UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Representación de números enteros: el convenio “exceso Z”

Apellidos, nombre	Martí Campoy, Antonio (amarti@disca.upv.es)
Departamento	Informàtica de Sistemes i Computadors
Centro	Escola Tècnica Superior d’Enginyeria Informàtica

UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

1 Resumen de las ideas clave

En este artículo se trata la problemática de la representación de los números enteros en los computadores. Así mismo, se presentará una posible solución a este problema, que recibe el nombre de representación en exceso Z. Los conocimientos previos que necesitas para abordar este artículo se presentan en la tabla 1.

Tabla 1. Conocimientos previos

Conocimientos previos
1. Sistemas de numeración posicionales
2. Sistema de numeración binario
3. Cambios de base, especialmente binario
4. Aritmética básica en base 2

2 Objetivos

Una vez acabes de leer este artículo docente y reproduzcas los ejemplos presentados, deberás ser capaz de **representar** números enteros en binario **aplicando** el convenio llamado exceso Z. Además podrás **calcular** el rango de representación para un tamaño de bits determinado. También serás capaz de **realizar** operaciones aritméticas de suma y resta de números enteros en binario y de extensión de signo utilizando la representación en exceso Z. Por último, podrás **razonar** sobre las ventajas y desventajas de este convenio de representación de números enteros.

3 Introducción

En la vida cotidiana los números enteros se representan mediante los 10 símbolos (del 0 al 9) de la base decimal, junto con los símbolos "+" y "-" para identificar a los números positivos y negativos, respectivamente.

A la hora de representar números enteros en un computador (para almacenarlos, operarlos o comunicarlos) el problema que surge es que en los circuitos digitales sólo se pueden utilizar dos valores, normalmente representados por los símbolos 0 y 1. No cabe la posibilidad de representar un tercer y cuarto símbolo para distinguir un número positivo de otro negativo.

Así surge la necesidad de crear y definir convenios para codificar el signo de un número entero utilizando únicamente los símbolos 0 y 1 disponibles en los circuitos digitales.

Antes de explicar el convenio exceso Z, objeto de este artículo, recordarte que los números se almacenan en circuitos digitales llamados registros, y que su longitud es fija. Es decir, cuando hablemos de un número entero representado en binario y en exceso Z deberemos indicar el número total de bits utilizados.

4 El convenio exceso Z

Este convenio recibe su nombre porque se utiliza una constante, Z, para sumarla a los números que se quieren representar. El valor de esta constante es arbitrario, es decir, se puede utilizar la que se quiera. Pero en la práctica se utilizan valores de Z relacionados con el número de bits utilizados para la representación, siendo los más comunes $Z = 2^{n-1}$ y $Z = 2^{n-1} -1$. Así, por ejemplo, si utilizamos 8 bits podemos usar valores de Z = 128 o Z = 127. Con estos valores se consigue un rango de representación casi simétrico como veremos a continuación.

4.1 Definición

En este convenio todos los números se representan de la misma forma, no importa si son positivos o negativos. Si consideramos que usamos n bits y un exceso Z, tenemos que:

- Un número entero A se representa por el binario natural de A + Z, utilizando n bits.

Ejemplo: utilizando 8 bits (n = 8) y $Z = 2^{n-1} -1 = 127$ representa el número +31 siguiendo el convenio exceso Z.

Realizamos la suma de +31 + 127 = +158 y lo representamos en binario natural con 8 bits, completando con ceros los bits de mayor peso si fuera necesario: 158 = 10011110_2. De este modo tenemos que +31 en exceso 127 se representa por:

$$31_{10} = 10011110_2$$

Ejemplo: utilizando 8 bits (n = 8) y $Z = 2^{n-1} -1 = 127$ representa el número -34 siguiendo el convenio exceso Z.

Realizamos la suma de -34 + 127 = +93 y lo representamos en binario natural con 8 bits, completando con ceros los bits de mayor peso si fuera necesario: 93 = 01011101_2. De este modo tenemos que -34 en exceso 127 se representa por:

$$-34_{10} = 01011101_2$$

Ejemplo: obtén el valor decimal correspondiente a 010010_2 sabiendo que está representado en exceso 31 utilizando 6 bits (n = 6).

$$010010_2 = 18_{10},$$

$$18_{10} - 31_{10} = -13_{10}$$

Por lo que 010010_2 en exceso 31 representa el valor -13_{10}

Ejemplo: obtén el valor decimal correspondiente a 101000_2 sabiendo que está representado en exceso 31 utilizando 6 bits (n = 6).

$$101000_2 = 40_{10},$$

$$40_{10} - 31_{10} = +9_{10}$$

Por lo que 101000_2 en exceso 31 representa el valor $+9_{10}$

Quiero que recuerdes una cosa importante. En exceso Z el bit de mayor peso, es decir el de la izquierda, puede, pero no siempre, representar el signo del número. En otros convenios de representación de enteros esto es siempre cierto, pero en la representación en exceso Z no siempre lo es, por lo que mejor no te confíes.

4.2 Rango

El rango de un sistema o convenio de representación es el conjunto de valores diferentes que pueden representarse.

La representación en exceso Z utiliza binario natural, por lo que la representación más pequeña posible es 000···000, que corresponde con el número entero 0 – Z = -Z. La representación más grande posible es 111···111 que corresponde con el entero (2^n-1) – Z. De esta forma el rango queda:

$$\text{Rango en binario:} \quad [000\cdots000\ , \quad 111\cdots111]$$
$$\text{Rango en decimal:} \quad [-Z\ , \quad +(2^n-1)-Z]$$

Podemos ver un ejemplo con 4 bits y Z = 7:

0000_2	-7	+1	1000_2
0001_2	-6	+2	1001_2
0110_2	-1	+7	1110_2
0111_2	0	+8	1111_2

Si te fijas, verás que el valor entero 0 se representa por el binario natural correspondiente a Z. También puedes ver que sólo hay un cero.

Y otra cosa importante, los números enteros están todos ordenados. Es decir, el negativo con mayor magnitud tiene la representación menor en binario natural, y el positivo con mayor magnitud tiene la mayor representación en binario natural. Esto hace muy útil este convenio de representación cuando lo que se quiere es comparar números enteros.

4.3 Suma y resta

Para sumar dos números enteros representados en exceso Z se opera siguiendo las reglas de suma de binario natural, independientemente del signo de los operandos. Sin embargo, tienes que tener en cuenta que al sumar dos números representados en exceso Z, el resultado estará expresado en exceso 2*Z. Esto es fácil de ver: sean A y B dos números enteros representados en exceso Z, es decir A = a+Z y B = b+Z. Si realizo la operación A + B lo que estamos haciendo en realidad es a + Z + b +Z = a + b + Z + Z.

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Esto tiene dos implicaciones. La primera es que al realizar la suma en binario tenemos que mantener el bit de acarreo final, porque puede ser significativo. La segunda es que una vez realizada la suma es necesario restar Z para obtener el resultado correctamente representado en exceso Z.

La resta también se realiza usando las reglas de la resta en binario natural (intercambiando el orden de los operandos si es necesario), pero sucede exactamente lo contrario que en la suma. Al hacer A – B, lo que estamos haciendo es a + Z – (b +Z) = a + Z – b – Z = a – b.

Es decir, el resultado no está en exceso Z, por lo que hay que sumarle a posteriori Z.

Pero en la práctica esta característica de la resta es una ventaja, ya que el exceso Z se suele utilizar para comparar números, y la resta nos dice exactamente cuántas unidades de diferencia hay entre el minuendo y el sustraendo. Por eso este convenio no se utiliza para hacer aritmética con enteros, pero sí para compararlos.

4.1 Desbordamiento en la suma y resta

Al realizar una suma o resta de números enteros es posible que el resultado exceda el rango de representación. En este caso se dice que no hay resultado o que el resultado no es representable.

Con operandos representados en exceso Z se produce desbordamiento si al corregir el resultado (restar Z tras una suma o sumar Z tras una resta) se produce o se mantiene un uno en el bit de acarreo final.

4.2 Extensión de signo

En algunos casos es necesario operar datos con diferentes tamaños. Para aumentar el número de bits con que se representa un dato se realiza la operación llamada extensión de signo.

En el caso de la representación en exceso Z no es posible realizar ninguna extensión de signo. Si es necesario cambiar el número de bits utilizados para la representación, hay que eliminar el exceso actual y añadir el nuevo exceso.

5 Ejercicios

A continuación tienes unos pocos ejercicios. Es muy conveniente que cojas lápiz y papel y los resuelvas. Recuerda que estas aprendiendo, por lo que puedes, y aún diría más, debes consultar las secciones anteriores de este documento para resolver los ejercicios. También tienes las soluciones de los ejercicios, pero te pido encarecidamente que no las mires hasta que no hayas intentado resolver todos los ejercicios.

5.1 Enunciados

1. Representa el número -63_{10} en binario exceso 127 con 8 bits.
2. Representa el número $+93_{10}$ en binario exceso 127 con 8 bits.
3. Indica la representación decimal de 10110011_2 sabiendo que está representado en exceso 127 con 8 bits.

4. Indica la representación decimal de 01101001_2 sabiendo que está representado en exceso 127 con 8 bits.
5. ¿Cuál es el rango de representación de exceso 1023 con 11 bits? Expresa el rango en decimal.

5.2 Soluciones

1. Representa el número -63_{10} en binario exceso 127 con 8 bits. Sol: 01000000_2
2. Representa el número $+93_{10}$ en binario exceso 127 con 8 bits. Sol: 11011100_2
3. Indica la representación decimal de 10110011_2 sabiendo que está representado en exceso 127 con 8 bits. Sol: 52_{10}
4. Indica la representación decimal de 01101001_2 sabiendo que está representado en exceso 127 con 8 bits. Sol: -22_{10}
5. ¿Cuál es el rango de representación de exceso 1023 con 11 bits? Expresa el rango en decimal. Sol: [-1023, +1024]

6 Conclusiones

Los circuitos digitales sólo pueden almacenar dos símbolos, por lo que es necesario establecer un acuerdo o convenio para utilizar estos dos símbolos, el 0 y el 1, para representar el signo de los números enteros. El convenio llamado representación en exceso Z es sencillo, pero su aritmética no es eficiente ya que requiere dos operaciones por cada una que se quiere hacer.

Sin embargo, es muy útil cuando lo que se busca es comparar valores y determinar la distancia entre ellos. Así, por ejemplo, se utiliza para representar los exponentes en el estándar IEEE754 para la representación de números reales.

7 Bibliografía

7.1 Libros:

[1] Pedro de Miguel Anasagasti. "*Fundamentos de los computadores*", 9ª ed. Madrid, Thomson-Paraninfo. 2004, 2007

[2] John F. Wakerly. *"Diseño digital : principios y prácticas".* Madrid. Pearson Educación. 2001

7.2 Recursos electrónicos:

[3] Martí Campoy, Antonio. "*Representación de enteros: Exceso Z*", Universitat Politècnica de València, 2011. http://hdl.handle.net/10251/10112