



SAPIENZA
UNIVERSITÀ DI ROMA

MODI INDIRIZZAMENTO

Dott. Franco Liberati

Argomenti

01

Modi di indirizzamento

02

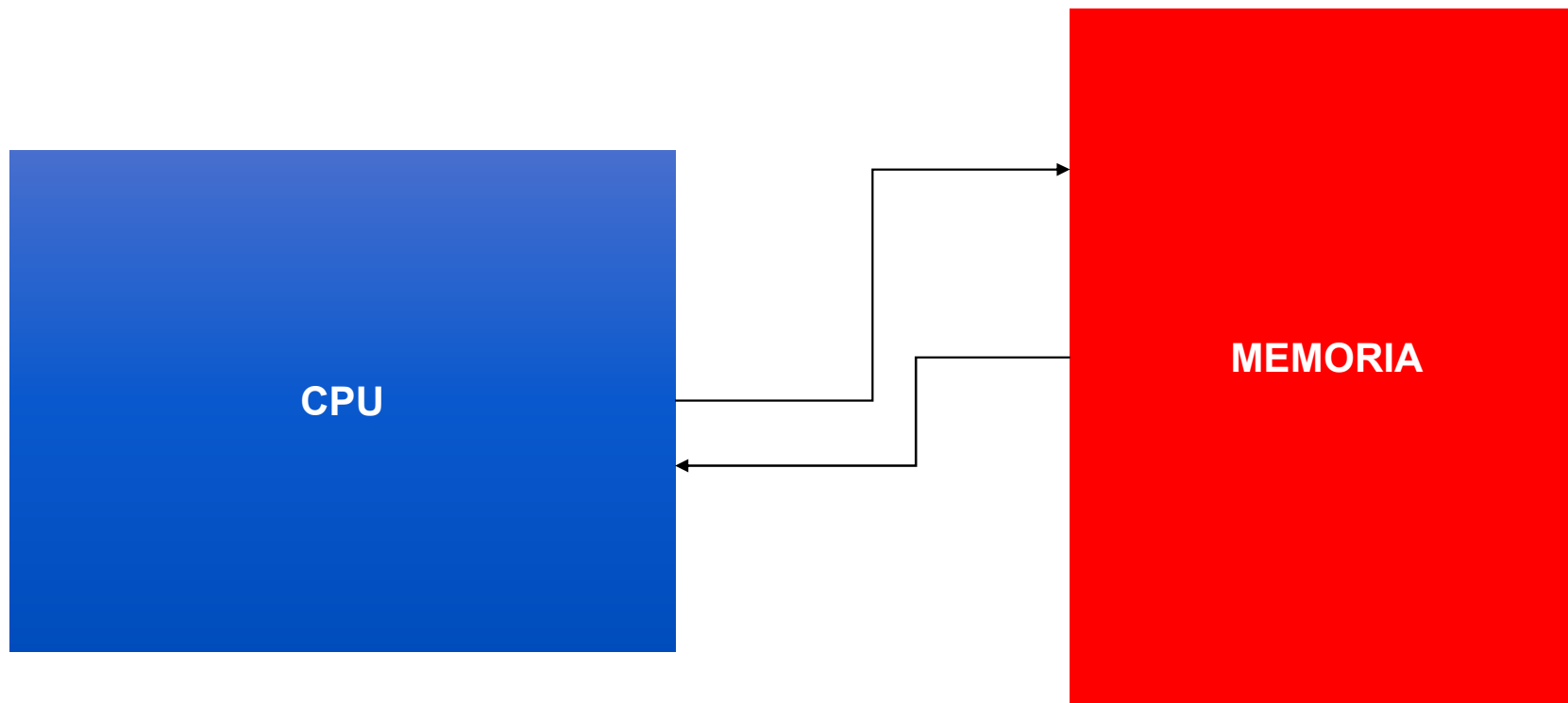
Modi di indirizzamento derivati

A glowing blue microchip is centered on a circuit board. The chip has a bright blue, textured surface and is surrounded by a glowing blue border. Numerous glowing blue lines and dots are scattered around the chip, suggesting a network or data flow. The background is a dark blue gradient with faint circuit patterns.

Modi di Indirizzamento

Modi di Indirizzamento

Il modo di indirizzamento è la strategia con la quale si specifica un indirizzo, e questo è utilizzato per accedere ad un operando in memoria oppure per raggiungere una posizione all'interno del programma.



Modi di Indirizzamento

Il MIPS ha tre modi di indirizzamento: l'**indirizzamento immediato** in cui l'operando è presente dopo l'opcode dell'istruzione: il **modo di indirizzamento a registro**, in cui gli operandi sono siti in dei registri (usato per le istruzioni logiche-aritmetiche); e, infine, il **modo di indirizzamento simbolico con spiazzamento** che consente uno scambio tra un operando sito in un registro e la memoria dati (e viceversa) e dal quale se ne derivano altri

Modo indirizzamento	Sintassi	Esempio	Indirizzo effettivo
Immediate	<istr> reg,imm <istr> imm	li \$t0,45 j pippo	L'operando è riportato all'interno dell'istruzione
Register	<istr> reg,reg,reg	add \$t0,\$t1,\$t2	Gli operandi sono nei registri
Symbol+Immediate Offset	<istr> reg,label+offset(\$reg)	lw \$t0,list+4(\$s0)	L'indirizzo è ottenuto sommando il contenuto del registro, l'etichetta e lo spiazzamento

Modi di Indirizzamento

Immediato

Il modo di indirizzamento immediato specifica direttamente un operando
L'operando è presente nel campo dell'istruzione subito dopo l'opcode

ESEMPIO:

li \$t0, 256

li \$t0, 1807198000

OSSERVAZIONE:

Il MIPS traduce l'istruzione:

li \$t0,256

in

addiu \$t0,\$zero,256

Mentre l'istruzione:

li \$t0,1807198000 (01101011 10110111 10100111 00110000)

25757

42800

*Poiché si eccedono i 16 bit di rappresentazione a disposizione del formato
è una pseudoistruzione che si risolve con LUI e OR*

Cioè:

lui \$at, 25757

#copia il valore nei 16bit più significativi di \$at

or \$t0,\$at, 42800

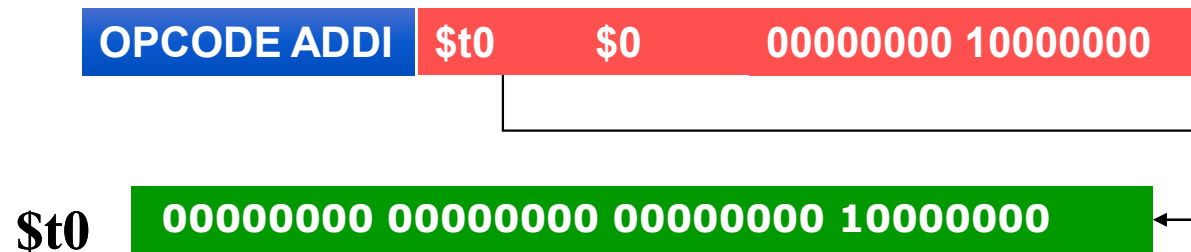
#copia il valore nei 16bit meno significativi di \$at

#e li sposta nel registro destinazione

Modi di Indirizzamento

Immediato

li \$t0, 128



Modi di Indirizzamento

Immediato

li \$t0,1807198000

OSS: $1807198000 = (01101011\ 10110111\ 10100111\ 00110000)_2$

OPCODE LUI \$at 01101011 10110111

\$at 01101011 10110111 00000000 00000000

OPCODE OR \$t0 \$at 10100111 00110000

\$t0 01101011 10110111 10100111 00110000

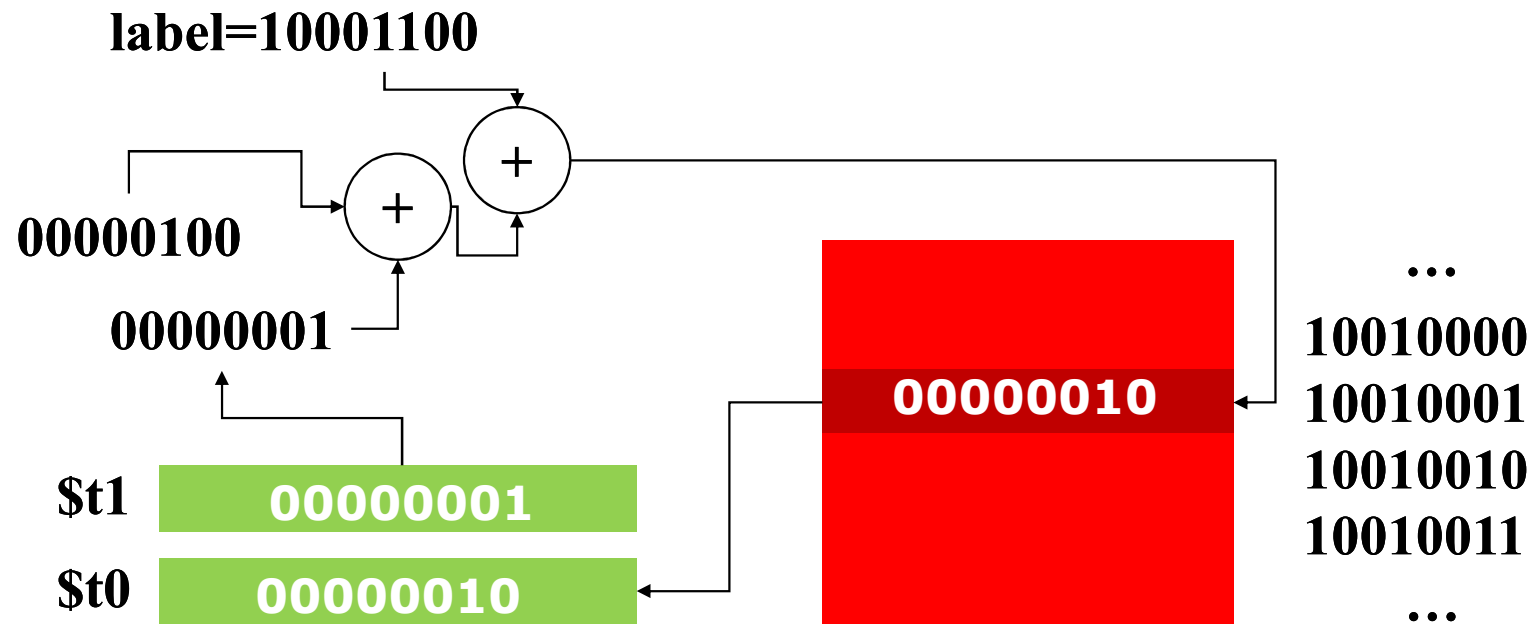
Modi di Indirizzamento

Simbolico con spiazzamento a registro

Questo modo di indirizzamento specifica un indirizzo tramite una etichetta, mentre l'offset è espresso attraverso un registro indice e uno spiazzamento ottenuto con una espressione

L'indirizzo è dato dalla somma dell'indirizzo associato all'etichetta più il contenuto del registro indice e il risultato dell'espressione

Esempio: **lb \$t0, label+4(\$t1)**



Modi di Indirizzamento

Simbolico con spiazzamento a registro (uso)

Questo modo di indirizzamento è utile, oltre per derivare gli altri modi di indirizzamento, anche per accedere alle strutture dati **aggregati** (la definizione di tipo, le **strutture** in C)

```
.text
.globl main
main:
    li $t7,0
    lw $t0,stud+0($t7)
    lh $t1,stud+4($t7)
    lb $t2,stud+6($t7)
    li $v0,10
    syscall

.data
stud:.word 11098231    #matricola
    .half 1974         #datanascita
    .byte 28          #mediavoti (interi)
```

Se avessi due aggregati *stud* come dovrei modificare \$t7 per prendere i campi del secondo aggregato?

A glowing blue microchip is centered on a circuit board. The chip and the board are illuminated with a bright blue light. Numerous glowing blue lines and dots are scattered around the chip, suggesting a network or data flow. The background is dark blue with some faint, out-of-focus lights.

Modi di indirizzamento derivati

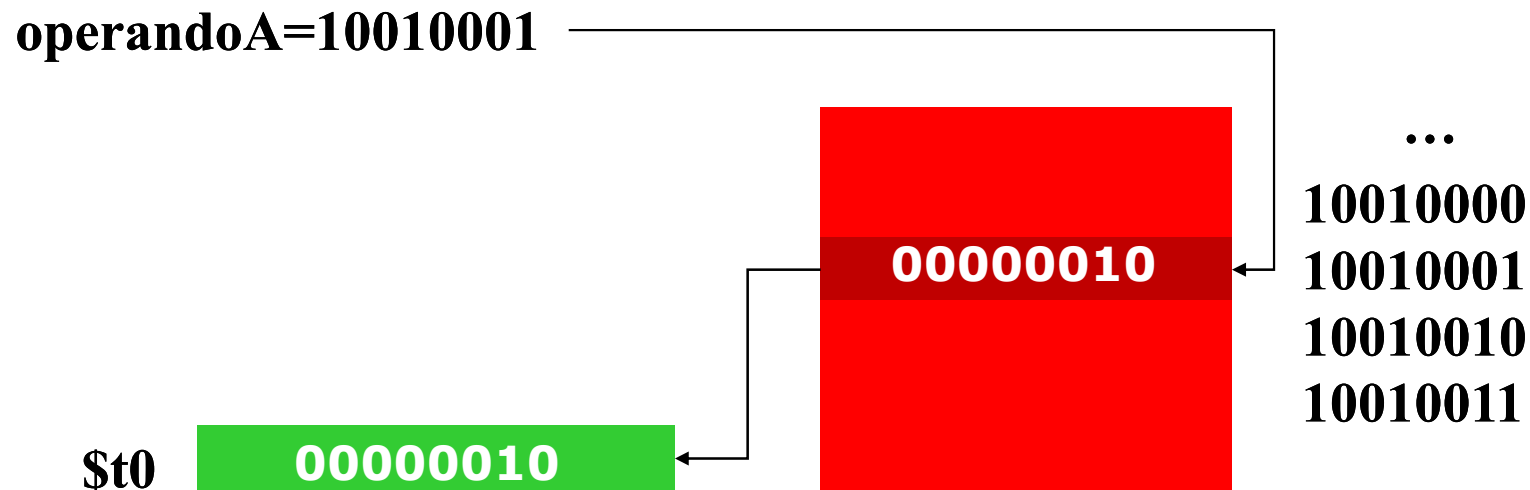
Modi di Indirizzamento

Simbolico

Il modo di **indirizzamento simbolico** (*symbol*) specifica un indirizzo tramite una etichetta.

L'assemblatore aggiunge al modulo oggetto generato, le informazioni di rilocalizzazione (sostituisce l'etichetta con l'indirizzo dove risiede l'operando)

Esempio: lb \$t0, operandoA



Modi di Indirizzamento

Simbolico (uso)

Questo modo di indirizzamento richiama il modo di indirizzamento assoluto e consente l'accesso ai dati in memoria

```
.text
.globl main
main:
    lw $t0,operW
    lh $t1,operH
    lb $t2,operB
    add $t3,$t0,$t1
    add $t3,$t3,$t2
    sw $t3,risultato
    li $v0,10
    syscall

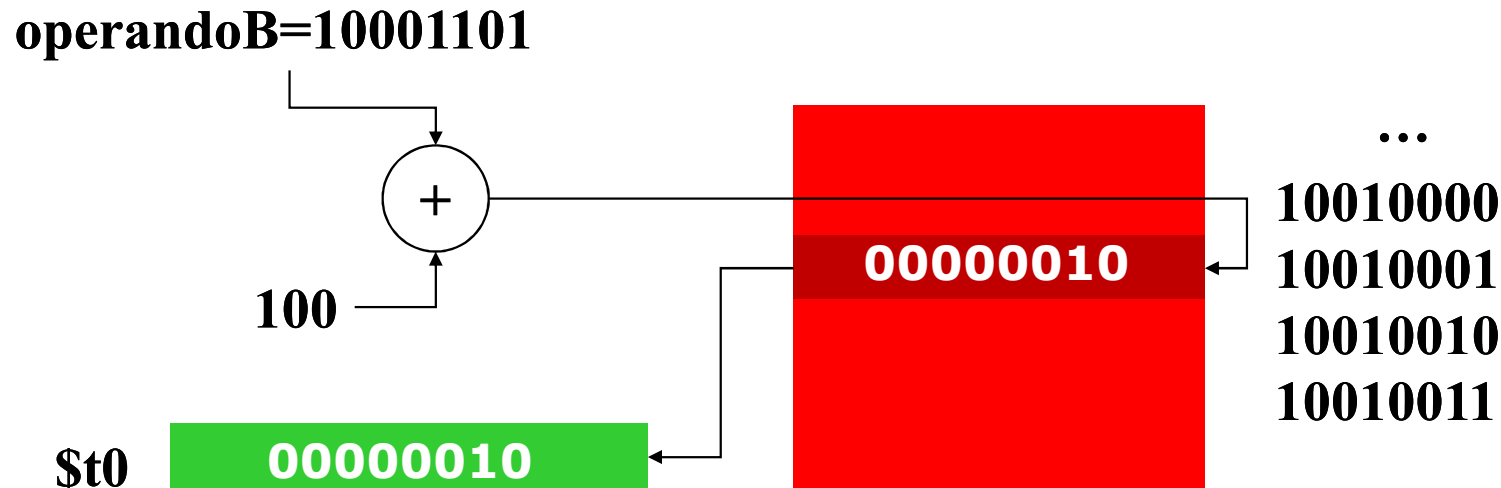
.data
operW: .word 11098231  #intero a 32bit
operH: .half 1974      #intero a 16bit
operB: .byte 28        #intero a 8bit
risultato:.word 0
```

Modi di Indirizzamento

Simbolico con spiazzamento

Il modo di indirizzamento simbolico con spiazzamento (*symbol+expression*) specifica un indirizzo base tramite una etichetta, mentre lo spiazzamento (offset) è specificato tramite un'espressione

Esempio: **lb \$t0, operandoA+4**



Modi di Indirizzamento

Simbolico con spiazzamento (uso)

Questo modo di indirizzamento offre l'opportunità di selezionare dati locati in posizione note

```
.text
.globl main
main:
    lw $t0, pippo
    lw $t1, pippo+4
    lw $t2, pluto
    lw $t3, paperino+-4 #contenuto di t1 e t3 è uguale al contenuto di t2

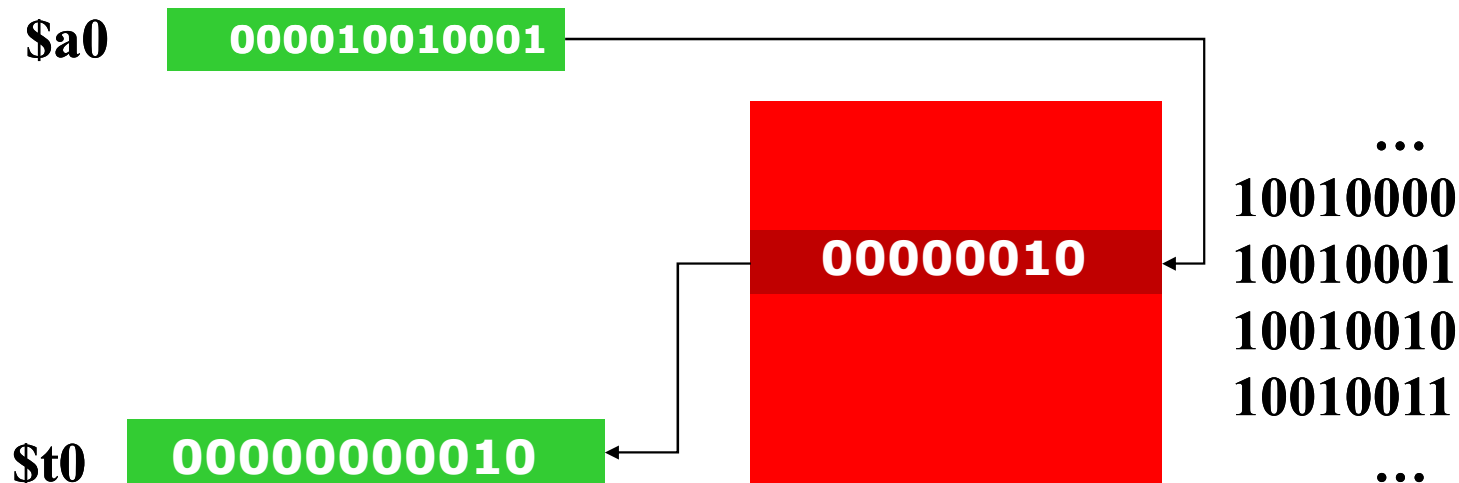
.data
pippo:.word 15 #si trova alla locazione 0x1001000
pluto:.word 256 #si trova alla locazione 0x1001004
paperino: .word 10 #si trova alla locazione 0x10010008
```

Modi di Indirizzamento

Differito a registro

Il **modo di indirizzamento differito a registro** specifica un indirizzo tramite un registro base (***base register***), assumendo come offset 0. L'operando si trova nella locazione di memoria indicata dal contenuto del registro base

Esempio: **lb \$t0, (\$a0)**



Modi di Indirizzamento

Differito a registro

Consente di sfruttare un registro e non un indirizzo per specificare la locazione di memoria dove risiede l'operando e utilizzare un modo di indirizzamento a lunghezza fissa.

```
.text
```

```
lw $t2,pippo
```

```
la $t0,pippo
```

```
lw $t1,($t0)    # $t1 e $t2 hanno lo stesso valore
```

```
.data
```

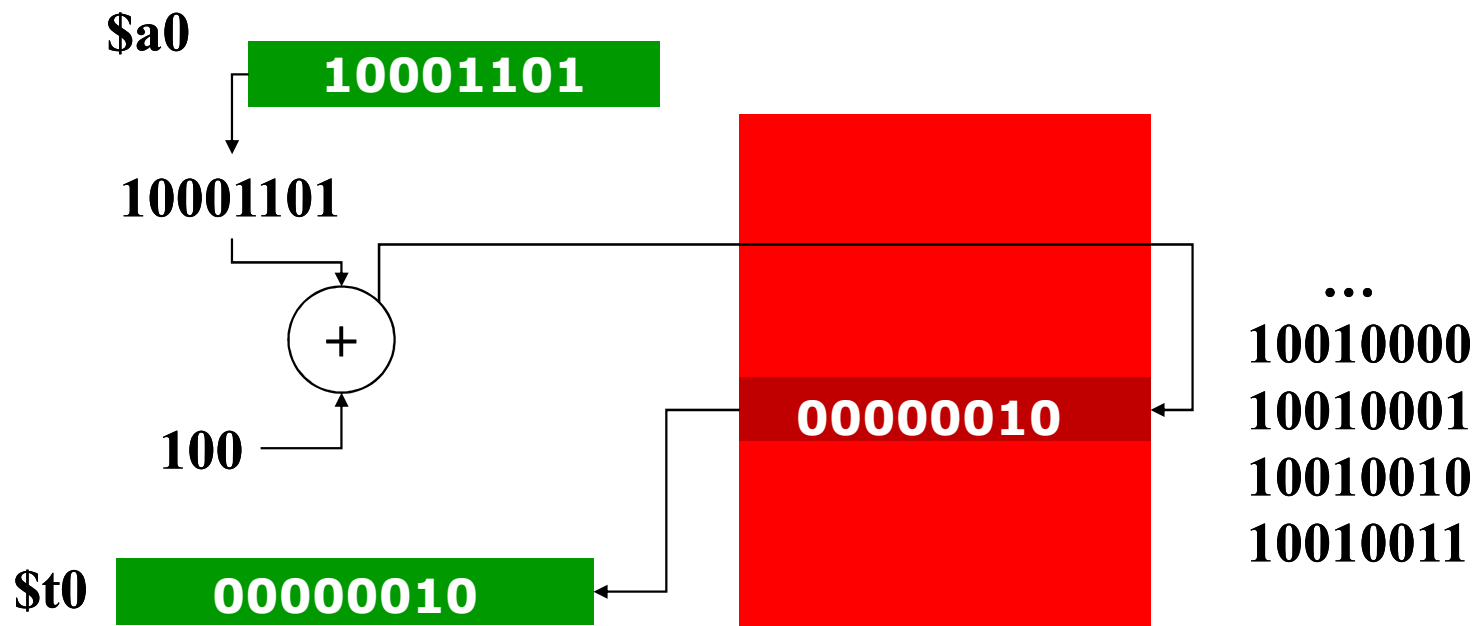
```
pippo: .word 256
```

Modi di Indirizzamento

Differito a registro con spiazzamento

Il modo di indirizzamento differito a registro con spiazzamento (*baseregister+expression*) specifica un indirizzo dato dalla somma tra il contenuto del registro base e uno spiazzamento (*offset*) descritto da una espressione

Esempio: `lb $t0, 4($a0)`



Modi di Indirizzamento

Differito a registro con spiazzamento

Consente di sfruttare un registro e non un indirizzo per specificare la locazione di memoria dove risiede l'operando e utilizzare un modo di indirizzamento a lunghezza fissa.

NB: Qualora lo spiazzamento superi i 16bit si ricorre ad uno sdoppiamento con LUI e ADDU

Inoltre questo modo di indirizzamento permette di gestire strutture dati con operandi contigui come i vettori

```
.text
```

```
la $t0, pippo
```

```
lw $t1, 8($t0)      # $t1 contiene 65000
```

```
.data
```

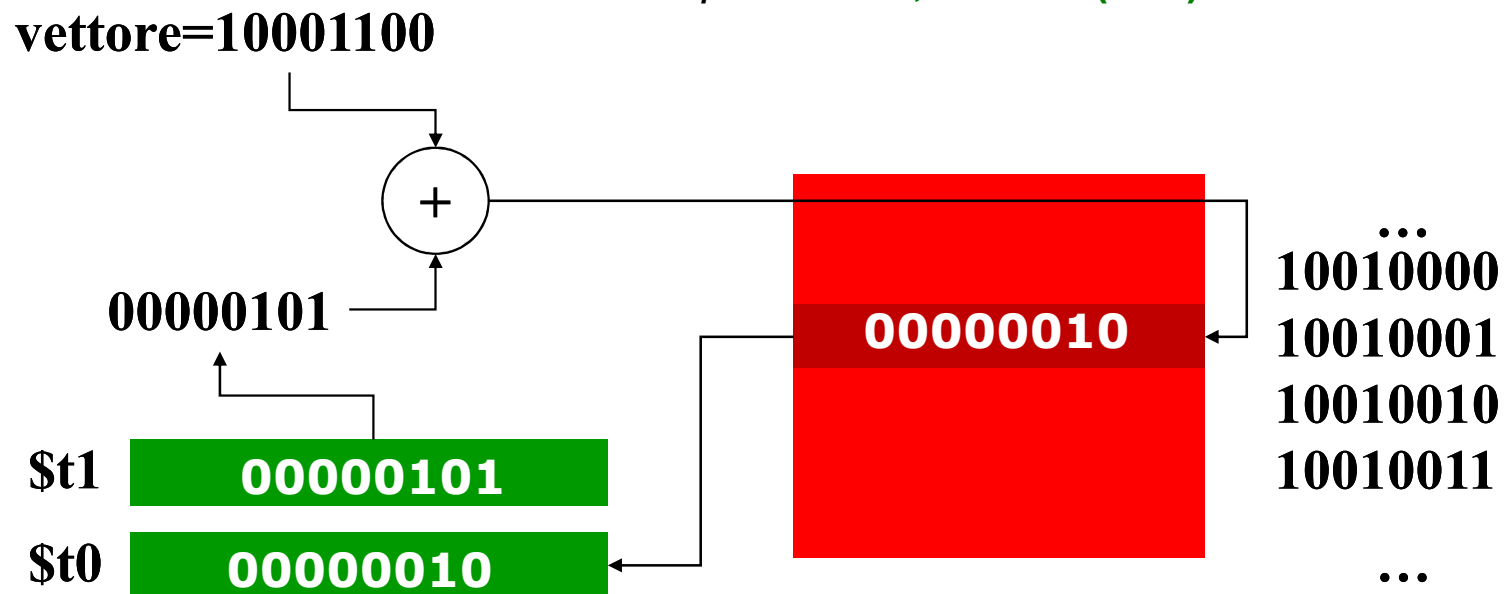
```
pippo: .word 256, 128, 65000, 45
```

Modi di Indirizzamento

Differito a registro con spiazzamento simbolico

Il modo di indirizzamento a registro con spiazzamento simbolico (*symbolic+baseregister+expression*) specifica un indirizzo tramite una etichetta, mentre l'offset viene specificato tramite un registro indice. L'indirizzo è dato dalla somma dell'indirizzo associato all'etichetta con il contenuto del registro indice.

Esempio: *lb \$t0, vettore(\$a0)*



Modi di Indirizzamento

Differito a registro con spiazzamento simbolico

Questo modo di indirizzamento permette di gestire strutture dati con operandi contigui come i vettori, aggregati e le matrici

```
.text
```

```
li $t0,8
```

```
lw $t1,pippo($t0)# $t1 contiene 65000
```

```
.data
```

```
pippo: .word 256,128,65000,45
```

The background is a dark blue gradient with intricate white and yellow circuit board traces and components. The traces are thin lines that branch out and connect various components. The components are represented by small white and yellow shapes, including rectangles, circles, and lines, which are scattered across the background. The word "FINE" is centered in the middle of the image in a bold, yellow, sans-serif font.

FINE