CS504

Section 2

* Methods for converting $(\ )_{10}$ to $(\ )_2$ & $(\ )_2$ to $(\ )_{10}$

→ unsigned Method.
→ Signed magnitude
→ 1's Complement
→ 2's Complement.

EX.

→ Convert $(11101)_2$ to $(?)_{10}$ using::

1) unsigned Method. (if the number was stored as unsigned number).

$$(\overset{16}{1}\,\overset{8}{1}\,\overset{4}{1}\,\overset{2}{0}\,\overset{1}{1})_2 = 16+8+4+1 = (29)_{10}$$

2) Signed - Magnitude

$$(\boxed{1}\,\underbrace{\overset{8}{1}\,\overset{4}{1}\,\overset{2}{0}\,\overset{1}{1}})_2$$

$$(-13)_{10}$$

3) 1's Complement method. (because the number is Negative →

The result = − 1's Complement of (11101) =
$-\ 00010 = (-2)_{10}$

4) 2's Complement method (because the number is Negative →

The result = − 2's Complement of (11101) =
$-\ 00011 = (-3)_{10}$

←

* Convert $(-12)_{10}$ to binary in 6 bits. using:

1) unsigned method

↳ $(001100)_2$

2) signed magnitude

↳ $(101100)_2$

3) 1's complement

↳ The result = 1's complement of (001100)

↳ $(110011)_2$

4) 2's complement

↳ The result = 2's complement of (001100)

$= (110100)_2$

←

Scanned by CamScanner

⇒ 9's Comp. & 10's Comp.
⇒ 15's Comp. & 16's Comp.

Ex.

⇒ Find The 9's Comp. & 10's Comp. of $(723500)_{10}$

1) 9's Comp. =

$$\begin{array}{cccccc} -9 & -9 & -9 & -9 & -9 & -9 \\ 7 & 2 & 3 & 5 & 0 & 0 \end{array}$$

$$(276499)_{9\text{'s Comp.}}$$

2) 10's Comp. =

$$\begin{array}{cccccc} -9 & -9 & -9 & -10 & & \\ 7 & 2 & 3 & 5 & 0 & 0 \\ & & & \downarrow & \downarrow & \downarrow \end{array}$$

$$(276500)_{10\text{'s Comp.}}$$

or:
10's Comp. =
9's Comp. + 1

⇒ Find The 15's Comp. & 16's Comp. of (3496)

1) 15's Comp. of $\begin{pmatrix} -15 & -15 & -15 & -15 \\ 3 & 4 & 9 & 6 \end{pmatrix}$

$$(CB69)_{15\text{'s Comp.}}$$

A 10
B 11
C 12
D 13
E 14
F 15

2) 16's Comp. of $\begin{pmatrix} -15 & -15 & -15 & -16 \\ 3 & 4 & 9 & 6 \end{pmatrix}$

$$(CB6A)_{16\text{'s Comp.}}$$

or:
16's Comp. =
15's Comp. + 1

* Another Codes for Converting $()_2$ to $()_{10}$ & Vs:-

⇒ BCD Codes.

B → Binary, C → Coded, D → Decimal

(4 bits for each digit) (from 0 to 9)

- → BCD 8421 (BCD)
- → BCD 2421
- → BCD 5421
- → BCD 84-2-1
- → BCD 6311
- → BCD 6421
- → BCD Excess 3.

BCD 8421

Dec	Binary 8	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

10
:
:
15

unused

BCD 2421

لا نستخدمه إلا بعد رقم ٤

أي رقم بعد (4) نستخدم فيه أخر weight

Dec.	Binary 2	4	2	1	unused
0	0	0	0	0	
1	0	0	0	1	
2	0	0	1	0	→ 1 0 0 0 X
3	0	0	1	1	→ 1 0 0 1 X
4	0	1	0	0	→ 1 0 1 0 X
5	1	0	1	1	→ 0 1 0 1 X
6	1	1	0	0	→ 0 1 1 0 X
7	1	1	0	1	→ 0 1 1 1 X
8	1	1	1	0	
9	1	1	1	1	

⇒ BCD 5421

نستخدم بعد ٤

أي رقم بعد (4) نستخدم منه

weight

Dec	Binary 5	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	1	0	0	0
6	1	0	0	1
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0

un used

$\boxed{0\ 1\ 1\ 0}$ X

$\boxed{0\ 1\ 1\ 1}$ X

BCD 84-2-1

Dec	Binary 8	4	-2	-1
0	0	0	0	0
1	0	1	1	1
2	0	1	1	0
3	0	1	0	1
4	0	1	0	0
5	1	0	1	1
6	1	0	1	0
7	1	0	0	1
8	1	0	0	0
9	1	1	1	1

BCD 6311

Dec	Binary 6	3	1	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	1	0	0
4	0	1	1	0
5	0	1	1	1
6	1	0	0	0
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0

BCD 6421

Dec	Binary 6 4 2 1
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	1 0 0 0
7	1 0 0 1
8	1 0 1 0
9	1 0 1 1

Excess 3 (4 bits for each digit) (From 0:9)

$(\ )_2$ in Excess $\xrightarrow{-3}$ $(\ )_{10}$ $\xrightarrow{+3}$ $(\ )_2$ in Excess

Dec	Binary 8 4 2 1
0+3	0 0 1 1
1+3	0 1 0 0
2+3	0 1 0 1
3+3	0 1 1 0
4+3	0 1 1 1
5+3	1 0 0 0
6+3	1 0 0 1
7+3	1 0 1 0
8+3	1 0 1 1
9+3	1 1 0 0

Ex: Convert $(347)_{10}$ to Binary using Excess_3 Method:

$$\begin{array}{ccc} 3 & 4 & 7 \\ \downarrow{+3} & \downarrow{+3} & \downarrow{+3} \\ 6 & 7 & 10 \end{array}$$

$(0110 \ 0111 \ 1010)_2$

Ex

⇒ Represent the decimal number $(235)_{10}$ in:

① BCD ② excess_3 Code. ③ 2421 Code ④ 84-2-1 Code
⑤ 6311 Code ⑥ 6421 Code ⑦ 9's comp. Code ⑧ 10's Code.
⑨ Signed Magnitude ⑩ 1's Complement ⑪ 2's Complement

in 10 bits (as Positive Number)
(as Negative Number.

ANS-

① BCD ⇒ (BCD 8421)

2 3 5

$(0010\ 0011\ 0101)_2$ (BCD 8421)

② excess 3 Code

2 3 5
+3 +3 +3
5 6 8

$(0101\ 0110\ 1000)_2$ (BCD Excess 3)

③ 2421 Code.

2 3 5

$(0010\ 0011\ 1011)$ (2421 Code)

	2	4	2	1
5 →	1	0	1	1
3 →	0	0	1	1
2 →	0	0	1	0

④ 84-2-1

2 3 5

$(0110\ 0101\ 1011)$ (BCD 84-2-1)

	8	4	-2	-1
5 →	1	0	1	1
3 →	0	1	0	1
2 →	0	1	1	0

⑤ 6311 Code

2	3	5

(0011 0100 0111) (BCD 6311)

	6	3	1	1
5 →	0	1	1	1
3 →	0	1	0	0
2 →	0	0	1	1

⑥ 6421 Code

2	3	5

(0010 0011 0101) (BCD 6421)

	6	4	2	1
5 →	0	1	0	1
3 →	0	0	1	1
2 →	0	0	1	0

⑦ 9's Comp. Code.

-9	-9	-9
2	3	5

(7 6 4) (9's Comp.) ~~Code~~.

⑧ 10's Comp. Code

-9	-9	-10
2	3	5

(7 6 5) 10's Comp.

←

Scanned by CamScanner

9 Signed Magnitude of $(235)_{10}$ → in 10 bits

+ or -

$(11101011)_2$

		remainder
2 ÷	235	1
	117	1
	58	0
	29	1
	14	0
	7	1
	3	1
	1	1
	0	

⇒ if the Decimal Number was Positive:

(0011101011) Signed magnitude

⇒ if the number was Negative:

(1011101011)

10 1's Comp. Code:

⇒ if the number was Positive:

The result = (0011101011)

⇒ if the number was Negative.

The result = (− 1's Comp. of (0011101011)

= −(1100010100)1's Comp.

11 2's Comp. Code.

⇒ if the number was Positive.

The Result = (0011101011)

⇒ if the number was Negative:

The Result = − 2's Comp. of (0011101011)

= (1100010101)2's Comp.

Q 1-33) in The book

→ The state of a 12 bit register is (100010010111)

What is the content if it represents:

a) three decimal digits in BCD?

b) " " " in Excess 3?

c) " " " in 84 -2 -1?

d) A binary Number?

ANS -

① BCD 8421

8	4	2	1	8	4	2	1	8	4	2	1
1	0	0	0	1	0	0	1	0	1	1	1

$(8 \quad 9 \quad 7)_{10}$

---

② Excess 3

8	4	2	1	8	4	2	1	8	4	2	1
1	0	0	0	1	0	0	1	0	1	1	1

8 −3, 9 −3, 7 −3

$(5 \quad 6 \quad 4)_{10}$

---

③ 8 4 -2 -1

8	4	-2	-1	8	4	-2	-1	8	4	-2	-1
1	0	0	0	1	0	0	1	0	1	1	1

$(8 \quad 7 \quad 1)_{10}$

←

d) a binary Number: (index method)

(نأخذ الرقم كامل لا يقسم
(4 bits لـ

$$\begin{array}{cccccccccccc} 11 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{array}$$

$$2^{11} + 2^{7} + 2^{4} + 2^{2} + 2^{1} + 2^{0}$$

$$= (2199)_{10}$$

Ex

↳ Convert the binary number 1000 0110 0001 to $(\ )_{10}$ using 2421 code.

$$\underbrace{\overset{2\,4\,2\,1}{1000}}_{2} \quad \underbrace{\overset{2\,4\,2\,1}{0110}}_{6} \quad \underbrace{\overset{2\,4\,2\,1}{0001}}_{\downarrow}$$

- 1000 → 2 invalid ↳ because the code of "2" is (0010)
- 0110 → 6 invalid ↳ because the code of "6" is (1100)
- 0001 → ✓

↳ Convert the binary number 1111 0100 1110 to $(\ )_{10}$ using Excess 3:

$$\underbrace{\overset{8\,4\,2\,1}{1111}}_{15} \quad \underbrace{\overset{8\,4\,2\,1}{0100}}_{4} \quad \underbrace{\overset{8\,4\,2\,1}{1110}}_{14}$$

- 15 − 3 = $\boxed{12}$ invalid ↳ because last digit in Excess 3 code is 9
- 4 − 3 = $\boxed{1}$ ✓
- 14 − 3 = $\boxed{11}$ invalid ↳ because last digit in Excess 3 code is 9

⇒ Gray Code (4 bits for each digit)

from (1 : 15)

⇒ to Convert from decimal to (?) Gray code →

① find the binary Number

$(\ )_{10} \longrightarrow (\ )_2 \longrightarrow (\ )_{\text{Gray Code}}$

EX:

* Convert $(5)_{10}$ to Gray Code :-

① $(5)_{10}$ in 4 bits $= (0101)_2$

②

$0 \quad 1 \quad 0 \quad 1$

$\oplus \quad \oplus \quad \oplus$

$(0 \quad 1 \quad 1 \quad 1)_{\text{Gray Code}}$

$\oplus \rightarrow$ Xor

$0 \oplus 0 = 0 \qquad 1 \oplus 1 = 0 \qquad 1 \oplus 0 = 1 \qquad 0 \oplus 1 = 1$

⇒ to Convert from Gray Code to decimal :-

$(\ )_{\text{Gray Code}} \longrightarrow (\ )_2 \longrightarrow (\ )_{10}$

EX:

* Convert $(1101)_{\text{Gray Code}}$ to $(\ )_{10}$:-

$1 \quad 1 \quad 0 \quad 1$

$\oplus \quad \oplus \quad \oplus$

$(1 \quad 0 \quad 0 \quad 1)_2$

$(1001)_2 = (9)_{10}$

Scanned by CamScanner

* Convert $(62)_{10}$ to Gray Code :

أي رقم أكبر من (15) نقسمه إلى (digits) وكل (digit) نحوله لـ Gray code

$$6 \rightarrow (0110)_2 \rightarrow (0101)_{\text{Gray code}}$$

$$2 \rightarrow (0010)_2 \rightarrow (0011)_{\text{Gray code}}$$

$$(62)_{10} = (01010011)_{\text{Gray Code (gc)}}$$

* Convert $(1101\ 0010\ 0011)_{\text{Gray Code}}$ to decimal?

$$1101 \rightarrow (1001)_2 \Rightarrow 9$$

$$0010 \rightarrow (0011)_2 \Rightarrow 3$$

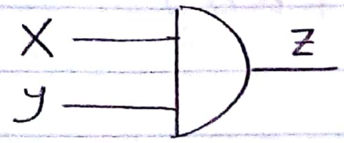
$$0011 \rightarrow (0010)_2 \Rightarrow 2$$

$$= (932)_{10}$$

Scanned by CamScanner

Gates (Switching Algebra):

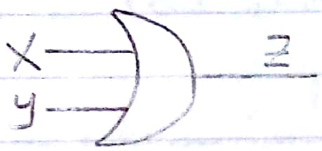
(1) AND



$Z = (X \cdot y)$

0 AND 0 = 0	0 AND 1 = 0	1 AND 0 = 0	1 AND 1 = 1

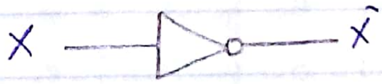
(2) OR



$Z = (X + y)$

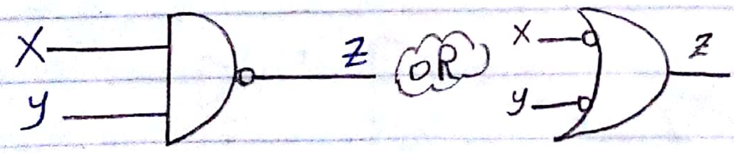
0 OR 0 = 0	0 OR 1 = 1	1 OR 0 = 1	1 OR 1 = 1

(3) Not



inverter (Not gate).

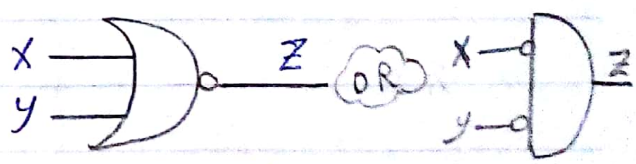
(4) NAND



$Z = (X \cdot y)'$

$= (X' + y')$

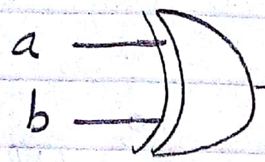
(5) NOR



$Z = (X + y)'$

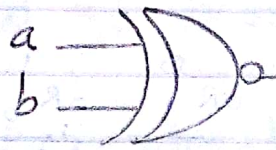
$= (X' \cdot y')$

(6) XOR



$(a \oplus b) = (a'b + ab')$

(7) XNOR.



$(a \oplus b)' = (a'b' + ab)$

The simplifications:

Basic Rules of Boolean Algebra:

1) $A \overset{OR}{+} 0 = A$

2) $A + 1 = 1$

3) $A \overset{AND}{*} 0 = 0$

4) $A * 1 = A$

5) $A \overset{OR}{+} A = A$

6) $A \overset{AND}{*} A = A$

7) $A + A' = 1$

8) $A * A' = 0$

9) $(A')' = A$

10) $A + AB = A$

$A(\underbrace{1 + B})$

$A * 1$

A

11) $A(B + C) = AB + AC$

12) $A + BC = (A + B)(A + C)$

13) $A + A'B = A + B$

$\downarrow$

$\cancel{(A + A')}(A + B) = A + B$

14) $A(A + B) = A$

$A \cdot A + AB$

$A + AB$

$A(1 + B)$

$= A$

Ex :

Simplify $F = x'y' + x'y + xy$

Solution:

$F = x'y + x'y + xy$

$= x'\underbrace{(y' + y)}_{1} + xy$

$= (x' + xy)$

$= \underbrace{(x' + x)}_{1}(x' + y)$

$= x' + y$

Ex:

Simplify: $F = A'B + B'C' + AB + B'C$

$= B\underbrace{(A' + A)}_{1} + B'\underbrace{(C' + C)}_{1}$

$= B * 1 + B' * 1$

$= \underbrace{B + B'}_{1}$

$= 1$

Ex:

Simplify: $f = y + x'z + xy'$

$= (y + xy') + x'z$

$= (y+x)\underbrace{\cancel{(y+y')}}_{1} + x'z$

$= (y+x) + x'z$

$= y + \underbrace{\cancel{(x+x')}}_{1}(x+z)$

$= y + (x+z)$

$= y + x + z$

Ex:

Simplify: $x'z + xy'z + xyz$

$= x'z + xz\underbrace{\cancel{(y'+y)}}_{1}$

$= x'z + xz$

$= z\underbrace{\cancel{(x'+x)}}_{1}$

$= z$

⇒ Consensus term (for minimization).

$\Rightarrow A t_1 \,\&\, A' t_2 \Rightarrow t_1 t_2$

$\Rightarrow \underline{A'} B C + \underline{A} C d \Rightarrow B C d$ is a Consensus term.

$\Rightarrow \underline{A'} \underline{B} C + \underline{A} \underline{B'} C d \Rightarrow$ there isn't any Consensus term because more than (Variable & its Complement) are found.

EX:

Simplify:

$$F = a'b'c' + a'bc' + a'bc + ab'c'$$

$$= b'c'(a' + a) + a'b(c' + c)$$

$$= b'c' + a'b$$

(Consensus Term لايوجد)

EX:

Simplify: $F = a'bc' + abc' + abd + acd$

$$= bc'(a' + a) + abd + acd$$

$$= b\underline{c'} + \cancel{abd} + a\underline{c}d$$

(abd) is a Consensus Term

تم حذفه

$$= bc' + acd$$