

1) Suppose you have a machine which executes a program consisting of 50% floating point multiply, 20% floating point divide, and the remaining 30% are from other instructions. Management wants the machine to run 4 times faster. You can make the the divide run at most 3 times faster and the multiply run at most 8 times faster. Can you meet management's goal by making only one improvement, and which one?

First we calculate the value of 'infinite' speedup for floating point multiplications

$$\begin{aligned} speedup &= \frac{1}{(1 - .5)} \\ &= 2 \end{aligned}$$

Next we calculate the value of 'infinite' speedup for floating point divide

$$\begin{aligned} speedup &= \frac{1}{(1 - .2)} \\ &= 1.25 \end{aligned}$$

We conclude that neither would satisfy managements goals

2) Computer A with 1 MHz clock rate runs a program in 10s. Computer B requires 1.5 times as many clock cycles as computer A for the same program and runs it in 3s. Find the clock speed of computer B.

We first calculate the clocks cycles for the program running on computer A

$$\begin{aligned} CC &= 10 \times 10^6 \\ &= 10^7 \end{aligned}$$

We next calculate the clock rate for computer B given the execution time and clock cycles

$$\begin{aligned} CR &= \frac{(1.5) \times 10^7}{3} \\ &= 5Mhz \end{aligned}$$

4) Convert -35.75 to its hexidecimal representation in IEEE floating point format:

0 × C20F0000

5) Convert the hexadecimal IEEE floating point number 0x40200000

2.5

6) Add the 8-bit 2's complement number 1110 1001 + 1100 0010. State the result of the addition, as well as whether overflow occurred if the number were treated as signed, and whether overflow occurred if the numbers were treated as unsigned.

no overflow with signed representation, overflow with signed representation

7) Add the 8-bit 2's complement numbers 0110 1111 + 0110 1010. State the result of the addition, as well as whether overflow occurred if the numbers were treated as signed, and whether overflow occurred if the numbers were treated as unsigned.

signed: overflow

unsigned: no overflow

8) Convert the decimal -176.375 to IEEE 754 32 bit floating point representation

$0 \times c3306000$

9) Translate the following MIPS code to C. assume that the variables f, g, h, i, and j are assigned to registers \$s0, \$s1, \$s2, \$s3, and \$s4, respectively. Assume that the base array A and B are in registers \$s6 and \$s7, respectively.

```
addi $t0, $s6, 4#$t0 = &A + 1
```

```
add $t1, $s6, $zero#$t1 = $A
```

```
sw $t1, 0($t0)#A[0] = &A
```

```
lw $t0, 0($t0)#$t0 = &A
```

```
add $s0, $t1, $t0#$s0 = 2 * &A
```

```
int s0 = 2 * &A
```

10) We are considering an enhancement to the processor of a web server. The new CPU is 20 times faster on search queries than the old processor. The old processor is busy with search queries 70% of the time, what is the speedup gained by integrating the enhanced CPU?

$$\begin{aligned} \text{speedup} &= \frac{1}{((1 - .7) + \frac{.7}{20})} \\ &= 2.99 \end{aligned}$$

11) Computer A has an overall CPI of 1.3 and can be run at a clock rate of 600MHz. Computer B has a CPI of 2.5 and can be run at a clock rate of 750MHz. We have a particular program we wish to run. When compiled for computer A, this program has exactly 100,000 instructions. How many instructions would the program need to have when compiled for Computer B, in order for the two computers to have exactly the same CPU execution time for this program?

65000 instructions

12) Suppose we have a 32-bit MIPS word containing the value 0x008A1021

We would like to know what MIPS instruction this represents

- a) Write this instruction word in binary  
0000 0000 1000 1010 0001 0000 0010 0001
- b) What type of instruction is this?  
R type
- c) op code rs rt rd shamt function  
6 bit 5 bit 5 bit 5 bit 5 bit 6 bit
- d) Translate this instruction to assembly language  
addu \$v0, \$a0, \$t2

13) FPSQR is responsible for 20% of the exec time of a machine. FP instructions are responsible for 50% of the execution time. Which is faster?

A. Add FPSQR hardware that can speed up this operation by 10

$$\begin{aligned} \text{speedup} &= \frac{1}{((1 - .20) + \frac{.2}{10})} \\ &= 1.22 \end{aligned}$$

B. Make all FP instructions twice faster

$$\begin{aligned} \text{speedup} &= \frac{1}{((1 - .5) + \frac{.5}{2})} \\ &= 1.333 \end{aligned}$$

From this we conclude that option B is better

14) A common transformation required in graphics engines is square root implementations of floating-point (FP) square root vary significantly in performance, especially among processors designed for graphics. Suppose FP square root (FPSQR) is responsible for 20% of the execution time of a critical graphics benchmark. One proposal is to enhance the FPSQR hardware and speed up this operation by a factor of 10. The other alternative is just to try to make all FP instructions in graphics processor run faster by a factor of 1.6; FP instructions are responsible for a total of 50% of the execution time for this for the application. The design team believes that they can make all FP instructions run 1.6 times faster with the same effort as required for the fast square root. Compare these design alternatives

$$\begin{aligned} \text{speedup} &= \frac{1}{((1 - .2) + \frac{.2}{10})} \\ &= 1.22 \end{aligned}$$

$$\begin{aligned} \text{speedup} &= \frac{1}{((1 - .5) + \frac{.5}{1.6})} \\ &= 1.23 \end{aligned}$$

We conclude the second optimization is better

15) Suppose that we are considering an enhancement that runs 10 times faster than the original machine but is only usable 40% of the time. What is the overall speedup gained by incorporating the enhancement?

$$\begin{aligned} \text{speedup} &= \frac{1}{((1 - .4) + \frac{.4}{10})} \\ &= 1.5625 \end{aligned}$$

16) We want to speed up computer performance with an additional unit for calculating in floating point format. This unit is 20 times faster than the same operations without units. What percentage of a total computer time must this unit be active to achieve an overall increase in computer speed up of 2.5 times?

$$\begin{aligned}
 2.5 &= \frac{1}{((1-f) + \frac{f}{20})} \\
 2.5 \cdot ((1-f) + \frac{f}{20}) &= 1 \\
 2.5 - 2.5 \cdot f + \frac{2.5 \cdot f}{20} &= 1 \\
 -2.5f + \frac{2.5 \cdot f}{20} &= -1.5 \\
 -50 \cdot f + 2.5 \cdot f &= -30 \\
 -47.5 \cdot f &= -30 \\
 f &= \frac{30}{47.5} \\
 f &\approx 63\%
 \end{aligned}$$

17) Write the value of 4.75 in 32-bit floating point format IEEE 754. Another floating point value is written in the memory as 44FAC000. Which value does it represents if it written in the format IEEE-754?

0 × 40980000  
 2006<sub>10</sub>

18) Assume that x5 holds the value 512. For the instruction add x30, x5, x6 what is the range(s) of values for x6 that would result in overflow?

$$\begin{aligned}
 512 + x6 &> 2^{63} - 1 \\
 x6 &> 2^{63} - 513
 \end{aligned}$$

19) What decimal number does the bit pattern 0 × 0CD50000 represent if it is a floating point number? Use the IEEE 754 standard.

Convert to binary 0000 1100 1101 0101 0000 0000 0000 0000  
 E = 25  
 Exp = -102  
 $(1 + 2^{-1} + 2^{-3} + 2^{-5} + 2^{-7}) \times 2^{-102}$

20) For the following C statements, write the corresponding RISC-V assembly code. Assume that the variables i, and j are assigned to registers x28, and x29, respectively. Assume that the base address of the arrays A and B are in registers x10 and x11, respectively.

```

B[8] = A[i - j]
sub x30, x28, x29
slli x30, x30, 3
add x30, x30, x10
ld x30, 0(x30)
sd x30, 64(x11)

```

21) For the RISC-V assembly instructions below, what is the corresponding C statement? Assume that the variables f, g, h, i and j are assigned to registers x5, x6, x7, x28, and x29 respectively. Assume that the base address of the arrays A and B are in registers x10 and x11, respectively.

```

slli x30, x5, 3
add x30, x10, x30
slli x31, x6, 3
add x31, x11, x31
ld x5, 0(x30)

```

```

addi x12, x30, 8
ld x30, 0(x12)
add x30, x30, x5
sd x30, 0(x31)

```

$B[g] = A[f] + A[f + 1]$

22)  $B[8] = A[i] + A[j]$

```

slli x5, x28, 3
add x5, x10, x5
ld x5, 0(x5)
ld x5, 0(x5)
slli x6, x29, 3
add x6, x10, x6
ld x6, 0(x6)
add x5, x5, x6
sd x5, 64(x11)

```

23) not x5, x6

xor x6, x6, 0xFFFFFFFFFFFFFFFF

add x5, x0, x6

24)  $A = C[0] \ll 4$

ld x5, 0(x17)

slli x6, x5, 4

24) The final value would be 20

25) The RISC-V instruction is add x1, x1, x1

26) Assume that registers x5 and x6 holds the values 0x8000000000000000 and 0xD000000000000000, respectively

a. What is the value of x30 for the following assembly code?

0x5....0

b. Is the result in x30 the desired result, or has there been overflow? There has been overflow

c. What is the value of x30 for the following assembly code?

sub x30, x5, x6

The value is 0xD....0

d. There is no overflow

e.

f. The first addition resulted in overflow, the second did not.

27) Computer A: 2 GHz, 10s CPU time. In order to a computer B with 6s CPU time and 1.2 times the clock cycles of computer A, how fast must computer B clock be?

$$10 \times 2 \times 10^9 = 2 \times 10^{10}$$

$$CR = \frac{(1.2 \times 2 \times 10^{10})}{6}$$

$$CR = 4 \text{ GHz}$$

28) Assuming a standard unicycle (CPI=1) machine running at 10KHz and the number of instructions needed for the code is 102, how long will take for the unicycle machine to execute the code?

$$XT = \frac{102 \times 1}{10 \times 10^3} = 10.2 \text{ microsecs}$$

29) Add number  $x = 0100\ 0010\ 0000\ 1111\ 0000\ 0000\ 0000\ 0000$  and  $y = 0100\ 0001\ 1010\ 0100\ 0000\ 0000\ 0000\ 0000$

convert both numbers to floating points

$$x = 1.0001111 \times 2^5$$

$$y = 1.01001 \times 2^4$$

Normalize

$$y = 0.101001 \times 2^5$$

add

$$1.0001111 \times 2^5$$

$$0.1010010 \times 2^5$$

$$1.110001 \times 2^5$$

convert to IEEE 754 representation

$$5 + 127 = 132$$

$$0100001001100010 \dots 0$$

30) Let  $A = (1000\ 1000\ 1000\ 1000\ 1000\ 0000\ 0000\ 0000)_2$ . Represent  $A$  as a decimal number in the form  $1 \dots \times 2^e$

Determine Exp

$$\text{exp} = -110$$

$$-(1 + 2^{-4} + 2^{-8}) \cdot 2^{-110}$$

31) Convert the decimal number 147.625 into single precision IEEE 754 format

01000011000100111010...0

32) Convert the hexadecimal number  $0 \times 438F0000$

0100 0011 1000 1111 0000 0000 0000 0000

$286.0_{10}$

33) Convert  $28.125_{10}$  to single precision IEEE 754 floating point representation

0100000111100100...0

34) Add  $2.25 \times 10^0$  to  $1.340625 \times 10^2$

$$0.0225 \times 10^2$$

$$1.340625 \times 10^2$$

$$1.363125 \times 10^2$$

35) Show that the IEEE 754 binary representation of the number  $-0.75_{ten}$  in single precision.

101111101000000000000000000000

36) What decimal number does this single precision float represent?

110000001010000000000000000000

37) What decimal number does this single precision single point represent?

-5



38) A program consists of 5,000 floating point and 25,000 integer instructions. Processor A has a clock rate of 2.5 GHz. Floating point operations take 7 cycles and integer operations take 1 cycles. How long does it take to run this program?

$$\begin{aligned} XT &= \frac{IC \times CPI}{CR} \\ &= \frac{(30,000) \times 2}{2.5 \times 10^9} \\ &= 2.4 \times 10^{-5} \end{aligned}$$

$$CPI = \frac{(5,000 \times 7 + 25,000 \times 1)}{30,000} = 2$$

What is the average CPI for this program

The CPI is 2

Processor A runs Program 2 consists of 100,000 floating point instructions and 50,000 integer instructions. What is the average CPI in the program?

$$\begin{aligned} CPI &= \frac{(100,000 \times 7 + 50,000 \times 1)}{150,000} \\ &= 5 \end{aligned}$$

Professor B has an average CPI for program 2 of 3.5. Its clock rate is 1.8 GHz. How much time does it take to execute the program?

$$\begin{aligned} XT &= \frac{CPI \times IC}{CR} \\ &= \frac{3.5 \times 150,000}{1.8 \times 10^9} \\ &= 2.9167 \times 10^{-4} \end{aligned}$$

Which processor has the highest performance expressed in instructions per second?

	Clock Rate (GHz)	CPI
P1	3	1.5
P2	2.5	1.0
P3	4	2.2

$$3/1.5 = 2 \times 10^9 \text{ instructions per second}$$

$$2.5/1 = 2.5 \times 10^9 \text{ instructions per second}$$

$$4/2.2 = 1.8 \times 10^9 \text{ instructions per second}$$

P2 has the highest performance expressed in instructions per second

39) Implement the following C code for RISC-V assembly. Assume that the values of a, b, i, j are in registers X5, X6, x7 and X29 respectively. Also assume that register X10 holds the base address of the array A

```

for(i      =      1; i < a; i++)
    for(j = 0;  j < b; j++)
        A[i]    = A[i] + 2 * j + i;

```

```

addi x7, x7, 0
FOR_I :
    beq x5,    x7,    END_I
    addi x29,   x29,   0
FOR_J :
    beq  x29, x6,    END_J
    slli x13, x29,   1
    slli x14, x7,    2
    add  x15, x14,   x10
    ld   x16,      0(x15)
    add  x16, x13,   x16
    add  x16, x16,   x7
    addi x29,    , x29  1
    beq x0,     x0,    FOR_J
END_J :
    addi x7, x7,    1
    beq x0,     x0,    FOR_I
END_I

```