EN605.204.82.SU17 Computer Organization

Caleb Bowers

Assignment 1

1.2

a. Assembly lines in automobile manufacturing: “Performance via Pipelining”

b. Suspension bridge cables: “Performance via Parallelism”

c. Aircraft and marine navigation systems that incorporate wind information: Performance via Prediction

d. Express elevators in buildings: “Make the Common Case Fast”

e. Library reserve desk: “Hierarchy of Memories”

f. Increasing the gate area on a CMOS transistor to decrease its switching time: “Design for Moore’s Law”

g. Adding electromagnetic aircraft catapults (which are electrically-powered as opposed to current steam-powered models), allowed by the increased power generation offered by the new reactor technology: “Dependability via Redundancy”

h. Building self-driving cars whose control systems partially rely on existing sensor systems already installed into base vehicle, such as lane departure systems and smart cruise control systems: “Use Abstraction to Simplify Design”

1.4

a. What is the minimum size in bytes of the frame buffer to store a frame?

Answer: There are 8 bits per primary color per pixel. There are three primary colors: red, green, and blue. Therefore, there are:

$$8 \text{ pixels} * 3 \text{ primary colors} = 24 \text{ bits per pixel}$$

which means there are 3 bytes per pixel. The frame size is 1280 pixels by 1024 pixels, which means there are a total of

$$1280 * 1024 = 1{,}310{,}720 \text{ pixels}$$

Given this number of bytes per pixel and this number of pixels the minimum size of the frame buffer, in bytes, to store a frame must be

$$1{,}310{,}720 * 3 = \mathbf{3{,}932{,}160 \text{ bytes}}$$

or **3.93216 megabytes**.

b. How long would it take, at a minimum, for the frame to be sent over a 100 Mbit/s network?

Answer: A 100 Mbit/s network translates to

$$\frac{100 \text{ Mbit/s}}{8 \text{ Mbit/Mbyte}} = 12.5 \text{ Mbyte/s}$$

So, it takes $\frac{1}{12.5}$ seconds to send 1 megabyte, which means that in order to send all 3.93126 megabytes it will take:

$$3.93126 \text{ mbytes} * \frac{1 \text{ second}}{12.5 \text{ mbytes}} = \mathbf{0.3145 \text{ seconds}}$$

1.7

a. Find the average CPI for each program given that the processor has a clock cycle of 1 ns.

Answer:

$$\text{CPI} = \frac{\text{CPU\_ClockCycles}}{\text{InstructionCount}} \qquad (1)$$

$$\text{CPU\_ClockCycles} = \frac{\text{ExecutionTime}}{\text{ClockCycleTime}} \qquad (2)$$

Using equation. 2, we can determine the number of clock cycles. Taking the clock cycles and using it equation. 1 we can determine the average CPI for a given compiler.

Compiler A CPI = 1.1 CPI

Compiler B CPI = 1.25 CPI

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?

Answer:

$$\text{ExecutionTime}_n = \text{InstructionCount}_n * \text{CPI}_n * \text{ClockCycleTime}_n \quad (3)$$

Since the execution times are the same for A and B then their respective execution time equations are equal:

$$\text{InstructionCount}_a * \text{CPI}_a * \text{ClockCycleTime}_a = \text{InstructionCount}_b * \text{CPI}_b * \text{ClockCycleTime}_b$$

By reducing this equation, $\mathbf{ClockCycleTime_a = 1.36 * ClockCycleTime_b}$. So, clock A is 1.36 times faster than clock B.

c. A new compiler is developed using 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?

Answer: Using eqtn. 3 we can calculate the execution time of this new compiler to be 0.66 seconds. Therefore, the new compiler is

1.66 times faster than Compiler A (1.1 seconds / .66 seconds)

2.27 times faster than Compiler B (1.5 seconds / .66 seconds)

1.10.1 Find the yield for both wafers

Wafer 1:
- Diameter: 15 cm
- Cost: 12
- 84 dies
- 0.020 defects/cm^2

Wafer 2:
- Diameter: 20 cm
- Cost: 15
- 100 dies
- 0.031 defects/cm^2

Yield for a given wafer:

$$\text{Yield} = \frac{1}{\left(1 + \left(\text{Defects per area} * \text{die area}/2\right)\right)^2}$$

Wafer 1 Yield: 0.96 **Wafer 2 Yield: 0.91**

1.10.2 Find the cost per die for both wafers

Cost per die for a given wafer:

$$\text{Cost per die} = \frac{\text{Cost per wafer}}{\text{Dies per wafer} * \text{yield}}$$

Wafer 1 cost per die: 0.149 **Wafer 2 cost per die: 0.16**

1.12

P1
Clock rate: 4 GHz
CPI: 0.9
Instructions: 5.0e9

P2
Clock rate: 3 GHz
CPI: .75
Instructions: 1.0e9

1.12.1

One usual fallacy is to consider the computer with the largest clock rate as having the largest performance. Check if this is true for P1 and P2.

$$\text{Performance} = \frac{1}{\text{ExecutionTime}} \quad (4)$$

$$\text{ExecutionTime} = \frac{\text{CPU\_ClockCycles}}{\text{ClockRate}} \quad (5)$$

$$\text{CPU\_ClockCycles} = \text{Instructions} * \text{CPI} \quad (6)$$

$$\text{Performance} = \frac{\text{ClockRate}}{\text{Instructions} * \text{CPI}} \quad (7)$$

P1 Performance: 0.88
P2 Performance: 4

Even though processor P1 has the largest clock rate, it still has a lower performance than processor P2.

1.12.2

In order to determine the number of instructions P2 can perform in the time it takes P1 to perform 1.0e9 instructions, we first need to determine the time P1 spends on executing 1.0e9 instructions. Using equation 5, we can determine that P1 spends .225 seconds processing 1.0e9 instructions.

By rearranging equation 5 to:

$$\text{Instructions} = \frac{\text{CPUTime} * \text{ClockRate}}{\text{CPI}}$$

We can determine how many instructions P2 processes in .225 seconds. From equation 5, P2 performs 9.0e8 instructions.

1.12.3

To determine the MIPS for each processor we use the following equation:

$$\text{MIPS} = \frac{\text{ClockRate}}{\text{CPI} * 10^6} \qquad (8)$$

P1 MIPS: 4.44e3
P2 MIPS: 4.0e3

P1 has a higher MIPS count, but from problem 1.12.1 we determined that P2 has a better performance, so MIPS is not a reliable metric for comparing the performance power of different processors.