Digital Electronics I: Logic, Flip-Flops, and Clocks

Purpose

This experiment introduces some of the fundamental circuit elements of digital electronics. These include three kinds of logic gate, two kinds of flip-flop (single bit memory), and the 555 timer chip used as a digital clock

Introduction

In almost all experiments, the signals that represent physical quantities start out as analog waveforms. To display and analyze the information contained in these signals, they must be converted to digital data. Often this is done inside a commercial instrument such as an oscilloscope or a lock-in amplifier, which is connected to a computer through a digital interface. In other cases data acquisition cards are added to a computer chassis and the analog signals can be connected directly to the computer. Scientists usually buy their data acquisition equipment rather than build it, so they often don"t have to know too much about the digital circuitry that makes it work. Almost all data is analyzed with a computer, but like other users scientists don"t often have to know much about the digital circuitry inside their computers. We emphasize analog electronics in the course because scientists usually have to know much more about it to design and build their experiments.

On the other hand, there are plenty of reasons to know something about digital methods. The author of our text (physicist Paul Horowitz) has built custom digital signal processors to search for signs of extraterrestrial intelligence in radio telescope signals, and particle physicists have built customized computer hardware to make calculations in quantum chromodynamics (the theory of the strong force). If you try to repair a commercial instrument like a modern lock-in you will find that it is full of digital electronics. The trend in modern instrument design is to do as much digitally as possible; even the front-panel knobs are not really analog controls, rather they are coded switches or optical encoders that generate digital data directly. An increasing number of analog parts can be controlled digitally, for example you can buy digital potentiometers that behave exactly like an analog pot, but look more like an op-amp chip, and instead of controlling the position of the wiper with a knob, you send a digital code into some of the extra pins.

In digital circuits the voltage on a wire takes one of only two values called logic HIGH and logic LOW, corresponding to a binary 1 or 0. Information is conveyed by the pattern of HIGH and LOW voltages. A single wire can convey just one bit of information at any one time. When the information to be conveyed

requires more than a single bit, either more wires can be used to convey data (parallel digital data), or a sequence of bits can be sent over time as HIGH"s and LOW"s moving along a single wire (serial digital data).

Analog information can be translated into digital form by a device called an Analog-to-Digital Converter (ADC). A set of N bits has 2N possible different values. If you try to represent an analog voltage by 7 bits, your uncertainty will be about 1%, since there are 27 = 128 possible combinations of 7 bits. For higher accuracy you will need more bits. There is also a device called a Digital-to-Analog Converter (DAC) that can convert digital data back into an analog waveform. You can choose either serial or parallel ADCs and DACs, depending on whether you are using serial or parallel digital data.

In this experiment, we will learn about the most basic elements of digital electronics, from which more complex circuits, including computers, can be constructed. Logic gates perform logical operations like AND and OR. The gates we will use are made of bipolar transistors and they come from a family of devices called TTL (transistor-transistor logic). There are many other logic families (some made of MOSFETs) offering various trade-offs between speed, power consumption, supply voltage, and output drive capability (see H&H 9.01, and for the most recent families see the Logic Selection Guide at Texas Instruments, www.ti.com). Logic gates alone can be used to construct arbitrary combinatorial logic (they can generate any truth-table), but to create a machine that steps through a sequence of states like a computer does, we need also memory and a clock.

The fundamental single-bit memory element of digital electronics is called a flip-flop. We will study two types, called SR (or RS) and JK. The flip-flops we have chosen are also from the TTL family. A digital clock is a repeating digital waveform used to step a digital circuit through a sequence of states. We will introduce the 555 timer chip and use it to generate a clock signal. Digital circuits able to step through a sequence of states with the aid of flip-flops and a clock are called sequential logic.

Readings

1. FC Chapter 11 (digital electronics)

2. H&H Chapter 8. Everything is this chapter is good to know about, but sections 8.01, 8.02, 8.04, 8.07-8.10, 8.12, 8.16 are most relevant. Also have a look at section 5.14 on the 555 timer chip.

3. (Optional) Diefenderfer 11.1-11.5, 12.1-12.5; Brophy Ch. 9 and pages 272-290; The TTL Cookbook, by Don Lancaster, SAMS (1974).

Theory

LOGIC STATES

The voltage in a digital circuit is allowed to be in only one of two states: HIGH or LOW. We usually abbreviate these as HI and LO.

HI is taken to mean logical (1) or logical TRUE. LO is taken to mean logical (0) or logical FALSE.

In the TTL logic family (see Figure 9.1), any voltage in the range 2.8 to 5.0 V is HI, and any voltage in the range 0 to 0.8 V is LO.

Any voltage outside this range is undefined, and therefore illegal, except briefly during transitions. If a TTL circuit is given a voltage in this undefined range, it might, unpredictably, interpret it as either a "1" or a "0."   We will sometimes refer to HI as the "5 volt" level, and LO as the "0 volt" level.
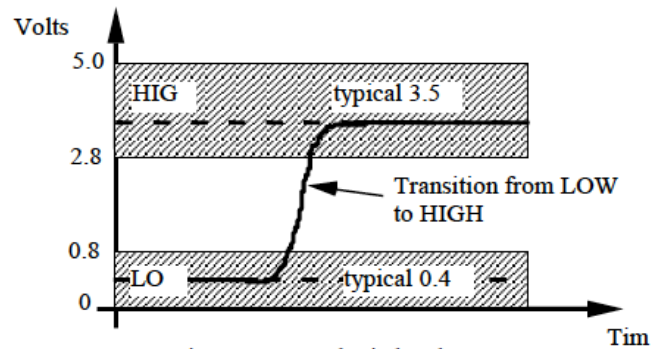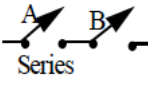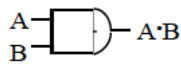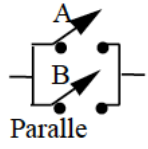

Figure 9.1 TTL logic levels

LOGIC GATES

The flow of digital signals is controlled by transistors in various configurations depending on the logic family (see H&H 8.09 for details). For most purposes we can imagine that the logic gates are composed of ideal switches with just two states: OPEN and CLOSED. The state of a switch is controlled by a digital signal. The switch remains closed so long as a logical (1) signal is applied. A logical (0) control signal keeps it open.

Logic signals interact by means of gates. The three fundamental gates AND, OR, and NOT, are named after the three fundamental operations of logic that they carry out. The AND and OR gates each have two inputs and one output. The output state is determined by the states of the two inputs.

The function of each gate is defined by a truth table, which specifies the output state for each possible combination of input states. The output values of the truth tables can be understood in terms of two switches. If the switches are in series, you get the AND function. Parallel switches perform the OR operation. The most common gates are shown in Fig. 9.2. A bubble after a gate or at an input indicates NOT.

Operation	Switches	Condition that circuit is closed	Boolean Notation	Symbol	Truth Table

AND — Series — (A AND B are closed) — $A \bullet B$ or AB — A·B

A	B	A·B
0	0	0
0	1	0
1	0	0
1	1	1

OR — Paralle — (A OR B is closed) — $A+B$ — A+B

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

NOT (same as invert) — Different kind of switch — 1 means open 0 means closed — NOT $A \equiv \bar{A}$ — $\bar{A}$

A	$\bar{A}$
0	1
1	0

The three compound gates NAND, NOR and XOR can be made from AND, OR, and NOT. NAND means an AND gate followed by a NOT, while NOR means an OR gate followed by a NOT. The EXCLUSIVE-OR (XOR) is similar to OR but it has a LO output if both inputs are HI, so you can think of it as one OR the other but NOT both.

NAND and NOR are more common than AND and OR because with the help of DeMorgan‟s theorems they can be used to simplify complex circuits (see below).

When several gates are combined to perform a complex logical operation, a good design uses as few as possible. Boolean Algebra, the mathematics of two valued variables, is the theoretical tool used to simplify complex logical expressions.

Compound Gates

NAND — A, B — $\overline{A \cdot B}$

NOR — A, B — $\overline{A+B}$

XOR — A, B — $A \oplus B$ = $\overline{A}B + A\overline{B}$
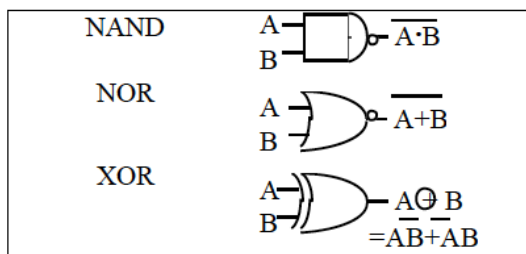
Figure 9.2 Basic Logic Gates

BOOLEAN ALGEBRA

Fundamental laws

We imagine a logical variable, A, that takes on the values 0 or 1. If $A = 0$ then $\bar{A} = 1$ and if $A = 1$ then $\bar{A} = 0$. Here are some obvious identities using the AND, OR and NOT operations. Looking at these identities you can see why the 'plus' symbol was chosen for OR and 'time' was chosen for AND.

	OR		AND		NOT

$$A + 0 = A \qquad\qquad A \bullet 0 = 0 \qquad\qquad A + \bar{A} = 1$$

$$A + 1 = 1 \qquad\qquad A \bullet 1 = A \qquad\qquad A \bullet \bar{A} = 0$$

$$A + A = A \qquad\qquad A \bullet A = A \qquad\qquad \bar{\bar{A}} = A$$

$$A + \bar{A} = 1 \qquad\qquad A \bullet \bar{A} = 0$$

Equality

Two Boolean expressions are equal if and only if their truth tables are identical.

Associative Laws

(A+B)+C = A+(B+C)

(AB)C=A(BC)

Distribution Laws

A(B+C) = AB+AB

(A+AB) = A)

(A+ Ā B) = A+B

(A+B)• (A+C) = (A+BC)

DeMorgan's Theorems

$$\overline{A \bullet B \bullet ...} = \bar{A} + \bar{B} + ...$$

$$\overline{A + B + ...} = \bar{A} \bullet \bar{B} \bullet ...$$

Example of Proof

Each of the above equalities is a theorem that can be proved. Let's do an example by directly comparing the truth tables for the left and right sides. We take on DeMorgan's first theorem for two variables, $\overline{AB} = \bar{A} + \bar{B}$:

A	B	AB	$\overline{AB}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

A	B	$\bar{A}$	$\bar{B}$	$\bar{A} + \bar{B}$
0	0	1	1	1
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0

The last columns of the truth tables are identical. Thus, the first theorem is proven for two variables.

Example of Simplification

Boolean algebra can be used to simplify logical expressions and reduce the number of gates required in a circuit. In Fig. 9.3 we show two ways to implement the expression, $Y = A + \bar{A}\,BC$.
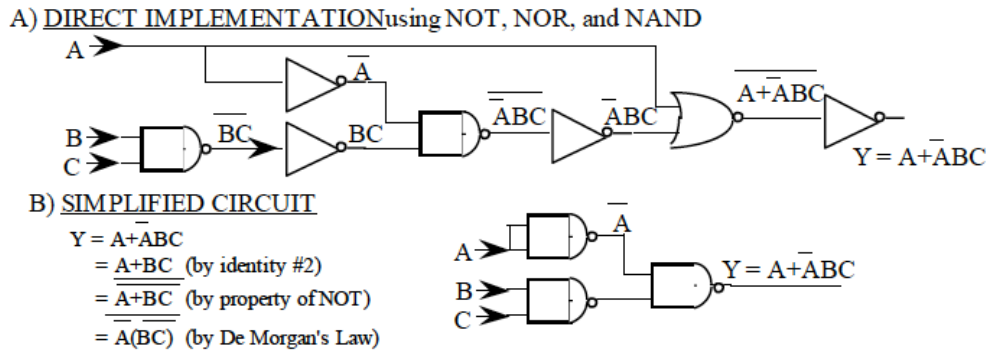

Fig. 9.3. Boolean simplification

Examples with many input variables

Here are two examples that illustrate the use of the double complement i.e., $\bar{\bar{A}} = A$ with DeMorgan's theorems for reducing expressions to a form that can be implemented with 2-input NAND and NOR, thus reducing the types of gates needed. The expressions we want are Y=ABCD and Y=A+B+C+D.

All of the above circuits are examples of combinatorial logic. The output appears almost immediately upon application of the inputs. The logic value of the output depends only upon the present-time combination of a number of parallel inputs and the arrangement of gates.
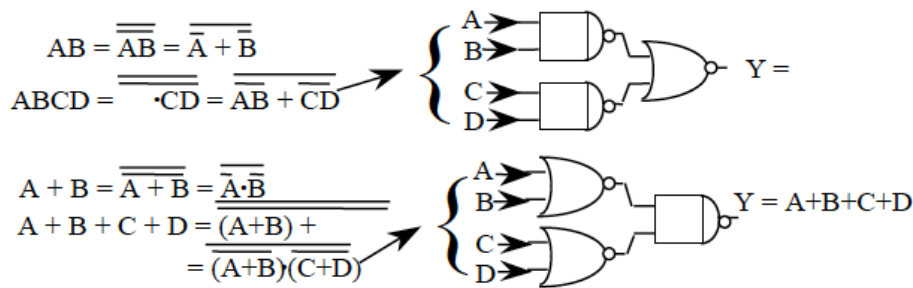

Fig. 9.4. Reduction to NAND and NOR via DeMorgan's Theorem

MEMORY ELEMENTS AND FLIP-FLOPS

In sequential logic circuits the output depends upon previous values of the input signals as well as their present-time values. Such circuits necessarily include memory elements that store the logic values of the

earlier signals. The fundamental circuit is the RS memory element. The JK flip-flop has an RS flip-flop at its core, but it adds circuitry that synchronizes output transitions to a clock signal. Timing control by a clock is essential to most complex sequential circuits.
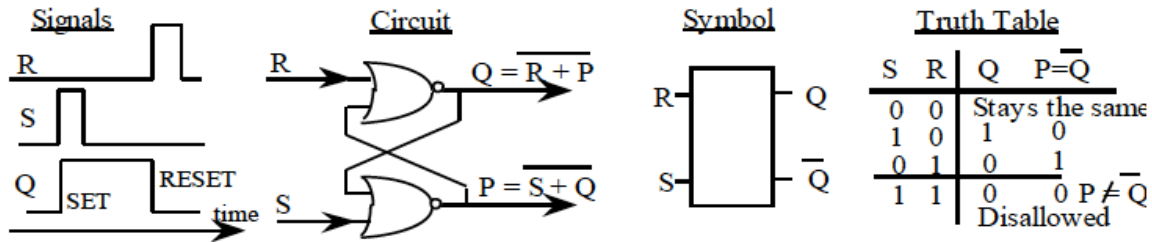
RS(Reset-Memory) Element

Fig. 9.5. RS memory element.

The truth table shows how the circuit remembers. Suppose that it is originally in a state with Q=0 and R=S=0. A positive pulse S at the input sets it into the state Q=1, where it remains after S returns to zero. A later pulse R on the other input resets the circuit to Q=0, where it remains until the next S pulse.

JK Flip-Flop (TTL74107)

There are three kinds of input to the JK flip flop

      1) data inputs J and K

      2) the clock C

      3) the direct input CLR (clear)
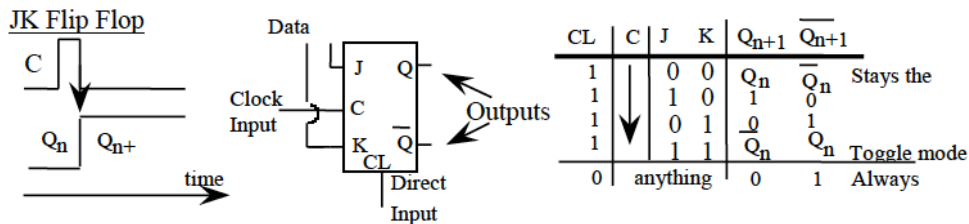
There are two outputs: $Q, \bar{Q}$

Fig. 9.6. JK flip-flop description.

In the absence of a clock pulse, the output remains unchanged at the previously acquired value, Q_n , which is independent of the present-time data inputs J and K. Only on arrival of a clock pulse, C, can the output

change to a new value, Q_{n+1}. The value of Q_n depends on the J and K inputs in the way specified in the truth table. The change occurs at the <u>downward</u> going trailing edge of the clock pulse, as indicated by the downward arrow in the truth table.

The direct input, CLR, overrides the clock and data inputs. During normal operation, CLR = 1. At the moment CLR goes to zero, the output goes to zero and remains there so long as CLR = 0.

555 Timer and Digital Clock

See FC section 11.14 for a description of the guts of the 555 timer chip. Figure 9.7 shows the circuit for generating a clock with the 555 and summarizes the formulas relating the resistor and capacitor values to the output low time T1 and the output high time T2

(b) Component values

Output High (charge time):
$$T2 = (RA+RB)C \ln2$$

Output Low (discharge):
$$T1 = RBC \ln2$$

Period: $T = T1 + T2$

(c) Limiting Values

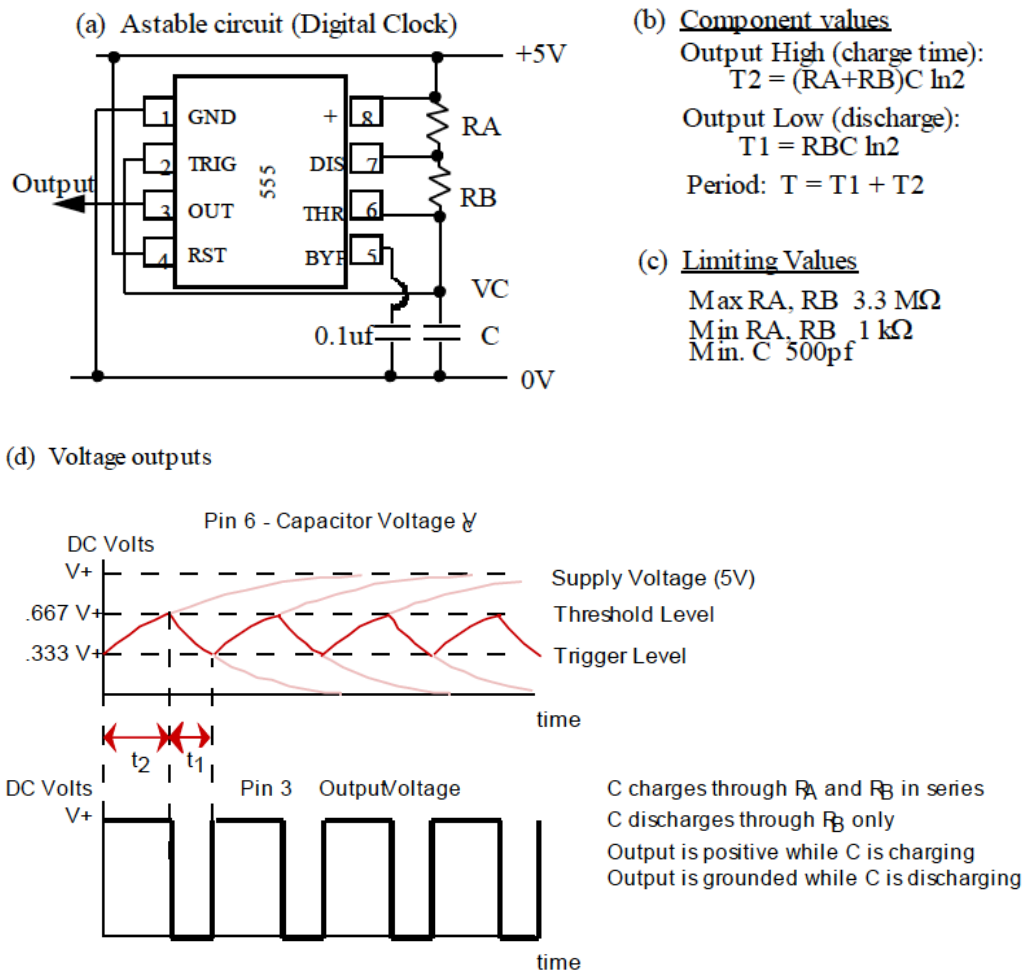Max RA, RB 3.3 MΩ
Min RA, RB 1 kΩ
Min. C 500pf

Figure 9.7   Astable circuit using 555 Timer chip

Pre-Lab Problems

1. Enter in your lab book the circuit diagrams and truth tables of all the circuits you will test. These include the NAND , NOR, and INVERT.

2. Prove DeMorgan''s second theorem by comparing the truth table for both sides of the equation:

$$\overline{A+B}=\overline{A}\bullet\overline{B}$$

Use the laws of Boolean algebra to derive the following:

$$A + A \bullet B = A$$
$$A \bullet (A + B) = A$$
$$A + \overline{A} \bullet B = A + B$$

3. Design a circuit to perform the EXCLUSIVE OR function using only NAND and NOR gates. Simplify the circuit so that you use the smallest possible number of NAND and NOR gates (and only use NAND and NOR gates). Show your Boolean calculation. Check the result using truth tables.

4. Derive the truth table for a RS memory element made from two NOR gates (see Fig. 9.5).

5. Design a 1 kHz clock using the 555 timer chip. Make the low level pulses 1/4 period in length. How large of a capacitor would you need to substitute so that you can you modify your clock so that it can run at 1 Hz (for visual observation of LEDs)?

6. A JK flip-flop with J=K=1 and CLR=1 is driven at the clock input by 1 kHz pulses from a NAND gate driven by a 555 timer (see Fig. 9.11 below). Diagram the waveforms for the clock and the Q output on the same time scale. Make sure to include enough periods of the clock signal to see all the behavior of the JK''s output.

New Apparatus and Methods

USING 7400 SERIES TTL CHIPS
1) Power supply:

Check your power supply before connecting to the circuit board. The Tektronix PS 280/3 has a fixed 5V output that you should use to power digital circuits. The logic chips burn out at around 6 V. If the voltage drops when you connect to the circuit, do not increase V. Either look for a short or increase the current limit instead.

      Normal supply voltage : +5.0V
      Absolute maximum: +5.5V

Current : For Types 7400 7402 7404: 12mA per chip,
                         For Type 7486:   30mA per chip

2) Output

The output from each individual gate can drive up to ten other TTL inputs. This is called the "fan-out number". The output is delayed 10 nsec after the input for the INV, NAND, and NOR gates. The delay is 18 nsec for the EXCLUSIVE OR, and 25 nsec for the JK flip-flop.
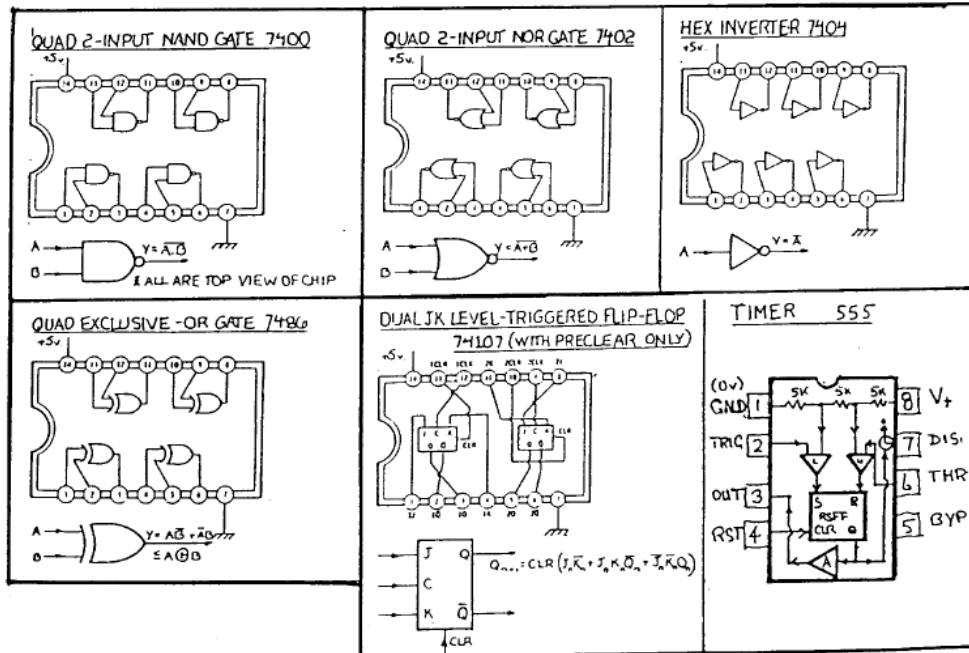

Fig. 9.8 Pin-outs for some TTL chips

3) Pin-outs

 Each chip has a dot or notch to indicate the end where pins 1 and 14 are located. The pin numbers increase sequentially as you go counter-clockwise around the chip in a top view. Pin layouts for logic chips are readily available on the web (Google „ttl pin-out" or find the data sheet at www.ti.com). Be careful to verify that you are looking at the pin-out for the package type that you are using (i.e. DIP, and not SOIC or other). In 14 pin logic chips, pin 7 is always grounded (0 V) and pin 14 is always connected to the +5 V supply.

4) Data Records

Write in your lab book the circuit, the Boolean equation that expresses its function, and the predicted truth table beforehand. Enter the observed logical values of the outputs in an adjacent but separate column.

5) Logical inputs and observation of logical outputs with LEDs

Input logical values can be set by connecting wires from the gate inputs to either 0 V (logical 0) or 5 V

(logical 1). The logic level of the output can be observed using a light emitting diode (LED) which is connected from the output to ground. The LED lights up when the output is +5 V and is off when the output on 0 V. The cathode of the LED is grounded, and must always have a 470 Ω to 680 Ω resistor in series to limit the current and prevent burnout.

A bank of ten LEDs in a DIP package (type MV57164) is available. We suggest that you keep one bank of LEDs on your board throughout the logic experiments. The pin diagram is given below.
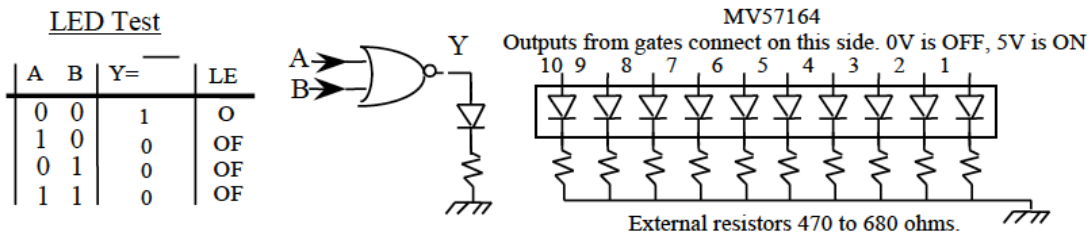

Fig. 9.9. LED test circuit.

Experiment

LED TESTING

Before doing anything else, check that each LED lights up when the positive end is connected to the 5 V supply. If it fails to light, check the polarity.

TRUTH TABLES FOR TTL GATES

· Verify the truth tables for the NAND (7400), NOR (7402), and INVERT (7404) gates, using the LED indicators.

· Connect a NAND gate so that it performs the INVERT function. Do this for a NOR gate also. This trick will be convenient in simplifying in complex circuits.

Occasionally you will find a non-functioning gate. Throw the chip into the trash once your are sure it is bad. Remember that most problems arise from wiring mistakes.

EXCLUSIVE OR

· Verify the truth tables for an EXCLUSIVE OR chip (7486).

· Now build and test the XOR circuit of your own design using only the NANDs and NORs.

THE RS FLIP-FLOP

· Build an RS flip-flop from two NOR gates.

· Demonstrate the memory property by going through a complete memory cycle: Set (R = 0, S = 1), Store (0, 0), Reset (1, 0), Store (0, 0), Set (0, 1).

· Examine the effect of the "illegal" input (R = 1, S = 1), for different initial states of the RS system.

TTL CLOCK

· Build the 1 kHz digital clock using a 555 Timer according to your design in problem 5. Verify with the oscilloscope that the frequency, the pulse length of 750 msec, and the nominal 5 volt amplitude are approximately correct.

· Check that a suitable large capacitor placed in parallel with the existing one converts the clock to 1 Hz.

· Set up a NAND gate to control the transmission of clock pulses by means of a logical 0 or 1 control voltage. The output pulses for the NAND should be positive.
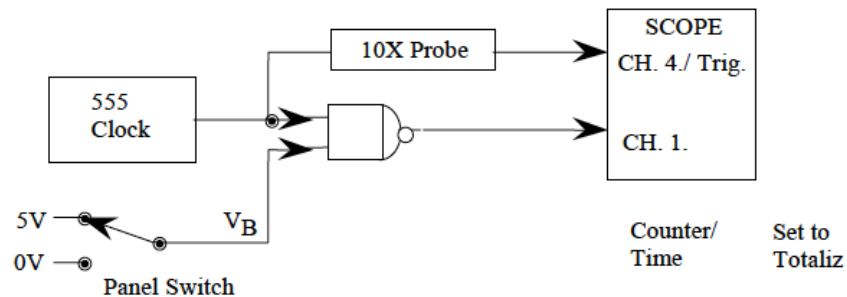

Fig. 9.10. Digital clock

THE JK FLIP-FLOP

· Construct a truth table for the JK from your observations using the LED indicators. Since the output depends upon the previous state, Q, you will need to tabulate Q_{n+1} for both possible previous states, $Q_n=0$ and $Q_n=1$. We suggest that you add a redundant column, Q_{n+2} , (see truth table in Fig. 9.6) to get a better feel for the behavior of the flip-flop.

· Set CLR = 1 and J = K = 1. Now drive the clock input of the JK with 1 kHz pulses from your clock circuit as shown in Fig. 9.11. Use the oscilloscope to observe the clock input (positive pulses out of the NAND gate), and the output, Q, of the JK. What happens when J = K = 0?
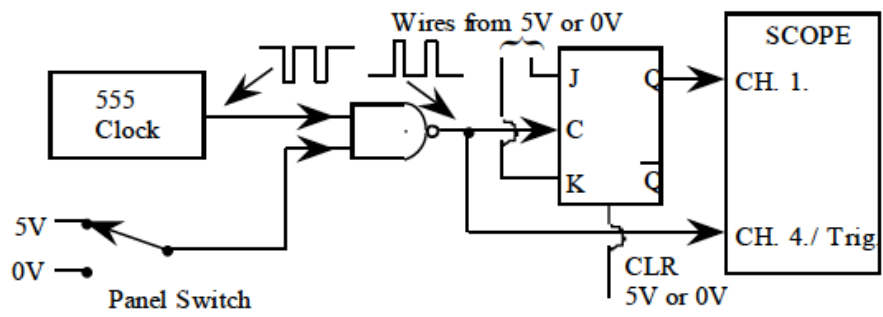
Fig. 9.11. JK test circuit.