

Operasi Unit Kontrol

Pertemuan 15

Oleh :

Riyanto Sigit, ST, M.Kom

Nur Rosyid, ST

Setiawardhana, ST

Hero Yudo Martono, ST

Tujuan

1. Mengetahui tentang unit kontrol
2. Memahami operasi mikro pada pada siklus pengambilan, siklus tak langsung, siklus Interupt, siklus eksekusi, siklus instruksi.
3. Memahami cara kerja unit kontrol secara eksplisit, memahami sinyal kontrol
4. memahami implementasi hardwired, input-input unit control, Logika unit Control
5. memahami unit kontrol pada Intel 8085

Unit Kontrol

- bagian CPU yang menyebabkan fungsi komputer dapat tercapai.
- mengeluarkan sinyal – sinyal kontrol yang bersifat internal bagi CPU untuk memindahkan data antar register, agar ALU melakukan fungsinya dan untuk mengatur operasi – operasi internal lainnya.
- mengeluarkan sinyal kontrol eksternal bagi pertukaran data memori dan modul – modul I/O.

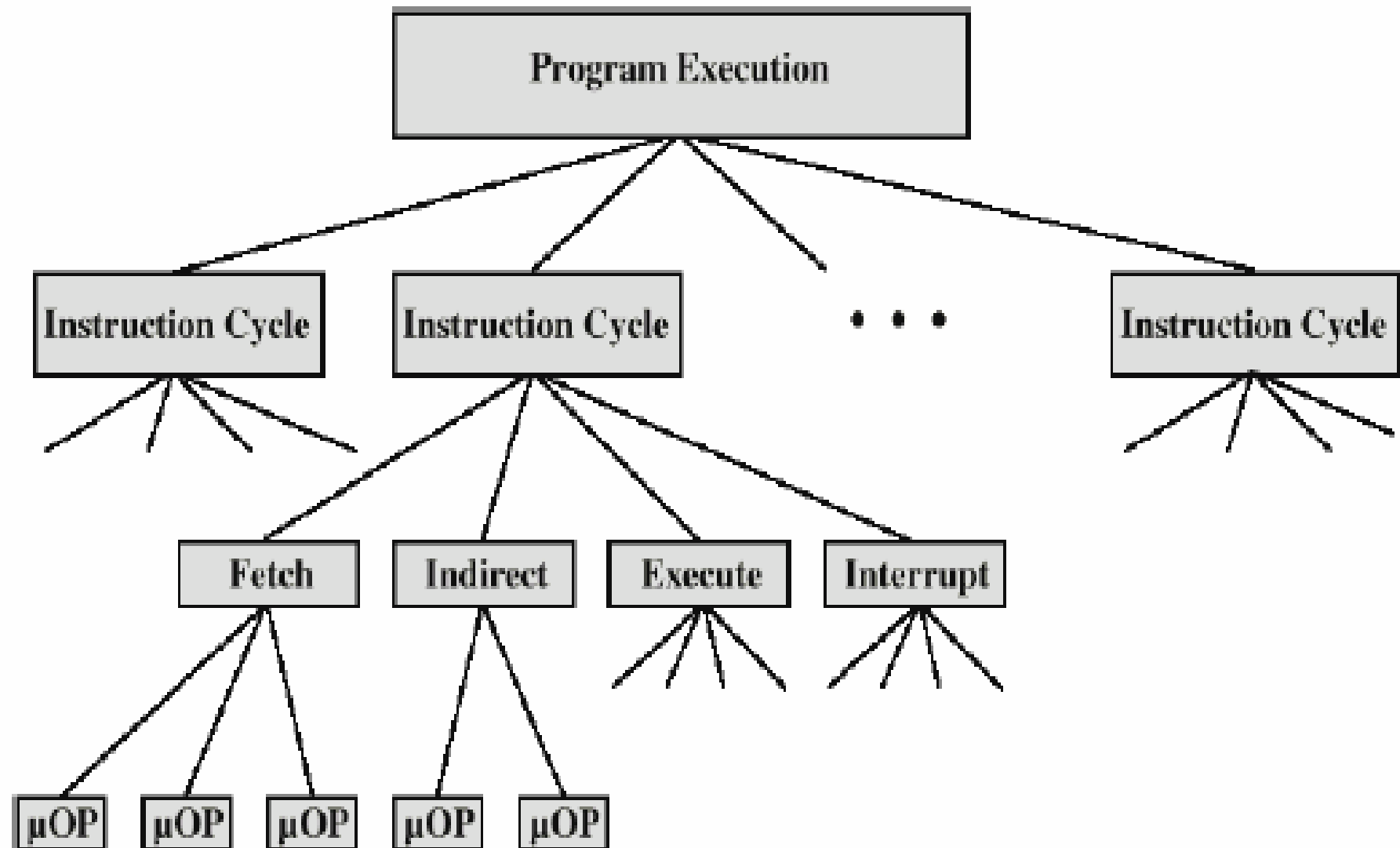
Fungsi CPU

1. Fungsi operasi (opcode)
2. Fungsi mode pengalamatan
3. Fungsi register
4. Fungsi antarmuka modul I/O
5. Fungsi modul memori
6. Fungsi pengolahan interrupt

Siklus Intruksi

- merupakan rangkaian tahapan operasi,
 - operasi pengambilan (fetch), indirect, eksekusi dan pengolahan interupsi bila ada
- dalam pembahasan unit kontrol, kita akan melihat tahapan operasi tersebut lebih mikro lagi, yang biasa disebut operasi mikro (micro operation)
 - Operasi fungsional atau atomik suatu CPU

Elemen Eksekusi Program



4 Register Operasi Pengambilan

1. Memory Address Register (MAR), dihubungkan pada bus alamat, untuk menspesifikasi alamat di dalam memori untuk operasi baca atau simpan.
2. Memory Buffer Register (MBR), dihubungkan pada bus data, untuk menyimpan data yang akan digunakan untuk operasi dan data akan disimpan ke memori.
3. Program Counter (PC), untuk menampung alamat instruksi berikutnya yang akan diambil.
4. Instruction Register (IR), untuk menampung instruksi terakhir yang diambil.

Langkah-langkah Siklus Pengambilan

1. Alamat instruksi berikutnya dimuatkan ke PC.
2. Pindahkan alamat ke MAR pada bus alamat.
3. Kontrol unit menspesifikasi perintah READ.
4. Hasil (data dari memori) diletakkan ke bus data.
5. Data di bus data dikirim ke MBR.
6. PC menambah 1 nilainya.
7. Data (instruksi) dipindahkan dari MBR ke IR.
8. MBR sekarang bebas untuk fetch berikutnya.

Penulisan Simbolik Perpindahan Data

t1 : MAR \leftarrow (PC) ; memindahkan isi PC ke MAR

t2 : MBR \leftarrow (memory) ; memindahkan isi lokasi memori yang dispesifikasi MAR ke MBR

PC \leftarrow (PC) + 1 ; menambah 1 ke isi PC

t3 : IR \leftarrow (MBR) ; memindah isi MBR ke IR

Aturan Pengelompokan Siklus

1. Rangkaian kejadian yang benar harus dipenuhi. Jadi operasi MAR © (PC) harus mendahului MBR © (memory) karena operasi pembacaan memori menggunakan alamat yang terdapat di dalam MAR. Kesimpulannya, rangkaian harus dibuat sesuai urutan kejadiannya.
2. Terjadinya konflik harus dihindari. Sebuah operasi mikro tidak diperbolehkan membaca dari register yang sama atau menulis ke register yang sama, karena hasilnya akan sulit diprediksi. Misalnya operasi mikro MBR © (memory) dan IR © (MBR) tidak boleh terjadi dalam satu satuan waktu yang sama.

Siklus Tak Langsung

- Setelah siklus pengambilan, siklus berikutnya adalah mengambil operand sumber.
- Dengan asumsi format instruksi satu alamat, dimana pengalamatan langsung dan tak langsung diijinkan.
- Apabila instruksi tersebut menspesifikasi alamat tak langsung, maka siklus tak langsung harus mendahului siklus eksekusi. Contoh aliran data tak langsung dalam operasi mikro :
 - t1 : MAR ζ (IRaddress)
 - t2 : MBR ζ (memory)
 - t3 : IRaddress ζ (MBRaddress)

Siklus Interrupt

- Pada akhir siklus eksekusi, dilakukan pengujian keberadaan interupsi.
- Bila terjadi interupsi, maka akan dijalankan siklus interupsi tersebut.
- Sifat siklus interupsi sebuah mesin berbeda sekali dengan interupsi pada mesin lainnya. Berikut sebuah contoh siklus interupsi dalam operasi mikro :
 - t1: MBR ζ (PC)
 - t2: MAR ζ save-address
 - PC ζ routine-address
 - t3: memory ζ (MBR)

Siklus Eksekusi

- Lebih kompleks dan banyak dibandingkan siklus – siklus sebelumnya yang telah dijelaskan.
- Pada siklus pengambilan, indirect dan interrupt cenderung sama untuk bermacam – macam program.
- Pada siklus eksekusi tergantung perintah eksekusinya.
- contoh operasi mikro untuk beberapa eksekusi instruksi
 - Operasi eksekusi penambahan
 - Operasi ISZ (increment and skip if zero)

• Operasi eksekusi penambahan :

- ADD R1,X ; tambahkan isi lokasi X ke R1 dan simpan hasilnya pada R1
- Operasi mikronya :
- t1: $\text{MAR} \Leftarrow (\text{IRaddress})$
- t2: $\text{MBR} \Leftarrow (\text{memory})$
- t3: $\text{R1} \Leftarrow \text{R1} + (\text{MBR})$

• Operasi ISZ (increment and skip if zero):

- ISZ X ; tambah X dengan 1, bila hasil = 0, loncati instruksi berikutnya.
- Operasi mikronya :
- t1: $\text{MAR} \Leftarrow (\text{IRaddress})$
- t2: $\text{MBR} \Leftarrow (\text{memory})$
- t3: $\text{MBR} \Leftarrow (\text{MBR}) + 1$
- t4: $\text{memory} \Leftarrow (\text{MBR})$
- if $(\text{MBR}) = 0$ then $\text{PC} \Leftarrow (\text{PC}) + 1$

Kontrol Prosesor

- Terdapat persyaratan fungsional bagi unit kontrol, yaitu fungsi – fungsi ini merupakan fungsi yang harus dilakukan unit kontrol.
- Definisi persyaratan fungsional ini merupakan dasar untuk perencanaan dan implementasi unit kontrol.
- Proses tiga langkah di bawah ini akan mampu memberikan karakterisasi unit kontrol :
 - 1. Menentukan elemen dasar CPU.
 - 2. Menjelaskan operasi mikro yang akan dilakukan CPU.
 - 3. Menentukan fungsi – fungsi yang harus dilakukan unit kontrol.