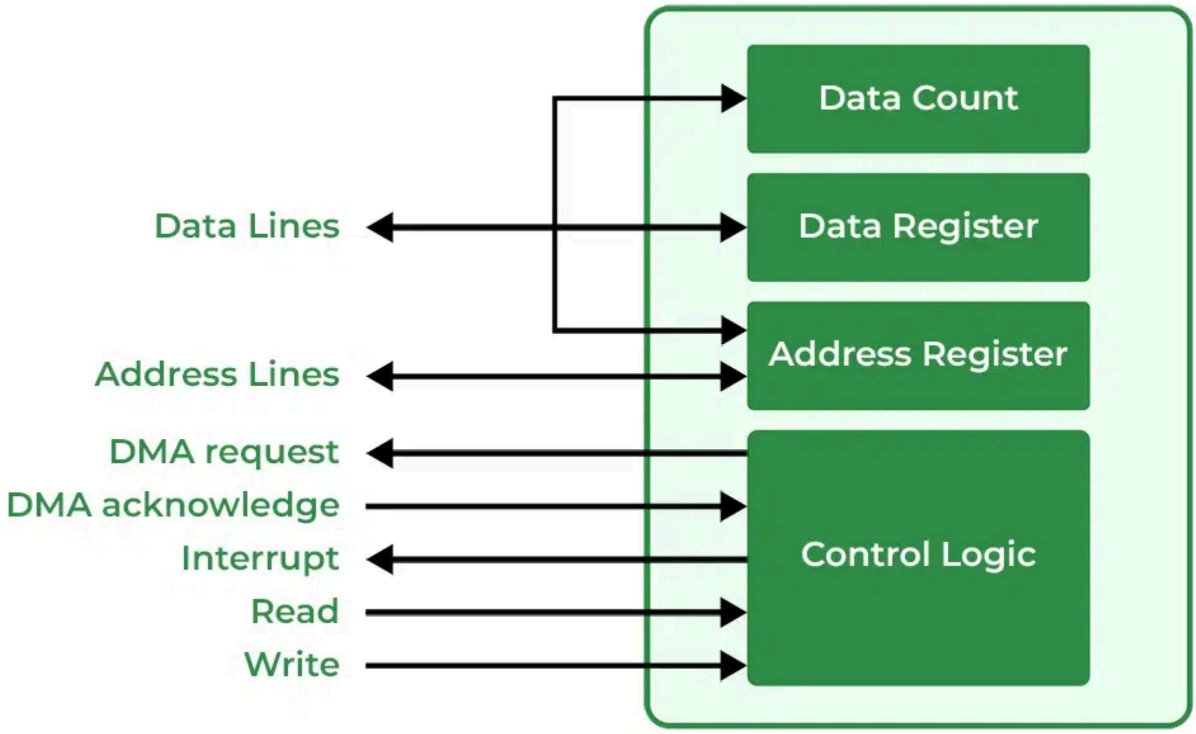
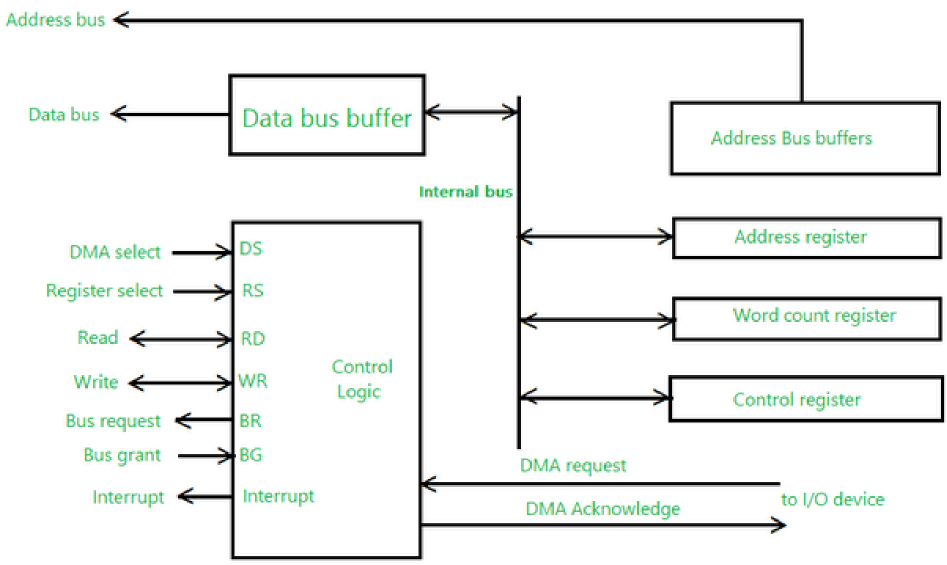
1. Explain DMA (Direct Memory Access) with diagrams.
   DMA Controller is a type of control unit that works as an interface for the data bus and the I/O Devices. As mentioned, the DMA Controller has the work of transferring the data without the intervention of the processors, processors can control the data transfer. DMA Controller also contains an address unit, which generates the address and selects an I/O device for the transfer of data. Here we are showing the block diagram of the DMA Controller.



   There are different types of DMA:
   - Burst Mode: Transfers data in bursts, blocking CPU access to memory.
   - Cycle Stealing Mode: CPU and DMA alternate memory access.
   - Transparent Mode: DMA only transfers data when the CPU is idle.



Working of DMA

- The CPU initiates the DMA transfer by configuring the DMA controller with the source address, destination address, and the amount of data to be transferred.
- The DMA controller takes control of the system bus and performs the data transfer between memory and the I/O device.
- Depending on the mode, the CPU may or may not be halted during the transfer.
- Once the data transfer is complete, the DMA controller sends an interrupt signal to the CPU, indicating the transfer is done, and the CPU can continue its operation.

2. What are the advantages and disadvantages of DMA?
   Advantages
   - Offloads CPU: By handling data transfers directly between memory and devices, DMA frees the CPU from spending time on data movement, allowing it to focus on more critical tasks.
   - Faster Transfers: DMA transfers data more quickly than CPU-managed transfers, particularly in cases involving large blocks of data.
   - Efficient Use of Resources: In modes like cycle stealing or transparent mode, both CPU and DMA can operate concurrently, making efficient use of system resources.

   Disadvantages
   - Increased Hardware Complexity: Requires additional circuitry, making the system more complex and expensive.
   - Bus Contention: CPU and DMA may compete for memory access, slowing down performance.
   - Interrupt Overhead: Frequent interrupts can reduce CPU efficiency in handling tasks.
   - Difficulty in Debugging: Independent DMA operations make tracing data transfer issues challenging.
   - Latency Issues: Cycle stealing can introduce small delays, affecting real-time performance.
3. What is meant by Synchronisation in computer architecture?
   Synchronization is the mechanism that ensures that multiple processes, threads, or devices in a system work together in harmony when accessing shared resources such as memory, data, or hardware components. Without synchronization, concurrent access to shared resources can lead to race conditions, data corruption, or unexpected behaviours.
4. What are the various synchronisation techniques?
   - Locks: A mechanism that ensures mutual exclusion, allowing only one thread or process to access a shared resource at a time.
   - Semaphores: A synchronization tool that controls access to resources by using a counter to limit how many threads can access a resource simultaneously.
   - Monitors: A high-level synchronization construct that provides mutual exclusion and condition variables to manage access to shared resources.
   - Barriers: A synchronization technique that ensures multiple threads or processes wait at a common point until all participants reach that point.
   - Message Passing: A method where processes communicate and synchronize by exchanging messages, typically used in distributed systems.
5. What is Paging? What is a Page table?
   Paging is a memory management scheme that eliminates the need for contiguous allocation of physical memory by dividing both virtual memory and physical memory into fixed-size blocks. Virtual memory is divided into pages, and physical memory is divided into frames.A page table is a data structure maintained by the operating system that stores the mapping between virtual page numbers and corresponding physical frame numbers.

6. Find the total number of frames if the size of the main memory is 2^{30} bytes, the page size is 4KB and the size of each page table entry is 32-bit.
   Main Memory Size: 2^{30} bytes = 1 GB.
   Page Size: 4KB = 2^{12} bytes.
   Number of Frames: The total number of frames is calculated as:
   Total Frames = Main Memory SizePage Size\Total Frames
   Substituting the values:
   Total Frames =$2^{30}/2^{18}$=2^{12} bytes.
7. Consider a system with page table entries of 8 bytes each. If the size of the page table is 256 bytes, what is the number of entries in the page table?
   Size of Page Table: 256 bytes.
   Size of Each Page Table Entry: 8 bytes.
   Number of Entries = Size of Each Page Table Entry/Size of Page Table = 32
8. Consider a machine with 32-bit logic addresses, 4KB page size and page table entries of 4 bytes each. Find the size of the page table in bytes. Assume the memory is byte-addressable.
   Address Space:
   Logical Addresses: 32 bits
   Number of Pages: 2^{32} logical addresses / 4KB page size
   Number of pages = 2^{20}
   Size of Each Page Table Entry: 4 bytes
   Size of Page Table = Number of Entries × Size of Each Entry =2^{22} bytes i.e. 4