CS311 Project 2: Building a Simple MIPS Emulator

Due 11:59pm, November 5th

1. Overview

This second project is to build an emulator of a subset of the MIPS instruction set. The emulator loads a MIPS binary into an emulated memory, and execute the instructions. Instruction execution will change the states of registers and memory.

2. Emulation Details

For a given input MIPS binary (the output binary file from the assembler built in Project 1), the emulator must be able to mimic the behaviors of the MIPS ISA execution.

2.1 States

The emulator must maintain the system states, which consist of the necessary register set (R0-R31, PC) and the memory. The register and memory must be created when the emulation begins.

2.2 Loading an input binary.

For a given input binary, the loader must identify the text and data section sizes. The text section must be loaded to the emulated memory from the address 0x400000. The data section must be loaded to the emulated memory from the address 0x10000000.

In this project, the simple loader does not create the stack region.

2.3 Initial states

PC: The initial value of PC is 0x400000.
Registers: All values of register0 to 31 are set to zero.
Memory: You may assume all initial values are zero, except for the loaded text and data sections.

2.4 Instruction execution

With the current PC, 4B from the memory is read. The emulator must parse the binary instruction and identify what the instruction is and what are the operands. Based on the MIPS ISA, the emulator must accurately mimic the execution, which will update either a PC, register, or memory.

2.5 Completion.

The emulator must stop after executing a give number of instructions.

2.6 Supported Instruction Set

ADDIU	ADDU	AND	ANDI	BEQ	BNE	J
JAL	JR	LUI	LW	LA*	NOR	OR

ORI	SLTIU	SLTU	SLL	SRL	SW	SUBU

3. Emulator Options and Output

3.1 Options

```
cs311em [-m addr1:addr2] [-d] [-n num_instr] inputBinary
```

- -m : Dump the memory content from addr1 to addr2
- -d : Print the register file content for each instruction execution. Print memory content too if –m option is enabled.

The default output is the PC and register file content after the completion of the given number of instructions. If –m option is specified, the memory content from addr1 to addr2 must be printed too.

If –d option is set, the register (and memory dump, if –m is enabled) must be printed for every instruction execution.

3.2 Formatting Output

PC and register content must be printed in addition to the optional memory content.
You should print the output with standard output.

1. If you type the command line as below, the output file should show only PC and register values like Figure 1.

$> ./cs311em –n 0 input.o

2. If you type the command line as below, the output file should show memory contents of specific memory region, PC and register values like Figure 2.

$> ./cs311em –m 0x400000:0x400010 –n 0 input.o

3. The TAs will provide the functions for printing output format through uploading util.c and util.h.

Figure 1. result1

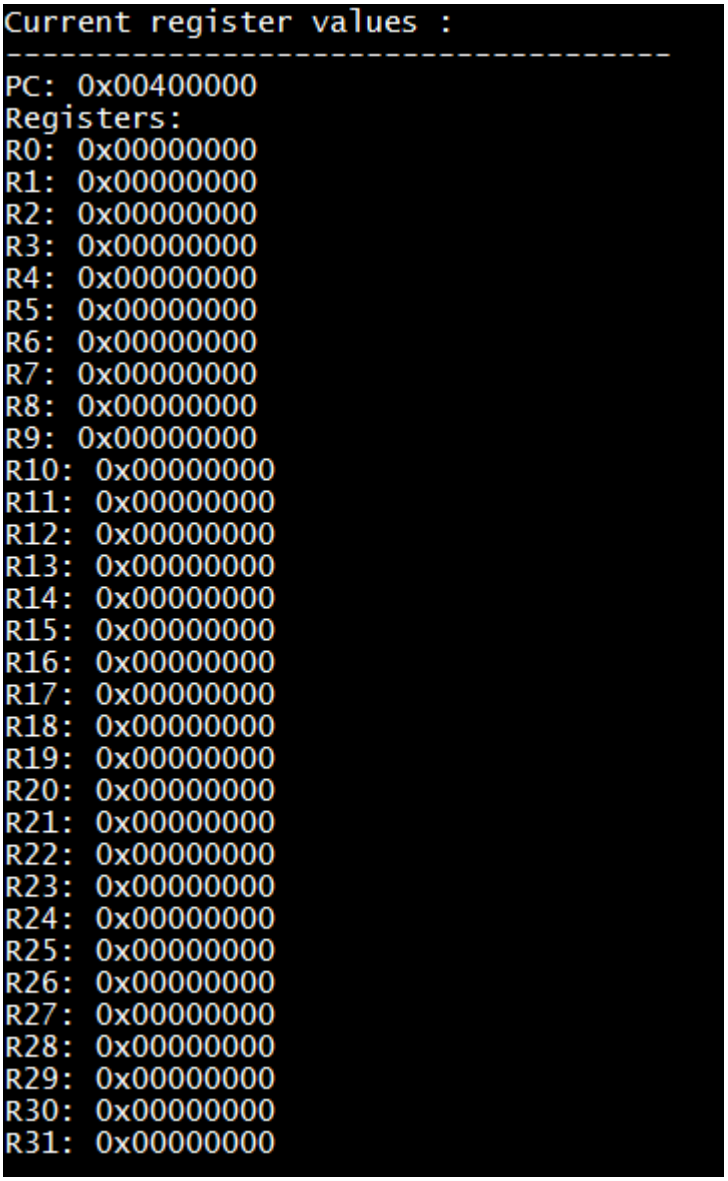
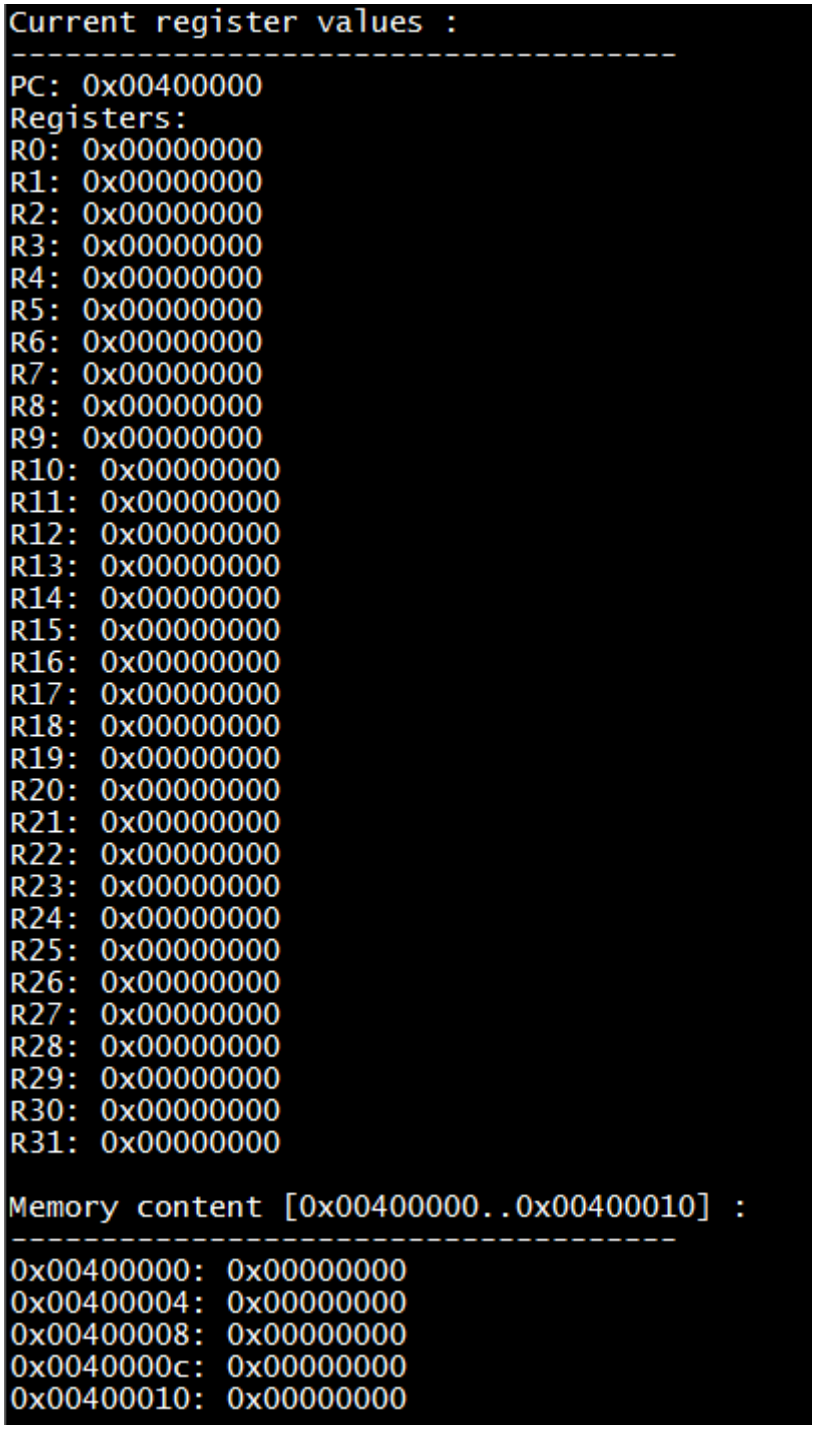


Figure 2. result2



4. Hand in

You should submit the compressed file of the source files through tar or zip program.
Please make the compressed file name with your team name. (team_name.tar or team_name.zip)
Ex) If you are team14, you should make the compressed file with “team14.tar” or “team14.zip”
Please send the email to cs311_ta@calab.kaist.ac.kr with attaching the compressed file until the due date.