Number Systems

Introduction / Number Systems

Data Representation

- Data representation can be *Digital* or *Analog*
- In Analog representation values are represented over a *continuous* range
- In Digital representation values are represented over a *discrete* range
- Digital representation can be
  - Decimal
  - Binary
  - Octal
  - Hexadecimal

We need to know how to use and convert from one to another!

Using Binary Representation

- Digital systems are binary-based
  - All symbols are represented in binary format
  - Each symbol is represented using Bits
  - A bit can be one or zero (on or off state)
- Comparing Binary and Decimal systems:
  - In Decimal a digit is [0-9] – base-**10**
  - In Binary a digit is [0-1] – base-**2**
  - In Decimal two digits can represent [0-99] → $\mathbf{10^2-1}$
  - In Binary two digits can represent [0-3] → $\mathbf{2^2-1}$

Binary Counting

2^3	2^2	2^1	2^0	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5

●

2^{-1}	2^{-2}	2^{-3}	2^{-4}	
0	0	0	0	0
1	0	0	0	0.5
0	1	0	0	0.25
0	0	1	0	0.125
0	1	1	0	0.375
1	0	1	0	0.625

Binary Representation

1	0	0	1	●	1	0	1

Decimal Representation

9.625

Binary Point

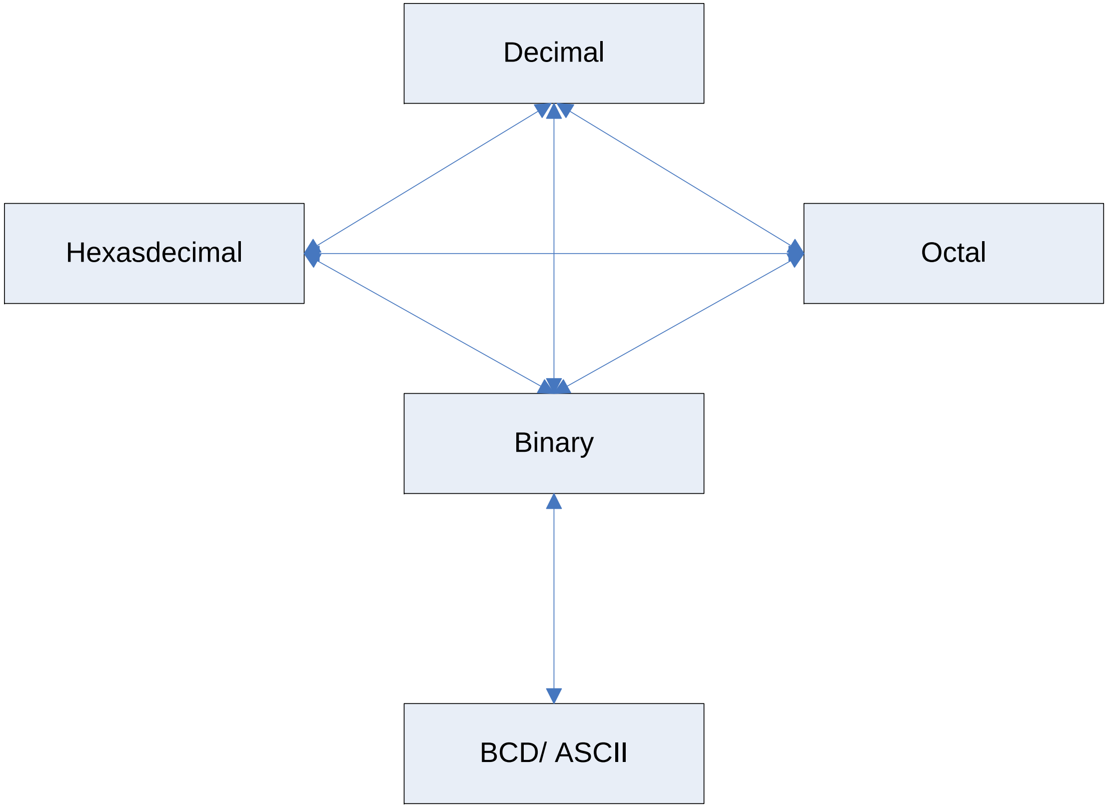
Counting in Different Numbering Systems

Demonstrating different number base or radix

- Decimal
  - 0,1,2,3,4,5,6,7,8,9,10,11,12…,19,20,21,…,29,30,…,39….
- Binary
  - 0,1,10,11,100,101,110,111,1000,….
- Octal
  - 0,1,2,3,4,5,6,7,10,11,12…,17,20,21,22,23…,27,30,…
- Hexadecimal
  - 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,10,…,1F,20,…,2F,30,.….

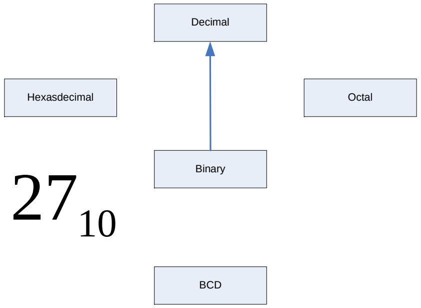
Remember: **aa.bb**
aa is whole number portion
bb is fractional portion
"." is the radix point

Learning Number Conversion



Binary-to-Decimal Conversions

11011 → Decimal



$$1\times 2^4 + 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 1\times 2^0 = 16 + 8 + 2 + 1 = 27_{10}$$

$$(n_{N-1}n_{N-2}\ldots n_3 n_2 n_1 n_0)_b \xrightarrow{Convert} n_0\times b^0 + n_1\times b^1 + n_2\times b^2 + n_3\times b^3 \ldots + n_{N-1}\times b^{N-1}$$

In the above example:
Binary is base-2 (b=2)
$n_0=1$
$n_1=1$
$n_2=0$
$n_3=1$
$n_4=1$

Q:	What is 11011.11 In Decimal?
Ans:	$=27+(1x2^{-1}+ 1x2^{-2})$ $=27+0.5+0.25$ $=27.75$

Decimal-to-Binary Conversions

Quotient + Remainder

$65/2 = 32 + Remainder\_of\_1$

$32/2 = 16 + Remainder\_of\_0$

$16/2 = 8 + Remainder\_of\_0$

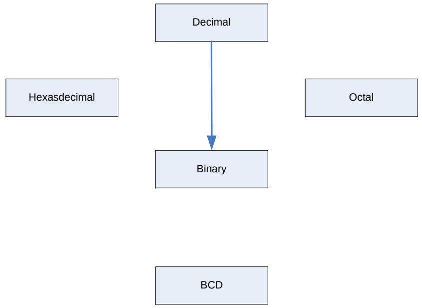
$8/2 = 4 + Remainder\_of\_0$

$4/2 = 2 + Remainder\_of\_0$

$2/2 = 1 + Remainder\_of\_0$

$1/2 = 0 + Remainder\_of\_1$

65 → Binary



1000001

MSB LSB

1- Save the remainder
2- Continue until Quotient = 0

Last one should be "0"

What if you are using a calculator?

65/2 = 32.5
0.5 x **2 (base-2)** = 1

LSB = Least Significant Bit
MSB= Most Significant Bit

Decimal-to-Binary Conversions

0.125 → Binary

Radix point +The whole portion + The fractional portion

$$0.125*2 = 0 + fractional\_portion\_of\_0.25$$

$$0.25*2 = 0 + fractional\_portion\_of\_0.5$$

$$0.5*2 = 1 + fractional\_portion\_of\_0$$

0.001

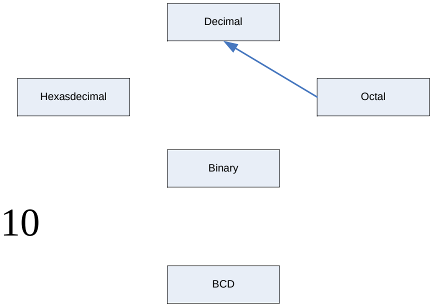
LSB

Last one should be "1"

Octal/Decimal Conversions

Binary is base-8 (b=8)

372 → Decimal



$$3\times 8^2 + 7\times 8^1 + 2\times 8^0 = 3\times 64 + 7\times 8 + 2\times 1 = 250_{10}$$

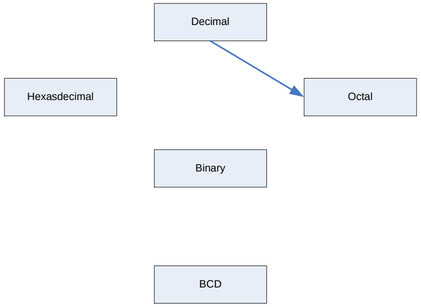
$$(n_{N-1}n_{N-2}...n_3n_2n_1n_0)_b \xrightarrow{Convert} n_0\times b^0 + n_1\times b^1 + n_2\times b^2 + n_3\times b^3 ... + n_{N-1}\times b^{N-1}$$

What about 372.2_8?

Ans: $=250+(2\text{x}8^{-1})=250.25$

Decimal-to-Octal:

266 → Octal



$$266/8 = 33 + Remainder\_of\_2$$

$$33/8 = 4 + Remainder\_of\_1$$

$$4/8 = 0 + Remainder\_of\_4$$

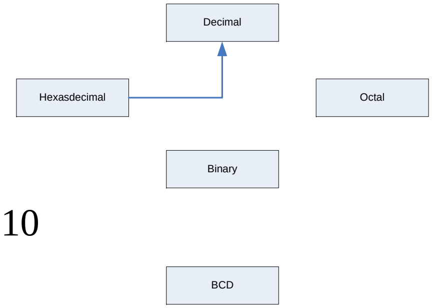
412

MSB LSB

Hex-to-Decimal Conversions

2AF → Decimal

Binary is base-16 (b=16)



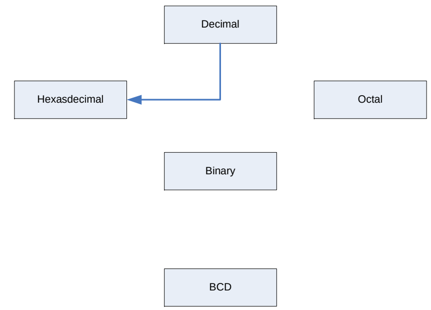
$$2 \times 16^2 + 10 \times 16^1 + 15 \times 16^0 = 512 + 160 + 16 = 687_{10}$$

$$(n_{N-1} n_{N-2} \ldots n_3 n_2 n_1 n_0)_b \xrightarrow{Convert} n_0 \times b^0 + n_1 \times b^1 + n_2 \times b^2 + n_3 \times b^3 \ldots + n_{N-1} \times b^{N-1}$$

Remember A=10, F=15

In the above example:
Binary is base-16 (b=16)
n_0=F which is 15
n_1=A which is 10
n_2=2

Decimal-to-Hex Conversions



423→ Hex

$$423/16 = 26 + \mathrm{Re}\,mainder\_of\_7$$
$$26/16 = 1 + \mathrm{Re}\,mainder\_of\_10$$
$$1/16 = 0 + \mathrm{Re}\,mainder\_of\_1$$

1A7

MSB LSB

Remember 10 in Hex is A

214→ Hex

$$214/16 = 13 + \mathrm{Re}\,mainder\_of\_6$$
$$13/16 = 0 + \mathrm{Re}\,mainder\_of\_13$$

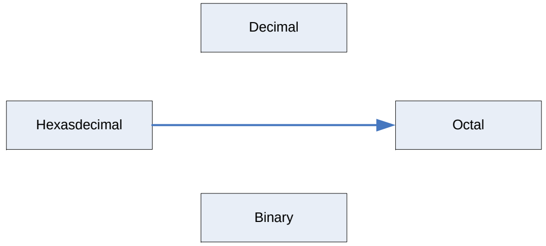
D6

Last one should be “0”

Remember 13 in Hex is D

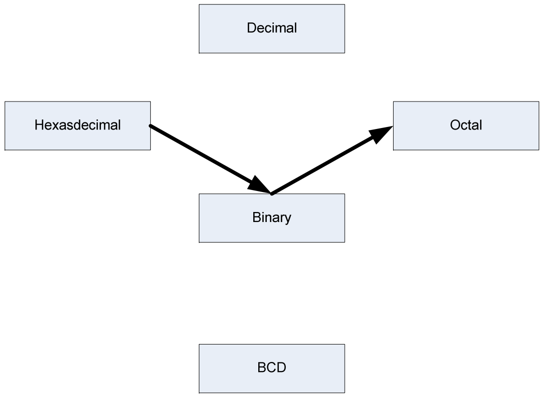
Converting from Hex-to-Octal

$124_{Hex} \rightarrow ??_{Oct}$



Always convert to Binary first and then from binary to Oct.

Ans: =124 Hex
= 0001 0010 0100
= 000 100 100 100
=0 4 4 4
=444 Oct



Counting

- Decimal
  - 0,1,2,3,4,5,6,7,8,9,10,11,12…,19,20,21,…,29,30,…,39….
- Binary
  - 0,1,10,11,100,101,110,111,1000,….
- Octal
  - 0,1,2,3,4,5,6,7,10,11,12…,17,20,21,22,23…,27,30,…
- Hexadecimal
  - 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,10,…,1F,20,…,2F,30,..…

Converting to BCD and ASCII

- We use Hex and Octal numbers to simplify number representation
- Any symbol can be represented by a *code*
  - Example: *American Standard Code for Information Interchange* (ASCII)
    - Each symbol is represented by a seven-bit code (How many symbols can be represented? – 127)
    - Example: A=100 0001 = 41 in Hex, 1=011 0000 = 31 in Hex, $=010 0100 = 24 in Hex (What is “DAD” in ASCII?)

Look at the ASCII code listing – Don’t memorize!

Converting to BCD and ASCII

- We use Hex and Octal numbers to simplify number representation
- Any symbol can be represented by a *code*
  - Example: *Binary-Coded-decimal* (BCD)
    - Each **digit** has its own binary code
    - Example: 6_{10}=0110, 16_{10}=0001 0110 (In binary 16 is?)
  - BCD can be packed or unpacked
    - 12→ Packed=0001 0010 ; unpacked=0000 0001 0000 0010

Terminologies

- BYTE
  - *8 bits is equivalent to one byte*
- NIBBLE
  - *4 bits is equivalent to one nibble*
- WORD
  - *16 bits is equivalent to one word*

1. ***128 bits is equivalent to how many bytes?*** 128/8 = 16
2. ***What is the maximum number that can be represented by 1 bytes?*** $2^8 - 1 = 255$

Switch State

In each case we have 16 switches.
1- What Binary/Decimal/Hexadecimal number does each switch represent?
2- What is the maximum binary number we can represent using these switches?

0000000100100011=291= 123

0100010101100111=17767=4567

1000100110101011=35243=89AB

1100110111101111=52719=CDEF

Maximum number: 2^{16}-1=65536-1=65535=64K in Computer terms!