SEE 1223 - Elektronik Digit

Bab 2

Sistem Nombor

Sistem Nombor

- Kebanyakan sistem komputer (sistem Digital) melakukan operasi pengiraan nombor dalam kuantiti yang banyak!
- Maka, Sistem Pernomboran yang digunakan oleh Sistem Digital perlu diketahui dari segi:-
  - Bagaimana pernyataan nombor tersebut!
  - Bagaimana operasi arithmetik dilakukan!

Sistem Nombor

- Jenis-jenis Sistem Nombor:-
  - Decimal (asas 10)
  - Binary (asas 2)
  - Octal (asas 8)
  - Hexadecimal (asas 16)

Sistem Nombor

- Nombor Decimal
  - Terdiri daripada 10 angka iaitu 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Ia merupakan nombor 'Asas 10'.
  - Salah satu contoh dalam sistem nombor Decimal adalah **1428.79** atau $\mathbf{1428.79_{10}}$. Kedudukan setiap digit menunjukkan magnitud bagi setiap digit tersebut iaitu:-

Pemberat	10^3	10^2	10^1	10^0		10^{-1}	10^{-2}
Nilai	1	4	2	8	.	7	9

  - Secara pernyataan matematik:-
    $\mathbf{1428_{10}} = \mathbf{1} \times 10^3 + 4 \times 10^2 + \mathbf{2} \times 10^{1} + \mathbf{8} \times 10^0$

Sistem Nombor

- Nombor Binary
  - Terdiri daripada 2 angka iaitu 0,1. Ia merupakan nombor 'Asas 2'.
  - Salah satu contoh dalam sistem nombor Binary adalah **1001.01** atau $\mathbf{1001.01_2}$. Kedudukan setiap digit menunjukkan magnitud bagi setiap digit tersebut iaitu:-

Pemberat	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}
Nilai	1	0	0	1	.	0	1

  - Secara pernyataan matematik:-
    $\mathbf{1001_2} = \mathbf{1} \times 2^3 + \mathbf{0} \times 2^2 + \mathbf{0} \times 2^{1} + \mathbf{1} \times 2^0$
  - Dalam no. Binary, bilangan digit dipanggil *bit*.

Sistem Nombor

- Nombor Octal
  - Terdiri daripada 8 angka iaitu 0, 1, 2, 3, 4, 5, 6, 7. Ia merupakan nombor 'Asas 8'.
  - Salah satu contoh dalam sistem nombor Octal adalah **5641.27** atau $\mathbf{5641.27_8}$. Kedudukan setiap digit menunjukkan magnitud bagi setiap digit tersebut iaitu:-

Pemberat	8^3	8^2	8^1	8^0		8^{-1}	8^{-2}
Nilai	5	6	4	1	.	2	7

  - Secara pernyataan matematik:-
    $\mathbf{5641_8} = \mathbf{5} \times 8^3 + \mathbf{6} \times 8^2 + \mathbf{4} \times 8^{1} + \mathbf{1} \times 8^0$

Sistem Nombor

- Nombor Hexadecimal
  - Terdiri daripada 16 angka iaitu 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Ia merupakan nombor 'Asas 16'.
  - Salah satu contoh dalam sistem nombor hexadecimal adalah **5B8F.21** atau **5B8F.21$_8$.** Kedudukan setiap digit menunjukkan magnitud bagi setiap digit tersebut iaitu:-

Pemberat	16^3	16^2	16^1	16^0		16^{-1}	16^{-2}
Nilai	5	B	8	F	.	2	1

  - Secara pernyataan matematik:-

    5B8F$_8$= **5** x 16^3 + **B** x 16^2 + **8** x 16^1 + **F** x 16^0

Sistem Nombor

- Rumusan: setiap sistem nombor di atas dinyatakan dalam bentuk 'Tatanda Nilai Kedudukan Berpemberat' (Weighted Positional Value Notation).
- Secara umumnya, suatu nombor $N = a_{n-1} \ldots a_0$ boleh dinyatakan dalam asas *b* seperti berikut:-

$$(N)_b = a_{n-1}b^{n-1} + a_{n-2}b^{n-2} + \ldots + a_1b^1 + a_0b^0$$

Most Significant Bit (MSB)

Least Significant Bit (LSB)

Decimal	Binary	Octal	Hexadecimal
0	0000	00	0
1	0001	01	1
2	0010	02	2
3	0011	03	3
4	0100	04	4
5	0101	05	5
6	0110	06	6
7	0111	07	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Penukaran Nombor

- Penukaran Binary → Decimal
  - **1001**$_2$ = **1** x 2^3 + **0** x 2^2 + **0** x 2^1 + **1** x 2^0

    = 8 + 0 + 0 + 1

    = 9_{10}
- Penukaran Decimal → Binary

  $18_{10} = ?_2$

  Berhenti apabila N = 0

  $18_{10} = 10010_2$

	N	X	Baki (N-X)	
2	18	=18	0	LSB
2	9	=8	1	
2	4	=4	0	
2	2	=2	0	
2	1	=0	1	MSB
	0			

10010

Penukaran Nombor

- Latihan
  - Tukar no. Decimal kepada no. Binary,

    (a) 33_{10} (b) 78_{10} (c) 101_{10}
  - Tukar no. Binary kepada no. Decimal,

    (a)111101_2 (b) 011010110_2 (c) 0101_2
  - Tukar no. pecahan kepada no. Binary,

    (a) 23.25_{10} (b) 68.75_{10}

Penukaran Nombor

- Penukaran Octal → Decimal
  - **1271**$_8$ = **1** x 8^3 + **2** x 8^2 + **7** x 8^1 + **1** x 8^0

    = 512 + 128 + 56 + 1

    = 697_{10}
- Penukaran Decimal → Octal

  $697_{10} = ?_8$

  Berhenti apabila N = 0

  $697_{10} = 1271_8$

	N	X	Baki (N-X)	
8	697	=696	1	LSB
8	87	=80	7	
8	10	=8	2	
8	1	=0	1	MSB
	0			

1 2 7 1

Penukaran Nombor

- Penukaran Octal → Binary dan Binary → Octal
  - Ada dua kaedah penukaran iaitu secara
    - 'terus' (direct conversion) atau
    - melalui decimal (octal ↔ decimal ↔ binary)
  - Untuk melaksanakan penukaran melalui kaedah direct conversion, hubungan antara no. octal '1' digit dan no. binary '3' digit perlu diketahui.
  - Kaedah kedua, iaitu melalui decimal telah dipelajari!
- Tukarkan no berikut:-
  - (a) $276_8 \rightarrow ?_2$ (b) $1010111_2 \rightarrow ?_8$

Octal	Binary
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Penukaran Nombor

- Penukaran Hexadecimal → Decimal
  - $\mathbf{1E5}_{16} = \mathbf{1} \times 16^2 + \mathbf{E} \times 16^1 + \mathbf{5} \times 16^0$
    $= 256 + 14X16 + 5$
    $= 485_{10}$
- Penukaran Decimal → Hex
  - $485_{10} = ?_{16}$
  - Berhenti apabila N = 0
  - $485_{10} = 1E5_{16}$

	N	X	Baki (N-X)	
16	485	=480	5	LSB
16	30	=16	$14_{10}=E_{16}$	
16	1	=0	1	MSB
	0			

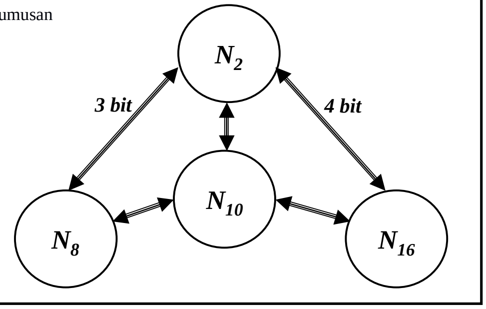
1 E 5

Penukaran Nombor

- Penukaran Hex → Binary dan Binary → Hex
  - Ada dua kaedah penukaran iaitu secara
    - 'terus' (direct conversion) atau
    - melalui decimal (hex ↔ decimal ↔ binary)
  - Untuk melaksanakan penukaran melalui kaedah direct conversion, hubungan antara no. hex '1' digit dan no. binary '4' digit perlu diketahui.
  - Kaedah kedua, iaitu melalui decimal telah dipelajari!
- Tukarkan no berikut:-
  - (a) $F5_{16} \rightarrow ?_2$ (b) $1010111_2 \rightarrow ?_{16}$
  - (c) $775_8 \rightarrow ?_{16}$ (d) $A1_{16} \rightarrow ?_8$

Penukaran Nombor

- Rumusan



Sistem Nombor

- Kenapa perlu ada berbagai sistem nombor?
  - Peralatan/peranti terdahulu melakukan pengiraan komputer dengan menggunakan suis dan geganti.
  - Suis/geganti hanya ada dua keadaan sahaja iaitu 'tutup' dan 'buka'.
  - Maka operasi komputer tersebut dapat dinyatakan dalam digit nombor Binary, yang terdiri daripada dua keadaan sahaja.
  - Jika sistem Decimal digunakan, maka ia memerlukan suatu peralatan/peranti yang boleh menghasilkan 10 keadaan! Hasilnya adalah suatu peranti yang kompleks, tak 'reliable', tidak cekap dan mahal!

Sistem Nombor

- Ok, no. Binary sesuai digunakan dalam sistem komputer, manakala no Decimal tidak cekap. Bagaimana pula dengan nombor Octal &Hexadecimal? Kenapa perlu wujudkan sistem nombor Octal dan Hex?
- Rujuk jadual!
- Bagi suatu nombor yang kecil,(cthnya 15_{10}), hanya 2 digit no Decimal shj diperlukan, tetapi 4 digit Binary diperlukan.
- Bagi no Hex, hanya 1 digit diperlukan!
- 'In terms of numbers ***representation/expression***, decimal is better, but Hex is the best!'
- Bagi 6 digit no. Decimal, 18 digit Binary diperlukan, 6 digit no Octal diperlukan dan hanya 4 digit Hex diperlukan!

Sistem Nombor

- Maka didapati, pernyataan nombor dapat dilakukan dengan lebih berkesan, dengan mengkodkan dan memadatkan maklumat dalam sistem nombor tersebut.
- Secara praktikal, ia banyak digunakan!
- Contohnya dalam penyataan alamat dalam 'memory'.

Sistem Nombor

Kod-kod Binary

- **Kod BCD (Binary Coded Decimal)**
  - Mengungkapkan setiap digit Decimal kepada 4 digit Binary.
  - Salah satu kod yang popular adalah kod BCD 8421, dimana pemberat bagi digit binary hanya terhad kepada $2^3, 2^2, 2^1, 2^0$.
- Tukarkan Decimal → BCD 8421 dan Binary

  (a) 30_{10} (b) 874_{10}
- Tukarkan BCD 8421 → Decimal

  (a) 01001001 (b) 100101110001

Kod BCD 8421

Decimal	Binary	BCD 8421
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001
10	1010	0001 0000
11	1011	0001 0001
12	1100	0001 0010

Kod-kod Binary

- **Kod Excess -3**
  - Kod ini terbit dengan menambah 3_{10} kepada kod BCD iaitu $N_{BCD} + 3_{10} = N_{BCD} + 11_2$.
  - Kod ini tidak berpemberat, dan ia merupakan salah satu kod BCD
- Tukarkan Decimal → BCD dan Binary

  (a) 30_{10} (b) 874_{10}
- Tukarkan BCD → Decimal

  (a) 01001001 (b) 100101110001

Kod Excess -3

Decimal	Binary	BCD 8421	Excess -3
0	0000	0000	0011
1	0001	0001	0100
2	0010	0010	0101
3	0011	0011	0110
4	0100	0100	0111
5	0101	0101	1000
6	0110	0110	1001
7	0111	0111	1010
8	1000	1000	1011
9	1001	1001	1100
10	1010	0001 0000	0100 0011
11	1011	0001 0001	0100 0100
12	1100	0001 0010	0100 0101

Kod-kod Binary

- **Kod Gray**
  - Dlm kod ini, perubahan satu nombor ke nombor selepasnya hanya 1 bit sahaja yang berubah!
  - Ctrhnya
    7_{10} = 0111 , nombor seterusnya, 8_{10} = 1111, bukan 1000.
  - Kod ini tidak berpemberat, dan ia juga merupakan salah satu kod BCD
- Kod Gray ke Binary

1 1 1 0 → Gray

1 0 1 1 → Binary

Kod-kod Binary

- **Binary ke kod Gray**

1 1 1 0 → Binary

1 0 0 1 → Gray

Kod-kod Binary

- **Kod ASCII**
  - ASCII merupakan singkatan kepada 'American Standard Code of Information Interchange'.
  - Merupakan kod 'Alphanumeric' iaitu kod yang dinyatakan dalam nombor dan abjad.
  - Terdiri daripada nombor 7 bit, utk mewakilkan 128 aksara, iaitu 2^n.
  - Huruf 'A' cthnya diwakilkan oleh kod ASCII 1000001_2, 'ESC' diwakilkan oleh 0011011_2
- Dapatkan maklumat yang dinyatakan oleh kod ASCII berikut:

  1001000 1000101 1001100 1010000

Saiz Nombor

- Julat sesuatu nombor ditentukan oleh saiz atau bilangan atau jumlah bit yang digunakan!
  - Ditentukan oleh persamaan $\mathbf{2^n}$ dimana n adalah bilangan bit!
  - Contohnya,
    Suatu nombor 4 bit boleh mewakili 2^4 = 16 nombor! (0-15)
    Suatu nombor 6 bit boleh mewakili 2^6 = 64 nombor! (0-63)
  - Saiz bit dan gelarannya
    - 4 bit → 1 nibble
    - 8 bit → 1 byte
    - 16 bit → 1 word
    - 32 bit → 1 long word

Aritmetic Binary

- Tambah
  - Penambahan adalah berasaskan 4 kombinasi di bawah:
    0+0 = 0
    0+1 = 1
    1+0 = 1
    1+1 = 0 bawa/carry 1
  - Contohnya,

01(1)	1001(9)
+11(3)	+1111(15)
100(4)	11000(24)

Aritmetic Binary

- Tolak
  - Penolakan adalah berasaskan 4 kombinasi di bawah:
    0-0 = 0
    0-1 = 1 pinjam 1
    1-0 = 1
    1-1 = 0
  - Contohnya,

111(7)	10100(20)
- 101(5)	- 10000(16)
010(2)	00100(4)

Nombor Bertanda

- Dalam sistem no. Decimal, nombor negatif ditandakan dengan '-'. Dalam sistem no. Binary, no negatif @ positif ditandakan dengan bit yang berada paling kiri!
- Bit '0' menandakan no. positif, bit '1' menandakan no. negatif!
- Ada 3 jenis nombor bertanda iaitu:
  - Perwakilan magnitud bertanda (sign-magnitude)
  - Perwakilan pelengkap 1 (1's complement)
  - Perwakilan pelengkap 2 (2's complement)

Nombor Bertanda

- Sistem Magnitud Bertanda (sign-magnitude)
  - Pada bit paling kiri, bit '0' menandakan no. positif, bit '1' menandakan no. negatif!
  - Contohnya, nombor $+25_{10}$, ditukarkan kepada nombor 'magnitud bertanda' adalah

$$\underset{\textbf{Tanda +ve}}{0}\underbrace{0011001}_{\textbf{Bit Magnitud}}$$

  - Tetapi nombor -25_{10} pula adalah

$$\underset{\textbf{Tanda -ve}}{1}\underbrace{0011001}_{\textbf{Bit Magnitud}}$$

  - Didapati magnitud masih sama tetapi tanda sahja yang berubah

Nombor Bertanda

- Sistem Pelengkap 1 (1's complement)
  - Nombor pelengkap 1 (pelengkap (r-1)) bagi sesuatu nombor N dalam dasar r yang mengandungi n digit adalah

    $(r^n - 1) - N$
  - Suatu no Binary 0111 ditukarkan kepada pelengkap (r-1)

    r = 2; n = 4; N = 0111;

    Maka, pelengkap 1 bagi $(0111)_2 = (2^4-1)_2 - 0111_2 = 1000_2$
  - Pelengkap 1 hanya digunakan untuk menandakan nombor negatif!
  - Nombor negatif untuk pelengkap 1 diperolehi dengan menggunakan persamaan di atas atau tukarkan bit '1' kepada '0' dan bit '0' kepada '1'.

Nombor Bertanda

- Sistem Pelengkap 2 (2's complement)
  - Nombor pelengkap 2 (pelengkap r) bagi sesuatu nombor N dalam dasar r yang mengandungi n digit adalah

    $[(r^n - 1) - N]+1$
  - Suatu no Binary 0111 ditukarkan kepada pelengkap (r)

    r = 2; n = 4; N = 0111;

    Maka, pelengkap 1 bagi $(0111)_2 = (2^4-1)_2 - 0111_2+1$

    $= 1001_2$
  - Pelengkap 2 juga hanya digunakan untuk menandakan nombor negatif!
  - Nombor negatif untuk pelengkap 1 diperolehi dengan menggunakan persamaan di atas atau tukarkan

Nombor Bertanda

- Sistem Pelengkap 2 (2's complement)
  - Nombor negatif untuk pelengkap 1 diperolehi dengan
    - menggunakan persamaan di atas, atau
    - menambahkan 1 kepada no. pelengkap 1, atau
    - mengekalkan bit '1' yang paling kanan, dan semua bit '0' di sebelah kanannya, manakala bit di sebelah kiri ditukarkan '1' kepada '0' dan '0' kepada '1'.
  - Contoh

    +35 = 00100011

    -35 = 10100011 (magnitud tanda)

    = 11011100 (pelengkap 1)

    = 11011101 (pelengkap 2)

Nombor Bertanda

- **Julat nombor**
  - Julat nombor ***tak bertanda*** ;

    $0\ldots\ldots2^n-1$, dimana n adalah bilangan bit

    bagi nombor 8 bit, julatnya adalah;

    $0\ldots\ldots2^8-1(255)$
  - Julat nombor ***bertanda***;

    $-2^{n-1}\ldots\ldots0\ldots\ldots2^{n-1}-1$, n adalah bilangan bit

    cth; bagi nombor 8 bit, julatnya adalah;

    $-2^{8-1}(-128)\ldots\ldots0\ldots\ldots2^{8-1}-1(127)$

Nombor Bertanda

- **Arithmetic nombor bertanda**
  - Dalam no pelengkap 2, semua operasi dilakukan dengan penambahan sahaja

```
  (50)    00110010          (50)   00110010
+ (20)    00010100        +(-20)   11101100
  (70)    01000110          (30)   00011110

  (-50)     11001110        (-50)     11001110
+ (-50)     11001110      +  (50)     00110010
 (-100)   1 10011100          (0)   1 00000000
          ↑                         ↑
       abaikan                   abaikan
```