Organización de Computadoras



Clase 2

# Temas de Clase

* Representación de datos
* Números con signo
* Operaciones aritméticas
* Banderas de condición
* Representación de datos alfanuméricos

# Representación en BCS

Con n bits, 1 bit representa al signo y n-1 bits a la magnitud

n-1 n-2 0

|  |  |
| --- | --- |
| SIGNO | MAGNITUD |

El bit n-1 (extremo izquierdo) representa sólo al signo

Los bits 0 a n-2 la magnitud

Un 0 en el bit de signo indica que el número es positivo

Un 1 en el bit de signo indica que el número es negativo

Los bits 0  n-2 representan el valor absoluto en binario

El rango: -(2n-1 – 1)  +(2n-1 – 1) con 2 ceros

# (2)

Ejemplos +

+3210= 00100000 -3210= 10100000

## 32 32

|  |  |
| --- | --- |
| +710=00000111 | - 710=10000111 |
| +4110=00101001 | -4110=10101001 |

# (3)

Ejemplo: n=8 bits

Números 11111111  -(2n-1 – 1)=-127 negativos ...

10000000  - 0

Números 01111111 +(2n-1 –1)=+127 positivos ...

00000000  +0

(4)

Ejemplo con n= 3 bits

111 = -3 = -(2n-1 –1)

110 = -2 101 = -1 100 = -0

011 = +3 = +(2n-1 – 1)

010 = +2

001 = +1

000 = +0

# Resumen: BCS

El intervalo es simétrico

El primer bit sólo indica el signo

Los positivos empiezan con cero (0)

Los negativos empiezan con uno (1)

Hay dos ceros

Números distintos: 2n

# Técnica de Complementos

* El complemento a un número N de un número A (A menor que N) es igual a la cantidad que le falta a A para ser N

Complemento a N de A = N - A

* El complemento a un número N del número (N-A) es igual a A.

Complemento a N de (N-A) = N - (N-A) = A

# Técnica de Complementos (2)

En un sistema con n dígitos podemos tener:

* Complemento a la base disminuida
* si N= basen – 1

En sistema binario es Complemento a 1 ó Ca1

* Complemento a la base
* si N= basen

En sistema binario es Complemento a 2 ó Ca2

# Representación en Ca1

Los n bits representan al número

n-1 0

Número

## Información del signo

Si el número es positivo, los n bits tienen la representación binaria del número

(como siempre )

Si el número es negativo, los n bits tienen el Ca1 del valor deseado. El Ca1 de un número en base 2 se obtiene invirtiendo todos los bits

•Los positivos empiezan con cero (0)

•Los negativos empiezan con uno (1)

•El rango va desde

– (2n-1 –1) a +(2n-1 –1) con dos ceros

Ejemplos

+

+3210= 00100000 -3210=11011111

+710= 00000111 -710= 11111000

+4110= 00101001 -4110=11010110 Ejemplo: n=8 bits

Números 11111111  -0 negativos ...

10000000  -(2n-1 –1 )=-127 Números 01111111 +(2n-1 –1)=+127 positivos ...

00000000  +0

Ejemplo con n= 3 bits

111 = -0

110 = -1 101 = -2

100 = -3= -(2n-1 –1 )

011 = +3 = +(2n-1 – 1)

010 = +2

001 = +1

000 = +0

Dada una cadena de bits ¿qué número decimal representa si lo interpretamos en Ca1?

Cuando es positivo:

01100000 = 1 x 26 + 1 x 25 = 64+32= 96

## Como siempre

Cuando es negativo, puedo hacer dos cosas: Ca1 del número y obtengo el positivo Ej.

11100000 = - 31

11100000 00011111 = +31 Otro método: el peso que tiene el primer dígito ahora es –(2n-1 –1 ) y el resto de los dígitos con pesos positivos como siempre

11100000= -1x(27 –1) + 1x26 + 1x25=

= - 127 + 64 + 32 = -31

## O por definición de Complemento a la base disminuida

 Ca1 = (bn –1) - No

# Resumen Ca1

El intervalo es simétrico

Los n bits representan al número

Los positivos empiezan con cero (0)

Los negativos empiezan con uno (1)

Hay dos ceros

Números distintos 2n

# Representación en Ca2

Los n bits representan al número

n-1 0

Número

Información del signo

# Representación en Ca2

Si el número es positivo, los n bits tienen la representación binaria del número

(como siempre)

Si el número es negativo, los n bits tienen el Ca2 del valor deseado.

El Ca2 de un número (en base 2) se obtiene invirtiendo todos los bits (Ca1) y luego sumándole 1.

Otra forma: “mirando” desde la derecha se escribe el número (base 2) igual hasta el primer “1” uno inclusive y luego se invierten los demás dígitos

## Otra forma: por definición de Complemento a la base

 Ca2 = bn - No

•Los positivos empiezan con cero (0)

•Los negativos empiezan con uno (1)

•El rango es asimétrico y va desde

– (2n-1 ) a +(2n-1 –1)

•Hay un solo cero

Ejemplos

+3210= 00100000 “mirando” desde la derecha

### - 3210= 11100000

Los dígitos en rojo se copiaron igual

Los dígitos en azul se invirtieron

# Ca2 (otra forma )

+3210=00100000

1111

11011111 invierto todos los bits

+ 1 le sumo 1

-3210=11100000 en Ca2

# Ca2 (otra forma)

* Ca2 = bn – No = 28 – 32 = 256-32=224
* Hagamos la cuenta en base 2

0 1 1

1101010 0 0 0 0 0

0 0 1 0 0 0 0 0

-32= 1 1 1 0 0 0 0 0 en Ca2

Ejemplo : n=8 bits

Números 11111111  -1 negativos ...

10000000 - (2n-1 )= -128 Números 01111111 +(2n-1 –1)=+127 positivos ...

00000000  +0

Ejemplo con n= 3 bits

111 = -1

110 = -2 101 = -3

100 = -4= -(2n-1)

011 = +3 = +(2n-1 – 1)

010 = +2

001 = +1

000 = +0

Dada una cadena de bits ¿qué número decimal representa si lo interpretamos en Ca2?

 Cuando es positivo:

01100000=1 x 26 + 1 x 25 =64+32=96

## Como siempre

Cuando es negativo, puedo hacer dos cosas:

Ca2 el número y obtengo el positivo Ej.

11100000 = - 32

11100000 00100000 = +32 Otro método: el peso que tiene el primer dígito ahora es –(2n-1) y el resto de los dígitos con pesos positivos como siempre

11100000 = -1x(27 ) + 1x26 + 1x25

= - 128 + 64 + 32 = -32

# Resumen Ca2

El intervalo es asimétrico, hay un - más

Los n bits representan al número

Los positivos empiezan con cero (0)

Los negativos empiezan con uno (1)

Hay un solo cero

Números distintos 2n

# Técnica del Exceso

* La representación de un número A es la que corresponde a la SUMA del mismo y un valor constante E (o exceso).

Exceso E de A = A + E

* Dado un valor, el número representado se obtiene RESTANDO el valor del exceso. A = (Exceso E de A) - E
* El signo del número A resulta de una resta
* En binario, NO sigue la regla del bit mas significativo

# Exceso 2n-1

Rango

-2(n-1)  x  2(n-1)-1

si n=6 **Exceso 32**

-2(6-1) = 0000002 2(6-1)-1 = 1111112

**= 0**  **- 32** **= 63 - 32 = 31**

010 = 1000002

**= 32 - 32 = 0**

Nuevas Banderas aritméticas

N (negativo): igual al bit más significativo del resultado.

Es 1 si el resultado es negativo

V (overflow): en 1 indica una condición de fuera de rango (desborde) en Ca2. El resultado no se puede expresar con el número de bits utilizado.

# Suma en Ca2

* Para sumar dos números en Ca2 se suman los n bits directamente.

Si sumamos dos números + y el resultado es – ó si sumamos dos – y el resultado es + hay overflow, en otro caso no lo hay.

* Si los Nos son de distinto signo nunca puede haber overflow.

# Resta en Ca2

* Para restar dos números en Ca2, se restan los n bits directamente. También se puede Ca2 el sustraendo y transformar la resta en suma.
* Si a un No + le restamos un No – y el resultado es – ó si a un No – le restamos un + y el resultado es + hay overflow en la resta.
* Si son del mismo signo nunca hay overflow

Operación NZVC Ca2 Sin signo

0100 0000 +4 +4 0010 +2 +2 0110 +6 +6 Los dos resultados son correctos.

Operación NZVC Ca2 Sin signo

0101 1010 +5 +5 0111 +7 +7

* 1. -4 overf. +12 Ca2 incorrecto, sin signo correcto.

Operación NZVC Ca2 Sin signo

* 1. 0101 -3 13

0011 +3 3

1 0000 0 carry 0 Ca2 correcto, sin signo incorrecto.

Operación NZVC Ca2 Sin signo

1001 0011 -7 9

1100 -4 12

1 0101 V +5 C 5 Los dos resultados son incorrectos.

# Bits de condición para la resta

Operación NZVC Ca2 Sin signo

1 0101 1001 +5 5

0111 +7 7

1110 -2 B 14 Ca2 correcto, sin signo incorrecto.

# Bits de condición para la resta

Operación NZVC Ca2 Sin signo

1001 0010 -7 9

0100 +4 4

0101 V +5 5 Ca2 incorrecto, sin signo correcto.

# Suma en BCS

1 001 -1

1 001 -1

1 010 -2

Para pensar.

# Representación alfanumérica

* Letras (mayúsculas y minúsculas)
* Dígitos decimales (0, ..., 9)
* Signos de puntuación Caracteres especiales
* “Caracteres” u órdenes de control

# Ejemplo

A cada símbolo un código en binario

Ejemplo: x, y, , , #, @, [, ]

|  |  |  |  |
| --- | --- | --- | --- |
| Ocho símbolos | | ¿Cuántos bits? ¿Por qué? | |
| 000 | x | @ | ... |
| 001 | y | [ |  |
| 010 |  |  |  |
| 011 |  | # |  |
| 100 | # |  |  |
| 101 | @ | y |  |
| 110 | [ | x |  |
| 111 | ] | ] |  |

# Algunos códigos

* FIELDATA
* 26 letras mayúsculas + 10 dígitos + 28 caracteres especiales
* Total 64 combinaciones  Código de 6 bits ASCII

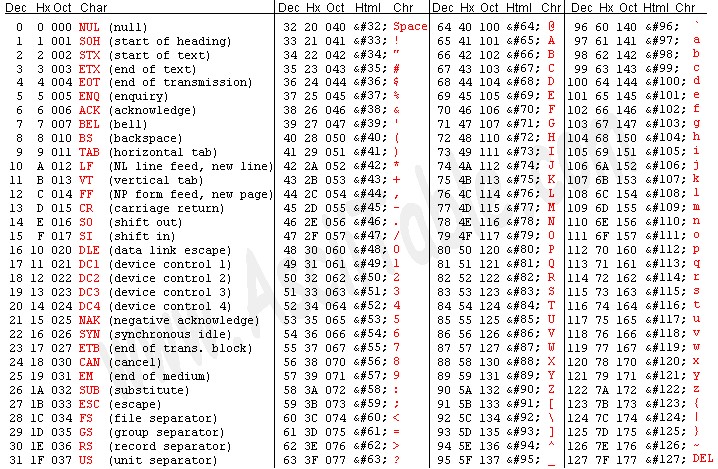
American Standard Code for Information Interchange

* FIELDATA + minúsculas + ctrl
* Total 128 combinaciones  Código de 7 bits

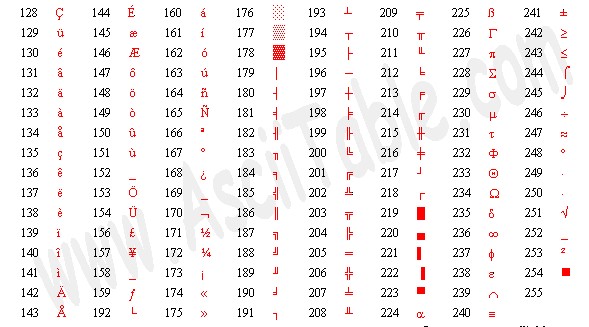
# Algunos códigos (2)

* ASCII extendido
* ASCII + multinacional + semigráficos + matemática
* Código de 8 bits
* EBCDIC - Extended BCD Interchange Code
* similar al ASCII pero de IBM
* Código de 8 bits

# Tabla ASCII



# Una extensión al ASCII



# mayor información …

* Capítulo 8: Aritmética del computador

(8.1., 8.2., 8.3.)

* Stallings, 5º Ed.
* Sistemas enteros y Punto fijo
* Apunte 1 de Cátedra
* Capítulo 3: Lógica digital y representación numérica
* Apuntes COC - Ingreso