Least Requirements CPU

Kian Holden

This machine has **4 general purpose registers** (no zero register or link register). **Each register can store 1 byte**, and there are **256 addresses in data memory**, each corresponding to one byte.

To use the machine:

Write your code as a series of the above instructions in the .txt file named ‘project2code.txt’. Note that commas and square brackets (] [,) are all optional, but **spaces between the mnemonics and operands are required** (there is an example already written there that uses all of the instructions in the manual). You can refer to registers as Xn or Rn (or any letter followed by the number 0,1,2, or 3. They all mean the same thing)

To run the assembler, go to the terminal and run the following command

$ python .\project2Assembler.py

This worked on my machine with python 3.1.1 installed (not sure about other versions)

After that,

1. open the circ file
2. navigate to the subcircuit called ‘instructionMemory’
3. reset the simulation (ctrl + r)
4. right click on the RAM in ‘instructionMemory’
5. click ‘load image’ and select ‘leastreqsImage.txt’ as the file
6. Then you can simulate the CPU by moving forward through clock cycles.

description	Operation	Opcode	Operands			
Rd = Rm	MOV Rd, Rm	1000 0001	R[d|t] (2-bit)	Rn (2-bit)	000 000	Rm (2-bit)
Rd=Rn+Rm	ADD Rd, Rn, Rm	1000 0101				
Rd=Rn-Rm	SUB Rd, Rn, Rm	1000 1101				
Rd=Rn*Rm	MUL Rd, Rn, Rm	1000 1001				
Rt=M[Rn+Rm]*	LDR Rt, [Rn, Rm]	1110 0101				
Rd=imm	MOVimm Rd, imm	1000 0010	R[d|t] (2-bit)	Rn (2-bit)	Simm (8-bit)	
Rd=Rn+imm	ADDimm Rd, Rn, imm	1000 0110				
Rd=Rn-imm	SUBimm Rd, Rn, Imm	1000 1110				
Rd=Rn*imm	MULimm Rd, Rn, imm	1000 1010				
Rt=M[Rn+imm]	LDRimm Rt, [Rn, imm]	1110 0110				
M[Rn+imm] = Rt	STRimm Rt, [Rn, imm]	0001 0110				

M[i] denotes the value of the byte in data memory at address i.

All the opcodes are 8 bits, each register needs 2 bits to encode because there are 4 registers total. The first set of instructions needs six zeroes before the last register because that's two less than the number of bits usually used for an immediate.

For MOV and MOVimm, the bits of Rn do not matter