Isai Mercado Oliveros

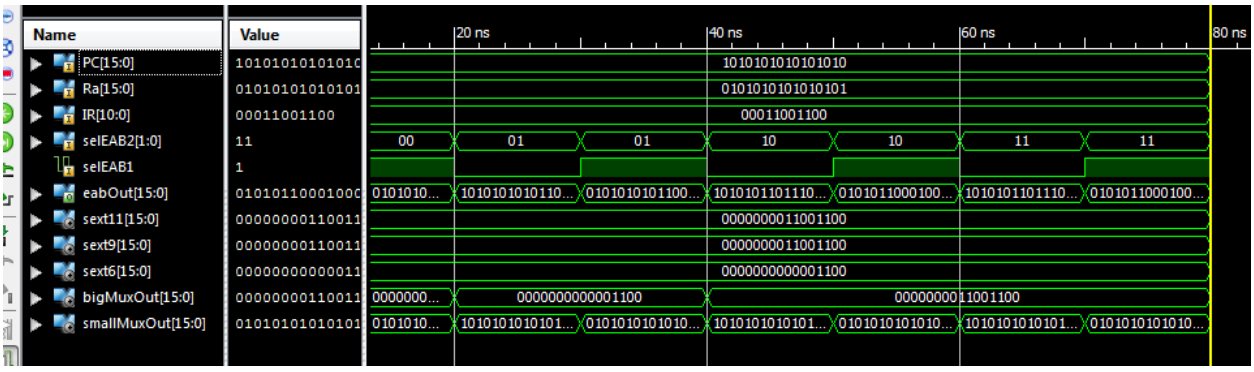
Lab 10

Nov 19, 2014

EAB Verilog file

```
//////////////////////////////////////////////////////////////////
module EAB(
    input[15:0] PC, Ra,
    input[10:0] IR,
    input[1:0] selEAB2,
    input selEAB1,
    output[15:0] eabOut
     );

    wire[15:0] sext11,sext9,sext6,bigMuxOut,smallMuxOut;

    assign sext11 = {{5{IR[10]}},IR[10:0]};
    assign sext9  = {{7{IR[8]}},IR[8:0]};
    assign sext6  = {{10{IR[5]}},IR[5:0]};

    assign bigMuxOut = (selEAB2 == 2'b00)? 16'd0 :
                       (selEAB2 == 2'b01)? sext6 :
                       (selEAB2 == 2'b10)? sext9 : sext11;

    assign smallMuxOut = (selEAB1 == 1'b0)? PC : Ra;

    assign eabOut = bigMuxOut + smallMuxOut;

endmodule
```

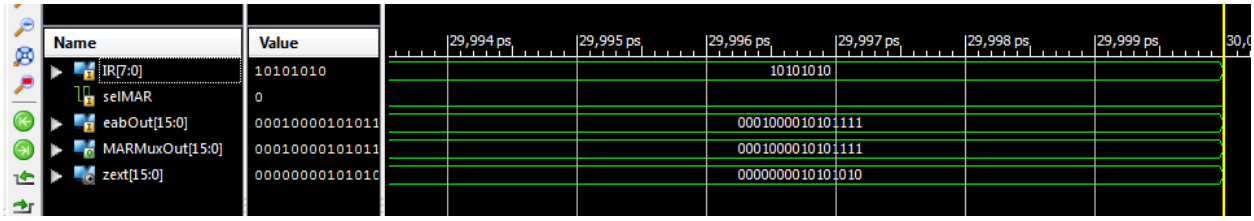
EAB Simulation waveform



MARMux Verilog file

```
20  //////////////////////////////////////////////////////////////////
21  module MARMux(
22     input[7:0] IR,
23     input selMAR,
24     input[15:0] eabOut,
25     output[15:0] MARMuxOut
26      );
27
28     wire[15:0] zext;
29
30     assign zext = {{8{0}},IR[7:0]};
31
32     assign MARMuxOut = (selMAR == 1'b1) ? zext : eabOut ;
33
34  endmodule
35
```

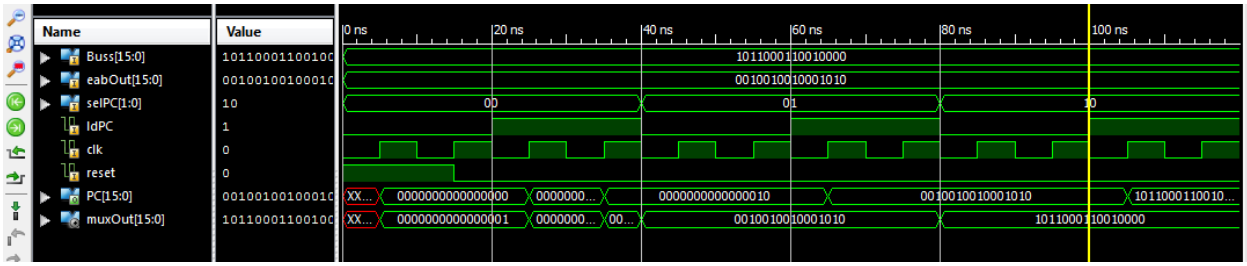
MARMux Simulation waveform



PC Verilog file

```
//
//////////////////////////////////////////////////////////////////
module PC(
   input[15:0] Buss, eabOut,
   input[1:0] selPC,
   input ldPC, clk, reset,
   output[15:0] PC
    );

   wire[15:0] muxOut;

   assign muxOut = (selPC == 2'b00) ? PC + 1 :
                   (selPC == 2'b01) ? eabOut:
                   (selPC == 2'b10) ? Buss : 16'd0 ;

   register rg (PC, clk, muxOut, reset, ldPC);

endmodule


module register(dout, clk, din, reset, load);

input clk, reset, load;
input [15:0] din;
output reg [15:0] dout;

always @(posedge clk)
 if (reset) dout <= 16'd0;
else if (load) dout <= din;
endmodule
```

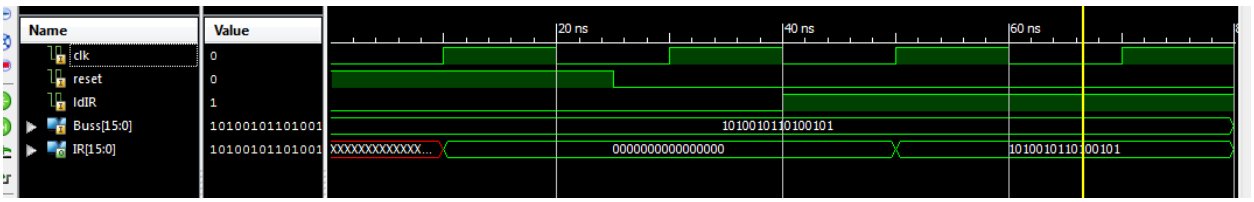
PC Simulation waveform



IR Verilog file

```
0  //////////////////////////////////////////
1  module IR(IR, clk, Buss, reset, ldIR);
2
3  input clk, reset, ldIR;
4  input [15:0] Buss;
5  output reg [15:0] IR;
6
7  always @(posedge clk)
8   if (reset) IR <= 16'd0;
9  else if (ldIR) IR <= Buss;
0  endmodule
1
```

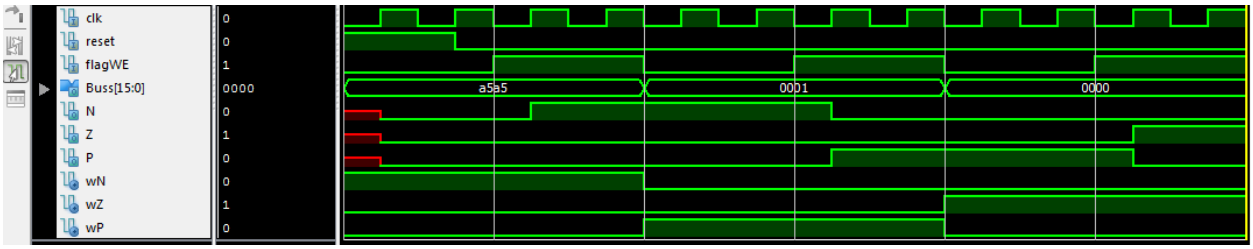
IR Simulation waveform



NZP Verilog file

```
20 //////////////////////////////////////////////////////////////////////////////////
21 module NZP(
22    input clk,reset,flagWE,
23    output[15:0] Buss,
24    output N,Z,P
25     );
26    wire wN,wZ,wP;
27
28    assign wN = (Buss[15] == 1'd1)       ? 1'd1 : 1'd0 ;
29    assign wZ = (Buss[15:0] == 16'd0)    ? 1'd1 : 1'd0 ;
30    assign wP = (Buss[15] != 1'd1 && Buss[15:0] != 16'd0)? 1'd1 : 1'd0 ;
31
32    FF_DCE ff0 (N,clk,wN,reset,flagWE);
33    FF_DCE ff1 (Z,clk,wZ,reset,flagWE);
34    FF_DCE ff2 (P,clk,wP,reset,flagWE);
35
36 endmodule
37
38
39 module FF_DCE(q, clk, d, clr, en);
40 input clk, clr, en, d;
41 output reg q;
42 always @(posedge clk)
43 if (clr) q <= 0;
44 else if (en)  q <= d;
45 endmodule
46
```

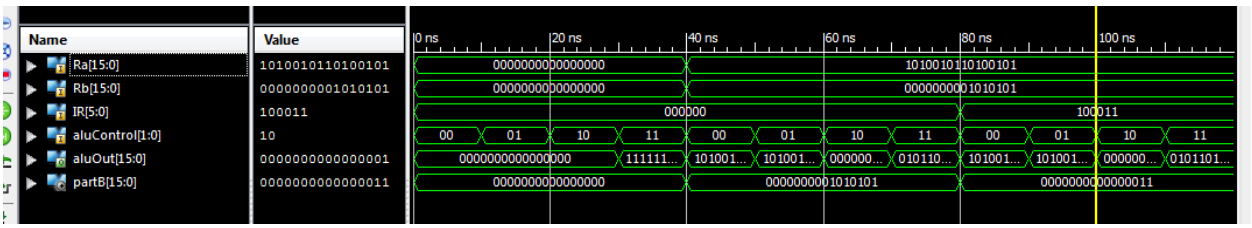
NZP Simulation waveform



ALU Verilog file

```
//
//////////////////////////////////////////////////////////////////////////////////
module ALU(
    input[15:0] Ra,Rb,
    input[5:0] IR,
    input[1:0] aluControl,
    output[15:0] aluOut
     );
    wire[15:0] partB;

    assign partB = (IR[5] == 1'b1)? {{11{IR[4]}},IR[4:0]} : Rb ;

    assign aluOut = (aluControl == 2'b00)? Ra         :
                    (aluControl == 2'b01)? partB + Ra :
                    (aluControl == 2'b10)? partB & Ra : ~Ra ;
endmodule

```

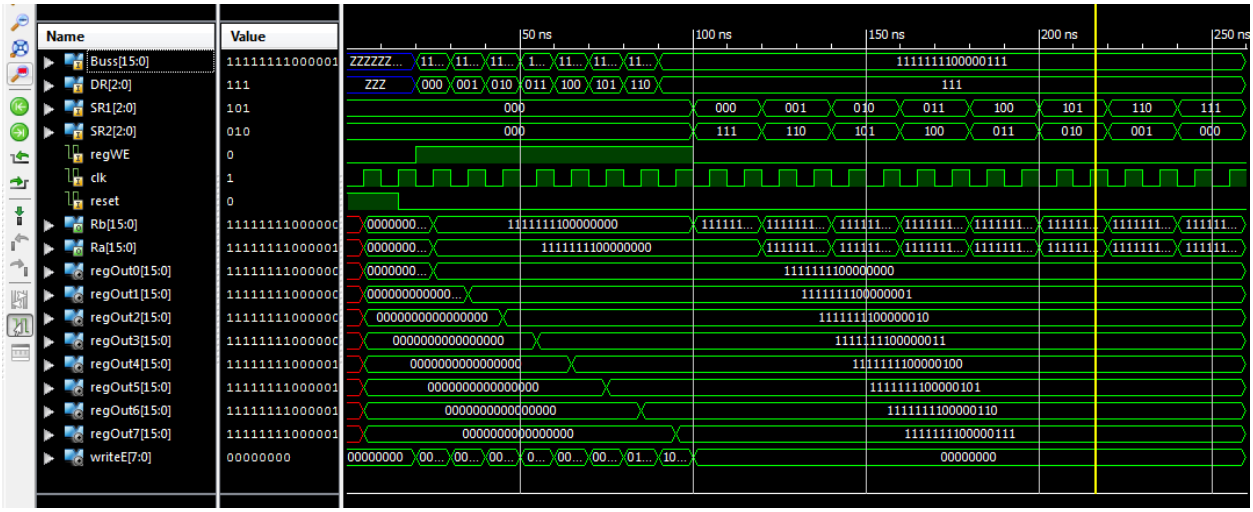
ALU Simulation waveform



RegFile Verilog file

```
20  //////////////////////////////////////////////////////////////////////////////////
21  module RegFile(
22     input[15:0] Buss,
23     input[2:0] DR, SR1, SR2,
24     input regWE, clk, reset,
25     output[15:0] Rb, Ra
26      );
27
28     wire[15:0] regOut0,regOut1,regOut2,regOut3,regOut4,regOut5,regOut6,regOut7;
29     wire[7:0] writeE;
30
31     assign Ra = (SR1 == 3'd0) ? regOut0 :
32                 (SR1 == 3'd1) ? regOut1 :
33                 (SR1 == 3'd2) ? regOut2 :
34                 (SR1 == 3'd3) ? regOut3 :
35                 (SR1 == 3'd4) ? regOut4 :
36                 (SR1 == 3'd5) ? regOut5 :
37                 (SR1 == 3'd6) ? regOut6 :
38                 (SR1 == 3'd7) ? regOut7 : 16'd0 ;
39
40     assign Rb = (SR2 == 3'd0) ? regOut0 :
41                 (SR2 == 3'd1) ? regOut1 :
42                 (SR2 == 3'd2) ? regOut2 :
43                 (SR2 == 3'd3) ? regOut3 :
44                 (SR2 == 3'd4) ? regOut4 :
45                 (SR2 == 3'd5) ? regOut5 :
46                 (SR2 == 3'd6) ? regOut6 :
47                 (SR2 == 3'd7) ? regOut7 : 16'd0 ;
48
49     assign writeE = (regWE == 1'b1) ? 8'b00000001 << DR : 8'd0;
50
51     register r0 (regOut0, clk, Buss, reset, writeE[0]); //(dout, clk, din, reset, load);
52     register r1 (regOut1, clk, Buss, reset, writeE[1]);
53     register r2 (regOut2, clk, Buss, reset, writeE[2]);
54     register r3 (regOut3, clk, Buss, reset, writeE[3]);
55     register r4 (regOut4, clk, Buss, reset, writeE[4]);
56     register r5 (regOut5, clk, Buss, reset, writeE[5]);
57     register r6 (regOut6, clk, Buss, reset, writeE[6]);
58     register r7 (regOut7, clk, Buss, reset, writeE[7]);
59  endmodule
60
```

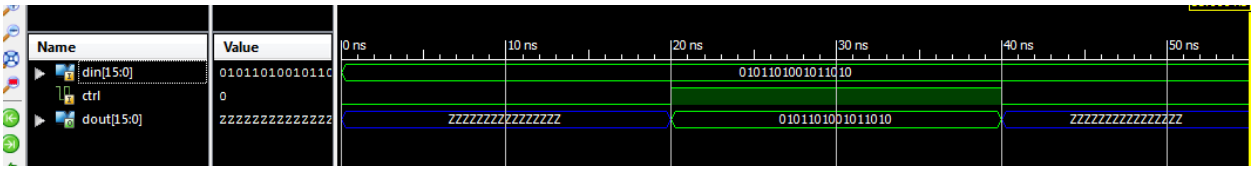
RegFile Simulation waveform



ts_driver Verilog file

```
0  ////////////////////////////////////////////////////
1  module ts_driver ( din, dout, ctrl );
2   input [15:0] din;
3  input ctrl;
4  output [15:0] dout;
5
6  assign dout = (ctrl)? din:(16'bZZZZZZZZZZZZZZZZ);
7  endmodule
8
```

ts_driver Simulation waveform



Anomalies

Fun lab. I am excited to build the LC-3!!!