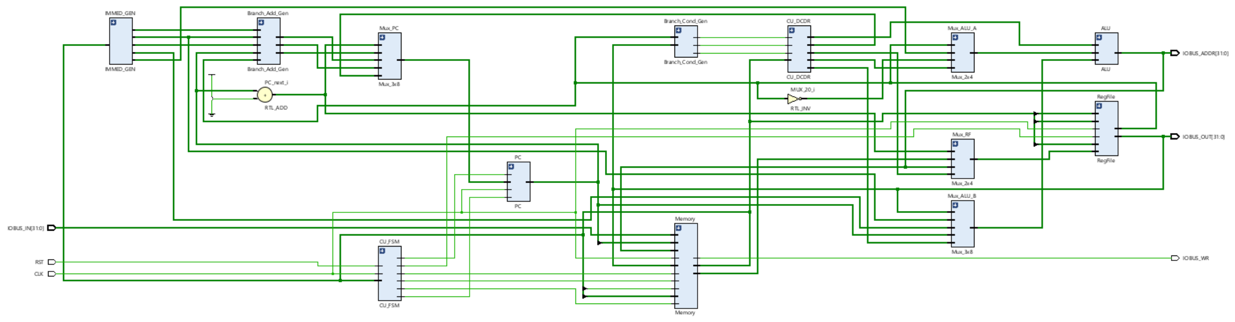
CPE 233 HW 6

Wyatt Tack

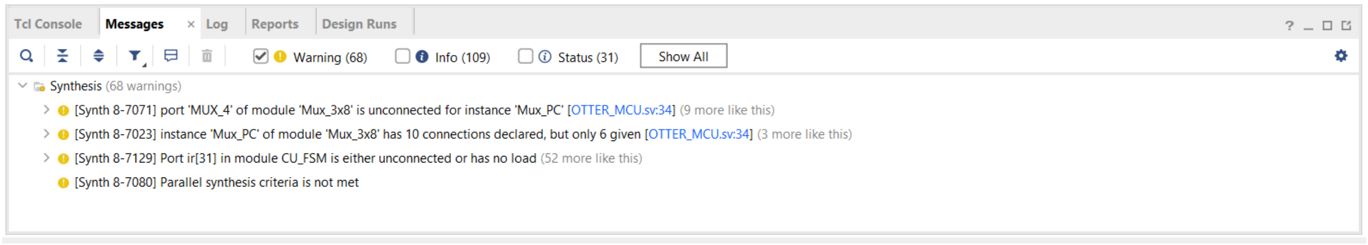
1. Behavior Description: The Control Unit Decoder and Control Unit FSM (finite state machine) act in a pair as the main conductor of the whole MCU, with the Decoder determining all of the asynchronous multiplexor switches, selecting which data goes to where, while the FSM controls the synchronous switches for the read/write enables on the data-storing parts of the microprocessor. The decoder is a combinational circuit that takes in the op-code and function code of the machine code stored in the program memory, and uses that to determine the flow of data throughout. The FSM uses a clock to schedule the circuit, to determine weather the memory module should take in a program address, or spit out a machine code, or for a few cases spit out data stored in the memory.

   Overall is the OTTER MCU, which stores machine code written using the assembler's manual for the Risc-V ISA. The otter uses a counter to address the program memory, and count up for each address, pointing to another instruction of machine code, going line by line, or for certain cases moving around the program. The memory spews out the machine code which is acted on by a series of combinational circuitry, and the next 'Fetching' of the machine code takes place, until the reset switch is hit and the program is started over.

2. Structural Design for whole OTTER:



3. Synthesis Warnings Listing:

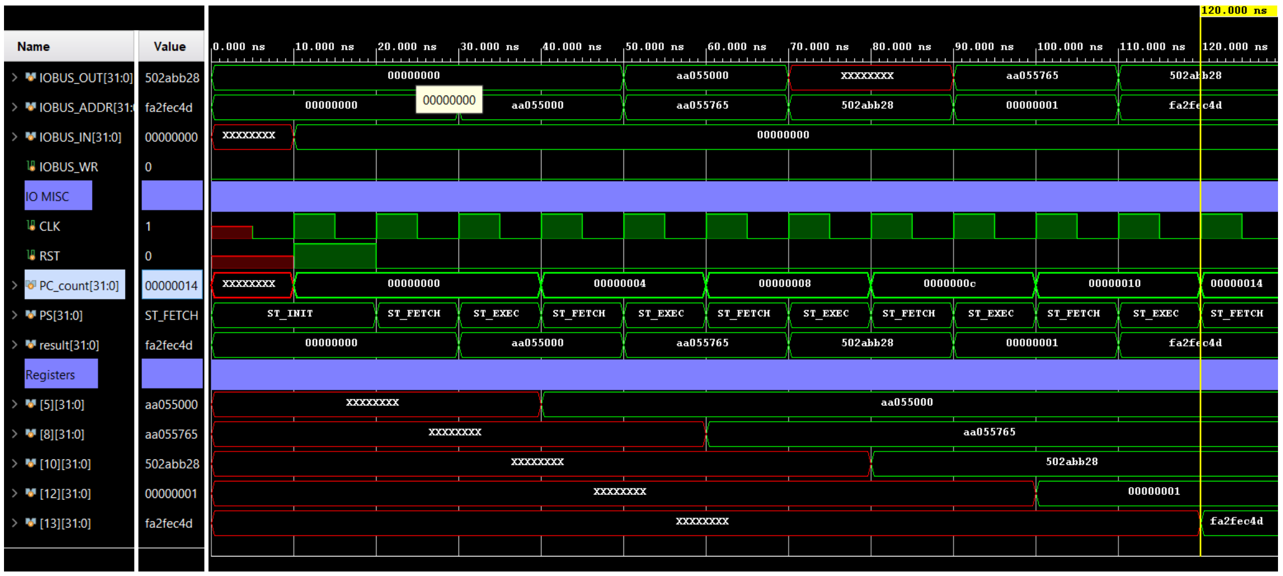


```
Synthesis (68 warnings)
  [Synth 8-7071] port 'MUX_4' of module 'Mux_3x8' is unconnected for instance 'Mux_PC' [OTTER_MCU.sv:34] (9 more like this)
  [Synth 8-7023] instance 'Mux_PC' of module 'Mux_3x8' has 10 connections declared, but only 6 given [OTTER_MCU.sv:34] (3 more like this)
  [Synth 8-7129] Port ir[31] in module CU_FSM is either unconnected or has no load (52 more like this)
  [Synth 8-7080] Parallel synthesis criteria is not met
```

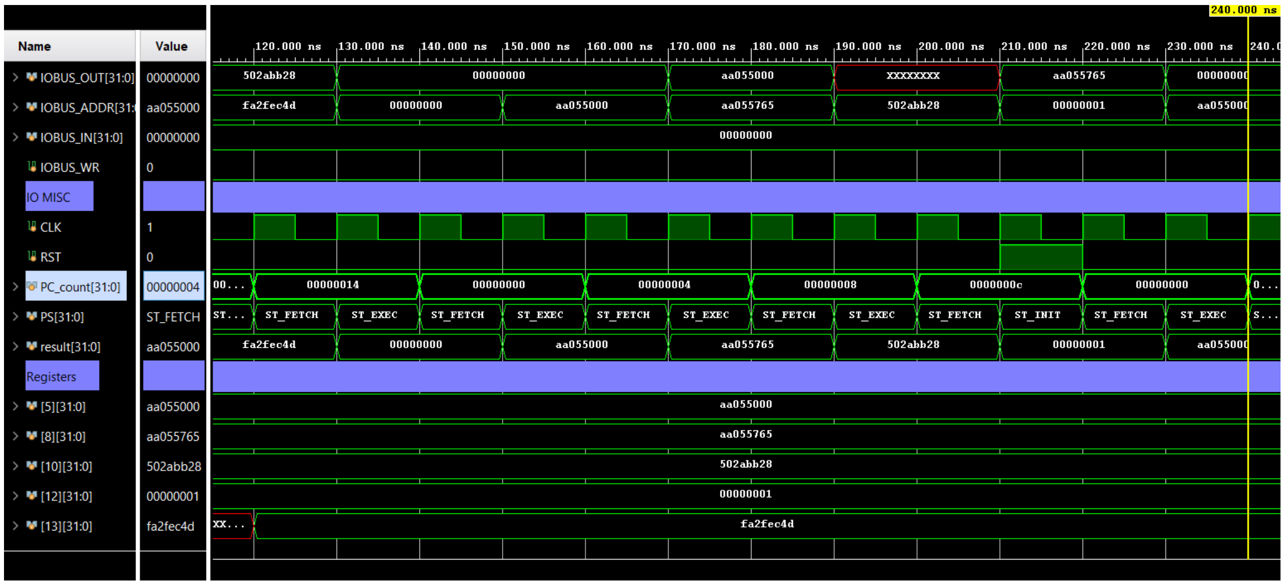
4. Verification:
    a. Assembly instructions:

```
main:  lui    x5,  0xAA055
       addi   x8,  x5, 0x765
       slli   x10, x8, 3
       slt    x12, x5, x8
       xor    x13, x8, x10
       beq    x0,  x0, main
```

Simulation:



Name	Value
IOBUS_OUT[31:0]	502abb28
IOBUS_ADDR[31:	fa2fec4d
IOBUS_IN[31:0]	00000000
IOBUS_WR	0
IO MISC	
CLK	1
RST	0
PC_count[31:0]	00000014
PS[31:0]	ST_FETCH
result[31:0]	fa2fec4d
Registers	
[5][31:0]	aa055000
[8][31:0]	aa055765
[10][31:0]	502abb28
[12][31:0]	00000001
[13][31:0]	fa2fec4d



Name	Value
IOBUS_OUT[31:0]	00000000
IOBUS_ADDR[31:	aa055000
IOBUS_IN[31:0]	00000000
IOBUS_WR	0
IO MISC	
CLK	1
RST	0
PC_count[31:0]	00000004
PS[31:0]	ST_FETCH
result[31:0]	aa055000
Registers	
[5][31:0]	aa055000
[8][31:0]	aa055765
[10][31:0]	502abb28
[12][31:0]	00000001
[13][31:0]	fa2fec4d

5. System Verilog Source Code:

For Control Unit Decoder:

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////
// Company: Cal Poly SLO
// Engineer: Wyatt Tack
//
// Create Date: 02/01/2024
// Design Name: Control Unit Decoder
// Project Name: OTTER MCU
// Target Devices: Basys 3 Board
// Description: Sends non timing dependent selection signals
//              for data manipulation in otter MCU
//
////////////////////////////////////////////////////////////

module CU_DCDR(
input [31:0] ir,
input br_eq, br_lt, br_ltu,
//input int_taken
output logic [3:0] ALU_FUN,
output logic [1:0] srcA_SEL,
output logic [2:0] srcB_SEL,
output logic [2:0] PC_SEL,
output logic [1:0] RF_SEL
    );
always_comb begin
    ALU_FUN = 0;
    srcA_SEL = 0;
    srcB_SEL = 0;
    PC_SEL = 0;
    RF_SEL = 0;
    case (ir[6:0]) //op-codes
    7'b0110011: //All R-Type
    begin
        ALU_FUN = {ir[30], ir[14:12]};
        srcA_SEL = 0;
        srcB_SEL = 0;
        PC_SEL = 0;
        RF_SEL = 3;
    end
    7'b0010011: //1st set of I-Type Instructions
    begin
        ALU_FUN = {1'b0, ir[14:12]};
        srcA_SEL = 0;
        srcB_SEL = 1;
        PC_SEL = 0;
        RF_SEL = 3;
    end
    7'b0000011: //2nd set of I-Type Instructions
    begin
        ALU_FUN = 4'b0000;
        srcA_SEL = 0;
        srcB_SEL = 1;
        PC_SEL = 0;
        RF_SEL = 2;
    end
    7'b1100111: //Last set of I-Type (for jalr)
    begin
        PC_SEL = 1;
        RF_SEL = 0;
    end
    7'b0100011: //All S-Type Instructions
    begin
        ALU_FUN = 4'b0000;
        srcA_SEL = 0;
        srcB_SEL = 2;
        PC_SEL = 0;
    end
    7'b1100011: //All B-Type Instructions (include conditions)
    begin
        case(ir[14:13])
        2'b00: //equal
        begin
        PC_SEL = {1'b0,(ir[12]^br_eq),1'b0};
        end
        2'b10: //less than
        begin
        PC_SEL = {1'b0,(ir[12]^br_lt),1'b0};
        end
        2'b11: //less than unsigned
        begin
        PC_SEL = {1'b0,(ir[12]^br_ltu),1'b0};
        end
        default:
        begin
        PC_SEL = 0;
        end
        endcase
    end
    7'b0110111: //1st set of U-Type (lui)
    begin
        ALU_FUN = 4'b1001;
        srcA_SEL = 1;
        PC_SEL = 0;
        RF_SEL = 3;
    end
    7'b0010111: //2nd set of U-Type (auipc)
    begin     //auipc********************************auipc
        ALU_FUN = 4'b0000;
        srcA_SEL = 1;
        srcB_SEL = 3;
        PC_SEL = 0;
        RF_SEL = 3;
    end
    7'b1101111: //All J-Type instructions (jal)
    begin
        PC_SEL = 3;
        RF_SEL = 0;
    end
    default:
    begin
    ALU_FUN = 0;
    srcA_SEL = 0;
    srcB_SEL = 0;
    PC_SEL = 0;
    RF_SEL = 0;
    end
    endcase
end
endmodule
```

For Control Unit FSM:

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////
// Company: Cal Poly SLO
// Engineer: Wyatt Tack
//
// Create Date: 02/01/2024
// Design Name: Control Unit FSM
// Project Name: OTTER MCU
// Target Devices: Basys 3 Board
// Description: Sends selection signals for timing specific
//              operations in memory of OTTER MCU
//
/////////////////////////////////////////////////////////////


module CU_FSM(
input RST, clk,
//input INTR,
input [31:0] ir,
output logic PC_WE, RF_WE, memWE2, memRDEN1, memRDEN2, reset
//output logic csr_WE, int_taken, mret_exec
    );
typedef enum {ST_INIT, ST_FETCH, ST_EXEC, ST_WRITE} state_type;
state_type NS, PS;
always_ff@(posedge clk) begin //state register
    if(RST == 1) PS<=ST_INIT;
    else PS<=NS;
end
always_comb begin //input/output logic
    PC_WE = 0;
    RF_WE = 0;
    memWE2 = 0;
    memRDEN1 = 0;
    memRDEN2 = 0;
    reset = 0;
//  csr_WE = 0;
//  int_taken = 0;
//  mret_exec = 0;
case(PS)
    ST_INIT: begin
    reset = 1'b1;
    NS = ST_FETCH;
    end
    ST_FETCH: begin
    memRDEN1 = 1'b1;
    NS = ST_EXEC;
    end
    ST_EXEC: begin
        NS = ST_FETCH;
        case (ir[6:0]) //op-codes
        7'b0110011: //All R-Type
        begin
            PC_WE = 1;
            RF_WE = 1;
        end
        7'b0010011: //1st set of I-Type Instructions
        begin
            PC_WE = 1;
            RF_WE = 1;
        end
        7'b0000011: //2nd set of I-Type Instructions
        begin
            NS = ST_WRITE;
            memRDEN2 = 1;
        end
        7'b1100111: //Last set of I-Type (for jalr)
        begin
            PC_WE = 1;
            RF_WE = 1;
        end
        7'b0100011: //All S-Type Instructions
        begin
            PC_WE = 1;
            memWE2 = 1;
        end
        7'b1100011: //All B-Type Instructions (include conditions)
        begin
            PC_WE = 1;
        end
        7'b0110111: //1st set of U-Type (lui)
        begin
            PC_WE = 1;
            RF_WE = 1;
        end
        7'b0010111: //2nd set of U-Type (auipc)
        begin
            PC_WE = 1;
            RF_WE = 1;
        end
        7'b1101111: //All J-Type instructions (jal)
        begin
            PC_WE = 1;
            RF_WE = 1;
        end
        default:
        begin

        end
        endcase
    end
    ST_WRITE: begin
        NS = ST_FETCH;
        PC_WE = 1;
        RF_WE = 1;
    end
    default: begin
    NS = ST_INIT;
    end
endcase
end
endmodule
```

For Whole OTTER MCU:

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////
// Company: Cal Poly SLO
// Engineer: Wyatt Tack
//
// Create Date: 02/07/2024
// Design Name: OTTER MCU
// Project Name: OTTER MCU
// Target Devices: Basys 3 Board
// Description: OTTER Microprocessor
//
//
/////////////////////////////////////////////////////////////////

module OTTER_MCU(
input RST, CLK,
//input INTR,
input [31:0] IOBUS_IN,
output IOBUS_WR,
output [31:0] IOBUS_OUT, IOBUS_ADDR
    );
//all signals mapped on OTTER diagram
logic[31:0] PC_count, PC_in, PC_next, ir, rs1, rs2, ALU_srcA,
         ALU_srcB, result, w_data, DOUT2;
logic [31:0] J_Type, B_Type, U_Type, I_Type, S_Type;
logic [31:0] jalr, branch, jal;
logic br_eq, br_lt, br_ltu;
logic [3:0] ALU_FUN;
logic [2:0] srcB_SEL, PC_SEL;
logic [1:0] srcA_SEL, RF_SEL;
logic PC_WE, RF_WE, memWE2, memRDEN1, memRDEN2, reset;
assign PC_next = PC_count + 4;
assign IOBUS_OUT = rs2;
assign IOBUS_ADDR = result;
//all modules mapped on OTTER diagram connected with respective signals
Mux_3x8 Mux_PC (.SEL(PC_SEL), .MUX_0(PC_next), .MUX_1(jalr),
         .MUX_2(branch), .MUX_3(jal), .MUX_out(PC_in));
PC PC (.PC_in(PC_in), .reset(reset), .PC_WE(PC_WE),
         .clk(CLK), .PC_count(PC_count));
Memory Memory (.MEM_CLK(CLK), .MEM_RDEN1(memRDEN1), .MEM_RDEN2(memRDEN2),
         .MEM_WE2(memWE2), .MEM_ADDR1(PC_count[15:2]), .MEM_ADDR2(result),
         .MEM_DIN2(rs2), .MEM_SIZE(ir[13:12]), .MEM_SIGN(ir[14]),
         .IO_IN(IOBUS_IN), .IO_WR(IOBUS_WR), .MEM_DOUT1(ir), .MEM_DOUT2(DOUT2));
Mux_2x4 Mux_RF (.SEL(RF_SEL), .MUX_0(PC_next), .MUX_2(DOUT2),
         .MUX_3(result), .MUX_out(w_data));
RegFile RegFile (.en(RF_WE), .adr1(ir[19:15]), .adr2(ir[24:20]), .w_adr(ir[11:7]),
         .w_data(w_data), .clk(CLK), .rs1(rs1), .rs2(rs2));
IMMED_GEN IMMED_GEN (.Instruction(ir), .U_Type(U_Type), .I_Type(I_Type), .S_Type(S_Type),
         .J_Type(J_Type), .B_Type(B_Type));
Branch_Add_Gen Branch_Add_Gen (.PC(PC_count), .JType(J_Type), .BType(B_Type),
         .IType(I_Type), .rs1(rs1), .jal(jal), .branch(branch), .jalr(jalr));
Mux_2x4 Mux_ALU_A (.SEL(srcA_SEL), .MUX_0(rs1), .MUX_1(U_Type), .MUX_2(~rs1), .MUX_out(ALU_srcA));
Mux_3x8 Mux_ALU_B (.SEL(srcB_SEL), .MUX_0(rs2), .MUX_1(I_Type), .MUX_2(S_Type),
         .MUX_3(PC_count), .MUX_out(ALU_srcB));
ALU ALU (.ALU_srcA(ALU_srcA), .ALU_srcB(ALU_srcB), .ALU_FUN(ALU_FUN), .result(result));
Branch_Cond_Gen Branch_Cond_Gen (.rs1(rs1), .rs2(rs2), .br_eq(br_eq), .br_lt(br_lt), .br_ltu(br_ltu));
CU_DCDR CU_DCDR (.ir(ir), .br_eq(br_eq), .br_lt(br_lt), .br_ltu(br_ltu), .ALU_FUN(ALU_FUN),
         .srcA_SEL(srcA_SEL), .srcB_SEL(srcB_SEL), .PC_SEL(PC_SEL), .RF_SEL(RF_SEL));
CU_FSM CU_FSM (.RST(RST), .clk(CLK), .ir(ir), .PC_WE(PC_WE), .RF_WE(RF_WE), .memWE2(memWE2),
         .memRDEN1(memRDEN1), .memRDEN2(memRDEN2), .reset(reset));

endmodule
```

OTTER MCU Test Bench:

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////
// Company: Cal Poly SLO
// Engineer: Wyatt Tack
//
// Create Date: 02/07/2024
// Design Name: OTTER MCU Test Bench
// Project Name: OTTER MCU
// Target Devices: Basys 3 Board
// Description: OTTER Microprocessor simulation source
//
//
////////////////////////////////////////////////////////////

module OTTER_MCU_TB();
logic RST, CLK;
//logic INTR,
logic [31:0] IOBUS_IN;
logic IOBUS_WR;
logic [31:0] IOBUS_OUT, IOBUS_ADDR;

OTTER_MCU UUT (.RST(RST), .CLK(CLK), .IOBUS_IN(IOBUS_IN),
    .IOBUS_WR(IOBUS_WR), .IOBUS_OUT(IOBUS_OUT),
    .IOBUS_ADDR(IOBUS_ADDR));
always begin
#5
CLK = 0;
#5
CLK = 1;
end
always begin
#10
IOBUS_IN = 0;
RST = 1;
#10;
RST = 0;
#180;
end
endmodule
```