**Trabajo Práctico N° 1:**

**Representación en Punto Fijo. Números sin Signo y Números con Signo. Operaciones Aritméticas. Flags.**

**Ejercicio 1.**

*Representar los siguientes números en el sistema BSS y en los sistemas BCS, Ca1, Ca2 y Ex2, todos restringidos a 8 bits. En los casos que no se pueda representar, aclarar por qué.*

*Recordar: 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. En 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 ) y, luego, se representa como BSS.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Decimal** | **BSS** | **BCS** | **Ca1** | **Ca2** | **Ex2** |
| 0 | 00000000 | 00000000 - 10000000 | 00000000 - 11111111 | 00000000 | 10000000 |
| 1 | 00000001 | 00000001 | 00000001 | 00000001 | 10000001 |
| 45 | 00101101 | 00101101 | 00101101 | 00101101 | 10101101 |
| 90 | 01011010 | 01011010 | 01011010 | 01011010 | 11011010 |
| 127 | 01111111 | 01111111 | 01111111 | 01111111 | 11111111 |
| 128 | 10000000 | --- | --- | --- | --- |
| 130 | 10000010 | --- | --- | --- | --- |
| 255 | 11111111 | --- | --- | --- | --- |
| 256 | --- | --- | --- | --- | --- |
| -1 | --- | 10000001 | 11111110 | 11111111 | 01111111 |
| -7 | --- | 10000111 | 11111000 | 11111001 | 01111001 |
| -56 | --- | 10111000 | 11000111 | 11001000 | 01001000 |
| -90 | --- | 11011010 | 10100101 | 10100110 | 00100110 |
| -127 | --- | 11111111 | 10000000 | 10000001 | 00000001 |
| -128 | --- | --- | --- | 10000000 | 00000000 |
| -139 | --- | --- | --- | --- | --- |
| 0,75 | 000000,11 | 000000,11 | 000000,11 | 000000,11 | 10000,11 |
| 2,5 | 0000010,1 | 0000010,1 | 0000010,1 | 0000010,1 | 1000010,1 |

En los casos que no se puede representar, es debido al rango de representación de los diferentes sistemas:

BSS: 0 X - 1= [0, 255].

BCS: -( - 1) X - 1= [-127, 127].

Ca1: -( - 1) X - 1= [-127, 127].

Ca2: - X - 1= [-128, 127].

Ex2: - X - 1= [-128, 127].

**Ejercicio 2.**

*Interpretar las siguientes cadenas de 8 bits en los sistemas BSS, BCS, Ca1, Ca2 y Ex2.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Cadena** | **BSS** | **BCS** | **Ca1** | **Ca2** | **Ex2** |
| 00000000 | 0 | 0 | 0 | 0 | -128 |
| 00000001 | 1 | 1 | 1 | 1 | -127 |
| 11111110 | 254 | -126 | -1 | -2 | 126 |
| 01111111 | 127 | 127 | 127 | 127 | -1 |
| 11111111 | 255 | -127 | 0 | -1 | 127 |
| 00010001 | 17 | 17 | 17 | 17 | -111 |
| 10011001 | 153 | -25 | -102 | -103 | 25 |
| 10101010 | 170 | -42 | -85 | -86 | 42 |
| 01100110 | 102 | 102 | 102 | 102 | -26 |

**Ejercicio 3.**

*Calcular el rango y resolución de un sistema de punto fijo en BSS con 7 bits de parte entera y 3 de fraccionaria y de un sistema de punto fijo en BCS con 1 bit de signo, 5 bits de parte entera y 4 de fraccionaria.*

|  |  |  |
| --- | --- | --- |
| **Sistema** | **Rango** | **Resolución** |
| BSS - 7 bits parte entera y 3 bits parte fraccionaria | [0; 127,875] | = 0,125 |
| BCS - 1 bit de signo, 5 bits parte entera y 4 bits parte fraccionaria | [-31,9375; 31,9375] | = 0,0625 |

**Ejercicio 4.**

*Representar los siguientes números en los sistemas del Ejercicio 3. Si no es posible, obtener una representación exacta, indicar cuál es la más próxima y calcular, en ese caso, el error cometido. Si el número a representar está fuera del rango del sistema, señalar que ese número “NO SE PUEDE REPRESENTAR”.*

|  |  |  |
| --- | --- | --- |
| **Número** | **BSS - 7 bits parte entera y 3 bits parte fraccionaria** | **BCS - 1 bit de signo, 5 bits parte entera y 4 bits parte fraccionaria** |
| 7 | 0000111 000 | 0 00111 0000 |
| 15,125 | 0001111 001 | 0 01111 0010 |
| 2,2 | 0000010 010 | 0 00010 0011 |
| 8,001 | 0001000 000 | 0 01000 0000 |
| 123,25 | 1111011 010 | --- |
| 50,5 | 0110010 100 | --- |
| 120 | 1111000 000 | --- |
| 1,2 | 0000001 010 | 0 00001 0011 |
| 1,25 | 0000001 010 | 0 00001 0100 |
| 35 | 0100011 000 | --- |
| -1,25 | --- | 1 00001 0100 |
| 1,0625 | 0000001 001 | 0 00001 0001 |
| -1,5625 | --- | 1 00001 1001 |
| -35,5 | --- | --- |

**Ejercicio 5.**

*Interpretar las siguientes cadenas en los sistemas del Ejercicio 3.*

|  |  |  |
| --- | --- | --- |
| **Cadena** | **BSS - 7 bits parte entera y 3 bits parte fraccionaria** | **BCS - 1 bit de signo, 5 bits parte entera y 4 bits parte fraccionaria** |
| 0000000000 | 0 | 0 |
| 0101010101 | 42,625 | 21,3125 |
| 1000000000 | 64 | 0 |
| 1111111110 | 127,75 | -31,875 |
| 1111111111 | 127,875 | -31,9375 |
| 1010101010 | 85,25 | -10,625 |
| 0111111111 | 63,875 | 31,9375 |
| 0110110110 | 54,75 | 27,375 |

**Ejercicio 6.**

*Representar los números 0, 1, 3, 8, 12, 13, 22, 35, 99, 100 y 1255 en los sistemas BCD y BCD empaquetado. Describir, con el mayor nivel de detalle posible, un procedimiento para calcular sumas en BCD. Sin considerar representación de signo, realizar las siguientes operaciones en BCD: 32 + 45; 22 + 89; 1307 + 708.*

|  |  |  |
| --- | --- | --- |
| **Número** | **BCD** | **BCD empaquetado** |
| 0 | 11000000 | 00001100 |
| 1 | 11000001 | 00011100 |
| 3 | 11000011 | 00111100 |
| 8 | 11001000 | 10001100 |
| 12 | 11110001 11000010 | 00000001 00011100 |
| 13 | 11110001 11000011 | 00000001 00111100 |
| 22 | 11110010 11000010 | 00000010 00101100 |
| 35 | 11110011 11000101 | 00000011 01011100 |
| 99 | 11111001 11001001 | 00001001 10011100 |
| 100 | 11110001 11110000 11000000 | 00010000 00001100 |
| 1255 | 11110001 11110010 11110101 11000101 | 00000001 00100101 01011100 |

Cuando la suma de los dos dígitos da mayor a 9, hay que generar el “acarreo” porque hay

seis combinaciones no usadas. Entonces, cuando la suma de los dígitos es mayor a 9, hay que sumar 6 en ese dígito.

|  |  |  |
| --- | --- | --- |
| **Suma** | **Operación** | **Resultado** |
| 32 + 45= 77 | 0011 0010  + 0100 0101 | 0111 0111 |
| 22 + 89= 111 | 0010 0010  + 1000 1001 =  1010 1011  + 0110 0110 | 0001 0001 0001 |
| 1307 + 708= 2015 | 0001 0011 0000 0111  + 0000 0111 0000 1000 =  0001 1010 0000 1111  + 0000 0110 0000 0110 | 0010 0000 0001 0101 |

**Ejercicio 7.**

*Escribir los números 13160, 2988, 927 y 87127 en los sistemas BCD, BCD empaquetado y BSS. Observar la cantidad de bits necesarios. ¿Qué conclusiones se saca respecto de las ventajas y desventajas del sistema BCD sobre BSS?*

|  |  |  |  |
| --- | --- | --- | --- |
| **Número** | **BCD** | **BCD empaquetado** | **BSS** |
| 13160 | 11110001 11110011 11110001 11110110 11000000 | 00010011 00010110 00001100 | 11001101101000 |
| 2988 | 11110010 11111001 11111000 11001000 | 00000010 10011000 10001100 | 101110101100 |
| 927 | 11111001 11110010 11000111 | 10010010 01111100 | 1110011111 |
| 87127 | 11111000 11110111 11110001 11110010 11000111 | 10000111 00010010 01111100 | 10101010001010111 |

Las conclusiones que se saca respecto de las ventajas y desventajas del sistema BCD sobre BSS son que, en aquél, es más rápida la representación, pero la cantidad de bits necesarios es mayor.

**Ejercicio 8.**

*Hacer el pasaje de binario a hexadecimal y de hexadecimal a BCH en forma directa (sin utilizar sistema decimal). ¿Por qué cree que el sistema hexadecimal es muy utilizado?*

|  |  |
| --- | --- |
| **Binario a Hexadecimal** | |
| 1010000010000 | 1410 |
| 1110001011101 | 1C5D |
| 111010011001011 | 74CB |
| 1001111100100011 | 9F23 |
| 1110101011001010 | EACA |
| 101101101011010 | 5B5A |

|  |  |
| --- | --- |
| **Hexadecimal a BCH** | |
| 2801 | 0010100000000001 |
| 1C5D | 0001110001011101 |
| 78AB | 0111100010101011 |
| F79A | 1111011110011010 |
| 7EF1 | 0111111011110001 |
| 324A | 0011001001001010 |

El sistema hexadecimal es muy utilizado porque permite representar números más grandes con menor cantidad de dígitos.

**Ejercicio 9.**

*Calcular el resultado de realizar las sumas (ADD) y restas (SUB) indicadas a continuación. Calcular el valor en el 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 dé un resultado negativo (flag N, de Negative).*

*Recordar que:*

*0+0=0 con C=0; 1+0=1 con C=0; 0-0=0 con B=0; 1-1=0 con B=0.*

*0+1=1 con C=0; 1+1=0 con C=1; 1-0=1 con B=0; 0-1=1 con B=1.*

*También, 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 se sumen números sin signo (en BSS). La interpretación de los flags corre por cuenta del programador.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Operación** | **Resultado** | **Z (zero)** | **N (negative)** | **C (carry)** | **V (overflow)** |
| 00011101  + 00011011 | 00111000 | 0 | 0 | 0 | 0 |
| 01110000  + 11110001 | 01100001 | 0 | 0 | 1 | 0 |
| 10011101  + 01110010 | 00001111 | 0 | 0 | 1 | 0 |
| 01001100  + 01110000 | 10111100 | 0 | 1 | 0 | 1 |
| 01110110  + 01110001 | 11100111 | 0 | 1 | 0 | 1 |
| 11001100  + 11110000 | 10111100 | 0 | 1 | 1 | 0 |
| 10111001  + 11100011 | 10011100 | 0 | 1 | 1 | 0 |
| 10000000  + 10000000 | 00000000 | 1 | 0 | 1 | 1 |
| 00111010  + 00001111 | 01001001 | 0 | 0 | 0 | 0 |
| 00000000  + 10000000 | 10000000 | 0 | 1 | 0 | 0 |
| 00011101  - 00011011 | 00000010 | 0 | 0 | 0 | 0 |
| 01110000  - 11110001 | 01111111 | 0 | 0 | 1 | 0 |
| 10011101  - 01110010 | 00101011 | 0 | 0 | 0 | 1 |
| 01001100  - 01110000 | 11011100 | 0 | 1 | 1 | 0 |
| 01110110  - 01110001 | 00000101 | 0 | 0 | 0 | 0 |
| 11001100  - 11110000 | 11011100 | 0 | 1 | 1 | 0 |
| 10111001  - 11100011 | 11010110 | 0 | 1 | 1 | 0 |
| 10000000  - 10000000 | 00000000 | 0 | 0 | 0 | 0 |
| 00111010  - 00001111 | 00101011 | 0 | 0 | 0 | 0 |
| 00000000  - 10000000 | 10000000 | 0 | 1 | 1 | 1 |

**Ejercicio 10.**

*Suponer que los operandos del ejercicio anterior (Ejercicio 9) eran números representados en BSS, BCS, Ca1, Ca2 y Exceso2 (todos para cada sistema de representación). Verificar 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.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Operación** | **Resultado** | **BSS** | **BCS** | **Ca1** | **Ca2** | **Ex2** |
| 00011101  + 00011011 | 00111000 | 29+27=56 | 29+27=56 | 29+27=56 | 29+27=56 | -99+(-101)=-72 |
| 01110000  + 11110001 | 01100001 | 112+241=97 | 112+(-113)=97 | 112+(-14)=97 | 112+(-15)=97 | -16+113=-31 |
| 10011101  + 01110010 | 00001111 | 157+114=15 | -29+114=15 | -98+114=15 | -99+114=15 | 29+(-14)=-113 |
| 01001100  + 01110000 | 10111100 | 76+112=188 | 76+112=-60 | 76+112=-67 | 76+112=-68 | -52+(-16)=60 |
| 01110110  + 01110001 | 11100111 | 118+113=231 | 118+113=-103 | 118+113=-24 | 118+113=-25 | -10+(-15)=-103 |
| 11001100  + 11110000 | 10111100 | 204+240=188 | -76+(-112)=-60 | -51+(-15)=-67 | -52+(-16)=-68 | 76+112=60 |
| 10111001  + 11100011 | 10011100 | 185+227=156 | -57+(-99)=-28 | -70+(-28)=-99 | -71+(-29)=-100 | 57+99=28 |
| 10000000  + 10000000 | 00000000 | 128+128=0 | 0+0=0 | -127+(-127)=0 | -128+(-128)=0 | 0+0=-128 |
| 00111010  + 00001111 | 01001001 | 58+15=73 | 58+15=73 | 58+15=73 | 58+15=73 | -70+(-113)=-55 |
| 00000000  + 10000000 | 10000000 | 0+128=128 | 0+0=0 | 0+(-127)=-127 | 0+(-128)=-128 | -128+0=0 |
| 00011101  - 00011011 | 00000010 | 29-27=2 | 29-27=2 | 29-27=2 | 29-27=2 | -99-(-101)=-126 |
| 01110000  - 11110001 | 01111111 | 112-241=127 | 112-(-113)=127 | 112-(-14)=127 | 112-(-15)=127 | -16-113=-1 |
| 10011101  - 01110010 | 00101011 | 157-114=43 | -29-114=43 | -98-114=43 | -99-114=43 | 29-(-14)=-85 |
| 01001100  - 01110000 | 11011100 | 76-112=220 | 76-112=-92 | 76-112=-35 | 76-112=-36 | -52-(-16)=92 |
| 01110110  - 01110001 | 00000101 | 118-113=5 | 118-113=5 | 118-113=5 | 118-113=5 | -10-(-15)=-123 |
| 11001100  - 11110000 | 11011100 | 204-240=220 | -76-(-112)=-92 | -51-(-15)=-35 | -52-(-16)=-36 | 76-112=-92 |
| 10111001  - 11100011 | 11010110 | 185-227=214 | -57-(-99)=-86 | -70-(-28)=-41 | -71-(-29)=-42 | 57-99=-86 |
| 10000000  - 10000000 | 00000000 | 128-128=0 | 0-0=0 | -127-(-127)=0 | (-128)-(-128)=0 | 0-0=-128 |
| 00111010  - 00001111 | 00101011 | 58-15=43 | 58-15=43 | 58-15=43 | 58-15=43 | -70-(-113)=-85 |
| 00000000  - 10000000 | 10000000 | 0-128=128 | 0-0=0 | 0-(-127)=-127 | 0-(-128)=-128 | -128-0=0 |

**Ejercicio 11.**

*Referido al Ejercicio 9 sobre la operación ADD: Observando cuáles resultados fueron correctos y cuáles fueron incorrectos y relacionándolos con los flags, describir una regla para determinar la correctitud de la operación ADD en el sistema BSS con la mera observación de los flags (sin verificar la operación pasando por el sistema decimal). Observar que, en el ejemplo dado para BSS, los flags V y N quedan en 1 y no importan, pues se supone que se está operando con números sin signo (BSS). Si se hace lo mismo con todos los ejercicios, se observará que, en los casos en que C= 1, el resultado es incorrecto, independientemente de los demás flags.*

En el sistema BSS, la correctitud de la operación ADD depende de la bandera C (carry). Si se tiene C= 1, el resultado de la operación ADD en BSS va a ser incorrecto.

**Ejercicio 12.**

*Trabajar de forma similar al Ejercicio 10, pero con la operación SUB. Luego, tratar 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. Observar 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.*

En el sistema Ca2, la correctitud de la operación ADD y SUB depende de la bandera V (overflow). Si se tiene V= 1, el resultado de estas operaciones en Ca2 va a ser incorrecto. Esta bandera será V= 1 cuando, en ADD, 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, en SUB, 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. Por otra parte, es indistinto que tengan valor uno o cero las banderas Z (zero), N (Negative) y C (carry).

**Ejercicio 13.**

*Considerar, en el Ejercicio 9, que el punto o coma fraccionaria se encuentra entre el bit 2 y el 3. Interpretar el valor que tendrán las cadenas de bits que representan los operandos y los resultados como BSS y como Ca2. Observar los flags. ¿Qué se concluye?*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Operación** | **Resultado** | **C (carry)** | **V (overflow)** | **BSS** | **Ca2** |
| 00011101  + 00011011 | 00111000 | 0 | 0 | 14 | 14 |
| 01110000  + 11110001 | 01100001 | 1 | 0 | 24,25 | 24,25 |
| 10011101  + 01110010 | 00001111 | 1 | 0 | 3,75 | 3,75 |
| 01001100  + 01110000 | 10111100 | 0 | 1 | 47 | -17 |
| 01110110  + 01110001 | 11100111 | 0 | 1 | 57,75 | -7,25 |
| 11001100  + 11110000 | 10111100 | 1 | 0 | 47 | -17 |
| 10111001  + 11100011 | 10011100 | 1 | 0 | 39 | -25 |
| 10000000  + 10000000 | 00000000 | 1 | 1 | 0 | 0 |
| 00111010  + 00001111 | 01001001 | 0 | 0 | 18,25 | 18,25 |
| 00000000  + 10000000 | 10000000 | 0 | 0 | 32 | -32 |
| 00011101  - 00011011 | 00000010 | 0 | 0 | 0,5 | 0,5 |
| 01110000  - 11110001 | 01111111 | 1 | 0 | 127 | 127 |
| 10011101  - 01110010 | 00101011 | 0 | 1 | 10,75 | 10,75 |
| 01001100  - 01110000 | 11011100 | 1 | 0 | 55 | -9 |
| 01110110  - 01110001 | 00000101 | 0 | 0 | 1,25 | 1,25 |
| 11001100  - 11110000 | 11011100 | 1 | 0 | 55 | -9 |
| 10111001  - 11100011 | 11010110 | 1 | 0 | 53,5 | -10,5 |
| 10000000  - 10000000 | 00000000 | 0 | 0 | 0 | 0 |
| 00111010  - 00001111 | 00101011 | 0 | 0 | 10,75 | 10,75 |
| 00000000  - 10000000 | 10000000 | 1 | 1 | 32 | -32 |

Por lo tanto, se concluye que, cuando C= 1, el resultado en BSS es incorrecto, mientras que, cuando V= 1, el resultado en Ca2 es incorrecto.

**Ejercicio 14.**

*Escribir todas las cadenas de los sistemas BSS, BCS, Ca1, Ca2 y restringido a 4 bits. Considerar el punto (o coma fraccionaria) fijo en cada una de todas las posibles posiciones (son 5 posibilidades en total, considerando que el punto fijo puede estar colocado a la izquierda del MSB y a la derecha del LSB) y obtener el rango y resolución de cada uno de los sistemas de punto fijo resultantes. ¿Cuántas cadenas se pueden escribir en cada caso? ¿Cuántos números se pueden representar en los distintos sistemas?*

Tabla 1 (Punto fijo en posición 1, antes del primer bit):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Cadena** | **BSS** | **BCS** | **Ca1** | **Ca2** |  |
| 0000 | 0 | 0 | 0 | 0 | -8 |
| 0001 | 1 | 1 | 1 | 1 | -7 |
| 0010 | 2 | 2 | 2 | 2 | -6 |
| 0011 | 3 | 3 | 3 | 3 | -5 |
| 0100 | 4 | 4 | 4 | 4 | -4 |
| 0101 | 5 | 5 | 5 | 5 | -3 |
| 0110 | 6 | 6 | 6 | 6 | -2 |
| 0111 | 7 | 7 | 7 | 7 | -1 |
| 1000 | 8 | 0 | -7 | -8 | 0 |
| 1001 | 9 | -1 | -6 | -7 | 1 |
| 1010 | 10 | -2 | -5 | -6 | 2 |
| 1011 | 11 | -3 | -4 | -5 | 3 |
| 1100 | 12 | -4 | -3 | -4 | 4 |
| 1101 | 13 | -5 | -2 | -3 | 5 |
| 1110 | 14 | -6 | -1 | -2 | 6 |
| 1111 | 15 | -7 | 0 | -1 | 7 |
| Rango | [0; 15] | [-7; 7] | [-7; 7] | [-8; 7] | [-8; 7] |
| Resolución | = 1 | = 1 | = 1 | = 1 | = 1 |

Tabla 2 (Punto fijo en posición 2, después del primer bit):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Cadena** | **BSS** | **BCS** | **Ca1** | **Ca2** |  |
| 0000 | 0 | 0 | 0 | 0 | -4 |
| 0001 | 0,5 | 0,5 | 0,5 | 0,5 | -3,5 |
| 0010 | 1 | 1 | 1 | 1 | -3 |
| 0011 | 1,5 | 1,5 | 1,5 | 1,5 | -2,5 |
| 0100 | 2 | 2 | 2 | 2 | -2 |
| 0101 | 2,5 | 2,5 | 2,5 | 2,5 | -1,5 |
| 0110 | 3 | 3 | 3 | 3 | -1 |
| 0111 | 3,5 | 3,5 | 3,5 | 3,5 | -0,5 |
| 1000 | 4 | 0 | -3,5 | -4 | 0 |
| 1001 | 4,5 | -0,5 | -3 | -3,5 | 0,5 |
| 1010 | 5 | -1 | -2,5 | -3 | 1 |
| 1011 | 5,5 | -1,5 | -2 | -2,5 | 1,5 |
| 1100 | 6 | -2 | -1,5 | -2 | 2 |
| 1101 | 6,5 | -2,5 | -1 | -1,5 | 2,5 |
| 1110 | 7 | -3 | -0,5 | -1 | 3 |
| 1111 | 7,5 | -3,5 | 0 | -0,5 | 3,5 |
| Rango | [0; 7,5] | [-3,5; 3,5] | [-3,5; 3,5] | [-4; 3,5] | [-4; 3,5] |
| Resolución | = 0,5 | = 0,5 | = 0,5 | = 0,5 | = 0,5 |

Tabla 3 (Punto fijo en posición 3, después del segundo bit):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Cadena** | **BSS** | **BCS** | **Ca1** | **Ca2** |  |
| 0000 | 0 | 0 | 0 | 0 | -2 |
| 0001 | 0,25 | 0,25 | 0,25 | 0,25 | -1,75 |
| 0010 | 0,5 | 0,5 | 0,5 | 0,5 | -1,5 |
| 0011 | 0,75 | 0,75 | 0,75 | 0,75 | -1,25 |
| 0100 | 1 | 1 | 1 | 1 | -1 |
| 0101 | 1,25 | 1,25 | 1,25 | 1,25 | -0,75 |
| 0110 | 1,5 | 1,5 | 1,5 | 1,5 | -0,5 |
| 0111 | 1,75 | 1,75 | 1,75 | 1,75 | -0,25 |
| 1000 | 2 | 0 | -1,75 | -2 | 0 |
| 1001 | 2,25 | -0,25 | -1,5 | -1,75 | 0,25 |
| 1010 | 2,5 | -0,5 | -1,25 | -1,5 | 0,5 |
| 1011 | 2,75 | -0,75 | -1 | -1,25 | 0,75 |
| 1100 | 3 | -1 | -0,75 | -1 | 1 |
| 1101 | 3,25 | -1,25 | -0,5 | -0,75 | 1,25 |
| 1110 | 3,5 | -1,5 | -0,25 | -0,5 | 1,5 |
| 1111 | 3,75 | -1,75 | 0 | -0,25 | 1,75 |
| Rango | [0; 3,75] | [-1,75; 1,75] | [-1,75; 1,75] | [-2; 1,75] | [-2; 1,75] |
| Resolución | = 0,25 | = 0,25 | = 0,25 | = 0,25 | = 0,25 |

Tabla 4 (Coma en posición 4, después del tercer bit):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Cadena** | **BSS** | **BCS** | **Ca1** | **Ca2** |  |
| 0000 | 0 | 0 | 0 | 0 | -1 |
| 0001 | 0,125 | 0,125 | 0,125 | 0,125 | -0,875 |
| 0010 | 0,25 | 0,25 | 0,25 | 0,25 | -0,75 |
| 0011 | 0,375 | 0,375 | 0,375 | 0,375 | -0,625 |
| 0100 | 0,5 | 0,5 | 0,5 | 0,5 | -0,5 |
| 0101 | 0,625 | 0,625 | 0,625 | 0,625 | -0,375 |
| 0110 | 0,75 | 0,75 | 0,75 | 0,75 | -0,25 |
| 0111 | 0,875 | 0,875 | 0,875 | 0,875 | -0,125 |
| 1000 | 1 | 0 | -0,875 | -1 | 0 |
| 1001 | 1,125 | -0,125 | -0,75 | -0,875 | 0,125 |
| 1010 | 1,25 | -0,25 | -0,625 | -0,75 | 0,25 |
| 1011 | 1,375 | -0,375 | -0,5 | -0,625 | 0,375 |
| 1100 | 1,5 | -0,5 | -0,375 | -0,5 | 0,5 |
| 1101 | 1,625 | -0,625 | -0,25 | -0,375 | 0,625 |
| 1110 | 1,75 | -0,75 | -0,125 | -0,25 | 0,75 |
| 1111 | 1,875 | -0,875 | 0 | -0,125 | 0,875 |
| Rango | [0; 1,875] | [-0,875; 0,875] | [-0,875; 0,875] | [-1; 0,875] | [-1; 0,875] |
| Resolución | = 0,125 | = 0,125 | = 0,125 | = 0,125 | = 0,125 |

Tabla 5 (Coma en posición 5, después del cuarto bit):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Cadena** | **BSS** | **BCS** | **Ca1** | **Ca2** |  |
| 0000 | 0 | 0 | 0 | 0 | -0,5 |
| 0001 | 0,0625 | 0,125 | 0,0625 | 0,0625 | -0,4375 |
| 0010 | 0,125 | 0,25 | 0,125 | 0,125 | -0,375 |
| 0011 | 0,1875 | 0,375 | 0,1875 | 0,1875 | -0,3125 |
| 0100 | 0,25 | 0,5 | 0,25 | 0,25 | -0,25 |
| 0101 | 0,3125 | 0,625 | 0,3125 | 0,3125 | -0,1875 |
| 0110 | 0,375 | 0,75 | 0,375 | 0,375 | -0,125 |
| 0111 | 0,4375 | 0,875 | 0,4375 | 0,4375 | -0,0625 |
| 1000 | 0,5 | 0 | -0,4375 | -0,5 | 0 |
| 1001 | 0,5625 | -0,125 | -0,375 | -0,4375 | 0,0625 |
| 1010 | 0,625 | -0,25 | -0,3125 | -0,375 | 0,125 |
| 1011 | 0,6875 | -0,375 | -0,25 | -0,3125 | 0,1875 |
| 1100 | 0,75 | -0,5 | -0,1875 | -0,25 | 0,25 |
| 1101 | 0,8125 | -0,625 | -0,125 | -0,1875 | 0,3125 |
| 1110 | 0,875 | -0,75 | -0,0625 | -0,125 | 0,375 |
| 1111 | 0,9375 | -0,875 | 0 | -0,0625 | 0,4375 |
| Rango | [0; 0,9375] | [-0,875; 0,875] | [-0,4375; 0,4375] | [-0,5; 0,4375] | [-0,5; 0,4375] |
| Resolución | = 0,0625 | = 0,125 | = 0,0625 | = 0,0625 | = 0,0625 |

Por lo tanto, las cadenas que se pueden escribir, en cada caso (BSS, BCS, Ca1, Ca2, ), dadas las 5 posibles posiciones del punto fijo, son 16. Por otra parte, la cantidad de números que se pueden representar en estos distintos sistemas son 48, 47, 47, 48 y 48, respectivamente.

**Ejercicio 15.**

*Definir el sistema Exceso a M (donde M es un entero cualquiera).*

El sistema Exceso a M es un sistema de representación numérica que utiliza un desplazamiento en la escala de números positivos para permitir la representación de números negativos. En este sistema, se asigna un valor base M y se utiliza un código que representa números positivos con valores entre 0 y M-1.

Para representar números negativos en el sistema Exceso a M, se utiliza un código que se desplaza en una cantidad fija M, de modo que el valor negativo se convierte en un valor positivo. Por lo tanto, el valor representado en el sistema Exceso a M es igual al valor real del número más M.

En general, en el sistema Exceso a M, se representa el número x como x + M en código binario, lo que permite la representación de números negativos mediante un simple desplazamiento en la escala de números positivos.

**Ejercicio 16.**

*Describir mecanismos para sumar y restar en BCS, Ca1 y Exceso, en base al análisis de los resultados y flags del Ejercicio 9, realizando la interpretación de los operandos y resultados en los distintos sistemas de representación citados. Observar de qué manera (qué operaciones deberían realizarse y en qué caso) se llegaría al resultado correcto.*

El resultado de sumar y restar es incorrecto:

* En BCS, cuando hay C= 1 o V= 1, pero no ambas. Se llegaría al resultado correcto agregando un bit.
* En Ca1, cuando hay C= 1 o V= 1 o ambas. Se llegaría al resultado correcto agregando un bit.
* En Ex2, siempre. Cuando C= 0, se llegaría al resultado correcto contemplando excesos distintos para los operandos y para el resultado, es decir, contemplando en los operandos, un exceso de y, en el resultado, un exceso de en la suma y ningún exceso en la resta.

**Ejercicio 17.**

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

|  |  |
| --- | --- |
| **Cadena** | **Interpretación** |
| 00100110 | 38 |
| 11011000 | -40 |
| 00111000 | 56 |
| 00000000 | 0 |
| 10000000 | -128 |

**Ejercicio 18.**

*Interpretar las siguientes cadenas descriptas en sistema con n= 8. ¿Qué pasa en el caso (e)?*

|  |  |
| --- | --- |
| **Cadena** | **Interpretación** |
| 10100110 | 38 |
| 01011000 | -40 |
| 10111000 | 56 |
| 10000000 | 0 |
| 00000000 | -128 |