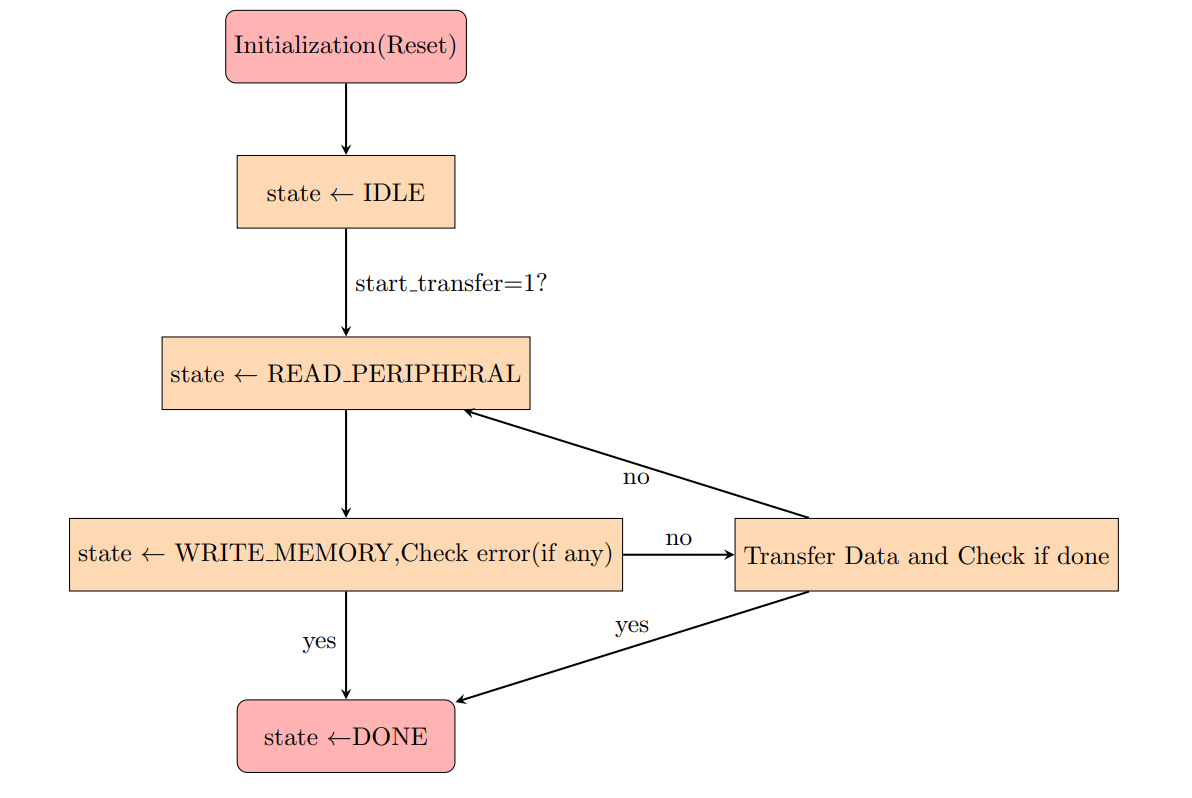
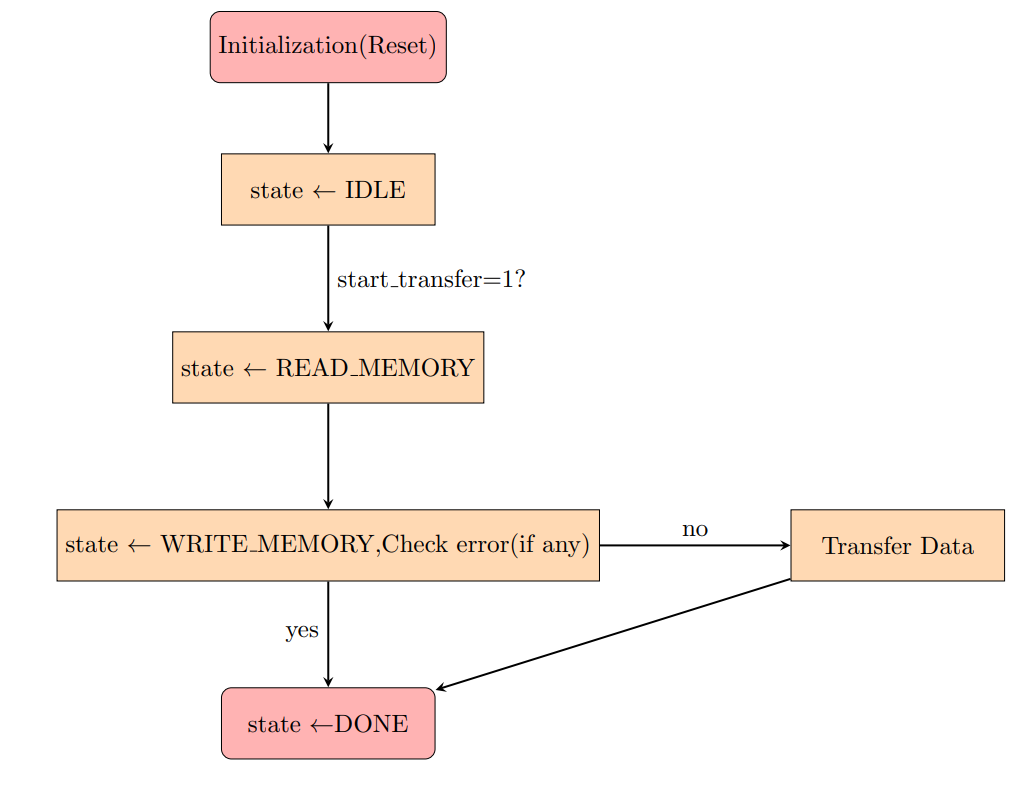
**QDMA**

QDMA, or Queue-based Direct Memory Access, is a way to handle data movement between  
memory, peripherals, and devices without any CPU intervention, by queuing up tasks and letting the hardware managethe transfers directly. This technique makes it possible to send large amounts of data over the network without overwhelming the processor, as in traditional methods. Through this technique, peripheral asks for DMA controller through the DMA request to transfer data to the memory. DMA hardware negotiates memory buses to send a HOLD request to CPU for less of interference during the process. To this Diana, CPU has an immediate reaction, by issuing the HOLD acknowledgement and initiating the DMA process by sending the starting memory block address, word count, read/write signal, and the start transfer signal. After that, the peripheral will accept the signal of DMA acknowledgement and transmit the data to the buffers DMA. Following the data transportation from peripheral to memory, CPUremanues its sign-exchange operations with the memory.  
  
  
**2. Working Principles of Block Transfer Technique**  
The module block transfer dedicates itself to the operation of data transfer between a peripheral buffer and memory. It uses a finite state machine (FSM) with four states: potent, monologues, macro epode, reading from a in structure, and production completion. When start transfer, the module is enabled and begins to transfer. For example, it gets various data from I/O devices peripheral buffer, then detects/corrects possible errors like undefined data and writes valid data into the memory via addressing increment. However, it is continued until it either reached the required number of words, or an error is returned. It then, broadcasts a signal to show that the payment has been conducted successfully.



1. **Working Principle of Continuous Transfer Method**

The process of the continuous data exchange between Memory 2 and Memory 3 is the Verilog module continuous\_transfer, where the Verilog module is activated by the transfer request signal. It employs a finite state machine (FSM) with four states: IDLE, READ\\_MEMORY, WRITE\\_MEMORY and DONE. On startup it fetches data present in Memory 2 and stores it in Memory 3. It serves to compensating for the read error and setting flags in accordance with the rule. On completion of the transfer, it returns the transfer\_done event to signal it has been successful.



1. **Working Principle of QDMA with both Block andContinuous Transfer Method**

The Verilog module qdma substantially optimizes data transfers within a digital system by submitting custom memory-to-memory accesses to the inter-connection network rather than the regular external I/O-to-memory traffic. When a transfer request from internal memory like Memory 2 to Memory 3 is received, for example, qdma concentrates on executing this task by performing all operations it inherently includes. It does this is by getting Memory 1 data and transfers it in the memory cycle to Memory 3 without disturbing the data processing. On the other hand, if an external I/O-to-memory transfer request is made while an internal did is being processed, qdma shall temporarily pause the subsequent transfer process, direct the internal data from the did and continue with the the internal transfer operation.

