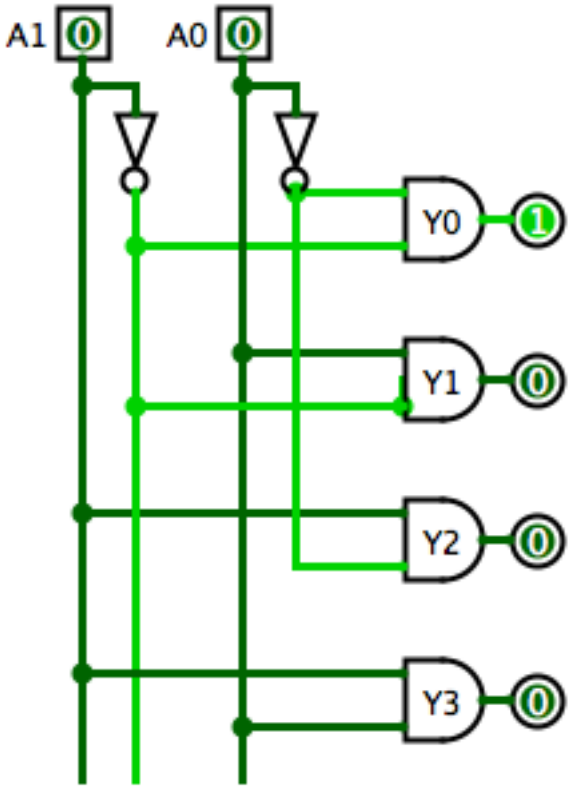
LD-Lab, Logic Design (IS1500)

Martin Hwasser, hwasser@kth.se

September 22, 2017

Task 1.1

Question 1. Construct a 2:4 decoder from the following components: 2 inputs, 4 outputs, 2 NOT-gates, and 4 AND-gates. Mark the two inputs as A0 and A1 and the four outputs as Y0, Y1, Y2, and Y3.



We verify that it works correctly by making sure it is one-hot encoding the inputs, eg for all possible input values, there's one corresponding Y value which is true, and the other outputs are false.

Task 2.1

Question 1. Explain why output "MUX out" is blue, why output signal "Equal?" is red, and the select signal of the 4:1 multiplexer is black.

MUX out is **blue** because it's a floating bit. There's nothing driving a value onto the wire.

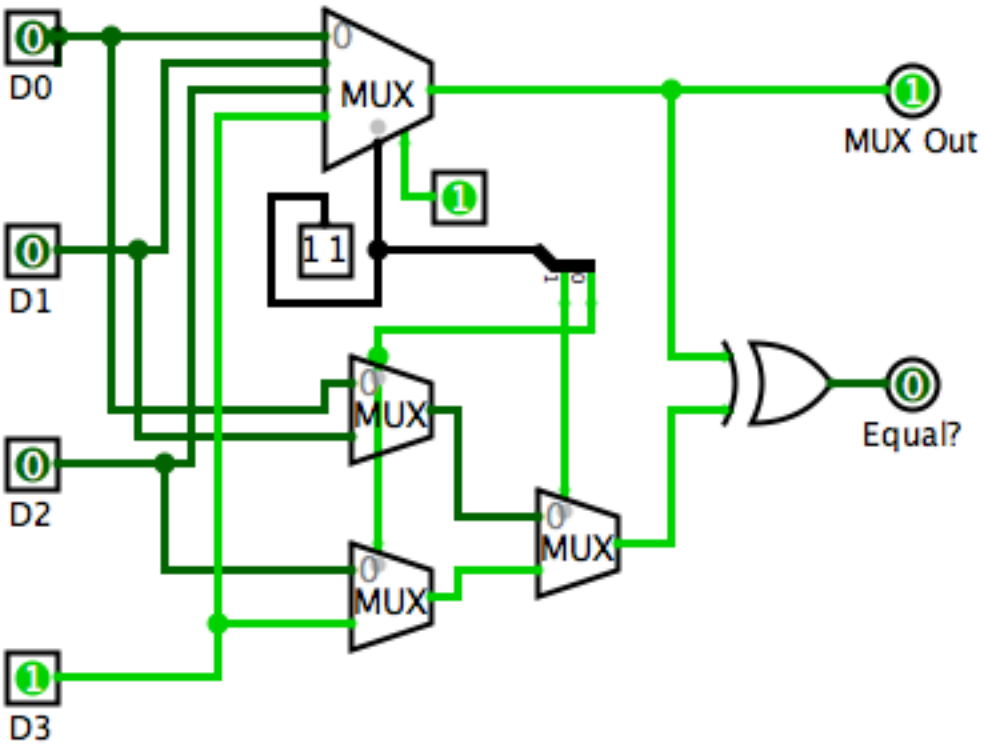
The output signal is **red** because it carries an error value. In this case it carries an error because it has no inputs and thus cannot determine the output.

The select signal is **black** because it's carrying a multi-bit value, in this case two bits.

Task 2.2

A 4:1 multiplexer can be constructed from 3 different 2:1 multiplexers (see H&H or the lecture slides). Your task is now to connect the three 2:1 multiplexers in such a way that their output gives the same behavior as a 4:1 multiplexer. You should also connect both the output from this new multiplexer (the combination of the three 2:1 multiplexers) and the old 4:1 multiplexer to the XOR gate.

Question 1. Include a screenshot of your complete design. Note that it should look similar to the figure in task 2.1, but with additional wires. You do not need to add any extra components.

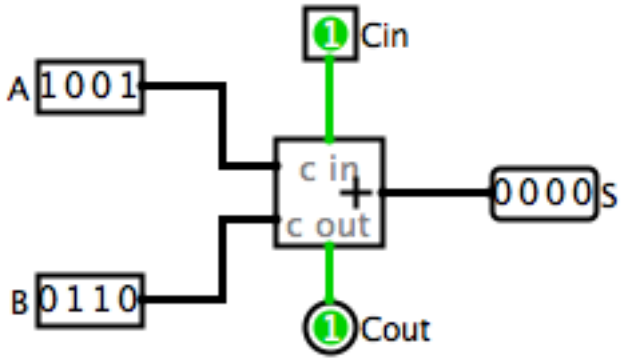


Question 2. Explain how you can test the correctness of the component using the output signal "Equal?".

Since we want both signals to be off at the same time, and on at the same time, the output signal "Equal?" should never be on since it's an XOR gate.

Task 3.1

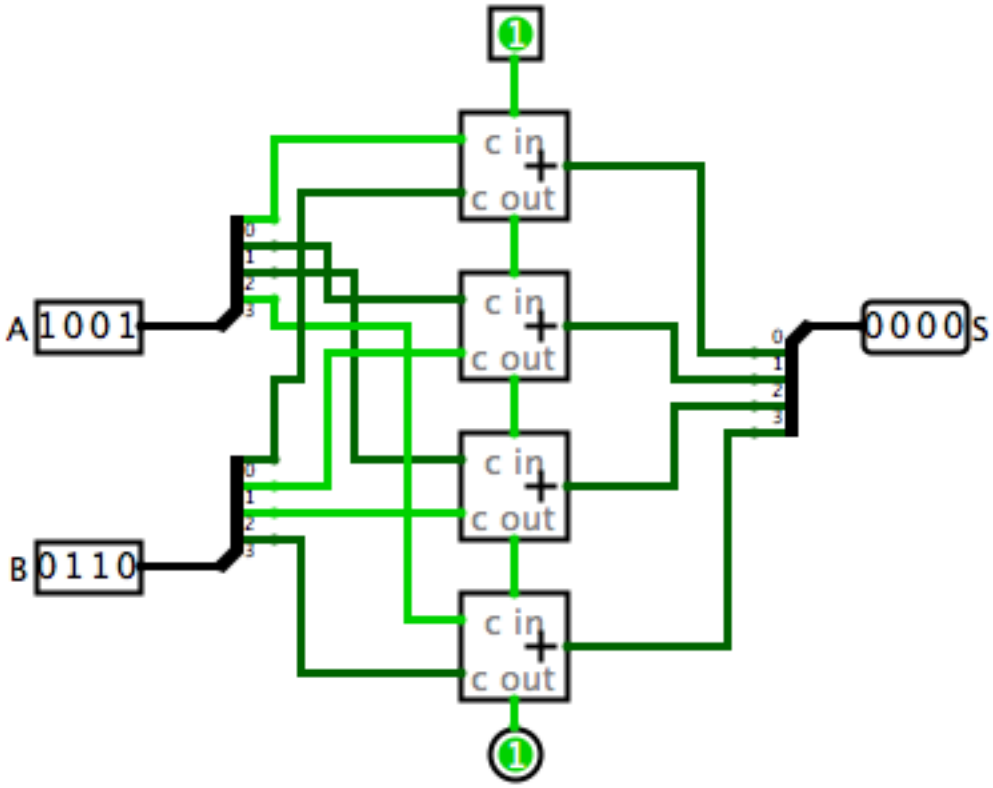
Question 1. Include a screenshot of your design. Note that you need to find out what input signal A should have, to fulfill the requirements in task 3.1. The figure should contain the correct input and output signal values.



Task 3.2

Construct a 4-bit ripple-carry adder by using four 1-bit full adders, two 4-bit input signals (A and B), one carry in signal, one carry out signal, and a 4-bit output signal (S). The design needs to use three 4-bit splitters.

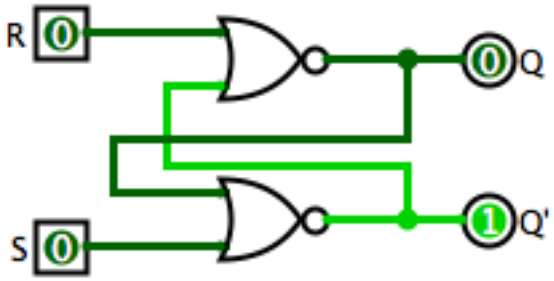
Question 1. Include a screenshot of your design. Your design should use the input and output values that were requested in task 3.1.



1 Task 4.1

Construct an SR latch by using two NOR-gates. Clearly mark S and R inputs, and the Q and Q' outputs.

Question 1. Include a screenshot where both S and R are 0. Explain what the possible values for Q and Q' can be.

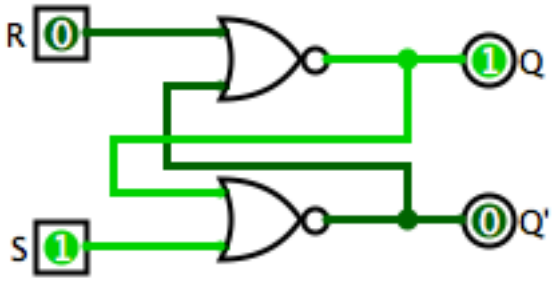


The possible values are:

- Error (if Q has not had a previous value)
- Q=1 and Q'=0, if Q was previously 1

- Q=0 and Q'=1, if Q was previously 0

Question 2. Include a screenshot where S = 1 and R = 0. Explain the possible values for Q and Q' in this case. If you toggle S (switch between 0 and 1), do then Q or Q' change? Why or why not?



When S=1 and R=0 then Q=1 and Q'=0. Q and Q' do not change when toggling S, it "latches" on the previous value.

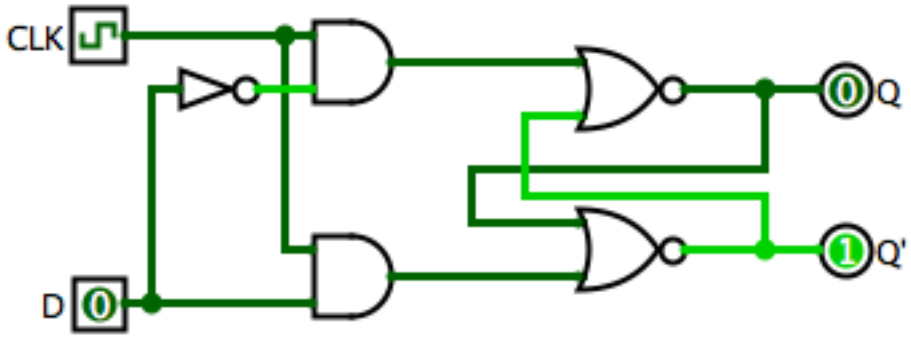
Question 3. What happens if S is equal to 0 and R is toggled between 1 and 0?

If S=0 and R=1, then Q=0 and Q'=1 and will remain stable even when toggling R.

Task 4.2

Create a D latch by extending the SR latch that you created in task 4.1.

Question 1. Include a screenshot of your complete design of the D latch, when both the D and the CLK signals are 0. Explain what happens if D is equal to 0 and you toggle CLK? What happens if CLK is 0 and you toggle D? Are Q and Q' changed? Why or why not?



When D is equal to 0 and CLK is toggled on, Q will become 0 and Q' will become 1. When CLK is toggled off the values remain "in memory". When CLK is 0 and D is toggled, Q and Q' latches onto their previous values. That is, when CLK is 0, both R and S and false, no matter the value of D, so Q and Q' will not change.

Question 2. What happens if CLK is 1 and you toggle D?

Q and Q' will switch values.

Question 3. Explain the benefits with the D latch compared to the SR latch.

The benefit is that there are no illegal states. Q and Q' will always be opposite of each other.

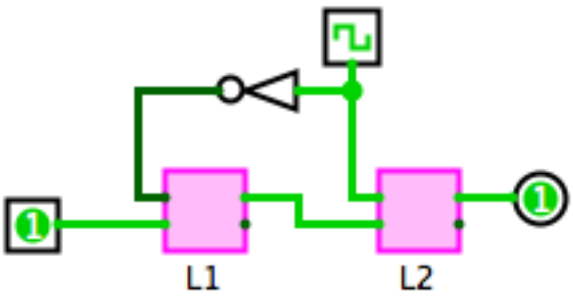
Question 4. Explain the problem with D latches and why they are not used in synchronous sequential logic.

D can change output at any point when CLK is true. This is not desired in synchronous sequential logic where we want state transitions to only happen on clock rises.

Task 5.1

Create a sub-circuit of the D latch. Instantiate two D latches and connect them together to form a D Flip-Flop (see H&H or the lecture slides). Your circuit should have two 1-bit input signals: (i) Din (data in), and (ii) CLK (clock signal). Moreover, it should have one output signal Q.

Question 1. Include a screenshot of your design of the D Flip-Flop, where the two D latches are subcircuits.



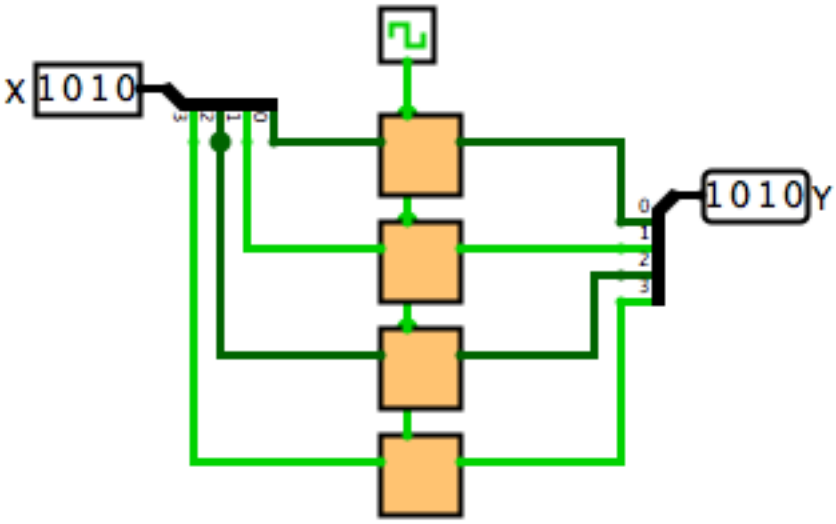


Figure 1: Task 5.2, Register

Question 2. Explain in what circumstances the value of Q changes. Explain by giving concrete examples.

In a D flip-flop, Q is only changed on the rising edge of the clock. That is, the value of Q changes only when CLK is toggled on and D has changed value, otherwise Q remains latched.

For example: if D is 1 and Q is 0, Q will become 1 when CLK is set to 1. After that, Q will not change to 0 until D is 0 and CLK is subsequently set to 1.

Task 5.2

Construct a 4-bit register by reusing the D Flip-Flop as sub-circuits. Your design should have a 4-bit input signal called X, a 4-bit output signal called Y, and a clock signal.

Question 1. Include a screenshot of your design.

See figure 1.

Task 6.1

Question 1. Include a screenshot of your design. Briefly explain how the circuit works.

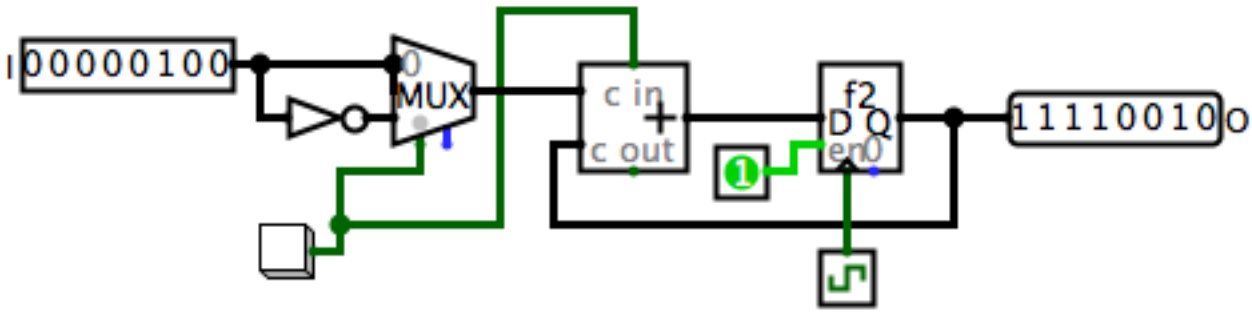


Figure 2: Task 6.1, Synchronous sequential circuit

See figure 2. The circuit selects either input I, or its complement if the button is pressed. If the button is pressed, it also passes 1 to the carry in of the adder, so that we are adding two's complement (eg subtracting) I from O. The output of the register is fed as input to the adder so it can remember its previous value and increment or decrement it.

Question 2. Explain the meaning of a synchronous sequential circuit. Which of the tasks in this lab can be classified as synchronous sequential circuits (assuming that they have a memory)?

Unlike combinatorial circuits which only depend on the input, sequential circuits have different states that depends on the input as well as the current state. When these sequential circuits are synchronous, they are guaranteed to change state only on the rising edge of the clock. This allows for a clear design where the state of the system is unambiguous.

The D-flipflop (task 5.1), the Register (task 5.2), and the final circuit (task 6.1) are examples of synchronous sequential circuits.