ENGR 210
Lab 10: Digital Logic

In this lab you will first investigate the operation of some basic digital logic gates. In particular you will investigate and use 7400 series digital logic and a transparent D-type latch. These components are then used to implement a simple digitally controlled programmable lock. The lock uses a 4-bit digital code can be any sequence of four ones and zeros and is stored in a D-type latch. You will build and test specific parts of the lock in this lab; finally, you will use a computer simulation of the entire circuit which will allows you to input a four bit code which the circuitry will compare with your programmed code. If the input code matches the stored code, the lock opens. If the input does not match the stored code, the lock will not open.

A. BACKGROUND

1. Some Basics About Digital Integrated Circuits (ICs)

In class and in homework assignments we have studied the fundamental logic operations, namely, the INVERT, AND, and OR, as well as the negated versions of the latter two, namely NAND and NOR. Integrated circuit devices which enable the implementation of these functions are available in several *families* of devices. Types of logic families you may work with are *complementary MOS logic* (CMOS), *transistor-transistor logic* (TTL), and *emitter-coupled logic* (ECL). The major reasons for choosing one family over the other for a specific application is the combination of speed of operation and power consumption. Table 1, below, lists approximate maximum speed of operation and typical power per gate for each of these families.

	CMOS	TTL	ECL
Power/gate (typical)	0.001 mW	5.0 mW	25 mW
Maximum frequency	30 MHz	100 MHz	250 MHz
Max. gate delay	15 ns	4.5 ns	1.5 ns

Table 1. Selected Logic Family Characteristics

When you read the top of the IC, you will notice that in most cases the "part number" has some letters in the middle of the numbers, e.g., 74LS00, or 74HC04. These letters describe which family of devices the digital IC belongs to. For example, LS stands for Low power Schottkey and HC stands for High speed CMOS. For various reasons, the original TTL family is now longer made — it has been replaced by a number of CMOS varieties such as HC which can electrically replace the original TTL integrated circuits. We won't concern ourselves with these somewhat esoteric issues now, but it is useful to know that a 74LS00, a 74S00, a 74HC00, a 74C00, etc. will all perform the same NAND operation and, for our purposes in this course, are electrically identical. In this lab you will probably use CMOS devices from the

HC or HCT family which electrically resemble the TTL family. One of the most common differences between the logic families is what voltage levels represent logical 0 and logical 1. In TTL logic circuits a logic 1, or high state, is represented by +3.4V (typically) and a logic zero by 0.2V (typically). In CMOS logic families logic 1 is very near 5 volts, and logic 0 very near 0 volts. These high and low thresholds are different for each logic family and you will measure them in this lab.

(a) 8 pin DIP (such as 741); (b) 14 pin DIP

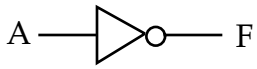
Figure 1. Physical configuration of 8

Most of the common digital ICs come in 14 pin dual inline packages (DIPs). In almost all cases, pin 7 is ground and pin 14 is for power input (+5 volts). There are some exceptions, of course, but this is the most common configuration. You should remember from your use of the 741 op amp how pin 1 can be identified. It is either the pin with a little dot (indentation) next to it, or it is the pin directly to the left of the "U" notch on the IC package. This is shown in Figure 1.

2. Review of Basic Digital Logic Functions

This section describes some of the basic digital electronic functions that are implemented in the 7400 series of devices. A memory device that will be used in this lab is also described. Each logic function is described in terms of its truth table. Its circuit symbol is also given.

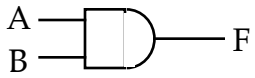
a. NOT gate, or inverter



A	F
0	1
1	0

Figure 2. A 7404 integrated circuit (IC) contains 6 inverters.

b. 2-input AND gate



A	B	F
0	0	0
1	0	0
0	1	0
1	1	1

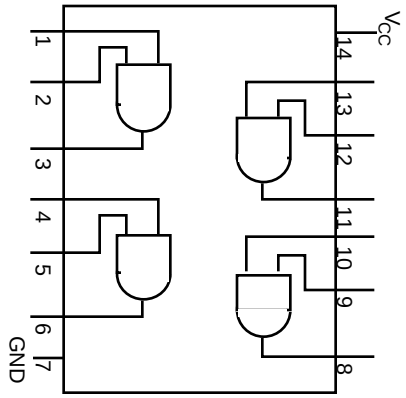
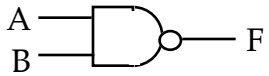


Figure 3. A 7408 IC contains 4 2-input AND gates.

c. 2-input NAND gate. Note that this is logically equivalent to an AND gate followed by a NOT.



A	B	F
0	0	1
1	0	1
0	1	1
1	1	0

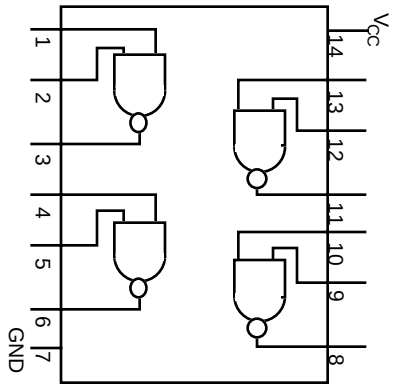
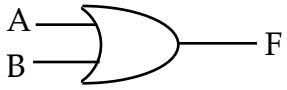


Figure 4. A 7400 IC contains 4 2-input NAND gates.

d. 2-input OR gate



A	B	F
0	0	0
1	0	1
0	1	1
1	1	1

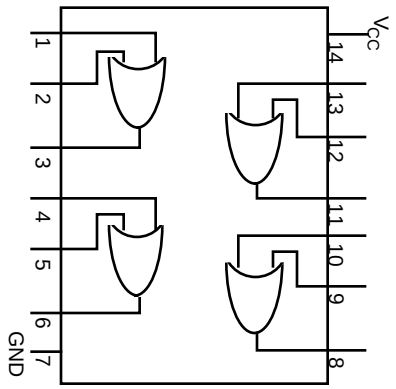
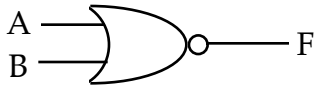


Figure 5. A 7432 IC contains 4 2-input OR gates.

e. 2-input NOR gate. This is just an OR followed by a NOT



A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

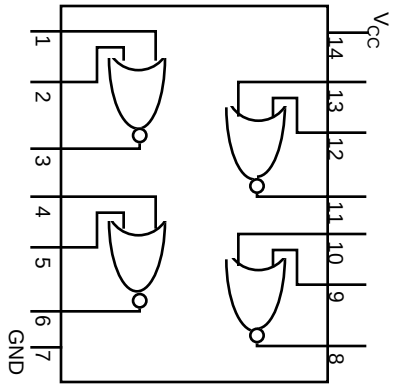
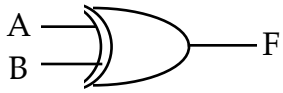


Figure 6. A 7402 IC contains 4 2-input NOR gates.

f. 2-input XOR (EXCLUSIVE OR) gate



A	B	F
0	0	0
1	0	1
0	1	1
1	1	0

Figure 7. A 7486 IC contains 4 2-input XOR gates.

g. The D-Latch

The D-latch is a rather different type of device from those discussed above. For all of the previous devices, called *combinational logic* devices, the outputs depend only on the combination of the present inputs. The circuit that you will study in this lab involves *sequential logic*. A sequential circuit has memory, i.e., its output depends not only on the present inputs, but also on the past inputs.

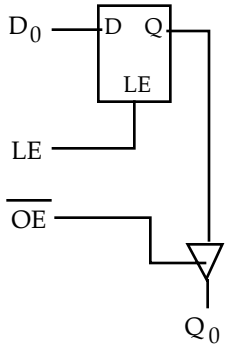


Figure 8. Logic diagram of D-latch.

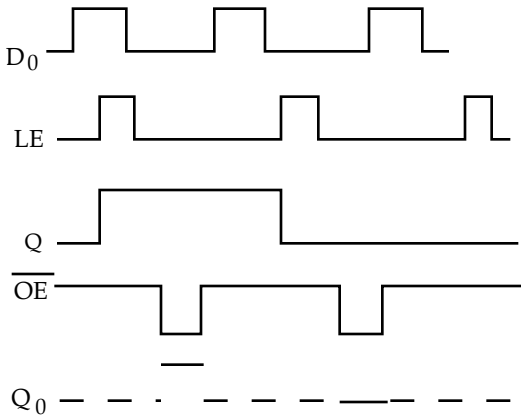


Figure 9. Timing diagram of D-latch. (Note that Q_0 is undefined when OE* is high.)

A logic diagram of a D-latch is shown in Figure 8. We will begin our consideration of the D-latch by examining the input section, called a D flip-flop. Data, D_0, is provided at the D input. This input has no effect on the output, at Q, unless permission is granted by a "latch enable" or "clock" signal, LE. If the voltage at the enable input is high (5 V), the signal applied at the D input appears at the output, Q. After the data has been placed in the flip-flop, taking LE low (0 V, or ground) stores it. If the voltage at the enable input is kept low (0 V, or ground), the output remains at the previously enabled value independent of the present data level. Thus the flip-flop "remembers" a past input. Figure 9, shows an example of data input, D_0, enable signal, LE, and output signal, Q, of the flip-flop as a function of time. a diagram such as this is known as a ***timing diagram***..

The second part of the D-latch is the output section. As shown in Figure 8, this section is a gated amplifier. The output of the D flip-flop is connected to this amplifier, which also is connected to a line labeled OE*, or "output enable." The asterisk means that this pin is "low true," i.e., for something to happen, this pin must be taken to 0 V, or ground.) If the OE* line is not low, the data that is stored in the flip-flop can not be read and the output is undefined. It should be noted that data stored in the flip-flop remains in the flip-flop, even after it is read, until new data is stored using the procedure described in the preceding paragraph.

The actual latch (the 74573) you will use in this lab is shown in Figure 10. The 74573 consists of eight D-type transparent latches with tri-state outputs. When the LE input is HIGH (+5 volts), data at the Dn inputs enters the latches. When the LE input is HIGH the latches are transparent, i.e., a latch output will change state each time the corresponding D-input changes. When the LE input is LOW (0 volts) the latch stores the information that was present at the time the LE input changed from HIGH to LOW. When the OE* input is LOW, the contents of the eight latches are available at the Qn outputs. When the OE* input is HIGH, the outputs go to the high-impedance OFF state. The LE and OE* inputs operate independently of each other. For example, operation of the OE* input does not change the contents of the latches. The 74373 is identical to the 74573 except that the connections to the pins (the pin numbering changes). The Electronics Workbench simulation uses the 74373 since it does not support the 74573. Also, the Electronics Workbench simulated 74373 does not use standard notation for the control inputs, i.e., LE is labeled C, and OE* is labeled OC. These are minor differences and do not affect the electrical operation of the integrated circuit.

3. A Programmable Digital Lock

In this lab you will build and test components of a programmable digital lock. The complete circuit is simulated in the Electronics Workbench file for this lab available on the course Web page. The actual circuit looks quite complex but actually consists of three more readily understood components: (1) data input, (2) data memory, and (3) data comparison.

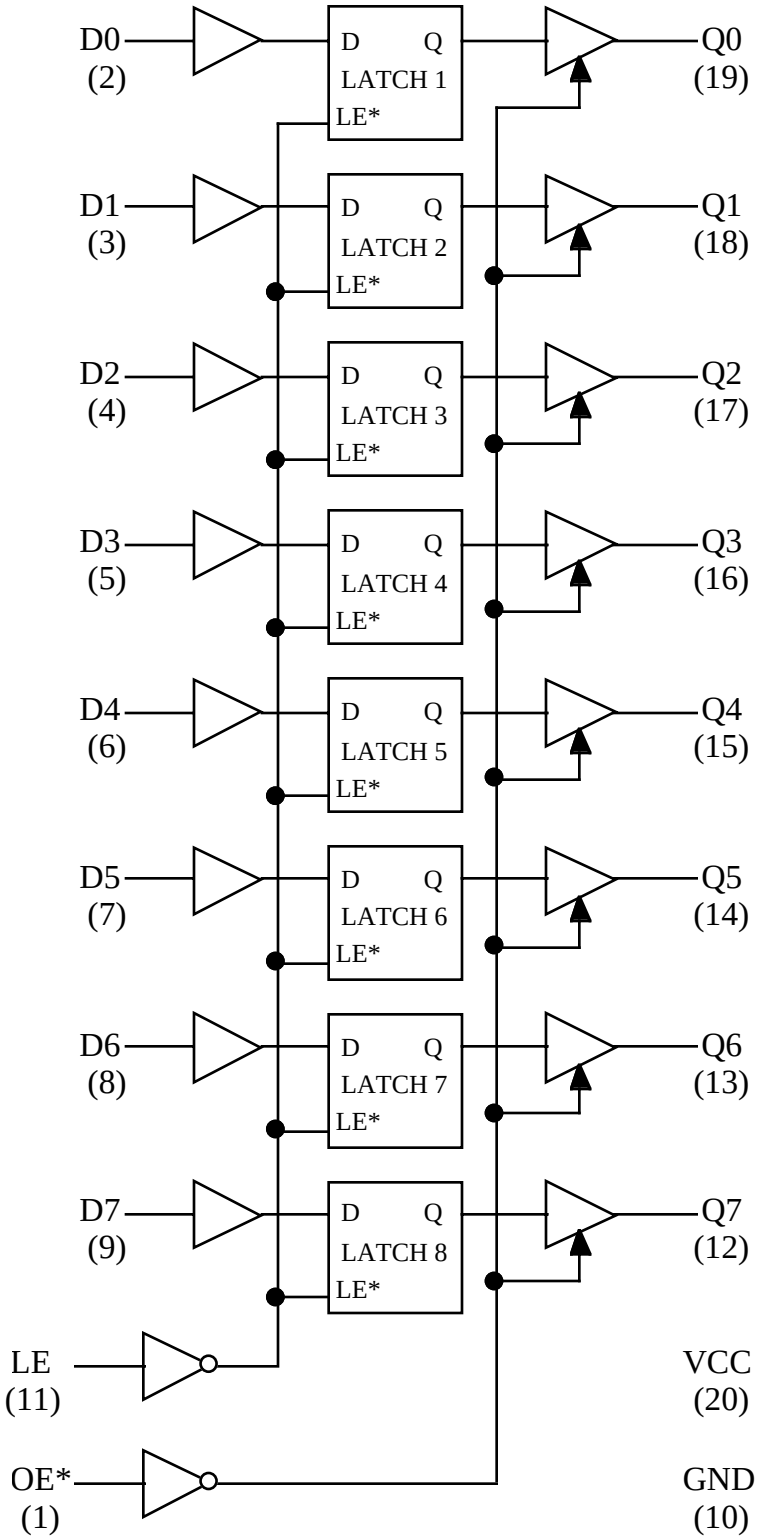


Figure 10. 74573 Octal D-type transparent latch

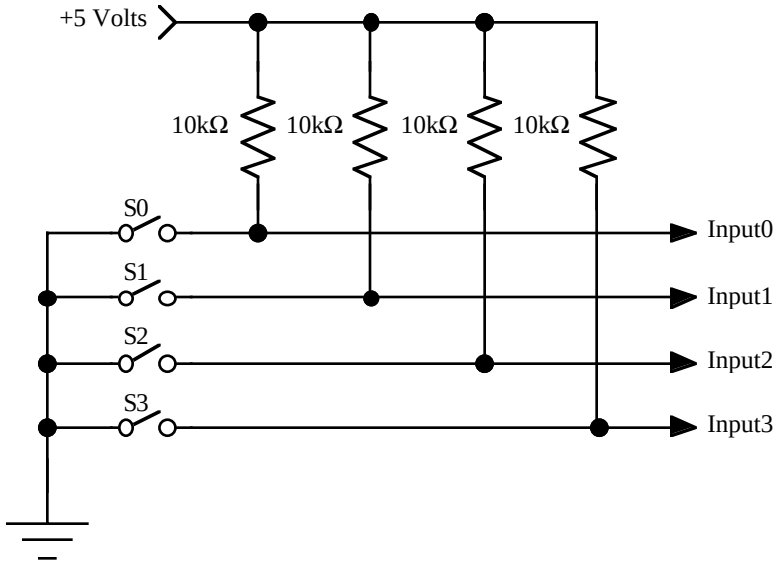


Figure 11. Data input

Figure 11 shows the data input component of the programmable digital lock. The data input component consists of four switches and resistors configured in what is known as a "pull up" arrangement as shown in Figure 12.

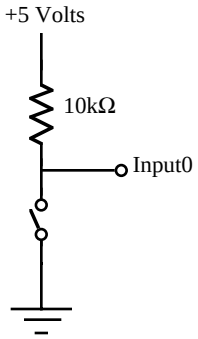


Figure 12. "Pull up" resistor arrangement

When the switch in Figure 12 is open the 5 Volt source is connected to the output (labeled Input0) through the 10k resistor and the output voltage is +5 volts (logical 1). When the switch is closed the bottom end of the 10k resistor is connected to ground through the switch producing an output voltage of 0 volts (logical 0). The function of the 10k resistor is to limit the current drawn from the +5 volt power supply when the switch is closed. As a result the actual value of resistance is not critical and real circuits typically use values between 1kΩ and 10kΩ. Arrays of switches such as shown in Figure 11 are usually implemented as "DIP" switches similar to those found on many peripheral boards in your computer.

The function of this module is to allow the user to input four digital bits of information using four switches.

The data memory section of the programmable digital lock is shown in Figure 13. It contains another four switches which function identically to those shown in Figure 11, i.e., when they are open the voltage applied to the D inputs of the 74573 is a logical "1" and when they are closed the input voltage is a logical "0". The 74573 is a special memory made from D-type flip flops and is known as a "transparent" D-type latch. The actual integrated circuit you will use in the lab will be either a 74373 or a 74573 — the only difference between these chips is in how the pins are configured.

This integrated circuit is complex to understand because it uses two electrical signals to control the input and output operation of the 74573.

The output of many integrated circuits including the 74573 (we will subsequently always refer to the 74573 although you may be actually using a 74373 in the lab) may be electrically switched on and off. This means that the output can be either a logical "1" (5 volts), a logical "0" (0 volts), or off (electrically open). As a result this is often called tri-state logic and is a term you will often encounter in digital design — especially on computer buses.

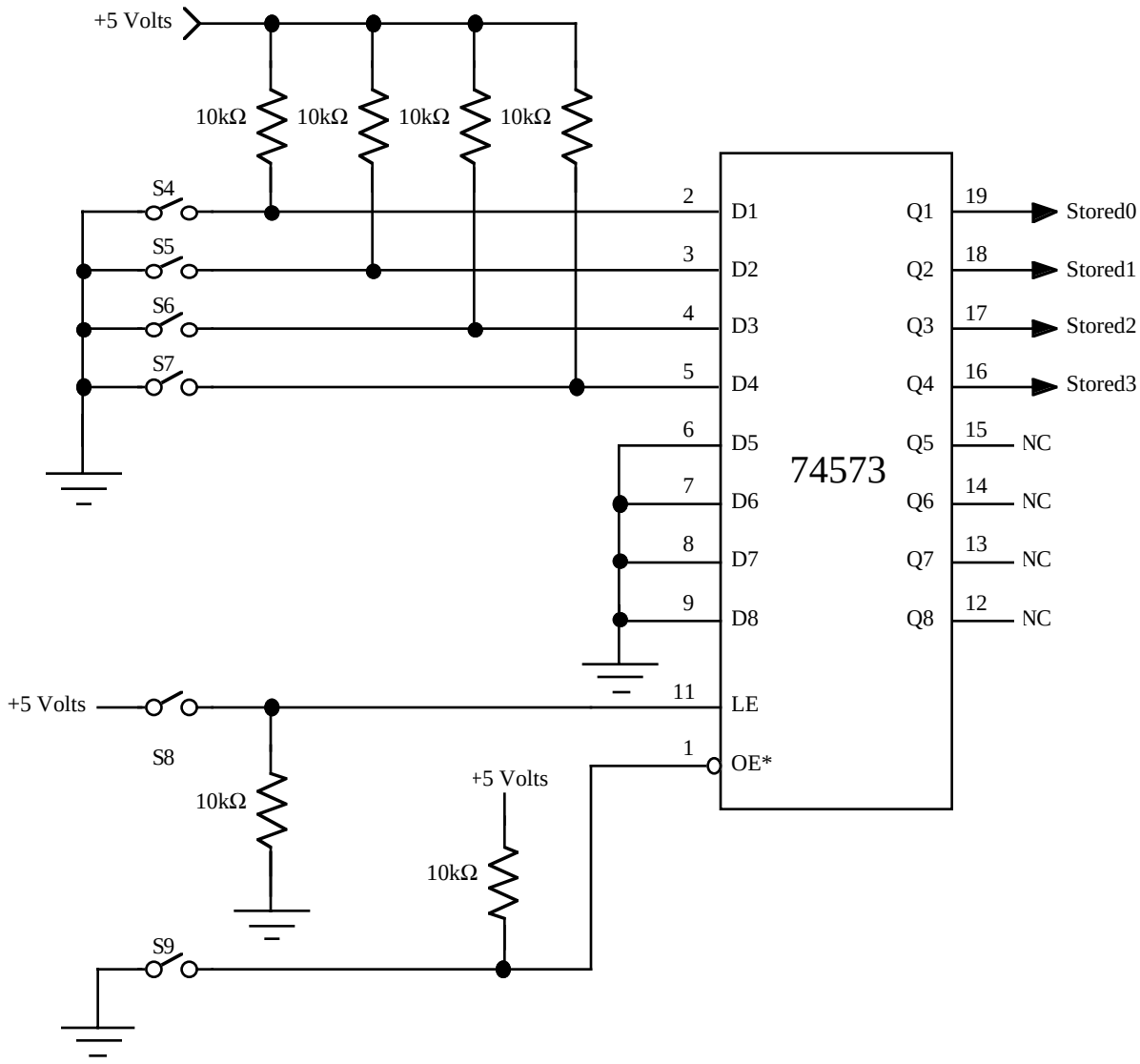


Figure 13. Data memory

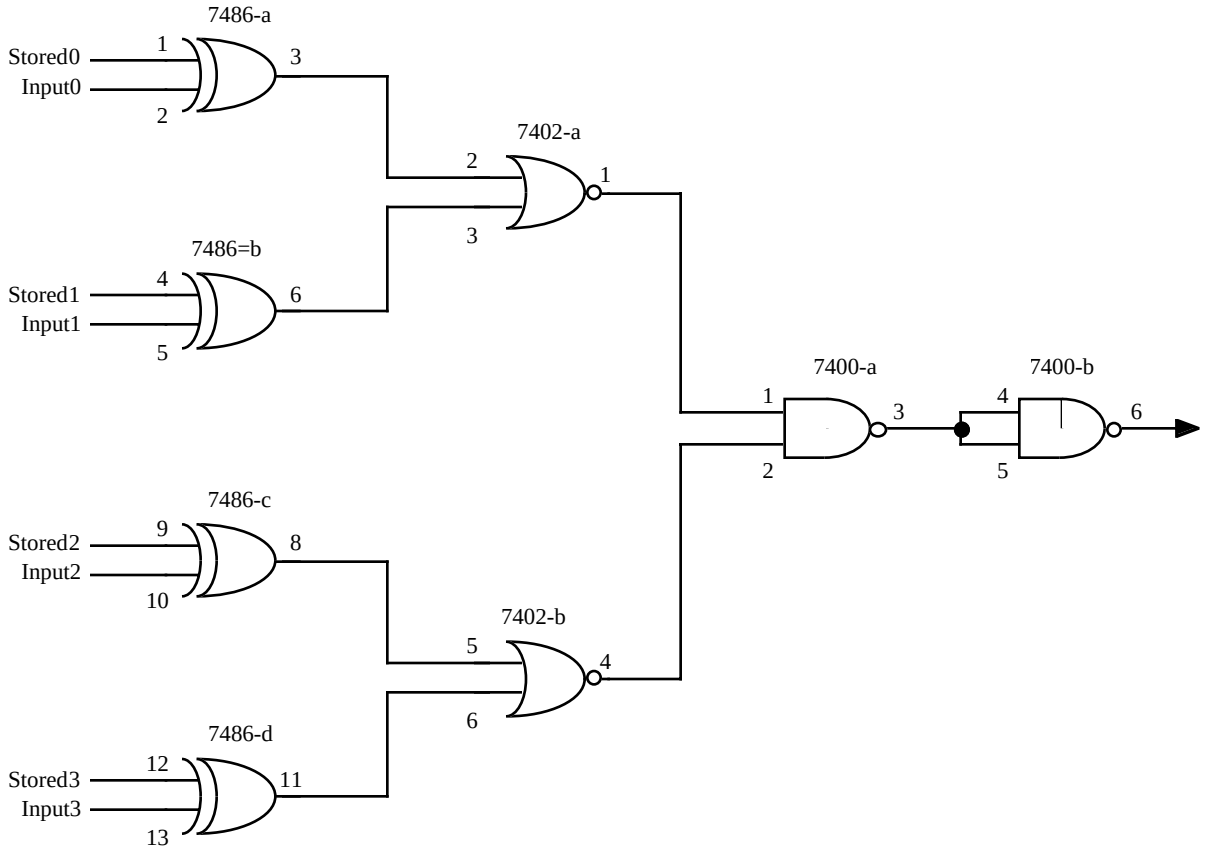


Figure 14. Combinational logic to compare two numbers.

The easiest part of the programmable digital lock to understand is the comparator. Its function is to compare a four digit binary number input by the user with the four bit binary number stored in the D-latch. When the two numbers are identical the lock will operate - in the Electronics Workbench simulation this will operate an LED and a buzzer at the output of the combinational logic shown in Figure 14. The operation of the combinational logic shown in Figure 14 is actually quite simple. The four XOR gates perform the actual comparison; they output a logical zero when then two input numbers are identical. The 7402's and 7400-a function as a AND gate. When the outputs of all XOR gates are zero indicating that the input numbers are identical, the 7402's and the 7400 will produce a high output which can be used to operate an indicator or other electrical device.

B. LAB INSTRUCTIONS

Part 1: Chip familiarization

1. Insert a 7400 quad NAND gate chip into your protoboard. Connect the 5 V power supply and ground to the appropriate inputs. Typically a 14-pin DIP will use pin 7 for the ground and pin 14 for +5 volts; however, you should verify this for every chip you will be using. Using Figure 4 as a reference select a pair of inputs (a single NAND gate) and generate the truth table for the gate. Generate A=logical zero and B=logical zero by using short wires to ground the A and B inputs. Measure the output of the NAND gate (called Y in this lab) — the DMM is probably best for doing this. Record your voltage levels in Data Table 1. Remove the ground wire to the A input of the 7400 and instead connect it to the +5 volts supply using a 10kΩ resistor. This corresponds to A=logical one and B=logical zero. Measure the Y output, and record all data in Data Table 1. Repeat for the remaining cases shown in Data Table 1.
2. Repeat Step 1 for the 7402 NOR gate. Record your data in Data Table 2.
3. Now insert a 74573 (or 74373) into your protoboard. Referring to your data sheet for the +5 and ground connections make the power and ground connections. You will build and test a subset of Figure 13 in this part of the lab.
4. Connect the D1 input to +5 through a 10kΩ resistor. Connect D2 to ground with another 10kΩ resistor. Connect pin 1 to ground with a wire; connect pin 11 to +5 volts through a 10kΩ resistor. Measure the input voltages D1 and D2 and the output voltages Q1 and Q2. Record your measurements in Data Table 3.
5. Repeat the process in step 4. for the remaining three combinations of D1 and D2. The different voltage signals are generated by connecting the appropriate 10kΩ resistor to either +5 volts or ground. Record your results in Data Table 3.
6. Connect the D1 input to +5 through a 10kΩ resistor. Connect D2 to ground with another 10kΩ resistor. Now connect pin 1 (OE*) to ground. Connect pin 11 (LE) to ground. Measure the input voltages D1 and D2 and the output voltages Q1 and Q2. Record your measurements in Data Table 4. Keeping the power on to the chip and without changing either pin 1 or 11 generate the remaining D1 and D2 combinations. Record your measurements for D1, D2, Q1 and Q2 in Data Table 4.
7. Now, connect pin 1 (OE*) to +5 volts through a 10kΩ resistor and pin 11 (LE) to +5 volts through a 10kΩ resistor. Connect the D1 input to +5 through a 10kΩ resistor. Connect D2 to ground with another 10kΩ resistor. . Measure the input voltages D1 and D2 and the output voltages Q1 and Q2. Record your measurements in Data Table 5. Keeping the power on to the chip and without changing either pin 1 or 11 generate the remaining D1 and D2 combinations. Record your measurements for D1, D2, Q1 and Q2 in Data Table 5.
8. Build the part of Figure 14 which consists of 7486-a, 7486-b, and 7402-a. Remember to connect pin 7 of each chip to ground and pin 14 to +5 volts. Use 10kΩ resistors to alternately connect Stored0, Input0, Stored1, and Input 1 to +5

volts or ground. Measure these inputs and the output at pin 1 of the 7402. Record your measurements in Data Table 6.

NOTE: The follow steps will use the Electronics Workbench simulation.

9. The best way to understand the operation of the overall circuit is to use the Electronics Workbench simulation. To get an electrical output you must turn on the output of the 74373 in the Electronics Workbench simulation (Electronics Workbench does not have a 74573 in Version 4.1 so the simulation uses the 74373). This is done by using the switch labeled O (capital O) to ground the OC (remember this is really the OE*) input of the 74573. Operate it by pressing capital-O on your computer keyboard. This produces a logical "1" [The round dot called a bubble indicates that the logical levels are reversed -— low true logic.] At this point the output of the 74573 is turned on. The Electronics Workbench simulation uses a feature which you will did not use in this lab — it uses simulated Light Emitting Diodes (LEDs) to monitor the electrical outputs of the 74573. These appear as four round circles which simulate real LED indicators connected to the outputs of the 74573. When the voltage at their input is +5 they will turn black or other colors on the circuit diagram (they will show only as black on a Macintosh, but as red, green and blue on PCs). If they show as round circles with a white center then the input voltage is zero and they are off. When the OE* output is HIGH the output of the D-type latches is turned off and all the LEDs should be off .

8. To investigate the transparent mode of the 74373 integrated circuit in Electronics Workbench operate the switch labeled I (capital I) by typing a capital-I on your computer keyboard. In this mode whatever appears at the input of the 74573 will appear at its output. Try typing any combination of capital-A, capital-B, capital-C, and capital-D. This will operate the four switches labeled A, B, C and D in the Electronics Workbench simulation. These operate as the switch shown in Figure 12 to generate logical "0" or "1" at the inputs of the 74573. As you type capital-A, capital-B, capital-C, and capital-D on your keyboard the LEDs connected to the output of the 74573 should change state to represent the state of the switches — this is what is meant by transparent, the 74573 acts like a wire in this mode. NOTE: If your LEDs do not change this typically means that the I or O switches are in the wrong position. The transparent mode is not a very interesting mode.

9. Using the Electronics Workbench simulation select a 4 digit code and program it into the memory section of your lock using switches A, B, C and D. To program the lock, first set the code on the four input switches labeled A, B, C and D, then latch it into the D-latch memory by typing capital-I to open the switch labeled I. This will produce a logical "0" at the LE (C) input of the 74573.

10. Test your lock by entering various 4 digit codes through switches W, X, Y and Z. Use switch O to turn on the output of the memory by grounding the OE* (OC) pin. When the two four bit binary numbers are identical the comparator circuit should output a logical 1 which will activate the buzzer and output LED.

DATA AND REPORT SHEETS FOR LAB 10

Student Name (Print): _____ Student ID: _____

Student Signature: _____ Date: _____

Student Name (Print): _____ Student ID: _____

Student Signature: _____ Date: _____

Lab Group: _____

Data Table 1. Truth Table for 7400 NAND gate.

A	B	Y
0	0	
0	1	
1	0	
1	1	

Data Table 2. Truth Table for 7402 NOR gate.

A	B	Y
0	0	
0	1	
1	0	
1	1	

Data Table 3. Truth Table for 74573 latch; OE*=0 volts; LE=5 volts

D1	D2	Q1	Q2

Data Table 4. Truth Table for 74573 latch; OE*=0 volts; LE=0 volts

D1	D2	Q1	Q2

Data Table 5. Truth Table for 74573 latch; OE*=5 volts; LE=0 volts

D1	D2	Q1	Q2

Data Table 6. Truth Table for 74573 latch; OE*=5 volts; LE=0 volts

Stored0	Input0	Stored1	Input1	7402-Pin1

C. QUESTIONS

1. What function does the NAND gate labeled 7400-b at the right of the logic circuit in Figure 14 perform?
2. What function do the "pull-up" and "pull-down" 10kΩ resistors used throughout this lab perform?
3. Explain what would happen if you wired the Q1 outputs of two different 74573 integrated circuits together and both OE* pins were LOW? What if one of the OE* inputs became HIGH? What if both OE* inputs were HIGH?
4. Describe how the programmable lock comparison logic works. You should go into some detail here about how the comparison between the stored code and the user input is done. (This involves a discussion of how the logic circuit works.) Some tables describing the logic of the circuit may be useful. What is the output of the logic circuit? What happens when the user input matches the stored code?
5. Explain how the 74573 can be operated as a memory device.