18-240: Structure and Design of Digital Systems

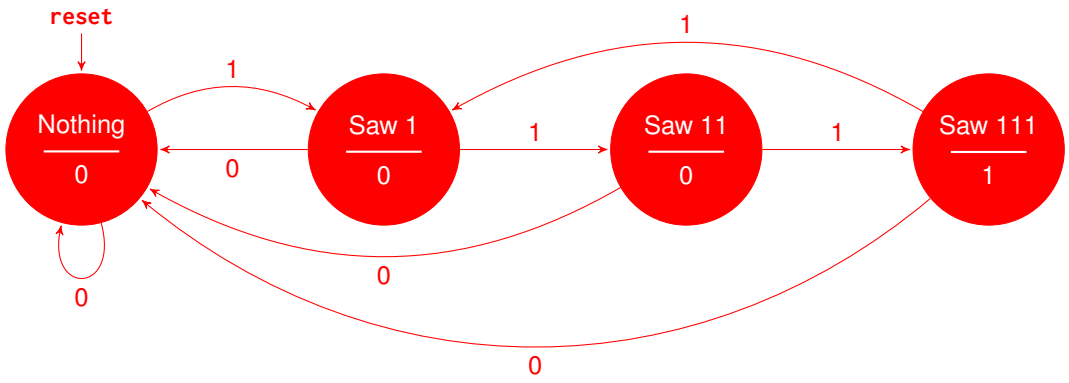
Electrical & Computer ENGINEERING

HW4 Solutions [9 problems, 64 points]

Covers lectures L08 – L10 *Due: 9 October 2023*

Drill Problems [34 points]

1. [6 points, Lecture 8] Draw the State Transition Diagram for a Moore FSM that detects the sequence **111** on its single input. It signals such a sequence with a **1** on its single output, as soon as it can after seeing the last '1' of the sequence. It should not detect overlapping sequences.



2. [8 points, Lecture 9] Finish the design and implementation of the Moore-style sequence detector from problem 1. Draw the state transition table and write the next-state and output generator equations. Finally, draw the schematic for the circuit.

   I will use the state assignment of Nothing = `2'b00`, Saw1 = `2'b01`, Saw11 = `2'b10`, and Saw111 = `2'b11`. If you chose differently, your answers will be different. `A` is my input variable. `B` is my output variable.

   My State Transition Table starts like the one on the left and then gets rewritten with the state assignment values:

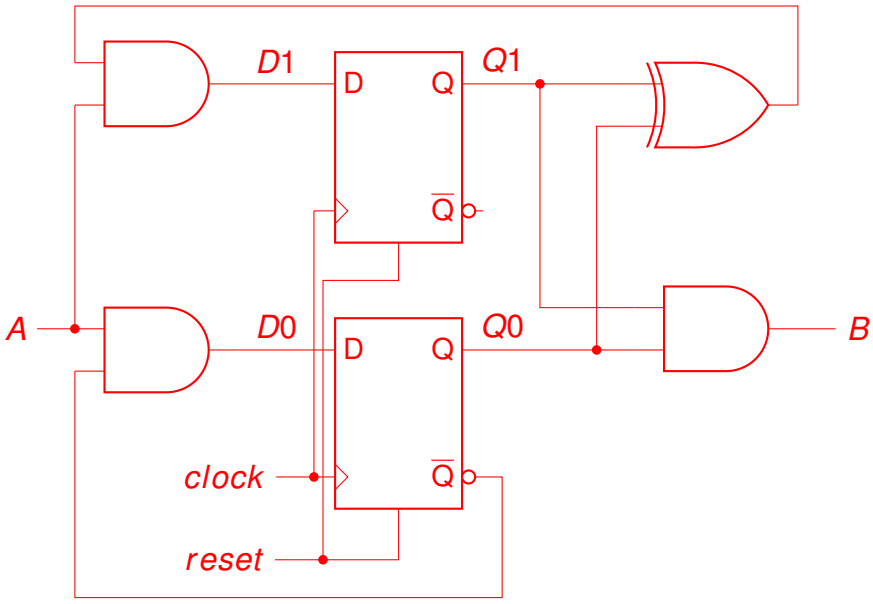
Current State	A	Next State	B
Nothing	0	Nothing	0
Nothing	1	Saw1	0
Saw1	0	Nothing	0
Saw1	1	Saw11	0
Saw11	0	Nothing	0
Saw11	1	Saw111	0
Saw111	0	Nothing	1
Saw111	1	Nothing	1

Q1	Q0	A	D1	D0	B
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	0	0	1

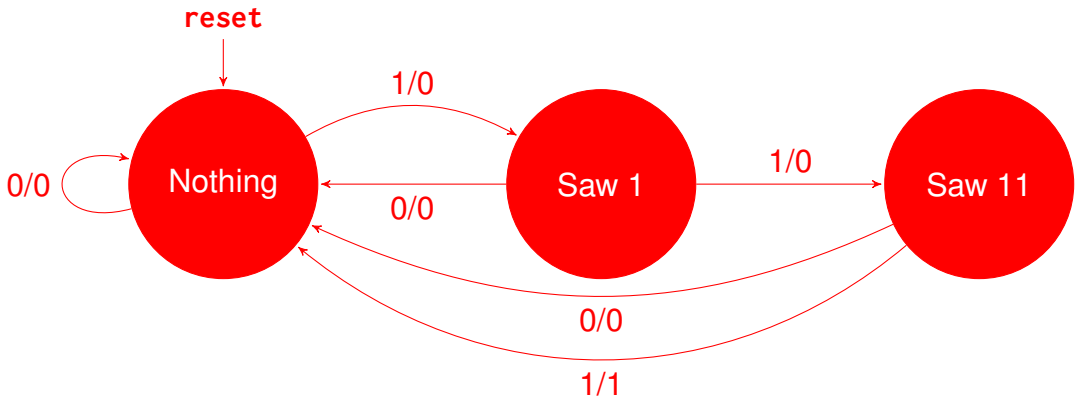
From the State Transition Table, I can use my Combinational Circuitry Foo™ to write the following equations:

$$
\begin{aligned}
D1 &= \overline{Q1} \cdot Q0 \cdot A + Q1 \cdot \overline{Q0} \cdot A \\
&= A \cdot (Q1 \oplus Q0) \\
D0 &= \overline{Q1} \cdot \overline{Q0} \cdot A + Q1 \cdot \overline{Q0} \cdot A \\
&= A \cdot \overline{Q0} \\
B &= Q1 \cdot Q0
\end{aligned}
$$

Which results in the following circuit:



3. [6 points, Lecture 8] Draw the State Transition Diagram for a Mealy FSM that detects the sequence **111** on its single input. It signals such a sequence with a **1** on its single output, as soon as it can after seeing the last '1' of the sequence. It should not detect overlapping sequences.



4. [8 points, Lecture 9] Finish the design and implementation of the Mealy-style sequence detector from problem 3. Point out the qualitative differences in the state transition table, equations and schematics (i.e. I'm not looking for detailed "this is a one, where that is a zero" type differences. Instead, show me the big differences.)

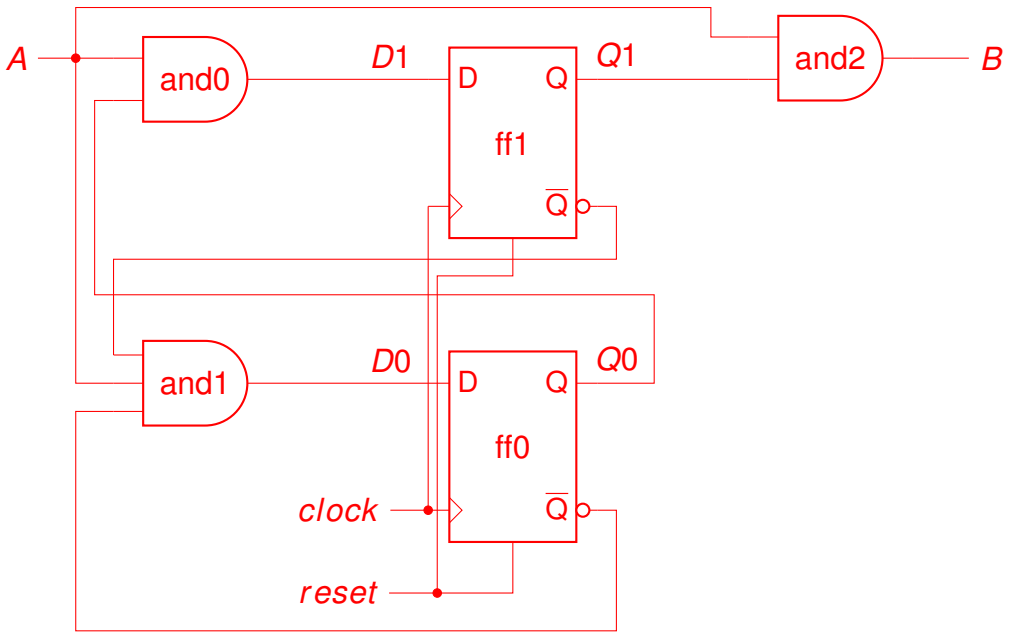
I will use the same state assignments, input names and output names as problem 3.

Current State	A	Next State	B
Nothing	0	Nothing	0
Nothing	1	Saw1	0
Saw1	0	Nothing	0
Saw1	1	Saw11	0
Saw11	0	Nothing	0
Saw11	1	Nothing	1

Q1	Q0	A	D1	D0	B
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	0	0	1
1	1	0	X	X	X
1	1	1	X	X	X

$$D1 = Q0 \cdot A$$
$$D0 = \overline{Q1} \cdot \overline{Q0} \cdot A$$
$$B = Q1 \cdot A$$

Which results in the following circuit:



Differences include:

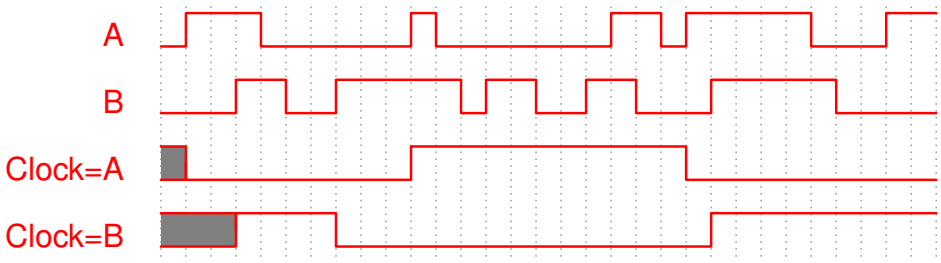
- One less state, which resulted in two rows of don't cares in the state transition table.
- Undoubtedly, the previous point led to simpler equations.
- Notice also that the output equation ($B = Q1 \cdot A$) includes the input as one of the literals. Mealy machine!
- You can see the Mealy machine in the schematic as well. The AND gate that generates B is hooked up to A.

5. [6 points, Lecture 8] Provide the State Transition Table for this State Transition Diagram.

State	A	B	Next State	Yellow	Red	Green
Maple	0	0	Oak	1	1	0
	0	1	Xmas	1	1	0
	1	0	Xmas	1	1	0
	1	1	Oak	1	1	0
Oak	X	X	Oak	0	1	0
Cherry	0	0	Oak	0	1	0
	0	1	Xmas	0	1	0
	1	0	Maple	0	1	0
	1	1	Cherry	0	1	0
Xmas	0	X	Xmas	0	1	1
	1	X	Oak	0	1	1

Non-Drill Problems [30 points]

6. [6 points, Lecture 8] Given the input waveform below, sketch the output (**Q**) of a D Flip Flop with **A** connected to the clock input and **B** connected to the **D** input. Then, reverse the connections (**B** $\Rightarrow$ **clock**, **A** $\Rightarrow$ **D**) and sketch again.

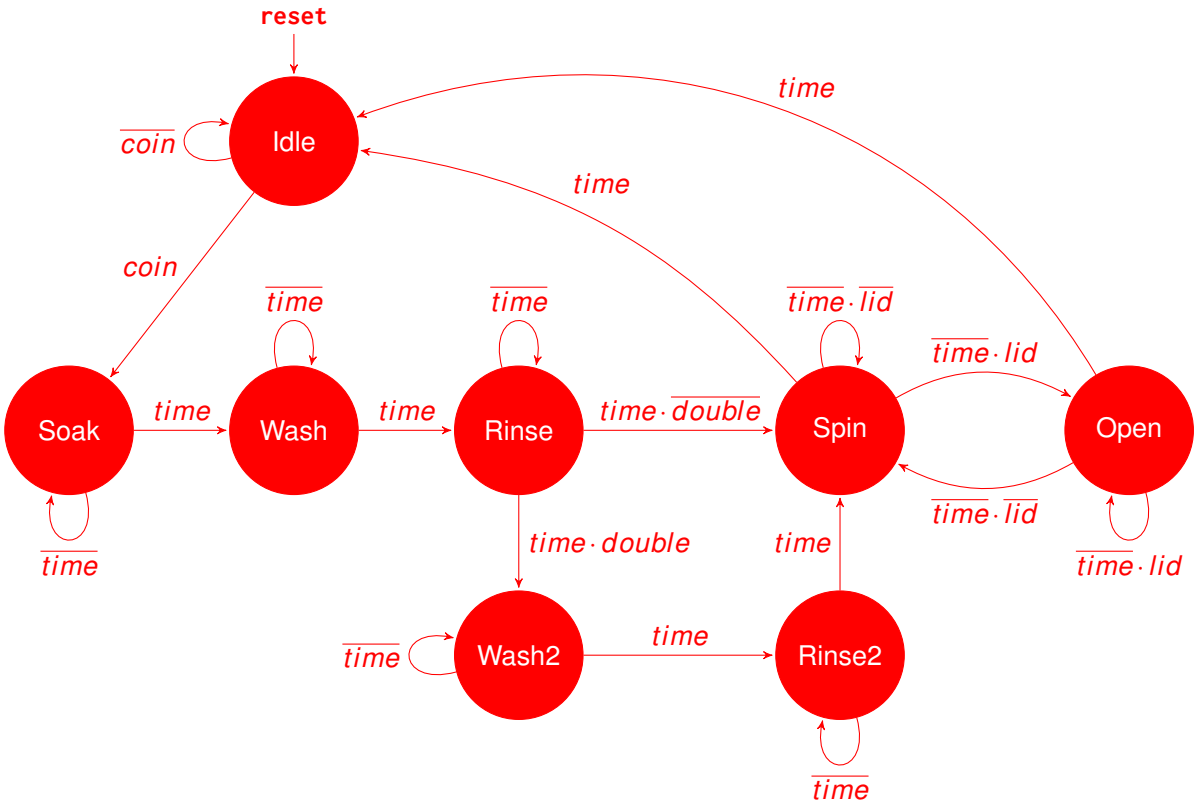


Notice that until the first clock edge occurs, we have no idea what the output will be. (That's the grey area for the first few time periods). Flip-flops don't power up with any set value stored inside them. You have to reset/preset them or give them a clock edge in order to initialize the Q output.

7. [6 points, Lecture 9] Write the SystemVerilog implementation of this STD. Your implementation should follow the style of Lecture 9, slide 21. Use the following header:

hw4prob7.sv is posted to Canvas. Note that my solution is based on Kmapping out the state transition table, which depends on my choice of state assignments. Your answer may vary a small bit if you used a different state assignment.

8. [10 points, Lecture 9] You are to develop a state diagram for a washing machine. You have four inputs: **coin**, **double**, **time**, **lid**. After **reset**, the machine waits until a coin is deposited (i.e. **coin** = 1). It then sequences through the following stages: soak, wash, rinse and spin. If **double** = 1 at the end of the first rinse phase, that means a "double wash" has been requested and the sequence should be: soak, wash, rinse, wash, rinse and spin.



9. [8 points, Lecture 9] Draw the State Transition Diagram for a Mealy machine with two inputs **(A, B)** and two outputs **(F, G)**. **F** is active anytime **A** is a one and **B** hasn't been a one for two clocks (i.e., the two most recent clock edges). **G** is active any time **A** is active and **B** was active at the last clock.

The states are important to keep track of how many times **B** has been active. In some states, the output is active only when **A** is. In the STD below, if nothing follows the slash, then no output is active.