UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

DISCA

Fonaments de computadors

Tema 5. REPRESENTACIÓ DE LA INFORMACIÓ

Objectius

- Recordar els sistemes binari, octal i hexadecimal i les seues relacions.
- Conèixer les operacions aritmètiques bàsiques en el sistema binari.
- Conèixer les distintes formes de representació de la informació en el computador

Recursos d'aprenentatge FCO

- Poliformat, secció "Recursos"
  - Exercicis sense solució.
  - Solucions als exercicis.
  - **Entrenador personal d'aritmètica.**
  - **Entrenador personal d'enters.**
  - **Convertidor IEEE754.**
  - **Entrenador personal de coma flotant.**
  - Exàmens d'anys anteriors.

- Poliformat, secció "Continguts"
  - Mòdul 12: *Operaciones básicas en binario.* (Teoria i exercicis)
  - Mòdul 13: *Representación de números enteros*. (Teo. i exercicis)
  - Mòdul 14: *Representación de números en coma flotante*.

Índex

- Introducció
- Operacions binàries bàsiques
- Representació de nombres enters
  - Conveni de representació: signe i magnitud
  - Conveni de representació: complement a 2
  - Conveni de representació: excés Z
- Representació de nombres reals
  - Coma fixa
  - Coma flotant. Format estàndard IEEE 754

Introducció

- Representació externa
  - Emprada per les persones: sistema decimal, caràcters alfanumèrics, gràfics, etc.

- Representació interna
  - Utilitzada per l'ordinador: sistema binari, codi ASCII per als caràcters, etc.

Sistemes de numeració: recordatori

FCO

- Conceptes a recordar dels sistemes de numeració
  - Sistemes de numeració posicionals, base, factor d'escala
  - Sistema binari: sistema posicional de base 2
  - Canvis de base: d'una base B qualsevol a decimal
    - Desenvolupament del polinomi de potències de la base
      - $a_{-f}\ a_{-f+1} \ldots a_{-1}$ és la part fraccionària (f dígits)
      - $a_0\ a1 \ldots a_{e-1}$ és la part entera(e dígits)

$$N = \sum_{i=-f}^{e-1} a_i B^i$$

  - Canvis de base: de decimal a una base B qualsevol
    - Divisions successives entre la base B per a la part entera
    - Multiplicacions successives per la base B per a la part fraccionària
    - La coma separa en les dues bases la part entera de la fraccionària

DISCA

Sistemes de numeració

- A més del sistema binari, s'utilitzen també:
  - Octal (base 8 = 2^3)
    - **Cada dígit octal representa un grup d'exactament 3 bits**
    - ***Dígits octals: 0, 1, 2, 3, 4, 5, 6, 7***
  - Hexadecimal (base 16 = 2^4)
    - **Cada dígit hexadecimal representa un grup d'exactament 4 bits**
    - ***Dígits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A (=10_{10}), B (=11_{10}), C (=12_{10}), D (=13_{10}), E (=14_{10}), F (=15_{10})***

- I el seu ús està estés per
  - La facilitat de conversió a / des de binari, i
  - Perquè permeten representar llargues seqüències de bits amb pocs dígits (més fàcils d'utilitzar que una seqüència de bits)

Canvi de bases binària, octal, hexadecimal FCO

- Atès que les bases octal i hexadecimal són potències de 2 (la base binària), es pot demostrar que
  - En octal (base 2^3) un dígit representa un grup de 3 bits
  - En hexadecimal (base 2^4) un dígit representa un grup de 4 bits
  - En aquests dos casos, el canvi d'una representació a una altra es realitza utilitzant una taula, agrupant els bits en blocs de 3 o 4

Octal	Binari
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Hexadecimal	Binari
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

Hex.	Binari
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Canvi de bases binària, octal, hexadecimal FCO

- Canvi a/de binari des de/a octal i hexadecimal
  - Quan el grup de 3/4 bits no està complet, s'ompli amb zeros
    - Zeros a l'esquerra si els bits són de la part entera
    - Zeros a la dreta si els bits són de la part fraccionària
  - Un grup de bits mai pot incloure la coma
    - No es poden barrejar bits de la part entera i de la fraccionària en el mateix grup
    - Començar les agrupacions al voltant de la coma

bit d'ompliment

$111000011011{,}10000001_2 = 111\ 000\ 011\ 011\ ,\ 100\ 000\ 010_2 = 7033{,}402_8$

$111000011011{,}10000001_2 = 1110\ 0001\ 1011\ ,\ 1000\ 0001_2 = E1B{,}81_{16}$

Canvi de bases binària, octal, hexadecimal FCO

- Exemple. Donat N = $F9,E3B_{16}$
  - Passar a decimal
  - Passar a binari

$N = F9,E3B_{16} =$

$F_{16} \times 16^1 + 9_{16} \times 16^0 + E_{16} \times 16^{-1} + 3_{16} \times 16^{-2} + B_{16} \times 16^{-3} =$

$= 15_{10} \times 16^1 + 9_{10} \times 16^0 + 14_{10} \times 16^{-1} + 3_{10} \times 16^{-2} + 11_{10} \times 16^{-3}$

$= 249,8894042969_{10}$

$N = F9,E3B_{16} =$ $1111\ 1001\ ,\ 1110\ 0011\ 1011_2$

Canvi de bases binària, octal, hexadecimal FCO

- Canvi d'octal a binari i hexadecimal

  – $15,36_8$ = 001 101 , 011 110 $_2$

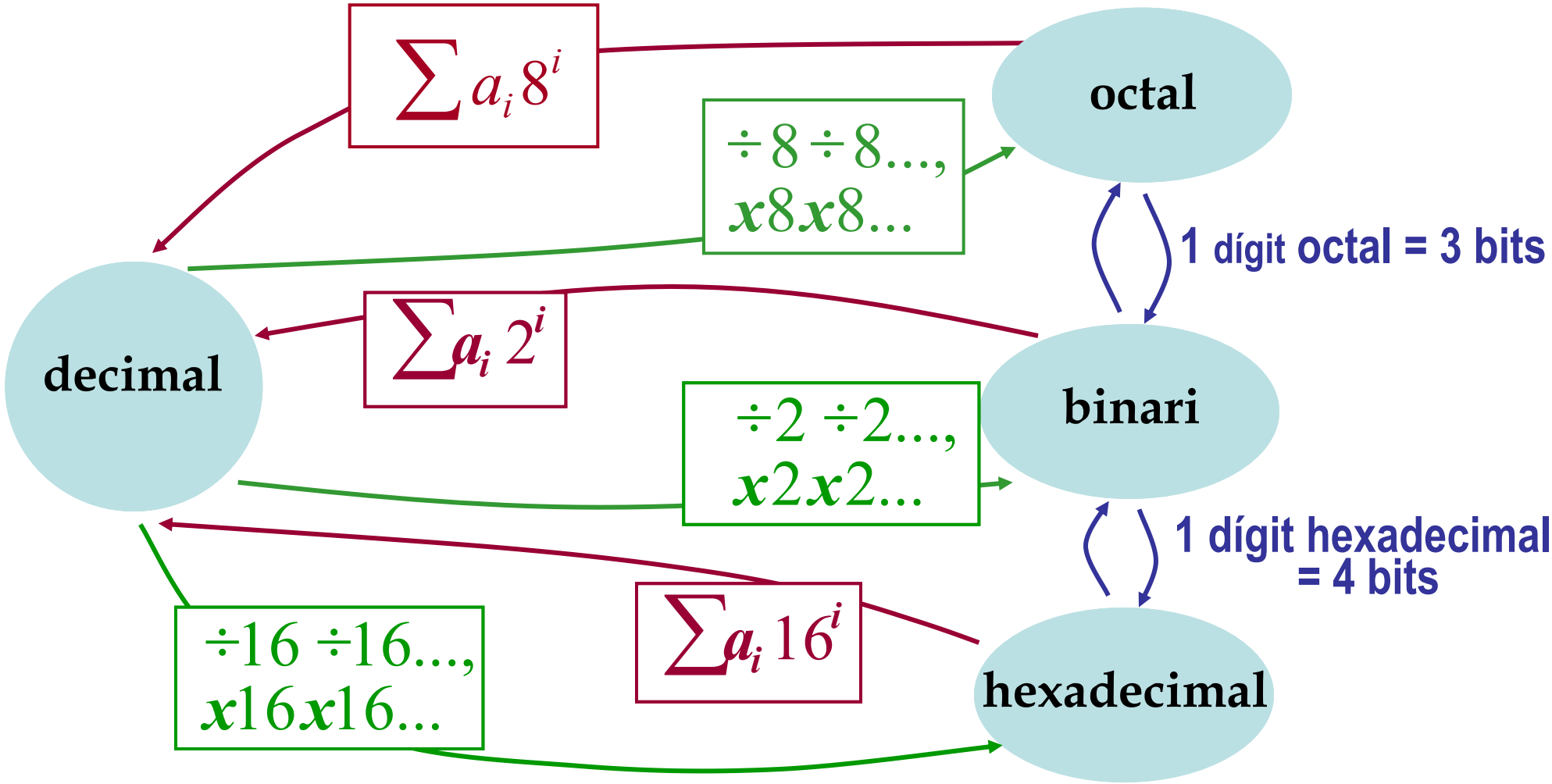
  – $15,36_8$ = 001 101 , 011 110 $_2$ =

  (i ara, agrupant els bits de 4 en 4)

  1101 , 0111 1000 $_2$ = $D,78_{16}$

Canvis de base. Resum

FCO



DISCA

Representació de nombres naturals

- Es denominen nombres naturals a aquells nombres sense signe i sense part fraccionària
- Els nombres naturals es representen en l'ordinador directament per mitjà del seu corresponent valor en binari
- És el que es denomina "binari natural"
- Exemple:

valors	
binari	natural
0 0 0	0
0 0 1	1
0 1 0	2
0 1 1	3
1 0 0	4
1 0 1	5
1 1 0	6
1 1 1	7

Representació de nombres naturals

- El rang de representació ve donat per l'interval de nombres representables en un format
- En aquest cas, format = longitud en bits
- Amb un total de n bits es poden representar els valors $[0, 2^n - 1]$
- És possible que en operar existisca desbordament (el resultat no és representable amb n bits)

Operacions binàries bàsiques amb naturals

- Les operacions aritmètiques bàsiques (suma, resta, multiplicació i divisió) utilitzen en binari les mateixes regles que en decimal, excepte el referent a la base
- En la suma de dos o més bits:
  - Es produeix bit de ròssec (*carry*) quan la suma dels operands és igual o superior a la base (>= 2 en binari, i no >= 10)
- En la resta de dos o més bits:
  - El préstec (*borrow*) es calcula com a base + minuend - subtrahend (2 + minuend - subtrahend, i no 10 + minuend - subtrahend)

Operacions binàries bàsiques amb naturals

- Com a funcions combinacionals que són, es poden formalitzar en una taula de veritat
- Suma de dos bits

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 10 (ròssec=1)

A	*B*	*Ròssec*	*Suma*
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Al circuit corresponent a aquesta taula de veritat es coneix com a semisumador o "*Half Adder*" (HA):

S = A $\oplus$ B, *ròssec* = A·B

Operacions binàries bàsiques amb naturals

- Com a funcions combinacionals que són, es poden formalitzar en una taula de veritat
- Resta de dos bits

0 - 0 = 0
0 - 1 = 1 (i 1 de préstec)
1 - 0 = 1
1 - 1 = 0

A	*B*	*Resta*	*Préstec*
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

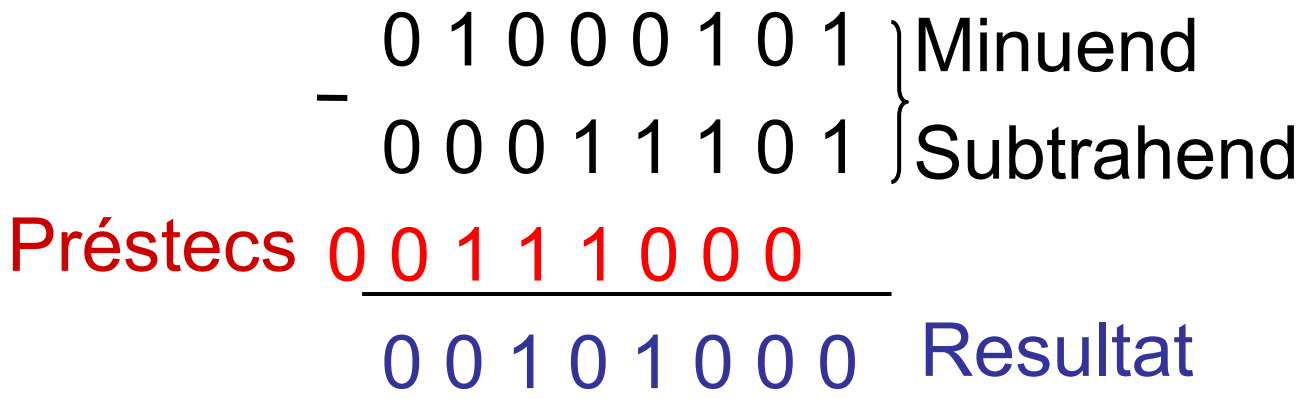
Operacions binàries bàsiques amb naturals

- Per a sumar dos o més nombres d'N bits,
  - El ròssec generat en una columna és un bit més a sumar en la columna següent

Bits de ròssec	0 1 1 1 1 1 0 1	
+	0 1 1 0 0 1 0 1	Sumands
	0 0 1 1 1 1 0 1	
	1 0 1 0 0 0 1 0	Resultat

Operacions binàries bàsiques amb naturals FCO

- Per a restar dos o més nombres d'N bits
  - El préstec generat en una columna és un bit més a restar en la columna següent



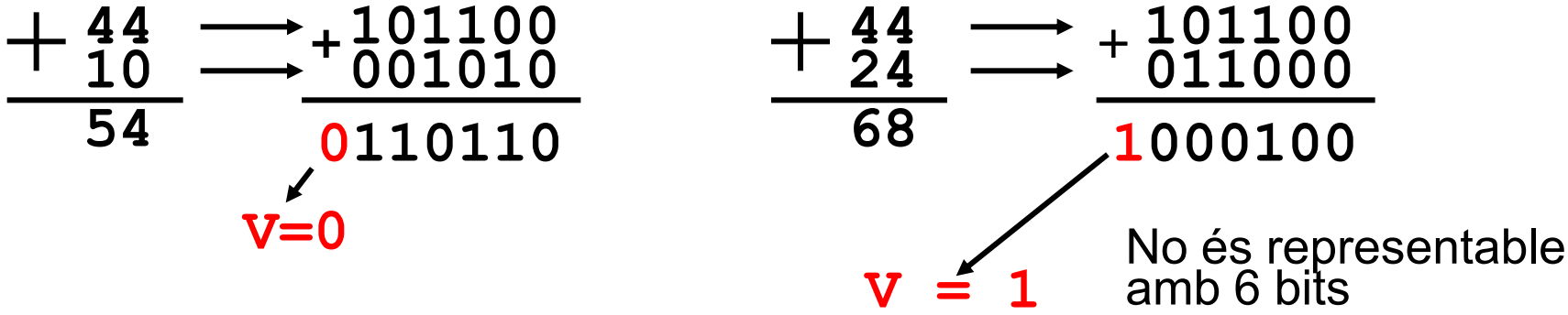
Desbordament en les operacions

- Desbordament o sobreeiximent (*overflow*)
- Ocorre quan el resultat d'una operació no és representable en el format utilitzat (queda fora del rang de representació)
  - Pot ocórrer en qualsevol conveni o format de representació
  - La limitació en el nombre de bits emprats per emmagatzemar les dades crea aquest problema
  - És una condició o adjectiu sobre el valor del resultat. Indica que la seqüència de bits calculada com a resultat no és vàlida, i que per tant no hi ha resultat
- *Flag* o indicador de desbordament (V o OV)
  - Eixida d'un circuit aritmètic per a indicar la situació de desbordament en l'operació

Desbordament en nombres naturals

- Desbordament en nombres naturals
  - Ocorre quan el ròssec (carry) de l'ultima etapa és 1 i indica que és necessari almenys un bit més per a representar el resultat
  - Expressió algebraica de l'indicador (flag), $V = C_{n-1}$

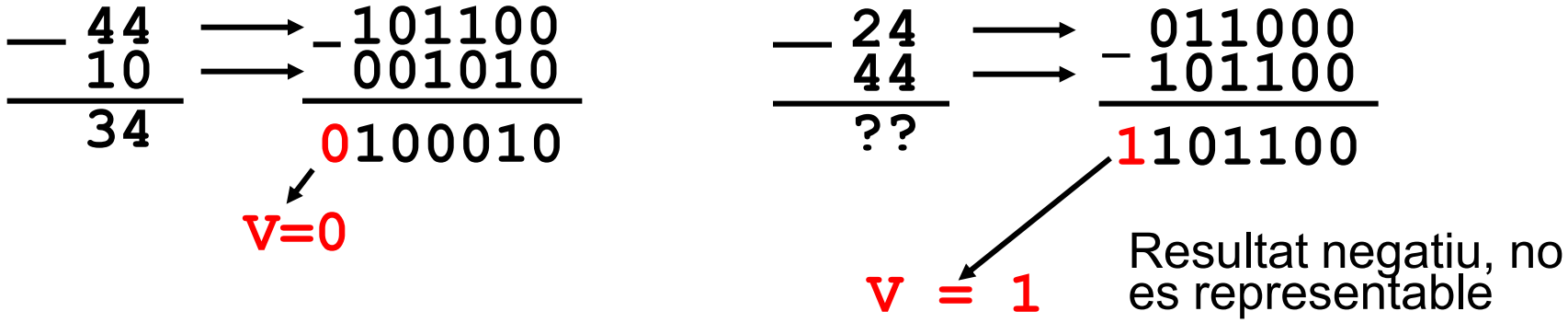
- Exemple. Naturals representats amb 6 bits



Desbordament en nombres naturals

FCO

- Exemple



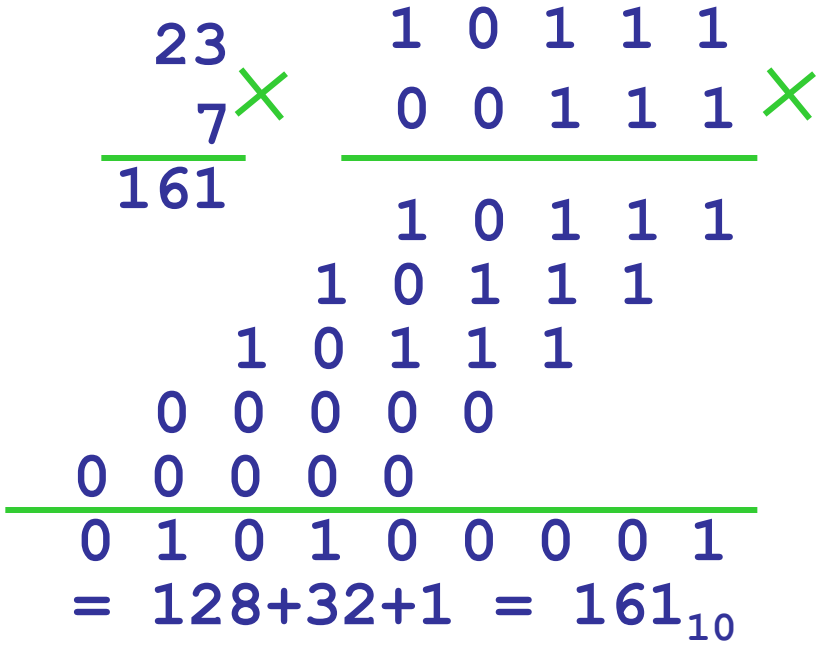
- Exercicis
  - Sumar 101110001_2 i 001110110_2
  - Restar 101110001_2 i 001110110_2
  - Sumar 101110001_2, 001110110_2 i 011001100_2

DISCA

Operacions binàries bàsiques

FCO

- Multiplicació
  - En multiplicar dos nombres d'n i m bits respectivament, el resultat sempre podrà representar-se amb un màxim de n+m bits



DISCA

Operacions binàries bàsiques

- Multiplicació
  - Exercicis

```
      1 0 0 1 1
      1 0 1 0 1   x
-------------------
      1 0 0 1 1
    1 0 0 1 1
  1 0 0 1 1
-------------------
1 1 0 0 0 1 1 1 1
```

```
      1 0 0 0 1
      1 1 1 0 1   x
-------------------
      1 0 0 0 1
    1 0 0 0 1
  1 0 0 0 1
-------------------
1 1 1 1 0 1 1 0 1
```

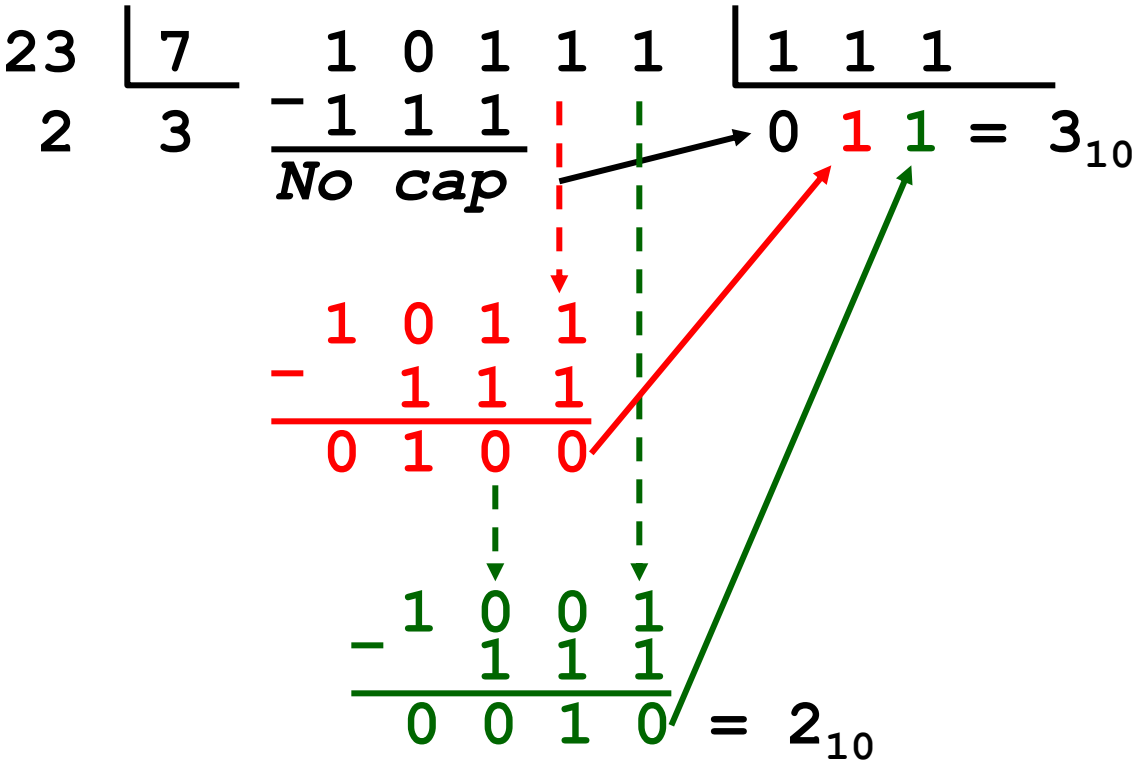
Operacions binàries bàsiques

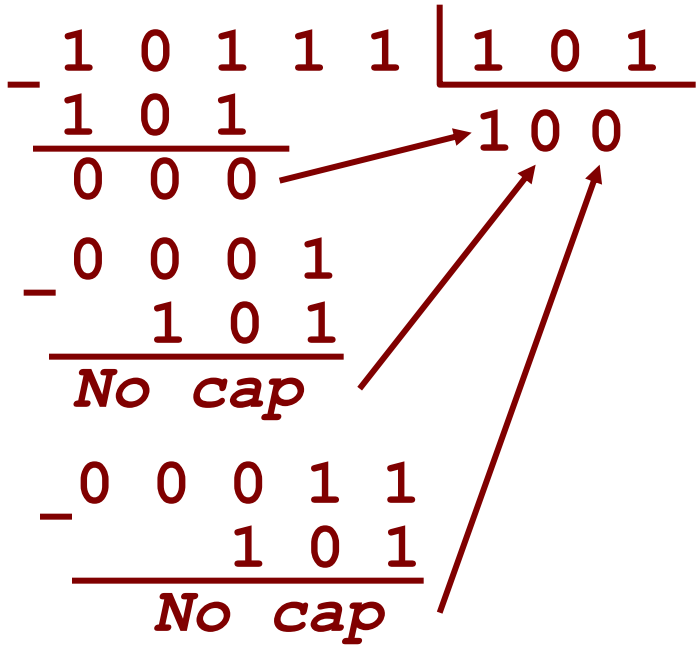
- Divisió
  - L'algorisme de divisió és el mateix que en decimal, excepte que quan el dígit del quocient no és zero, només pot ser un 1 (és a dir, si és el cas el dígit del quocient és segur un 1) quan en decimal cal provar entre 1 i 9.

Operacions binàries bàsiques

FCO

- Divisió
  - Exemples





DISCA

Operacions binàries bàsiques

- Complement a 1 d'un nombre X d'n bits (sense signe)
  - Definició: $Ca1(X) = 2^n - X - 1$
  - Es pot calcular també invertint els bits d'X
- Complement a 2 d'un nombre X d'n bits (sense signe)
  - Definició: $Ca2(X) = 2^n - X$
  - Es pot calcular també invertint els bits d'X i sumant 1,
  - i també com $Ca2(X) = Ca1(X) + 1$

Operacions binàries bàsiques

- Els complements són operacions reversibles,
  - Ca1(Ca1(X)) = X
    - Ca1(Ca1(X)) = 2^n - (2^n - X - 1) - 1 = 2^n - 2^n + X + 1 - 1 = X
  - Ca2(Ca2(X)) = X
    - Ca2(Ca2(X)) = 2^n - (2^n - X) = 2^n - 2^n + X = X
- Exercicis:
  - Ca1(111000101) = 000111010
  - Ca1(000111010) = 111000101
  - Ca2(111000101) = 000111011
  - Ca2(000111011) = 111000101

Representació d'enters

- Nombres enters
  - Nombres amb signe però sense part fraccionària
- Problema de representació
  - En intentar emmagatzemar un nombre enter en la memòria de l'ordinador, aquest no emmagatzema signes com "+" o "-", només bits.

Representació d'enters

- Solució
  - Definir convenis de representació, regles arbitràries que ens permeten emmagatzemar valors tant positius com negatius utilitzant únicament seqüències d'1s i 0s
- Extensió de signe
  - Capacitat d'ampliar de forma senzilla (sense realitzar operacions aritmètiques) el nombre de bits utilitzat per a representar una determinada quantitat i que mantinga el criteri de representació utilitzat

Representació d'enters

- Criteri 1: Signe i Magnitud
  - Es reserva el bit MSB per a representar el signe del nombre;
    - MSB = 0 per als nombres positius
    - MSB = 1 per als nombres negatius
  - La resta de bits representa el valor absolut en binari natural
  - Rang de representació simètric [- (2^{n-1}-1), + 2^{n-1}-1]
  - Hi ha dos zeros, l'un "positiu" i l'altre "negatiu"
  - Abans de sumar o restar cal analitzar el signe dels operands

Representació d'enters

- Criteri 1: Signe i Magnitud
  - Es pot fer extensió de signe sense més que afegir els zeros que siguen necessaris a l'esquerra de la magnitud, mantenint el bit de signe com l'MSB

```
-3 =>   1 11          +3 =>   0 11
        1 011                 0 011
        1 0011                0 0011
        1 00011               0 00011
```

S. i Mag.	Decimal
0 000…000	+0
0 000…001	+1
0 000…010	+2
0 000…011	+3
…	…
0 111…110	$+(2^{n-1} - 2)$
0 111…111	$+(2^{n-1} - 1)$

S. i Mag.	Decimal
1 000…000	-0
1 000…001	-1
1 000…010	-2
1 000…011	-3
…	…
1 111…110	$-(2^{n-1} - 2)$
1 111…111	$-(2^{n-1} - 1)$

Representació d'enters

- Criteri 2: Complement a 2
  - No cal confondre aquest conveni de representació amb l'operació matemàtica Ca2
  - Utilitza dos mètodes diferents, en funció del signe del nombre. Per a una representació amb n bits:
    - Els positius es representen pel seu valor absolut, en binari natural, amb n-1 bits, i se l'afig un 0 a l'esquerra.
    - Els negatius es representen pel Ca2 (positiu)
  - Exemple amb 5 bits. Representar +4 i -4

+4 amb 4 bits = 0100; afegim el zero: $00100_{c2} = +4_{c2}$

-4 = Ca2 (+4) = Ca2 (00100) = $11100_{c2} = -4_{c2}$

Representació d'enters

- Criteri 2: Complement a 2
  - En la representació resultant, el bit MSB indica el signe
    - MSB=0 per als nombres positius
    - MSB=1 per als nombres negatius
  - Rang de representació asimètric [- 2^{n-1}, + 2^{n-1}-1]
  - Hi ha un únic zero

Compl. A 2	Decimal
0000…000	+0
0000…001	+1
0000…010	+2
0000…011	+3
…	…
0111…110	+(2^{n-1} - 2)
0111…111	+(2^{n-1} - 1)

Compl. A 2	Decimal
1000…000	-2^{n-1}
1000…001	-(2^{n-1} - 1)
…	…
1111…100	-4
1111…101	-3
1111…110	-2
1111…111	-1

Representació d'enters

- Criteri 2: Complement a 2
  - Les operacions de suma i resta no necessiten realitzar ajustos per a obtindre el resultat de l'operació representat segons aquest conveni de representació. Açò fa que siga el conveni més utilitzat quan es tracta d'operar amb nombres enters
  - És possible realitzar extensió de signe, replicant el bit de signe

```
-2 => 110          +2 => 010
      1110                0010
      11110               00010
      111110              000010
```

Representació d'enters

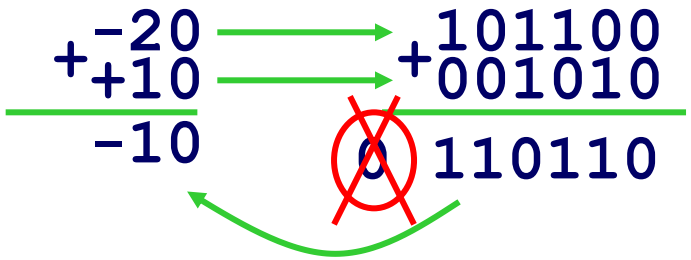
- Sumes i restes en Ca2
  - Siguen A i B dos nombres enters (positius o negatius) representats segons el conveni de Ca2
  - Per a calcular R=A+B i obtindre R en el conveni de representació Ca2, realitzem una suma binària i ignorarem el ròssec final
  - Per a calcular R=A-B i obtindre R en el conveni de representació Ca2, realitzem la suma binària A + Ca2(B) i ignorem el ròssec final

$$A - B = A + (-B) = A + Ca2(B)$$

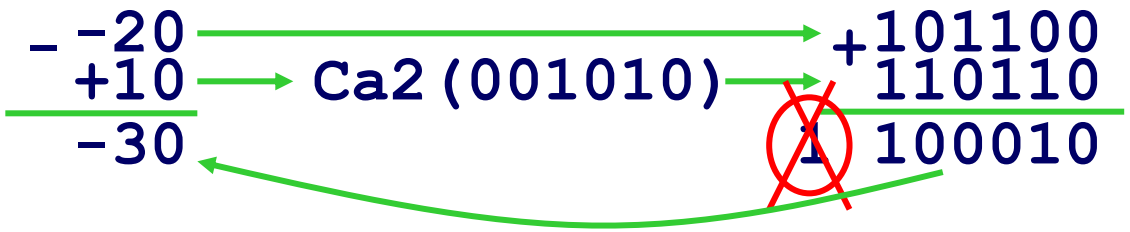
Representació d'enters

- Exercici: A=-20, B=+10. Calcular, representant en Ca2 i amb 6 bits

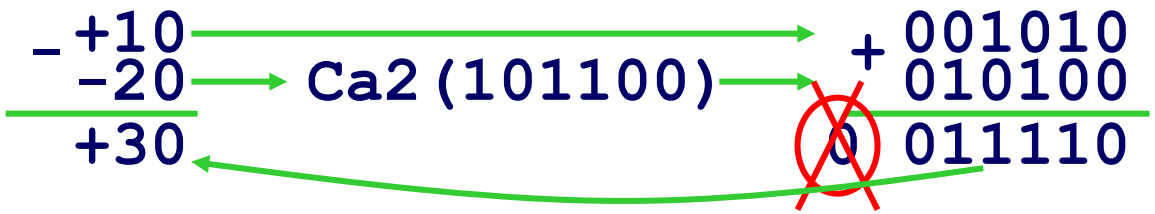
- A+B



- A-B



- B-A



Desbordament en les operacions en Ca2

FCO

- Desbordament en operar amb nombres en Ca2
  - Ocorre quan el resultat d'una suma en Ca2 no és representable, bé per que se'n ix per la dreta (positiu massa gran), bé per que se'n ix per l'esquerra (negatiu massa gran)
  - Els humans ho poden detectar mirant el bit de signe: la suma de dos positius dóna un de negatiu, o la suma de dos negatius dóna un de positiu
  - Els computadors necessiten un circuit per a detectar i informar del desbordament.

DISCA

Desbordament en les operacions en Ca2

FCO

- Desbordament en Ca2

$$A + B = \begin{array}{cccccc} C_{n-1} & C_{n-2} & \cdots & & & \\ A_{n-1} & A_{n-2} & \cdots & A_1 & A_0 & \\ + & & & & & \\ B_{n-1} & B_{n-2} & \cdots & B_1 & B_0 & \\ \hline \cancel{C_{n-1}} & R_{n-1} & R_{n-2} & \cdots & R_1 & R_0 \end{array}$$

A_{n-1}	B_{n-1}	C_{n-2}	C_{n-1}	R_{n-1}	
0	0	0	0	0	A i B >= 0
0	0	1	0	1	→ Desb. si R < 0
0	1	0	0	1	
0	1	1	1	0	A>=0 i B<0 o viceversa
1	0	0	0	1	Desbordament impossible
1	0	1	1	0	
1	1	0	1	0	A i B < 0
1	1	1	1	1	→ Desbordament si R>=0

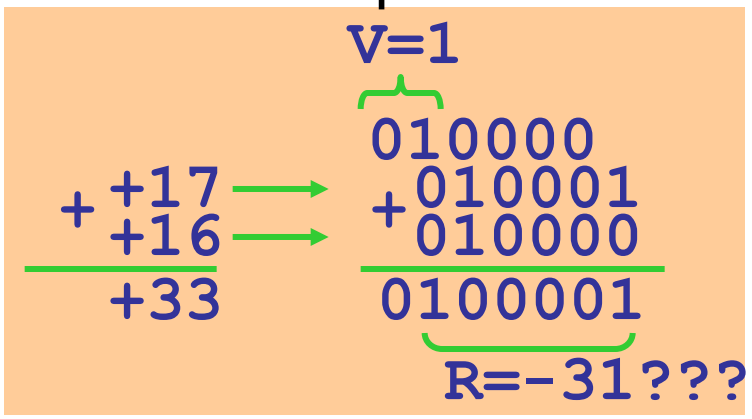
$$V = C_{n-1} \oplus C_{n-2}$$

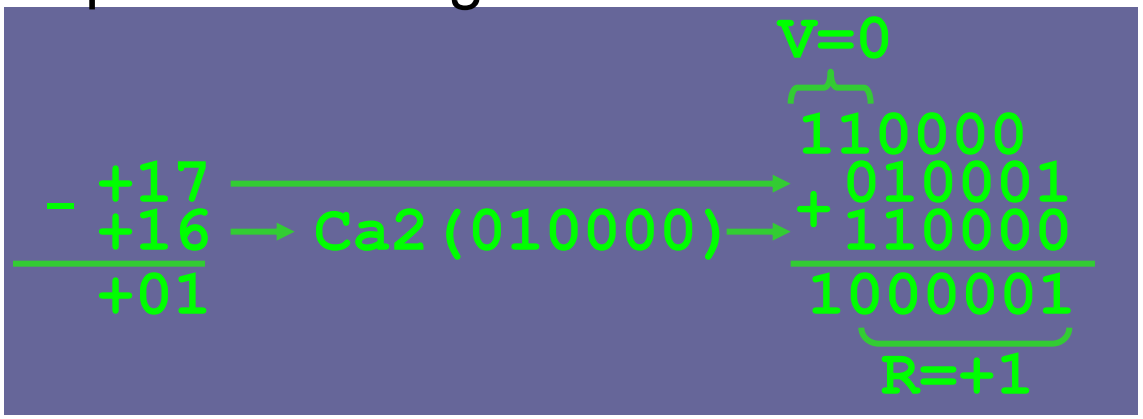
DISCA

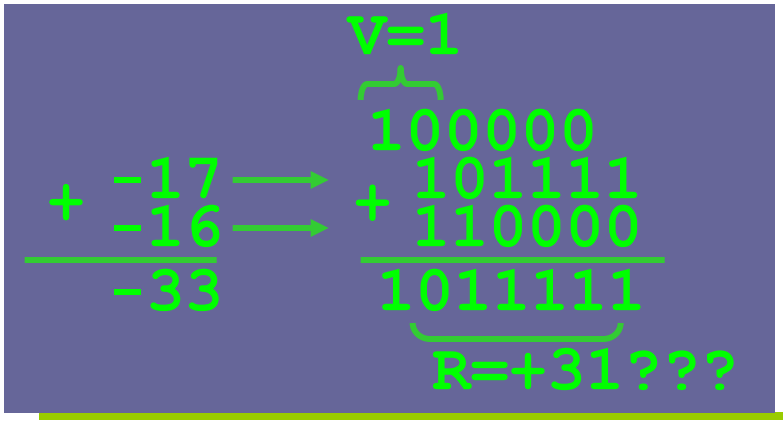
Desbordament en les operacions en Ca2

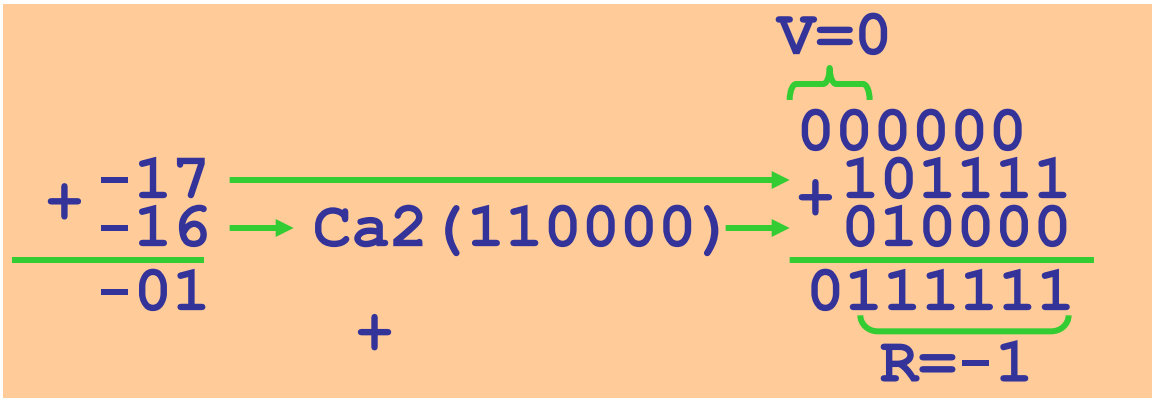
FCO

- Desbordament en operar amb nombres en Ca2.
  - Exemples: realitze les operacions següents amb 6 bits









DISCA

Representació d'enters

- Criteri 3: Excés Z
  - Es tria un valor arbitrari denominat excés o biaix (Z). El binari natural de Z representarà el zero
  - Qualsevol enter A es representa pel binari natural A+Z. Això implica que A+Z >= 0 per a poder representar en binari natural
  - Exemple: representeu +4 i -4 amb 6 bits i Excés 31

$$+4_{10} = 31 + 4 = 35 = 100011_{z31}$$
$$-4_{10} = 31 - 4 = 27 = 011011_{z31}$$

Representació d'enters

- Criteri 3: Excés Z
  - Rang de representació asimètric [- Z, + 2^n - 1 - Z]
  - Cada valor de Z defineix un rang de representació diferent
  - Es recomana assignar a Z el valor $2^{n-1} - 1$, on n = nombre de bits → Rang [- Z, Z+1]
  - Hi ha un únic zero

Excés $2^{n-1}-1$	Decimal
0000…000	$-(2^{n-1} - 1)$
0000…001	$-(2^{n-1} - 2)$
0000…010	$-(2^{n-1} - 3)$
0000…011	$-(2^{n-1} - 4)$
…	…
0111…110	-1
0111…111	0

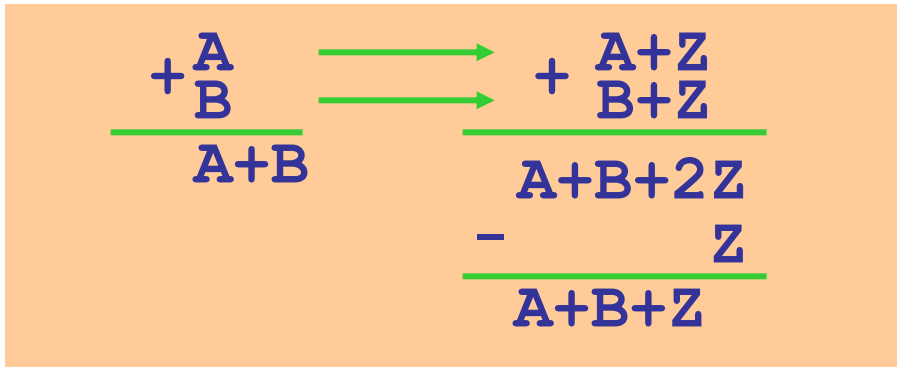
Excés $2^{n-1}-1$	Decimal
1000…000	+1
1000…001	+2
…	…
1111…100	$+(2^{n-1} - 3)$
1111…101	$+(2^{n-1} - 2)$
1111…110	$+(2^{n-1} - 1)$
1111…111	$+2^{n-1}$

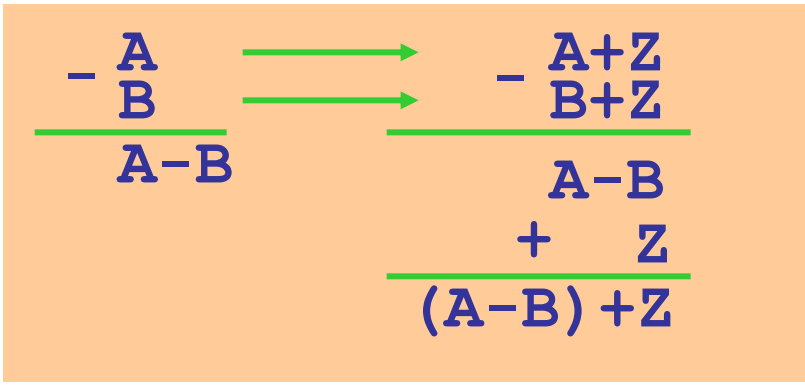
Representació d'enters

- Criteri 3: Excés Z
  - En la representació resultant, el bit MSB no indica el signe, ja que la representació depén del valor de Z
  - L'avantatge d'aquest criteri és que les quantitats representades estan ordenades de menor a major. És a dir, des de la major quantitat negativa fins a la major quantitat positiva. Açò permet comparar dos nombres amb un senzill circuit i sense fer operacions aritmètiques
  - No és possible fer extensió de signe sense fer operacions matemàtiques, ja que habitualment el valor de l'excés depén de la quantitat de bits disponibles.

Representació d'enters

- Criteri 3: Excés Z
  - En fer operacions de suma o resta, el resultat no està representat d'acord al criteri, per la qual cosa cal ajustar-lo





Representació d'enters

- Exemples (n=6 bits)

Decimal	*SiM*	*Ca2*	*Excés 31*
0	0 00000	000000	011111
+1	0 00001	000001	100000
-1	1 00001	111111	011110
+5	0 00101	000101	100100
-5	1 00101	111011	011010
+31	0 11111	011111	111110
-31	1 11111	100001	000000
+32	----	----	111111
-32	----	100000	----
+33	----	----	----
-33	----	----	----
	[-31,+31]	***[-32,+31]***	***[-31,+32]***

Representació d'enters

- Resum

	Signe i Magnitud	Compl. a 2	Excés $2^{n-1}-1$
Rang	$[-(2^{n-1}-1), +2^{n-1}-1]$	$[-2^{n-1}, +2^{n-1}-1]$	$[-(2^{n-1}-1), +2^{n-1}]$
Ext. signe	Sí (amb compte)	Sí	No
Inconvenients	-Sumes i restes complicades - +0 i -0		Sumes i restes complicades
Ventages	Facilitat d'us	Facilitat per operar	Facilitat per comparar

Representació de reals

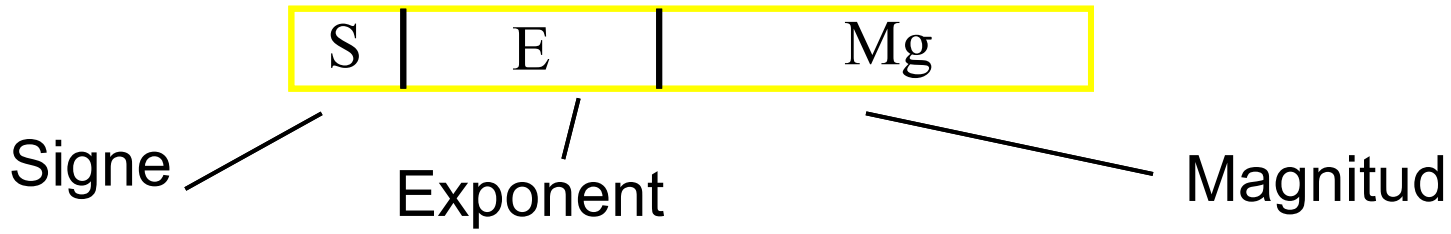
- Dues maneres de representar el nombres reals:
  - Coma fixa
  - Coma flotant
- Coma fixa: es dediquen n bits per a la part entera i p bits per a la part fraccionaria. n i p són fixes.
  - $R = e_{n-1}\ e_{n-2} \ldots e_1\ e_0\ ,\ f_{-1}\ f_{-2} \ldots f_{-p+1}\ f_{-p}$
  - L'aritmètica coincideix amb l'aritmètica entera
  - Problema a causa de la longitud fixa dels camps:
    - Amb un format de coma fixa de 3 dígits per a la part entera i 3 més per a la part fraccionària, feu el càlcul $000{,}125_{10} \times 000{,}001_{10}$
    - $000{,}125_{10} \times 000{,}001_{10} = 000{,}000125_{10} = Truncant = 000{,}000_{10}$

Representació de reals

- Coma flotant
  - Un nombre real R es representa mitjançant l'expressió:

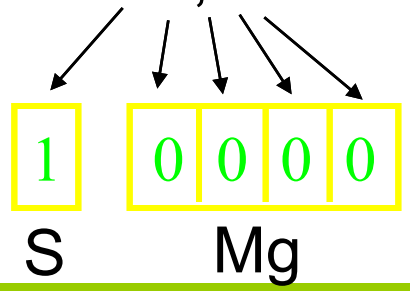
$$R = M \times B^E$$

    - M és la mantissa, amb q bits, fraccionari, en coma fixa i amb signe (SiM): $m_{q-1}\ m_{q-2}\ \ldots\ m_1\ m_0$
    - B és la base de la potència, generalment 2
    - E és l'exponent, amb p bits, enter, i generalment representat en excés $2^{p-1} - 1$: $e_{p-1}\ e_{p-2}\ \ldots\ e_1\ e_0$
  - La representació en la memòria de l'ordinador sol ser:



Representació de reals

- Coma flotant
  - L'aritmètica és més complexa que l'aritmètica entera
    - Hi ha circuits específics (unitats de coma flotant, FPU: *Floating Point Unit*)
- Normalització de la mantissa
  - L'objectiu de normalitzar la mantissa és no perdre bits significatius (1) que malbaraten l'espai per a emmagatzemar bits no significatius (0).
  - Exemple: emmagatzemar la mantissa –0,00010001 amb un format de 5 bits

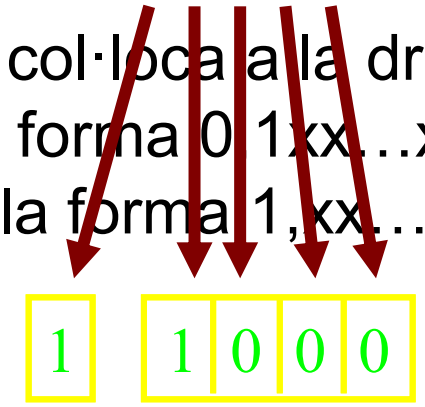


Representació de reals

- Normalització de la mantissa
  - En normalitzar la representació de la mantissa s'assegura que s'emmagatzemen el nombre més gran possible d'uns
  - El primer bit significatiu (1) es col·loca al voltant de la coma. Per a moure-la caldrà modificar el valor de l'exponent
  - Exemple: mantissa normalitzada: $-1{,}0001 \times 2^{-4}$
  - Si el primer bit significatiu es col·loca a la dreta de la coma, els nombres tindran la forma 0,1xx…x. Si es col·loca a l'esquerra, tindran la forma 1,xx…x

Representació de reals

- Normalització de la mantissa
  - En normalitzar la representació de la mantissa s'assegura que s'emmagatzemen el nombre més gran possible d'uns
  - El primer bit significatiu (1) es col·loca al voltant de la coma. Per a moure-la caldrà modificar el valor de l'exponent
  - Exemple: mantissa normalitzada: $-1{,}0001 \times 2^{-4}$
  - Si el primer bit significatiu es col·loca a la dreta de la coma, els nombres tindran la forma 0,1xx…x. Si es col·loca a l'esquerra, tindran la forma 1,xx…x

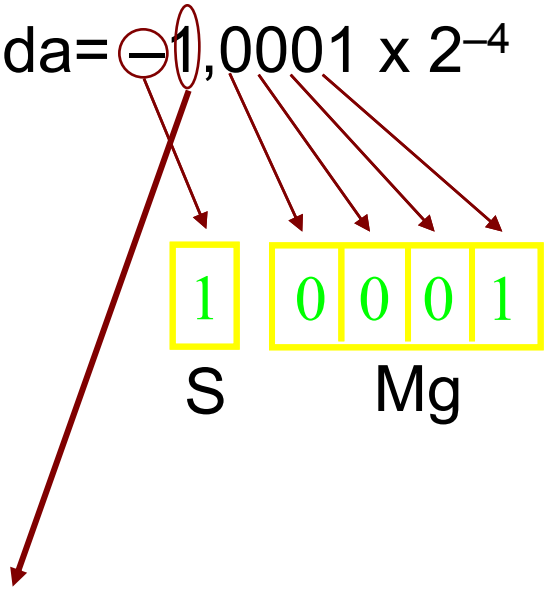


Representació de reals

- Tècnica del bit implícit
  - Atés que s'assegura la posició del primer bit significatiu, no cal emmagatzemar-lo (encara que s'haurà de tenir en compte en les operacions)
  - Exemple: mantissa normalitzada= $-1{,}0001 \times 2^{-4}$

Representació en memòria

amb la tècnica de bit implícit:



No s'emmagatzema per ser sempre 1!

Rep. de reals: estàndard IEEE 754

- Format estàndard IEEE 754
  - Sorgeix per a facilitar el transvasament d'informació entre distintes màquines
  - Dues versions:

Precisió	*Total bits*	*Signe*	*Exponent*	*Magnitud*
Simple	32	1	8	23*
Doble	64	1	11	52*

*Hi ha un bit addicional que no s'emmagatzema (bit implícit)

Rep. de reals: estàndard IEEE 754

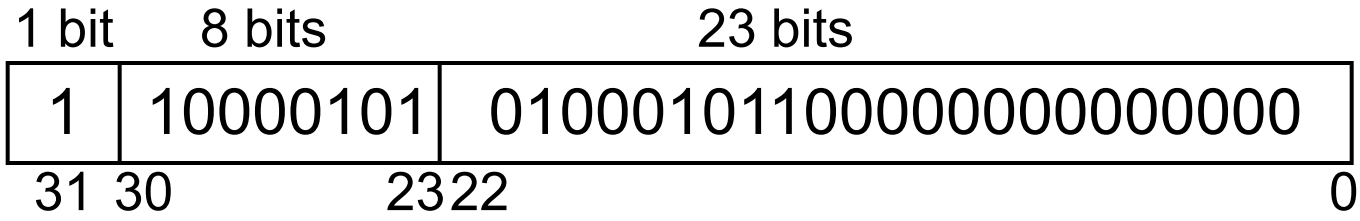
FCO

- Característiques
  - La mantissa es normalitza en la forma 1,xx…x. Es representa en signe i magnitud i s'utilitza la tècnica del bit implícit
  - L'exponent està en excés $2^{n-1}-1$ (127 o 1023).
  - L'ordre dels camps és S, E i M. La base B és 2
  - Els nombres "normals" tenen la forma: $\pm 1,M \times 2^{E-127}$
  - Hi ha nombres "desnormalitzats" que tenen la forma $\pm 0,M \times 2^{-126}$

DISCA

Rep. de reals: estàndard IEEE 754

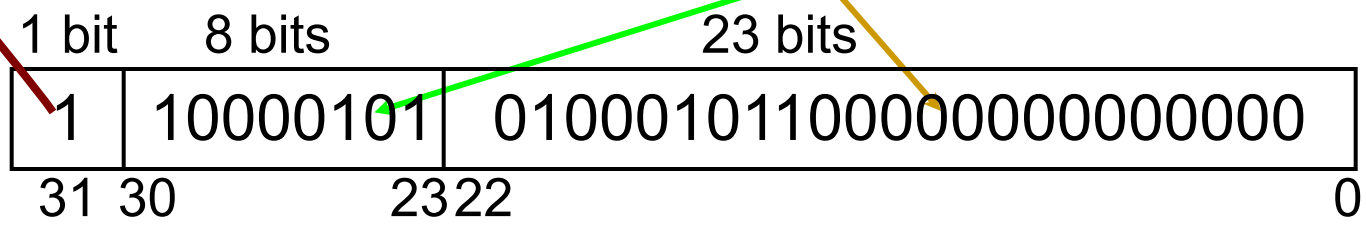
- Exemple.
  - Representeu el nombre - 81,375 en format IEEE de simple precisió (format normalitzat)
    - Part entera: - 81_{10} = -1010001_2
    - Part decimal: $0{,}375_{10}$ = 011_2
    - Mantissa = - $1010001{,}011_2$
    - Normalitzada = $-1{,}010001011_2 \times 2^{+6}$
    - Exponent = 6_{10}
    - Exponent en excés 127 = 133_{10} = 10000101_2



Rep. de reals: estàndard IEEE 754

- Exemple
  - Representeu el nombre - 81,375 en format IEEE de simple precisió (format normalitzat)
    - Part entera: $-81_{10} = -1010001_2$
    - Part decimal: $0{,}375_{10} = 0{,}011_2$
    - Mantissa = $-1010001{,}011_2$
    - Normalitzada = $-1{,}010001011_2 \times 2^{+6}$
    - Exponent = 6_{10}
    - Exponent en excés 127 = $133_{10} = 10000101_2$

Bit implícit que no s'emmagatzema



Rep. de reals: estàndard IEEE 754

- Exercicis
  - La representació de 5,6875 en IEEE 754 de simple precisió és
    - $+5{,}6875_{10} = 101{,}1011_2 = 101{,}1011 \times 2^0 = 1{,}011011 \times 2^2$

S	E	Mg
0	10000001	01101100000000000000000

Rep. de reals: estàndard IEEE 754

- Exercicis
  - Si la seqüència de bits (en hexadecimal) és C0B60000, s'està representant el valor

S	*E*	*Mg*
1	10000001	01101100000000000000000

Rep. de reals: estàndard IEEE 754

- Casos especials del format estàndard
  - Si E = 00…0 i M = 0, s'està representant el valor 0, que no és representable i utilitza el format normalitzat
  - Si E = 00…0 i M != 0, s'està representant un valor molt petit en format desnormalitzat: $\pm 0{,}M \times 2^{-126}$
  - Si E = 11…1 y M = 0, s'està indicant que el resultat de l'operació no és representable. Serà $+\infty$ o $-\infty$ segons el signe
  - Si E = 11…1 i M != 0, s'està indicant que el resultat d'una operació no té sentit (exemple, 0÷0). Aquest tipus de resultats es coneixen com NaN ("Not a Number")
- Les seqüències 00…000 i 11…111 de l'exponent no serveixen per a representar nombres "normals"

Rep. de reals: estàndard IEEE 754

- Rang de representació
  - Determinat principalment pel nombre de bits assignats a E
- Precisió: distància entre dos valors representables consecutius
  - Determinat principalment pel nombre de bits assignats a M
- Possibles desbordaments en representar valors
  - *Overflow*. Quan el nombre està tan allunyat del 0 (valor absolut molt gran) que no es pot representar
  - *Underflow*. Quan el nombre està tan pròxim al 0 (valor absolut molt petit) que no es pot representar

Rep. de reals: estàndard IEEE 754

FCO

- Rang de representació dels nombres normalitzats
  - Del rang original [-127, +128] (o [-1023, +1024]) de l'exponent, les seqüències binàries 00…000 i 11…111 són indicacions de casos especials, i no serveixen per a representar nombres normals
  - Per tant, el rang disponible és [-126, +127] (o [-1022, +1023])
  - En valor absolut, el nombre més petit representable en simple precisió és $1{,}00\ldots00_2 \times 2^{-126}$
  - En valor absolut, el nombre més gran representable en simple precisió és $1{,}11\ldots111_2 \times 2^{+127} \approx 1{,}0 \times 2^{+128}$

DISCA

Rep. de reals: estàndard IEEE 754

FCO

- Rang de representació dels nombres desnormalitzats
  - Els nombres desnormalitzats ofereixen un altre conjunt de nombres representables, amb el valor de l'exponent fix: -126 (o -1022)
  - En valor absolut, el nombre més petit en simple prec. és

    $0{,}00\ldots001_2 \times 2^{-126} = 1{,}0 \times 2^{-126-23} = 1{,}0 \times 2^{-149}$
  - En valor absolut, el nombre més gran en simple prec. és

    $0{,}11\ldots111_2 \times 2^{-126} \approx 1{,}0 \times 2^{-126}$

    que coincideix amb l'inici del rang dels nombre normalitzats

DISCA

Rep. de reals: estàndard IEEE 754

- Rang complet en simple precisió

$$0 \cup \left[\pm 2^{-149}, \pm 2^{-126}\right[\cup \left[\pm 2^{-126}, \pm 2^{+128}\right[\approx \left[\pm 2^{-149}, \pm 2^{+128}\right]$$

Representació de caràcters

FCO

- Són caràcters
  - Les lletres (“a”, …, ”z”, “A”, …, “Z”)
  - Els dígits (“0”, …, “9”)
  - Els signes de puntuació (“.”, “,”, “;”, …)
  - I els símbols especials (“*”, “&”, “$”, …)
- Per a representar caràcters s’assigna un codi numèric a cadascun dels caràcters per mitjà d’una taula
- L’ordinador sempre treballa amb els codis, mai amb els símbols gràfics

Representació de caràcters

- Les característiques d'una representació són
  - Longitud en bits dels codis
  - Nombre de caràcters representable
  - L'assignació de codis a cada caràcter (la taula de caràcters)
- ASCII (*American Standard Code for Information Interchange*)
  - Longitud fixa, igual per a tots els codis
  - ASCII original. Longitud de codi de 7 bits
  - ASCII estés. Ampliació per a caràcters internacionals, amb una longitud de codi de 8 bits

Representació de caràcters

FCO

- EBCDIC (*Extended Binary Coded Decimal Interchange Code*)
  - Sorgit el1964 amb el sistema IBM S360
  - Longitud fixa de 8 bits
  - Només s'usa en alguns sistemes *mainframe*
- Unicode: Universalitat, Uniformitat, Unicitat
  - Tractament de textos en diferents llengües
  - Textos en llengües mortes i altres disciplines
  - Es poden gastar 8, 16 i 32 bits
  - És el futur, ara!

DISCA

Representació de caràcters

FCO

- Taula ASCII original (7 bits)

	0	16	32	48	64	80	96	112
+0	*NUL*	*DLE*	*SP*	0	@	P	`	p
+1	*SOH*	*DC1*	!	1	A	Q	a	q
+2	*STX*	*DC2*	"	2	B	R	b	r
+3	*ETX*	*DC3*	#	3	C	S	c	s
+4	*EOT*	*DC4*	$	4	D	T	d	t
+5	*ENQ*	*NAK*	%	5	E	U	e	u
+6	*ACK*	*SYN*	&	6	F	V	f	v
+7	*BEL*	*ETB*	'	7	G	W	g	w
+8	*BS*	*CAN*	(	8	H	X	h	x
+9	*HT*	*EM*	)	9	I	Y	i	y
+10	*LF*	*SUB*	*	:	J	Z	j	z
+11	*VT*	*ESC*	+	;	K	[	k	{
+12	*FF*	*FS*	,	<	L	\	l	|
+13	*CR*	*GS*	-	=	M	]	m	}
+14	*S0*	*RS*	.	>	N	^	n	~
+15	*S1*	*US*	/	?	O	_	o	*DEL*

El codi ASCII de "z" és

112 + 10 = 122

DISCA

Representació de caràcters

- Propietats del Codi ASCCI
- És fàcil convertir entre lletres majúscules i minúscules, simplement canviant el bit 5 o sumant/restant 32:
  - Lletra “A”: Codi 65 = 0x41 = 0100 0001_2
  - Lletra “a”: Codi 97 = 0x61 = 0110 0001_2

- És fàcil convertir un símbol numèric al seu valor, simplement ficant a zero la part alta o restant-li 48:
  - Símbol “2”: Codi 50 = 0x32 = 0011 0010_2
  - Símbol “9”: Codi 57 = 0x39 = 0011 1001_2

Recursos d'aprenentatge FCO

- Poliformat, secció "Recursos"
  - Exercicis sense solució.
  - Solucions als exercicis.
  - **Entrenador personal d'aritmètica.**
  - **Entrenador personal d'enters.**
  - **Convertidor IEEE754.**
  - **Entrenador personal de coma flotant.**
  - Exàmens d'anys anteriors.

- Poliformat, secció "Continguts"
  - Mòdul 12: *Operaciones básicas en binario.* (Teoria i exercicis)
  - Mòdul 13: *Representación de números enteros*. (Teo. i exercicis)
  - Mòdul 14: *Representación de números en coma flotante*.

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

DISCA

Fonaments de computadors

Tema 5. REPRESENTACIÓ DE LA INFORMACIÓ