CS 251, Spring 2023, Assignment 2

Due Thursday, June 15, 10:00 PM
No Late submission accepted.

You are required to read, complete and sign, and submit (as well as follow) the following statement of Academic Integrity.

Statement of Academic Integrity for CS 251 Spring 2023, **Assignment 2**
I declare the following statements to be true:

- I have not used any unauthorized aids.
- I recognize that while I can discuss the questions in this assignment on Piazza and other forums with the instructors and with other students in the class, the write up that I am submitting is my own.
- I am aware that misconduct related to course work can result in significant penalties, including failing the course and suspension (this is covered in Policy 71:)

  `https://uwaterloo.ca/secretariat/policies-procedures-guidelines/policy-71`

Student Name: Gesge Wang

UW ID#: 20943813

Signature: [signature]

Date: Jun. 14, 2023

The purpose of this assignment is to practice designing and implementing synchronous sequential circuits.

Coverage material for this assignment can be found on the course website on the Lecture Notes webpage in the

- June 1st - June 13th Lectures on Data Representation and ALU operations

For this assignment, make note of the following details:

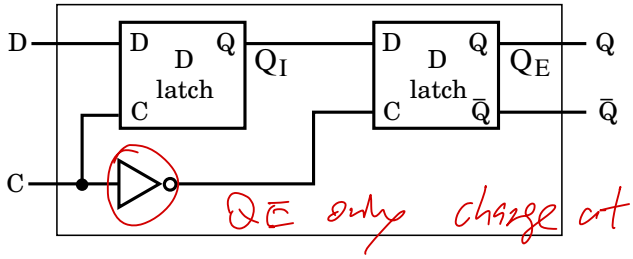
- Any diagrams that are part of your solution must be drawn neatly, using rectilinear lines and clearly labelled inputs and outputs.

We have an A2 Official Post on piazza. In this post, we will include all A2 assignment related information such as

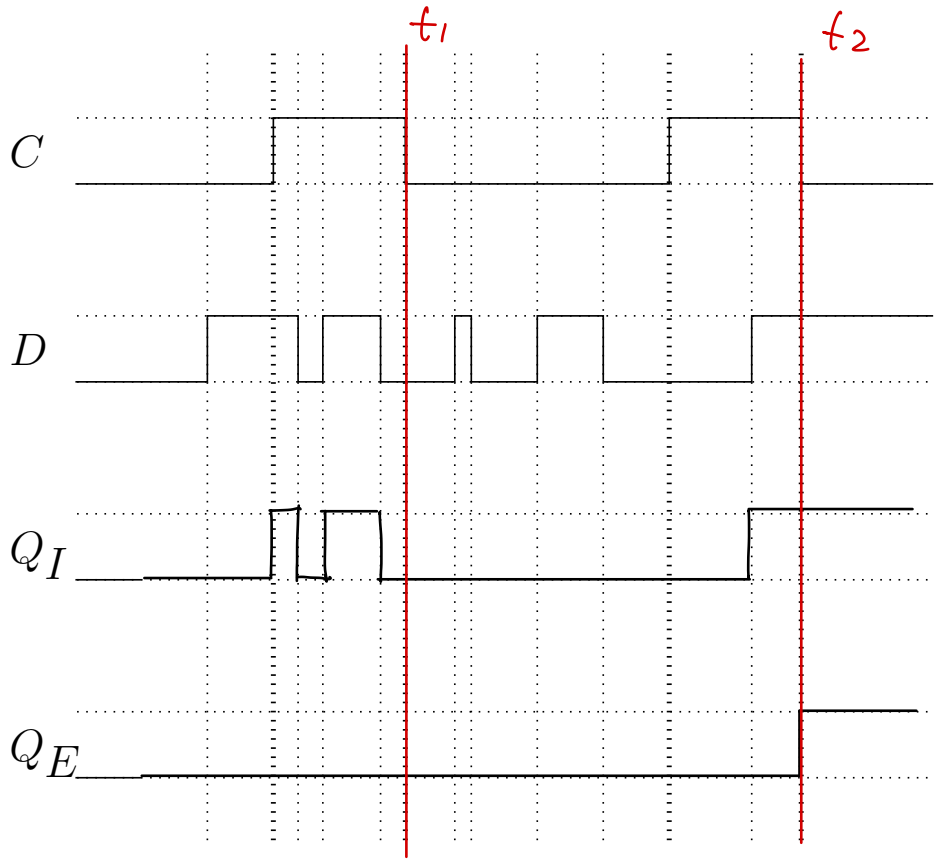
- updates/corrections made to the assignment, if any
- remark request due date
- FAQs

1. (10 points) D Flip-flop.

   Consider the D flip flop below:



   (a) (6 points) In the figure below are traces of input signals D and C. Draw the resulting traces of Q_I and Q_E.



   (b) (2 points) In your solution above, mark the falling-edges of the the clock signal (C) with labels t_1 and t_2, for the first and second falling edge, respectively.

(c) (1 point) What is the relationship of input D to Q_E at the falling edges of C?

Q_E only update to D at falling edge of C.

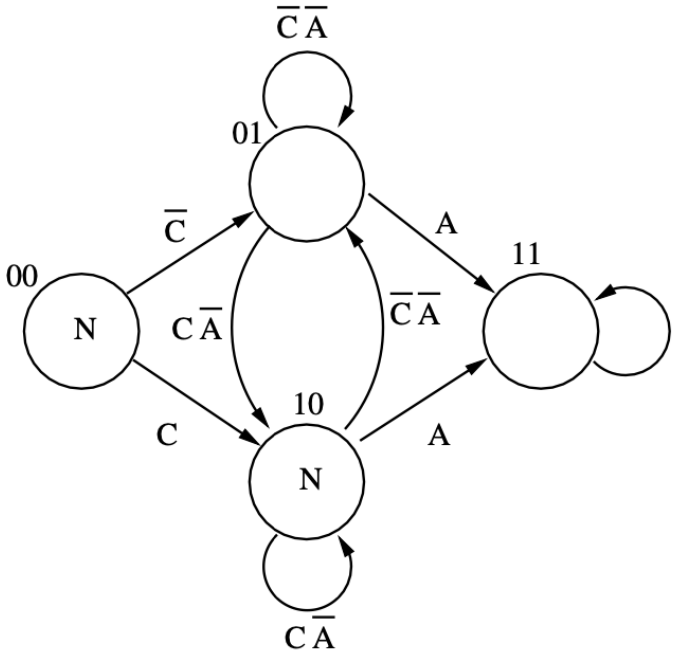
(d) (1 point) What is the relationship of D and Q_E in one clock cycle, that is, within the period t_1 to t_2?

Q_E remains unchanged between t_1, t_2.

Changes of D during t_1, t_2 period does not affect Q_E

2. (20 points) Finite State Machine.

   Consider the following finite state machine:



   (a) (5 points) Complete the truth tables for the next-state and output functions below.

Next State Table

State		Inputs		Next State	
S_1	S_0	A	C	S_1'	S_0'
0	0	0	0	0	1
0	0	0	1	1	0
0	0	1	0	0	1
0	0	1	1	1	0
0	1	0	0	0	1
0	1	0	1	1	0
0	1	1	0	1	1
0	1	1	1	1	1
1	0	0	0	0	1
1	0	0	1	1	0
1	0	1	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1

Output Table

S_1	S_0	N
0	0	1
0	1	0
1	0	1
1	1	0

(b) (5 points) Compress the truth table for next state function from part (a) using don't cares (X) where applicable.

State		Input		Next State	
S_1	S_0	A	C	S_1'	S_0'
X	0	0	0	0	1
0	X	0	0	0	1
0	0	X	0	0	1
X	0	0	1	1	0
0	X	0	1	1	0
0	0	X	1	1	0
1	1	X	X	1	1
0	1	1	X	1	1
1	0	1	X	1	1

(c) (5 points) Give the Boolean formulas for each next state variable in **unreduced** form.

$S_1' = (S_1 + S_0 + A + C)(S_1 + S_0 + \overline{A} + C)(S_1 + \overline{S_0} + A + C)(\overline{S_1} + S_0 + A + C)$

$S_0' = (S_1 + S_0 + A + \overline{C})(S_1 + S_0 + \overline{A} + \overline{C})(S_1 + \overline{S_0} + A + \overline{C})(\overline{S_1} + S_0 + A + \overline{C})$

(d) (5 points) Complete the table below tracing the above finite state machine on a particular sequence of input bits. Note: that the Next State of one column is the State for the next column. We have done the first step for you as an example.

State	00	**01**	01	10	10	01	01
Input C	0	0	1	1	0	0	0
Input A	0	0	0	0	0	0	0
Output N	**1**	0	0	1	1	0	0
Next State	**01**	01	10	10	01	01	01

3. (10 points) Finite State Machines.

Consider the following truth tables for the next-state and output functions:

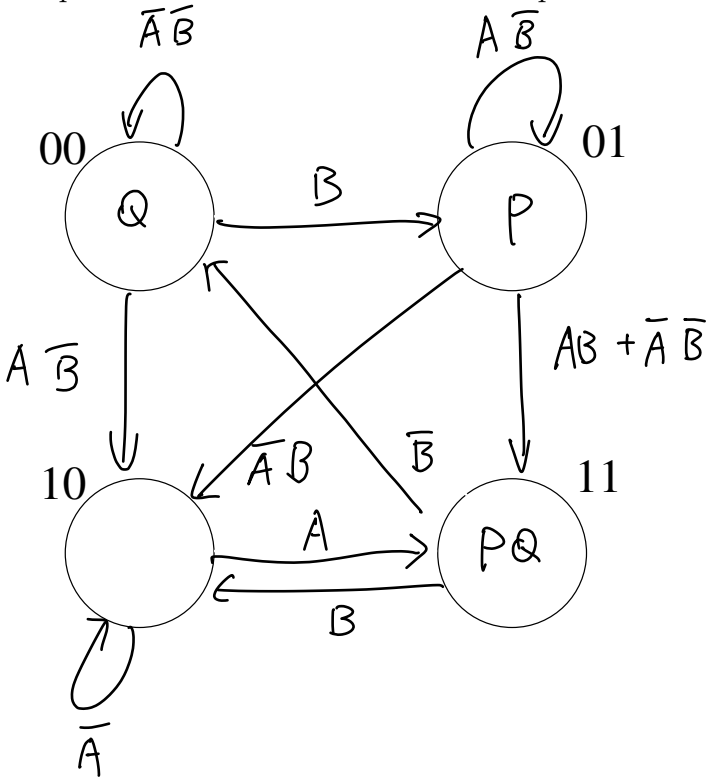
Truth table for next state function

S_1	S_0	A	B	S_1'	S_0'
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	0	1
0	1	0	0	1	1
0	1	0	1	1	0
0	1	1	0	0	1
0	1	1	1	1	1
1	0	0	X	1	0
1	0	1	X	1	1
1	1	X	0	0	0
1	1	X	1	1	0

Truth table for output function

S_1	S_0	P	Q
0	0	0	1
0	1	1	0
1	0	0	0
1	1	1	1

Complete the finite state machine shown below using the next state and output functions above. Label the transitions with Boolean expressions of inputs A and B. Use **simplified** Boolean expressions for the transitions where possible.



4. (6 points) Data Representation

(a) (2 points) Write the number 25 as a signed 8 digit binary number. Show your work.

$25 \div 2 = 12 \cdots 1$

$12 \div 2 = 6 \cdots 0$

$6 \div 2 = 3 \cdots 0$

$3 \div 2 = 1 \cdots 1$

$1 \div 2 = 0 \cdots 1$

00011001

(b) (2 points) Compute the 2's complement negation of 25. Note: you should **not** use software to compute the 2's complement. Instead, show your work full points.

flip all bits of $(25)_{10}$ then add 1 to get $(-25)_{10}$

00011001 flip bits get 11100110 add 1 get 11100111

(c) (2 points) Multiply the following unsigned binary numbers. Show your work. You may omit the "zero lines" if you wish

$$\begin{array}{r} 0111\,1011 \\ \times 0000\,0111 \\ \hline 01111011 \\ 01111011 \\ +\ 01111011 \\ \hline 10100111 \end{array}$$

5. (13 points) Binary Arithmetic.

   (a) (10 points) Add the following pairs of 8-bit two's complement binary numbers, giving an 8-bit result (i.e., throw away the carry-out).

   Then, give the signed decimal value of the 8-bit result and answer the questions pertaining to each addition.

	0001 0011 + 0010 1111
2's Complement Binary:	0100 0010
Signed Decimal:	66
Is there a carryout?	no
Is there an overflow?	no
Is the sum correct?	yes

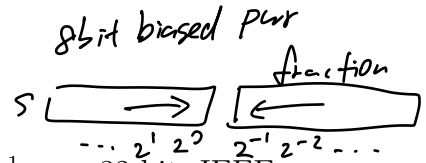
	0101 0101 + 0110 1000
2's Complement Binary:	1011 1101
Signed Decimal:	-67
Is there a carryout?	No
Is there an overflow?	Yes
Is the sum correct?	NO

	0100 0001 + 0111 1001
2's Complement Binary:	1011 1010
Signed Decimal:	-70
Is there a carryout?	NO
Is there an overflow?	Yes
Is the sum correct?	NO

	0011 0001 + 0101 1011
2's Complement Binary:	1000 1100
Signed Decimal:	-116
Is there a carryout?	NO
Is there an overflow?	Yes
Is the sum correct?	NO

   (b) (3 points) Compute the following bit-wise logic operations.

Operand 1	Operation	Operand 2	Result
0001 1111	AND	0011 0101	0001 0101
0001 1111	OR	0011 0101	0011 1111
0001 1111	XOR	0011 0101	0010 1010



6. (5 points) Floating Point Numbers.

   (a) (3 points) Write the base 10 number $-1.6015625 \times 10^{-1}$ as a 32-bit, IEEE normalized floating point number with biased exponent.

   You should follow the algorithm discussed in class to convert the fractional decimal portion of the number to binary.

   First you must represent the decimal number **not** in scientific notation and then begin the conversion to binary.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
1	0	1	1	1	1	1	0	0	0	1	0	0	1	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

$1.6015625 \times 10^{-1} = 0.16015625 \times 2 = 0.3203125$ 0

$0.3203125 \times 2 = 0.640625$ 0

$0.640625 \times 2 = 1.28125$ 1

$0.28125 \times 2 = 0.5625$ 0

$0.5625 \times 2 = 1.125$ 1

$0.125 \times 2 = 0.25$ 0

$0.25 \times 2 = 0.5$ 0

$0.5 \times 2 = 1.0$ 1

$-3+127=124$

$124 \div 2 = 62 \cdots 0$

$62 \div 2 = 31 \cdots 0$

$31 \div 2 = 15 \cdots 1$

$15 \div 2 = 7 \cdots 1$

$7 \div 2 = 3 \cdots 1$

$3 \div 2 = 1 \cdots 1$

$1 \div 2 = 0 \cdots 1$

$124_{10} = 1111100_2$

$= 0111\ 1100$

So $(1.6015625 \times 10^{-1})_{10} = (0.00101001)_2 = (1.01001 \times 2^{-3})_2$

   (b) (2 points) Complete the following table, showing the sign & magnitude and the corresponding biased representations with a bias of 127.

Sign & Magnitude	Biased
60	187
-13	114
-17	110
74	201

7. (6 points) ALU Operations.

Consider the diagram below for a 64-bit ALU as developed in Appendix A.5 of the course text. Modify this diagram to have the following three additional outputs (**ON**, **GT12** and **IOSOE**)

In the figure, label each output with the name given in bold in the question. For the inputs to your circuits, you may use R0, etc., instead of Result0, etc. Assume the inputs and the result are all 2's complement numbers. Do not use multiplexors in your solutions.

Below is the diagram for a ~~32~~ 64-bit ALU as developed in Appendix B.5 of the course text. Modify this diagram to have the following three additional outputs. In the figure, label each output with the name given in bold in the question. For the inputs to your circuits, you may use R_0, etc., instead of Result0, etc.

$R_0=1 \quad R_{63}=1$

(a) (2 pts) **ON**: This should be HIGH (1) when the output is an odd, negative number.

(b) (2 pts) **GT12**: This should be HIGH (1) when the signed output is a greater than 12.

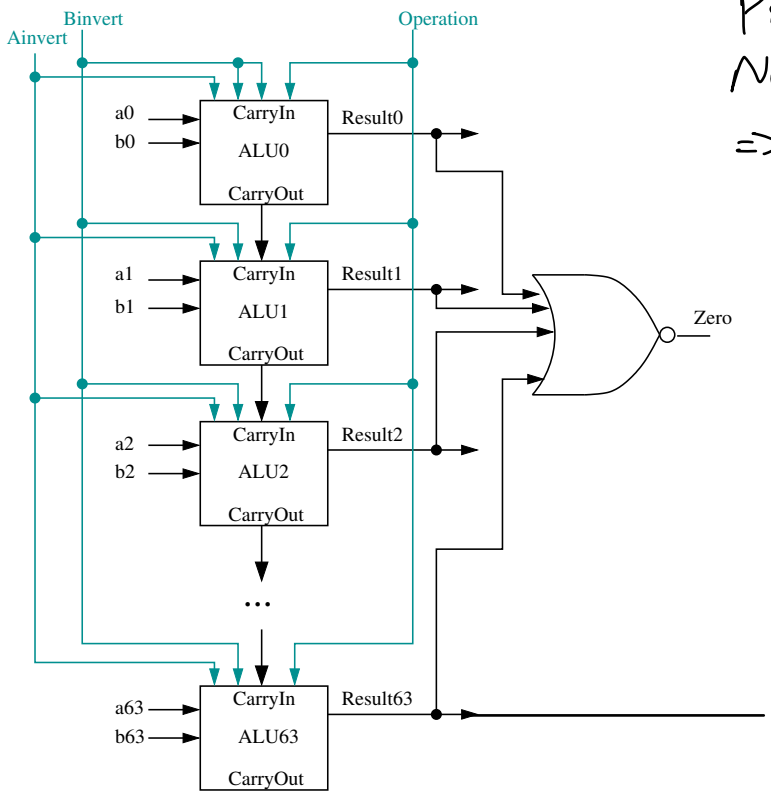
$A_{63}=0$ & $((A_2=A_3=1, A_0+A_1=1)$ OR $A_4+\cdots+A_{62}=1)$

(c) (2 pts) **IOSOE**: This should be HIGH (1) when the **I**nputs are of **O**pposite **S**ign and the **O**utput is **E**ven.

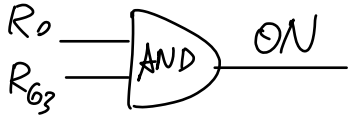
Pos even: $R0=0, R_{63}=0$

Neg even: $R_{63}=1, R0=0$

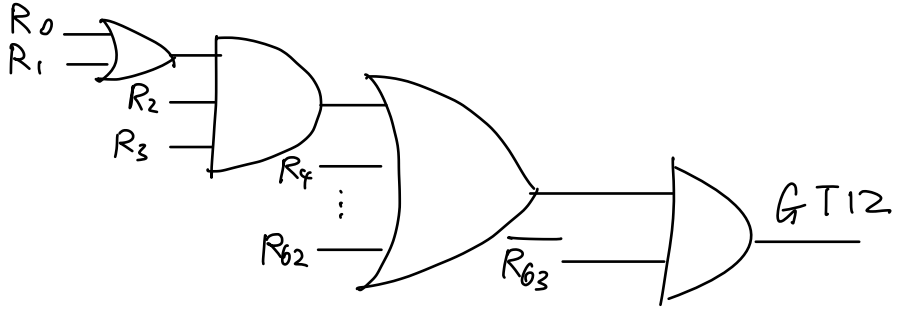
$\Rightarrow$ even: $R_0=0$



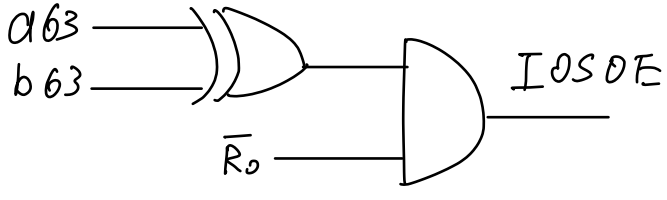
a)


R_0
R_{63}
AND
ON

b)


R_0
R_1
R_2
R_3
R_4
R_{62}
$\overline{R_{63}}$
GT12

c)


a63
b63
$\overline{R_0}$
IOSOE

8. (12 points) ALU Operations.

Consider the 1-bit ALU studied in class repeated four times below to work with two 4-bit 2's complement numbers.

If the inputs are $\mathbf{a} = 6_{10} = 0110_2$ and $\mathbf{b} = 4_{10} = 0100_2$, use the four ALUs to compute $a - b$ or $(6 - 4)$. $a - b = a + (-b) = a + (\overline{b} + 1)$

Label the inputs and outputs marked with bold lines for a_i, b_i, R_i, `Binvert`, `CarryIn`, and `Operation` to compute this subtraction.