T34 Emulator Documentation

Release 0.1

Chezka Gaddi

Nov 10, 2019

CONTENTS

CHAPTER

ONE

T34 EMULATOR TUTORIAL

This is the tutorial on how to use the T34 Emulator module.

1.1 Running the Application

1.2 Functionality

The monitor will have similar functionality as an OS. The T34 monitor has six functions;

1. *Load a Program*
2. *Display the content of a specific memory address*
3. *Display the content of a range of memory addresses*
4. *Edit memory locations*
5. *Run program starting as a specified address*
6. *Exit the program*

1.3 Load a Program

The machine can start in two modes. Either the user provided an object file (a program), if so, the program is loaded into the correct memory location, or the user just starts the emulator without any program. In both cases the monitor is started, and the user is provided with the monitor prompt (>).

To start the application with a program, run the application with the name of the object file.

```
$ python3 t34.py [filename]
```

1.4 Display the content of a specific memory address

By typing in the memory address in HEX at the Monitor prompt, the Monitor returns the byte (in HEX format) at that location.

```
> 200
 200    A9
```

1.5 Display the content of a range of memory addresses

By typing in the starting address in HEX, followed by a period and finally the ending address in HEX at the Monitor prompt, the Monitor returns the bytes between those locations.

```
> 200.20F
 200    A9 00 85 00 A5 00 8D 00
 208    80 E6 00 4C 04 02 00 00
```

1.6 Edit memory locations

By typing in the starting address in HEX, followed by a colon, and then the new values for the memory locations at the Monitor prompt, the monitor updates the current locations.

```
> 300: A9 04 85 07 A0 00 84 06 A9 A0 91 06 C8 D0 FB E6 07
> 300.310
 300    A9 04 85 07 A0 00 84 06
 308    A9 A0 91 06 C8 D0 FB E6
 310    07
```

1.7 Run program starting as a specified address

By typing in the starting address in HEX, followed by an R at the Monitor prompt. The monitor will execute all code starting at the address and up until the first BRK (opcode 00).

```
> 200R
 PC  OPC  INS   AMOD OPRND  AC XR YR SP NV-BDIZC
 200
```

1.8 Exit the program

The user should be able to exit the monitor (and python) in three ways:

1. Ctrl-C (keyboard interrupt)
2. Ctrl-D (EOF)
3. Type exit at the monitor prompt (> exit)

CHAPTER

TWO

DOCUMENTATION FOR THE CODE

2.1 Emulator – auto members

class t34.Emulator.**Emulator**(*program_name=None*)

Class to store an emulator and runs program files.

access_memory(*address*)

Accesses the memory address and displays the contents.

Parameters **address** (*str*) – HEX address of the memory to be accessed.

Returns memory content

Return type string

access_memory_range(*begin*, *end*)

Accesses a memory range and displays all the contents.

Parameters

- **begin** (*str*) – beginning HEX address of the memory to be accessed.
- **end** (*str*) – end HEX address of the memory to be accessed.

Return out contents of the memory range.

Return type string

edit_memory(*address*, *data*)

Edits the contents of a specific memory address.

Parameters

- **address** (*str*) – HEX address of the memory to be edited.
- **data** (*str*) – data to store into the memory address.

load_program()

Loads the program.

Returns successful read

Return type bool

run_program(*address*)

Runs and executes the program. It also prints out the contents of the registers.

Parameters **address** – Location of the command to be executed.

Return output Contents of all the registers.

Return type string

start_emulator()
Starts the emulator and evaluates and executes commands.

CHAPTER

THREE

TESTING THE PROGRAM

All of the functionality of the *Emulator* class is tested with the unittest found in the TestEmulator module.

3.1 Test Emulator

class t34.test_emulator.**TestEmulator**(*methodName='runTest'*)

Unit testing class for all the functionality of the Emulator class.

setUp()

Setup the Emulator object to be used for all the tests.

test_access_memory()

Test access to a memory address.

test_access_memory_range()

Test access to a memory address range.

test_edit_memory_locations()

Test edit of a memory location.

test_run_program()

Test running the program command.

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX