Representación digital de los números

Fundamentos de Computadores

Escuela Politécnica Superior. U.A.M

Índice de la Unidad 6

U6. Representación digital de los números.

Escuela Politécnica Superior

Representación de números enteros, positivos y negativos

Recuerda: Un número binario natural, entero sin signo, utiliza un sistema numérico posicional.

- Con un número de n bits: $b_{n-1}\, b_{n-2} \ldots b_1 b_0$, se pueden representar **2^n** números diferentes en el rango **$[0, 2^n-1]$**.

$$\boxed{N = \sum_{i=0}^{n-1} b_i \times 2^i}$$

¿Cómo se representa en binario un entero con signo?

¿Cómo se representa en binario un número real?

Representación de números enteros, positivos y negativos

Signo-Magnitud

- Para un número de **n** bits (b_{n-1}, b_{n-2} … b_1, b_0), el bit más significativo (msb) señala el signo, los **n-1** bits restantes la magnitud.
  - Si msb = ‘0’, el número es positivo.
  - Si msb = ‘1’, el número es negativo.

$$N_{sm} = (-1)^{b_{n-1}} x \sum_{i=0}^{n-2} \mathrm{b}_i \times 2^i$$

- Ejemplo: escribir con 6 bits y representación signo-magnitud los números decimales +6, -6, +12, -24, +32 y -40.

+6 =	+12 =	+32 =
- 6 =	- 24 =	- 40 =

Escuela Politécnica Superior

Representación de números enteros, positivos y negativos

Signo-Magnitud

- Rango de la representación de enteros en signo-magnitud:

- Problemas de la representación de enteros en signo-magnitud:
  - ✓ Dos representaciones para el cero "±0" (000..00 y 100..00)
  - ✓ La extensión en bits del número no es igual para ambos signos
  - ✓ La suma de números con distinto signo (resta) no funciona bien
    - ✓ Ejem: (-6) + (+6) =>

```
  10000110
+ 00000110
  --------
  10001100 (-12, ¡error!)
```

Representación de números enteros, positivos y negativos

Complemento a 2

- Representa el valor de un número en un sistema binario posicional, en donde para un número de **n** bits (b_{n-1}, b_{n-2} ... b_1, b_0), el bit más significativo (msb) tiene el valor de -2^{n-1}.
  - Si msb = '0', el número es positivo.
  - Si msb = '1', el número es negativo.

$$\boxed{N_{c2} = b_{n-1} x(-2^{n-1}) + \sum_{i=0}^{n-2} \mathrm{b}_i \times 2^i}$$

- Cálculo del complemento a 2 de un número entero binario.

  1. Invertir todos los bits
  2. Sumarle 1 al resultado.

Representación de números enteros, positivos y negativos

Complemento a 2

- Ejemplo: escribir con 8 bits y representación c2 los números enteros dados en decimal:

+6 =	+89 =	+112 =	-125 =
- 6 =	- 24 =	- 75 =	- 142 =

- Ejemplo: calcular el valor en decimal de los números escritos en binario con 8 bits y representación en c2:

01110000 =	11010011 =
11110000 =	01000100 =

- Ejemplo: calcular el valor en decimal de los números escritos en hexadecimal y representación en c2:

0xC5 =	0x7F =
0x6D =	0x84 =

Escuela Politécnica Superior

Representación de números enteros, positivos y negativos

Complemento a 2

- Rango de la representación de enteros en Complemento a 2:

- La extensión de signo no modifica el valor del número:
  - Ejemplo 4=>8 bits, nº positivo: (+5) = 0101 = 00000101
  - Ejemplo (4=>8 bits, nº negativo: (-5) = 1011 = 11111011
- La operación de restar es equivalente a sumar el minuendo con el c2 del sustraendo:
  - ✓ Ejem: (+6) + (-6) =>

```
   00000110
+  11111010
-----------
   00000000 (¡correcto!)
```

Suma de números en binario

- Suma decimal

$$\begin{array}{r} 11 \leftarrow \text{carries} \\ 3734 \\ +\ \ 5168 \\ \hline 8902 \end{array}$$

- Suma Binaria

$$\begin{array}{r} 11 \leftarrow \text{carries} \\ 1011 \\ +\ \ 0011 \\ \hline 1110 \end{array}$$

Ejemplos

$$\begin{array}{r} 1001 \\ +\ \ 0101 \\ \hline \end{array}$$

$$\begin{array}{r} 1011 \\ +\ \ 0110 \\ \hline \end{array}$$

El problema del overflow (desbordamiento) en los sistemas digitales

- Los sistemas digitales operan con un número fijo de bits.
- El resultado de una operación (suma) puede sobrepasar el rango de representación de los bits utilizados.

Escuela Politécnica Superior

Operaciones en complemento a 2

Ejemplos: Utilizando números de 8 bits en la notación de c2, realizar las operaciones:

- ✓ Sumar (+45) + (+32)
- ✓ Restar (+45) - (+32)
- ✓ Sumar (-35) + (-27)
- ✓ Restar (+35) - (-27)
- ✓ Sumar (100) + (-12)
- ✓ Restar (-128) - (+64)

Sumador binario

- **Semisumador (1 bit):** Circuito combinacional para la suma aritmética de los dos bits de la entrada (a_i y b_i), obteniendo a la salida un bit para la suma y un bit para el acarreo (s_i y c_{i+1})

Tabla de verdad:

Bit1	Bit2	Suma	C_{out}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

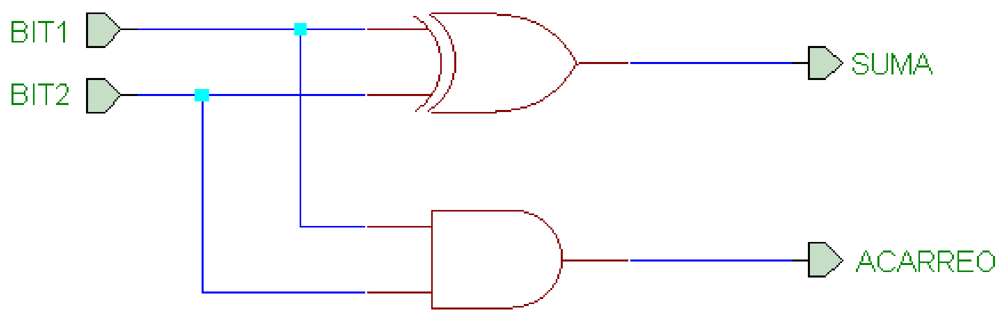
La ecuación para el bit de suma corresponde a una operación XOR:

$$\boxed{Suma = Bit1 \oplus Bit2}$$

La ecuación para el bit de acarreo corresponde a una AND:

$$\boxed{C_{out} = Bit1 \cdot Bit2}$$

Circuito:



Sumador binario

- ✓ **Sumador completo (1 bit):** Circuito combinacional para la suma aritmética de los dos bits de la entrada mas el acarreo del bit anterior (a_i, b_i y c_i), obteniendo a la salida un bit para la suma y un bit para el acarreo (s_i y c_{i+1})

Tabla de verdad:

C_{in}	Bit1	Bit2	Suma	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Donde:

C_{in}: Acarreo de entrada

C_{out}: Acarreo de salida

La ecuación para el bit de suma:

$$\boxed{Suma = (Bit1 \oplus Bit2) \oplus C_{in}}$$

La ecuación para el bit de acarreo:

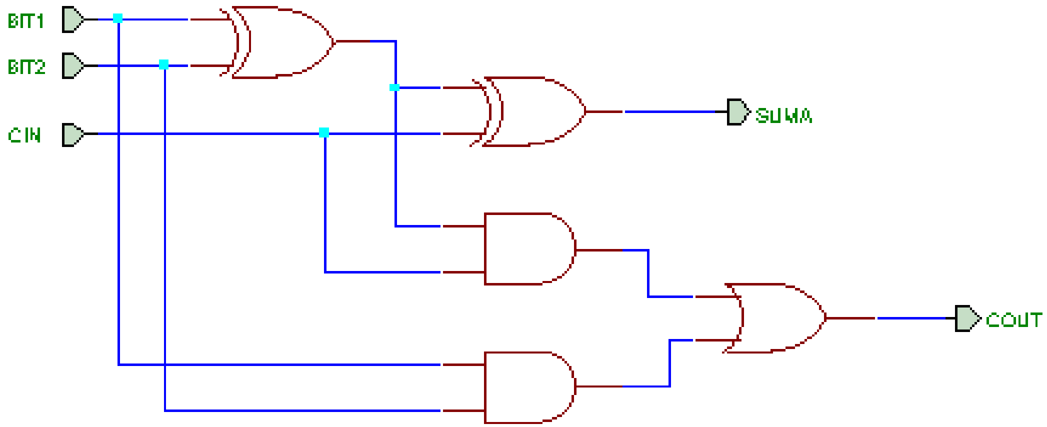
$$\boxed{C_{out} = Bit1 \cdot Bit2 + (Bit1 \oplus Bit2) \cdot C_{in}}$$

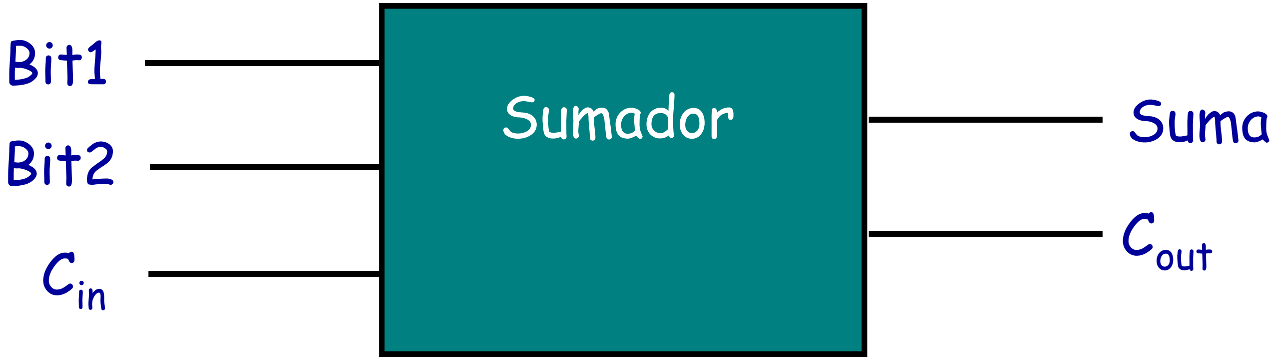
Escuela Politécnica Superior

Sumador binario

✓ Sumador completo. Circuito

Circuito esquemático

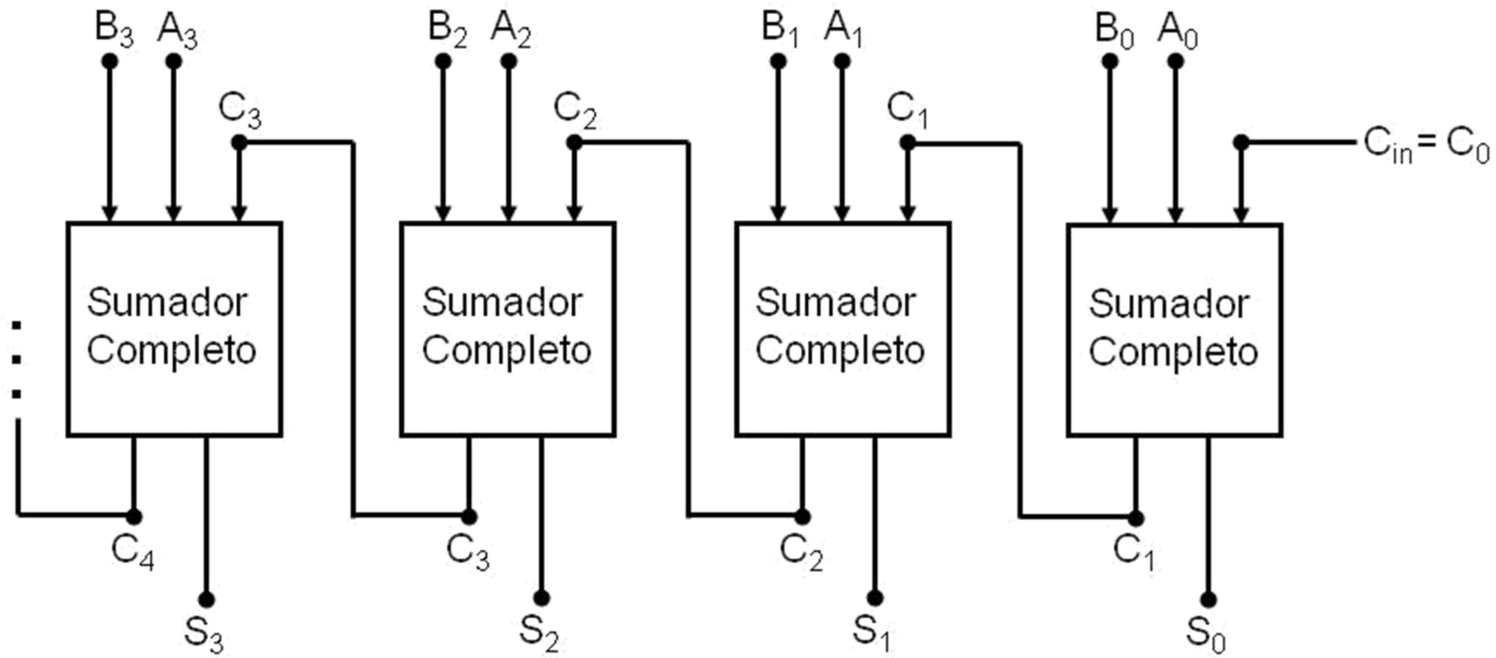




Esquema de bloque

Sumador binario

- ✓ **Circuito sumador para n bits en paralelo con acarreo en serie.**
  - ✓ Un circuito sumador elemental para cada bit.
  - ✓ Los bits del mismo peso se suman dos a dos
  - ✓ Para obtener cada suma parcial se necesita el acarro que se produce en la suma precedente.



Representación de números reales

✓ Representación en coma fija

Es una representación binaria numérico-posicional en la cual el número de bits dedicados a la parte entera y a la parte fraccionaria es fijo. Los números negativos se representan en complemento a 2:

$$\boxed{N = b_{n-1} x(-2^{n-1}) + \sum_{i=-k}^{n-2} b_i \times 2^i}$$

Ejemplos (n = 8; k = 2)

$30{,}3125_{10} =$

$0{,}375_{10} =$

$-0{,}1875_{10} =$

¡¡¡ Existe un problema de precisión !!!

Otros códigos binarios

Código decimal binario (BCD)

Es una representación para números enteros sin signo, en la que cada dígito decimal (0, 1, ..., 9) tiene su equivalente binario en 4 bits.

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

Ejemplos de números BCD

$30_{10} =$

$375_{10} =$

$1875_{10} =$

Escuela Politécnica Superior

Otros códigos binarios

Representación de texto. El estándar ASCII

ASCII (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange) es un código binario que se utiliza para representar caracteres. El código ASCII extendido utiliza 8 bits para identificar caracteres adicionales a un alfabeto tradicional.

Carácter	**ASCII**	**Carácter**	**ASCII**	**Carácter**	**ASCII**
0	30_{16}	A	41_{16}	"espacio"	20_{16}
1	31_{16}	B	42_{16}	%	25_{16}
2	32_{16}	C	43_{16}	~	$7E_{16}$
…	…	…	…	/	$2F_{16}$
7	37_{16}	a	61_{16}	ñ	$A4_{16}$
8	38_{16}	b	62_{16}	à	$A0_{16}$
9	39_{16}	c	63_{16}	@	64_{16}

Códigos para tratamiento de errores

Necesidad del tratamiento de errores

- Posibilidad de cometer errores
  - En un sistema informático la información circula entre diferentes elementos digitales y se almacena en otros dispositivos también digitales.
  - Puede haber errores debido a:
    - Ruidos en las comunicaciones
    - Defectos en las superficies de los discos, etc...
  - Los errores consisten en la modificación de la información desde que se emite (o almacena) hasta que se recibe (o se recupera).
    - Cambio de valor de algunos bits ($0 \Leftrightarrow 1$)

Escuela Politécnica Superior

Códigos para tratamiento de errores

- Códigos de paridad
  - **VRC** (**V**ertical **R**edundancy **C**hecking)
    - La información se coloca en bloques de longitud fija
    - A los bloques se les añade un bit llamado de paridad y que, normalmente, precede a la información

Criterios para la paridad

- Paridad par:
  - Nº total de "1" (en datos) par: Bit de paridad = 0
  - Nº total de "1" (en datos) impar: Bit de paridad = 1
- Paridad impar:
  - Nº total de "1" (en datos) par: Bit de paridad = 1
  - Nº total de "1" (en datos) impar: Bit de paridad = 0

Escuela Politécnica Superior

Códigos para tratamiento de errores

Comprobación de paridad

Completar el bit de paridad con criterio:

- ✓ (1) paridad impar
- ✓ (2) paridad par

1	2	Información						
		1	0	0	0	0	0	1
		0	1	0	1	1	1	1
		1	1	0	1	0	0	0
		1	1	1	0	1	1	1
		1	0	1	0	0	0	1
		0	1	1	1	1	1	1
		0	1	1	1	0	0	1
		0	0	0	1	1	0	1

Escuela Politécnica Superior

Códigos para tratamiento de errores

- **Paridad vertical, longitudinal y cruzada**

  La información se coloca en grupos de (m) bloques de longitud fija (k) como matriz k x m o m x k

  Ejemplo: Se quiere enviar la información "IBM" en ASCII (7 bits), es decir: 49_{16} 42_{16} $4D_{16}$ = $1001001\ 1000010\ 1001101_2$ (m=3, k=7)

  Si paridad "par", se añade:

  - Bit para VRC criterio par (verde, primera columna)
  - Bit para LRC criterio par (azul, última fila)
  - Bit de paridad cruzada criterio par (rojo)

VCR

1	1	0	0	1	0	0	1
0	1	0	0	0	0	1	0
0	1	0	0	1	1	0	1
1	1	0	0	0	1	1	0

Paridad Cruzada

LCR

Se transmite: $C9424DC6_{16}$

Códigos para tratamiento de errores

- **Checksum**
  - Añade uno o más bytes a la información para alcanzar un resultado conocido en la suma total.

  Ej.: Dato: 37 4A. Cheksum: 7F (37+4A+7F = 1**00**)

  Información transmitida: 37 4A 7F

- **Códigos polinómicos o de redundancia cíclica (CRC)**
  - Añade bits a la información para alcanzar una división exacta por un polinomio conocido.

  Ej.: $G(x)=(x^3+x+1)$. Dato: 11000011. {11000011 mod 1011 =1000}

  Información transmitida: 110000111000

- **Códigos i en n**
  - Utiliza códigos con el mismo número de bits de valor ‘1’.

  Ej: código 5043210 (2 en 7): 0->0100001; 1->0100010; 2->0100100…

  …7->1000100; 8->1001000; 9->1010000

Escuela Politécnica Superior