

User GUIDE

① Open **simulator-1.0-SNAPSHOT.jar**

② Click Button **[IPL]**, and the Simulator is initialized.

If you want to use the simulator, you must click IPL first.

How to Test Floating Point Instructions/Vector Operations

Here are some instructions:

✧ Deposit FR0 and memory[22] a floating point number first. (e.g. FR0 = 2.0, memory[22] = 3.25)

➤ FADD FR0,0,0,22 (FR0 = FR0 + memory[22])

0110110000010110

➤ FSUB FR0,0,0,22 (FR0 = FR0 – memory[22])

0111000000010110

✧ Deposit R0 0 or 1, if R0 = 0, deposit memory[23] a floating point number, if R0 = 1, deposit memory[23] a fixed point number.

➤ CNVRT R0,0,0,23 (if R0 = 0, R0 = (float->int) memory[23]; if R0 = 1, FR0 = (int->float) memory[23])

0111110000010111

✧ Deposit some values to memory and registers

FR0	2.0
Memory[20]	100
Memory[21]	200
Memory[100]	1
Memory[101]	2
Memory[200]	3
Memory[201]	4

➤ VADD FR0,0,0,20

(Vector1 starts from 100(memory[20]) and Vector2 starts from 200(memory[21]))

1000110000010100

➤ VSUB FR0,0,0,20

1001000000010100

✧ memory[24] = 3;

memory[25] = 0.25(0010000000000000->Fixed-point decimal representation);

➤ LDFR FR0,0,0,24 (FR0 = memory[24]. memory[25])

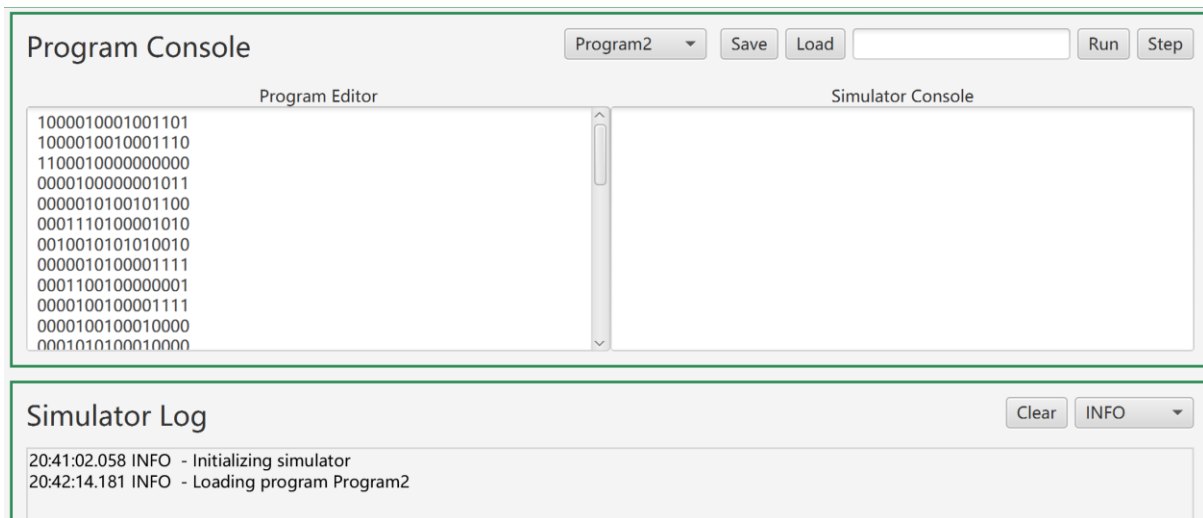
1010000000011000

➤ STFR FR0,0,0,24 (memory[24]. memory[25] = FR0)

1010010000011000

How to Run Program2 (included in Part3)

Program2 is pre-stored in a file, thus you can select Program2 in **ComboBox [programNameSelector]** and see pre-stored instructions for program 2 in **Program Editor**.



To test program 2:

① Click Button **[PreStroeMemoryForProgram2]**

That will store some values into memory, which helps run the program2;

memory[12]	0000001000000000	512	Word store start
memory[13]	0000000001000000	64	for X1
memory[14]	0000000001100000	96	for X2
memory[15]	0000000000000001	1	sentence number
memory[16]	0000000000000001	1	word number

② Click Button **[LoadProgram2]**

That will store instructions used program2 into memory [64-100] and set the PC =64.

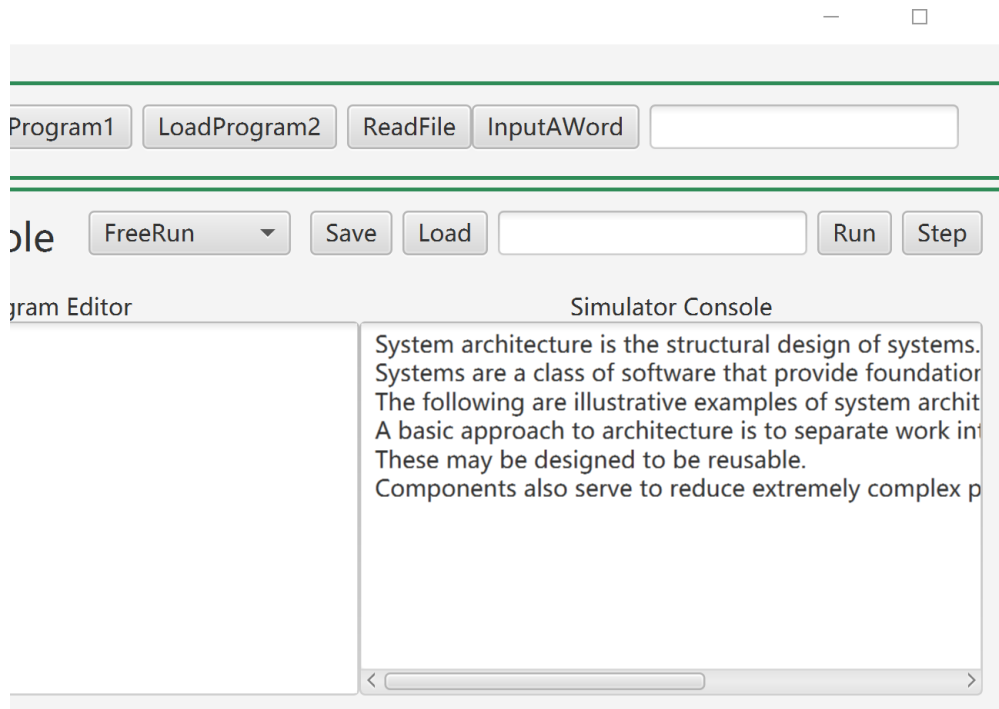
For more details of programs, please check [program2.xlsx](#)

If you want to load program2 manually, make sure program2 is stored from memory [64].

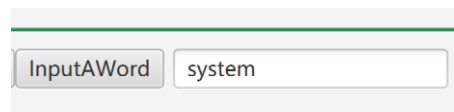
③ Click Button **[ReadFile]**

Our simulator will read "program2_paragraph.txt" and store each word into memory automatically.

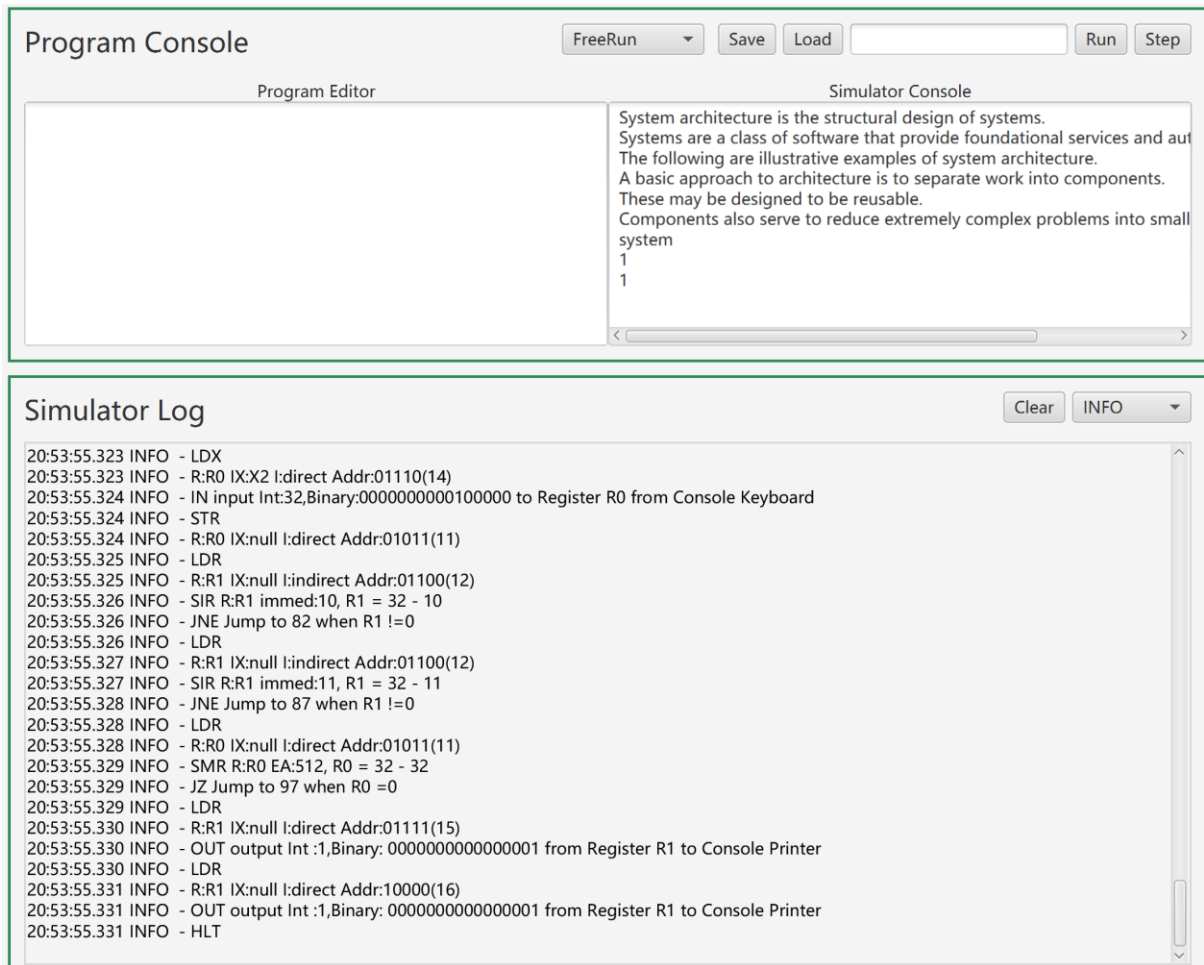
("simulator-1.0-SNAPSHOT.jar" and "program2_paragraph.txt" should be in the same folder.)



- ④ Input a word here, such as “system,” then click **[InputAWord]**.



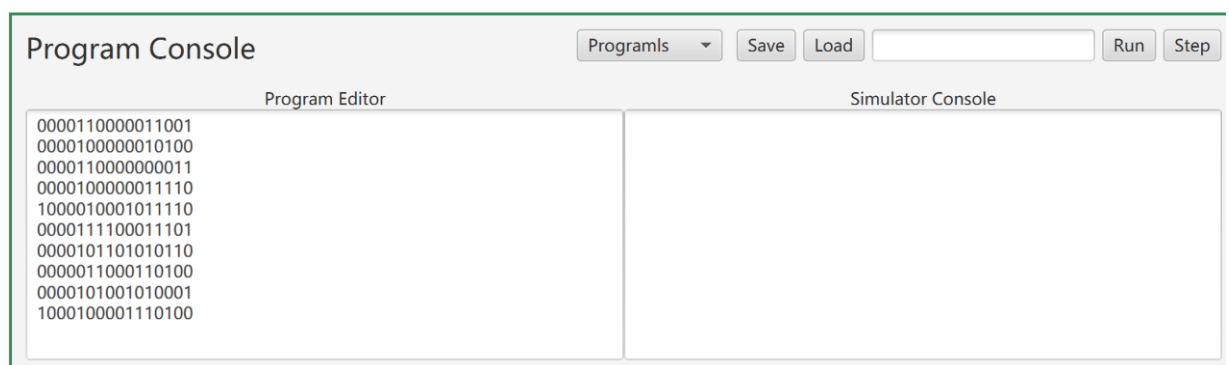
- ⑤ Then click **[Run]** or **[Step]** to run the program2. After executing instructions orderly, the Simulator Log will output information about what our simulator has done.
- ⑥ The word, sentence number and word number will output in Simulator Console.



A quick test on Instructions (included in Part2)

We have pre-stored a “Load and Store Test Program” supporting quick experiments.

- ① Select **[Programls]**, it will be loaded automatically.



- ② Input a number in the *TextField* if you need it.
- ③ Click Button **[Load]**, and then the “**programls**” will be loaded to memory start from StartIndex(input in last step) or Default Location(no input), which is 32.

- ④ Then click **[Run]** or **[Step]** to run it.

Test instructions freely (included in Part2)

You can use instructions freely by choosing **[FreeRun]** and set your own instructions.

- ① Input an instruction in Program Editor.
Such as "0010000000011111" (JZ R0,0,31)
- ② Input in *TextField* an memory index where the instruction will be sored. E.g., we store it in memory[20]

The screenshot shows a software interface for a simulator. The top section is titled "Program Console" and contains a "Program Editor" on the left with the text "0010000000011111" and a "Simulator Console" on the right which is empty. Above the editor is a dropdown menu set to "FreeRun", and buttons for "Save", "Load", "Run", and "Step". A text field containing "20" is located between the "Load" and "Run" buttons. The bottom section is titled "Simulator Log" and contains a list of log entries. A "Clear" button and a dropdown menu set to "ALL" are located at the top right of the log area.

Program Console

FreeRun Save Load 20 Run Step

Program Editor Simulator Console

0010000000011111

Simulator Log

Clear ALL

20:38:34.279 INFO - Saving the contents of program FreeRun
20:38:35.146 INFO - Loading Program
20:38:35.146 INFO - Setting Line: 0010000000011111
20:38:35.147 DEBUG - Storing 8223(0010000000011111) to memory index 20

- ③ Click **[Run]** or **[Step]**, this instruction is executed

20:38:56.892 INFO - Running Program
20:38:56.893 DEBUG - Fetching 8223(0010000000011111) from memory index 20
20:38:56.893 INFO - JZ Jump to 31 when R0 =0
20:38:56.893 DEBUG - Fetching 0(0000000000000000) from memory index 31
20:38:56.893 INFO - HLT

Tests for Machine Fault and Trap (included in Part3)

- 0 Illegal Memory Address to Reserved Locations **MFR set to binary 0001**
- 1 Illegal TRAP code **MFR set to binary 0010**
- 2 Illegal Operation Code **MFR set to 0100**
- 3 Illegal Memory Address beyond 2048 (memory installed) **MFR set to binary 1000**

① Illegal Memory Address to Reserved Locations

STR R0,0,2 0000100000000010

Memory[2] is a reserved location, so it will throw IllegalMemoryAccess Exception.

Registers

Register Name	Description	Binary	Decimal
PC	Program Counter	000000100001	33
CC	Condition Code	0000	0
IR	Instruction Register	0000000000000000	0
MAR	Memory Address Register	0000000000000010	2
MBR	Memory Buffer Register	0000000000100001	33
MFR	Machine Fault Register	0000000000000001	1
R0	General Purpose Register	0000000000000000	0

Memory

Index	Binary	Decimal
0	000001111010000	2000
1	00000000000000110	6
2	00000000000100001	33
3	0000000000000000	0
4	0000000000000000	0

Program Console

0000100000000010

Simulator Log

```

17:24:04.953 INFO - Initializing simulator
17:24:04.958 INFO - Setting default exception table at memory location 2000 for 16 values -> 6
17:24:04.959 INFO - Setting default machine fault pointer to 6
17:24:06.552 INFO - Saving the contents of program FreeRun
17:24:07.889 INFO - Loading Program
17:24:08.880 ERROR - IllegalMemoryAccess: Will not store/fetch index: 2 from reserved memory location 0-5
17:24:08.881 INFO - Error routine pointing to 1
17:24:08.882 INFO - Exception occurred, executing fault location: 1
17:24:08.882 INFO - HLT
17:24:08.882 INFO - Finished trap routine, resuming to 33
    
```

② Illegal TRAP code

01111000000000000 TRAP 0

We don't have routine 0, so Trap 0 will throw IllegalTrapCode Exception.

Registers

Register Name	Description	Binary	Decimal
PC	Program Counter	000000100001	33
CC	Condition Code	0000	0
IR	Instruction Register	00000000000000110	6
MAR	Memory Address Register	0000000000000010	2
MBR	Memory Buffer Register	0000000000100001	33
MFR	Machine Fault Register	0000000000000010	2
R0	General Purpose Register	0000000000000000	0

Memory

Index	Binary	Decimal
0	000001111010000	2000
1	00000000000000110	6
2	00000000000100001	33
3	0000000000000000	0
4	0000000000000000	0
5	0000000000000000	0

Program Console

0111100000000000

Simulator Log

```

17:27:25.101 INFO - Initializing simulator
17:27:25.102 INFO - Setting default exception table at memory location 2000 for 16 values -> 6
17:27:25.102 INFO - Setting default machine fault pointer to 6
17:27:32.245 INFO - Saving the contents of program FreeRun
17:27:32.840 INFO - Loading Program
17:27:33.899 INFO - TRAP
17:27:33.899 ERROR - Received trap code: illegal trap code
17:27:33.900 INFO - Error routine pointing to 0
17:27:33.900 INFO - Resolved routine at location: 2002
17:27:33.900 INFO - Executing trap routine: 2
17:27:33.901 INFO - HLT
17:27:33.901 INFO - Finished trap routine, resuming to 33
    
```

③ Illegal Operation Code

1111110000000000

“111111” is not an operation type, so it will throw IllegalOpcode Exception.

The screenshot shows the CSCI 6461 Simulator interface. The top menu bar includes 'Initialize', 'IPL', 'Input', 'PreStoreMemoryForProgram1', 'PreStoreMemoryForProgram2', 'LoadProgram1', 'LoadProgram2', 'ReadFile', and 'InputAWord'. The main window is divided into four panes:

- Registers:** A table with columns 'Register Name', 'Description', 'Binary', and 'Decimal'. The registers shown are PC (Program Counter), CC (Condition Code), IR (Instruction Register), MAR (Memory Address Register), MBR (Memory Buffer Register), MFR (Machine Fault Register), and RO (General Purpose Register).
- Program Console:** A pane with a 'FreeRun' dropdown, 'Save', 'Load', 'Run', and 'Step' buttons. It contains a 'Program Editor' and a 'Simulator Console'.
- Memory:** A table with columns 'Index', 'Binary', and 'Decimal'. The index is set to 0.
- Simulator Log:** A pane with 'Clear' and 'INFO' buttons. It displays a log of events, including an error message: '17:28:15.005 ERROR - Illegal Opcode: 111111 non supported operation'.

④ Illegal Memory Address beyond 2048

Memory [21] = 2049

STR R0,1,21 0000100000110101

Memory[2049] is out of bounds, so it will throw MemoryOutOfBounds Exception.

The screenshot shows the CSCI 6461 Simulator interface. The top menu bar is the same as in the previous screenshot. The main window is divided into four panes:

- Registers:** A table with columns 'Register Name', 'Description', 'Binary', and 'Decimal'. The registers shown are PC (Program Counter), CC (Condition Code), IR (Instruction Register), MAR (Memory Address Register), MBR (Memory Buffer Register), MFR (Machine Fault Register), and RO (General Purpose Register).
- Program Console:** A pane with a 'FreeRun' dropdown, 'Save', 'Load', 'Run', and 'Step' buttons. It contains a 'Program Editor' and a 'Simulator Console'.
- Memory:** A table with columns 'Index', 'Binary', and 'Decimal'. The index is set to 21.
- Simulator Log:** A pane with 'Clear' and 'INFO' buttons. It displays a log of events, including an error message: '17:30:16.045 ERROR - MemoryOutOfBounds: Will not store/fetch from index: 2049, greater than max memory: 2047'.

⑤ Trap

0111100000000001 Trap 1: IllegalMemoryAccess

Program Console

FreeRun Save Load Run Step

Program Editor Simulator Console

0111100000000001

Simulator Log

Clear INFO

17:33:56.014 INFO - Initializing simulator
17:33:56.015 INFO - Setting default exception table at memory location 2000 for 16 values -> 6
17:33:56.015 INFO - Setting default machine fault pointer to 6
17:33:57.972 INFO - Saving the contents of program FreeRun
17:33:58.420 INFO - Loading Program
17:33:59.646 INFO - TRAP
17:33:59.647 ERROR - Received trap code: illegal memory access
17:33:59.647 INFO - Error routine pointing to 0
17:33:59.647 INFO - Resolved routine at location: 2001
17:33:59.648 INFO - Executing trap routine: 1
17:33:59.648 INFO - HLT
17:33:59.648 INFO - Finished trap routine, resuming to 33

0111100000000010 Trap 2: IllegalTrapCode

Program Console

FreeRun Save Load Run Step

Program Editor Simulator Console

0111100000000010

Simulator Log

Clear INFO

17:34:47.605 INFO - Initializing simulator
17:34:47.606 INFO - Setting default exception table at memory location 2000 for 16 values -> 6
17:34:47.606 INFO - Setting default machine fault pointer to 6
17:34:49.181 INFO - Saving the contents of program FreeRun
17:34:49.630 INFO - Loading Program
17:34:50.341 INFO - TRAP
17:34:50.341 ERROR - Received trap code: illegal trap code
17:34:50.341 INFO - Error routine pointing to 0
17:34:50.341 INFO - Resolved routine at location: 2002
17:34:50.341 INFO - Executing trap routine: 2
17:34:50.342 INFO - HLT
17:34:50.342 INFO - Finished trap routine, resuming to 33

0111100000000011 Trap 3: IllegalValue

Program Console

FreeRun Save Load Run Step

Program Editor

0111100000000011

Simulator Console

Simulator Log

Clear INFO

```
17:35:24.639 INFO - Initializing simulator
17:35:24.640 INFO - Setting default exception table at memory location 2000 for 16 values -> 6
17:35:24.640 INFO - Setting default machine fault pointer to 6
17:35:26.137 INFO - Saving the contents of program FreeRun
17:35:26.586 INFO - Loading Program
17:35:27.283 INFO - TRAP
17:35:27.283 ERROR - Received trap code: illegal values
17:35:27.284 INFO - Error routine pointing to 0
17:35:27.284 INFO - Resolved routine at location: 2003
17:35:27.284 INFO - Executing trap routine: 3
17:35:27.284 INFO - HLT
17:35:27.284 INFO - Finished trap routine, resuming to 33
```

01111000000000100 Trap 4: IllegalOpcode

Program Console

FreeRun Save Load Run Step

Program Editor

01111000000000100

Simulator Console

Simulator Log

Clear INFO

```
17:36:23.938 INFO - Initializing simulator
17:36:23.938 INFO - Setting default exception table at memory location 2000 for 16 values -> 6
17:36:23.938 INFO - Setting default machine fault pointer to 6
17:36:25.153 INFO - Saving the contents of program FreeRun
17:36:25.607 INFO - Loading Program
17:36:26.244 INFO - TRAP
17:36:26.244 ERROR - Received trap code: illegal op code
17:36:26.244 INFO - Error routine pointing to 0
17:36:26.245 INFO - Resolved routine at location: 2004
17:36:26.245 INFO - Executing trap routine: 4
17:36:26.245 INFO - HLT
17:36:26.245 INFO - Finished trap routine, resuming to 33
```

0111100000000101 Trap 5: IllegalRegisterAccess

Program Console

FreeRun

Save

Load

Run

Step

Program Editor

0111100000000101

Simulator Console

Simulator Log

Clear

INFO

```
17:36:59.365 INFO - Initializing simulator
17:36:59.366 INFO - Setting default exception table at memory location 2000 for 16 values -> 6
17:36:59.366 INFO - Setting default machine fault pointer to 6
17:37:01.029 INFO - Saving the contents of program FreeRun
17:37:01.456 INFO - Loading Program
17:37:02.077 INFO - TRAP
17:37:02.077 ERROR - Received trap code: illegal register access
17:37:02.077 INFO - Error routine pointing to 0
17:37:02.077 INFO - Resolved routine at location: 2005
17:37:02.077 INFO - Executing trap routine: 5
17:37:02.077 INFO - HLT
17:37:02.078 INFO - Finished trap routine, resuming to 33
```

0111100000001000 Trap 8: MemoryOutOfBounds

Program Console

FreeRun

Save

Load

Run

Step

Program Editor

0111100000001000

Simulator Console

Simulator Log

Clear

INFO

```
17:37:28.198 INFO - Initializing simulator
17:37:28.199 INFO - Setting default exception table at memory location 2000 for 16 values -> 6
17:37:28.200 INFO - Setting default machine fault pointer to 6
17:37:31.505 INFO - Saving the contents of program FreeRun
17:37:32.092 INFO - Loading Program
17:37:33.205 INFO - TRAP
17:37:33.205 ERROR - Received trap code: memory out of bounds
17:37:33.205 INFO - Error routine pointing to 0
17:37:33.205 INFO - Resolved routine at location: 2008
17:37:33.205 INFO - Executing trap routine: 8
17:37:33.206 INFO - HLT
17:37:33.206 INFO - Finished trap routine, resuming to 33
```