Group 9 **ECE 366 – Fall 2018** 11/01/2018

**Project 3: ISA Simulation**

**Table of Contents**



[A) ISA Intro](#page2) [2](#page2)

[1. Introduction](#page2) [2](#page2)

[2. Instruction List](#page2) [2](#page2)

[3. Register Design](#page2) [2](#page2)

[4. Control Flow (Branches)](#page3) [3](#page3)

[5. Data addressing modes](#page3) [3](#page3)

[B) Answers to Questions](#page4) [4](#page4)

[C) ISA Package](#page4) [4](#page4)

[1. Algorithms (in assembly code) of the two programs](#page4) [4](#page4)

[2. Machine code for P1 and P2](#page11) [11](#page11)

[3. Pattern C and D](#page16) [16](#page16)

[4. Python Simulator Code](#page16) [16](#page16)

[5. Hardware Schematics](#page18) [18](#page18)

[a. ALU schematic](#page18) [18](#page18)

[b. CPU Datapath](#page18) [18](#page18)

[c. Control logic](#page19) [19](#page19)

1

|  |  |  |
| --- | --- | --- |
| Group 9 | **ECE 366 – Fall 2018** | 11/01/2018 |
|  | **Project 3: ISA Simulation** |  |

1. **ISA Intro**



1. Introduction

* **Name of Architecture:** HT
* **Overall Philosophy:**

The philosophy behind this architecture is to have an efficient and user-friendly ISA Design.

* **Goals:**

o Achieve Modular Exponentiation Program for any 16-bit P and Q

* 1. Achieve Best Match Count Program for any best-matching score

1. Instruction List

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Instruction** | **Format** | **Opcode** |  | **Example** |
|  |  |  |  | |
| Add | Add Rx, Ry | 000 xx yy | Rx = Rx + Ry | |
|  |  |  |  | |
| Sub | Sub Rx, Ry | 001 xx yy | Rx = Rx - Ry | |
|  |  |  |  | |
| Load | Load Rx, (Ry) | 010 xx yy | Rx <- M[Ry] | |
|  |  |  |  | |
| Store | Store Rx, (Ry) | 011 xx yy | M[Ry] <- Rx | |
|  |  |  |  | |
| Jump | Jump Rx | 100 00 xx | PC -= Rx | |
|  |  |  |  | |
| bezR1 | bezR1 Rx | 101 00 xx | If R1==0, PC += Rx | |
|  |  |  |  | |
| SltR1 | SltR1 Rx, Ry | 110 xx yy | if Rx < Ry, R1 = 1 | |
|  |  |  |  | |
| Init | Init Rx, imm | 111 0 xx i | Rx = imm | |
|  |  |  |  |  |
| ShiftL | ShiftL | 111 10 00 | R2 | << |
|  |  |  |  |  |
| ShiftR | ShiftR | 111 10 01 | R2 | >> |
|  |  |  |  | |
| AndR3 | AndR3 | 111 11 01 | R3 AND 1 | |
|  |  |  |  |  |
| XorR2R1 | XorR2R1 | 111 11 00 | R2 | = R2 XOR R1 |
|  |  |  |  |  |
|  |  |  |  |  |
| Finish | Finish | 111 1111 | Finish | |
|  |  |  |  | |

3. Register Design

**Number of registers:** 4 (R0 to R3)

R1 is for testing purposes when using bezR1 or sltR1.

2

|  |  |  |
| --- | --- | --- |
| Group 9 | **ECE 366 – Fall 2018** | 11/01/2018 |
|  | **Project 3: ISA Simulation** |  |

4. Control Flow (Branches)

* **Branches supported:**

o Jump: only goes backward, based on value in register Rx

* 1. bezR1: branch if R1 == 0, only goes forward, based on value in register Rx
* **Target address calculus:** PC = PC ± Rx
* **Maximum branch distance:**
  1. For Jump: Rx (bin) = value of Rx (only goes backward)
  2. For bezR1: Rx (bin) = value of Rx (only goes forward)
* **Examples:**

|  |  |  |
| --- | --- | --- |
| **Assembly Instruction** | **Machine Code** | **Function** |
|  |  |  |
| Jump R2 | 1100 0010 | PC = PC – R2 |
|  |  |  |
| bezR1 R0 | 0101 0000 | If R1==0: PC += R0 |
| Else: PC++ |
|  |  |
|  |  |  |

5. Data addressing modes

What addressing modes are supported for data memory? How are the addresses calculated? Give examples of your assembly load / store instructions and their corresponding machine code.

* **Addressing Modes:** M[Rx] with Rx being a register with a 16-bit value
* **Address Calculation:** M[0] is the 16-bit word in address 0 of the data memory
* **Examples:**

|  |  |  |
| --- | --- | --- |
| **Assembly Instruction** | **Machine Code** | **Function** |
|  |  |  |
| Load R0, (R1) | 0010 0001 | R0 <- M[R1] |
|  |  |  |
| Store R2, (R3) | 1011 1011 | M[R3] <- R2 |
|  |  |  |

3

|  |  |  |
| --- | --- | --- |
| Group 9 | **ECE 366 – Fall 2018** | 11/01/2018 |
|  | **Project 3: ISA Simulation** |  |

* 1. **Answers to Questions**



1. **What are the most significant advantages of your ISA (with regard to the two programs, hardware implementation, ease of programming, etc)? What are the main limitations? What are the main compromises that you have done to make things work, rather than perfecting everything?**

The most significant advantages are that our ISA is that it takes full advantage of the limited hardware. The main limitations are the 7 bits we have to use for the instructions. We had to store and load values in memory because of the limited bits we had for the immediate.

1. **What have you done towards the goals of low DIC and HW simplification? What could have been done differently to better optimize for each of the two goals, if to start over?**

We had to optimize the DIC by loading and storing values in memory to make the computation faster and more efficient. If we were to start over, we would have to optimize our ISA even more to take advantage of shortcuts and less instructions.

1. **Reflect on this project(1-3) experience:**
2. **What did you learn from this project? What was the best / worst thing about it?**

Making an ISA with limited hardware is very challenging. It was cool to see how complex these systems are, but this was very demanding and time consuming.

1. **What advice would you give to someone taking this project in a future semester?**

Prepare to spend a lot of time on this project. Don’t be afraid to start over to make your ISA faster and easier to work with.

1. **How would you describe the value of this project experience in a job interview?**

This was a very interesting and demanding project that should reflect well on job interviews.

1. **Simulation Results**



1. Algorithms (in assembly code) of the two programs

4

|  |  |  |  |
| --- | --- | --- | --- |
| Group 9 | | **ECE 366 – Fall 2018** | 11/01/2018 |
|  |  | **Project 3: ISA Simulation** |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  | |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  | |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  | | |
|  | | |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  | |
|  |  |  |  |
|  |  |  | |
|  |  | 5 |  |

|  |  |  |
| --- | --- | --- |
| Group 9 | **ECE 366 – Fall 2018** | 11/01/2018 |
|  | **Project 3: ISA Simulation** |  |

6

|  |  |  |
| --- | --- | --- |
| Group 9 | **ECE 366 – Fall 2018** | 11/01/2018 |
|  | **Project 3: ISA Simulation** |  |
|  | 7 |  |

|  |  |  |
| --- | --- | --- |
| Group 9 | **ECE 366 – Fall 2018** | 11/01/2018 |
|  | **Project 3: ISA Simulation** |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | 8 |  |

|  |  |  |
| --- | --- | --- |
| Group 9 | **ECE 366 – Fall 2018** | 11/01/2018 |
|  | **Project 3: ISA Simulation** |  |
|  |  |  |
|  |  |  |
|  |  | |
|  | |  |
|  |  | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  | |
|  |  |  |
|  |  |  |
|  |  | |
|  |  |  |
|  |  | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | 9 |  |

|  |  |  |
| --- | --- | --- |
| Group 9 | **ECE 366 – Fall 2018** | 11/01/2018 |
|  | **Project 3: ISA Simulation** |  |

10

|  |  |  |
| --- | --- | --- |
| Group 9 | **ECE 366 – Fall 2018** | 11/01/2018 |
|  | **Project 3: ISA Simulation** |  |

11

Group 9 **ECE 366 – Fall 2018** 11/01/2018

**Project 3: ISA Simulation**

12

Group 9 **ECE 366 – Fall 2018** 11/01/2018

**Project 3: ISA Simulation**

13

Group 9 **ECE 366 – Fall 2018** 11/01/2018

**Project 3: ISA Simulation**

14

Group 9 **ECE 366 – Fall 2018** 11/01/2018

**Project 3: ISA Simulation**

15

|  |  |  |
| --- | --- | --- |
| Group 9 | **ECE 366 – Fall 2018** | 11/01/2018 |
|  | **Project 3: ISA Simulation** |  |

16

Group 9 **ECE 366 – Fall 2018** 11/01/2018

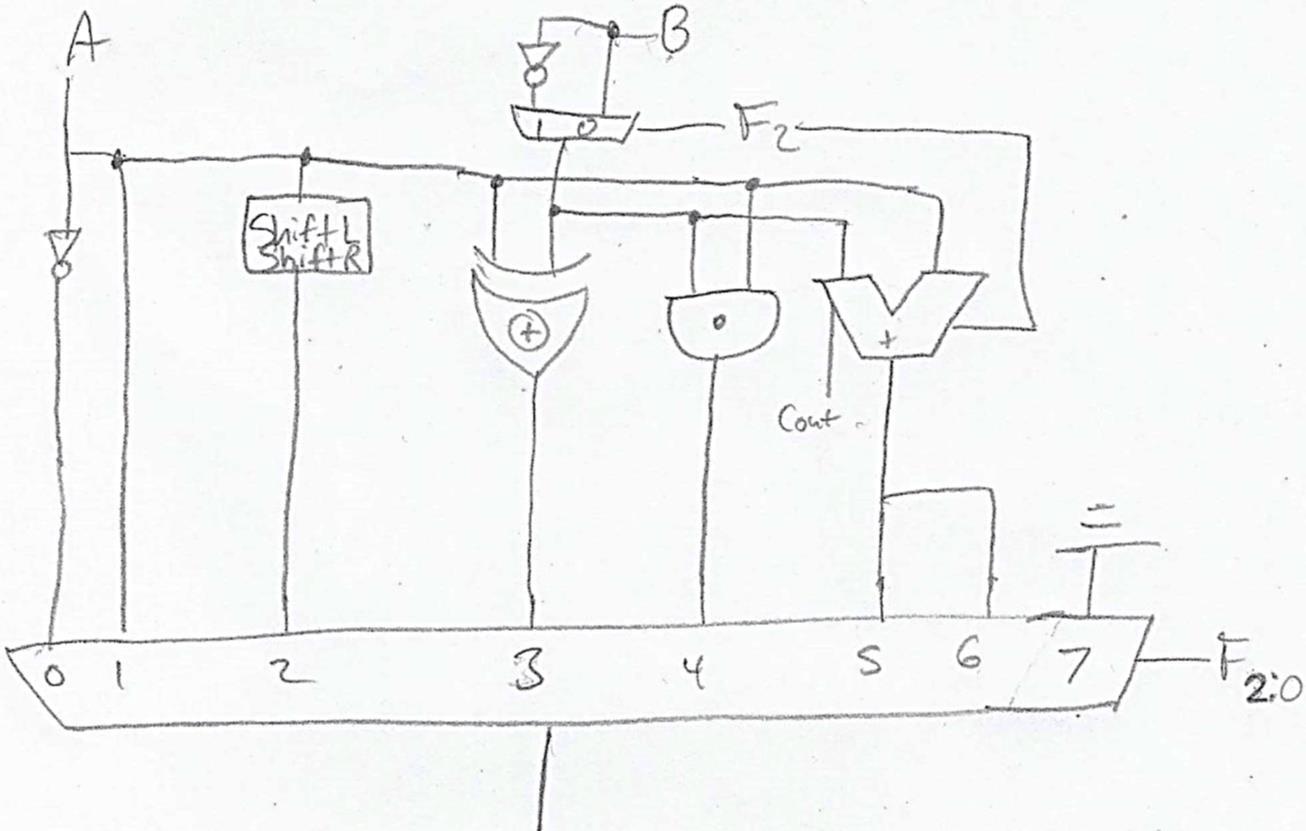
**Project 3: ISA Simulation**

17

|  |  |  |
| --- | --- | --- |
| Group 9 | **ECE 366 – Fall 2018** | 11/01/2018 |
|  | **Project 3: ISA Simulation** |  |

5. Hardware Schematics

**a. ALU schematic**



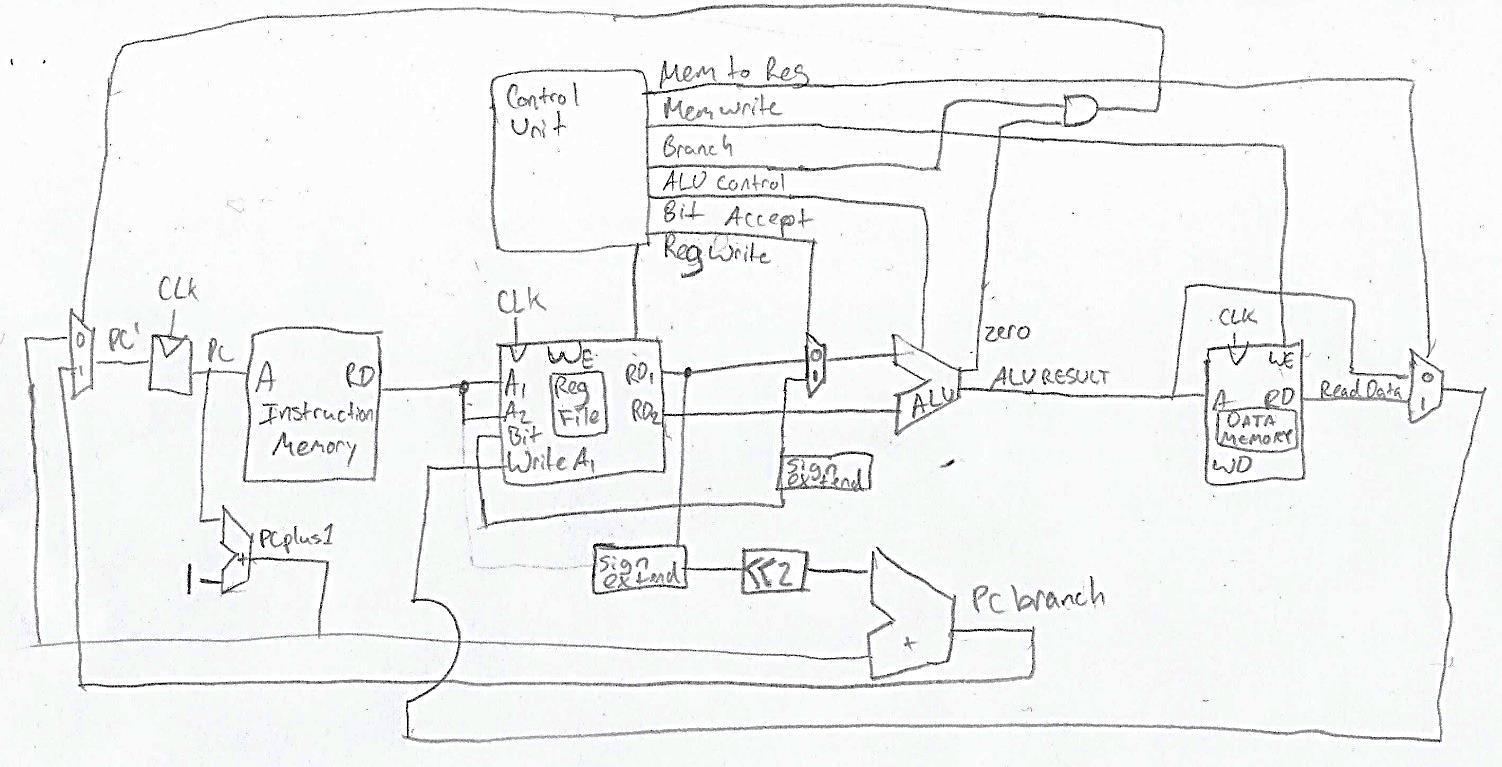
If F2:0 picks:

* 000 we get “Init A,0”
* 001 we get “Init A,1”
* 010 we get “ShiftL” or “ShiftR”
* 011 we get “XorR2R1”
* 100 we get “AndR3”
* 101 we get “Sub” or “Add”
* 110 we get “sltR1”
* 111 is grounded so it cannot be picked from the MUX

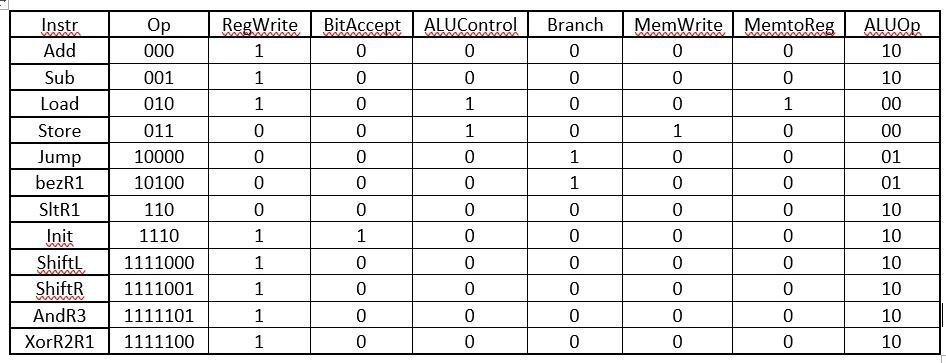
18

|  |  |  |
| --- | --- | --- |
| Group 9 | **ECE 366 – Fall 2018** | 11/01/2018 |
|  | **Project 3: ISA Simulation** |  |

**b. CPU Datapath**



**c. Control logic**



19