

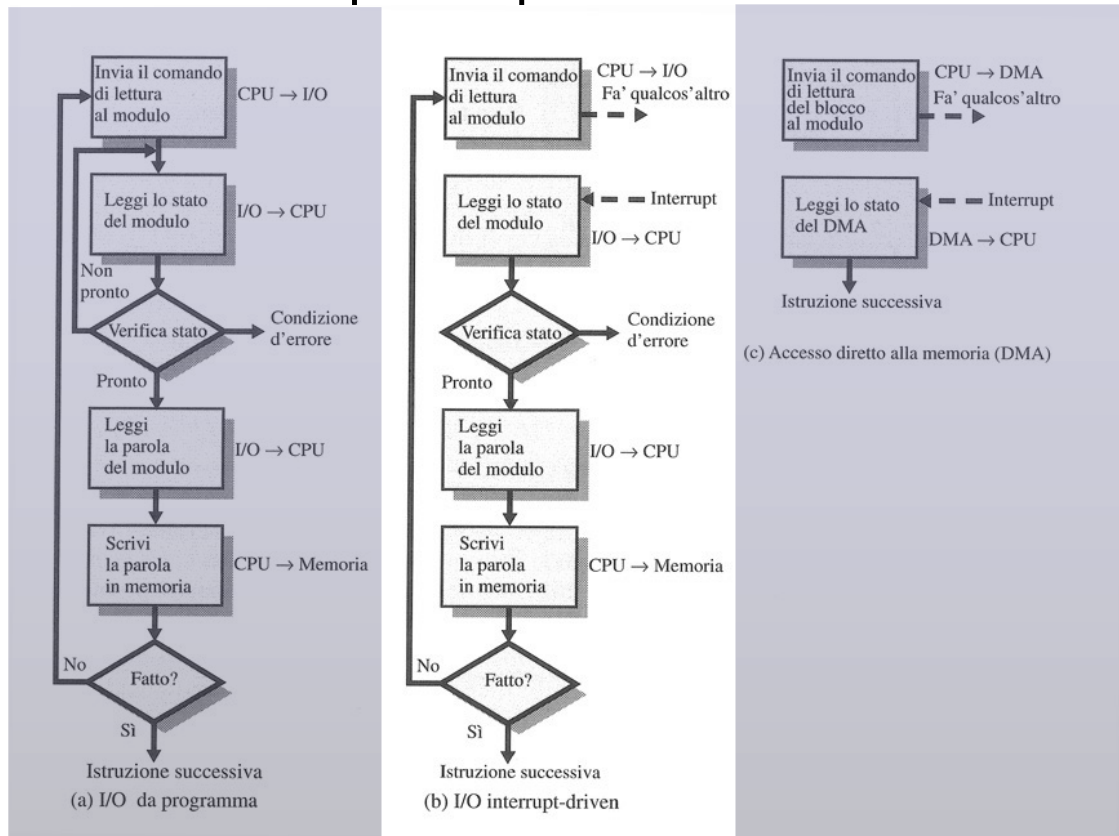
# I/O Interrupt Driven



- Evita l'attesa da parte della CPU
- Nessun controllo ripetuto dello stato del dispositivo da parte della CPU
- Il modulo di I/O interrompe la CPU quando è pronto



## Tre tecniche per l'input di un blocco di dati



# I/O Interrupt Driven Operazioni Base



- CPU rilascia comando di lettura
- Modulo I/O ottiene i dati dalla periferica mentre la CPU svolge altro lavoro
- Modulo I/O interrompe la CPU
- CPU richiede i dati al modulo I/O
- Modulo I/O trasferisce i dati alla CPU



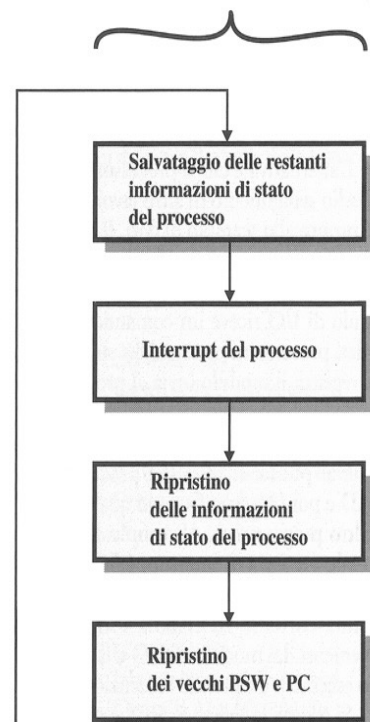
## Semplice elaborazione delle interruzioni



### Hardware



### Software

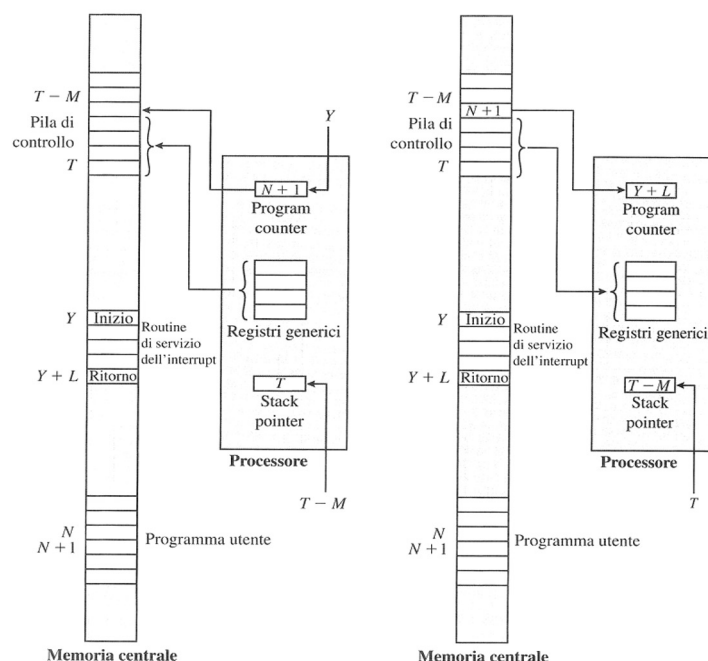


# Punto di Vista della CPU

- Rilascia comando di lettura
- Esegue altro lavoro
- Controlla se c'è interruzione alla fine di ogni ciclo di istruzione (ciclo fetch/execute con trattamento delle interruzioni)
- Se interruzione presente:
  - Salva contesto (PC e registri)
  - Interruzione del processo corrente e elaborazione interrupt
    - Lettura dati da modulo I/O e scrittura in memoria



## Cambiamento in Memoria e Registri per un Interrupt

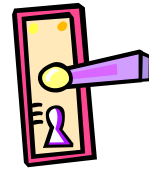


(a) L'interrupt avviene dopo un'istruzione all'indirizzo N

(b) Ritorno dall'interrupt

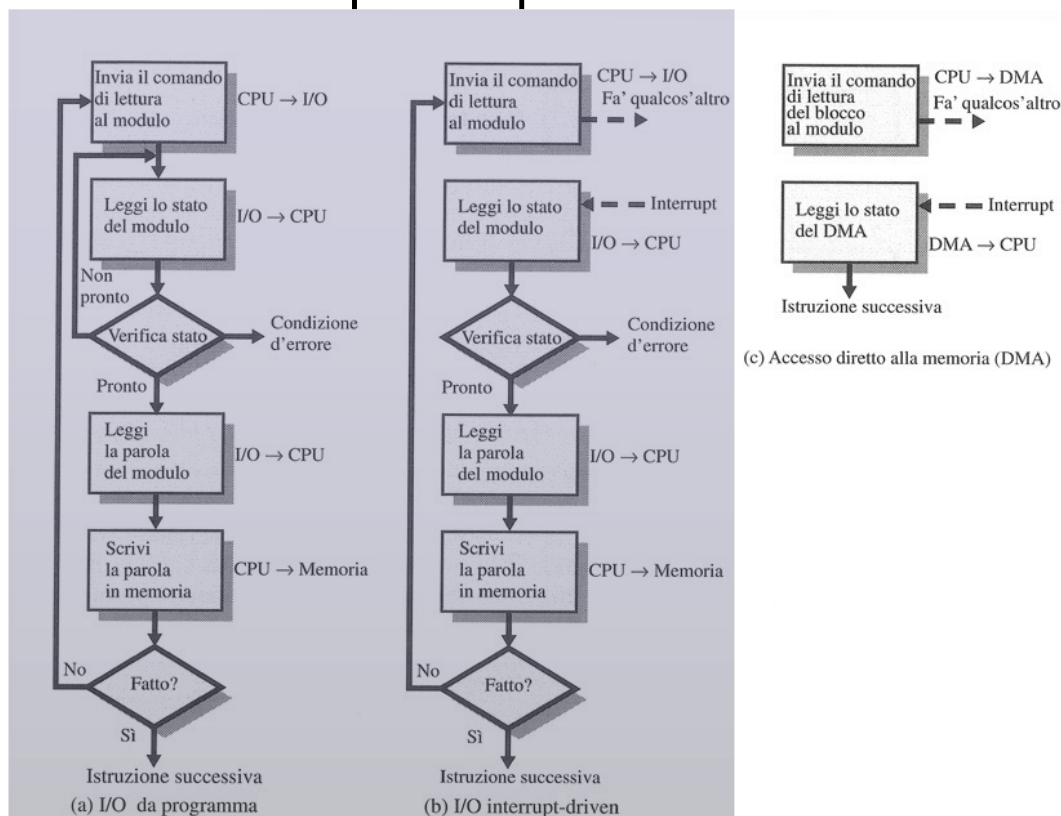


# Accesso Diretto alla Memoria (Direct Memory Access)



- Sia I/O da programma che interrupt driven richiedono l'intervento attivo della CPU
  - Il tasso di trasferimento dei dati è limitato
  - CPU è impegnata in tali operazioni e non può svolgere altre attività per lei più specifiche
- DMA riduce l'intervento della CPU al minimo necessario

## Tre tecniche per l'input di un blocco di dati

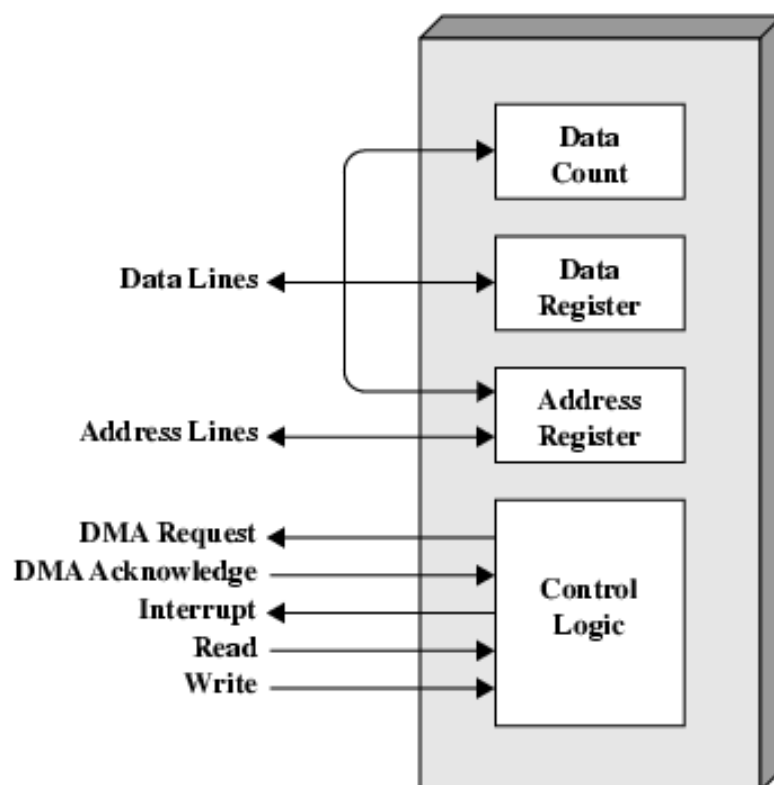


# Funzione del DMA



- Modulo (hardware) aggiuntivo, connesso al bus
- Il controllore DMA sostituisce la CPU per la maggior parte delle attività di I/O

## Diagramma di un tipico modulo DMA



# Operazioni DMA



- CPU comunica al controllore DMA:
  - lettura/scrittura
  - indirizzo dispositivo
  - indirizzo iniziale in memoria del blocco dati coinvolto nell'operazione (da dove leggere o dove scrivere i dati)
  - quantità di dati da trasferire
- CPU prosegue eseguendo altre attività
- Il controllore DMA si occupa del trasferimento dei dati (“colloquia” direttamente con la memoria centrale)
- Il controllore DMA invia un interrupt alla CPU quando ha terminato il trasferimento

## Trasferimento dati DMA



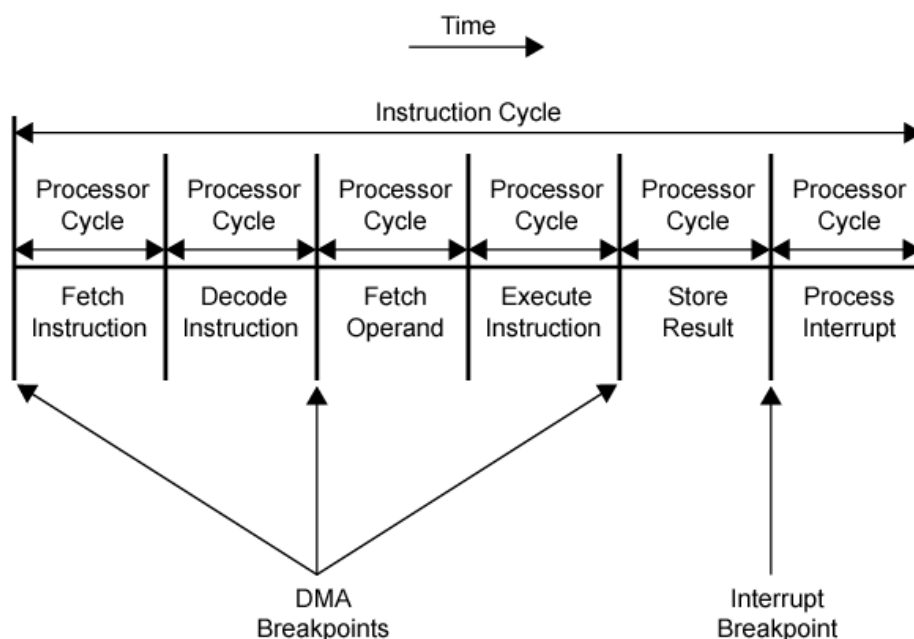
- Il *DMA controller* può accedere al canale dati in uno di due modi
  - **Una parola alla volta**, sottraendo di tanto in tanto alla CPU il controllo sul canale (*cycle stealing*)
  - **Per blocchi**, prendendo possesso del canale per una serie di trasferimenti (*burst mode*)
  - La CPU è bloccata in entrambi i casi, ma il *burst mode* è **più efficace** perché l'acquisizione del canale è onerosa

# Trasferimento dati DMA (cycle stealing)

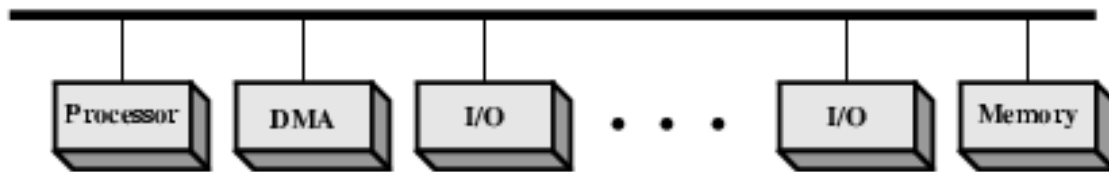


- Il controllore DMA prende possesso del bus per un ciclo
- Trasferisce una parola (word) di dati
- Non è una interruzione
  - La CPU non cambia contesto
- La CPU rimane “sospesa” proprio nel momento prima che acceda al bus
  - Ad esempio, prima del caricamento di un dato e/o operando o di una scrittura
- Rallenta la CPU ma non così tanto come nel caso in cui sia la CPU stessa ad occuparsi del trasferimento dati

## Breakpoint di DMA e di Interrupt durante un ciclo di istruzione

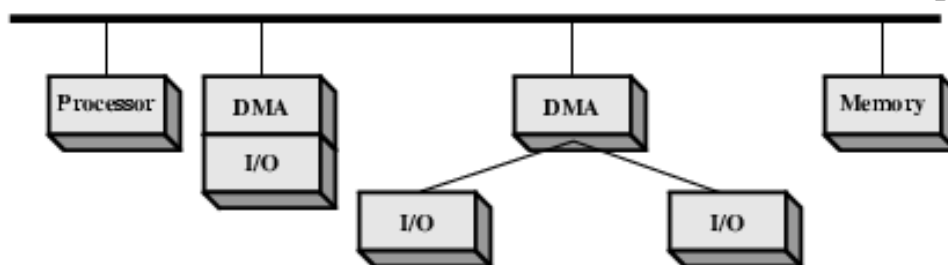


# Configurazioni DMA



- Bus singolo, controller DMA isolato
- Ogni trasferimento usa il bus due volte
  - da I/O a DMA e poi da DMA alla memoria
- CPU perde il possesso del bus due volte

# Configurazioni DMA

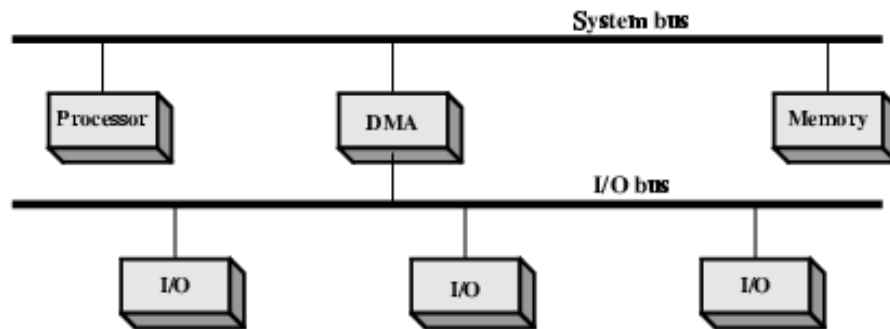


(b) Single-bus, Integrated DMA-I/O

- Bus singolo, controllore DMA integrato con I/O
- Può controllare più di un dispositivo
- Ogni trasferimento usa il bus una volta
  - da DMA a memoria
- CPU perde il controllo del bus una sola volta



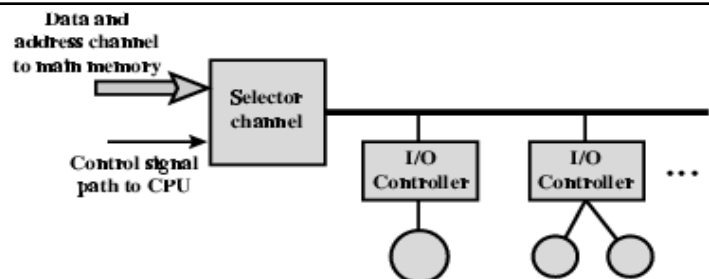
# Configurazioni DMA



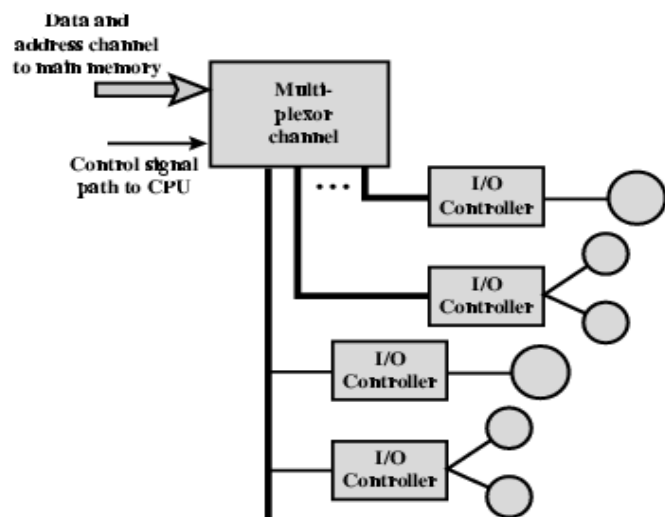
(c) I/O bus

- Bus di I/O separato
- DMA necessita di una sola interfaccia I/O
- Ogni trasferimento usa il bus di sistema una sola volta
  - da DMA a memoria
- CPU perde il controllo del bus una sola volta

## Canali I/O



(a) Selector



(b) Multiplexor