1.23 Lab Assignment 0

Start Assignment

Due Feb 14 by 11:59pm **Points** 75 **Submitting** a file upload **File Types** txt
Available after Feb 1 at 10am

Purpose

In this assignment you are to use the website that is the home page of the MARS to learn about MARS and download it. You are also to become familiar with basic MIPS I architecture assembly language instructions, and run one assembly language program.

MARS emulates the MIPS I architecture. It is not a physical device that has all the physical components of the MIPS I. It is an IDE that uses the Java Virtual Machine to emulate how a physical implementation of the MIPS I would behave with a MIPS R2000 CPU.

You must have Java installed on your computer in order to be able to use MARS, because MARS is a Java program that uses the Java Virtual Machine to simulate the physical implementation of the MIPS 32 bit architecture.

Installing Java is outside the scope of instructions for this assignment. **Select This Text If You Wish To Go To An External Website And Read Instructions About How To Install Java and Perform The Installation Of Java (https://www.java.com/en/)**.

Select This Text To Go To An External Website That Has Instructions Of How To Use MARS As An IDE. (https://courses.missouristate.edu/KenVollmar/MARS/Help/MarsHelpIDE.html)

- Note: MARS also has a command line interface that is text based and runs in either a Windows Command Prompt (cmd) or in a Macintosh Terminal window. **Select This Text To Go To An External Website If You Wish To See How To Use MARS As A Command Line Program (https://courses.missouristate.edu/KenVollmar/MARS/Help/MarsHelpCommand.html)**.

Tasks

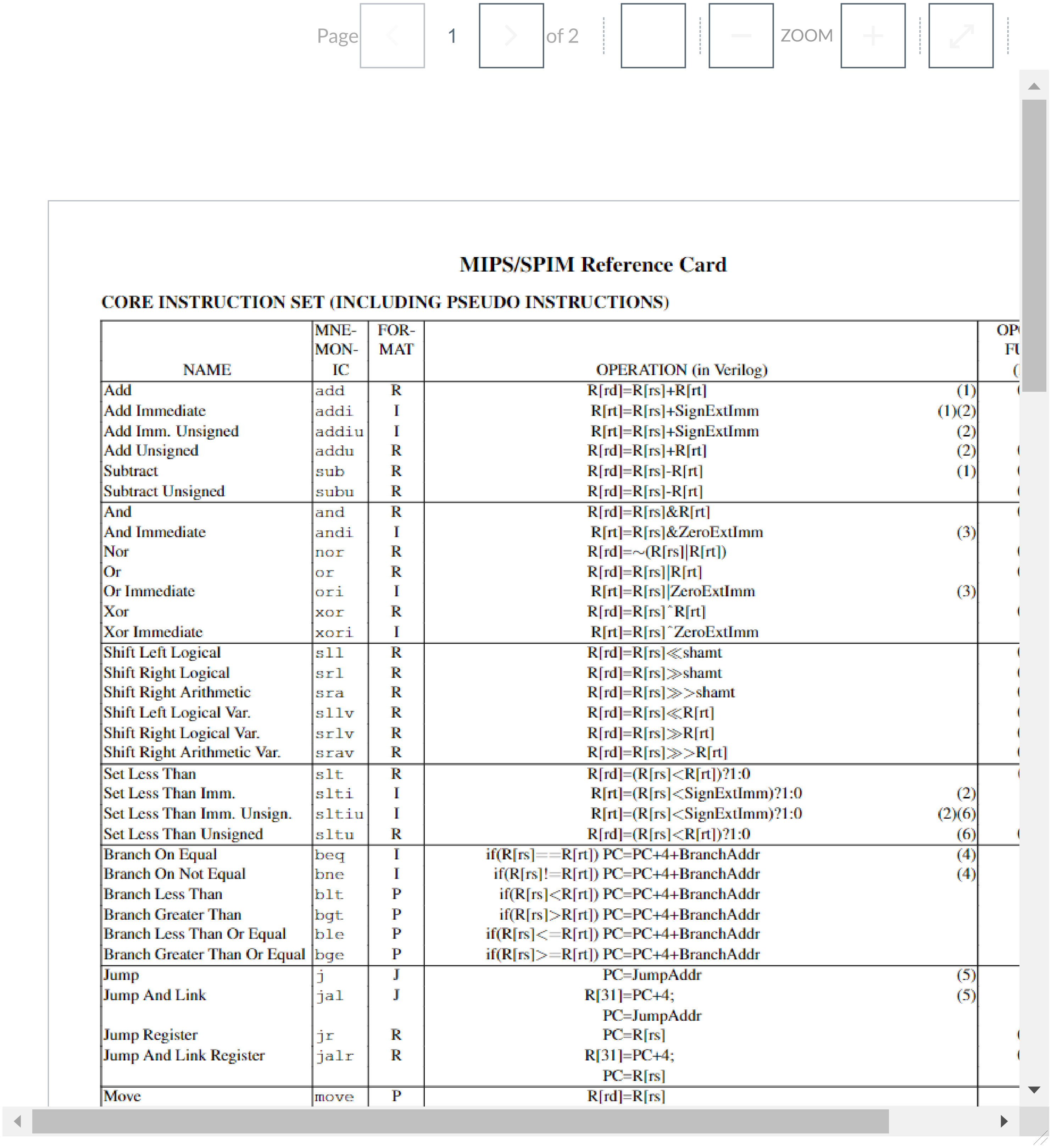
You must perform Tasks #1 - #4 given below:

Task #1 — **Select This Text To Go To The External MARS website (http://courses.missouristate.edu/KenVollmar/mars/)** . Read the website to learn about what MARS is and how it works. Once you have done that, look for the section of the website that provides information about how to download and run MARS. You will see there is more than one option to choose for running MARS. Choose the option that works for your particular computer and operating system.

*Note: **Task #1 is the primary task for this lab. Task #1 must be performed completely before moving on to the other tasks of this lab. It may take some trial and error to get the MARS to work.** Task #1 is meant to have you experience downloading a virtual computer architecture environment based and get it to work — something often required by assembly language programmers to perform programming.*

Task # 2 — **Select This To Read The Documentation. This documentation is meant for you to get a sense of the types of instructions, directives, registers, and other MIPS I architecture information (This is a PDF file)** **(https://ccsf.instructure.com/courses/47907/files/7300664/download?download_frd=1)**

Minimize File Preview



MIPS/SPIM Reference Card

CORE INSTRUCTION SET (INCLUDING PSEUDO INSTRUCTIONS)

NAME	MNE-MON-IC	FOR-MAT	OPERATION (in Verilog)		OP FU (
Add	`add`	R	R[rd]=R[rs]+R[rt]	(1)	(
Add Immediate	`addi`	I	R[rt]=R[rs]+SignExtImm	(1)(2)	
Add Imm. Unsigned	`addiu`	I	R[rt]=R[rs]+SignExtImm	(2)	
Add Unsigned	`addu`	R	R[rd]=R[rs]+R[rt]	(2)	(
Subtract	`sub`	R	R[rd]=R[rs]-R[rt]	(1)	(
Subtract Unsigned	`subu`	R	R[rd]=R[rs]-R[rt]		(
And	`and`	R	R[rd]=R[rs]&R[rt]		(
And Immediate	`andi`	I	R[rt]=R[rs]&ZeroExtImm	(3)	
Nor	`nor`	R	R[rd]=∼(R[rs]|R[rt])		(
Or	`or`	R	R[rd]=R[rs]|R[rt]		(
Or Immediate	`ori`	I	R[rt]=R[rs]|ZeroExtImm	(3)	
Xor	`xor`	R	R[rd]=R[rs]ˆR[rt]		(
Xor Immediate	`xori`	I	R[rt]=R[rs]ˆZeroExtImm		
Shift Left Logical	`sll`	R	R[rd]=R[rs]≪shamt		(
Shift Right Logical	`srl`	R	R[rd]=R[rs]≫shamt		(
Shift Right Arithmetic	`sra`	R	R[rd]=R[rs]≫>shamt		(
Shift Left Logical Var.	`sllv`	R	R[rd]=R[rs]≪R[rt]		(
Shift Right Logical Var.	`srlv`	R	R[rd]=R[rs]≫R[rt]		(
Shift Right Arithmetic Var.	`srav`	R	R[rd]=R[rs]≫>R[rt]		(
Set Less Than	`slt`	R	R[rd]=(R[rs]<R[rt])?1:0		(
Set Less Than Imm.	`slti`	I	R[rt]=(R[rs]<SignExtImm)?1:0	(2)	
Set Less Than Imm. Unsign.	`sltiu`	I	R[rt]=(R[rs]<SignExtImm)?1:0	(2)(6)	
Set Less Than Unsigned	`sltu`	R	R[rd]=(R[rs]<R[rt])?1:0	(6)	(
Branch On Equal	`beq`	I	if(R[rs]==R[rt]) PC=PC+4+BranchAddr	(4)	
Branch On Not Equal	`bne`	I	if(R[rs]!=R[rt]) PC=PC+4+BranchAddr	(4)	
Branch Less Than	`blt`	P	if(R[rs]<R[rt]) PC=PC+4+BranchAddr		
Branch Greater Than	`bgt`	P	if(R[rs]>R[rt]) PC=PC+4+BranchAddr		
Branch Less Than Or Equal	`ble`	P	if(R[rs]<=R[rt]) PC=PC+4+BranchAddr		
Branch Greater Than Or Equal	`bge`	P	if(R[rs]>=R[rt]) PC=PC+4+BranchAddr		
Jump	`j`	J	PC=JumpAddr	(5)	
Jump And Link	`jal`	J	R[31]=PC+4; PC=JumpAddr	(5)	
Jump Register	`jr`	R	PC=R[rs]		(
Jump And Link Register	`jalr`	R	R[31]=PC+4; PC=R[rs]		(
Move	`move`	P	R[rd]=R[rs]		

For any given computer architecture, you should be able to read documentation like this to get a sense of the physical components of the architecture, and how to use assembly language to write programs for the architecture. Though this class focuses on the MIPS I architecture, you should become used to reading documentation like this for any computer architecture that describes the architecture's physical components and the architecture's assembly language commands.

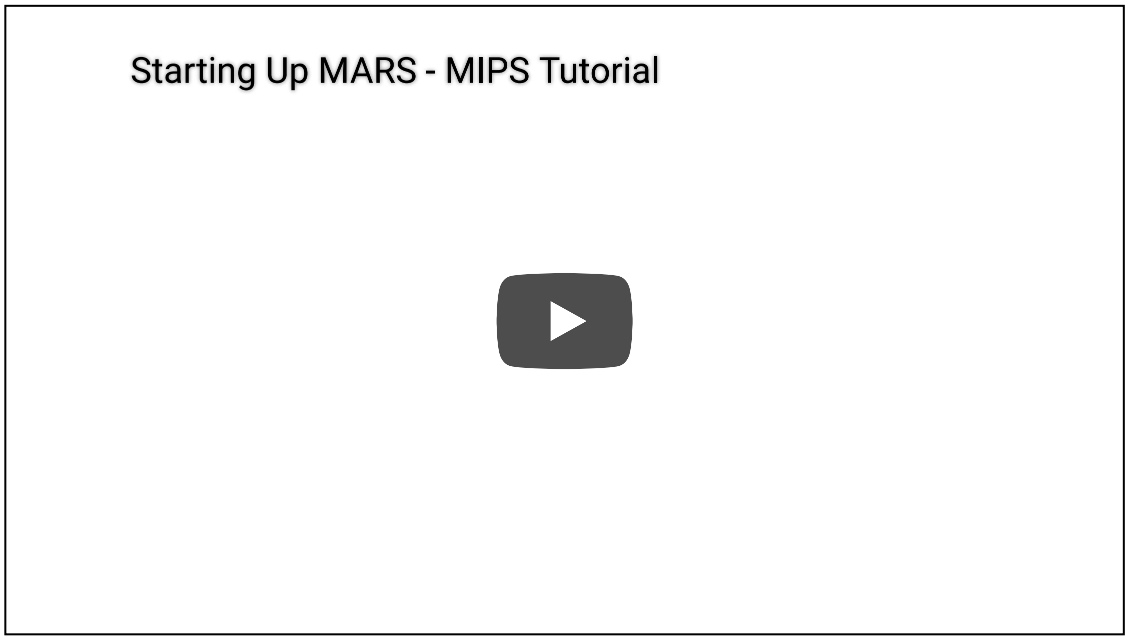
This is not a complete list of all the commands of the assembly language, but it will give you a feel for some of the most fundamental assembly language commends.

As you are reading about the commands, pay close attention to how the commands use certain operands. Be able to distinguish between operands that are literal values and operands that are registers.

Task # 3 — Watch A Tutorial Video Segment About MARS And Using MARS and run an assembly language program

Watch the following tutorial video segment:

Start At 1:37 End At 4:30



Open and use the following program in MARS:

task3MIPSassemblylanguageprogram.s (https://ccsf.instructure.com/courses/47907/files/7300660/download?download_frd=1)

Select This Text If You Wish To View task3MIPSassemblylanguageprogram.s As A Text File Here In Canvas. (https://ccsf.instructure.com/courses/47907/files/7300667/download?download_frd=1)

When using the program in MARS, you should save the file as a .s file because .s is the file extension for MIPS Assembly language files.

Write a 1 sentence description explaining what the program outputs when you execute the program in MARS. Save this description in a text file named task3desc.txt.

Task #4 -- Submitting Your Task #3 File To Earn Your Grade For This Lab Assignment

Submit the Task #3 file you created named task3desc.txt for this assignment.   Submit the file here in Canvas by selecting the Submit Assignment button here in this assignment.   Please submit task3desc.txt in text file format only.   To receive credit for this lab assignment, you must submit task3desc.txt in text file format.