

PERTEMUAN

9

CENTRAL PROCESSING UNIT

ALU

(Arithmetic and Logic Unit)

- ALU merupakan bagian komputer yang berfungsi membentuk operasi-operasi aritmatika dan logik terhadap data
- Semua elemen lain sistem komputer (control unit, register, memori, I/O) berfungsi terutama untuk membawa data ke ALU untuk selanjutnya di proses dan kemudian mengambil kembali hasilnya.

Representasi Integer

$$- 1101.0101 = -11.3125$$

Representasi Nilai Tanda

- Bentuk yang paling sederhana representasi yang memakai bit tanda adalah representasi nilai tanda.
- Misal :
 $+18 = 00010010$
 $-18 = 10010010$
(sign magnitude/nilai tanda)
- Terdapat kekurangan pada cara diatas

Komplement-2

- $+7 = 0111$ $+18 = 00010010$
- $-7 = 1001$ $-18 = 11101101$
- Dapat di simpulkan bahwa hasil akan berbeda dengan nilai tanda

Representasi fixed point

Semua representasi di atas dapat pula disebut dengan fixed point, karena radix pointnya (binary pointnya) tetap dan di asumsikan akan berada di sebelah kanan.

6. Aritmatika Integer #1

A. Negasi

Untuk membuat negasi gunakan komplement dua (dianjurkan)

Penjumlahan negasi :

$$+7 = 0111$$

$$-7 = 1001$$

maka bila ada soal $(-7) + (+5) = 1001$

$$\begin{array}{r} 0101 \\ \hline 1110 \end{array}$$

Aritmatika Integer #2

Hasil = 1110 adalah bilangan negatif maka positifnya adalah = komplement 2-kan bilangan tersebut : 0010 = +2 maka bilangan 1110 adalah negatif dari 2 atau (-2)

Aturan overflow = Bila dua buah bilangan di tambahkan, dan keduanya positif atau keduanya negatif maka over flow akan terjadi jika dan hanya jika hasilnya memiliki tanda yang berlawanan.

Contoh Pengalian#1

B. Pengalian :

$$\begin{array}{r} 1011 \\ \times 1101 \\ \hline 1011 \\ 0000 \\ 1011 \\ \hline 1011 \\ 10001111 \end{array}$$

Perkalian

- Perkalian dengan bilangan negatif juga akan sama cuma negatif tersebut harus dihasilkan dari komplement 2
- Karena hasil kali $(-)$ dengan $(+)$ = $(-)$ maka hasil kali tersebut komplement duakan untuk mengetahui hasilnya.

C. Pembagian

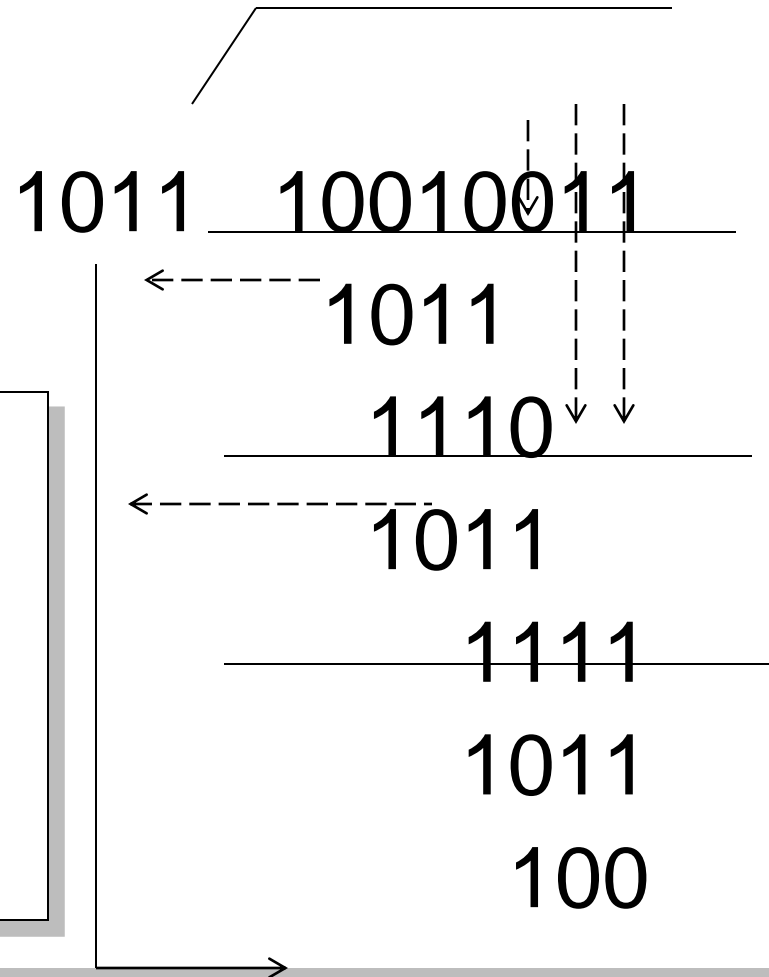
Keterangan :

1011 = divisor

10010011 = dividend

Hasil = quotient

Sisa = remainders



Representasi Floating Point

Representasi Floating Point

Misal :

$$976.000.000.000 = 9,76 \times 10^{11}$$

MENJADI

$$0,00000000000976 = 9,76 \times 10^{-10}$$

Aritmetika Floating Point

Penambahan dan pengurangan

- a. periksa bilangan-bilangan nol
- b. ratakan significand
- c. tambahkan atau kurangkan significand
- d. normalisasi hasilnya

Contoh soal

contoh :

$$123 \times 100 \longrightarrow 123 \times 100$$

$$456 \times 10^{-2} + \longrightarrow \frac{4,56 \times 100 +}{127,56 \times 100}$$

Perkalian dan Pembagian

a. Kalikan atau bagi significand

b. tambahkan atau kurangkan eksponensial

contoh :

$$123 \times 10^2$$

$$3 \times 10^{-4}$$

$$369 \times 10^{-2}$$

Operasi Mikro

- Fungsi dari sebuah komputer adalah untuk eksekusi program
- Setiap siklus yang lebih kecil akan terdiri dari sejumlah langkah yang masing-masing langkah tersebut terdiri dari register-register CPU. Dapat di sebut langkah-langkah tersebut sebagai operasi mikro.
- Operasi mikro adalah operasi fungsional atau atomik suatu CPU.

SIKLUS PENGAMBILAN # 1

- MAR dihubungkan dengan alamat bus sistem. MAR menspesifikasikan alamat di dalam memori untuk operasi read dan write.
- MBR dihubungkan dengan saluran data bus sistem.

MBR berisi nilai yang akan disimpan di memori atau nilai terakhir yang di baca dari memori

SIKLUS PENGAMBILAN # 2

- PC

Menampung alamat instruksi berikutnya yang akan di ambil.

- IR

Menampung instruksi terakhir yang diambil.

Siklus Pengambilan :

T1 : MAR	—————→	(PC)
T2 : MBR	—————→	Memori
PC	—————→	(PC) + 1
T3 : IR	—————→	(MBR)

Siklus Tak Langsung

T1 : MAR → (IR(Alamat))
T2 : MBR → Memori
T3 : IR (Alamat) → (MBR(Alamat))

Siklus Interrupt

T1 : MBR —————> (PC)
T2 : MAR —————> Alamat-simpan
PC —————> Alamat –rutin
T3 : Memori —————> (MBR)

Siklus Eksekusi #1

Penambahan (ADD)

1. ADD R1,X = Menambahkan isi lokasi X ke register R1

T1 : MAR \longrightarrow (IR(alamat))

T2 : MBR \longrightarrow memori

T3 : R1 \longrightarrow (R1)+(MBR)

Siklus Eksekusi #2

2. ISZ X = Isi lokasi X ditambahkan dengan 1. Apabila hasilnya sama dengan nol, maka instruksi berikutnya dilompati.

→

T1 : MAR → (IR(alamat))

T2 : MBR → Memori

T3 : MBR → (MBR) + 1

T4 : Memori (MBR)

If (MBR=0) then (PC PC+1)

Siklus Eksekusi #3

- 3. BSA X : Alamat instruksi yang berada setelah instruksi BSA disimpan di lokasi X, dan eksekusi dilanjutkan pada lokasi X+1. Alamat yang di simpan akan di gunakan kemudian untuk keperluan return.

T1 : MAR \longrightarrow (IR(alamat))
MBR \longrightarrow (PC)
T2 : PC \longrightarrow (IR(Alamat))
Memori \longrightarrow (MBR)
T3 : PC \longrightarrow (PC) + 1

Siklus Instruksi

- Setiap fase siklus instruksi dapat di uraikan menjadi operasi mikro elementer.
- Ada empat buah kode siklus instruksi (ICC).
- ICC menandai status CPU dalam hal bagian tempat siklus tersebut berada.

Kode ICC :

- 00 : fetch
- 01 : Indirect
- 10 : execute
- 11 : interrupt

KONTROL CPU

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 agar menyebabkan pembentukan operasi mikro

Elemen Dasar Fungsional CPU

1. ALU
2. Register-register
3. Lintasan data internal
4. Lintasan data eksternal
5. Unit Kontrol

Unit Kontrol melakukan dua tugas dasar :

- Pengurutan
- Eksekusi

Sinyal Kontrol

Input sinyal kontrol :

- Clock
- register Instruksi
- sinyal kontrol dari bus kontrol
- flag

Output sinyal kontrol :

- Sinyal kontrol di dalam CPU
- Sinyal kontrol bagi bus kontrol

selesai