CSED311 Lab3: Single Cycle CPU

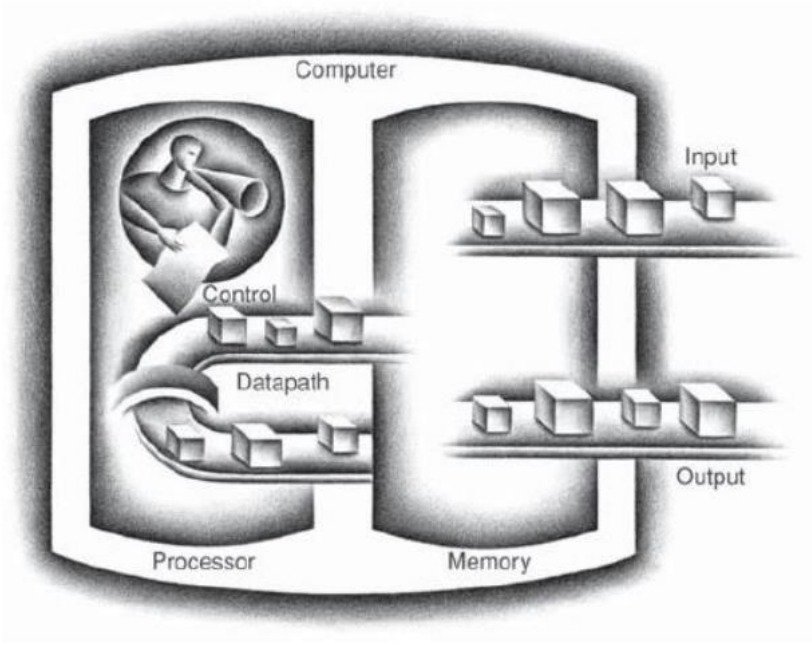
Jaejun Ha
dreamline91@postech.ac.kr

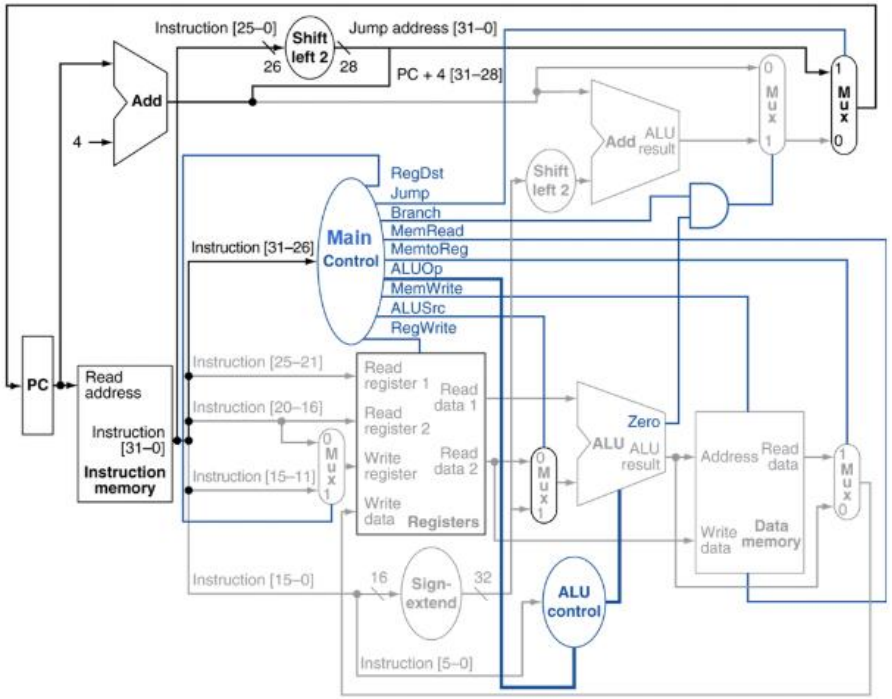
POSTECH

Contents

- CPU structure
  - Datapath
  - Control Unit
- TSC CPU
- Assignment
- Tips
- Lab2 Demo

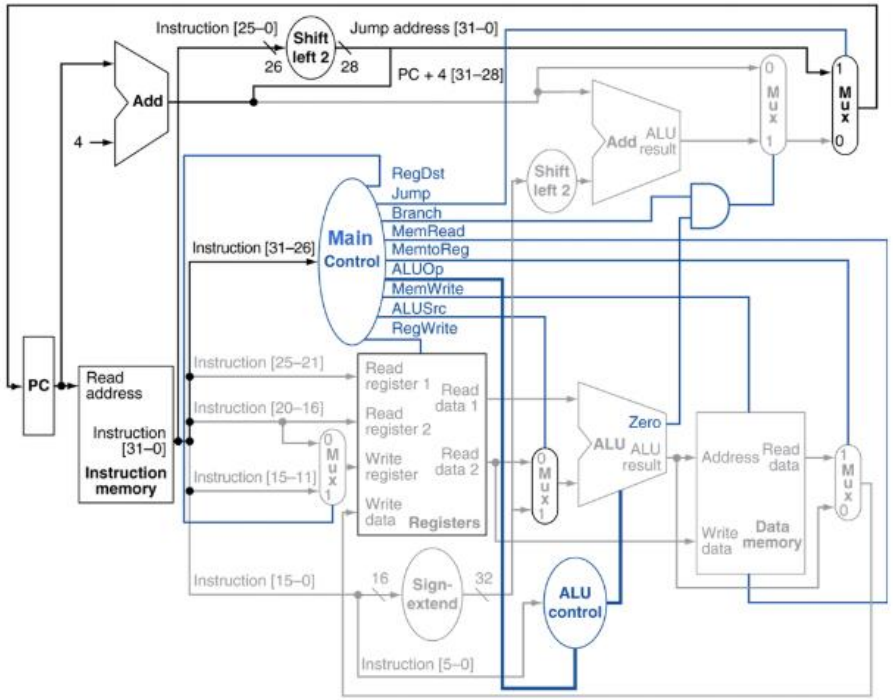
CPU structure

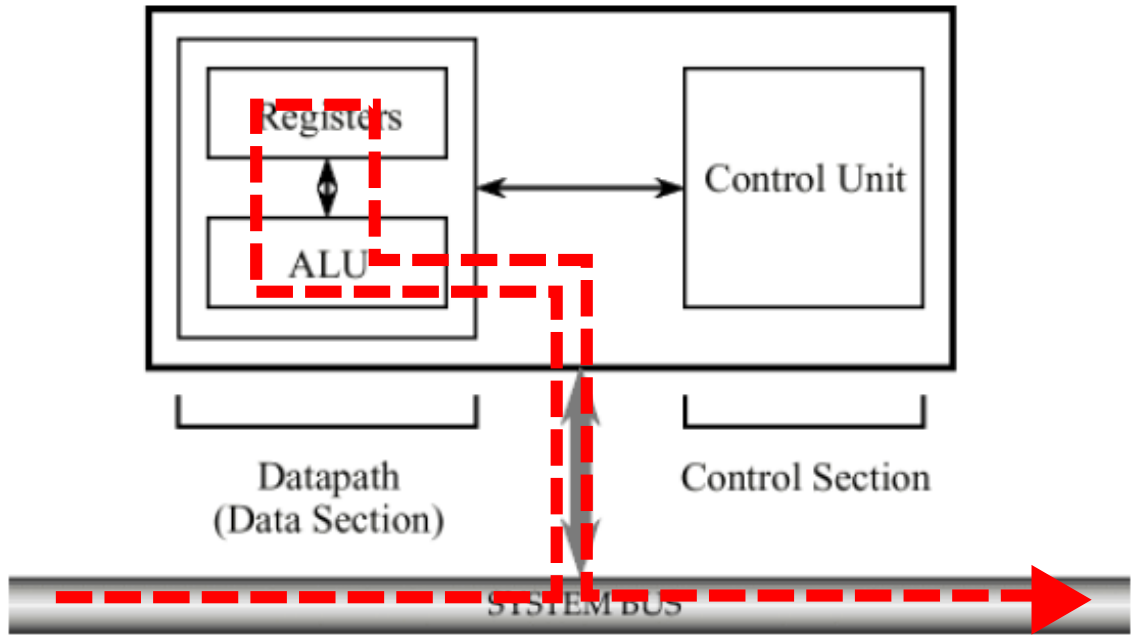




- CPU consists of two components
  - Datapath
  - Control Unit

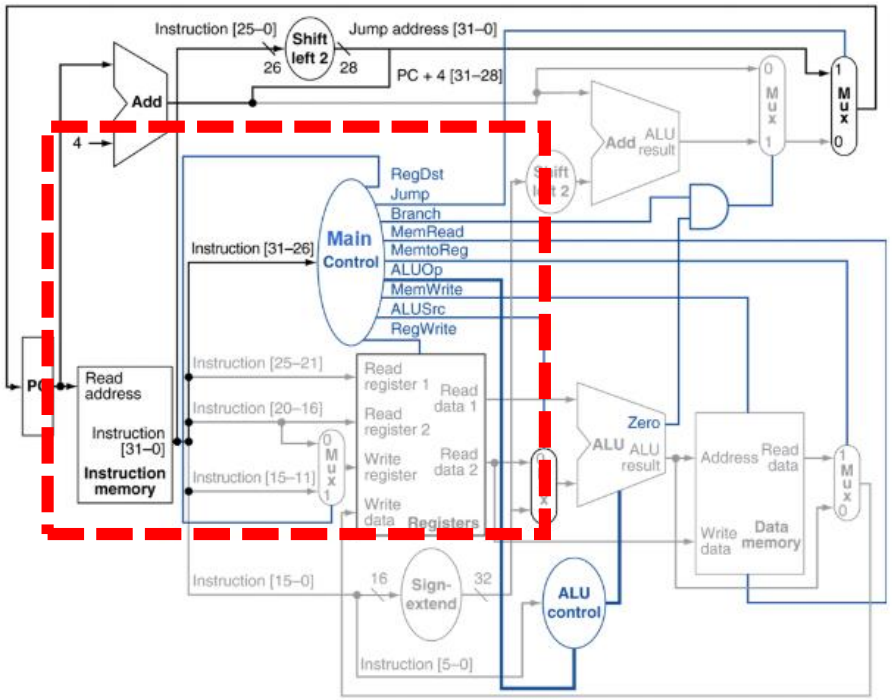
Datapath

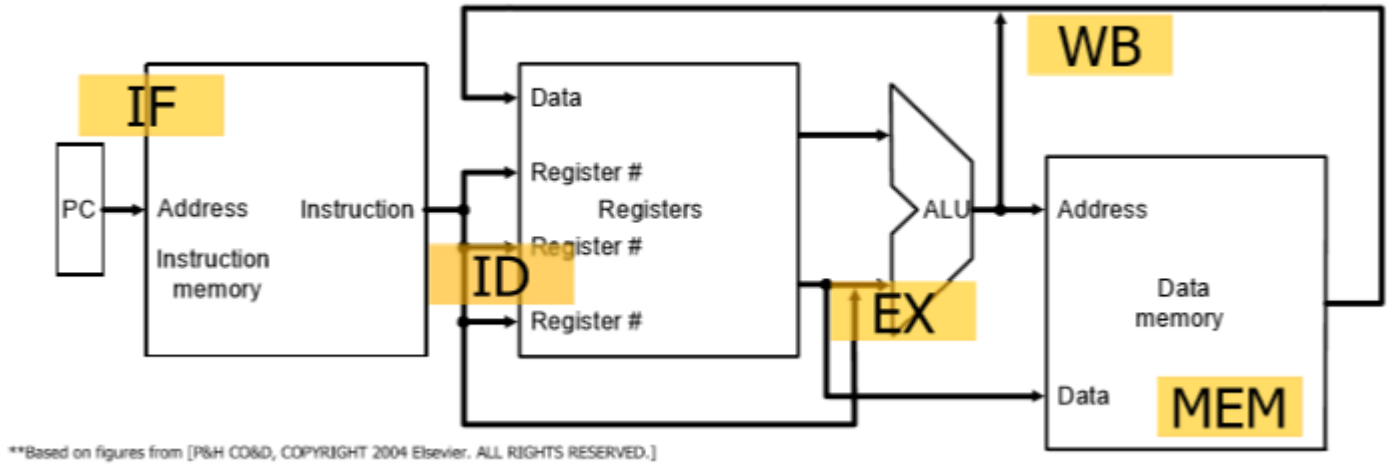




- Datapath:
  - Units in the path of data
    - SYSTEM BUS → ALU → Register → ALU → SYSTEM BUS
  - Instruction fetch, ALU, Register, Memory

Control Unit



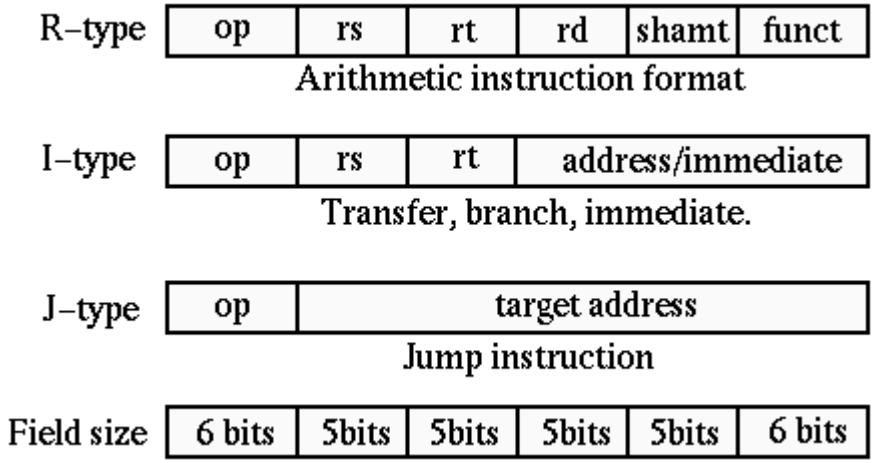


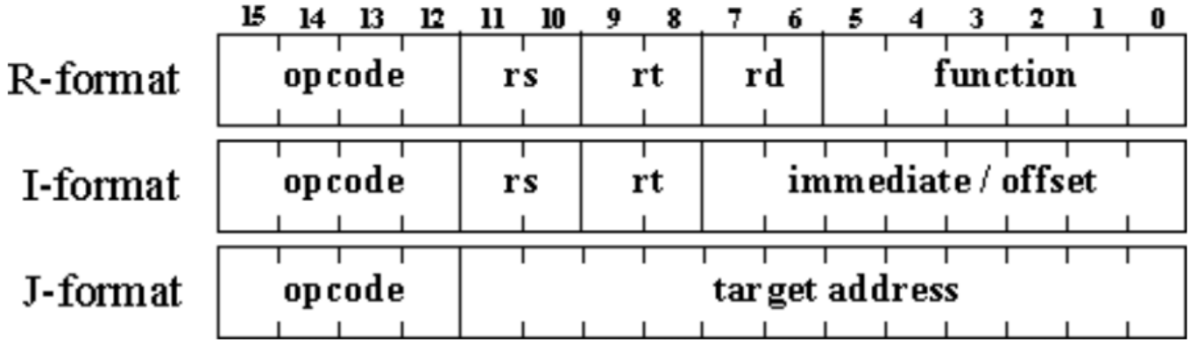
**Based on figures from [P&H CO&D, COPYRIGHT 2004 Elsevier. ALL RIGHTS RESERVED.]

- Control Unit:
  - It decodes instruction
  - It generates control signals which are used in datapath

TSC CPU

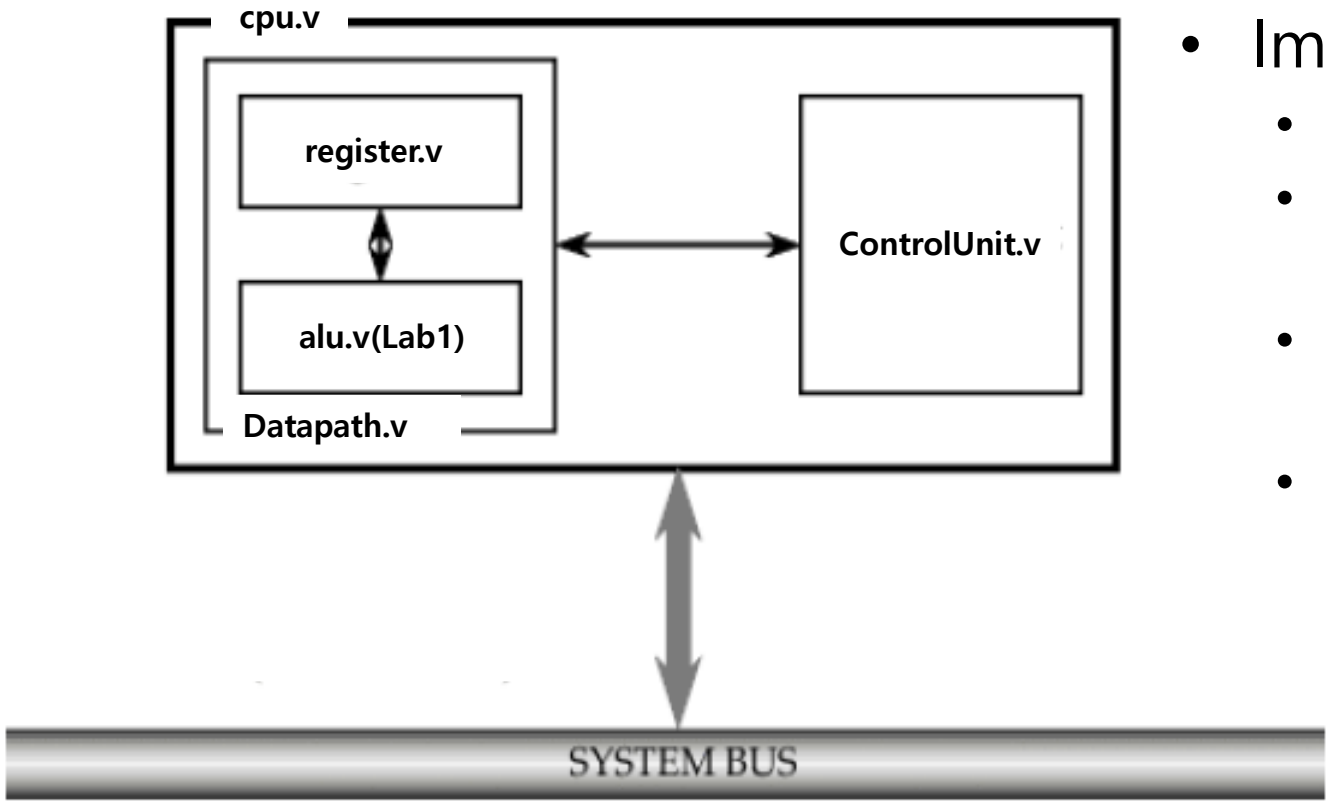
- MIPS CPU vs TSC CPU





  - Has less bits
  - more simple structure

Assignment



- Implement Single Cycle CPU
  - Single-cycle CPU
  - Datapath
    - for 16-bit CPU (4 registers)
  - Control unit
    - to generate control signals used in datapath
  - Instructions
    - refer to opcodes.v

Tips

- CPU module port

output	readM	"read" signal to memory
output	writeM	"write" signal to memory
output	[`WORD_SIZE-1:0] address	target memory address
input	[`WORD_SIZE-1:0] data	data for reading or writing
input	ackOutput	signal from memory ("data is written")
input	inputReady	signal from memory ("data is ready for reading")
input	reset_n	reset CPU
input	clk	clock signal

Tips

- Testbench

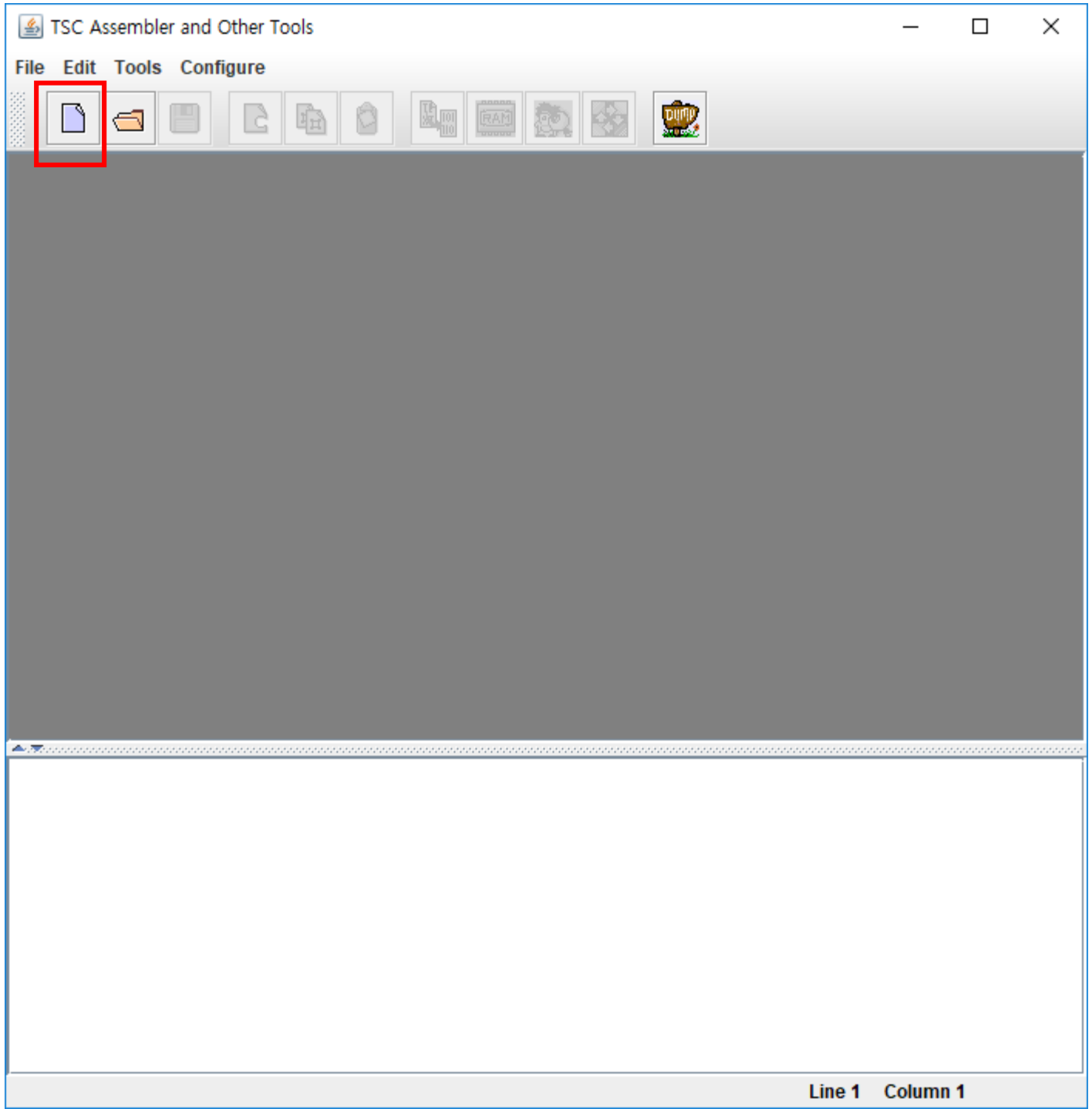
```
assign data = (readM || inputReady) ? loadedData : `WORD_SIZE'bz;
always begin
loadedData = `WORD_SIZE'bz;
#`PERIOD1;
        forever begin
        wait (readM == 1 || writeM == 1);
                if (readM == 1) begin
                #`READ_DELAY;
                loadedData = memory[address];
                        inputReady = 1;
                #(`STABLE_TIME);
                        inputReady = 0;
                loadedData = `WORD_SIZE'bz;
                end else if (writeM == 1) begin
                        memory[address] = data;
                        #`WRITE_DELAY;
                        ackOutput = 1;
                        #(`STABLE_TIME);
                        ackOutput = 0;
                end
end  // of forever loop
end  // of always block for memory read
```

Tips

- TSC assembler
  - tsc_assem > assem > runit.bat
  - It needs Java
  - It can make binary code

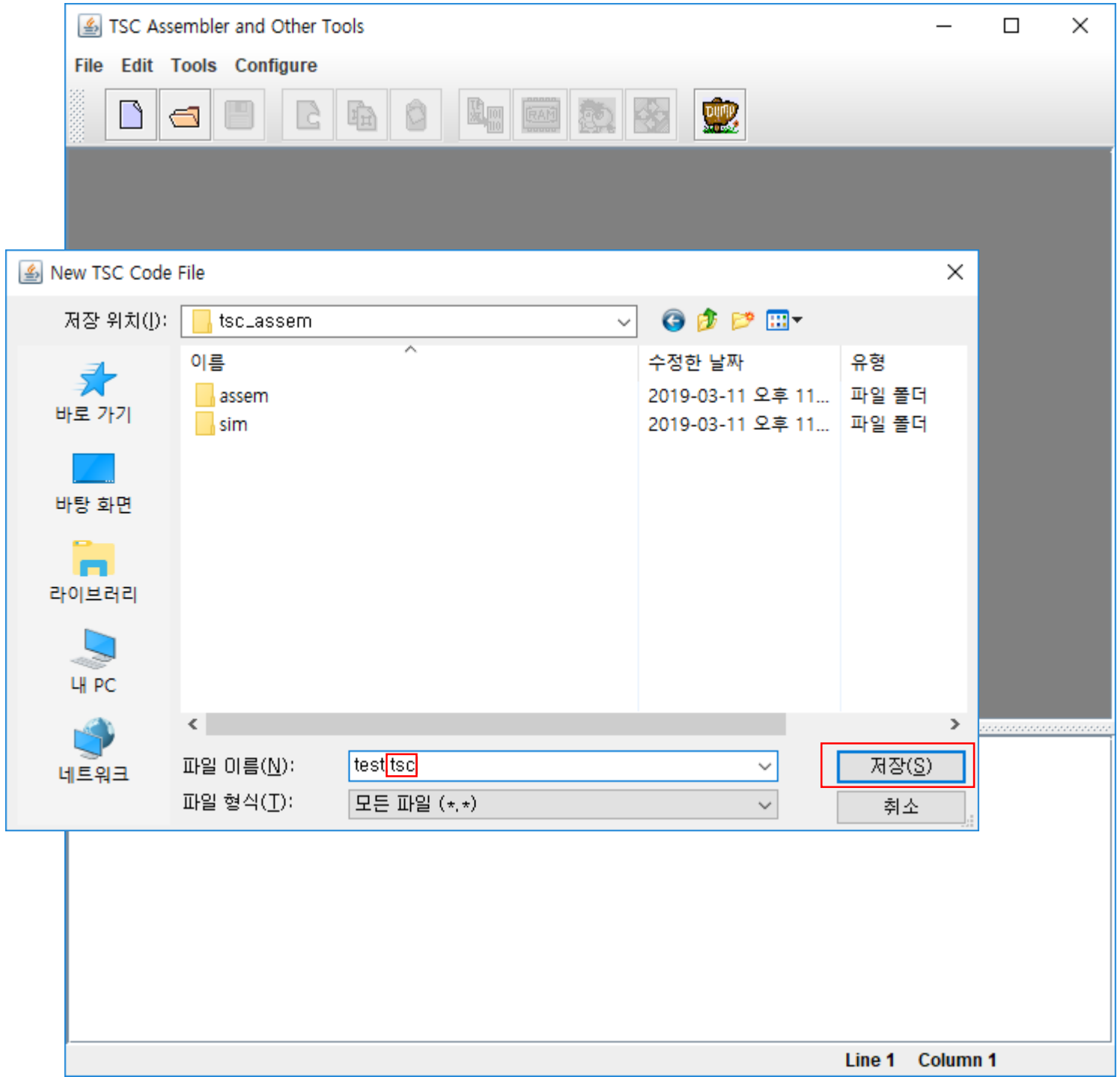
Tips

- TSC assembler



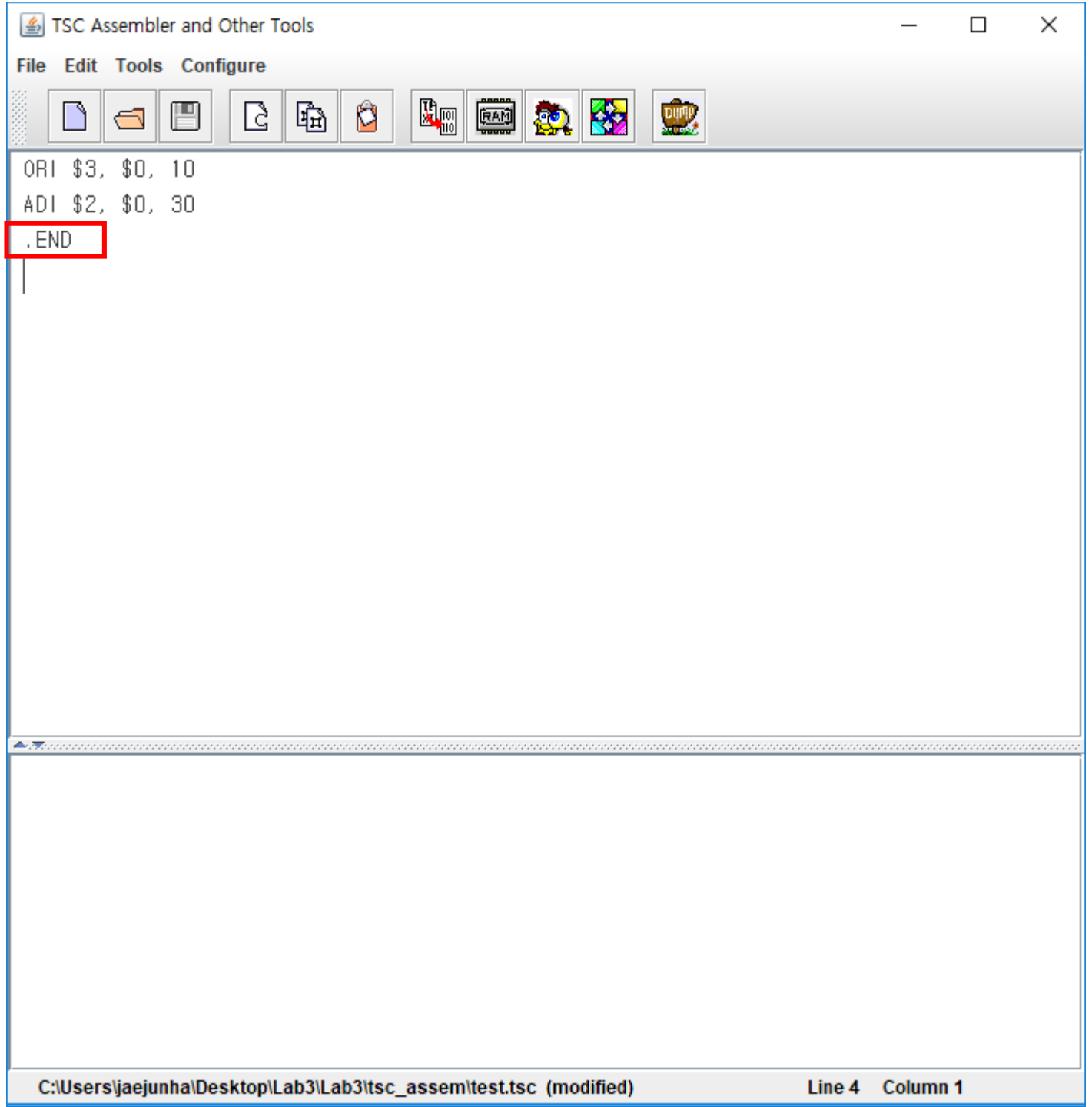
Tips

- TSC assembler



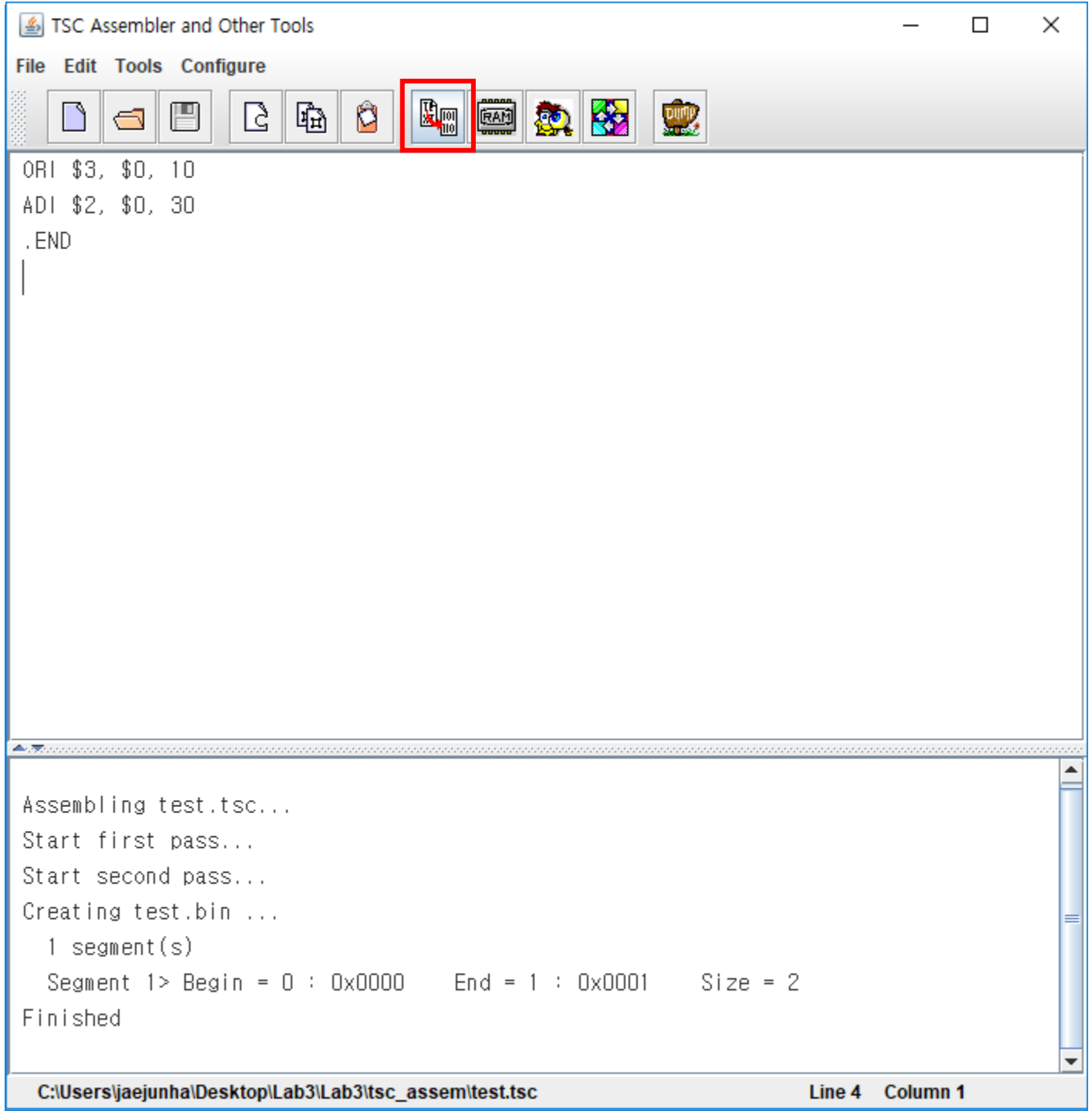
Tips

- TSC assembler



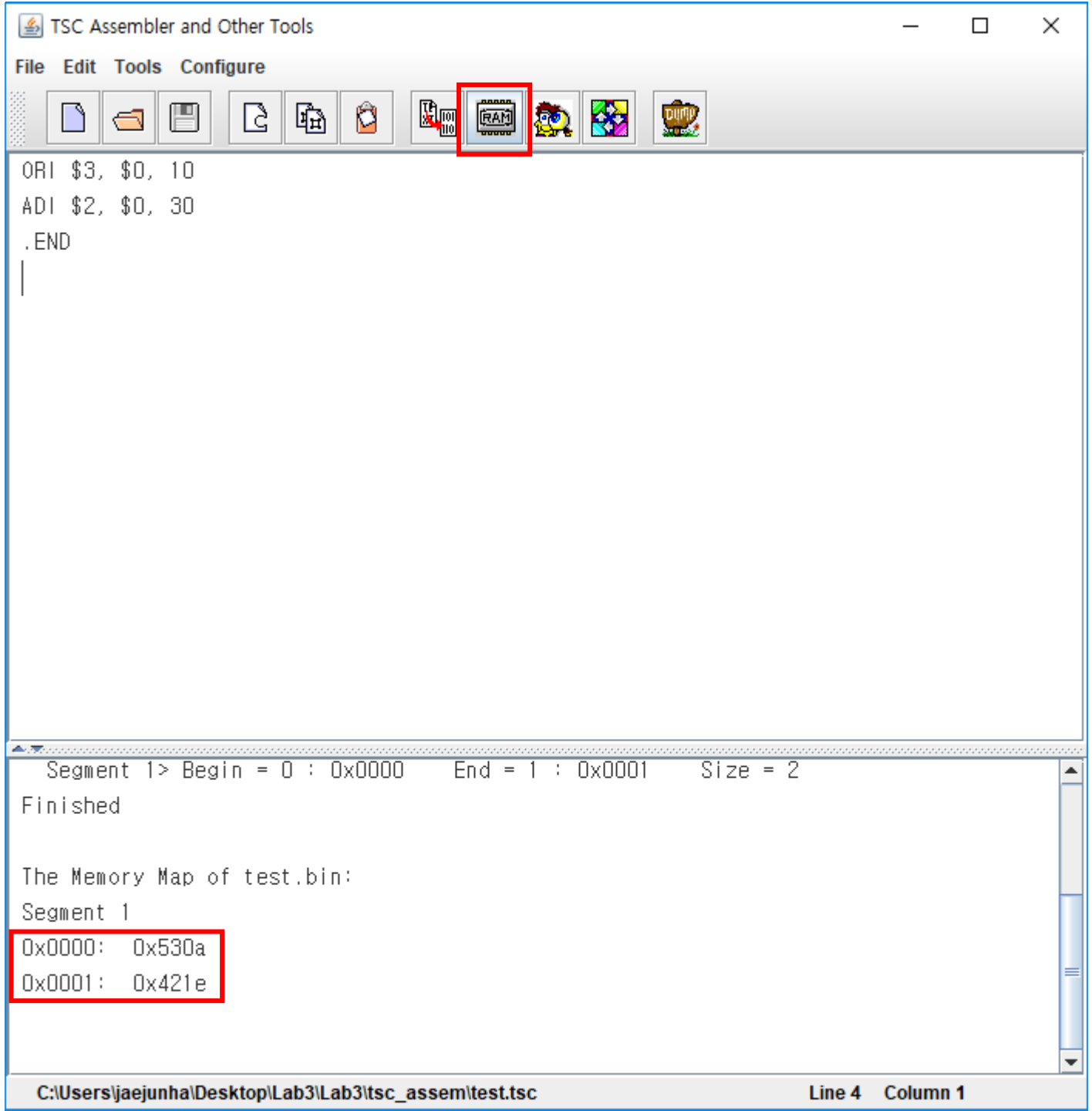
Tips

- TSC assembler



Tips

- TSC assembler

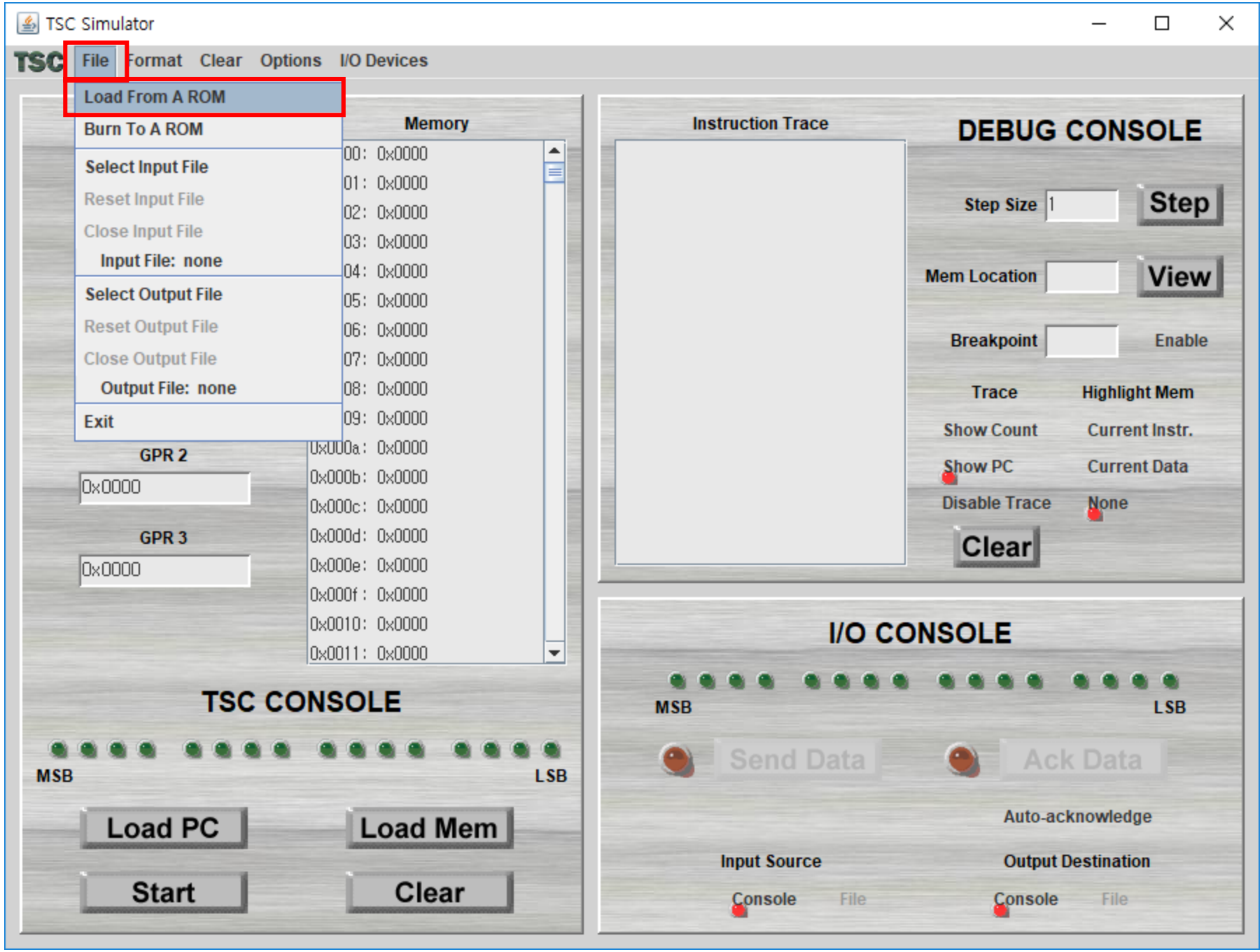


Tips

- TSC simulator
  - tsc_assem > sim > runit.bat
  - It needs Java
  - It can check value stored in memory / register

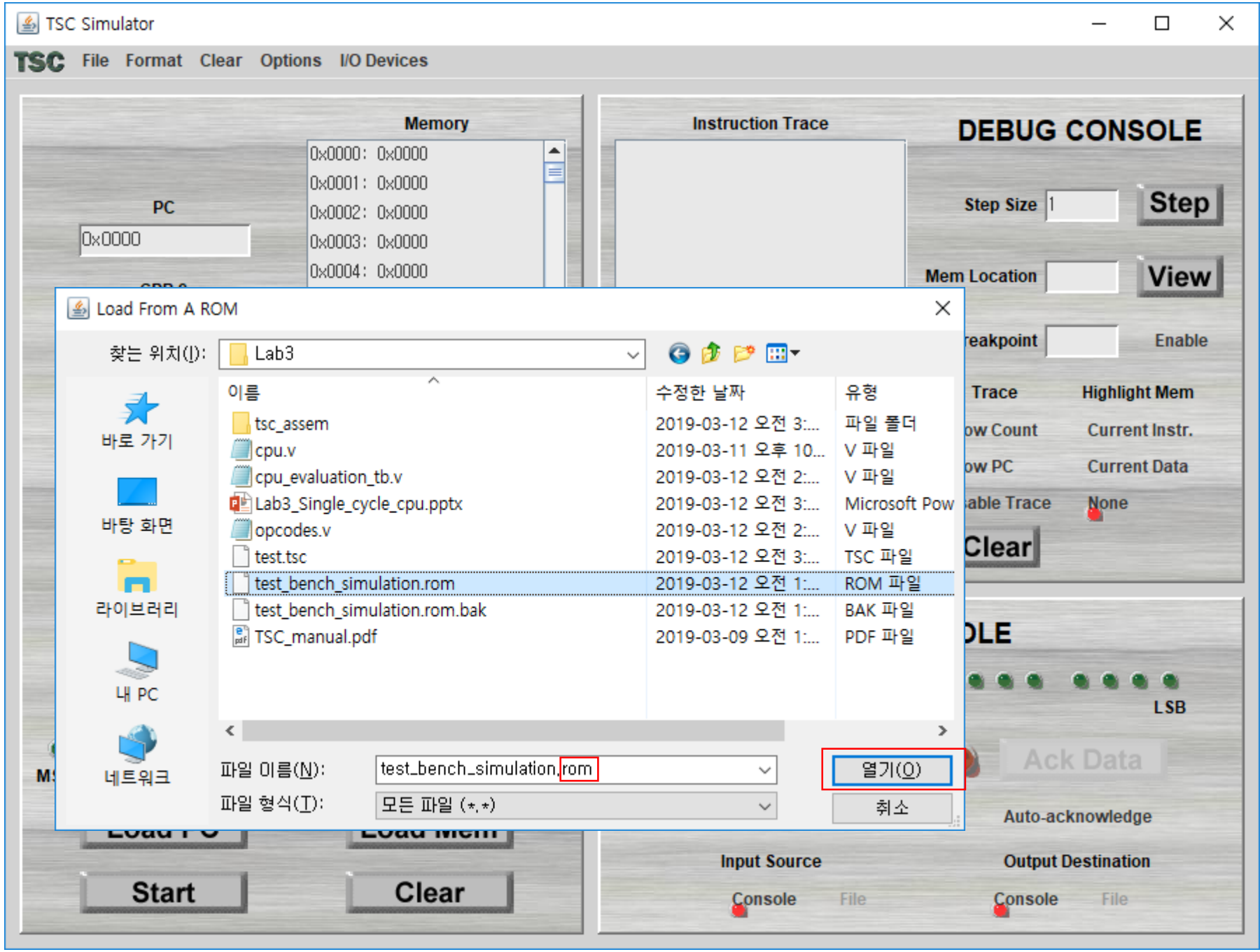
Tips

- TSC simulator



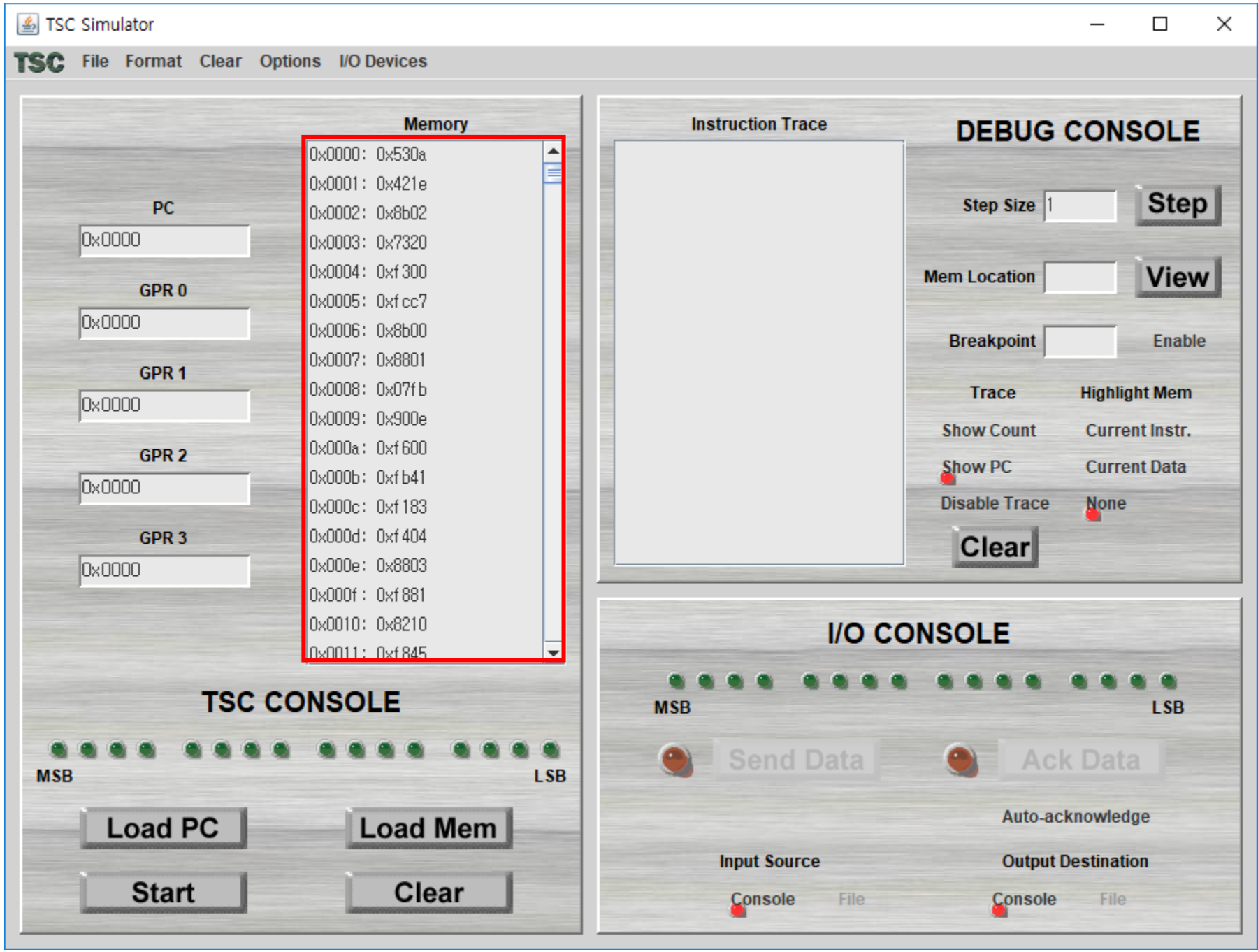
Tips

- TSC simulator



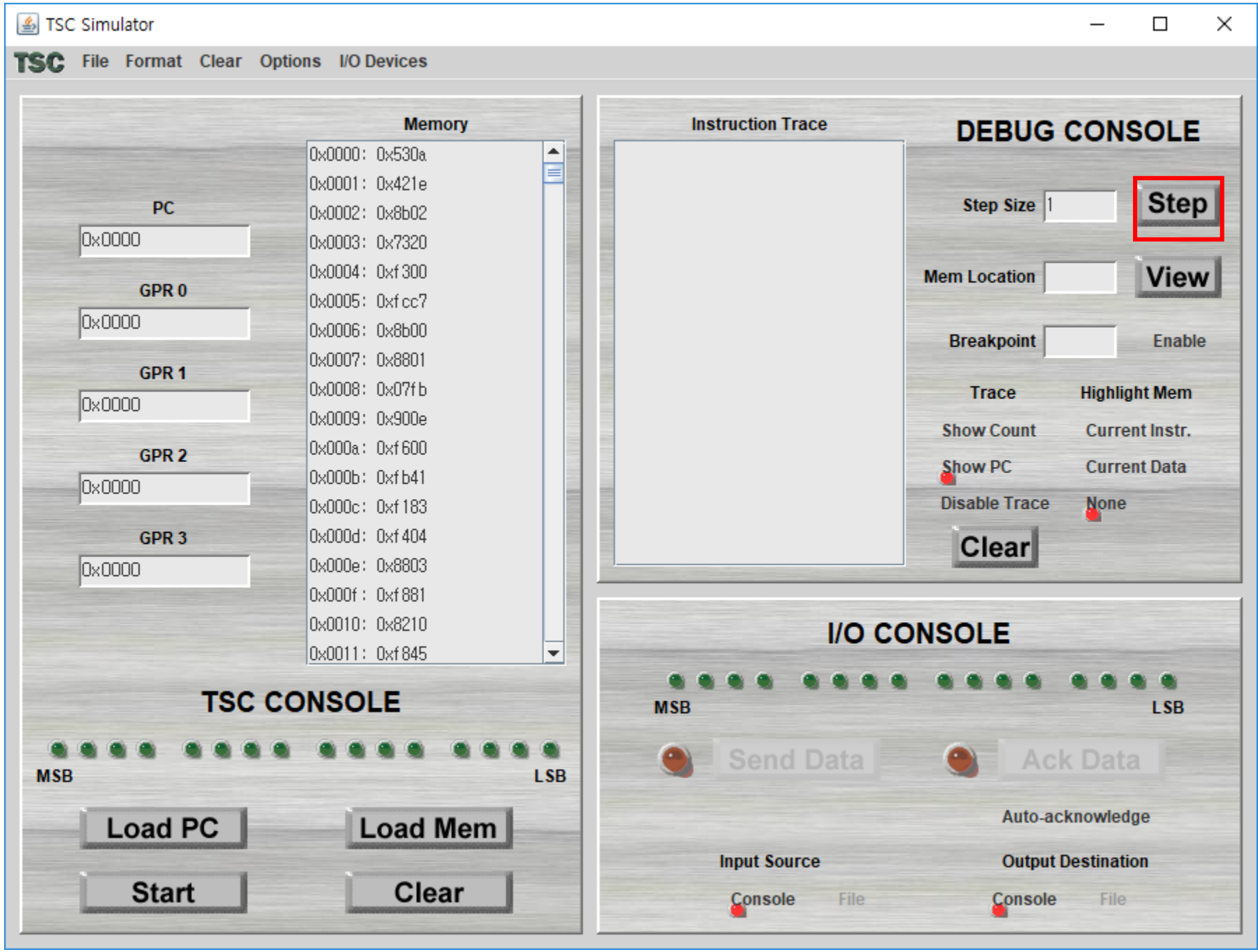
Tips

- TSC simulator



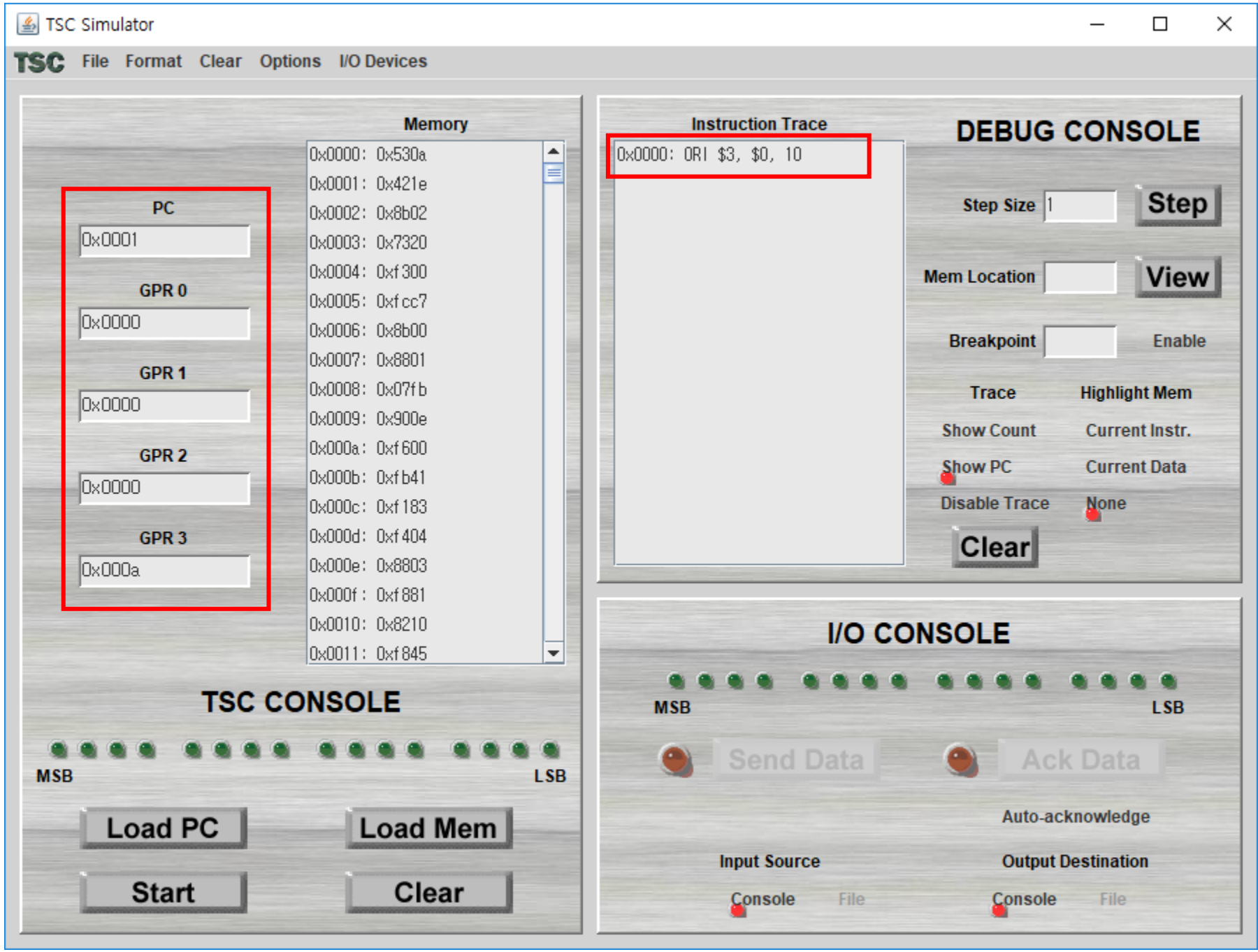
Tips

- TSC simulator



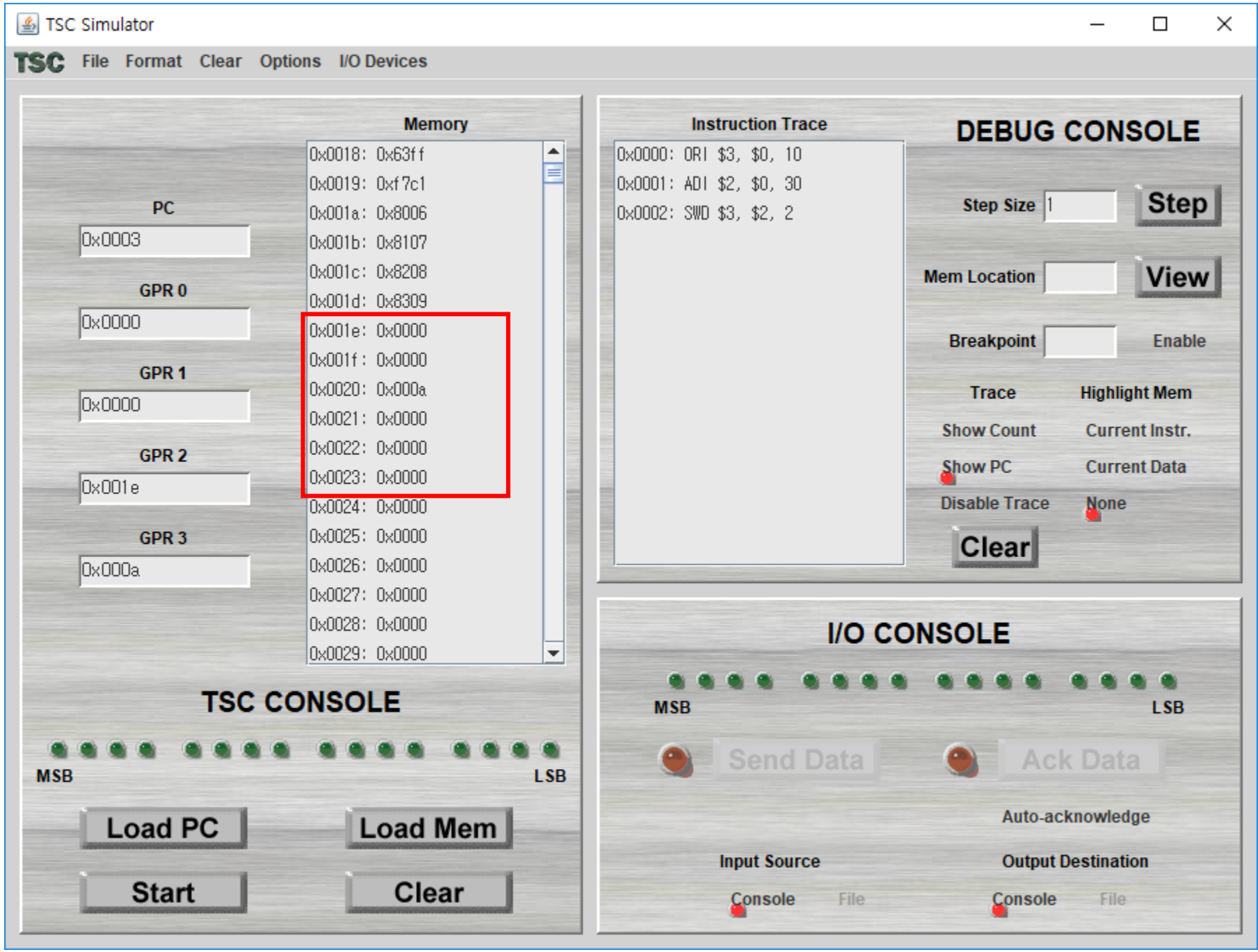
Tips

- TSC simulator



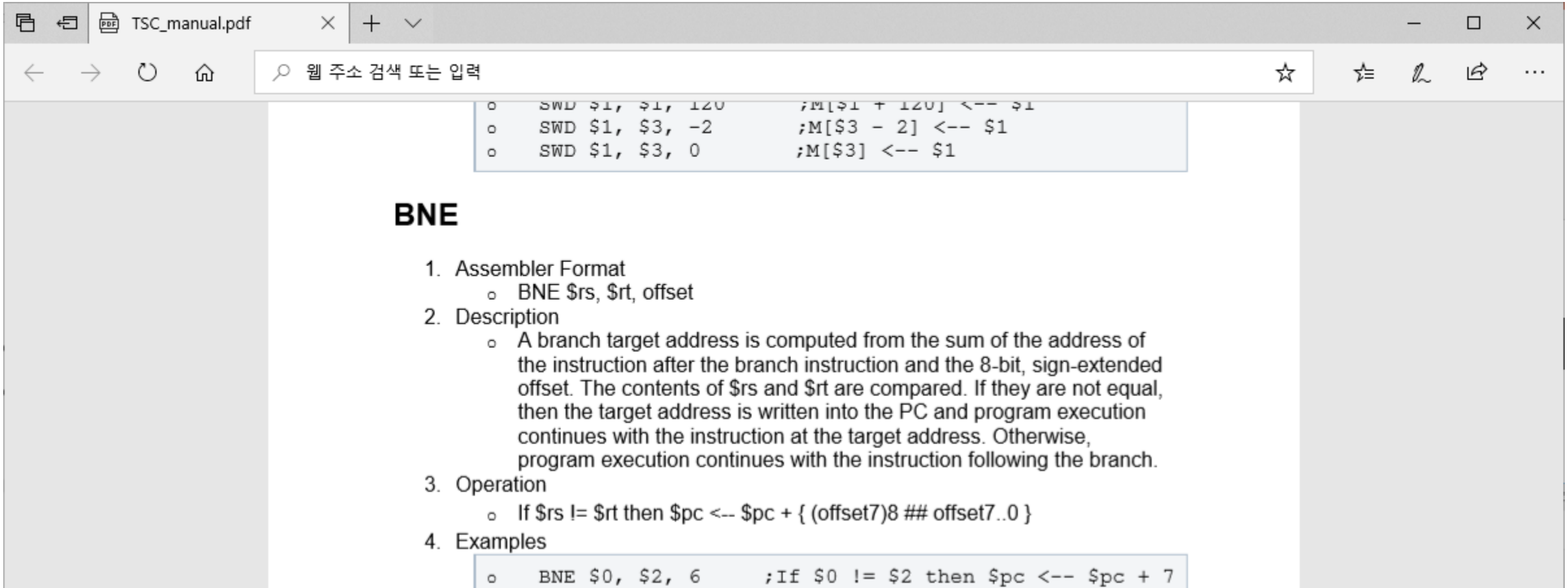
Tips

- TSC simulator



Tips

- TSC_manual.pdf



Lab2 Demo