Homework 1

#1.2

a. Performance via Pipeline
b. Dependability via Redundancy
c. Performance via Prediction
d. Make the Common Case Fast
e. Hierarchy of Memories
f. Performance via Parallelism
g. Use Abstraction to Simplify Design

#1.7

a)

$$ET = (InstructionCount * CPI)/ClockRate$$
$$ET_{p1} = (10^5 + 4 * 10^5 + 15 * 10^5 + 6 * 10^5)/2.5 * 10^9 = 1.04 * 10^{-3}$$
$$ET_{p2} = (2 * 10^5 + 4 * 10^5 + 10 * 10^5 + 4 * 10^5)/3 * 10^9 = 6.67 * 10^{-4}$$

$$CPI_{p1} = .1(1) + .2(2) + .5(3) + .2(3) = 2.6$$
$$CPI_{p2} = .1(2) + .2(2) + .5(2) + .2(2) = 2.0$$

b)

$$ClockCycles = InstructionCount * CPI$$
$$ClockCycles_{p1} = 2.6 * 10^6$$
$$ClockCycles_{p2} = 2.0 * 10^6$$

#1.8

a)

$$ET_A = 1.1s, ET_B = 1.5s, ClockRate = 1 * 10^9,$$
$$CPI = ET * ClockRate / InstructionCount$$
$$CPI_A = 1.1 * 1.0 * 10^9 / 1.0 * 10^9 = 1.1$$
$$CPI_B = 1.5 * 1.0 * 10^9 / 1.2 * 10^9 = 1.25$$

Average CPI of compiler A = 1.1

Average CPI of compiler B = 1.25

b)

$$ClockRate_A / ClockRate_B = (1.2 * 10^9 * 1.25)/(1.0 * 10^9 * 1.1) = 1.36$$

The clock of the processor running compiler A’s code is 1.36x faster than the clock of the processor running compiler B’s code.

c)

$$ET_n = (6 * 10^8 * 1.1)/(1.0 * 10^9) = 0.66$$
$$ET_A / ET_n = 1.1/0.66 = 1.67$$
$$ET_B / ET_N = 1.5/0.66 = 2.27$$

Speedup of new compiler vs compiler A = 1.67

Speedup of new compiler vs compiler B = 2.27