Number systems

Decimal
Binary
Hexa Decimal

2 5 6 3

2x1000 + 5X100 6X10 3X1

2000 + 500 + 60 +3

= 2563

decimal binary

8 4 2 1

.... 1000 100 10 1

if if circle

$1011 = (11)_{10}$

1X8 0X4 1X2 1X1

8 +0+2+1

0000 0000 → 0
0100 0000 → 64
1000 0000 → 128
1100 0000 → 192
1111 1111 – 256

0000 0001 = 1
0 0000010 = 2
0 0000100 = 4
0 0001000 = 8
0 0010000 = 16
0 0100000 = 32
0 1000000 = 64
1000 0000 = 128

Scanned with OKEN Scanner

Number system 1 – Assessment

① $(15)_{10} \rightarrow (?)_2$ $\boxed{1111}$

② $(1101)_2 = (?)_{10}$ $\boxed{13}$

③ $(76)_{10} = (?)_2$

100
64+8+4
01001100

④ $(01011010)_2 = (?)_{10}$

64+16+8+2
⇒ 90

⑤ $(11011010)_2 = (?)_{10}$

128+64+16+8+2 ⇒ 218

⑥ $(156)_{10} = (?)_2$

128+16+8+4
= 10011100

⑦ $(356)_{10} = (?)_2$

256+64+32+4
101100100

⑧ $(101011011)_2 = (?)_{10}$

256+64+16+8+2+1
= 347

⑨ Decimal 793 $= (?)_2$

512+256+16+8+1

$\boxed{1100011001}$

⑩ $(1010110111) = (?)_{10}$

512+128+16+32+4+2+1
$= (695)_{10}$

Hexa decimal system

2^{10} – 1024 – 1 kilo

2^{20} – $2^{10} \times 2^{10}$ – 1 mega

2^{30} – $2^{10} \times 2^{10} \times 2^{10}$ – 1 giga

2^{40} – $2^{10} \times 2^{10} \times 2^{10} \times 2^{10}$ – 1 Tera

2^{50} – .. – 1 peta

$2^{24} = ?$

$2^{20} \times 2^4 = 16M$

$2^{32} = ?$

$2^{30} \times 2^2 = 4G$

$2^{18} = 2^{10} \times 2^8$
$= 256K$

$2^{47} = 2^{40} \times 2^7$
$= 128$ Tera

Tip

2^{24} → 16 m , 2^{46} → 64 Tera

$128K = 2^{17}$, $256G = 2^{38}$

$\cancel{16}6T = 2^{44}$

1010 — 10 — A
1011 — 11 — B
1100 — 12 — C
1101 — 13 — D
1110 — 14 — E
1111 — 15 — F

Number system Assessment-II

1. $C = (\ )_2$ ⇒ 1101
2. $(1011)_2 = (\ )_{16}$ ⇒ B
3. $(76)_{16} = (\ )_2$ ⇒ 0111 0110
4. $(01011010)_2 = (\ )_{16}$ ⇒ 5A
5. $(97)_{16} = (\ )_2 = (\ )_{10}$ ⇒ 1001 0111 , 97 → $16 \times 9 + 16 \times 7$ = 151
6. $(97)_{10} = (\ )_{16}$ ⇒ 16 | 97, 6 − 1 ⇒ 61
7. $(97)_{16} = (\ )_8$ ⇒ 8 | 151, 8 | 18 − 7, 2 − 2 ⇒ $(227)_8$
8. $(77)_8 = (\ )_{16}$ → 56 + 7 $\cancel{63}$, 16 | 63, 3 − 15 → 3F
9. $(97)_{10} = (\ )_8$ → 8 | 97, 8 | 12 − 1, 1 − 4 ⇒ $(141)_8$
10. $(77)_8 = (\ )_{10}$ → $8 \times 7 + 7$ =

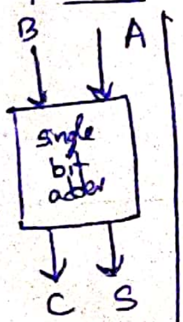
Assessment – 3

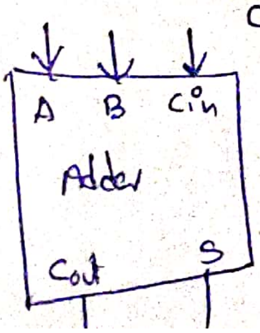
1. $2^0 = 1$
2. $2^1 = 2$
3. $2^2 = 4$
4. $2^3 = 8$
5. $2^4 = 16$
6. $2^5 = 32$
7. $2^6 = 64$
8. $2^7 = 128$
9. $2^8 = 256$
10. $2^9 = 512$

Assessment – 4

1. $2^x = 128K \Rightarrow 2^{17}$
2. $2^x = 256M \Rightarrow 2^{28}$
3. $2^x = 64G \Rightarrow 2^{36}$
4. $2^x = 32T \Rightarrow 2^{45}$
5. $2^x = 16P \Rightarrow 2^{54}$
6. $2^x = 4G \Rightarrow 2^{32}$
7. $2^x = 8M \Rightarrow 2^{23}$
8. $2^x = 2K \Rightarrow \cancel{2^{14}}\ 2^{11}$
9. $2^x = 512T \Rightarrow 2^{49}$
10. $2^x = 64P \Rightarrow 2^{56}$

Binary operations





A	B	Cin	Cout	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

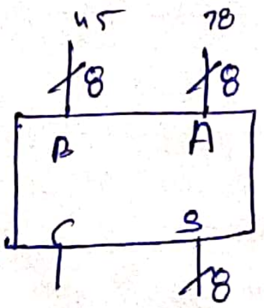
Binary addition Assement

①

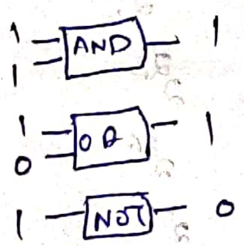
0100
1100 ⇒ 10000

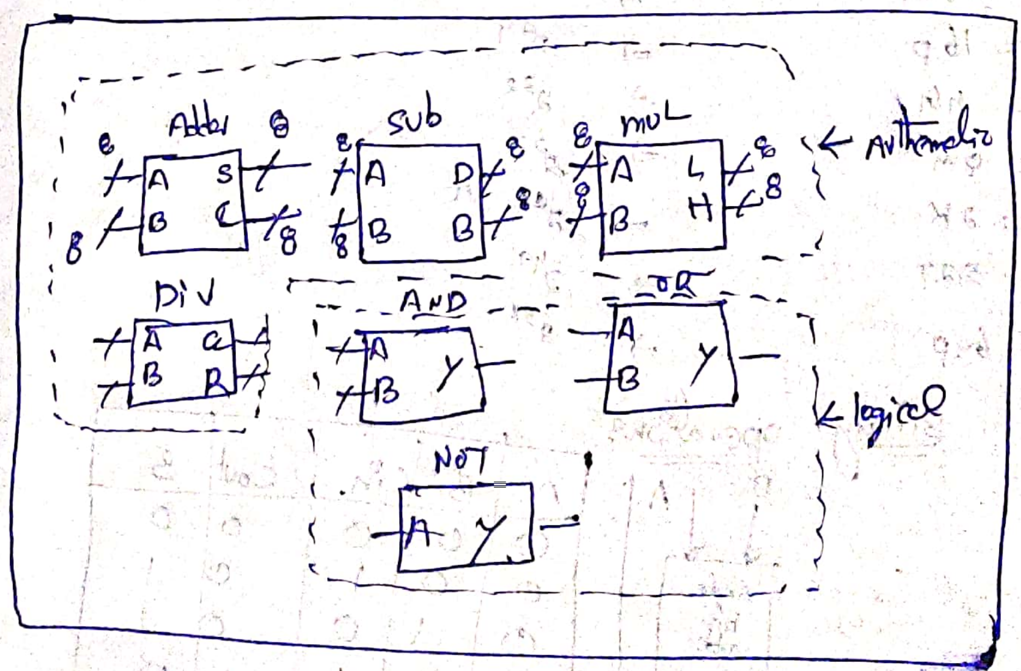
②

01101100 ⇒ 100100001
10110101
100100001



0111 1000
0100 0001
10111001
= B D





ALU

ALU Assesement

①

  1100
- 0101
  0111

②
```
  11101100
- 00110101
----------
  10110111
```

③ OR
```
  11101100
  00110101
----------
  11111101
```

④ AND
```
  11101100
  00110101
----------
  00100100
```

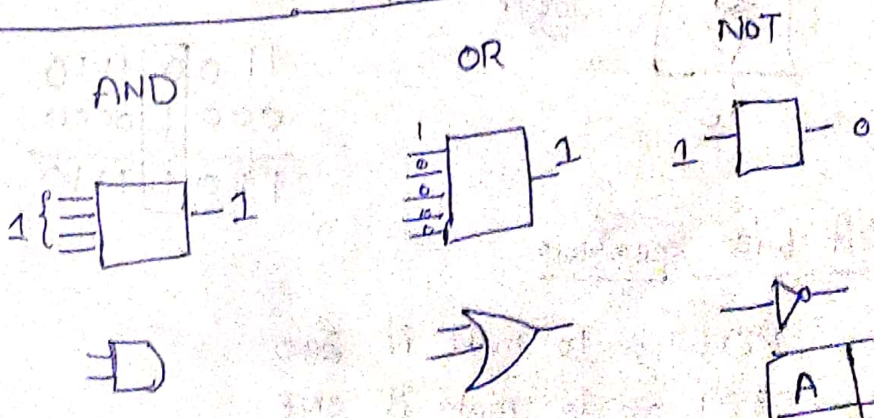
⑤ NOT
```
  11101100
----------
  00010011
```

⑥ XOR
```
  11101100
  00110101
----------
  11011001
```

⑦ one's complement
```
  11101100
----------
  00010011
```

⑧ 2's cpl
```
  11101100
----------
  00010011
         1
----------
  00010100
```

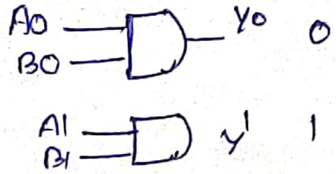
⑨
```
  10101100
- 11110101
----------
+1.10110111
 -10110111
```



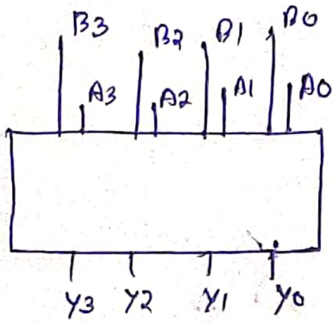
A	Y
0	1
1	0

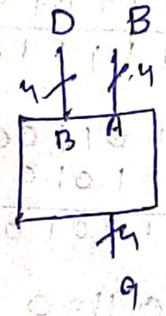
A	B	OR	AND
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

A = 10
B = 11.



A = 1011
B = 1101
1001

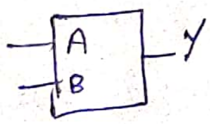




if we wat to ~~set~~ Reset particular bit use masking method i.e AND operation with all bits are one except desired bit

→ To make desired bit set do OR with all zero except desired Bit position

Exclusive OR



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Toggle 4th bit

1100 $\boxed{0}$ 1110
000 $\boxed{1}$ 0000
1101 $\boxed{1}$ 1110

Bit wise operations

AND — RESET — to make it zero
OR — SET — to make it ONE
NOT — compliment → Negate
XOR — Toggle

Assessment

①	7 6 5 4 3 2 1 0 1 1 1 0 1 1 0 0	SET Bit 4.	sol do OR with 00010000
②	1 1 1 0 1 1 0 0	Reset Bit 5	sol AND
③	1 1 1 0 1 1 0 0	Toggle Bit 2	sol 0001 0000
④	1 1 1 0 1 1 0 0 OR X X X X X X X X	SET Bit 4	sol 11011111
⑤	1 1 1 0 1 1 0 0 AND X X X X X X X X	Reset Bit 5	sol 00000010
⑥	1 1 1 0 1 1 0 0 XOR X X X X X X X X	Toggle Bit 2	sol 10010000
⑦	1 1 1 0 1 1 0 0 OR X X X X X X X X	SET Bit 4 & 7	sol 11010111
⑧	1 1 1 0 1 1 0 0 AND X X X X X X X X	Reset Bit 5 & 3	sol 01000100
⑨	1 1 1 0 1 1 0 0 XOR X X X X X X X X	Toggle Bit 2 & 6	sol 1111 1111
⑩	1 1 1 0 1 1 0 0 XOR X X X X X X X X	Toggle all the bits	

ASCII Conversion - Assessment

① 0 Binary? — 0000 0000

② 0011 0010 ASCII? — 2

③ $\boxed{A}$ ASCII — Binary?

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

= 01000001

④ 0100 0110 — ASCII $\boxed{}$

sol $2+4+64 = 70 = F$

⑤ ASCII $\boxed{A}$ — Decimal $\boxed{}$

$A = 65$

⑥ $\boxed{6\,6}$ Decimal — $\boxed{}$ ASCII

sol $66 = B$

⑦ ASCII $\boxed{E}$ Hexa $\boxed{}$ $(69)_{10}$

⇒ 16 | 69, 4 − 5 = $\boxed{45}$

⑧ HEX $\boxed{48}$ ASCII $\boxed{}$

⇒ $(48)_{16} = (72)_{10}$

72 = $\boxed{H}$

⑨ ASCII $\boxed{d}$ HEX $\boxed{}$

⇒ d = 100, 16 | 100, 6 − 4

$\boxed{64}$

⑩ HEX $\boxed{66}$ ASCII $\boxed{}$

⇒ $(66)_{16} = (102)_{10}$

102 = f

Binary fraction Assessement

① 1110 1100 shift Let 3 bits sol 0110 0000
xxxx xxxx

② 1110 1100 shift right 2 Bits sol 0011 1011
xxxx xxxx

③ 1110 1100 Rotate left 3 bits sol 01100111
xxxx xxxx

④ 1110 1100 Rotate right 2 bits sol 0011 1011
xxxx xxxx

⑤ 1110 1100 shift right 3 bits & shift left 3 bits. >>3 0001 1101 <<3 1110 1000
xxxx xxxx

⑥ 11101100 shift left 2 bits and shift right 3 bits <<2 1011 0000 >>3 000 10110
xxxx xxxx

⑦ 1110 1100 mask the lower nibble. sol mean AND with F0.
xxxx xxxx
1110 0000

⑧ 1110 1100 Mask the upper nibble sol AND with 0F
xxxx xxxx
0000 1100

⑨ 0010 1100 multiply this number by 5 using binary operator sol ~~AND~~ left shift cc
xxxx xxxx

⑩ 0110 1000 Divide this number by 8 using binary operator sol
xxxx xxxx

0000.
1000) 0110 1000
0
01
0
011
0
0110
0
0001
0
0

1000>0
1000>01
1000>011
1000>0110
1000<1101

sol 1000) 0110 1000

1000 > 0
1000 > 01
1000 > 011
1000 > 110
1000 < 1101
1000 < 1010
1000 > 100
1000 = 1000

```
  1000 ) 0110 1000
         1 0 0 0
         --------
         0 1 0 1 0
       - 1 0 0 0
         --------
         0 0 1 0 0 0
             1 0 0 0
             -------
             0 0 0 0
```

another way

in Binary arithmetic, dividing by a power 2 is achieved using the right shift (>>) operator

Divide by 2 = shift right by 1 bit (>>1)
" " 4 = " " " 2 " (>>2)
" " 8 = " " " 3 " (>>3)

multiplication

By power 2 is (<<) operator

Binary Fraction Assessment

(01) $(1.5)_{10} = (\ \ )_2$

sol step Take 1.5 integer part
Note (integer part) 1.5 1

step 2 : multiply fraction part with 2 & note integer part again mul fraction with 2 until zero

1.5 integer part 1

$0.5 \times 2 = 1.0$ 1

(2) $(2.25)_{10} = (\ \ )_2$

sol integer part - 2 so as usual

010

then fraction part

$0.25 \times 2 = 0.5$ integer part = 0

$0.5 \times 2 = 1.0$ integer part = 1

fraction is zero so stop

$\boxed{010.01}$

(3) $(100.0010)_2 = (\ \ )_{10}$ sol integer part 100 = 4

Next $0 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} = 0.125$ 4

$\boxed{4.125}$

Scanned with OKEN Scanner

④ $(0101.1010)_2 = (\ )_{10}$ **sol** Integer part $0101 = 5$

$1\times2^{-1}+0\times2^{-2}+1\times2^{-3}+0\times2^{-4} = 0.625$

$\therefore \boxed{5.625}$

⑤ $(69.65625) = (\ )_2$ **sol** integer part

$$\begin{array}{r|l} 2 & 69 \\ 2 & 34-1 \\ 2 & 17-0 \\ 2 & 8-1 \\ 2 & 4-0 \\ 2 & 2-0 \\ & 1-0 \end{array} = 1000101$$

Fraction part

	integer part note
$0.65625\times2 = 1.3125$	1
$0.3125\times2 = 0.625$	0
$0.625\times2 = 1.25$	1
$0.25\times2 = 0.5$	0
$0.5\times2 = 1.0$	1

∴ fraction is zero so, stop

$\boxed{1000101.10101}$

⑥ $(01101010.10011000) = (\ )_{10}$ **sol** integer part

~~$0+1\times2^7+1\times2^5+0\times2^5+1\times2^3+0\times2^2+1\times2^1$~~ $+0\times2^0$

$= 2^6+2^5+2^3+2^1$

106

Fraction part | Integer part

~~$0.10011000\times2 = 0.20022$~~ 0

~~$0.20022\times2 = 0.40044$~~ 0

~~$0.40044\times2 = 0.80088$~~ 0

$1\times2^{-1}+0\times2^{-2}+0\times2^{-3}+1\times2^{-4}+1\times2^{-5} = 0.59375$

$\boxed{106.0.59375}$

⑦ $(0101.1010) = 1.011010 \times 2^x$

sol in original number point is there after 4th position

step 2: shift the pattern to make 1.xxxx here decimal is after 1 position from left

so $x = 4-1$
$= 3$

(08) $0.0001010 = 1.01000 \times 2^x$ sol

here original number
0.00011010

to make
$1.xxx$
Need to shift 4 position

so $x = 4$

(09) $19.25 = \boxed{1.\ \ \ \ \ \ \ } \times 2^x$ sol take integer part

$$\begin{array}{r|l} 2 & 19 \\ 2 & 9 - 1 \\ 2 & 4 - 1 \\ 2 & 2 - 0 \\ & 1 - 0 \end{array} = 10011$$

after that decimal need to calculate
But simply
10011 how many shifts need
to make as $1.xxx$
$= 4$

(10) $0.078125 = \boxed{1.\ \ \ \ \ \ \ } \times 2^x$ = sol
find binary value for

	part
$0.078125 \times 2 = 0.15625$	0
$0.15625 \times 2 = 0.3125$	0
$0.3125 \times 2 = 0.625$	0
$0.625 \times 2 = 1.25$	1
$0.25 \times 2 = 0.5$	0
$0.5 \times 2 = 1.0$	1

$.000101$

so Need to shift by 4

$\therefore \boxed{x = -4}$

fraction part.

	inte part
	1
~~$0.078125 \times 2 = 1.5625$~~	
~~$0.5625 \times 2 = 1.125$~~	1
~~$0.125 \times 2 = 0.25$~~	0
~~0.25×2~~ $= 0.5$	0
~~0.5×2~~ $= 1.0$	

so, ~~stop~~, 1100

~~original number.~~

~~0.1100.~~

~~To make as $1.xxx$ need to~~

~~shift by~~ 1

so $\boxed{\cancel{x = 1}}$