Information Technology University of the Punjab
SE201T Digital Logic Design – Fall 2024
Assignment 2 [CLO1, CLO2]

Deadline: 11th October, 2024 (Start of the Friday class) **Total Marks: 100**

Instructions:
1. Use A4 size sheets to prepare the solution.
2. You must attempt all questions by hand. Use an ink pen or ball point. Word processors are not advised to prepare the solution.
3. Clearly label the start of all questions and their parts. Also, make sure to highlight the final answer in each part.
4. This is an individual assignment so every student must submit their own solution.
5. You can take help from the textbook/reference books. Discussion among peers without showing the solution is acceptable, but you must attempt individually. Plagiarism, if found, will be dealt with according to the ITU anti-plagiarism policy.
6. Make sure to submit it by the deadline. Late submissions will not be accepted.

Q1 [CLO1: Number system basics and conversions]

a. List the binary, octal and hexadecimal numbers from $(12)_{10}$ to $(20)_{10}$. [3]

Q1a.

Decimal	Binary	Octal	Hexadecimal
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14

b. What is the exact number of bits in a memory that contains (i) 128K bits; (ii) 32M bytes? [1]

(i) 128K bits
$= 128 \times 2^{10}$
$= 128 \times 1024$
$= 131072$ bits

(ii) 32 Mbytes
$= 32 \times 2^{20} \times 8$
$= 268435456$ bit

c. How many different possible numbers can be stored in a register having 10 bits? [1]

Q1c.

The total possible numbers that can be stored in a 10-bit register is: $2^{10} = 1024$

d. What is the range of signed numbers that can be represented by a 16-bit binary number that is in (i) sign-magnitude form; (ii) 1's complement form; (iii) 2's complement form? [3]

Q1d. Sign-magnitude : $+0$ — $+32767$, -0 — -32767

1's complement: $+0$ — $+32767$, -32767 — -0

2's complement : $+0$ — $+32767$, -32768 — -1

e. Convert the following numbers from the given base to the other three bases listed in the table: [6]

Decimal	Binary	Octal	Hexadecimal
369.3125	?	?	?
?	10111101.101	?	?
?	?	326.5	?
?	?	?	F3C7.A

Q1.e.

$(369.3125)_{10} = (\ ?\ )_2 = (\ ?\ )_8 = (\ ?\ )_{16}$

Conversion to binary.

Whole number:

2	369
2	184 - 1
2	92 - 0
2	46 - 0
2	23 - 0
2	11 - 1
2	5 - 1
2	2 - 1
	1 - 0

Fraction:

2 ×	0.3125
2 ×	0.625 - 0
2 ×	0.25 - 1
2 ×	0.5 - 0
	1.0 - 1

So, $(369.3125)_{10} = (101110001.0101)_2$

$$\underbrace{101}_{5}\,\underbrace{110}_{6}\,\underbrace{001}_{1}\,.\,\underbrace{010}_{2}\,\underbrace{100}_{4}$$

Octal: $(561.24)_8$

$$\underbrace{0001}_{1}\,\underbrace{0111}_{7}\,\underbrace{0001}_{1}\,.\,\underbrace{0101}_{5}$$

Hexadecimal: $(171.5)_{16}$

$(10111101.101)_2 = (\ ?\ )_8 = (\ ?\ )_{16} = (\ ?\ )_{10}$

$(\ ?\ )_8$

$$\underbrace{010}_{2}\,\underbrace{111}_{7}\,\underbrace{101}_{5}\,.\,\underbrace{101}_{5}$$

$(?)_8 = (275.5)_8$

$$\underbrace{1011}_{B}\,\underbrace{1101}_{D}\,.\,\underbrace{1010}_{A}$$

$(?)_{16} = (BD.A)_{16}$

$$
\begin{aligned}
(?)_{10} &= B \times 16 + D + A \times 16^{-1} \\
&= 11 \times 16 + 13 + 10 \times 16^{-1} \\
&= (189.625)_{10}
\end{aligned}
$$

$(326.5)_8 = (\ ?\ )_2 = (\ ?\ )_{16} = (\ ?\ )_{10}$

$$\underset{011}{\underline{3}} \quad \underset{010}{\underline{2}} \quad \underset{110}{\underline{6}} \quad \cdot \quad \underset{101}{\underline{5}}$$

$$(\ ?\ )_2 = (011010110.101)_2$$

$$\underbrace{0000}_{0}\ \underbrace{1101}_{D}\ \underbrace{0110}_{6} \cdot \underbrace{1010}_{A}$$

$$(?)_{16} = (0D6.A)_{16}$$

$$(?)_{10} = D \times 16 + 6 \times 16^0 + A \times 16^{-1}$$
$$= 208 + 6 + 0.625$$
$$= (214.625)_{10}$$

$(F3C7.A)_{16}$

$$(\ ?\ )_2 = (\underset{F}{\underline{1111}}\ \underset{3}{\underline{0011}}\ \underset{C}{\underline{1100}}\ \underset{7}{\underline{0111}} \cdot \underset{A}{\underline{1010}})_2$$

$(\ ?\ )_8$:

$$\underline{001}\,\underline{111}\ \underline{001}\,\underline{1\ 11}\,\underline{00\ 0}\,\underline{111} \cdot \underline{101}\ \underline{000}$$

$$(?)_8 = (1\ 7\ 1\ 7\ 0\ 7 \,.\, 5\ 0)_8$$

$$(\ ?\ )_{10} = F \times 16^3 + 3 \times 16^2 + C \times 16^1 + 7 \times 16^0 + A \times 16^{-1}$$
$$= 61440 + 768 + 192 + 7 + 0.625$$
$$= (62407.625)_{10}$$

f. Express each decimal number as an 8-bit number in the sign-magnitude, 1's complement and the 2's complement form: [4.5]

i. +101
ii. -68
iii. -125

Q1f.

	Sign-magnitude	1's complement	2's complement
+101	01100101	01100101	01100101
-68	11000100	10111011	10111100
-125	~~11111101~~ 11111101	10000010	10000011

g. Determine the decimal value of the following 8-bit signed numbers in each case of sign-magnitude, 1's complement and 2's complement representation: [4.5]

i. 10011001
ii. 01110100
iii. 10111111

Q1g.

	Sign-mag	1's comp	2's comp
10011001	-25	-102	-103
01110100	+116	+116	+116
10111111	-63	-64	-65

h. The following calculation was performed by a particular breed of unusually intelligent chicken. If the radix r used by the chicken corresponds to its total number of toes, how many toes does the chicken have? [2]

$$((34)_r + (24)_r) \times (21)_r = (1338)_r$$

Q1h. $((34)_r + (24)_r) \times (21)_r = (1338)_r$

Convert L.H.S to decimal:

$((3\times r + 4) + (2\times r + 4)) \times (2\times r + 1)$

$= (5r + 8)(2r + 1)$

$= 10r^2 + 21r + 8$

Convert R.H.S to decimal

$(1338)_r = 1r^3 + 3r^2 + 3r + 8$

L.H.S = R.H.S

$10r^2 + 21r + \cancel{8} = r^3 + 3r^2 + 3r + \cancel{8}$

$r^3 - 7r^2 - 18r = 0$

$r(r^2 - 7r - 18) = 0$

so, $r = 0$ or $r^2 - 7r - 18 = 0$

$r^2 - 9r + 2r - 18 = 0$

$r(r-9) + 2(r-9) = 0$

$(r-9)(r+2) = 0$

so, $\boxed{r = 9}$ ✓

or $\boxed{r = -2}$ X → Ignore -ve base

Chicken has 9 toes

Q2 [CLO1: Number system arithmetic]

a. Perform the following arithmetic operations on unsigned binary numbers: [3]

  i. 11110011 + 11101111
  ii. 1111001 × 1101
  iii. 10101110 ÷ 1100

Q2a.

(i)
```
     1 1 1 1 1 1 1
     1 1 1 1 0 0 1 1
   + 1 1 1 0 1 1 1 1
   -----------------
   1 1 1 1 0 0 0 1 0
```

(ii)
```
             1 1 1 1 0 0 1
           ×       1 1 0 1
           ---------------
             1 1 1 1 0 0 1
           0 0 0 0 0 0 0 X
         1 1 1 1 0 0 1 X X
       1 1 1 1 0 0 1 X X X
     ---------------------
     1 1 0 0 0 1 0 0 1 0 1
```

(iii)

```
               1110
       ____________
1100 ) 10101110
       1100
       ----
        10011
         1100
        -----
         01111
          1100
         -----
           110
```

Quotient : 1110

Remainder : 110

b. Perform the following operations on binary numbers in 2's complement form: [4]

i. 00110011 – 00010000

ii. 01100101 – 11101000

iii. 01101010 × 11110001 (Give result in 16 bits)

iv. 10001100 + 00111001

Q2 b. (i) 00110011 – 00010000

= 00110011 + (– 00010000)

Taking 2's complement.

00110011 + 11110000

```
    1 1 1 1
    0 0 1 1 0 0 1 1
    1 1 1 1 0 0 0 0
  -----------------
  1 0 0 1 0 0 0 1 1
```

Discard as there is no over flow. Result is 00100011

(ii) 01100101 − 1110 1000

= 0110 0101 + (−1110 1000)

Taking 2's complement.

0 1100101 + (00011 000)

```
  0 1 1 0 0 1 0 1
  0 0 0 1 1 0 0 0
  ---------------
  0 0 1 1 1 1 0 1
```

No overflow; Result is 01111101

(iii) 0110 1010 × 1111 0001

The multiplier 11110001 is negative due to 1 at MSB. Let's take its 2's complement to obtain its +ve value.

2's complement: 00001111

Now multiplying

```
          0 1 1 0 1 0 1 0
          0 0 0 0 1 1 1 1
          ---------------
          0 1 1 0 1 0 1 0
        0 1 1 0 1 0 1 0 x
      0 1 1 0 1 0 1 0 x x
    0 1 1 0 1 0 1 0 x x x
    ---------------------
    1 1 0 0 0 1 1 0 1 1 0
```

In 16 bits : 0000 0110 0011 0110

Take 2's complement to obtain the −ve value:

1111 1001 1100 1010

c. Perform the following subtractions: [3]

i. $A5_{16} - 98_{16}$
ii. $F1_{16} - A6_{16}$
iii. $653_8 - 456_8$

Q2c.

(i) $A5_{16} - 98_{16}$

$$\begin{array}{rr} 9\cancel{A} & {}^{1}5 \\ - \;\; 9 & 8 \\ \hline (0 & D)_{16} \end{array}$$

(ii) $F1_{16} - A6_{16}$

$$\begin{array}{rr} E\cancel{F} & {}^{1}1 \\ A & 6 \\ \hline (4 & B)_{16} \end{array}$$

(iii) $653_8 - 456_8$

$$\begin{array}{rrr} 5\cancel{6} & \overset{14}{\cancel{5}} & {}^{1}3 \\ 4 & 5 & 6 \\ \hline (1 & 7 & 5)_8 \end{array}$$

Q3 [CLO 1: Binary-coded decimal and other digital codes]

a. Represent the decimal number 6,248 in the following codes: [2]
   i. BCD
   ii. excess-3 code
   iii. 2421 code
   iv. 6311 code

Q3 a.

(i) BCD:

6 2 4 8

0110 0010 0100 1000

(ii) Excess-3:

6+3 2+3 4+3 8+3

1001 0101 0111 1011

(iii) 2421:

6 2 4 8

0+4+2+0 0+0+2+0 0+4+0 0 2+4+2+0

0110 0010 0100 1110

(iv) 6311

6 2 4 8

6+0+0+0 0+0+1+1 0+3+0+1 6+0+1+1

1000 0011 0101 1011

b. The following is a string of ASCII characters whose bit patterns have been converted into hexadecimal for compactness: 73 F4 E5 76 E5 4A EF 62 73. Of the eight bits in each pair of digits, the leftmost is a parity bit. The remaining bits are the ASCII code. [4]

   i. Convert the string to bit form and decode the ASCII.

   ii. Determine the parity used: odd or even?

Q3b.

(i)

7 3 F 4 E 5 7 6 E 5 4 A

0111 0011 (s) 1111 0100 (t) 1110 0101 (e) 0111 0110 (v) 1110 0101 (e) 0100 1010 (J)

E F 6 2 7 3

1110 1111 (o) 0110 0010 (b) 0111 0011 (s)

steveJobs

(ii) Parity can be found by looking at any one character, e.g. 73 (0111 0011)
Number of 1s are odd and parity bit (MSB) is 0, so odd parity is being used.

c. Add the following BCD numbers and state the result in decimal: [3]

i. 01010001 + 01011000
ii. 10011000 + 10010111
iii. 010101100001 + 011100001000

Q3c.

(i)

```
        0101 0001
        0101 1000
        ---------
        1010 1001
      +  110
      ---------
0001    0000
```

so result is $(0001\ 0000\ 1001)_{BCD}$

(ii)

```
       1001 1000
       1001 0111
      -----------
      10001 1111
      + 110  + 110
      -----  -----
      11000  0101
```

so result is $(0001\ 1001\ 0101)_{BCD}$

(iii)

```
     0101   0110   0001
     0111   0000   1000
    -------------------
     1100   0110   1001
     +110
    -----
1    0010
```

Result: $(0001\ 0010\ 0110\ 1001)_{BCD}$

d. Encode these Unicode code points in UTF-8. Show the binary and hexadecimal value for each encoding: [3]
i. U+0040
ii. U+00A2
iii. U+1F6B2

(i) U+0040 is in range U+0000 0000 — U+0000 007F

so encoding would be of this form

$\underline{0}xxxxxxx$

$40 = (0100\ 0000)_2$

so, encoding is $\underline{0}100\ 0000$

(ii) U+00A2

Range: B U+00000080 — U+000007FF

A2 in binary $\underbrace{00010}\underbrace{100010}$

$\underline{110}xxxxx$ $\underline{10}xxxxxx$

$\underline{110}00010$ $\underline{10}100010$

(iii) U+1F6B2

Range: U+00010000 — U+0010FFFF

1F6B2 = $\underbrace{00001\ 1111}\ \underbrace{0110\ 10}\underbrace{11\ 0010}$

Encoding template

11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

$\underline{11110}000$ $\underline{10}011111$ $\underline{10}011010$ $\underline{10}110010$

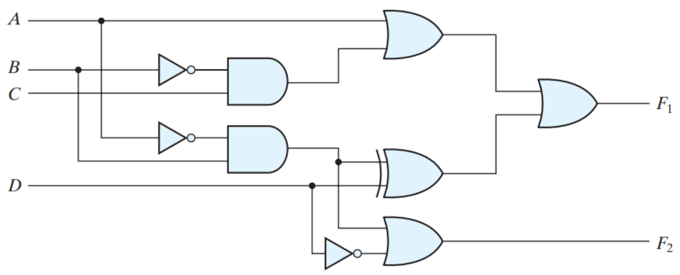
e. List the binary-reflected hexadecimal Gray code. [3]

Q3e.

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

Q4 [CLO2: Combinational circuit analysis and design]

a. Consider the combinational circuit shown in the figure below: [5]



i. List the truth table for F_1 and F_2.

ii. From the truth table, find the Boolean equations for F_1 and F_2.

Q4a.

(i)

A	B	C	D	F_1	F_2
0	0	0	0	0	1
0	0	0	1	1	0
0	0	1	0	1	1
0	0	1	1	1	0
0	1	0	0	1	1
0	1	0	1	0	1
0	1	1	0	1	1
0	1	1	1	0	1
1	0	0	0	1	1
1	0	0	1	1	0
1	0	1	0	1	1
1	0	1	1	1	0
1	1	0	0	1	1
1	1	0	1	1	0
1	1	1	0	1	1
1	1	1	1	1	0

From above table: $F_1 = \Sigma m(1,2,3,4,6,8,9,10,11,12,13,14,15)$

$F_2 = \Sigma(0,2,4,5,6,7,8,10,12,14)$

(ii)

F_1

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

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

$F_1 = A + C\bar{D} + \bar{B}D + B\bar{D}$ $\qquad F_2 = \bar{A}B + \bar{D}$

b. Design a combinational circuit that accepts a 4-bit number and generates a 3-bit binary number output that approximates the square root of the number. For example, if the square root is 3.5 or larger, give a result of 4. If the square root is < 3.5 and ≥ 2.5, give a result of 3. [5]

Q4b.

Num	A	B	C	D	Q_2	Q_1	Q_0	Sqrt	Approx. Sqrt.
0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	1	1	1
2	0	0	1	0	0	0	1	1.414	1
3	0	0	1	1	0	1	0	1.73	2
4	0	1	0	0	0	1	0	2	2
5	0	1	0	1	0	1	0	2.23	2
6	0	1	1	0	0	1	0	2.44	2
7	0	1	1	1	0	1	1	2.64	3
8	1	0	0	0	0	1	1	2.82	3
9	1	0	0	1	0	1	1	3	3
10	1	0	1	0	0	1	1	3.16	3
11	1	0	1	1	0	1	1	3.31	3
12	1	1	0	0	0	1	1	3.46	3
13	1	1	0	1	1	0	0	3.6	4
14	1	1	1	0	1	0	0	3.74	4
15	1	1	1	1	1	0	0	3.87	4

$Q_2 = \Sigma m(13, 14, 15)$

$Q_1 = \Sigma m(3,4,5,6,7,8,9,10,11,12)$

$Q_0 = \Sigma m(1,2,7,8,9,10,11,12)$

Q_2

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

Q_1

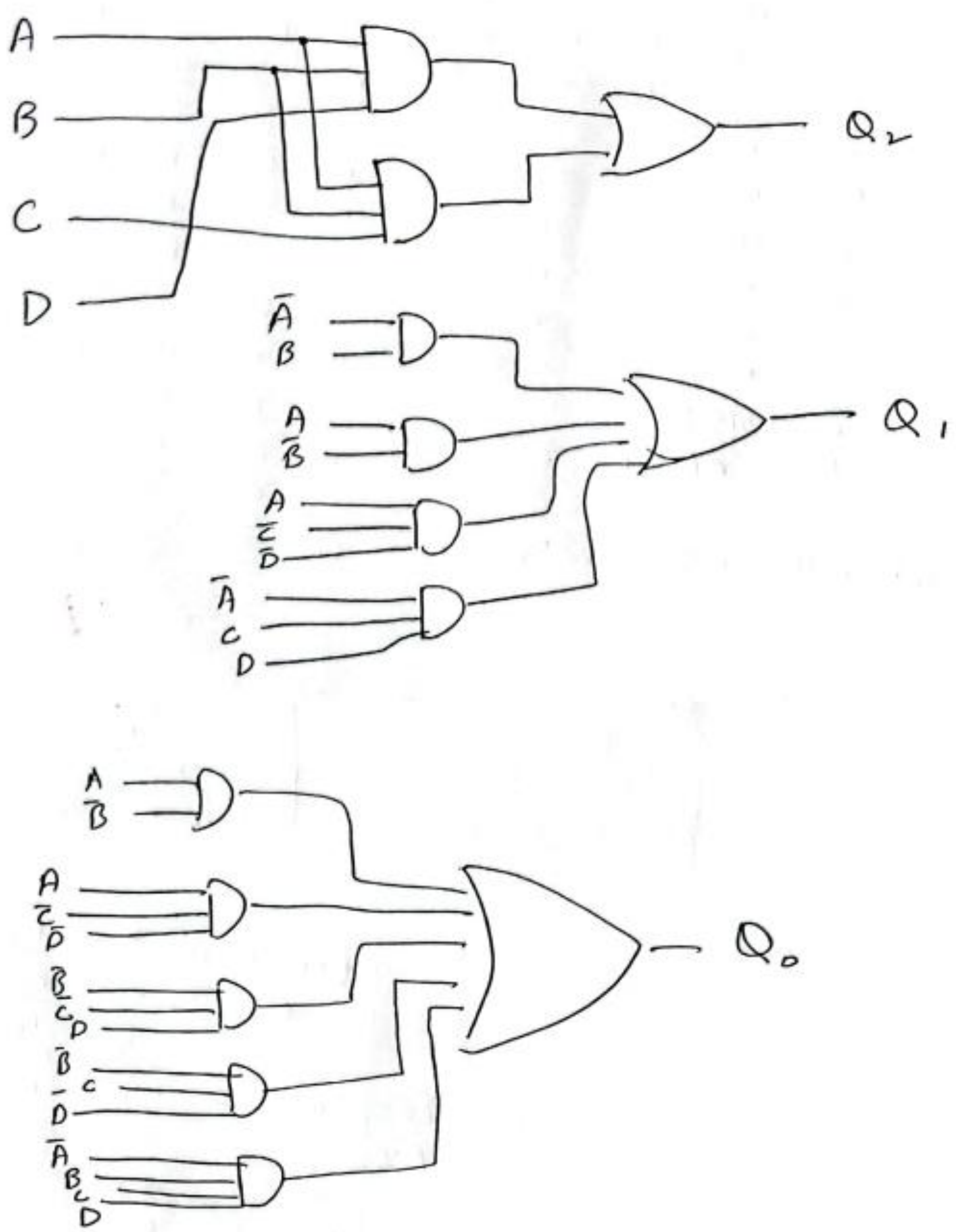
AB \ CD	00	01	11	10
00			1	
01	1	1	1	1
11	1			
10	1	1	1	1

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

$Q_2 = ABD + ABC$

$Q_1 = \bar{A}B + A\bar{B} + A\bar{C}\bar{D} + \bar{A}CD$

$Q_0 = A\bar{B} + A\bar{C}\bar{D} + \bar{B}\bar{C}D + \bar{B}C\bar{D} + \bar{A}BCD$

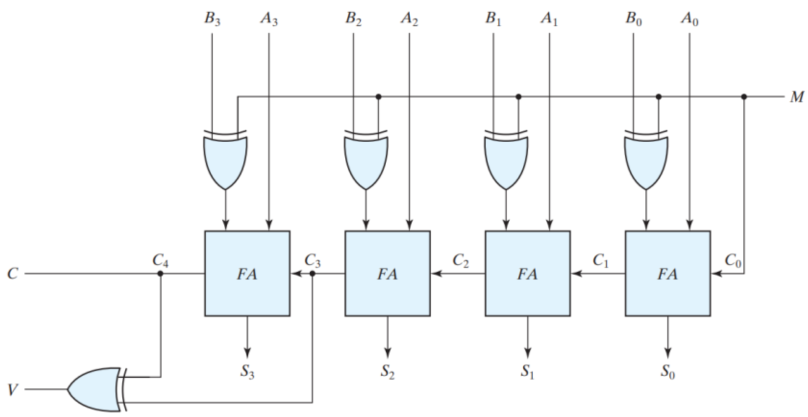


Q5 [CLO2: Adders]

a. The adder-subtractor circuit (see figure below) has the values of *M* and data inputs *A* and *B* given in this table:

Case	*M*	*A*	*B*
(a)	0	0111	0110
(b)	0	1000	1001
(c)	1	1100	1000
(d)	1	0101	1010
(e)	1	0000	0001

In each case, determine the values of the four SUM outputs, the carry *C*, and overflow *V*. [5]



Case(a)

Since $M = 0$, A and B will be added.

$$\begin{array}{r} 0\,1\,1\,0\,0 \rightarrow C_0 \text{ (as } M = C_0) \\ 0\,1\,1\,1 \\ 0\,1\,1\,0 \\ \hline 1\,1\,0\,1 \end{array}$$

So, $S_3 S_2 S_1 S_0 = 1101$

As, $C_3 = 1$ and $C_4 = 0$, $V = 1 \oplus 0 = 1$

and $C = C_4 = 0$

Case(b)

$M = 0$, so A and B are added.

$$\begin{array}{r} 1\,0\,0\,0\,0 \\ 1\,0\,0\,0 \\ 1\,0\,0\,1 \\ \hline 1\,0\,0\,0\,1 \end{array}$$

$S_3 S_2 S_1 S_0 = 0001$

$C = C_4 = 1$

$V = 1 \;\; (C_4 \oplus C_3 = 1 \oplus 0 = 1)$

Case (c)

$M = C_0 = 1$ so A, one's complement of B and C_0 will be added.

$$
\begin{array}{r}
1\,1\,1\,1\,1 \\
1\,1\,0\,0 \\
0\,1\,1\,1 \\
\hline
1\,0\,1\,0\,0
\end{array}
$$

$$S_3 S_2 S_1 S_0 = 0100$$

$$C = C_4 = 1$$

$$V = 0 \; (\underset{1}{C_4} \oplus \underset{1}{C_3})$$

Case (d)

$M = C_0 = 1$, so A, one's complement of B and C_0 will be added.

$$
\begin{array}{r}
0\,1\,0\,1\,1 \\
0\,1\,0\,1 \\
0\,1\,0\,1 \\
\hline
1\,0\,1\,1
\end{array}
$$

$$S_3 S_2 S_1 S_0 = 1011$$

$$C = C_4 = 0$$

$$V = C_4 \oplus C_3 = 1$$

Case (c)

$M = C_0 = 1$, so A, one's complement of B and C_0 will be added.

$$\begin{array}{ccccc} 0 & 1 & 0 & 0 & 1 \\ & 0 & 0 & 0 & 0 \\ & 1 & 1 & 1 & 0 \\ \hline & 1 & 1 & 1 & 1 \end{array}$$

$S_3 S_2 S_1 S_0 = 1111$

$C = C_4 = 0$

$V = C_4 \oplus C_3 = 0$

b. If carry propagate and carry generate are defined as

$$P_i = A_i + B_i$$
$$G_i = A_i B_i$$

respectively, show that the output carry and output sum of a full adder becomes: [3]

$$C_{i+1} = (C_i' G_i{}' + P_i')'$$
$$S_i = (P_i G_i{}') \oplus C_i$$

$P_i = A_i + B_i$

$G_i = A_i . B_i$

In terms of new definitions of ~~$\emptyset$~~ P_i and G_i, we ~~are given~~ have to prove that: $C_{i+1} = (C_i' G_i' + P_i')'$ — 1

and $S_i = (P_i G_i') \oplus C_i$ — 2

Now from theory/book, we know that:

$C_{i+1} = A_i B_i + C_i \cdot (A_i \oplus B_i)$ — 3

$= A_i B_i + A_i C_i + B_i C_i$ (original representation of C_{i+1} from which the above is obtained)

and $S_i = A_i \oplus B_i \oplus C_i$ — 4

So, we can take equations 1 and 2 and substitute P_i and G_i and try to obtain the equations 3 and 4.

$$C_{i+1} = (C_i' G_i' + P_i')'$$

Let's substitute G_i and P_i

$$C_{i+1} = [C_i'(A_iB_i)' + (A_i + B_i)']'$$

$$= [C_i'(A_i' + B_i') + A_i'.B_i']'$$

$$= [C_i'A_i' + C_i'B_i' + A_i'B_i']'$$

$$= (A_i'C_i')'.(C_i'B_i')'.(A_i'.B_i')'$$

$$= (A_i + C_i)(C_i + B_i)(A_i + B_i)$$

$$= (A_iC_i + A_iB_i + C_i + B_iC_i)(A_i + B_i)$$

$$= A_iC_i + A_iB_i + A_iC_i + A_iB_iC_i$$
$$+ A_iB_iC_i + A_iB_i + B_iC_i + B_iC_i$$

$$= A_iB_i + A_iC_i + B_iC_i + A_iB_iC_i$$

By absorption:

$$= A_iB_i + A_iC_i + B_iC_i$$

which is equal to 3

Now,

$$S_i = (P_i G_i') \oplus C_i$$

$$= [(A_i + B_i)(A_i B_i)'] \oplus C_i$$

$$= [(A_i + B_i).(A_i' + B_i')] \oplus C_i$$

$$= (A_i A_i' + A_i B_i' + A_i' B_i + B_i . B_i') \oplus C_i$$

$$= (A_i B_i' + A_i' B_i) \oplus C_i$$

$$= A_i \oplus B_i \oplus C_i$$

which is equal to S_i in (4)

c. Each of the eight full-adders in an 8-bit ripple carry adder exhibits the following propagation delay:

A to *S* and *Cout*:	20 ns
B to *S* and *Cout*:	20 ns
Cin to *S*:	30 ns
Cin to *Cout*:	25 ns

Determine the maximum total time for the addition of two 8-bit numbers. [2]

Cin to Cout is 25 ns

so,

Cin_0 to $Cout_0$ is 25 ns

$(Cout_0 = Cin_1)$ to $Cout_1$ is 25 ns

$(Cout_1 = Cin_2)$ to $Cout_2$ is 25 ns

⋮

$(Cout_6 = Cin_7)$ to $Cout_7$ is 25 ns.

so maximum delay for $Cout_7$ or last-stage Carryout is: (Cin-to-Cout) × Number of stages

$$25 \times 8 = 200 \text{ ns}$$

Maximum delay for last stage S would be equal to delay for $Cout_6 + S_7$:

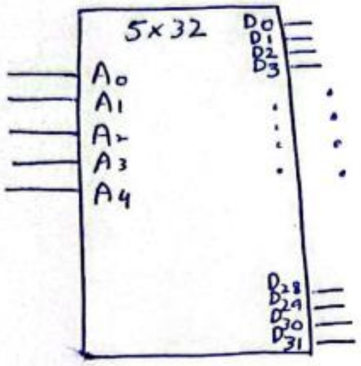
$$25 \times 7 + 30 = 205 \text{ ns}$$

so maximum total time would be 205 ns.

Q6 [CLO2: Decoders and encoders]

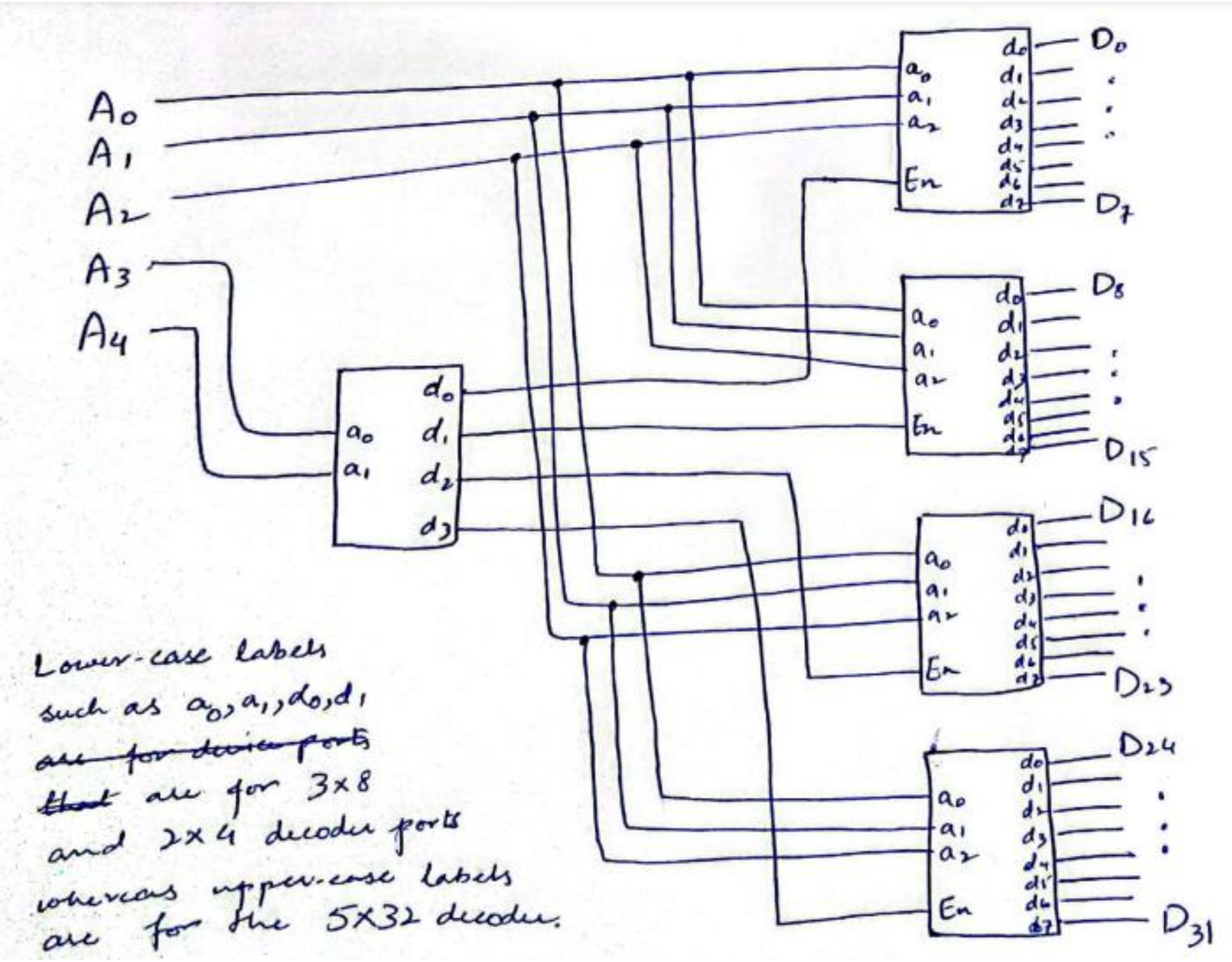
a. Construct a 5-to-32-line decoder with four 3-to-8-line decoders with enable and a 2-to4-line decoder. Use block diagrams for the components. [5]

5-to-32-line decoder using 4 3-to-8-line decoders and one 2-to-4-line decoder.



- The above is a 5×32 decoder.
- It has 5 inputs $A_0 - A_4$
- It has 32 outputs $D_0 - D_{31}$

The following configuration implements 5×32 using four 3×8 and ~~two~~ one 2×4 decoder.



b. A combinational circuit is specified by the following three Boolean functions:

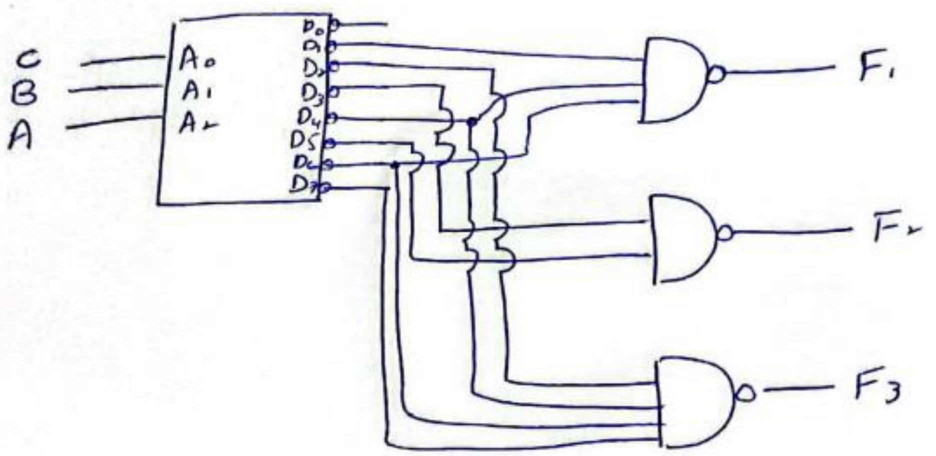
$F_1(A, B, C) = \sum(1, 4, 6)$

$F_2(A, B, C) = \sum(3, 5)$

$F_3(A, B, C) = \sum(2, 4, 6, 7)$

Implement the circuit with a decoder constructed with NAND gates and NAND or AND gates connected to the decoder outputs. Use a block diagram for the decoder. Minimize the number of inputs in the external gates. [5]

A decoder constructed with NAND gates has active-low outputs, i.e. only one decoder input is 0 and rest are 1 at any given time. So, we will use NAND instead of OR to implement functions.



c. Design a four-input priority encoder with inputs as in the following table, with input D0 having the highest priority and input D3 the lowest priority. [5]

Inputs				Outputs		
D_3	**D_2**	**D_1**	**D_0**	**x**	**y**	**V**
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

For x:

D_3D_2 \ D_1D_0	00	01	11	10
00	X	1	1	1
01	0	1	1	1
11	0	1	1	1
10	0	1	1	1

D_0 D_1

For y:

D_3D_2 \ D_1D_0	00	01	11	10
00	X	1	1	0
01	1	1	1	0
11	1	1	1	0
10	0	1	1	0

$D_1'D_2$ D_0

$$x = D_0 + D_1$$

$$y = D_0 + D_1'D_2$$

$$V = D_0 + D_1 + D_2 + D_3$$

Q7 [CLO2: Multiplexers]

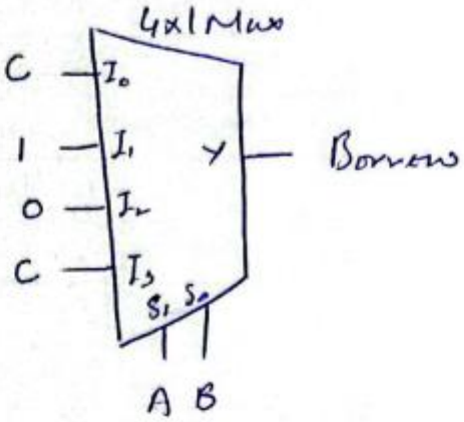
a. Implement a full-subtractor (*A*-*B*-*C*) circuit with two 4 × 1 multiplexers. [7]

- Full-subtractor truth table

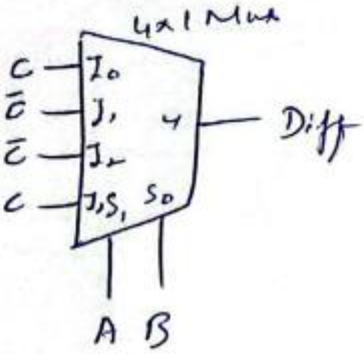
A	B	C	Borrow		Diff	
0	0	0	0	C	0	C
0	0	1	1		1	
0	1	0	1	1	1	C’
0	1	1	1		0	
1	0	0	0	0	1	C’
1	0	1	0		0	
1	1	0	0	C	0	C
1	1	1	1		1	

Q7a.

Borrow:



Difference:

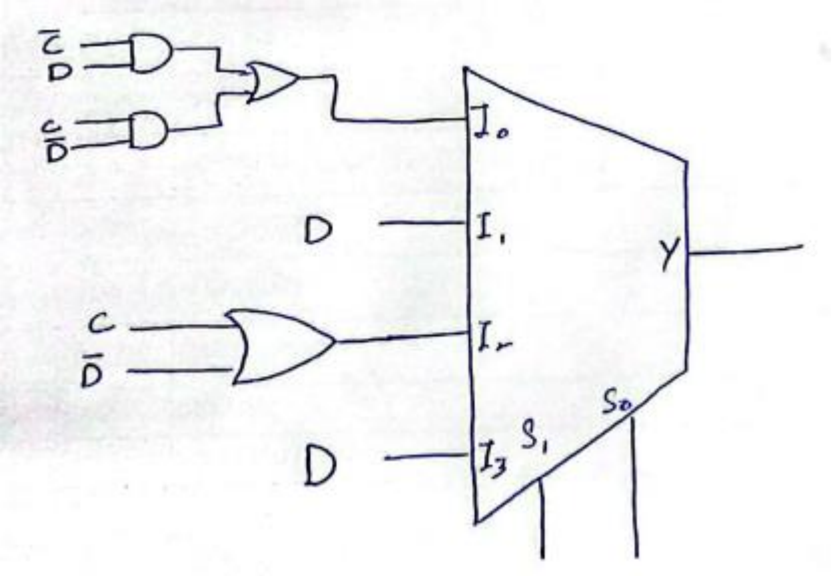


b. Implement the following Boolean function with a 4 × 1 multiplexer and external gates: [8]

$$F(A, B, C, D) = \sum(1, 2, 5, 7, 8, 10, 11, 13, 15)$$

Connect inputs A and B to the selection lines. The input requirements for the four data lines will be a function of variables C and D. These values are obtained by expressing F as a function of C and D for each of the four cases when AB = 00, 01, 10, and 11. These functions may have to be implemented with external gates.

A	B	C	D	F	F
0	0	0	0	0	C’D + CD’
0	0	0	1	1	
0	0	1	0	1	
0	0	1	1	0	
0	1	0	0	0	C’D + CD = D
0	1	0	1	1	
0	1	1	0	0	
0	1	1	1	1	
1	0	0	0	1	C’D’ + CD’ + CD = D’ + CD = C + D’
1	0	0	1	0	
1	0	1	0	1	
1	0	1	1	1	
1	1	0	0	0	C’D + CD = D
1	1	0	1	1	
1	1	1	0	0	
1	1	1	1	1	



~~c. Construct a 16 × 1 multiplexer with two 8 × 1 and one 2 × 1 multiplexers. Use block diagrams. [5]~~