CHAPTER 1

Problem 1.1

a. Using Amdahl's speedup,

$$\frac{1}{1 - F_{fp} + \frac{F_{fp}}{10}} > \frac{1}{1 - F_{ls} + \frac{F_{ls}}{2}}$$

Therefore,

$$\frac{F_{fp}}{F_{ls}} > \frac{5}{9}$$

b.Can you still find out which improvement is better based on these numbers?

Yes. The reason is F_{fp}/F_{ls} is equal to the ratio of the execution time of floating point instructions ($ExTime_{fp}$) and the execution time of loads and stores ($ExTime_{ls}$). We can get the ratio of these execution time with given information. If the ratio is larger than 5/9, which is the value obtained in part a, then we can say the floating point upgrade is better than the loads/stores upgrade.

$$ExTime_{fp} = IC_{fp} \times CPI_{fp} \times T_c$$

$$ExTime_{ls} = IC_{ls} \times CPI_{ls} \times T_c$$

$$ExTime_{total} = IC_{total} \times CPI_{total} \times T_c$$

Therefore,

$$\frac{F_{fp}}{F_{ls}} = \frac{IC_{fp} \times CPI_{fp} \times T_c}{IC_{ls} \times CPI_{ls} \times T_c} = \frac{ExTime_{fp}}{ExTime_{ls}}$$

Can you still estimate the maximum speedup of each upgrade using Amdahl's law?

No, because the total execution time or average CPI of *all* instructions is not given and we cannot get the fraction of execution time spent in floating point and loads/stores instructions respectively.

c. Floating-point improvement:
$1.5 = 1 / (1-F_{fp} +F_{fp}/10)$
Therefore, $F_{fp} = 0.3707$

Loads and Stores improvement:
$1.5 = 1 / (1-F_{ls} + F_{ls}/2)$
Therefore, $F_{ls} = 0.67$

d. After upgrading to the floating point units, the execution time is

$$ExTime_{fp} = \left(0.7 + \frac{0.3}{10}\right) \times ExTime_{base} = 0.73 \times ExTime_{base}$$

and the new fraction of time spent in loads and stores after the floating point upgrade is 0.20/0.73 = 0.274(27.4%).

Therefore, the maximum speedup of the cache upgrade after the floating point unit upgrade is

$$Speedup = \frac{1}{1 - 0.274 + \frac{0.274}{2}} = 1.1587$$

Problem 1.2

We solve this problem assuming that N≥1024 and the number of available processors *P* in the machine goes from 1 to 1024.

a. $$Speedup = \frac{T_1}{T_p} = \frac{NT_c}{\frac{N}{P}T_c + NT_b} = \frac{PR}{R+P}$$

b. For a given *R*, the speedup increases monotonically as *P* increases. The maximum speedup is thus achieved when *P* is 1024 and the maximum speedup is

$$Speedup_{max} = \frac{1024R}{1024 + R} \quad :$$

c. $P > \frac{R}{R-1}$

d. The execution time stays constant at NxT_c for all *P*'s >1 and given *N*. As *P* increases from 1 to 1024, the workload size (N_1) increases so that:

$$NT_c = \frac{N_1}{P}T_c + N_1 T_b$$

and:

$$N_1 = \frac{NPR}{R+P}$$

As *P* grows N_1 tends to an asymptote equal to NxR.

e. Reconsidering a-c above in the context of growing workload size.
In this part, for given *N*, the workload (N_1) grows according to d) above, so that the execution time remains constant. In this context T_1 is equal to $N_1 T_c$ and T_P remains fixed at NT_c

$$Speedup = \frac{N_1 T_c}{NT_c} = \frac{PR}{R+P}$$

Surprisingly the speedup with a growing workload is the same as the speedup in part a with constant size workload, and therefore the maximum speedup and minimum *P* are also the same as in part a. The reason is that the serial part (the bus accesses) grows with *P*, contrary to Amdahl's or Gustafson's laws where the serial part remains a constant.

f. Let $O = T_o/T_b$.

$$Speedup = \frac{T_1}{T_p} = \frac{NT_c}{\frac{N}{P}T_c + PT_o + NT_b} = \frac{PR}{R + P + \frac{OP^2}{N}}$$

Problem 1.3

a.

Table 1: Speedups of the three programs and average execution times (sec)

Machines	Program 1	Program 2	Program 3	Average Normalized Execution Time
Base machine	1	1	1	3.67
Base + FP units	1	5	1.67	2.334
Base + cache	1.43	1.11	1.25	2.903

Table 2: Average speedups

Machines	S1	S2	S3	S4
Base + FP units	1.57	2.55	1.67	2.03
Base + cache	1.26	1.26	1.25	1.26

The ratio of average execution time (S1): Base + FP units is better than Base + cache by 25%.

Arithmetic Mean (S2): Base + FP unit is better than Base + cache by a factor of 2.

Harmonic Mean (S3): Base + FP unit is better than Base + cache by 33%.

Geometric Mean (S4): Base + FP unit is better than Base + cache by 61%.

b. Even if we use normalized execution times of programs, the speedup for each program is equal to the speedup based on execution time in part a). Only the average normalized execution time is slightly different. Hence, average speedups are unchanged, except for S1 (see Table 3 and Table 4). In all cases, Base + FP units is still better than Base + cache.

Table 3: Speedups of the three programs and average normalized execution times

Machines	Program 1	Program 2	Program 3	Average Normalized Execution Time
Base machine	1	1	1	1
Base + FP units	1	5	1.67	0.6
Base + cache	1.43	1.11	1.25	0.8

Table 4: Average speedups

Machines	S1	S2	S3	S4
Base + FP units	1.67	2.55	1.67	2.03
Base + cache	1.25	1.26	1.25	1.26

c. Yes, this is a good idea. Comparing the execution times of the base machine and of the new machine with BLT-type instructions, the new machine is better than the base machine. Even though the CPI and the cycle time of the new machine are raised, the number of instructions (IC) is reduced. Therefore, the execution time of the new machine is shorter than the base machine.

$$ExTime_{base} = IC_{base} \times CPI_{base} \times Tc_{base}$$

$$ExTime_{new} = IC_{new} \times CPI_{new} \times Tc_{new} = 0.9 \times IC_{base} \times \frac{1.544}{1.49} \times CPI_{base} \times 1.05 \times Tc_{base}$$

$$= 0.98 \times ExTime_{base}$$

Problem 1.6

Since the data is related to the new machine, we have to find the data for the base machine without improvements to obtain the speedup.

The fraction of time that the new machine with 16 cores runs a single core is 25%. During that time 30% is used for floating point operations, which are 4 times faster than on the base machine. The fraction of time on the new machine is 0.25x.3=.075.
The fraction of time with a single core and no floating point operation is 0.25x.7=.175.

The fraction of time the new machine runs 16 cores is 75%. During that time each core runs floating point operations 30% of the time. Thus the fraction of time on the new machine is 0.75x.3=.225
The fraction of time with 16 cores and no floating point operation is 0.75x.7=.525.

First consider the upgrade to a 16-way CMP, with no fp unit. Let T_{16_nofp} be the execution time on this new machine and T_{base} the execution time on the base machine. In the phases when the 16-core machine executes 16 threads in parallel, the base machine must executed them one at a time.
T_{base_nofp}= (.25+.75x16)xT_{16_nofp}=12.25xT_{16_nofp}
Now consider adding the floating point units.
T_{base_fp}=(.3x4+.7)xT_{base_nofp}= 1.2x12.25xT_{16_nofp}=14.7xT_{16_nofp}

Therefore the speedup is 14.7.

Correction: