



Architettura degli elaboratori

DMA

Dott. Franco Liberati

ARGOMENTI DELLA LEZIONE

- ❑ DMA

- ❑ DMA Motorola MCF5307



DMA



Architettura degli elaboratori

Interazione con i dispositivi di I/O



I/O

Generalità

- ❑ Un calcolatore comunica con l'esterno attraverso apposite apparecchiature denominate **dispositivi esterni** (periferiche o device I/O), che consentono di trasferire informazioni dalla memoria interna all'esterno (dispositivi di output) e viceversa (dispositivi di input)
- ❑ I dispositivi di I/O hanno **caratteristiche peculiari** a seconda delle componenti che sono usate e delle funzioni che svolgono nonché dalla natura delle informazioni trasferite, al modo in cui queste sono rappresentate esternamente al calcolatore
- ❑ Queste caratteristiche influenzano uno dei parametri significativi per i dispositivi di I/O: **la velocità di trasferimento** definita come numero di parole copiate in un secondo dalla memoria principale al dispositivo periferico e viceversa
- ❑ La velocità di trasferimento varia a seconda della struttura fisica dei dispositivi (qualora intervengano componenti elettromeccaniche ci sarà un rallentamento maggiore rispetto ai dispositivi fatti con componenti elettronici)

DISPOSITIVI	Velocità di trasferimento media (ordine di grandezza)
Tastiere	10 byte/sec
Terminali Video	10^2 - 10^4 byte/sec
Stampanti	10^2 - 10^4 byte/sec
Nastri Magnetici	10^7 byte/sec
Dischi Magnetici	10^8 byte/sec
Dischi Ottici	10^7 byte/sec
Convertitori Analogico-Digitali	10^7 byte/sec
Dispositivi per collegamento reti di calcolatori	10^7 - 10^8 byte/sec
Memorie a stato solido	10^8 - 10^9 byte/sec
Processore	10^9 byte/sec



I/O

Strategia di trasferimento dati

- ❑ I **metodi di trasferimento dati I/O** prevedono che vi sia scambio di informazione tra la memoria e il dispositivo utilizzando dei protocolli per il trasferimento
- ❑ Pertanto è possibile operare:
 - ❖ utilizzando un ciclo di attesa con una istruzione che testa l'effettiva disponibilità del dispositivo (**daisy chain**)
 - ❖ mediante una routine di servizio attivata dal dispositivo nel momento in cui è pronto (**interruzione**)



I/O

Accesso Diretto alla Memoria DMA

- ❑ Con il **DAISING CHAIN** si riscontra una perdita significativa del tempo in cui la CPU potrebbe essere coinvolta in elaborazioni più rilevanti come, ad esempio, l'esecuzione di una istruzione di un programma
- ❑ Un sistema a **INTERRUZIONE** risulta essere efficiente se il trasferimento è per pochi dati, ma diventa effimera se il trasferimento dati coinvolge un gran numero di parole (in questo caso la velocità di trasferimento è limitata dalla durata del servizio dell'interruzione) e soprattutto se lo scambio avviene non in maniera consecutiva
 - ❖ Alcuni dispositivi hanno la capacità di trasferire molte parole in un secondo (es.: i dischi magnetici trasferiscono i dati a gruppi di 4KB)

Osservazione.

Se il dispositivo è veloce, la tecnica ad interrupt pone dei limiti alla velocità di trasferimento dei dati.

Si supponga infatti che la routine di servizio dell'interrupt trasferisca un byte alla volta richiedendo 20ms per l'esecuzione.

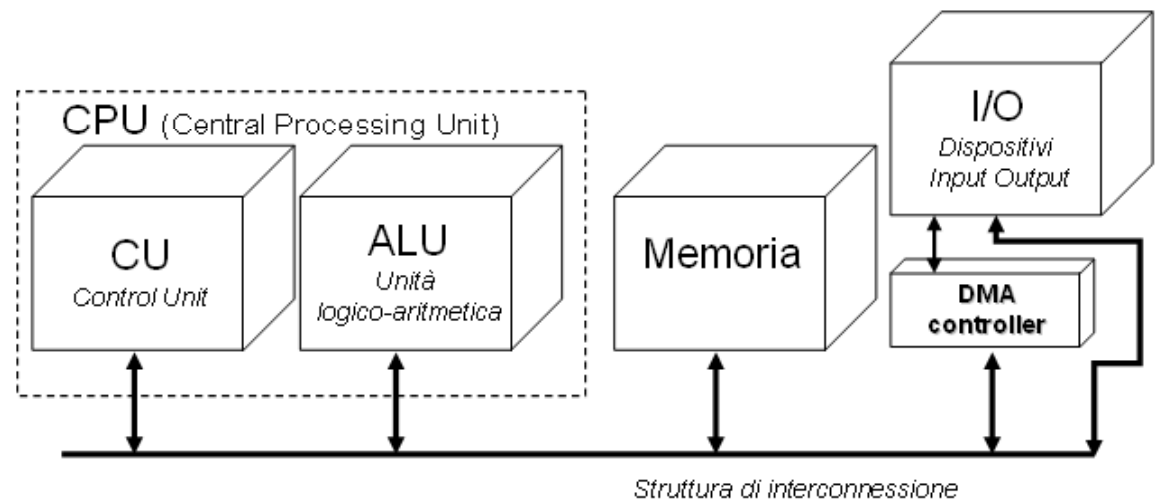
Supponendo inoltre che il tempo necessario per la commutazione del contesto sia di 5ms, il tempo necessario per il trasferimento di un singolo byte risulta essere di 25ms.

Ciò significa che la velocità massima di trasferimento è di 40KBytes/s, con un impegno di CPU del 100%

Accesso Diretto alla Memoria DMA

Generalità

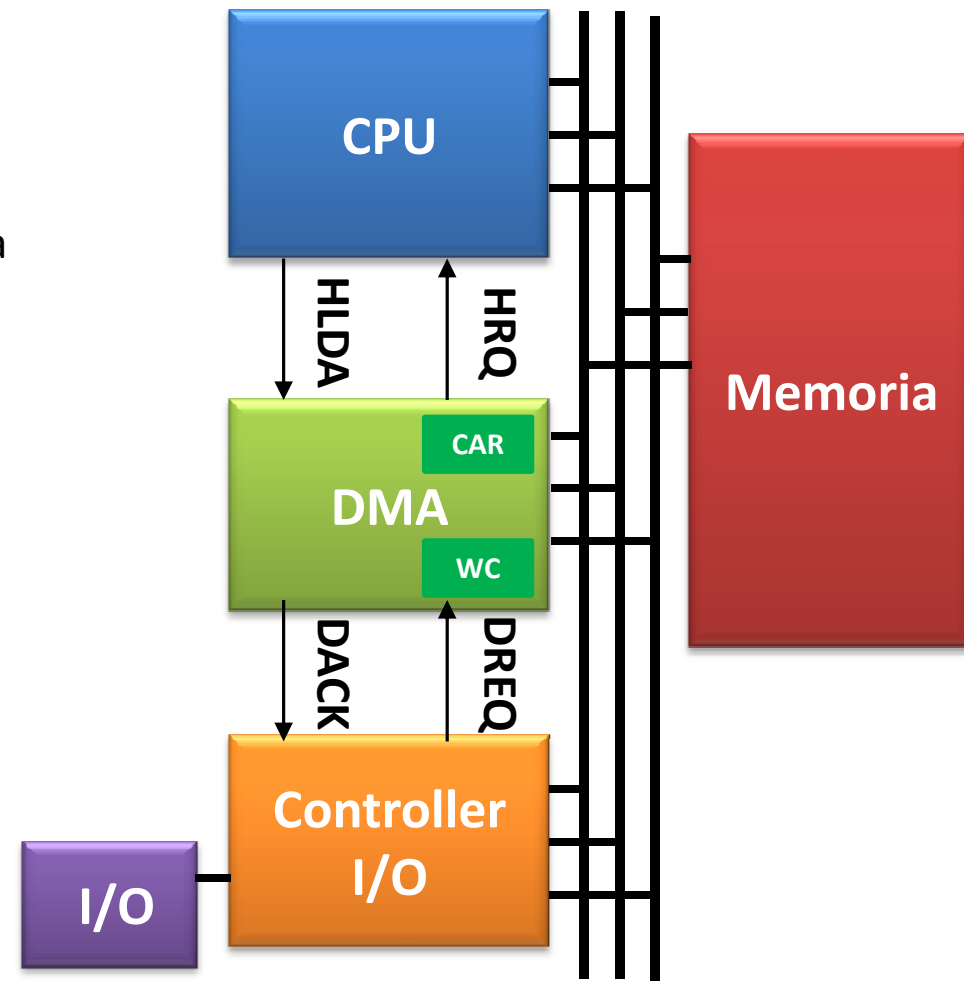
- ❑ Per superare questi limiti utilizza l'**accesso diretto alla memoria** (*access direct memory, DMA*) in cui i dati sono trasferiti autonomamente ad opera di un dispositivo che può accedere direttamente alla memoria senza l'intervento costante del processore (si parla anche di *I/O autonomo*)



Accesso Diretto alla Memoria DMA

Componenti fisiche di base

- ❑ Il DMA si ottiene grazie ad un hardware dedicato, il **controllore del DMA** (DMA controller) presente nel modulo di interfaccia di ciascun dispositivo di I/O
- ❑ Nel DMA controller sono presenti (nella forma più ridotta):
 - ❖ Il registro **CAR** (*Current Address Register*): che svolge la funzione di puntatore alla locazione di memoria destinata a ricevere (input) o a fornire (output) il prossimo dato da trasferire. A volte si sdoppia in due registri con l'indirizzo sorgente (**SAR**) e quello di destinazione (**DAR**)
 - ❖ il **WC** (*Word Counter*): il contatore dei dati da trasferire (in alcuni testi **BCR**, *byte count register*)
 - ❖ Un registro di stato
- ❑ Inoltre ci sono delle linee di comando atte a gestire la richiesta e la relativa assegnazione del bus per il trasferimento dati, in modo tale da scongiurare eventuali conflitti di accessi contemporanei del processore e di altri dispositivi





Accesso Diretto alla Memoria DMA

Funzionamento

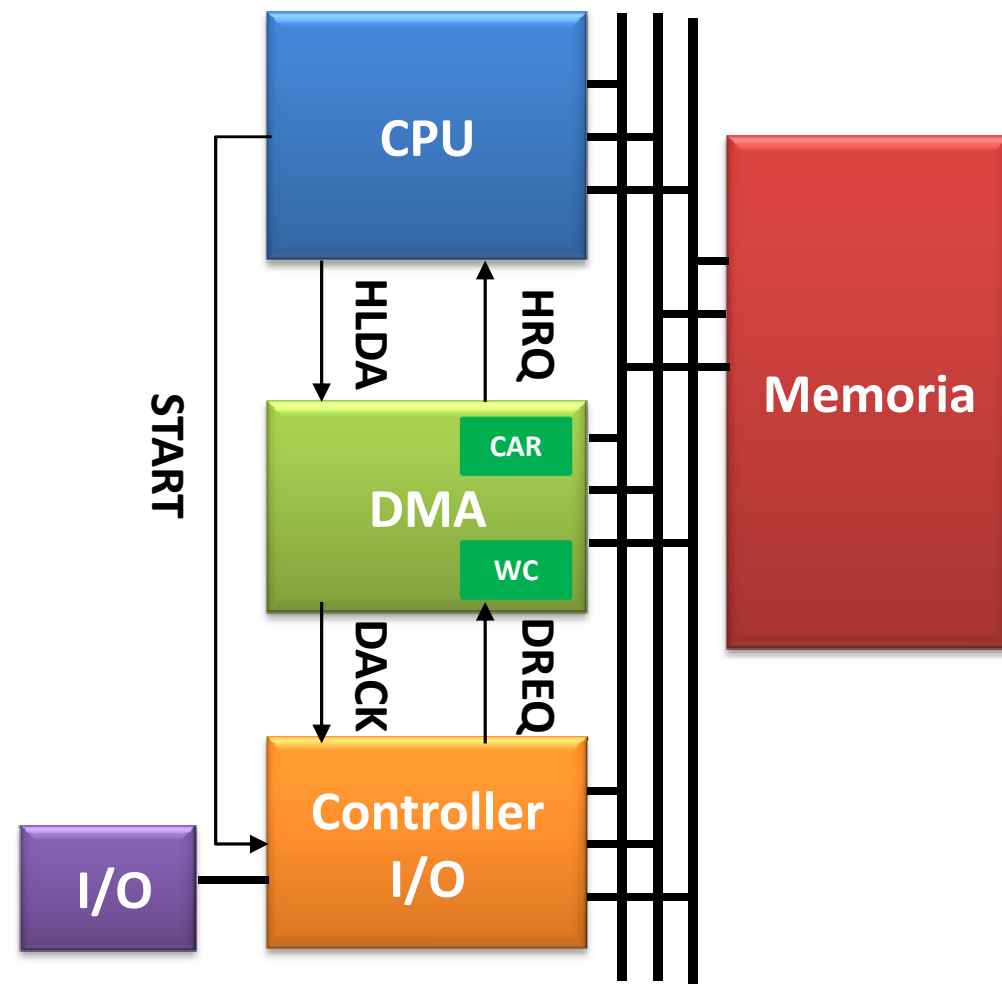
- ❑ I passi elementari che devono essere fatti per eseguire un **trasferimento** possono essere sintetizzati come segue:
 1. Si accede, grazie all'informazione contenuta nel registro CAR alla posizione di memoria interessata al trasferimento
 2. Si trasferisce la parola lungo il bus;
 3. Si incrementa o decrementa (a seconda dell'organizzazione delle parole nel calcolatore elettronico) il CAR
 4. Si decrementa il contenuto di WC
- ❑ Dopo che l'ultimo dato è stato copiato, il portarsi a zero del contenuto del registro WC provoca la richiesta di una interruzione di fine trasferimento dati
- ❑ Come è possibile capire, dalla sequenza sopra descritta, **il processore non esegue alcuna istruzione per la copia dei dati**, il suo intervento rimane confinato solo per predisporre le informazioni necessarie per comandare il trasferimento

Accesso Diretto alla Memoria DMA

Protocollo

❑ Le fasi che interessano in trasferimento di un blocco di dati utilizzando il DMA possono essere così descritte (es.: salvataggio di una immagine digitale su un dispositivo di output, disco magnetico o pen drive):

1. L'utente fa la richiesta di salvataggio
2. La routine di interruzione di salvataggio del file individua il dispositivo di output e **inizializza i registri CAR e WC** del relativo DMA. Infine invia un segnale di inizio **START_IO** al dispositivo. Il dispositivo è abilitato ad avviare le operazioni di trasferimento utilizzando il DMA controller

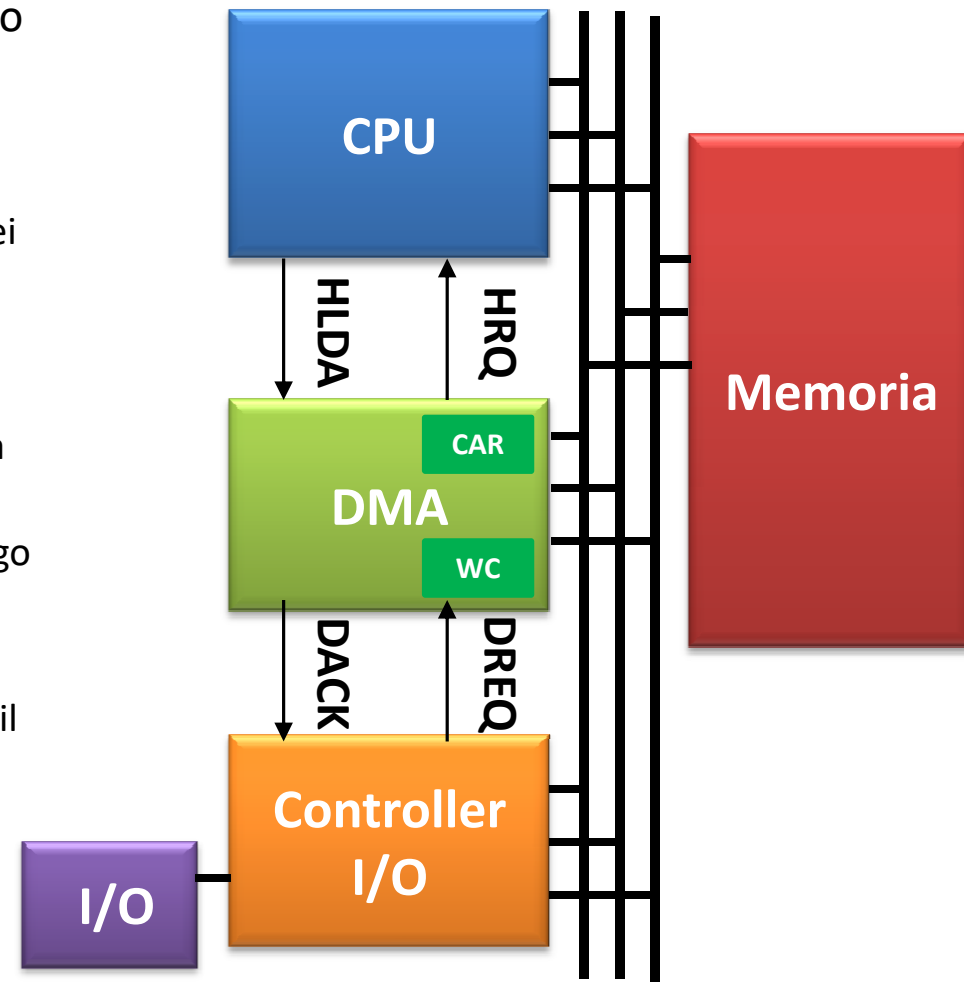


Accesso Diretto alla Memoria DMA

Protocollo

3. La memoria, il DMA controller e il dispositivo di output operano così:

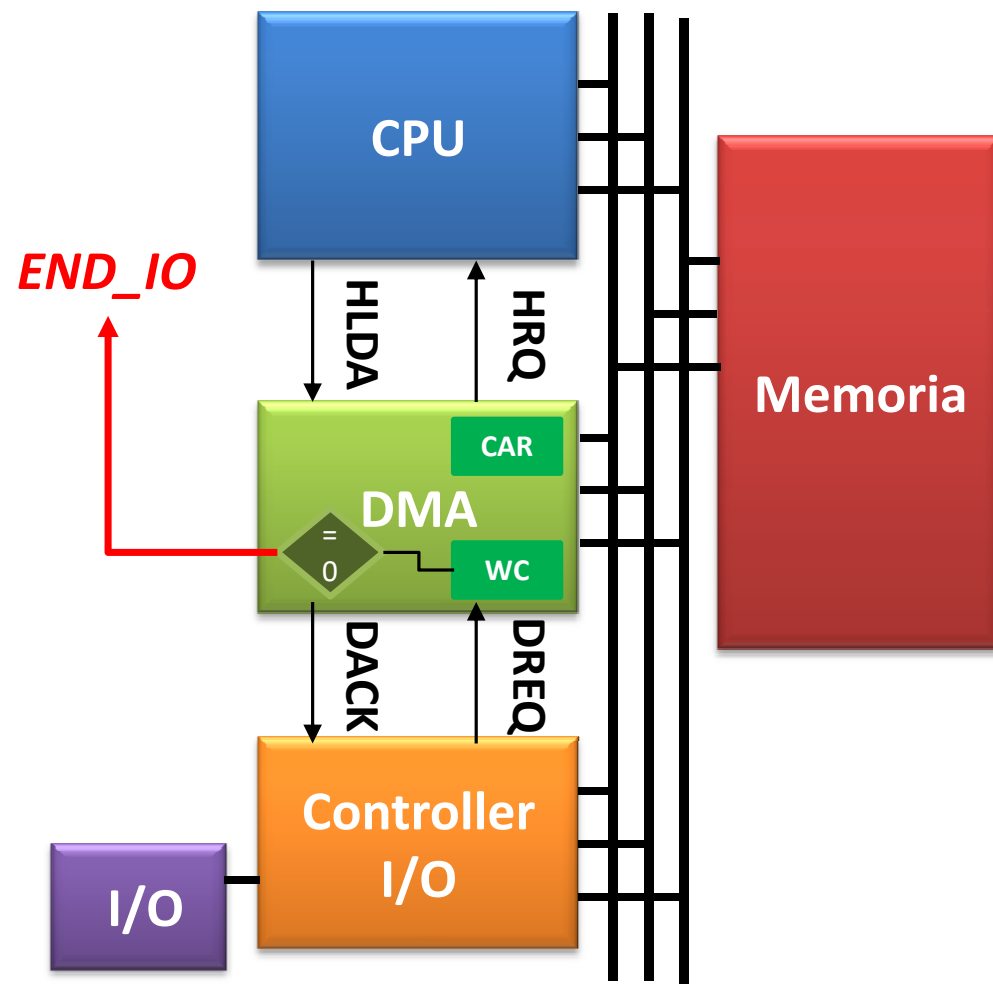
- Il dispositivo di output invia la **richiesta di uso del DMA** (DREQ, *DMA request*) settando dei parametri nel registro di stato interno al DMA (es.: se si tratta di una lettura e scrittura, l'indirizzo di destinazione dei dati nel dispositivo,...)
- Il **DMA richiede** alla CPU l'**uso del bus** (HRQ, *Hold request*)
- la **CPU** appena ha finito **rilascia il bus al DMA** (HLDA, *Hold acknowledgement*). Il segnale HRQ rimane attivo per tutta la richiesta del dispositivo (**furto di un ciclo**, *cycle stealing*)
- il DMA invia l'**indirizzo CAR** (da dove prelevare i dati in memoria) lungo il bus degli indirizzi
- Il DMA invia alla periferica **un segnale di conferma** (DACK, *DMA ack.*)
- Il DACK consente di iniziare il trasferimento dati dalla memoria verso il dispositivo di output
- Ad ogni trasferimento si incrementa o decrementa (a seconda dell'organizzazione delle parole nel calcolatore elettronico) il **CAR** e si decrementa il contenuto di **WC**



Accesso Diretto alla Memoria DMA

Protocollo

- Quando il registro **WC** arriva a zero (tutti i dati sono trasferiti) il DMA controller disattiva HRQ consentendo alla CPU di riprendere il controllo sul bus e genera una richiesta di interruzione **END_IO** (o DMA_INT)
- Si attiva la routine di servizio che, eventualmente, recupera l'area di memoria che conteneva i dati trasmessi (output), aggiorna la tabella dei frame liberi,...





Accesso Diretto alla Memoria DMA

Accesso al bus e furto di un ciclo

Osservazione. Le operazioni di accesso al bus di memoria avvengono in intervalli temporali di durata corrispondente al tempo di accesso in memoria stessa. Ciascuno di questi intervalli può essere utilizzato sia dal processore (ad esempio nella fase di LOAD) che dai dispositivi (ad esempio per una operazione DMA) i quali peraltro hanno un diritto di prelazione nei sistemi real time (perché altrimenti ci potrebbe essere una perdita dei dati rilevati)

In realtà, la logica di assegnazione del bus è conglobata nel processore e realizzata con la modalità master slave, tale cioè che il processore ne abbia il controllo (master) e ne rilascia l'uso ai dispositivi (slave) quando questi ultimi ne fanno richiesta

Nel caso in cui ci siano conflittualità, il processore rimanda ad un ciclo successivo la propria operazione di accesso alla memoria privilegiando le richieste di DMA (si parla anche di *cycle stealing*, furto di un ciclo). Ovviamente mentre le richieste di interruzione avvengono sempre alla fine dell'esecuzione di una istruzione, le richieste di DMA possono essere accettate alla fine di ogni singolo ciclo di accesso alla memoria. Inoltre si può evidenziare che tanto più è elevata la velocità e la quantità dei trasferimenti DMA tanto è maggiore la probabilità che si verifichino conflitti che determinano lo slittamento dei cicli di tempo utilizzati dal processore e quindi un rallentamento nell'esecuzione del programma.

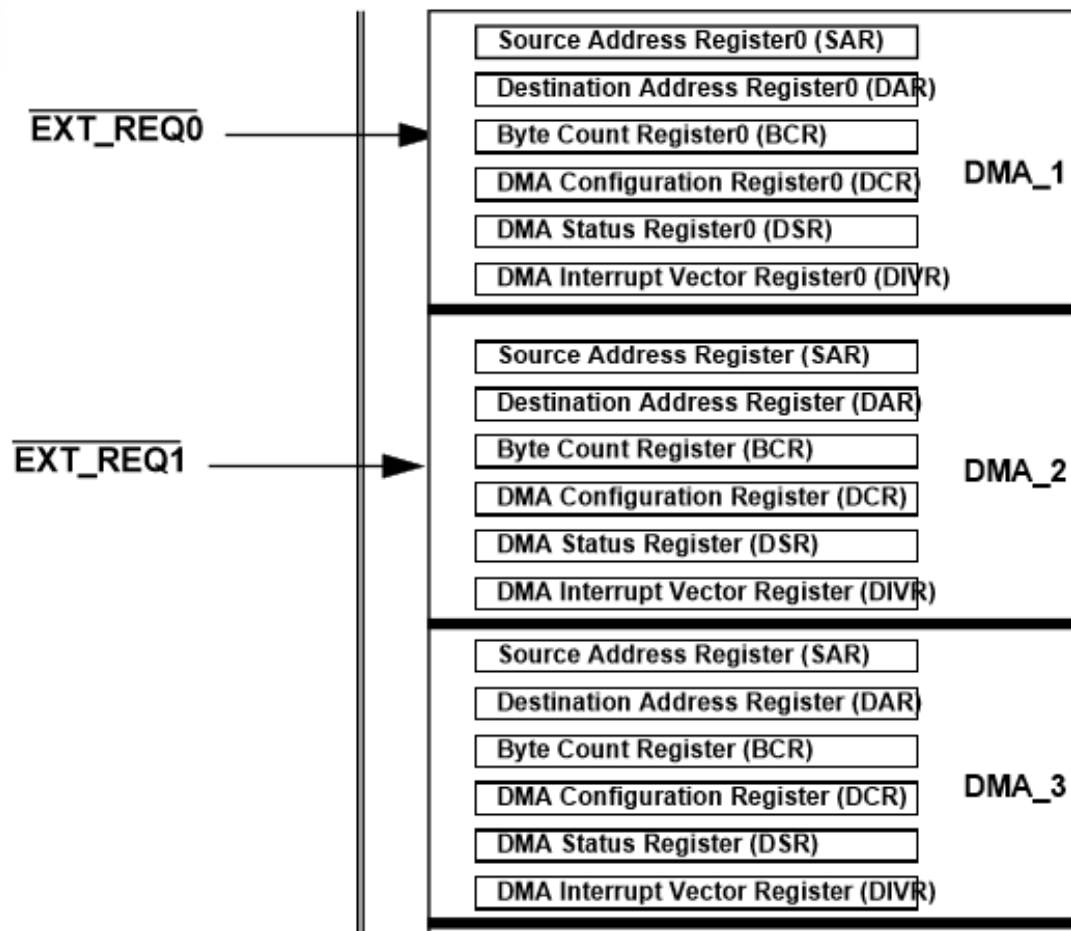
Nelle attuali architetture il furto di ciclo è un caso superato perché è possibile accedere direttamente alle celle di memoria (grazie anche all'introduzione della cache) e in modalità parallela (con i multibus)



Un caso reale **MOTOROLA MCF5307**

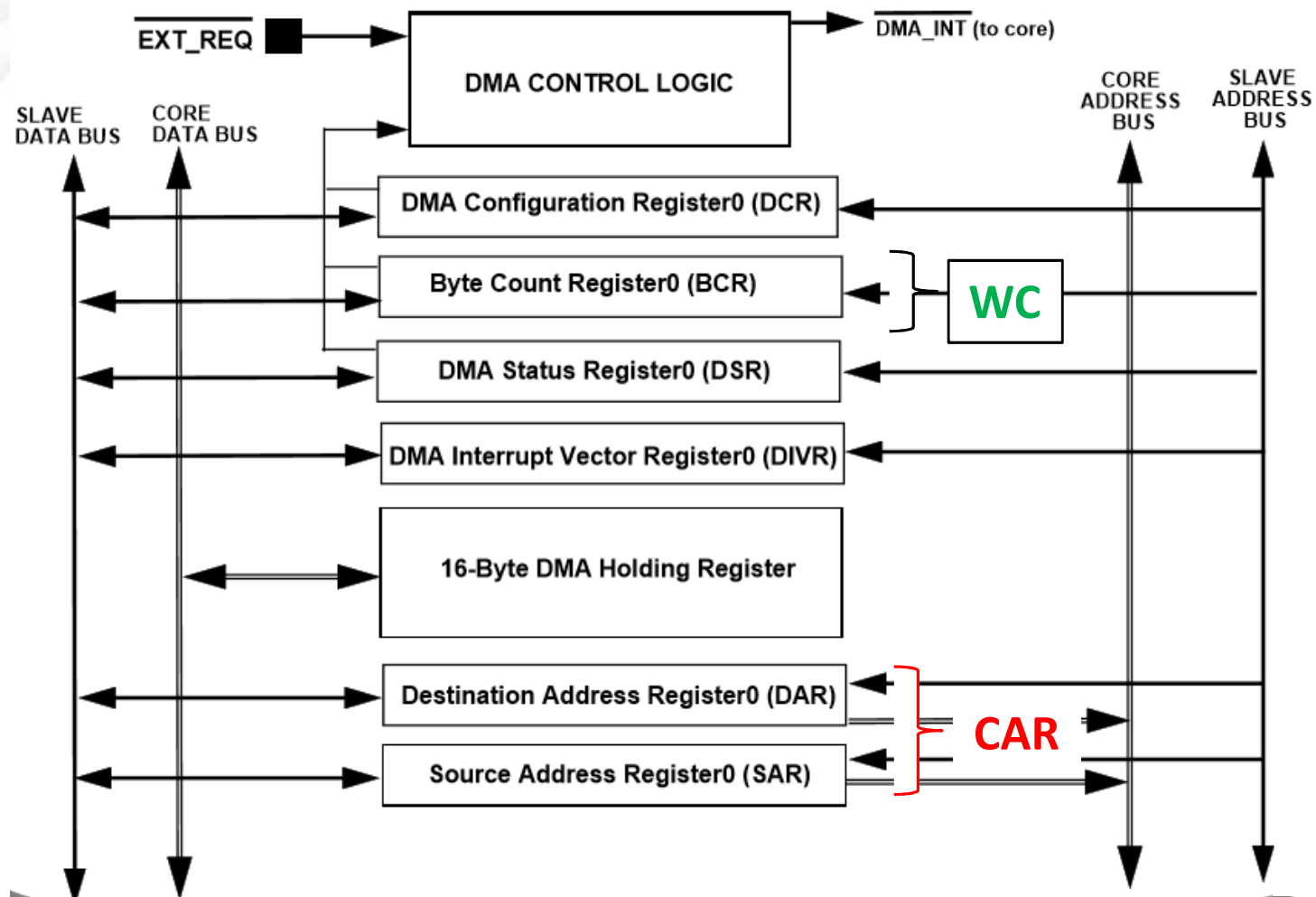
DMA Motorola MCF5307

Schema generico dei DMA dei diversi dispositivi



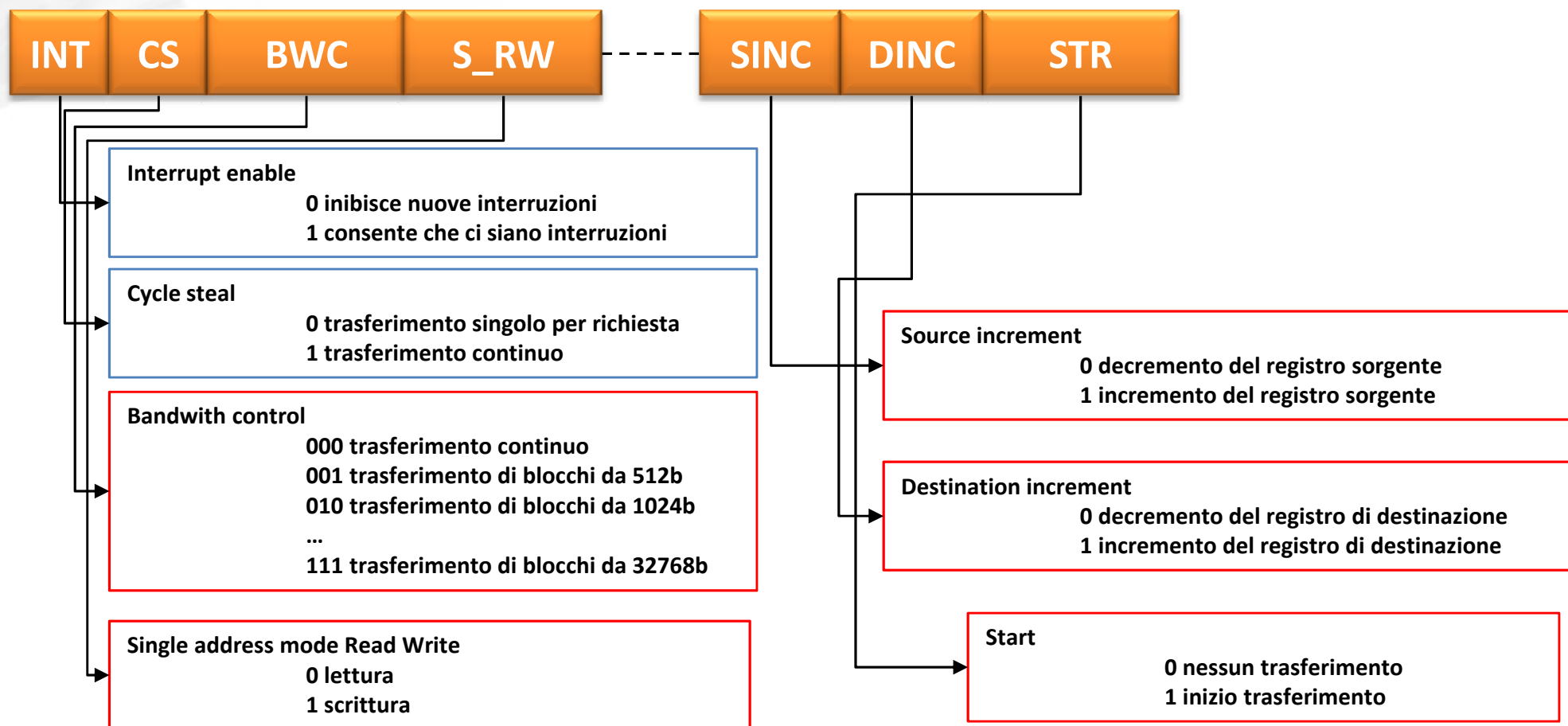
DMA Motorola MCF5307

DMA



DMA Motorola MCF5307

DMA: registro di controllo (campi principali)



DMA Motorola MCF5307

DMA: registro di stato



Bus Error on Source

1 lettura terminata con un BusError in trasferimento dalla sorgente
0 lettura terminata correttamente

Bus Error on Destination

1 scrittura terminata con un BusError durante il trasferimento in destinazione
0 scrittura terminata correttamente

Configuration error

0 nessun errore nel registro contatore
1 errore

Done

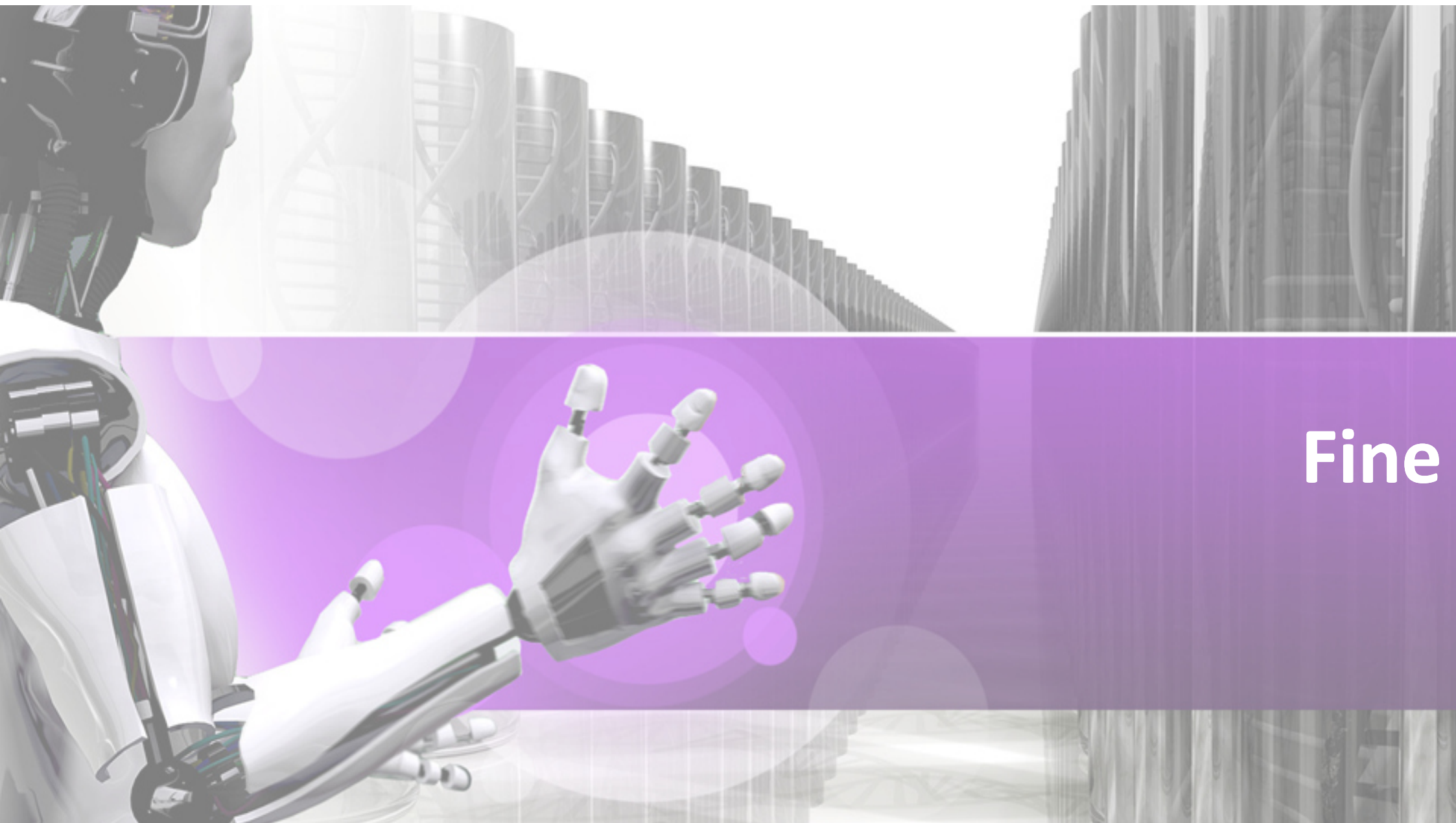
1 trasferimento svolto dal DMA finito
0 trasferimento svolto dal DMA in corso

Busy

1 trasferimento lungo il canale di comunicazione DMA in corso
0 trasferimento lungo il canale di comunicazione DMA eseguito

Request

0 DMA non ha richiesto il trasferimento
1 DMA ha richiesto il trasferimento



Fine