2012

EECE 280 B2

Red (Kyle) Garsuta
Paul Cheng
Kevin Chang

ADVANCED MULTIMETER WITH VHDL

Contents

1 Introduction 2
1.1 Specifications 2
1.2 Overview and Design Approach 2
2 Description and Evaluation of Individual Blocks 3
2.1 The DE2 Board 3
2.2 State Machine 3
2.2.1 States and State Transitions 4
2.2.2 Testing 5
2.3 Analog to Digital Converter 5
2.3.1 Accuracy 6
2.4 Voltmeter 6
2.4.1 Development and Testing 6
2.5 Ammeter 6
2.5.1 Testing 8
2.6 Ohmmeter 8
2.7 Beta Meter 9
2.8 The Casing 9
3 Evaluation of the Complete System 10
3.1 Testing 10
3.2 Strengths and Weaknesses of the Design 10
4 Conclusions 10
5 References 11
6 Appendices 12
6.1 Appendix A: Voltmeter Conversion VHDL Code 12
6.2 Appendix B: Binary to BCD VHDL Code 12
6.3 Appendix C: Snapshot of the Final Product 13

1 Introduction

1.1 Specifications

The specifications were to design, build, and test a digital multimeter. The logic component of the multimeter must be implemented into the FPGA of the Altera DE2 board using VHDL. The requirements are as follows. [1]

1. The multimeter is powered with a supply of our own design, using the transformers available in the laboratory.
2. All the circuitry must be built using solder boards, and properly packed in a metal or non-flammable box (we are allowed to purchase a box or build our own).
3. The multimeter uses the 7-segment LED display of the Altera board to show measurements with a resolution of at least 4 ½ digits.

The multimeter measures:

4. Voltage in the ranges: ±0.20000, ±2.0000, and ±20.000 DC V with automatic range change.
5. Current in the ranges: ±2.0000, ±20.000, and ±200.00 DC mA with automatic range change; should be able to measure current not connected to common ground.
6. Resistance in the range 10 to 2M Ω with automatic range change.
7. The Beta of NPN and PNP transistors.

1.2 Overview and Design Approach

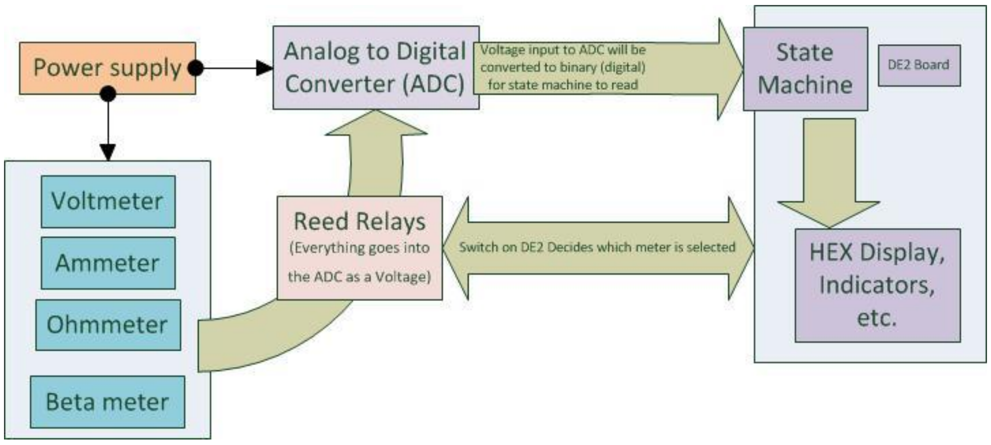

Power supply
Analog to Digital Converter (ADC)
Voltage input to ADC will be converted to binary (digital) for state machine to read
State Machine
DE2 Board
Voltmeter
Ammeter
Ohmmeter
Beta meter
Reed Relays (Everything goes into the ADC as a Voltage)
Switch on DE2 Decides which meter is selected
HEX Display, Indicators, etc.

Figure 1.1: Top-level schematic

The design of this project began with a simple idea of expandability. Looking far beyond the duration of this project, we wanted to make a digital multimeter with features that can easily be expanded (i.e. to measure barometric pressure, etc.) by attaching the appropriate hardware with minor modification to the source code.

[1] Jesus Calvino-Fraga, *Project 2: Advanced Multimeter Using VHDL* (University of British Columbia, 2012), p.1.

The Digital Multimeter (DMM) design is comprised of five main blocks: (1) The Power Supply, (2) The Meters, (3) The Reed Relays, (4) The Analog to Digital Converter (ADC), and (5) The DE2 Board[2]. The power supply is connected to a step-down transformer available in laboratory, and power to the meters and the ADC. All of the timing is done via the State Machine programmed in the DE2 using VHDL and communication between the DE2 and the rest of the circuitry is done via a 40-pin GPIO expansion header[3]. Switching between the different meters is done by flipping physical switches on the DE2, consequently enabling the appropriate circuits using the Reed Relays and displaying the corresponding readings on the DE2 7-Segment displays. Figure 1.1 shows how each of the components mentioned above interact.

2 Description and Evaluation of Individual Blocks

2.1 The DE2 Board

The Altera DE2 Board serves as the 'brain' of the DMM and is programmed with the state machine implementation. The physical switches on the DE2 serve to toggle between different modes (i.e. voltmeter, ammeter); the state machine then responds to accordingly. In the next section we delve into details of the state machine implantation.

2.2 State Machine

The state machine is programmed on the DE2 Board using Quartus II[4] and VHDL. Figure 2.1 shows the top-level block diagram of the software implementation in Quartus. The left-most component is the *State Machine*, which is connected to various inputs and outputs, including the 40-pin GPIO header that controls the board's interaction with outside[5] circuitry.

The state machine takes a 16 bit binary input from the ADC and converts it into the corresponding measured value, depending on the mode of operation (i.e. into the appropriate voltage in voltmeter mode, etc.). This is done by multiplication by corresponding factors the convert the value read by the ADC to the value we want to display as the meter reading. Please see Appendix A for a sample conversion. Note the multiplication "inVolt," the voltage factor that we discussed. A subsequent subtraction by "offsetV" is done to compensate for the internal impedance of the ADC and other circuitry[6]. We will not go into the specific conversions for each meter nor details of the pin connections but instead focus on the discussing the software functionality.

[2] Altera Development and Education Board. For more information: http://www.altera.com/education/univ/materials/boards/de2/unv-de2-board.html
[3] Altera Corporation, *Development* and Education Board User Manual (2006), p. 35.
[4] Quartus II is a programmable logic device design software by Altera Corp.
[5] We refer to circuitry *outside* the DE2 Board.
[6] Analog Devices, Inc., *AD976/AD976A Datasheet* (Norwood, 1999), p.2.

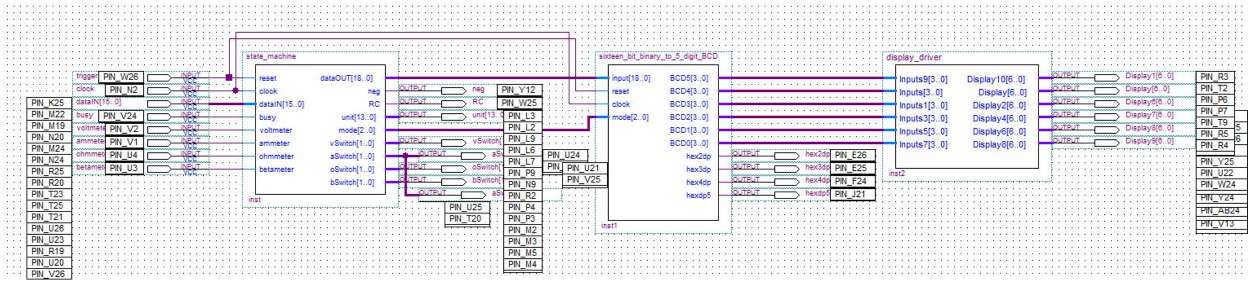


Figure 2.1: State machine transition diagram

The component in the middle, the *Binary to BCD Decoder* takes a binary input from the state machine which represents the final value to be displayed, then extracts each digit and then converts it into BCD. The algorithm we developed involves converting the binary number into an integer in VHDL, then utilizes division by a tenth factor less than the most significant digit, then taking the remainder of the division and then dividing by a tenth factor less, repeating this until all digits are extracted. Please refer to Appendix B for snippets of the code. In the next section, we go into the details of the state machine transitions.

2.2.1 States and State Transitions

As seen on the transition state diagram in Figure 2.2, the state machine cycles through three main states. When a reset is triggered, the state is set to State ‘00’[7]. In this state, the Read/Convert Input (RC) on the AD976 is set to logic 0. A falling edge on RC initializes the ADC conversion, and the state machine moves on to State ‘01’.

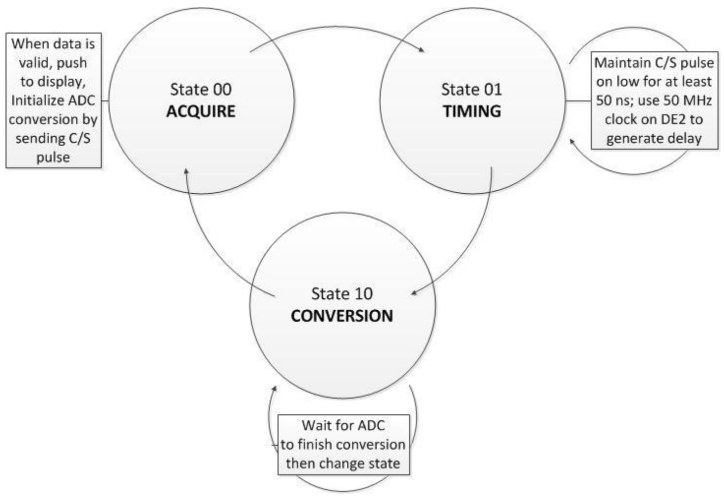

When data is valid, push to display, Initialize ADC conversion by sending C/S pulse
State 00
ACQUIRE
State 01
TIMING
Maintain C/S pulse on low for at least 50 ns; use 50 MHz clock on DE2 to generate delay
State 10
CONVERSION
Wait for ADC to finish conversion then change state

Figure 2.2: State machine transition diagram

[7] State numbers were chosen as 2-digit binary representations of the decimals 1 to 3 (i.e. 00, 01, 10).

The state machine stays in the Timing state until the 'timer' limit is reached. The 'timer' signal serves to generate a time delay and is incremented by 1 with each loop. Once the desired time has elapsed, the state machine moves on to State ‘10’.

In the Conversion state, the state machine transitions to State 00 when the BUSY signal from the AD976 is a logic 1, indicating that the conversion has completed and the data is ready to be read in the next state. Figure 2.3 shows the states in relation to the AD976 conversion timing. Note that the ‘state’ row was added to highlight state machine – ADC relationship, while the rest of the diagram was taken from the AD976 Datasheet[8].

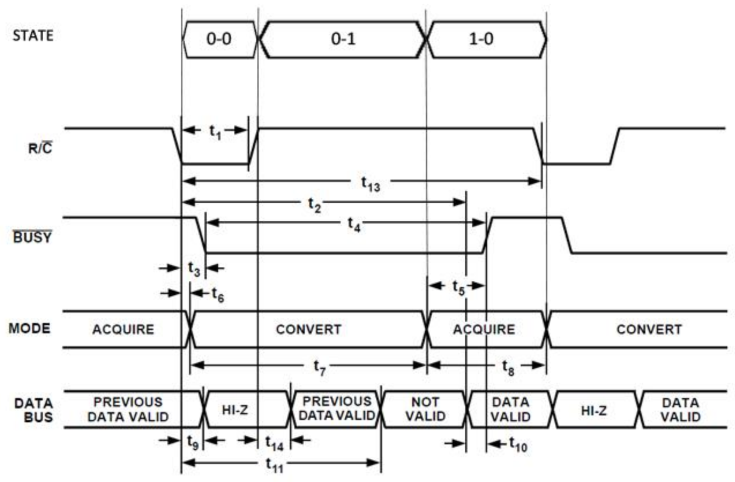


Figure 2.3: State machine timing

2.2.2 Testing

Because the State machine is one of the most crucial components of the DMM, a real-time approach was carried out in the testing process. Every minor change to the source code was immediately tested by uploading the software into the DE2 via JTAG mode. This offered a quick and reliable way of testing the functionality with close to real-time feedback. With each software modification, the behaviour was immediately tested against various inputs, comparing the output to the desired behaviour.

2.3 Analog to Digital Converter

The AD976 is a high speed, low power 16-bit A/D converter that operates from a single 5 V supply. The analog full-scale input is the standard industrial range of ±10 V. [9] Figure 2.4 shows the State Machine – ADC relationship.

[8] Analog Devices, Inc., *AD976/AD976A Datasheet* (Norwood, 1999), p.8.
[9] Ibid., p.1.

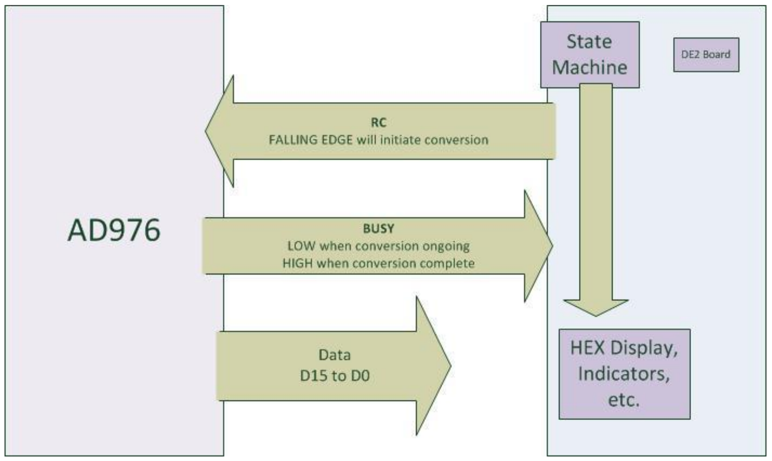

AD976
RC
FALLING EDGE will initiate conversion
BUSY
LOW when conversion ongoing
HIGH when conversion complete
Data
D15 to D0
State Machine
DE2 Board
HEX Display, Indicators, etc.

Figure 2.4: State machine – ADC Interface

2.3.1 Accuracy

As a consequence of the ±10 V threshold and 16 bit signed output[10] of the ADC, we are able to detect a minimum of approximately 0.00061 V change in voltage. As an example, calculations for the voltmeter are shown in Figure 2.5.

$$\frac{V_{max}}{2^{xbits}} = \frac{20V}{2^{15}} \cong 0.00061035V$$

Figure 2.5: Voltmeter calculations

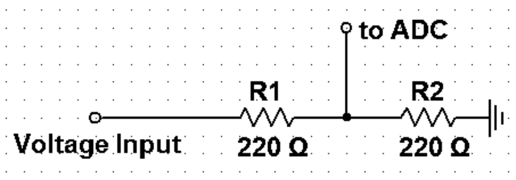


Figure 2.4: Voltmeter schematic

2.4 Voltmeter

Because the maximum threshold of the ADC was ±10 V and we needed to measure up to ±20 V, a simple divider circuit was implemented as seen on Figure 2.4.

2.4.1 Development and Testing

The voltmeter was developed alongside and tested together with the State machine. As such, the voltmeter provided a way of directly testing the State machine code functionality while the component itself was assessed continually.

2.5 Ammeter

The design of ammeter is based on the current to voltage converter circuit. The converter converts an input current to an equal voltage value with different unit.

Our first design is based on an operational amplifier. In figure below the 10V voltage source and 1000Ohm resistor will simulate a circuit with current 10mA. With a voltmeter we are able to

[10] The ADC has a 2's complement signed output.

obtain a current equivalent voltage. This design was not used due to the power requirement of the operational amplifier (TL072CP).

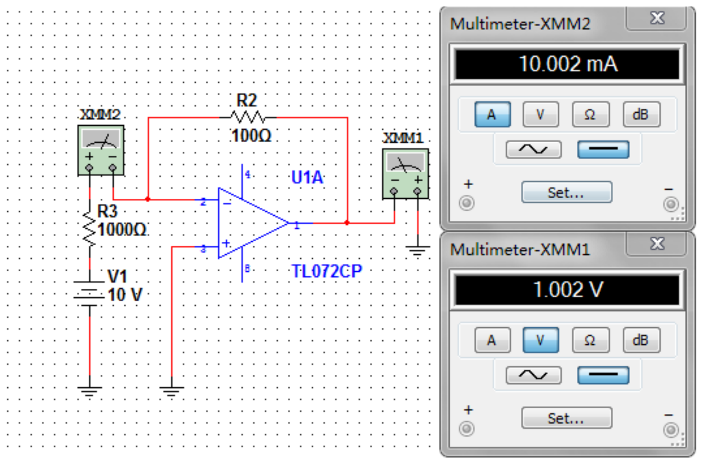


Figure 2.5: Ammeter Schematic 1 – Gain 100

	Expected	**Simulation**
Tested current	$\frac{10V}{1000\ Ohm} = 10mA$	$10.002mA$
Voltage equivalent current	$10\times100 = 1000mV = 1V$	$1.002V$

The second design uses two resistors connected in parallel as a current to voltage converter. The second design is designed with a gain of 50.

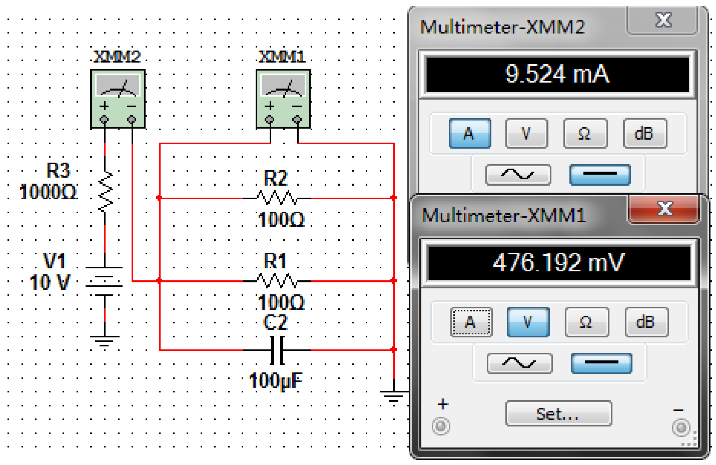


Figure 2.6: Ammeter Schematics 2 – Gain 50

	Expected	**Simulation**
Tested current	$\frac{10V}{1000\ Ohm} = 10mA = 0.01A$	$9.542mA$
V equivalent current	$10\times50 = 500mV$	$476.19mV$

2.5.1 Testing

When the Ammeter was combined with voltmeter we observe a linear offset with the expected value. We plotted the offset and applied it accordingly. Below is a graph of this behaviour.

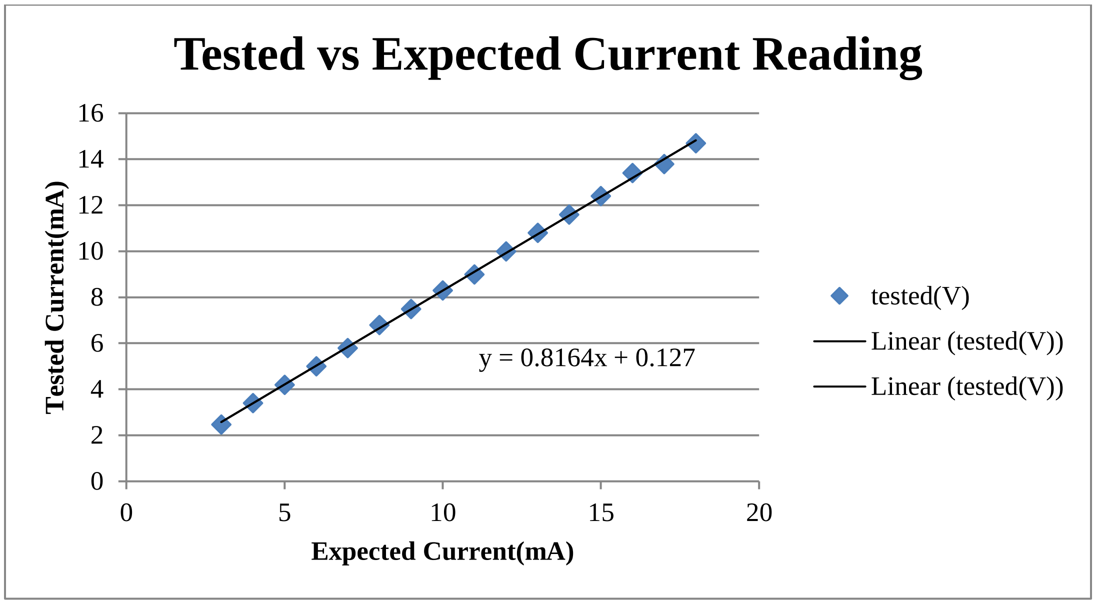


Figure 2.7: Equation Graph of Data Correction

2.6 Ohmmeter

The idea of our design on ohm meter is to build a constant current source which could create a constant current through the tested resistor and create a voltage which will be the input voltage to our analog-digital converter. We build two constant current sources which will change back and forth depends on the value of the resistor. The detail design is described below.

The first constant current source is made of two 1N4148 diode, one 2N3906 PNP transistor and 2 resistors. We set the current source at 5 micro amperes; thus, when we test the 2M resistor, which is the largest value of our resistor range, we will create a 10 volts input voltage. The schematic of this current source is provided below.

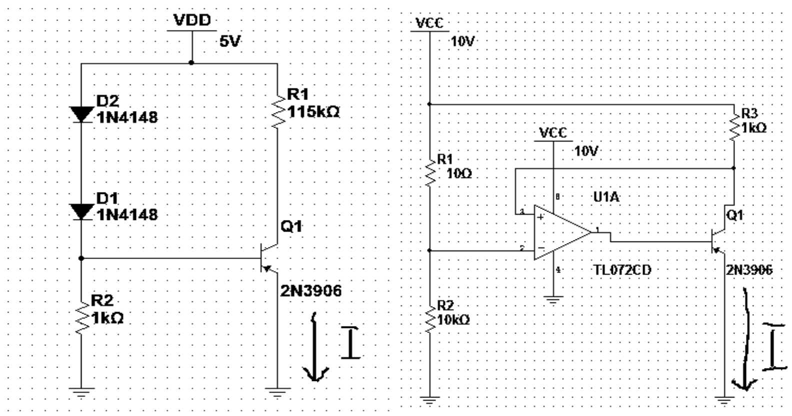


Figure 2.7: Ohmmeter Schematics

However, in reality, our current changes while we test a large resistor value, such as 1M. This forces us to create another constant current source which can still produce 5 micro amps with a large value resistor. Our second constant current source uses an operational amplifier, one 2N3906 PNP transistor and 3 resistors to create a new constant current source. This current source allows us to create 5 micro amperes while testing a large value of resistor. However, when we test small resistor value such as 10 ohms or 100 ohms, the current source increase its current output which leads to an inaccurate voltage output.

2.7 Beta Meter

The idea of our beta meter is that we set a constant current go through the base of a transistor and put an ammeter to test out the current on the emitter side. After we convert the amperes we detected into voltage, we transmit the voltage into our DE2 board, and use VHDL to manipulate the result into beta. By following the formula,

$$\beta = \frac{I_c}{I_b}$$

we should get the beta easily through a calculation in VHDL. However, this design did not work properly and we end up exclude the beta meter in our project.

2.8 The Casing

With the concept of expandability in mind, we saw the DMM as sort of *expansion* of the DE2 Board. As such, we decided to build a box inspired by the durability and practicality of the DE2 Board, with a quick hint of “electronic coolness” by similarly using a transparent plastic case to show off the circuitry while protecting it at the same time.

The box consists of two main parts, a transparent acrylic cover and a metal base. The base and cover is held together with bolts and nuts. The circuit board is placed between the cover and base. To avoid any short circuit, plastic spacer is used to separate the metal base with the back of the circuit board.

The transparent acrylic cover gives us a clear view on what is in our project. The acrylic cover also has cut outs for the 40 pin GPIO header. The openings at the top and sides make it easy to connect the input power and expansion headers. The entire box is held together with just few bolts and nuts; this allows us to easily assemble and dissembled the box for trouble shooting. As such, we designed a simpler, more elegant, and modernized *'box'*, far from its traditional definition. Appendix C shows a snapshot of the finished product.

3 Evaluation of the Complete System

3.1 Testing

Sufficient testing was done for the Voltmeter and Ammeter that the designs came together with no problems. The Ohmmeter and Beta meter, on the other hand, met numerous complications. As such, Even though each individual component functioned properly during their individual testing stages, only the Voltmeter and Ohmmeter functioned as planned.

3.2 Strengths and Weaknesses of the Design

The State Machine design was tested quite heavily without fail and thus, we are confident with the durability of the design. It also makes sense that the rest of the components are heavily dependent on the State Machine.

Despite heavy testing and modifications to the design, there are inherent concerns with the accuracy of our design. First of all, there is a considerable amount of noise in the various circuits and readings for all the meters fluctuate. Nevertheless, there is no fluctuation on the first three significant bits (i.e. For a 19.753 V reading, digits 5 and 3 fluctuate by approximately ±00.010 V). Secondly, as mentioned in Section 2.3.1, the minimum change in voltage that can be detected by the meter is approximately 0.00061 V, thus rendering the DMM unable to measure any changes less than this value.

4 Conclusions

The team collectively spent a total of approximately 240 hours on this project. With the use of project management software, we were able to track each task assignment and approximate the time it took to finish corresponding tasks. Regardless of having encountered problems with group cooperation, eventually landing the team with less manpower, the group was able to follow through and systematically step through the design process and successfully complete the project.

Despite not having accomplished all the individual requirements, a significant part of our DMM (namely the Voltmeter and Ammeter) is functional. As it stands, it is quite simple to systematically add any other meters to the DMM as long as each reading can be converted to a voltage. This can easily be done with few modifications to the State Machine code. As such, this project was successfully completed with an unwavering commitment to the original prospect of expandability in mind.

5 References

Altera Corporation. *Development and Education Board User Manual.* (2006).

Analog Devices. *AD976/AD976A Datasheet.* (Norwood, 1999).

Calvino-Fraga, Jesus. *Module 4: Counters and Amplifiers.* (Vancouver: University of British Columbia, 2012).

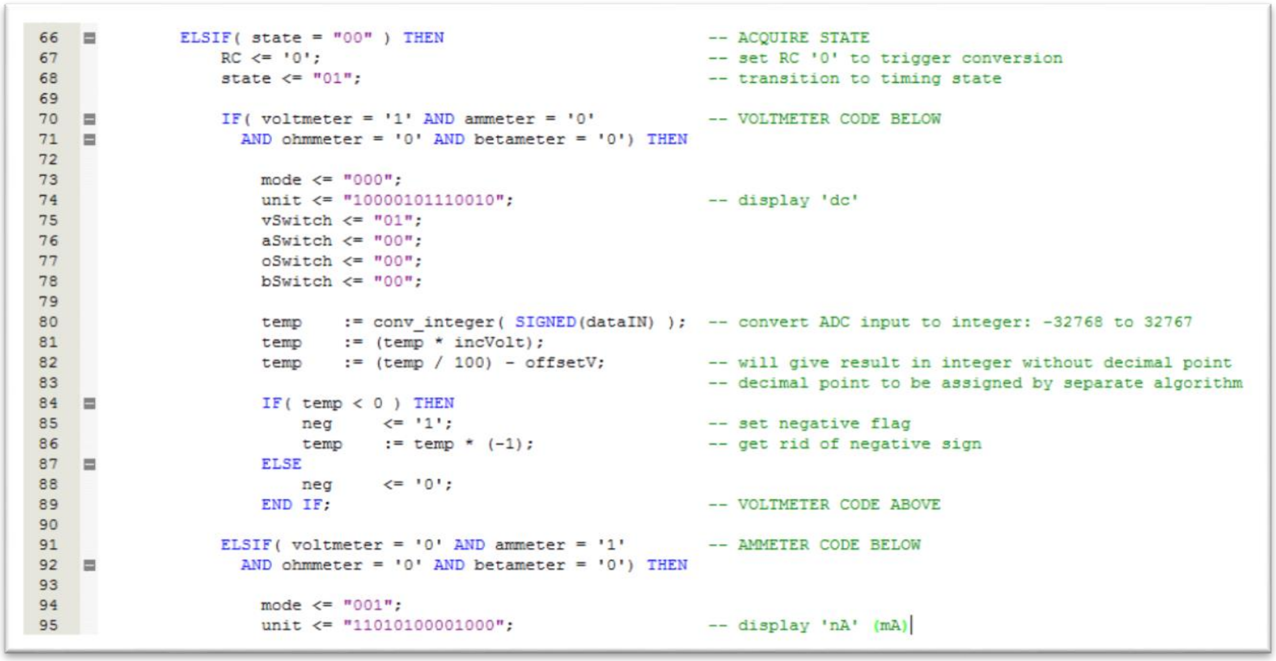
Calvino-Fraga, Jesus. *Module 5: Diodes Bipolar Junction Transistors.* (Vancouver: University of British Columbia, 2012).

Calvino-Fraga, Jesus. *Module 6: Power Supply Design.* (Vancouver: University of British Columbia, 2012).

Calvino-Fraga, Jesus. *Project 2: Advanced Multimeter Using VHDL.* (Vancouver: University of British Columbia, 2012).

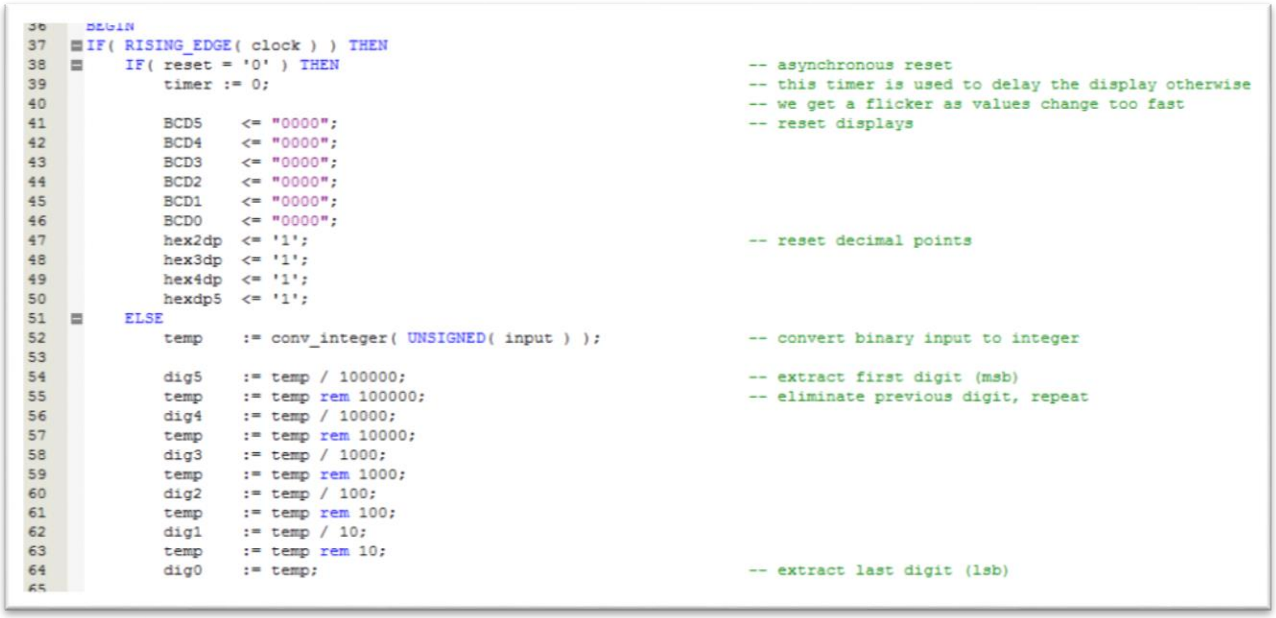
6 Appendices

6.1 Appendix A: Voltmeter Conversion VHDL Code



```vhdl
        ELSIF( state = "00" ) THEN                      -- ACQUIRE STATE
            RC <= '0';                                  -- set RC '0' to trigger conversion
            state <= "01";                              -- transition to timing state

            IF( voltmeter = '1' AND ammeter = '0'       -- VOLTMETER CODE BELOW
              AND ohmmeter = '0' AND betameter = '0') THEN

                mode <= "000";
                unit <= "10000101110010";               -- display 'dc'
                vSwitch <= "01";
                aSwitch <= "00";
                oSwitch <= "00";
                bSwitch <= "00";

                temp    := conv_integer( SIGNED(dataIN) ); -- convert ADC input to integer: -32768 to 32767
                temp    := (temp * incVolt);
                temp    := (temp / 100) - offsetV;      -- will give result in integer without decimal point
                                                        -- decimal point to be assigned by separate algorithm
                IF( temp < 0 ) THEN
                    neg     <= '1';                     -- set negative flag
                    temp    := temp * (-1);             -- get rid of negative sign
                ELSE
                    neg     <= '0';
                END IF;                                 -- VOLTMETER CODE ABOVE

            ELSIF( voltmeter = '0' AND ammeter = '1'    -- AMMETER CODE BELOW
              AND ohmmeter = '0' AND betameter = '0') THEN

                mode <= "001";
                unit <= "11010100001000";               -- display 'nA' (mA)
```

6.2 Appendix B: Binary to BCD VHDL Code



```vhdl
    BEGIN
    IF( RISING_EDGE( clock ) ) THEN
        IF( reset = '0' ) THEN                                  -- asynchronous reset
            timer := 0;                                         -- this timer is used to delay the display otherwise
                                                                -- we get a flicker as values change too fast
            BCD5    <= "0000";                                  -- reset displays
            BCD4    <= "0000";
            BCD3    <= "0000";
            BCD2    <= "0000";
            BCD1    <= "0000";
            BCD0    <= "0000";
            hex2dp  <= '1';                                     -- reset decimal points
            hex3dp  <= '1';
            hex4dp  <= '1';
            hexdp5  <= '1';
        ELSE
            temp    := conv_integer( UNSIGNED( input ) );       -- convert binary input to integer

            dig5    := temp / 100000;                           -- extract first digit (msb)
            temp    := temp rem 100000;                         -- eliminate previous digit, repeat
            dig4    := temp / 10000;
            temp    := temp rem 10000;
            dig3    := temp / 1000;
            temp    := temp rem 1000;
            dig2    := temp / 100;
            temp    := temp rem 100;
            dig1    := temp / 10;
            temp    := temp rem 10;
            dig0    := temp;                                    -- extract last digit (lsb)
```

6.3 Appendix C: Snapshot of the Final Product