1.5

a. Instruction per second = instruction count / CPU time= instruction count/ (instruction count * CPI /clock rate)= clock rate/ CPI

Thus for P1, 3*10^9/1.5= 2*10^9 instructions/second

For P2, 2.5*10^9/1.0= 2.5*10^9 instructions/second

P3, 4*10^9/2.2= 1.82*10^9 instructions/second

b.

clock cycles= CPU time * clock rate

Thus for P1, 10*3*10^9= 3*10^10 cycles

For P2, 10*2.5*10^9= 2.5*10^10 cycles

For P3, 10*4*10^9= 4*10^10 cycles

Instruction count= clock cycles / CPI

Thus for P1, 10*3*10^9/1.5= 2*10^10 instructions

For P2, 10*2.5*10^9/1.0= 2.5*10^10 instructions

For P3, 10*4*10^9/2.2= 1.82*10^10 instructions

c.

CPU time= CPI *instruction count / clock rate

0.7*CPU time= 1.2*CPI *instruction count / expected clock rate

Thus expected clock rate = (1.2/0.7)*clock rate

Thus for P1, (1.2/0.7)*3= 5.14 GHz

For P2, (1.2/0.7)*2.5= 4.29 GHz

For P3, (1.2/0.7)*4= 6.86 GHz

1.6

a.

given the program, for P1, global CPI= 1*0.1+2*0.2+3*0.5+3*0.2=2.6 cycle/instruction

for P2, global CPI= 2*0.1+2*0.2+2*0.5+2*0.2=2.0 cycle/instruction

b.

clock cycles= CPI* instruction count

For P1, 2.6 * 10^6= 2.6 * 10^6 clock cycles

For P2, 2.0*10^6= 2.0*10^6 clock cycles

To say which one is faster, we know that: CPU time = clock cycles/ clock rate

Thus for P1, CPU time= 2.6 * 10^6 / (2.5*10^9)= 1.04*10^(-3) seconds

for P2, CPU time= 2.0 * 10^6 / (3*10^9)= 6.67*10^(-4) seconds

So in terms of CPU time, P2 is faster.

1.7

a.

average CPI= execution time /(clock cycle time * instruction count)

For A, average CPI= 1.1 / (10^(-9)*10^9)=1.1 cycle/second
For B, average CPI= 1.5 / (10^(-9)*1.2*10^9)=1.25 cycle/second

b.
Here we assume CPI is not affected by the new processor, otherwise we won't be able to solve this problem.
Since execution time= CPI * instruction count* clock cycle time
1.1*10^9*clock time A= 1.25*1.2*10^9*clock time B
So clock time A/ clock time B= 1.25*1.2/1.1=1.36
Since clock rate= 1/ clock time, clock rate A/ clock rate B =1.1/1.5= 0.73
So the clock rate of compiler A is 0.73 times that of compiler B.

c.
Here we measure speed up with CPU time. CPU time= instruction count * CPI* clock cycle time. And we assume that clock cycle time is the same for all compilers.
CPU performance new / CPU performance A = CPU time A/ CPU time new= (1.1*10^9)/(1.1*6*10^8)= 1.67
CPU performance new / CPU performance B = CPU time B/ CPU time new= (1.25*1.2*10^9)/(1.1*6*10^8)= 2.27
Thus, speed up of new compiler versus A is 1.67, speed up of new compiler versus B is 2.27.

1.13
1.13.1
0.8*70+85+40+55=236 seconds 1-236/250= 5.6%
So total time is reduced by 5.6%.

1.13.2
Here we assume INT operations time is the unmentioned time, which is 55 seconds, otherwise we won't be able to solve this problem. And we assume all times except INT operation time remain the same.
0.2*250= 50 seconds 50/55=90.9%
So INT operation time is reduced by 90.9% if total time is reduced by 20%.

1.13.3
250*0.2=50 seconds > 40 seconds
So the expected reduce time is greater than the total time of current branch instructions. Thus this can't be done.