Chapter - 1

Digital Systems

By : Jibreel Bornat

System converting

① Binary to decimal ($2 \rightarrow 10$):-

Way 1 :-

Decimal value $= (d_{(n-1)} \times 2^{(n-1)}) + (d_{(n-2)} \times 2^{(n-2)}) + \ldots + (d_1 \times 2^1) + (d_0 \times 2^0)$

Exp 1 :-

$$(10011101)_2 = (1 \times 2^7) + (2^4) + (2^3) + (2^2) + (2^0)$$

7 6 5 4 3 2 1 0

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

$$= 128 + 16 + 8 + 4 + 1$$

$$= \boxed{157}$$ # Done

Way 2 :-

By Putting the answers above of each bit

Exp :-

128	64	32	16	8	4	2	1
1	0	0	1	1	1	0	1

فقط نجمع الأرقام التي تحتها واحد :-

$$128 + 16 + 8 + 4 + 1 = \boxed{157}$$ # Done

② Decimal to Binary (10 → 2):-

We make a table then divide by 2

Exp :- Convert $(157)_{10}$ to Binary

157	2
78	1
39	0
19	1
9	1
4	1
2	0
1	0
0	1

Answer : $(10011101)_2$

طريقة الحل :-

① بعمل جدول : 2 على اليمين والرقم على اليسار

② بظل اقسم على 2 : على اليمين بحط الباقي وعلى اليسار بحط الناتج

③ لما يصير الناتج صفر بوقف

④ بكتب الجواب من تحت لفوق ↑

③ Anything to Decimal :-

How to convert any system to decimal ?

① find the radix

Radix is the number of the system:

Decimal $\Longrightarrow$ Radix = 10

Binary $\Longrightarrow$ Radix = 2

Octal $\Longrightarrow$ Radix = 8

Hexadecimal $\Longrightarrow$ Radix = 16

② Use this way to convert to decimal :

Decimal value = $(d_{(n-1)} \times r^{(n-1)}) + (d_{(n-2)} \times r^{(n-2)}) + \ldots\ldots + (d_1 \times r^1) + (d_0 \times r^0)$

* Since : r is the "radix"
n is the "bit number"
d is the "number"

Exp: -

$(11011)_2 = (1+2^4)+(1\times2^3)+(1\times2^1)+(1\times2^0)=27$

1

$(2107)_8 = (2\times8^3)+(1\times8^2)+(7\times8^0)=1095$

$(B2)_{16} = (11\times16)+(2\times16^0)=178$

④ Decimal to Anything:-

1095	8
136	7
17	1
2	0
0	2

Answer :- 2107

طريقة الحل :-

① بعمل جدول : r عاليمين والرقم عاليسار

② بظل اقسم عr : عاليمين بحط الباقي وعاليسار بحط الناتج

③ لما يصير الناتج صفر بوقف

④ بكتب الجواب من تحت لفوق ↑

⑤ Representing Fractions :-

Decimal value $= (d_{(n-1)} \times r^{(n-1)}) + \ldots + (d_1 \times r^1) + (d_0 \times r^0) + (d_{-1} \times r^{-1}) + \ldots + (d_{-m} \times r^{-m})$

From Anything to Decimal

$(1101.1001)_2 =$

$2^3 + 2^2 + 1 + \frac{1}{2} + \frac{1}{16} = 13.5625$

$(F22.35)_{16} =$

$16 * 16^2 + 2 * 16 + 2 + 3 * 16^{-1} + 5 * 16^{-2} = 6088.1875$

From Decimal to Anything

Number × radix	Answer
N × r	Answer

0.8125 × 8	6.5	6
0.5 × 8	4.0	4
0 لما أوصل صفر بوقف		

Answer = 0.64

Exp :-

- Convert $N = 0.6875$ to Radix 2
- Solution: Multiply N by 2 repeatedly & collect integer bits

Multiplication	New Fraction	Bit	
0.6875 × 2 = 1.375	0.375	1	→ First fraction bit
0.375 × 2 = 0.75	0.75	0	
0.75 × 2 = 1.5	0.5	1	
0.5 × 2 = 1.0	0.0	1	→ Last fraction bit

- Stop when new fraction = 0.0, or when enough fraction bits are obtained
- Therefore, $N = 0.6875 = (0.1011)_2$

Exp :-

- Convert $N = 139.6875$ to Octal (Radix 8)
- Solution: $N = 139 + 0.6875$ (split integer from fraction)
- The integer and fraction parts are converted separately

Division	Quotient	Remainder
139 / 8	17	3
17 / 8	2	1
2 / 8	0	2

Multiplication	New Fraction	Digit
0.6875 × 8 = 5.5	0.5	5
0.5 × 8 = 4.0	0.0	4

- Therefore, $139 = (213)_8$ and $0.6875 = (0.54)_8$
- Now, join the integer and fraction parts with radix point

$N = 139.6875 = (213.54)_8$

Complement Numbers

الهدف: تبسيط عمليات الطرح وبعض العمليات المنطقية

① N Complement :- "n-digits number"

10's complement:- 1000 ---- n times

9's complement:- 9999 ---- n times

8's complement:- 8888 --- n times

N's complement:- N N N N --- n times

The rule :-

Base - N

Since :- Base : This One

N : الرقم الذي نريد متممته

Examples :-

① 10's complement of 546700 = 1000000 - 546700 = 453300

② 10's complement of 012398 = 1000000 - 012398 = 987602

③ 9's complement of 546700 = 999999 - 546700 = 453299

④ 9's complement of 012398 = 999999 - 012398 = 987601

⑤ 8's complement of 546700 = 888888- 546700 = 342188

⑥ 8's complement of 012398 = 888888 - 012398 = 876490

Examples :-

① Using 10's comp do 72532 – 3250

First: Find the Comp of the negative number

10000 –3250 = 6750

Second: Add it to the first number

72532 + 6750 = 169282

Finally: if there is Overflow we discard it and the answer is positive.

Final Answer = 69282

② Using 10's comp do 3250 – 72532

100000 – 72532 = 27468

3250 + 27468 =30,718

No overflow ==> Negative

Since it's negative :-

Answer = – (10's comp for 30718)

= – (100000 – 30718)

= – 69282

② 1's complement

- Convert 1 to 0 and 0 to 1

Example

11 011000 becomes
00100111

③ 2's complement

- 1's complement + 1

Example

11 011000 becomes
00100111
+ 1
00101000

① تحويل الرقم إلى 1's Complement
② أضاف 1 للناتج

① أضع الأرقام تحت بعضها
② أجد 2's comp للرقم المطروح (السالب))
③ اجمع , اذا كان في زيادة يعني الرقم موجب

Example

Find 13−6 by 2's comp

① 00001101 − 00000110

② Twos comp of 6 is 11111001
+ 1
11111010

③ 00001101
+ 11111010
1 00000111

Carry => Positive Answer

④ Answer = 00000111 = 7

Difference between carry and over flow

① carry:-

Happen when we add / subtract un signed numbers their sum was out of range (dealing with 4-bit and the answer was 5-bit)

② overflow :-

Happen when dealing with signed numbers

* When we add two positive integers the answer is negative
* When we add two negative integers the answer is positive

①

```
  00001111    15
  00001000     8
  --------
  00010111    23
```

No Carry , No overflow

② They are alike ⇒ No overflow

```
 1 1 1 1 1
  00001111    15
  11111000    -8
  --------
 100000111     7
```

Yes Carry , No overflow

③ They are different ⇒ overflow

```
 0 1
  01001111    79
  01000000    64
  --------
  10001111   143
```

NO Carry , Yes Overflow

④

```
 1 0 0 1 1 0 0 0
  11011010    -38
  10011101    -99
  --------
 101110111   -137
```

Yes Carry , Yes overflow

• How to know if the number is negative on positive?

I have to check it its signed (2's complement) or unsigned

Signed :-

1000 = −8 not 8 , why ?

When it says to me signed I have to look at the mostleft bit

if it's 1 → negative number | لما يبلشى الرقم بـ 1 ⇐ سالب *

If it's 0 → positive number | لما يبلشى الرقم بـ 0 ⇐ موجب

Examples :-

	Signed	Unsigned
0011	3	3
1000	-8	8
1001	-7	9
1100	-4	12
1111	-1	15

1001 = −7 +0101 = 5 1110 = −2 (a) (−7) + (+5)	11 1100 = −4 +0100 = 4 10000 = 0 (b) (−4) + (+4)
Carry 00 0011 = 3 +0100 = 4 0111 = 7 (c) (+3) + (+4)	Carry 11 1100 = −4 +1111 = −1 11011 = −5 (d) (−4) + (−1)
Carry 01 0101 = 5 +0100 = 4 1001 = Overflow (e) (+5) + (+4)	Carry, No overflow 10 1001 = −7 +1010 = −6 10011 = Overflow (f) (−7) + (−6)

0010 = 2 +1001 = −7 1011 = −5 (a) M = 2 = 0010 S = 7 = 0111 −S = 1001	0101 = 5 +1110 = −2 10011 = 3 (b) M = 5 = 0101 S = 2 = 0010 −S = 1110
1011 = −5 +1110 = −2 11001 = −7 (c) M = −5 = 1011 S = 2 = 0010 −S = 1110	0101 = 5 +0010 = 2 0111 = 7 (d) M = 5 = 0101 S = −2 = 1110 −S = 0010
0111 = 7 +0111 = 7 1110 = Overflow (e) M = 7 = 0111 S = −7 = 1001 −S = 0111	1010 = −6 +1100 = −4 10110 = Overflow (f) M = −6 = 1010 S = 4 = 0100 −S = 1100

Minimum number of bits required

$2^{n-1} < M < 2^n$ ⟹ M : The wanted number

Example :-

How many bits do you need to represent 10 decimal digits

Solution :-

$\log_2^{10}$ = Answer ⟹ $\log_2^{10} = 4$

⟹ to represent 10 we need 2^4 Bit = 16

Decimal codes

① To simplify conversions, decimal codes can be used

② Define a binary code for each decimal digit

③ Since 10 decimal digits exit, a 4-bit code is used

Binary coded decimal (BCD)

BCD is a weighted code like binary (8, 4, 2, 1)

There are six invalid codes (1010, 1011, 1100, 1101, 1110, 1111)

Coding:

$13_{10} = (0001\ \ 0011)_{BCD}$

$307_{10} = (0011\ \ 0000\ \ 0111)_{BCD}$

Conversion:

$13_{10} = (1101)_2$

$307_{10} = (100110011)_2$

BCD Arithmetic:-

① write the representation of each number

② add each digit with the digit below it

③ if the answer is more than "9" (Num > 9)
add "1" carry in the next set of numbers

④ after finishing, if the answer of the set is
more than "9", add 6 to it "0110"

Example 1:-

❖ Add 2905_{BCD} to 1897_{BCD} showing carries and digit corrections.

		1 Carry	1 Carry	1 Carry	
1897_{BCD}		0001	1000	1001	0111
2905_{BCD}	+	0010	1001	0000	0101
No plus 6 Because its not Bigger than 9	⇐	0100	10010 >9	1010 >9	1100 >9
		0000	0110	0110	0110 Plus 6
		0100	1000	0000	0010
		4	8	0	2

Example 2:-

Add 5789 to 3901 using BCD showing each step

Carry Because > 9

	1		1	
5789	0101	0111	1000	1001
+ 3901	0011	1001	0000	0001
	1001	10000	1001	1010
	0000	0110	0000	0110
	1001	10110	1001	10000

Add 6 Because >9

Delete Any Carry

Follow the colours to understand

● → ● → ●

Final Answer :- 1001 0110 1001 0000 = 9690

Gray Code

Used to detect if there is any error occurred during any process in the computer

From binary to Gray code:-

① put the most left bit

② compare each digit with the digit beside it

- Same (00 , 11) : put 0
- Different (01 , 10) : put 1

Example 1:-

Convent 011_2 to Gray code

Answer = 0 1 0

Example 2:-

Convent 1 1 0 0 1 0 to Gray code

Answer = 1 0 1 0 1 1

From Gray code to binary:-

① put the most left bit

② compare each digit with the digit next to its above

Example 1:-

Convent 0 1 0_{Gray} to Binary

Always

Answer = 0 1 1

Example 2:-

Convent 1 0 1 0 1 1_{Gray} to Binary

Answer = 1 1 0 0 1 0

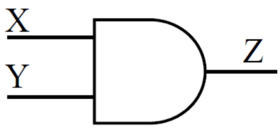
Binary Logic

① AND

$Z = X \text{ and } Y$

$Z = X \cdot Y$

$Z = XY$

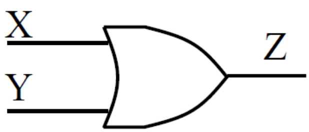


X	Y	Z=XY
0	0	0
0	1	0
1	0	0
1	1	1

② OR

$Z = X \text{ or } Y$

$Z = X + Y$

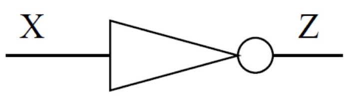


X	Y	Z=X+Y
0	0	0
0	1	1
1	0	1
1	1	1

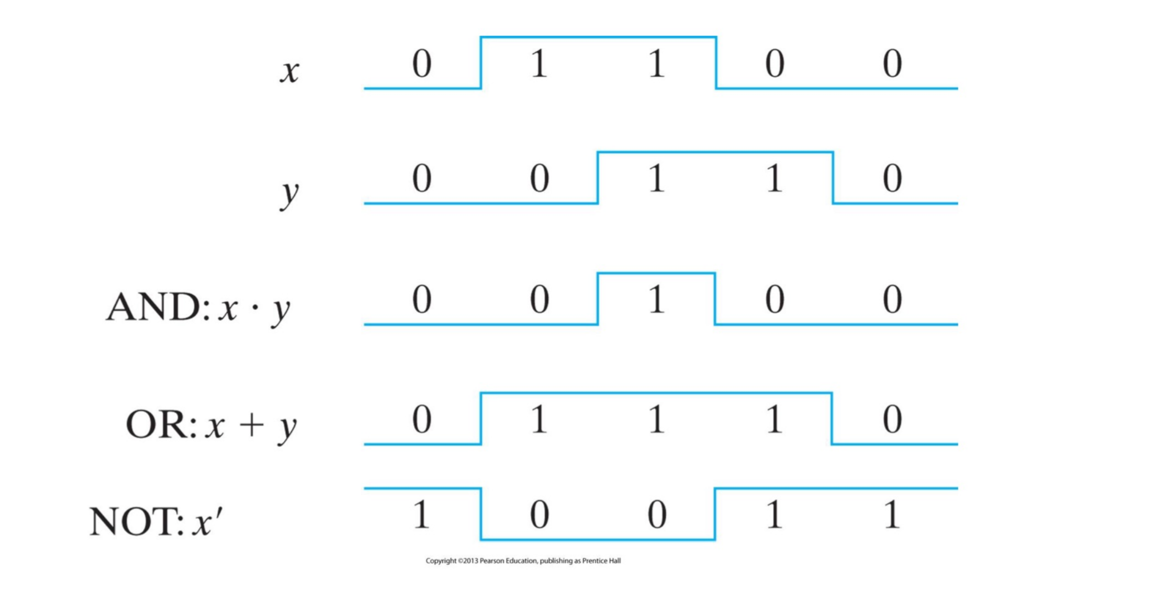
③ NOT

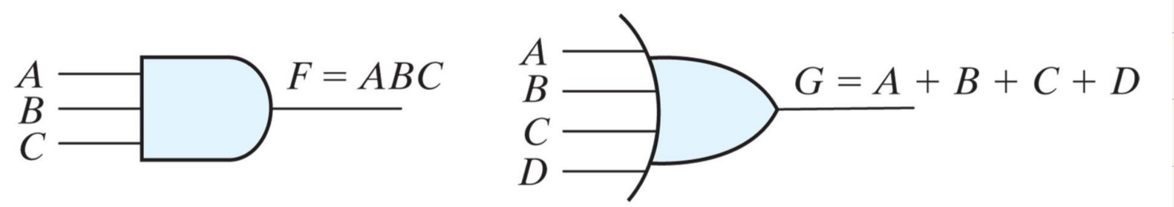
$Z = \bar{X}$ or $Z = X'$

X	Z=X′
0	1
1	0



Examples





(a) Three-input AND gate (b) Four-input OR gate

Believe

YOURSELF

CO