LAB 2-4 – NAME DETECTION LAB

DSD – 2131

CLIFF CHAPMAN

OBJECTIVE

The purpose of this lab was an introduction into basic logic system analysis, design and implementation using several different tools and processes for each step. The process of following a logical system from design analysis through to implementation on several different platforms is a long and complex procedure. Even with simple systems such as the name detection circuit each step can take more than the 2 hours allotted for each lab period. Taking each week to walk through each phase in the design process allows for introspection into how these classroom learned ideas apply to the real world.

Each lab had a separate sub-objective related to the final synthesis of a circuit capable of detecting letters in a target name. Each lab will be discussed in detail in the Procedure section.

2. Lab 2 consisted of deriving the final logic equations and synthesis of the schematic.
3. Lab 3 consisted of implementing the schematic in discrete chips on a breadboard.
4. Lab 4 consisted of implementing the equations in VHDL on the DE-2 FPGA board.

During this lab sequence multiple data views were recorded, and signoffs written. The signoffs will be mentioned in the Results section where appropriate.

PROCEDURE

The contents of the design, analysis and implementation process were spread across 3 separate labs with different purposes. Each lab consisted of a pre-lab and a lab set of objectives to be completed each week.

The primary design for the process was a name detection circuit based on ASCII codes. An 8 bit value would be passed to the circuit, with the first 3 bits being known to “010”. The following 5 bits covered the uppercase range of ASCII values for letters. For simplification purposes this lab was limited to uppercase ASCII characters only.

LAB 2 PRE-LAB

Lab 2’s pre-lab was the design portion of this process. K-maps were created to express the requirements of the logic circuit, from which the final logic equation was derived. The k-maps were generated off of a simple truth table. The table can be seen here:

Letter	A	B	C	D	E	FN	LN
A	0	0	0	0	1		1
C	0	0	0	1	1	1	1
F	0	0	1	1	0	1	
H	0	1	0	0	0		1
I	0	1	0	0	1	1	
L	0	1	1	0	0	1	
M	0	1	1	0	1		1
N	0	1	1	1	0		1
P	1	0	0	0	0		1

This truth table allowed for the creation of two k-maps. As the letter “P” is the only letter to require A to be true, this simplified the other letters to be 4-variable k-maps, as A could be assumed to be false. P was left as a separate equation for simplification purposes and required no additional k-map.

The generated k-maps are as follows:

FIRST NAME

!A	!B!C	!BC	BC	B!C
!D!E			1	
!DE				1
DE	1			
D!E		1		

LAST NAME

!A	!B!C	!BC	BC	B!C
!D!E				1
!DE	1		1	
DE	1			
D!E			1	

Both k-maps were unfortunately not capable of much reduction in complexity. The letters are rather spread about either board. It might be possible for the first name to operate with an XOR, but our understanding of k-map XOR reduction is limited. It was concluded that both maps would be best left to their simplistic reduction.

From here the two logic equations could be derived.

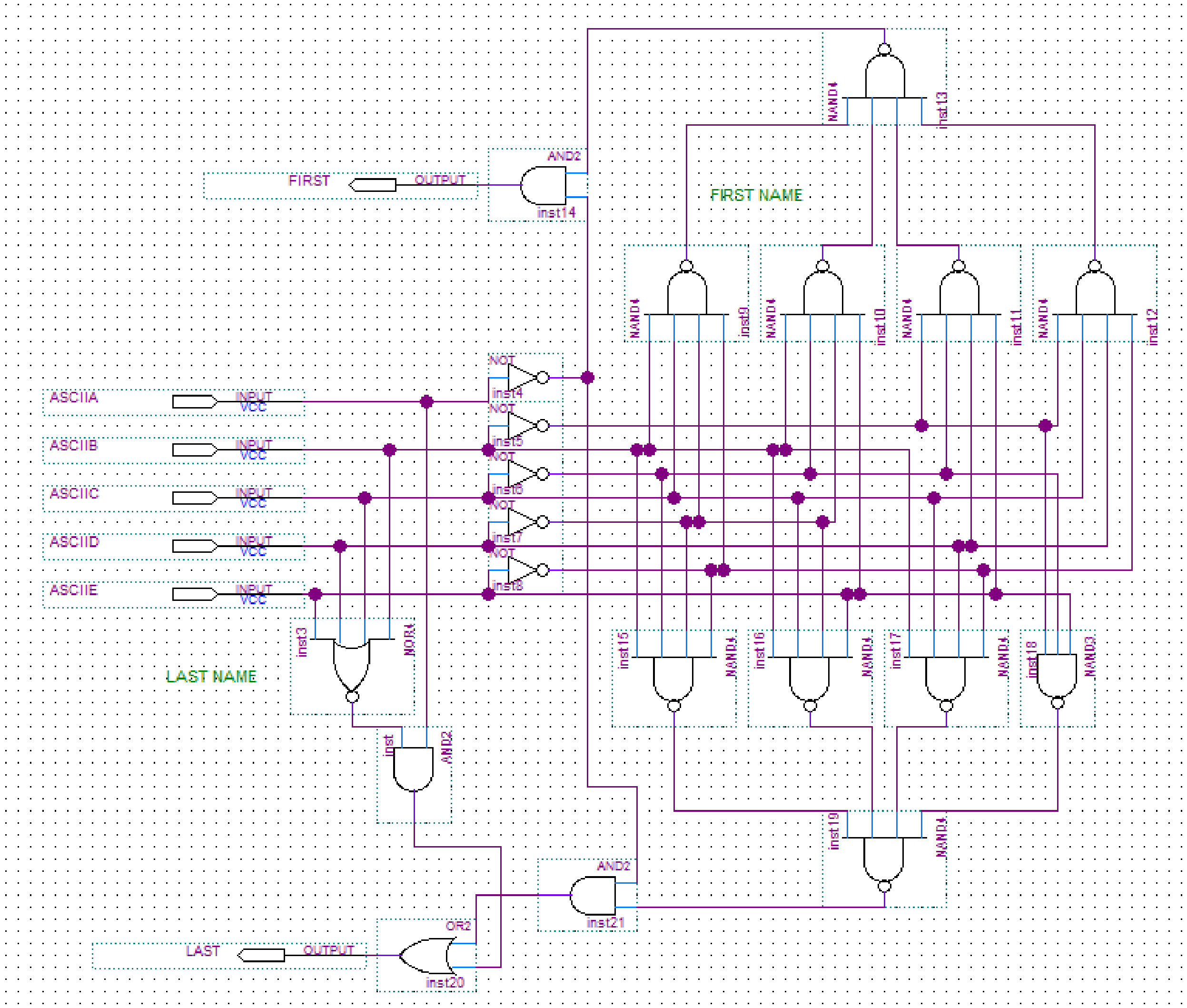
The first name equation:

$$\bar{A}(\bar{B}\bar{C}DE + \bar{B}CDE + \bar{B}\bar{C}DE + \bar{B}CD\bar{E})$$

The last name equation:

$$\bar{A}(B\bar{C}\bar{D}\bar{E} + \bar{B}\bar{C}E + BC\bar{D}E + BCD\bar{E}) + A\bar{B}\bar{C}\bar{D}\bar{E}$$

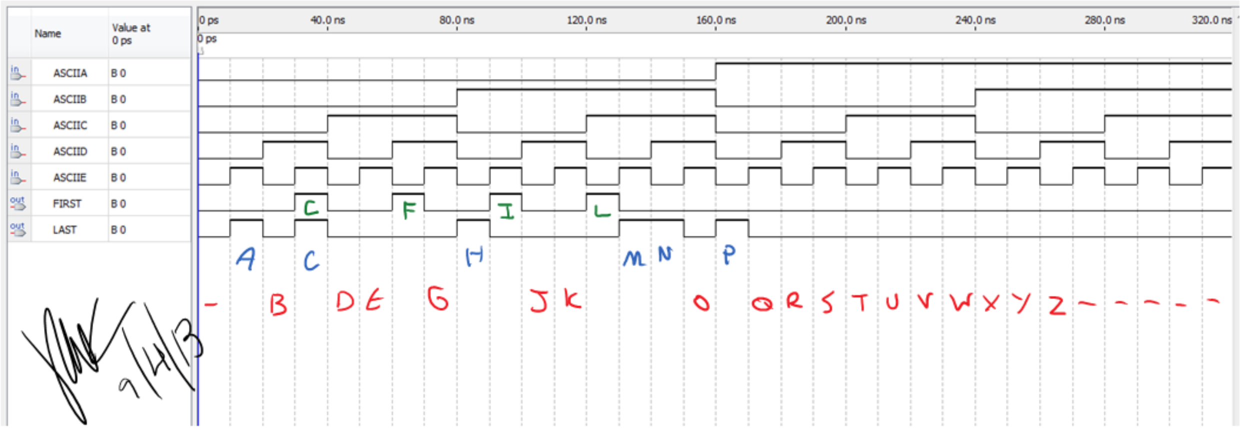
From here we used this set of equations to create a schematic limited by the available chips in the ESD lab. This is the schematic designed based on those limitations.



Note the use of a 4-input NOR gate to complete the equation for the letter "P" located beneath the 5 ASCII inputs. This was an error in reading the list of available chips. As a quad input NOR gate is not available, a replacement was created for the chip implementation in Lab 3. This substitution is elaborated upon in Lab 3's section.

LAB 2

During lab 2 the Quartus schematic was used for a timing simulation analysis. This simulation had two modes, a functional assessment and a timing simulation. The functional assessment generated a waveform output on the output pins to verify proper function of the simulation. The functional output can be seen here. In addition is the signoff for Lab 2's completion.



Unfortunately due to lack of foresight, the timing simulation is not available. The timing analysis showed minor aberrations in the output surrounding various output sequences.

LAB 3 PRE-LAB

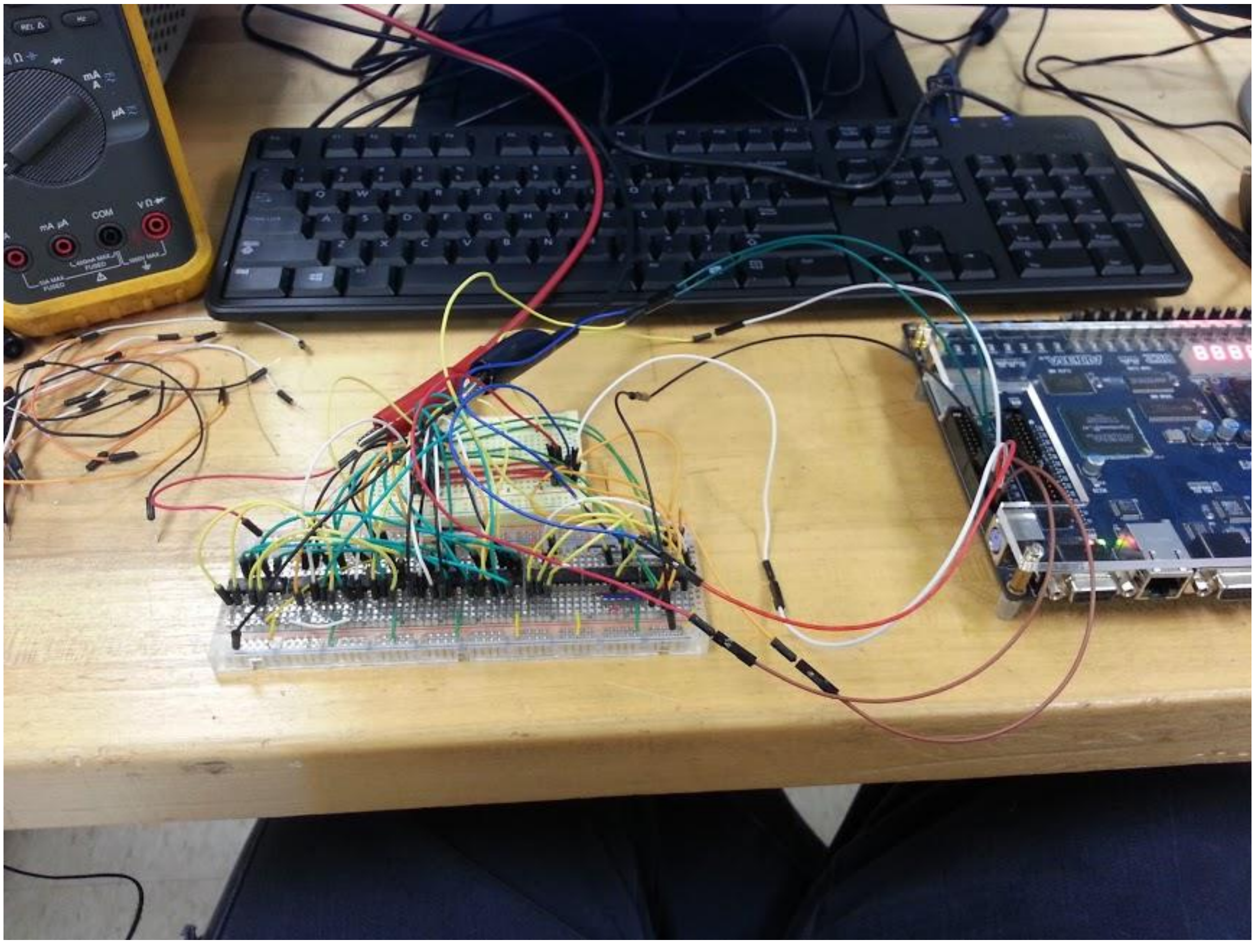
The pre-lab for Lab 3 was primarily a planning task for implementation of the name detector circuit in discrete logic components. The schematic from Lab 2 was labeled and a Bill of Materials generated for what to use during Lab 3.

It was at this point that an issue with the schematic was noted, namely the use of a quad-input NOR gate. Such a gate is not available in the ESD lab, as such a replacement circuit was designed. The replacement circuit replaced the quad-input NOR gate with three OR gates and one NOT gate. Fortunately this change did not actually impact the BoM, part counts remained identical.

LAB 3

Lab 3 saw the construction of the circuit designed in the pre-lab. My implementation of only 8 chips allowed for the construction to fit on a single breadboard, simplifying the overall construction. The final result was connected to the DE-2 running a simple switch to GPIO forwarding program. The implementation was then tested using the switches available on the DE-2. The functionality was confirmed for the professor.

Discrete implementation:



Signoff for discrete implementation:

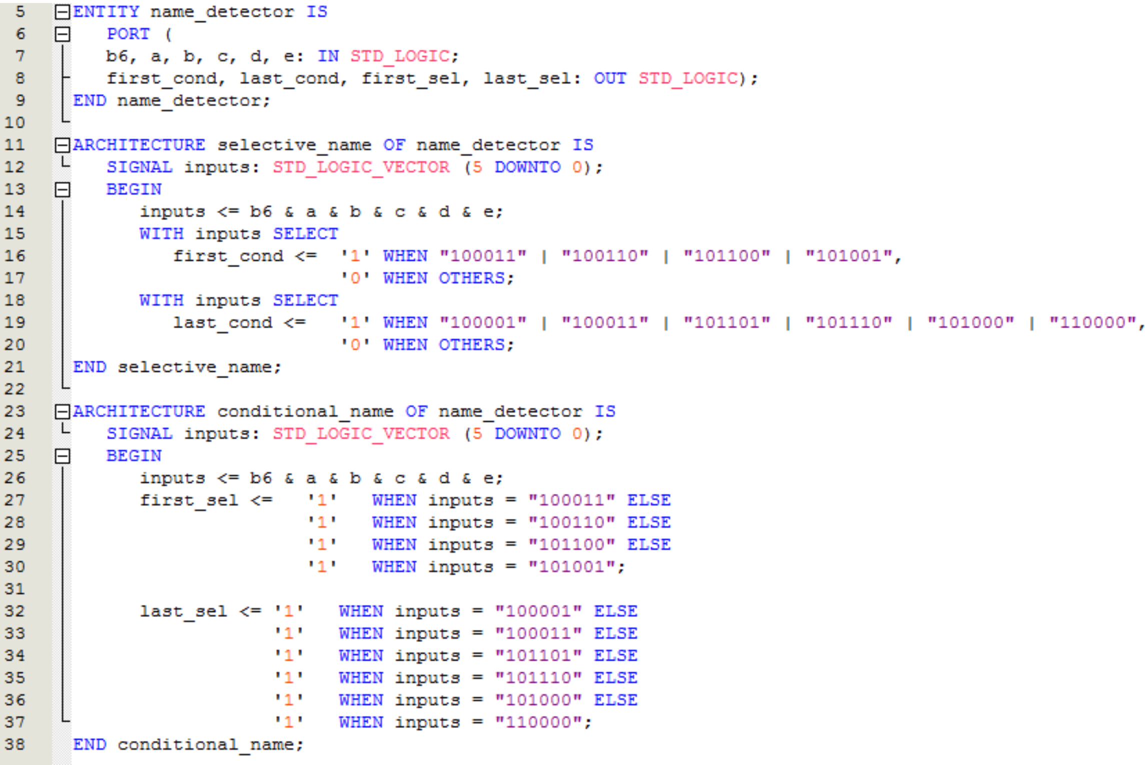
first and last

9/11/13

LAB 4 PRE-LAB

Lab 4 was centered on creating a VHDL implementation of the name detector circuit. Two architectures were specified; a conditional signal assignment and a selected signal assignment architecture structure. These instructions were originally taken at face value, and two separate architecture definitions were created. This failed to work, and after some guidance a complete VHDL entity with both signal assignment methods were present in a single architecture description.

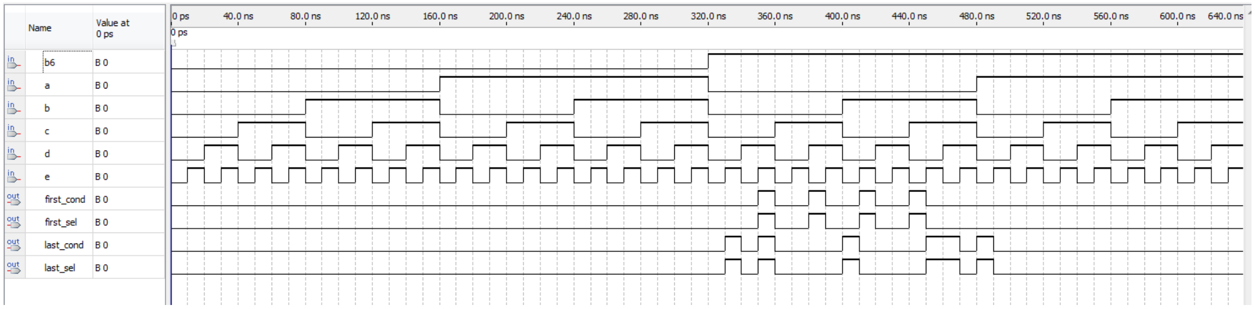
VHDL implementation before integration into a single architecture:



```
ENTITY name_detector IS
   PORT (
   b6, a, b, c, d, e: IN STD_LOGIC;
   first_cond, last_cond, first_sel, last_sel: OUT STD_LOGIC);
END name_detector;

ARCHITECTURE selective_name OF name_detector IS
   SIGNAL inputs: STD_LOGIC_VECTOR (5 DOWNTO 0);
   BEGIN
      inputs <= b6 & a & b & c & d & e;
      WITH inputs SELECT
         first_cond <=  '1' WHEN "100011" | "100110" | "101100" | "101001",
                        '0' WHEN OTHERS;
      WITH inputs SELECT
         last_cond <=   '1' WHEN "100001" | "100011" | "101101" | "101110" | "101000" | "110000",
                        '0' WHEN OTHERS;
END selective_name;

ARCHITECTURE conditional_name OF name_detector IS
   SIGNAL inputs: STD_LOGIC_VECTOR (5 DOWNTO 0);
   BEGIN
      inputs <= b6 & a & b & c & d & e;
      first_sel <=   '1'   WHEN inputs = "100011" ELSE
                     '1'   WHEN inputs = "100110" ELSE
                     '1'   WHEN inputs = "101100" ELSE
                     '1'   WHEN inputs = "101001";

      last_sel <= '1'   WHEN inputs = "100001" ELSE
                  '1'   WHEN inputs = "100011" ELSE
                  '1'   WHEN inputs = "101101" ELSE
                  '1'   WHEN inputs = "101110" ELSE
                  '1'   WHEN inputs = "101000" ELSE
                  '1'   WHEN inputs = "110000";
END conditional_name;
```

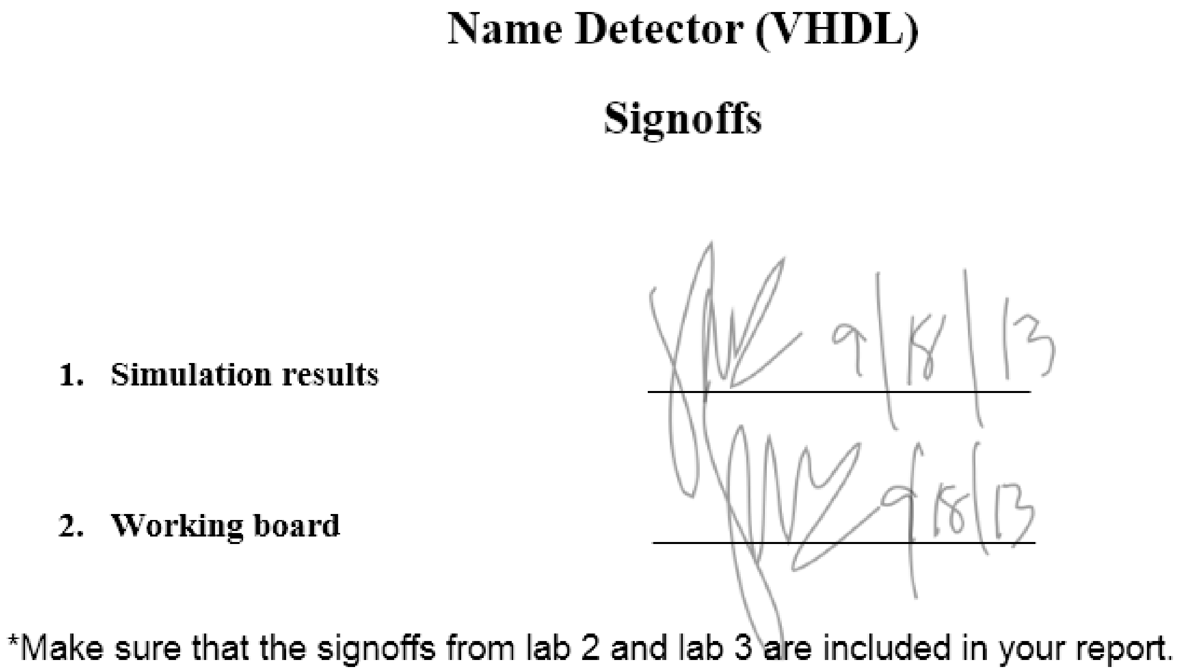
LAB 4

During Lab 4 the simulator program was used again to verify the VHDL implementation. The final waveform can be seen here. Note that first name and last name outputs are grouped for ease of verification.



Note that this waveform includes the ASCII b6 bit. This was not present for the discrete implementation for simplification purposes. It was not difficult to include the appropriate bit with the VHDL design, as such it was included here. Unfortunately due to this inclusion in the VHDL design verses the discrete implementation the equations generated by Quartus do not match up. During both Lab 3 and Lab 4, equations were generated by the compilation of both designs to demonstrate similarity to the original K-map. Due to the modification of pin assignments and the optimization present in Quartus, these equation similarities are not present. As such the equations section has been omitted.

Once the waveform was complete the program was ported to the DE-2 board. The operation of the board was confirmed. These are the signoffs for Lab 4:



Name Detector (VHDL)

Signoffs

1. **Simulation results** 9/18/13
2. **Working board** 9/18/13

*Make sure that the signoffs from lab 2 and lab 3 are included in your report.

CONCLUSIONS

All four implementations of the circuit resulted in the same output. There were very few resultant differences in the functioning of the system. Though all four implementations were separate processes for implementing the circuit, the logic circuit remains the same through each system. Put simply, they are different ways of going about the same activity.

The VHDL implementation was, by far, easiest. Generating the schematic took close to an hour and a half. Constructing and debugging the discrete circuit took three hours, an extra hour of which came from before lab started. Many students took several consecutive labs to finalize their discrete circuits. Though more accurate and cheaper than the FPGA-based implementations the time requirement was quite involved, even though this was a relatively simple circuit.

The VHDL implementation took half an hour, and debugging another half an hour at most. At that point it was on the board and ready to be tested. Modifications were fast and the compiler was able to sanity-check my code quickly. Constructing the circuit “virtually” was significantly easier overall for rapid prototyping. Additionally, this did not require the use of any more resources than the DE-2 board, which can rapidly be re-tasked to other uses. Compare this to the breadboard, which utilized a large swath of my jumper wires, two breadboards and eight discrete chips. For fast proof-of-concepts and prototypes the DE-2 with VHDL was simply the best approach.

Even within VHDL there are multiple alternatives for creating an implemented circuit, all with appropriate tradeoffs. The two signal assignment systems we developed showed equivalent performance, showing that VHDL can be very light weight to write and maintain.