

DMA Controllers

Intel 8237

Trasferimento dati e DMA

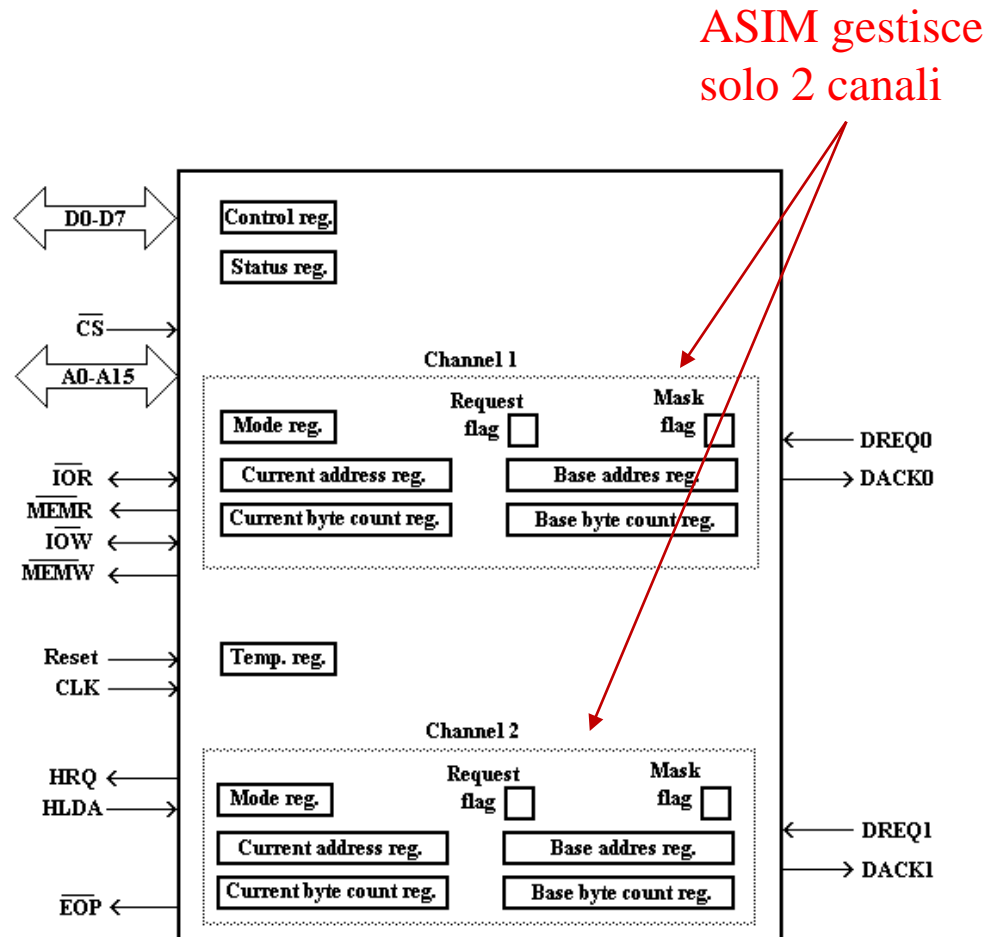
- Le **operazioni di trasferimento dati da e verso i moduli di I/O** gestite mediante interruzioni o polling richiedono l'esecuzione di codici operativi da parte del processore
 - la velocità di trasferimento è limitata dalla velocità con cui il processore testa e serve il device e
 - il processore è impegnato nell'eseguire un certo numero di istruzioni per ogni operazione di I/O
- Per trasferire grandi volumi di dati si ricorre ad un meccanismo più efficiente basato sul **DMA**, in grado di operare parallelamente al processore e gestire, mediante hardware specializzato, le operazioni di trasferimento.
- Un dispositivo DMA controller è capace, opportunamente programmato, di diventare il master del bus e supervisionare un trasferimento di tipo *memoria-memoria*, *device-memoria* o *memoria-device*, senza intervento del processore.
 - Il processore intanto può eseguire altri codici operativi (sempre che non si creino conflitti di accesso sul bus) mentre il controllore DMA si occupa del trasferimento dei dati colloquiando direttamente con la memoria centrale.

Il dispositivo Intel 8237

- Il dispositivo **Intel 8237** dispone di **4 canali** per il collegamento con altrettanti device
- In caso di competizione su due o più canali, il controllore DMA può selezionare il canale da attivare in base ad uno schema prioritario fisso, in cui l'ordine prioritario è legato alla posizione del canale (3 più prioritario e 0 meno prioritario) o dinamico a priorità ruotante (round-robin) in cui l'ultimo canale che ha ricevuto il servizio passa in posizione meno prioritaria.
- Durante il trasferimento dati, il DMA può accedere al canale dati secondo 4 modalità:
 - **Single**: trasferisce una parola alla volta e dopo ogni trasferimento rilascia il bus al processore per almeno un ciclo di bus
 - **Block**: trasferisce un intero blocco una volta ottenuto il controllo del bus. Alla fine del trasferimento del blocco il controllore DMA invia un interrupt alla CPU al fine di avviare le operazioni di chiusura dell'operazione.
 - **Demand**: simile al modo **block**, con la differenza che il trasferimento del blocco continua fin quando la linea di richiesta è attiva ma, quando il trasferimento viene sospeso e poi ripreso, esso inizia dal punto in cui era stato sospeso;
 - **Cascade**: permette di realizzare, collegando più controllori 8237 in cascata, sistemi DMA con più di 4 canali.

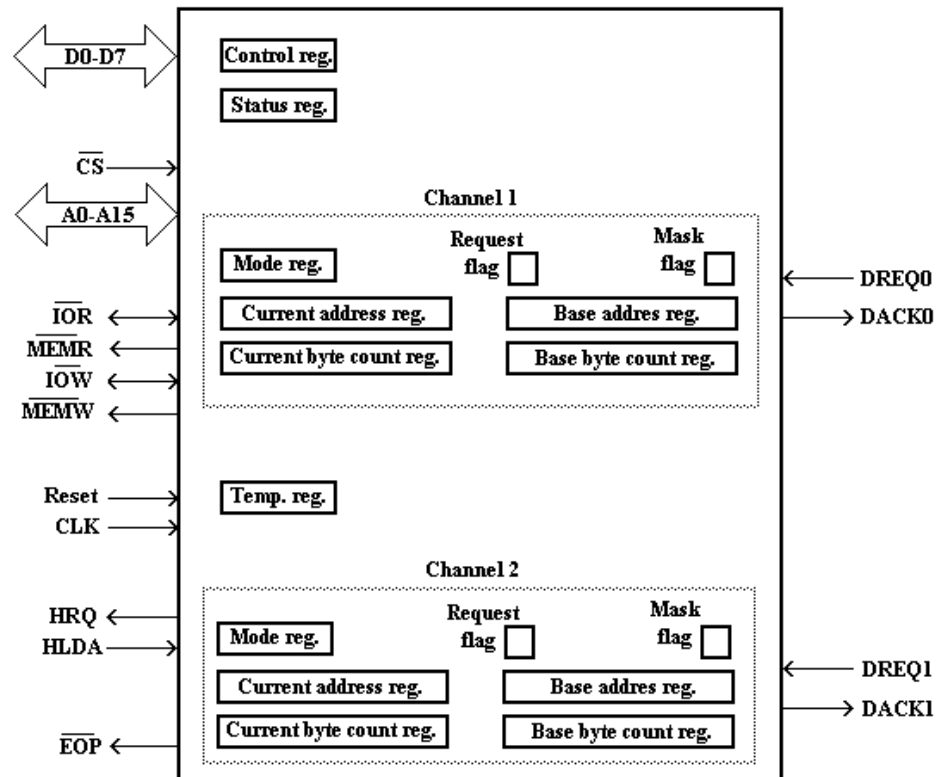
Architettura e Segnali

- **D0-D7** sono 8 linee che vanno connesse al bus dati per il trasferimento dei dati da e verso il dispositivo.
- **CS** è utilizzata per la selezione del dispositivo.
- la selezione dei 16 registri interni è effettuata mediante le linee **A0-A3** del bus indirizzi e dai segnali di lettura-scrittura sul componente: **IOW** e **IOR**.
- Il controllore, per eseguire un trasferimento, deve attivare i segnali necessari per effettuare le operazioni di lettura e scrittura in memoria (**MEMR** e **MEMW**) e sulle interfacce periferiche (**IOR** e **IOW**).



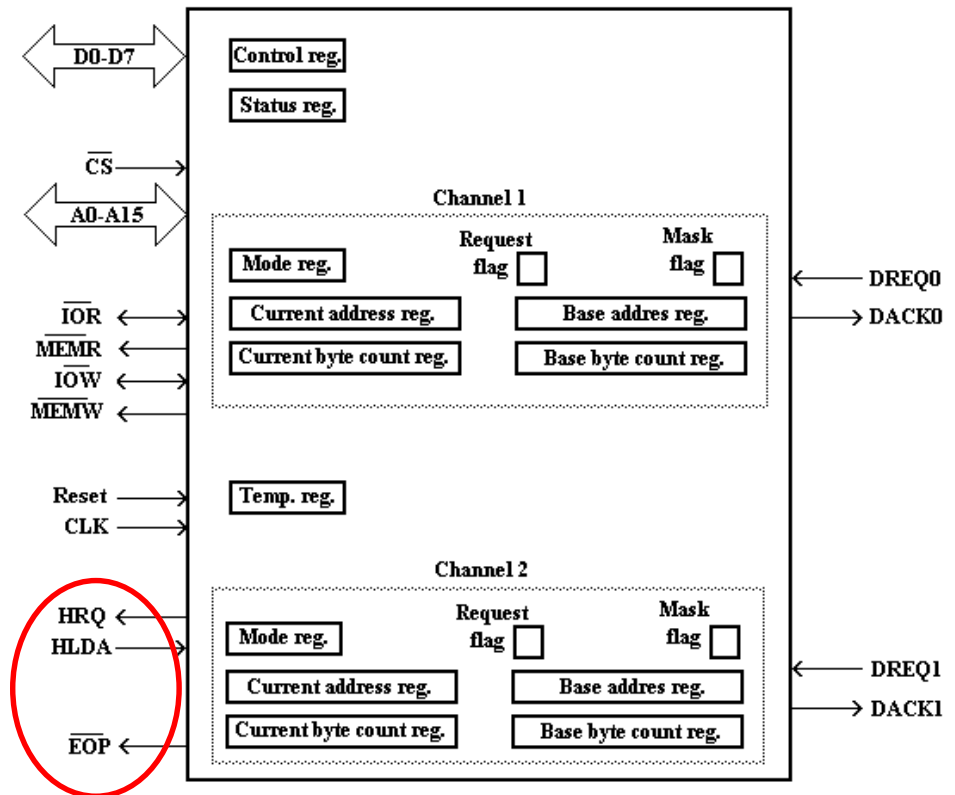
Architettura e Segnali: connessione verso i device

- **DREQ0** e **DREQ1** sono le linee di richiesta, usate dalle periferiche collegate ai rispettivi canali, per ottenere dei cicli DMA.
- Le richieste che arrivano su **DREQ0** hanno precedenza su quelle che arrivano su **DREQ1** (schema a priorità fissa).
- Le linee **DACK1** e **DACK2** informano la periferica, connessa a quel canale, che è stata selezionata per un ciclo DMA.



Architettura e Segnali: connessione verso la CPU

- La linea **HRQ** (Hold Request) è adoperata per spedire una richiesta al processore di controllo del bus (essa normalmente è applicata all'ingresso HOLD della CPU).
- Un segnale **HLDA** (hold Acknowledgement), in arrivo dalla CPU, segnala che è stato acquisito il sistema bus.
- Il segnale **EOP** indica l'avvenuta terminazione di un trasferimento di blocco (ASIM gestisce solo le modalità *single* e *block*)



Protocollo generale per trasferire da device a memoria

- Il device fa una richiesta al DMA quando deve trasferire dati in memoria.
- Il DMA richiede il bus al processore (master), e appena riceve il **BUSGRANT**:
 - Pone l'indirizzo base sull'address bus,
 - Manda un **ACK** al device,
 - Abilita i segnali **IORead** e **MemoryWrite**
 - Il device pone i dati sul bus e disabilita la sua richiesta
 - La memoria scrive i dati e si incrementa l'indirizzo base e decrementa il contatore di byte
 - Il ciclo continua, quando il contatore raggiunge lo zero il processo si ferma e viene inviata una interruzione al processore o al device per indicare “trasferimento completato”
- Il trasferimento dalla memoria al device è identico, ma vengono abilitati i segnali di **IOWrite** a **MemoryRead**.

Configurazione in ASIM

Name	I8237DMA
Type	Device
Identif	Intero (\$01..\$FF) che identifica univocamente l'oggetto in una configurazione
Address1	Indirizzo base del dispositivo
Address2	Indirizzo base+\$F (DMA occupa 16 locazioni di memoria)
BUS	Ident. del bus a cui il dispositivo è connesso
COM1	Ident. della CPU master per il trasferimento
COM2	vettore (b15-b8) priorità (b7-b4) linea Int (b3-b0)
COM3	Ident. Gestore interruzioni
COM4	bit0-3 e bit4-7 ident. dispositivi connessi al canale 0 e 1

- Il dispositivo didattico simulato in ASIM deriva dall'Intel 8237, ma presenta 2 canali invece che 4, e solo 2 modi di trasferimento, single e block

Registri (1/3)

- Il dispositivo è costituito da **due canali** DMA simmetrici programmabili mediante i registri:
 - BADDR: indirizzo base del trasferimento
 - CADDR: indirizzo corrente del trasferimento
 - BCOUNT: valore base del conteggio espresso come numero di byte da trasferire
 - CCOUNT: valore corrente del conteggio
- A tali registri si aggiunge:
 - il registro MODE che codifica la modalità del trasferimento
 - il flag RF che consente di avviare via software il trasferimento (senza la richiesta di una periferica) e
 - MF per mascherare il trasferimento sul canale.
- Il registro di controllo/stato CNTRL è comune ai due canali, così come il registro TEMP utilizzato per salvare provvisoriamente il dato da trasferire nei trasferimenti memoria-memoria

Registri (2/3)

- **CADDR0** e **CADDR1** (indirizzo corrente del trasferimento) hanno parallelismo 16 bit e contengono l'indirizzo della locazione di memoria interessata al trasferimento.
 - Nel caso del trasferimento da memoria a memoria CADDR0 contiene l'indirizzo sorgente e CADDR1 l'indirizzo destinazione.
 - Entrambi i registri sono accessibili sia in lettura che in scrittura ed il loro indirizzo relativo è 0 per CADDR0 e 2 per CADDR1.
-
- **BADDR0** e **BADDR1** sono i registri indirizzo di base ed hanno la funzione di memorizzare gli indirizzi iniziali rispettivamente di CADDR0 e CADDR1.
 - Essi sono a 16 bit e sono accessibili solo in fase di scrittura, infatti, quando viene scritto un valore in un registro indirizzo corrente, questo viene copiato anche nel relativo registro di base ed il valore rimane immutato fino a quando non si verifica un'altra scrittura.

Registri (3/3)

- **CCOUNT0** e **CCOUNT1** sono i registri di conteggio correnti, anch'essi a 16 bit, memorizzano il numero di byte da trasferire; sono accessibili in lettura e scrittura ed il loro indirizzo relativo è 1 per CCOUNT0 e 3 per CCOUNT1. Nel caso di trasferimento da memoria a memoria il conteggio viene effettuato da CCOUNT1.
- **BCOUNT0** e **BCOUNT1** sono registri a 16 bit che memorizzano il conteggio di base ed hanno la funzione di conservare gli indirizzi iniziali rispettivamente di CCOUNT0 e CCOUNT1, essi sono accessibili solo in fase di scrittura.
- Quando viene scritto un valore in un registro di conteggio corrente questo viene copiato anche nel relativo registro di base ed il valore rimane immutato fino a quando non si verifica un'altra scrittura.

Mode Register

- **MODE0** e **MODE1** sono i registri che contengono le informazioni relative al modo di funzionamento dei rispettivi canali. Essi sono di sola scrittura ed hanno entrambi indirizzo relativo pari a \$B;
- la selezione tra i due avviene sul valore del bit meno significativo del dato: se 0, il dato viene scritto in **MODE0** altrimenti in **MODE1**.

bit	significato
0	Se posto a 0 selezione il canale 0 se posto a 1 il canale 1
1-2	Bit non utilizzati
3	indica la direzione di trasferimento : 0 per trasferimenti da memoria ad interfaccia, 1 da interfaccia a memoria
4	il valore 1 abilita l'auto inizializzazione, cioè al termine del conteggio i registri indirizzo e conteggio corrente sono caricati con i valori dei rispettivi registri di base
5	il valore 1 abilita il decremento di una unità del valore contenuto in CADDR di quel canale dopo ogni trasferimento di un byte; deve essere posto a 0, se vogliamo l'incremento
6	Bit non utilizzato
7	determina il modo del trasferimento: 0 per il modo Single ed 1 per il modo Block ; nel primo caso il bus viene rilasciato al processore alla fine di ogni trasferimento, viceversa, nel modo block il bus viene rilasciato dopo il trasferimento dell'intero blocco

Control Register

- Il registro CNTRL è suddiviso in 2 parti: i 4 bit meno significativi rappresentano i bit di stato del componente, mentre quelli più significativi i bit di controllo. Su questo registro possono essere fatti accessi, all'indirizzo relativo \$8, sia in lettura che in scrittura

<i>bit</i>	<i>azione svolta se il bit è posto ad 1</i>
0	TC0: termine conteggio per il canale 0
1	TC1: termine conteggio per il canale 1
2	DREQ0: richiesta inoltrata al canale 0
3	DREQ1: richiesta inoltrata al canale 1
4	non utilizzato
5	abilita trasferimento da memoria a memoria
6	in un trasferimento memoria-memoria, impone che l'indirizzo sorgente deve rimanere costante per trasferire un byte in più locazioni di memoria (inizializzazione di un blocco di memoria a valore costante)
7	abilita il DMA controller

Flag

- **RF0** e **RF1** sono i flag dove indirizzare le richieste di tipo software al DMA, e producono gli stessi effetti di quelle provenienti dalle interfacce dei dispositivi.
- Questi flag sono accessibili solo in scrittura e l'indirizzo relativo per entrambi è \$9. Anche in questo caso la selezione del canale avviene sul bit meno significativo del dato: 0 per il canale 0 ed 1 per il canale 1.
- Il valore che deve assumere il flag deve essere posto sul bit numero 3 (quarto bit) del dato.
 - `MOVE.B #$08,9(A0)` mette 1 nel flag RF0 per avviare il DMA via software
- **MF0** e **MF1**, accessibili in sola scrittura, mascherano le richieste dei rispettivi canali se il mask flag di quel canale è posto ad 1.
- L'indirizzo relativo di entrambi i flag è \$A. Il canale è selezionato in base al valore del bit 0, il valore del flag va posto sul bit 2 del dato.

Trasferimento Memory to Memory

- E' possibile il trasferimento di blocchi dalla memoria alla memoria; in questo caso un byte del blocco da trasferire viene portato dalla sorgente ad un registro temporaneo interno al DMA, TEMP, e poi trasferito alla destinazione.
- Sono necessari due cicli e due canali, nel primo si indica la sorgente, nel secondo la destinazione