Jair Ulises Nino-Espino
10/05/2018

Q1. MIPS Instruction Set

(a) [5 points] What MIPS instruction is this? 0000 0000 0101 0001 0100 1000 0010 0110

Split: 0000 0000 0101 0001 0100 1000 0010 0110 → 000000 00010 10001 01001 00000 100110

000000 = Special
00010 = rs = v0
10001 = rt = s1
01001 = rd = t1
00000 = shamt = 0
100110 = XOR

MIPS Instruction: **xor $t1 $v0 $s1**

(b) [5] What is the binary representation of this instruction? lw $t2, 4($s2)

LW rt, offset(base)

lw: 100011
$t2: 01010
$s2: 10010
4(offset) = 0000000000000100

Binary: **1000 1110 0100 1010 0000 0000 0000 0100**

Q2. C compilation and MIPS assembly language

(a) In the unoptimized version, the get_random procedure takes up 66 lines, whereas the optimized version has 33 lines for the same procedure.

(b) The unoptimized version uses 18 memory access, the optimized version only uses 6.

(c) The optimized version of get_random uses more registers than the unoptimized version.

(d) Base on the observation mention above, a strategy that the optimized version is using to enhance perforce is decreasing the number of memory accesses and instead using more registers. From the optimized code, the resistors are reused over and over again and replacing its old value with a new value. Registers are at the core of the CPU, accessing a register's value is fast, in comparison to accessing a value in memory. Therefore, by decreasing the number of memory accesses, the optimized version is able to improve the performance of the code.