Representació de nombres enters: el conveni “complement a dos”

Cognoms, nom	Martí Campoy, Antonio (amarti@disca.upv.es)
Departament	Informàtica de Sistemes i Computadors
Centre	Escola Tècnica Superior d’Enginyeria Informàtica

UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

1 Resum de les idees clau

En aquest article es parla de la problemàtica de la representació de nombres enters en els computadors. També es presentarà una possible solució a aquest problema, que rep el nom de "representació en complement a dos". Els coneixements previs necessaris per treballar aquest article es presenten en la taula 1.

Taula 1. Coneixement previs

Coneixements previs
1. Sistemes de numeració posicionals
2. Sistema de numeració binari
3. Canvis de base, especialment binari
4. Aritmètica bàsica en base 2

2 Objectius

Una vegada acabes de llegir aquest article docent i reproduïsques els exemples presentats, hauràs de ser capaç de **representar** nombres enters en binari **aplicant** el conveni anomenat "complement a dos". A més a més, podràs **calcular** el rang de representació per a una grandària de bits determinada. També seràs capaç de **realitzar** operacions aritmètiques de suma i resta de nombres enters en binari i d'extensió de signe utilitzant la representació complement a dos. Finalment, podràs **raonar** sobre els avantatges i desavantatges d'aquest conveni de representació de nombres enters.

3 Introducció

En la vida quotidiana els nombres enters es representen mitjançant els deu símbols (del 0 al 9) de la base decimal, juntament amb els símbols "+" i "-" per identificar els nombres positius i negatius, respectivament.

Quan es representen nombres enters en un computador (per emmagatzemar-los, operar-los o comunicar-los) el problema que hi apareix és que en els circuits digitals només es poden utilitzar dos valors, normalment representats pels símbols 0 i 1. No hi ha cap possibilitat de representar un tercer símbol ni un quart per distingir-ne els nombres positius dels nombres negatius.

D'aquesta manera sorgeix la necessitat de crear i definir convenis per a codificar el signe d'un nombre enter emprant únicament els símbols 0 i 1 disponibles en els circuits digitals.

Abans de presentar-te el conveni complement a dos, objecte d'aquest article docent, vull recordar-te que els nombres s'emmagatzemen en circuits digitals anomenats registres, i que la grandària és fixa. És a dir, quan parlem d'un nombre enter representat en binari i complement a dos cal indicar el nombre total de bits utilitzats.

4 El conveni complement a dos

El nom d'aquest conveni ve donat perquè s'utilitza l'operació aritmètica de complement a dos per tal de representar els nombres negatius. Per això és molt **important** que no confongues l'operació aritmètica de complement a dos (**fem** el complement a dos a un nombre) amb la representació d'un nombre enter **en** complement a dos (codifiquem o representem un nombre seguint el conveni).

També pots trobar referències a aquest conveni com a Ca2 i C'2, entre d'altres.

4.1 Definició

En aquest conveni es diferencia la manera en què es representa un nombre enter positiu d'un de negatiu. Considerem que utilitzem *n* bits per representar els nombres enters. En el conveni, un acord arbitrari, es diu que:

- Si el nombre és **positiu,** es representa la seua magnitud amb n-1 bits, i s'afegeix un 0 a l'esquerra:

0	*Magnitud* (n-1 bits)

- Si el nombre és **negatiu,** es representa seguint el procediment següent. Es representa el seu equivalent positiu, utilitzant n-1 bits per a la magnitud i afegint un zero a l'esquerra, de la mateixa manera com s'ha descrit abans. Una vegada tenim la representació de l'equivalent positiu, li fem el complement a dos, i aquesta serà la representació del nombre negatiu:

Ca2(| 0 | Magnitud (n-1 bits) |)

Dues coses importants. La primera és recordar que el Ca2 d'un nombre es pot calcular o fer invertint-ne els bits, és a dir, canviant els uns per zeros i els zeros per uns, i sumant-li u a continuació. El *carry* final, siga zero o u es descarta.

La segona cosa important és que, en utilitzar la representació **en** complement a dos, el bit de major pes, el de l'esquerra, indica el signe del nombre i rep el nom de **bit de signe**.

Exemple. Utilitzant 8 bits (n = 8), representa el nombre +21 seguint el conveni de representació en complement a dos.

En primer lloc es converteix la magnitud o valor absolut, 21, a binari natural amb n-1=7 bits, i es completen amb zeros els bits de major pes si és necessari:

$$21_{10} = 0010101_2$$

Com que el nombre que volem representar és positiu, afegim un zero a l'esquerra:

$$+21_{10} = \mathbf{0}0010101_2$$

Exemple. Utilitzant 8 bits (n = 8), representa el nombre -26, seguint el conveni de representació en complement a dos.

Com que el nombre que s'ha de representar és negatiu, necessitem obtenir en primer lloc la representació del seu equivalent positiu. Convertim la magnitud o valor absolut, 26, a binari natural amb n-1=7 bits, i completem amb zeros els bits de major pes si és necessari:

$$26_{10} = 0011010_2$$

Ara afegim un zero a l'esquerra per obtenir la representació de +26:

$$+26_{10} = \mathbf{0}0011010_2$$

Però, el que realment volem fer és representar el nombre negatiu -26, per la qual cosa fem el complement a dos a +26:

$$\text{Ca2}\ (+26_{10}) = \text{Ca2}\ (\mathbf{0}0011010_2) = \mathbf{1}1100101_2 + 1 = \mathbf{1}1100110 = -26_{10}$$

(recorda, en fer la suma en el Ca2 es descarta el *carry* final.)

Exemple. Obtín el valor decimal corresponent a 010010_2 i 101000_2 tenint en compte que estan representats en complement a dos utilitzant 6 bits (n = 6).

$010010_2 \longrightarrow$ Com que el bit de major pes (bit de signe) és 0, sabem que es tracta d'un nombre positiu. Seguint el conveni, retirem el zero de major pes i ens queda la magnitud $10010_2 = 18_{10}$ i, per tant, $010010_2 = +18_{10}$.

$101000_2 \longrightarrow$ Com que el bit de major pes (bit de signe) és 1, sabem que es tracta d'un nombre negatiu. En aquest cas aprofitem que l'operació de complement a dos és reversible (Ca2(Ca2(x)) = x) per a obtenir l'equivalent positiu, i d'aquesta manera podem obtenir la magnitud:

$$\text{Ca2}(101000_2) = 010111_2 + 1 = 011000_2 = +24_{10}$$

Eliminem el bit de signe i ens queda la magnitud $11000_2 = 24_{10}$

Per tant, en complement a dos tenim que $101000_2 = -24_{10}$.

Vull que recordes dues coses importants. La primera és que no sabem llegir nombres negatius representats en complement a dos i, per això, hem de trobar l'equivalent positiu, que sí que el sabem llegir. I la segona, que en aquest procés no pots oblidar-te d'indicar el signe al final de la conversió.

4.2 Rang

El rang d'un sistema o conveni de representació és el conjunt de valors diferents que poden representar-se. Estudiarem el rang separadament per als nombres positius i per als nombres negatius:

Positius		Negatius	
$000\cdots000_2$	+0	-0	$000\cdots000_2$
$000\cdots001_2$	+1	-1	$111\cdots111_2$
$011\cdots110_2$	$+(2^{n-1}-2)$	$-(2^{n-1}-1)$	$100\cdots001_2$
$011\cdots111_2$	$+(2^{n-1}-1)$	$-(2^{n-1})$	$100\cdots000_2$

En aquesta taula hi ha alguns aspectes interessants. Per a veure-ho més senzillament treballarem uns exemples amb 4 bits:

Positius		Negatius	
0000_2	+0	-0	0000
0001_2	+1	-1	1111_2
0110_2	+6	-7	1001_2
0111_2	+7	-8	1000_2

Aquests són els aspectes als quals cal parar atenció:

- Només hi ha un zero i és positiu. Si fem Ca2(0000) = 1111 + 1 = 0000 (recorda, el *carry* final es descarta). És a dir, l'equivalent negatiu del zero és aquest mateix.
- El nombre 1000 és negatiu, i si busquem l'equivalent positiu trobem que Ca2(1000) = 0111 +1 = 1000, que torna a ser negatiu. Quan sàpigues sumar serà fàcil demostrar que aquest valor correspon a -8. I si parlem de *n* bits, correspon a $-(2^{n-1})$.
- Els nombres negatius no estan ordenats de la mateixa manera que si foren naturals.

Amb tot això, podem concloure que el rang de representació emprant *n* bits en complement a dos és:

$$
\begin{array}{rccccc}
\text{Rang en binari:} & [100\cdots000, & 111\cdots111, & 000\cdots000, & 011\cdots111] \\
\text{Rang en decimal:} & [-(2^{n-1}), & -1, & 0, & +(2^{n-1}-1)]
\end{array}
$$

El rang és asimètric, és a dir, inclou un valor més per als nombres negatius que per als positius, i només inclou una representació per al zero. També, com hem dit abans, els nombres negatius no estan en l'ordre "natural", i això en complica la interpretació per part dels éssers humans.

4.3 Suma i resta

L'operació de suma de nombres representats en complement a dos es realitza utilitzant les regles de la suma de binari natural, independentment del signe dels operands, i **descartant** el *carry* final. Això vol dir que tant fa que en sumes dos de positius o dos de negatius, o un de positiu i un altre de negatiu, simplement se sumen. I a més a més, el resultat de la suma es troba directament representat en complement a dos. Aquest és el principal avantatge d'aquest conveni de representació i la raó per la qual el 100% dels sistemes informàtics l'utilitzen per a la representació de nombres enters.

L'operació de resta es porta a terme mitjançant una suma, en la qual es canvia el signe al subtrahend. És a dir, A - B = A + (-B). I com hem vist abans, canviar el signe a un nombre representat en complement a dos és molt senzill, tan sols cal fer-li el complement a dos.

Com a exemple, es presenta a continuació una suma i una resta de nombres representats en binari complement a dos amb 4 bits (i els equivalents en decimal):

Binari Ca2 amb 4 bits	0100 + 1101 **0001**	El *carry* final és 1, però es descarta
Decimal	+4 + -3 +1	

Binari Ca2 amb 4 bits	1101 - 0010	Fem Ca2(0010) i es transforma en suma	1101 + 1110 **1011**
Decimal	-3 - +2 -5	Ca2(**1011**) = 0100+1 = 0101 **1011** = -5_{10}	

4.4 Desbordament en la suma i la resta

En fer una suma de nombres enters és possible que el resultat excedisca el rang de representació. En aquest cas es diu que no hi ha resultat o que el resultat no és representable.

Amb operands representats en complement a dos es produeix desbordament o sobreeiximent en fer una suma si els valors de l'últim i penúltim bits de *carry* són diferents. Una senzilla porta lògica or-exclusiva (XOR) permet detectar la condició de desbordament. A continuació tens un exemple on es produeix desbordament. En aquest exemple els *carry* apareixen amb cursiva:

Binari Ca2 amb 4 bits	*1000* 1101 + 1010 0111	Es produeix desbordament i no hi ha resultat	-3 + -9 = +7?

4.5 Extensió de signe

En alguns casos és necessari operar amb dades de diferent grandària. Per a augmentar el nombre de bits amb què es representa una dada es realitza l'operació anomenada "extensió de signe".

En el cas de la representació en complement a dos, l'extensió de signe es realitza replicant el bit de signe.

Exemple. Tenint els nombres enters 0010_2 i 1110_2 representats en complement a dos amb 4 bits, fes l'extensió de signe per a representar-los amb 8 bits.

$$\mathbf{0}010_2 = \mathbf{00000}010_2$$
$$\mathbf{1}110_2 = \mathbf{11111}110_2$$

5 Exercicis

A continuació tens uns quants exercicis. És molt important que agafes paper i llapis i els resolgues. Recorda que estàs aprenent i que, per tant pots, i encara diria més, cal que consultes les seccions anteriors d'aquest document per resoldre els exercicis. També tens les solucions dels exercicis però et demane amb fermesa que tractes de resoldre tots els exercicis abans, i no les consultes.

5.1 Enunciats

1. Representa el nombre -66_{10} en binari complement a dos amb 8 bits.
2. Representa el nombre $+99_{10}$ en binari complement a dos amb 8 bits.
3. Indica la representació decimal de 10110001_2 tenint en compte que està representat en complement a dos amb 8 bits.
4. Indica la representació decimal de 00101001_2 tenint en compte que està representat en complement a dos amb 8 bits.
5. Quin és el rang de representació en complement a dos amb 8 bits? Expressa'n el rang en decimal.
6. Tenint els nombres enters representats en complement a dos amb 8 bits A = 10000101_2 i B = 01011011_2, realitza les operacions A + B, A – B i B – A, i indica si el resultat és correcte o no ho és.
7. Realitza l'extensió de signe a 16 bits de 11000101_2 tenint en conte que esta representat en complement a dos amb 8 bits.
8. Realitza l'extensió de signe a 16 bits de 01110110_2 tenint en compte que està representat en complement a dos amb 8 bits.

5.2 Solucions

1. Representa el nombre -66_{10} en binari complement a dos amb 8 bits. Sol: 10111110_2
2. Representa el nombre $+99_{10}$ en binari complement a dos amb 8 bits. Sol: 01100011_2
3. Indica la representació decimal de 10110001_2 tenint en conte que esta representat en complement a dos amb 8 bits. Sol: -79_{10}
4. Indica la representació decimal de 00101001_2 tenint en conte que esta representat en complement a dos amb 8 bits. Sol: $+41_{10}$
5. Quin és el rang de representació en complement a dos amb 8 bits? Expressa el rang en decimal. Sol: $[-2^7, +(2^7-1)] = [-128, +127]$
6. Tenint els nombres enters representats en complement a dos amb 8 bits $A = 10000101_2$ i $B = 01011011_2$, realitza les operacions A + B, A – B i B – A, indicant si el resultat és correcte o no ho és. Sol: A + B = 11100000; A - B = desborda; B –A = desborda.
7. Realitza l'extensió de signe a 16 bits de 11000101_2 tenint en conte que esta representat en complement a dos amb 8 bits. Sol: 1111111111000101_2
8. Realitza l'extensió de signe a 16 bits de 01110110_2 tenint en conte que esta representat en complement a dos amb 8 bits. Sol: 0000000001110110_2

6 Conclusions

Els circuits digitals només poden emmagatzemar dos símbols, per la qual cosa és necessari establir un acord o conveni per a utilitzar aquests dos símbols, el 0 i l'1, per a representar el signe d'un nombre enter. El conveni anomenat "representació en complement a dos" és senzill i presenta una aritmètica molt senzilla i útil. Construir un únic circuit sumador/restador per a nombres representats en complement a dos és molt senzill. Aquesta és una de les raons per les quals aquest és el conveni de representació d'enters utilitzat en tots els computadors moderns.

Recorda, **representar** en complement a dos no vol dir **fer** el complement a dos a tots els nombres, tan sols a aquells que indica el conveni, els negatius.

7 Bibliografia

7.1 Llibres

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

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

7.2 Recursos electrònics

[3] Martí Campoy, Antonio. *Representación de enteros: Complemento a 2*, Universitat Politècnica de València, 2009. http://hdl.handle.net/10251/5234