

COMPUTER SIMULATOR MANUAL

Components:

I: Basic Machine	Design and implement the basic machine architecture. Implement a simple memory Execute Load and Store instructions Build initial user interface to simulator
II. Machine and Cache Design	Design and implement the modules for enhanced memory and cache operations. Show how the cache works via a GUI panel display of the cache. Implement all instructions except for <ul style="list-style-type: none">- CHK- Floating Point/Vector operations- Trap Extend the user interface. Demonstrate 1st program running on your simulator.

Characteristics:

- Registers
 - 4 General Purpose Registers (GPRs) – each 16 bits in length
 - The four GPRs are numbered 0-3 and are referred to as GPR0 – GPR3
 - 3 Index Registers – 16 bits in length
 - The three IXRs are numbered 0-3 and are referred to as IXR1 – IXR3
 - Program Counter (PC) – 12 bits in length
 - Instruction Register (IR) – 16 bits in length
 - Memory Address Register (MAR) – 16 bits in length
 - Memory Buffer Register (MBR) – 16 bits in length
 - Machine Fault Register (MFR) – 4 bits in length
 - Condition Code (CC) – 4 bits in length
- Simple Memory
 - 2048 words, expandable to 4096 words
 - Upon powering up the system, all elements of memory are set to zero
 - 16-bit words
 - Word addressable
- Simple Cache
 - Sits between memory and the rest of the processor
 - Fully-associative and unified
- GUI
 - Operator's Console
 - Display for all registers
 - Display for machine status and condition registers
 - Buttons
 - Load
 - Loads user-inputted instruction into memory
 - LD (10x)

- Loads user-inputted instruction into relevant register
- Store
 - Stores user-inputted instruction into memory
- IPL
 - the ROM contents are read into memory and control is transferred to the first instruction of the ROM Loader program. The ROM can be either a file on your computer or just an array of instructions in your program
- Run
 - Runs the loaded program until finished
- SS
 - Runs the next instruction of the program
- St+
 - Stores the value of MBR into MEM[MAR] and iterates the MAR
- Reset
 - Resets the computer so that memory and all registers are empty.
 - Clears program data, so new program can be loaded
- Enter
 - Sends the user input to the simulator to provide functionality for Program 1
- Test
 - Initiates the running of a command entered into the testing console at the bottom left of the GUI
- Switches (simulated as buttons) to load data into registers, to select displays, and to initiate certain conditions in the machine.
- Field Engineers Console
 - Command-line interface
 - Displays memory status after each instruction
 - Displays the execution steps of each instruction as it is executed

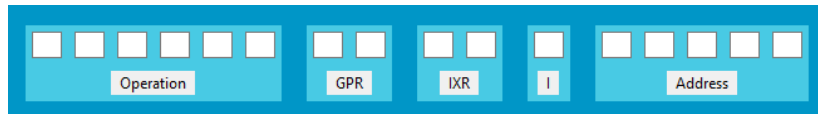
Updated 3/14/22

Implemented Instructions:

Opcode	Instruction
00	HLT
01	LDR
02	STR
03	LDA
04	AMR
05	SMR
06	AIR
07	SIR
10	JZ
11	JNE
12	JCC
13	JMA
14	JSR
15	RFS
16	SOB
17	JGE
20	MLT
21	DVD
22	TRR
23	AND
24	ORR
25	NOT
31	SRC
32	RRC
41	LDX
42	STX
61	IN
62	OUT

Quick Start Guide:

- Double-click on “main_Program1.exe” executable
- Then you have 2 options:
 - Click “IPL” Button
 - Press “Run” button to run Program 1
 - “Run” light is turned on until program finishes execution
 - When program is finished executing, “Halt” light will turn on
 - Press “SS” button to run next instruction in Program 1
 - “Run” light is turned on until instruction finishes execution
 - When instruction is finished executing, “Halt” light will turn on
 - Press “St+”
 - Manually run commands
 - Using Primary Input Console (only basic instructions)



- Enter instruction in binary
- Press “Load”, “LD”, or “Store” button
- Using Instruction Testing Console (bottom left of console)



- Input Instruction
 - Format = COMMAND, param₁, param₂, . . . , param_n
 - Example
 - Command = LOAD (LDR)
 - GPR = 01
 - IXR = 01
 - I = 0
 - Address = 01010
 - Enter “LDR, 01, 01, 0, 01010” into text box
- Press “Test” button to execute Instruction
- Press “Reset” button to restart simulator