20181614 김주연

Computer Architecture and Logic HW#3

Due: 5/17(Sun) 23:59

Problem 1. [40points]

Company A has designed a processor P1 with the following CPIs. The clock rate of P1 is 800 MHz.

Instruction type	CPI
Load	5
Store	4
ALU operations	3
Branch	5
Jumps	3

(a) **[12 points]** You have written an application program that has 2 million instructions in a loop that is executed 1,000 times with the following instruction mix.

Instruction type	Frequency
Load	40%
Store	15%
ALU operations	30%
Branch	10%
Jumps	5%

What is the expected CPI for this workload? How long does the benchmark take to execute? **(Make sure you answer both questions.)**

∴ CPI : 4.15 ,
Execution Time : 10.375s

CPI = 5X0.4 + 4X0.15 + 3X0.3 + 5x0.1 + 3X0.05 = 4.15

Execution Time = IC X CPI X (1/800MHz) X 1,000 times
= 2,000,000 X 4.15 X 1250ps X1,000 = 10.375s

(b) [20 points] You propose to add the following new instruction.

Madd R1, R2, [R3,#offset]

This new instruction performs following operations.

1. read a memory location referenced by ([R3,#offset]).
2. add the content of the memory (just read) to the register (R2).
3. write the result to R1.

After the addition of a new instruction, the percentage profile for different operations has changed. This special addition instruction merges the load and add operation. 2/3 of ALU instruction can be merged with preceding load instruction. The CPI of the new instruction is 7 cycles and clock frequency has changed to 600 MHz. It is better to add this new instruction? Please justify your answer by calculating the new execution time and comparing that with the answer in part (a).

Execution Time New = 13.166... ~= 13.167s

·. It is better NOT to add this new instruction,
as the new execution time is increased from
the original execution time.

(c) **[8 points]** What's the maximum speed-up you can get by reducing the CPI of only ALU and branch operations. Do not consider the change made in (b). This problem is based on the assumption in (a).

CPI = 5X0.4 + 4X0.15 + 1X0.3 + 1x0.1 + 3X0.05 = 3.15

Execution Time = IC X CPI X (1/800MHz) X 1,000 times
= 2,000,000 X 3.15 X 1250ps X1,000 = 7.875s

∴ 6.5s can be reduced

Problem 2. [40points]

We design a new 10-bit floating point format with 1 sign bit, 4 exponent bits, and 5 fraction bits.

It is identical to the 32-bit and 64-bit formats in terms of the meaning of fields and special encodings.

The bias used for the exponent field 7. The bit fields in a number are (sign, exponent, fraction).

Assume that we use flooring any intermediate and final results for the rounding (내림).

(a) **[8 points]** Express the largest positive number using this format.

∴ 0111011111

(c) **[8 points]** Translate the largest number in decimal number. (The exponent form is allowed as an answer.)

$$\begin{array}{ccl} \underline{0} & \underline{1110} & \underline{11111} \\ \downarrow & \downarrow & \downarrow \\ + & 14(10) & 1.11111(2) \\ & & = 1.96875(10) \end{array}$$

∴ 252

(d) **[14 points]** Add following two numbers in 10-bit format and express the result using the same format.

A = 1101000100 1 1010 00100 = (−1) X 1.001(2) X 2^3

B = 0110001010 0 1100 01010 = 1.0101(2) X 2^5

101.01(2) X 2^3 + −1.001(2) X 2^3 = 100.001(2) X 2^3 = 100.00(2) X 2^3 = 1.0000(2) X 2^5

∴ 0110000001

(e) **[10 points]** Please write down the range for the positive denormalized number. (You can use the exponential form but the significand should be a decimal number. In addition, you should exclude 0.)

0000011111 = 0.11111(2) X 2^−6 = 0.96875 X 2^−6
0000000001 = 0.00001(2) X 2^−6 = 0.03125 X 2^−6

∴ 0.03125 X 2^−6 ~ 0.96875 X 2^−6