

Circuitos electrónicos digitales

Tema 4

Circuitos aritméticos Universidad de Sevilla

Usted es libre de copiar, distribuir y comunicar públicamente la obra y de hacer obras derivadas siempre que se cite la fuente y se respeten las condiciones de la licencia "Atribución-Share alike de Creative Commons".
Texto completo de la licencia en:
<http://creativecommons.org/licenses/by-sa/3.0/es/>

Departamento de Tecnología Electrónica – Universidad de Sevilla

1

Índice

5.1 Aritmética binaria

5.2 Circuitos sumadores de 1 bit: semisumador y sumador completo

5.3 Sumador paralelo de n bits

5.4 Sumador/Restador

5.5 Unidad aritmético-lógica (ALU)

2

Suma de números sin signo. Ejemplos

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

Rango para n bits: desde 0 a 2^n-1

Ejemplo (n=4 bits):

peso	2^3	2^2	2^1	2^0
8	4	2	1	
7	0	1	1	1
3	0	0	1	1
9	1	0	0	1

$$\begin{array}{r} 7+3=10 \\ \begin{array}{r} 0111 \\ + 0011 \\ \hline 1010 \end{array} \end{array} \quad \begin{array}{r} 9+7=16 \\ \begin{array}{r} 1001 \\ + 0111 \\ \hline 10000 \end{array} \end{array}$$

"me llevo"

OJO, en binario:

$1+1=10$ ("cero y me llevo una")
 $1+1+1=11$ ("una y me llevo una")

Departamento de Tecnología Electrónica – Universidad de Sevilla

3

Multiplicación binaria. Ejemplos

Multiplicación binaria. La multiplicación binaria sigue las mismas reglas que la correspondiente a la base diez, salvo que la suma final se realiza en binario. Ejemplos:

$$\begin{array}{r} 1001 \leftarrow A \\ \times 101 \leftarrow B \\ \hline 1001 \\ 0000 \\ 1001 \\ \hline 101101 \leftarrow P \end{array}$$

$$\begin{array}{r} 1101 \leftarrow A \\ \times 1011 \leftarrow B \\ \hline 1101 \\ 1101 \\ 0000 \\ 1101 \\ \hline 10001111 \leftarrow P \end{array}$$

4

División binaria. Ejemplos

División binaria. Ejemplos:

$$\begin{array}{r} 101111 \overline{)1000-} \\ \underline{1000-} \\ 1111 \\ \underline{1000-} \\ 1111 \\ \underline{1000-} \\ 0111 \end{array}$$

$$\begin{array}{r} 100101 \overline{)1110} \\ \underline{1000-} \\ 0110 \\ \underline{0110-} \\ 01 \end{array}$$

Departamento de Tecnología Electrónica – Universidad de Sevilla

5

Suma de números con signo. Notación signo-magnitud

Rango para (n+1) bits: desde $-(2^n-1)$ hasta (2^n-1)

Rango para 5 bits: desde -15 al +15

Ejemplo (n=5 bits): $7+(-3)=+4$

peso	2^4	2^3	2^2	2^1	2^0
8	4	2	1		
7	0	0	1	1	1
-3	1	0	0	1	1
-10	1	1	0	1	0

$$\begin{array}{r} 111 \\ + 00111 \\ \hline 10011 \\ 11010 \end{array} = -10$$

¡¡resultado absurdo!!

La notación signo-magnitud NO sirve para realizar sumas de números con signo

Departamento de Tecnología Electrónica – Universidad de Sevilla

6

Suma de números con signo en Notación Ca1

Rango para (n+1) bits: desde $-(2^n-1)$ hasta 2^n-1 : [-15,+15]

Ejemplo (n=5 bits)

n= 5 bits: desde -15 a +15

7	0	0	1	1	1
+3	0	0	0	1	1
-3	1	1	1	0	0
4	0	0	1	0	0

$$\begin{array}{r}
 7 + (-3) = +4 \\
 \begin{array}{r}
 1\ 1 \\
 0\ 0\ 1\ 1\ 1 \\
 +\ 1\ 1\ 1\ 0\ 0 \\
 \hline
 (1)\ 0\ 0\ 0\ 1\ 1 \\
 \hline
 0\ 0\ 1\ 0\ 0
 \end{array}
 \end{array}$$

ii +3!!
+4
ii incorrecto!!

En notación Ca1, al sumar, si se genera carry, hay que sumárselo al resultado



Departamento de Tecnología Electrónica – Universidad de Sevilla

7

Overflow ("no cabe") en Notación Ca1

Rango para n+1 bits: desde $-(2^n-1)$ hasta 2^n-1

Ejemplo (n=5 bits):

7	0	0	1	1	1
-7	1	1	0	0	0
9	0	1	0	0	1
-9	1	0	1	1	0

$$\begin{array}{r}
 -7 + (-9) = -16 \\
 \begin{array}{r}
 1 \\
 1\ 1\ 0\ 0\ 0 \\
 +\ 1\ 0\ 1\ 1\ 0 \\
 \hline
 (1)\ 0\ 1\ 1\ 1\ 0 \\
 \hline
 0\ 1\ 1\ 1\ 1
 \end{array}
 \end{array}$$

ii +14!!
+15
ii MAL!!

Tenemos que saber identificar cuándo el resultado "no cabe" (overflow) y se necesita un bit más



Departamento de Tecnología Electrónica – Universidad de Sevilla

8

Overflow ("no cabe") en Notación Ca1 (ii)

Rango para n= 5 bits: (desde -15 hasta +15)

-7+(-9)= -16

$$\begin{array}{r}
 1 \\
 1\ 1\ 0\ 0\ 0 \\
 +\ 1\ 0\ 1\ 1\ 0 \\
 \hline
 (1)\ 0\ 1\ 1\ 1\ 0 \\
 \hline
 0\ 1\ 1\ 1\ 1
 \end{array}$$

ii el resultado no cabe en 5 bits!!
ii necesito un bit más!!

-7+(-9)= -16

$$\begin{array}{r}
 1 \\
 1\ 1\ 1\ 0\ 0\ 0 \\
 +\ 1\ 1\ 0\ 1\ 1\ 0 \\
 \hline
 (1)\ 1\ 0\ 1\ 1\ 1\ 0 \\
 \hline
 1\ 0\ 1\ 1\ 1\ 1
 \end{array}$$

-16
ii correcto!!

Hay overflow si:

- Sumo dos números positivos y el resultado es negativo
- Sumo dos números negativos y el resultado da positivo

El resultado "no cabe" y necesito un bit más (en el ejemplo 6 bits)



Departamento de Tecnología Electrónica – Universidad de Sevilla

9

Overflow ("no cabe") en Notación Ca1 (iii)

Rango para n= 5 bits: (desde -15 hasta +15)

7+9= 16

$$\begin{array}{r}
 1\ 1\ 1\ 1\ 1 \\
 0\ 0\ 1\ 1\ 1 \\
 +\ 0\ 1\ 0\ 0\ 1 \\
 \hline
 1\ 0\ 0\ 0\ 0
 \end{array}$$

overflow

7+9= 16

$$\begin{array}{r}
 1\ 1\ 1\ 1\ 1 \\
 0\ 0\ 0\ 1\ 1\ 1 \\
 +\ 0\ 0\ 1\ 0\ 0\ 1 \\
 \hline
 0\ 1\ 0\ 0\ 0\ 0
 \end{array}$$

+16
ii correcto!!

En Notación CA1 hay overflow si:

- Sumo dos números positivos y el resultado es negativo
- Sumo dos números negativos y el resultado da positivo

El resultado "no cabe" y necesito un bit más (en el ejemplo 6 bits)



Departamento de Tecnología Electrónica – Universidad de Sevilla

10

Suma de números con signo en Notación Ca2

Rango para (n+1) bits: desde -2^n hasta 2^n-1

Ejemplo (n=5 bits)

n= 5 bits: desde -16 a +15

+7	0	0	1	1	1
+3	0	0	0	1	1
-3	1	1	1	0	1
4	0	0	1	0	0

$$\begin{array}{r}
 7 + (-3) = +4 \\
 \begin{array}{r}
 1\ 1\ 1\ 1\ 1 \\
 0\ 0\ 1\ 1\ 1 \\
 +\ 1\ 1\ 1\ 0\ 1 \\
 \hline
 \cancel{0}\ 0\ 1\ 0\ 0
 \end{array}
 \end{array}$$

+4
ii incorrecto!!

En notación Ca2, al sumar, si se genera carry, se desprecia

Recordad que: $Ca2(N) = Ca1(N) + 1$



Departamento de Tecnología Electrónica – Universidad de Sevilla

11

Overflow ("no cabe") en Notación Ca2

Rango para (n+1) bits: desde -2^n hasta 2^n-1

Ejemplo (n=5 bits):

-8+(-9)= -17

8	0	1	0	0	0
-8	1	1	0	0	0
9	0	1	0	0	1
-9	1	0	1	1	1

$$\begin{array}{r}
 1 \\
 1\ 1\ 0\ 0\ 0 \\
 +\ 1\ 0\ 1\ 1\ 1 \\
 \hline
 (\cancel{1})\ 0\ 1\ 1\ 1\ 1
 \end{array}$$

+15
ii MAL!!

En Notación Ca2 (como en Ca1) tenemos que saber identificar cuándo el resultado "no cabe" (overflow) y se necesita un bit más



Departamento de Tecnología Electrónica – Universidad de Sevilla

12

Overflow ("no cabe") en Notación Ca2 (ii)

Rango para $n = 5$ bits: (desde -16 hasta +15)

$$\begin{array}{r} -8+(-9) = -17 \\ \begin{array}{r} 1\ 1\ 0\ 0\ 0 \\ +\ 1\ 0\ 1\ 1\ 1 \\ \hline 1\ 0\ 1\ 1\ 1 \end{array} \quad \text{ii necesito un bit más!!} \end{array}$$

Hay overflow si:

- Sumo dos números positivos y el resultado es negativo
- Sumo dos números negativos y el resultado da positivo

El resultado "no cabe" y necesito un bit más (en el ejemplo 6 bits)



Departamento de Tecnología Electrónica – Universidad de Sevilla

13

Overflow ("no cabe") en Notación Ca2 (iii)

Rango para $n = 5$ bits: (desde -16 hasta +15)

$$\begin{array}{r} 8+9 = 17 \\ \begin{array}{r} 0\ 1\ 0\ 0\ 0 \\ +\ 0\ 1\ 0\ 0\ 1 \\ \hline 1\ 0\ 0\ 0\ 1 \end{array} \quad \text{ii MAL!!} \end{array}$$

En Notación Ca2 hay overflow si:

- Sumo dos números positivos y el resultado es negativo
- Sumo dos números negativos y el resultado da positivo

El resultado "no cabe" y necesito un bit más (en el ejemplo 6 bits)



Departamento de Tecnología Electrónica – Universidad de Sevilla

14

Índice

5.1 Aritmética binaria

5.2 Circuitos sumadores de 1 bit: semisumador y sumador completo

5.3 Sumador paralelo de n bits

5.4 Sumador/Restador

5.5 Unidad aritmético-lógica (ALU)



Departamento de Tecnología Electrónica – Universidad de Sevilla

15

5.2. Sumadores de 1 bit

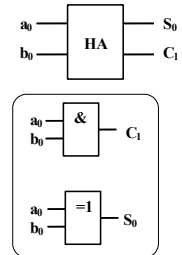
Semisumador o Half Adder (HA):

Suma dos números de 1 bit, generando el bit de suma y el carry

a_0	b_0	C_1	S_0
0	0	0	0
1	0	0	1
0	1	0	1
1	1	1	0

$$C_1 = a_0 \cdot b_0$$

$$S_0 = a_0 \oplus b_0$$



Departamento de Tecnología Electrónica – Universidad de Sevilla

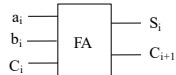
16

5.2. Sumadores de 1 bit (ii)

Si queremos imitar el algoritmo habitual de suma, necesitaremos un **Sumador completo Full Adder (FA)**

Full-adder: suma dos números de 1 bit y un bit de carry, generando el bit de suma y el nuevo carry

C_{out}	c_{n-1}	c_2	c_1	a_{n-1}	a_2	a_1	a_0	b_{n-1}	b_2	b_1	b_0	S_{n-1}	S_2	S_1	S_0
				0	0	0	0	0	0	0	0	0	0	0	0
				0	0	1	0	0	1	0	1	0	1	0	1
				0	1	0	0	0	1	0	1	0	1	0	1
				0	1	1	1	0	1	1	0	1	0	1	0
				1	0	0	0	0	1	0	1	1	0	1	0
				1	0	1	1	1	0	1	0	1	1	0	1
				1	1	1	0	1	1	0	1	1	1	0	1
				1	1	1	1	1	1	1	1	1	1	1	1



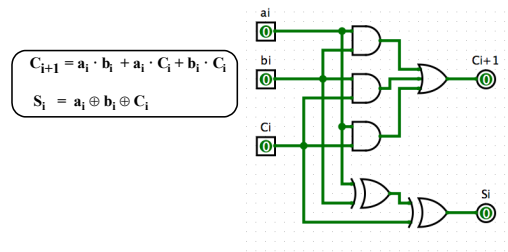
Departamento de Tecnología Electrónica – Universidad de Sevilla

17

5.2. Sumadores de 1 bit (iii)

Sumador completo Full Adder (FA)

Una implementación mediante puertas lógicas:



Departamento de Tecnología Electrónica – Universidad de Sevilla

18

Índice

- 5.1 Introducción
- 5.2 Circuitos sumadores de 1 bit: semisumador y sumador completo
- 5.3 Sumador paralelo de n bits
- 5.4 Sumador/Restador
- 5.5 Unidad aritmético-lógica (ALU)

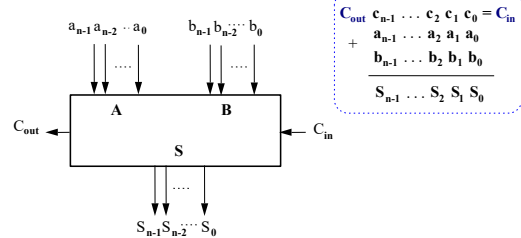


Departamento de Tecnología Electrónica – Universidad de Sevilla

19

5.3. Sumador paralelo de n bits

Realiza la suma de dos números binarios de n bits y un carry de entrada (C_{in}), generando el resultado de la suma (n bits) y un bit de carry de salida (C_{out}).

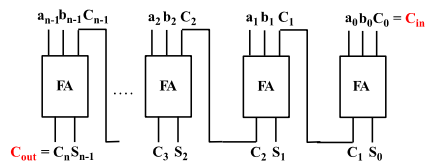


20

5.3. Sumador paralelo de n bits (ii)

Sumador paralelo con acarreo serie

- Se utilizan n Full-adders conectados en una cadena en serie, imitando el algoritmo de suma
- Se trata de un circuito modular

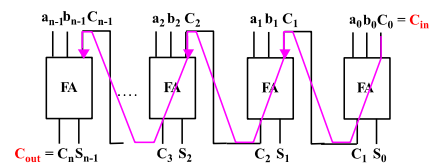


Departamento de Tecnología Electrónica – Universidad de Sevilla

21

5.3. Sumador paralelo de n bits (iii)

- Es lento debido a la **propagación** serie del **acarreo**
- El tiempo** que tarda en realizarse una suma **crece linealmente** con el número de bits

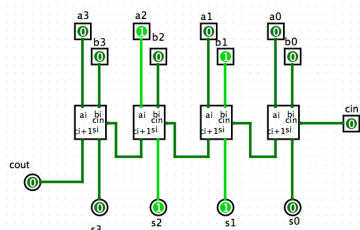


Departamento de Tecnología Electrónica – Universidad de Sevilla

22

5.3. Sumador paralelo de n bits (iv)

Ejemplo: Sumador completo de 4 bits



SUMADOR COMPLETO DE 4 BITS



Departamento de Tecnología Electrónica – Universidad de Sevilla

23

OVERFLOW (desbordamiento) en Notaciones Ca1 y Ca2

- Se pone de manifiesto porque el resultado no cabe en los n bits (necesito un bit más) y **el bit de signo no es correcto**:
 - al sumar dos números positivos se obtiene uno negativo
 - al sumar dos números negativos se obtiene uno positivo
- En caso de desbordamiento, el resultado correcto está en el número de $n+1$ bits **$A + B = C_{out}S_{n-1} \dots S_0$**
- La detección del desbordamiento se lleva a cabo mediante una señal adicional: **el bit de overflow (V)**

$$\begin{array}{r} C_n \quad C_{n-1} \\ 0 \quad 1 \\ 0 \quad 1 \\ + \quad 0 \quad a_{n-2} \dots a_0 \\ + \quad 0 \quad b_{n-2} \dots b_0 \\ \hline 1 \quad S_{n-2} \dots S_0 \end{array} \quad \begin{array}{r} C_n \quad C_{n-1} \\ 1 \quad 0 \\ 1 \quad 0 \\ + \quad 1 \quad a_{n-2} \dots a_0 \\ + \quad 1 \quad b_{n-2} \dots b_0 \\ \hline 0 \quad S_{n-2} \dots S_0 \end{array}$$

$$V = C_n \oplus C_{n-1} = C_{out} \oplus C_{n-1}$$



Departamento de Tecnología Electrónica – Universidad de Sevilla

24

Ejemplos de OVERFLOW en Ca2

- La suma de números con signo en Ca1 y Ca2, puede generar "OVERFLOW"
- Ejemplos (Ca2)

$\begin{array}{r} 1001 = -7 \\ 0101 = +5 \\ \hline 1110 = -2 \end{array}$	$\begin{array}{r} 1100 = -4 \\ 0100 = +4 \\ \hline 10000 = 0 \end{array}$	$\begin{array}{r} 0011 = +3 \\ 0100 = +4 \\ \hline 0111 = +7 \end{array}$
$\begin{array}{r} 1100 = -4 \\ 1111 = -1 \\ \hline 11011 = -5 \end{array}$	$\begin{array}{r} 0101 = +5 \\ 0100 = +4 \\ \hline 1001 = -7 \end{array}$	$\begin{array}{r} 1001 = -7 \\ 1010 = -6 \\ \hline 10011 = +3 \end{array}$

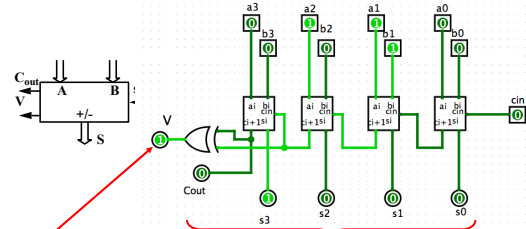
desbordamiento



Departamento de Tecnología Electrónica – Universidad de Sevilla

25

Sumador 4 bits con salidas CARRY y OVERFLOW



Si $V=1$, el resultado correcto se obtiene con 5 bits, incluyendo el bit Cout



Departamento de Tecnología Electrónica – Universidad de Sevilla

26

Índice

- 5.1 Introducción
- 5.2 Circuitos sumadores de 1 bit: semisumador y sumador completo
- 5.3 Sumador paralelo de n bits
- 5.4 Sumador/Restador de n bits
- 5.5 Unidad aritmético-lógica (ALU)



Departamento de Tecnología Electrónica – Universidad de Sevilla

27

5.4. Sumador/restador de números con signo de n bits

Una vez diseñado el sumador de n bits, aprovechamos la propiedad de las notaciones Ca1 y Ca2 para convertir restas en sumas: "restar es sumar en complemento"

Notación Ca1: $A - B = A + \text{Ca1}(B)$

Notación Ca2: $A - B = A + \text{Ca2}(B) = A + \text{Ca1}(B) + 1$

A, B y el resultado son números con signo



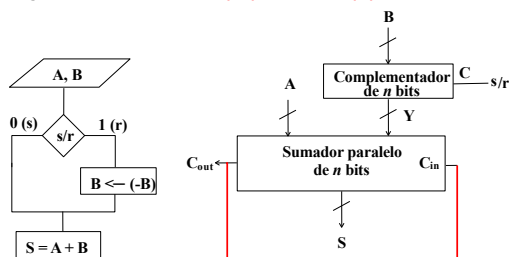
Departamento de Tecnología Electrónica – Universidad de Sevilla

28

Sumador/restador n bits (ii)

La suma-resta de números en Notación Ca1

- En general: $A - B = A + (-B) = A + \text{Ca1}(B)$



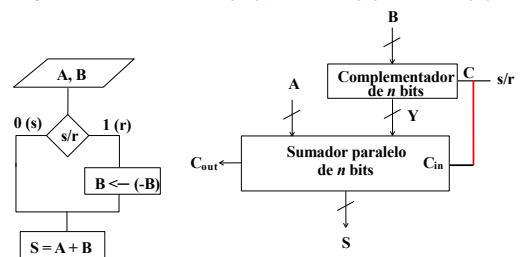
Departamento de Tecnología Electrónica – Universidad de Sevilla

29

Sumador/restador n bits (iii)

La suma-resta de números en complemento a 2

- En general: $A - B = A + (-B) = A + \text{Ca2}(B) = A + \text{Ca1}(B) + 1$



Departamento de Tecnología Electrónica – Universidad de Sevilla

30

Unidad aritmético-lógica (ALU)

- Es el circuito donde se realiza el procesamiento de datos
 - Procesado: operaciones aritméticas y lógicas. Normalmente se opera sobre dos datos
 - Usualmente pueden realizar diversas operaciones. Para elegir las se incluyen unas señales de selección
 - Además de las salidas que muestran el resultado de la operación, se incluyen otras salidas (*flags*) de estado o de condición.
 - Típicamente son C_{out} , V , Z ($Z=1$ si el resultado es 0) y S (signo)

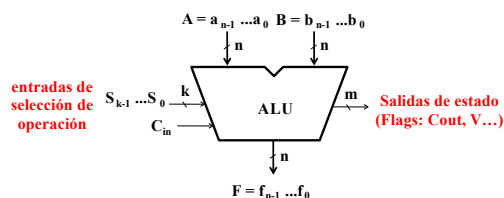


Departamento de Tecnología Electrónica – Universidad de Sevilla

37

Unidad aritmético-lógica (ALU)

- Representación gráfica de una ALU



Departamento de Tecnología Electrónica – Universidad de Sevilla

38

Unidad aritmético-lógica (ALU)

- Ejemplo** de diseño de una ALU

$S_2 S_1 S_0$	Función ALU	
	$C_{in} = 0$	$C_{in} = 1$
0 0 0	$F = A$	$F = A + 1$
0 0 1	$F = A + B$	$F = A + B + 1$
0 1 0	$F = A + /B$	$F = A + /B + 1 = A - B$
0 1 1	$F = A + 1111$	$F = A + 1111 + 1$
1 0 0	$F = A \text{ AND } B$	
1 0 1	$F = A \text{ OR } B$	
1 1 0	$F = \text{NOT } A$	
1 1 1	$F = A \text{ XOR } B$	

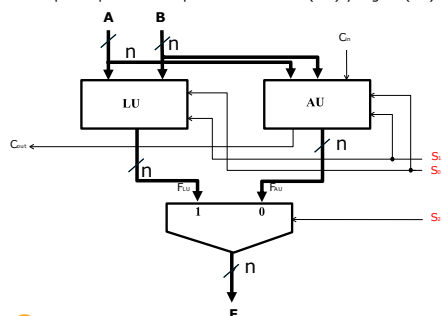


Departamento de Tecnología Electrónica – Universidad de Sevilla

39

Diagrama de bloques de una ALU

Diseñamos "por separado" las partes aritmética (AU) y lógica (LU).



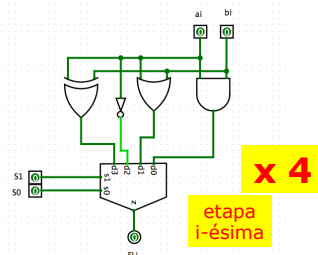
Departamento de Tecnología Electrónica – Universidad de Sevilla

40

Diseño del bloque lógico de la ALU del ejemplo (etapa típica)

El Bloque Lógico tiene un diseño modular, repitiendo la etapa típica tantas veces como bits tengan A y B

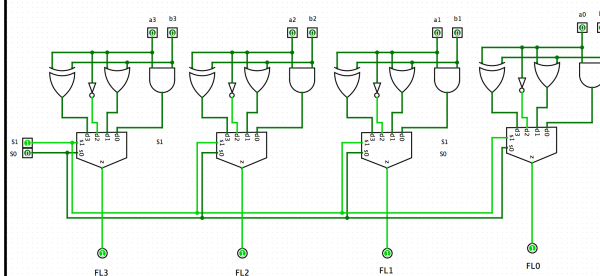
$S_2 S_1 S_0$	Función ALU
1 0 0	$F = A \text{ AND } B$
1 0 1	$F = A \text{ OR } B$
1 1 0	$F = \text{NOT } A$
1 1 1	$F = A \text{ XOR } B$



Departamento de Tecnología Electrónica – Universidad de Sevilla

41

Diseño del bloque lógico de la ALU del ejemplo (a nivel de bits)

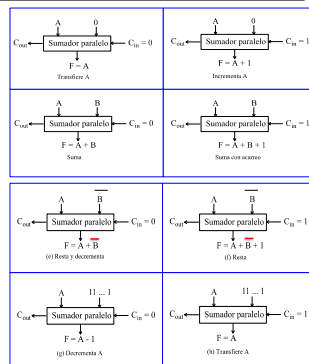


Departamento de Tecnología Electrónica – Universidad de Sevilla

42

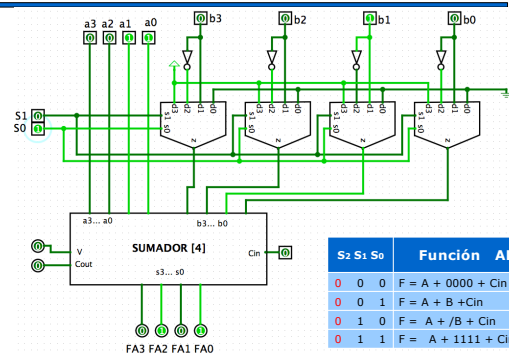
Diseño del Bloque Aritmético de la ALU del ejemplo

S ₂	S ₁	S ₀	Función ALU
0	0	0	$F = A + 0000 + Cin$
0	0	1	$F = A + B + Cin$
0	1	0	$F = A + B / Cin$
0	1	1	$F = A + 1111 + Cin$



43

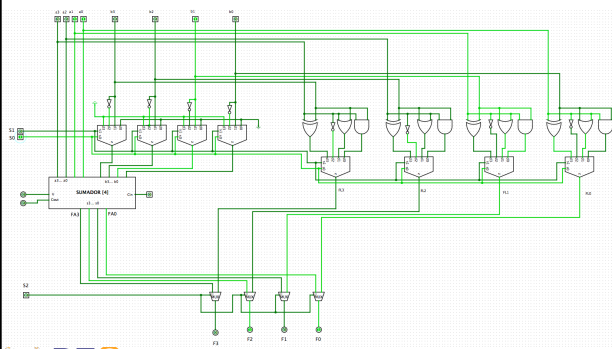
Diseño del Bloque Aritmético de la ALU



S ₂	S ₁	S ₀	Función ALU
0	0	0	$F = A + 0000 + Cin$
0	0	1	$F = A + B + Cin$
0	1	0	$F = A + B / Cin$
0	1	1	$F = A + 1111 + Cin$

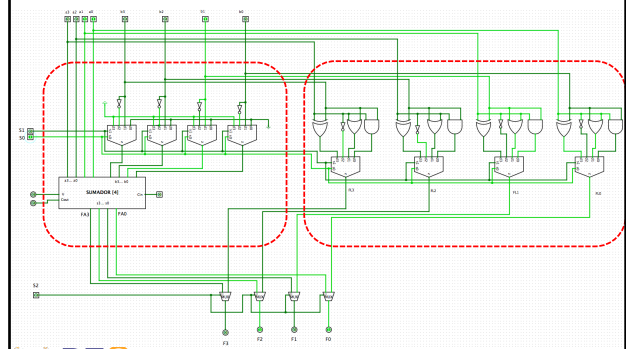
44

ALU [4] (diseño a nivel de bits)



45

ALU [4] (diseño a nivel de bits)



46