EXPERIMENT 6

Timing and Counting Circuits

Introduction

The experiments in this laboratory exercise will help you learn about digital timing and counting circuits. You will learn how to build two common circuits with a 555 timer chip, and how to use a 74169 binary counter chip.

Experiment Objectives:

- Learn how to build astable and monostable timing circuits.
- Learn how to build a binary counter circuit
- Continue to develop proficiency with the Bit Bucket breadboarding system and the oscilloscope.
- Continue to develop professional lab skills and written communication skills.

Bring to Lab:

Your completed Pre-Lab. Turn this in when you get to lab.
Several sheets of Engineering Paper.

Theory: Timing Circuits (Clocks)

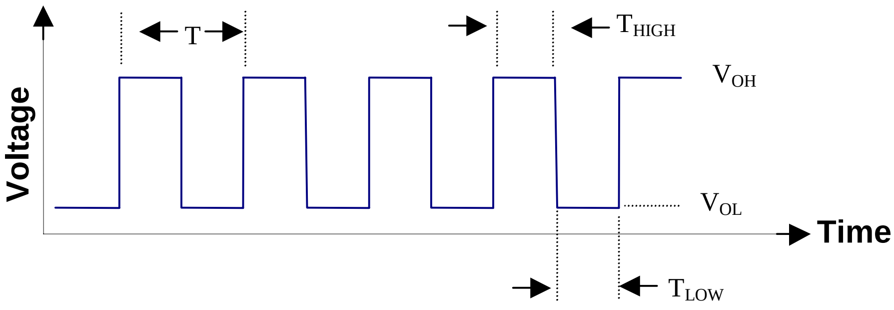
All computer systems require timing signals to keep their internal operations synchronized. Also, we often use electronic circuits to operate devices or record information in circumstances where time is a crucial factor. Therefore, we need to be able to generate logic signals that occur at precise time intervals.

In this experiment, we will learn to generate two types of timing signals- a repeating sequence of pulses, and a one-time, fixed duration pulse.

Repeating Sequence of Pulses (Astable)

This type of timing signal is called *free-running*, or *astable*. A diagram of this signal is shown in Figure 1.

Figure 1. Astable (Free-Running) Clock Signal



A circuit that generates this signal is often called a *clock circuit*, and the signal is often called a *clock signal*, or just a *clock*. The signal from a clock will normally start by itself whenever the power to the circuit is turned on. Some clock circuits have an extra *enable* input that allows the clock to be interrupted or paused. The astable clock signal shown in Figure 1 is characterized by the properties listed in Table 1.

Table 1. Astable (Free-Running) clock signal properties shown in Figure 1.

Property	Notation	Units
Period	T	time
Output low interval	T_{LOW}	time
Output high interval	T_{HIGH}	time
Duty Cycle	T_{LOW} / T	percentage
Output high voltage	V_{OH}	volts
Output low voltage	V_{OL}	volts

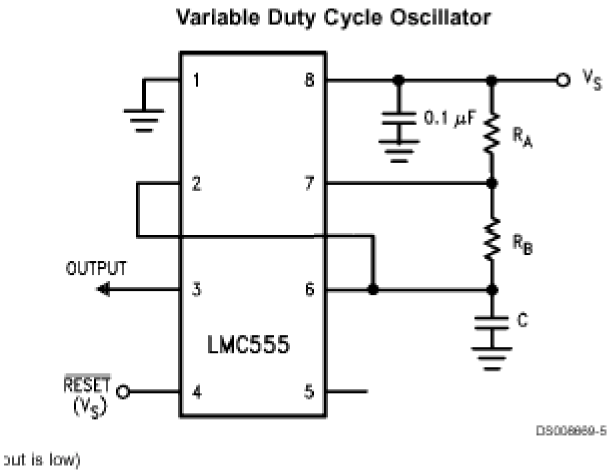
Building an Astable Clock Signal Generator Using the 555 Timer Chip

The 555 Timer is an industry standard 8-pin integrated circuit widely used to generate clock signals. (This is the chip used to generate the on-board clock signal on the Bit Bucket.) Several manufacturers provide essentially identical versions of the 555 chip. The data sheet for the part we will use (LMC555) can be accessed from National Semiconductor at

http://www.national.com/pf/LM/LMC555.html

The circuit diagram from that data sheet is copied below in Figure 2. There are three required external parts which determine the timing: two resistors labeled R_A and R_B, and a capacitor labeled C.

Figure 2. Astable Circuit Using the 555 Timer Chip



The timing equations for this circuit are as follows:

$$T_{LOW} = ln(2)R_B C = 0.693 R_B C \tag{1}$$

$$T_{HIGH} = ln(2)\left(R_A + R_B\right)C = 0.693\left(R_A + R_B\right)C \tag{2}$$

$$T = \text{Period} = T_{LOW} + T_{HIGH} = ln(2)\left(R_A + 2R_B\right)C = 0.693\left(R_A + 2R_B\right)C \tag{3}$$

$$D = \text{Duty Cycle} = \frac{T_{LOW}}{T} = \frac{R_B}{R_A + 2R_B} \tag{4}$$

One further constraint is placed on R_A:

$$R_A \text{ must be no less than 1 k}\Omega. \tag{5}$$

We can use these equations to design a clock circuit as illustrated in the following example.

Example 1 – Astable Design

Design a 555 astable circuit to have a frequency as close as possible to 1 kHz and a duty cycle as close as possible to 50%. You have access to capacitor values .01, .1, 1, and 10 μF, and resistor values 1, 10, and 100 kΩ.

Solution: From Eqn. 4, we see that choosing the smallest possible value for R_A will get D as close as possible to 50%. Equation 5 says the smallest allowed value for R_A is 1 kΩ. So we choose

$$\boxed{R_A = 1\text{ k}\Omega.}$$

The required period is

$$T = 1/f = 1/(1000\text{ Hz}) = 1\text{ ms}.$$

In Table 2 we try a few combinations of R_B and C, and calculate T using Eqn. 3 and D using Eqn. 4:

Table 2. Using trial and error to find the best combination of R_B and C.

R_B (kΩ)	C (μF)	T (ms)	D (%)
10	.01	0.146	47.6
100	.01	1.39	49.8
10	.1	1.46	47.6

The second row is the combination that yields T closest to 1 ms and D closest to 50%. So we choose

$$\boxed{R_B = 100\text{ k}\Omega \quad C = .01\ \mu\text{F}}$$

This is the best design using the limited number of fixed component values allowed. Of course, if we have a wider selection of resistors and capacitors to choose from, or if we can use potentiometers, we can achieve f = 1 kHz much more precisely.

One-time, fixed duration pulse (Monostable)

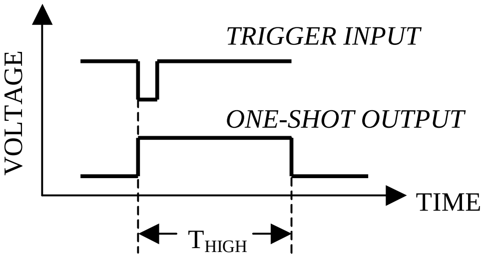
The 555 can be used to generate single pulses of fixed duration. The duration is set by an external resistor and capacitor. The pulse is started by an external trigger signal. The circuit that generates this type of signal is called a monostable or one-shot circuit. The signal is shown in Figure 3, and the circuit is shown in Figure 4.

The timing equation for the monostable circuit is:

$$T_{HIGH} = (1.1)R_A C \tag{6}$$

where T_{HIGH} is the duration that the output remains high after the trigger signal is applied. As shown in Figure 3, the output pulse starts on the *falling edge* of the trigger. The falling edge is when the trigger changes from logic HIGH to logic LOW.

Figure 3. Monostable Waveforms



Here are two additional facts about the one-shot circuit:

- The trigger pulse should be shorter than the output pulse. Otherwise, the output will remain HIGH for the duration of the trigger rather than the preset time T_{HIGH}.
- Once the output is triggered, additional trigger signals will have no effect during T_{HIGH}.

We can use Eqn. 6 to design a one-shot circuit as shown in Example 2.

Example 2 – Monostable Design

Design a 555 one-shot circuit to have a pulse width as close as possible to one second (1 s). You have access to capacitor values .01, .1, 1, and 10 μF, and resistor values 1, 10, and 100 kΩ.

Solution: In Table 3 we try a few combinations of R_A and C, and calculate T_{HIGH} using Eqn. 6:

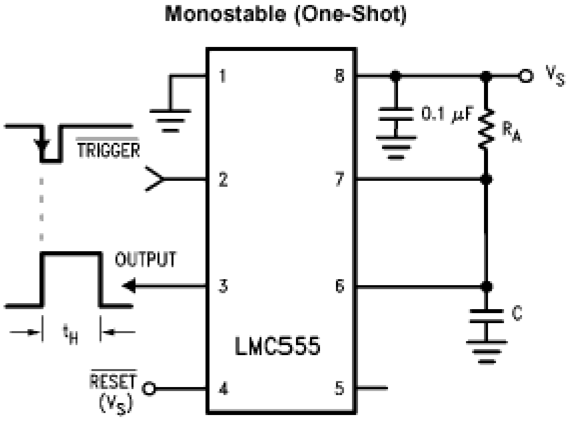
Table 3. Using trial and error to find the best combination of R_A and C.

R_A (kΩ)	**C (μF)**	**T_{HIGH} (ms)**
10	.01	0.11
100	.01	1.1
100	10	1100

The last row is the combination that yields T closest to 1s (= 1000 ms). So we choose

R_A = 100 kΩ C = 10 μF

Figure 4. Monostable Circuit Using the 555 Timer Chip.



Theory: Counting Using the 74169 Up/Down Binary Counter Chip

There are many uses for counting in digital systems. Here we discuss counting in the binary number system, the fundamental language of computers, and then we discuss the 74169 counter.

The Binary number system

The binary, or base-2 number system has only two digits – zero (0) and one (1). The decimal number system represents numbers by integers 0 through 9 multiplied by powers of ten; the binary number system represents numbers as integer multiples (0 or 1) of powers of two. The integers 0 through 9 in the decimal system are called digits. In the binary system, the integers 0 and 1 are called *bits* (short for binary digits).

The relationship between binary and decimal is illustrated with two examples below. Many scientific calculators have a built-in base conversion function that converts between decimal and binary automatically.

Example 3: Convert the binary number 1101_2 to decimal.
Solution: Each bit multiplies a power of 2 as shown in the table below:

Table 4. Converting binary to decimal.

Bit (binary digit), B	1	1	0	1	
Place (power of two), n	3	2	1	0	
2^n	8	4	2	1	
$B*2^n$	8	4	0	1	sum = 13_{10}

Each bit multiplies its corresponding power of two, resulting in the terms in the last row. The terms are summed to get the decimal value. Thus $1101_2 = 8 + 4 + 0 + 1 = 13_{10}$.

Example 4: Convert the decimal number 26 to binary.
Solution: Start by making a table containing the powers of two up to the highest one that is less than 26. Working in sequence starting with the highest power of two (in this case 16), if you can subtract it from the current remainder without getting a result less than zero, do so, and then write a value of one for the corresponding bit, otherwise write 0. (The first remainder is the number you are converting.) The results for our example are shown in Table 5.

Table 5. Converting decimal 26 to binary 11010.

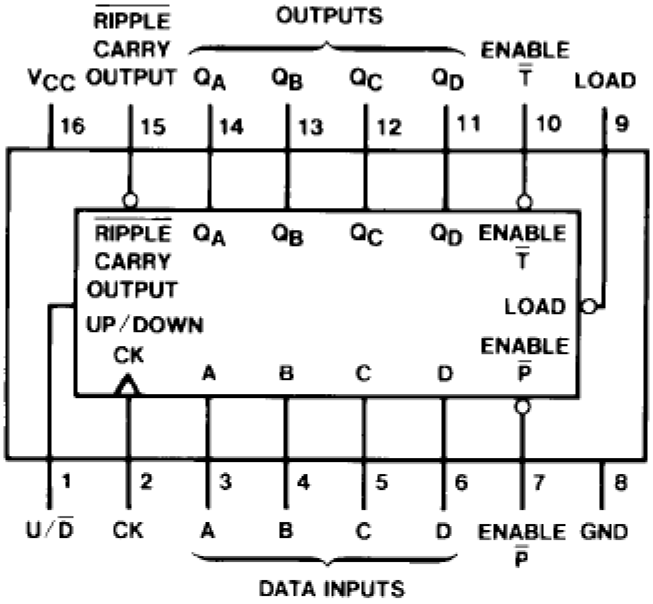
Remainder	2^n	can I subtract?	new remainder	bit
26_{10}	16	yes	10_{10}	1 (MSB)
10_{10}	8	yes	2_{10}	1
2_{10}	4	no	2_{10}	0
2_{10}	2	yes	0	1
0	1	no	0	0 (LSB)

In this case, 26 – 16 leaves 10, so the highest bit (called the most-significant bit, or MSB) is 1 and the remainder is 10. Since 10 – 8 leaves 2, the next bit is 1 and the remainder is 2. The next bit is 0 since we cannot subtract 4 from 2. The next bit is 1, since 2 – 2 = 0. The last bit is 0 since 1 cannot be subtracted from 0. (The last bit is called the least significant bit, or LSB). So we can write $26_{10} = 11010_2$.

Using the 74169 counter chip

The 74169 counts pulses applied to its input pin, and outputs the running total as a four-bit binary value. When the count reaches $15_{10} = 1111_2$, the next pulse causes the output to roll over back to zero (0000_2). There is a separate pin (U/D) that allows you to specify whether the counter counts up or down. The pin-out is shown below in Figure 5. The entire data sheet is available at http://www.fairchildsemi.com/pf/DM/DM74ALS169B.html .

Figure 5. Pin-out Diagram for the 74169 Binary Counter



The outputs are Q_D (MSB), Q_C, Q_B, and Q_A (LSB). The input pulses to be counted are applied to pin 2 (CK). Logic 1 on pin 1 causes the chip to count up, whereas logic 0 causes it to count down. In order to operate, power (5 V) and ground must be applied to pins 16 and 8 respectively, and pins 7 and 10 (ENABLE T-bar[*] and ENABLE P-bar)must be grounded. The remaining pins (3, 4, 5, 6, 9, 15) will not be used for our experiment, and may be left unconnected.

---

[*] Pin functions that are TRUE when the voltage is logic LOW are indicated by an overbar and/or with an open circle. For example, pin 1 is read “up slash down-bar,” which means DOWN (count down) is TRUE when that input is logic LOW.

Your Name____

Prelab Questions (10 points)

Answer these questions before coming to lab and turn them in when you arrive. You may do your work on separate paper (for example you might want to do your work on a computer), but please attach your work to this sheet for submission.

1. Following Example 1, design an astable circuit with frequency close to 10 kHz and duty cycle close to 50%. You have access to the same component values as those given in the example.

2. Convert 10010111_2 to decimal.

3. The present output of a 74169 is shown below. What is the output after 5 more input pulses?

	Q_D (MSB)	Q_C	Q_B	Q_A (LSB)
Present output	1	1	0	1
Output after 5 more input pulses				

4. Design a circuit that will produce a pulse each time a button is pushed, and count the number of pulses. The pulse width should be close to 1 second. Use a 555-based one-shot connected to a 74169 counter. You have access to the same component values as in Example 2. Draw each chip with pin numbers labeled, and show all connections and component values.