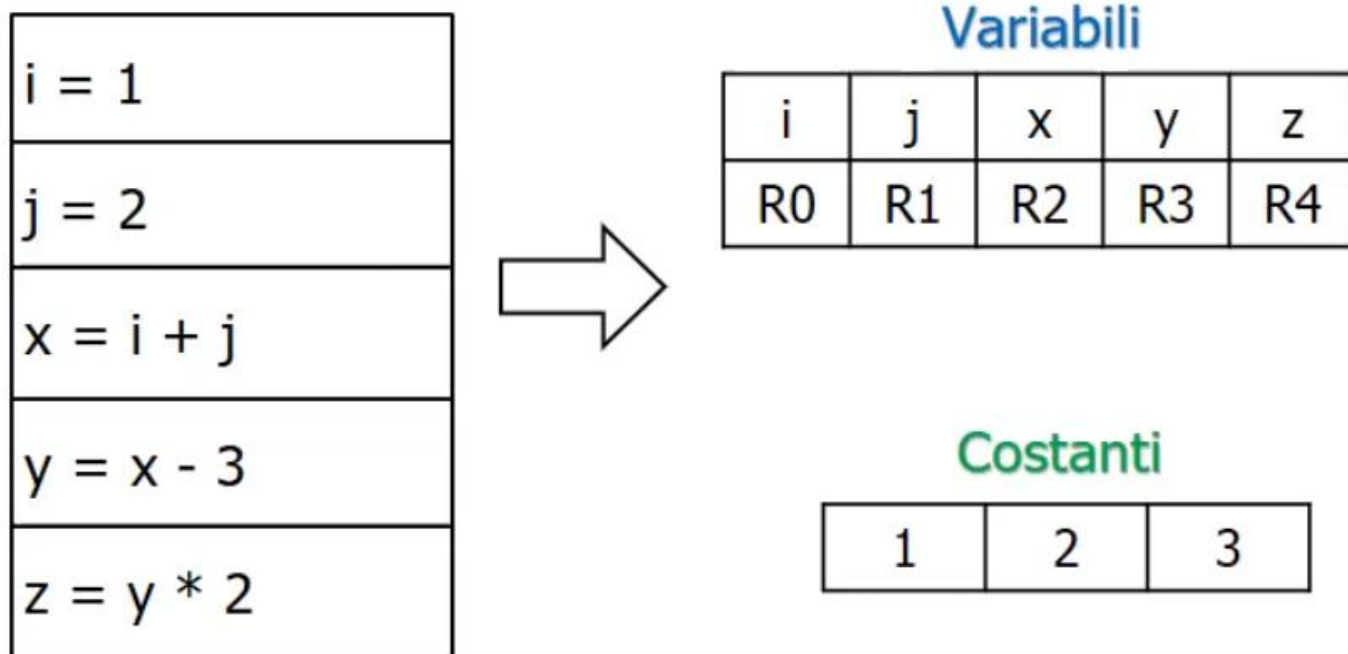


Traduzione dei blocchi DNS in linguaggio Assembly ARM

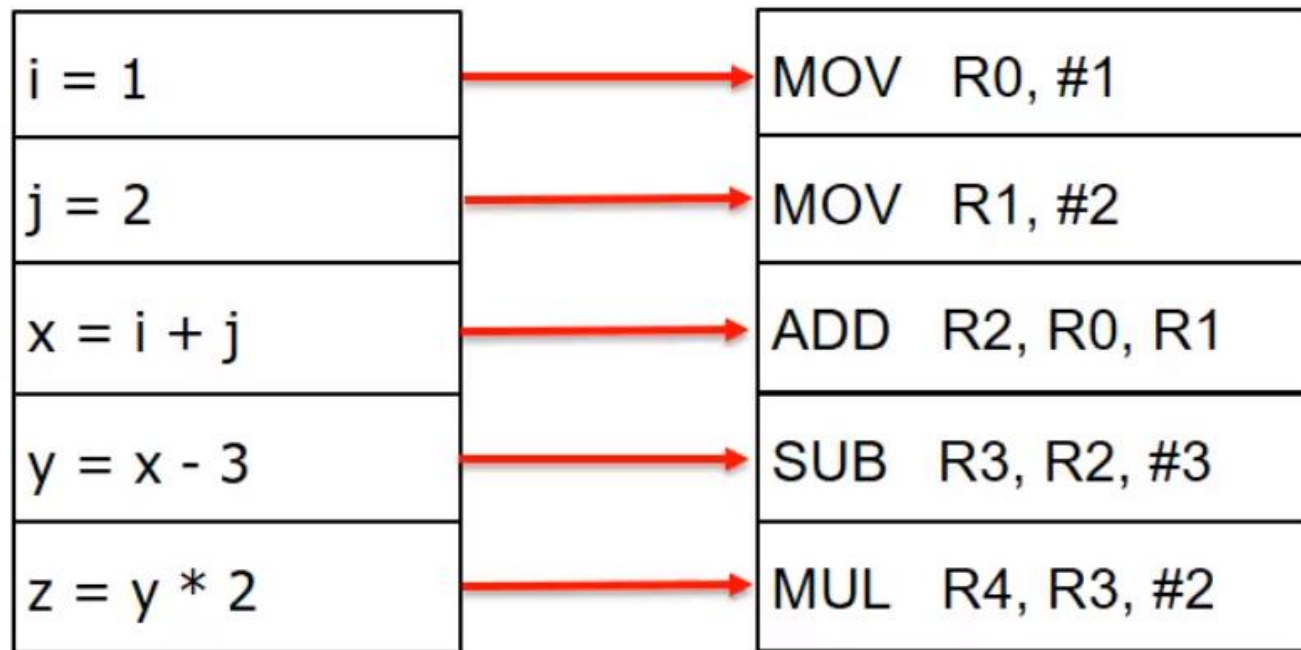
Sequenza di blocchi

Inizialmente si cercano tutte le variabili e costanti che compaiono ne DNS. Si assegna ogni variabile ai registri disponibili, se il loro numero è tale da contenere tutte le variabili (ed il numero di byte tali da poter essere memorizzati in un registro), altrimenti occorre memorizzarle in memoria.

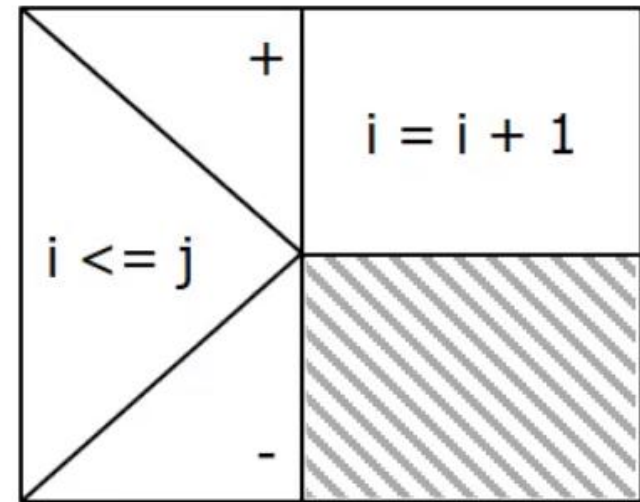
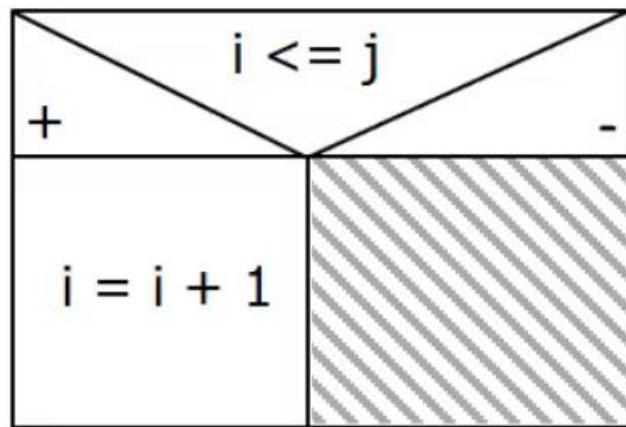


Sequenza di blocchi

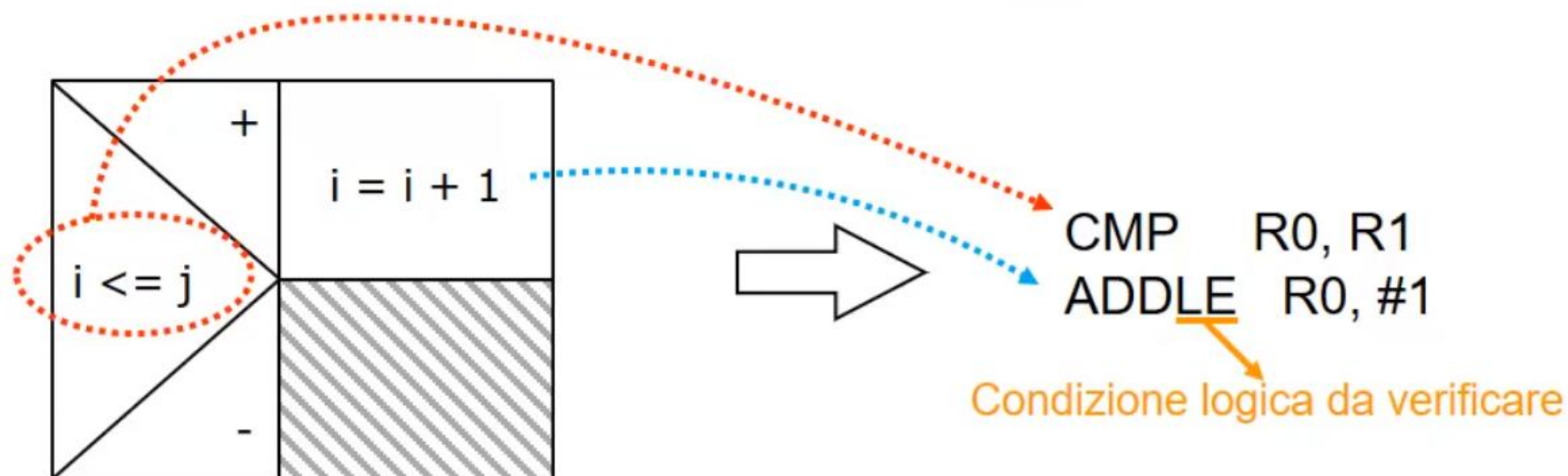
Se il blocco contiene una istruzione semplice, si traduce con la corrispondente istruzione assembly, altrimenti si segue la procedura che segue per ciascuna tipologia di blocco:



Selezione singola



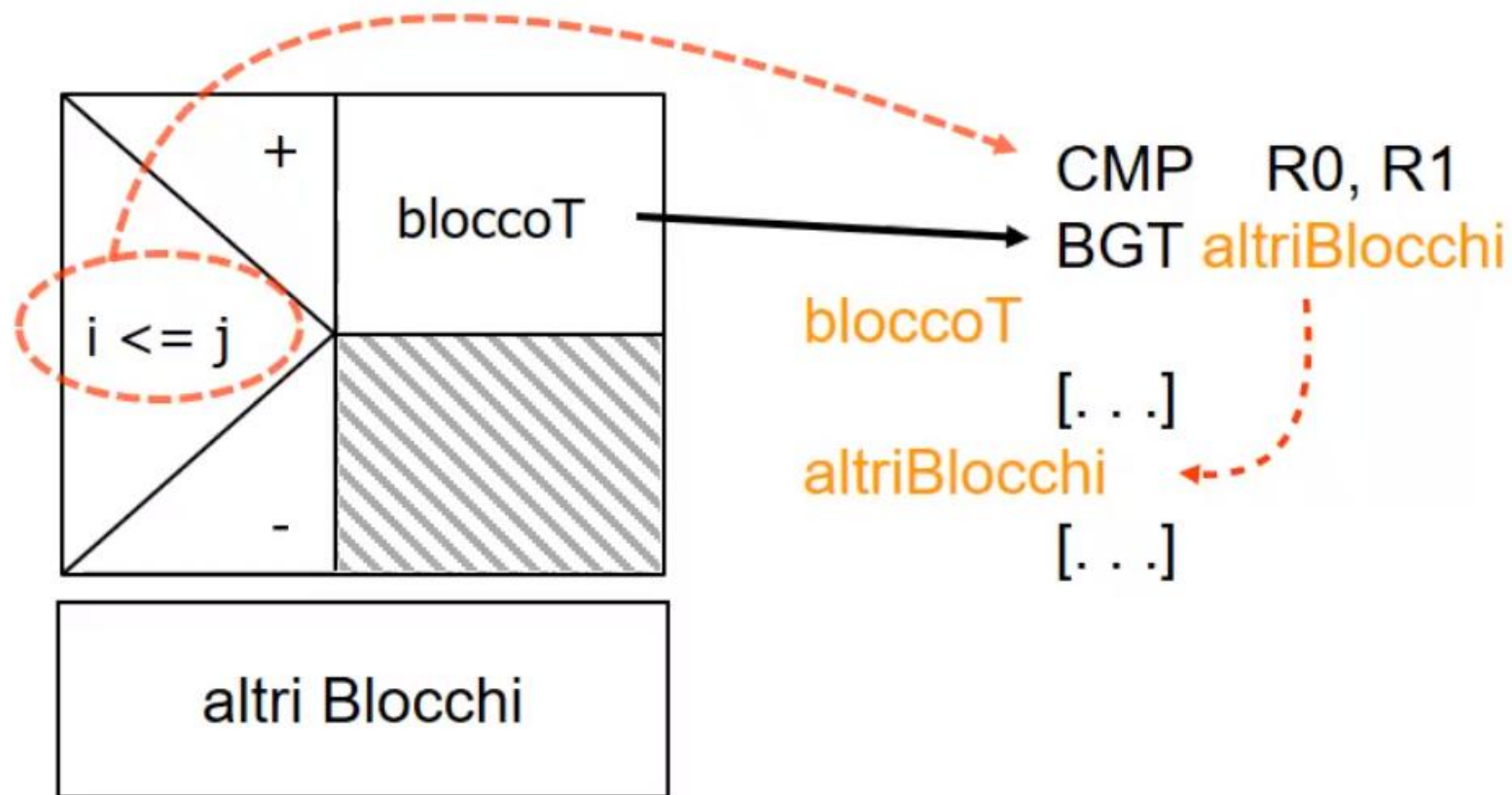
Selezione singola



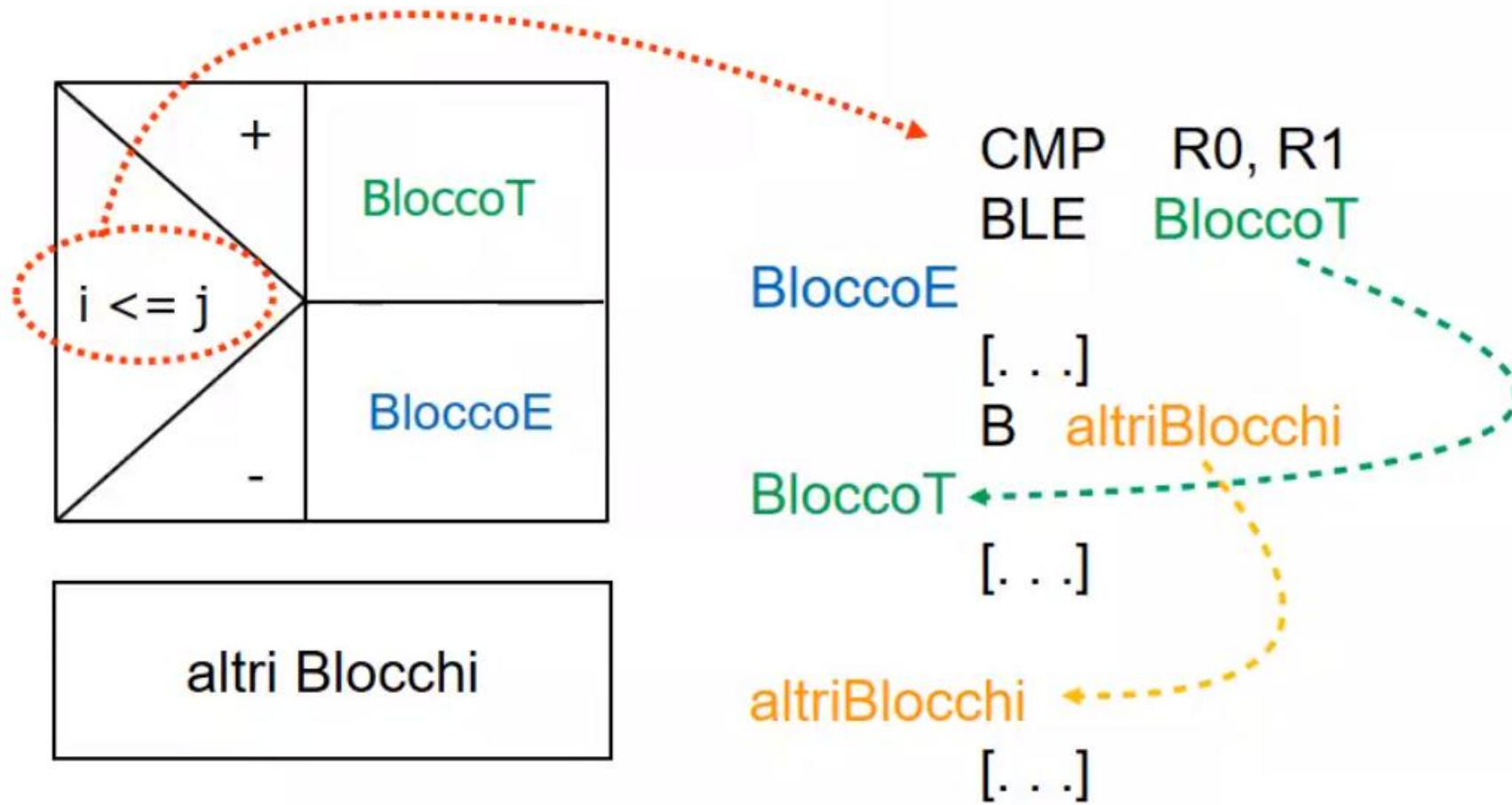
Se la condizione logica è verificata viene eseguita l'istruzione ADD, altrimenti il flusso di esecuzione proseguirà il normale corso.

Selezione singola con branch

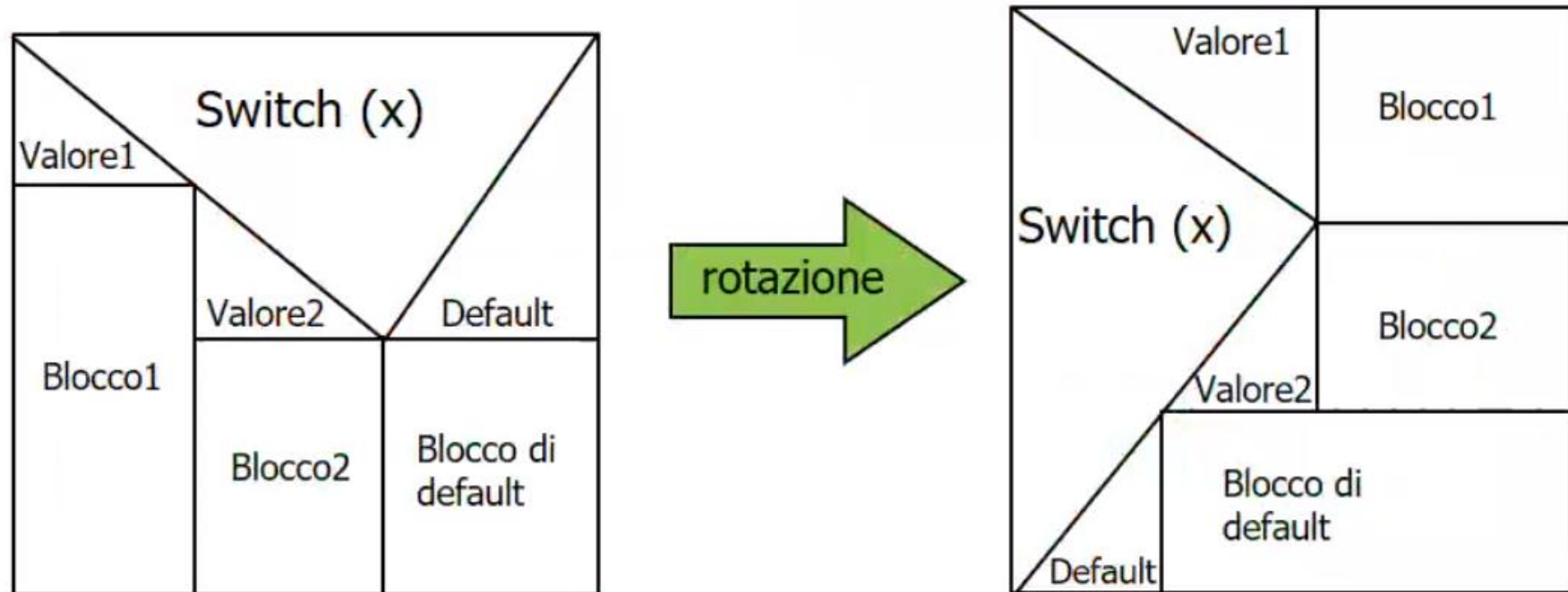
La selezione singola con esecuzione condizionata funziona nel caso ci sia una sola istruzione nel blocco, altrimenti è necessario utilizzare l'istruzione di branch.



Selezione binaria



Selezione multipla

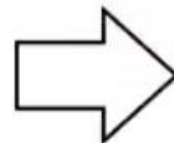
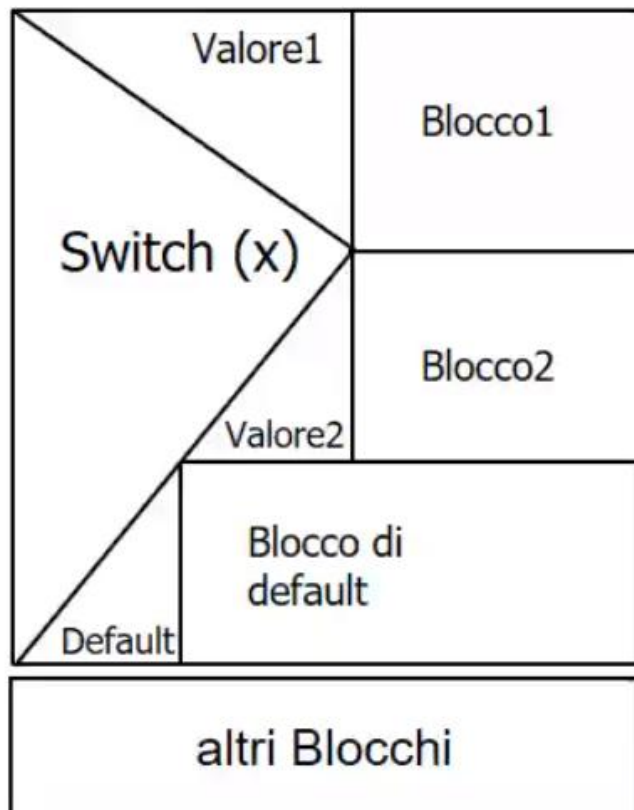


Variabili

| x | Valore1 | Valore2 |
|----|---------|---------|
| R0 | R1 | R2 |

Selezione multipla

$x = \text{Valore1} ? \longrightarrow \text{CMP } R0, R1$
 BEQ Blocco1
 $x = \text{Valore2} ? \longrightarrow \text{CMP } R0, R2$
 BEQ Blocco2
 $\text{Default} \longrightarrow \text{B Default}$



Blocco1

[. . .]
B altriBlocchi

Blocco2

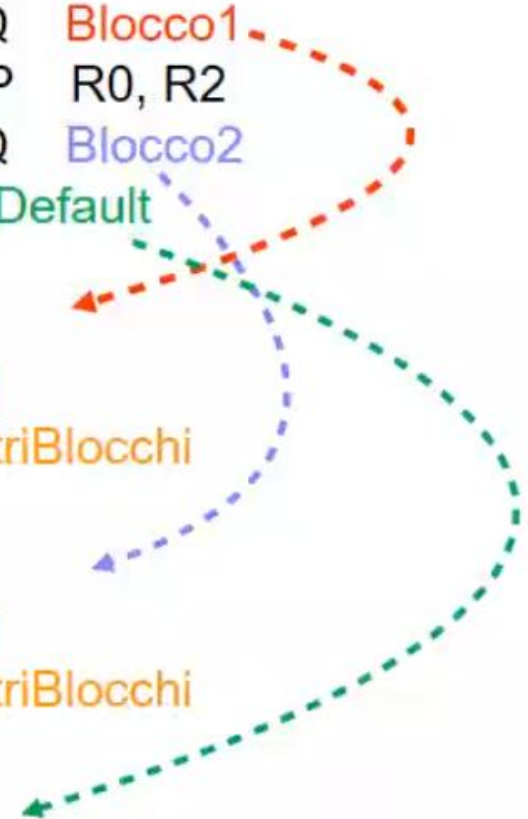
[. . .]
B altriBlocchi

Default

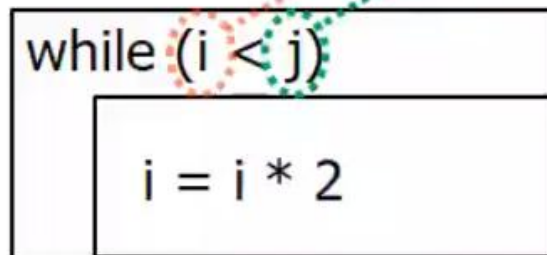
[. . .]
B altriBlocchi

altriBlocchi

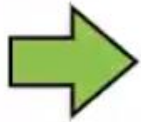
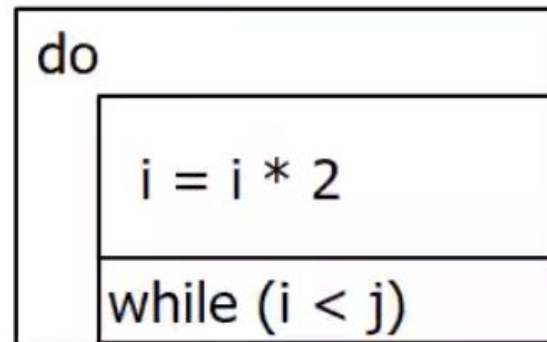
[. . .]



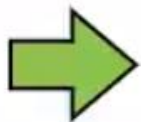
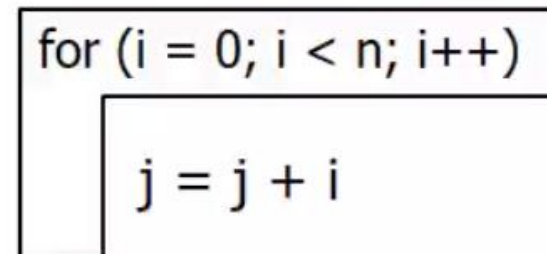
Blocchi iterativi



ciclo
CMP R0, R1
BGE end
MUL R0, R0, #2
B ciclo



end
ciclo
MUL R0, R0, #2
CMP R0, R1
BLT ciclo



MOV R0, #0
ciclo
CMP R0, Rn
BGE end
ADD R1, R1, R0
ADD R0, R0, #1
B ciclo
end

Massimo tra tre numeri interi

Esercizio

Calcolare il massimo tra tre numeri interi $x, y, z \in \mathbb{Z}$.

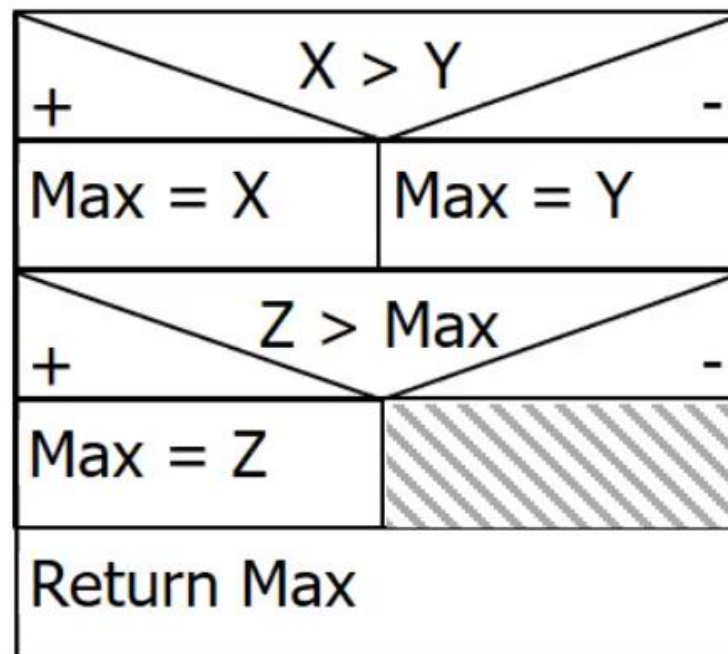
...proviamo a svolgerlo autonomamente, senza aiuto



DNS

Il corrispondente diagramma DNS è:

int max(X , Y , Z)



Una soluzione alternativa

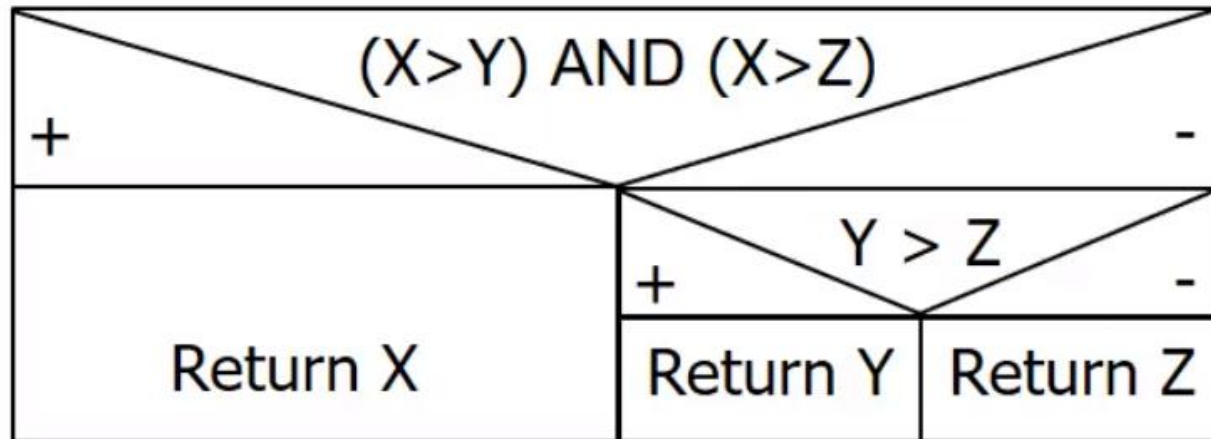
Tuttavia per ogni problema si possono trovare più soluzioni con diverse caratteristiche in termini di efficienza, complessità, riusabilità, ...

```
int max( X , Y , Z )
```

| | | | |
|----------|----------|----------|----------|
| X > Y | | | |
| + | | | - |
| X > Z | | Y > Z | |
| + | - | + | - |
| Return X | Return Z | Return Y | Return Z |

Terza soluzione

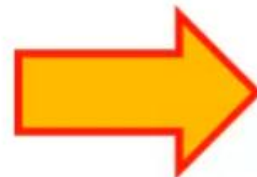
```
int max( X , Y , Z )
```



Codice assembly ARM finale

Sintassi GNU

```
.equ X, 10  
.equ Y, 30  
.equ Z, 20
```



Sintassi KEIL

```
X EQU 10  
Y EQU 30  
Z EQU 20
```

```
MOV    R0, #X  
MOV    R1, #Y  
MOV    R2, #Z
```

```
CMP     R0, R1    // X>Y ?  
MOVGT   R3, R0    // MAX=X  
MOVLTE  R3, R1    // MAX=Y  
CMP     R2, R3    // Z>MAX ?  
MOVGT   R3, R2    // MAX=Z  
          // R3=MAX
```

Moltiplicazione

Esercizio

Calcolare la moltiplicazione tra due numeri interi positivi $x, y \in \mathbb{N}$ senza utilizzare l'istruzione MUL.

...proviamo a svolgerlo autonomamente, senza aiuto

