1 a) Amdahl's law states that the speed up that can be achieved on using a particular feature is limited by the fraction of the time that feature is used.

b)

CPI = CPU clock cycles for a program / (instruction count)

i) Suppose the program contains 100 instructions. Therefore, the distribution of instructions would be -

ALU = 35
Load and store = 25
Branches = 25
Floating point = 15

Number of cycles would be -
ALU = 35
Load and store = 50
Branches = 75
Floating Point = 75
Total cycles = 235

CPI = 235/100 = 2.35

ii) Number of cycles new floating point unit would take = 13*15 = 195
Total number of cycles = 355
CPI = 355/100 = 3.55

iii) Best possible speed up -
Overall Speed up is defined as -

Speed up_overall = 1/ ((1 – Fraction_enh) + (Fraction_enh / Speedup_enh))

The floating point corresponds to only 15% of the total instructions
The original processor took 5 cycles to complete it, hence the minimum number of clock cyles possible is 1 cycle. Therefore the enhanced speed up of floating point unit is equal to 5 (as the new processor would be 5 times faster than the original one)

Speed up_overall = 1/((1 - .15) + (.15/5)) = 1.136

2) Parameter	Pipelined	Non-Pipelined
Clock Rate	500 Mhz	350 Mhz
CPI for ALU	1	1
CPI for control	2	1
CPI for mem	2.7	1

i)
ALU = 20%
Control = 10%
Memory = 70%
Faster design would be the one which takes less time to execute the above instruction stream.

Calculations for Pipelined version -
CPI = CPU clock cycles for a program / (instruction count)
For ALU
1 = x/(20) => Number of clock cycles for ALU operation = 20

For Control
2 = y/(10) => Number of clock cycles for control instructions = 20
For Memory
2.7 = z/(70) => Number of clock cycles for memory operations = 189
Total number of cycles = 229
1 CPU clock cycle = 1/(500MHz) = 2 ns
Time taken by pipelined processor = **458 ns**

Non-Pipelined processor -
CPI = CPU clock cycles for a program / (instruction count)
For ALU
1 = x/(20) => Number of clock cycles for ALU operation = 20
For Control
1 = y/(10) => Number of clock cycles for control instructions = 10
For Memory
1 = z/(70) => Number of clock cycles for memory operations = 70
Total number of cycles = 100
1 CPU clock cycle = 1/(350MHz) = 2.857 ns
Time taken by pipelined processor = **285.7 ns**

For above instruction stream the non-pipelined version performs better than the pipelined proceesor

ii)
ALU = 80%
Control = 10%
Memory = 10%
Faster design would be the one which takes less time to execute the above instruction stream.

Calculations for Pipelined version -
CPI = CPU clock cycles for a program / (instruction count)
For ALU
1 = x/(80) => Number of clock cycles for ALU operation = 80
For Control
2 = y/(10) => Number of clock cycles for control instructions = 20
For Memory
2.7 = z/(10) => Number of clock cycles for memory operations = 27
Total number of cycles = 127
1 CPU clock cycle = 1/(500MHz) = 2 ns
Time taken by pipelined processor = **254 ns**

Non-Pipelined processor -
CPI = CPU clock cycles for a program / (instruction count)
For ALU
1 = x/(80) => Number of clock cycles for ALU operation = 80
For Control
1 = y/(10) => Number of clock cycles for control instructions = 10
For Memory
1 = z/(10) => Number of clock cycles for memory operations = 10
Total number of cycles = 100
1 CPU clock cycle = 1/(350MHz) = 2.857 ns
Time taken by pipelined processor = **285.7 ns**

For above instruction stream the pipelined version performs better than the non-pipelined proceesor.

3) Percentage of the time attributed to the disk system = 40%
Average reduction in the latency of the disk system = 50%
Therefore the enhanced speed up achieved = 2 (as the current disk system is twice as fast as the old one)

Speed up_overall = 1/ ((1 – Fraction_enh) + (Fraction_enh / Speedup_enh))
Speed up_overall = 1/ ((1 - .4) + (.4/2)) = 1.25

The perecentage improvement in the processor would be -
%Improvement = 1.25 -1 /1 * 100% = 25%

New Execution Time = Old execution time * 1/(overall speed up)
New execution time = old execution time * 1/1.25 = 0.8* old execution time

Reduction in execution time = (old execution time – new execution time) / (new execution time) * 100 %

Reduction in execution time = .2* 100 % = 20%

Due to the innovations in the disk storage, the current disk system is 100 (maximum) times faster than the old server
This would give us the maximum speed up
Overall Speed Up = 1 / ((1 - .4) + (.4/100)) = ~1.67% (Hence the maximum speed up possible is limited to 1.67% times only)

4) Time spent in floating point addition = 30%
Time spent in floating point multiplication = 60%
Time spent in floating point division = 10%

i) Redesign the floating point adder to make it twice as fast

The overall speed up achieved in this case would be -
Overall speed up = 1/((1-.3) + (.3/2)) = 1.176

ii) Redesign the floating point multiplier to make it thrice as fast

The overall speed up achieved in this case would be -
Overall speed up = 1/((1-.6) + (.6/3)) = 1.667

iii) Redesign the floating point division to make it ten times fast

The overall speed up achieved in this case would be -
Overall speed up = 1/((1-.1) + (.1/10)) = 1.09

The overall speed up is maximum for the optimizations done to floating point multiplier. Hence this sounds like the correct design decision for the processor.

5) Pentium 4 Processor -
Clock Rate = 3.6 Ghz
Voltage = 1.25 V
Static Power = 10W
Dynamic Power = 90 W

Core i5 Ivy Bridge Processor -
Clock Rate = 3.4 Ghz
Voltage = 0.9 V
Static Power = 30W
Dynamic Power = 40 W

i) Average capacitve loads

Dynamic Power = 1/2 * Capacitive loads * (Voltage)2 * Frequency Switched
Pentium 4 Processor -
90 = 1/2 * C * 1.25 * 1.25 * 3.6G => C = 32 nF
Core i5
30 = 1/2 * C * 0.9*0.9*3.4G => C = 21 nF

ii) Percentage of static power dissipated to total power dissipated -
Pentium 4 Processor -
%static power = 10/(10+90) * 100% = 10%
Core i5
%static power = 30/(30+40) * 100% = 42.85%

6)
a) The two major factors for performance improvement are bandwidth and latency. Bandwidth is the amount of work done in given time where as latency is the time between the start and end of an event. Clearly, increasing these are two major performance improvement areas.

b) The two main reasons for shifting to multicore processors from unicore processors in 2003 -
i) The maximum power dissipation of these air cooled chips (the limits on power)
ii) Lack of more instruction level parallelism to exploit efficiency

c) Different classes of computers -

i) Personal Mobile Device (PMD) – This class of computer includes various tablets, cell phones and other wireless as well as multimedia devices. The key design constraints are -
> Cost - It is the prime factor and a lot of design thought is based keeping this in mind as the user won't pay more than a few hundred dollars for this class of computers
> Energy Efficiency – Since most of the PMDs are used on the go, hence the battery should be able to survive at least a day. Hence energy play a major role in the design processs
> Real Time Performance – This class of computer is also used as a multimedia device and hence it is essential that it can play multimedia without any interruptions

ii) Desktop Computing – This includes a major market segment ranging low class of notebook computers to high end workstations. The major design concerns are driven by -
> Price Performance – Since this class of computers are used in a variety of purposes, a major factor which decides its use the cost the user pay for the performance of the device. Hence a lot of optimization is done to get the right amount of performance at a lower cost.

iii) Servers – Servers are basically used for large scale and highly reliable computing needs. There are many factors which comes into picture during the design -

> Availability – Since this class of computer is used in many critical applications (Banks, ATMs) it would be catastrophic if there is a failure of such server systems. Hence it is a must that these servers operate 24 by 7

> Scalability – Scalability in terms of computing power, memory and I/o bandwidth. Since servers are used for large scale computing they should able to meet the ever increasing demands for the services

> Efficient Throughput – Again servers provide computing help to a large number of resources hence a major factor is the amount of time required to service one resource. The overall performance of the server system depends on the time spent on a single transaction

iv) Cluster/Warehouse computers – This includes the collection of desktop computers/servers via a local area network to perform a role of a single huge computing unit. The design concerns for this class of computer includes -

> Price-performace & power – Just like for the desktop computers the price-performance plays a crucial role on the design constraints. Most of the cost of these huge computers are spent on powering them as well as on cooling.

> Availability – Since this class of computers include servers as well, the failure of this system is not acceptable.

v) Embedded Computers – These computers are the one which exist in almost in our everyday needs like referigerator, washing machines, etc. Such computers are closely related to PMDs. Design constraints include -

> Cost & performance – Since these are used in various small end applications it is necessary that the cost of such computers shouldn't be more than a few hundred dollars. Along with low cost these mini computers should be able to execute millions of operations correctly!

d) There are basically two kinds of parallelism which are improving the performance of computers -

> Data-level parallelism – This is due to the fact that a similar operation can be performed on a huge data at the same time

> Task-level parallelism – This comes into the picture as different tasks can be performed independently as well as parallely

These two factors are used in the design to have -
instruction level parallelism, vector architectures, thread level parallelism, request level parallelism

e) The two properties on which Flynn's classified the computer architectures are based upon the number of instruction streams and the number of data streams available in the architecture. Based on this his taxonomy consists of four models. These model are -

> Single instruction stream, single data stream (SISD) – Uniprocessors belong to this category. This simply means that there is a sequential instruction computer with some benefits from instruction level parallelism

> Single instruction stream, multiple data stream (SIMD) – This applies that a similar operation is executed by multiple processors using diffirent (multiple) data streams. These exploit the data-level parallelism.

> Multiple instruction stream, single data stream (MISD) – This simply means that single data is operated on by multiple instructions. Though are there no currently available processors which use such a scheme.

> Multiple instruction stream, multiple data stream (MIMD) – This targets the task level parallelism since each processor has its own instruction stream and uses its own data. These can also exploit data level parallelism by action like SIMD model.

f) Bandwidth – This is defined as the amount of work done by the machine in the given amount of time.
Latency – This is defined as the amount of time taken to do the work from the start to end.
As per the graph mentioned in figure 1.10 the relative latency improvement of memories as compared to bandwidth improvement is much slower. Memories have improved greatly in bandwidht requirements but the latency rate is too high.

g) Techniques used to reduce dynamic power in the modern processors are -

> Less optimization – Dynamic power can be reduced if the in-active could be switched off by turning off the clock to thosse units. One of the example could be the floating point unit when it is not in operation

> Dynamic Voltage and Frequency Scaling – Since the dynamic power is directly proportional to the operating frequency, and with increase in frequency the dynamic power also increases. To reduce this the frequency is reduced during the period of low activity as there is no need to operate at the highest frequency.

> Specific design – For PMDs/mobile processors the design can leverage the fact that these devices would be inactive for some time and thus the desgin can enable them to switch into low power mode. This would help in reducing the dynamic power and would save power when the processor is idle.

Techniques used to reduce static power in the modern processors are -
The main cause of the static power is due to the leakage current which flows through the transistors when when they are not in the active mode (OFF). The static power is directly proportional to the static current and the operating voltage.
To reduce static power, low power devices even cut-off the power supply to modules which are currently not in use. This technique is known as power gating.
Also, if the processor is fast and energy efficient, this would enable other systems on the chip to quicky go to sleep when not in use. This is known as race-to-halt method.

h) Major factors contributing to integrated circuit costs -

> Silicon manufacturing proces – The underlying manufacturing process of integrated circuits is still remains the same, using the wafer chopped into dies which are sent.

The cost of the integrated circuits furhter depends on the following -
i) Cost of die
ii) Cost of die testing
iii) Cost of packaging plus final test
This can be reduced if the test yield is high.

The cost of die further depends on the following criteria -
Cost of die = Cost of wafer / (die yield + Dies per wafer)

Thus, if the dies per wafer are more then the cost could be reduced. This is somewhat dependant on the area the chip consumes. An optimized design should be optimized on Area as well.
The yield is dependant on the number of chips which pass the testing procedure. The number of defects per unit area too contributes to this.
Even when the dies are produced, before packaging the die goes through a series of testing procedure. To rule out bad from the good one. These testing procedures also add to the overall cost of manufacturing.

> Cost of mask set – A separate mask is needed on every process of the integrated circuit manufacturing. This is a large fixed costs (around $1 M) for current modern day processors with around 6 metal layers. This does contribute to the cost of integrated circuits.

i) The availability of the component can be increases by -
Module availability is defined as = MTTF / (MTTF + MTTR)
where,
MTTF – mean time to failure
MTTR – mean time to repair
Clearly, if the MTTF is high then the mean time to failure would be more and hence the product would be available for a longer period of time. That is the reason why availability is considered as on of the most important consideration for design of servers/warehouse computing.
It is also observed that the fault detection techniques could lead to a lower availability. If the machine crashes the program on detection of a transient error this would lead to a lower availability unnecessarily. Thus, there methods which only detect error but do not correct it to avoid reduced availability.

j) The processor performance equation is -

CPU time = (Instruction/Program) x (Cycles/Instruction) x (Seconds/Cycle)
The performance of the processor is dependant on the above three measures -

> Instruction per program – That is the number of instructions required to achieve the correct functionality of the program. Generally, the CISC architectures as compared to RISC have less instructions per program and are thus benefitted by this.

> Cycles per Instruction – The number of cycles it takes to execute the instruction also decides the performance. Given that the CISC architectures have complex instructions thus they take more time to execute it as compared to RISC where the instructions are simple. In general the RISC architectures have less clock cycles per instruction.

> Seconds (Time) per cycle – This is nothing but the clock cycle of the processor. This is dependant on various micro-architecture design needs of the processor. The most criticial path would is one of the factors that decides the clock cycle of the processor and is decided how well the underlying micro-architecture performs.

The above measure helps in deciding the CPU time which in-turns decides the processor performance.