

Modi di indirizzamento

Capitolo 11

Vari modi di specificare l'indirizzo degli operandi

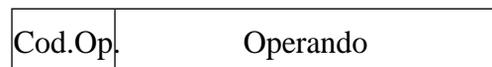
- Immediato
- Diretto
- Indiretto
- Registro
- Registro indiretto
- Spiazzamento
- Pila

Indirizzamento immediato

- L'operando è parte dell'istruzione (campo indirizzo)
- Vantaggio: nessun accesso in M per prendere l'operando
- Svantaggio: valore limitato dalla dimensione del campo indirizzo

Indirizzamento immediato

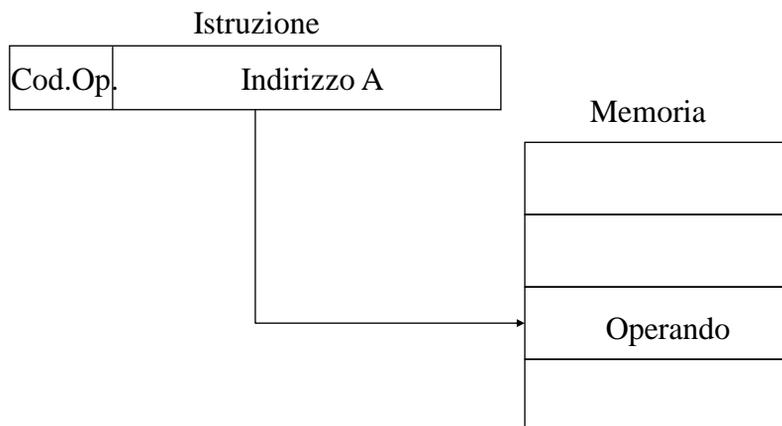
Istruzione



Indirizzamento diretto

- Campo indirizzo = indirizzo dell'operando
- Esempio: ADD A
 - Somma il contenuto della cella A all'accumulatore
 - Bisogna andare in M all'indirizzo A per trovare l'operando
- Un singolo accesso in M per prendere l'operando
- Spazio di indirizzamento limitato

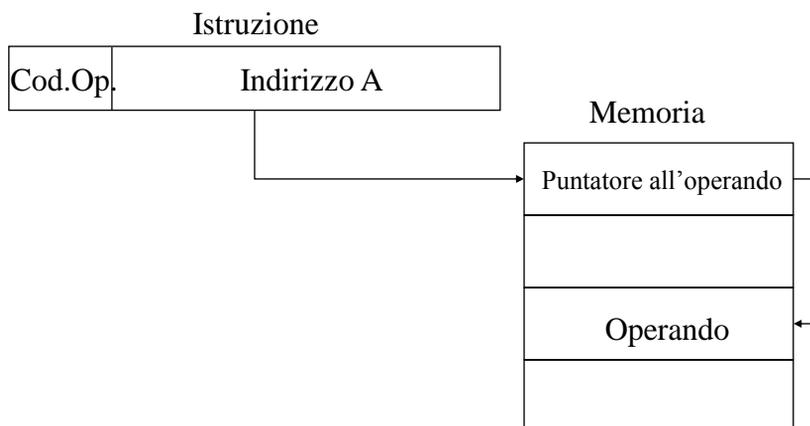
Indirizzamento diretto



Indirizzamento indiretto

- Il campo indirizzo contiene l'indirizzo di una cella di M, che contiene l'indirizzo dell'operando
- Vantaggio: parole di lunghezza N permettono di indirizzare 2^N entità diverse
 - In realtà 2^K , dove K è la lunghezza del campo indirizzo
- Svantaggio: due accessi in M per ottenere l'operando
- Esempio: ADD A
 - Somma il contenuto della cella puntata dal contenuto di A all'accumulatore

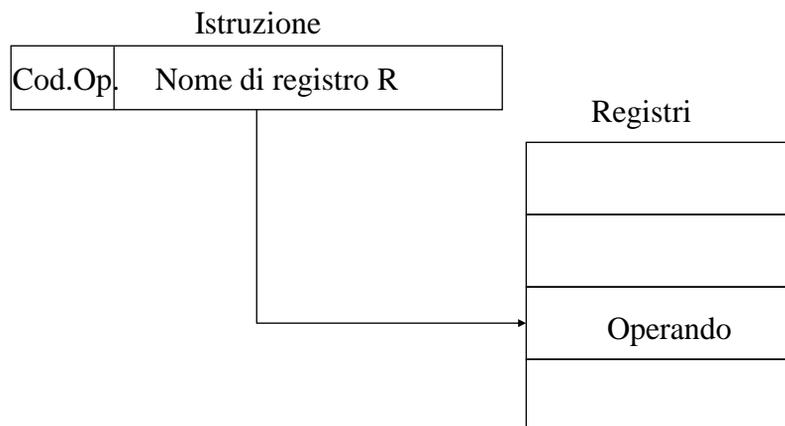
Indirizzamento indiretto



Indirizzamento registro

- L'operando è in un registro indicato nel campo indirizzo
- Numero limitato di registri
- Pochi bit necessari per il campo indirizzo
 - Istruzioni più corte
 - Fase di fetch più veloce (nessun accesso in M per prendere l'operando)

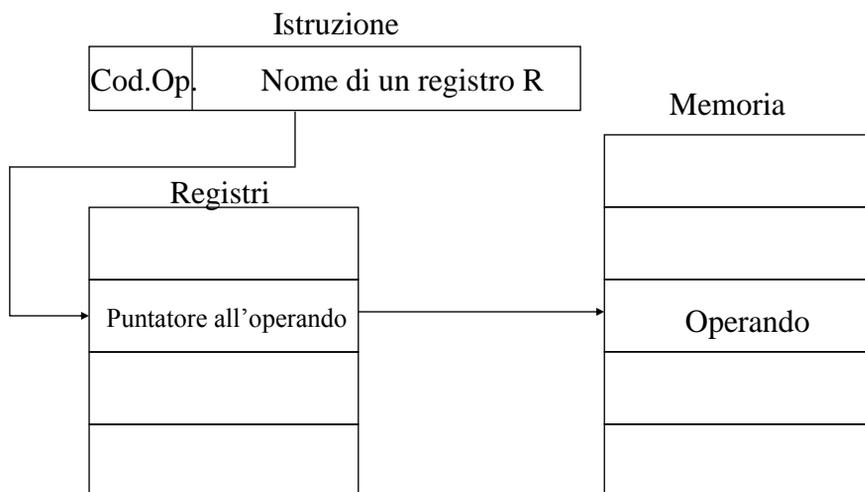
Indirizzamento registro



Indirizzamento registro indiretto

- Stesso principio dell'indirizzamento indiretto
- L'operando è in una cella di M puntata dal contenuto del registro R
- Grande spazio di indirizzamento (2^n)
- Un accesso in meno in M rispetto all'indirizzamento indiretto

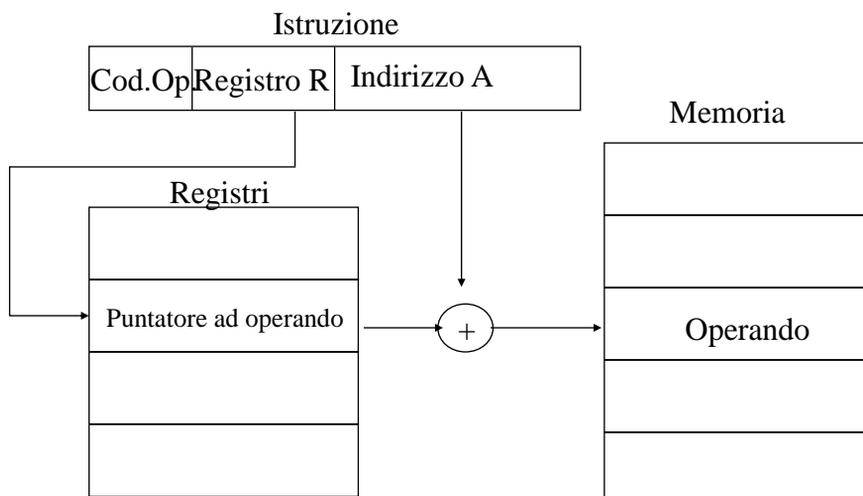
Indirizzamento registro indiretto



Indirizzamento con spiazzamento

- Combinazione di indirizzamento diretto e indirizzamento registro indiretto
- Il campo indirizzo ha due sottocampi
 - A = valore di base (diretto)
 - R = registro che contiene l'indirizzo di un valore da sommare ad A per ottenere l'indirizzo
 - o viceversa (R base e A spiazzamento)

Indirizzamento con spiazzamento



Indirizzamento relativo

- Una versione dell'indirizzamento con spiazzamento
- R = registro PC (program counter)
- Indirizzo dell'operando = $A + (PC)$
 - A celle dalla cella di M puntata da PC

Indirizzamento registro-base

- A contiene lo spiazzamento
- R contiene il puntatore all'indirizzo base
 - R può essere esplicito o implicito

Indicizzazione

- A = base
- R = spiazzamento
- Esempio: elenco di dati memorizzati in M a partire da un indirizzo A
 - Per accedere a tutti, la sequenza di indirizzi è $A, A+1, A+2, \dots$
 - A viene messo nel campo indirizzo e il registro (indice) contiene 0 all'inizio e viene incrementato di 1 dopo ogni accesso

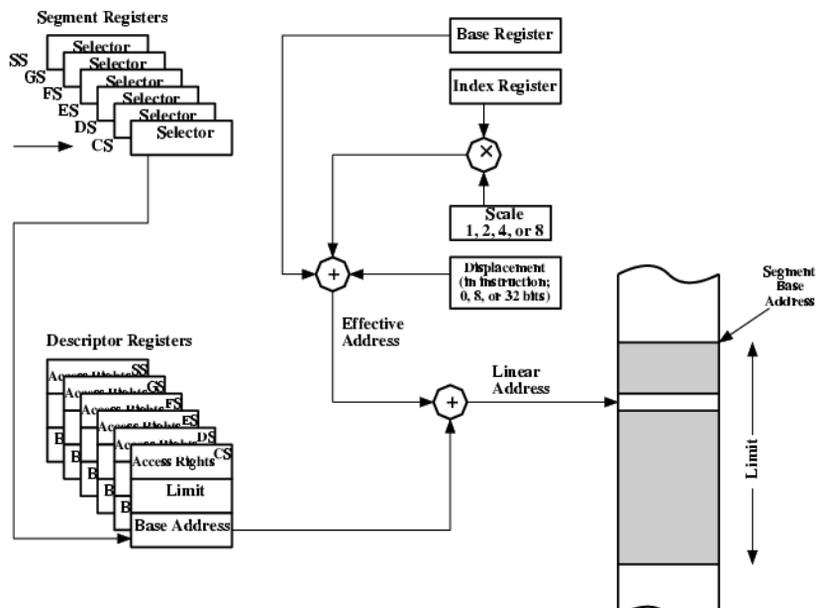
Indirizzamento a pila

- Pila: sequenza lineare di locazioni riservate di M
- Puntatore (nel registro SP , stack pointer) con l'indirizzo della cima della pila
- L'operando è sulla cima della pila
- Quindi è un esempio di indirizzamento a registro indiretto

Modi di indirizzamento del Pentium

- 12 modi di indirizzamento
 - Immediato
 - Registro
 - Spiazzamento
 - Base
 - Base con spiazzamento
 - Indice scalato con spiazzamento
 - Indice scalato con base e spiazzamento
 - Relativo

Modi di indirizzamento del Pentium



Formato delle istruzioni

- Struttura dei campi dell'istruzione
- Include il codice operativo
- Include (in modo implicito o esplicito) uno o più operandi
- Di solito più di un formato per un linguaggio macchina

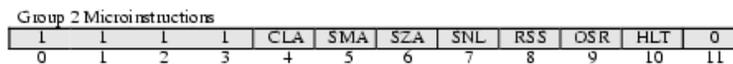
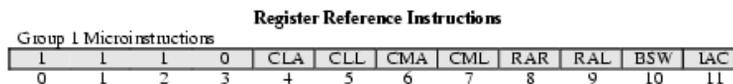
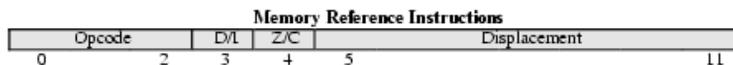
Lunghezza delle istruzioni

- Condiziona ed è condizionata da:
 - Dimensione della M
 - Organizzazione della M
 - Struttura del bus
 - Complessità della CPU
 - Velocità della CPU
- Compromesso tra repertorio di istruzioni potente e necessità di risparmiare spazio

Allocazione dei bit

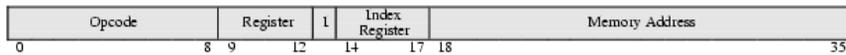
- Vari modi di indirizzamento
- Vari numeri di operandi (di solito 1 o 2)
- Registri verso M (di solito almeno 32 registri)
- Banchi di registri (esempio: Pentium ha due banchi)
 - Due banchi da 8 registri ciascuno → solo 3 bit per indicare un registro (il codice operativo indica il banco)
- Intervallo di indirizzi
- Granularità degli indirizzi (es.: byte o parola)
 - L'indirizzamento di byte richiede più bit ma è utile (es. per manipolare caratteri)

Formato delle istruzioni del PDP-8



- | | |
|---|---|
| D/I = Direct/Indirect address
Z/C = Page 0 or Current page
CLA = Clear Accumulator
CLL = Clear Link
CMA = CoMplement Accumulator
CML = CoMplement Link
RAR = Rotate Accumulator Right
RAL = Rotate Accumulator Left
BSW = Byte SWap | LAC = Increment ACCumulator
SMA = Skip on Minus Accumulator
SZA = Skip on Zero Accumulator
SNL = Skip on Nonzero Link
RSS = Reverse Skip Sense
OSR = Or with Switch Register
HLT = HaLT
MQA = Multiplier Quotient into Accumulator
MQL = Multiplier Quotient Load |
|---|---|

Formato delle istruzioni del PDP-10

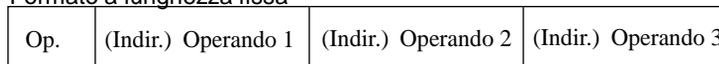


I = indirect bit

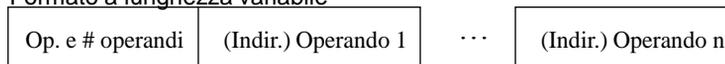
Formato a lunghezza variabile

- Fino ad ora abbiamo visto esempi di formati a lunghezza fissa
- Alcune architetture utilizzano formati a lunghezza variabile o ibrida

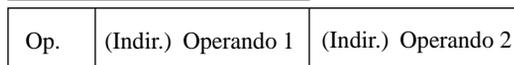
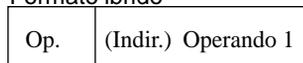
Formato a lunghezza fissa



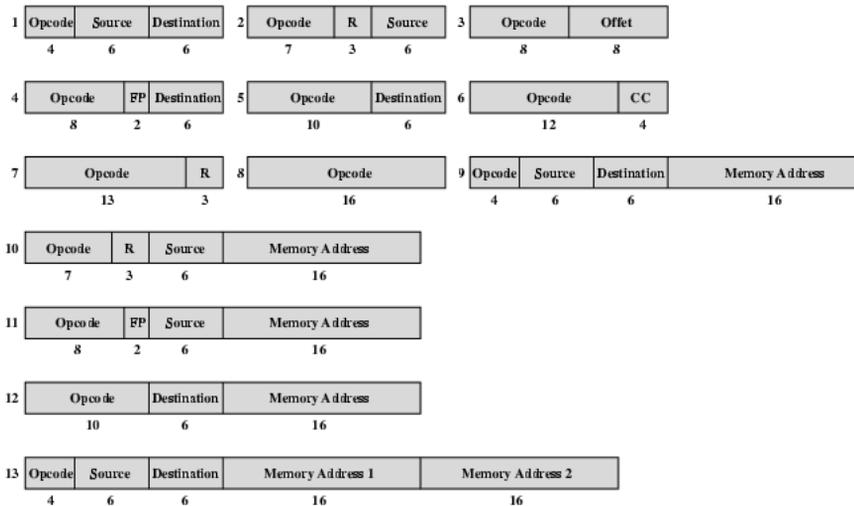
Formato a lunghezza variabile



Formato ibrido



Formato delle istruzioni del PDP-11

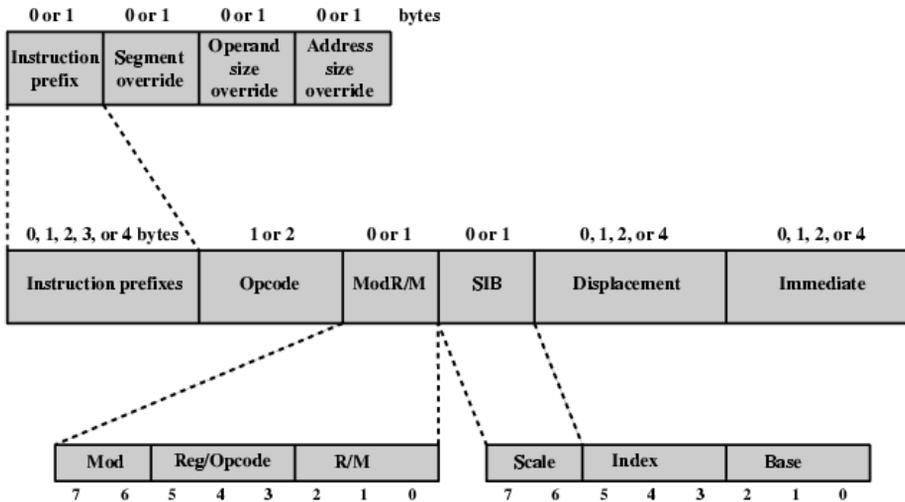


Numbers below fields indicate bit length
 Source and Destination each contain a 3-bit addressing mode field and a 3-bit register number
 FP indicates one of four floating-point registers
 R indicates one of the general-purpose registers
 CC is the condition code field

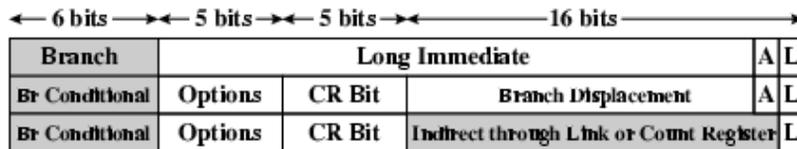
Formato delle istruzioni del VAX

Hexadecimal Format	Explanation	Assembler Notation and Description												
<div style="border: 1px solid black; padding: 2px; display: inline-block;"> ← 8 bits → <table border="1" style="border-collapse: collapse;"> <tr><td>0</td><td>5</td></tr> </table> </div>	0	5	Opcode for RSB	RSB Return from subroutine										
0	5													
<table border="1" style="border-collapse: collapse;"> <tr><td>D</td><td>4</td></tr> <tr><td>5</td><td>9</td></tr> </table>	D	4	5	9	Opcode for CLR.L Register R9	CLR.L R9 Clear register R9								
D	4													
5	9													
<table border="1" style="border-collapse: collapse;"> <tr><td>B</td><td>0</td></tr> <tr><td>C</td><td>4</td></tr> <tr><td>6</td><td>4</td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>A</td><td>B</td></tr> <tr><td>1</td><td>9</td></tr> </table>	B	0	C	4	6	4	0	1	A	B	1	9	Opcode for MOV.W Word displacement mode, Register R4 356 in hexadecimal Byte displacement mode, Register R11 25 in hexadecimal	MOV.W 356(R4), 25(R11) Move a word from address that is 356 plus contents of R4 to address that is 25 plus contents of R11
B	0													
C	4													
6	4													
0	1													
A	B													
1	9													
<table border="1" style="border-collapse: collapse;"> <tr><td>C</td><td>1</td></tr> <tr><td>0</td><td>5</td></tr> <tr><td>5</td><td>0</td></tr> <tr><td>4</td><td>2</td></tr> <tr><td>D</td><td>F</td></tr> </table>	C	1	0	5	5	0	4	2	D	F	Opcode for ADD.L3 Short literal 5 Register mode R0 Index prefix R2 Indirect word relative (displacement from PC)	ADD.L3 #5, R0, @A[R2] Add 5 to a 32-bit integer in R0 and store the result in location whose address is sum of A and 4 times the contents of R2		
C	1													
0	5													
5	0													
4	2													
D	F													
<table border="1" style="border-collapse: collapse;"> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> </table>							Amount of displacement from PC relative to location A							

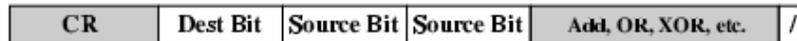
Formato delle istruzioni del Pentium



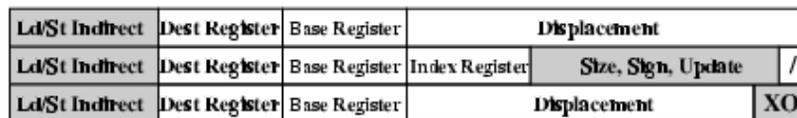
Formato delle istruzioni del PowerPC (1)



(a) Branch instructions



(b) Condition register logical instructions



(c) Load/store instructions

Formato delle istruzioni del PowerPC (2)

Ld/St Indirect	Dest Register	Base Register	Displacement		
Ld/St Indirect	Dest Register	Base Register	Index Register	Size, Sign, Update	/
Ld/St Indirect	Dest Register	Base Register	Displacement		
					XO *

(c) Load/store instructions

Arithmetic	Dest Register	Src Register	Src Register	O	Add, Sub, etc.	R
Add, Sub, etc.	Dest Register	Src Register	Signed Immediate Value			
Logical	Src Register	Dest Register	Src Register	ADD, OR, XOR, etc.		R
AND, OR, etc.	Src Register	Dest Register	Unsigned Immediate Value			
Rotate	Src Register	Dest Register	Shift Amt	Mask Begin	Mask End	R
Rotate or Shift	Src Register	Dest Register	Src Register	Shift Type or Mask		R
Rotate	Src Register	Dest Register	Shift Amt	Mask	XO	S R *
Rotate	Src Register	Dest Register	Src Register	Mask	XO	R *
Shift	Src Register	Dest Register	Shift Type or Mask			S R *

(d) Integer arithmetic, logical, and shift/rotate instructions

Flt sgl/dbl	Dest Register	Src Register	Src Register	Src Register	Fodd, etc.	R
-------------	---------------	--------------	--------------	--------------	------------	---

(e) Floating-point arithmetic instructions

A = Absolute or PC relative **XO** = Opcode extension
L = Link or subroutine **S** = Part of shift amount field
O = Record overflow in XER * = 64-bit implementation only
R = Record condition in CR1