**Computer Organization 2018**

**HOMEWORK I**

**Due day: 3/29 Due**

This homework is 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 in this homework to implement Fibonacci number computation.

**General rules for deliverables**

* 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**

**F740XXXXX\_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.

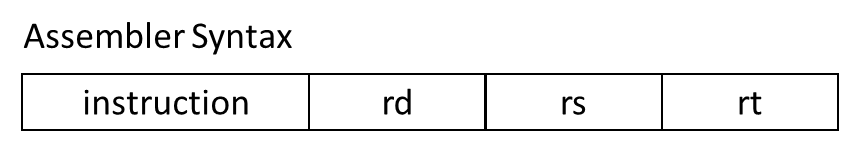
**Exercise**

Implement Fibonacci number computation by using MIPS instructions listed in the table below.

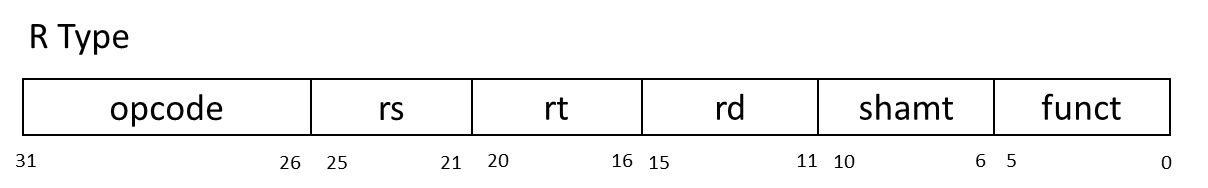
We list some basics instructions for you. DO NOT using MIPS instructions not listed in this table. (Because you will follow this instruction table to implement your CPU. If you use MIPS instructions not in this table, you may not be able to verify your CPU with HW1.)

**MIPS ISA**

**R Type**

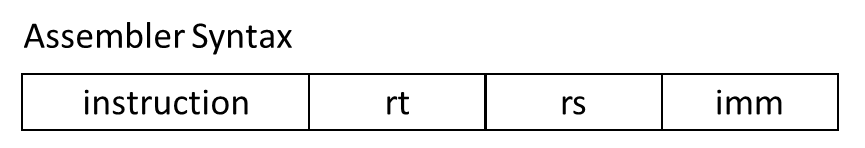
****

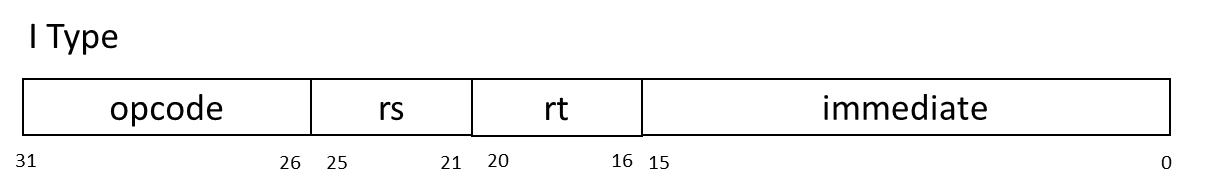
Machine code Format

****

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

****

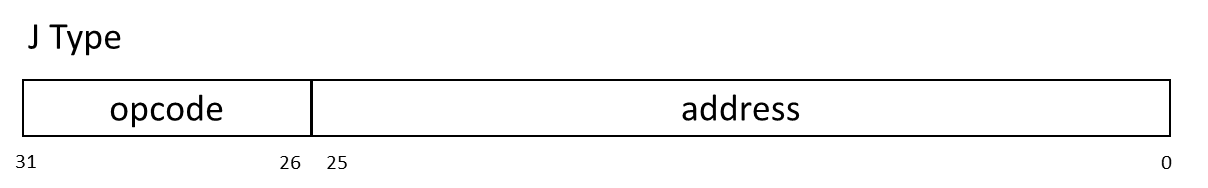
Machine code Format****

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 |

**Fibonacci number**

In mathematics, the Fibonacci numbers or Fibonacci sequence are the numbers in the following integer sequence:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233……

In mathematical terms, the sequence Fn of Fibonacci numbers is defined by the recurrence relation:

F0=0

F1=1

Fn = Fn-1+ Fn-2（n≧2）

By definition, the first two numbers in the Fibonacci sequence are 0 and 1, depending on the chosen starting point of the sequence, and each subsequent number is the sum of the previous two.

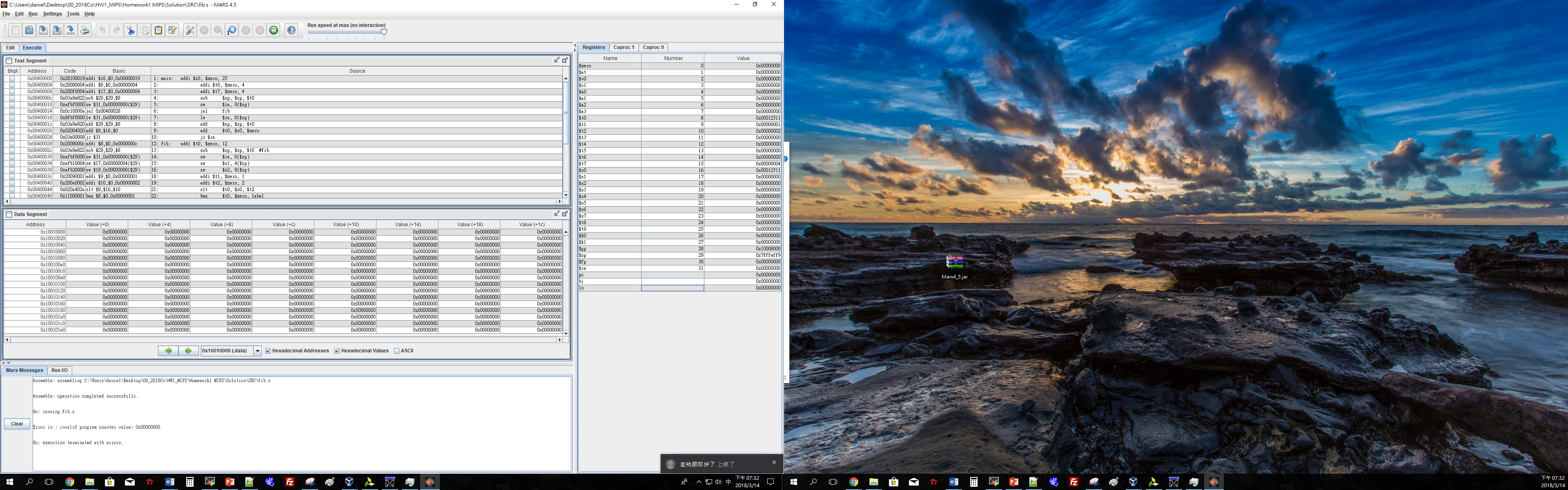
**Fibonacci Pseudo code**

|  |
| --- |
| function fibonacci(n)  integer a = 1  integer b = 0  integer t  for i from 1 to n  t = a + b  b = a  a = t  return a |

**Homework Requirements**

1. Implement Fibonacci number computation according to the above MIPS instruction table.
2. Use MIPS Simulator (Mars) to run your assembly code to compute Fibonacci(25) and store result into register $t0.
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 you have done and followed all requirements, including **File hierarchy**, **Requirement file** and **Report format**.  If you have any questions, please contact us. |