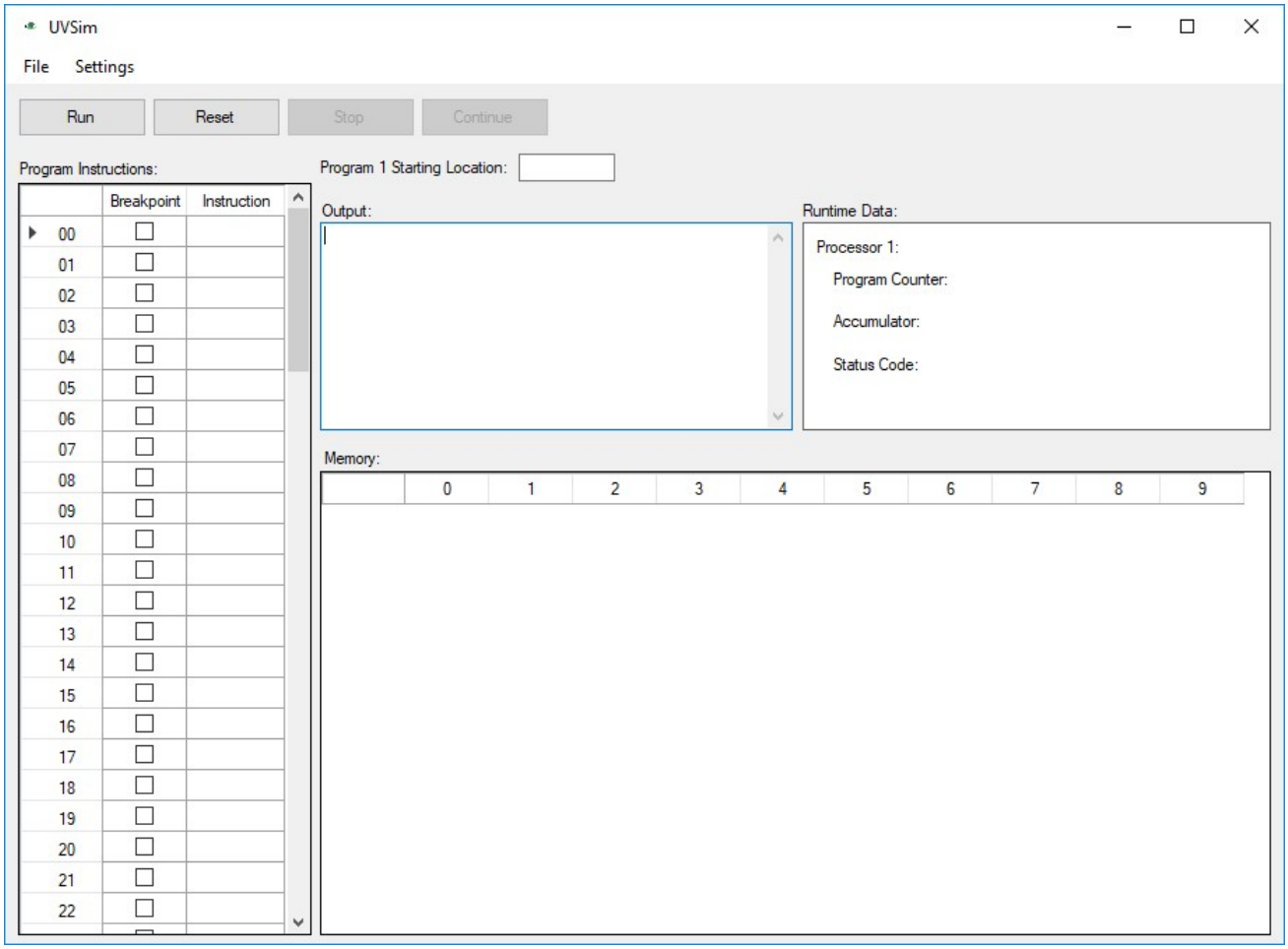
UVSim

Description

Basic processor for use with education of computer architecture. Runs the Basic Machine Language (BasicML) ISA.

Contributors:

- Riley G
- Caiden K
- Emily S
- Brad V



Installation

The UVSim is a self-contained binary application that can be run without the need to install library files or configure registry keys. To run the application, copy and paste the UVSim.exe application file to a desired location and double-click to execute.

To run from source, open the UVSIM.sln file from Visual Studio 2017 and build or use the following MSBuild command

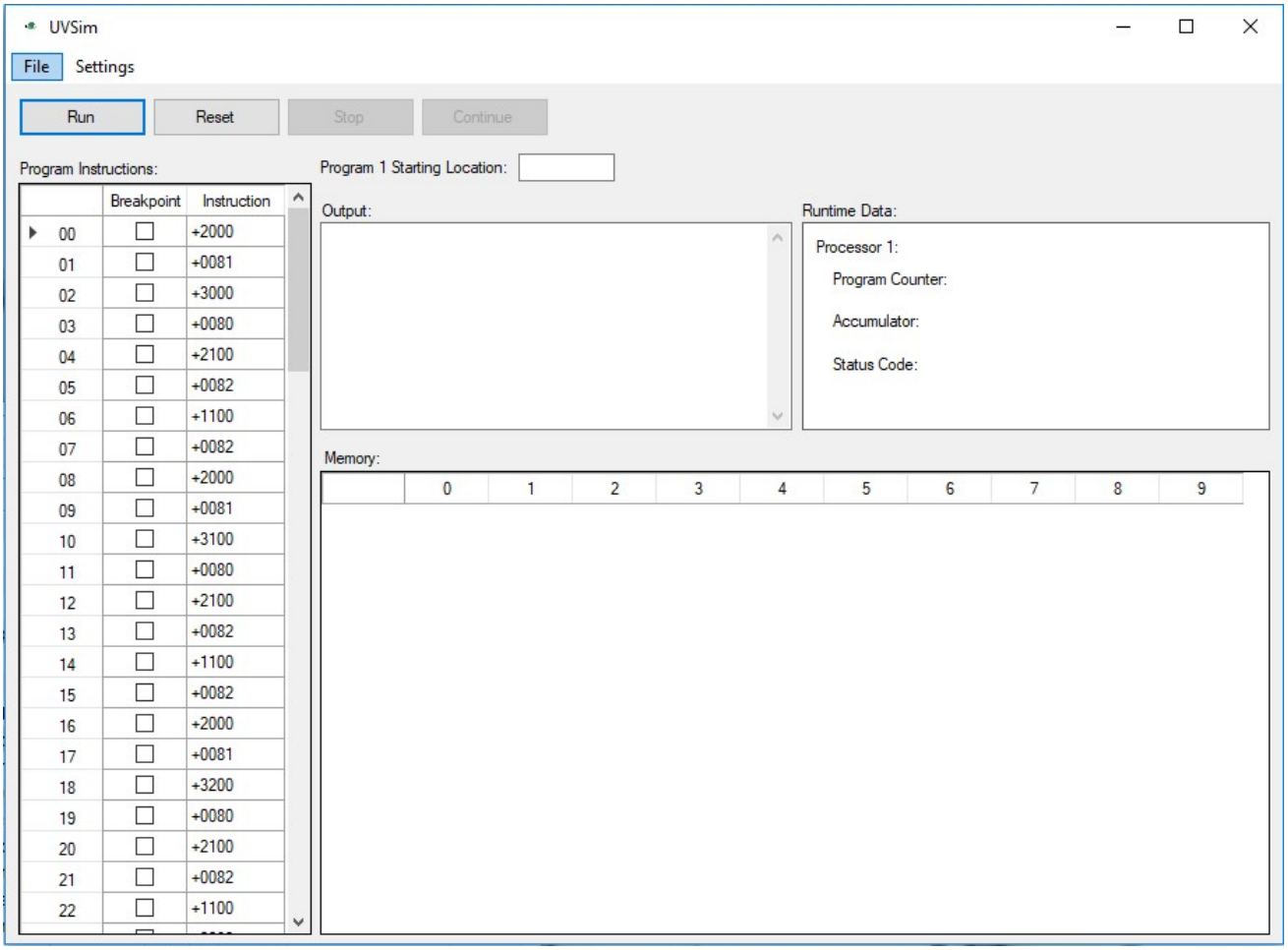
```
msbuild UVSIM.sln /target:exe
```

Once complete, you should have the UVSim.exe application binary to run.

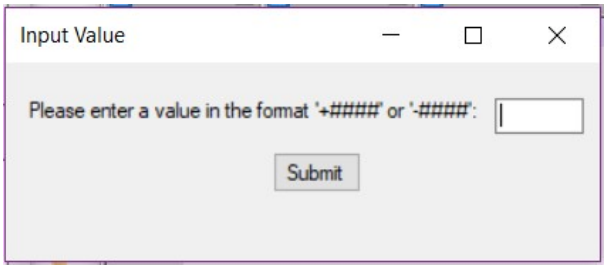
Entering Instructions

Manually Typing in

In order to run the processor, program instructions must be entered. With the application running, Enter into a memory address space and enter a valid data (e.g. `+1002`).



Input is strictly limited to entering the sign of the data (+/-) followed by four digits.

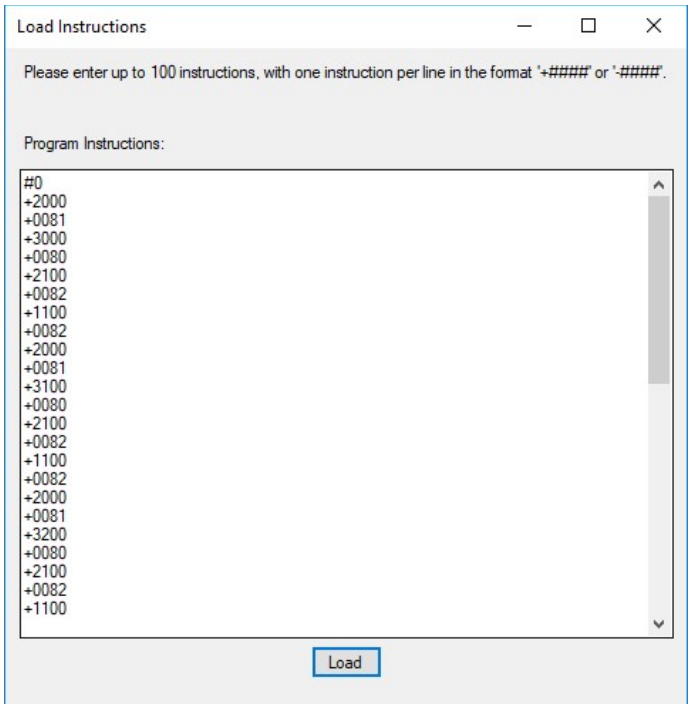


Batch Load

Pressing `CTRL + L` opens the load commands window.

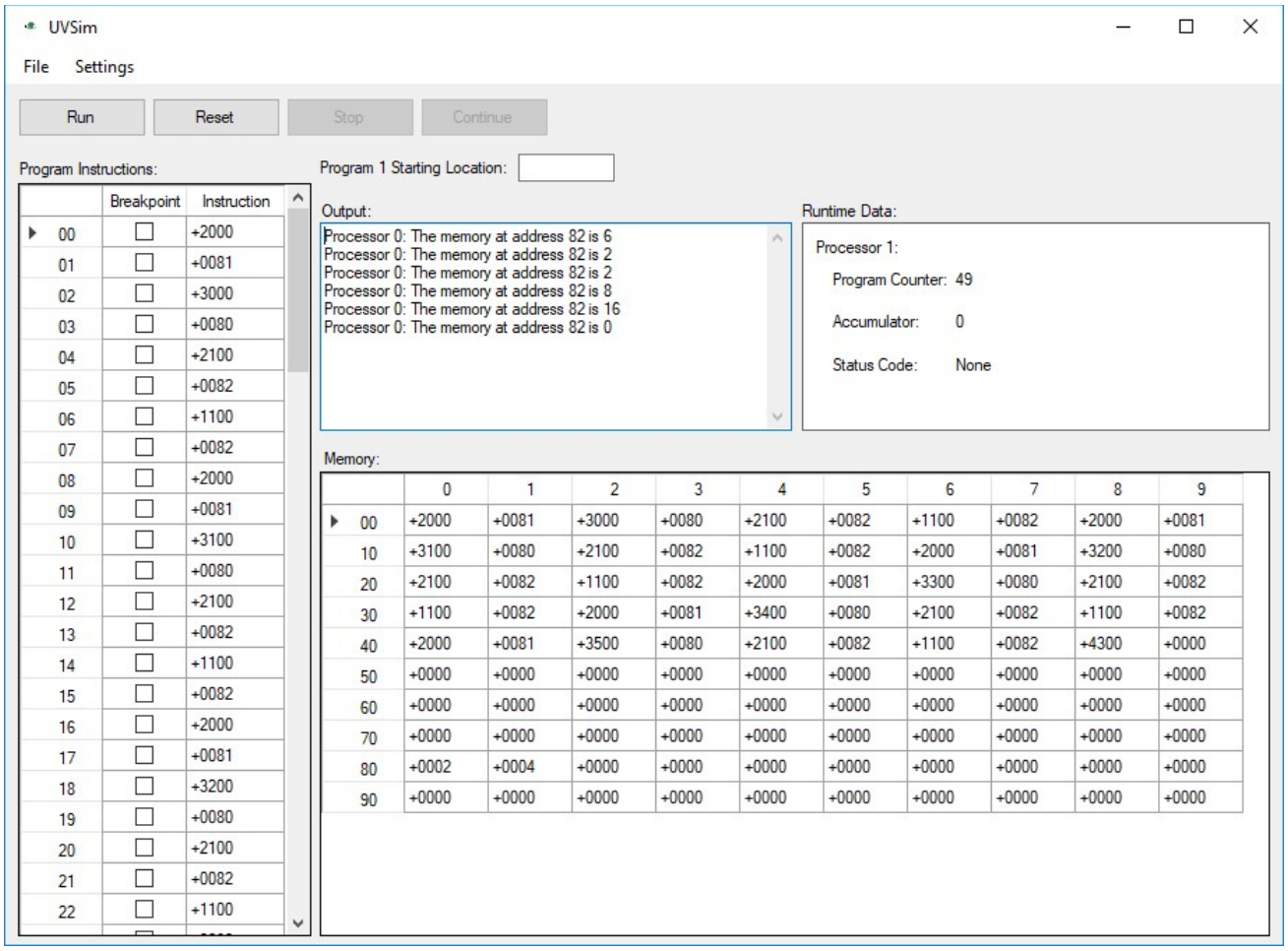
You may copy and paste your entire script in here. Press the "Load" button to load the instructions into memory.

The Symbol # is added in this mode. The pound key jumps to a location in memory and sequentially continues adding data till the end of file.



Running a Program

Once all instructions for your BML program has been entered in, you can then press "Run" to start the processor.



	Breakpoint	Instruction
00	☐	+2000
01	☐	+0081
02	☐	+3000
03	☐	+0080
04	☐	+2100
05	☐	+0082
06	☐	+1100
07	☐	+0082
08	☐	+2000
09	☐	+0081
10	☐	+3100
11	☐	+0080
12	☐	+2100
13	☐	+0082
14	☐	+1100
15	☐	+0082
16	☐	+2000
17	☐	+0081
18	☐	+3200
19	☐	+0080
20	☐	+2100
21	☐	+0082
22	☐	+1100

	0	1	2	3	4	5	6	7	8	9
00	+2000	+0081	+3000	+0080	+2100	+0082	+1100	+0082	+2000	+0081
10	+3100	+0080	+2100	+0082	+1100	+0082	+2000	+0081	+3200	+0080
20	+2100	+0082	+1100	+0082	+2000	+0081	+3300	+0080	+2100	+0082
30	+1100	+0082	+2000	+0081	+3400	+0080	+2100	+0082	+1100	+0082
40	+2000	+0081	+3500	+0080	+2100	+0082	+1100	+0082	+4300	+0000
50	+0000	+0000	+0000	+0000	+0000	+0000	+0000	+0000	+0000	+0000
60	+0000	+0000	+0000	+0000	+0000	+0000	+0000	+0000	+0000	+0000
70	+0000	+0000	+0000	+0000	+0000	+0000	+0000	+0000	+0000	+0000
80	+0002	+0004	+0000	+0000	+0000	+0000	+0000	+0000	+0000	+0000
90	+0000	+0000	+0000	+0000	+0000	+0000	+0000	+0000	+0000	+0000

Instruction set

Instructions have three key parts A + or a - sign followed by two digits(the opcode) followed by two more digits(the operand)

Read

```
Opcode                          -       10
Operand use                     -       None
Next instruction use    -       Location to store the read in data
Notes                           -       Takes input from the user and stores it in memory at a given location
```

Write

```
Opcode                          -       11
Operand use                     -       None
Next instruction use    -       Location of data to write
Notes                           -       Takes a given location and outputs that locations data to the user
```

Load

```
Opcode                          -       20
Operand use                     -       none
Next instruction use    -       Location of data to load into Accumulator
Notes                           -       Loads a given locations data into the Accumulator
```

Store

```
Opcode                          -       21
Operand use                     -       None
Next instruction use    -       Location to store the Accumulator
Notes                           -       Stores the data within the Accumulator into the given location
```

ArithmeticLogicUnit Notes

When performing ALU operations, values greater than what the Accumulator is set for (4 digits or 6 digits) will result in an overflow. E.g. If the Accumulator is set for 6 digits, and you add 999999 and 2 it will overflow to become 1.

Addition

```
Opcode                              -       30
Operand use                         -       None
Next instruction use    -       The location of the number to add
Notes                               -       {ACU} + {number at location} = {new ACU}
```

Subtraction

```
Opcode                              -       31
Operand use                         -       None
Next instruction use    -       The location of the number to add
Notes                               -       {ACU} - {number at location} = {new ACU}
```

Division

```
Opcode                              -       32
Operand use                         -       None
Next instruction use    -       The location of the number to add
Notes                               -       {ACU} / {number at location} = {new ACU}
```

Multiplication

```
Opcode                              -       33
Operand use                         -       None
Next instruction use    -       The location of the number to add
Notes                               -       {ACU} * {number at location} = {new ACU}
```

Exponent

```
Opcode                              -       34
Operand use                         -       None
Next instruction use    -       The location of the number to add
Notes                               -       {ACU} ^ {number at location} = {new ACU}
```

Modulo

```
Opcode                              -       35
Operand use                         -       None
Next instruction use    -       The location of the number to add
Notes                               -       {ACU} mod {number at location} = {new ACU}
```

Branch

```
Opcode                              -       40
Operand use                         -       None
Next instruction use    -       The location to branch to
Notes                               -       Jumps to a location in code
```

Branch Negative

```
Opcode                              -       41
Operand use                         -       None
Next instruction use    -       The location to branch to
Notes                               -       Jumps to a location in code if the Accumulator < 0
```

Branch Zero

```
Opcode                              -       42
Operand use                         -       None
Next instruction use    -       The location to branch to
Notes                               -       Jumps to a location in code if the Accumulator = 0
```

Halt

```
Opcode                              -       43
Operand use                         -       None
Next instruction use    -       None
Notes                               -       Stops the program
```

Support

If you encounter a bug within your application, please report them through the Issue Tracker made available at the InnovatED Source Code Repository Site.