Branden Hitt

CS3650

Simple Computer Simulation

My program acts much like you would expect a computer to. It builds all of the components, loads a program from a file into the memory of the computer, and then executes the program from memory. Of course, the computer itself is broken up into many different components.

Outside of the ISA, is the Program Loader. This is a class that simply reads in a file, then parses out the relevant data to be used in the program. First, it displays the name of the program which is given as the first line of the file. Then it takes the second line as the starting address to place the data in memory. Following that, it takes the third line to designate the address of the first instruction to be fetched and executed. Any other lines are taken as data/instructions that are prepared to be placed in memory.

The execution of the program (that is placed in memory) works in cycles. It all resides in a run() function that is called after the build and after the program is placed into memory. It begins with a fetch() instruction. This instruction places the op code on the instruction register for the current instruction. Next, an adjustPC() function is called to do just that. Since the op code takes up one slot of memory, you need to increment the PC counter one word so that it can be pointed at the next relevant piece of information. Following that, the execute() method decodes the op code and then makes the relevant calls to the correct instructions.

The ISA itself is composed of many pieces. It contains Registers R0,R1,R2,R3 along with a MemoryControl that handles the internal memory. The Bus is broken up into the various lines it needs: AddressLines, ControlLines, and DataLines. There is a Printer and Reader for input/output that communicates using the bus and register R0. For mathematical use, there is included an adder and complementor. As mentioned earlier, there is a ProgramCounter and InstructionRegister to handle some of the program flow. There is also a Status object to handle the state, as well as an MAR and MDR to help communicate data. Other than that, there is a lot of implementation of various methods/instructions that are needed to carry out a program.

The current program that runs through version 3 of the simulator follows instructions contained on a file (SimpleComputer\_Program1.txt). The program itself is simple but effective at showing off the different tasks such as reading/writing to memory and performing operations with register values. It begins by placing the program data into memory. Next, it begins running instructions such as reading in a value. The value is then moved into a register. Following that, a second value is read in and moved into a different register. The two values are added from their registers into a third register which is then displayed to the screen. Following that, the program stores a number into memory. It then decrements that value and displays it to the screen.

All of the code is included and visible at <https://github.com/Bhitt/Computer_Architecture> . Some sample input/output can be seen here:

Input File

Program Trivial Example

400

403

29 //Data

505 //Data

211 //Data

810 //READ

510 0 1 //MOVE R0, R1

810 //READ

510 0 2 //MOVE R0, R2

110 1 2 3 //ADD R1, R2, R3

510 3 0 //MOVE R3, R0

820 //PRINT

640 3 732 //LOAD R3, #732

710 3 909 //STORE R3, 909

610 0 909 //LOAD R0, 909

160 0 //DEC R0

820 //PRINT

999 //HALT

OUTPUT

Program Trivial Example

Starting Location: 403 OPCode:810

\* 40

Starting Location: 404 OPCode:510

Starting Location: 407 OPCode:810

\* 23

Starting Location: 408 OPCode:510

Starting Location: 411 OPCode:110

Starting Location: 415 OPCode:510

Starting Location: 418 OPCode:820

>> 63

Starting Location: 419 OPCode:640

Starting Location: 422 OPCode:710

Starting Location: 425 OPCode:610

Starting Location: 428 OPCode:160

Starting Location: 430 OPCode:820

>> 731

Starting Location: 431 OPCode:999

HALT() -> Program will terminate.

Memory Dump Address:400 | Content:29

Memory Dump Address:401 | Content:505

Memory Dump Address:402 | Content:211

Memory Dump Address:403 | Content:810

Memory Dump Address:404 | Content:510

Memory Dump Address:406 | Content:1

Memory Dump Address:407 | Content:810

Memory Dump Address:408 | Content:510

Memory Dump Address:410 | Content:2

Memory Dump Address:411 | Content:110

Memory Dump Address:412 | Content:1

Memory Dump Address:413 | Content:2

Memory Dump Address:414 | Content:3

Memory Dump Address:415 | Content:510

Memory Dump Address:416 | Content:3

Memory Dump Address:418 | Content:820

Memory Dump Address:419 | Content:640

Memory Dump Address:420 | Content:3

Memory Dump Address:421 | Content:732

Memory Dump Address:422 | Content:710

Memory Dump Address:423 | Content:3

Memory Dump Address:424 | Content:909

Memory Dump Address:425 | Content:610

Memory Dump Address:427 | Content:909

Memory Dump Address:428 | Content:160

Memory Dump Address:430 | Content:820

Memory Dump Address:431 | Content:999

Memory Dump Address:909 | Content:732

Branden Hitt 2018-11-20T00:43:37.302

BUILD SUCCESSFUL (total time: 9 seconds)

**SOURCE CODE** Included in following pages