DIRECT MEMORY ADDRESS AND 8237 DMA CONTROLLER
LECTURE 4

Basic DMA concept

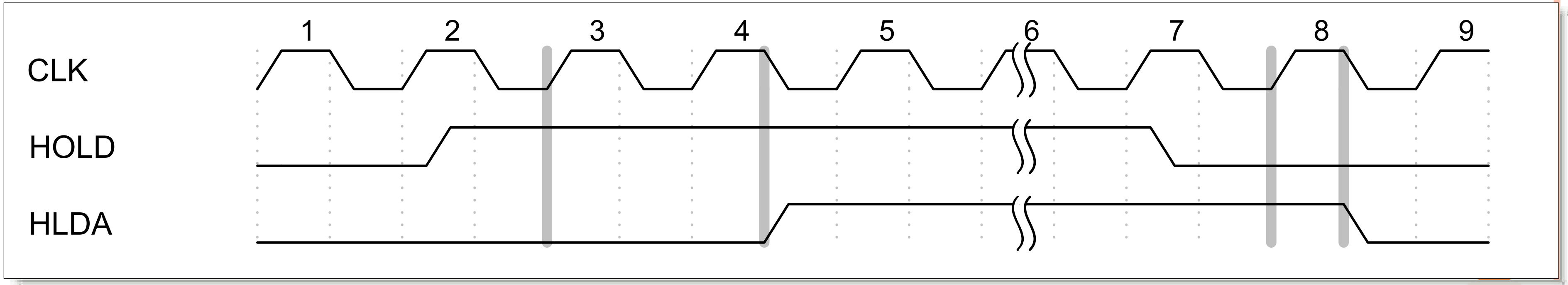
- **Direct memory access** (**DMA**) is a feature of modern computer systems that allows certain hardware subsystems to read/write data to/from memory without microprocessor intervention, allowing the processor to do other work.
- Used in disk controllers, video/sound cards etc, or between memory locations.
- Typically, the CPU initiates DMA transfer, does other operations while the transfer is in progress, and receives an interrupt from the DMA controller once the operation is complete.
- Can create cache coherency problems (the data in the cache may be different from the data in the external memory after DMA)

BASIC DMA TERMINOLOGY

- *DMA channel*: system pathway used by a device to transfer information directly to and from memory. There are usually 8 in a computer system
- *DMA controller*: dedicated hardware used for controlling the DMA operation
- *Single-cycle mode:* DMA data transfer is done one byte at a time
- *Burst-mode:* DMA transfer is finished when all data has been moved

DMA PINS AND TIMING

- x86 Interrupt Pins
  - HOLD: DMA request.
    - Sampled in the middle of any clocking cycle
  - HLDA: DMA acknowledge signal.
    - The address, data and control buses are set to high-Z, so the I/O devices can control the system bus



DMA ON THE 8086 MICROPROCESSOR

- 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.
- 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.

EXAMPLE

- Assuming that a DMA initialization has an overhead of 10 cycles, while a CPU transfer to/from memory requires 4 cycles (no wait states required), compare a DMA and a CPU transfer from one memory location to another of
  - One byte of data
  - A block of 1Kbytes in burst mode
  - A block of 64Kbytes in burst mode

THE 8237 DMA CONTROLLER

- Supplies memory and I/O with control signals and addresses during DMA transfer
- 4-channels (expandable)
  - 0: DRAM refresh
  - 1: Free
  - 2: Floppy disk controller
  - 3: Free
- 1.6MByte/sec transfer rate
- 64 KByte section of memory address capability with single programming
- “fly-by” controller (data does not pass through the DMA-only memory to I/O transfer capability)
- Initialization involves writing into each channel:
  - i) The address of the first byte of the block of data that must be transferred (called the base address).
  - ii) The number of bytes to be transferred (called the word count).

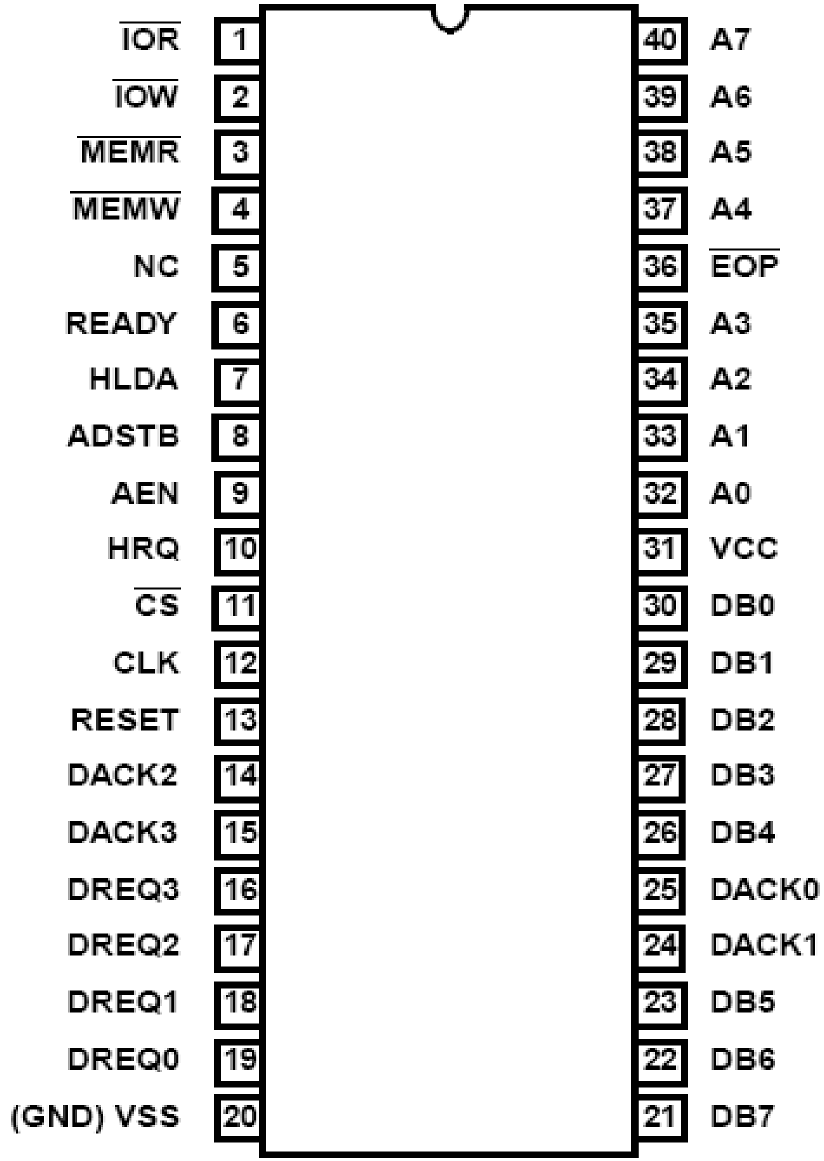
8237 PINS

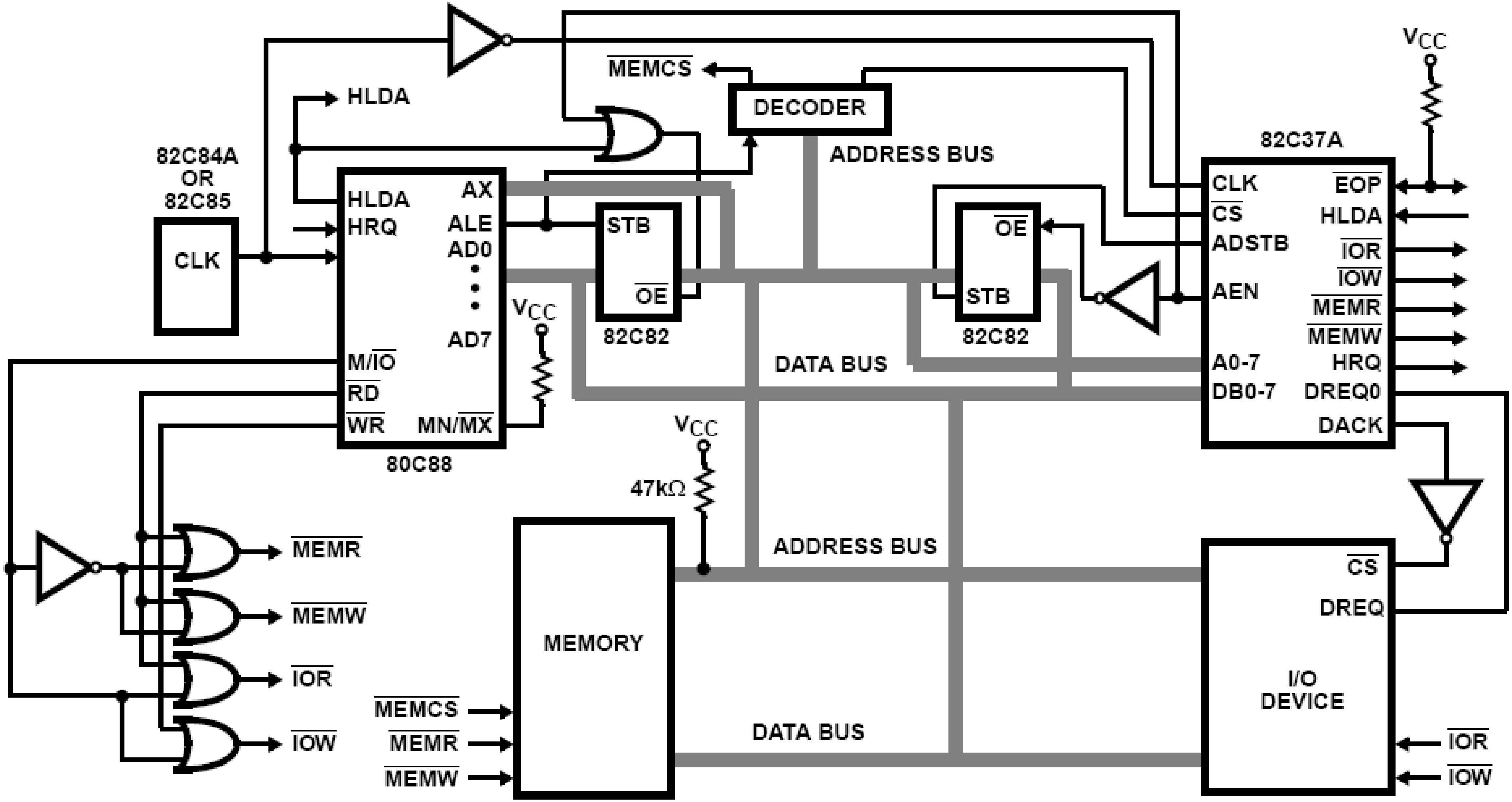
- CLK: System clock
- CS´: Chip select (decoder output)
- RESET: Clears registers, sets mask register
- READY: 0 for inserting wait states
- HLDA: Signals that the µp has relinquished buses
- DREQ3 – DREQ0: DMA request input for each channel
- DB7-DB0: Data bus pins
- IOR´: Bidirectional pin used during 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 PIN DIAGRAM




HLDA
MEMCS
DECODER
ADDRESS BUS
82C84A
OR
82C85
CLK
HLDA
HRQ
AX
ALE
AD0
AD7
M/IO
RD
WR
MN/MX
80C88
VCC
STB
OE
82C82
DATA BUS
OE
STB
82C82
82C37A
VCC
CLK
CS
ADSTB
AEN
A0-7
DB0-7
EOP
HLDA
IOR
IOW
MEMR
MEMW
HRQ
DREQ0
DACK
VCC
47kΩ
ADDRESS BUS
MEMORY
DATA BUS
MEMR
MEMW
IOR
IOW
MEMCS
MEMR
MEMW
CS
DREQ
I/O
DEVICE
IOR
IOW

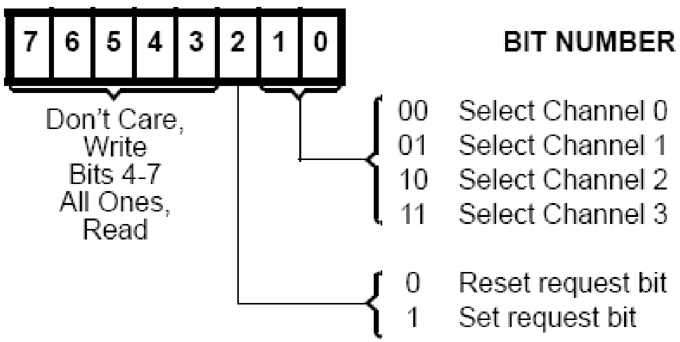
8237 REGISTERS

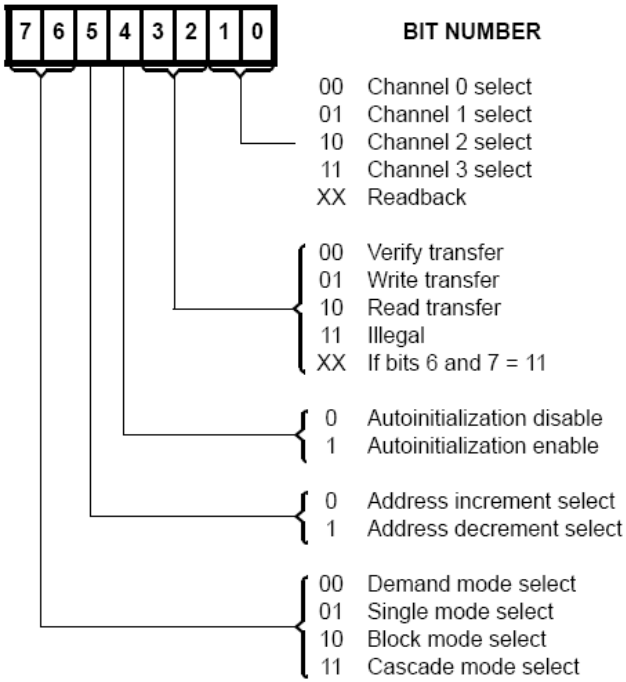
- CAR (Current Address Register): holds the 16-bit memory address used for the DMA transfer (one for each channel), either incremented or decremented during the operation
- CWCR (Current Word Count Register): Programs a channel for the number of bytes (up to 64K) transferred during a DMA operation
- 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

- MR (Mode Register):

8237 REGISTERS

- Programs the mode of operation for a channel (one for each channel).

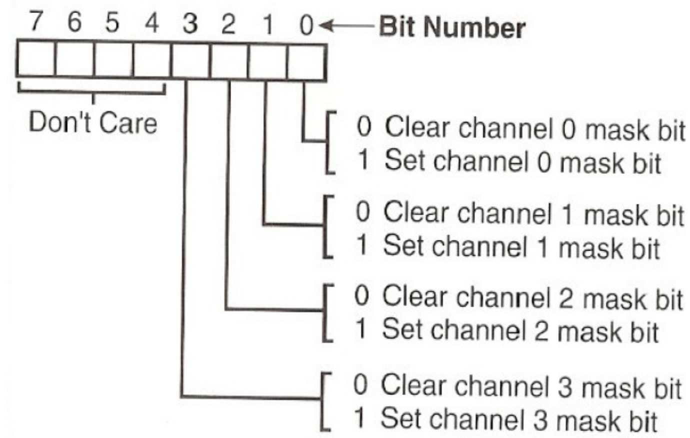




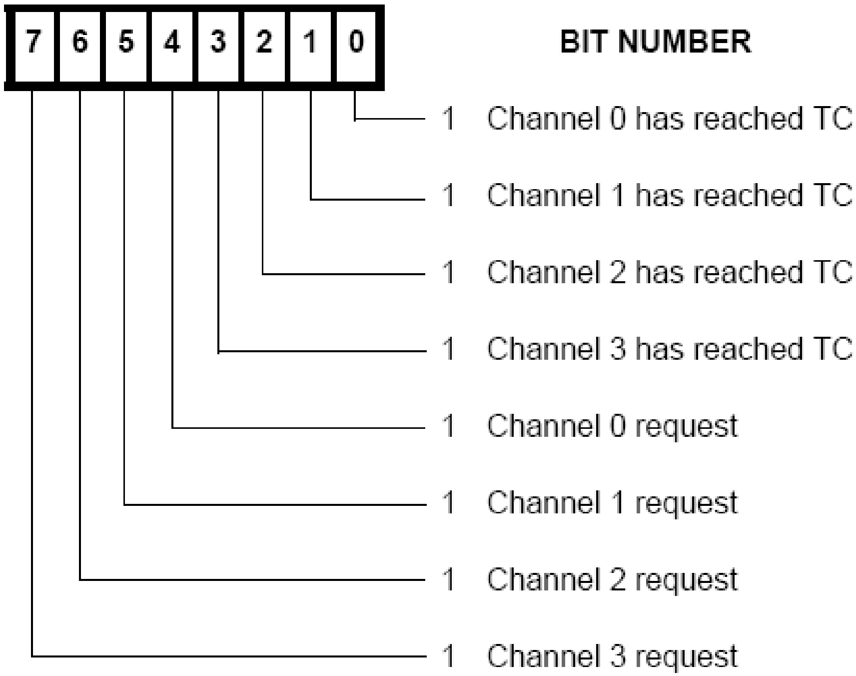
- RR (Request Register): Used to request DMA transfer via software (memory-to-memory transfers)

8237 REGISTERS

- MR (Mask Register):



- SR (Status Register): Shows the status of each DMA channel



8237 Software Commands

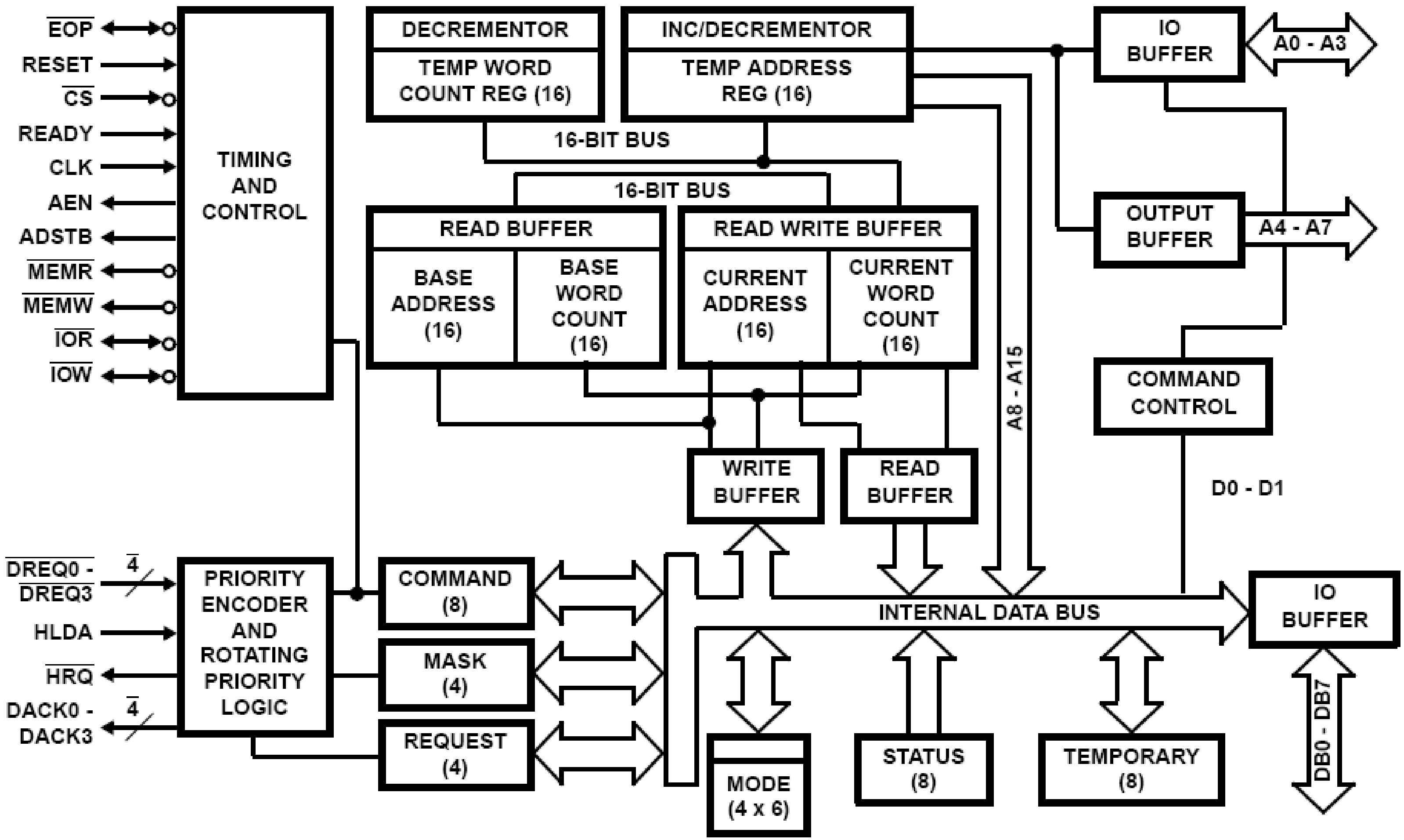
OPERATION	A3	A2	A1	A0	$\overline{IOR}$	$\overline{IOW}$
Read Status Register	1	0	0	0	0	1
Write Command Register	1	0	0	0	1	0
Read Request Register	1	0	0	1	0	1
Write Request Register	1	0	0	1	1	0
Read Command Register	1	0	1	0	0	1
Write Single Mask Bit	1	0	1	0	1	0
Read Mode Register	1	0	1	1	0	1
Write Mode Register	1	0	1	1	1	0
Set First/Last F/F	1	1	0	0	0	1
Clear First/Last F/F	1	1	0	0	1	0
Read Temporary Register	1	1	0	1	0	1
Master Clear	1	1	0	1	1	0
Clear Mode Reg. Counter	1	1	1	0	0	1
Clear Mask Register	1	1	1	0	1	0
Read All Mask Bits	1	1	1	1	0	1
Write All Mask Bits	1	1	1	1	1	0

8237 Software commands

- **Clear First/Last Flip-Flop -** This command is executed prior to writing or reading new address or word count information to the 82C37. This command initializes the flipflop to a known state (low byte first) so that subsequent accesses to register contents by the microprocessor will address upper and lower bytes in the correct sequence.
- **Set First/Last Flip-Flop -** This command will set the flip-flop to select the high byte first on read and write operations to address and word count registers.
- **Master Clear -** This software instruction has the same effect as the hardware Reset. The Command, Status, Request, and Temporary registers, and Internal First/Last Flip-Flop and mode register counter are cleared and the Mask register is set. The 82C37A will enter the idle cycle.
- **Clear Mask Register -** This command clears the mask bits of all four channels, enabling them to accept DMA requests.
- **Clear Mode Register Counter -** Since only one address location is available for reading the Mode registers, an internal two-bit counter has been included to select Mode registers during read operation. To read the Mode registers, first execute the Clear Mode Register Counter command, then do consecutive reads until the desired channel is read. Read order is channel 0 first, channel 3 last. The lower two bits on all Mode registers will read as ones.

Channel	Register	Operation	Signals							Internal Flip-Flop	Data Bus DB0-DB7
			$\overline{CS}$	$\overline{IOR}$	$\overline{IOW}$	A3	A2	A1	A0		
0	Base and Current Address	Write	0	1	0	0	0	0	0	0	A0-A7
			0	1	0	0	0	0	0	1	A8-A15
	Current Address	Read	0	0	1	0	0	0	0	0	A0-A7
			0	0	1	0	0	0	0	1	A8-A15
	Base and Current Word Count	Write	0	1	0	0	0	0	1	0	W0-W7
			0	1	0	0	0	0	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	0	0	1	0	W0-W7
			0	0	1	0	0	0	1	1	W8-W15
1	Base and Current Address	Write	0	1	0	0	0	1	0	0	A0-A7
			0	1	0	0	0	1	0	1	A8-A15
	Current Address	Read	0	0	1	0	0	1	0	0	A0-A7
			0	0	1	0	0	1	0	1	A8-A15
	Base and Current Word Count	Write	0	1	0	0	0	1	1	0	W0-W7
			0	1	0	0	0	1	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	0	1	1	0	W0-W7
			0	0	1	0	0	1	1	1	W8-W15
2	Base and Current Address	Write	0	1	0	0	1	0	0	0	A0-A7
			0	1	0	0	1	0	0	1	A8-A15
	Current Address	Read	0	0	1	0	1	0	0	0	A0-A7
			0	0	1	0	1	0	0	1	A8-A15
	Base and Current Word Count	Write	0	1	0	0	1	0	1	0	W0-W7
			0	1	0	0	1	0	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	1	0	1	0	W0-W7
			0	0	1	0	1	0	1	1	W8-W15
3	Base and Current Address	Write	0	1	0	0	1	1	0	0	A0-A7
			0	1	0	0	1	1	0	1	A8-A15
	Current Address	Read	0	0	1	0	1	1	0	0	A0-A7
			0	0	1	0	1	1	0	1	A8-A15
	Base and Current Word Count	Write	0	1	0	0	1	1	1	0	W0-W7
			0	1	0	0	1	1	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	1	1	1	0	W0-W7
			0	0	1	0	1	1	1	1	W8-W15

8237 BLOCK DIAGRAM



INITIATING A DMA TRANSACTION

- Save the current interrupt status and disable interrupts by executing the CLI instruction
- Disable the channel that will be used for the transaction
- Reset the flip-flop by writing a value of 0X to the register
- Set the Mode Register
- Set the Page Register
- Set the Offset Register
- Set the Block Size Register
- Enable the channel that will be used for the transaction
- Restore the interrupt status