Questions

• What is CPI?

• Suppose that 15% of the instructions of a program take 2 cycles, 25% take 3 cycles, and 60% take 1 cycle. What is the CPI of the program?
Answers

• What is CPI? A: Cycles per instruction
• Suppose that 15% of the instructions of a program take 2 cycles, 25% take 3 cycles, and 60% take 1 cycle. What is the CPI of the program?
  \[ -0.15 \times 2 + 0.25 \times 3 + 0.6 \times 1 = 1.65 \]
Question

• What is CPI?

• Suppose that 15% of the instructions of a program take 2 cycles, 25% take 3 cycles, and 60% take 1 cycle. What is the CPI of the program?

• Suppose the same program above executes 1,000,000 instructions. How many cycles would it take to execute the program?
Answer

- What is CPI?
- Suppose that 15% of the instructions of a program take 2 cycles, 25% take 3 cycles, and 60% take 1 cycle. What is the CPI of the program?
- Suppose the same program above executes 1,000,000 instructions. How many cycles would it take to execute the program?

- \[ \text{CPI} = 1.65 \]
- \[ 1,000,000 \times 1.65 = 1,650,000 \text{ cycles} \]
Question

• Suppose that the above program runs on a machine that has a cycle time of 200 ps. What is the execution time of the program on this machine?
Answer

• Suppose that the above program runs on a machine that has a cycle time of 200 ps. What is the execution time of the program on this machine?
  
  – 1,650,000 cycles * 200 * 10^{-12} = 3.3 * 10^{-4} sec = 0.33 msec
Question

• Suppose that compiler optimization is used to compile the same program as before. The optimization reduces the total number of instructions by 10% and now 12% of the instructions of the program take 2 cycles, 28% take 3 cycles, and 60% take 1 cycle. What is the execution time of the program now?
Answer

• Suppose that compiler optimization is used to compile the same program as before. The optimization reduces the total number of instructions by 10% and now 12% of the instructions of the program take 2 cycles, 28% take 3 cycles, and 60% take 1 cycle. What is the execution time of the program now?

\[-0.9 \times 1,000,000 \times (0.12 \times 2 + 0.28 \times 3 + 0.6 \times 1) \times 200 \times 10^{-12} = 3.36 \times 10^{-4} \text{ sec}\]
Question

• Consider that 20% percent of a program’s instructions are branch instructions and that the CPI for these instructions is 2. The CPI for the remaining instructions is 1.8. What would be the CPI of the program if the hardware designers improved the branch prediction algorithm so that the CPI of branch instructions went down to 1.2?
Answer

• Consider that 20% percent of a program’s instructions are branch instructions and that the CPI for these instructions is 2. The CPI for the remaining instructions is 1.8. What would be the CPI of the program if the hardware designers improved the branch prediction algorithm so that the CPI of branch instructions went down to 1.2?

\[-0.2 \times 1.2 + 0.8 \times 1.8 = 1.68\]
Question

• Which of these elements can influence the number of instructions executed by a program?
  – The algorithm
  – Its input data
  – The language in which it is written
  – The compiler
  – The ISA
Answer

• Which of these elements can influence the number of instructions executed by a program?
  – The algorithm
  – Its input data
  – The language in which it is written
  – The compiler
  – The ISA

  Answer: all
Question

• How would you compute the CPU time of a program as a function of the number of instructions, the CPI, and the clock cycle duration?
Answer

• How would you compute the CPU time of a program as a function of the number of instructions, the CPI, and the clock cycle duration?

CPU time = # instructions * CPI * clock cycle duration
Question

• What is the motivation to design multicore computers?
Answer

• What is the motivation to design multicore computers?

The power wall. Processors are using too much power and dissipating too much heat at current clock frequencies
Question

• Is there any instruction in the MIPS ISA that allows a number in main memory to be added to a number in a register?
Answer

• Is there any instruction in the MIPS ISA that allows a number in main memory to be added to a number in a register?

No.
Question

• What is the MIPS instruction needed to load element A[2] of array A into register $t0 assuming the address of the array is stored at register $s0 and that each element of the array is a 4-byte integer?
Answer

• What is the MIPS instruction needed to load element A[2] of array A into register $t0 assuming the address of the array is stored at register $s0 and that each element of the array is a 4-byte integer?

  – lw $t0, 8 ($s0)
Questions

• Why MIPS does not have a subtract immediate instruction?
• How are negative integer numbers represented in MIPS and in the majority of processors?
Answers

• Why MIPS does not have a subtract immediate instruction?
  – Because this can be accomplished by a addi in which one of the operands is negative

• How are negative integer numbers represented in MIPS and in the majority of processors?
  – 2’s complement
Questions

• What is the sign bit of negative integer numbers in 2’s complement?
• How do you negate a number?
Answers

• What is the sign bit of negative integer numbers in 2’s complement?
  – 1

• How do you negate a number?
  – Flip the bits and add 1
Question

• How would you compile the statement into MIPS using 3 instructions?

```c
if (i==j) f = g;
else f = h;
```

where $i, j$ are in $t0, t1$, $f, g$, and $h$ are stored in $s0, s1, s2$. 
• How would you compile the statement into MIPS using 3 instructions?

```mips
if (i==j) f = g;
else f = h;
```

where i, j, are in $t0, $t1, f, g, and h are stored in $s0, $s1, and $s2.

```mips
add $s0, $zero, $s1    # f = g
beq $t0,$t1,LABEL    # skip else if i = j
ELSE add $s0, $zero, $s2    # f = h
LABEL -------
```
Question

Consider the instructions

\[
\begin{align*}
\text{slt} & \; \text{\$t0, \$s1, \$s2} \\
\text{bne} & \; \text{\$t0, \$zero, L1}
\end{align*}
\]

And consider that \$s1 = 3 and \$s2 = 5.
What is the address branched to by the bne instruction?
Answer

Consider the instructions

```assembly
slt $t0, $s1, $s2
bne $t0, $zero, L1
```

And consider that $s1 = 3 and $s2 = 5. What is the address branched to by the bne instruction?

$t0$ is set to 1. Then, branch to L1.
Question

• What is the purpose of the instruction below and what it does?

jal Label
Answer

• What is the purpose of the instruction below and what it does?

jal Label

It saves the address of the instruction following the jal in the $ra register and changes the PC to the address of the instruction that corresponds to Label
Question

• What is the purpose of the instruction below and what it does?

jr $ra
Answer

• What is the purpose of the instruction below and what does it do?
  jr $ra

It jumps to the address stored in the register $ra
Question

- Consider the beq instruction stored at address \(1000_{10}\). The value of the address field is \(200_{10}\). What is the address of the next instruction if \(rs\) and \(rt\) are equal?

<table>
<thead>
<tr>
<th>op</th>
<th>rs</th>
<th>rt</th>
<th>constant or address</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 bits</td>
<td>5 bits</td>
<td>5 bits</td>
<td>16 bits</td>
</tr>
</tbody>
</table>
Answer

• Consider the beq instruction stored at address $1000_{10}$. The value of the address field is $200_{10}$. What is the address of the next instruction if $rs$ and $rt$ are equal?

<table>
<thead>
<tr>
<th>op</th>
<th>rs</th>
<th>rt</th>
<th>constant or address</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 bits</td>
<td>5 bits</td>
<td>5 bits</td>
<td>16 bits</td>
</tr>
</tbody>
</table>

If $rs$=rt, the target address of the branch is $(1000+4) + 200*4 = 1804_{10}$
Consider the jump instruction stored at address $A80094_{16}$. The value of the address field is $800_{10}$. What is address in binary of the next instruction to be executed?

<table>
<thead>
<tr>
<th>op</th>
<th>address</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 bits</td>
<td>26 bits</td>
</tr>
</tbody>
</table>
Answer

• Consider the jump instruction stored at address $A8009004_{16}$. The value of the address field is $800_{10}$. What is address in binary of the next instruction to be executed?

$$A8009004_{16} + 4_{10} = A_{16}A8009008_{16}$$

$=> A = 1010_2$; $800*4=3200_{10} = 000...110010000000_2$

Target address $= 1010$: $800*4 = \mathbf{1010}000...110010000000_2$
Question

• Consider adding the numbers -100 and -64 represented in 2’s complement using 8 bits. What is the result of the computation?
Answer

• Consider adding the numbers -100 and -64 represented in 2’s complement using 8 bits. What is the result of the computation?

• $100_{10} = 01100100_2$ => $-100_{10} = 10011100_2$
• $64_{10} = 01000000_2$ => $-64_{10} = 11000000_2$
• $-100-64 = 01011100_2$

• Adding two negative numbers results in a positive number => overflow
Question

Consider a 2 x 3 matrix stored in memory in \textit{column major order}, i.e., elements are stored column by column. Each element is 4-bytes long. What is the byte offset of element \(i,j\)?
Answer

Consider a $2 \times 3$ matrix stored in memory in \textit{column major order}, i.e., elements are stored column by column. Each element is 4-bytes long. What is the byte offset of element $i,j$?

Byte offset of $[i,j] = [j \times 2 + i] \times 4$ because before $[i,j]$ there are $j$ full columns and $i$ elements.
Question

Write a minimal set of MIPS assembly instructions that does the identical operation as the C code below. Assume the base address of C is in $s1 and that A is in $s2. Use the minimum number of registers. Do not destroy the contents of $s1 or $s2.

A = C[0] << 4;
Write a minimal set of MIPS assembly instructions that does the identical operation as the C code below. Assume the base address of C is in $s1 and that A is in $s2. Use the minimum number of registers. Do not destroy the contents of $s1 or $s2.

A = C[0] << 4;

```
lw     $t1, 0($s1)  # $t1 <- C[0]
sll    $t1, $t1,4   # $t1 <- $t1 << 4
sw     $t1, 0($s2)  # A <- $t1
```
Exercise 2.26.1

Consider the following MIPS code with the following initial values: $t1 = 10$ and $s2 = 0$.

```
LOOP:  slt   $t2, $0, $t1
       beq   $t2, $0, DONE
       subi  $t1, $t1, 1
       addi  $s2, $s2, 2
       j     LOOP
DONE:
```

What is the final value of $s2$?
Solution to Exercise 2.26.1

Consider the following MIPS code with the following initial values: $t1 = 10$ and $s2 = 0$.

```
LOOP: slt $t2, $0, $t1  # if $t1 > 0 then $t2 = 1 else $t2 = 0
    beq $t2, $0, DONE  # if $t2 = 0 then go to DONE
    subi $t1, $t1, 1    # $t1 = $t1 -1
    addi $s2, $s2, 2    # $s2 = $s2 + 2
    j LOOP              # Go to LOOP
DONE:
```

Number of loop executions:
$t1$ at top = 10; $t1$ at bottom = 9
...
$t1$ at top = 1; $t1$ at bottom = 0 ➔ 10 executions ➔ $s2 = 2 \times 10 = 20$
Question

Describe what the following MIPS code does.

```
addi $s2,$0,$0
addi $t1,$0,$0
LOOP
lw $s1,0($s0)
add $s2,$s2,$s1
addi $s0,$s0,4
addi $t1,$t1,1
slt $t2,$t1,100
bne $t2,$0,LOOP
DONE:
```
Answer

Describe what the following MIPS code does.

```mips
addi $s2, $0, $0  # $s2 = 0
addi $t1, $0, $0  # $t1 = 0
LOOP
lw  $s1, 0($s0)  # $s1 = Mem[$s0]
add  $s2, $s2, $s1  # $s2 = $s2 + Mem[$s0]
addi  $s0, $s0, 4  # $s0 = $s0 + 4
addi  $t1, $t1, 1  # $t1 = $t1 + 1
slti  $t2, $t1, 100  # $t1 = 1 if $t1 < 100; $t1 = 0 otherwise
bne  $t2, $0, LOOP  # branch to LOOP if $t2 ≠ 0 ($t1 < 100)
DONE:
```

Code meaning: store in $s2 the sum of all 100 words stored starting at address $s0
Consider a multiprocessor with p processors. Assume that 25% of the instructions of a program can be executed in parallel using all p processors. The remaining 75% of the instructions have to be executed sequentially. Assume that the time to execute the program sequentially (i.e., using only one processor) is $T_s$. Give an expression for $S(p)$, the speedup obtained when using p processors.

What is the maximum possible speedup? i.e. ($\lim_{p \to \infty} S(p)$)
Answer

Consider a multiprocessor with \( p \) processors. Assume that 25\% of the instructions of a program can be executed in parallel using all \( p \) processors. The remaining 75\% of the instructions have to be executed sequentially. Assume that the time to execute the program sequentially (i.e., using only one processor) is \( T_s \). Give an expression for \( S(p) \), the speedup obtained when using \( p \) processors.

\[
S(p) = \frac{T_s}{0.75 T_s + 0.25 \frac{T_s}{p}} = \frac{1}{0.75 + 0.25/p}
\]

\[
\lim_{p \to \infty} S(p) = \frac{1}{0.75} = 4/3 = 1.33
\]
Provide the values of RegWrite, AluSrc, MemRead, MemWrite, MemtoReg, Branch for add $t1, $t2, $t3
RegWrite=1, AluSrc=0, MemWrite=0, MemRead=0, MemtoReg=0, Branch=0 for add $t1, $t2, $t3
Provide the values of RegWrite, AluSrc, MemRead, MemWrite, MemtoReg, Branch for `lw $t1, 32($t3)`
RegWrite=1, AluSrc=1, MemRead=1, MemWrite=0, MemtoReg=1, Branch=0 for lw $t1, 32($t3)
Provide the values of RegWrite, AluSrc, MemRead, MemWrite, MemtoReg, Branch for sw $t1, 32($t3)
RegWrite=0, AluSrc=1, MemRead=0, MemWrite=1, MemtoReg=*, Branch=0 for sw $t1, 32($t3)
Provide the values of RegWrite, AluSrc, MemRead, MemWrite, MemtoReg, Branch for beq $t1,$t2,label
RegWrite=0, AluSrc=0, MemRead=0, MemWrite=0, MemtoReg, Branch=1 for beq $t1,$t2,label