EE116C/CS151B Homework 1 Solution

Problem 1

a. Which processor has the highest performance expressed in the instructions per second.

Instruction/second = 1 / cycles/instruction * cycles/second
so for:
P1: instructions/second = 1/1.5 * 3*10^9 = 2*10^9
P2: instructions/second= 2.5*10^9
P3: instructions/second= 1.818*10^9
thus P2 has the best performance.

b. If the processors each execute a program in 10 seconds, int he number of cycles and the number of instructions.
cycles counts = CPU time * cycles/second
Instruction counts = CPU time * cycles/second * instruction/cycles
so for:
P1: cycles counts = 10*3*10^9 = 3*10^10; Instruction counts = 10*3*10^9/1.5 = 2*10^10;
P2: cycles counts = 2.5*10^10; Instruction counts = 2.5*10^10;
P3: cycles counts = 4*10^10; Instruction counts = 1.818*10^10;

c. We are trying to reduce the time by 30% but this leads to an increase of 20% in the CPI. what clock rate should we have to get this time reduction?
CPU time = instructions / (instruction/cycles * cycles/second)
CPU time (new)/CPU time (old) = instruction/cycles (old) / instruction/cycle (new) * cycles/second (old)/ cycles/second (new) = 0.7;
thus, clock rate (new) / clock rate (old) = 1.2/0.7 = 1.7143;
thus for:
P1: 3GHz*1.7143=5.1429GHz;
P2: 4.2858GHz;
P3: 6.8572GHz;

Problem 2

a. what is the global CPI for each implementation.
P1: global CPI = 1*10% + 2*20% + 3*50% + 3*20% = 2.6
P2: global CPI = 2*10% + 2*20% + 2*50% + 2*20% = 2

b. Find the clock cycles required in both cases.
clock cycles = global CPI * 10^6 instructions
so for:
P1 = 2.6*10^6
P2 = 2*10^6

c. Which implementation is faster?
CPU time = clock cycles * time / cycle = clock cycles / clock rate
P2 is faster.

problem 3

a. find the average CPI for each program given that the processor has a clock cycle time of 1ns.
execution time = instruction count * cycles/instruction * cycles/second
so CPI = execution time / instruction count / cycles/second
so for:
A: CPI = 1.1 / 1.0E9 / (1/1E-9) = 1.1
B: CPI = 1.5 / 1.2E9 / (1/1E-9) = 1.25

b. assume the compiled programs run on two different processors. If the execution times on the two processors are the same, how much faster is the clock of the processor running compiler A's code versus the clock of the processor running compiler B's code?
execution time = instruction count * cycles/instruction * cycles/second
thus, execution time (A) / execution time (B) = 1
thus,
instruction count * cycles/instruction * cycles/second (A) = instruction count * cycles/instruction * cycles/second (B)
thus, rate(A)/rate(b) = 1.2E9*CPI(B)/(1E9*CPI(A)) = 1.3636

c. A new compiler is developed that uses only 6.0E8 instructions and has an average CPI of 1.1. What is the speedup of using this new compiler versus using compiler A or B on the original processor?
execution time = instruction count * cycles/instruction * cycles/second
thus, execution time (new) = 6E8*1.1*rate
thus,
t(A)/t(new) = 1.667
t(B)/t(new) = 2.7273

problem 4

As for the Jump (j) instruction:
The previous PC is Ox2000 0000, thus it is
$(0010\ 0000\ 0000\ 0000,\ 0000\ 0000\ 0000\ 0000)_2$ in binary address;
The format of Jump is opcode (6 bit) + address (26 bit)
The first 4 digital is [0010], the farthest place you can jump is to set address as Ox 03FF FFFF, with a multiple of 4, the farthest place is [0010] [1111 1111 1111 1111 1111 1111 1100] = Ox 2FFF FFFC
So, it is not possible.
As for the branch instruction:

The beq format is:
Opcode (6 bit) + operand1 (5 bits) + operand2 (5 bits) + offset (16 bits)
The farthest place that can reach is to set offset as: $[0111\ 1111\ 1111\ 1111]_2$;
Thus the farthest offset is SE(offset)*4 = Ox 0001 FFFC
Thus, it is not possible.

problem 5

a. Assume that the register \$t1 is initialized to the value 10. What is the value in register \$s2 assuming \$s2 is initially zero?

The program is equal to:

```
do{
if 0<t1
      t2=1;
if t2=0
      break;
t1=t1-1;
s2=s2+2;
} while (1);
```

Thus, the loop is executed for 10 times before t1 becomes 0.
Thus, s2 = 20.

b. For the loops written in MIPS assembly above, assume that the register \$t1 is initialized to the value N. How many MIPS instructions are executed?

There are 5 instruction in one loop. As the loop has been executed for N times, and the first two line of the program is executed for 2 more times before the program jump to done, the total instructions executed are: 5N+2.