# Assignment 3 for COMP 2280

## Due: 11:30 PM on Wednesday March 23th, 2022

**Total Marks: 40**

**Notes**:

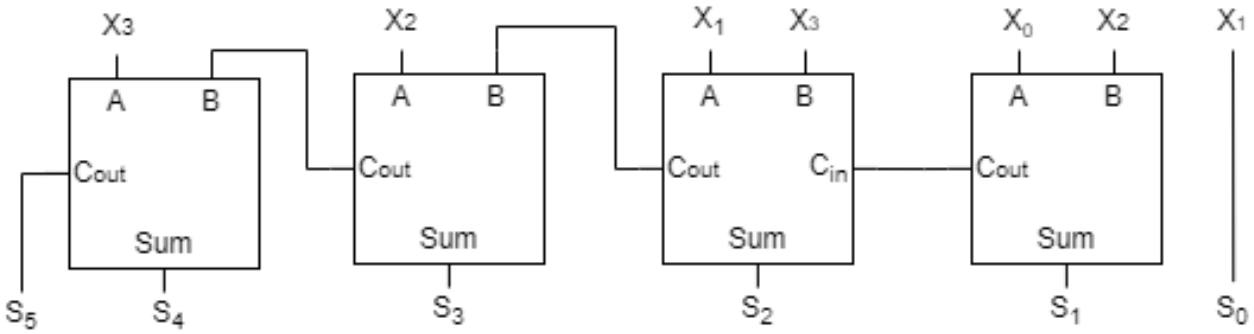
1. For the written questions show all of your work and submit either a PDF or text file.
2. For programming questions hand in your source program (.asm file).
3. Name your programs LastnameFirstnameAxQy.asm, replace x with the assignment number and y with the question number.
4. Please comment appropriately for programming question. Please read the "programming standards" for COMP 2280. Marks are allocated for good documentation.
5. Hand in your assignment through the UMLearn

## Written Part

## Question 1, Adders [4 marks]

Design a circuit that adds together 2*X and (X/2) to give 5*(X/2), where X is a 4-bit binary number composed of $X_3X_2X_1X_0$, and gives a 6-bit sum composed of $S_5S_4S_3S_2S_1S_0$.

- This can be done using 3 half adders and 1 full adder, no additional gates are required.
- First, find the relationship between X ($X_3X_2X_1X_0$) and 2*X, and X/2 in terms of ($X_3X_2X_1X_0$), there are no decimal values.



# Assignment 3 for COMP 2280

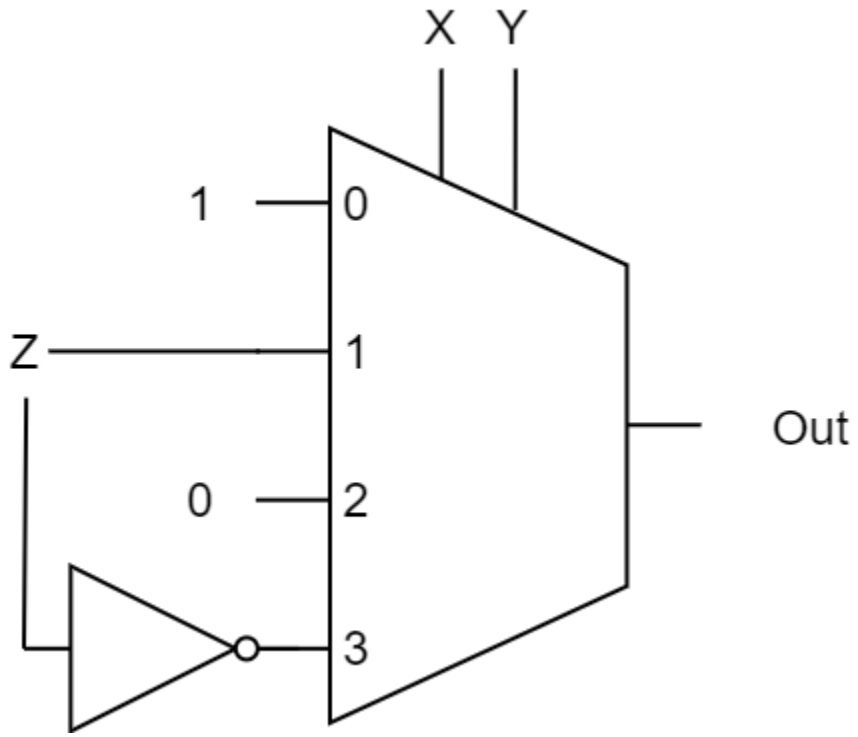
## Due: 11:30 PM on Wednesday March 23th, 2022

**Total Marks: 40**

(a) [4 marks] Using only a 4-to-1 multiplexor and one inverter (NOT gate) design a circuit that implements the truth table given above.

- Do not draw the individual gates of the Multiplexor, use the logic symbol from the course notes.
- Use X and Y as the select lines to choose which of the 4 inputs is connected to OUT.
- OUT can be represented as a function of Z or as constant 0 or as constant 1 for each pair of values of X and Y. Aka, what is connected to the 4 inputs selected by X and Y?

| X | Y | Z | OUT | OUT==?? |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | CONST-1 |
| 0 | 0 | 1 | 1 | CONST-1 |
| 0 | 1 | 0 | 0 | Z |
| 0 | 1 | 1 | 1 | Z |
| 1 | 0 | 0 | 0 | CONST-0 |
| 1 | 0 | 1 | 0 | CONST-0 |
| 1 | 1 | 0 | 1 | NOT(Z) |
| 1 | 1 | 1 | 0 | NOT(Z) |




(b) [4 marks] Using only a 3-to-8 decoder and an OR gate with any many inputs as

# Assignment 3 for COMP 2280

## Due: 11:30 PM on Wednesday March 23th, 2022

**Total Marks: 40**

necessary design, a circuit that implements the truth table given above. Do not draw the individual gates of the decoder, use the logic symbol from the course notes.

**Hint: Use X, Y and Z as the select lines.**
**Hint: Imagine having 8 light-switches, which of them are turning on the lightbulb**

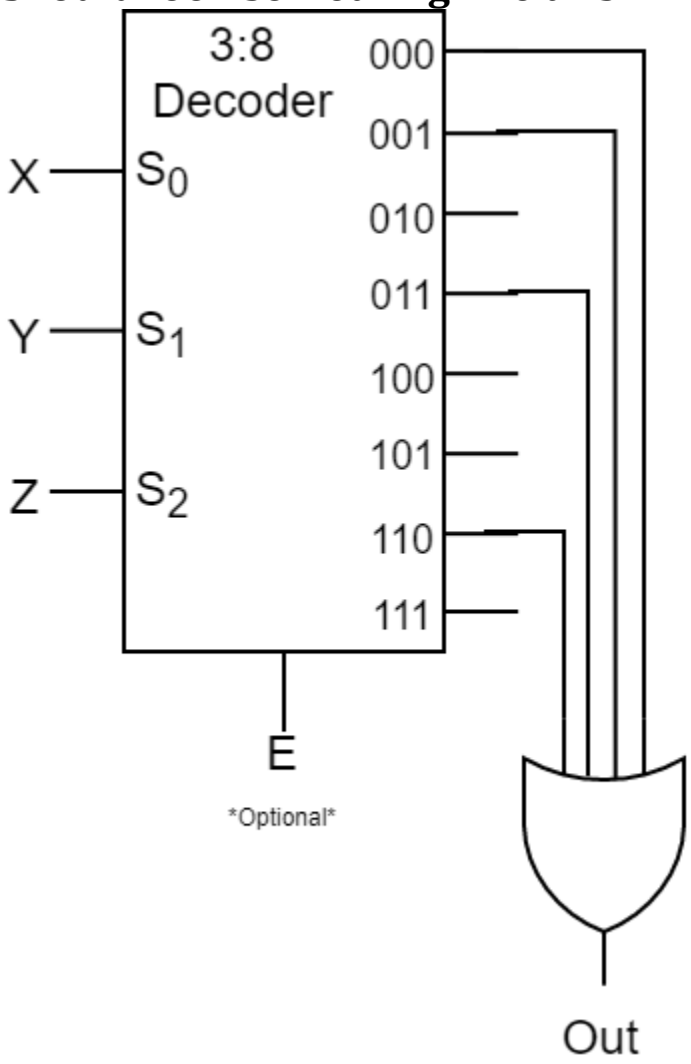
| X | Y | Z | OUT |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

# Assignment 3 for COMP 2280

**Due: 11:30 PM on Wednesday March 23th, 2022**

**Total Marks: 40**

## Should look something like this:



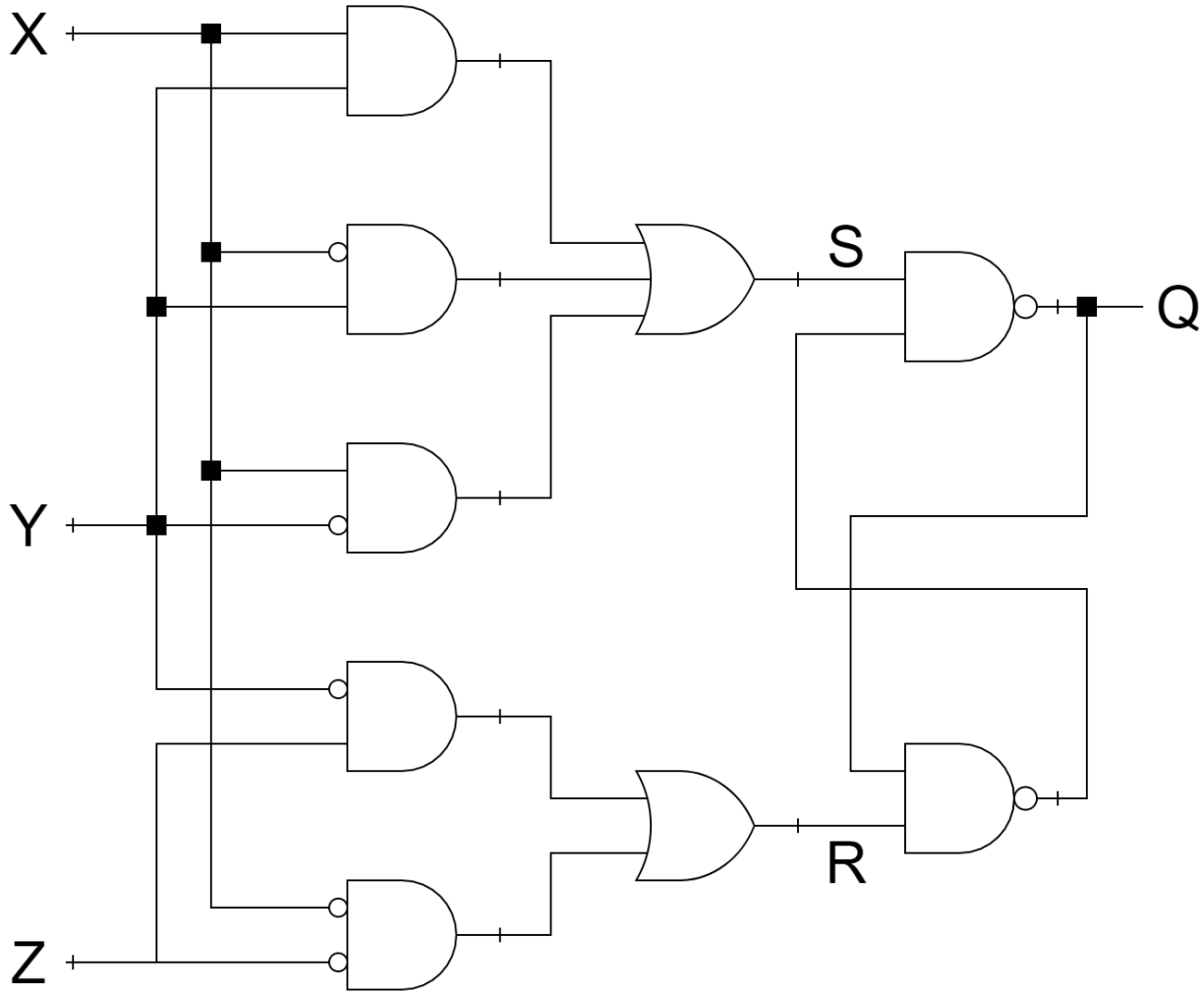
# Assignment 3 for COMP 2280

## Due: 11:30 PM on Wednesday March 23th, 2022

**Total Marks: 40**

## Question 3, Sequential Circuit [8 marks]

Give the truth table for the following circuit.




(A) Write out a truth table for S, R and Q where 'S' and 'R' are functions of X, Y, and Z and 'Q' is a function of 'S' and 'R'.

a. The headings for the columns in the truth table should be X, Y, Z, S, R and Q.
b. X, Y and Z are inputs, and S, R and Q are outputs.

| X | Y | Z | S | R | Q |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | Q |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | Q |
| 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 |

# Assignment 3 for COMP 2280

## Due: 11:30 PM on Wednesday March 23th, 2022

**Total Marks: 40**

(B) Write the expressions for S and R in terms of X, Y and Z. You can do this either from the **truth table** or the **circuit diagram**. In either case write the simplest sum-of-products expressions possible.

$$S = XY+X'Y+XY'$$

$$R = Y'Z + X'Z'$$

(C) State the value of Q for each combination of X, Y, Z. In each case give the value of Q after there are no more changes due to gate delays. These steps happen in sequential order, one after the other.

I. Set X = 0, Y = 0, Z = 0. After all the changes due to gates delays what is Q?
II. Change Z to 1. After all the changes due to gate delays what is the value of Q?
III. Change X to 1. After all the changes due to gate delays what is the value of Q?
IV. Change Y to 1. After all the changes due to gate delays what is the value of Q?
V. Change Y to 0. After all the changes due to gate delays what is the value of Q?
VI. Change X to 0. After all the changes due to gate delays what is the value of Q?
VII. Change Y to 1. After all the changes due to gate delays what is the value of Q?
VIII. Change Z to 0. After all the changes due to gate delays what is the value of Q?

| | X | Y | Z | S | R | Q |
|---|---|---|---|---|---|---|
| I | 0 | 0 | 0 | 0 | 1 | 1 |
| II | 0 | 0 | 1 | 0 | 1 | 1 |
| III | 1 | 0 | 1 | 1 | 1 | 1 |
| IV | 1 | 1 | 1 | 1 | 0 | 0 |
| V | 1 | 0 | 1 | 1 | 1 | 0 |
| VI | 0 | 0 | 1 | 0 | 1 | 1 |
| VII | 0 | 1 | 1 | 1 | 0 | 0 |
| VIII | 0 | 1 | 0 | 1 | 1 | 0 |

# Assignment 3 for COMP 2280

## Due: 11:30 PM on Wednesday March 23th, 2022

**Total Marks: 40**

## Question 4, Programming [20 marks]

Write a subroutine that performs a linear search on an unsorted array of 16-bit 2's compliment integers.

- The subroutine has three parameters,
  - The address of the array
  - 'N' which is the number of elements in the array
  - A target value to search for in the array.
- The parameters must be on the stack.
- If the target value is found in the array return the position at which it is found, otherwise return -1.
- The first element is at position 0 and the last element is at position n-1.
- The result must be returned to the caller via the stack.
- Use R5 as the frame pointer and R6 as the stack pointer.
- Save and restore all registers used in the subroutine using the stack.
- Include a register dictionary and a map of the activation record for the subroutine.

- The mainline must include the following declarations:

```
ASCII           .fill       #48
FOUNDMSG        .stringz    "\nFound at position: "
NOTFOUNDMSG     .stringz    "\nNot found."
EOPMSG          .stringz    "\nEnd of Processing"
N               .fill       #10
SOURCE          .fill       #99
                .fill       #67
                .fill       #-33
                .fill       #0
                .fill       #-123
                .fill       #29
                .fill       #17
                .fill       #79
                .fill       #22
                .fill       #-1
NUNMTARGETS     .fill       #4
TARGETS         .fill       #-123
                .fill       #79
                .fill       #66
                .fill       #67
```

- You may add any other declarations you require.
- As in the subroutine R6 must be used as the stack pointer.
- The array TARGETS contains target values to be searched for in the array SOURCE.
- 'N' is the number of elements in the array SOURCE
- NUMTARGETS is the number of elements in the array TARGETS.

# Assignment 3 for COMP 2280

## Due: 11:30 PM on Wednesday March 23th, 2022

**Total Marks: 40**

For each element in the array TARGETS call the linear search subroutine to determine if the element / target is found in the SOURCE array and display the appropriate message as shown in the sample output given below.

- Before calling the subroutine push the arguments onto the stack in the following order:
  - Address of the array SOURCE
  - The value of N
  - The current target value.
- Reserve space on the stack for the value returned by the subroutine.

The target 79 is found at position 7 within the SOURCE array so “"\nFound at position: “ and “7” are displayed, but the target 66 is not found in the SOURCE array so "Not found." is displayed.

Using the declarations given above the output must be:

Found at position: 4
Found at position: 7
Not found.
Found at position: 1

See .ASM File