MARS Simulator Tutorial

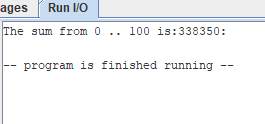
Anand Rajan

Pennsylvania State University, University Park

Section 002R

CMPEN 331

Q1. When the given program is written, assembled, and run, the following line is what is printed from the Run I/O Screen in the MARS Simulator.



Q2. When the program ends, the value in register $t7 is provided by the following row-wise information:



This tells us that the value stored in the register is 0x00002710 in hexadecimal. In decimal, this value comes out to be 10,000.

Q3. When the breakpoint is added at line 13 of the assembler code, and that single line is run, the register $t8 is highlighted, then the next instruction highlights $t9. This implies that that instruction was changing $t8, but being the first iteration, it didn’t change the value at all. On the second iteration, $at, $t0, $t6, and $t7 have all been incremented by one. $t7 is actually incremented on the very previous step. On the breakpoint, the lo register is highlighted, and increments after being run. The next step highlights $t8, which doesn’t get changed but $t9 does. Again, on the third iteration, $t0 is incremented by one, $t6 is incremented by one, and $t7 is incremented by 3 on the very previous step (likely because it stores the value of the square). On the actual step, lo is also saved to the same value as $t6, and $t8 is highlighted as it will be changed in the next step ($t9 gets changed next). This forms a visible cycle of constant changes in the same registers as that breakpoint is hit and then iterated through cycle after cycle. This will end when $t0 stores a value greater than 100.