HOMEWORK 2 MIPS

Due date:

Overview

This homework aims to help you get familiar with the MIPS instruction set. In this homework, we introduce the format of the MIPS instruction set architecture (ISA), the MIPS assembly language and a MIPS simulation tool. You need to use instructions listed below to implement two programs.

General rules

- You need to complete this homework INDIVIDUALLY. You can discuss the homework with other students, but you need to do the homework by yourself. You should not copy anything from someone else, and you should not distribute your homework to someone else. If you violate any of these rules, you will get NEGATIVE scores, or even fail this course directly
- When submitting your homework, compress all files into a single **zip** file, and upload the compressed file to Moodle.
  - Please follow the file hierarchy shown in Figure 1.

    F740XXXXX (your id) (folder)

    src (folder) * Store your source code

    report.docx (project report. The report template is already included. Follow the template to complete the report.)

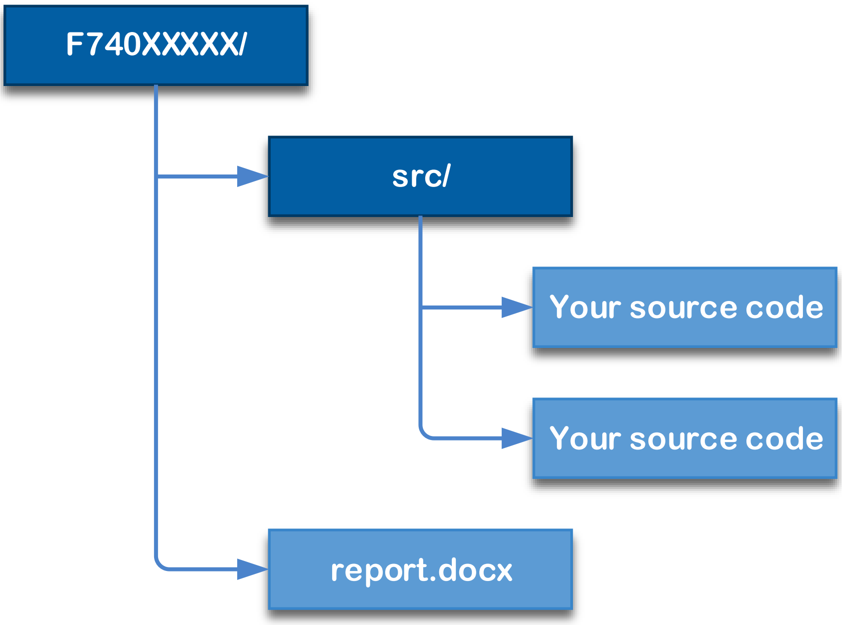


Figure 1. File hierarchy for homework submission

- **Important!** DO NOT submit your homework in the last minute. Late submission is not accepted.
- You should finish all the requirements (shown below) in this homework and Project report.
- If your code can not be recompiled by TA successfully using Mars, you will receive NO credit.

- **MIPS ISA**

The format of the MIPS instruction set architecture (ISA) used this homework is as follows. It contains three types: R type, I type, and J type. The details of each type are shown in the below.

R Type

Assembler Syntax

instruction	rd	rs	rt

Machine code Format

opcode	rs	rt	rd	shamt	funct
31 26	25 21	20 16	15 11	10 6	5 0

opcode	Mnemonics	SRC1	SRC2	DST	funct	Description
000000	nop	00000	00000	00000	000000	No operation
000000	add	$Rs	$Rt	$Rd	100000	Rd = Rs + Rt
000000	sub	$Rs	$Rt	$Rd	100010	Rd = Rs – Rt
000000	and	$Rs	$Rt	$Rd	100100	Rd = Rs & Rt
000000	or	$Rs	$Rt	$Rd	100101	Rd = Rs | Rt
000000	xor	$Rs	$Rt	$Rd	100110	Rd = Rs ^ Rt
000000	nor	$Rs	$Rt	$Rd	100111	Rd = ~(Rs | Rt)
000000	slt	$Rs	$Rt	$Rd	101010	Rd = (Rs < Rt)?1:0
000000	sll		$Rt	$Rd	000000	Rd = Rt << shamt
000000	srl		$Rt	$Rd	000010	Rd = Rt >> shamt
000000	jr	$Rs			001000	PC=Rs

I Type

Assembler Syntax

instruction	rt	rs	imm

Machine code Format

opcode	rs	rt	immediate
31 26	25 21	20 16	15 0

opcode	Mnemonics	SRC1	DST	SRC2	Description
001000	addi	$Rs	$Rt	imm	Rt = Rs + imm
001100	andi	$Rs	$Rt	imm	Rt = Rs & imm
001010	slti	$Rs	$Rt	imm	Rt = (Rs < imm) ? 1 : 0
000100	beq	$Rs	$Rt	imm	If(Rs == Rt) PC=PC+4+imm

000101	bne	$Rs	$Rt	imm	If(Rs != Rt) PC=PC+4+imm
100011	lw	$Rs	$Rt	imm	Rt = Mem[Rs + imm]
101011	sw	$Rs	$Rt	imm	Mem[Rs + imm] = Rt

J Type

Assembler Syntax

instruction	Target(label)

Machine code Format

opcode	address
31 26	25 0

opcode	Mnemonics	Address	Description
000010	j	jumpAddr	PC = jumpAddr
000011	jal	jumpAddr	R[31] = PC + 8 ; PC = jumpAddr

- **Program 1 (30%): Find the average of three numbers**

  Given three numbers, please implement a MIPS program to find the average of these numbers. Use MIPS Simulator (Mars) to run your assembly codes and store result into register $v0. **You should implement your division instructions and should not use division instructions not listed in this homework.**

- **Program 2(70%): Implement the Pascal’s formula(60%)**

  In mathematics, Pascal's rule (or Pascal's formula) is a combinatorial identity about binomial coefficients. It states that for positive natural numbers n and k,

  $\mathrm{C}(n,k) = \mathrm{C}(n-1,k-1) + \mathrm{C}(n-1,k)$ with $1 \le k \le n$

  where $\mathrm{C}(n,k)$ is the binomial coefficient of the x^k term in the expansion of $(1+x)^n$. Pascal's rule can also be generalized to apply to multinomial coefficients.

The pseudocode for the Pascal’s formula Pseudo code is as follows:

```
function pascal(n, m)
    if (m == n || m == 0)
        return 1
    else
        return pascal(n – 1, m - 1) + pascal(n – 1, m)
```

Your jobs are to implement Pascal's formula computation by using MIPS instructions listed in the table above. Use MIPS Simulator (Mars) to run your assembly code to compute pascal(10, 5) and store the result into register $v0.

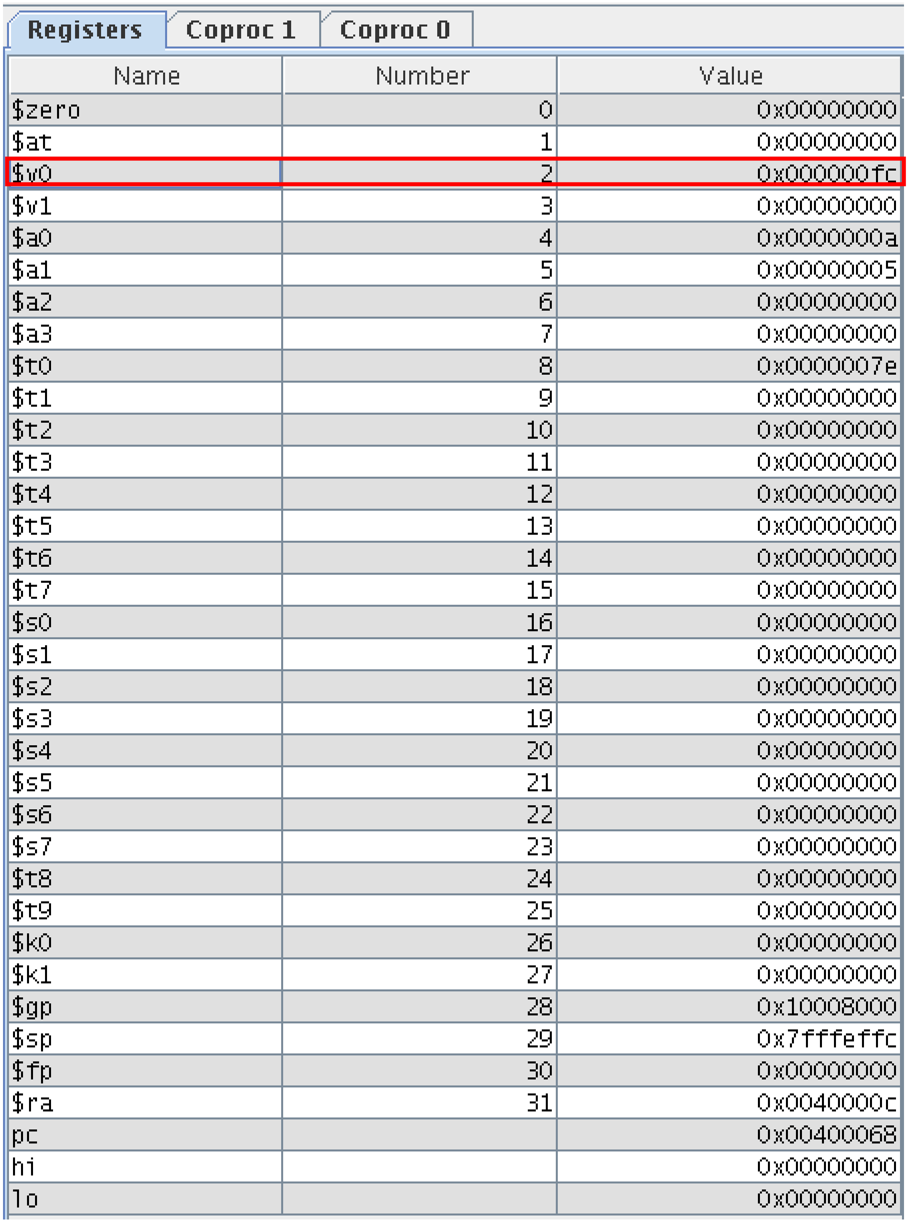
We have listed the instructions that are needed to implement the program for you. DO NOT using MIPS instructions not listed in this table.

Homework Requirements

1. **Save the first program in the q1.S**
2. **Save the second program in the q2.S.**
3. Finish your Project report

   Note: please take snapshot of your result and paste into your report.

   Example: Fig. 2



Name	Number	Value
$zero	0	0x00000000
$at	1	0x00000000
$v0	2	0x000000fc
$v1	3	0x00000000
$a0	4	0x0000000a
$a1	5	0x00000005
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x0000007e
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x0040000c
pc		0x00400068
hi		0x00000000
lo		0x00000000

Fig2. Snapshot of result

Important

When you upload your file, please check if you have done and followed all requirements, including **File hierarchy**, **Requirement file** and **Report format**.

If you have any questions, please contact us.