This is the documentation for

MIPS SIMULATOR

How to use it:

- Make sure you have the latest version of python3 installed (3.6.3).
- Install ttkthemes by typing `pip install ttkthemes` in your terminal.
- You can either download the provided `icarus-verilog.zip` and extract in in the simulator's folder or download icarus-verilog from the provided link than setup it and copy the installation folder into the simulator's folder.
  The latter is recommended.
- Run the program by going into the simulator's folder in your terminal than type `python mips-sim.py` .
- You can change how many clock cycles to play.
- Make sure that your computer-language is set to "English" before trying to copy or paste anything using the keyboard's shortcuts, other languages aren't supported.
- The input must be an assembly instruction, It must be written using the registers' names and decimal numbers.
  Both assembly instruction and registers' names are provided below.
- You can put comments in the assembly code by using "#", anything typed after the hash won't be compiled.
- The hex-instructions of the entered assembly-instructions are saved in 'out.hex' file in the simulator's folder.
- The program wasn't programmed to take wrong inputs, it will terminate if the input isn't correct.

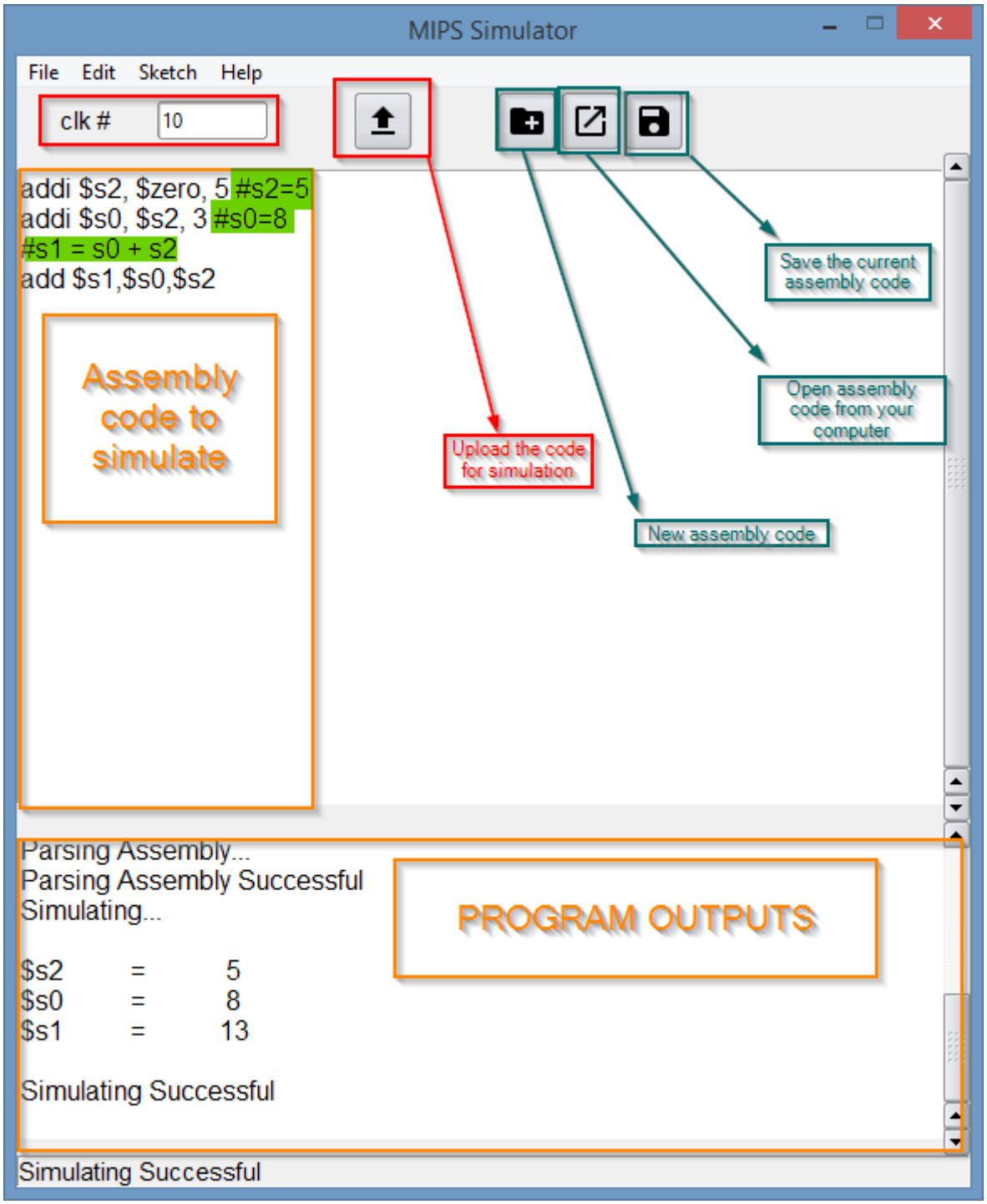
Registers:

Register Name	Register number in the mips register file
$zero	0
$at	1
$v0	2
$v1	3
$a0	4
$a1	5
$a2	6
$a3	7
$t0	8
$t1	9
$t2	10
$t3	11
$t4	12
$t5	13
$t6	14
$t7	15
$s0	16
$s1	17
$s2	18
$s3	19
$s4	20
$s5	21
$s6	22
$s7	23
$t8	24
$t9	25
$k0	26
$k1	27
$gp	28
$sp	29
$fp	30
$ra	31

Instructions:

R	add
	sub
	or
	and
	nor
	sll
	srl
	slt
I	addi
	beq
	sw
	lw
	ori

Screenshot:


MIPS Simulator
File
Edit
Sketch
Help
clk #
10
addi $s2, $zero, 5 #s2=5
addi $s0, $s2, 3 #s0=8
#s1 = s0 + s2
add $s1,$s0,$s2
Assembly code to simulate
Upload the code for simulation
Save the current assembly code
Open assembly code from your computer
New assembly code
Parsing Assembly...
Parsing Assembly Successful
Simulating...
$s2 = 5
$s0 = 8
$s1 = 13
Simulating Successful
PROGRAM OUTPUTS
Simulating Successful