* 1. List and describe three types of computers.

These three types of computers are:

.Analog computers.

.Digital computers.

.Hybrid computers.

\*Analog computers: These computers were specifically designed to process analog data. For readers who are not familiar with the term, analog data is a type of continuous data that continually changes and does not have discrete values.

\*Digital computers: Digital computers were invented to perform different calculations and logical operations at a very high speed. These computers accept the raw data as input, done in binary numbers (0 and 1) or digitals.

\*Hybrid computers: Hybrid computers are devices that have features of both digital and analog computers. These devices are similar in speed to analog computers and are identical to digital computers in their memory and accuracy.

* 1. a. Assembly lines in automobile manufacturing => **Performance via pipelining**

b. Suspension bridge cables => **Use Abstraction to Simplify Design**

c. Aircraft and marine navigation systems that incorporate wind information => **Performance via parallelism**

d. Express elevators in buildings => **Make the common case fast**

e. Library reserve desk => **Dependability via redundancy**

f. Increasing the gate area on a CMOS transistor to decrease its switching time => **Hierarchy of memories**

g. 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 => **Performance via prediction**

* 1. 1- A special kind of program called a compiler reads the high-level source code and translates it into a program in assembly language

2- Another program called an assembler transforms the program in assembly language into a binary machine language program, which is what a computer understands and can execute directly

Some compilers "cut the middleman" and produce machine code directly

* 1. a) A color display using 8 bits => (red, green, blue) = 24 bits = 3 byte;

A frame size of 1280 \* 1024 => 1280\*1024\*3 = 3,932,160 bytes

So, the minimum size in bytes of the frame buffer to store a frame is 3,932,160 bytes.

b) 100Mbit/s = 10^8 bit/s

Time = size/time = (3,932,160 \* 8)/(10^8) = 0,3145728 s

* 1. a) P1: 3GHz / 1.5 = 2 \* 10^9 instructions per second

P2: 2.5GHz / 1.0 = 2.5 \* 10^9 instructions per second

P3: 4GHz / 2.2 = 1.82 \* 10^9 instructions per second

So P2 has the highest performance among the three

b) P1: 3GHz \* 10 = 3 \* 10^10 cycles

P2: 2.5GHz \* 10 = 2.5 \* 10^10 cycles

P3: 4GHz \* 10 = 4 \* 10^10 cycles

Num of instructions:

P1: 3GHz \* 10 / 1.5 = 2 \* 10^10 instructions

P2: 2.5GHz \* 10 / 1.0 = 2.5 \* 10^10 instructions

P3: 4GHz \* 10 / 2.2 = 1.82 \* 10^10 instructions

c) We have: Execution time = (Num of instructions \* CPI) / (Clock rate)  
 Follow topic: Execution time\*0.7 = (Num of instructions \* 1.2 \*CPI) / New Clock rate

==> New Clock rate = (Num of instructions \* 1.2 CPI) / (Execution time\*0.7) = (Num of instructions \* 1.2\* CPI)\*(Clock rate) / 0.7\* (Num of instructions \* CPI) = (1.2/0.7)\*Clock rate = 1.71\*Clock rate

New Clock rate for each processor:

P1: 3GHz \* 1.71 = 5.13 GHz

P2: 2.5GHz \* 1.71 = 4.27 GHz

P3: 4GHz \* 1.71 = 6.84 GHz

* 1. a) What is the global CPI for each implementation?

Class A : 10^5 instr.

Class B : 2.10^5 instr.

Class C : 5.10^5 instr.

Class D : 2 10^5 instr.

Time = (Instructione\*CPI)/Clock rate

Total time P1 = (10\_x0005\_^5 + 2.10^5 \* 2 + 5.10^5 \* 3 + 2.10^5 \* 3)/(2,5.10^9) = 104 .10^-5 s

Total time P2 = (10^5 \* 2+2.10^5 \* 2+5.10^5 \* 2 + 2.10^5 \* 2)/ (3.10^9) = 66,7.10^−5 s

==> P2 is faster than P1

CPI(P1) = (104.10^−5 \* 2,5.10^9)/10^6 = 2.6

CPI(P2) = (66,7.10^−5 \* 3.10^9)/10^6 = 2.0

b) Find the clock cycles required in both cases.

Clock cycles = Instruction count \* CPI

Clock cycles (P1) = 10^5 + 2.10^5\*2 + 5.10^5\*3 + 2.10^5\*3 = 26.10^5

Clock cycles (P2) = 10^5\*2 + 2.10^5\*2 + 5.10^5\*2 + 2.10^5\*2 = 20.10^5

* 1. a) We have: Execution time = NumOfInstructions\*CPI\*Clock Cycle Time.

=> CPI = Execution time/(NumOfInstructions\*Clock Cycle Time.)

CPI(A) = 1,1s/(10^9\*10^-9) = 1,1

CPI(B) = 1,5s/(1,2\*10^9\*10^-9) = 1,25

b) We have: E(A) = E(B)

<=>(Num of instructions(A) \* CPI(A)) / (Clock rate(A)) = (Num of instructions(B) \* CPI(B)) / (Clock rate(B))

<=>Clock rate(A)/Clock rate(B) = (Num of instructions(A) \* CPI(A)) / (Num of instructions(B) \* CPI(B))

<=>Clock rate(A)/Clock rate(B) = (10^9\*1,1)/(1,2.10^9\*1,25) = 11/15

=> So the Clock rate of A is 27% slower than Clock rate of B

c)

C & A:

Execution time(C) = NumOfInstructions(C)\*CPI(C)\*Clock Cycle Time(C) = 6.10^8\*1,1\*10^-9 = 0,66s

CPU Time of A \* Performance of A = CPU Time of C \* Performance of C

<=>Performance of C/Performance of A = 1,1/0.66 = 1,67

==> So, C is 1.67 times faster than A

C & B:

Execution time(C) = NumOfInstructions(C)\*CPI(C)\*Clock Cycle Time(C) = 6.10^8\*1,1\*10^-9 = 0,66s

CPU Time of B \* Performance of B = CPU Time of C \* Performance of C

<=>Performance of C/Performance of B = 1,5/0.66 = 2,27

==> So, C is 2,27 times faster than A

* 1. a)We have: Power = (1/2).C.V^2.F

=> C = (2.P(Static))/(V^2.F)

C(Pentium 4 Prescott) = (2.90)/(1.25^2\*3,6.10^9) = 3,2.10^-8 F

C(Core i5) = (2.40)/(0,9^2\*3,4.10^9) = 2,9.10^-8 F

b)\*The percentage of the total dissipated power comprised by static power:

[static power/(static power + dynamic power)]\*100  
 Pentium 4 Prescott: [10/(10+90)]\*100 = 10%

Core i5: [30/(30+40)]\*100 = 42,86 %

\* The ratio of static power to dynamic power for each technology.

Static power/dynamic power  
 Pentium 4 Prescott: 10/90 = 0.11 | Core i5: 30/40 = 0.75

c) We have: Power = Static Power(SP) + Dynamic Power(DP)  
 and: SP = U.I = V.I

Pentium 4 Prescott: SP = V.I => I = SP/V = 10/1,25 = 8 A  
 Core i5: SP = V.I => I = SP/V = 30/0,9 = 33,33A  
  
 the total dissipated power is to be reduced by 10%: P(new)/P(old) = 0,9

Pentium 4 Prescott: (V.8+(1/2)\*3,2.10^-8\*V^2\*3,6.10^9)/(10+90) = 0.9

<=> 57,6V^2 + 8V - 90 = 0  
 <=> V = 1,18V or -1,32V | Choose V = 1,18V so we should reduce (1,25 - 1,18) = 0,07V to maintain the same leakage current

Core i5: (V.33,33+(1/2)\*2,9.10^-8\*V^2\*3,4.10^9)/(30+40) = 0.9

<=> 49,3V^2 + 33,33V - 63 = 0  
 <=> V = 0,84V or -1,52V | Choose V = 0,84V so we should reduce (0,9 - 0.84) = 0,06V to maintain the same leakage current