Sequential Logic Design

CS 64: Computer Organization and Design Logic
Lecture #15

Ziad Matni
Dept. of Computer Science, UCSB

Administrative

- *Only 2 weeks left!!!!!!!! **OMG**OMGOMGOMGOMGOMG!!!!!!!!!!!!!!*

- 5/29 More Sequential Logic!
- Th. 5/31 Finite State Machines
  - *Lab# 8 Sequential Logic and CPU Design*

- Tu. 6/4 More Finite State Machines!
- Th. 6/8 CS Ethics and Societal Impact of CS
  - *Lab# 9 Finite State Machines*
  - *Lab# 10 Ethics/Societal Impact (short; online)*

Lecture Outline

- Exercise with Latches and Flip-Flops!
  - Lots of note taking in this lecture
  - Many details will be on the blackboard

Any Questions From Last Lecture?

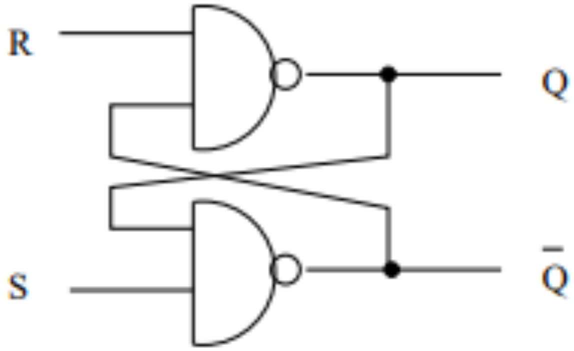
Pop Goes The Quiz!

Combine *multiple* 1-bit, 2:1 muxes to create a **1-bit 8:1 mux**.

You can choose to use any other type of combinatorial logic blocks in your design, if you need it.

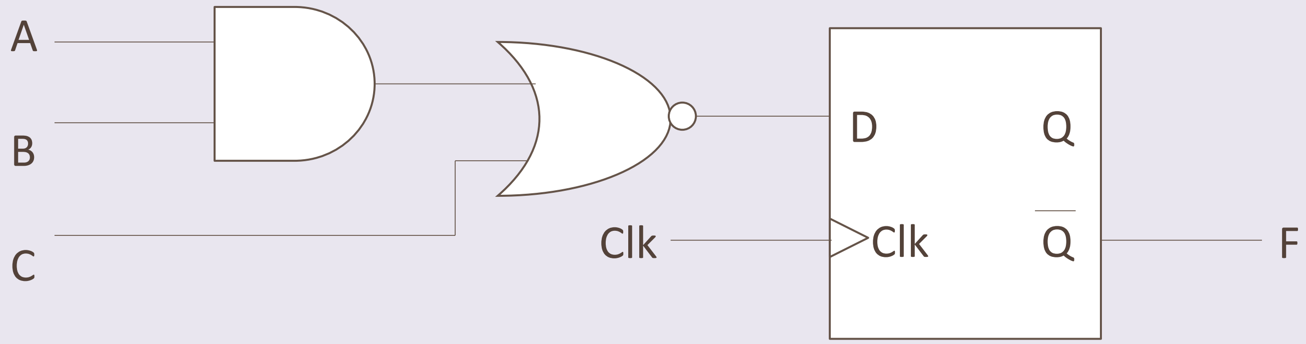
Exercise 1

The figure below shows an RS latch made out of NAND gates (rather than NOR gates). How do Q and $\overline{Q}$ depend on the RS inputs? i.e. verify that the circuit can indeed be used as a RS latch.



Exercise 2

Given waveforms for A, B, C, and Clk (see blackboard), determine the output waveform for F

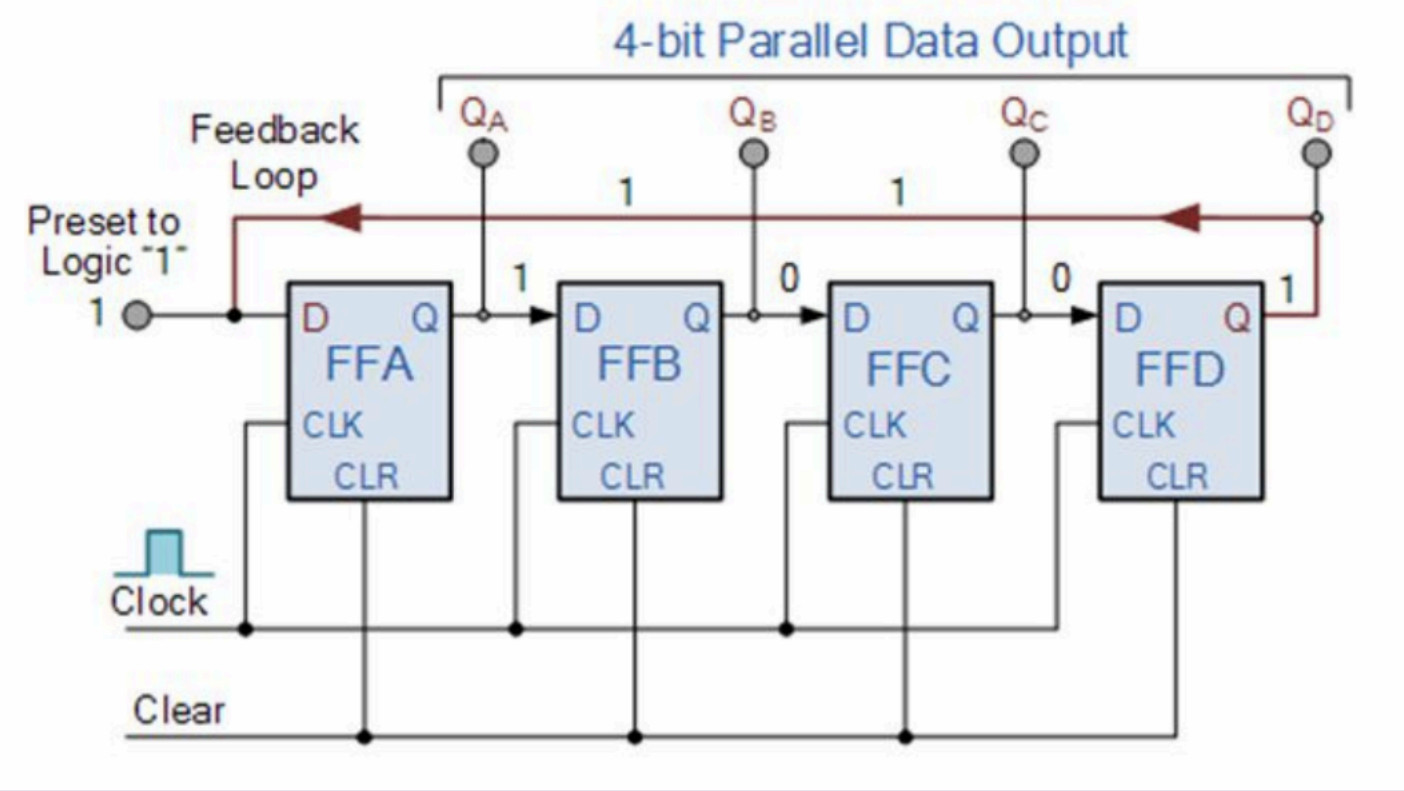


Exercise 3

This is known as a "4-Bit Ring Counter". Let's analyze what it can do…

Assume that, initially, input into register A is 1 and all others are 0…

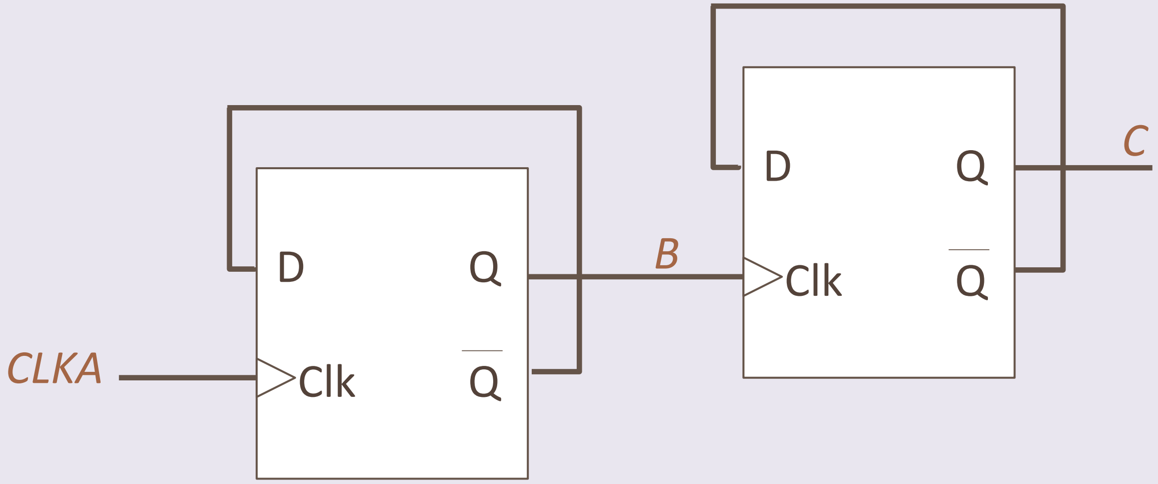
Done by applying a reset to everything and then a pre-set for just reg.A



Exercise 4

- Draw the waveforms **B** and **C** for this digital circuit, given CLKA is a regular clock input.
  - Assume all inputs to the D-FFs are initially at 0.

- What do you conclude about what this does?



Exercise 5

- Let’s design a 3-bit counter using D-FFs and logic gates.

- What’s needed:
  - This counts 000 → 001 → 010 → … → 111 → 000
    - i.e. from 0 to 7 and then loops again to 0, etc…

- To describe this behavior, let’s start with a T.T.
  - We’ll utilize K-Maps, if needed to figure out what the “next states” look like based on “current states”
  - We’ll translate that into a digital circuit design

Lab 8

Note When Everything is Due…

- Lab#8 in progress…
- See TAs during their office hours (We. & Fr.) for help
- Due on **MONDAY 6/4** --- Not the usual Friday!

- **Lab#9** will be ISSUED ON MONDAY 6/4
  & BE DUE ON FRIDAY 6/8
- **Lab#10** will be ISSUED ON THURSDAY 6/7
  & BE DUE ON FRIDAY 6/8

What’s Lab8 About?

- Design a “simple” ALU (Task 1)
- Design a “simple” register block using D-FFs (Task 2)
- Be given a specification for a “simple” CPU that uses:
  - 1 “Simple” Register Block
  - 1 “Simple” ALU
  - 1 Abstract Computer Memory Interface
  - As many ANDs, ORs, NOTs, XORs, Muxes that you need
- Design this CPU! (Task 3)

- You will draw all of these (BE NEAT!)
  - Take pictures or (better yet) scan them, then turnin

Your To Dos

- Lab #8 - due Monday 6/4

- Lab #9 will be issued next weekend
  - Due Friday 6/8 (last day of classes)
- Lab #10 will be issued in the last week
  - It’s short and due Friday 6/8 (last day of classes)

</LECTURE>