Victoria University of Bangladesh

Final-Term Examination– Fall 2022

Submitted To:

Syeda Tithi

Lecturer

Victoria University of Bangladesh

Submitted By:

Name: Anny Konika Das,

ID: 2120190011

Program: B.Sc in CSE.

Semester: Fall-2022.

Course Title: Computer Architecture

Course Code: CSE-313.

1.NO.Qus.Ans

a)ans: : The IBM z EC12 I/O channel subsystem is the central component of the input/output (I/O) system in IBM's mainframe computers. It provides communication between the central processing unit (CPU) and the attached peripheral devices, such as disk drives and tape drives. The I/O channel subsystem is organized into several distinct components, each with a specific function.

1. I/O Processors (IOPs): Responsible for managing and coordinating the flow of data between the CPU and I/O devices.
2. Channels: Provide a high-speed path between the IOPs and I/O devices
3. Control Units (CUs): Manage the physical devices and perform functions such as data buffering, error correction, and device control.

4.Subchannel Sets (SCSSs): A group of subchannels that are associated with a single control unit.

5. Subchannels: Provide a communication path between the IOPs and control units.

6.Device Drivers: Software components that interact with the control units to manage I/O operations.

7. Channel Subsystem Controller (CSC): The CSC is responsible for managing the flow of data between the CPU and the peripheral devices. It acts as the central hub, directing incoming and outgoing data to the appropriate destination.
8. Channel Paths: Channel paths are the physical connections between the CSC and the peripheral devices. They provide the communication pathways for the data.
9. Control Units: Control units are specialized devices that manage the flow of data to and from the peripheral devices. They work in conjunction with the CSC to ensure that data is transmitted and received correctly.
10. Input/Output Processors (IOPs): IOPs are specialized processors that handle I/O operations for the channel subsystem. They are responsible for managing the flow of data between the peripheral devices and the main memory.

11. Peripheral Devices: Peripheral devices are the physical components, such as disk drives and tape drives, that are connected to the I/O channel subsystem. They provide storage for data and programs, and allow for input and output operations.

All these components work together to provide a high-performance, scalable, and flexible I/O infrastructure for IBM z EC12 systems. Together, these components form the IBM z EC12 I/O channel subsystem, a highly integrated and efficient system that provides fast and reliable communication between the CPU and the peripheral devices.

b)ans:

When an I/O device completes an I/O operation, the following sequence of hardware events occur:

1. Interrupt Request: The I/O device generates an interrupt request to the computer's interrupt controller to indicate that the I/O operation is complete.
2. Interrupt Acknowledge: The interrupt controller sends an interrupt acknowledge signal to the I/O device to confirm receipt of the interrupt request.
3. Context Switch: The operating system saves the current state of the CPU and switches to a context of the operating system to handle the interrupt.
4. Interrupt Service Routine (ISR): The operating system invokes the corresponding Interrupt Service Routine (ISR) associated with the I/O device.
5. I/O Completion: The ISR retrieves the data from the I/O device and updates relevant data structures in the operating system, such as the I/O queue.
6. Interrupt End: The ISR sends an end-of-interrupt signal to the interrupt controller to indicate that the interrupt handling is complete.
7. Acknowledgement: The ISR sends an acknowledgement to the I/O device to indicate that the I/O operation is complete.
8. Context Restore: The operating system restores the saved state of the CPU and returns control to the original process.
9. Interrupt Masking: The interrupt controller may mask the interrupt from the I/O device to prevent further interrupts until the previous interrupt has been fully processed.
10. Interrupt Clearing: The interrupt controller may clear the interrupt request from the I/O device to prevent re-triggering of the same interrupt.

These events occur rapidly and in a specific order to ensure that I/O operations are handled efficiently and effectively by the computer system.

c)ans: There are several techniques for inputting data into a block diagram, here are three common ones:

1. Direct Input: This method involves manually entering data into the input blocks of a block diagram. This technique is suitable for small systems with simple data inputs, where manual data entry is feasible. The data can be entered using a keyboard or other input device, such as a mouse or touch screen.
2. File Input: This method involves reading data from a file and using it as input to the block diagram. This technique is useful for processing large amounts of data, where manual data entry is not feasible. The file can be in various formats such as text, binary, or spreadsheets. The data can be easily managed and updated by changing the contents of the file.
3. Device Input: This method involves reading data from an external device, such as a sensor or a keyboard, and using it as input to the block diagram. This technique is useful for real-time systems where data needs to be acquired in real-time from a device. The device can be connected to the computer using various interfaces, such as USB, serial, or Ethernet, and the data can be transmitted using protocols such as TCP/IP or Modbus.

These techniques can be used in combination or separately, depending on the specific requirements of the system.

The choice of input technique will depend on factors such as the size and complexity of the system, the type of data being input, and the desired level of automation. The input technique should be chosen based on the trade-off between the cost, complexity, and performance of the system.

2.NO.Qus.Ans:

a)Ans: Cache and main memory are two important types of computer memory used to store data and instructions.

Cache memory is a small, fast memory system that acts as a buffer between the CPU and main memory. It stores frequently accessed data and instructions to reduce the number of accesses required from the slower main memory. Cache memory is much faster than main memory because it is physically closer to the CPU, reducing the time it takes for data to be transferred.

Main memory, also known as RAM (Random Access Memory), is a larger, slower memory system that is used to store data and instructions for immediate use by the CPU. Main memory stores the data and instructions that are currently being used or

processed by the CPU. Unlike cache memory, main memory is not integrated into the CPU and is instead located on a separate memory module.

In summary, cache memory is used to store frequently accessed data and instructions to improve the performance of the system, while main memory is used to store data and instructions for immediate use by the CPU. Cache memory is faster than main memory, but it is also smaller in size and stores only a portion of the data stored in main memory.

Both Cache and Main Memory are essential components of a computer system, working together to provide fast and efficient access to data and instructions.

b)ans: A cache is a small and fast memory system that stores frequently accessed data to reduce the number of reads from a slower storage system like a hard disk drive or a solid-state drive. There are two main types of caches: logical and physical.

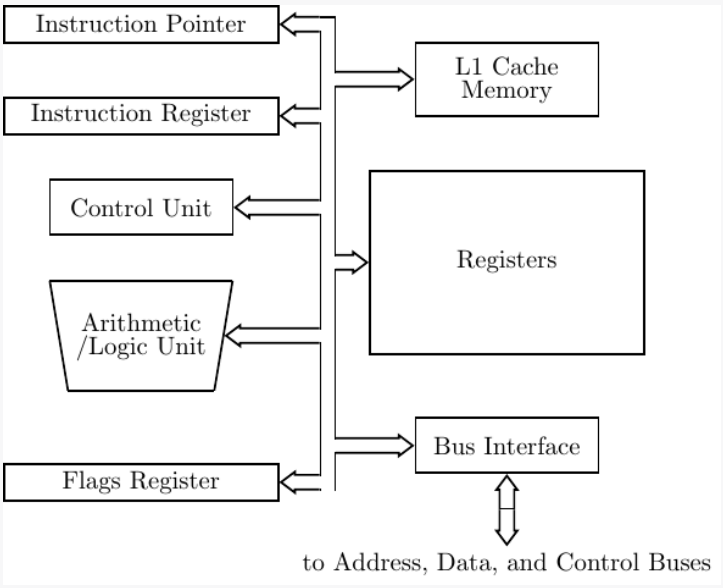
Logical cache: A Logical cache is a type of cache used by software applications and operating systems to improve the performance of frequently accessed data. It is a mapping between logical memory addresses used by a program and the physical memory addresses of the hardware. A logical cache is a software-based cache that is used by operating systems, web browsers, and other software applications to cache frequently used data. This type of cache is stored in the system's RAM and is managed by the operating system or the application.

Physical Cache: Physical cache refers to the small, fast memory chips on the processor or motherboard of a computer that store frequently accessed data from the main memory. This type of cache serves as a buffer between the processor and main memory, allowing the processor to quickly access frequently used data without having to access the slower main memory. Physical caches are typically organized into levels, with the level 1 (L1) cache being the smallest and fastest, and the level 2 (L2) and level 3 (L3) caches being progressively larger and slower. A physical cache, on the other hand, is a hardware-based cache that is integrated into a computer's central processing unit (CPU). This type of cache is much faster than a logical cache as it is physically located on the CPU itself, reducing the distance the data must travel to be processed. Physical caches are typically smaller in size than logical caches, but they provide a significant performance boost for a system.

In summary, the main difference between logical and physical caches is that the former is software-based and managed by the operating system or application, while the latter is hardware-based and integrated into the CPU.

3.NO.Qus.Ans:

a)Ans: The main components of a central processing unit (CPU) and their interactions with the main memory and I/O devicesThe main components of a CPU are the following:



The central processing unit (CPU) is the heart of a computer system, and it is responsible for executing instructions and processing data. The main components of a CPU are the Arithmetic and Logic Unit (ALU), the Control Unit (CU), and the Registers.

The ALU performs arithmetic operations like addition, subtraction, and multiplication, as well as logical operations like bit shifting and comparison.

The CU controls the flow of data and instructions in the CPU and determines which operation to perform based on the instruction being executed. It also communicates with other components of the computer, such as the main memory and I/O devices.

The Registers are high-speed memory units within the CPU that store data temporarily while it is being processed.

When a program is executed, the CPU retrieves the instructions and data from the main memory. The CPU then processes the data and sends the results back to the main memory or to an I/O device for storage or further processing.

For example, if a program requires input from a user, the CPU will send a request to the I/O device for input. The I/O device will receive the input, send it to the main memory, and then the CPU will retrieve it and use it for processing. Similarly, if the program requires data to be stored, the CPU will send the data to the main memory or to an I/O device for storage.

In summary, the CPU interacts with the main memory and I/O devices to retrieve instructions and data, perform operations, and store results. These interactions are essential for the functioning of a computer system.

Regenerate response

- Arithmetic Logic Unit (ALU): performs arithmetic and logical operations on data.
- Control Unit (CU): directs and coordinates the operations of the other components in the CPU.
- Registers: small, fast storage locations within the CPU that temporarily hold data and instructions.

The CPU interacts with the main memory and I/O devices as follows:

- Main Memory: the CPU retrieves data and instructions from the main memory to execute them. It also stores results back into the main memory.
- Input/Output (I/O) Devices: the CPU receives input from I/O devices and sends output to them, such as keyboard input, mouse input, and display output.

The CPU communicates with these components through a set of control and data buses that carry signals between the components.

b)Ans: Indirect addressing is a mode of addressing used in the x86 architecture family of microprocessors that allows a memory address to be stored in a register or memory location, and then used to access the actual data. In other words, the address of the

data is stored in a register or memory location, and the actual data is accessed by using the value stored in that register or memory location.

For example, consider the following x86 instruction:

MOV AX, [BX]

In this instruction, the data stored at the memory location pointed to by the value in the BX register is moved into the AX register. The square brackets around the BX register indicate that the data is being accessed indirectly.

In indirect addressing, the contents of a register or memory location can be used as an address to access data, providing greater flexibility in accessing data stored in memory. Indirect addressing is commonly used in x86 programming, particularly in low-level system programming and assembly language programming.

“THE END”