Reza Jahani	Navid Dehban	Computer Architecture	Due Date: 15/10/1400
810198377	810198390	CA3 Report	Teacher: Dr Safari

In this project we aim to design and implement a Multi-Cycle Processor. This processor has three different types of instructions.

- 1. Data Processing Instructions
- 2. Data Transfer Instructions
- 3. Branch Instructions

Data Processing Instructions result in writing a data in one of the registers in register file, however two of the instructions only produce flags and no data is written in register file.

Data Transfer Instructions are divided into Load & Store. These instructions are seperated via a bit named L. Load instruction reads a data from the memory and writes it in register file while Store instruction saves a data which is available in register file in the memory.

Branch Instructions are used to control the flow of the program. This makes the program able to jump from an instruction to another.

• Datapath & Controller of the processor are in file 1-DP&CU.pdf

In figure 1, wired datapath is shown.

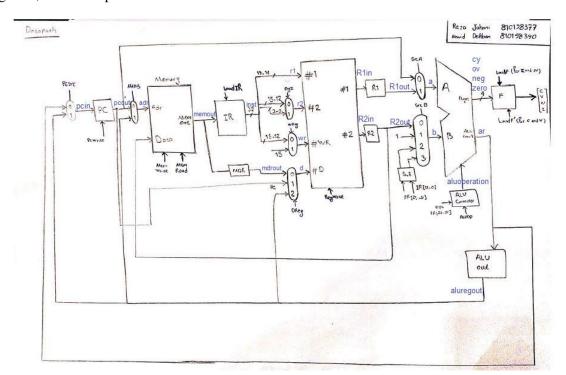


Figure 1: Wired Datapath

For testing the functionality of the processor, a program is designed to find the minimum of an array with length of 10.

The assembly code is presented in figure 2. Machine code is also presented in figure 3. These are the instructions written in memory.

```
R1, R0, 1000
    C=11 add
    C=11 add
                  R2, R0, 0
1
2
3
4
5
6
7
8
9
10
11
12
13
    C=11 load
                  R3, R1(0)
    C=11 add
                  R7, R0, 0
           add
                  R4, R0, 10
           cmp
                  --, R2, R4
   C=00
          В
                    8
           load
                  R5, R1(0)
                  --, R5, R3
          cmp
    C=01 B
          add
                  R3, R5, R0
          add
                  R7, R2, R0
          add
                  R1, R1, 1
                  R2, R2, 1
          add
14
                    -10
          В
15
          store R3, R0(2000)
16
          store R7, R0(2004)
```

Figure 2: Assembly Code

```
11000000100000000001001111101000
1100000010000000001000000000000
11010000000000010011000000000000
1100000010000000011100000000000
11000000100000000100000000001010
110000000110001000000000000000100
11010000000000010101000000000000
11000000011001010000000000000011
1100000000001010011000000000000
11000000000000100111000000000000
110000001000000100010000000000001
110000001000001000100000000000001
111010111111111111111111111110110
11010000000100000011011111010000
11010000000100000111011111010100
```

Figure 3: Machine Code

The array we use as the input to the program is as follows.

arr=[10,8,7,6,3,9,5,4,1,12]

Starting address of the array is 1000.

The minimum element in the array is stored in address of 2000 of the memory. In addition the index of the this element is saved in address 2004 of the memory.

The result can be observed in figure 4, presented in shape of wavefrom.

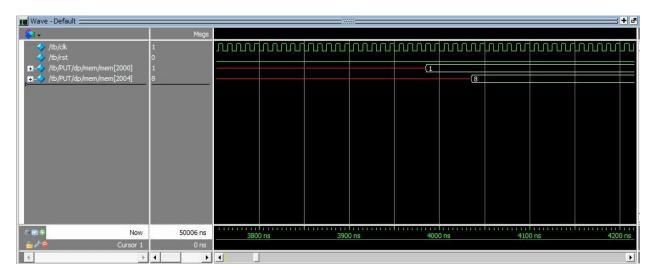


Figure 4: The Output Waveform.

As it can be observed in figure 4, the minimum element has been detected and wirtten in address 2000 of the memory (1) and the index of this element has been saved in address of 2004 of the memory (8).

- Modules used in datapath are in *DP_Modules.v*
- Modules used in controller are in CU_Modules.v
- Datapath is wired in *Datapath.v*
- Controller unit is wired in *Controller.v*
- The Processor is wired in *Processor.v*