CS 223

DIGITAL DESIGN AND COMPUTER ARCHITECTURE

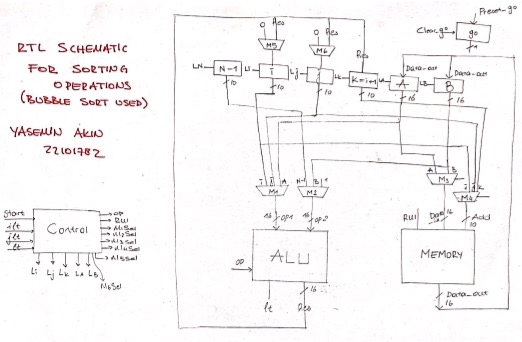
PROJECT REPORT

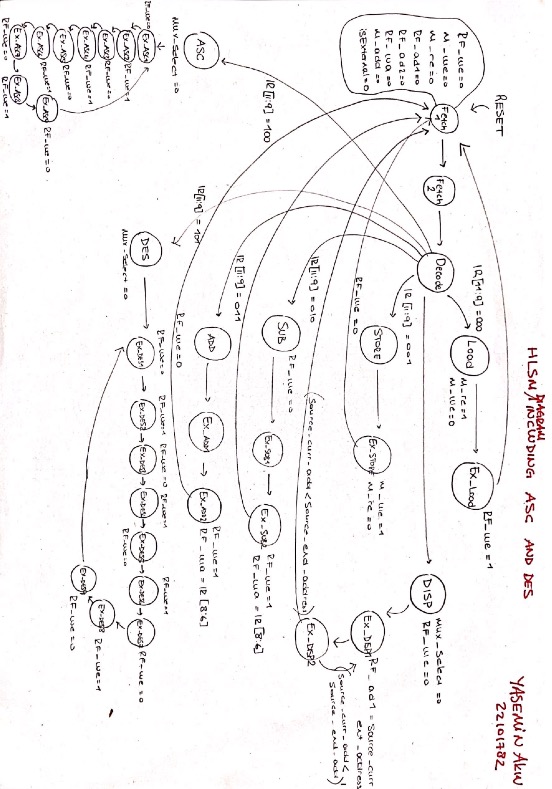
YASEMİN AKIN

22101782

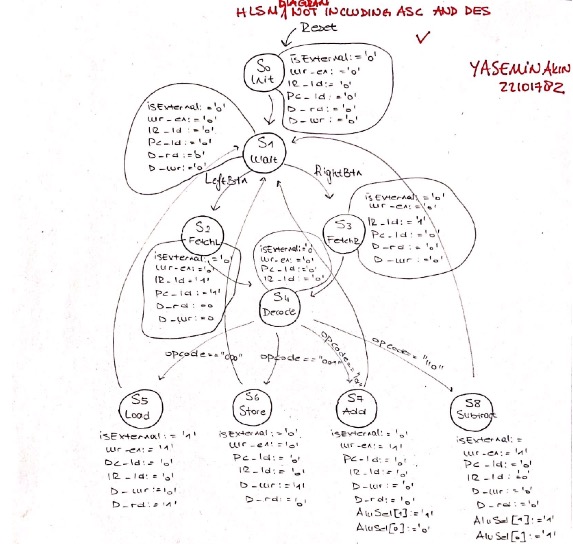
19 DECEMBER 2022

b) RTL schematics for Ascending and Descending Sorting operations



c) Controller High-Level State Machine Diagram

(including ascending and descending sorting operations 🡪 I could not implement this yet so I am putting an another HLSM diagram below, which does not include these operations and that I have implemented.)



d) Controller Block Diagram

Diagram, schematic

Description automatically generated

e) Controller/Datapath Top Module Block Diagram

Diagram, schematic

Description automatically generated

f) Testbenchs

//Testbenches

//control unit testbench:

`timescale 1ns/1ps

module control\_Unit\_Test();

logic clk, reset, leftBt, rightBt, switchEn; logic [11:0] switch;

logic [3:0] D\_addr;

logic D\_rd, D\_wr;

logic [2:0] RF\_addr1;

logic [2:0] RF\_addr2;

logic [2:0] RF\_waddr;

logic [1:0]ALUSel;

logic isExternal, wr\_en;

logic[11:0] IM\_ins ;

Control\_Unit dut(clk, reset, leftBtn, rightBtn, switchEn, switch, D\_addr,

D\_rd, D\_wr, RF\_addr1, RF\_addr2, RF\_waddr, ALUSel, isExternal, wr\_en, IM\_ins);

initial

clk = 1;

always begin

#15; clk = ~clk;

end

initial begin

#10; reset = 1;

#10; reset = 0;

#20; leftBtn = 1;

#30; leftBtn = 0;

#20; leftBtn = 1;

#50; leftBtn = 0;

#20; leftBtn = 1;

#30; leftBtn= 0;

#20; leftBtn = 1;

end

endmodule

//simple processor testbench:

`timescale 1ns/1ps

module processor\_Test();

logic clk, rightBtn, switchEn, leftBtn, reset; logic [11:0] switch;

logic [11:0]IM; logic [6:0] seg; logic [3:0] an;

logic [3:0] Rel; logic dp;

Simple\_Processor dut\_sp ( clk, reset, leftBtn, rightBtn, switchEn, switch,seg, dp, an, Rel, IM );

initial

clk = 1;

always begin

#1; clk = ~clk;

end

initial begin

#10; reset = 1;

#15; reset = 0;

#20; leftBtn = 1;

#25; leftBtn = 0;

#20; leftBtn = 1;

#30; leftBtn = 0;

#30; leftBtn = 1;

#40; leftBtn = 0;

#35; leftBtn = 1;

end

endmodule