CSC 258H1 S 2016 Midterm Test
Duration — 1 hour and 50 minutes
Aids allowed: none

Student Number: ⌊ ⌋

UTORid: ⌊ ⌋

Last Name: _____ **First Name:** _____

Question 0. [1 MARK]

Circle the **lecture section** you are enrolled in.

Instructor	Lecture Section
Xander Chin	L0101 (MWF 11-12)
	L0201 (MWF 12-1)
Myrto Papadopoulou	L5101 (T 6-9)
	L5201 (M 6-9)

*Do **not** turn this page until you have received the signal to start.*
(Please fill out the identification section above, **write your name on the back of the test**, and read the instructions below.)
Good Luck!

This midterm is double-sided, and consists of 9 questions on 15 pages (including this one). When you receive the signal to start, please make sure that you have all pages.

- If you use any space for rough work, indicate clearly what you want marked.
- Do not remove any pages from the exam booklet.
- You may use a pencil; however, work written in pencil will not be considered for remarking.

0: _____/ 1
1: _____/14
2: _____/ 9
3: _____/ 9
4: _____/14
5: _____/15
6: _____/ 8
7: _____/10
8: _____/10

TOTAL: _____/90

Total Pages = 15

Question 1. [14 MARKS]

Answer the following questions in the space provided. When providing a written answer, please write **as clearly and legibly as possible**. Marks will not be awarded to unreadable answers.

(a) Convert the following two numbers to hexadecimal. You need to use the smallest number of hexadecimal digits possible.

- Decimal number 16: **10**
- Binary number 10100111: **a7**

(b) Write the signed representations of the following decimal numbers (in 2's complement format). You need to use the smallest number of binary bits possible.

- Decimal number -8: **1000**
- Decimal number 19: **010011**

(c) Write down the ModelSim command(s) needed to make 1-bit signal w have the following behaviour. It should start with a logic value of 0 and change to logic-1 8ns later.

force {w} 0 0ns, 1 8ns

or alternatively

force {w} 0
run 8ns
force {w} 1

(d) Circle the correct statement(s) below. Marks will be deducted for incorrect answers, so do not guess!

(i) Minterm $m0$ is the complement of maxterm $M0$ **CORRECT**

(ii) If F is $m0 + m1$ for a 2-input circuit, then F' is $M1' + M3'$

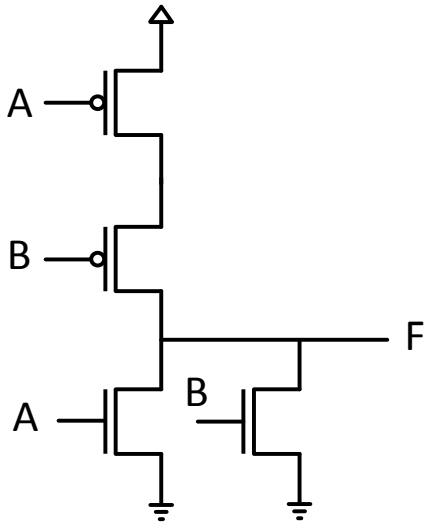
(iii) If F is $m2 + m0$ for a 2-input circuit, then F' is $M1' + M3'$ **CORRECT**

(iv) Minterm $m7$ does not exist for a 3-input circuit.

(e) Complete the following sentences:

- If we dope silicon with **Phosphorus** atoms we get an n-type material.
- When bringing an n-type material in contact with a p-type material a **depletion** layer is formed.
- In a PN-junction, the drift current is caused by **electric field**.
- If a PN-junction is reverse biased, the positive terminal of the battery is connected to the **n**-type material and the negative terminal is connected to the **p**-type material.

(f) What circuit is this?



NOR gate

Question 2. [9 MARKS]

For each provided Verilog code snippet identify the schematic(s) with **equivalent behaviour** by writing the corresponding schematic number(s) in the blanks below.

There might be one or more schematics that are correct for a given Verilog snippet. You need to specify all of them for full marks. Marks will be deducted for every incorrect answer (when multiple are given), so do NOT guess!

(i) **4**

```
wire A, B, C, F;

assign F = (A & B) | B | C;
```

(ii) **2, 6, 8**

```
wire A, B, C;
reg F;

always @ (*) begin
    if (A == 1'b0)
        F = B;
    else
        F = C;
end
```

(iii) **3**

```
wire A, B, C, F;

assign F = & {B, C, A | 1'b1};
```

(iv) **10**

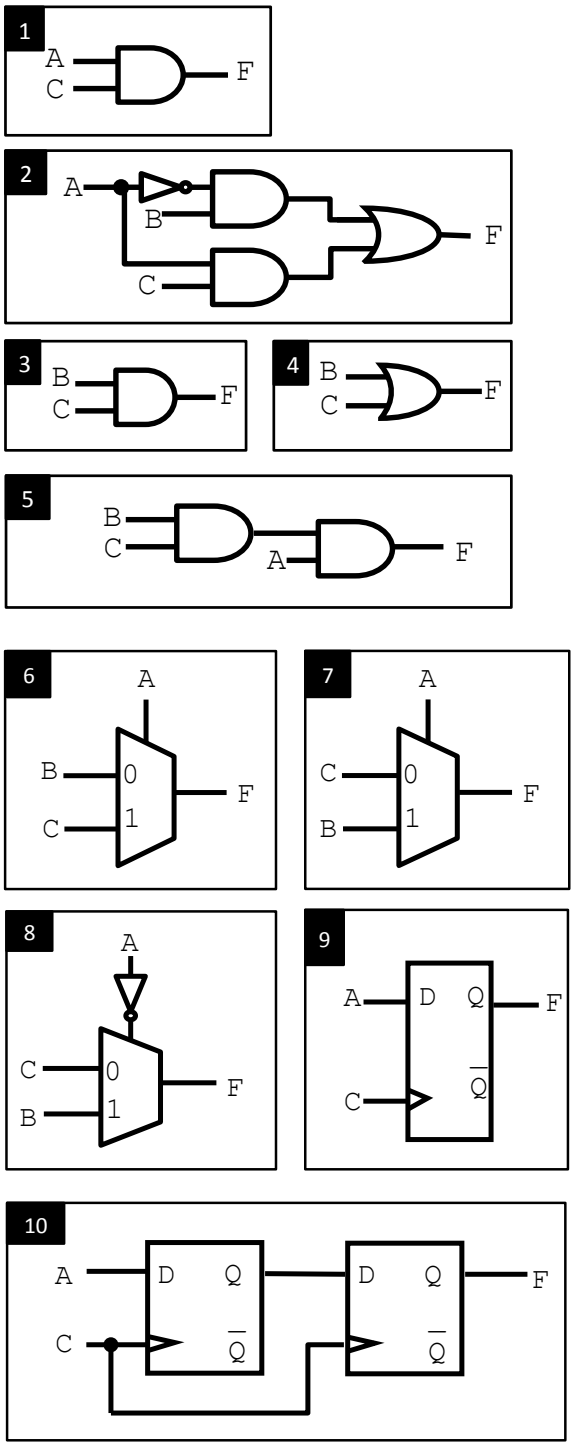
```
wire A, C;
reg B, F;

always @ (posedge C) begin
    B <= A;
    F <= B;
end
```

(v) **1**

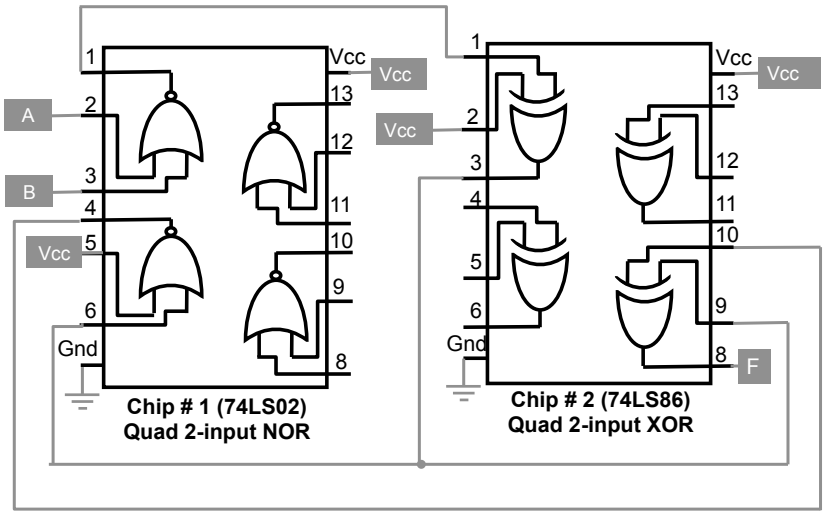
```
wire A, C;
reg B, F;

always @ (*) begin
    if (C == 1'b1) begin
        B = A;
        F = B;
    end
    else
        F = 0;
end
```



Question 3. [9 MARKS]

Your instructor wired up the following circuit in the lab and recorded some test results with the help of the logic probe for inputs A = 0 and B = 1. These measurements are listed in the table below. Note that the Vcc and Gnd pins of both chips were also measured and yielded correct results (i.e., logic-1 and logic-0 respectively).



Chip #	Pin #	Logic Probe Value (0 or 1)
1	1	0
1	2	0
1	3	1
1	4	1
1	5	1
1	6	0
2	1	0
2	2	1
2	3	0
2	8	1
2	9	0
2	10	1

(a) Based on those measurements for the above circuit, one or more gates from the two chips are malfunctioning. Circle the malfunctioning gates in the provided diagram above, and specify below the output pin(s) with incorrect logic value(s) in *(Chip #, Pin #)* format.

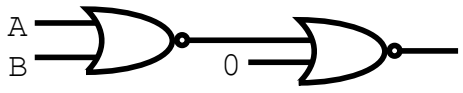
(Chip 1, Pin 4), (Chip 2, Pin 3)

(b) Determine the logic function F this schematic is implementing (assuming no malfunctioning gate(s)). You should write the function in its simplest possible form in the space below. The Boolean expression should include inputs A and B.

A + B (Alternate notations A or B, A ∨ B)

(c) Can you implement logic function F using only chip # 1 or only chip # 2? If yes, draw the circuit (just the gates and their connections - no chips) below. Also, use the least amount of gates possible.

Here are 2 possible implementations using 2 NOR gates:



Question 4. [14 MARKS]

Part (a) [8 MARKS]

You are to design a multiplier circuit that takes two unsigned 2-bit inputs $A(a_1a_0)$ and $B(b_1b_0)$, producing their 4-bit product, $P(p_3p_2p_1p_0) = A \times B$. We provide you with a truth table for all possible combinations of your inputs.

Use the Karnaugh maps below to find **optimized** Boolean expressions for your circuit. After you fill in the corresponding K-maps, show your groupings clearly. Write the Boolean expression corresponding to your groupings below each Karnaugh map.

Inputs				Outputs			
a_1	a_0	b_1	b_0	p_3	p_2	p_1	p_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

p_3 / a_1a_0 vs b_1b_0:

b_1b_0 \ a_1a_0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	1	0
10	0	0	0	0

p_2 / a_1a_0 vs b_1b_0:

b_1b_0 \ a_1a_0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	1
10	0	0	1	1

$p_3 = a_1a_0b_1b_0$

$p_2 = a_1b_1\bar{b_0} + a_1\bar{a_0}b_1$

p_1 / a_1a_0 vs b_1b_0:

b_1b_0 \ a_1a_0	00	01	11	10
00	0	0	0	0
01	0	0	1	1
11	0	1	0	1
10	0	1	1	0

p_0 / a_1a_0 vs b_1b_0:

b_1b_0 \ a_1a_0	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	0	1	1	0
10	0	0	0	0

$p_1 = a_1\bar{b_1}b_0 + a_1\bar{a_0}b_0 + \bar{a_1}a_0b_1 + a_0b_1\bar{b_0}$

$p_0 = a_0b_0$

Part (b) [6 MARKS]

You gain further knowledge that the input B will never be 3. How does this change your design? Show your redesigned Boolean functions below.

p_3 a_1a_0

b_1b_0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	X	X	X	X
10	0	0	0	0

p_2 a_1a_0

b_1b_0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	X	X	X	X
10	0	0	1	1

$p_3 = 0$

$p_2 = a_1b_1$

p_1 a_1a_0

b_1b_0	00	01	11	10
00	0	0	0	0
01	0	0	1	1
11	X	X	X	X
10	0	1	1	0

p_0 a_1a_0

b_1b_0	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	X	X	X	X
10	0	0	0	0

$p_1 = a_1b_0 + a_0b_1$

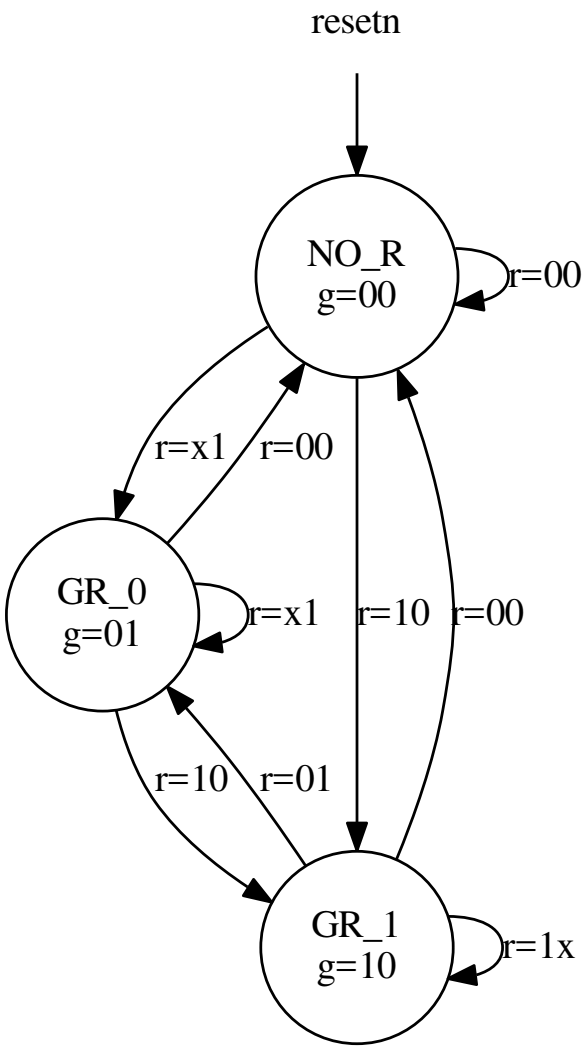
$p_0 = a_0b_0$

Question 5. [15 MARKS]

In many digital systems, there will be a single resource (maybe a multiplier or divider circuit) that will be shared by different subsystems. If more than one subsystem would like to use the single resource at the same time, there needs to be an *arbiter* that decides which subsystem gets to use the resource and which subsystem will wait. The following state diagram shows the FSM for a *priority arbiter* that arbitrates between two subsystems. The arbiter has two inputs, request0 and request1 ($r = \{r1, r0\}$) and two outputs grant0 and grant1 ($g = \{g1, g0\}$).

If subsystem0 wants access to the resource, it will raise r0 to 1 and wait for the arbiter to respond with g0 raised to 1. Once granted, the arbiter has to wait for the corresponding request line to be lowered before granting other requests. When subsystem0 is done using the resource, it will lower r0 to 0. The arbiter is then free to grant requests to other subsystems. The same principles apply for subsystem1 too. However, if a request comes from both subsystem0 and subsystem1 at the same time, when no other request is granted, the arbiter will prioritize subsystem0 by granting its request over subsystem1.

The state diagram below shows the three states of the arbiter, **No Request** (NO_R), **Grant 0** (GR_0), and **Grant 1** (GR_1) as well as the transitions between the corresponding states. A transition labelled r=x1 means that the transition will take place if r=01 or r=11.



Part (a) [10 MARKS]

First fill in the State Table below. Next, given the following state assignments, complete the State Assigned Table and derive the next state and output logic for the priority arbiter. There are two state bits, $S1$ and $S0$, that are implemented using flip-flops. There are alternate formats for the state table and the state-assigned table on page 11. Specify which *table format* you want us to mark or 0 marks will be assigned if both are used.

State Assignments

State	$S1$	$S0$
NO_R	0	0
GR_0	0	1
GR_1	1	0

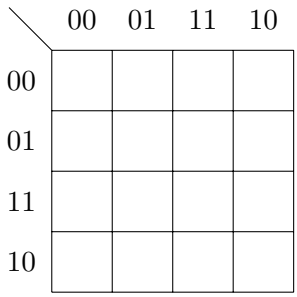
State Table: (Mark this table: Yes / No)

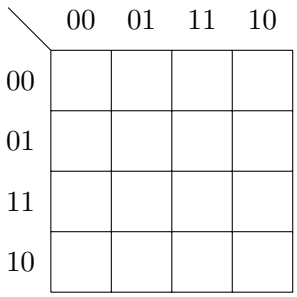
CurrentState	*NextState*			
	$r = 00$	$r = 01$	$r = 10$	$r = 11$
NO_R	NO_R	GR_0	GR_1	GR_0
GR_0	NO_R	GR_0	GR_1	GR_0
GR_1	NO_R	GR_0	GR_1	GR_1

State Assigned Table: (Mark this table: Yes / No)

CurrentState	*NextState*								*Outputs*	
	$r = 00$		$r = 01$		$r = 10$		$r = 11$			
$S1S0$	$S1$	$S0$	$S1$	$S0$	$S1$	$S0$	$S1$	$S0$	$g1$	$g0$
00	0	0	0	1	1	0	0	1	0	0
01	0	0	0	1	1	0	0	1	0	1
10	0	0	0	1	1	0	1	0	1	0

Next State Logic: Use K-maps if needed (the K-maps will not be marked). Also use the following notation. $S1_D/S1_Q$ refers to the input/output of the flip-flop storing $S1$. $S0_D/S0_Q$ refers to the input/output of the flip-flop storing $S0$.





$S1_D = r_1\bar{r_0} + S1_Q r_1$

$S0_D = \bar{r_1} r_0 + r_0 \overline{S1_Q}$

Output Logic:

$g1 = S1_Q$ $g0 = S0_Q$

Part (b) [5 MARKS]

If the following "one-hot" encoding is used, what is the corresponding Next State and Output Logic?

State	$S2$	$S1$	$S0$
NO_R	0	0	1
GR_0	0	1	0
GR_1	1	0	0

Use the following notation. $S2_D/S2_Q$ refers to the input/output of the flip-flop storing $S2$. $S1_D/S1_Q$ refers to the input/output of the flip-flop storing $S1$. $S0_D/S0_Q$ refers to the input/output of the flip-flop storing $S0$.

Next State Logic:

$S2_D = S2_Q r_1 + S1_Q r_1 \bar{r_0} + S0_Q r_1 \bar{r_0}$

$S1_D = S2_Q \bar{r_1} r_0 + S1_Q r_0 + S0_Q r_0$

$S0_D = S2_Q \bar{r_1} \bar{r_0} + S1_Q \bar{r_1} \bar{r_0} + S0_Q \bar{r_1} \bar{r_0}$

Output Logic:

$g1 = S2_Q$

$g0 = S1_Q$

Alternate Format Tables. Only use these if you answered No for the state and state assigned tables on page 9.

Alternate Format for State Table: (Mark this table: Yes / No)

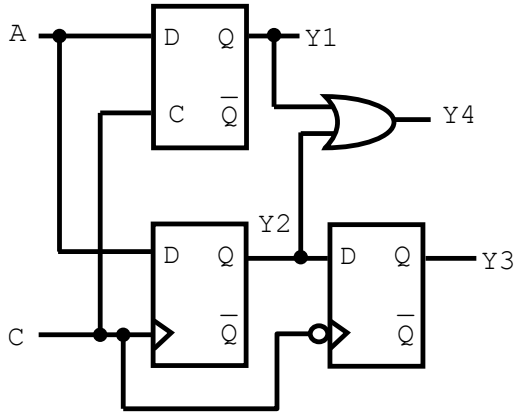
CurrentState	*Inputs*		*NextState*
	r1	r0	
NO_R	0	0	
	0	1	
	1	0	
	1	1	
GR_0	0	0	
	0	1	
	1	0	
	1	1	
GR_1	0	0	
	0	1	
	1	0	
	1	1	

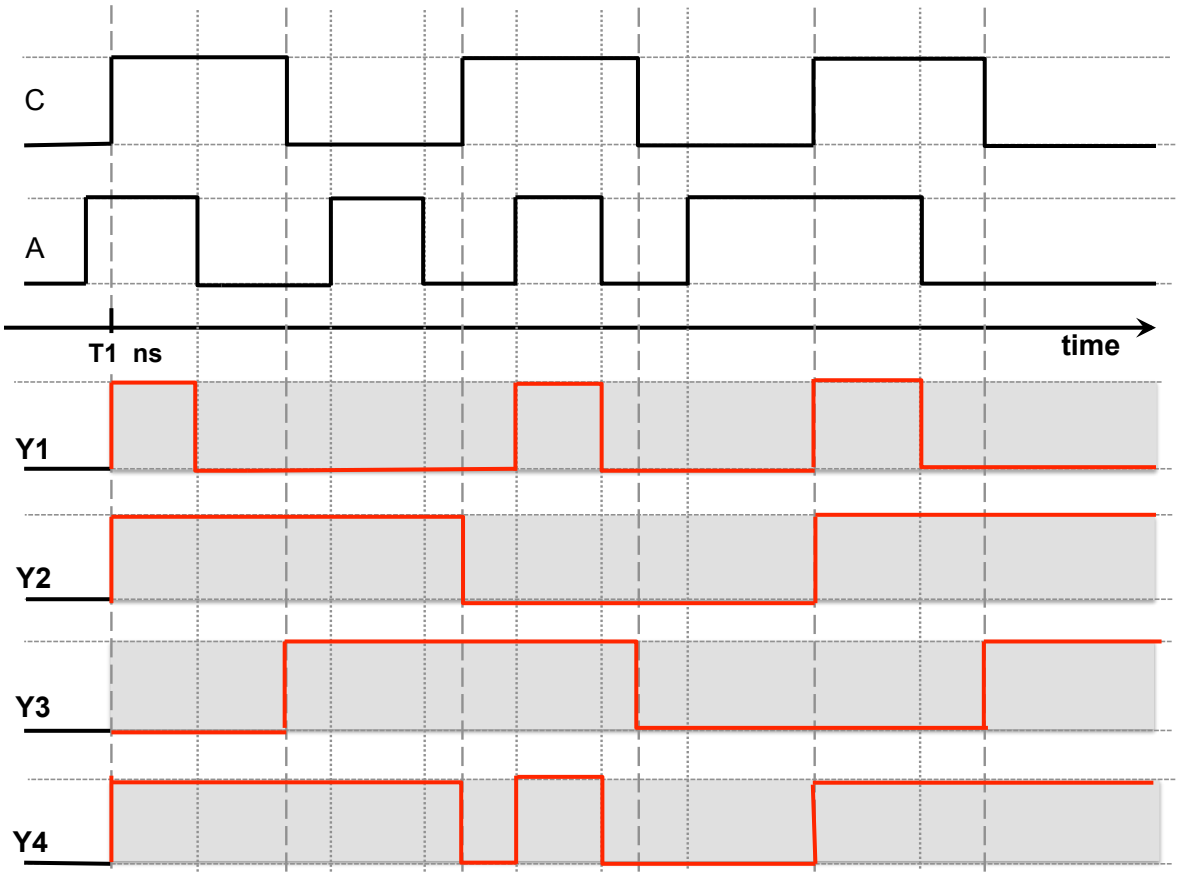
Alternate Format for State Assigned Table: (Mark this table: Yes / No)

CurrentState		*Inputs*		*NextState*		*Outputs*	
S1	S0	r1	r0	S1	S0	g1	g0

Question 6. [8 MARKS]

You are given the following circuit along with waveforms for its two inputs A and C. Fill in the waveforms for Y1, Y2, Y3 and Y4 in the gray highlighted space provided below (i.e., starting from time T1 ns). Assume that Y1, Y2, Y3 and Y4 all have initial values of zero as shown in their waveforms. You may also assume that signals change instantly.



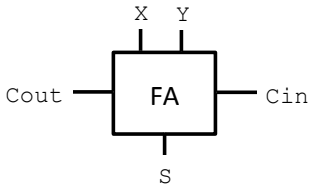


Question 7. [10 MARKS]

You are asked to design a circuit that computes the absolute value of a signed number W. Your input W is a 4-bit signed number (in 2's complement form) which is only permitted to have values from -7 to 7, including those numbers.

Here are the constraints on the resources you can use in your design:

- You **must** use **full-adders** (FA schematic shown below) in your circuit. We have already provided 3 full-adders for you to include (i.e., wire) in your design.

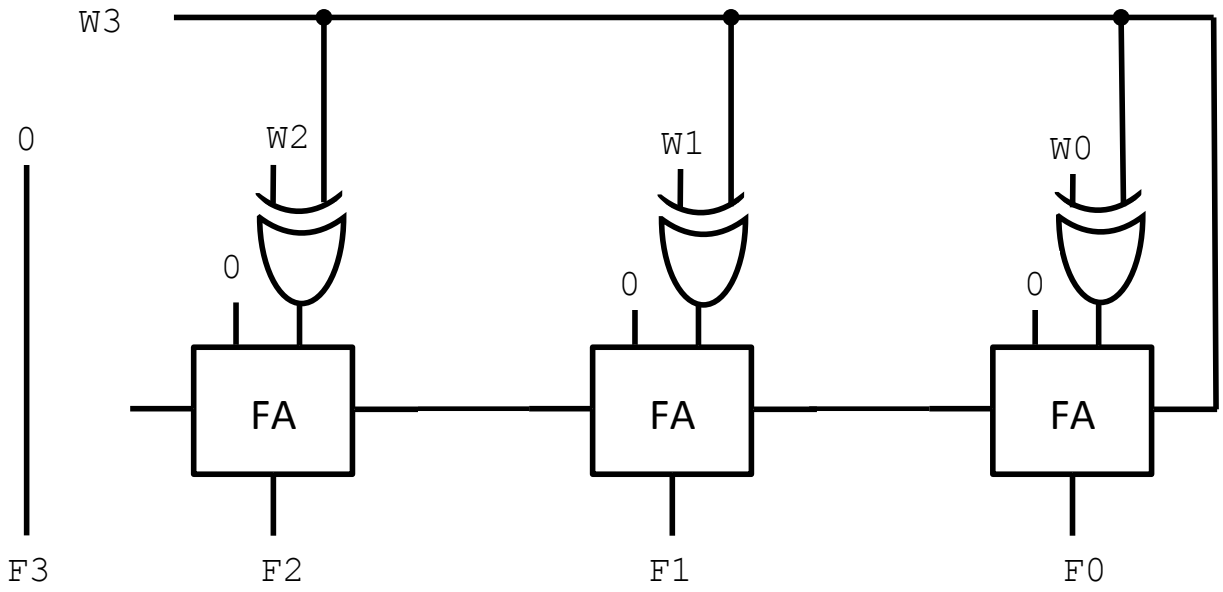


- You can also use multiple logic gates and/or logic devices in your circuit. However, only optimized designs will be awarded full marks.

Advice:

- Use the notation $W_3W_2W_1W_0$ for your input W and $F_3F_2F_1F_0$ for your output F (i.e., the computed absolute value of W).
- Think about how you can identify if a signed-number is positive or negative.
- You are allowed to hard-wire inputs (other than W of course!) to logic-0 or logic-1 as needed.

Here is one possible solution. We use the adder/subtractor circuit shown in class to either do 0 + W for a positive W, or 0 - W for a negative W. W_3 is the sign bit.



Question 8. [10 MARKS]

Using T-Flip-Flops, design a 3-bit *up/down-counter* that counts up or down depending on the $u/\bar{d}$ signal. If the *en* signal is high and $u/\bar{d}$ is also high, the counter should count up on each positive clock edge. If the *en* signal is high and $u/\bar{d}$ is low, the counter should count down on each positive clock edge. If the *en* signal is low, the counter should preserve its value over each clock edge.

Use the following template along with **any** other necessary **gates** or **logic devices** to create the up/down-counter. All signals are inputs except for q, which should be your counter output. The clock signal and resetn signals have already been wired and should not be changed.