Analog Sorting Using Pulse Width Modulation

A Senior Project

presented to

the Faculty of the Computer Engineering Department

California Polytechnic State University, San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science

by

Riley Olson,

June, 2018

Analog Sorting using Pulse Width Modulation

Riley Olson rcolson@calpoly.edu

***Abstract*—As time goes on, computers become more and more powerful. However, as processing time becomes less of a limiting factor for computing tasks, power consumption takes its place for many tasks. This paper proposes and tests a new method for sorting analog signals. This new sorting method converts analog signals into Pulse Width Modulated(PWM) signals of varying duty cycle , which are then sorted by a simple network of combinational logic, and then converted to a normal binary representation. In order to implement this new method, multiple circuits had to be designed and and tested to ensure their functionality and power efficiency. Three circuits total were designed to convert analog voltages into variable duty cycle PWM signals, to sort the PWM signals, and to convert the PWM signals into a binary representation. Simulations of these circuits were tested in LTspice and Vivado to confirm their functionality and determine their power consumption. The simulation data was compared with power dissipation data for a standard Analog to Digital conversion and comparison-based sorting algorithm. The new method proposed in this paper provides an improvement in energy efficiency over the latter method, and can be used as a more efficient method of converting and sorting data received from analog sensors.**

***Index Terms*—Analog Signals, Sorting, Pulse Width Modulation, Combinational Logic, Analog to Digital Conversion, Energy Efficient Circuits.**

I. Introduction

The goal of this project was to find new ways to perform common computing tasks in a more energy efficient manner. It took many weeks of researching various computing problems before I chose analog sorting as the task that I would try to optimize. I chose this problem because it seemed that there was very little research in this area. Most sorting is done on digital data because of the speed of digital computers are in modern times. The process of converting analog signals to digital binary representations before manipulating them is commonplace, so most people never consider the benefits of using other methods. I want to show that there is merit in using other data representations to optimize applications.

The newly proposed sorting method generates a PWM signal from the analog voltage using a monostable timing circuit. This timing circuit is a modified version of a standard monostable circuit. The timing circuit is periodically set in the active state by a trigger pulse, and the timing characteristics of the circuit determine how quickly the circuit resets. These timing characteristics change depending on the applied analog voltage, resulting in the varying pulse duration required to sort the values. The sorting network is a simple network of AND and OR gates. As long as the rising edges of the PWM signals are synchronized, these AND and OR gates perform maximum and minimum operations on the pulse duration of the PWM signals. Minimum and maximum operations, when used on the same set of two values, performs a comparison on the two values by differentiating the larger and smaller pulse. This can be used to perform a comparison-based sort on the PWM signals with simple circuit components. Once the signals have been sorted, the PWM signals are converted into a binary representation using a clock-gated counter.

II. Background and Related Works

A. Alternative Information Encodings

There are many alternative ways to represent information, and each one of those ways has advantages and disadvantages when they are used. The advantage of analog is that it can represent a continuous range of values; However, its disadvantage is that it is extremely susceptible to noise. Digital, on the other hand, is resistant to signal noise, but can only represent a discrete range of values. This is important because it allows things to be optimized just by changing the representation of the data. If the advantages of using a data representation outweigh the disadvantages for a certain application, switching the data representation will improve the efficiency. This is not the first time a time-based representation of data has been used to gain advantages in energy efficiency. A group at UCSB created a hardware architecture in which all values are represented by time delays. Their architecture takes advantage of race conditions to find the shortest path through a directed acyclic graph, which is a common and extremely important problem in many fields. [2].

B. Sorting

Sorting methods are an important topic for this project, as the end result of this project is to create a sorting method using an alternative data representation that offers improved energy efficiency in some applications. The most common type of sort used today is a comparison based sort. Comparison based sorts sort data elements by comparing the elements with each other, and arrange them in an order based on their relative value. The minimum number of comparisons needed to sort a set of data using a comparison sort is on the order of $\mathcal{O}(nlog(n))$. In order to ensure the efficiency of the sorting algorithm, I had to make sure that I was comparing the signals in an efficient manner.

C. Analog to Digital Conversion

Analog to Digital Conversion is an important topic because the PWM generation necessary for the proposed sorting method must be able to compete in power efficiency with current Analog to Digital Conversion processes and hardware. In order to create a digital signal from an analog signal, a few key steps must be performed on the signal. The first step that must occur is sampling. Analog signals have a defined value at all times, while digital signals only have one value per clock cycle. Quantization is the next step required to convert an analog signal to a digital representation. Analog voltages can represent a continuous range of values, but digital values are discrete by definition. In order to convert a continuous value into a discrete value, it is quantized by blocking analog voltage regions into discrete groups which represent the same digital value.

D. Circuit Design and Simulation

A majority of this project relied on designing and testing circuits using hardware and circuits simulators. The circuit simulator used for this project was LTspice XVII. LTspice is a SPICE(Simulation Program with Integrated Circuit Emphasis) implementation that supports a front end capable of automatically generating netlists from circuit schematics. This allowed me to test the circuits without having to buy all the components and manually connect them together. The digital hardware design environment used for this project is Vivado. Vivado is a hardware design environment that supports hardware simulation and implementation for Xilinx FPGA boards. This software allowed me to to design the digital circuitry necessary to convert the PWM signals into binary.

III. Approach

My approach for designing the circuits was to identify which steps of the sorting process were the most important, and then create circuits that accomplished these steps in a simple and efficient manner. Splitting the process into steps allows for each step to be individually optimized more than if the device was made in one piece. There were three main circuits that were created were the PWM Generator Circuit, the Sorting Network, and the PWM to Binary Converter. Each circuit tackled different problems, and required different approaches to solve them.

A. PWM Generation

PWM generation is the first, and most important step in the PWM sorting method. The circuit has to be able to convert a difference in analog voltage into a difference in timing. Since timing is usually generated by RC circuits, a timing circuit to generate the PWM will need to be able to change timing characteristics in response to a change in input voltage. A linear voltage-varying resistance that charges a capacitor was my solution to this. By creating the timing circuit and connecting to a 555 timer, I was able to generate timed pulses. The 555 timer is an integrated timing circuit that can be configured to function in multiple different modes. In this project, the timer was configured to act as a monostable multivibrator. This means that when the timer receives a trigger pulse, it will be set in the active state for an amount of time dependent on the RC timing circuit. Sending continuous trigger pulses to the 555 timer causes the circuit to generate a PWM signal that represents the analog input voltage.

B. Sorting Network

The sorting network that I created is a simple and efficient solution to the sorting problem. The sorting network is comprised of AND/OR gates, which are simple logic components that use little energy compared to the circuitry needed to compare binary numbers. This circuit must be capable of performing two key operations in order to sort data: maximum and minimum. Finding the maximum and minimum of a set of analog voltages can be very difficult. However, finding the maximum and minimum pulse widths of a set of PWM signals is very easy as long as the rising edge of the signals are synchronized. Inputting synchronized PWM signals into ordinary AND and OR logic gates causes the output of the gates to be the minimum and maximum pulse widths of the input signals, respectively. By creating a network of AND/OR gate pairs, multiple PWM signals can be compared and sorted.

C. PWM to Binary Conversion

Once the data has been sorted, it still needs to be converted into digital representation in order for it to be stored on digital computers. Io convert the data from the PWM signal into a binary value, something capable of counting time is needed. A clocked digital counter is capable of doing this very efficiently. The clock input and reset lines of the digital counter also needed to be tied to the PWM signal. The input clock signal must be active when the PWM signal is high, and the reset line must trigger every time the PWM signal changes from a low value to a high value. In order to do this, a one-shot circuit must be connected to the appropriate inputs of the counter to ensure it resets and counts properly. The end result was a circuit capable of counting the number of clock cycles in the active portion of the PWM waveform, which is a binary representation of the time delay present in the original pulse.

IV. Methodology

The methodology involved in gathering the data for this project involved a lot of simulations. All circuit designs were simulated during this project, and data was taken directly from those simulations.

A. PWM Generation

The simulations of the PWM generation circuit were done in LTspice. The design schematics were created by laying out the components using the graphical schematic layout functionality of LTspice. A simulation netlist was then generated automatically from the schematic, and from there functionality and power simulations were run and data was collected in the form of .CSV files.

B. Sorting Network

The sorting network was also simulated in LTspice. Because the sorting network was composed of multiple copies of the same logic gates, only one of each gate needed to be simulated in order to gather the necessary data. I modeled the digital logic in the sorting network using CMOS technology. This data was also saved in .CSV files like the PWM generation data previously collected.

C. *PWM to Binary Conversion*

The binary conversion circuit was simulated differently than the other two circuits, because LTspice is not as effective at simulating clocked digital logic. The circuit was designed and simulated using the Vivado Digital Design Suite from Xilinx. The circuit was described using VHDL, and simulated using the tools built into Vivado. These simulations contain I/O charts showing circuit functionality, and estimations of power consumption.

V. Results

A. *PWM Generation*

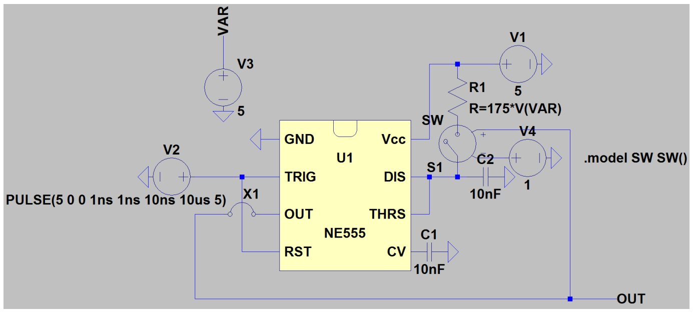


Fig. 1. Schematic for the final version of the PWM generator circuit.

1) Functionality: The functional schematic of the PWM Generator is displayed in Figure 1. The change in analog voltage clearly causes a change in the duty cycle of the PWM signal. In figure 2 we see the functional output of the circuit when an input of 0.1V is applied. As the voltage is increased, so too does the time delay, as seen in Figures 3 and 4. This continues until the value reaches a maximum of 5V, at which point the PWM reaches nearly 100% duty cycle, as seen in Figure 5.

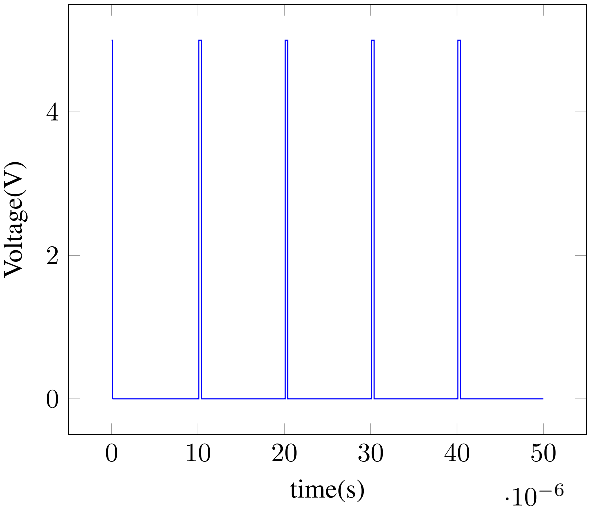


Fig. 2. Functional Output for 0.1 Volt Input. The pulse width of the PWM signal is almost as small as the reset pulse sent to the circuit.

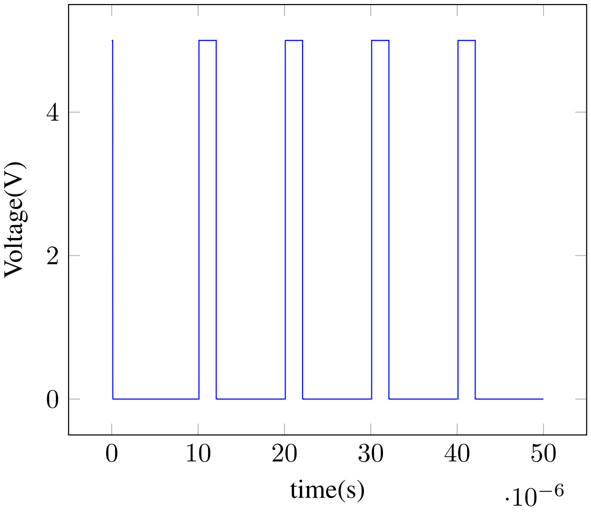


Fig. 3. Functional Output for 1 Volt Input. The pulse width of the PWM signal is slightly larger than in Figure 2, but still relatively small.

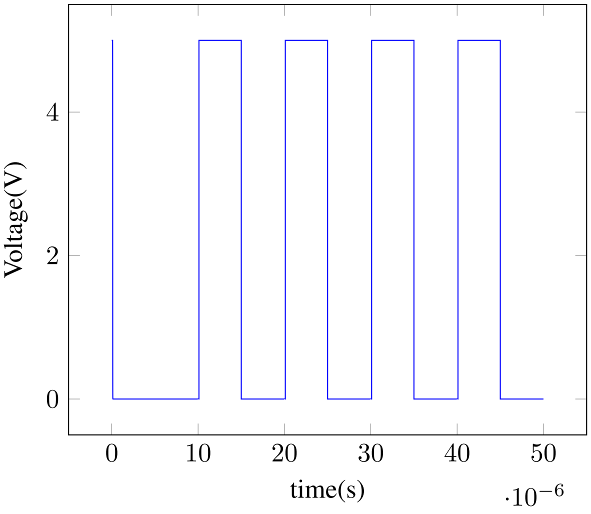


Fig. 4. Functional Output for 2.5 Volt Input. The pulse width of the PWM signal accounts for a large portion of the interval between reset pulses, giving the PWM signal an approximate 50% duty cycle.

2) Energy: There is a lot to talk about when the reviewing the power data for the PWM generations circuit. Although initial results seemed promising for high input voltages, there was a fatal flaw with the original design. As can be seen in Figure 6, the power dissipation of the circuit while it was inactive and is extremely high, and is inversely related to the analog voltage input. This is because the discharge pin of the chip is connected directly to ground while inactive. which allowed large amounts of current to flow through the variable resistor to ground. This meant that the circuit was using a large amount of power that could be saved. This led me to add a switch to the varying resistance so that it can only conduct when in the active state. This ensured that the device would stay in a low power state as long as the trigger is not pulsed, greatly reducing the total power dissipation. This can be easily seen in Figure10. The average active power dissipation of the circuit is approximately 15mW based on the simulation

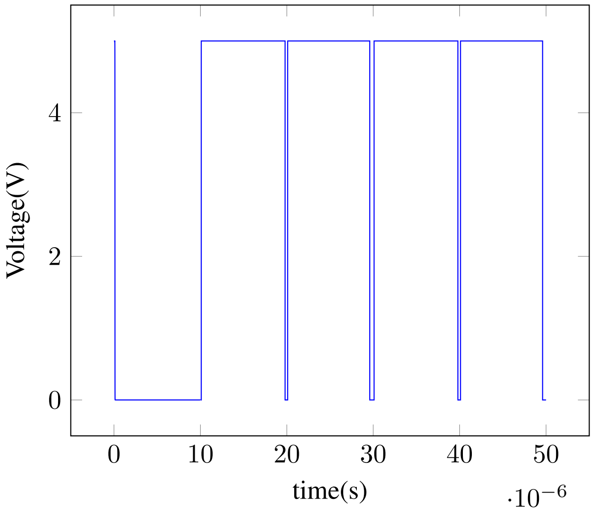


Fig. 5. Functional Output for 5 Volt Input. The pulse width of the PWM signal accounts for almost the entire interval between reset pulses. The circuit should be tuned so the pulse width is never larger than the interval between reset pulses.

data for a 2.5V input. This is less than the average power dissipation of a standard 16-bit analog to digital converter, which is 110mW according to the data sheet [1].

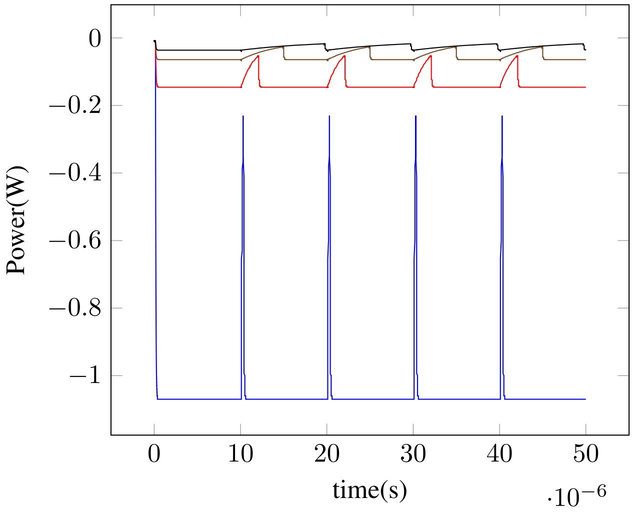


Fig. 6. Initial Design Power Comparison Graph. The power comparison shows that both the active and passive power consumption of the circuit rises as the applied analog voltage decreases.

B. Sorting Network

1) Functionality: The functional schematic of the AND/OR sorting network can be seen in Figure 16. The individual AND/OR gates function as min/max comparisons, as seen in Figures 18 and 20. Pairs of these gates are able to perform a sort of two input PWMs. Using these sorting blocks as a basis, full sorting of multiple PWM signals can be performed.

2) Energy: The power used by CMOS technology is very low, and CMOS technology is so common that even normal binary sorting circuits are implemented using CMOS technology. The power output of standard CMOS AND and OR gates can

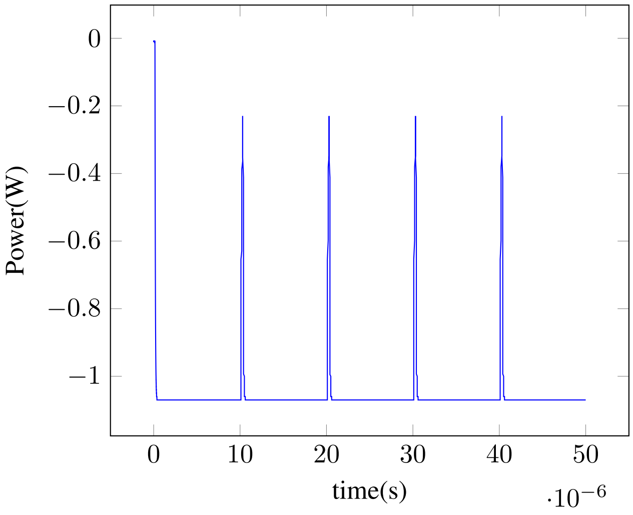


Fig. 7. Power Data for 0.1 Volt Input. The passive power consumption is very high, and the spikes caused by active power are large as well.

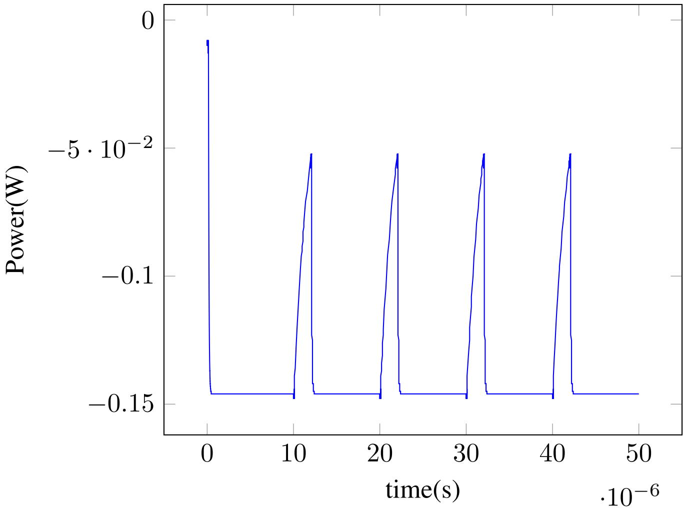


Fig. 8. Power Data for 1 Volt Input. The passive power consumption is about 10x less than in Figure 7, and the active power spikes are less pronounced.

be seen in Figures 21 and 22. The new sorting method offers a reduction in the number of transistors needed to perform the sort, thereby reducing power consumption. In order to sort two numbers in this representation, one AND and one OR gate are needed, each of which requires 6 transistors to implement. This means that 12 transistors are needed to perform a comparison of PWM signals. For a comparison-based sort using a 16-bit binary representation, each binary number must be compared using a full adder circuit, which requires 2 AND gates, two XOR gates, and an OR gate for each bit. Since XOR gates require 8 transistors to implement using CMOS technology, approximately 34 transistors per bit are needed to perform a digital comparison on a 16-bit number. This is why the proposed method of sorting offers an improvement in power dissipation over the normal sorting method.

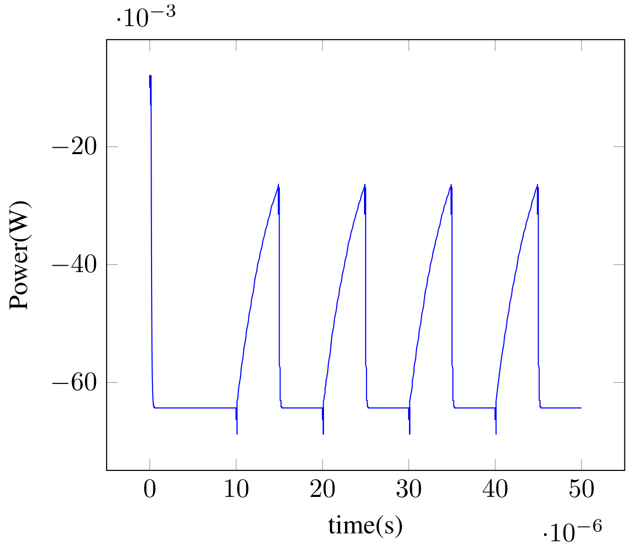


Fig. 9. Power Data for 2.5 Volt Input. The passive power consumption is about 2.5x less than in Figure 8, and the active power spikes are even less pronounced.

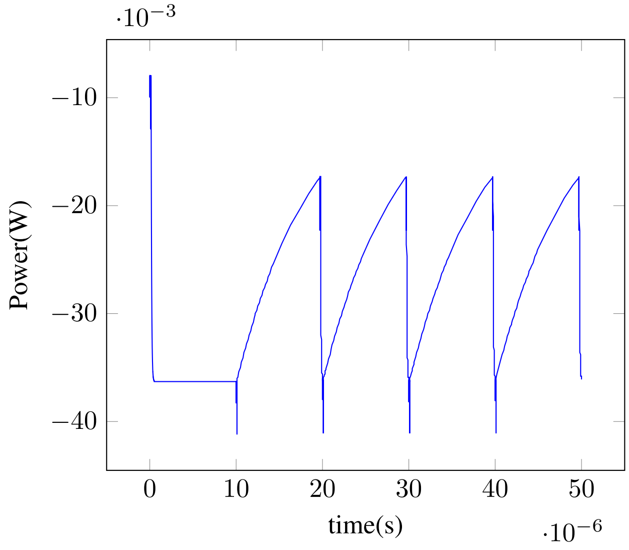


Fig. 10. Power Data for 5 Volt Input. The active power spikes are relatively small, and the circuit never remains inactive for long.

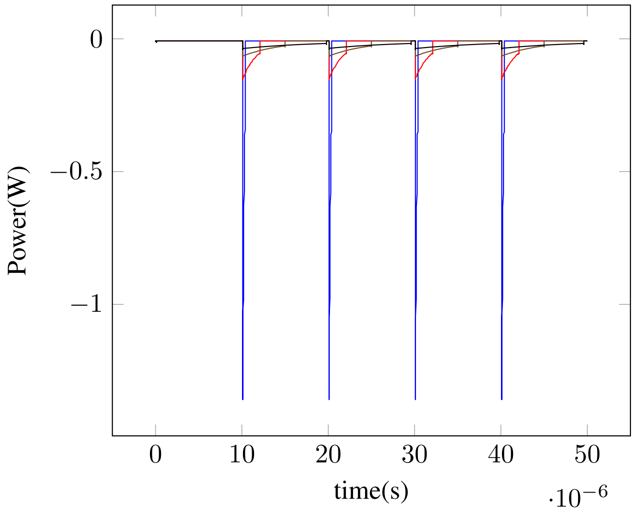


Fig. 11. Improved Power Comparison Graph. After the unwanted ground path was removed, the power comparison shows that the power spikes that occur when the capacitor is charging have variable sizes. The passive power draw is much lower, and is consistent for all input values.

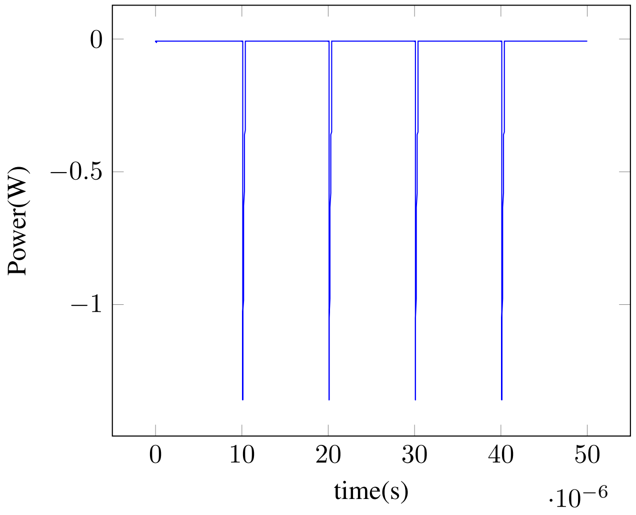


Fig. 12. Improved Power Data for 0.1 Volt Input. The active power spikes are large and pronounced, while the passive power is very low.

C. *PWM to Binary Conversion*

1) Functionality: The functional schematic of the PWM to Binary Converter can be seen in Figure 23. In Figure 24, the counter can be seen to reset when the PWM value transitions from low to high voltage. The counter then begins counting the rising edges of the clock signal until the PWM transitions back from high to low voltage, at which point, the data ready line is pulsed to indicate a new value has been generated.

2) Energy: The energy consumption of the PWM to binary converter is shown in Figure 25. The power dissipation for the I/O of the Xilinx board can be ignored, yielding a power use of 42mW for the circuit. By adding the power dissipation of the generator circuit and binary to digital conversion circuit, the total power consumption for the conversion circuits is found to be 57mW. This is slightly more than half of the 110mW power dissipation of a standard 16-bit ADC [1].

VI. Conclusion and Future Works

This project shows that there are alternative ways to sort data that are more energy efficient than the commonly accepted method of analog to digital conversion coupled with binary data sorting. By converting analog data into a PWM signal before sorting, the amount of power needed to sort the data is drastically reduced. The sorting of PWM signals is inherently more energy efficient than the binary sorting because the number of transistors needed to compare the PWM signals is far lower than the number needed to compare binary numbers.

In the future, I would like to refine this design and assemble a functional version of the circuit instead of just using simulations. I also have been working on an idea that would serialize the PWM signals, so that the sorted values can be

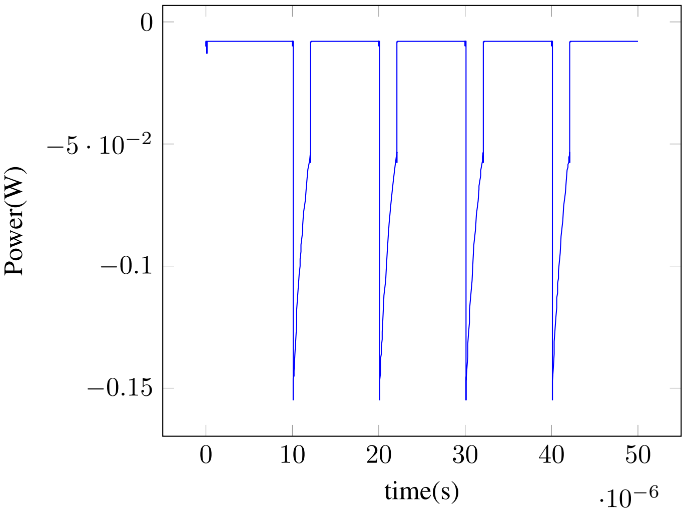


Fig. 13. Improved Power Data for 1 Volt Input. The active power spikes are still large, but smaller than in Figure 12.

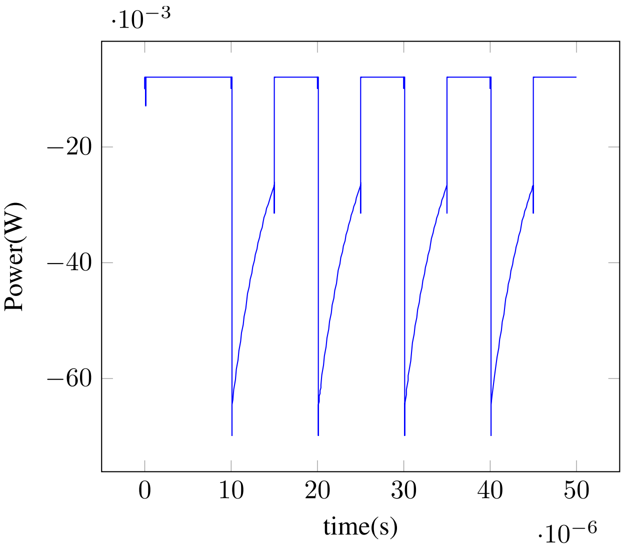


Fig. 14. Improved Power Data for 2.5 Volt Input. The active power spikes are shallower and wider.

represented by only one signal, greatly decreasing the amount of circuitry needed to convert all of the signals back to a binary representation. I would need to keep researching this idea, and there a possibility of publishing a paper about it.

VII. References

[1] 16-bit 250ksps serial cmos sampling analog-to-digital converter data sheet. http://www.ti.com/lit/ds/slas462d/slas462d.pdf. Online. Accessed 6/4/18.

[2] Advait Madhavan, Timothy Sherwood, and Dmitri Strukov. Race logic: A hardware acceleration for dynamic programming algorithms. In *Proceeding of the 41st Annual International Symposium on Computer Architecture*, ISCA ’14, pages 517–528, Piscataway, NJ, USA, 2014. IEEE Press.

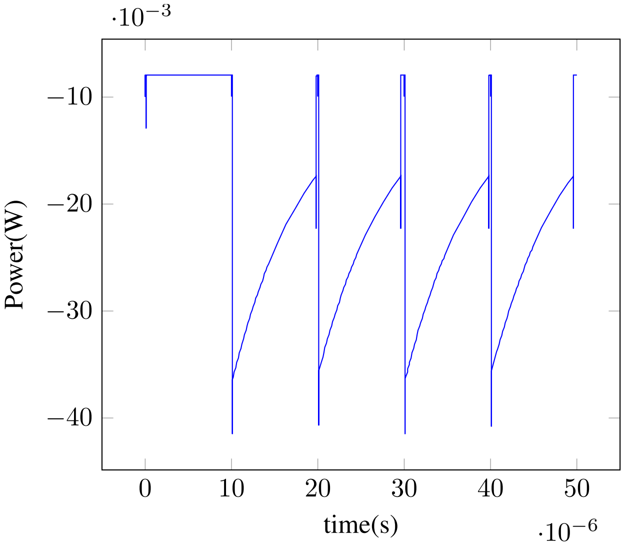


Fig. 15. Improved Power Data for 5 Volt Input. The active power spikes are very small, and the circuit is almost never in the passive power mode.

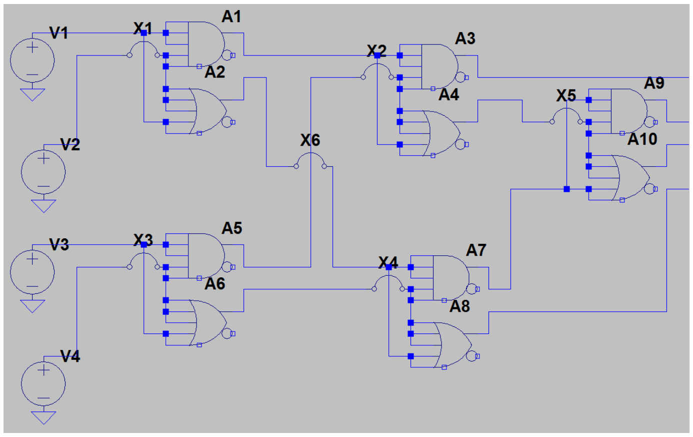


Fig. 16. Schematic for the final version of the PWM sorting network.

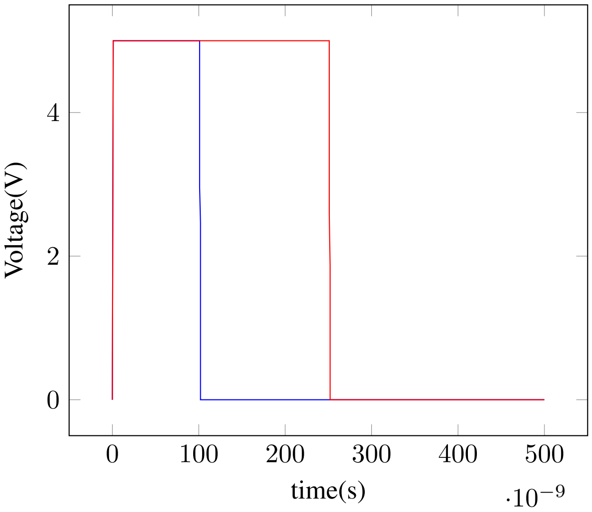


Fig. 17. Functional Input of OR gate. The pulse widths of the two input signals are 100ns and 250ns, respectively.

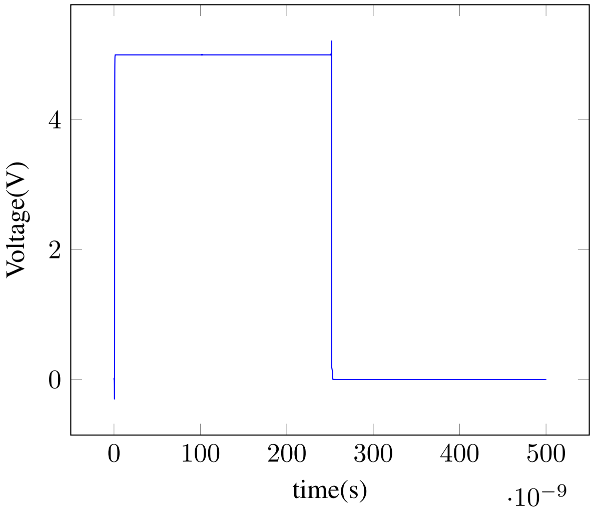


Fig. 18. Functional Output of OR gate. The OR gate functions as a maximum operation, propagating the signal with the longest pulse width through.

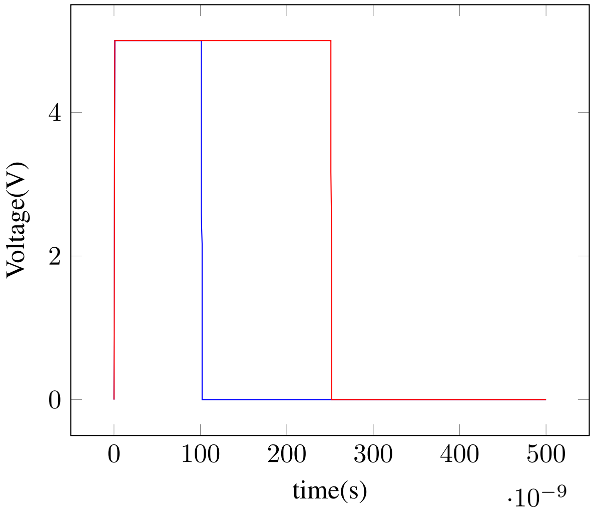


Fig. 19. Functional Input of AND gate. The pulse widths of the two input signals are 100ns and 250ns, respectively.

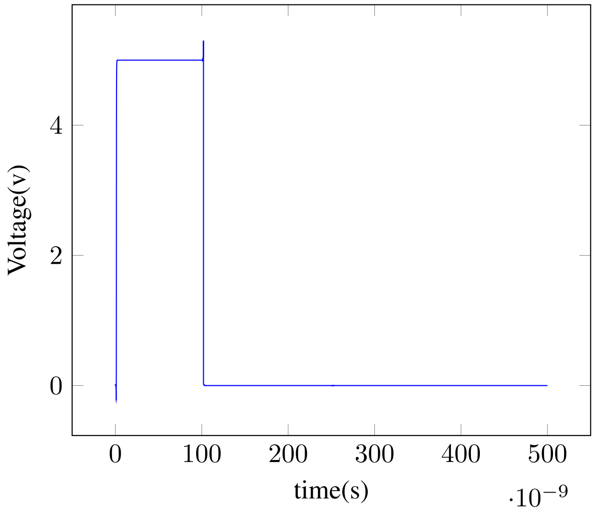


Fig. 20. Functional output of AND gate. The AND gate functions as a minimum operation, propagating the signal with the shortest pulse width through.

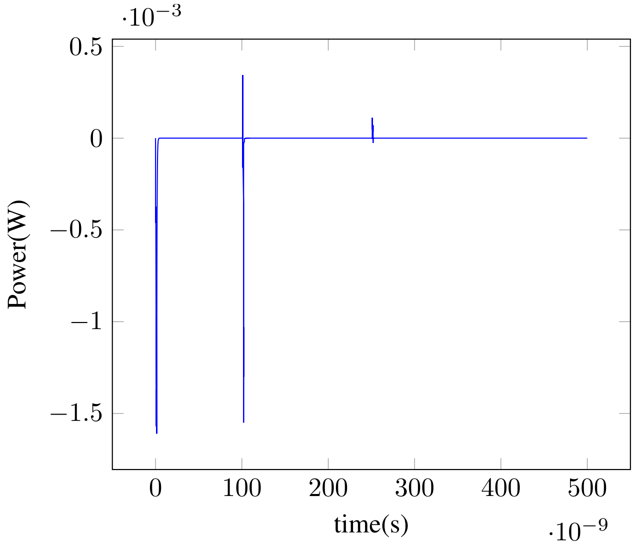


Fig. 21. Power data for OR gate. The CMOS technology in the OR gate consumes almost no power except for when the gate output switches logical state.

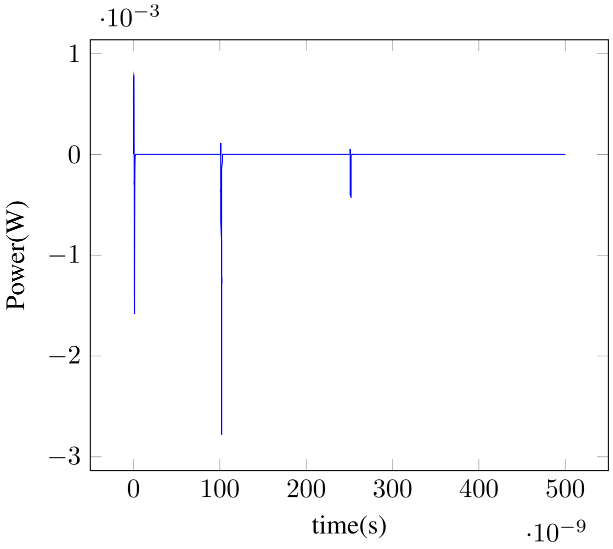


Fig. 22. Power data for AND gate.The CMOS technology in the OR gate consumes almost no power except for when the gate output switches logical state.

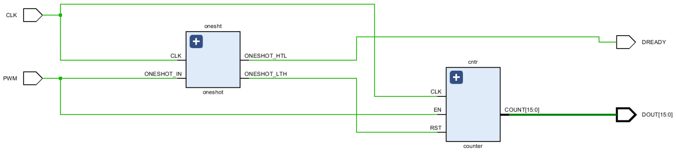


Fig. 23. Schematic of the PWM to Binary conversion circuit. The circuit was created from a one-shot generator and a counter with reset.

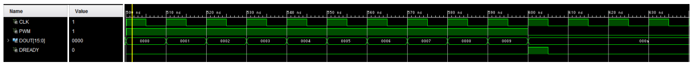
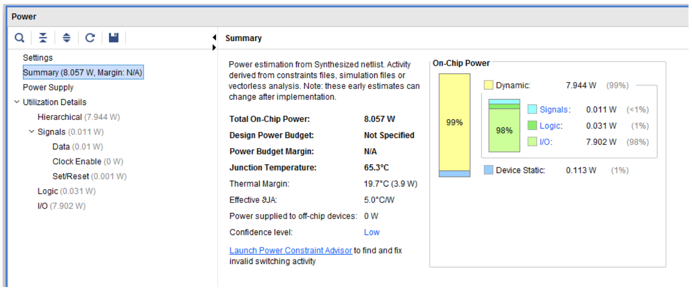


Fig. 24. Functional simulation of the PWM to Binary Conversion circuit. The conversion circuit successfully counts the clock cycles in the PWM signal pulse region.


Power

Settings
Summary (8.057 W, Margin: N/A)
Power Supply
Utilization Details
Hierarchical (7.944 W)
Signals (0.011 W)
Data (0.01 W)
Clock Enable (0 W)
Set/Reset (0.001 W)
Logic (0.031 W)
I/O (7.902 W)

Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	**8.057 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**65.3°C**
Thermal Margin:	19.7°C (3.9 W)
Effective ϑJA:	5.0°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

Launch Power Constraint Advisor to find and fix invalid switching activity

On-Chip Power

Dynamic:	7.944 W	(99%)
Signals:	0.011 W	(<1%)
Logic:	0.031 W	(1%)
I/O:	7.902 W	(98%)
Device Static:	0.113 W	(1%)

Fig. 25. Power Estimation of the PWM to Binary Conversion circuit. The I/O power can be ignored, giving a power consumption of approximately 42mW.