

Organisasi Sistem Komputer

Bab 1. Arsitektur Komputer dan Representasi Bilangan

1.1 Sejarah Komputer

1.2 Arsitektur Komputer

1.3 Representasi Bilangan



Pembahasan:

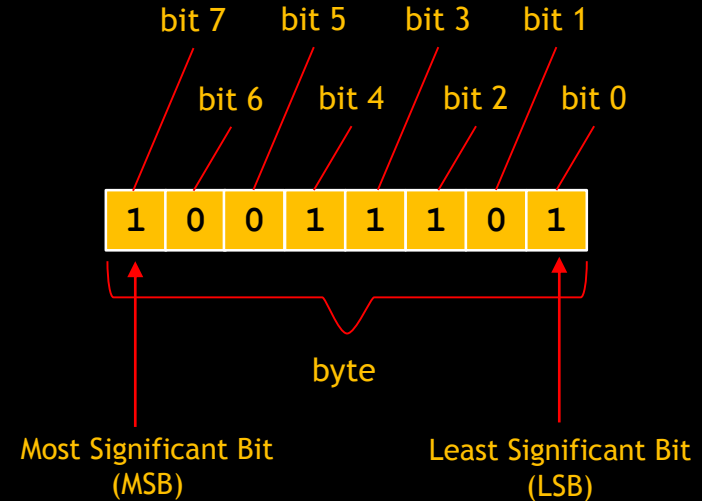
- Bilangan Biner
- Heksadesimal
- Representasi Bilangan Bertanda
 - Signed Magnitude
 - One's Complement
 - Two's Complement

Sistem Bilangan Posisional

- Kita sehari-hari menggunakan angka dalam bilangan desimal, yaitu sistem bilangan posisional dengan basis (atau radix) 10
- Setiap angka dalam desimal mempunyai nilai sama dengan penjumlahan dari hasil perkalian setiap digit dengan basisnya (yaitu 10) yang dipangkatkan dengan posisi digit tersebut
 - ❑ $25 = 2 \times 10^1 + 5 \times 10^0$
 - ❑ $136 = 1 \times 10^2 + 3 \times 10^1 + 6 \times 10^0$
- Sistem bilangan posisional basis tertentu yang sering digunakan dalam mempelajari komputer:
 - ❑ Basis 2: disebut dengan **biner**; setiap digit dapat berupa 0 atau 1
 - ❑ Basis 8: disebut dengan **octal**; setiap digit dapat berupa 0, 1, 2, 3, 4, 5, 6, 7
 - ❑ Basis 16: disebut dengan **heksadesimal**; setiap digit dapat berupa 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Bilangan Biner

- Komputer hanya mengerti biner
- Komputer menggunakan biner karena mudah direpresentasikan dalam sistem digital:
 - ❑ voltase rendah = 0 dan voltase tinggi = 1
- Digit biner disebut dengan **bit** (*binary digit*)
- Unit data terkecil yang dapat disimpan dalam komputer disebut dengan **byte** (biner 8 bit)
- Bit terkecil (paling kanan) disebut dengan **Least Significant Bit (LSB)**
- Bit terbesar (paling kiri) disebut dengan **Most Significant Bit (MSB)**



Bilangan Biner 8-bit

- Biner dengan banyak bit n mempunyai nilai dari 0 s.d $2^n - 1$
- Biner 8-bit (1 byte) merepresentasikan nilai 0 sampai dengan 255

Biner (8 bit)	Desimal
0000 0000	0
0000 0001	1
0000 0010	2
0000 0011	3
0000 0100	4
0000 0001	5
0000 0001	6
...	...
1111 1011	251
1111 1100	252
1111 1101	253
1111 1110	254
1111 1111	255

Konversi Biner ke Desimal

Konversi 10010110_2 ke desimal !

$$\begin{array}{cccccccc} 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ \hline 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \end{array}$$

$$= (1 \times 2^7) + (0 \times 2^6) + (0 \times 2^5) + (1 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (0 \times 2^0)$$

$$= 128 + 0 + 0 + 16 + 0 + 4 + 2 + 0$$

$$= 150_{10}$$



Konversi Desimal ke Biner

Konversi 87_{10} ke biner 8-bit !

		sis	
2	87	1	← bit 0
2	43	1	← bit 1
2	21	1	← bit 2
2	10	0	← bit 3
2	5	1	← bit 4
2	2	0	← bit 5
2	1	1	← bit 6
	0		

0	1	0	1	0	1	1	1
---	---	---	---	---	---	---	---

Penjumlahan Biner

$$1001_2 + 1111_2$$



Heksadesimal

- Bilangan berbasis 16 yang setiap digitnya dapat berupa 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
- Heksadesimal 1 digit merepresentasikan biner 4-bit

Heksadesimal	Biner (4 bit)	Desimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15


Heksadesimal Pada Biner 8-bit

- Heksadesimal sering digunakan untuk menyingkat penulisan biner
- Konversi biner ke heksadesimal dan sebaliknya mudah:
 - ❑ Biner ke Heksadesimal: konversi setiap 4-bit biner ke 1 digit heksadesimal
 - ❑ Heksadesimal ke Biner: konversi setiap digit heksadesimal ke 4-bit biner

Heksadesimal	Biner (8 bit)	Desimal
00	0000 0000	0
01	0000 0001	1
02	0000 0010	2
...
0A	0000 1111	15
10	0001 0000	16
11	0001 0001	17
12	0001 0010	18
...
1F	0001 1111	31
20	0010 0000	32
21	0010 0001	33
...
FD	1111 1101	253
FE	1111 1110	254
FF	1111 1111	255

Konversi Heksadesimal ke Desimal

Konversi AF_{16} ke desimal !

$\frac{A}{16^1}$	$\frac{F}{16^0}$		$\frac{10}{16^1}$	$\frac{15}{16^0}$
		konversi setiap digit ke desimal		

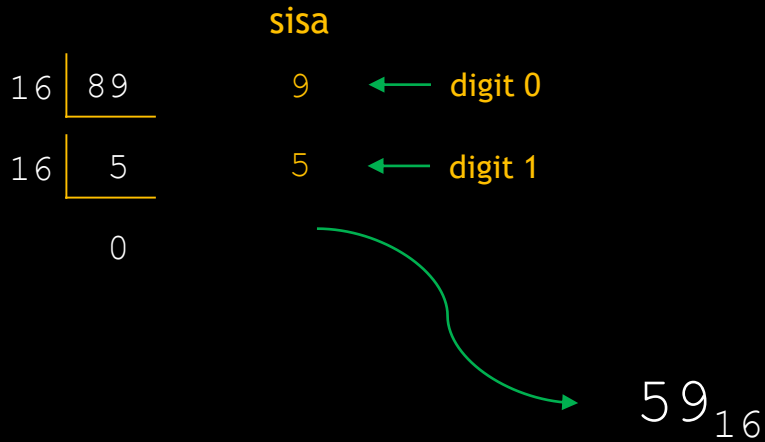
$$= (10 \times 16^1) + (15 \times 16^0)$$

$$= 160 + 15$$

$$= 175_{10}$$

Konversi Desimal ke Heksadesimal

Konversi 89_{10} ke heksadesimal !



Konversi Heksadesimal ke Biner

Konversi $A43FE2_{16}$ ke biner !

digit 5

$$\begin{aligned} &= A_{16} = 10_{10} \\ &= 1010_2 \end{aligned}$$

digit 4

$$\begin{aligned} &= 4_{16} = 4_{10} \\ &= 0100_2 \end{aligned}$$

digit 3

$$\begin{aligned} &= 3_{16} = 3_{10} \\ &= 0011_2 \end{aligned}$$

digit 2

$$\begin{aligned} &= F_{16} = 15_{10} \\ &= 1111_2 \end{aligned}$$

digit 1

$$\begin{aligned} &= E_{16} = 14_{10} \\ &= 1100_2 \end{aligned}$$

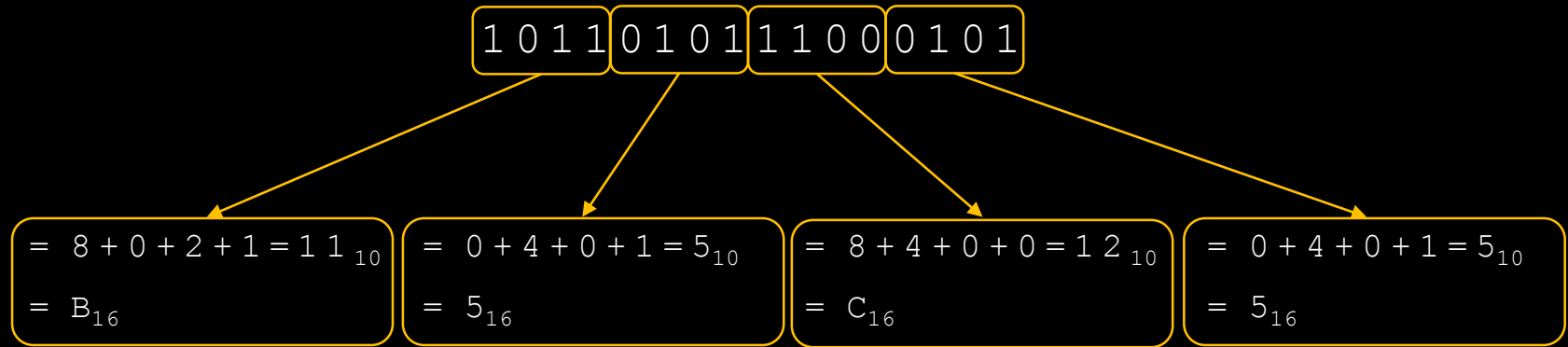
digit 0

$$\begin{aligned} &= 2_{16} = 2_{10} \\ &= 0010_2 \end{aligned}$$

1010 0100 0011 1111 1100 0010



Konversi Biner ke Heksadesimal



B5C5₁₆

Penjumlahan Heksadesimal

Hitung $A3_{16} + 17_{16}$!



Representasi Bilangan Bulat

- Komputer menyimpan bilangan bulat (*integer*) dalam satuan byte (8-bit) dan kelipatannya
- Terminologi untuk kelipatan byte:
 - ❑ 1-byte (8-bit) = **byte**
 - ❑ 2-byte (16-bit) = **word**
 - ❑ 4-byte (32-bit) = **double word**
 - ❑ 8-byte (64-bit) = **paragraph**
- Dua macam integer:
 - ❑ **Unsigned Integer** (bilangan bulat tak bertanda): bilangan bulat positif dan 0
 - ❑ **Signed integer** (bilangan bulat bertanda): bilangan bulat positif, negatif, dan 0
- Tiga metode untuk merepresentasikan bilangan bulat bertanda dalam biner:
 - ❑ Signed magnitude (Besaran bertanda)
 - ❑ One's complement (Komplemen satu)
 - ❑ Two's complement (Komplemen dua)



Signed Magnitude

- Most significant bit (MSB) digunakan sebagai tanda:

❑ MSB = 1 → tanda negatif

❑ MSB = 0 → tanda positif

Bilangan bulat
tidak bertanda

37_{10}

0	0	1	0	0	1	0	1
---	---	---	---	---	---	---	---

Bilangan bulat
bertanda (besaran
bertanda)

$+37_{10}$

0	0	1	0	0	1	0	1
---	---	---	---	---	---	---	---

-37_{10}

1	0	1	0	0	1	0	1
---	---	---	---	---	---	---	---

- Nilai 0 mempunyai dua representasi → memperumit logika CPU
- Tidak lagi di-implementasi dalam arsitektur CPU

Signed Magnitude (8-bit)	Integer Bertanda
1111 1111	-127
1111 1110	-126
1111 1101	-125
...	...
1000 0011	-3
1000 0010	-2
1000 0001	-1
1000 0000	-0
0000 0000	+0
0000 0001	+1
0000 0010	+2
0000 0011	+3
...	...
0111 1101	+125
0111 1110	+126
0111 1111	+127

One's Complement

- Nilai negatif didapat dengan membalik nilai setiap bit dari nilai positif

Bilangan bulat
tidak bertanda

87_{10}

0	1	0	1	0	1	1	1
---	---	---	---	---	---	---	---

Bilangan bulat
bertanda
(komplemen satu)

$+87_{10}$

0	1	0	1	0	1	1	1
---	---	---	---	---	---	---	---

-87_{10}

1	0	1	0	1	0	0	0
---	---	---	---	---	---	---	---

- Nilai 0 mempunyai dua representasi → memperumit logika CPU
- Tidak lagi di-implementasi dalam arsitektur CPU

One's Complement (8-bit)	Integer Bertanda
1000 0000	-127
1000 0001	-126
1000 0010	-125
...	...
1111 1100	-3
1111 1101	-2
1111 1110	-1
1111 1111	-0
0000 0000	+0
0000 0001	+1
0000 0010	+2
0000 0011	+3
...	...
0111 1101	+125
0111 1110	+126
0111 1111	+127

Two's Complement

- Nilai negatif didapat dengan menambahkan nilai 1 pada komplement satu

$+87_{10}$	<div>01010111</div>
(komplement satu)	
-87_{10}	<div>10101000</div>
	+ 1
(komplement dua) -87_{10}	<div>10101001</div>

- Nilai 0 hanya mempunyai satu representasi:
0000 0000
- Komputer saat ini menggunakan metode two's complement

Two's Complement (8-bit)	Integer Bertanda
1000 0000	-128
1000 0001	-127
1000 0010	-126
1000 0011	-125
...	...
1111 1101	-3
1111 1110	-2
1111 1111	-1
0000 0000	+0
0000 0001	+1
0000 0010	+2
0000 0011	+3
...	...
0111 1101	+125
0111 1110	+126
0111 1111	+127

Penjumlahan Komplemen Dua

- Misal, kita mempunyai dua angka 1-byte, A3 dan 17, dan kita menambahkan keduanya:

$$A3_{16} + 17_{16} = BA_{16}$$

- Jika saya mengartikannya sebagai bilangan tidak bertanda:

- $A3_{16} = 163_{10}$
- $17_{16} = 23_{10}$
- $BA_{16} = 186_{10}$
- dan tentu saja, $163_{10} + 23_{10} = 186_{10}$

- Jika saya mengartikannya sebagai bilangan bertanda (komplemen dua):

- $A3_{16} = -93_{10}$
- $17_{16} = 23_{10}$
- $BA_{16} = -70_{10}$
- dan tentu saja, $-93_{10} + 23_{10} = -70_{10}$



Penjumlahan Komplemen Dua

- Komputer tidak mengerti nilai-nilai yang kita simpan dan lakukan penjumlahan adalah bilangan bulat bertanda atau bilangan bulat tidak bertanda
- Kita yang harus mengartikannya sendiri
- Selama kita konsisten dalam mengartikan jenis bilangan, penjumlahan biner komplemen dua akan menghasilkan hasil yang benar
- Berlaku juga untuk pengurangan



Ringkasan

- Komputer hanya mengerti biner
- Heksadesimal digunakan untuk menyingkat biner
- Representasi bilangan bulat bertanda yang saat ini digunakan adalah two's complement (komplemen dua)
- Penjumlahan dan pengurangan dua bilangan bulat bertanda komplemen dua menghasilkan nilai yang benar jika kita konsisten mengartikannya