

NÚMEROS ENTEROS

REBOSE (OVERFLOW)

- Cuando el resultado de una operación aritmética no se puede representar en el n° de bits utilizado se produce un rebose (*overflow*).
- En sumas y restas, el rebose se puede detectar:
 - Aritmética **sin signo**: si el acarreo (*carry*) es 1.
 - Aritmética **con signo**:
 - Suma: si ambos operandos tienen el mismo signo y el resultado tiene signo distinto.
 - Resta: si ambos operandos tienen distinto signo y el signo del resultado coincide con el del sustraendo.
- La multiplicación y la división sí son dependientes del tipo de aritmética (sin o con signo), por lo que habrá operaciones distintas.

NÚMEROS ENTEROS

EJEMPLOS

SUMAS:

$$\begin{array}{r} 0110 \quad 6 \quad (+6) \\ + 0001 \quad 1 \quad (+1) \\ \hline 0111 \quad 7 \quad (+7) \end{array} \quad C=0, V=0$$

$$\begin{array}{r} 0110 \quad 6 \quad (+6) \\ + 0010 \quad 2 \quad (+2) \\ \hline 1000 \quad 8 \quad (-8) \end{array} \quad C=0, V=1$$

$$\begin{array}{r} 0110 \quad 6 \quad (+6) \\ + 1010 \quad 10 \quad (-6) \\ \hline 0000 \quad 0 \quad (+0) \end{array} \quad C=1, V=0$$

$$\begin{array}{r} 1010 \quad 10 \quad (-6) \\ + 1101 \quad 13 \quad (-3) \\ \hline 0111 \quad 7 \quad (+7) \end{array} \quad C=1, V=1$$

RESTAS:

$$\begin{array}{r} 0110 \quad 6 \quad (+6) \\ - 0001 \quad 1 \quad (+1) \\ \hline 0101 \quad 5 \quad (+5) \end{array} \quad C=0, V=0$$

$$\begin{array}{r} 1000 \quad 8 \quad (-8) \\ - 0001 \quad 1 \quad (+1) \\ \hline 0111 \quad 7 \quad (+7) \end{array} \quad C=0, V=1$$

$$\begin{array}{r} 0011 \quad 3 \quad (+3) \\ - 0100 \quad 4 \quad (+4) \\ \hline 1111 \quad 15 \quad (-1) \end{array} \quad C=1, V=0$$

$$\begin{array}{r} 0110 \quad 6 \quad (+6) \\ - 1110 \quad 14 \quad (-2) \\ \hline 1000 \quad 8 \quad (-8) \end{array} \quad C=1, V=1$$

NÚMEROS ENTEROS

INDICADORES DE LA ALU

- Toda CPU posee un registro con *indicadores* de la ALU: bits que expresan qué ha sucedido en la última operación. Típicamente esos indicadores son:
 - **Carry (C)**: Acarreo desde el bit más significativo (tras una suma o resta).
 - **Overflow (V)**: Se ha producido un rebose ($V=1$) o no ($V=0$). Tras una suma o resta sólo tiene sentido si usamos aritmética con signo.
 - **Zero (Z)**: El resultado ha sido cero ($Z=1$) o no ($Z=0$).
 - **Negative (N)**: Copia del bit más significativo del resultado. Sólo tiene sentido si usamos aritmética con signo.

NÚMEROS ENTEROS

COMPARACIONES ENTRE NÚMEROS

- Toda CPU cuenta con una instrucción de *comparación* entre dos números. Simplemente realiza una resta pero no se almacena el resultado. Su único objetivo es afectar los indicadores de la ALU.
- Tras una instrucción de comparación siempre se programa una instrucción *condicional*, típicamente de *salto* (modificación del PC). Esa instrucción evalúa cierta *condición* y sólo realiza su operación si resulta cierta.
- Este mecanismo es la base para poder tomar decisiones durante la ejecución de un programa, y así implementar estructuras iterativas y de decisión.

NÚMEROS ENTEROS

EJEMPLO: CONDICIONES EVALUABLES POR EL 68000

Condición	Significado	Evaluación
T	True	1
F	False	0
NE	Not Equal	\bar{Z}
EQ	Equal	Z
CC/HS	Carry Clear / Higher or Same	\bar{C}
CS/LO	Carry Set / Lower	C
HI	Higher	$\bar{C} \cdot \bar{Z}$
LS	Lower or Same	C + Z
VC	Overflow Clear	\bar{V}
VS	Overflow Set	V
PL	Plus	\bar{N}
MI	Minus	N
GE	Greater or Equal	$\bar{V} \cdot \bar{N} + V \cdot N$
LT	Less Than	$V \cdot \bar{N} + \bar{V} \cdot N$
GT	Greater Than	$(\bar{V} \cdot \bar{N} + V \cdot N) \cdot \bar{Z}$
LE	Less or Equal	$(V \cdot \bar{N} + \bar{V} \cdot N) + Z$

