The Real Time Clock (RTC)

Calling the clock real-time is somewhat of a misnomer because it only reflects the time setting it has been given. The RTC is the other half of the chip that has the CMOS memory and can be thought of as a set of counters.

The first one counts from 0 to 9 and then tells the next counter (the 10's place counter) to count up once. The first counter then starts counting again and counts from 0 to 9 and again telling the next higher counter (the 10's counter) to set its counter up one more (now at 2), and so on. Now the 10's place counter (which is counting 10's of seconds) only counts from 0 to 5 and then it tells the minute counter to do its increment, which will then start its journey from 0 to 9 and so on.

Of course the next counter after the minute counter is the minute 10's place counter, which also counts from 0 to 5 and then tells the hours counter to count once etc, etc.

The process goes on through the hour 10's place as it counts from 0 to 2, the day counter-which goes from 1 to 30, or 31, or sometimes 28 or maybe 29 depending on what the rules for which month has how many days and which years have 29 day Februarys. The month counter proceeds to go from 1 to 12, and then of course the year counter starts its journey with the good old 0 to 9, and finally we have the year 10's counter again with values going from 0 to 9.

DMA

DMA stands for **D**irect **M**emory **A**ccess

- Uses same Address/Data lines on ISA bus
- Controls the ISA bus instead of the processor ("bus master")
- Floppy Drive (DRQ2)
- Hard Drive (varies)
- Sound Card (varies)

- DMA vs. Progammed I/O
- Requires less processor load
- Can continually transfer the same block (loop indefinitely)
- Devices which are demand-based can be serviced more efficiently

DMA characteristics:

8-bit or 16-bit transfers

HOW DMA WORKS to transfer data from memory to a peripheral:

1. Peripheral Device requests DMA service by pulling DREQ line high

2. DMA controller requests CPU go on hold by pulling CPU's HRQ line high

3. CPU finishes current bus cycle, then acknowledges
   with HLDA The CPU relinquishes ALL bus control
   now (3-state)
4. DMA controller activates DACK line to Acknowledge DMA to the peripheral

5. DMA begin transferring data to the peripheral

DMA Controller

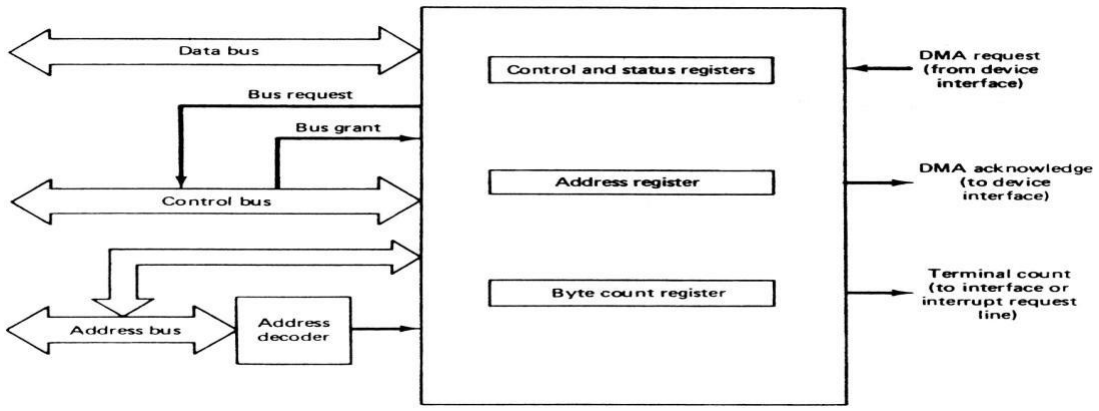

Figure 4.17: General Organization of a DMA Controller

Intel's 8237 DMA controller:

Steps involved in transferring a block of data from I/O devices (e.g. a disk) to memory:

1. the CPU sends a signal to initiate a disk transfer through the I/O interface
2. the CPU sends the starting address of the block
3. the disk driver reads the starting address, and reads a block of data and puts it in its own buffer
4. the disk driver sends a interrupt signal to the CUP
5. the CPU reads the datum into its registers (accumulator)
6. the CPU checks if there is more to transfer, if yes, the CPU signals the disk driver to do so meanwhile
7. the CPU transfer the datum from register to memory and increments its pointer to memory
8. The DMA controller takes care of the last few steps (from signaling disk to transfer) .A

Intel's 8237 DMA controller:

- The 8237 has 4 independent I/O channels

- It has 27 registers, 7 of which are system-wide registers and 5 for each channels.

- out of 5 regs: 4 are 16-bit and 1 is 6-bit. they are

- 6-bit is the mode register

- DMA base address

- DMA current address

- DMA total

- DMA remaining

DMA Hardware (8237 DMAC)

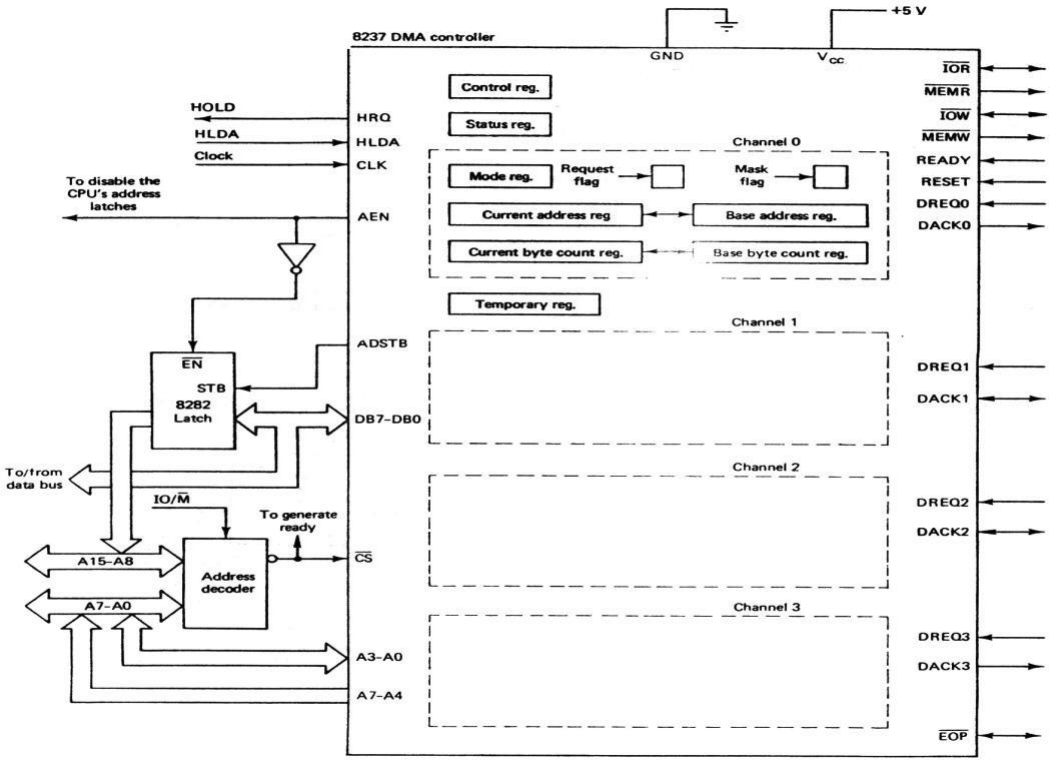

Figure 4.18: Organization of an 8237 and its associated logic

- Processor has HOLD/HOLD Acknowledge lines to interact with 8237
  - DMAC can gain control of ISA bus by asserting HOLD
  - Processor acknowledges with HLDA

- DRQ4 services slave controller

- Priorities are set as fixed

  - DRQ0 highest,DRQ7 lowest

  - Set at POST

  - Can be reprogrammed for rotating priority

- ISA address/data/control lines are also connected (not shown)

- Can access control registers through ports

- Each channel has a page register associated with it

8237 Modes

Intel 8237 can be set to four different style of transfer:

1. Single - One transfer at a time, allow processor access to the bus between transfers

2. Block - Transfer all data, do not allow processor access to the bus (may cause problems with memory refresh)

3. Demand - Keep transferring as long as target keeps DRQ asserted

4. Cascade - allow a slave controller use of the DMAC (used for DRQ4)

   In addition, the DMA controller can be set to make continuous transfers

   - known as auto-initialized DMA

   - normal DMA is known as "single-cycle"

8237 is clocked at 1/2 of ISA Bus (0.5 *BLCK)

- up to 4.166MHz (8.33 Mhz ISA)
- Maximum transfer rate: 4.166MB/s (16-bit DMA)
- Maximum Programmed I/O transfer rate: 2.77 MB/s

Size of transfer

- Master can only generate word-sized transfers
- Slave can generate byte-sized transfers
- Minimum transfer size: 1 byte
- Maximum transfer size: 64KB (8-bit),128KB (16-bit)