Experiment 7 - Digital Logic Devices and the 555-Timer

Project 3 requires both a counter and a 555 circuit, keep them wired after you complete this experiment.

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 *LTspice*
- 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, Digital IO in Scopy.
- Simulate basic combinational and sequential logic gate configurations, generating a timing diagram with *LTspice*
- Characterize the operation of a JK Flip-Flop both experimentally and using *LTspice* simulation
- Characterize the operation of a Binary Counter both experimentally and using *LTspice* simulation
- Characterize the operation of a 555 Timer in Astable Multivibrator configuration both experimentally and using *LTspice* simulation.
- Use a counter to count the pulses produced by a 555 Timer Astable Multivibrator.

Equipment Required:

- **M2k/Analog Discovery** (with Scopy/Waveforms Software)
- **Voltmeter** (DMM or M2k/Analog Discovery)
- **Oscilloscope** (M2k/Analog Discovery)
- **Function Generator** (M2k/Analog Discovery)
- +5V (+Vcc) Power Supply (M2k/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

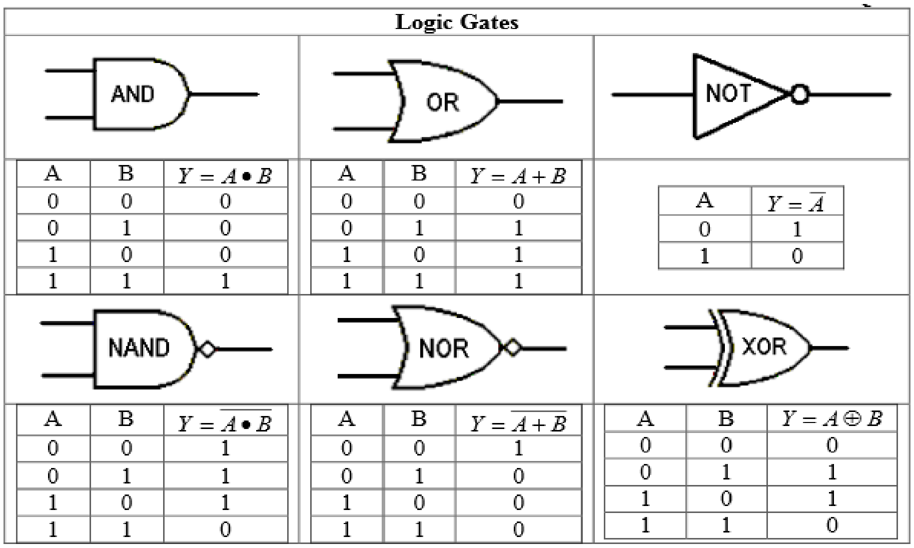
Required Reading: Before beginning the lab, at least one team member must read over and be generally acquainted with this document and the other **required reading** materials listed under Experiment 7 on the EILinks page.

Hand-Drawn Circuit Diagrams: Before beginning the lab, hand-drawn circuit diagrams must be prepared for all circuits physically built and characterized using your M2k/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.


Logic Gates
AND
OR
NOT
NAND
NOR
XOR

A	B	$Y = A \bullet B$
0	0	0
0	1	0
1	0	0
1	1	1

A	B	$Y = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

A	$Y = \overline{A}$
0	1
1	0

A	B	$Y = \overline{A \bullet B}$
0	0	1
0	1	1
1	0	1
1	1	0

A	B	$Y = \overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0

A	B	$Y = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Figure A-1. (From Quiz 3 Crib Sheet)

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.) When viewed with pin 1 at the upper left, the upper right pin on a digital chip is typically 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 the power needed for the operations the chips perform. Generally circuit diagrams do not show these two power connections. ***If you forget to connect these two pins, your circuit will not function. The chip doesn't work for free.***

Timing diagrams: Timing diagrams are a special kind of transient output which are useful for viewing many binary signals. Unfortunately LTspice doesn't have a simple way to do timing diagrams. A way to simulate them in LTspice is to set up logic gates that have inputs and output of 0V and 1V, the 0 and 1 can be viewed as logic level rather than the actual voltages. 0 and 1V are the default values for LTspice, see A-5

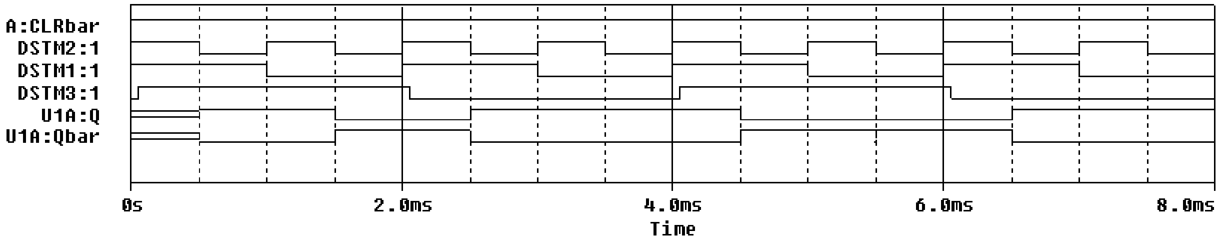


Figure A-2. Typical timing diagram, LTspice can mimic this, see figure A-5

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. *View the video on LMS about digital I/O for the M2k and Analog Discovery. Notes on Part A, Exp 7.*

- Wire the circuits in Figure A-3.1 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/M2k, is also shown.

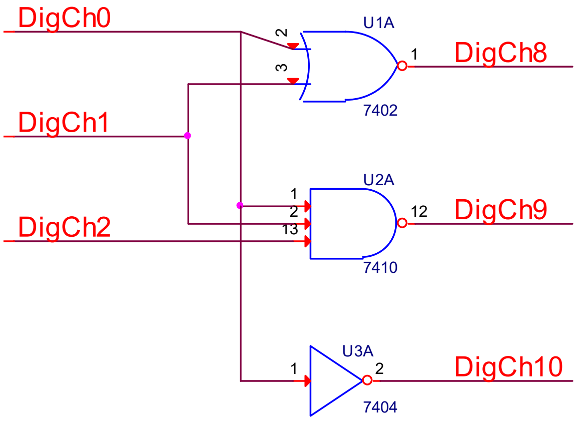


Figure A-3.1

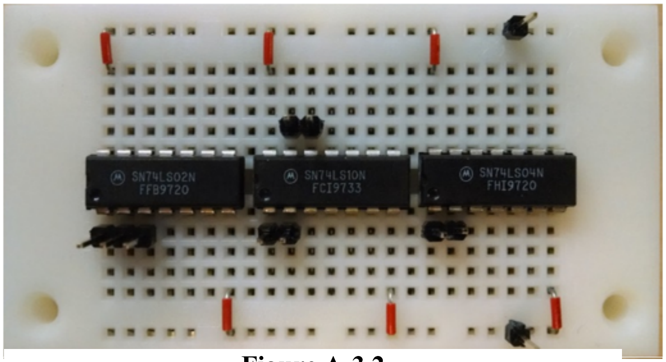


Figure A-3.2

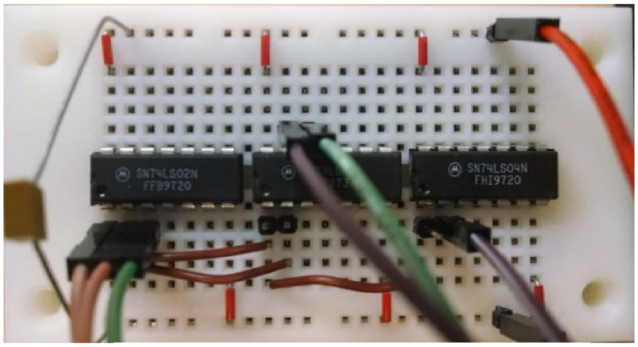


Figure A-3.3

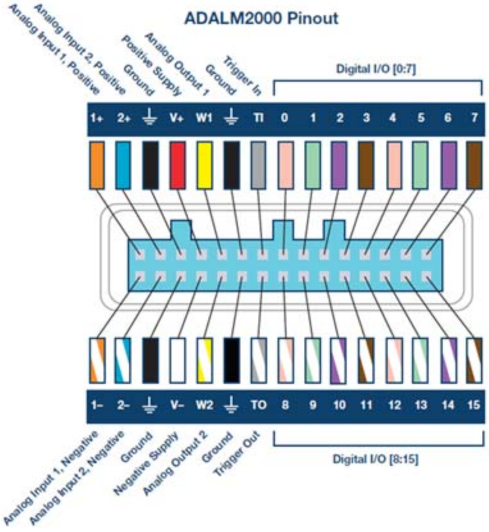


Figure A-3.4 Pin out of board

For the first measurements, we use the *Digital IO of the M2k* or the *Digital Static I/O feature of Analog Discovery*.
For the Analog Discovery: 1) be sure that all 16 channels are configured Bit I/O (menus on the left). 2) channels 8-15 are LED (for measurements, the LED is on when the voltage is HIGH and off when it is LOW) and 3) 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). 4) Enable the Static IO
For the M2k: 1) channels 0, 1, and 2 are "out" (output to circuit). 2) Channels 8, 9, and 10 are "in", input from circuit. And 3) enable the Digital IO.
For either board we will use the outputs from Digital Channels 0, 1, 2 to provide inputs to the gates and inputs to 8, 9, and 10 to measure the outputs from the gates.

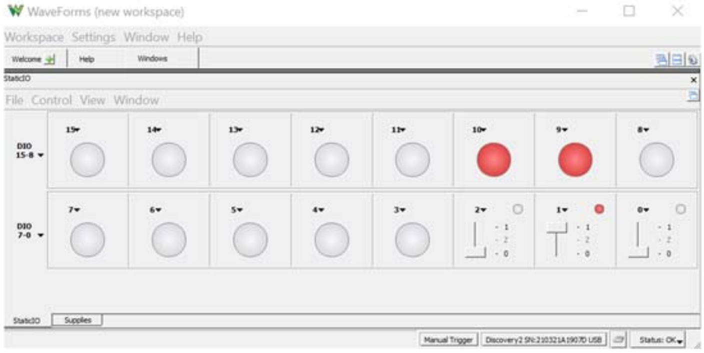


Figure A-4a Waveforms (Analog Discovery)

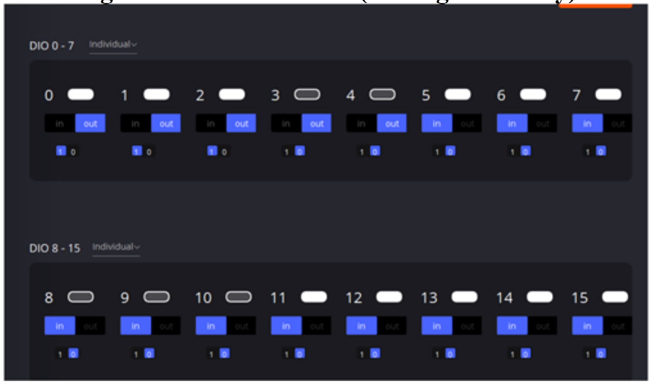


Figure A-4b Scopy (M2k)

- 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 of Scopy/Waveforms 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 of Scopy/Waveforms 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 shot 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 *LTspice*.

- Create the following circuit (Figure A-5) in *LTspice*. Note that the nodes have been labelled. Using this feature makes it possible to identify the voltages plotted.
- Use the "or" gate for the NOR but connect to the circle (inverting) output. OR followed by NOT = NOR
- Use the "and" for the NAND and also connect to the circle (inverting) output. An inverted AND = NAND

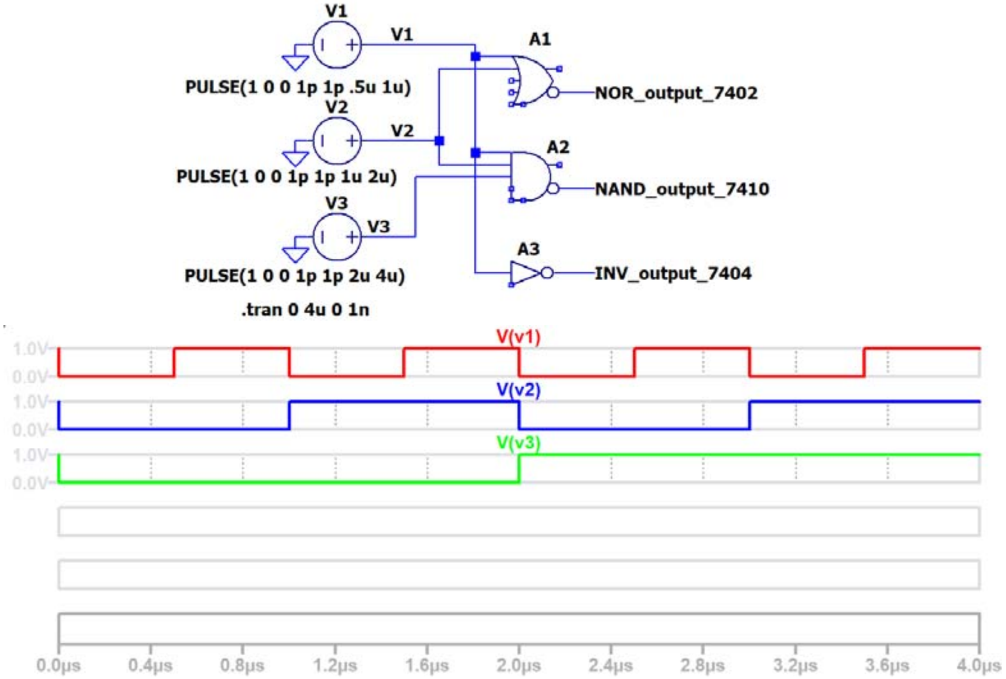


Figure A-5. Simulate the gates circuits

- Wiring the circuit in *LTspice* is somewhat different than on the protoboard.
  - *LTspice* assumes power and ground are connected.
  - **The logic gates in *LTspice* have default values of 0V and 1V, so these can be used to represent logic level of 0 and logic level of 1. 1V represents a logic true or high. 0V represents logic false or low.**
  - PULSE voltage sources are used to present all possible input combinations.
    - V1 – Vinitial=1, Von=0, Trise=1p, Tfall=1p, Ton=0.5u, Tperiod=1u
    - V2 – Vinitial=1, Von=0, Trise=1p, Tfall=1p, Ton=1u, Tperiod=2u
    - V3 – Vinitial=1, Von=0, Trise=1p, Tfall=1p, Ton=2u, Tperiod=4u
  - Run a transient analysis for 4u
  - **It is impossible to see 6 traces on 1 plot so right click in the plot plane, Add Plot Plane, do this to have a total of 6 plots. Plot only one signal in one plot. Plot V1, V2, V3, NOR_output_7402, NAND_output_7410 and INV_output_7404.**
  - **Include the schematic and the 6 plots in your report. The 3 voltage traces are shown in Figure A-5 but not the outputs.** The unused axis color was lightened figured A-5, you don't need to do this.
  - Confirm that your LTspice results agree with the experimental results.

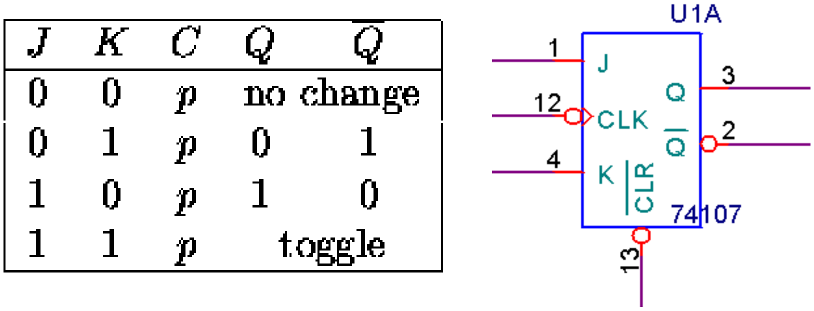
Summary

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. There are several different kinds of flip flops with slightly different characteristics. In this course we use the JK flip flop. A symbol for a JK flip flop is pictured in Figure B-1. JK flip flops have four inputs, two outputs and the usual two power connections (V_{CC} and ground). The outputs are labeled Q and $\overline{Q}$ (also called Qbar and NQ). They are complements of one another. Thus, when Q is LOW, Qbar is HIGH, etc. CLK is a digital clock input. 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 $\overline{CLR}$, when LOW, will reset the outputs to a known state. It has the following truth table:



J	K	C	Q	$\overline{Q}$
0	0	p	no change	
0	1	p	0	1
1	0	p	1	0
1	1	p	toggle	

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 set/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 principal. With every clock cycle, it looks at its inputs and changes state accordingly.

The flip flop is a memory device. If both the J and K inputs are zero, its output will remain the same indefinitely. A bank of 8 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 J and K to 1.

Timing diagram for a flip flop: LTspice doesn't have a JK flip flop model. To illustrate the behavior of a JK flip flop, we wired the circuit in Figure B-2 in a different pspice simulator and created the timing diagram shown in Figure B-3.

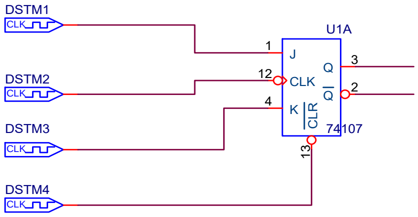


Figure B-2.

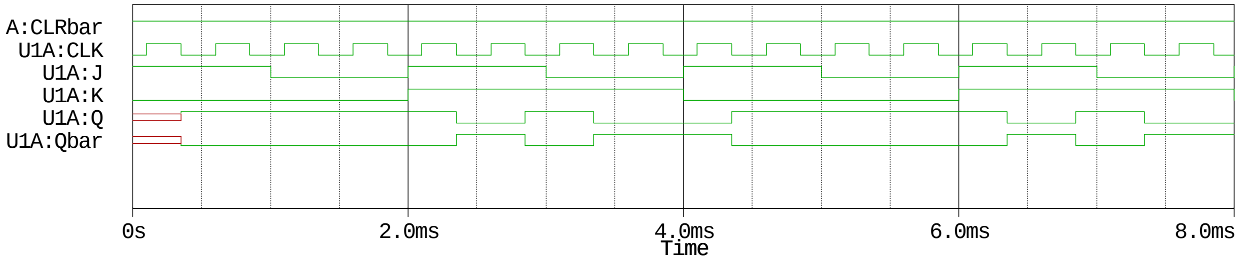


Figure B-3. Red traces indicate unknown state, it depends a past history.

U1A:CLK is the clock input. Each time the clock falls, look at the values of the inputs at J (UA1:J), K (UA1:K), 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. The A:CLRbar is held high for this run. A low on this input clears Q.

In Figure B-3 an 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 J and K so that they do not change state at the same time the clock transitions from high to low. If we had the simulation produces errors because it cannot properly determine the output as J and K inputs are changing.

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.

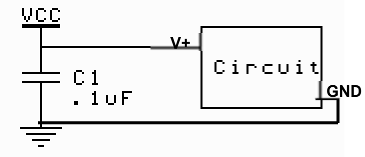


Figure B-4.

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.

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, Fig B-1.
  - 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 (Waveforms) or "out" (Scopy) 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 (Waveforms) or "in" (scopy) to sense the outputs from the flip flop.
  - For a short period of time set the CLR to zero and then keep it high. This forces the flip flop to start in a known state, with Q low and Qbar high. This can be done by moving a wire to ground and moving and leaving it on +5V or by using yet another output channel.
  - 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.

Q (before pulse)	J	K	Q (after pulse)	Qbar (after pulse)
0	0	1		
1	0	1		
0	1	0		
1	1	0		
0	0	0		
1	0	0		
0	1	1		
1	1	1		

  - Take a screen shot 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 shot 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

Binary Counters: Flip flops can be connected together to form a binary counter. While it can be done using JK flip flops it is simpler to use D flip flops. D flip flops are also called D latches, when a rising clock pulse occurs the output Q will be set equal to logic level of the D input. That value is latched or held until the next clock pulse. These can be used for binary counting by using the circuit shown in Figure C-1. The output of each flip flop is used as the input clock to the next flip flop. The D input is set to the opposite of the present output by using Qbar. This means that the D flip flop will toggle on each pulse of their clocks. The first flip flop toggles rising edge of a clock cycle. The output will complete one cycle with 2 rising edges of the clock. This causes a full cycle at the output at a rate equal to half of the clock rate. By the same reasoning, the second flip flop cycles at a rate of ½ of the 1st flip flop or ¼ of the clock rate. The D flip flop gates shown also have the ability to be cleared using the CLR pin or preset high using the PRE.

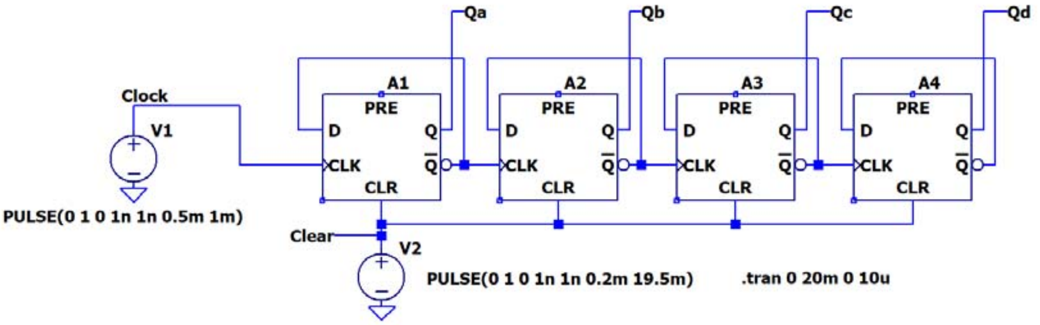
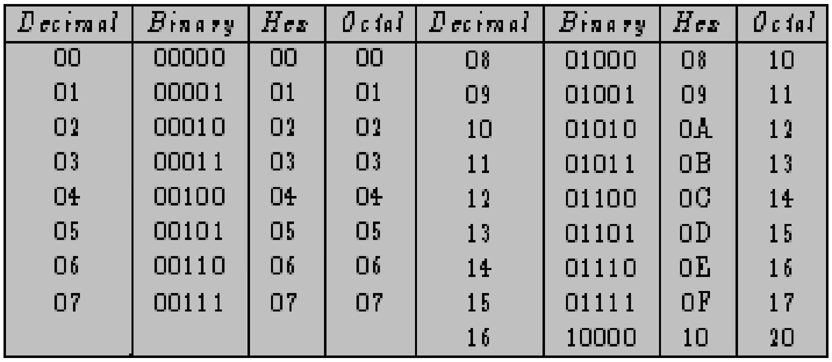


Figure C-1. In LTspice the D flip flp is called dflip

If we attach a bit of a binary number, (b0,b1,b2), to the output of each flip flop (b0 to the first, b1 to the second, and b2 to the last). We get a pattern out which corresponds to the binary counting shown in Figure C-2 below:



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

Figure C-2.

The lowest order bit of the binary numbers toggles from 0 to 1: (0-1-0-1-0-1….). The second order bit also toggles between 1 and 0, but at half the rate: (0-0-1-1-0-0-1-1-….). The third order bit toggles also, but at half the rate of the previous bit: (0-0-0-0-1-1-1-1-0-0-….). 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 binary counters each with 4 bits.

Experiment - *Simulation of a binary counter*

In this part of the experiment, we will use *LTspice* to create the circuit in C-1. This will simulation one of the 4 bit binary counters available in the 74393 logic chip. More gates could be added to allow counting to greater numbers.

- Simulate the circuit in Figure C-1 in *LTspice*. The gate used is called *dflip*. It is in the digital library.
- As was done in Part A, figure A-5, you will use the *TLspice* default logic levels of 0V and 1V. A 0V signal will be considered a logic level of 0. A 1V signal will be a logic 1. The real circuits work at higher voltages but using 0 and 1 allows you to think of them as logic levels rather than voltage levels.
- 2 PULSE voltage sources are used.
    - V1 is the clock with Vinitial=0, Von=1, Trise=1n, Tfall=1n, Ton=0.5m and Tperiod=1m
    - V2 is a reset pulse. Set Vinitial=0, Von=1, Trise=1n, Tfall=1n, Ton=0.2m and Tperiod=19.5m
- Run the simulation for 20ms with a step size of 10us.
- Set up 6 plot planes as was also done for Part A, figure A-5.
    - Put one signal per plot.
    - Plot the reset signal, the clock, Qa, Qb, Qc, and Qd
- Using the voltage traces verify that the D flip flops are actually counting. At any given time you should be able to get a binary number by looking at signals Qd, Qc, Qb, and Qa in that order.
    - What is the maximum count that is achieved? Express this as both a binary number and a decimal number.
    - What is the count value just before the second reset pulse? At what time does it reset? Write it on the timing diagram (voltage traces) for the circuit.
    - If there were 8 gates rather than 4 what would be the maximum count obtainable? Assume that there aren't any reset pulses after the initial one at t=0.
    - <u>Include the voltage traces diagram in your report.</u>

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-3 on your protoboard. **Keep this together, you use it again in Part D.**

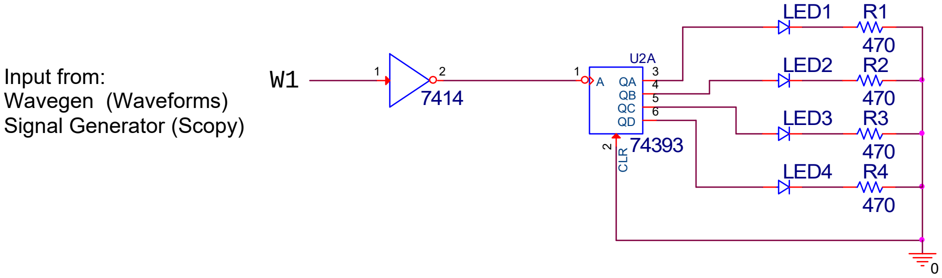


Figure C-3. The Schmitt trigger inverter cleans up the signal from instrumentation board. Don't forget that the 74LS14 and the 74LS393 both need power and ground. Use the colored LEDs. You might have 2 clear ones but one emits in the IR range and the other is a photo transistor.

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 it might be an IR LED or 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 (470 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 470 Ohms.

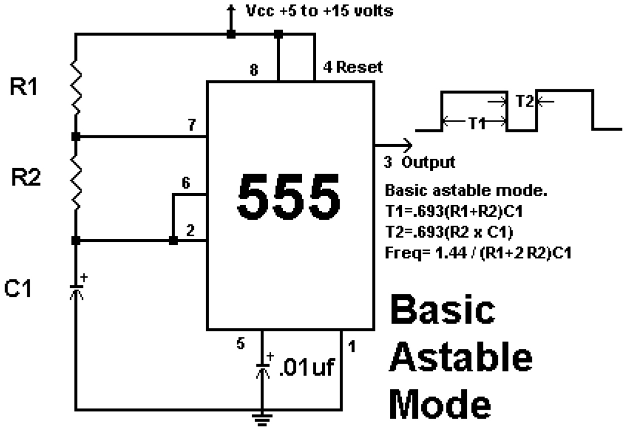
  - Use the Function Generator (Analog Discovery) or Signal Generator (M2k) for the clock. Set it for a square wave with a frequency of 20Hz. We are using a fairly low frequency so that we will be able to see the switching with the LEDs. *Use the offset feature to shift the square wave up such that it cycles between 0V and 4V. 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 470 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 toward ground.
  - If you cannot find 470 ohm resistors, use 1kΩ (LEDs will be dim) or 100Ω (will cause a voltage drop at the output of the 74393 chip but should work.) It is preferred to use resistors in the 300-500Ω if available.
  - Set the CLR to +Vcc for a momen*t and then connect it to ground.* The connection to Vcc will clear the outputs. Then attaching it to 0V to enable the chip to function. Note that this is the opposite of the JK flip flop but the same as the D flip flop. The little circle on the clear pin of the JK Flip Flop indicated that the clear function is active when a low signal is applied. The clear on the 74393 doesn't have that circle – the clear is active with a high signal. 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 M2k/Analog Discovery. <u>Include them in your report.</u>
    - 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 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.



$$Freq = \frac{1}{T_1 + T_2}$$

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. The SR flip flop is like a JK flip flop but doesn't need a clock.

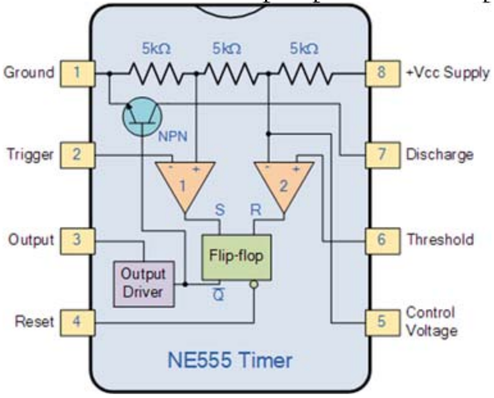


Figure D-2. https://commons.wikimedia.org/wiki/File:Ne555.gif
Attribution: Radinpirouz, CC BY-SA 4.0 <https://creativecommons.org/licenses/by-sa/4.0>, via Wikimedia Commons

Triangles labeled 1 and 2 are voltage comparators. The SR flip flop works much as a JK flip flop but doesn't need a clock. The NPN transistor works as a switch. First note that there is a voltage divider at the top of the diagram. This divides a DC source voltage at Vcc into three equal voltages. Comparator #1 compares the trigger signal to (1/3)Vcc. Comparator #2 compares the Threshold to (2/3)Vcc. When comparator #1 sees the Trigger below (1/3)Vcc it sets the output of the flip flop. When comparator #2 sees the Threshold above (2/3)Vcc it resets the flip flop. The Qbar of the flip flop does 2 things, it controls the NPN transistor as a switch. Qbar high turns on the transistor and shorts the Discharge pin to ground. Qbar low turns of the transistor off so that no current flows into the Discharge pin. Qbar also drives the output through an inverter. When Qbar is high the Output will be low and if Qbar is low the Output is high. When the output is high, the 555 is consider to be in the on state, for low output it is in the off state.

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, Figure D-1. 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 τ = 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 τ_{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 V_{CC} 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:

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

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).

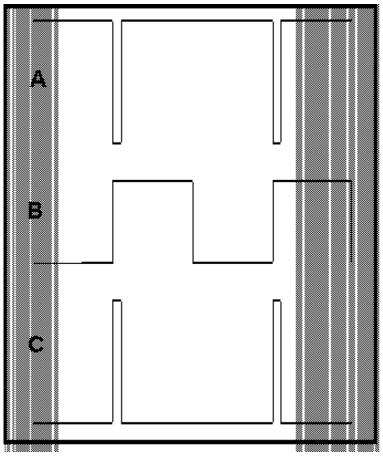


Figure D-3. Top trace has high duty cycle and high average voltage, middle trace is 50%, bottom has low duty cycle and a low average voltage.

Experiment

Simulation of a 555-timer circuit

In this part of the experiment, we will use *LTspice* to demonstrate the operation of the 555-timer chip in astable mode. *(In all simulations, use NE555, it is located in the Misc library.)*

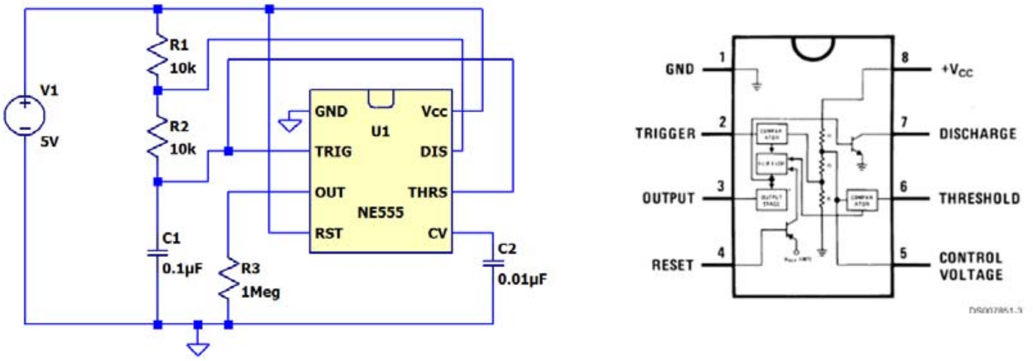


Figure D-4.

- Wire the circuit in Figure D-4 in *LTspice*.

- Run the simulation.
  - Perform a transient analysis for 5ms with a maximum step size of 2us.
  - 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 so plot one only.) 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 condition.* 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 it 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.
  - LTspice can do arithmetic on a waveform. With LTspice open, click on the Help > Help Topics and search on waveform arithmetic. We want to look at the average value of the output signal.
    - The waveform viewer can integrate a trace to obtain the average and RMS value over the displayed region. First zoom the waveform to the region of interest, then move the mouse to the label of the trace, hold down the control key and left mouse click.
    - A box will appear with the average listed. Record the average.
  - Include this 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. <u>Include these two plots in your report and include the measured average value.</u>
- 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.

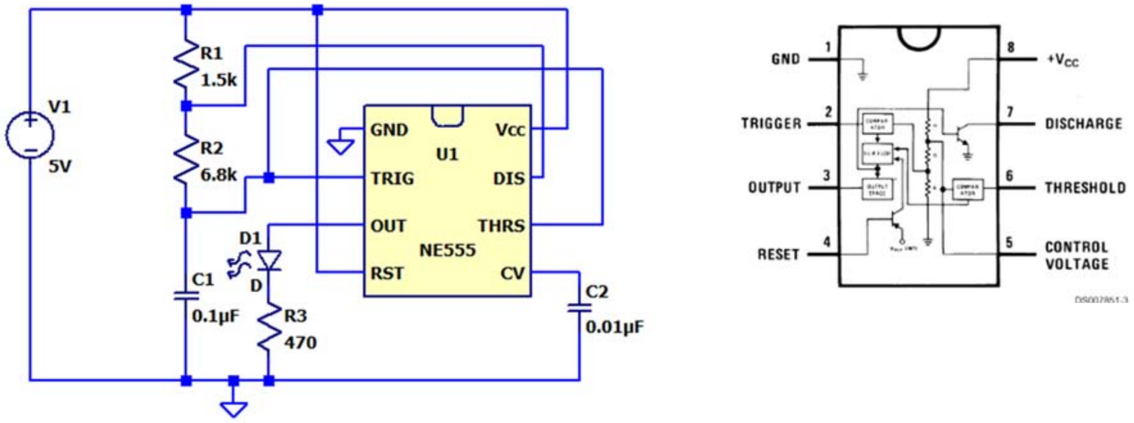


Figure D-5. D1 is an LED

- Record your results.
  - You will not be able to see the LED flash because the period of your circuit is too fast.
  - <u>Take an M2k/Analog Discovery picture of your output.</u>
  - <u>Copy this plot and include it in your report.</u>
  - 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. ***The rule in engineering is that you do not make changes with power applied. Turn power off before you make the changes and then turn it back on. But ---- with our instrumentation it is very difficult to damage anything while doing these experiments. So you can break the rule here but it is better to be in the habit of powering down for any changes.***
  - 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 – Figure C-3.
  - Remove the function generator from pin 1 of the inverter chip.
  - Connect the output at pin 3 of the 555-timer circuit to pin 1 of the inverter.
  - Does the counter count the 555-timer pulses?

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 physically built and characterized using your board.

Truth Tables: Note for all truth tables generated, you should include a screen shot of the Static I/O (Waveforms) or Digital IO (Scopy) 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 M2k/Analog Discovery screen shot of a single input/output configuration. (2 pt)
2. NAND gate truth table and M2k/Analog Discovery screen shot of a single input/output configuration. (2 pt)
3. NOT gate truth table and M2k/Analog Discovery screen shot of a single input/output configuration. (2 pt)
4. *LTspice* timing diagram for the three gates in the circuit with output traces marked. (2 pt)

Answer the following questions:

1. How do the truth tables generated using the actual chips correspond to the truth tables you generated using your *LTspice* output? (2 pt)
2. If you had a gate with four inputs, how many cases would you have to consider to create its truth table? (2 pt)

Part B (10 points)

Include the following:

1. Flip flop truth table and M2k/Analog Discovery screen shot 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)

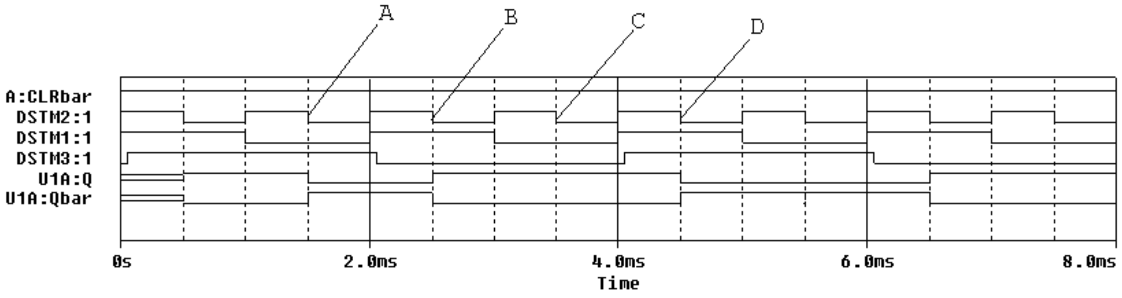


Figure S-1.

Part C (14 points)

Include the following plots (plots 2-5 are best combined in the same figure):

1. *LTspice* timing diagram of counters. (2 pt)
2. M2k/Analog Discovery plot of clock and QA. (2 pt)
3. M2k/Analog Discovery plot of clock and QB. (2 pt)
4. M2k/Analog Discovery plot of clock and QC. (2 pt)
5. M2k/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. What is the count value just before the second reset pulse and at what time does it reset? (1 pt)
3. If 8 gates were used what is the maximum count that can be obtained, assuming no reset pulses? (1 pt)

Part D (36 points)

Include following plots:

1. *LTspice* plots for the astable circuit with R1 = 10k and R2 = 10k. (2 pt)
2. *LTspice* determination of the average Vout voltage for the plot with R1 = R2 = 10k. (1 pt)
3. *LTspice* plot of Vout and state the average voltage of the astable circuit with highest duty cycle. (1 pt)
4. *LTspice* plot of Vout and state the average voltage of the astable circuit with lowest duty cycle. (1 pt)
5. M2k/Analog Discovery plot of output from 555-timer circuit when R1 = 1.5k, R2 = 6.8k and C1 = 0.1μF. (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 τ (the decay constant) that controls the rate at which the capacitor C1 charges. Calculate the value of τ (the decay constant) that controls the rate at which the capacitor C1 discharges. (4 pt)
4. 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)
5. 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)
6. 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)
7. 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)
8. With the NE555 timer driving the 74393 counter, do the LEDs cycle as expected? Explain your answer. (3pts)

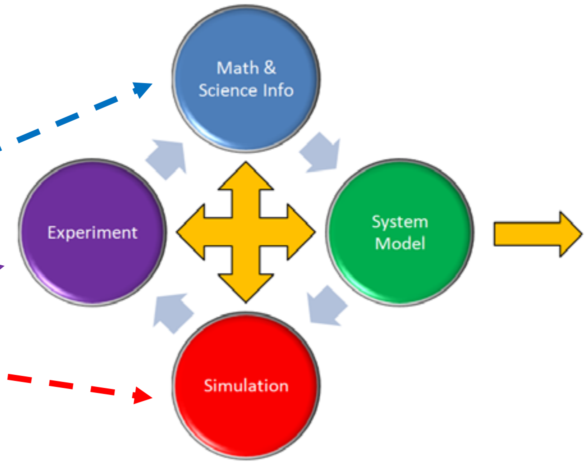
Organization (8 points)

1. Material should be in logical order, easy to follow and complete. (8 pt)

List group member *responsibilities*. (0 to -4pts) 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
<u>20 points for attendance</u>
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 shot of an I/O configuration ____________**
2. **NAND gate truth table & screen shot of an I/O configuration**
3. **NOT gate truth table & screen shot of an I/O configuration**
4. ***LTspice* timing diagram for the three gates in the circuit with output traces marked**

Questions 1-2

PART B: Flip Flops

1. **Flip flop truth table & screen shot for J = K = 1 _______________**

Questions 1-2

PART C: Counters

1. ***LTspice* timing diagram of counters**
2. **M2k/Analog Discovery plot of clock and QA**
3. **M2k/Analog Discovery plot of clock and QB ________________________**
4. **M2k/Analog Discovery plot of clock and QC**
5. **M2k/Analog Discovery plot of clock and QD**

Questions 1-3

PART D: 555-Timer

1. ***LTspice* plot astable circuit with R1 = 10k, R2 = 10k**
2. ***LTspice* plots average voltage astable circuit with R1 = R2 = 10k**
3. ***LTspice* plots average voltage astable circuit with highest duty cycle**
4. ***LTspice* plots average voltage astable circuit with lowest duty cycle**
5. **M2k/Analog Discovery plot output from 555-timer circuit R1 = 1.5k, R2 = 6.8k and C1 = 0.1μF ___________________________**

Question 1-8

Format

Member Responsibilities

Summary/Overview