1. **Explain the difference between the concepts of parallelism and pipelining using an example.**
2. **Textbook problem 1.2, Page 54. [10]**
3. **Textbook problem 1.5, Page 55 [10]**
4. **Textbook problem 1.6, Page 55 [10]**

Moore's Law: The number of transistors double about every 2 years. Performance continues up

Abstraction to simplify design: Uses levels with each level hiding the details of the levels below it

Make the common case fast: Areas where the current design is spending the most time

Performance via parallelism: Doing different parts of a task in parallel is faster than doing them sequentially

Performance via pipelining: Handling activities in instruction execution in stages

Performance via prediction: Make an informed guess about the outcome of the condition test

Hierarchy of memories: Fastest, smallest, most expensive memory per pit at the top. Slow, large, and cheap at the bottom

Dependability via redundancy: If one thing fails, it is still recoverable from other things

1) Parallelism is like a plane using more than one engine to fly because it increases performance. An example of pipelining is an assembly line where each person does a specific task then passes it on to the next person instead of doing it all by themselves

2) Matching the eight ideas from computer architecture:

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 – Memory Hierarchy

d. Express elevators in buildings – Abstraction to simplify design

e. Library reserve desk – Performance via prediction

f. Increasing the gate area on a CMOS transistor to decrease its switching time – Make the common case fast

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 – 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 – Parallelism

3) P1 = 3 GHz clock rate & CPI = 1.5 P2 = 2.5 GHz clock rate & CPI = 1 P3 = 4 GHz clock rate & CPI = 2.2

a. P1 = 3/1.5 = 2 IPS * 10^9 (instructions per second) P2 = 2.5/1 = 2.5 * 10^9 IPS P3 = 4/2.2 = 1.818 * 10^9 IPS. Processor 2 has the highest performance

b. If the processors each execute a program in 10 seconds, find the number of cycles and the number of instructions.

   i. P1 = 2 * 10 = 20 * 10^9 instructions. P2 = 2.5 * 10 = 25 * 10^9 instructions. P3 = 4/2.2 * 10 = 18.181 * 10^9 instructions. P1 = 20*1.5 = 30 * 10^9 cycles. P2 = 25 * 1 = 25 *10^9 cycles. P3 = 18.181 * 2.2 = 40 * 10^9 cycles.

c. We are trying to reduce the execution 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?

i. 100% reducing the execution time by 30% = 70% = .7. If CPI increased by 20% then 1 + .2 = 1.2. P1 CPI = 1.2 * 1.5 = 1.8 New CPI. P2 CPI = 1.2 * 1 = 1.2 New CPI. P3 = 1.2*2.2 = 2.64 New CPI. Reducing 30% of 10 seconds is 7 seconds. CPU time = 7 seconds. P1 = 20 * 1.8 / 7 = 5.14 GHz. P2 = 25 * 1.2 / 7 = 4.29 GHz. P3 = 18.181 * 2.64 / 7 = 6.86 GHz.

4) Consider two different implementations of the same instruction set architecture. The instructions can be divided into four classes according to their CPI (class A, B, C, and D). P1 with a clock rate of 2.5 GHz and CPIs of 1, 2, 3, and 3, and P2 with a clock rate of 3 GHz and CPIs of 2, 2, 2, and 2. Given a program with a dynamic instruction count of 1.0E6 instructions divided into classes as follows: 10% class A, 20% class B, 50% class C, and 20% class D, which implementation is faster?

a. What is the global CPI for each implementation?

b. Find the clock cycles required in both cases

1.0E6 = 1 mil

P1 clock cycle = 1.0E6 * 1 * 10% (A) + ← .1

+ 1.0E6 * 2 * 20% (B) ← .4

+ 1.0E6 * 3 * 50% (C) ← 1.5

+ 1.0E6 * 3 * 20% (D) ← .6 .1 + .4 + 1.5 + .6 = 2.6

= 2.6 * 1.0E6 Clock cycles

P2 clock cycle = 1.0E6 * 2 * 10% (A) + ← .2

+ 1.0E6 * 2 * 20% (B) ← .4

+ 1.0E6 * 2 * 50% (C) ← 1

+ 1.0E6 * 2 * 20% (D) ← .4 .2 + .4 + .4 + 1 = 2

= 2 * 1.0E6 clock cycles

CPU time = clock cycle / clock rate

P1 = 2.5 GHz

P2 = 3 GHz

P1 time = 2.6 Clock cycle / 2.5 GHz = 1.04

P2 time = 2 / 3 = .66 repeating

P2 is faster