CS 223 Digital Design

Section 5

Project Assignment

Name: Esad Ismail Tok

ID: 21801679

Date: 25.12.2020

Introduction

In this project assignment it is aimed to design a single cycle processor that will either fetch an instruction from the instruction memory or that will use the custom instructions decided by the switches.

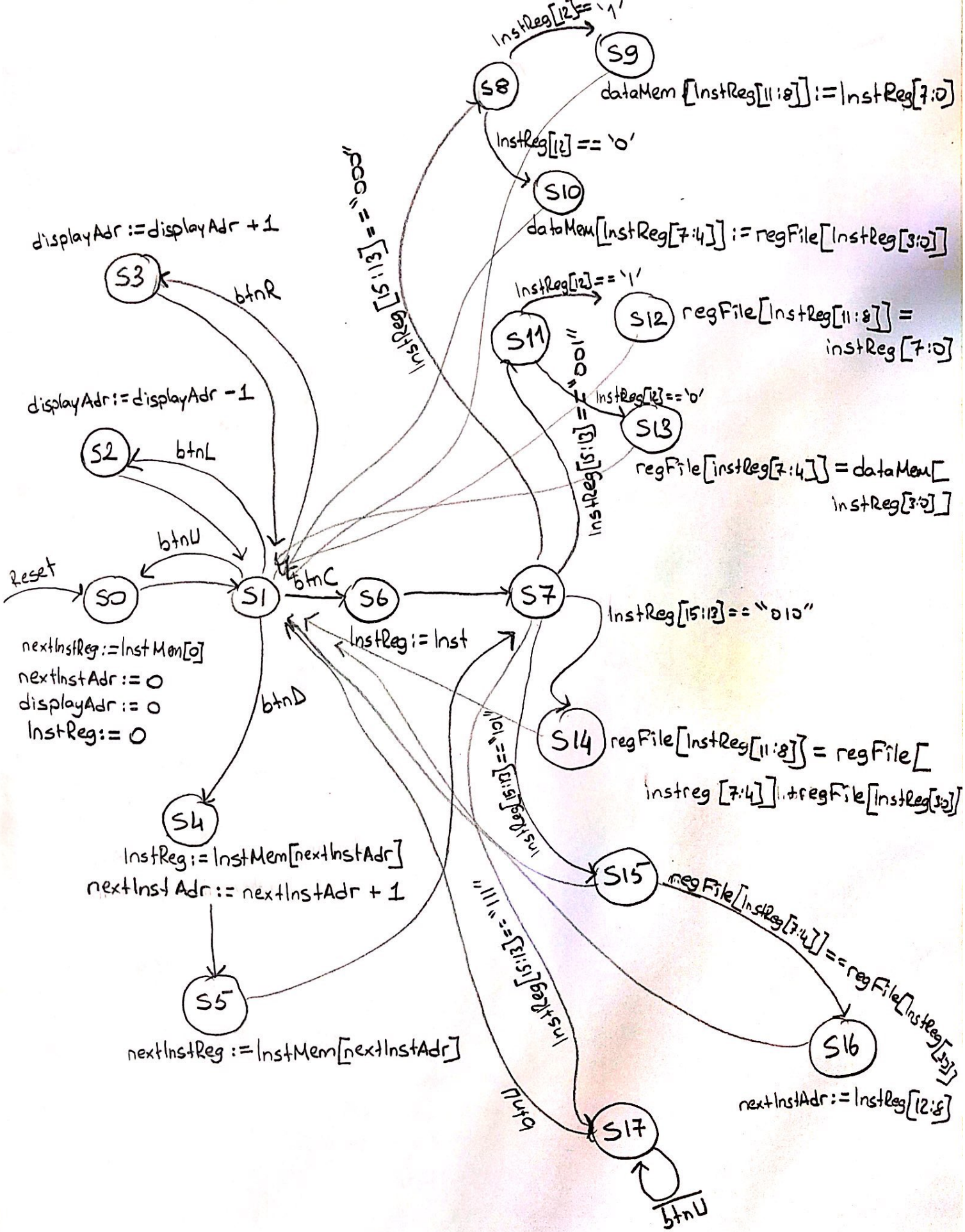
The single cycle processor includes three memory component. First one is the data memory which is 16x8 bits memory and has two read and one write ports. Second one is the register file which is again 16x8 bits memory and has two read and one write ports. The third and last memory component is the instruction memory which is a read only memory and it can store up to 32 instructions each 16 bits long.

According to the opcode of the instruction the single cycle processor decides which operation to perform and in each operation, the specific data addresses and input datas are provided by the instruction bits as well.

The detailed explanation of the HLSM is provided in the next sections.

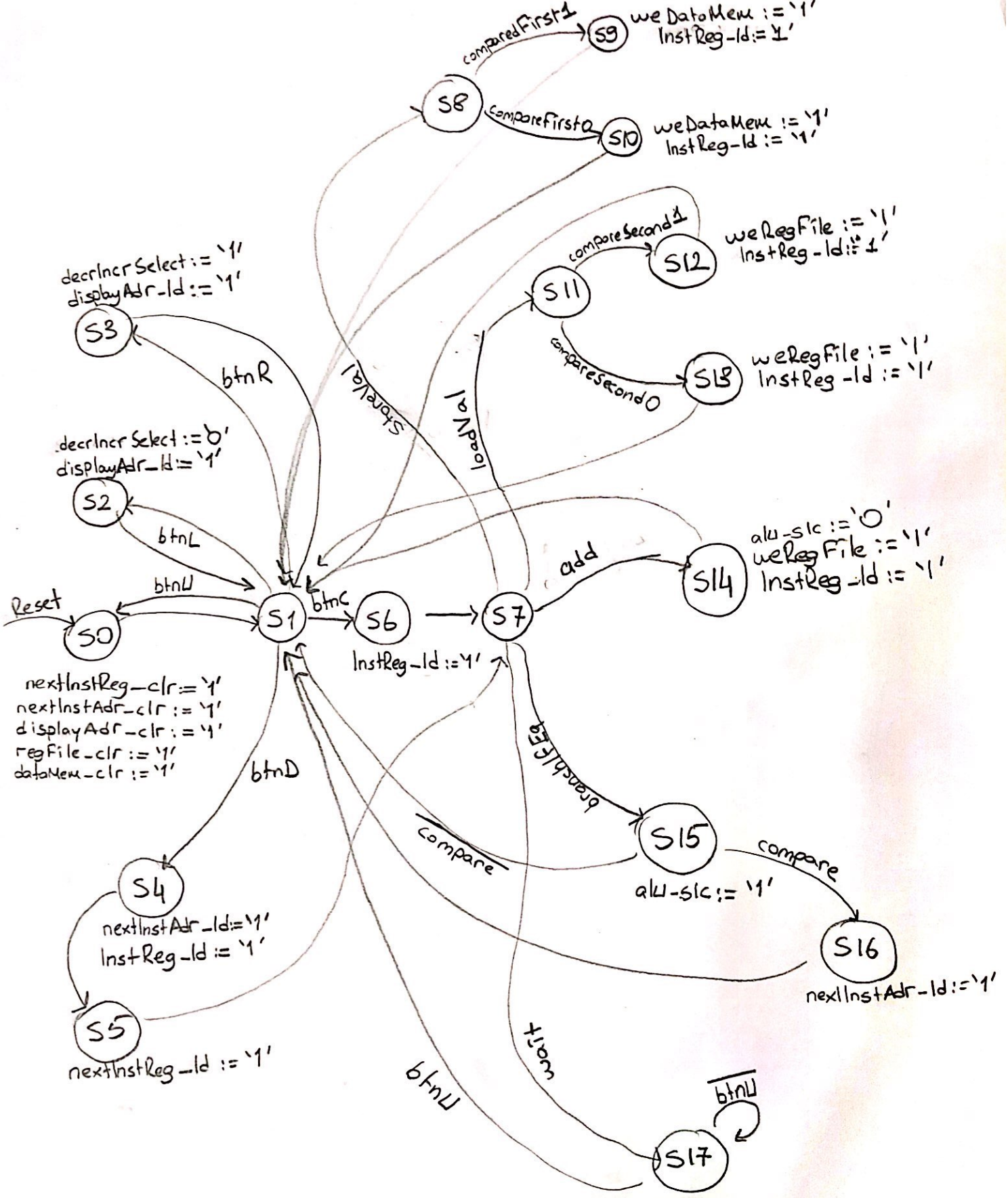
Detailed HLSM Diagram

The processor will start the transitions with the initial state which will assign all local storage components to have initially the zero value.
After the initialization transition to the wait state occurs. In this state the machine waits for a button to be pressed.
Pressing the left button lets the machine to display the previous data memory location and similarly pressing the right button lets the machine to display the next location.
If the down button is pressed the machine will fetch the instruction from the instruction memory and if the middle button is pressed the processor will take the instruction from the switches instead of the memory.
Pressing the up button will make the machine to go to the initial state which resets the local storages.
After the instruction is decided the state transitions to the next state according to the first three bits of the instruction namely opcode.
In the store value operation data is stored in the data memory by using the data from the instruction if the 12. bit of the instruction is 1, and data is taken from the register file if the 12. bit of the instruction is 0.
The reverse case is valid for loading data. Data is loaded in the register file using the data from the instruction if the 12. Bit of the instruction is 1, and data is taken from the data memory if the bit is 0.
In the addition operation the two data to be added are obtained from the register file and is written to another address of the register file.
In the branch if equals operation, if the two datas in the specified addresses of the register file are equal the instruction to be fetched from the instruction memory changes and takes the value of the data which is in the address specified by the jump address.
And as final operation if the stop execution is asserted the machine will wait until the up button namely reset button is pressed.
At the end of the each endmost state the machine will transitions to the state 1 which is the waiting state.
The detailed state transition diagram is provided below.



Detailed Controller’s FSM Diagram

Controller’s FSM has the same states and same transitions as the HLSM diagram. But all the multibit inputs and outputs are moved to the datapath so controller has the single bit control signals which constitute the interface between the controller and datapath.
In the initial state of the diagram all the clear signals of the local storages are asserted. So that every register has the value 0 initially. The button inputs are not changed since they are all single bit inputs.
In each state, whenever a local storage is changed in the HLSM diagram, the corresponding register’s load signal is asserted and the assignment Is handled using those control signals.
At the end of each endmost state, the machine transitions to the wait state to get another button signal.
The detailed state transition diagram of the controller’s FSM is provided below.



Detailed Controller Datapath Diagram

By moving all the multibit input outputs, local storages, arithmetic operations and multibit operations to the datapath the machine becomes considerably easy to control using only control signals that goes into the controller and comes from it.
Every local storage and memory component has the clear and load signals that will enable us to control those components and every multiplexer has the select input which is also controlled by the controller.
Three memory components which are the data memory, register file, and instruction memory are placed first into the datapath and then by placing the registers and arithmetic logic, datapath is constructed. Finally all control signals are placed so that it constitutes an interface between controller and datapath.
The detailed controller datapath block diagram is provided below.