

Exam 2 Review

1. Memory Components

State Transition Table

D Latch Validation: Truth Table

D T-F $\rightarrow$ Timing vs Latch

Memory hierarchy

2. Finite State Machines

Behavior $\rightarrow$ State Diagram Find the states needed for the operations

State Table $\rightarrow$ Excitation Table $\rightarrow$ KMap $\rightarrow$ Logic

Mealy vs Moore machines

State vs output Timing of Difference on the Transition

3. Timing & Retiming

Longest Paths

Deadlocks

Shortest Paths

bounds

lower bound

$\leq$ upper bound

Skew

$\rightarrow$ Precision

Retiming

Linear inequality expressions

$\rightarrow$ Find the opt. solutions.

...total available

...total available

...total available

...total available

...total available

...total available

...total available

...total available

...total available

...total available

...total available

...total available

...total available

CSE140 - Spring 2019

Midterm 2 - Solution

Section A

Tuesday May 14, 2019

Name: _____

PID: _____

Name of person to your left: _____

Name of person to your right: _____

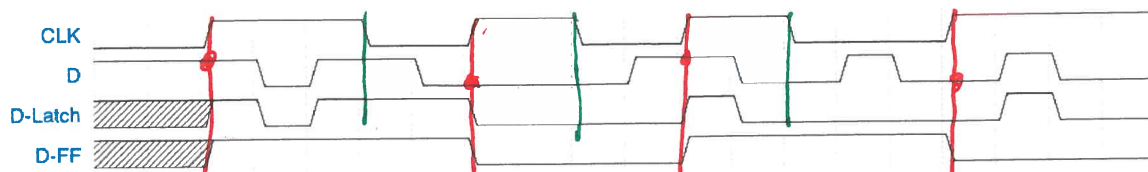
Please read the following instructions carefully:

1. This exam has **5 problems** and **8 pages**.
2. **Write your name and student ID on every page.** Write the names of the students to your left and right in the space provided on this page.
3. **Write your answers in designated areas only.** You can use the rest of the page as scratch paper.
4. If you need more space to finish a problem, ask a proctor for paper. **Do not use your own paper.**
5. **Do not detach any of the papers.**
6. If you are unclear about any of the questions on the exam, make the most plausible assumption to answer the question. Instructors and proctors will not answer questions on the exam material.

All the Best!

1 Timing Diagram of Latch and Flip-Flop (10 Points)

Given the input waveforms shown below, sketch the output waveforms of a **D-latch** and a **D-flip-flop**.



Rubrics:

- 1 point for error(s) in each interval.
- 2 points for swapping D-FF and D-Latch outputs.
- 5 points for correct D-FF/D-Latch but wrong D-Latch/D-FF.

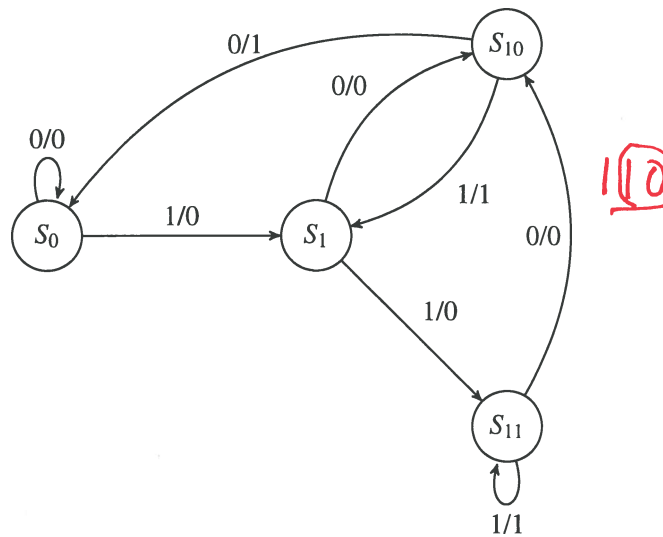
2 Finite State Machine (30 Points)

A pattern recognizer takes a string of binary bits $x(t)$ as input and outputs $y(t) = 1$ whenever $x(t-2, t) = 111, 101$, or 100 and $y(t) = 0$ otherwise. Design a **Mealy** state machine for the pattern recognizer that has **no more than four states**.

A. Draw the state diagram in the box below. We have drawn the starting state, S_0 , for you. (12 Points)

Rubrics:

- 2 points for each incorrect transition.
- Adjustable deduction on other minor mistakes.



B. Fill the state table of the state machine. (6 Points)

Rubrics:

- 1 point for each mistake.
- 3 points for missing output.

State Table

Present State s^t	Next State, Output	
	$x(t) = 0$	$x(t) = 1$
s_0	$s_0, 0$	$s_1, 0$
s_1	$s_{10}, 0$	$s_{11}, 0$
s_{10}	$s_0, 1$	$s_1, 1$
s_{11}	$s_{10}, 0$	$s_{11}, 1$

s^{t+1}, y

Design the system with **two** D flip-flops. Describe the output $y(t)$ and flip-flop inputs $D_1(t)$ and $D_0(t)$ as functions of the input $x(t)$ and flip-flop outputs $Q_1(t)$ and $Q_0(t)$ in **minimal sum-of-products** expressions. No need to draw the logic diagram.

C. Write the **excitation table** in the box below. (6 Points)

Rubrics:

- 1 point for each mistake.
- 2 points for missing column/row.

State Name	Binary Encoding (Q_1Q_0)
s_0	00
s_1	01
s_{10}	10
s_{11}	11

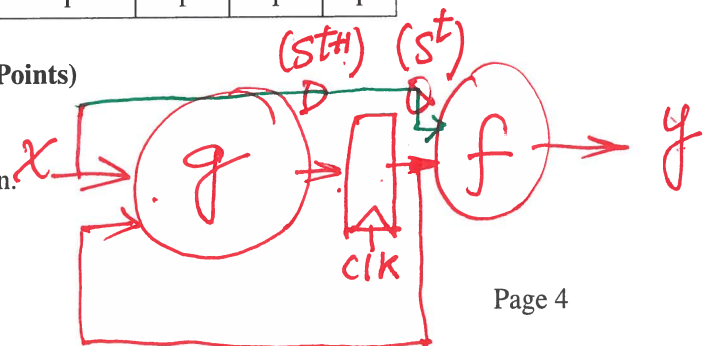
Excitation Table

id	$Q_1(t)$	$Q_0(t)$	$x(t)$	$Q_1(t+1)$	$Q_0(t+1)$	$D_1(t)$	$D_0(t)$	$y(t)$
0	0	0	0	0	0	0	0	0
1	0	0	1	0	1	0	1	0
2	0	1	0	1	0	1	0	0
3	0	1	1	1	1	1	1	0
4	1	0	0	0	0	0	0	1
5	1	0	1	0	1	0	1	1
6	1	1	0	1	0	1	0	0
7	1	1	1	1	1	1	1	1

D. Write the **Boolean expressions** in the box below. (6 Points)

Rubrics:

- 2 points for each wrong or non-minimal expression.



−3 points for not using don't cares.

Here is the K-map for $D_1(t)$, which gives: $D_1(t) = Q_0(t)$.

		$Q_0(t)x(t)$			
		00	01	11	10
$Q_1(t)$	0	0	0	1	1
	1	0	0	1	1

Here is the K-map for $D_0(t)$, which gives: $D_0(t) = x(t)$.

		$Q_0(t)x(t)$			
		00	01	11	10
$Q_1(t)$	0	0	1	1	0
	1	0	1	1	0

And here is the K-map for $y(t)$, which gives: $y(t) = Q_1(t) \cdot Q_0'(t) + Q_1(t)x(t)$.

		$Q_0(t)x(t)$			
		00	01	11	10
$Q_1(t)$	0	0	0	0	0
	1	1	1	1	0

3 Mealy and Moore Machine Conversion (25 Points)

Given the state table below for a Mealy state machine with input x and output y ,

Present State	Next State, Output	
	$x(t) = 0$	$x(t) = 1$
S0	S0,0	S1,0
S1	S1,0	S2,0
S2	S1,0	S3,0
S3	S0,1	S4,0
S4	S0,0	S1,1

$S0,0$
 $S0,1 \rightarrow S5$
 $S1,0$
 $S1,1 \rightarrow S6$
 $S2,0$
 $S3,0$
 $S4,0$
 $S4,1$
 $S5,1$

A. Fill the following table as the machine starts at state S0 and process through the given input sequence. (5 Points)

Rubric: -1 point for each mistake

CLK	1	2	3	4	5	6	7	8	9	10
x	0	1	1	1	0	1	1	1	1	0
state	S0	S0	S1	S2	S3	S0	S1	S2	S3	S4
y	0	0	0	0	1	0	0	0	0	0

B. Convert the above machine into a Moore machine. Write the **state table**. (15 Points)

Rubrics:

-10 points if answer is not a Moore machine (at least one state with different outputs). No deduction for incorrect outputs.

-2 points for drawing a state diagram instead of a state table.

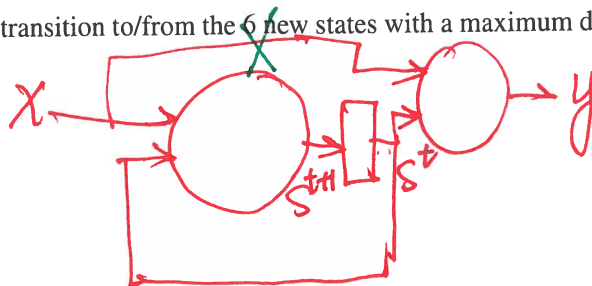
-2 points for ambiguous state encoding.

-2 points for each missing/extra state with a maximum deduction of -3 points.

-1 point for each incorrect output with a maximum deduction of -3 points.

-1 point for each incorrect transition to/from the 8 original states with a maximum deduction of -3 points.

-1 point for each incorrect transition to/from the 6 new states with a maximum deduction of -6 points.



Present State	Next State		Output
	$x(t) = 0$	$x(t) = 1$	
S00	S00	S10	0
S01	S00	S10	1
S10	S10	S2	0
S11	S10	S2	1
S2	S10	S3	0
S3	S01	S4	0
S4	S00	S11	0

Another correct answer with different state names:

Present State	Next State		Output
	$x(t) = 0$	$x(t) = 1$	
S0	S0	S1	0
S1	S1	S2	0
S2	S1	S3	0
S3	S5	S4	0
S4	S0	S6	0
S5	S0	S1	1
S6	S1	S2	1

C. Use the Moore machine to fill the table starting at state S0 and process through the given input sequence.
(5 Points)

Rubric: –1 point for each mistake

CLK	1	2	3	4	5	6	7	8	9	10
x	0	1	1	1	0	1	1	1	1	0
state	S00/S01	S00	S10	S2	S3	S01	S10	S2	S3	S4
y	0/1	0	0	0	0	1	0	0	0	0

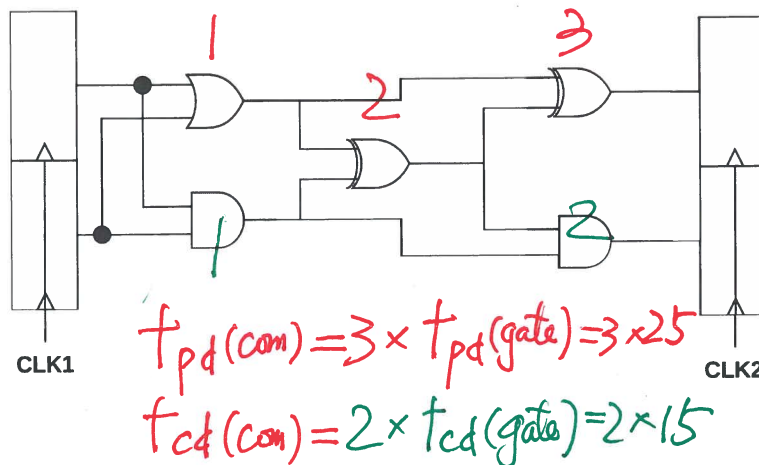
Another correct answer with different state names:

CLK	1	2	3	4	5	6	7	8	9	10
x	0	1	1	1	0	1	1	1	1	0
state	S0/S5	S0	S1	S2	S3	S5	S1	S2	S3	S4
y	0/1	0	0	0	0	1	0	0	0	0

4 Timing and Retiming (15 Points)

For the following circuit, the timing characteristics of the components are summarized below.

- Flip-flops: clock-to-Q maximum delay (propagation delay) $t_{pcq} = 30ps$, clock-to-Q minimum delay (contamination delay) $t_{ccq} = 20ps$, setup time $t_{setup} = 20ps$, and hold time $t_{hold} = 35ps$.
- Logic gates (each AND, OR, XOR): propagation delay $t_{pd}(gate) = 25ps$ and contamination delay $t_{cd}(gate) = 15ps$.



A. Derive the minimal clock cycle time T . Suppose that there is **no clock skew**. (7.5 Points)

Rubrics:

- 2.5 points for incorrect longest path.
- 2.5 points for incorrect formula.
- 2.5 points for incorrect final answer.

To find the maximum frequency, we need the minimum clock period that does not violate the setup time constraint of any flip-flop. So we need to identify the longest path of the circuit. Since all gates have the same propagation delay, the longest path is the one with the most number of gates. Any of the paths going through the XOR gate in the middle is one such path and has three gates, so its delay is $3 \times t_{pd}(gate) = 3 \times 25ps = 75ps$. We can then write the setup time constraint to find the minimum clock period.

$$T > \overset{(con)}{t_{pd}} + t_{setup} + t_{pcq} \quad (1)$$

$$T > 75ps + 20ps + 30ps \quad (2)$$

$$T > 125ps \quad (3)$$

So $T_{min} = 125ps$ and $F_{max} = \frac{1}{T_{min}} = 8GHz$.

B. Derive the minimal clock cycle time T using a **designated clock skew**. (7.5 Points)

Rubrics:

- 2.5 points for incorrect shortest path.
- 2.5 points for incorrect formula.
- 2.5 points for incorrect final answer and incorrect clock skew.
- 1.5 points for incorrect final answer and correct clock skew.

For retiming, the setup time constraint and minimum clock period are

$$T > t_{pd} + t_{setup} + t_{pcq} - t_{skew} \quad \text{longest Paths} \quad (4)$$

$$T_{min} = t_{pd} + t_{setup} + t_{pcq} - t_{skew} \quad (5)$$

So, to minimize the clock period T , clock skew t_{skew} should be minimized. Then we need to find the minimum clock skew that would not violate the hold time constraint. For that, we need to find the shortest path of the circuit. Any of the paths that does not go through the XOR gate in the middle is the shortest path with two gates and a contamination delay of $2 \times t_{cd}(\text{gate}) = 2 \times 15ps = 30ps$. So we can write the hold time constraint

$$t_{hold} + t_{skew} < t_{cd} + t_{ccq} \quad \text{shortest paths} \quad (6)$$

$$35 + t_{skew} < 30 + 20 \quad (7)$$

$$t_{skew} < 15ps \quad (8)$$

Substituting in Equation 5 gives

$$T_{min} = 75ps + 20ps + 30ps - 15ps \quad (9)$$

$$T_{min} = 110ps \quad (10)$$

$$\text{So } F_{max} = \frac{1}{T_{min}} = 9.1GHz.$$

$$\text{longest path } 125ps \leq T - t_{skew} \quad \textcircled{1}$$

$$t_{hold} + t_{skew} \leq 50$$

$$35 + t_{skew} \leq 50 \quad \textcircled{2}$$

$$\textcircled{1} \quad 125 \leq T + t_{skew}$$

$$\textcircled{2} \quad t_{skew} \leq 15$$

Goal min T

$$\Rightarrow t_{skew} = 15$$

$$\Rightarrow T = 125 - t_{skew} = 110$$

5 Decoders and Multiplexers (20 Points)

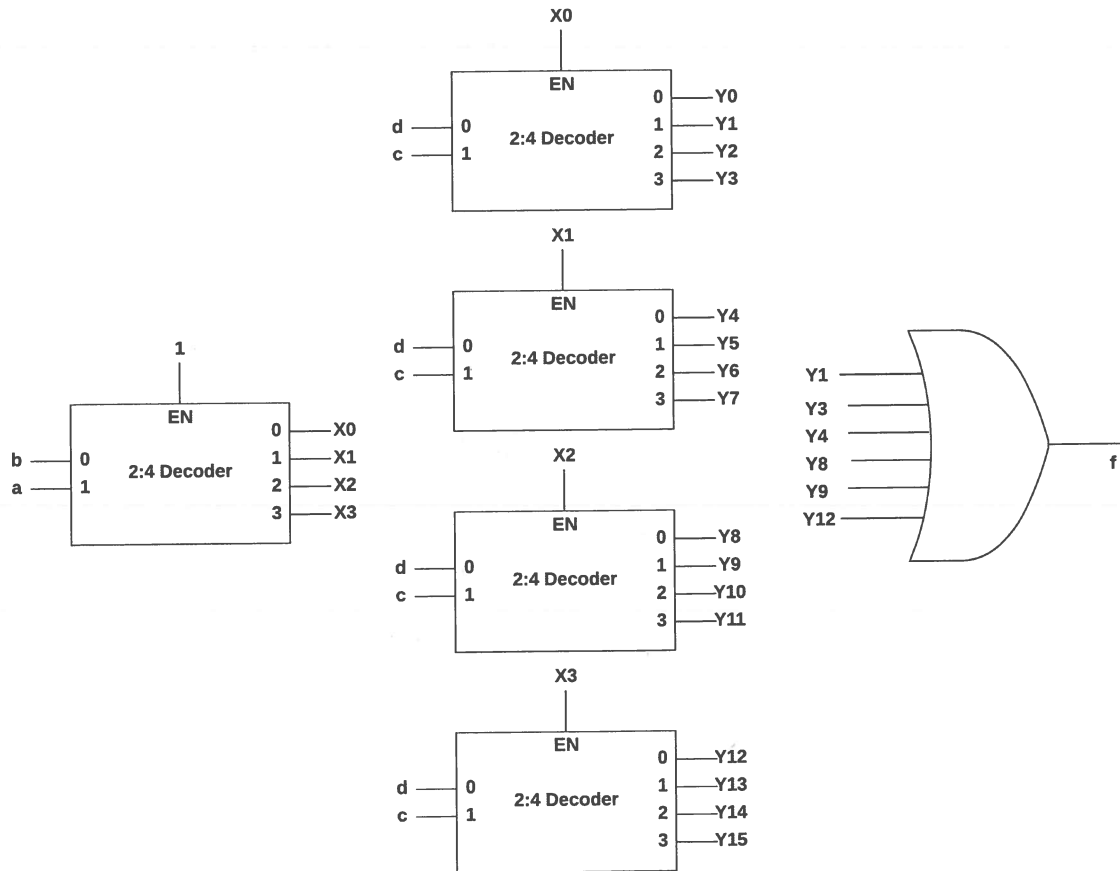
Assume a dual-railed system, where you have access to any variable and its complement. Given a four-input Boolean function

$$f(a, b, c, d) = \sum m(1, 3, 4, 8, 9, 12) + \sum d(0, 11, 14).$$

A. Implement the function using a minimal network of **2:4 decoders** and OR gates. (10 Points)

Rubrics:

- 8 for using incorrect decoders
- 1 for incorrect/missing EN input/label
- 1 for incorrect/missing indexing of inputs/outputs (inside/outside decoder)
- 2 for incorrect/missing inputs to OR gate



2. Implement the function using a minimal network of **4:1 multiplexers**. (10 Points)

Rubrics:

- 0.75 for no/wrong selector/input label on muxes
- 3 for using gates
- 2 for using 2:1 mux instead of 4:1 mux
- 2 for using more than the 2 muxes
- 1 for each wrong input

<i>a</i>	<i>b</i>	<i>cd</i>				<i>D(c,d)</i>
		00	01	10	11	
0	0	X	1	0	1	<i>D</i> ₀
0	1	1	0	0	0	<i>D</i> ₁
1	0	1	1	0	X	<i>D</i> ₂
1	1	1	0	X	0	<i>D</i> ₃

Using the K-map below, we can write $D_0 = d$.

		<i>d</i>	
		0	1
<i>c</i>	0	X	1
	1	0	1

Using the K-map below, we can write $D_1 = c'd'$.

		<i>d</i>	
		0	1
<i>c</i>	0	1	0
	1	0	0

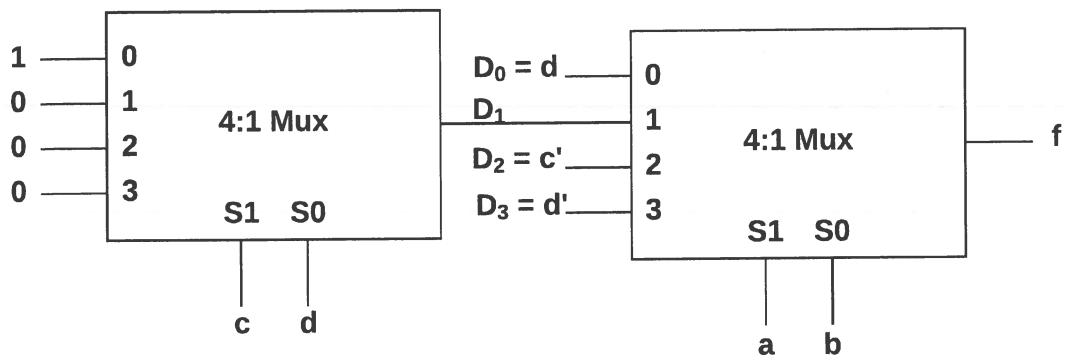
Using the K-map below, we can write $D_2 = c'$.

		d	
		0	1
c	0	1	1
	1	0	X

Using the K-map below, we can write $D_3 = d'$.

		d	
		0	1
c	0	1	0
	1	X	0

So we need another mux for D_1 but we do not need to write the truth table to implement a 2-variable function using a 4:1 mux. Here is the resulting logic diagram.



CSE140 - Spring 2019

Midterm 2 - Solution

Section B

Tuesday May 14, 2019

Name: _____

PID: _____

Name of person to your left: _____

Name of person to your right: _____

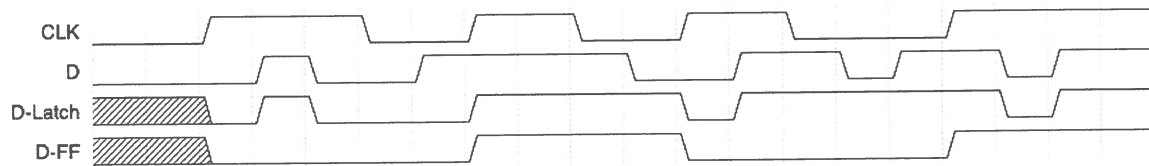
Please read the following instructions carefully:

1. This exam has **5 problems** and **8 pages**.
2. **Write your name and student ID on every page.** Write the names of the students to your left and right in the space provided on this page.
3. **Write your answers in designated areas only.** You can use the rest of the page as scratch paper.
4. If you need more space to finish a problem, ask a proctor for paper. **Do not use your own paper.**
5. **Do not detach any of the papers.**
6. If you are unclear about any of the questions on the exam, make the most plausible assumption to answer the question. Instructors and proctors will not answer questions on the exam material.

All the Best!

1 Timing Diagram of Latch and Flip-Flop (10 Points)

Given the input waveforms shown below, sketch the output waveforms of a **D-latch** and a **D-flip-flop**.



Rubrics:

- 1 point for error(s) in each interval.
- 2 points for swapping D-FF and D-Latch outputs.
- 5 points for correct D-FF/D-Latch but wrong D-Latch/D-FF.

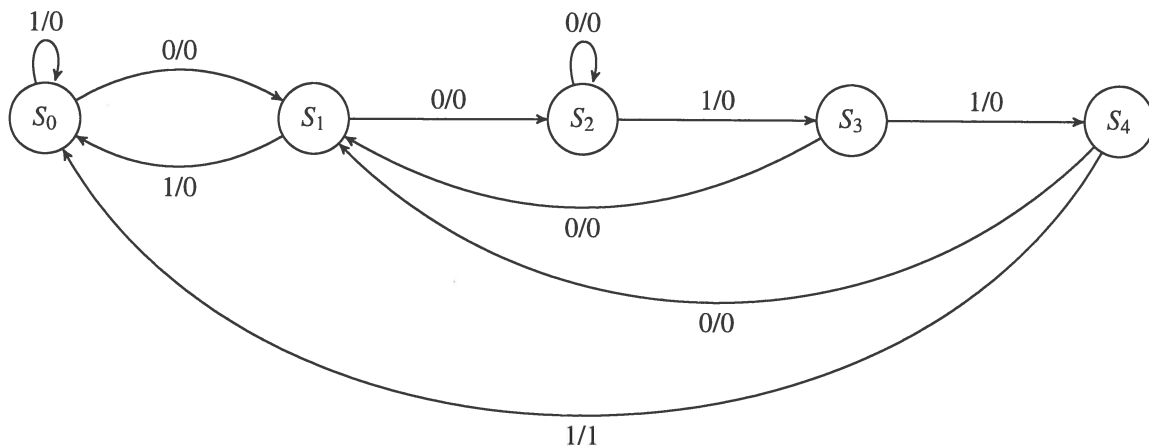
2 Finite State Machine (30 Points)

A pattern recognizer takes a string of binary bits $x(t)$ as input and outputs $y(t) = 1$ whenever $x(t-4, t) = 00111$ and $y(t) = 0$ otherwise. Design a **Mealy** state machine for the pattern recognizer that has **no more than five states**.

A. Draw the **state diagram** in the box below. We have drawn the starting state, S_0 , for you. (12 Points)

Rubrics:

- 2 points for each incorrect transition.
- Adjustable deduction on other minor mistakes.



B. Fill the **state table** of the state machine. (6 Points)

Rubrics:

- 1 point for each mistake.
- 3 points for missing output.

Present State	Next State, Output	
	$x(t) = 0$	$x(t) = 1$
S_0	$S_1, 0$	$S_0, 0$
S_1	$S_2, 0$	$S_0, 0$
S_2	$S_2, 0$	$S_3, 0$
S_3	$S_1, 0$	$S_4, 0$
S_4	$S_1, 0$	$S_0, 1$

Design the system with **three** D flip-flops. Describe the output $y(t)$ and flip-flop inputs $D_2(t)$, $D_1(t)$, and $D_0(t)$ as functions of the input $x(t)$ and flip-flop outputs $Q_2(t)$, $Q_1(t)$, and $Q_0(t)$ in **minimal sum-of-products** expressions. No need to draw the logic diagram.

C. Write the **excitation table** in the box below. (6 Points)

Rubrics:

- 1 point for each mistake.
- 2 points for missing column/row.

State Name	Binary Encoding (Q_1Q_0)
S_0	000
S_1	001
S_2	010
S_3	011
S_4	100

id	$Q_2(t)$	$Q_1(t)$	$Q_0(t)$	$x(t)$	$Q_2(t+1)$	$Q_1(t+1)$	$Q_0(t+1)$	$D_2(t)$	$D_1(t)$	$D_0(t)$	$y(t)$
0	0	0	0	0	0	0	1	0	0	1	0
1	0	0	0	1	0	0	0	0	0	0	0
2	0	0	1	0	0	1	0	0	1	0	0
3	0	0	1	1	0	0	0	0	0	0	0
4	0	1	0	0	0	1	0	0	1	0	0
5	0	1	0	1	0	1	1	0	1	1	0
6	0	1	1	0	0	0	1	0	0	1	0
7	0	1	1	1	1	0	0	1	0	0	0
8	1	0	0	0	0	0	1	0	0	1	0
9	1	0	0	1	0	0	0	0	0	0	1
10	1	0	1	0	X	X	X	X	X	X	X
11	1	0	1	1	X	X	X	X	X	X	X
12	1	1	0	0	X	X	X	X	X	X	X
13	1	1	0	1	X	X	X	X	X	X	X
14	1	1	1	0	X	X	X	X	X	X	X
15	1	1	1	1	X	X	X	X	X	X	X

D. Write the **Boolean expressions** in the box below. (6 Points)

Rubric: -1.5 points for each wrong or non-minimal answer.

Here is the K-map for $D_2(t)$, which gives: $D_2(t) = Q_1(t)Q_0(t)x(t)$.

		$Q_0(t)x(t)$			
		00	01	11	10
$Q_2(t)Q_1(t)$	00	0	0	0	0
	01	0	0	1	0
	11	X	X	X	X
	10	0	0	X	X

Here is the K-map for $D_1(t)$, which gives: $D_1(t) = Q_1(t)Q'_0(t) + Q'_1(t)Q_0(t)x'(t)$.

		$Q_0(t)x(t)$			
		00	01	11	10
$Q_2(t)Q_1(t)$	00	0	0	0	1
	01	1	1	0	0
	11	X	X	X	X
	10	0	0	X	X

Here is the K-map for $D_0(t)$, which gives: $D_0(t) = Q_1'(t)Q_0'(t)x'(t) + Q_1(t)Q_0(t)x'(t) + Q_1(t)Q_0'(t)x(t)$.

		$Q_0(t)x(t)$			
		00	01	11	10
$Q_2(t)Q_1(t)$	00	1	0	0	0
	01	0	1	0	1
	11	X	X	X	X
	10	1	0	X	X

And here is the K-map for $y(t)$, which gives: $y(t) = Q_2(t)x(t)$.

		$Q_0(t)x(t)$			
		00	01	11	10
$Q_2(t)Q_1(t)$	00	0	0	0	0
	01	0	0	0	0
	11	X	X	X	X
	10	0	1	X	X

3 Mealy and Moore Machine Conversion (25 Points)

Given the state table below for a Mealy state machine with input x and output y ,

Present State	Next State, Output	
	$x(t) = 0$	$x(t) = 1$
S0	S0,0	S2,1
S1	S1,0	S3,0
S2	S1,0	S2,0
S3	S0,0	S4,1
S4	S0,1	S3,0

A. Fill the following table as the machine starts at state S0 and process through the given input sequence. (5 Points)

Rubric: –1 point for each mistake

CLK	1	2	3	4	5	6	7	8	9	10
x	0	1	1	1	0	1	1	0	1	0
state	S0	S0	S2	S2	S2	S1	S3	S4	S0	S2
y	0	1	0	0	0	0	1	1	1	0

B. Convert the above machine into a Moore machine. Write the **state table**. (15 Points)

Rubrics:

–10 points if answer is not a Moore machine (at least one state with different outputs). No deduction for incorrect outputs.

–2 points for drawing a state diagram instead of a state table.

–2 points for ambiguous state encoding.

–2 points for each missing/extra state with a maximum deduction of –3 points.

–1 point for each incorrect output with a maximum deduction of –3 points.

–1 point for each incorrect transition to/from the 8 original states with a maximum deduction of –3 points.

–1 point for each incorrect transition to/from the 6 new states with a maximum deduction of –6 points.

Present State	Next State		Output
	$x(t) = 0$	$x(t) = 1$	
S00	S00	S21	0
S01	S00	S21	1
S1	S1	S3	0
S20	S1	S20	0
S21	S1	S20	1
S3	S00	S4	0
S4	S01	S3	1

Another correct answer with different state names:

Present State	Next State		Output
	$x(t) = 0$	$x(t) = 1$	
S0	S0	S6	0
S1	S1	S3	0
S2	S1	S2	0
S3	S0	S4	0
S4	S5	S3	1
S5	S0	S6	1
S6	S1	S2	1

C. Use the Moore machine to fill the table starting at state S0 and process through the given input sequence. (5 Points)

Rubric: -1 point for each mistake

CLK	1	2	3	4	5	6	7	8	9	10
x	0	1	1	1	0	1	1	0	1	0
state	S00/S01	S00	S21	S20	S20	S1	S3	S4	S01	S21
y	0/1	0	1	0	0	0	0	1	1	1

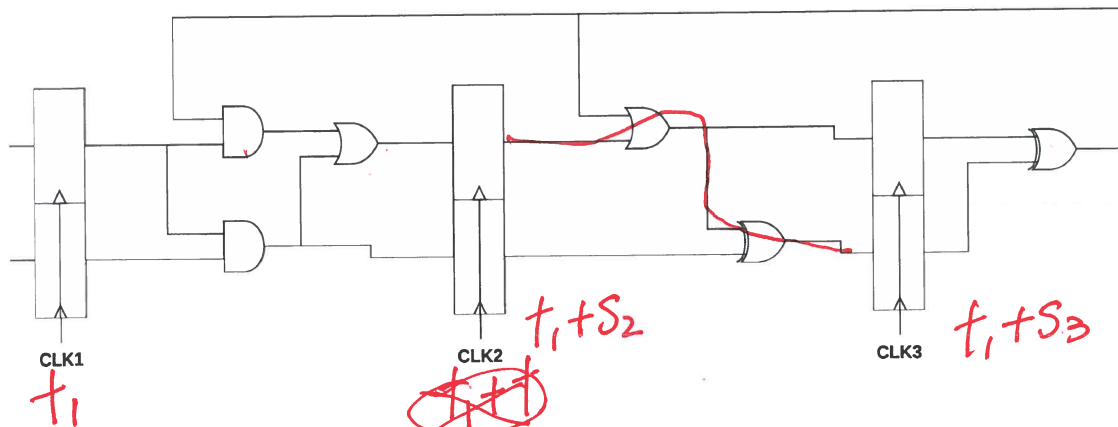
Another correct answer with different state names:

CLK	1	2	3	4	5	6	7	8	9	10
x	0	1	1	1	0	1	1	0	1	0
state	S0/S5	S0	S6	S2	S2	S1	S3	S4	S5	S6
y	0/1	0	1	0	0	0	0	1	1	1

4 Timing and Retiming (15 Points)

For the following circuit, the timing characteristics of the components are summarized below.

- Flip-flops: clock-to-Q maximum delay (propagation delay) $t_{pcq} = 30ps$, clock-to-Q minimum delay (contamination delay) $t_{ccq} = 20ps$, setup time $t_{setup} = 20ps$, and hold time $t_{hold} = 25ps$.
- Logic gates (each AND, OR, XOR): propagation delay $t_{pd}(gate) = 25ps$ and contamination delay $t_{cd}(gate) = 15ps$.



A. Derive the minimal clock cycle time T . Suppose that there is **no clock skew**. (7.5 Points)

Rubrics:

- 2.5 points for incorrect longest path.
- 2.5 points for incorrect formula.
- 2.5 points for incorrect final answer.

clk1 → 2	$t_{pd}(com)$	$t_{cd}(com)$
2 → 3	2	1
3 → 2	3	3
3 → 3	~~2~~ 3	2

To find the maximum frequency, we need the minimum clock period that does not violate the setup time constraint of any flip-flop. So we need to identify the longest path of the circuit. Since all gates have the same propagation delay, the longest path is the one with the most number of gates. The paths from the third column of flip-flops to the second column are the longest with three gates and a delay of $3 \times t_{pd}(gate) = 3 \times 25ps = 75ps$. We can then write the setup time constraint to find the minimum clock period.

$$\begin{aligned}
 T &> 3 \times t_{pd}(gate) + t_{pcq} & (1) \\
 T &> 75ps + 20ps + 30ps & (2) \\
 T &> 125ps & (3)
 \end{aligned}$$

So $T_{min} = 125ps$ and $F_{max} = \frac{1}{T_{min}} = 8GHz$.

B. How much clock skew t_{skew} can the circuit tolerate before it experiences a hold time violation? (7.5 Points)

Rubrics:

- 2.5 points for incorrect shortest path.
- 2.5 points for incorrect formula.
- 2.5 points for incorrect final answer.

To write the hold time constraint, we need to identify the shortest path of the circuit. The paths from the first column of flip-flops to the second column and from the second column to the third that go through only one gate are the shortest paths and have a contamination delay of $1 \times t_{cd}(\text{gate}) = 15\text{ps}$. The hold time constraint is

$$t_{\text{hold}} + t_{\text{skew}} < t_{\text{cd}} + t_{\text{ccq}} \quad (4)$$

$$25 + t_{\text{skew}} < 15 + 20 \quad (5)$$

$$t_{\text{skew}} < 10\text{ps} \quad (6)$$

So the circuit can tolerate a maximum clock skew of 10ps .

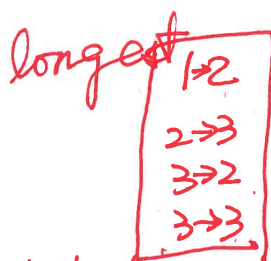
$$t_{\text{hold}} < t_{\text{cd}}(\text{com}) + t_{\text{ccq}}$$

$$25 < 1 \times t_{\text{cd}}(\text{gate}) + t_{\text{ccq}}$$

$$25 < 15 + 20$$

$$+ t_{\text{skew}}$$

$$t_{\text{hold}} + t_{\text{skew}} < \text{shortest path}$$



$$\leq T + \text{skew}_{2 \rightarrow 1} = T + S_2$$

$$\leq T + \text{skew}_{3 \rightarrow 2} = T + S_3 - S_2$$

$$\leq T + \text{skew}_{2 \rightarrow 3} = T + S_2 - S_3$$

$$\leq T + \text{skew}_{3 \rightarrow 3} = T + S_3 - S_3$$

$t_{\text{hold}} + S_2 \leq$	shortest 1→2
$t_{\text{hold}} + S_3 - S_2 \leq$	2→3
$t_{\text{hold}} + S_2 - S_3 \leq$	3→2
$t_{\text{hold}} + S_3 - S_3 \leq$	3→3

Obj: min T
variables S_2 & S_3

5 Decoders and Multiplexers (20 Points)

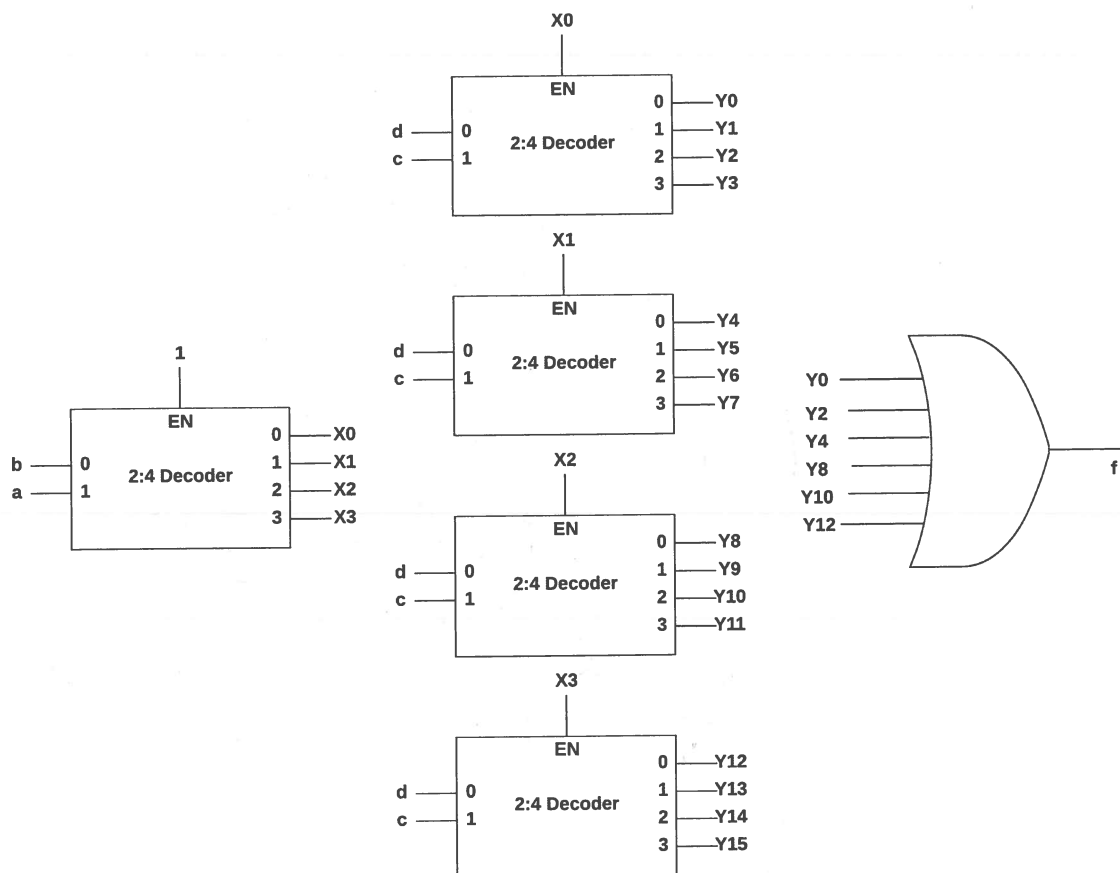
Assume a dual-railed system, where you have access to any variable and its complement. Given a four-input Boolean function

$$f(a, b, c, d) = \sum m(0, 2, 4, 8, 10, 12) + \sum d(1, 5, 9, 11).$$

A. Implement the function using a minimal network of 2:4 decoders and OR gates. (10 Points)

Rubrics:

- 8 points for using incorrect decoders
- 1 point for incorrect/missing EN input/label
- 1 point for incorrect/missing indexing of inputs/outputs (inside/outside decoder)
- 2 points for incorrect/missing inputs to OR gate



2. Implement the function using a minimal network of **4:1 multiplexers**. (10 Points)

Rubrics:

- 0.75 points for no/wrong selector/input label on muxes
- 3 points for using gates
- 2 points for using 2:1 mux instead of 4:1 mux
- 2 points for using more than the 2 muxes
- 1 point for each wrong input

<i>a</i>	<i>b</i>	<i>cd</i>				$D(c,d)$
		00	01	10	11	
0	0	1	X	1	0	D_0
0	1	1	X	0	0	D_1
1	0	1	X	1	X	D_2
1	1	1	0	0	0	D_3

Using the K-map below, we can write $D_0 = d'$.

		<i>d</i>	
		0	1
<i>c</i>	0	1	X
	1	1	0

Using the K-map below, we can write $D_1 = c'$.

		<i>d</i>	
		0	1
<i>c</i>	0	1	X
	1	0	0

Using the K-map below, we can write $D_2 = 1$.

		<i>d</i>	
		0	1
<i>c</i>	0	1	X
	1	1	X

Using the K-map below, we can write $D_3 = c'd'$.

		<i>d</i>	
		0	1
<i>c</i>	0	1	0
	1	0	0

So we need another mux for D_3 but we do not need to write the truth table to implement a 2-variable function using a 4:1 mux. Here is the resulting logic diagram.

