CS 2110 Lab 5
Mux's, Decoders, Adders, Registers, Flip-Flops, Memory, Oh My!

Written by your TAs

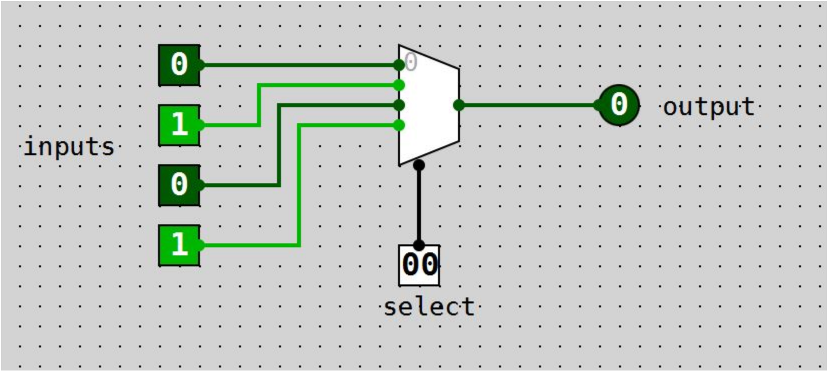
Fall 2018

1 Outline

1. Multiplexers
   Selects one of the input lines and drives it to the output.
2. Decoders
   Based on the value of the input, the corresponding output will hold a value of 1.
3. Adders
   This circuit can add two one-bit numbers together. When daisy-chained together, this circuit can add n-bit numbers together.
4. Level-Triggered Logic
   (a) SR Latches
   The simplest form of sequential logic.
   (b) Gated D Latches
   Another form of level-triggered sequential logic that has more control over state than the SR latch.
5. Edge-Triggered Logic
   (a) D Flip-Flops
   A circuit implementing edge-triggered sequential logic.
6. Registers
   Built out of latches or flip-flops, this circuitry holds values.

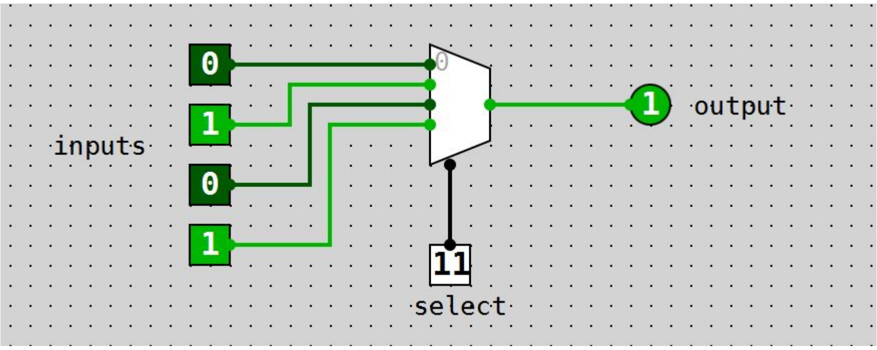
2 Multiplexers

A multiplexers selects one of the input lines and drives it to the output.



The above image shows a mux with four inputs, one output, and a 2-bit select. The value of the select bit will choose which input line will be sent to the output. Since the select bit is currently 0b00, this means that the value at the 0'th input line (0) will be sent to the output.

If we change the select bit to be 0b11, then the value at the 3rd input line (1) will be sent to the output as seen in the below image.

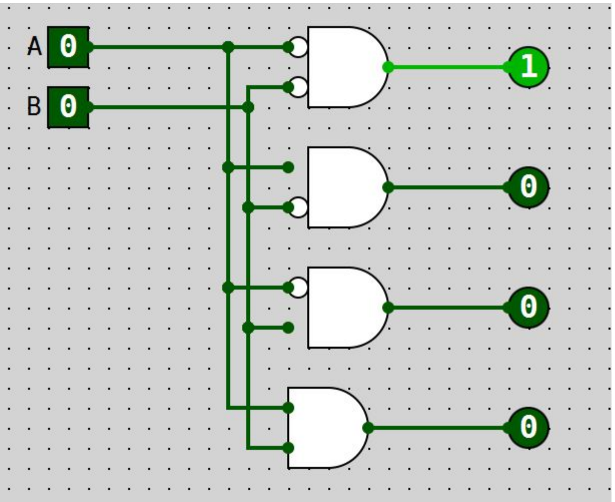


2.1 Things to know

- In general, with n select bits you can have 2^n inputs and 1 output
- Multiplexors are used throughout digital logic, whenever the circuit wants to select a line. One example of mux use is the ALU which selects which completed operation to send to the output. Another place muxes pop up is with the LC3 datapath, where it helps decide which line to choose.

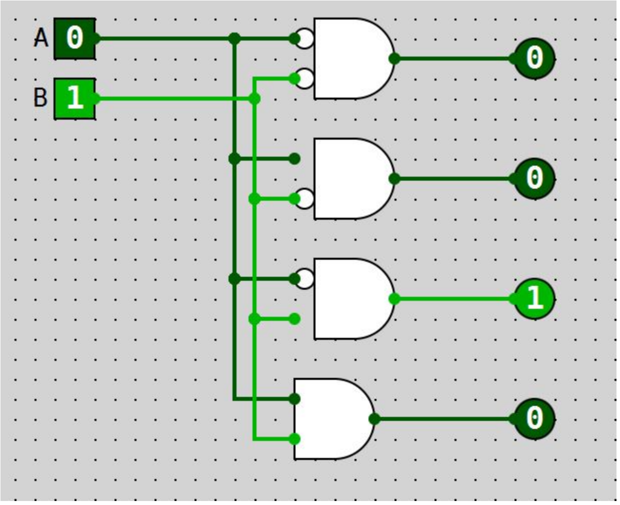
3 Decoder

A decoder has a n-bit input and 2^n outputs. The input determines which corresponding output will have a value of 1.



In the above example, the input is 0b00, leading to the 0th output to have a value of 1.

If the input changes to 0b10, then the 2nd output will have a value of 1 as seen in the below image.

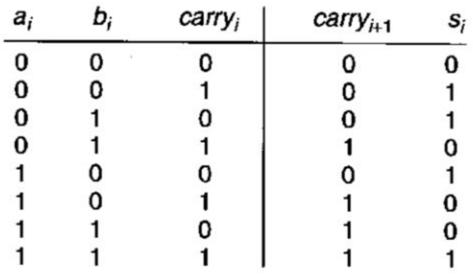


3.1 Things to know

- With an n-bit input, you will have 2^n outputs.
- Decoders are used primarily for the LC3 datapath. The opcode of an instruction is decoded to set certain flags on the datapath. We will discuss this more in depth once we get to the datapath.

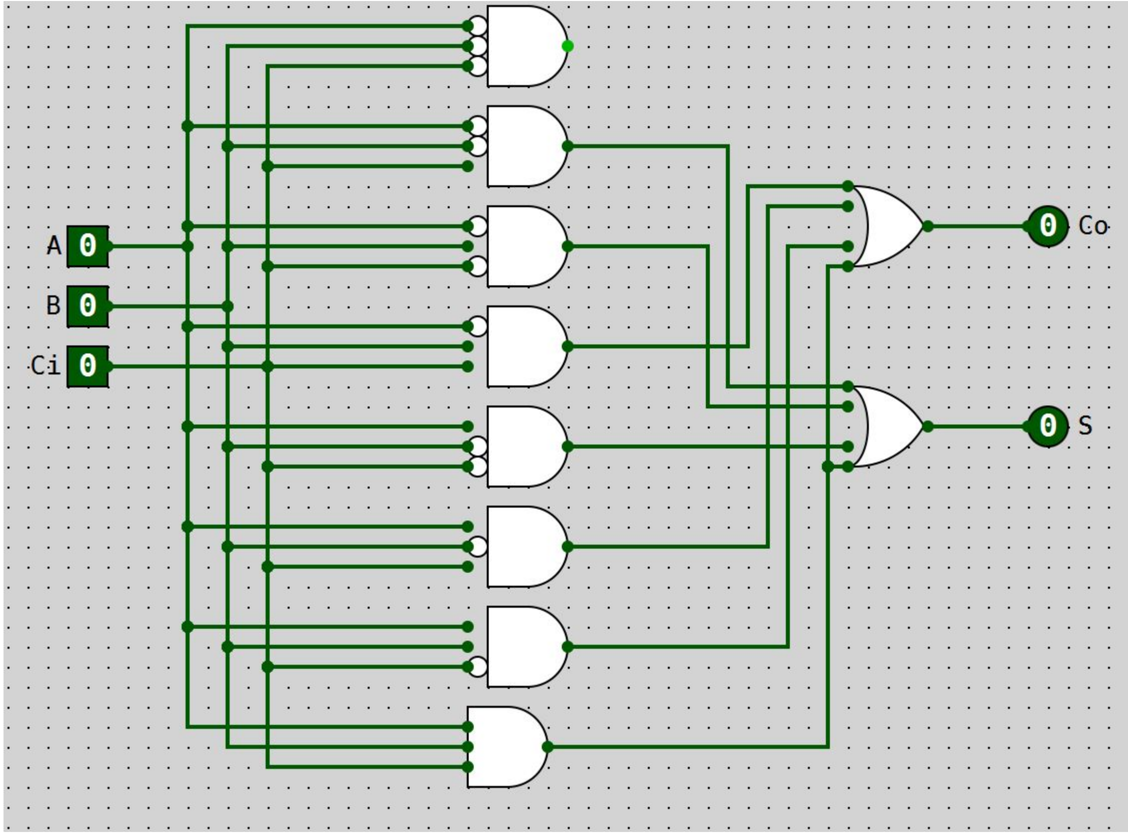
4 Adder

An adder takes two one-bit inputs and adds them to a carry-in. Building this is easiest with a truth table

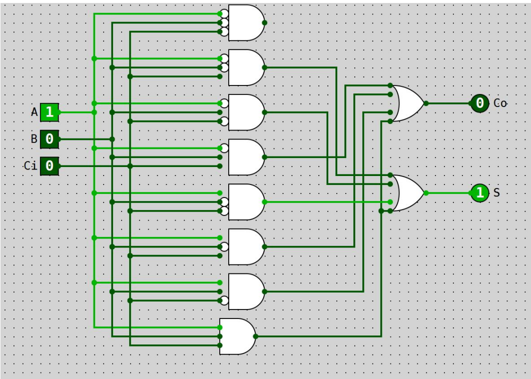


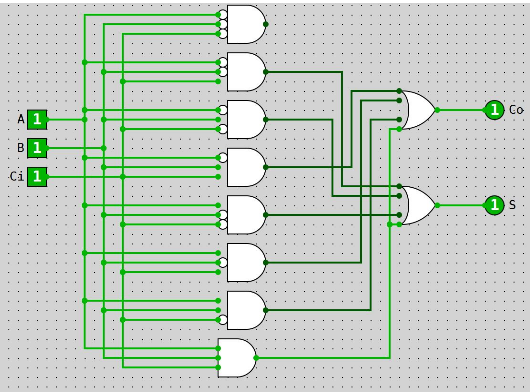
a_i	b_i	$carry_i$	$carry_{i+1}$	s_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

that will lead to this circuitry for a **one-bit adder**

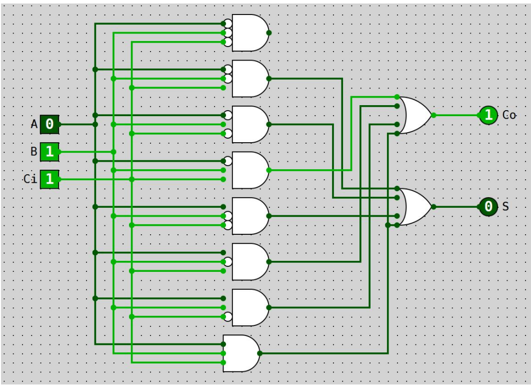


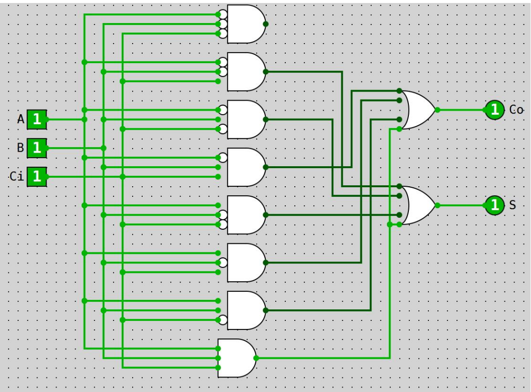
For the adder, note that the Sum output has a value of 1 whenever an odd number of inputs have the value of 1



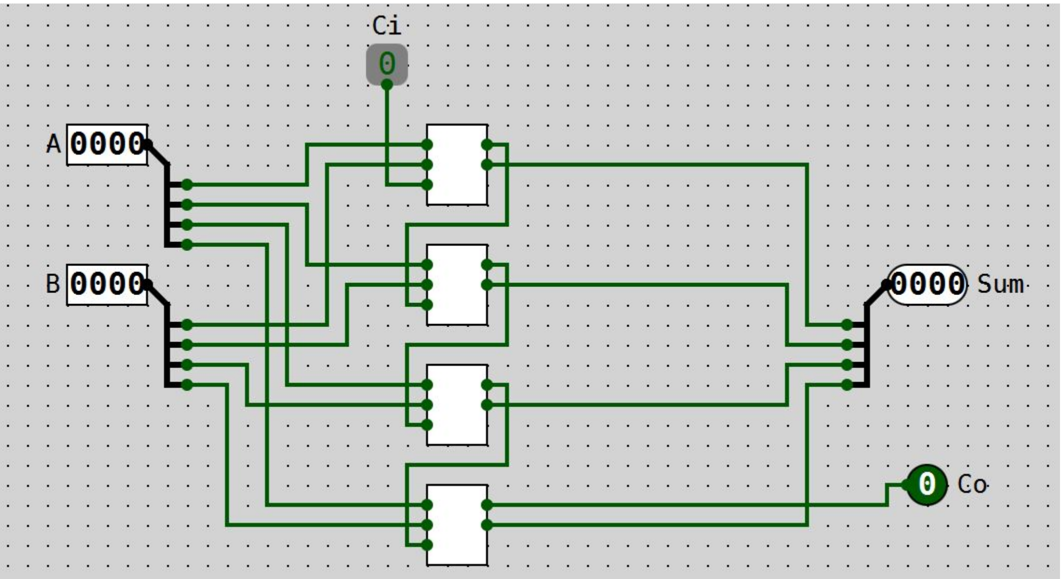


Also, the carry-out output has a value of 1 whenever two of the inputs have the value of 1 or all of the inputs have the value of 1.





Note: This circuitry is only for a one-bit adder. To add more than one-bit elements, you connect one-bit adders together such that the carry-out of the n-1 bit will be the carry-in of the nth bit (this is called daisy-chaining)

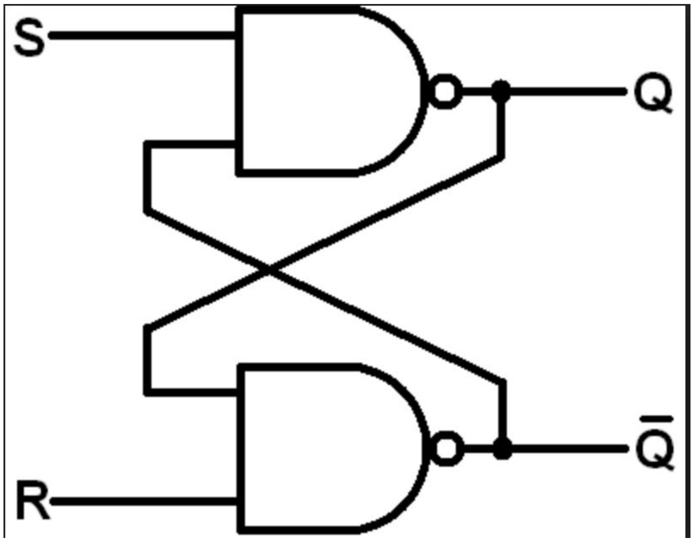


5 Latches and Flip Flops

Sequential Logic is the idea that the output of the circuit depends on the current inputs and previous values of the inputs.

5.1 RS Latch

The basic circuit for this is the RS Latch.

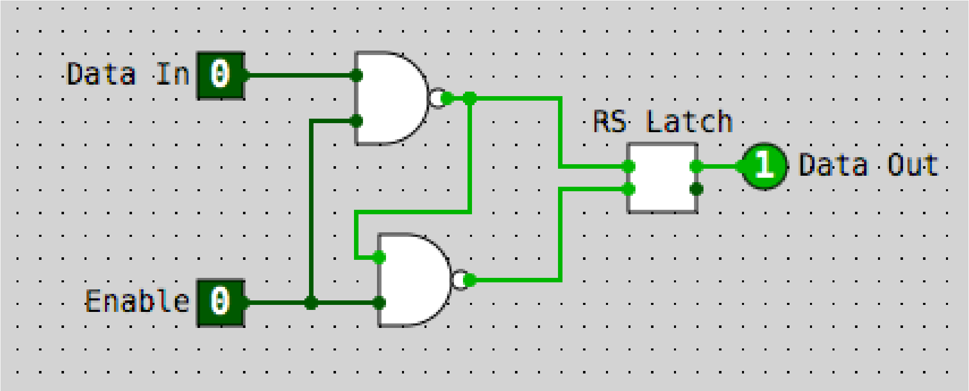


The RS Latch has three interesting states:
R=1 S=1
R=1 S=0
R=0 S=1
The key thing to notice is that the state R=1 S=1 does not always correspond to the same value for the output. If it was previously in the state R=0 S=1 then the output will be 0, but if it was previously in the state R=1 S=0, then the output will be 1. This is a perfect example of sequential logic! You can also notice the second output always corresponds to the opposite of the first when in any of these 3 states, so we usually ignore it.

Note: What we described is technically an not S not R latch, we could also create an RS Latch using 2 nor gates instead of a nand gate. This changes the illegal state to R=1 S=1 instead of R=0 S=0 as we just described. The importance with either, however, is that they latch onto a value and are examples of basic sequential logic. We'll refer to the above example with two nand gates as an RS latch for the rest of this and HW4.

5.2 Gated D Latch

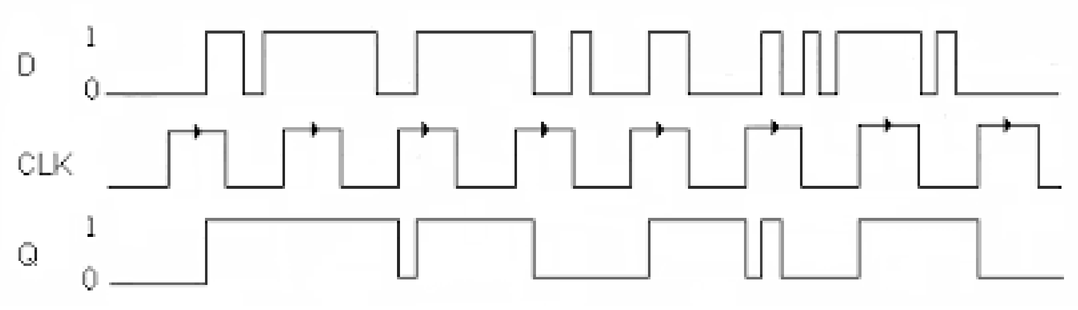
The Gated D Latch is essentially just an RS Latch with extra control so we can choose when to save the output and what to save it as, using our data and enable inputs.



You can notice that when the enable input is 0, changing our input data makes no change to the output data. We have successfully saved a single bit!

5.2.1 Edge-Triggered Logic

Both RS Latches and Gated D-Latches are examples of level-triggered logic. This means the output will reflect the input anytime the enable is on. We can visualize this by the below diagram. If the enable(CLK) is on (1), then whatever the input(D) is, the output(Q) will become.



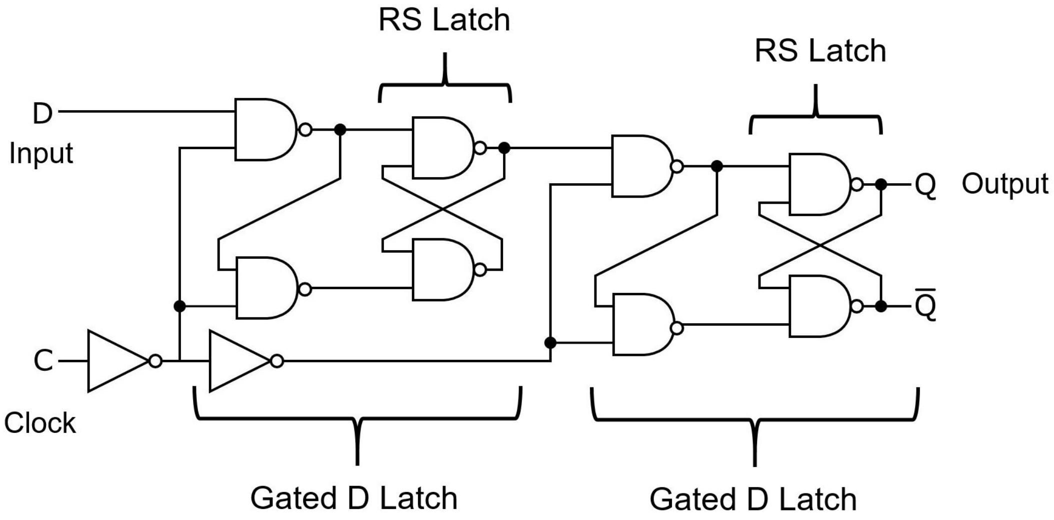
Currently, whenever the enable is 1, we can change the output as many times as we want. However, modern memory systems utilize **edge-triggered** logic. This means that the output only changes on an edge, usually the rising edge, when enable changes from 0 to 1. The amount of time for an edge is almost instantaneous, whereas the amount of time for the level is not. By waiting for the edge trigger, we can synchronize all our memory systems to update at the same time. This lets all of the memory be in a single state at any given time.

5.2.2 The Clock

One aspect of updating all latches simultaneously is the clock. In the above diagram, the enable was represented by CLK, which is the clock. The clock essentially acts as our enable pin, but oscillates between 1 and 0 at a set frequency. With the clock connected to every single latch, we can set the enable bit to 1 or 0 on every single latch simultaneously. We still can have an enable pin, however, that will override the clock to still disable changing the state of our memory while the clock is going.

5.3 The D Flip-Flop

The circuit that implements edge-triggered logic is called a **D Flip-Flop**, and is essentially two Gated D Latches back-back.

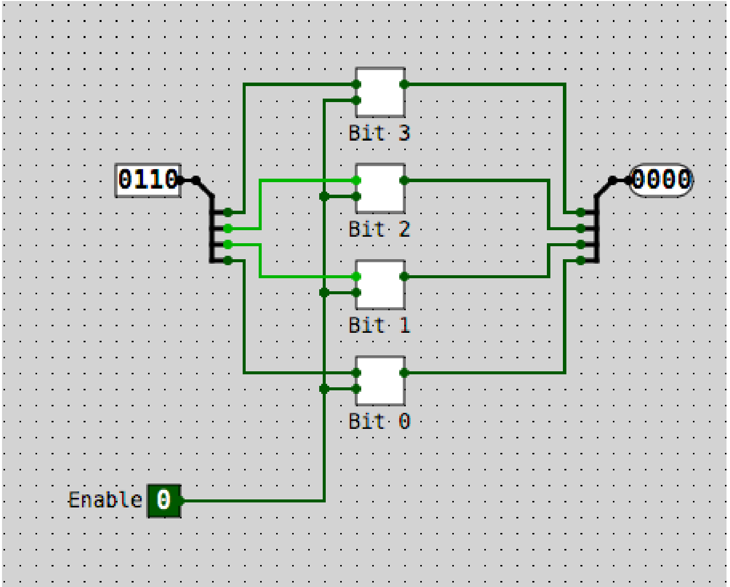


5.4 Main Takeaways

- Latches are level-triggered
- Flip-flops are edge-triggered.
- Edge-triggered devices combined with the clock as the enable bit allows for all sequential logic devices to be updated simultaneously.

6 Register

Registers can be built out of latches or flip-flops. These circuits hold values and are updated on the positive level (if implemented with latches) or on the rising edge of the clock (if implemented with flip-flops).
The below circuit chains multiple Gated D Latches together to save a number comprised of multiple bits.



The above image is a 4-bit register. So it can store the value of a 4-bit number. Just as we were able to chain Gated D Latches, we could have chained D Flip-Flops to make our register use edge-triggered logic.