ENGR 210 Lab 6
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 which 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. 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)

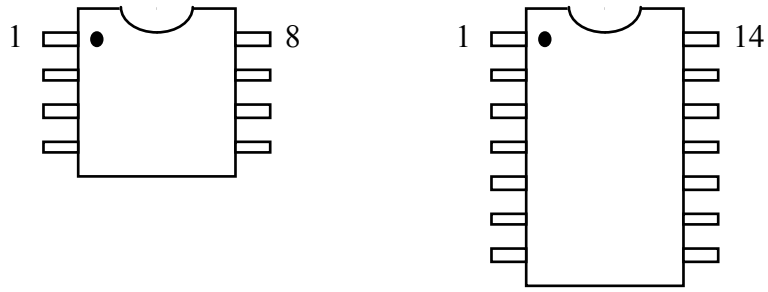
In digital logic there are several fundamental logic operations, namely, the INVERT (or NOT), AND, and OR, as well as the negated versions of the latter two, namely NAND and NOR. Integrated circuit devices which enable the implementation of these functions are available in several *families* of devices. 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 Schottky and HC stands for High speed CMOS. For various reasons, the original TTL family is no longer made — it has been replaced by a number of CMOS varieties such as HC which can electrically replace the original TTL integrated circuits.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

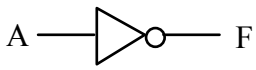
Figure 1. Physical configuration of 8 and 14 pin DIP IC devices.

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) the NOT gate, or inverter



(a) logic symbol

A	F
0	1
1	0

(b) Corresponding TRUTH table

Figure 2. Inverter

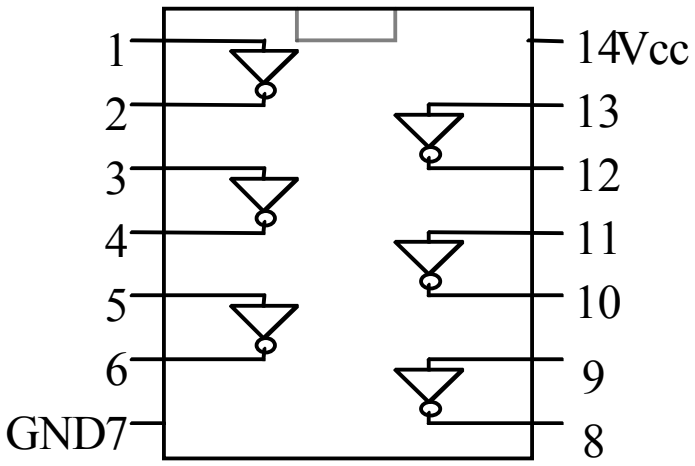
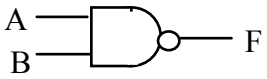


Figure 3. A 7404 integrated circuit (IC) contains 6 inverters.

6. 2-input NAND gate.



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

(a) logic symbol (b) Corresponding TRUTH table

Figure 4. NAND - logically equivalent to an AND gate followed by a NOT.

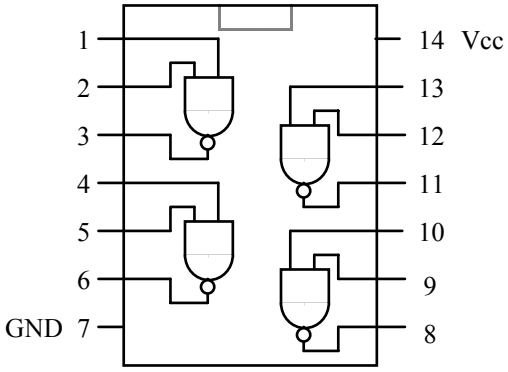
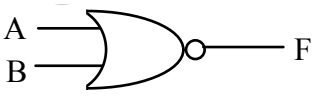


Figure 5. A 7400 IC contains 4 2-input NAND gates.

(c) 2-input NOR gate.



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

(a) logic symbol (b) Corresponding TRUTH table

Figure 6. NOR - logically equivalent to an OR gate followed by a NOT.

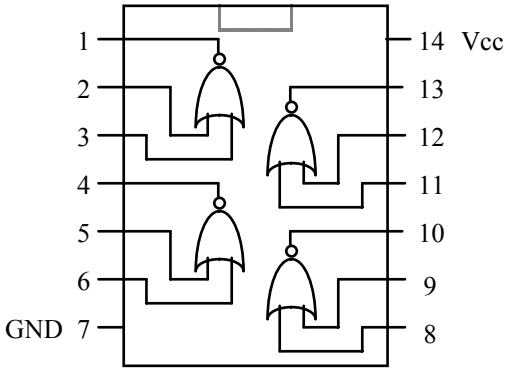
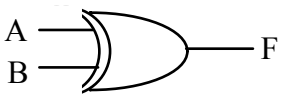


Figure 7. A 7402 IC contains 4 2-input NOR gates.

(d) 2-input XOR (EXCLUSIVE OR) gate



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

(a) logic symbol (b) Corresponding TRUTH table

Figure 8. XOR - Exclusive OR

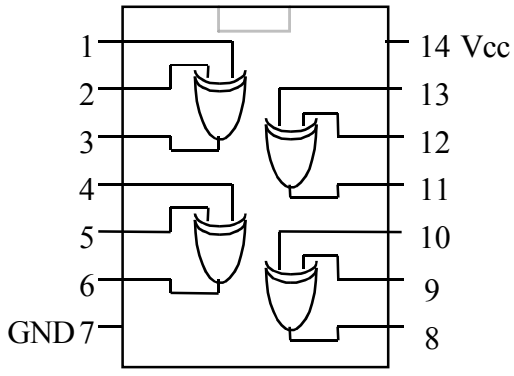


Figure 9. A 7486 IC contains 4 2-input XOR gates.

(e) 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.

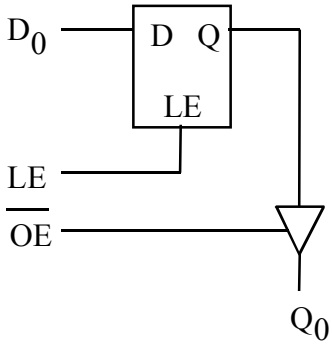


Figure 10. Logic diagram of D-latch.

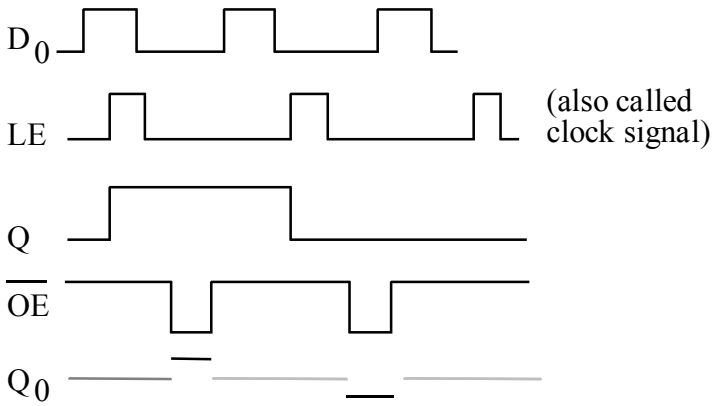


Figure 11. Timing diagram of D-latch. (Note that Q_0 is undefined when OE* is high.)

A logic diagram of a D-latch is shown in Figure 10. 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 11, 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 10, 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 cannot 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 74373) you will use in this lab is shown in Figure 12. The 74373 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.

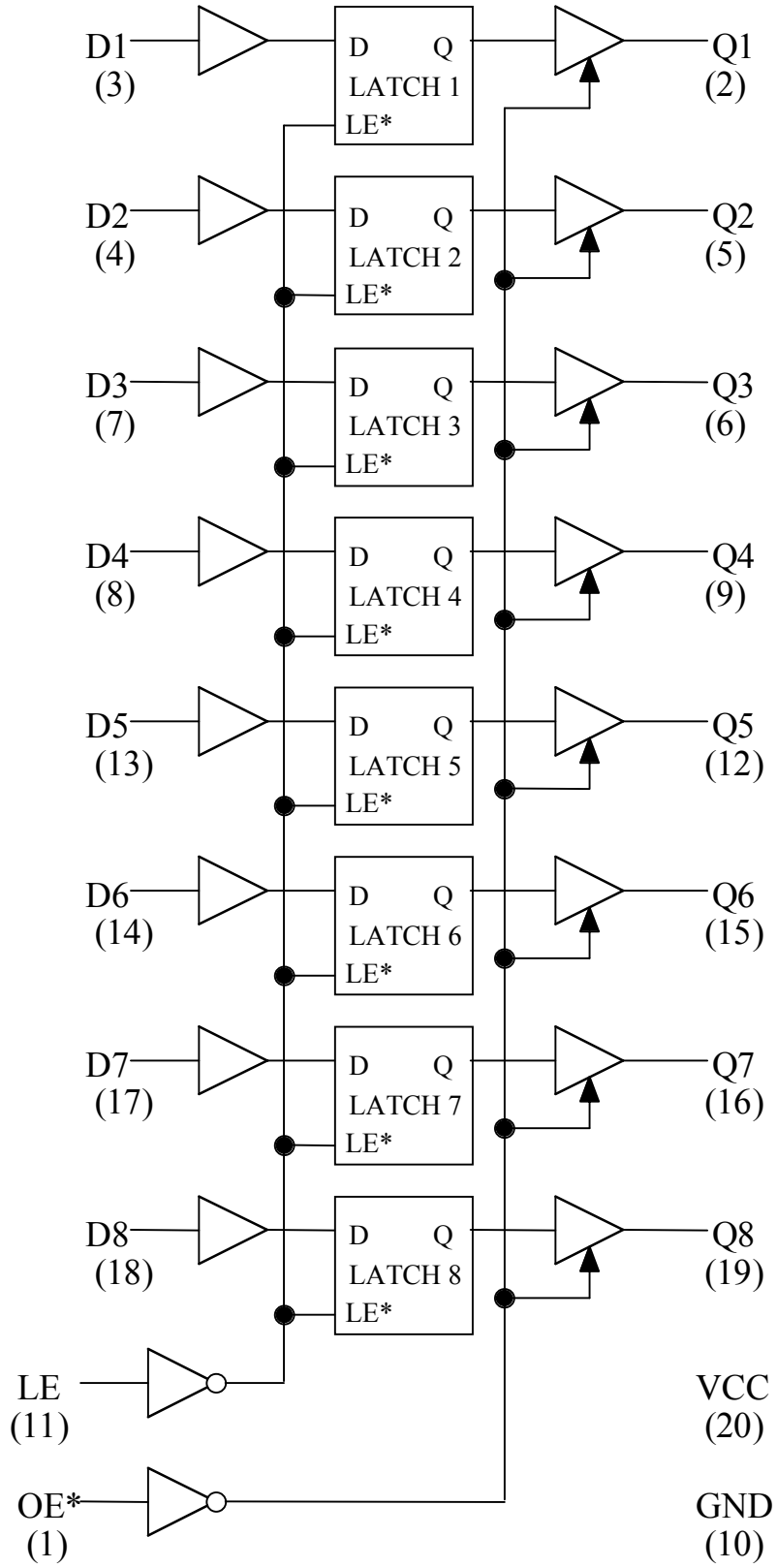


Figure 12. 74373 Octal D-type transparent latch

3. A Programmable Digital Lock

In this lab you will build and test components of a programmable digital lock. 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.

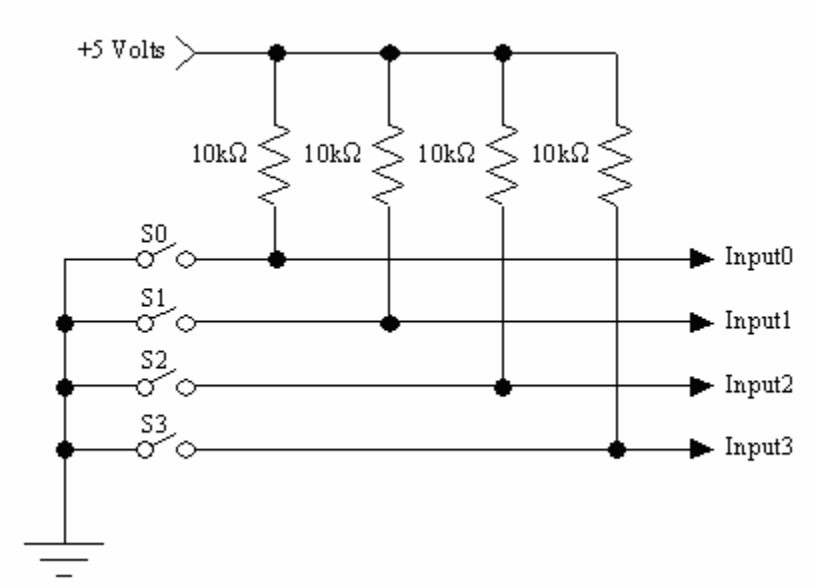


Figure 13. Data input

Figure 13 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 14.

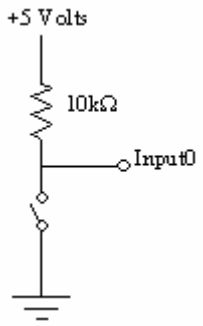


Figure 14. "Pull up" resistor arrangement

When the switch in Figure 14 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 those shown in Figure 13 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 15. It contains another four switches which function identically to those shown in Figure 13, i.e., when they are open the voltage applied to the D inputs of the 74373 is a logical "1" and when they are closed the input voltage is a logical "0". The 74373 is a special memory made from D-type flip flops and is known as a "transparent" D-type latch.

This integrated circuit is complex to understand because it uses two electrical signals to control the input and output operation of the 74373.

The output of many integrated circuits including the 74373 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.

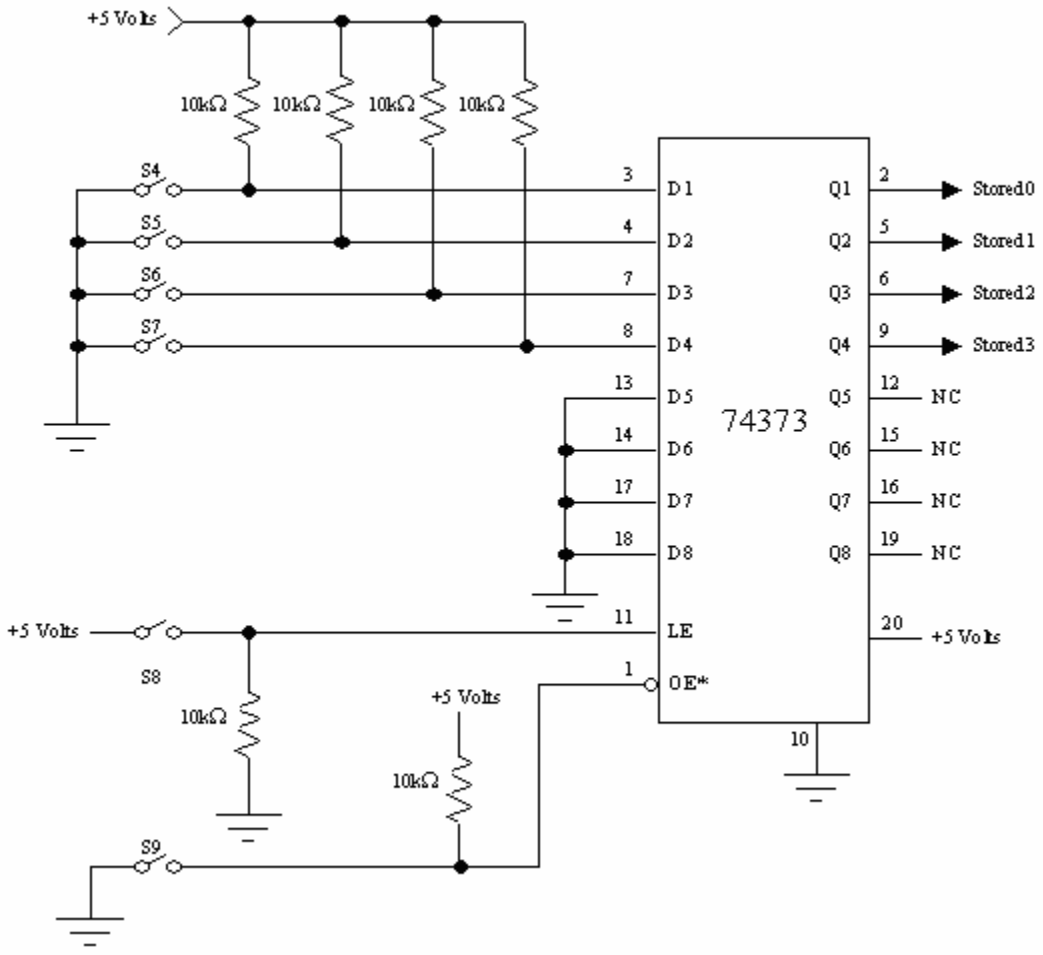


Figure 15. Data memory

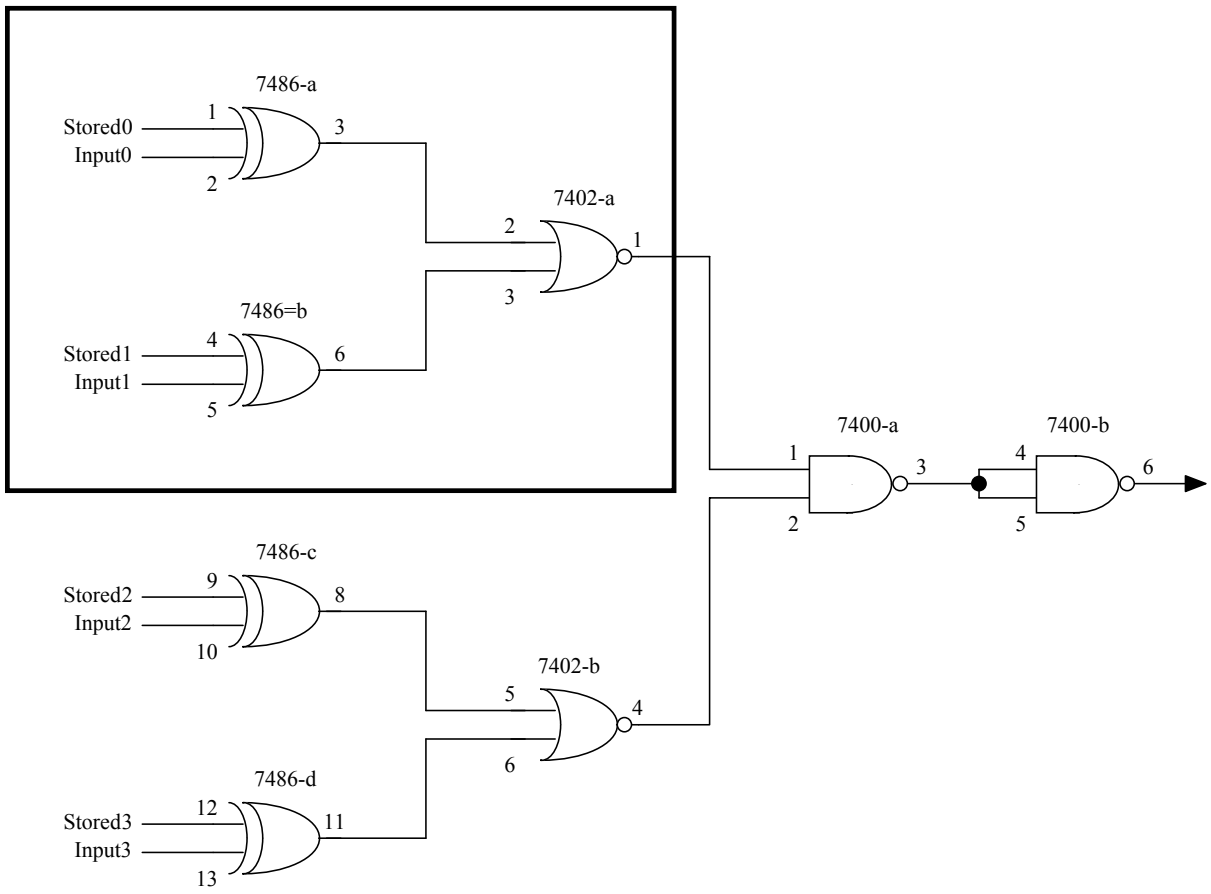


Figure 16. Combinational logic to compare two numbers.

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. The operation of the combinational logic shown in Figure 16 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 7400s 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 7400s will produce a high output which can be used to operate an indicator or other electrical device.

B. LAB INSTRUCTIONS

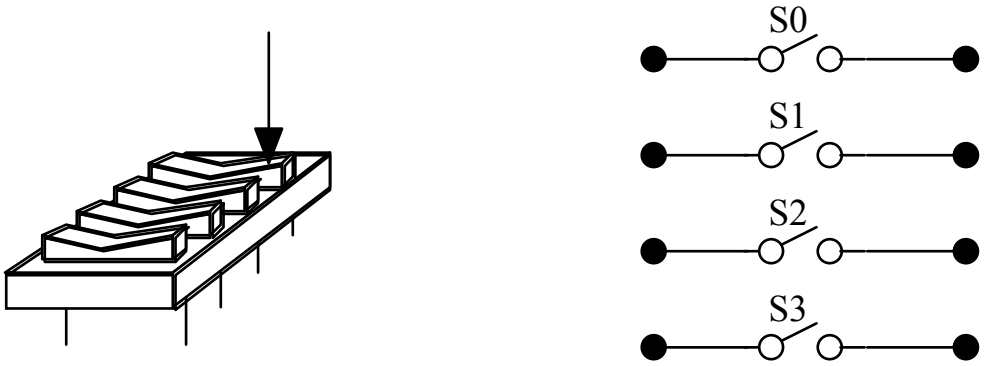
Part 1: Chip familiarization

1. Get the parts indicated in Table 2 from the Instrument Table at the back of the lab (where you got your potentiometers and OP AMPs last week).

Qty	Description
1	7400 NAND gate integrated circuit
1	7402 NOR gate integrated circuit
1	7486 XOR gate integrated circuit
1	74373 octal transparent D latch integrated circuit
1	DIP switches
6	10kΩ resistors

Table 2. Parts needed for Lab

2. Build a data input circuit according to Figure 13. This circuit uses a DIP (Digital In-line Package) switch that looks like an integrated circuit. Figure 17(a) shows a DIP switch which contains four mechanical switches as shown in Figure 17(b). These on-off switches can be conveniently switched using a pen or pencil. This circuit will be used for all of the digital logic chips that you will test. HINT: Verify that your input circuit operates as expected using a digital multimeter.



(a) Physical DIP switch (b) Electrical equivalent of DIP switch

Figure 17. DIP (Digital Inline Package) Switch

3. 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 5 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 the data input circuit that you constructed in Step 2. 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. This corresponds to A=logical zero 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.

4. Repeat Step 3 for the 7402 NOR gate. See Figure 7 for the 7402 pinouts. Record your data in Data Table 2.

5. Now insert a 74373 into your protoboard. Referring to Figure 12 for the +5 and ground connections, make the power and ground connections. You will build and test a subset of Figure 15 in this part of the lab.

6. Connect your data input circuit (Figure 13) to the 74373's inputs as shown in Figure 15. Connect pin 1 to ground with a wire; connect pin 11 to +5 volts through a 10kΩ resistor. Unlike Figure 15 there are no connections to D3 and D4 from your data input circuit. Similarily Q3 and Q4 are not used in this lab.

   OPTIONAL: If you wish you can connect Input 2 and Input 3 from your data input circuit to pins 11 and 1 respectively; however, make sure that you generate the appropriate inputs to these pins as called for in the following steps.

   Measure the input voltages D1 and D2 and the output voltages Q1 and Q2. Record your measurements in Data Table 3.

7. Repeat the process in step 6. for the remaining three combinations of D1 and D2. The different voltage signals are generated by switching the appropriate DIP switch to output either +5 volts or ground. Record your results in Data Table 3.

8. Pin 1 (OE*) should remain connected to ground; however, now connect pin 11 (LE) to ground. Keeping the power on to the chip and without changing either pin 1 or 11 generate all possible D1 and D2 combinations. Measure the input voltages D1 and D2 and the output voltages Q1 and Q2 for each combination. Record your measurements for D1, D2, Q1 and Q2 in Data Table 4.

9. Now, connect pin 1 (OE*) to +5 volts through a 10kΩ resistor and pin 11 (LE) to +5 volts through a 10kΩ resistor. Keeping the power on to the chip and without changing either pin 1 or 11 generate all possible D1 and D2 combinations using your data input circuit. Measure the input voltages D1 and D2 and the output voltages Q1 and Q2 for each combination. Record your measurements for D1, D2, Q1 and Q2 in Data Table 5.

10. Build the part of Figure 16 shown in the box 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 the data input circuit to alternately connect Stored0, Input0, Stored1, and Input 1 to +5 volts or ground, rather than using the outputs from the 74373. Measure the output at pin 1 of the 7402. Record your measurements in Data Table 6.

DATA AND REPORT SHEETS FOR LAB 10

Student Name (Print): _____ Student ID: _____

Student Signature: _____ Date: _____

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 74373 latch; OE*=0 volts; LE=5 volts

D1	D2	Q1	Q2
0	0		
0	1		
1	0		
1	1		

Data Table 4. Truth Table for 74373 latch; OE*=0 volts; LE=0 volts

D1	D2	Q1	Q2
0	0		
0	1		
1	0		
1	1		

Data Table 5. Truth Table for 74373 latch; OE*=5 volts; LE=5 volts

D1	D2	Q1	Q2
0	0		
0	1		
1	0		
1	1		

Data Table 6. Truth Table for Boxed Logic of Figure 16.

Stored0	Input0	Stored1	Input1	7402-Pin1
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

QUESTIONS

1. What function does the NAND gate labeled 7400-b at the right of the logic circuit in Figure 16 perform?
2. What function do the "pull-up" 10kΩ resistors used throughout this lab perform?
3. Explain what would happen if you wired the Q1 outputs of two different 74373 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? HINT: This is called bus arbitration.
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 74373 can be operated as a memory device.