James Johnston
CS 6133
Module 4

Consider the following logic delay:

L.E. / Temp.	Gate	Register
	Propagation Delay	
85 °C	.9ns	.9ns
50 °C	.7ns	.9ns
10 °C	.6ns	.7ns

1 For each temperature profile, calculate the longest and smallest combinatorial path delay in the Simple CPU described in this lecture.

Longest path : Instruction Memory – >Register File – >Mux – >ALU – >Data Memory – >Mux

Shortest path : Adder – >Adder – >Mux

Instruction Memory has 2 gates and 1 register

Register File has 4 gates and 1 register

MUX has 3gates

ALU has 3 gates

Data Memory has 1 register

Adder has 3 gates

At 85 degrees Celsius the longest path delay is 15 gates X .9ns + 3 registers X .9ns = 16.2ns
The shortest path delay at 85 degrees Celsius is 9 gates X .9ns = 8.1ns

At 50 degrees Celsius the longest path delay is 15 gates X .7ns + 3 registers X .9 ns = 13.2ns

The shortest path delay for 50 degrees Celsius is 9 X .7ns = 6.3ns

At 10 degrees Celsius the longest path delay is 15 gates X .6 + 3 registers X .7ns = 11.1ns

The shorter path delay for 10 degrees Celsius is 9 X .6ns = 5.4ns

SignalTap II and Memory Editor: The simple CPU will uses Quartus II SignalTap II and Memory Editor software features. SignalTap II is an in-logic analyser used to debug designs implemented in Altera FPGAs. In this activity, you will have an opportunity to get introduce to these features and get accustomed to their debugging capabilities. For this activity you would be using mips basic.qar project archive.
You will using three Quartus tools in this activity:

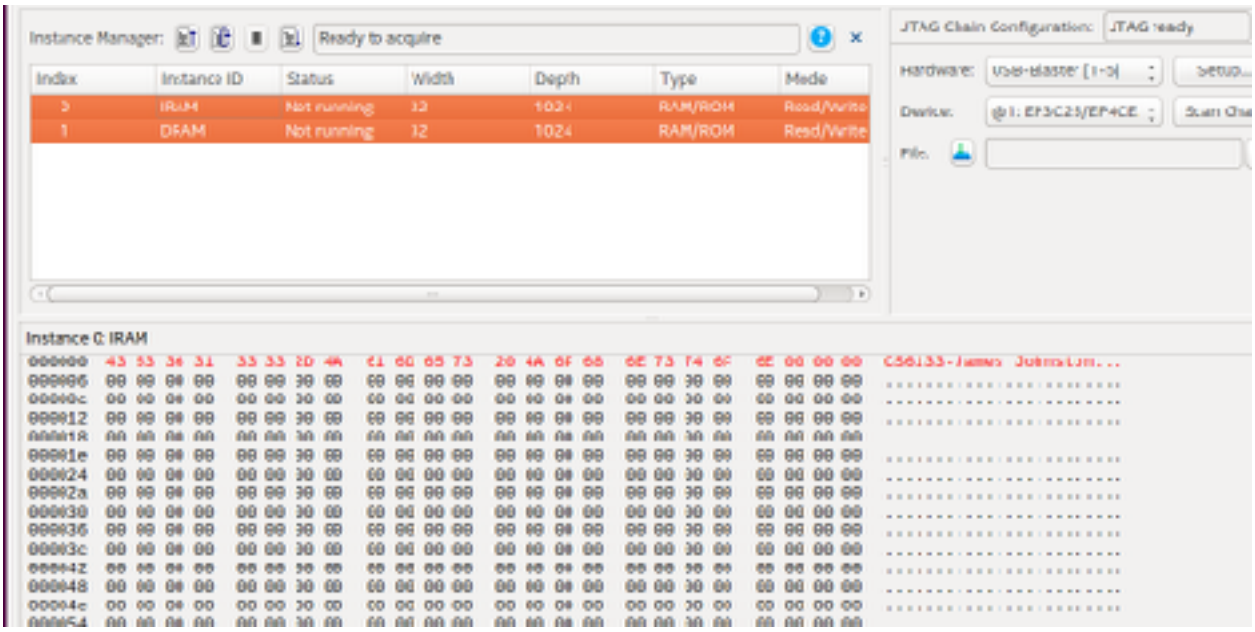
1 Custom GUI tcl script (gui front 2.tcl) and “quartus stp”
2 Quartus II - In-System Memory Content Editor
3 Quartus II - Signal Tap II

1 Memory Initialization File MIF: What is a .mif file and how is it used?
Describe .mif file format that is used by Altera’s Quartus II software.

A MIF file is an ASCII text file, with the extension .mif that specifies the initial content of a memory block, CAM, RAM, or ROM. It is the initial values for each address. MIF files were used on project compilation and simulation. We can create a Memory Initialization File in the Memory Editor, the In-System Memory Content Editor, or the Quartus II Text Editor.
A MIF file is an input file for memory initialization in the Compiler and Simulator. You can also use a Hexadecimal, Intel-Format, File, .hex, to provide memory initialization data.

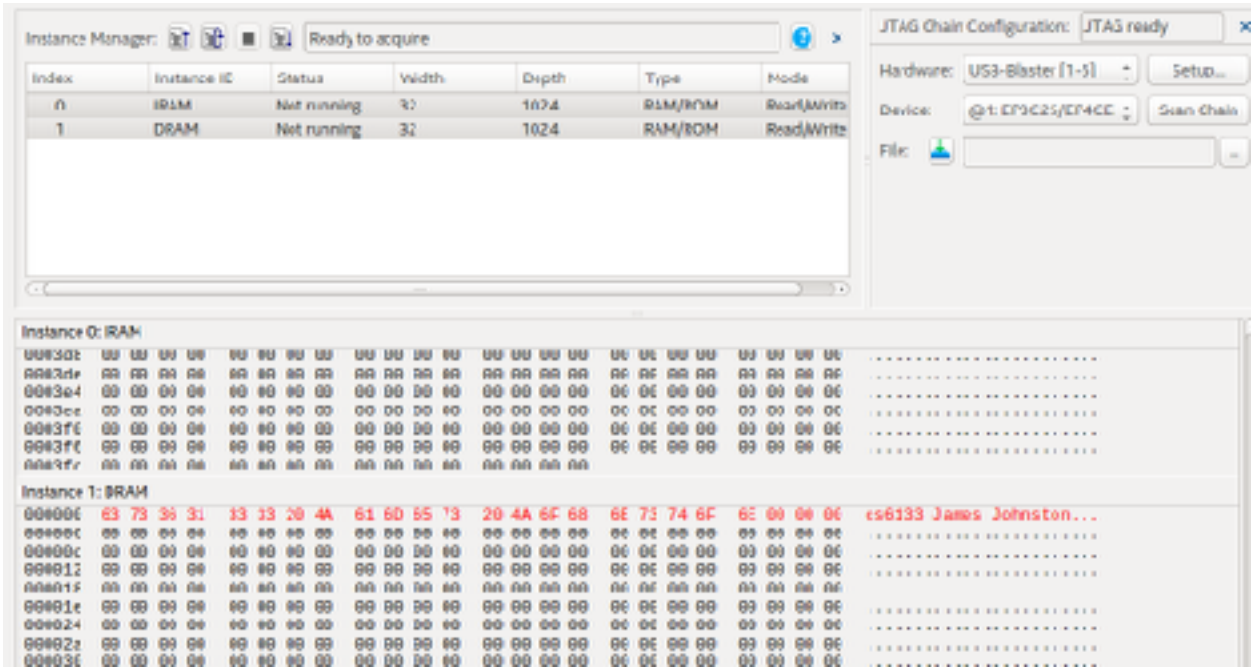
Using “In-System Memory Content Editor” write the following into memories and submit a screen capture for each.

1 Write “CS6133-your name” into IRAM. See video for demonstration



2 Write “CS6133-your last name” into DRAM. See video for demonstration

2 SignalTap II and gui front 2.tcl:
Launch SignalTap II by typing “quartus stpw mips basic.stp &” in the command line from the project directory. (See demonstration)
Launch gui front 2.tcl by typing “quartus stp -t gui front 2.tcl” in the command line from the project directory
Read SignalTap II documentation 1 and submit procedures to perform the following tasks



When creating a Memory Initialization File in the Quartus II Text Editor, I should begin with the DEPTH, WIDTH, ADDRESS_RADIX and DATA_RADIX keywords. I can use Tab "\t" and Space " " characters as separators, and insert a single comment with double dash "--" characters. Address : data pairs represent data contained inside certain memory addresses and I should place them between the CONTENT BEGIN and END keywords.

1 Testing ALU's functionalities using gui front 2.tcl and SignalTap II

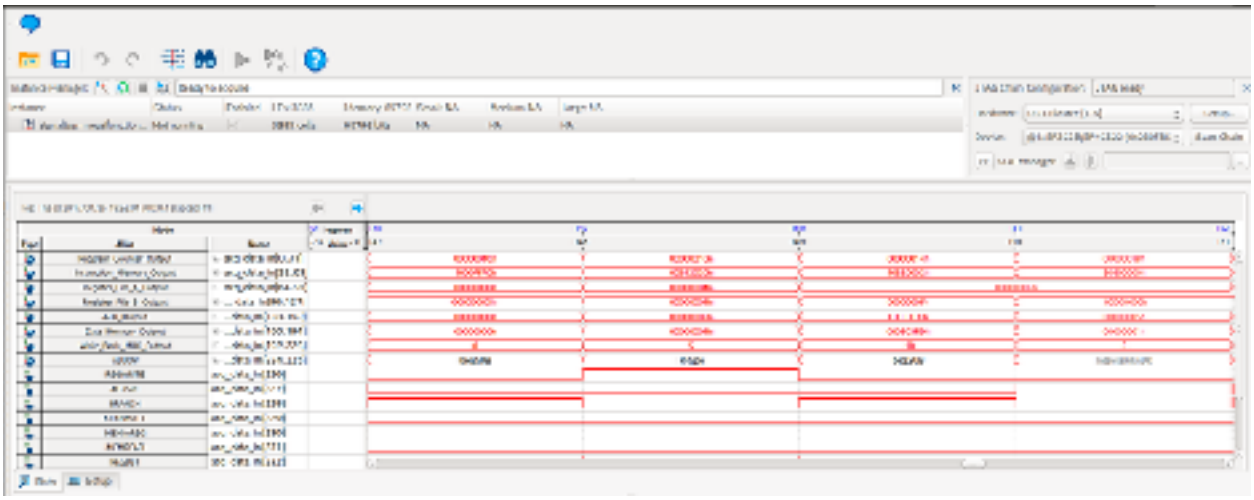
To test gui front 2.tcl I input a count and reset the then jtag refresh. I collect data when the ALU signal creates an LED to fire.

2 Testing Program Counter's functionalities using gui front 2.tcl and SignalTap II

I can change the count in the user input and see the out in my signal tap

3 Reading the contents on instruction and data memory using SignalTap II.

4 Reading the contents of register file using SignalTap II



3 Answer the following questions:

1 Explain the purpose of strb module in mips basic.

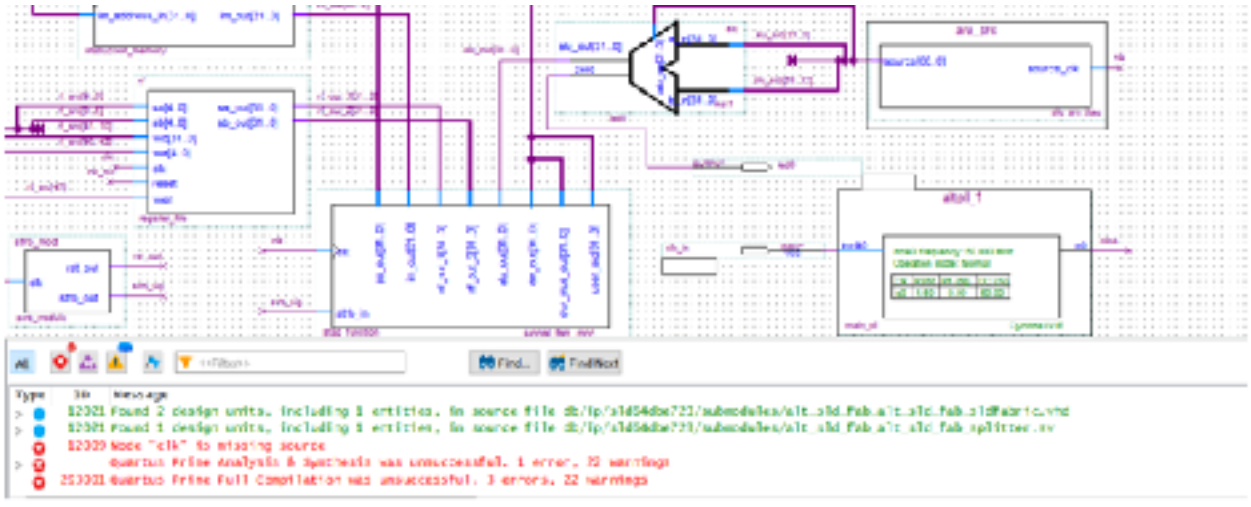
The strb moduls is used to reveal clock signals in the signal tap module.

2 When and why does “led0” connected to ALU light up?

The led0 lights up when there is a subtraction at the ALU and the result is 0. Due to the fact that the led0 is connected by a nor gate to the adder, which is a component in the ALU.

3 The clock wire in the schematic seems to be disconnected, how is this working?

In Quartus it auto connects two different parts that have the same name. So our clock signal will be auto-connect in the schematic of MIPS. For example, I can change the clk signal`s name into “clk A”and the other clk remains the same. After I compile, we can see from the report that all modules are missing clk signal. So we can concluded, “clk A” isn`t connect to the right place because of it name.



4 What is purpose of “rst out” signal?

The ‘rst out’ signal is used to clean away data stored in the registers.

5 What is the clock frequency used in mips basic?

The clock frequency used in MIPS basic is 50MHz.

6 Is mips basic ALU same as the ALU in this lecture? If not, why? support your answer with details.

The MIPS basic ALU is different from the ALU in the lecture.The ALU in lecture only has an add function and it is connect to a XNOT gate to perform subtraction. When the ALU is doing subtraction, the input of XNOT gate will be 1. The input of the adder will be in reverse and add 1(That is in 2`s complement). For example, if I want to perform 4-5. The two inputs to ALU will be “4”and “5”, and the input of XNOT gate will be 1. Changing “5”into “-5” in 2`s complement in this case. But the adder in MIPS basic ALU has add and subtract function.

Consider the Simple CPU blocks, presented in this lecture, and Dr. Neumann’s digital computer.

1 Is it possible to use a RAM/ROM for control unit? Justify your answer with details.

RAM is volatile memory, which means the data store inside the control unit will disappear after we power down. This can cause a very serious error in the data path. Therefore RAM cannot be used as a control unit. ROM, however, can be used as a control unit because it is nonvolatile memory and is very stable.

2 Compare the control units (both the main control unit and ALU control unit) with Dr. Neumann’s CC and answer the following questions:

1 List their similarities and differences

Similarities	**Differences**
Both control units in Dr. Neumann’s CC and the lecture can perform receiving instructions and control other elements (MUX, data memory and ALU, etc.) so that the whole CPU can run efficiently.	The control unit in Dr. Neumann’s CC cannot control instruction memory, but the control unit in the lecture can control instruction memory.

2 Does Simple CPU's control units (Main and ALU control unit) confirm to CC's requirements? Justify your answer with details.

The simple CPU's control units, the main and ALU control unit, does confirm the CC's requirements. This is because its control units can control both the instruction memory and the other modules.