

CMPE324: Quiz Sample questions

Q1) Assume that we have an application composed of 100 instructions, in which 30% of them are ALU instruction with CPI=1, 30% load/store instructions with CPI of 10, 10% floating point instructions with CPI of 20, and 30% branch instructions with CPI of 5. If the processor runs at 1GHz, answer the following:

Total number of clock cycles =

The average CPI for the program=

Q2) Consider the following hexadecimal value that represents one MIPS machine language instruction:

0X20A5FFFF

Convert this value to a 32-bit binary value and answer the following:

A/ What is the opcode value for this instruction?.....

B/ What is the name of the instruction?

C/ What is the instruction format (R, I, or J)

D/ what the equivalent MIPS instruction?

Q3) Consider the following machine code segment and the corresponding MIPS instructions. Complete the missing instructions. Note the following MIPS instructions formats:

Inst. Type	31-26	25-21	20-16	15-11	10-6	5-0
R	op= 0	Rs	Rt	rd	shamt	func
I	op= 1, 4-62	Rs	Rt	Immediate		
J	op= 2, 3	target address				

Address	Machine Inst.	MIPS Instruction
0x00800008	0x00001025	add \$v0,\$0,\$0
0x0080000c	0x0005402A	Loop: slt \$t0,\$0,\$a1
0x00800010	beq \$t0,\$0,exit
0x00800014	0x00441020	add \$v0,\$0,\$0
0x00800018	0x20A5FFFF	addi \$a1,\$a1,-1
0x0080001c	j Loop
0x00800020		Exit:

Q4) For each of the following high-level code segments, complete the given equivalent MIPS assembly implementation.

C++ Code segment	Equivalent MIPS
<pre>int pow = 1; int x = 0; while (pow != 256) { pow = pow * 4; x = x + 1; }</pre>	<pre># \$s0 = pow, \$s1 = x addi \$s0, \$0, 1 add \$s1, \$0, \$0 addi \$t0, \$0, 256 while: addi \$s1, \$s1, 1 done:</pre>
<pre>int x; if (x ≥ 0 && x < 8) x++;</pre>	<pre># \$s1 = x addi \$s0, \$zero, 8 addi \$s1, \$s1, 1 exit:</pre>

Q5) For the following C++ statements, write the corresponding MIPS implementation. Use register names as the variable names. If necessary, use \$t0 only as temporarily register.

<p>A/ Loop:</p> <p> # loop stuff</p> <p> if (s3 < s2) go to Loop;</p>	<p>Loop:</p> <p> # loop stuff</p> <p> </p>
<p>B/ s1 = s1 + s5[s3];</p>	<p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p>