Problem 1.4

a. Not enough data
The reason is that the total execution time (in cycles) is not given.

b. The clock cycle of M2 is 1.25 times the clock cycle of M1. But the improvement for all Multiplies is a factor 20.
$Time_{M2} = 1.25*(0.8*Time_{M1} + 1/20*0.2*Time_{M1}) = 1.0125\ Time_{M1}$
Therefore,

$$\frac{Time_{M2}}{Time_{M1}} = 1.0125$$

M2 is 1.25% slower than M1. Thus the proposal is not an improvement.

c.This is caused by a simulation bug. According to Amdahl's law, the maximum speedup cannot be than 1.

$$Speedup = \frac{Time_{M1}}{Time_{M3}} = \frac{1}{\left(0.8 + \frac{0.2}{s}\right) \times 1.25} \le 1$$

Problem 1.5

a. Average $CPI_{base} = 0.4*1 + 0.25*2+0.1*1 + 0.08*1 + 0.12*3+0.05*1 = 1.49$

b. Branch instructions including taken and untaken are 20% of the Instruction Count of the base machine. Therefore, 10% of Instruction Count is removed from the Instruction Count of the base machine and

$$IC_{new} = 0.9*IC_{base}$$

We also need to get the new fraction of execution of each instruction class in order to compute CPI_{new}. For every instruction inst the new fraction of instructions of type *inst* left in the code is:

$$f_{inst\_new} = IC_{inst} / IC_{new}$$

Because SLT is an Arithmetic/Logic instruction, only IC_{ALU} (the instruction count of Arithmetic/Logic) is changed, and other IC_{inst} are unchanged. Each IC_{inst} can be obtained from Table 1.6.
For Arithmetic/Logic instructions,

$$IC_{ALU\_new} = (0.4-0.1)*IC_{base}$$

For other instructions,

$$IC_{inst} = f_{inst\_base} * IC_{base}$$

For ALU instructions, the new frequency is:
$f_{ALU\_new} = IC_{ALU\_new} / IC_{new} = (0.4-0.1)*IC_{base} / (0.9*IC_{base}) = 0.33$
For instructions other than ALU instructions, the frequency is:
$f_{inst\_new} = IC_{inst}/IC_{new} = f_{inst\_base} * IC_{base} / (0.9*IC_{base}) = f_{inst\_base} /0.9$

Hence, the new CPI is

$$CPI_{new} = f_{ALU\_new}*1+f_{Load\_new}*2+f_{Stores\_new}*1+f_{Br\_nt\_new}*1+f_{Br\_t\_new}*3+f_{misc\_new}*1$$
$$= 0.33*1 + f_{Load\_base}*2+f_{Stores\_base}*1+f_{Br\_nt\_base}*1+f_{Br\_t\_base}*3+f_{misc\_base}*1$$

$$CPI_{New} = \frac{1}{0.9}((0.4 - 0.1) \times 1 + 0.25 \times 2 + 0.1 \times 1 + 0.08 \times 1 + 0.12 \times 3 + 0.05 \times 1) = 1.544$$

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.