**Computer Organization 2020**

**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 Pascal’s formula computation.

**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.

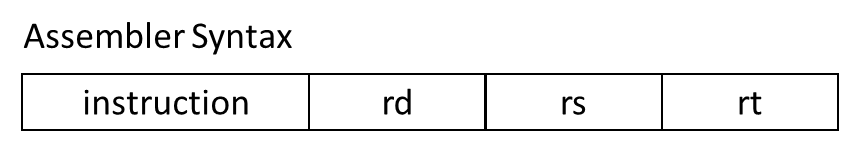
**Exercise**

Implement Pascal’s formula computation by using MIPS instructions listed in the table below.

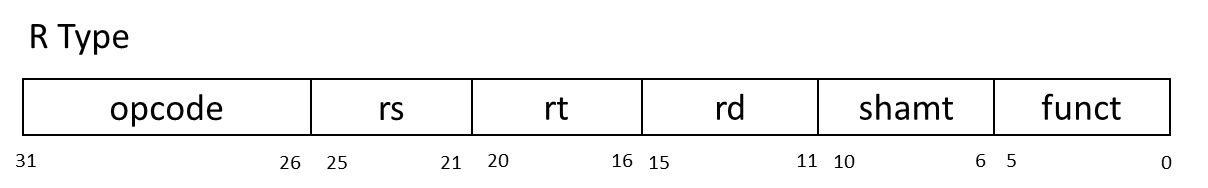
We list some basics instructions for you. DO NOT using MIPS instructions not listed in this table.

**MIPS ISA**

**R Type**

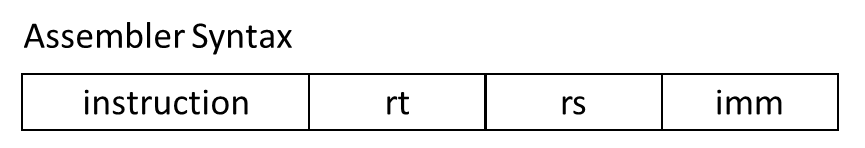
****

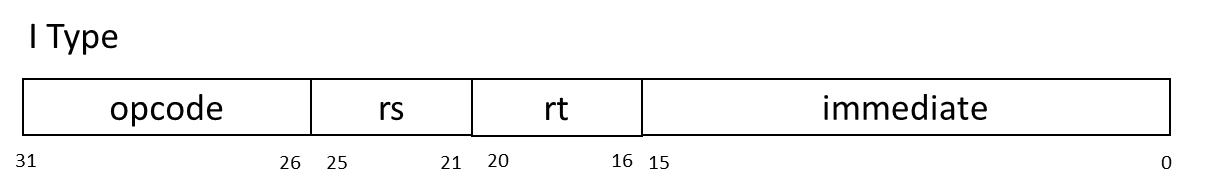
Machine code Format

****

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 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**

****

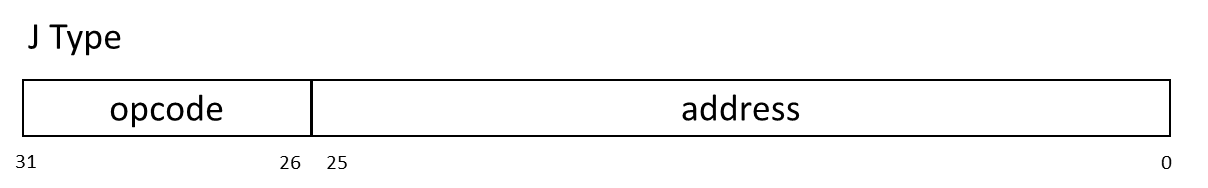
Machine code Format****

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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**

****

Machine code Format

****

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

**Pascal’s formula**

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,

with

where is the binomial coefficient of the term in the expansion of .

Pascal's rule can also be generalized to apply to multinomial coefficients.

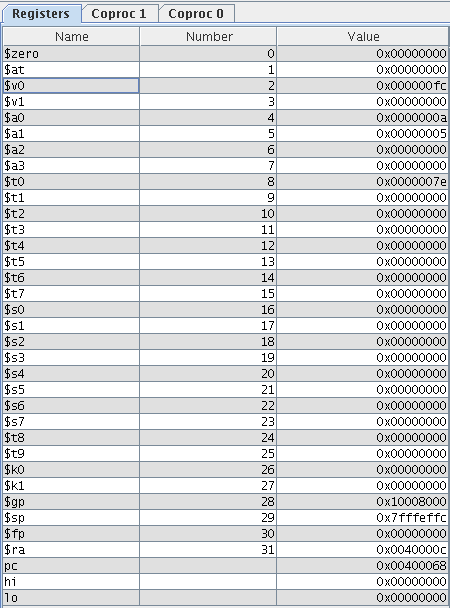
**Pascal’s formula Pseudo code**

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

**Homework Requirements**

1. Implement Pascal's formula computation according to the above MIPS instruction table.
2. Use MIPS Simulator (Mars) to run your assembly code to compute pascal(10, 5) and store result into register $v0.
3. Finish your Project report

Note: please take snapshot of your result and paste into your report. Example: Fig. 2



**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. |