**Organización de Computadoras 2003**

***Apunte 1: Sistemas de Numeración: Sistemas Enteros y Punto Fijo***

Los siquientes son ejercicios resueltos sobre sistemas enteros y punto fijo.

# Conversiones entre los distintos sistemas

1. Convertir el número (529)10 en su equivalente binario.

529 / 2 = 264 con resto 1 (LSD, dígito menos significativo)

264 / 2 = 132 con resto 0

132 / 2 = 66 con resto 0

66 / 2 = 33 con resto 0

33 / 2 = 16 con resto 1

16 / 2 = 8 con resto 0

8 / 2 = 4 con resto 0

4 / 2 = 2 con resto 0

2 / 2 = 1 con resto 0

1 / 2 = 0 con resto 1 (MSD, dígito más significativo)

El número se lee de abajo hacia arriba, o sea 1000010001, de modo que (529)10 = (1000010001)2

1. Convertir el número (529)10 en su equivalente octal.

529 / 8 = 66 con resto 1 (LSD)

66 / 8 = 8 con resto 2

8 / 8 = 1 con resto 0

1 / 8 = 0 con resto 1 (MSD)

El número se lee de abajo hacia arriba, o sea 1021, de modo que (529)10 = (1021)8

1. Convertir el número (529)10 en su equivalente hexadecimal.

529 / 16 = 33 con resto 1 (LSD)

33 / 16 = 2 con resto 1

2 / 16 = 0 con resto 2 (MSD)

El número se lee de abajo hacia arriba, o sea 211, de modo que (529)10 = (211)16

1. Convertir la fracción (0.371)10 en su equivalente binario.

0.371 x 2 = 0.742 con parte entera 0 (MSD, dígito más significativo)

0.742 x 2 = 1.484 con parte entera 1

0.484 x 2 = 0.962 con parte entera 0

0.962 x 2 = 1.936 con parte entera 1

0.936 x 2 = 1.872 con parte entera 1

0.872 x 2 = 1.744 con parte entera 1 0.744 x 2 = 1.488 con parte entera 1

0.488 x 2 = 0.976 con parte entera 0

0.976 x 2 = 1.952 con parte entera 1 (LSD, dígito menos significativo)

El número se lee de arriba hacia abajo, o sea 0.010111101, de modo que (0.371)10 =

(0.010111101)2

1. Convertir la fracción (0.371)10 en su equivalente octal.

0.371 x 8 = 2.968 con parte entera 2 (MSD, dígito más significativo)

0.968 x 8 = 7.744 con parte entera 7

0.744 x 8 = 5.952 con parte entera 5 (LSD, dígito menos significativo)

El número se lee de arriba hacia abajo, o sea 0.275, de modo que (0.371)10 = (0.275)8

1. Convertir la fracción (0.371)10 en su equivalente hexadecimal.

0.371 x 16 = 5.936 con parte entera 5 (MSD)

0.936 x 16 = 14.976 con parte entera 14 (LSD)

El número se lee de arriba hacia abajo, o sea 5E (E es equivalente a 14), de modo que

(0.371)10 = (0.5E)16

1. Convertir el número binario 1001.101 en su equivalente decimal.

N = 1x23 + 0x22 + 0x21 + 1x20 + 1x2-1 + 0x2-2 + 1x2-3

= 8 + 1 + 1/2 + 1/8

= 9.625

Por lo tanto, (1001.101)2 = (9.625)10

1. Convertir el número octal 1311.56 en su equivalente decimal.

N = 1x83 + 3x82 + 1x81 + 1x80 + 5x8-1 + 6x8-2

= 512 + 192 + 8 + 1 + 0.625 + 0.09375

= 713.71875

Por lo tanto, (1311.56)8 = (713.71875)10

1. Convertir el número hexadecimal 2C9.B8 en su equivalente decimal.

N = 2x162 + Cx161 + 9x160 + Bx16-1 + 8x16-2

= 2x162 + 12x161 + 9x160 + 11x16-1 + 8x16-2

= 512 + 192 + 9 + 0.6875 + 0.03125

= 713.71875

Por lo tanto, (2C9.B8)16 = (713.71875)10

1. Convertir el número hexadecimal 2E9.5C en su equivalente binario.

(2E9.5C)16 = (0010 1110 1001.0101 1100)BCH (BCH, hexadecimal codificado en binario)

= (1011101001.010111)2

1. Convertir el número octal 631.47 en su equivalente binario.

(631.47)8 = (110 011 001.100 111)BCO (BCO, octal codificado en binario)

= (110011001.1)2

1. Convertir el número binario 111100011.101101 en su equivalente hexadecimal.

(111100011.101101)2 = (001 1110 0011.1011 0100)BCH = (1E3.B4)16

1. Convertir el número binario 10111011.1011 en su equivalente octal.

(10111011.1011)2 = (010 111 011.101 100)BCO = (273.54)8

1. Convertir el número octal 134.57 en su equivalente hexadecimal.

(134.57)8 = (001 011 100.101 111)BCO = (1011100.101111)2

= (0101 1100.1011 1100)BCH = (5C.BC)16

1. Convertir el número hexadecimal F17.A6 en su equivalente octal.

(F17.A6)16 = (1111 0001 0111.1010 0110)BCH = (111100010111.1010011)2

= (111 100 010 111.101 001 100)BCO = (7427.514)8

# Operaciones aritméticas

1. Realizar la suma entre (101001011001.1111)2 y (1111100.00011)2.

1111 11 111 Acarreos

101001011001.1111

+ 1111100.00011

------------------

101011010110.00001

1. Realizar la suma entre (5131.74)8 y (174.06)8.

1 1 1 Acarreos

5131.74

+ 174.06

-------

5326.02

1. Realizar la suma entre (A59.F)16 y (7C.18)16.

12 Acarreos

A59.F

+ 7C.18

------

AD6.08

1. Realizar la resta entre (111010.001)2 y (1111.00001)2.

10 1 10 1

* + 1. ~~0~~ 10 ~~0~~ 10 0 10 10
    2. ~~1~~ ~~1~~ ~~0~~ ~~1~~ ~~0~~ . 0 0 ~~1~~ ~~0~~ ~~0~~
  + 1 1 1 1 . 0 0 0 0 1

----------------------------------

* + 1. 0 1 0 1 1 . 0 0 0 1 1

1. Realizar la resta entre (72.1)8 y (17.02)8.

* + 1. 12 0 10
    2. ~~2~~ . ~~1~~ ~~0~~
  + 1 7 . 0 2

-------------

5 3 . 0 6

1. Realizar la resta entre (3A.2)16 y (F.08)16.

* + 1. 1A 1 10
    2. ~~A~~ . ~~2~~ ~~0~~
  + F . 0 8

-------------

* + 1. B . 1 8

# Representación en complemento a la base reducida en el sistema binario (Ca1)

1. Representar en Ca1 el número –68 tomando como base una computadora con palabra de 8 bits.

El número 68 sería 01000100, siendo el dígito más significativo el bit de signo. El Ca1 de 01000100 es 10111011 (se obtiene simplemente reemplazando los ceros por unos y los unos por ceros), entonces 10111011 es equivalente a –68.

1. Dar las representaciones posibles en Ca1 del 0 tomando como base una computadora con palabra de 8 bits.

En Ca1 hay dos representaciones posibles del 0, 00000000 y 11111111.

1. Averiguar qué número decimal expresa el binario 11010011 representado en Ca1 en una palabra de 8 bits.

Recomplementando el 11010011 se obtiene el 00101100, el cual representa el número decimal 44.

1. Sumar los números –28 y +122 representándolos en Ca1 en palabras de 8 bits.

11100011 representación en Ca1 de –28

+ 01111010 representación en Ca1 de +122

--------

101011101

1 se suma el acarreo

--------

01011110

Convirtiendo el resultado a decimal obtenemos que la suma da +94.

1. Representando en Ca1 en palabras de 8 bits efectuar la operación +43-93.

00101011 representación en Ca1 de +43

+ 10011101 representación en Ca1 de -98

--------

11001000

Como el resultado es un número negativo (el bit de signo está en 1) para leerlo hay que recomplementarlo (obteniendo 00110111) y recién entonces convertirlo a decimal; en nuestro caso obtenemos como resultado –55.

# Representación en complemento a la base en el sistema binario (Ca2)

1. Representar en Ca2 el número –56 tomando como base una computadora con palabra de 8 bits.

En el sistema binario el Ca2 de un número provee, al igual que en Ca1, el equivalente negativo del número que se está complementando.

56 00111000

Ca1 11000111

+ 1

-------- Ca2 11001000 -56

1. Representar en Ca2 el número -88.

Una forma inmediata de obtener el Ca2 es recorrer los bits de derecha a izquierda y copiarlos tal cual están hasta el primer 1 inclusive, y los restantes bits tratarlos como al obtener el Ca1, o sea invertirlos, reemplazandos los ceros por unos y viceversa.

0101 1000 88

<- ->

1010 1000 -88

1. Dar las representaciones posibles en Ca2 del 0 tomando como base una computadora con palabra de 8 bits.

La ventaja de la representación en Ca2 sobre Ca1 y BCS (binario con signo o representación en signo y módulo) es que el cero tiene una sola representación, no hay cero positivos y cero negativo como sucede en los otros dos casos, porque el Ca2 de 0 es 0, independientemente de la longitud de la palabra donde esté representado.

1. ¿Qué número decimal representa el número 11010010 representado en Ca2 en una palabra de 8 bits?.

Al igual que en Ca1, un número negativo no se puede leer directamente convirtiéndolo a decimal, ya que no obtendremos el equivalente decimal del número que está representando. Para saber de qué número se trata hay que recomplementarlo y recién entonces hacer la conversión.

En nuestro caso, si recomplementamos el 11010010 obtenemos 00101110, que representa el número decimal +46.

1. Efectuar la operación 117-36 representando los números en Ca2 en palabras de 8 bits.

Al sumar dos números en Ca2 también se incluye en la operación el bit de signo. El acarreo producido por el bit de signo se desprecia (no se tiene en cuenta).

01110101 representación en Ca2 de 117

+ 11011100 representación en Ca2 de –36

--------

101010001 el acarreo se desprecia

Como el acarreo se desprecia el resultado es 01010001, que convirtiéndolo a decimal nos da 81.

1. Sumar los números –115 y +87 representándolos en Ca2 en palabras de 8 bits.

10001101 representación en Ca2 de -115

+ 01010111 representación en Ca2 de 87

--------

11100100

Como el resultado es un número negativo (el bit de signo es 1) para leerlo primero hay que recomplementarlo, obteniendo 00011100, y recién entonces convertirlo a decimal; el resultado de esta operación es –28.

# Representación en Exceso

En general los excesos son a la 2n-1, para que haya igual cantidad de números positivos y negativos. El exceso, a diferencia del Ca1 y Ca2, si empieza con 0 es negativo y si empieza con 1 es positivo. En exceso, al igual que en Ca2, existe una única representación del 0.

1. Interpretar el valor de 10110110 que se encuentra en exceso.

Si un número está en exceso y quiero saber su valor le resto el exceso (2n-1). Cuando empieza con 1 es positivo y le resto 128, es decir, 10000000 (también conocido como universal/2 o U/2).

10110110 188

* + 10000000 - 128

-------- ---

00110110 60

Por lo tanto, obtenemos que 10110110 es el 60 sin exceso.

1. Interpretar el valor de 00011101 que se encuentra en exceso.

Si un número en exceso empieza con 0 es negativo, y para saber su valor se calcula – (U/2-a).

10000000 128

* + 00011101 - 29 El 29 es el número en exceso

-------- ---

01100011 99 El –99 es el número sin exceso

Por lo tanto, obtenemos que 00011101 es el –99 sin exceso.

1. Escriba el 15 en exceso con 8 bits.

00001111 15

+ 10000000 + 128

-------- ---

10001111 143

Por lo tanto, el 15 en exceso es 10001111.

1. Escriba el -123 en exceso con 8 bits.

El 123 en binario es 01111011

10000000 128

* + 01111011 - 123

-------- ---

00000101 5

Por lo tanto, el –123 en exceso es 00000101.

# Overflow y Carry

Tanto en la representación en Ca1 como en Ca2 una operación puede dar como resultado un número que excede la capacidad de la palabra de memoria, produciéndose así el overflow.

Al sumar dos números el overflow se puede dar sólo si los dos tienen el mismo signo; la suma de dos números de distinto signo nunca dará como resultado un número con módulo mayor al de mayor módulo de los dados, al máximo será igual (al sumarle 0 a otro número), pero en general será menor, por lo tanto no puede exceder la capacidad de la palabra de memoria.

El overflow se reconoce cuando los bits de signo de los dos números que se suman son iguales entre si pero distintos del bit de signo del resultado, o sea cuando los números son positivos y da resultado negativo o viceversa. En este caso el contenido de la palabra de memoria es incorrecta.

1. Sumar 5 y 3 en representación en BSS en palabras de 3 bits.

* 1. representación en BSS de 5

+ 011 representación en BSS de 3

----

1000

Si tengo una suma en BSS y me da carry significa que el resultado es erróneo. Me da algo por afuera de lo que puedo escribir. El resultado no se puede escribir en el rango que hay.

1. Sumar 5 y 6 en representación en BCS en palabras de 4 bits.

0101 representación en BCS de 5

+ 0110 representación en BCS de 6

----

1011 representación en BCS de -2

La suma da overflow ya que al sumar los dos positivos dio negativo. El resultado es incorrecto.

1. Sumar 92 y 53 en representación en Ca1 en palabras de 8 bits.

92 01011100 representación en Ca1 de 92

+ 53 + 00110101 representación en Ca1 de 53

--- --------

145 10010001

El bit más significativo es 1, por lo tanto, hay overflow.

En la suma representada en decimal se puede deducir que habrá overflow porque el resultado es mayor que 127, que es el mayor número representable en Ca1 en una palabra de 8 bits.

1. Sumar -3 y 3 en representación en Ca2 en palabras de 3 bits.

* 1. representación en Ca2 de -3

+ 011 representación en Ca2 de 3

----

1000

Si me da carry cuando trabajo en Ca2 se desprecia, ya que el resultado queda bien.

1. Sumar -83 y -70 en representación en Ca2 en palabras de 8 bits.

-83 10101101 representación en Ca2 de -83

+ -70 + 10111010 representación en Ca2 de -70

--- --------

-153 101100111

El acarreo en Ca2 se desprecia. El bit más significativo es 1, por lo tanto, hay overflow. El overflow cuando trabajo en Ca2 significa que el resultado es erróneo.

Observando el resultado en decimal se puede asegurar que habrá overflow, ya que es más chico que –128, que es el menor número que se puede representar en Ca2 en una palabra de 8 bits.0

# Capacidad de representación, resolución y rango en sistemas restringidos a n bits

*Capacidad de representación*: Es la cantidad de números que se pueden representar. Ya sea en punto fijo o no es bn. Por ejemplo, si tengo un sistema restringido a 5 bits, sería 25 números, es decir, 32 números.

*Resolución*: Es la mínima diferencia entre un número representable y el siguiente. Se podría decir que es la diferencia entre el 0 y el siguiente. Por ejemplo, en binario con dos dígitos fraccionarios es 0.01.

*Rango*: El rango de un sistema está dado por el número mínimo representable y el número máximo representable. Por ejemplo, en binario con cinco dígitos es [0, 31] (donde el 0 es 00000 y el 31 es 11111). El número máximo representable en un sistema para la parte entera es bn-1.

1. Indicar cuál es la capacidad de representación, la resolución y el rango de un sistema BSS de 5 bits.

El número máximo representable es 11111, es decir, el 31 decimal (o 25-1).

El número mínimo representable es 00000, es decir, el 0 decimal.

Por lo tanto, el rango es [0, 31]. Como podemos ver el rango en un sistema BSS es [0 … 2n-1].

La capacidad de representación es 25, es decir, 32 números.

La resolución es 1, lo cual obtenemos al hacer la resta entre 00000 y su próximo número representable, es decir, 00001.

1. Indicar cuál es la capacidad de representación, la resolución y el rango de un sistema BCS de 5 bits.

El número máximo representable es 01111, es decir, el 15 decimal (o 24-1).

El número mínimo representable es 11111, es decir, el -15 decimal.

Por lo tanto, el rango es [-15, 15]. Como podemos ver el rango en un sistema BCS es [(2n-1-1) … 2n-1-1].

La capacidad de representación sigue siendo bn, es decir 25, 32 números (hay dos representaciones posibles del 0).

La resolución sigue siendo 1.

1. Indicar cuál es la capacidad de representación, la resolución y el rango de un sistema binario con 4 bits para la parte entera y 3 para la parte fraccionaria.

El número máximo representable es 1111.111.

Vmax = 1x23 + 1x22 + 1x21 + 1x20 + 1x2-1 + 1x2-2 + 1x2-3

= 8 + 4 + 2 + 1 + 1/2 + 1/4 + 1/8

= 15 + 0.875

= 15.875

El número mínimo representable es 0000.000, es decir, 0.

Por lo tanto, el rango es [0, 15.875].

La capacidad de representación es 27, es decir, 128 números.

La resolución es de 0.125, obteniéndolo al interprentar el valor de 0000.001. Como podemos ver, con los números fraccionarios perdemos rango pero podemos representar números con más precisión.

1. Indicar cuál es la capacidad de representación, la resolución y el rango de un sistema binario con 3 bits para la parte entera y 2 para la parte fraccionaria.

El número máximo representable es 111.11.

Vmax = 1x22 + 1x21 + 1x20 + 1x2-1 + 1x2-2

= 4 + 2 + 1 + 1/2 + 1/4

= 6 + 0.75

= 6.75

El número mínimo representable es 000.00, es decir, 0.

Por lo tanto, el rango es [0, 6.75].

La capacidad de representación es 25, es decir, 32 números.

La resolución es de 0.25, obteniéndolo al interprentar el valor de 000.01.

1. Especificar cuál es el rango de un sistema de representación binaria entera con n bits, con signo, en Ca1, Ca2 y en exceso.

El rango en Ca1 es [-(2n-1-1) … (2n-1-1)].

El rango en Ca2 es [-(2n-1-1)-1 … (2n-1-1)], o [-2n-1 … (2n-1-1)]. El rango en exceso es [-(2n-1) … (2n-1-1)].