

# Lezione 8: Gestione dell'ingresso e uscita nell'architettura di LC-3

Elementi di Architettura e Sistemi Operativi

Corso di Laurea in Bioinformatica

Dicembre 2013

# Argomenti della lezione

- Come si opera con i registri dei dispositivi ?  
Con istruzioni per la memoria (mappati su memoria) o con istruzioni speciali.
- Come si gestisce la tempistica del trasferimento ?  
Trasferimento asincrono o sincrono.
- Chi controlla il trasferimento ?  
Il processore (interrogazione, polling) o il dispositivo (interruzione, interrupt).

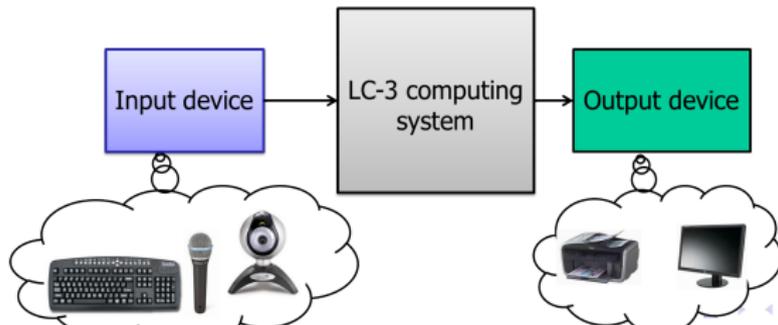
Fonte:

**Patt & Patel:** *“Introduction to Computing Systems: From Bits and Gates to C and Beyond”*. Ch. 8.

# Ingresso/uscita di LC-3

Tipi di dispositivi d'ingresso/uscita e loro caratteristiche:

- Comportamento: ingresso, uscita, memoria
  - ▶ ingresso: tastiera, microfono, videocamera,...etc
  - ▶ uscita: schermo, stampante, interfaccia di rete,..etc
  - ▶ memoria: disco rigido, dischetto CD-ROM
- Velocita' di trasferimento: quanto velocemente si possono trasferire i dati ?
  - ▶ tastiera: 100 bytes/sec
  - ▶ disco rigido: 30 MB/s
  - ▶ rete: 1 Mb/s - 1 Gb/s



# Ingresso/uscita di LC-3

Cercheremo di astrarre i dispositivi per gestirli con interfacce che sfruttino le caratteristiche condivise.

In questa lezione, prenderemo in considerazione la tastiera e lo schermo come esempi dei dispositivi piu' semplici.

Chiamate di sistema per I/U disponibili nell'architettura di LC-3:

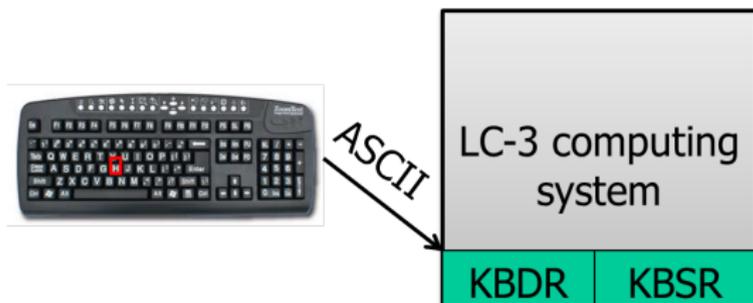
- Ingresso: TRAP x23, semantica: tastiera  $\rightarrow$  R0[7:0]
- Uscita: TRAP x21, semantica: R0[7:0]  $\rightarrow$  schermo

COME FUNZIONANO IN PRATICA?

# Dispositivo d'ingresso

Servono un registro dei dati e uno dello stato per gestire l'ingresso dalla tastiera

- KBDR - Memorizza il codice ASCII del carattere inserito dalla tastiera.
- KBSR - Segnala al processore che un nuovo carattere e' stato inserito dalla tastiera.

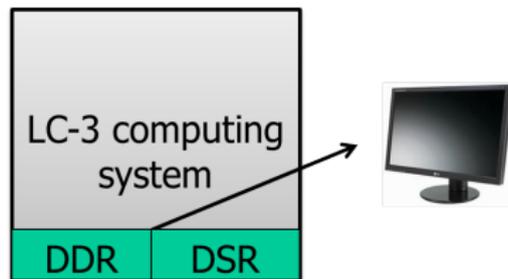


- KBDR: KeyBoard Data Register
- KBSR: KeyBoard Status Register

# Dispositivo d'uscita

Servono un registro dei dati e uno dello stato per gestire l'uscita da visualizzare sullo schermo.

- DDR - Memorizza il dato da visualizzare.
- DSR - Segnala allo schermo che c'e' un nuovo dato da visualizzare.



- DDR: Display Data Register
- DSR: Display Status Register

Il processore come accede ai registri dei dati ?

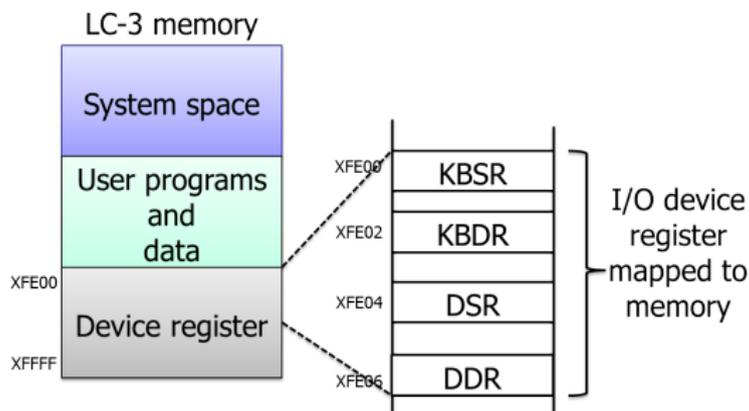
# Registri d'ingresso e uscita

Ci sono due modi perche' il processore possa accedere a tali registri:

- un modo di accedere a tali registri e' di definire nell'architettura delle istruzioni separate dedicate ai dispositivi d'ingresso-uscita.
- un altro modo di accedere a tali registri e' di usare le istruzioni **esistenti** nell'architettura di LC-3 per l'accesso a memoria.
  - ▶ KBDR & DDR devono apparire come locazioni di memoria per operare su di essi con istruzioni per la memoria.
  - ▶ Per esempio, le istruzioni LD/ST possono essere usate per leggere un valore dal registro KBDR o scriverlo nel registro DDR.
  - ▶ Si definiscono 4 nuovi registri di LC-3:
    - ★ KBDR & DDR per memorizzare i dati.
    - ★ KBSR & DSR per segnalarne la disponibilita'.
  - ▶ Inoltre questi registri sono mappati su un insieme d' *indirizzi* che sono riservati ai dispositivi I/U invece che alla memoria.

# Memoria mappata sull'ingresso/uscita

- Si assegna un indirizzo di memoria a ogni registro di dispositivo.
- Si usano delle istruzioni per trasferimento dati (LD/ST) per il controllo e trasferimento dati dei registri.



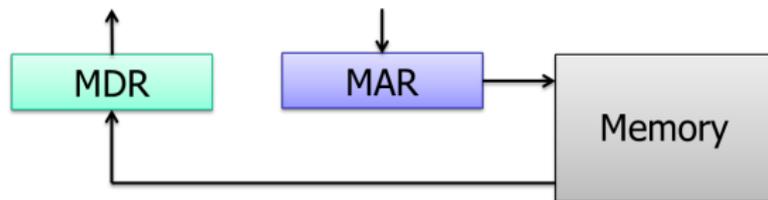
# Ingresso dalla tastiera

Accesso a memoria convenzionale: LD DR, ADDR

MAR  $\leftarrow$  ADDR

MDR  $\leftarrow$  MEM[MAR]

DR  $\leftarrow$  MDR



Accesso ai registri d'ingresso: LD DR0, xFE02 e LD DR1, xFE00

MAR  $\leftarrow$  xFE02 (indirizzo registro KBDR)

MDR  $\leftarrow$  KBDR (contenuto registro KBDR)

DR0  $\leftarrow$  MDR

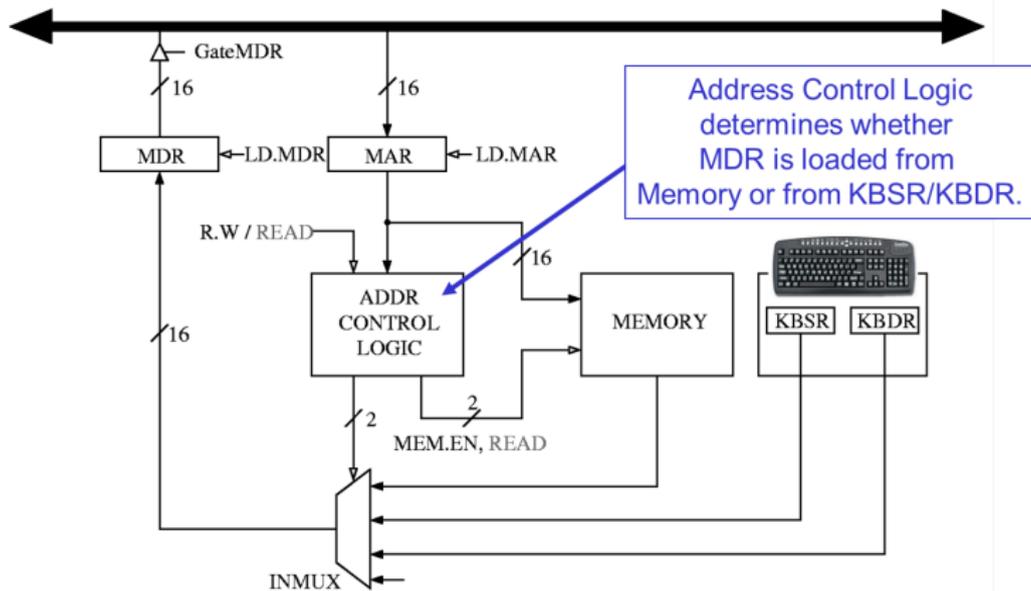
...

MAR  $\leftarrow$  xFE00 (indirizzo registro KBSR)

MDR  $\leftarrow$  KBSR (contenuto registro KBSR)

DR1  $\leftarrow$  MDR

# Ingresso dalla tastiera

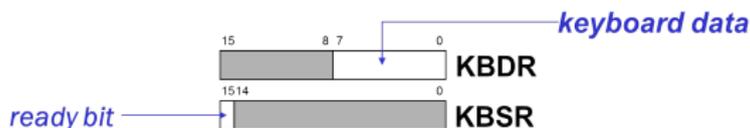


# Ingresso dalla tastiera

Si utilizzano i registri KBSR & KBDR.

Quando si dattilografa un carattere:

- Il suo codice ASCII e' memorizzato nelle posizioni [7:0] del registro KBDR (le posizioni [15:8] sono sempre a zero).
- Il "segnale di pronto"(KBSR[15]) e' posto a 1.
- La tastiera e' disattivata – un qualsiasi carattere dattiloscritto sara' ignorato.



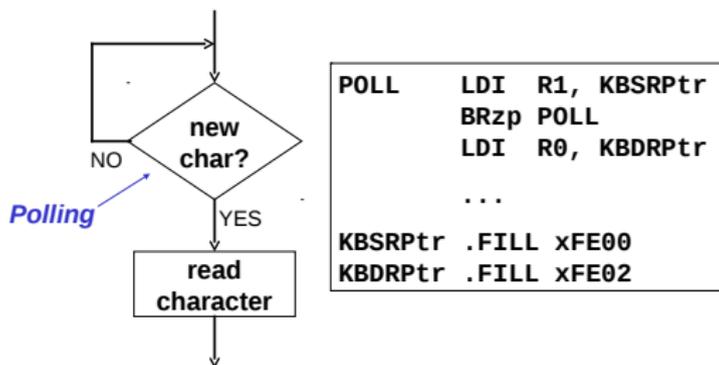
Quando il registro KBDR e' letto dal processore:

- KBSR[15] e' posto a 0.
- La tastiera e' abilitata per trasferire un nuovo carattere in KBDR.

# Ingresso dalla tastiera

Soluzione:

si usa il segnale di pronto del registro KBSR per verificare se da tastiera si e' memorizzato un nuovo dato nel registro KBDR.



# Ingresso dalla tastiera

```
POLL LDI R1, KBSR
```

```
...
```

```
KBSR .FILL xFE00
```

Si usa LDI (e non semplicemente LD) quando l'indirizzo dell'operando (che puo' trovarsi dappertutto in memoria) non e' ottenibile sommando il contatore di programma con il campo [8:0] dell'istruzione.

Nel nostro caso l'esecuzione di LDI prevede due fasi: una di accesso a memoria convenzionale, una di accesso al registro KBSR (poiche' xFE00 e' un indirizzo di registro e non di memoria).

Accesso a memoria convenzionale

MAR  $\leftarrow$  indirizzo locazione KBSR

MDR  $\leftarrow$  MEM[KBSR] = xFE00

Accesso al registro KBSR mappato sulla memoria

MAR  $\leftarrow$  MDR = xFE00

MDR  $\leftarrow$  contenuto registro KBSR

R1  $\leftarrow$  MDR

# Uscita sullo schermo

Accesso a memoria convenzionale: ST SR, ADDR

$MDR \leftarrow SR$

$MAR \leftarrow ADDR$

$MEM[MAR] \leftarrow MDR$



Accesso ai registri d'uscita: ST SR0, xFE06 e LD SR1, xFE04

$MDR \leftarrow SR0$

$MAR \leftarrow xFE06$  (indirizzo registro DDR)

$DDR \leftarrow MDR$  (contenuto registro DDR)

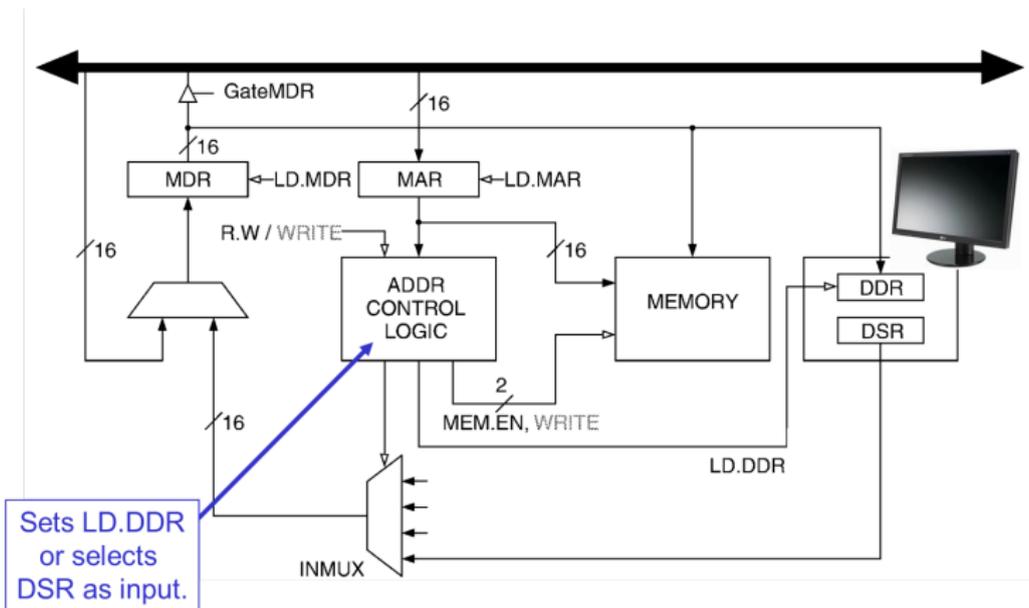
...

$MAR \leftarrow xFE04$  (indirizzo registro DSR)

$MDR \leftarrow DSR$  (contenuto registro DSR)

$DR1 \leftarrow MDR$

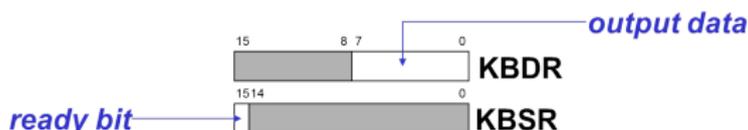
# Uscita sullo schermo



# Uscita sullo schermo

Quando lo schermo e' pronto a visualizzare un altro carattere:

- Il "segnale di pronto "(DSR[15]) e' posto a 1.

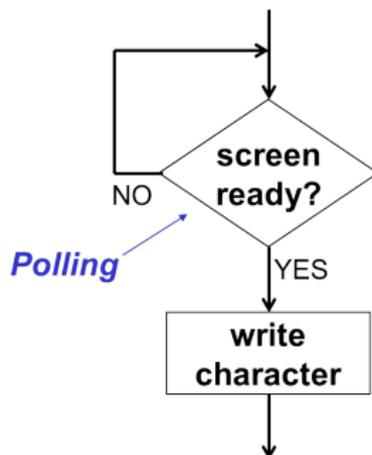


Quando un dato e' scritto nel registro DDR:

- DSR[15] e' posto a 0 (non si possono scrivere altri caratteri nel registro DDR per tutto il tempo in cui DSR[15] e' 0).
- Quando il carattere nel registro DDR[7:0] e' visualizzato sullo schermo l'elettronica dello schermo pone DSR[15] di nuovo a 1 (ed ora si puo' scrivere un altro carattere in DDR).

# Uscita sullo schermo

- Procedura d'uscita



*Polling*

```
POLL  LDI  R1, DSR
      BRzP POLL
      STI  R0, DDR
      ...
DSR   .FILL xFE04
DDR   .FILL xFE06
```

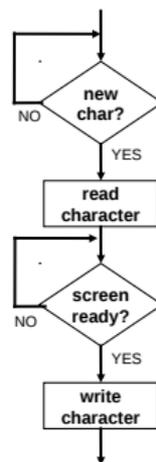
# Uscita sullo schermo

- Procedura di eco della tastiera (i caratteri in ingresso sono anche visualizzati sullo schermo per un controllo da parte dell'utente).

```
POLL1  LDI  R1, KBSRPtr
        BRzp POLL1
        LDI  R0, KBDRPtr
POLL2  LDI  R1, DSRPtr
        BRzp POLL2
        STI  R0, DDRPtr

        ...

KBSRPtr .FILL xFE00
KBDRPtr .FILL xFE02
DSRPtr  .FILL xFE04
DDRPtr  .FILL xFE06
```



# I/O gestita con interruzioni

Con un'interruzione, un dispositivo esterno puo':

- Fermare il programma attualmente in esecuzione.
- Interagire con il processore per compiere un'operazione d'ingresso o uscita.
- Riprendere il programma interrotto come se non fosse successo nulla.

Perche' ne vale la pena ?

- L'interrogazione (polling) potrebbe sprecare un sacco di cicli macchina, specialmente nel caso di eventi rari. Questi cicli potrebbero essere usati per altre elaborazioni del processore.
- Esempio: si eseguono operazioni sull'ingresso precedente mentre deve arrivare l'ingresso corrente (si veda l'Esempio 8.1 nel libro di testo).

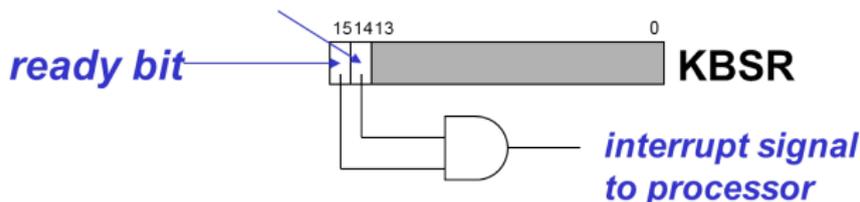
Per realizzare un meccanismo d'interruzione, serve:

- Un modo per il dispositivo d'I/O di segnalare al processore che e' successo un evento di cui occuparsi.
- Un modo per il processore di verificare se il segnale d'interruzione e' a 1 e se la sua priorita' e' piu' alta di quella del programma in esecuzione.

# I/O gestita con interruzioni

## Segnale di generazione

- Quando il processore vuol permettere al dispositivo d'I/O di richiedere un'interruzione, il processore pone a 1 il campo KBDR[14] che rappresenta il segnale di "abilitazione dell'interruzione" del dispositivo (IE, interrupt enable). Altrimenti, se KBDR[14] e' posto a 0, il processore non autorizza le interruzioni e quindi si dovra' ricorrere all'interrogazione (polling) per interagire con tale dispositivo.
- Quando i segnali di pronto KBDR[15] e di abilitazione dell'interruzione KBDR[14] sono a 1, si segnala che c'e' un'interruzione (interrupt signal).

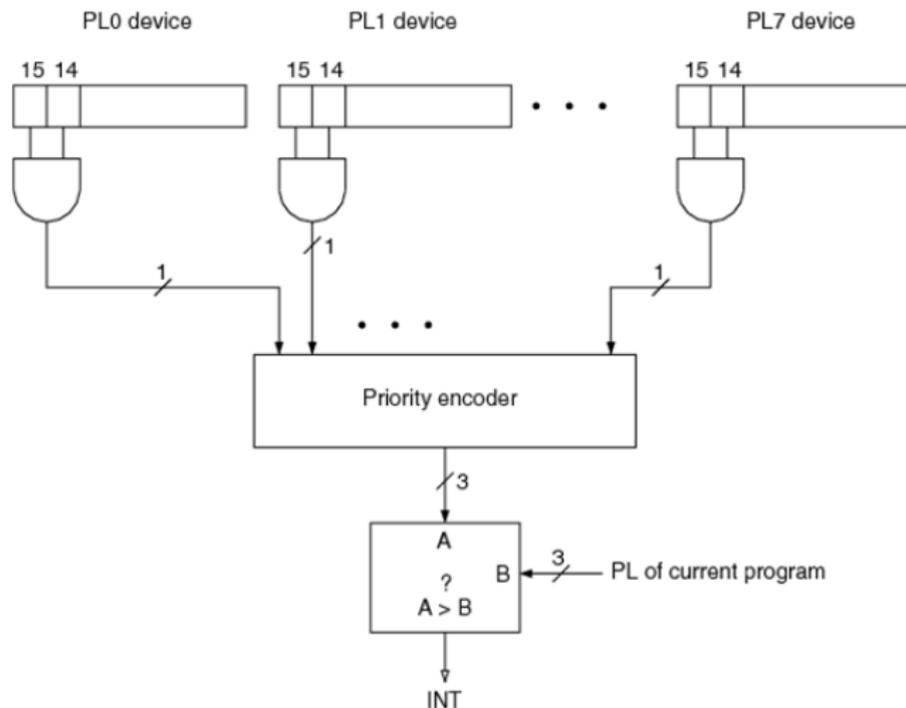


Ogni istruzione esegue a un certo livello di priorita'.  
Nell'architettura LC-3 ci sono 8 livelli di priorita' (PL0-PL7).

- Consideriamo il seguente esempio:
  - ▶ C'e' un programma per le buste paga che esegue al livello PL0.
  - ▶ C'e' un programma di verifica dei guasti di un reattore nucleare che esegue al livello PL6.
- E' consentito ad una richiesta a livello PL6 d'interrompere un'esecuzione a livello PL0, ma non viceversa.

Il codificatore di priorita' seleziona il dispositivo a piu' alta priorita', lo confronta con il livello di priorita' del processo corrente, e genera un segnale d'interruzione se ne e' il caso.

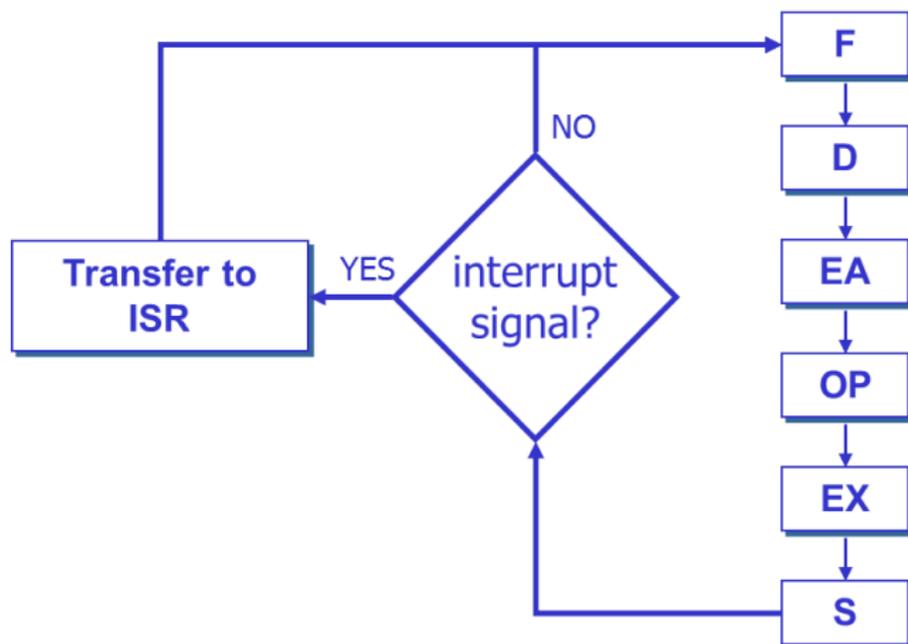
# Priorita'



# Verifica dell'arrivo del segnale d'interruzione

Un processore esegue in continuazione il ciclo di prelievo ed esecuzione di un'istruzione articolato nelle sue varie fasi. Per gestire le interruzioni, il processore verifica se c'è il segnale d'interruzione tra le fasi di scrittura (STORE RESULT) e prelievo (FETCH), cioè prima d'incominciare una nuova istruzione. Se il segnale d'interruzione non è a 1, il processore continua con l'istruzione successiva. Se il segnale d'interruzione è a 1, il processore trasferisce il controllo alla procedura che gestisce l'interruzione, salvando prima i dati dell'esecuzione corrente e poi caricando il contatore di programma con l'inizio della procedura d'interruzione.

# Verifica dell'arrivo del segnale d'interruzione





# Realizzazione completa dell'I/U mappato su memoria in LC-3

- MIO.EN: indica che un trasferimento dati in ingresso o uscita deve avvenire in questo ciclo (asserito dalla macchina a stati finiti dell'unita' di controllo).

# Accesso diretto tra unita' I/U e memoria (DMA)

- S'introduce un dispositivo specializzato a trasferire dati tra memoria e un'unita' d'ingresso/uscita (es. disco fisso).
- Il processore scrive l'indirizzo iniziale e la dimensione della memoria da copiare.
- Il dispositivo specializzato esegue i trasferimenti in sottofondo, accedendo alla memoria solo quando non serve al processore (si dice che ruba cicli).

# Concetti imparati in questa lezione

- Come si opera con i registri dei dispositivi.  
Con istruzioni per la memoria (dispositivi d'ingresso e uscita mappati su memoria) o con istruzioni speciali.
- Come si gestisce la tempistica del trasferimento.  
Trasferimento asincrono o sincrono.
- Chi controlla il trasferimento.  
Il processore (interrogazione, polling) o il dispositivo (interruzione, interrupt).