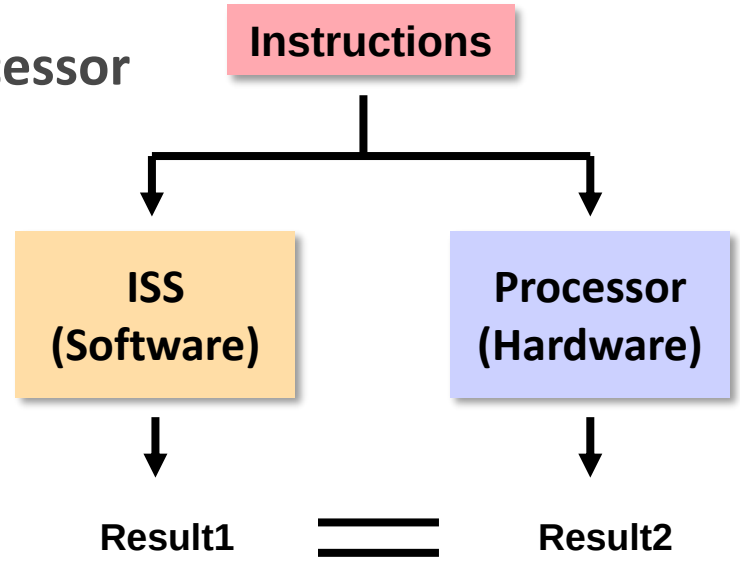
Instruction Set Simulator (ISS) Implementation

E-mail: jylee@ics.kaist.ac.kr
bsjang@ics.kaist.ac.kr

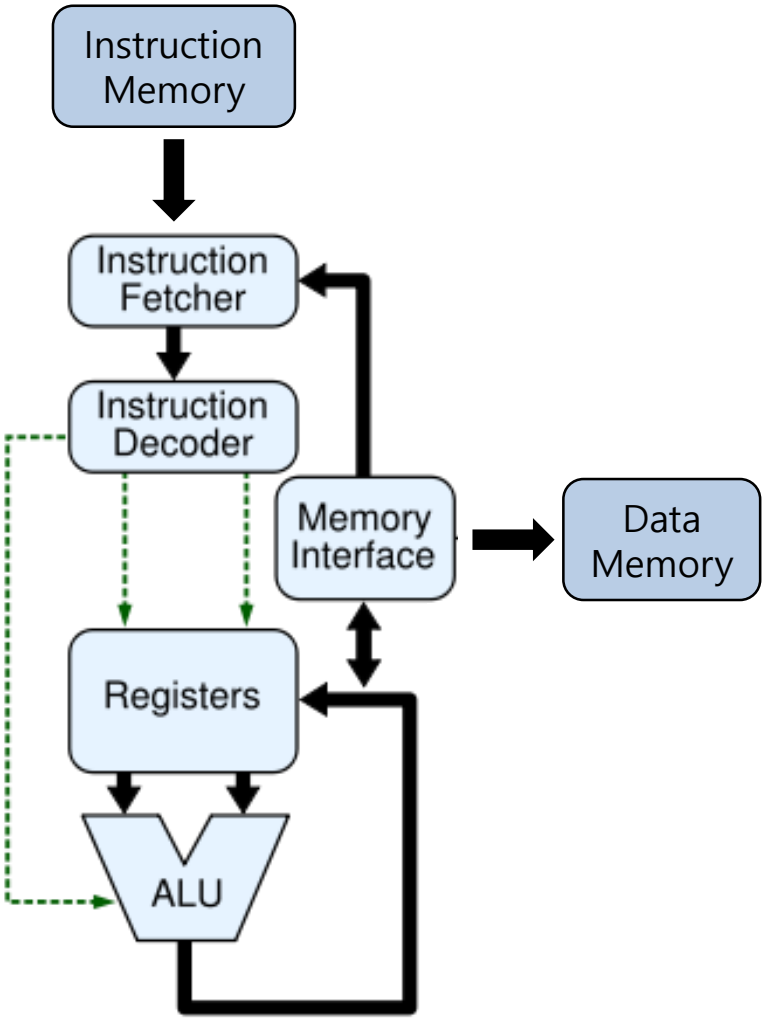
OVERVIEW ON ISS

- ISS is a software program that executes applications as if they were running on the real hardware processor.
- Allowing you to develop and benchmark code before hardware is available – usually used in code verification and driver development
- Provides programmers view of the processor
  - Instructions
  - Contents of registers
  - Contents of memory
  - Status flags



PROJECT SCOPE

- **Overall flow of the ISS**
  - Basically, it is the same as CPU.
  - Fetch an instruction from the instruction memory.
  - Decode the instruction.
  - Execute an operation specified in the instruction.
  - Save the result in registers or data memory.



EXPECTED RESULTS

- **After processing each instruction, you should check below values**
  - Special-purpose registers
    - PC, IR, IE, and IPC
  - 32 General-purpose registers
  - Accessed memory address & data
  - When you type **s** which means step,

```
./krpiss input_file output_file
================================================================================
                              KRP2.0 Instruction Set Simulator
================================================================================
Request:
>> s
PC :00000000
IR :12233112
IE : 0
IPC:00000000
reg[ 0]: 0      reg[ 1]: 0      reg[ 2]: 0      reg[ 3]: 0
reg[ 4]: 0      reg[ 5]: 0      reg[ 6]: 0      reg[ 7]: 0
reg[ 8]: 0      reg[ 9]: 0      reg[10]: 0      reg[11]: 0
reg[12]: 0      reg[13]: 0      reg[14]: 0      reg[15]: 0
reg[16]: 0      reg[17]: 0      reg[18]: 0      reg[19]: 0
reg[20]: 0      reg[21]: 0      reg[22]: 0      reg[23]: 0
reg[24]: 0      reg[25]: 0      reg[26]: 0      reg[27]: 0
reg[28]: 0      reg[29]: 0      reg[30]: 0      reg[31]: 0
>>
```

EXPECTED RESULTS

- When **r**, recursion, is entered

```
./krpiss input_file output_file
================================================================================
                              KRP2.0 Instruction Set Simulator
================================================================================
Request:
>> b                                  ──► Repeat until PC = (3 << 2) = 12
break point: 3
>> r
Executed instruction: 12233112
Executed instruction: 31122331        } Do not need to be shown
Executed instruction: 23311223
>> d
PC :0000000C
IR :12233112                          ──► After the execution, registers are displayed.
IE : 0
IPC:00000000
reg[ 0]: 0     reg[ 1]: 0     reg[ 2]: 0     reg[ 3]: 0
reg[ 4]: 0     reg[ 5]: 0     reg[ 6]: 0     reg[ 7]: 0
reg[ 8]: 0     reg[ 9]: 0     reg[10]: 0     reg[11]: 0
reg[12]: 0     reg[13]: 0     reg[14]: 0     reg[15]: 0
reg[16]: 0     reg[17]: 0     reg[18]: 0     reg[19]: 0
reg[20]: 0     reg[21]: 0     reg[22]: 0     reg[23]: 0
reg[24]: 0     reg[25]: 0     reg[26]: 0     reg[27]: 0
reg[28]: 0     reg[29]: 0     reg[30]: 0     reg[31]: 0
>>
```

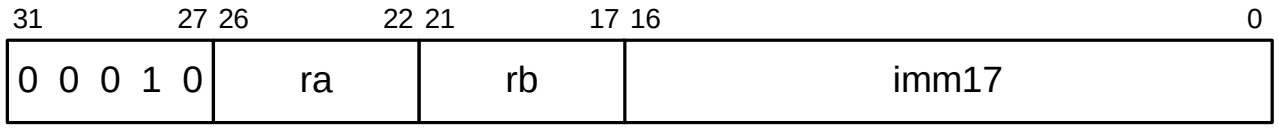
How to Make Input File

- A bunch of instructions is written in a binary file.
- Example

Your test vectors:

```
ANDI r0, r1, #2
ADD r2, r0, r1
...
```

KRP 2.0 manual ANDI:

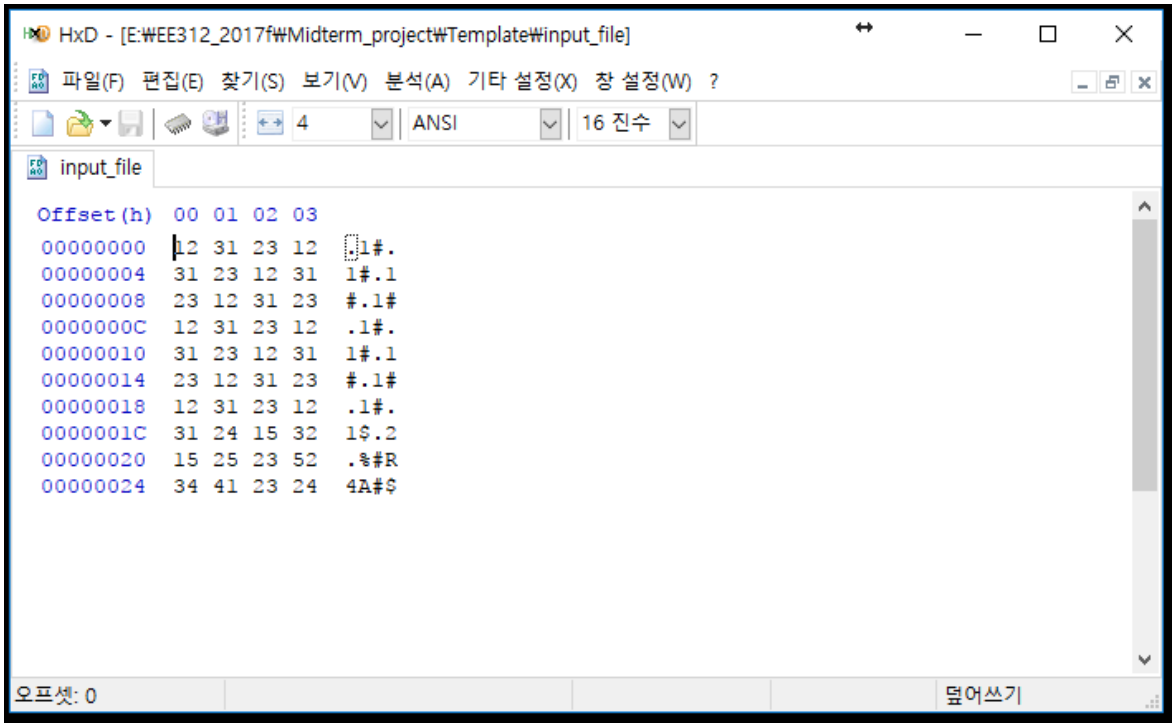


Binary format (example)

```
00 00 00 08 00 00 42 08 02 00 00 00 03 00 42 00 ; ......B.......B.
```

How to Make Input File

- **Hex editor is useful to make a binary file**
  - Free hex editor can be found at
    - http://mh-nexus.de/en/downloads.php?product=HxD
    - You can execute it without installation by downloading the portable version.



Output File

- Memory dump file can be like this form.
  (Contents of data memory)

```
0x00000000 : 00 00 00 00
0x00000004 : 00 00 00 00

...

0x00000FFF : 00 00 00 00
```