ECEN220

Jae Lee

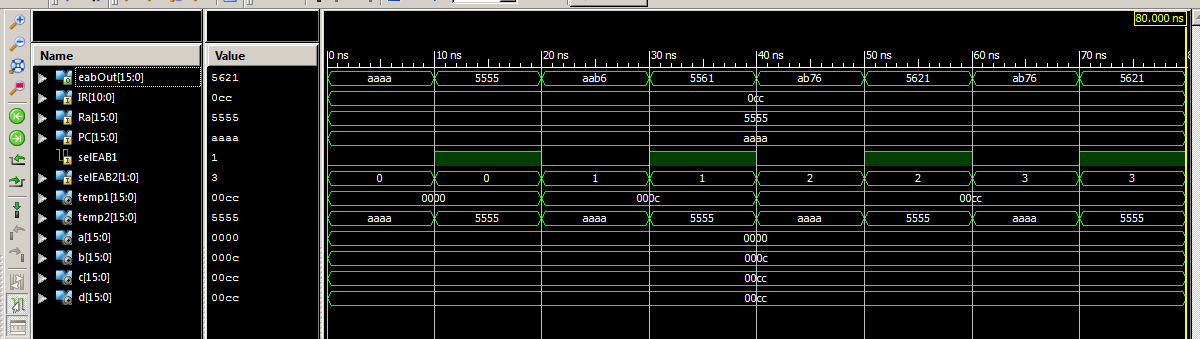
Lab #10 – Designing the LC3 Datapath

11/19/2013

**EAB Verilog file (3pt)**

|  |
| --- |
| `default\_nettype none // force error on undeclared net names  module EAB(eabOut, IR, Ra, PC, selEAB1, selEAB2);  output[15:0] eabOut;  input[10:0] IR;  input[15:0] Ra, PC;  input selEAB1;  input[1:0] selEAB2;    wire[15:0] temp1, temp2, a,b,c,d;  assign eabOut = temp1 + temp2;  assign a = 16'd0;  assign b = {{10{IR[5]}},IR[5:0]};  assign c = {{7{IR[8]}},IR[8:0]};  assign d = {{5{IR[10]}},IR[10:0]};    ADDR2MUX addmux2(temp1, selEAB2, a,b,c,d);  mux21n addmux1(temp2, selEAB1, PC, Ra);  endmodule |

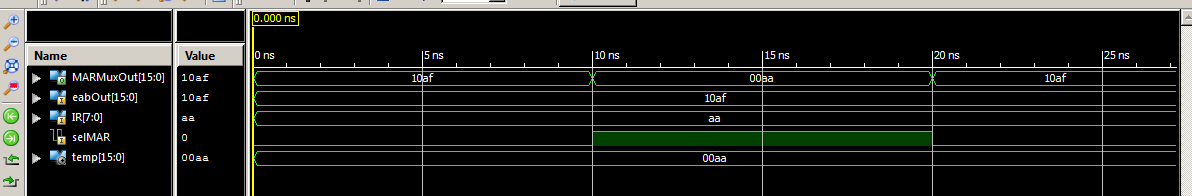
**EAB Simulation waveform (2pt)**



**MARMux Verilog file (3pt)**

|  |
| --- |
| `default\_nettype none // force error on undeclared net names  module MARMux(MARMuxOut, IR, eabOut, selMAR);  output[15:0] MARMuxOut;  input[15:0] eabOut;  input[7:0] IR;  input selMAR;    wire[15:0] temp;    assign temp = {{8'd0},IR};  mux21n mux1(MARMuxOut, selMAR, eabOut, temp);  endmodule |

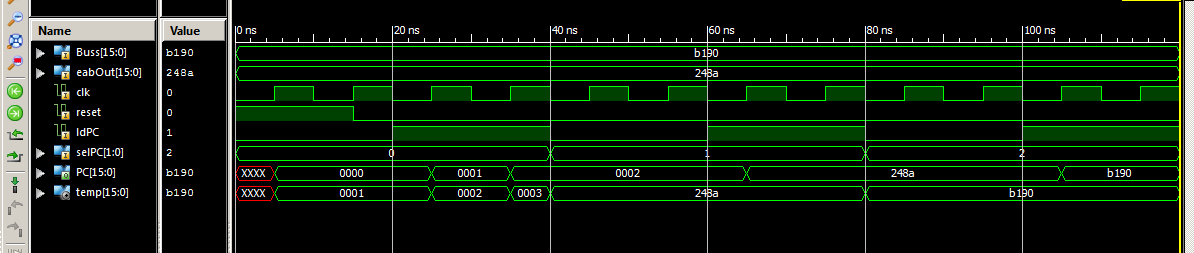
**MARMux Simulation waveform (2pt)**



**PC Verilog file (3pt)**

|  |
| --- |
| `default\_nettype none // force error on undeclared net names  module PC(Buss, clk, reset, ldPC, eabOut, selPC, PC);  input[15:0] Buss, eabOut;  input clk, reset, ldPC;  input[1:0] selPC;    output[15:0] PC;  wire[15:0] temp;    register R1(PC, clk, temp, reset, ldPC);      PCMUX p1(temp, selPC, PC+1, eabOut, Buss);    endmodule |

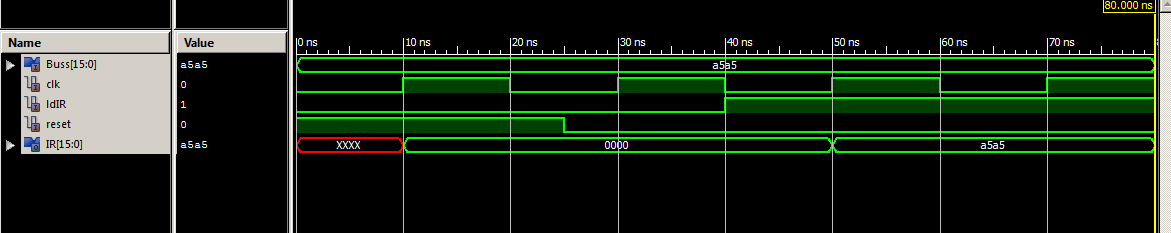
**PC Simulation waveform (2pt)**



**IR Verilog file (3pt)**

|  |
| --- |
| module IR(IR, clk, ldIR, reset, Buss);  input[15:0] Buss;  input clk, ldIR, reset;  output[15:0] IR;    register r1(IR, clk, Buss, reset, ldIR);  endmodule |

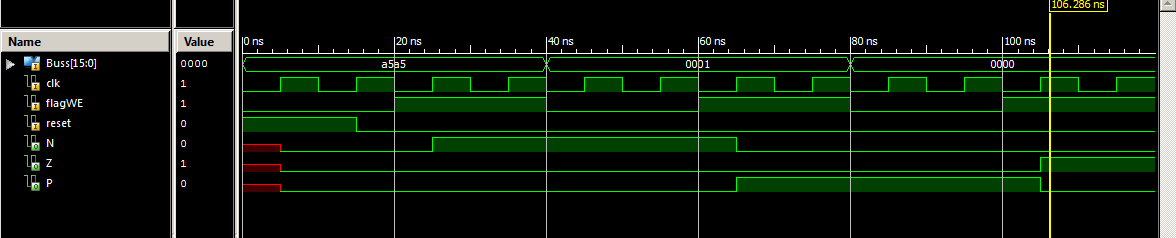
**IR Simulation waveform (2pt)**



**NZP Verilog file (3pt)**

|  |
| --- |
| module NZP(N, Z, P, clk, flagWE, reset, Buss);  input[15:0] Buss;  input clk, flagWE, reset;  output N, Z, P;  wire Nn, Nz, Np;    assign Nn = Buss[15];  assign Nz = ~|Buss;  assign Np = ~Buss[15] && ~Nz;    ff\_dce ffn(N, clk, Nn, reset, flagWE);  ff\_dce ffz(Z, clk, Nz, reset, flagWE);  ff\_dce ffp(P, clk, Np, reset, flagWE);  endmodule |

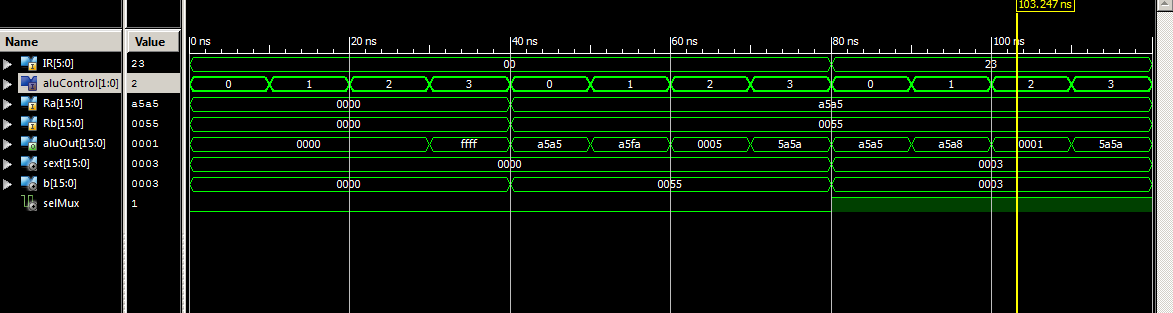
**NZP Simulation waveform (2pt)**



**ALU Verilog file (3pt)**

|  |
| --- |
| module ALU(Ra, Rb, IR, aluControl, aluOut);  input[5:0] IR;  input[1:0] aluControl;  input[15:0] Ra, Rb;  output[15:0] aluOut;  wire[15:0] sext, b;  wire selMux;      assign selMux = IR[5];  assign sext = {{11{IR[4]}} , IR[4:0]};  mux21n m21(b, selMux, Rb, sext);    assign aluOut = (aluControl == 2'b00) ? Ra:  (aluControl == 2'b01) ? Ra + b:  (aluControl == 2'b10) ? Ra & b:  ~Ra;    endmodule |

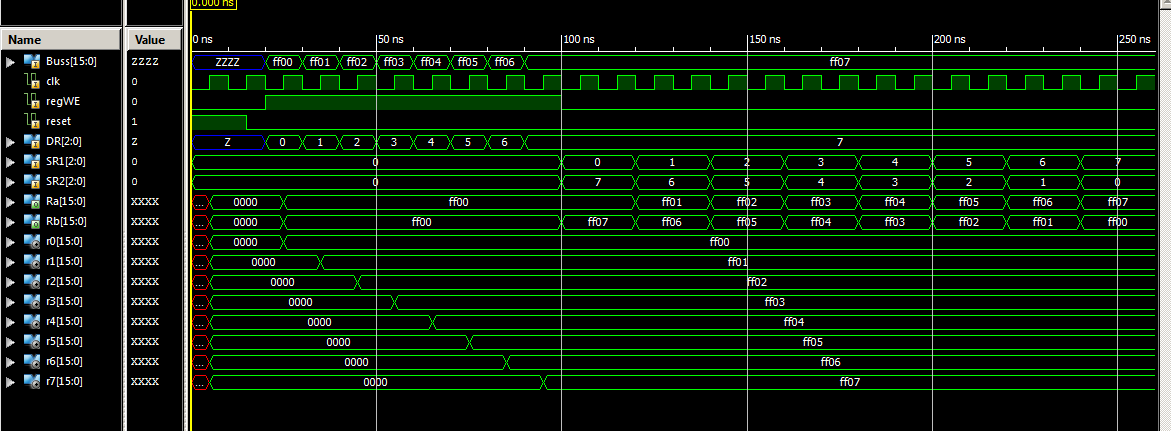
**ALU Simulation waveform (2pt)**



**RegFile Verilog file (3pt)**

|  |
| --- |
| module RegFile(Ra, Rb, Buss, clk, regWE, reset, DR, SR1, SR2);  input[15:0] Buss;  input clk, regWE, reset;  input[2:0] DR, SR1, SR2;  output[15:0] Ra, Rb;  wire[15:0] r0, r1, r2, r3, r4, r5, r6, r7;  wire[7:0] ld;    assign Ra = (SR1 == 3'b000)? r0:  (SR1 == 3'b001)? r1:  (SR1 == 3'b010)? r2:  (SR1 == 3'b011)? r3:  (SR1 == 3'b100)? r4:  (SR1 == 3'b101)? r5:  (SR1 == 3'b110)? r6:  r7;    assign Rb = (SR2 == 3'b000)? r0:  (SR2 == 3'b001)? r1:  (SR2 == 3'b010)? r2:  (SR2 == 3'b011)? r3:  (SR2 == 3'b100)? r4:  (SR2 == 3'b101)? r5:  (SR2 == 3'b110)? r6:  r7;    assign ld = (regWE == 1'b1 && DR == 3'b000)? 8'd1:  (regWE == 1'b1 && DR == 3'b001)? 8'd2:  (regWE == 1'b1 && DR == 3'b010)? 8'd4:  (regWE == 1'b1 && DR == 3'b011)? 8'd8:  (regWE == 1'b1 && DR == 3'b100)? 8'd16:  (regWE == 1'b1 && DR == 3'b101)? 8'd32:  (regWE == 1'b1 && DR == 3'b110)? 8'd64:  (regWE == 1'b1 && DR == 3'b111)?8'd128:  8'd0;    register R0(r0, clk, Buss, reset, ld[0]);  register R1(r1, clk, Buss, reset, ld[1]);  register R2(r2, clk, Buss, reset, ld[2]);  register R3(r3, clk, Buss, reset, ld[3]);  register R4(r4, clk, Buss, reset, ld[4]);  register R5(r5, clk, Buss, reset, ld[5]);  register R6(r6, clk, Buss, reset, ld[6]);  register R7(r7, clk, Buss, reset, ld[7]);  endmodule |

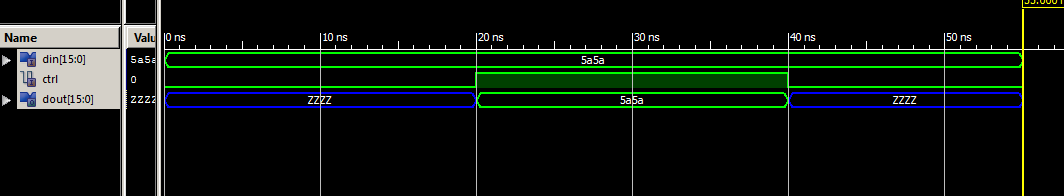
**RegFile Simulation waveform (2pt)**



**ts\_driver Verilog file (3pt)**

|  |
| --- |
| module ts\_driver ( din, dout, ctrl );  input [15:0] din;  input ctrl;  output [15:0] dout;  assign dout = (ctrl)? din:(16'bZZZZZZZZZZZZZZZZ);  endmodule |

**ts\_driver Simulation waveform (2pt)**



**Anomalies**

This lab was fun because I had good interest and understanding in LC3. It took a while to finish, but there weren’t many obstacles. Thank you for your help TAs. You guys are awesome !