

Flags

Sitio: Agencia de Habilidades para el Futuro

Curso: Lógica Computacional 1°D

Libro: Flags

Imprimido por: RODRIGO PINTO

Día: martes, 26 de noviembre de 2024, 18:37

Tabla de contenidos

1. Registros Flag

1.1. Introducción a los Flags

1.2. Flags: ¿para qué sirven?

1.3. Banderas aritméticas

1.4. Flag Z

1.5. Flag N

1.6. Flag C

1.7. Flag V

1.8. Otros flags

1. Registros Flag



Comencemos con un disparador, una pregunta...

Comencemos con un disparador, una pregunta:

¿Cómo podemos determinar, si dos números son iguales?

Nosotros tenemos las condiciones para evaluar si dos números son iguales, pero la computadora, en sí, no lo sabe. Le podemos pasar dos valores, A y B, y nosotros a simple vista podemos resolver el "algoritmo" o la duda que se plantea, pero la máquina, los va a tomar como una variable más, y va a tener que requerir a un circuito, o cuenta, para resolver la consulta.

¿Cómo lleva a cabo ese proceso?

En el caso de nuestro disparador, por medio de una resta. Si le doy dos valores, lo que va a hacer es restar un valor a otro, y lo que va a hacer a continuación, es evaluar el resultado: Si el resultado es 0, va a entender que sí, que ambos valores son iguales, dado que dos valores con la misma magnitud, van a "pesar" lo mismo. Si el resultado es distinto de 0, quiere decir que ambos valores son distintos.

1.1. Introducción a los Flags

Los FLAGS son un registro de 16 bits, de los cuales 9 sirven para indicar el estado actual de la máquina y el resultado del procesamiento. Podemos encontrar los siguientes flags o banderas:

OF (overflow, desborde)

DF (dirección)

IF (interrupción)

TF (trampa)

SF (signo)

ZF (zero, cero)

AF (acarreo auxiliar)

PF (paridad)

CF (carry, acarreo).

1.2. Flags: ¿para qué sirven?

Cuando estás programando, escribís líneas de código que hacen determinadas acciones, pero que desconocemos cuál es el proceso interno que llevan a cabo para poder actuar en consecuencia.



Supongamos que tenemos un código que nos pide que recorramos un arreglo con X cantidad de valores y los acumulemos en un registro. Ese recorrido tendrá lugar hasta que, supongamos, el valor que aparezca, sea IGUAL A 0.

Bien, nosotros vamos a desarrollar ese código y le vamos a poner una **CONDICIÓN** de fin, que el programa va a evaluar.

Esa evaluación la hace de manera "interna", y la hace por medio de los llamados **FLAGS**, o Banderas.

Los Flags no son más que bits que el procesador establece de modo automático acorde al resultado de cada operación realizada.

Sus valores le van a permitir tomar decisiones tales como:

- Realizar o no una transferencia de control.
- Determinar relaciones entre números (mayor, menor, igual)
- Etc.

1.3. Banderas aritméticas

A continuación, vamos a conocer los Flags que corresponden a las condiciones aritméticas:

Z (cero): vale 1 si el resultado de la operación es 0, vale 0 si el resultado de la operación es distinto de 0.

C (carry): en la suma vale 1 si hay acarreo del bit más significativo; en la resta vale 1 si hay borrow ("pido prestado") hacia el bit más significativo.

Cuando una operación involucra números **sin signo**, C=1 indica una condición fuera de rango.

V (overflow): significa que el resultado no se puede representar.

N (negative): significa que el primer bit del resultado vale 1.

1.4. Flag Z

Para que quede más claro el desarrollo del tema, les paso ejemplos de cada Flag y su activación. Arrancamos con el flag Z, la bandera que se "levanta" cuando TODOS LOS BITS del resultado son 0.

$$\begin{array}{r} 111 \\ - 111 \\ \hline 000 \end{array}$$

$$Z = 1$$

$$\begin{array}{r} 111 \\ + 001 \\ \hline 000 \end{array}$$

$$Z = 1$$

$$\begin{array}{r} 101 \\ + 001 \\ \hline 110 \end{array}$$

$$Z = 0$$

1.5. Flag N

Como ya vimos en los distintos sistemas numéricos que estudiamos en las semanas anteriores (signo magnitud, Ca1, Ca2), el indicador de que una cadena representa un valor negativo, es que el primer bit del resultado, el más significativo, vale 1.

$$\begin{array}{r} 100 \\ - 001 \\ \hline \end{array}$$
$$011$$
$$N = 0$$
$$\begin{array}{r} 101 \\ + 001 \\ \hline \end{array}$$
$$110$$
$$N = 1$$
$$\begin{array}{r} 011 \\ + 011 \\ \hline \end{array}$$
$$110$$
$$N = 1$$

Recordando lo que dijimos anteriormente con respecto a los sistemas numéricos...

¿Qué significa en SM que una cadena empiece con 1?

Significa que es un número negativo.

¿Qué significa en Ca2 que una cadena empiece con 1?

Significa que es un número negativo.

¿Qué significa en BSS que una cadena empiece con 1?

Nada... simplemente representa a un número cuyos bits, en su totalidad, representan la magnitud de este y el valor es NO NEGATIVO.

1.6. Flag C

Al terminar, me llevé, o pedí, uno

$$\begin{array}{r} 011 \\ - 101 \\ \hline \end{array}$$

110

C = 1

$$\begin{array}{r} 101 \\ + 101 \\ \hline \end{array}$$

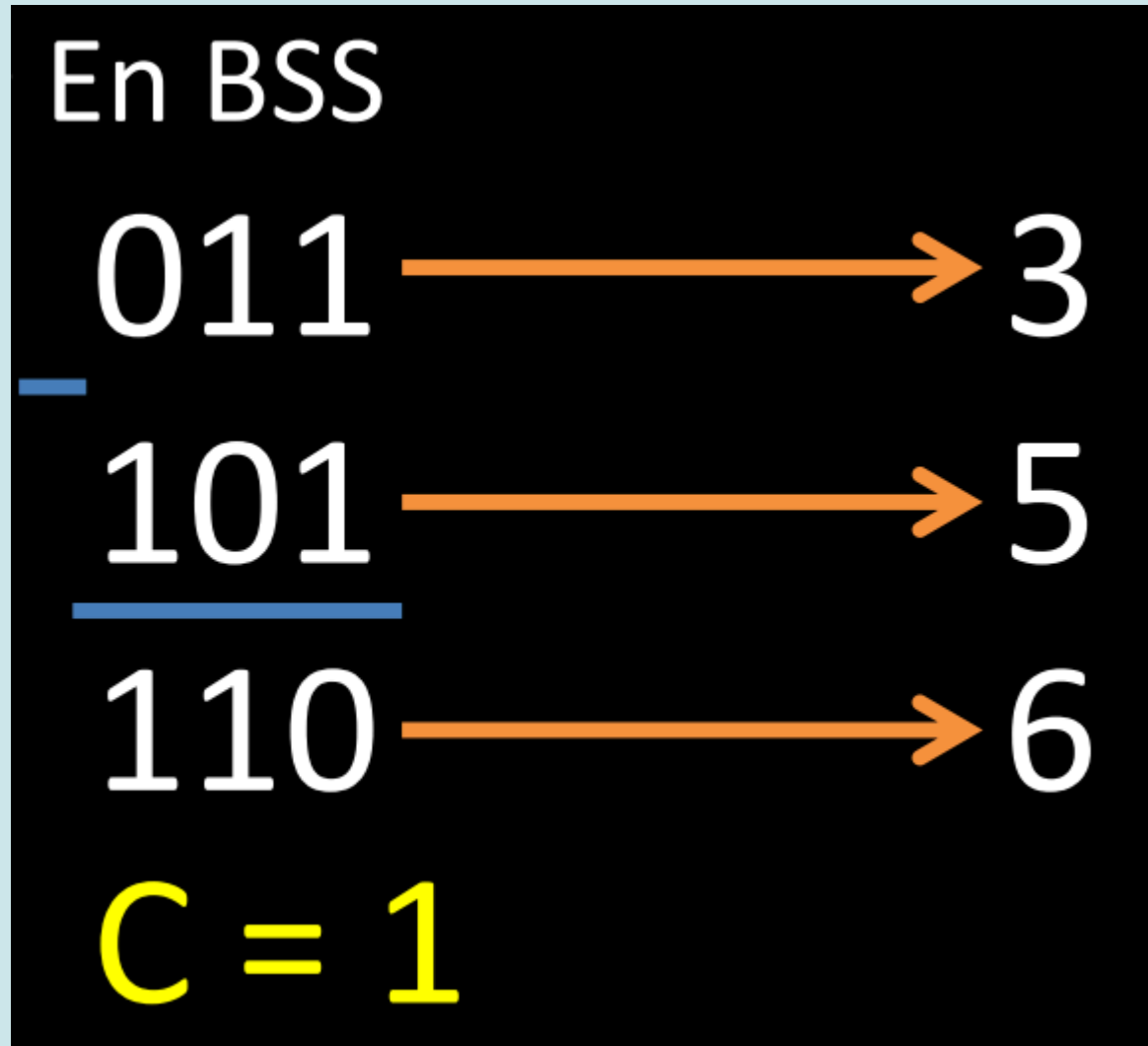
010

C = 1

$$\begin{array}{r} 011 \\ + 011 \\ \hline \end{array}$$

110

C = 0



En BSS

$$\begin{array}{r} 101 \longrightarrow 5 \\ + 101 \longrightarrow 5 \\ \hline 010 \longrightarrow 2 \\ C = 1 \end{array}$$

Cuando el Carry se "enciende" en BSS, significa que el resultado no se puede representar.

En CA2

$$\begin{array}{r} 111 \\ + 111 \\ \hline 110 \end{array}$$

Diagram illustrating binary addition in 2's complement (CA2) for the values 111 and 111. The result is 110.

Each 3-bit value is mapped to its decimal equivalent:

- 111 → -1
- 111 → -1
- 110 → -2

Que el Carry se "encienda" en Ca2, no **significa nada...** Ya que se puede encender estan el resultado OK o no.

1.7. Flag V

El flag V, es el flag de desbordamiento, o desborde, u overflow. Indica en Ca2 que el resultado no se puede representar.

Ejemplo: sumo dos números positivos y me da uno negativo

The diagram shows a binary addition on a black background. The first row is 010 with an orange arrow pointing to the decimal value 2 . The second row is $+ 010$ with an orange arrow pointing to the decimal value 2 . A blue horizontal line is drawn under the second row. The third row shows the result 100 with an orange arrow pointing to the decimal value -4 . Below the result, the text $V = 1$ is displayed in yellow.

Sumo dos números negativos y me da uno positivo

Diagram illustrating binary addition with carry:

| | | | | | |
|--------------|---|---|---|----|----|
| 1 | 0 | 0 | → | -4 | |
| + | 1 | 0 | 0 | → | -4 |
| <hr/> | | | | | |
| 0 | 0 | 0 | → | 0 | |
| V = 1 | | | | | |

Resto a un negativo, un positivo y me da positivo

$$\begin{array}{r} 110 \\ - 011 \\ \hline 011 \end{array}$$

$V = 1$

Resto a un positivo, un negativo y me da negativo

The diagram illustrates a binary subtraction operation on a black background. It shows three rows of binary numbers. The first row is 001 with an orange arrow pointing to the decimal value 1 . The second row is 100 with an orange arrow pointing to the decimal value -4 . A blue horizontal line is drawn under the 100 . The third row is 101 with an orange arrow pointing to the decimal value -3 . Below these, the text $V = 1$ is displayed in yellow.

$$\begin{array}{r} 001 \\ - 100 \\ \hline 101 \end{array}$$

$V = 1$

1.8. Otros Flags

Les dejo un listado de flags que también son interesantes para conocer.

- **DF (direction flag):** controla la selección de incremento o decremento de los registros SI y DI en las operaciones con cadenas de caracteres (1=decremento y 0=incremento). La bandera DF se controla, se controla con las instrucciones STD y CLD
- **IF (interrupt flag):** controla el disparo de las interrupciones (1= habilita las interrupciones; 0=deshabilita las interrupciones). La interrupción no enmascarable es la única que no puede ser bloqueada por esta bandera. El estado de la bandera IF se controla con las instrucciones STI y CLI.
- **TF (trap flag):** permite la operación del procesador en modo de depuración
- **SF (sign flag):** contiene el signo resultante de una operación aritmética (0=positivo, 1=negativo).
- **AF(auxiliar carry flag):** contiene el acarreo del bit 3. Esta bandera se prueba con las instrucciones DAA y DAS para ajustar el valor de AL después de una suma o resta BCD.
- **PF (parity flag):** indica si el número de bits 1, del byte menos significativos de una operación, es par (0=número de bits 1 es impar; 1=número de bits 1 es par).
- **CF (carry flag):** contiene el acarreo del bit de mayor orden después de una operación aritmética; también almacena el contenido del último bit en una operación de desplazamiento o de rotación.

