

# CS1026 – Digital Logic Design

## Introduction to Flip Flops and Latches

Shane Sheehan <sup>1</sup>

<sup>1</sup>ADAPT  
Trinity College Dublin

February 7, 2017

# Today's Overview

- 1 Introduction
- 2 Flip-flops vs Latches
- 3 Flip-flop/Latch Taxonomy
- 4 A Simple Set-Reset Latch

# Making Memory I

In electronics, a flip-flop or latch denotes a circuit that has:

- 1 Two stable states
- 2 Used to store state information.

Sound familiar?

# Making Memory II

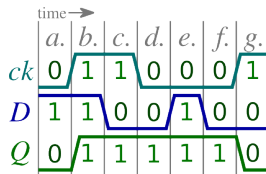
A flip-flop is a bistable multivibrator:

- Change state by signals applied to one or more control inputs

It is the basic storage element in sequential logic:

- Flip-flops/latches form fundamental building blocks for computers, embedded systems, etc.

# Making Memory III



A flip-flop/latch stores a single bit (binary digit) of data:

- one state represents a “one”
- the other represents a “zero”.

This allows the storage of state

# Sync vs. Async I

Flip-flops:

- Clocked (synchronous)

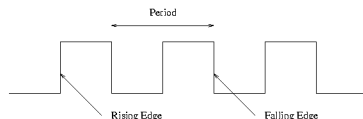
Latches:

- Simple (transparent or asynchronous)

## Hint

- Latch mainly used for storage elements
- Clocked devices denote Flip Flops

# Sync vs. Async II

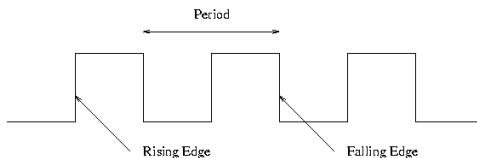


We use clocked devices for synchronous systems:

- Such devices ignore their inputs except at the transition of a dedicated clock signal
- Clocking causes the flip-flop to either change or retain its output signal
  - Based upon the values of the input signals at the transition

# Sync vs. Async III

Some flip-flops change output on the rising edge of the clock..



...others on the falling edge.

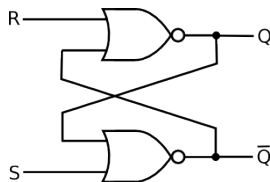


# Types of Flip-Flop/Latch I

Several Types:

- SR – “Set-reset”
- D – “Data” or “Delay”
- T – “Toggle”
- JK – Jack Kilby

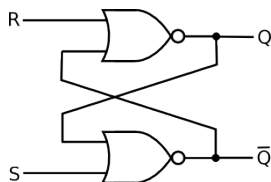
# Types of Flip-Flop/Latch II



The behavior of a particular type can be described by what is termed the characteristic equation:

- This derives the “next” output,
- $Q_{next}$  in terms of the input signal(s) and/or the current output,  $Q$ .

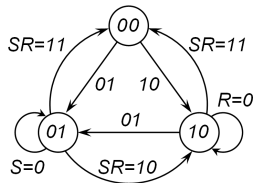
# SR NOR Latch [Kojima, 2013] I



Using logic gates, the most fundamental latch is the simple SR latch:

- Constructed from a pair of cross-coupled NOR logic gates
- The stored bit outputted at  $Q$

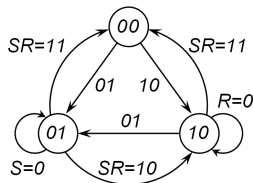
# SR NOR Latch [Kojima, 2013] II



While the R and S inputs are both low:

- Feedback maintains the  $Q$  and  $Q'$  outputs in a constant state
- Note:  $Q'$  the complement of  $Q$

# SR NOR Latch [Kojima, 2013] III



If S (Set) is pulsed high while we keep R (Reset) low:

- Then Q output becomes high
- Stays high when S returns to low

# SR NOR Latch [Kojima, 2013] IV

Characteristic Table				Excitation Table			
S	R	$Q_{next}$	Action	Q	$Q_{next}$	S	R
0	0	Q	Hold State	0	0	0	X
0	1	0	Reset	0	1	1	0
1	0	1	Set	1	0	0	1
1	1	X	Now allowed	1	1	X	0

Table: Characteristic and Excitation Table

Note: X means don't care

# SR NOR Latch [Kojima, 2013] V

The  $R = S = 1$  denotes a restricted combination or a forbidden state:

- Both NOR gates output zeros
- Then we have the logical equation  $Q = Q'$

Whoops!

# SR NOR Latch [Kojima, 2013] VI

Also inappropriate:

- Inputs go low simultaneously (i.e. a transition from restricted to keep)
- The output would lock at either 1 or 0 depending on the propagation time between the gates
- *We have a race condition*



# Ahead of schedule

## Any Problems?

- Ask!
- E-Mail: *sheehas1@scss.tcd.ie*

# References (Homework) I



Kojima, S. (2013).

Sr flip-flop.

US Patent 8,497,722.