EE109: Introduction to Embedded Systems
Spring 2021 - Final Exam
5/8/21, 2PM – 3:40PM + 20 min to upload

[Complete all the information in the box below.]

<No Need to Fill this out due to Gradescope>

Name:___Solutions__________

Student ID: ____

Email: _____________________@usc.edu

~~**Lecture section (Circle One):**~~

~~Redekopp~~	~~Redekopp~~	~~Weber~~	~~Qian~~
~~9:30 a.m.~~	~~11 a.m.~~	~~12:30 p.m.~~	~~2 p.m.~~

Calculators ARE allowed.

Page	Ques.	Your score	Max score	Recommended Time
			0	0 min.
	1		8	8 min.
	2		12	10 min.
	3		10	12 min.
	4		12	13 min.
	5		10	15 min.
	6		10	12 min.
	7		12	15 min.
	8		8	15 min.
	Total		82	100 min.
		Scan/Upload		20 min.

1. **Multiple Choice / Short answer (8 pts.)**: Answer the questions below.

1.1 An _______ (**ASIC** / **FPGA**) cannot be reconfigured and is fixed once fabricated.

1.2 A state machine with 4 flip-flops can implement a state machine with a maximum of ___ (**2 / 4 / 8 / 16**) states.

1.3 **True / False** ___ Caching breaks logic into multiple stages to overlap their execution.

1.4 Having completed the labs of EE 109, a student should know NOT to:

a.) Use volatile variables in an ISR
b.) Use 'float' and 'double' types if it can be avoided
c.) Use global variables in embedded programs
d.) Use nested if statements to implement a state machine

1.5 To ensure devices correctly interpret the **timing** of bits sent over an asynchronous RS-232 connection, both devices must use a common

a.) Ground signal
b.) Baud rate
c.) Prescaler

1.6 An edge-triggered D flip-flop can be built from how many level-sensitive D-Latches?

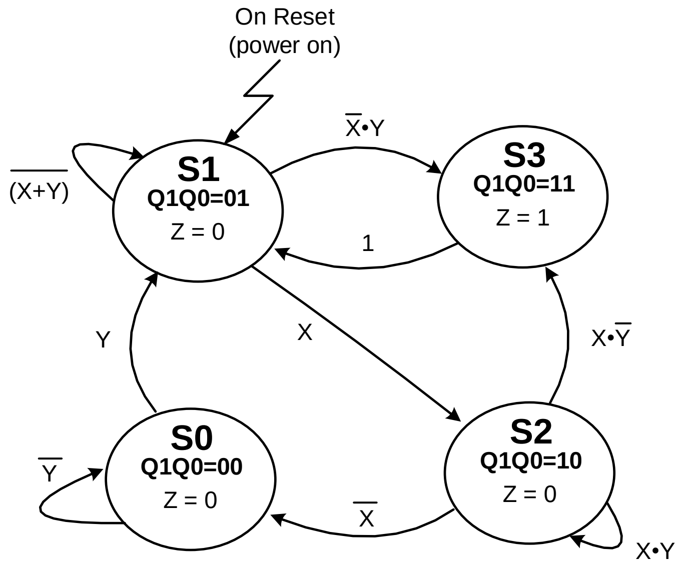
a.) 1
b.) 2
c.) A D flip-flop cannot be built from level-sensitive D-Latches

1.7 A low-resistance (R=0) pathway between two points with a voltage difference is referred to as a(n) ______ (**short / open / high-impedance**) circuit.

1.8 To make a signal a digital output on the Arduino, set the appropriate bit of the _________ (general name of the) register to a ________ (1 / 0).

a.) DDR / 0
b.) PORT / 0
c.) DDR / 1
d.) PORT / 1

2. **State Machines I (12 pts.)**: Consider the **completed** state diagram shown below to answer the questions below.



a.) Complete the state transition table by filling in the next state columns and the output column in the table below.

Current State		Next State				Output
		X Y = **0 0**	X Y = **0 1**	X Y = **1 0**	X Y = **1 1**	
State	Q1 Q0	State*	State*	State*	State*	Z
S0	0 0	S_0_	S_1_	S_0_	S_1_	_0_
S1	0 1	S_1_	S_3_	S_2_	S_2_	_0_
S2	1 0	S_0_	S_0_	S_3_	S_2_	_0_
S3	1 1	S_1_	S_1_	S_1_	S_1_	_1_

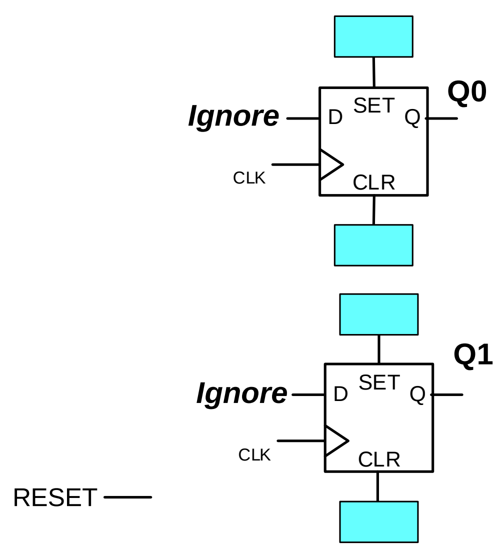
b.) To implement the reset condition, what should be connected to the following flip flop inputs?

i.) The **SET of the Q0 flip-flop** should be connected to:
- **a. RESET**
- b. ~RESET
- c. 0 (GND)
- d. 1 (Vdd)

ii.) The **CLR of the Q0 flip-flop** should be connected to:
- a. RESET
- b. ~RESET
- **c. 0 (GND)**
- d. 1 (Vdd)

iii.) The **SET of the Q1 flip-flop** should be connected to:
- a. RESET
- b. ~RESET
- **c. 0 (GND)**
- d. 1 (Vdd)

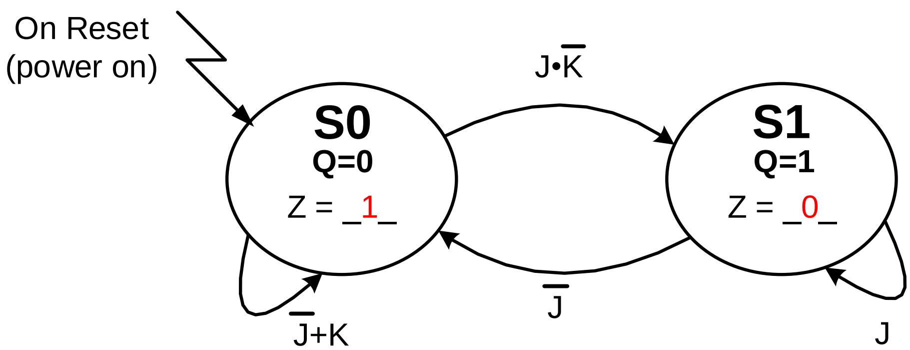
iv.) The **CLR of the Q1 flip-flop** should be connected to:
- **a. RESET**
- b. ~RESET
- c. 0 (GND)
- d. 1 (Vdd)



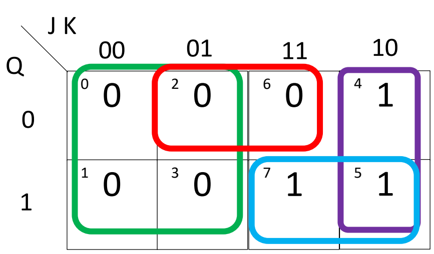
3. **State Machines II (10 pts).** You are given a state machine with 1 flip-flop, **Q**, and two inputs: **J** and **K** and two states: S0 and S1 whose desired behavior is shown in the table below. Answer the following questions.

Current State		Next State								Output
		J K = **0 0**		J K = **0 1**		J K = **1 0**		J K = **1 1**		
State	Q	State*	Q*	State*	Q*	State*	Q*	State*	Q*	Z
S0	0	S0	0	S0	0	S1	1	S0	0	1
S1	1	S0	0	S0	0	S1	1	S1	1	0

a.) Use the state table above to complete the corresponding state diagram (***fill in/draw all the correct state transitions*** and be sure to label them correctly based on the table). For each transition you must arrive at a minimal SOP expression (i.e. combine multiple transitions to the same state to form a single, minimal SOP expression for the transition condition). ***Fill in the Z output values*** for each state.



b.) Find a **minimal, POS** equation for D (input to the **single** flip-flop, Q) and a **minimal equation** (SOP or POS) for Z. **Show your work below** (to get full credit) and put your final answer in the blanks below.



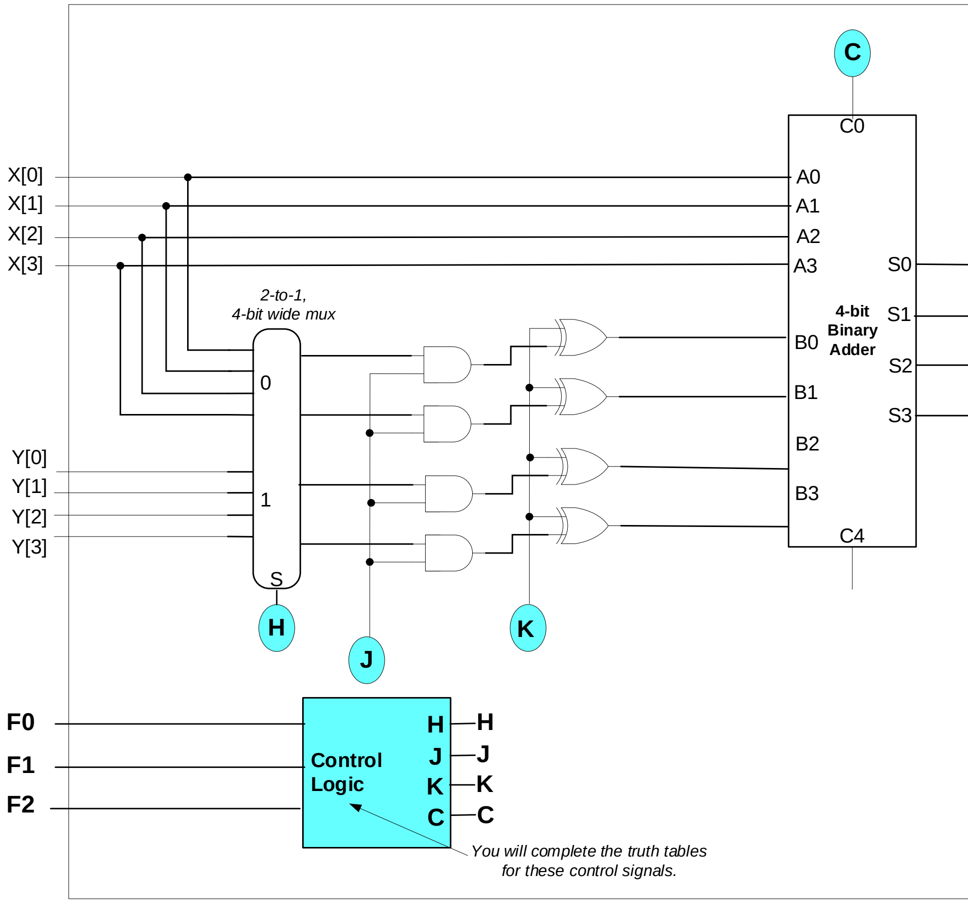
Dpos = J• (K'+Q)

Dsop = JK' + JQ <- **Half credit for SOP**

D (minimal **POS**) = ___see above___

Z = _ **not Q (i.e. ~Q, or Q')**_

4. **Datapath Design I (10 pts.):** Consider the datapath below with the accompanying table showing the correspondence of the function select bits **F[2:0]** to the **resulting arithmetic operation performed to produce the output R[3:0].**. <u>**Complete the table**</u> for the control signals: H, J, K, C to achieve the desired operations. Finally, find the logic for (only) the signals, **H** and **K**. All input and output numbers are 2's complement numbers. Do not worry about overflow.



Complete the table for the values of H, J, K, and C. Use **d** for don't care where appropriate.

F2	F1	F0	H	J	K	C	Desired Arithmetic Operation (resulting output for R[3:0])
0	x	x	d	0	0	1	**X[3:0]+1**
1	0	0	d	0	1	0	**X[3:0] - 1**
1	0	1	1	1	1	1	**X[3:0] – Y[3:0]**
1	1	0	1	1	0	0	**X[3:0] + Y[3:0]**
1	1	1	0	1	0	0	**2*X[3:0]**

What is the minimal SOP logic for H : ___ F1' + F0'_________________________

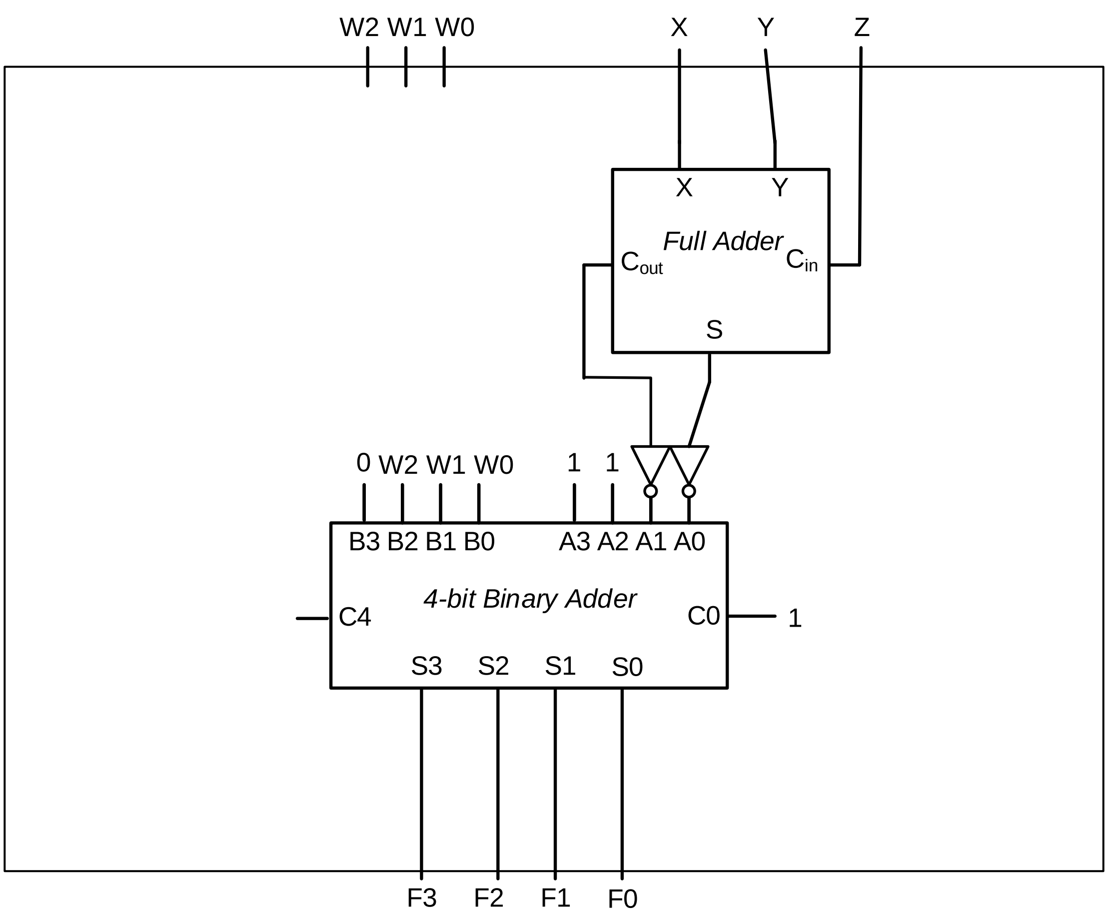
What is the minimal SOP logic for K : ____F2 • F1'_______________________________

5. **Adder Design (10 + 6 pts.)**: You are given a **<u>3-bit unsigned</u>** number, W[2:0], and three single bit inputs: X, Y, and Z. Design a circuit that generates a **2's complement system output F = W – X – Y – Z**.

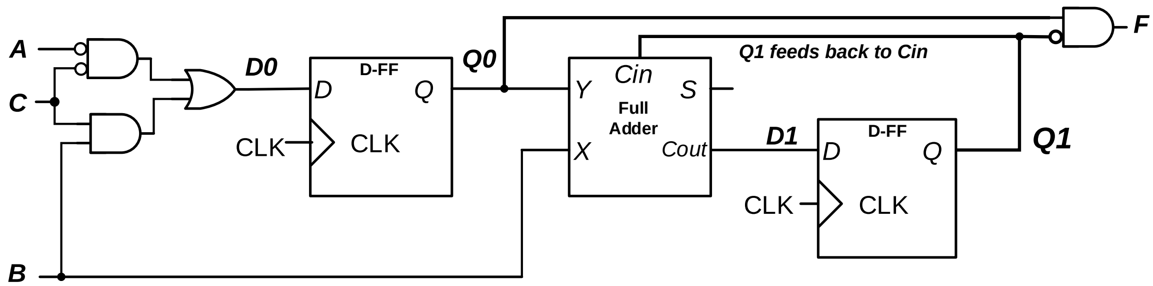
   a. **Complete the statement to make it true**:
   The output values for F can range from __**-3**_____ decimal to ___**+7**____ decimal.

   b. **Complete the statement to make it true**:
   To represent the range of F you found above requires ___**4**______ (how many) bits.

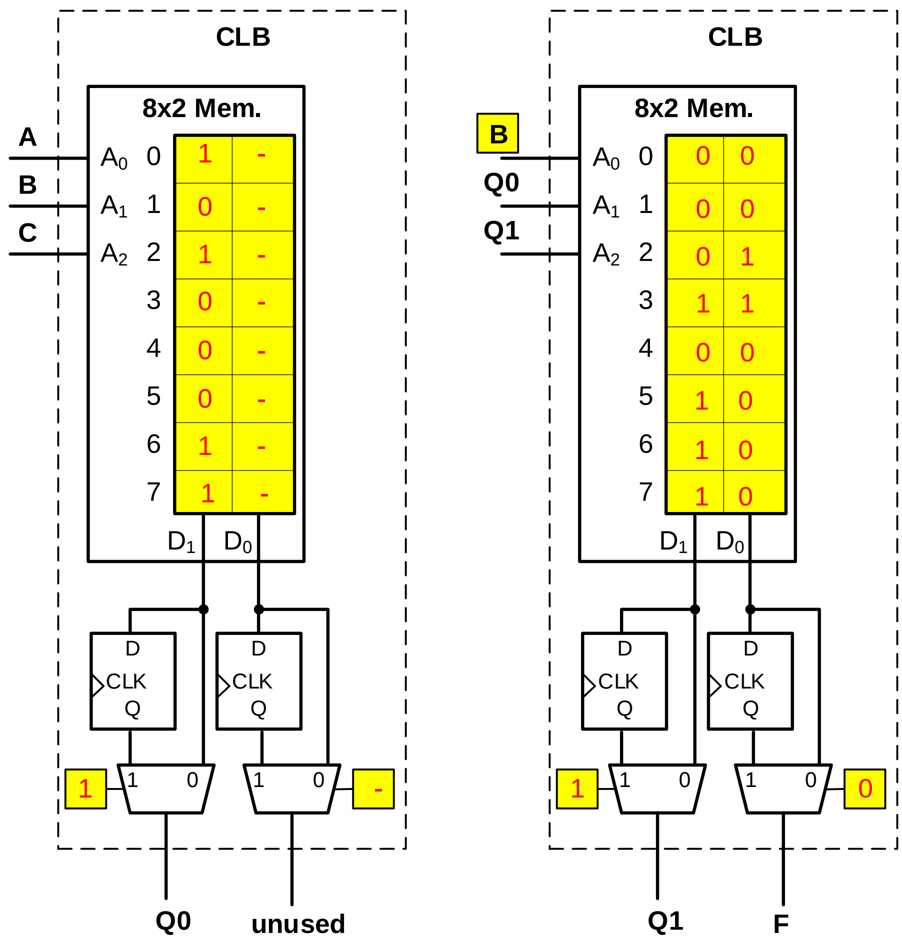
   c. **Design** the circuit to generate the appropriate number of outputs for F using a single 4-bit adder and a **<u>minimal number</u>** of full and half adders along with simple AND, OR, NOT gates**.** [**Hint: W-X-Y-Z = W – (X+Y+Z)**]



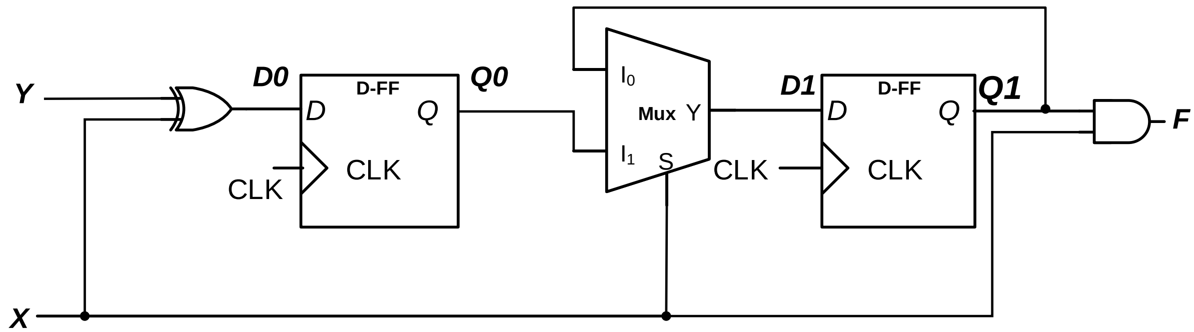
6. **FPGAs/Memories Design (10 pts.)**: Consider the circuit shown below. Show how to implement the design using the <u>two</u> 3-input, 2-output CLBs below by determining the contents of the 8x2 memory and the mux selects. If necessary, place a **dash ('-')** in any memory cell (bit place) that is a don't care.

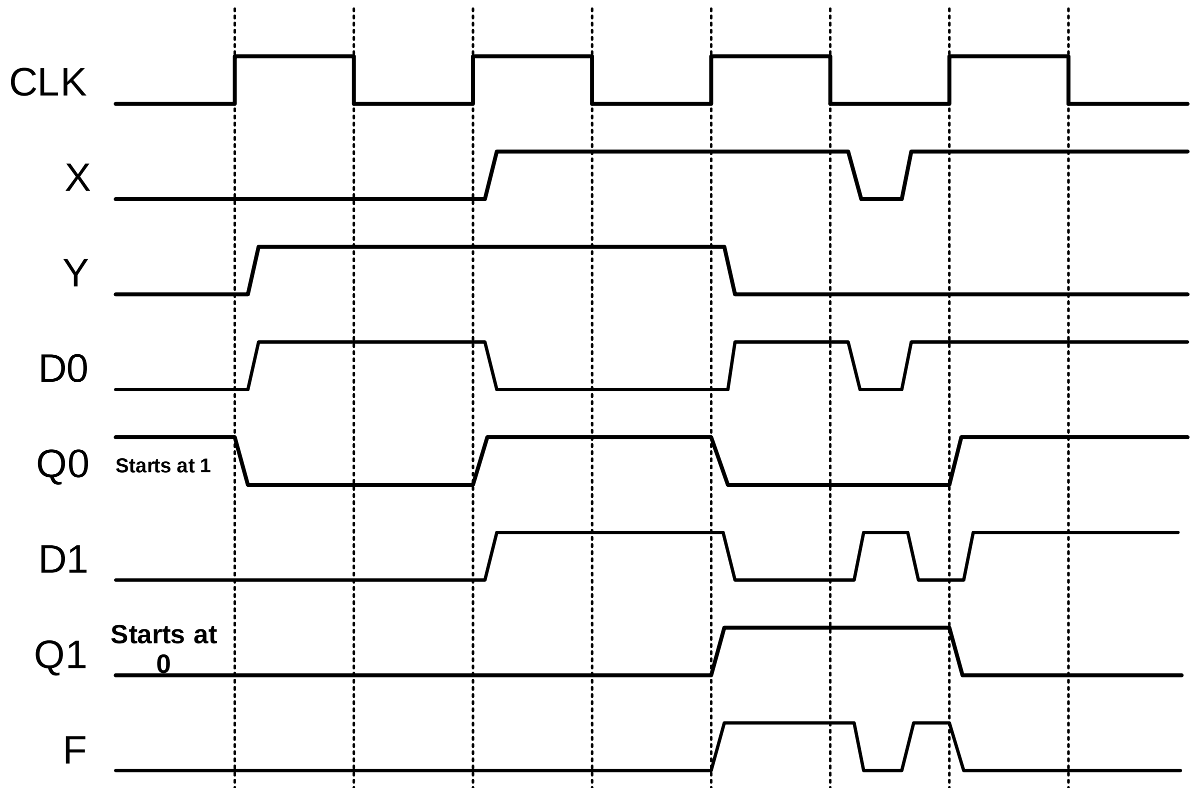


Circuit to be implemented using the CLB's below

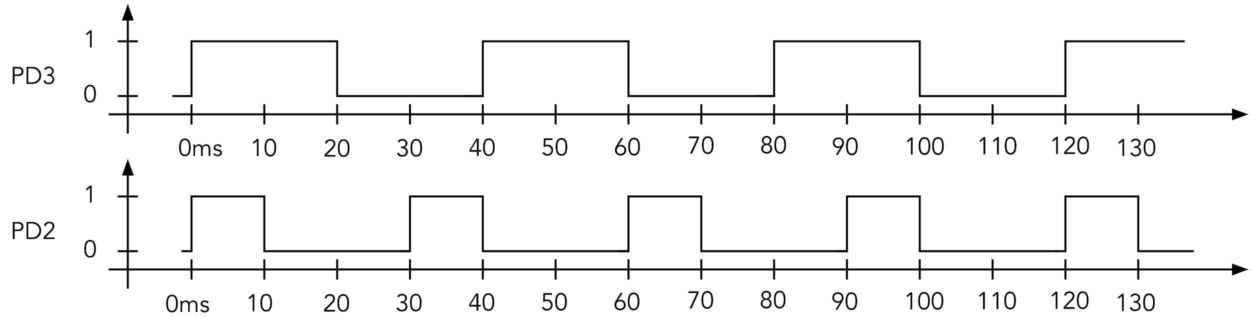


7. **Sequential Logic (12 pts.)**: Complete the waveform diagram below for D0, Q0, D1, Q1 and F. Assume the D flip-flops shown below are **positive-edge triggered.** **Q0 starts at 1 and Q1 starts at 0.** (You can determine the starting value of D0, D1, and F from the given information.)





8. **Interrupts and Arduino Coding (8 pts.)** Your Arduino needs to output the **two** periodic signals shown below (which only show a few cycles of the desired pattern which continue indefinitely) on Port D, bit 2 (PD2) and Port D, bit 3 (PD3). The signal on PD2 has a period of 30ms and duty cycle of 33.3% (high for 10ms, low for 20ms). The signal on 3 has a period of 40ms and duty cycle of 50% (high for 20ms, low for 20ms).



You must use the 16-bit TIMER1 to produce both signals. **You may not use any delay functions to produce the output.** Determine the prescaler and counter modulus value (max count) to store in the OCR1A register, and write the ISR (TIMER1_COMPA_vect) that will generate the desired signals. If multiple pairs of prescalar and modulus values will work, use ones that give the more accurate timing. **Note**: The Arduino system clock is 16MHz. *(note: 1 millisecond (ms) =* 10^{-3} *seconds, 1 microsecond (*μs*)=* 10^{-6} *seconds)*

Complete your answers in the linked sp21-fi-isr.c file and then upload the completed file to Gradescope. The contents are reproduced below.

```
// Inidicate what prescalar you choose by deleting the options you don't
// want then write your chosen OCR value below.
// ---------- FILL IN THE 2 LINES BELOW ----------------
// Prescalar:  1 / 8 / 64 / 256 / 1024
// OCR1A: __________
Working prescalar/OCR1 combinations:  8 and 20,000 /  64 and 2500 /  256 and 625
// Assume the outputs have been initialized and set to output a '1', the
// TIMER1 is initialized correctly with the values you chose above, and
// that interrupts have been enabled so that the ISR below is called each time
// TIMER1 reaches your OCR1A value.
// ---------- COMPLETE THE CODE BELOW ----------------
// declare any needed global variables here

See accompanying source solution file

ISR(TIMER1_COMPA_vect)
{  // show ISR code here..NO delay functions are allowed (losing all credit)
See accompanying source solution file



}
```

Intentionally blank for scratch work. You may detach it but please turn it in with your exam:

Name: ____ Section time: _____________

Single-Variable Theorems

(T1)	X + 0 = X	(T1')	X • 1 = X	(Identities)
(T2)	X + 1 = 1	(T2')	X • 0 = 0	(Null elements)
(T3)	X + X = X	(T3')	X • X = X	(Idempotency)
(T4)	(X')' = X			(Involution)
(T5)	X + X' = 1	(T5')	X • X' = 0	(Complement)

Two- and Three-Variable Theorems

(T6)	X +Y = Y + X	(T6')	X • Y = Y • X	(Commutativity)
(T7)	(X+Y)+Z = X+(Y+Z)	(T7')	(X•Y) •Z = X• (Y•Z)	(Associativity)
(T8)	X•(Y+Z) = X•Y + X•Z	(T8')	X+(Y•Z) = (X+Y) • (X+Z)	(Distributivity)
(T9)	X + X•Y = X	(T9')	X • (X + Y) = X	(Covering)
(T10)	X•Y + X•Y' = X	(T10')	(X+Y) • (X+Y') = X	(Combining)
(T11)	X•Y+X'•Z+Y•Z = X•Y+X'Z	(T11')	(X+Y)•(X'+Z)•(Y+Z) = (X+Y)•(X'+Z)	(Consensus)

DeMorgan's Theorem

(X • Y)' = X' + Y'	(X +Y)' = X' • Y'	(DeMorgan's)