



# FONDAMENTI DI INFORMATICA

## Lezione n. 12

- **STACK, SOTTOPROGRAMMI, INTERRUZIONI**
- **GESTIONE DELLE INTERRUZIONI**
- **SET DI ISTRUZIONI**

**In questa lezione approfondiremo le caratteristiche interne delle architetture con attenzione alle chiamate a sottoprogramma.**

**Illustreremo le ragioni e i meccanismi legati alla presenza delle interruzioni.**

**Descriveremo poi le caratteristiche principali del set di istruzione di una architettura di una CPU di tipo generale.**


## SOTTOPROGRAMMI

**Un sottoprogramma è una sequenza di istruzioni che viene usata più volte nel corso di un programma.**

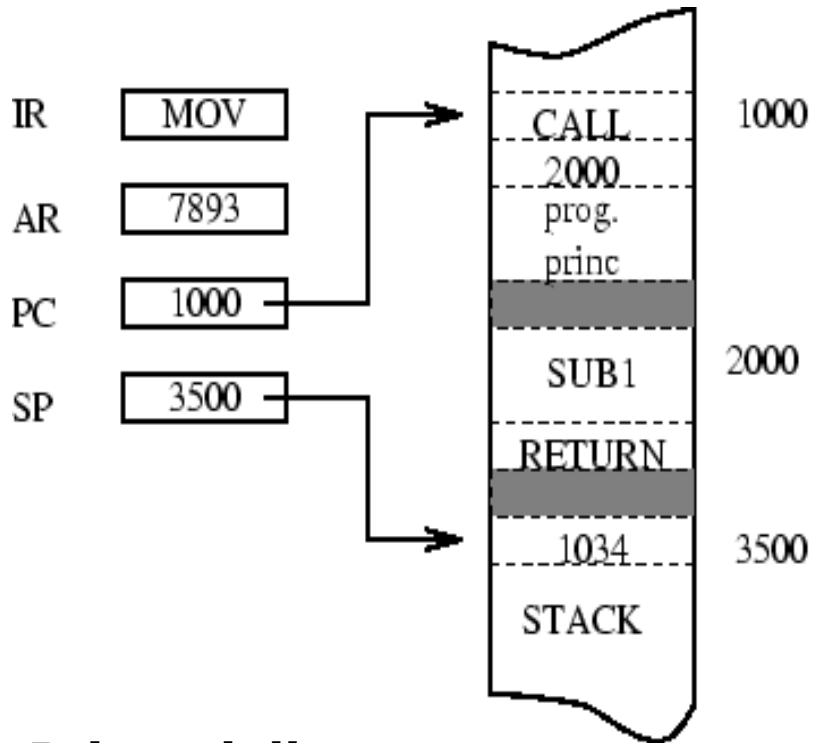
```
...  
    CALL SUB1  
NEXT1: ...  
...  
    CALL SUB1  
...  
  
...  
SUB1: ...  
...  
    RETURN
```

**Programma principale**

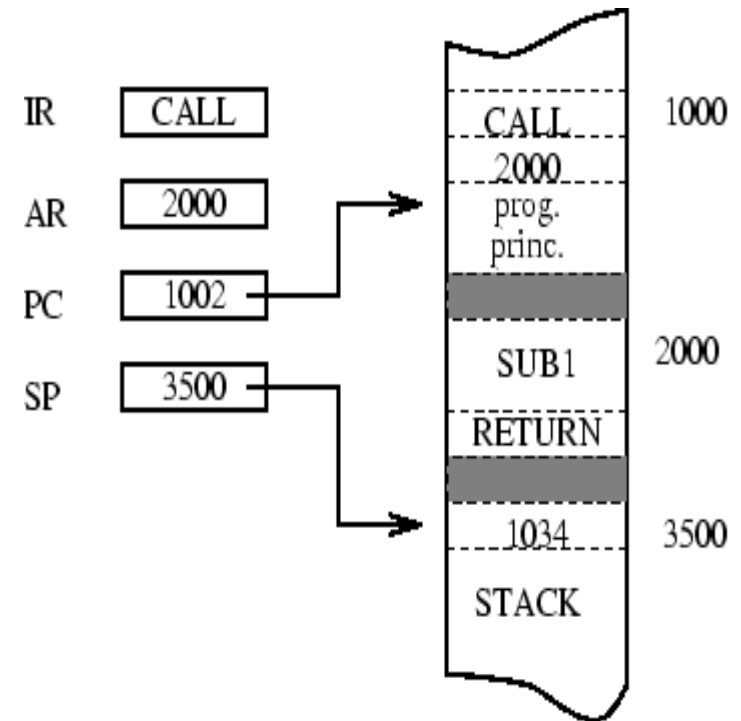
**Sottoprogramma**



## SOTTOPROGRAMMI

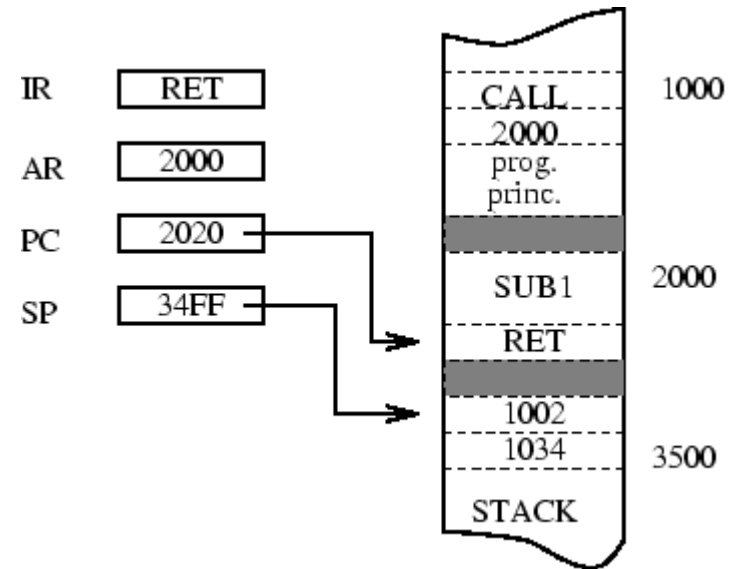
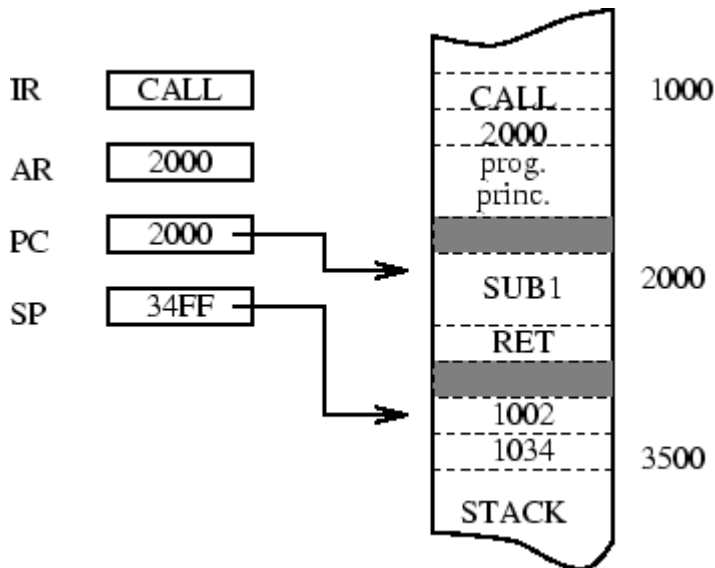


**Prima della  
istruzione di CALL**



**Esecuzione  
istruzione di CALL  
(fase di fetch)**

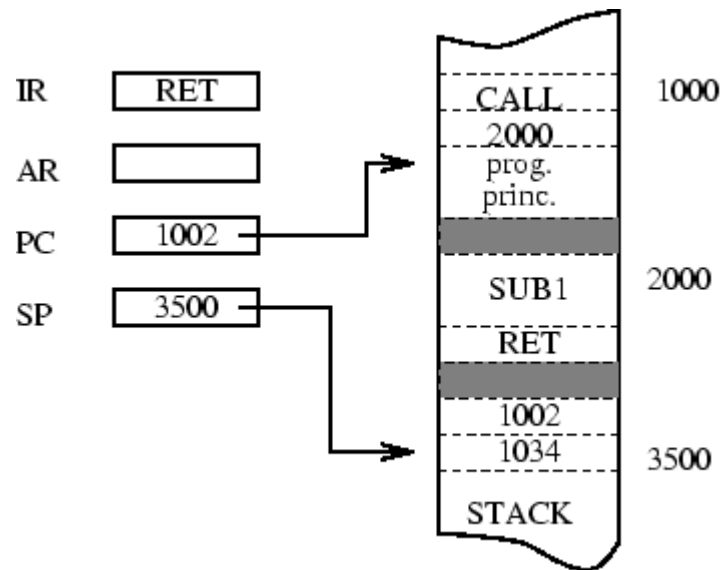
## SOTTOPROGRAMMI



**Esecuzione  
istruzione di CALL**

**L'istruzione di RETURN:  
riporta nel PC il valore  
puntato da SP e  
incrementa SP.  
RETURN  $\Rightarrow$  MOV PC,(SP)+**

## SOTTOPROGRAMMI



**Il sottoprogramma  
è stato eseguito e il  
programma procede  
ad eseguire  
l'istruzione dopo la  
CALL**

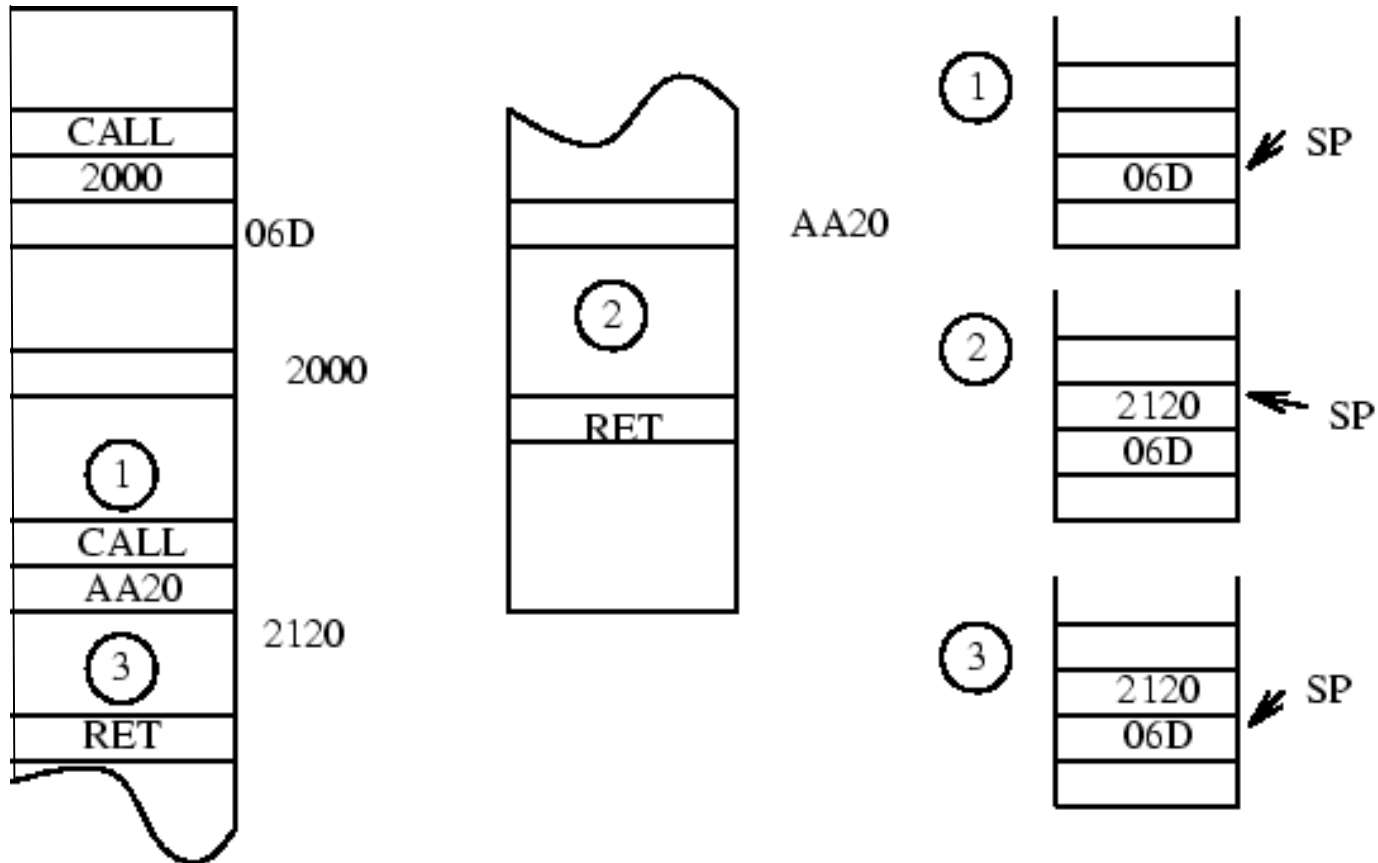


## ANNIDAMENTI

- **Questa tecnica permette l'annidamento o nesting dei sottoprogrammi.**
- **Annidamento: chiamata a sottoprogramma all'interno di un sottoprogramma.**
- **Stack interno o esterno (memoria principale).**



## ANNIDAMENTI



# INTERRUPT

- **Evento infrequente ed eccezionale.**
- **Generato internamente o esternamente.**
- **Causa il trasferimento del controllo dal programma corrente a un programma specifico di servizio dell'evento.**

**L'interruzione è utilizzata per la gestione degli apparati di I/O. Per ottenere risposte rapide nella gestione di dischi, tastiere (keyboard), modem, ...**

**Questi elementi devono richiamare l'attenzione della CPU molto rapidamente e il trasferimento dei dati da e verso gli questi organi deve avvenire nel modo più efficiente possibile.**

**Un ulteriore aspetto è la necessità di assegnare urgenze o priorità diverse alle varie richieste di intervento in caso di contemporaneità.**



## GESTIONE INTERRUPT

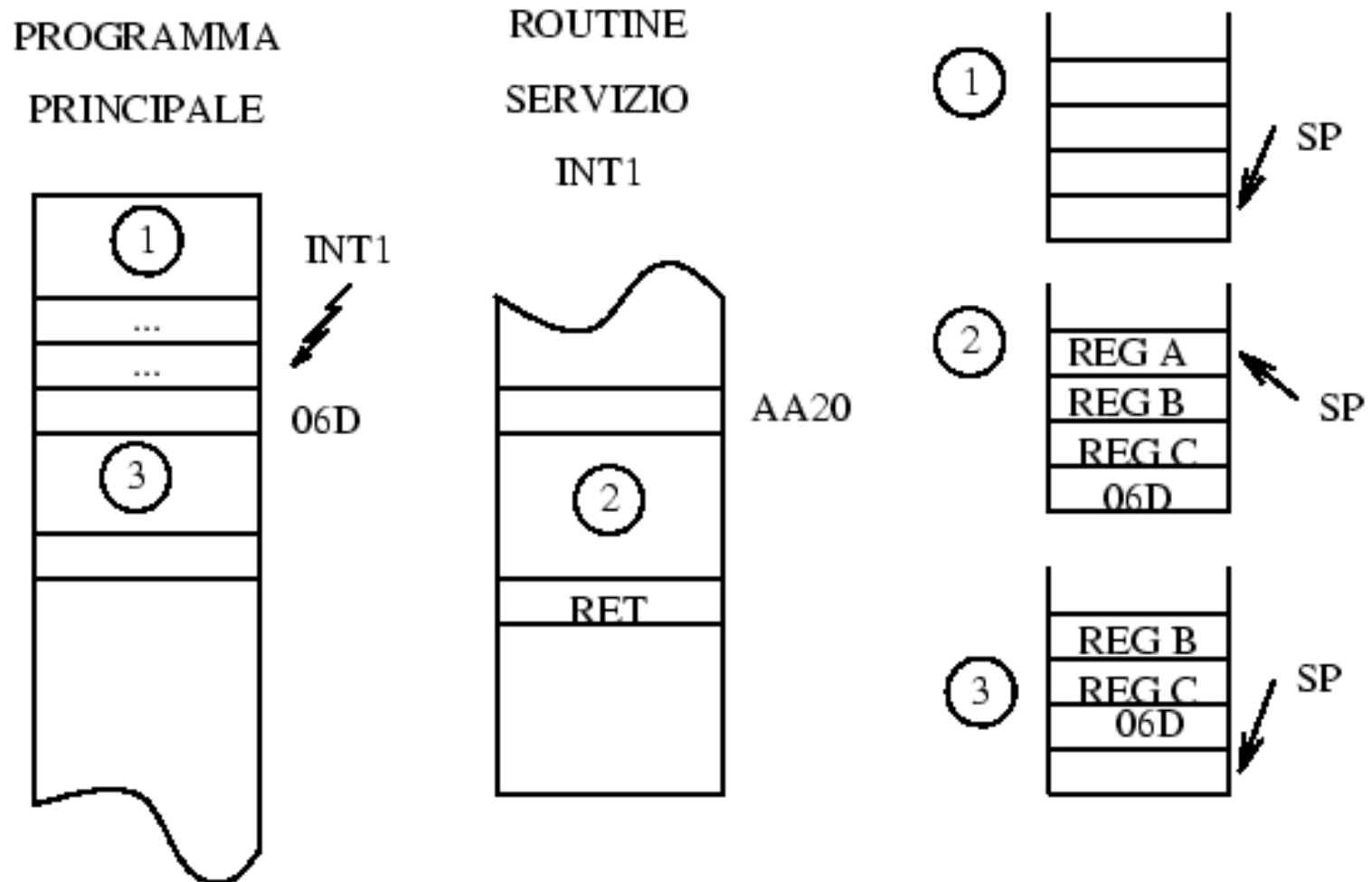
- **La presenza di un interrupt è segnalata alla CPU da una linea proveniente dall'esterno (o da un segnale interno).**
- **Il segnale viene memorizzato e testato dalla CPU alla fine di ogni ciclo di istruzione.**
- **La CPU risponde trasferendo il controllo a un altro programma.**
- **L'evento che causa l'interruzione è asincrono rispetto all'esecuzione del programma.**
- **Trasparenza delle interruzioni.**

## GESTIONE INTERRUPT

### **Azioni della CPU in risposta alla richiesta di interruzione (continua):**

- **Lo stato complessivo della CPU viene memorizzato a cura del programma di gestione.**
- **Nel PC viene immagazzinato l'indirizzo del sottoprogramma di gestione.**
- **L'esecuzione del sottoprogramma continua fino all'istruzione di RETURN che riporta il controllo al programma interrotto.**

## GESTIONE INTERRUPT



# CARATTERISTICHE DI UN SET DI ISTRUZIONI

- **COMPLETEZZA:**

**Deve essere possibile valutare qualunque funzione che sia calcolabile con una disponibilità ragionevole di memoria.**

- **EFFICIENZA:**

- **Istruzioni usate frequentemente devono essere eseguite rapidamente.**
- **Il controllo sull'efficienza deve essere misurato sul codice generato da compilatori.**
- **Fornire primitive, non soluzioni.**

# CARATTERISTICHE DI UN SET DI ISTRUZIONI

- **REGOLARITA':**

- **Le istruzioni devono comportarsi in modo omogeneo rispetto ai modi di indirizzamento.**
- **Quando ogni istruzione consente tutti i modi di indirizzamento possibili su tutti gli operandi:**

## **ORTOGONALITA'**

- **COMPATIBILITA'**

- **Compatibilità sorgente: Sono compatibili i codici mnemonici del linguaggio assembler.**
- **Compatibilità binaria: Sono compatibili i codici macchina.**
- **Il codice macchina deve essere eseguibile su processori precedenti della stessa famiglia.**
- **8080  $\Rightarrow$  8086  $\Rightarrow$  80186  $\Rightarrow$  ...  $\Rightarrow$  Pentium  $\Rightarrow$  ...**
- **PDP11  $\Rightarrow$  VAX**

# CARATTERISTICHE DI UN SET DI ISTRUZIONI

## ESEMPIO PDP11

**Istruzioni a un operando:**



**Istruzioni a due operandi:**



## MODI DI INDIRIZZAMENTO

**Operando (sorgente o destinazione):**

**3 bit modo:**

**1 bit per ind. indiretto**

**2-3 bit (registro, autoinc., autodec., con indice)**

**3 bit per selezionare 1 su 8 registri.**



## TIPI DI ISTRUZIONE

### TRASFERIMENTO DATI

<b>MOVE</b>	<b>Trasferisce dati da sorgente a destinazione</b>
<b>LOAD</b>	<b>Trasf. dati dalla memoria</b>
<b>STORE</b>	<b>Trasf. dati nella memoria</b>
<b>EXCHANGE</b>	<b>Scambia dati</b>
<b>SET/RESET</b>	<b>Pone ad I/O</b>
<b>PUSH/POP</b>	<b>Gestione stack</b>

### OPERAZIONI ARITMETICHE

<b>ADD/SUB</b>	<b>Somma / differenza</b>
<b>MULT/DIV</b>	<b>Moltiplicazione/divisione</b>
<b>ABS</b>	<b>Valore assoluto</b>
<b>NEG</b>	<b>Cambio segno</b>
<b>INC/DEC</b>	<b>Incrementa/decrementa</b>



## TIPI DI ISTRUZIONE

### OPERAZIONI ARITMETICHE

#### Scalari

#### Vettoriale

	Virg.fissa		Virg.mobile	
	+, -	x, :	+, -, x, :	
Intel 8085	x			
Motor. 68020	x	x		
IBM 360	x	x	x	x
Cray-1	x	x	x	x
Pentium	x	x	x	(x)

**Le attuali CPU VLSI svolgono attualmente le stesse operazioni che svolgevano i supercalcolatori di una decina di anni fa.**



## TIPI DI ISTRUZIONE

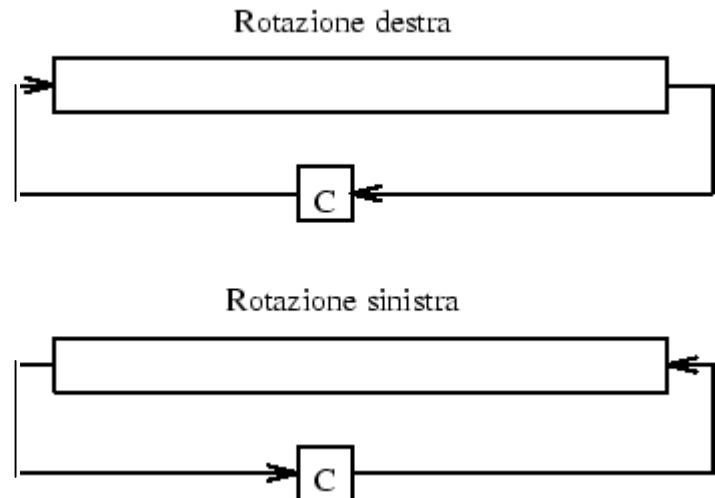
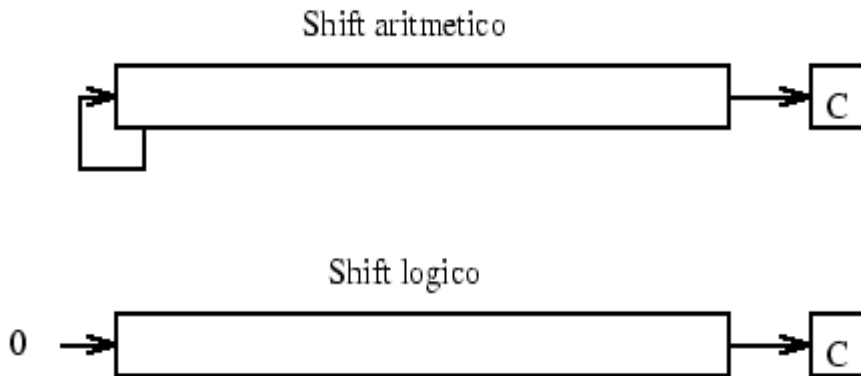
### LOGICHE

**AND/OR/NOT/EX-OR**  
**CONVERT**

**Calcola le funzioni logiche**  
**Cambia formato ai dati**

### SHIFT

### ROTATE





## TIPI DI ISTRUZIONE

### CONTROLLO PROGRAMMA

<b>JUMP</b>	<b>Salto incondizionato</b>
<b>JUMP COND</b>	<b>Salto condizionato</b>
<b>CALL (COND)</b>	<b>Salto a sottoprogramma</b>
<b>RET (COND)</b>	<b>Uscita da sottoprogramma</b>

### CONTROLLO CPU

<b>HALT</b>	<b>Blocco operazioni</b>
<b>WAIT / HOLD</b>	<b>Blocco operazioni con ripresa condizionata o su interrupt.</b>
<b>NOP</b>	<b>Non svolge operazioni.</b>



# TIPI DI ISTRUZIONE

## INGRESSO E USCITA

<b>INPUT (READ)</b>	<b>Trasferimento dati da I/O a memoria o registro</b>
<b>OUTPUT (WRITE)</b>	<b>Trasferimento dati verso porta di I/O.</b>