Organización de Computadoras 2020

Práctica 2 - Explicación

Representación en Punto Fijo (Parte 2): Números con signo

Objetivos de la práctica: que el alumno sea capaz de:

- ***Representar e interpretar números con signo.***
- ***Realizar operaciones aritméticas e interpretar los flags de overflow y negativo.***

Bibliografía:

- ***"Organización y Arquitectura de Computadoras" de W. Stalling, capítulo 8.***
- ***Apunte 1 de la cátedra, "Sistemas de Numeración: Sistemas Enteros y Punto Fijo".***

Representación de números en Punto Fijo con signo. Operaciones Aritméticas

1. Represente los siguientes números en los sistemas BCS, Ca1, Ca2 y Ex2, restringidos a 8 bits. En los casos que no se pueda representar, aclarar por qué.

   0; 1; 127; 128; 255; 256; -1; -8 ; -127; -128; -199; -256; 137; 35; 100; -100; 0,5; 1,25.

Recuerde: Los positivos se representan igual en los sistemas BSS, BCS, Ca1 y Ca2 (ver representación de números en binario en el apunte). Los negativos en BCS, signo en el bit de mayor peso (0 positivos y 1 negativos) y los restantes son módulo. Los negativos en Ca1, se obtiene el BSS del número en 8 bits, y luego se cambian unos por ceros y ceros por unos. Los negativos en Ca2, se obtienen sumando 1 a la representación de Ca1, o copiando hasta el primer 1 (incluido) desde la derecha el número en BSS, y luego se cambian unos por ceros y ceros por unos. En Ex2, se suma siempre el exceso (que en n bits será 2^{n-1}) y luego se representa como BSS.

Ejemplos

Represente los números +32 y -32:

En BSS, visto en la práctica 1 la cadena +32 se representa como 00100000

En BCS, el bit más significativo (n-1), el del extremo izquierdo, representa sólo al signo y no forma parte del valor del número. El resto de los bits, 0 a n-2, representan el valor (módulo) del número en BSS.

+32 en decimal = **00100000** en BCS
-32 en decimal = **10100000** en BCS

En el primer ejemplo vemos que +32 empieza con 0 indicando que es positivo y en el segundo ejemplo -32 empieza con 1 indicando que es negativo.

En ambos ejemplos el resto de los bits son iguales pues representan al número 32 (módulo).

En CA1: Si el número es positivo los n bits tienen la representación binaria del número, es decir en este sistema los números positivos coinciden con la representación en BCS y BSS.

+32 en base 10= **00100000** en Ca1

Si el número es negativo, el mismo es representado por el complemento a 1 del número deseado. Para obtener el Ca1 de un número podemos utilizar la siguiente regla práctica: invertir todos los bits.

Para representar – 32 en Ca1, se invierten todos los bits de +32

-32 en decimal = **11011111** en Ca1

En CA2: Si el número es positivo los n bits tienen la representación binaria del número, es decir en este sistema los números positivos coinciden con la representación en BCS, BSS y Ca1

+32 en base 10= **00100000** en Ca2

Si el número es negativo, podemos obtener el Ca1 invirtiendo todos los bits y luego sumarle 1.

-32 en decimal = 11011111 en Ca1

Luego 11011111
+ 00000001
11100000

-32 en decimal = **11100000** en Ca2

En Ex2: al número dado le sumo el exceso 2 elevado a la N -1 (N cantidad de bits, en este caso 8) y luego lo represento en BSS:

+32 en exceso?

Primero escribimos +32 en Ca2 00100000

Luego sumamos el exceso 10000000

00100000
+ 10000000
10100000 => la cadena en Exceso es 10100000

-32 en exceso?

Primero escribimos -32 en Ca2, calculado antes:

-32 en Ca2 = 11100000

Luego le sumamos el exceso 10000000

11100000
+ 10000000
01100000 con carry= 1 que se desprecia => la cadena en Exceso es **01100000**

Recordar que puede ocurrir que el número dado no se pueda representar debido al sistema dado restringido a 8 bits, puede ocurrir que sea porque el número está fuera del rango de representación.

Enlaces asociados :

- Binario Sin Signo (BSS) y Binario Con Signo (BSS, BCS) Rango BCS
- Complemento a 1 Rango Ca1

- Complemento a 2 Rango Ca2
- Exceso Rango Exceso
- Rango de los sistemas restringidos a n bits
- Ejemplo Ejemplo 2

2. Interprete las siguientes cadenas de 8 bits en los sistemas BSS, BCS, Ca1, Ca2 y Ex2.
00000000; 01010101; 10000000; 11111110; 11111111; 10101010; 01111111; 01100110

Recuerde: En Ex2, se interpreta como BSS y luego se resta el exceso (que en n bits es 2^{n-1}).

Ejemplo 1

00001011:

En BSS, aplicamos el teorema fundamental de la numeración:

(1 * 2 elevado a la 3) + (1 * 2 elevado a la 1) + (1 * 2 elevado a la 0) = 8 + 2 + 1 = **11**

En BCS, nos fijamos el signo en el bit más significativo, en este caso es 0, por lo cual el número es positivo.

Luego el valor del número se interpreta de la misma forma que en BSS para el resto de los bits: entonces el resultado en BCS es **+11**

En Ca1, como el número empieza con cero el valor coincide con los anteriores: **+11**

En Ca2, como el número empieza con cero el valor coincide con los anteriores: **+11**

En Exceso: como el dígito más significativo es 1, el número es negativo.

Como el sistema está restringido a 8 bits, es Exceso 2 elevado a la (8-1) = Exceso 128;

Si lo calculo en decimal, al número decimal obtenido de la cadena representada en BSS le resto el exceso:
11 – 128 = **-117**

Otra forma: cambio el bit más significativo y me fijo que valor es en CA2.

En el número dado 00001011 invierto el bit más significativo 10001011

Luego me fijo su valor en CA2, como empieza con 1 es negativo, un método posible es copiar de derecha a izquierda hasta el primer 1 y luego invierto el resto de los dígitos, obtenemos 01110101, este número en decimal es el 117, entonces el número original:

10001011 es el -117; por lo tanto el valor dado en Exceso en **-117**

Ejemplo 2

10001011:

En BSS, aplicamos el teorema fundamental de la numeración:

(1* 2 elevado a la 7) + (1 * 2 elevado a la 3) + (1 * 2 elevado a la 1) + (1 * 2 elevado a la 0) = 128 + 8 + 2 + 1 = **139**

En BCS, nos fijamos el signo en el bit más significativo, en este caso es 1, por lo cual el número es negativo:

Luego el valor del número se interpreta de la misma forma que en BSS (sin considerar el bit de signo): entonces aplicando el teorema:

(1 * 2 elevado a la 3) + (1 * 2 elevado a la 1) + (1 * 2 elevado a la 0) = 8 + 2 + 1 = **11**

Como vimos que el bit más significativo del número es negativo, el resultado en BCS es - **11**

En Ca1, como el número 10001011 empieza con 1, sabemos que es negativo. Para obtener el valor debo complementarlo (invierto el valor de todos sus bits)

Obtengo 01110100 y luego interpreto este número en BSS: 64 + 32 + 16 + 4 = 116

Entonces, el Ca1 de 10001011 es **-116**

En Ca2, igual que en Ca1 y BCS, como el número 10001011 empieza con 1, sabemos que es negativo.

Método 1: Para obtener el valor debo complementarlo (invierto el valor de todos sus bits) y luego sumarle 1:

Invierto y obtengo 01110100 ; luego:

```
   01110100
+  00000001
   01110101
```

y luego interpreto este número en BSS: 64 + 32 + 16 + 4 + 1= 117

Entonces, el Ca2 de 10001011 es **-117**

Método 2: en el número dado copio los valores de izquierda a derecha hasta encontrar el primer 1 que también copio:

En este caso en el número 10001011 el bit menos significativo ya es 1, solo copio ese valor, y el resto los invierto

Nos queda 01110101**,** y luego interpreto este número en BSS, obtenemos también 117.

Entonces, el Ca2 de 10001011 es **-117**

En Exceso: el dígito más significativo es 1 => el número es positivo.

1) Cambio el bit más significativo: 00001011
2) Me fijo su valor en CA2: 11 => La cadena dada en exceso representa el valor **11**

Otra forma es directamente al valor en BSS obtenido antes restarle el exceso: 139 – 128 = **11**.

Enlaces asociados :

- Ejemplo
- Ejemplo 2
- Ejemplo 3

3. Calcule el rango y resolución de un sistema de punto fijo en BCS con 1 bit de signo, 5 bits de parte entera y 4 de fraccionaria.

El primer dígito indica el signo, 0 positivo y 1 negativo
Para la parte entera del número tenemos 5 bits,
Para la parte fraccionaria del número tenemos 4 bits

Para obtener el número más grande en BCS,, el signo debe ser positivo y debemos maximizar parte entera y fraccionaria, pues debe ser el "positivo más grande":

El número más grande será 0 11111 1111, por teorema fundamental de la numeración:

16 + 8 + 4 + 2 + 1 + 0,5 + 0,25 + 0,125 + 0,0625 = **31,9375**

Para obtener el número más chico, debemos cambiarle el signo al número y maximizar como en el caso anterior la parte entera y fraccionaria, pues debe ser el "negativo más grande"

El número más chico será 1 11111 1111, el mismo pero negativo: **-31,9375**

Enlaces asociados :

- Resolución en sistemas de Punto Fijo
- Ejemplo

4. Represente los siguientes números en el sistema del ejercicio 3. Si no es posible obtener una representación exacta, indique cuál es la más próxima y calcule en ese caso el error cometido. Si el número a representar está fuera del rango del sistema, señale que ese número "NO SE PUEDE REPRESENTAR".

1,2; 1,25 ; 35 ; -1,25 ; 1,0625 ; -1,5625 ; -35,5

Ejercicio similar al de ejercicio 1 de práctica 1, pero para el sistema indicado.

Enlaces asociados:

- Error absoluto y error absoluto máximo
- Error Relativo

5. Interprete las siguientes cadenas en el sistema del ejercicio 3.

0000000000; 0101010101; 1000000000; 1111111110; 1111111111; 1010101010; 0111111111; 0110001100

Para resolverlo, aplicar el teorema fundamental de la numeración

Ejemplo: **1010101010** para verlo más claramente separo bit de signo, parte entera y parte fraccionaria:

1 01010 1010

El número es negativo por el bit de signo en 1.

Luego, por teorema: (1*2 elevado a la 3) + (1 * 2 elevado a la 1) + (1 * 2 elevado a la -1) + 1 * 2 elevado a la -3)

- (8 + 2 + 0,5 + 0,125) = **10,625**

6. Calcule el resultado de realizar las sumas (ADD) y restas (SUB) indicadas en la tabla. Calcule el valor en que quedarán los flags luego de realizada cada operación, de acuerdo a que haya habido acarreo (flag C, de Carry) o se haya producido borrow (flag B, es el mismo que C pero en la resta), o que el resultado sea cero en todos sus bits (flag Z, de Zero)), se haya producido desbordamiento (flag V, de oVerflow), o de un resultado negativo (flag N, de Negative).

ADD 00011101 00011011	**ADD 01110000 11110001**	**SUB 00011101 00011011**	**SUB 01110000 11110001**
ADD 10011101 01110010	**ADD 01001100 01110000**	**SUB 10011101 01110010**	**SUB 01001100 01110000**
ADD 01110110 01110001	**ADD 11001100 11110000**	**SUB 01110110 01110001**	**SUB 11001100 11110000**
ADD 10111001 11100011	**ADD 10000000 10000000**	**SUB 10111001 11100011**	**SUB 10000000 10000000**
ADD 00111010 00001111	**ADD 00000000 10000000**	**SUB 00111010 00001111**	**SUB 00000000 10000000**

Recuerde que, además de acarreo (ver práctica 1), tendremos casos de exceso en el rango de representación (llamado overflow) si a un número positivo se le suma otro positivo y da un resultado negativo ó a un número negativo se le suma otro negativo y da uno positivo ó a un número positivo se le resta otro negativo y da uno negativo ó a un número negativo se le resta otro positivo y da uno positivo.

En todos estos casos de errores en la operación aritmética, se advierte el error pues la ALU encenderá (pondrá en 1) el flag de overflow (V=1). Es de hacer notar que el flag V se encenderá aunque sumemos números sin signo (en BSS), la interpretación de los flags corre por cuenta del programador.

Ejemplo de suma:

```
   1111        □ Acarreos
   01001111
+  01111000
   11000111
```

Flags: Carry=0; Zero=0; Negative=1; oVerflow=1.

Flags:

Z (cero): este bit vale 1 si el resultado de la suma ó resta son todos bits 0.

C (carry): en la suma este bit vale 1 si hay acarreo en el bit más significativo y en la resta vale 1 si hay borrow hacia el bit más significativo.

N (negativo): igual al bit más significativo del resultado. Si el resultado empieza con 1, entonces N=1

V (overflow): cuando la suma ó resta involucra a números con signo (Ca2), V=1 indica una condición de fuera de rango (desborde), quiere decir que la cantidad de bits disponibles para expresar el resultado no son suficientes.

Enlaces asociados :

- Flag de Signo
- Flag de Overflow
- Ejercicios

7. Suponga que los operandos del ejercicio anterior eran números representados en BSS, BCS, Ca1, Ca2 y Exceso2 (todos para cada sistema de representación). Verifique la correctitud del resultado interpretando el resultado obtenido y comparando con el resultado esperado. En caso de que la operación haya dado resultado incorrecto, indicar la posible cadena de bits que representa el resultado correcto.

Del ejemplo anterior, los operandos y resultado son interpretados como cadenas de bits BSS.

```
   1111        □ Acarreos   | Interpretación en BSS
   01001111                 |    79
+  01111000                 | +  120
   11000111                 |   199
```

Por lo que, si verificamos realizando a mano la operación interpretada en base 10, el resultado es correcto.

Volviendo al ejemplo, si interpretamos ahora los operandos y resultados como cadenas de bits en los 4 sistemas de representación de números con signo, tendremos:

	Acarreos	BCS	Ca1	Ca2	Exceso	Observe los flags!
1111						
01001111		79	79	79	-49	
+ 01111000		+ 120	+ 120	+ 120	+ -8	
11000111		-71	-56	-57	71	

Flags: Carry=0; Zero=0; Negative=1; oVerflow=1.

8. Sobre la operación ADD del punto anterior (el 7): Observando cuáles resultados fueron correctos y cuáles fueron incorrectos y relacionándolos con los flags trate de descubrir una regla para determinar la correctitud de la operación ADD en el sistema **BSS** con la mera observación de los flags.

Los flags toman en el ejemplo los siguientes valores: Carry=0; Zero=0; Negative=1; oVerflow=1.

Que los flags V y N estén en 1 no importa pues asumimos que estamos operando con números sin signo (BSS). Si hacemos lo mismo con todos los ejercicios, observaremos que en los casos en que **C=1 el resultado es incorrecto**, independientemente de los demás flags.

Tener en cuenta entonces que en BSS, si C= 1 el resultado es incorrecto.

9. Trabaje de forma similar al punto anterior (el 8) pero con la operación SUB. Luego trate de descubrir reglas análogas para ADD y SUB para el sistema Ca2, basándose en los ejercicios cuya cadena resultado es diferente de la correcta y observando los flags. Observe qué flags se encienden en los casos que da incorrecto y cuáles no, como así también los que es indistinto que tengan valor uno o cero.

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 números son de distinto signo nunca puede haber overflow.

Si sumo en Ca2 y hay Overflow el resultado en Ca2 es incorrecto, independientemente que haya o no haya Carry.

Resta en Ca2 Para restar dos números en Ca2, se restan los n bits directamente. También se puede hacer Ca2 el sustraendo y transformar la resta en suma. Si a un Número + le restamos un Número – y el resultado es – ó si a un Número – le restamos un + y el resultado es + hay overflow en la resta. Si son del mismo signo nunca hay overflow.

Si resto en Ca2 y hay Overflow el resultado en Ca2 es incorrecto, independientemente que haya o no haya Carry.

10. Considere en los ejercicios del punto 6 un punto fraccionario entre el bit 2 y el 3. Interprete los operandos y resultados como Ca2. Observe los flags. ¿Qué concluye?

Que haya un punto fraccionario no cambia las reglas deducidas en el ejercicio anterior.

11. Escriba las cadenas de los sistemas BCS, Ca1, Ca2 y $Ex2^{(n-1)}$ restringido a 4 bits. Obtenga el rango de cada sistema en 4 bits y para n bits. ¿Cuántas cadenas se pueden escribir en cada caso?. ¿Cuántos números se pueden representar en los distintos sistemas?

Ejemplo, 3 bits

Recordar:
BCS, Ca1 y Ca2 positivos, coinciden con el binario sin signo
Ca1: negativos, invierto los dígitos del positivo correspondiente
Ca2 negativos, Ca1 del negativo correspondiente + 1
Exc2, al Ca2 le sumo el exceso (en este caso 100)

Decimal	BCS	Ca1	Ca2	Ex2
-4	---	---	100	000
-3	111	100	101	001
-2	110	101	110	010
-1	101	110	111	011
-0	100	111	---	---
0	000	000	000	100
1	001	001	001	101
2	010	010	010	110
3	011	011	011	111

La cantidad de cadenas representables es la misma en todos los sistemas, siempre depende de la cantidad de dígitos, en este caso 3 dígitos, 2 elevado a la 3, 8 cadenas.

BCS y Ca1 tienen 2 representaciones del cero, mientras que Ca2 y Ex2 incorporan un negativo más, en este caso el -4

12. Defina el sistema Exceso a M (donde M es un entero cualquiera).

Exceso a M (M seria el valor que tomo como “mi 0” en la representación/interpretación del número) significa que de todas las combinaciones que puedo hacer en binario con una X cantidad de bits (y estos ordenados y siguiendo la notación posicional; sugiero leer https://es.wikipedia.org/wiki/Notaci%C3%B3n_posicional) tomo M-1 combinaciones para mis números negativos y 2x -M para mis números positivos, incluido el 0. Y por consiguiente todos los números en orden que son inferiores a M serán los negativos, M es el “0” y los superiores a M los positivos.
Y la pregunta es ¿Y cómo calculo el valor que representa un número? Si M es “0” entonces el valor en binario sin signo del número binario (valga la redundancia) menos M será el valor que representa. Y si lo que tengo (o se) es el valor representado entonces a este le sumo M y ese es el numero que tengo que escribir en binario sin signo.

Ejemplo:
Tomemos 4 bits (XXXX) con lo que los numeros que puedo representar en bss son del 0 al 15, es decir 16 combinaciones, ahora digo que solo me interesan 4 números negativos, entonces 0000, 0001, 0010 y 0011 serán mis negativos, el “0” será 0100 y todo número por encima de éste será positivo.
Si tengo 1100 (12) y quiero saber cuanto representa, a este le resto 0100 (M=4) y esto da 1000 que es 8 (12-4)
Si tengo 0010 (2) y quiero saber cuanto representa, a este le resto 0100 (M=4) y esto dá -2 (2-4), en binario es 1110
Si quiero representar el -3 entonces -3 + 4 = 1, escribo el 1 (0001) y si quiero representar el 5, es 5+4=9, escribo el 9 1001

Exceso a la mitad en 4 bits es 8, si las representaciones que puedo ahcer con 4 bits es 16 entonces la mitad es 8; y con 8 bits es 128.

Ahora vamos a escribir esto de manera formal, con lo que estaremos resolviendo el ejercicio:
Un número n se representa en exceso M como n + M = N, donde N es el número entero natural
Por lo tanto un número N - M = n (n = valor representado)
sugiero ver https://es.wikipedia.org/wiki/Representaci%C3%B3n_de_n%C3%BAmeros_con_signo#En_Exceso_a_K

13. Describa mecanismos para sumar y restar en BCS, Ca1 y Exceso, en base al análisis de los resultados y flags del punto 6, realizando la interpretación de los operandos y resultados en los distintos sistemas de representación citados. Observe de qué manera (qué operaciones deben realizarse y en qué caso) se llegaría al resultado correcto.

14. Interprete las siguientes cadenas descriptas en sistema Ca2. ¿Qué pasa en el caso (e)?

a. 00100110 b. 11011000 c. 00111000 d. 00000000 e. 10000000

Recordando que en CA2 las cadenas que comienzan con 0 representan a un número positivo y las que comienzan con un 1 a un número negativo entonces para encontrar el decimal representado por cada cadena procedemos de la siguiente manera:

Miramos si la cadena representa a un número positivo o negativo, en caso de ser positivo encontramos el decimal directamente.
En caso de ser negativo anotamos el signo menos para no olvidarnos y luego un método posible es aplicar el descomplemento a dos, copiando la cadena igual hasta el primer 1 (comenzando desde la derecha) y a partir de ahí cambiar unos por ceros y ceros por unos.

Ejemplos:

a. 00100110

Verificamos que por empezar con 0 esta cadena representa a un número positivo, entonces buscamos el número binario representado directamente. Obtenemos el número decimal 38.
Recordando que el número era positivo, entonces la cadena dada en CA2 representa al número decimal **+38**.

b. 11011000

Verificamos que por empezar con 1 esta cadena representa a un número negativo, entonces escribimos el signo - para no olvidarnos y descomplementamos a dos la cadena. Obtenemos esta nueva cadena 00101000. Pasando a decimal obtenemos el número 40.
Recordando que el número era negativo, entonces la cadena dada en CA2 representa al número decimal **-40**.

e. 10000000

Verificamos que por empezar con 1 esta cadena representa a un número negativo, entonces escribimos el signo - para no olvidarnos y descomplementamos a dos la cadena. En este caso obtenemos la misma cadena 10000000. Pasando a decimal obtenemos el número 128. Por lo tanto el número decimal representado es el **-128.**

Observación: Esta representación extiende el rango en el CA2 respecto al CA1 en un valor representable más llegando a poder representar el decimal -128 (en el CA1 llegábamos hasta el -127).
Lo anterior se logra al haber podido eliminar el 0 negativo del CA1, lo cual representa una ventaja del CA2 sobre el CA1.

15. Interprete las siguientes cadenas descriptas en sistema $Ex2^{(n-1)}$ con n=8. ¿Qué pasa en el caso (e)?

a. 10100110 b. 01011000 c. 10111000 d. 10000000 e. 00000000

Recordando que en Ex2 las cadenas que comienzan con 1 representan a un número positivo y las que comienzan con un 0 representan a un número negativo procedemos (al revés que en BCS, CA1 y CA2) y que el "exceso" en este caso $Ex2^{(n-1)}$ vale 10000000 entonces:

a. 10100110

La cadena representa a un número positivo entonces le restamos a la cadena el exceso

10100110 - 10000000 = 00100110, su representación en decimal es 38. => el número representado por la cadena es **+38**

b. 01011000

La cadena representa a un número negativo entonces al exceso le restamos la cadena dada:

10000000 - 01011000 = 00101000 donde su representación en decimal es 40 pero recordando que la cadena era de un número negativo nos queda **-40.**

e. 00000000

La cadena representa a un número negativo entonces al exceso le restamos la cadena dada:

10000000 - 00000000 = 10000000 donde su representación en decimal es 128 pero recordando que la cadena era de número negativo nos queda **-128.**

Observación: si recordamos que el número -128, en CA2 se representa por la cadena 10000000 y en exceso $Ex2^{(n-1)}$ por la cadena 00000000 podemos concluir que: un número decimal representado por una cadena dada en exceso difiere de el mismo número decimal representado por una cadena dada en CA2, solo en el bit de signo de dichas cadenas siendo el opuesto entre una y la otra.