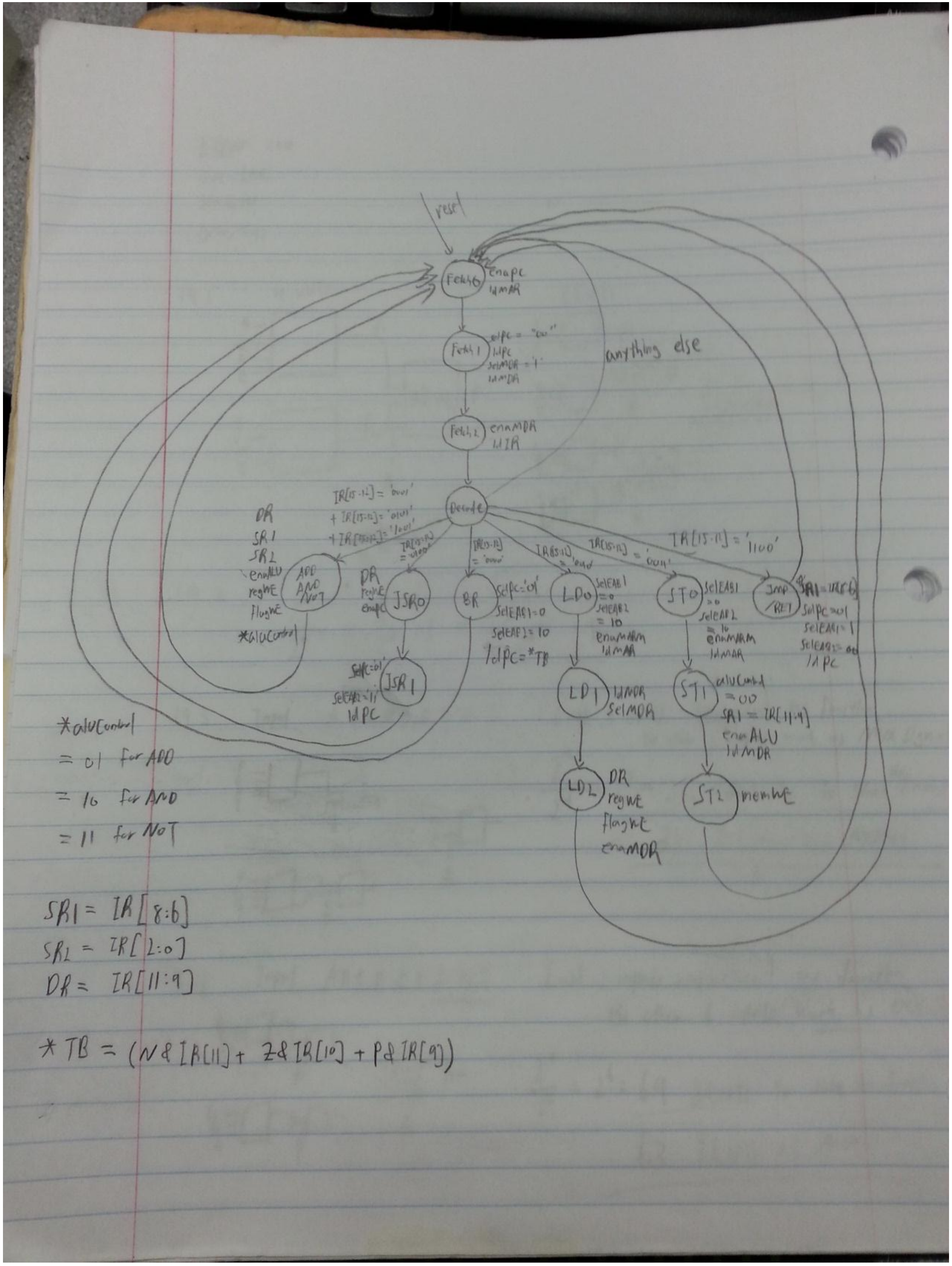
ECEN 220
Jae Lee
Lab 12 – LC3 Full simulation
12/11/2013

State Machine Diagram (6pt)



*aluControl
= 01 for ADD
= 10 for AND
= 11 for NOT

SR1 = IR[8:6]
SR2 = IR[2:0]
DR = IR[11:9]

*TB = (N & IR[11] + Z & IR[10] + P & IR[9])

Controller Verilog file (7pt)

```
module LC3_Control(IR,N,Z,P,aluControl,SR1,SR2,DR,selPC,selEAB1,selEAB2,

    enaALU,regWE,flagWE,enaMARM,selMAR,enaPC,ldPC,ldIR,ldMAR,ldMDR,selMDR,memWE,enaMDR,reset,clk);

        input reset,clk;
        input[15:0] IR;
        input N,Z,P;
        output reg[1:0] aluControl,selPC,selEAB2;
        output reg[2:0] SR1,SR2,DR;
        output reg selEAB1,enaALU,regWE,flagWE,enaMARM,selMAR,enaPC,ldPC,ldIR,ldMAR,ldMDR,selMDR,memWE,enaMDR;

        reg[14:0] CS;
        reg[3:0] OPcode;

        parameter fetch0 = 15'h0001;
        parameter fetch1 = 15'h0002;
        parameter fetch2 = 15'h0004;
        parameter decode = 15'h0008;
        parameter AAN = 15'h0010;
        parameter JSR0 = 15'h0020;
        parameter JSR1 = 15'h0040;
        parameter BR = 15'h0080;
        parameter LD0 = 15'h0100;
        parameter LD1 = 15'h0200;
        parameter LD2 = 15'h0400;
        parameter ST0 = 15'h0800;
        parameter ST1 = 15'h1000;
        parameter ST2 = 15'h2000;
        parameter JMRE = 15'h4000;

        always @(posedge clk)
        begin
            if(reset)
                CS <= fetch0;

            else
                case(CS)
                    fetch0:
                        CS = fetch1;
                    fetch1:
                        CS = fetch2;
                    fetch2:
                        CS = decode;
                    decode:
```

```
                    begin
                        OPcode = IR[15:12];
                        if(OPcode == 4'b0001 || OPcode == 4'b0101 ||
OPcode == 4'b1001)
                            CS = AAN;
                        else if(OPcode == 4'b0100)
                            CS = JSR0;
                        else if(OPcode == 4'b0000)
                            CS = BR;
                        else if(OPcode == 4'b0010)
                            CS = LD0;
                        else if(OPcode == 4'b0011)
                            CS = ST0;
                        else if(OPcode == 4'b1100)
                            CS = JMRE;
                        else
                            CS = fetch0;
                    end
                    AAN:
                        CS = fetch0;
                    JSR0:
                        CS = JSR1;
                    JSR1:
                        CS = fetch0;
                    BR:
                        CS = fetch0;
                    LD0:
                        CS = LD1;
                    LD1:
                        CS = LD2;
                    LD2:
                        CS = fetch0;
                    ST0:
                        CS = ST1;
                    ST1:
                        CS = ST2;
                    ST2:
                        CS = fetch0;
                    JMRE:
                        CS = fetch0;
                endcase
        end

        always @(CS)
        begin
            enaALU = 1'b0;
            enaMARM = 1'b0;
            enaPC = 1'b0;
```

```
enaMDR = 1'b0;
ldIR  = 1'b0;
ldPC = 1'b0;
ldMAR = 1'b0;
ldMDR = 1'b0;
regWE = 1'b0;
memWE = 1'b0;
flagWE = 1'b0;

case(CS)

        fetch0://fetch0
        begin
                enaPC = 1'b1;
                ldMAR  = 1'b1;
        end

        fetch1://fetch1
        begin
                ldPC = 1'b1;
                selPC = 2'b00;
                ldMDR = 1'b1;
                selMDR = 1'b1;
        end

        fetch2://fetch2
        begin
                ldIR = 1'b1;
                enaMDR = 1'b1;
        end

        AAN://add,and,not
        begin
                enaALU = 1'b1;
                regWE = 1'b1;
                flagWE = 1'b1;
                aluControl = (IR[15:12] == 4'b0001)? 2'b01:
                                         (IR[15:12] == 4'b0101)? 2'b10:
                                         2'b11;
                SR1 = IR[8:6];
                SR2 = IR[2:0];
                DR = IR[11:9];
        end

        JSR0://JSR0
        begin
                DR = 3'b111;
                regWE = 1'b1;
                enaPC = 1'b1;
        end

        JSR1://JSR1
        begin
                selPC = 2'b01;
                selEAB1 = 1'b0;
                selEAB2 = 2'b11;
                ldPC = 1'b1;
        end

        BR://BR
        begin
                selPC = 2'b01;
                selEAB1 = 1'b0;
                selEAB2 = 2'b10;
                ldPC = ((N & IR[11]) | (Z & IR[10]) | (P & IR[9]));
        end

        LD0://LD0
        begin
                selEAB1 = 1'b0;
                selEAB2 = 2'b10;
                enaMARM = 1'b1;
                selMAR = 1'b0;
                ldMAR = 1'b1;
        end

        LD1://LD1
        begin
                ldMDR = 1'b1;
                selMDR = 1'b1;
        end

        LD2://LD2
        begin
                DR = IR[11:9];
                regWE = 1'b1;
                flagWE = 1'b1;
                enaMDR = 1'b1;
        end
        ST0://ST0
        begin
                selEAB1 = 1'b0;
                selEAB2 = 2'b10;
                enaMARM = 1'b1;
                selMAR = 1'b0;
                ldMAR = 1'b1;
```

```
                end

                ST1://ST1
                begin
                    aluControl = 2'b00;
                    SR1 = IR[11:9];
                    enaALU = 1'b1;
                    ldMAR = 1'b1;
                end

                ST2://ST2
                begin
                    memWE = 1'b1;
                end

                JMRE://JMP,RET
                begin
                    SR1 = IR[8:6];
                    selPC = 2'b01;
                    selEAB1 = 1'b1;
                    selEAB2 = 2'b00;
                    ldPC = 1'b1;
                end
            endcase
        end

endmodule
```

Full LC3 Verilog file (7pt)

```
module LC3(reset, clk);

    input reset, clk;

    wire[15:0] IR;
    wire N,Z,P;

    wire[1:0] aluControl,selPC,selEAB2;
    wire[2:0] SR1,SR2,DR;
    wire selEAB1,enaALU,regWE,flagWE,enaMARM,selMAR,enaPC,ldPC,ldIR,ldMAR,ldMDR,selMDR,memWE,enaMDR;


    LC3_Control LC3_C(IR,N,Z,P,aluControl,SR1,SR2,DR,selPC,selEAB1,selEAB2,

    enaALU,regWE,flagWE,enaMARM,selMAR,enaPC,ldPC,ldIR,ldMAR,ldMDR,selMDR,memWE,enaMDR,reset,clk);
```

```
            Datapath DP(reset, clk, IR, N, Z, P, aluControl, SR1, SR2, DR, selPC, selEAB2, enaALU,
            regWE, enaMARM, selMAR, selEAB1, enaPC, ldPC, ldIR, ldMAR, ldMDR, selMDR, memWE,
enaMDR, flagWE);


endmodule
```

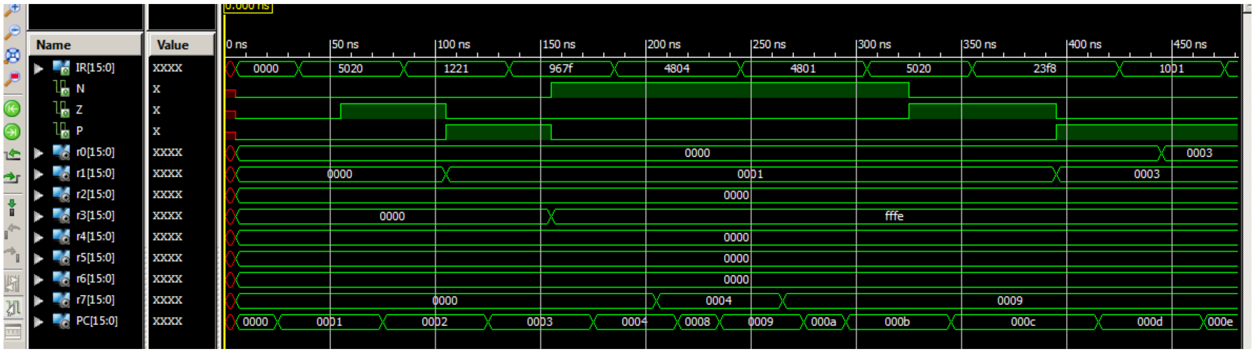
Master TCL file (2pt)

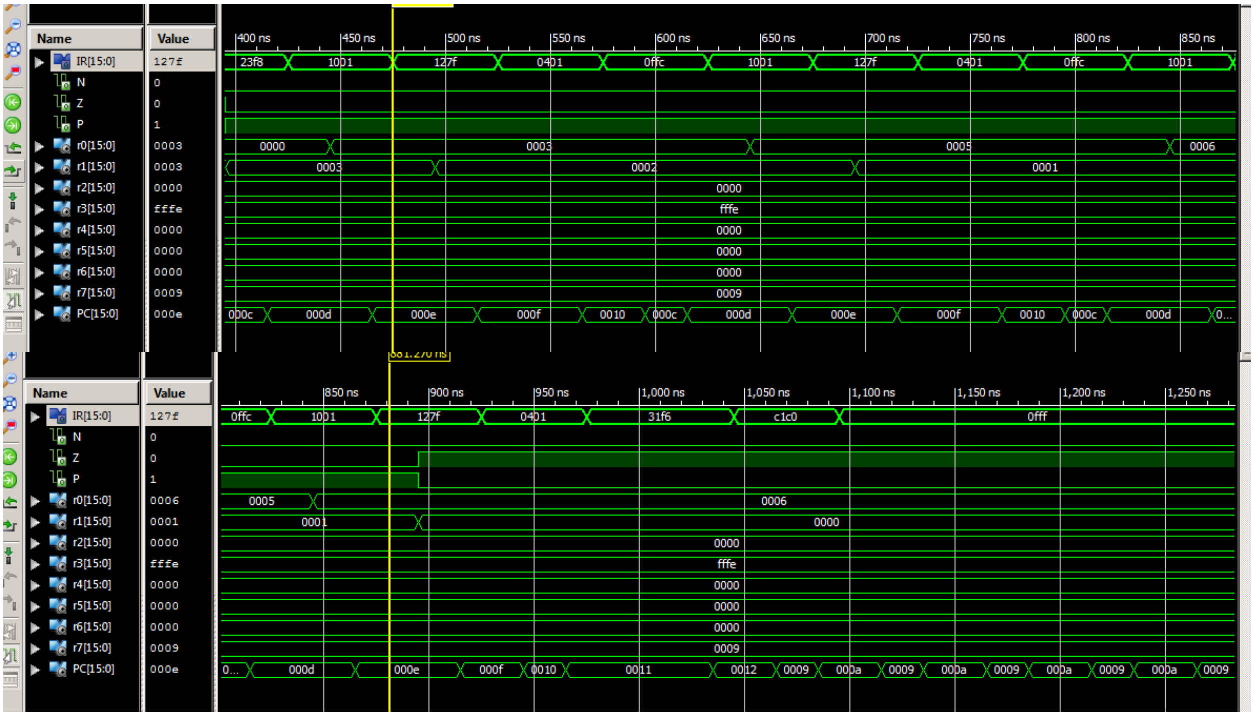
```
wave add /DP/IR -radix hex
wave add /DP/N -radix hex
wave add /DP/Z -radix hex
wave add /DP/P -radix hex
wave add /DP/reg0/r0 -radix hex
wave add /DP/reg0/r1 -radix hex
wave add /DP/reg0/r2 -radix hex
wave add /DP/reg0/r3 -radix hex
wave add /DP/reg0/r4 -radix hex
wave add /DP/reg0/r5 -radix hex
wave add /DP/reg0/r6 -radix hex
wave add /DP/reg0/r7 -radix hex
wave add /DP/PC -radix hex



isim force add clk 0 -time 0 -value 1 -time 5ns -repeat 10ns
isim force add reset 1 -time 0 -value 0 -time 8ns
run 2000ns
```

All Simulation waveforms(18pt)





Anomalies

My lab partner and I had a hard time to make LC3 control Verilog code because we were not familiar with the Behavioral Verilog. However, we found a good Verilog example of putting a state machine into a Verilog code. Once things started rolling, everything fell into the place.

TA evaluation

All of you guys are great. I really learned a lot from you guys. Thank you for your help on home works and labs. I don't have any suggestion for individual person, but in general, sometimes I had to wait so long to get help while doing lab.