Class Definition Doc

Simulator.py

Simulator Class – Holds the list of commands and simulator functions

read(addr) – Reads a word from keyboard into a specific register

write(addr) – Writes a word from a register to screen

load(addr) – Loads a word from a register into the accumulator

store(addr) – Stores a word from the accumulator into a specific register

add(addr) – Adds a word from a specific register with word in accumulator. Leaves result in accumulator

subtract(addr) – Subtracts a word from a specific register from the word in the accumulator. Leaves result in accumulator

divide(addr) – Divides a word in the accumulator by a word from a specific register. Leaves result in accumulator

multiply(addr) – Multiplies a word in the accumulator by a word from a specific register. Leaves result in accumulator

branch(addr) – Branches to a specific location in memory

branch_neg(addr) – Branches to a specific location in memory if accumulator is negative

branch_zero(addr) – Branches to a specific location in memory if accumulator is zero

halt() – Halts the program (by returning False)

GUI_app.py

GUI Controller Class - Controls most of the updates to the GUI.

save_file() – Saves the file

save_as()- Saves the file the file anywhere in the OS

save_operation()- Writes to the save file the state of the registers

clear_table() – Clears the table

update_table() – Places the new values in the table (updating it).

refresh_table() – Calls clear_table() and update_table().

refresh_accumulator() – Deletes the value in the accumulator and correctly displays the new one.

reset_memory() – resets the accumulator, current address, and tables.

clear_console() – Clears anything being displayed to the console box.

show_input() – Enables the user input box and submit button.

hide_input() – Disables the user input box and submit button.

highlight_reg() – Highlights the current instruction being executed

terminate() - Deletes the temp file once the operation is completed

GUI_Subwindows Class – Generates and controls all the GUI subwindows

- load_instructions()- Opens instructions input subwindow
    - open_file() –Opens a file containing instructions (.txt)
    - process()- Processes the user inputs
    - validate_input_size()- Verifies to see if user input will fit properly in registers
    - bit_conversion(instruction_list) – One way converter, 4 bit to 6 bit.
    - validate_instructions(loaded_instructions)- Checks to see if instructions are all valid
    - populate_registers(loaded_instructions)- Populators all the registers with the user’s input.
- table_edit(event) – Opens the subwindow to edit individual registers
    - edit_submit() – Processes the user input into the register
    - validate_input(user_input)- Validates if the input they entered is a valid input.
- new_window() – Opens a new window to run the program.

Style Controller Class – Controls all the color schemes

change_all_colors() – Changes all the colors to the new primary and secondary color scheme

choose_color(): Uses tkinter colorchooser to get user color scheme and calls change_all_colors().

Simulator_Controller class – Holds the GUI simulator functions.

run_cancel_control() – Controls behavior of run/cancel button

run()- Runs each line of the simulator and calls controller for the appropriate instructions

controller(instruction, addr)- Directs the simulator along with desired address (addr) to appropriate function call based on the instruction parameter.

read_console(addr) – Prepares GUI to accept user input.

submit_input() – Gets the user input, and validates and formats it. It then clears and updates the table.

console_write(addr) – Properly enables and disables the console box and writes to the console box the value in register {addr}.

halt_console() – Performs all the GUI operations necessary after halting (enabling and disabling buttons and writing to console box).

GUI.py also contains our instances of Simulator(), GUI_Controller(), GUI_Subwindows(), Simulator_Controller(), Style_Controller() and lastly our front-end GUI display using tkinter.