Data Representation

Computer System Architecture

By

M. Morris Mano

Data Types

- Information that a Computer is dealing with
  - Data
  - Relationship between data elements
  - Program (Instruction)
- Binary information is stored in ***memory*** or ***processor registers***
- Registers contain either ***data*** or ***control information***
  - ***Data*** are numbers and other binary-coded information
  - ***Control information*** is a bit or a group of bits used to specify the sequence of command signals

Data Types cont..

- **Data**
  - **Numeric Data**
    - Numbers(Integer, real)
  - **Non-numeric Data**
    - Letters, Symbols
- Data types found in the registers of digital computers
  - ***Numbers*** used in arithmetic computations
  - ***Letters*** of the alphabet used in data processing
  - ***Other discrete symbols*** used for specific purpose

Number Systems

1. **Nonpositional number system**
   – Roman number system
2. **Positional number system**
   – Each digit position has a value called a *weight* associated with it.
   – Also called **weight System.**
   – Most common number system
      • Decimal, Octal, Hexadecimal, Binary

Base or **Radix r system**

- Uses distinct symbols for ***r digits***
- r =10   Decimal number system
- r = 2   Binary
- r = 8   Octal
- r =16   Hexadecimal
- $\mathbf{r^2\ r^1 r^0 . r^{-1}\ r^{-2}\ r^{-3}}$
- Multiply each digit by an integer power of r and then form the sum of all weighted digits
- **Ex:** $\mathbf{A_R = a_{n-1}\ a_{n-2} \ldots a_1\ a_0 . a_{-1} \ldots a_{-m}}$

  – $\mathbf{V(A_R) =} \sum_{i=-m}^{n-1} a_i R^i$

Radix point(.) separates the integer portion and the fractional portion

Common number system

- **Decimal System/Base-10 System**
  - Composed of 10 symbols or numerals
  - (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
- **Binary System/Base-2 System**
  - Composed of 2 symbols or numerals
  - (0, 1)
- **Hexadecimal System/Base-16 System**
  - Composed of 16 symbols or numerals
  - (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F)

Common number system cont..

Table 3-2

Hex	Binary	Decimal
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

Conversions

- **Binary-to-Decimal Conversions**

$$1011.101_2 = (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) + (1 \times 2^{-1}) + (0 \times 2^{-2}) + (1 \times 2^{-3})$$
$$= 8_{10} + 0 + 2_{10} + 1_{10} + 0.5_{10} + 0 + 0.125_{10}$$
$$= 11.625_{10}$$

- **Decimal-to-Binary Conversions**
  - **Repeated division**

37 / 2 = 18 remainder 1 (binary number will end with 1) : **LSB**
18 / 2 = 9 remainder 0
9 / 2 = 4 remainder 1
4 / 2 = 2 remainder 0
2 / 2 = 1 remainder 0
1 / 2 = 0 remainder 1 (binary number will start with 1) : **MSB**

Read the result upward to give an answer of $37_{10} = 100101_2$

Conversions cont..

- **Hex-to-Decimal Conversion**

$$2AF_{16} = (2 \times 16^2) + (10 \times 16^1) + (15 \times 16^0)$$
$$= 512_{10} + 160_{10} + 15_{10}$$
$$= 687_{10}$$

- **Decimal-to-Hex Conversion**

423_{10} / 16 = 26 remainder 7 (Hex number will end with 7) : **LSB**

26_{10} / 16 = 1 remainder 10

1_{10} / 16 = 0 remainder 1 (Hex number will start with 1) : **MSB**

Read the result upward to give an answer of $423_{10} = 1A7_{16}$

Conversions cont..

- **Hex-to-Binary Conversion**

$$9F2_{16} = 9 \quad F \quad 2$$

$$\downarrow \quad \downarrow \quad \downarrow$$

$$= 1001 \quad 1111 \quad 0010$$

$$= 100111110010_2$$

- **Binary-to-Hex Conversion**

$$1\,1\,1\,0\,1\,0\,0\,1\,1\,0_2 = \underbrace{0\,0\,1\,1}_{3}\ \underbrace{1\,0\,1\,0}_{A}\ \underbrace{0\,1\,1\,0}_{6}$$

$$= 3A6_{16}$$

Binary-Coded-Decimal Code

- Each digit of a decimal number is represented by its binary equivalent

8	7	4	(Decimal)
↓	↓	↓	
1000	0111	0100	(BCD)

- Only the four bit binary numbers from 0000 through 1001 are used
- Comparison of BCD and Binary

$137_{10} = 10001001_2$ **(Binary)** - *require only 8 bits*

$137_{10} = 0001\ 0011\ 0111_{BCD}$ **(BCD)** - *require 12 bits*

Alphanumeric Representation

- Alphanumeric character set
  - 10 decimal digits, 26 letters, special character($, +, =,….)
  - A complete list of ASCII
  - ASCII (American Standard Code for Information Interchange)
  - Standard alphanumeric binary code uses seven bits to code 128 characters

Character Representation ASCII

MSB (3 bits)

LSB (4 bits)

	0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P	‘	P
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	“	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	‘	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	I	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	m	n	~
F	SI	US	/	?	O	n	o	DEL

Complements

- ***Complements*** are used in digital computers for simplifying the ***subtraction operation*** and for ***logical manipulation***
- There are two types of complements for base r system

1. **r's complement**
2. **(r-1)'s complement**

– Binary number : 2's or 1's complement

– Decimal number : 10's or 9's complement

(r-1)'s Complement and r's Complement

- (r-1)'s Complement
  - (r-1)'s Complement of N = (r^n-1)-N
    - » 9's complement of N=***546700***

      (10^6-1)-546700= (1000000-1)-546700= 999999-546700

      = ***453299***
    - » 1's complement of N=***101101***

      (2^6-1)-101101= (1000000-1)-101101= 111111-101101

      = ***010010***

N : given number
r : base
n : digit number

546700(N) + 453299(9's com)
=999999

101101(N) + 010010(1's com)
=111111

- r's Complement
  - r's Complement of N = r^n-N
    - » 10's complement of ***2389***= 7610+1= ***7611***
    - » 2's complement of ***1101100***= 0010011+1= ***0010100***

* ***r's Complement***
(r-1)'s Complement +1 =(r^n-1)-N+1= r^n-N

Subtraction of Unsigned Numbers

Subtraction of Unsigned Numbers — $(M\text{-}N),\ N \neq 0$

- 1) $M + (r^n\text{-}N)$
- 2) $M \geq N$: Discard end carry, Result = M-N
- 3) $M < N$: No end carry, Result = - (N-M) = r's complement of (N-M)

  = - r's complement of (M-N)

Example

» *Decimal Example)*

M ≥ N

72532(M) - 13250(N) = 59282

72532

+ *86750* (10's complement of 13250)

Discard End Carry

1 59282

Result = *59282*

M < N

13250(M) - 72532(N) = -59282

13250

+ *27468* (10's complement of 72532)

No End Carry

0 40718

Result = -(10's complement of 40718)

= -(59281+1) = *-59282*

» *Binary Example)*

X ≥ Y

1010100(X) - 1000011(Y) = 0010001

1010100

+ *0111101* (2's complement of 1000011)

1 0010001

Result = *0010001*

X < Y

1000011(X) - 1010100(Y) = -0010001

1000011

+ *0101100* (2's complement of 1010100)

0 1101111

Result = -(2's complement of 1101111)

= -(0010000+1) = *-0010001*