Sequential Logic Circuits

CS 350: Computer Organization & Assembler Language Programming

[3/9: Split FSMs to next lecture]

A. Why?

- Sequential logic units are circuits that combine calculation with stored information.
- Clocks are used with flip-flops to separate computations/memory reads & memory writes.

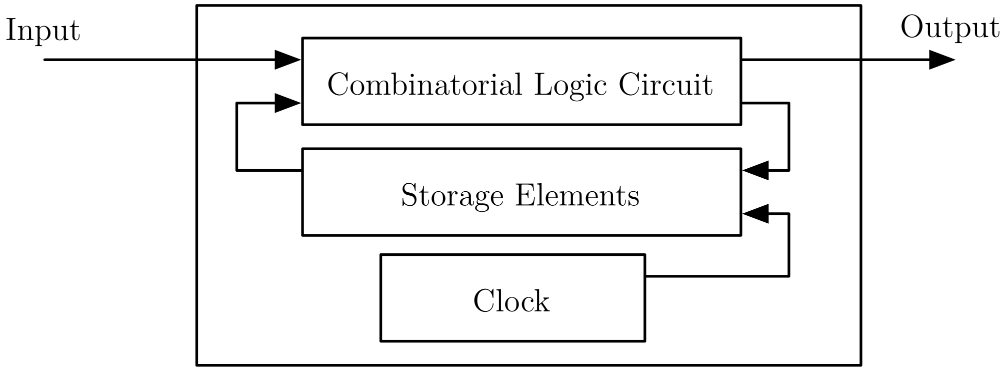
B. Outcomes

After this lecture, you should know

- What sequential logic circuits, clocks, and flip-flops do and why we have them.

C. Sequential Logic Circuits

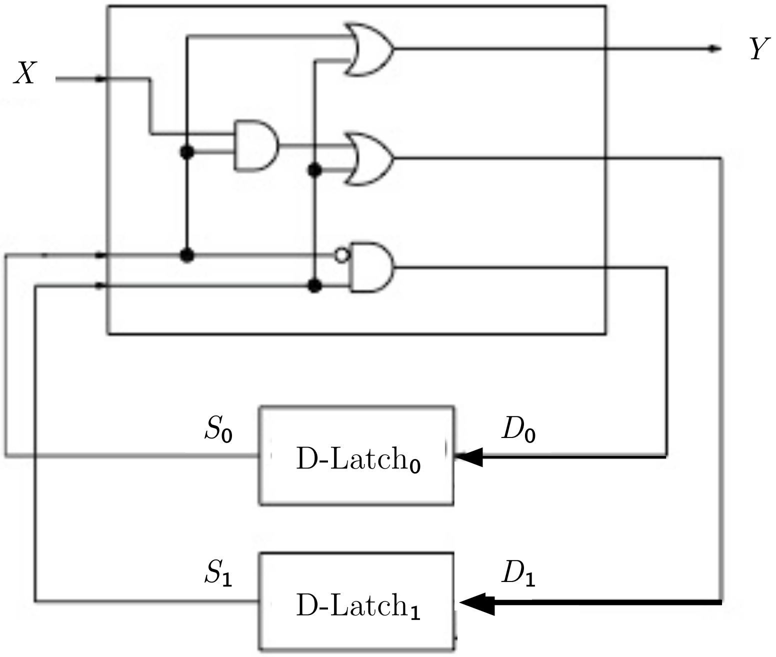
- A **Sequential logic circuit** combines combinatorial circuits and storage elements. The combinatorial circuit calculations take input values and memory values and produce output values and updated memory values.
- Calculations and memory read/writes are synchronized:
  - Read memory & get inputs, then do calculation, then update memory & produce outputs.
  - To help us synchronize, we'll use a clock (cycles between 0 and 1 at regular intervals). We'll allow read memory/get input/do calculation during one interval and allow memory update during the other interval.



Modified Figure 3.23

D. Clocks and Flip-Flops

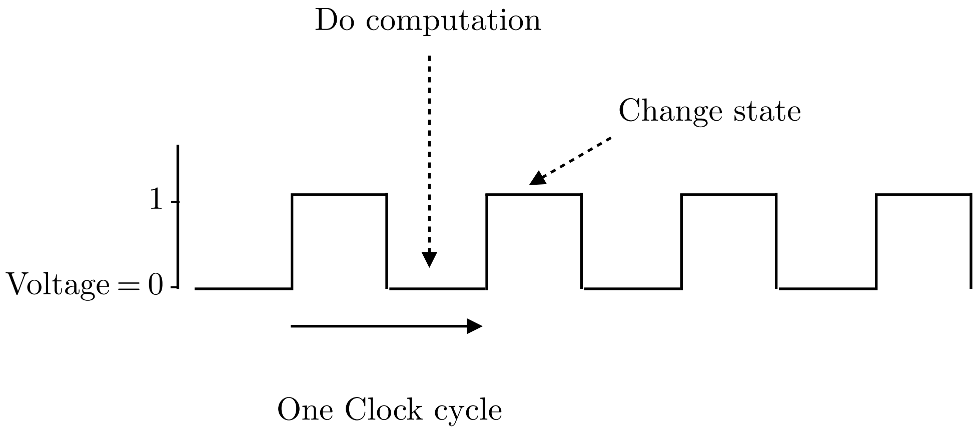
- Different parts of a circuit can take different amounts of time to do their work.
  - Longer wires take more time; a long sequence of gates takes more time.
- If different parts of memory get updated at different rates then memory might mix already-computed values with not-yet-computed values.
- Could cause problems; this is why we need synchronization.



Modified Figure 3.43

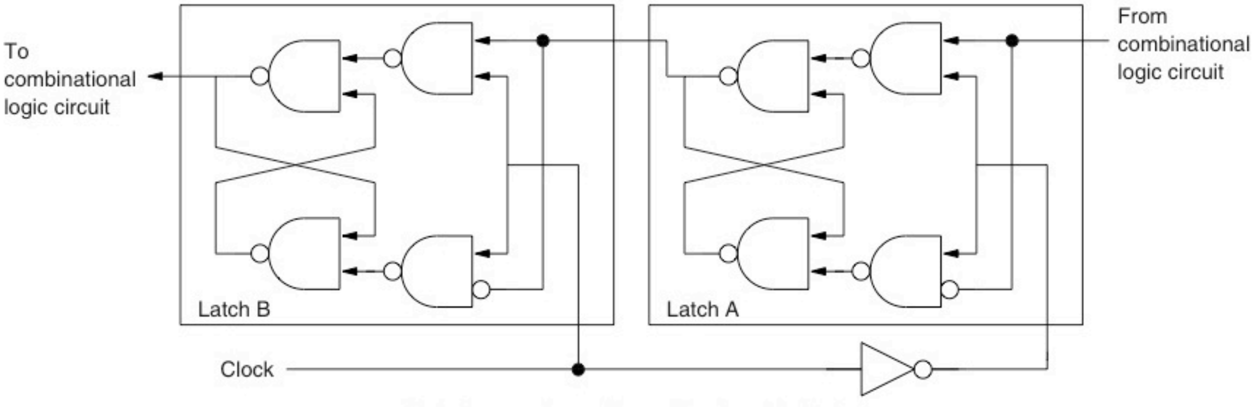
- Above, $Y = S_0 + S_1$; $D_0 = \neg(S_0)\,S_1$; $D_1 = X\,S_0 + S_1$.

- We want to set S_1 and S_0 to D_1 and D_0, but the ordering makes a difference:
  - Ordering 1: Separate calculations and memory updates.
    - Calculate Y, D_0, and D_1, then set $S_0 = D_0$ and $S_1 = D_1$.
  - Ordering 2: Interleave calculations and memory updates.
  - Calculate Y and D_0, set $S_0 = D_0$, calculate D_1, and finally set $S_1 = D_1$.
  - (There exist other orderings too.)
- We want to alternate between using memory and setting memory.
  - For the example above, we want to calculate both D bits, then set both S bits, then calculate the D bits, etc. If we don't, then the two orderings might produce different results (try $S_1 = 0$ and $X = S_0 = 1$).
- To solve this problem, we can use a **Clock Signal** to control whether we're calculating or setting memory. A clock signal alternates between on and off at a regular speed. We will update memory during half the cycle and do calculations during the other half.



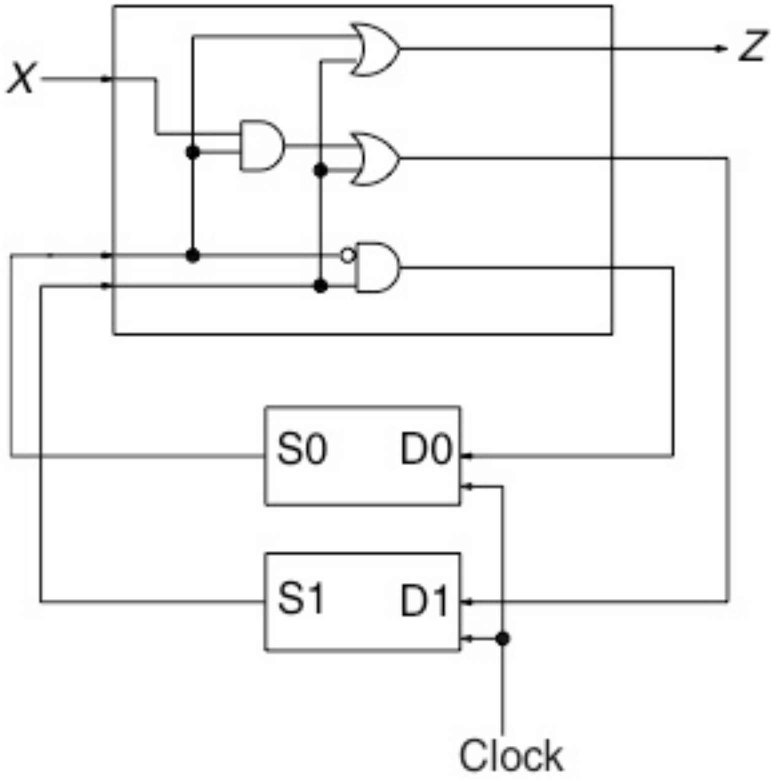
- A **Master-Slave Flip-Flop** stores 1 bit of data; it alternates between two modes. In the first mode, it reads and stores 1 bit of input but its output is the old value of the bit. In the second mode, it ignores its input but updates its output to match the stored bit value.
  - It has two D-latches and a clock connection.
    - The right-hand D-latch output is the left-hand D-latch input.
  - When the clock is 0

  - The combinatorial circuit does its calculation based on the value in the left-hand latch and sends its output to the right-hand latch
  - The right-hand latch sets its state to the value of the combinatorial circuit.
  - The left-hand latch sends its value as the flip-flop output but ignores the value of the right-hand latch (as it possibly changes).
- When the clock is 1
  - The left-hand latch reads the value in the right-hand latch.
  - The right-hand latch ignores the value being produced by the combinatorial circuit.
  - The combinatorial circuit does its calculation based on the changing value of the left-hand latch.


To combinational logic circuit
From combinational logic circuit
Latch B
Latch A
Clock

(c) A storage element (a master-slave flip-flop)

- For an m-bit register, we have m flip-flops all connected to the same clock.
  - Either m bits of data are being read into the register (without changing the register's output).
  - Or, m bits of data are being output from the register (while ignoring the data entering the register).
- In the unmodified Figure 3.43, we have two flip-flops connected to the clock, so D_0 and D_1 get calculated, then S_0 and S_1 get set, then D_0 and D_1 get calculated, and so on.



Unmodified Figure 3.43

Sequential Logic Circuits

CS 350: Computer Organization & Assembler Language Programming

E. Why?

- Sequential logic units are circuits that combine calculation with a fixed amount of stored information.

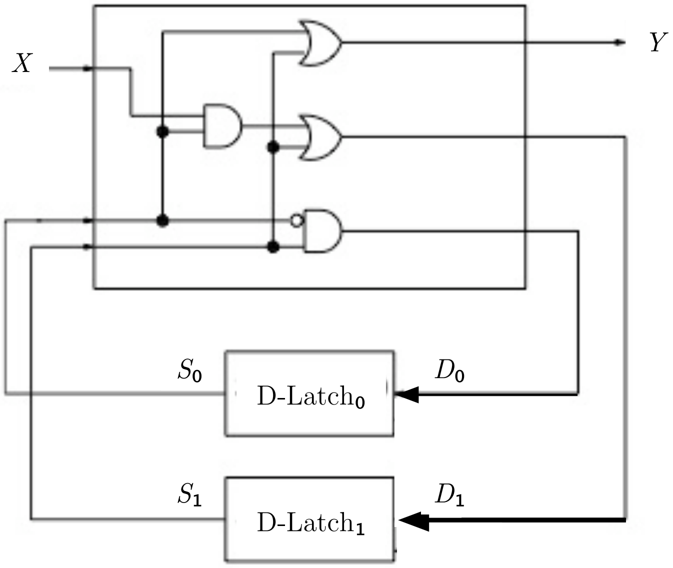
F. Objectives

At the end of this activity, you should:

- Be able to trace the execution of a simple sequential logic circuit.

G. Questions

1. Refer to the modified Figure 3.43 from lecture:



(a) Say $S_1 = 0$ and $X = S_0 = 1$. What are the sequence of Y, S_0, and S_1 values we get if we calculate Y, then D_0, then D_1, and then set S_0 and S_1 (and repeat)?

(b) Repeat part (a) but assume $X = S_1 = 0$ and $S_0 = 1$.

(c) Repeat (part a) but assume we calculate Y then D_0, then set S_0, then calculate D_1, and then set S_1 (and repeat).

Solution

1a. In general $D_0 = \overline{S}_0\ S_1$, $D_1 = X\ S_0 + S_1$, and $Y = S_0 + S_1$.
We start with $S_0 = 1$, $S_1 = 0$, and $X = 1$. Then

- $Y = S_0 + S_1 = 1 + 0 = 1$
- $D_0 = \overline{S}_0\ S_1 = \overline{1}\ 0 = 0$ and $D_1 = X\ S_0 + S_1 = 1\,1 + 0 = 1$
- $S_0 \leftarrow D_0 = 0$ and $S_1 \leftarrow D_1 = 1$

Repeating, we get

- $Y = S_0 + S_1 = 0 + 1 = 1$
- $D_0 = \overline{S}_0\ S_1 = \overline{0}\ 1 = 1$ and $D_1 = X\ S_0 + S_1 = 1\,0 + 1 = 1$
- $S_0 \leftarrow D_0 = 1$ and $S_1 \leftarrow D_1 = 1$

1b. This time we start with $S_0 = 1$, $S_1 = 0$, and $X = 0$. Then

- $Y = S_0 + S_1 = 1 + 0 = 1$
- $D_0 = \overline{S}_0\ S_1 = \overline{1}\ 0 = 0$ and $D_1 = X\ S_0 + S_1 = 0\,1 + 0 = 0$
- $S_0 \leftarrow D_0 = 0$ and $S_1 \leftarrow D_1 = 0$

Repeating, we get

- $Y = S_0 + S_1 = 0 + 0 = 0$
- $D_0 = \overline{S}_0\ S_1 = \overline{0}\ 0 = 0$ and $D_1 = X\ S_0 + S_1 = 0\,0 + 0 = 0$
- $S_0 \leftarrow D_0 = 0$ and $S_1 \leftarrow D_1 = 0$

1c. We again start with $S_0 = 1$, $S_1 = 0$, and $X = 1$ but change the order of computations.

- $Y = S_0 + S_1 = 1 + 0 = 0$
- $D_0 = \overline{S}_0\ S_1 = \overline{1}\ 1 = 0$ and then $S_0 \leftarrow D_0 = 0$
- $D_1 = X\ S_0 + S_1 = 0\,0 + 0 = 0$ and then $S_1 \leftarrow D_1 = 0$

Repeating, we get

- $Y = S_0 + S_1 = 0 + 0 = 0$
- $D_0 = \overline{S}_0\ S_1 = \overline{0}\ 0 = 0$ and then $S_0 \leftarrow D_0 = 0$
- $D_1 = X\ S_0 + S_1 = 1\,0 + 0 = 0$ and then $S_1 \leftarrow D_1 = 0$