

Códigos Binarios



Rancho el Milagro
Cerralvo N.L.

**"No basta con tener conocimiento;
lo esencial es poner en práctica ese
conocimiento"**

JAGG

	Actividad	Puntos	Fecha límite
PF6	Solución del examen	F	
PF7	Diseño Combinacional con HDL	F	
PF8	Multiplexor	5	
AF3	Decodificador con Display	10	
PF9	Flip Flops	F	
PF10	Pulsos de sincronía	5	
AF4	Diseño Secuencial	10	
AF5	PIA	40	Día del examen

Códigos Binarios

Códigos alfanuméricos

Código ASCII, Código EBCDIC

Códigos numéricos Binarios:

***$N_{(2)}$, Gray, JOHNSON, de Anillo,
Wan-Go, 2 de 5, 3 de 6, entre otros***

Decimales representados en Binario:

BCD, Exceso3, 2421, 5211, 8 4-2-1

Códigos alfanuméricos

Código ASCII (0 -127)

American Standard Code for Information Interchange

Caracteres no imprimibles				Caracteres imprimibles											
Nombre	Dec	Hex	Car.	Dec	Hex	Car.	Dec	Hex	Car.	Dec	Hex	Car.	Dec	Hex	Car.
Nulo	0	00	NUL	32	20	Espacio	64	40	@	96	60	`			
Inicio de cabecera	1	01	SOH	33	21	!	65	41	A	97	61	a			
Inicio de texto	2	02	STX	34	22	"	66	42	B	98	62	b			
Fin de texto	3	03	ETX	35	23	#	67	43	C	99	63	c			
Fin de transmisión	4	04	EOT	36	24	\$	68	44	D	100	64	d			
enquiry	5	05	ENQ	37	25	%	69	45	E	101	65	e			
acknowledge	6	06	ACK	38	26	&	70	46	F	102	66	f			
Campanilla (beep)	7	07	BEL	39	27	'	71	47	G	103	67	g			
backspace	8	08	BS	40	28	(	72	48	H	104	68	h			
Tabulador horizontal	9	09	HT	41	29	)	73	49	I	105	69	i			
Salto de línea	10	0A	LF	42	2A	*	74	4A	J	106	6A	j			
Tabulador vertical	11	0B	VT	43	2B	+	75	4B	K	107	6B	k			
Salto de página	12	0C	FF	44	2C	,	76	4C	L	108	6C	l			
Retorno de carro	13	0D	CR	45	2D	-	77	4D	M	109	6D	m			
Shift fuera	14	0E	SO	46	2E	.	78	4E	N	110	6E	n			
Shift dentro	15	0F	SI	47	2F	/	79	4F	O	111	6F	o			
Escape línea de datos	16	10	DLE	48	30	0	80	50	P	112	70	p			
Control dispositivo 1	17	11	DC1	49	31	1	81	51	Q	113	71	q			
Control dispositivo 2	18	12	DC2	50	32	2	82	52	R	114	72	r			
Control dispositivo 3	19	13	DC3	51	33	3	83	53	S	115	73	s			
Control dispositivo 4	20	14	DC4	52	34	4	84	54	T	116	74	t			
neg acknowledge	21	15	NAK	53	35	5	85	55	U	117	75	u			
Sincronismo	22	16	SYN	54	36	6	86	56	V	118	76	v			
Fin bloque transmitido	23	17	ETB	55	37	7	87	57	W	119	77	w			
Cancelar	24	18	CAN	56	38	8	88	58	X	120	78	x			
Fin medio	25	19	EM	57	39	9	89	59	Y	121	79	y			
Sustituto	26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z			
Escape	27	1B	ESC	59	3B	;	91	5B	[	123	7B	{			
Separador archivos	28	1C	FS	60	3C	<	92	5C	\	124	7C				
Separador grupos	29	1D	GS	61	3D	=	93	5D	]	125	7D	}			
Separador registros	30	1E	RS	62	3E	>	94	5E	^	126	7E	~			
Separador unidades	31	1F	US	63	3F	?	95	5F	_	127	7F	DEL			

Códigos alfanuméricos

Código ASCII extendido (128-255) American Standard Code for Information Interchange

128	Ç	144	É	160	á	176	░	193	⊥	209	⌒	225	ß	241	±
129	ü	145	æ	161	í	177	▒	194	⌒	210	π	226	Γ	242	≥
130	é	146	Æ	162	ó	178	▓	195	⌒	211	ℓ	227	π	243	≤
131	â	147	ô	163	ú	179		196	—	212	ℓ	228	Σ	244	∫
132	ä	148	ö	164	ñ	180	⌒	197	+	213	ℓ	229	σ	245	∫
133	à	149	ò	165	Ñ	181	⌒	198	⌒	214	π	230	μ	246	÷
134	â	150	û	166	²	182	⌒	199	⌒	215	⌒	231	τ	247	≈
135	ç	151	ù	167	°	183	π	200	ℓ	216	⌒	232	Φ	248	°
136	ê	152	—	168	¿	184	⌒	201	ℓ	217	⌒	233	⊕	249	.
137	ë	153	Ö	169	—	185	⌒	202	⌒	218	⌒	234	Ω	250	.
138	è	154	Ü	170	¬	186	⌒	203	⌒	219	■	235	δ	251	√
139	í	156	£	171	½	187	⌒	204	⌒	220	■	236	∞	252	—
140	î	157	¥	172	¼	188	⌒	205	=	221	■	237	φ	253	²
141	ï	158	—	173	¡	189	⌒	206	⌒	222	■	238	ε	254	■
142	Ä	159	ƒ	174	«	190	⌒	207	⌒	223	■	239	∧	255	
143	Å	192	Ł	175	»	191	⌒	208	⌒	224	α	240	≡		

Códigos alfanuméricos

Código EBCDIC

Extended Binary Coded Decimal Interchange Code

		Posicion 0 y 1															
		00 Bit posiciones 2 y 3				01 Bit posiciones 2 y 3				10 Bit posiciones 2 y 3				11 Bit posiciones 2 y 3			
		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
Posiciones 4,5,6 y 7	0000						&	.						>	<	#	0
	0001							/						A	J		1
	0010													B	K		2
	0011													C	L	S	3
	0100													D	M	U	4
	0101													E	N	V	5
	0110													F	O	W	6
	0111													G	P	X	7
	1000													H	Q	Y	8
	1001								"					I	R	Z	9
	1010					?	!		:								
	1011					.	\$	,	#								
	1100						*	%	@								
	1101					(	)		'								
	1110					+	:	-	=								
	1111																

Código Binario Base 2 $N_{(2)}$

Es un sistema de notación posicional que solo utiliza dos dígitos 0 o 1 para cada una de las posiciones.

Ponderación

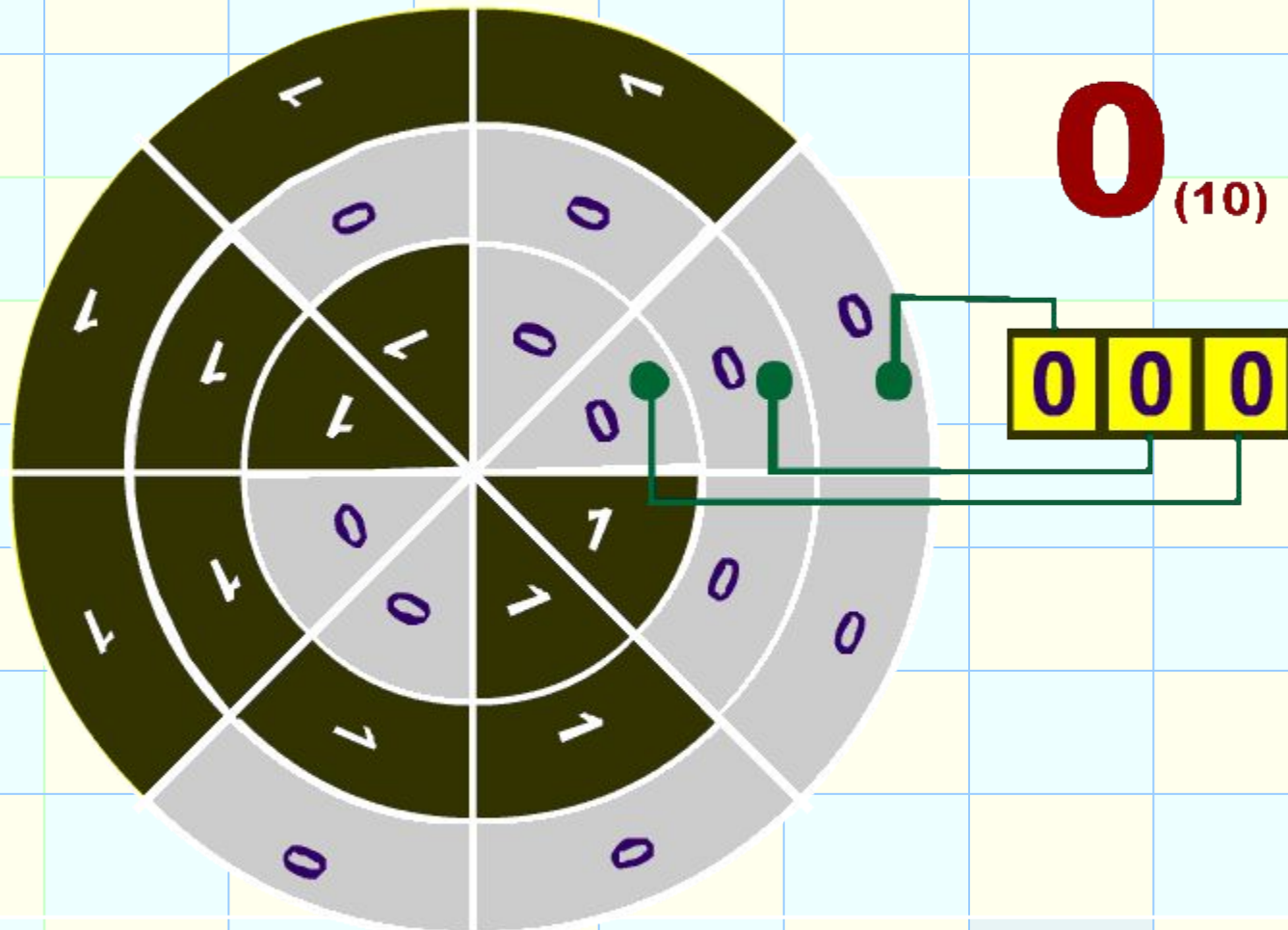
El sistema binario es un sistema de notación posicional ***ponderado***.

Notación Posicional

128	64	32	16	8	4	2	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
H	G	F	E	D	C	B	A

$$N = \underline{H \cdot 128 + G \cdot 64 + F \cdot 32 + E \cdot 16 + D \cdot 8 + C \cdot 4 + B \cdot 2 + A}$$

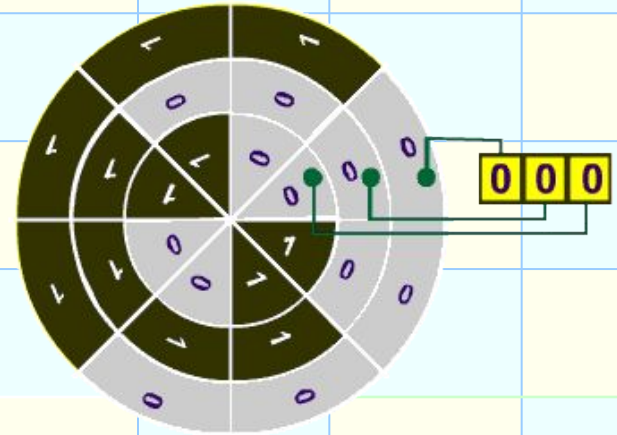
Cual es la característica o patrón que describe este código ?



Códigos binarios

Código GRAY

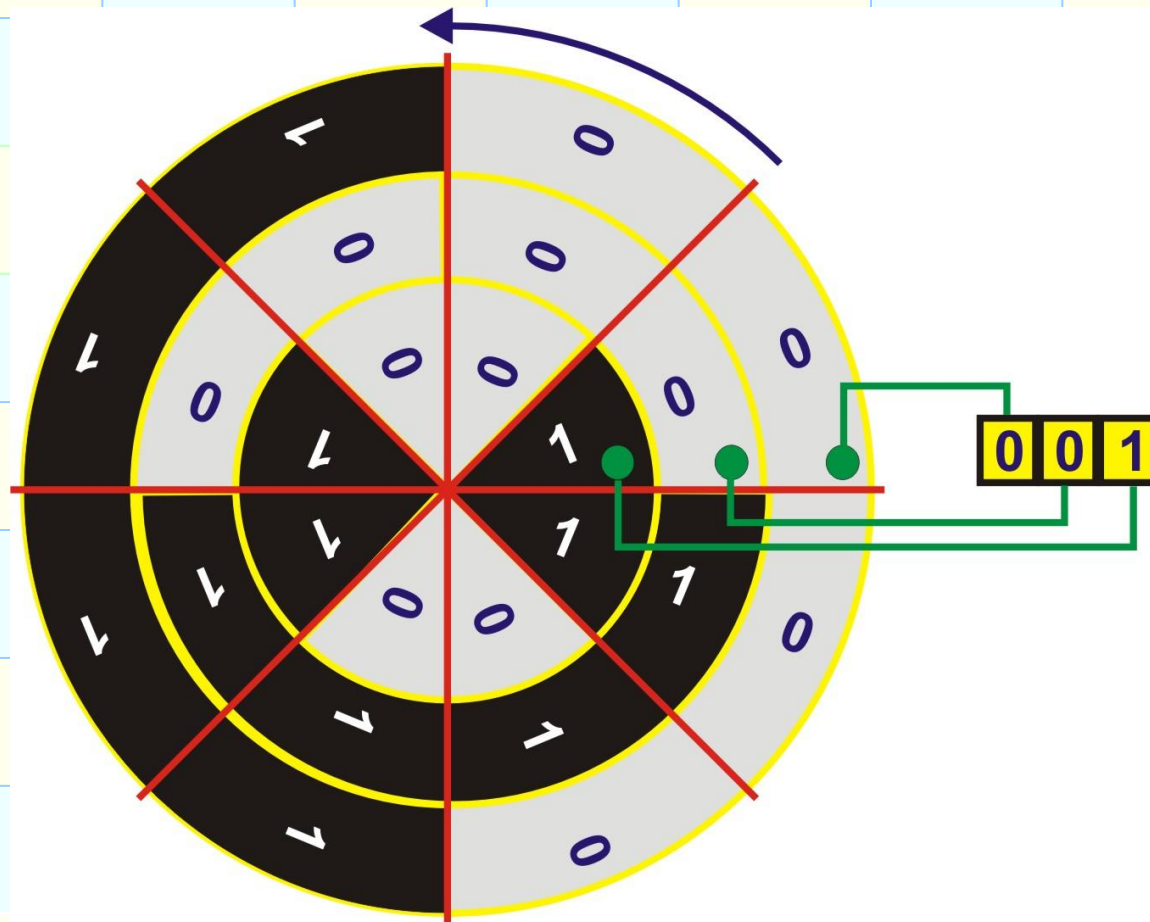
- El código GRAY es otro tipo de código basado en un sistema binario, pero de una construcción muy distinta a la de los demás códigos.
- Es un código continuo, cíclico y no ponderado.
- Su principal característica es que entre dos números consecutivos, solo cambia un solo bit.



	A	B	C	$N_{(10)}$
0	0	0	0	0
1	0	0	1	1
2	0	1	1	3
3	0	1	0	2
4	1	1	0	6
5	1	1	1	7
6	1	0	1	5
7	1	0	0	4

Código GRAY

Codificador de posición angular (encoder de tres bit's)



Código GRAY

Mapa de Karnaugh

F6

00

01

11

10

00

01

11

10

0

4

12

8

0000

0100

1100

1000

1

5

13

9

0001

0101

1101

1001

3

7

15

11

0011

0111

1111

1011

2

6

14

10

0010

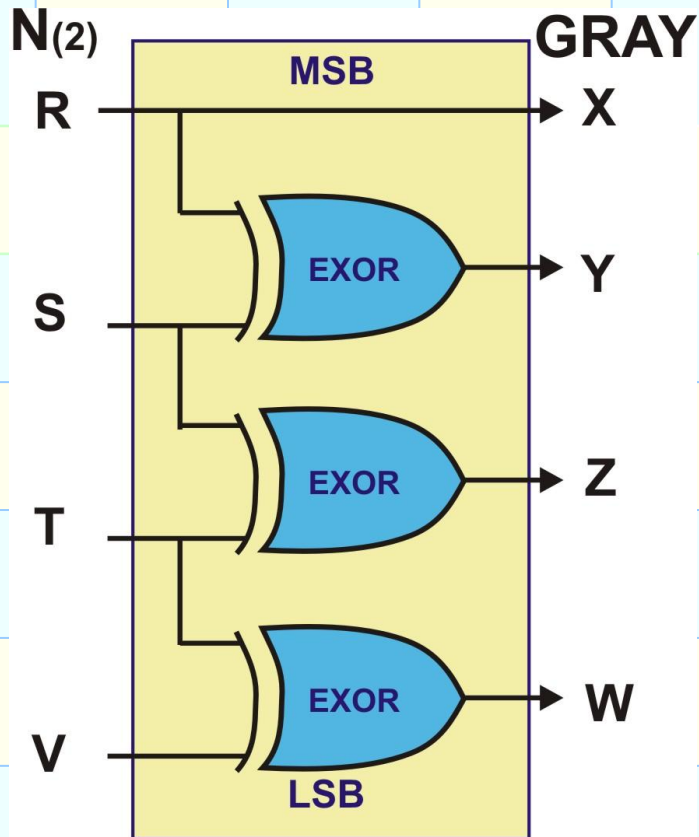
0110

1110

1010

Entre cuadros adyacentes
solo hay un cambio

Conversión de $N(2)$ a GRAY



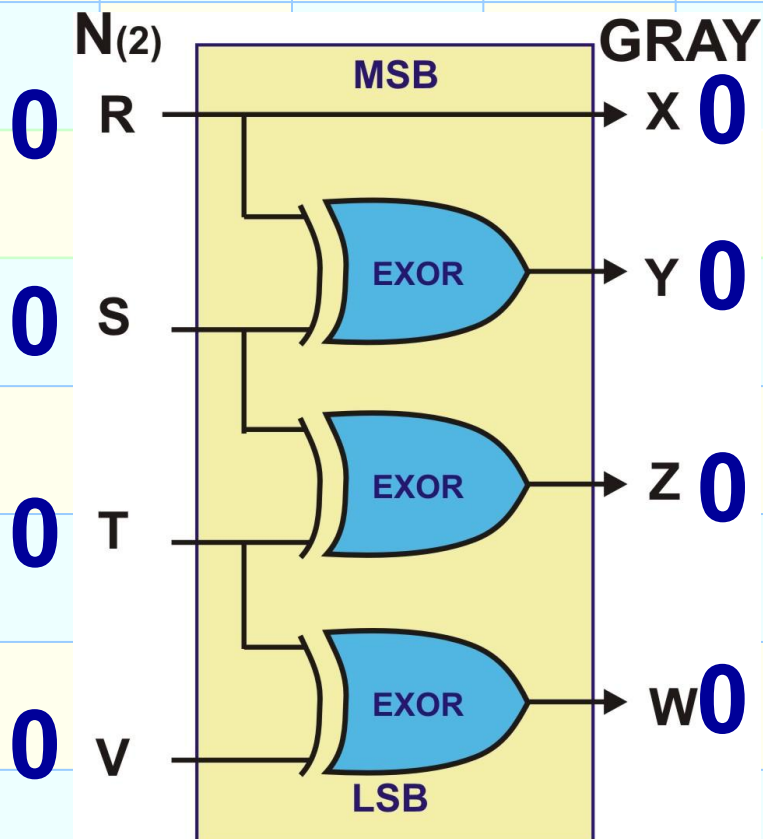
$$X = R$$

$$Y = R \oplus S$$

$$Z = S \oplus T$$

$$W = T \oplus V$$

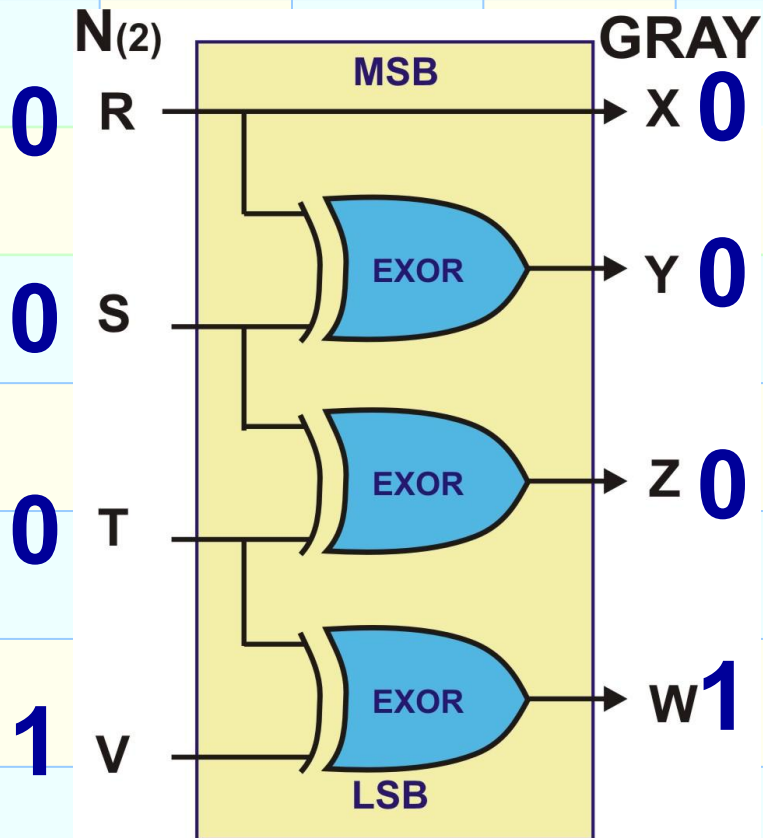
Conversión de $N(2)$ a GRAY



m	$N(2)$				GRAY			
	R	S	T	V	X	Y	Z	W
0	0	0	0	0	0	0	0	
1	0	0	0	1	0			
2	0	0	1	0				
3	0	0	1	1				
4	0	1	0	0				
5	0	1	0	1				
6	0	1	1	0				
7	0	1	1	1				
8	1	0	0	0				
9	1	0	0	1				
10	1	0	1	0				
11	1	0	1	1				
12	1	1	0	0				
13	1	1	0	1				
14	1	1	1	0				
15	1	1	1	1				

JAGG

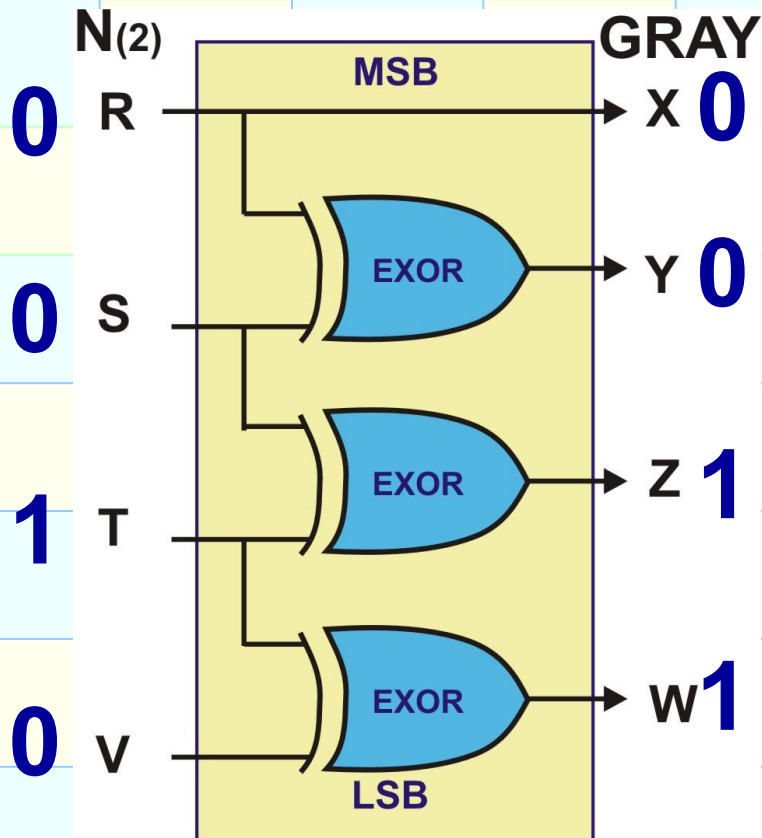
Conversión de $N(2)$ a GRAY



m	$N(2)$				GRAY			
	R	S	T	V	X	Y	Z	W
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0				
3	0	0	1	1				
4	0	1	0	0				
5	0	1	0	1				
6	0	1	1	0				
7	0	1	1	1				
8	1	0	0	0				
9	1	0	0	1				
10	1	0	1	0				
11	1	0	1	1				
12	1	1	0	0				
13	1	1	0	1				
14	1	1	1	0				
15	1	1	1	1				

JAGG

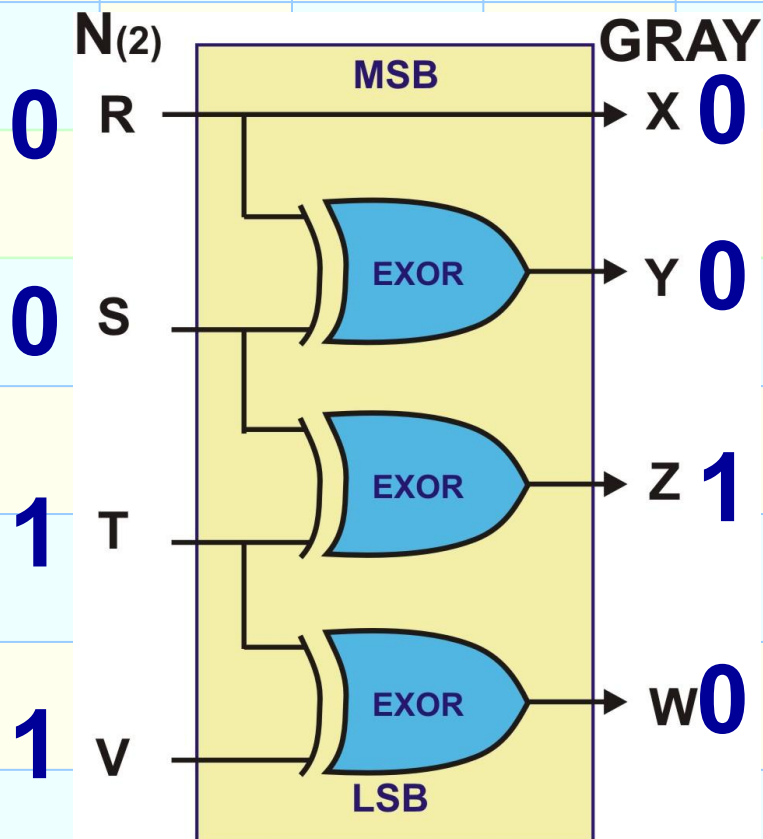
Conversión de $N(2)$ a GRAY



m	$N(2)$				GRAY			
	R	S	T	V	X	Y	Z	W
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1				
4	0	1	0	0				
5	0	1	0	1				
6	0	1	1	0				
7	0	1	1	1				
8	1	0	0	0				
9	1	0	0	1				
10	1	0	1	0				
11	1	0	1	1				
12	1	1	0	0				
13	1	1	0	1				
14	1	1	1	0				
15	1	1	1	1				

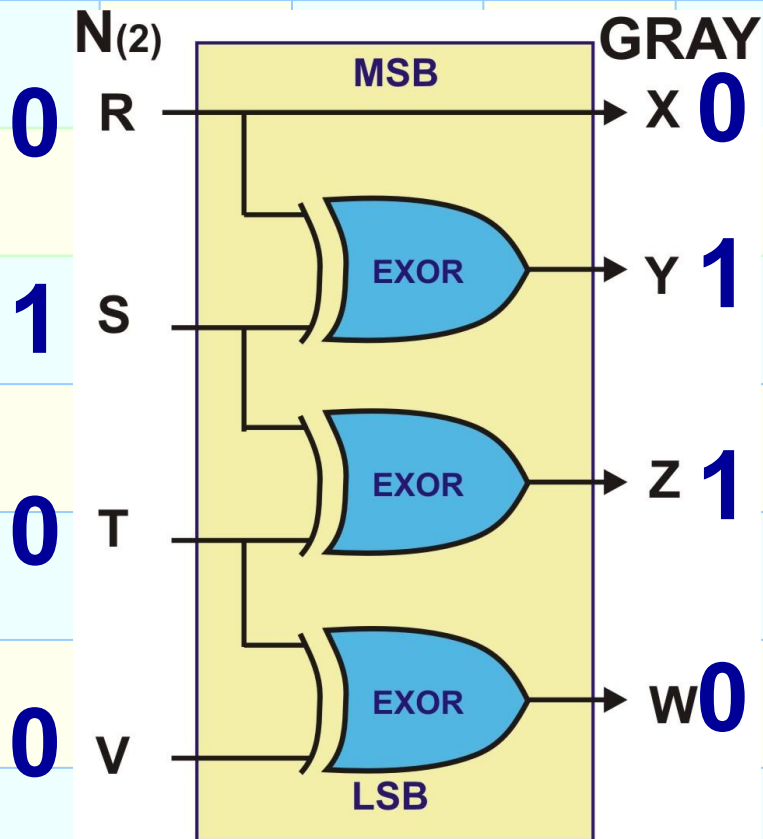
JAGG

Conversión de $N(2)$ a GRAY



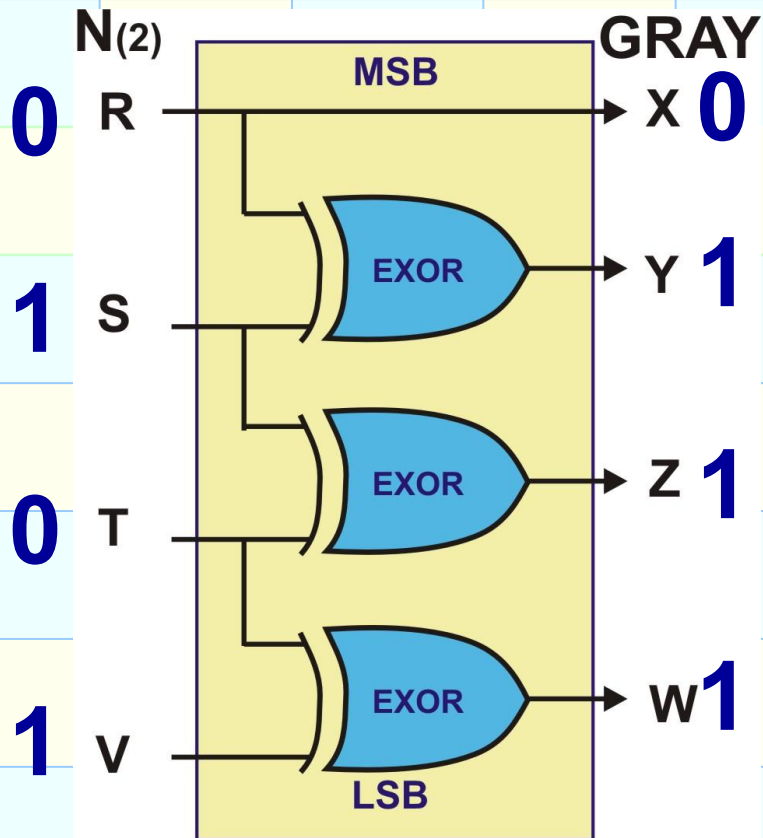
m	$N(2)$				GRAY			
	R	S	T	V	X	Y	Z	W
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0				
5	0	1	0	1				
6	0	1	1	0				
7	0	1	1	1				
8	1	0	0	0				
9	1	0	0	1				
10	1	0	1	0				
11	1	0	1	1				
12	1	1	0	0				
13	1	1	0	1				
14	1	1	1	0				
15	1	1	1	1				

Conversión de $N(2)$ a GRAY



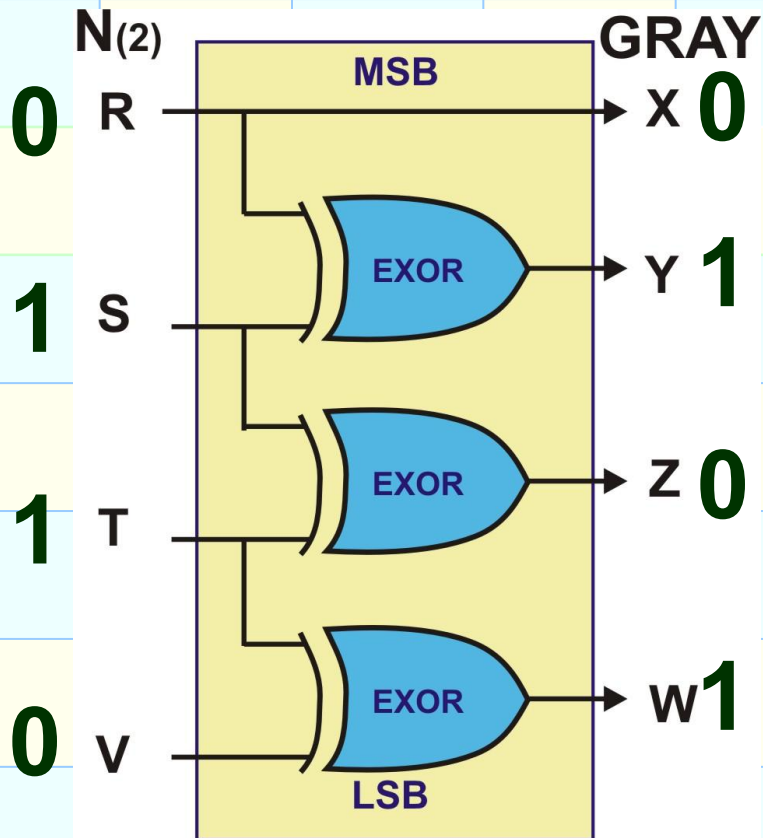
m	$N(2)$	GRAY			
	RSTV	X	Y	Z	W
0	0 0 0 0	0	0	0	0
1	0 0 0 1	0	0	0	1
2	0 0 1 0	0	0	1	1
3	0 0 1 1	0	0	1	0
4	0 1 0 0	0	1	1	0
5	0 1 0 1				
6	0 1 1 0				
7	0 1 1 1				
8	1 0 0 0				
9	1 0 0 1				
10	1 0 1 0				
11	1 0 1 1				
12	1 1 0 0				
13	1 1 0 1				
14	1 1 1 0				
15	1 1 1 1				

Conversión de $N(2)$ a GRAY



m	$N(2)$				GRAY			
	R	S	T	V	X	Y	Z	W
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0				
7	0	1	1	1				
8	1	0	0	0				
9	1	0	0	1				
10	1	0	1	0				
11	1	0	1	1				
12	1	1	0	0				
13	1	1	0	1				
14	1	1	1	0				
15	1	1	1	1				

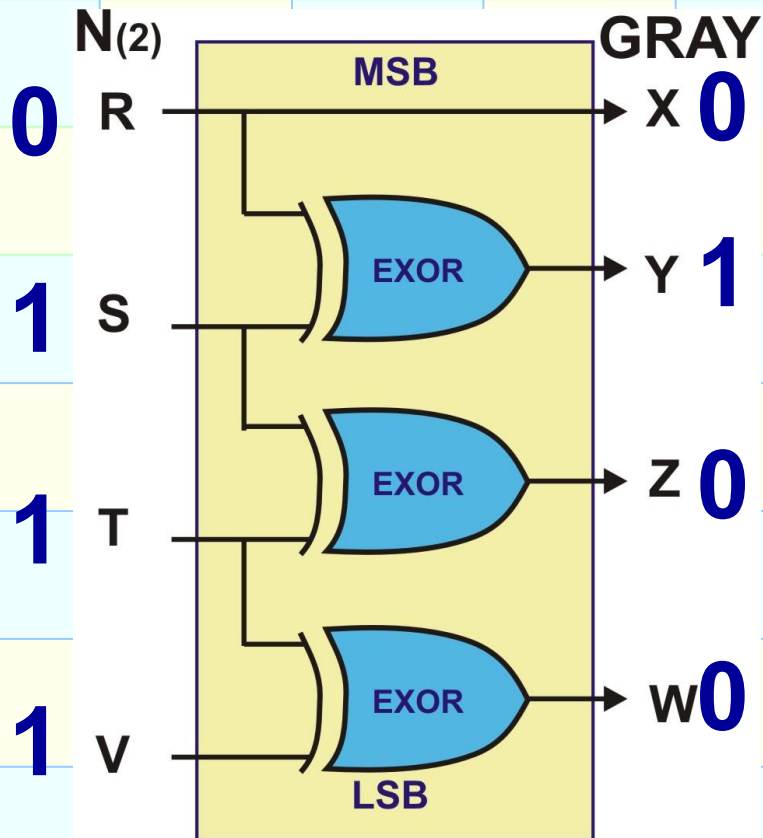
Conversión de $N(2)$ a GRAY



m	N ₍₂₎				GRAY			
	R	S	T	V	X	Y	Z	W
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1				
8	1	0	0	0				
9	1	0	0	1				
10	1	0	1	0				
11	1	0	1	1				
12	1	1	0	0				
13	1	1	0	1				
14	1	1	1	0				
15	1	1	1	1				

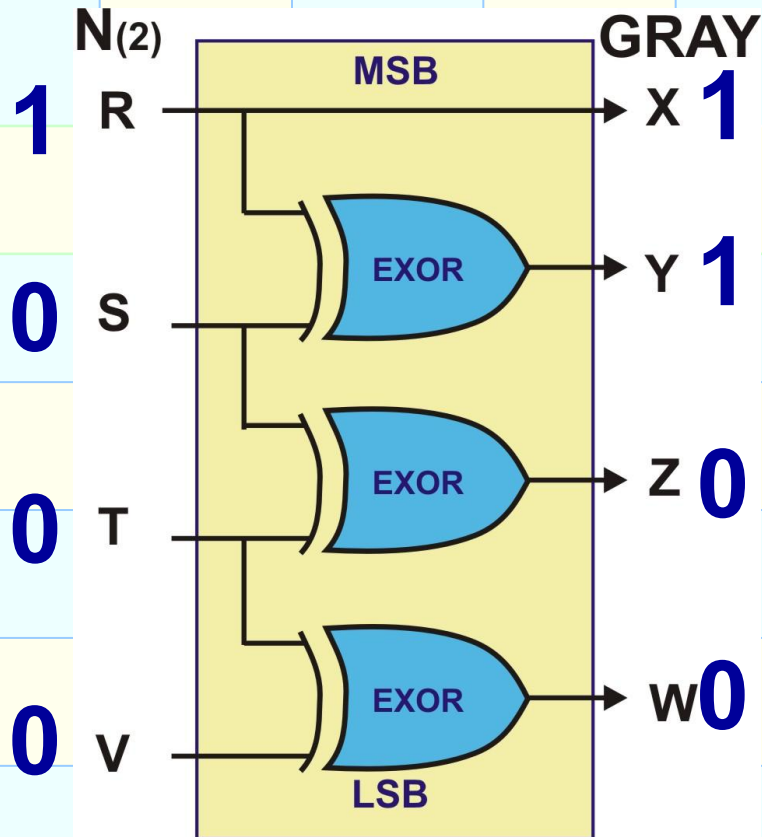
JAGG

Conversión de $N(2)$ a GRAY



m	$N(2)$				GRAY			
	R	S	T	V	X	Y	Z	W
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0				
9	1	0	0	1				
10	1	0	1	0				
11	1	0	1	1				
12	1	1	0	0				
13	1	1	0	1				
14	1	1	1	0				
15	1	1	1	1				

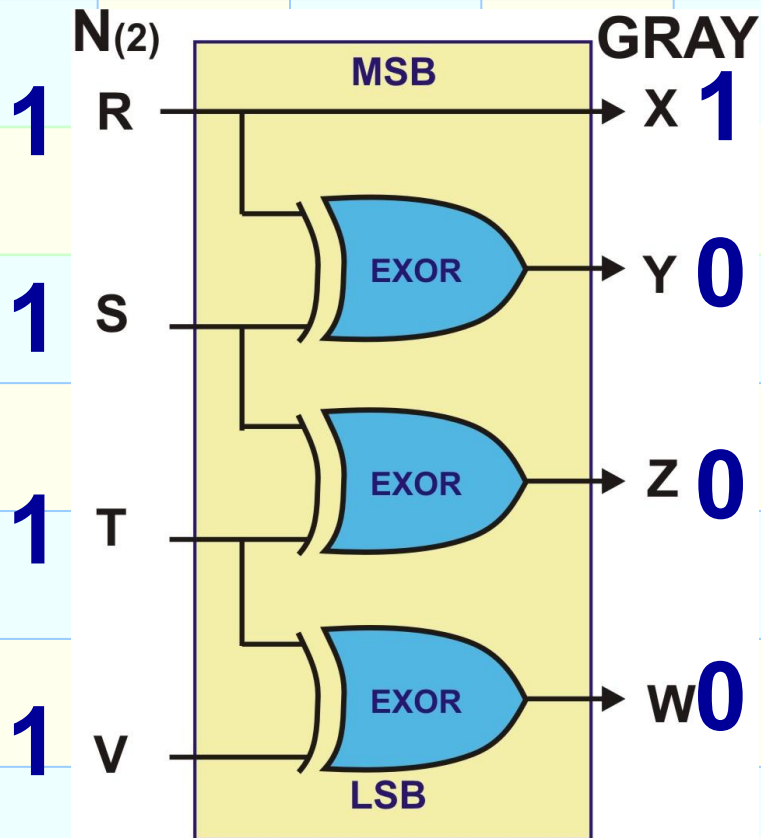
Conversión de $N(2)$ a GRAY



m	$N(2)$				GRAY			
	R	S	T	V	X	Y	Z	W
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1				
10	1	0	1	0				
11	1	0	1	1				
12	1	1	0	0				
13	1	1	0	1				
14	1	1	1	0				
15	1	1	1	1				

JAGG

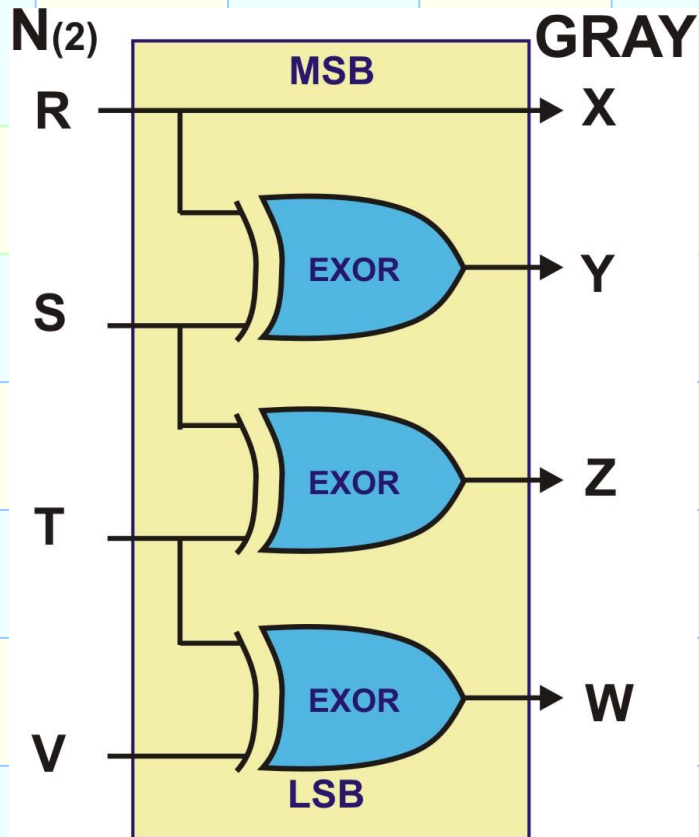
Conversión de $N(2)$ a GRAY



m	$N(2)$				GRAY			
	R	S	T	V	X	Y	Z	W
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1				
10	1	0	1	0				
11	1	0	1	1				
12	1	1	0	0				
13	1	1	0	1				
14	1	1	1	0				
15	1	1	1	1	1	0	0	0

JAGG

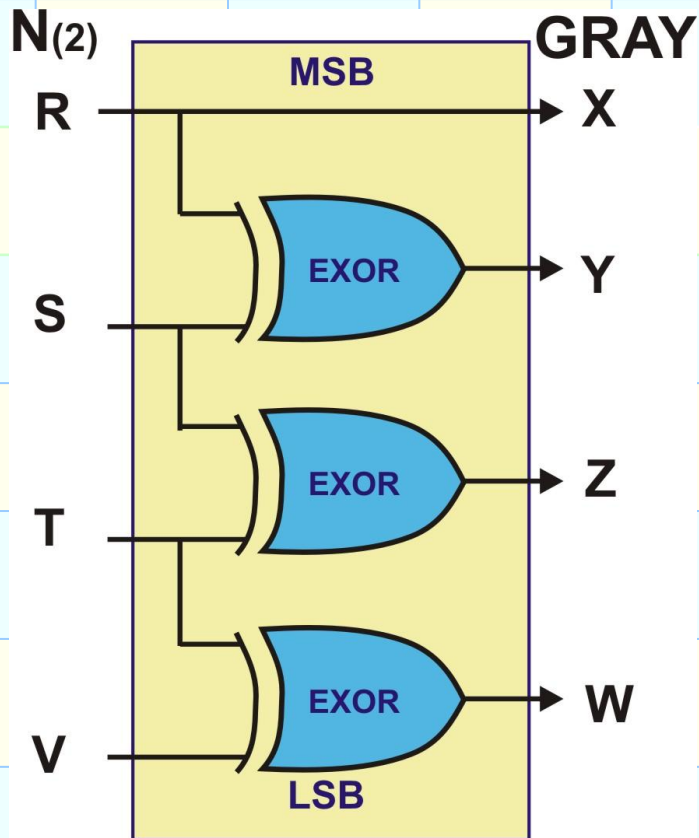
Conversión de $N(2)$ a GRAY



m	$N(2)$	GRAY			
	R S T V	X	Y	Z	W
0	0 0 0 0	0	0	0	0
1	0 0 0 1	0	0	0	1
2	0 0 1 0	0	0	1	1
3	0 0 1 1	0	0	1	0
4	0 1 0 0	0	1	1	0
5	0 1 0 1	0	1	1	1
6	0 1 1 0	0	1	0	1
7	0 1 1 1	0	1	0	0
8	1 0 0 0	1	1	0	0
9	1 0 0 1	1	1	0	1
10	1 0 1 0	1	1	1	1
11	1 0 1 1	1	1	1	0
12	1 1 0 0	1	0	1	0
13	1 1 0 1	1	0	1	1
14	1 1 1 0	1	0	0	1
15	1 1 1 1	1	0	0	0

JAGG

Conversión de $N(2)$ a GRAY



m	$N(2)$				GRAY				V(10)
	R	S	T	V	X	Y	Z	W	
0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	1
2	0	0	1	0	0	0	1	1	3
3	0	0	1	1	0	0	1	0	2
4	0	1	0	0	0	1	1	0	6
5	0	1	0	1	0	1	1	1	7
6	0	1	1	0	0	1	0	1	5
7	0	1	1	1	0	1	0	0	4
8	1	0	0	0	1	1	0	0	12
9	1	0	0	1	1	1	0	1	13
10	1	0	1	0	1	1	1	1	15
11	1	0	1	1	1	1	1	0	14
12	1	1	0	0	1	0	1	0	10
13	1	1	0	1	1	0	1	1	11
14	1	1	1	0	1	0	0	1	9
15	1	1	1	1	1	0	0	0	8

JAGG

Código Gray

0

0

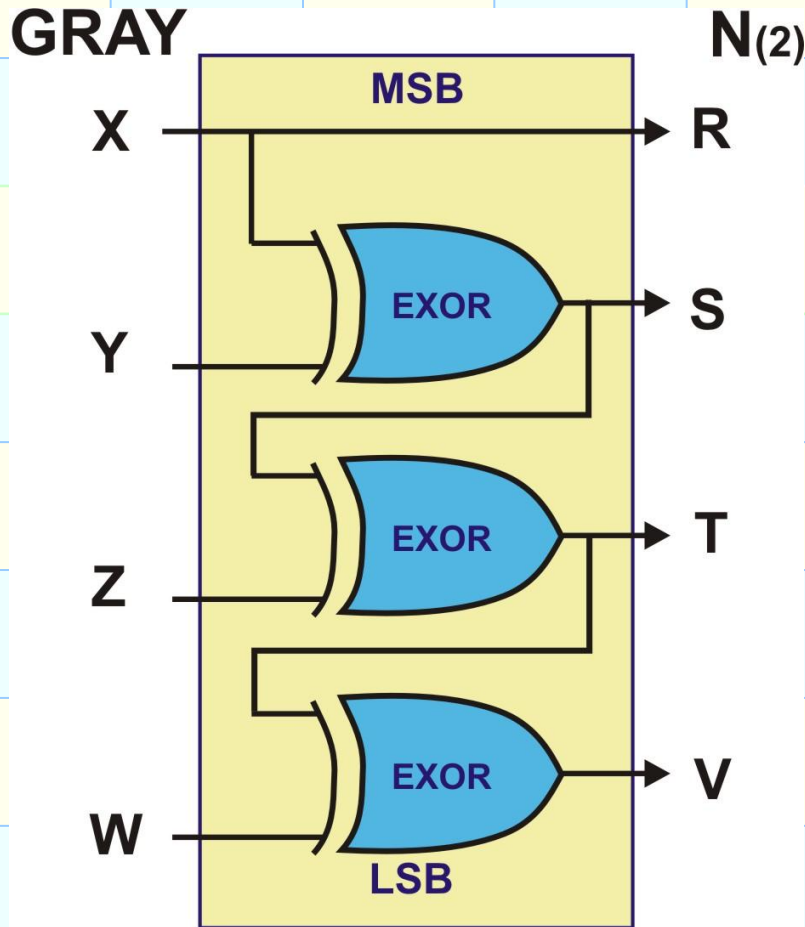
0

0



0

Conversión de GRAY a N(2)



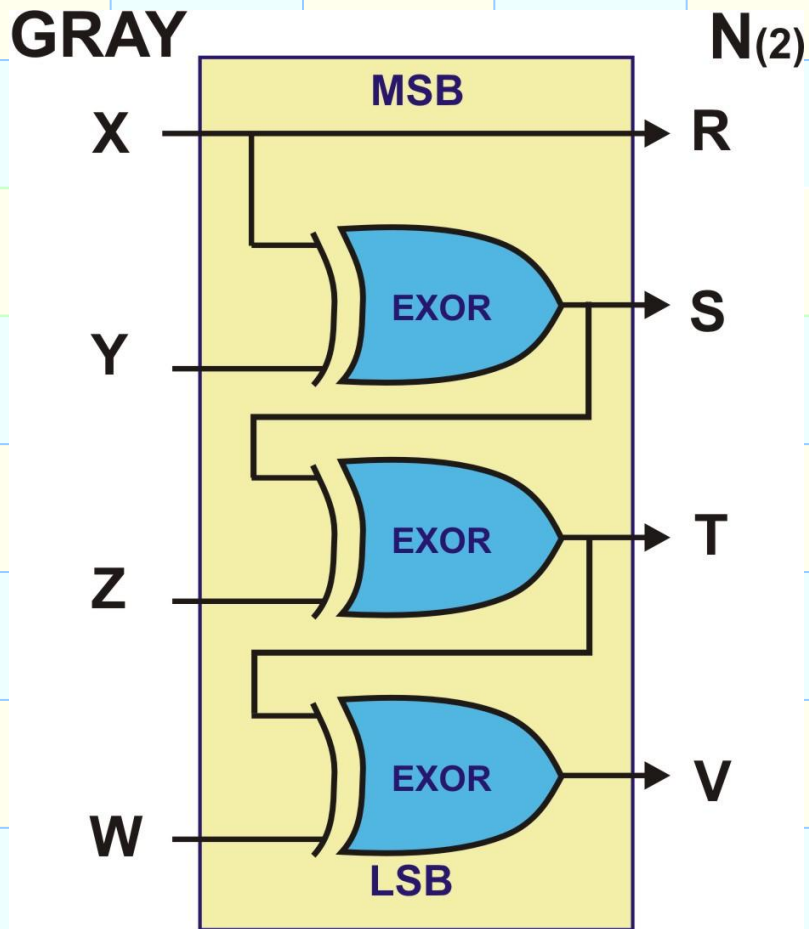
$$R = X$$

$$S = X \oplus Y$$

$$T = X \oplus Y \oplus Z = S \oplus Z$$

$$V = X \oplus Y \oplus Z \oplus W = T \oplus W$$

Conversión de GRAY a N(2)

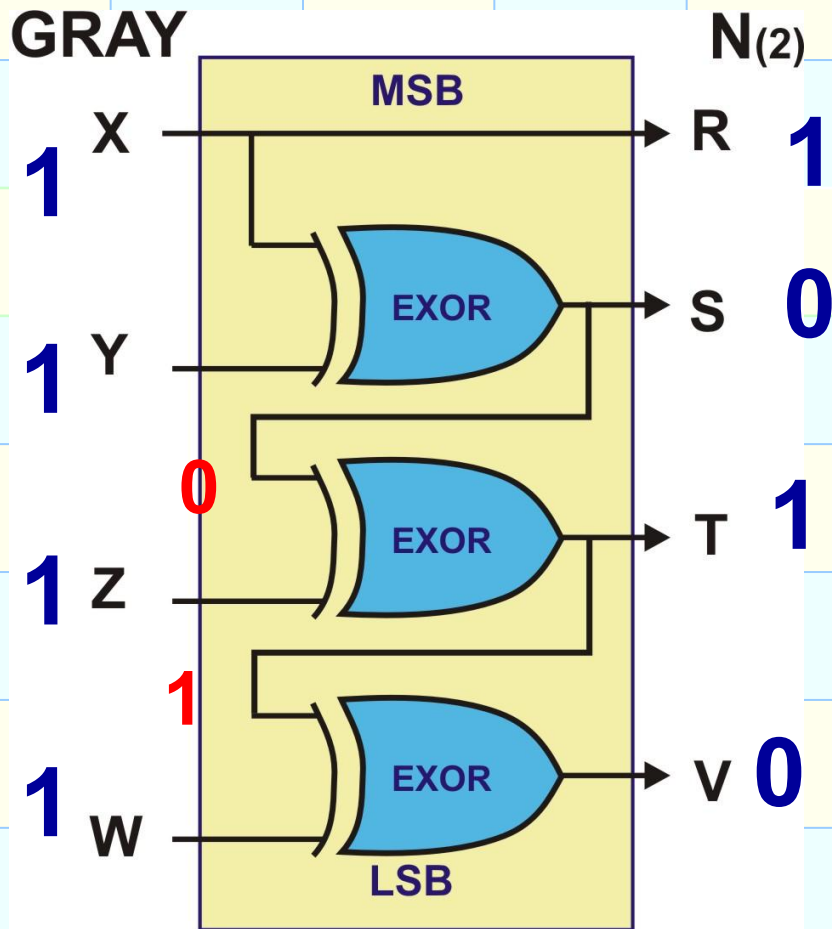


m	Gray	Binario			
	R S T V	X	Y	Z	W
0	0 0 0 0	0	0	0	0
1	0 0 0 1	0	0	0	1
2	0 0 1 0	0	0	1	1
3	0 0 1 1	0	0	1	0
4	0 1 0 0	0	1	1	1
5	0 1 0 1	0	1	1	0
6	0 1 1 0				
7	0 1 1 1				
8	1 0 0 0				
9	1 0 0 1				
10	1 0 1 0				
11	1 0 1 1				
12	1 1 0 0				
13	1 1 0 1				
14	1 1 1 0				
15	1 1 1 1				

JAGG



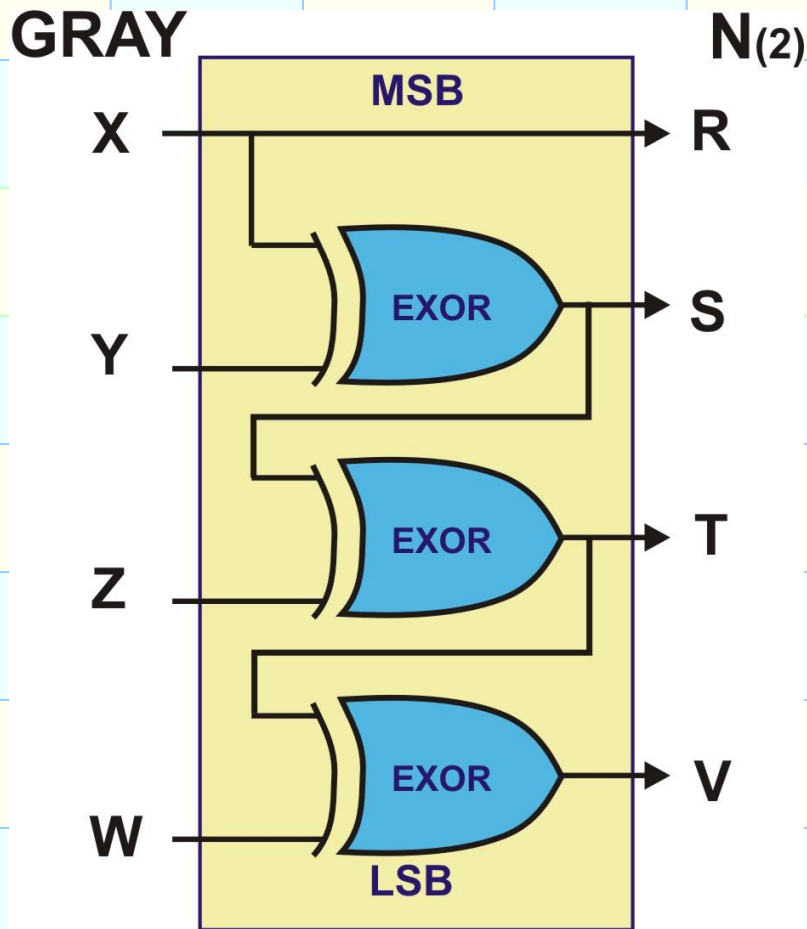
Conversión de GRAY a N(2)



	Gray	Binario			
m	RSTV	X	Y	Z	W
0	0 0 0 0	0	0	0	0
1	0 0 0 1	0	0	0	1
2	0 0 1 0	0	0	1	1
3	0 0 1 1	0	0	1	0
4	0 1 0 0	0	1	1	1
5	0 1 0 1	0	1	1	0
6	0 1 1 0				
7	0 1 1 1				
8	1 0 0 0				
9	1 0 0 1				
10	1 0 1 0				
11	1 0 1 1				
12	1 1 0 0				
13	1 1 0 1				
14	1 1 1 0				
15	1 1 1 1	1	0	1	0

JAGG

Conversión de GRAY a N(2)



	Gray	Binario			
m	R S T V	X	Y	Z	W
0	0 0 0 0	0	0	0	0
1	0 0 0 1	0	0	0	1
2	0 0 1 0	0	0	1	1
3	0 0 1 1	0	0	1	0
4	0 1 0 0	0	1	1	1
5	0 1 0 1	0	1	1	0
6	0 1 1 0	0	1	0	0
7	0 1 1 1	0	1	0	1
8	1 0 0 0	1	1	1	1
9	1 0 0 1	1	1	1	0
10	1 0 1 0	1	1	0	0
11	1 0 1 1	1	1	0	1
12	1 1 0 0	1	0	0	0
13	1 1 0 1	1	0	0	1
14	1 1 1 0	1	0	1	1
15	1 1 1 1	1	0	1	0

JAGG

El código Gray fue diseñado originalmente para prevenir señales ilegales (*señales falsas o viciadas en la representación*) de los switches electromecánicos, y actualmente es usado para facilitar la corrección de errores en los sistemas de comunicaciones, tales como algunos sistemas de televisión por cable y la televisión digital terrestre, además de usarse en situaciones en las cuales otros códigos, como el binario, podrían producir resultados erróneos o ambiguos durante esas transiciones en las cuales *Más de un bit del código cambia.*

Por ejemplo: Al aplicar el *código binario* y pasar de

Código Binario				
7	0	1	1	1
8	1	0	0	0

se requiere que los cuatro bits cambien simultáneamente.

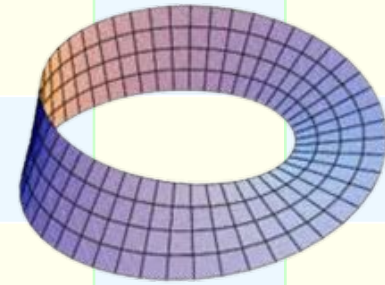
Según el dispositivo o circuito que genere los bits puede haber una diferencia significativa en los tiempos de transición de los diferentes bits.

Si es así, la transición de 0111 a 1000 podría producir uno o más estados intermedios.

En el :

Código Gray				
7	0	1	0	0
8	1	1	0	0

Código Johnson-Mobius

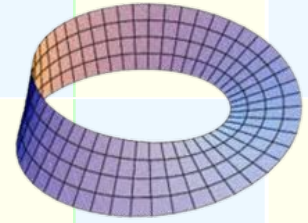


Es un código binario en el que sólo se permite **el cambio de un bit** al pasar de un número al siguiente, y los bits cambiados permanecen hasta completar un ciclo y el cambio se inicia en el mismo extremo de cada ciclo.

0

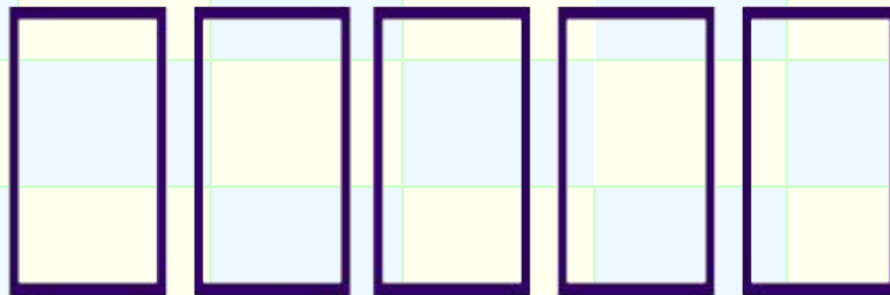


Código Johnson-Mobius



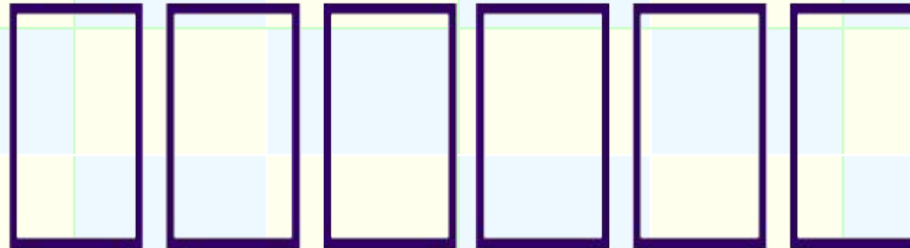
Es un código continuo y cíclico, los números de combinaciones que podemos obtener son $2n$, donde n es el número de bits del código.

0



Código JOHNSON

las combinaciones que podemos obtener son 2^n , donde n es el número de bits del código.



En este Código
Se utilizan solo 12 de las
64 combinaciones posibles de 6 bit's

Código contador de anillo

las combinaciones que podemos obtener son N , donde N es el número de bits del código.

S5

S4

S3

S2

S1

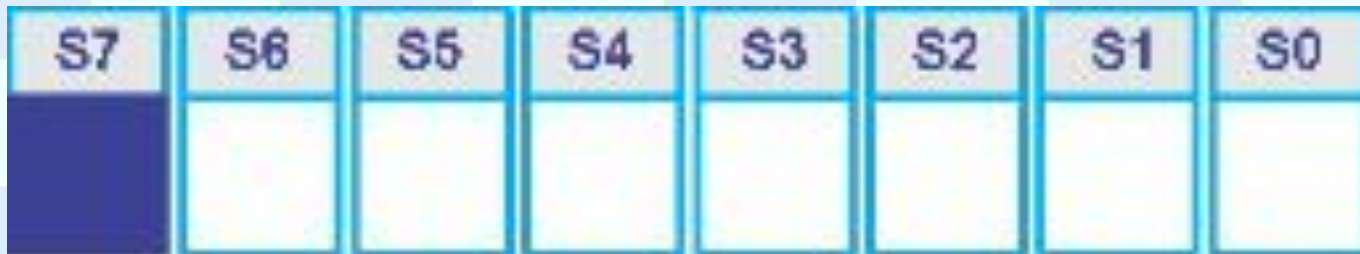
S0



En este Código
Se utilizan solo 6 de las
64 combinaciones posibles de 6 bit's

Código contador de anillo

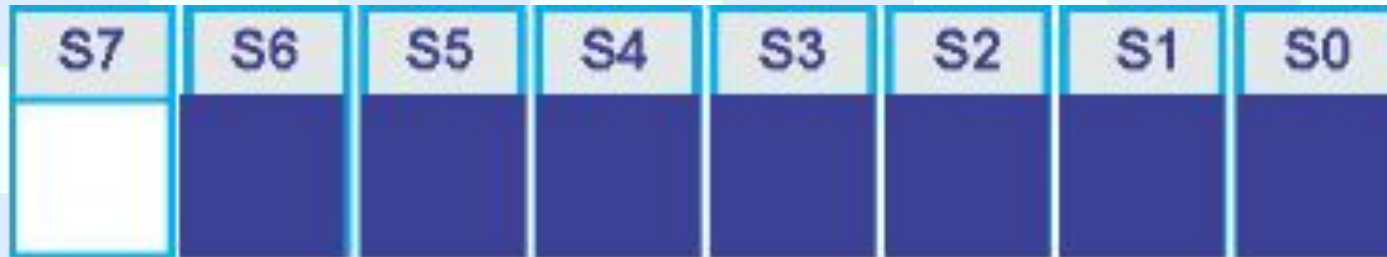
las combinaciones que podemos obtener son N , donde N es el número de bits del código.



En este Código
Se utilizan solo 8 de las
combinaciones posibles de 8 bit's

Código contador de anillo

las combinaciones que podemos obtener son N , donde N es el número de bits del código.



En este Código
Se utilizan solo 8 de las
combinaciones posibles de 8 bit's

Código Wan-Go



$$NC = [N(N+1)/2] + 1$$

Ejemplo para 5 bits

$$NC = [5(5+1)/2] + 1$$

$$NC = [5(6)/2] + 1$$

$$NC = [30/2] + 1$$

$$NC = [15] + 1 = 16$$

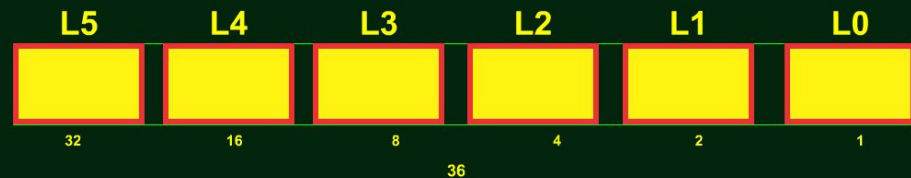
NC = 16 combinaciones (32)

Bits	Combinaciones	
1	2	2
2	4	4
3	7	8
4	11	16
5	16	32
6	22	64
7	29	128
8	37	256
9	46	512
10	56	1024
11	67	2048
12	79	4096
13	92	8192

Hecho en FIME

WanGo 6 bits

	5	4	3	2	1	0	V10	
	32	16	8	4	2	1		
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	1	1
3	0	0	0	0	1	0	2	1
4	0	0	0	1	0	0	4	1
5	0	0	1	0	0	0	8	1
6	0	1	0	0	0	0	16	1
7	1	0	0	0	0	0	32	1
8	0	0	0	0	1	1	3	2
9	0	0	0	1	1	0	6	2
10	0	0	1	1	0	0	12	2
11	0	1	1	0	0	0	24	2
12	1	1	0	0	0	0	48	2
13	0	0	0	1	1	1	7	3
14	0	0	1	1	1	0	14	3
15	0	1	1	1	0	0	28	3
16	1	1	1	0	0	0	56	3
17	0	0	1	1	1	1	15	4
18	0	1	1	1	1	0	30	4
19	1	1	1	1	0	0	60	4
20	0	1	1	1	1	1	31	5
21	1	1	1	1	1	0	62	5
22	1	1	1	1	1	1	63	6



$$NC = [N(N+1)/2] + 1$$

Ejemplo para 6 bits

$$NC = [6(6+1)/2] + 1$$

$$NC = [6(7)/2] + 1$$

$$NC = [42/2] + 1$$

$$NC = [21] + 1 = 22$$

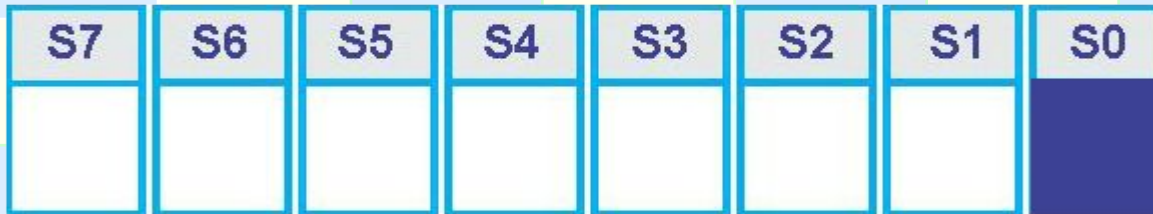
NC = 22 combinaciones (64)

**En este Código de 6 Bits
Se utilizan solo 22 de las
64 combinaciones posibles**

Hecho en FIME

Código Wan-Go

7 Bits



$$NC = [N(N+1)/2] + 1$$

Ejemplo para 7 bits

$$NC = [7(7+1)/2] + 1$$

$$NC = [7(8)/2] + 1$$

$$NC = [56/2] + 1$$

$$NC = [28] + 1 = 29$$

NC = 29 combinaciones (128)

En este Código de 7 bits Se utilizan solo 29 de las 128 combinaciones posibles

Hecho en FIME

	6	5	4	3	2	1	0	V10	
	64	32	16	8	4	2	1		
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	1	1	1
3	0	0	0	0	0	1	0	2	1
4	0	0	0	0	1	0	0	4	1
5	0	0	0	1	0	0	0	8	1
6	0	0	1	0	0	0	0	16	1
7	0	1	0	0	0	0	0	32	1
8	1	0	0	0	0	0	0	64	1
9	0	0	0	0	0	1	1	3	2
10	0	0	0	0	1	1	0	6	2
11	0	0	0	1	1	0	0	12	2
12	0	0	1	1	0	0	0	24	2
13	0	1	1	0	0	0	0	48	2
14	1	1	0	0	0	0	0	96	2
15	0	0	0	0	1	1	1	7	3
16	0	0	0	1	1	1	0	14	3
17	0	0	1	1	1	0	0	28	3
18	0	1	1	1	0	0	0	56	3
19	1	1	1	0	0	0	0	112	3
20	0	0	0	1	1	1	1	15	4
21	0	0	1	1	1	1	0	30	4
22	0	1	1	1	1	0	0	60	4
23	1	1	1	1	0	0	0	120	4
24	0	0	1	1	1	1	1	31	5
25	0	1	1	1	1	1	0	62	5
26	1	1	1	1	1	0	0	124	5
27	0	1	1	1	1	1	1	63	6
28	1	1	1	1	1	1	0	126	6
29	1	1	1	1	1	1	1	127	7

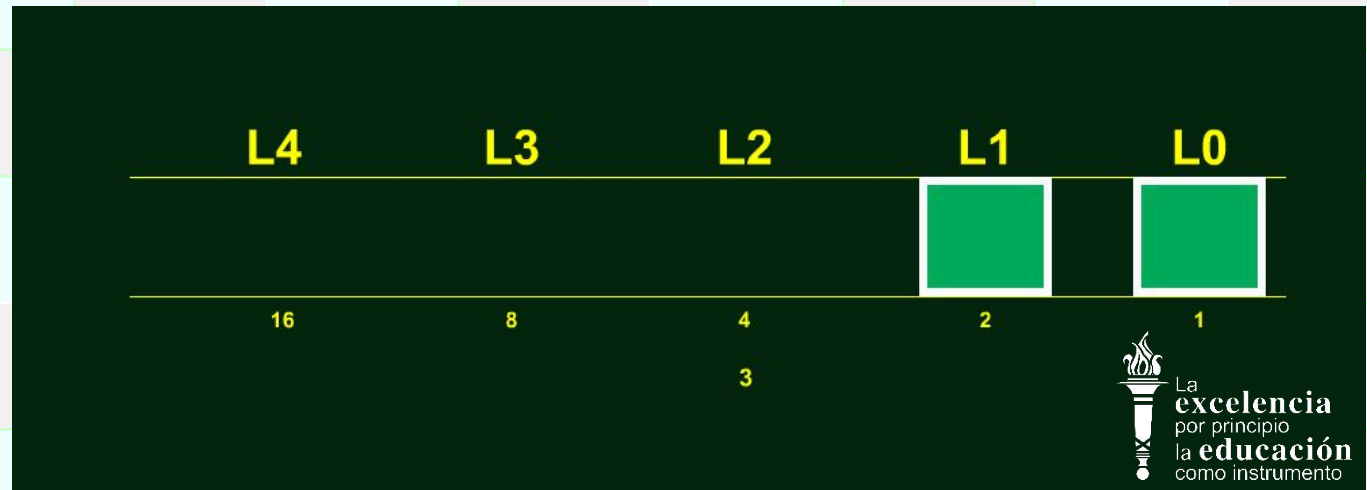
JAGG

Aplicaciones

- Probar displays LED en secuencia
- Simular un **barrido dinámico** con varios bits
- Pruebas de salidas digitales en prácticas con **microcontroladores o PLDs**
- Efectos visuales progresivos
- Rastreo progresivo por intensidad

Código 2 de 5

m	A	B	C	D	E
0	0	0	0	1	1
1	0	0	1	0	1
2	0	0	1	1	0
3	0	1	0	0	1
4	0	1	0	1	0
5	0	1	1	0	0
6	1	0	0	0	1
7	1	0	0	1	0
8	1	0	1	0	0
9	1	1	0	0	0



En este Código
Se utilizan solo 10 de las
32 posibles combinaciones con 5 bit's

Código 2 de 6

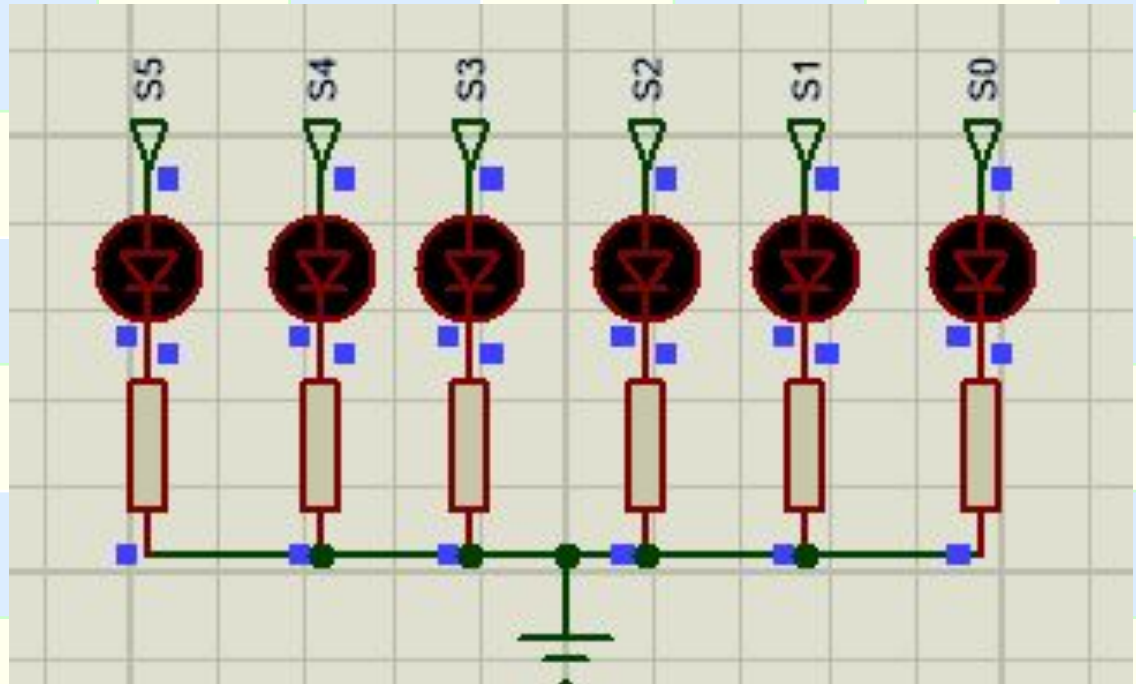
m	A	B	C	D	E	F
0	0	0	0	0	1	1
1	0	0	0	1	0	1
2	0	0	0	1	1	0
3	0	0	1	0	0	1
4	0	0	1	0	1	0
5	0	0	1	1	0	0
6	0	1	0	0	0	1
7	0	1	0	0	1	0
8	0	1	0	1	0	0
9	0	1	1	0	0	0
10	1	0	0	0	0	1
11	1	0	0	0	1	0
12	1	0	0	1	0	0
13	1	0	1	0	0	0
14	1	1	0	0	0	0



**En este Código
Se utilizan solo 15 de las
64 combinaciones posibles**

Código 3 de 6

	S5	S4	S3	S2	S1	S0	$v_{(10)}$
0	0	0	0	1	1	1	7
1	0	0	1	0	1	1	11
2	0	0	1	1	0	1	13
3	0	0	1	1	1	0	14
4	0	1	0	0	1	1	19
5	0	1	0	1	0	1	21
6	0	1	0	1	1	0	22
7	0	1	1	0	0	1	25
8	0	1	1	0	1	0	26
9	0	1	1	1	0	0	28
10	1	0	0	0	1	1	35
11	1	0	0	1	0	1	37
12	1	0	0	1	1	0	38
13	1	0	1	0	0	1	41
14	1	0	1	0	1	0	42
15	1	0	1	1	0	0	44
16	1	1	0	0	0	1	49
17	1	1	0	0	1	0	50
18	1	1	0	1	0	0	52
19	1	1	1	0	0	0	56



**En este Código
Se utilizan solo 20 de las
64 combinaciones posibles**

Código 2 de 7

t	M7	M6	M5	M4	M3	M2	M1	
0	0	0	0	0	0	1	1	3
1	0	0	0	0	1	0	1	5
2	0	0	0	0	1	1	0	6
3	0	0	0	1	0	0	1	9
4	0	0	0	1	0	1	0	10
5	0	0	0	1	1	0	0	12
6	0	0	1	0	0	0	1	17
7	0	0	1	0	0	1	0	18
8	0	0	1	0	1	0	0	20
9	0	0	1	1	0	0	0	24
10	0	1	0	0	0	0	1	33
11	0	1	0	0	0	1	0	34
12	0	1	0	0	1	0	0	36
13	0	1	0	1	0	0	0	40
14	0	1	1	0	0	0	0	48
15	1	0	0	0	0	0	1	65
16	1	0	0	0	0	1	0	66
17	1	0	0	0	1	0	0	68
18	1	0	0	1	0	0	0	72
19	1	0	1	0	0	0	0	80
20	1	1	0	0	0	0	0	96

**En este Código
Se utilizan solo 21 de las
256 combinaciones posibles con 6 Bits**

Código 5 de 7

t	M7	M6	M5	M4	M3	M2	M1	
0	0	0	1	1	1	1	1	31
1	0	1	0	1	1	1	1	47
2	0	1	1	0	1	1	1	55
3	0	1	1	1	0	1	1	59
4	0	1	1	1	1	0	1	61
5	0	1	1	1	1	1	0	62
6	1	0	0	1	1	1	1	79
7	1	0	1	0	1	1	1	87
8	1	0	1	1	0	1	1	91
9	1	0	1	1	1	0	1	93
10	1	0	1	1	1	1	0	94
11	1	1	0	0	1	1	1	103
12	1	1	0	1	0	1	1	107
13	1	1	0	1	1	0	1	109
14	1	1	0	1	1	1	0	110
15	1	1	1	0	0	1	1	115
16	1	1	1	0	1	0	1	117
17	1	1	1	0	1	1	0	118
18	1	1	1	1	0	0	1	121
19	1	1	1	1	0	1	0	122
20	1	1	1	1	1	0	0	124

**En este Código
Se utilizan solo 21 de las
256 combinaciones posibles con 6 Bits**

Código 6 de 8

	N7	N6	N5	N4	N3	N2	N1	N0	V10
1	0	0	1	1	1	1	1	1	63
2	0	1	0	1	1	1	1	1	95
3	0	1	1	0	1	1	1	1	111
4	0	1	1	1	0	1	1	1	119
5	0	1	1	1	1	0	1	1	123
6	0	1	1	1	1	1	0	1	125
7	0	1	1	1	1	1	1	0	126
8	1	0	0	1	1	1	1	1	159
9	1	0	1	0	1	1	1	1	175
10	1	0	1	1	0	1	1	1	183
11	1	0	1	1	1	0	1	1	187
12	1	0	1	1	1	1	0	1	189
13	1	0	1	1	1	1	1	0	190
14	1	1	0	0	1	1	1	1	207
15	1	1	0	1	0	1	1	1	215
16	1	1	0	1	1	0	1	1	219
17	1	1	0	1	1	1	0	1	221
18	1	1	0	1	1	1	1	0	222
19	1	1	1	0	0	1	1	1	231
20	1	1	1	0	1	0	1	1	235
21	1	1	1	0	1	1	0	1	237
22	1	1	1	0	1	1	1	0	238
23	1	1	1	1	0	0	1	1	243
24	1	1	1	1	0	1	0	1	245
25	1	1	1	1	0	1	1	0	246
26	1	1	1	1	1	0	0	1	249
27	1	1	1	1	1	0	1	0	250
28	1	1	1	1	1	1	0	0	252

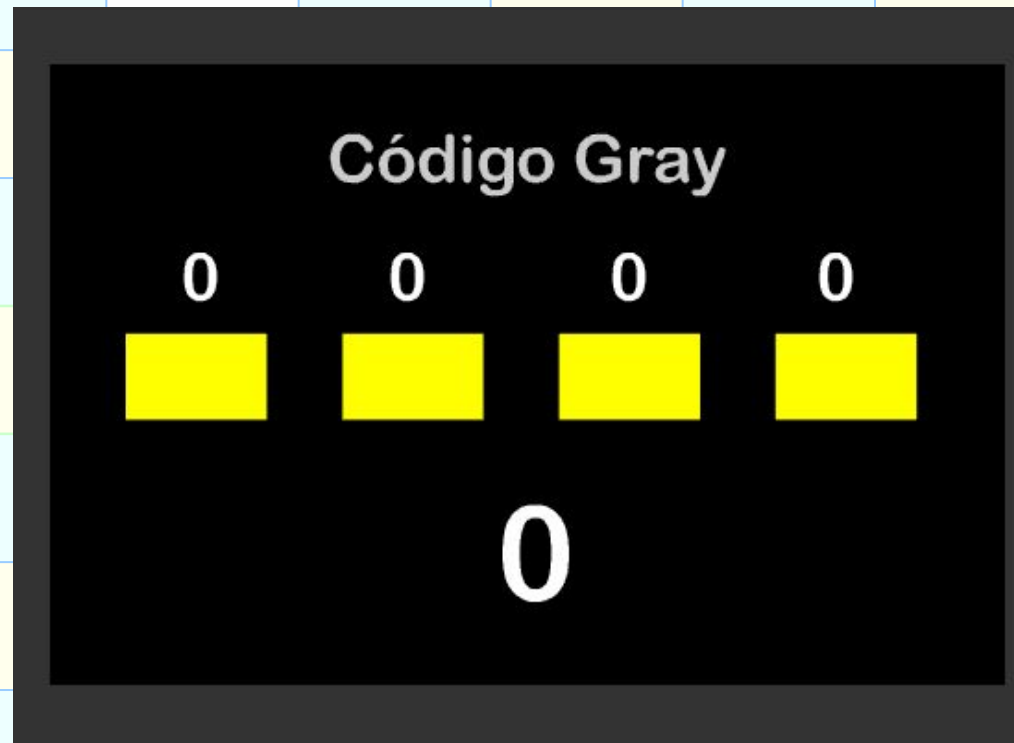
**En este Código
Se utilizan solo 28 de las
256 combinaciones posibles con 6 Bits**

Código 2 de 8

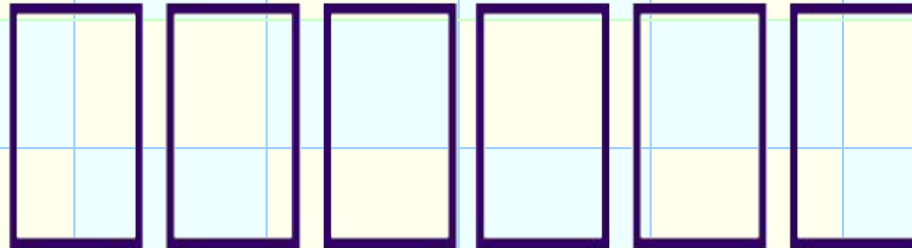
	N7	N6	N5	N4	N3	N2	N1	N0	V10
1	0	0	0	0	0	0	1	1	3
2	0	0	0	0	0	1	0	1	5
3	0	0	0	0	0	1	1	0	6
4	0	0	0	0	1	0	0	1	9
5	0	0	0	0	1	0	1	0	10
6	0	0	0	0	1	1	0	0	12
7	0	0	0	1	0	0	0	1	17
8	0	0	0	1	0	0	1	0	18
9	0	0	0	1	0	1	0	0	20
10	0	0	0	1	1	0	0	0	24
11	0	0	1	0	0	0	0	1	33
12	0	0	1	0	0	0	1	0	34
13	0	0	1	0	0	1	0	0	36
14	0	0	1	0	1	0	0	0	40
15	0	0	1	1	0	0	0	0	48
16	0	1	0	0	0	0	0	1	65
17	0	1	0	0	0	0	1	0	66
18	0	1	0	0	0	1	0	0	68
19	0	1	0	0	1	0	0	0	72
20	0	1	0	1	0	0	0	0	80
21	0	1	1	0	0	0	0	0	96
22	1	0	0	0	0	0	0	1	129
23	1	0	0	0	0	0	1	0	130
24	1	0	0	0	0	1	0	0	132
25	1	0	0	0	1	0	0	0	136
26	1	0	0	1	0	0	0	0	144
27	1	0	1	0	0	0	0	0	160
28	1	1	0	0	0	0	0	0	192

**En este Código
Se utilizan solo 28 de las
256 combinaciones posibles con 6 Bits**

Códigos Binarios

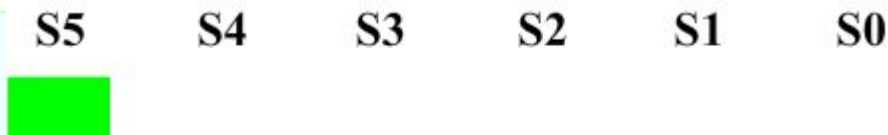


Código JOHNSON



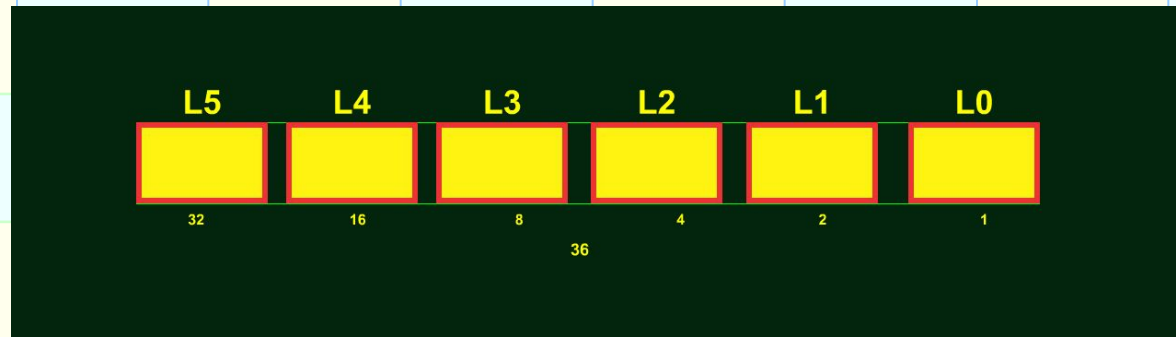
Código contador de anillo

las combinaciones que podemos obtener son 2^n , donde n es el número de bits del código.

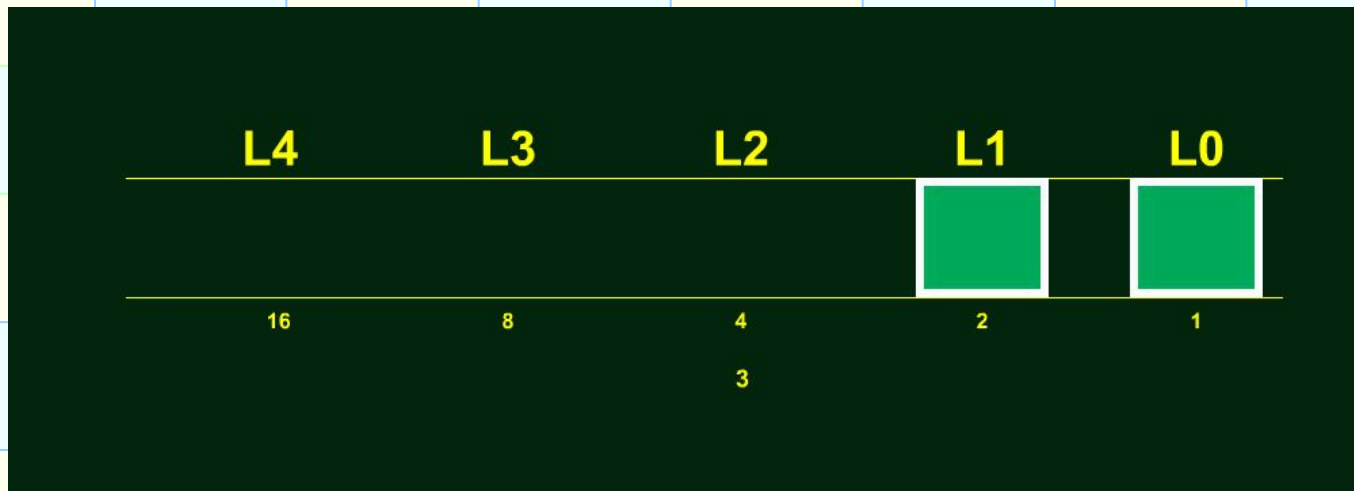


En este Código
Se utilizan solo 6 de las
64 combinaciones posibles de 6 bit's

Codigo Wan-Go



Código 2 de 5



Código 2 de 6



Códigos Decimales representados en binario

1	BCD
2	Exceso 3 o Stibitz
3	2421 o AIKEN
4	5211
5	8 4 -2 -1

Códigos numéricos decimales Expresados en Binario

Código BCD Decimal Expresado en Binario

Ponderado

$$m = A*8 + B*4 + C*2 + D*1$$

Donde m es el valor decimal

	8	4	2	1
m	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Códigos numéricos decimales

Código BCD Decimal Expresado en Binario

Combinaciones no usadas

10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

	8	4	2	1
m	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Códigos numéricos decimales

Código BCD

Decimal Expresado en Binario

825₍₁₀₎

1000 0010 0101_(BCD)

m	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Códigos numéricos decimales

**Exceso 3
o
EX3**

Ponderado

$$m = E*8 + F*4 + G*2 + H*1 - 3$$

Donde m es el valor decimal

m	E	F	G	H
0	0	0	1	1
1	0	1	0	0
2	0	1	0	1
3	0	1	1	0
4	0	1	1	1
5	1	0	0	0
6	1	0	0	1
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0

Códigos numéricos decimales

Exceso 3
o
EX3

m	E	F	G	H
0	0	0	1	1
1	0	1	0	0
2	0	1	0	1
3	0	1	1	0
4	0	1	1	1
5	1	0	0	0
6	1	0	0	1
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0

Códigos numéricos decimales

Exceso 3 o EX3

m	E	F	G	H
0	0	0	1	1
1	0	1	0	0
2	0	1	0	1
3	0	1	1	0
4	0	1	1	1
5	1	0	0	0
6	1	0	0	1
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0

Códigos numéricos decimales

Exceso 3
o
EX3

m	E	F	G	H
0	0	0	1	1
1	0	1	0	0
2	0	1	0	1
3	0	1	1	0
4	0	1	1	1
5	1	0	0	0
6	1	0	0	1
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0

Códigos numéricos decimales

Exceso 3
o
EX3

m	E	F	G	H
0	0	0	1	1
1	0	1	0	0
2	0	1	0	1
3	0	1	1	0
4	0	1	1	1
5	1	0	0	0
6	1	0	0	1
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0

Códigos numéricos decimales

Exceso 3 o EX3

m	E	F	G	H
0	0	0	1	1
1	0	1	0	0
2	0	1	0	1
3	0	1	1	0
4	0	1	1	1
5	1	0	0	0
6	1	0	0	1
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0

Códigos numéricos decimales

**Exceso 3
o
EX3**

Código reflejado

m	E	F	G	H
0	0	0	1	1
1	0	1	0	0
2	0	1	0	1
3	0	1	1	0
4	0	1	1	1
5	1	0	0	0
6	1	0	0	1
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0

Códigos numéricos decimales

**Exceso 3
o
EX3**

825₍₁₀₎

1011 0101 1000_(EX3)

1000 0010 0101_(BCD)

m	E	F	G	H
0	0	0	1	1
1	0	1	0	0
2	0	1	0	1
3	0	1	1	0
4	0	1	1	1
5	1	0	0	0
6	1	0	0	1
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0

Códigos numéricos decimales

Exceso 3 o EX3

Combinaciones no usadas

0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

m	E	F	G	H
0	0	0	1	1
1	0	1	0	0
2	0	1	0	1
3	0	1	1	0
4	0	1	1	1
5	1	0	0	0
6	1	0	0	1
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0

Códigos numéricos decimales

2421 o AIKEN

	2	4	2	1
m	I	J	K	L
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	0	1
8	1	1	1	0
9	1	1	1	1

JAGG

Códigos numéricos decimales

2421 o AIKEN

Ponderado

$$m = I*2 + J*4 + K*2 + L*1$$

Donde m es el valor decimal

	2	4	2	1
m	I	J	K	L
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	0	1
8	1	1	1	0
9	1	1	1	1

Códigos numéricos decimales

2421 o AIKEN

Ponderado

$$m = I*2 + J*4 + K*2 + L*1$$

Donde m es el valor decimal

	2	4	2	1
m	I	J	K	L
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	0	1
8	1	1	1	0
9	1	1	1	1

Códigos numéricos decimales

2421 o AIKEN

Ponderado

$$m = I*2 + J*4 + K*2 + L*1$$

Donde m es el valor decimal

	2	4	2	1
m	I	J	K	L
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	0	1
8	1	1	1	0
9	1	1	1	1

Códigos numéricos decimales

2421 o AIKEN

Ponderado

$$m = I*2 + J*4 + K*2 + L*1$$

Donde m es el valor decimal

	2	4	2	1
m	I	J	K	L
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	0	1
8	1	1	1	0
9	1	1	1	1

Códigos numéricos decimales

2421 o AIKEN

Ponderado

$$m = I*2 + J*4 + K*2 + L*1$$

Donde m es el valor decimal

	2	4	2	1
m	I	J	K	L
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	0	1
8	1	1	1	0
9	1	1	1	1

Códigos numéricos decimales

2421 o AIKEN

Ponderado

$$m = I*2 + J*4 + K*2 + L*1$$

Donde m es el valor decimal

	2	4	2	1
m	I	J	K	L
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	0	1
8	1	1	1	0
9	1	1	1	1

Códigos numéricos decimales

2421 o AIKEN

Ponderado

$$m = I*2 + J*4 + K*2 + L*1$$

Donde m es el valor decimal

	2	4	2	1
m	I	J	K	L
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	0	1
8	1	1	1	0
9	1	1	1	1

Códigos numéricos decimales

2421 o AIKEN

Combinaciones no usadas

V ₍₁₀₎	I	J	K	L
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0

m	I	J	K	L
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	0	1
8	1	1	1	0
9	1	1	1	1

Códigos numéricos decimales

2421 o AIKEN

825₍₁₀₎

1110 0010 1011₍₂₄₂₁₎

1000 0010 0101_(BCD)

1011 0101 1000_(EX3)

m	I	J	K	L
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	0	1
8	1	1	1	0
9	1	1	1	1

Códigos numéricos decimales

5211

Ponderado

$$m = M*5 + N*2 + O*1 + P*1$$

	5	2	1	1
m	M	N	O	P
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	1	0	1
4	0	1	1	1
5	1	0	0	0
6	1	0	1	0
7	1	1	0	0
8	1	1	1	0
9	1	1	1	1

Códigos numéricos decimales

5211

Ponderado

$$m = M*5 + N*2 + O*1 + P*1$$

	5	2	1	1
m	M	N	O	P
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	1	0	1
4	0	1	1	1
5	1	0	0	0
6	1	0	1	0
7	1	1	0	0
8	1	1	1	0
9	1	1	1	1

Códigos numéricos decimales

5211

Combinaciones no usadas

V ₍₁₀₎	M	N	O	P
2	0	0	1	0
4	0	1	0	0
6	0	1	1	0
9	1	0	0	1
11	1	0	1	1
13	1	1	0	1

m	M	N	O	P
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	1	0	1
4	0	1	1	1
5	1	0	0	0
6	1	0	1	0
7	1	1	0	0
8	1	1	1	0
9	1	1	1	1

Códigos numéricos decimales

5211

825₍₁₀₎

1110 0011 1000₍₅₂₁₁₎

1000 0010 0101_(BCD)

1011 0101 1000_(EX3)

1110 0010 1011₍₂₄₂₁₎

m	M	N	O	P
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	1	0	1
4	0	1	1	1
5	1	0	0	0
6	1	0	1	0
7	1	1	0	0
8	1	1	1	0
9	1	1	1	1

JAGG

Códigos numéricos decimales

8 4 -2 -1

Reflejado

	8	4	-2	-1
m	Q	R	S	T
0	0	0	0	0
1	0	1	1	
2	0	1	1	
3	0	1	0	
4	0	1	0	
5	1	0	1	
6	1	0	1	
7	1	0	0	
8	1	0	0	
9	1	1	1	

Códigos numéricos decimales

8 4 -2 -1

Ponderado

$$m = Q*8 + R*4 + S*(-2) + T*(-1)$$

	8	4	-2	-1
m	Q	R	S	T
0	0	0	0	0
1	0	1	1	
2	0	1	1	
3	0	1	0	
4	0	1	0	
5	1	0	1	
6	1	0	1	
7	1	0	0	
8	1	0	0	
9	1	1	1	

Códigos numéricos decimales

8 4 -2 -1

825₍₁₀₎

1000 0110 1011_(8 4-2-1)

m	Q	R	S	T
0	0	0	0	0
1	0	1	1	
2	1	1	1	
3	0	1	0	
4	0	1	0	
5	1	0	1	
6	1	0	1	
7	0	0	0	
8	1	0	0	
9	0	1	1	

JAGG

Códigos numéricos decimales

8 4 -2 -1

Combinaciones no usadas

V ₍₁₀₎	Q	R	S	T
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0

m	Q	R	S	T
0	0	0	0	0
1	0	1	1	1
2	0	1	1	0
3	0	1	0	1
4	0	1	0	0
5	1	0	1	1
6	1	0	1	0
7	1	0	0	1
8	1	0	0	0
9	1	1	1	1

Códigos numéricos decimales

8 4 -2 -1

825₍₁₀₎

1000 0110 1011₍₈₄₋₂₋₁₎

1110 0011 1000₍₅₂₁₁₎

1000 0010 0101_(BCD)

1011 0101 1000_(EX3)

1110 0010 1011₍₂₄₂₁₎

m	Q	R	S	T
0	0	0	0	0
1	0	1	1	1
2	0	1	1	0
3	0	1	0	1
4	0	1	0	0
5	1	0	1	1
6	1	0	1	0
7	1	0	0	1
8	1	0	0	0
9	1	1	1	1

JAGG

Códigos Decimales expresados en binario

BCD Decimal Expresado en Binario

EX3 Exceso 3 Stibitz

2421 AIKEN

5211

8 4 -2 -1

825⁽¹⁰⁾
1110 0011 1000 (5211)
1000 0010 0101 (BCD)
1011 0101 1000 (EX3)
1110 0010 1011 (2421)
1000 0110 1011 (84-2-1)

Todos ellos representan dígitos decimales expresados en binario y para cada uno de los dígitos se utilizan 4 bits y la diferencia entre ellos es que usan diferentes combinaciones.

	BCD	Exceso 3	2421	5211	8 4 -2 -1
m	A B C D	E F G H	I J K L	M N O P	W X Y Z
0	0 0 0 0	0 0 1 1	0 0 0 0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 1 0 0	0 0 0 1	0 0 0 1	0 1 1 1
2	0 0 1 0	0 1 0 1	0 0 1 0	0 0 1 1	0 1 1 0
3	0 0 1 1	0 1 1 0	0 0 1 1	0 1 0 1	0 1 0 1
4	0 1 0 0	0 1 1 1	0 1 0 0	0 1 1 1	0 1 0 0
5	0 1 0 1	1 0 0 0	1 0 1 1	1 0 0 0	1 0 1 1
6	0 1 1 0	1 0 0 1	1 1 0 0	1 0 1 0	1 0 1 0
7	0 1 1 1	1 0 1 0	1 1 0 1	1 1 0 0	1 0 0 1
8	1 0 0 0	1 0 1 1	1 1 1 0	1 1 1 0	1 0 0 0
9	1 0 0 1	1 1 0 0	1 1 1 1	1 1 1 1	1 1 1 1

	BCD		Exceso 3		2421		5211		8,4,-2,-1
m	A B C D		E F G H		I J K L		M N O P		Q R S T
0	0 0 0 0		0 0 1 1		0 0 0 0		0 0 0 0		0 0 0 0
1	0 0 0 1		0 1 0 0		0 0 0 1		0 0 0 1		0 1 1 1
2	0 0 1 0		0 1 0 1		0 0 1 0		0 0 1 1		0 1 1 0
3	0 0 1 1		0 1 1 0		0 0 1 1		0 1 0 1		0 1 0 1
4	0 1 0 0		0 1 1 1		0 1 0 0		0 1 1 1		0 1 0 0
5	0 1 0 1		1 0 0 0		1 0 1 1		1 0 0 0		1 0 1 1
6	0 1 1 0		1 0 0 1		1 1 0 0		1 0 1 0		1 0 1 0
7	0 1 1 1		1 0 1 0		1 1 0 1		1 1 0 0		1 0 0 1
8	1 0 0 0		1 0 1 1		1 1 1 0		1 1 1 0		1 0 0 0
9	1 0 0 1		1 1 0 0		1 1 1 1		1 1 1 1		1 1 1 1
N o u s a d as	1 0 1 0		0 0 0 0		0 1 0 1		0 0 1 0		0 0 0 1
	1 0 1 1		0 0 0 1		0 1 1 0		0 1 0 0		0 0 1 0
	1 1 0 0		0 0 1 0		0 1 1 1		0 1 1 0		0 0 1 1
	1 1 0 1		1 1 0 1		1 0 0 0		1 0 0 1		1 1 0 0
	1 1 1 0		1 1 1 0		1 0 0 1		1 0 1 1		1 1 0 1
	1 1 1 1		1 1 1 1		1 0 1 0		1 1 0 1		1 1 1 0

Códigos decimales expresados en binario

	BCD	Exceso 3	2421	5211	8 4 -2 -1
m	A B C D	E F G H	I J K L	M N O P	W X Y Z
0	0 0 0 0	0 0 1 1	0 0 0 0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 1 0 0	0 0 0 1	0 0 0 1	0 1 1 1
2	0 0 1 0	0 1 0 1	0 0 1 0	0 0 1 1	0 1 1 0
3	0 0 1 1	0 1 1 0	0 0 1 1	0 1 0 1	0 1 0 1
4	0 1 0 0	0 1 1 1	0 1 0 0	0 1 1 1	0 1 0 0
5	0 1 0 1	1 0 0 0	1 0 1 1	1 0 0 0	1 0 1 1
6	0 1 1 0	1 0 0 1	1 1 0 0	1 0 1 0	1 0 1 0
7	0 1 1 1	1 0 1 0	1 1 0 1	1 1 0 0	1 0 0 1
8	1 0 0 0	1 0 1 1	1 1 1 0	1 1 1 0	1 0 0 0
9	1 0 0 1	1 1 0 0	1 1 1 1	1 1 1 1	1 1 1 1

¿Cuál código te gusto más ?

Binarios $N_{(2)}$	Decimales expresados en Binario
Gray	BCD
Jonhson	Exceso 3 o Stibitz
Anillo	2421 o AIKEN
Wan-GO	5211
2 de 5	8 4 -2 -1
2 de 6	
3 de 6	
2 de 7	
6 de 7	
5 de 7	
2 de 8	
6 de 8	

Con que se quedan del tema de Diseño de sistemas combinacionales

**1.- Método de Diseño, HDL,
PF7 Equations, Truth_Table**

**2.- Multiplexor
PF8 When, Then , Else**

**3.- Sistemas Combinacionales que no están completamente especificados
Don't' Care, Cant Happen**

**4.- Codificadores y Decodificadores
AF3 BCD a 7 segmentos**

**5.- Unidades Lógicas Aritméticas
Sumador SN74283 y Comparador SN7485**

**6.- Códigos Binarios
Binarios y Decimales**

