Lingbo Meng

Mat Maternowsky

EE 4305

Computer Architecture

Professor: Dr. Taek Kwon

Lab #8: MIPS-32 Implementation: Decode Modules

1. Objective of this lab

Designing a decode module of the MIPS-32 processor that is responsible for decoding of fetched instructions from fetch module, which is designed in last lab. Showing the register-rs, register-rt, register-rd, jump address and immediate on tera terminal and showing the index on seven segment display.

1. What this decode module implements and do

Decoding rs, rt, rd, immediate respectively, extending sign bits for immediate, determining write whether memory data or Alu result by MemtoReg signal, deciding whether rs ot rd will go into the write register by RegDst signal and giving jump address.

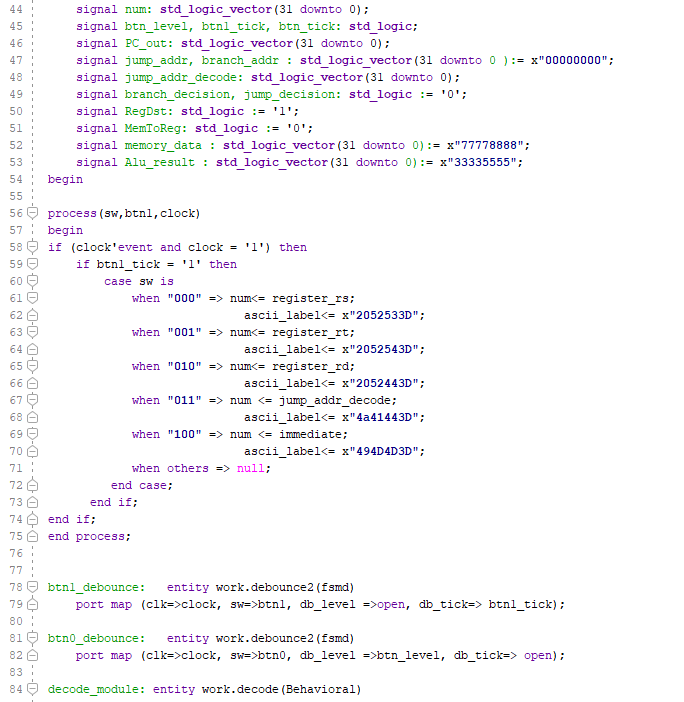
1. What did we learn from this lab

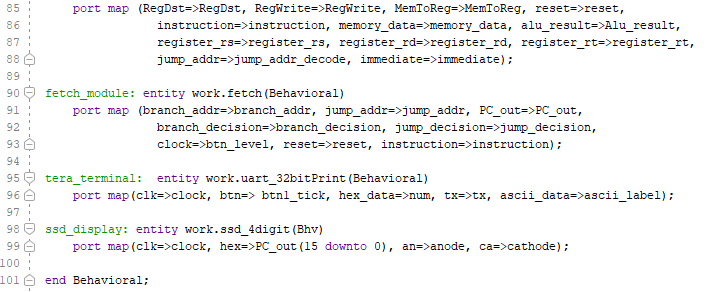
In this lab we learned how to implement a decode module into our Mips-32 processor. We learned how to read the value of the [25:0] portion of our instruction and categorize it into rs, rt, rd, jump address, and immediate and display each category on the uart and the index of the instruction on the ssd. We learned how to sign extension a signal. We learned how to implement a RegWrite signal.

1. VHDL source code

Top module:







Decode module:

