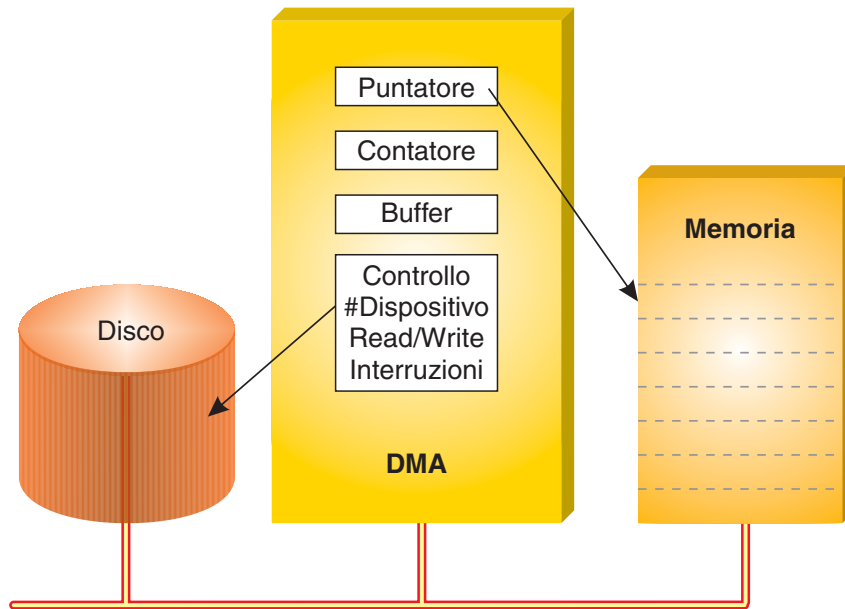


L'accesso diretto alla memoria e le architetture DMA

Come si è visto nel Paragrafo 3, il DMA realizza il trasferimento diretto dei dati dall'unità periferica alla memoria senza coinvolgere i registri del processore e liberando la CPU dalla gestione del trasferimento.



Nell'hardware del DMA ci sono diversi registri:

- un registro **Puntatore** che contiene la posizione in memoria dove collocare, o dove prelevare, i dati da trasferire;
- un registro **Contatore** dove è memorizzato il numero di byte da trasferire;
- un registro **Buffer** usato dal DMA come memoria temporanea per spostare i dati tra memoria e periferica;
- un insieme di registri per il controllo del DMA.

Il controllo del DMA è attuato dal processore che specifica nei registri di controllo il numero (**#Dispositivo**) del dispositivo interessato al trasferimento e il tipo di operazione da eseguire (**Read/Write**). Il dispositivo può essere abilitato a eseguire le interruzioni mediante un apposito bit (**Interruzioni**) nei registri di controllo.

Per effettuare un'operazione di I/O, per esempio la lettura di 650 byte dal dispositivo numero 55 e il trasferimento in memoria a partire dell'indirizzo 135700, il processore memorizza nei registri del DMA:

- l'indirizzo di memoria destinazione del trasferimento, *135700 in Puntatore*;
- il numero di byte da trasferire, *650 in Contatore*;
- il dispositivo interessato al trasferimento, *55 in #Dispositivo*;
- l'operazione da eseguire, *Read*.

A questo punto il processore abilita le interruzioni, attiva il DMA e, sotto il controllo del sistema operativo, cambia contesto di esecuzione per eseguire un altro processo. Il trasferimento dei dati dall'unità periferica alla memoria, attuato dall'hardware del DMA, è descritto dal seguente codice.

```

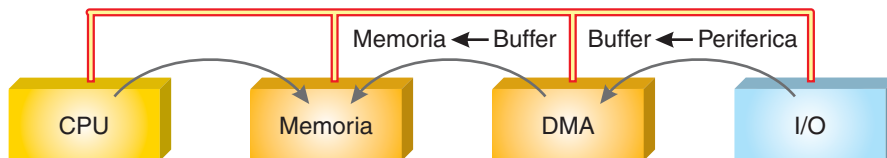
/*-----
 * Azioni del DMA: operazione di lettura da una periferica
 *-----*/
do
{
  buffer = leggiDaPeriferica(55);
  memoria[puntatore] = buffer;
  puntatore++;
  contatore--;
}
while ( contatore > 0 );
InviaSegnaleInterruzione();

```

Al termine del trasferimento dei dati in memoria il DMA invia un segnale di interruzione al processore

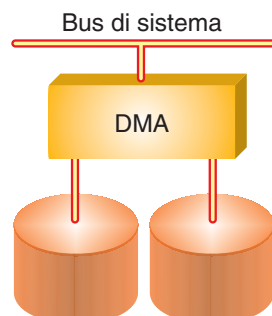
In sostanza il DMA effettua il trasferimento dei dati senza utilizzare i registri del processore e, al termine del trasferimento, invia un segnale di interruzione per segnalare che il trasferimento è stato completato.

Il DMA può essere realizzato secondo diverse architetture. In quella più semplice, un solo DMA serve diversi dispositivi tramite il *bus di sistema*.

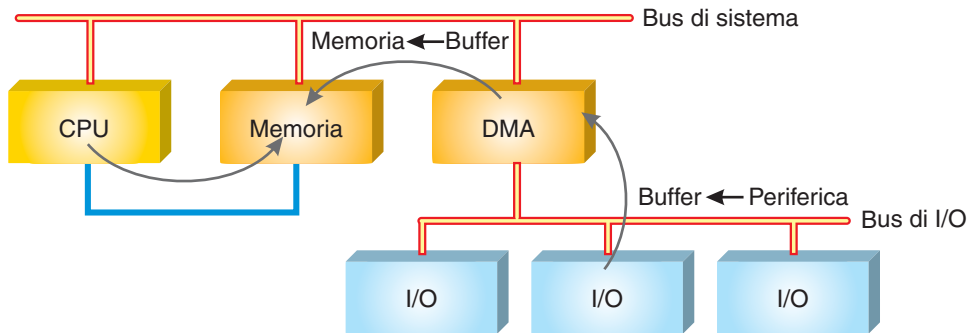


Poiché il bus è condiviso, l'azione del DMA causa l'aumento del tempo di accesso alla memoria da parte del processore. Si noti anche che, con questa architettura, ogni trasferimento di dati dalla periferica alla memoria richiede due accessi al bus di sistema, aumentando la probabilità che il processore trovi il bus occupato.

Una differente organizzazione del DMA, che offre prestazioni migliori, si ottiene integrando il DMA con uno o più dispositivi di I/O, per esempio integrando l'hardware del DMA nel controller dei dischi. In questo modo il trasferimento dei dati dalla periferica al DMA non utilizza il bus di sistema, che è impegnato solo per il trasferimento dal DMA alla memoria centrale. La probabilità che il processore trovi il bus occupato è dimezzata e il rallentamento del processore diminuisce.



In un'architettura più efficiente, il DMA riceve i dati dai dispositivi di I/O attraverso un **bus di I/O** indipendente dal bus di sistema: quest'ultimo è usato solo per trasferire i dati tra DMA e memoria.



Inoltre un *bus locale* ad alta velocità effettua il trasferimento dei dati tra memoria e processore, al di fuori del bus di sistema. In questo modo gli accessi alla memoria da parte della CPU e del DMA avvengono lungo percorsi differenti e possono evolvere in parallelo. Il processore risulta comunque rallentato, perché la memoria non può essere letta o scritta in parallelo e il controllore della memoria, che gestisce le richieste di accesso, deve accodarle in sequenza.