

Midterm 2 Review for CSE 140

Chester Holtz

Based on FA18 & WI16 slides

May 14, 2021

Logistics

- ▶ Released on course website:
`http://cseweb.ucsd.edu/classes/sp21/cse140-a/`
- ▶ Full 48 hours, submission on gradescope
- ▶ Released Sunday 5/16 10:00 am PST, due Tuesday 5/18 10:00 am PST
- ▶ Format same as midterm 1—2 sections:
 - ▶ Section A: 90% question-solving style, roughly 80 minutes
 - ▶ Section B: 10% exploratory style, no penalties

Overview

- ▶ Flip-Flops
- ▶ Mealy & Moore machines
- ▶ State diagrams
- ▶ Timing and retiming

D-Latch

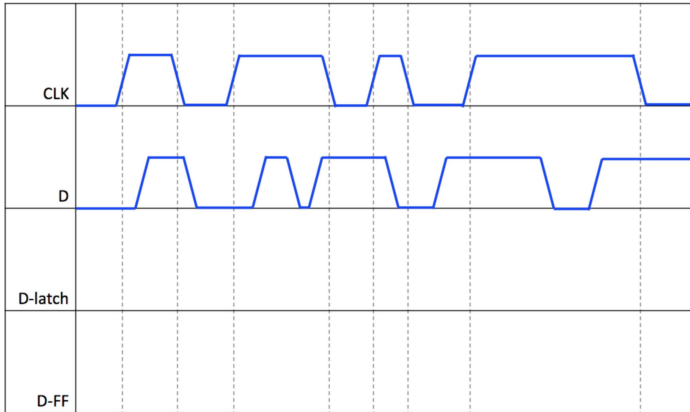
- ▶ Two inputs: clk, D
 - ▶ clk: Controls when the output changes
 - ▶ D (the data input): controls what the output changes to
- ▶ Function
 - ▶ When $\text{clk}=1$, D passes through to Q (the latch is transparent)
 - ▶ When $\text{clk}=0$, Q holds its previous value (the latch is opaque)

D Flip-Flop

- ▶ Two inputs: clk, D
 - ▶ clk: Controls when the output changes
 - ▶ D (the data input): controls what the output changes to
- ▶ Function
 - ▶ The flip-flop “samples” D on the rising edge of clk
 - ▶ When CLK goes from 0 to 1, D passes through Q
 - ▶ Otherwise, Q holds its previous value
 - ▶ Q changes only on the rising edges of CLK

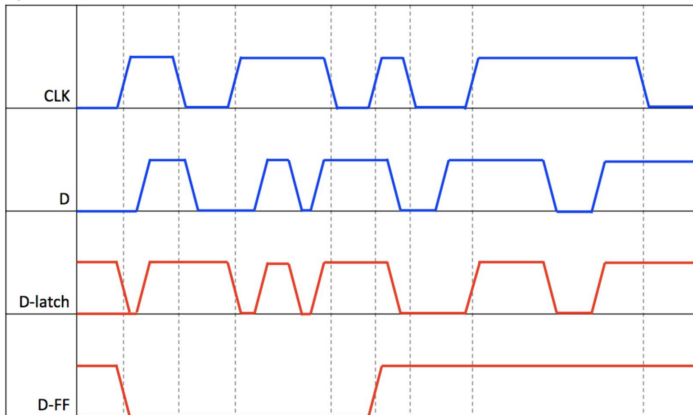
D-Latch vs. D Flip-Flop

Example



D-Latch vs. D Flip-Flop

Example



Mealy and Moore Machines

- ▶ Mealy machine: general
 - ▶ $y_i(t) = f_i(X(t), S(t))$
- ▶ Moore machine: output is independent of current input
 - ▶ $y_i(t) = f_i(S(t))$
 - ▶ $S_i(t + 1) = g_i(X(t), S(t))$

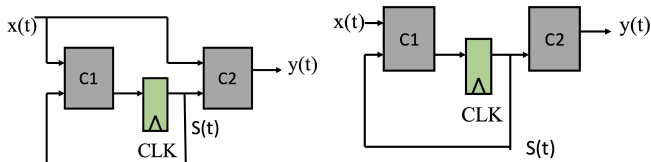
Mealy and Moore Machines

- ▶ Mealy machine: general
 - ▶ $y_i(t) = f_i(X(t), S(t))$
- ▶ Moore machine: output is independent of current input
 - ▶ $y_i(t) = f_i(S(t))$
 - ▶ $S_i(t+1) = g_i(X(t), S(t))$

Mealy and Moore Machines

- ▶ Mealy machine: general
 - ▶ $y_i(t) = f_i(X(t), S(t))$
- ▶ Moore machine: output is independent of current input
 - ▶ $y_i(t) = f_i(S(t))$
 - ▶ $S_i(t + 1) = g_i(X(t), S(t))$

Which is which?



Conversion from Mealy to Moore

Algorithm

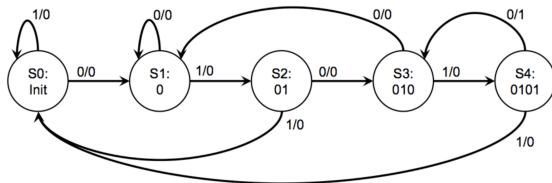
1. Identify distinct (NS, y) pair
2. Replace each distinct (NS, y) pair with distinct new states
3. Insert rows of present state=new states
4. Append each present state with its output y .

Conversion from Mealy to Moore

Algorithm

1. Identify distinct (NS, y) pair
2. Replace each distinct (NS, y) pair with distinct new states
3. Insert rows of present state=new states
4. Append each present state with its output y .

A pattern recognizer produces output $y = 1$ when $x(t - 4, t) = 01010$



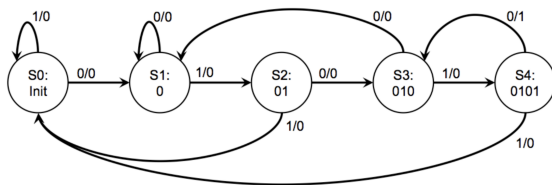
	Input	
	$x = 0$	$x = 1$
S_0		
S_1		
PS S_2		
S_3		
S_4		

Conversion from Mealy to Moore

Algorithm

1. Identify distinct (NS, y) pair
2. Replace each distinct (NS, y) pair with distinct new states
3. Insert rows of present state=new states
4. Append each present state with its output y .

A pattern recognizer produces output $y = 1$ when $x(t-4, t) = 01010$



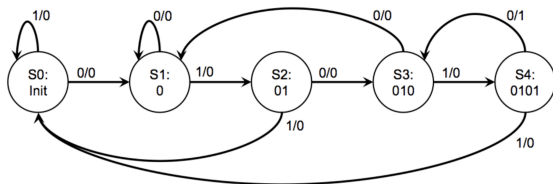
	Input	
	$x = 0$	$x = 1$
S_0	$(S_1, 0)$	$(S_0, 0)$
S_1	$(S_1, 0)$	$(S_2, 0)$
PS S_2	$(S_3, 0)$	$(S_0, 0)$
S_3	$(S_1, 0)$	$(S_4, 0)$
S_4	$(S_3, 1)$	$(S_0, 0)$

Conversion from Mealy to Moore

Algorithm

1. Identify distinct (NS, y) pair
2. Replace each distinct (NS, y) pair with distinct new states
3. Insert rows of present state=new states
4. Append each present state with its output y .

A pattern recognizer produces output $y = 1$ when $x(t-4, t) = 01010$



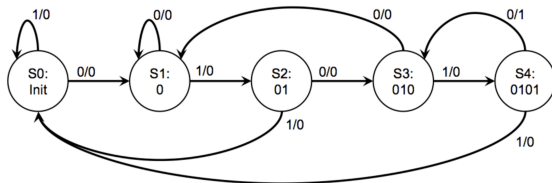
	Input	
	$x = 0$	$x = 1$
S_0	$(S_1, 0)$	$(S_0, 0)$
S_1	$(S_1, 0)$	$(S_2, 0)$
PS S_2	$(S_3, 0)$	$(S_0, 0)$
S_3	$(S_1, 0)$	$(S_4, 0)$
S_4	$(S_3, 1)$	$(S_0, 0)$

Conversion from Mealy to Moore

Algorithm

1. Identify distinct (NS, y) pair
2. Replace each distinct (NS, y) pair with distinct new states
3. Insert rows of present state=new states
4. Append each present state with its output y .

A pattern recognizer produces output $y = 1$ when $x(t - 4, t) = 01010$



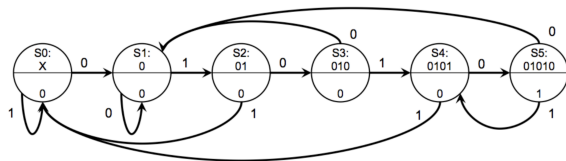
	Input Out		
	x=0	x=1	y
PS	S ₀	S ₁ S ₀	0
	S ₁	S ₁ S ₂	0
	S ₂	S ₃ S ₀	0
	S ₃	S ₁ S ₄	
	S ₄	S ₅ S ₀	
	S ₅		

Conversion from Mealy to Moore

Algorithm

1. Identify distinct (NS, y) pair
2. Replace each distinct (NS, y) pair with distinct new states
3. Insert rows of present state=new states
4. Append each present state with its output y .

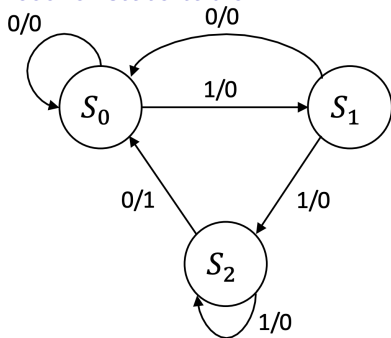
A pattern recognizer produces output $y = 1$ when $x(t - 4, t) = 01010$



	Input Output		
	x=0	x=1	y
S_0	S_1	S_0	0
S_1	S_1	S_2	0
S_2	S_3	S_0	0
S_3	S_1	S_4	0
S_4	S_5	S_0	0
S_5	S_1	S_4	1

State Diagrams

Recover state table



	Input	
	$x = 0$	$x = 1$
S_0		
PS S_1		
S_2		

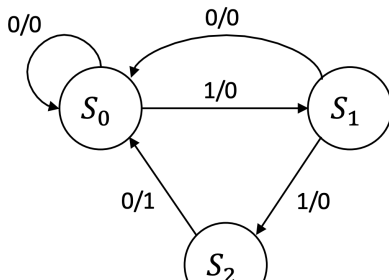
State Diagrams

Algorithm

1. State table
2. Excitation table
3. Kmaps

Recover state table Algorithm

1. State table
2. Excitation table
3. Kmaps



	Input	
	$x = 0$	$x = 1$
S_0	$(S_0, 0)$	$(S_1, 0)$
PS S_1	$(S_0, 0)$	$(S_2, 0)$

State Diagrams

Recover excitation table

$Q_1(t)$	$Q_0(t)$	$X(t)$	$Q_1(t+1)$	$Q_0(t+1)$	y
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

		Input	
		$x = 0$	$x = 1$
PS	S_0	$(S_0, 0)$	$(S_1, 0)$
	S_1	$(S_0, 0)$	$(S_2, 0)$
	S_2	$(S_0, 1)$	$(S_2, 0)$

State Diagrams

Recover excitation table

$Q_1(t)$	$Q_0(t)$	$X(t)$	$Q_1(t+1)$	$Q_0(t+1)$	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	—	—	—
1	1	1	—	—	—

		Input	
		$x = 0$	$x = 1$
PS	S_0	$(S_0, 0)$	$(S_1, 0)$
	S_1	$(S_0, 0)$	$(S_2, 0)$
	S_2	$(S_0, 1)$	$(S_2, 0)$

State Diagrams

Derive equations

- ▶ K-maps for D_0, D_1, y
- ▶ Use D Flip-Flop:
 $Q(t+1) = D(t)$

$Q_1(t)$	$Q_0(t)$	$X(t)$	D_1	D_0	$Q_1(t+1)$	$Q_0(t+1)$	y
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	0	0	0	0	0
0	1	1	1	0	1	0	0
1	0	0	0	0	0	0	1
1	0	1	1	0	1	0	0
1	1	0	—	—	—	—	—
1	1	1	—	—	—	—	—

State Diagrams

Derive equations

- ▶ K-maps for D_0, D_1, y
- ▶ Use D Flip-Flop:
 $Q(t+1) = D(t)$

$$D_1(t) = X(t)Q_0(t) + X(t)Q_1(t)$$

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

$Q_1(t)$	$Q_0(t)$	$X(t)$	D_1	D_0	$Q_1(t+1)$	$Q_0(t+1)$	y
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	0	0	0	0	0
0	1	1	1	0	1	0	0
1	0	0	0	0	0	0	1
1	0	1	1	0	1	0	0
1	1	0	—	—	—	—	—
1	1	1	—	—	—	—	—

State Diagrams

Derive equations

$$D_1(t) = X(t)Q_0(t)' + X(t)Q_1(t)'$$

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

$$D_0(t) = X(t)Q_0'(t)Q_1'(t)$$

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

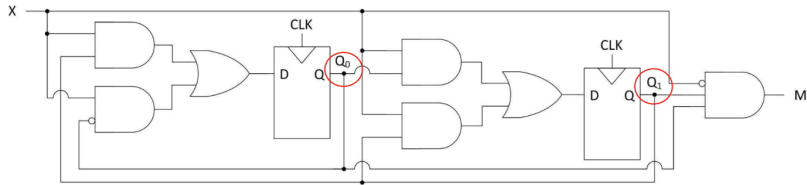
$Q_1(t)$	$Q_0(t)$	$X(t)$	D_1	D_0	$Q_1(t+1)$	$Q_0(t+1)$	y
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	0	0	0	0	0
0	1	1	1	0	1	0	0
1	0	0	0	0	0	0	1
1	0	1	1	0	1	0	0
1	1	0	—	—	—	—	—
1	1	1	—	—	—	—	—

$$y(t) = X'(t)Q_1(t)$$

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

Circuit Implementation

Recover excitation table

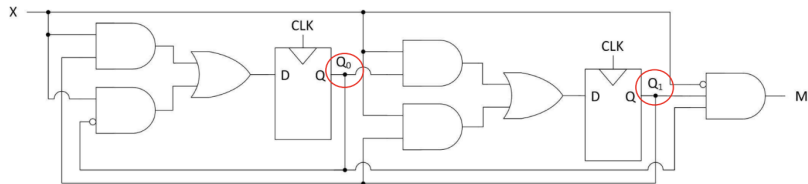


State equations

- ▶ $Q_0(t+1) =$
- ▶ $Q_1(t+1) =$
- ▶ $M =$

Circuit Implementation

Recover excitation table



State equations

- ▶ $Q_0(t+1) = XQ_1(t) + XQ_0(t)'$
- ▶ $Q_1(t+1) = XQ_1(t) + XQ_0(t)$
- ▶ $M = X'Q_1(t)Q_0(t)$

Circuit Implementation

Recover excitation table

- ▶ $Q_0(t+1) = XQ_1(t) + XQ_0(t)'$
- ▶ $Q_1(t+1) = XQ_1(t) + XQ_0(t)$
- ▶ $M = X'Q_1(t)Q_0(t)$

$Q_1(t)$	$Q_0(t)$	$X(t)$	$Q_1(t+1)$	$Q_0(t+1)$	M
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Circuit Implementation

Recover excitation table

- ▶ $Q_0(t+1) = XQ_1(t) + XQ_0(t)'$
- ▶ $Q_1(t+1) = XQ_1(t) + XQ_0(t)$
- ▶ $M = X'Q_1(t)Q_0(t)$

$Q_1(t)$	$Q_0(t)$	$X(t)$	$Q_1(t+1)$	$Q_0(t+1)$	M
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	0

Circuit Implementation

Recover state table

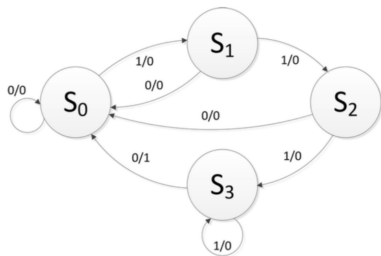
$$\begin{array}{rcl} & Q_1 & Q_0 \\ S_0 & \leftarrow 0 & 0 \\ S_1 & \leftarrow 0 & 1 \\ S_2 & \leftarrow 1 & 0 \\ S_3 & \leftarrow 1 & 1 \end{array}$$

$Q_1(t)$	$Q_0(t)$	$X(t)$	$Q_1(t+1)$	$Q_0(t+1)$	M
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	0

state n input	$X = 0$	$X = 1$
S_0	$S_0, 0$	$S_1, 0$
S_1	$S_0, 0$	$S_2, 0$
S_2	$S_0, 0$	$S_3, 0$
S_3	$S_0, 1$	$S_3, 0$

Circuit Implementation

Derive state diagram



state n input	$X = 0$	$X = 1$
S_0	$S_0, 0$	$S_1, 0$
S_1	$S_0, 0$	$S_2, 0$
S_2	$S_0, 0$	$S_3, 0$
S_3	$S_0, 1$	$S_3, 0$

Sequential Networks & Timing Analysis

Definitions

- ▶ Setup time t_{setup} Time before the clock data that must be stable (i.e. not change)
 - ▶ Setup time violation occurs if the input signal D does not settle (set up) to the stable value at least t_{setup} before the clock edge.
- ▶ Hold time t_{hold} Time after the clock edge that data must be stable
 - ▶ Hold time violation occurs if the input signal D does not remain unchanged (hold) at least t_{hold} after the clock edge.

Sequential Networks & Timing Analysis

Definitions

- ▶ Setup time t_{setup} Time before the clock data that must be stable (i.e. not change)
 - ▶ Setup time violation depends on maximum delay through the combinational logic
- ▶ Hold time t_{hold} Time after the clock edge that data must be stable
 - ▶ Hold time violation depends on the minimum delay through the combinational logic

Sequential Networks & Timing Analysis

Timing constraints

- ▶ Skew constraint

$$t_{\text{skew}} = \text{CLK1} - \text{CLK2}$$

- ▶ Setup time constraint

$$T_C \geq t_{pcq} + t_{pd} + t_{\text{setup}} - t_{\text{skew}}$$

- ▶ Hold time constraint

$$t_{ccq} + t_{cd} > t_{\text{hold}} + t_{\text{skew}}$$

Circuit Properties	
Flip-Flops	Gates
t_{setup} t_{hold} t_{skew} t_{pcq} t_{ccq}	t_{pd} (prop. delay) t_{cd} (contam. delay)

Sequential Networks & Timing Analysis

Timing constraints

- ▶ Skew constraint

$$t_{\text{skew}} = \text{CLK1} - \text{CLK2}$$

- ▶ Setup time constraint

$$T_C \geq t_{pcq} + t_{pd} + t_{\text{setup}} - t_{\text{skew}}$$

- ▶ Hold time constraint

$$t_{ccq} + t_{cd} > t_{\text{hold}} + t_{\text{skew}}$$

Circuit Properties	
Flip-Flops	Gates
t_{setup} t_{hold} t_{skew} t_{pcq} t_{ccq}	t_{pd} (prop. delay) t_{cd} (contam. delay)

Sequential Networks & Timing Analysis

Timing constraints

- ▶ Skew constraint

$$t_{\text{skew}} = \text{CLK1} - \text{CLK2}$$

- ▶ Setup time constraint

$$T_C \geq t_{pcq} + t_{pd} + t_{\text{setup}} - t_{\text{skew}}$$

- ▶ Hold time constraint

$$t_{ccq} + t_{cd} > t_{\text{hold}} + t_{\text{skew}}$$

Circuit Properties	
Flip-Flops	Gates
t_{setup}	t_{pd} (prop. delay) t_{cd} (contam. delay)
t_{hold}	
t_{skew}	
t_{pcq}	
t_{ccq}	

Examples given a circuit:

- ▶ What is the maximum clock frequency?
- ▶ Clock skew tolerance s.t. hold time constraint satisfied?

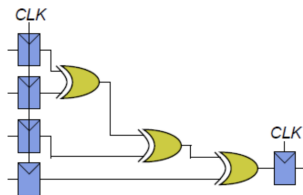
Sequential Network & Timing Analysis Example

Setup time constraint

$$T_C \geq t_{pcq} + t_{pd} + t_{setup} - t_{skew}$$

Hold time constraint

$$t_{ccq} + t_{cd} > t_{hold} + t_{skew}$$



FA'10 Midterm2 Q.4

Circuit Properties	
Flip-Flops	Gates
$t_{setup} = 60ps$	$t_{pd} = 100ps$ (prop. delay) $t_{cd} = 55ps$ (contam. delay)
$t_{hold} = 20ps$	
$t_{pcq} = 70ps$	
$t_{ccq} = 50$	

1. If there is no clock skew, what is the *maximum operating frequency*?
2. How much clock skew can the circuit tolerate before *hold-time violation*?

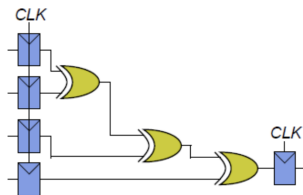
Sequential Network & Timing Analysis Example

Setup time constraint

$$T_C \geq t_{pcq} + t_{pd} + t_{setup} - t_{skew}$$

Hold time constraint

$$t_{ccq} + t_{cd} > t_{hold} + t_{skew}$$



FA'10 Midterm2 Q.4

1. If there is no clock skew, what is the *maximum operating frequency*?

Circuit Properties	
Flip-Flops	Gates
$t_{setup} = 60ps$ $t_{hold} = 20ps$ $t_{pcq} = 70ps$ $t_{ccq} = 50$	$t_{pd} = 100ps$ (prop. delay) $t_{cd} = 55ps$ (contam. delay)

$$\begin{aligned} T_C &\geq t_{pcq} + t_{pd} + t_{setup} - t_{skew} \\ &\geq 70 + 3 \cdot 100 + 60 = 430 \text{ ps} \end{aligned}$$

$$F = \frac{1}{T_C} = 2.33 \text{ GHz}$$

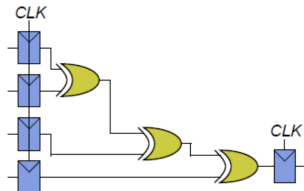
Sequential Network & Timing Analysis Example

Setup time constraint

$$T_C \geq t_{pcq} + t_{pd} + t_{setup} - t_{skew}$$

Hold time constraint

$$t_{ccq} + t_{cd} > t_{hold} + t_{skew}$$



FA'10 Midterm2 Q.4

2. How much clock skew can the circuit tolerate before *hold-time violation*?

Circuit Properties	
Flip-Flops	Gates
$t_{setup} = 60ps$	
$t_{hold} = 20ps$	
$t_{pcq} = 70ps$	$t_{pd} = 100ps$ (prop. delay)
$t_{ccq} = 50$	$t_{cd} = 55ps$ (contam. delay)

$$\begin{aligned} T_{ccq} + T_{cd} &\geq T_{hold} + T_{skew} \\ T_{skew} &\leq 50 + 1 \cdot 55 - 20 \\ &\leq 85 \text{ ps} \end{aligned}$$

Some more tips

- ▶ Part A (problem-solving)
 - ▶ Similar to previous exams
 - ▶ 4 Subsections
 - ▶ Latch & Flip-Flop
 - ▶ State + Mealy/Moore machines
 - ▶ Timing analysis (helpful tutorial in case we didn't get here:
<https://www.youtube.com/watch?v=JElzhGPYp68>)
- ▶ Part B (exploratory)
 - ▶ 3 Subsections

Good Luck!