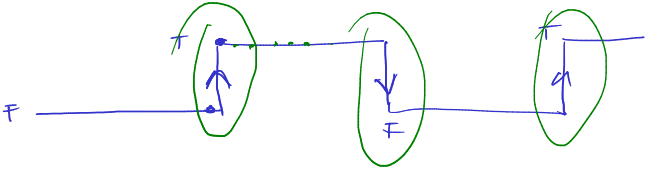
Embedded System Design and Modeling

IV. FSM Patterns

FSM Patterns

- **Design patterns**: reusable templates which appear often in applications
- Patterns
  - Operating on signal transitions
  - Debounce (one-sided, two-sided)

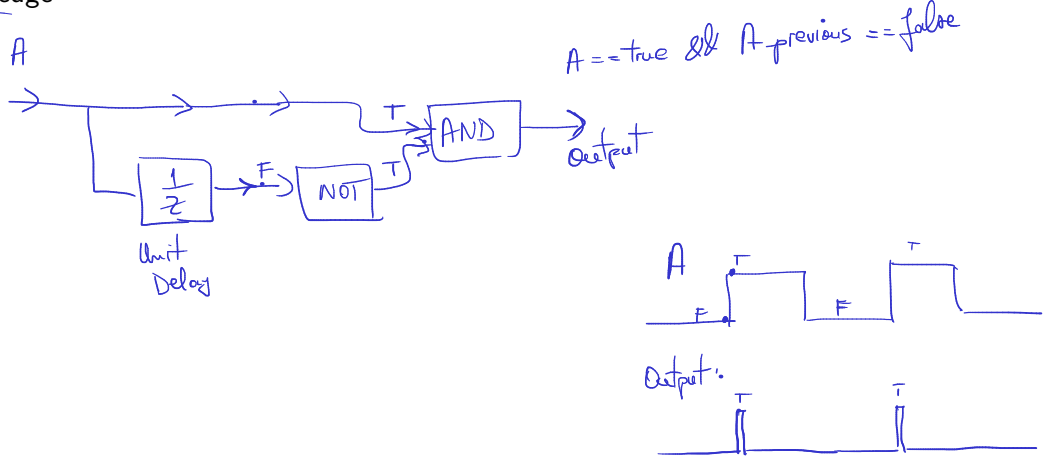
Operating on signal transitions



- For boolean signals/conditions
- Use when information is in the signals' **fronts** (edge / transition) rather than in its values
- Solution: detect signal transitions
  - rising edge
  - falling edge
  - both

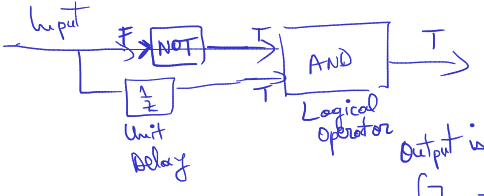
Detect rising edge

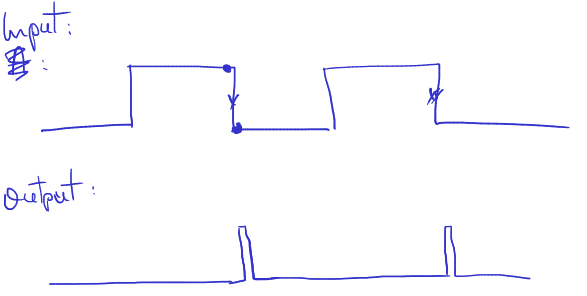
- Draw here: detect rising edge



Detect falling edge

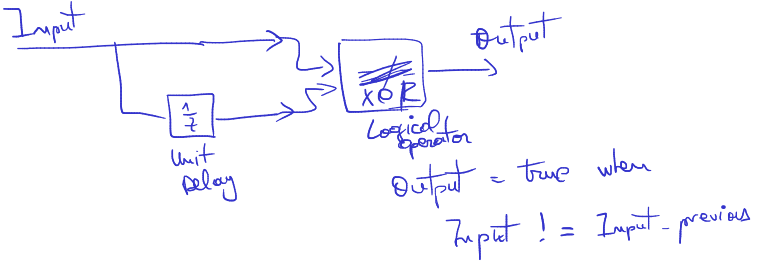
- Draw here: detect falling edge





Detect any edge

- Draw here: detect any edge
-
-
-
-



Debouncing

- For boolean signals/conditions
- Bouncing: real signals look like this:

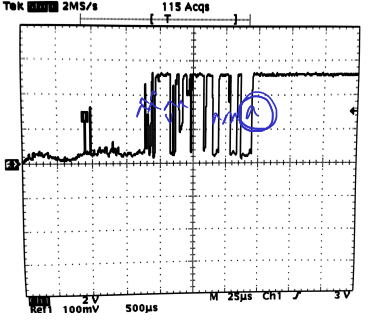
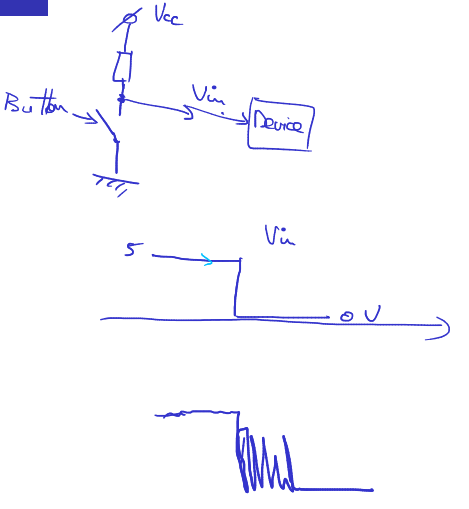


Figure 1: Signal change when pushing a button

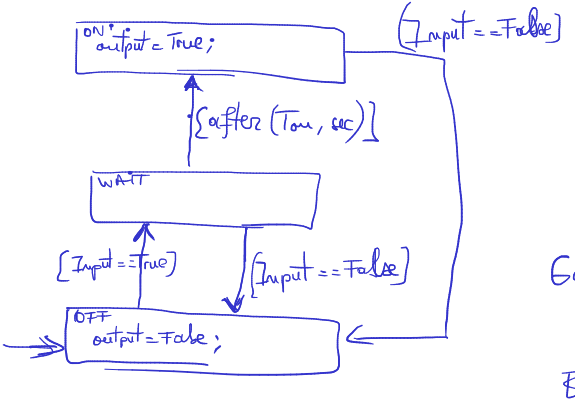


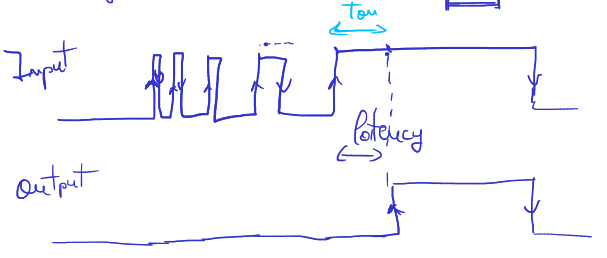
- Use debouncing to avoid spurious transitions

Debouncing rising edge

When you have a rising edge, only consider it if the signal stays on True for at least T_{on} = 200ms

- Draw here: debounce rising edge



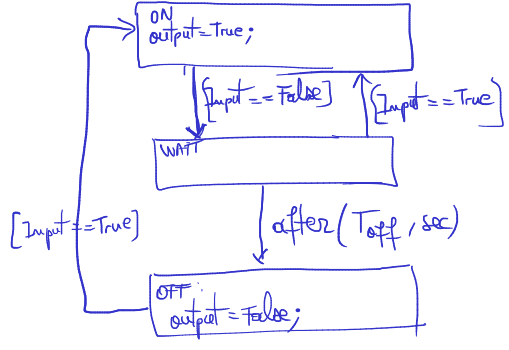


Good: Clean all false starts of length < Ton

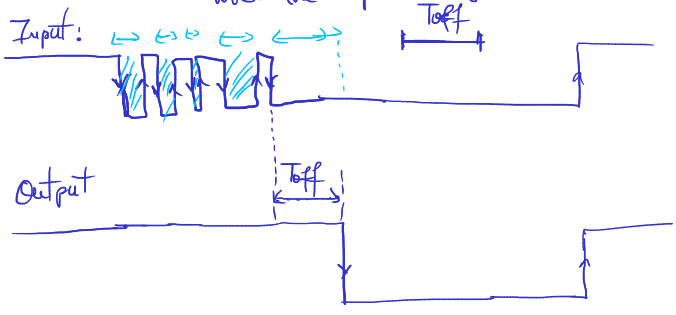
Bad:

Debouncing falling edge

- Draw here: debounce falling edge



When you have a falling edge in the input, only trust it when the signal stays down for at least Toff



Debouncing both edges

- Draw here: debounce both edges