Billy Koech and Nicolas Weinger

CS 141: Computing Hardware

Lab 4 Part 1 : MIPS Datapath (R-Typs)

Testing methodology

The MIPS data path is implemented to support just R types (except for jr which will be implemented later along with J types). The following blocks are implemented as modules and instantiated in the MIPS core module:

1. Control
2. ALU control
3. ALU
4. PC Register
5. Register File
6. IorD mux
7. IR register
8. MDR register
9. RegDst mux
10. MemtoReg mux
11. A register
12. B register
13. ALUSrcA 3 to 1 mux
14. ALUSrcB 5 to 1 mux
15. PCSource 3 to 1 mux
16. ALUout register

The Control FSM module is tested by creating a testbench that resets the FSM and runs it for about 100 clock cycles. The waveform of the enable pins (*IRWrite, MemWrite RegWrite, PCWriteCond*) and select pins (*IorD, ALUSrcA, ALUSrcB, ALUOp, PCSource, RegDst,*

*MemtoReg*) are then observed in each state and checked for consistency with the MIPS R-Type FSM diagram.

In order to test for functionality of the entire data path we convert assembly code to machine code and load it into the instruction memory. Below is an example of how we tested nor, srl, slt, add and sub:

nor $t0, $t0, $t0

srl $t1, $t0, 6

slt $t2, $t0, $t1

add $t3, $t0, $0

sub $t4, $t0, $t3

We then viewed the waveform of the 32 registers in the Register File and check whether they hold store the following values for this specific code:

$t0: 11111111111111111111111111111111 // creating data to manipulate later

$t1: 00000011111111111111111111111111 // shifting right

$t2: 00000000000000000000000000000001

$t3: 11111111111111111111111111111111

$t4: 00000000000000000000000000000000