

# **ELECTRÓNICA DIGITAL**

## **TEMA 1**

### **CÓDIGOS BINARIOS**

# CÓDIGOS BINARIOS

## CÓDIGO BINARIO NATURAL

Sistema binario de numeración	Sistema decimal de numeración
0 0 0 0	0
0 0 0 1	1
0 0 1 0	2
0 0 1 1	3
0 1 0 0	4
0 1 0 1	5
0 1 1 0	6
0 1 1 1	7
1 0 0 0	8
1 0 0 1	9
1 0 1 0	10
1 0 1 1	11
1 1 0 0	12
1 1 0 1	13
1 1 1 0	14
1 1 1 1	15

# **CÓDIGOS BINARIOS**

## **CÓDIGO BCD NATURAL**

*(BCD: Binary Coded decimal)*

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

# CÓDIGOS BINARIOS

## Sistema de numeración hexadecimal

Sistema binario de numeración	Sistema hexadecimal de numeración
0 0 0 0	0
0 0 0 1	1
0 0 1 0	2
0 0 1 1	3
0 1 0 0	4
0 1 0 1	5
0 1 1 0	6
0 1 1 1	7
1 0 0 0	8
1 0 0 1	9
1 0 1 0	A
1 0 1 1	B
1 1 0 0	C
1 1 0 1	D
1 1 1 0	E
1 1 1 1	F

# **CÓDIGOS BINARIOS**

## **Códigos continuos**

**Códigos en los que las combinaciones correspondientes a números decimales consecutivos son adyacentes.**

## **Combinaciones binarias adyacentes**

**Difieren solamente en un bit.**

### **Código continuo cíclico**

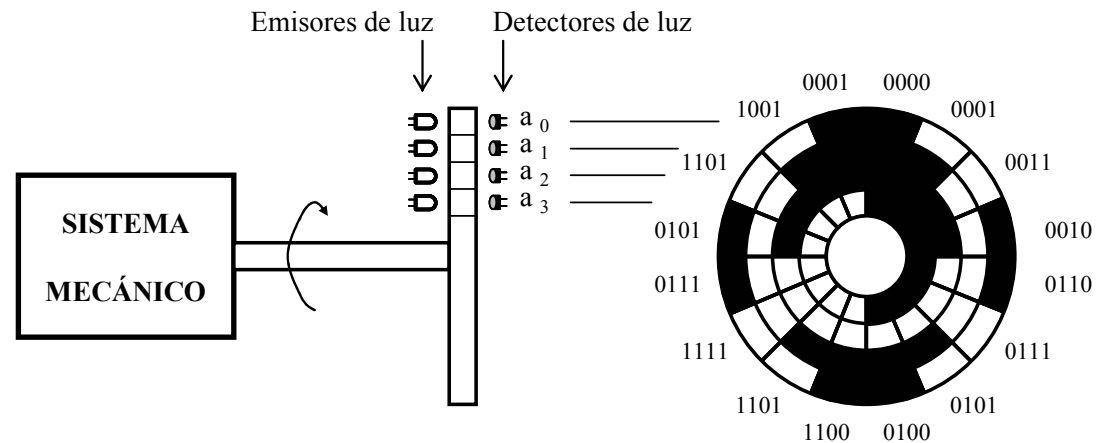
**La última combinación es adyacente a la primera. Los códigos continuos y cíclicos mas utilizados son:**

- Código reflejado o Gray.**
- Codigo Johnson**

# CÓDIGOS BINARIOS

## CÓDIGO GRAY

2 bits	3 bits	4 bits
00	000	0000
01	001	0001
11	011	0011
10	010	0010
	110	0110
	111	0111
	101	0101
	100	0100
		1100
		1101



# **CÓDIGOS BINARIOS**

## **CÓDIGO JOHNSON**

**Código utilizado en contaje**

0	00000
1	00001
2	00011
3	00111
4	01111
5	11111
6	11110
7	11100
8	11000
9	10000

# CÓDIGOS BINARIOS

**CÓDIGOS  
 ALFANUMÉRICOS  
 Código ASCII  
 de 6 bits  
 (AMERICAN STANDARD  
 CODE FOR INFORMATION  
 INTERCHANGE)**

					$b_5 b_4$			
					00	01	10	11
$b_3$	$b_2$	$b_1$	$b_0$	HEX	0	1	2	3
0	0	0	0	0	@	P	SP	0
0	0	0	1	1	A	Q	!	1
0	0	1	0	2	B	R	»	2
0	0	1	1	3	C	S	#	3
0	1	0	0	4	D	T	\$	4
0	1	0	1	5	E	U	%	5
0	1	1	0	6	F	V	&	6
0	1	1	1	7	G	W	'	7
1	0	0	0	8	H	X	(	8
1	0	0	1	9	I	Y	)	9
1	0	1	0	A	J	Z	*	:
1	0	1	1	B	K	[	+	;
1	1	0	0	C	L	\	,	<
1	1	0	1	D	M	]	-	=
1	1	1	0	E	N	↑	.	>
1	1	1	1	F	O	←	/	?

SP = equivale a espacio en blanco (*Space*)



# CÓDIGOS BINARIOS

## CÓDIGOS ALFANUMÉRICOS CÓDIGO ASCII DE 7 bits

					b <sub>6</sub> b <sub>5</sub> b <sub>4</sub>																							
					0	0	0	0	0	1	0	1	0	0	1	1	1	0	0	1	0	1	1	1	0	1	1	1
b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>	HEX	0	1	2	3	4	5	6	7																
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p																
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q																
0	0	1	0	2	STX	DC2	"	2	B	R	b	r																
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s																
0	1	0	0	4	EOT	DC4	\$	4	D	T	d	t																
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u																
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v																
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w																
1	0	0	0	8	BS	CAN	(	8	H	X	h	x																
1	0	0	1	9	HT	EM	)	9	I	Y	i	y																
1	0	1	0	A	LF	SUB	*	:	J	Z	j	z																
1	0	1	1	B	VT	ESC	+	;	K	[	k	{																
1	1	0	0	C	FF	FS	,	<	L	\	l																	
1	1	0	1	D	CR	GS	-	=	M	]	m	}																
1	1	1	0	E	SO	RS	.	>	N	^	n	~																
1	1	1	1	F	SI	US	/	?	O	_	o	DEL																

# CÓDIGOS BINARIOS

## CÓDIGO UNO ENTRE $N$ (*ONE HOT*).

Estados	Código uno entre n		
$E_0$	0	0	1
$E_1$	0	1	0
$E_2$	1	0	0

**Código utilizado para codificar los estados internos de un sistema digital**

## **CODIGOS DETECTORES DE ERRORES (*ERROR DETECTING CODES*)**

Para establecer la condición necesaria y suficiente para que un código binario permita detectar errores hay que definir el concepto de distancia mínima de un código.

### **Distancia entre dos combinaciones binarias**

Número de bits de una de ellas que deben ser modificados para obtener la otra.

### **Distancia mínima de un código**

Es la menor de las distancias entre dos combinaciones binarias cualesquiera pertenecientes al mismo. El valor de la distancia mínima de los códigos anteriores es la unidad y, por tanto, un error en uno solo de los bits de una combinación binaria perteneciente a cualquiera de ellos, puede convertirla en otra perteneciente al mismo y hacer que el error no sea detectable.

De todo lo dicho se deduce que, para que un código pueda detectar errores su distancia mínima ha de ser superior a la unidad.

# **CODIGOS DETECTORES DE ERRORES**

**Las ventajas que presentan el código binario natural y el BCD natural para realizar operaciones aritméticas hacen que en la práctica resulte más adecuado generar a partir de ellos combinaciones binarias que faciliten la detección de errores. En este caso se añaden los bits adecuados a cada combinación del código correspondiente y se obtiene una combinación que tiene información redundante (*Redundant*), es decir que no sería necesaria si no se quisiese detectar errores.**

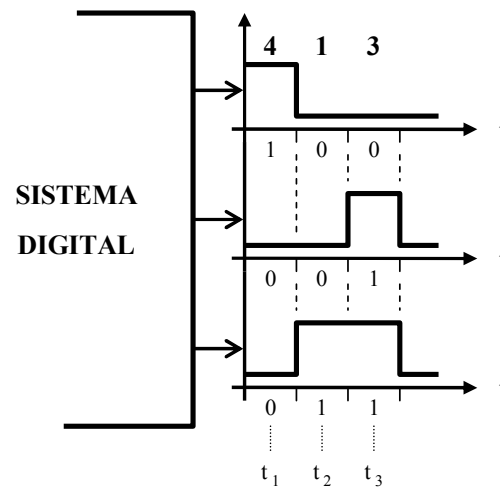
**Existen diversas maneras de generar combinaciones redundantes, que están relacionadas con la forma en que se transmite la información que puede ser:**

- En formato paralelo**
- En formato serie.**

# CODIGOS DETECTORES DE ERRORES

## Formato paralelo

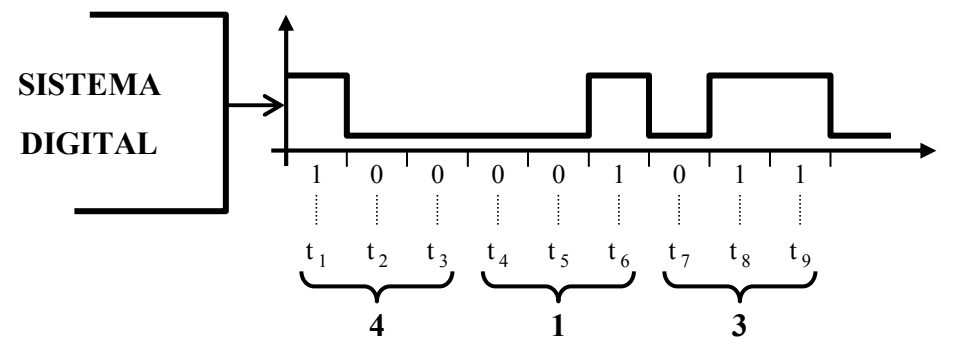
- La información se representa mediante  $n$  señales diferente
- Se utiliza especialmente para transferir información a distancias pequeñas.
- Se lleva a cabo a través de  $n$  canales de comunicación.



# CODIGOS DETECTORES DE ERRORES

## Formato serie

- La información esta constituida por una secuencia de niveles cero y uno de una única señal.
- Se utiliza para transmitir información a distancias elevadas a través de un único canal de comunicaciones.



# **DETECCIÓN DE ERRORES**

## **Códigos de distancia unidad**

**El número de unos de algunas de sus combinaciones binarias es par y el de otras es impar.**

## **Código de paridad constante**

**Se obtiene añadiendo a las combinaciones de un código de distancia unidad un bit llamado de paridad (*Parity bit*)**

**Puede ser:**

- De paridad par (*Even parity code*)**
- De paridad impar (*Odd parity code*)**

# DETECCIÓN DE ERRORES

**Código decimal codificado en binario natural (BCD natural) con bit de paridad impar**

Sistema decimal de numeración	Código BCD natural con bit de paridad impar				
	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>	BPI
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	1



# **DETECCIÓN DE ERRORES**

**Posibilidad de utilizar códigos detectores de errores de paridad constante, obtenidos a partir de los códigos de distancia mínima igual a la unidad como el binario natural o el BCD natural**

**Es una de las características de los sistemas electrónicos digitales que los hacen idóneos para la transmisión de información, no sólo en paralelo sino también en serie.**

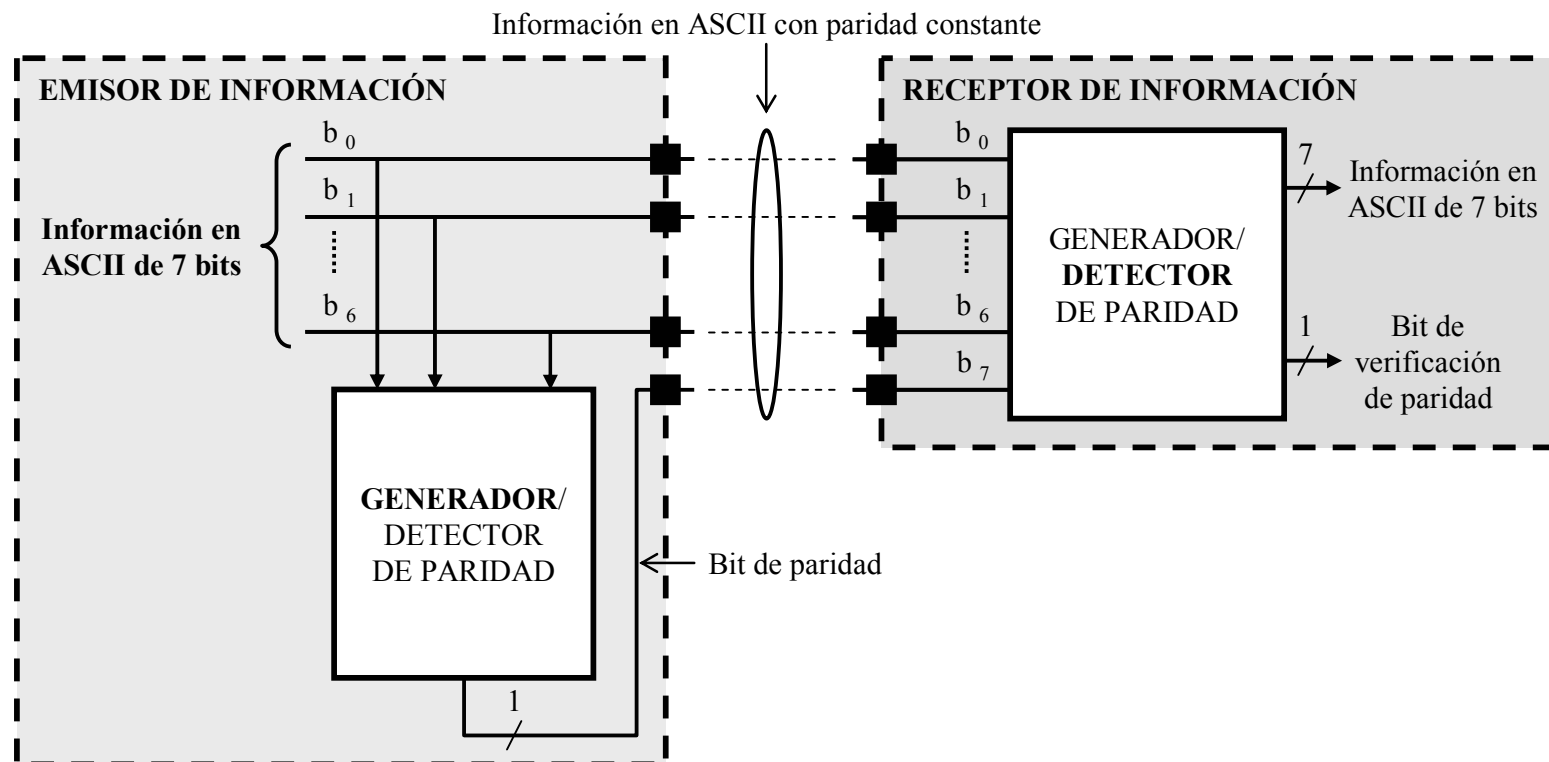
## **Detección de errores en los códigos de paridad constante**

**Comprobación de si es par (códigos de paridad par) o impar (códigos de paridad impar) el número de unos de cada combinación.**

# CODIGOS DETECTORES DE ERRORES

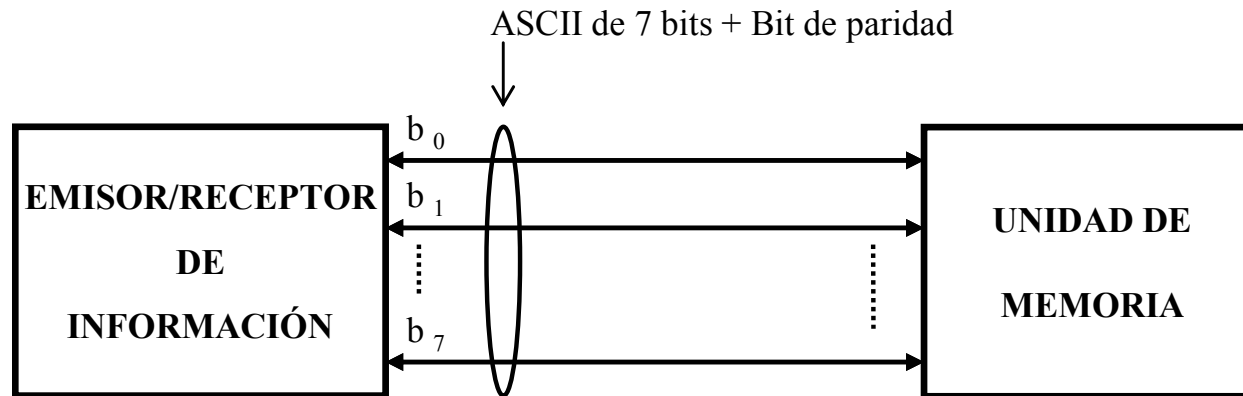
## Detección de errores en la comunicación en paralelo mediante bits de paridad

Esquema de bloques básico de un sistema de comunicación digital en paralelo con capacidad de detección de errores de un bit.



## CODIGOS DETECTORES DE ERRORES

### Detección de errores en la comunicación en paralelo mediante bits de paridad



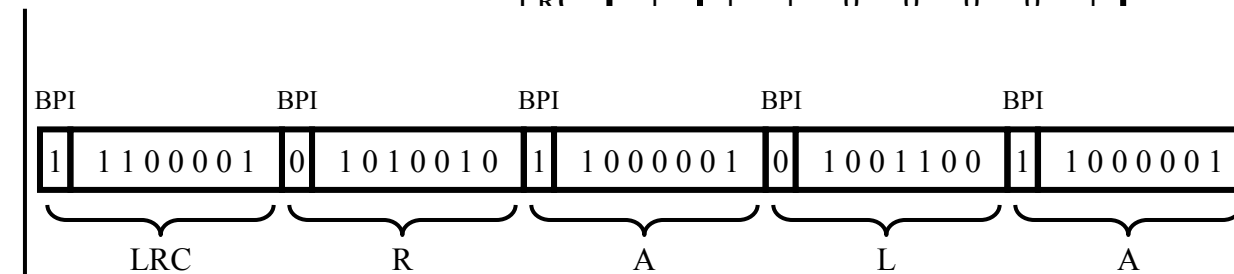
Escritura y lectura de información, codificada en el código ASCII de 7 bits ( $b_0$  a  $b_6$ ) y 1 bit adicional ( $b_7$ ) de paridad (redundante), en una memoria de entrada y salida en paralelo.

# DETECCIÓN DE ERRORES

Detección de errores en la comunicación serie mediante bits de paridad

- VRC (acrónimo de *Vertical Redundancy Check*)
- LRC (acrónimo de *Longitudinal Redundancy Check*)

	BPI							
	(VRC) b <sub>6</sub>	b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>	
Información a transmitir	A	1	1	0	0	0	0	1
	L	0	1	0	0	1	1	0
	A	1	1	0	0	0	0	1
	R	0	1	0	1	0	0	1
LRC	1	1	1	0	0	0	0	1



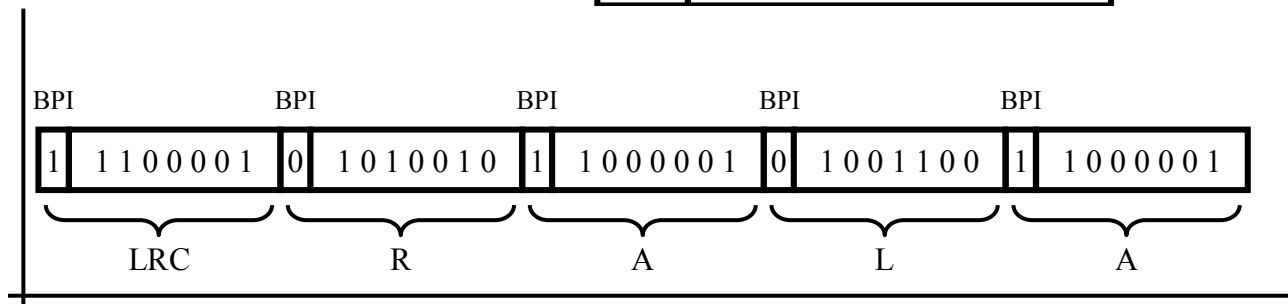
Fundamentos de la detección de errores en una comunicación digital en serie mediante paridad constante vertical y longitudinal

# DETECCIÓN DE ERRORES

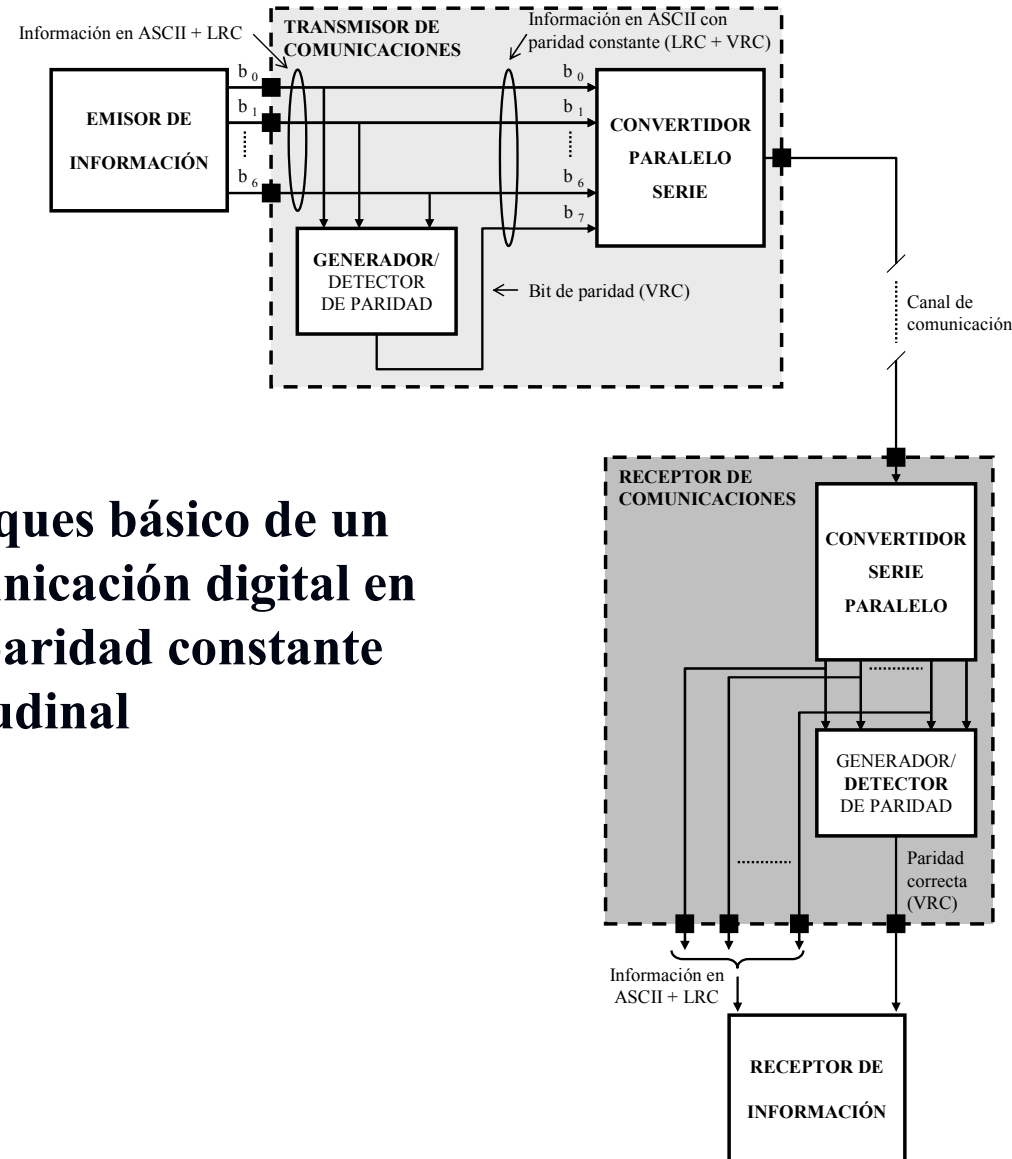
Detección de errores mediante bits de paridad en la transmisión de caracteres.

- VRC (acrónimo de *Vertical Redundancy Check*)
- LRC (acrónimo de *Longitudinal Redundancy Check*)

	BPI (VRC)						
	b <sub>6</sub>	b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>
Información a transmitir	A	1	1	0	0	0	1
	L	0	1	0	1	1	0
	A	1	1	0	0	0	1
	R	0	1	0	1	0	1
LRC	1	1	1	0	0	0	1



# DETECCIÓN DE ERRORES

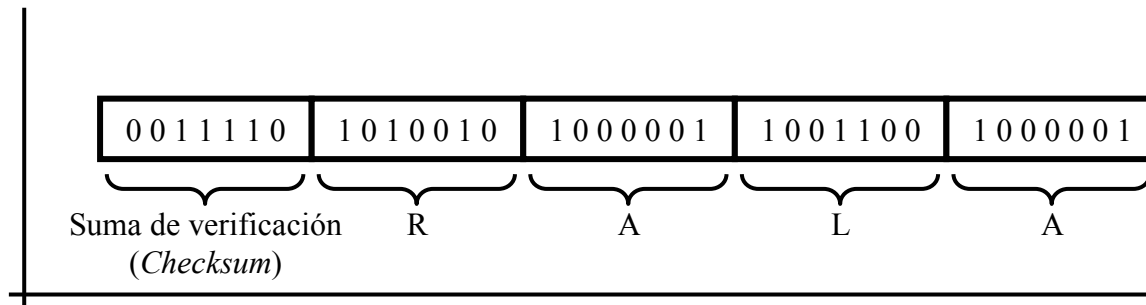


Esquema de bloques básico de un sistema de comunicación digital en serie mediante paridad constante vertical y longitudinal

# DETECCIÓN DE ERRORES

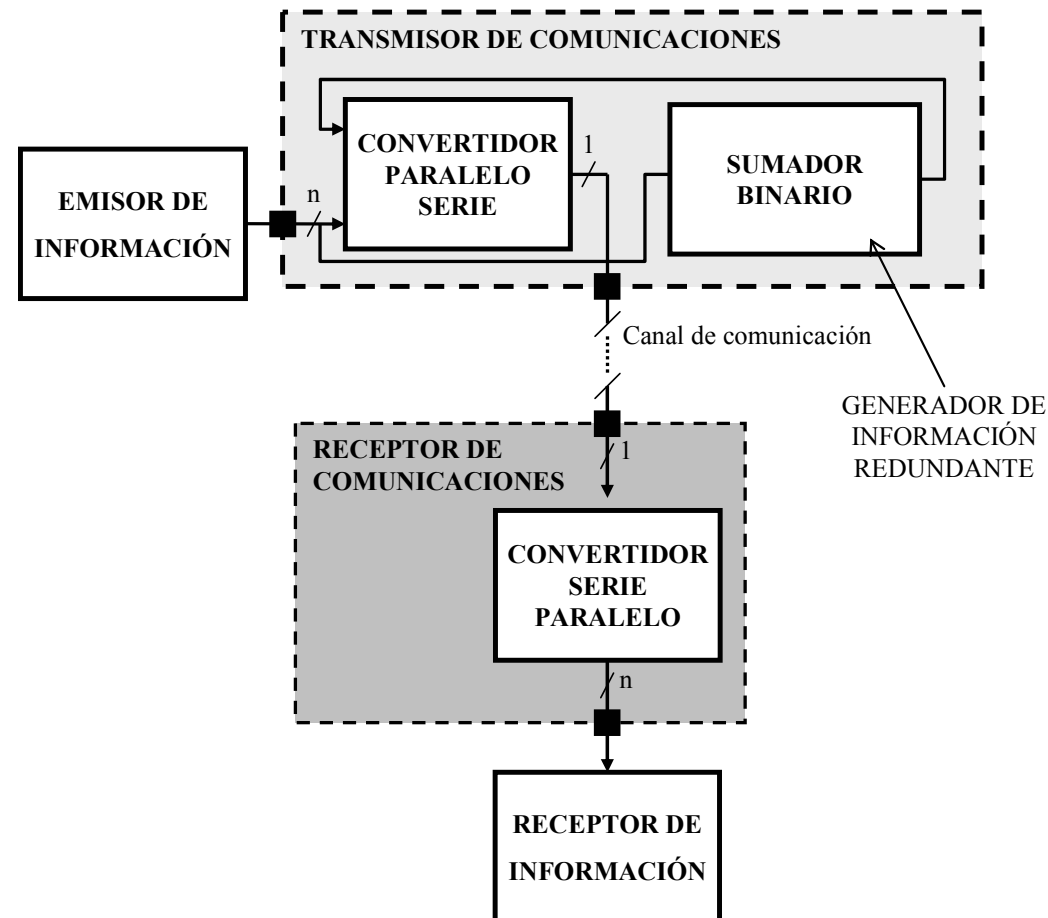
## Detección de errores mediante suma en la transmisión de caracteres

		b <sub>6</sub>	b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>
Información a transmitir	A	1	0	0	0	0	0	1
	L	1	0	0	1	1	0	0
	A	1	0	0	0	0	0	1
	R	1	0	1	0	0	1	0
Suma de verificación (Checksum)		0	0	1	1	1	1	0



# DETECCIÓN DE ERRORES

Esquema de bloques de un sistema de comunicación digital en serie que detecta errores mediante suma.





## DETECCIÓN DE ERRORES

### *Detección de errores mediante redundancia cíclica*

Se utiliza cuando no solo se transmiten caracteres.

- Es conocida como *verificación mediante redundancia cíclica o CRC* (acrónimo de *Cyclic Redundancy Check*) y se basa en la división binaria.
- A partir de la información útil formada por un conjunto o bloque de  $n_1$  bits se obtiene una combinación de  $n_2$  bits que cumple la condición de que el bloque formado por  $n_1$  y  $n_2$  es divisible por un número determinado, es decir el resto de la división es cero. En la recepción, la información formada por  $n_1$  y  $n_2$  se divide por el citado número y se detecta la presencia de errores si el resto obtenido es distinto de cero.
- La generación de la información redundante se realiza siempre en la práctica en el transmisor de comunicaciones porque la división binaria se lleva a cabo en serie mediante un registro de desplazamiento adecuadamente realimentado.

## DETECCIÓN DE ERRORES

Esquema de bloques de un sistema de comunicación digital en serie que detecta errores mediante redundancia cíclica.

