

Livello di Linguaggio Macchina

❖ **ISA** (Instruction Set Architecture)

❖ **Tipi di istruzioni**

ISA

- 1. Tipi di dati**
 - a. Rappresentazione dei dati
 - b. Precisione dei dati
 - c. Riferimento ai dati
- 2. Insiemi di registri**
 - a. Registri dedicati
 - b. Registri operazionali
- 3. Insiemi di istruzioni**
 - a. Modalità di indirizzamento
 - b. Tipi di istruzioni**

3.b Tipi di istruzioni

- Classificazione dei tipi di istruzioni basata sul tipo di dati

Istruzioni su dati non tipizzati

(determinano operazioni eseguibili su tutti i tipi di dati)

1. Istr. di trasferimento dati
2. Istr. di I/O
3. Istr. di traslazione e rotazione

Istruzioni su dati tipizzati

(operano su un tipo di dati)

1. Istr. aritmetiche
2. Istr. logiche
3. Istr. di trasferimento di controllo
4. Istr. di elaborazione stringhe
5. Istr. di elaborazione indirizzi

Istruzioni di Trasferimento Dati

- Istruzioni di trasferimento di dati da una locazione (sorgente) ad un'altra (destinazione).

❖ Istruzioni tipiche in linguaggio macchina

- **Move**: trasferimento da sorgente a destinazione
- **Load**: trasferimento da locazione di memoria a registro
- **Store**: trasferimento da registro a locazione di memoria
- **Exchange**: scambio dei contenuti di sorgente e destinazione
- **Clear**: azzeramento del contenuto della destinazione
- **Push**: trasferimento da sorgente a stack
- **Pop**: trasferimento da stack a destinazione

Istruzioni del Pentium II

MOV dst, src	copia src (source) in dst (destination)
PUSH src	pone src nello stack
POP dst	estrae da stack dst
PUSHA	pone tutti i registri nello stack
LEA dst, src	carica lo spiazzamento di src in dst

Istruzioni di Traslazione e Rotazione

- Istruzioni logiche monadiche.
 - Istruzioni che hanno un operando.
- ❖ Istruzioni tipiche in linguaggio macchina.
 - **Shift** (Traslazione): trasferimento di ogni bit di una word nella posizione successiva (a destra o a sinistra), con perdita dei bit "espulsi" dalla fine della parola.
 - **Logical Shift**: inserimento di zeri nelle posizioni vuote.
 - **Arithmetic Shift**: shift a destra con replica del bit del segno o shift a sinistra con esclusione del bit del segno.
 - **Rotate** (Rotazione): shift a destra o a sinistra con reinserimento del bit espulso.

□ Istruzioni del Pentium II

SHL/SHR dst, # shift logico di dst a sx/dx di # bit
SAL/SAR dst, # shift aritmetico di dst a sx/dx di # bit
ROL/ROR dst, # rotazione di dst a sx/dx di # bit

Istruzioni di Traslazione e Rotazione



Istruzioni di I/O

- ❑ Istruzioni legate alla modalità di indirizzamento dei dispositivi di I/O adottata nel sistema.
 - ❖ **Memory-Mapped I/O**
 - Lo spazio di indirizzamento è condiviso dalla memoria e dai dispositivi di I/O ai quali è riservato un sottoinsieme degli indirizzi di memoria (non esistono istruzioni di I/O).
 - ❖ **Istruzioni di I/O**
 - Istruzioni speciali per i trasferimenti di I/O che consentono l'accesso ai registri dati dei dispositivi periferici.

- ❑ Istruzioni del Pentium II
- ❖ **IN** trasferimento da PORTA a locazione
- ❖ **OUT** trasferimento da locazione a PORTA

Istruzioni Aritmetiche

- ❑ Istruzioni diadiche
 - Istruzioni che hanno due operandi.
 - ❖ Istruzioni aritmetiche per i diversi tipi di dati numerici:
 - Istruzioni aritmetiche su numeri interi (a virgola fissa).
 - Istruzioni aritmetiche su numeri floating point.
 - Istruzioni aritmetiche su numeri interi decimali (BCD).
 - ❖ Istruzioni aritmetiche tipiche in linguaggio macchina
 - **Add**
 - **Subtract**
 - **Multiple**
 - **Divide**
 - **Absolute**
 - **Increment**
 - **Decrement**

Istruzioni Aritmetiche

- ❖ Tutte le operazioni aritmetiche sono eseguite su numeri a precisione finita e predefinita.
 - La proprietà di chiusura non è soddisfatta rispetto a nessuna delle quattro operazioni aritmetiche.
 - Una operazione viola la proprietà di chiusura quando il suo risultato:
 - è maggiore del valore massimo dell'insieme dei valori rappresentabili (errore di **Overflow**) o minore del valore minimo (errore di **Underflow**);
 - non appartiene all'insieme dei valori rappresentabili.
- ❖ Il risultato di un'operazione aritmetica può influenzare i flag del registro di stato del processore.

Istruzioni Aritmetiche

Status Word Register



- ❖ N (Negativo) 1 se il risultato è negativo (altrimenti zero)
- ❖ Z (Zero) 1 se il risultato è zero "
- ❖ D (Decimale) 1 per operazioni in BCD "
- ❖ V (oVerflow) 1 se si verifica un overflow "
- ❖ C (Carry) 1 se il risultato genera un riporto



Tale flag rende possibile operazioni su operandi più lunghi della parola del processore → Aritmetica a precisione multipla

Istruzioni Aritmetiche

□ Principali istruzioni aritmetiche del Pentium II

- ADD dst, src somma src a dst
 - ADDC dst, src somma src a dst con carry
 - SUB dst, src sottrae dst da src
 - MUL src moltiplica EAX per src (unsigned)
 - IMUL src moltiplica EAX per src (signed)
 - DIV src divide EAX per src (unsigned)
 - IDIV divide EAX per src (signed)
 - INC dst somma 1 a dst
 - DEC dst sottrae 1 a dst
- ❖ EAX registro di 32 bit utilizzato per operazioni aritmetiche.
- Per la moltiplicazione e la divisione, un prodotto o un dividendo da 64 bit viene memorizzato in EAX (parte meno significativa) e in EDX (parte più significativa).

Istruzioni Logiche

□ Istruzioni diadiche che effettuano le operazioni logiche su tutti i bit di dati logici.

- Operano su **vettori booleani**, aggregati (word o byte) di variabili logiche generalmente non omogenee.
 - Operano in parallelo su tutti gli elementi del vettore booleano ("bit a bit") e consentono la manipolazione di singoli bit di una parola o di altre unità indirizzabili.
- ❖ Istruzioni logiche tipiche in linguaggio macchina
- **AND**
 - **OR**
 - **NOT**
 - **XOR**
 - **Test**
 - **Compare**
 - **Set control variable**

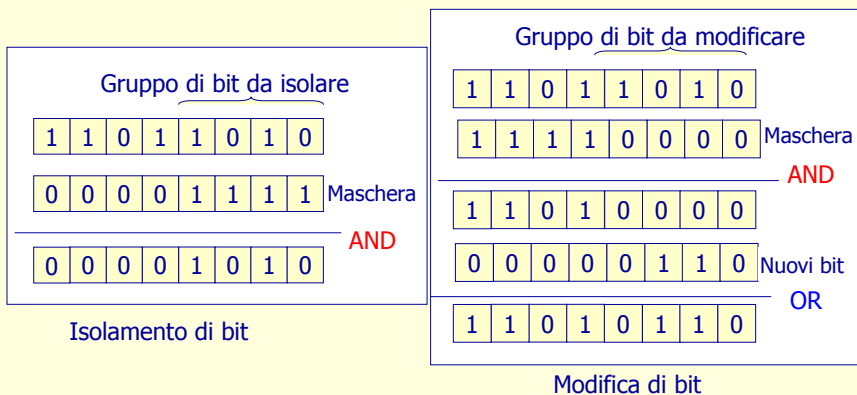
Istruzioni Logiche

- ❖ Test
 - Verifica la condizione specificata e imposta i flag in base al risultato.
- ❖ Compare
 - Esegue un confronto logico o aritmetico su due o più operandi e imposta i flag in base al risultato.

Istruzioni del Pentium II	0 0 1 0 1 1 0 0	Operando 1
	1 0 1 0 0 1 1 0	Operando 2
NOT dst	0 0 1 0 0 1 0 0	AND
AND dst, src	1 1 0 1 0 0 1 1	NOT
OR dst, src	1 0 1 0 1 1 1 0	OR

Istruzioni Logiche

- ❖ Isolamento e modifica di bit tramite AND e OR



Istruzioni di Elaborazione Stringhe

- Istruzioni che consentono di effettuare operazioni sulle stringhe.
 - ❖ Stringa: insieme di caratteri.
 - Necessità di rendere sequenziali le operazioni su stringhe
 - ❖ Due tipi di istruzioni:
 1. **Istruzioni per elaborare un elemento della stringa**
 - Istruzioni orientate al byte.
 2. **Istruzioni per elaborare tutta la stringa**
 - Istruzioni tradotte dall'interprete in una sequenza di passi.

Per elaborare una stringa occorrono **puntatore alla stringa, contatore, istruzioni di ripetizione.**

Istruzioni di Elaborazione Stringhe

- Istruzioni del Pentium II
 - ❖ **MOVS** dst, src
 - Trasferimento di un elemento della stringa
 - SI : registro indice per stringa src, inizializzato con l'indirizzo del primo elemento della stringa
 - DI : registro indice per stringa dst, inizializzato con l'indirizzo del primo elemento della stringa
 - CX: registro contatore, inizializzato con il numero degli elementi della stringa
 - ❖ **REP MOVS** dst, src
 - Trasferisce gli elementi di src a dst fino a che CX è diverso da zero.
 - Decremento automatico di CX

Istruzioni di Elaborazione Indirizzi

- Istruzioni che elaborano indirizzi.
 - ❖ Indirizzo: dato "intero binario senza segno".
 - Un indirizzo può essere contenuto in:
 - Registri specifici (es. Stack pointer, registri indice)
 - Locazioni di memoria
 - Registri concatenabili
 - ❖ Istruzioni che accedono all'indirizzo.
 - L'indirizzo è prima calcolato secondo le modalità descritte nel campo operandi e poi trasferito.

LEA dst, src

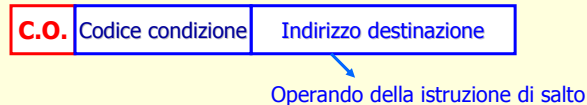
Istruzioni di Trasferimento di Controllo

- Istruzioni per modificare la sequenza di esecuzione delle istruzioni di un programma, normalmente controllata dal meccanismo automatico di avanzamento del P.C.
 - ❖ L'esecuzione di tali istruzioni determina l'aggiornamento del P.C. con l'indirizzo di un'altra istruzione.
 - ❖ Tipi di istruzioni di controllo del flusso di esecuzione.
 1. Istruzioni di Salto
 - Salto condizionato
 - Salto non condizionato
 2. Istruzioni di Iterazione
 3. Istruzioni di Chiamata di Procedura

Istruzioni di Trasferimento di Controllo

❑ Istruzioni di Salto

- ❖ L'indirizzo di destinazione può essere specificato mediante:
 - **Indirizzamento assoluto**: l'indirizzo è nella istruzione.
 - **Indirizzamento a registro indiretto**: l'indirizzo è nel registro specificato nell'istruzione.
 - **Indirizzamento relativo**: la distanza (displacement) tra destinazione ed istruzione di salto è nella istruzione (istruzioni di Branch).
 - Spostamento rispetto al contenuto del contatore di programma (codifica con pochi bit).



Istruzioni di Trasferimento di Controllo

Jump

Salto incondizionato

- ❖ Effettua il salto all'indirizzo di destinazione senza condizioni.

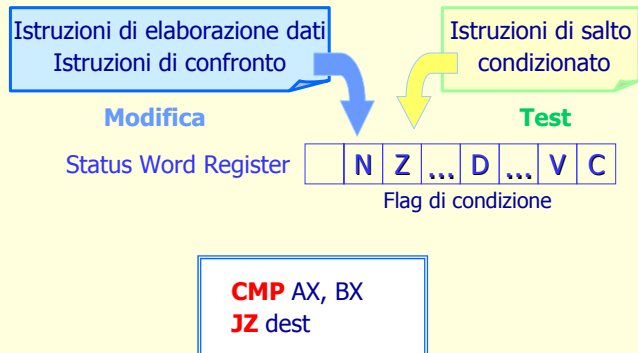
Jcond

Salto condizionato

- ❖ Controlla una condizione ed effettua il salto all'indirizzo di destinazione solo se la condizione è verificata.
 - Il controllo della condizione è effettuato testando l'informazione presente in bit singoli (codici di condizione dello Status Word Register).

Istruzioni di Trasferimento di Controllo

- Istruzioni di salto condizionato



Istruzioni di Trasferimento di Controllo

- **Istruzioni di Iterazione**
 - ❖ Istruzioni che facilitano la esecuzione di un gruppo di istruzioni per un certo numero di volte.
 - Le istruzioni di iterazione (ciclo) comprendono:
 - **Istruzione di aggiornamento** di una variabile (contatore).
 - **Istruzione di controllo** di una condizione.
 - Controllo a fine ciclo.
 - Controllo ad inizio ciclo.
 - **Istruzione di salto.**
 - ❖ Istruzioni tipiche in linguaggio macchina
 - **LOOP**
 - **LOOPE**
 - **LOOPNE**

Istruzioni di Trasferimento di Controllo

- ❖ Ricerca del primo elemento del vettore "Array" diverso da zero

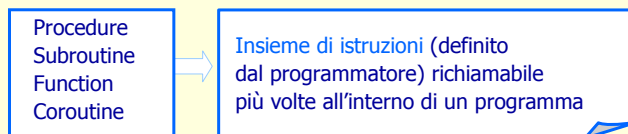
```
MOV CX, LENGTH ARRAY
MOV SI, -1
NEXT: INC SI
      CMP ARRAY[SI], 0
      LOOPZ NEXT
```

- LOOP : ciclo per CX diverso da zero
- LOOPZ: ciclo per CX diverso da zero e Z=1

Istruzioni di Trasferimento di Controllo

□ Istruzioni di Chiamata di Procedura

- ❖ Un programma può essere organizzato in moduli, detti "procedure".



- ❖ Una procedura è un programma indipendente che può essere chiamato da un altro programma.
 - Una procedura permette l'uso della stessa porzione di codice più volte.
- ❖ Una istruzione di chiamata a procedura può essere vista come una istruzione macchina complessa.

Istruzioni di Trasferimento di Controllo

❖ Il meccanismo della chiamata di procedura coinvolge due istruzioni basilari:

1. Istruzione di chiamata di procedura (**Call**)
 - Istruzione di salto, il cui operando è l'indirizzo della prima istruzione della procedura chiamata, detto "entry-point address".
2. Istruzione di ritorno da procedura (**Return**)
 - Istruzione di salto il cui operando è l'indirizzo della istruzione che segue la chiamata, detto "return address".

Necessità di "conservare il return address"

Istruzioni di Trasferimento di Controllo

❖ Modi di memorizzazione del return address

1. **Registro o locazione di memoria**
 - Impossibilità di chiamare un'altra procedura all'interno di quella chiamata.
2. **Locazione di memoria associata alla procedura chiamata**
 - L'istruzione di return è una istruzione di salto alla locazione di memoria che conserva l'address della istruzione della procedura chiamante da eseguire.
 - È possibile l' **annidamento (nesting)** delle procedure.
3. **Stack**
 - È possibile sia l' annidamento delle procedure che la **ricorsività** (abilità di una procedura a chiamare se stessa).
 - Soluzione adottata in quasi tutte le architetture Isa.

CALL dest

$SP = (SP) - 1$

$(SP) = (PC)$

$PC = \text{dest}$

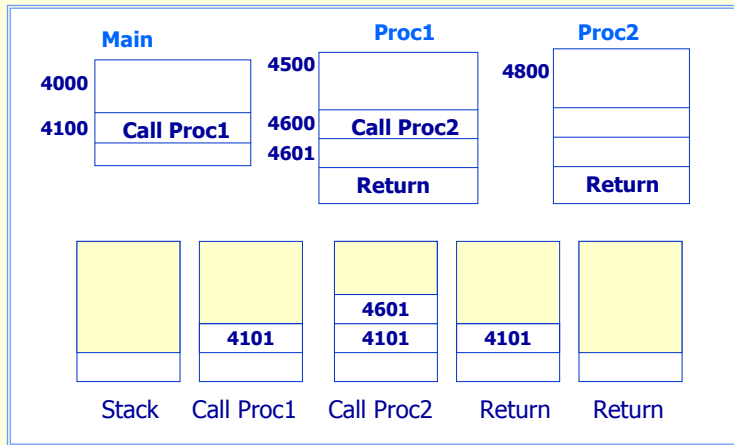
RETURN

$PC = ((SP))$

$SP = SP + 1$

Istruzioni di Trasferimento di Controllo

❖ Procedure annidate



Istruzioni di Trasferimento di Controllo

- ❖ Il salvataggio del contenuto del PC ed il suo ripristino è **implicitamente** effettuato dalle istruzioni di chiamata e di ritorno da procedura.
- ❖ Il salvataggio e ripristino del contenuto dei registri usati dalla procedura (stato del processore) deve essere **esplicitamente** formulato.
 - Istruzioni specifiche per salvare e ripristinare il contenuto dei registri usati dalla procedura.
 - Le istruzioni di salvataggio dei registri sono utili non solo per le chiamate a procedura ma anche per le routine di servizio agli interrupt.

Istruzioni di Trasferimento di Controllo

❖ Passaggio di parametri tra procedure

- La procedura chiamante può passare alla procedura chiamata dei valori detti parametri o argomenti.
 - Nella CALL si devono fornire gli "operandi", o i loro indirizzi, da usare nella elaborazione.
 - **Parametri formali o virtuali:** parametri presenti nella lista della procedura all'atto della sua definizione.
- La procedura chiamata fornisce dei parametri (risultati) alla procedura chiamante.
 - Il passaggio dei parametri può avvenire per valore o indirizzo (registri, locazioni di memoria, stack).
 - **Parametri effettivi o reali:** parametri presenti nella lista della procedura all'atto della sua chiamata.

Istruzioni di Trasferimento di Controllo

❖ Metodi di passaggio di parametri

1. La procedura chiamante pone i parametri in una "parameter list" e trasmette l'address della lista alla procedura chiamata.
2. La procedura chiamante pone gli address dei parametri in una "parameter-address list " e trasmette l'address della lista alla procedura chiamata.
3. La procedura chiamante pone i parametri, prima di effettuare la chiamata, nello stack.
4. La procedura chiamante pone gli address dei parametri nello stack.

I parametri effettivi (reali) devono essere in corrispondenza esatta con quelli formali (formali).