12-1-2015

Fast Transient Digitizer and PCB Interface

Kevin Thomas Buck
University of Nevada, Las Vegas

FAST TRANSIENT DIGITIZER AND PCB INTERFACE

By

Kevin Buck

Bachelor of Science in Electrical and Computer Engineering

Oregon State University

2008

A thesis submitted in partial fulfillment

of the requirements for the

Master of Science in Engineering – Electrical Engineering

Department of Electrical and Computer Engineering

Howard R. Hughes College of Engineering

The Graduate College

University of Nevada, Las Vegas

December 2015

UNLV | GRADUATE COLLEGE

Thesis Approval

The Graduate College
The University of Nevada, Las Vegas

November 24, 2015

This thesis prepared by

Kevin Buck

entitled

Fast Transient Digitizer and PCB Interface

is approved in partial fulfillment of the requirements for the degree of

Master of Science in Engineering – Electrical Engineering
Department of Electrical and Computer Engineering

R. Jacob Baker, Ph.D.
Examination Committee Chair

Kathryn Hausbeck Korgan, Ph.D.
Graduate College Interim Dean

Brendan Morris, Ph.D.
Examination Committee Member

Peter Stubberud, Ph.D.
Examination Committee Member

Yi-Tung Chen, Ph.D.
Graduate College Faculty Representative

ABSTRACT

This thesis outlines the design, layout and characterization of an application specific integrated circuit (ASIC) and a printed circuit board (PCB) designed to discretize an input signal and read out the data at a user desired interval to replicate the input signal at a lower frequency or pulse rate with no quantization error. The ASIC for this task utilizes 128 individual capacitive storage cells to capture data at approximately 200 ps intervals, thus the chip can capture approximately 25 ns of an input signal. The read out rate can be adjusted to the desired rate by the user by changing the rate of the input clock signal to the PCB. The circuit dissipates no appreciable DC power and uses a 5 V power supply.

Each of the capacitive cells on the chip is driven by a transistor receiving a buffered (delayed) trigger signal, the trigger signal controls the data capture process. On the falling edge of the trigger the transistor shuts off and the charge from the input signal is captured on the capacitor. The clock signal is propagated to the next cell through another buffer which provides the necessary delay. The output stage is controlled by a PMOS transistor that is driven by the capacitor; this transistor is fabricated in a separate n-well to eliminate the body effect. The PMOS transistor is connected to the output through an NMOS switch that is connected to a pullup resistor, the switch is controlled with a select signal that allows the data to be read out. During the read out process the select signal is driven high for each stage sequentially, when the switch is activated the charge stored on the capacitor drives the PMOS transistor and sets the output current creating a current controlled voltage drop across the pullup resistor that replicates the input signal.

DEDICATION

For my wife Marzieh, thank you for your support. I love you.

TABLE of CONTENTS

LIST of FIGURES

CHAPTER 1: INTRODUCTION AND MOTIVATION

1.1 BACKGROUND

In modern electronics it is often necessary to capture and disseminate information from very high frequency analog waveforms or short duration transient signals. These high speed signals arise often in scientific instrumentation and electro-optic interfaces. In cases where the original analog signal is a very high frequency or a very fast transient it can be useful to discretize the data and read out the signal at a reduced rate. Discretization is the process of capturing a continuous (analog) input signal and storing discrete values of the signal at regular intervals. The process of discretization is most often seen in analog to digital converters (ADCs). After discretization in an ADC the data undergoes a process called quantization in which each data point is assigned a value with a finite precision. Quantization has the undesirable effect of information being lost due to the effect of the data being rounded up or down to the nearest value.

1.2 MOTIVATION

In order to prevent significant amounts of data from being lost from very fast signals the resolution of the quantization must be extremely high or alternatively the quantization process can be skipped. After discretization the stored information can be used directly to reconstruct the input signal at a slower rate with less information lost than a system that quantizes the discrete values. A very fast input signal requires sensitive low-noise circuitry to minimize the information lost during discretization. With packaged products the parasitic resistance and capacitance will often cause too much distortion of the input signal to be useful. Using an ASIC greatly reduces the effect of parasitic devices and using an unpackaged chip reduces it even further. To perform the type of experiments necessary to analyze high speed signals a

specialized low-noise and minimal parasitic implementation is a requirement. This requirement led to the research and design of an ASIC for capturing fast analog signals with a very high sampling rate and on-chip control logic with the ability to reconstruct the analog input signal.

1.3 THESIS ORGANIZATION

This thesis contains the design, layout, simulation, fabrication and testing of an ASIC designed to discretize a fast analog waveform as well as the design and layout of a corresponding PCB to interface with the ASIC. Chapter 2 develops the theory of operation of each individual cell for the digitization circuit and presents the simulation and testing results for the individual cells. Chapter 3 describes the operation of the overall circuit on the chip, including the simulations and how the on-chip control logic interfaces with a control system. Chapter 4 details the design and layout of the PCB developed to control the chip and interface with the captured data. Chapter 5 details the complete test results and compares them to the calculations and simulation results from Chs. 2 and 3. Chapter 6 is the conclusion which includes a summary of the results found and proposals for future research regarding this project.

CHAPTER 2: THEORY OF OPERATION FOR INDIVIDUAL CELLS

2.1 CIRCUIT TOPOLOGY AND DESIGN

The building block for the digitizer circuit is a cell that uses a buffer (two inverters) to delay the clock signal before capturing the charge from the analog input on a poly-poly capacitor. There are two phases of operation: the first phase of operation is data capture; the second phase is replicating the captured data. A schematic of the cell can be seen in Fig. 1 below (the layout is shown in Fig. 10).

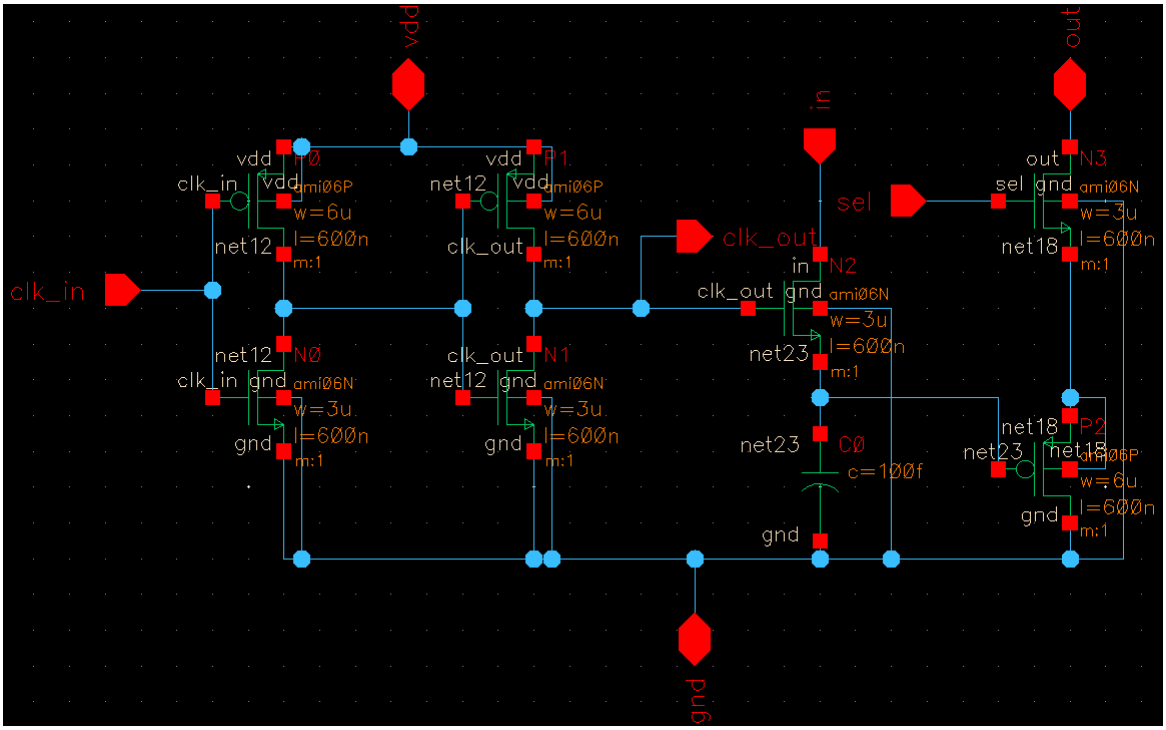

Figure 1: Cell schematic.

The delayed clock signal is connected to the gate of an NMOS transistor, the drain of the NMOS is connected to the analog input signal, the source of the NMOS is connected to the top plate terminal of the capacitor and the bottom plate terminal of the capacitor is connected to ground. On the falling edge of the clock the NMOS acts as a switch and turns off storing the

charge from the analog input on the capacitor. The top plate terminal of the capacitor is also connected to the gate of a PMOS transistor which acts as a source follower [1]. In order to eliminate the body effect the PMOS transistor is fabricated in a separate n-well [1]. The drain of the PMOS is connected to ground and the source is connected to the source of an NMOS transistor, the drain of the NMOS is the output and is connected to a pull up resistor. When the proper select signal is applied the gate of the NMOS will be at logic 1 and it acts as a switch allowing the PMOS to set the current through the output branch, the control logic will be explained in detail in Ch. 3. The current set by the PMOS will control the voltage drop across the pullup resistor and create a scaled representation of the analog input voltage at the time of capture.

2.2 CALCULATIONS AND SIMULATION RESULTS

The devices for the cells were designed to maximize data capture speed which means using the minimum length allowed by the design rules to minimize the parasitic capacitance of the transistors used for the buffer while matching the transconductance parameters of NMOS (β_n) and PMOS (β_p) devices [1]. The transconductance parameters are proportional to the low field mobility, see Eqs. (2.1) and (2.2) below [2].

$$KP_{n,p} = \mu_{0n,p} * C'_{ox} \tag{2.1}$$

$$\beta_{n,p} = KP_{n,p} * \frac{W}{L} \tag{2.2}$$

The electron mobility for NMOS devices (μ_{0n}) is 458.4 cm^2 / (V sec) and the hole mobility for PMOS devices (μ_{0p}) is 212.0 cm^2 / (V sec) [3]. This means the PMOS devices should be 2.16 times wider than the NMOS devices to achieve the desired result of $\beta_n \approx \beta_p$. This design uses a ratio of 2 as an approximation, because of variability in the fabrication process using a precise ratio is not critical [1]. The minimum size allowed is 10/2 which is the size used

for the NMOS transistors, the PMOS transistors are 20/2 [3]. Equations (2.3) through (2.7) illustrate why maximum data capture speed is attained through minimum length devices [1].

$$C_{ox} = C'_{ox} * W * L * (scale)^2 \tag{2.3}$$

$$C_{total} = C_{oxp} + C_{oxn} + \frac{3}{2}\left(C_{oxp} + C_{oxn}\right) = \frac{5}{2}\left(C_{oxp} + C_{oxn}\right) \tag{2.4}$$

$$R_{n,p} = R'_{n,p} * \frac{L}{W} = \frac{VDD}{I_{D,sat}} \rightarrow R'_{n,p} = \frac{VDD}{I_{D,sat}} * \frac{W}{L} \tag{2.5}$$

$$t_{PHL,PLH} = 0.7 * R_{n,p} * C_{total} \tag{2.6}$$

$$t_{PHL,PLH} = 0.7 * \frac{5}{2} * R'_{n,p} * C'_{ox} * (scale)^2 * \left(\frac{W_n + W_p}{W_{n,p}}\right) * L^2 \tag{2.7}$$

The relationship in Eq. (2.7) clearly shows that the propagation delays for an inverter are proportional to L^2, thus using minimum length devices will minimize the inverter delay and maximize the data capture speed [1]. To calculate the propagation delay the values for the effective switching resistance (R_n and R_p) and parasitic capacitance (C_{oxn} and C_{oxp}) must be determined. The capacitances can be calculated from physical constants and SPICE parameters using Eqs. (2.3) and (2.4), the switching resistances can be determined experimentally through a SPICE simulation using the results for I_{on} from Fig. 2 and Eq. (2.5). The schematic for the circuit used to generate the plot in Fig. 2 is shown in Fig. 3 on the following page.

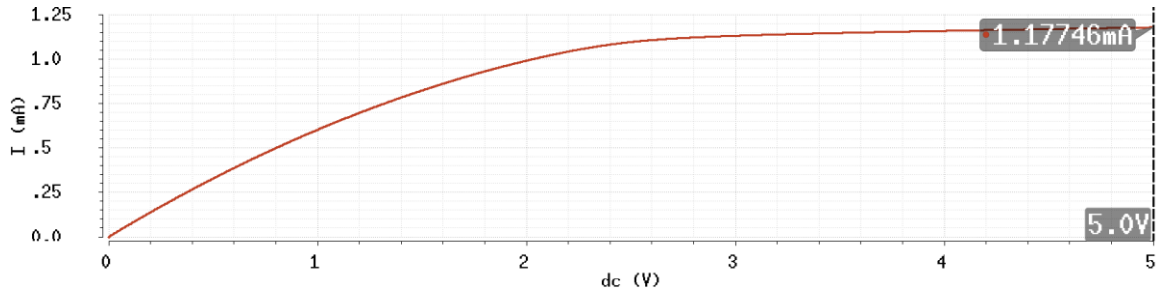

Figure 2: IV plot to estimate average switching resistance. (I_{ON} = 1.177 mA at V_{DS} = 5 V).

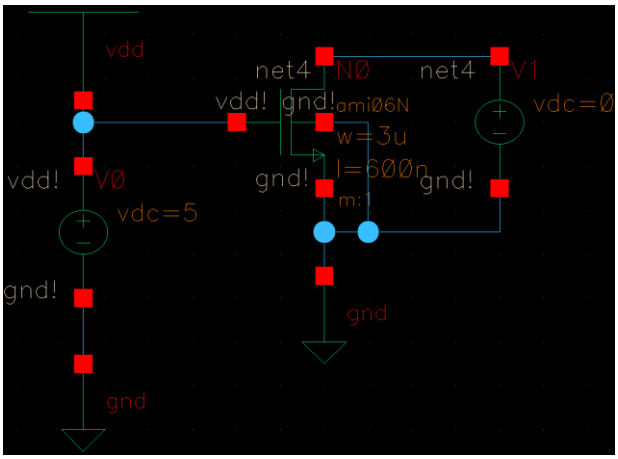

Figure 3: Schematic used to generate IV plot in figure 2.

$$C'_{ox} = \frac{\varepsilon_{ox}}{t_{ox}} = \frac{3.97\varepsilon_o}{t_{ox}} = \frac{3.97\left(8.85\ \frac{aF}{\mu m}\right)}{0.0139\ \mu m} \approx 2.5\ \frac{fF}{\mu m^2} \tag{2.8}$$

$$C_{oxn} = 2.5\frac{fF}{\mu m^2} * 10 * 2 * (0.3\mu m)^2 = 4.5\ fF, C_{oxp} = 2 * C_{oxn} = 9\ fF \tag{2.9}$$

$$C_{total} = 33.75\ fF \tag{2.10}$$

$$R'_n = \frac{5\ V}{1.177\ mA} * \frac{10}{2} = 21.2\ k\Omega \approx 20\ k\Omega\ and\ R'_p \approx 2 * R'_n \approx 40\ k\Omega \tag{2.11}$$

The propagation delay of each buffer can be calculated using Eqs. (2.7) through (2.11), remembering that the second inverter drives one additional NMOS transistor.

$$t_{extra} = 0.7 * 4\ k\Omega * 4.5\ fF = 12.6\ ps \tag{2.12}$$

$$t_{PHL} = t_{PLH} = 94.5\ ps \tag{2.13}$$

$$t_{delay} = t_{PHL} + t_{PLH} + t_{extra} = 201.6\ ps \approx 200\ ps \tag{2.14}$$

$$f_s = \frac{1}{t_{delay}} = \frac{1}{200\ ps} = 5\ GHz \tag{2.15}$$

Because the inverters drive their own output capacitances, the input capacitance for the next inverter and one of the inverters drives one additional NMOS (the delay from this transistor is shown as t_{extra} in the calculations) these equations fully characterize the delay for each stage as

well as the sampling frequency of the digitizer. The simulation results for the digitizer clock signals are shown below in Figs. 4 and 5 as a comparison to the theoretical calculations.

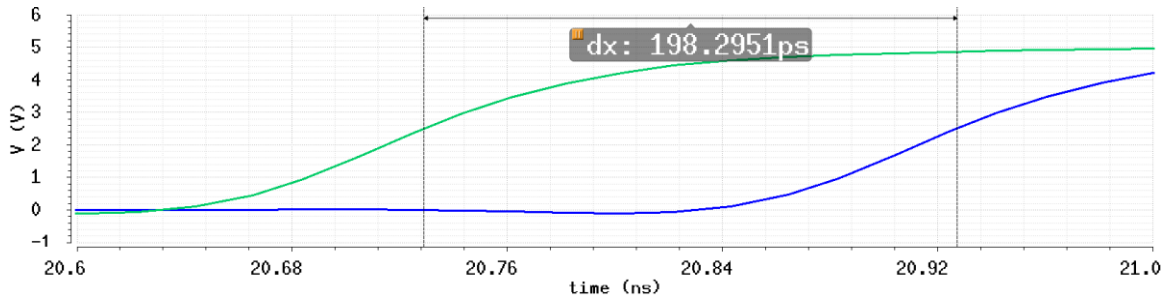

Figure 4: Simulated results for t_{PLH}.

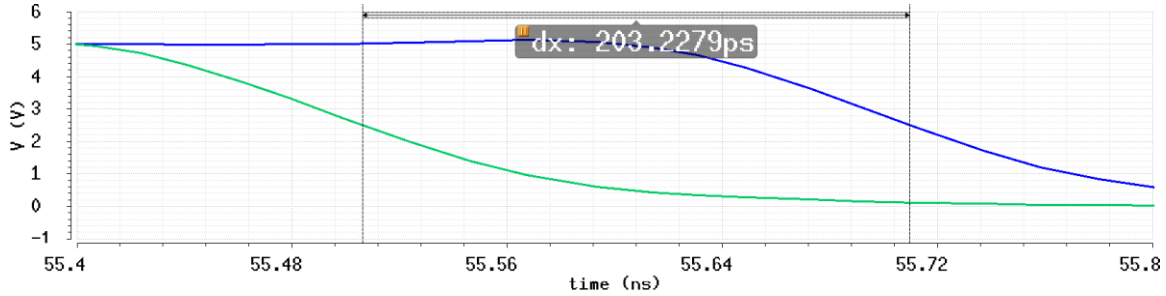

Figure 5: Simulated results for t_{PHL}.

The difference between t_{PLH} (198.3 ps) and t_{PHL} (203.2 ps) in the simulations is 4.9 ps. This means that the sampling frequency is approximately 5 GHz, exactly what was calculated in Eq. (2.15).

Another useful parameter is the limitation on the analog input voltage. In order for the output to be a linear representation of the input the PMOS in the output stage must operate in the saturation region, this means that the relationships in Eqs. (2.16) and (2.17) must be true. The PMOS will operate on the border between triode and saturation when the relationship in Eq. (2.18) is true. The maximum source voltage for the PMOS can be approximated using Eq. (2.19).

$$V_{SG} \geq V_{THP} \tag{2.16}$$

$$V_{SD} \geq V_{SG} - V_{THP} \tag{2.17}$$

$$V_{SD,sat} = V_{SG} - V_{THP} \rightarrow V_{SD,sat} = \left(V_{S,max} - V_{G,max}\right) - V_{THP} \tag{2.18}$$

$$V_{S,max} = VDD - V_{THN} \tag{2.19}$$

Knowing that the drain of the PMOS is connected to ground it is clear that the input on the gate can be driven all the way to ground while remaining in the saturation region. $V_{SD,sat}$ (0.396 V), V_{THP} (0.732 V) and V_{THN} (1.23 V) can all be determined from the transient operating points when simulating the cell output, this is shown in Fig. 6.

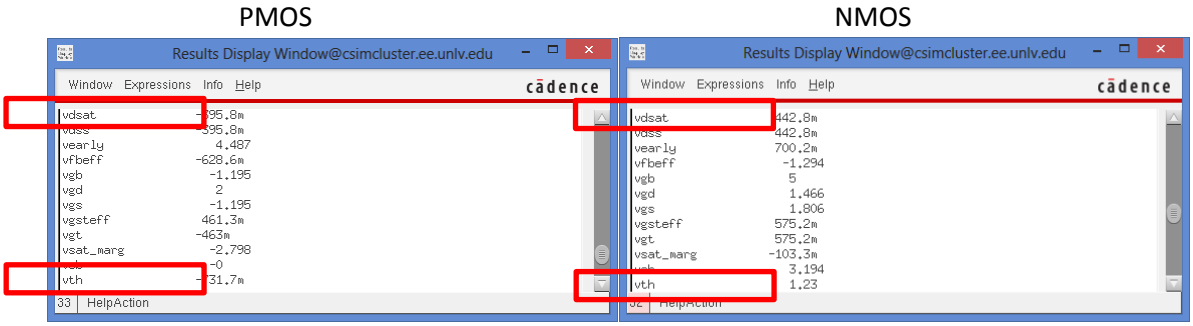

Figure 6: Transient operating point parameters $V_{SD,sat}$, V_{THP}, $V_{DS,sat}$ and V_{THN}.

Figure 7 shows the input (green), shifted output (red) and the derivative of the current of the PMOS (blue) of a single cell sweeping the input voltage from ground to *VDD*.

$$V_{G,max} = VDD - V_{THN} - V_{THP} - V_{SD,sat} \rightarrow V_{G,max} = 2.572\,V \approx 2.6\,V \tag{2.20}$$

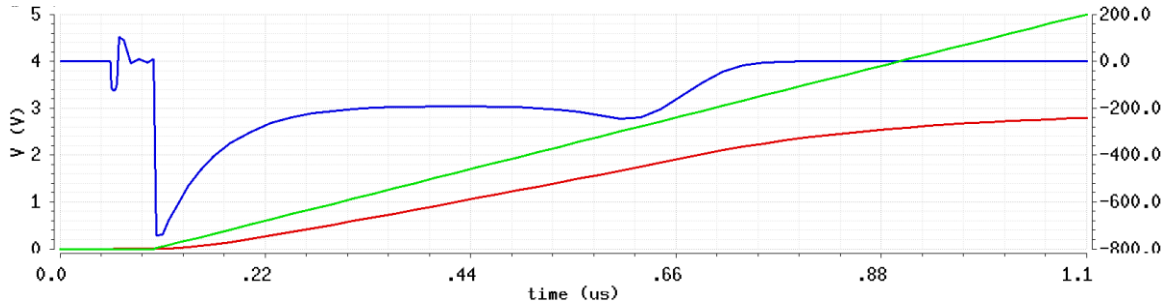

Figure 7: Cell output sweeping the input from ground to *VDD*.

The breakdown in linearity begins when the input is increased above 2.6 V as expected from Eq. (2.20). The ratio appears non-linear at voltages near ground because of the abrupt change in current and simulation step size. The same simulation is shown in Fig. 8 with the input swept from ground to 2 V to expand the view of the linear region. The schematic used for the cell simulations is shown in Fig. 9. Due to process variations the actual circuit testing will limit the input signal to 2 V to ensure linear replication of the input signal.

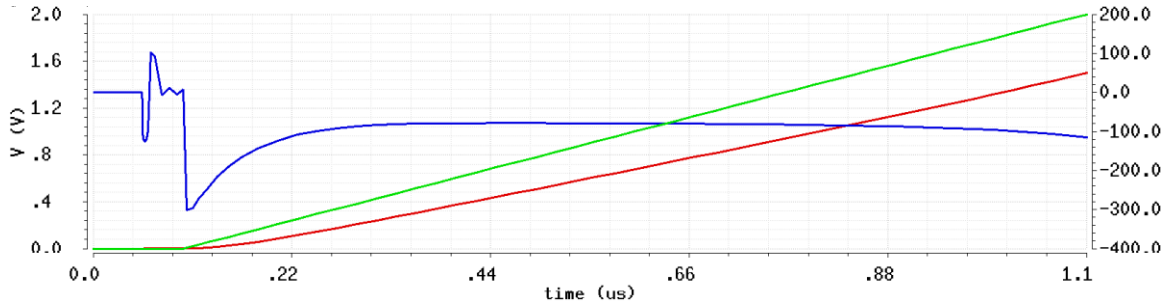

Figure 8: Cell output sweeping the input from ground to 2 V.

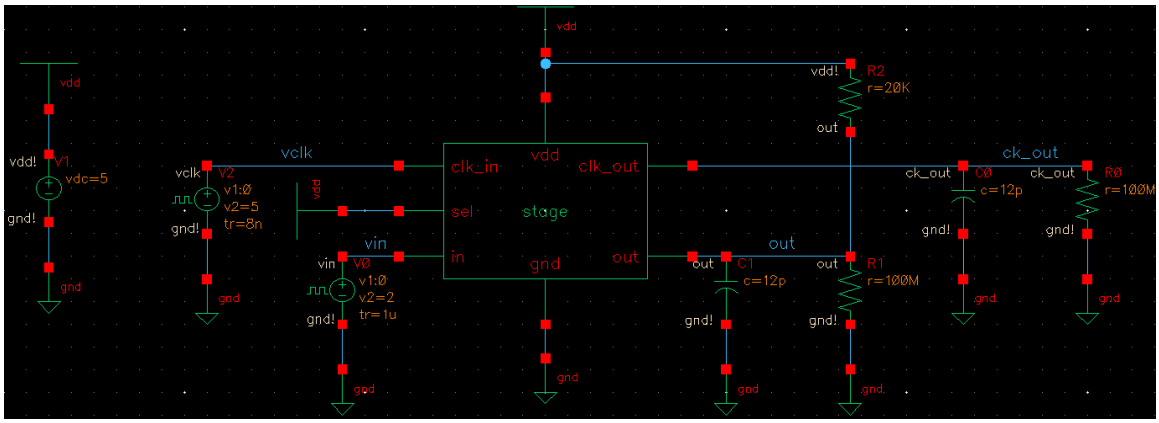

Figure 9: Schematic used to generate figures 6 through 8. (The symbol labelled stage is the circuit shown in figure 1 on page 3).

The last circuit parameters of interest are the minimum and maximum output voltage. The maximum output voltage can be determined from $V_{S,max}$ (see Eq. (2.19) on page 8) and $V_{DS,sat}$ (0.443 V, see Fig. 6 on page 8), the minimum output voltage is 2 V less than the maximum output voltage.

$$V_{out,max} = V_{S,max} + V_{DS,sat} = VDD - V_{THN} + V_{DS,sat} = 4.21\,V \qquad (2.21)$$

$$V_{out,min} = V_{out,max} - 2\,V = 2.21\,V \qquad (2.22)$$

2.3 LAYOUT OF CELL

The cell layout is shown in Fig. 10. The four transistors on the left side are the delay element (buffer), to the right of the buffer is the poly-poly capacitor that stores charge from the input signal, the PMOS on the top right is the current source for the output, the NMOS below it is the switch that captures the data and the NMOS on the bottom right is the switch that controls the output.

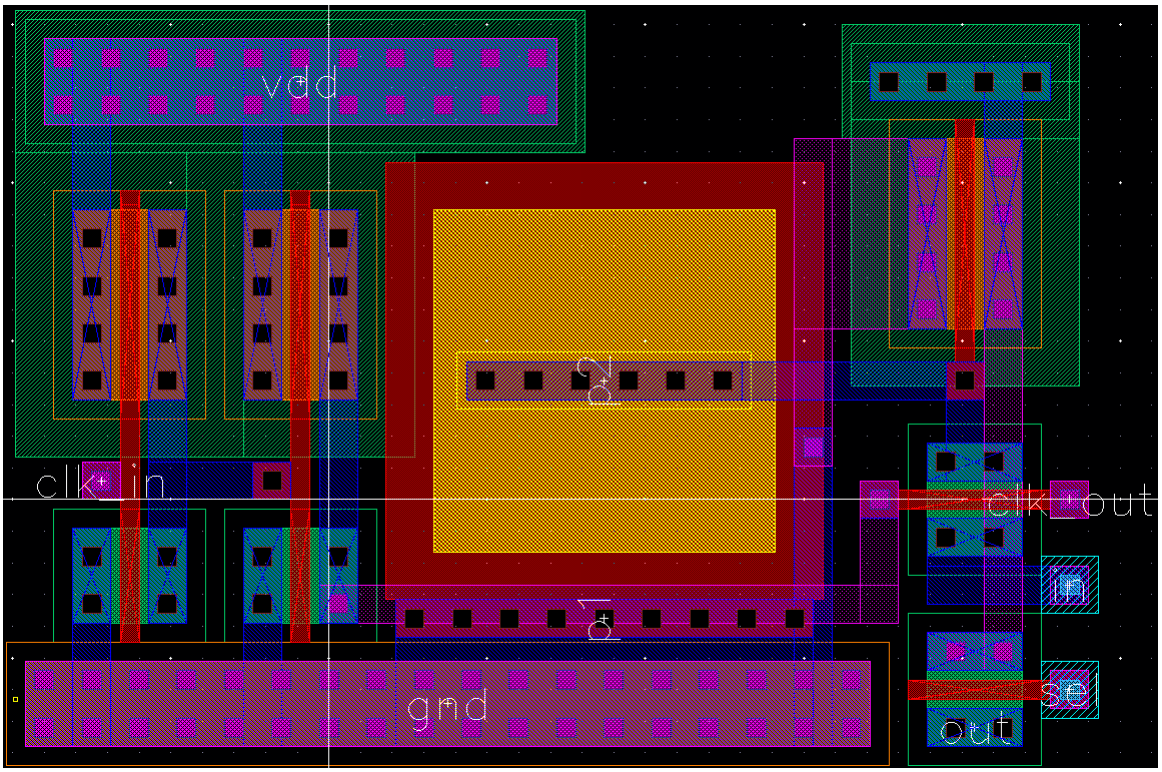

Figure 10: Cell layout.

2.4 TESTING RESULTS

DC characterization of the single cell was performed using the same test conditions shown in Fig. 9 with the clock input and select signals at *VDD* and DC voltages from ground to 2 V applied to the input node in 10 mV increments. An excel chart plotting the input/output

relationship of the circuit is shown in Fig. 11. It is clear from Fig. 11 that the output is not perfectly linear, this nonlinearity can be corrected after the output data is captured by storing the raw data from an oscilloscope and adjusting it with a lookup table. The oscilloscope and DC power supply available have a minimum resolution of 10 mV and at lower voltages some of the output voltages repeat, for example input voltages of 50 mV and 60 mV both produce an output voltage of 2.19 V. Lab equipment with a higher precision is necessary to more accurately characterize this relationship. A 50 kHz saw tooth wave was also applied to the input to show the signal replication at the output, a screen capture from the oscilloscope is shown in Fig. 12, the nonlinearity that was just discussed can also be seen in this figure. A picture of the test setup is shown in Fig. 13. The PCB used for controlling the chip and the test setup will be discussed in detail in Chs. 4 and 5.

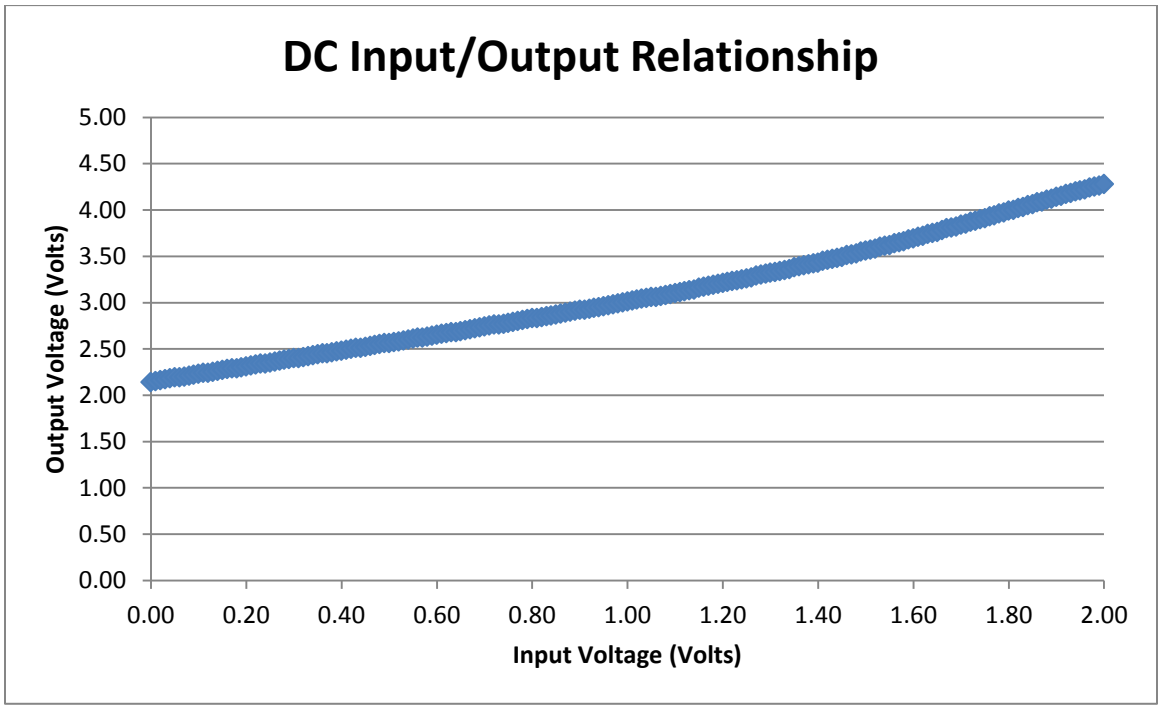

Figure 11: DC Input/Output Relationship.

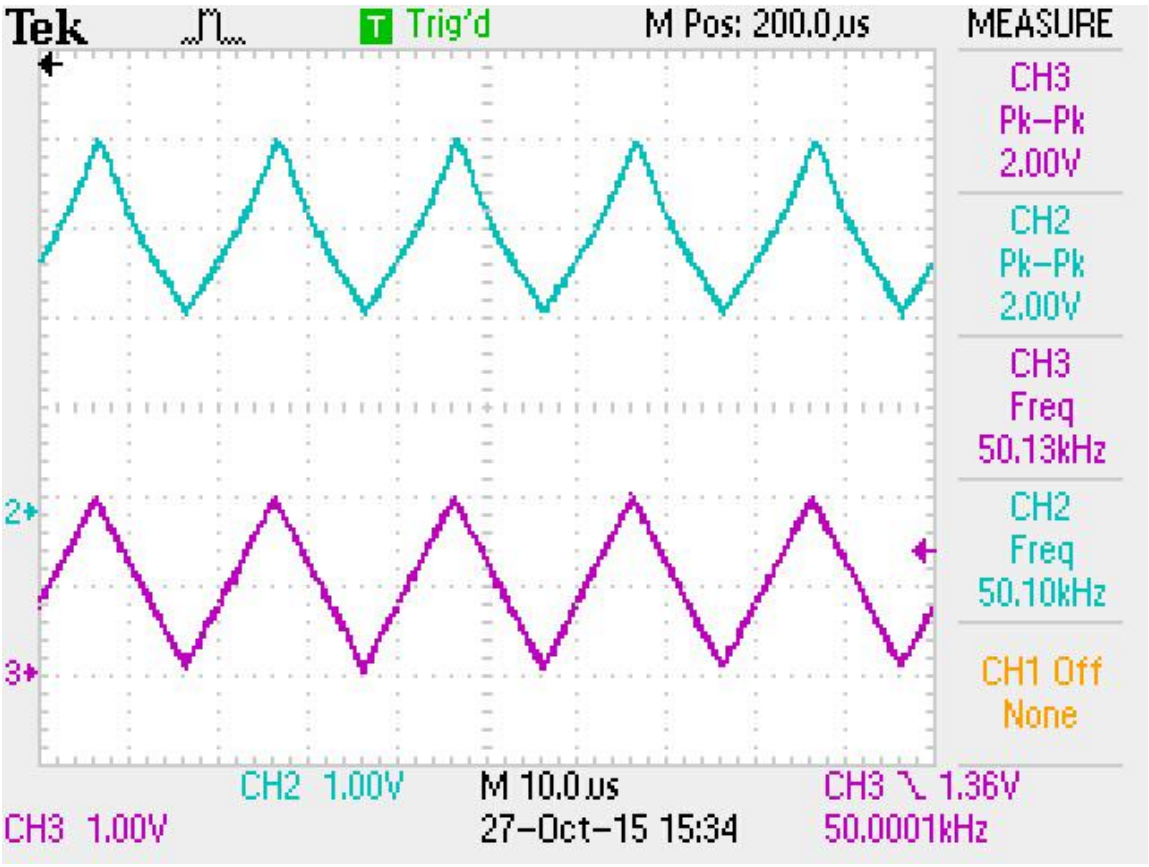

Figure 12: 50 kHz saw tooth wave ground to 2 V. (Input: purple, output: blue).

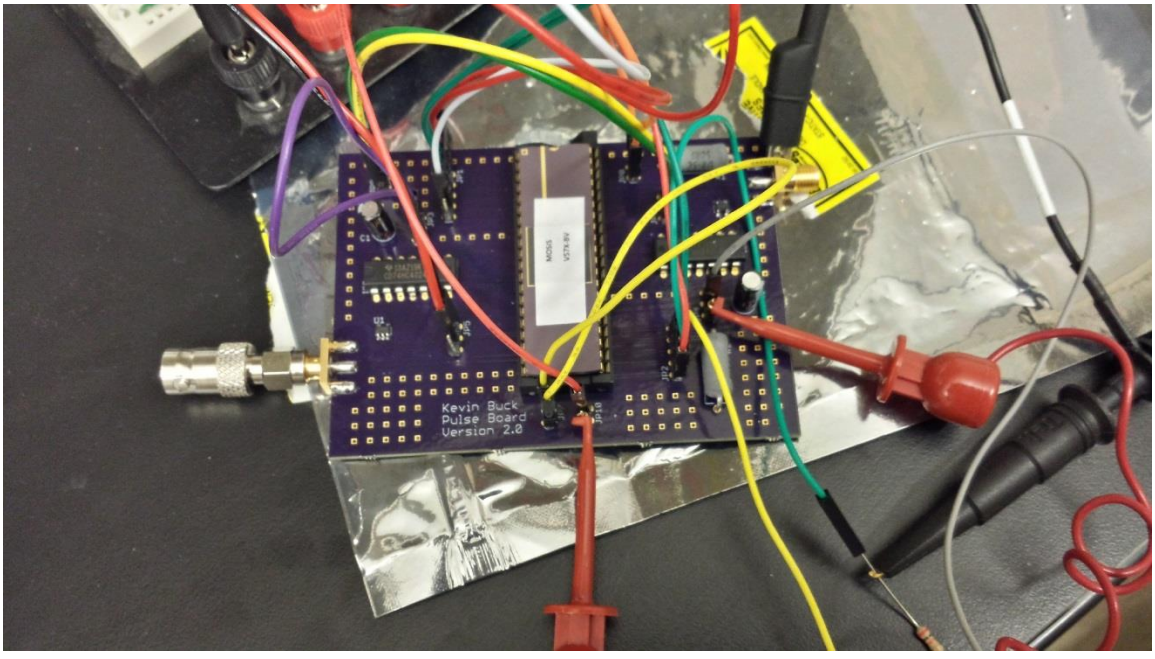

Figure 13: DC characterization test setup.

CHAPTER 3: THEORY OF OPERATION FOR THE TRANSIENT DIGITIZER

3.1 CIRCUIT TOPOLOGY AND DESIGN

The digitizer circuit is made from 128 cells described in Ch. 2 in addition to the control logic for the select signals. There are 7 control signals and an enable signal that are decoded by the control logic to allow each cell to be read out sequentially from the first cell to the last. Figure 14 shows how the cells are connected without the control logic.

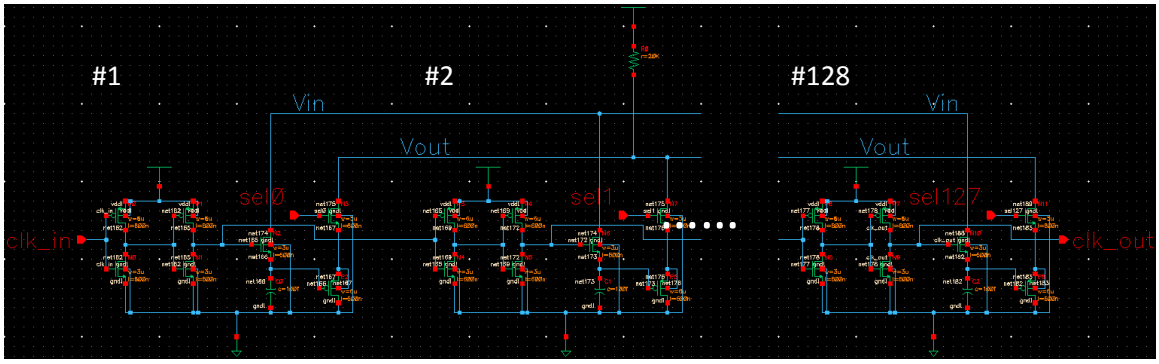

Figure 14: Schematic of the digitizer without control logic.

In the layout there are 8 rows each with 16 cells in addition to the control logic (see Fig. 26 on page 23). The control logic consists of a 5 input AND gate accompanying each cell and a 4 input AND gate at one end of each row. The three most significant bits of the select signal are the same for each row so the 4 input AND gate acts as a row select, three of the inputs are the control logic and the fourth input is the enable signal. The 5 input AND gates connected to each cell receive the four least significant bits of the select signal and the row select signal (the output of the 4 input AND gate) as inputs. The reason for not using NAND gates (smaller layout and lower power consumption) is that the extra inverter delay in an AND gate is needed to ensure that the select signals do not overlap. The schematics for the control logic are shown in Figs. 15 and 16.

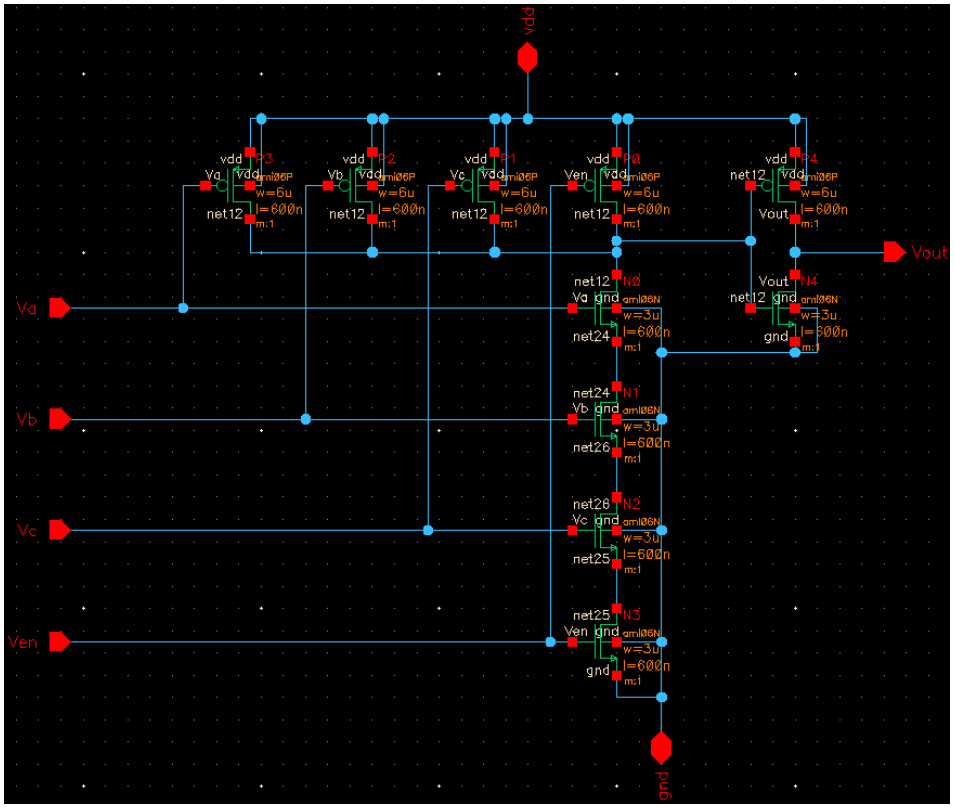

Figure 15: 4 input AND gate used for row select signal.

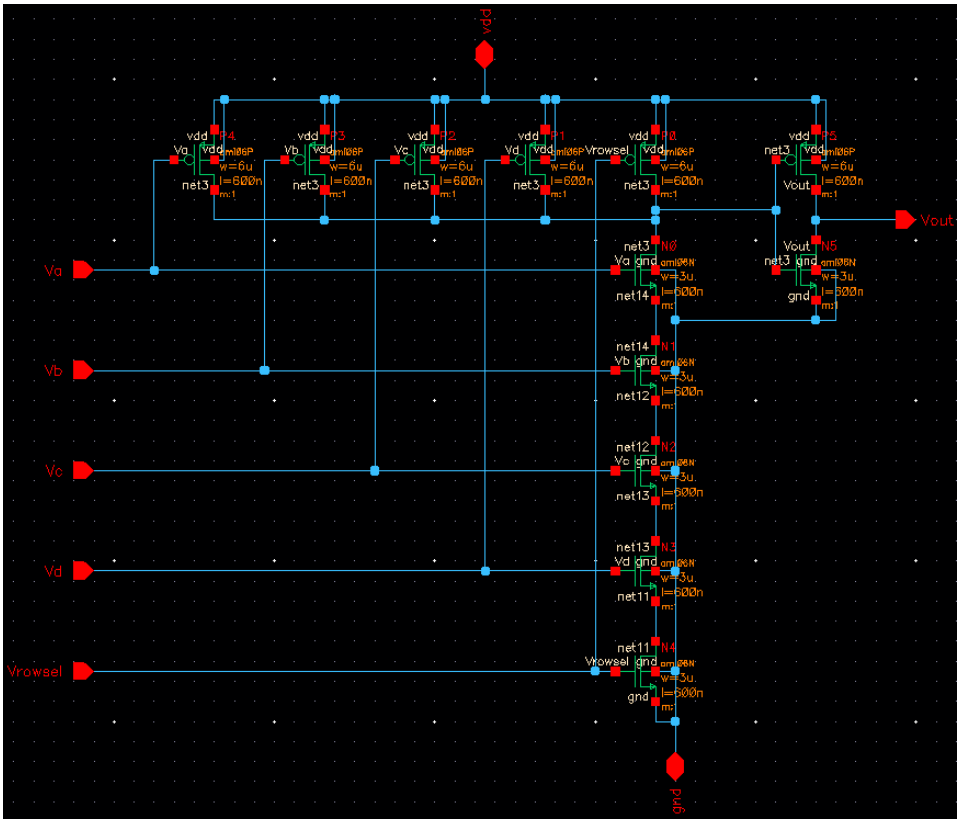

Figure 16: 5 input AND gate used for each cell.

3.2 CALCULATIONS AND SIMULATION RESULTS

The results for the buffer delay and sampling frequency for the digitizer were derived in Ch. 2.2 (see Eqs. (2.14) and (2.15) on page 6) and the comparison to simulation results are shown in Figs. 4 and 5.

An important consideration for using this circuit is the length of time after the data is captured that the stored information remains valid. The stored data will have some leakage through the NMOS device that controls when the data is captured due to the off current and through the reverse biased PN junction to the substrate. A SPICE simulation with a gate voltage of 0 V and V_{DS} = 2 V (the worst case scenario for the operating limits outlined) results in a drain current of 2 pA. Using the current voltage relationship for a capacitor we can define the leakage relationship in Eq. (3.1).

$$\frac{I}{C} = \frac{dV}{dt} \rightarrow \frac{2\ pA}{0.1\ pF} = \frac{2\ mV}{100\ \mu s} \tag{3.1}$$

In the test setup used the readout process takes 138 μs; this means that the output error is less than 3 mV. This error can be further reduced by increasing the readout speed.

When reconstructing the output signal it is also useful to know the offset voltage. In Ch. 2.2 it the minimum and maximum voltages were calculated (see Eqs. (2.21) and (2.22) on page 10). The minimum and maximum output voltage simulation is shown in Fig. 17.

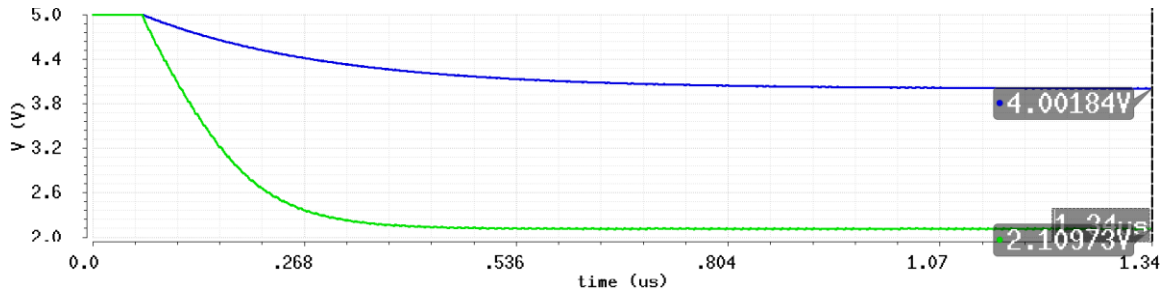

Figure 17: Minimum and maximum output voltages. (0 V and 2 V inputs).

Another parameter of interest is the total data capture time for the overall circuit. The delay for each stage was derived in Ch. 2.2 (see Eq. (2.14) on page 6).

$$t_{capture} = \frac{t_{delay}}{cell} * 128\ cells = 25.6\ ns \tag{3.2}$$

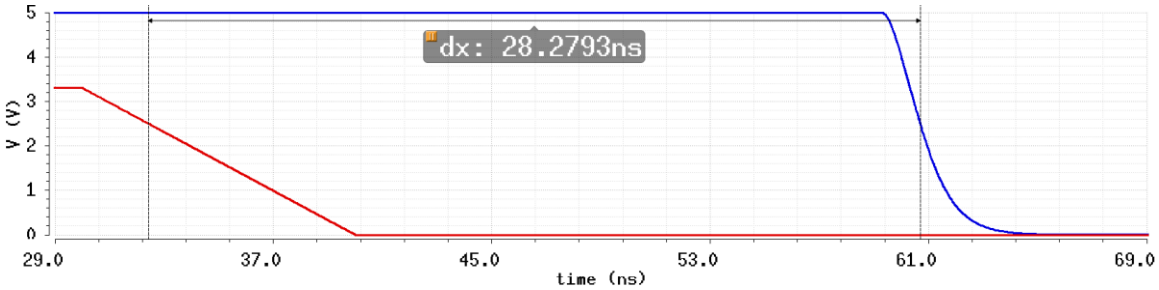

Figure 18: Delay from the input clock to the final stage.

The capture time in the Fig. 18 simulation is 28.3 ns, approximately the same capture time calculated in Eq. (3.2). The actual capture time of the circuit will be slightly less than what is shown in the simulation because the oscilloscope output was modeled as a large resistance and capacitance in parallel and the data capture begins approximately 200 ps after the falling edge of the input clock. All of the simulations in this section modeled the oscilloscope probe as a 100 MΩ resistor in parallel with a 12 pF capacitor with the exception of Fig. 20.

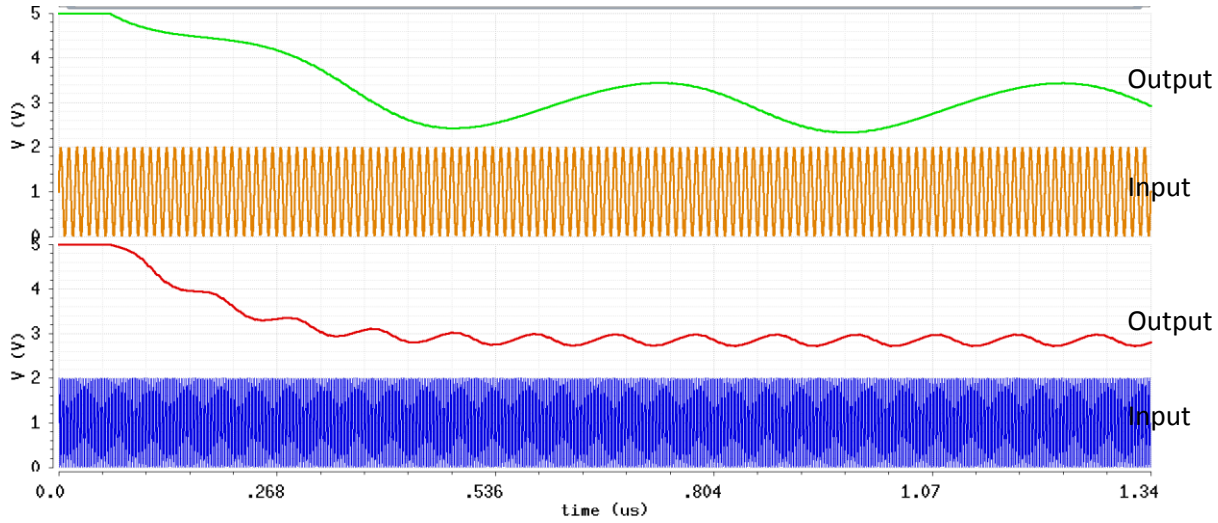

Figure 19: Sinusoidal inputs. (100 MHz and 500 MHz).

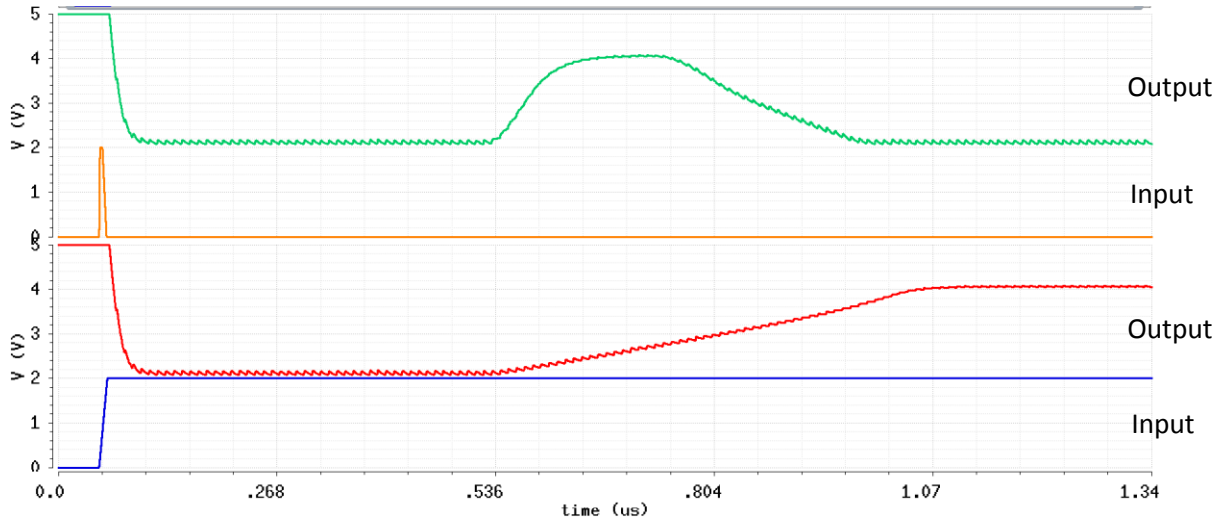

Figure 20: Pulse inputs. (1 ns rise time, 3 ns on time, 5 ns fall time and 10 ns rise time).

Figures 19 and 20 are simulation results using sinusoids and pulses as inputs to the complete circuit. Figure 21 shows the schematic used to generate Figs. 17 through 20; it is also the schematic for the complete chip. For simulating fast pulse inputs a lower output capacitance of 1 pF was needed to create an accurate replication of the input signal. If the output signal replication during testing with a standard oscilloscope probe is poor it may be necessary to use a low capacitance probe or add a small capacitance (1 pF or less) in series with a standard probe.

The simulation results shown in this section are how the chip is expected to behave when tested in the laboratory. The testing results in Ch. 5 will be compared to the simulations and calculations shown here as well as Ch. 2.

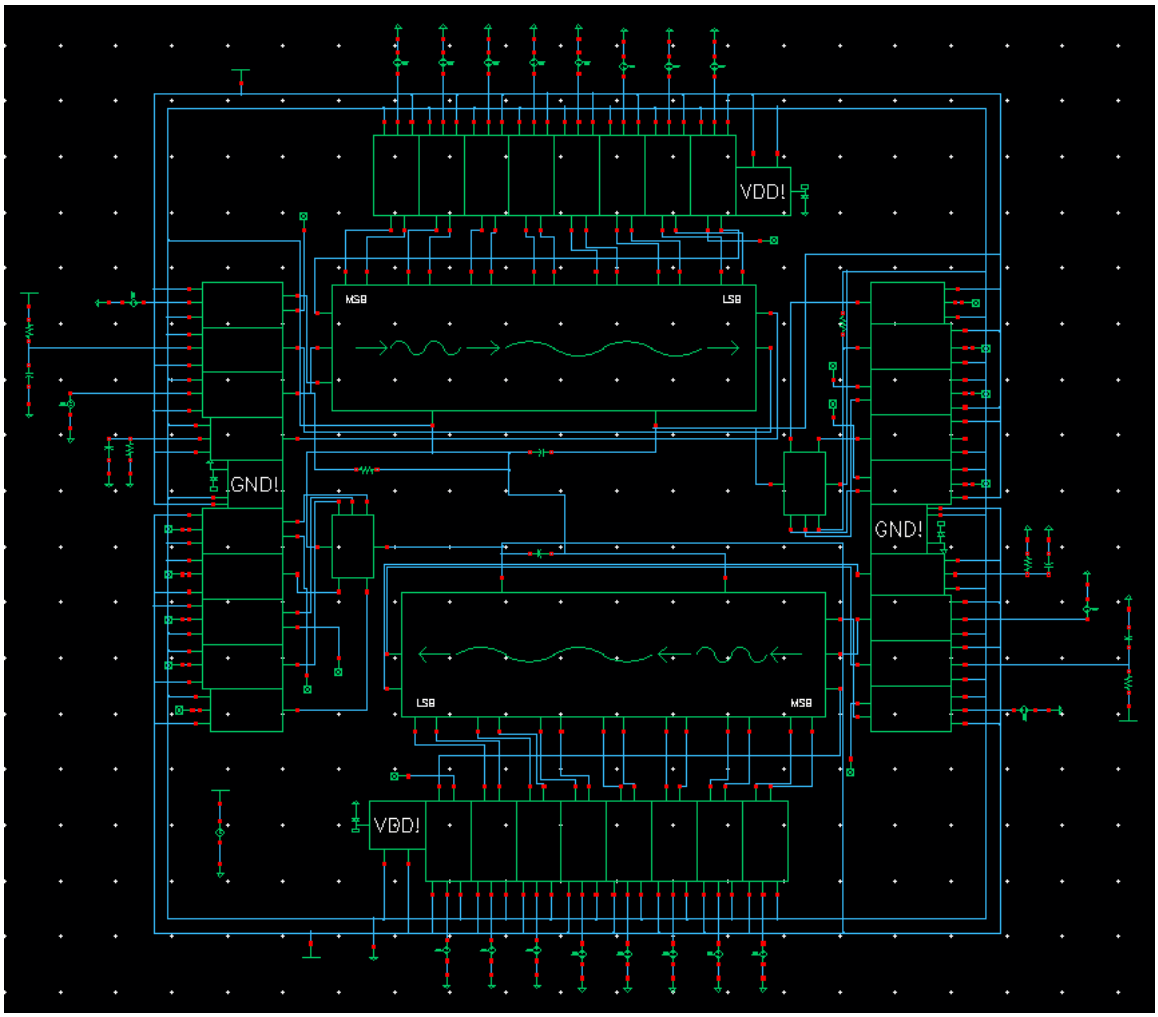

Figure 21: Schematic of the complete chip. (Used for generating figures 17 through 20).

3.3 LAYOUT OF THE TRANSIENT DIGITIZER

This section contains the overall chip layout (Fig. 22 below), a photo of the bare die (see Fig. 23 on page 21), the 4 input AND gate layout used for the row select (see Fig. 24 on page 22), the 5 input AND gate layout used for each cell (see Fig. 25 on page 22) and the transient digitizer layout (see Fig. 26 on page 23). There is also a description of the overall chip architecture at the end of the section.

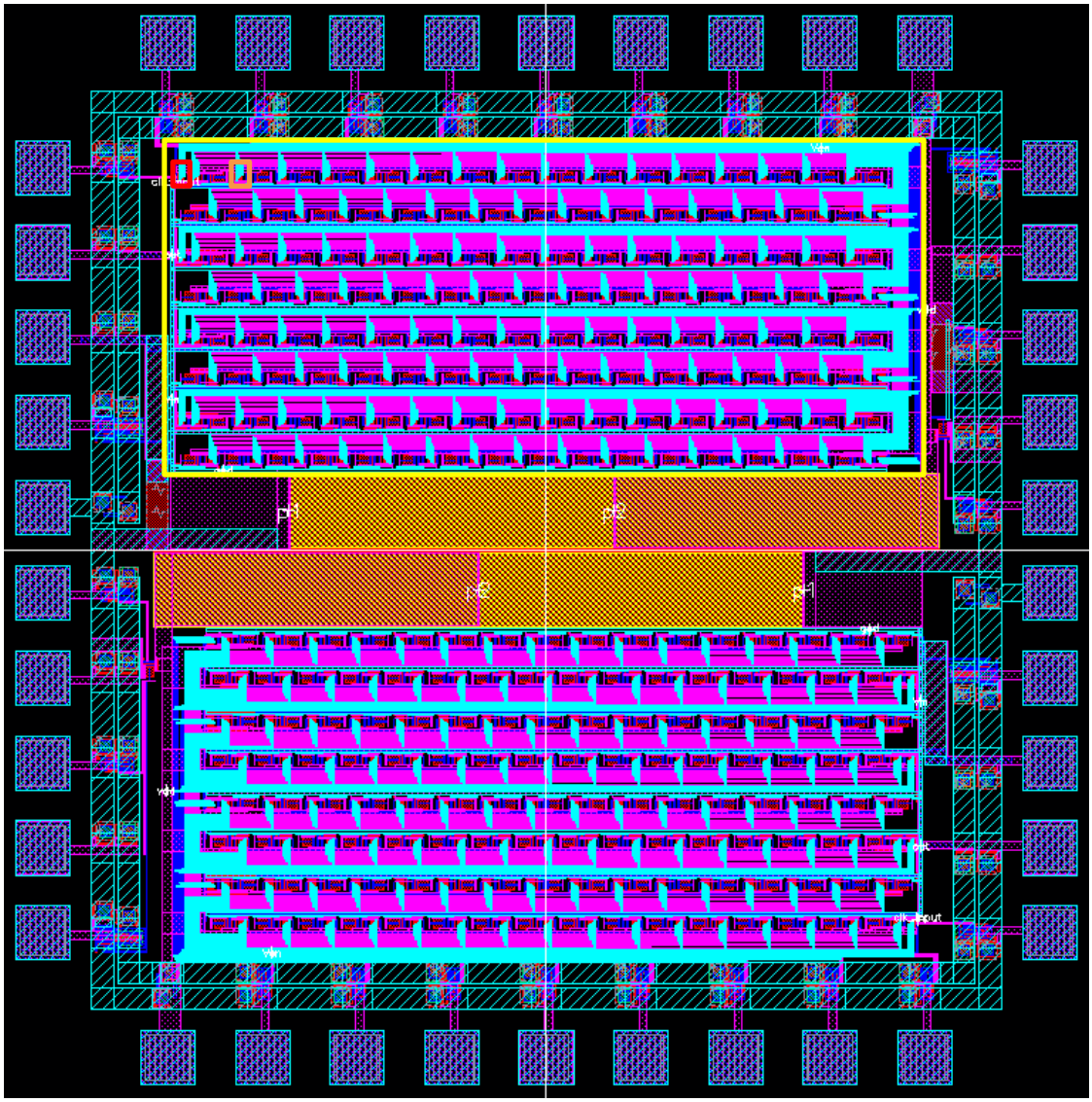

Figure 22: Complete chip layout, 1.5 mm x 1.5 mm. (colored boxes show the areas of the expanded views shown in figures 24 through 26).

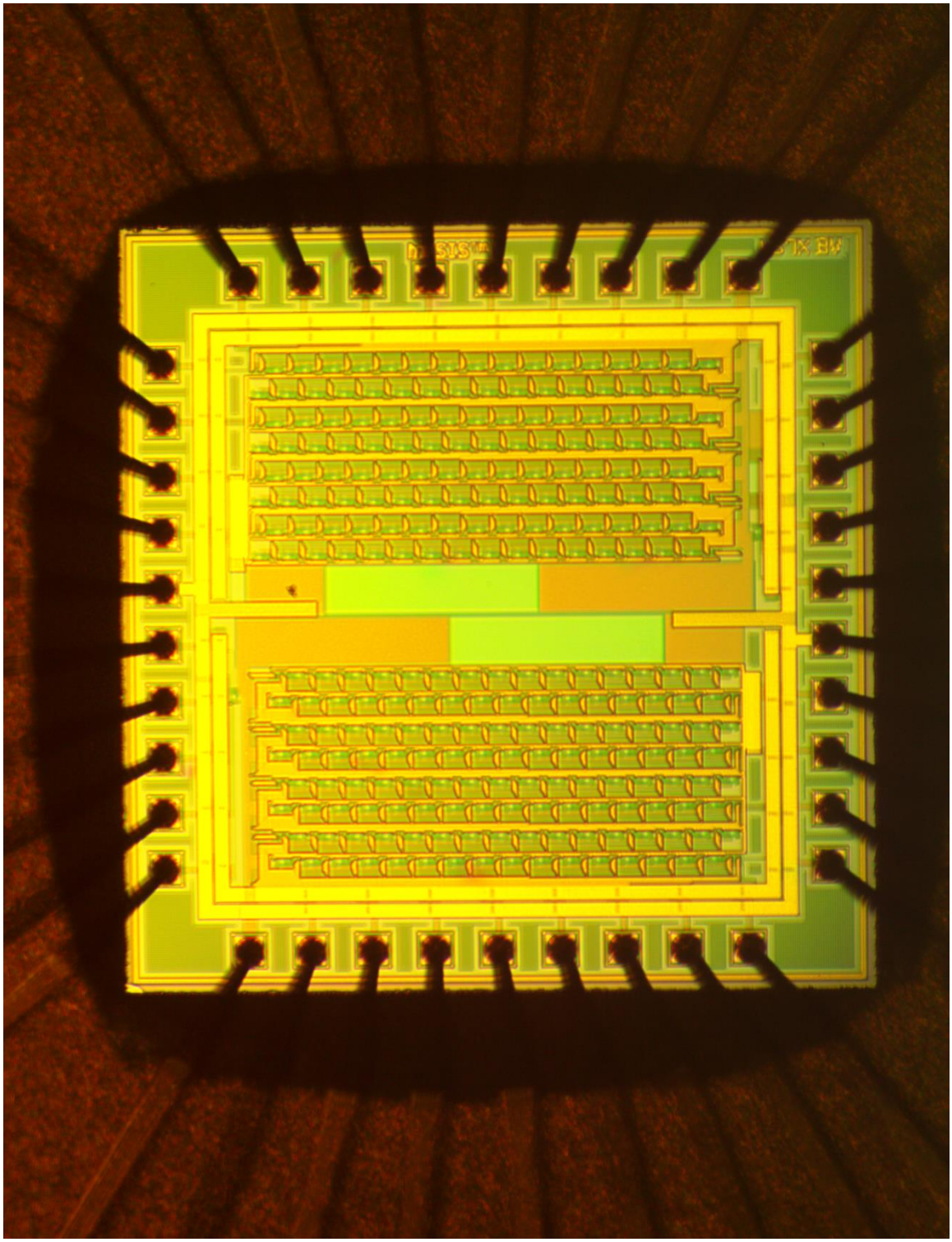

Figure 23: Photograph of the chip.

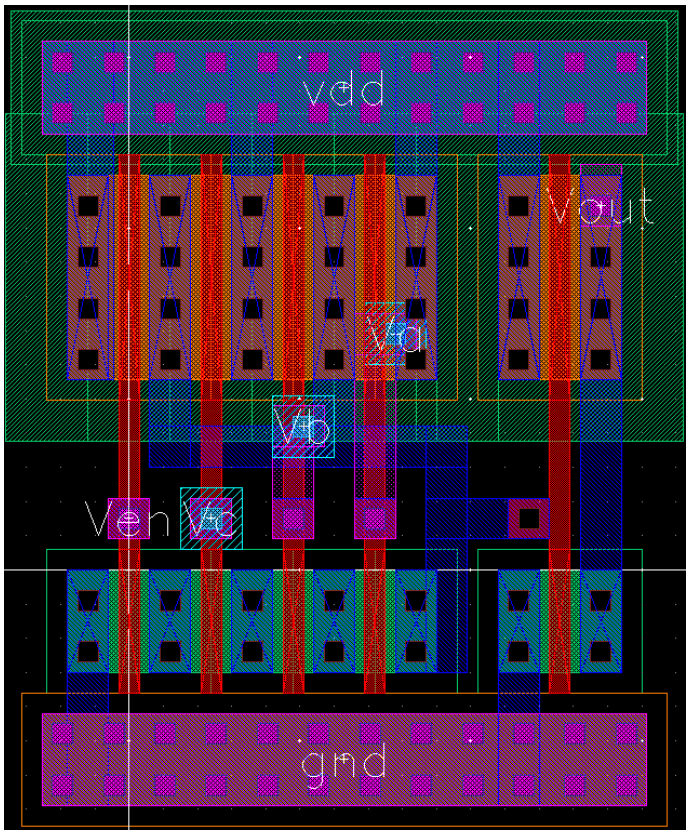

Figure 24: Layout of 4 input AND gate used for row select signal. (Red box on figure 22).

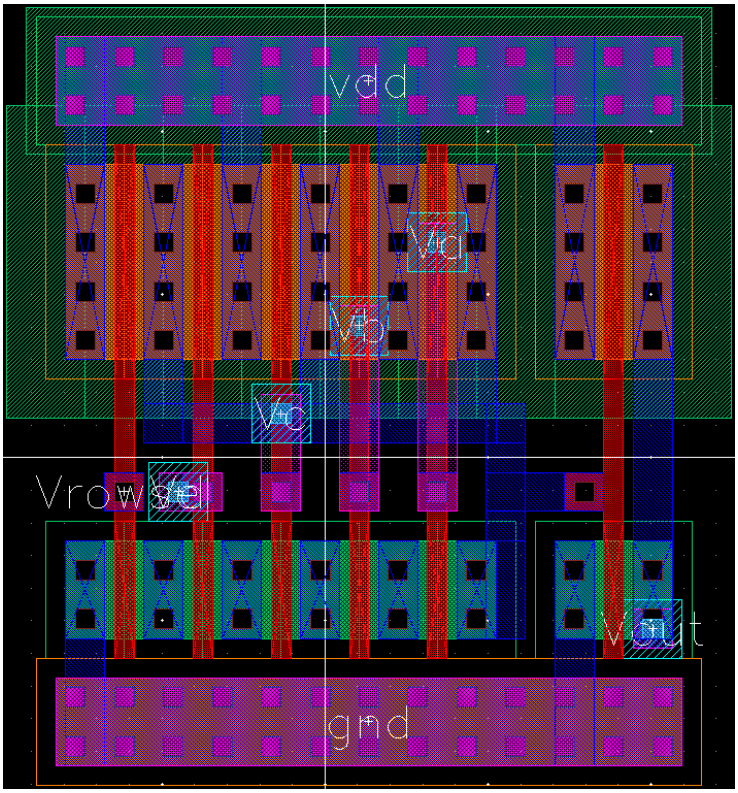

Figure 25: Layout of 5 input AND gate used for each cell. (Orange box in figure 22).

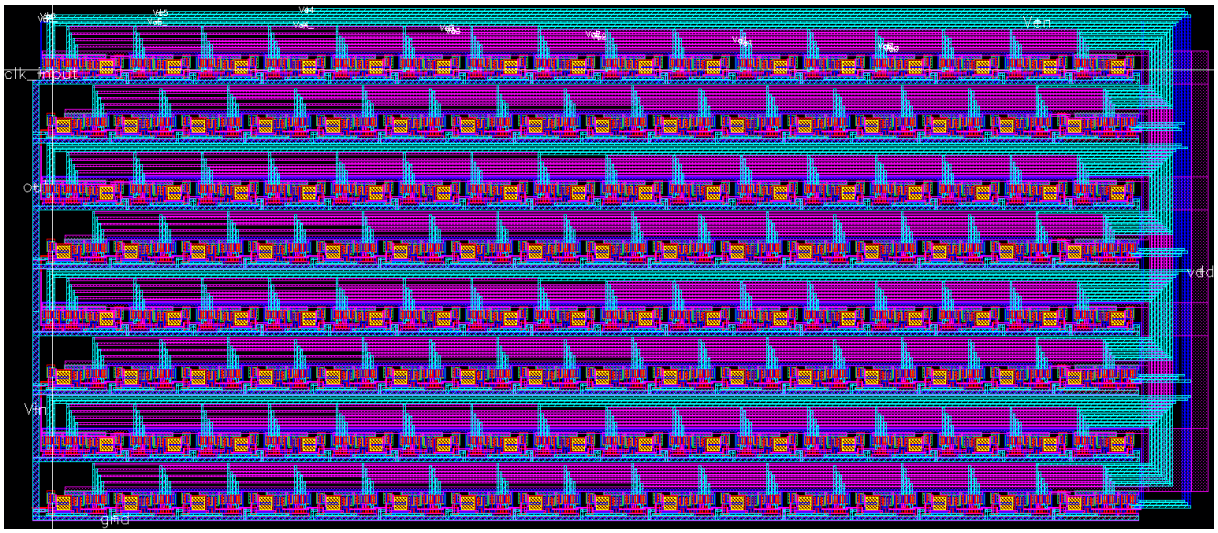

Figure 26: Layout of digitizer with control logic. (Yellow box in figure 22).

The layout of the chip includes two complete circuits, two test structures and two decoupling capacitors between *VDD* and ground. One of the circuits and test structures has an on-chip 50 Ω termination, the other circuit and test structure are terminated off-chip with 50 Ω resistors on the PCB. The test structures are single cells that are used for the DC characterization of the input/output relationship.

CHAPTER 4: DESIGN AND LAYOUT OF PCB AND INTERFACE

4.1 PCB DESIGN AND LAYOUT

The PCB is designed to generate the necessary control signals for proper functionality of the digitizer chip. Four input signals control the operation of the circuit: the analog input, the trigger, the enable for the digitizer (also acts as a reset for the counter), and the clock for the counter. Figures 27 through 30 show a block diagram, schematic, layout and a sample timing diagram for the PCB.

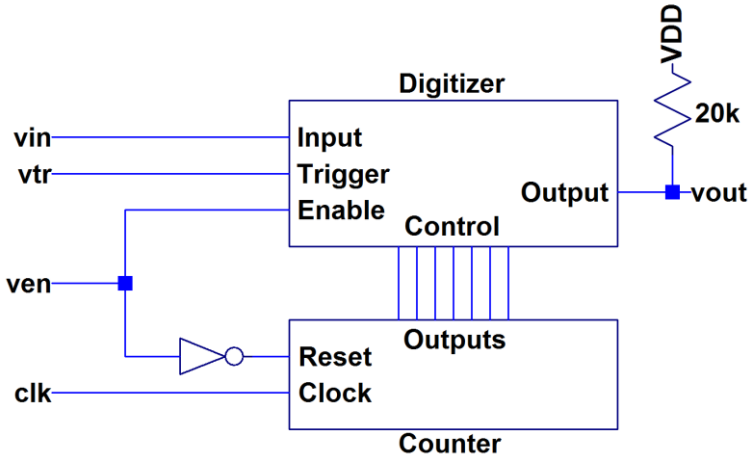

Figure 27: Simplified block diagram of the control system PCB.

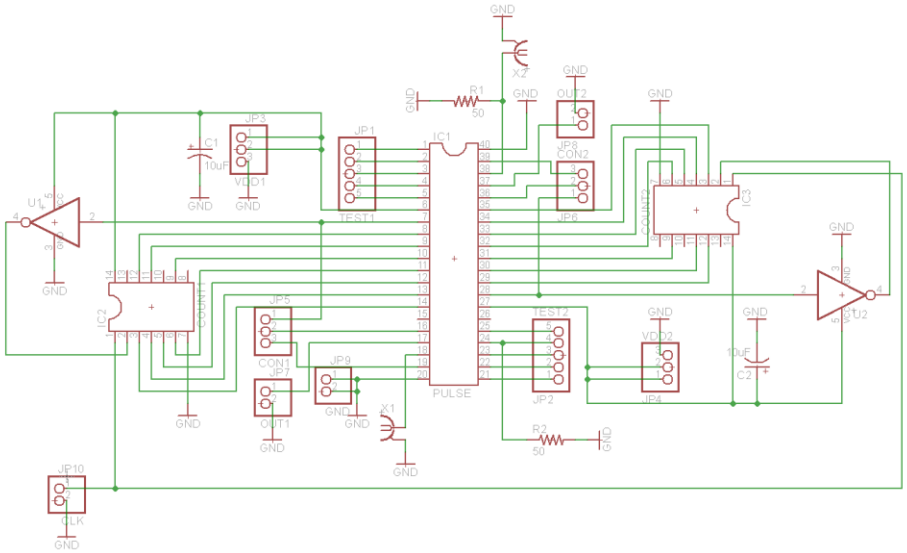

Figure 28: Complete schematic of the control system PCB.

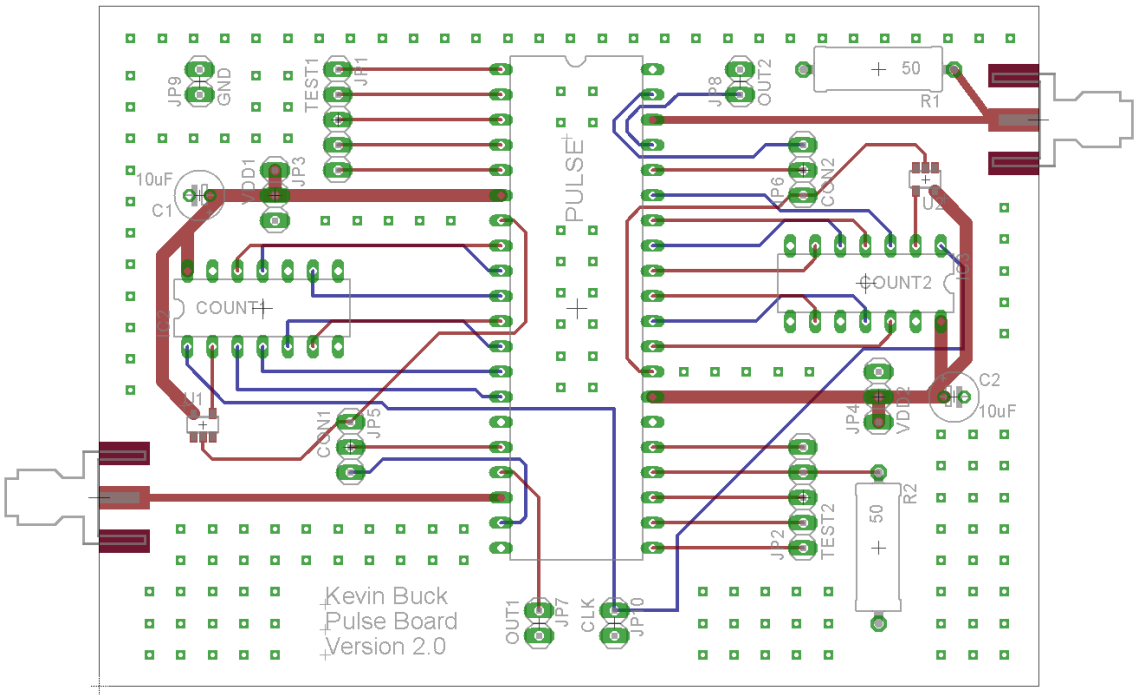

Figure 29: Layout of the control system PCB.

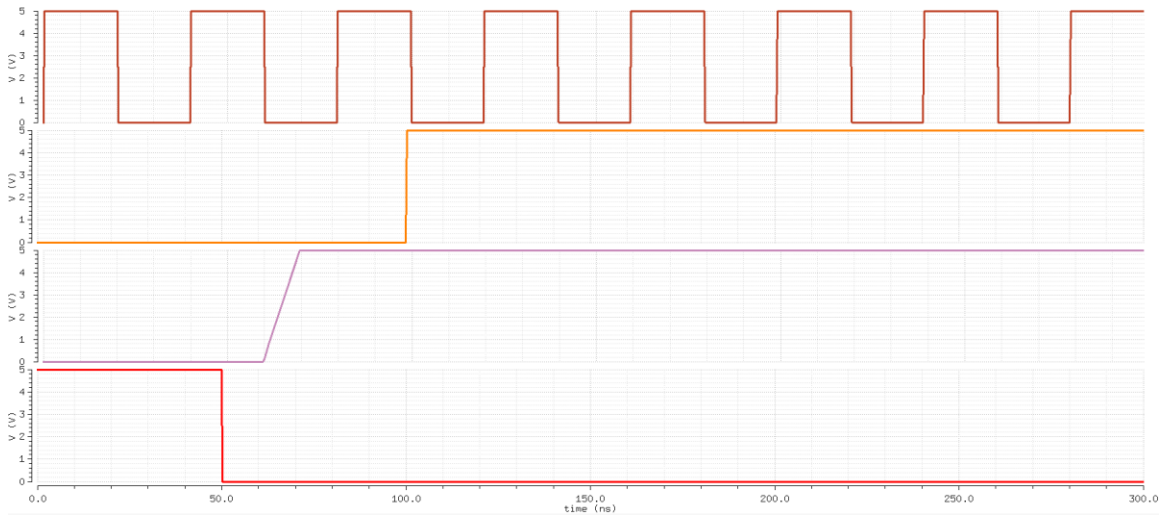

Figure 30: Sample timing diagram for the control system PCB.

When the trigger signal (bottom in red) is driven high the digitizer is sensing the analog input signal, on the falling edge of the trigger data capture begins by shutting off the switch allowing the signal to pass to the storage capacitor. In previous chapters the trigger was referred to as clock, the name change is made in this chapter to avoid confusion when referring to the

clock signal used to read out the data. The input signal (shown in pink above the trigger signal) is captured during the next 25 ns at 200 ps intervals, (see the theoretical calculation is shown in Eq. (2.14) on page 6). After the data is captured the enable signal (shown in orange above the input) is driven high and data can be read from the digitizer. The counter reset signal is the complement of the enable so once enable is driven high the counter increments at a rate equal to the incoming clock signal (shown in brown at the top). In practice the clock signal will be much slower than the time scale shown in the diagram, it is for illustrative purposes only. An oscilloscope probe with a trigger setting just below the maximum possible output is set to capture the reconstructed output signal and the circuit resets to a sensing operation.

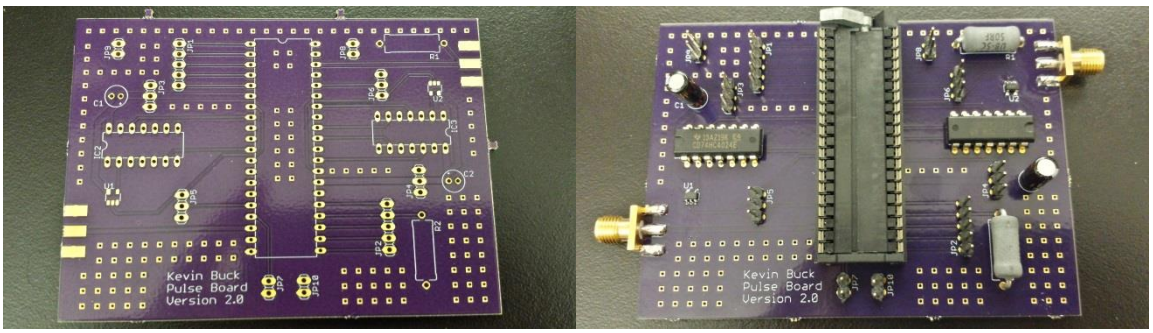

Figure 31: Photo of an unpopulated and populated PCB.

The 40 pin dual inline package (DIP) in the middle of the PCB is the transient digitizer circuit, the two 14 pin DIPs are the counters used to read out the data (Texas Instruments CD74HC4024E), the two SOT23-5 surface mount components are the inverters used for the reset signals to the counters (Texas Instruments SN74LVC1G04), the SMA connectors are for the analog input signal to the digitizer circuit, the 50 Ω resistors are for matching termination for a source generator (one of the circuits and test structures is terminated on-chip, these two resistors are for the off-chip terminated circuit and test structure) and the male headers are for inputs,

outputs, *VDD* and ground. The PCB was designed using EAGLE and was fabricated by OSH Park.

4.2 INTERFACE SYSTEM

The PCB requires properly timed control signals in order to function as intended. An ATMEL SAMD20 microcontroller was used to generate the proper clock and enable signals as well as a trigger for a function generator to create the desired input signal. Figure 32 shows a simplified block diagram of the system used to test the complete circuit. The pulse trigger output is technically optional; it is used to synchronize data to the capture time of the circuit. For example when the input is a fast transient signal the pulse trigger is necessary so the data is actually captured. For periodic signals the pulse trigger is not needed unless it is necessary to precisely control when the input will begin capturing. A photo of the microcontroller is shown in Fig. 33. The code for the microcontroller was created by modifying software supplied from the Atmel Software Framework library and can be found in the appendix.

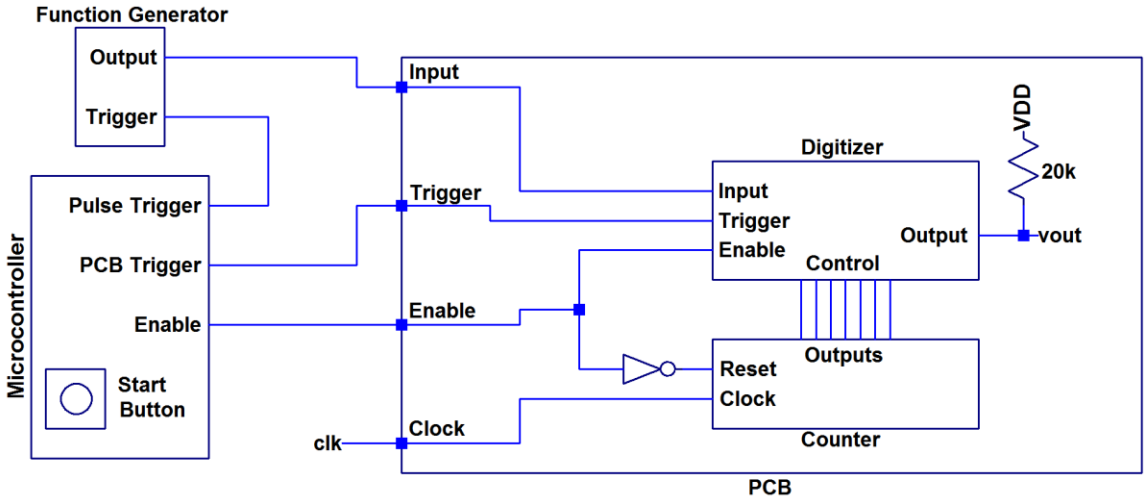

Figure 32: Simplified block diagram of the complete testing system.

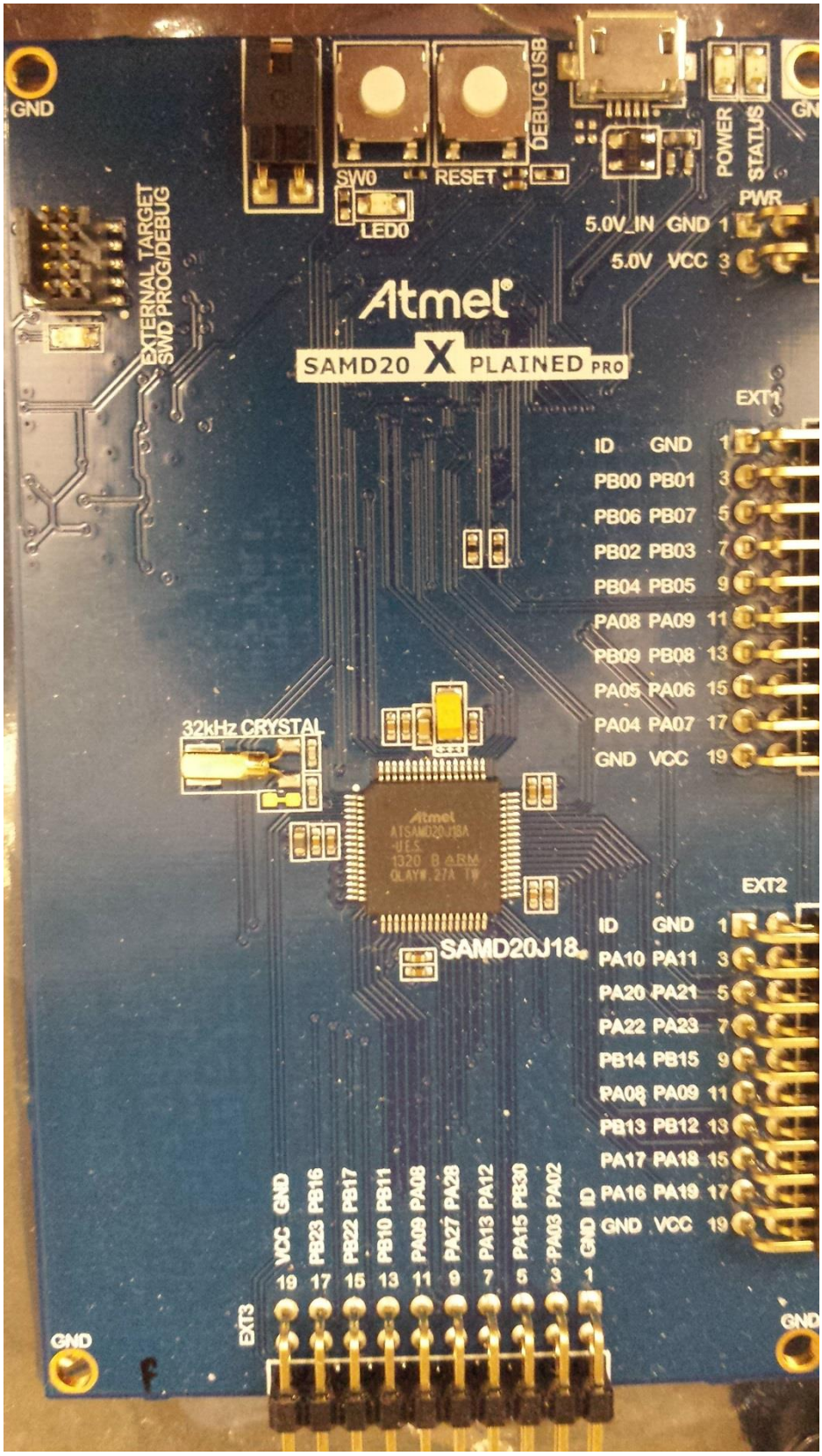

Figure 33: ATMEL SAMD20 microcontroller.

CHAPTER 5: COMPARISON OF SIMULATION AND TEST RESULTS

5.1 MINIMUM AND MAXIMUM OUTPUT VOLTAGE

The minimum and maximum output voltage of the circuit was found during the DC characterization testing shown in Ch. 2.4 (see Fig. 11 on page 12). The minimum output voltage corresponding to an input of 0 V is 2.14 V and the maximum output voltage corresponding to an input of 2 V is 4.28 V. The experimental minimum and maximum output voltages were 2.11 V and 4.00 V (see Fig. 17 on page 16).

5.2 CAPTURE TIME AND PROPAGATION DELAY

Since it is not possible to probe internal nodes on the chip, the best estimates that can be made regarding the capture period and propagation delay are measuring the difference between the input and output clock and providing a known frequency at the input and measuring the number of cycles at the output before a discontinuity. Both measurements are a rough approximation but they provide a reasonable estimate for the amount of data that can be captured by the circuit and the propagation delay of a cell.

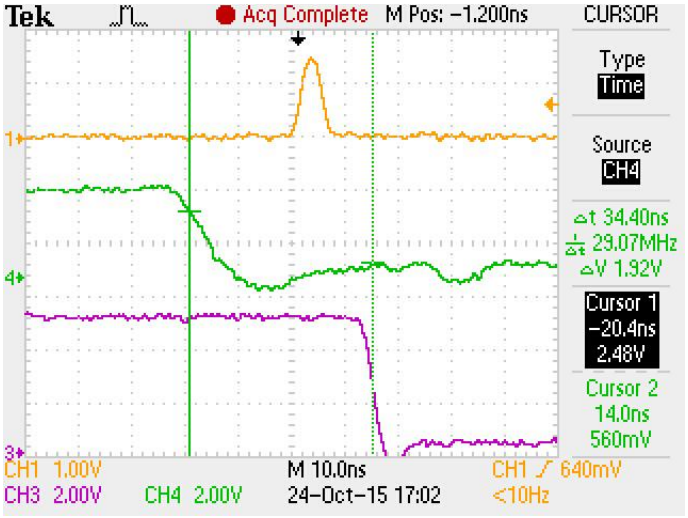

Figure 34: Measured delay from the falling edge of the clock to the output.

The measured time between the input and output crossing 2.5 V in Fig. 34 is 34.4 ns but there is a 200 ps delay before the first capture and the output has to drive a large capacitance from the oscilloscope probe. Since we know the output should take approximately 200 ps to transition, we can take the difference between the measurement and this approximation and subtract that value as well as the 200 ps delay on the input to estimate the capture time, this estimate is shown in Eq. (5.1). The estimated propagation delay is shown in Eq. (5.2).

$$t_{capture,measured} = 34.4\,ns - (3\,ns - 200\,ps) - 200\,ps = 31.4\,ns \qquad (5.1)$$

$$t_{delay,measured} = \frac{t_{capture,measured}}{128} = 245\,ps \qquad (5.2)$$

A second way to estimate the capture time is to input a sinusoid of a known frequency, measure the known period on the output and relate it to the input.

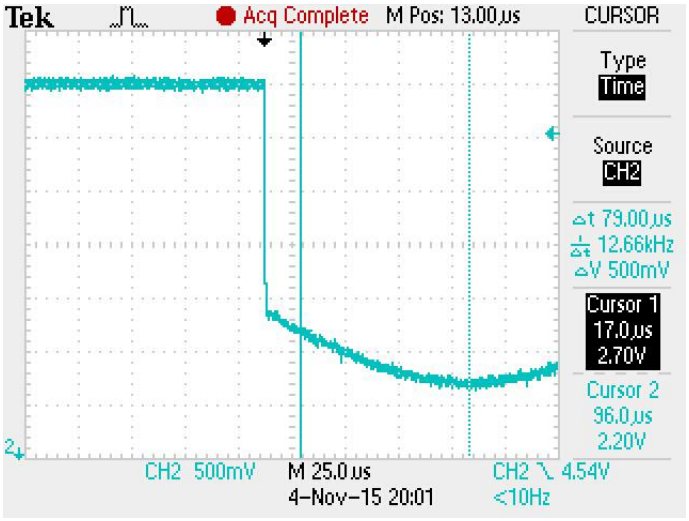

Figure 35: Output time of ¼ of a cycle for a 15 MHz sinusoidal input.

The output data in Fig. 35 is from a 1 V_{pp} 15 MHz sinusoidal input. Since the output voltage scales at 1 V/V with the input the time between the peak or valley of a sinusoid on the output to +/– 500 mV is ¼ of a cycle (the cursor in Fig. 35 shows ΔV = 500 mV and an amplitude cursor

was used to determine where to position the time cursor). The time difference shown for ¼ of a cycle is 79 μs on the output and the total output time is 138 μs (the total output time is defined as the time between discontinuities at the output; the trigger is set so the discontinuity is just off screen). This means that 79/138 is the ratio of the capture time used to read this portion of the signal which results in Eq. (5.3). This is also the time expected for the output period, the function generator used for the counter clock was set to 1 MHz but the measured frequency is 926 kHz or T = 1.08 μs, so 128 periods is 138 μs. The estimated propagation delay from this calculation is shown in Eq. (5.4).

$$t_{capture,inferred} = \frac{\left(\frac{1}{4} * \frac{1}{15\ MHz}\right)}{\left(\frac{79}{138}\right)} = \frac{1}{4} * \frac{1}{15\ MHz} * \frac{138}{79} = 29.1\ ns \qquad (5.3)$$

$$t_{delay,inferred} = \frac{t_{capture,inferred}}{128} = 227\ ps \qquad (5.4)$$

From Eqs. (2.14) and (3.2) the capture time was estimated to be 25.6 ns and the delay time was estimated to be 200 ps, in Fig. 18 the estimated capture time from the simulation was 28.3 ns. The estimates measured and calculated in this section are a close approximation to the mathematically derived and simulated values.

5.3 PULSE AND SINUSOIDAL INPUTS

This section will show the comparison of various inputs to their respective outputs, it will be divided into pages showing several varieties of signals with varying amplitudes. Inputs will be shown on the left in yellow and outputs will be shown on the right in blue, the output repeats every 138 microseconds after the initial falling edge however the oscilloscope settings are set so that only one cycle is shown. All test results shown are from the on-chip 50 Ω termination.

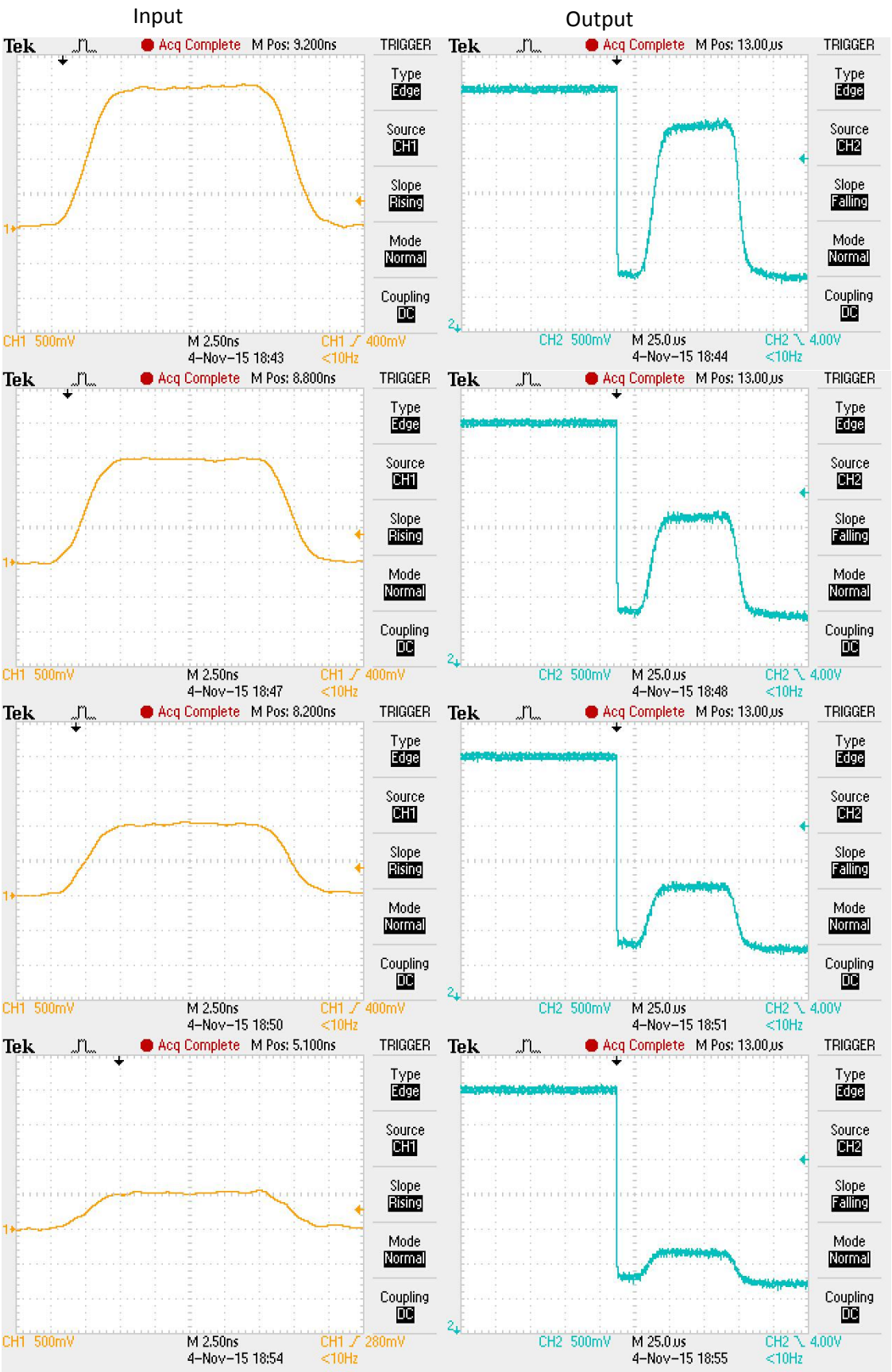

Figure 36: 20 ns pulse width with 2 V, 1.5 V, 1 V and 0.5 V amplitudes.

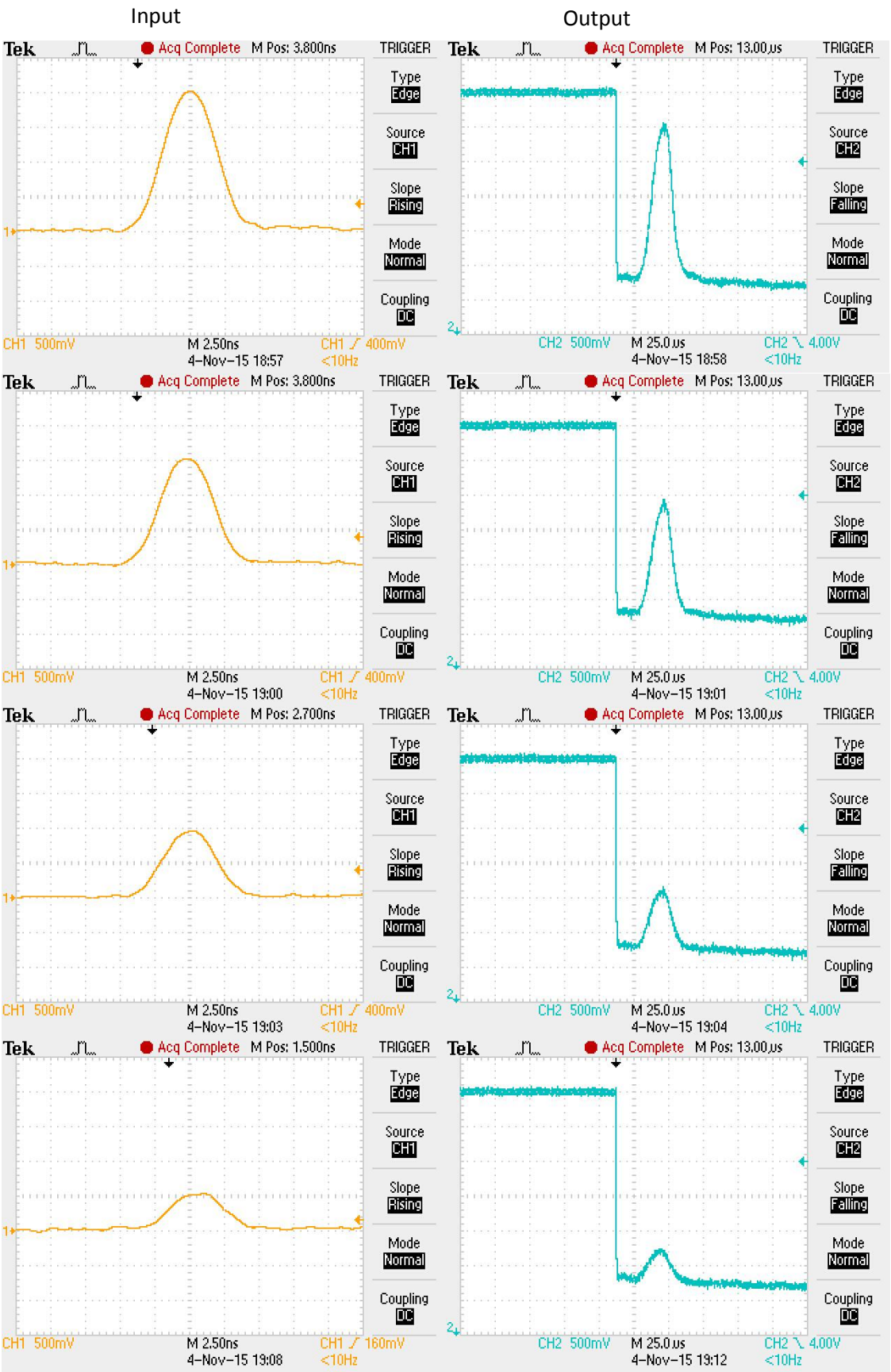

Figure 37: Minimum pulse width with 2 V, 1.5 V, 1 V and 0.5 V amplitudes.

Figure 38: 15 ns rise time with 2 V, 1.5 V, 1 V and 0.5 V amplitudes.

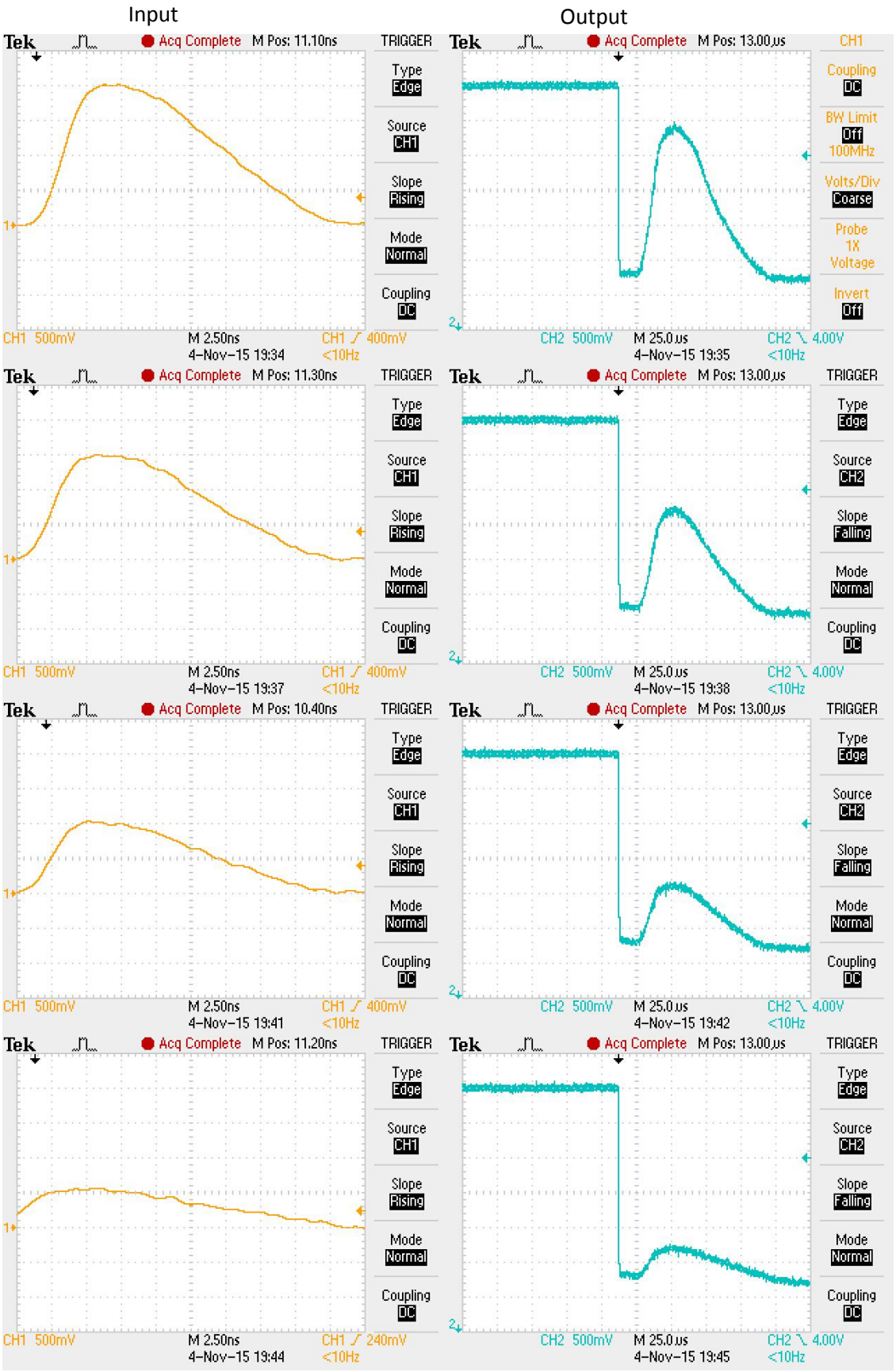

Figure 39: 15 ns fall time with 2 V, 1.5 V, 1 V and 0.5 V amplitudes.

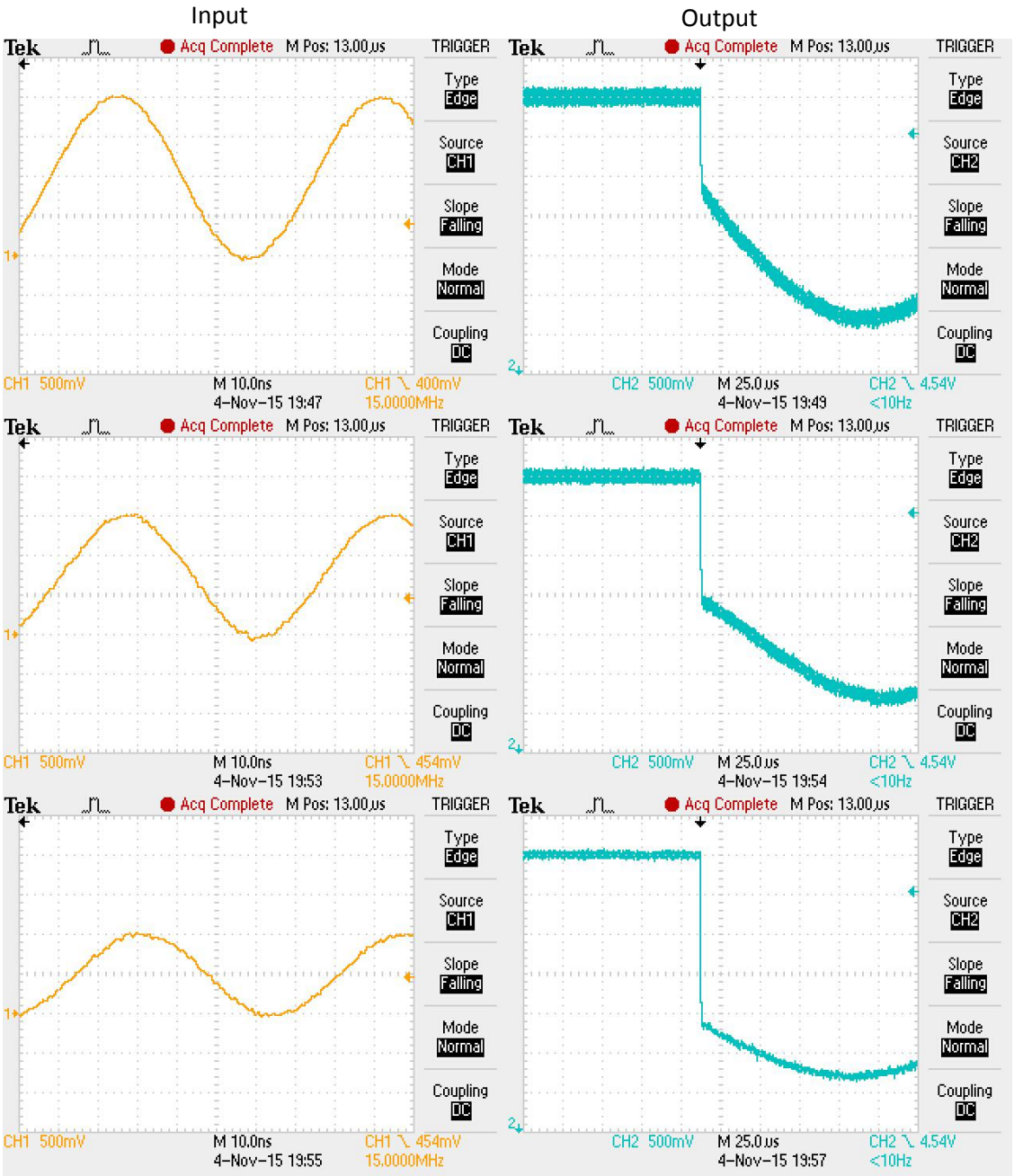

Figure 40: 15 MHz sinusoid with 2 V, 1.5 V and 1 V peak to peak amplitudes. (Signals are offset so that the minimum voltage is 0 V).

The circuit performs as expected for all of the inputs shown. The highest frequency available on the test equipment available is only 15 MHz so a full cycle cannot be shown on the sinusoidal inputs. However the circuit performs as expected from the discussion in section 5.2, the peak/valley to zero crossing takes approximately 78 µs for each of the input amplitudes.

5.4 TEST RESULTS COMPARISON TABLE

	Calculation	Simulation		Test	
Minimum Output Voltage	2.21 V	2.11 V		2.14 V	
Maximum Output Voltage	4.21 V	4.00 V		4.28 V	
Propagation Delay	200 ps	t_{PLH}	t_{PHL}	Measured	Inferred
		198.3 ps	201.4 ps	245 ps	227 ps
Capture Time	25.6 ns	28.3 ns		31.4 ns	29.1 ns

Table 1: Comparison table for calculations, simulations and results.

5.5 TEST SETUP AND EQUIPMENT

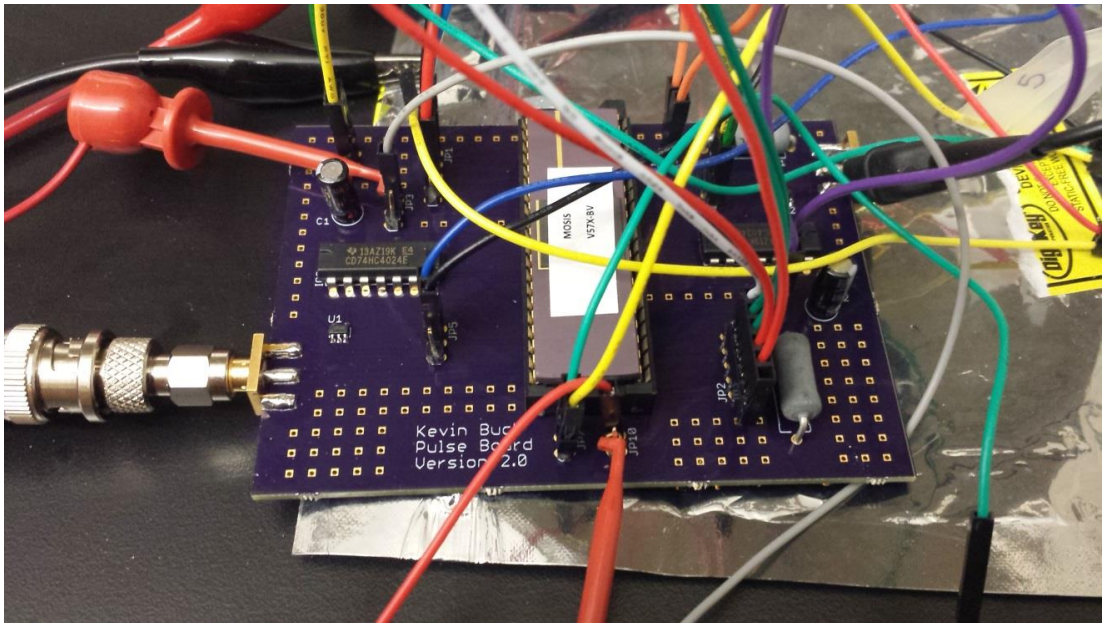

Figure 41: Transient and sinusoidal input test setup.

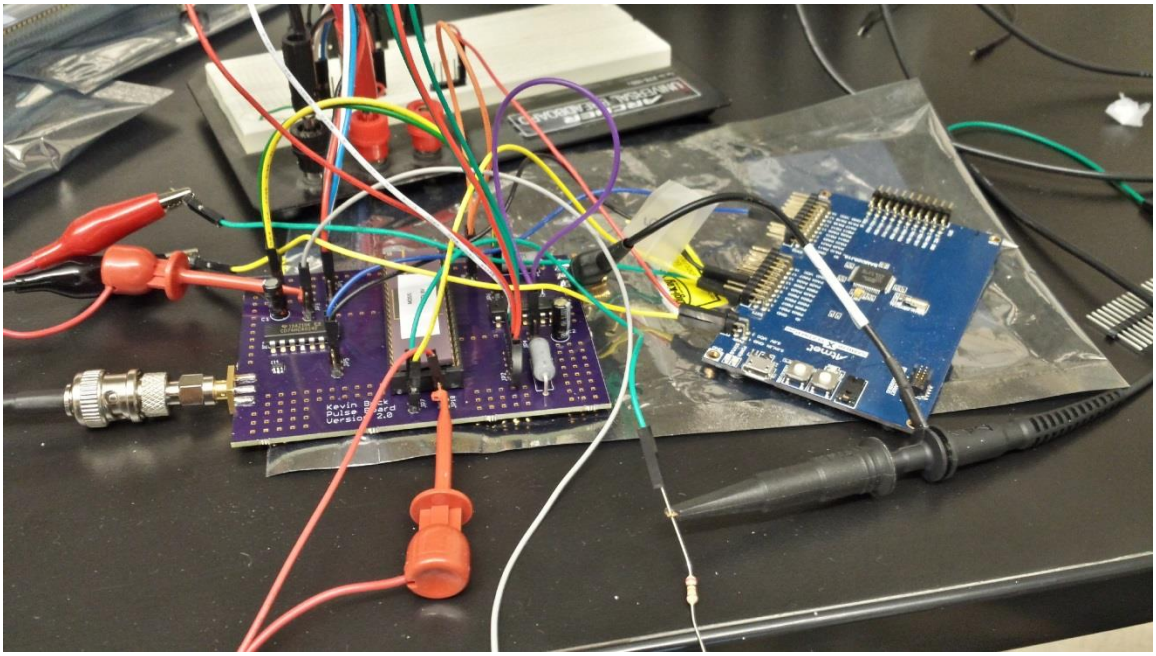

Figure 42: PCB and ATMEL SAMD20 microcontroller during testing. (Breadboard is used for additional *VDD* and ground connections).

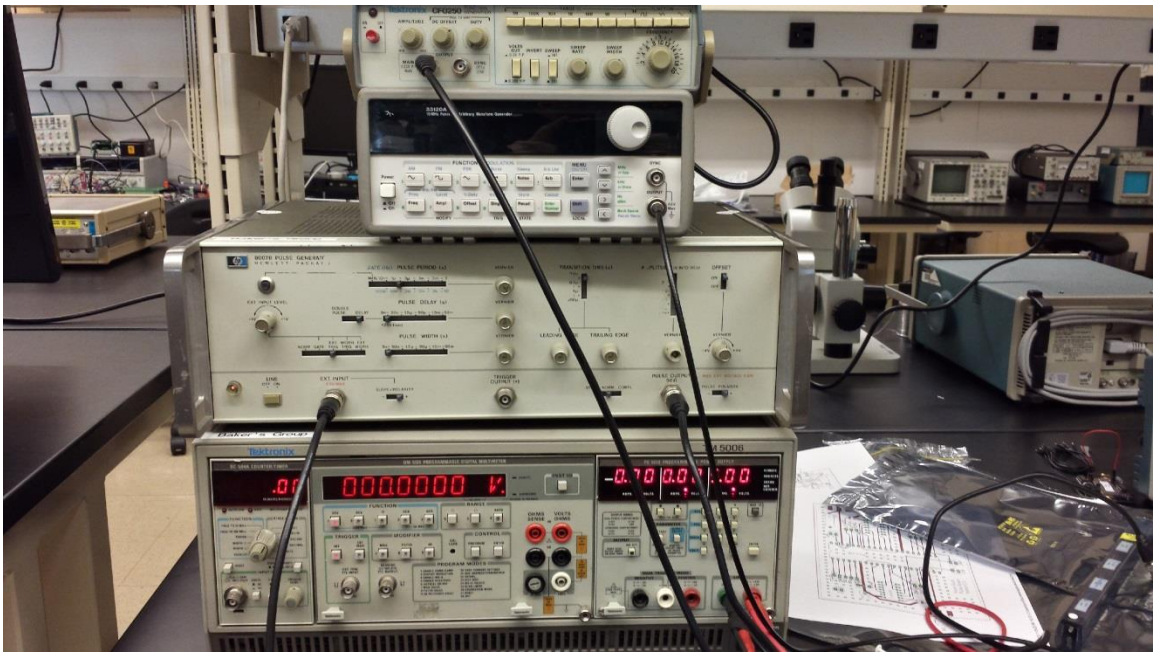

Figure 43: Test equipment. (Listed from top to bottom: function generator for clocking the counter, function generator for sinusoidal input, pulse generator for pulse input, DC power supply for *VDD* and precision DC test voltages).

CHAPTER 6: CONCLUSION AND FUTURE WORK

Electrical testing of the final circuit confirmed the calculations and simulation results shown in Chs. 2 and 3. The ASIC designed and outlined in this paper discretizes an analog input voltage over a 25 to 30 ns capture period and allows the user to read out the data at a selectable rate by varying the counter clock. The circuit can reliably recreate any signal applied to the input in the proper voltage range. In theory the circuit can handle periodic signals into the very low GHz range however the test equipment available was not sufficient to test signals at this rate. Test results are shown with the on-chip termination because the off-chip resistor introduces additional noise on the input. All of the signals shown in the test results are brief transients with an initial and final voltage of 0 V or high frequency sinusoids, for transient signals with a non-zero initial and/or final voltage or a low frequency sinusoid the off-chip termination must be used because the on-chip termination cannot handle a significant DC current.

Future work on this project includes implementing a bare die solution to reduce the parasitic effects of the packaged chip. The DC testing needs higher resolution test equipment for a more accurate characterization of the input and output voltage relationship. Output noise can be reduced by connecting the output directly to an SMA connector with a surface mounted resistor wired to *VDD* directly on the PCB. Input noise for the off-chip terminated circuit can be reduced by using a surface mounted resistor. The capture time of the circuit can be expanded by increasing the number of cells and control signals, every doubling of the capture time requires one more control signal. Testing up to the very high megahertz and very low gigahertz range needs to be performed to characterize the speed limitations of the circuit.

APPENDIX

These are the files taken from the Atmel Software Framework library that were modified for this work. There are other files in the code that are not included that were not modified. The file modifications made are as follows:

1. main.c – this was a blank skeleton code that was written from scratch other than the system_init() function. The code is a polling loop that checks the status of the button on the microcontroller and when the status changes the code is executed. When the button is pressed the read operation is executed, when it is not pressed the circuit remains in the sensing operation until the next time the button is pressed.
2. board_init.c – the only section of this code that was modified are the lines under the comment "Configure I/O pins." The modifications define the I/O pins used on the microcontroller as outputs and set them to the correct initial logic levels.
3. conf_clocks.h – this section of code was modified to run the microcontroller at the highest possible clock frequency (48 MHz), the modifications made have the original configuration shown as a comment next to the corresponding line.

main.c

```
#include <asf.h>

int main (void)
{
	//Initialize board
	system_init();

	//Previous state check
	int flag = 0;

	//Polling loop
	while (1) {
		// Is button pressed?
		if (port_pin_get_input_level(BUTTON_0_PIN) == BUTTON_0_ACTIVE) {
			// Yes, execute if previous state was unpressed.
			if(flag == 0){
				//Trigger signal to function generator
				PORT_IOBUS->Group[1].OUTTGL.reg = (1<<07);
				//Do nothing for 1 clock cycle (delay 21 ns)
				asm("nop");
				//Falling edge of clock signal to begin data capture
				PORT_IOBUS->Group[1].OUTTGL.reg = (1<<06);
				//Enable signal to read out data
				PORT_IOBUS->Group[0].OUTTGL.reg = (1<<20);
			}
			//Change previous state to pressed for next loop
			flag = 1;
		}else {
			// No, execute if previous state was pressed.
			if(flag == 1){
				//Reset I/O pins to sensing mode
				PORT_IOBUS->Group[1].OUTTGL.reg = (1<<06);
				PORT_IOBUS->Group[0].OUTTGL.reg = (1<<20);
				PORT_IOBUS->Group[1].OUTTGL.reg = (1<<07);
			}
			//Change previous state to unpressed for next loop
			flag = 0;
		}
	}
}
```

board_init.c

```
#include <compiler.h>
#include <board.h>
#include <conf_board.h>
#include <port.h>

#if defined(__GNUC__)
void board_init(void) WEAK __attribute__((alias("system_board_init")));
#elif defined(__ICCARM__)
void board_init(void);
#  pragma weak board_init=system_board_init
#endif

void system_board_init(void)
{
	struct port_config pin_conf;
	port_get_config_defaults(&pin_conf);

	/* Configure I/O pins */
	pin_conf.direction  = PORT_PIN_DIR_OUTPUT;
	port_pin_set_config(LED_0_PIN, &pin_conf);
	port_pin_set_config(EXT1_PIN_GPIO_0, &pin_conf);
	port_pin_set_config(EXT1_PIN_GPIO_1, &pin_conf);
	port_pin_set_config(EXT2_PIN_GPIO_0, &pin_conf);
	port_pin_set_config(EXT3_PIN_10, &pin_conf);
	port_pin_set_output_level(LED_0_PIN, LED_0_INACTIVE);
	port_pin_set_output_level(EXT1_PIN_GPIO_0, GPIO_0_ACTIVE);
	port_pin_set_output_level(EXT1_PIN_GPIO_1, GPIO_1_ACTIVE);
	port_pin_set_output_level(EXT2_PIN_GPIO_0, GPIO_2_ACTIVE);

	struct system_pinmux_config pinmux_config;
	system_pinmux_get_config_defaults( &pinmux_config );

	pinmux_config.mux_position = MUX_PA27H_GCLK_IO0;
	system_pinmux_pin_set_config(EXT3_PIN_10, &pinmux_config);

	/* Set buttons as inputs */
	pin_conf.direction  = PORT_PIN_DIR_INPUT;
	pin_conf.input_pull = PORT_PIN_PULL_UP;
	port_pin_set_config(BUTTON_0_PIN, &pin_conf);
#ifdef CONF_BOARD_AT86RFX
	port_get_config_defaults(&pin_conf);
	pin_conf.direction  = PORT_PIN_DIR_OUTPUT;
	port_pin_set_config(AT86RFX_SPI_SCK, &pin_conf);
	port_pin_set_config(AT86RFX_SPI_MOSI, &pin_conf);
	port_pin_set_config(AT86RFX_SPI_CS, &pin_conf);
	port_pin_set_config(AT86RFX_RST_PIN, &pin_conf);
	port_pin_set_config(AT86RFX_SLP_PIN, &pin_conf);
	port_pin_set_output_level(AT86RFX_SPI_SCK, true);
	port_pin_set_output_level(AT86RFX_SPI_MOSI, true);
	port_pin_set_output_level(AT86RFX_SPI_CS, true);
	port_pin_set_output_level(AT86RFX_RST_PIN, true);
	port_pin_set_output_level(AT86RFX_SLP_PIN, true);
#ifdef EXT_RF_FRONT_END_CTRL
	port_pin_set_config(AT86RFX_CPS, &pin_conf);
	port_pin_set_output_level(AT86RFX_CPS, HIGH);
	port_pin_set_config(AT86RFX_CSD, &pin_conf);
```

```
        port_pin_set_output_level(AT86RFX_CSD, HIGH);
#endif

        pin_conf.direction  = PORT_PIN_DIR_INPUT;
        port_pin_set_config(AT86RFX_SPI_MISO, &pin_conf);

#endif
}
```

conf_clocks.h

```
#include <clock.h>

#ifndef CONF_CLOCKS_H_INCLUDED
#  define CONF_CLOCKS_H_INCLUDED

/* System clock bus configuration */
#  define CONF_CLOCK_CPU_CLOCK_FAILURE_DETECT     false
#  define CONF_CLOCK_FLASH_WAIT_STATES            1
#  define CONF_CLOCK_CPU_DIVIDER                  SYSTEM_MAIN_CLOCK_DIV_1
#  define CONF_CLOCK_APBA_DIVIDER                 SYSTEM_MAIN_CLOCK_DIV_1
#  define CONF_CLOCK_APBB_DIVIDER                 SYSTEM_MAIN_CLOCK_DIV_1

/* SYSTEM_CLOCK_SOURCE_OSC8M configuration - Internal 8MHz oscillator */
#  define CONF_CLOCK_OSC8M_PRESCALER              SYSTEM_OSC8M_DIV_1
#  define CONF_CLOCK_OSC8M_ON_DEMAND              true
#  define CONF_CLOCK_OSC8M_RUN_IN_STANDBY         true//false

/* SYSTEM_CLOCK_SOURCE_XOSC configuration - External clock/oscillator */
#  define CONF_CLOCK_XOSC_ENABLE                  false
#  define CONF_CLOCK_XOSC_EXTERNAL_CRYSTAL        SYSTEM_CLOCK_EXTERNAL_CRYSTAL
#  define CONF_CLOCK_XOSC_EXTERNAL_FREQUENCY      12000000UL
#  define CONF_CLOCK_XOSC_STARTUP_TIME            SYSTEM_XOSC_STARTUP_32768
#  define CONF_CLOCK_XOSC_AUTO_GAIN_CONTROL       true
#  define CONF_CLOCK_XOSC_ON_DEMAND               true
#  define CONF_CLOCK_XOSC_RUN_IN_STANDBY          true//false

/* SYSTEM_CLOCK_SOURCE_XOSC32K configuration - External 32KHz crystal/clock oscillator */
#  define CONF_CLOCK_XOSC32K_ENABLE               true//false
#  define CONF_CLOCK_XOSC32K_EXTERNAL_CRYSTAL     SYSTEM_CLOCK_EXTERNAL_CRYSTAL
#  define CONF_CLOCK_XOSC32K_STARTUP_TIME         SYSTEM_XOSC32K_STARTUP_65536
#  define CONF_CLOCK_XOSC32K_AUTO_AMPLITUDE_CONTROL  false
#  define CONF_CLOCK_XOSC32K_ENABLE_1KHZ_OUPUT    false
#  define CONF_CLOCK_XOSC32K_ENABLE_32KHZ_OUTPUT  true
#  define CONF_CLOCK_XOSC32K_ON_DEMAND            true
#  define CONF_CLOCK_XOSC32K_RUN_IN_STANDBY       true//false

/* SYSTEM_CLOCK_SOURCE_OSC32K configuration - Internal 32KHz oscillator */
#  define CONF_CLOCK_OSC32K_ENABLE                false
#  define CONF_CLOCK_OSC32K_STARTUP_TIME          SYSTEM_OSC32K_STARTUP_130
#  define CONF_CLOCK_OSC32K_ENABLE_1KHZ_OUTPUT    true
#  define CONF_CLOCK_OSC32K_ENABLE_32KHZ_OUTPUT   true
#  define CONF_CLOCK_OSC32K_ON_DEMAND             true
#  define CONF_CLOCK_OSC32K_RUN_IN_STANDBY        false

/* SYSTEM_CLOCK_SOURCE_DFLL configuration - Digital Frequency Locked Loop */
#  define CONF_CLOCK_DFLL_ENABLE                  true//false
#  define CONF_CLOCK_DFLL_LOOP_MODE               SYSTEM_CLOCK_DFLL_LOOP_MODE_CLOSED
#  define CONF_CLOCK_DFLL_ON_DEMAND               false
#  define CONF_CLOCK_DFLL_RUN_IN_STANDBY          true//false

/* DFLL open loop mode configuration */
#  define CONF_CLOCK_DFLL_COARSE_VALUE            (0x1f / 4)
#  define CONF_CLOCK_DFLL_FINE_VALUE              (0xff / 4)
```

```
/* DFLL closed loop mode configuration */
#  define CONF_CLOCK_DFLL_SOURCE_GCLK_GENERATOR   GCLK_GENERATOR_1
#  define CONF_CLOCK_DFLL_MULTIPLY_FACTOR        (48000000 / 32768UL)
#  define CONF_CLOCK_DFLL_QUICK_LOCK             false//true
#  define CONF_CLOCK_DFLL_TRACK_AFTER_FINE_LOCK   false//true
#  define CONF_CLOCK_DFLL_KEEP_LOCK_ON_WAKEUP     true
#  define CONF_CLOCK_DFLL_ENABLE_CHILL_CYCLE      true
#  define CONF_CLOCK_DFLL_MAX_COARSE_STEP_SIZE    3//(0x1f / 4)
#  define CONF_CLOCK_DFLL_MAX_FINE_STEP_SIZE      1//(0xff / 4)


/* Set this to true to configure the GCLK when running clocks_init. If set to
 * false, none of the GCLK generators will be configured in clocks_init(). */
#  define CONF_CLOCK_CONFIGURE_GCLK               true

/* Configure GCLK generator 0 (Main Clock) */
#  define CONF_CLOCK_GCLK_0_ENABLE                true
#  define CONF_CLOCK_GCLK_0_RUN_IN_STANDBY        false
#  define CONF_CLOCK_GCLK_0_CLOCK_SOURCE
SYSTEM_CLOCK_SOURCE_DFLL//SYSTEM_CLOCK_SOURCE_OSC8M
#  define CONF_CLOCK_GCLK_0_PRESCALER             1
#  define CONF_CLOCK_GCLK_0_OUTPUT_ENABLE         true

/* Configure GCLK generator 1 */
#  define CONF_CLOCK_GCLK_1_ENABLE                true//false
#  define CONF_CLOCK_GCLK_1_RUN_IN_STANDBY        false
#  define CONF_CLOCK_GCLK_1_CLOCK_SOURCE          SYSTEM_CLOCK_SOURCE_XOSC32K
#  define CONF_CLOCK_GCLK_1_PRESCALER             1
#  define CONF_CLOCK_GCLK_1_OUTPUT_ENABLE         false

/* Configure GCLK generator 2 (RTC) */
#  define CONF_CLOCK_GCLK_2_ENABLE                false
#  define CONF_CLOCK_GCLK_2_RUN_IN_STANDBY        false
#  define CONF_CLOCK_GCLK_2_CLOCK_SOURCE          SYSTEM_CLOCK_SOURCE_OSC32K
#  define CONF_CLOCK_GCLK_2_PRESCALER             32
#  define CONF_CLOCK_GCLK_2_OUTPUT_ENABLE         false

/* Configure GCLK generator 3 */
#  define CONF_CLOCK_GCLK_3_ENABLE                false
#  define CONF_CLOCK_GCLK_3_RUN_IN_STANDBY        false
#  define CONF_CLOCK_GCLK_3_CLOCK_SOURCE          SYSTEM_CLOCK_SOURCE_OSC8M
#  define CONF_CLOCK_GCLK_3_PRESCALER             1
#  define CONF_CLOCK_GCLK_3_OUTPUT_ENABLE         false

/* Configure GCLK generator 4 */
#  define CONF_CLOCK_GCLK_4_ENABLE                false
#  define CONF_CLOCK_GCLK_4_RUN_IN_STANDBY        false
#  define CONF_CLOCK_GCLK_4_CLOCK_SOURCE          SYSTEM_CLOCK_SOURCE_OSC8M
#  define CONF_CLOCK_GCLK_4_PRESCALER             1
#  define CONF_CLOCK_GCLK_4_OUTPUT_ENABLE         false

/* Configure GCLK generator 5 */
#  define CONF_CLOCK_GCLK_5_ENABLE                false
#  define CONF_CLOCK_GCLK_5_RUN_IN_STANDBY        false
#  define CONF_CLOCK_GCLK_5_CLOCK_SOURCE          SYSTEM_CLOCK_SOURCE_OSC8M
#  define CONF_CLOCK_GCLK_5_PRESCALER             1
#  define CONF_CLOCK_GCLK_5_OUTPUT_ENABLE         false
```

```
/* Configure GCLK generator 6 */
#  define CONF_CLOCK_GCLK_6_ENABLE                false
#  define CONF_CLOCK_GCLK_6_RUN_IN_STANDBY        false
#  define CONF_CLOCK_GCLK_6_CLOCK_SOURCE          SYSTEM_CLOCK_SOURCE_OSC8M
#  define CONF_CLOCK_GCLK_6_PRESCALER             1
#  define CONF_CLOCK_GCLK_6_OUTPUT_ENABLE         false

/* Configure GCLK generator 7 */
#  define CONF_CLOCK_GCLK_7_ENABLE                false
#  define CONF_CLOCK_GCLK_7_RUN_IN_STANDBY        false
#  define CONF_CLOCK_GCLK_7_CLOCK_SOURCE          SYSTEM_CLOCK_SOURCE_OSC8M
#  define CONF_CLOCK_GCLK_7_PRESCALER             1
#  define CONF_CLOCK_GCLK_7_OUTPUT_ENABLE         false

#endif /* CONF_CLOCKS_H_INCLUDED */
```

REFERENCES

[1] Baker, R. (2010). The Inverter. CMOS Circuit Design, Layout and Simulation (3rd ed., pp. 105-160, 331-352). Hoboken, NJ, John Wiley & Sons.

[2] Sedra, A. and Smith, K. (1998). Field-Effect Transistors (FETs). Microelectronic Circuits (4th ed. pp 363). New York, NY, Oxford University Press.

[3] MOSIS SPICE parameters. (n.d.). BSIM3 Models for AMI Semiconductor's C5 Process. Retrieved August 8, 2014, from http://cmosedu.com/jbaker/courses/ee421_ecg621/f14/ee421_ecg621.htm

CURRICULUM VITAE

Graduate College

University of Nevada, Las Vegas

Kevin Buck

Degree:

Bachelor of Science, Electrical and Computer Engineering, 2008

University of Nevada, Las Vegas

Employment:

University of Nevada, Las Vegas (UNLV) 2014 – 2015

Intel Corporation, 2007

Hewlett-Packard, 2006

Thesis Title: Fast Transient Digitizer and PCB Interface

Thesis Examination Committee:

Chairperson, R. Jacob Baker, Ph. D.

Committee Member, Brendan Morris, Ph. D.

Committee Member, Peter Stubberud, Ph. D.

Committee Member, Yi-Tung Chen, Ph. D.