8237-DMA

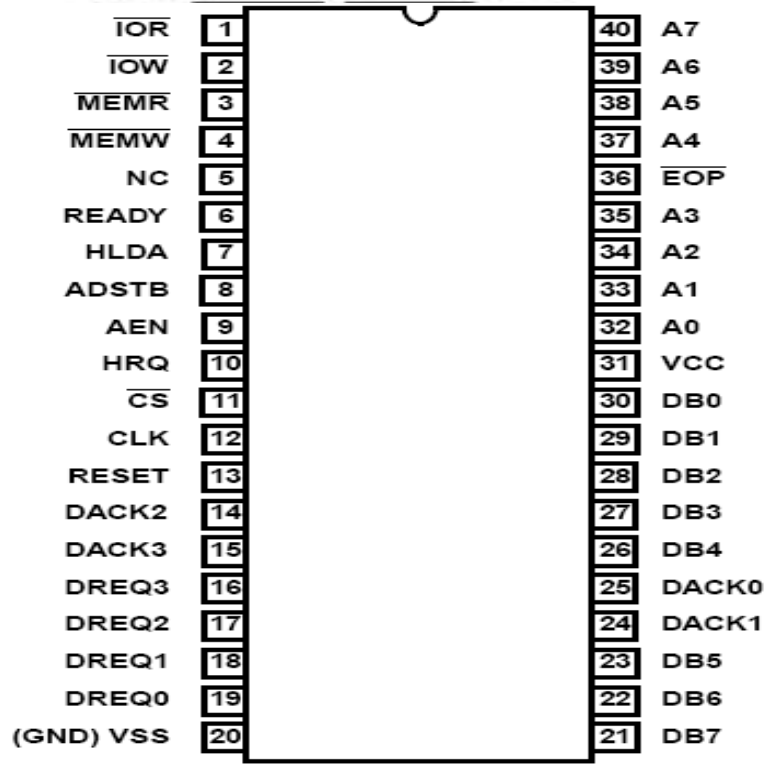
<INTRODUCTION>

- Direct Memory Access (DMA) is a method of allowing data to be moved from one location to another in a computer without intervention from the central processor (CPU).
- It is also a fast way of transferring data within (and sometimes between) computer.
- The DMA I/O technique provides direct access to the memory while the microprocessor is temporarily disabled.
- The DMA controller temporarily borrows the address bus, data bus and control bus from the microprocessor and transfers the data directly from the external devices to a series of memory locations (and vice versa).

BASIC DMA OPERATION

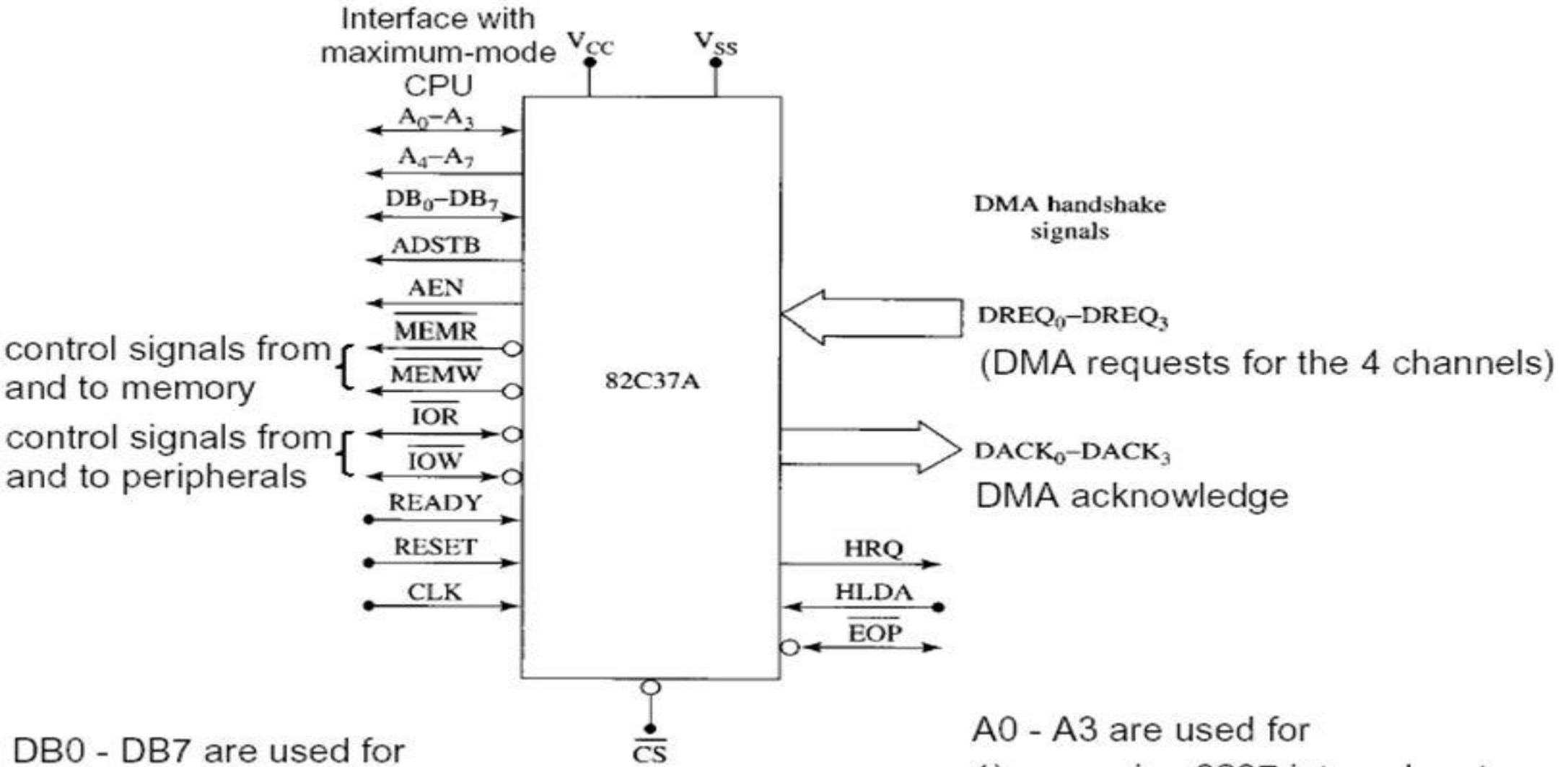
- Two control signals are used to request and acknowledge a direct memory access (DMA) transfer in the microprocessor-based system.
  - The HOLD signal as an input(to the processor) is used to request a DMA action.
  - The HLDA signal as an output that acknowledges the DMA action.
- When the processor recognizes the hold, it stops its execution and enters hold cycles.
- HOLD input has higher priority than INTR or NMI.
- The only microprocessor pin that has a higher priority than a HOLD is the RESET pin.
- HLDA becomes active to indicate that the processor has placed its buses at high-impedance state.

8237 pins



- CLK: System clock
- CS′: Chip select (decoder output)
- RESET: Clears registers, sets mask register
- READY: 0 for inserting wait states
- HLDA: Signals that the µp has relinquished buses
- DREQ3 – DREQ0: DMA request input for each channel
- DB7-DB0: Data bus pins
- IOR′: Bidirectional pin used during programming and during a DMA write cycle
- IOW′: Bidirectional pin used during programming and during a DMA read cycle
- EOP′: End of process is a bidirectional signal used as input to terminate a DMA process or as output to signal the end of the DMA transfer
- A3-A0: Address pins for selecting internal registers
- A7-A4: Outputs that provide part of the DMA transfer address
- HRQ: DMA request output
- DACK3-DACK0: DMA acknowledge for each channel.
- AEN: Address enable signal
- ADSTB: Address strobe
- MEMR′: Memory read output used in DMA read cycle
- MEMW′: Memory write output used in DMA write cycle

8237 DMA controller



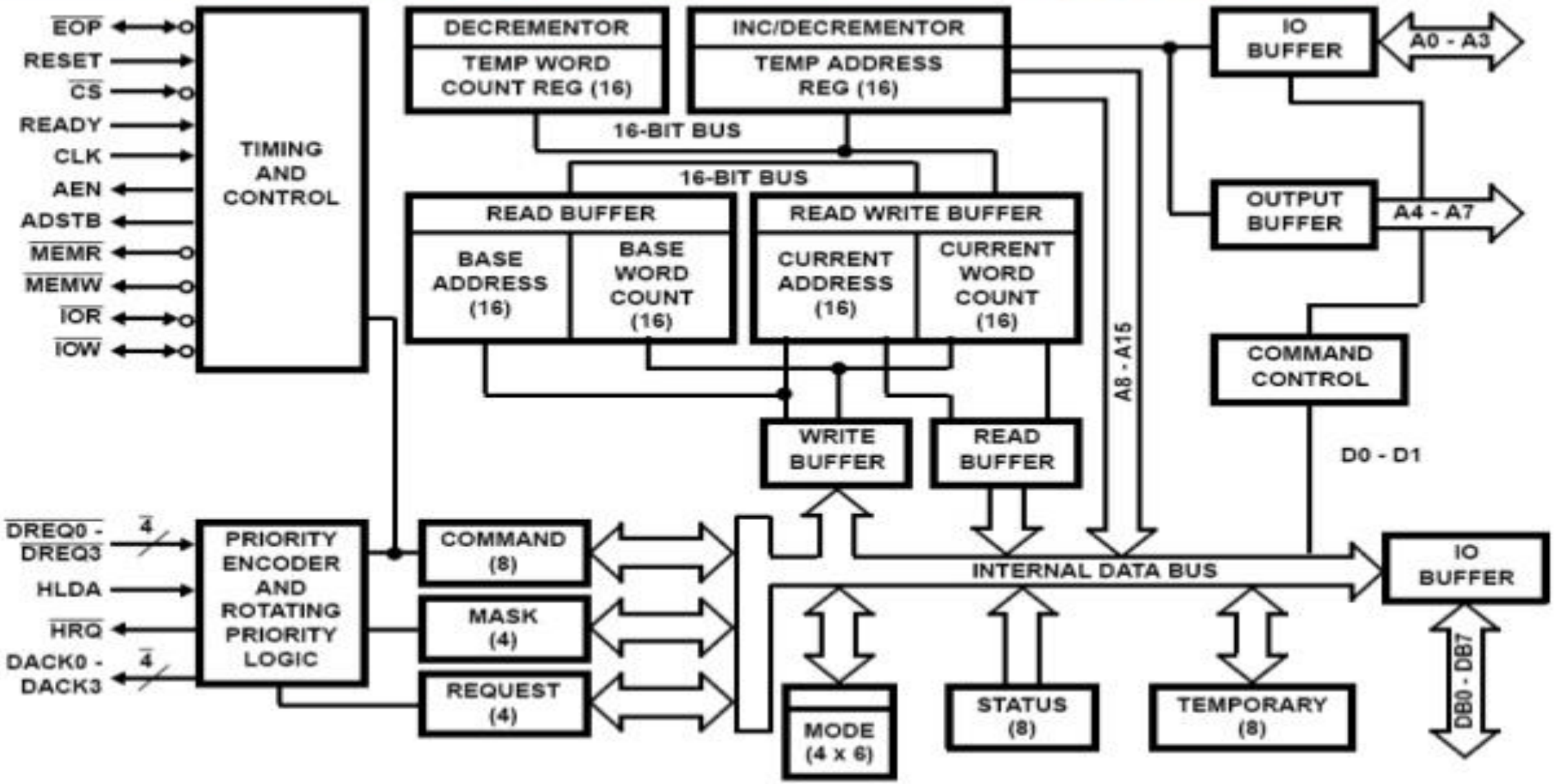
DB0 - DB7 are used for

1) transfer of data
2) 8237 programming

A0 - A3 are used for

1) accessing 8237 internal ports
2) carrying memory address in DMA read and write operations

8237 block diagram



BLOCK DIAGRAM

- CAR (Current Address Register): holds the 16-bit memory address used for the DMA transfer (one for each channel), either incremented or decremented during the operation
- CWCR (Current Word Count Register): Programs a channel for the number of bytes (up to 64K) transferred during a DMA operation
- BA (Base Address) and WC (Word Count): Used when auto-initialization is selected for a channel, to reload the CAR and CWCR when DMA is complete.
- CR (Command Register): Programs the operation of the controller

BLOCK DIAGRAM

MR

- The mode register programs the mode of operation for a channel.
- Each channel has its own mode register as selected by bit positions 1 and 0.

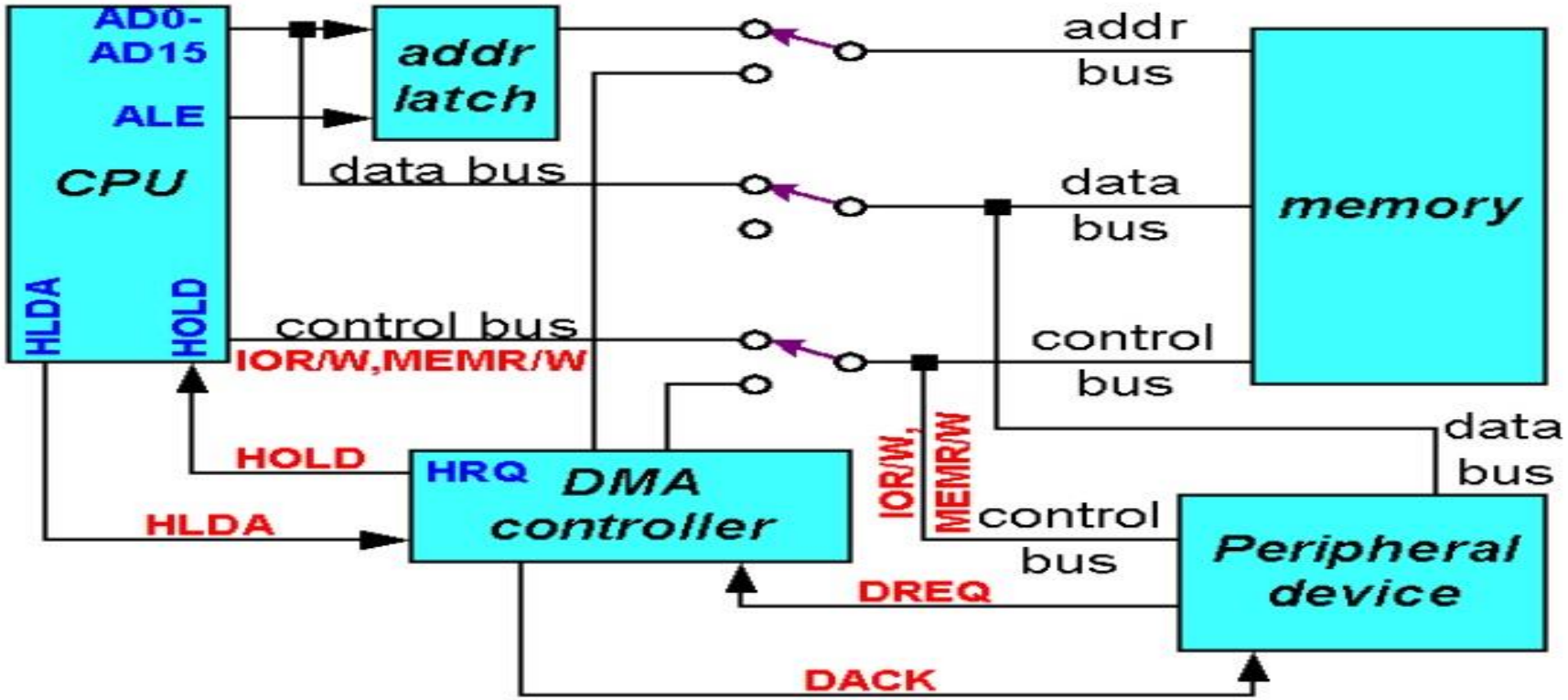
- RR (Request Register): Used to request DMA transfer via software (memory-to-memory transfers)

MSR

- The mask register clears or sets all of the masks with one command instead of individual channels, as with the MRSR.

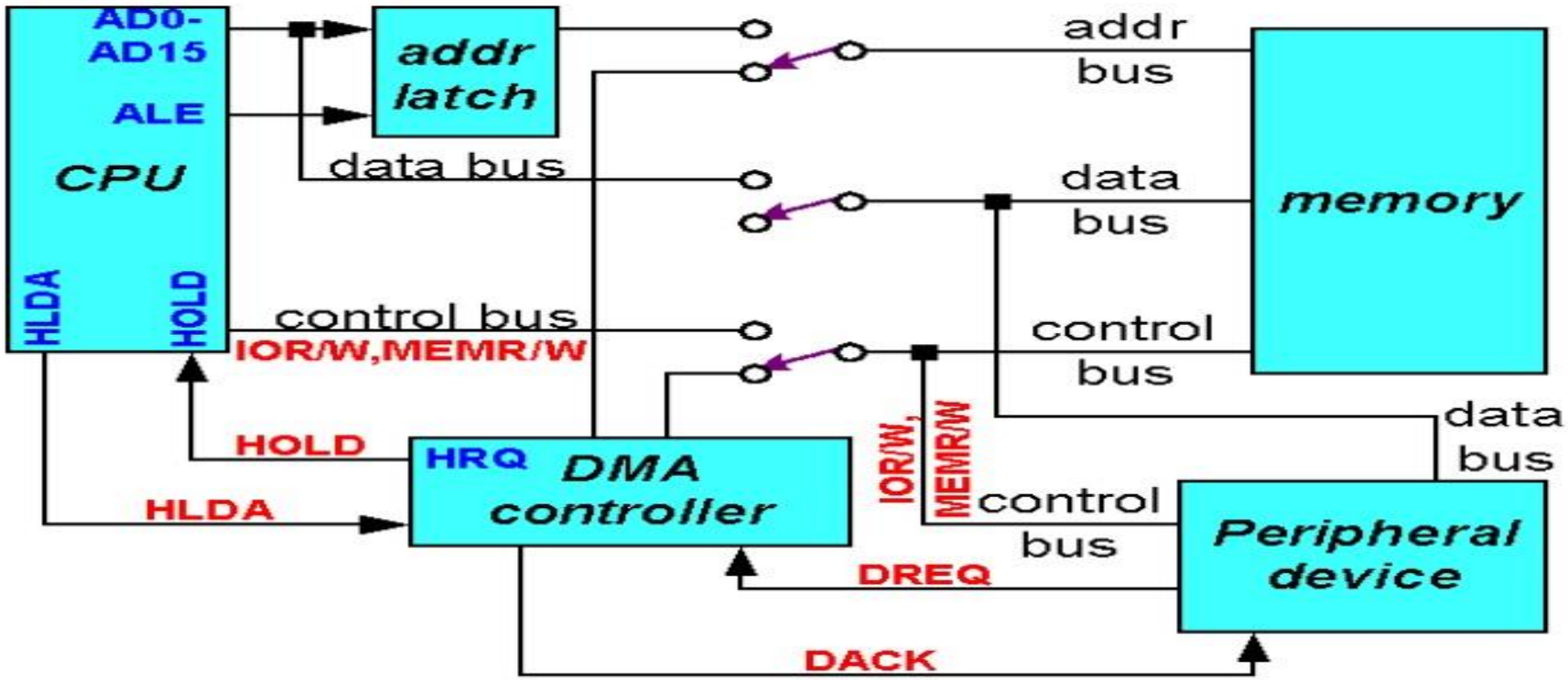
INTERFACING CIRCUIT WITH 8086 (CPU)

CPU having the control over the bus:



INTERFACING CIRCUIT WITH 8086 (CPU)

When DMA operates:



WORKING OF DMA CONTROLLER

- **The I/O device asserts the appropriate DRQ signal for the channel.**
- **The DMA controller will enable appropriate channel, and ask the CPU to release the bus so that the DMA may use the bus. The DMA requests the bus by asserting the HOLD signal which goes to the CPU.**
- **The CPU detects the HOLD signal, and will complete executing the current instruction. Now all of the signals normally generated by the CPU are placed in a tri-stated condition (neither high or low) and then the CPU asserts the HLDA signal which tells the DMA controller that it is now in charge of the bus.**
- **The CPU may have to wait (hold cycles).**
- **DMA activates its -MEMR, -MEMW, -IOR, -IOW output signals, and the address outputs from the DMA are set to the target address, which will be used to direct the byte that is about to transferred to a specific memory location.**

WORKING OF DMA CONTROLLER

- The DMA will then let the device that requested the DMA transfer know that the transfer is commencing by asserting the -DACK signal.
- The peripheral places the byte to be transferred on the bus Data lines.
- Once the data has been transferred, The DMA will de-assert the -DACK2 signal, so that the FDC knows it must stop placing data on the bus.
- The DMA will now check to see if any of the other DMA channels have any work to do. If none of the channels have their DRQ lines asserted, the DMA controller has completed its work and will now tri-state the -MEMR, -MEMW, -IOR, -IOW and address signals.
- Finally, the DMA will de-assert the HOLD signal. The CPU sees this, and de-asserts the HOLDA signal. Now the CPU resumes control of the buses and address lines, and it resumes executing instructions and accessing main memory and the peripherals.

The END