Week 1

a. Some of the most significant things I learned in the first week were abstraction, gate logic, computer systems, and high-level programming. It is important to comprehend the layers involved and how components at higher abstraction layers use components from lower layers without requiring knowledge of the fine details.

b. Although it goes deeper into the internal workings, interactions of a computer system, and programming while the usefulness of abstraction and how they help to make complicated systems simpler, this idea is still connected to what I already know about the fundamentals of computers.

c. My course team wants me to learn this because, any field involving technology or computing needs to have a basic understanding of how computers work, which can be acquired by studying computer systems and abstractions, in order to write effective code and comprehend how software interacts with hardware at various levels, it is imperative that we understand operating systems, abstractions, and high-level programming.

Week 2

a. Logic gate implementation, HDL programming, and hardware simulations were the main topics. This week's most important lesson was learning the procedures needed to put a logic gate into practice. Another significant learning point was the importance of Hardware Description Languages (HDLs) in validating correctness and performance before building an actual chip.

b. With my experience in computer systems and hardware design, I can relate to the fundamental building blocks of digital circuit design. These ideas are essential to comprehending contemporary computing from the ground up, giving students the knowledge and abilities, they need to create accurate and efficient real-world applications. This relates to my existing knowledge of programming languages and their role in software development.

c. Acquiring knowledge of this is crucial as it establishes the foundation for designing more intricate hardware and comprehending digital systems for diverse uses. Through the acquisition of these abilities, learners can adopt a methodical approach to hardware design, guaranteeing resilient and enhanced solutions for diverse computing requirements. It also fits with the course’s objective of creating a contemporary computer from the ground up. I can develop a more thorough grasp of computer systems design and development by learning HDL programming, which will help me understand how software and hardware interact.

Week 3

a. The most important topics in this week are the binary numbers and logic gates to build adders and an ALU. Then there was two's complement which is a way to represent negative numbers in binary. I also learned how to use HDL and hardware simulator to design and test circuits.

b. I can relate to the fundamentals of binary numbers and doing mathematical operations with them which I learned back in school but in a much deeper perspective and the main reason behind it and the root cause for the two's complement made me realize how important they are in the real-world application such as in encryptions methods.

c. This knowledge is essential for any computer science student, as it will help us appreciate the design and implementation of modern hardware and software systems. It will also enable us to apply your skills in various domains, such as robotics, artificial intelligence, cryptography, and more. Learning this topic will also develop our logical thinking, problem-solving, and creativity, which are valuable skills for any profession.

Week 4

a. In week 4 some of the most important things I learned were the concepts of registers, exploration of sequential logic, and delving into the fundamental understanding of clocks and clock cycles.

b. This relates to my existing knowledge as it emphasizes the significance of memory units, expands the understanding of combinational logic by introducing the concepts of memory elements, and directly relates to what I already know about the synchronization and coordination of tasks in a computer system.

c. The course team wants me to learn this, so that I can comprehend the foundational role of memory in computer systems and its impact on the execution of instructions and data storage which provides a strong foundation of the system's architecture, then learn about the sequential logic to lay the groundwork for getting the idea of how memory elements are integrated with combinational logic to create more complex systems and functions within a computer system, and finally for the seamless operation and control of various components within a computer systems.

Week 5

a. This week I learnt about memory hierarchy, how different types of memory devices are organized and accessed in a computer system, then machine language, where low-level instructions are encoded and executed by the CPU and how they manipulate data and memory locations, then registers and addressing modes, how the CPU uses registers to store and operate on data, finally, program counter and flow control, how the CPU keeps track of the next instruction to be executed.

b. I can relate to this from my existing knowledge about the memory of the CPU, and how the machine language can change according to the instruction given to the computer. I can also relate to the memory hierarchical which I had experience before but after this the understanding of memory hierarchy has improved a lot.

c. By learning the basics of computer architecture and organization, we can gain a deeper understanding of how computers work, and how to design and develop software that can run efficiently and effectively on different platforms. This can also help us to appreciate the challenges and trade-offs involved in computer design, and to explore the possibilities and limitations of computation.

Week 6

a. This week some of the important things I learnt were The Hack Computer, an overview of the hardware where Hack Machine Language is operated on. The Hack Machine Language, a low-level language that has two types of instructions: A and C instructions. The hack Assembler which translates symbolic hack programs into binary code. the CPU Emulator which simulates the hack computer, and the input/output devices the hack computer interacts with.

b. From this I can relate to the computer systems, hardware, and HDL programming I learnt before, I can see parallel operation between of the hack computers as a simplified model of a real computer, and the hack machine language as a basic form of communication that the computer understands.

c. They want me to learn this so that I can know how a computer hardware components interact and how program run on a machine, can gain deeper understanding of both low-level and high-level programs, learn to use assemblers and emulators, which will be essential for translating human-readable code into machine-executable instructions and testing and debugging code, and finally they want me to understand how computers interact with the external devices to program anything that involves user interaction or data input/output.