Project 2 – Documentation

*This is a general documentation of the project, inside the module each function is documented explaining each argument and the functions return value.

Functions related to the simulator:

- `int execute_command(char all_commands[512][9])` - executes all of the commands
- `void add` - handels assembly command addition
- `void sub` - handels assembly command subtraction
- `void and` - handels assembly command bit-wise and
- `void or` - handels assembly command bit-wise or
- `void sll` – handels assembly command logical left shift
- `void sra` – handels assembly command arithmetic right shift
- `void mac` – handels assembly command multiplication
- `void branch` - handels assembly command branch
- `void in` – puts the value of a certain IORegister to a user chosen register
- `void out` - puts the value of a certain user register to a certain IORegister
- `void jal` - handels assembly command jump and link
- `void lw` - handels assembly command load-word
- `void sw` – assembly command store-word
- `void jr` - handels assembly command jump register
- `char* int_to_hexa_string(int num)` - converts an integer to hexadecimal string
- `int sign_extension(int num)` - sign extends a number

Functions related to Basys:

- `void Init()` - initialize all the necessary modules
- `void check_and_apply_sw_status()` - Checks the status of the SW buttons and displays text on the LCD accordingly
- `void display_sw5_sw6(int button)` - a utility function for check_and_apply_sw_status(), displays the appropriate text that depends on SW5 and SW6
- `void check_and_apply_io_status()` - Checks the status of the IORegisters and change the LD lights or SSD screen accordingly
- `void load_fib_memin_vals()` - loads fibonacci memin.txt program into an array
- `void load_timer_memin_vals()` - loads the timer memin.txt program into an array