CS 223

Digital Design

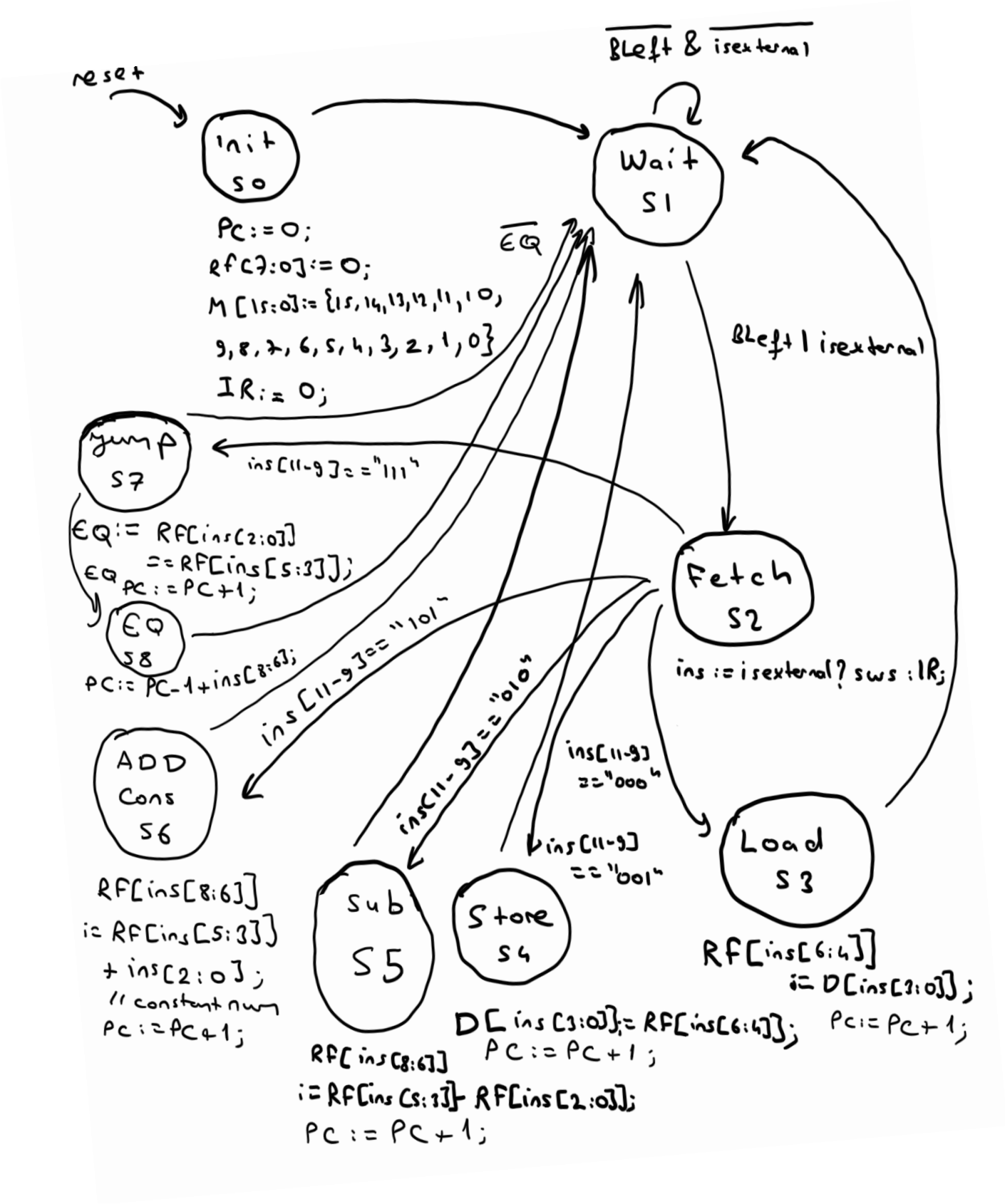
Section 5 Lab Project

Ömer Oktay Gültekin

21901413

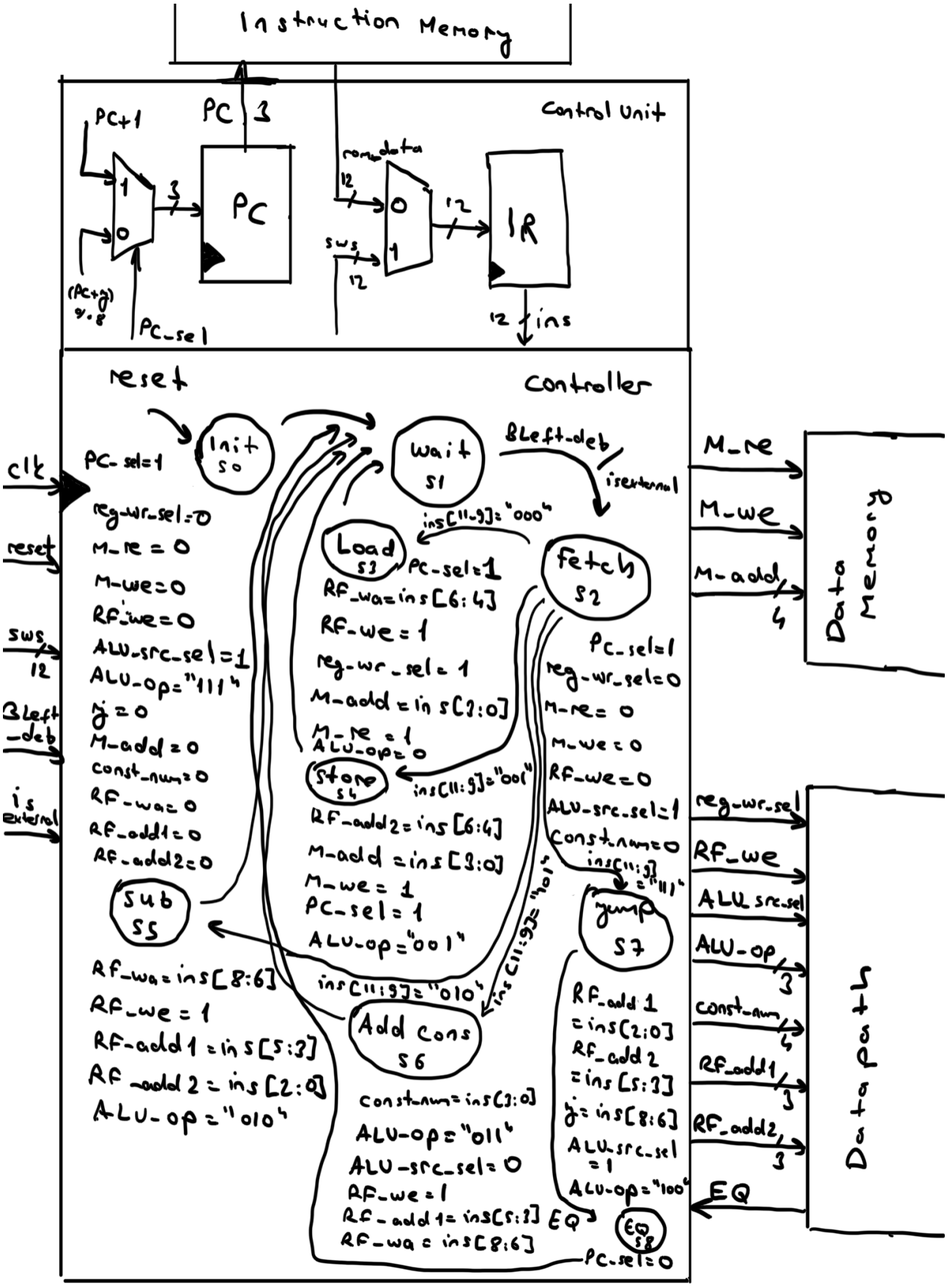
27/12/2021

Part B) Controller High-Level State Machine Diagram

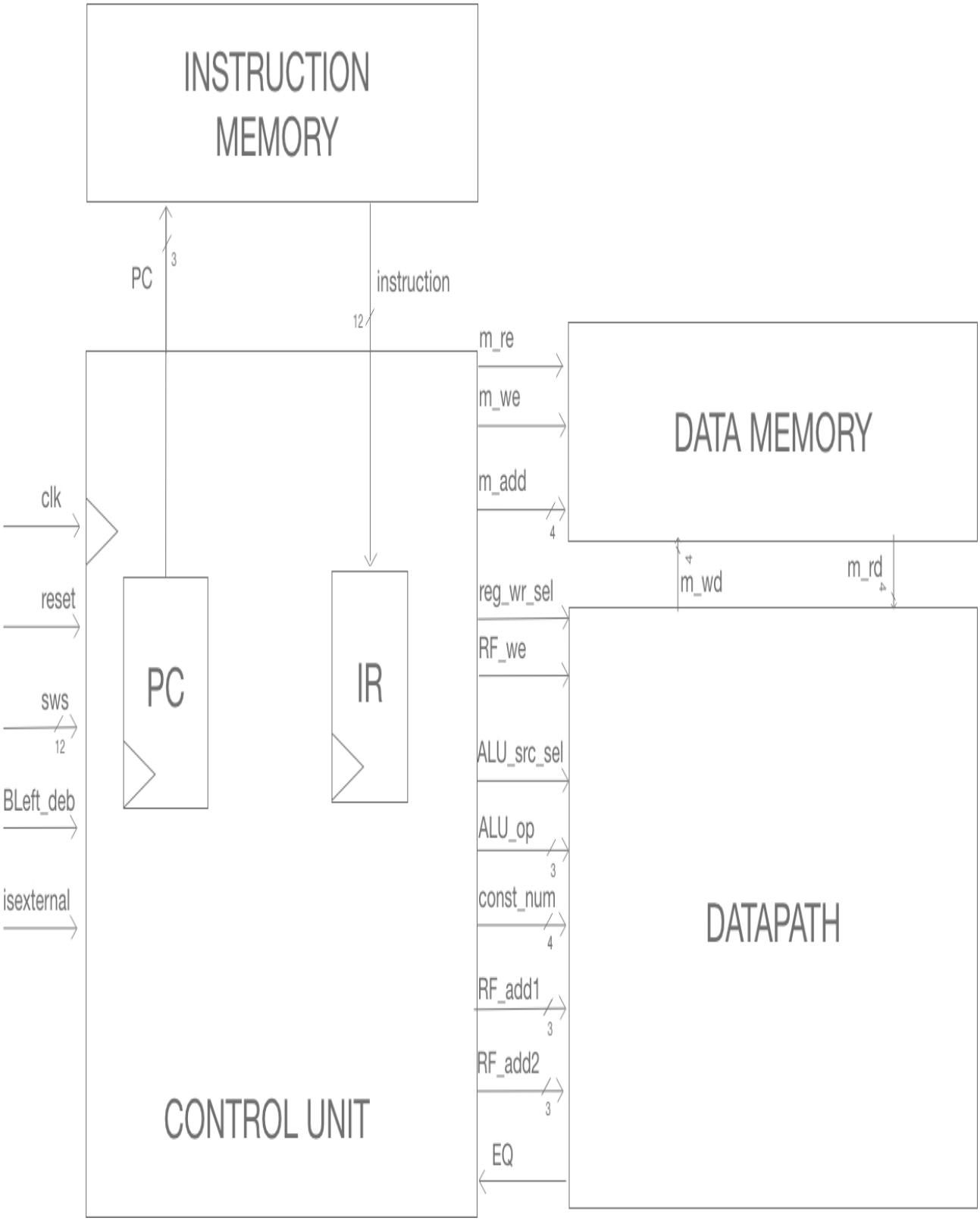


Note: More detailed diagram is in part C

Part C) Controller Block Diagram



PART D) Controller/Datapath Top Module Block Diagram



Part E) Optional Test Bench

```
module testbench();

    logic reset, isexternal, EQ;

    logic [11:0] sws;

    logic reg_wr_sel, M_re, M_we, RF_we, ALU_src_sel;

    logic [2:0] ALU_op;

    logic [3:0] M_add, const_num;

    logic [2:0] RF_wa, RF_add1, RF_add2;

    logic [3:0] RF_d1;

    logic [3:0] RF_d2;

    logic [3:0] RF_wd;

    logic [3:0] ALU_res;

    logic [3:0] M_wd;

    logic [3:0] M_rd;

    logic [2:0] PC;

    logic PC_sel;

    logic [11:0] rom_data;

    logic [11:0] instruction;

    logic [2:0] j;

    logic clk;

    logic BLeft_deb;


    PC pc(clk, reset, ~isexternal, BLeft_deb, PC_sel, j, PC);

    InstructionMemory rom(PC, rom_data);

    IR inst_reg(clk, reset, isexternal, BLeft_deb, rom_data, sws, instruction);

    Controller cont(clk, reset, BLeft_deb, isexternal, EQ, instruction, PC_sel, reg_wr_sel, M_re,
M_we, RF_we, ALU_src_sel,
```

```
    ALU_op, M_add, const_num, j, RF_wa, RF_add1, RF_add2);

    RegisterFile reg_file(RF_we, clk, reset, ALU_src_sel, RF_add1, RF_add2, RF_wa, RF_wd,
const_num, RF_d1, RF_d2, M_wd);

    ALU alu(RF_d1, RF_d2, ALU_op, EQ, ALU_res);

    DataMemory mem(M_add, M_wd, M_re, M_we, clk, reset, M_rd);

    Mux2to1For4Bit wr_sel_mux(M_rd,ALU_res, reg_wr_sel,RF_wd);


    initial begin

      clk = 1; #4; reset = 1;  isexternal = 0; sws = 0; #2;

      reset = 0; #4; BLeft_deb = 1; #4;

      BLeft_deb = 0; #12;

      BLeft_deb = 1; #4;

      BLeft_deb = 0; #24;

      sws = 12'b001_00_001_0101; #4;

      isexternal = 1; #8;

      isexternal = 0;

      BLeft_deb = 1; #4;

      BLeft_deb = 0; #12;

      BLeft_deb = 1; #4;

      BLeft_deb = 0; #12;

      BLeft_deb = 1; #4;

      BLeft_deb = 0; #12;

      BLeft_deb = 1; #4;

      BLeft_deb = 0; #12;

      BLeft_deb = 1; #4;

      BLeft_deb = 0; #12;

      BLeft_deb = 1; #4;
```

```
        BLeft_deb = 0; #12;
        BLeft_deb = 1; #4;
        BLeft_deb = 0; #12;
        BLeft_deb = 1; #4;
        BLeft_deb = 0; #12;
        BLeft_deb = 1; #4;
        BLeft_deb = 0; #12;
        reset = 1; #4;
        reset = 0;
        BLeft_deb = 1; #4;
        BLeft_deb = 0; #12;
        BLeft_deb = 1; #4;
      end
    always #2 clk <= ~clk;

endmodule
```