Sean Connor 5 June 2018
Homework 1 605.204.81.SU18

1.2 The eight great ideas in computer architecture are similar to ideas from other fields. Match the eight ideas from computer architecture, “Design for Moore’s Law”, “Use Abstraction to Simplify Design”, “Make the Common Case Fast”, “Performance via Parallelism”, “Performance via Pipelining”, “Performance via Prediction”, “Hierarchy of Memories”, and “Dependability via Redundancy” to the following ideas from other fields:

a. Assembly lines in automobile manufacturing

PERFORMANCE VIA PIPELINING

b. Suspension bridge cables

DEPENDABILITY VIA REDUNDANCY

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

PERFORMANCE VIA PARALLELISM

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

DESIGN FOR MOORE’S LAW

h. Building self-driving cars whose control systems partially rely on existing sensor systems already installed into the base vehicle, such as lane departure systems and smart cruise control systems

USE ABSTRACTION TO SIMPLIFY DESIGN

Sean Connor
5 June 2018
Homework 1
605.204.81.SU18

1.4 Assume a color display using 8 bits for each of the primary colors (red, green, blue) per pixel and a frame size of 1280 × 1024.

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

$$1024 \times 1280 \times 3 \frac{colors}{pixel} \times 8 \frac{bits}{\text{color}} = 31.457\,Mb = 3.932\,MB$$

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

$$\frac{31.457\,Mb}{100 \frac{Mb}{s}} = 0.315\,s$$

Sean Connor
5 June 2018
Homework 1
605.204.81.SU18

1.7 Compilers can have a profound impact on the performance of an application. Assume that for a program, compiler A results in a dynamic instruction count of 1.0E9 and has an execution time of 1.1 s, while compiler B results in a dynamic instruction count of 1.2E9 and an execution time of 1.5 s.

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

$$time = I \times CPI \times c \rightarrow CPI = \frac{t}{I \times c}$$

$$A: CPI = \frac{1.1\text{e}9\, ns}{1.0\text{e}9 \times 1\, ns} = 1.1 \quad B: CPI = \frac{1.5\text{e}9\, ns}{1.2\text{e}9 \times 1\, ns} = 1.25$$

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?

$$t_A = t_B \rightarrow 1.9\text{e}9 \times 1.1 \times c_A = 1.2\text{e}9 \times 1.25 \times c_B \rightarrow \frac{c_B}{c_A} = \frac{1.0\text{e}9 \times 1.1}{1.2\text{e}9 \times 1.25} = 0.733$$

$$\frac{1}{0.733} = 1.364 \text{ times faster}$$

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?

$$time = I \times CPI \times c = 6\text{e}8 \times 1.1 \times 1\, ns = 6.6\text{e}8\, ns = 0.66\, s$$

$$A: \frac{1.1\, s}{0.66\, s} = 1.67 \text{ times faster} \quad B: \frac{1.5\, s}{0.66\, s} = 2.27 \text{ times faster}$$

Sean Connor 5 June 2018
Homework 1 605.204.81.SU18

1.10 Assume a 15 cm diameter wafer has a cost of 12, contains 84 dies, and has 0.020 defects/cm 2 . Assume a 20 cm diameter wafer has a cost of 15, contains 100 dies, and has 0.031 defects/cm2.

1.10.1 Find the yield for both wafers.

Required formulas are:

$$A = \pi r^2 \quad yield = \frac{1}{\left(1+\left(\frac{defects}{area}\times\frac{die\,area}{2}\right)\right)^2} \quad die\,area = \frac{wafer\,area}{dies\,per\,wafer}$$

Combining these we see…

$$y_1 = \frac{1}{\left(1+\left(0.02\times\frac{28.125\pi}{84}\right)\right)^2} = 0.959 \quad y_2 = \frac{1}{\left(1+\left(0.031\times\frac{50\pi}{100}\right)\right)^2} = 0.909$$

1.10.2 Find the cost per die for both wafers.

Required formula is:

$$cpd = \frac{cost/wafer}{(dies/wafer)\times yield}$$

Thus…

$$cpd_1 = \frac{12}{84\times0.959} = 0.149 \quad cpd_2 = \frac{15}{100\times0.909} = 0.165$$

Sean Connor
5 June 2018
Homework 1
605.204.81.SU18

1.12 Section 1.10 cites as a pitfall the utilization of a subset of the performance equation as a performance metric. To illustrate this, consider the following two processors. P1 has a clock rate of 4 GHz, average CPI of 0.9, and requires the execution of 5.0E9 instructions. P2 has a clock rate of 3 GHz, an average CPI of 0.75, and requires the execution of 1.0E9 instructions.

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.

$$time = I \times CPI \times c$$
$$t_1 = \frac{5.0e9 \times .9}{4.0e9} = 1.125s \quad t_2 = \frac{1.0e9 \times .75}{3.0e9} = 0.25s$$

The computer with the lower clock rate performs faster in this particular case.

1.12.2 Another fallacy is to consider that the processor executing the largest number of instructions will need a larger CPU time. Considering that processor P1 is executing a sequence of 1.0E9 instructions and that the CPI of processors P1 and P2 do not change, determine the number of instructions that P2 can execute in the same time that P1 needs to execute 1.0E9 instructions.

$$t_1 = \frac{1.0e9 \times .9}{4.0e9} = 0.225s$$
$$Instructions = \frac{t}{CPI \times c} = \frac{0.225}{0.75 \times (1 \ / \ 3.0e9)} = 9.0e8$$

1.12.3 A common fallacy is to use MIPS (millions of instructions per second) to compare the performance of two diff erent processors, and consider that the processor with the largest MIPS has the largest performance. Check if this is true for P1 and P2.

$$MIPS = \frac{Instructions}{time \times 10^6}$$
$$MIPS_1 = \frac{5.0e9}{1.125 \times 10^6} = 4444.4 \quad MIPS_2 = \frac{1.0e9}{0.25 \times 10^6} = 4000.0$$

As shown before, P2 has greater performance, but in this case the MIPS value for P2 is lower than for that of P1.

1.12.4 Another common performance figure is MFLOPS (millions of floating-point operations per second), defined as

MFLOPS = No. FP operations / (execution time × 1E6)

but this figure has the same problems as MIPS. Assume that 40% of the instructions executed on both P1 and P2 are floating-point instructions. Find the MFLOPS figures for the programs.

This is simply given by multiplying the MIPS figure by 0.4, because of the instructions carried out, 40% are floating point operations and thus MFLOPS will be 40% of MIPS.

$$MFLOPS_1 = .4 \times 4444.4 = 1777.8 \quad MFLOPS_2 = .4 \times 4000 = 1600$$