T34 Emulator Documentation

CSC 317

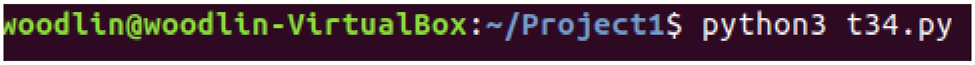
Woodlin Smith

1. Description

This emulator will allow the user to run programs designed for the T34 processor. It will also provide a “monitor” functionality the allows the user to examine and edit memory addresses. This emulator is written in Python 3.7. Currently, the monitor functionality and instruction execution is implemented. A user can add instructions in Opcode format to memory. **All instructions with Implied, Accumulator, Immediate, Zeropage, Indirect, Relative, and Absolute addressing modes are implemented.**

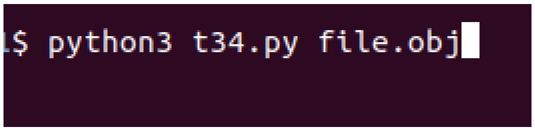
2. Usage

The T34 emulator is started from the command line. The general usage is:



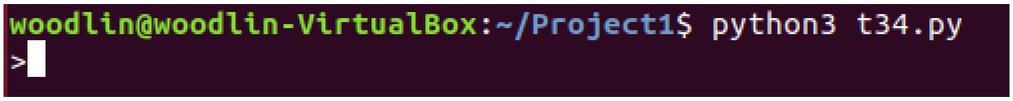
```
woodlin@woodlin-VirtualBox:~/Project1$ python3 t34.py
```

This will start the monitor, and initialize every memory location to 0. The user can also start the program with an Intel HEX file. This will load all the relevant data into the correct memory addresses.



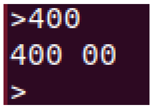
```
1$ python3 t34.py file.obj
```

In either case, the user is then presented with the monitor prompt. From there, the user has a number of options.



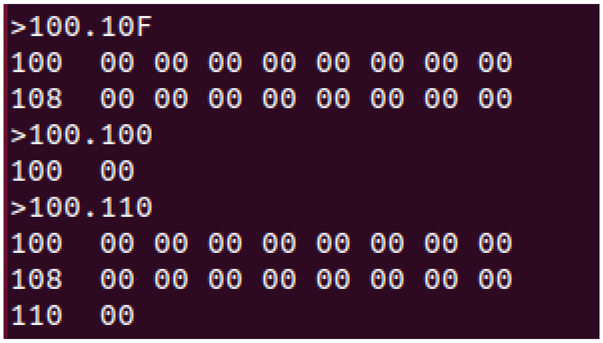
```
woodlin@woodlin-VirtualBox:~/Project1$ python3 t34.py
>
```

Inputting a hex number will display the value stored at that memory address (in this case 0, as it was opened without a file) and then prompt the user again.



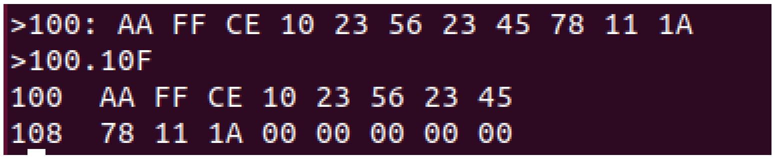
```
>400
400 00
>
```

The user can display a range of addresses by inputting 2 hex numbers separated by a period.



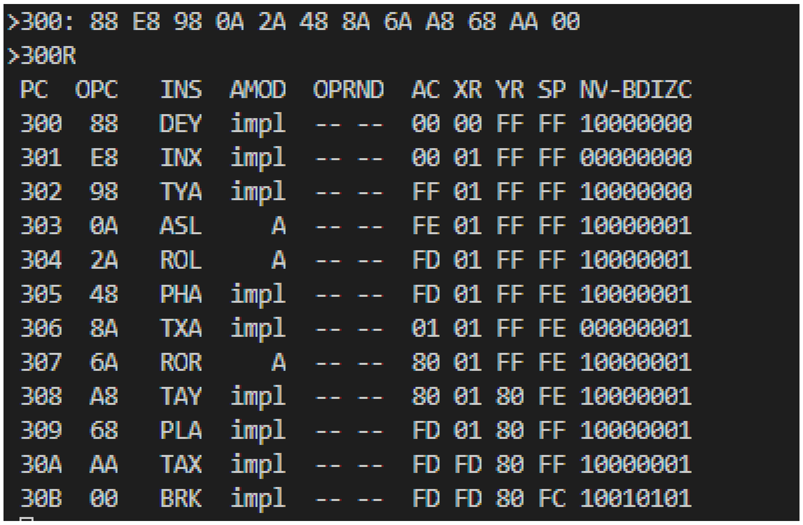
```
>100.10F
100  00 00 00 00 00 00 00 00
108  00 00 00 00 00 00 00 00
>100.100
100  00
>100.110
100  00 00 00 00 00 00 00 00
108  00 00 00 00 00 00 00 00
110  00
```

The user can also edit values stored at a specific locations by inputting the starting address followed by a colon and then the desired new values.



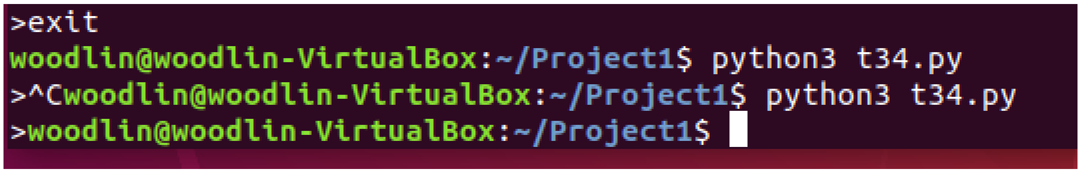
```
>100: AA FF CE 10 23 56 23 45 78 11 1A
>100.10F
100  AA FF CE 10 23 56 23 45
108  78 11 1A 00 00 00 00 00
```

The user can also run programs for the T34. It will start executing instructions and continue until it reaches a break (0x00). At each step, it will display the contents of the various registers and the status of the program.



```
>300: 88 E8 98 0A 2A 48 8A 6A A8 68 AA 00
>300R
 PC  OPC   INS  AMOD  OPRND  AC XR YR SP NV-BDIZC
 300  88   DEY  impl  -- --  00 00 FF FF 10000000
 301  E8   INX  impl  -- --  00 01 FF FF 00000000
 302  98   TYA  impl  -- --  FF 01 FF FF 10000000
 303  0A   ASL     A  -- --  FE 01 FF FF 10000001
 304  2A   ROL     A  -- --  FD 01 FF FF 10000001
 305  48   PHA  impl  -- --  FD 01 FF FE 10000001
 306  8A   TXA  impl  -- --  01 01 FF FE 00000001
 307  6A   ROR     A  -- --  80 01 FF FE 10000001
 308  A8   TAY  impl  -- --  80 01 80 FE 10000001
 309  68   PLA  impl  -- --  FD 01 80 FF 10000001
 30A  AA   TAX  impl  -- --  FD FD 80 FF 10000001
 30B  00   BRK  impl  -- --  FD FD 80 FC 10010101
```

The user can exit 3 ways: typing exit, ctrl-C (keyboard interrupt), or ctrl-D (EOF) on Linux. It will close the program and Python.



```
>exit
woodlin@woodlin-VirtualBox:~/Project1$ python3 t34.py
>^Cwoodlin@woodlin-VirtualBox:~/Project1$ python3 t34.py
>woodlin@woodlin-VirtualBox:~/Project1$ 
```

2. Testing

Functions in the utils.py file were unit tested with Python’s unit testing framework in tests.py. As other functions were generally larger, they were system-level tests.

3. Functions

1. t34.py

main():

Author: Woodlin Smith

Description: Serves as the main processing loop for the program.

2. system.py

__init__(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: The constructor builds the core of the emulator.

It initializes the registers and main memory, and sets the base

address for the memory to be above the stack.

parse_input(self, u_input):

Author: Woodlin Smith

Param: self - the emulator object

Param: u_input - the user input, as a string

Description parse_input determines what the user wants to do based on their input. It also error checks the input, doing nothing if the user inputs something invalid.

init_mem_from_file(self, input_obj_file):

Author: Woodlin Smith

Param: self - the emulator object

Param: input_obj_file - the object file that the user wants to load in

Description: Loads in an Intel HEX file, parses it line-by-line, and then loads the values into the main memory

print_address_range(self, u_input):

Author: Woodlin Smith

Parameter: self - the emulator object

Parameter: u_input - the user's input as a string

Description: Prints a range of memory addresses specified by the user

edit_mem_loc(self, u_input):

Author: Woodlin Smith

Parameter: self - the emulator object

Parameter: u_input - the user's input as a string

Description: Overwrites memory starting at the specified address

run_program(self, u_input):

author Woodlin Smith

param self - the emulator object

param u_input - the user input as a string

description For now, all this function does is clear

the registers, and then load the PC with the user's input

print_mem_address(self, u_input, str_val):

Author: Woodlin Smith

Parameter: self - the emulator object

Parameter: u_input - the user's input as a string

Parameter: str_val - the value stored at the address

Description: Prints the value stored at a specified

memory address

hex_to_fmt_string(self, value):

Author: Woodlin Smith

Parameter: self - the emulator object

Parameter: value - the integer value to be formatted

Description: Takes a base 16 number and converts it to a formatted string for printing

Return: val_string - the formatted string

print_run_table(self, u_input):

Author: Woodlin Smith

Parameter: self - the emulator object

Parameter: u_input - a parsed form of the user's input for running a program

Description: Prints a table of (mostly blank) information about registers and instructions.

__create_ins_table(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Creates the dictionary of instructions

__accum_shift_left(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Does a left shift on the Accumulator, updates flags

__rotate_accum_left(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Performs a left rotation on the accumulator

__break(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Essentially works as an interrupt. Pushes the PC and Status regs to the stack, and sets the interrupt flag

__clear_carry(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Clears the carry flag

__clear_decimal_mode(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Clears the decimal flag

__clear_interr_disable(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Clears the interrupt flag

__clear_overflow(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Clears the overflow

__decr_X(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Decrements the X register by 1 and updates flags

__decr_Y(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Decrements the Y register by 1 and updates flags

__incr_x(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Increments the X register by 1 and updates flags

__incr_y(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Increments the Y register by 1 and updates flags

__shift_accum_right(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Does a right shift on the accumulator and updates the flags

__no_op(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: No operation

__push_accum_stack(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Pushes the accumulator onto the stack

__pull_accum_stack(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Pulls the accumulator from the stack

__push_status_stack(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Pushes the status register onto the stack

__pull_status_stack(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Pulls the status register off of the stack

__rotate_accum_right(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Rotates the accumulator right, updates flags

__set_carry(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: sets the carry flag

__set_decimal(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: sets the decimal flag

__set_interr(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Sets the interrupt flag

__transfer_accum_x(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Transfers the accumulator into X and updates flags

__transfer_accum_y(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Transfers the accumulator into Y and updates flags

__transfer_sp_x(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Transfers the stack pointer into x and updates flags

__transfer_x_accum(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Transfers X into the accumulator and updates flags

__transfer_x_sp(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Transfers x into the stack pointer

__transfer_y_accum(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Transfers Y into the accumulator and updates the flags

__update_status_reg(self, neg, overflow, brk, decimal, interrupt, zero, carry):

Author: Woodlin Smith

Parameter: self - the emulator object

Parameter: neg - the value of the negative flag

Parameter: overflow - the value of the overflow flag

Parameter: brk - the value of the break flag

Parameter: decimal - the value of the decimal flag

Parameter: interrupt - the value of the interrupt flag

Parameter: zero - the value of the zero flag

Parameter: carry - the value of the carry flag

Description: ORs all the values together and updates the status register

__get_current_status_reg(self):

Author: Woodlin Smith

Param: self - the emulator object

Description: Gets a list of the current flags

__execute_instruction(self,opcode):

Author: Woodlin Smith

Parameter: self - the emulator object

Parameter: opcode - the opcode of the instruction to be run

Description: Gets an instruction from the table and then executes it.

__add_imm_accum(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Adds an immediate value to the accumulator

__and_imm_accum(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Does a bitwise AND between an immediate value and the accumulator

__cmp_imm_accum(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Compares an immediate value with the accumulator

__cmp_imm_x(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Compares an immediate value with the x reg

__cmp_imm_y(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Compares an immediate value with the y reg

__xor_imm_accum(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: XOR’s an immediate value with the accumulator

__load_imm_accum(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Loads the accumulator with an immediate value

__load_imm_x(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Loads an immediate value into the x register

__load_imm_y(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Loads an immediate value into the y register

__or_imm_accum(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Performs a bitwise or between the memory and the accumulator

__add_zpg_accum(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Adds a zeropage addressed value to the accumulator

__and_zpg_accum(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Performs a bitwise and between a zeropage address and the accumulator

__shift_zpg_left(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Shifts a value in the zeropage one bit left

__test_mem_accum(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Performs a test between a zeropage value and the accumulator

__cmp_zpg_accum(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Compares a zpg value with the accumulator

__cmp_zpg_x(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Compares a zpg value with the x register

__cmp_zpg_y(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Compares a zpg value with the y register

__decr_mem_zpg(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Decrements a value in the zeropage

__incr_mem_zpg(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Increments a value in the zeropage

__xor_zpg_accum(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: XOR’s a value from the zeropage with the accumulator

__load_zpg_accum(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Loads a value from the zeropage into the accumulator

__load_zpg_x(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Loads a value from the zeropage into the x register

__load_zpg_y(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Loads a value from the zeropage into the y register

__shift_zpg_right(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Performs a right shift on a value in the zeropage

__or_zpg_accum(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: ORs a zeropage value with the acumulator

__rotate_zpg_left(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Rotates a value in the zeropage left

__rotate_zpg_right(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Rotates a value in the zeropage right

__sub_zpg_accum(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Subtracts a zeropage value with borrow from the accumulator

__store_accum_zpg(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Stores the accumulator in a zeropage address

__store_x_zpg(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Stores the x register in a zeropage address

__store_y_zpg(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Stores the y register in a zeropage address

__add_abs_accum(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Adds an absolute value to the accumulator

__and_abs_accum(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Adds an absolute value to the accumulator

__shift_abs_accum(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Shifts an absolute addressed value to the left

__test_abs_accum(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Tests the bits of a value in memory against the accum

__cmp_abs_accum(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Compares a value in memory with the accumulator

__cmp_abs_x(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Compares a value in memory with the x register

__cmp_abs_y(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Compares a value in memory with the y register

__decr_mem_abs(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Decrements a value in memory by 1

__incr_mem_abs(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Increments a value in memory by 1

__xor_abs_accum(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: XORS the accumulator with a value in memory

__jmp_abs(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Sets the PC to the operands

__return_from_subroutine(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Returns from a jump

__jump_save_return(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Jumps to a new address, but saves the return address

__load_abs_accum(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Loads the accumulator with an absolute addressed value in memory

__load_abs_x(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Loads the x register with an absolute addressed value in memory

__load_abs_y(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Loads the y register with an absolute addressed value in memory

__shift_abs_right(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Performs a right shift on a value in memory

__or_abs_accum(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Performs an OR between the accumulator and a value in memory

__rotate_abs_left(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Rotates a value in memory to the left

__rotate_abs_right(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Rotates a value in memory to the right

__subtract_abs_accum(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Subtracts an absolute addressed value from the accumulator

__store_accum_abs(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Stores the accumulator at an absolute address

__store_x_abs(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Stores the x register at an absolute address

__store_y_abs(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Stores the y register at an absolute address

__jmp_ind(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Jumps to the address stored at the absolute address

__branch_carry_clear(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Adds a new offset to the PC if the carry bit is clear

__branch_carry_set(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Adds a new offset to the PC if the carry bit is set

__branch_equal(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Adds a new offset to the PC if the zero bit is set

__branch_neg(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Adds a new offset to the PC if the negative bit is set

__branch_not_zero(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Adds a new offset to the PC if the zero bit is clear

__branch_positive(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Adds a new offset to the PC if the negative bit is clear

__branch_overflow_clear(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Adds a new offset to the PC if the overflow bit is clear

__branch_overflow_set(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Adds a new offset to the PC if the overflow bit is set

__get_operands(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Parameter: opcode – the current opcode

Description: Determines how many operands to grab based on the opcode

__get_one_operand(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Gets one operand and updates the PC offset

__get_two_operands(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Gets two operands and updates the PC offset

3. utils.py

handle_command_line_args(arguments):

Author: Woodlin Smith

Parameter: arguments - a list of command line arguments

Description: Checks the validity of command line arguments, prints a usage statement if they don't work. Also sets a flag to let the emulator know whether or not to load a file into memory

Return: -1 - invalid arguments

Return: 2 - load in a file

Return: 1 - normal operation

usage_statement():

Author: Woodlin Smith

Description: Prints a usage statement

parse_hex_line(in_line):

Author: Woodlin Smith

Parameter: in_line - the line we are reading in

Description: Parses a line of an Intel HEX file for its important elements, and packages them in a tuple

Return: info_tuple - the package of all the relevant data

get_byte_count(in_line):

Author: Woodlin Smith

Parameter: in_line - the input line from the file

Description: gets how many bytes of data are in the line

Return: count - the byte count

get_addr(in_line):

Author: Woodlin Smith

Parameter: in_line - the input line from the file

Description: Gets the offset from the base address to store the data

Return: addr - the address offset

get_record_type(in_line):

Author: Woodlin Smith

Parameter: in_line - the input line from the file

Description: Gets the data record's type

Return: r_type the type

get_data(in_line, byte_count):

Author: Woodlin Smith

Parameter: in_line - the input line from the file

Parameter: byte_count - the amount of bytes

Description: - gets the data from the line

Return: "" - no data to be retrieved

Return: dat - the data to be retrieved

is_hex(user_input):

Source: https://stackoverflow.com/questions/11592261/check-if-a-string-is-hexadecimal

Parameter: user_input - the user input as a string

Description: Checks if a string is a valid hex string

Return: true - the string is a hex string

Return: false - the string is not a hex string

get_ones_comp(number):

Source: https://www.geeksforgeeks.org/find-ones-complement-integer/
Parameter: number – the number to invert

Description: Returns the 1's complement of a number

4. CurrentStatus.py

Most of the functions in this file are simply setters for data elements in the CurrentStatus class.

build_info_str(self):

Author: Woodlin Smith

Parameter: self - the emulator object

Description: Creates a formatted status string for each instruction executed.