Experiment 7
Digital Logic Devices and the 555-Timer

Purpose: In this experiment we address the concepts of digital electronics and look at the 555-timer, a device that uses digital devices and other electronic switching elements to generate pulses.

Background: Before doing this experiment, students should be able to

- Analyze simple circuits consisting of combinations of resistors, inductors, capacitors and op-amps.
- Do a transient (time dependent) simulation of circuits using Capture/PSpice
- Build simple circuits consisting of combinations of resistors, inductors, capacitors, and op-amps on protoboards and measure input and output voltages vs. time.
- Review the background for the previous experiments.

Learning Outcomes: Students will be able to

- Identify basic logic gates and look up and use their truth tables
- Experimentally generate a truth table for basic logic gates using the Static I/O functionality of WaveForms.
- Simulate basic combinational and sequential logic gate configurations, generating a timing diagram with PSpice
- Characterize the operation of a JK Flip-Flop both experimentally and using PSpice simulation
- Characterize the operation of a Binary Counter both experimentally and using PSpice simulation
- Characterize the operation of a 555 Timer in Astable Multivibrator configuration both experimentally and using PSpice simulation
- Use a counter to count the pulses produced by a 555 Timer Astable Multivibrator.

Equipment Required:

- Analog Discovery (with Waveforms Software)
- Voltmeter (DMM or Analog Discovery)
- Oscilloscope (Analog Discovery)
- Function Generator (Analog Discovery)
- +5V (+Vcc) Power Supply (Analog Discovery, be sure to use V+ and Ground and not V-)
- 555-Timer, 7402, 7404, 7410, 7414, 74107, 74393 ICs

Helpful links for this experiment can be found on the links page for this course.

Pre-Lab

Required Reading: Before beginning the lab, at least one team member must read over and be generally acquainted with this document and the other required reading materials listed under Experiment 7 on the EIlinks page.

Hand-Drawn Circuit Diagrams: Before beginning the lab, hand-drawn circuit diagrams must be prepared for all circuits either to be analyzed using PSpice or physically built and characterized using your Analog Discovery board.

Part A – Basic Logic Gates

Background

Digital logic gates: All digital logic gates are based on binary logic. Binary logic has two values, called TRUE and FALSE, LOGIC 1 and LOGIC 0, ON and OFF, or HIGH and LOW. The corresponding binary number can have two possible values, 1 and 0. Digital logic gates perform many common logic operations on binary signals, such as AND, OR, NOT, NAND, and NOR. The table in Figure A-1 contains the common symbol for each type of gate, an expression for the function of the gate in Boolean algebra, and a truth table for the device. The truth table shows how the gate will behave for all possible combinations of digital inputs.

K.A. Connor, S. Bonner, P. Schoch

Rensselaer Polytechnic Institute

Revised: 29 March 2018

Troy, New York, USA
Digital logic chips: In TTL (transistor-transistor logic) digital electronic circuits, the representation of binary numbers in terms of voltages is about 5V for LOGIC 1 and about 0V for LOGIC 0. About 5V usually means a voltage between 2 and 5V while about 0V means any voltage in the range 0 to 0.8V. The voltage levels when using TTL devices must always be in the ranges indicated or the circuits will not function correctly. LOGIC 1 and LOGIC 0 are the only output levels one should see with logical devices. This is one characteristic that makes them differ from analog devices. They also switch very fast from one state to the other. Switching speeds are usually much faster than for analog devices, especially cheap devices like the 741 op-amp.

A digital chip generally has 14- or 16-pins. It usually contains more than one of the same logic gate. (For example, a 14-pin chip will have six single-input gates or four 2-input gates.) By convention, the upper right pin on a digital chip is always connected to HIGH (+Vcc) and the bottom left pin to LOW (0V). Vcc is usually either +4V or +5V, depending on what supplies are available. Most logic chips will operate over a range of supply voltages. On a 14-pin chip this corresponds to pin 7 (0V) and pin 14 (+Vcc) and on a 16-pin chip, pin 8 (0V) and pin 16 (+Vcc). These two connections provide reference values for the operations the chips perform. Generally circuit diagrams do not show these two reference connections. If you forget to connect these two pins, your circuit will not function.

Timing diagrams: Timing diagrams are a special kind of transient output which are useful for viewing many binary signals. Since the voltage levels of binary signals can only be either high or low, knowing the exact voltage level is not as significant as knowing when the different signals transition from high to low (or low to high). A timing diagram is much easier to read when you need to compare many binary signals. Unlike a regular PSpice plot (where all signals are displayed with the same voltage and time axis) a timing diagram displays the signals on separate lines with the same time scale. A sample is shown in Figure A-2.

Note that there are six signals shown on this plot. We can see where they are high and low and we can also see the relative time that each signal changes state.
Experiment

Truth Tables of Basic Logic Gates
We will now consider three basic logical elements: a two input NOR gate, a three input NAND gate and an INVERTER.

- Wire the circuits in Figure A-3 on your protoboard. Do not forget to tie pin 14 to +Vcc (+5V) and pin 7 to 0V (Ground) on each chip. (Also note that the 74107 chip and the 7410 chip in your kit are not the same chip. We want the 7410 here.) The protoboard, with and without connections to Analog Discovery, is also shown.

![Diagram of circuits](image-url)
For the first measurements, we use the Digital Static I/O feature of Analog Discovery. Enable Static I/O and be sure that all 16 channels are configured Bit I/O (menus on the left) and channels 8-15 are LED (for measurements, the LED is on when the voltage is HIGH and off when it is LOW) and channels 0-7 are Push/Pull Switch (for output voltages, when the switch is up, the output is HIGH and its little LED in the upper right hand corner will turn on or when the switch is down, the output is LOW and its LED will turn off). We will use the outputs from Digital Channels 0, 1, 2 to provide inputs to the gates and inputs to 8, 9, 10 to measure the outputs from the gates.

- Consider all possible combinations of inputs to generate a truth table for each device.
  - The NOT gate has only one input. Therefore, we need only need to observe the output when the input is HIGH and LOW.
    - First set channel 0 to HIGH and record the status of channel 10.
    - Then set channel 0 to LOW and record the status of channel 10.
    - Does the gate invert the input?
    - Take a picture (screen capture) of one of the input/output combinations.
    - Record the truth table for this gate. Include the picture and truth table in your report.
  - Now you will repeat this process for the other two gates.
    - The NOR gate has two inputs, so we must observe the output at pin 1 (Digital I/O channel 8) for all possible combinations of binary inputs at pins 2 and 3: (LOW, LOW), (LOW, HIGH), (HIGH, LOW) and (HIGH, HIGH).
    - Take a picture (screen capture) of one of the input/output trace combinations.
    - Record the truth table on the plot. Include the picture and truth table in your report.
    - The NAND gate has three inputs. How many combinations of HIGH and LOW are required to fully test this gate?
• Record the input and output for this gate in a truth table and a sample screen capture of one input/output combination, as well. Include the picture and truth table in your report.

Simulation of Basic Logic Gates
We will now wire the same three basic logical elements using PSpice.

• Create the following circuit (Figure A-5) in PSpice. Note that Place Net Alias has been used to name the nodes whose voltages are to be displayed. Using this feature helps to more easily identify the voltages plotted.

Figure A-5.

- Wiring the circuit in PSpice is somewhat different than on the protoboard.
  - PSpice assumes that the +5V and 0V references have already been wired, so you do not need to make these connections.
  - We cannot simply move the wires to record all possibilities. Therefore, we use digital clocks with different pulse lengths to create the signals we need to test the gates.
  - We have removed the resistors connecting the gate outputs to ground. This tells PSpice to output timing diagrams by default.
- Now we need to set the clocks up to work with different pulse lengths.
  - Use DigClock in the SOURCE library
  - Use the default settings for DSTM1 (no delay, on time = 0.5us, off time = 0.5us).
  - For DSTM2, double the on and off times to 1us.
  - For DSTM3, double them again to 2us.

• Run a simulation
  - Simulate for 8us with a step size of 0.01us.
  - Display all the inputs from the clocks and the output of each of the three gates.
  - Produce a hardcopy of the timing diagram with the inputs and the outputs for all three gates.
    - Mark the output trace for each gate on the diagram. For each gate, generate the truth table for the device based on the outputs and inputs you observe on the timing diagram. Write them on the output plot.
    - Include this plot in your report.
    - Do your results agree with the truth tables you found using from the circuits you built?

Summary

Basic logic gates allow you to use electronic signals to perform operations on digital signals. They can also be combined to perform more complex operations, such as addition and subtraction. This makes them a basic building
block of digital computers. We have only considered the most common gates. You should look up others such as XNOR, buffer, and negative input gates. The latter negates the input before applying the gate function.

Part B – Flip Flops

Background

Flip Flops: It is possible using basic logic gates to build a circuit that remembers its present condition. These circuits are called flip flops. The PSpice symbol for a JK flip flop is pictured in Figure B-1. There are several different kinds of flip flops with slightly different characteristics. In this course we use the JK flip flop. JK flip flops, like other flip flops have four inputs, two outputs and the usual two power connections (Vcc and ground). The outputs are labeled Q and Q (also called Qbar and NQ). They are complements of one another. Thus, when Q is LOW, Qbar is HIGH, etc. CLK is the digital clock. A flip flop only changes its output when the clock pulse at CLK goes from HIGH (+Vcc) to LOW (0V). This is called the “falling edge” of the clock. The input CLR, when LOW, will reset the outputs to a known state. It has the following truth table:

<table>
<thead>
<tr>
<th>J</th>
<th>K</th>
<th>C</th>
<th>Q</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>p</td>
<td>no change</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>p</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>p</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>p</td>
<td>toggle</td>
<td></td>
</tr>
</tbody>
</table>

Figure B-1.

Note that the flip flop is an edge-triggered device. This means that instead of changing state as soon as its inputs change, it “waits” until it receives a falling edge at the clock input (CLK). To decide how to set the output, the flip flop “looks” at the values at the inputs at J and K AND at the current value of the output. Based on these three values, it decides how to reset the output.

You may be familiar with clocks. They are used to coordinate the instructions performed by the CPU and other devices in a computer. When a computer has a clock speed of 1GHz it can handle 1×10^9 instructions per second; one for every clock cycle. The flip flop works on the same principle. With every clock cycle, it looks at its inputs and changes state accordingly.

The flip flop is a memory device. If both inputs are zero, its output will remain the same indefinitely. A bank of 4 flip flops can store a digital byte of memory in a computer. If you want to change the value of a single bit of that byte to zero, you can set the inputs to the corresponding flip flop to J = 0 and K = 1. On the next clock cycle, the output will change to zero. If you want to change a single bit of the byte to one, you can set the inputs of the corresponding flip flop to J = 1 and K = 0. On the next clock cycle, the output will change to 1. You can also toggle the value of the flip flop output by setting both inputs to 1.

Timing diagram for a flip flop: To illustrate the behavior of a JK flip flop, we wired the circuit in Figure B-2 in PSpice and created the timing diagram shown in Figure B-3.
DSTM2 is the clock. Each time the clock falls, look at the values of the inputs at J (DSTM1), K (DSTM3), and the output signal (U1A:Q). If the truth table is correct, what should the output value at U1A:Q be for each combination? Is the timing diagram correct? Please check the datasheet for the SN74107 flip flop located on the course web page for details about this device.

**Noise and Digital Circuits:** Any time you build a circuit, there will be noise. In an edge-triggered device, where timing is a factor, noise can cause many problems. A noise spike might be interpreted as the falling edge of the clock. If this happens, the flip flop will change state at the wrong time and possibly with the wrong inputs. To avoid this problem, it is essential to use a **bypass capacitor** in every timed digital circuit you build. A bypass capacitor is simply a capacitor placed between the source voltage and ground. What it does is filter out high frequency noise, so that any spikes that might be misinterpreted are filtered out. Figure B-4 shows a bypass capacitor.

To understand how the bypass capacitor works, we only need to recall that a capacitor looks like an open circuit at low frequencies and a short at high frequencies. Digital signals consist of two DC values, 0V and +Vcc. A DC voltage has zero frequency. Therefore the bypass capacitor will look like an open circuit and all of the DC signal will pass into the circuit. A noise spike is a very sudden high frequency event. At high frequencies, the capacitor looks like a short. Therefore, the high frequency event will pass through the capacitor to ground and not continue on into the circuit.

When a signal changes from low to high (or high to low), the temporary noise of the transition may cause the device to make the wrong output decision. Since the values of the inputs, as they are changing, are indeterminate, the output that a flip flop will generate if the inputs are changing is unknown. In PSpice, this unknown state is depicted by a double red line. If you look at the timing diagram above you will see that both Q and Qbar start out in an unknown state until the first falling edge of the clock. Also notice that the transitions for J and K take place well before the falling edge of the clock. We deliberately set up the timing of DSTM1 and DSTM3 so that they do not change the state of the input at the same time the clock transitions from high to low. If we had, PSpice would
continue to display the double red line, signifying that it cannot properly set the output because it isn’t sure what the inputs are supposed to be.

Experiment

*The JK Flip Flop*

In this part of the experiment, we will look at the behavior of a flip flop on your protoboard.

- Set up the 74107 JK flip flop on the protoboard.
  - Provide 0V at pin 7 of the chip and +Vcc to pin 14.
  - Place a 0.1uF by-pass capacitor between 0 and +Vcc.
  - Use three Digital I/O channels to create the J, K, and Clock signals. These should be set as Push/Pull Switches (outputs) to provide the three inputs to the flip flop.
  - Use 2 Digital I/O channels to read the Q and Qbar signals. These are set as LEDs (inputs) to sense the outputs from the flip flop.
  - Set the CLR to zero to be sure that the flip flop begins in a known state. THEN attach it to +Vcc to enable the function of the chip. (Or use yet another Digital I/O channel as an output and cycle it low and then high. Either approach is fine, but the latter provides a simple switch to cycle the value.)
  - Run all possible combinations of logic levels for J and K (see below). *Each time you change J or K, you need to cycle the Digital I/O that is the Clock signal. This means to make the changes in J and/or K and then switch the clock from low to high and then back to low.* By monitoring the Q output, you should be able to complete the table below. It may take a little thought. You should notice whether the flip flop is cycling when the clock goes up (leading edge) or down (trailing edge).
  - Fill in the following truth table for this device. You will not be able to fill in the table in order. You will need to skip around depending on the current state (before pulse) of the flip flop.

<table>
<thead>
<tr>
<th>Q (before pulse)</th>
<th>J</th>
<th>K</th>
<th>Q (after pulse)</th>
<th>Qbar (after pulse)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Take a screen capture of the Digital I/O status for the case when J and K equal to 1. Label each Digital I/O channel used with J, K, Clock, Q and Qbar. Include the truth table above for the flip flop and the screen capture in your report.

Summary

The JK flip flop is an edge-triggered device that uses an external clock to coordinate state changes with other clocked devices in a circuit. It is capable of holding its output stable, changing its output directly to high or low, or toggling the output to the opposite of its current value. It can be used as a memory device or as a building block to create more complex devices, such as counters.

Part C – Counters

Background

*K.A. Connor, S. Bonner, P. Schoch*
**Binary Counters:** JK flip flops can be connected in a counter configuration as shown in Figure C-1. The output of each flip flop is used as the input clock to the next flip flop. On all flip flops, J and K are tied high. This means that they will toggle on each pulse of their clocks. The first flip flop toggles every clock cycle. The second flip flop toggles every time the output from the first flip flop changes. This causes it to toggle at a rate equal to half of the clock rate. By the same reasoning, the third flip flop toggles at a rate ¼ of the clock rate.

![Figure C-1](image)

If we attach a bit of a binary number, \((b_0, b_1, b_2)\), to the output of each flip flop \((b_0\) to the first, \(b_1\) to the second, and \(b_2\) to the last). We get a pattern out which corresponds to the binary counting shown in Figure C-2 below:

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Binary</th>
<th>Hex</th>
<th>Octal</th>
<th>Decimal</th>
<th>Binary</th>
<th>Hex</th>
<th>Octal</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>00000</td>
<td>00</td>
<td>00</td>
<td>08</td>
<td>01000</td>
<td>08</td>
<td>10</td>
</tr>
<tr>
<td>01</td>
<td>00001</td>
<td>01</td>
<td>00</td>
<td>09</td>
<td>01001</td>
<td>09</td>
<td>11</td>
</tr>
<tr>
<td>02</td>
<td>00010</td>
<td>02</td>
<td>00</td>
<td>10</td>
<td>01010</td>
<td>0A</td>
<td>12</td>
</tr>
<tr>
<td>03</td>
<td>00011</td>
<td>03</td>
<td>00</td>
<td>11</td>
<td>01011</td>
<td>0B</td>
<td>13</td>
</tr>
<tr>
<td>04</td>
<td>00100</td>
<td>04</td>
<td>00</td>
<td>12</td>
<td>01100</td>
<td>0C</td>
<td>14</td>
</tr>
<tr>
<td>05</td>
<td>00101</td>
<td>05</td>
<td>00</td>
<td>13</td>
<td>01101</td>
<td>0D</td>
<td>15</td>
</tr>
<tr>
<td>06</td>
<td>00110</td>
<td>06</td>
<td>00</td>
<td>14</td>
<td>01110</td>
<td>0E</td>
<td>16</td>
</tr>
<tr>
<td>07</td>
<td>00111</td>
<td>07</td>
<td>00</td>
<td>15</td>
<td>01111</td>
<td>0F</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>16</td>
<td>10000</td>
<td>10</td>
<td>20</td>
</tr>
</tbody>
</table>

![Figure C-2](image)

The lowest order bit of the binary numbers toggles from 0 to 1: \((0-1-0-1-0-1-\ldots)\). The second order bit also toggles between 1 and 0, but at half the rate: \((0-0-1-0-1-1-0-\ldots)\). The third order bit toggles also, but at half the rate of the previous bit: \((0-0-0-0-1-1-1-0-\ldots)\). The pattern in the table is the same as the pattern created by the cascaded set of flip flops. Thus, the counter is counting.

It is not necessary for us to configure several flip flops to create a counting circuit, because this is already done in many kinds of chips. The SN74393, for example, has two sets of four JK flip flops connected as binary counters.

**Light Emitting Diodes:** LEDs (light emitting diodes) are very useful when dealing with digital circuits. Because they have two states, ON and OFF, you can hook them to a binary signal, and use them directly to observe whether or not the signal is HIGH or LOW. Although there are conventions for the polarity of diodes, many manufacturers do not seem to follow them. The best way to determine the polarity of a diode is to place it directly between +Vcc and ground (with a series resistor). If it lights, use it in that direction. If it doesn’t light, try switching the polarity. If it lights that way, then use that polarity. If it lights in neither polarity, throw it away. It is burnt out.

LEDs are like light bulbs. They burn out after a while. In order to prolong the life (and brightness) of an LED, it is a good idea to wire it in series with a small resistor (330 Ohms is about right). If the LED is not bright enough, you can replace the resistor with a smaller one. You will see how to determine the optimum value for the current limiting resistor in Experiment 8. In the following experiment, the resistor is chosen to be 220 Ohms.
Experiment

Simulation of a cascaded binary counter
In this part of the experiment, we will use PSpice to create a simulation of two counters cascaded together. This will allow us to count to numbers greater than 15.

- Simulate the circuit in Figure C-3 in PSpice.

![Circuit Diagram]

**Figure C-3.**

- DSTM1 is the actual clock for the counter, DigClock in SOURCE library with OFFTIME = ONTIME = 0.5us.
- DSTM2 (also DigClock) is set up so that it first clears the counters, lets them count, and then clears them again. (OFFTIME = 45us, ONTIME = 0.5us, DELAY = 0.7us)
- If the PSpice models for counters or flip flops are not cleared initially, they will have indeterminate data.
- By connecting the two counters together as shown, the sequence of numbers 2QD, 2QC, 2QB, 2QA, 1QD, 1QC, 1QB, 1QA (IN THAT ORDER) form the binary number. Since the counter is set up to count clock pulses, it will count up from 0 (at reset) to the number of pulses sequentially.

- Run the simulation.
  - Generate the output for this circuit for time 0 to 100us using 1us increments.
  - You should display the reset pulse, the input clock at pin 1 of U1A, and all eight of the counter outputs (QA, QB, QC, QD for both counters, selected in order from most significant to least significant bit).
  - Using the timing diagram, verify that the counters are actually counting. You can use the cursor to easily get your binary number. The binary value is displayed on the left by the y-axis.
  - What is the highest number it counts to before it resets? Express this number both as a binary number and a decimal number. At what time does it reset?
  - Write it on the timing diagram for the circuit. How many pulses will the clock have to cycle through between the time it is reset and when it hits its maximum value of 11111111?
  - Include the timing diagram output in your report.

Build a counter circuit
In this part of the experiment, we will build a counter circuit on the protoboard.

- Build the circuit in Figure C-4 on your protoboard.
Figure C-4. The Schmitt trigger inverter cleans up the signal from AWG1. Don’t forget that the 74LS14 and the 74LS393 both need power and ground.

- Use the function generator for the clock. Set it for a square wave with a frequency of 20Hz. We are using a fairly low frequency so that we will be able to see the switching with the LEDs. *Use the offset feature to shift the square wave up such that it cycles between 0V and 4V.* With the Analog Discovery Waveform Generator, setting the offset to 2V and the peak-peak voltage to 4V will output the required signal. Be sure that you use the oscilloscope to check the operation of the function generator.
- Tie pin 7 to ground and pin 14 to +Vcc on both the 74LS14 and the 74LS393 logic chips.
- Place a 0.1uF by-pass capacitor between +Vcc and ground.
- Check your LEDs by connecting one with a 220 Ohm resistor between +Vcc and ground to see which polarity causes it to light. Place it in the circuit with the correct polarity. Usually the lead on the flattened side of the case goes to ground.
- If you cannot find 220 ohm resistors, use values close to 220 ohms.
- Set the CLR to +Vcc to be sure that the counter begins counting at zero. THEN attach it to 0V to enable the chip to function. Note that this is the opposite of the flip flop. The circle at the input to the clear tells you whether the CLR signal must be high or low. (Circle ➔ Hold high to disable clear & enable counting. No Circle ➔ Hold low to disable clear & enable counting.) If you tie this pin incorrectly, the chip will continuously reset itself and will not work.

- Once you have the circuit working, observe the output.
  - Are they changing at the expected rates? Change the function generator frequency if it helps you determine the frequencies.
  - Place channel 1 of the scope on the clock signal and set the time controls so that it displays 50 clock cycles. Do not change the time scale as you take your pictures. All four should show 50 clock cycles for comparison purposes.
  - You will need to take four pictures using Analog Discovery. Include them in your report.
    - Channel 1 = clock and Channel 2 = QA
    - Channel 1 = clock and Channel 2 = QB
    - Channel 1 = clock and Channel 2 = QC
    - Channel 1 = clock and Channel 2 = QD
  - What do you observe about the rates of the different signals with respect to the clock? Can you tell which output corresponds to which bit in the binary number?
  - Keep this circuit for part D.

Part D – The 555-Timer

Background

*The 555-Timer:* The 555-timer is a chip that allows us to create a variety of useful digital and analog signals. Much like the op-amp, it can be used to perform different functions depending upon what circuit you place it into. The

K.A. Connor, S. Bonner, P. Schoch

Rensselaer Polytechnic Institute

Troy, New York, USA

Revised: 29 March 2018
555-timer can be used to generate digital pulses. When it is wired as a “one-shot” (also called mono-stable mode), it generates a single, clean, digital pulse at the output, when it experiences a (possibly noisy) pulse at the input. This is useful when de-bouncing a mechanical switch (Project 3). In this experiment, we are concerned with the 555-timer when it is wired in astable mode. This is also called an astable multivibrator. In this mode, the 555-timer circuit creates a stream of regular pulses. The wiring diagram for the 555-timer in astable mode is shown in Figure D-1.

![Figure D-1](image1)

**Figure D-1.**

*Inside the 555-Timer:* In order to understand how the 555-timer can create this regular stream of pulses, we need to look inside and see how it functions. As you can see in Figure D-2, the inside of the device contains many of the components we have studied in experiments 6 and 7.

![Figure D-2](image2)

**Figure D-2.**

First note that there is a voltage divider along the left side of the diagram. This divides a DC source voltage at Vcc into three equal voltages. Therefore, P1 is equal to \((2/3)Vcc\) and P2 is equal to \((1/3)Vcc\). Next to the voltage divider, there are two comparators. The Threshold Comparator compares the voltage at pin 6 (the Threshold) to the voltage at P1. Since the Threshold is at the non-inverting input, the comparator will saturate high when the Threshold exceeds P1. The Trigger Comparator compares the voltage at pin 2 (the Trigger) to the voltage at P2. Since the Trigger is at the inverting input, the comparator will saturate high when the Trigger dips below P2. The outputs of these two comparators are used to control a flip flop. (You may also see an “RS” flip flop here, but we have used a JK flip flop because that is the one you are familiar with.) When the Threshold comparator outputs a high signal, the K input is high and the output of the flip flop goes low. When the Trigger comparator outputs a high signal, the J input is high.

K.A. Connor, S. Bonner, P. Schoch

Rensselaer Polytechnic Institute

Troy, New York, USA

Revised: 29 March 2018
and the output of the flip flop goes high. The output of the flip flop is attached to the 555-timer chip’s pin 3 (Output). The 555-timer chip has one more feature, a transistor switch. This switch will be off when the Output pin is high and the signal at pin 7 (Discharge) will not be influenced by the switch. When the Output pin 3 is low, however, the transistor switch is closed. This forces the Discharge pin to ground.

The 555-Timer in Astable mode: When we wire the 555-timer in astable mode, we create a circuit that generates a string of pulses with the same period and duty cycle. The nature of these pulses is determined by the values of the R1-R2-C1 combination on the outside of the timer in the diagram for Basic Astable Mode on the previous page. We have seen that when current flows through a series combination of a resistor R and a capacitor C, that the circuit responds with a characteristic time constant \( \tau = RC \). The on-time for each pulse is determined by how fast the capacitor C1 charges when the transistor switch is open and the output of the timer chip at pin 3 is high. In this case, the capacitor is attached to the source voltage through the resistors R1 and R2. The charging time constant is \( \tau_{charge} = C1(R1+R2) \). When the capacitor has charged up to \( (2/3)Vcc \), the Threshold Comparator saturates high, the flip flop switches, the output goes low and the transistor switch closes. The off-time for each pulse is determined by how fast the capacitor discharges to ground through the transistor. The discharge path is through R2 to pin 7 to ground, so the discharging rate is \( \tau_{discharge} = C1(R2) \). When the capacitor has discharged down to \( (1/3)Vcc \), the Trigger Comparator saturates high, the flip flop switches, the output goes high, the transistor opens, and the capacitor is no longer attached to ground at pin 7. The capacitor begins to charge again and the cycle repeats.

By selecting just the right values for the resistors and capacitors in this circuit, we can make the voltage at pin 3 (the OUTPUT) go from zero to \( Vcc \) at whatever rate we desire. We can also control the percentage of time that the output will be on relative to the length of an entire cycle. The equations that govern this behavior are:

\[
T_{ON} = 0.693(R1 + R2)C1 \\
T_{OFF} = 0.693(R2)C1 \\
f = \frac{1}{T_{ON} + T_{OFF}} = \frac{1.44}{(R1 + 2R2)C1}
\]

**Pulse width modulation:** One of the most important things we can use 555-timers for is to control and drive a large variety of systems with pulse width modulation. Please read over the links on motor control and flow valve control on the course links page. The power of pulse width modulation comes from its simplicity. Rather than controlling the flow of some liquid by carefully opening a valve part way, you can alternately open and close the valve fully in such a manner that the average open time produces the same effect as a partially open valve. In effect, the rate of flow is controlled by the duty cycle of the controlling voltage. It is much easier to fully open or close a valve than to precisely open it part way. One can also apply power to a motor in this manner to control the speed of rotation. The key goal of this modulation process is to achieve a desired average value for some process. The range of possibilities is shown in Figure D-3 where A has a high duty cycle (fast) and C a low duty cycle (slow).
Experiment Simulation of a 555-timer circuit

In this part of the experiment, we will use PSpice to demonstrate the operation of the 555-timer chip in astable mode. (In all simulations, use the 555 timer from the EVAL library.)

- Wire the circuit in Figure D-4 in PSpice.

![Figure D-4](image)

- Run the simulation.
  - Perform a transient analysis in increments of 2us up to 5ms.
  - Plot the threshold/trigger, discharge and output voltages. The trigger voltage is pin 2 and the threshold voltage is pin 6. (They are tied together.) The discharge voltage is pin 7, and the output is taken at pin 3.
  - Include a copy of the plot in your report.
  - Verify that the timer output changes according to the rules listed for the 555-timer in astable mode. Use the plot to find the time period that the output is ON and the time period that the output is OFF. Note: Do not use the first cycle of pulses produced by the timer circuit. It takes one cycle to settle in to its steady-state
One of these times should be equal to $0.693(R1+R2)C1$ while the other should be equal to $0.693(R2)C1$. Which is which? What is the total period of this output?

- Which of the three signals on your plot corresponds to the charging and discharging of the capacitor, $C1$? To what voltage does it charge each time? To what voltage does it discharge? What is the rate of charge? Is the rate of discharge the same?

Determine the average voltage of the signal.
- Change the end time for the transient analysis to 60ms.
- Display only the output voltage (pin 3) on your plot.
- Rerun the simulation.
- Add a trace of the average of the output using the average function, AVG(). This should add a trace of the time average of the output voltage. The average at any instant of time is the average of the voltage from time = 0 to the time of interest. Thus, you should see that the average asymptotically approaches a particular value.
- Copy this plot and write the approximate voltage the average is approaching on the plot.
- Include this plot in your report.

Find a larger and smaller average voltage for your circuit.
- Find an expression for the duty cycle of an astable 555-timer circuit using the equations given. Consider what relative values of $R1$ and $R2$ would produce the highest duty cycle and what relative values of $R1$ and $R2$ would produce the smallest duty cycle.
- Now, using any combination for 3kΩ, 10kΩ or 30kΩ resistors (only one of each value) for $R1$ and $R2$, find the combination of two resistors which results in the largest average voltage and the combination of two resistors which results in the smallest average voltage.
- Copy the plot for each of these two cases, write the values for $R1$ and $R2$ you used on each plot. Include these two plots in your report.
- Verify in each case that the pulses produced by the multivibrator circuit obey the design rules. If the simulation does not work, the design rules are probably violated.

Build the 555-timer circuit on your protoboard
In this part of the experiment, we will build the astable multivibrator and then use it as the clock for your timer circuit.

- Wire the astable multivibrator shown in Figure D-5 on your protoboard.

- Record your results.
  - You will not be able to see the LED flash because the period of your circuit is too fast.
  - Take an Analog Discovery picture of your output.
Copy this plot and include it in your report.

What is the period of your output signal? What are the off-time and on-time? Use the equations to calculate what these values should be. How do they compare?

Slow down the pulses so that you can observe them with the LED. Do not change components with power applied to the circuit. Turn power off before you make the changes and then turn it back on.

Keeping the resistors R1 and R2 the same, determine a new value for C1 such that the period of the timer will be around 1 second.

Replace C1 in your circuit and observe the LED. Does it flash once a second?

What is your on-time and off-time now? How are these related to the on- and off-times of the original circuit? Why does this relationship hold?

Use the 555-timer circuit as the clock for your counter.

Remove the function generator from pin 1 of the counter.

Connect the output at pin 3 of the 555-timer circuit to pin 1 of your counter circuit.

Does the counter count the 555-timer pulses?

Can you see the lower order bits change now?
Report and Conclusions

The following should be included in your experimental checklist. Everything should be labeled and easy to find. Credit will be deducted for poor labeling or unclear presentation. ALL PLOTS SHOULD INDICATE WHICH TRACE CORRESPONDS TO THE SIGNAL AT WHICH POINT AND ALL KEY FEATURES SHOULD BE LABELED.

Hand-Drawn Circuit Diagrams for all circuits that are to be analyzed using PSpice or physically built and characterized using your Analog Discovery board.

Truth Tables: Note for all truth tables generated, you should include a screen capture of the Static I/O configuration for at least one of the input conditions (one of the rows of the table).

Part A (12 points)

Include the following:
1. NOR gate truth table and Analog Discovery screen capture of a single input/output configuration. (2 pt)
2. NAND gate truth table and Analog Discovery screen capture of a single input/output configuration. (2 pt)
3. NOT gate truth table and Analog Discovery screen capture of a single input/output configuration. (2 pt)
4. PSpice timing diagram for the three gates in the circuit with output traces marked. (2 pt)

Answer the following questions:
1. What are the actual voltage values you observed as HIGH and LOW states in the hardware realization of the circuit? (2 pt)
2. How do the truth tables generated using the actual chips correspond to the truth tables you generated using your PSpice output? (1 pt)
3. If you had a gate with four inputs, how many cases would you have to consider to create its truth table? (1 pt)

Part B (10 points)

Include the following:
1. Flip flop truth table and Analog Discovery screen capture when inputs J = K = 1. (2 pt)

Answer the following questions:
1. Flip flops are called memory devices. Why do you think this is true? (2 pt)
2. Show that the flip flop is giving the correct output at the clock cycles (A, B, C and D) indicated on the timing diagram in Figure S-1. DSTM2 is the clock signal, DSTM1 is J, and DSTM3 is K. Show how the truth table you found for the actual flip flop is consistent with the timing diagram at those four points. (6 pt)

Part C (14 points)

Include the following plots (plots 2-5 are best combined in the same figure):
1. PSpice timing diagram of counters. (2 pt)
2. Analog Discovery plot of clock and QA. (2 pt)
3. Analog Discovery plot of clock and QB. (2 pt)
4. Analog Discovery plot of clock and QC. (2 pt)
5. Analog Discovery plot of clock and QD. (2 pt)

Answer the following questions:
1. For the counter circuit configuration just studied, what is the highest number it counts to in the time shown on your output? Express as both a binary and decimal number. (2 pt)
2. How many pulses will the clock have to cycle through after it resets before the counter hits its maximum value? (1 pt)
3. Which output of the timer (QA, QB, QC, QD) correspond to the bits in the binary number (b3 b2 b1 b0):
   \[ N = b3 \times 2^3 + b2 \times 2^2 + b1 \times 2^1 + b0 \times 2^0. \] (1 pt)

Part D (36 points)

Include following plots:
1. PSpice plot for the astable circuit with R1 = 10k and R2 = 10k. (2 pt)
2. PSpice plots for the average voltage of the astable circuit with R1 = R2 = 10k. (1 pt)
3. PSpice plot for the average voltage of the astable circuit with highest duty cycle. (1 pt)
4. PSpice plot for the average voltage of the astable circuit with lowest duty cycle. (1 pt)
5. Analog Discovery plot of output from 555-timer circuit when R1 = 1.8k, R2 = 6.8k and C1 = 0.1\( \mu \)F. (4 pt)

Answer following questions:
1. What are the on-time, the off-time, and the period of the signal in plot 1.? What are the calculated values for these? Are they consistent? (4 pt)
2. What are the maximum and minimum values for the voltage across the capacitor C1 (at pins 2 and 6)? (Ignore the voltage at times before it reaches steady state.) Why do these values make sense? (3 pt)
3. Calculate the value of \( \tau \) (the decay constant) that controls the rate at which the capacitor C1 charges. Calculate the value of \( \tau \) (the decay constant) that controls the rate at which the capacitor C1 discharges. (4 pt)
4. What is the minimum duty cycle that can be obtained from the astable multivibrator we modeled using PSpice? Can you show this mathematically using Duty Cycle = \( T1/(T1+T2) \)? (3 pt)
5. What was the average voltage for your original circuit? What were the minimum and maximum average voltages when you considered different combinations of R1 and R2? (3 pt)
6. What are the on-time, the off-time, and the period of the signal in plot 5.? What are the calculated values for these? Are they consistent? (4 pt)
7. How did you find the value for C1 that gave the circuit you built a one second period? What value did you find? (2 pt)
8. What are the calculated on-time, off-time and period values for your circuit with the new capacitor? How do these relate to the initial values? Why? (4 pt)

Organization and responsibilities (8 points)

1. Discuss mistakes and problems. (6 pt)
2. List member responsibilities (see below). (2 pt)
List group member responsibilities. Note that this is a list of responsibilities, not a list of what each partner did. It is very important that you divide the responsibility for each aspect of the experiment so that it is clear who will make sure that it is completed. Responsibilities include, but are not limited to, reading the full write up before the first class; collecting all information and writing the report; building circuits and collecting data (i.e. doing the experiment); setting up and running the simulations; comparing the theory, experiment and simulation to develop the practical model of whatever system is being addressed, etc.

Summary/Overview (0 to -10 pts) There are two parts to this section, both of which require revisiting everything done on this experiment and addressing broad issues. Grading for this section works a bit differently in that the overall report grade will be reduced if the responses are not satisfactory.

1. Application: Identify at least one application of the content addressed in this experiment. That is, find an engineered system, device, process that is based, at least in part, on what you have learned. You must identify the fundamental system and then describe at least one practical application.

2. Engineering Design Process: Describe the fundamental math and science (ideal) picture of the system, device, and process you address in part 1 and the key information you obtained from experiment and simulation. Compare and contrast the results from each of the task areas (math and science, experiment, simulation) and then generate one or two conclusions for the practical application. That is, how does the practical system model differ from the original ideal? Be specific and quantitative. For example, all systems work as specified in a limited operating range. Be sure to define this range.

Total: 80 points for experiment packet
0 to -10 points for Summary/Overview
20 points for attendance
100 points

Attendance (20 possible points)
2 classes (20 points), 1 class (10 points), 0 class (0 points)
Minus 5 points for each late.
No attendance at all = No grade for this experiment.
Experiment 7
Section: ______
Report Grade: ______

________________________________________ Name
________________________________________ Name

Checklist w/ Signatures for Main Concepts
For all plots that require a signature below, you must explain to the TA or instructor:

- the purpose of the data (using your hand-drawn circuit diagram),
- what information is contained in the plot and
- why you believe that the plot is correct.
Any member of your group can be asked for the explanation.

PART A: Basic Logic Gates
1. NOR gate truth table & screen capture of an I/O configuration
2. NAND gate truth table & screen capture of an I/O configuration
3. NOT gate truth table & screen capture of an I/O configuration
4. PSpice timing diagram for the three gates in the circuit with output traces marked
Questions 1-3

PART B: Flip Flops
1. Flip flop truth table & screen capture for J = K = 1
Questions 1-2

PART C: Counters
1. PSpice timing diagram of counters
2. Analog Discovery plot of clock and QA
3. Analog Discovery plot of clock and QB
4. Analog Discovery plot of clock and QC
5. Analog Discovery plot of clock and QD
Questions 1-3

PART D: 555-Timer
1. PSpice plot astable circuit with R1 = 10k, R2 = 10k
2. PSpice plots average voltage astable circuit with R1 = R2 = 10k
3. PSpice plots average voltage astable circuit with highest duty cycle
4. PSpice plots average voltage astable circuit with lowest duty cycle
5. Analog Discovery plot output from 555-timer circuit R1 = 1.8k, R2 = 6.8k and C1 = 0.1μF
Question 1-8

Member Responsibilities
Summary/Overview
K.A. Connor, S. Bonner, P. Schoch

Rensselaer Polytechnic Institute

Revised: 29 March 2018
Troy, New York, USA