Name: ____________________________________ ID# ________________________

Date Submitted: ____________________ Lab Section # ______________________

CSE 2441 Introduction to Digital Logic Fall Semester 2020

Lab Number 8 – Flip-Flops, Registers, and Counters

100 Points

Due: 10/25/2020, 11:59 PM

NOTE: This is a BitBoard Lab.

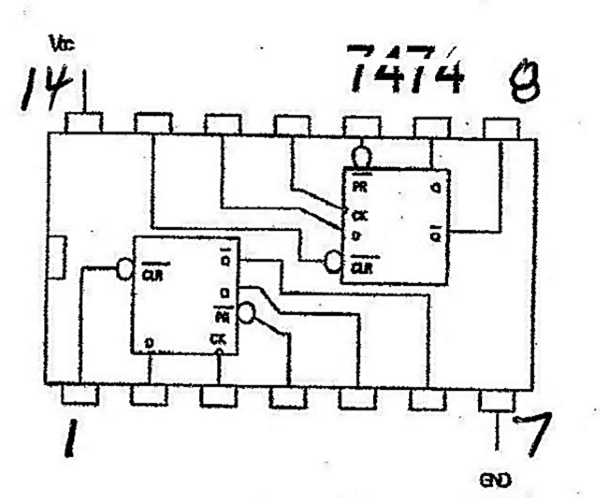
CSE 2441 LABORATORY ASSIGNMENT 6 FALL 2020

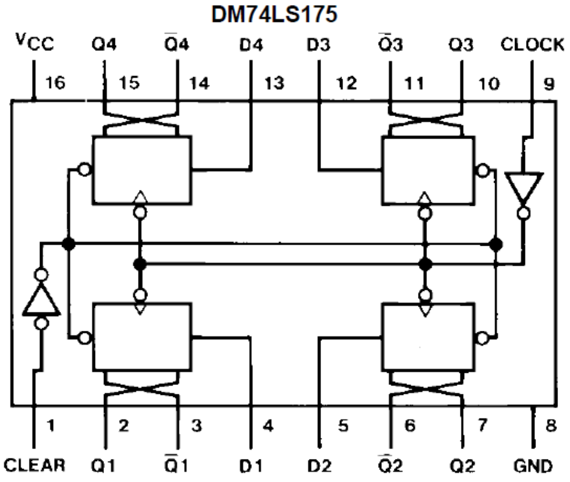
FLIP-FLOPS, REGISTERS, AND COUNTERS

(100 POINTS)

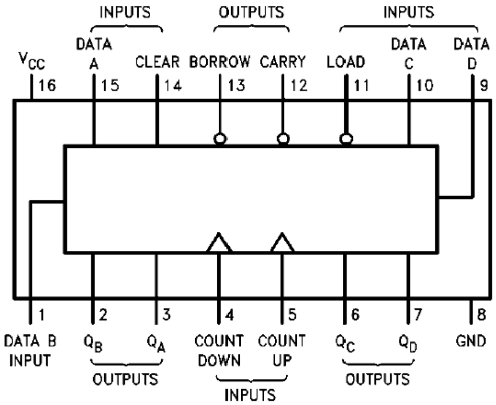
PURPOSE/OUTCOMES -- To study the functional characteristics of D flip-flops and their use in realizing basic registers and counters. You'll design, implement, and test a parallel-in/parallel-out (PIPO) register that will be used later in the term project. You'll also design, implement, and test a binary counter that will also be used in the term project. After completing this laboratory, you'll have an understanding of how flip-flops work and be able to analyze and design basic registers and counters. Devices and circuits studied in this laboratory wil be implemented on The BitBoard.

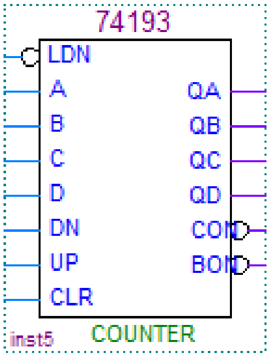
BACKGROUND -- Flip-flops and latches are used as memory devices in sequential logic circuits. The D flip-flop is commonly used in the realization of registers, counters, and other sequential circuits. The figures below show pin-out diagrams for the ***SN7474*** and ***SN74175*** integrated circuits (ICs). The former is an implementation of a D flip-flop. Note that the IC contains two flip-flops and that each flip-flop has data inputs, D, clock inputs, clear and preset inputs, and complementary outputs. The SN74175, or DM74LS175, is a four-bit data register containing four D flip-flops with common clock and clear signals.





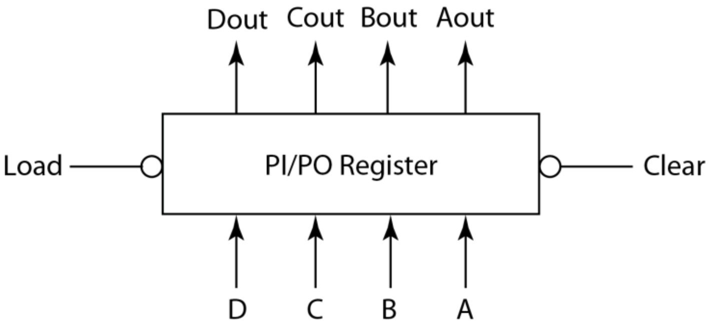
The ***SN74193*** is a four-bit binary up/down counter with the pin-out diagrams shown below. The device counts *up* on the positive edge of a signal applied to the *COUNT-UP* pin when COUNT-DOWN = logic-1 and counts *down* on the positive edge of a signal applied to the COUNT-DOWN pin when COUNT-UP = logic-1. Another feature of the 74193 is that it can be loaded with 0000 by applying logic-1 to the *CLEAR* pin. The counter can be loaded with any four-bit binary number by applying the number to the INPUT pins, *ABCD*, and then applying logic-0 to the *LOAD* pin. The count value is displayed on pins Q_A, Q_B, Q_C, Q_D. Note that *A* is the LSB and *D* the MSB. *CARRY* and *BORROW* outputs are also available but will not be used in this exercise.





DESIGN ASSIGNMENT

1. Design a four-bit parallel-in/parallel-out register with active-low *LOAD* and *CLEAR* signals as shown below. **Note – this register will be used in the term project for the Instruction Register (IR) and the Flag Register (FR).**

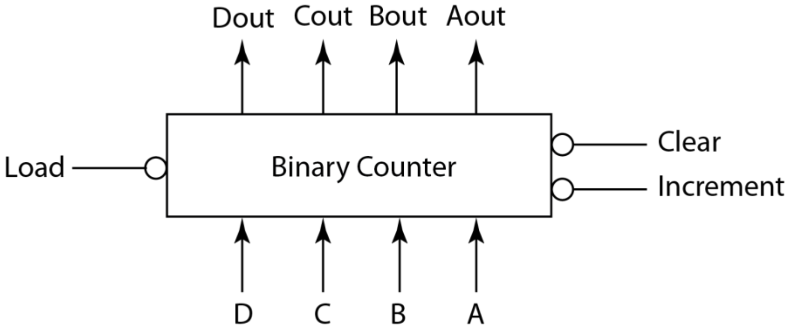


a. Realize your design using 7474 ICs and other logic gates, as needed. Capture and simulate your design using Quartus II to verify its correctness.

b. Now realize your design using 74175 ICs and other logic gates, as needed. Capture and simulate your design using Quartus II.

c. Compare your two realizations in terms of chip count, cost, and wiring complexity. Assume 74175 chips cost $0.75 each and all other chips you use cost $0.50 each. Which realization would you choose to implement on The BitBoard? Why? Capture and save the "best" design as a block symbol file.

2. Design a four-bit binary counter that has the pin-outs and operations shown below. Use an SN74193 and other logic gates as necessary in your design. Capture and simulate your design using Quartus II. Save your design as a symbol file. **Note – this counter will be used in the term project for the Accumulator.**



LAB ASSIGNMENT (80 points) -- Complete each of the exercises detailed below on The BitBoard and record your results for your laboratory report. Take pictures of each of your circuits and include them in your reports.

Part 1 – Learn the functionality of D flip-flop (SN7474) *Preset ($\overline{PR}$)* and *Clear ($\overline{CLR}$)* inputs.

The *Preset* and *Clear* inputs are called asynchronous inputs since they override the *D* input and are not synchronized with the *CK* input. They function as follows.

Applying Preset = 0 and Clear = 1 will place the flip-flop in state Q = 1 ($\overline{Q}$ = 0).
Applying Preset = 1 and Clear = 0 will place the flip-flop in state Q = 0 ($\overline{Q}$ = 1).
Applying Preset = 1 and Clear = 1 will leave the flip-flop state unchanged.
Applying Preset = 0 and Clear = 0 will leave the flip-flop in an invalid state (Q=$\overline{Q}$=1).

Experimentally derive the response of an SN7474 to the following sequence of inputs and record the corresponding values of Q and $\overline{Q}$ below.

PresetClear: 01 – 11 – 01 – 11 – 10 – 11 – 10 – 11 – 01 – 00 – 01 – 11 – 10

Q$\overline{Q}$:

Identify any invalid responses in the above.

Use the following pin assignments when connecting the 7474 to The BitBoard/DE1.

D (pin 2) – SW7	*CK* (*pin 3*) – KEY0
PR (*pin 4*) – SW1	*CLR (pin 1)* – SW0
Q (pin 5) – LEDR7	*$\overline{Q}$ (pin 6)* – LEDR6
Power (pin 14) – +5 V	*Ground* (*pin 7*) – GND

Part 2 – Learn the functionality of D flip-flop (SN7474) *Data (D)* and *Clock (CK)* inputs.

The *Data (D)* input is called a synchronous input since it is read in synchronization with the *Clock (CK)* signal which also determines when any state changes occur. *D* and *CK* inputs are overridden by the Preset and Clear inputs, so these inputs must be held at logic-1 when *D* and *CK* are in use.

Experimentally derive the state table of a SN7474 and record your results in the table below. Use *Preset* and *Clear* inputs to place the flip-flop in the appropriate *Present State* before generating its *Next State*.

Present State (Q)	*Next State (Q*)*	
	PresetClear = 11	
	D = 0	D = 1
0		
1		

Part 3 – Implement and test your "best" register design from the design assignment.

Your test procedure should demonstrate that *CLEAR* works correctly and that *LOAD* works for input patterns *0101*, *0110*, *1001, 1010*, and *1111*.

Part 4 – Implement and test your counter design from the design assignment.

Your test procedure should demonstrate that *CLEAR* works correctly and that LOAD works for input patterns *0101*, *0110*, *1001, 1010*, and *1111*. Also test that your counter counts from *0000* to *1111* and repeats.

REPORT REQUIREMENT

1. Cover sheet (as shown on this assignment)
2. Lab purpose
3. Circuit diagrams of your register and counter designs
4. Simulation results (waveforms) showing the verification of your designs
5. Cost comparison analysis of your two register designs
6. Symbol file input/output diagrams of your designs
7. Test results tables for parts 1, 2, 3, and 4
8. Pictures of your circuits under test