classmate

Date ______

Page ______

☆ Data representation:

Magnitude

unsigned — ⊕ only

signed — both ⊕ & ⊖

Complement

1's complement — ⊕ & ⊖

2's complement — ⊕ & ⊖

→ In all 4 rep. ⊕ num. is represented same.

☆ unsigned rep:

$+6 = 110$

$-6 =$ we can't represent in unsigned

☆ signed mag:

$+6 = \boxed{0}110$

$-6 = 1110$

→ so in signed magnitude method write 6 and add zero in-front of number. and convert 0 to 1 so signed magnitude applies.

$+13 = 01101$

$-13 = 11101$

Range of signed magnitude:

$-(2^{n-1}-1)$ to $+(2^{n-1}-1)$

$n = 4$

$-(2^3-1)$ to (2^3-1)

-7 to 7

so, for 4 variable number range is -7 to 7.

H.W. $+5$ and -5, $5 = 0101$, $-5 = 1101$

$+9$ and -9, $9 = 01001$, $-9 = 11001$

$+16$ and -16, $16 = 010000$, $-16 = 110000$

classmate

Date ____
Page ____

☆ represent in 1's compliment:

Range: $-(2^{n-1}-1)$ to $(2^{n-1}-1)$

+6 = 0110 +9 = 01001

-6 = 1001 -9 = 10110

+0 = 0000 (Positive zero)

-0 = 1111 (negative zero)

H.W. +10 = 01010 | +4 = 0100 | +23 = 010111

-10 = 10101 | -4 = 1011 | -23 = 101000

→ so, convert into (-) in 1's compliment put one 0 at starting of number.

☆ 2's compliment :-

+6 = 0110 +4 = 0100

-6 = 1001 1's = 1011

+ 1 + 1

1010 2's comp. 1100 = -4

→ In 2's compliment only +0 is exist, no -0.

so, range is -2^{n-1} to $+(2^{n-1}-1)$

n=4

-8 to 7

-5 ← 1011 ← 2's compliment

0100 ← 1's com

+ 1

0101 = 2's comp = +5, so we can conclude given no. is -5.

H.W.

+5 = 0101

1's = 1010

+ 1

1011

(ii) +8 = 1000

1's = 0111

+ 1

1000 = -8 ⟵ is include in 4 variable.

+8 = 01000 ⟵ because +8 is not valid for 4 digit

+9 = 01001

1's = 10110

2's = +1

= 10111

Binary subtraction using 1's compliment:

A - B = A + (-B)

A = 1100 (12)   12 - 5 = 7

B = 0101 (5)

1's comp of B = 1010 = -B

A + (-B) = 1100

+ 1010

10110

+ 1

0111 = Ans

= 7

S1: convert number to be substracted in 1's compliment
S2: Perform addition.
S3: If the final carry is 1, then add it to the result obtain at step 2. If final carry is 0, result obtained in step 2 is negative and in 1's compliment form.

ex: 2. 0101 - 1100

A = 0101
B = 1100
-B = 0011

1 1 1
0 1 0 1 +5
+1 0 0 1 1 - 12 = -7
0 1 0 0 0 ← so by S3 this is ⊖ve so convert to compliment
1 0 1 1 1
−
−7

2 - 1

0010 - 0001

0010
+ 1110
1 0000
+ 1
0001 ← 1

→ Substruction using 2's compliment :
→ in 2's compliment S-3 is change
S-3 : If final carry is generated then the result is positive and in true form. If final carry not produced, then the result is negative and in its 2's complement form.

(Signed carry)

If . F.C = 1 : neglect the f.c. in 2's comp. method

ex. 1 $(1001)_2 - (0100)_2$

A = 1001
B = 0100 → 4
-B = 1011
+ 1
-B = 1100

1001 +9 ← In this case signed carry act as signed method and if carry is 1 then answer of negative number goes to (+) number and if f.c. is 0 then number is negative.
+ 1100 -4
1 0101 +5
f.c. Ans.

range = -2^{n-1} to $(2^{n-1}-1)$
-8 to +7

ex. 2 0110 - 1011

A = 0110
B = 1011
-B = 0100
+ 1
0101

1
0110 +6
+ 0101 -11
0 1011 (-5)
f.c. (then we have to find 2's complement of it)

0100
+ 1
0101 +5

H.W (i) 0110 - 0100

A = 0110 → 6
B = 0100 → 4
−B = 1011
+ 1
1100

  0110
+ 1100
1 0010 = 2

(ii) (0111) − (1110)

A = 0111 → 7
B = 1110 → 14
−B = 0001
+ 1
0010

  0111 7
+ 0010 −14
0 1001 (−7) Ans.

2's com = 0110
+ 1
0111 (+7)

(iii) 10110 − 1111

A = 10110
B = 01111
−B = 10000
+ 1
10001

  10110
+ 10001
1 0 0111

★ ADD - shift - 3 method for Binary to BCD converse:

15 = 1111 = Binary
15 = 0001 0101 = BCD

Operation	tens	ones	Decimal
or. no			1111
Shift		1	111
Shift		1 1 1	11
Shift		1 1 1 (>4)	1
Add 3		+ 0 1 1	
		1 0 1 0	
Shift	0 0 0 1	0 1 0 1	

(We have to stop shifting when in ones column no. is >4.)

BCD = 15

★ 2421 code :- (one type of BCD)

Decimal digit	Set - I	Set - II
	2 4 2 1	2 4 2 1
0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 0
3	0 0 1 1	0 0 1 1
4	0 1 0 0	0 1 0 0
5	0 1 0 1	1 0 1 1
6	0 1 1 0	1 1 0 0
7	0 1 1 1	1 1 0 1
8	1 1 1 0	1 1 1 0
9	1 1 1 1	1 1 1 1

(1000 not use because this is more significant)

→ this code is called as self-compliment also.
because $9 = (\bar{0})$, $8 = (\bar{1})$, $7 = (\bar{2})$ ...

classmate
Date
Page

H.W. convert $(37)_{10}$ to 2421 code

3 7

0011 1101 = Ans.

(ii) 0 1 0 0 1 1 0 0 1 1 1 0 convert 2421 to decimal

2 4 2 1 2 4 2 1 2 4 2 1

4 6 8 = Ans.

☆ excess -3 code :- (xs-3 code or x3 code)

Decimal → 8-4-2-1 code (BCD code) $\xrightarrow[0011]{ADD}$ excess-3 code

like, BCD only 0 to 9 contain

Decimal	BCD	+0011	xs-3
0	0000	→	0011
1	0001		0100
2	0010		0101
3	0011		0110
4	0100		0111
5	0101		1000
6	0110		1001
7	0111		1010
8	1000		1011
9	1001		1100

→ this is also self compliment code means 0 = ($\bar{9}$), 8 = ($\bar{1}$) ...

Ex. 2 4

 0010 0100
+ 0011 0011
 0101 0111 = 0101 0111 = Ans. = ~~xs-3 code~~

Jul 24

(ii) 6 5 8

0110 0101 1000

+ 0011 0011 0011

1001 1000 1011 = Ans = xs-3 code

→ xs-3 code is only unweighted code which is self-complimenting.

if $w_3 + w_2 + w_1 + w_0 = 9$ then code is self-complimenting

like, 2421 code = 2+4+2+1 = 9, so 2421 code is self complimenting.

but, BCD code = 8421 code = 8+4+2+1 ≠ 9, so not self " ".

→ H.W. (i) 16

0001 0110

+ 0011 0011

0100 1001 = Ans

(ii) 3 9

0011 1001

+ 0011 0011

0110 1100 = Ans.

(iii) 1208

0001 0010 0000 1000

+ 0011 0011 0011 0011

0100 0101 0011 1011 = Ans.

★ xs-3 code addition:-

$(2)_{10} + (5)_{10}$ = 0010 + 0101 (BCD) = 0101 + 1000 (xs-3)

0101 xs-3

+ 1000 xs-3

1101 xs-6

− 0011

1010 = Ans (10)

General

2 $\xrightarrow{xs-3}$ 5

+ 5 $\xrightarrow{xs-3}$ 8

7 $\xrightarrow{xs-6}$ 13

(so need to −3 from sum)

classmate

Date
Page

ex.2 $(27)_{10} + (39)_{10}$

0010 0111   0011 1001   ← BCD

+ 0011 0011   0011 0011

0101 1010   0110 1100   ← xs-3

G2   G1   G1

0101 1010

+ 0110 1100

1 0000 0110

- 0011 + 0011

1001 1001 = 99

for, G1 = f.c. = 1

G2 = f.c. = 2

if f.c. is 1 add 0011

if, f.c is 0 sub. 0011

27

+ 39

66 → normy

+ 33

99 ← xs-3

H.W (i) $(8)_{10} + (9)_{10}$

1000   1001

+ 0011   0011

1011   1100

1011

+ 1100

10111

- 0011

10100 = 20

11

+ 6

23-320

(ii) $(23)_{10} + (33)_{10}$

0010 0011   0011 0011

0011 0011   0011 0011

0101 0110   0110 0110

G2 → 2

0101 0110 ← G1

0110 0110

1011 1100   (both group f.c. is 0)

- 0011 0011

1000 1001 = 89

23

+ 33

56

+ 33

89

~~(iii) $(3)_{10} + (11)_{10}$~~

~~0011   0001 0001~~

~~+ 0011   0011 0011~~

~~0110   0100 0100~~

~~0000   0110~~

~~+ 0110   0100~~

~~0110   1010~~

~~- 0011   0011 1~~

~~0001   0111~~

☆ Gray code:-

- Also known as <u>Reflected Binary code.</u>
- <u>unweighted code</u>
- <u>unit distance code</u> & <u>minimum error code</u>

⟹ two ~~[illegible]~~ succ. value differ in only 1 bit. Binary no. is converted to G.C to reduce switching operation.

- <u>"Cyclic code"</u>
- all under the other name of G.C.

⟹ Binary to Gray-code conversion:-

S-1: Record the M.S.B as it is

S-2: add the MSB to the next bit, record sum and neglect the carry.

S-3: Repeat the process.

ex.1 Convert 1011 to gray

M.S.B $b_3 + b_2\ b_1 + b_0$

1 0 1 1   here carry is 1 but neglect

1 0 1

$\boxed{1\ 1\ 1\ 0}$ ← gray code

$g_3\ g_2\ g_1\ g_0$

XOR's

A	B	$Y = A\bar{B} + \bar{A}B$
0	0	0
0	1	1
1	0	1
1	1	0

$g_3 = b_3$

$g_2 = b_2 \oplus b_3$

$g_1 = b_1 \oplus b_2$

$g_0 = b_0 \oplus b_1$

ex.2 1 1 1 0 $1 \oplus 1 = 0$

$1 \oplus 0 = 1$

$\boxed{1\,0\,0\,1}$ ← Gray code

H.W (i) 1 0 0 1 (ii) 1 0 1 0

$\boxed{1\,1\,0\,1}$ ← Gray code 1 1 1 1

(iii) 1 1 1 1

1 0 0 0

★ Gray code to Binary

S-1: Record MSB as it is
S-2: Add MSB to the next bit of Gray code, record the sum and neglect the carry.
S-3: Repeat the process.

ex.(i) 1 1 1 0 ← G.C.

1 0 1 1 ← Binary

EX.2 $g_3\ g_2\ g_1\ g_0$
1 0 0 1
1 1 1 0
$b_3\ b_2\ b_1\ b_0$

$b_3 = g_3$
$b_2 = b_3 \oplus g_2$
$b_1 = b_2 \oplus g_1$
$b_0 = b_1 \oplus g_0$

H.W (i) 1 1 0 0 ← G.C.

1 0 0 0 ← Bin

(ii) 1 0 1 1 (iii) 1 0 0 0

1 1 0 1 1 1 1 1

★ Logic gates:- It is a physical device which perform logic ope. on one or more logical inputs, and produces a single logical output

(i) Basic gates : Not, And & Or
(ii) Universal gate : NAND & NOR
(iii) Arithmetic gates : XOR & X-NOR

★ K'map (karnaugh map)

$F = A + B\bar{C}$

A	B	C	F	
0	0	0	0	m0
0	0	1	0	m1
0	1	0	1	m2
0	1	1	0	m3
1	0	0	1	m4
1	0	1	1	m5
1	1	0	1	m6
1	1	1	1	m7

because only one bit change

A \ BC	00	01	11	10
0	0	0	0	1
1	1	1	1	1

$F = A + B\bar{C}$

16 ↓ 8 Pair ↓ 4 ↓ 2 ↓ 1

ex1 $f(A,B,C) = \Sigma m(1,3,5,7)$

1) find out no. of variable
$n = 3 \, (A, B, C)$
Cells no. $= 2^3 = 8$

A \ BC	00	01	11	10
0	0^0	1^1	1^3	0^2
1	0^4	1^5	1^7	0^6

$F = C$

(ii) $F(A,B,C) = \Sigma(0,1,2,4,7)$

A \ BC	00	01	11	10
0	1	1	0	1
1	1	0	1	0

$F = \bar{B}\bar{C} + \bar{A}\bar{B} + ABC + \bar{A}\bar{C}$

(iii) $F(A,B,C) = \sum m(1,3,6,7)$

A \ BC	00	01	11	10
0	0	1	1	0
1	0	0	1	1

$F = A \cdot B + \bar{A} C$

(iv) $F(A,B,C) = \sum m(0,1,5,6,7)$

A \ BC	00	01	11	10
0	1	1	0	0
1	0	1	1	1

$F = \bar{A}\bar{B} + AC + AB$

$F = \bar{A}\bar{B} + AB + \bar{B}C$

So, the result is minimum but not be same.

★ Implicants:- the group of 1 is called as implicants.
ex- 1, 2, 4, 8, 16 ..

Prime implicants:- Largest possible group of 1

Essential prime Implicant:- atleast there is a single 1 which can not be combined in any other way.

ex.

AB \ CD				
				1
			1	1
		1	1	
		1		

I = EPI (Essential prime)
II = NEPI (Not ---)
III = EPI

→ 4 variable k'map:

(i) $f(A,B,C,D) = \sum m(0,2,3,7,11,13,14,15)$

AB \ CD	00	01	11	10
00	1		1	1
01			1	
11		1	1	1
10			1	

because after

4 → 12

0100 1100 ✓

4 → 8

0100 1000 ✗

$F = CD + ABD + ABC + \bar{A}\bar{B}\bar{D}$

combine

2 1's ⇒ 1 variable loss

4 1's ⇒ 2 ''

8 1's ⇒ 3 ''

16 1's ⇒ 4 ''

☆ Don't care in k'map:

ex: $F(A,B,C) = \sum m(2,3,4,5) + \sum d(6,7)$

A \ BC	00	01	11	10
0	0	0	1	1
1	1	1	x	x

consider 1 in case of minterm

d → 1

or

d → 0

→ we taking 1 to d, reduce when use minterm.

$F = B + A$

→ we taking 0 to d, reduce when use maxterm.

☆ K-map using Max-terms:

A \ BC	00	01	11	10
0	0	0	0	1
1	1	1	1	1

$$\bar F = \bar A \bar B + \bar A C$$
$$F = \overline{(\bar A \bar B + \bar A C)}$$
$$= \overline{(\bar A \bar B)} \cdot \overline{(\bar A C)}$$
$$= (A+B)\cdot(A+\bar C)$$

POS

$F = A + B\bar C$ (distributive law)

Ex-2 $F(A,B,C,D) = \sum m(1,3,4,5,9,11,14,15)$

$F(A,B,C,D) = \prod M(0,2,6,7,8,10,12,13)$

AB \ CD	00	01	11	10
00	0	1	1	0
01	1	1	0	0
11	0	0	1	1
10	0	1	1	0

$\cancel{\bar F = A\bar C\bar D + AB\bar C + \bar A BC + \bar B\bar D}$

$\cancel{F = (A\bar C\bar D + AB\bar C + \bar A BC + \bar B \bar D)'}$

$\cancel{F = (\bar A + C + D)\cdot(\bar A + \bar B + C)\cdot(A + \bar B + \bar C)\cdot B}$

$$\bar F = AB\bar C + \bar A BC + \bar B \bar D$$
$$F = (\bar A + \bar B + C)\cdot(A + \bar B + \bar C)\cdot(B + D)$$

★ 5 variable k'map :-

F(A,B,C,D,E)

ABCDE

0
0
·
·
0

1
1
·
·
1

A = 0 ⟶ A = 1

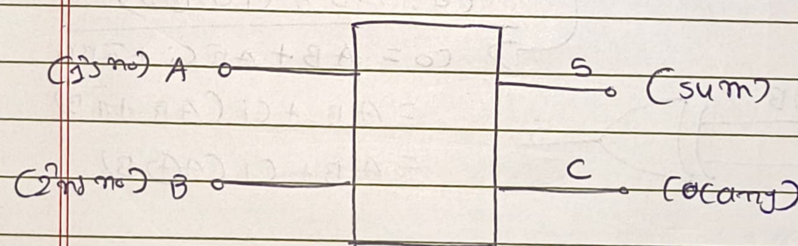
BC \ DE	00	01	11	10
00	1		1	1
01		1	1	
11			1	1
10	1		1	1

BC \ DE	00	01	11	10
00	1			1
01				
11		1	1	1
10	1		1	1

$F = \overline{C}\,\overline{E} + BD + \overline{A}DE + BCE$

because table-2 is not participale so $\overline{A}$ is consider

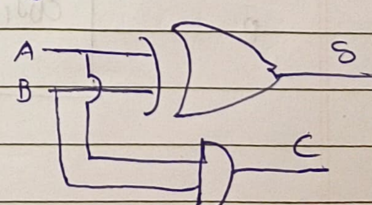
★ Half adder



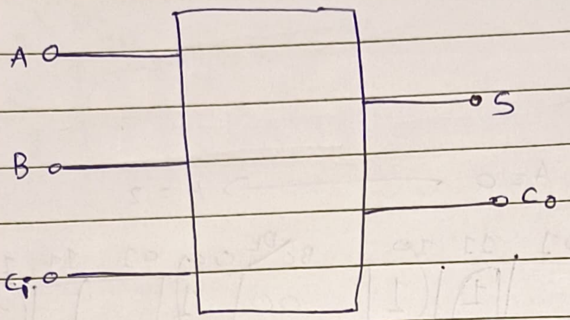
→ single bit no's
→ does not take carry from previous sum

A	B	S	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$S = A \oplus B$
$C = AB$



☆ Full adder:



A	B	Ci	S	Co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

for, S

A \ BCi	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$F = A \oplus B \oplus C$

Check board cond.

for, Co

A \ BCi	00	01	11	10
0	0	0	1	0
1	0	1	1	1

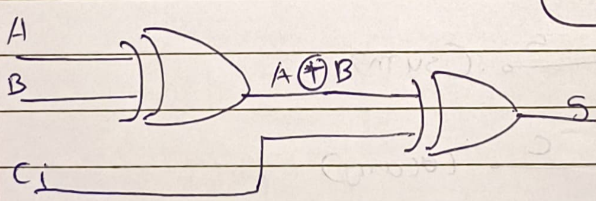
$C_o = AC_i + AB + BC_i$

$C_o = AB + C_i(A \oplus B)$

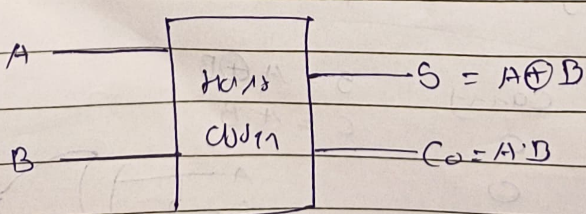
$C_o = AB + A\bar{B}C_i + \bar{A}BC_i$

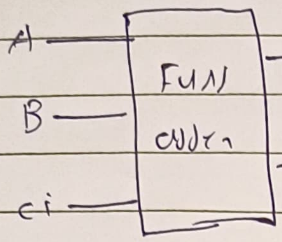
$= AB + C_i(A\bar{B} + \bar{A}B)$

$= AB + C_i(A \oplus B)$



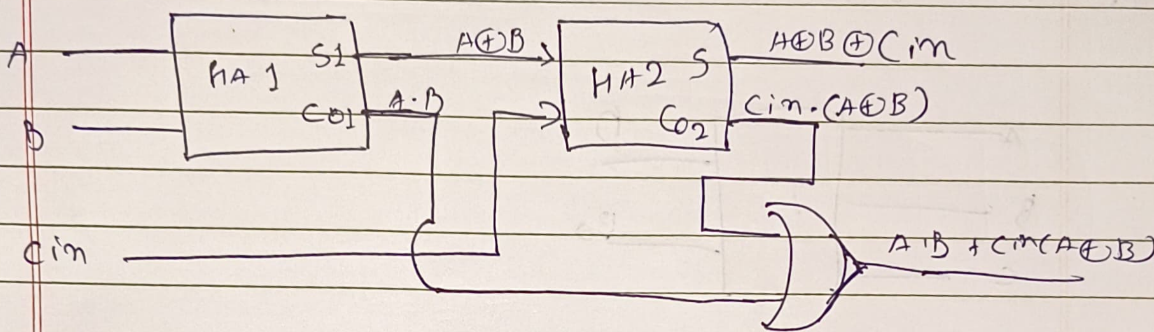
☆ Full-adder using half-adder:





$S = A \oplus B \oplus C$

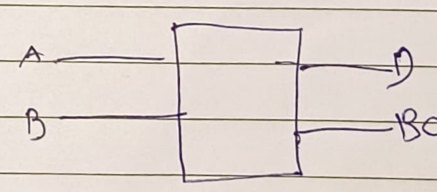
$C_o = A \cdot B + C_{in}(A \oplus B)$ or $A \cdot B + B \cdot C_{in} + A \cdot C_{in}$



Let's prove $A \cdot B + C_{in}(A \oplus B)$ is $= A \cdot B + B \cdot C_{in} + A \cdot C_{in}$

$$
\begin{aligned}
A \cdot B + C_{in}(A \oplus B) &= A \cdot B + C_{in}(AB' + A'B) \\
&= AB + AB'C_{in} + A'BC_{in} \\
&= A(B + B'C_{in}) + A'B \cdot C_{in} \quad (\because x + x'y = x + y) \\
&= A(B + C_{in}) + A'B \cdot C_{in} \\
&= AB + AC_{in} + A'B \cdot C_{in} \\
&= AB + (A + A'B)C_{in} \quad (\because \text{———}) \\
&= AB + (A + B) \cdot C_{in} \\
&= AB + A \cdot C_{in} + BC_{in}
\end{aligned}
$$

Half Subtractor:-

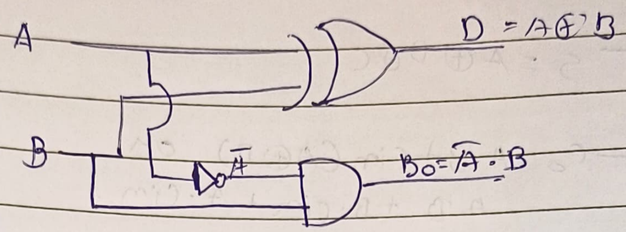


Half subtractor

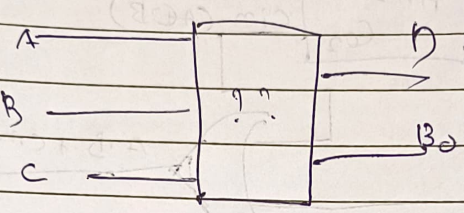
A	B	D	Bo
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$D = A \oplus B$

$B_o = \overline{A}B$



☆ Full substractor



A	B	C	D	B_o
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$D = A \oplus B \oplus C$

A \ BC	00	01	11	10
0	0	1	1	1
1	0	0	1	0

$B_o = \bar{A}C + BC + \bar{A}B$