Experiment 11
Digital Logic Circuits

1 Motivation

Logic circuits are used extensively. The most obvious is computers, but almost every modern consumer electronics device has some form of digital circuitry. It is difficult to overstate the importance of these nonlinear circuits. The primary aim of digital circuitry is to create stable states that are easily identifiable as a digital "1" or digital "0". These map to "high" and "low" voltages. As long as a "high" or "low" voltage is above or below a defined threshold, the state is resolved. This makes logic circuits relatively robustness to interference and noise.

2 Background

The integrated logic circuits you will use in this experiment are members of the Transistor-Transistor Logic (TTL) family. The test circuit board chassis includes a mixture of circuits from the 74LS00 series ("Low-power Schottky," introduced 1976) and 74HCT00 series ("High-speed CMOS, TTL compatible," introduced 1982). These are manufactured in the form of 14-pin to 24-pin plastic epoxy packages, originally as "dual-in-line" (DIP) packages and now primarily as miniaturized surface-mount integrated circuits.

Basics for 7400-Series Digital Logic Integrated Circuits

- 7400-series TTL integrated circuits operate with a single DC voltage supply, nominally $V_{CC} = +5$ V. They can be permanently damaged if the supply voltage is raised above above +7.0 V or made negative even briefly (below −0.5 V for 74HCT). For this reason, the test circuit board you will use is enclosed in a chassis with its own DC voltage supply to minimize the chance of damage.
- The threshold voltages associated with the logic outputs and inputs are:
  - +4.75 V: Voltage V_{CC}; also maximum allowed input voltage
  - 0.00 V: Ground; also minimum allowed input voltage
  - +2.40 V: V_{OH}, min. output voltage for logical ONE @ 0.4 mA
  - +2.00 V: V_{IH}, min. input voltage for logical ONE @ 0.04 mA max I_{IN}
  - +1.30 V: V_S, threshold voltage (typical value – varies)
  - +0.80 V: V_{IL}, max. input voltage for logical ZERO @ −1.6 mA max. I_{SINK}
  - +0.40 V: V_{OL}, max. output voltage for logical ZERO @ −16 mA
- The time it takes for a logic signal's transmission through a gate is called the *propagation delay*. In the 74LS and 74HCT series, the minimum propagation delay is 10 ns for a basic gate, like a NAND. The delay increases in more complicated integrated circuits.
- The current into or out of a logic gate is largest when held LOW, since a transistor is switched "on" (conducts current). A 7400 series LOW output can sink up to 16 mA, and an input can source up to 1.6 mA. This implies a maximum "fanout" of 10 inputs connected to one output.
- Unless you are not using an entire gate or section of a TTL integrated circuit, all of its unused inputs must be connected to LOW or HIGH. *

*Unconnected TTL inputs will float to logical HIGH. A circuit constructed with floating inputs might appear to work, but it will not be reliable. As Horowitz and Hill state in *The Art of Electronics*, leaving digital inputs unconnected is "foolish and dangerous" (Chapter 10, 3rd edition).

3 Equipment

For this experiment, you will use:

- One "logic board" chassis for testing logic circuits
- One Tektronix MSO 2014B Digital Storage Oscilloscope
- One Tektronix P3010 10X scope probe

4 Procedure

All of the digital circuits needed for this experiment are mounted on a circuit board inside the "logic board" chassis. The $V_{CC} = +5$ V power supply is also located inside the chassis. Do not attempt to open the chassis on your own. If you are interested to see the board and circuity, ask your lab instructor to open one, if available.

The logic board chassis contains the following functions (not all used in this experiment):

16 – 2-input NAND gates

3 – 3-input NAND gates

8 – JK flip-flops

1 – 4-bit adder

1 – UART (universal asynchronous receiver-transmitter)

10 – indicator LEDs

4 – push-button switches

4 – toggle switches

Please do not introduce external voltages into the logic board. Key features and accuators on the logic board chassis are:

- **Gray colored connectors are inputs**: Use banana-connector patch cables to connect logic signals from various locations on the board (but not external sources). The input to a gate can be the output of a different gate. This is how more complicated digital circuits are constructed. Inputs can also come from pull-up and pull-down circuitry associated with the switches described below.
- **Green colors connectors are outputs**: The outputs of the digital circuits are made available as banana-connector terminals. You can monitor the logic level of an output using the LED indicators described below. Outputs from one gate can be used as inputs to other gates to build more complex logic circuits. Each output can be connected to multiple inputs The number of inputs that can be be connected to one single output is called the *fanout* capacity. To prevent damage by accidental shorting, a 47 Ω resistor is wired in series with each output. This reduces the fanout capacity to 4.
- **Light emitting diodes** (LEDs) can be used to indicate the digital level at selected points in a circuit: LED "on" indicates logical ONE, while LED "off" indicates logical ZERO.
- **Push-button and toggle switches** can be used to source logical ONE or logical ZERO inputs for other circuits on the board.

Pull-up resistors are installed internal to the chassis that connect unused inputs to HIGH (logical ONE). This means you do not have to worry about floating inputs.

The exercises below are designed to reveal the relative simplicity of a digital circuit and illustrate how they can are combined to perform complex operations.

1. Experimentally verify the truth table for a 3-input NAND gate.
2. Use 2-input NAND gates to construct circuits that perform the following logical functions. In each case, draw the circuit diagram and experimentally verify the truth table.
   (a) 2-input AND gate: $A \text{ AND } B = A \cdot B$.
   (b) 2-input OR gate: $A \text{ OR } B = A + B$.
   (c) 2-input NOR gate: $A \text{ NOR } B = \overline{A + B}$.
   (d) 2-input XOR gate: $A \text{ XOR } B = A \oplus B$.
3. Use NAND gates to construct circuits that perform the functions listed below. Try to minimize the total number of gates required, e.g., using a 3-input NAND gate where it helps simplify. For each case, draw your circuits and write out at least 8 states of the truth table. (You can use "Don't Care" entries to simplify portions of the table.) Experimentally verify the states you listed in your table.

   (a) $(A \cdot B) + C$
   (b) $((A \cdot B) + C) \cdot D$
   (c) $(A \cdot B) + (C \cdot D)$
   (d) $(A \cdot B \cdot C) + D + E$.

4. A logic comparator is a circuit that assesses the values of two binary numbers, A and B, and outputs a ONE if $A = B$ or a ZERO if $A \neq B$. Consider a 2-bit comparator: such a circuit must have 4 inputs, one for each bit of each number. Let A_1 and A_0 be the most and least significant bit of A, respectively, and similarly for B_1 and B_0.
   (a) Write down a logical expression for C in terms of A_0, A_1, B_0, and B_1. (It might help to write out a truth table if you seem to be stuck.)
   (b) Design a circuit that uses 2-input or 3-input NAND gates, i.e, those available on the logic board, to perform a 2-bit comparison. Draw your circuit schematic in your lab notebook. (*Hint:* The XOR circuit in part 2d is close to a 1-bit comparator – compare the truth tables! You will need two such circuits, one for the first bit and another for the second bit. You may use XOR as a block in your schematic.)
   (c) Construct the circuit and experimentally verify your design.
5. A basic circuit for adding two bits, called a *half adder*, has the truth table shown in Table 1. A half adder will add bits A_0 and B_0 to produce their sum S_0 and a carry bit C_0.

Inputs		Outputs	
A_0	B_0	C_0	S_0
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Table 1: Truth table for a half adder.

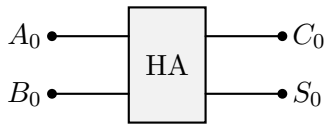

Figure 1: A block schematic symbol for a half adder.

   (a) Write down logical expressions for S_0 and C_0 in terms of A_0 and B_0.
   (b) Design a half adder circuit that uses 2-input or 3-input NAND gates available on the logic board.
   (c) Construct your half adder design and verify that it produces the truth table shown in Table 1.

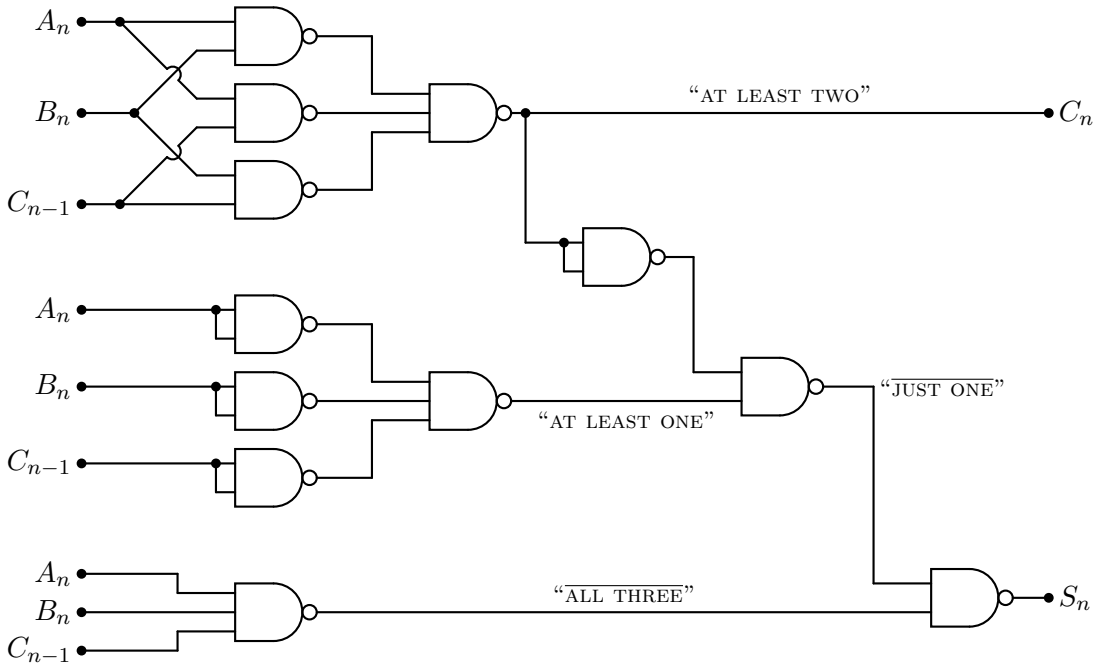

Figure 2: A full adder built with NAND gates. Labels describe the functions of blocks.

6. The circuit shown in Figure 2 is referred to as a *full adder*. A chain of N full adders can add two N-bit numbers A and B one bit at a time, starting from the least significant digit, giving their N-bit sum.
   Each full addition step combines three inputs:
   - A_n, the nth bit from A
   - B_n, the nth bit from B
   - C_{n-1}, the carry bit from the previous step

   and generates two outputs:
   - S_n, the nth bit of the sum of A and B
   - C_n, a carry bit for the next step.

   The full adder is an extremely important circuit found in every digital calculator and computer.
   (a) What should you do with the carry input to the LSB of an N-bit adder?
   (b) For the last addition in an N-bit sum, the final carry bit C_{N-1} is also called the *overflow bit*. Why?
   (c) Write down the truth table for a full adder. You may reference Figure 2, or you can start from the functional definition of a full adder.
7. The 4-bit adder available on the logic board is a single integrated circuit (74LS283) that performs the function of four full-adder circuits like the one in Figure 2. A schematic diagram of four full adders connected to form a single 4-bit adder with carry is shown in Figure 3. Notice that the carry input and the carry output allow you to connect N 74LS283 integrated circuits together to obtain a $4N$-bit adder.

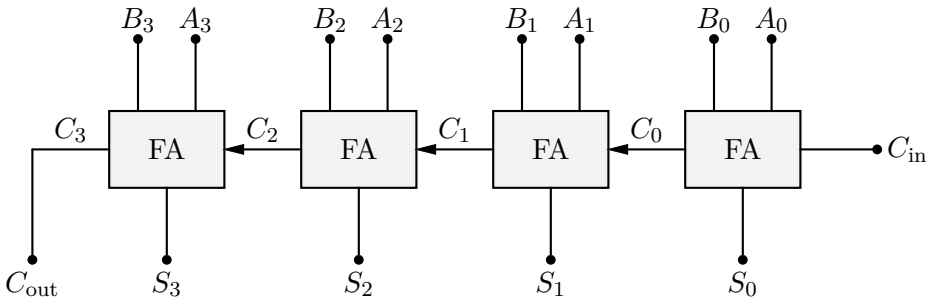

Figure 3: A block diagram of a 4-bit adder built from full adders.

Use the 4-bit adder to sum the following pairs of decimal numbers:

(a) $6 + 7$
(b) $13 + 5$
(c) $11 + 14$

Now use the adder to sum signed numbers. Use two's-complement encoding for negative numbers, e.g. $-3_{10} = \texttt{1101}_2$, which restricts the signed adder to the range -8_{10} to $+7_{10}$. (There are 16 distinct numbers using a 4-bit word, with one bit used for sign "+/−".) Add the following pairs of decimal numbers:

(d) $(-3) + 5$
(e) $3 + (-5)$
(f) $(-2) + (-3)$

Observe what happens if you just try to "sign" a number using the most-significant bit, e.g., $-3_{10} = \texttt{1011}_2$ (called "sign-and-magnitude" representation). Why is two's-complement the preferred representation for negative numbers?

If you have time, complete one or more of the optional exercises below:

8. (Optional) Measure the propagation delay of a NAND gate using the oscilloscope's single-sweep trigger mode.
9. (Optional) Construct a "ring oscillator" using seven NAND gates connected in series as inverters. Measure the oscillation frequency using the oscilloscope and calculate the propagation delay per gate.
10. (Optional) Design a circuit which returns the two's-complement negation of an input 4-bit number. You may use all of the circuits available on the logic board. Construct and test your circuit design.
11. (Optional) A *multiplexer* is a very useful circuit which has two sets of inputs: address lines, A, and data lines, D, along with a single output, Q. The address lines choose which data line is copied to the output. The number d of data lines is at most 2^a.
    (a) Write down the truth table for a 2-input multiplexer with address line A and data lines D_0 and D_1: $Q = D_1$ if A is HIGH or $Q = D_0$ if A is LOW.
    (b) Design, build, and test a 2-input multiplexer circuit.
    (c) Design, build, and test a 4-input multiplexer circuit.

URLs for relevant datasheets

• http : //www.ti.com/lit/ds/symlink/sn74s00.pdf
• http : //www.ti.com/lit/ds/symlink/sn74ls283.pdf