Organización del Computador 1

Práctica 1

1er cuatrimestre 2022

Índice

1. Ejercicio 1

1.a.

$$
\begin{aligned}
33 &= 16 \times 2 + 1\\
16 &= 8 \times 2 + 0\\
8 &= 4 \times 2 + 0\\
4 &= 2 \times 2 + 0\\
2 &= 0 \times 2 + 1
\end{aligned}
$$
$\Longrightarrow\ 33_{10} = 10001_2$

$$
\begin{aligned}
33 &= 11 \times 3 + 0\\
11 &= 3 \times 3 + 2\\
3 &= 1 \times 3 + 0\\
1 &= 0 \times 3 + 1
\end{aligned}
$$
$\Longrightarrow\ 33_{10} = 1020_3$

$$
\begin{aligned}
33 &= 6 \times 5 + 3\\
6 &= 1 \times 5 + 1\\
1 &= 0 \times 5 + 1
\end{aligned}
$$
$\Longrightarrow\ 33_{10} = 113_5$

$$
\begin{aligned}
100 &= 50 \times 2 + 0\\
50 &= 25 \times 2 + 0\\
25 &= 12 \times 2 + 1\\
12 &= 6 \times 2 + 0\\
6 &= 3 \times 2 + 0\\
3 &= 1 \times 2 + 1\\
1 &= 0 \times 2 + 1
\end{aligned}
$$
$\Longrightarrow\ 100_{10} = 1100100_2$

$$
\begin{aligned}
100 &= 33 \times 3 + 1\\
33 &= 11 \times 3 + 0\\
11 &= 3 \times 3 + 2\\
3 &= 1 \times 3 + 0\\
1 &= 0 \times 3 + 1
\end{aligned}
$$
$\Longrightarrow\ 100_{10} = 10201_3$

$$
\begin{aligned}
100 &= 20 \times 5 + 0\\
20 &= 4 \times 5 + 0\\
4 &= 0 \times 5 + 4
\end{aligned}
$$
$\Longrightarrow\ 100_{10} = 400_5$

$$
\begin{aligned}
1023 &= 511 \times 2 + 1\\
511 &= 255 \times 2 + 1\\
255 &= 127 \times 2 + 1\\
127 &= 63 \times 2 + 1\\
63 &= 31 \times 2 + 1\\
31 &= 15 \times 2 + 1\\
15 &= 7 \times 2 + 1\\
7 &= 3 \times 2 + 1\\
3 &= 1 \times 2 + 1\\
1 &= 0 \times 2 + 1
\end{aligned}
$$
$\Longrightarrow\ 1023_{10} = 1111111111_2$

$$
\begin{aligned}
1023 &= 341 \times 3 + 0\\
341 &= 113 \times 3 + 2\\
113 &= 37 \times 3 + 2\\
37 &= 12 \times 3 + 1\\
12 &= 4 \times 3 + 0\\
4 &= 1 \times 3 + 1\\
1 &= 0 \times 3 + 1
\end{aligned}
$$
$\Longrightarrow\ 1023_{10} = 1101220_3$

$$
\begin{aligned}
1023 &= 204 \times 5 + 3\\
204 &= 40 \times 5 + 4\\
40 &= 8 \times 5 + 0\\
8 &= 1 \times 5 + 3\\
1 &= 0 \times 5 + 1
\end{aligned}
$$
$\Longrightarrow\ 1023_{10} = 13043_5$

1.b.

$1111_2 = (1 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$
$= 8 + 4 + 2 + 1$
$= 15$

$1111_3 = (1 \times 3^3) + (1 \times 3^2) + (1 \times 3^1) + (1 \times 3^0)$
$= 27 + 9 + 3 + 1$
$= 40$

$1111_5 = (1 \times 5^3) + (1 \times 5^2) + (1 \times 5^1) + (1 \times 5^0)$
$= 125 + 25 + 5 + 1$
$= 156$

$\mathtt{CAFE}_{16} = (12 \times 16^3) + (10 \times 16^2) + (15 \times 16^1) + (14 \times 16^0)$
$= 49152 + 2560 + 240 + 14$
$= 51966$

Nota: a la derecha del igual se interpretan todos los números en base 10.

1.c.

$17_8 = (1 \times 8^1) + (7 \times 8^0) = (15)_{10}$

$$
\begin{aligned}
15 &= 3 \times 5 + 0\\
3 &= 0 \times 5 + 3
\end{aligned}
$$

$\Longrightarrow\ 17_8 = 15_{10} = 30_5$

$\texttt{BABA}_{13} = (11 \times 13^3) + (10 \times 13^2) + (11 \times 13^1) + (10 \times 13^0) = (26010)_{10}$

$$
\begin{array}{rcrcccc}
26010 & = & 4335 & \times & 6 & + & 0 \\
4335 & = & 722 & \times & 6 & + & 3 \\
722 & = & 120 & \times & 6 & + & 2 \\
120 & = & 20 & \times & 6 & + & 0 \\
20 & = & 3 & \times & 6 & + & 2 \\
3 & = & 0 & \times & 6 & + & 3
\end{array}
$$

$\Longrightarrow \texttt{BABA}_{13} = 26010_{10} = 320230_6$

1.d.

$(10\ 01\ 01\ 10\ 10\ 10\ 01\ 01)_2 = 21122211_4$
$(001\ 001\ 011\ 010\ 100\ 101)_2 = 113245_8$
$(1001\ 0110\ 1010\ 0101)_2 = \texttt{96A5}_{16}$

2. Ejercicio 2

Consideramos que hubo acarreo en la operación cuando hay acarreo en la suma del bit más significativo. En estos casos es cuando el número ya no puede ser representado con el sistema de precisión fija.

	111111	1111		1 1
100001_2	100001_2	01111_2	$\texttt{9999}_{16}$	$\texttt{F0F0}_{16}$
+ 011110_2	+ 011111_2	+ 01111_2	+ $\texttt{1111}_{16}$	+ $\texttt{F0CA}_{16}$
---------	----------	--------	--------	---------
111111_2	1000000_2	11110_2	$\texttt{AAAA}_{16}$	$\texttt{1E1BA}_{16}$
No hubo acarreo.	Hubo acarreo.	No hubo acarreo.	No hubo acarreo.	Hubo acarreo.

3. Ejercicio 3

No puede haber acarreo mayor a 1. *Demostración pendiente.*

4. Ejercicio 4

En base b con k dígitos, el número más grande que se puede representar es $b^k - 1$. Consideremos 2 números tales que $n = m = b^k - 1$. Luego, $n \times m = (b^k - 1) \times (b^k - 1) = (b^k - 1)^2 = b^{2k} - 2b^k + 1$.

Queremos saber si este número se puede representar con $2k$ dígitos. Es decir, si es menor o igual que $b^{2k} - 1$, el número más grande que se puede representar con $2k$ dígitos.

$b^{2k} - 2b^k + 1 \leq b^{2k} - 1 \iff b^k \geq 1$ lo cual es verdadero ya que $k \geq 0$.

5. Ejercicio 5

signo+magnitud	complemento a 2	sin signo
$0_{10} = (0000\ 0000)_2$	$0_{10} = (0000\ 0000)_2$	
$-1_{10} = (1000\ 0001)_2$	$-1_{10} = (1111\ 1111)_2$	
$-1_{10} = (1000\ 0000\ 0000\ 0001)_2$	$-1_{10} = (1111\ 1111\ 1111\ 1111)_2$	
	$255_{10} = (0000\ 0000\ 1111\ 1111)_2$	$255_{10} = (1111\ 1111)_2$
	$-128_{10} = (1000\ 0000)_2$	
	$-128_{10} = (1111\ 1111\ 1000\ 0000)_2$	
	$128_{10} = (0000\ 0000\ 1000\ 0000)_2$	$128_{10} = (1000\ 0000)_2$

6. Ejercicio 6

Numerales dados

r = $1011\ 1111_2$
s = $1000\ 0000_2$
t = $1111\ 1111_2$

Complemento a 2

r = -65_{10}
s = -128_{10}
t = -1_{10}

Signo+magnitud

r = -63_{10}
s = 0_{10}
t = -127_{10}

7. Ejercicio 7

Representación en complemento a 2 con 4 bits:

$2_{10} = 0010_2$
$-5_{10} = 1011_2$
$0_{10} = 0000_2$

7.a.

Bits invertidos en el mismo sistema:

$1101_2 = -3_{10}$
$0100_2 = 4_{10}$
$1111_2 = -1_{10}$

7.b.

Dada la representación de un número en complemento a 2, para obtener la representación de su inverso aditivo (también en complemento a 2) podemos seguir estos pasos:

1. Invertir todos los bits.
2. Sumamos 1 e ignoramos el overflow (si es que hay).

8. Ejercicio 8

<table>
<tr><td></td><td colspan="4">-4</td><td colspan="4">-3</td><td colspan="4">-2</td><td colspan="4">-1</td><td colspan="4">0</td><td colspan="4">1</td><td colspan="4">2</td><td colspan="4">3</td></tr>
<tr><td>2</td><td colspan="4">overflow</td><td colspan="4">overflow</td><td>-</td><td>-</td><td>1</td><td>0</td><td>-</td><td>-</td><td>1</td><td>1</td><td>-</td><td>-</td><td>0</td><td>0</td><td>-</td><td>-</td><td>0</td><td>1</td><td colspan="4">overflow</td><td colspan="4">overflow</td></tr>
<tr><td>3</td><td>-</td><td>1</td><td>0</td><td>0</td><td>-</td><td>1</td><td>0</td><td>1</td><td>-</td><td>1</td><td>1</td><td>0</td><td>-</td><td>1</td><td>1</td><td>1</td><td>-</td><td>0</td><td>0</td><td>0</td><td>-</td><td>0</td><td>0</td><td>1</td><td>-</td><td>0</td><td>1</td><td>0</td><td>-</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>4</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
</table>

9. Ejercicio 9

Consideramos que hubo acarreo en la operación cuando hay acarreo en la suma del bit más significativo. Este acarreo puede o no ser overflow. Una regla sencilla para detectar overflow es fijarse si el acarreo que entró al bit más significativo es distinto al acarreo que salió. En ese caso hubo overflow.

No se produzca acarreo ni overflow

$$
\begin{array}{rl}
 & 0001_2 = 1_{10} \\
+ & 0010_2 = 2_{10} \\
\hline
 & 0011_2 = 3_{10}
\end{array}
$$

Se produzca acarreo pero no overflow

$$
\begin{array}{rl}
 & 111 \\
 & 0111_2 = 7_{10} \\
+ & 1110_2 = -2_{10} \\
\hline
 & 0101_2 = 5_{10}
\end{array}
$$

Se produzca acarreo y overflow

$$
\begin{array}{rl}
 & 1\ 11 \\
 & 1011_2 = -5_{10} \\
+ & 1011_2 = -5_{10} \\
\hline
 & 0110_2 = 6_{10} \neq -10_{10}
\end{array}
$$

No se produzca acarreo pero sí overflow

$$
\begin{array}{rl}
 & 1\ 1 \\
 & 0101_2 = 5_{10} \\
+ & 0101_2 = 5_{10} \\
\hline
 & 1010_2 = -6_{10} \neq 10_{10}
\end{array}
$$

Se produzca acarreo y el resultado sea cero

$$
\begin{array}{rl}
 & 1111 \\
 & 1111_2 = -1_{10} \\
+ & 0001_2 = 1_{10} \\
\hline
 & 0000_2 = 0_{10}
\end{array}
$$

No se produzca acarreo y el resultado sea cero

$$
\begin{array}{rl}
 & 0000_2 = 0_{10} \\
+ & 0000_2 = 0_{10} \\
\hline
 & 0000_2 = 0_{10}
\end{array}
$$

El resultado sea negativo y se produzca overflow

$$
\begin{array}{rl}
 & 1 \\
 & 0100_2 = 4_{10} \\
+ & 0100_2 = 4_{10} \\
\hline
 & 1000_2 = -8_{10} \neq 8_{10}
\end{array}
$$

El resultado sea negativo y no se produzca overflow

$$
\begin{array}{rl}
 & 0001_2 = 1_{10} \\
+ & 1110_2 = -2_{10} \\
\hline
 & 1111_2 = -1_{10}
\end{array}
$$

10. Ejercicio 10

Rangos de representación con k dígitos:

- Complemento a 2: $[-2^{k-1}, 2^{k-1} - 1]$
- Signo+magnitud: $[-2^{k-1} + 1, 2^{k-1} - 1]$

Podemos observar que en complemento a 2 tenemos 1 número más que podemos representar: -2^{k-1}.

11. Ejercicio 11

Utilizando el sistema de representación complemento a 2, podemos reasignar la representación del 0, es decir, el numeral compuesto de todos 0s, al número 2^{k-1}.

De esta forma el rango de representación resulta $[-2^{k-1}, 2^{k-1}] - \{0\}$. Si bien no tenemos forma de representar el 0, podemos representar exactamente 2^{k-1} números positivos y 2^{k-1} números negativos.

El total de números representables resulta $2^{k-1} + 2^{k-1} = 2^k$ y esta es exactamente la cantidad de numerales que se pueden formar con k dígitos, por lo tanto la representación es biyectiva.

12. Ejercicio 12

La afirmación es verdadera. Al tratarse de cadenas binarias, con k dígitos podemos obtener 2^k numerales distintos. Observemos que este número es par. Luego, si asignamos uno de estos numerales al 0, nos quedan $2^k - 1$ numerales para distribuir entre los números positivos y negativos. Como $2^k - 1$ es impar, no podemos dividir esta cantidad en exactamente 2 partes iguales, y por lo tanto siempre va a resultar que vamos a tener 1 numeral extra, ya sea para los positivos o los negativos.

13. Ejercicio 13

Pendiente

14. Ejercicio 14

Rango de representación: $[-2^{16}, 2^{16} - 1] = [-65536, 65535]$. Reordenamos los términos para que las sumas parciales cada 2 términos no se vayan fueran del rango de representación.

$$
\begin{array}{lcl}
\ \mathtt{7744}_{16} = (\mathtt{0111\ 0111\ 0100\ 0100})_2 = \mathtt{30532}_{10} \\
+\ \mathtt{88BD}_{16} = (\mathtt{1000\ 1000\ 1011\ 1101})_2 = \mathtt{-30531}_{10} \\
+\ \mathtt{6788}_{16} = (\mathtt{0110\ 0111\ 1000\ 1000})_2 = \mathtt{26504}_{10} \\
+\ \mathtt{9879}_{16} = (\mathtt{1001\ 1000\ 0111\ 1001})_2 = \mathtt{-26503}_{10} \\
+\ \mathtt{5499}_{16} = (\mathtt{0101\ 0100\ 1001\ 1001})_2 = \mathtt{21657}_{10} \\
+\ \mathtt{AB68}_{16} = (\mathtt{1010\ 1011\ 0110\ 1000})_2 = \mathtt{-21656}_{10} \\
\hline
\ \mathtt{0003}_{16} = (\mathtt{0000\ 0000\ 0000\ 0011})_2 = \mathtt{3}_{10}
\end{array}
$$