1st Homework for Computer Architecture

Submission deadline: March. 29, 9:30am

（total 160 points）

1.1 [20/10/10/15] <1.6> In this exercise, assume that we are considering enhancing a machine by adding a vector mode to it. When a computation is run in vector mode it is 20 times faster than the normal mode of execution. We call the percentage of time that could be spent using vector mode the *percentage of vectorization*. Vectors are discussed in Appendix B, but you don't need to know anything about how they work to answer this question!

a. [20] <1.6> Draw a graph that plots the speedup as a percentage of the computation performed in vector mode. Label the *y* axis "Net speedup" and label the *x* axis "Percent vectorization."

b. [10] <1.6> What percentage of vectorization is needed to achieve a speedup of 2?

c. [10] <1.6> What percentage of vectorization is needed to achieve one-half the maximum speedup attainable from using vector mode?

d. [15] <1.6> Suppose you have measured the percentage of vectorization for programs to be 70%. The hardware design group says they can double the speed of the vector rate with a significant additional engineering investment. You wonder whether the compiler crew could increase the use of vector mode as another approach to increasing performance. How much of an increase in the percentage of vectorization (relative to current usage) would you need to obtain the same performance gain? Which investment would you recommend?

a) (20 points)

$$\text{speedup} = \frac{1}{(1-f) + f / 20}$$

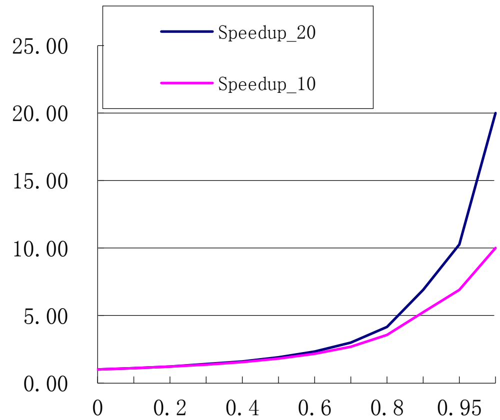
b) (10 points)

If we want to achieve a speedup of 2，

then from

$$2 = \frac{1}{(1-f) + f / 20}$$

we can calculate the f. f =52.6%.



c) (10 points)

If we want to achieve one-half the maximum speedup(应为 20), which equals 10, then

$$10 = \frac{1}{(1-f) + f / 20}$$

the percentage of vectorization must be 94.7%.

d) （15 points）

What should we do is to compare the two solutions of performance enhancement.

1) Hardware solution: to increase the S from 20 to 40.

2) Software method: to increase the F.

If the percentage of vectorization is now known as 70%, then from the hardware solution we can achieve the

$$\text{speedup} = \frac{1}{(1-70\%) + 70\% / 40}$$

If we want to achieve the same speedup as that from hardware solution, the compiler should increase the use of vector mode. So we get the equation as following.

$$\boxed{\frac{1}{(1-f) + f/20} = \frac{1}{(1-70\%) + 70\% / 40}}$$

From the equation above , the f can be calculated. F =71.8%.

Because it is more difficult to double the speed of the vector hardware than to increase the percentage of vectorization by 1.8%, it is better to adopt the software method.

1.2 [15/10] <1.6> Assume—as in the Amdahl's Law Example on page 30—that we make an enhancement to a computer that improves some mode of execution by a factor of 10. Enhanced mode is used 50% of the time, measured as a percentage of the execution time *when the enhanced mode is in use*. Recall that Amdahl's Law depends on the fraction of the original, *unenhanced* execution time that could make use of enhanced mode. Thus, we cannot directly use this 50% measurement to compute speedup with Amdahl's Law.

a. [15] <1.6> What is the speedup we have obtained from fast mode?

b. [10] <1.6> What percentage of the original execution time has been converted to fast mode?

$$\text{a. Speedup} = \frac{T_{old}}{T_{new}} = \frac{50\% + 50\% * 10}{50\% + 50\%} = 5.5$$

$$\text{b.} \quad F = \frac{50\% * 10}{50\% + 50\% * 10} = 90.9\%$$

1.3 [10/10/15] Suppose that a processor with a load/store architecture and no delayed branches executes at a clock rate of 2GHz. Arithmetic and logic instructions require 1 cycle, load and store operations 2 cycles, and conditional branches 3 cycles because of the control hazard involved.

The typical applications run on this processor contain a mix of 60% arithmetic and logic instructions, 20% load and store instructions and 20% conditional branches instructions.

An engineer proposes a modification in the architecture which introduces speculation. The branch prediction algorithm would be correct 50% of the time. When correct, branches would take 1 cycle, when incorrect, 3 cycles as before. However, the modification requires the reduction of the clock frequency to 1.8GHz.

a)[10] What is the average cycles per instruction of the original processor?
b)[10] What is the average cycles per instruction of the speculative processor?
c) [15] Is the speculative processor faster or slower than the original one? By how much?

Given :

ALU operations	60%	1		
Load/store	20%	2	CR = 2GHz	CC = 0.5 ns
Branches	20%	3		

a. CPI = 60%*1 + 20%*2 + 20% *3 = 1.6

b. CPI-speculation = 60%*1 + 20%*2 + 20%*(50% *1 + 50%*3) = 1.4

$$\text{c.Speedup} = \frac{\text{CPUtime}}{\text{CPUtime-specultation}} = \frac{\text{IC * CPI * CC}}{\text{IC * CPI-spec*CC-spec}} = \frac{1.6 * 1.8}{1.4 * 2}$$

= 1.029

So speculation is faster by 2.9%.   Timeup = 2.78%

1.4 P = ∑ wattage of each component / power supply efficiency

= (79W + 3.7W * 2 + 7.9W * 2) / 70%

= 146W

7.9W * (1 – 40%) + 4.0W * 40% = 6.34W

1.13 [15/15/15] <1.6,1.9> Three enhancements with the following speedups are proposed for a new architecture:

$\text{Speedup}_1 = 30$

$\text{Speedup}_2 = 20$

$\text{Speedup}_3 = 10$

Only one enhancement is usable at a time.

a. [15] <1.6> If enhancements 1 and 2 are each usable for 30% of the time, what fraction of the time must enhancement 3 be used to achieve an overall speedup of 10?

b. [15] <1.6,1.9> Assume the distribution of enhancement usage is 30%, 30%, and 20% for enhancements 1, 2, and 3, respectively. Assuming all three enhancements are in use, for what fraction of the reduced execution time is no enhancement in use?

c. [15] <1.6> Assume for some benchmark, the fraction of use is 15% for each of enhancements 1 and 2 and 70% for enhancement 3. We want to maximize performance. If only one enhancement can be implemented, which should it be? If two enhancements can be implemented, which should be chosen?

Amdahl's Law can be generalized to handle multiple enhancements. If only one enhancement can be used at a time during program execution, then

$$\text{speedup} = [\, 1 - \Sigma FEi + \Sigma\, FEi/Si \,]^{-1}$$

a.

$$10 = \frac{1}{1-(0.3+0.3 + FE3) + 0.3/30 + 0.3/20 + FE3/10}$$

so FE_3=36.1%

b.

$$T_{unenhanced}\% = \frac{T_{unenhanced}}{T\_total\ improved} = \frac{1 - \Sigma FEi}{1 - \Sigma FEi + \Sigma\, FEi/Si}$$

$$= \frac{1-(30\% + 30\% + 20\%)}{1-(30\% + 30\% + 20\%) + (30\%/30 + 30\%/20 + 20\%/10)} = 81.6\%$$

C.

$$\text{Speedup1} = \frac{1}{(1-15\%) + 15\%/30} = 1.17$$

$$\text{Speedup2} = \frac{1}{(1-15\%) + 15\%/20} = 1.17$$

$$\text{Speedup3} = \frac{1}{(1-70\% + 70\%/10)} = 2.7$$

$$\text{Speedup12} = \frac{1}{(1-15\%-15\%) + 15\%/30 + 15\%/20} = 1.4$$

$$\text{Speedup13} = \frac{1}{(1-15\%-70\%) + 15\%/30 + 70\%/10} = 4.4$$

$$\text{Speedup23} = \frac{1}{(1-15\%-70\%) + 15\%/20 + 70\%/10} = 4.4$$