EE 316 - Electronic Design Project

Project: P2

PWM Data Encoder

Final Project Report

10 June 2024

Objective

Pulse Width Modulation (PWM) is a technique based on representing a signal as a rectangular wave with a varying duty cycle. The main objective of PWM is to control a load by selecting its PWM switching frequency. The most important feature of using the PWM method in data transmission is that since we can adjust the duty cycle of the signals, it provides controllable systems such as speed of a motor, amplitude of a sound in radio, brightness of the displays, etc. This project proposes PWM data transmission at varying frequencies so that the system could still transmit data even if the frequency conditions are changed.

Group Members

Common efforts: Research on designing the subsystems for the overall system

Cem Tolga Münyas: PWM circuits implementation, averaging filter design, mapping circuit design & implementation

Ozan Gülbaş: Data generation D1 & D2 signals, PWM circuits implementation, voltage divider circuit design & implementation

Revision History

Week-1: Planning and Research

Week-2: Setup Simulation Environment and Signal Generation Schema

Week-3: Simulate Signal Generation and Decoder Implementation

Week-4: Setup Monostable Multi Vibrators (MMVs) and Averaging Filter Design

Week-5: Averaging Filter Design

Week-6: Averaging Filter Implementation on Simulation Environment

Week-7: Frequency to Voltage (F2V) and Voltage Mapper Circuit Implementation

Week-8: Prepare Bill of Materials (BOM) List and Shop for Necessary Parts

Week-9: Setup Signal Generator, MMV and Decoder Circuits on Breadboard

Week-10: Setup Frequency to Voltage and Averaging Filter on Breadboard

Week-11: Setup Voltage Mapper Circuit on Breadboard and End to End Testing

Week-12: Testing

Week-13: Testing

Week-14: Final Report

Week-15: Presentation

1. Introduction

To make a PWM Data Encoder that satisfies given specifications, we need a couple peripheral circuits and components other than the PWM Data Encoding component of the system, which in turn increases the complexity of the overall system. The system consists of these components:

1. Data Generation: The system is connected to a clock signal that can vary between 10kHz to 20kHz. This clock signal determines the data transmission speed of the system, and we derive our test data by conducting several operations to clock signal.

2. PWM Encoder: This is the main part of the system that encodes given data to pulse width modulated symbols using a 2x4 Decoder and 4 MMVs that are adjusted to generate %25, %50, %75, and %99 duty cycle square waves.

3. Frequency-to-Voltage Converter: Since our clock signal can change from 10kHz to 20kHz, our MMVs need to compensate for this change and adjust the appropriate T_{on} duration to generate consistent PWM signals. This is achieved by using a MMV, a Low-Pass Filter and benefiting from semiconductor physics.

4. Voltage Mapper: This component is needed for scaling the output of the Frequency to Voltage Converter to the MMV's suitable supply voltage range.

The following paragraphs summarize the necessary background information for the key circuits utilized in this project.

1.1. Data Generation

Our system is capable of 2-bit data transmission using 2 data buses, D1 and D2. Since our expected data to encode is in the form of high and low signals, representing "1" s and "0" s, for a duration of 1 clock cycle, we can divide the clock signal frequency in half two times using D-Flip-flops, and have our first data line, D1. To simulate every possible transition that may occur in a data transmission we need a very specific D2 signal, which we can generate by taking XNOR between the half of the clock signal and D1 data signal. This gives us a similar signal to D1 but shifted one clock signal period forward.

With this setup we have a complete test signal which will allow us to see if our encoding schema works correctly.

1.2. PWM Encoder

This part is the main part of the overall system that generates PWM signals according to incoming signal. Since we have high and low signals coming from the data generator part of our system, we need a circuit to first determine what symbol is needed to encode given data, and a circuit to generate the corresponding PWM signal to that symbol.

To determine which PWM generating circuit will be used to encode data signals, we can use a 2x4 Decoder, since we only have 2 bits to encode and 4 different MMV circuits to generate corresponding PWM signals.

Monostable Multi Vibrator (MMV) is a special type of circuit that has one stable state and one quasi-stable state. When a trigger signal is applied, the circuit is switched between these two states. To be more specific, the circuit is in stable state until a trigger signal is applied and when a trigger signal is applied, the circuit stays in quasi-stable state for a specific duration of time, also referred to as T_{on} duration. This duration can be adjusted using external resistors and capacitors and also changing the supply voltage of the MMV.

We have four different MMV circuits, each is adjusted to generate %25, %50, %75 and %99 duty cycle PWM signal when a trigger signal is applied at 20kHz clock signal. Each of these MMV circuits trigger pins are connected to the 2x4 decoder outputs with an edge detector circuit with negative edge isolation in between.

1.3. Frequency-to-Voltage Converter

One of the biggest challenges of this system is to maintain consistent encoding, meaning no data loss, with varying clock signal. All four MMVs are adjusted to generate their corresponding duty cycle PWM signals at 20kHz clock signal, thus when the clock signal is decreased to 10kHz, if the T_{on} duration won't change its duty cycle will decrease against increasing clock period. This means the same data encoded at 20kHz will not be the same as encoded at 10kHz.

To overcome this challenge, we need a way to change T_{on} duration of the MMVs without changing the external resistor and capacitors that are used to generate corresponding duty cycle PWM signals at 20kHz. Luckly, by the help of the semiconductor physics, the T_{on} duration is not only dependent on the external resistors and capacitors, but also a coefficient "K" which changes by temperature and supply

voltage. Since the temperature of the circuit is hard to control effectively, we can use changing supply voltage to compensate widening clock signal period.

We can use a MMV to generate a PWM signal with a specific duty cycle within each clock cycle and a low-pass filter to take the average of the PWM signal generated by the MMV. With this setup, when the clock cycle changes, the duty cycle of the PWM signal generated by the MMV is changed and when we take the mean of that signal, we get a varying constant voltage value with the maximum value at 20kHz and the minimum value at 10kHz.

One of the challenges we faced when applying this circuit is to design a low-pass filter that can take average of the signal that MMV generated using as minimum clock cycles and ripple voltage as possible.

1.4. Voltage Mapper

We solved how the PWM Encoder will maintain its PWM signals with proper duty cycles by using Frequency to Voltage converter. However, the maximum voltage value we get at 20kHz and the minimum voltage value we get at 10kHz are out of bounds of the MMVs operating voltage range. Thus, we need a circuit to map Frequency to Voltage Converter output to the MMVs operating voltage range.

When this type of operation is mathematically represented, we can design such a circuit using operational amplifiers and map the input range to desired output range.

2. Technical Description

2.1 Data Generation

Every encoder needs something to encode, and, in this project, we need a data stream that can simulate all the possible transitions and bit combinations for 2-bit data transmission through two data buses. We already have a clock signal as an external input and each of our bits' length should be exactly one clock cycle. If we had a signal that is half the clock signal, we would use it as one of the data signals since it would have lows and highs changing at one clock cycle consecutively.

However, since the requirement for this project is to transmit 2-bits we need another signal too. If we delay the first signal one clock signal and use it as the second data signal, we will have appropriate data signals, but it would lack the requirement that is the data signals we need should simulate all the bit combinations and transitions. Thus, we need a different approach.

If we take half of the signal one more time, meaning dividing the clock signal frequency by four, we will have a data signal that changes between highs and lows in two clock cycles. This means that we have low (0 in binary) for two clock cycles and high (1 in binary) for two clock cycles, on repeat. Now if we delay this signal by a one clock cycle and use it as the secondary data signal, we successfully simulate all the combinations and transitions in the 2-bit data stream.

For this concept to be applicable, we need to find a way to divide the frequency of a signal by half, and a way to delay a signal by one clock cycle. To divide the frequency by half, we can use a D-Flipflop in a specific configuration shown in Figure 1.

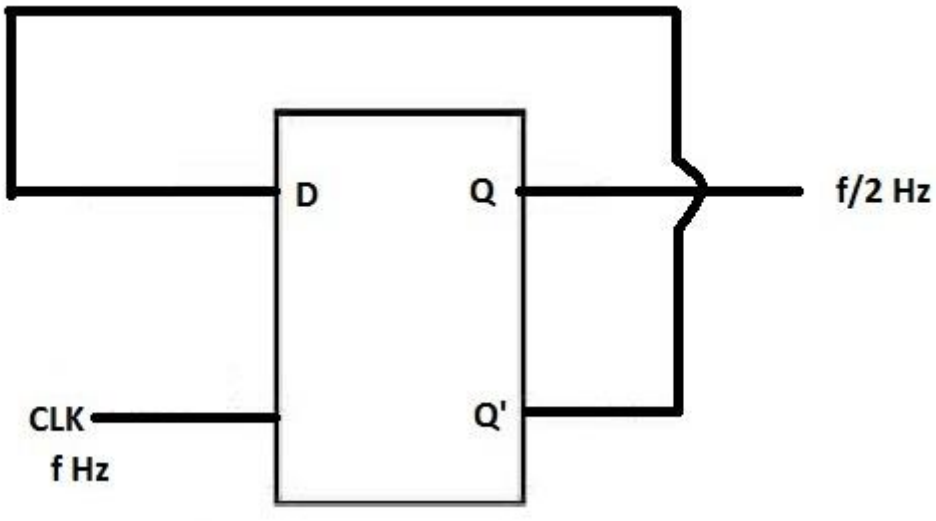

Figure 1. A frequency divider using D-Flipflop

With this circuit we can generate our first data signal (D1) using the circuit above two times. For the second data signal (D2) we still need to find a way to delay the D1 signal one clock cycle. When we look at the derived signals while we generate the D1 signal, we can use the signal with the half the frequency of the clock signal and D1 by passing them through a XNOR gate and the resulting signal is exactly one clock cycle delayed version of the D1 signal. In Table 1 we can clearly see why this method works.

CLK/2	D1	D2
0	0	1
1	0	0
0	1	0
1	1	1
0	0	1
1	0	0
0	1	0
1	1	1

Table 1. Truth table of D2 signal

By combining these we have a unit that takes a clock signal and generates two data signals that simulate all 2-bit data combinations and transitions. The complete circuit is shown in Figure 2.

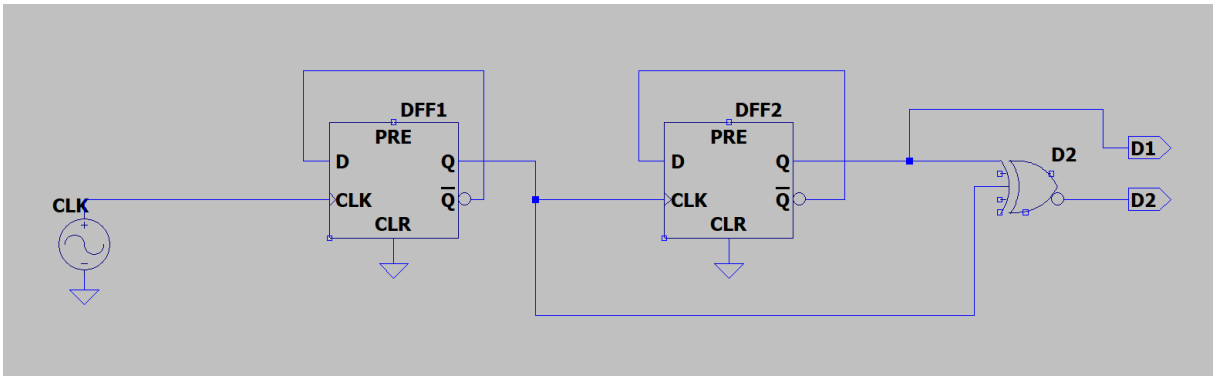

Figure 2. Data generation unit

2.2 PWM Encoder

At this point of the project, we need to convert the data signals D1 and D2 into PWM signals by using the monostable multivibrators. Basically, we want to transmit 2 bits so that we need to have 4 PWM signals with different duty cycles. We chose the duty cycles as %25, %50, %75 and %99. These signals represent the bits that we

transmit in one clock cycle. To realize this system architecture, monostable multivibrators were used. By only changing the resistor values, we could change the duty cycle of the output signals.

We used the MMV to generate signals with different duty cycle values because the data should be transmitted when an external trigger signal is applied. In MMVs, there is one stable and one quasi-stable state. The output remains in the stable state as default. When the external trigger signal is applied, then output goes into the quasi-stable state for a moment. This mechanism is generally used for pulse width modulation as in our case.

Within the scope of this process, 74HCT123 dual re-triggerable MMV is used for generating 4 pulse width modulated signals with different duty cycles as mentioned earlier.

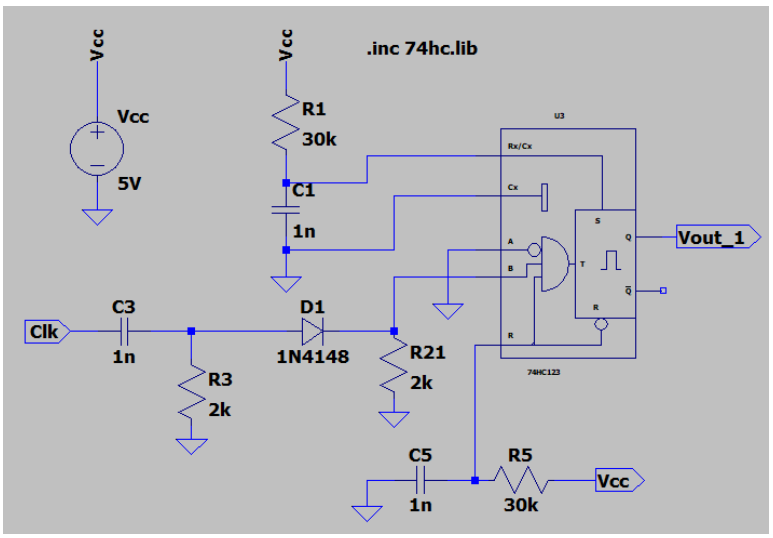

Figure 2. 74HC123 module which generates PWM signal with %25 duty cycle

The duty cycle is basically adjusted with the time constant $\tau = R \times C$. We choose the capacitors that are used in MMV circuits to be 1 nF everywhere. So, when the resistor values are changed logically, we could adjust the time constant for 4 different duty cycles. From the datasheet of 74HC123, typical output pulse width can be calculated as

$$T_W = K \times R_{ext} \times C_{ext}$$

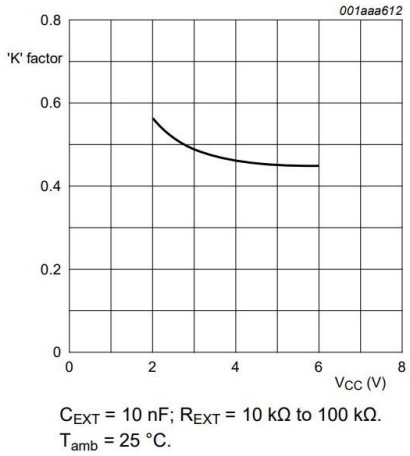

Figure 3. Typical K constant as a function of V_{CC}

From Figure 3, we deduce that changing the V_{CC} without changing any capacitor or resistor value in circuit could directly affect the pulse width of the modulated signal which is basically the duty cycle which we focus on.

Here are the simulation results for the PWM output for the %25 duty cycle signal.

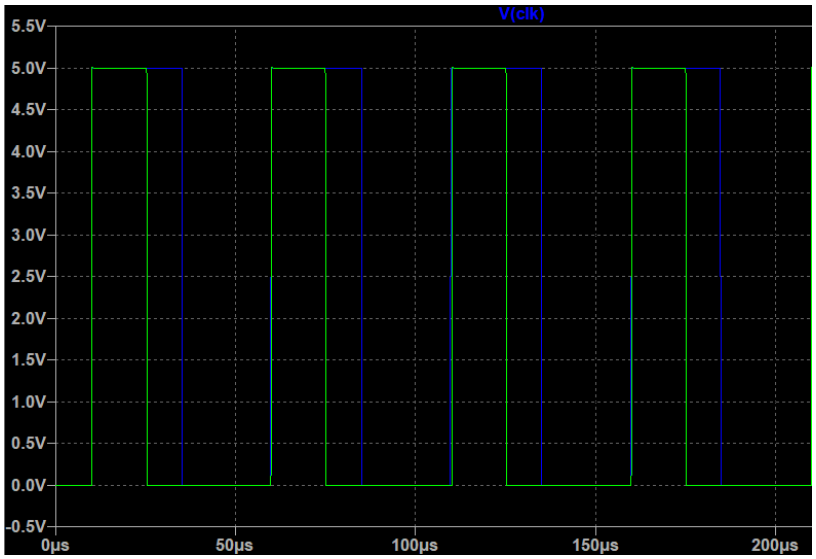

Figure 4. PWM output for %25 duty cycle signal

Moreover, the PWM circuit has also an edge detector circuit which consists of a capacitor and a resistor and is fed by +5V. It ensures that the monostable multivibrator is triggered by specific signal edges. The external capacitor and external resistor R1 and C1 determine the duration of the output pulse which is the duty cycle of the output signal. The input is a 20 kHz clock signal that switches between 0 and 5V.

The combination of C3 and R3 differentiates the clock signal, producing short pulses on both the rising and the falling edges. After that, diode D1 allows only positive pulses to pass, clamps the negative pulses. The resulting pulse is used to trigger the MMV.

This edge detection mechanism ensures that the MMV responds reliably to the start of each input clock signal, providing a precise output pulse which we later use one of them for the F2V converter also.

Description of design target	Min.	Max.	Unit
Input voltage	0	5	V-rms
Output low voltage	1.35	2.1	V-rms
Output high voltage	2.4	3.15	V-rms
Output duty cycle	25	99	%
Supply Voltage	2.0	6.0	V
Operating frequency	10	20	kHz

Table 2. Design targets for generating the PWM signals

2.3 Frequency-to-Voltage (F2V) Converter

A frequency-to-voltage converter (F2V) is the circuit that converts the frequency of an input signal into a proportional level of voltage output. Since the operating frequency interval in PWM data encoder is between 10 kHz and 20 kHz, we need to convert the frequency values into the proportional voltage levels by using an F2V converter. The input signal is the clock signal, and the frequency of this signal varies within the range of 10 kHz and 20 kHz.

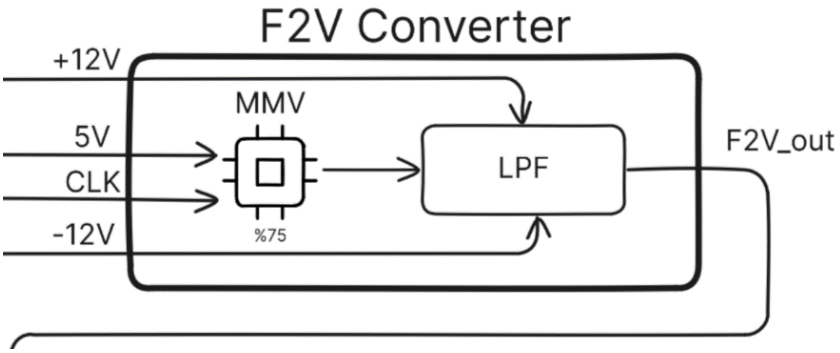

Figure 5. F2V Converter block diagram

Specifically, we chose that the MMV produce %75 duty cycle PWM signal at 20 kHz, and then passes it through an averaging filter to produce a DC voltage that is mean of the MMV's PWM output, providing a output voltage which corresponds to the frequency range of the clock signal. We designed a low pass filter as an averaging filter which basically aims to smoothen the PWM signal into a DC voltage that is proportional to the signal's duty cycle. The low pass filter removes the high-frequency components (switching components of the PWM signal) and averages the pulse train into a steady DC voltage.

The averaging filter used is a fourth order Bessel low pass filter. It is basically the combination of two operational amplifiers with selected capacitors and resistors to minimize the delay time and the ripple of the output signal. Delay time for the low pass filter output is 4 clock cycles and the ripple voltage is 0.95 mV at 20kHz, so it can be considered as good value for the low pass filter. Here is the schematic of the fourth order Bessel low pass filter to obtain the F2V output.

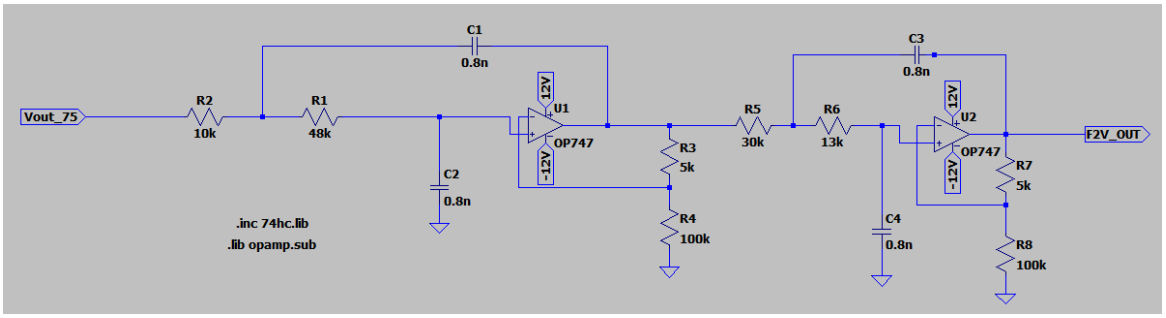

Figure 6. Averaging filter design

From the simulation output of the F2V (Figure 7) we can clearly see that the delay time is approximately 4 clock cycles which means at the beginning of the data transmission, we lost 4 bits. This data loss is negligible. On the other hand, averaging filter output converges to 4.2 V. These calculations are based on the clock signal which is the 20 kHz clock signal varies between 0-5 V. Since the operating interval is between 10 kHz and 20 kHz, we also generated the same operation and noted the results. At 10 kHz, the delay time and the ripple level of the signal is way larger than the 20 kHz clock signal. To minimize the delay in 1 clock cycle, and the ripple voltage the capacitor and resistor values we use in this averaging filter are the results of a long trial and error operation.

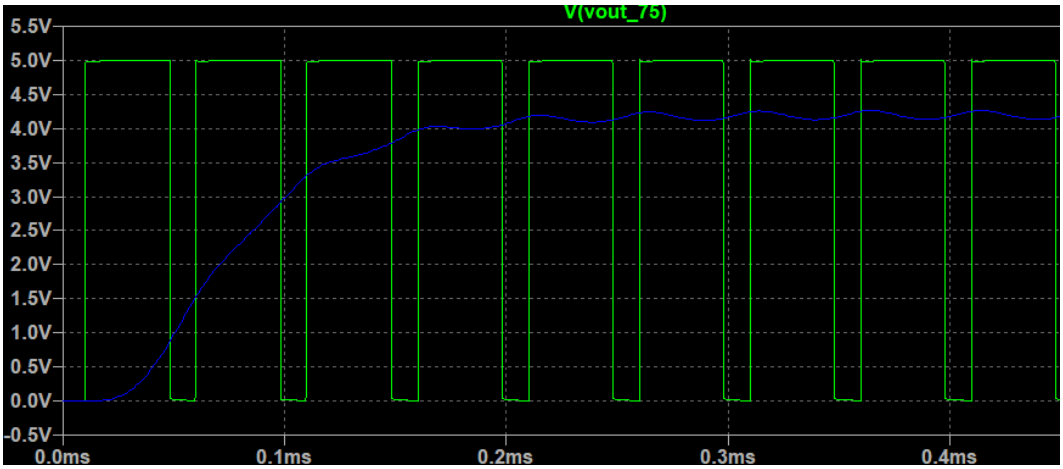

Figure 7. F2V output for 20 kHz

Description of design target	Min.	Max.	Unit
Input voltage	0	5	V-rms
Input signal duty cycle	75	75	%
Delay time	4	5	clk-cyc
Ripple voltage	95	585	mV
Non-inverting supply voltage	-12	+12	V
F2V output voltage	2.0	4.20	V
Operating frequency	10	20	kHz

Table 3. Design targets for the averaging filter

2.4 Voltage Mapper

The results show that the operating interval for the F2V output is between

The results show that the operating interval for the F2V output is between 2.0V and

4.2 V. At this point, we need to convert this operating interval to

$$3.2\ V \leq V_{CC}\ (V_{out}) \leq 5.0\ V$$

Which is the operating range of the used MMVs.

A simple linear mapping function is used to convert the operation range.
where *a* is the scaling factor and *b* is the DC offset.

$$Vout = a\ x\ Vin + b$$

The difference of the output range:

$$5V - 3.2V = 1.8V$$

The difference of the input range:

$$4.2V - 2.0V = 2.2V$$

The scaling factor a can be found as

$$a = \frac{Output\ span}{Input\ span} = \frac{1.8\ V}{2.2\ V} \cong 0.818$$

To determine the DC gain (offset), we used one point to solve for b.

$$3.2V = 0.818\ x\ 2.0V + b$$

$$b = 3.2 - (0.818\ x\ 2.0) = 1.564$$

For V_{in} = *2V* and V_{out} = *3.2V,* the DC gain can be calculated as

$$3.2 = -0.818\ x\ (-2 - 1{,}911\ )$$

To implement the voltage mapper that we found the values from the mapping equation, the combination of negative adder and inverting amplifier is used. At the first phase, we add the DC gain and the F2V output negatively. After that phase, inverting amplifier amplifies the resultant signal from the first phase and changes the sign of it.

There is a drawback here, since we first used the negative adder and then scaling factor, we need to revise our calculations because of the operation priority. Here is the calculation that we rearranged.

$$a\ Vin + b = -a\ (-Vin - \frac{b}{a})$$

$$0.818\ x\ 2 + 1.564 = -0.818\ x\ (-2 - \frac{1.564}{0.818})$$

The DC gain that we added at the first phase is revised as 1.911V. Here is the combination of negative adder and the inverting amplifier operational amplifiers. This combination forms the voltage mapper.

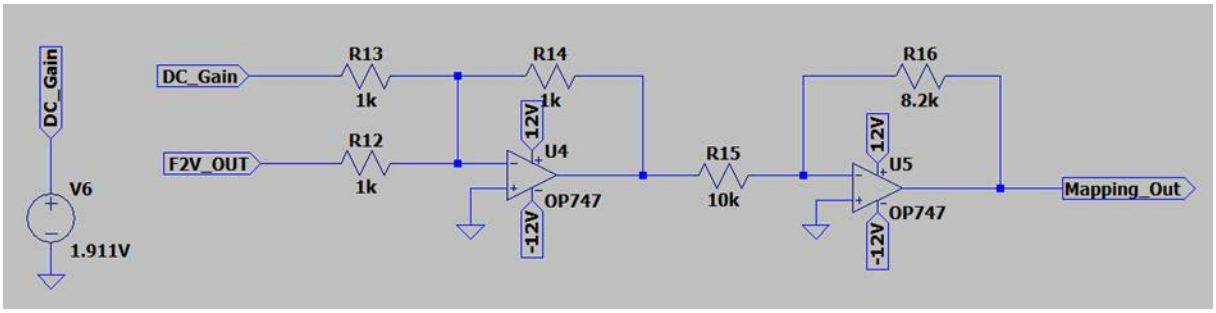

Figure 8. Voltage mapper circuit

The first op-amp circuit constitutes the negative adder feature by using 3 identical resistors. The output of the negative adder is then transmitted to another op-amp which acts as an inverting amplifier to generate the voltage mapper output.

Through the voltage mapper circuit, we ensure that K constant in MMV increases or decreases proportionally with the varying frequencies between 10 kHz and 20 kHz.

Here is the simulation result which shows that mapping output is generated by using the F2V output and the DC gain that we determined as 1.911V before.

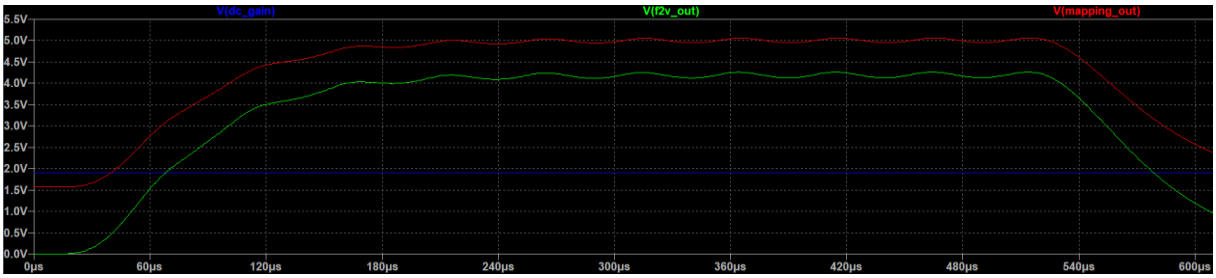

Figure 9. Simulation results for voltage mapping output

The target of this design is basically adjusting the K constant in the pulse width modulation relation

$$Tw = K\ x\ Rext\ x\ Cext$$

so that when we change the frequency between 10 kHz and 20 kHz, K is decreased or increased proportionally to satisfy the voltage requirements.

Description of design target	Min.	Max.	Unit
Input voltage (F2V output)	2.0	4.2	V
Input voltage (DC gain)	1.911	1.911	V
Mapping output voltage	3.2	5	V
Non-inverting supply voltage	-12	+12	V
Operating frequency	10	20	kHz

Table 4. Design targets for the voltage mapper

The voltage mapper block is shown below.

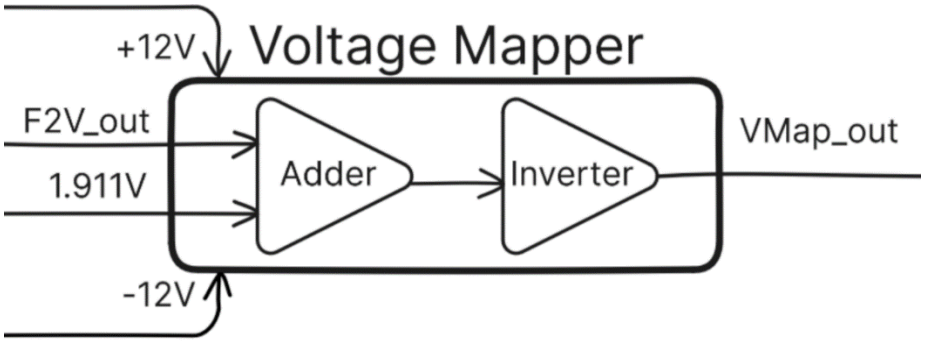

Figure 10. Voltage mapper block diagram

2.4.1 Voltage Divider

One of the input signals in voltage mapper block is the 1.911V DC gain. Since we cannot apply directly 1.911V to the circuit just for the DC gain, we need to convert 5V V_{CC} signal into the DC gain by some voltage manipulation method. This method is voltage divider which uses 2 different resistors R_1 and R_2, so that we could obtain the 1.911V in the most efficient way. The figure shown below indicates the voltage divider

circuit

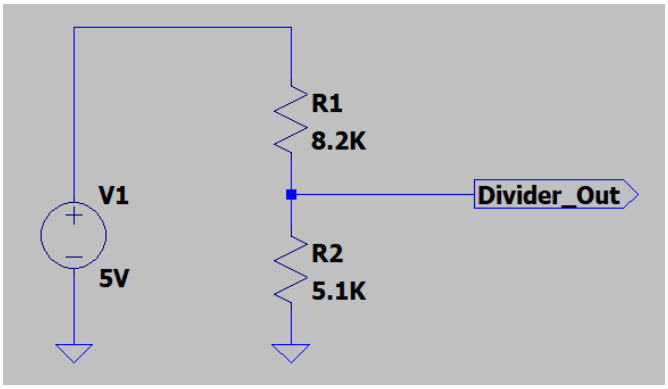

Figure 11. Voltage divider circuit

Here are the calculations to determine the proportional values of the resistors R1 and R2 to obtain 1.911V in the divider output.

$$5V\ x\ \frac{R2}{R1+R2} = 1.911\ V$$

$$1.616\ R2 = R1$$

$$R1 = 8.2\ k\Omega \quad and \quad R2 = 5.1\ k\Omega$$

Finally, a buffer is added to isolate the load from the voltage divider circuit. This way we can have desired voltage from the voltage divider circuit without the effect of load.

Description of design target	**Min.**	**Max.**	**Unit**
Input voltage	**5**	**5**	**V**
Output divider voltage	**1.911**	**1.911**	**V**

Table 5. Design targets for the voltage divider circuit.

3. Test Results

3.1 PWM Signal Generation Test Results

The measurements corresponding to the generating PWM signals for 4 different duty cycles as targeted aspect are given in the following table.

PWM Signal Generation Test Results			
Design Target	**Min.**	**Max.**	**Unit**
Input voltage	**0**	**4.5**	**V-rms**
Output low voltage	**1.35**	**2.1**	**V**
Output high voltage	**2.4**	**3.1**	**V**
Output duty cycle	**25**	**90**	**%**
Supply Voltage	**2.0**	**6.0**	**V**
Operating frequency	**10**	**20**	**kHz**

Table 1. Test results for the PWM signal generation

We have measured the pulse width modulated signals with duty cycles %25, %50, %75 and %99 using an oscilloscope. We also checked the frequency of the PWM signals over a period of time to check for stability. We have faced some problems in this section. MMV modules was working stable at the frequency range between 10 kHz and 20 kHz. In spite of that, we want our signals to be changed in terms of duty cycles when the frequency varies. However, under the frequencies of 15 kHz, duty cycles of the modulated signals were not giving what is expected. Except this issue, there was something else which was the signal mirroring in the signals of %50 and %99 duty cycles. Signal mirroring caused difficulties for distinguishing between high and low states. We overcome this problem by decreasing the supply voltage to 4.5V.

The PWM signal generation tests have generally met the design targets, with minor deviations observed in duty cycle accuracy.

3.2 Frequency-to-Voltage (F2V) Converter Test Results

The measurements corresponding to the frequency-to-voltage converter using the 74HC123 module for %75 duty cycle PWM data signal and utilizing the averaging filter can be shown below in the following table.

F2V Converter Test Results			
Design Target	**Min.**	**Max.**	**Unit**
Input voltage	**0**	**5**	**V-rms**
Input signal duty cycle	**70**	**75**	**%**
Delay time	**1**	**3**	**clk-cyc**
Ripple voltage	**100**	**600**	**mV**
Non-invertive input voltage	**-12**	**+12**	**V**
F2V output voltage	**2.0**	**4.23**	**V**
Operating frequency	**10**	**20**	**kHz**

Table 2. Test results for the F2V converter

The frequency to voltage converter successfully converts a PWM signal with a 75% duty cycle into a stable DC voltage. The use of a 4th order Bessel function filter ensures smooth and consistent output, with performance characteristics (delay time and ripple voltage) closely matching simulation results. F2V operates effectively within the 10 kHz and 20 kHz frequency range, which makes it suitable for the input of the voltage mapper.

3.3 Voltage Mapper Test Results

The measurements corresponding to the voltage mapper by combining a negative adder and an inverting amplifier can be shown below on the following table.

Voltage Mapper Test Results			
Design Target	**Min.**	**Max.**	**Unit**
Input voltage (F2V output)	**2.0**	**4.2**	**V**
Input voltage (DC gain)	**1.938**	**1.967**	**V**
Mapping output voltage	**3.2**	**5**	**V**
Non-invertive output voltage	**-12**	**+12**	**V**
Operating frequency	**10**	**20**	**kHz**

Table 3. Test results for the voltage mapper

Voltage mapper effectively adjusts the F2V converter output to the desired voltage interval using a combination of a negative adder and an inverting amplifier. We measured the DC gain as 1.967. It was very close to the value that we calculated. The small difference is the reason of resistor values that we chose for the voltage divider, and it does not affect the system output at all.

4. Conclusion

The primary objective of this project was to design and implement a PWM data encoder which is basically capable of transmitting data bits with the varying pulse width modulated signals over the frequency range of 10 kHz and 20 kHz. By using the clock signal, we have generated two synthetic data signals which was D1 and D2. After that phase, we decided to use the Gray Encoding algorithm in order to generate a meaningful data sequence that we will transmit. We encoded this binary data sequence as 4 different PWM data signals with duty cycles as %25, %50, %75 and %99. The PWM signals with the specified duty cycles were successfully generated, and the use of 74HCT123 monostable multivibrator allowed for precise pulse width control. These generated PWM signals with varying duty cycles representing the binary data so we basically designed a Digital-to-Analog converter (DAC) which could be used in many applications in transmitting data bits.

Afterwards, we designed a frequency-to-voltage (F2V) converter to ensure data transmission frequencies between 10 kHz and 20 kHz. The F2V converter accurately converted the frequency of the %75 duty cycle PWM signal to a proportional voltage. The averaging filter (low pass filter) effectively eliminated the high-frequency components of PWM signal, so we end up with a stable DC voltage.

Additionally, the voltage mapping stage was implemented to shift the voltage range of F2V which was 2V – 4V to the voltage range 3.2V – 5V so that the K constant in MMVs could change proportionally with the decreasing or increasing frequency. It enables the specified requirement of pulse width conservation over the changing frequencies between 10 kHz and 20 kHz. At the end of this phase, we observed that, output of the mapping circuit gives the desired values within the frequency range of 10 kHz and 20 kHz.

Despite the overall success of the project, several issues were encountered during the project. One of the major issues during the project was changing the duty cycle of the PWM signal while changing the operation frequency. Voltage limitation of the MMV did not allow us to change the %25 duty cycle of the PWM signal into %50 duty cycle within the frequency range 10 kHz and 15 kHz.

The system was able to change the duty cycle within the range 15 kHz and 20 kHz. Because of the voltage limitations of the MMV over 6V, we cannot change its duty cycle by only adjusting the frequency. It was one of the major issues that we faced during the project. The other issue which can be considered as an issue is that we though that we can feed non-invertive output voltage of the op-amps with $\pm 5V$ in order to use less voltage source in the hardware phase. When we do this operation, we faced that under some frequencies, the mapping output was not changing proportionally as expected. We overcome this issue by giving $\pm 12V$ to exact same pins of the op-amps. Other than that, the project of PWM data encoding was pretty problem-free.

To sum up the achievements in this project, the data signals D1 and D2 are created by some manipulations to the clock signal. After that, we used Gray encoding in order to generate a meaningful binary data sequence which will be transmitted. In order to represent the binary data in this sequence, we utilized the PWM outputs which generates 4 signals for 4 different duty cycles. The generated PWM signals met the duty cycle specifications (%25, %50, %75, %99). The design objective of precise pulse width modulation was achieved. Afterwards, by using an F2V converter, the frequency range between 10 kHz and 20 kHz are converted into the proportional voltage level. Then finally, this proportional voltage level is mapped to the input voltage range as the desired output voltage range which meets the project's objectives. In this way, we can transmit synthetic binary data sequences as PWM signals with varying duty cycles as in the frequency range 10 kHz and 20 kHz.

5. Component List

Component description	Part Number	Manufacturer	Supplier
Dual retriggerable monostable multivibrator with reset (MMV)	**74HCT123**	Philips	Çankaya
Dual D-Type positive edge triggered flip-flop with clear and preset	**SN74HC74**	Texas Instruments	Electronics Lab.
Dual operational amplifiers	**LM747**	Texas Instruments	Çankaya
Quad 2-input AND gate	**SN74LS08**	Texas Instruments	Çankaya
Quad 2-input XOR gate	**SN74AC86**	Texas Instruments	Çankaya
Triple 3-input NAND gate	**SN74LS10**	Fairchild Semi	Electronics Lab.
Buffer	**DM74LS125**	Fairchild Semi	Çankaya
Small signal fast switching diodes	**1N4148**	onsemi	Çankaya
Capacitors	**-**	-	Çankaya
Resistors	**-**	-	Çankaya

References

[1] *ALL ABOUT ELECTRONICS, "What is Multivibrator ? Astable, Monostable and Bistable Multivibrators Explained," YouTube, Aug. 22, 2018. https://www.youtube.com/watch?v=5clfiJtRhR8&pp=ygUYbW9ub3N0YWJsZSBtdWx0aXZpYnJhdG9y (accessed Jun. 09, 2024).*

[2] *engine euproject, "Active filters - Butterworth/Bessel/Chebyshev 4th order low pass filters (LTSpice)," YouTube, May 05, 2023. https://www.youtube.com/watch?v=FJXcgPk4Fv0&pp=ygUfNHRoIG9yZGVyIGJlc3NlbGxvdyBwYXNzIGZpbHRlcg%3D%3D (accessed Jun. 09, 2024).*

[3] *E. Tutorial, "Monostable Multivibrator - The One-shot Monostable," Basic Electronics Tutorials, Sep. 05, 2013. https://www.electronics-tutorials.ws/waveforms/monostable.html*

[4] *EL-PRO-CUS, "How To Design of 2 to 4 Line Decoder Circuit, Truth Table and Applications," ElProCus - Electronic Projects for Engineering Students, Jan. 23, 2018. https://www.elprocus.com/designing-of-2-to-4-line-decoder/*

[5] *Engineers Academy, "Simulating an Active Low Pass Filter in LTspice," YouTube, Feb. 22, 2023. https://www.youtube.com/watch?v=-ko3OPj7aGI (accessed Jun. 09, 2024).*

[6] *A. Lorsawatsiri, W. Kiranon, V. Silaruam, W. Sangpisit, and P. Wardkein, "Simple and Accurate Frequency to Voltage Converter." Available: https://web.mit.edu/Magic/Public/papers/05491499.pdf*

[7] *NXP, ". General description 74HC123; 74HCT123 Dual retriggerable monostable multivibrator with reset," 2015. Available: https://docs.rs-online.com/5ada/0900766b813eb825.pdf*