CDA 3201L Thursday Section
Lab number 06 -- Sequential Logic Circuits II
06/28/14, John Gangemi

PURPOSE AND OBJECTIVES

Lab assignment number 6 requires the design of a shift register using D flip-flops and then to examine the functionality of the TTL IC 74LS194, a 4-bit Bidirectional Shift Register.

Part A

The design for the shift register must handle a 4-bit word and provide an asynchronous parallel load, serial input, serial output, and parallel out bus. Next modify the design to provide synchronous parallel load, then modify the design again to provide synchronous right shift along with synchronous parallel load.

Part B

Design a circuit using only one 74LS194 IC to perform the following functions by manipulating the control signals to the integrated circuit.

1. Parallel load
2. Circular Left Shift
3. Circular Right shift

COMPONENTS USED

- Integrated Circuits For Demonstration
  - 74LS74 Dual D Flip-Flop with Preset and Clear
  - 74LS00 Quad NAND gate
  - 74LS194 4-bit Bidirectional Shift Register
  - TLC555 LinCMOS TIMER (astable operation @ 1Hz)
    - 1K ohm Resistor
    - 4.7K ohm Resistor
    - 10K ohm Resistor
    - 47 microFarad Electrolytic Capacitor
    - .01 microFarad Metal Film Capacitor
- Integrated Circuits in Report
  - 74LS04 Hex Inverter
  - 74LS32 Quad OR gate
  - 74LS157 Quad 2:1 Multiplexer
  - 74LS153 Dual 4:1 Multiplexer
- 8 x 5mm Red LEDs
- 8 x 470 ohm Resistors

- 5 Volt Regulated DC Power Supply
- Assortment of 22 AWG Jumper Wires
- 2250 point Breadboard
- Logisim Program version 2.7.1

DESIGN DESCRIPTION

Part A

A D flip-flop can hold a single binary value, in order to process four bits of binary data there must be exactly four D flip-flops in the design of a shift register. The most common input to a shift register is the synchronous serial input where data is entered to the shift register in a series, or one bit by one bit, and controlled by the positive edge trigger of a clock.

The connections between the four D flip-flops of the shift register are unique to the direction in which you choose to shift bits in and out of the register. For a synchronous left shift register the output of the first D flip-flop is connected to the input of the adjacent D flip-flop to the left and so forth. This daisy chaining between the D flip-flops ensures that the contents of one is driven to the next only when the clock pulses are high, however this method does incur some delay. The greater the number of latches in the circuit the higher the delay from serial in to serial out, thus to alleviate this problem the method of parallel loading was introduced.

When dealing with parallel loading there are two options for entering data to the registers, asynchronously or synchronously. In the design of the circuit one of the guidelines is to provide an asynchronous load, that is the loading of data into each D flip-flop simultaneously without the need of a clock. Asynchronous parallel loading uses the Preset and Clear pins available to the 74LS74 package and appropriately sets one of two pins to low (active) thus storing a bit in the respective D flip-flop.

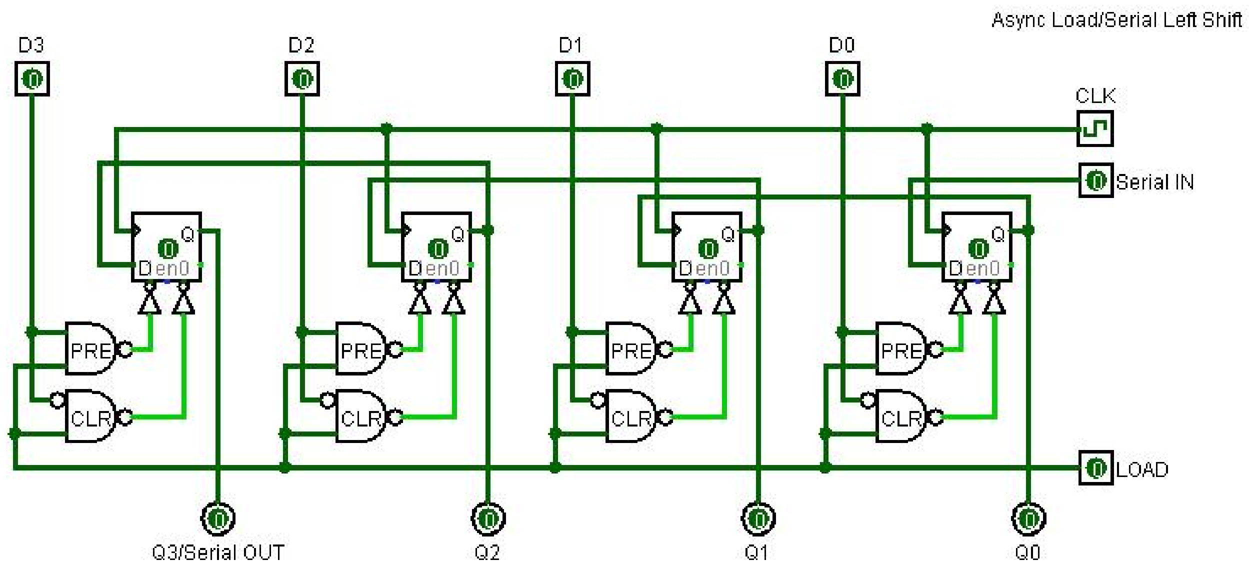
Using an enable bit for the option of asynchronous parallel loading ensures that the logic connected to the Preset and Clear pins is only ever low when the enabled bit is set high preventing unwanted changes to the value stored in a D flip-flop when using the synchronous serial input.

A truth table of asynchronous parallel loading using an enable bit for a single D flip-flop:

Inputs		**Outputs (active low)**	
Enable	**D**	**Preset'**	**Clear'**
0	0	x	x
0	1	x	x
1	0	0	1
1	1	1	0

$$Preset \ = \ (Enable * D)'$$
$$Clear \ = \ (Enable * D')'$$

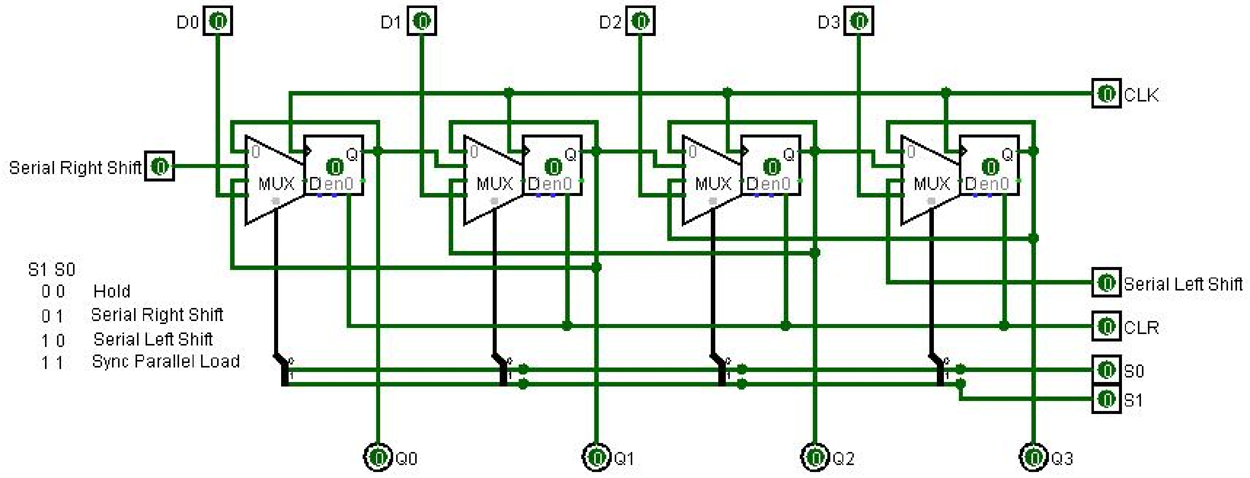
To implement the asynchronous loading each D flip-flop requires two NAND gates and a single Inverter. Please note that the Inverters attached to the D flip-flops in Logisim are there to mimic the behavioral characteristics of the 74LS74 active low Preset and Clear inputs, they are not part of the physical implementation.



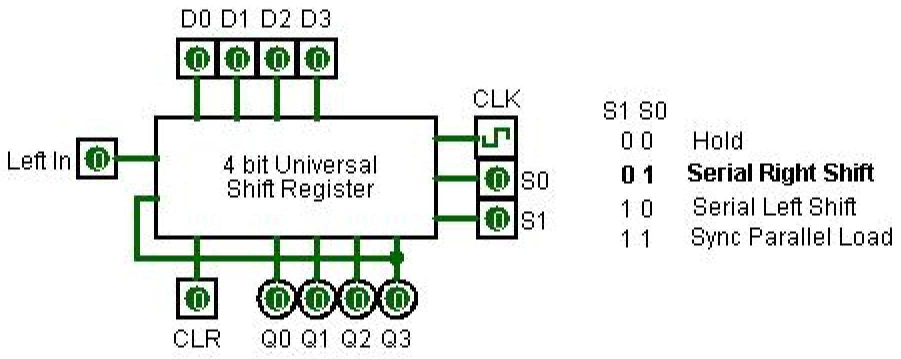
Part B
In order to successfully design a circuit using the 74LS194 IC the pin layout must first be examined. The modifications required for Part A of the lab mimic the functionality of the 74LS194 IC such that the complete design for Part A in Logisim can be used in Part B.

As per the requirements for Part B, a circular shift left or right can be performed by connecting the last stage D flip-flop output (serial out) to the serial input of the first stage D flip-flop (serial in). A loop has essentially been created where the last stage output is "held" in memory as its transferred to the beginning of the loop. How does data then enter the shift register? Using the control signals of the 74LS194 to select the synchronous parallel load data can enter simultaneously to each each D flip-flop and by switching back to synchronous serial input data can be looped around the circuit.

From Part A the complete 4-bit Universal Shift Register with synchronous operations:

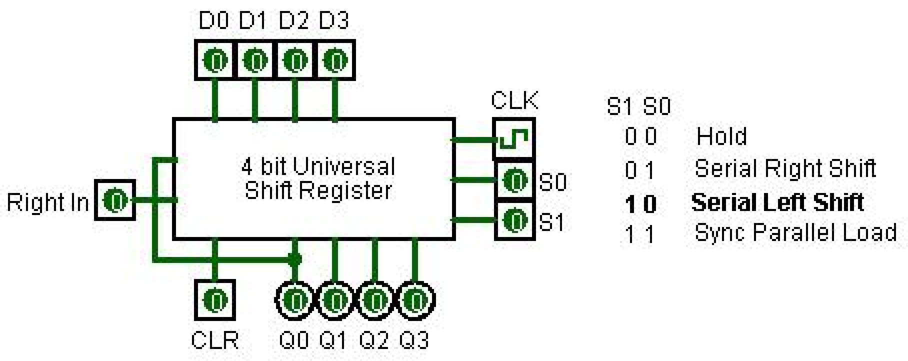


Using the 4-bit Universal Shift Register to perform circular right shift:



As shown the last stage output Q3 is connected to the first stage serial input. Control signals S1=0 and S0=1 select the correct operating mode of serial right shift.

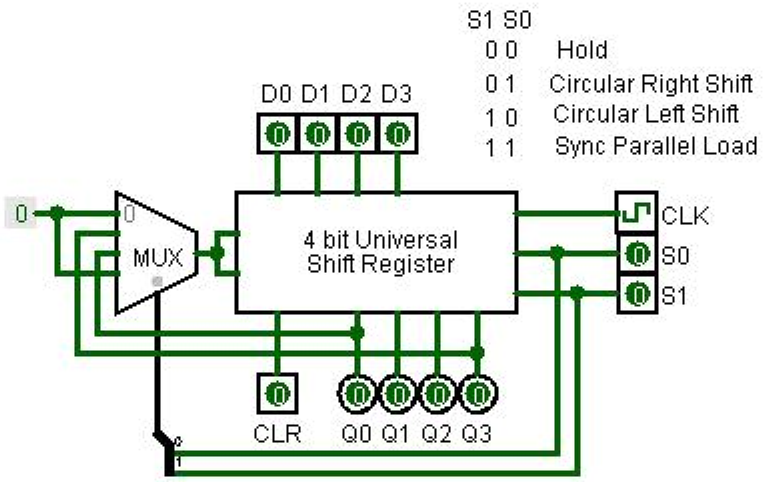
Using the 4-bit Universal Shift Register to perform circular left shift:



As shown the last stage output Q1 is connected to the first stage serial input. Control signals S1=1 and S0=0 select the correct operating mode of serial left shift.

Being able to select between three operating conditions can be done using a 4:1 multiplexer where the control signals are the same as those sent to the 74LS194 IC effectively copying the control signal scheme of the integrated circuit.

S1	S0	Mode
0	0	Hold
0	1	Circular Right Shift
1	0	Circular Left Shift
1	1	Sync Parallel Load

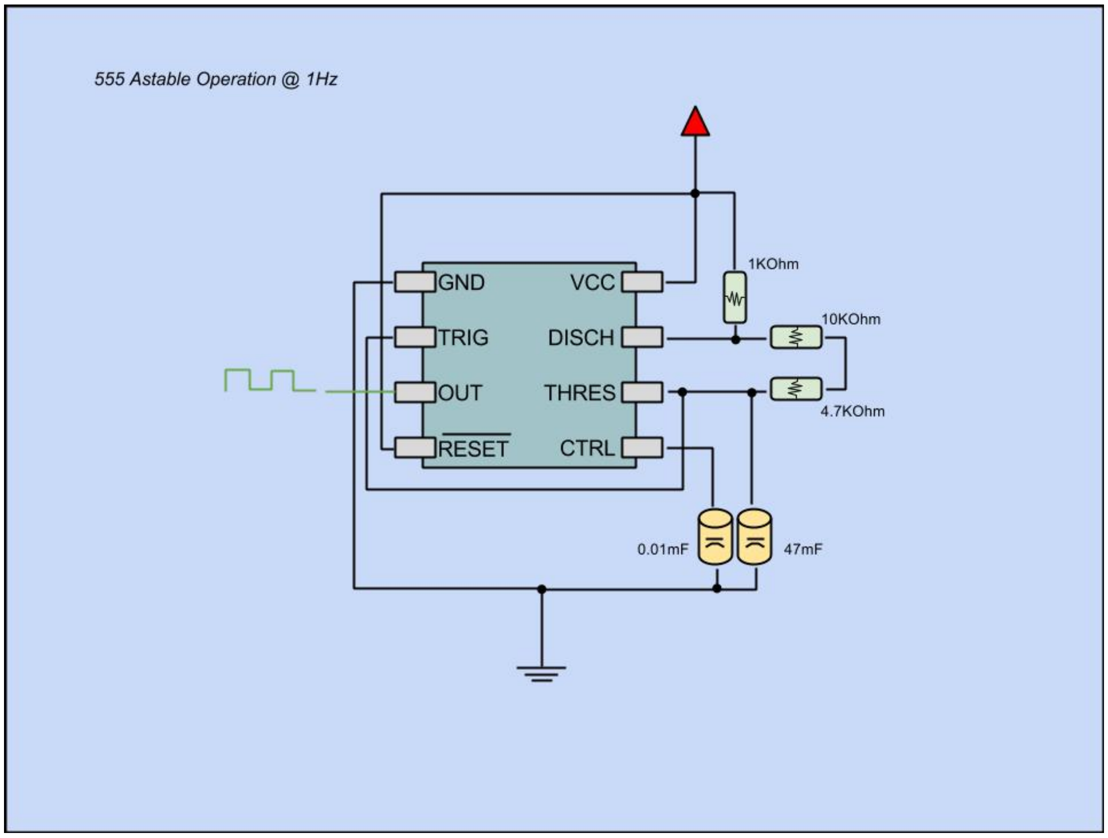


OBSERVATIONS AND DATA ANALYSIS

It was extremely difficult to find a functioning signal generator in the lab that provided consistent clock cycles or all together worked properly. This led to some research on the internet for a signal/function generator but it was immediately apparent that the cost of these units are prohibitive for mild use in and outside lab at this point. I then searched for a solution in the form of an integrated circuit and was pleasantly surprised to that such a device existed, not only was the total cost of supplies acceptable but the chip was capable of performing many functions.

The 555 timer can generate clock signals when configured in an “astable” condition, that is the 555 has no stable state and fluctuates between set and reset (high and low) through the use of a capacitor and two resistors (values selected to operate at 1Hz). It produces a signal pattern similar to a clock and can repeat such wave pattern indefinitely. There is one drawback to using the 555 timer as a clock source and that is the chips capability of producing a near 50% duty cycle but never an exact 50%. The consequence of such limitation is the time from Hi to Lo is a fraction longer than the time from Lo to Hi, however for the purposes of lab this more than acceptable. (Molloy, Derek)

A diagram of the astable 555 timer:



With the inclusion of the 555 timer all circuits have been performing as expected allowing the circuits to be analyzed correctly. The physical implementation of the shift register behaves exactly like the simulated version used in Logisim. Clock pulses alter the current state of each D flip-flop appropriately and the use of an asynchronous parallel load for Part A shows how the clock is “ignored” and data can quickly loaded and sent to the parallel bus outputs.

DISCUSSION AND CONCLUSION

For lab number 6 the use of sequential logic showed how data can be converted from serial to parallel efficiently with the use of D flip-flops and either asynchronous/synchronous inputs.

In Part A of the lab the requirements for the complete design of the circuit were functionally similar to that of the integrated circuit used in Part B, the 74LS194. The design of the circuit for Part A in Logisim helped realize the logic design for the circuit in Part B.

QUESTIONS AND ANSWERS

Part A

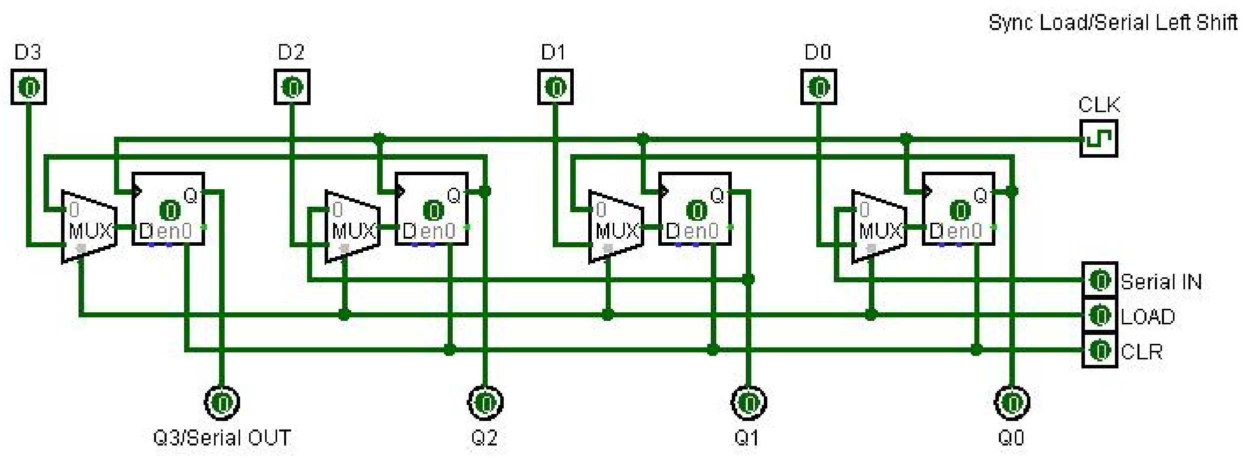
1. How would you modify this design to provide synchronous parallel load instead of asynchronous parallel load?

To get the desired effect of synchronous parallel load one must use the synchronous input to the D flip-flop, that being the “D” input. Changing the parallel inputs from controlling the Preset and Clear pins will allow the parallel inputs to be loaded when the next positive edge of a clock is present across all four D flip-flops. Implementing the switching between serial and parallel inputs is done with a 2:1 multiplexer. The 2:1 multiplexer requires a single control line and has been configured in a manner that when set Hi the multiplexers select the appropriate parallel inputs thus the control signal for the multiplexers is aptly named “LOAD”.

The control signal scheme:

LOAD	Output
0	serial
1	parallel

Logic diagram for synchronous parallel load and left shift:



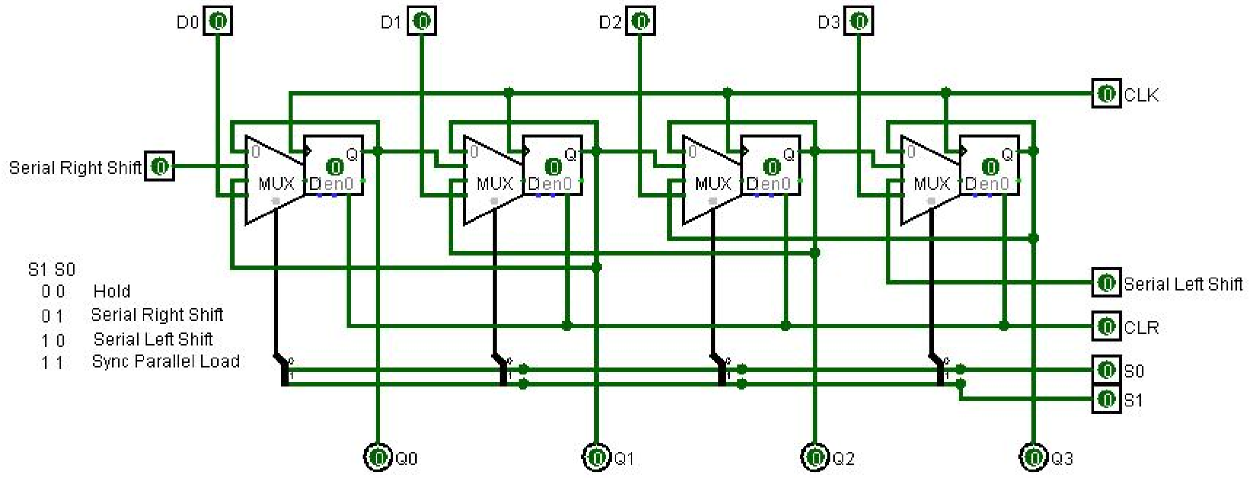
2. How would you modify this design to allow a synchronous right shift along with synchronous parallel load?

For synchronous parallel load it can already been seen that multiplexing the synchronous input of each D flip-flop will suffice. Then for synchronous right shift the connections from output to input of each D flip-flop need to be reversed, however this could potentially cause unwanted results if the serial left input is set Hi or Lo. In order to allow for both serial right and left shift the inputs of the D flip-flops must be multiplexed as before for synchronous parallel load and serial input. More-so the requirement of 3 inputs (serial in right, serial in left & sync parallel) can no longer be accomplished using a 2:1 multiplexer, for 3 inputs at minimum a 4:1 multiplexer must be used.

The control signal scheme:

S1	**S0**	**Mode**
0	0	Hold
0	1	Serial Right Shift
1	0	Serial Left Shift
1	1	Sync Parallel Load

Logic diagram for synchronous parallel load, serial right and left shift:



REFERENCES

- Molloy, Derek. "5. Clock Generator Circuits - EE223 - Digital and Analogue Electronics." EE223 - Digital and Analogue Electronics. Google Sites, n.d. Web. 26 June 2014. <http://ee223.eeng .dcu.ie/experiments/clock-generator-circuits>.