MANIPAL UNIVERSITY

INSPIRED BY LIFE

DEPARTMENT OF INFORMATION & COMMUNICATION TECHNOLOGY

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL

Digital Systems Lab [ICT 2163]/ Digital System Design Lab [ICT 2165]

Second Year B. Tech. (IT/CCE) Degree – (2021-22)

INSTRUCTIONS TO THE STUDENTS

- **The students need to report to the labs/departments with clearance card issued by the institute/MAHE. They need to wear mask and gloves in the lab classes.**
- **Students should carry a book for noting observations and required stationeries to every lab session**
- **For every lab, exercise problems have to be solved depicting all design steps in the observation book before coming to the lab.**
- **Be well within the time and must sign in the log register provided**
- **Make sure to occupy the allotted seat and answer the attendance**
- **Show the program/circuit with results to the faculty in charge, on completion of experiments**
- **On receiving approval, copy the program/circuit with obtained results in the lab record**
- **Plagiarism (copying from others) is strictly prohibited and would invite severe penalty in evaluation**
- **Adhere to the rules, follow the institution dress code and maintain the decorum.**

STUDENTS SHOULD NOT

- **Bring mobile phones or any other electronic gadgets to the lab except calculators.**
- **Bring eatables to the lab.**
- **Go out of the lab without permission.**

<table>
<tr><th>Lab
No</th><th>Syllabus covered and list of experiments</th></tr>
<tr><td rowspan="2">Lab
1</td><td><b>Syllabus: K-Map</b></td></tr>
<tr><td>

- **F= XY + Z’ (Demonstration)**

1. **Simplify the Boolean function ‘F’ using K maps & implement using NAND gates only**
F = x’z’+y’z’+yz’+xyz.
2. **Design a combinational circuit that accepts a 3 bit number and generates an output binary number equal to the square of the input number. [Hint : Since the maximum 3 bit number is 7, and the square of it is 49, the number of output lines required is 6]. Realize using basic logic gates**

Truth Table:

<table>
<tr><th>X</th><th>Y</th><th>Z</th><th>P5</th><th>P4</th><th>P3</th><th>P2</th><th>P1</th><th>P0</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>0</td><td>0</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>0</td><td>1</td><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>0</td><td>1</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>1</td><td>0</td><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>1</td><td>0</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>1</td><td>1</td><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>1</td><td>1</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

Simplified SOP expressions:

Hardware Requirements:

</td></tr>
<tr><td rowspan="2">Lab
2</td><td><b>Syllabus : Code Converters</b></td></tr>
<tr><td>

Design following code converters

1. **BCD to excess-3 using NAND gates only**

Truth Table:

<table>
<tr><th>Decimal</th><th colspan="4">BCD</th><th colspan="4">Excess -3</th></tr>
<tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th><th>W</th><th>X</th><th>Y</th><th>Z</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td></td><td></td><td></td><td></td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td></td><td></td><td></td><td></td></tr>
<tr><td>2</td><td>0</td><td>0</td><td>1</td><td>0</td><td></td><td></td><td></td><td></td></tr>
<tr><td>3</td><td>0</td><td>0</td><td>1</td><td>1</td><td></td><td></td><td></td><td></td></tr>
<tr><td>4</td><td>0</td><td>1</td><td>0</td><td>0</td><td></td><td></td><td></td><td></td></tr>
<tr><td>5</td><td>0</td><td>1</td><td>0</td><td>1</td><td></td><td></td><td></td><td></td></tr>
<tr><td>6</td><td>0</td><td>1</td><td>1</td><td>0</td><td></td><td></td><td></td><td></td></tr>
<tr><td>7</td><td>0</td><td>1</td><td>1</td><td>1</td><td></td><td></td><td></td><td></td></tr>
</table>

</td></tr>
</table>

8	**1**	**0**	**0**	**0**				
9	**1**	**0**	**0**	**1**				

Hardware Requirements:

2. **Self-complementary 8 4 -2 -1 to self-complementary 2 4 2 1 using NOR gates only**

Truth Table:

Decimal	**8**	**4**	**-2**	**-1**	**2**	**4**	**2**	**1**
	A	**B**	**C**	**D**	**W**	**X**	**Y**	**Z**
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**				

Hardware Requirements:

Lab 3

Syllabus :Adders , Multipliers

1. **Design a full subtractor .**
2. **Design a 4-bit binary adder/subtractor using 7483 IC and external gates.**
3. **Design 2 bit X 3 bit binary multiplier using 7483 ICs and external gates.**

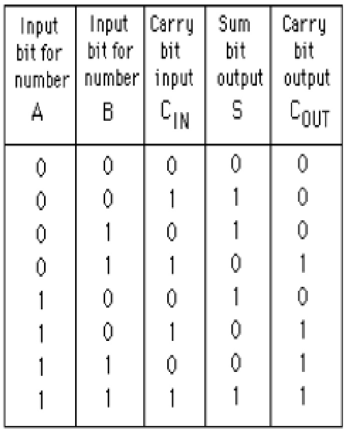
Lab 4

Syllabus Magnitude Comparators and Decoder

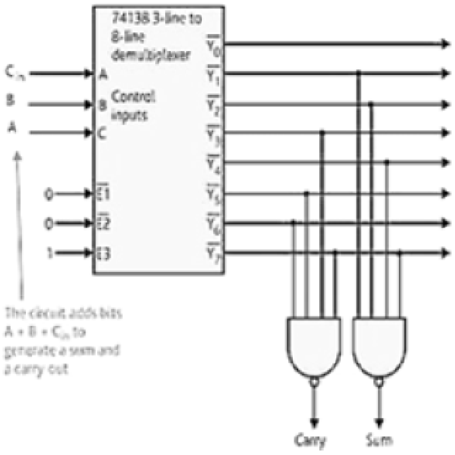
1. **Design a 1 bit magnitude comparator with cascading inputs using basic gates.**

2. **Design 5-bit magnitude comparator using 7485 ICs**

3. **Implement one-bit full adder using 74138 IC and external NAND gates only.**
   The truth table of 1-bit full adder is given below. The circuit diagram using the 74138 and NAND gates is shown below.



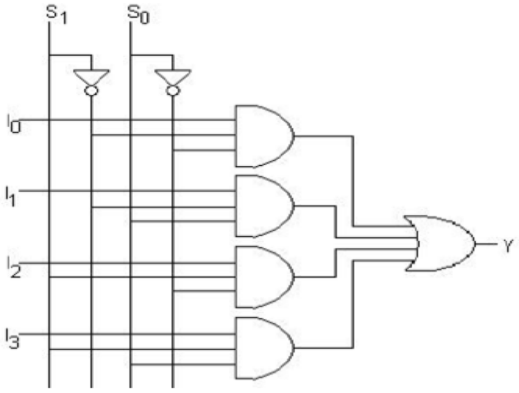
Input bit for number A	Input bit for number B	Carry bit input C_{IN}	Sum bit output S	Carry bit output C_{OUT}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

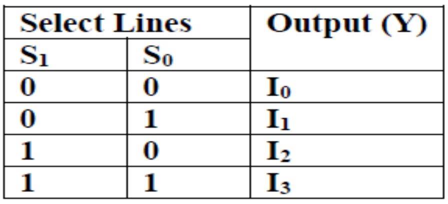

74138 3-line to 8-line demultiplexer
Cin
B
A
Control inputs
Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7
0
0
1
E1
E2
E3
The circuit adds bits A + B + Cin to generate a sum and a carry out
Carry
Sum

Lab 5

Syllabus MUX, Flip Flops

1. **Design a 4:1 MUX using basic logic gates.**





Select Lines		Output (Y)
S_1	S_0	
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

2. **Design a combinational circuit using 74151 IC and external gates to implement the following functions.**
   F_1(w,x,y,z) = wx'y'+w'xy'z F_2(w,x,y,z) = xz'+wx'y

3. **Convert a given D flip flop to work as a JK flip flop.**

Lab 6

Syllabus : Asynchronous and Synchronous counters

1. **Design a counter that performs UP count from 00 to 81 using 7490 IC and external gates.**

2. **Design an asynchronous counter that counts from 00h to 24H using 7493 IC and external gates.**

3. **Design a 2-bit binary UP/DOWN counter such that when control input Y=0, it performs BINARY UP count and when Y=1,**

Lab 7

Syllabus : Sequence Generators and Shift Registers

1. **Design a sequential logic circuit to generate a binary sequence 110010 using Johnson counter.**

2. **Design a 4 - bit universal shift register using D flip flops and 74153 ICs**
   Function Table of Universal Shift Register:

Select Lines		**Function performed**
S_1	**S_0**	**F**
0	**0**	**No change**
0	**1**	**Left Shift**
1	**0**	**Right Shift**
1	**1**	**Parallel Load**

   Hardware Requirements:

3. **Design a 4-bit ring counter using JK flip flops**
   Hardware Requirements:

Q_D	**Q_C**	**Q_B**	**Q_A**

Lab 8

Syllabus : Sequence Detectors

1. **Design a sequential model that detects the sequence 1011 using T flip flops and external gates. Assume that overlapping sequence is to be detected.**

State Table with T flip flop excitation

Present State			Input	Next State			D flip flop input			Output
				X = 0			X = 0			
Q_C	**Q_B**	**Q_A**	**X**	**Q_C^+**	**Q_B^+**	**Q_A^+**	**T_C**	**T_B**	**T_A**	**Y**
0	0	0	0							
0	0	0	1							
0	0	1	0							
0	0	1	1							
0	1	0	0							
0	1	0	1							

0	1	1	0							
0	1	1	1							
1	0	0	0							
1	0	0	1							
1	0	1	0							
1	0	1	1							
1	1	0	0							
1	1	0	1							
1	1	1	0							
1	1	1	1							

2. Design a sequential model that detects the sequence 101 using JK Flip Flop. Assume non overlapping sequence is to be detected.