ELEC 2210 - EXPERIMENT 2
Simulation of Logic Circuits

The objectives of this experiment:

This experiment will introduce you to using computer software for simulating digital logic circuits. The objectives of this experiment include:

- Learn how to simulate digital logic circuits using *Multisim* software
- Continue to develop professional lab skills and written communication skills

This experiment is designed for you to work on in a self-paced mode during the week. GTA help will be available at certain times, as posted on the class website.

I. Introduction

Computers are often used in engineering to perform simulation of real-world systems. Some examples of specific software and applications are listed in Table 1.

Table 1. Some simulation software used in engineering.

Simulation Software	**Application**
Multisim	Analog and digital circuits
Simulink	Feedback control systems
PSPICE	Analog and digital circuits (transistor level)
SUPREM	Integrated circuit processing
ANSYS	Mechanical and structural assemblies
EMAPS	Electrical machines
Modelsim	Digital systems (using VHDL & Verilog)

The advantages of simulation are that it is often less expensive and less time-consuming than building and testing the real system and that it is usually much easier to make a large number of changes rapidly, as is often desired for preliminary investigation of a design concept.

The limitation of simulation is that there are always some real-world effects that cannot be accurately modeled in software, so the designer must ultimately build a real prototype for final testing.

In this experiment you will be introduced to *Multisim*, from National Instruments, which is a software package for simulating analog, digital logic, and mixed analog-digital circuits. It allows you to construct a model of a circuit with parts selected from a library (circuit components, logic chips and other parts such as hex displays, LEDs, clocks and switches). Once the parts are interconnected, you can apply a variety of inputs to the circuit and monitor the outputs in several forms, including probes, 7-segment displays and timing diagrams. The latter shows the relationship between various inputs and outputs as a function of time.

An example timing diagram for a synchronous sequential circuit is shown in Figure 1. This particular diagram is taken from the data sheet for the 74193 up/down binary counter manufactured by Texas Instruments. The complete data sheet is available at
http://pdf1.alldatasheet.com/datasheet-pdf/view/51050/FAIRCHILD/74193.html
The manufacturer uses the timing diagram to graphically illustrate the effect of the various inputs on the circuit outputs. Careful analysis of this diagram together with the verbal description and logic tables provided in the data sheet give a complete picture of the operation of this device.

II. Pre-Lab

1. Review your class notes from ELEC 2200 and then design and sketch a circuit diagram of a 3-bit synchronous binary counter using only three JK flip flops (use the JK flip flops in TTL standard part SN74LS73) and two 2-input AND gates (from TTL standard part 74LS08). **Refer to Section 7.3.1 and Figure 7.11 of the ELEC 2200 text (*Digital Logic Circuit Analysis & Design*, by Nelson, et. al.) for a discussion and example of this counter circuit.** Chapters 2 and 6 of that textbook describe the 7408 and 7473 chips, respectively, or refer to the following data sheets on the Internet.
   http://focus.ti.com/docs/prod/folders/print/sn74ls73a.html
   http://focus.ti.com/docs/prod/folders/print/sn74ls08.html

   The counter should have three inputs: an active-low clock input, CLOCK, an active-high “enable” signal, ENABLE, and an asynchronous, active-low reset signal, RESET#. The counter should have three outputs, Q2-Q1-Q0, representing the current state of the counter.
   - When counting is enabled (ENABLE = 1 and RESET# = 1), the count is to increment on each falling edge of CLOCK. When the count reaches 111, it should roll over to 000 on the next falling clock edge.
   - When counting is disabled (ENABLE = 0), the count should remain unchanged.
   - When RESET# is activated (RESET# = 0), the count should reset to and remain 000, and be unaffected by CLOCK and ENABLE.

2. Answer the following about the 74193 binary counter. It will help to download or view pages 1-6 of the data sheet at the URL give above. Show and/or explain your work including intermediate steps.

   a. For the counter to operate, what must be the logic level on the CLEAR (CLR) line (high or low)?

   b. What are the binary and decimal values of the counter output $Q_DQ_CQ_BQ_A$ immediately after the $\overline{\mathrm{LOAD}}$ pulse is applied?

   c. What is the output (binary and decimal) at the end of the COUNT UP interval?

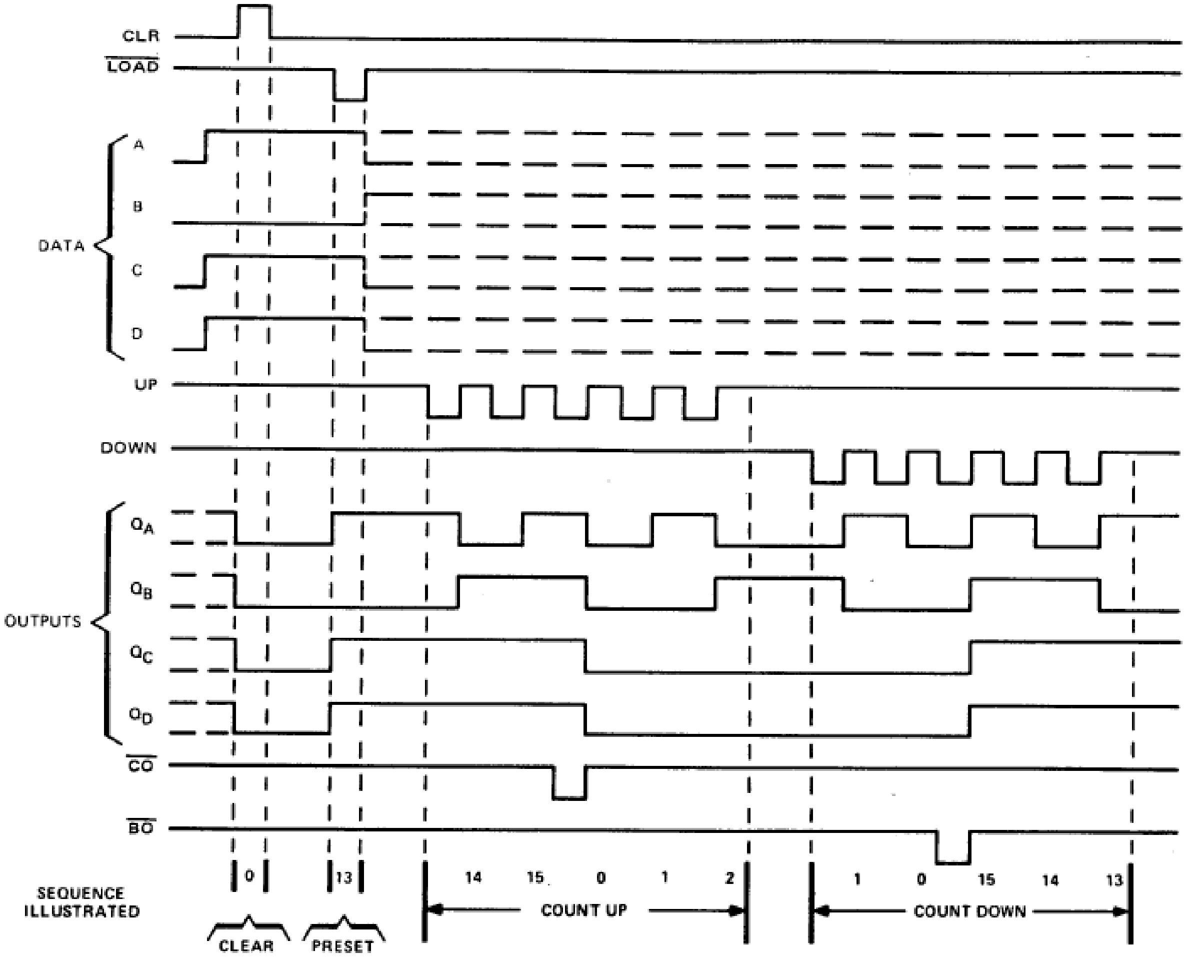
   d. What are the functions of the BORROW OUT $(\overline{\mathrm{BO}})$ and CARRY OUT $(\overline{\mathrm{CO}})$ signals? (See the "Description" section on the first page of the complete data sheet.)

'193, 'LS193 BINARY COUNTERS

typical clear, load, and count sequences

Illustrated below is the following sequence:

1. Clear outputs to zero.
2. Load (preset) to binary thirteen.
3. Count up to fourteen, fifteen, carry, zero, one, and two.
4. Count down to one, zero, borrow, fifteen, fourteen, and thirteen.



NOTES: A. Clear overrides load, data, and count inputs.

B. When counting up, count-down input must be high; when counting down, count-up input must be high.

Figure 1. Example timing diagram for a binary counter.

III. Lab Exercise

This experiment will be performed on a computer using *Multisim* software. The PC labs in Broun 308, 310 and the basement are available for your use, except when reserved for other classes. *Multisim* is installed on each of the PCs in these labs, as well as in the ELEC 2210 lab.

The *Multisim* tutorial is designed to be self-explanatory, and regardless of the availability of your instructor, you should complete this experiment on your own time. Report any computer and network problems in 308/310 to the engineering network service[*]. Report any problems in the ELEC 2210 lab room to your GTA.

STEP 1.

Complete the *Multisim* "10-minute Tutorial," contained in file "Multisim_Tutorial.pdf." Work through the tutorial to learn how to use *Multisim* to simulate a simple digital logic circuit.

STEP 2.

In *Multisim*, draw and simulate the 3-bit binary counter that you designed in the pre-lab. Use source type INTERACTIVE_DIGITAL_CONSTANT for the CLOCK, ENABLE and RESET# inputs, so that you can change the inputs interactively. Connect the outputs to probes and to the three least significant input bits of a hex display digit (tie the 4th input of the hex display digit to 0) to capture and display the states of the counter. Connect all 3 inputs and the 3 outputs to a logic analyzer, to capture and display a timing diagram.

Your counter is to perform as follows:

(1) When the RESET# input is active (i.e. 0), the output should reset to 000. (Test this when the count is 111 to ensure that all three flip flops reset to 0.) Also, verify that changes in ENABLE and CLOCK to not affect the circuit outputs while RESET# is active.

(2) With the RESET# input set back to 1 and ENABLE set to 1, verify that the circuit counts up each time the CLOCK changes from 1 to 0. Check that the proper count sequence is produced: 000-001-010-011-100-101-110-111-000-… (rolling over from 111 to 000).

(3) Verify that the count remains unchanged if there are CLOCK pulses while RESET# = 1 and ENABLE = 0.

(4) When your circuit is working properly, use the text tool to put your name and the date on the diagram. Then save your circuit on your H: drive and/or your personal USB drive.

(5) Print your circuit and the timing diagram to include in your lab report. You may need to use screen capture to get the hardcopy.

STEP 3.

In *Multisim*, draw and simulate the 4-bit binary counter circuit shown in Figure 2, which is based on the 74193 binary counter you studied in the pre-lab work. Use a NI ELVISmx Digital Writer

[*] Email: admin@eng.auburn.edu

instrument to drive the 8 inputs, so that you can change the inputs interactively. Connect the outputs to a DCD_HEX_GREEN indicator to display the count value. Connect all 8 inputs and the 4 outputs to a logic analyzer, to capture and display a timing diagram.

Your counter must perform as follows:

(1) When the CLR input is 1, the output should reset to 0.

(2) When the CLR input is set back to 0, the circuit should count up for each 0-to-1 transition on the UP input, and count down for each 0-to-1 transition on the DOWN input. By changing the inputs in the proper order, the timing diagram shown in Figure 3 should be generated.

(3) When your circuit is working properly, use the text tool to put your name and the date on the diagram. Then save your circuit on your H: drive and/or your personal USB drive.

(4) Print your circuit and the timing diagram to include in your lab report. You may need to use screen capture to get the hardcopy.

STEP 4.

Email your two circuits (.ms11 files) from Steps 2 and 3 to your GTA as attachments. The subject line in your email should read "2210 Expt 2 Circuits from *userID*." This counts as part of your lab report submission. Your GTA's email address is on the lab web page.

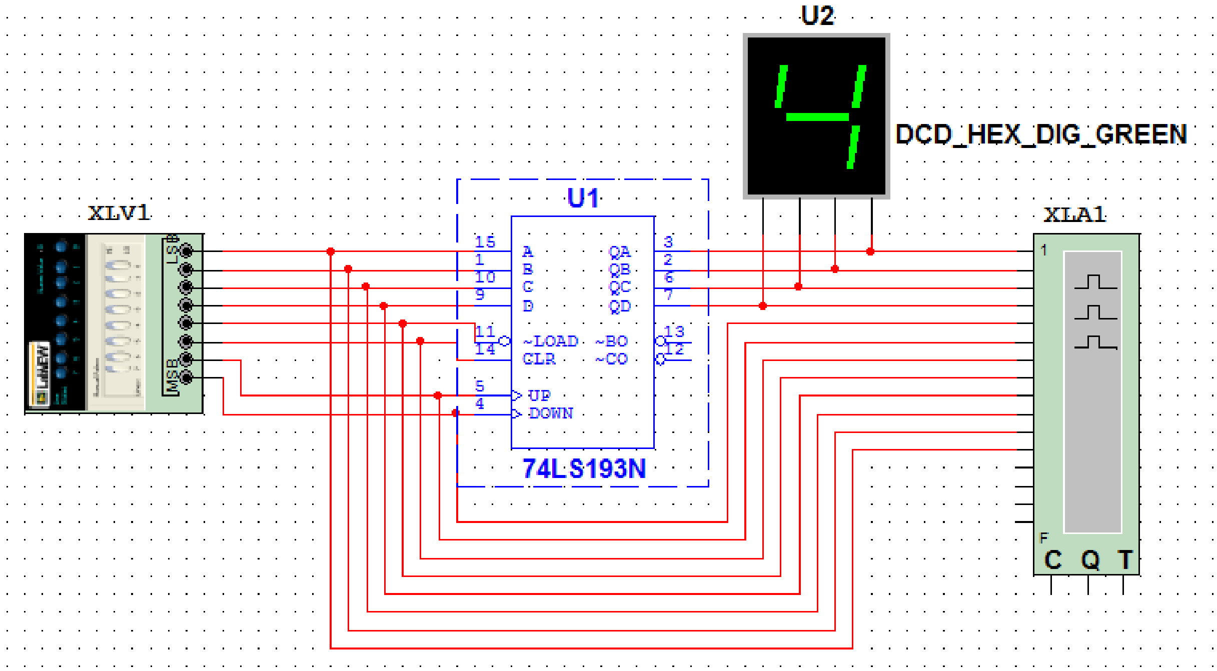
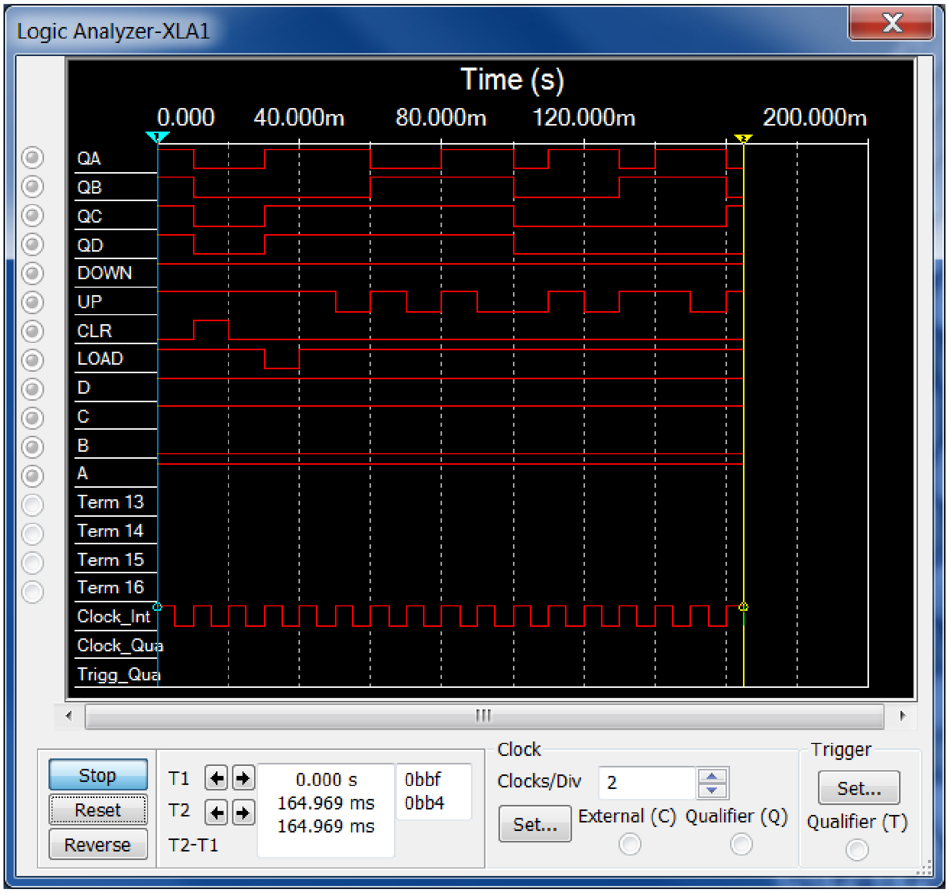


Figure 2. Test Circuit



T1 0.000 s 0bbf
T2 164.969 ms 0bb4
T2-T1 164.969 ms

Clock
Clocks/Div 2

Figure 3. Test Circuit Output (Logic Analyzer Display)