1. Suppose I design a processor that consumes 65 watts of power (This is total power) while running at 2 GHz. How much dynamic power would the same processor consume if I increase its frequency to 5 GHz? (10 points)

Solution:

Given:

Processor Details:

Power, P = 65 W
Frequencey, f = 2 GHz

We need to find the dynamic power of the processor when the frequency of the processor is increased to 5 GHz.

Assumption: Since the processor is the same, the static power is assumed to be constant. And we assume that the dynamic power takes a major portion of the total power.

Dynamic Power is directly proportional to Frequency, f.

P1/F1 = P2/F2

P2 = P1*F2/F1 = 65*5/2 = **162.5 W**

2. I have a processor X that runs at 2 GHz and a program (Program A) that has a CPI of 3. If the the program completes 5 billion instructions from start to finish, how long did this program run for? I get another processor Y from the market, that has a clock frequency of 4 GHz. Program A on this new processor executes 8 billion instructions with an average CPI of 5. What is the speedup (if any) of program A on processor Y over processor X? (20 points)

Solution:

Given:

Frequency of Processor X = 2 GHz
CPI of Program A = 3
Total number of Instruction for Program A= 5 billion = 5x 10^9.

CPI of Program B = 5
Total number of Instruction for Program B= 8 billion = 8x 10^9.

Execution time = clock cyle time x no. of instructions x CPI

Execution time for Program A = $\frac{5 \, x \, 10^9 x3}{2x10^9}$ **=** $7.5 \, s$

Execution time for Program B = - $\frac{8 \, x \, 10^9 x5}{2x10^9}$ = 10 s

Speed Up = $\frac{Performance \, B}{Performance \, A}$ = $\frac{7.5}{10}$ = **0.75**

3. Assume that I build a processor that runs at 1 GHz. Program A executes 7 billion instructions on this processor and has an average CPI of 1.5. Now, I changed the design of the processor so that I can run it at 2 GHz, dropping the execution time of Program A to a quarter of the original value. What is the average CPI of Program A on the new design? (20 points)

Solution:

Frequency = 1 GHz

No. of instructions for Program A = 7 billion = 7 x 10^9

Average CPI = 1.5

Execution = $\frac{7x \, 10^9 \, x \, 1.5}{1 \, x \, 10^9}$ = 10.5 s

New Execution time = $\frac{10.5}{4}$ = 2.625

CPI = $\frac{New \, Execution \, Time \, \, x \, 2 \, x \, 10^9}{7 \, x \, 10^9}$ = $\frac{2.625 \, x \, 2 \, x \, 10^9}{7 \, x \, 10^9}$ = **0.75**

4. We have a processor with two cores. Program A completes in 4 seconds on core 1 and has a CPI of 8 and program B completes in 5 seconds on core 2 and has a CPI of 5. Both cores run at 1 GHz. What is the combined throughput of the processor? (20 points)

Solution:

Throughput is the maximum number of tasks that can be done per unit time.

Execution = $\frac{CPI \, x \, \#Instructions}{Frequency}$

Instructions of Program A = $\frac{4x10^9}{8}$ =0.5 billion

Instructions of Program B = $\frac{5x10^9}{5}$ = 1 billion

Total Number of Instructions = 1.5 billion

Throughput = Total Number of instructions/Time
= 1.5/(9)
= **0.1667 billion instructions/s**

5. Implement a program to find the factorial of numbers from 1 to 100 in C in the following manner:
    1. Using recursion
    2. Using loop
    3. Using recursion and memoization
    4. Using loop and memoization

And find the speedup of all the programs on your machine by keeping program (1) as the baseline. (80 points)
Tips: Measure the time taken by the program on the CPU using timespec.

Solution:
The code for the question can be accessed here.

	Time (nanoseconds)			
Experiments	**Recursion**	**Loop**	**Recursion and Memoization**	**Loop and Memoization**
1	14386	12331	1600	651
2	20260	12213	968	641
3	24810	11127	1005	667
4	14343	12329	1141	401
5	14453	11197	1590	402
Average	17650.4	11839.4	1260.8	552.4
Speed Up	1	1.49	14	31.95

Remarks:

1. The time taken to complete are in the following order:
- Recursion > Loop > Recursion and Memoization > Loop and Memoization.

2. A drastic reduction in time can be noticed between normal computation and that of the memoized version of the same. In the memoized version, previously computed values of factorial are stored in arrays and is used for finding the next factorial values. Thus reducing the computational time involved in the complexity. and hence reduction in the time.

3. Order of Speedup:

- Recursion < Loop < Recursion and Memoization < Loop and Memoization.