Numbering System.

- $(125)_{10} \xrightarrow{\text{magnitude}} 5\times10^0 + 2\times10^1 + 1\times10^2$

  5: least significent digit.

  1: most significent digit.

Conversion

- $(1010)_2 \rightsquigarrow (\ ?\ )_{10}$

  $0\times2^0 + 1\times2^1 + 0\times2^2 + 1\times2^3 = 10 \Rightarrow (10)_{10}$

- $(235)_8 \rightsquigarrow (\quad)_{10}$

  $5\times8^0 + 3\times8^1 + 2\times8^2 = 157 \Rightarrow (157)_{10}$

- $(153.5)_{10} \rightsquigarrow (\quad)_8$

8	153
1	19
3	2
2	0

$\rightsquigarrow 0.5\times8 = 4.0$

$\rightsquigarrow 0\times8 = 0.0$

$\Rightarrow (281.4)_8$

octal to Binary

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

hexa to Binary

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

- $(721)_8 \longrightarrow (111\ 010\ 001)_2$
- $(101\ 001.010)_2 \longrightarrow (51.2)_8$
- $(A1.8)_{16} \longrightarrow (1010\ 0001.1000)_2$

```
  101101
+ 100111
--------
 1010100
```

```
  0100
-    1
------
  0011
```

```
  11001010
- 10011011
----------
  00101111
```

مش مهم

complements of Numbers.

1) diminished radix complement.
2) radix complement.

Diminished radix complement

Given a number N in base r having n digits, the (r−1) complement of N $\Longrightarrow (r^n - 1) - N$.

- The 9's complement of 012898 is $\Rightarrow$ 999999 − 012398 = 987601
- The 1's complement of 1011000 is $\Rightarrow$ 0100111

Radix complement

- The r's complement of N is defined $\Rightarrow (r^n - N)$
- r comp = (r−1) comp + 1

What is the 16' comp of F0EF ?

$$
\begin{array}{r}
FFFF \\
F0EF \\
\hline
0F10 \\
+\ 1 \\
\hline
0F11
\end{array}
$$

Subtraction with complements

1) $M > N$

- there is an end carry (must be discarded)
- $M - N = M +$ r comp of N

unsigned
غ علامة الاشارة

2) $M < N$

- there is no end carry.
- $M - N = M +$ r's comp of N
- take the r's comp of answer.

- Using 10's complement, subtract $72532 - 3250$

$$
\begin{array}{rr}
 & M = 72532 \\
\oplus & \text{10 comp of N}\ \ 96750 \\
\hline
 & \cancel{1}69282 \\
 & 10000\ - \\
\hline
\text{Answer} \rightsquigarrow & 69282
\end{array}
$$

- Using 10's complement, subtract 3250 − 72532

$$\begin{array}{r} 99999 \\ -\ 72532 \\ \hline 27467 \\ +\ 1 \\ \hline 27468 \end{array}$$

M = 03250

10 comp of N 27468

$$\begin{array}{r} 03250 \\ +\ 27468 \\ \hline 30718 \end{array}$$

10 comp of

$$\begin{array}{r} 99999 \\ -\ 30718 \\ \hline 69281 \\ +\ 1 \\ \hline 69282 \end{array}$$

(−) 69282

Signed Binary Numbers

- there is one way to represent the positive numbers
  Signed Magnitude

- there are three ways to represent negative numbers.
  1) Signed magnitude
  2) First complement
  3) Two complement

0 → positive
1 → negative

01001 → 9 (unsigned), +9 (signed)

11001 → 25 (unsigned), −9 (signed)

+9 $\xrightarrow{8\text{ bit}}$ 00001001

−9:
- signed mag : 10001001
- signed 1's comp : 11110110
- signed 2's comp : 11110111

Arithmetic Addition

- if the signs are the same, we add the two magnitudes and give the sum the common sign.
- if the signs are the different, we subtract the smaller magnitude from the larger and give the difference the sign of the larger magnitude.

a carry out of the sign bit position is discarded

```
 -6    11 111010
+13    00 001101
----------------
 +7    00000111
```

A = -15, B = 25 signed two's comp in 6 bit representation

```
+15 → 001111          +25 → 011001
-15 → 110000          -25 → 100110
          1+                    1+
      ------                ------
      110001                100111
```

```
A + B          110001
-15 + 25       011001 +
               -------
+10           ~~1~~001010
```

```
2|10
 0| 5
 1| 2
 0| 1
 1| 0
```

```
A - B          110001
-15 - 25       100111 +
               -------
-40            1011000  ← (overflow)
```

```
2|40
 0|20   → +40   0 101000   ⇒ overflow
 0|10
 0| 5     -40   1) 1101000
 1| 2           2) 1010111 +
 0| 1
 1| 0   -40 → 3) [1011000]
```

* Carry is important when

- adding or sub unsigned integers.
- indicates that the unsigned sum is out of range.

* Overflow is important when.

- adding or sub signed integers.
- Indicates that the signed sum is out of range.

```
 [1] 1 1 1 1                    [1] 0 0 0 1
      0 0 0 0 1 1 1 1                1 1 0 1 1 0 1 0
      1 1 1 1 1 0 0 0                1 0 0 1 1 1 0 1
      ---------------                ---------------
      0 0 0 0 0 1 1 1                0 1 1 1 0 1 1 1
```

carry: 1 overflow: 0

carry: 1
overflow: 1

```
 1 [0]
   0 1 0 0 1 1 1 1
   0 1 0 0 0 0 0 0
   ---------------
   1 0 0 0 1 1 1 1
```

carry: 0
overflow = 1

Binary codes decimal

Decimal	BCD
0	000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

10 XXXX
11 XXXX
12 XXXX
13 XXXX
14 XXXX
15 XXXX

1010
⋮
1111

- $13 \Leftrightarrow (0001\ 0011)_{BCD}$ This is coding.
- $13_{10} \Leftrightarrow (1101)_2$ This is conversion.

- coding required more bits than conversion

Binary to Gray Code

$B \rightsquigarrow G$

- 011 → 010

$S \rightarrow 0$

$D \rightarrow 1$

$B \rightarrow G$

$110 \rightarrow 101$

$B \rightarrow G$

$101011 \rightsquigarrow 111110$

Gray to Binary

$G \rightsquigarrow B$

110

100

$S \rightarrow 0$

$D \rightarrow 1$

$G \rightsquigarrow B$

110010

100011

- BCD مثل code 2421

- Excess-3

0 0011 — بتاخذ كود ال 3

1 0100

2

3

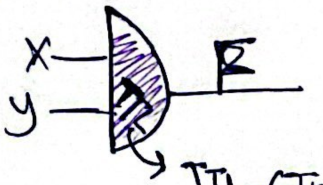
ASCII code

A → $(65)_H$ | a → $(97)_H$ | 0 → $(48)_H$
B → $(66)_H$ | b → $(98)_H$ | 1 → 49
⋮ | | ⋮
Z $(90)_H$ | | 9 → $(57)_H$

$(13)_d \xrightarrow{ASCII}$ (01001 001 01010001)

$(49)_H$ $(51)_H$

AND gate



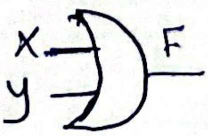
TTL (Transistor transistor level)

x	y	F
0	0	0
0	1	0
1	0	0
1	1	1

$F = x \cdot y$

Graphical symbol

OR gate

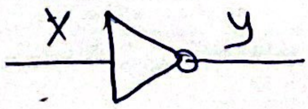


$F = x + y$

x	y	F
0	0	0
1	0	1
0	1	1
1	1	1

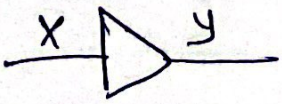
Inverter (NOT)

x	y
0	1
1	0



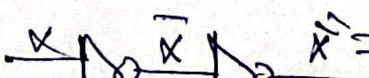
$y = \bar{x} = x' = x^c$

Buffer



x	y
0	0
1	1

$y = x$



$\bar{\bar{x}} = x$

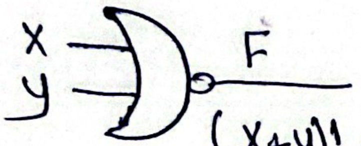
NAND gate (not AND)

$(x.y)'$

x	y	x.y	$\overline{x.y}$
0	0	0	1
1	0	0	1
0	1	0	1
1	1	1	0

$$F = (x.y)' = (\overline{x.y}) = (x.y)^c$$

NOR gate (Not OR)

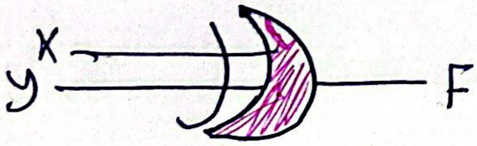


$(x+y)' = (\overline{x+y}) = (x+y)^c$

x	y	F
0	0	1
1	0	0
0	1	0
1	1	0

بال design يفضل استخدام NAND, NOR . ⇐ عدد الترانزستور فيها أقل.

XOR gate

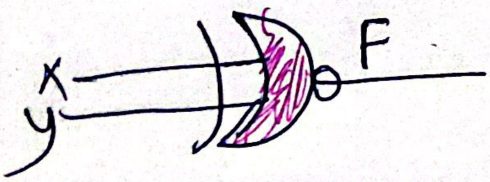


(odd function)

x	y	F
0	0	0
1	0	1
0	1	1
1	1	0

$F = x \oplus y$

XNOR gate (Not XOR)



(even function)

$F = (x \oplus y)^c$