State Machine Outputs and Logic Glitches

Announcements

- Homework #10 due Today
- Homework #11 due in one week
- Exam #3 Monday
  - Study Jam
  - Study Topics will be posted
- Read Chapter 11 Sections 1-4

Exam #3 Study Jam

Sunday November 17, 2019

12:00 - 4:00 pm

GOL 1360

Bring questions

State Machine VHDL in ECTET

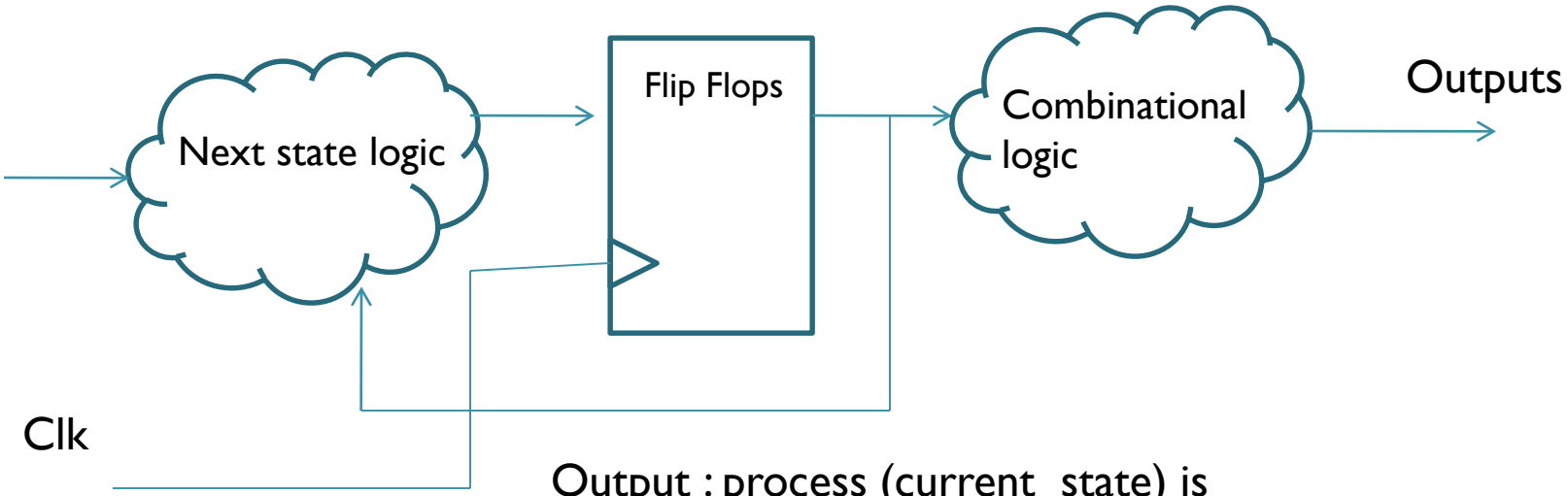
- We follow a 3 (+) process model for state machines
  - Synchronizing process
    - Current_state <= next_state
  - Combinational process
    - Sets up the next_state signal
    - Next_state should be the only output from this process
  - Output process(es)
    - Each state machine output should have its own process

State Machine VHDL in ECTET

- Any counters or timers that are needed should also be in their own process

Synchronizing FSM outputs

- FSM with unsynchronized outputs

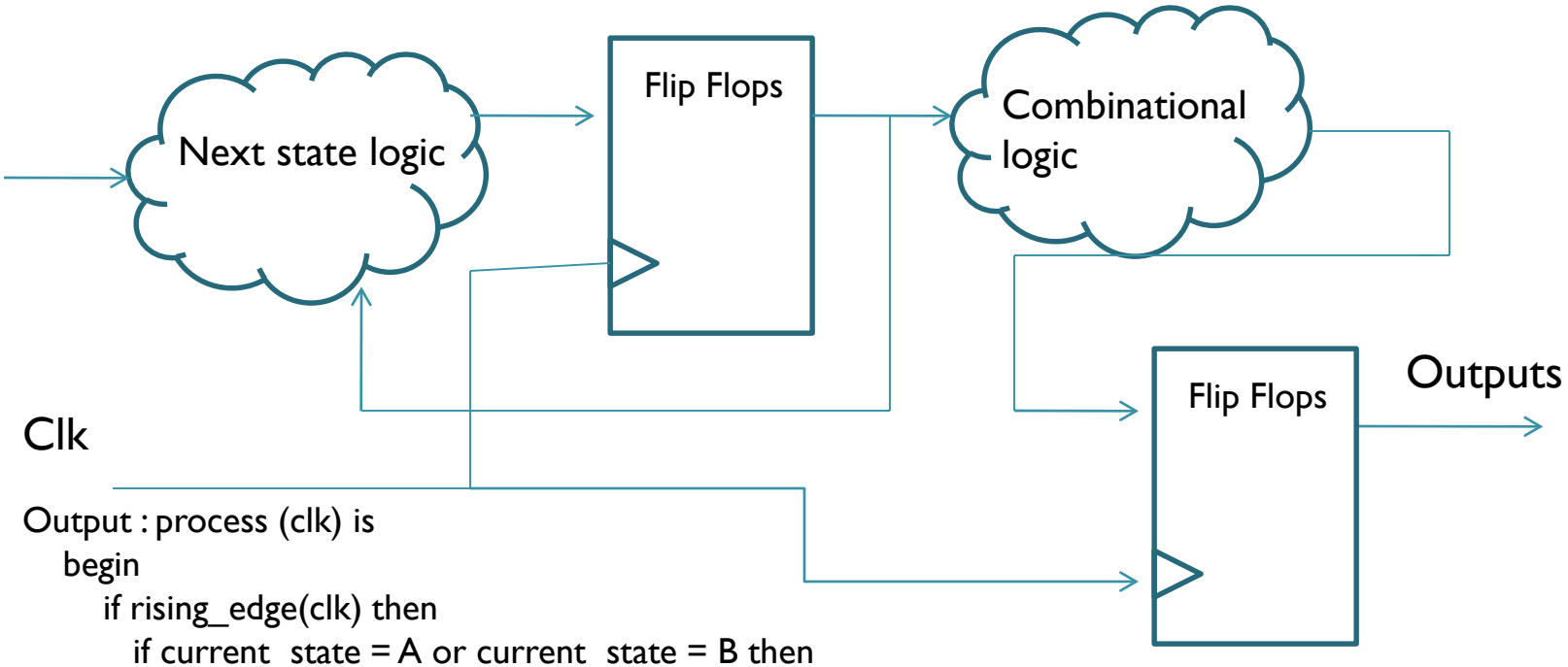


Problem?
Combinational logic on Output prone to glitches

```
Output : process (current_state) is
   begin
      if current_state = A or current_state = B then
            outputAB <= '1';
      else
            outputAB <= '0';
      end if;
   end process;
```

Synchronizing FSM outputs

- This would solve the potential glitch problem



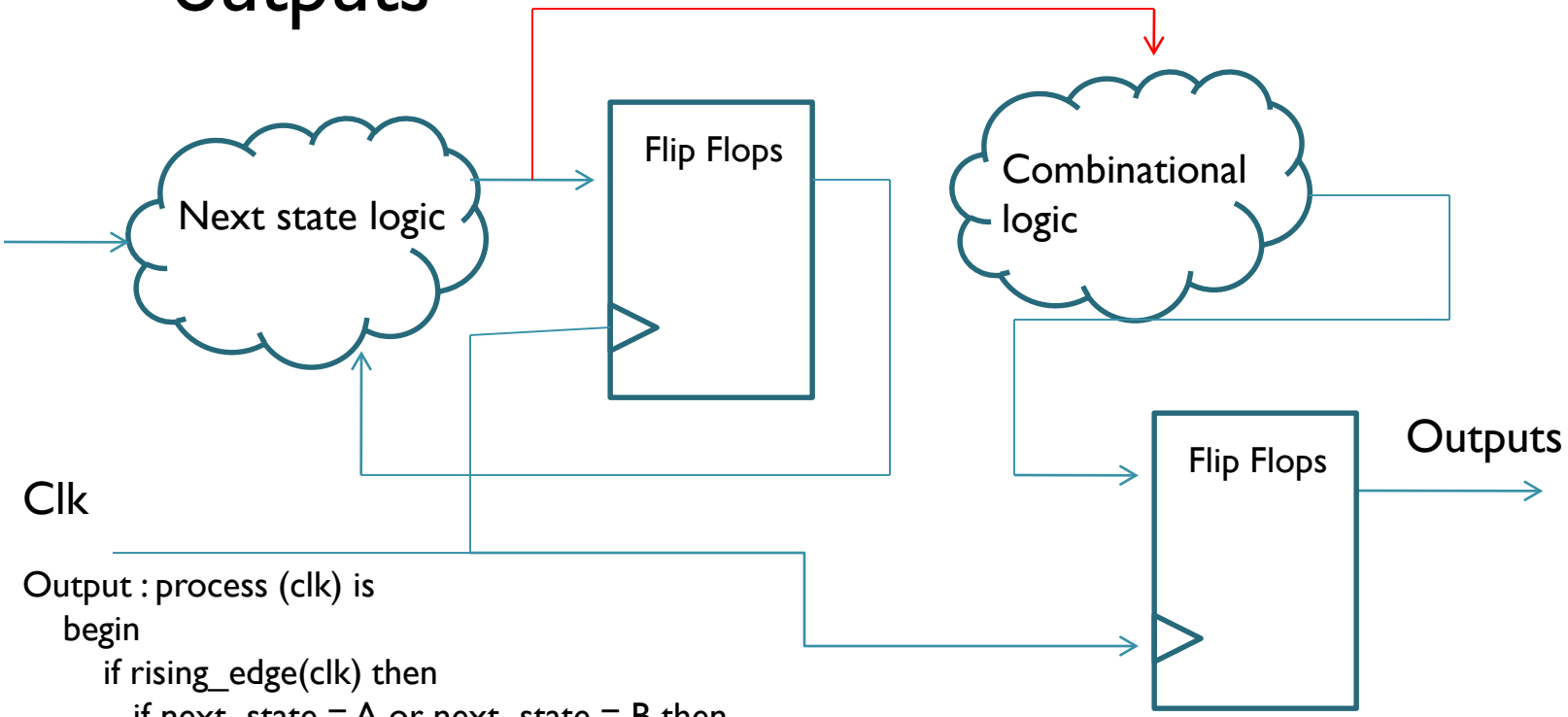

```
Output : process (clk) is
   begin
      if rising_edge(clk) then
         if current_state = A or current_state = B then
               outputAB <= '1';
         else
               outputAB <= '0';
         end if;
      end if;
   end process;
```

Problem?

Outputs are delayed by 1 clock cycle

Synchronizing FSM outputs

- Solve the problem without delaying outputs

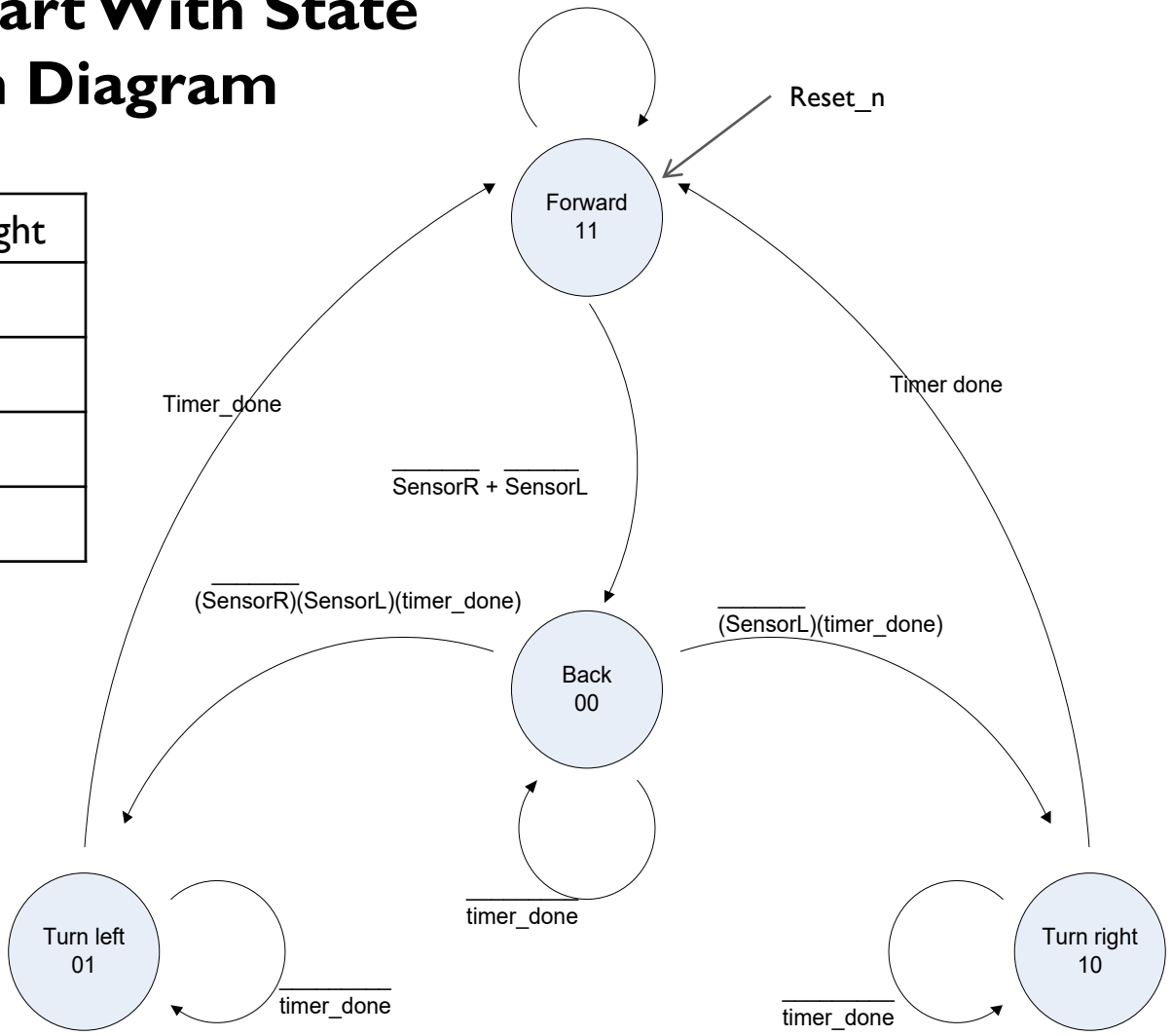



```
Output : process (clk) is
   begin
      if rising_edge(clk) then
         if next_state = A or next_state = B then
               outputAB <= '1';
         else
               outputAB <= '0';
         end if;
      end if;
   end process;
```

FSM VHDL example

Always Start With State Transition Diagram

State	Left	Right
Forward	1	1
Back	0	0
TurnR	1	0
TurnL	0	1



FSM VHDL example

- Enumerated State Type

- Synchronizing process

FSM VHDL example

- Combinatorial process (next state logic)

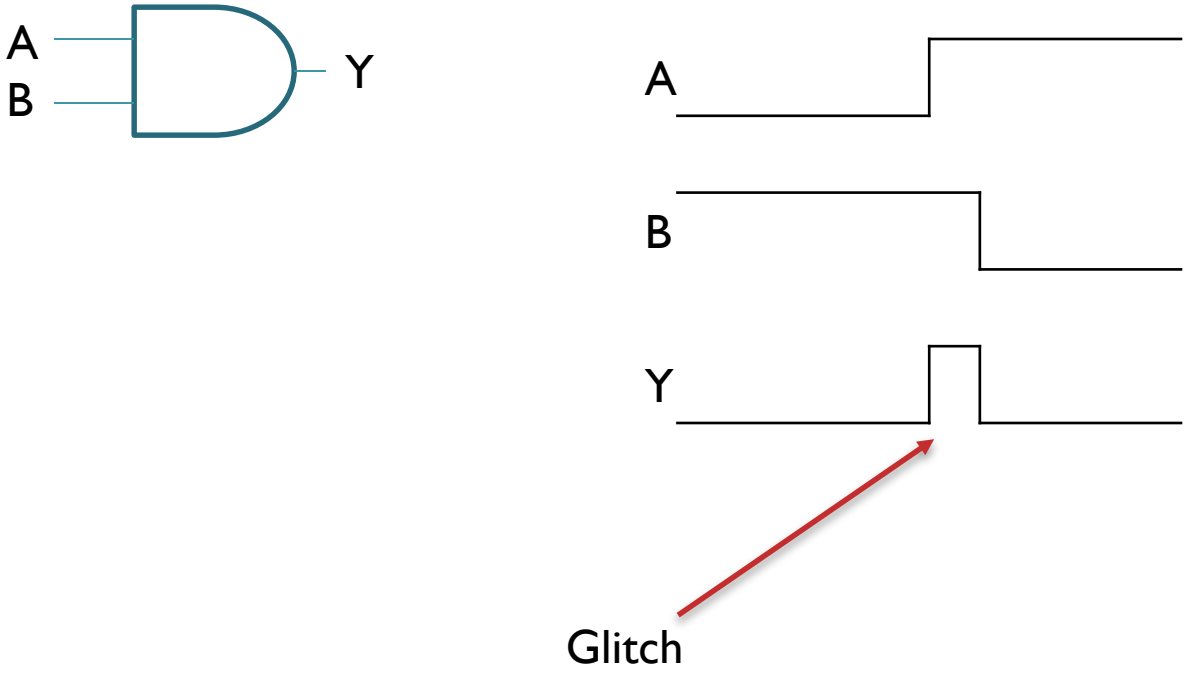
FSM VHDL example

- Outputs

Timing Diagrams and Logic Glitches

Simple Glitch

- A simple glitch can occur because not all signals change at the same time due to varying path lengths

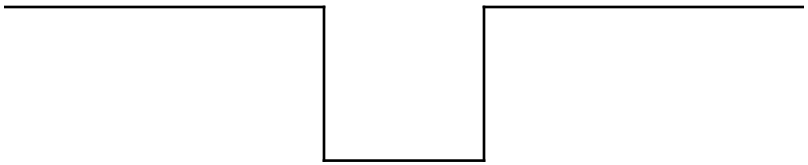


Definitions

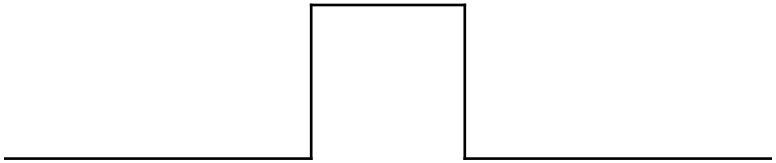
- Steady-State Behavior
  - Assumes inputs have been stable for a long time
- Transient Behavior
  - Takes into account circuit delays
- Hazards
  - An unintended change in output value
  - The digital circuit is **logically correct** but the output is incorrect

Static hazards

- Static-1 hazard
  - Output was expected to stay high, but glitches low momentarily



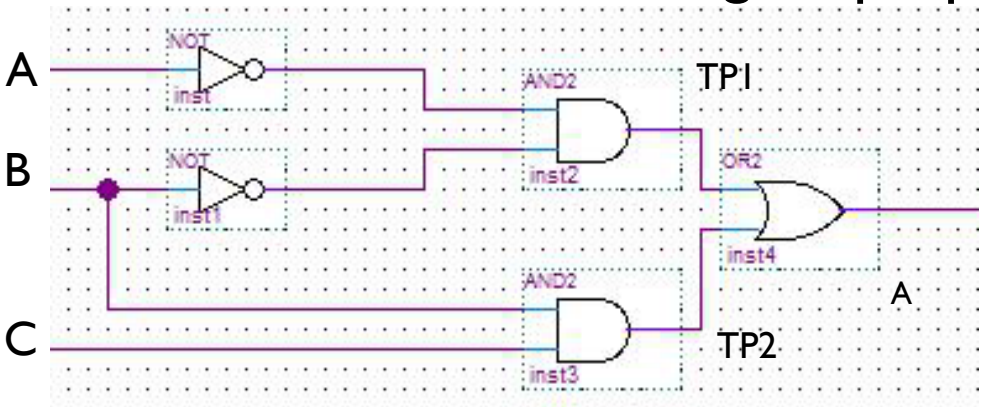
- Static-0 hazard
  - Output was expected to stay low but glitches high momentarily

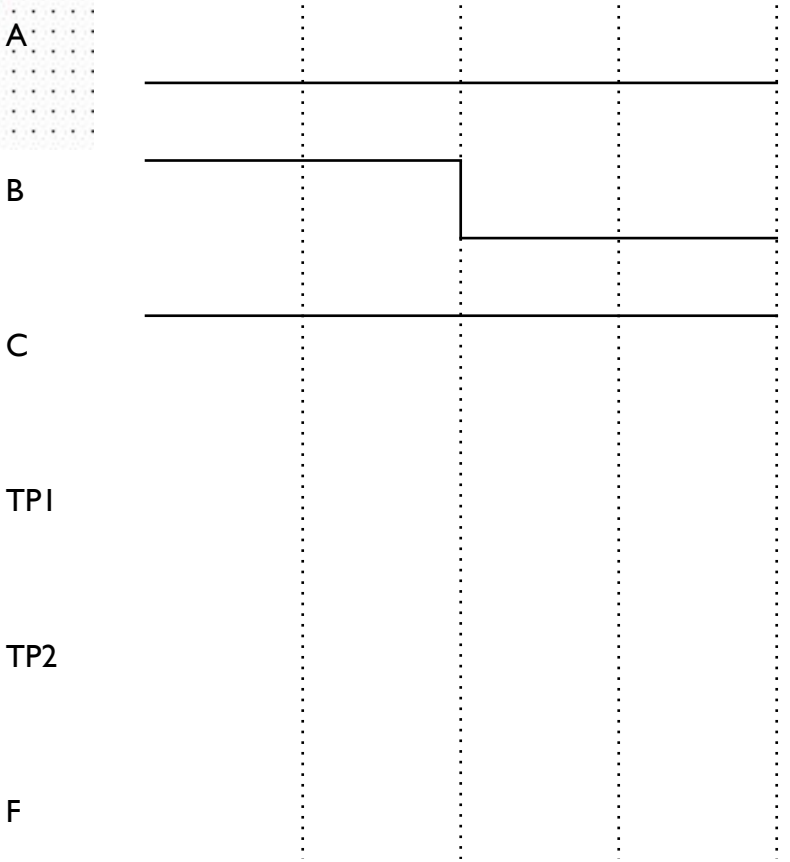


- Causes
  - When a single input has two different paths to the output and the paths are different lengths
  - Each path has a different prop delay

Static Hazard Example

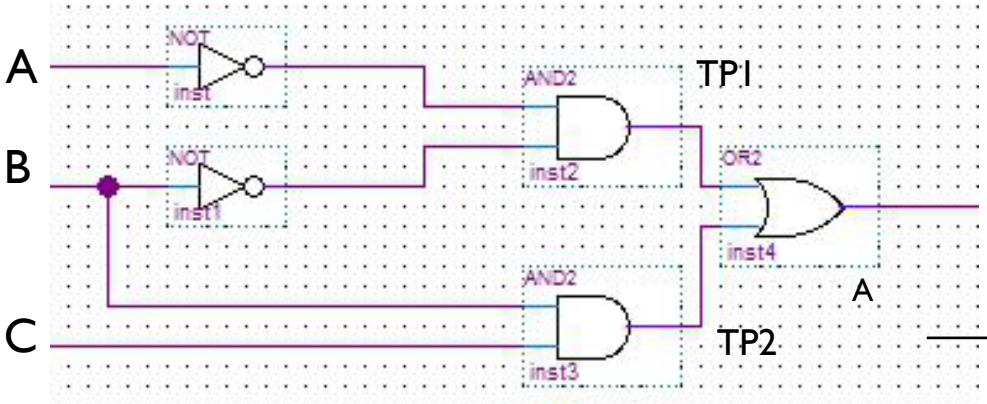
- Determine F assuming no prop delay

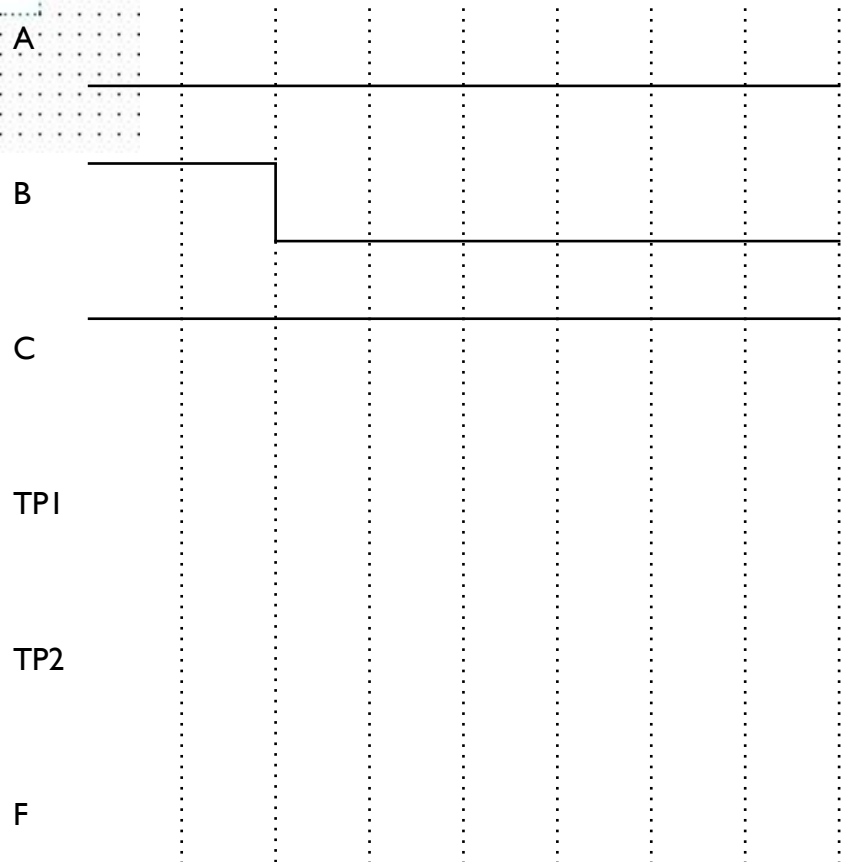




Static Hazard example

- Now add prop delays (assume 1 unit)





Why do hazards Matter?

- Often the output of one circuit is the input of another
  - Input could be read at a precise moment in time
  - Input could drive clocked logic
  - Output could be controlling logic such as a traffic light