|  |  |
| --- | --- |
| **Author:** | Glenda Salas |
| **Student ID:** | \*20295844 |
| **E-Mail:** | [gsalas1@uco.edu](mailto:gsalas1@uco.edu) |
| **Course:** | CMSC 3833 – Computer Organization and Architecture II |
| **Assignment:** | P3 |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Scoring block** | | | |
| **Problem** | **Maximum** | **Earned** | **Explanation** |
| **Q1** | **10** |  |  |
| **Q2** | **10** |  |  |
| **Q3** | **10** |  |  |
| **Q4** | **15** |  |  |
| **Q5** | **15** |  |  |
| **Total** | **60** |  |  |

Note: -5 points deduction on an incorrect submission format.

Make sure all code is executed properly in the Java simulator, which will be used for grading.

Marie.js.org and Java simulator’s results may **not** be the same.

How to take a screenshot and make a PDF:

* Windows
  + <https://www.cnet.com/how-to/7-ways-you-can-take-screenshots-in-windows-10/>
  + <https://www.makeuseof.com/tag/use-trick-convert-image-pdf-windows-10/>
* Mac
  + <https://support.apple.com/en-us/HT201361>
  + <https://support.apple.com/guide/preview/convert-graphics-file-types-prvw1012/mac>

**Project 3: Extending Assembler and Simulator with new instructions**

**Q1. Changing the instruction semantic (10 pts)**

We will make a modified MARIE architecture called MARIE X1.

In this new architecture, we will change the semantic of one existing instruction and use one unused instruction number.

We will change the semantics of the “Clear” instruction (instruction opcode is 10 = 0xA).

Please make sure now the use of this instruction is illegal by changing the assembler.

Any usage in the code should be reported as an error by modifying the assembler.

Just in case you used this instruction in your program, please modified the code to equivalent code as follows.

* Change any usage of “Clear” to a combination of “Load” and a declaration of a variable having the value zero.

From:

Clear

Should be changed to

To:

Load XXXZERO // XXX in XXXZERO is just to avoid conflict with any existing ZERO variable for other purposes.

XXXZERO, HEX 0 // After declaring a variable XXXZERO.

Test code:

Clear

(Your modified assembler should show an error)

Please include the screenshot of the output of your modified assembler for the test code.

QA: The source code of the assembler to edit is Assembler.java

Q: How to test the assembler?

A: Click the File menu and then the Edit menu. The assembler does not run independently, but you can easily make it to run independently by creating a dummy test main file

A picture containing graphical user interface

Description automatically generated

**Q2. Extension of Assembler and Simulator for the MULT instruction (10 pts)**

Now we would like to replace the “clear” instruction with a new multiplication instruction.

Opcode number: 10 (0xA = 1010\_2)

Instruction: MULT X

RTN

MAR <- X

MBR <- M[MAR]

AC <- AC \* MBR

Here X is a memory parameter based on direct addressing.

Extend Assembler.java to support MULT instructions.

Extend MarieSim.java to support MULT instructions.

Test code:

Load X

MULT Y

Output

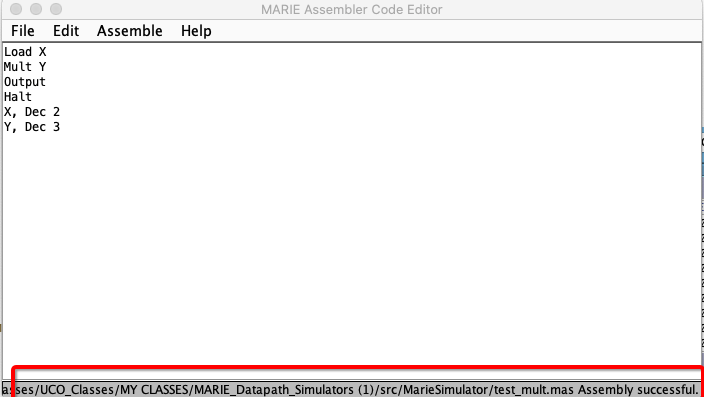
HALT

X, Dec 2

Y, Dec 3

(You should see 6 in decimal)

Example of screenshot of the assembler



Please include the screenshot of the output of your modified assembler for the test code.

Please include the screenshot of the output of your modified simulator for the test code.

**Q3. Extension of Assembler and Simulator for the DIV instruction (10 pts)**

The original MARIE used only 15 instructions. Now we would like to add a new division instruction.

Opcode number: 15 (0xF = 1111\_2)

Instruction: DIV X

RTN

MAR <- X

MBR <- M[MAR]

AC <- AC / MBR

Here X is a memory parameter based on direct addressing.

We assume the divider is a positive integer number. (If it is a non-positive number, please generate an assembler error)

Extend Assembler.java to support DIV instructions.

Extend MarieSim.java to support DIV instructions

Test code:

Load X

DIV Y

Output

HALT

X, Dec 6

Y, Dec 2

(You should see 3 in decimal)

Please include the screenshot of the output of your modified assembler for the test code.

Please include the screenshot of the output of your modified simulator for the test code.

**Q4. Exponent (Iterative algorithm) (15 pts)**

* Get a user input of a positive integer number: A
* Get a user input of zero or a positive integer number: B
* Calculate C is an exponent of A to the power of B (C=AB)
* Print C

Test cases:

|  |  |  |
| --- | --- | --- |
| Input A | Input B | Output C |
| 2 | 0 | 1 |
| 3 | 2 | 9 |

This program should use your MULT instructions.

In Question 4, please submit the following items.

Write the algorithm in a flow chart. The arithmetic operations are limited to only addition, subtraction, and multiplication. You can use chart software such as Lucidchart, powerpoint, or Visio. Please export it to a pdf file. (e.g.., Q4\_flowchart.PNG/PDF)

Implement the algorithm in C. The arithmetic operations can additionally include multiplication and division.

A screenshot of the final execution states for a "test case" of your C code. Note your program needs to be general to take any input values using input instructions. However, a test input is used just to create a screenshot. (e.g., Q4.PNG/JPG)

Implement the algorithm for MARIE architecture (e.g., Q4\_MARIE.mas)

A screenshot of the final execution states for a "test case" of your MARIE assembly code. Note your program needs to be general to take any input values using input instructions. However, a test input is used just to create a screenshot. (e.g., Q4\_MARIE\_screenshot.PNG/JPG)

Please submit the output of instruction count (Q4\_INSTRS.txt)

Please submit the following info using a text file (Q4\_PERF.TXT)

Count the number of instructions executed without the multiplication instructions as X1.

Count the number of instructions executed with the multiplication instructions as X2.

Calculate the instruction reduction rate R=X2/X1\*100

X1 = 40

X2 = 20

R= X2/X1\*100 = 50%

**Q5. Logarithm (Iterative algorithm) (15 pts)**

* Get a user input of a positive integer number: A
* Get a user input of zero or a positive integer number: B
* A is a base and B is an exponent of A
* Calculate C= Log AB
* Print C

Test cases

|  |  |  |
| --- | --- | --- |
| Input A | Input B | Output C |
| 2 | 16 | 4 |
| 3 | 9 | 2 |

In Question 5 please submit the following items.

Write the algorithm in a flow chart. The arithmetic operations are limited to only addition, subtraction, and division. You can use chart software such as Lucidchart, powerpoint, or Visio. Please export it to a pdf file. (e.g.., Q5\_flowchart.PNG/PDF)

Implement the algorithm in C. The arithmetic operations can additionally include multiplication and division.

A screenshot of the final execution states for a "test case" of your C code. Note your program needs to be general to take any input values using input instructions. However, a test input is used just to create a screenshot. (e.g., Q5.PNG/JPG)

Implement the algorithm for MARIE architecture (e.g., Q5\_MARIE.mas)

A screenshot of the final execution states for a "test case" of your MARIE assembly code. Note your program needs to be general to take any input values using input instructions. However, a test input is used just to create a screenshot. (e.g., Q5\_MARIE\_screenshot.PNG/JPG)

The numbers of instructions in Q5\_INSTRS.txt (Text document)

Please submit the following info using a text file (Q5\_PERF.TXT)

Count the number of instructions executed without the division instructions as X1.

Count the number of instructions executed with the division instructions as X2.

Calculate the instruction reduction rate R=X2/X1\*100

X1 = 40

X2 = 20

R= X2/X1\*100 = 50%

**Submission and Grading Criteria**

(If code doesn't compile or run, no point)

When your submission is decompressed, if the layout is not correct, there will be -10 pts.

Project\_3\_LastName\_FirstName

Q1\_2\_3/ (30 pts)

Assembler.java

MarieSim.java

Assembly\_screenshot\_Q1.PNG/JPG

Assembly\_screenshot\_Q2.PNG/JPG

Simulator\_screenshot\_Q2.PNG/JPG

Assembly\_screenshot\_Q3.PNG/JPG

Simulator\_screenshot\_Q3.PNG/JPG

Q4/ (15 pts)

Q4\_flowchart.png/pdf

Q4.c

Q4.PNG/JPG

Q4\_MARIE.mas

Q4\_MARIE\_screenshot.PNG/JPG

Q4\_INSTRS.txt

Q4\_PERF.txt

Q5/ (15 pts)

Q5\_flowchart.png/pdf

Q5.c

Q5.PNG/JPG

Q5\_MARIE.mas

Q5\_MARIE\_screenshot.PNG/JPG

Q5\_INSTRS.txt

Q5\_PERF.txt

These files are compressed as P3\_FirstName\_LastName.zip