



CLOUD V

2

## Goals

1. Clone a repository
2. Create a Verilog module from a cloned repository
3. Simulate a Verilog module using a self-checking testbench



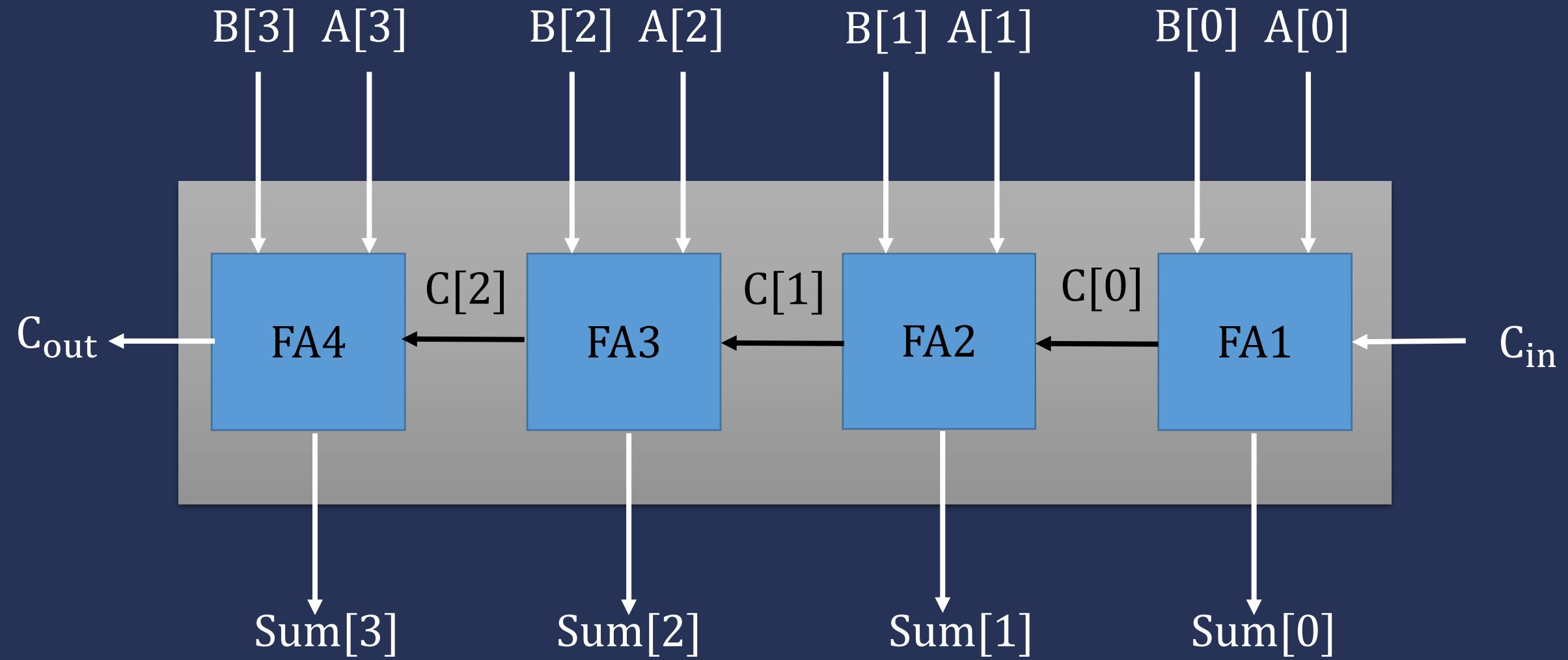
CLOUD V

1

## Goals

1. Register into CloudV
2. Log in CloudV
3. Create a Verilog module
4. How to simulate a Verilog module

# 4-bit Ripple Carry Adder



# Self-checking tesbench

- A testbench that checks for itself

Why?

- A  $0 - 2^4$
- B  $0 - 2^4 \geq 16 \times 16 = 256 \times 2 = 512$





CLOUD V



CLOUD V

3

## Goals

1. A BCD counter
2. A BCD to 7-segment decoder
3. Implement the code of a 7-segment decoder on the TinyFPGA



CLOUD V

## BCD (Binary coded decimal)

System where each decimal digit is represented by its binary equivalent.

$$5 \rightarrow 0101$$



CLOUD V

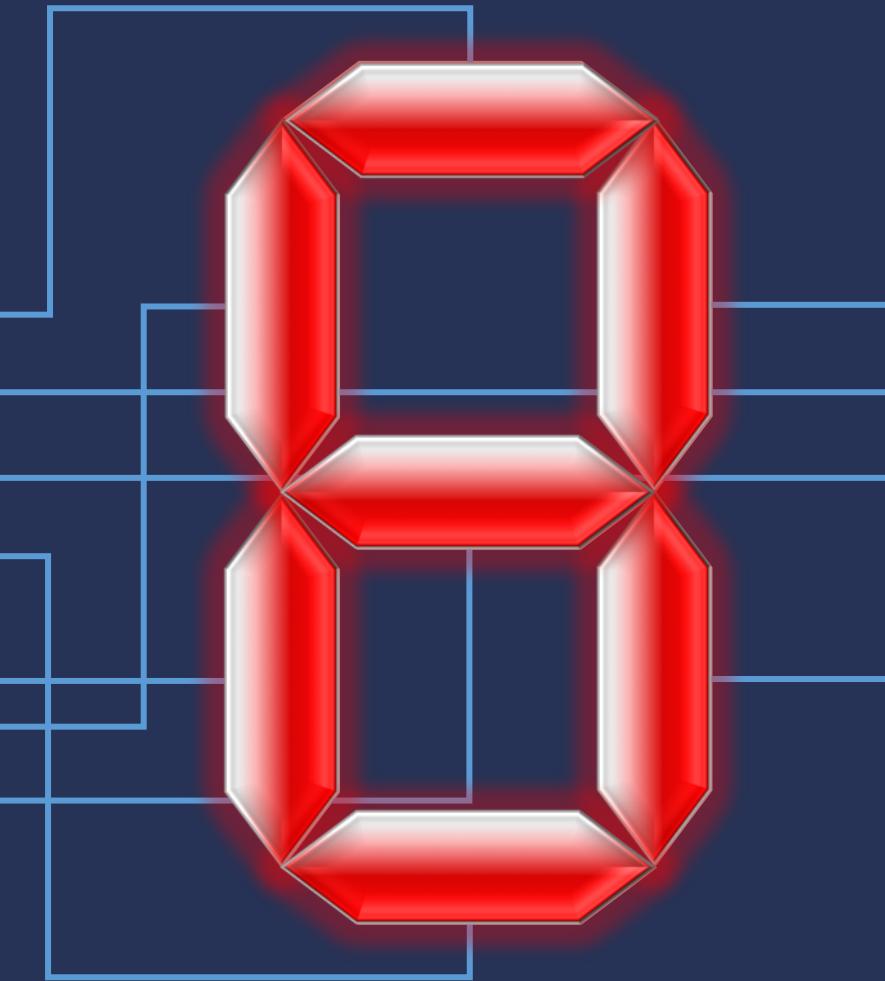




CLOUD V

$b_0$   
 $b_1$   
 $b_2$   
 $b_3$

7-segment  
decoder

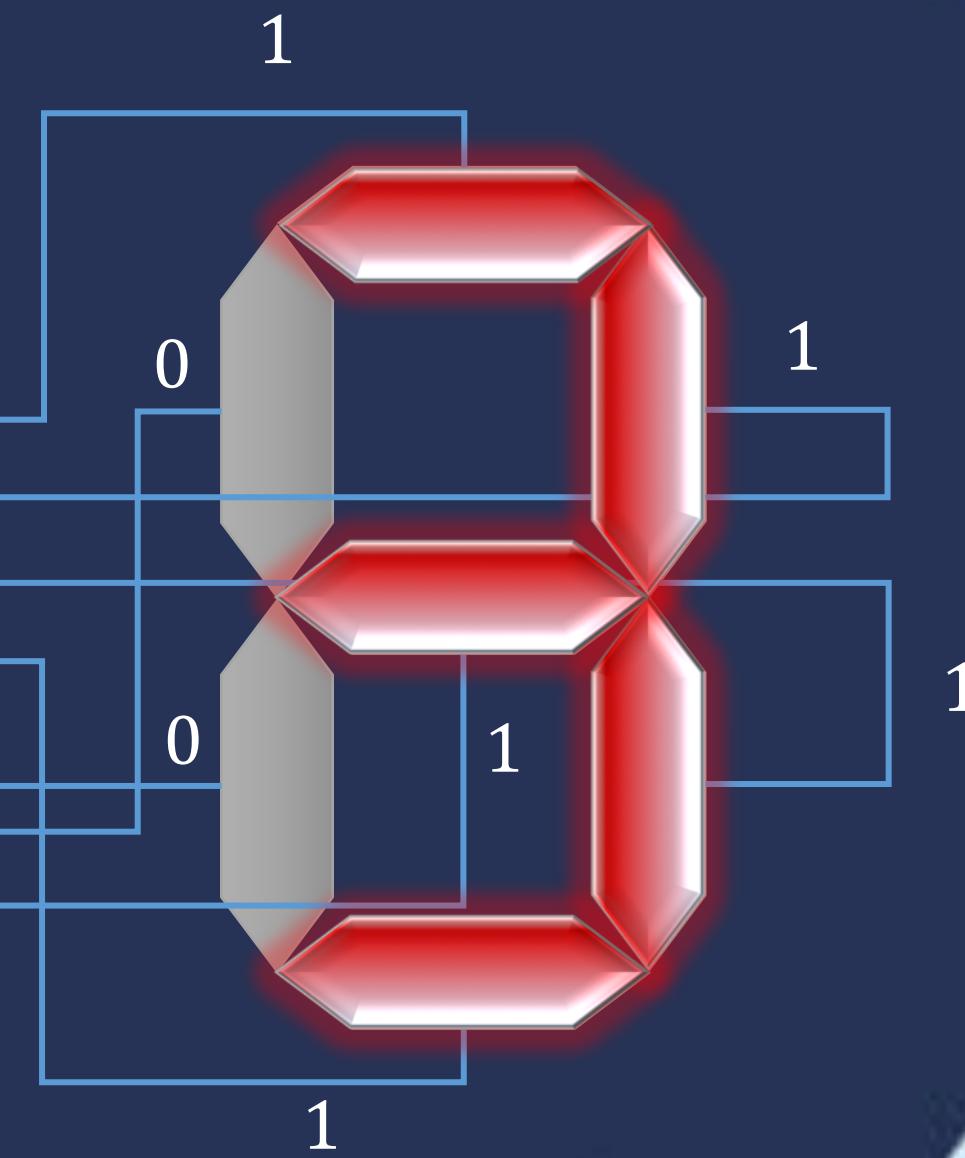




CLOUD V

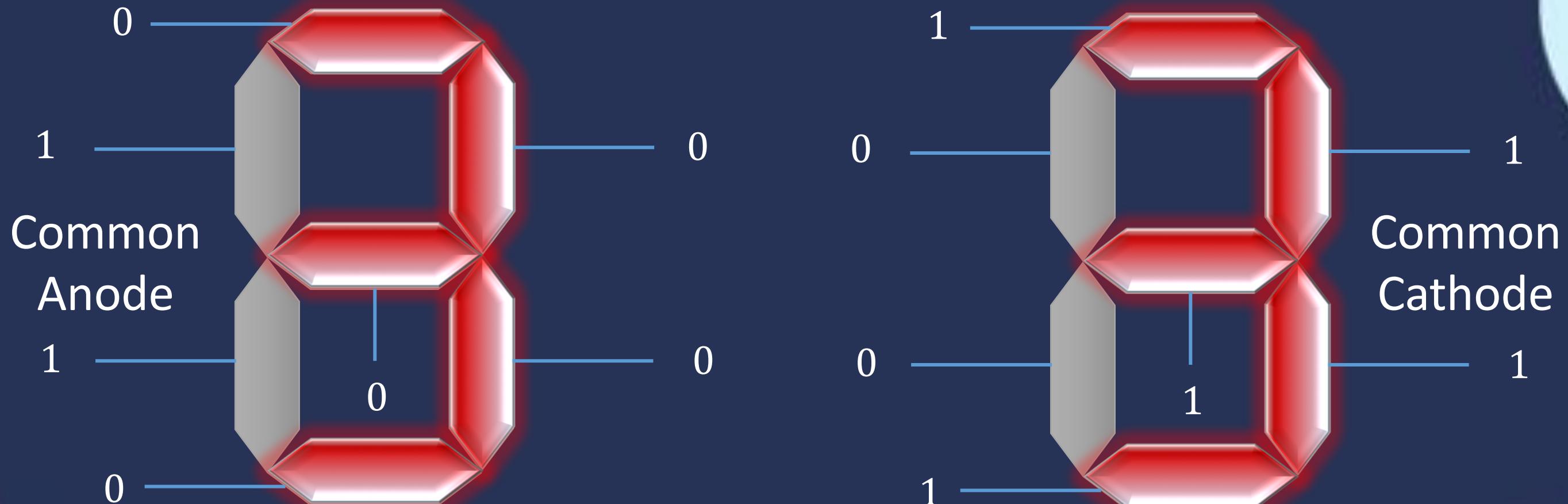
1  
1  
0  
0

7-segment  
decoder





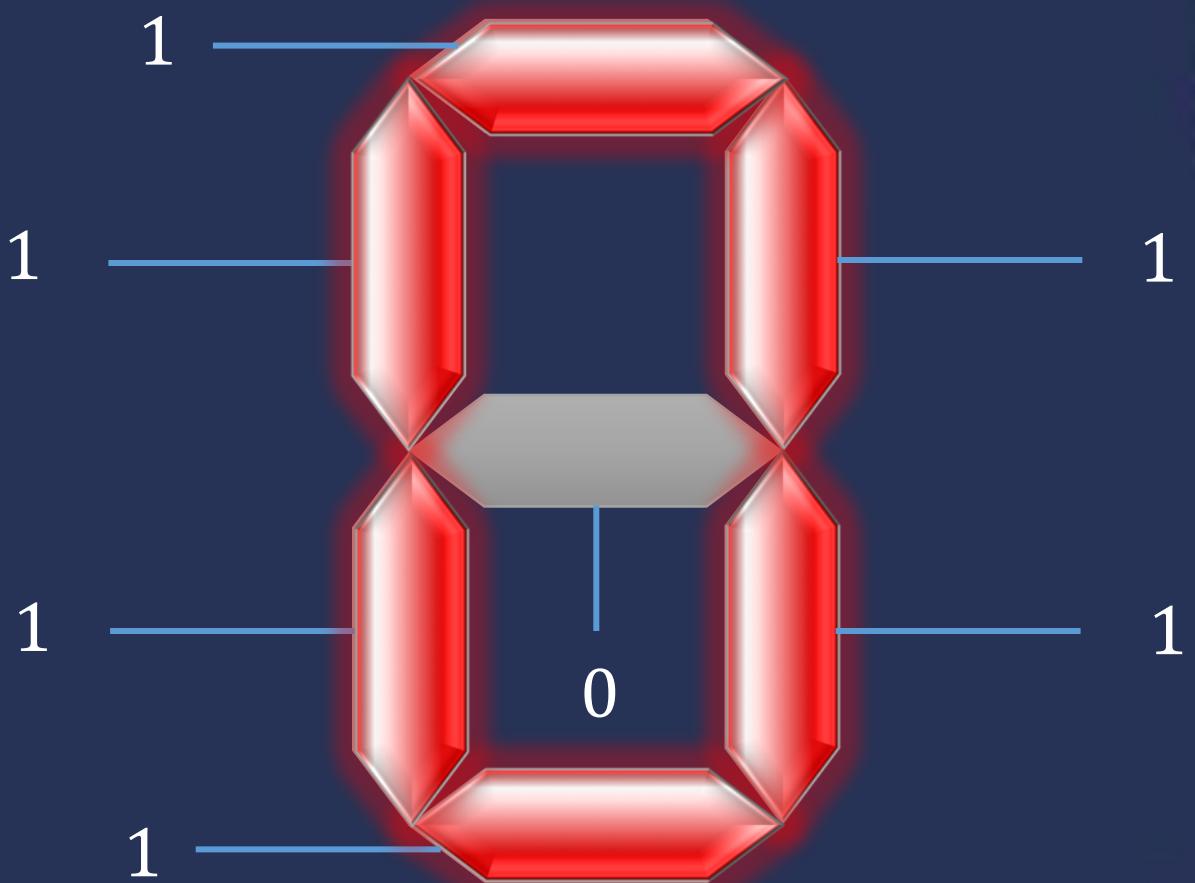
CLOUD V





CLOUD V

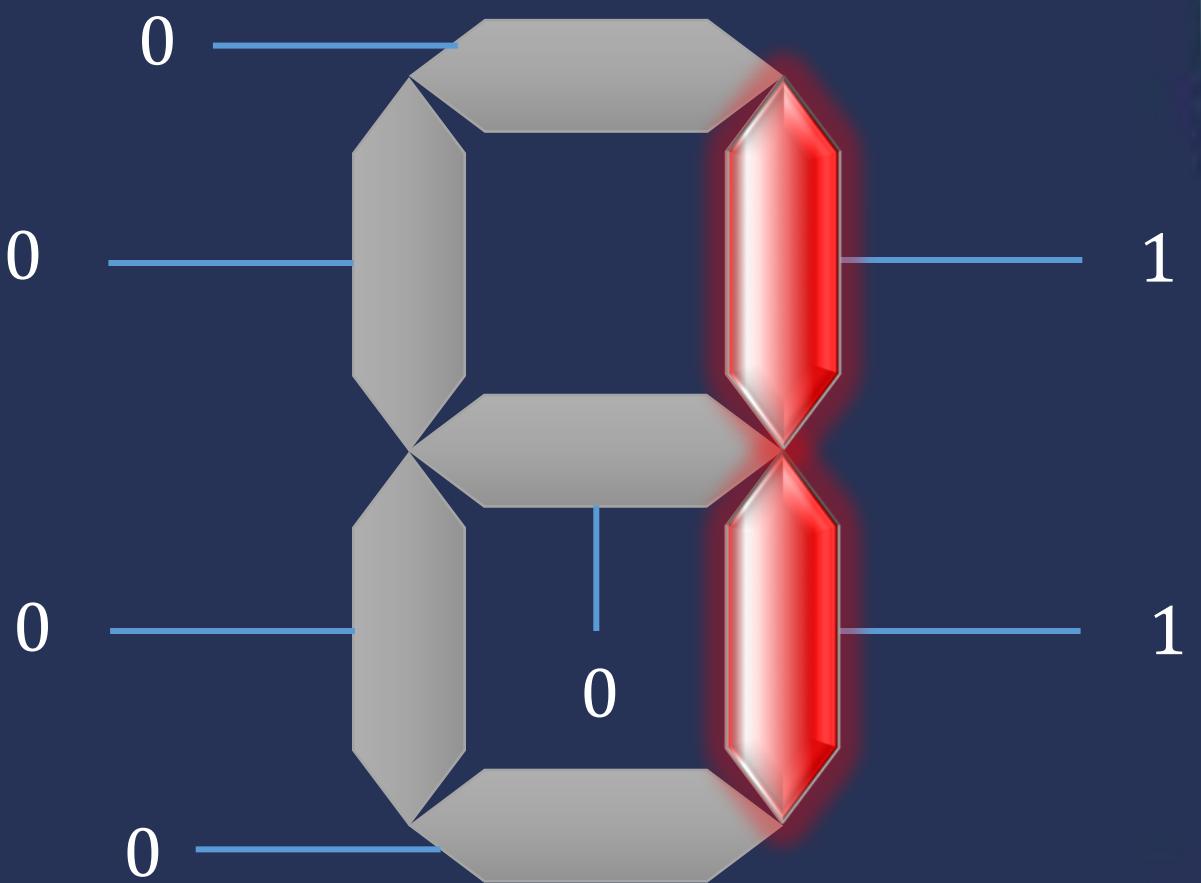
0





CLOUD V

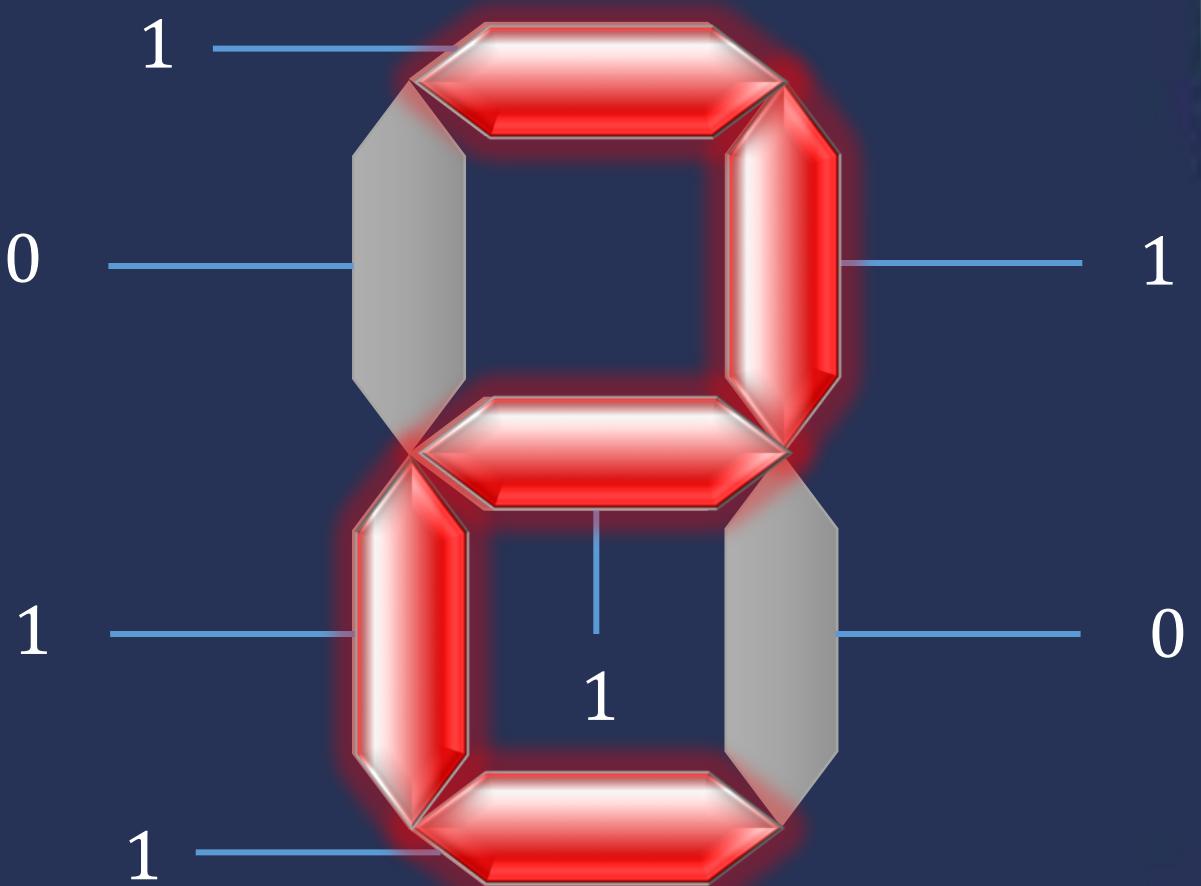
1





CLOUD V

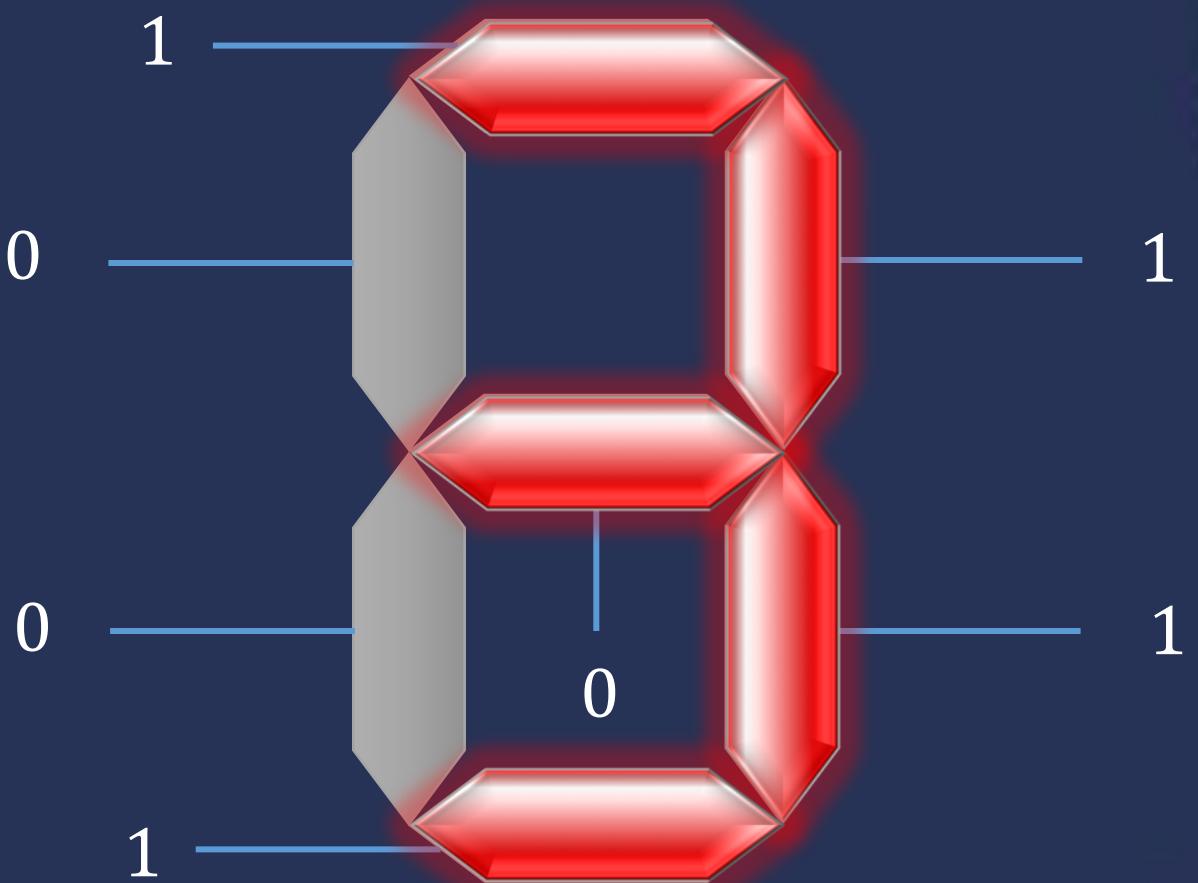
2





CLOUD V

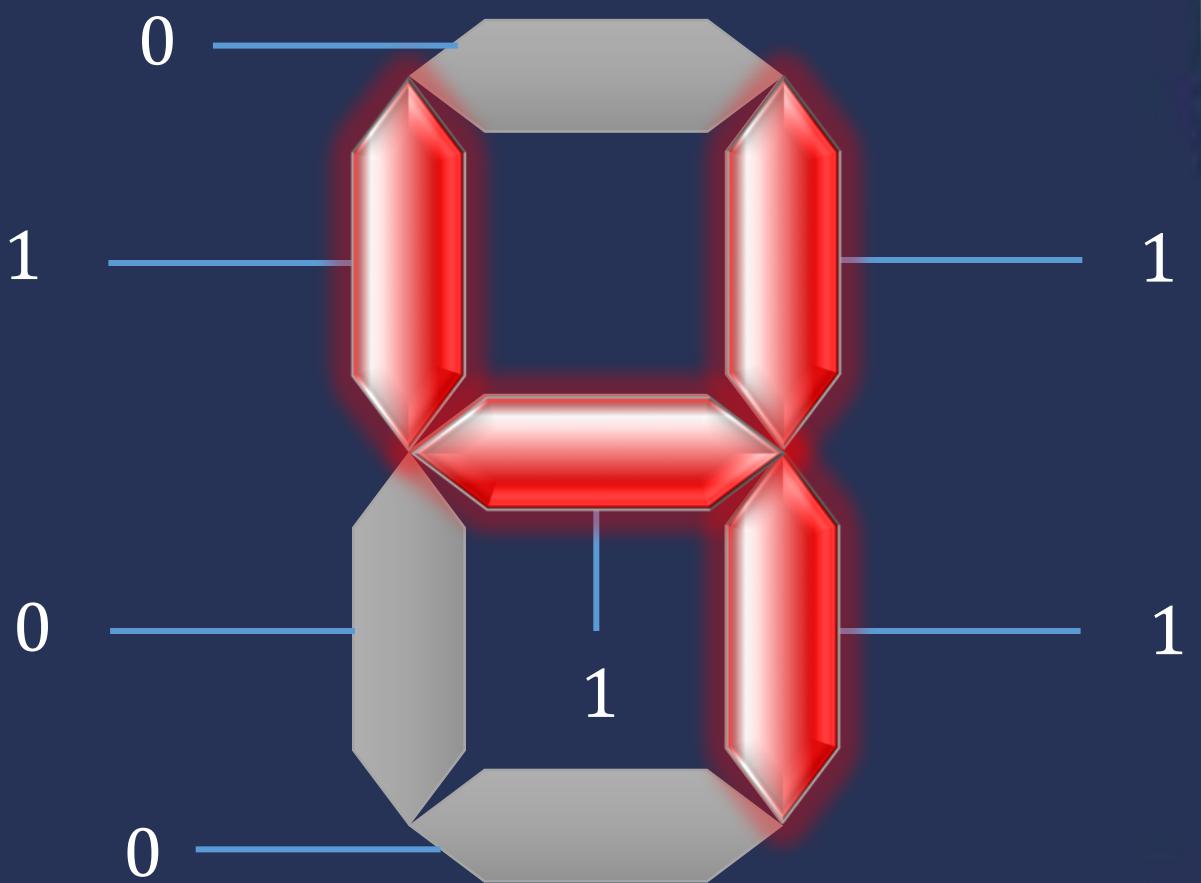
3





CLOUD V

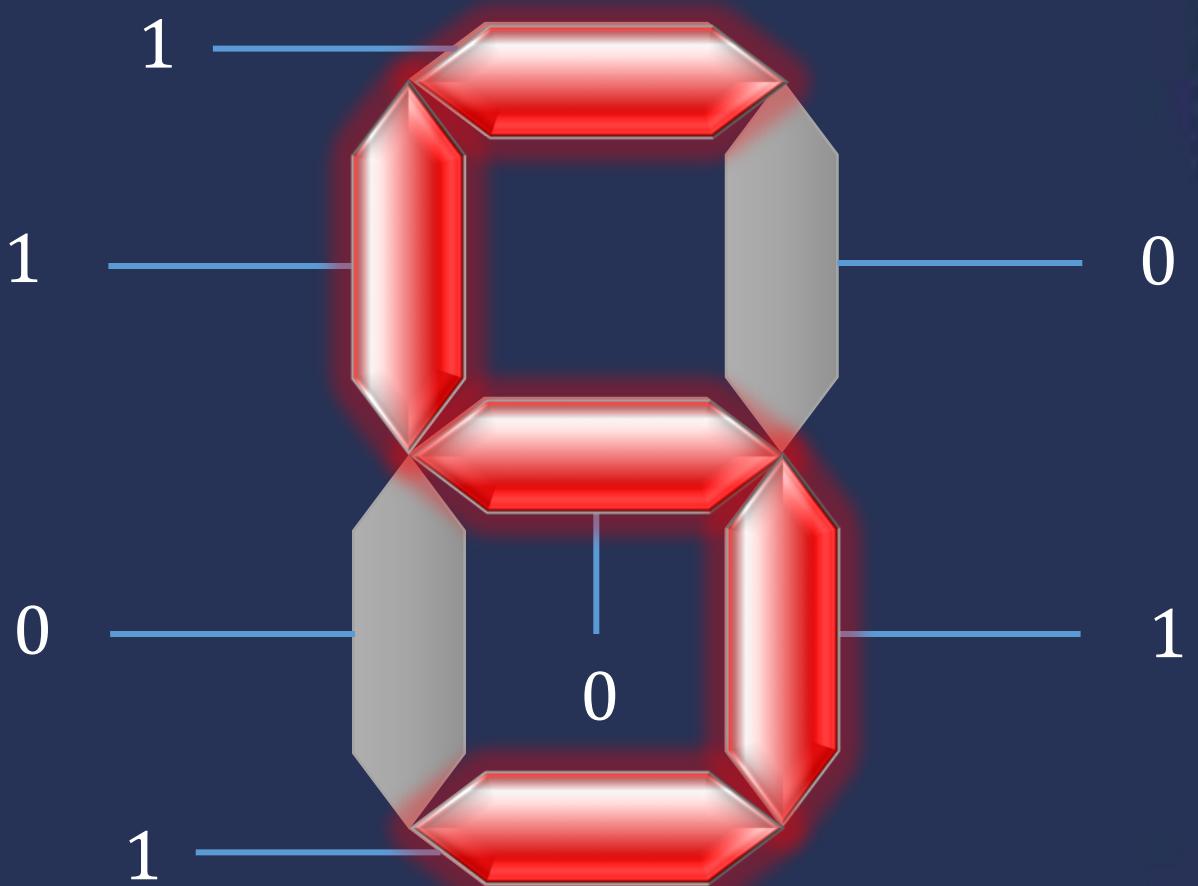
4





CLOUD V

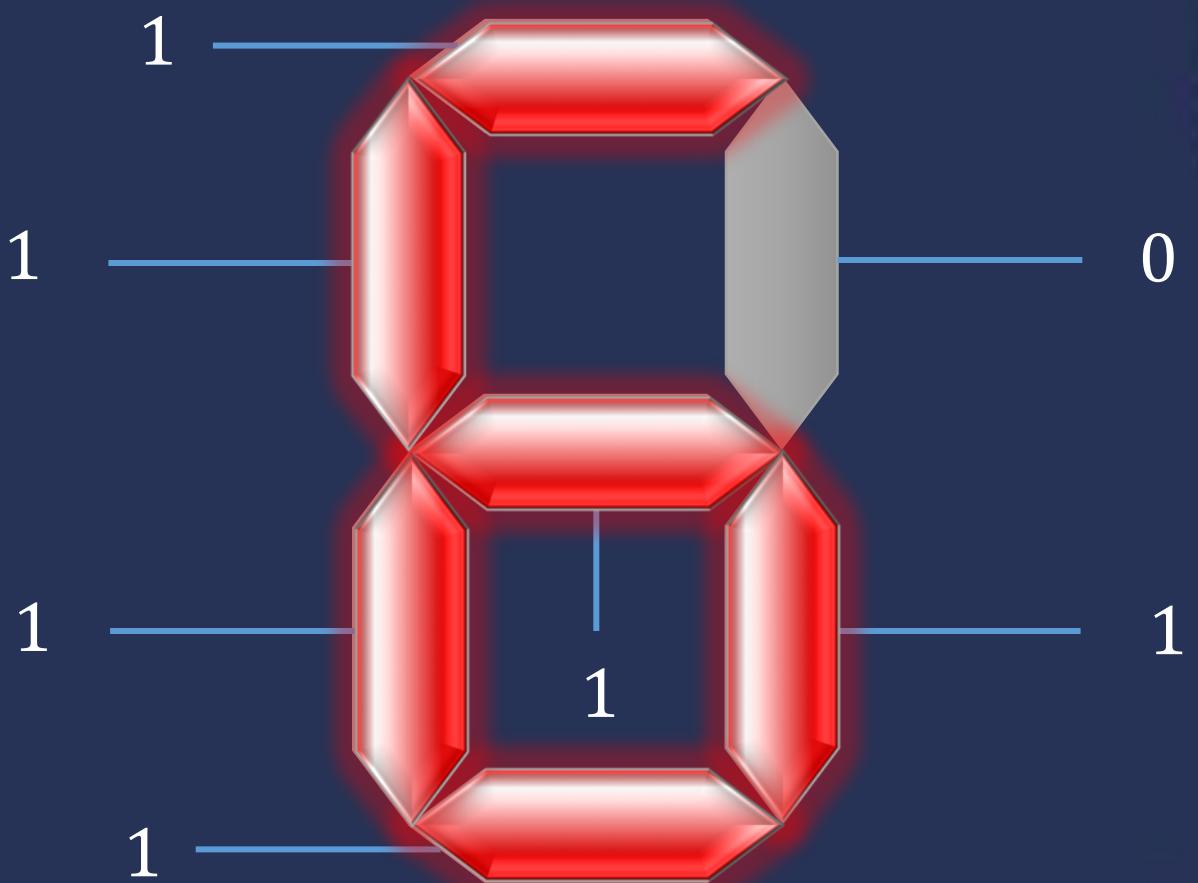
5





CLOUD V

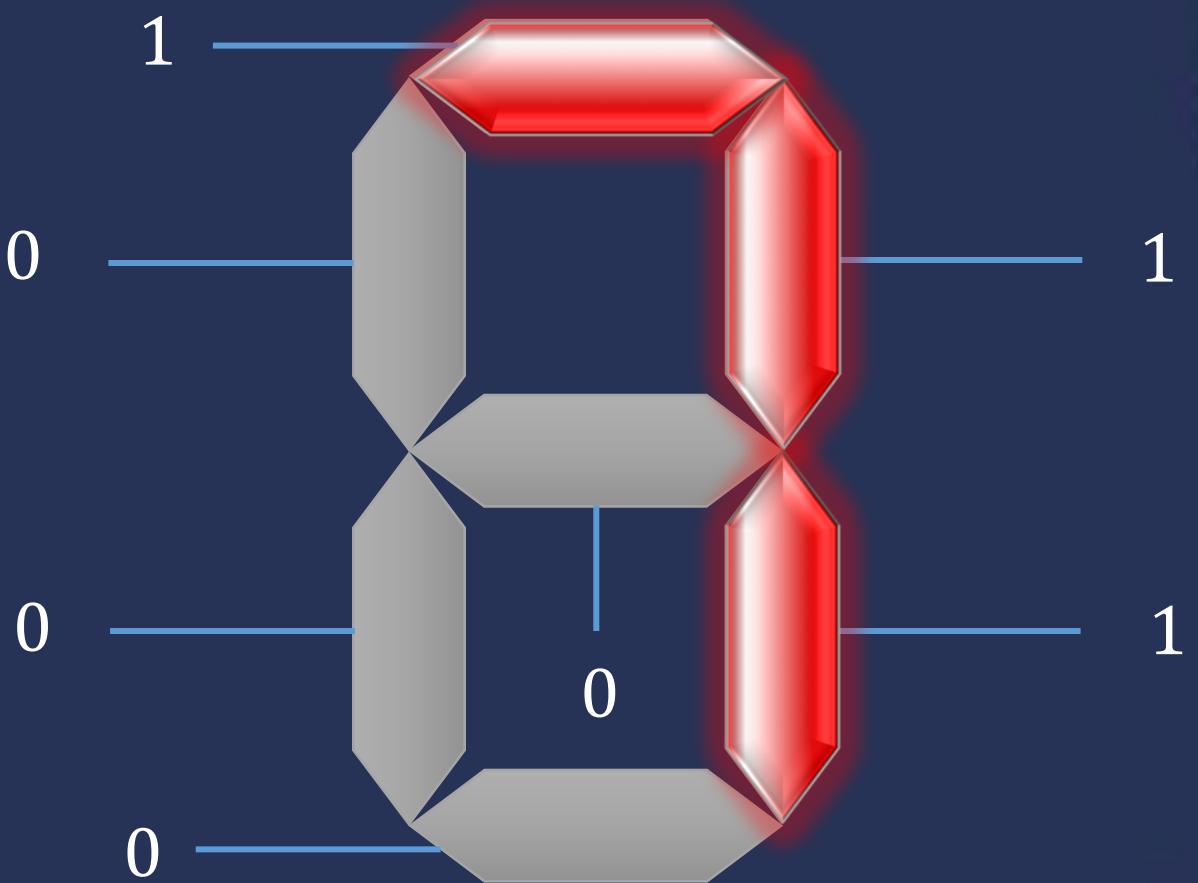
6





CLOUD V

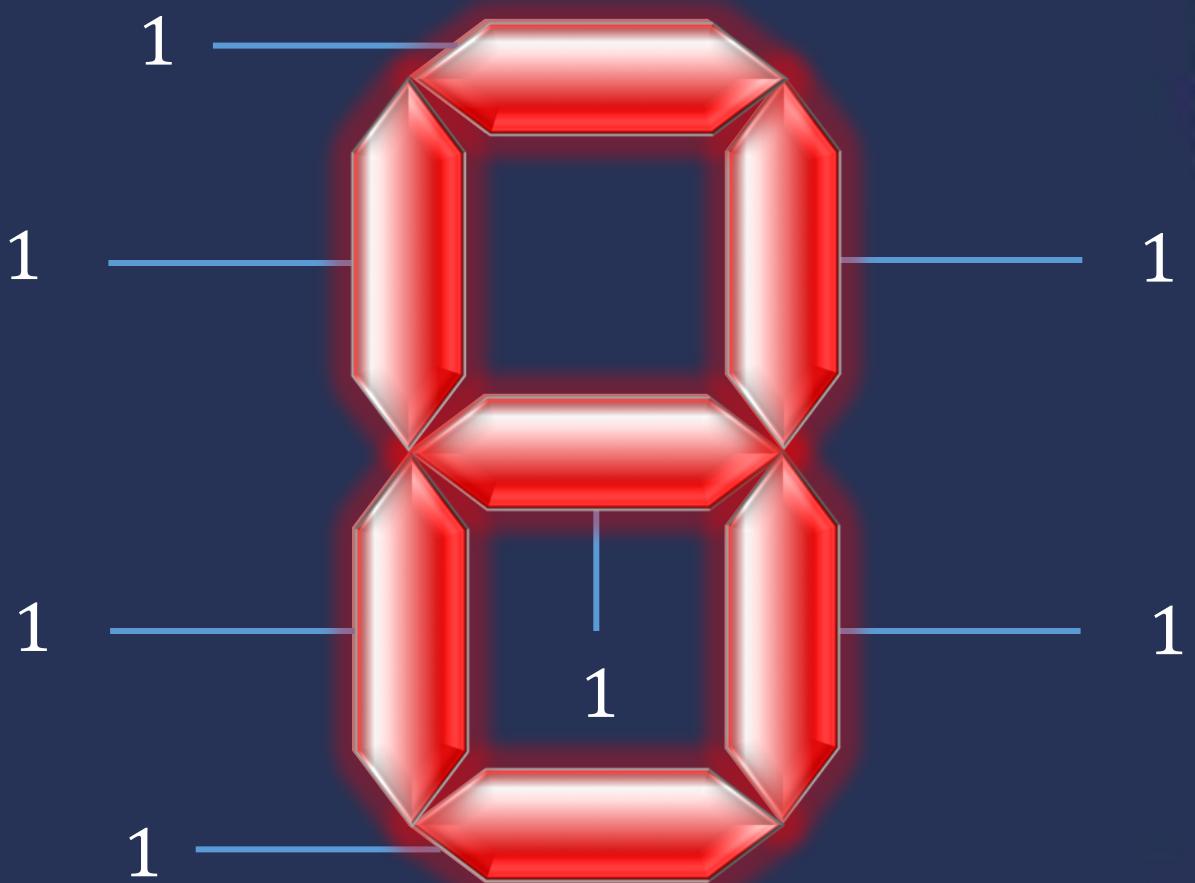
7





CLOUD V

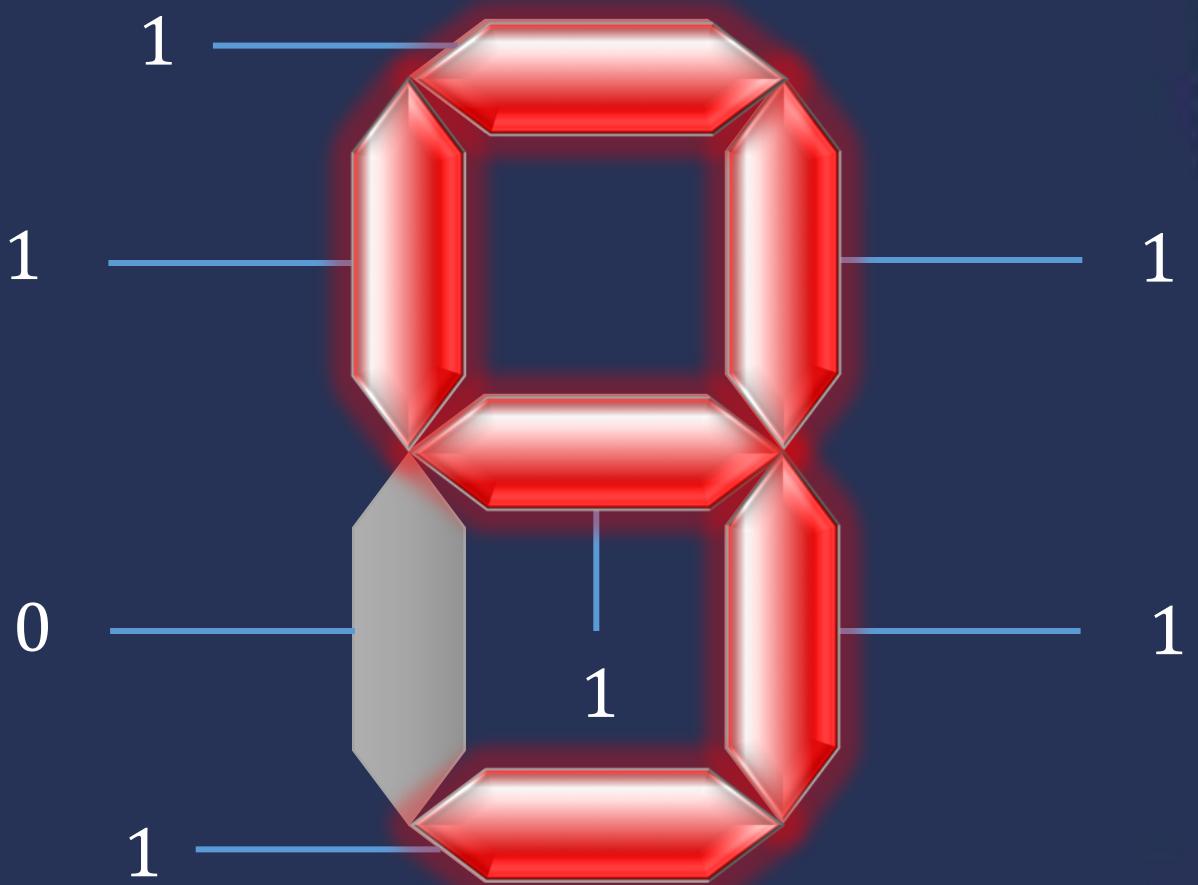
8

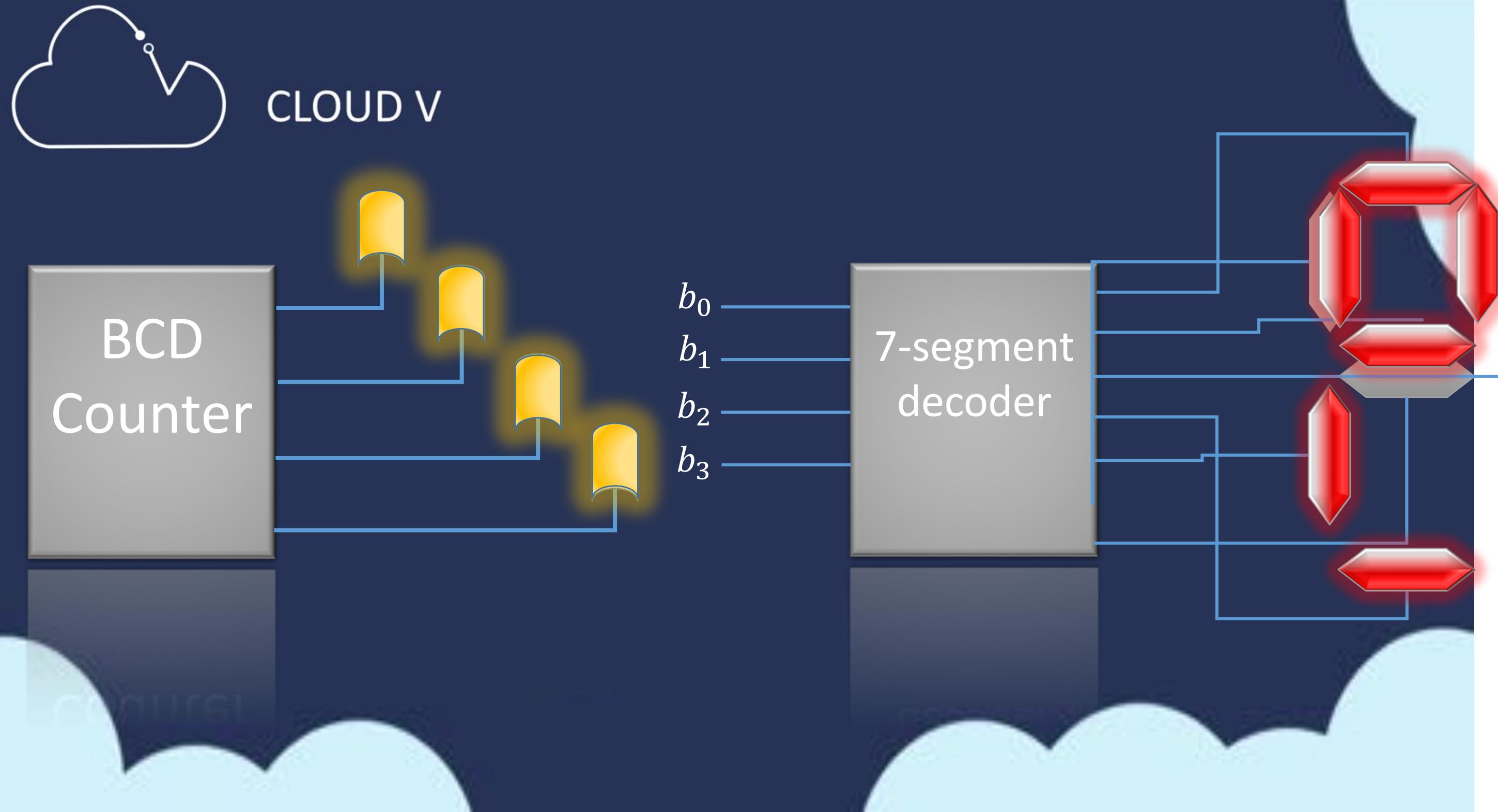




CLOUD V

9







## Goals

1. Modify the BCD counter to support
  - Decrementing / incrementing
2. Input from real world
  - Edge detector
  - Synchronizer
  - Debouncer



CLOUD V

3

## Goals

1. TinyFPGA BX
2. Blink an LED using TinyFPGA.
  - Create a constraints file
  - Generate a bitstream file
  - Program the TinyFPGA



CLOUD V

### TinyFPGA BX:

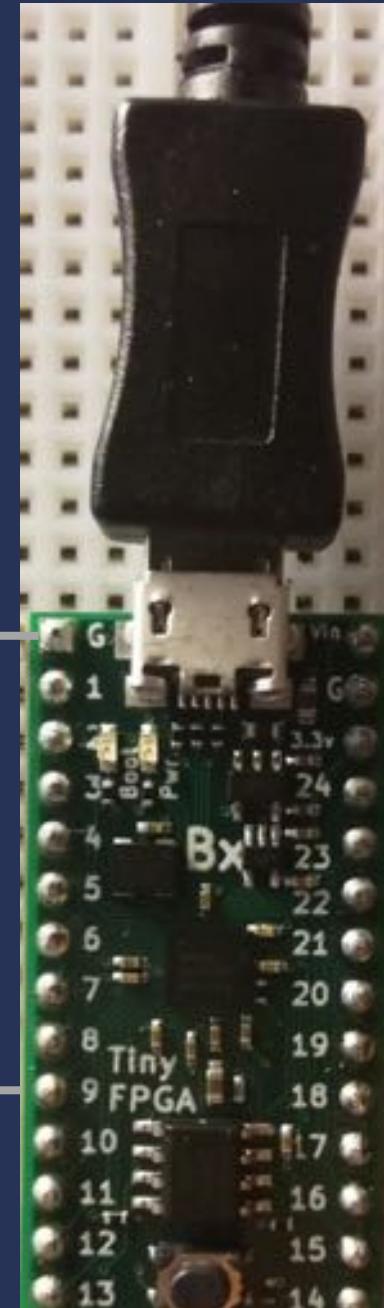
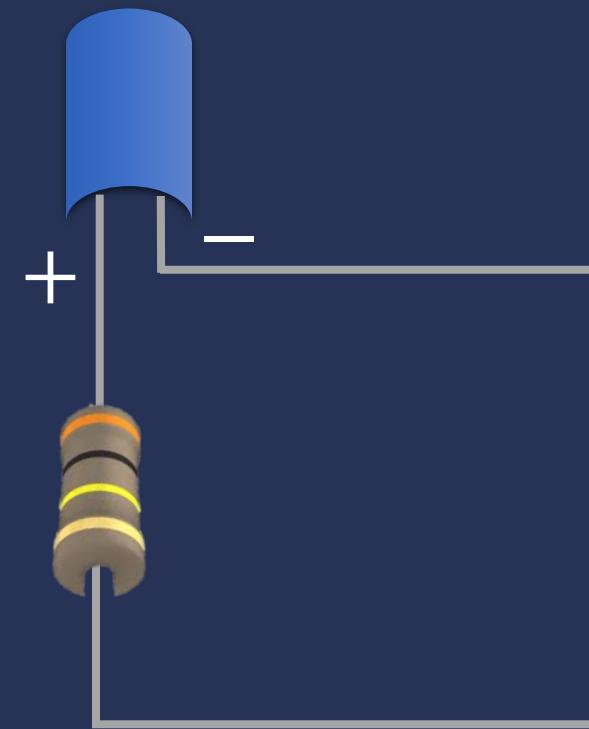
- Small
- Cheap
- Clock source of 16 MHz on port B2





CLOUD V

## LED Circuit

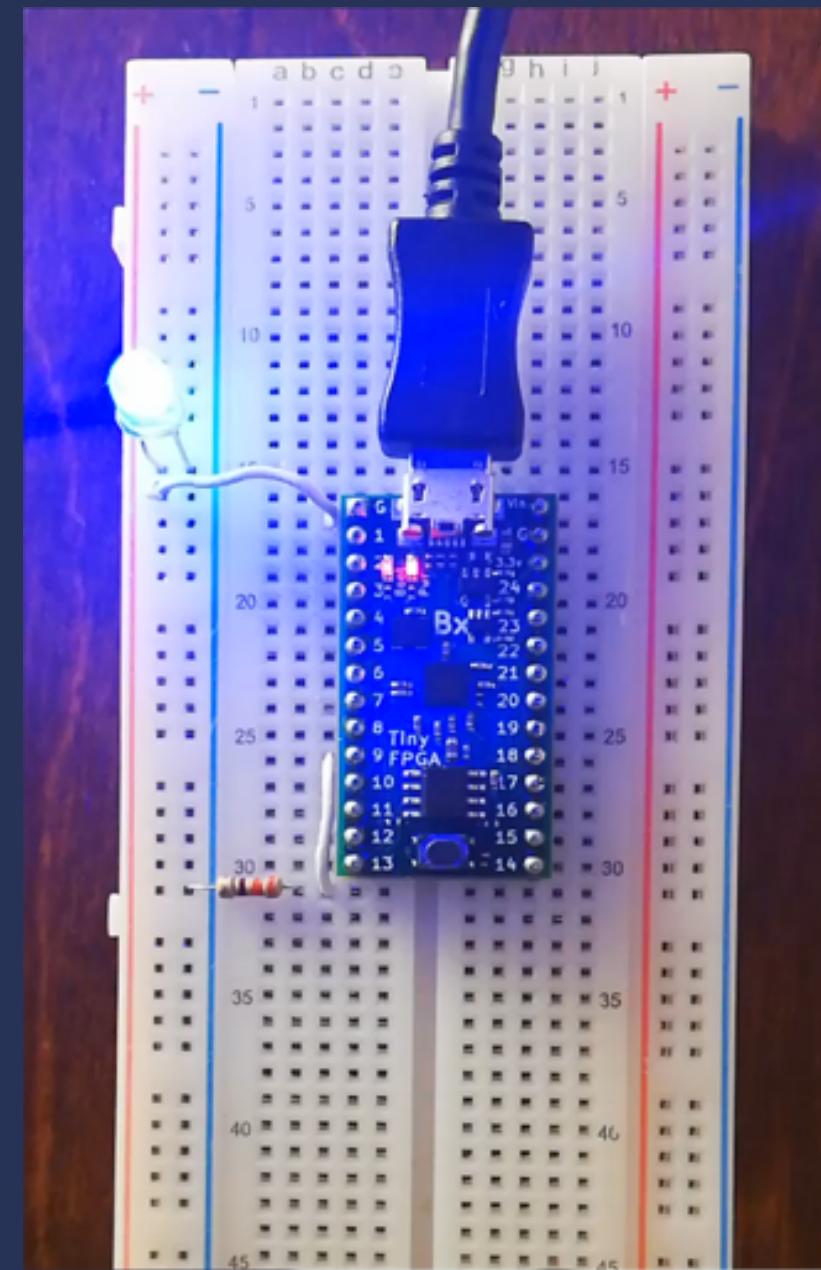




CLOUD V

### TinyFPGA BX:

- Clock source of 16 MHz on port B2





CLOUD V

CLK



Counter

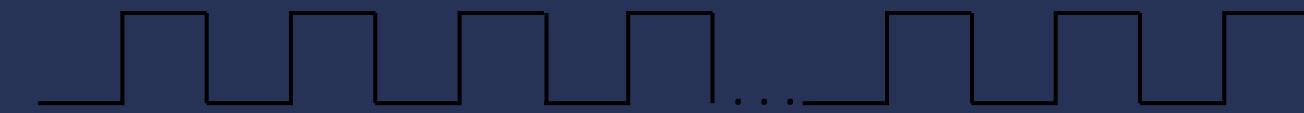
1 2 3 4

16,000,000

LED

← →  
1 Second

CLK



Counter

1 2 3 4

16,000,000

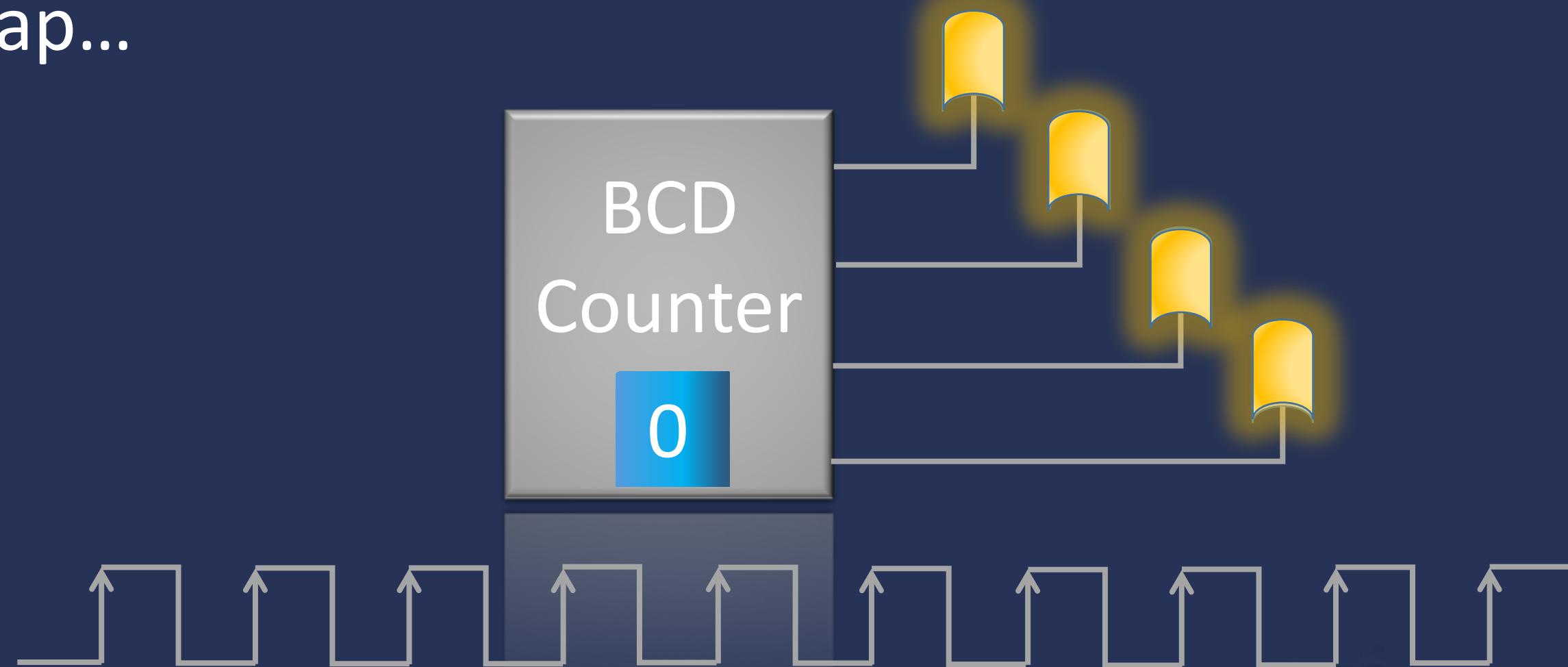
LED

← →  
1 Second



CLOUD V

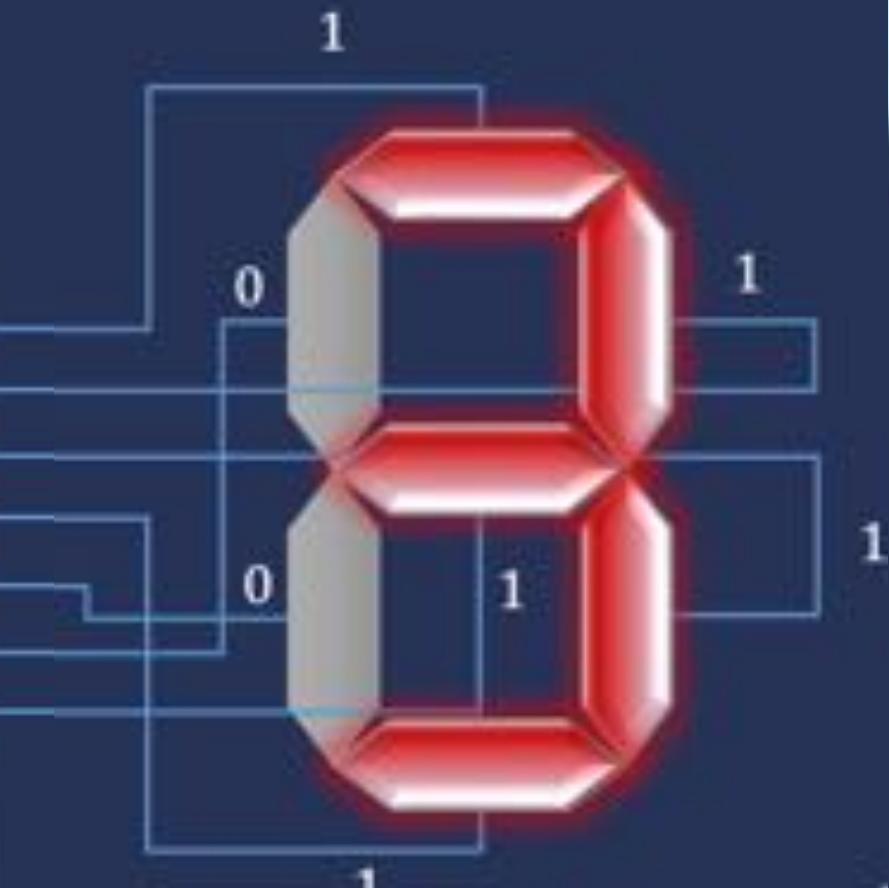
Recap...





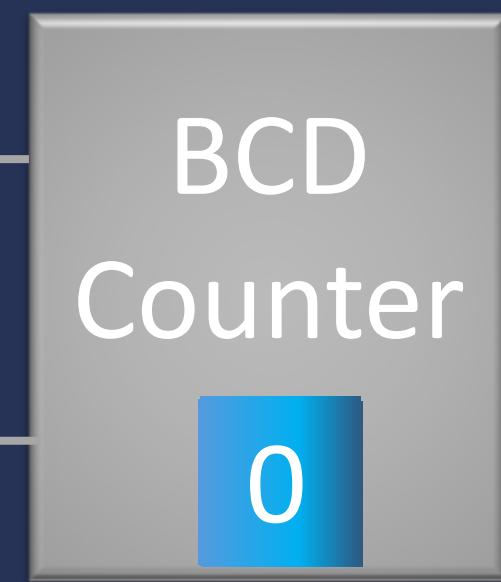
CLOUD V

Clock  
16 MHz





CLOUD V



conusel



CLOUD V

# Metastability

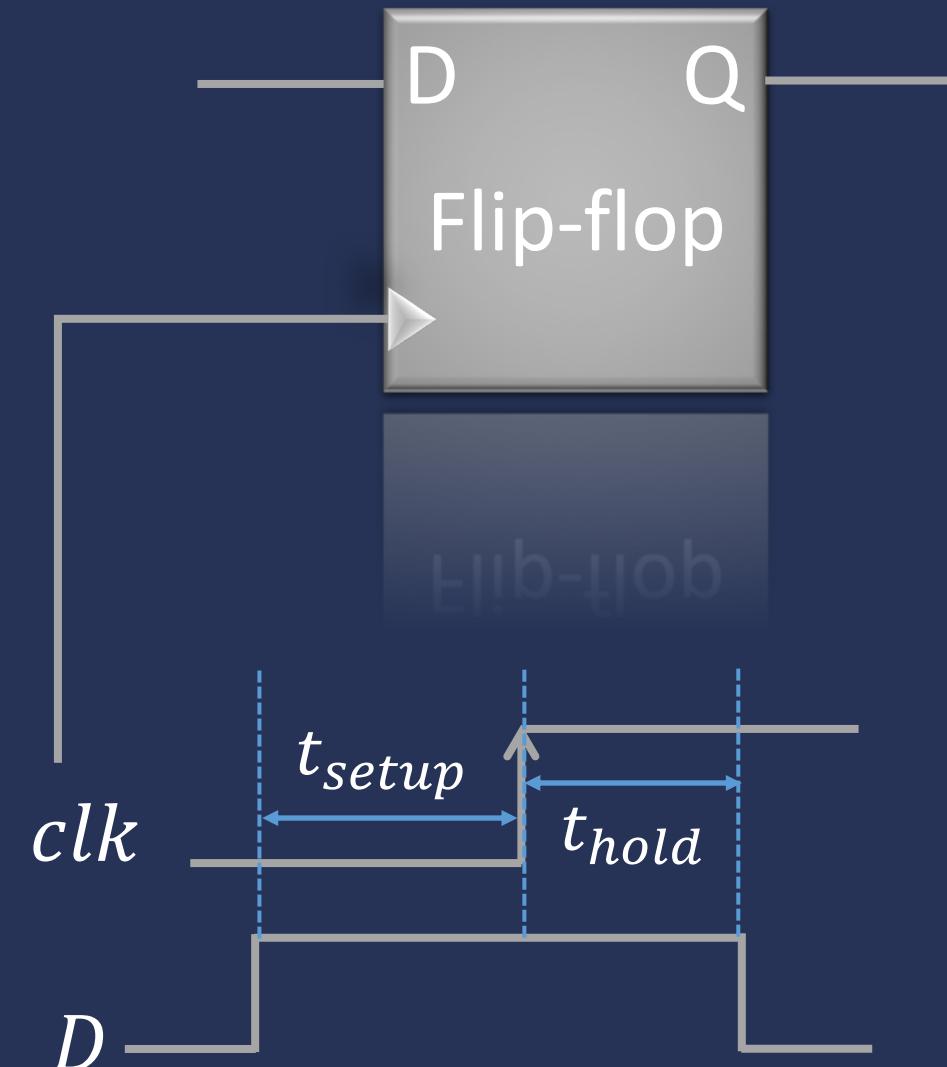
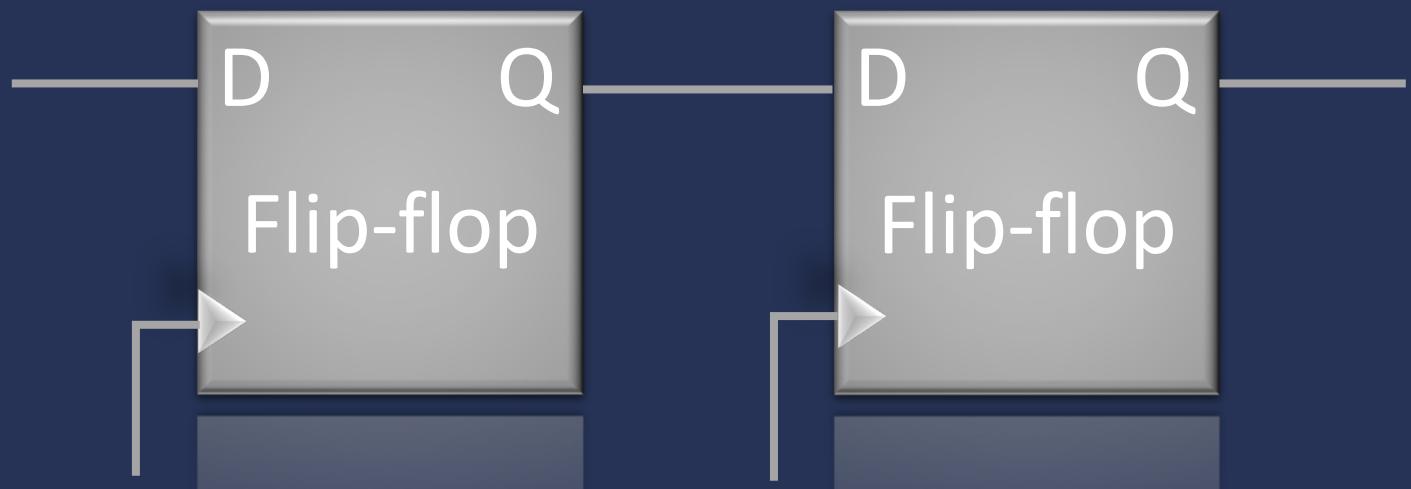
When timing constraints are not met :

- Q: random value 0 or 1 regardless the value of D
- Q: value between 0 and 1
- Q: oscillate between 0 and 1 very fast



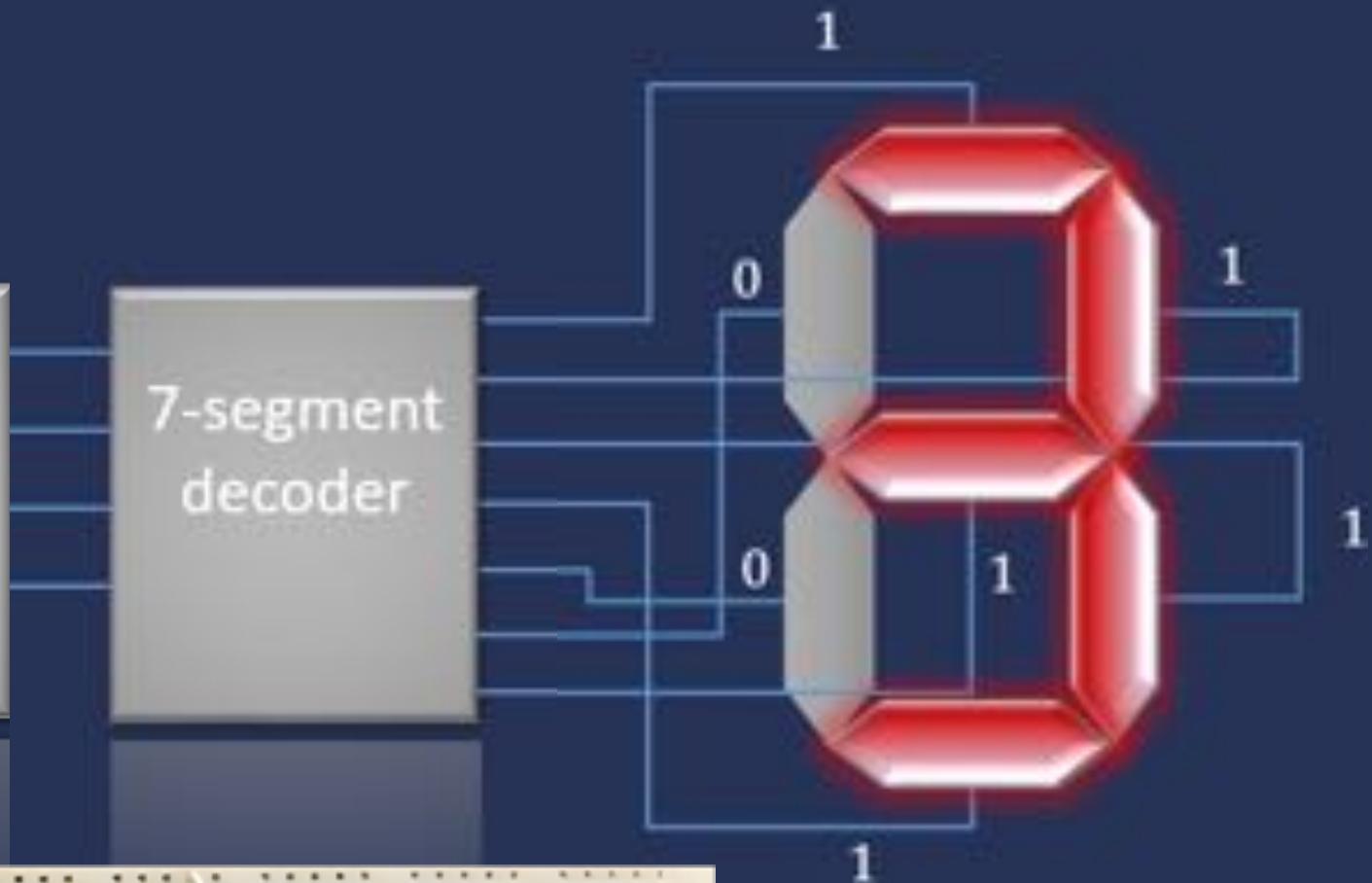
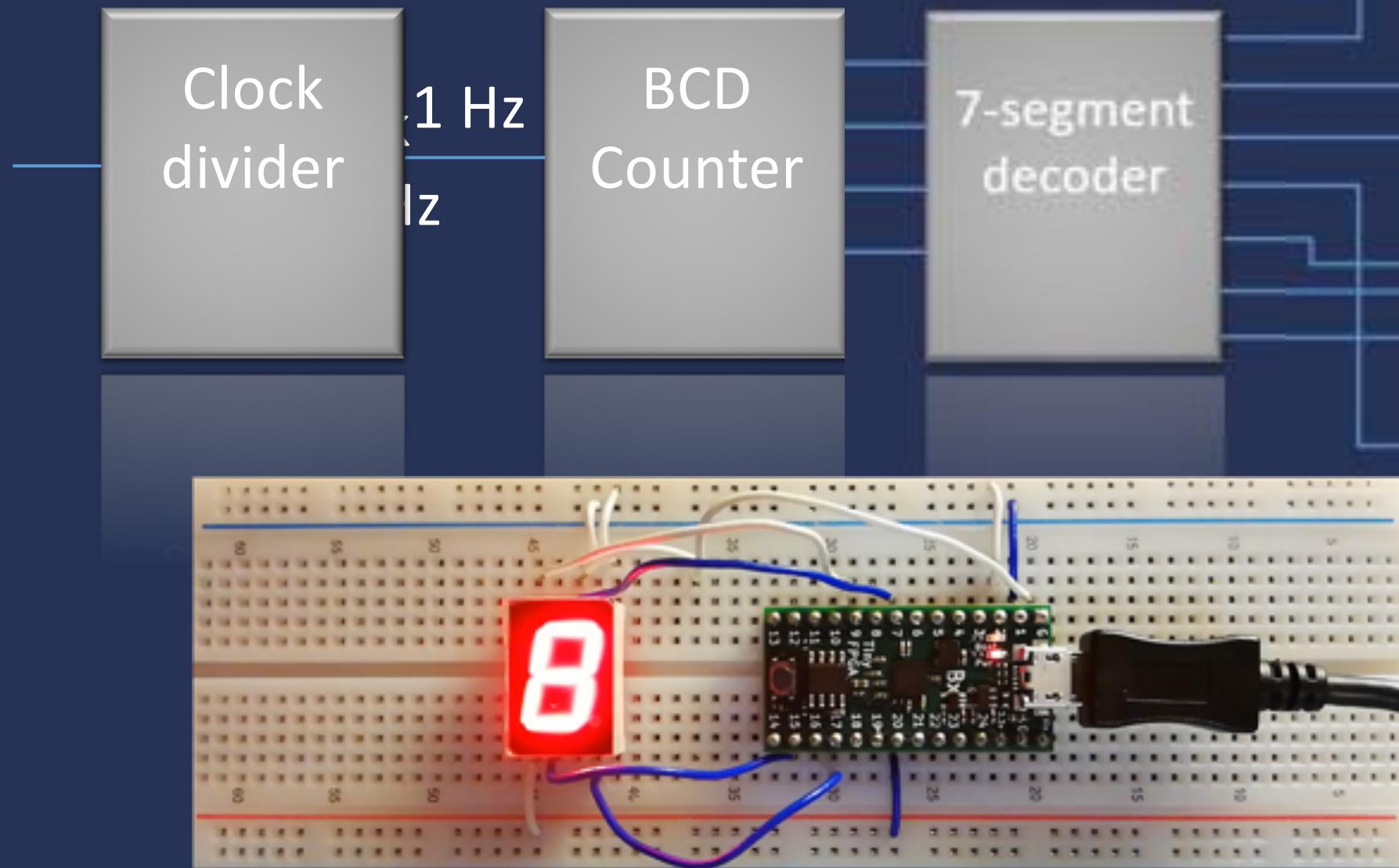
**Solution: Synchronizer**

- Adds delay to the input signal





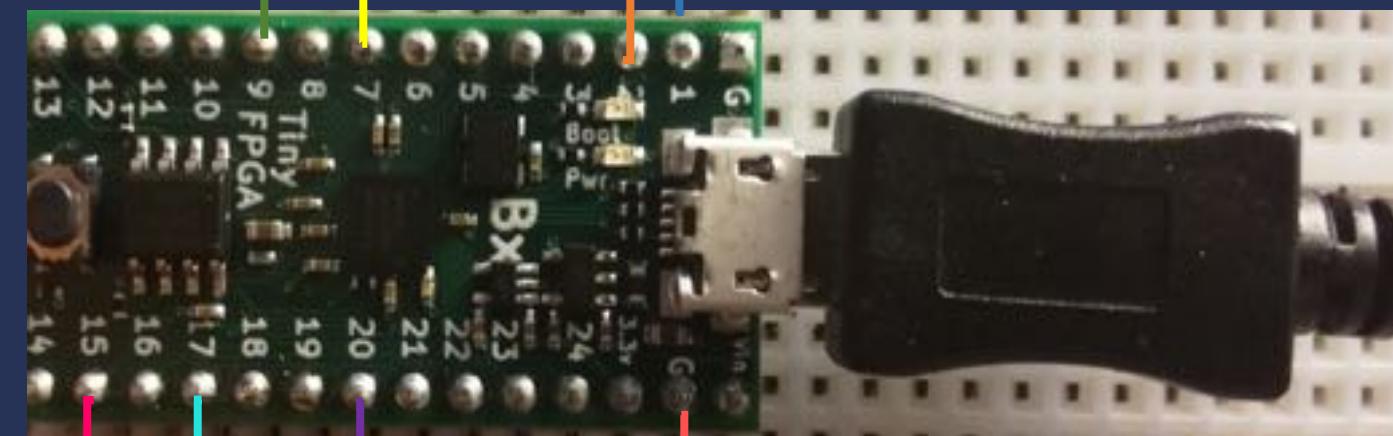
CLOUD V





CLOUD V

# BCD Circuit



9





CLOUD V

# 1. Bouncing

Push buttons are mechanical devices

- Bounce back and forth before settling
- Bounces settle in a duration of 20 ms
- Bouncing lead to glitches in the input signal



## Solution: Debouncer

- Filter out the glitches associated with the push buttons





CLOUD V

## 2. Metastability

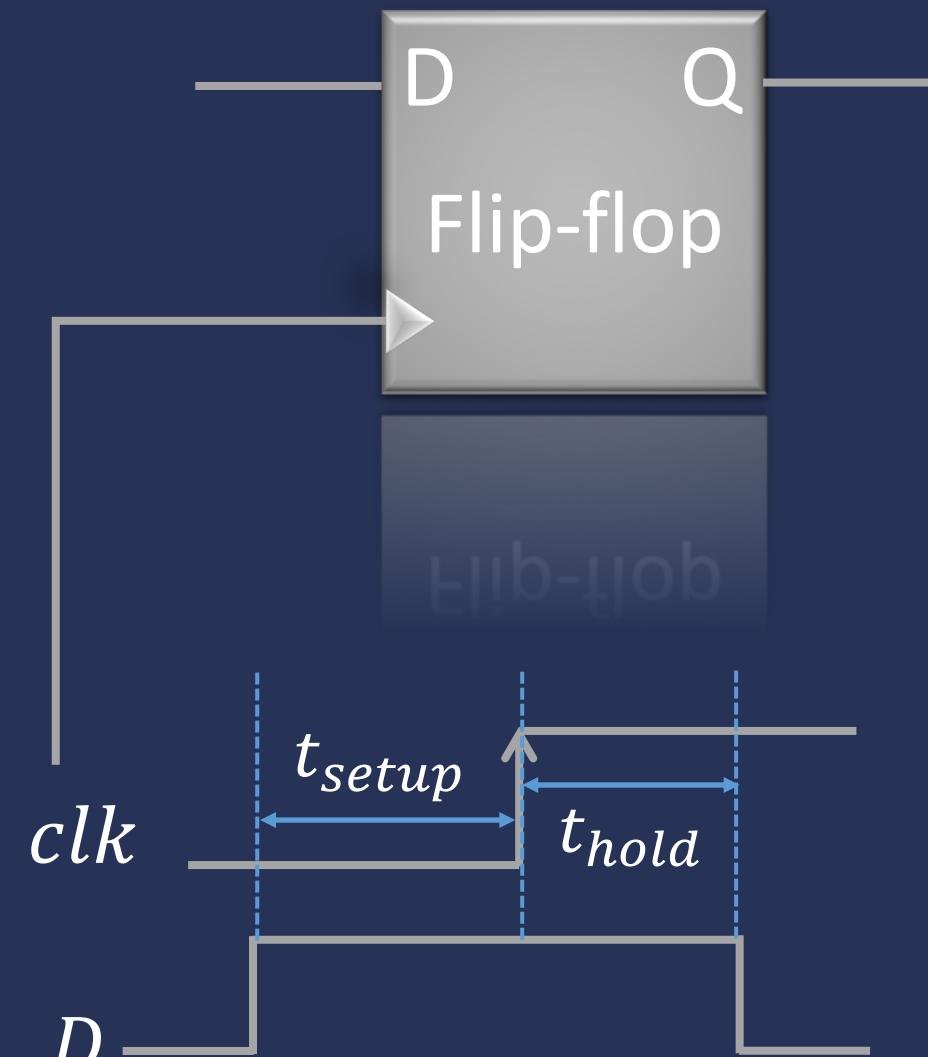
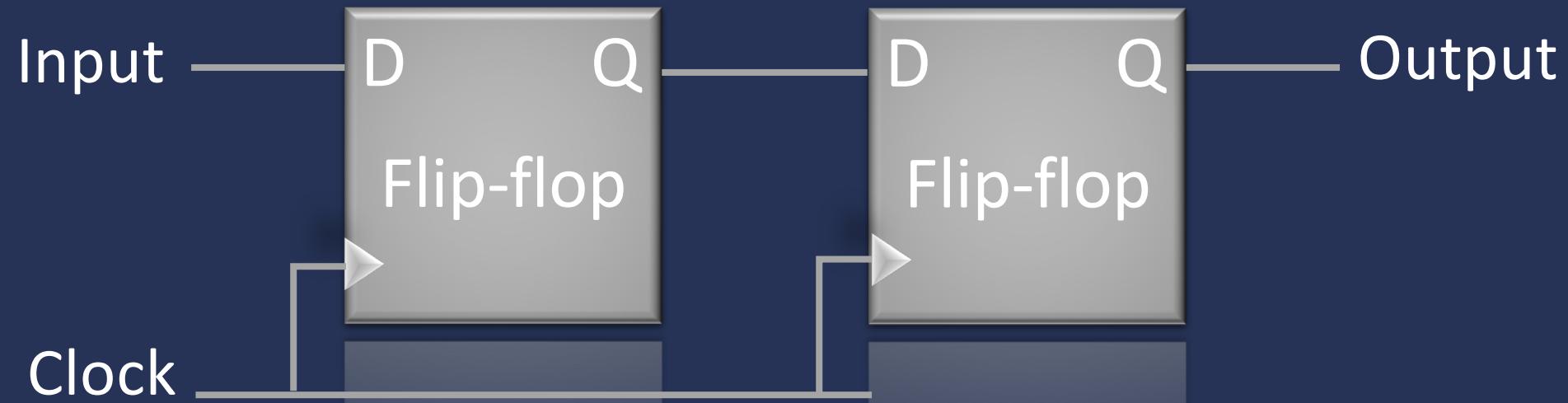
When timing constraints are not met :

- Q: random value 0 or 1 regardless the value of D
- Q: value between 0 and 1
- Q: oscillate between 0 and 1 very fast



**Solution: Synchronizer**

- Adds delay to the input signal





CLOUD V

### 3. Positive edge detection

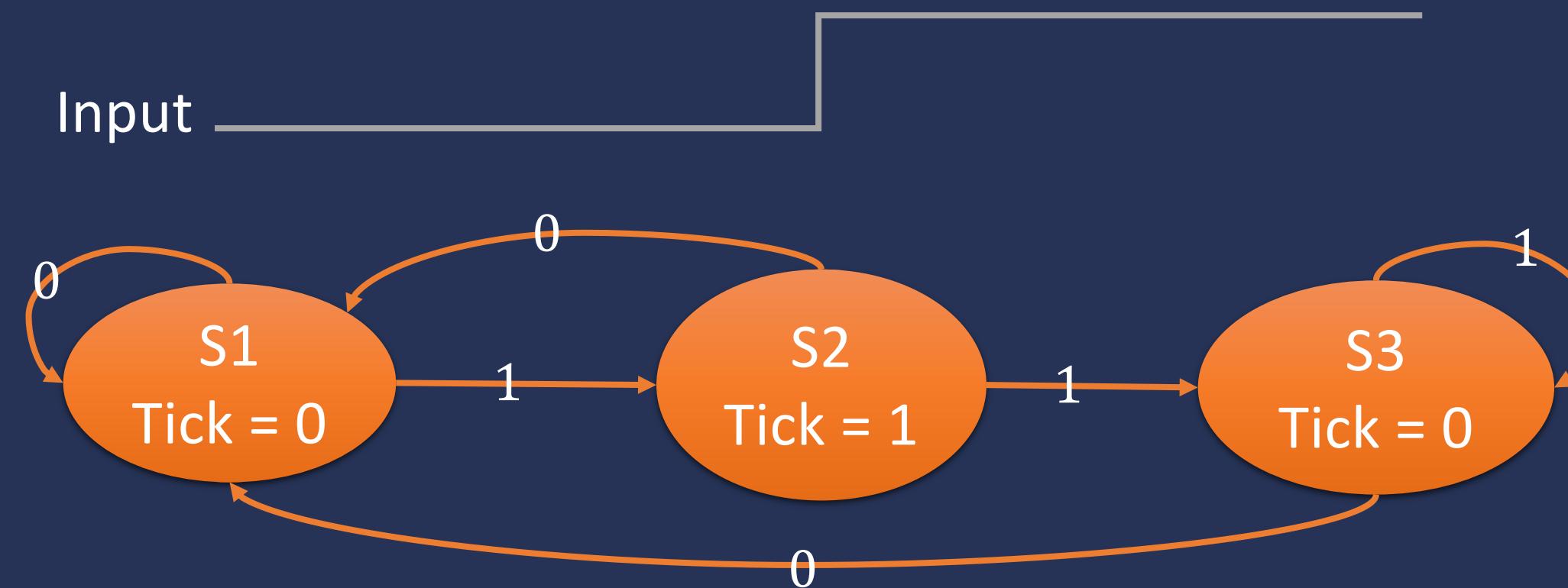
Push buttons are relatively slow with respect to the clock

- One press on the button may be detected as several presses



Solution: edge detector

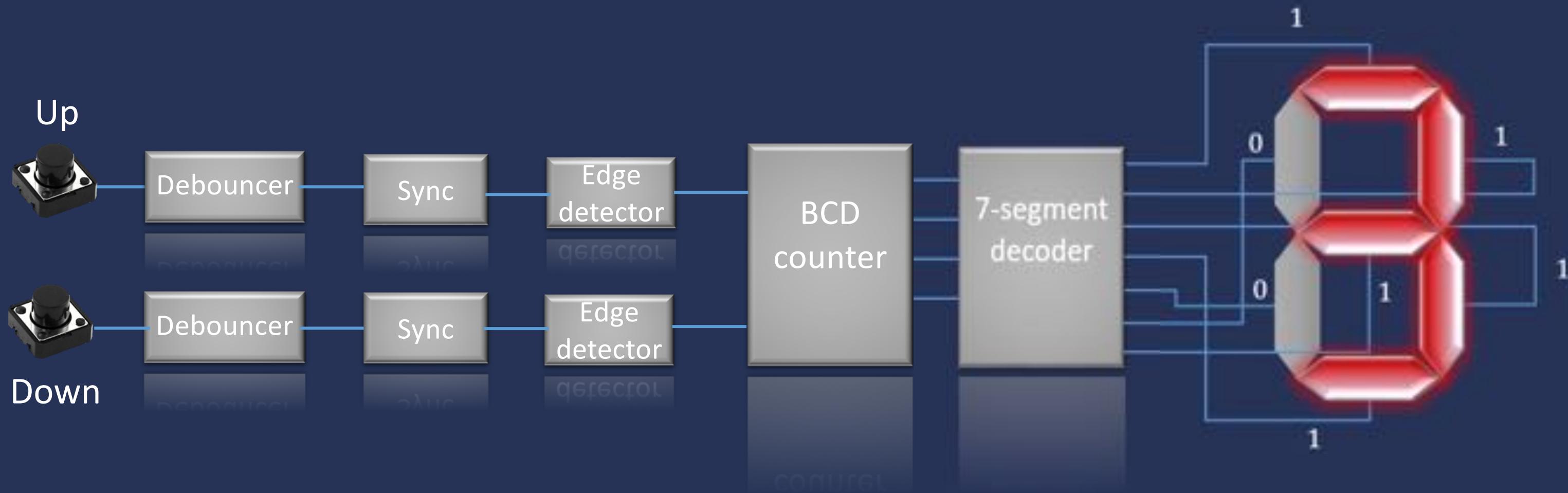
Indicates the beginning of a slow time varying input signal





CLOUD V

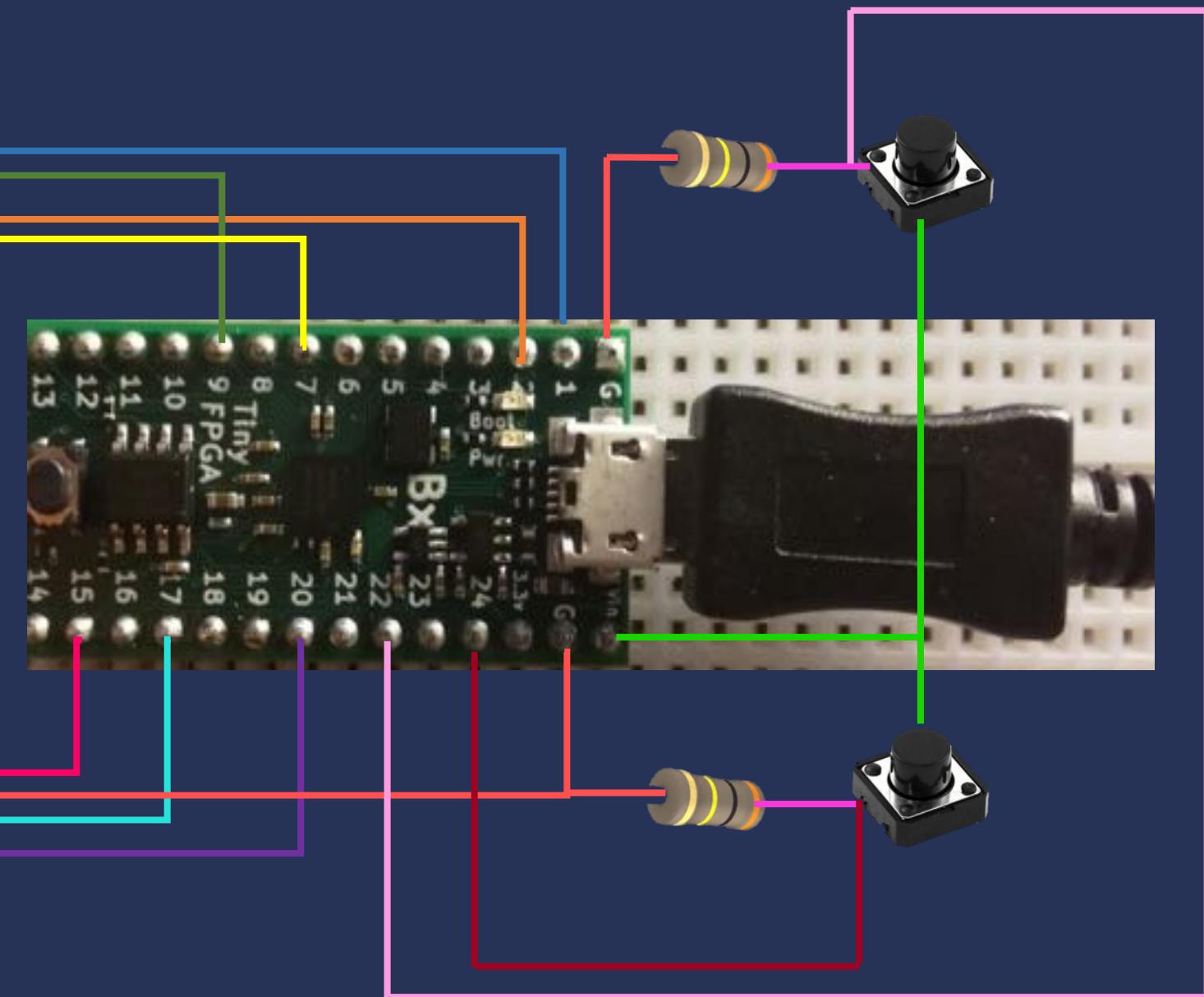
# Top module





CLOUD V

# BCD\_upDown Circuit



- <https://alchitry.com/blogs/tutorials/metastability-and-debouncing>