

## Carry flag(C)

- After a sum, C = 1 if the result size is 33 bit.

$$\begin{array}{r} 1010\ 0001\ 0001\ 0001\ 0001\ 0001\ 0001\ 0001\ + \\ 1011\ 0101\ 0101\ 0101\ 0101\ 0101\ 0101\ 0101\ = \\ \hline \text{C} = 1\ 0101\ 0110\ 0110\ 0110\ 0110\ 0110\ 0110\end{array}$$

- After a subtraction, C is inverted. Here C = 1:

$$\begin{array}{r} 1101\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ - \\ 0101\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ = \\ \hline 1000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\end{array}$$

- After a move or logical instruction, C is the result of an inline barrel shifter operation.

*NOTA BENE: Il carry non ha significato con operazioni con segno.*

*NOTA BENE2: Una sottrazione unsigned A-B viene convertita in una somma signed A+(-B) e quindi essa genererà un carry uguale a 1.*

## Negative flag(N)

- It corresponds to the first bit of the result.
- If N = 1, a 2's complement number is negative.
- Example:  $-4 + (-3) = -7 \rightarrow N = 1$

$$\begin{array}{r} 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1100\ + \\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1101\ = \\ \hline \text{N} = 1\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1001\end{array}$$

- Why N = 1 in the following sum?

$$\begin{array}{r} 0111\ 0001\ 0001\ 0001\ 0001\ 0001\ 0001\ 0001\ + \\ 0011\ 0101\ 0101\ 0101\ 0101\ 0101\ 0101\ 0101\ = \\ \hline \text{N} = 1\ 0101\ 0110\ 0110\ 0110\ 0110\ 0110\ 0110\end{array}$$

## Overflow flag(V)

# Overflow flag V

- It is set if, in a sum of values with the same sign, there is a change in the MSB

This is a carry

$$\begin{array}{r} 1010\ 0001\ 0001\ 0001\ 0001\ 0001\ 0001\ 0001\ 0001\ 0001\ + \\ 1011\ 0101\ 0101\ 0101\ 0101\ 0101\ 0101\ 0101\ 0101\ 0101\ = \\ (V = 1)\ 1\ 0101\ 0110\ 0110\ 0110\ 0110\ 0110\ 0110\ 0110\ 0110 \end{array}$$

- V = 0 if both conditions occur: the result is right

$$\begin{array}{r} 1110\ 0001\ 0001\ 0001\ 0001\ 0001\ 0001\ 0001\ 0001\ 0001\ + \\ 1011\ 0101\ 0101\ 0101\ 0101\ 0101\ 0101\ 0101\ 0101\ 0101\ = \\ (V = 0)\ 1\ 1001\ 0110\ 0110\ 0110\ 0110\ 0110\ 0110\ 0110\ 0110 \end{array}$$

A differenza del carry, il V flag è sul 32esimo bit, mentre il C flag sul 33esimo(bit in eccesso)

## Zero flag(Z)

Se il risultato è tutto 0

## Comparison instructions

Confrontano un valore o testano dei bit.

Hanno la forma seguente: *compare/test* <Rd>, <operand2>

aggiornando i flag e non aggiornando Rd.

Il secondo operando può essere:

- registro shiftato
- una costante

## CMP and CMN

**CMP:** confronta Rd-operand2

**CMN:** confronta operand2+Rd

## TST and TEQ

**TST:** testa Rd AND operand2

**TEQ:** testa Rd EOR operand2

## Accessing PSR

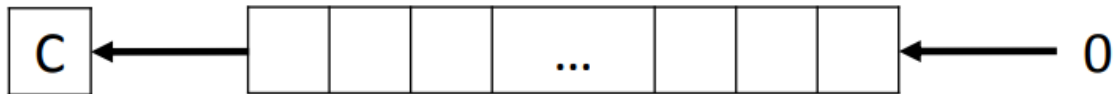
**MRS <Rn>,<Sreg>:** copia *Sreg* in *Rn*

**MSR <Sreg><Rn>:** copia *Rn* in *Sreg*

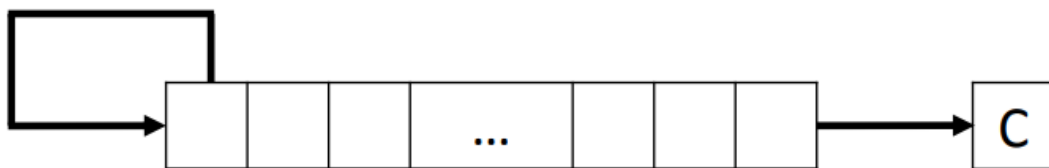
## **Shift/rotate instructions**

# Shift instructions

- LSL <Rd>, <Rn>, <op2>

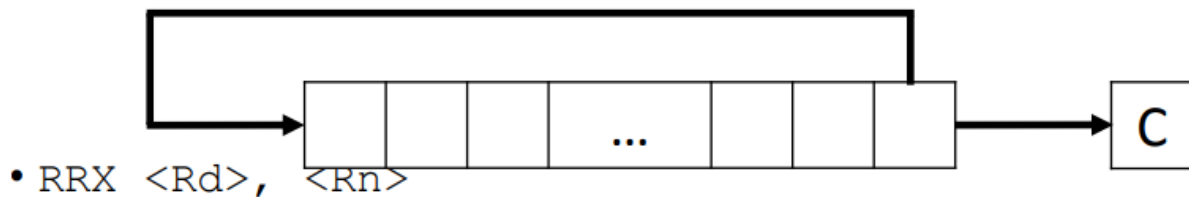


- LSR <Rd>, <Rn>, <op2>

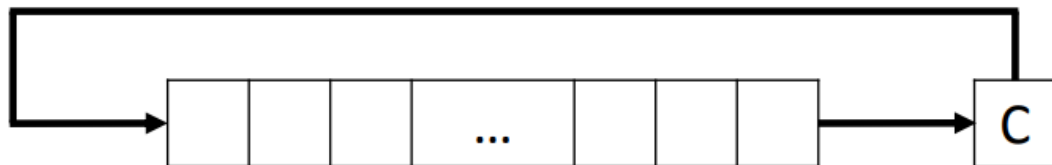


## Rotate instructions

- ROR <Rd>, <Rn>, <op2>



- RRX <Rd>, <Rn>



Queste istruzioni possono essere usate anche come parte del barrel shifter di una comune istruzione ALU (come l'ADD)