Achieving Nanosecond Timing with the Vernier Method

Rebecca E. Cooper
Spring 2009

Advisors:
Dr. Elizabeth George and Dr. Paul Voytas
Department of Physics, Wittenberg University

ABSTRACT

Many subatomic processes occur in a small time frame, on the order of nanoseconds. We report on a project to create a circuit that can measure time intervals accurately on such a scale. To achieve this we used the Vernier Method, which is based on coincidence between two oscillators of slightly different frequencies. For accurate timing, the oscillators needed to have a stable, clean signal. The frequency needed to be in the MHz range, producing a pulse period of ~100 ns, and the pulse width needed to be narrow enough, on the order of a few nanoseconds, to produce only one coincidence. Relaxation oscillators based on unbuffered inverters in an RC circuit were used to generate square pulses at the required frequency. To shorten the pulses, we used a high-speed comparator circuit that created two pulses from the input pulse, one of which was inverted and delayed relative to the other, and an AND gate that combined the two, thus creating a shorter pulse. We have achieved a frequency of 4.5 MHz (a pulse period of 220 ns) and a width of 8 ns. Our circuit was tested by creating two input signals (a start signal and a stop signal delayed 150 ns by a 100 ft cable) with a Hewlett Packard 8082A pulse generator. Each signal triggered one of the oscillators to oscillate and the coincidence between them was observed. A flip-flop and counter can then be used to measure the time interval between start and stop.

1 INTRODUCTION

On the subatomic level, many reactions and decays occur on the order of nanoseconds. An example of such a decay is positronium decay. Positronium is created when an electron and its anti-particle, a positron, orbit each other. The positron may come from the radioactive decay of certain
atoms, such as potassium-40 or nitrogen-13. The electron and positron will orbit each other either as parapositronium, where their spins are anti-parallel, or as orthopositronium, where their spins are parallel. Parapositronium has a mean life of \(~0.1\) nanoseconds and orthopositronium has a mean life of \(~100\) nanoseconds [1]. Positronium decays through the process of converting the rest energy of the electron and the positron into energy of emitted photons. There are typically two or three photons.

Another example of subatomic decay is muon decay. A muon is an elementary particle with a charge that is the same as that of the electron, but whose rest mass is 207 times that of the electron [2]. The muon decays into an electron and two neutrinos. The muon’s mean life is about 2.2 \(\mu\)s [2].

The time for each of these decays can be measured by two events. The two events are “start” and “stop.” The start event is when, in the case of positronium decay, the positron enters a detector. Once inside the detector, the positron interacts with an electron and creates positronium. All of the positronium's mass energy turns into photons and there is the possibility of some of the photons entering a second detector. The stop event is when one of the photons enters the second detector. In the case of the muon decay, the start event is when the muon enters a detector. The muon will then decay inside the detector into an electron and two neutrinos. The stop event in this case is when the electron enters the second detector. We wish to measure the time between these two events. One way to do this is to directly count the nanosecond intervals. This requires a GHz clock and counter, which is not easy or cheap to do. Instead, several methods have been developed to make this measurement easier. A method we chose to use is the Vernier Method [3].

2 BACKGROUND

The Vernier Method is one way of achieving better precision in timing short intervals. This method makes use of two oscillators of slightly different frequencies. Our start pulse triggers the first oscillator to begin cycling with some period, \(T_1\). The stop pulse, at some later time \(\tau\), triggers the second oscillator to begin cycling with a period \(T_2\). Because the two oscillators are cycling through at slightly different periods, the oscillator with the larger period will “catch up” with the oscillator with the smaller period, as shown in Figure 1. Eventually there will be a coincidence, where the two oscillators will have a cycle where they overlap. This coincidence can be used to stop timing. Figure 1 shows this process and the time period, \(\tau\), that we are interested in finding. In the case that is represented in Figure 1, the number of cycles, \(n\), each oscillator goes through before coincidence is the same. This is because the second oscillator begins to oscillate before the second period of the first oscillator. The time interval between which the first oscillator starts and when it reaches a coincidence...
with the second oscillator can be written as \( nT_1 \). The time interval in which the second oscillator starts and when it reaches a coincidence with the first oscillator can be written as \( nT_2 \). The difference between these two time intervals is \( \tau \), the time interval between the start pulse and the stop pulse.

![Diagram of time expansion using the vernier principle.](image)

Figure 1 [3] – Time expansion using the vernier principle.

Taking the difference between these two time intervals gives us

\[
\tau = n(T_1 - T_2)
\]

(1)

where \( \tau < T_1 \). By counting \( n \) and knowing \( T_1 \) and \( T_2 \), we can find \( \tau \). With \( \tau < T_1 \), we can measure time periods up to the value of the period of the first oscillator, which means we want our first oscillator to have a period in the range of 100 ns to possibly a few μs, based on the time it takes positronium and muons to decay.

To create a signal with high resolution, leading to a more accurate measurement, we need the frequencies of our oscillators to be stable for accuracy and for their frequency difference to be as small as possible for precision. Our goal is to create two frequencies that are different by \( \sim 1\% \). The minimum time that we can measure is \( \tau_{min} \), which is when \( n = 1 \). We can express the minimum time interval as:

\[
\tau_{min} = T_1 - T_2
\]

We can write this expression in terms of frequencies:

\[
\tau_{min} = \left( \frac{1}{f_1} - \frac{1}{f_2} \right)
\]

\[
\tau_{min} = \left( \frac{f_2}{f_1 f_2} - \frac{f_1}{f_1 f_2} \right)
\]
Because our frequencies are very close in value, we can say $f_1 \sim f_2 \equiv f$, giving us

$$\tau_{\text{min}} = \left( \frac{\Delta f}{f^2} \right)$$

We can then rewrite $f^2$ in terms of frequency and period:

$$\tau_{\text{min}} = \left( \frac{\Delta f}{f} \right) T$$

where $f$ and $T$ are the frequency and period of each oscillator, respectively. From this expression, if we make our period $T = 100$ ns and $\frac{\Delta f}{f} = 1\%$, we get $\tau_{\text{min}} = 1$ ns, which is the minimum time interval that we are trying to measure.

For the Vernier Method to work with our timing requirements, we need two stable oscillators with $\frac{\Delta f}{f} = 1\%$ and frequencies in the MHz range so that we will get $T \sim 100$ ns. When the start and stop pulses occur, we need a way for those pulses to trigger each oscillator to start pulsing. Once the oscillators reach a coincidence, we need to be able to reset their oscillations when another start pulse occurs. A counter is necessary to keep track of the number of cycles each oscillator has gone through before their coincidence. We also need our components to be temperature independent, allowing for stable signals, and our signals to be clean for more accurate reading.

Finally, the oscillators’ pulses need to have a width such that there will be a coincidence so the Vernier method will work. Figure 1 shows pulses with zero pulse width, which means there is only one occurrence of a coincidence, when the two pulses are directly lined up with each other. If there is too much width to the pulses, that is, if $\Delta T < (2*\text{pulse width})$, there may be times before and after a complete coincidence where the pulses overlap. In reality, there will be width to our pulses, creating a limit in what resolution is achievable. The narrower we can make our signals, the better the resolution. With better resolution, we will be able to better distinguish between the different $\tau$. We want our pulses to have a width large enough so that they will reach one coincidence.
3 EXPERIMENTAL SETUP

A functional block diagram of our experimental setup is shown in Figure 2. We have two pulses, a start pulse and a stop pulse. Each pulse is fed into an oscillator through a JFET p-channel transistor, which will synchronize the oscillators. The two oscillators, with slightly different frequencies, are each an input to a logic gate, whose output tells us whether or not there is a coincidence. Before the oscillators reach a coincidence, one input to the logic gate is high while the other input is low (a description of logic levels can be found in Appendix 9.1). This produces a unique output. When the oscillators reach a coincidence, each input to the logic gate is high, changing the logic level of the logic gate’s output, letting us know that a coincidence was reached. To keep track of the number of cycles, n, each oscillator goes through before coincidence, there is a counter that counts the number of cycles. To stop the counter once the oscillators reach a coincidence, the signal from the AND gate is fed into the reset input of a set-reset flip-flop. When there is a coincidence between the two oscillators, the flip-flop is reset and its output is then high, which is fed into a NAND gate. The NAND gate’s output becomes a low, according to its truth table found in Appendix 9.2, and the counter stops counting. The next start pulse will set the flip-flop’s output back to low, which enables the counter to start counting again.
4 DATA AND ANALYSIS
4.1 Relaxation oscillators

We built two relaxation oscillators as shown in Figure 3. Each oscillator consists of three MC74HC04A unbuffered inverters, the third acting as a buffer, and an RC circuit. The unbuffered inverters were used instead of buffered inverters because the unbuffered inverters had a shorter propagation time, allowing for higher frequencies for our oscillators. The resistor $R_2$, whose value is $R_2 = 10R_1$, allows only $1/10^{th}$ of the current to pass through that part of the circuit. This allows for most of the current to flow in the RC circuit, but still lets some of the current flow in the left side of the circuit so that the U1A inverter can change logic levels. The signal at the output of each inverter is, as expected, the opposite of its input. When the input of the U1A inverter is a logic high, the input to the U2A oscillator is a logic low, with its output a logic high. This means that a signal comes out of the second inverter and charges up the capacitor from the logic high to the logic low of the second inverter. The voltage level at the point P starts off at a logic high. As the capacitor charges in the counterclockwise direction, the voltage level at that point decreases. Assuming that the voltage level at P is the same as the voltage level at the input of the U1A inverter, due to the nature of the unbuffered inverter, we can say that once the voltage level at P reaches ~2.0 V, the level at P switches to a logic low and the voltage level at the U1A input switches to a logic low. The levels at the inputs of the other inverters then switch as well. The capacitor will then begin to discharge and current will travel in the clockwise direction. The process then repeats. The frequency of the oscillator depends on the values of $R_1$ and C. We chose our resistor to have a value of $R_1 = 500 \, \Omega$ and our capacitor to have a value of

![Figure 3](image)

**Figure 3 [4] – RC Oscillator**
C = 23.5 pF so that our frequency would be in the MHz range. We found our frequency from the relationship [4]:

$$f = \frac{1}{2.3R_1C}$$

This equation gives us a frequency of $f = 37$ MHz; however, due to extra stray capacitance in our breadboard, our actual frequency is $f = 4.5$ MHz. Our second oscillator has a frequency of $f = 4.3$ MHz with resistor and capacitor values of $R_1 = 500 \, \Omega$ and $C = 20$ pF.

4.1.1 Synchronizing the oscillators

A JFET J176 p-channel transistor switch was used to control the oscillations of our oscillators, as shown in Figure 4. Without the transistor, we are just left with our original relaxation oscillator, free to oscillate. Adding the transistor gives us control over the oscillations of our oscillator. The drain of the transistor connects to the RC circuit to keep it from charging or discharging and the source of the transistor is connected to ground. When the input to the transistor (our start or stop pulse) is low, the transistor is on, pulling point A of the circuit to ground and preventing the oscillator from oscillating. When the input pulse is high, the transistor is turned off and the voltage at point A floats, allowing the oscillator to oscillate freely.

We wish for the oscillator to oscillate between when there is an input signal and when there is a coincidence. Because the transistor is a p-channel, it is off only when the input signal is high, so we will only have oscillations when we have an input signal and no oscillations during the time between the input signal and a coincidence. An inverter was used to solve this problem. The inverter inverts the incoming start signal so that the input pulse that comes to the transistor is low when the start signal is high, which prevents the oscillator from oscillating until the start signal goes low, at which point the transistor is turned off and the oscillator begins to oscillate. Therefore, the oscillator is triggered to begin oscillating at the end of the start or stop signal.
4.1.2 Different frequencies

The Vernier method depends on the two oscillators having different frequencies. Our two oscillators were originally built on the same breadboard. When we tried to make their frequencies slightly different by changing the value of the capacitance of our second oscillator, the frequencies locked to a constant relative phase with identical periods. We managed different frequencies at $\frac{\Delta f}{f} = 8\%$. Making the frequency difference any smaller than $\frac{\Delta f}{f} = 8\%$ resulted in their frequencies locking. We suspected that the oscillators were interfering with each other through the breadboard circuitry, so we moved one oscillator to another breadboard. This gave better results, bringing the frequency difference down to $\frac{\Delta f}{f} = 4\%$, with $f_1 = 4.5$ MHz and $f_2 = 4.3$ MHz.

4.2 Short pulses

As mentioned earlier, short pulses are very important in the Vernier method. Figure 1 shows pulses with zero width. This condition results in a single coincidence. If the pulses have too much width, there may be instances before and after this exact coincidence where the two pulses still overlap. This would result in more than one coincidence for each start or stop cycle. Our goal is to have only one coincidence, which means that there is only one time when both signals overlap. We used a high-speed comparator circuit to shorten our pulses, as seen in Figure 5. Our input pulse, coming from an oscillator, enters the first comparator, $IC_1$, inverting the input pulse. The output then drives two RC timing circuits, each with slightly different time constants. One of the signals enters the $IC_2$ comparator in the non-inverting input. The inverting input of the comparator is held at 2.5 V. When
the input voltage is greater than this reference voltage, $V_r = 2.5$ V, then the output will be high. When the input voltage is less than the 2.5 V, then the output will be low. The signal going into the $IC_3$ comparator enters the inverting input, with the non-inverting input held at $2.5$ V. When the input voltage is less than $2.5$ V, the output will be high and when the input voltage is greater than $2.5$ V, the output will be low. The RC circuits produce an exponentially saturating voltage with a time constant determined by RC. Because the comparator output does not switch until $V_- > V_+$ or $V_- < V_+$, the comparator with the smaller time constant will have its input reach $2.5$ V first, causing its output to switch first. The other comparator will then switch at a later time, dependent upon how long it takes for its slower input to reach $2.5$ V. The delay can be varied by varying the RC constant of each RC timing circuit.

The output of each comparator is an input of an AND gate. The two input signals are inverted with respect to each other and slightly delayed from each other, as traces B and C in Figure 6. There is a time interval when each signal is high at the same time. Because of the AND gate logic, when both signals are high the output is high. The short overlap between the two signals creates a short output signal, which can also be seen in Figure 6. The smallest width we achieved while keeping an amplitude of 5 V, which is required for our AND gate to register a logic high, was 8 ns. This gave us one overlap during a coincidence. The results can be seen in Figure 7.

Figure 5 [5] – A short-pulse generator using comparator circuits.
Figure 6 [5] – Wave-forms from the short pulse generator. Signal A is the input signal (width ~ 50 ns), signals B and C are the two input signals to the AND gate, and signal D is the output signal (width ~ 15 ns).

Figure 7 – Shortened pulses. The top pulses are before shortening, with a width of ~ 150 ns. The bottom pulses are after being shortened, with a width of ~ 8 ns.
4.3 The counter

A counter is needed to keep track of the number of cycles each oscillator goes through before a coincidence. We used a package TC74HC4040 DIP counter with 12 outputs. Each output represents a value of $2^x$, where $x$ is the output number, starting with zero. So output #0 has a value of $2^0 = 1$, output #1 has a value of $2^1 = 2$, and so on, up to $2^{12} = 4096$. Adding up the values of each output gives us the highest value to which it can count, which is 8,191. This means that

$$\tau = n\Delta T$$

$$= n\left(\frac{1}{4.3 \text{ MHz}} - \frac{1}{4.5 \text{ MHz}}\right)$$

$$= n(10 \text{ ns})$$

We have 10 ns resolution and can count up to 84 $\mu$s. This is far beyond our expectations. Because $\tau < T$, we only need to count up to the value of the period of our oscillator, which is $T \approx 220$ ns, so our counter meets our requirements. When the input pulse reaches a trailing edge of the square pulse signal the outputs count up by one in binary. The outputs are reset to zero when the clear input receives a logic high signal.

We have found a way to synchronize our oscillators and now need a way to stop counting pulses when the oscillators reach a coincidence. A set-reset flip-flop latch was used, as shown in Figure 8. Our set input $S$ is controlled by the start signal and our reset input $R$ is controlled by the output of our coincidence NAND gate. We chose our logic gate to be a NAND gate because the reset input of our flip-flop is triggered when it receives a low input signal. Because the output of the NAND gate is low when both inputs are high, the reset input of the flip-flop is triggered when the two oscillators reach a coincidence.

When we get a start signal, that signal is then inverted and goes into the set input of the flip flop. When $S$ is a logic low (with the start pulse a logic high), the output of the flip flop is set to a logic high and remains high until it is reset. The output of the flip flop is then one of the inputs to a NAND gate. The other input to the NAND gate is controlled by one of our oscillators. According to NAND gate logic, that can be found in Appendix 9.2, when one of the inputs is high, the output is determined by what the logic level of the second input is. Because the second input is controlled by an oscillator, that input will oscillate between high and low, thus the output will then oscillate between low and high, allowing our counter to count. When our oscillators reach a coincidence, both inputs to the coincidence
NAND gate will be high, making the output a low. That output is fed into the reset input. When $\bar{R}$ is low, the output of the flip flop is reset to low, making one of the inputs to the NAND gate low. According to the truth table for the NAND gate, when one input is low, the output is high, no matter what the logic level of the second input is. Because the output of the NAND gate is held at a logic high, the counter no longer has counts to count. The next start pulse will set the output of the flip-flop back to high, allowing the counter to count again.

![Set-Reset Flip-Flop Diagram](image)

**Figure 8 – Set-Reset Flip-Flop**

4.4 A clean signal

The material of the capacitor and the resistors affects the susceptibility to noise and the stability (independence of temperature) of the signal. We started with carbon film resistors and regular ceramic capacitors. Our signal was noisy and temperature dependent, which made it unstable and difficult to interpret. We chose to change the type of capacitor and resistor so that its material was less temperature dependent, allowing for a more stable signal. We changed to metal film resistors, which have better tolerance and a smaller temperature coefficient. They are less dependent on temperature and also reduce the noise of our signal. We changed to C0G ceramic capacitors, which have a capacitance more nearly independent of voltage, frequency, and temperature, and they have low power dissipation. Changing the material of the capacitor and the resistors resulted in a cleaner, more stable signal.
5 CIRCUIT TESTS

Our circuit was tested with start and stop signals created by a Hewlett Packard 8082A pulse generator. Pulses were created with a period of $T \sim 1 \mu$s and a width of $\sim 50$ ns. The stop pulses were then delayed through a long cable. Because we know that

$$t = \frac{d}{v}$$

and the speed of our pulse in the cable is $\frac{2}{3}$ the speed of light, $\frac{2}{3}c = 2 \times 10^8 \text{ m/s}$, 1 ft (0.3 meters) of cable would create a time delay of

$$t = \frac{0.3 \text{ m}}{2 \times 10^8 \text{ m/s}}$$

$$= 1.5 \times 10^{-9} \text{ s} = 1.5 \text{ ns}$$

From this, we knew that 100 ft would give us 150 ns of delay, which is approximately what we wanted based on the lifetime of positronium. The results from this test are in Figure 9. There are oscillations while the start pulse is low. When the start pulse goes high, the oscillator is pulled to ground. Because the start pulse is so short and occurs while the oscillator is in the “low” part of its cycle already, it is difficult to see the oscillator stop completely; however, where the start signal is high, the oscillator is pulled to ground and is then free to oscillate when the start signal goes back to a logic low.

Figure 9 – A start pulse with oscillations from oscillator 1.
Figure 10 shows the output of our coincidence AND gate for the test conditions described above. An AND gate was used here instead of a NAND gate so that we could easily interpret our coincidence signal. This verifies that a single coincidence is produced for this combination of start and stop pulses.

6 CONCLUSION

The Vernier Method has proven to be effective for making accurate measurements on the nanosecond scale. Using two oscillators at slightly different frequencies, 4.3 MHz and 4.5 MHz, gave us a resolution of ~ 10 ns, which is acceptable for the time scales we’re dealing with for positronium and muon decay. We achieved a difference in frequency of $\frac{\Delta f}{f} = 4\%$, though our goal is still $\frac{\Delta f}{f} = 1\%$.

Our next step in reaching our goal of $\frac{\Delta f}{f} = 1\%$ is to take the oscillators off of the breadboards and put them on a circuit board. We will also put each oscillator in a shielding box to further reduce the interference of the two oscillators’ signals with each other. Narrow pulses of ~ 8 ns provided us with one coincidence for every start or stop cycle, which can be seen in Figure 10, and an amplitude of 5 V allowed for our logic gates to read the signal as a logic high. It remains for us to integrate the flip-flop and counter into the circuit. So far, we have only tested these separately. We also need to interface the counter to a microprocessor so the counter readings can be stored and processed.

7 ACKNOWLEDGEMENTS

I would like to thank Dr. George and Dr. Voytas for their time and guidance throughout my
research. I would also like to thank Dr. Williams for his idea of the comparator circuit for narrowing our oscillator pulses. Finally, I would like to thank Dr. George, Dr. Fleisch, and Dr. Parker for being on my thesis committee.

8 REFERENCES

[2] ISIS, What is a Muon?. http://www.isis.rl.ac.uk/aboutIsis/index.htm?content_area=/aboutIsis/whatisaMuon.htm&side_nav=/aboutIsis/aboutisisSideNav.htm&


9 APPENDIX
9.1 Logic Levels
In the area of logic, there are two values for a signal: “high” and “low.” When a signal is low, it is nominally at zero volts. When a signal is high, it is nominally at five volts. There are ranges for each of the levels, shown in Figure 11.
Figure 11 - Voltage ranges and threshold voltages for TTL [6].

9.2 Logic gates and Truth Tables

**AND**

<table>
<thead>
<tr>
<th>Input 1</th>
<th>Input 2</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>H</td>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
<td>L</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
<td>L</td>
</tr>
</tbody>
</table>

**NAND**

<table>
<thead>
<tr>
<th>Input 1</th>
<th>Input 2</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>H</td>
<td>L</td>
</tr>
<tr>
<td>H</td>
<td>L</td>
<td>H</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
<td>H</td>
</tr>
</tbody>
</table>
### Inverter

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>L</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
</tr>
</tbody>
</table>

### Set-Reset Flip-Flop

<table>
<thead>
<tr>
<th>$\bar{S}$</th>
<th>$\bar{R}$</th>
<th>$Q$</th>
<th>$\bar{Q}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>L</td>
<td>Indefinite</td>
<td>Indefinite</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
<td>H</td>
<td>L</td>
</tr>
<tr>
<td>H</td>
<td>L</td>
<td>L</td>
<td>H</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
<td>No change</td>
<td>No change</td>
</tr>
</tbody>
</table>

### Comparator

If $V_- > V_+$, then $V_{out} = L$.
If $V_- < V_+$, then $V_{out} = H$. 