Numbering System

{1} $10\,(\text{Decimal})_{10}$ 0, 1, 2, 3, 4, ..., 9

$(125)_{10}$ ⇝ magnitude

$5\times10^0 + 2\times10^1 + 1\times10^2$

1: most significant digit.

5: least significant digit.

{2} $\text{Binary}_{(2)}$ 0, 1

$(101)_{(2)}$ ⇝ magnitude

$1\times2^0 + 0\times2^1 + 1\times2^2$

1 ← least significant bit

1 ← Most significant bit

Note: bit ⇐ binary ← digit

{3} $\text{Octal}_{(8)}$ ⇒ 0, 1, 2, 3, 4, 5, 6, 7

$(517)_{(8)}$ ⇝ magnitude

$7\times8^0 + 1\times8^1 + 5\times8^2$

7 ⇐ least significant digit.

5 ⇐ Most significant digit.

4) Hexadecimal (16)

0 1 2 3 4 5 6 .... 9 A B C D E F

Conversion

* from system x to Decimal

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

1) Multiply each bit by 2^n where n is the weight starting from 0 on the left

2) Add result.

$0\times2^0+1\times2^1+0\times2^2+1\times2^3=10$

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

* $(1010.01)_2 \rightsquigarrow (?)_{10}$

$0\times2^0+1\times2^1+0\times2^2+1\times2^3$

$+ 0\times2^{-1}+1\times2^{-2} \Rightarrow 10.\frac{1}{4}$

$(1010.01)_2 \rightarrow (10.25)_{10}$

* $(235)_8 \rightsquigarrow (?)_{10}$

$5\times8^0+3\times8^1+2\times8^2 = 5+24+128$

$= 157$

$(235)_8 \rightarrow (157)_{10}$

* $(235.4)_8 \longrightarrow (?)_{10}$

$5\times8^0 + 3\times8^1 + 2\times8^2 + 4\times8^{-1}$ → $\frac{4}{8} = 0.5$

$= 157.5$

$(235.4)_8 \longrightarrow (157.5)_{10}$

* $(A1.8)_{16} \longrightarrow (?)_{10}$

$1\times16^0 + 10\times16^1 + 8\times16^{-1}$

$= 1 + 160 + \frac{8}{16}$

$(A1.8)_{16} \longrightarrow (161.5)_{10}$

* $(235)_x \longrightarrow (157)_{10}$

$5n^0 + 3n + 2n^2 = 157$

$3n + 2n^2 = 152$

$n(3 + 2n) = 152$

$2\times8^2 + 3\times8 = 128 + 24 = 152$

* $(41)_{10} \longrightarrow (?)_2$

2	41
1	20
0	10
0	5
1	2
0	1
1	0

$(41)_{10} \longrightarrow (101001)_2$

Note: 1 Kbit = 1024 bits

2^{10} bits

1 Kbyte = 1×2^{10} byte

$= 1\times2^{10}\times8$ bits

$= 1024\times8$ bits.

2^0	2^1	2^2	2^3	2^4	2^5	2^6	.....
1	2	4	8	16	32	64	128

$32 < 41 < 64$ → $32+8+1$ → $(2^5\ 2^3\ 2^0)$

2^0	2^1	2^2	2^3	2^4	2^5	2^6	يقلب واحد
1	0	0	1	0	1	0	

* $(65.4)_{10} \longrightarrow (\quad)_2$

2	65
1	32
0	16
0	8
0	4
0	2
0	1
1	0

$0.4 \times 2 = 0.8$
$0.8 \times 2 = 1.6$
$0.6 \times 2 = 1.2$
$0.2 \times 2 = 0.4$
$0.4 \times 2 = 0.8$

$(65.4)_{10} \longrightarrow (100\,000).\overline{0110})_2$

---

* $(153.5)_{10} \longrightarrow (\quad)_8$

8	153
1	19
3	2
2	0

$0.5 \times 8 = 4.0$
$0 \times 8 = 0.0$

$(153.5)_{10} \longrightarrow (231.4)_8$

From octal to Binary

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

* $(721)_8 \rightarrow (111\ 010\ 001)_2$
* $(101\ 001.010)_2 \rightarrow (?)_8$
  ↳ $(101\ 001.010)_2 \rightarrow (51.2)_8$

From Hex.16 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

محاضرة (٢) .

Note : الـdesign انا بعطيك الـproblem وانت بتعمل ديزاين السيركت لحل المشكلة.

الـAnalysis : انا بعطيك الـcircuit وانت بتحولها.

* $(A1.8)_{16} \rightarrow (1010\ 0001.1000)_2$
* $(\underbrace{1010}_{A}\underbrace{1011}_{B}.\underbrace{1}_{8})_2 \rightarrow (AB.8)_{16}$

Central processing unit (cpu) وحدة المعالجة المركزية

1. ALU : Arithmetic & logic unit (Design) (add / sub, Multi / comparison)
2. CU : control unit X (غير موجود)
3. Register : (memory) inside cpu.

There is no substraction operation

* 9+9=18 (add)
* 8-9 → no (-) in computer
* Substraction must be changed to addition (complement)
* complement : $M - N = M + \text{complement } N$

* There are two types of complements
  1) diminished complement.
  2) Radix complement.

- Diminished complement : Given a number N in base r having n digits.

  ↳ the $(r-1)$'s comp of N is $(r^n-1)-N$

addition 16+16 = 32 adder cs = 1
10000+10000 = 100000 بالكومبيوتر بنجمع اوتوماتيك

There is no sub in computer.

- What is the 9 comp of 666 in base 10?

$$(10^3-1)-666=833$$

- What is the first complement of 1011 in base 2?

$$(2^4-1)-1011$$
$$15-1011$$
$$1111-1011=0100$$

بنعوض اسفل 0 ب 1 و 1 ب 0

Radix complement : Given a number N in base r having n digit, the r's complement of N is defined $(r^n - N)$ for $N \neq 0$.

* What is the 10's comp of 666 ? (in base 10)

$10^3 - 666 = 334$

→ r's comp = (r-1) comp + 1

* What is the 16's comp of F0EF ? (in base 16)

16' comp = 15' comp + 1

- FFFF
- F0EF
- 0F10
- 14
- 0F11

16 comp of 234 = 15 comp of 234 + 1

- FFF
- 234
- DCB
- 1
- DCC

* What is the 2' comp of 1011 ? (base 2)

1st comp + 1 = 0100 + 1 = 0101

* (subtraction) with complement

1) M − N = M + r's complement N

if M > N result (+)

There is an end carry must be discarded

in base 10 ⇒ 999 − 665 = 999 + 10's comp 665

= 999 + 335

```
  1 1
  999
+ 335
-----
 1334
```

end carry M > N ⇒ (circled)

in base 2 ⇒ 1011 − 0111

1011 + 2's comp 0111

= 1011 + 1's comp + 1

1011 + 1000 + 1

```
  1 0
  1011
+ 1001
------
 10100
```

2) M < N There is no end carry, answer is negative to get the final answer take the r's comp of answer.

ex in base 10 666 − 999

666 + 10's comp 999

= 666 + 1 = 667 → negative

Final answer = 10's comp of 667

↳ (−) 333

```
  999
- 667
-----
  332
+   1
-----
  333
```

ex: in base 2 $0111 - 1011$

$0111 + 2'\text{comp } 1011$

Final answer = $2'\text{comp } 1100$

$0011 + 1$

$(-)\,0100$

$$\begin{array}{r} 0111 \\ 0101+ \\ \hline 1100 \end{array}$$

• Numbers

unsigned (غير سالبة) (pos كل الاعداد)

$M > N$

$M < N$

unsigned 3 bit numbers

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

signed (سالبة وموجبة)

signed 3 bit Numbers

8 numbers

(-) بدون (+) بدون

signed Binary Number

- there is no sign way in computer (−) (+)

left most bit ⟶ sign

(اخر اقصى اليسار)

اذا كانت الاشارة (−) الرقم 1

اذا كانت الاشارة (+) الرقم 0

Note: there is one way to represent the positive Numbers in computer (signed magnitude).

+7 ⇒ 0+ 111 (3bits)

signed magnitude

+15 ⇒ 0+ 1111 (5bits)

+9 (6bit) 0+ 01001

bit اضفنا واحد

هو الاشارة

+9 ⇒ 0 1001 (5bits)

هادي 9 لانه الاشارة

(+9) 4 bit overflow

محاضرة (٣)

* There are three ways to represent negative numbers.

1) signed magnitude
2) First ~~magnitude~~ complement
3) Two complement.

1) signed magnitude

-9 ⟶ (-) 1 1001 (9)

+9 ⟶ 0 1001 (+) 9

-7 using 3 bit ⟶ (-) (1) 111 (7)

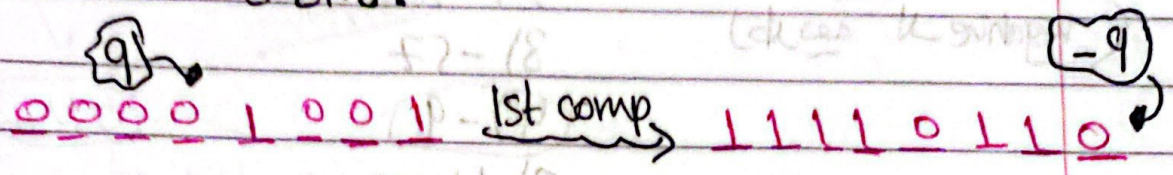
لسه لازم ٤ بت مش ٣ بالتالي overflow

overflow : لما الرقم بدو memory اكبر من الموجودة.

- $(1111)_2$ unsigned ⟶ $(15)_{10}$ ⟶ $1\times2^0+1\times2^1+1\times2^2+1\times2^3 = 15$
- $(1111)_2$ signed mag ⟶ $(-7)_{10}$

2) First complement

represent -9 using First comp using 8 bits.



$(1111\ 0110)_2$ First comp → $(0000\ 1001)$

1) 20
2) 30
3) -9
4) 60

(لما بنصحي هاي الطريقة غير مع الاسئلة)

3) Two complement

represent -9 using two' comp using 6 bits.

001001 2'comp → 110110

1+

110111

-9

Summary

1) signed magnitude (-9 6bits) ⇒ -(1)01001
2) (-9 1st comp 6bit) ⇒ 001001 → 110110
3) (-9 2nd comp) ⇒ 001001 → 110111

$(110111)_2^{2}$ Two comp (س) ⇒ (−) 001001+

two comp معرف ما مكتوب
يعني في الاشارة
(طريقة لـ negative)

1) 57
2) −20
3) −57
4) −9
5) None

A = −15, B = 25 do the following operation using signed two's comp in 6 bit representation.

+15 → 001111
−15 two comp → 110000
1+
110001 → عبارة عن −15

معرف ما مكتوب two comp يعني في الاشارة.

25 → 011001
−25 two comp → 100110
+1
100111

A + B
−15 + 25
110001 + 011001

1 1 0 0 0 1 1
110001
011001 +
1 001010

discarder (نحذفه) no overflow

الجواب 001010

A − B

−15 − 25

−40

```
  ①⓪⓪⓪①①⓪
    110001
    100111
  ─────────
  1011000
```

مختلفان كتا overflow

يمثل بـ 6 bit واللي مع الإشارة 7 bit بالتالي overflow

لكن الكمبيوتر حيوخذ هاد الشكل ، الكمبيوتر بس بيتعامل بال carry .

Binary codes → ترميز

language $2^2 = 4$

A B C D (SAlma code)

00 01 10 11

sender receiver

ABD ABD

000111 000111

language

A B C D 0 1 2 3 (Sami code)

000 001 010 011 100 101 110 111

ABD ABD

000 001011 000 001 011

ASCII code ⇒ only text.

(عدد ال Char اكبر من 128 واقل من 256) 128 < 200 < 256

2^7 2^8

لهيك لازمنا 8 bit

A $\xrightarrow{ASCII}$ $(65)_H$ → 01100101

0 $\xrightarrow{ASCII}$ $(48)_H$ → $(0100\ 1000)_2$

3 $\xrightarrow{ASCII}$ $(51)_H$ → $(0101\ 0001)_2$

1 → $(49)_H$ → $(0100\ 1001)_2$

* $(13)_d$ $\xrightarrow{conversion}$ $(\ \ )_2$

$(1101)_2$

* $(13)_d$ $\xrightarrow{ASCII}$ (01 0011 0101 0001)

2	13
1	6
0	3
1	1
1	0

BCD (Binary coded decimal)

↳ code valid for numbers from 0-9

Decimal	BCD
0	0000
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

- $(123)_d \xrightarrow{BCD} (0001\ 0010\ 0011)_{BCD}$
- $(16)_d \xrightarrow{BCD} (0001\ 0110)$
- $(16)_d \xrightarrow{conversion} (10000)$

Addition in BCD

In BCD

```
       1000
+ 4    0100
  4    0100
      ------
       1000   -> 8
```

valid, yes
اقل من 9

In BCD

```
       0 1 0   (carries)
+ 5    0101
  5    0101
      ------
       1010
```

↓

no valid

اذا طلع مش valid بنضيفله ٦
لحتى يصير ← valid

```
       1 1 0 0   (carries)
         1010
       + 0110     + 6
      --------
       1 0000
       1    0
```

In BCD

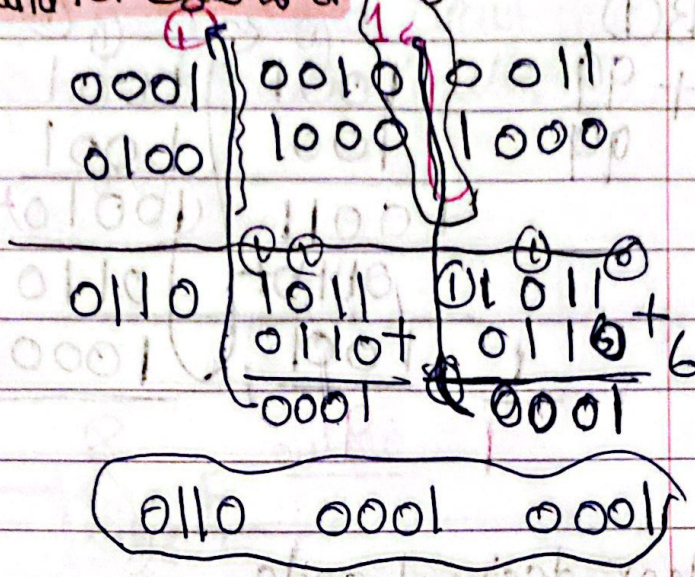
```
       1 1 0 0   (carries)
+ 6    0110
  6    0110
      ------
      01100
       0110      + 6
      ------
     1 0010
     1    2
```

اذا كان valid بنطلعه اذا 8 بنضيف 6 بالأول

سواء بزيد نطلع الـ carry هيك لا بما بنشوف اذا valid

In BCD

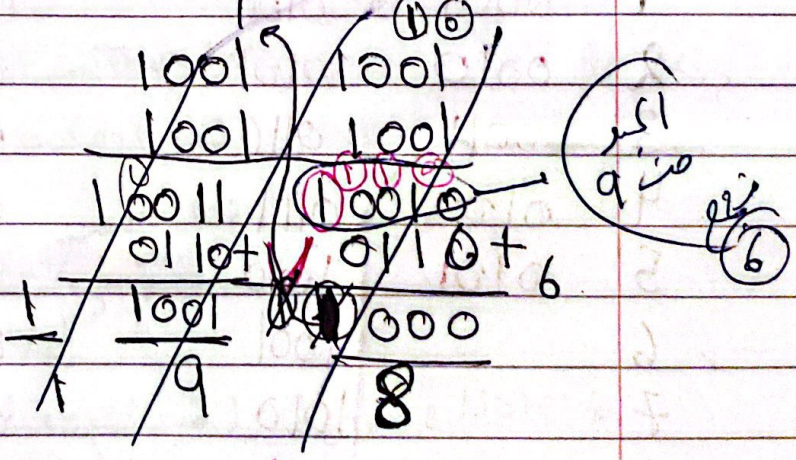
123
+ 488



In BCD

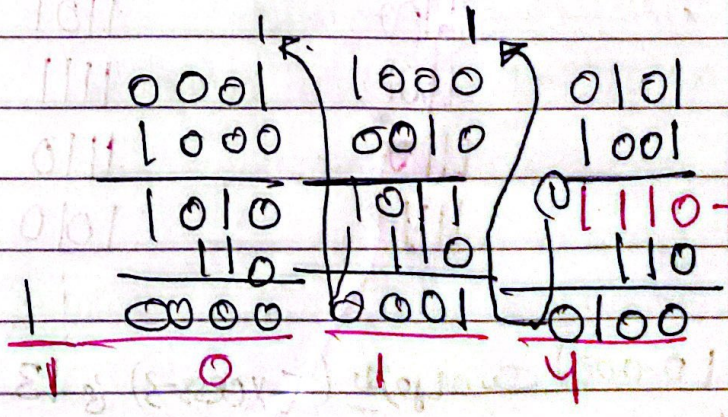
99
+ 99

بالإضافة (الخطأ)



In BCD

185
+ 829



In BCD

+ 99	1001	1001
99	1001	1001
	10011	10010
	0110 +	0110
1	1001	1000
1	9	8

اكبر من 9 نجمع 6 ثم نرفع ال carry

• other decimal code

	BCD	Excess-3	Gray code		
0	0000	0011	0000	13	1011
1	0001	0100	0001	14	1001
2	0010	0101	0011	15	1000
3	0011	0110	0010		
4	0100	0111	0110		
5	0101	1000	0111		
6		1001	0101		
7		1010	0100		
8		1011	1100		
9		1100	1101		
10		1101	1111		
11		1110	1110		
12		1111	1010		
⋮			⋮		

* Note: 13 في (Excess-3) بيكون (10000) 17

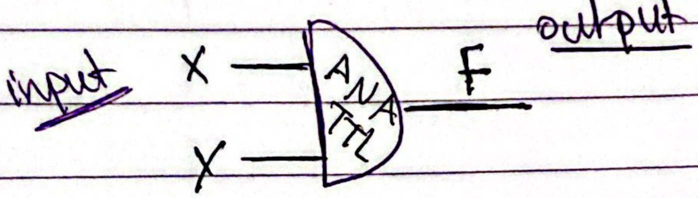
* Note: (Gray code) كل رقم الـ step والي بعده بيختلفوا بـ bit واحدة.

* Note: code 2421 ← مثل BCD

* Main goal ⇒ Design digital circuit that do severals operations (adder/multi/sub)

* Digital circuit consist of Logic gates.

① And gate



* Note: input اقلها ٢ .

* لازم يكون discreat signals صفر فولت او 5 فولت .

* TTL: (Transistor _ Transistor level): عبارة عن electronic devices بسمح للتيار بالمرور ، وعكسها بيمنع.

Truth table

$F = X \cdot Y$ (output: F; input: X, Y; "·" = and)

X	y	F
0	0	0
0	1	0
1	0	0
1	1	1

* Note: مرور التيار شرط كلهم ٥ فولت (ال input)

input is defined by user

output is defined by the system

② OR gate

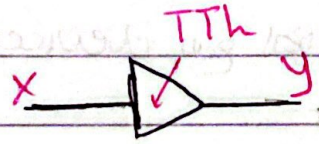
$F = x + y$ (or)

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

* يمر التيار (اذا كانت احدهم اقوت)

③ Buffer

- one input, one output
- synchronization (تزامن)
- هدف الـ Buffer اسرع بالزمن ←
- delay بتعمله
- السرعة بالـ Buffer تصبح بطيئة ←



x	y
0	0
1	1

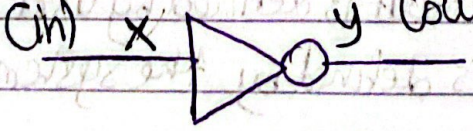
$y = x$

④ Inverter (not gate)

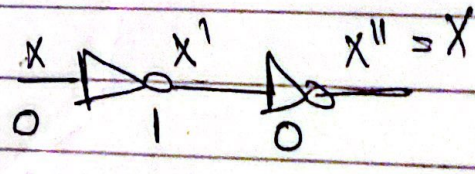
$y = not\ x$

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

$= x$ complement

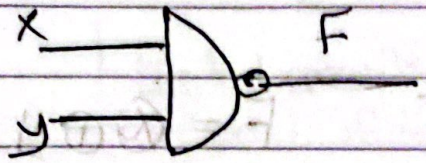


x	y
0	1
1	0



(delay كل ما عززنا حجم السرعة تقل)

⑤ NAND gate (not AND)



* ممكن اكثر من input اكثر من ٢.

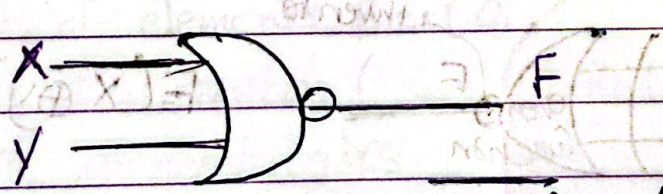
AND $\Rightarrow F = x \cdot y$

notAND $\Rightarrow F = \text{not}(x \cdot y)$

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

x, y	$x \cdot y$	$\overline{x \cdot y}$
0 0	0	1
0 1	0	1
1 0	0	1
1 1	1	0

⑥ NOR gate (Not OR)



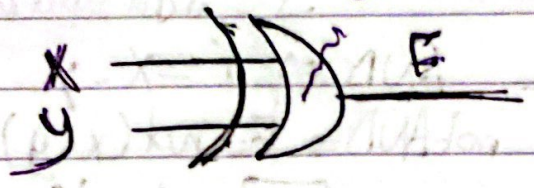
$F = (x+y)^c = \overline{(x+y)} = (x+y)'$

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

* كل الديجيتل ديزاينر ما يعملوا ديزاين للسركت NAND , NOR ، ما بستخدموا or, AND
* عدد الترانزستور في NAND, NOR اقل من AND, OR.

⑦ XOR gate

odd function



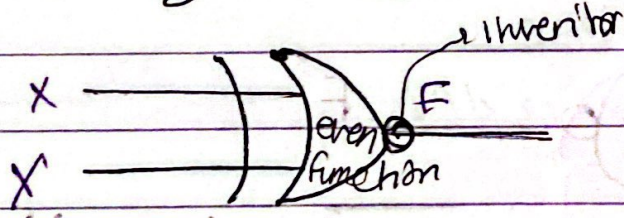
$F = x \oplus y$

$F = x \oplus y$

← عدد ① فردي بتعتبر XOR

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

XNOR gate (not XOR)



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

← عدد (1) = 1 زوج ← بتعتبر 0

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