

SISTEMA DE NUMERACIÓN

Consiste en un conjunto ordenado de símbolos llamados dígitos. En este conjunto se definen reglas para efectuar operaciones matemáticas, la base de un sistema indica la cantidad de dígitos incluidos en dicho conjunto ordenado. En un sistema de numeración se pueden representar números que estén integrados por una parte entera y una fraccional.

$$\sum_{i=-m}^{n-1} a_i r^i$$

r base del sistema numérico
ai i-esimo dígito
m cantidad de dígitos parte fraccionaria
n cantidad de dígitos parte entera

$$a_{-m}r^{-m} + a_{-m+1}r^{-m+1} + \dots + a_{-1}r^{-1} + a_0r^0 + a_1r^1 + \dots + a_{n-2}r^{n-2} + a_{n-1}r^{n-1}$$

su representación es de la siguiente forma

$$(a_{n-1}a_{n-2} \dots a_1a_0.a_{-1}a_{-2} \dots a_{-m+1}a_{-m}r^{-m})_r$$

para $r \leq 10$

$$a_i = \{0, 1, 2, 3, \dots, r-1\}$$

para $r > 10$

$$a_i = \{0, 1, 2, 3, \dots, 9, A, B, C, D, E, F, \dots, r-1\}$$

CONVERSIONES

Dado un numero de base 10 convertirlo a base r

“divisiones sucesivas”

(238.12)₁₀

Primer residuo es el menos significativo

4 7

$$\begin{array}{r} 5 \overline{) 238} \\ \underline{38} \\ 3 \end{array}$$

$$\begin{array}{r} 9 \\ 5 \overline{) 47} \\ \underline{2} \end{array}$$

$$\begin{array}{r} 1 \\ 5 \overline{) 9} \\ \underline{4} \end{array}$$

Ultimo dígito es el más significativo

$$\begin{array}{r} 0 \\ 5 \overline{) 1} \\ \underline{1} \end{array}$$

$$(0.12)_{10} \rightarrow \begin{array}{rcl} 0.12 & \times & 5 \\ 0.6 & \times & 5 \end{array} \begin{array}{l} 0.6 \\ 3.0 \end{array}$$

Obteniendo los enteros de los resultados de las multiplicaciones

$$(238.12)_{10} = (1423.03)_5$$

DADO UN NUMERO EN BASE R CONVERTIRLO A LA BASE 10

$$(1423.03)_5 = ()_{10}$$

$$\begin{aligned} 1(5)^{4-1} + 4(5)^{4-2} + 2(5)^{4-3} + 3(5)^{4-4} + 0(5)^{-1} + 3(5)^{-2} = \\ 1(5)^3 + 4(5)^2 + 2(5)^1 + 3(5)^0 + 0(5)^{-1} + 3(5)^{-2} = \end{aligned}$$

$$1(125) + 4(25) + 2(5) + 3(1) + 0\left(\frac{1}{5}\right) + 3\left(\frac{1}{25}\right) =$$

$$125 + 100 + 10 + 3 + 0 + \left(\frac{3}{25}\right) =$$

= 238.12

$$(1423.03)_5 = (238.12)_{10}$$

SISTEMA DE NUMERACION BINARIO

R=2

$A_i = \{0,1\}$

0/1	0/1	0/1	0/1	0/1
2^4	2^3	2^2	2^1	2^0
16	8	4	2	1

	2^4	2^3	2^2	2^1	2^0
	16	8	4	2	1
2					
7			0	1	0
15			1	1	1
		1	1	1	1

SISTEMA DE NUMERACION OCTAL

R=8

$A_i = \{0,1,2,...,7\}$

0-7	0-7	0-7	0-7	0-7
8^4	8^3	8^2	8^1	8^0
4096	512	64	8	1

	8^4	8^3	8^2	8^1	8^0
	4096	512	64	8	1
2					
7					2
15				1	7
12				1	4

Checa colocamos una unidad (1) donde queremos que represente un 8 al hacer la conversión.

SISTEMA DE NUMERACION HEXADECIMAL

R=16

$A_i = \{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F\}$

				2^4
0-F	0-F	0-F	0-F	0-F
16^4	16^3	16^2	16^1	16^0
65536	4096	256	16	1

	16^4	16^3	16^2	16^1	16^0
	4096	512	64	16	1
2					2
7					7
15					F
12					C
20				1	4

Checa colocamos una unidad (1) donde queremos que represente un 8 al hacer la conversión.

OPERACIONES ARITMETICAS

SUMA y MULTIPLICACION

$$\begin{array}{r} + \quad | \quad 0 \quad 1 \\ 0 \quad | \quad 0 \quad 1 \\ 1 \quad | \quad 1 \quad 10 \end{array}$$

$$\begin{array}{r} \times \quad | \quad 0 \quad 1 \\ 0 \quad | \quad 0 \quad 0 \\ 1 \quad | \quad 0 \quad 1 \end{array}$$

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

RESTA

$$\begin{array}{r} \quad 5 \quad \quad 1 \quad 2^0 \quad 1 \\ - \quad 2 \quad \quad 0^{+1} \quad 1 \quad 0 \\ \hline \quad 3 \quad \quad 0 \quad 1 \quad 1 \end{array}$$

$$\begin{array}{r} \quad 299 \quad \quad 1 \quad \quad 2^0 \quad \quad 2^0 \quad \quad 1 \quad \quad 2^0 \quad \quad 1 \quad \quad 1 \\ - \quad 214 \quad \quad 1^{+1} \quad \quad 1^{+1} \quad \quad 1 \quad \quad 0^{+1} \quad \quad 1 \quad \quad 0^{+1} \quad \quad 1 \quad \quad 0 \\ \hline \quad 85 \quad \quad 0 \quad \quad 0 \quad \quad 1 \quad \quad 0 \quad \quad 1 \quad \quad 0 \quad \quad 1 \quad \quad 1 \end{array}$$

$$1-1=0 \quad 2-2=0 \quad 2-1=1 \quad 1-1=0 \quad 2-1=1 \quad 1-1=0 \quad 2-1=1 \quad 1-1=0 \quad 1-0=1$$

$$\begin{array}{r} 2 \\ - 5 \\ \hline +1 \\ C=1 \end{array} \quad \begin{array}{r} 20 \quad 1 \quad 20 \\ \hline 1 \quad 0+1 \quad 1 \\ 0 \quad 1 \end{array}$$

MULTIPLICACION

$$\begin{array}{r} 1 \ 0 \ 1 \\ x 1 \ 0 \\ \hline 0 \ 0 \ 0 \\ 1 \ 0 \ 1 \\ \hline 1 \ 0 \ 1 \ 0 \end{array} \quad \begin{array}{r} 5 \\ x 2 \\ \hline 10 \end{array}$$

DIVISION

$$\begin{array}{r} \\ 1 \ 0 \ 1 \overline{) 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1} \\ \underline{1 \ 0 \ 1} \\ 0 \ 0 \ 1 \ 1 \ 0 \\ \underline{1 \ 0 \ 1} \\ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \\ \underline{1 \ 0 \ 1} \\ 0 \ 0 \ 0 \end{array}$$

Cuántas veces cabe el 5 en el 6, ps 1

SISTEMAS DE NUMERACION COMPLEMENTARIOS

Para abordar este tema es necesario ver la representación de números negativos, el cual la mayor parte de las computadoras emplean alguno de los sistemas numéricos de complemento que se presentarán posteriormente.

Representación de magnitud con signo

En el sistema de magnitud con signo, un número se compone de una magnitud y de un símbolo que indica si la magnitud es positiva o si es negativa. De esta forma, interpretamos los números decimales +98, -57, +123.5, y -13, de la manera habitual, y también suponemos que el signo es "+0" y "-0", pero ambas tienen el mismo valor.

El sistema de magnitud con signo se aplica a los números binarios haciendo uso de una posición de bit extra para representar el signo (el bit de signo). Tradicionalmente, el bit más significativo (MSB) de una cadena de bits es empleado como bit de signo (0=signo +, 1= signo -) y los bits de menor orden contienen la magnitud. Así podemos describir varios enteros de 8 bits con magnitud de signo y sus equivalencias decimales.

$$01010101_2 = 85_{10} \quad 11010101_2 = -85_{10}$$

el sistema de magnitud con signo tiene un número idéntico de enteros positivos y negativos. Un entero de magnitud con signo de n bits está situado dentro del intervalo que va desde $-(2^{n-1}-1)$ hasta $+(2^{n-1}-1)$ y existen dos posibles representaciones del cero.

Mientras un sistema de magnitud con signo convierte en negativo un número al cambiar su signo, un **sistema numérico de complemento** convierte en negativo un número tomando su complemento como definido por el sistema. Tomar el complemento es más difícil que cambiar el signo, pero dos números en un sistema numérico de complemento pueden sumarse o restarse directamente sin tener que usar las verificaciones de magnitud y signo que requiere el sistema de magnitud con signo. Describiremos dos sistemas numéricos de complemento, llamados "el complemento de base" y "el complemento de base reducida".

En cualquier sistema numérico de complemento, normalmente tratamos con un número fijo de dígitos, digamos n. (Sin embargo, podemos aumentar el número de dígitos mediante "extensión de signo" como se muestra en el siguiente ejercicio, y disminuir el número mediante el truncamiento de los dígitos de orden mayor como se muestra en el segundo ejercicio.) suponemos adicionalmente que la base es r, y que los números tienen la forma

$$D = d_{n-1}d_{n-2}d_{n-3} \dots d_1d_0$$

El punto de base se encuentra a la derecha y por tanto el número es un entero. Si una operación produce un resultado que requiera más de n dígitos, eliminamos el (los) dígito(s) extra de mayor orden. Si un número D se complementa dos veces, el resultado será D .

REPRESENTACION DE COMPLEMENTO DE BASE

En un sistema de complemento de base, el complemento de un número de n dígitos se obtiene al restarlo de r^n . en el sistema numérico decimal, el complemento de base se denomina complemento de 10. Algunos ejemplos utilizando números decimales de 4 dígitos (y resta de 10,000) se muestran en la siguiente tabla.

número	Complemento de 10	Complemento de 9
1849	8151	8150
2067	7933	7932
100	9900	9899
7	9993	9992
8151	1849	1848
0	10000 (=0)	9999

REPRESENTACION DE NUMEROS BINARIOS SIGNADOS UTILIZANDO COMPLEMENTOS

A) COMPLEMENTO A LA BASE DISMINUIDA (B-2). Si se tiene un numero binario signado de 4 bits

D3	D2	D1	D0		D3	D2	D1	D0	
0	0	0	0	+0	1	1	1	1	-0
0	0	0	1	+1	1	1	1	0	-1
0	0	1	0	+2	1	1	0	1	-2
0	0	1	1	+3	1	1	0	0	-3
0	1	0	0	+4	1	0	1	1	-4
0	1	0	1	+5	1	0	1	0	-5
0	1	1	0	+6	1	0	0	1	-6
0	1	1	1	+7	1	0	0	0	-7

(cambian 1's por 0's)

Operaciones

$$\begin{array}{r}
 0010 \quad +2 \\
 + 0100 \quad +4 \\
 \hline
 0110 \quad +6
 \end{array}$$

$$\begin{array}{r}
 1110 \quad -1 \\
 + 1100 \quad -3 \\
 \hline
 C=1 \quad 1010 \quad \text{---} \\
 \quad \quad + 1 \quad \text{---} \\
 \hline
 1011 \quad -4
 \end{array}$$

$$\begin{array}{r}
 0011 \quad +3 \\
 + 1101 \quad -2 \\
 \hline
 C=1 \quad 0000 \quad \text{---} \\
 \quad \quad + 1 \quad \text{---} \\
 \hline
 0001 \quad +1
 \end{array}$$

TENEMOS DEFINIDOS DOS CEROS +0 Y -0 y se necesita sumar el acarreo para obtener el número verdadero.

B) CON COMPLEMENTO A LA BASE. Se tiene un numero binario signado de 4 bits

D3	D2	D1	D0		D3	D2	D1	D0	
0	0	0	0	+0					
0	0	0	1	+1	1	1	1	1	-1
0	0	1	0	+2	1	1	1	0	-2
0	0	1	1	+3	1	1	0	1	-3
0	1	0	0	+4	1	1	0	0	-4
0	1	0	1	+5	1	0	1	1	-5
0	1	1	0	+6	1	0	0	1	-6
0	1	1	1	+7	1	0	0	0	-7

(se intercambian 1's por 0's y viceversa y se le suma la unidad)

$$\begin{array}{r}
 0 \ 0 \ 1 \ 0 \quad +2 \\
 + \ 0 \ 1 \ 0 \ 0 \quad +4 \\
 \hline
 \mathbf{0 \ 1 \ 1 \ 0} \quad +6
 \end{array}$$

$$\begin{array}{r}
 1 \ 1 \ 1 \ 1 \quad -1 \\
 + \ 1 \ 1 \ 0 \ 1 \quad -3 \\
 \hline
 C=1 \ \mathbf{1 \ 1 \ 0 \ 0} \quad -4
 \end{array}$$

$$\begin{array}{r}
 0 \ 0 \ 1 \ 1 \quad +3 \\
 + \ 1 \ 1 \ 1 \ 0 \quad -2 \\
 \hline
 C=1 \ \mathbf{0 \ 0 \ 0 \ 1} \quad +1
 \end{array}$$

i ya no se necesita sumar el acarreo para obtener el verdadero numero i

CODIGOS

Codificar es un proceso en el cual a una combinación de dígitos se le asigna una etiqueta o una identificación en nuestro caso la codificación de dígitos son de tipo binario.

BIT 1 bit
NIBBLE 4 dígitos binarios
BYTE 8
WORD 16

	BCD				EXCESO 3				DE PESO			
	8	4	2	1					8	4	-2	-1
	D3	D2	D1	D0	D3	D2	D1	D0	D3	D2	D1	D0
0	0	0	0	0	0	0	1	1	0	0	0	0
1	0	0	0	1	0	1	0	0	0	1	1	1
2	0	0	1	0	0	1	0	1	0	1	1	0
3	0	0	1	1	0	1	1	0	0	1	0	1
4	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
6	0	1	1	0	1	0	0	1	1	0	1	0
7	0	1	1	1	1	0	1	0	1	0	0	1
8	1	0	0	0	1	0	1	1	1	0	0	0
9	1	0	0	1	1	1	0	0	1	1	1	1

CODIGO REFLEJADO

	D3	D2	D1	D0
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	1	0	0
6	1	0	1	1
7	1	0	1	0
8	1	0	0	1
9	1	0	0	0

CODIGO DE DISTANCIA UNITARIA

	D3	D2	D1	D0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	1	1
6	1	1	1	1
7	1	0	1	1
8	1	0	0	1
9	1	0	0	0

CODIGO GRAY

	D3	D2	D1	D0		
0	0	0	0	0	Todos ceros	
1	0	0	0	1	Cambia el primer bit	1
2	0	0	1	1	Cambia el segundo bit	2
3	0	0	1	0	Cambia el primer bit	1
4	0	1	1	0	Cambia el tercer bit	3
5	0	1	1	1	Cambia el primer bit	1
6	0	1	0	1	Cambia el segundo bit	2
7	0	1	0	0	Cambia el primer bit	1
8	1	1	0	0	... tocaría cambiar el cuarto bit	4
9	1	1	0	1	Cambia el primer bit	1
10	1	1	1	1	Se reflejan los bits	2
11	1	1	1	0	Cambia el primer bit	1
12	1	0	1	0	Cambia el primer bit	3
13	1	0	1	1	Cambia el primer	1
14	1	0	0	1	Cambia	2
15	1	0	0	0	Cambia	1

BCD A GRAY

0 1 0 0

→ SIN CONTAR ACARREO, SUMAR LOS DIGITOS INMEDIATOS Y
SERÁ EL RESULTADO.

0 1 1 0

GRAY A BCD

0 → 1 1 0

SUMAR EL NUMERO BAJADO CON EL SEGUNDO DIGITO, SE
OBTIENE EL SEGUNDO DIGITO BCD, SIGUIENTE SUMAS
SEGUNDO DIGITO BCD CON TERCER DIGITO GRAY Y SE
OBTIENE EL TERCER DIGITO BCD Y ASI SUCESIVAMENTE,
SIN CONTAR ACARREO.

0 1 0 0

CODIGOS DE DETECCION DE ERRORES

BIT DE PARIDAD → PAR – IMPAR

Ejemplo: dato de 4 bits.

	D3	D2	D1	D0	PAR	IMPAR
0	0	0	0	0	0	1
1	0	0	0	1	1	0
2	0	0	1	0	1	1
3	0	0	1	1	0	1
4	0	1	0	0	1	1
5	0	1	0	1	0	0
6	0	1	1	0	0	1
7	0	1	1	1	1	0
8	1	0	0	0	1	0
9	1	0	0	1	0	1
10	1	0	1	0	0	1
11	1	0	1	1	1	0
12	1	1	0	0	0	1
13	1	1	0	1	1	0
14	1	1	1	0	1	0
15	1	1	1	1	0	1

Dato + bit de polaridad

Par → si la cantidad de 1's es par
Impar → si la cantidad de 1's es impar

Tarea 2

CODIGOS ALFANUMERICOS

En adición a los datos numéricos, una computadora debe ser capaz de manejar información no numérica. En otras palabras, una computadora debe reconocer código que representa letras del alfabeto, puntuaciones, y otros caracteres especiales así como números. Estos códigos son llamados códigos alfanuméricos. Un código alfanumérico completo incluiría las 26 letras minúsculas, 26 letras mayúsculas, 10 dígitos decimales, 7 marcas de puntuación y cualesquier otros 20 a 40 caracteres. Tales como +, -, %, # y así sucesivamente. Podemos decir que un código alfanumérico representa varios caracteres y funciones que están contempladas en un teclado estándar.

CODIGO ASCII

El código alfanumérico mas usado, el American Estándar Code for Information Interchange (ASCII), es usado en la mayoría de las microcomputadoras y minicomputadoras, y en la mayoría de mainframes. El código ASCII es un código de 7 bits, y así tiene grupos de $2^7=128$. Esto es más que suficiente para representar todos los caracteres del teclado así como una lista parcial de funciones de control tales como <RETURN> y <LINEFEED>. La siguiente tabla muestra una lista parcial del código ASCII y en adición se muestra su equivalente en octal y hexadecimal.

Carácter	ASCII 7 bit	Octal	Hex
A	100 0001	101	41
B	100 0010	102	42
C	100 0011	103	43
D	100 0100	104	44
E	100 0101	105	45
F	100 0110	106	46
G	100 0111	107	47
H	100 1000	110	48
I	100 1001	111	49

J	100 1010	112	4 ^a
K	100 1011	113	4B
L	100 1100	114	4C
M	100 1101	115	4D
N	100 1110	116	4E
O	100 1111	117	4F
P	101 0000	120	50
Q	101 0001	121	51
R	101 0010	122	52
S	101 0011	123	53
T	101 0100	124	54
U	101 0101	125	55
V	101 0110	126	56
W	101 0111	127	57
X	101 1000	130	58
Y	101 1001	131	59
Z	101 1010	132	5 ^a
0	011 0000	60	30
1	011 0001	61	31
2	011 0010	62	32
3	011 0011	63	33
4	011 0100	64	34
5	011 0101	65	35
6	011 0110	66	36
7	011 0111	67	37
8	011 1000	70	38
9	011 1001	71	39
BLANK	010 0000	040	20
.	010 1110	056	2E
(010 1000	050	28
+	010 1011	053	2B
\$	010 0100	044	24
*	010 1010	052	2 ^a
)	010 1001	051	29
-	010 1101	055	2D
/	010 1111	057	2F
,	010 1100	054	2C
=	011 1101	075	3D
<RETURN>	000 1101	015	0D
<LINEFEED>	000 1010	012	0A

Y se define como un campo

Codigo
Código de hamming
Códigos alfanuméricos

III Algebra de boole

Son las matemáticas de los sistemas digitales y se define como un campo de operadores, axiomas probados y postulados. $B=\{0,1\}$, vamos a definir dos operaciones binarias: operación or (+) , y operación and (*).

PROPIEDADES

cerradura

$u+v=t$; u,v,t pertenecen a B

$u*v=t$; u,v,t pertenecen a B

asociatividad

sean u,v,t pertenecen a B, se tiene

$$(u+v)+t=u+(v+t)$$

$$(u*v)*t=u*(v*t)$$

conmutatividad

sean u,v pertenecen a B, se tiene

$$u+v=v+u$$

$$u*v=v*u$$

distributividad

sean u,v,t pertenecen a B, se tiene

$$u+(v*t)=(u+v)*(u+t)$$

la operación OR es distributiva sobre la operación and

$$u*(v+t)=(u*v)+(u*t)$$

la operación AND es distributiva sobre la operación or

elemento identidad

$$u+eor=u$$

u pertenece a B; eor=0

$$u*eand=u$$

u pertenece a B; eor=1

elemento complemento

$$u+u_neg=1 \quad u,u_neg \text{ pertencen } B$$

$$u*u_neg=0$$

$$\text{si } u=0 \quad \text{y } u_neg=?$$

$$U_neg=1$$

$$0+u_neg=1$$

$$0 * u_{neg} = 0$$

$$\begin{aligned} \text{si } u=1 \quad u_{neg}=? \quad U_{neg}=0 \\ 1+u_{neg}=1 \\ 1*u_{neg}=0 \end{aligned}$$

en resumen: dos operaciones binarias y una operación unitaria

or

u	v	
0	0	0
0	1	1
1	0	1
1	1	1

and

u	v	
0	0	0
0	1	0
1	0	0
1	1	1

not

u	v
0	1
1	0

al menos existen dos elementos que pertenecen a B tal que **u diferente v, u,v pertenecen B.**

$$B = \{0, 1\}$$

Teorema de identidad

$$U + 0 = U$$

$$U * 1 = U$$

Teorema de la ídem potencia

$$U + u = u$$

$$U * u = u$$

Teorema de involución

$$(U_neg)_neg=u$$

teorema de la absorción

$$u+u*v=u$$

$$u*(u+v)=u$$

teorema de dualidad

$$u+u*v=u$$

$$u*(u+v)=u$$

ejemplo

$$u+0=u$$

$$u*1=u$$

$$u+1=1$$

$$u*0=0$$

demostrar

$$u+(u*v)=u$$

$$u + (uv) = u$$

$$(u+u)*(u+v)=u$$

$$\begin{array}{l} (u+u)(u+v) = u \\ u(u+v) = u \\ u+(u) \end{array}$$

$$u*(u+v)=u$$

$$u+(u*v)=u$$

$$u*1+u*v=u$$

$$u*(1+v)=u$$

$$(1+v)=1$$

por lo que

$$u=u$$

$$u+u_neg*v=?$$

$$(u+u_neg)*(u+v)=$$

$$1*(u+v)=$$

$$u+u_neg*v=u+v$$

$$u + \overline{u}v = u + v$$

Ingresa diagramas de interruptores para OR (conexión en paralelo) y AND (conexión en serie) y NOT (transistor en emisor común):
FUNCION BOOLEANA

$$f(x_2, x_1, x_0) = \overline{x_1} \overline{x_0} + \overline{x_2} x_1 \overline{x_0} + x_2 \overline{x_0} + x_2 x_1$$

$$f(x_2, x_1, x_0) = \overline{x_1} \overline{x_0} + \overline{x_2} x_1 \overline{x_0} + x_2 \overline{x_0} + x_2 x_1$$

$$F(x_2, x_1, x_0) = x_1 \cdot x_0 + x_2 \cdot x_1 \cdot x_0 + x_2 \cdot x_0 + x_2 \cdot x_1$$

TABLA DE VERDAD

X2	X1	X0	X1_*x0_	X2_x1x0_	X2x0_	X2x1	F	$f(x_2, x_1, x_0) = \overline{x_0} + x_2 x_1$
0	0	0	1	0	0	0	1	
0	0	1	0	0	0	0	0	
0	1	0	0	1	0	0	1	
0	1	1	0	0	0	0	0	
1	0	0	1	0	1	0	1	
1	0	1	0	0	0	0	0	
1	1	0	0	0	1	1	1	
1	1	1	0	0	0	1	1	

$$f(x_2, x_1, x_0) = \overline{x_0} (\overline{x_1} + \overline{x_2} x_1 + x_2) + x_2 x_1$$

viendo las variables podemos aplicar la propiedad

$$f(x_2, x_1, x_0) = \overline{x_0} (\overline{x_1} + [\overline{x_2} x_1 + x_2]) + x_2 x_1 \quad \rightarrow \text{aplicando } \boxed{\overline{u} + \overline{uv} = \overline{u} + \overline{v}}$$

$$f(x_2, x_1, x_0) = \overline{x_0} (\overline{x_1} + x_1 + x_2) + x_2 x_1$$

$$f(x_2, x_1, x_0) = \overline{x_0} (1 + x_2) + x_2 x_1$$

$$f(x_2, x_1, x_0) = \overline{x_0} (1) + x_2 x_1$$

$$\boxed{f(x_2, x_1, x_0) = \overline{x_0} + x_2 x_1}$$

Con esto reducimos el número de componentes o de compuertas lógicas.

Insertar ejemplos hechos en clase

Ejemplo 2

$$f(x_2, x_1, x_0) = \overline{x_2}x_0 + x_1x_0 + x_2\overline{x_1}x_0 + \overline{x_2}x_1\overline{x_0} + x_2x_1\overline{x_0}$$

$$f(x_2, x_1, x_0) = x_0(\overline{x_2} + x_1 + x_2\overline{x_1}) + x_1\overline{x_0}(\overline{x_2} + x_2)$$

$$f(x_2, x_1, x_0) = x_0(\overline{x_2} + [x_1 + x_2\overline{x_1}]) + x_1\overline{x_0}([x_2 + x_2])$$

$$f(x_2, x_1, x_0) = x_0(\overline{x_2} + [x_1 + x_2]) + x_1\overline{x_0}([1])$$

$$f(x_2, x_1, x_0) = x_0(\overline{x_2} + x_1 + x_2) + x_1\overline{x_0}([1])$$

$$f(x_2, x_1, x_0) = x_0(x_1 + 1) + x_1\overline{x_0}$$

$$f(x_2, x_1, x_0) = x_0(1) + x_1\overline{x_0}$$

$$f(x_2, x_1, x_0) = x_0 + x_1\overline{x_0}$$

$$f(x_2, x_1, x_0) = x_0 + x_1$$

Ejemplo 3

$$f(x_2, x_1, x_0) = \overline{x_2}x_1 + x_2x_1x_0 + x_1\overline{x_0} + \overline{x_2}x_1\overline{x_0}$$

$$f(x_2, x_1, x_0) = x_1(\overline{x_2} + x_2x_0 + \overline{x_0}) + \overline{x_2}x_1\overline{x_0}$$

--> aplicando $[u + \overline{uv} = \overline{u} + \overline{v}]$

$$f(x_2, x_1, x_0) = x_1(\overline{x_2} + [x_2x_0 + \overline{x_0}]) + \overline{x_2}x_1\overline{x_0}$$

$$f(x_2, x_1, x_0) = x_1(\overline{x_2} + [x_2 + \overline{x_0}]) + \overline{x_2}x_1\overline{x_0}$$

$$f(x_2, x_1, x_0) = x_1(\overline{x_2} + x_2 + \overline{x_0}) + \overline{x_2}x_1\overline{x_0}$$

$$f(x_2, x_1, x_0) = x_1(1 + \overline{x_0}) + \overline{x_2}x_1\overline{x_0}$$

$$f(x_2, x_1, x_0) = x_1 + \overline{x_2}x_1\overline{x_0}$$

--> aplicando $[u + \overline{uv} = \overline{u} + \overline{v}]$

$$f(x_2, x_1, x_0) = x_1 + \overline{x_2}x_1\overline{x_0}$$

$$f(x_2, x_1, x_0) = x_1 + \overline{x_2}x_0$$

- **Construir circuitos digitales con compuertas NAND y compuertas NOR**

$$\text{NOR} \quad \overline{a+b}$$

$$\text{NAND} \quad \overline{a \cdot b}$$

- **Minimización por karnaugh**
- **Minimización por método de Qeen McLausky**