Questionnaire: Exploring Chapter 1 (Computer Abstractions and Technology)

**1.2 Eight Great Ideas in Computer Architecture  
1. What are the "Eight Great Ideas" in computer architecture? Provide a brief explanation of each.**

Moore’s Law design is used by predicting that integrated circuit resources will double approximately every 18-24 months. Abstraction to simplify design is used to help manage complexity by hiding lower-level details and presenting a simpler model at higher levels. Common Case Fast design is used by optimizing the most frequent operations to enhance overall performance more effectively than optimizing rare cases. Performance via Parallelism is designed to perform multiple operations simultaneously to improve performance. Performance via Pipelining is a specific form of parallelism where multiple stages of a process are executed in an overlapping fashion. Performance via Prediction design is a prediction of the outcome of operations. Rather than waiting for certainty. Hierarchy of Memories is designed to balance speed, size, and cost. A memory hierarchy is used where the fastest, smallest, and most expensive memory per bit is at the top of hierarchy. The slowest, largest, and cheapest memory at the bottom. Dependability via Redundancy design is used to ensure dependability, systems include useless components that can take over in case of a failure, enhancing reliability and fault tolerance.

**2. How do these ideas influence the design and functionality of modern computer systems?**

The ideas give us the architectural decisions needed to maximize performance, reliability, and scalability.

**1.3 Below Your Program   
3. What happens "below your program" when you execute a simple command in a high-level programming language? Describe the process in terms of machine language and hardware interactions.**

High-Level Language allows the programmer to code in algebraic notation to make it easy for people to understand. The Compiler translates the high-level language into assembly language. The Assembly Language is a symbolic representation of machine instruction. The Assembler translates each assembly instruction into binary code that corresponds to a machine instruction. The Machine Language is the binary code that directly controls the computer hardware. Hardware Interactions signal electronic components of the computer, such as the CPU, memory, and I/O devices.

**4. Why is it important for computer architects to understand what happens beneath the high-level programming layer?**

It helps them design hardware that runs efficiently and meets the needs of the software. This knowledge enables them to create innovative hardware features, enhance security, and ensure that hardware and software work well together.

**1.4 Under the Covers   
5. What is meant by "Under the Covers" in computer architecture? Provide examples of what this term encompasses.**

It refers to the internal components and workings of a computer that aren't immediately visible to the user. Including the essential hardware components that carry out the core functions of a computer: inputting data, processing data, storing data, and outputting data. For example, input devices like keyboards and microphones feed data into the system, while output devices like displays and speakers provide the results to the user. Inside the computer, the CPU handles data processing, RAM stores data temporarily, and storage devices keep data for long-term use.

**6. How do the elements “under the covers” contribute to overall system performance and efficiency?**

The processor executes instructions and performs calculations. Its speed and architecture directly impact how quickly tasks are completed. Memory provides fast access to the data the processor needs in real-time. This allows for smoother multitasking and quicker program execution.

**1.5 Technologies for Building Processors and Memory  
7. What are the key technologies used in building processors and memory today? Compare and contrast at least two different technologies.**

Transistors and integrated circuits are the key technologies. Especially very large-scale integrated circuits. Transistors act like tiny switches that control electrical signals and are the basic building blocks for processors. Integrated circuits take combining many of these transistors into one small chip. This helps make designs more compact and efficient, which is crucial for modern tech.

**8. How have advances in processor and memory technology impacted the development of modern computing devices?**

Manufacturing techniques have improved processors to become faster and more efficient, while memory has expanded. This progress means our devices can handle more complex tasks, run smoother, and offer advanced features, driving the rapid growth and innovation we see in technology today

**1.6 Performance  
9. What metrics are commonly used to measure computer performance? Explain how these metrics are calculated and why they are important.**

Architects want to know how quickly a task can be completed or how much work on the computer can be done over time. The Execution Time determines how long it takes for a computer to complete a task from start to finish. The Bandwidth is determined by how much work computers can handle within a given time. The Clock Cycle Time is determined by how quickly the computer’s clock ticks. Faster ticks usually mean better performance. Clock Rate is determined by the speed of these ticks, usually measured in GHz.

**10. How does the choice of processor architecture affect overall system performance?**

The processor architecture affects how well a processor handles tasks. It includes the types of instructions the processor can execute, how quickly it operates, and how efficiently it processes each instruction. A smartly designed architecture means the processor can work faster with fewer delays, which translates to better performance overall.

**1.7 The Power Wall   
11. What is the "Power Wall," and why is it a significant challenge in computer architecture?**

Processors have become faster and consumed more power and generated more heat, creating a challenge for cooling systems and power supplies. This problem restricts the ability to boost performance solely through increasing clock speeds, making it a significant hurdle in computer architecture.

**12. What strategies have been employed to overcome the limitations imposed by the Power Wall?**

Architects focused on making processors more energy efficient. They use multi-core processors to spread out tasks across several cores, which boosts performance without drastically increasing power use. Additionally, improvements in cooling systems and power management help handle the extra heat and power demands of high-performance processors.

**1.8 The Sea Change: The Switch from Uniprocessors to Multiprocessors   
13. Why did the computing industry shift from uniprocessors to multiprocessors?**

There was a limitation of continuation of increased clock rates for single processors. Instead of continuing to push clock speeds, the industry adopted multiprocessor systems to enhance performance through parallelism. By incorporating multiple cores on a single chip, processors could handle more tasks simultaneously, thereby improving overall throughput.

**14. Discuss the advantages and challenges associated with multiprocessor systems.**

Multiprocessor systems bring a lot of benefits by letting multiple cores work on tasks at the same time. This parallel processing means better overall performance and faster handling of complex jobs, especially when software is designed to take advantage of multiple cores. By adding more cores, these systems can handle more demanding workloads efficiently, making them great for both high-performance tasks and everyday computing. However, switching to multiprocessor systems also comes with some hurdles. Programming these systems can be tricky because it requires managing how tasks are divided among cores, keeping them in sync, and handling communication between them. Balancing the workload is essential—if one core is overburdened while others are idle, performance can suffer. Plus, not all software is built to use multiple cores effectively, which might limit the performance improvements for some applications.

**1.9 Real Stuff: Benchmarking the Intel Core i7   
15. What is benchmarking, and why is it essential in evaluating processors like the Intel Core i7?**

Benchmarking is used to process a set of standardized tasks and to compare with other systems. The Intel Core i7 provides objective data on how well the processor performs under different workloads. By using benchmarks, we can assess various aspects of processor performance, such as speed, efficiency, and handling of complex tasks. These standardized tests help to ensure that comparisons between different processors are fair and consistent, allowing users to make informed decisions based on reliable performance metrics.

**16. Describe the key findings from benchmarking the Intel Core i7. What does this reveal about its performance?**

The Intel Core i7 performed well across a range of applications. Specifically, the Core i7 demonstrates strong performance in tasks such as video compression and gene sequence searching, with execution times and SPEC ratios reflecting its high processing power and efficiency. Overall suitable for both everyday computing and demanding applications.