ECE 2020 IE3, Fall 2024

Problem Set #4

Due: November 8 (+2 bonus)
Hard Deadline: November 10

6 problems, 24 points possible

Problem 1: Resilient Electric Microgrid

You are installing a microgrid system within a rural area of the power grid. This system lets the rural area to disconnect—or *island*—itself from a nearby city's power grid during extreme weather events, and power itself via peer-to-peer sharing local renewable energy until the connection can be safely restored. The set of states for the system are:

1. *Interconnected*: The microgrid operates normally, and draws most of its power from the city grid.
2. *Disconnecting*: The microgrid decreases the electric power it draws from the city grid.
3. *Islanded*: The microgrid is using peer-to-peer renewable energy and battery storage sharing.
4. *Connecting*: The microgrid begins reconnecting to the city grid.
5. *Blackout*: There is no power available, and the microgrid cannot operate.

The microgrid behaves in the following way:

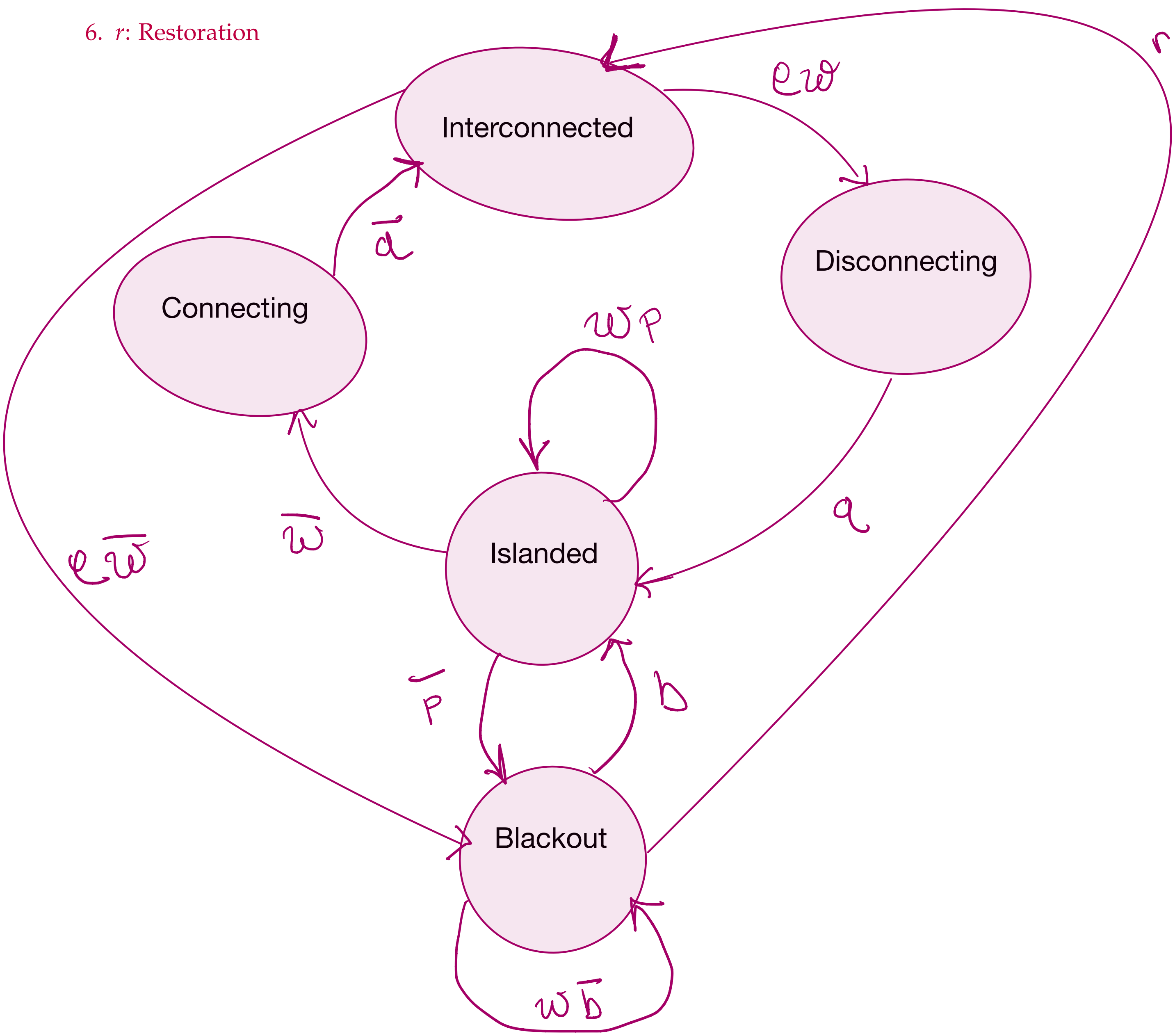
1. If an event occurs and a warning is received, the microgrid begins disconnecting from the city grid, until its control system is online, then it moves into *islanded* mode. It stays in *islanded* mode while the event warning is active and it has renewable power available.
2. If an event occurs and the city grid fails to send a warning, the microgrid goes into *blackout*.
3. If the microgrid is in island mode and it runs out of renewable power, it goes into *blackout*.
4. If the microgrid has battery energy storage while in blackout, it returns to *islanded* mode.
5. When the extreme event warning is cleared, the microgrid goes into *connecting* mode.
6. When the control system is offline and the warning is cleared, the microgrid goes into *interconnected* mode.
7. If the microgrid has no battery energy storage, and the warning is still active, it stays in *blackout* until a restoration occurs, directly taking it back to being *interconnected*.

Problem 1 (2pts)
Sketch a state machine diagram for the microgrid system. You don't need to use any numbers, you can just use words. It might be helpful to define some symbols.

Solution 1

Define the following symbols:

1. e: A disasterous event occurs
2. w: A warning signal is received
3. p: Renewable power is available
4. a: Control system is active
5. b: Battery storage is available
6. r: Restoration



Problem 2: Return of the Smart Thermostat

Consider the smart thermostat system from Problem Set 1. Equipped with your new knowledge of *sequential logic*, you consider the following inputs for a state machine representation:

1. Input 1: A "too cold" signal $T_C \in \{0, 1\}$, where $T_C = 1$ if the house is too cold, that is, temperature $<$ comfort$^\circ$.
2. Input 2: A "too hot" signal $T_H \in \{0, 1\}$, where $T_H = 1$ if the house is too hot, that is, temperature $>$ comfort$^\circ$.

The state machine has two outputs:

1. Heating mode, $H \in \{0, 1\}$: The HVAC system warms the household
2. Cooling mode, $C \in \{0, 1\}$: The HVAC system cools the household

Suppose that you want to represent your design via a Moore State Machine. The state machine behaves according to these rules:

1. If the the house is too cold, on the next clock edge, the HVAC should begin heating. If the temperature is no longer too cold, the heater should turn off on the next clock edge.
2. If the the house is too hot, on the next clock edge, the HVAC should begin cooling. If the temperature is no longer too hot, the cooler should turn off on the next clock edge.
3. However, after the heater or cooler turns off, *neither may turn on again* for two clock cycles, even if the temperature is out of the desired range.

Assume that the "too hot" and "too cold" signals are never asserted at the same time, that is $T_C \cdot T_H = 0$ for all time.

Problem 2 (6pts +2 bonus)

Update the smart thermostat system into a state machine form.

1. Design a Moore state machine that implements the smart thermostat.
2. Suppose the clk signal, T_C, and T_H signals evolve over time as shown in Fig. 1. Sketch the heating and cooling signals H and C for your smart thermostat system. You can assume a rising edge trigger.
3. Propose an improved design of this system that uses the motion detection signal M from Problem Set 1; specifically, replace the two clock-cycle delay in the problem statement with requiring $M = 1$.
4. **[BONUS]** Draw a state-transition table and state machine for your improved design.

Solution 2

Note: For this problem, lots of solutions are possible, depending on your interpretation of the design. I will present a solution that has 5 states, as that makes the most sense to me. We will accept many different answers for this question.

1. Let's define the set of all states:

$$\mathcal{Q} = \{S_0, S_1, S_2, S_3, S_4\},$$

   where:

   (a) S_0 is a "ready" state, also the **initial state**, that indicates the system is ready to activate the HVAC (i..e, at least 2 cycles after previous shut off, or the system is currently at the initial state.)

   (b) S_1 is the *heating* state

   (c) S_2 is the *cooling* state

   (d) S_3 is a state that represents the system being *off and not ready to change after 0 cycles have passed*.

   (e) S_4 is a state that represents the system being *off and not ready to change after 1 cycle has passed*.

   We can define the set of all inputs as

$$\mathcal{X} = \{T_C, T_H\},$$

   and the set of all outputs as

$$\mathcal{F} = \{H, C\}.$$

   The state diagram has been sketched on the subsequent page.

2. The timing diagram has been sketched in Fig. 1

3. To improve the design, you can use just 4 states–let S_0, S_1, and S_2 be the same, but now define a state of the form

$$S_3 = \begin{cases} 1 & \text{system off and } M = 0. \\ 0 & \text{otherwise.} \end{cases}$$

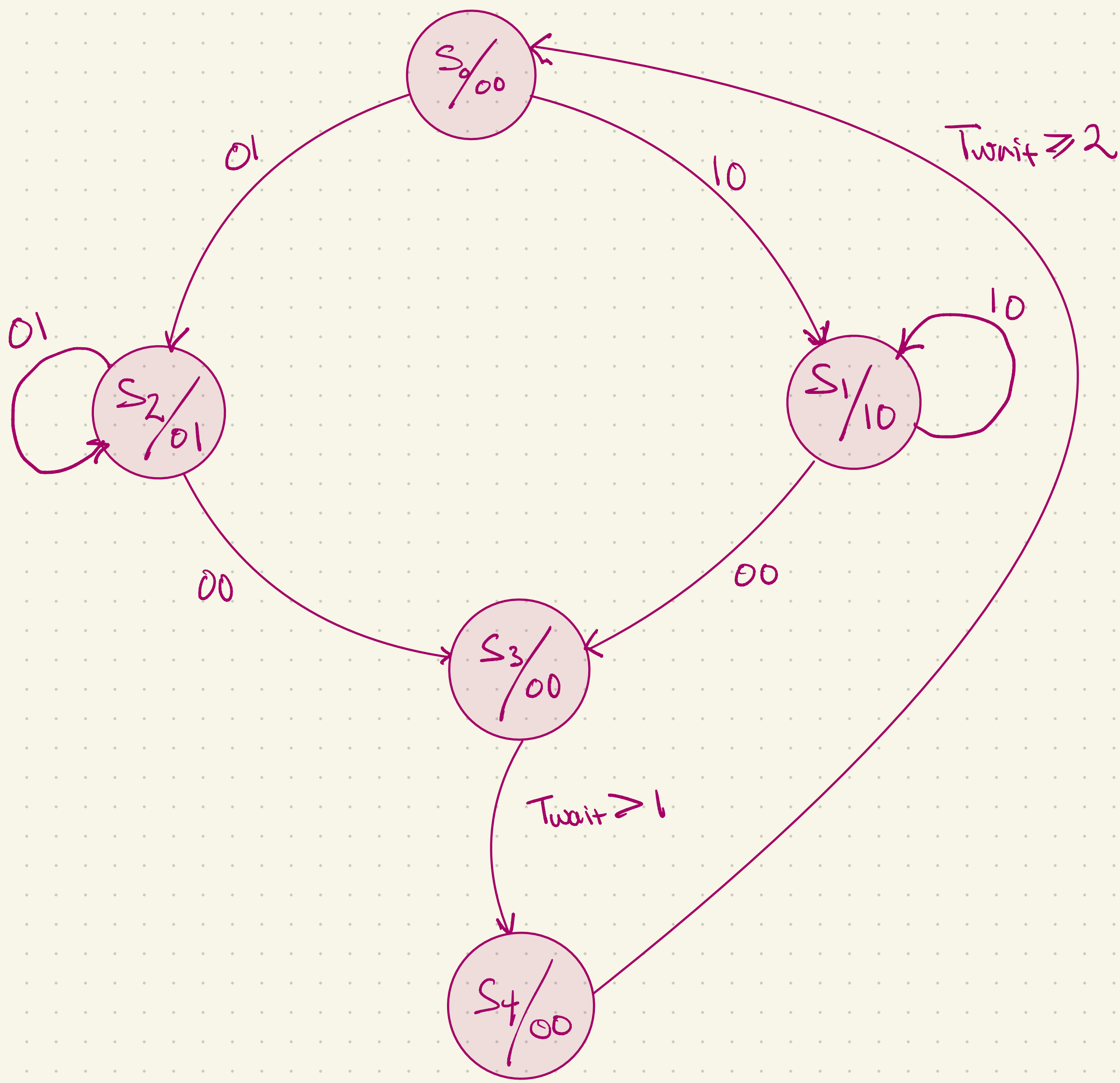
   and let the inputs now take the form

$$\mathcal{X} = \{T_C, T_H, M\}.$$

   with the outputs unchanged. The new state diagram is sketched on the subsequent page.

Part 1:

* Let $T_{wait} \equiv$ total amount of cycles passed since off
* Let the input be $X = (T_C\, T_H) = (X_1 X_0)$
* Le the output be $F = (H\, C) = (F_1 F_0)$



Assuming rising edge trigger (falling also accepted)

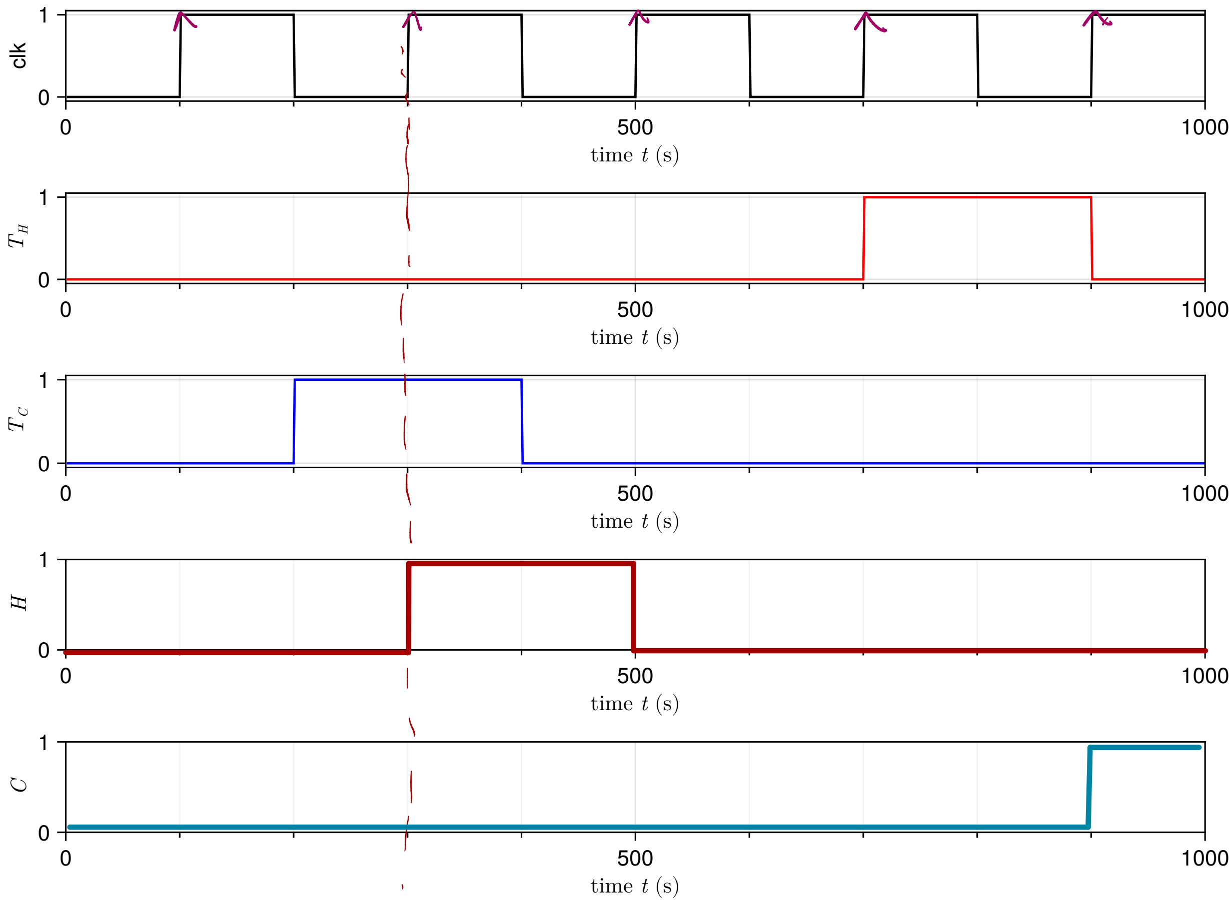
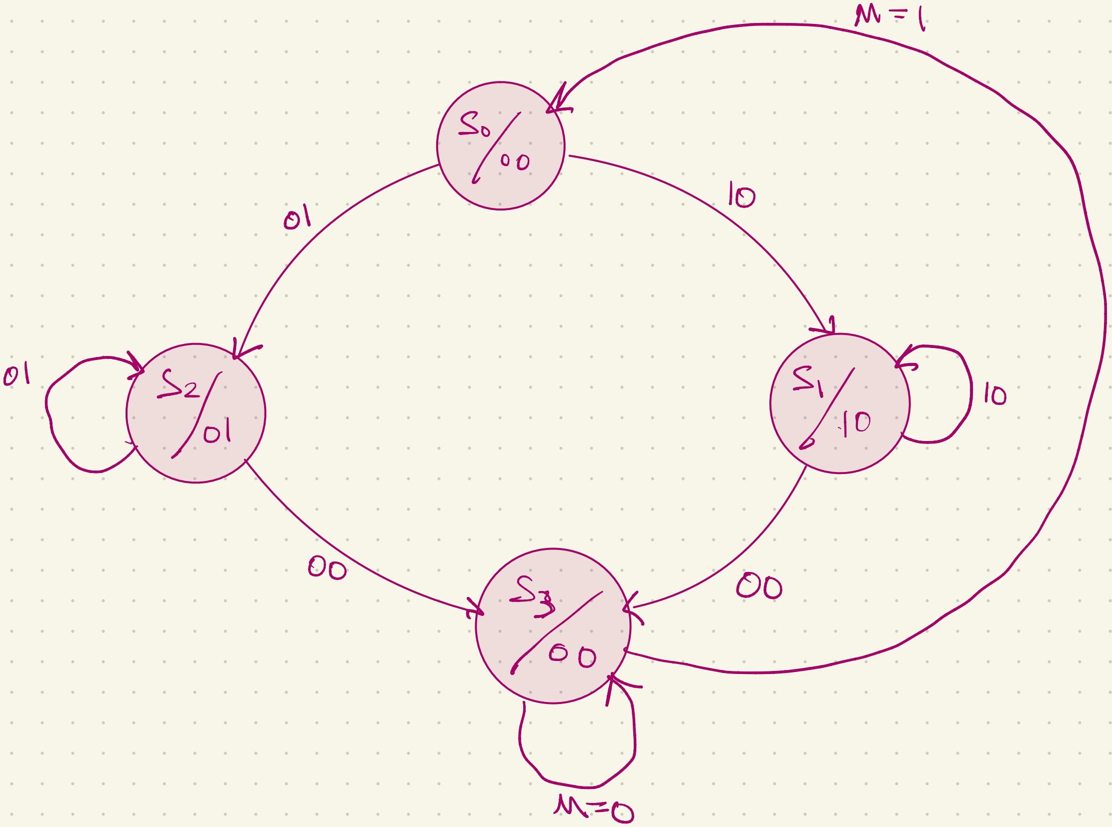


Figure 1: Solution: Timing diagram for the smart thermostat

Part 3/4 :



When M=1 in state 3, reset. Can be ignored on other states for simplicity.

Problem 3: Mealy and Moore

Consider a Mealy finite state machine with states

$$\mathcal{Q} = \{\boldsymbol{q} : \boldsymbol{q} \in \{0,1\}^2\},$$

and inputs and outputs

$$\mathcal{X} = \{x : x \in \{0,1\}\} \qquad \text{and} \qquad \mathcal{Y} = \{y : y \in \{0,1\}\},$$

respectively. Suppose that the state machine is described by the following state-transition table.

q_1	q_0	x	q_1^+	q_0^+	y
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	0	1	0
0	1	1	1	1	1
1	0	0	1	0	1
1	0	1	0	1	0
1	1	0	1	1	1
1	1	1	0	0	0

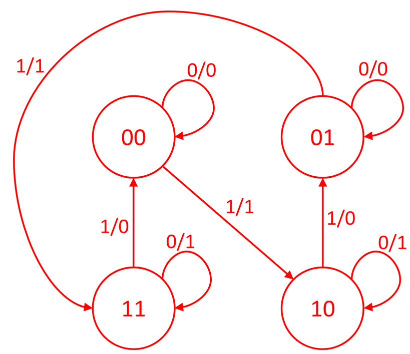
Table 1: The state-transition table for a Mealy finite state machine with one-dimensional inputs and outputs x and y, respectively, and a two-dimensional binary state $\boldsymbol{q}$.

Problem 3 (6 points)
Explore the connection between Mealy and Moore state machines in the following problems.

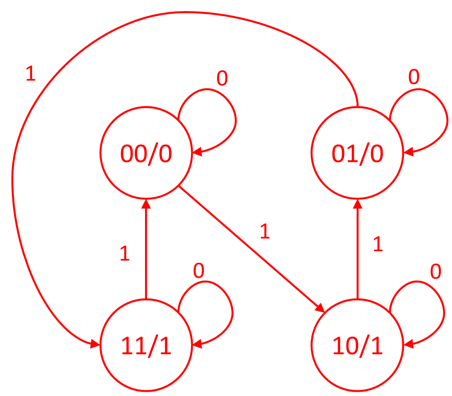
1. Draw a Mealy state machine diagram for the state-transition table given in Table 1.
2. Draw a diagram for an equivalent Moore state machine.
3. Draw a state-transition table for your Moore state machine in Part 2.

Solution 3

1. Mealy form diagram:



2. Moore form diagram:



3. State transition table for the Moore form (notice it is the same)

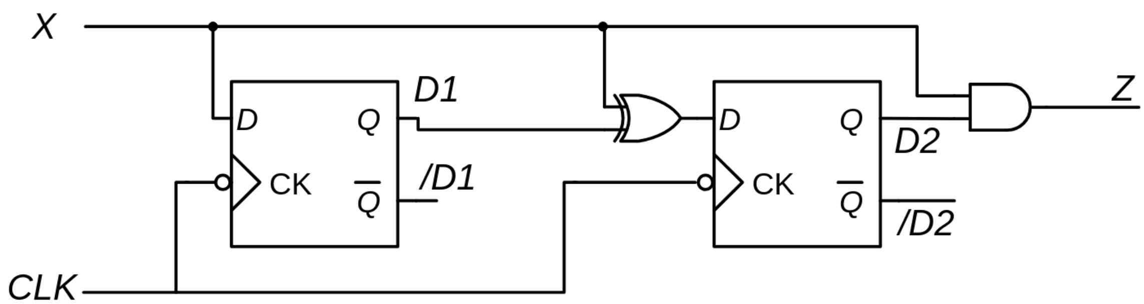
q_1	q_0	x	q_1^+	q_0^+	y
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	0	1	0
0	1	1	1	1	1
1	0	0	1	0	1
1	0	1	0	1	0
1	1	0	1	1	1
1	1	1	0	0	0

Problem 4: Flipping and Flopping

Problem 4 (2pts)
Consider the sequential logic circuit shown in Fig. 2 using *negative edge-triggered D* flip-flops. Derive logic expressions for D_1, D_2, and Z, and then sketch their waveforms shown below.

Solution 4

Note that $D_1 = X$, $D_2 = D_1 \oplus X$, and $Z = D_2 \cdot X$. Using these expressions, we sketch the result shown in Fig. 2.



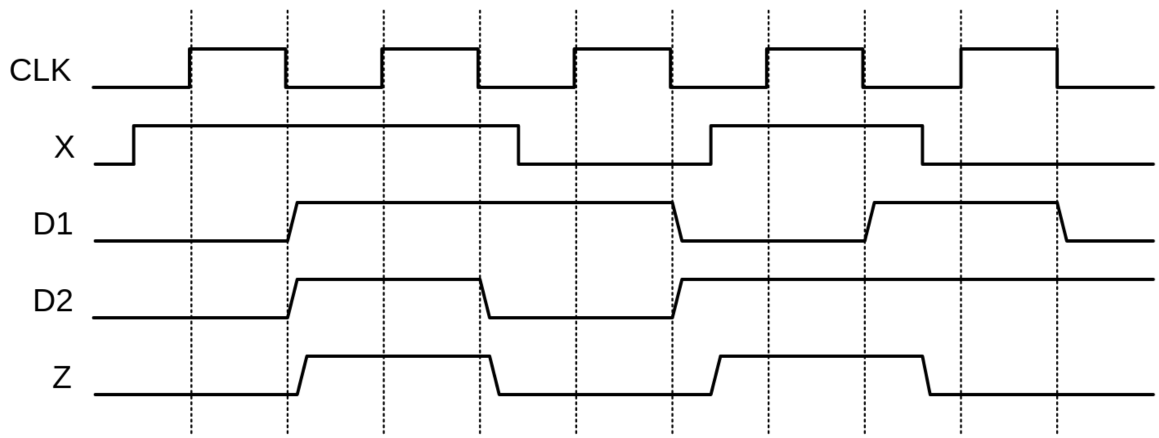


Figure 2: Solution: Timing diagram for the above sequential logic circuit with negative edge-triggered D flip-flops.

Problem 5: I can't believe it's not a truth table (jk)

current state	next step w/ input 0	next state w/ input 1	output
	$X=0$	$X=1$	
$Q(t)$	$Q(t+1)$	$Q(t+1)$	Z
A (initial)	B	D	0
B	C	B	0
C	B	A	1
D	B	C	0

Table 2: The state-transition table for a positive edge-triggered JK flip flop system.

Problem 5 (2pts + 2 bonus)

We can represent the state-transition Table 3 in binary, which we explore in this problem.

1. Suppose you want to design a state machine that implements the state-transition table shown in Table 3 using two positive edge-triggered JK flip-flops. Re-write the state-transition table shown in Table 3 in terms of binary definitions of the states: $A = (00)_2$, $B = (01)_2$, $C = (10)_2$, and $D = (11)_2$.

2. **[BONUS]:** Derive the state transition function and simplify the expressions for the data signals D_1, D_0 associated with the two JK flip flops using K-maps. Optionally sketch your sequential logic circuit design.

Solution 5

We begin by applying **compact encoding** to derive boolean expressions for each state bit.

current state	next step w/ input 0	next state w/ input 1	output
	$X=0$	$X=1$	
$(Q_1(t), Q_0(t))$	(Q_1^+, Q_0^+)	(Q_1^+, Q_0^+)	Z
$A/00$	$B/01$	$D/11$	0
$B/01$	$C/10$	$B/01$	0
$C/10$	$B/01$	$A/00$	1
$D/11$	$B/01$	$C/10$	0

Table 3: The state-transition table for a Mealy finite state machine with one-dimensional inputs and outputs x and y, respectively, and a two-dimensional binary state $\boldsymbol{q}$.

We now need to derive the simplified expressions for the evolutions of the state bits Q_1^+ and Q_0^+, and the output Z.

1. The state evolution Q_1^+ can be simplified with a Kmap as follows:

Q_1Q_0 \ X	0	1
00	0	1
01	1	0
11	0	1
10	0	0

   We can infer that: $\boxed{J_1 = Q_1^+ = \overline{X} \cdot \overline{Q}_1 \cdot Q_0 + X \cdot (\overline{Q}_1 \cdot \overline{Q}_0 + Q_1 \cdot Q_0)}$

2. The state evolution Q_0^+ can be simplified with a Kmap as follows:

Q_1Q_0 \ X	0	1
00	1	1
01	0	1
11	1	0
10	1	0

   We can infer that: $\boxed{J_0 = Q_0^+ = \overline{X} \cdot (Q_1 + \overline{Q}_0) + X \cdot \overline{Q}_1}$

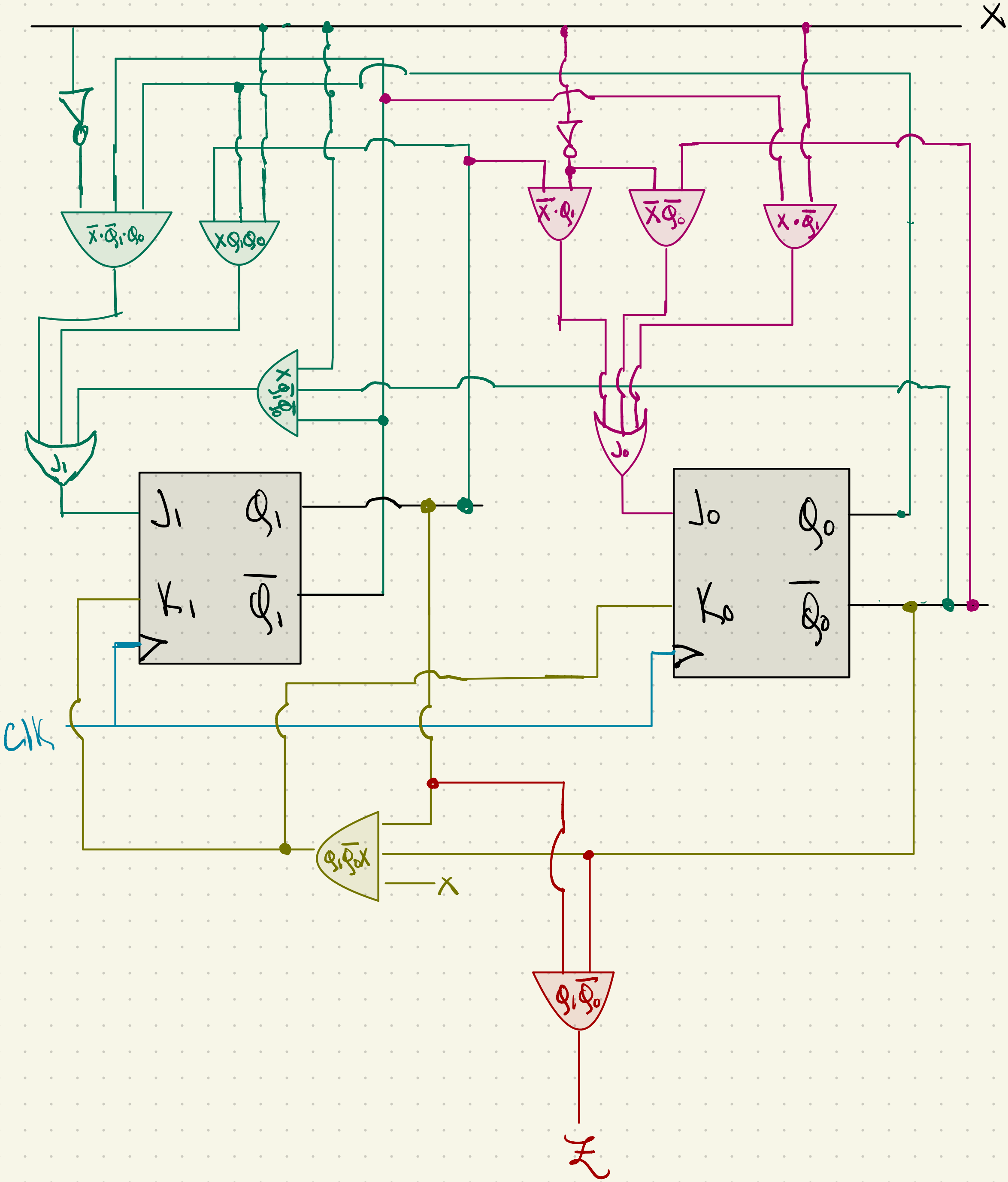
3. The output can be inferred directly from the table as $\boxed{Z = Q_1 \cdot \overline{Q}_0.}$

Note that the reset signals, K_1, K_0 need to be set so that we return to the initial state, A, which is encoded as $Q = (Q_1Q_0) = (00)$. Thus, we can set:

$$\boxed{K_1 = K_0 = Q_1 \cdot \overline{Q}_0 \cdot X}.$$

The circuit schematic is on the next page.

P5.



X
$\bar{X}\cdot\bar{Q}_1\cdot Q_0$
XQ_1Q_0
$\bar{X}\cdot Q_1$
$\bar{X}\bar{Q}_0$
$X\cdot\bar{Q}_1$
$XQ_1\bar{Q}_0$
J_1
J_0
J_1
Q_1
K_1
$\bar{Q}_1$
J_0
Q_0
K_0
$\bar{Q}_0$
Clk
$Q_1\bar{Q}_0X$
X
$Q_1\bar{Q}_0$
Z

Problem 6: Ups and Downs

Consider the prediction device from Problem 1 of Exam 2. Suppose that you want to upgrade the device to set the tie-breaking signal B by *counting the number of times* that each predictor P_k has been right in the past.

This new part of the circuit maintains a *count* Q_k for each $k \in \{0, 1, 2, 3\}$. The circuit has *two inputs* for the stock that are *functions of time*:

1. $U(t)$, UP signal: The price of the stock went *up* at the clock cycle t.
2. $D(t)$, DOWN signal: The price of the stock went *down* at the clock cycle t.

The circuit needs to meet the following requirements for each predictor k:

1. The circuit does not count, that is, $Q_k(t) = Q_k(t-1)$ if $D(t) = U(t) = 1$ or $D(t) = U(t) = 0$
2. The circuit should count *up* for k if predictor k *was right* at the last clock cycle, that is, $Q_k(t+1) = Q_k(t) + 1$, if the following is true: ($U(t) = 1$ AND $D(t) = 0$ AND $P_k(t) = 1$) OR ($D(t) = 0$ AND $U(t) = 0$ AND $P_k(t) = 0$).
3. The circuit should count *down* for k if predictor k *was wrong* at the last clock cycle, that is, $Q_k(t+1) = Q_k(t) - 1$, if the following is true: ($D(t) = 1$ AND $U(t) = 0$ AND $P_k(t) = 1$) OR ($U(t) = 1$ AND $D(t) = 0$ AND $P_k(t) = 0$).

We will make the following modeling **assumptions:**

1. Assume that the circuit can simply be copied for each predictor k, so we can design the circuit for a single predictor k. Therefore, drop the notation subscript k for the rest of the problem; that is, we will write the signals as $P_k(t) \stackrel{\text{def}}{=} P(t)$, and $Q_k(t) \stackrel{\text{def}}{=} Q(t)$ for convenience.
2. Assume that the circuit can only remember the past 4 trading cycles, so all numbers can be represented as 2-bit binary numbers.

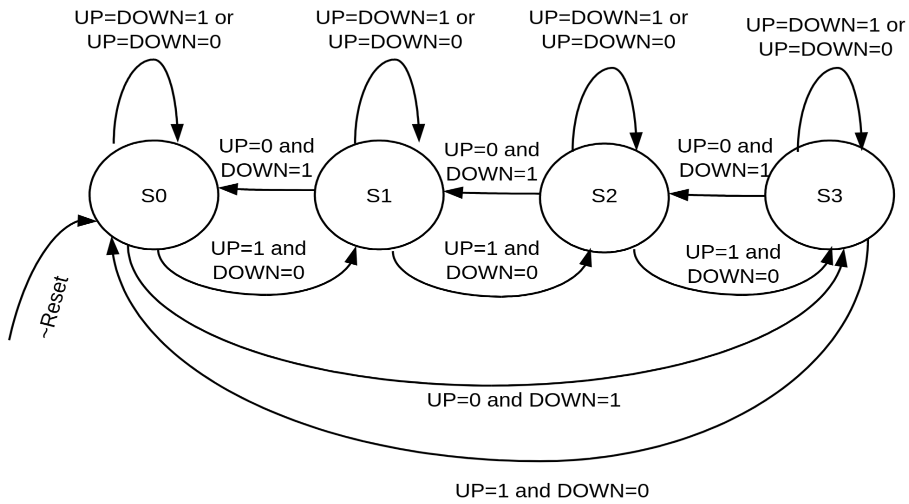
Problem 6 (6pts + 2 bonus)

Design an electronic counter for tracking the performance of your predictors.

1. **Part 1:** Suppose that $P(t) = 1$ for all time, (i.e., the predictor is always bullish). Therefore, calculating $C(t)$ is equivalent to simply summing the UPs and subtracting the DOWNs.
   (a) Draw a state diagram of this machine. Let the initial state be $Q(0) = (00)_2$.
   (b) Using two positive edge-triggered D flip-flops, implement the circuit, *showing all steps*. You are not required to draw out the circuit, just come up with equations for the flip-flop inputs. *Hint: Use a state transition table, and two K-maps*
   (c) Sketch the timing diagram shown below in Fig. 3.
2. **Part 2 [BONUS]:** Assume that $P(t)$ can vary across time. Propose a revision of your design in Part 1 that meets the requirements of the circuit in the problem statement.

Solution 6

1. The state diagram is given as follows:



Note–I incorrectly said on Thursday 11/14's office hours that the counter should maintain S_3 if $U = 1$ and $D = 0$ is the input and that the counter should maintain S_0 if $U = 0$ and $D = 0$. Please note that the counter should reset in this case. If you followed my instructions that day, you won't lose credit. Please note the correct diagram.

2. Because we have 4 states, we need a **minimum of two state bits, and hence two flip flops,** to represent the system. **(Note, this means we are using compact encoding. If we used one-hot encoding, we would need 4 flip flops and 4 state bits.)** Let's derive the state transition table:

Current state	Next state			
	$(UD) = (00)$	$(UD) = (01)$	$(UD) = (10)$	$(UD) = (11)$
Symbol/Encoded	Symbol/Encoded	Symbol/Encoded	Symbol/Encoded	Symbol/Encoded
$S_0/00$	$S_0/00$	$S_3/11$	$S_1/01$	$S_0/00$
$S_1/01$	$S_1/01$	$S_0/00$	$S_2/10$	$S_1/01$
$S_2/10$	$S_2/10$	$S_1/01$	$S_3/11$	$S_2/10$
$S_3/11$	$S_3/11$	$S_2/10$	$S_0/00$	$S_3/11$
$(Q_1(t), Q_0(t))$	(Q_1^+, Q_0^+)	(Q_1^+, Q_0^+)	(Q_1^+, Q_0^+)	(Q_1^+, Q_0^+)

(a) The Kmap for Q_1^+ is given as follows:

Q_1Q_0 \ UD	00	01	11	10
00	0	1	0	0
01	0	0	0	1
11	1	1	1	0
10	1	0	1	1

$$\boxed{Q_1^+ = \overline{U} \cdot \overline{D} \cdot Q_1 + D \cdot Q_1 \cdot Q_0 + U \cdot Q_1 \cdot \overline{Q}_0 + \overline{U} \cdot D \cdot \overline{Q}_1 \cdot \overline{Q}_0 + U \cdot \overline{D} \cdot \overline{Q}_1 \cdot Q_0}$$

(b) The Kamp for Q_0^+ is given as follows:

Q_1Q_0 \ UD	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	1	0	1	0
10	0	1	0	1

$$\boxed{Q_0^+ = \overline{U} \cdot \overline{D} \cdot Q_0 + U \cdot D \cdot Q_0 + \overline{U} \cdot D \cdot \overline{Q}_0 + U \cdot \overline{D} \cdot \overline{Q}_0}$$

3. The timing diagram is shown below in Fig. 3

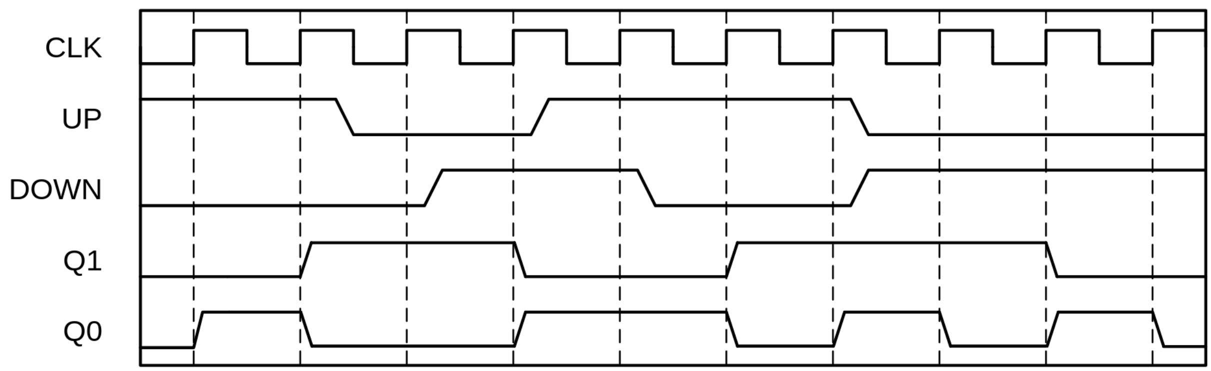


Figure 3: Solution: The sketched waveform for the prediction accuracy counter