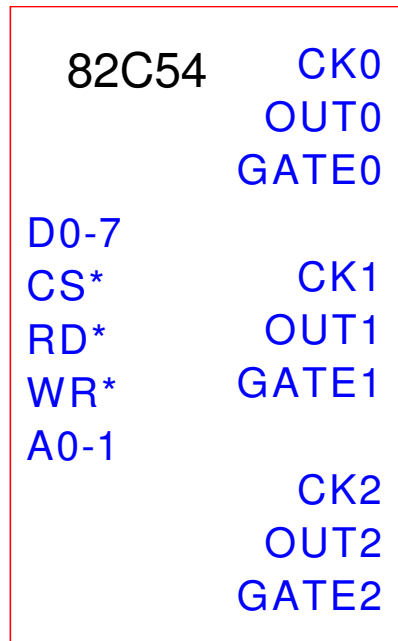


Principali periferiche

Timer

- ◆ Periferica per il conteggio esatto del tempo
- ◆ Esempio: 8254
- ◆ 3 contatori "indietro" da 16 bit: si può impostare un valore iniziale fino a 0xFFFF in ciascuno dei tre contatori; ad ogni fronte negativo del clock (CKi in figura) il contatore viene decrementato
- ◆ E' possibile leggere il valore corrente del conteggio
- ◆ 6 diverse modalità di conteggio; a seconda della modalità, si ha un impulso su OUTi al termine o durante il conteggio
- ◆ Quattro indirizzi: tre per i contatori (00, 01, 10) e uno per la porta di controllo (11)



Interrupt controller

Arbitra tra diverse richieste di interrupt provenienti da sorgenti diverse (IRi). Tra queste, sceglie quella di priorità più alta e la inoltra alla CPU. Se la CPU la accetta, trasmette a seguire un byte sul data bus, che è il vettore della interruzione

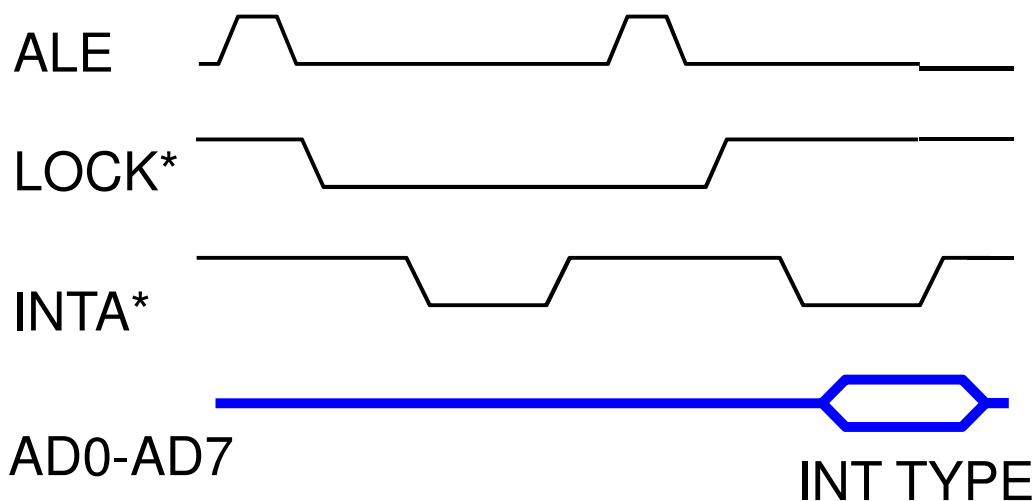
- ◆ Esempio: 8259
- ◆ Gestisce 8 interruzioni da solo e fino a 64 se connesso in cascata
- ◆ Occupa 2 locazioni di indirizzo
- ◆ Riconosce le interruzioni esterne a livello o sul fronte positivo
- ◆ Possibilità: impostazione delle priorità, programmazione dei vettori di interruzione, mascheratura degli IRi,...

8259

CS*	IR0
A0	IR1
WR*	IR2
RD*	IR3
D0-7	IR4
CAS0-2	IR5
INT	IR6
INTA*	IR7
SP/EN*	

Interrupt acknowledge

- Interrupt acknowledge: riconoscimento dell'interruzione da parte da parte della CPU
- La CPU riceve il segnale di INT dall'8259 e, se le interruzioni non sono mascherate, genera due impulsi del segnale di INTA* per l'8259:
- Al primo impulso di INTA*, l'8259 decide quale tra le richieste di interruzione IRI sia quella a più alta priorità e debba quindi essere inviata alla CPU
- Al secondo impulso di INTA*, l'8259 rilascia sul bus il vettore di interruzione (semplicemente, un byte, il cui valore tra 0 e 255 specifica quale routine di interruzione debba essere eseguita dalla CPU)



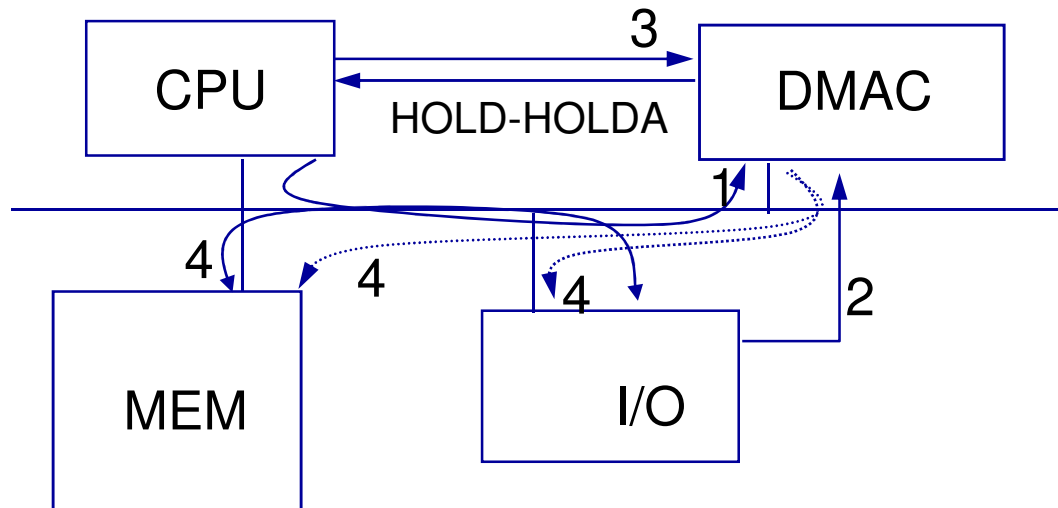
Direct Memory Access

Direct Memory Access (DMA)

- ◆ Meccanismo per cui, sotto controllo hardware di un **DMAC** (DMA Controller), una periferica si interfaccia direttamente con la memoria senza passare per la CPU. Il DMAC prende il controllo del bus e gestisce il trasferimento.

- ◆ Vantaggi:
 - il DMAC può essere più veloce della CPU nel trasferire i dati dalla periferica alla memoria
 - la CPU può proseguire l'elaborazione se non ha bisogno di accedere al bus (ad esempio, può eseguire istruzioni su operandi registro o immediati)

DMA

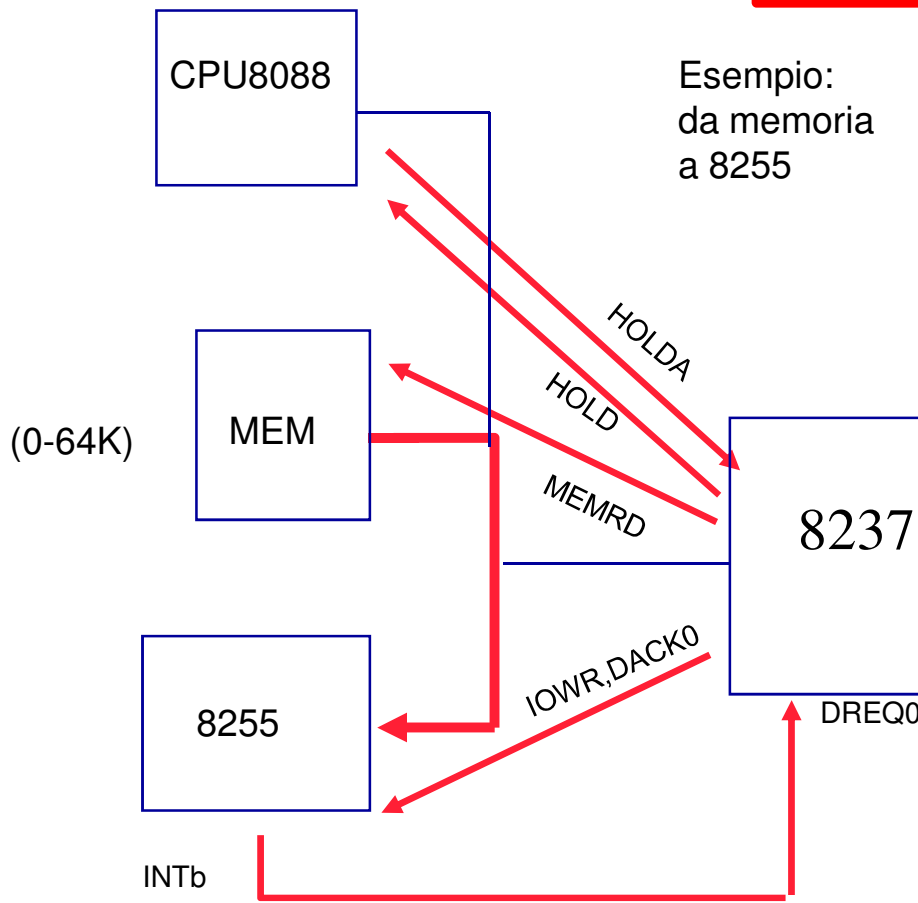


- 1) La CPU programma il DMAC
 - 2) Al DMAC arriva la richiesta di trasferimento
 - 3) il DMA requisisce il controllo del bus alla CPU (invia HOLD e riceve HOLDA: bus requisito)
 - 4) il DMAC genera i segnali per gestire il trasferimento: indirizzi per la memoria, selezione per la periferica, MEMRD e IOWR o MEMWR e IORD a seconda della direzione
- ◆ **Trasferimento fly-by** dati non attraversano il DMAC, ma passano direttamente dalla periferica alla memoria, o viceversa
 - ◆ **Trasferimento flow-through** dati attraversano il DMAC e vengono memorizzati in registri interni del DMAC (permette così il trasferimento anche memory to memory)

DMAC 8237

- ◆ DMAC 8237
- ◆ gestisce fino a 4 richieste di DMA
- ◆ bus address e dati multiplexato
- ◆ i segnali di DACK (acknowledge) sono utilizzati per il CS della periferica

IORD*	
IOWR*	
MEMRD*	DREQ0
MEMWR*	DACK0
READY	DREQ1
HOLD	DACK1
HLDA	DREQ2
AEN	DACK2
ADSTB	DREQ3
CS*	DACK3
CLK	EOP*
RESET	
D0-7	
A0-7	



DMAC: Modalità

Il DMAC viene in genere utilizzato per trasferire blocchi di dati dalla periferica a indirizzi consecutivi di memoria, o viceversa.

Tre diverse modalità:

- 1) **Single transfer mode** Viene eseguito un singolo trasferimento di dato e poi rilasciato il bus per almeno un ciclo (il DMA pone HOLD a 0); questo consente alla CPU di proseguire la sua elaborazione intercalandosi sul bus con il DMAC
- 2) **Block mode** Una volta requisito il bus, il DMAC effettua il trasferimento di tutti i dati (si assume che la memoria e la periferica siano sempre pronte al trasferimento di un dato)
- 3) **Demand mode** Come sopra, ma non si assume che la periferica sia sempre pronta; la periferica dichiara la propria disponibilità tenendo alto DREQ. Se la periferica non è pronta, si restituisce nel frattempo il bus alla CPU.

Controllore USB

Gestisce la comunicazione con il bus USB, bus seriale per la connessione di fino a 127 dispositivi in topologia ad albero

- ◆ Segnale seriale differenziale D+, D-, più alimentazione a +5 V (VBUS) e massa
- ◆ Banda passante del bus USB 1.1: 1.5 MB/s (12 Mb/s) dell'USB 2.0: 480 Mb/s
- ◆ Encoding: NRZI (Non Return to Zero Invert):
 - se trasmette un 1: il segnale conserva il valore precedente
 - se trasmette uno 0: il segnale commuta rispetto al valore precedente
- ◆ Esempio: Philips PDIUSB12:

