Linear Integrated Circuits

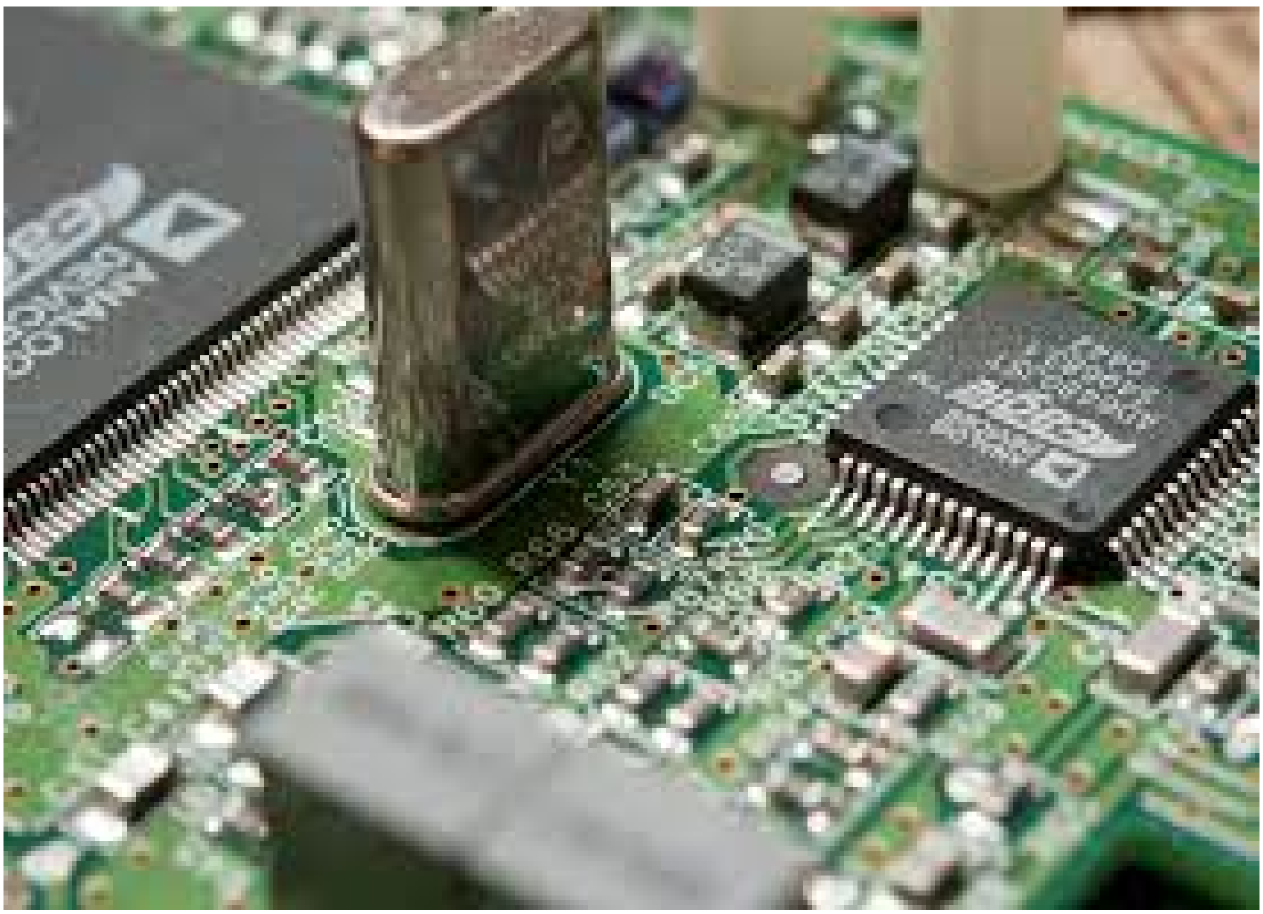
PWM Signal Generation and Simulation Using 555 Timer in LTSpice

Submitted by –

KESHAV SINGH(2K19/EE/134)

Submitted to –

DR. RAM BHAGAT



Problem Statement

Most machines (such as a sewing machine motor) require partial or variable power. In the past, control (such as in a sewing machine's foot pedal) was implemented by use of a rheostat connected in series with the motor to adjust the amount of current flowing through the motor, before that people used to use mechanical power to control speed. PWM is an energy efficient solution to such problems.
In this problem we aim to :

- Explore and study PWM signals.
- Learn to use LTSpice(create and manage different symbols and scripts).

Pulse Width Modulation

[1]Pulse-width modulation (PWM), or pulse-duration modulation (PDM), is a method of reducing the average power delivered by an electrical signal, by effectively chopping it up into discrete parts. The average value of voltage (and current) fed to the load is controlled by turning the switch between supply and load on and off at a fast rate. The longer the switch is on compared to the off periods, the higher the total power supplied to the load. Along with maximum power point tracking (MPPT), it is one of the primary methods of reducing the output of solar panels to that which can be utilized by a battery.[1] PWM is particularly suited for running inertial loads such as motors, which are not as easily affected by this discrete switching, because their inertia causes them to react slowly. The PWM switching frequency has to be high enough not to affect the load, which is to say that the resultant waveform perceived by the load must be as smooth as possible.

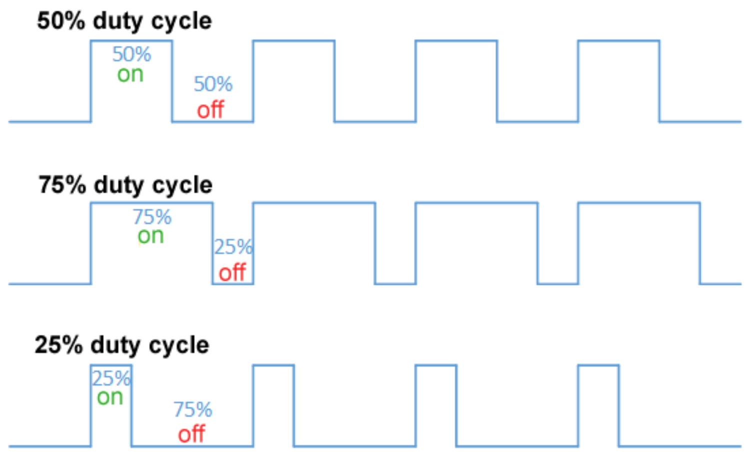


FIgure1: PWM Signal example

Motor Speed Control using PWM

[2] To control the speed of the motor, the average value of the PWM pulse is controlled by a switch. If the switch is 'on' for a long duration, the average value of the PWM pulse increases, driving the motor to high speed, and vice versa.

The use of pulse width modulation to control a small motor has the advantage in that the power loss in the switching transistor is small because the transistor is either fully "ON" or fully "OFF". As a result the switching transistor has a much reduced power dissipation giving it a linear type of control which results in better speed stability.

Also the amplitude of the motor voltage remains constant so the motor is always at full strength. The result is that the motor can be rotated much more slowly without it stalling. So how can we produce a pulse width modulation signal to control the motor? Easy, use an Astable 555 Oscillator.

Most circuits use a potentiometer as the variable switch to generate PWM. It is cumbersome to use because it needs manual movement of the potentiometer by the user. In this circuit, we use a thermometer decoding circuit to control the switch resistance. This, in turn, controls the duty cycle of the pulse, thereby producing PWM. THe simulation is given in a further section.

What is a Ripple Counter?

[3] Before jumping to Ripple Counter we should be familiar with the terms Synchronous and Asynchronous counters. Counters are circuits made using flip-flops.

Synchronous Counters

Synchronous counters, as the name suggests have all the flip-flops working in sync with clock pulse as well as each other. Here the clock pulse is applied to every flip flop.

Asynchronous Counter

Whereas in Asynchronous counter clock pulse is applied only to the initial flip flop whose value would be considered as LSB. Instead of the clock pulse, the output of the first flip-flop acts as a clock pulse to the next flip flop. The outputs of all flip-flops do not change affect at the same time.Here flip-flops are connected in Master-Slave arrangement.

Ripple counter is an Asynchronous counter. It got its name because the clock pulse ripples through the circuit. An n-MOD ripple counter contains n number of flip-flops and the circuit can count up to 2n values before it resets itself to the initial value.

These counters can count in different ways based on their circuitry.

- **UP COUNTER**: Counts the values in ascending order.
- **DOWN COUNTER**: Counts the values in descending order.
- **UP-DOWN COUNTER**: A counter which can count values either in the forward direction or reverse direction.

- **DIVIDE by N COUNTER**: Instead of a binary, we may sometimes require to count up to N which is of base 10. A Ripple counter which can count up to value N which is not a power of 2 is called Divide by N counter.

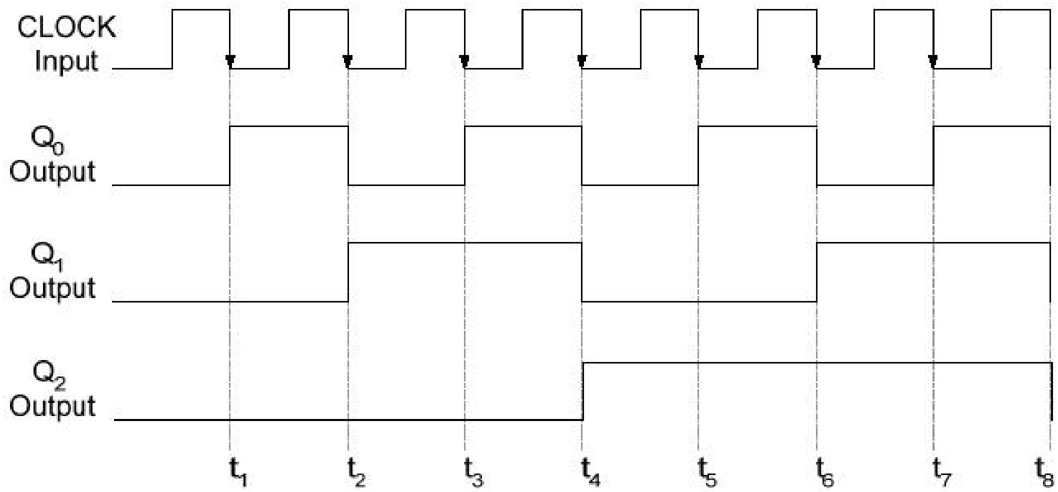


Figure2: Time signal

The working of the ripple counter can be best understood with the help of an example. Based on the number of flip flops used there are 2-bit, 3-bit, 4-bit..... ripple counters can be designed. Let us look at the working of a 2-bit binary ripple counter to understand the concept.

555 Timer Circuit

[4] Multivibrators and CMOS Oscillators can be easily constructed from discrete components to produce relaxation oscillators for generating basic square wave output waveforms.

One such device that has been around since the early days of IC's and has itself become something of an industry "standard" is the 555 Timer Oscillator which is more commonly called the "555 Timer".

The basic 555 timer gets its name from the fact that there are three internally connected 5kΩ resistors which it uses to generate the two comparators reference voltages. The 555 timer IC is a very cheap, popular and useful precision timing device which can act as either a simple timer to generate single pulses or long time delays, or as a relaxation oscillator producing a string of stabilised waveforms of varying duty cycles from 50 to 100%.

The 555 timer chip is extremely robust and stable 8-pin device that can be operated either as a very accurate Monostable, Bistable or Astable Multivibrator to produce a variety of applications such as one-shot or delay timers, pulse generation, LED and lamp flashers, alarms and tone generation, logic clocks, etc.

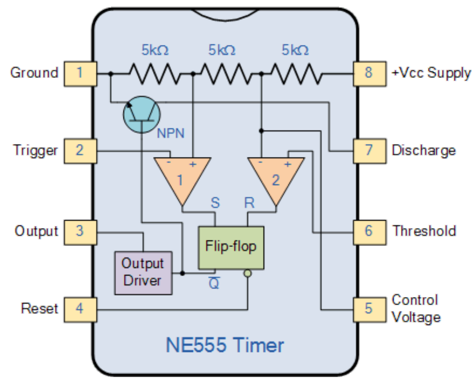


Figure3: NE555 Timer Circuit

Thermometer Code

[5] Thermometer code resembles the output produced by a thermometer. In thermometer code, a value representing number 'N' has the lowest 'N' bits as '1'; others as 0. So, to move from N to 'N+1', just change the rightmost '0' to '1'. As is evident, each value resembles a reading in the thermometer. This is how, thermometer code got its name. Flash ADCs, time-to-digital converters (TDC) are some of the circuits that utilize thermometer code.

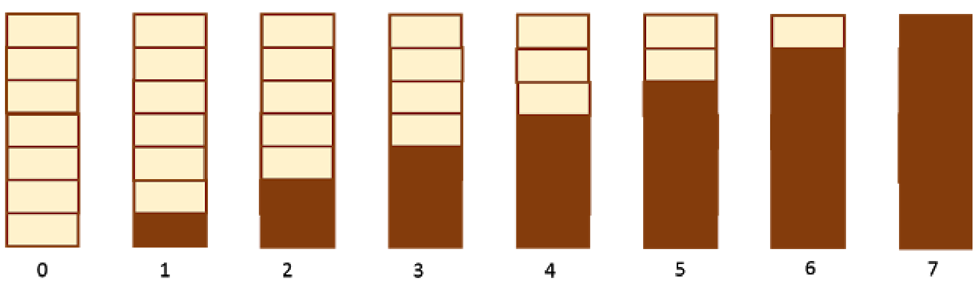


Figure4 : Thermometer Code with 7 Symbols

Characteristics of Thermometer Code

- Each symbol in thermometer code is a sequence of 0s followed by a sequence of 1s
- There cannot be 0s in-between two 1s. For example, a symbol 01011 is invalid in thermometer code
- For an n-bit binary code, the corresponding thermometer code will have 2n – 1 symbols; hence, as many bits will be needed to represent thermometer code for the same.

INPUTS			OUTPUTS							
a	b	c	O0	O1	O2	O3	O4	O5	O6	O7
0	0	0	1	1	1	1	1	1	1	1
0	0	1	0	1	1	1	1	1	1	1
0	1	0	0	0	1	1	1	1	1	1
0	1	1	0	0	0	1	1	1	1	1
1	0	0	0	0	0	0	1	1	1	1
1	0	1	0	0	0	0	0	1	1	1
1	1	0	0	0	0	0	0	0	1	1
1	1	1	0	0	0	0	0	0	0	1

FIgure5: Logic Table for Binary-to-Thermometer Decoder

SIMULATION in LTSPICE

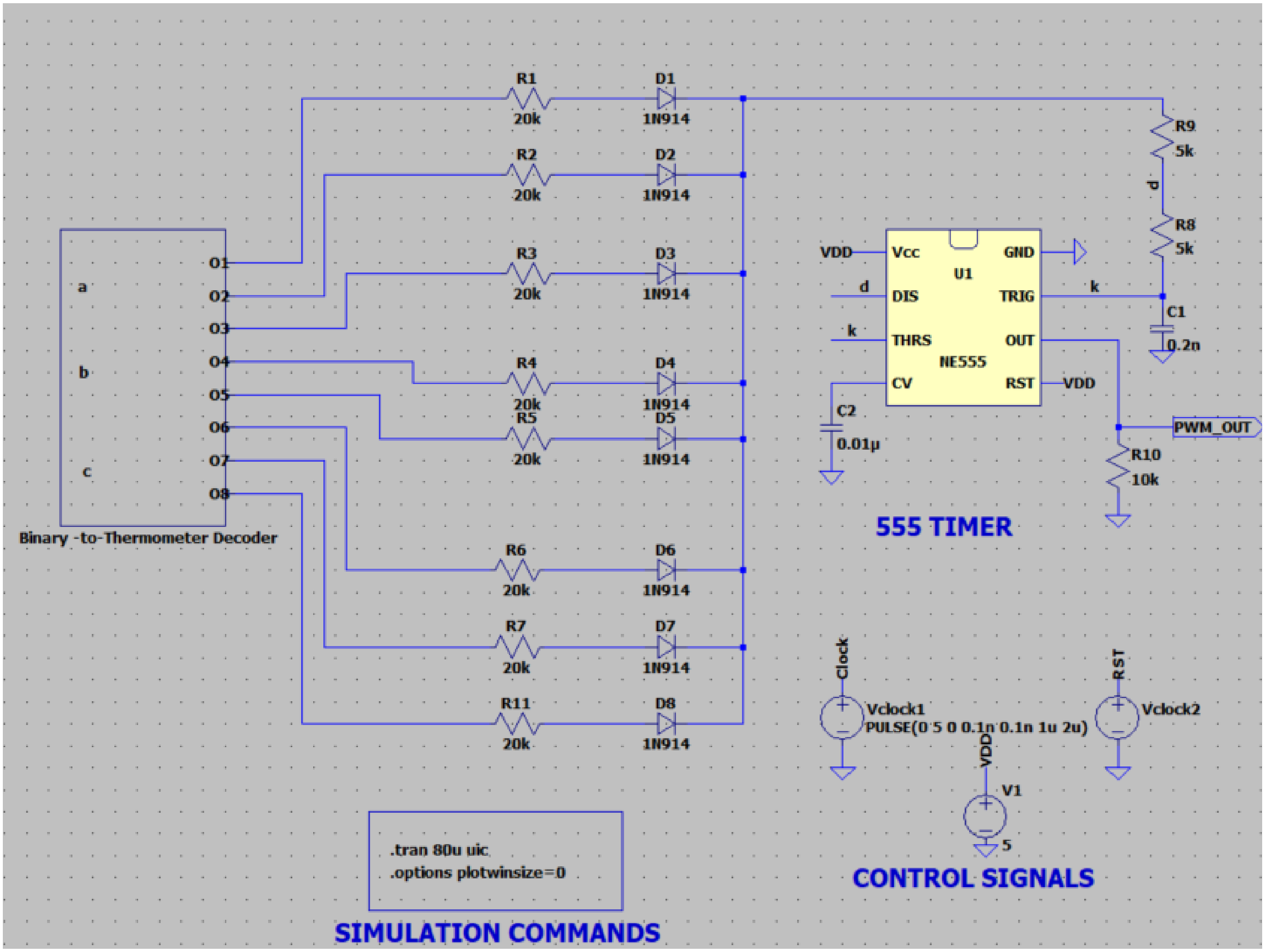


FIgure6: Full Circuit SImulation in LTSPICE

- This simple circuit based around the familiar NE555 or 555 timer chip is used to produce the required pulse width modulation signal at a fixed frequency output. The timing capacitor C is charged and discharged by current flowing through the timing networks.

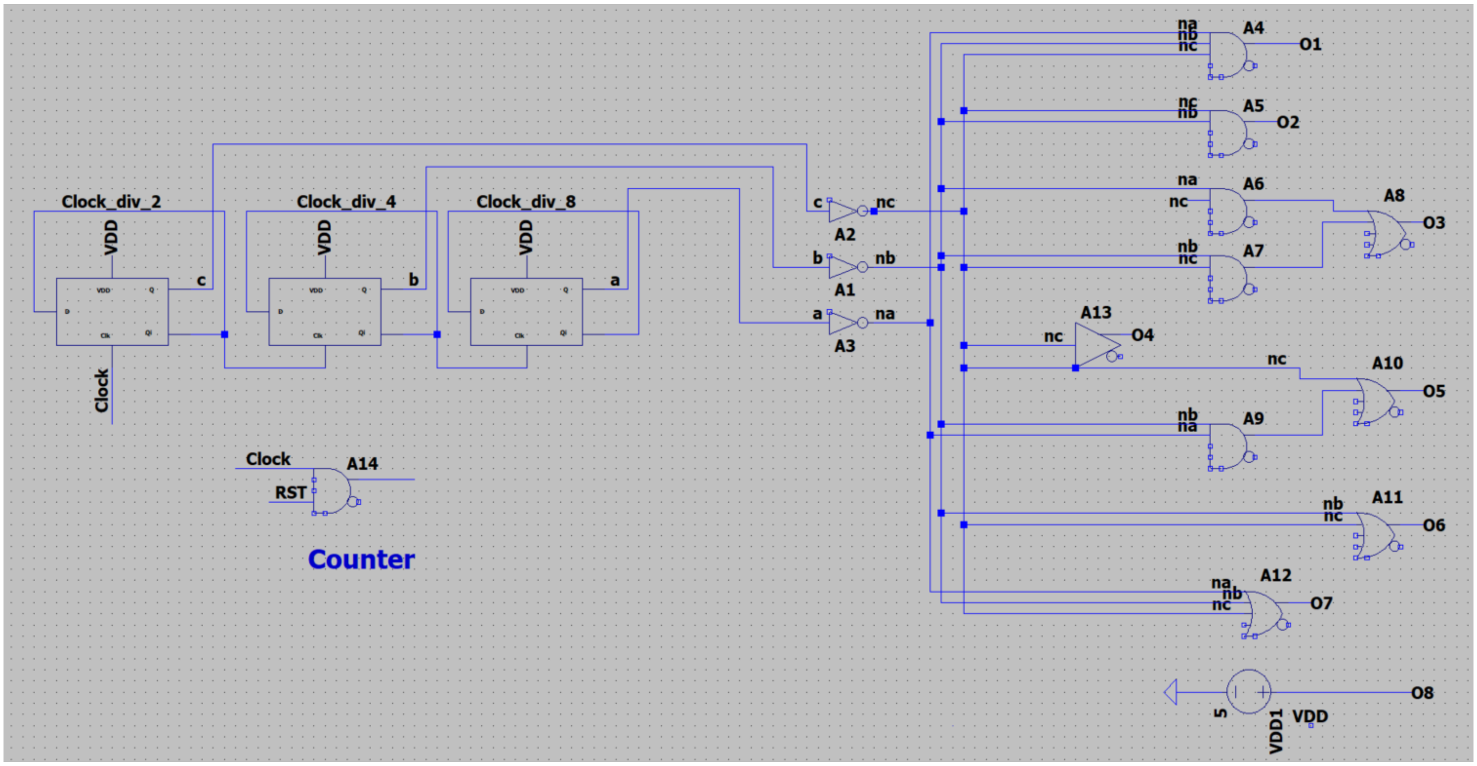


FIgure7: BInary-to-THermometer Decoder

- The circuit mainly consists of the following components: 3-bit counter, binary-to-thermometer decoder logic, 555 timer IC, resistors (R1-R8=20 kilo-ohms, R11=R12=100 kilo-ohms), diodes (1N914), capacitors (C1= 0.2 nF, C2=0.01 μF) and supply voltage (VDC= 5V).
- To construct this circuit we use D Flips Flops and Logic gates, the theory behind this circuit is given in the above sections.
- The decoder circuit is derived using the logic table given in Figure5.

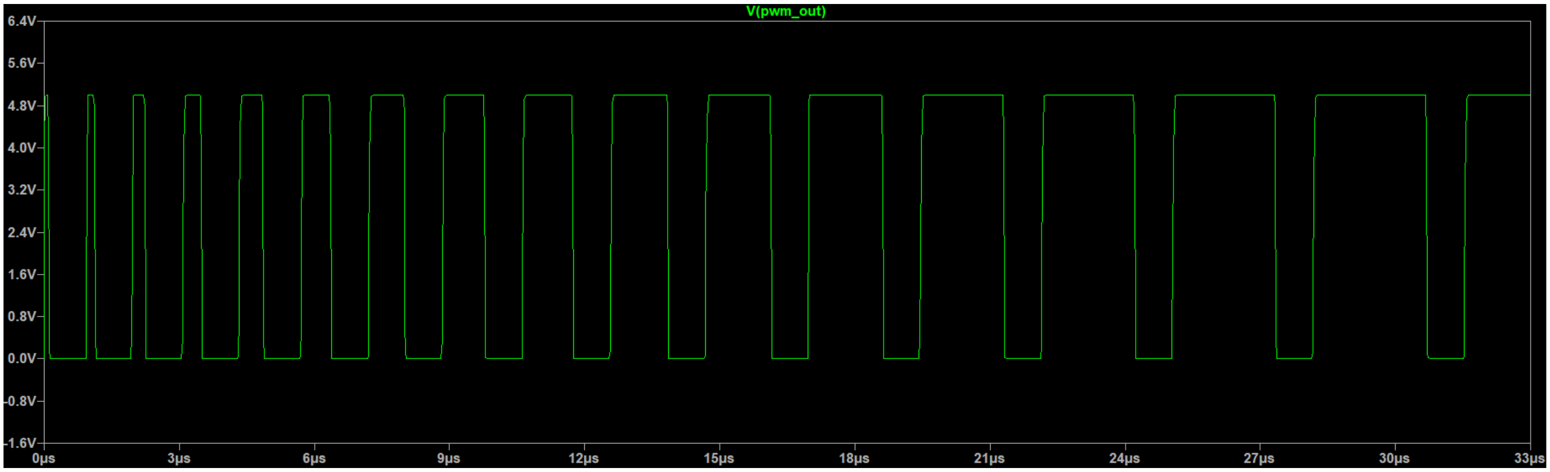


FIgure8: Output SIgnal

- In the above graph(Figure8) the result of the schematic shown in Fig. 1 is given, without RST logic but with a clock signal given directly to the counter. This output is obtained at the output pin of the 555 timer circuit to see the pulse-width-modulated output.
- Here you can see that the duty cycle of the waveform continuously changes as the binary code transitions from 000 to 111. If these pulses are fed to a DC motor, the speed of the motor will vary as per the width of the pulse train.

Conclusion

This report describes the complete design and implementation of a PWM signal generator with variable duty cycles, this can be majorly used for speed control of motors. This experiment can be further improved upon by giving a wider range of duty cycles values instead of just 8, also the concepts of Digital Circuit and Design systems can be used to further reduce the complexity of such a circuit.

References

- [1] https://en.wikipedia.org/wiki/Pulse-width_modulation
- [2] https://www.electronics-tutorials.ws/blog/pulse-width-modulation.html
- [3]https://www.elprocus.com/a-brief-about-ripple-counter-with-circuit-and-timing-diagrams/
- [4] https://www.electronics-tutorials.ws/waveforms/555_timer.html
- [5]https://en.wikipedia.org/wiki/Unary_coding#:~:text=Unary%20coding%2C%20or%20the%20unary,number%20is%20understood%20as%20strictly