

Computer Architecture II  
Computer Engineering Department  
An-Najah National University  
Dr. Suleiman Abu Kharmeh  
Assignment #3 (final): Completing the MIPS Pipeline

Student names:

Rawan Hossam Hassoun : 11819677

Amna Abd Alraheem Khdaire: 11819646

---

In this assignment we aim to support a new instructions that not supported in ModelSim  
We choose shift instructions according our university IDs.

♣ **We implement two new instructions:**

1-  $7 \bmod 8 = 7$

The instruction is: Shift right logical (variable)

SRLV \$1, \$2, \$3

2-  $6 \bmod 8 = 6$

The instruction is: Shift left logical (variable)

SLLV \$1, \$2, \$3

♣ **In Mars:**

```
xori $s1, $s0, 0x00005555  
xori $s2, $s0, 0x0000ffaa  
sllv $s3, $s1, $s2  
srlv $s4, $s1, $s2
```

Figure1: The two new instructions we want to support in ModelSim, we find the binary output of it in MARs.

- ♣ The result value of the register:

\$s0	16	0x00000000
\$s1	17	0x00005555
\$s2	18	0x0000ffaa
\$s3	19	0x01555400
\$s4	20	0x00000015

Figure2: The registers result from MARs.

- ♣ Find the binary value of the instruction from MARs that we will use in ModelSim:

```
00111010000100010101010101010101
00111010000100101111111101010101
00000010010100011001100000000100
00000010010100011010000000000110
```

Figure3: The binary value of instructions from MARs.

- ♣ We update the alu.v file as needed in our instructions:

```
always @(*) begin
  casex(ALUControl)
    3'b100: Output= BussB << BussA[4:0];
    3'b101: Output= BussB >> BussA[4:0];
    default:Output=out_old;
  endcase
end
endmodule
```

Figure4: apply the new instructions in ModelSim file.

♣ We update ALUControl\_Block.v:

```
module ALUControl_Block( ALUControl, ALUOp, Function);
output [2:0] ALUControl;
reg [2:0] ALUControl;
input [1:0] ALUOp;
input [5:0] Function;
wire [7:0] ALUControlIn;
assign ALUControlIn = {ALUOp,Function};
always @(ALUControlIn)
case (ALUControlIn)
8'b11xxxxxx: ALUControl=3'b001;
8'b00xxxxxx: ALUControl=3'b000;
8'b01xxxxxx: ALUControl=3'b010;
8'b10100000: ALUControl=3'b000;
8'b10100010: ALUControl=3'b010;
8'b10101010: ALUControl=3'b011;
8'b10000100: ALUControl=3'b100;
8'b10000110: ALUControl=3'b101;
default: ALUControl=3'b000;
endcase
endmodule
```

Figure5: define a new ALU control inputs for the new instructions by update the ALUControl\_Block.v.

♣ The wave form of the output signal in ModelSim:

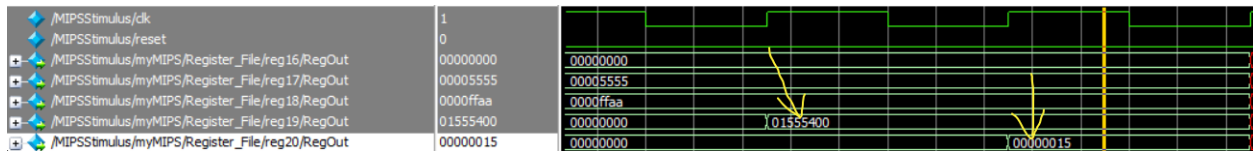


Figure6: The result values of the registers of the new instructions in ModelSim.

As we notice in the ModelSim result(Figure 6) , its the same result in MARs result(Figure 2), so we achieve our goal to support the new instructions(srlv, slv) in ModelSim.