Christopher Bero
10-4-2015
CPE 324

Lab Report #8

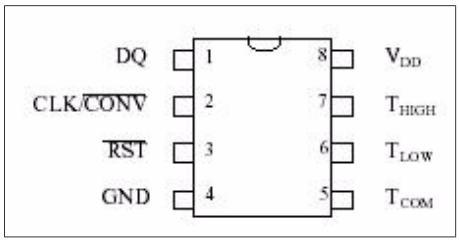
Temperature Sensor

Introduction

In this lab we operate with some level of real-world scenario by designing a single module to integrate with an existing set of IP written in a variety of formats. These modules will still operate and synthesize in the Quartus IDE, but will require some additional posturing to continue development as a normal Verilog project. We also look at some higher level peripheral communication from the FPGA package than previous labs where the system designed is contained on the DE2-115. Another aspect of working with the VHDL and separate temperature package is to simulate working in a team/workgroup environment, as laid out by the lab manual. By creating, simulating, integrating, synthesizing, and testing this lab with the supplied IP we will effectively operate with virtual coworkers.

Theory and Analysis

Much of the analytic basis for this lab has been squared away via the initial set of modules in Verilog and VHDL we have to work with. Most of the communication theory boils down to the provided state machine which dictates the parsing of serial data between the DE2-115 and DS-1620.



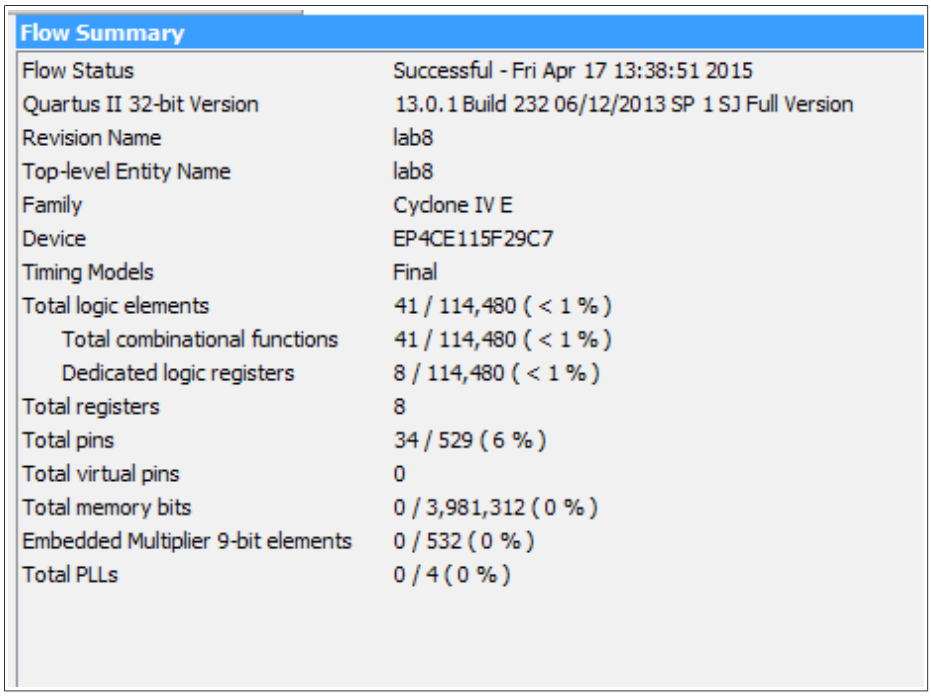
Rather than use the T_{high}, T_{low}, or T_{com} pins on the DS1620, we'll use the device in “three wire” mode where DQ handles input for commands and output for serial temperature data.

Procedures

Procedure was followed as closely as possible from the lab manual, “Laboratory Assignment #8, Temperature Sensor”. There was some confusion on integrating VHDL components into the project, but on the third attempt the instructions listed at the end of the manual worked to allow the project to synthesize.

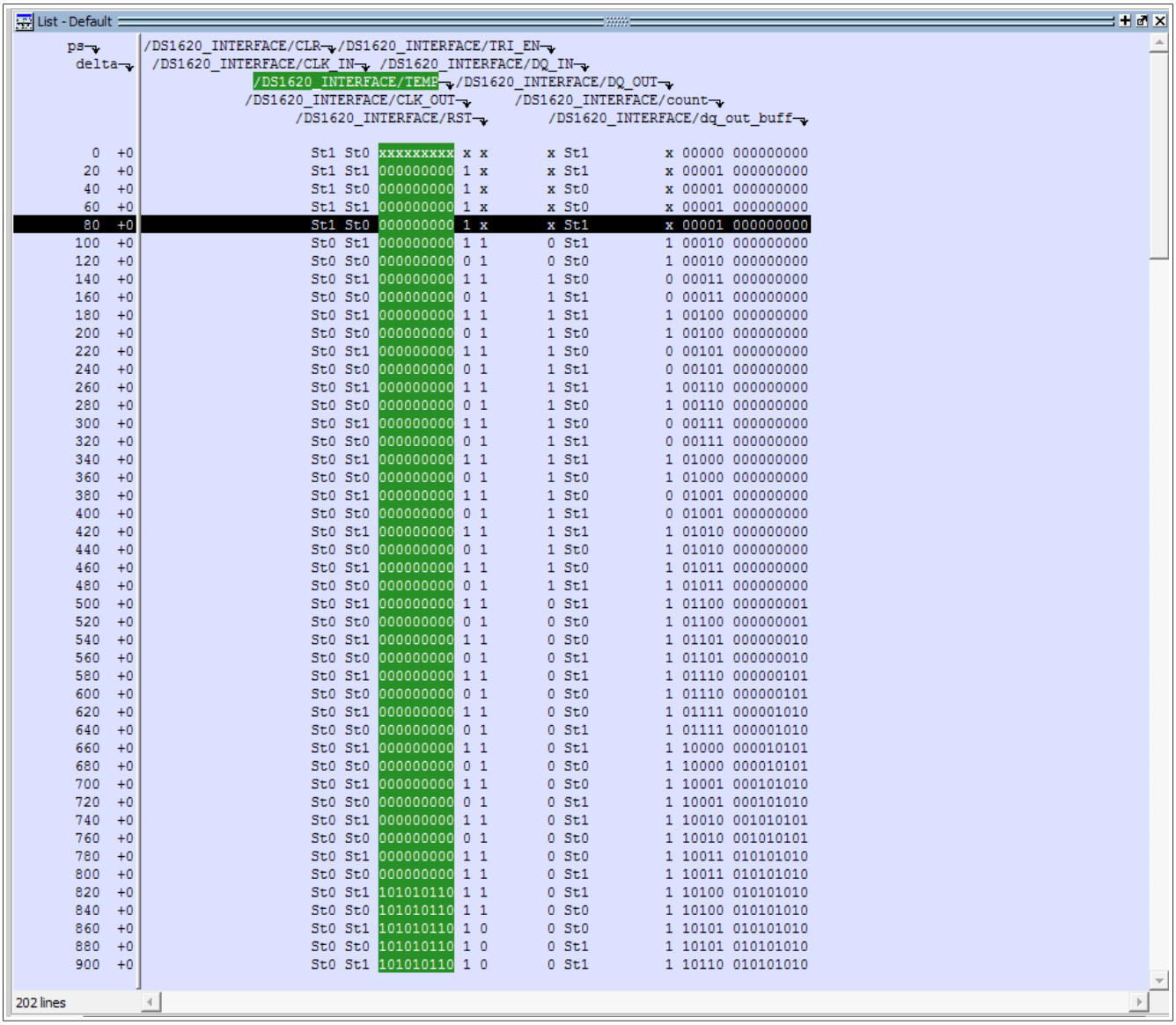
Results

The downloaded project operated in a reasonably precise manner, returning to the same value after being heated up by convection. The measured temperature was not accurate, however, as the display read roughly 1–2% lower than the real value. Simulation of the program appeared to return correct values, but a intricacy of the state machine may still be at fault for the discrepancy.



Flow Summary	
Flow Status	Successful - Fri Apr 17 13:38:51 2015
Quartus II 32-bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Full Version
Revision Name	lab8
Top-level Entity Name	lab8
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	41 / 114,480 (< 1 %)
Total combinational functions	41 / 114,480 (< 1 %)
Dedicated logic registers	8 / 114,480 (< 1 %)
Total registers	8
Total pins	34 / 529 (6 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)

```
#
# Stimulus
# [0] is LSB
# SUB_ADD: 0=add (A+B), 1=subtract (A-B)
# Library: cycloneive_ver
#
force CLR 1 0, 0 100
force DQ_IN 1 0, 0 40 -repeat 80
force CLK_IN 0 0, 1 20 -repeat 40
run 4000
```



Simulation output shows the system correctly processing DQ input and providing parsed temperature output. Other factors in the physical implementation may include the breadboard connections, true voltage supplied, and DS1620 quality/condition.

Conclusion and Discussion

Implementing the DS1620 has been one of the more challenging labs for this class thus far, as working with other modules and VHDL required more time to investigate the language's syntax and overall project structure. Its the belief of this lab member that a more appropriate or well designed Verilog module can be made for the lab's DS1620_INTERFACE, but time and resource limitations had the item left at its current state of completeness. Operation of the DS1620 was taken from the Dallas Semiconductor datasheet, “DS1620 Digital Thermometer and Thermostat”. The IC operates on a 3-wire serial interface in which the temperature, when requested, is converted into a 9-bit value within one second and sent as a reply over the DQ pin. This is a new method of serial communication than discussed in previous classes, though it does appear to model SPI in several views.