CA2020 – HW2

RISC-V Assembly Code – A simple calculator

Description

- In this homework, you have to use a RISC-V simulator to develop a simple calculator program

Jupiter

- An open source RISC-V assembler and runtime simulator (32bit) [GitHub Repo]
- Jupiter is available on the CSIE Workstation, both CLI and GUI are supported
- To use the GUI interface, you can install Jupiter on your PC or use X11 to run the GUI of Jupiter installed on CSIE workstation

Jupiter

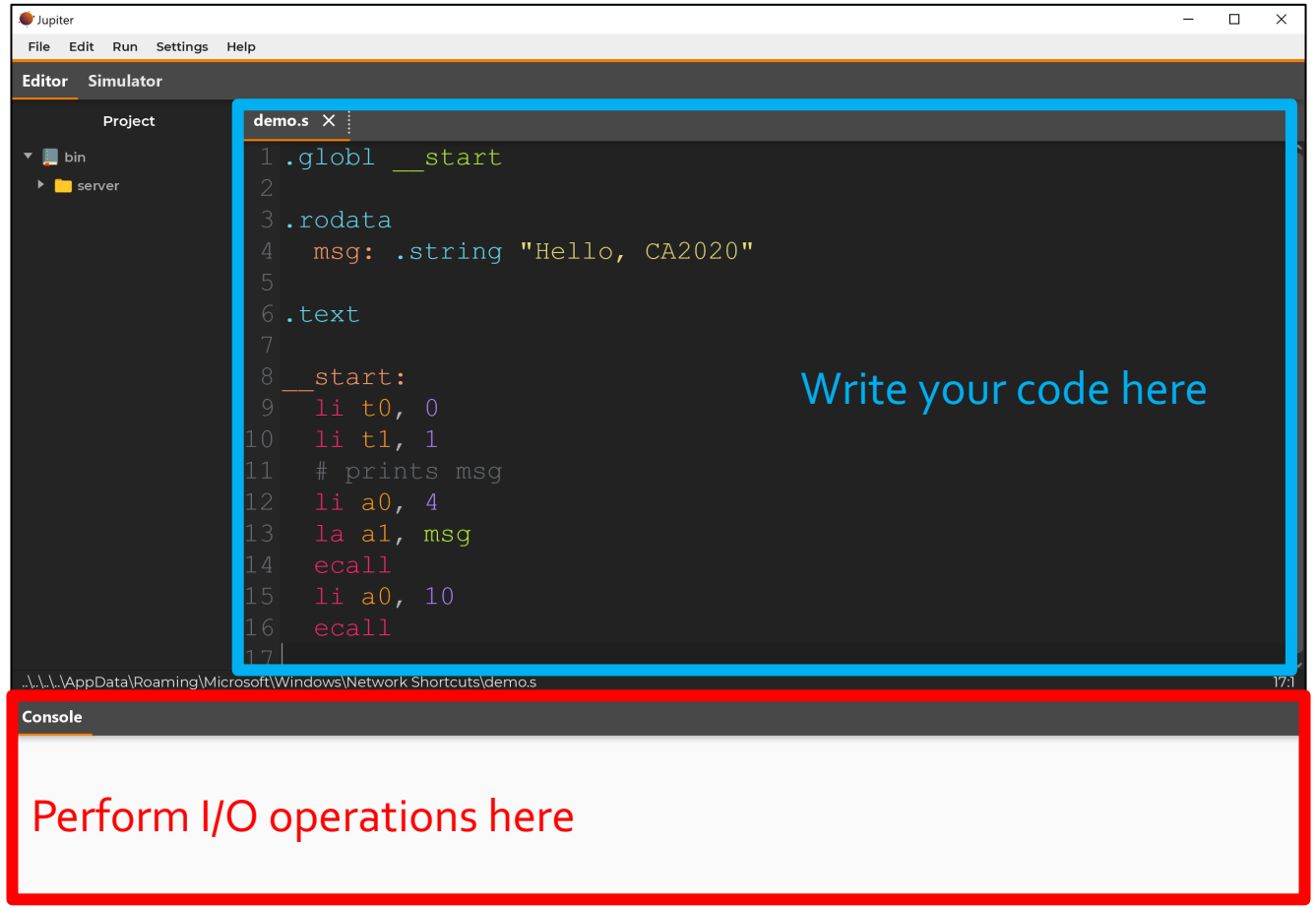
RISC-V Assembler & Runtime Simulator

Jupiter tutorial – CLI

```
d08922025@linux1 [~] jupiter -h
        __          _ __
    __ / /_ _____  (_) /____ ____
   / // / // / _ \/ / __/ -_) __/
   \___/\_,_/ .__/_/\__/\__/_/
           /_/

RISC-V Assembler & Runtime Simulator

usage: jupiter [options] <files>

[General Options]
  -h, --help                show Jupiter help message and exit
  -v, --version             show Jupiter version
  -l, --license             show Jupiter license

[Simulator Options]
  -b, --bare                bare machine (no pseudo-instructions)
  -s, --self                enable self-modifying code
  -e, --extrict             assembler warnings are consider errors
  -g, --debug               start debugger
      --start <label>       set global start label (default: __start)
      --hist <size>         history size for debugging

[Cache Options]
  -c, --cache               enable cache simulation
      --assoc <assoc>       cache associativity as a power of 2 (default: 1)
      --block-size <size>   cache block size as a power of 2 (default: 16)
      --num-blocks <num>    number of cache blocks as a power of 2 (default: 4)
      --policy <policy>     cache block replace policy (LRU|FIFO|RAND) (default: LRU)

[Dump Options]
      --dump-code <file>    dump generated machine code to a file
      --dump-data <file>    dump static data to a file

Please report issues at https://github.com/andrescv/Jupiter/issues
```

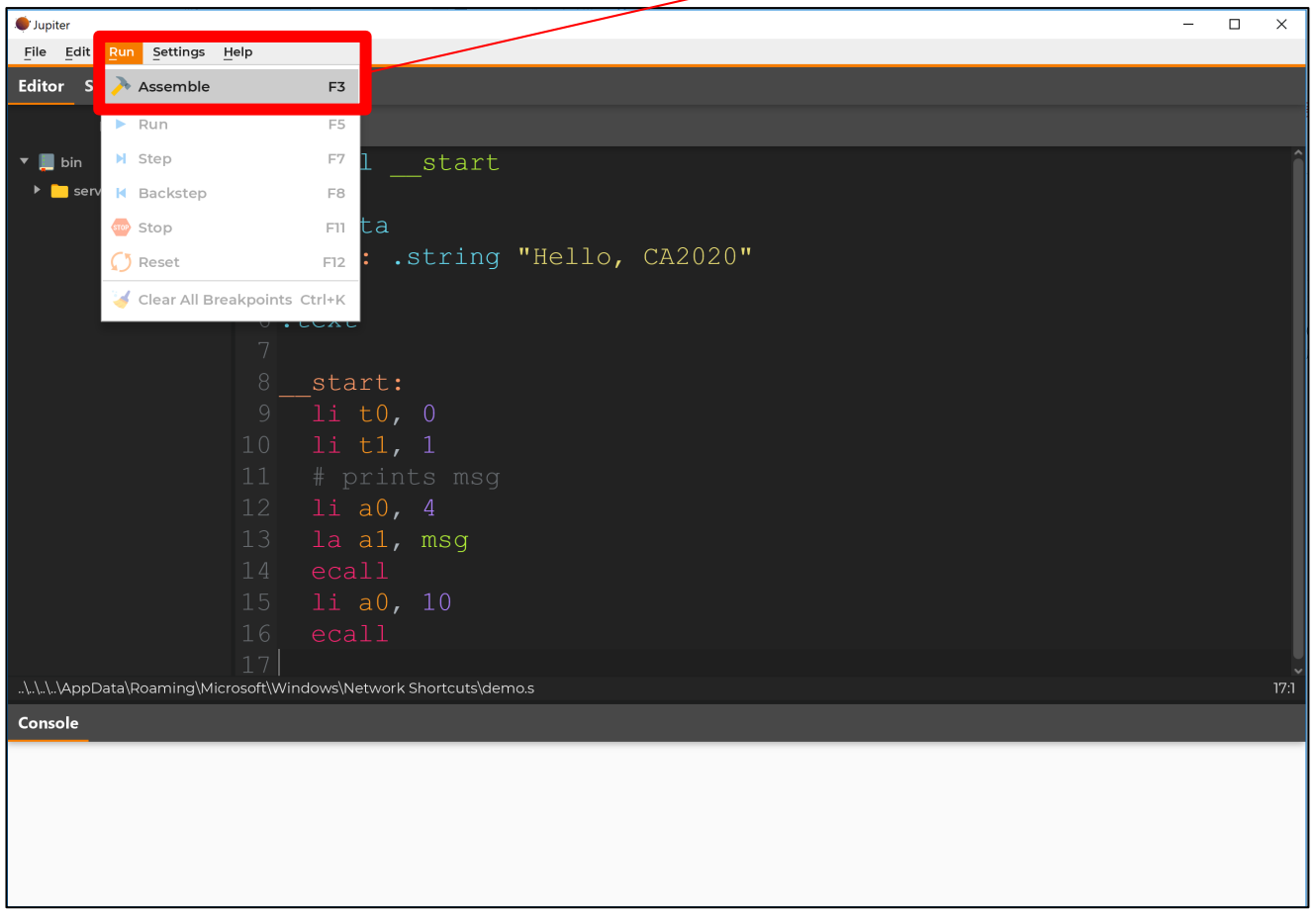
```
d08922025@linux1 [~] jupiter demo.s
Hello, CA2020

Jupiter: exit(0)
```

Jupiter tutorial - GUI



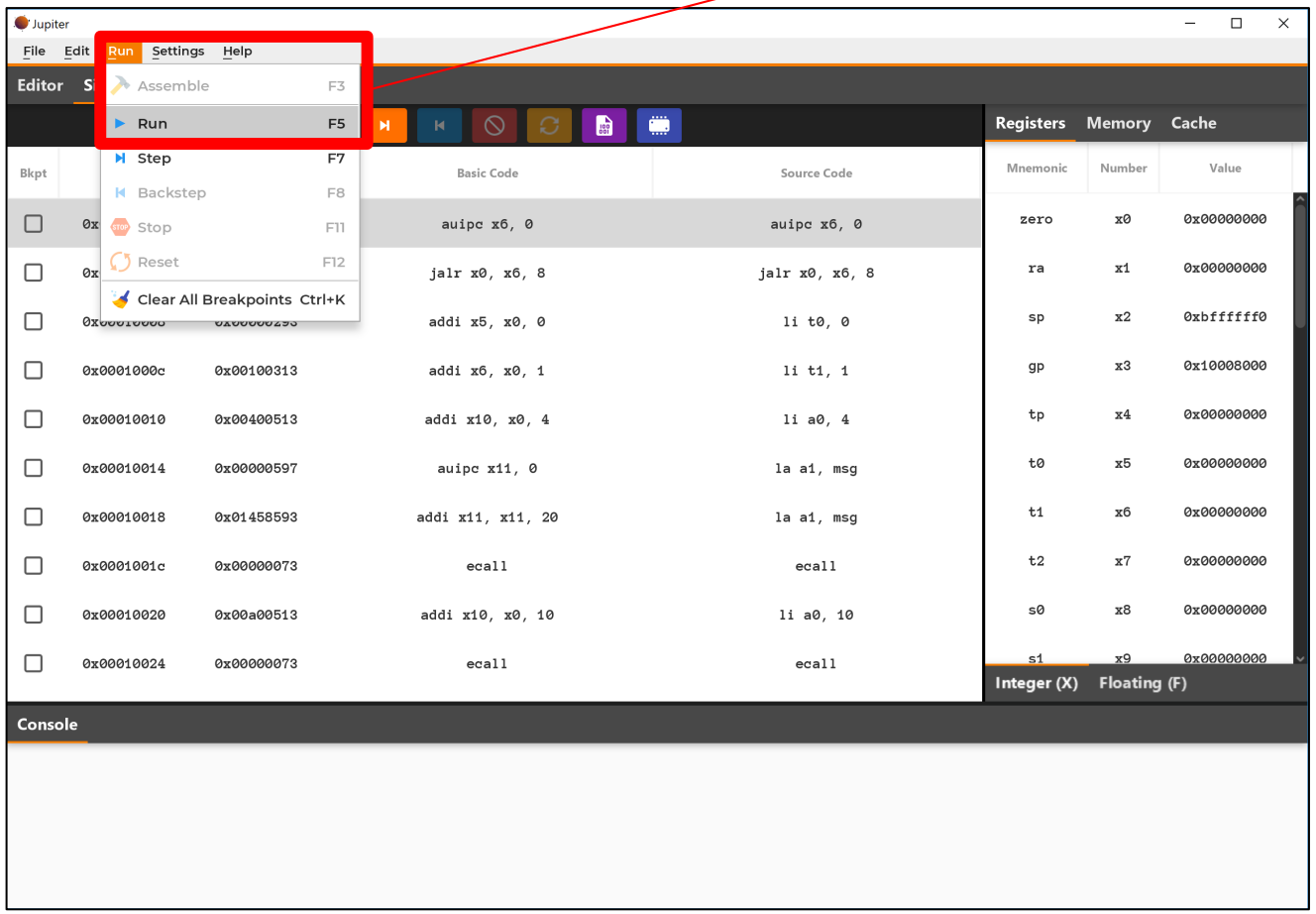
Jupiter tutorial - GUI

Click **Run** > **Assemble**

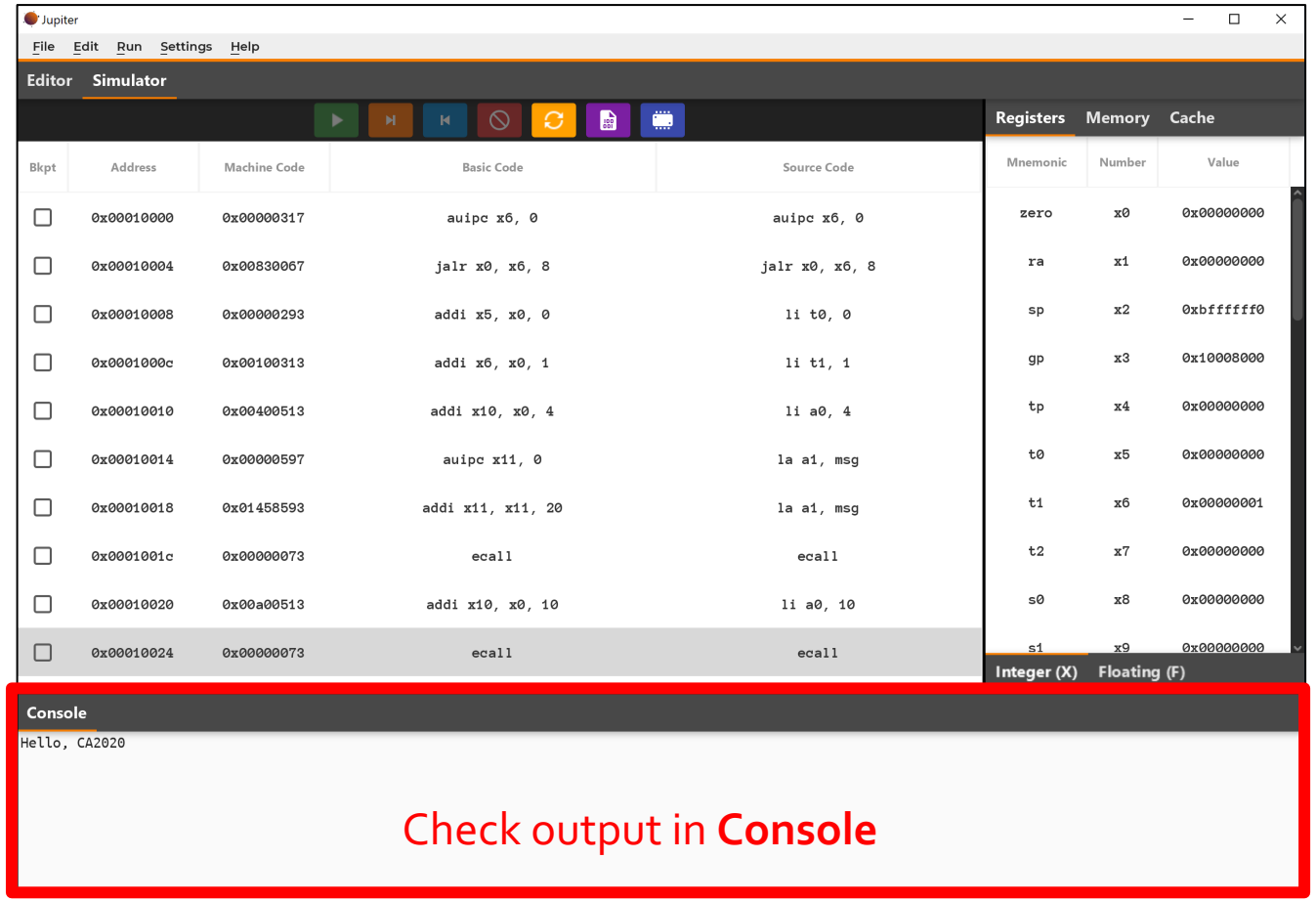


Jupiter tutorial - GUI

Click **Run** > **Run** to execute your code



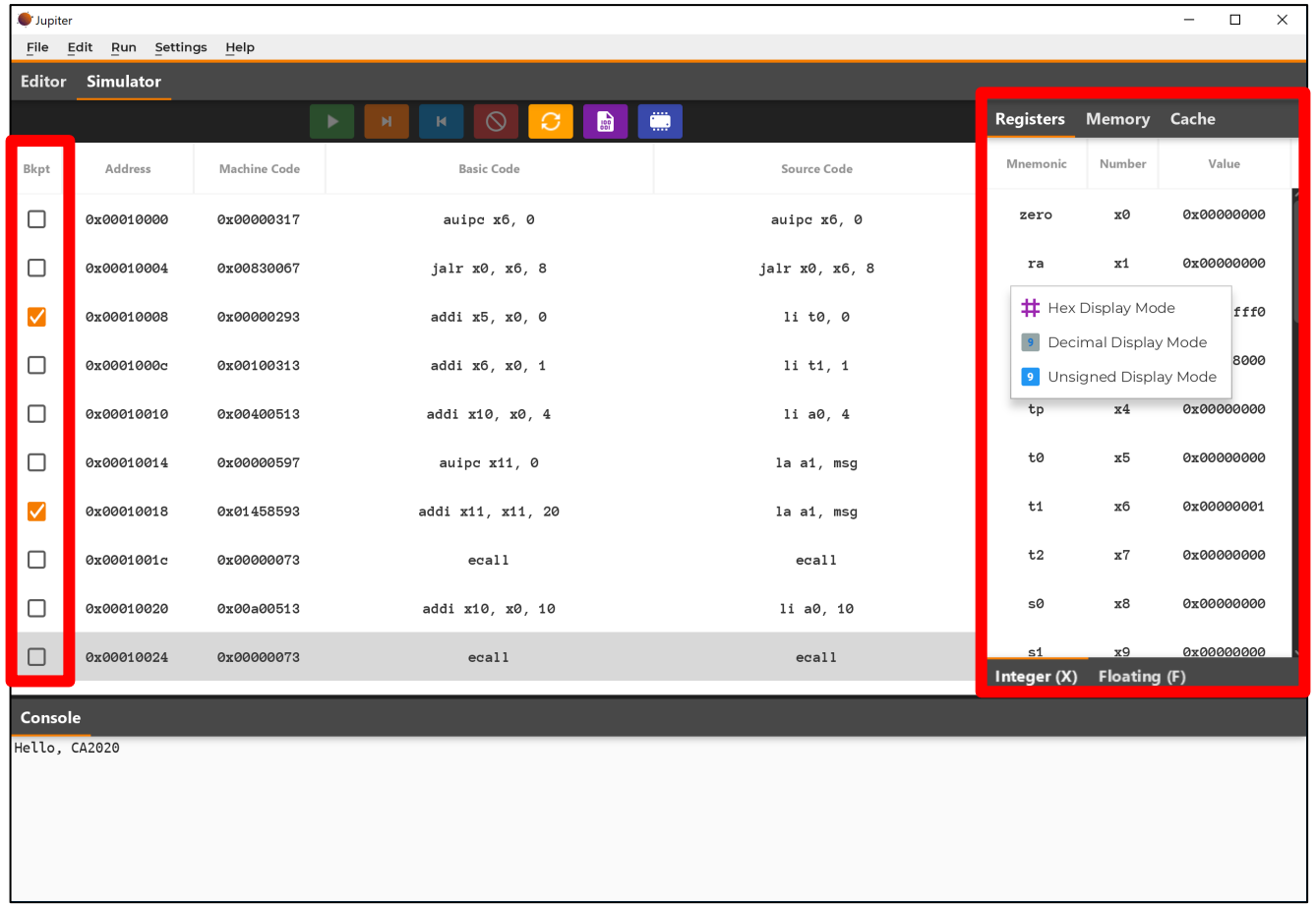
Jupiter tutorial - GUI



Check output in **Console**

Jupiter tutorial - debug

Set breakpoint



Check registers
(right click to change display mode)

TODO: A simple calculator

- In this homework, you are asked to implement a simple calculator.
- The calculator should be able to output the correct results of $A\ op\ B$
  - $-1024 \leq A, B \leq 1024; A, B, op \in \mathbb{Z}$
  - If $op = 0$, calculate $A + B$ and output the result
  - If $op = 1$, calculate $A - B$ and output the result
  - If $op = 2$, calculate $A \times B$ and output the result
  - If $op = 3$, calculate $A \div B$ and output the result (integer division)
  - If $op \notin \{0, 1, 2, 3\}$, you should print "$op\ not\ supported$"
  - If divided by zero, you should print "$divided\ by\ zero$"
- Input format
  - Input file will contain 3 lines. Line1: A, Line2: op, Line3: B

Sample code

- Operations related to I/O have been implemented in the sample code. A, op, B will be stored in register $s0$, $s1$, $s2$ respectively. You have to store the result to $s3$ and jump to `result`. Please DON'T modify the code outside the TODO block.

```
##################################################################
# TODO block: Add your code here
#
##################################################################
```

- To print the "*zero division*" or "*op not supported*" message, just jump to `zero_div` or `op_not_supported`. You don't have to implement by yourself.

```
zero_div:
    li a0, 4
    la a1, zero_div_msg
    ecall

    j return
```

```
op_not_supported:
    li a0, 4
    la a1, op_not_supported_msg
    ecall

    j return
```

Sample output

```
d08922025@linux1 [~] jupiter hw2.s
3
0
-4
-1

Jupiter: exit(0)
d08922025@linux1 [~] jupiter hw2.s
3
1
-4
7

Jupiter: exit(0)
d08922025@linux1 [~] jupiter hw2.s
3
2
-4
-12

Jupiter: exit(0)
```

```
d08922025@linux1 [~] jupiter hw2.s
3
3
-4
0

Jupiter: exit(0)
d08922025@linux1 [~] jupiter hw2.s
3
3
0
divided by zero

Jupiter: exit(0)
d08922025@linux1 [~] jupiter hw2.s
3
5
-4
op not supported

Jupiter: exit(0)
```

Scoring

- We will judge the correctness of your code by running `$ jupiter hw2.s < input_file` on CSIE workstation
- Don't worry about overflow/underflow, it won't happen in this homework
- 100 pts for calculator (20 testcase, 5 pts per testcase)
- 10 pts off per day for late submission
- You will get zero pts for plagiarism

Submission

- Due date: 2020/10/21 Wed. 14:20
- Please compress your homework into a *.zip file and upload to NTUCOOL.
- After unzipping, the folder should have the following structure:
  - d08922025_hw2 (lowercase)
    - readme.txt (Write down what platform you use. Linux, windows, MacOS …)
    - hw2.s (Remember to change the filename)