Direct Memory Access (DMA)

- There is often a need to transfer a large number of data between memory and peripherals such as disk drives.
- Using the CPU to transfer such data is too slow. DMA is a mechanism to bypass the CPU and provide direct connection between peripherals and memory. Thus DMA method is more efficient than normal data transfer.
- During the transfer process, DMA controller will take control of the system busses and transfer data between RAM and peripheral devices.

DMA Protocol (I)

- DMA operation is initiated by
  - Hardware, peripheral I/O sends a request to DMA controller (DMAC) by pulling DREQ (DMA request) high
  - Software, CPU write a request command to DMA/C.
- The DMAC will put a high on its HRQ (Hold request), signaling the CPU through its HOLD pin that it needs to use the buses.
- The CPU will finish the present bus cycle and respond to the DMA request by putting high on its HDLA (hold ack.). HOLD must remain active high as long as DMA is performing its task.
- DMAC will activate DACK (DMA ack.) which tells the peripheral device that it will start to transfer the data.
DMA Protocol (II)

- DMAC starts and ends DMA bus cycle.
- After the DMAC has finished its job it will deactivate HRQ, signaling the CPU that it can regain control over its buses.
- While the DMA is using the buses to transfer data, the DMA is idle. By the same token, when the CPU is using the bus, the DMA is idle.

DMA Usage of System Bus

8237 DMA System Interface

8237 Internal Architecture
Mode Register

- Bit Number
- 00: Channel 0 select
- 01: Channel 1 select
- 10: Channel 2 select
- 11: Channel 3 select
- 02: Verify transfer
- 01: Write transfer
- 10: Read transfer
- 11: Illegal
- XX: If bits 6 and 7 = 11

- 0: Autoinitialization disable
- 1: Autoinitialization enable

- 0: Address increment select
- 1: Address decrement select

- 00: Demand mode select
- 01: Single mode select
- 10: Block mode select
- 11: Cascade mode select

Command Register

- Bit Number
- 0: Memory-to-memory disable
- 1: Memory-to-memory enable
- 0: Channel address held disable
- 1: Channel 0 address held enable
- X: If bit 9 = 0

- 0: Controller enable
- 1: Controller disable

- 0: Normal timing
- 1: Compressed timing
- X: If bit 3 = 1

- 0: Fixed priority
- 1: Rotating priority

- 0: Late write selection
- 1: Extend with select
- X: If bit 3 = 1

- 0: DREQ sense active high
- 1: DREQ sense active low

- 0: DACK sense active low
- 1: DACK sense active high

Request Register

- Bit Number
- 00: Select channel 0
- 01: Select channel 1
- 10: Select channel 2
- 11: Select channel 3

- 0: Reset request bit
- 1: Set request bit

Mask Register

- Bit Number
- 00: Select channel 0 mask bit
- 01: Select channel 1 mask bit
- 10: Select channel 2 mask bit
- 11: Select channel 3 mask bit

- 0: Clear mask bit
- 1: Set mask bit
### Definition of Register Code

<table>
<thead>
<tr>
<th>Register</th>
<th>Operation</th>
<th>Signals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>Write</td>
<td>0 1 0 1 0 0 0</td>
</tr>
<tr>
<td>Mode</td>
<td>Write</td>
<td>0 1 0 1 0 1 1</td>
</tr>
<tr>
<td>Request</td>
<td>Write</td>
<td>0 1 0 1 0 0 1</td>
</tr>
<tr>
<td>Mask</td>
<td>Set/Reset</td>
<td>0 1 0 1 0 1 0</td>
</tr>
<tr>
<td>Mask</td>
<td>Write</td>
<td>0 1 0 1 1 1 1</td>
</tr>
<tr>
<td>Temporary</td>
<td>Read</td>
<td>0 0 1 1 1 0 1</td>
</tr>
<tr>
<td>Status</td>
<td>Read</td>
<td>0 0 1 1 0 0 0</td>
</tr>
</tbody>
</table>

### Status Register

![Status Register Diagram]

- Bit 7: Channel 0 has reached TC
- Bit 6: Channel 1 has reached TC
- Bit 5: Channel 2 has reached TC
- Bit 4: Channel 3 has reached TC
- Bit 3: Channel 0 request
- Bit 2: Channel 1 request
- Bit 1: Channel 2 request
- Bit 0: Channel 3 request

### DMA Timing

![DMA Timing Diagram]