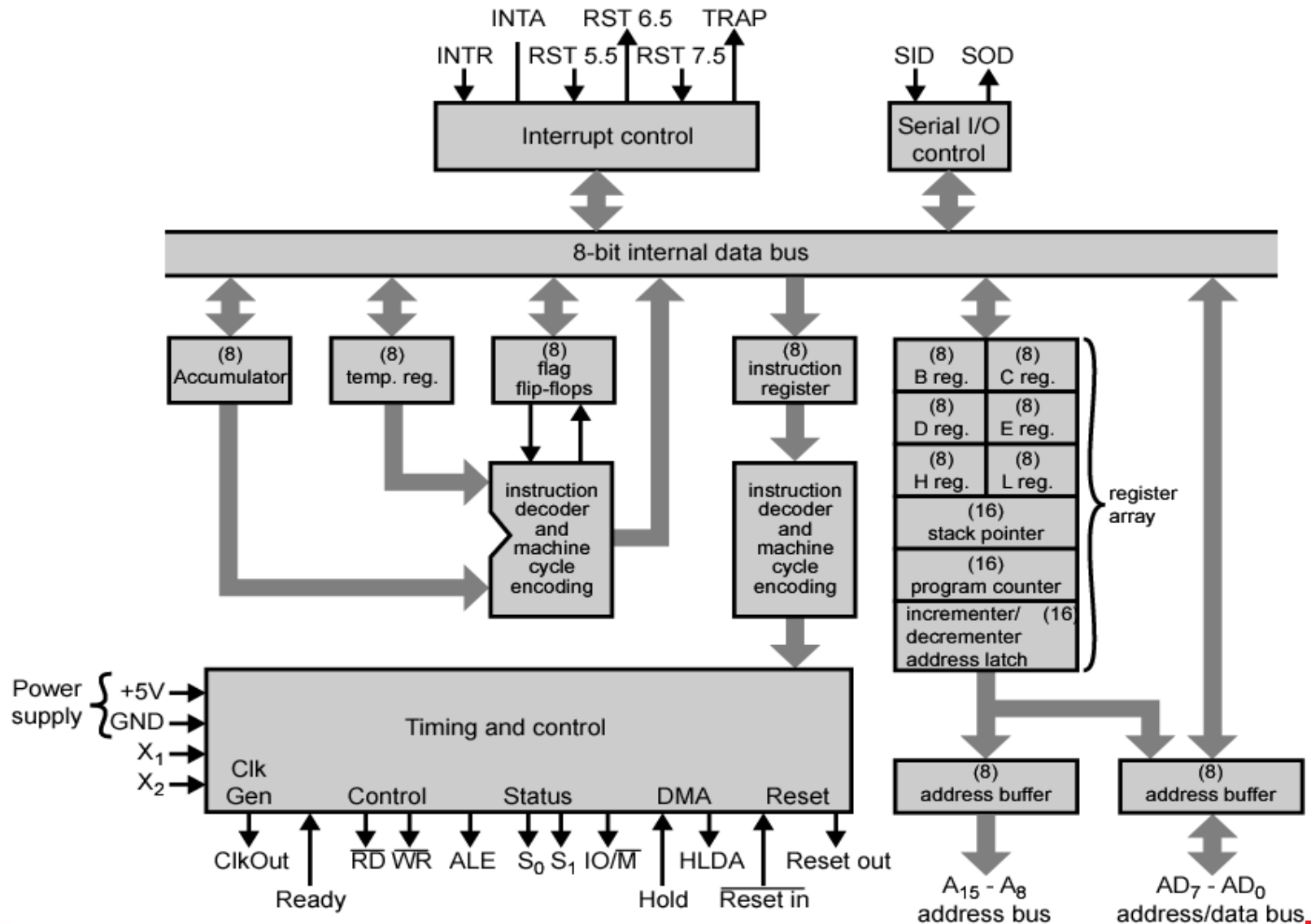
# Organisasi Internal

- Biasanya bus internal tunggal
- Gerbang mengontrol pergerakan data ke dalam dan ke luar bus
- Sinyal kontrol mengontrol transfer data ke dan dari bus sistem eksternal
- Register sementara diperlukan untuk pengoperasian ALU yang tepat

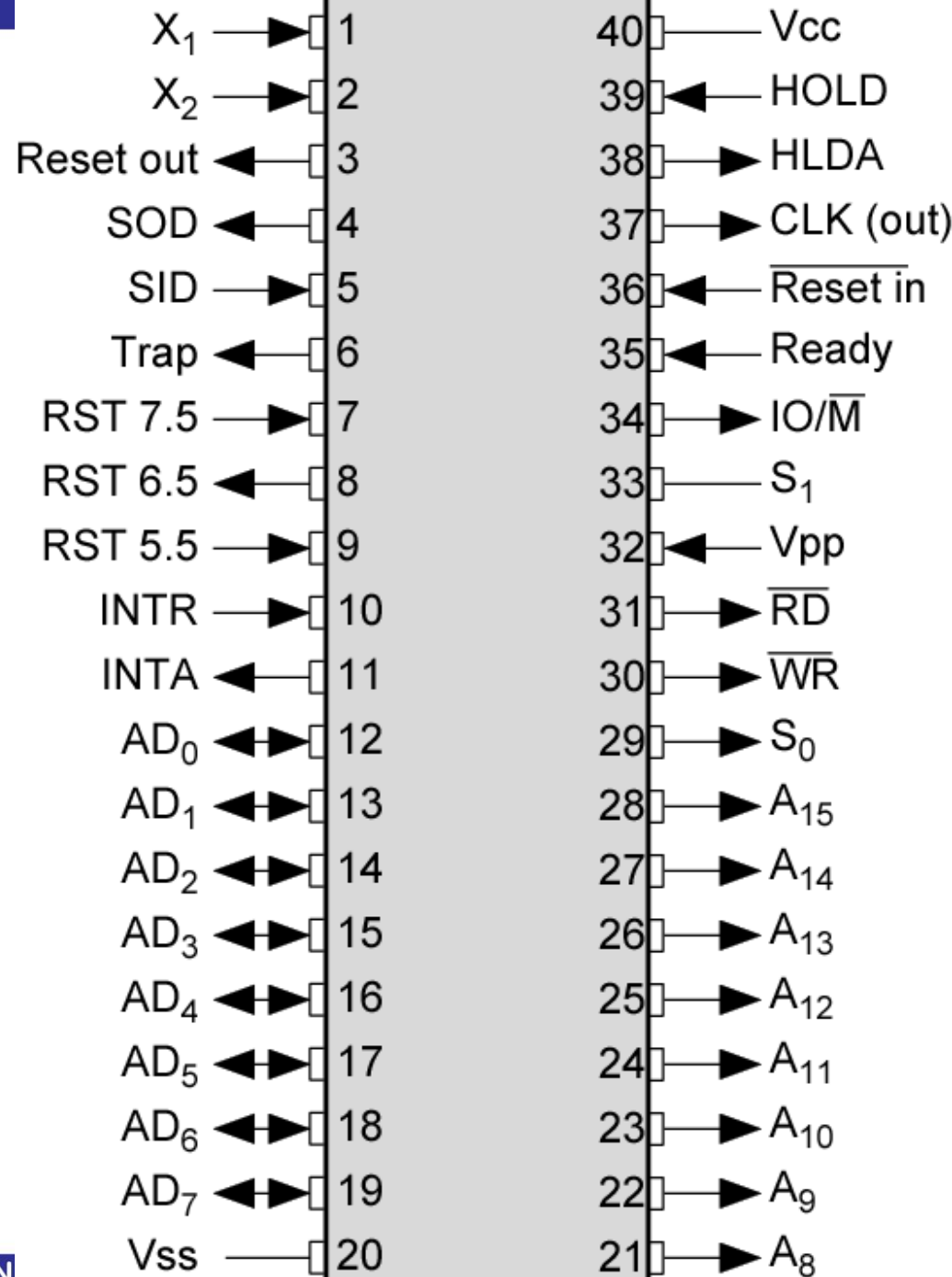
# CPU dengan Bus Internal



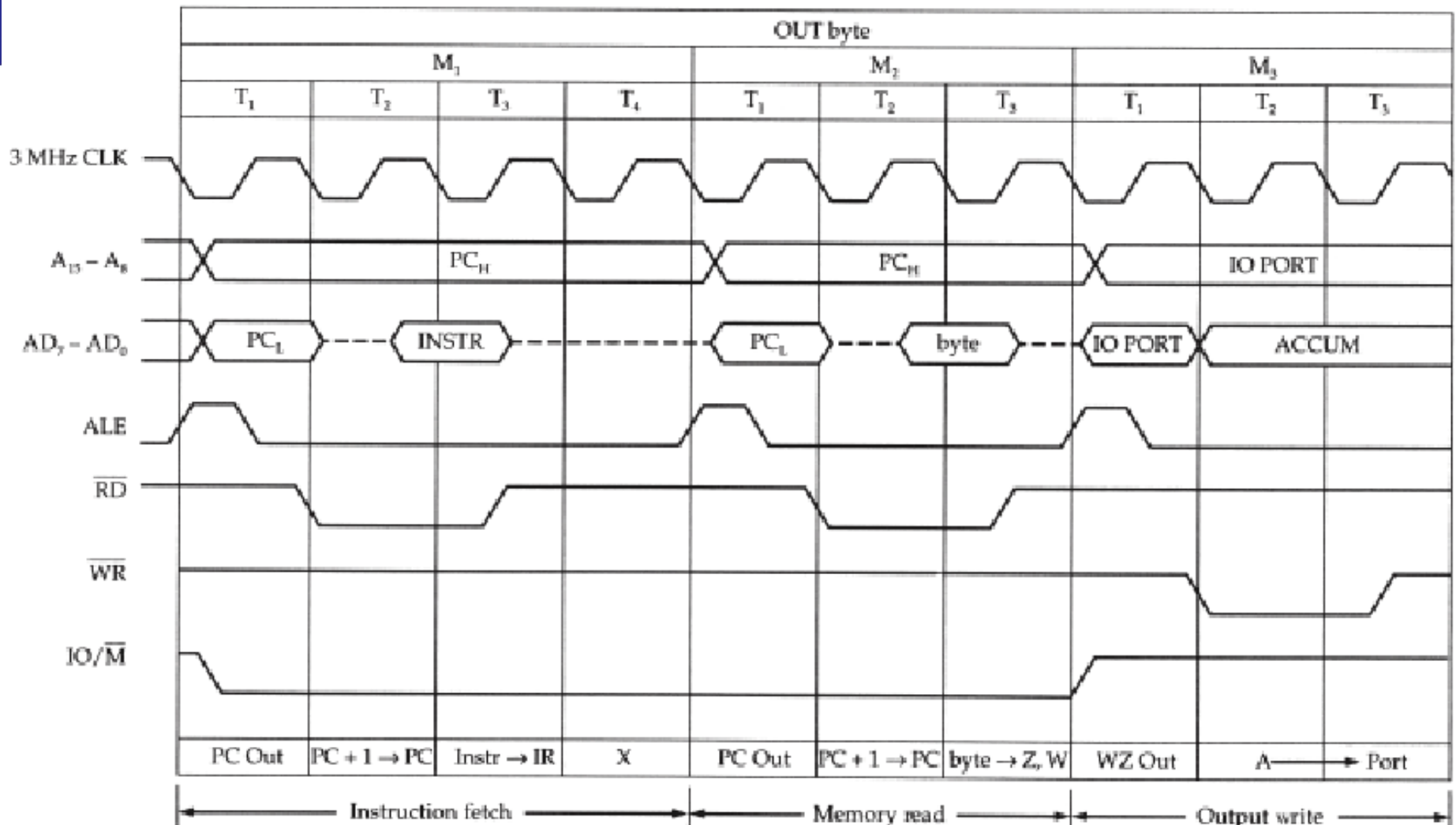
# Intel 8085 CPU Block Diagram



# Intel 8085 Pin Configuration



# Intel 8085 OUT Instruction Timing Diagram



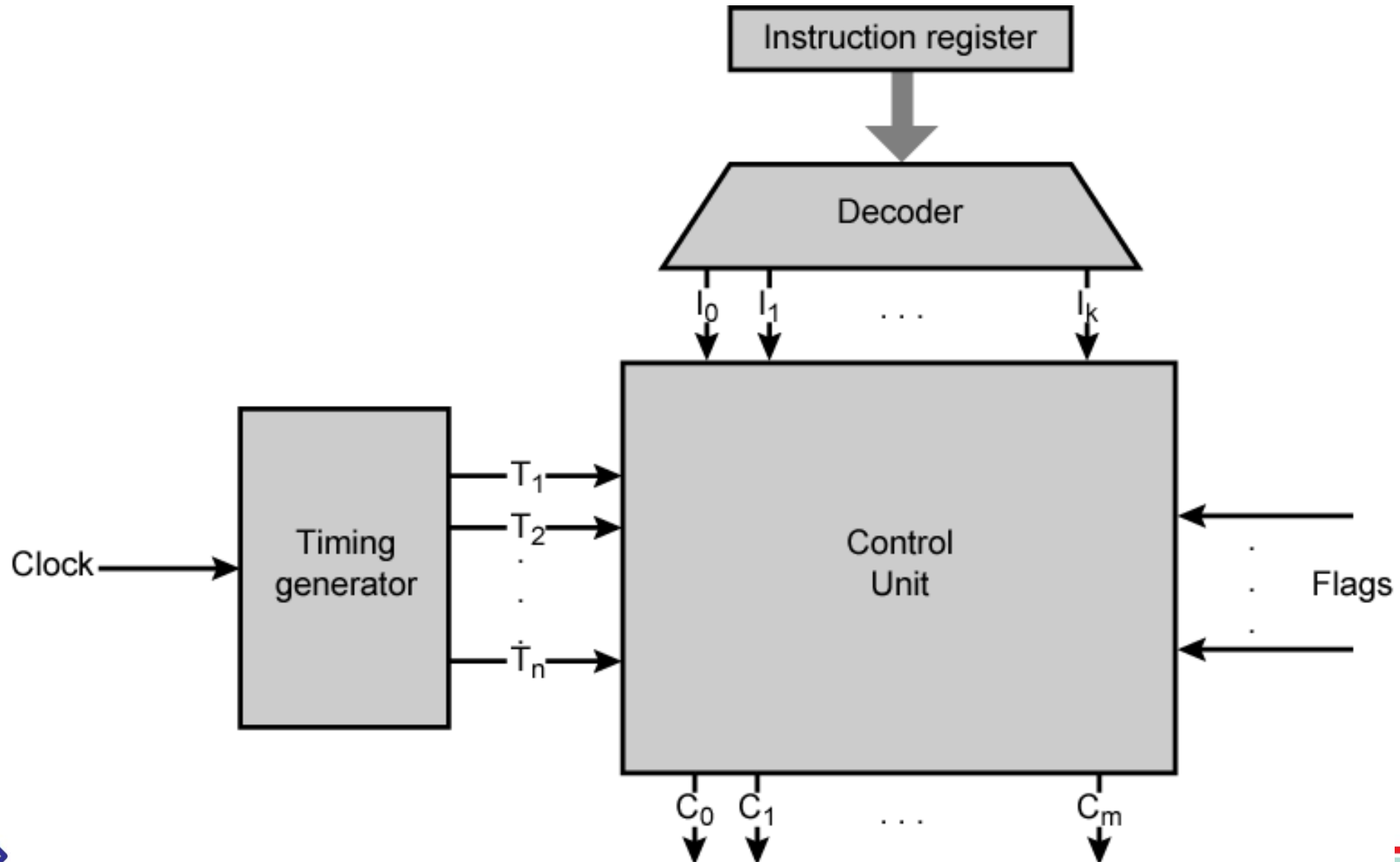
# Implementasi Terprogram (1)

- Control unit inputs
- Flags dan control bus
  - Berpengaruh pada setiap bit
- Instruction register
  - Op-code menyebabkan sinyal kontrol yang berbeda untuk setiap instruksi yang berbeda
  - Logika unik untuk setiap op-code
  - Decoder mengambil input yang disandikan dan menghasilkan output tunggal
  - $n$  binary inputs and  $2^n$  outputs

# Implementasi Terprogram (2)

- Clock
  - Urutan pulse yang berulang
  - Berguna untuk mengukur durasi micro-ops
  - Harus cukup lama untuk memungkinkan perambatan sinyal
  - Sinyal kontrol yang berbeda pada waktu yang berbeda dalam siklus instruksi
  - Perlu penghitung dengan sinyal kontrol yang berbeda untuk  $t_1$ ,  $t_2$  etc.

# Control Unit dengan Decoded Inputs





# Masalah Dengan Desain Terprogram

- Pengurutan komplek & logika micro-operation Difficult to design and test
- Desain tidak fleksibel
- Sulit untuk menambahkan instruksi baru

# Kesimpulan

- Control Unit Adalah salah satu bagian dari CPU yang bertugas untuk memberikan arahan / kendali / kontrol terhadap operasi yang dilakukan di bagian ALU (Arithmetic Logical Unit) di dalam CPU tersebut. Control unit dari sebuah prosesor memiliki 2 peran penting.
- Pertama, control unit mengatur processor agar melakukan semua micro-operation dalam urutan yang benar.
- Kedua, control unit menghasilkan control signal yang memastikan supaya semua micro-operation dieksekusi. Terdiri dari 2 jenis yaitu Single & Multi Control. Siklus instruksi terdiri dari micro operation, fetch, indirect, interrupt dan execution cycle.

# selesai