

Assembler README

Nishi Baranwal, CS23BTECH11041
Paidala Vindhya, CS23BTECH11044

Contents

1 Download	1
2 How To Use	2
3 Cleaning	2
4 Usage Example	3

1 Download

- First, download the zip file **Lab3_CS23BTECH11041_CS23BTECH11044.zip** and extract it in your computer.
- The folder will contain -
 - Header file - assembler.h
 - Source files - main.cpp, gen_func.cpp, b_format.cpp, i_format.cpp, j_format.cpp, r_format.cpp, s_format.cpp, u_format.cpp
 - Input file - input.s : write your assembly code in this file
 - Makefile
 - Test Cases - file containing cases we tested our code with while developing
 - README file
 - Report file
- Open the folder in terminal and run the command

make

This command will create the .o object files and the riscv_asm executable.

- You can also run the command **make all** instead of make to do the same thing.

2 How To Use

- Since we have already created the executable file : `riscv_asm` (by using the command `make` - please see the [Download](#) section).
To use the executable we have the syntax :

`./riscv_asm`

- To convert the RISC-V (RV64I variant) assembly code to its equivalent machine code, write the code in the **input.s** file.
- The input instructions should follow the format :

instruction_name appropriate_operands

where, there is only 1 space in between the instruction and the first operand. Also, only 1 space between “,” and second operand and so on. Similarly, one colon after the label and then one space.

- Example :

`addi x1, x0, 4`

- The program should start from the first character in each line.
- There should not be any blank lines in the input assembly code file.
- There should only be one instruction per line.
- There should not be any comments in the input assembly code file.
- There should not be any pseudo instructions in the input assembly code file.
- Label names cannot have spaces or colons.
- The output for each line will be printed in the **output.hex** file.
- In the event of format not being correct, the Error will be printed on the **Terminal** and the **output.hex** file.

3 Cleaning

- To remove all the object files and the executable file created by the command `make` run the following command :

`make clean`

- After running the above command your directory will be left with just the files that you had initially downloaded and the output file - `output.hex` .

4 Usage Example

```
● cookie@cookie-ThinkPad-X1-Yoga-Gen-5:~/Khushi/IITH/Y02S03/CS2323 - Computer Arch./Lab/Lab 3/Assembler$ make
g++ -c main.cpp
g++ -c gen_func.cpp
g++ -c b_format.cpp
g++ -c i_format.cpp
g++ -c j_format.cpp
g++ -c r_format.cpp
g++ -c s_format.cpp
g++ -c u_format.cpp
g++ main.o gen_func.o b_format.o i_format.o j_format.o r_format.o s_format.o u_format.o -o riscv_asm
● cookie@cookie-ThinkPad-X1-Yoga-Gen-5:~/Khushi/IITH/Y02S03/CS2323 - Computer Arch./Lab/Lab 3/Assembler$ ./riscv_asm
○ cookie@cookie-ThinkPad-X1-Yoga-Gen-5:~/Khushi/IITH/Y02S03/CS2323 - Computer Arch./Lab/Lab 3/Assembler$
```

asm input.s	output.hex
1 addi x1, x1, -1	1 fff08093
2 add x1, x0, x0	2 000000b3
3 Exit: beq x0, x0, Exit	3 00000063
	4

Figure 1: Executable example with correct input and using make command

```
● cookie@cookie-ThinkPad-X1-Yoga-Gen-5:~/Khushi/IITH/Y02S03/CS2323 - Computer Arch./Lab/Lab 3/project (3)$ ./riscv_asm
Error in Line 3 : Incorrect Instruction Name
○ cookie@cookie-ThinkPad-X1-Yoga-Gen-5:~/Khushi/IITH/Y02S03/CS2323 - Computer Arch./Lab/Lab 3/project (3)$
```

asm input.s	output.hex
1 addi x1, x1, -1	1 fff08093
2 add x1, x0, x0	2 000000b3
3 Exit: bqe x0, x0, Exit	3 Error: Incorrect Instruction Name
	4

Figure 2: Executable example with incorrect input

```
● cookie@cookie-ThinkPad-X1-Yoga-Gen-5:~/Khushi/IITH/Y02S03/CS2323 - Computer Arch./Lab/Lab 3/project (3)$ make clean
rm -f *.o
rm -f riscv_asm
○ cookie@cookie-ThinkPad-X1-Yoga-Gen-5:~/Khushi/IITH/Y02S03/CS2323 - Computer Arch./Lab/Lab 3/project (3)$
```

Figure 3: Cleaning - make clean