



# Architettura degli Elaboratori I - B

Formato di Istruzione

**Branch**

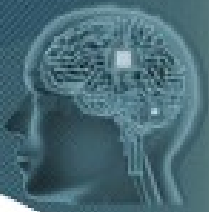
Daniel Riccio/Alberto Aloisio  
Università di Napoli, Federico II

27 marzo 2018



**Rif. Capitolo 6**

Digital Design and Computer Architecture-ARM, Harris-Harris, Edition-Morgan Kaufmann.



# Istruzioni di branch (o salto)

## **Branching** instruction

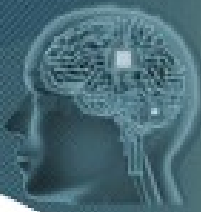
Un programma di solito esegue in sequenza, incrementando il Program Counter (PC) di 4 (32 bit) dopo ciascuna istruzione, in modo da puntare alla successiva istruzione.

Le istruzioni Branch permettono di cambiare il valore del PC. ARM include due tipi di branch: simple branch (B) e branch and link (BL).

Come altre istruzioni ARM, i branch possono essere condizionati o incondizionati.

Il codice assembly utilizza le etichette per indicare i blocchi di istruzioni nel programma.

Quando il codice assembly è tradotto in codice macchina, queste etichette vengono tradotte in indirizzi di istruzione.



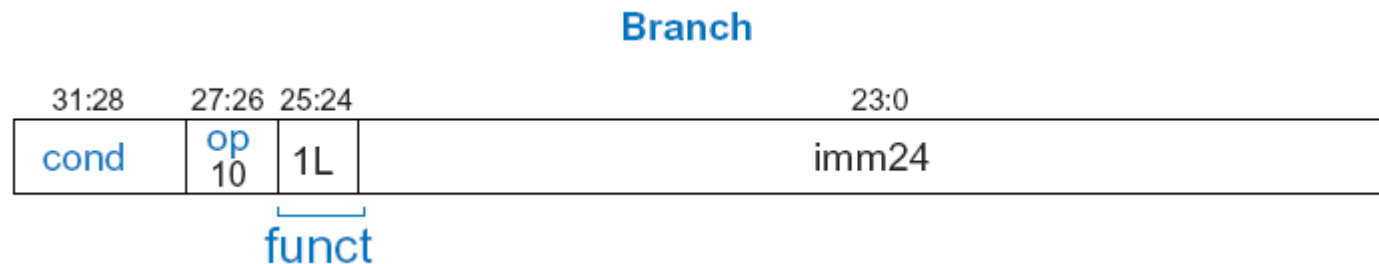
# Istruzioni di Branch

Le istruzioni di **Branch** utilizzano un unico operando costante di 24 bit, **imm24**.

Esse hanno un campo **cond** di 4 bit e un campo **op** di 2 bit, il cui valore è  $10_2$ .

Il campo **funct** ha solo 2 bit. Il bit più significativo è sempre 1 per i branch. Il bit meno significativo, **L**, indica il tipo di operazione di branch: 1 per **BL** e 0 per **B**.

I restanti 24 bit, **imm24**, rappresentano un valore in complemento a due, che specifica la posizione dell'istruzione relativamente all'indirizzo **PC** + 8.





# Il datapath per i branch

L'istruzione di salto somma una costante a **24-bit** a **PC+8** e scrive il risultato di nuovo nel **PC**.

La costante viene moltiplicata per 4 ed estesa con segno. Pertanto, la logica **Extend** necessita di una ulteriore modalità. **ImmSrc** è, quindi, esteso a 2 bit.

ImmSrc	ExtImm	Description
00	{24 0s} <i>Instr</i> <sub>7:0</sub>	8-bit unsigned immediate for data-processing
01	{20 0s} <i>Instr</i> <sub>11:0</sub>	12-bit unsigned immediate for LDR/STR
10	{6 <i>Instr</i> <sub>23</sub> } <i>Instr</i> <sub>23:0</sub> 00	24-bit signed immediate multiplied by 4 for B



# Istruzioni di Branch

L'istruzione **BL** (Branch and Link) è usata per la chiamata di una subroutine

- ▶ Salva l'indirizzo di ritorno (**R15**) in **R14**
- ▶ Il ritorno dalla routine si effettua copiando **R14** in **R15**:

**MOV R15, R14**

Il registro **R14** ha la funzione (architetturale) di subroutine Link Register (**LR**).

In esso viene salvato l'indirizzo di ritorno (ovvero il contenuto del registro **R15**) quando viene eseguita l'istruzione **BL** (Branch and Link).

```
....  
BL    function           ; call 'function'  
....  
                                           ; procedure returns to here  
....  
function                   ; function body  
....  
....  
MOV   PC, LR             ; Put R14 into PC to return
```



# Il datapath completo

Dato che **PC+8** è letto dalla prima porta del register file, è necessario un **multiplexer** per selezionare **R15** come ingresso di **RA1**. Il multiplexer è controllato dal segnale **RegSrc**, il cui valore è preso dai bit **Instr<sub>19:16</sub>**, per la maggior parte delle istruzioni ed è impostato a **15** per le istruzioni di branch (**B**).

**MemtoReg** è impostato a **0** e **PCSrc** è impostato a **1** per selezionare il nuovo **PC** da **ALUResult**.

