# Técnicas Digitales II

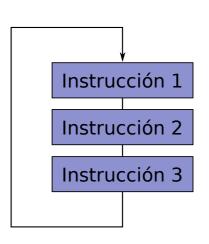
Banderas y Ejecución Condicional

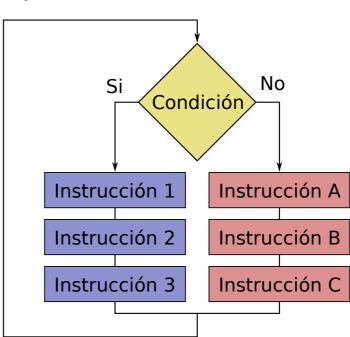
### Ejecución Condicional

 Los programas tendrían poco sentido si solo pueden ejecutar las instrucciones en un orden predefinido y de la misma manera todas las veces que son corridos.

 La ejecución condicional, significa que una instrucción o grupo de ellas pueden ser ejecutadas o no en función de una

condición.





### Ejecución Condicionada

• En lenguaje C, teníamos la instrucción **If() else** que permitía seleccionar que instrucciones serían ejecutadas de acuerdo a una condición definida en la propia instrucción **if()**.

```
If (a==b || b>c) {
    Intrucción1;
    Intrucción2;
    ...
} else {
    Intrucción A;
    Intrucción B;
    ...
}
```

• En ensamblador, es algo mas complicado

### Ejecución Condicionada

- Las Instrucciones en ARM pueden opcionalmente establecer banderas de condición, basadas en: si el resultado es negativo, si es cero, etc.
- Las instrucciones siguientes, se pueden ejecutar condicionadas a el estado de esas banderas.
- Es decir, la ejecución condicionada se divide en dos partes:

Aquí NO se define condición, solo se actualizo las banderas

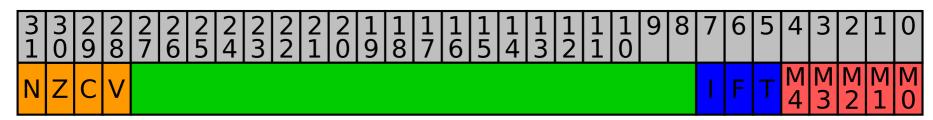
Se realiza alguna operación entre Los dos valores a comparar, Por ejemplo CMP, esto permite establecer las banderas

Aquí SI se define la condición (mayor, menor, igual, etc)

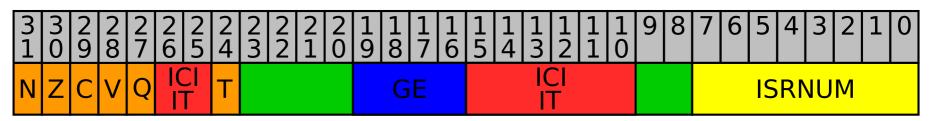
Se condiciona la o las siguientes instrucciones a el resultado de la comparación anterior

#### **Banderas**

 El primer paso es actualizar las banderas, estas banderas son cuatro y se encuentran en el Current Program Status Register (CPSR).



Registro de Estado ARM7TDMI



Registro de Estado Cortex-M4

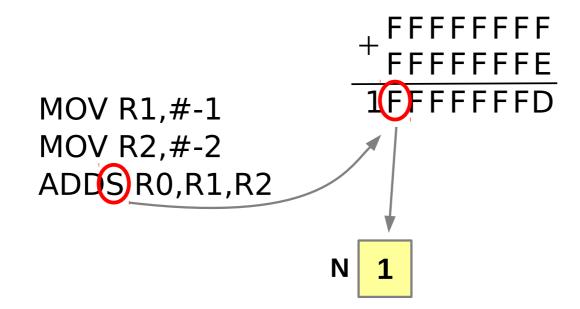
#### **Banderas**

# El Registro de Status, puede ser modificador de cuatro maneras

- Instrucciones de Comparación, alguna de las 4 instrucciones específicas que solo modifican las banderas tales como CMP, TST, etc.
- Instrucciones de procesamiento de datos con el agregado de la "S" al nemónico, esto aplica para todas las instrucciones de la ALU que no son de comparación.
- Operando directamente con el Registro de Status, activando o borrando alguna bandera.
- Las instrucciones Thumb de 16 bits

### Bandera N (Negativo)

- Esta bandera permite chequear si un número es negativo, tomando el estado del bit 31 del resultado.
- Esta bandera es útil si el número a evaluar es con signo, cualquier valor almacenado será evaluado por esta bandera como número con signo.



### Bandera N (Negativo)

- No siempre el estado de esta bandera debe ser tomado como cierto.
- En el ejemplo se suma dos números grandes.

MOV R1,#2 063 597 568 MOV R2,#80 530 6368 ADDS R0,R1,R2

Bit 31 y N en uno (1)

Bit 31 en cero (0) 7B000000 30000000 AB000000

El resultado se puede interpretar de dos maneras

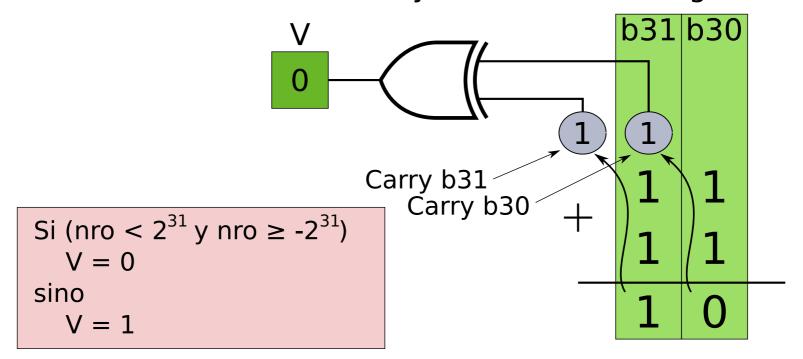
Si los números son naturales (sin signo) el resultado no es negativo, es un número con el bit mas significativo en 1

En ninguno de los casos N representa el signo del resultado

Si los números son enteros con signo el resultado no es negativo, es que el resultado no puede representarse en 32 bits (overflow)

#### Bandera V (Desborde o Overflow)

- Esta bandera, permite determinar luego de una operación con signo tal como suma, resta o comparación, si el resultado sufrió un desborde.
- Se verifica el acarreo del bit 30 y del bit 31 si son iguales.



#### Bandera V (Desborde o Overflow)

- Ejemplo, dos números con signo son sumados
- Se verifica el acarreo del bit 30 y del bit 31 si son iguales.

#### Bandera Z (Cero)

- Es la bandera mas sencilla de interpretar.
- Indica si la totalidad de los bit en la última operación resultaron cero (0).
- Un ejemplo simple es un contador que llegó a cero, esta bandera indicaría la situación de final de cuenta.

- Esta bandera se activa de acuerdo a las siguientes condiciones:
  - La suma es mayor o igual a 2<sup>32</sup>.
  - El resultado de una resta es positivo.
  - Por el resultado de un corrimiento.

#### Ejemplo de suma de dos números.

• El resultado excede el valor 2<sup>32</sup>.

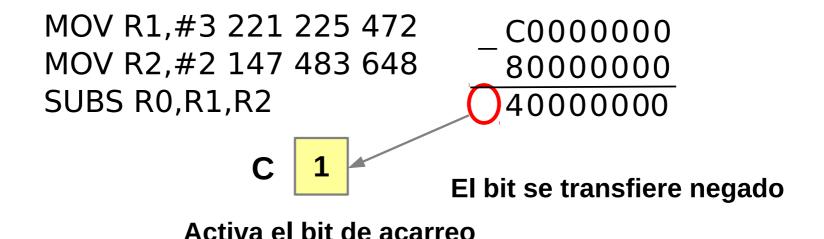
MOV R1,#2 063 597 568 MOV R2,#4 026 531 840 ADDS R0,R1,R2

7B000000 F0000000

Activa el bit de acarreo

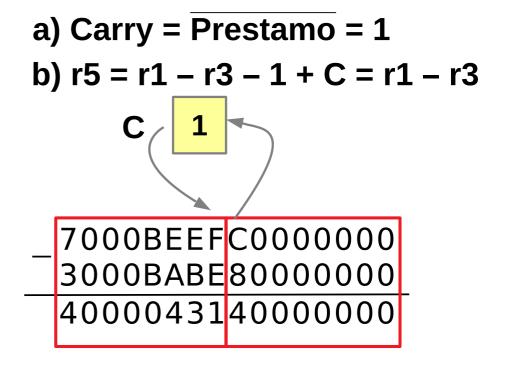
#### Ejemplo de una resta de dos números

 Aquí el bit de acarreo de la resta, es invertido, representado mas un préstamo que un acarreo (Motorola MC6800 y posteriores).



#### Ejemplo de una resta de dos números de 64 bits

 La combinación de SUBS con SBC permite realizar una resta de dos números de 64 bits.



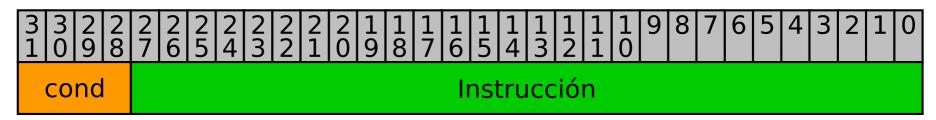
#### **Otras Banderas**

- El conjunto de banderas visto es el mas común.
- Otras familias de microprocesadores, o familias mas nuevas de ARM, aparecen una o mas banderas extras.

Intel x86	P Paridad: 0 paridad impar 1 paridad par A Acarreo auxiliar: indica el acarreo entre el bit 3 y 4 (utilizado en suma de BCD)
ARM	<b>Q Saturación</b> : Utilizada en conjunto con insrucciones específicas, permite detectar saturación en variables de 32 bits con contenido de 16 bits

#### Condición

- Finalmente y luego de haber actualizado las banderas, se condiciona la ejecución de la instrucción siguiente utilizando el estado de estas.
- Para esto, todas las instrucciones en ARM, poseen 4 bits en un campo denominado condición, permitiendo 16 condiciones distintas.
- En ensamblador, la condición se define en el nemónico de la instrucción como un sufijo con otro nemónico de la condición.



Campo con la condición de la instrucción

#### Condición

#### **Ejemplo: Valor absoluto**

- Compara el valor del registro a calcular con 0.
- Se condiciona la resta reversa a si r1 es negativo.

CMP r1,#0 RSBMI r1,r1,#0 "MI" condiciona la ejecución si el resultado fue negativo (bandera N=1) Si se cumple, entonces la resta reversa calcula r1=0-r1=-r1

La comparación de un registro con 0, permite evaluar el propio registro, de esta forma podemos determinar por ejemplo si el registro es cero o su signo.

#### Listado de condiciones

EQ	Igual	Z = 1	0000
NE	Diferentes	Z = 0	0001
CS/HS	Con Acarreo / Sobre o Igual	C = 1	0010
CC/LO	Sin Acarreo / Debajo	C = 0	0011
MI	Negativo	N = 1	0100
PL	Positivo	N = 0	0101
VS	Desborde	V = 1	0110
VC	Sin Desborde	V = 0	0111
HI	Sobre	C = 1 y Z = 0	1000
LS	Debajo o Igual	C = 0  o  Z = 1	1001
GE	Mayor o Igual	N xorV	1010
LT	Menor	N xor V	1011
GT	Mayor	$Z = 0 y (\overline{N \times V})$	1100
LE	Menor o Igual	Z = 1 o (N xor V)	1101
AL	Siempre / incondicional	Ignora las banderas	1110

- En comparación de números, las condiciones difieren si se trata de números con o sin signo
- Resulta obvio que la bandera N se utiliza para números con signo mientras que el C no tiene mucha utilidad con estos tipos de números.
- Se observa entonces, que las banderas N y V serán utilizadas solo para comparaciones con signo, C lo será para comparaciones sin signo y Z podrá utilizarse en ambos tipos.
- Se divide entonces la lista de condiciones en subconjuntos, sin signo y con signo

#### • Sin Signo

CS	Con acarreo	CC	Sin acarreo
HS	Sobre o igual	LS	Debajo o Igual
HI	Sobre	LO	Debajo

#### Con Signo

VS	Con Desborde	VC	Sin Desborde
GE	Mayor o igual	LE	Menor o Igual
GT	Mayor	LT	Menor
MI	Negativo	PL	Positivo

$$\begin{array}{c} \text{Sin signo} \quad \text{Con signo} \\ \text{CMP r1,r2} & \text{r1=} \ 1001 & 9 & -7 \\ \text{r2=} \ 0010 & 2 & 2 \\ \\ + \frac{1001}{1110} & \\ \hline 10111 & \longrightarrow & \text{N=0, C=1, V=1, Z=0} \\ \text{HS = Verdadero (C=1)} \\ \text{GE = Falso ($\overline{N}$ xor $V$ = 0 xor $1=$\overline{1}=0$)} \end{array}$$

### Ejemplo

```
R2 = 0x80000000

R3 = 0x00000001
```

```
+ 80000000

FFFFFFF

17FFFFFF → N=0, C=1, V=1, Z=0
```

CMP R2, R3

ADDEQ R4, R5, #78 Z=0 Falso

ANDHS R7, R8, R9 C=1 Verdadero

ORRMI R10, R11, R12 N=0 Falso

EORLT R12, R7, R10  $(N \times V) = (0 \times I) = 1 \text{ Verdadero}$ 

#### Bibliografía

Harris & Harris. Digital design and computer architecture: ARM edition. Elsevier, 2015. Capítulo 6.

William Hohl & Christopher Hinds. ARM assembly language. Fundamentals and techniques. 2nd edition. CRC press, 2015. Capítulo 7.

¿ Preguntas ?