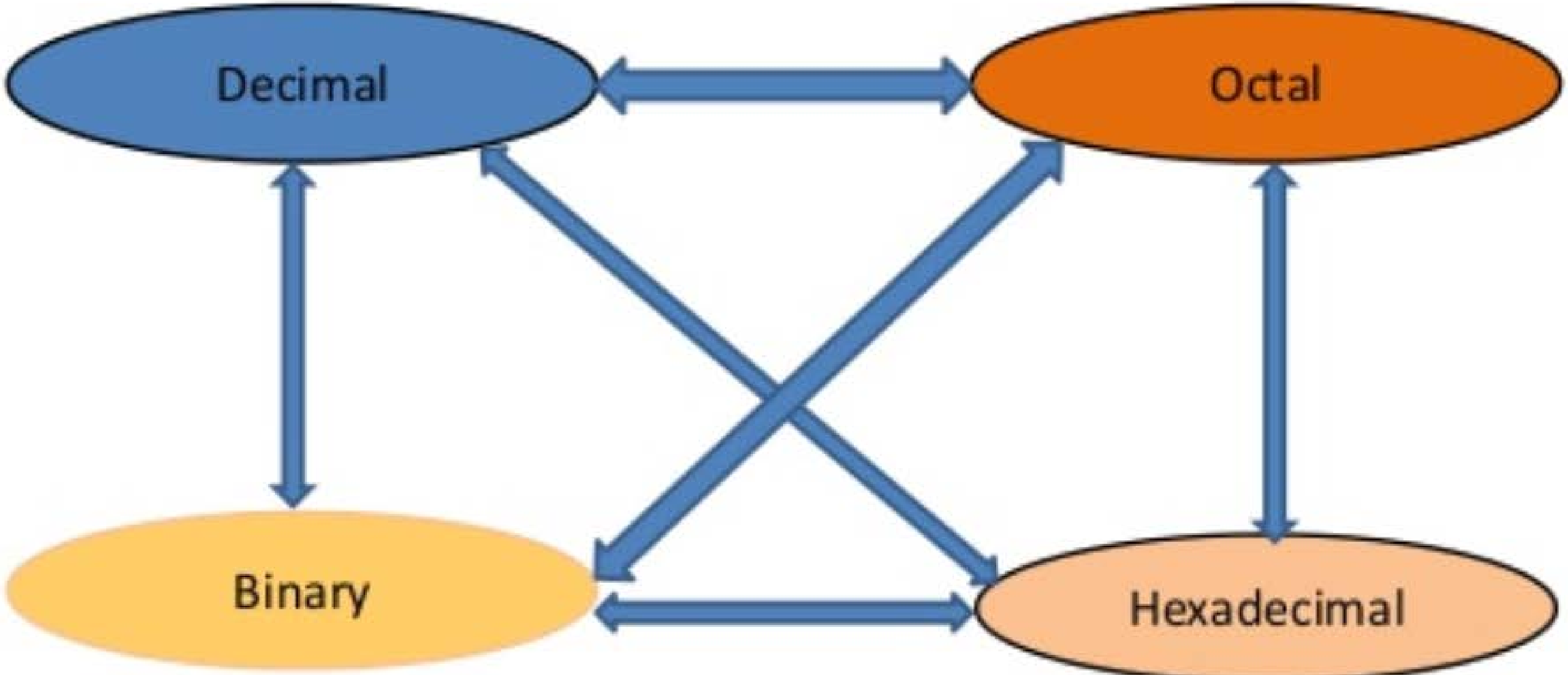
Number-Base Conversions

Conversion Among Bases

- The possibilities:



Binary –to– Decimal Process

The Process : *Weighted Multiplication*

a) Multiply each bit of the *Binary Number* by it corresponding bit-weighting factor (i.e. Bit-0→2^0=1; Bit-1→2^1=2; Bit-2→2^2=4; etc).

b) Sum up all the products in step (a) to get the *Decimal Number*.

Example:

Convert the decimal number 0110_2 into its decimal equivalent.

$$\begin{array}{ccccccccc} 0 & & 1 & & 1 & & 0 & & \\ 2^3 & & 2^2 & & 2^1 & & 2^0 & & \\ 8 & & 4 & & 2 & & 1 & & \\ 0 & + & 4 & + & 2 & + & 0 & = & 6_{10} \end{array}$$

Bit-Weighting Factors

$\therefore\ 0110_2 = 6_{10}$

Binary → Dec : Example #1

Example:

Convert the binary number 10010_2 into its decimal equivalent.

Binary → Dec : Example #1

Example:

Convert the binary number 10010_2 into its decimal equivalent.

Solution:

1		0		0		1		0		
2^4		2^3		2^2		2^1		2^0		
16		**8**		**4**		**2**		**1**		
16	**+**	**0**	**+**	**0**	**+**	**2**	**+**	**0**	=	18_{10}

$$\boxed{\therefore 10010_2 = 18_{10}}$$

Binary → Dec : Example #2

Example:

Convert the binary number 0110101_2 into its decimal equivalent.

Binary → Dec : Example #2

Example:

Convert the binary number 0110101_2 into its decimal equivalent.

Solution:

0	1	1	0	1	0	1		
2^6	2^5	2^4	2^3	2^2	2^1	2^0		
64	32	16	8	4	2	1		

$$0 + 32 + 16 + 0 + 4 + 0 + 1 = 53_{10}$$

$$\therefore 0110101_2 = 53_{10}$$

Binary → Dec : More Examples

a) $0110_2 = ?$

b) $11010_2 = ?$

c) $0110101_2 = ?$

d) $11010011_2 = ?$

Binary → Dec : More Examples

a) $0110_2 = ?$ 6_{10}

b) $11010_2 = ?$ 26_{10}

c) $0110101_2 = ?$ 53_{10}

d) $11010011_2 = ?$ 211_{10}

Binary → Dec : Fraction

Example:

Convert the binary number 0.1101_2 into its decimal equivalent.

Solution:

	1/2	1/4	1/8	1/16
	---	---	---	----
.	1	1	0	1

Just as with integers, we multiply each digit by its place value and add the results. For my example, we'd get:

$$
\begin{aligned}
&1 * .5 = .5 \\
+&1 * .25 = .25 \\
+&0 * .125 = .0 \\
+&1 * .0625 = .0625
\end{aligned}
$$

$$\boxed{0.1101_2 = 0.8125_{10}}$$

Binary → Dec : Fraction

Example:

Convert the binary number 0.0001_2 into its decimal equivalent.

Binary → Dec : Fraction

Example:

Convert the binary number 0.0001_2 into its decimal equivalent.

0 *.5= 0

0 * .25=0

0 *.125=0

1 *.0625 = .0625

$\boxed{0.0001_2 = 0.0625_{10}}$

Range of Numbers

n = no. of bits

No. of positions = 2^n

Maximum no. $=2^n-1$

With 8 bits no. of positions = $2^8 = 256$

With 8 bits maximum no. = $2^8-1 = 255$

ex: 1 1 1 1 1 1 1 1

128+64+32+16+8+4+2+1 = 255

Range of Numbers

How many bits do we need to write $(512)_{10}$ in binary?

Range of Numbers

How many bits do we need to write $(512)_{10}$ in binary?

We need 10 bits to write $(512)_{10}$ in binary

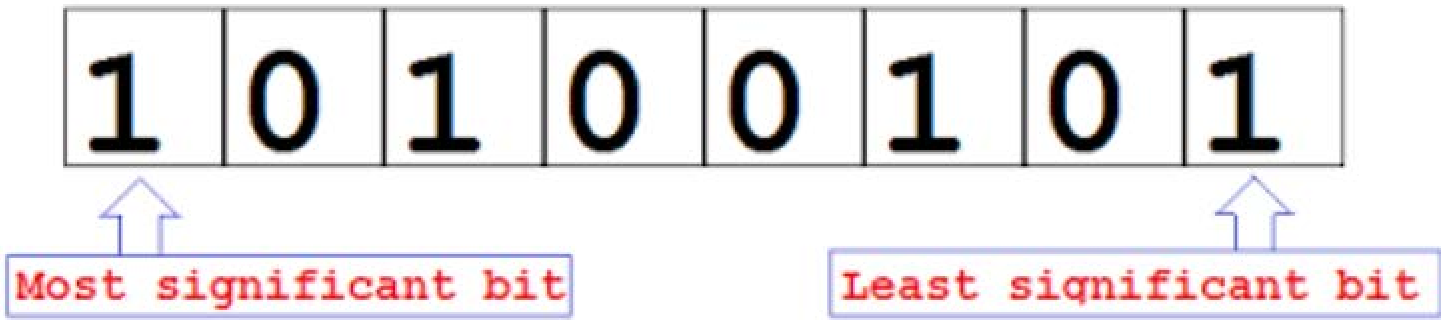
2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	0	0	0	0	0	0	0	0

How can you tell only looking at a binary number that it is odd or even?

How can you tell only looking at a binary number it is odd or even?

If the LSB is 0 the number is even otherwise it's odd

MSB &LSB



Binary → Dec : Fraction

Example:

Convert the binary number 0.0001_2 into its decimal equivalent.

Solution:

1/2	1/4	1/8	1/16
---	---	---	----
. 0	0	0	1

Just as with integers, we multiply each digit by its place value and add the results. For my example, we'd get:

0 * .5 =.0
+0 * .25 = .0
+ 0 * .125 = .0
+1 * .0625 = .0625

$0.0001_2 = 0.0625_{10}$

Decimal –to– Binary Conversion

The Process : *Successive Division*

a) Divide the *Decimal Number* by 2; the remainder is the LSB of *Binary Number* .

b) If the quotient is zero, the conversion is complete; else repeat step (a) using the quotient as the Decimal Number. The new remainder is the next most significant bit of the *Binary Number.*

Example:

Convert the decimal number 6_{10} into its binary equivalent.

$$2\,\overline{)\,6\,}^{\,3} \quad r = 0 \leftarrow \text{Least Significant Bit}$$

$$2\,\overline{)\,3\,}^{\,1} \quad r = 1$$

$$2\,\overline{)\,1\,}^{\,0} \quad r = 1 \leftarrow \text{Most Significant Bit}$$

$$\boxed{\therefore\; 6_{10} = 110_2}$$

Dec → Binary : Example #1

Example:

Convert the decimal number 26_{10} into its binary equivalent.

Dec → Binary : Example #1

Example:

Convert the decimal number 26_{10} into its binary equivalent.

Solution:

$$2\overline{\smash{)}26}^{\;13} \quad r = 0 \leftarrow \text{LSB}$$

$$2\overline{\smash{)}13}^{\;6} \quad r = 1$$

$$2\overline{\smash{)}6}^{\;3} \quad r = 0$$

$$2\overline{\smash{)}3}^{\;1} \quad r = 1$$

$$2\overline{\smash{)}1}^{\;0} \quad r = 1 \leftarrow \text{MSB}$$

$$\boxed{\therefore\ 26_{10} = 11010_2}$$

Dec → Binary : Example #2

Example:

Convert the decimal number 41_{10} into its binary equivalent.

Dec → Binary : Example #2

Example:

Convert the decimal number 41_{10} into its binary equivalent.

Solution:

$$2\,\overline{\smash{)}41}^{\;20} \quad r = 1 \leftarrow \text{LSB}$$

$$2\,\overline{\smash{)}20}^{\;10} \quad r = 0$$

$$2\,\overline{\smash{)}10}^{\;5} \quad r = 0$$

$$2\,\overline{\smash{)}5}^{\;2} \quad r = 1$$

$$2\,\overline{\smash{)}2}^{\;1} \quad r = 0$$

$$2\,\overline{\smash{)}1}^{\;0} \quad r = 1 \leftarrow \text{MSB}$$

$$\boxed{\therefore\ 41_{10} = 101001_2}$$

Dec $\rightarrow$ Binary : More Examples

a) $13_{10} = ?$

b) $22_{10} = ?$

c) $43_{10} = ?$

d) $158_{10} = ?$

Dec → Binary : More Examples

a) 13_{10} = ? $1\,1\,0\,1_{\,2}$

b) 22_{10} = ? $1\,0\,1\,1\,0_{\,2}$

c) 43_{10} = ? $1\,0\,1\,0\,1\,1_{\,2}$

d) 158_{10} = ? $1\,0\,0\,1\,1\,1\,1\,0_{\,2}$

Dec → Binary : Fraction Examples

Example:

Convert the decimal number $(0.6875)_{10}$ into its binary equivalent.

	Integer		Fraction
$0.6875 \times 2 =$	1	+	0.3750
$0.3750 \times 2 =$	0	+	0.7500
$0.7500 \times 2 =$	1	+	0.5000
$0.5000 \times 2 =$	1	+	0.0000

$$\boxed{(0.6875)_{10} = (0.1011)_2.}$$

Octal and Hexadecimal Numbers

Octal Number

- Base or radix 8 .
- 1 octal digit is equivalent to 3 bits.
- Octal numbers are 0 to7.
- Numbers are expressed as powers of 8.

Octal Number

Decimal	Octal
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	10
9	11
10	12
11	13
12	14
13	15
14	16
15	17

Binary-Octal

Step 1: Group the binary digits by threes, separate them by space

110011001100_2	becomes	$110\ 011\ 001\ 100_2$
11110000_2	becomes	$11\ 110\ 000_2$

Step 2: Add zeros to the left (most significant digits) to make all numbers 3 bit numbers:

$11\ 110\ 000_2$	becomes	$011\ 110\ 000_2$

Step 3: Now convert each three bit number into a octal number: 0..7

$110\ 011\ 001\ 100_2$	becomes	6314_8
$011\ 110\ 000_2$	becomes	360_8

Binary -> Octal	Octal -> Binary
01101001=	264_8=
10101010=	701_8=
11000011=	076_8=
10100101=	567_8=

Binary -> Octal	Octal -> Binary
01101001= 001 101 001 = 151	264_8= 010 110 100
10101010= 010 101 010) = 252	701_8= 111 000 001
11000011= 011 000 011 = 303	076_8= 000 111 110
10100101= 010 100 101= 245	567_8= 101 110 111

Hexadecimal Number

- Base or radix 16 number system.
- 1 hex digit is equivalent to 4 bits. $2^4 = 16$
-
- Numbers are 0,1,2…..8,9, A, B, C, D, E, F.
- B is 11, E is 14
- Numbers are expressed as powers of 16.

Hexadecimal Number

Decimal	Hexadecimal
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F

Signed Number

Sign-Magnitude Notation

The simplest form of representation that employs a sign bit, the leftmost significant bit. For an N-bit word, the rightmost N-1 bits hold the magnitude of the integer. Thus,

- 00010010 = +18
- 10010010 = -18

2's Complement Notation

Method:

Step1: Form the regular binary representation to the required number of bits.

Ste2: Invert it.

Step3: Add a 1.

Ex1: We need to write -10 with 8 bits

Step1: First write +10 with 8 bits: 0 0 0 0 1 0 10

Step2: Invert it: 1 1 1 1 0 1 0 1

Step3: Add a 1: 1 1 1 1 0 1 0 1

+1

1 1 1 1 0 1 1 0

Gray Code

- Gray coding is used for its speed & freedom from errors.
- In BCD or 8421 BCD when counting from 7 (0111) to 8 (1000) requires 4 bits to be changed simultaneously.
- If this does not happen then various numbers could be momentarily generated during the transition so creating spurious numbers which could be read.
- Gray coding avoids this since only one bit changes between subsequent numbers. Two simple rules.
  - 1. Start with all 0s.
  - 2. Proceed by changing the least significant bit (lsb) which will bring about a new state.

Gray Code Continued

Decimal	Gray Code
0	0000
1	0001
2	0011
3	0010
4	0110
5	0111
6	0101
7	0100
8	1100
9	1101
10	1111
11	1110
12	1010
13	1011
14	1001

ASCII Code

- **ASCII:** American Standard Code for Information Interchange
- The standard ASCII code defines 128 character codes (from 0 to 127), of which, the first 32 are control codes (non-printable), and the other 96 are representable characters.
- In addition to the 128 standard ASCII codes there are other 128 that are known as extended ASCII, and that are platform-dependent.

Table 1.7
American Standard Code for Information Interchange (ASCII)

	$b_7b_6b_5$							
$b_4b_3b_2b_1$	**000**	**001**	**010**	**011**	**100**	**101**	**110**	**111**
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	“	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	‘	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	–	=	M	]	m	}
1110	SO	RS	.	>	N	∧	n	~
1111	SI	US	/	?	O	—	o	DEL

Decode the following ASCII code:
1010011 1110100 1100101 1110110 1100101 0100000 1001010 1101111 1100010 1110011.

Decode the following ASCII code:
1010011 1110100 1100101 1110110 1100101 0100000 1001010 1101111 1100010 1110011.

Steve Jobs