

ETSI Telecomunicación

Fundamentos de Computadores

Fundamentos de Computadores

Apuntes de Pak (Francisco José Rodríguez Fortuño)
ETSI Telecomunicación. Universidad Politécnica de Valencia.
Segundo cuatrimestre de 1^{er} curso
Curso 2003/2004

Contenido

- Referencia muy rápida
- Boletín de problemas

Fecha de última actualización: 26 Junio 2009

FCO

Tema 1 Codificación binaria

Pasar a binario y sumar

$$A_{16} = 10101011$$

$$13_{16} = 00010011$$

base b
con n dígitos
 $0 \rightarrow b^n - 1$
total de b^n

$$\begin{array}{r} 10101011 \\ + 00010011 \\ \hline 10111110 \end{array} \quad \begin{array}{r} 10101011 \\ - 00010011 \\ \hline 10011000 \end{array}$$

truco $A_{16} = 10_{10}$
 $A_{16} = 1010_2$

	+	-
	C S	C R
3	1 1	1 0 1
2	1 0	0 0 0
1	0 1	-1 1 1
0	0 0	-2 1 0

$F_{15} = 15$
 $10_{16} = 16$

Tema 2 Representación enteros y reales

S/M

S M

Rango $[-(2^{n-1}-1), 2^{n-1}-1]$

ej 8 bits Rango $[-127, 127]$ repite el cero $000 = +0$
 $100 = -0$

ej: 63_{10}

$63 \rightarrow 31 \rightarrow 15 \rightarrow 7 \rightarrow 3 \rightarrow 1$

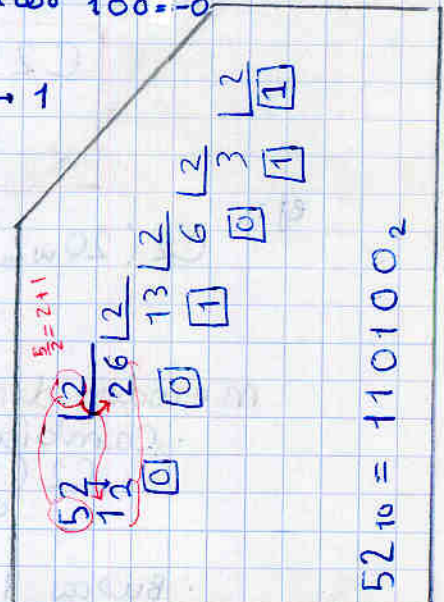
$63_{10} = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline S & & M & & & & & \end{array}$

ej: -25_{10}

$25 \rightarrow 12 \rightarrow 6 \rightarrow 3 \rightarrow 1$

$-25_{10} = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ \hline \end{array}$

ej: 129_{10} fuera de rango!



Ca 2

no repite el cero

$000 = 0$
 $100 = -4$

$V \geq 0 \rightarrow S/M \quad V = W_{10}$

$V < 0 \rightarrow 2^n - |V| = W_{10} \rightarrow W_2$

tiene que ser positivo!

tiene que dar n bits

Rango $[-2^{n-1}, 2^{n-1}-1]$

8 bits

$63_{10} = 00111111_{C2-8}$

$-25_{10} \quad W_{10} = 2^8 - |-25| = 231 = W_{10}$

$W_2 \quad 231 \rightarrow 115 \rightarrow 57 \rightarrow 28 \rightarrow 14 \rightarrow 7 \rightarrow 3 \rightarrow 1 \rightarrow 0$

$-25_{10} = 11100111_{C2-8}$

el $1111_{Ca-2} = -1_{10}$ ya que $W_2 = 2^n - |V| = 15 = 1111$



1 1 1 1 0 1 1 0 c2-8
128 64 32 16 8 4 2 1

• es negativo
 $W_{10} = 246$

$$2^8 - |V| = 246$$

$$|V| = 10$$

$$V = -10_{10}$$

0 0 1 1 0 0 0 1 c2-8

es positivo
 $W_{10} = 49$

$$V = 49$$

para pasar a mas bits hay que repetir el signo

$$1 1 1 1 0 1 1 0 \text{ c2-8} = 111111111110110 \text{ c2-16}$$

$$0 0 1 1 0 0 0 1 \text{ c2-8} = 000000000110001 \text{ c2-16}$$

Transformación C2

Cambia el signo de un número en complemento a 2

$$C2(X) = 2^n - X = Y$$

$$\uparrow$$

$$W_2 = W_{10}$$

$$\uparrow$$

$$W_2 = W_{10}$$

ej:

$$C2(20_{W_{10}}) = 2^n - 20 = 44_{W_{10}} \quad W_2 = 101100$$

$$V = -20$$

Métodos directos

• Cambiar todos y sumar uno

$$C2(11011010_2)$$

$$00100101 + 1 \quad \hline \quad 00100110$$

• Buscar 1 menos significativo y modificar a partir de ahí

$$C2(11011010_2)$$

$$00100110$$

Exceso Z

$$W_{10} = V + Z \longrightarrow W_2$$

$$Z = 2^{n-1} \text{ o } 2^{n-1} - 1$$

ej $E128 \rightarrow 2^7 \rightarrow n=8$

$E1023 \rightarrow 2^{10} - 1 \rightarrow n=11$

rango $[-Z, 2^n - 1 - Z]$

ej. $E127 [-127, 128]$

si $Z \neq 2^{n-1}$ o $Z \neq 2^{n-1} - 1$
matemáticamente se puede
pero no habrá bit de signo

$$10110110_{E127} \rightarrow W_{10} = 182 \rightarrow V = W_{10} - Z = 182 - 127 = 55$$

128 64 32 16 8 4 2 1

$$-52 \rightarrow W_{10} = -52 + 127 = 75 \rightarrow W_2 = 01001011_{E127}$$

$$100 \rightarrow W_{10} = 100 + 127 = 227 \rightarrow W_2 = 11100011_{E127}$$

$$0 = 01111111_{E127}$$

$$01111111_{E127} \rightarrow W_{10} = 127 \rightarrow V = W_{10} - Z = 127 - 128 = -1$$

Representación Reales

$$10' 25_{10} \rightarrow 1010' 01_2$$

base decimal

$$0,25 \times 2 = 0,5$$

$$0,5 \times 2 = 1$$

$$10' 75_{10} \rightarrow 1010' 11$$

$$0,75 \times 2 = 1,5$$

$$0,5 \times 2 = 1$$

$$2' 4_{10} \rightarrow 10' 0110$$

$$0,4 \times 2 = 0,8$$

$$0,8 \times 2 = 1,6$$

$$0,6 \times 2 = 1,2$$

$$0,2 \times 2 = 0,4$$

$$0,4 \times 2 = \dots$$

$$10' 0110 \cdot 2^{-3}$$

$$(2^1 + 2^{-2} + 2^{-3}) \cdot 2^{-3}$$

Normalizar: llevar la coma hasta el uno más significativo

$$1010' 11_2 \rightarrow 1' 01011_2 \cdot 2^3$$

$$10' 0110_2 \rightarrow 1' 00110_2 \cdot 2^1$$

$$101110' 11_2 \cdot 2^{20} \rightarrow 1' 0111011_2 \cdot 2^{25}$$

$$0' 001_2 \cdot 2^{-100} \rightarrow 1 \cdot 2^{-103}$$

se consigue que números iguales tengan misma representación

IEEE

S	EXP	Mantra
---	-----	--------

Simple Precisión:
Doble Precisión:

E127 → 32 bits →
E1023 → 64 bits →

1	8	23
S	EXP	M
1	11	52

nosotros usamos S.P.

ej: $1' 01011 \cdot 2^3$
¡ el uno va implícito!

S = 0
M = 010110...0
E = 3 + 127 = 130 = 10000010 E127

S	E	M
0	130	01011000000000000000000
8	1	2
C	O	O

ej: $2' 4_{10} \rightarrow 10' 0110_2$
 $\rightarrow 1' 00110_2 \cdot 2^1$
¡ el uno va implícito!

S = 0
E = 1 + 127 = 128 = 10000000 E127
M = 00110011001100110011001

Excepciones.

E₁₂₇
como le quitamos
2 códigos para
las excepciones el
rango es
[126, 127)

S E M
E = 11111111 → ERROR (desbord
div por cero)

E = 00000000 → NO NORM

$$\pm 0' M \cdot 2^{-126}$$

el cero
es rode
ceros

E = xxxxxxxx → NORM

$$\pm 1' M \cdot 2^{\text{exp}}$$

No normalizado:

hay que mover la coma para que el exponente sea -126

ej: $0'000101 \cdot 2^{-125} \rightarrow 0'00101 \cdot 2^{-126}$

implícito

$$S = 0$$

$$E = 00000000$$

$$M = 001010...0$$

se pueden representar números que no se podían representar normalizándolo.

ej: $101001 \cdot 2^{125} \rightarrow 1'01001 \cdot 2^{130}$

$$E = 11111111$$

ej: $-0'00000001 \cdot 2^{-128} \rightarrow -0'001000001 \cdot 2^{-126}$

$$S = 1$$

$$E = 00000000$$

$$M = 00000100...0$$

ej: $10000000010100...0 \rightarrow \text{no norm}$
 $\rightarrow -0'10100 \cdot 2^{-126}$

ej: $011111110000...0 \rightarrow \text{error}$

ej: $01000000011000...0 \rightarrow$

$$10000000_{E_{127}} \rightarrow W_{10} = 128 = V + Z$$

$$V = 1$$



$$\rightarrow 1'011 \cdot 2^1$$

$$2^0 2^{-1} 2^{-2} 2^{-3}$$

$$(1 + 2^{-1} + 2^{-2} + 2^{-3}) \cdot 2^1 = 2'75_{10}$$

ASCII y UNICODE

ASCII → 7 bits

$$00 \rightarrow 7F$$

$$0111 \quad 1111$$

3 bits

control: 00 → 1F

nºs: 30 → 39

letras: 41 → 5A

61 → 7A

01011

3 bits

este siempre cero

ascii a decimal

$$30 \rightarrow 0$$

$$31 \rightarrow 1$$

$$32 \rightarrow 2$$

$$\vdots$$

$$39 \rightarrow 9$$

mayus - minus

41 A

61 a

42 B

62 b

43 C

63 c

cambiar (3 bits)

UNICODE → 16 bits
compatible con ascii

Resumen: Representación de enteros y reales:

• enteros positivos: $[0, 2^n - 1]$

• signo/magnitud

S	M
---	---

 $[-(2^{n-1}-1), (2^{n-1}-1)]$

Complemento a 2

$$V \geq 0 \rightarrow W_{10} = V$$

$$V < 0 \rightarrow W_{10} = 2^n - |V|$$

Transformación C2

$$C2(W_{10}) = 2^n - W_{10}$$

$$[-2^{n-1}, 2^{n-1} - 1]$$

Truco:

$$[b_3|b_2|b_1|b_0] = b_0 \cdot 2^0 + b_1 \cdot 2^1 + b_2 \cdot 2^2 - b_3 \cdot 2^3$$



i.e. $5+4=-7$
 $5+4-6=3$ bien!!

Exceso Z

$$W_{10} = V + Z$$

$$Z = (2^{n-1} - 1) \text{ ó } (2^{n-1})$$

$$[-Z, 2^n - 1 - Z]$$

Reales

IEEE-754

simple 32 bits

1	8	23
S	E	M

NORM: $E = xxxxxxxx$

$$\pm 1^1 M \cdot 2^E$$

$$E: [Z+1, 2^n-1-Z-1]$$

no norm excep

NO NORM: $E = 00000000$
 $= -127_{E127}$

$$\pm 0^1 M \cdot 2^{-126}$$

$$-126 = -Z + 1$$

ERROR: $E = 11111111$ desbord / div por cero

doble precisión: 64 bits

S	E	M
1	11	52

se le restan 2 por lo no norm y las excep.

I. Problemas de codificación binaria de datos

1. ¿Cuál es el polinomio característico de los siguientes números? (El subíndice indica la base de numeración)

a) 489201_{10}

$$4 \cdot 10^5 + 8 \cdot 10^4 + 9 \cdot 10^3 + 2 \cdot 10^2 + 0 \cdot 10^1 + 1 \cdot 10^0$$

b) 24013_5

c) $2AF019_{16}$

a)

b)

c) $2 \cdot 16^5 + A \cdot 16^4 + F \cdot 16^3 + 0 \cdot 16^2 + 1 \cdot 16^1 + 9 \cdot 16^0$
 $2 \cdot 16^5 + 10 \cdot 16^4 + 16 \cdot 16^3 + 0 \cdot 16^2 + 1 \cdot 16^1 + 9 \cdot 16^0$

2. ¿Cuántos dígitos se requieren para contar hasta 24 en base 6? ¿Cuál sería la cuenta?

0	1	2	3	4
5	10	11	12	13
14	15	20	21	22
23	24	25	30	31
32	33	34	35	40

24

3. Obtén la cuenta hasta 33 mediante dos dígitos en base 16.

00	01	02	03	04	05
06	07	08	09	0A	0B
0C	0D	0E	0F	10	11
12	13	14	15	16	17
18	19	1A	1B	1C	1D
1E	1F	20	21	22	23

4. ¿Cuántos dígitos hacen falta para contar hasta 264750 en base 16?

$$16^4 = 65536$$

$$16^5 = 1048576$$

con $n = 5$ bits: $0 \rightarrow 1\ 048\ 575$

5. ¿Cuántos bits se requieren para contar en binario hasta 27? Obtén la cuenta binaria de 0 a 27.

0	1	10	11	100
101	110	111	1000	1001
1010	1011	1100	1101	1110
1111	10000	10001	10010	10011
10100	10101	10110	10111	11000
11001	11010	11011	11100	11101

$2^4 2^3 2^2 2^1 2^0$

6. ¿Cuál es la representación hexadecimal de cada uno de los siguientes números octales?

a) 27103

b) 102546

c) 7526

a)

$$\begin{array}{cccc} 7 & 5 & 2 & 6_8 \\ \hline 111 & 101 & 010 & 110 \\ \hline F & 5 & 6 & 6_{16} \end{array}$$

b)

c)

7. ¿Cuál es la representación octal de los siguientes números hexadecimales?

a) FEDCBA

b) A3B8

c) 011001

a)

F E D C B A ₁₆

1111	1110	1101	1100	1011	1010
7	7	5	5	6	2

₈

b)

0 1 1 0 0 1 ₁₆

0000	0001	0001	0000	0000	0001
0	0	2	1	0	0

₈

c)

8. ¿Cuál es la representación decimal de los siguientes números?

a) 1011011₂

b) 21307₈

c) 7A51D₁₆

d) 5241003₆

1 0 1 1 0 1 1 ₂

64 32 16 8 4 2 1 = 91 ₁₀

a)

2 1 3 0 7 ₈

8⁴ 8³ 8² 8¹ 1 = 8720 ₁₀

b)

7 A 5 1 D ₁₆

16⁴ 16³ 16² 16¹ 1 = 501 021

c)

d)

9. ¿Cuál es la representación del número decimal 746 en las bases siguientes?

- a) En base 4
b) En base 8
c) En base 16

a)

$$\begin{array}{r}
 746 \text{ } \overline{)4} \\
 744 \text{ } \overline{)186} \\
 \textcircled{2} \text{ } \overline{)184} \\
 \textcircled{2} \text{ } \overline{)46} \\
 \textcircled{2} \text{ } \overline{)44} \\
 \textcircled{2} \text{ } \overline{)11} \\
 \textcircled{2} \text{ } \overline{)8} \\
 \textcircled{2} \text{ } \overline{)4} \\
 \textcircled{2} \text{ } \overline{)0}
 \end{array}
 \quad
 \boxed{023222}
 \quad
 \begin{array}{c}
 4^4 \ 4^3 \ 4^2 \ 4^1 \ 4^0 \\
 = 746
 \end{array}$$

b)

$$\begin{array}{r}
 746 \text{ } \overline{)8} \\
 744 \text{ } \overline{)93} \\
 \textcircled{2} \text{ } \overline{)88} \\
 \textcircled{5} \text{ } \overline{)11} \\
 \textcircled{3} \text{ } \overline{)8} \\
 \textcircled{1} \text{ } \overline{)0}
 \end{array}
 \quad
 \boxed{1352}
 \quad
 \begin{array}{c}
 8^3 \ 8^2 \ 8^1 \ 8^0 \\
 = 746
 \end{array}$$

c)

$$\begin{array}{r}
 746 \text{ } \overline{)16} \\
 736 \text{ } \overline{)46} \\
 \textcircled{10} \text{ } \overline{)32} \\
 \textcircled{14} \text{ } \overline{)0}
 \end{array}
 \quad
 \boxed{2EA}
 \quad
 \begin{array}{c}
 16^2 \ 16^1 \ 16^0
 \end{array}$$

IMPORTANTE
ponerlo al revés

- 10 A
11 B
12 C
13 D
14 E
15 F

10. Comprueba que los resultados obtenidos en el ejercicio anterior son correctos obteniendo la representación binaria de cada resultado.

$$023222_4 = 001011101010_2$$

$$1352_8 = 001011101010_2$$

$$A E 2_{16} = 001011101010_2$$

$$\begin{array}{c}
 0 \quad 0000 \\
 1 \quad 0001 \\
 2 \quad 0010 \\
 3 \quad 0011
 \end{array}$$

11. ¿Como se representan en BCD los siguientes números decimales?

a) 4560

b) 972

c) 15283

a)

0100 0101 0110 0000

b)

1001 0111 0010

c)

0001 0101 0010 1000 0000 11 × 10

0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
	1010
	1011

12. ¿Que números decimales representan los siguientes números codificados en BCD?

a) 010110011000

b) 101001110101

c) 00110110000

a)

4 8 7

b)

9 6 4

c)

2 5 0

13. Realiza la suma en binario de los números en base dos 01100011 y 01010101. Comprueba el resultado pasando los operandos a decimal y realizando la suma.

	128	64	32	16	8	4	2	1	
	0	1	1	0	0	0	1	1	99
	0	1	0	1	0	1	0	1	85
Bits de acarreo:	1				1	1	1		
Resultado:	1 0 1 1 1 0 0 0								184

14. Realiza la suma en binario de los números en base dos 10100101 y 00110101. Comprueba el resultado pasando los operandos a decimal y realizando la suma.

	128	64	32	16	8	4	2	1	
	1	0	1	0	0	1	0	1	165
	0	0	1	1	0	1	0	1	53
Bits de acarreo:	1				1		1		
Resultado:	1 1 0 1 1 0 1 0								218

15. Realiza la suma en binario de los números en base dos 01111010 y 00011101. Comprueba el resultado pasando los operandos a decimal y realizando la suma.

	128	64	32	16	8	4	2	1	
01111010	0	1	1	1	1	0	1	0	122
00011101	0	0	0	1	1	1	0	1	29
Bits de acarreo:	1	1	1	1					
Resultado:	1 0 0 1 0 1 1 1								191

16. Realiza la suma en binario de los números en base dos 11000011 y 11010101. Comprueba el resultado pasando los operandos a decimal y realizando la suma.

	128	64	32	16	8	4	2	1	
11000011	1	1	0	0	0	0	1	1	
11010101	1	1	0	1	0	1	0	1	
Bits de acarreo:	1	1			1	1	1		
Resultado:	1 1 0 0 1 1 0 0 0								

17. Realiza la resta en binario de los números en base dos 10110011 y 10011001. Comprueba el resultado pasando los operandos a decimal y realizando la resta.

	128	64	32	16	8	4	2	1	
	1	0	1	1	0	0	1	1	179
	1	0	0	1	1	0	0	1	153
Bits de acarreo:				1	1				
Resultado:	0	0	0	1	1	0	1	0	26

$$-1 \rightarrow \frac{1}{1}$$

$$-2 \rightarrow \frac{1}{0}$$

18. Realiza la resta en binario de los números en base dos 01100011 y 00011001. Comprueba el resultado pasando los operandos a decimal y realizando la resta.

	128	64	32	16	8	4	2	1	
	0	1	1	0	0	0	1	1	99
	0	0	0	1	1	0	0	1	25
Bits de acarreo:				1	1				
Resultado:	0	1	0	0	1	0	1	0	74

19. Realiza la resta en binario de los números en base dos 11110001 y 11111001. Comprueba el resultado pasando los operandos a decimal y realizando la resta.

	128	64	32	16	8	4	2	1	
	1	1	1	1	0	0	0	1	241
	1	1	1	1	1	0	0	1	249
Bits de acarreo:	1	1	1	1	1				
Resultado:	... 1 1 1 1 1 1 0 0 0								-8

20. Realiza la resta en binario de los números en base dos 11110111 y 01111001. Comprueba el resultado pasando los operandos a decimal y realizando la resta.

	128	64	32	16	8	4	2	1	
	1	1	1	1	0	1	1	1	247
	0	1	1	1	1	0	0	1	121
Bits de acarreo:	1	1	1	1					
Resultado:	0 1 1 1 1 1 1 0								126

21. ¿Cuales son las representaciones de los caracteres de la primera columna de la tabla en los códigos ASCII y UNICODE? Obtén los códigos en decimal, binario y hexadecimal

	ASCII			UNICODE		
	Dec	Bin	Hex	Dec	Bin	Hex
a	97	01100001	61	97	0110	0061
y	89		59			
*			2A			
~	126		7E			
ñ	—	—	—			00F1
ú	—	—	—			00FA

II. Problemas de representación de enteros y reales

1. ¿Cuál es la codificación en signo magnitud con 8 bits de los siguientes números?

a) 114

b) -118

c) -130

a) + 114
0 1110010

b) - 118
1 1110110

c) - 130
fuera de rango!

rango

$$\begin{array}{c} n \\ \hline 1 \quad k = n-1 \\ 0 \rightarrow 2^k - 1 \end{array}$$

$$[-(2^{(n-1)} - 1), 2^{n-1} - 1]$$

$$\begin{array}{rcl} 114 & \div 2 & \\ \textcircled{0} & 57 & \div 2 \\ \textcircled{1} & 28 & \div 2 \\ \textcircled{0} & 14 & \div 2 \\ \textcircled{0} & 7 & \div 2 \\ \textcircled{1} & 3 & \div 2 \\ \textcircled{1} & 1 & \end{array}$$

$$\begin{array}{rcl} 118 & \div 2 & \\ \textcircled{0} & 59 & \div 2 \\ \textcircled{1} & 29 & \div 2 \\ \textcircled{1} & 14 & \div 2 \\ \textcircled{0} & 7 & \div 2 \\ \textcircled{1} & 3 & \div 2 \\ \textcircled{1} & 1 & \end{array}$$

2. Obtén la representación en signo-magnitud de 6 bits de los numeros enteros en el intervalo $[-5, +4]$

00000

10101	1 0100	1 0011	1 0010	1 0001	1 0000	0 0001	0 0010	0 0011	0 0100
-5	-4	-3	-2	-1	0	1	2	3	4

3. ¿Cuales son los rangos de representación en signo-magnitud para 6, 10 y 16 bits?

$$\begin{array}{c}
 \boxed{n} \\
 \underbrace{\hspace{1cm}}_{k=n-1} \rightarrow 0 \rightarrow 2^k - 1 \\
 \underbrace{\hspace{1cm}}_{k=n-1} \rightarrow [- (2^k - 1) \rightarrow 2^k - 1] \\
 [- (2^{n-1} - 1), (2^{n-1} - 1)]
 \end{array}$$

4. Obtén la codificación completa de la representación en complemento a 2 con 4 bits.

0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	1	2	3	4	5	6	7	-8	-7	-6	-5	-4	-3	-2	-1

5. Obtén la representación en complemento a dos con 8 bits de los siguientes números: +64, -64, -111 y -129

$$+64 \rightarrow W_{10} = V \Rightarrow 01000000$$

$$-64 \rightarrow W_{10} = 2^8 - |V| = 192 \rightarrow 11000000$$

$$-111 \rightarrow W_{10} = 2^8 - |V| = 145$$

$$W_2 \rightarrow 10010001$$

$$-129 \rightarrow W_{10} = 2^8 - |V|$$

$$\begin{array}{l}
 192 \xrightarrow{L2} 96 \xrightarrow{L2} 48 \xrightarrow{L2} 24 \xrightarrow{L2} 12 \xrightarrow{L2} 6 \xrightarrow{L2} 3 \xrightarrow{L2} 1 \\
 145 \xrightarrow{L2} 72 \xrightarrow{L2} 36 \xrightarrow{L2} 18 \xrightarrow{L2} 9 \xrightarrow{L2} 4 \xrightarrow{L2} 2 \xrightarrow{L2} 1
 \end{array}$$

Ca-2

$$V \geq 0 \rightarrow \text{S/M} \quad V = W_{10}$$

$$V < 0 \rightarrow 2^n - |V| = W_{10} \rightarrow W_2$$

$$[-2^{n-1}, 2^{n-1} - 1]$$

000
001
010
011
100
101
110
111

6. ¿Cuales son los rangos de representación en complemento a dos para 6, 10 y 16 bits?

$$[-2^{n-1}, 2^{n-1} - 1]$$

6 bits $[-32, 31]$
 10 bits $[-512, 511]$
 16 bits $[-32768, 32767]$

7. Obtén la representación en complemento a dos de 7 bits de los números siguientes: -64, +15, -45 y +64.

$$-64 \rightarrow W_{10} = 2^n - |V| = 2^7 - 64 = 64$$

$$W_2 = 1000000$$

$$+15 \rightarrow W_2 = 0001111$$

$$-45 \rightarrow W_{10} = 2^n - |V| = 2^7 - 45 = 83$$

$$W_2 = 1010011$$

$$+64 \text{ fuera de rango!}$$

$$2^7 - 1 = 63$$

$$\begin{array}{r} 64 \text{ L2} \\ \textcircled{0} \end{array} \quad \begin{array}{r} 32 \text{ L2} \\ \textcircled{0} \end{array} \quad \begin{array}{r} 16 \text{ L2} \\ \textcircled{0} \end{array} \quad \begin{array}{r} 8 \text{ L2} \\ \textcircled{0} \end{array} \quad \begin{array}{r} 4 \text{ L2} \\ \textcircled{0} \end{array} \quad \begin{array}{r} 2 \text{ L2} \\ \textcircled{0} \end{array} \quad \begin{array}{r} 1 \text{ L2} \\ \textcircled{0} \end{array}$$

$$\begin{array}{r} 15 \text{ L2} \\ \textcircled{1} \end{array} \quad \begin{array}{r} 7 \text{ L2} \\ \textcircled{1} \end{array} \quad \begin{array}{r} 3 \text{ L2} \\ \textcircled{1} \end{array} \quad \begin{array}{r} 1 \text{ L2} \\ \textcircled{1} \end{array}$$

$$\begin{array}{r} 83 \text{ L2} \\ \textcircled{1} \end{array} \quad \begin{array}{r} 41 \text{ L2} \\ \textcircled{1} \end{array} \quad \begin{array}{r} 20 \text{ L2} \\ \textcircled{0} \end{array} \quad \begin{array}{r} 10 \text{ L2} \\ \textcircled{0} \end{array} \quad \begin{array}{r} 5 \text{ L2} \\ \textcircled{1} \end{array} \quad \begin{array}{r} 2 \text{ L2} \\ \textcircled{0} \end{array} \quad \begin{array}{r} 1 \text{ L2} \\ \textcircled{1} \end{array}$$

8. ¿Qué números decimales representan los siguientes valores binarios codificados en complemento a dos con 8 bits: 10110110, 01011110, 11011100 y 10000001

neg. $\boxed{1}0110110 \rightarrow W_{10} = 182 = 2^n - |V|$

128 64 32 16 8 4 2 1

$$|V| = 2^8 - 182 = 74$$

$$V = -74$$

$$01011110 \rightarrow W_{10} = 94 = V$$

neg. $\boxed{1}0111100 \rightarrow W_{10} = 220 = 2^n - |V|$

$$V = 220 - 2^8 = -36$$

neg. $\boxed{1}0000001 \rightarrow W_{10} = 129 = 2^n - |V|$

$$V = 129 - 2^8 = -127$$

9. Considerando la representación en complemento a dos de 8 bits, verifica que:

a) $C2(-57_{C2,8}) = 57_{C2,8}$ y $C2(57_{C2,8}) = -57_{C2,8}$

b) $C2(C2(83)) = 83$

c) $C2(C2(-78)) = -78$

dos
métodos

a) $-57_{C2,8} \rightarrow W_{10} = 2^8 - 57 = 199$
 $W_2 = 11000111$
 $\rightarrow C2(11000111) = 00111001$
 $V = 57$

$\rightarrow C2(199_{W_{10}}) = 2^8 - 199 = 57_{W_{10}} \rightarrow V = 57$

b)

$83_{C2,8} \rightarrow W_{10} = 83$

$C2(C2(83)) = 2^8 - (2^8 - 83) = 2^8 - 2^8 + 83 = 83$

c) $-78_{C2,8} \rightarrow W_{10} = 2^8 - 78 = 184$

$C2(C2(184)) = 184_{W_{10}} \rightarrow -78_{C2,8}$

199 $\xrightarrow{L2}$ 99 $\xrightarrow{L2}$ 49 $\xrightarrow{L2}$ 24 $\xrightarrow{L2}$ 12 $\xrightarrow{L2}$ 6 $\xrightarrow{L2}$ 3 $\xrightarrow{L2}$ 1
 ① ① ① ① ① ① ① ①

10. Obtén mediante los dos métodos estudiados el resultado de la transformación C2 aplicada a los siguientes números en complemento a dos de 8 bits: 10101111, 10110000 y 00111010. ¿Qué valores decimales representan los números originales y los números transformados?

$C2(10101111) = 01010000 + 1$
 $= 01010001$

$10101111 \rightarrow W_{10} = 175 \rightarrow V = 175 - 2^8 = -81$

$01010001 \rightarrow W_{10} = 81 \rightarrow V = 81$

$10110000 \rightarrow W_{10} = 176 \rightarrow V = 176 - 2^8 = -80$

$C2(10110000) = 01010000$

$01010000 \rightarrow W_{10} = 80$

11. Demuestra la validez del método de extensión de signo estudiado para los números negativos en complemento a 2.

$$\begin{aligned}
 & \text{8 bits} \quad 1 \dots \dots \Rightarrow W_{10} = 2^7 + \dots \\
 & \quad \quad \quad V_1 = 2^8 - (2^7 + \dots) \\
 & \text{9 bits} \quad 11 \dots \dots \Rightarrow W_{10} = 2^8 + 2^7 + \dots \\
 & \quad \quad \quad V_2 = 2^9 - (2^8 + 2^7 + \dots) \\
 & \quad \quad \quad = \underbrace{2^9 - 2^8} - (2^7 + \dots) \\
 & \quad \quad \quad V_2 = 2^8 - (2^7 + \dots) \\
 & \quad \quad \quad \boxed{V_1 = V_2}
 \end{aligned}$$

$$Z = (2^{n-1}) \text{ o } (2^{n-1} - 1)$$

$$\text{Rango } [-Z, 2^n - 1 - Z]$$

FCO

II. Problemas de representación de enteros y reales

12. Obtén la codificación en exceso 16 para los valores en el intervalo $[-8, +4]$

Exceso 16 $16 = 2^4$ $n = 5$

-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4
01000	01001	01010	01011	01100	01101	01110	01111	10000	10001	10010	10011	10100

13. ¿Cuál es el rango de representación para los siguientes formatos de números enteros?

a) Exceso 127

b) Exceso 128

c) Exceso 115

$$[-Z, 2^n - 1 - Z]$$

a)

$$[-127, 128]$$

b)

$$[-128, 127]$$

c)

$$[-115, 140]$$

14. Codifica los siguientes números en exceso 127: -126, 0, 15 y -128

$$V = -126 \quad W_{10} = V + Z = 1 \rightarrow W_2 = 00000001$$

$$V = 0 \quad W_{10} = V + Z = 127 \rightarrow W_2 = 01111111$$

$$V = 15 \quad W_{10} = V + Z = 142 \rightarrow W_2 = 10001110$$

$$V = -128 \quad W_{10} = V + Z = -1$$

fuera de intervalo !!

$$\begin{array}{r} 142 \text{ L2} \\ 0 \text{ 71} \text{ L2} \\ 0 \text{ 35} \text{ L2} \\ 0 \text{ 17} \text{ L2} \\ 0 \text{ 8} \text{ L2} \\ 0 \text{ 4} \text{ L2} \\ 0 \text{ 2} \text{ L2} \\ 0 \end{array}$$

15. Decodifica (o sea obtén su representación decimal) los siguientes números representados en exceso 127: 01011000, 10011110, 00001100 y 00000001

128 64 32 16 8 4 2 1

$$01011000 \rightarrow W_{10} = 88$$

$$V = 88 - Z = \underline{-39}$$

$$10011110 \rightarrow W_{10} = 158$$

$$V = 158 - Z = \underline{31}$$

$$00001100 \rightarrow W_{10} = 12$$

$$V = 12 - Z = \underline{-115}$$

$$00000001 \rightarrow W_{10} = 1$$

$$V = 1 - Z = \underline{-126}$$

16. Obtén la representación binaria de los siguientes números decimales:

a) 46.125

b) 123.14

c) 38.235

a)

46.125

$0.125 \times 2 = 0.25$

$0.25 \times 2 = 0.5$

$0.5 \times 2 = 1$

46.125
 $\begin{array}{r} 46 \text{ L2} \\ \textcircled{0} 23 \text{ L2} \\ \textcircled{0} 11 \text{ L2} \\ \textcircled{0} 5 \text{ L2} \\ \textcircled{0} 2 \text{ L2} \\ \textcircled{0} 1 \end{array}$

1011101001

b)

123.14

$0.14 \times 2 = 0.28$

$0.28 \times 2 = 0.56$

$0.56 \times 2 = 1.12$

$0.12 \times 2 = 0.24$

$0.24 \times 2 = 0.48$

$0.48 \times 2 = 0.96$

$0.96 \times 2 = 1.92$

$0.92 \times 2 = 1.84$

c) $0.84 \times 2 = 1.68$

$0.68 \times 2 = 1.36$

$0.36 \times 2 = 0.72$

$0.72 \times 2 = 1.44$

$0.44 \times 2 = 0.88$

$0.88 \times 2 = 1.76$

$0.76 \times 2 = 1.52$

$0.52 \times 2 = 1.04$

$0.04 \times 2 = 0.08$

$0.08 \times 2 = 0.16$

$0.16 \times 2 = 0.32$

$0.32 \times 2 = 0.64$

$0.64 \times 2 = 1.28$

0.28×2

123.14
 $\begin{array}{r} 123 \text{ L2} \\ \textcircled{0} 61 \text{ L2} \\ \textcircled{0} 30 \text{ L2} \\ \textcircled{0} 15 \text{ L2} \\ \textcircled{0} 7 \text{ L2} \\ \textcircled{0} 3 \text{ L2} \\ \textcircled{0} 1 \end{array}$

123.14

11110111001000111101011100001

precisión simple

S	E	M
1	8	23

FCO

II. Problemas de representación de enteros y reales

17. Obtén la mantisa binaria y el exponente referido a la base 2 de los siguientes números representados en el formato IEEE-754 de precisión simple:

- a) 0 11011100 010100000000000000000000
- b) 1 10000000 100000000000000000000000
- c) 0 00000001 000000000000000000000000
- d) 0 10000001 110000000000000000000000

a)

$$S = 0 \rightarrow \text{positivo}$$

$$E = 11011100_{E_{127}} \rightarrow W_{10} = 220 \quad V = 220 - Z = 93$$

$$1'0101 \cdot 2^{93}$$

b)

$$S = 1 \rightarrow \text{negativo}$$

$$E = 10000000_{E_{127}} \rightarrow W_{10} = 128 \quad V = 128 - Z = 1$$

$$-1'1 \cdot 2^1$$

c)

$$S = 0 \rightarrow \text{negativo}$$

$$E = 00000001_{E_{127}} \rightarrow W_{10} = 1 \quad V = 1 - Z = -126$$

$$-1 \cdot 2^{-126}$$

el mínimo,
pues
-127
está
reservado
para
no normaliz.

d)

$$S = 0 \rightarrow \text{positivo}$$

$$E = 10000001 \rightarrow W_{10} = 129 \quad V = 129 - Z = 2$$

$$1'11 \cdot 2^2$$

18. Obtén la representación decimal de los números obtenidos en el ejercicio anterior.

$$1'0101 \cdot 2^{93} = (2^0 + 2^{-2} + 2^{-4}) \cdot 2^{93} = 1'29984 \cdot 10^{28}$$

$$-1'1 \cdot 2^1 = -(2^0 + 2^{-1}) \cdot 2^1 = -3$$

$$-1 \cdot 2^{-126} = -(2^0) \cdot 2^{-126} = -1'17549 \dots 10^{-38}$$

19. ¿Cuál es la representación de los siguientes números en el formato de coma flotante IEEE-754 de precisión simple?

- $+1.01101_2 \cdot 2^6$
- $+0.001001_2 \cdot 2^{-124}$
- $-0.001001_2 \cdot 2^{-3}$
- $-11111.0_2 \cdot 2^{-124}$

a)

$$S = 0$$

$$E = \quad \quad \quad W_{10} = Z + V = 127 + 6 = 133$$

$$E = 10000101_{E127}$$

$$M = 011010 \dots 0$$

$$\begin{array}{r} 133 \text{ L2} \\ \textcircled{1} \text{ 66} \text{ L2} \\ \textcircled{0} \text{ 33} \text{ L2} \\ \textcircled{1} \text{ 16} \text{ L2} \\ \textcircled{0} \text{ 8} \text{ L2} \\ \textcircled{0} \text{ 4} \text{ L2} \\ \textcircled{0} \text{ 2} \text{ L2} \\ \textcircled{1} \end{array}$$

b)

$$S = 0$$

$$E = \quad \quad \quad W_{10} = Z + V = 3$$

$$E = 00000011_{E127}$$

c)

$$M = 0010010 \dots 0$$

$$S = -1$$

$$E = \quad \quad \quad W_{10} = Z + V = 124$$

$$E = 01111100$$

$$\begin{array}{r} 124 \text{ L2} \\ \textcircled{0} \text{ 62} \text{ L2} \\ \textcircled{0} \text{ 31} \text{ L2} \\ \textcircled{0} \text{ 15} \text{ L2} \\ \textcircled{0} \text{ 7} \text{ L2} \\ \textcircled{0} \text{ 3} \text{ L2} \\ \textcircled{0} \end{array}$$

d)

$$M = 0010010 \dots 0$$

$$-11111'0 \cdot 2^{-124} \rightarrow \text{normalizar} \quad -1'11110 \cdot 2^{-120}$$

$$= 21441100'0 \cdot 2^{-126} \quad \text{NO NORMALIZADO}$$

$$S = 1$$

$$E = 00000000 \quad \text{no normalizado}$$

$$M = 0 \dots 01111100$$

20. ¿Cuales son los rangos de representación normalizado y no normalizado para el siguiente formato de coma flotante de 16 bits?

- El bits MSB es el bit de signo (0 para positivo y 1 para negativo)
- 5 bits para el exponente (b14..b10) codificado en exceso 15
- 10 bits para la mantisa (b9..b0) codificada con bit implícito

normalizado E: $[-Z+1, 2^n-1-Z-1]$
reservado a error y a no normalizado

$$E: [-14, 15]$$

$$\text{rango } \pm [1'000000000 \cdot 2^{-14}, 1'111111111 \cdot 2^{15}]$$

no normalizado:

$$[-0'111111111 \cdot 2^{-15}, 0'111111111 \cdot 2^{-15}]$$

21. ¿Cuales son los rangos de representación normalizado y no normalizado para los formatos IEEE-754 de precisión simple y de precisión doble?

simple

normalizado

$$E: [-Z+1, 2^n-1-Z-1]$$

$$[-126, 127]$$

$$\text{rango } \pm [1'00000 \dots 0 \cdot 2^{-126}, 1'111 \dots 1 \cdot 2^{127}]$$

23 unos 23 unos

no normalizado

$$\text{rango } [-0'11111111 \dots 1 \cdot 2^{-126}, 0'1111 \dots 1 \cdot 2^{-126}]$$

23 unos 23 unos

doble

normalizado

$$E: [-Z+1, 2^n-1-Z-1]$$

$$[-1022, 1023]$$

$$\text{rango } \pm [1'00 \dots 0 \cdot 2^{-1022}, 1'11 \dots 1 \cdot 2^{1023}]$$

52 unos 52 unos

$$z =$$

Exp 11 bits
 $z = 2^{n-1} - 1$
 $z = 1023$

no normalizado

$$\text{rango } [-0'111 \dots 11 \cdot 2^{-1022}, 0'111 \dots 1 \cdot 2^{-1022}]$$

52 unos 52 unos

The first part of the paper is devoted to the study of the properties of the function $f(x)$ defined by the equation $f(x) = \sum_{n=0}^{\infty} a_n x^n$. It is shown that $f(x)$ is analytic in the region $|x| < 1$ and that it satisfies the functional equation $f(x) = 1 - x f(x^2)$.

In the second part of the paper, we consider the problem of finding the coefficients a_n of the power series expansion of $f(x)$. It is shown that the coefficients a_n are given by the formula $a_n = \frac{1}{n!} \left(\frac{d}{dx} \right)^n f(0)$.

The third part of the paper is devoted to the study of the properties of the function $g(x)$ defined by the equation $g(x) = \sum_{n=0}^{\infty} b_n x^n$. It is shown that $g(x)$ is analytic in the region $|x| < 1$ and that it satisfies the functional equation $g(x) = 1 - x g(x^2)$.

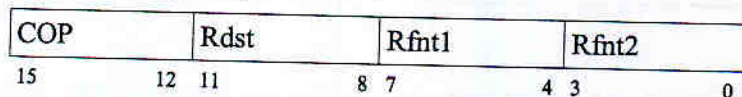
In the fourth part of the paper, we consider the problem of finding the coefficients b_n of the power series expansion of $g(x)$. It is shown that the coefficients b_n are given by the formula $b_n = \frac{1}{n!} \left(\frac{d}{dx} \right)^n g(0)$.

The fifth part of the paper is devoted to the study of the properties of the function $h(x)$ defined by the equation $h(x) = \sum_{n=0}^{\infty} c_n x^n$. It is shown that $h(x)$ is analytic in the region $|x| < 1$ and that it satisfies the functional equation $h(x) = 1 - x h(x^2)$.

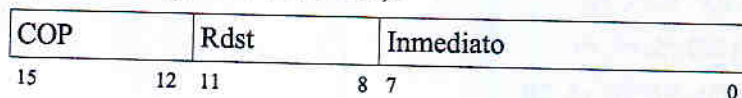
III. Problemas de codificación de las instrucciones

1. Un procesador que dispone de 16 registros en el banco de registros, utiliza tres formatos de instrucción de 16 bits para codificar las instrucciones:

Aritméticas (directo a registro):



Aritméticas (directo a registro e inmediato):



Salto y control:

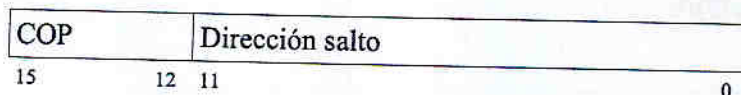


Tabla de códigos de operación:

Operación	COP
MOVE	0000
ADD	0101
COMPARE	0111
STOP	1111
JUMP	1000
JEQUAL	1001

0	0000	—
1	0001	—
2	0010	—
3	0011	—
4	0100	—
5	0101	—
6	0110	—
7	0111	—
8	1000	—
9	1001	—
A	1010	—
B	1011	—
C	1100	—
D	1101	—
E	1110	—
F	1111	—

Codifica en binario este programa:

MOVE r5, 0A	0000	0101	0000	1010
MOVE r15, 1B	0000	1111	0001	1011
ADD r1, r5, r15	0101	0001	0101	1111
JUMP 0BD	1000	0000	1011	1101
STOP	1111			

MOVE r3, 0A	
MOVE r15, 1B	
ADD r1, r5, r15	
JUMP 0BD0	
STOP	

2. Codifica este programa para el juego de instrucciones del procesador del ejercicio anterior.

```

MOVE r5, 10
ADD r5, r5, r5
COMPARE r5, 20
JEQUAL 0001
STOP

```

MOVE r5, 10	
ADD r5, r5, r5	
COMPARE r5, 20	
JEQUAL 0001	
STOP	

3. Realiza un programa que sume el contenido de las direcciones de memoria 0x0E y 0x0F, dejando el resultado en la dirección 0x0D.

```

MOVE RA, [0E]
ADD RA, [0F]
MOVE [0D], RA

```


4. Realiza un programa que calcule $9+5-2$, dejando el resultado en el registro RA.

```

MOVE RA, 9
ADDI RA, 5
SUBI RA, 2

```

5. Realiza un programa que sume el contenido de las direcciones de memoria 0x0E, 0x0F y 0x0D y lo incremente en 1, dejando el resultado en la dirección 0x0C

```

MOVE RA, [0E]
ADD RA, [0F]
ADD RA, [0D]
INC RA
MOVE [0C], RA

```

6. Realiza un programa que calcule la transformación C2 del contenido de memoria de la dirección 0x10 y deje el resultado en la dirección 0x11.

$$C2(W_2) = 2^n - W_2$$

10010110 → 01101010

```

  11111111
- 10010110
-----
  01101001

```

```

MOVE RA, FF
SUB RA, [10]
MOVE [11], RA

```

7. Realiza un programa que realice una suma con saturación con límite superior igual a 0x0F. En una suma con saturación, si el resultado es superior al límite, éste debe igualarse a dicho límite. Los dos operandos a sumar estarán en las direcciones 0x10 y 0x11 y el resultado deberá dejarse en la dirección 0x12.

```
MOVE RA, [10]
ADD RA, [11]
COMPAREI RA, 0F
JLESS dir
MOVE RA, 0F
dir MOVE [12], RA
```

8. Realiza un programa que invierta el orden de los contenidos de las direcciones 0x18, 0x19, 0x1A y 0x1B utilizando la pila como almacenamiento temporal.

```
MOVE RA [18]
PUSH RA
MOVE RA [19]
PUSH RA
MOVE RA [1A]
PUSH RA
MOVE RA [1B]
MOVE [18], RA
POP RA
MOVE [19], RA
POP RA
MOVE [1A], RA
POP RA
MOVE [1B], RA
```

9. Realiza una subrutina que multiplique el contenido de RA por dos. Realiza el programa principal que invoque dicha subrutina para RA=0x10.

```

sub { MOVE MOVE [aux], RA
      ADD RA, [aux]
      RET

```

```

MOVE RA, [10]
CALL sub

```

10. Realiza una subrutina que calcule 2^{RA} . Realiza un programa que invoque dicha subrutina.

```

MOVE [cont], RA
MOVE RA, 1
MOVE [aux], RA
ADD RA, [aux]
COMPAREI
bucle MOVE [aux], RA
      MOVE RA, [cont]
      COMPAREI RA, 0
      JEQUAL [fin]
      DEC RA
      MOVE [cont], RA
      MOVE RA, [aux]
      ADD RA, [aux]
      JUMP [bucle]
[fin] MOVE [aux], RA
      RET

```


11. Realiza un programa que cuente desde 00 hasta FF y muestre dicha cuenta en el display.

```

        MOVE RA, 00
bucle  OUT 1
        INC RA
        COMPAREI RA, FF
        JEQUAL [fin]
        JUMP [bucle]
fin     OUT 1

```

12. Realiza una subrutina que lea un valor del teclado y lo ponga en el registro RA. El valor leído debe estar comprendido entre 00 y 0F. La lectura de teclado se repetirá mientras el valor leído no esté dentro de dicho rango. El programa principal deberá llamar a dicha subrutina y visualizar el valor leído en el display.

```

bucle  IN 0
        COMPAREI RA, 0F
        JLESS [fin]
        JGREATER [bucle]
        RET

```

```

CALL bucle
OUT 1

```

13. Realiza un programa que lea dos valores del teclado y los sume, generando un resultado de 16 bits. Visualizar dicho resultado en el display mostrando primero el byte de mayor peso y luego el de menor peso. Supón que los números son enteros sin signo.

```

IN 0
MOVE [v1], RA
IN 0
ADD RA, [v1]
MOVE [byte2], RA
JLESS [dir]
MOVE RA, 00
: MOVE [byte1], RA
- - JUMP [dir2]
dir MOVE RA, 01
dir 2 OUT 1 -
      MOVE RA, [byte2]
      OUT 1

```

si al sumar desborda,
C = 1 y el byte de
mayor peso deberá
ser 0000001

JGREATER salta si C = 0
y Z = 0

JLESS salta si C = 1

14. Realiza un programa que lea dos valores del teclado y los multiplique, generando un resultado de 16 bits. Visualiza dicho resultado en el display mostrando primero el byte de mayor peso y luego el de menor peso. Supón que los números son enteros sin signo.

$[20] \times [21] \rightarrow [30][31]$

```

bucle : MOVE RA, [20]
        MOVE [aux], RA
        MOVE RA, [20]
        COMPAREI RA, 0
        JEQUAL [fin]
        MOVE [aux], RA
        MOVE RA, [20]
        DEC RA
        MOVE [20], RA
        MOVE RA, (AUX)
        ADD RA, [21]
        MOVE (AUX), RA
        JLESS [acarreo]
        JUMP [bucle]
acarreo MOVE RA, [30]
        INC RA
        MOVE [30], RA
        JUMP [bucle]
[fin]  MOVE RA, (aux)
        MOVE [31], RA

```

15. Realiza un programa en el que el usuario tenga que adivinar un número. El programa asignará una constante a dicho número y pedirá al usuario que introduzca un número por el teclado. Si el número introducido es el correcto se visualizará en el display de siete segmentos el número 0xC0 (que recuerda a la sílaba CO de COorrecto). Si el número adivinar es mayor se visualizará 0x01, y si el número es menor 0xFF. El programa seguirá pidiendo números al usuario hasta que éste acierte.

```

bucle: IN 0
        COMPAREI RA, 11 ← de
        JEQUAL [fin]
        JGREATER [mayor]
        MOVE RA, FF
        OUT 1
        JUMP [bucle]
mayor:  MOVE RA, 01
        OUT 1
        JUMP [bucle]
fin     MOVE RA, C0
        OUT 1
  
```


16. Realiza un programa que permita jugar al Nim. El Nim es un juego que consiste en que cada jugador retire 1, 2 o 3 palillos de una fila. El jugador que quita el último palillo en juego es el que pierde. Considerando una versión sencilla del juego en el que sólo haya una fila, implementad un programa que muestre en pantalla el número de palillos que queda en la fila. El programa pedirá al usuario que introduzca un 1, un 2 o un 3 y dicho valor se decrementará de la fila. (El programa no llevará control de los turnos ni del número de jugadores). El programa terminará cuando quede un palillo, puesto que el jugador que tenga el turno en ese momento es el que pierde.

introducir los
palillos en la
fila

} IN 0
MOVE [palillos], RA

bucle: OUT 1...
COMPAREI RA, 01
JEQUAL [fin]

pedir: IN 0
COMPAREI, RA, 3
JGREATER [pedir]
MOVE [dec], RA
MOVE RA, [palillos]
SUB RA, [dec]
MOVE [palillos], RA
JUMP [bucle]

fin

STOP

17. Realiza un programa que utilice el display de siete segmentos como un marcador deportivo. El dígito de la izquierda será el marcador local (de 0 a F) y el de la derecha el visitante (de 0 a F). El programa deberá tener dos subrutinas. La primera incrementará el marcador local y la segunda el visitante. El programa principal esperará a que se introduzca por teclado un 1 (local) o un 2 (visitante), indicando el equipo que marcó el último tanto. Una vez introducido dicho valor el programa llamará a la subrutina correspondiente.

```

gol:  IN 0
      MOVE RA, 00
      MOVE [res], RA
      COMPARE I RA, 1
      JEQUAL [local], 1
      COMPARE I RA, 2
      JEQUAL [visit], 2
      JUMP [fin]
local: MOVE RA, [res]
      ADDI RA, 10
      MOVE [res], RA
      JUMP [fin]
visit: MOVE RA, [res]
      ADDI RA, 01
      MOVE [res], RA
fin:   OUT 1
      JUMP [gol]

```