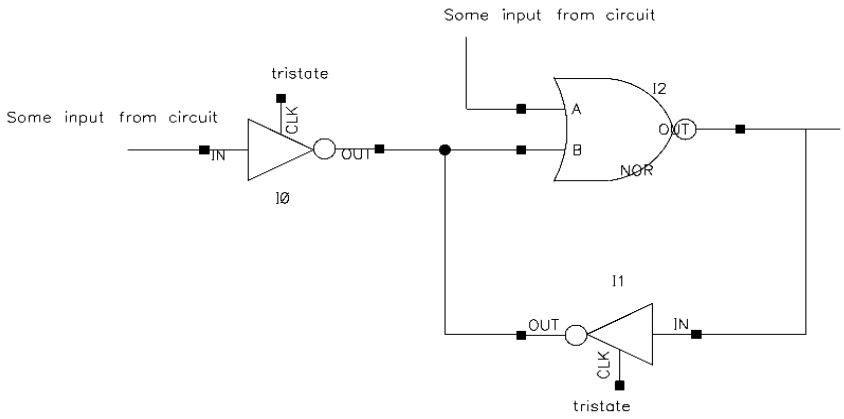
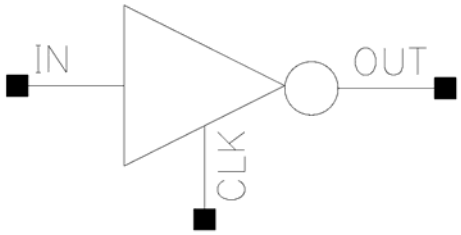
Lab 6 Supplement:
Guide to Designing CMOS Flip Flops, Multiplexers, and Shift Registers
A 410 Lab Help Document

Guide to Designing CMOS Flip Flops

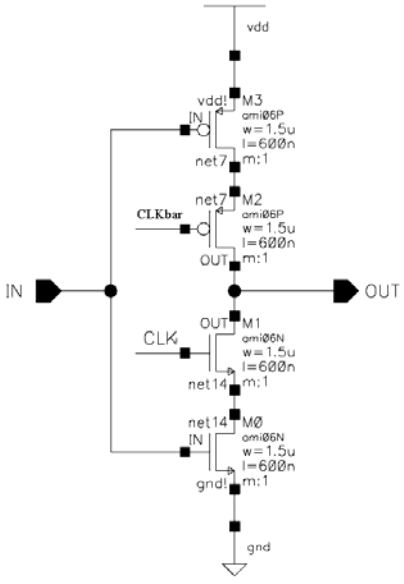
The provided flip flop layout may be hard to interpret, but it does follow the basic structure for a master-slave d-type flip flop with reset, DFFR. Portions of the given flip flop make use of a latch structure similar to the following figure:



In this latch schematic, the tristate circuit is just a clock controlled inverter that will evaluate the input when the clock is high and hold the output when the clock is low, using the tristate schematic shown below. If the CLK and CLKbar inputs are switched the circuit will evaluate on clock low and hold on clock high.



Tristate Symbol



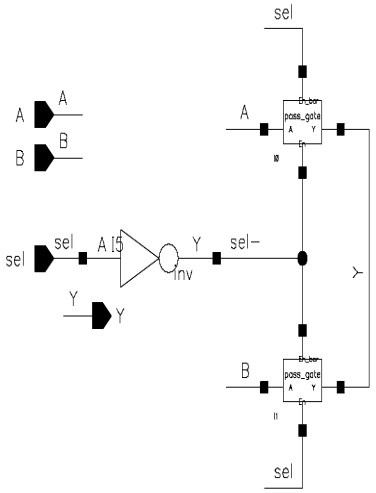
Tristate Schematic

This information should help you begin to trace the schematic from the DFFR layout. However, the DFFR cell is more complex than any cells from pervious labs, so take your time analyzing the DFFR layout.

Guide to Designing Multiplexers

A multiplexer (mux) is a circuit used to select a specific output from a group of input signals. The information below shows how to design transmission gate multiplexers and use smaller multiplexers to construct large ones.

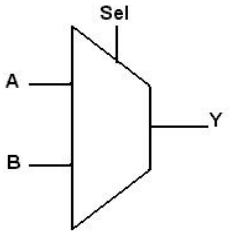
2:1 MUX HINT



Schematic of a 2:1 multiplexer.

MUX output can be expressed as:
If (sel=0), Y=A;
Else Y=B;

(Note: When you *Check and Save* the schematic, you will get a warning that the outputs of the two pass-gates are shorted. You can ignore the warning message.)

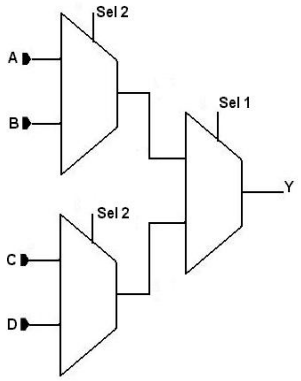


2:1 Mux Symbol

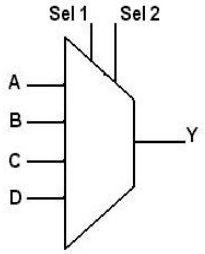
Sel		Y
0		A
1		B

Truth Table

4:1 MUX



Schematic of a 4:1 multiplexer.



4:1 Mux Symbol

Sel 1	Sel 2		Y
0	0		A
0	1		B
1	0		C
1	1		D

4:1 Mux Truth Table

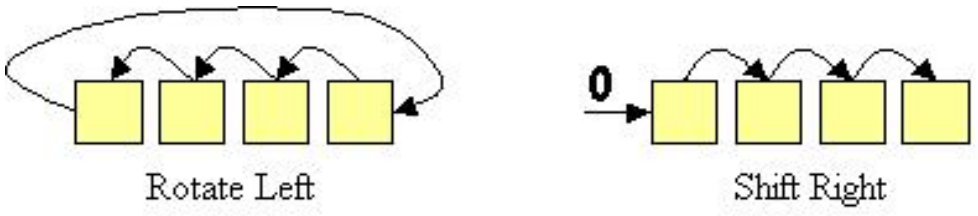
Guide to Designing Shift Registers

Shift Register Notes:
A shift register that is capable of multiple synchronous functions (shift left, shift right, load external data, and load preset value) and asynchronous reset can be constructed from only DFFR and MUX cells.

Rotate and shifts are very common operations for the register. Rotation is the act of moving each bit of data to an adjacent bit while rolling the end bit to the opposite end. For example, if a rotate left is performed, the most significant bit will move to the least significant bit position. Shifting is the act of moving each bit of data to an adjacent bit while loading a zero into the open-end bit. For example, if a shift left is performed, the least significant bit will become a zero while all other bits are shifted one place to the left.

E**xamples: 4b operations on data $a_3a_2a_1a_0$**

Rotate Left: output = $a_2a_1a_0a_3$
Rotate Right: output = $a_0a_3a_2a_1$



Simulations to verify all operations of a multi-function shift register may take a long time and will consume a lot of disk space. Be sure to use your class directory to do the simulations.