

# ETC-Tema-3.pdf



TurbApuntPatata



Estructura de Computadores



2º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingeniería Informática  
Universidad Politécnica de Valencia

**BBVA**

1/6  
Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

Ábrete la Cuenta Online de BBVA y llévate 1 año de **Wuolah PRO**

cómo??



Ventajas Cuenta Online de BBVA

0€  
Sin comisión de administración o mantenimiento de cuenta.  
(0 % TIN 0 % TAE)

0€  
Sin comisión por emisión y mantenimiento de Tarjeta  
Aqua débito.

0  
Sin necesidad de domiciliar nómina o recibos.

Las ventajas de **WUOLAH PRO**

✖ Di adiós a la publi en los apuntes y en la web

✖ Descarga carpetas completas de un tirón

✖ Acumula tickets para los sorteos

cómo???



1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

# ESTRUCTURA DE COMPUTADORES

## Tema 3

ventajas

PRO



Di adiós a la publi en los apuntes y en la web



Acumula tickets para los sorteos



Descarga carpetas completas

estudia sin publi  
**WUOLAH** PRO

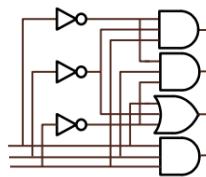
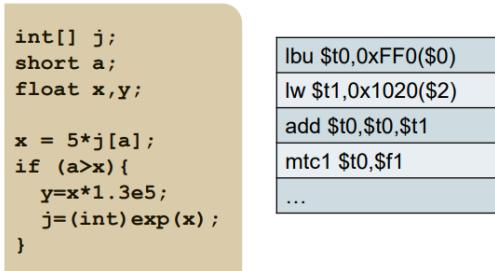


# TEMA 3

## Aritmética de Enteros

### Operaciones y operadores

*Los cálculos en los computadores de las operaciones lógicas y aritméticas expresadas en alto nivel se traducen en datos e instrucciones de código máquina y Durante el ciclo de instrucción, el procesador aplica operadores para procesar los datos*

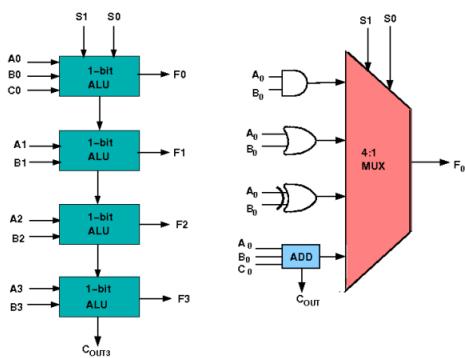
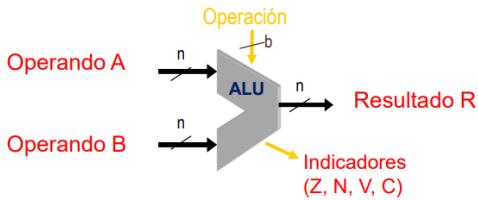


### La unidad aritmético-lógica (ALU)

Esta es la encargada de realizar todas las operaciones aritméticas y lógicas del procesador. Esta tiene, además de la salida de resultado R, una salida de indicadores (flags) que dan información del resultado

ALU (Arithmetic Logic Unit)

- **Z=1** si R=0
- **N=1** si R < 0
- **V=1** si desbordamiento en Ca2



Estructura típica de la ALU. Ejemplo con 4 operaciones y 4 bits

### Parámetros de una ALU

#### Funcionales: Operaciones que puede realizar

- **Operaciones lógicas** (AND, OR, NOT, XOR, ...) y desplazamientos
- **Cálculo elemental**: suma, resta, multiplicación y división
- **Comparación**: =, ≠, <, ≤, ≥, >
- **Conversión entre tipos**
- **Tipos de operandos que puede manejar**: CBN, entero, real, ...



1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

# Ábrete la Cuenta Online de BBVA y llévate 1 año de Wuolah PRO



cómo??

## Las ventajas de Wuolah PRO



Di adiós a la publi en  
los apuntes y en la web



Descarga carpetas  
completas de un tirón



Acumula tickets  
para los sorteos

## Ventajas Cuenta Online de BBVA

0€

Sin comisión de administración o  
mantenimiento de cuenta.  
(0 % TIN 0 % TAE)

0€

Sin comisión por emisión y  
mantenimiento de Tarjeta  
Aqua débito.

0

Sin necesidad de domiciliar  
nómina o recibos.

## Prestaciones (coste temporal): *Tiempo de respuesta (retardo)*

- **Tiempo necesario** para **realizar un cálculo**. Cuanto más corto, mejor
- Se mide en **unidades de tiempo**: ns, ps, retardo de puerta...

## Productividad

- **Número de operaciones** por **unidad de tiempo**: Cuanto más grande, mejor
- Se puede medir con **unidades genéricas OPS** (operaciones por segundo), KOPS, MOPS, GOPS... → caso de coma flotante: FLOPS, MFLOPS, etc

## Complejidad (coste espacial)

- Número de **puertas**, Número de **transistores**, **Área** de chip. **Unidades típicas**:  $\mu\text{m}^2$ ,  $\text{nm}^2$ , **Potencia**. **Unidades**:  $\mu\text{W}$ ,  $\text{mW}$

## Representación de Enteros

Con  $n$  bits sólo se pueden representar  $2n$  palabras diferentes.

**Conjunto N:** números naturales, se codifican en **código binario natural CBN**.

**Conjunto Z:** números enteros con signo, se codifican en **complemento a dos CA2**

### Rangos de representación con $n$ bits

- **Para N:** CBN Rango  $[0 \dots +2^{n-1}]$
- **Para Z:** codificación en complemento a 2:  $[-2^{n-1} \dots +2^{n-1}-1]$

$n$	Sin signo	Con signo
8	0 ... 255	-128 ... +127
16	0 ... 65.535	-32.768 ... +32.767
32	0 ... 4.294.967.295	-2.147.483.648 ... +2.147.483.647
64	0 ... $1.84 \cdot 10^{19}$ (aprox)	$-9.2 \cdot 10^{18} \dots +9.2 \cdot 10^{18}$ (aprox)

## Fundamentos

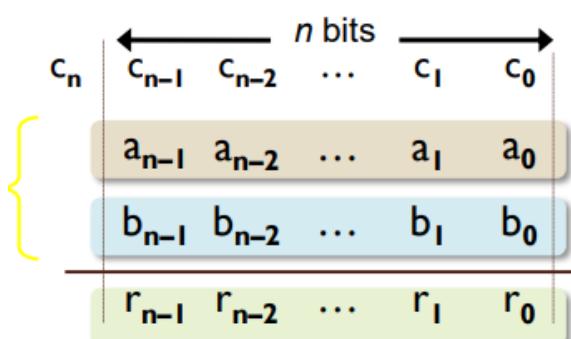
### Anatomía de la suma

El procedimiento general calcula  $R = A + B + c_0$ .  $R$  es el número  $A$  más el  $B$  más los acarreos

#### Acarreos

- $c_0$  (**de entrada**), habitualmente  $c_0 = 0$
- $c_{n-1} \dots c_1$ , forman parte del cálculo
- $c_n$  (**de salida**) útil a veces

← 4 bits →	$0 \quad 1 \quad 0 \quad 1$	CBN	5	Ca2
+	$1 \quad 0 \quad 0 \quad 1$	$+ 9$	$+ 5$	$+ -7$
		<hr/>	<hr/>	<hr/>
	$1 \quad 1 \quad 1 \quad 0$	$14$	$-2$	



## Desbordamiento de la suma

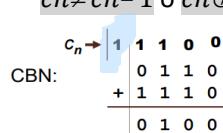
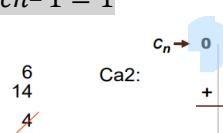
Su detección en CBN es con el **carri final  $C_n=1$**  y En aritmética en Ca2, si los **signos de los sumandos son iguales**. Dos maneras de detectarlo:

**Signo Invertido:** El signo del resultado generado está **invertido (por software)**:

$$an-1 = bn-1 \neq rn-1 \text{ o } Not(an-1 \oplus bn-1) \text{ and } (bn-1 \oplus rn-1) = 1$$

**Diferente Carri final:** Los dos carrys finales son **distintos** (por hardware) OverFlow:

**Ejemplo** con **n = 4 bits**

CBN: 	Ca2: 
---	--

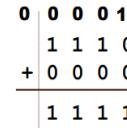
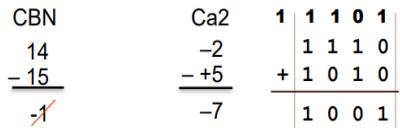
**Solo hay desbordamiento** cuando se suman cosas del mismo signo, 2 "+" o 2 "-".

**Si se hace una resta (+ y -) NO PEDE HABER DESBORDAMIENTO**

## Anatomía de la resta

Puede hacerse como una suma  $R = A + Ca2(B) = A + Ca1(B) + 1$   $A-B = A+(-B)$

**En CBN:** El desbordamiento se detecta cuando  $Cn = 0$

	Ca2: 
--	---

**En Ca2:** El desbordamiento se detecta cuando  $Cn \neq cn-1$

## Comparación de 2 enteros

### Comparación de dos enteros CBN

El **resultado** de una comparación es un **bit**:

(1 = cierto, 0 = falso)

La comparación  $A < B$  se puede hacer restando ( $R = A - B$ ) y analizando los acarreos y el signo de R. El valor concreto de R es irrelevante

Comparación	CBN
$A == B$	$R = 0$
$A >= B$	$c_n = 1$ (R es representable)
$A < B$	$c_n = 0$ (R no es representable)

### Comparación de dos enteros en Ca2

El **resultado** de una comparación es un **bit**:

(1 = cierto, 0 = falso)

La comparación  $A < B$  se puede hacer restando ( $R = A - B$ ) y viendo si hay desbordamiento, bit V y analizando el signo del resultado

Comparación	Ca2	
$A == B$	$R = 0$	$R = 0$
$(V = 0)$ (R es representable)	$R$ es positivo $A \geq B$	$R$ es negativo $A < B$
$(V = 1)$ (R no es representable)	$R$ es positivo $A < B$	$R$ es negativo $A > B$

## Instrucciones

operación	formato	con signo	sin signo
suma	R	ADD	ADDU
suma	I	ADDI	ADDIU
resta	R	SUB	SUBU
comparación	R	SLT	SLTU
comparación	I	SLTI	SLTIU

No mucho a tener en cuenta de esta parte. Solo es que **algunas instrucciones pueden desbordar** y hacer excepciones

cómo???



1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

ventajas

PRO



Di adiós a la publicidad en los apuntes y en la web



Acumula tickets para los sorteos



Descarga carpetas completas

estudia sin publicidad

**WUOLAH PRO**

## Operadores de Suma

### La suma en serie

*La suma en serie reproduce el procedimiento humano de suma. Se hace por orden, desde el LSB hacia el MSB.*

En cada **columna i**, hay que sumar los bits de los sumandos  $a_i$ ,  $b_i$  y el acarreo  $c_i$  que llega de la columna  $i-1$  para obtener el **bit del resultado  $s_i$**  y generar el acarreo  $c_{i+1}$  hacia la columna  $i+1$  siguiente.

**Acarreo de entrada  $c_0 = 0$**

### El sumador completo de un bit (Full adder)

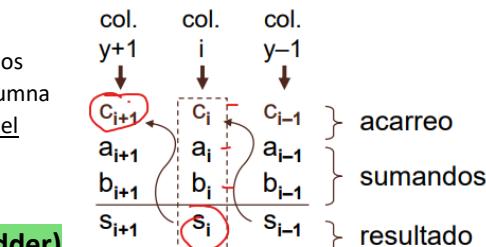
*Implementa los cálculos de una columna de la suma en serie. Este admite 3 entradas de un bit y produce dos salidas*

#### Tiempo de retardo

$$T_s = T_{not} + T_{and} + T_{or}$$

$$T_c = T_{and} + T_{or}$$

*El bicho este está compuesto por muchas puertas lógicas. El tiempo que tardan en salir la C y la S es lo de arriba*

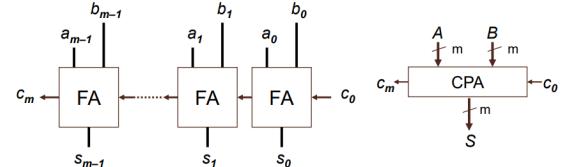


$a_i$	$b_i$	$c_i$	$c_{i+1}$	$s_i$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

### El sumador en serie de n bits: CPA (Carry Propagation Adder)

Par hacer un sumador de números de **m bits**, se conectan **m sumadores completos en cascada**

- La **salida de acarreo de un sumador** se **conecta a la entrada de acarreo** del sumador siguiente
- El operador resultante tiene una **entrada** y una **salida de acarreo global**



#### Tiempo:

$$CPA(n) = (n - 1) \cdot tc + \max\{tc, ts\}$$

Ejemplo  $CPA(10) = (10 - 1) \cdot 2 + \max\{2, 3\} = 21T$        $Tc = 2T$ ,     $Ts = 2T$

## Mejoras de la Suma

### Carry Lookahead Adder (CLA)

- Sumador con **anticipación de acarreo**
- **Calcula** los bits de acarreo  $c_i$  **antes** que los bits de **suma**  $s_i$
- El **retardo** del cálculo de los acarreos es  **$O(\log(n))$**

## CSA (Carry Select Adder): Acelera la operación de suma a base de invertir más circuitos

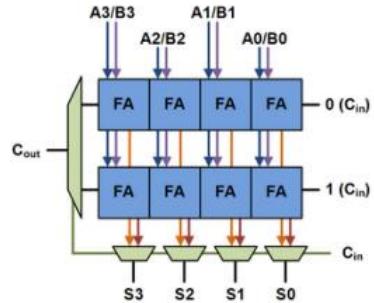
Divide la suma en dos partes y gestiona la parte alta con los posibles valores del acarreo proveniente de la parte baja: Pre-cálculo de la suma con  $Cin=0$  y  $Cin=1$

Elemento básico de constructor (4 bits) Pero puede no ser así. Ej:

- $n=16$  bits  $\rightarrow 4-4-4-4$ , caso ideal:  $\sqrt{n}$
- $n=16$  bits  $\rightarrow 5-4-3-2-2$

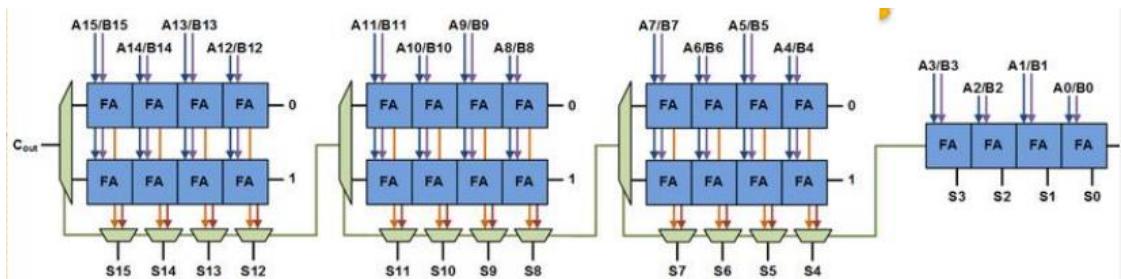
Se hacen las sumas con los 2 posibles acarreos de manera paralela, lo haces todo y ya luego usas los acarreos que correspondería

**Retardo:** Retardo  $\sim = T_{cpa} + (N - 1)T_{mux}$

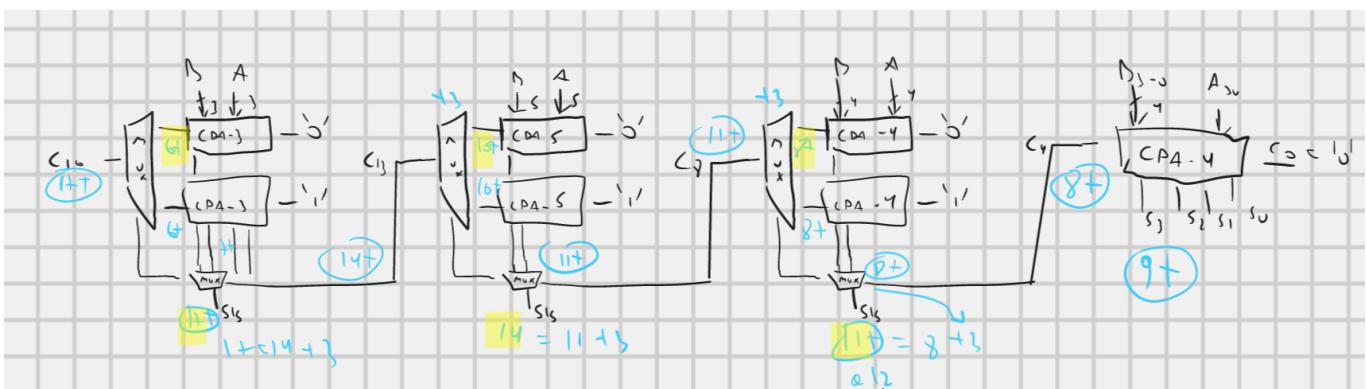


$N$  = número de bloques

## Sumador de 16 bits



Ejemplo de retador con un CSA(3,5,4,4)

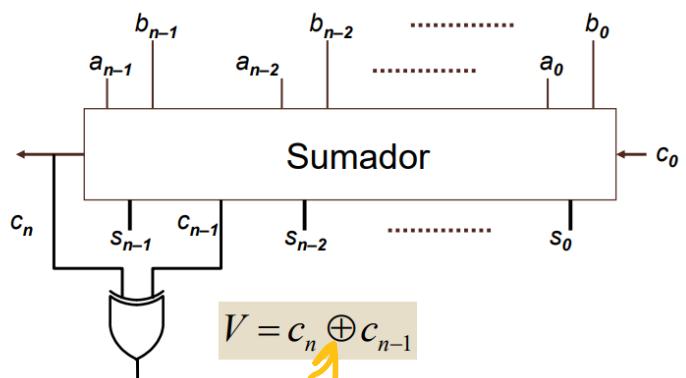


## Desbordamiento

- Se detecta cuando los bits de acarreo de orden  $n$  y  $n-1$  NO son iguales. Osea que los 2 últimos NO son =

El signo del resultado  $s_{n-1}$  es incorrecto

Se hace la operación Xor esa tope wapa

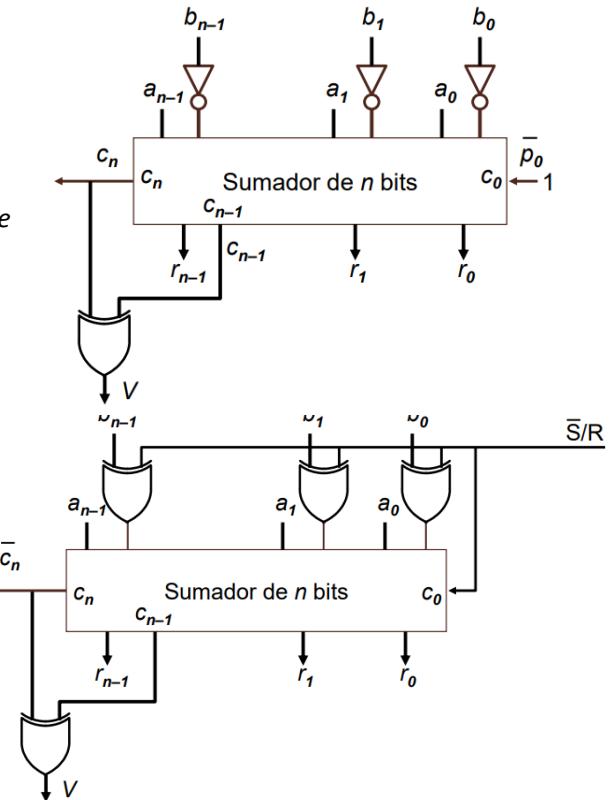


## Operadores de resta

Diseño del **restador a partir del sumador** donde:  
 $R = A - B = A + Ca2(B) = A + \text{not}(B) + 1$

Osea que se le **invierte cada entrada de bit** para hacerle el Ca2 de un número negativo. La detección de desbordamiento es igual que con la suma

- En CBN:  $c_n=0$
- En Ca2:  $p_n \neq p_{n-1} \rightarrow (c_n \text{ Xor } c_{n-1})=1$



## Operadores de Suma/Resta

Funciona igual que los otros 2 pero en las **entradas de bits** hay **puertas Xor** para decidir si invertir o no la señal de entrada

## Desplazamientos y aritmética entera

Desplazar **n bits hacia la izquierda** es equivalente a multiplicar por  **$2^n$  (binario)**

Entrar **n ceros por la derecha**, esta Operación es válida para enteros con y sin signo. Nombre de la operación: desplazamiento lógico hacia la izquierda

$$\begin{array}{r}
 0\ 0\ 0\ 1\ 1\ 0 \quad 6 \\
 \xrightarrow{\ll 2} \quad \xrightarrow{x 2^2} \\
 0\ 1\ 1\ 0\ 0\ 0 \quad 24
 \end{array}
 \quad
 \begin{array}{r}
 1\ 1\ 1\ 0\ 1\ 0 \quad -6 \\
 \text{Se pierden, pero esto oké} \\
 \text{xq es un número negativo} \\
 \xrightarrow{\ll 2} \quad \xrightarrow{x 2^2} \\
 1\ 0\ 1\ 0\ 0\ 0 \quad -24
 \end{array}$$

Desplazar **n bits hacia la derecha** es equivalente a dividir por  **$2^n$**

Entrar **n ceros o unos por la izquierda**, números positivos **0s**, negativos **1s**. Nombre de la operación: desplazamiento lógico hacia la derecha

$$\begin{array}{r}
 1\ 0\ 1\ 0\ 0\ 0 \quad 40 \\
 \xrightarrow{>> 2} \quad \xrightarrow{/ 2^2} \\
 0\ 0\ 1\ 0\ 1\ 0 \quad 10
 \end{array}
 \quad
 \begin{array}{r}
 1\ 0\ 1\ 0\ 0\ 0 \quad -24 \\
 \xrightarrow{>> 2} \quad \xrightarrow{/ 2^2} \\
 1\ 1\ 1\ 0\ 1\ 0 \quad -6
 \end{array}$$

Si se hace con **enteros sin signo** (desplazamiento lógico) da igual que sea un número positivo o negativo, se ponen sólo **0s**

## Anatomía de la multiplicación sin signo

Para representar el producto de dos números **de n bits** hacen falta  $2n$  bits. El procedimiento humano se basa en sumas y desplazamientos

### Notación

- M = Multiplicando;  $m_i$  = bit i-ésimo
- Q = Multiplicador;  $q_i$  = bit i-ésimo
- P = Producto;  $p_i$  = bit i-ésimo

$$117 = 1101 \cdot (1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0)$$

$$P = M \times Q = \sum_{i=0}^{n-1} M q_i 2^i$$

			1	1	0	1
	x	1	0	0	1	
			1	1	0	1
			0	0	0	0
		0	0	0	0	0
		1	1	0	1	0
	0	1	1	1	0	1
2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>0</sup>
						Pesos

1310  
910

### Problemática de la multiplicación con signo

Hay que extender el signo del multiplicando y El peso del bit de signo del multiplicador es de  $-2^{n-1}$  (y no  $2^{n-1}$ ) → Osea que tienes que poner un "signo" en vez de por ejemplo usar un 1 o un 0 para saber si es positivo o negativo (creo)

### División

Dados un dividendo y un divisor, produce **dos resultados**: el **cociente** y el **resto** (o módulo)

Ejemplo sin signo: algoritmo humano de restas y desplazamientos

- Si NO cabe (X): 0 al cociente
- Si cabe (√): restar y 1 al cociente



Con signo: Por convención, el resto tiene **el mismo signo que el dividendo**

## Multiplicaciones Divisiones y Desplazamientos en el MIPS

### Instrucciones de desplazamiento

Son de la forma "operación **Rr, Ri, Desp**", donde **Desp** puede ser una constante o un registro y El desplazamiento máximo es de 31 posiciones. Sólo cuentan los 5 bits de menor peso de Long.

#### Instrucciones

Shamt: **shift amount**

tipo	formato R
izquierda	sll rd,rt,shamt sllv rd,rs,rt
derecha (lógico)	srl rd,rt,shamt srlv rd,rs,rt
derecha (aritmética)	sra rd,rt,shamt sra v rd,rs,rt

### Instrucciones de multiplicación y división generales

Dos registros especiales de 32 bits: **HI** y **LO**. Combinados forman un registro de 64 bits

#### Operaciones

mult \$2, \$3:	HI-LO ← \$2*\$3; Operandos con signo
multu \$2, \$3:	HI-LO ← \$2*\$3; Operandos positivos sin signo
div \$2, \$3:	LO ← \$2/\$3; HI ← \$2 mod \$3; Con signo
divu \$2, \$3:	LO ← \$2/\$3; HI ← \$2 mod \$3; Sin signo

Ninguna de estas instrucciones comprueba desbordamientos o división por cero: hay que hacerlo por software

También hay **pseudoinstrucciones** que permiten guardar directamente el resultado

#### Transferencia de resultados

- mfhi \$2: \$2 ← HI
- mflo \$2: \$2 ← LO

- Mul rd, rs, rt
- Div rd, rs, rt

cómo???



1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

ventajas

PRO



Di adiós a la publi en los apuntes y en la web



Acumula tickets para los sorteos



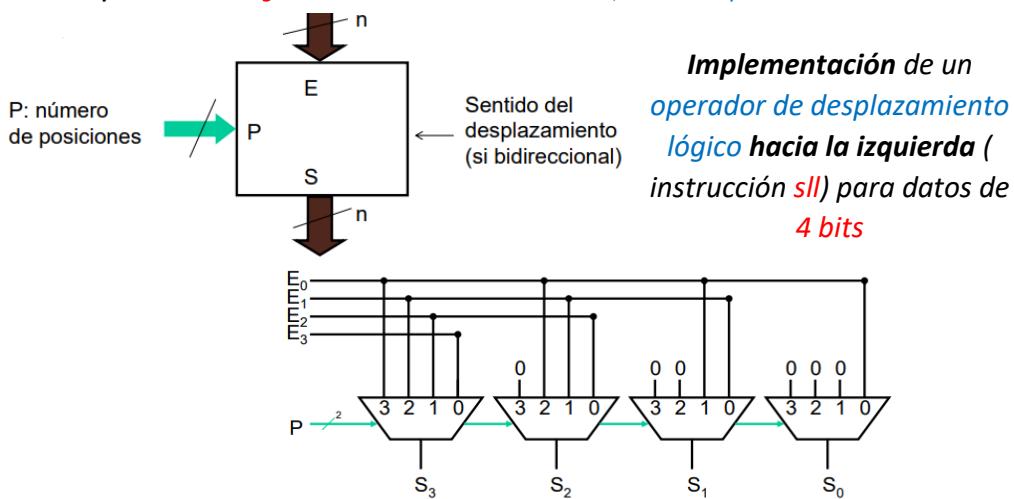
Descarga carpetas completas

estudia sin publi

WUOLAH PRO

## Barrel Shifter (Desplazador)

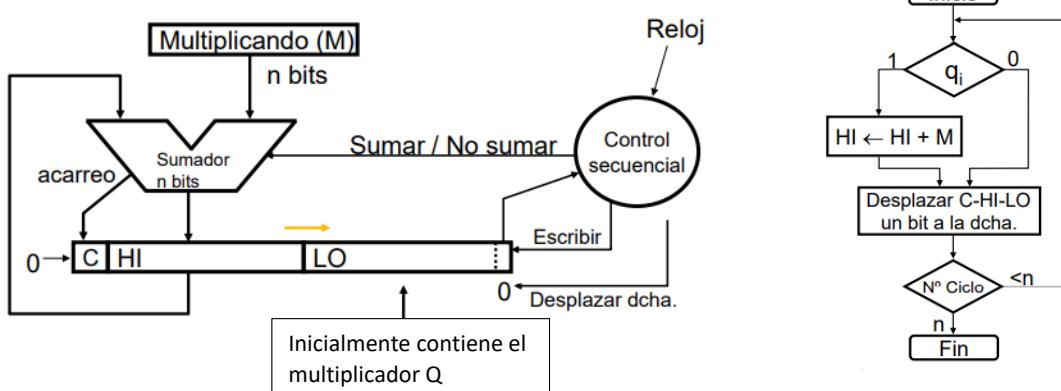
Un Barrel Shifter es un circuito que **permite realizar desplazamientos variables** sobre datos de  $n$  bits. Puede implementarse mediante multiplexores y Dependiendo del diseño, hace **desplazamientos lógicos o aritméticos** hacia la **derecha**, hacia la **izquierda** o **bidireccionales**



## Multiplicación

### Operadores de multiplicación sin signo

Operador secuencial para el **algoritmo de sumas y desplazamientos**, resultado con el **doble de bits que los multiplicandos**: **M** y **Q** de  $n$  bits; **P** de  $2n$  bits



**Resultado:** Al finalizar, **HI** y **LO** contienen el producto

**Unidad de control:** La unidad de control es un circuito secuencial síncrono gobernado por un reloj que marca los diferentes pasos de la multiplicación

El algoritmo requiere **n ciclos** y En cada ciclo hay que hacer, como máximo, hasta **una suma** y **un desplazamiento**: Reloj del multiplicador ≠ reloj del procesador

## Retardo y productividad del multiplicador

**T (período del reloj del multiplicador) =**

*Inspección Qi + suma + escribir en HI + desplazar C-HI-LO + evaluar el nº ciclo*

**Retardo-multiplicador =**  $n \cdot T + t$  inicialización

Puede ser que la inicialización se haga con un ciclo extra so:  $(n+1) \cdot T$

**Productividad =** 1 operación/retardo

Retardo en "ns", **Productividad** =  $(1000/\text{retardo}) \text{ MOPS}$  (megaOpreaciones por segundo)

**Ejemplo:**

$$\begin{aligned} n=4; M=1011_2; Q=0101_2 \\ (11_{10} \times 5_{10} = 55_{10}) \\ 0011 \quad 0111 \end{aligned}$$

Ciclo	Acción	C-HI-LO
0	Valores iniciales	0 0000 0101
1	$HI \leftarrow HI + M$	0 1011 0101
	Desplazar C-HI-LO 1 bit a la derecha	0 0101 1010
2	No sumar	0 0101 1010
	Desplazar C-HI-LO 1 bit a la derecha	0 0010 1101
3	$HI \leftarrow HI + M$	0 1101 1101
	Desplazar C-HI-LO 1 bit a la derecha	0 0110 1110
4	No sumar	0 0110 1110
	Desplazar C-HI-LO 1 bit a la derecha	0 0011 0111

## Operadores de multiplicación con signo

Se trata el **signo de M** y de **Q** por separado. Lo malo es que requiere Hardware extra para complementarlo.

La base es que, al tratar el signo por separado, luego haces la multiplicación sin signo y los juntas

```
Signo_Prod ← Signo(M) XOR Signo(Q);
si M < 0 entonces M ← -M; fin si;
si Q < 0 entonces Q ← -Q; fin si;
P ← M × Q; # multiplicador sin signo
si Signo_Prod = 1 entonces P ← -P; fin si;
```

Hay otros métodos mejores para tratar números con y sin signo: el algoritmo de Booth.

## Método de Booth

Consiste en transformar el número Q en otro número en binario siguiendo una norma (La tabla). Para ello coges bits de 2 en 2 (el 1 con el 2, el 2 con el 3...) de derecha a izquierda, y para traducirlo miras la tabla y pones TEXTUALMENTE 0, +1 o -1.

Después haces la multiplicación normal con el número M y el nuevo Q':

- Cuando sea 0 la línea es **todo 0**
- Cuando sea +1 la línea es el **número normal**
- Cuando sea -1 la línea **Complemento a2 de M**

Debe suponerse un bit **implícito a "0"** a la **derecha del LSB** para poder usarlo al transformar las parejas.

q <sub>i</sub>	q <sub>i-1</sub>	Dígito Booth
0	0	0
0	1	+1
1	0	-1
1	1	0

Después de poner todas las líneas según 0, +1 y -1 se suman todas las líneas de números

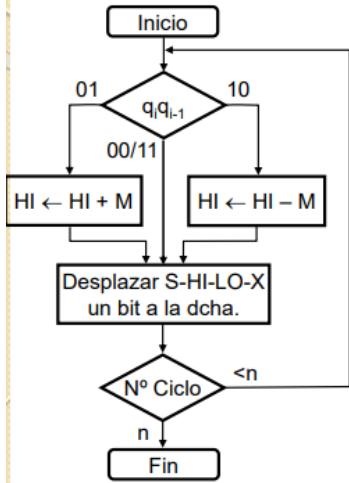
**Ejemplo** Obténgase el código Booth de  $1110\ 0111\ 0011$  ( $-39710$ )

Solución:

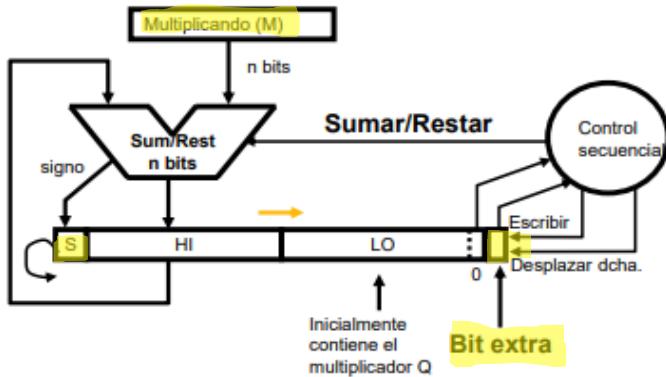
$0\ 0\ -1\ 0\ +1\ 0\ 0\ -1\ 0\ +1\ 0\ -1$

$$-1 \cdot 2^0 + 1 \cdot 2^2 - 1 \cdot 2^4 + 1 \cdot 2^7 - 1 \cdot 2^9 = -1 + 4 - 16 + 128 - 512 = -397$$

### Algoritmo y operador secuencial para Booth



El algoritmo **requiere n ciclos**: En cada ciclo hay que hacer una suma o una resta (según el signo que salga) más un desplazamiento



### Retardo y productividad del multiplicador de Booth

**T (período del reloj del multiplicador) =**

Inspección  $q_i q_{i-1}$  + resta + escribir en HI + + desplazar C-HI-LO + evaluar el nº ciclo

**Retardo-multiplicador =  $n \times T + t$  inicialización**

Si la inicialización se hace en un ciclo a parte: Retardo - multiplicador =  $(n + 1) \times T$

**Productividad = 1 operación/retardo**

Si retardo viene en ns, **Productividad =  $(1000/\text{retardo}) \text{ MOPS}$**

**Detalle del cálculo del bit de signo adicional:** Signo del resultado: si hay desbordamiento, invertir el MSB

**Ejemplo:**  $n = 4$ ;  $M = 0010_2$ ;  $Q = 1001_2$   
 $(2_{10} \cdot (-7)_{10}) = -14_{10}$

En un inicio en Lo estará el Q y en X el bit extra

Se observan por **pares los 2 números más a la derecha** (se empieza con  $X = 0$ , el extra).

Según se evalúa a 0, -1 o +1 se pone en S y Hi M, 0 o Ca2(M).

Ciclo	Acción	S-HI-LO-X
0	Valores iniciales	0 0000 <u>1001</u> 0 Bit extra
1	Caso 10: $HI \leftarrow HI - M$	1 1110 <u>1001</u> 0
	Desplazar S-HI-LO 1 bit a la derecha	1 1111 <u>0100</u> 1
2	Caso 01: $HI \leftarrow HI + M$	0 0001 <u>0100</u> 1
	Desplazar S-HI-LO 1 bit a la derecha	0 0000 <u>1010</u> 0
3	Caso 00: No hacer nada	0 0000 <u>1010</u> 0
	Desplazar S-HI-LO 1 bit a la derecha	0 0000 <u>0101</u> 0
4	Caso 10: $HI \leftarrow HI - M$	1 1110 <u>0101</u> 0
	Desplazar S-HI-LO 1 bit a la derecha	1 1111 <u>0010</u> 1

Después se desplazan a la derecha todos los números para poder repetir la operación.

Ahora, cuando pongas M en HI tendrás uno de los dígitos "graduado" en la parte del LO, por lo que no se reescribirá, solo se **reescribe** lo que **estaba en el HI**.

Todo el rato se hacen **desplazamientos entre el Singo (S) la Respuesta (HI), lo que está Siendo Q (LO) y el bit extra (X)**

El resultado final se quedará en Hi Lo, y dirás ¿!QUÉ COJONES!?

Pues mira mu fácil, si tienes en mente el esquemita de escalera de una multiplicación, cada vez pones las "sub multiplicaciones" un 0 más a la izquierda para luego sumar, no? Pues aquí haces eso, primero sumando y **luego desplazando** → los bits que entran en la parte del LO no se vuelven a modificar, xq por ejemplo la primera fila de la multiplicación, el último bit solo lo afecta el primer dígito de Q, por lo que lo mueves luego al LO para no volver a tocarlo

$$\begin{array}{r}
 & 0 & 1 & 1 & 0 & 1 \\
 & \times & 0 & -1 & 0 & -2 \\
 \hline
 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\
 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
 \hline
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 \hline
 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0
 \end{array} \quad (-78_{10})$$

## Recodificación por parejas de bits

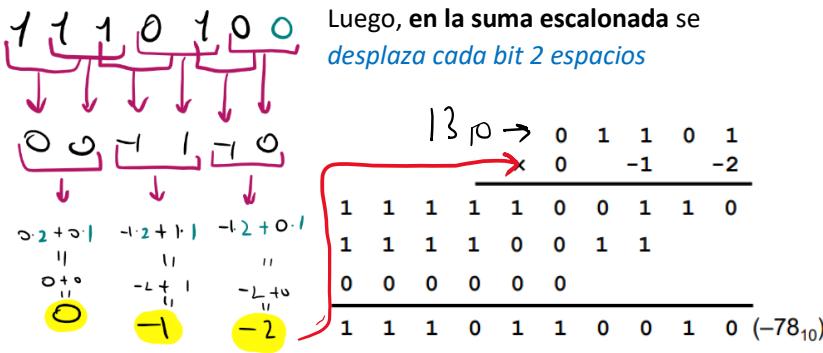
**Extensión de Booth** que **reduce a la mitad el número de dígitos de Q**, reduciendo así el número de productos intermedios a sumar. Es como aplicar el algoritmo de booth 2 veces

Hay 2 opciones, agrupar los bits de 3 en 3 y al sustituirse lo hacen según esta tabla o aplicas booth 2:

Se aplica una vez el algoritmo de booth 2 veces. La primera se hace normal, y la segunda, sobre el resultado, se obtiene el valor según el peso de cada bit al agruparlos por parejas

			Booth		Parejas	
$q_{i+1}$	$q_i$	$q_{i-1}$	$q'_{i+1}$	$q'_i$	$q''_i$	Acción
0	0	0	0	0	0	Nada
0	0	1	0	1	1	Sumar M
0	1	0	1	-1	1	Sumar M
0	1	1	1	0	2	Sumar 2xM
1	0	0	-1	0	-2	Restar 2xM
1	0	1	-1	1	-1	Restar M
1	1	0	0	-1	-1	Restar M
1	1	1	0	0	0	Nada

Ciclo	Acción	S-HI-LO-X
0	Valores iniciales	0 000000 111010 0
1	Caso 100: HI ← HI - 2M Desplazar S-HI-LO 2 bits a la derecha	1 100110 111010 0
2	Caso 101: HI ← HI - M Desplazar S-HI-LO 2 bits a la derecha	1 101100 101110 1
3	Caso 111: No hacer nada Desplazar S-HI-LO 2 bits a la derecha	1 111011 001011 1



El algoritmo requiere  $n/2$  ciclos, En cada ciclo hay que hacer hasta una suma o resta y dos desplazamientos

**T (período del reloj del multiplicador) =**

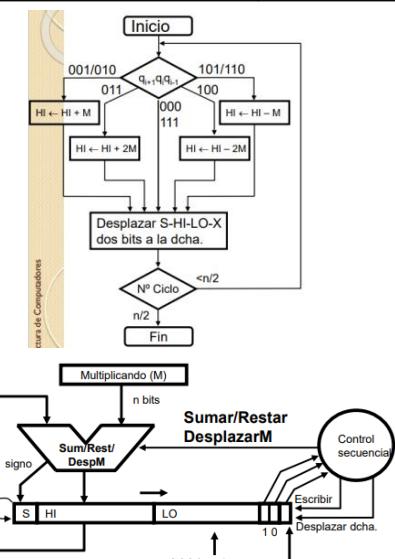
Inspección  $q_{i+1}q_iq_{i-1} + 2M + \text{resta} + \text{escribir en HI} + \text{desplazar C - HI - LO 2 posiciones} + \text{evaluar el } n^{\text{o}} \text{ ciclo}$

**Retardo-multiplicador** =  $\frac{n}{2} \cdot T + t \text{ inicialización}$

Si la **inicialización** se hace en un **ciclo a parte**:  $R - M = \left(\frac{n}{2} + 1\right) \cdot T$

**Productividad** = 1 **operación/retardo** Si retardo viene en ns,

**Productividad** = **(1000/retardo)** MOPS



Los registros HI y LO deben tener un número par de bits



Ábrete la Cuenta Online de BBVA y  
llévate 1 año de Wuolah PRO

cómo??  
↗



1/6

Este número es  
indicativo del  
riesgo del  
producto, siendo  
1/6 indicativo de  
menor riesgo y  
6/6 de mayor  
riesgo.

BBVA está  
adherido al  
Fondo de  
Garantía de  
Depósitos de  
Entidades de  
Crédito de  
España.  
La cantidad  
máxima  
garantizada es  
de 100.000 euros  
por la totalidad  
de los depósitos  
constituidos  
en BBVA por  
persona.

**Seguidme en wuolah “TurboApuntesPat” hay apuntes de todas las asignaturas, no te arrepentirás juju.** Sígueme en twitter @MaikPatataOtaku ;)  
Y en Instagram también juju @potatproductions :p

ventajas

PRO



Di adiós a la publi  
en los apuntes y  
en la web



Acumula tickets  
para los sorteos



Descarga  
carpetas  
completas

estudia sin publi

**WUOLAH PRO**

Reservados todos los derechos.  
No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.