Lab 11: Digital Logic and Audio Synthesizers

Prelab

Watch the following video as an introduction to boolean logic and logic gates.
www.youtube.com/watch?v=gI-qXk7XojA.

In your textbook, read the introduction to chapter 12 (page 726) and the section "Universal Capability of NAND and NOR Gates" (starting on page 735).

When will the output of an AND gate with two inputs be TRUE (logic 1)? When will it be FALSE (logic 0)?

When will the output of a NAND gate be TRUE (logic 1)? When will it be false (logic 0)?

Skim the beginning of the section in your textbook about 555 timer chips starting on page 686 (10.2 through 10.2.4).

Describe the output of astable and monostable configurations.

Be sure that you or your partner bring the textbook!

Part I: Introduction to Logic

1.1 Logic

The concept of logical operations on boolean values is the basis of how all modern computers operate. Transistors allowed the creation of low power hardware which could implement these logical operations by controlling voltage. Boolean values can only have two states, but these states have various representations.

- TRUE / High / 1 / On / 5 volts / 3.3 volts
- FALSE / Low / 0 / Off / Ground / Zero volts

The term TTL refers to Transistor-Transistor-Logic which is often used to mean logic circuits where TRUE is represented by 5 volts. (Recently more and more logic circuits are using 3.3 volts or even lower to be more power efficient.)

Truth tables are used to represent the relationship between the inputs and outputs of logical operations such as AND, OR, NOT, etc...

For example, the AND logical operation takes in two or more inputs (we'll only be using two) and outputs TRUE only if all inputs are TRUE. Table 1 shows the truth tables for six of the basic logic operations.

In A	In B	NOT A	A AND B	A NAND B	A OR B	A NOR B	A XOR B
0	**0**	1	0	1	0	1	0
0	**1**	1	0	1	1	0	1
1	**0**	0	0	1	1	0	1
1	**1**	0	1	0	1	0	0

Table 1: Truth tables showing the output states for various input states.

1.2 Universal Gates

Certain logic gates can be added together to form any other gate, the one you will be working with is the NAND gate. On page 736 your textbook shows how to make seven different logic gates using only NAND gates or only NOR gates.

Part II: Logic Circuits

2.1 NAND Gate

Figure 1 shows the pinout for an integrated circuit which contains four separate NAND gates.

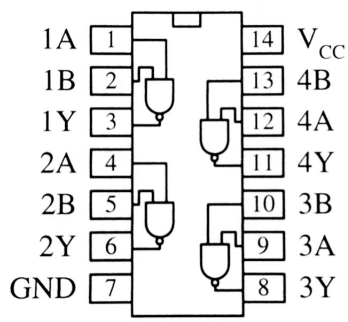


Figure 1: Pinout for a SN74HCT00N Quad NAND Gate

For all of the logic gate circuits, use the bench HP/Agilent power supply set to 5 volts and with current limiting set to 30 mA.

Using red LEDs for the inputs and a blue LED for the output, construct the the circuit shown in figure 2. Confirm that the logic values indicated by the input and output LEDs for all four configurations of input values match with the truth table for a NAND gate.

2.2 Universal Gates

You will now construct four circuits corresponding to the logic gates NOT, AND, OR and NOR using only NAND gates as shown in your textbook on page 736. Figure 3 demonstrates how to wire the AND gate.

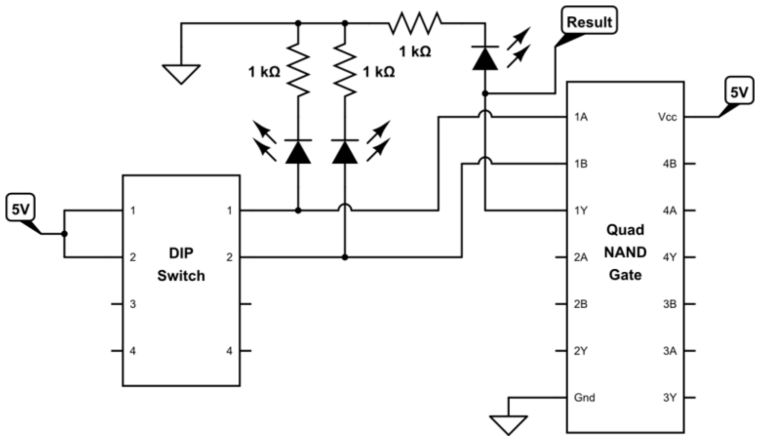


Figure 2: A single NAND gate wired to show both input and output.

Confirm the correctness of each circuit by comparing to the truth table for each logical operator. Take a picture of each circuit in any state which has the final LED turned on.

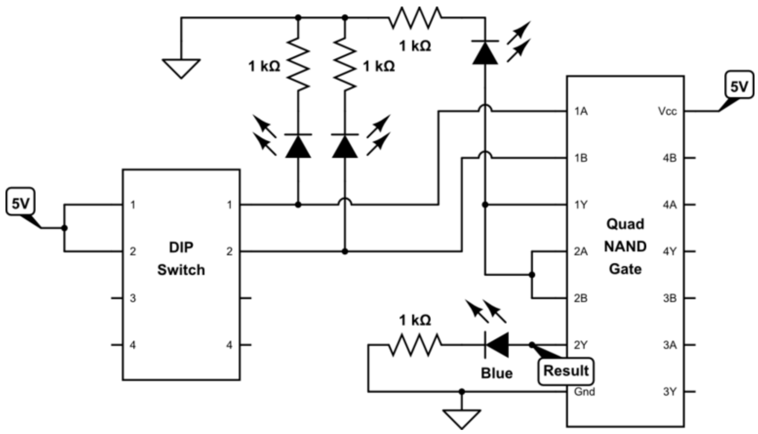


Figure 3: Two NAND gates wired to form an AND gate.

Part III: Audio Synthesizer

3.1 The 555 Timer

The 555 timer integrated circuit is both incredibly simple and incredibly versatile. Fundamentally, it just switches it's output between a HIGH and LOW value, which you should now recognize as the sign of a digital circuit. The usefulness of the appears when it is wired in various configurations and combined with other 555 timers or other ICs such as decade counters.

You'll be using two 555 timers to create a simple audio synthesizer. First you'll build an astable oscillator and then use that to drive a monostable oscillator.

3.2 Astable Mode

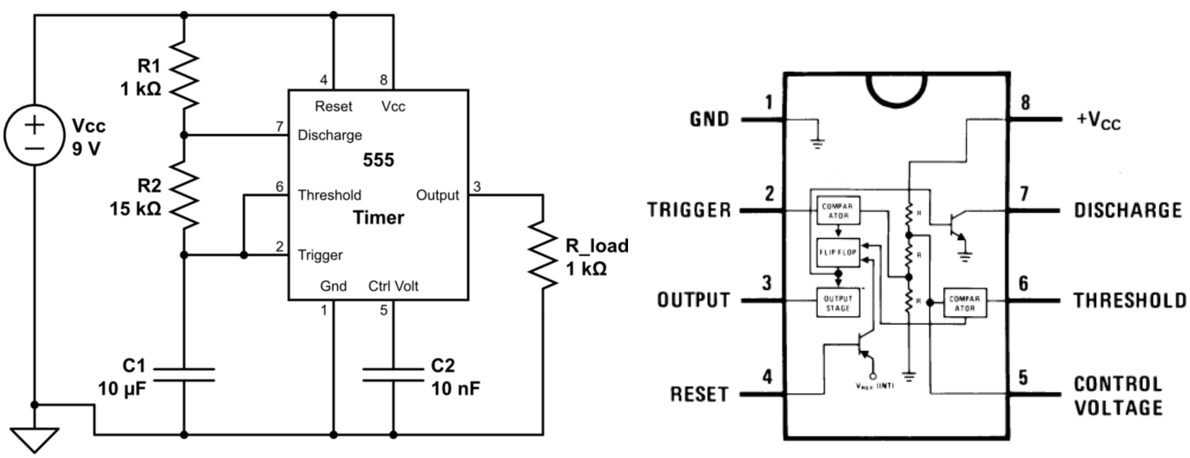
In astable mode the 555 timer output continuously changes between low and high, forming a square wave. (Note that this is a DC square wave since the output never goes negative, unlike the square wave from your function generator.)

The timing of the square wave is controlled by the charging and discharging of the capacitor connected to the trigger and threshold pins. When the tigger voltage goes below (1/3)Vcc the output will go high and the capacitor will charge through (R1+ R2). When the threshold voltage goes above (2/3)Vcc the output will go low and the capacitor will discharge through R2 and the discharge pin which gets internally connected to ground through a transistor.

Time spent high is $t_H = \ln(2) \times C_1 \times (R_1 + R_2)$
Time spent low is $t_L = \ln(2) \times C_1 \times (R_2)$

Given the values in the circuit below, what will t_H and t_L be?

For all the 555 circuits, use the bench (HP/Agilent) power supply with current limiting set to 250 mA.



(a) A 555 timer in astable configuration

(b) Physical pinout for a 555 timer

Figure 4: A 555 oscillator

- Construct the circuit shown in figure 4a. **Note that the pin locations in the circuit diagram don't match the physical pin locations!**
- Connect channel 1 of your scope to the output (pin 3). Connect channel 2 to the Trigger/Threshold voltage. *Overlay the signal on top of each other and take a picture.*

What is the maximum output voltage? What is the difference between that and Vcc? Does that number seem familiar? (If not, try dividing it by two.)

What is the maximum voltage on the capacitor? What is the minimum? Why those values?

- Replace R2 with a 100kΩ potentiometer wired as a variable resistor (use only the middle wire and one side wire).

What minimum and maximum t_L and t_H are you able to achieve?

LEAVE THIS CIRCUIT ON YOUR BOARD.

3.3 Monostable Mode

In monostable mode the 555 timer only goes high when an external trigger signal is applied. It then stays high for a time $t_H = \ln(3) \times R \times C$.

Construct the circuit shown in figure 5. Use the output from the astable circuit to drive the input.

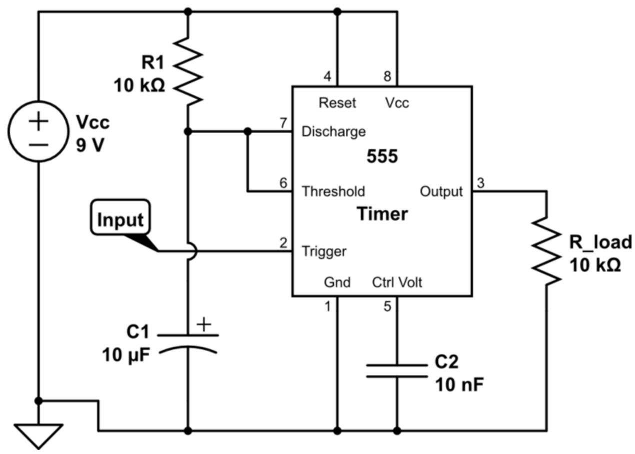


Figure 5: A 555 timer in monostable configuration

Connect the oscilloscope to show the trigger voltage on channel 1 and the output on channel 2.

Describe what happens as you adjust the potentiometer for the astable circuit.

3.4 Atari Punk Console

Connecting the output of the second timer to a speaker creates old school video game type sounds and is known as the "Atari Punk Console."

Figure 6 shows the complete circuit which is composed of the two circuits you just built with a few small changes which are detailed below.

- Change C1 to 10nF and C2 to 30nF (Three 10nF caps in parallel)
- Add in the diode on the first IC so that the capacitor only charges through R1 and only discharges through the pot. This creates very short output pulses with variable spacing.
- Replace the timing resistor on the second 555 with a 100 kΩ pot (using only the center and one edge wire) and add a 100Ω resistor.
- Remove the both load resistors and add in R4, C5 and a speaker (or two speakers in parallel).
- NOT SHOWN: On each chip add a 10nF capacitor between Vcc (pin 8) and ground to stabilize the signal. The connections should be as physically close to pin 8 and pin 1 as possible. This is standard practice for integrated circuits, but rarely shown on diagrams.

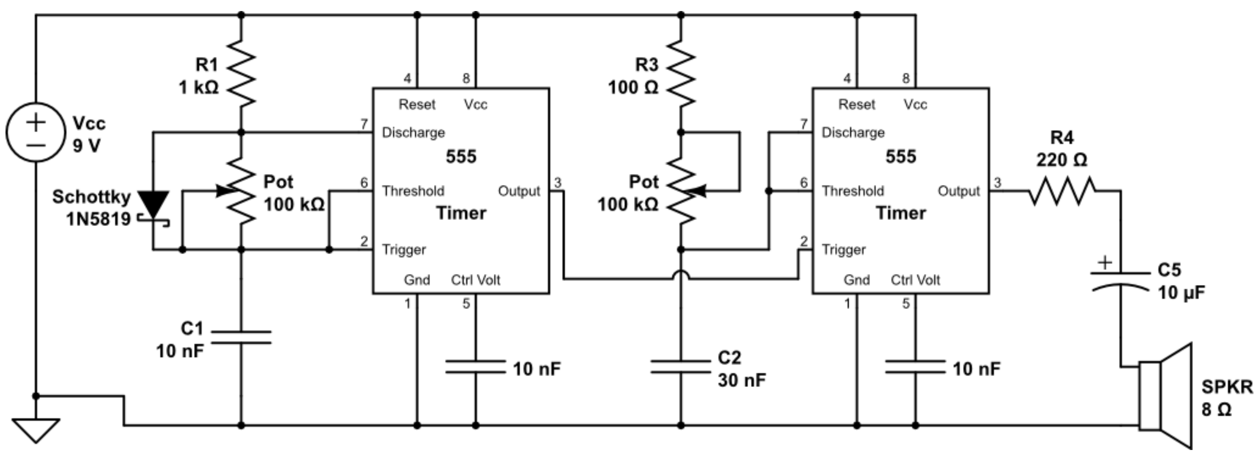


Figure 6: The Atari Punk Console synth (with a few modifications)

- Connect your scope to show the output of each IC and observe how each output changes as you adjust the potentiometers.
- Have fun seeing what sounds you can create.
- *Include a picture of the scope output in your report.*
- You may decrease R4 down to 100Ω to increase the volume.