Computer Architecture
HW#1 Simple Calculator

Dept. of Computer Engineering

Jaehyun Nam

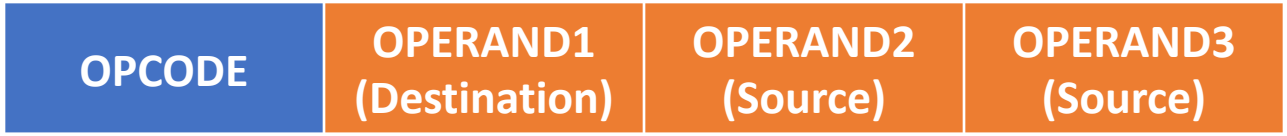
jaehyun.nam@dankook.ac.kr

Simple Calculator

- Goal
  - To do some calculation corresponding to input strings
- Instruction Set Architecture
  - Supports 10 registers (**r0 – r9**)
  - Supports basic arithmetic binary operations
    - **ADD(+), SUB(-), MUL(*), DIV(/)**
  - Supports **MOV** to move register values
  - Supports **LW** and **SW** to load data from the input and print out the result
- Requirements
  - Gets the strings from the input file, **input.txt**
  - Prints out display along with all state changes of used registers
  - Handles exception gracefully

Instructions

- Instruction format

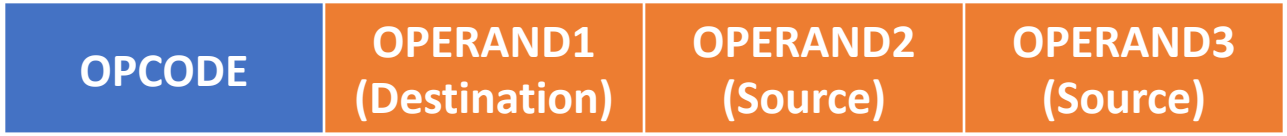


- Opcodes are ADD, SUB, MUL, DIV, MOV, LW, SW, and RST (8 in total)

  - ADD [DST] [SRC1] [SRC2]
  - SUB [DST] [SRC1] [SRC2]
  - MUL [DST] [SRC1] [SRC2]
  - DIV [DST] [SRC1] [SRC2]
  - MOV [DST] [SRC]
  - LW [register] [value from string]
  - SW [register] STDOUT
  - RST # special opcode to reset all registers

Instructions

- Instruction format



- One instruction in one line
- Operands have prefix (begins with)
  - '**0x**' for hexadecimal number
  - '**r**' for register number
- Halt (complete execution) when reaching EOF(End of File)

Workflow

- Compute the instructions while reading each line from **input.txt**
  - Update registers
  - Do arithmetic/data transfer operations
  - Print out the changes of used registers during the execution of each line

- Example

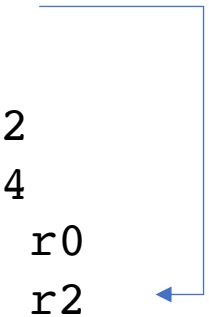
```
LW  r0 0xF          # The second operand is a number
LW  r1 0x4
ADD r2 r0 r1
MOV r0 r2
SW  r0 STDOUT       # STDOUT is a signal to print out $r0
RST                 # Reset all registers
LW  r0 0x2
LW  r1 0x4
MUL r2 r0 r1
MOV r0 r2
SW  r0 STDOUT
```

Extra Point #1

- JMP instruction
  - JMP [line number]

- Example

```
 1: LW  r0 0x5          # The second operand is a number
 2: LW  r1 0xF
 3: ADD r2 r0 r1
 4: JMP 0x9
 5: RST                 # Reset all registers
 6: LW  r0 0x2
 7: LW  r1 0x4
 8: MUL r2 r0 r0
 9: ADD r3 r1 r2
10: MOV r0 r3
11: SW  r0 STDOUT
```



Extra Point #2

- BEQ, BNE, SLT instructions
  - BEQ [OP1] [OP2] [line number] # if (OP1 == OP2) then go to [line number]
  - BNE [OP1] [OP2] [line number] # if (OP1 != OP2) then go to [line number]
  - SLT **r0** [OP1] [OP2] # if (OP1 < OP2) then set r0 = 1 else r0 = 0

- Example

```
1: LW   r0 0x5
2: LW   r1 0xA
3: BEQ r0 r1 0x6
4: MOV r0 0x0
5: JMP 0x7
6: MOV r0 0x1
7: SW   r0 STDOUT
```

```
1: LW   r0 0x5
2: LW   r1 0xA
3: BNE r0 r1 0x6
4: MOV r0 0x0
5: JMP 0x7
6: MOV r0 0x1
7: SW   r0 STDOUT
```

```
1: LW   r0 0x5
2: LW   r1 0xA
3: SLT r2 r0 r1
4: BNE r0 0x0 0x7
5: MOV r0 0x0
6: JMP 0x7
7: MOV r0 0x1
8: SW   r0 STDOUT
```

Programming Language

- **C** is highly recommended since we need **C** for the following assignments
- But **Java** and **Python** are also okay in this assignment

Document

- Introduction
  - Brief description of the assignment
- Background
  - Important concept, specific considerations for implementation
- Implementation
  - How you organized your program (design)
  - What parts you implemented and what parts you didn't (or couldn't)
    - Including the implementation for extra points
  - Explain why
- Environment
  - How to build the development environment for testing (+ screenshots)
    - Should be specific
    - If I can't create an environment by doing what you've described, I won't grade your code
  - How to compile and run your program
    - Should be specific
  - Screenshots working proofs with explanation
- Lesson
  - What was hard, What you thought while doing the assignment, etc.

E-mail Submission

- Send your assignment to jaehyun.nam@dankook.ac.kr

- E-mail title
  - [2023-1 ARCHITECTURE] HW1 [StudentID] [Name]
  - Ex) [2023-1 ARCHITECTURE] HW1 12345678 남재현

- E-mail body
  - **YOU MUST CHANGE THE NAMES OF YOUR CODE AND DOCUMENT AS FOLLOWS**
  - **IF NOT, YOU WILL GET A PENALTY**
  - Source code
    - **HW1_[학번]_[이름]_code.zip**
    - Ex) HW1_12345678_남재현_code.zip
  - Document
    - **HW1_[학번]_[이름]_document.pdf**
    - Ex) HW1_12345678_남재현_document.pdf

Due Date

- 14:30 on April 7th, 2023
  - Right before class