CMPS 10 11-3-09

NUMBER BASE CONVERSIONS:

EX. base 2 to base 10

$$[10110011]_2$$

$$= 1\cdot 2^7 + 0\cdot 2^6 + 1\cdot 2^5 + 1\cdot 2^4 + 0\cdot 2^3 + 0\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0$$

$$= 128 + 32 + 16 + 2 + 1$$

$$= [179]_{10}$$

Ex base 8 (octal) to base 10 (decimal

$$[4\,3\,2]_8$$

$$= 4 \cdot 8^2 + 3 \cdot 8^1 + 2 \cdot 8^0$$

$$= 4 \cdot 64 + 24 + 2 = 256 + 26 = [282]_{10}$$

Ex. octal to decimal.

$$[1\,0\,3\,7]_8 = 1 \cdot 8^3 + 0 \cdot 8^2 + 3 \cdot 8^1 + 7 \cdot 8^0$$

$$= 512 + 24 + 7 = [543]_{10}$$

Ex. hex to dec.

$$[ABC]_{16} = 10 \cdot 16^2 + 11 \cdot 16^1 + 12 \cdot 16^0$$

$$= 10 \cdot 256 + 11 \cdot 16 + 12$$

$$= 2560 + 176 + 12$$

$$= 2736 + 12 = [2748]_{10} \checkmark$$

Ex. decimal to binary

$$[357]_{10} = [101100101]_2$$

Powers of 2:

k	0 ✓	1	2 ✓	3	4	5 ✓	6 ✓	7	8 ✓	9
2^k	1	2	4	8	16	32	64	128	256	512

$$[357]_{10} = 256 + 101$$

$$= 256 + 64 + 37$$

$$= 256 + 64 + 32 + 5$$

$$= 256 + 64 + 32 + 4 + 1$$

$$= 2^8 + 2^6 + 2^5 + 2^2 + 2^0$$

$$= 1 \cdot 2^8 + 0 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 +$$
$$+ 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

$$= [1\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1]_2$$

$$= [545]_8$$

$$= [165]_{16}$$

Ex. binary to octal

$$[\underbrace{101}_{5}\,\underbrace{100}_{4}\,\underbrace{101}_{5}]_2 = [545]_8$$

$0 = 000$

$1 = 001$

$2 = 010$

$3 = 011$

$4 = 100$

$5 = 101$

$6 = 110$

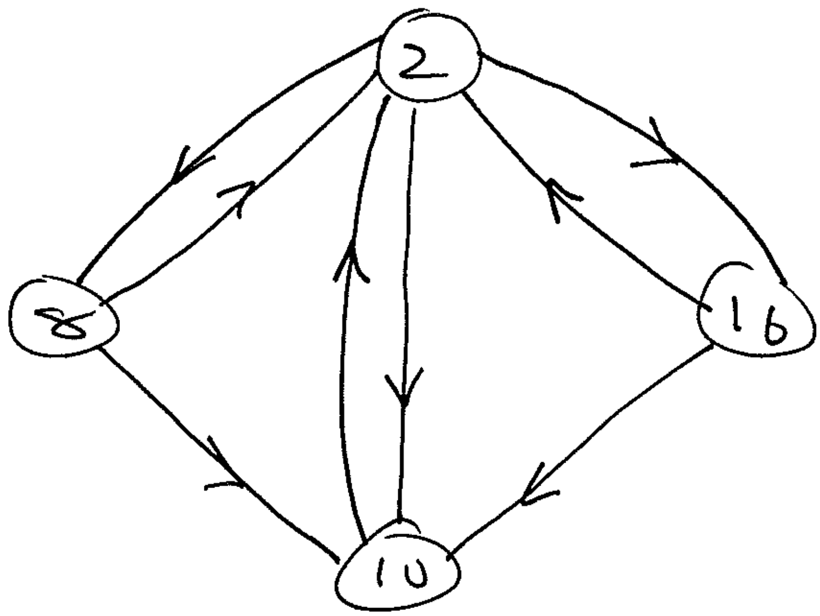
$7 = 111$

<u>Ex.</u> binary to hexadecimal

$$[\underbrace{0001}_{1}\underbrace{0110}_{6}\underbrace{0101}_{5}]_2 = [165]_{16}$$

0	=	0000
1	=	0001
2	=	0010
3	=	0011
4	=	0100
5	=	0101
6	=	0110
7	=	0111
8	=	1000
9	=	1001
A	=	1010
B	=	1011
C	=	1100
D	=	1101
E	=	1110
F	=	1111

number base conversions



what about fractions ?

Ex. $[12.75]_{10}$

$$12.75 = 8+4+.5+.25$$

$$= 1\cdot 2^3 + 1\cdot 2^2 + 0\cdot 2^1 + 0\cdot 2^0 + 1\cdot 2^{-1} + 1\cdot 2^{-2}$$

$$= [1100.11]_2$$

Ex. $[15.375]_{10}$

$$15.375 = 8+4+2+1+.25+.125$$

$$= 1\cdot 2^3 + 1\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0 + 0\cdot 2^{-1} + 1\cdot 2^{-2} + 1\cdot 2^{-3}$$

$$= [1111.011]_2$$

note add

$$\begin{array}{rrl}
 & 1111\ 1111\ 1111\ 111\ \ & \\
 & 1111\ 1111\ 1111\ 1111 & = x \\
+ & 0000\ 0000\ 0000\ 0001 & = 1 \\
\hline
 & 1\ 0000\ 0000\ 0000\ 0000 & = 2^{16}
\end{array}$$

$$x+1 = 2^{16}$$

$$x = 2^{16} - 1$$

Ex Suppose 16 bits are available for storage of a non-negative int.

maximum such int is

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

$= 2^{16} - 1 = 65535$

min such int is

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

$= 0$

Range is: 0 TO 65535

OF #S is: $2^{16} = 65536$

Ex. Suppose 32 bits are available for storage of non-negative (i.e. unsigned) ints.

$$\max = \overbrace{11\cdots\cdots1}^{32} = 2^{32}-1$$
$$= 4\,294\,967\,295$$
$$\min = \underbrace{00\cdots0} = 0$$

Range: 0 to $2^{32}-1$

of #s: 2^{32}

In General

$$\{\#\text{ of bit strings of length } n\} = 2^n$$

why?

$$\underbrace{\underline{\overset{0}{\underset{1}{\text{or}}}}\ \underline{\overset{0}{\underset{1}{\text{or}}}}\ \underline{\overset{0}{\underset{1}{\text{or}}}}\ \underline{\overset{0}{\underset{1}{\text{or}}}}\ \cdots\cdots\ \underline{\overset{0}{\underset{1}{\text{or}}}}}_{n \text{ bits to fill}}$$

\# ways of constructing such a bit str.

$$2\cdot2\cdot2\cdot2 \cdots 2 = 2^n$$