LAB REPORT
CSE4010 Computer Architecture
Instructor: Lawrence Orijuela

Name: Daniel Delgado Acosta SCORE: _ __/30 Student ID: 006896598 DUE: 5/11/23 LAB: Lab6

Report

The revised MIPS datapath is essentially a processor. It shows how instructions are executed.

Source Code

mux.v

```
//Multiplexer module

module mux (

    output wire [31:0] y,   // Output wire of Mux

    input wire [31:0] a,    // Input wire a of Mux

                      b,    // Input wire b of Mux

    input wire sel          // Select Input

    );



    assign y = sel ? a : b; // Mux



endmodule
```

incr.v

```
// Incrementer module

module incrementer (

    input wire [31:0] pcin,     // Input of increment

    output wire [31:0] pcout    // Output of increment

    );

    assign pcout = pcin + 1;    // Increment pc by 1

endmodule
```

Mem.v

```
// Instruction memory module
module memory (
    output reg [31:0] data,     // Output
    input wire [31:0] addr      // Input
    );

    //Register declarations
    reg [31:0] MEM[0:127];      // 128 words of 32 bit memory

    //Inititalize registers
    initial begin
    //Memory cells
    MEM[0] <= 'hA00000AA;
    MEM[1] <= 'h10000011;
    MEM[2] <= 'h20000022;
    MEM[3] <= 'h30000033;
    MEM[4] <= 'h40000044;
    MEM[5] <= 'h50000055;
    MEM[6] <= 'h60000066;
    MEM[7] <= 'h70000077;
    MEM[8] <= 'h80000088;
    MEM[9] <= 'h90000099;
    end
    // I type intrsuction
    always @ (addr) data <= MEM[addr];
endmodule
```

pc_mod.v

```
//Program counter module

module pc_mod (

    output reg [31:0] PC,   // Output

    input wire [31:0] npc   // Input

    );



    //initializes counter

    initial begin

        PC <= 0;

    end

    //points to next instruction

    always @ ( npc) begin

        #1 PC <= npc;

    end



endmodule
```

if_id.v

```
// The IF stage of the pipeline module

module if_id (

    output reg [31:0] instrout,  // Inputs of increment

                      npcout,

    input wire [31:0]    instr,     // Outputs of increment

                         npc

    );



    //points variables

    initial begin

        instrout <= 0;

        npcout   <= 0;

    end

    //points variables to initialized variables

    always @* begin

        #1 instrout <= instr;

             npcout <= npc;

    end

endmodule
```

Ifetch.v

```
// Include .v files

`include "mux.v"

`include "mem.v"

`include "incr.v"

`include "if_id.v"

`include "pc_mod.v"


// The ifetch module of theb IF stage of the pipeline

module I_FETCH (

    output wire [31:0] IF_ID_instr,     // Output

    output wire [31:0] IF_ID_npc,       // Output

    input wire         EX_MEM_PCSrc,    // Input

    input wire [31:0]  EX_MEM_NPC      // Input

    );


    //signals

    wire [31:0] PC;

    wire [31:0] dataout;

    wire [31:0] npc,npc_mux;


    //instantations

mux mux1 (.y(npc_mux),
```

```
        .a(EX_MEM_NPC),
        .b(npc),
        .sel (EX_MEM_PCSrc));

pc_mod pc_mod1 (.PC(PC),
                .npc(npc_mux));

memory memory1 (.data(dataout),
                .addr(PC));

if_id if_id1   (.instrout(IF_ID_instr),
                .npcout(IF_ID_npc),
                .instr(dataout),
                .npc(npc));
// displays output variables
initial begin
    $display("Time\t PC\t npc\t dataout of MEM\t IF_ID_instr\t IF_ID_npc");
    $monitor("%0d\t %0d\t %0d\t %h\t %h\t %d0", $time, PC, npc, dataout, IF_ID_instr,
IF_ID_npc);
    #20 $finish;
    end

endmodule
```

pipeline.v

```
//include .v file

`include "ifetch.v"



//module that simulates the function of the IF stage of the pipeline

module pipeline ();



    //wires

    wire[31:0] IF_ID_instr, IF_ID_npc;

    reg EX_MEM_PCSrc;

    reg [31:0] EX_MEM_NPC;

        //fetches variables

        I_FETCH I_FETCH1( .IF_ID_instr(IF_ID_instr),

                          .IF_ID_npc(IF_ID_npc),

                          .EX_MEM_PCSrc(EX_MEM_PCSrc),

                          .EX_MEM_NPC(EX_MEM_NPC) );

        //initializes variables

        initial begin

        EX_MEM_PCSrc <= 0;

        EX_MEM_NPC <= 0;

        end
```

```
endmodule
```

Screenshot

```
PS E:\2023\Spring Semester\CSE 4010\CSE4010_Labs\Lab6a> iverilog -o pipeline.vvp pipeline.v
PS E:\2023\Spring Semester\CSE 4010\CSE4010_Labs\Lab6a> vvp pipeline.vvp
Time     PC      npc     dataout of MEM  IF_ID_instr     IF_ID_npc
0        0       z       a00000aa        00000000                00
1        z       z       xxxxxxxx        a00000aa                z0
2        z       z       xxxxxxxx        xxxxxxxx                z0
./ifetch.v:40: $finish called at 20 (1s)
PS E:\2023\Spring Semester\CSE 4010\CSE4010_Labs\Lab6a>
```