Direct Memory Access (DMA)

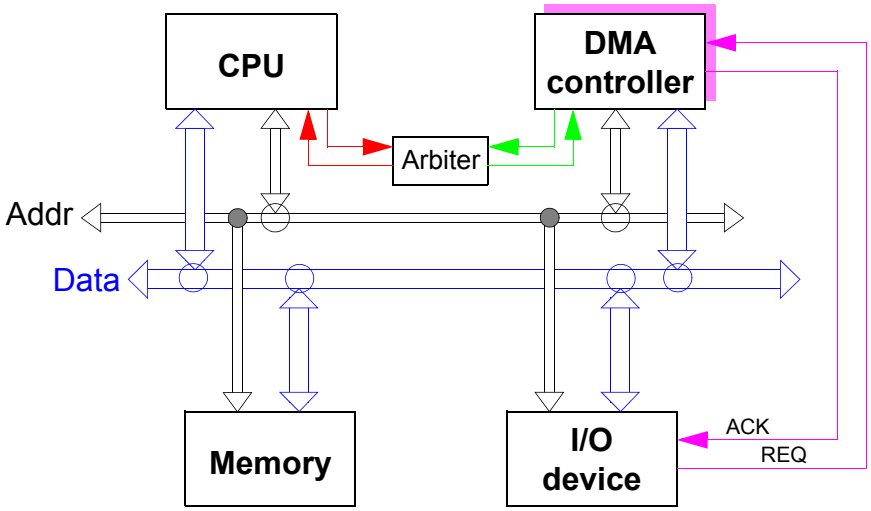
DMA Basics

Definition : ***A direct memory access (DMA) is an operation in which data is copied (transported) from one resource to another resource in a computer system without the involvement of the CPU.***

The task of a DMA-controller (DMAC) is to execute the copy operation of data from one resource location to another. The copy of data can be performed from:

- I/O-device to memory
- memory to I/O-device
- memory to memory
- I/O-device to I/O-device

A DMAC is an independent (from CPU) resource of a computer system added for the concurrent execution of DMA-operations. The first two operation modes are 'read from' and 'write to' transfers of an I/O-device to the main memory, which are the common operation of a DMA-controller. The other two operations are slightly more difficult to implement and most DMA-controllers do not implement device to device transfers.



simplified logical structure of a system with DMA

The DMAC replaces the CPU for the transfer task of data from the I/O-device to the main memory (or vice versa) which otherwise would have been executed by the CPU using the programmed input output (PIO) mode. PIO is realized by a small instruction sequence executed by the processor to copy data. The 'memcpy' function supplied by the system is such a PIO operation.

The DMAC is a master/slave resource on the system bus, because it must supply the addresses for the resources being involved in a DMA transfer. It requests the bus whenever a data value is available for transport, which is signaled from the device by the REQ signal.

The functional unit DMAC may be integrated into other functional units in a computer system, e.g. the memory controller, the south bridge, or directly into an I/O-device.

Direct Memory Access (DMA)

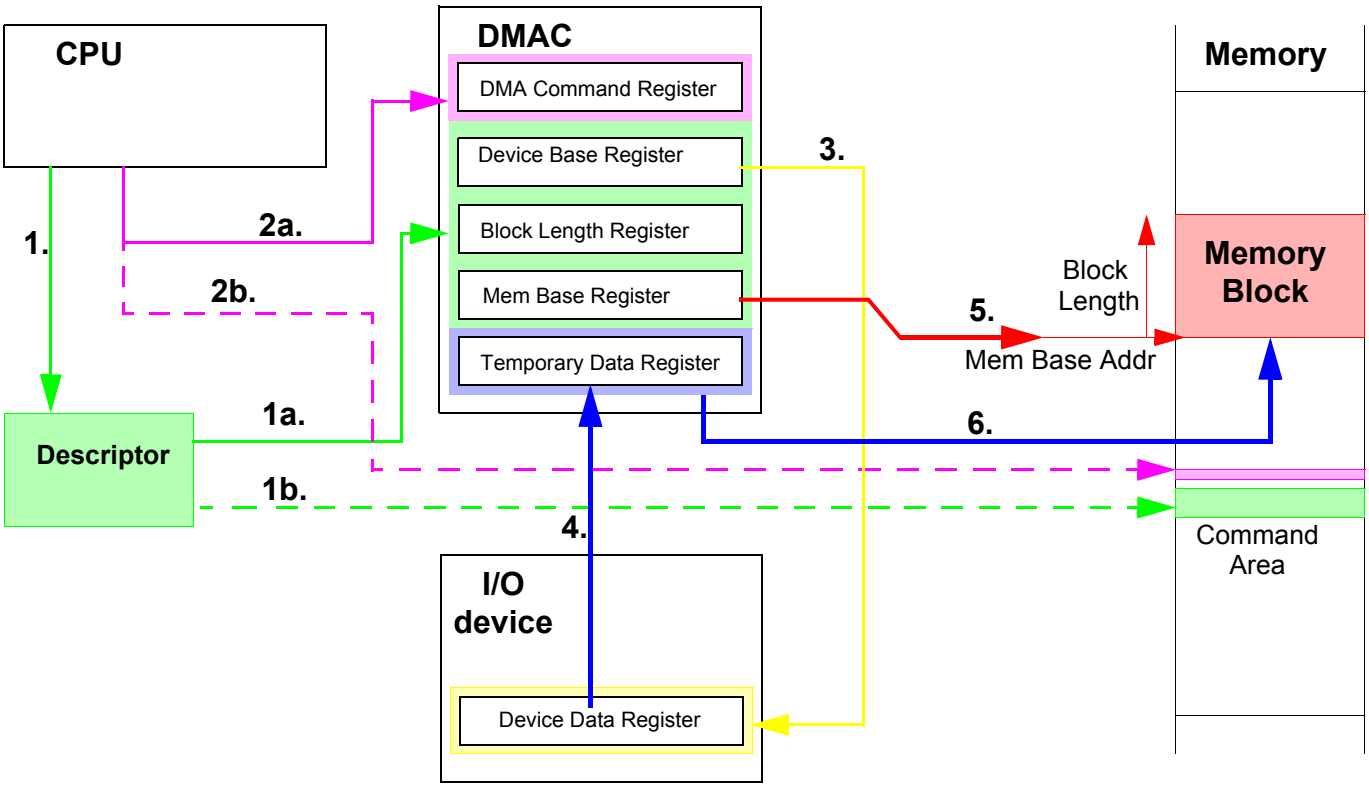
DMA Operations

A lot of different operating modes exist for DMACs. The simplest one ist the single block transfer copying a block of data from a device to memory. For the more complex operations please refer to the literature [Mot81]. Here, only a short list of operating modes is given:

- single block transfer
- chained block transfers
- linked block transfers
- fly-by transfers

All these operations normally access the block of data in a linear sequence. Nevertheless, there are more usefull access functions possible, as there are:

constant stride, constant stride with offset, incremental stride, ...



Execution of a DMA-operation (single block transfer)

The CPU prepares the DMA-operation by the construction of a descriptor (1), containing all necessary information for the DMAC to independently perform the DMA-operation (off-load engine for data transfer). It initalizes the operation by writing a command to a register in the DMAC (2a) or to a special assigned memory area (command area), where the DMAC can poll for the command and/or the descriptor (2b). Then the DMAC addresses the device data register (3) and read the data into a temporary data register (4). In another bus transfer cycle, it addresses the memory block (5) and writes the data from the temporary data register to the memory block (6).

Direct Memory Access (DMA)

DMA Operations

The DMAC increments the memory block address and continue with this loop until the block length is reached. The completion of the DMAoperation is signaled to the processor by sending an IRQ signal or by setting a memory semaphore variable, which can be tested by the CPU.

multiple channels

physical addressing, address translation

snooping for cache coherency

DMA control signals (REQ, ACK) are used to signal the availability of values in the I/O-device for transportation.

DMAC is using bus bandwidth which may slow down processor execution by bus conflicts (solution for high performance systems: use xbar as interconnect!)

[Flik] Mikroprozessortechnik, CISC, RISC Systemaufbau Assembler und C, Flik,Thomas, Springer Verlag, 6.Aufl. 2001.

Completion Signaling of an operation

Completion Signaling

For all communication function it is important to know, when an operation is completed. Signalling this event to the ’process being interested’ in this information is very difficult. The most common way is to throw an interrupt, which stops normal processing of a CPU and activates the interrupt handler. Beside the fact that interrupt processing has speed up significantly in the last years, it need to save the CPU state and in the newest processors the register file is larger than ever.

Design decisions:

- IRQ | polling at device reigster | replication/mirroring in main memory | notification queue | thread scheduling
- Application Processor-Communication Processor model, active messages,

NICs like Infiniband use the concept of a notification queue. For every communication instruction a corresponding entry in the completion notification queue is written, when the operation has finished. This can be testeted by the user process owning the queue.

Direct Memory Access (DMA)