

Notas de la clase de Sistemas Digitales

Marcela Martínez Gutiérrez

Alumna de la Carrera IAS

UANL FIME

Semestre Enero – Junio 2010

Maestro

M.C. Juan Angel Garza Garza

Un agradecimiento especial a **Marcela Martínez Gutiérrez** por su dedicación y talento para elaborar estas notas de clase, que son un documento obligado de consulta de la clase de Sistemas Digitales.

# Sistemas digitales

20-Enero-09

## Sistema

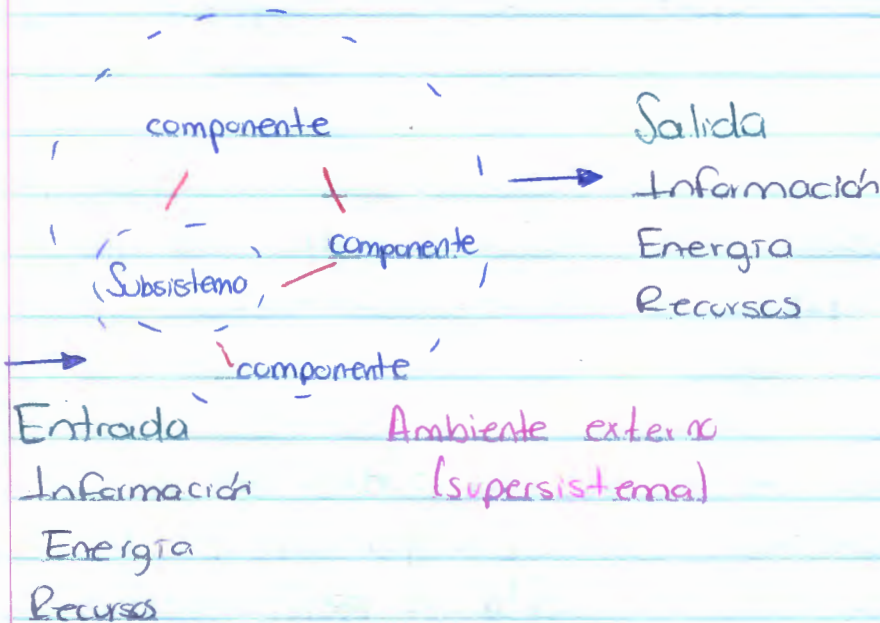
- Un **sistema** comprende un conjunto de componentes que presentan una estructura organizada, habiendo entre ellas una relación tal que lleve al sistema a alcanzar los objetivos propuestos por el mismo.

Ludwig von Bertalanffy

- **Sistema** conjunto de partes o elementos organizadas y relacionados que interactúan entre sí para lograr un objetivo

Los sistemas reciben (**entradas**) datos, energía o materia del ambiente y proveen (**salida**) información, energía o materia

Diccionario informático



20-Enero-09

## Sistema

- 1 m. Conjunto de reglas o principios sobre una materia racionalmente enlazados entre sí.
  - 2 m. Conjunto de cosas que relacionadas entre sí ordenadamente contribuyen a determinado objeto.
  - 3 m. Biol. Conjunto de órganos que intervienen en alguna de las principales funciones vegetativas. Sistema nervioso
- [www.rae.es](http://www.rae.es)

- **Sistema** Conjunto de cosas o partes coordinadas según una ley, o que, ordenadamente relacionadas entre sí, contribuyen a determinado objeto o función.

[Diccionario General de la Lengua Española Vox](#)

## Sistema

Medio o manera usadas para hacer una cosa.  
Manera de estar dispuesto un aparato o utensilio

## Sistema

Una primera clasificación de los sistemas podría ser la siguiente: Existen sistemas **naturales** (los generados por la naturaleza) y sistemas **artificiales** (hechos por el hombre).  
Esta claro que los sistemas electrónicos se encuentran dentro de los s. artificiales

20-Enero-09

Digital

Adjetivo. De los dedos o relativo a ellos: huella digital

Digital Que expresa o suministra los datos por medio de numeros. Larouse 2004

Digital

Fisica: Se dice del aparato o instrumento que mide cantidades y las representa con numeros digitos: reloj digital.

Instrumento que suministra su informacion mediante numeros. reloj circuito

Digital

• Cualquier señal o modo de transmisión que utiliza valores discretos en lugar de un espectro continuo de valores (como las señales analógicas) *Diccionario Informatica*

• Representación de información de modo binario (2 estados). Suele utilizarse en pedagogía y psicología para hablar de la destreza digital.

Esta destreza interviene en los trabajos de precisión, como en el manejo de piezas finas, que exigen un desarrollo adecuado de la motricidad fina.

Sistema que usa muestras digitales (valores discretos codificados en binario) para representar señales analógicas.

20-Enero-09

**Digital:** f. planta herbácea de tallo sencilla o poco ramosa, hojas vellosas y flores en racimo, que se emplea en medicamentos que combaten la insuficiencia cardíaca.

## Analogico

Adjetivo Analogo Que tiene analogia con algo.

Fisica Que representa de manera continua en el tiempo la evolucion de una magnitud: señal analógica

## Sistemas Analógicos

Estas magnitudes son:

Temperatura, presión, longitud, velocidad, tensión, intensidad, aceleración, etc. que tienen un carácter continuo o analógico.

Según la naturaleza de la información que lleva la señal eléctrica esta puede clasificarse en:

- Analógica
- Digital

## Señal analógica

El modelo matemático que la describe es una función continua, por tanto transporta una información analógica y puede tomar infinitos valores frente al tiempo.

20-enero-09

## Señal Digital

Señal digital el modelo matemático que la describe es una función que solo puede tomar un conjunto finito de valores, que transporta una información digital.

El tipo de señal con la que trabajaremos en electrónica digital será un caso particular de la señal digital, la señal digital binaria, que toma dos valores lógicos, normalmente uno y cero.

### Codificación analógica y digital

El proceso que consiste en convertir una señal o variable analógica en digital se denomina digitalización.

En la figura se muestra la digitalización de una señal analógica senoidal con una resolución de 3 bits.

La mayoría de los sistemas electrónicos construidos procesan señales digitales, pero el mundo físico es fundamentalmente analógico como hemos visto.

En consecuencia un sistema digital muy a menudo debe tratar con señales analógicas en su punto de contacto con el mundo exterior (sus entradas)

20-Enero-10

## Conversión analógico-Digital (ADC, Analogic to Digital conversion)

La conversión analógico-Digital consta de varios procesos)

- Muestreo
- Cuantización
- Codificación

## Ventajas de los sistemas digitales

1. Mas fáciles de diseñar
2. Facilidad para almacenar y procesar la información
3. Programación de la operación
4. Los circuitos digitales se afectan menos con el ruido

## Limitaciones de los sistemas digitales

Cuando empleamos técnicas digitales existe sola una desventaja:

El mundo real es fundamentalmente analógico

## Sistema Digital

- Es cualquier dispositivo destinado a la generación, transmisión, procesamiento o almacenamiento de señales digitales.
- Para el análisis y la síntesis de sistemas digitales binarios se utiliza como herramientas el álgebra de Boole.

20-Enero-10

## Un sistema digital

Los sistemas digitales pueden ser de dos tipos,

- **Sistemas digitales combinatoriales**

Aquellos en las que sus salidas solo depende del estado de sus entradas en un momento dado. Por lo tanto, no necesita modulos de memoria, ya que las salidas no dependen de los estados previos de las entradas.

- **Sistemas digitales Secuenciales.**

Aquellos en las que sus salidas dependen ademas del estado de sus entradas en un momento dado, de estados previos. Esta clase de sistemas necesitan elementos de memoria que recojan la informacion de la 'historia pasada' del sistema.

Para la implementacion de los circuitos digitales, se utilizan puertas logicas (AND, OR y NOT), construidos generalmente a partir de transistores

Estas puertas siguen el comportamiento de algunas funciones del booleanas.

- Según el proposito los sistemas digitales se clasifican en:

- a) Sistemas de propósitos especiales

- b) Sistemas de propósitos generales

Estos ultimas permiten el cambio de su comportamiento mediante la programación de algoritmos de soluciones de problemas específicos



20-Enero-10

## Conclusiones

Para aprovechar las técnicas digitales cuando se tienen entradas y salidas analógicas, se siguen los 3 pasos.

1. Convertir las entradas analógicas a la forma digital
2. Almacenar y procesar la información digital
3. Convertir las salidas digitales a la forma analógica del mundo real.

## Actividad 2

1. Encontrar por lo menos 3 definiciones de S.D. en libros (no válida Internet)
2. Se recomienda hojear los libros para ver el contenido acerca del curso
3. Redactar una definición propia de Sistemas digitales tomando en cuenta que no debe de usarse lenguaje muy técnico

Revisar Bibliografía

Nombre del libro, Autor, editorial, año de la edición, ISBN, páginas de referencia, biblioteca donde se encuentra.

Para entregar el próximo

Lunes 25 de Enero

antes de las 14:00 hrs en Informática

20-Enero-10

Analogica - mas comoda ¿Cuanto falta para?

Digital - ¿Que hora es? exactitud

Exactitud - Depende de la calidad de los componentes

Principal ventaja de Sistemas Digitales.

- Almacena mas facil informacion

- Procesa la informacion

1.

"Nombre del libro": Fundamentos de Diseño Digital

Autor: Ing. Cesar A. Leal Chapa

Editorial: Fime UANL

Año edición: Febrero de 1989

~~Pag.~~ ISBN: PDF en pagina web

Páginas de referencia: Pag. 10

Bibliotecas donde se encuentra: 1

2. Nombre del libro: "Electronica Digital <sup>Teoria, Problemas y Simulación</sup> Introducción a la logica D."

Autor: Santiago Acha • Manuel A. Castro • Julio Perez • Nigel A. Rosser

Editorial: Alfaomega La-Ma

Año edición: 2003

ISBN: 970-15-0892-0

Páginas de referencia: 4, 5, 6

Biblioteca donde se encuentra:

3. Nombre del libro: "Sistemas de comunicaciones electronicas"

Autor: Tomasi, Wayne

Editorial: Pearson Educación 4ª

Año edición: 2003

ISBN: 970-28-0318-1

Páginas de referencia: 889

Biblioteca donde se encuentra:

# Sistemas Numericos

22-Ene-10

Con numeros se puede demostrar cualquier cosa.

Thomas Carlyle

Numeración Sistema de simbolos o signos utilizados para expresar los numeros.

Numeración griega, Numeración China, Numeración Maya

Numeros Romanos Es un sistema de numeración que usa letras mayusculas a las que se ha asignado un valor numerico.

Se usa principalmente:

- En los numeros de capitulos y tomos de una obra.
- En los actos y escenas de una obra de teatro.
- En los nombres de papas, reyes y emperadores.
- En la designacion de congresos, olimpiadas, asambleas, certámenes.
- En la fecha de las peliculas

1 = I	2 = II	3 = III	4 = IV	5 = V	6 = VI
7 = VII	8 = VIII	9 = IX	10 = X	11 = XI	12 = XII
13 = XIII	14 = XIV	15 = XV	16 = XVI	17 = XVII	18 = XVIII
19 = XIX	20 = XX	21 = XXI	29 = XXIX	30 = XXX	31 = XXXI
39 = XXXIX	40 = XL	50 = L	51 = LI	59 = LIX	60 = LX
61 = LXI	68 = LXVIII	69 = LXIX	70 = LXX	71 = LXXI	79 = LXXIX
76 = LXXVI	77 = LXXVII	78 = LXXVIII	79 = LXXIX	80 = LXXX	81 = LXXXI
88 = LXXXVIII	89 = LXXXIX	90 = XC	91 = XCI	99 = XCI	100 = C
101 = CI	109 = CIX	114 = CXIV	149 = CXLIX	399 = CCCXCIX	400 = CD
444 = CDXLIV	445 = CDXLV	449 = CDXLIX	450 = CDL	899 = DCCCXCIX	900 = CM
989 = CMLXXXIX	990 = CMXC	999 = CMXCIX	1000 = M	1010 = MX	1050 = ML

22-feb-10

**Numeración Arabiga** El sistema corriente de notación numerica que es utilizado hoy y en casi todo el mundo es la numeración arabiga

Glifo es un signo grabado, o por extensión, pintado

¿Por que el "1" significa "uno" y el "2" significa "dos"?

Se trata de angulos

Si escribes el numero en su forma primitiva, veras que:

- El numero 1 tiene un angulo
- El numero 2 tiene dos angulos
- El numero 3 tiene tres angulos
- y el "0" no tiene angulos



## Numeración Arabiga

Este sistema fue desarrollado primero por los hindúes y luego por los arabes que introdujeron la innovación de la

### Notación posicional

Solo es posible si existe un numero para el cero.

El guarisma 0 permite distinguir entre 11, 101 y 1001 sin tener que agregar símbolos adicionales.

### La notación posicional

En la notación posicional los numeros cambian su valor según su posición. Por ejemplo el dígito 2 en el numero 20 y el mismo dígito en el 2,000 toman diferente valor.

22-enero-09

## Formula general

Los sistemas numericos que utilizan la notacion posicional se pueden describir con la siguiente formula.

$$N = \sum_{i=n-1}^{i=0} a R^i$$

$N$  = numero  $n$  = el numero de digitos  
 $i$  = Posición  $R$  = rais o base  
 $a$  = coeficiente

Subindice para indicar a que base pertenecen.

Los numeros de notacion posicional se e usa el subindice.

$385_{(10)}$  es el numero trescientos ochenta y cinco de base diez, el subindice  $(10)$  indica que pertenece al sistema decimal.

$10_{(10)}$   $10_{(2)}$   $10_{(16)}$   $10_{(7)}$

## Identificación de la posición

$i$  Posición

+n 4 3 2 1 0 -1 -2 -3 -n

8 2 4 5 7 . 3 1 9

↑ Punto decimal

## Ejemplo

385

En donde el digito 5 ocupa la posición cero, el 8 la uno y el 3 la posición dos, como lo indica la figura.

2 1 0  
3 8 5<sub>(10)</sub>

$$N = \sum_{i=n-1}^{i=0} a R^i$$

$$N = 3(10)^2 + 8(10)^1 + 5(10)^0$$

$$N = 3(100) + 8(10) + 5(1)$$

22-Enero-10

En donde se puede observar que el número adquiere valor dependiendo la posición que guarde.

El 3 que está en la posición 2 se multiplica por 100 que es  $10^2$  como lo llamamos tradicionalmente centenas.

al 8 de posición uno por  $10^1$  o decenas unidades.

al 5 de posición cero 100 unidades

Numero	Posición	Potencia	Nombre
1	0	$10^0$	Unidades
10	1	$10^1$	Decenas
100	2	$10^2$	Centenas
1000	3	$10^3$	Unidades de Millar
10000	4	$10^4$	Decenas de Millar
100000	5	$10^5$	Centena de Millar
1000000	6	$10^6$	Unidad de Millon
10000000	7	$10^7$	Decena de Millon
100000000	8	$10^8$	Centena de Millon
1000000000	9	$10^9$	Unidad de Millon de Millon
10000000000	10	$10^{10}$	Decena de Millar de Millon
100000000000	11	$10^{11}$	Centena de Millar de Millon
1000000000000	12	$10^{12}$	Unidad de Billon

Además del sistema decimal existen otras bases de notación posicional que son empleadas en los sistemas digitales como:

**Binario** o base 2 que consta de solo dos símbolos 0 y 1

22-Enero-10

**Octal** o base 8 consta de ocho símbolos (0,1,2,3,4,5,6,7) y es una representación corta del binario.

Ejemplo:

$$111101110_2 = 756_{(8)}$$

**Hexadecimal** o base 16 consta de 16 símbolos (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F), es la representación corta más usada del binario.

Ejemplo:

$$11110111010_2 = F7A_{(16)}$$

Decimal N(10)	Binario N(2)	Octal N(8)	Hexadecimal N(16)	Quinario N(5)	Base 11 N(11)
0	0	0	0	0	0
1	1	1	1	1	1
2	10	2	2	2	2
3	11	3	3	3	3
4	100	4	4	4	4
5	101	5	5	10	5
6	110	6	6	11	6
7	111	7	7	12	7
8	1000	10	8	13	8
9	1001	11	9	14	9
10	1010	12	A	20	A
11	1011	13	B	21	10
12	1100	14	C	22	11
13	1101	15	D	23	12
14	1110	16	E	24	13
15	1111	17	F	30	14
16	10000	20	10		15
17	10001	21	11		16



22-Enero-10

## Conversiones entre sistemas numericos

$N(x) \rightarrow N(y)$  [ 1) Formula General  
2) Multiplicar por la base y sumar

$N(y) \rightarrow N(x)$  [ 1) Extracci6n de potencias  
2) Residuos

$N(2) \leftrightarrow N(8)$   
 $N(2) \leftrightarrow N(16)$  [ Múltiple

## Formula general

Para numeros con decimales

$$N = \sum_{i=n-1}^{i=0} a_i r^i$$

### Ejemplo 1

Convertir un numero binario a decimal:

$1011.11_2 \rightarrow N_{(10)}$

$$\begin{array}{ccccccc} 3 & 2 & 1 & 0 & & -1 & -2 \\ \leftarrow & & & & & & \rightarrow \\ 1 & 0 & 1 & 1 & . & 1 & 1 \end{array} \quad (2)$$

L Punto decimal

$$N_{(10)} = 1(2)^3 + 0(2)^2 + 1(2)^1 + 1(2)^0 + 1(2)^{-1} + 1(2)^{-2}$$

$$N_{(10)} = 1(8) + 0(4) + 1(2) + 1(1) + 1(0.5) + 1(0.25)$$

$$N_{(10)} = 8 + 0 + 2 + 1 + 0.5 + 0.25 = 11.75_{(10)}$$

$$\boxed{1011.11_2 \rightarrow 11.75_{(10)}}$$

22-febrero-10

### Ejercicio 1

• Convertir

$$100.01_{(2)} \rightarrow N_{(10)}$$

$$1 \overset{2}{\leftarrow} 0 \overset{1}{\leftarrow} 0 \overset{-1}{\leftarrow} . \overset{-1}{\leftarrow} 0 \overset{-2}{\leftarrow} 1_{(2)} = \boxed{4.25_{(10)}}$$

### Ejemplo 2

• Convertir un número octal a decimal

$$25.4_{(8)} \rightarrow N_{(10)} \quad 36_{(7)}$$

1 0 -1

$$25.4_{(8)}$$

$$N_{(10)} = 2(8)^1 + 5(8)^0 + 4(8)^{-1} = 3(7)^1 + 6(7)^0$$

$$= 2(8) + 5(1) + 4(0.125)$$

$$\boxed{25.4_{(8)} \rightarrow 21.5_{(10)}}$$

$$= 16 + 5 + .5 = 21.5_{(10)}$$

$$3(7) + 6(1)$$

$$21 + 6 = 27$$

### Ejercicio 2

• Convertir un número octal a decimal

$$5.2_{(8)} \rightarrow N_{(10)} = \boxed{5.25_{(10)}}$$

### Ejemplo 3

• Convertir un número hexadecimal a decimal

$$AB.8_{(16)} \rightarrow N_{(10)}$$

$$A = 10$$

1 0 -1  
$$AB.8_{(16)}$$

$$B = 11$$

$$C = 12$$

$$N_{(10)} = 10(16)^1 + 11(16)^0 + 8(16)^{-1}$$

$$D = 13$$

$$= 10(16) + 11(1) + 8(1/16)$$

$$E = 14$$

$$= 160 + 11 + 0.5 = \boxed{171.5_{(10)}}$$

$$F = 15$$

22-enero-10

### Ejemplo 3

Convertir un numero de base 5 a decimal.

$$34.2 (5) \rightarrow N(10)$$

1 0 -1

$$34.2 (5)$$

$$3(5)^1 + 4(5)^0 + 2(5)^{-1}$$

$$3(5) + 4(1) + 2(.2) = \boxed{19.4(10)}$$

### Conversiones entre Sistemas numericos

$$N(x) \rightarrow N(10)$$

- (1) Formula General
- (2) Multiplicar por la base y sumar

$$N(10) \rightarrow N(x)$$

- (1) Extraccion de potencias
- (2) Residuos

$$N(2) \leftrightarrow N(8)$$

$$N(2) \leftrightarrow N(16)$$

Multiplo

Multiplicar por la base y sumar y sumar

$$N(x) \rightarrow N(10)$$

Para numeros enteros

22- Enero - 10

En un número de notación posicional el dígito más significativo es el que tiene la ponderación más alta (MSD) y se encuentra más a la izquierda y el dígito menos significativo es el que tiene la ponderación más baja (LSD) y se encuentra más a la derecha.

MSD → 4 5 7 3 ← LSD  
MSD Dígito más significativo  
LSD Dígito menos significativo

En el caso del sistema binario se le llama Bit (Dígito Binario)

MSB → 1 0 1 1 1 ← LSD  
MSB Bit más significativo  
LSD Bit menos significativo

• **Bit** La unidad de medida más pequeña de la información digital. Un bit sólo tiene dos posibles valores: 0 ó 1. La palabra 'bit' se forma al combinar "b" de binary y la letra "t" de digit, o sea dígito binario.

• **Byte** = Unidad de medida de la información digital, equivalente a 8 bits o un carácter de información.

- El byte es una unidad común de almacenamiento en un sistema de cómputo y es sinónimo de carácter de datos o de texto; 10000 bytes equivalente a 100,000 caracteres.

- Los bytes se emplean para hacer referencia a la capacidad del hardware, al tamaño del software o a la información.

- Se llama también octeto

## Multiplicar por la base y Sumar.

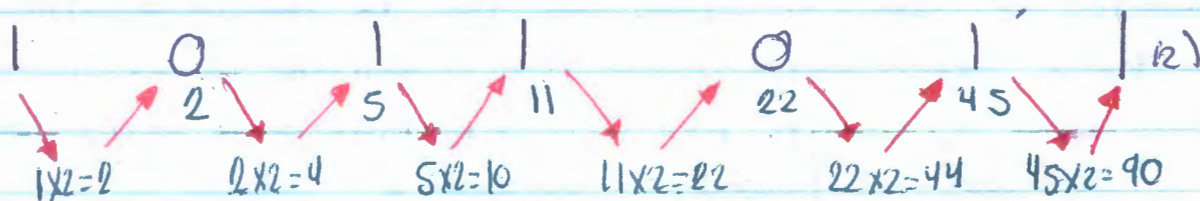
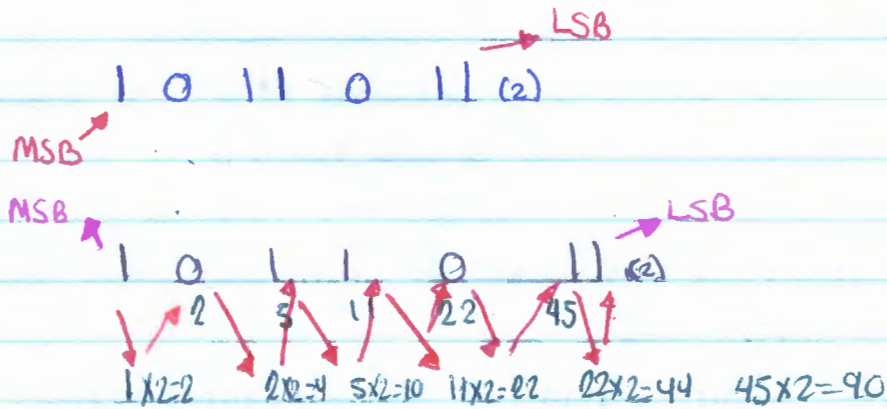
Este método consiste en multiplicar el MSD o MSB (mas significativo dígito o mas significativo Bit) por la base y el producto se suma al valor del dígito siguiente, el resultado se multiplica de nuevo por la base y el producto se suma al dígito siguiente y así sucesivamente hasta llegar al LSD o LSB, de modo que el resultado de todas las operaciones es el número equivalente decimal.

## Multiplicar por la base y Sumar

### Ejemplo 1

Convertir un número binario a decimal:

$$1011011_2 \rightarrow N_{10}$$



$$\boxed{= 91}$$

22-Enero-10

### Ejemplo 2

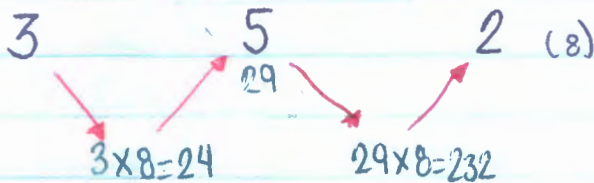
Convertir un numero Octal a decimal:

$210$   
 $352$

$$3(8)^2 + 5(8)^1 + 2(8)^0$$

$192 \quad 40$

$$352_{(8)} \rightarrow N_{(10)}$$



### Ejemplo 3

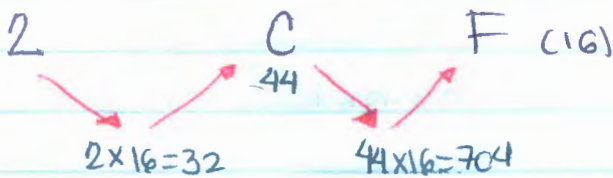
Convertir un numero Hexadecimal a decimal

A=10    D=13

B=11    E=14

C=12    F=15

$$2CF_{(16)} \rightarrow N_{(10)}$$



### Ejemplo 4

Convertir un numero de base Cinco a decimal:

$$223_{(5)} \rightarrow N_{(10)}$$



22-enero-10

Ejemplo 5 convertir un numero de base siete a decimal

$$340_7 \rightarrow N_{(10)}$$

$$\begin{array}{ccc} 3 & 4 & 0 \\ \swarrow \nearrow & \swarrow \nearrow & \\ 3 \times 7 = 21 & 4 \times 7 = 28 & \end{array}$$

$$\boxed{175_{(10)}}$$

Realice la siguiente Actividad

Convertir un numero binario a decimal

$$11001_2 \rightarrow N_{(10)}$$

$$\boxed{11001_2 = 25_{(10)}}$$

Convertir un numero de base 4 a decimal:

$$1121_4 \rightarrow N_{(10)}$$

$$\boxed{1121_4 = 89_{(10)}}$$

Conversiones entre sistemas numericos

$$N_{(x)} \rightarrow N_{(10)} \left[ \begin{array}{l} 1) \text{ Formula general} \\ 2) \text{ Multiplicar por la base y sumar} \end{array} \right.$$

$$N_{(10)} \rightarrow N_{(x)} \left[ \begin{array}{l} 1) \text{ Extracción de potencias} \\ 2) \text{ Residuos} \end{array} \right.$$

$$\begin{array}{l} N_{(2)} \leftrightarrow N_{(8)} \\ N_2 \leftrightarrow N_{(16)} \end{array} \left[ \text{Multipl} \right.$$

25-Ene-10

## Extracción de Potencias $N_{(10)} \rightarrow N_{(2)}$

Para números con decimales.

Este método consiste en tres pasos.

**Primero** elaborar una tabla de potencias de la base a la cual se va a convertir el número decimal.

**Segundo** Restar sucesivamente al número en base diez la potencia igual o próxima menor hasta que la diferencia sea igual a cero.

**Tercer** con las potencias utilizadas en la resta formar el número.

**Ejemplo 1** Convertir un número decimal a binario

$$25.5_{(10)} \rightarrow N_{(2)}$$

En donde el rango de valores asignado a la tabla para efectuar la resta deberá cubrir de un valor menor a 0.5 que representa la parte más pequeña de número 25.5 la potencia requerida es  $2^{-2} = 0.25$  y un valor mayor a 25 como  $2^5 = 32$

### 1° Tabla de Potencias

$$2^{-2} \quad .25$$

$$2^{-1} \quad .5$$

$$2^0 \quad 1$$

$$2^1 \quad 2$$

$$2^2 \quad 4$$

$$2^3 \quad 8$$

$$2^4 \quad 16$$

$$2^5 \quad 32$$

### 2° Restar Sucesivamente

$$25.5$$

$$\underline{-16.0} \quad 2^4$$

$$9.5$$

$$\underline{-8.0} \quad 2^3$$

$$1.5$$

$$\underline{-1.0} \quad 2^0$$

$$0.5$$

$$\underline{-0.5} \quad 2^{-1}$$

$$0.0$$

### 3° Formar el número

4	3	2	1	0	-1
1	1	0	0	1	1

$$25.5_{(10)} = 11001.1_{(2)}$$



25-ene-10

Ejemplo 2  $25.5_{(10)} \rightarrow N_{(8)}$

1° Tabla de Potencias

$8^{-1}$	.125
$8^0$	1
$8^1$	8
$8^2$	64

2° Restar sucesivamente

$$\begin{array}{r} 25.5 \\ - 24.0 \quad 3 \text{ veces } 8^1 \\ \hline 1.5 \\ - 1.0 \quad 8^0 \\ \hline 0.5 \\ - 0.5 \quad 4 \text{ veces } 8^{-1} \\ \hline 0.0 \end{array}$$

3° Formar el número

1	0	-1
3	1	4

$$25.5_{(10)} = 31.4_{(8)}$$

Ejemplo 3  $27.5_{(10)} \rightarrow N_{(16)}$

1° Tabla de Potencias

$16^{-1}$	.0625
$16^0$	1
$16^1$	16
$16^2$	256

2° Restar sucesivamente

$$\begin{array}{r} 27.5 \\ - 16.0 \quad 16^1 \\ \hline 11.5 \\ - 11.0 \quad 11 \text{ veces } 16^0 \\ \hline 0.5 \\ - 0.5 \quad 8 \text{ veces } 16^{-1} \\ \hline 0.0 \end{array}$$

3° Formar el número

1	0	-1
1	B	8

$$27.5_{(10)} = 1B.8_{(16)}$$

25-eneio-10

Ejemplo 4  $16.5_{(10)} \rightarrow N_{(16)}$

1° Tabla de Potencias

$16^{-1}$	.0625
$16^0$	1
$16^1$	16
$16^2$	256

2° Restar sucesivamente

$$\begin{array}{r}
 16.5 \\
 - 16.0 \\
 \hline
 0.5 \\
 - 0.5 \quad 8 \text{ veces } 16^{-1} \\
 \hline
 0.0
 \end{array}$$

3° Formar el numero

1	0	-1
1	0	.8

Realice la siguiente Actividad

$$16.5_{(10)} = 10.8_{(16)}$$

$27.6_{(10)} \rightarrow N_{(5)}$

1° Tabla de Potencias

$5^{-1}$	.2
$5^0$	1
$5^1$	5
$5^2$	25

2° Restar sucesivamente

$$\begin{array}{r}
 27.6 \\
 - 25.0 \quad 5^2 \\
 \hline
 2.6 \\
 - 2.0 \quad 2 \text{ veces } 5^0 \\
 \hline
 0.6 \\
 - 0.6 \quad 3 \text{ veces } 5^{-1} \\
 \hline
 0.0
 \end{array}$$

3° Formar el numero

2	1	0	-1
1	0	2	3

$$27.6_{(10)} = 102.3_{(5)}$$

# Conversiones entre sistemas Numericos 25-Ene-10

$N_{(10)} \rightarrow N_{(x)}$  [ 1) Extracción de Potencias  
2) Residuos

## Residuos $N_{(10)} \rightarrow N_{(x)}$

Este metodo consiste en dividir sucesivamente el numero decimal entre la base a la que se desee convertir hasta que el cociente sea menor que la base.

El numero equivalente se forma con el ultimo cociente y los residuos.

**Ejemplo 1** Convertir un numero decimal a binario

$$35_{(10)} \rightarrow N_{(2)}$$

$\begin{array}{r} 17 \\ 2 \overline{) 35} \\ \underline{15} \end{array}$	$\begin{array}{r} 8 \\ 2 \overline{) 17} \\ \underline{1} \end{array}$	$\begin{array}{r} 4 \\ 2 \overline{) 8} \\ \underline{0} \end{array}$	$\begin{array}{r} 2 \\ 2 \overline{) 4} \\ \underline{0} \end{array}$	$\begin{array}{r} 1 \\ 2 \overline{) 2} \\ \underline{0} \end{array}$
15	1	0	0	0

MSB  $\leftarrow$  (1)  
LSB  $\rightarrow$  1

$$\boxed{100011}_{(2)}$$

**Ejemplo 2** Convertir un numero decimal a octal

$$85_{(10)} \rightarrow N_{(8)}$$

$\begin{array}{r} 10 \\ 8 \overline{) 85} \\ \underline{05} \end{array}$	$\begin{array}{r} 1 \\ 8 \overline{) 10} \\ \underline{2} \end{array}$
05	2

MSD  $\leftarrow$  (1)

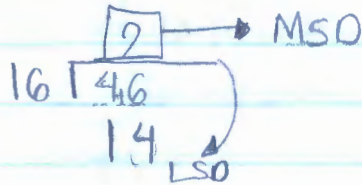
$$\boxed{125}_{(8)}$$

25-Ene-10

### Ejemplo 3 Convertir un numero decimal a Hexadecimal

46(10)  $\rightarrow$  N(16)

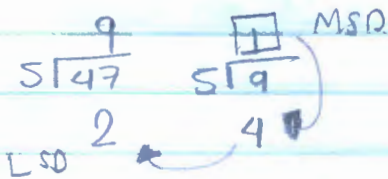
A=10    D=13  
B=11    E=14  
C=12    F=15



2E(16)

### Ejemplo 4 Convertir un numero decimal a base 5

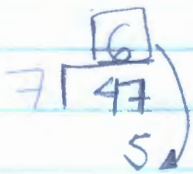
47(10)  $\rightarrow$  N(5)



142(5)

### Ejemplo 5 Convertir un numero decimal a base 7

47(10)  $\rightarrow$  N(7)



65(7)

\* Realice la siguiente Actividad N(x)  $\rightarrow$  N(x) Multiplicar por la base y sumar  
47(8)  $\rightarrow$  N(16)

N(10)  $\rightarrow$  N(x) Residuos

27(16)

25-Enero-10

## Multiplo en Potencia

La relación que existe entre la base dos y la base ocho es de 3 ya que  $2^3=8$ .

De la misma forma entre la base dos y el Hexadecimal es de 4 ya que  $2^4=16$

$$N(2) \longleftrightarrow N(8) = R=3$$

$$N(2) \longleftrightarrow N(16) = R=4$$

### Ejemplo 1 (Conversion de $N(2) \rightarrow N(8)$ )

$$10110101(2) \rightarrow N(8)$$

Separe de en grupos de tres bits iniciando con la de menor peso, como lo indicad la figura

$$10110101(2)$$

$$N(2) \longleftrightarrow N(8) = R=3 \quad 2^3=8$$

De el valor de 1, 2 y 4 a cada digito correspondiente como la muestra la figura

2	1	4	2	1	4	2	1
1	0	1	1	0	1	0	1

$$10110101(2) = 265(8)$$

2      6      5

Obtenga el valor de la suma de los tres bits tomando en cuenta solo los unos.

Siempre se pone 421?

25-Enero-10

Realice la siguiente actividad

Convertir un número binario a octal

$$1010000101_{(2)} \rightarrow N_{(8)}$$

$$1010000101_{(2)}$$

1	4	2	1	4	2	1	4	2	1
1	0	1	0	0	0	0	1	0	1

$$1010000101_{(2)} = 1205_{(8)}$$

1 2 0 5

Conversion de  $N_{(8)} \rightarrow N_{(2)}$

$$603_{(8)} \rightarrow N_{(2)}$$

Cada dígito del octal tiene que representarse por 3 bits

$2^2$	$2^1$	$2^0$
4	2	1

6	0	3						
1	1	0	0	0	0	0	1	1

$$603_{(8)} = 110000011_{(2)}$$

Realice la siguiente Actividad

Convertir un número octal a binario

$$4172_{(8)} \rightarrow N_{(2)}$$

$2^2$	$2^1$	$2^0$
4	2	1

4	1	7	2								
1	0	0	0	0	1	1	1	1	0	1	0

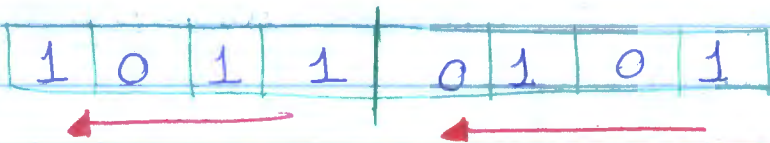
$$4172_{(8)} = 100000111010_{(2)}$$

25-Enero-10

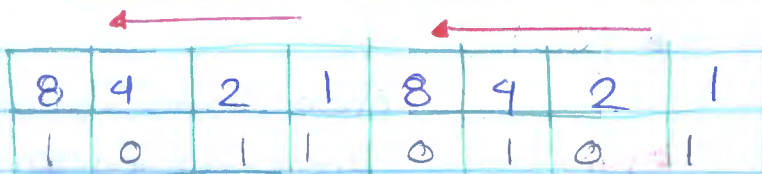
Conversion de  $N(2) \rightarrow N(16)$

ejemplo  $10110101_{(2)} \rightarrow N(16)$

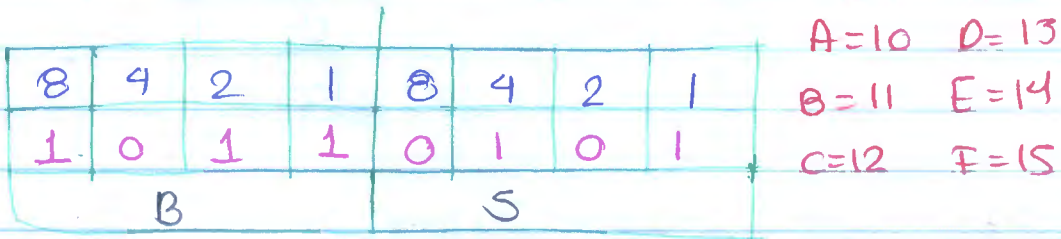
Separe de en grupos de cuatro bits iniciando con lo de menor peso como lo indica la figura.



De el valor de 1, 2, 4 y 8 a cada dígito correspondiente como lo muestra la figura



Obtenga el valor de la suma de los cuatro bits tomando en cuenta solo los unos.



$10110101_{(2)} \rightarrow B5C16$

25-Ene-10

Realice la siguiente Actividad

Convertir un número Binario a Hexadecimal

$$1010,1100_{(2)} \rightarrow N_{(16)}$$

$$A=10 \quad D=13$$

$$B=11 \quad E=14$$

$$C=12 \quad F=15$$

8	4	2	1	8	4	2	1
1	0	1	0	1	1	0	0
A				C			

$$8+4=12=C$$

$$8+2=10=A$$

$$10101100_{(2)} = AC_{(16)}$$

Conversion de  $N_{(16)} \rightarrow N_{(2)}$

$$2DF_{(16)} \rightarrow N_{(2)}$$

Cada Dígito del Hexadecimal tiene que representarse por 4 Bits

$$A=10 \quad D=13$$

$$B=11 \quad E=14$$

$$C=12 \quad F=15$$

$2^3$	$2^2$	$2^1$	$2^0$
8	4	2	1

2	0	F								
0	0	1	0	1	1	0	1	1	1	1

Cada dígito del Hexadecimal tiene que representarse por 4 Bits

$$2DF_{(16)} \rightarrow 1011011111_{(2)}$$



25-Enero-10

Convertir un número Hexadecimal a Binario

5BC<sub>(16)</sub> → N<sub>(2)</sub>

A=10 D=13

B=11 E=14

C=12 F=15

$2^3$	$2^2$	$2^1$	$2^0$
8	4	2	1

5			B=11			C=12				
0	1	0	1	1	0	1	1	1	0	0

5BC<sub>(16)</sub> = 10110111100<sub>(2)</sub>

Conversiones entre sistemas numericos

N<sub>(x)</sub> → N<sub>(10)</sub> [ 1) Formula General  
2) Multiplicar por la base y sumar

N<sub>(10)</sub> → N<sub>(x)</sub> [ 1) extracción de potencias  
2) Residuos

N<sub>(2)</sub> ↔ N<sub>(8)</sub> [ Multiplo  
N<sub>(2)</sub> ↔ N<sub>(16)</sub>

$\begin{array}{r} \sqrt{12} \\ \times 16 \\ \hline 72 \\ \times 12 \\ \hline 192 \end{array}$

25-Enero-10

**Ejemplo 1 Convertir un numero binario  $N(c_2)$  a  $N(c_8)$ ,  $N(c_{10})$ ,  $N(c_{16})$  y  $N(c_5)$ .**

Secuencia propuesta:

$N(c_2) \rightarrow N(c_8)$  Multiplo (Separar de 3 bits en 3 empezando del LSB)

$N(c_2) \rightarrow N(c_{16})$  Multiplo (separar de 4 bits en 4 empezando del LSB)

$N(c_8) \rightarrow N(c_{10})$  Multiplicar por base y sumar

$N(c_{10}) \rightarrow N(c_5)$  Residuos

**Ejemplo 2 Convertir un numero Octal  $N(c_8)$  a  $N(c_2)$ ,  $N(c_{10})$ ,  $N(c_{16})$  y  $N(c_5)$**

Secuencia Propuesta:

$N(c_8) \rightarrow N(c_2)$  Multiplo (cada digito corresponde a de 3 Bits)

$N(c_2) \rightarrow N(c_{16})$  Multiplo (separar de 4 bits en 4 empezando del LSB)

$N(c_{16}) \rightarrow N(c_{10})$  Multiplicar por base y sumar

$N(c_{10}) \rightarrow N(c_5)$  Residuos  $N(c_2) \rightarrow N(c_{16})$   $N(c_8) \rightarrow N(c_2)$

Ej. 1

100100 (2)

44 (8)

24 (16)

36 (10)

100 (5)

8	4	2	1	8	4	2	1
1	1	0	0	0	0	1	0
12 = C				2			

$2^2$	$2^1$	$2^0$
4	2	1

$$\begin{array}{r} 6 \overline{) 36} \quad 6 \overline{) 6} \\ 0 \quad 0 \end{array}$$

3	0	2
0	1	1
0	0	0
0	0	1
0	1	0

11000010

Ej. 2

302 (8)

De base 8 a decimal siempre pasa a decimal (primero)

1100 0010 (2)

C 2 (16)

194 (10)

1234 (5)

16

$\times 10$

160

32

192

$$\begin{array}{r} 38 \\ 5 \overline{) 194} \\ 44 \end{array}$$

$$\begin{array}{r} 27 \\ 5 \overline{) 38} \\ 3 \end{array}$$

$$\begin{array}{r} 1 \\ 5 \overline{) 7} \\ 5 \end{array}$$

44

3

4

3

2

$N(c_8) \rightarrow N(c_{10})$

2  
 $12 \times 16 = 192 + 2 = 194$

# Algebra Booleana

26-Enero-10

## Historia

"George Boole (1815-1864)"

Lógico y matemático británico.

Nacido el 2 de Noviembre de 1815 en Lincoln, Lincolnshire (Inglaterra),

En 1854, escribió Investigación sobre

Las leyes del pensamiento

An Investigation of the Laws of Thought

- Boole fue un niño inteligente y su primer interés fue hacia los idiomas, siendo capaz de dominar el latín completamente con 12 años. Aunque no había estudiado para ella, empezó dedicándose a la enseñanza siendo a los 16 años profesor auxiliar en un colegio.
- También pensó realizar la carrera eclesiástica, pero en 1835 decidió abrir su propio colegio y fue cuando empezó a estudiar matemáticas por su cuenta, estudiando los trabajos de Laplace y Lagrange.
- Se encaminó hacia el Álgebra publicando una aplicación de métodos algebraicos para la resolución de ecuaciones diferenciales por el que recibió la medalla de la Real Sociedad Matemática de Londres.
- En 1849 fue nombrado catedrático de matemáticas en el Queens College, donde ejerció la enseñanza el resto de su vida.
- En 1854 publicó sus estudios sobre las teorías matemáticas

25 Enero 10

de lógica y probabilidad. Boole redujo la lógica a una algebra sencilla, naciendo así lo que se conoce como algebra booleana, la cual influyo en el desarrollo de la informatica

• Boole murio a los 49 años por causa de una pulmonia. Aunque Boole tiene otros muchos estudios en el universo de las matematicas sin duda alguna se le recordará por su algebra, que fue un paso fundamental en el desarrollo de la computadoras.

"George Boole: Las leyes del Pensamiento"

Falso Verdadera

Inteligencia Artificial?

¿Que es lo que hacemos los humanos que dificilmente las maquinas pueden realizar?

Las conexiones entre los telefonos eran manuales, a traves de las centrales por medio de una operadora.

En 1879 la primera patente sobre un sistema de conmutación automatico de Connolly, Connolly, Mc Tighe. Gracias al algebra Booleana se automatizo la conexión.

Claude Elwood Shannon

(30 de abril de 1916, Michigan - 24 de febrero de 2001)

Ingeniero Electricista y matematico, Universidad de Michigan

"El padre de la Teoria de la informacion" Boole 1854

Creator de la era digital

Shanon 1938

- El AND es en serie.

de-cualquier entrada 1 es salida 1.

26-enero-10

En su tesis de maestría en el MIT, demostró como el álgebra booleana se podría utilizar en el análisis y la síntesis de la conmutación y de los circuitos digitales. La tesis despertó un interés considerable cuando apareció en 1938 en las publicaciones especializadas.

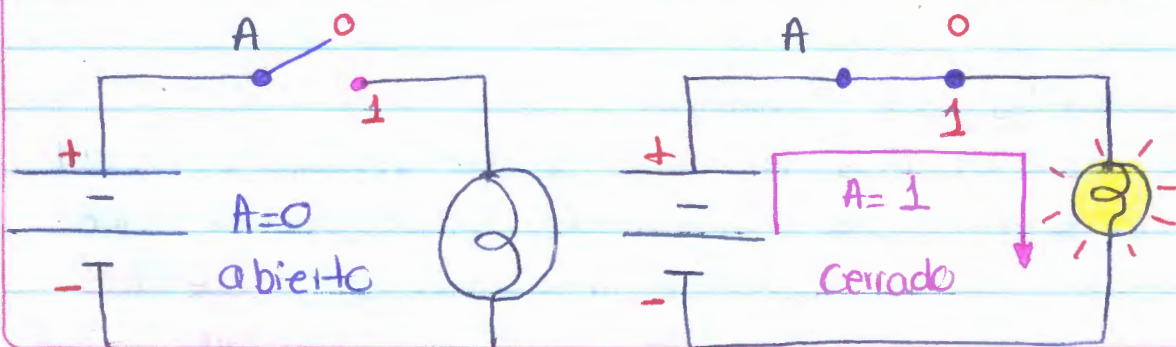
En 1940 le fue concedido el Premio a Ingenieros americanos del Instituto Americano Alfred Nobel de Estados Unidos, una concesión dada cada año a una persona de no más de treinta años.

Un cuarto de siglo más tarde H.H. Goldstine, en su libro "Las computadoras desde Pascal hasta Von Neumann", citó su tesis como una de las más importantes de la historia que ayudó a cambiar el diseño de circuitos digitales.

Una variable Booleana puede tomar solo dos valores:

Falso = 0

Verdadero = 1



Preguntar si aqui no va nada

And-Serie

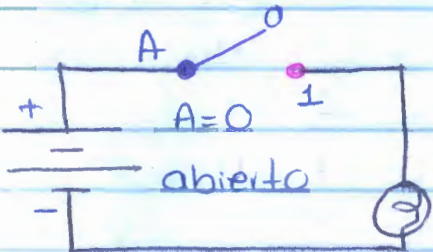
Not se cierra con cero

Or-Paralelo

26-Enero-10

### Tabla de Verdad

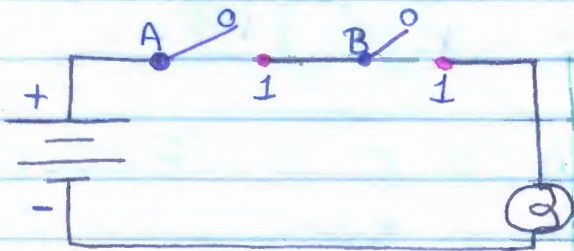
Una variable



m	Entrada	Salida
0	A	Y
1	0	0
2	1	1

### Tabla de Verdad

Das variables



m	Entrada	Salida
0	A B	Y
1	0 0	0
2	0 1	0
3	1 0	0
4	1 1	1

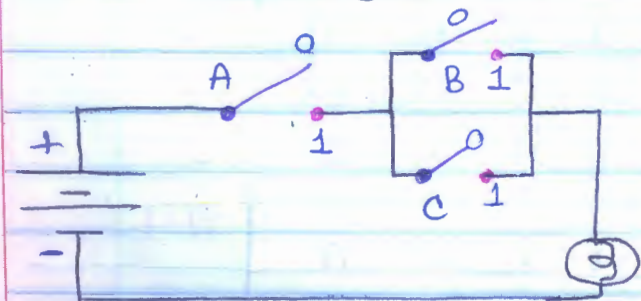
Cuatro combinaciones

### Tabla de Verdad

Cuántas combinaciones se

pueden tener con

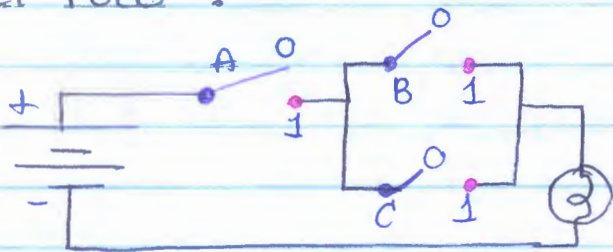
Tres Variables



m	A	B	C	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

26-enero-10

Obtenga la tabla de verdad para que combinaciones enciende el foco?



m	A	B	C	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

### Cuatro Variables

16 combinaciones del 0 al 15 en N(2)

Tabla de verdad

	A	B	C	D	F
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	
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	

26-Enero-10

El número de combinaciones  $m$  depende del número de variables  $N$

$$m = 2^N$$

1 variable 2 combinaciones 4 variables 16 combinaciones  
2 variables 4 combinaciones 5 variables 32 combinaciones  
3 variables 8 combinaciones 6 variables 64 combinaciones

## Operadores Lógicos

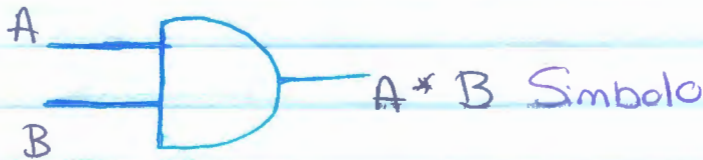
And	Nand	- Nombre	- Tabla de verdad
Or	Nor	- Característica	- Circuito equivalente
Not	Exor	- Símbolo	- Diagrama de Tiempos
	Exnor	- Expresión matemática	

## And condición

La operación And está relacionada con el término de condición y es exactamente igual que la multiplicación ordinaria de unos y ceros.

Una salida igual a 1 ocurre solo en el único caso donde todas las entradas son 1.

La salida es cero cuando una o más de las entradas son igual a 0.



AB    A \* B    A & B    Expresión matemática



26-enero-10

And (condición)

equivalent electrico

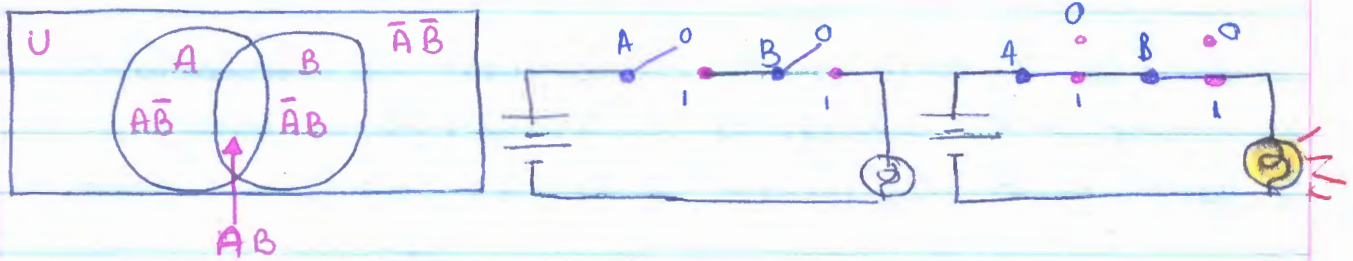
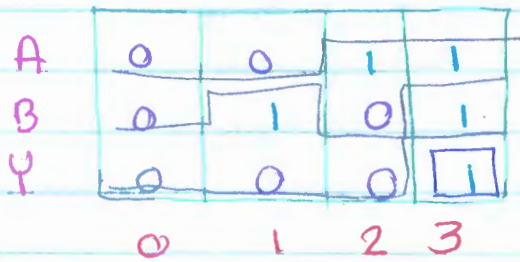
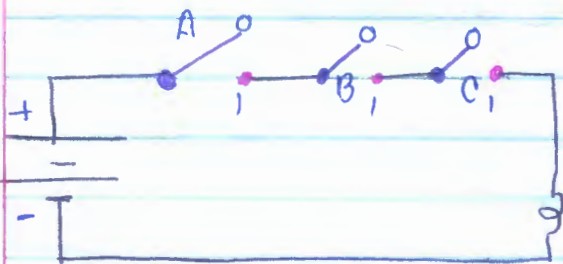


Diagrama de Tiempos

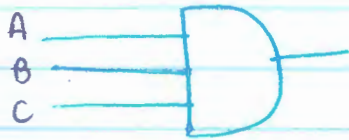
m	A	B	S
0	0	0	0
1	0	1	0
2	1	0	0
3	1	1	1



And de tres entradas



m	A	B	C	S
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1



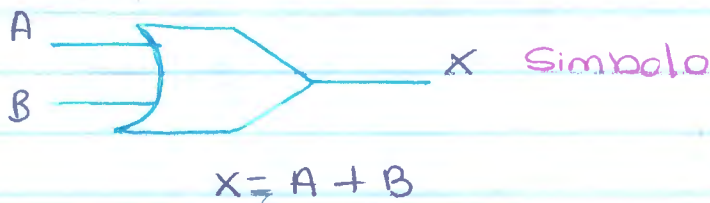
26-enero-10

### And

Nombre	<b>AND</b>	Tabla de verdad	
Característica	Condición	Diagrama de Tiempos	
Símbolo		Circuito eléctrico equivalente	
Expresión Matemática	$S = A \cdot B$ $S = AB$		

### OR (Alternativa)

- Alternativa (opción entre dos cosas, una, otra o ambas)
- La operación Or está relacionada con el término de alternativa y produce un resultado 1, cuando cualquiera de las variables de entrada es 1.
- La operación Or, genera un resultado de 0 sólo cuando todas las variables de entrada son 0.



$A + B$      $A \cup B$      $A \# B$     Expresión matemática

26-Enero-10

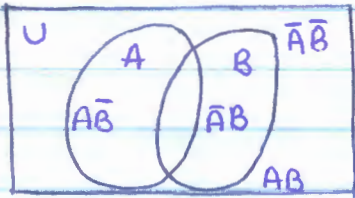


Tabla de Verdad

m	A	B	S
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	1

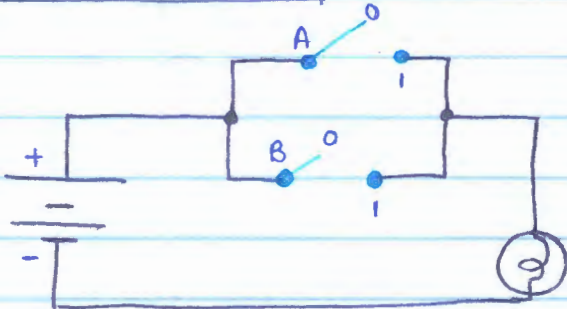
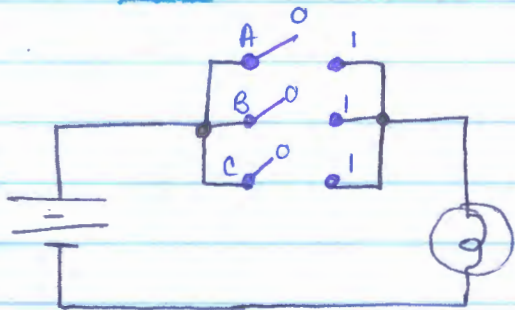


Diagrama de Tiempos

A	0	0	1	1
B	0	1	0	1
Y	0	1	1	1
	0	1	2	3

OR De tres entradas



m	A	B	C	S
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

**OR**

Notación: **OR**
Característica: Alternativa
Tabla de Verdad

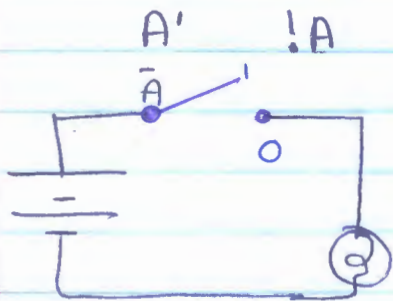
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	1

Expresión Matemática: **S = A + B**
Circuito eléctrico equivalente

Símbolo
Diagrama de Tiempos

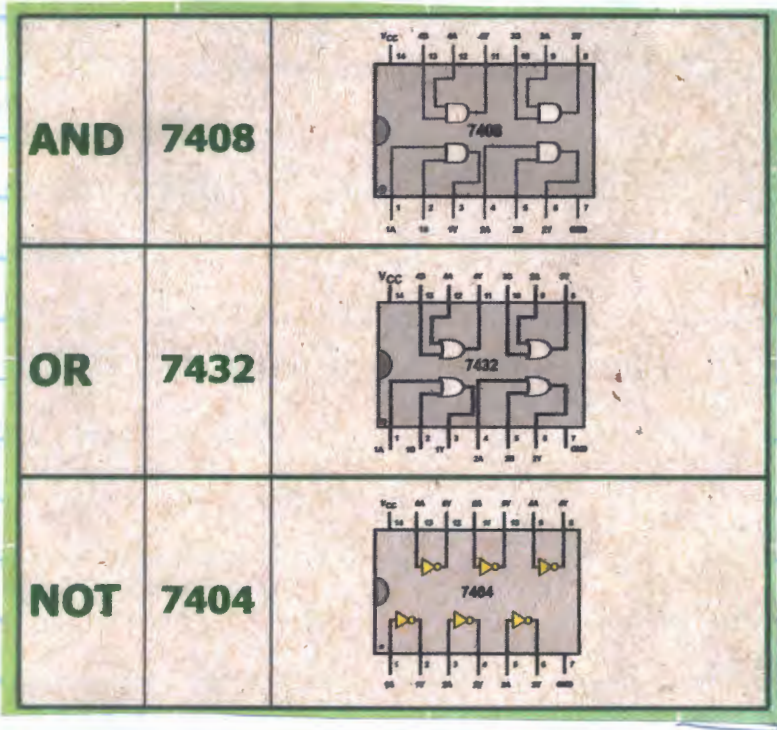
## NOT Negar

La operación Not esta definida para una sola variable y es muy simple ya que solo tiene dos posibilidades si la entrada es cero la salida es igual a uno y viceversa.



m	A	$\bar{A}$
0	0	1
1	1	0





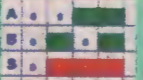

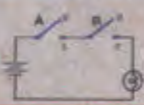
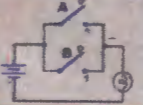
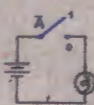
26-Enero-10



		Condición	Alternativa
m	ABC	And	Or
0	000	0	0
1	001	0	1
2	010	0	1
3	011	0	1
4	100	0	1
5	101	0	1
6	110	0	1
7	111	1	1

26-Enero-10

Universidad Autónoma de Nuevo León  
Facultad de Ingeniería Mecánica y Eléctrica

Nombre	AND	OR	NOT																																														
Característica	Condición	Alternativa	Negar																																														
Símbolo																																																	
Expresión Matemática	$S=AB$	$S=A+B$	$S=\bar{A}$																																														
Tabla de Verdad	<table border="1"> <thead> <tr> <th>m</th> <th>A</th> <th>B</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>2</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>3</td> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	m	A	B	S	0	0	0	0	1	0	1	0	2	1	0	0	3	1	1	1	<table border="1"> <thead> <tr> <th>m</th> <th>A</th> <th>B</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>2</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>3</td> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	m	A	B	S	0	0	0	0	1	0	1	1	2	1	0	1	3	1	1	1	<table border="1"> <thead> <tr> <th>A</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	S	0	1	1	0
m	A	B	S																																														
0	0	0	0																																														
1	0	1	0																																														
2	1	0	0																																														
3	1	1	1																																														
m	A	B	S																																														
0	0	0	0																																														
1	0	1	1																																														
2	1	0	1																																														
3	1	1	1																																														
A	S																																																
0	1																																																
1	0																																																
Diagrama de Tiempos																																																	
Circuito eléctrico equivalente																																																	

Sistemas Digitales.

28-Enero-10

### Actividad 3

1. Formule el orden de las conversiones mas convenientes, para convertir un numero en base 10 ( $N_{(10)}$ ) a  $N_{(2)}$ ,  $N_{(8)}$ ,  $N_{(16)}$  y  $N_{(7)}$  indicando el metodo y posteriormente proponga un numero decimal de 3 digitos y compruebe el orden propuesto.

Orden

- $N_{(10)} \rightarrow N_{(2)}$  Residuos
- $N_{(2)} \rightarrow N_{(8)}$  Multiplo
- $N_{(2)} \rightarrow N_{(16)}$  Multiplo
- $N_{(10)} \rightarrow N_{(7)}$  Residuos

•  $172_{(10)} \rightarrow N_{(2)}$  Residuos

$$\begin{array}{r}
 86 \\
 2 \overline{) 172} \\
 \underline{12} \\
 6
 \end{array}
 \quad
 \begin{array}{r}
 43 \\
 2 \overline{) 86} \\
 \underline{06} \\
 0
 \end{array}
 \quad
 \begin{array}{r}
 21 \\
 2 \overline{) 43} \\
 \underline{03} \\
 1
 \end{array}
 \quad
 \begin{array}{r}
 10 \\
 2 \overline{) 21} \\
 \underline{01} \\
 0
 \end{array}
 \quad
 \begin{array}{r}
 5 \\
 2 \overline{) 10} \\
 \underline{0} \\
 0
 \end{array}
 \quad
 \begin{array}{r}
 2 \\
 2 \overline{) 5} \\
 \underline{1} \\
 1
 \end{array}
 \quad
 \begin{array}{r}
 1 \\
 2 \overline{) 2} \\
 \underline{0} \\
 0
 \end{array}
 \quad
 \boxed{10101100_{(2)}}$$

•  $10101100_{(2)} \rightarrow N_{(8)}$  Multiplo

$$\begin{array}{cccccc}
 1 & 0 & 10 & 11 & 00 & \\
 \underbrace{\hspace{1.5cm}} & & \underbrace{\hspace{1.5cm}} & & \underbrace{\hspace{1.5cm}} & \\
 \end{array}
 \quad
 \begin{array}{|c|c|c|c|c|c|c|}
 \hline
 2 & 1 & 4 & 2 & 1 & 4 & 2 & 1 \\
 \hline
 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
 \hline
 \end{array}
 \quad
 \begin{array}{l}
 = 254_{(8)} \\
 2 \quad 5 \quad 4
 \end{array}$$

•  $10101100_{(2)} \rightarrow N_{(16)}$  Multiplo

$$\begin{array}{cccc}
 10 & 10 & 11 & 00 \\
 \underbrace{\hspace{1.5cm}} & & \underbrace{\hspace{1.5cm}} & \\
 \end{array}
 \quad
 \begin{array}{|c|c|c|c|c|c|c|}
 \hline
 8 & 4 & 2 & 1 & 8 & 4 & 2 & 1 \\
 \hline
 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
 \hline
 \end{array}
 \quad
 \begin{array}{l}
 = AC_{(16)} \\
 10 = A \quad 12 = C
 \end{array}$$

•  $172_{(10)} \rightarrow N_{(7)}$  Residuos

$$\begin{array}{r}
 24 \\
 7 \overline{) 172} \\
 \underline{32} \\
 4
 \end{array}
 \quad
 \begin{array}{r}
 3 \\
 7 \overline{) 24} \\
 \underline{3} \\
 0
 \end{array}
 \quad
 = \boxed{334_{(7)}}$$

28-enero-10

2. Formule el orden de las conversiones mas convenientes para convertir un numero en base 10  $N_{(10)}$  a  $N_{(2)}$ ,  $N_{(8)}$ ,  $N_{(16)}$  y  $N_{(9)}$  indicando el metodo y posteriormente proponga un numero Hexadecimal de 3 digitos y compruebe el orden propuesto.

Orden

- $N_{(16)} \rightarrow N_{(2)}$  Multiplo
- $N_{(2)} \rightarrow N_{(8)}$  Multiplo
- $N_{(8)} \rightarrow N_{(10)}$  Multiplicar por la base y sumar
- $N_{(10)} \rightarrow N_{(9)}$  Residuos

•  $6DC_{(16)} \rightarrow N_{(2)}$  Multiplo

6			D=13			C=12		
0	1	1	0	1	1	0	1	1
0	1	1	0	1	1	0	0	0

$2^3$	$2^2$	$2^1$	$2^0$
8	4	2	1

=  $1101101100_{(2)}$

•  $1101101100_{(2)} \rightarrow N_{(8)}$  Multiplo

$1101101100_{(2)}$

2	1	4	2	1	4	2	1	4	2	1
1	1	0	1	1	0	1	1	1	0	0
3		3		3		3		4		

=  $3334_{(8)}$

•  $3334_{(8)} \rightarrow N_{(10)}$  Multiplicar por base y sumar

$$\begin{array}{r}
 3 \quad 3 \quad 3 \quad 4 \\
 24 \quad 27 \quad 219 \quad 1756 \\
 \hline
 1756
 \end{array}$$

=  $1756_{(10)}$

•  $1756_{(10)} \rightarrow N_{(9)}$

$$\begin{array}{r}
 175 \\
 10 \overline{) 1756} \\
 \underline{075} \phantom{0} \\
 056 \phantom{0} \\
 \underline{06} \phantom{0} \\
 0
 \end{array}$$

$$\begin{array}{r}
 17 \\
 10 \overline{) 175} \\
 \underline{075} \phantom{0} \\
 05
 \end{array}$$

$$\begin{array}{r}
 1 \\
 10 \overline{) 17} \\
 \underline{07} \phantom{0} \\
 0
 \end{array}$$

=  $1756_{(9)}$



28-Enero-10

3. Formule el orden de las conversiones más convenientes para convertir un número en base 5 ( $N_{(5)}$ ) a  $N_{(2)}$ ,  $N_{(8)}$ ,  $N_{(10)}$  y  $N_{(16)}$  indicando el método y posteriormente proponga un número en base cinco de 3 dígitos y compruebe el orden propuesto.

Orden

- $N_{(5)} \rightarrow N_{(10)}$  Multiplicar por la base y sumar
- $N_{(10)} \rightarrow N_{(2)}$  Residuos
- $N_{(2)} \rightarrow N_{(8)}$  Múltiplo
- $N_{(2)} \rightarrow N_{(16)}$  Múltiplo

•  $421_{(5)} \rightarrow N_{(10)}$  Multiplicar por la base y sumar

4	2	1	=	$45_{(10)}$
20	22	44		

•  $45_{(10)} \rightarrow N_{(2)}$  Residuos

$\begin{array}{r} 22 \\ 2 \overline{)45} \\ \underline{05} \\ 1 \end{array}$	$\begin{array}{r} 11 \\ 2 \overline{)22} \\ \underline{02} \\ 0 \end{array}$	$\begin{array}{r} 5 \\ 2 \overline{)11} \\ \underline{1} \end{array}$	$\begin{array}{r} 2 \\ 2 \overline{)5} \\ \underline{1} \end{array}$	$\begin{array}{r} 1 \\ 2 \overline{)2} \\ \underline{0} \end{array}$	=	$101101_{(2)}$
--	--	---	--	--	---	----------------

•  $101101_{(2)} \rightarrow N_{(8)}$  Múltiplo

$\underbrace{101}_{5}$	$\underbrace{101}_{5}$	<table style="display: inline-table; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 2px;">4</td> <td style="border: 1px solid black; padding: 2px;">2</td> <td style="border: 1px solid black; padding: 2px;">1</td> <td style="border: 1px solid black; padding: 2px;">4</td> <td style="border: 1px solid black; padding: 2px;">2</td> <td style="border: 1px solid black; padding: 2px;">1</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">1</td> <td style="border: 1px solid black; padding: 2px;">0</td> <td style="border: 1px solid black; padding: 2px;">1</td> <td style="border: 1px solid black; padding: 2px;">1</td> <td style="border: 1px solid black; padding: 2px;">0</td> <td style="border: 1px solid black; padding: 2px;">1</td> </tr> <tr> <td colspan="2" style="border: none; text-align: center;">5</td> <td colspan="2" style="border: none; text-align: center;">5</td> <td colspan="2" style="border: none;"></td> </tr> </table>	4	2	1	4	2	1	1	0	1	1	0	1	5		5				=	$55_{(8)}$
4	2	1	4	2	1																	
1	0	1	1	0	1																	
5		5																				

•  $101101_{(2)} \rightarrow N_{(16)}$

$\underbrace{101}_{2}$	$\underbrace{101}_{13=0}$	<table style="display: inline-table; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 2px;">2</td> <td style="border: 1px solid black; padding: 2px;">1</td> <td style="border: 1px solid black; padding: 2px;">8</td> <td style="border: 1px solid black; padding: 2px;">4</td> <td style="border: 1px solid black; padding: 2px;">2</td> <td style="border: 1px solid black; padding: 2px;">1</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">1</td> <td style="border: 1px solid black; padding: 2px;">0</td> <td style="border: 1px solid black; padding: 2px;">1</td> <td style="border: 1px solid black; padding: 2px;">1</td> <td style="border: 1px solid black; padding: 2px;">0</td> <td style="border: 1px solid black; padding: 2px;">1</td> </tr> <tr> <td colspan="2" style="border: none; text-align: center;">2</td> <td colspan="2" style="border: none; text-align: center;">13=0</td> <td colspan="2" style="border: none;"></td> </tr> </table>	2	1	8	4	2	1	1	0	1	1	0	1	2		13=0				=	$2D_{(16)}$
2	1	8	4	2	1																	
1	0	1	1	0	1																	
2		13=0																				

29-enero-10

## Ejercicio 1

1. La salida es cero cuando cualquier entrada es igual a cero

And

2. Cualquier entrada uno produce una salida uno

Or

3. Solamente cuando todas las entradas son cero producen una salida cero

Or

4. La salida es uno solamente cuando todas las entradas son uno

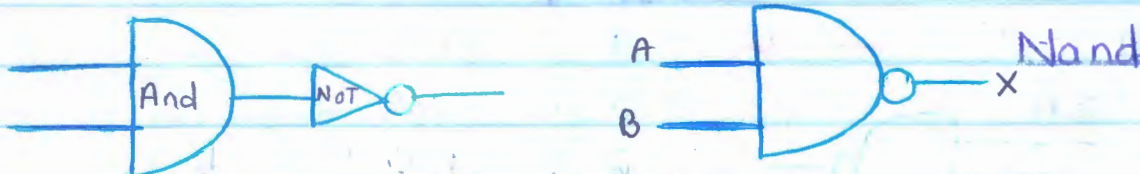
And

5. La salida es siempre lo contrario a la entrada.

Not

## Nand

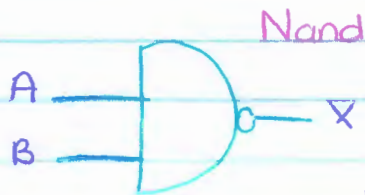
La operación nand es el negado de la salida de la operación And.



$$X = \overline{A * B}$$

29-Enero-10

La operación Nand es el negado de las entradas de la operación OR



Nand

$$X = \overline{A * B}$$

$$x = \overline{\overline{A} + \overline{B}}$$

Tabla de verdad



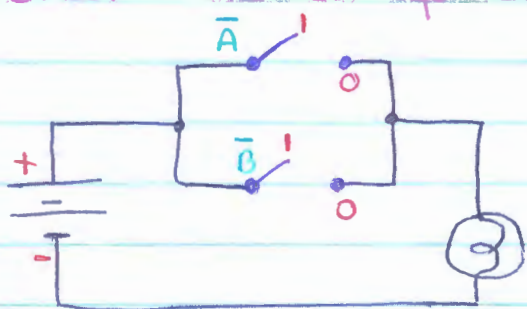
$$X = \overline{A * B}$$

$$X = \overline{!(A \& B)}$$

m	A B	$\overline{A B}$
0	0 0	1
1	0 1	1
2	1 0	1
3	1 1	0

Nand

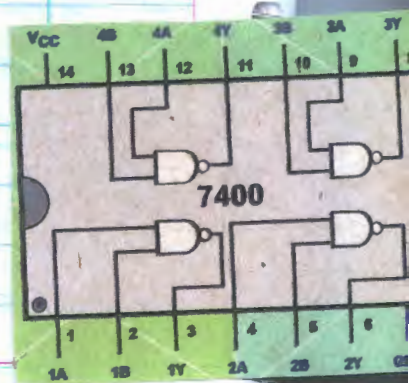
Circuito electrico equivalente



$$F(A,B,C) = \overline{A B C}$$

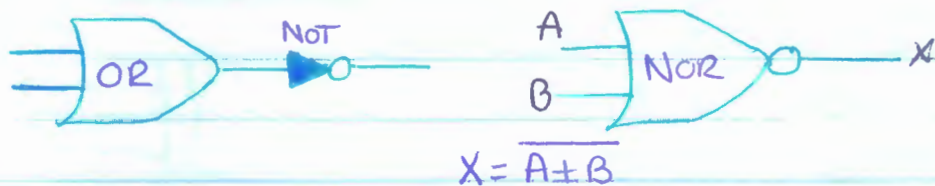
m	A B C	$\overline{A B C}$
0	0 0 0	1
1	0 0 1	1
2	0 1 0	1
3	0 1 1	1
4	1 0 0	1
5	1 0 1	1
6	1 1 0	1
7	1 1 1	0

Nand de 3 entradas

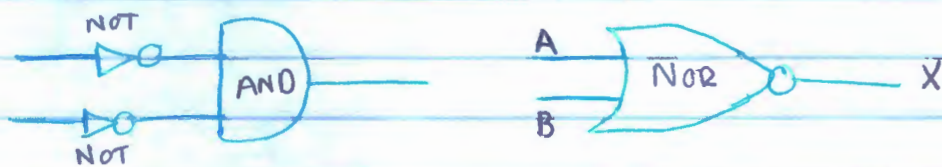


# NOR

La operación Nor es el negado de la salida de la operación OR.



La operación Nor es el negado de las entradas de la operación AND.

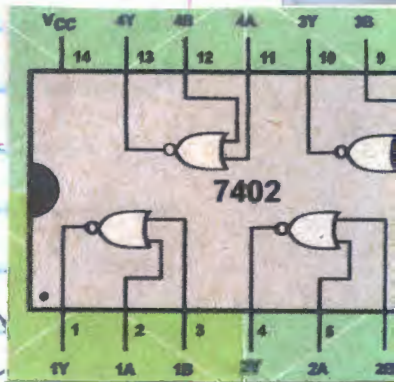
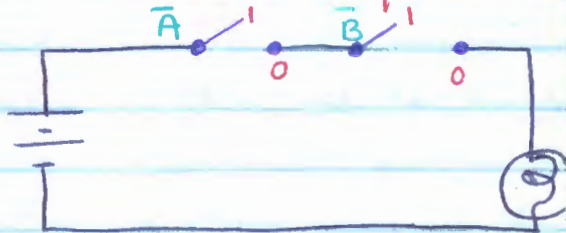


$X = \overline{A \cdot B}$     $\overline{A+B}$     $\sim$     $X = \overline{A+B}$     $X = \overline{!(A \neq B)}$

## Tabla de verdad

m	AB	$\overline{A+B}$
0	0 0	1
1	0 1	0
2	1 0	0
3	1 1	0

## Circuito eléctrico equivalente



## Nor de tres entradas

$F(A, B, C) = \overline{A+B+C}$



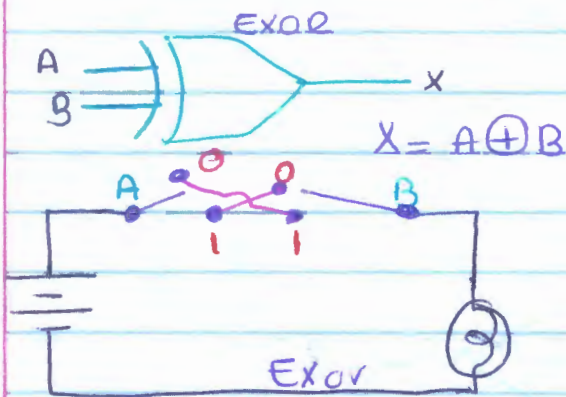
m	ABC	$\overline{A+B+C}$
0	0 0 0	1
1	0 0 1	0
2	0 1 0	0
3	0 1 1	0
4	1 0 0	0
5	1 0 1	0
6	1 1 0	0
7	1 1 1	0

29-enero-10

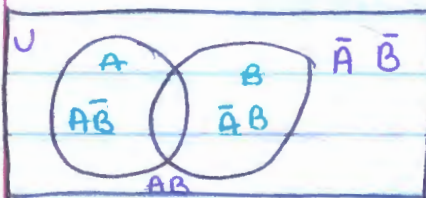
## EXOR (Alternativa, Exclusiva)

(opción entre dos cosas, una, otra pero no ambas).

La operación Exor produce un resultado 1, cuando un número impar de variables de entrada valen 1.

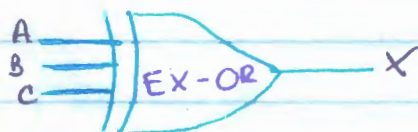


m	A	B	$X = A \oplus B$
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0



Exor, produce un resultado 1, cuando un número impar de variables de entrada valen 1.

m	A	B	C	X
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

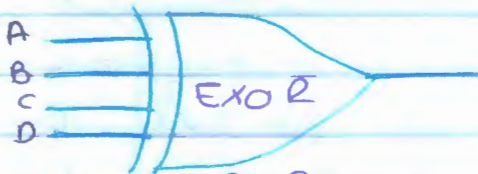


$$X = A \oplus B \oplus C$$

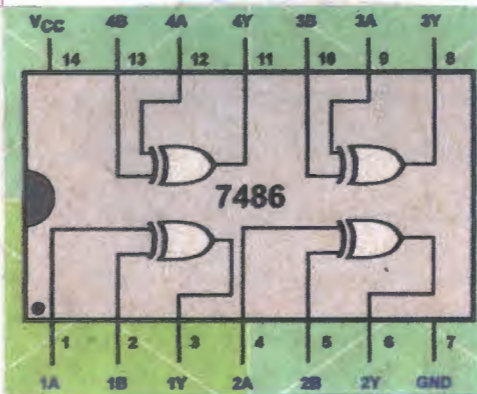
29-Enero-10

### Exor

Produce un resultado 1, cuando un numero impar de variables de entrada valen 1.



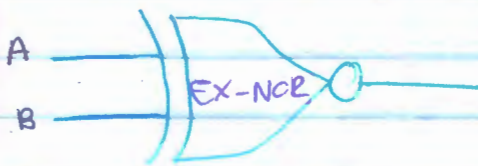
$$X = A \oplus B \oplus C \oplus D$$



m	A	B	C	D	X
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	0

### EXNOR

La operación Exnor es el negado de la salida de la operación Exor



$$\overline{A \oplus B}$$

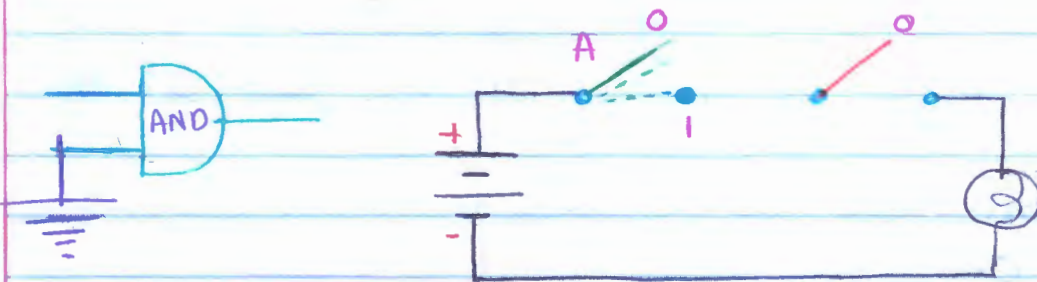
m	Entradas		Salida
	A	B	S
0	0	0	1
1	0	1	0
2	1	0	0
3	1	1	1

29-Enero-10

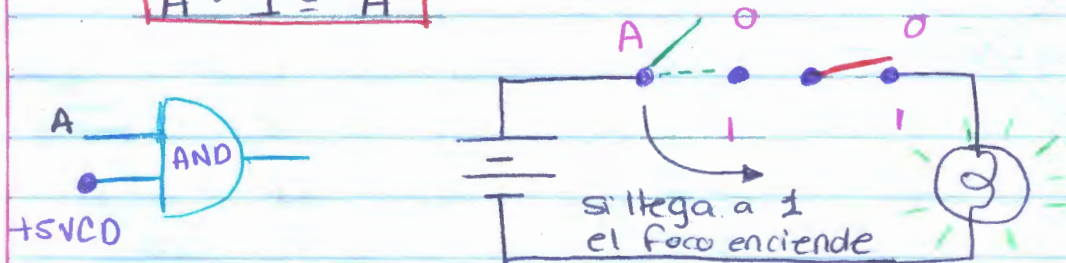
m	A B C	Condición	Alternativa	Impar	Negado de And	Negado de Exor	Negado de Or
		And	Or	Exor	Nand	Ex-Nor	Nor
0	0 0 0	0	0	0	1	1	1
1	0 0 1	0	1	1	1	0	0
2	0 1 0	0	1	1	1	0	0
3	0 1 1	0	1	0	1	1	0
4	1 0 0	0	1	1	1	0	0
5	1 0 1	0	1	0	1	1	0
6	1 1 0	0	1	0	1	1	0
7	1 1 1	1	1	1	0	0	0

## Leyes y teoremas del algebra Booleana

$$A \cdot 0 = 0$$

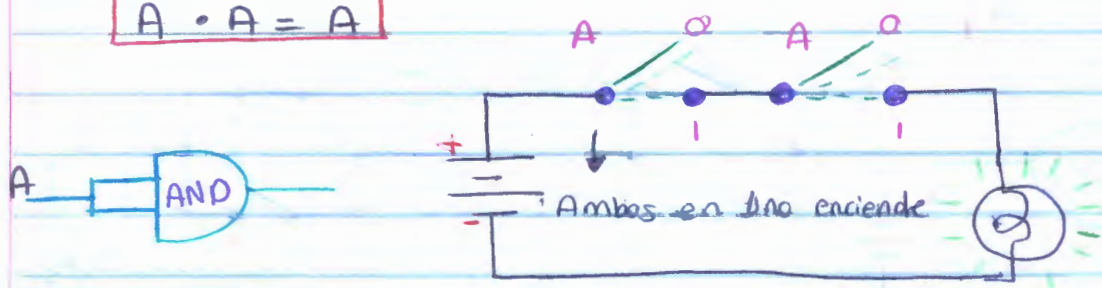


$$A \cdot 1 = A$$

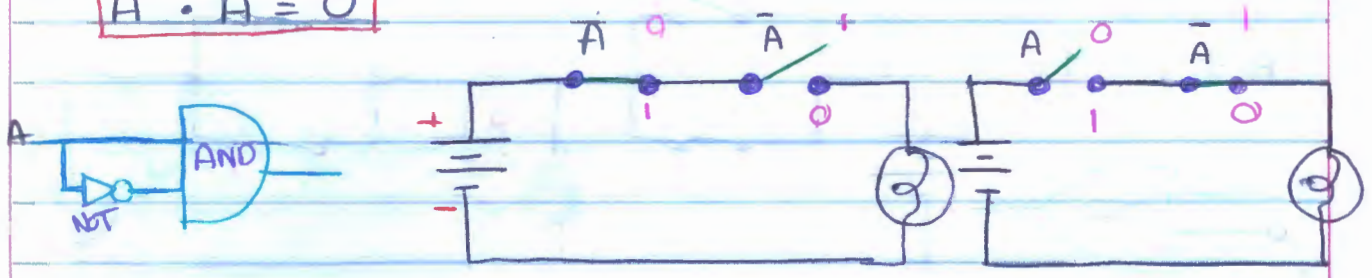


29-Enero-10

$A \cdot A = A$



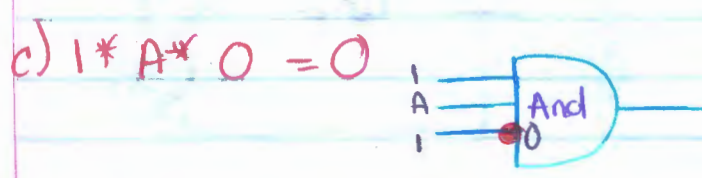
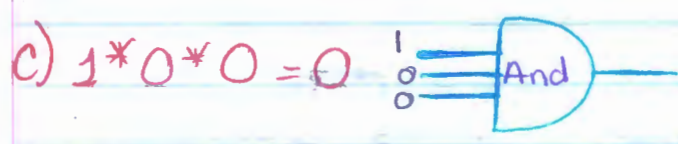
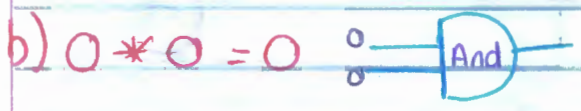
$A \cdot \bar{A} = 0$



Evaluar las siguiente operación



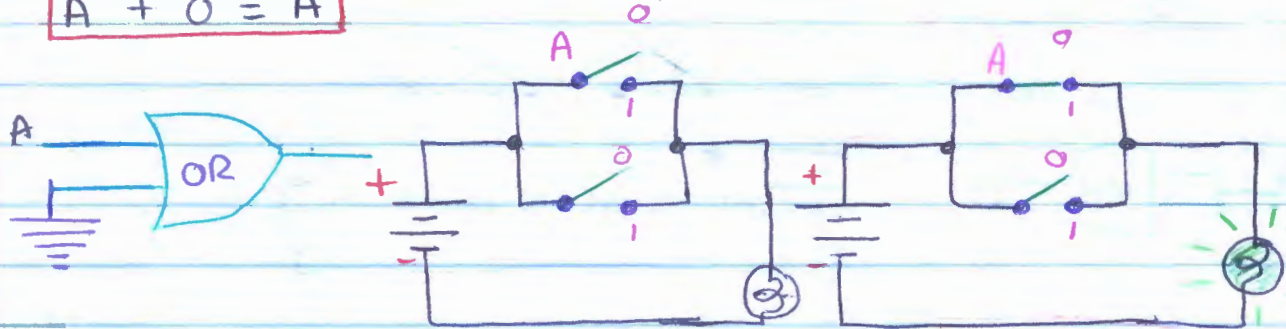
Un and con cualquier cero la salida es cero.



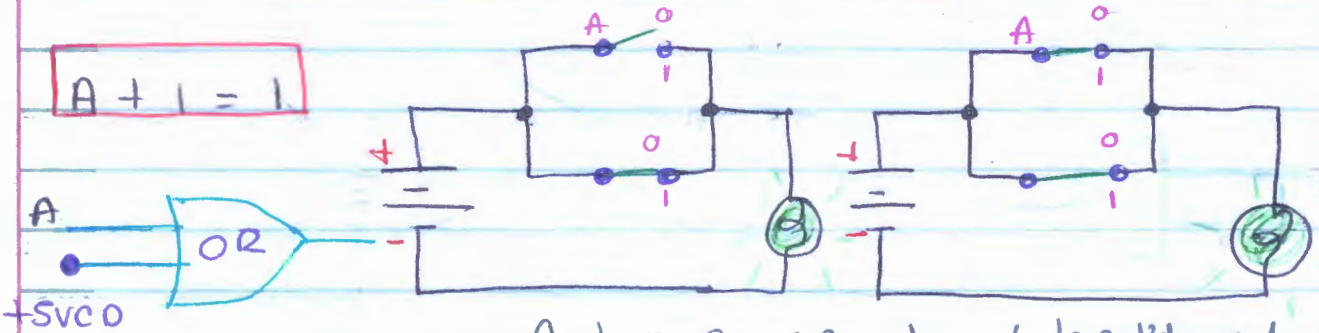


29-Enero-10

$$A + 0 = A$$

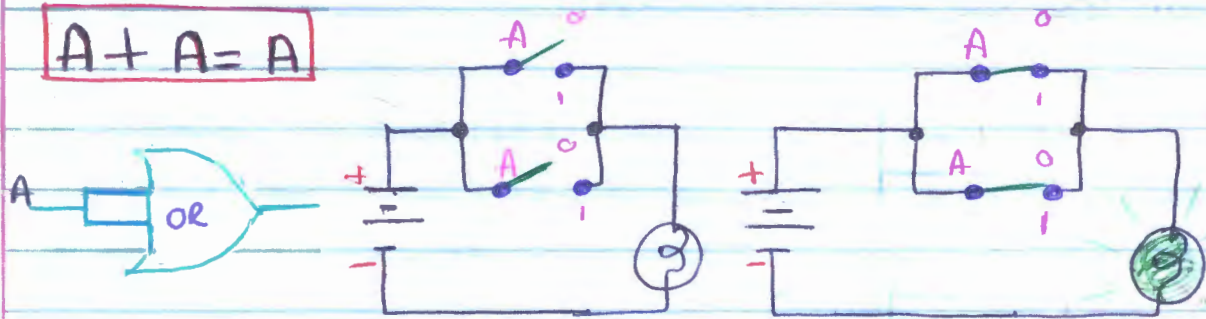


$$A + 1 = 1$$



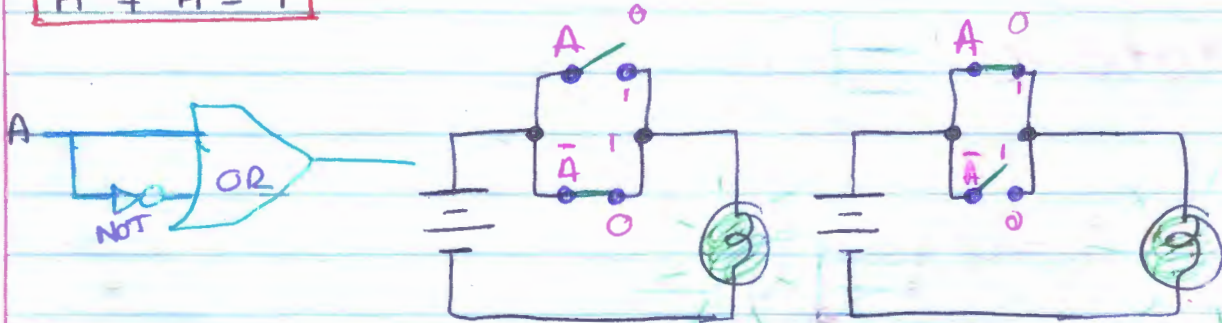
Cualquier Or con cualquier 1 la salida es 1.

$$A + A = A$$



Depende del valor de A  
Para encender por eso es = A.

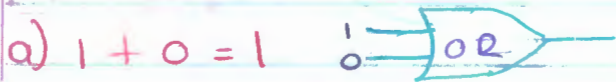
$$A + \bar{A} = 1$$



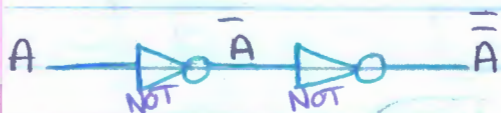
Si siempre esta encendido es = a 1 aki pk siempre hay uno encendido.

29-Enero-10

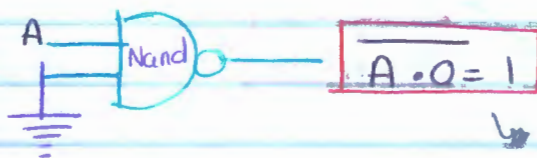
Evaluar la siguiente operación



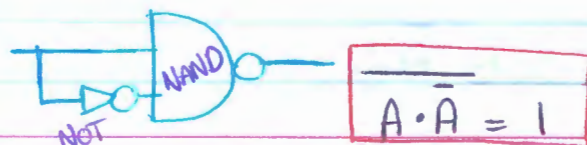
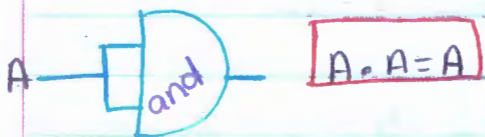
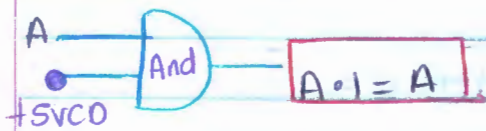
$$\overline{\overline{A}} = A$$



And y Nand

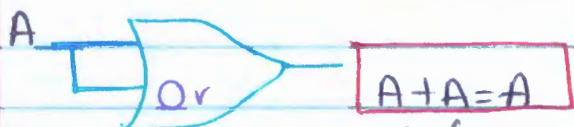
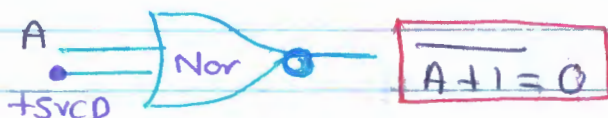
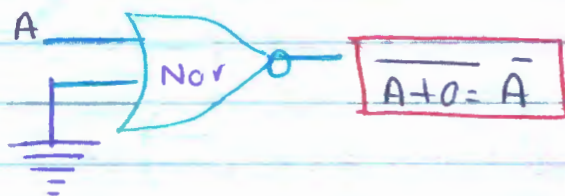
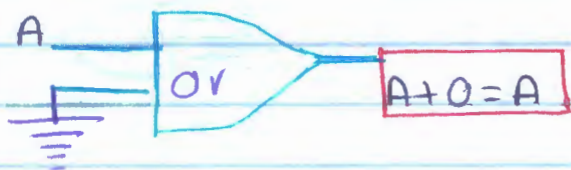


↳ Da 1 ok es negada

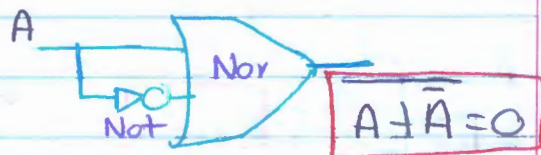
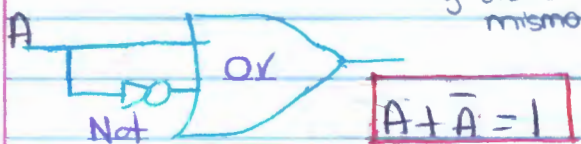


29-Enero-10

## Or y Nor



Los dos se abren y cierran al mismo tiempo.



Resuelva las siguientes proposiciones TAREA

1.  $A \oplus 0 = A$

5.  $\overline{A \oplus 0} = \bar{A}$

2.  $A \oplus 1 = \bar{A}$

6.  $\overline{A \oplus 1} = A$

3.  $A \oplus A = 0$

7.  $\overline{A \oplus A} = 1$

4.  $A \oplus \bar{A} = 1$

8.  $\overline{A \oplus \bar{A}} = 0$

29-Enero-10

## Propiedades

- Conmutativa
- Asociativa
- Distributiva

### Conmutativa

$$AB \stackrel{\text{And}}{=} BA$$



### Conmutativa

Or

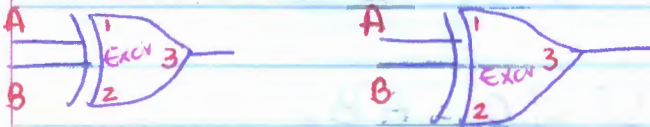
$$A+B = B+A$$



### Conmutativa

Exor

$$A \oplus B = B \oplus A$$



### Conmutativa

And  $AB = BA$

Or  $A+B = B+A$

Exor  $A \oplus B = B \oplus A$

Nand  $\overline{AB} = \overline{BA}$

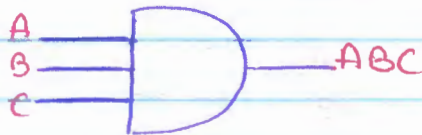
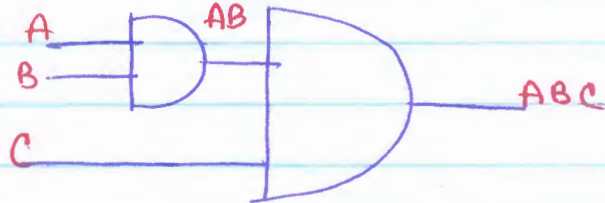
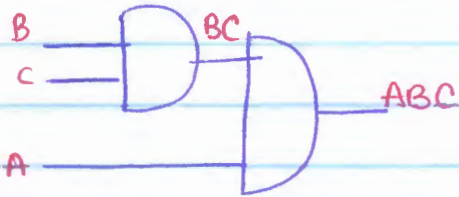
Nor  $\overline{A+B} = \overline{B+A}$

Exnor  $\overline{A \oplus B} = \overline{B \oplus A}$

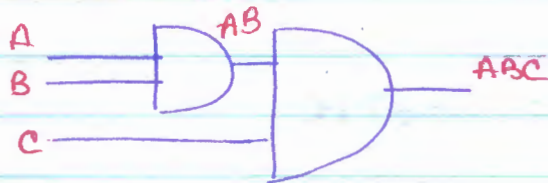
29-enero-10

### Asociativa

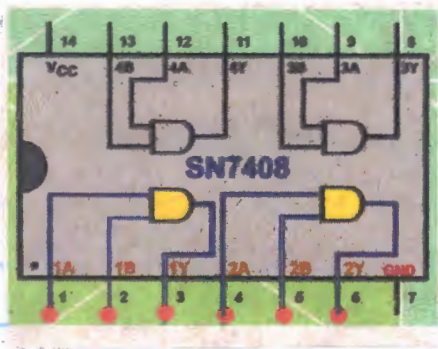
$$\text{And } A(BC) = (AB)C = ABC$$



### Asociativa



$$(AB)C = ABC$$



29-Enero-10

### Asociativa

Es valida para todas las positivas.

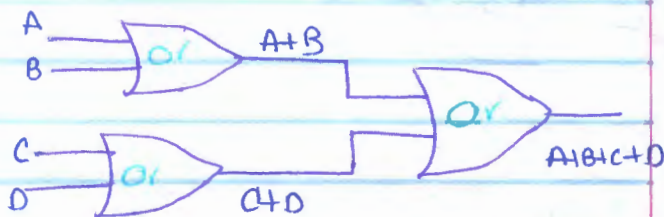
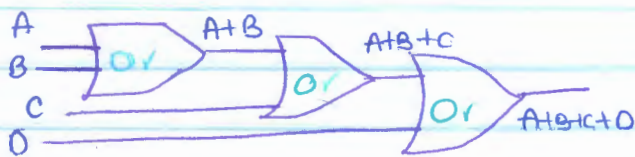
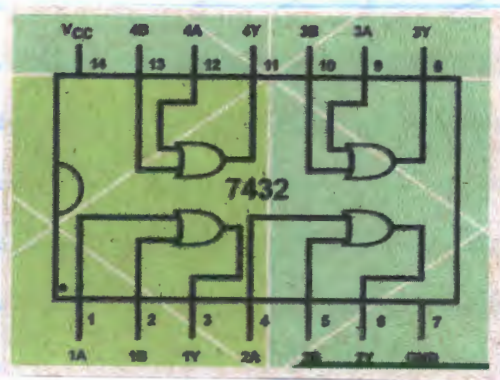
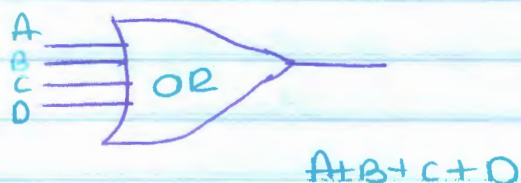
And  $A(BC) = (AB)C = ABC$

Or  $A+(B+C) = (A+B)+C = A+B+C$

Exor  $A\oplus(B\oplus C) = (A\oplus B)\oplus C = A\oplus B\oplus C$

### Asociativa

Or  $A+B+C+D$



Or  $(A+B)+C+D = (A+B)+(C+D)$

### Asociativa

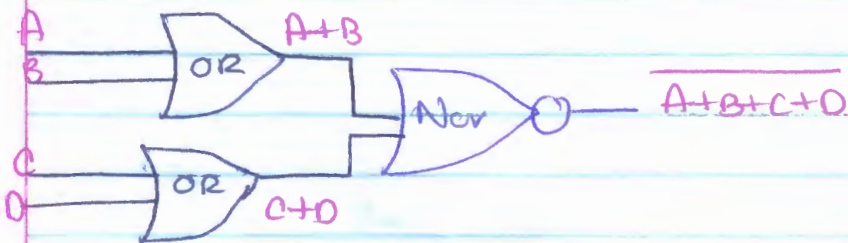
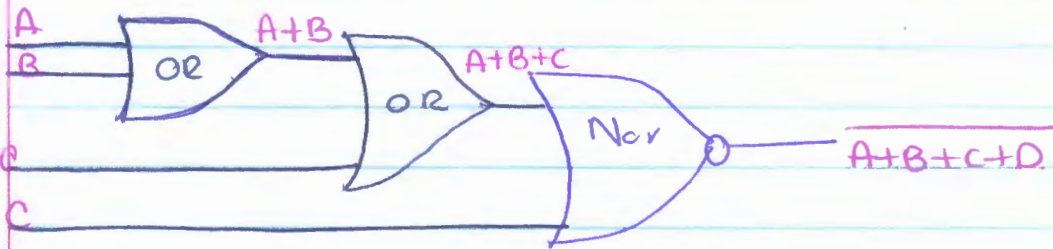
Nand  $[A(BC)'] \neq [(AB)'C]' \neq (ABC)'$

Nor  $[A+(B+C)'] \neq [(A+B)'+C]' \neq (A+B+C)'$

Exor  $[A\oplus(B\oplus C)]' \neq [(A\oplus B)\oplus C]' \neq (A\oplus B\oplus C)'$

29-enero-10

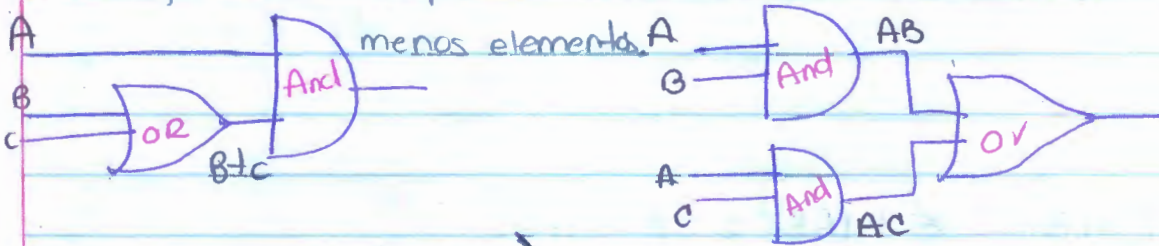
### Asociativa



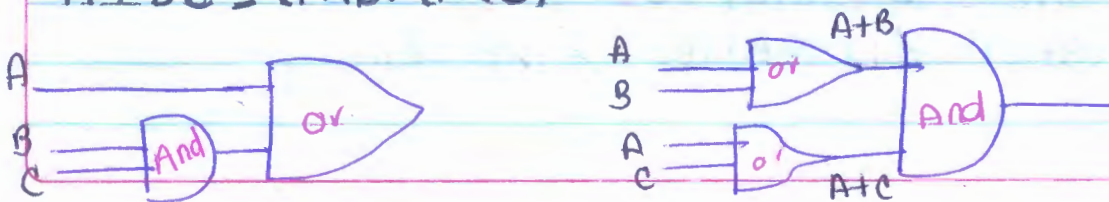
### Distributiva

$$A(B+C) = AB + AC$$

Es mejor usar este pk + ene



$$A+BC = (A+B)(A+C)$$



29-enero-10

$$A + BC = (A+B)(A+C)$$

$$\boxed{AA} + Ac + AB + BC$$

$$A + Ac + AB + BC$$

$$A + \underline{Ac} + \underline{AB} + BC$$

$$A(\underline{1+C+B}) + BC$$

$$A * 1 + BC$$

$$\boxed{A + BC = A + BC}$$

$$AA + Ac + AB + BC$$

$$A + Ac$$

$$A(1+C) + AB + BC$$

$$A + AB + BC$$

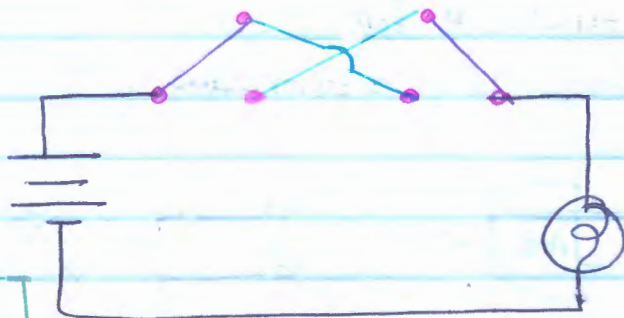
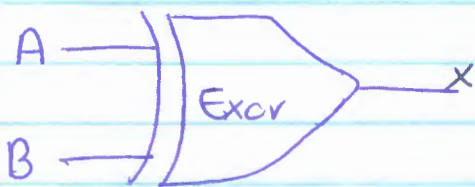
Resuelva las siguientes proposiciones

1.  $A \oplus 0 = A$

2.  $A \oplus 1 = \bar{A}$

3.  $A \oplus A = 0$

4.  $A \oplus \bar{A} = 1$



$$x = A \oplus B$$

m	A	B	$x = A \oplus B$
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0



3- Febrero-09

## Teorema de D' Morgan

Un adivinanza

Augustus de Morgan (?-1871) fue un matemático inglés nacido en la India.

Acostumbraba a recrearse en el planteamiento de adivinanzas y problemas ingeniosos.

Este personaje nacido en el siglo XIX, planteaba esta adivinanza sobre su edad.

"El año  $x^2$  tenía  $x$  años. ¿En qué año nació?"

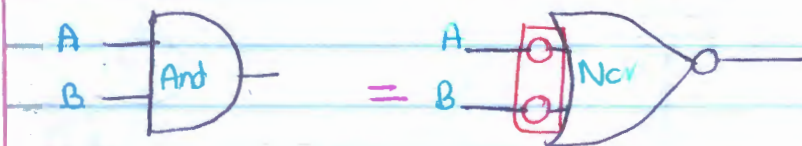
August de Morgan  $42^2 = 1764$   $184 - 43 = \underline{1806}$   
? - 187  $43^2 = 1849$

Por medio de este teorema se obtiene el **Equivalente** de una función Booleana.

## Teorema de D' Morgan

And  $\rightarrow$  Nor

Negando las entradas



$$A \cdot B = \overline{\bar{A} + \bar{B}}$$

9-Feb-10

m	A	B	A · B	$\overline{A+B}$
0	0	0	0	0
1	0	1	0	0
2	1	0	0	0
3	1	1	1	1

Nor  $\rightarrow$  And

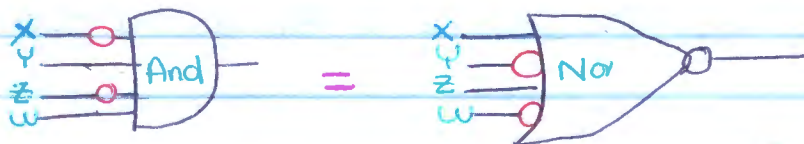
Negando las entradas



$$\overline{A+B} = \overline{A} \cdot \overline{B} \quad \boxed{\overline{A+B}}$$

And  $\rightarrow$  Nor

Negando las entradas



$$x' y' z' w = \overline{x + y' + z + w'} \quad x y z w$$

Or  $\rightarrow$  Nand

Negando las entradas



$$A+B = \overline{\overline{A} \cdot \overline{B}}$$

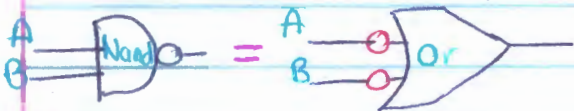
$\overline{A \cdot B}$

A+B

9-Febrero-10

Nand  $\rightarrow$  Or

Negando las entradas



$$\overline{A \cdot B} = \bar{A} + \bar{B}$$

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

Teorema de D' Morgan

And  $\rightarrow$  Nor

Or  $\rightarrow$  Nand

Negando las entradas

Negando las entradas

Nor  $\rightarrow$  And

Nand  $\rightarrow$  Or

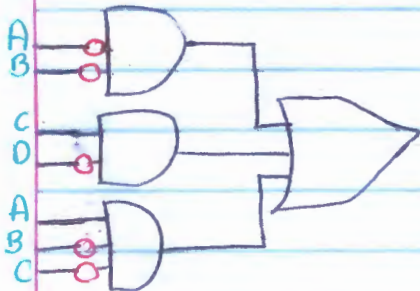
Negando las entradas

Negando las entradas

### Actividad

Obtenga el circuito equivalente aplicando el teorema de D' Morgan y compare el resultado usando la tabla de verdad

$$F(A, B, C, D) = \bar{A}\bar{B} + C\bar{D} + A\bar{B}\bar{C}$$



And  $\rightarrow$  Nor

Or  $\rightarrow$  Nand

Negando las entradas

Negando las entradas

Nor  $\rightarrow$  And

Nand  $\rightarrow$  Or

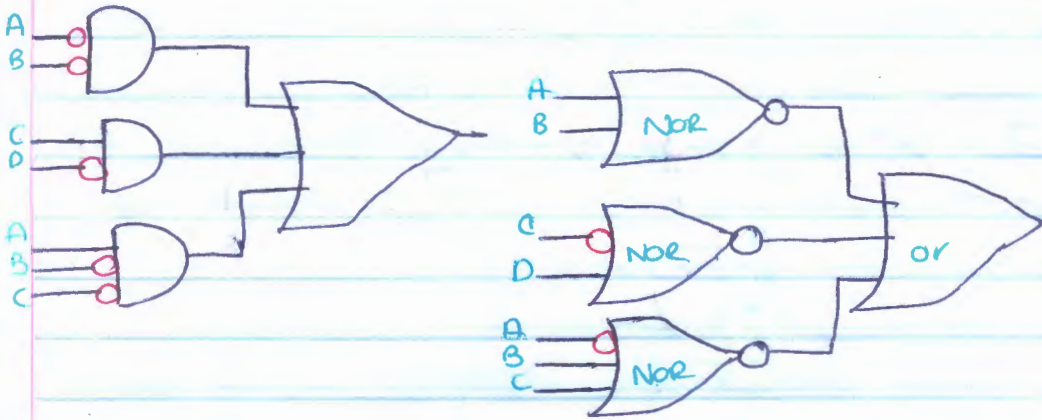
Negando las entradas

Negando las entradas

9-Feb-10

$$F(A, B, C, D) = \bar{A}\bar{B} + C\bar{D} + A\bar{B}\bar{C}$$

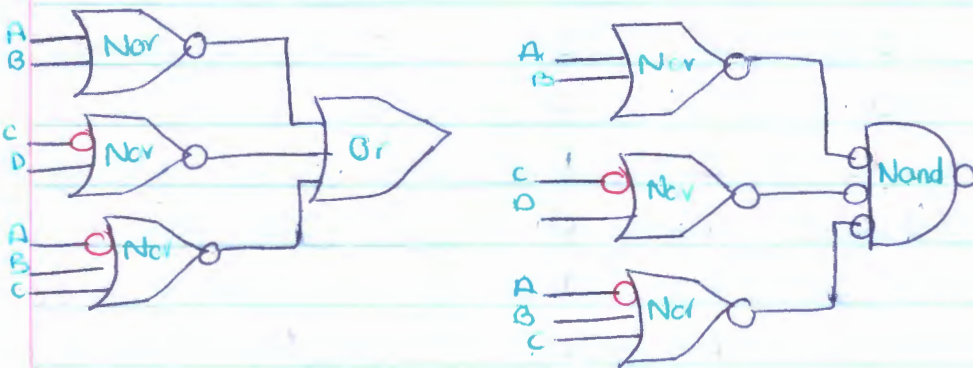
$$F(A, B, C, D) = \overline{(A+B)' + (C'+D)' + (A'+B+C)'} \\ (\overline{A+B}) (\overline{C'+D}) (\overline{A'+B+C})$$



$$F(A, B, C, D) = \bar{A}\bar{B} + C\bar{D} + A\bar{B}\bar{C}$$

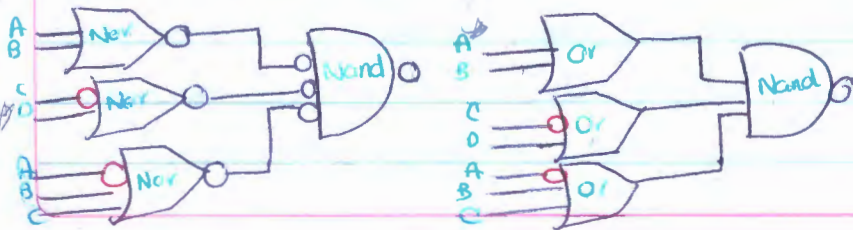
$$F(A, B, C, D) = \overline{[(A+B)' + (C'+D)' + (A'+B+C)']} \rightarrow$$

$$F(A, B, C, D) = \overline{(A+B) (C'+D) (A'+B+C)}$$



$$F(A, B, C, D) = \bar{A}\bar{B} + C\bar{D} + A\bar{B}\bar{C}$$

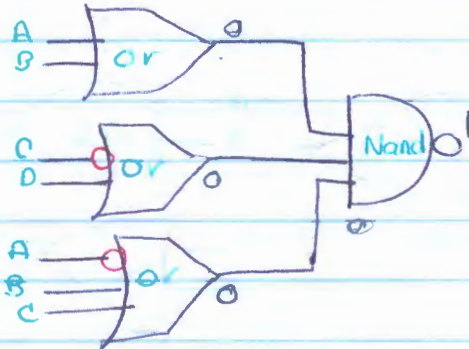
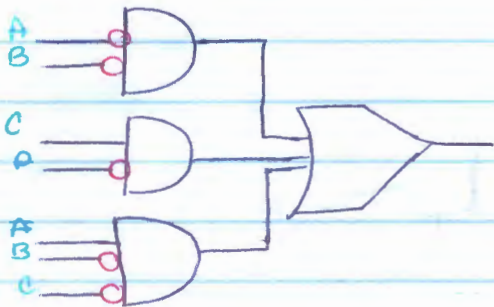
$$F(A, B, C, D) = \overline{[(A+B) (C'+D) (A'+B+C)]'} = \overline{(A+B) (C'+D) (A'+B+C)}$$



9-Feb-10

$$F(A,B,C,D) = \bar{A}\bar{B} + C\bar{D} + A\bar{B}\bar{C}$$

$$F(A,B,C,D) = [(A+B)(C'+D)(A'+B+C)]'$$



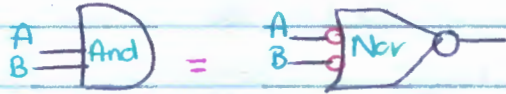
m	A	B	C	D	
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	1
15	1	1	1	1	0

9 Feb-10

# Teorema de D' Morgan

And  $\rightarrow$  Ncr

Negando las entradas



Ncr  $\rightarrow$  And

Negando las entradas



Or  $\rightarrow$  Nand

Negando las entradas



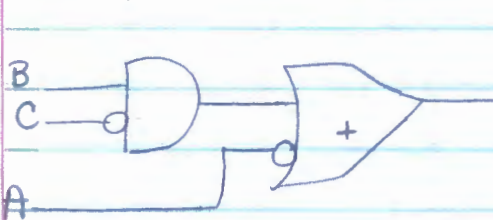
Nand  $\rightarrow$  Or

Negando las entradas



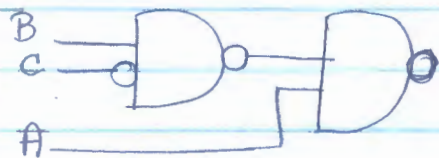
## Ejemplo Salon

$F(A, B, C) = A' + BC'$      $\overline{A + BC}$      $A + \overline{BC}$      $A + \overline{BC}$



$\overline{A + BC}$

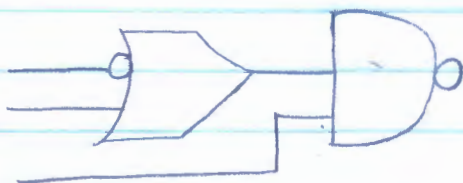
- Or    Proponga un uno
- And   Proponga un cero
- Nand   Proponga un uno
- Ncr    Proponga un cero



$(\overline{A + BC}) \cdot A$



Ejecutar cmd ipconfig /all



Vienes función

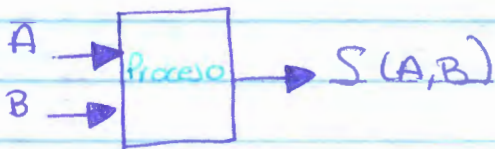
9-Feb-10

## Función

¿Que es una función?

La función es en esencia, un dispositivo de entrada-salida. Es una regla matemática que se asigna a cada valor de entrada un y solo un valor de salida.

Una función la podemos ver como un proceso que recibe valores de entrada y a partir de ellos produce un valor de salida.

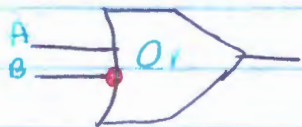


$$F_1(A,B) = A \bar{B}$$



m	AB	$A\bar{B}$
0	00	0
1	01	0
2	10	1
3	11	0

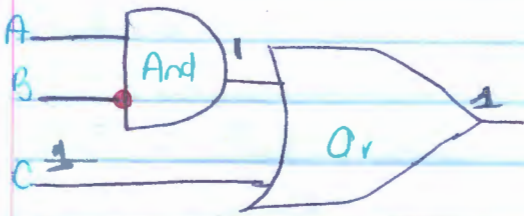
$$F_2(A,B) = A + \bar{B}$$



m	AB	$A + \bar{B}$
0	00	1
1	01	0
2	10	1
3	11	1

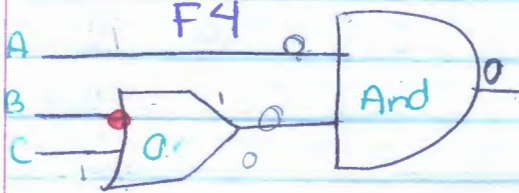
9-Feb-10

F3



$$F_3(A, B, C) = A\bar{B} + C$$

F4



$$F_4(A, B, C) = A(\bar{B} + C)$$

Equivalent or Different?

m	A	B	C	F3	F4
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	0	0
3	0	1	1	1	0
4	1	0	0	1 ✓	1
5	1	0	1	1 ✓	1
6	1	1	0	0	0
7	1	1	1	1	1

$$A\bar{B} + AC$$

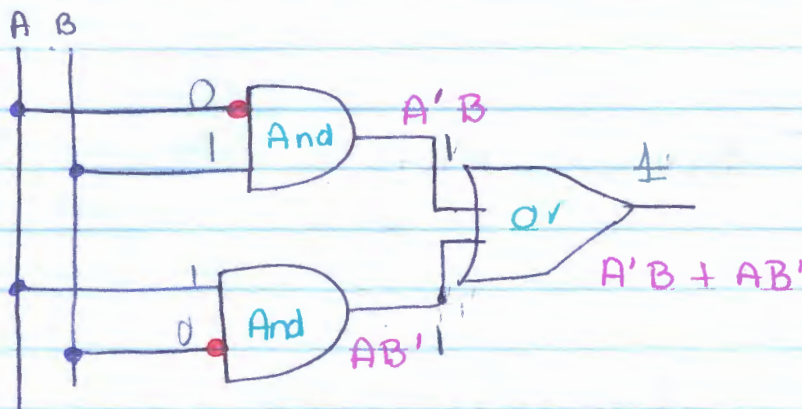
$$A(\bar{B} + C)$$

$$F_3 \neq F_4$$



9-Feb-10

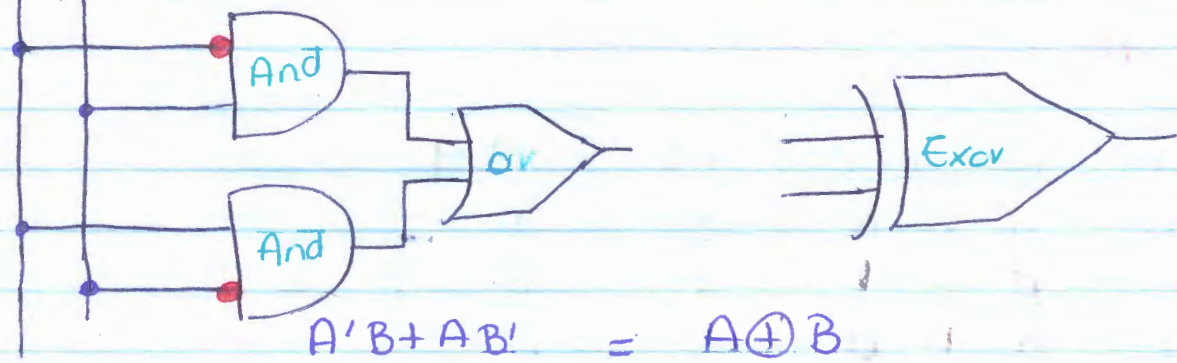
F5



m	A	B	F5
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0

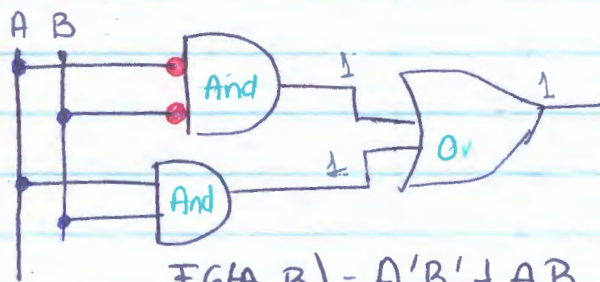
$A'B + AB' = A \oplus B$

A B



$A'B + AB' = A \oplus B$

F6

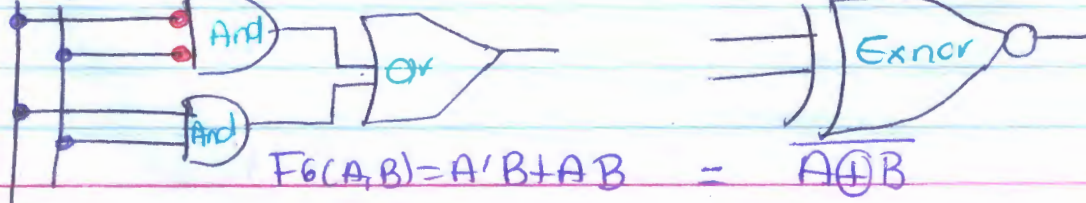


m	A	B	F6
0	0	0	1
1	0	1	0
2	1	0	0
3	1	1	1

$F6(A, B) = A'B' + AB$

$F6(A, B) = A'B + AB = \overline{A \oplus B}$

A B



$F6(A, B) = A'B + AB = \overline{A \oplus B}$

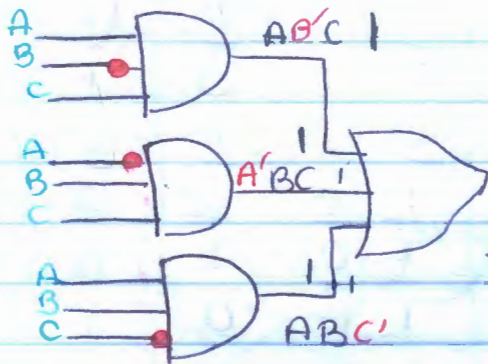
9-Feb-10

F7

Obtenga:

1- La ecuación

2- La tabla de verdad



$$F7(A,B,C) = AB'C + A'BC + ABC'$$

101 011 110

m	A	B	C	F7
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	<u>0</u>	<u>1</u>	<u>1</u>	1
4	<u>1</u>	0	0	0
5	<u>1</u>	<u>0</u>	<u>1</u>	1
6	<u>1</u>	<u>1</u>	0	1
7	<u>1</u>	<u>1</u>	<u>1</u>	0

Saca 3 primers

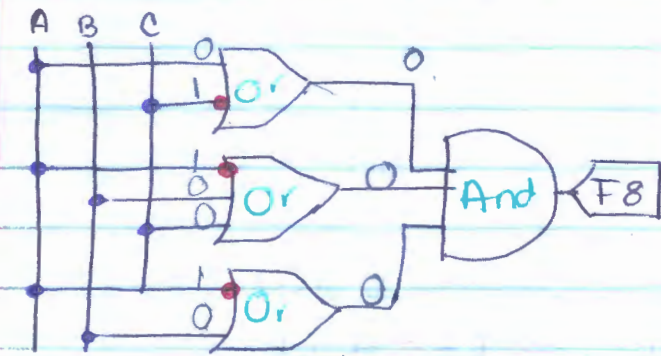
9-Feb-10

### F8

Obtenga:

Saco ceros primero

1. La ecuación
2. La tabla de verdad



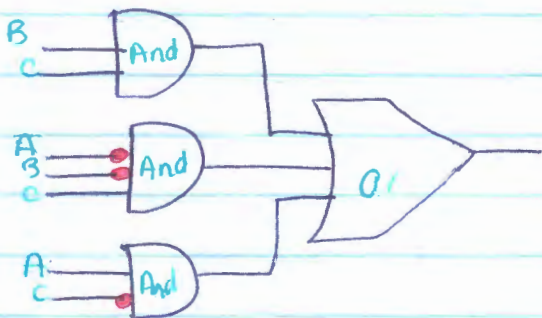
m	A	B	C	F8
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

$$F8(A,B,C) = (A+C')(A'+B+C)(A'+B)$$

### F9

Obtenga:

1. La ecuación
2. La tabla de verdad



m	A	B	C	F9
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

$$F9(A,B,C) = BC + A'B'C + AC'$$

11 001 10

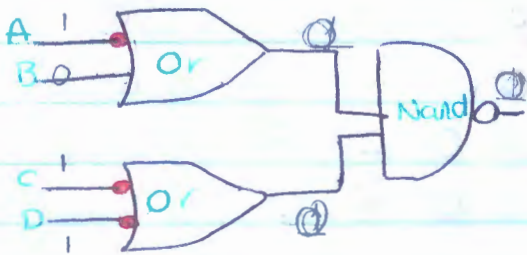
10-Feb-10

F10

Obtenga:

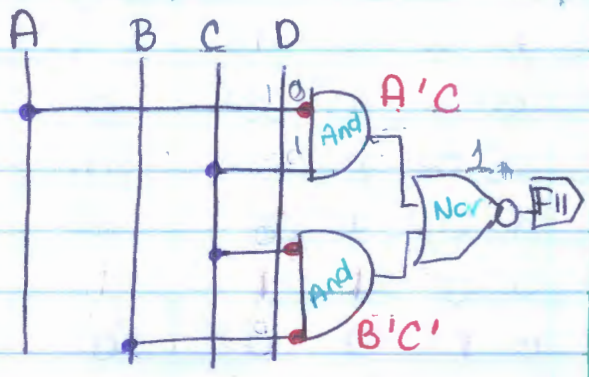
1. El circuito
2. La tabla de verdad

$$F_{10}(A, B, C, D) = (A' + B)(C' + D')$$



m	A	B	C	D	F10
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	1

F11	m	A	B	C	D	F11
Obtenga:	0	0	0	0	0	0
1. El circuito	1	0	0	0	1	0
2. La tabla de verdad	2	0	0	1	0	0
	3	0	0	1	1	0
	4	0	1	0	0	1
	5	0	1	0	1	1
	6	0	1	1	0	0
	7	0	1	1	1	0
	8	1	0	0	0	0
	9	1	0	0	1	0
	10	1	0	1	0	1
	11	1	0	1	1	1
	12	1	1	0	0	1
	13	1	1	0	1	1
	14	1	1	1	0	1
	15	1	1	1	1	1



$F(A,B,C,D) = A'C + B'C'$

Función Booleana

- Elementos de las funciones
  - Circuito, Ecuación y Tabla de verdad
- Jerarquía de las operaciones
- Equivalencias del Exor y Exnor
- Obtener la tabla de verdad a partir del circuito
  - Si termina en Or suponer un uno a la salida
  - Si termina en And suponer un cero a la salida
- Circuito a partir de la ecuación
- Ecuación a partir del circuito


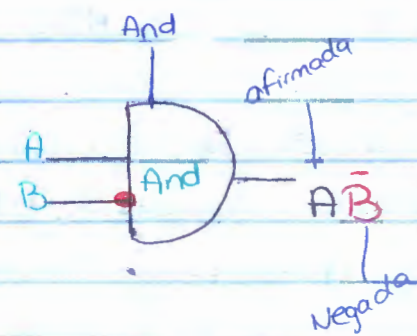
# Miniterminos Captura Esquemática

9-Feb-10

## Minitermino


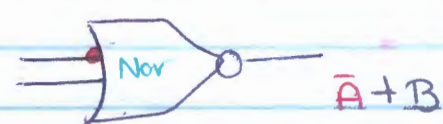
Termino producto (AND) que contiene todas las variables de la función ya sea en su forma normal (afirmada) o complementada (negada).

		Entrada		Salida
m	A	B	S	
0	0	0	0	
1	0	1	0	
2	1	0	1	
3	1	1	0	

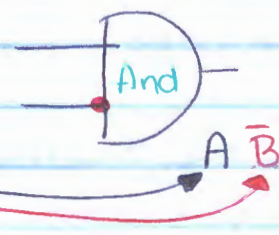



## Minitermino

		Entrada		Salida
m	A	B	S	
0	0	0	0	
1	0	1	0	
2	1	0	1	
3	1	1	0	

		Entrada		Salida
m	A	B	S	
0	0	0	0	
1	0	1	0	
2	1	0	1	
3	1	1	0	



9-Feb-10

### Minitermino

m	Entrada		Salida
	A	B	S
0	0	0	$\bar{A}\bar{B}$
1	0	1	$\bar{A}B$
2	1	0	$A\bar{B}$
3	1	1	$AB$

Con el Minitermino podemos obtener la ecuación a partir de la tabla de verdad

m	Entrada		Salida
	A	B	S
0	0	0	0
1	0	1	1
2	1	0	0
3	1	1	1

SOP

$$F_S(A,B) = \bar{A}B + AB$$

Suma de productos

m	Entradas		Salidas
	A	B	S
0	0	0	0
1	0	1	1
2	1	0	0
3	1	1	1

Forma Canonica

$$F_S(A,B) = m_1 + m_3$$

$$F_S(A,B) = \sum m(1, 3)$$

Minitermino

$$F_S(A,B) = \bar{A}B + AB \text{ SOP}$$

$$F_S(A,B) = \sum m(1, 3) \text{ Forma canonica}$$

9-Feb-10

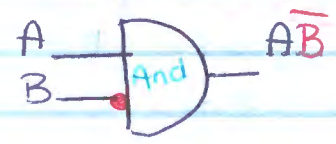
	m	A	B	C	
Minitermino	0	0	0	0	$\bar{A}\bar{B}\bar{C}$
3 variables	1	0	0	1	$\bar{A}\bar{B}C$
	2	0	1	0	$\bar{A}B\bar{C}$
	3	0	1	1	$\bar{A}BC$
	4	1	0	0	$A\bar{B}\bar{C}$
	5	1	0	1	$A\bar{B}C$
	6	1	1	0	$AB\bar{C}$
	7	1	1	1	$ABC$

Tabla

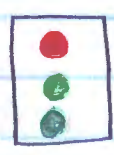
Tabla → ecuación → Circuito

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

$F(A,B) = \sum m(1,3)$



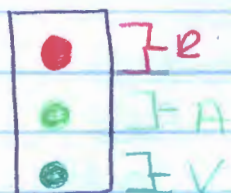
### Detector de errores de un semaforo



Se considera error cuando no enciende ninguna luz o encienden mas de una luz a la vez

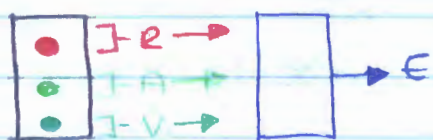


## Detector de errores de un semaforo

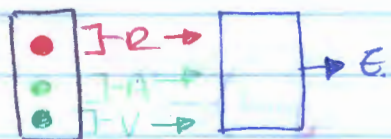


Se colocan fotoceldas para detectar el estado de cada luz de modo que:

Si enciende el foco la fotocelda emite un uno y si el foco esta apagado la fotocelda emite un cero



Cuando el sistema detecte un error la salida  $E=1$



m	R	A	V	E
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Detector de errores de un semaforo

Forma SOP

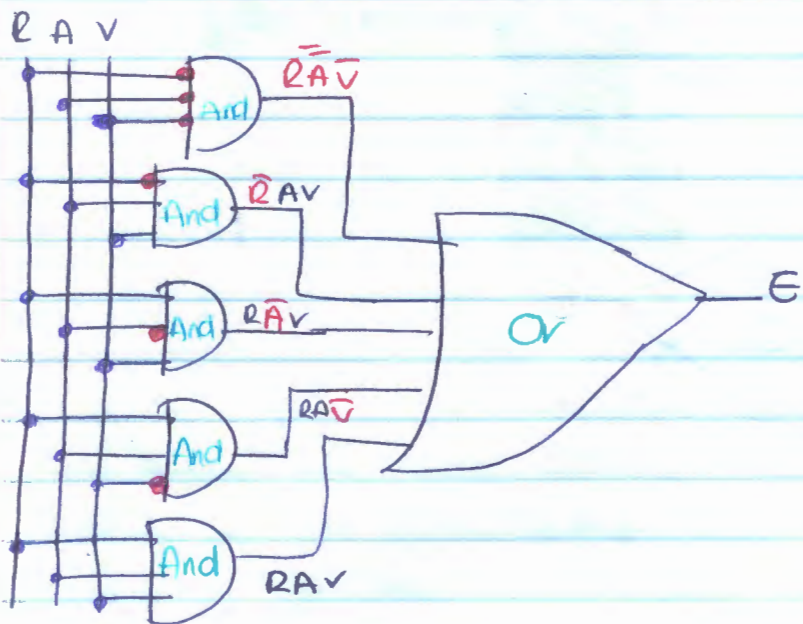
$$F_E(R, A, V) = \bar{R}\bar{A}\bar{V} + \bar{R}AV + R\bar{A}V + RA\bar{V} + RAV$$

Forma Canonica

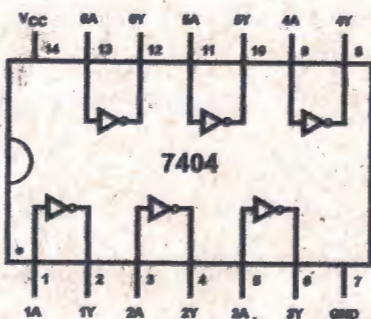
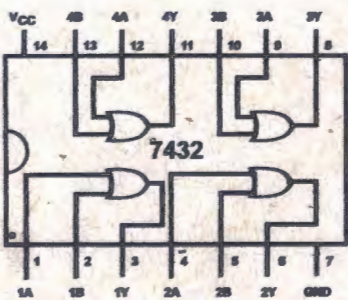
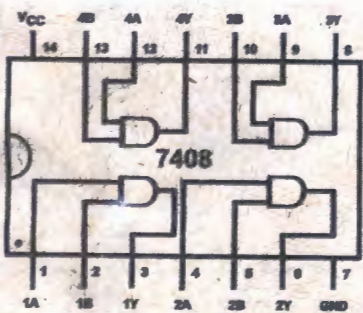
$$F_E(R, A, V) = \sum m(0, 3, 5, 6, 7)$$

9-Feb-10

### Circuito



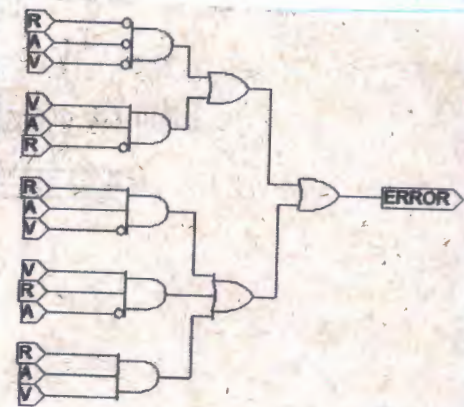
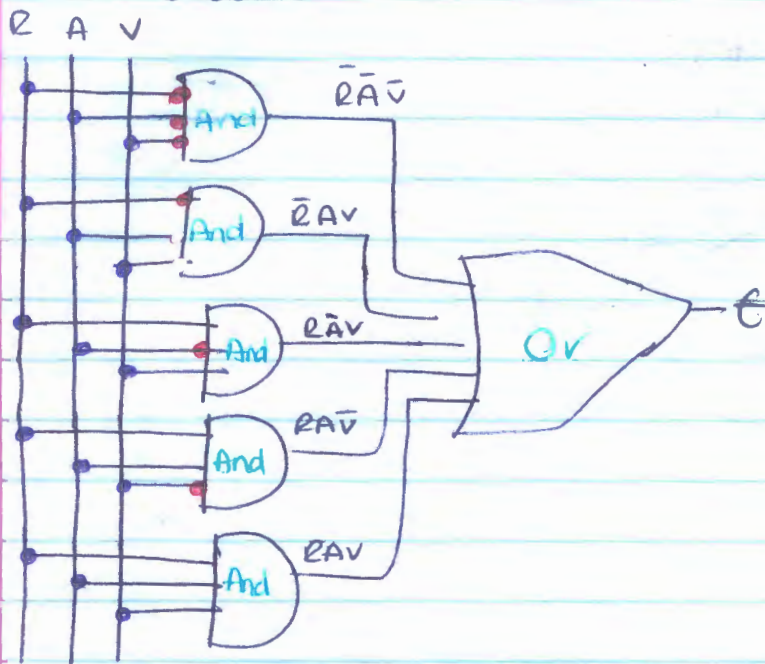
¿Cuántos circuitos integrados TTL de función fija se requieren para implementar este circuito?



9-Feb-10

$$F(E, A, V) = \bar{E}\bar{A}\bar{V} + \bar{E}AV + E\bar{A}\bar{V} + EAV + EAV$$

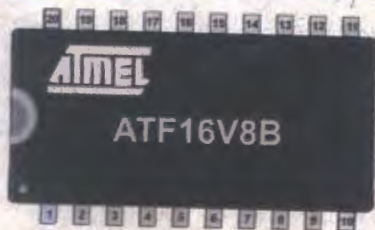
Circuito



Captura esquemática



Lattice Semiconductor



Atmel Corp.

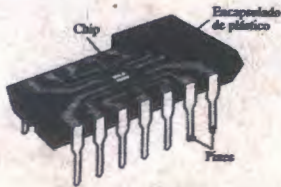
9- Feb-10

## Condiciones de Operación



		Min	Max	Uni
VCC	Voltaje de alimentación	4.75	5.25	vohs
TA	Temperatura en °C	0	75	°C
VIL	Input Low Voltage		0.8	Volts
VIH	Input High Voltage	2		Volts
IOH	High level output current		3.2	mA

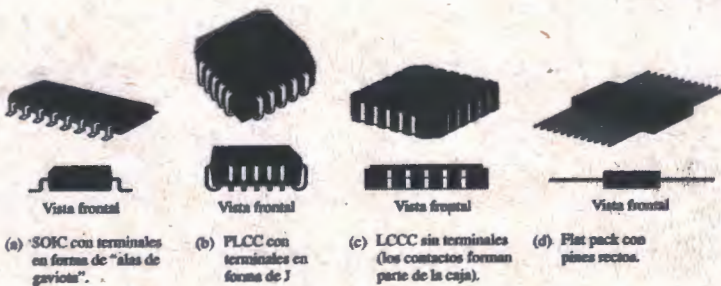
## Encapsulados de circuitos integrados digitales



(a) Encapsulado DIP



(b) Encapsulado SOIC



# Maxitermino

```
Modelo Maxi
Esto es una Salida estructura de datos
" entrada
x A, e PIN;
" Salida
ERROR PIN IS TYPE 'com';
TEST VECTORS
[0, 0, 0] -> .x;
[0, 0, 1] -> .x;
[0, 1, 0] -> .x;
[0, 1, 1] -> .x;
[1, 0, 0] -> .x;
[1, 0, 1] -> .x;
[1, 1, 0] -> .x;
[1, 1, 1] -> .x;
```

Diagrama de bloques

# Maxitermino

11-Feb-10

Termino suma (OR) que contiene todas las variables de la función ya sea en su forma normal (afirmada) o complementada (negada)

Detector de errores de un semáforo:



m	R	A	V	E
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Función negada como alternativa

m	R	A	V	E	$\bar{E}$
0	0	0	0	1	0
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	0

m	R	A	V	E	$\bar{E}$
0	0	0	0	1	0
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	0

Miniterminos de la función E negada

$$F\bar{E}(R,A,V) = \bar{R}\bar{A}V + \bar{R}A\bar{V} + R\bar{A}\bar{V}$$

$$F\bar{E}(R,A,V) = \bar{R}\bar{A}V + \bar{R}A\bar{V} + R\bar{A}\bar{V}$$

$$F\bar{E}(R,A,V) = \bar{R}\bar{A}V + \bar{R}A\bar{V} + R\bar{A}\bar{V}$$

11 Feb-10

### Miniterminos de la función negada

$$F_e(r, A, v) = \overline{R} \overline{A} \overline{v} + \overline{R} A \overline{v} + R \overline{A} \overline{v} \quad R A \overline{v} + 0$$

Aplicando el teorema de D'Morgan AND por NOR

$$F_e(r, A, v) = (R + A + \overline{v}) + (R + \overline{A} + v) + (\overline{R} + A + v)$$

Aplicando el teorema de D'Morgan NOR por AND

$$F_e(r, A, v) = (R + A + \overline{v})(R + \overline{A} + v)(\overline{R} + A + v)$$

### Maxiterminos

$$\overline{(R + A + \overline{v})} + \overline{(R + \overline{A} + v)} + \overline{(\overline{R} + A + v)}$$

m	R	A	v	E
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

$$F_e(r, A, v) = (R + A + \overline{v})(R + \overline{A} + v)(\overline{R} + A + v)$$

POS

### Maxiterminos

Termino suma OR que contiene todas las variables de la función ya sea en su forma normal (afirmada) o complementada (negada)

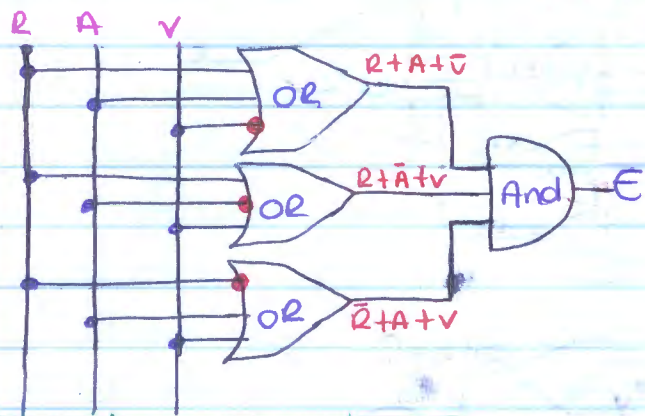
$$F_e(r, A, v) = (R + A + \overline{v})(R + \overline{A} + v)(\overline{R} + A + v)$$

11-Feb-10

Maxiterminos

Forma POS

Productos de Sumas  $F_{(R,A,V)} = (R+A+\bar{V})(R+\bar{A}+V)(\bar{R}+A+V)$



$$\bar{A} + B\bar{C}$$

$$\frac{\bar{A} + B\bar{C}}{A(\bar{B} + C)}$$

m	R	A	V	E
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

POS:  $F_{(R,A,V)} = (R+A+\bar{V})(R+\bar{A}+V)(\bar{R}+A+V)$

Forma

Canonica  $F_{(R,A,V)} = \prod M(1, 2, 4)$

$$\bar{A} + B\bar{C}$$

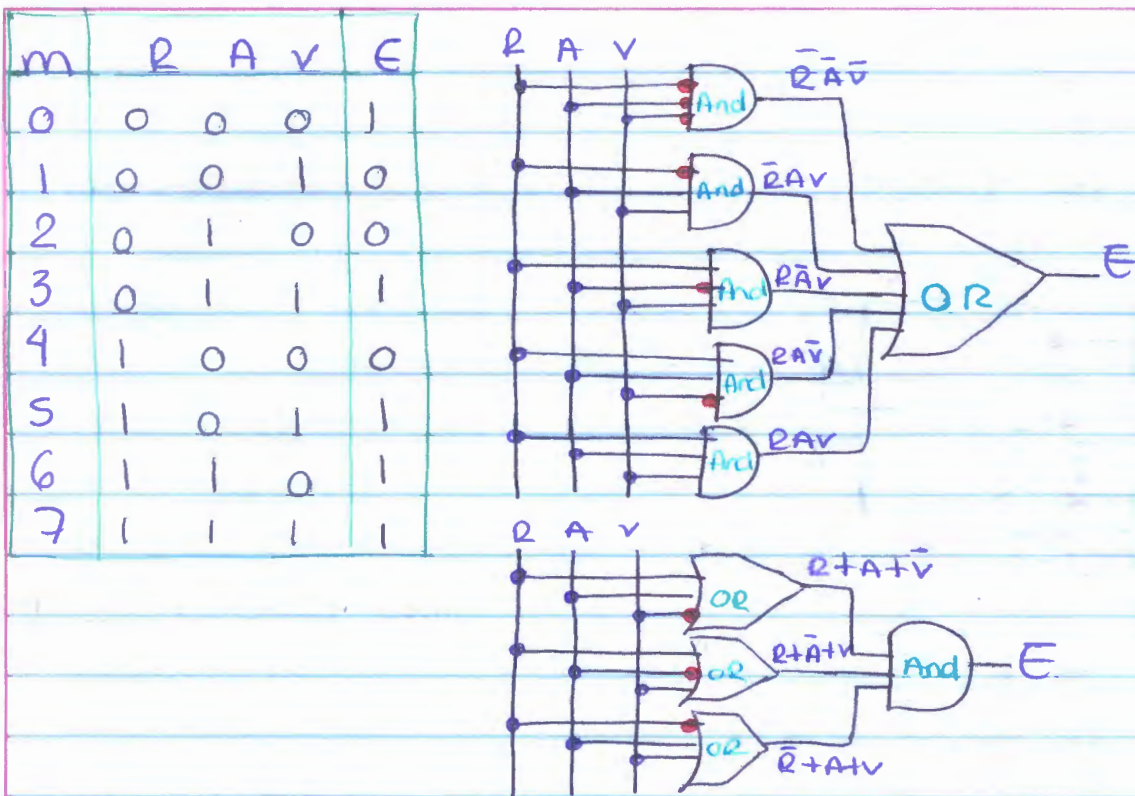
$$\frac{\bar{A} + B\bar{C}}{A(\bar{B} + C)}$$

$$A(\bar{B} + C)$$

$$\frac{\bar{A} + B\bar{C}}{A(\bar{B} + C)}$$



11 Feb-10



Maxiterminos  
Pos

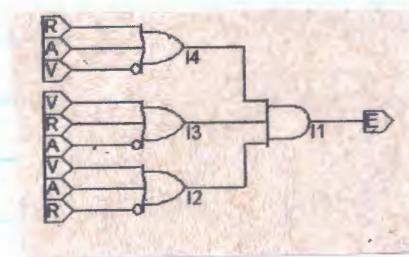
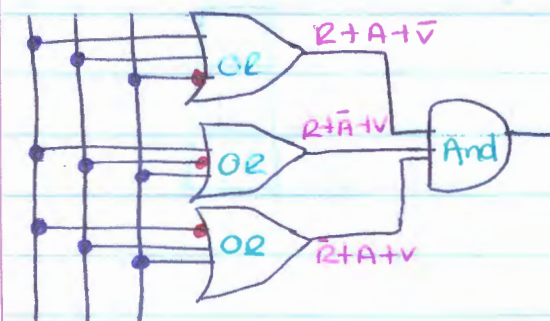
m	A	B	C	Maxitermino	Minitermino
0	0	0	0	$A+B+C$	$\bar{A}\bar{B}\bar{C}$
1	0	0	1	$A+B+\bar{C}$	$\bar{A}\bar{B}C$
2	0	1	0	$A+\bar{B}+C$	$\bar{A}B\bar{C}$
3	0	1	1	$A+\bar{B}+\bar{C}$	$\bar{A}BC$
4	1	0	0	$\bar{A}+B+C$	$A\bar{B}\bar{C}$
5	1	0	1	$\bar{A}+B+\bar{C}$	$ABC$
6	1	1	0	$\bar{A}+\bar{B}+C$	$AB\bar{C}$
7	1	1	1	$\bar{A}+\bar{B}+\bar{C}$	$ABC$

De que depende elegir Miniterminos a Maxiterminos?  
 Depende de que sean menos ceros o unos.

11-Feb-10

	A	B	C	D	F1	F2	F3	
0	0	0	0	0	1	1	1	Obtenga las formas SOP o POS y la más conveniente para cada función
1	0	0	0	1	1	1	0	
2	0	0	1	0	0	0	1	
3	0	0	1	1	0	1	0	
4	0	1	0	0	0	1	1	Miniterminos
5	0	1	0	1	0	0	0	F1 Forma SOP y Canónica Σ
6	0	1	1	0	1	1	1	
7	0	1	1	1	0	0	0	Maxiterminos
8	1	0	0	0	0	0	0	F2 Forma POS y Canónica Π
9	1	0	0	1	0	1	1	
10	1	0	1	0	0	1	0	F3 Formas POS y SOP
11	1	0	1	1	1	1	1	además de las canónicas ΣΠ
12	1	1	0	0	0	0	0	
13	1	1	0	1	0	0	1	
14	1	1	1	0	1	1	1	
15	1	1	1	1	0	1	0	

### Simulación



Captura esquemática

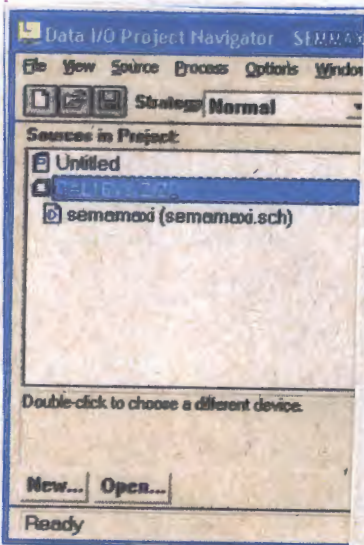
11-Feb-10

Archivo  
.abv  
ABEL VECTORS

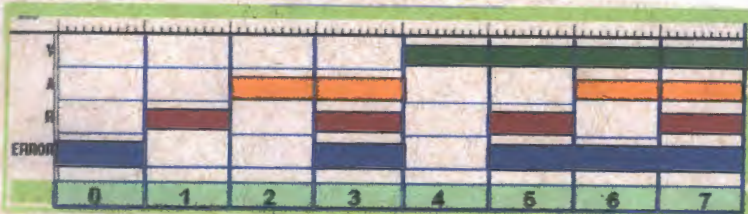
```

Text Editor (error.abv)
File Edit View Templates Tools Options Window Help
Module error
"Entradas
V,A,R pin;
"Salida
E pin istype 'com';
test_vectors
([V,A,R]->E);
[0,0,0]->.x.;
[0,0,1]->.x.;
[0,1,0]->.x.;
[0,1,1]->.x.;
[1,0,0]->.x.;
[1,0,1]->.x.;
[1,1,0]->.x.;
[1,1,1]->.x.;
end

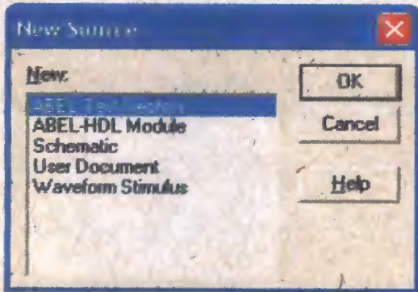
```



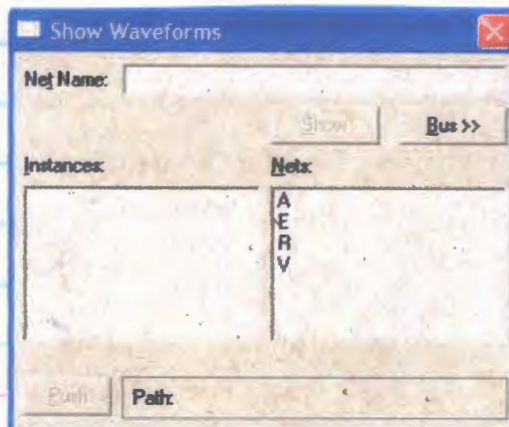
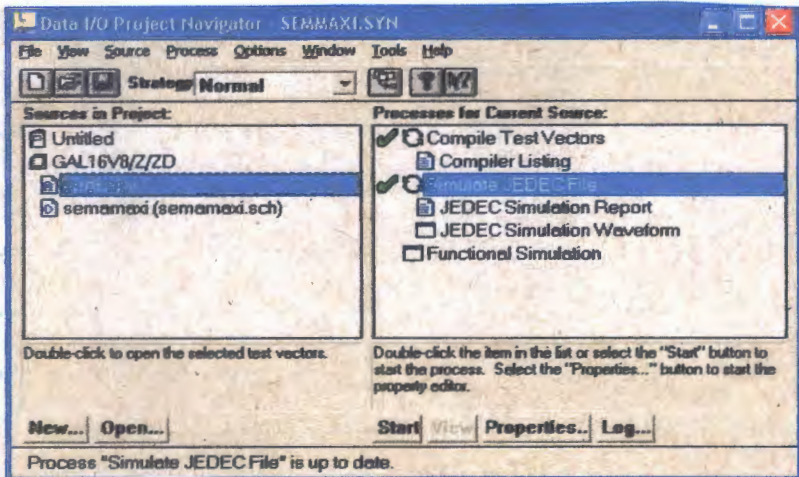
m	R	A	V	E
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1



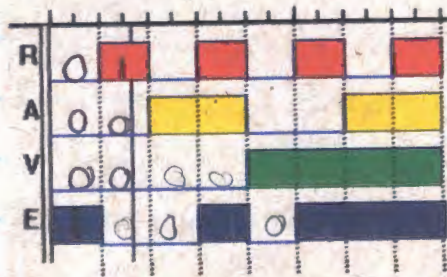
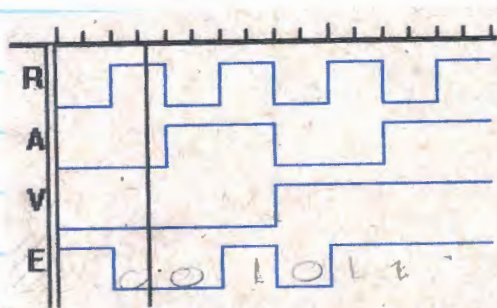
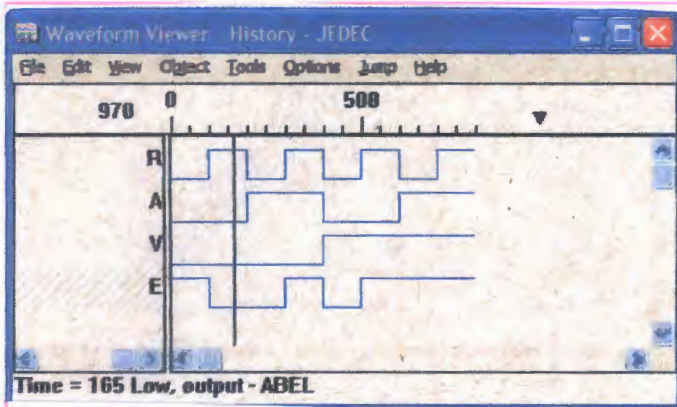
11-Feb-10



```
Text Editor [error.abv]
File Edit View Templates Tools Options Window Help
Module error
"Entradas
V,A,R pin;
"Salida
E pin istype 'com';
test_vectors
([V,A,R]->E);
[0,0,0]->.x.;
[0,0,1]->.x.;
[0,1,0]->.x.;
[0,1,1]->.x.;
[1,0,0]->.x.;
[1,0,1]->.x.;
[1,1,0]->.x.;
[1,1,1]->.x.;
end
```



11-Feb-10



```
MODULE SEM
V,A,R PIN;
ERROR PIN ISTYPE 'COM';
TEST_VECTORS
([V,A,R]->[ERROR]);
[0,0,0]->.X.;
[0,0,1]->.X.;
[0,1,0]->.X.;
[0,1,1]->.X.;
[1,0,0]->.X.;
[1,0,1]->.X.;
[1,1,0]->.X.;
[1,1,1]->.X.;
END
```

Teorema d' Morgan

$\bar{A} + B\bar{C}$

$(x' + y)z'$

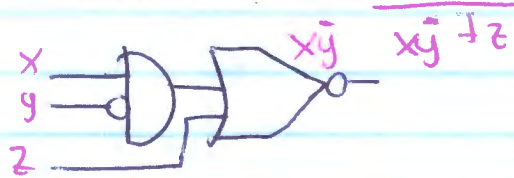
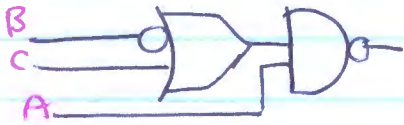
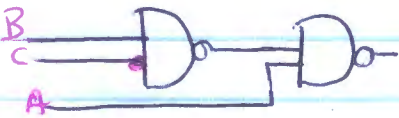
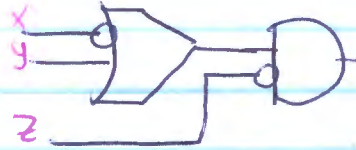
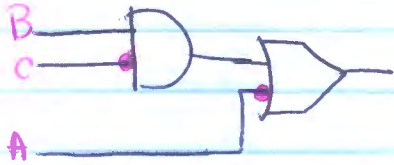


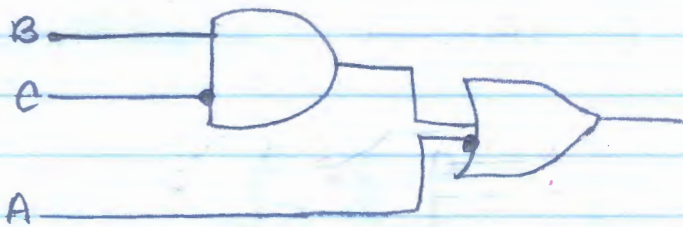
Diagrama de Bloques  
cuantas entradas y salidas

# Teorema de D' Morgan

12-Feb-10

## Actividad

✓  $F(A, B, C) = A' + BC'$



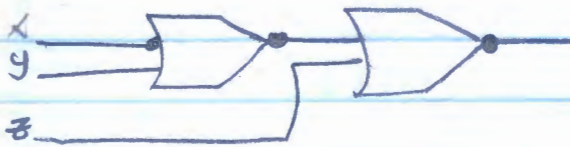
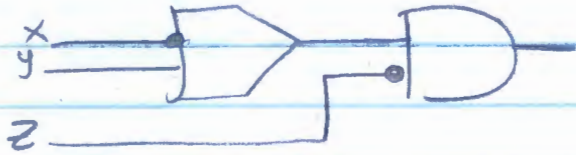
$BC' + A'$

12-Feb-10

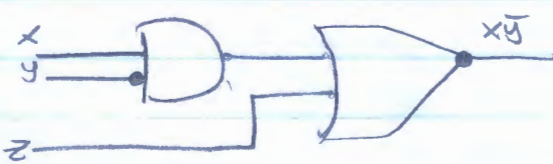
✓  $f(x,y,z) = (x' + y)z'$

$(\bar{x} + y) \bar{z}$

$\overline{x\bar{y} + z}$



AND → NAND  
OR → NAND



$x\bar{y}$

$(\bar{x} + y) \bar{z}$

$\overline{x\bar{y} + z}$

$\overline{(x\bar{y}) \bar{z}}$

$\overline{\overline{(x\bar{y} + z)'}}$

$(x\bar{y} + z)'$



12-Feb-10

$$F(A,B,C) = [(A+B')C]'$$

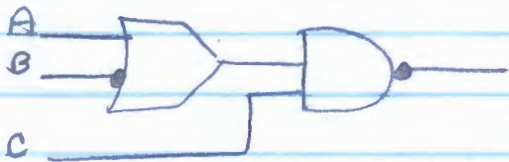
$$\overline{AB + c}$$

$$(\overline{A+B})C'$$

$$A\overline{B}C$$

$$[(A+B)']C'$$

$$\overline{A+B}C'$$



10



$$[(A+B)C]'$$

$$(\overline{A+B})\overline{C}$$

$$\overline{(A\overline{B})\overline{C}}$$



$$[\overline{(A\overline{B} + c)}]'$$

$$A\overline{B} + \overline{C}$$

$$[\overline{A\overline{B} + c}]'$$

$$[(A+B')C]'$$

$$[\overline{(A\overline{B})\overline{C}}]'$$

$$\overline{A\overline{B} + \overline{C}}$$

$$[\overline{A\overline{B}\overline{C}}]'$$

$$A\overline{B}$$

$$[(A+B)C]'$$

$$\overline{[(A\overline{B})\overline{C}]}$$

12-Feb-10

Obtenga el circuito equivalente y la ecuación aplicando el teorema de D' Morgan y compare el resultado usando la tabla de verdad

$$F_2(A,B,C) = [(A'+B+C)' (A+B+C)']'$$

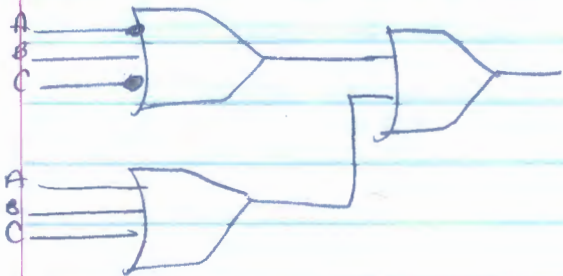


$$[(A'+B+C)' (A+B+C)']'$$

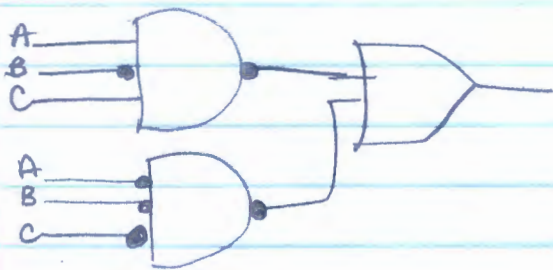


$$[(A\bar{B}\bar{C})' (\bar{A}\bar{B}\bar{C})']'$$

$$[(A\bar{B}\bar{C})'' + (\bar{A}\bar{B}\bar{C})'']'$$



$$A\bar{B}\bar{C}' + \bar{A}$$



$$A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C}$$

17-Feb-10

# Minimización de Funciones Booleanas

## Minimización de Funciones Booleanas

- Manipulación Algebraica
- Mapas de Karnaugh
- Software LogicAid

## Manipulación Algebraica

- Factorización
- Duplicando un término ya existente
- Teorema del consenso
- Propiedades distributiva
- Identidades
- Teorema de D' Morgan

## Mapas de Karnaugh

	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	1	1
10	0	0	0	1

Handwritten annotations: 'FG' in a circle next to the first column, 'ab' above the second column, and 'cd' to the left of the first row.

### INPUT TRUTH TABLE FORMAT

Number of Variables

Number of Functions

OK

Cancel

#### Variable and Function Names:

Use Default Names (Do Not Enter)

Enter Names

Use Current Names

#### Format for Input Combinations:

Straight Binary Order (Auto Entry Mode)

Decimal

Hexadecimal

Binary (0,1,-)

Octal

#### Format for Output Combinations:

Decimal

Hexadecimal

Binary (0,1,-)

Octal

#### Output Value for Remaining Rows:

0-Terms

1-Terms

X-Terms

18-Feb-10

## Manipulación Algebraica

- Factorización
- Duplicando un término ya existente
- Teorema del consenso
- Propiedad distributiva

## Factorización

Manipulación algebraica

m	A	B	F
0	0	0	0
1	0	1	1
2	1	0	0
3	1	1	1

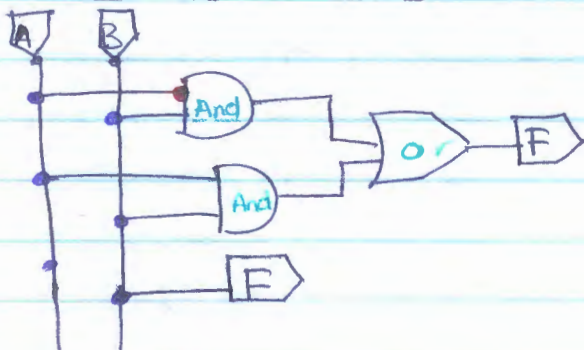
$$F = \bar{A}B + AB$$

$$F = B(\bar{A} + A) \quad \text{Factor común B}$$

$$F = B(\bar{A} + A)$$

$$F = B$$

$$F(A, B) = \bar{A}B + AB = B$$



La factorización se efectúa cuando solo cambia una variable entre dos términos y esta variable se elimina

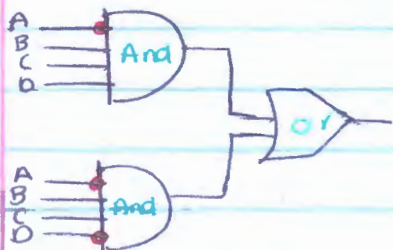
$$F = \bar{A}BC + ABC$$

$$F = BC$$

18-Feb-10

$$F = \bar{A}BCD + \bar{A}Bc\bar{D}$$

$$F = \bar{A}BC$$



$$F = \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}C\bar{D}$$

$$F = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C$$

$$F = \bar{A}\bar{B}$$

m	S	P	E	CS	CP	CE
0	0	0	0	0	0	0
1	0	0	1	0	0	1
2	0	1	0	0	1	0
3	0	1	1	0	1	0
4	1	0	0	1	0	0
5	1	0	1	1	0	0
6	1	1	0	1	0	0
7	1	1	1	1	0	0

$$C(S, P, E) = SP\bar{E}' + SP'E + SP\bar{E}' + SP'E$$

18-Feb-10

$$CS(s, p, e) = s'p'e' + s'p'e + spe' + spe$$

$$S(s, p, e) = s'p'(e' + e) + sp'(e' + e)$$

$$CS(s, p, e) = s'p' + sp'$$

$$CS(s, p, e) = s(p' + p)$$

$$\underline{CS(s, p, e) = S}$$

$$CS(s, p, e) = s'p'e' + s'p'e + spe' + spe = S$$

m	s	p	e	cs	cp	ce
0	0	0	0	0	0	0
1	0	0	1	0	0	1
2	0	1	0	0	1	0
3	0	1	1	0	1	0
4	1	0	0	1	0	0
5	1	0	1	1	0	0
6	1	1	0	1	0	0
7	1	1	1	1	0	0

$$CP(s, p, e) = s'p'e' + s'p'e$$

$$CP(s, p, e) = s'p'(e' + e)$$

$$CP(s, p, e) = s'p$$

$$\underline{CP(s, p) = s'p}$$

Duplicando un termino ya existente

$$F = \bar{A}\bar{B} + \bar{A}B + A\bar{B}$$

$$F = \bar{A}\bar{B} + \bar{A}B + \bar{A}\bar{B} + A\bar{B}$$

$$F = \bar{A} + \bar{B}$$

$$A + A(\bar{A} + \bar{B})$$

18-Feb-10

m	A	B	C	X
0	0	0	0	1
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	0
7	1	1	1	0

$$F_X(A,B,C) = A'B'C' + A'BC + A'BC'$$

$$F_X(A,B,C) = A'B'C' + A'BC + A'BC' + A'B'C'$$

$$F_X(A,B,C) = A'B' + A'C'$$

$$F_X(A,B,C) = A'(B' + C')$$

m	A	B	C	D	S
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

$$S(A,B,C,D) = A'B'C'D' + A'B'CD + A'BC'D' + A'BCD + A'BC'D + AB'CD + AB'C'D + ABC'D'$$

$$S(A,B,C,D) = A'BC' + A'B'C + AB'CI + 'BC'D'$$

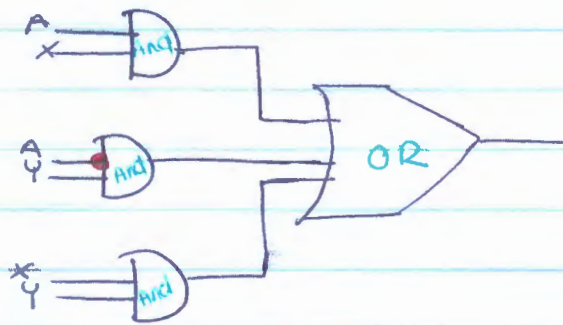
$$S(A,B,C,D) = B'C' + A'B' + 'BC'D'$$

$$S(A,B,C,D) = A'B' + B'C' + C'D'$$

18 - Feb - 10

### Teorema del Consenso

$$F = AX + \bar{A}Y + XY$$



Consenso m. Acuerdo

Producido por consentimiento entre todos los miembros de un grupo o entre varios grupos.

### Teorema del consenso

$$F = AX + \bar{A}Y + XY$$

$$F = AX + \bar{A}Y + XY (A + \bar{A})$$

$$F = AX(1 + Y) + \bar{A}Y(1 + X)$$

$$F = AX + \bar{A}Y + AXY + \bar{A}XY$$

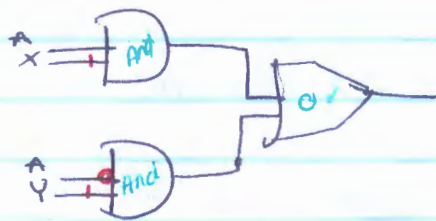
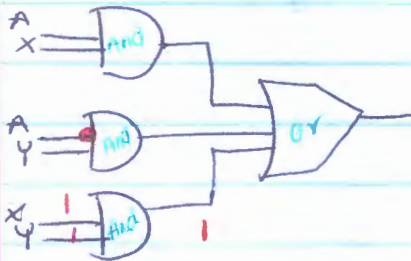
$$F = AX + AXY + \bar{A}Y + \bar{A}XY$$

$$\underline{F = AX + \bar{A}Y} \quad \times$$

### Teorema del consenso

$$F = AX + \bar{A}Y + \boxed{XY}$$

$$F = AX + \bar{A}Y$$

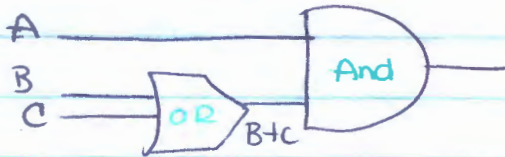




18-Feb-10

## Propiedad Distributiva

- $F = (A+B)(A+C) = A+BC$



$$F = (A+B)(A+C) = A+BC$$

$$F = (A+B)(A+\bar{C})(A+D)(A+E)$$

$$F = (A+B)(A+\bar{C})(A+D)(A+E) = A+B\bar{C}DE$$

$$A+BC = (A+B)(A+C)$$

$$F = A + \bar{A}C = (A + \bar{A})(A + C)$$

$$F = A + C$$

- $F = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$

$$F = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$$

$$F = A'B + AB' + AB + A'C'$$

$$F = B + A + A'C'$$

$$F = B + (A+A')(A+C')$$

$$F = B + A + C'$$

$$F = A + BC'$$

19-Feb-10

## Identidades

AND

OR

$$A+A=A$$

$$A+A=A$$

$$A0=0$$

$$A+0=A$$

$$A1=A$$

$$A+1=1$$

$$AA'=0$$

$$A+A'=1$$

## 2. Factorización

$$AB' + AB = A(B' + B) = A$$

## 3. Propiedad Distributiva

$$x+yz = (x+y)(x+z)$$

$$x(y+z) = xy + xz$$

## 4. Teorema del consenso

$$AB + A'C + BC = AB + A'C$$

## 5. Teorema de DeMorgan

$$(AB)' = A' + B' \quad (A+B)' = A'B'$$

$$A+B = (A'B')' \quad AB = (A'B')'$$

$$F_0(A, B, C) = A'BC + A'BC + ABC$$

$$= A'BC + ABC$$

$$= BC(\bar{A} + A)$$

$$= BC$$

$$F_0(A, B) = A(A+B)$$

$$= AA + AB$$

$$= A(1)$$

$$= A + AB$$

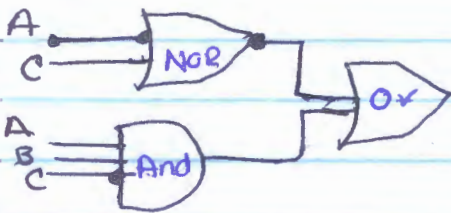
$$= A$$

$$= A(1+B) \rightarrow \text{Identidad}$$

19-Feb-10

$$F_7(A, B, C) = \frac{A + A'BC}{x \quad y \quad z} = (x+y)(x+z)$$
$$= (A + A')(A + BC)$$
$$= \boxed{A + BC}$$

$$F_8(A, B, C) = \overline{(A + \bar{B} + C + 1)}$$
$$= \bar{1}$$
$$= \boxed{0}$$



$$\bar{A}B\bar{C} + \overline{(A\bar{C})}$$
$$A\bar{B}\bar{C} + A\bar{C}$$
$$A\bar{C}(B + 1)$$
$$\boxed{A\bar{C}}$$

$$F_{10}(C, B, A) = C'BA + C'B'A + C'BA'$$
$$= \boxed{ABC'} + \boxed{AB'C'} + \boxed{A'BC'}$$
$$= \cancel{BC'} + AC'$$
$$= \boxed{C'(BA)}$$

19-Feb-10

$$F_{11}(A, B) = A(A' + AB)$$

$$= AA' + AAB$$

$$= \boxed{AB}$$

$$F_{12}(A, B) = A(A + A'B)$$

$$= AA + AA'B$$

$$= A + 0 = \boxed{A}$$

$$F_{13}(A, B, C) = \frac{(A+B')}{x+y} \frac{(A+C)}{x+z}$$

$$= (x+y)(x+z)$$

$$= \boxed{A + B'C}$$

$$F_{14}(A, B, C, D, E) = A'B + A'BC' + ABCD + A'BC'D'E$$

$$= A'B(1 + C' + CD + C'D'E)$$

$$= \boxed{A'B}$$

$$F_{15}(A, B, C) = AB + (AB)'C + A$$

$$= \underbrace{(X + X')C} + A$$

$$= (x + \bar{x})(x + c) + A$$

$$= 1(x + c) + A$$

$$= AB + C + A$$

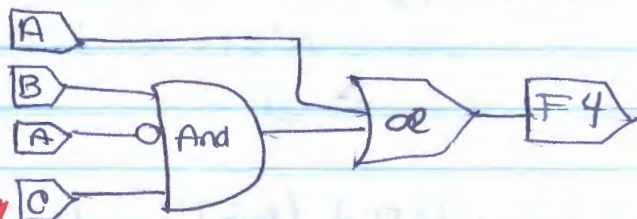
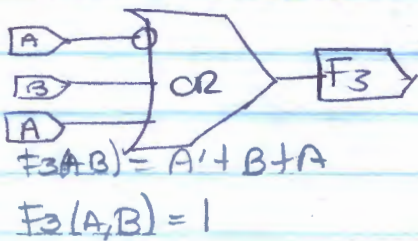
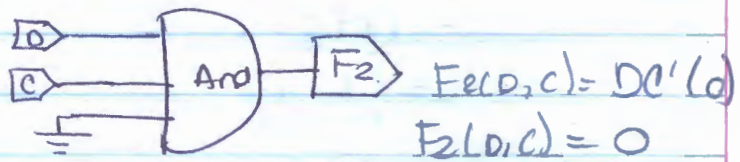
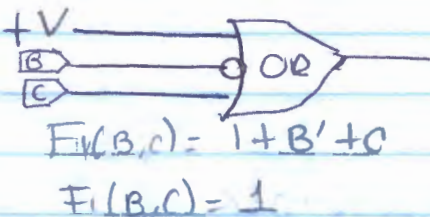
$$= A(B + 1) + C$$

$$= \boxed{A + C}$$

19-Feb-10

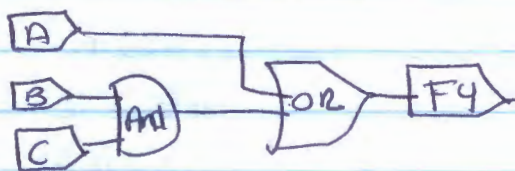
Resuelva las siguientes funciones:

1.  $1 + B' + C$
2.  $DC' + 0$
3.  $A' + B + A$
4.  $A + A'BC$
5.  $A'BC + A'BC'$



$F_4(A,B,C) = A + A'BC$   
 $F_4(A,B,C) = (A + A')(A + BC)$   
 $F_4(A,B,C) = A + BC$

Reducción



22-Feb-10

$$F(x, y, p) = (x+p)(p'+y)(x+y)$$

$$(x+p)(x+y)(p'+y)$$

$$(x+py)(p'+y)(p'+p)$$

$$(x+py)(p'+py)$$

$$\boxed{py + xp'}$$

Nota A

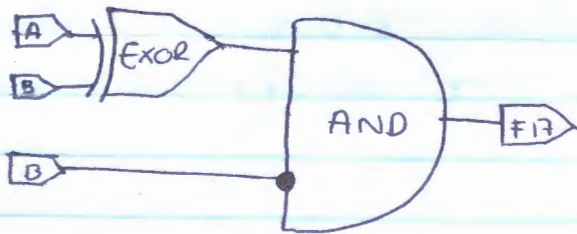
lo paso p' no afecta p' es como 1 y lo paso para eliminar el p y p'

$$A \oplus B = \bar{A}B + A\bar{B}$$

Teorema del Consenso

$$ax + \bar{a}y + xy = ax + \bar{a}y$$

$$(p+x)(p'+y)(x+y) = (p+x)(p'+y)$$



$$(A \oplus B) \bar{B}$$

$$\bar{B} (\bar{A}B + A\bar{B})$$

$$\bar{A}\bar{B}B + A\bar{B}\bar{B}$$

$$\bar{A}0 + A\bar{B}$$

$$0 + A\bar{B} = \boxed{A\bar{B}}$$

$$F1(A, B, C) = A'B'C' + A'BC + ABC' + (A'B'C)'$$

$$\bar{A}\bar{B}\bar{C} + \overline{A\bar{B}\bar{C}}$$

$$= \bar{A}\bar{B}\bar{C} + \overline{A\bar{B}\bar{C}}$$

$$x + \bar{x} = 1$$

$$= x + \bar{x} = 1$$

$$A'BC + ABC' + 1$$

Un uno es una Or con =  $\boxed{1}$

cualquier 1 es = 1

22-Feb-10

$$F_{19}(x, y, z, w) = (x + y + z' + w)(y' + z)(x'y'zw')$$

$$A \quad (y' + z) \quad \bar{A}$$

$$A\bar{A} = 0$$

$$(y' + z) 0$$

$$= \boxed{0}$$

$$*F_{20}(x, y, z) = (x + yz)(x + y') + (x + y)'$$

$$= (x + yz + y') + (x + y)'$$

$$= (x + 0) + (x + y)'$$

$$= x + x'y'$$

$$= (x + x')(x + y')$$

$$= 1(x + y')$$

$$= \boxed{x + y'}$$

P. Dist.

$$x + yz = (x + y)(x + z)$$

$$F_{21}(A, B, C) = AB + (A' + B')C + AB = 1$$

$$= AB + (A' + B')C$$

$$= \overline{ABC}$$

$$= AB + \overline{ABC}$$

$$= x + \bar{x}C$$

$$= (x + \bar{x})(x + C)$$

$$= 1(x + C)$$

$$= x + C$$

$$= \boxed{AB + C}$$

22-Feb-10

$$F_{22}(x, y, z) = (x \oplus y) + xy'z + x'$$

$$= \bar{x}y + \underline{x\bar{y}} + \underline{xy'z} + \bar{x}$$

$$= \bar{x} + \bar{x}y + x\bar{y} + x\bar{y}z$$

$$= \bar{x}(1+y) + x\bar{y}(1+z)$$

$$= \bar{x} + x\bar{y}$$

$$= (\bar{x}+x)(\bar{x}+\bar{y})$$

$$= 1(\bar{x}+\bar{y}) = \boxed{\bar{x}+\bar{y}}$$

$a+bc = (a+b)(a+c)$   
 $\bar{x}+x\bar{y} = (\bar{x}+x)(\bar{x}+\bar{y})$

$$F_{23}(x, y, z) = \bar{x}yz + x\bar{y}\bar{z} + xyz + x\bar{y}z$$

$$= \bar{x}yz + xyz + x\bar{y}\bar{z} + x\bar{y}z$$

$$= \boxed{yz + x\bar{y}}$$



$$F_{32}(x, y, z) = \bar{x}y + z\bar{y} + x$$

$$= x + \bar{x}y + z\bar{y}$$

$$= (x + \bar{x})(x + y) + z\bar{y}$$

$$= x + y + z\bar{y}$$

$$= x + (y+z)(y+\bar{y})$$

$$= x + y + z + 1 = \boxed{x + y + z}$$

$$F_{33}(x, y, z) = (x + \bar{y} + xy)(xy + x'z + yz)$$

$$= (x + y' + xy')(xy + x'z)$$

$$= x(1 + y') + y'$$

$$= (x + y')(xy + x'z)$$

$$= (xy + 0 + 0 + x'y'z)$$

$$= \boxed{xy + x'y'z}$$



Tarea

24-Feb-10

$$F_3(A, B, C) = A + A'BC + A'A'BC$$

$$F_9 = (AB'1)'$$

$$\begin{aligned} F_{24}(A, B, C, D, E) &= B'C'DE + A(AB'+E)' + C'(AB+E) \\ &= B'C'DE + AAB'E + C'ABE \\ &= B'C'DE + AAB'E + ABC'E \\ &= BC'DE + AB'E + ABC'E \end{aligned}$$

$$F_{25}(x, y, z) = (x + y' + xy')(xy + x'z + yz)$$

$$\begin{aligned} F_{26}(A, B, C) &= A'BC' + B'(AC') \\ &= A'BC' + AB'C + B'C' \\ &= \dots \\ &= C'(1 + A) = C' \end{aligned}$$

$$\begin{aligned} &\overline{A}B\overline{C} + \overline{B}C + A\overline{B}C \\ &\overline{C}(\overline{A}B + \overline{B}) + A\overline{B}C \\ &\overline{C}(\overline{A} + \overline{B}) + A\overline{B}C \\ &\overline{A}C + \overline{B}C + A\overline{B}C \end{aligned}$$

26-Feb-10

$$F_{27}(B, C, D) = BC' + B'C'D + BCD'$$

$$\begin{aligned} F_{28}(A, B, C) &= (A+A')(AB+ABC') \\ &= 1(AB+ABC') \\ &= AB+ABC' \end{aligned}$$



$$F_{29}(A, B, C, D) = (A+B+C'+D')(A+B+C)(A+C+D')(C'D)'$$

(C+D)

$$F_{30}(X, Y, Z, W) = (X+Y+Z'+W)(Y'+Z)(X'Y'ZW')$$

$$F_{31}(A, B, C, D) = AB+B'C + AC + ACD$$

20-Feb-10

$$F_{34}(x, y) = xy(x+y)$$

$$F_{35}(B, C, D) = BC' + B'C'D$$

$$F_{36}(A, C) = \begin{matrix} A' & + & AC \\ x & y & z \end{matrix}$$
$$= (\cancel{A'+A'}) (A'+C) = \boxed{A'+C}$$

$$F_{37}(A, B, C) = A'BC' + B'(A+C')$$
$$= A'BC' + AB'C'$$
$$= \boxed{C'}$$

$$F_{38}(A, B, C, D) = (A'+B+C'+D')(A+B+C)(A+C+D')(C'D)'$$

$$F_{39}(B, C, D, E) = BD + B(D + E) + D'(D + E)$$

$$F_{40}(A, B, C) = AB'C + A'BC + A'B'C$$

$$F_{41}(A, B, C) = (A + B')(A + C)$$

$$F_{42}(A, B, C, D, E) = A'B + A'BC' + A'BCD + A'BC'D'E$$

$$F_{43}(A, B, C) = (A + A')(AB + ABC')$$

26 Feb -10

$$F_{44}(A, B, C) = AB(A' + B')C + AB$$

$$F_{46}(A, B, C) = A'B'C + (A+B+C) + A'B'C'D$$

$$F_{47}(B, C, D) = (B+BC)(B+B'C)(B+D)$$

$$F_{48}(A, B, C, D) = ABCD + AB(CD)' + (AB)'CD$$

$$F_{49}(A, B, C) = ABC[AB + C'(B'C + AC)]$$

01-034

26 Feb-10

$$F_{50} \quad z = xw + (xy + y')x'$$

$$51. \quad z = w + xy + xw + w' + x$$

$$52. \quad z = xw(xy + x'y' + x'y + xy')$$

$$53. \quad z = (xw')(wx')$$

$$54. \quad z = (x' + w)(w' + x)$$

20-Feb-10

$$55. z = (w+y)(w+Q) + w'$$

$$56. z = xy' + x'y'Q$$

$$57. z = x'(w+x)'w' + x$$

$$58. z = y + wx + wy + x'$$

$$59. z = x'(y+w')[(w'+x)'+x]$$

26-Feb-10

$$F_{60}(A, B, C) = ACD + BC + AC + AB'B$$

$$F_{61}(w, x, y, z) = (w+y+x)(x+z)(w+x)(wtw')$$

$$F_{62}(A, B, C, D) = (A'+D')(B'+C'+D)(A+D')$$

$$F_{63}(a, b, c, d) = c(a \oplus b) + a(b' + c'd)$$



26-Feb-10

$$\begin{aligned}F_{64} F(x, y, z) &= x'yz + xy z' + xyz \\ &= y + xyz\end{aligned}$$

$$\begin{aligned}F_{65} (x, y, z) &= xy(x+y) + zz' \\ &= AA + zz' \\ &= A + zz' \\ &= xy + 0 \\ &= \boxed{xy}\end{aligned}$$

$$F_{66} (x, d, q, z) = x'd + x'z + zd' + x'dq'$$

$$F_{67} (a, b, d, f) = (a+fd)(a+fd')(a+f'd)(a+b')$$

# Minimizaci3n de Funciones Booleanas

26-Feb-10

## Mapas de Karnaugh

### Maurice Karnaugh

Nacio el 4 de Octubre de 1924 en New York. Fue un fisico americano que tuvo como principal aporte. En 1953 la creaci3n de los Mapas de Karnaugh o Diagrama de Veitch, cuya funci3n es la de minimizar funciones algebraicas booleanas.

Graduado de Ingeniero en Telecomunicaciones en la universidad de Yale en 1952. Es actualmente gobernador emérito del ICC (International Council for Computer Communication).

Trabaja como investigador en los Laboratorios Bell desde 1952 a 1966 y en el centro de investigaci3n de IBM de 1966 a 1993. Impartió classer de informatica en el Politecnico de Nueva York de 1980 a 1999, y desde 1975 es miembro de IEEE (Institute of Electrical and Electronics Engineers) por sus aportaciones sobre la utilizaci3n de metodos numericos en las telecomunicaciones

### Tabla o mapa de Karnaugh, Kmap

Procedimiento grafico para la simplifiaci3n de funciones algebraicas de  $n$  numero de variables relativamente pequeño (en la practica se puede utilizar para funciones de hasta seis variables)

26-Feb-10

## Tabla o mapa de Karnaugh

Un diagrama o mapa de Karnaugh es una tabla de verdad dispuesta de manera adecuada para determinar por inspección la expresión mínima de suma de productos de una función lógica.

	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

Con 2 variables A y B se pueden tener 4 términos

$$\bar{A}\bar{B}$$

$$\bar{A}B$$

$$A\bar{B}$$

$$AB$$

La factorización se efectúa cuando solo cambia una variable entre dos términos y esta variable se elimina. Cada término de dos variables tiene dos posibilidades de factorización.

## Kmap para 2 variables

$$\bar{A}\bar{B}$$

$$\bar{A}B$$

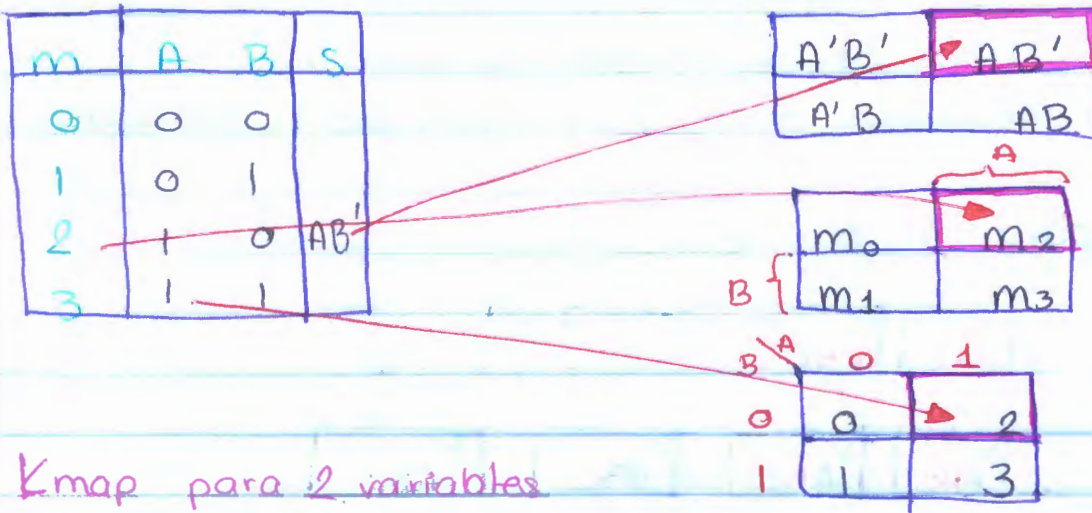
$$A\bar{B}$$

$$AB$$

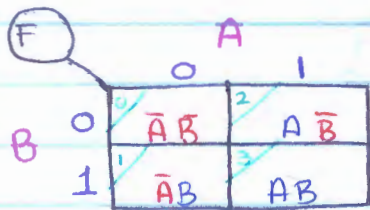
	A=0	A=1
B=0	$\bar{A}\bar{B}$	$A\bar{B}$
B=1	$\bar{A}B$	$AB$

26 Feb - 10

### Mapa de Karnaugh para dos variables



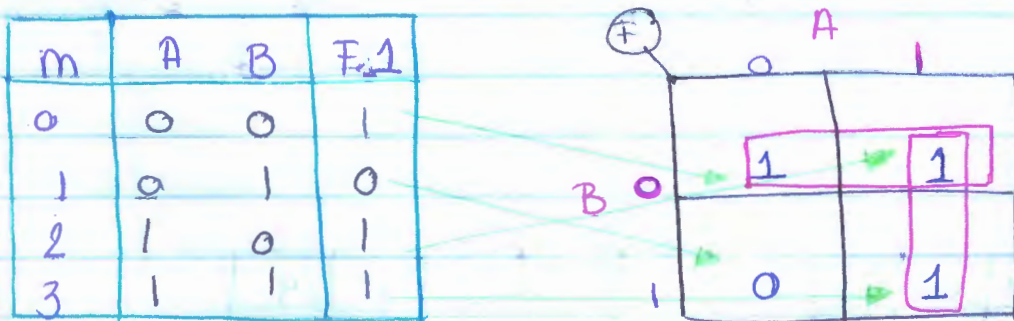
### Kmap para 2 variables



### Kmap para 2 variables



### Como llenar el Kmap para 2 variables



$$F(A,B) = A'B' + AB' + AB$$

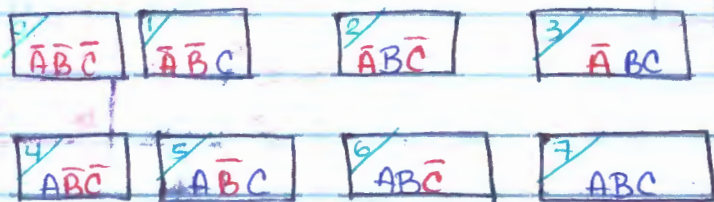
$$F(A,B) = A + B'$$

26-Feb-10

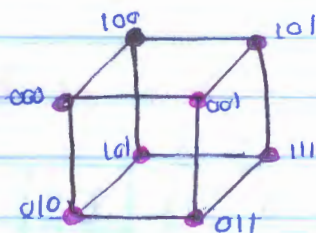
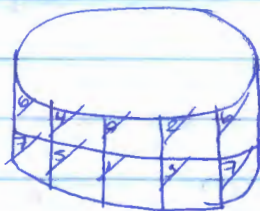
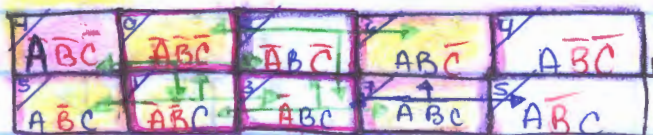
### Kmap para 3 variables

Con 3 variables se tienen 8 terminos,  
 Y cada termino tiene 3 posibles de factorización.

### Kmap para 3 variables



Cada termino tiene 3 posibilidades de factorización:



### Mapa de Karnaugh para 3 variables

	AB	00	01	11	10
C	0	A'B'C'	A'BC'	ABC'	AB'C'
	1	A'BC	A'BC	ABC	AB'C

	AB	00	01	11	10
a	0	0	2	6	4
	1	1	3	7	5

La idea en la codificación es poder usar el Pq.  $ab + ab' = a$

26-Feb-10

Kmap para 3 variables

4 A $\bar{B}\bar{C}$	0 $\bar{A}\bar{B}\bar{C}$	2 $\bar{A}B\bar{C}$	6 A $B\bar{C}$	4 A $\bar{B}\bar{C}$
5 A $\bar{B}C$	1 $\bar{A}\bar{B}C$	3 $\bar{A}BC$	7 A $BC$	5 A $\bar{B}C$

AB

	00	01	11	10
0	$\bar{A}\bar{B}\bar{C}$	A $\bar{B}\bar{C}$	A $B\bar{C}$	A $\bar{B}\bar{C}$
1	$\bar{A}\bar{B}C$	$\bar{A}BC$	A $BC$	A $\bar{B}C$

AB

	00	01	11	10
0				
1				

→ Kmap para 3 variables

$$F(A,B,C) = A'BC' + AB'C' + ABC' + ABC$$

	00	01	11	10
0		1	1	1
1			1	

- A'BC'
- AB'C'
- ABC'
- ABC

$$F(A,B,C) = BC' + AC' + AB$$

26-Feb-10

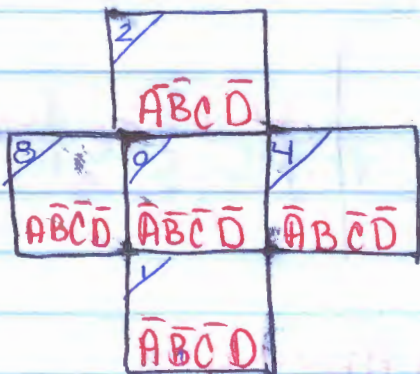
## Kmap para 4 variables

Con 4 variables se tienen 16 terminos

Y cada termino tiene 4 posibilidades de factorización

Cada termino tiene 4 posibilidades de factorización

0 $\bar{A}\bar{B}\bar{C}\bar{D}$	1 $\bar{A}\bar{B}\bar{C}D$	2 $\bar{A}\bar{B}C\bar{D}$	3 $\bar{A}\bar{B}CD$	4 $\bar{A}B\bar{C}\bar{D}$	5 $\bar{A}B\bar{C}D$	6 $\bar{A}BC\bar{D}$	7 $\bar{A}BCD$
8 $A\bar{B}\bar{C}\bar{D}$	9 $A\bar{B}\bar{C}D$	10 $A\bar{B}C\bar{D}$	11 $A\bar{B}CD$	12 $AB\bar{C}\bar{D}$	13 $AB\bar{C}D$	14 $ABC\bar{D}$	15 $ABCD$



26-Feb-10

Kmap para 4 variables

AB →

	10	00	01	11	10
10	10 $\bar{A}\bar{B}\bar{C}\bar{D}$	2 $\bar{A}\bar{B}C\bar{D}$	6 $\bar{A}BC\bar{D}$	14 $ABC\bar{D}$	10 $A\bar{B}C\bar{D}$
00	8 $\bar{A}\bar{B}C\bar{D}$	0 $\bar{A}\bar{B}C\bar{D}$	4 $\bar{A}\bar{B}C\bar{D}$	12 $ABC\bar{D}$	8 $A\bar{B}C\bar{D}$
01	9 $\bar{A}\bar{B}C\bar{D}$	1 $\bar{A}BC\bar{D}$	5 $\bar{A}\bar{B}C\bar{D}$	13 $A\bar{B}C\bar{D}$	9 $A\bar{B}C\bar{D}$
11	11 $\bar{A}BC\bar{D}$	3 $\bar{A}\bar{B}C\bar{D}$	7 $\bar{A}BC\bar{D}$	15 $ABC\bar{D}$	11 $A\bar{B}C\bar{D}$
10	10 $\bar{A}\bar{B}C\bar{D}$	2 $\bar{A}\bar{B}C\bar{D}$	6 $\bar{A}BC\bar{D}$	4 $ABC\bar{D}$	10 $A\bar{B}C\bar{D}$

Kmap para 4 variables

	00	01	11	10
00	0 $\bar{A}\bar{B}C\bar{D}$	4 $ABC\bar{D}$	12 $ABC\bar{D}$	8 $A\bar{B}C\bar{D}$
01	1 $\bar{A}\bar{B}C\bar{D}$	5 $ABC\bar{D}$	13 $ABC\bar{D}$	9 $A\bar{B}C\bar{D}$
11	3 $\bar{A}\bar{B}C\bar{D}$	7 $ABC\bar{D}$	15 $ABC\bar{D}$	11 $A\bar{B}C\bar{D}$
10	2 $\bar{A}\bar{B}C\bar{D}$	6 $\bar{A}BC\bar{D}$	14 $ABC\bar{D}$	10 $A\bar{B}C\bar{D}$

10	2 $\bar{A}\bar{B}C\bar{D}$	6 $\bar{A}BC\bar{D}$
00	8 $A\bar{B}C\bar{D}$	4 $ABC\bar{D}$



20-Feb-10

Mapa de Karnaugh para 4 variables

	AB	00	01	11	10
CD	00	$\bar{A}\bar{B}\bar{C}\bar{D}$	$A'B\bar{C}'D'$	$A\bar{B}C'D'$	$AB'C'D'$
	01	$A'B\bar{C}'D$	$A'BC'D$	$ABC'D$	$AB'C'D$
	11	$A'B\bar{C}D$	$A'BCD$	$ABC\bar{D}$	$AB'CD$
	10	$A'\bar{B}C'D$	$A'BCD'$	$AB\bar{C}D'$	$AB'CD'$

	AB	00	01	11	10
CD	00	0	4	12	8
	01	1	5	13	9
	11	3	7	15	11
	10	2	6	14	10

Kmap para 4 variables

		00	01	11	10
	00	0	4	12	8
	01	1	5	13	9
CD	11	3	7	15	11
	10	2	6	14	10

26-Feb-10

## Kmap para 5 variables

Con 5 variables se tienen 32 terminos  
y cada termino tiene 5 posibilidades de factorización

## Kmap para 5 variables

0	4	12	8
1	5	13	9
3	7	15	11
2	6	14	10

16	20	28	24
17	21	29	25
19	23	31	27
18	22	30	26

0	4	12	8
1	5	13	9
3	7	15	11
2	6	14	10

26-Feb-10

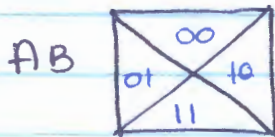


BC

DE

	00	01	11	10
00	0 16	4 20	12 18	8 24
01	3 17	5 21	13 29	9 25
11	2 19	6 23	14 31	10 27
10	7 18	15 22	11 30	14 26

Kmap para 6 variables.



CD

EF

	00	01	11	10
00	0 16 32 48	4 20 36 52	12 28 44 60	8 24 40 56
01	1 17 33 49	5 21 37 53	13 29 45 61	9 25 41 57
11	2 18 34 50	6 22 38 54	14 30 46 62	10 26 42 58
10	3 19 35 51	7 23 39 55	15 31 47 63	11 27 43 59

26-Feb-10

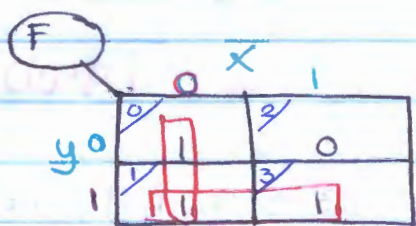
## Reglas para el uso de Kmap

1. Formar el menor número de grupos
2. Cada grupo lo más grande posible
3. Todas las unos deberán de ser agrupados
4. Un solo uno puede formar un grupo.
5. Casillas de un grupo pueden formar parte de otro de otro grupo

**Grupo:** Unas adyacentes enlazadas (paralelogramos) en una cantidad a una potencia entera de dos, e.g. (1, 2, 4, 8, ...).

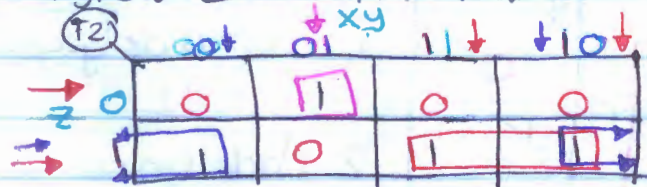
## Ejemplos de Kmap.

m	x	y	F
0	0	0	1
1	0	1	1
2	1	0	0
3	1	1	1



$$F(x,y) = x' + y$$

$$F_2(x,y,z) = \sum m(1, 2, 5, 7)$$

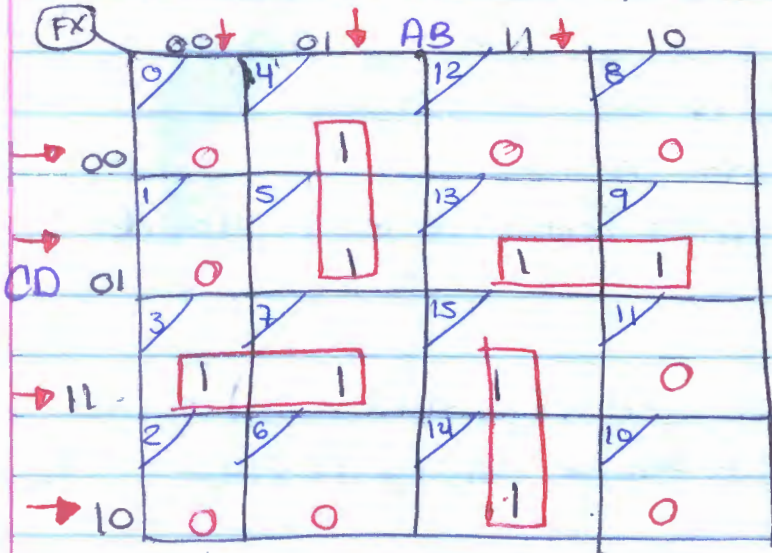


ejemplos de Kmap

$$F_2(x,y,z) = xz + y'z + x'y z'$$

26-Feb-10

$$F_x(A, B, C, D) = \sum m(3, 4, 5, 7, 9, 13, 14, 15)$$

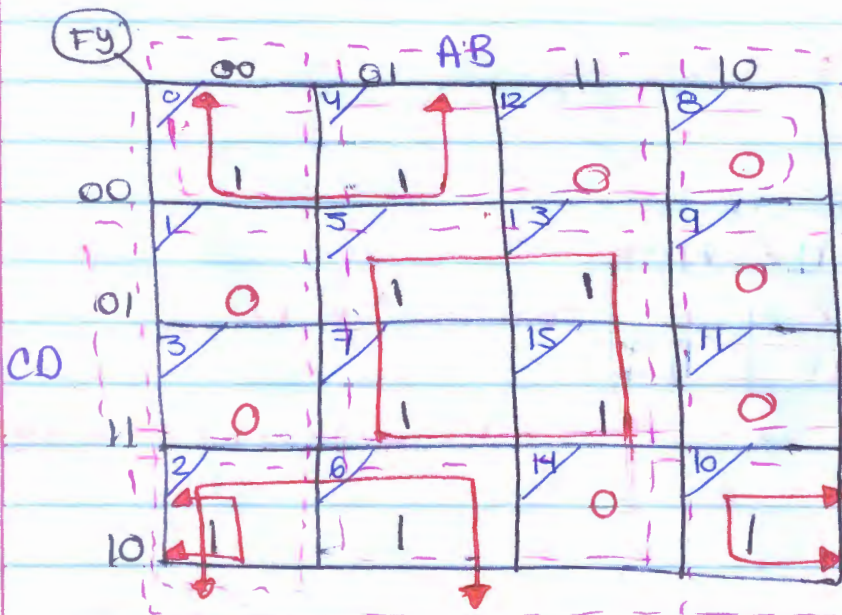


1. Formar el menor número de grupos

2. Cada grupo lo más grande posible

$$F_x(A, B, C, D) = A'BC' + AC'D + A'CD + ABC$$

$$F_y(A, B, C, D) = \prod m(1, 3, 8, 9, 11, 12, 14)$$



1. Formar el menor número de grupos

2. Cada grupo lo más grande posible

$$F_y(A, B, C, D) = B'CD' + BD + A'D'$$

26-Feb-10

$$F_3(A, B, C, D) = \sum m(0, 2, 5, 6, 7, 8, 12, 14)$$

AB

$F_3$	00	01	11	10
00	1		1	1
01		1		
11		1		
10	1	1	1	

AB

$F_3$	00	01	11	10
00	1		1	1
01		1		
11		1		
10	1	1	1	

AB

$F_3$	00	01	11	10
00	1			1
01		1		
11		1		
10	1	1	1	

$$F_3 = A'B'D' + AC'D' + A'BD + BCD'$$

$$F_3 = B'C'D' + A'CD' + A'BD + ABD'$$

26-feb-10

$$F_4(A, B, C) = \prod m(2, 7)$$

		AB			
		00	01	11	10
C	0	1	0	1	1
	1	1	1	0	1

$$F_5(x, y, z, w) = \sum m(0, 2, 7, 8, 10, 12, 13, 14)$$

	00	01	11	10
00	0 1	4 5	12 13	8 9
01	3 0	7 0	15 0	11 0
11	2 0	6 1	14 0	10 0
10	0 1	4 0	12 1	8 1

$$\underline{xw'} + \underline{xy'z'} + \underline{x'yzw} + \underline{y'w'}$$
 Forma SOP

Tomando las ceros

$$w\bar{y} + \bar{x}y\bar{z} + \bar{x}y\bar{w} + xzw$$

$$(y + \bar{w})(x + \bar{y} + z)(x + \bar{y} + w)(\bar{x} + \bar{z} + w)$$
 Forma POS



en el Logic Aid para decir que son ceros se pone una coma al principio.

2-Marzo-10

Siempre debe terminar con punto.

$$F(A, B, C, D) = \Pi m(9, 11, 15)$$

AB

	00	01	11	10
00	0 1	4 1	12 1	8 1
01	1 3	5 7	13 15	9 11
11	2 1	6 1	14 1	10 0
10	1 1	1 1	1 1	1 0

Los unos

$$= \bar{A} + \bar{D} + BC'$$

Los ceros

$$= A\bar{B}D + A\bar{C}D$$

En la columna no cambia

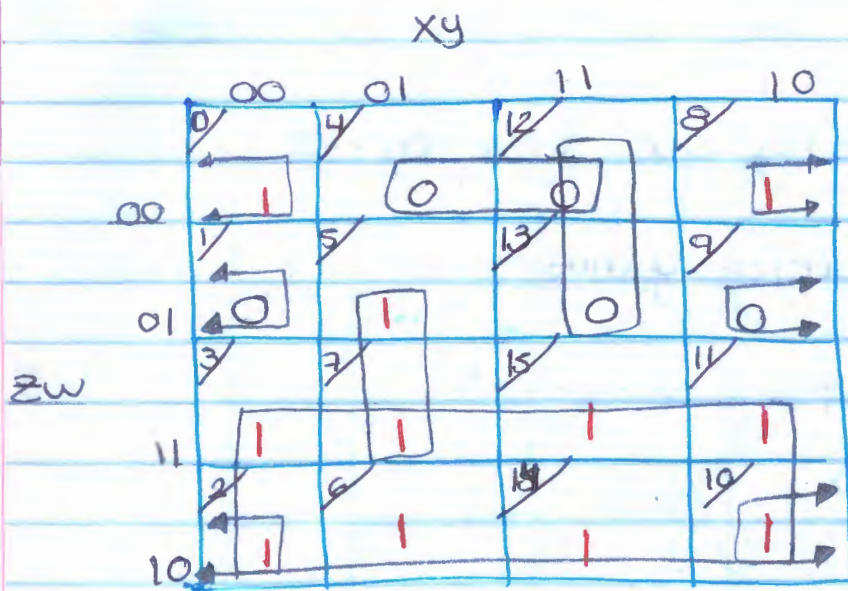
$$= (\bar{A} + B + D)(\bar{A} + C + \bar{D})$$

Esta a no cambio entre 0 y 1 por eso se pone.

Si cambio no se pone

1-Marzo-10

$$F_8(x, y, z, w) = \sum m(0, 2, 3, 5, 6, 7, 8, 10, 11, 14, 15)$$



Unos

$$= y'w' + wx'y + z$$

$$= y'z'w + xyz' + z'w'y$$

$$= (y+z+w')(x'+y'+z)(z+w+y')$$

3-Marzo-10

$$F(x, y, z, w) = \sum m(4, 7, 9, 10, 12, 13, 14, 15)$$

xy

	00	01	11	10
00	0 0	4 1	12 0	8 0
01	1 0	5 0	13 1	9 1
11	3 0	7 1	15 1	11 0
10	2 0	6 0	14 1	10 1

No está  
seguro

Unos

$$y\bar{z}\bar{w} + x\bar{z}w + yz\bar{w} + xz\bar{w}$$

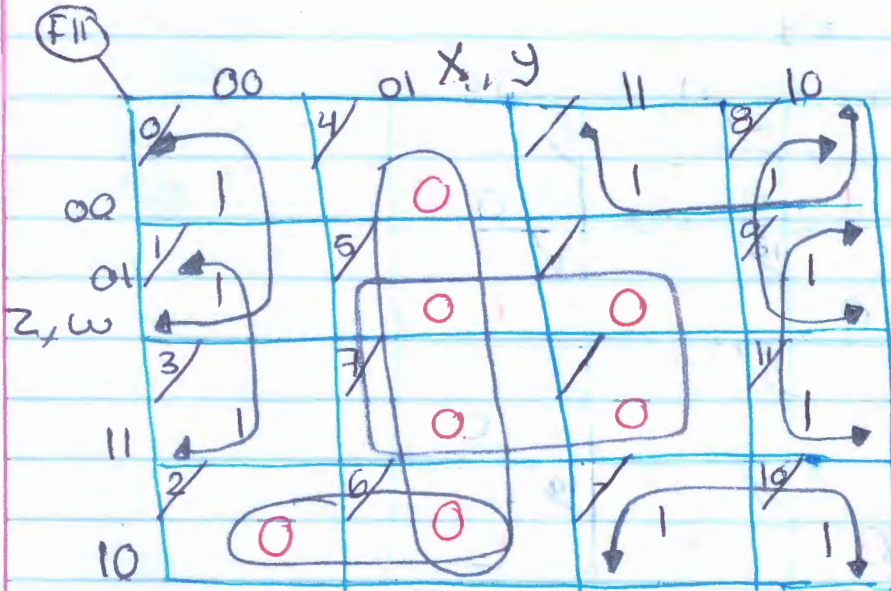
Ceros

$$x\bar{y}z + \bar{x}\bar{z}w + xz\bar{w} + y\bar{z}\bar{w} + yz\bar{w}$$

$$(x+y+\bar{z})(x+z+\bar{w})(x+z+w)(y+\bar{z}+w)(y+\bar{z}+w)$$

3-Marzo-10

$$F_{11}(x, y, z, w) = xy' + xyw' + x'y'w + x'y'z'w' \quad \text{SOP} = \text{MMI} = 1$$



Un grupo de 4 elimina 2 variables

$$F_{11} = \bar{y}z + xw' + y'w$$

Ceros

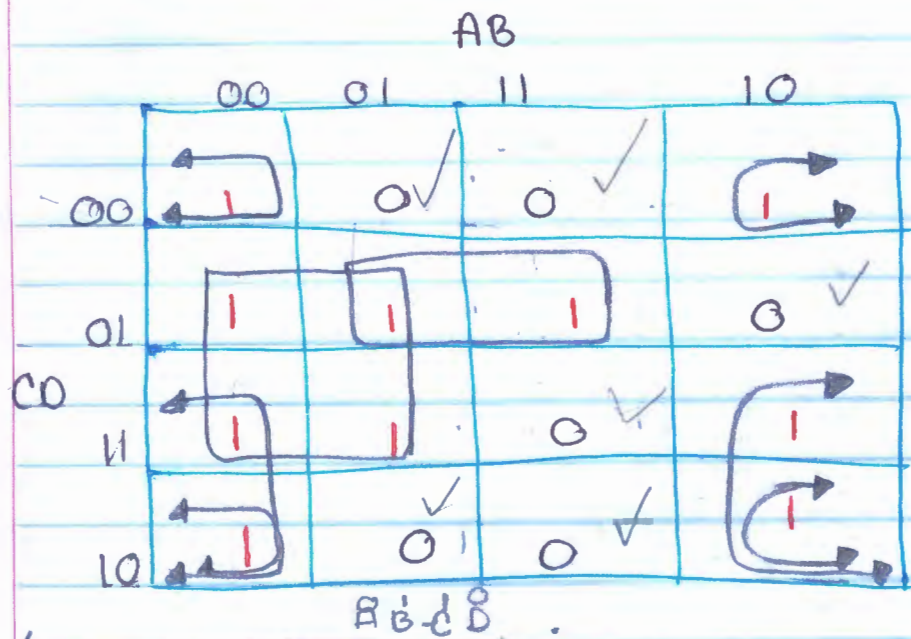
Siempre hay que sacar las 2 ecuaciones de ceros y unos

$$F_{11} = \bar{x}y + yw + \bar{x}z\bar{w}$$

) cambio de signos

$$= (x + \bar{y})(\bar{y} + w)(x + \bar{z} + w)$$

$$F(A, B, C, D) = (B' + C + D)(B' + C' + D)(A' + B' + C' + D')(A' + B + C + D')$$



$$\begin{array}{cccc}
 (\bar{B} + C + D) & (B + \bar{C} + D) & \bar{A} + B + C + \bar{D} & (\bar{A} + \bar{B} + \bar{C} + \bar{D}) \\
 \begin{array}{ccc} 1 & 0 & 0 \\ B & \bar{C} & \bar{D} \end{array} & \begin{array}{ccc} 1 & 1 & 0 \\ B & C & \bar{D} \end{array} & \begin{array}{ccc} 1 & 0 & 0 & 1 \\ A & \bar{B} & \bar{C} & D \end{array} & \begin{array}{ccc} 1 & 1 & 1 & 1 \\ A & B & C & D \end{array}
 \end{array}$$

Unos

$$B\bar{D} + \bar{A}D + B\bar{C}D + \bar{B}C$$

Ceros

$$B\bar{D} + ABC + A\bar{B}\bar{C}D$$

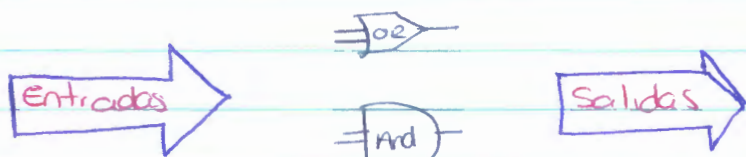
$$(\bar{B} + D)(\bar{A} + \bar{B} + \bar{C})(\bar{A} + B + C + \bar{D})$$

# Sistema Combinacional

22-Marzo-10

## Sistema combinacional

Es aquel bloque digital en donde los valores de salida dependen únicamente de las combinaciones de entrada.



## Bloque digital

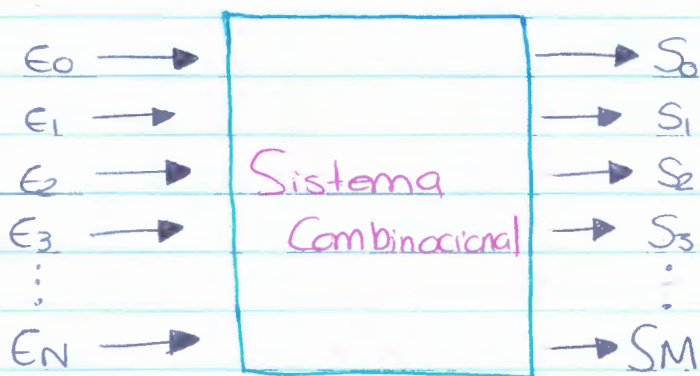
Un sistema combinacional puede estar compuesto de una sola operación.

¿de que depende que la salida de la operación AND valga uno?  
De que sus entradas tengan el valor de uno

La salida depende de las combinaciones de entrada



Un sistema combinacional puede tener una o mas entradas y/o una o mas salidas y el numero de entradas puede ser mayor, menor o igual al numero de salidas



22-Marzo-10

## Metodología del Diseño Combinacional

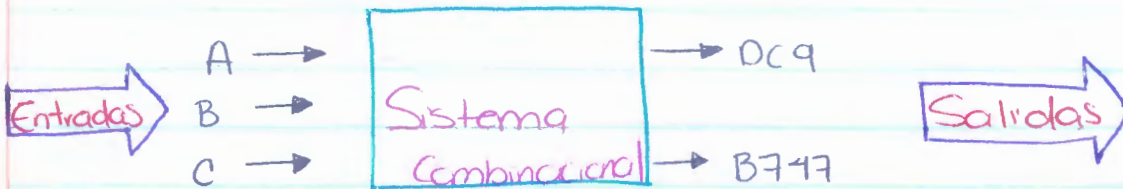
1. Especificar el sistema
2. Determinar entradas y salidas
3. Construir la tabla de Verdad
4. Minimizar
5. Diagrama esquemático
6. Implementar

### 1. Especificar el Sistema

En esta parte se detalla el propósito del diseño ?

### 2. Determinar entradas y salidas

De las variables que intervienen en el problema hay que identificar cuales y cuantas son de entrada y de salida



### 3. Construir la Tabla de verdad

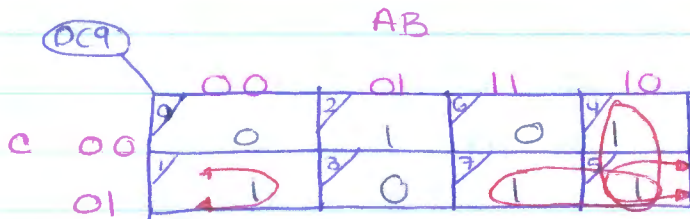
Trasladar el comportamiento del sistema a una Tabla de verdad, indicando para cada combinación de entrada la salida o salidas más convenientes para el diseño

m	A	B	C	F3	F4
0	0	0	0		
1	0	0	1		
2	0	1	0		
3	0	1	1		
4	1	0	0		
5	1	0	1		
6	1	1	0		
7	1	1	1		

22-Marzo-10

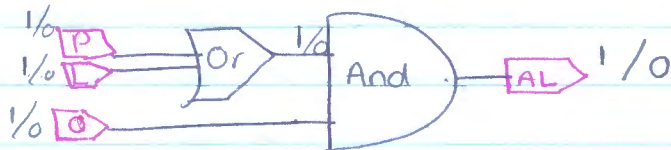
#### 4. Minimizar

Para obtener las ecuaciones minimas se puede utilizar algun metodo de simplificacion como manipulacion algebraica, mapas de Karnaugh, etc.



#### 5. Diagrama esquematico

Despues de haber obtenido las ecuaciones minimas se representa en forma de simbolos para su analisis y comprension



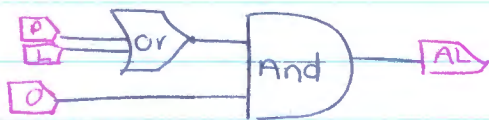
#### 6. Implementar

Se tienen dos opciones para la implementacion

- Circuitos integrados de funcion fija (TTL o CMOS)
- Dispositivos Logicos programables (PLD's)

En los **Dispositivos Logicos Programables (PLD's)** se puede disenar mediante:

- Captura esquematica
- Un Lenguaje de Descripcion de Hardware (HDL)



```
MODULE comp
X=x;" igualdad
*Entradas
A,B pin 1,2;
*En Salidas los signos de - > <
*están reservados para los
*operadores relacionales usaremos
- AeqB en lugar de A=B
- AmyB en lugar de A>B
- AmoB en lugar de A<B
AeqB, AmyB, AmoB pin 19..17 istype 'com';
equations
AeqB= !(A&B);
AmyB= A&!B;
AmoB= !A&B;
test_vectors
{[A,B]->[AeqB, AmyB, AmoB]}
{[0,0]->[X,X,X];
 [0,1]->[X,X,X];
 [1,0]->[X,X,X];
 [1,1]->[X,X,X];
}
END
```





22-Marzo-10

## Ejemplo 1

En una granja se tiene:

- Un granero con una puerta muy grande y pesada en donde se requiere de varias personas para abrirla o cerrarla
- Un corral de ovejas
- Además ocasionalmente llegan lobos

El granero necesita el diseño de un sistema de alarma de modo que:

1. Se active cuando las ovejas estén fuera del corral y la puerta abierta, hacer una acción correctiva ya sea cerrar la puerta del granero o poner las ovejas en su corral.
2. También deberá de activarse la alarma cuando estén los lobos próximos y las ovejas fuera del corral, para hacer la acción correctiva de ahuyentar a los lobos.

### 1. Especificar el sistema

Las variables que intervienen son Puerta, Ovejas, Lobos y la Alarma para las primeras tres se tienen sensores de detección de modo que:

**Puerta** Si esta abierta = 1, Si esta cerrada = 0

**Ovejas** Si están fuera del corral = 1, Si están dentro del corral = 0

**Lobos** Si están próximos = 1, Si están lejos = 0

Para el dispositivo de alarma se considera que:

**Alarma** Se activa con un 1, Se desactiva con un 0

22-Marzo-10

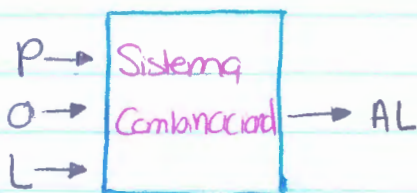
## 2. Determinar entradas y salidas

Podemos decir que:

La puerta, Ovejas y Lobos (P, O y L) son las entradas del sistema.

Mientras que la alarma (AL) es la salida.

Representada a continuación en un diagrama de bloques



## 3. Trasladar el compartimiento a una tabla de verdad.

En este paso hay que decidir el valor de las salidas (0 ó 1) para cada una de las posibles combinaciones de entrada

	m	P	O	L	AL	
Puerta						Las ovejas
Si esta abierta = 1	0	0	0	0	0	esten fuera
Si esta cerrada = 0	1	0	0	1	0	del corral y
Ovejas	2	0	1	0	0	la puerta
Si están fuera del corral = 1	3	0	1	1	1	abierta o
Si están dentro = 0	4	1	0	0	0	los lobos
Lobos	5	1	0	1	0	proximos y le
Si están proximos = 1	6	1	1	0	1	ovejas fuera del
Si están lejos = 0	7	1	1	1	1	corral 1

Alarma

Se activa con un 1

Se desactiva con un 0

Las ovejas esten fuera del corral y la puerta abierta 0

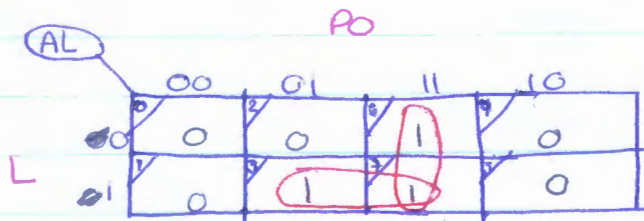
Los lobos proximos y las ovejas

fuera del corral 0

22-Marzo-10

#### 4. Minimizar

Para efectuar la simplificación función AL podemos hacer uso del mapa de Karnaugh agrupando unos (SOP)

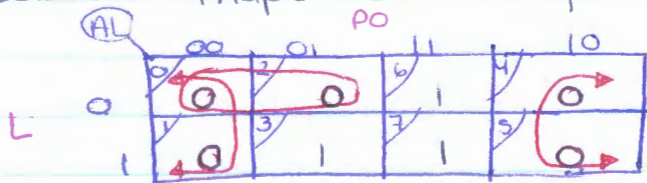


Las ovejas estén fuera del corral y la puerta abierta o los lobos próximos y las ovejas fuera del corral.

$$FAL(P,O,L) = PO + OL$$
$$PO + OL = O(P+L)$$

En lo que podemos concluir que la alarma se activa cuando la puerta está abierta y los ovejas fuera (PO) o también los ovejas fuera y los lobos próximos (OL).

Uso del mapa de Karnaugh agrupando ceros (POS).



$$FAL(P,O,L) = O(P+L)$$

#### 5. Diagrama esquemático

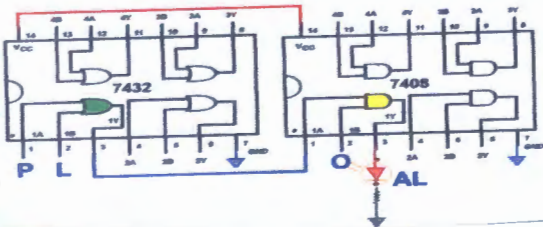


$$FAL(P,O,L) = O(P+L)$$

#### 6. Implementar

La implementación se puede realizar con:

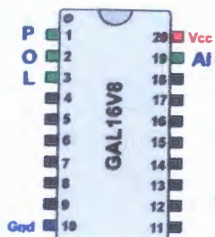
- Circuitos de función fija TTL.



22-Marzo-10

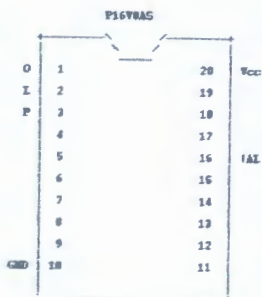
La implementación se puede realizar con:

- Dispositivo Logico Programable (PLD) como el GAL16V8



Para programar un DLP con la función deseada puede ser a través de:

1. Captura esquemática
2. Lenguaje de Descripción de Hardware (HDL)
  - a) Las ecuaciones
  - b) La tabla de verdad
  - c) La descripción del problema



22 - Marzo - 10

# Lenguaje de Descripción de Hardware (HDL)

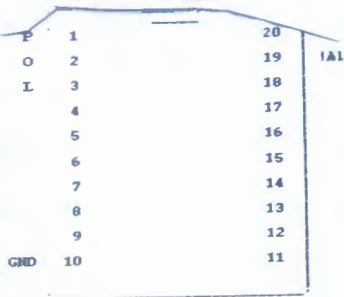
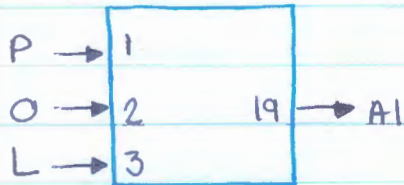
Operador	Descripción	Ecuación en ABEL-HDL	Símbolo
!	NOT	!A	
&	AND	A&B	
#	OR	A#B	
\$	EXOR	A\$B	
!&	NAND	!(A&B)	
!#	NOR	!(A#B)	
!\$	EXNOR	!(A\$B)	

## Ecuación

$$A1 = O \& (P \# L)$$

## Formato Abel-HDL

$$A1 = O \& (P \# L);$$



## Ecuaciones

### MODULE ovejas

"Entradas

P.O.L pin 1.2.3:

"Salida

A1 pin 19 `istype 'com'`;

`equations`

`A1=O&(P#L);`

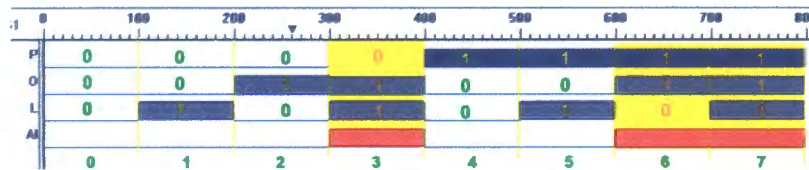
`END`

22-Marzo-10

Archivo en formato ABEL-HDL incluyendo la simulación

```
MODULE ovejas
"entradas
P.O.L pin 1,2,3:
"salida
Al pin 19 istype 'com':
equations
A1=O&(P#L):
test_vectors
((P,O,L)->A1)
[0,0,0]->x.;
[0,0,1]->x.;
[0,1,0]->x.;
[0,1,1]->x.;
[1,0,0]->x.;
[1,0,1]->x.;
[1,1,0]->x.;
[1,1,1]->x.;
END
```

Simulación



Lenguaje de Descripción de Hardware (HDL)

- a) Las ecuaciones
  - b) La Tabla de verdad
  - c) La descripción del problema
- ABEL Advanced Boolean Equation language  
Lenguaje Avanzado de Expresiones Booleanas

Partes de un programa ABEL-HDL

1. **Module** inicio del programa
2. "comentarios" opcional
3. **Declaration** asignación de terminales de entrada y salida
4. **Descripción lógica** (ecuaciones, tabla de verdad, etc.)
5. **Test vectors** (vectores de prueba opcional)
6. **End** fin del programa

22-Marzo-10

## Lenguaje de Descripción de Hardware (HDL) Ecuaciones

```
Text Editor [al.abl]
File Edit View Templates Tools Options Window Help
[Icons]
MODULE al
"este es un archivo en formato ABEL-HDL
"para implementar en un GAL16V8
"el problema de la alarma
" Resuelto por la ecuacion AL=O(P+L)
"Entradas con asignación de terminales
P.O.L pin 1,2,3;
" Salida asignada a la terminal 19
AL pin 19 istype 'com';
equations
AL= O&(P#L);
"se incluye la simulación
Test_vectors
((P,O,L)->AL)
[0,0,0]->.x.;
[0,0,1]->.x.;
[0,1,0]->.x.;
[0,1,1]->.x.;
[1,0,0]->.x.;
[1,0,1]->.x.;
[1,1,0]->.x.;
[1,1,1]->.x.;
END
```

Dc1



22-Marzo-10

## Ejemplo 2

Diseñe un sistema capaz de cubrir las necesidades de control de aterrizaje de un pequeño aeropuerto. Consta de tres pistas llamadas A, B y C.

Aterrizan dos tipos de aviones

Un DC9 que requiere de una sola pista para aterrizar

Un B747 que requiere de dos pistas

El avión B747 tiene prioridad de aterrizar sobre el DC9

### 1. Especificar el Sistema

Las variables que intervienen son:

Pistas A, B y C

Disponibile = 1

No disponible = 0

Aviones DC9 y B747

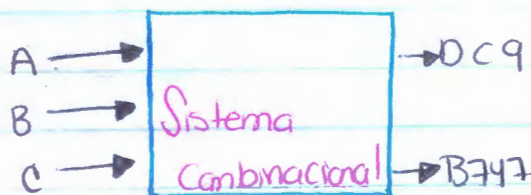
Permiso para aterrizar = 1

No permiso para aterrizar = 0

### 2. Determinar las entradas y salidas

Las pistas A, B, C son las entradas del sistema.

Mientras que permiso para aterrizar para el DC9 o B747 son las salidas que representamos a continuación en un diagrama de bloques.



22-Marzo-10

### 3. Trasladar el comportamiento a una tabla de verdad

Hay que decidir el valor mas convenientes de las salidas (0 ó 1) para cada una de las combinaciones de entrada.

m	A	B	C	Amarillo DC9	Verde B747
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	1	0
6	1	1	0	0	1
7	1	1	1	1	1

### 4. Minimización

DC9

	AB			
c	00	01	11	10
0	0	1	0	1
1	1	0	1	1

$$F_{DC9}(A,B,C) = A'BC' + AB' + AC + B'C$$

B747

	AB			
c	00	01	11	10
0	0	0	1	0
1	0	1	1	0

$$F_{B747}(A,B,C) = AB + BC = B(A+C)$$

Para minimizar podemos hacer uso de los mapas de Karnaugh para simplificar las Funciones DC9 y B747

$$F_{DC9}(A,B,C) = A'BC' + AB' + AC + B'C$$

$$F_{B747}(A,B,C) = AB + BC = B(A+C)$$

22-Marzo-10

- $F_{DC9}(A,B,C) = A'BC' + AB' + AC + B'C$
- $F_{B747}(A,B,C) = AB + BC = B(A+C)$

Haciendo un análisis de las ecuaciones obtenidas para el DC9 puede aterrizar cuando:

$A'BC'$  Solo este disponible la pista B

$AB'$  Cuando este disponible la pista A pero no la B'

$AC$  Cuando estén disponibles las pistas A y C independientemente de la B.

$B'C$  Cuando este disponible la pista C pero no la B'

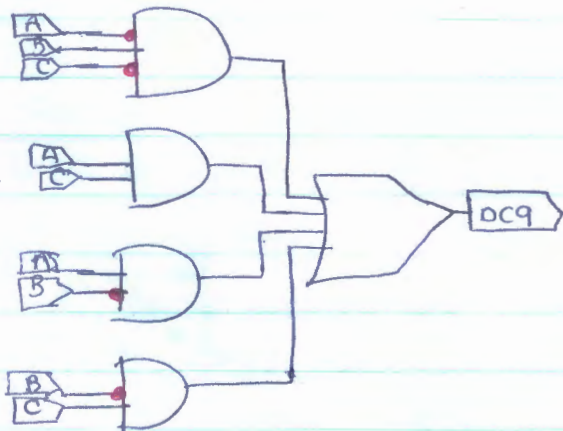
El B747 puede aterrizar cuando

$AB$  Cuando estén disponibles las pistas A y B independientemente de la C

$BC$  Cuando estén disponibles las pistas B y C independientemente de la A.

## 5. Diagrama esquemático

$$F_{DC9}(A,B,C) = A'BC' + AB' + AC + B'C$$



22-Marzo-10

$$F_{B747}(A,B,C) = AB + BC = B(A+C)$$



S Implementación

**MODULE** aviones

Entradas

A,B,C PIN 1..3;

Salidas

DC9,B747 pin 19,18 istype 'com';

**TRUTH Table**

([A,B,C]->[DC9,B747])

[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]->[1.0];

[1.1.0]->[0.1];

[1.1.1]->[1.1];

END

test\_vectors

([A,B,C]->[DC9,B747])

[0.0.0]->[0.0];

[0.0.1]->[1.0];

[0.1.0]->[1.0];

[0.1.1]->[0.1];

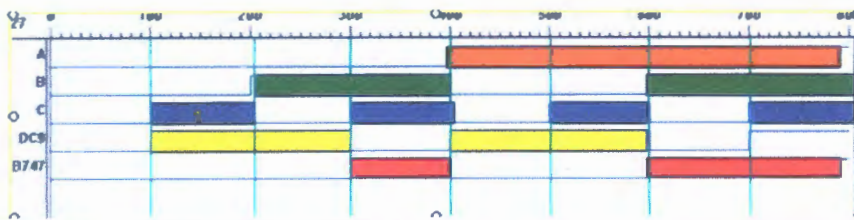
[1.0.0]->[1.0];

[1.0.1]->[1.0];

[1.1.0]->[0.1];

[1.1.1]->[1.1];

END



22-Marzo-10

## Proyecto Adicional 1

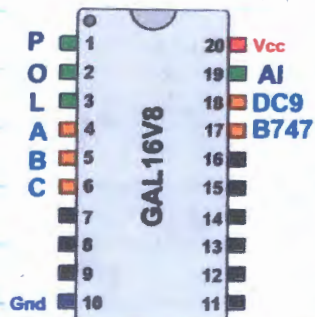
P/ Martes 13-Abril

Implementar los ejemplos de:

- La alarma y los aviones en un solo circuito integrado
- Utilizando lenguaje de descripción de Hardware ABEL-ADL con los comando utilizando **Equations** o **Truth-table**
- No necesariamente trabajaron en forma simultanea

### Reporte

1. Portada
2. Redacción del problema
3. Diagrama de bloques (entradas y Salidas)
4. Tabla de verdad
5. Código ABEL
6. Simulación
7. Archivo RPT (ecuaciones y pin out)
8. Foto del circuito
9. Conclusiones y recomendaciones (tiempo estimado)



DC2

1110 - Si el número es par es par  
1010  
1000

si termina en 0 = par  
si termina en 1 = impar

## Metodología del Diseño Combinacional

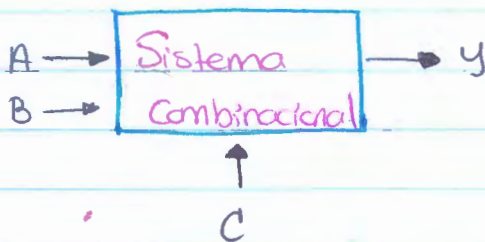
1. Especificar el sistema
2. Determinar entradas y salidas
- ~~3. Construir la Tabla de Verdad~~
- ~~4. Minimizar~~
- ~~5. Diagrama esquemático~~
6. Implementar

### Ejemplo 3

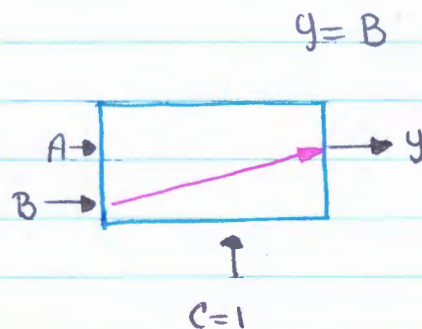
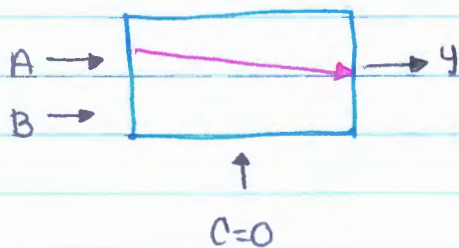
Diseñar un Multiplexor (selector datos) de 2 a 1 línea  
Data Selectors / Multiplexers 2-Line To 1-Line

Das entradas de datos A y B.

Una entrada de control C, Una salida y



Si  $C=0$  entonces la salida  $y=A$  Si  $C=1$  entonces la salida



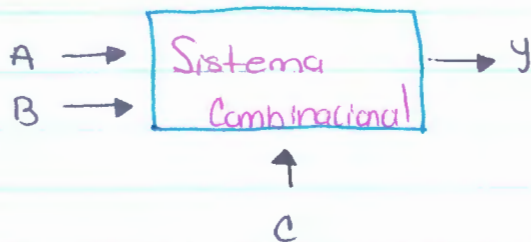
24-Marzo-10

### 1. Especificar el Sistema

En la redacción del problema está acbrado el proposito y las variables que intervienen en el problema.

Si  $C=0$  entonces la salida  $y=A$

Si  $C=1$  entonces la salida  $y=B$



### 2. Determinar entradas y salidas



### 3. Tabla de verdad

m	C	A	B	y
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

$C=0, y=A$

$C=1, y=B$

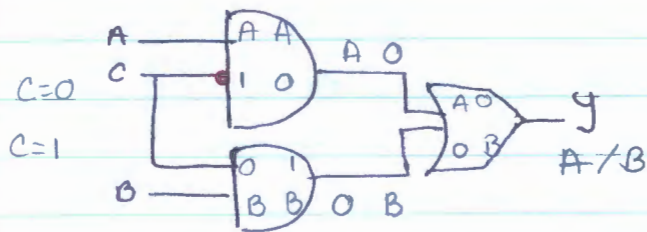
### 4. Obtener las ecuaciones minimas

$$F_y(C, A, B) = C'A + CB$$

24-Marzo-10

## 5. Diagrama esquemático

$$F_y(C, A, B) = C'A + CB$$



## 6. Implementación en ABEL-HDL por ecuaciones

$$F_y(C, A, B) = C'A + CB$$

```
MODULE muxeq
"entradas
A,B,C pin 1,2,3;
"Salida
Y pin 19 istype 'com';
equations
Y = !C&A#C&B;
END
```

## ABEL-HDL Por tabla de verdad

```
MODULE muxtt
"entradas
A,B,C pin 1,2,3;
"Salida
Y pin 19 istype 'com';
Truth_table
([C,A,B]->Y)
[0.0.0]->0;
[0.0.1]->0;
[0.1.0]->1;
[0.1.1]->1;
[1.0.0]->0;
[1.0.1]->1;
[1.1.0]->0;
[1.1.1]->1;
END
```



24-Marzo-10

## When, Then, Else

**WHEN !C THEN**  $y=A$ ;

"Si (WHEN)  $c=0$  (!C) entonces (THEN) la salida  $y=A$

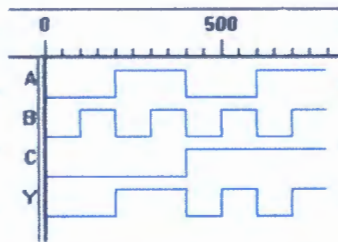
**WHEN C THEN**  $y=B$ ;

"Si (WHEN)  $c=1$  (C) entonces (THEN) la salida  $y=B$

**WHEN !C THEN**  $y=A$  **else**  $y=B$ ;

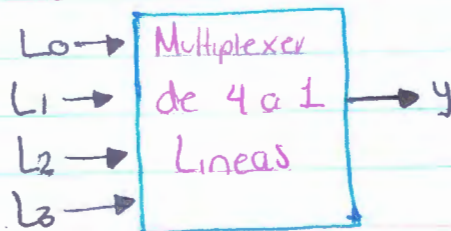
```
MODULE muxwte
"entradas
A,B,C pin 1,2,3;
"Salida
Y pin 19 istype 'com';
equations
WHEN !C THEN Y=A else Y=B;
END
```

```
MODULE muxwte
"entradas
A,B,C pin 1,2,3;
"Salida
Y pin 19 istype 'com';
equations
WHEN !C THEN Y=A else
Y=B;
Test_vectors
([C,A,B]>Y)
[0,0,0]>0:
[0,0,1]>0:
[0,1,0]>1:
[0,1,1]>1:
[1,0,0]>0:
[1,0,1]>1:
[1,1,0]>0:
[1,1,1]>1:
END
```



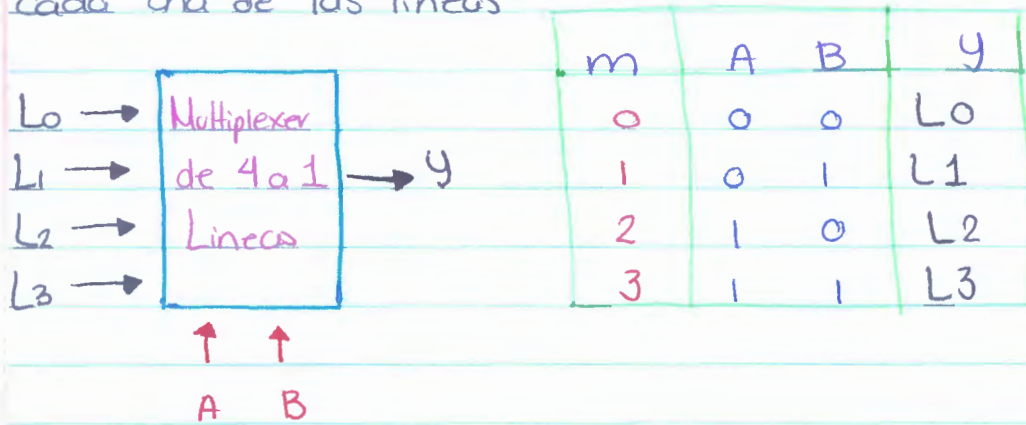
## Multiplexers de 4 a 1 línea Data Selectors/Multiplexers

4-Line To 1-Line

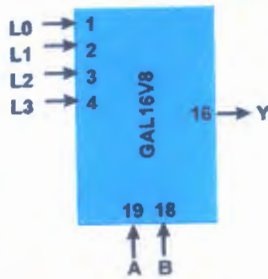


23-Marco-10

Cuántas entradas de control se requieren para seleccionar cada una de las líneas



m	A	B	y
0	0	0	L0
1	0	1	L1
2	1	0	L2
3	1	1	L3



Elabore el archivo en ABEL-HDL usando los comandos When, Then, Else

When, Then, Else

m	A	B	y
0	0	0	L0
1	0	1	L1
2	1	0	L2
3	1	1	L3

```

MODULE muxeq
"entradas
A,B,L0..L3 pin 19,18,1..4;
"Salida
Y pin 16 istype 'com';
equations
WHEN !A&!B THEN Y=L0;
WHEN !A&B THEN Y=L1;
WHEN A&!B THEN Y=L2;
WHEN A&B THEN Y=L3;
END
    
```

24-Marco-10

Para no listar las 64 combinaciones Usamos el Don't Care .X.

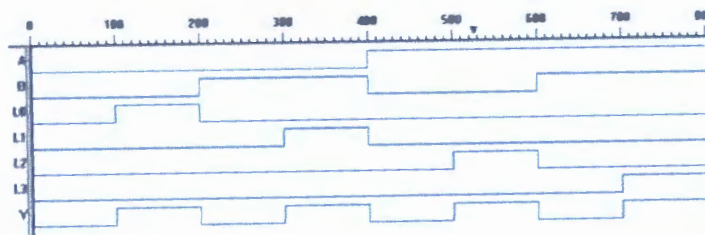
```
X=.x.;
Test_vectors
([A,B,L3,L2,L1,L0]->[Y])
[0,0,X,X,X,0]->[0];
[0,0,X,X,X,1]->[1];
[0,1,X,X,0,X]->[0];
[0,1,X,X,1,X]->[1];
[1,0,X,0,X,X]->[0];
[1,0,X,1,X,X]->[1];
[1,1,0,X,X,X]->[0];
[1,1,1,X,X,X]->[1];
```

```
X=.x.;
Test_vectors
([A,B,L3,L2,L1,L0]->[Y])
[0,0,X,X,X,0]->[0];
[0,0,X,X,X,1]->[1];
[0,1,X,X,0,X]->[0];
[0,1,X,X,1,X]->[1];
[1,0,X,0,X,X]->[0];
[1,0,X,1,X,X]->[1];
[1,1,0,X,X,X]->[0];
[1,1,1,X,X,X]->[1];
```

```
X=.x.;
Test_vectors
([A,B,L3,L2,L1,L0]->[Y])
[0,0,X,X,X,0]->[0];
[0,0,X,X,X,1]->[1];
[0,1,X,X,0,X]->[0];
[0,1,X,X,1,X]->[1];
[1,0,X,0,X,X]->[0];
[1,0,X,1,X,X]->[1];
[1,1,0,X,X,X]->[0];
[1,1,1,X,X,X]->[1];
```

```
MODULE mux
X=.x.;
"entradas
A,B,L0..L3 pin 19,18,1..4;
"Salida
Y pin 16 istype 'com';
equations
WHEN !A&!B THEN Y=L0;
WHEN !A&B THEN Y=L1;
WHEN A&!B THEN Y=L2;
WHEN A&B THEN Y=L3;
```

```
Test_vectors
([A,B,L3,L2,L1,L0]->[Y])
[0,0,X,X,X,0]->[0];
[0,0,X,X,X,1]->[1];
[0,1,X,X,0,X]->[0];
[0,1,X,X,1,X]->[1];
[1,0,X,0,X,X]->[0];
[1,0,X,1,X,X]->[1];
[1,1,0,X,X,X]->[0];
[1,1,1,X,X,X]->[1];
END
```

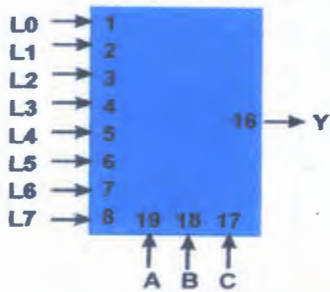


23-Marco-10

# Multiplexar de 8 a 1 linea

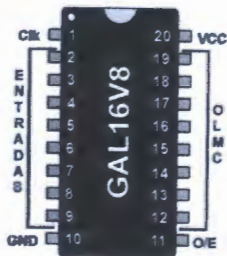
When, Then, Else

m	A	B	C	y	L0	L1	L2	L3	L4	L5	L6	L7
0	0	0	0	L0	L0	L1	L2	L3	L4	L5	L6	L7
1	0	0	1	L1	L0	L1	L2	L3	L4	L5	L6	L7
2	0	1	0	L2	L0	L1	L2	L3	L4	L5	L6	L7
3	0	1	1	L3	L0	L1	L2	L3	L4	L5	L6	L7
4	1	0	0	L4	L0	L1	L2	L3	L4	L5	L6	L7
5	1	0	1	L5	L0	L1	L2	L3	L4	L5	L6	L7
6	1	1	0	L6	L0	L1	L2	L3	L4	L5	L6	L7
7	1	1	1	L7	L0	L1	L2	L3	L4	L5	L6	L7



```

MODULE muxeq
" Entradas de datos
L0..L7 pin 1..8;
" Entradas de control
A,B,C pin 19,18,17;
" Salida
Y pin 16 istype 'com';
Equations
WHEN !A&!B&!C THEN Y=L0;
WHEN !A&!B&C THEN Y=L1;
WHEN !A&B&!C THEN Y=L2;
WHEN !A&B&C THEN Y=L3;
WHEN A&!B&!C THEN Y=L4;
WHEN A&!B&C THEN Y=L5;
WHEN A&B&!C THEN Y=L6;
WHEN A&B&C THEN Y=L7;
END
  
```



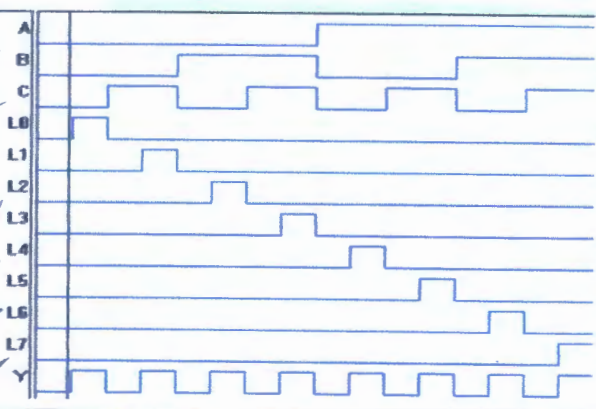
23-Marzo-10

# Test - vectors de Multiplexor de 8 a 1 linea.

## Test vectors

([A,B,C,L7,L6,L5,L4,L3,L2,L1,L0]->[Y])

- [0,0,0,X,X,X,X,X,0]->[0];
- [0,0,0,X,X,X,X,X,1]->[1];
- [0,0,1,X,X,X,X,X,0,X]->[0];
- [0,0,1,X,X,X,X,X,1,X]->[1];
- [0,1,0,X,X,X,X,0,X,X]->[0];
- [0,1,0,X,X,X,X,1,X,X]->[1];
- [0,1,1,X,X,X,0,X,X,X]->[0];
- [0,1,1,X,X,X,1,X,X,X]->[1];
- [1,0,0,X,X,X,0,X,X,X]->[0];
- [1,0,0,X,X,X,1,X,X,X]->[1];
- [1,0,1,X,X,0,X,X,X,X]->[0];
- [1,0,1,X,X,1,X,X,X,X]->[1];
- [1,1,0,X,0,X,X,X,X,X]->[0];
- [1,1,0,X,1,X,X,X,X,X]->[1];
- [1,1,1,0,X,X,X,X,X,X]->[0];
- [1,1,1,1,X,X,X,X,X,X]->[1];



## Test vectors

([A,B,C,L7,L6,L5,L4,L3,L2,L1,L0]->[Y])

- [0,0,0,X,X,X,X,X,X,0]->[0];
- [0,0,0,X,X,X,X,X,X,1]->[1];
- [0,0,1,X,X,X,X,X,0,X]->[0];
- [0,0,1,X,X,X,X,X,1,X]->[1];
  
- [1,1,1,0,X,X,X,X,X,X]->[0];
- [1,1,1,1,X,X,X,X,X,X]->[1];

# Proyecto Adicional 2

DC3

Multiplexor de 8 - 1 linea para viernes 16 Abril

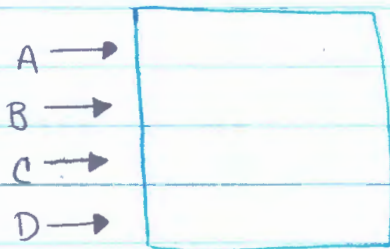
24-Marzo-10

### Ejemplo 4

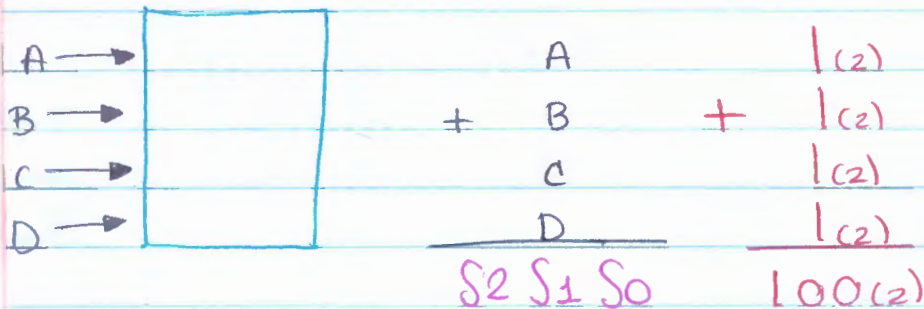
Diseñe un Sumador Binario de cuatro números de un bit cada número

1, 2. Especificar el problema, definir entradas y salidas

Los cuatro números binarios los llamaremos A, B, C y D respectivamente y representan la entrada dis sistema combinatoria



Cuántas son las salidas mínimas requeridas 3  $S_0, S_1, S_2$

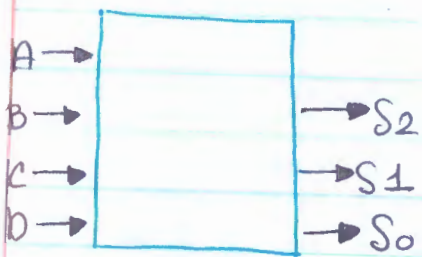


El resultado mas grande se obtiene cuando todas las entradas A, B, C y D tienen el valor de uno y tendríamos  $1+1+1+1$

En donde el resultado es 4 y expresado en binario 100 se requieren 3 bits, los llamaremos  $S_2, S_1$  y  $S_0$

24-Marzo-10

### 3. Tabla de Verdad



m	A	B	C	D	S2	S1	S0	
0	0	0	0	0	0	0	0	0(10)
1	0	0	0	1	0	0	1	1(10)
2	0	0	1	0	0	0	1	1(10)
3	0	0	1	1	0	1	0	2(10)
4	0	1	0	0	0	0	1	1(10)
5	0	1	0	1	0	1	0	2(10)
6	0	1	1	0	0	1	0	2(10)
7	0	1	1	1	0	1	1	3(10)
8	1	0	0	0	0	0	1	1(10)
9	1	0	0	1	0	1	0	2(10)
10	1	0	1	0	0	1	0	2(10)
11	1	0	1	1	0	1	1	3(10)
12	1	1	0	0	0	1	0	2(10)
13	1	1	0	1	0	1	1	3(10)
14	1	1	1	0	0	1	1	3(10)
15	1	1	1	1	1	0	0	4(10)

$$\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ + 0 & 0 & 1 \\ \hline 0 & 1 & 0 \end{array}$$

$$0(10) = 00(10) \quad 1(10) = 00(10)$$

$$\begin{array}{cc} 0 & 0 \\ 0 & 1 \\ + 1 & 0 \\ \hline 1 & 0 \end{array}$$

$$2(10) = 01(10) \quad 1(10) = 00(10)$$

$$\begin{array}{cc} 0 & 1 \\ 1 & 1 \\ + 1 & 1 \\ \hline 1 & 1 \end{array}$$

$$3(10) = 01(10) \quad 4(10) = 100(10)$$

4. Ecuaciones Minimales  
 $S_2(A, B, C, D) = ?$   
 $S_2(A, B, C, D) = ABCD$

24-Marzo-10

### 4. Ecuaciones Minimales

$S1(A,B,C,D) = ?$

31

		00	01	11	10
00	0	0	0	1	0
01	1	0	1	1	1
11	3	1	1	0	1
10	2	0	0	1	1

Usar LogicAid para obtener todas las posibles soluciones

1	$S1 = ABD' + AB'C + AC'D + BCD' + B'CD + A'CD$
2	$S1 = ABD' + AB'C + AC'D + BCD' + A'BD + B'CD$
3	$S1 = ABD' + AB'C + AC'D + BCD' + A'BD + A'CD$
4	$S1 = ABD' + AB'C + AC'D + A'BC + A'BD + B'CD$
5	$S1 = ABD' + AB'C + AC'D + A'BC + A'BD + A'CD$
6	$S1 = ABD' + AB'C + AC'D + A'BC + B'CD + B'CD$
7	$S1 = ABD' + AB'C + AC'D + A'BC + B'CD + A'CD$
8	$S1 = ABD' + ACD' + AC'D + BCD' + A'BD + B'CD$
9	$S1 = ABD' + ACD' + AC'D + A'BC + A'BD + B'CD$
10	$S1 = ABD' + ACD' + AC'D + A'BC + B'CD + B'CD$
11	$S1 = ABD' + AB'C + A'BD + BCD' + B'CD + A'CD$
12	$S1 = ABD' + AB'C + A'BD + A'BC + B'CD + B'CD$
13	$S1 = ABD' + AB'C + A'BD + A'BC + B'CD + A'CD$
14	$S1 = ABD' + ACD' + A'BD + BCD' + B'CD + A'CD$
15	$S1 = ABD' + ACD' + A'BD + A'BC + B'CD + B'CD$
16	$S1 = ABD' + ACD' + A'BD + A'BC + B'CD + A'CD$

17	$S1 = ABC' + ACD' + A'BD + BCD' + B'CD + A'CD$
18	$S1 = ABC' + ACD' + A'BD + BCD' + A'BD + B'CD$
19	$S1 = ABC' + ACD' + A'BD + BCD' + A'BD + A'CD$
20	$S1 = ABC' + ACD' + A'BD + A'BC + A'BD + B'CD$
21	$S1 = ABC' + ACD' + A'BD + A'BC + A'BD + A'CD$
22	$S1 = ABC' + ACD' + A'BD + A'BC + B'CD + B'CD$
23	$S1 = ABC' + ACD' + A'BD + A'BC + B'CD + A'CD$
24	$S1 = ABC' + ACD' + AC'D + BCD' + A'BD + B'CD$
25	$S1 = ABC' + ACD' + AC'D + A'BC + A'BD + B'CD$
26	$S1 = ABC' + ACD' + AC'D + A'BC + B'CD + B'CD$
27	$S1 = ABC' + AB'C + AC'D + BCD' + B'CD + A'CD$
28	$S1 = ABC' + AB'C + AC'D + BCD' + A'BD + B'CD$
29	$S1 = ABC' + AB'C + AC'D + BCD' + A'BD + A'CD$
30	$S1 = ABC' + AB'C + A'BD + BCD' + B'CD + A'CD$
31	$S1 = ABC' + AB'C + A'BD + BCD' + A'BD + B'CD$
32	$S1 = ABC' + AB'C + A'BD + BCD' + A'BD + A'CD$

$S0(A,B,C,D) = ?$

50

		00	01	11	10
00	0	0	1	0	1
01	1	1	0	1	0
11	3	0	1	0	1
10	2	1	0	1	0

Se forman ocho grupos de un uno

$S0(A,B,C,D) = A'B'C'D + A'B'CD' + A'BC'D + A'BCD + ABC'D + ABCD' + AB'C'D' + AB'CD$



24 Marzo-10

$$F_{SO(A,B,C,D)} = A'B'C'D + A'B'CD' + A'BC'D' + A'BCD + ABC'D + ABCD' + AB'C'D + AB'CD$$

Se puede efectuar una simplificación de la función buscando llegar a un Exor ya que los grupos están en diagonal.

$$\begin{aligned} &A'B'(C'D + CD') + A'B(C'D' + CD) + AB(C'D + CD') + AB'(C'D' + CD) \\ &A'B'(C \oplus D) + A'B(C \oplus D)' + AB(C \oplus D) + AB'(C \oplus D)' \\ &(C \oplus D)(A'B' + AB) + (C \oplus D)'(A'B + AB') \\ &(C \oplus D)(A \oplus B)' + (C \oplus D)'(A \oplus B) = A \oplus B \oplus C \oplus D \end{aligned}$$

El resultado es un Exor de cuatro entradas

$$SO(A,B,C,D) = A \oplus B \oplus C \oplus D$$

El resultado es un Exor de cuatro entradas

### Preguntas

1. ¿Qué es lo que determina si un número binario es par o impar?

$$100_{(2)} = 4 \quad 101_{(2)} = 5$$

$$\text{|||||} 100_{(2)} = \text{par}$$

2. ¿Cuándo es verdadero el resultado de un Exor

Cuando un número impar de variables de entrada son una

24-Marzo-10

**MODULE sum**

\*Entradas

A,B,C,D PIN 1..4;

\*Salidas

S2,S1,S0 PIN 19..17 ISTYPE 'COM';

S=[S2,S1,S0];

TRUTH TABLE

(A,B,C,D)→(S)

[0,0,0,0]→[0];

[0,0,0,1]→[1];

[0,0,1,0]→[1];

[0,0,1,1]→[2];

[0,1,0,0]→[1];

[0,1,0,1]→[2];

[0,1,1,0]→[2];

[0,1,1,1]→[3];

[1,0,0,0]→[1];

[1,0,0,1]→[2];

[1,0,1,0]→[2];

[1,0,1,1]→[3];

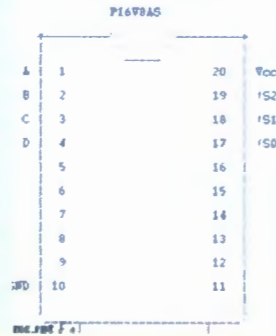
[1,1,0,0]→[2];

[1,1,0,1]→[3];

[1,1,1,0]→[3];

[1,1,1,1]→[4];

End



S2 = ( D & C & B & A );

S1 = ( D & C & B & A # D & C & B & A # D & C & B & A # D & C & B & A # D & C & B & A # D & C & B & A # D & C & B & A # D & C & B & A );

S0 = ( D & C & B & A # D & C & B & A # D & C & B & A # D & C & B & A # D & C & B & A # D & C & B & A # D & C & B & A # D & C & B & A );

# D & C & B & A # D & C & B & A # D & C & B & A # D & C & B & A # D & C & B & A # D & C & B & A # D & C & B & A # D & C & B & A );

**Ejemplo 5**

Diseñe un sistema combinatorial capaz de:

a) Sumar dos numeros Binarios de dos bits cada numero.

b) Multiplicar dos numeros binarios de dos bits cada numero

c) Restar dos numeros binarios de dos bits cada numero, en este ejemplo incluya una salida para indicar el signo de la diferencia si es positivo o nulo = a cero y si la diferencia es negativa = 1

a) Sumar dos numeros Binarios de dos bits cada numero

$\begin{array}{r} A_1 A_0 \\ + B_1 B_0 \\ \hline S_2 S_1 S_0 \end{array}$	$\begin{array}{r} 11(2) \\ + 11(2) \\ \hline 110(2) \end{array}$	$\begin{array}{r} 3(10) \\ + 3(10) \\ \hline 6(10) \end{array}$
---	--	---



24-Marc-10

MODULE todos

"Entradas

A1,A0,B1,B0 pin 1..4;

"Salidas de la suma

S2..S0 pin 19..17 istype 'com';

A=[A1,A0];

B=[B1,B0];

truth\_table

([A,B]->[S2,S1,S0])

[0.0]->[0.0.0];

[0.1]->[0.0.1];

[0.2]->[0.1.0];

[0.3]->[0.1.1];

[1.0]->[0.0.1];

[1.1]->[0.1.0];

[1.2]->[0.1.1];

[1.3]->[1.0.0];

[2.0]->[0.1.0];

[2.1]->[0.1.1];

[2.2]->[1.0.0];

[2.3]->[1.0.1];

[3.0]->[0.1.1];

[3.1]->[1.0.0];

[3.2]->[1.0.1];

[3.3]->[1.1.0];

END

MODULE todos

"entradas

A1,A0,B1,B0 pin 1..4;

"Salidas de la suma

S2..S0 pin 19..17 istype 'com';

A=[A1,A0];

B=[B1,B0];

Z=[S2,S1,S0];

truth\_table

([A,B]->Z)

[0.0]->0;

[0.1]->1;

[0.2]->2;

[0.3]->3;

[1.0]->1;

[1.1]->2;

[1.2]->3;

[1.3]->4;

[2.0]->2;

[2.1]->3;

[2.2]->4;

[2.3]->5;

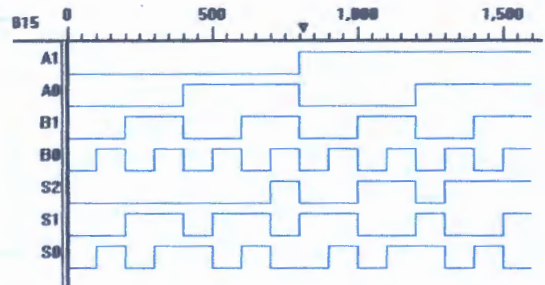
[3.0]->3;

[3.1]->4;

[3.2]->5;

[3.3]->6;

END



MODULE todos

"entradas

A1,A0,B1,B0 pin 1..4;

"Salidas de la suma

M3..M0 pin 19..16 istype 'com';

A=[A1,A0];

B=[B1,B0];

M=[M3..M0];

truth\_table

([A,B]->[M])

[0.0]->[0];

[0.1]->[0];

[0.2]->[0];

[0.3]->[0];

[1.0]->[0];

[1.1]->[1];

[1.2]->[2];

[1.3]->[3];

[2.0]->[0];

[2.1]->[2];

[2.2]->[4];

[2.3]->[6];

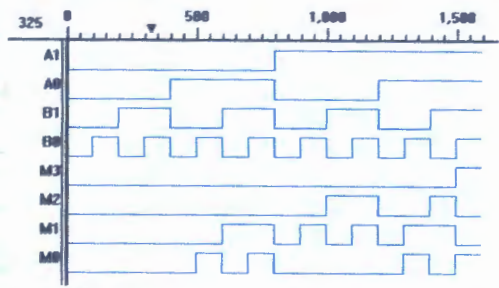
[3.0]->[0];

[3.1]->[3];

[3.2]->[6];

[3.3]->[9];

End



24-Marzo-10

MODULE todos

"entradas

A1,A0,B1,B0 pin 1..4;

"Salidas de la suma

S,R1,R0 pin 19..17 istype 'com';

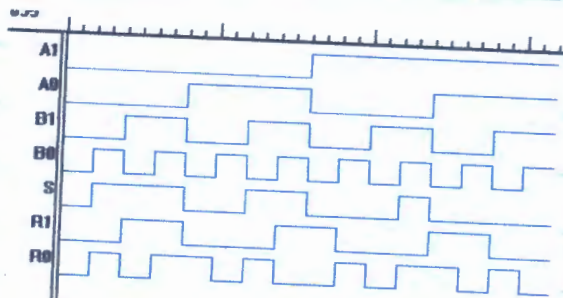
A=[A1,A0];

B=[B1,B0];

R=[R1,R0];

```

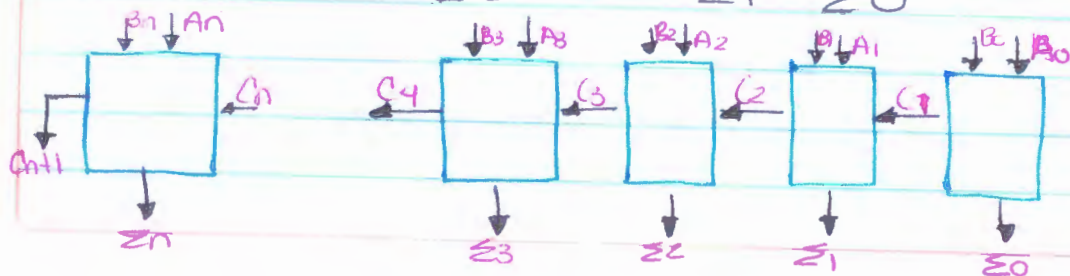
truth_table
([A,B]->[S,R])
[0.0]->[0.0]:
[0.1]->[1.1]:
[0.2]->[1.2]:
[0.3]->[1.3]:
[1.0]->[0.1]:
[1.1]->[0.0]:
[1.2]->[1.1]:
[1.3]->[1.2]:
[2.0]->[0.2]:
[2.1]->[0.1]:
[2.2]->[0.0]:
[2.3]->[1.1]:
[3.0]->[0.3]:
[3.1]->[0.2]:
[3.2]->[0.1]:
[3.3]->[0.0]:
end
    
```



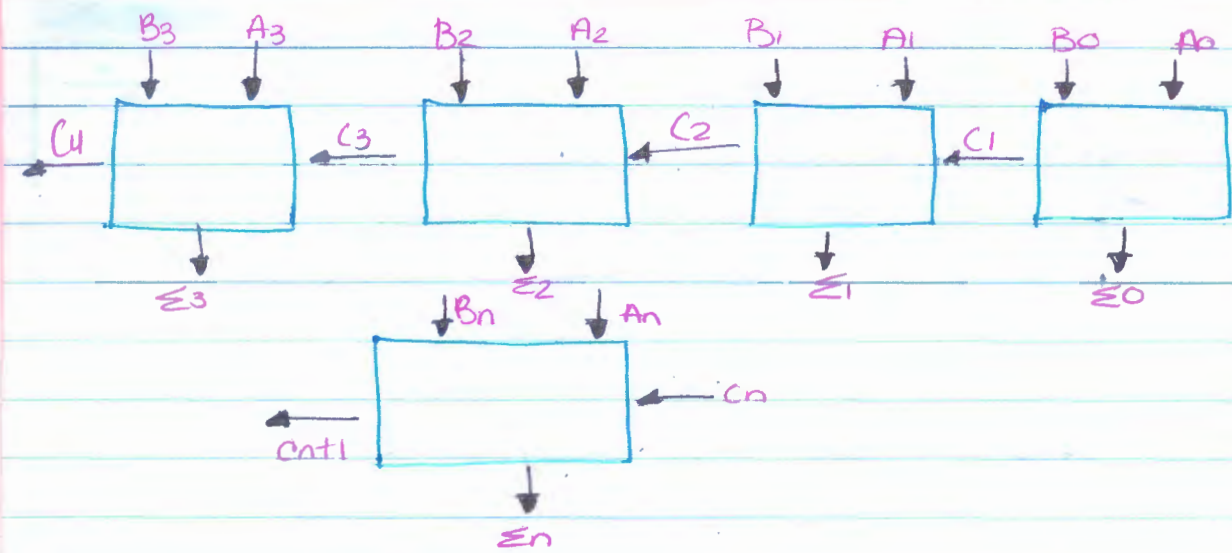
Diseñe un sistema combinatorial capaz de sumar 2 números binarios de n Bits cada numero

n Bits=?

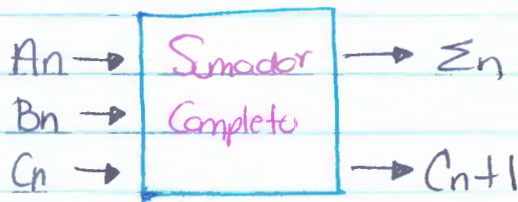
$$\begin{array}{r}
 C_n \qquad \qquad \qquad C_3 \quad C_2 \quad C_1 \\
 + \quad A_n \dots \dots \dots A_3 \quad A_2 \quad A_1 \quad A_0 \\
 B_n \dots \dots \dots B_3 \quad B_2 \quad B_1 \quad B_0 \\
 \hline
 C_{n+1} \quad \Sigma_n \qquad \qquad \Sigma_3 \quad \Sigma_2 \quad \Sigma_1 \quad \Sigma_0
 \end{array}$$



24-Marzo-10



### Diagrama de bloques



### Tabla de verdad

m	$C_n$	$A_n$	$B_n$	$C_{n+1}$	$\Sigma_n$
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	1

24 Marzo - 10

### Ecuaciones minimas

$Cn$   $Cn \ A_n$

	00	01	11	10
$B_n$ 0	0	0	1	0
1	0	1	1	1

$$F_{Cn}(A_n, B_n, C_n) = A_n B_n + A_n C_n + B_n C_n$$

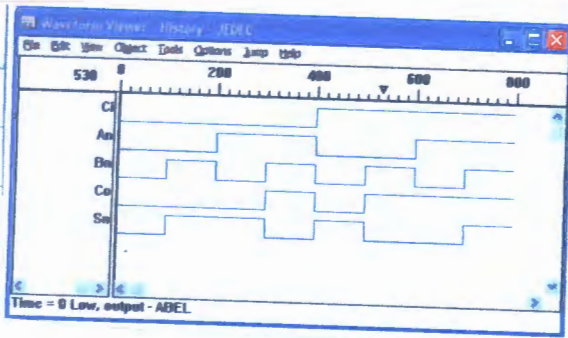
$Cn$   $Cn \ A_n$

	00	01	11	10
$B_n$ 0	0	1	0	1
1	1	0	1	0

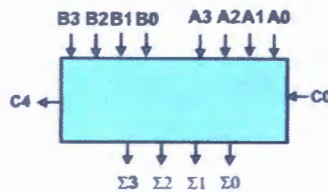
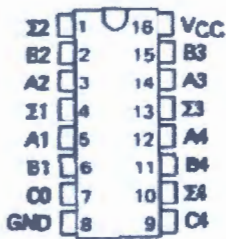
$$F_{S_n}(A_n, B_n, C_n) = A \oplus B \oplus C$$

```
MODULE sumc
"entradas
An, Bn, Ci pin 1..3 :
"salidas
Sn, Co pin 19,18 istype 'com':
Equations
Sn=An$Bn$Ci;
Co=An&Bn#An&Ci#Bn&Ci;
declarations
".set
R=[Ci, An, Bn];
x=.x.;
test_vectors
(R->[Sn, Co])
0->[x, x];
1->[x, x];
2->[x, x];
3->[x, x];
4->[x, x];
5->[x, x];
6->[x, x];
7->[x, x];
END
```

24-Marco-10

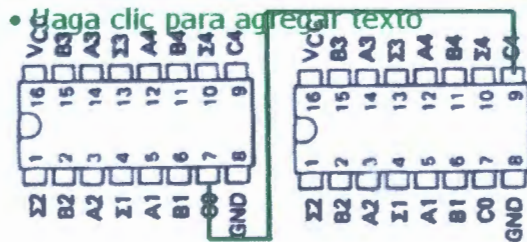
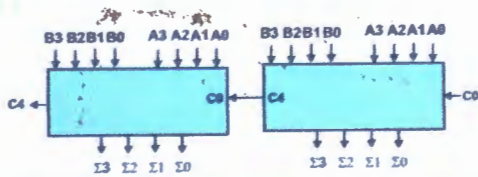


SN74283 | P/Lines



SN74283

Sumador de dos numeros de 8 bits cada numero





24-Marzo-10

## Reporte para esta Actividad

1. Portada
  2. Redacción del Problema
  3. Diagrama de bloques (entradas y salidas)
  4. Tabla de Verdad
  5. Código ABEL
  6. Simulación
  7. Archivo EPT (ecuaciones y pin cut)
  8. Foto del circuito
  9. Conclusiones y Recomendaciones
- } Hoja de datos  
del fabricante  
[www.ti.com](http://www.ti.com)

Dc4

## Proyecto adicional 3

Sumador de 2 números binarios de 8 bits C/n (2S74283)

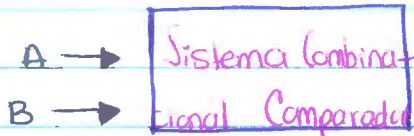
Comparador de 2 números binarios de 8 bits C/n (2S7485)

30-Marzo-10

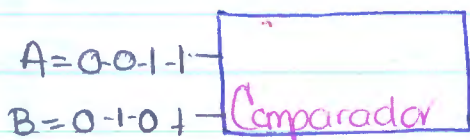
### Ejemplo

Diseñe un Sistema Combinacional capaz de comparar dos números binarios de un bit cada número.

Los números binarios los llamaremos A y B respectivamente y representan la entrada del sistema combi-nacional.

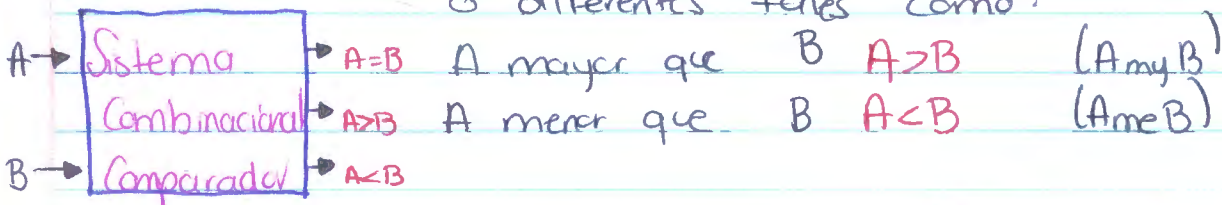


Al comparar dos números los posibles resultados pueden ser:



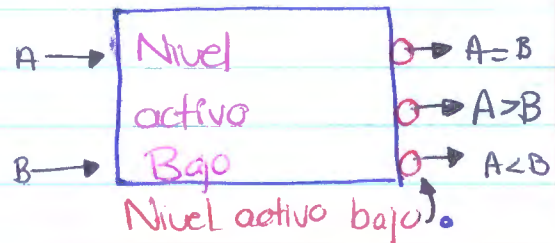
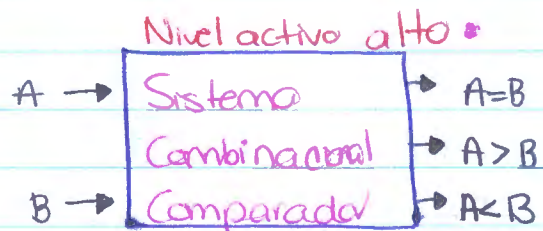
Al comparar dos números los posibles resultados pueden ser que sean: iguales  $A=B$  ( $A \text{ eq } B$ )

o diferentes tales como:



### Tabla de Verdad

m	A	B	A=B	A>B	A<B
0	0	0	1	0	0
1	0	1	0	0	1
2	1	0	0	1	0
3	1	1	1	0	0



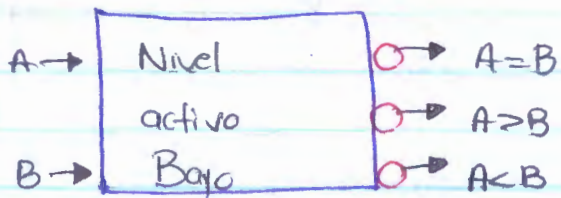
30-Marzo-10

m	A	B	A=B	A>B	A<B
0	0	0	1	0	0
1	0	1	0	0	1
2	1	0	0	1	0
3	1	1	1	0	0

Nivel activo alto

m	A	B	A=B	A>B	A<B
0	0	0	0	1	1
1	0	1	1	1	0
2	1	0	1	0	1
3	1	1	0	1	1

Nivel activo bajo



m	A	B	A=B	A>B	A<B
0	0	0	1	0	0
1	0	1	0	0	1
2	1	0	0	1	0
3	1	1	1	0	0

### Ecuaciones minimas

$$F_{A=B}(A,B) = \overline{A \oplus B}$$



$$F_{A>B}(A,B) = A\bar{B}$$



$$F_{A<B}(A,B) = \bar{A}B$$



30-Marzo-10

## Abel-Hdl Ecuaciones

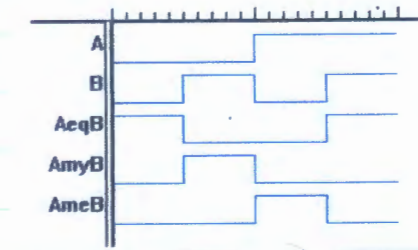
```
MODULE comp
"Entradas
AB pin 1,2;
"Salidas
AeqB, AmyB, AmeB pin 19..17 istype 'com';
equations
AeqB =!(A&B);
AmyB =A&B;
AmeB =!A&B;
test_vectors
([A,B]->[AeqB, AmyB, AmeB])
[0,0]->[1,0,0];
[0,1]->[0,0,1];
[1,0]->[0,1,0];
[1,1]->[1,0,0];
END
```

$$F_{A=B}(A,B) = \overline{A \oplus B}$$

$$F_{A>B}(A,B) = A\bar{B}$$

$$F_{A<B}(A,B) = \bar{A}B$$

## Abel-Hdl Simulación



## Abel-Hdl Tabla de Verdad

```
MODULE comp
"Entradas
AB pin 1,2;
"Salidas
AeqB, AmyB, AmeB pin 19..17 istype 'com';
Truth table
([A,B]->[AeqB, AmyB, AmeB])
[0,0]->[1,0,0];
[0,1]->[0,0,1];
[1,0]->[0,1,0];
[1,1]->[1,0,0];
test_vectors
([A,B]->[AeqB, AmyB, AmeB])
[0,0]->[1,0,0];
[0,1]->[0,0,1];
[1,0]->[0,1,0];
[1,1]->[1,0,0];
END
```

m	AB	A=B	A>B	A<B
0	00	1	0	0
1	01	0	0	1
2	10	0	1	0
3	11	1	0	0

30-Marzo-10

## Abel-Hdl descripción

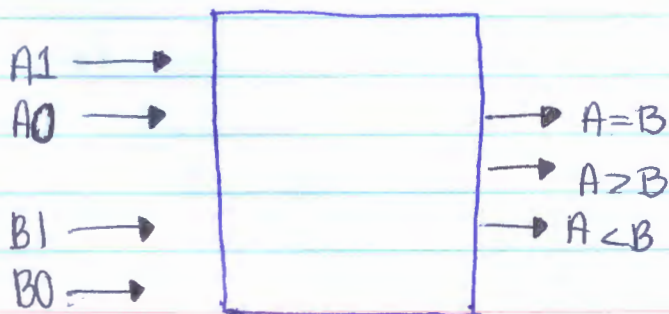
Operador	Descripción	
$=$	Igual	Operaciones Relacionales
$\neq$	Diferente	
$<$	Menor que	
$>$	Mayor que	
$\leq$	Menor igual que	
$\geq$	Mayor igual que	

## Abel-Hdl descripción

```
MODULE comp
  "Entradas
  A,B pin 1,2;
  "Salidas
  AeqB,AmyB,AmeB pin 19..17 istype 'com';
  equations
  when A==B then AeqB=1;
  when A>B then AmyB=1;
  when A<B then AmeB=1;
  test_vectors
  ([A,B]->[AeqB,AmyB,AmeB])
  [0,0]->[1,0,0];
  [0,1]->[0,0,1];
  [1,0]->[0,1,0];
  [1,1]->[1,0,0];
  END
```

## Ejemplo

Disene un Sistema Combinacional capaz de comparar dos numeros binarios de dos bit's cada numero



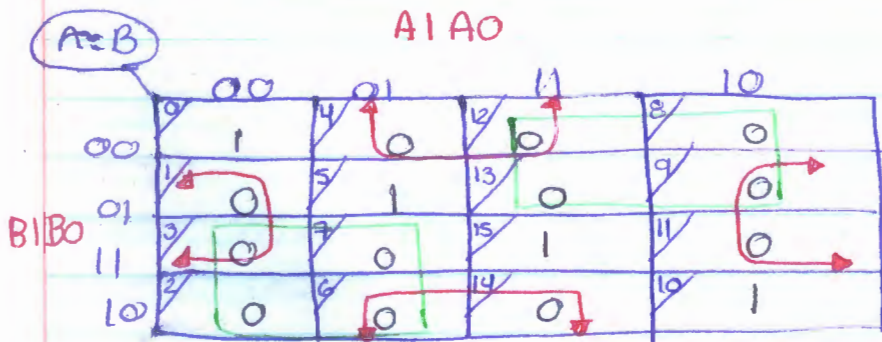
## Tabla de verdad

m	A	B	$A=B$	$A>B$	$A<B$
0	0	0	1	0	0
1	0	1	0	0	1
2	0	2	0	0	1
3	0	3	0	0	1
4	1	0	0	1	0
5	1	1	1	0	0
6	1	2	0	0	1
7	1	3	0	0	1
8	2	0	0	1	0
9	2	1	0	1	0
10	2	2	1	0	0
11	2	3	0	0	1
12	3	0	0	1	0
13	3	1	0	1	0
14	3	2	0	1	0
15	3	3	1	0	0

30-Marco-10

Tabla de Verdad

m	A1	A0	B1	B0	A=B	A>B	A<B
0	0	0	0	0	1	0	0
1	0	0	0	1	0	0	1
2	0	0	1	0	0	0	1
3	0	0	1	1	0	0	1
4	0	1	0	0	0	1	0
5	0	1	0	1	1	0	0
6	0	1	1	0	0	0	1
7	0	1	1	1	0	0	1
8	1	0	0	0	0	1	0
9	1	0	0	1	0	1	0
10	1	0	1	0	1	0	0
11	1	0	1	1	0	0	1
12	1	1	0	0	0	1	0
13	1	1	0	1	0	1	0
14	1	1	1	0	0	1	0
15	1	1	1	1	1	0	0



$$F_{A=B}(A_1, A_0, B_1, B_0) = (A_1' + B_1)(A_1 + B_1')(A_0 + B_0')(A_0' + B_0)$$

30-Marzo-10

**A > B**

A1 A0

	00	01	11	10
00	0	1	1	1
01	0	0	1	1
11	0	0	0	0
10	0	0	1	0

B1 B0

$$F_{A>B}(A_1, A_0, B_1, B_0) = A_1 B_1' + A_0 B_1' B_0' + A_1 A_0 B_0'$$

**A < B**

A1 A0

	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0

B1 B0

$$F_{A<B}(A_1, A_0, B_1, B_0) = A_1' B_1 + A_0' B_1 B_0 + A_1' A_0' B_0$$

**MODULE compare**

"Entradas

A1, A0, B1, B0 pin 1..4;

"Salidas

AeqB, AmyB, AmeB Pin 19..17 istype 'com';

**A=[A1, A0];**

**B=[B1, B0];**

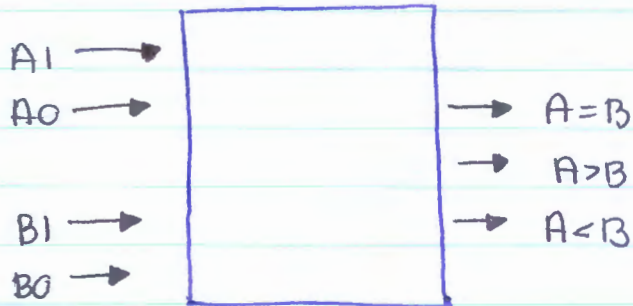
equations

**When A == B then AeqB=1;**

**When A > B then AmyB=1;**

**When A < B then AmeB=1;**

**End**





### Ejemplo

Diseñe un Sistema Combinacional capaz de comparar dos numeros binarios de cuatro bit's cada numero



MODULE compara

"Entradas

A3..A0, B3..B0 pin 1..8;

"Salidas

AeqB, AmyB, AmeB Pin 19..17 istype 'com';

**A=[A3,A2,A1,A0];**

**B=[B3,B2,B1,B0];**

equations

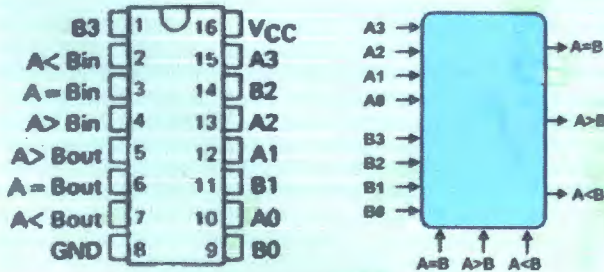
**When A == B then AeqB=1;**

**When A > B then AmyB=1;**

**When A < B then AmeB=1;**

**End**

SN7485 4-bit binary or BCD  
magnitude comparators.



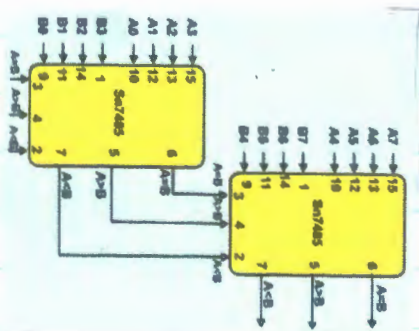
SN7485 -2

30-Marzo-10

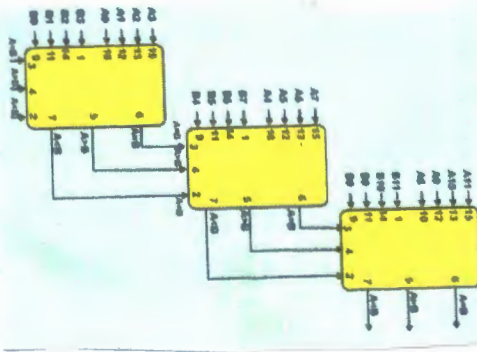
## Tabla de funcionamiento

Entradas				Cascada			Salidas		
A3,B3	A2,B2	A1,B1	A0,B0	A>B	A<B	A=B	A>B	A<B	A=B
A3>B3	X	X	X	X	X	X	H	L	L
A3<B3	X	X	X	X	X	X	L	H	L
A3=B3	A2>B2	X	X	X	X	X	H	L	L
A3=B3	A2<B2	X	X	X	X	X	L	H	L
A3=B3	A2=B2	A1>B1	X	X	X	X	H	L	L
A3=B3	A2=B2	A1<B1	X	X	X	X	L	H	L
A3=B3	A2=B2	A1=B1	A0>B0	X	X	X	H	L	L
A3=B3	A2=B2	A1=B1	A0<B0	X	X	X	L	H	L
A3=B3	A2=B2	A1=B1	A0=B0	X	X	X	L	L	H

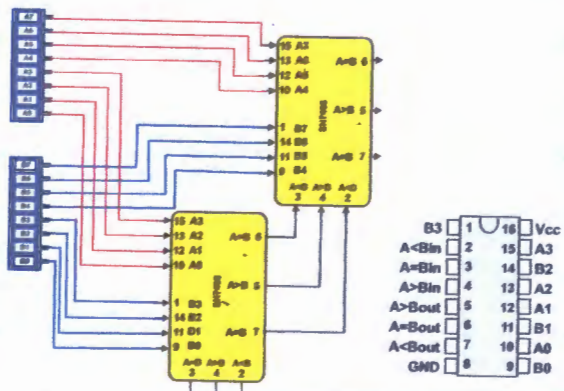
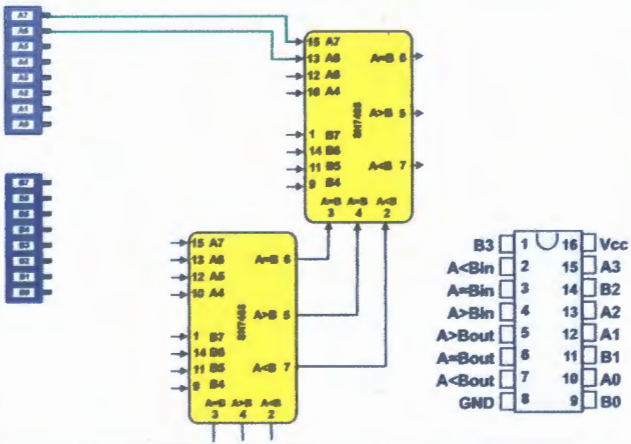
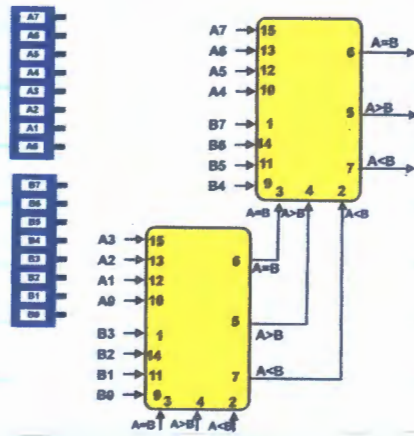
Comparación de 2 números de 8 bit c/n usando SN7485 en cascada



Comparación de 2 números de 12 bit c/n usando SN7485 en cascada

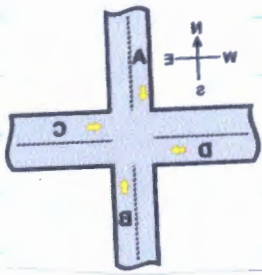


30-Marco-10



Recommended operating conditions	MIN	NOM	MAX	UNIT
Supply voltage, $V_{cc}$	4.75	5	5.25	V
High level output current, $I_{OH}$			-400	$\mu A$
Low level output current, $I_{OL}$			16	mA
Operating free-air temperature, $T_A$	0		70	$^{\circ}C$

30 - Marzo - 10



Sistemas digitales  
Principios y aplicaciones  
Autores: Ronald J. Tocci

La Figura muestra la intersección de una Autopista principal con un camino de Acceso secundario.

Se colocan detectores de vehículos a lo largo de los carriles C y D (camino principal) y en los carriles A y B (camino de acceso).

Las lecturas o salidas de los detectores son bajas "0" cuando no pasa ningún vehículo y altas "1" cuando pasa algún vehículo

a) El semáforo E-W estará en luz verde siempre que los carriles C y D estén ocupados.

b) El semáforo E-W estará en luz verde siempre que C o D estén ocupados y siempre y cuando A y B no estén ocupados

c) El semáforo N-S estará en luz verde siempre que los carriles A y B estén ocupados y siempre y cuando C y D no estén ocupados.

d) El semáforo N-S también estará en luz verde siempre que los carriles A o B estén ocupados en tanto C y D estén vacantes

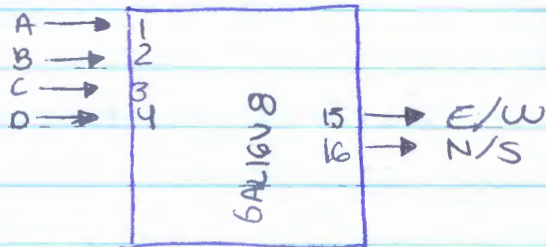
e) El semáforo E-W estará en luz verde cuando haya vehículos transitando.

Utilizando las salidas de los sensores A B C y D como entradas de un circuito lógico para controlar el semáforo, debe de tener dos salidas, N/S y E/W que sean alto "1"

30-Marzo-10

cuando corresponda la luz verde, y bajo "0" donde corresponda a la luz roja.

Indique a que combinación de la tabla de verdad corresponde cada uno de los incisos arriba mencionados



m	A	B	C	D	E/W	N/S	Inciso
0	0	0	0	0	1	0	e
1	0	0	0	1	1	0	b
2	0	0	1	0	1	0	b
3	0	0	1	1	1	0	a
4	0	1	0	0	0	1	d
5	0	1	0	1	1	0	b
6	0	1	1	0	1	0	b
7	0	1	1	1	1	0	a
8	1	0	0	0	0	1	d
9	1	0	0	1	1	0	b
10	1	0	1	0	1	0	b
11	1	0	1	1	1	0	a
12	1	1	0	0	0	1	c
13	1	1	0	1	0	1	c
14	1	1	1	0	0	1	c
15	1	1	1	1	1	0	a

30-Marzo-10

### e) Ecuaciones minimas

EW

	00	01	11	10
00	0	0	0	0
01	1	1	0	1
11	1	1	1	1
10	1	1	0	1

$$EW = \underline{CD} + \underline{B'C} + \underline{B'D} + \underline{A'C} + \underline{A'D} + \underline{A'B'}$$

$$EW = (A'+B'+D)(A'+B'+C)(A'+C+D)(B'+C+D)$$

	00	01	11	10
00	0	0	0	0
01	1	1	0	1
11	1	1	1	1
10	1	1	0	1

### Resumen

1. Comparación numerica
2. Nivel activo (salida)
3. Operadores relacionales
4. Conveniencia de programas Vs Funcion fija
5. Metodo de comparacion
6. Entradas de cascada
7. Pasar de redaccion a tabla de verdad

dc5

# Sistemas Combinacionales que no 30-Marzo-10

Están completamente especificados

Un sistema combinatorial se puede declarar que no está completamente especificado por dos razones.

**Can't Happen** Una o varias combinaciones de entrada que debido a las características del sistema no se puede presentar.  
No puede suceder

**Don't care** Un valor de salida o una combinación de entrada que no importa el valor que se le asigne, el sistema no es afectado.  
No importa

Can't happen No puede suceder	Don't care No importa
----------------------------------	--------------------------

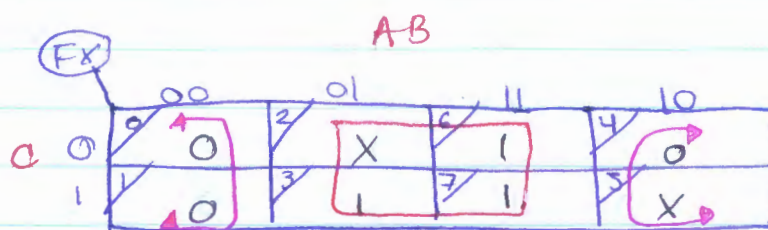
En ambos casos se aprovecha que la entrada no se presente o que el valor de salida no importe, por lo que se le asigne un valor de  $x$  a la salida en la tabla de verdad.

En donde ese valor de  $x$  individualmente se toma como cero o uno según convenga a una mejor minimización

30-Marzo-10

$$F_x(A, B, C) = \sum m(3, 6, 7), d(2, 5)$$

m	A	B	C	F <sub>x</sub>
0	0	0	0	0
1	0	0	1	0
2	0	1	0	X
3	0	1	1	1
4	1	0	0	0
5	1	0	1	X
6	1	1	0	1
7	1	1	1	1



$$F_x(A, B, C, D) = B$$

$$F_x(A, B, C, D) = B$$

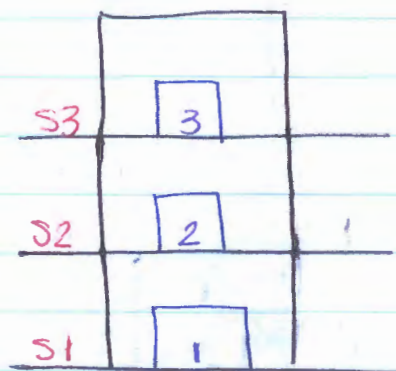
Control de la puerta de un elevador de 3 pisos  
Sensores

M Motor del elevador

S1 Sensor del piso 1

S2 Sensor del piso 2

S3 Sensor del piso 3



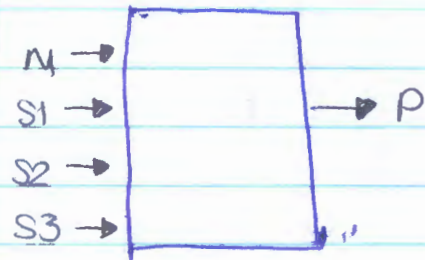
Solo se puede abrir la puerta cuando el motor esta parado  $M=0$  y el elevador este en cualquiera de los pisos

$$S1=1 \text{ o } S2=1 \text{ o } S3=1$$



30-Marzo-10

m	M	s1	s2	s3	P
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	X
4	0	1	0	0	1
5	0	1	0	1	X
6	0	1	1	0	X
7	0	1	1	1	X
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	X
12	1	1	0	0	0
13	1	1	0	1	X
14	1	1	1	0	X
15	1	1	1	1	X



MSI

MSI

	00	01	11	10
00	0	1	0	0
01	1	X	X	0
11	X	X	X	X
10	1	X	X	0

Red circles highlight the 1s in the 00, 01, 10, and 11 rows of the 00 column. A red box highlights the 1s in the 00, 01, 10, and 11 rows of the 01 column.

$$P(M, s1, s2, s3) = M's1 + M's3 + M's2$$

$$P(M, s1, s2, s3) = M'(s1 + s2 + s3)$$

30-Marzo-10

### Archivo en formato-Abel - HDL

F PIN istype 'dc, com';

Para obtener ventaja de las combinaciones que no se presentan (can't happen) o las salidas que no importa el valor (Don't care) es necesario incluir el comando DC (Don't Care) en la línea de las declaraciones de salidas UP, CP, DP, M PIN19...16 istype 'dc, com';

Si alguna combinación de la tabla de verdad no es incluida esta se tomara como X.

En caso de que se listaran es necesario sustituir el valor de salida por .X.

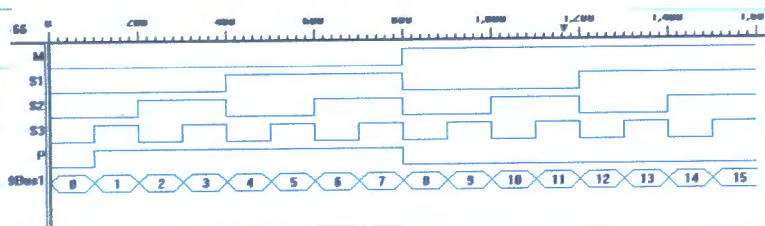
Con dc si alguna combinación de la tabla de verdad no es incluida se tomara como X.

*Allegre*

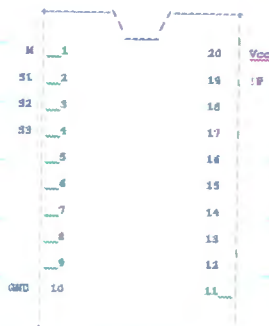
```

MODULE poly
  Entradas
  M,S1,S2,S3 pin 1..4;
  Salida
  P pin 19 istype 'dc,com';
  F={M,S1,S2,S3};
  truth table
  (F>P)
  0->0;
  1->1;
  2->1;
  3->.X.;
  4->1;
  5->.X.;
  6->.X.;
  7->.X.;
  8->0;
  9->0;
  10->0;
  11->.X.;
  12->0;
  13->.X.;
  14->.X.;
  15->.X.;
END

```



$$P = 1 (S3 \& !S2 \& !S1 \# M);$$



30-Marzo-10

## Detector de monedas

Se desea detectar que tipos de monedas se insertan en una maquina expendedora. Las monedas que se aceptan son:  
\$ 1 (UP)      \$ 10 (DP)  
\$ 5 (CP)

Se colocan 3 fotoceldas a distancia conveniente de modo que:

La moneda de \$ 1 solo tapara la fotocelda C.



La moneda de \$ 5 tapara las fotoceldas B y C.



La moneda de \$ 10 tapara las tres fotoceldas A, B y C.



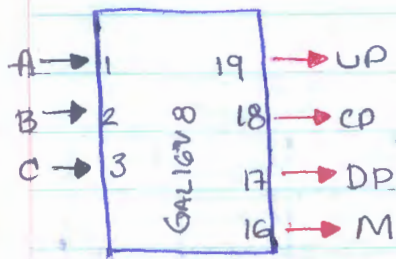
El sistema consta de tres entradas A, B y C en donde toman el valor de uno cuando hay moneda presente y de cero cuando no hay moneda.

Se requieren de tres salidas (UP, CP y DP) de modo que cuando la moneda es la indicada la salida tomará un valor de uno.

Es conveniente incluir una cuarta salida llamada mantenimiento (M) que tome el valor de uno cuando ocurra una combinación de entrada no prevista.

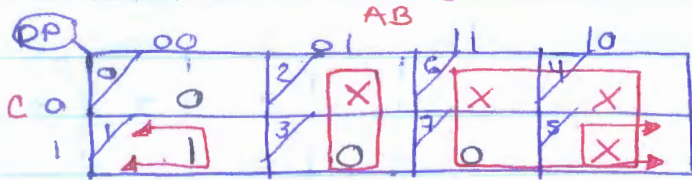
30-Marzo-10

Tabla de Verdad

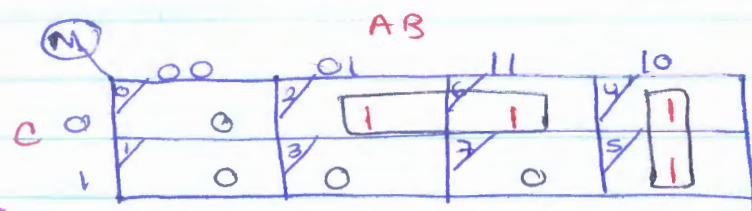


m	A	B	C	UP	CP	DP	M
0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0
2	0	1	0	X	X	X	1
3	0	1	1	0	1	0	0
4	1	0	0	X	X	X	1
5	1	0	1	X	X	X	1
6	1	1	0	X	X	X	1
7	1	1	1	0	0	1	0

Ecuaciones minimas

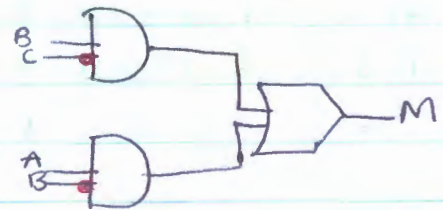
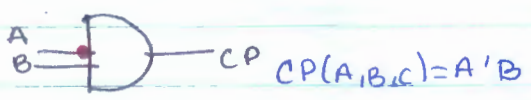
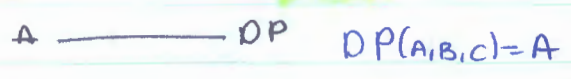
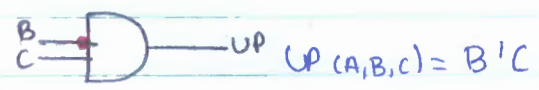


$UP(A,B,C) = B'C$   
 $CP(A,B,C) = A'B$   
 $DP(A,B,C) = A$



$M(A,B,C) = BC' + AB'$

Diagrama esquematico

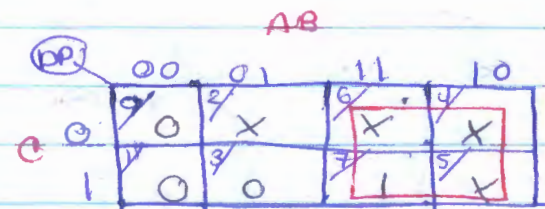
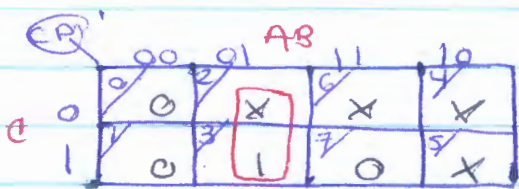
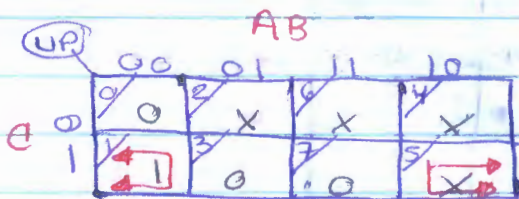


$M(A,B,C) = BC' + AB'$

30-Marzo-10

Cuales seran los valores de salida si se presentara la combinacion S.

m	A	B	C	UP	CP	DP	M
0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0
2	0	1	0	X	X	X	1
3	0	1	1	0	1	0	0
4	1	0	0	X	X	X	1
5	1	0	1	X	X	X	1
6	1	1	0	X	X	X	1
7	1	1	1	0	0	1	0



Respuesta

Los valores de salida seran los que se le asignaron a las X en el mapa

m	A	B	C	UP	CP	DP	M
0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0
2	0	1	0	0	1	0	1
3	0	1	1	0	1	0	0
4	1	0	0	0	0	1	1
5	1	0	1	1	0	1	1
6	1	1	0	0	0	1	1
7	1	1	1	0	0	1	0

# Archivo en formato ABEL-HDL

```

MODULE monedas
  *Simplificación de variables
  x = .x.;
  Entradas
  A,B,C pin 1,2,3;
  *Salidas
  UP,CP,DP,M pin 19..16 istype 'dc,com';
  *SET
  E=[A,B,C];
  truth_table
  (E->[UP,CP,DP,M])
  0->[0,0,0,0];
  2->[x,x,x,1];
  1->[1,0,0,0];
  3->[0,1,0,0];
  4->[x,x,x,1];
  5->[x,x,x,1];
  6->[x,x,x,1];
  7->[0,0,1,0];
  
```

## ARCHIVO EN FORMATO ABEL-HDL

```

TEST_VECTORS
(E->[UP,CP,DP])
0->[x,x,x];
1->[x,x,x];
2->[x,x,x];
3->[x,x,x];
4->[x,x,x];
5->[x,x,x];
6->[x,x,x];
7->[x,x,x];
END
  
```

	A	B	C	UP	CP	DP	M
0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0
2	0	1	0	X	X	X	1
3	0	1	1	0	1	0	0
4	1	0	0	X	X	X	1
5	1	0	1	X	X	X	1
6	1	1	0	X	X	X	1
7	1	1	1	0	0	1	0

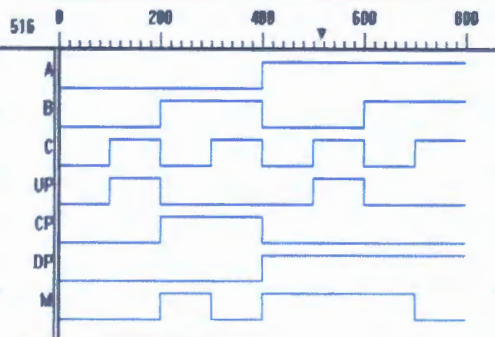
$$UP(A,B,C) = B'C$$

$$CP(A,B,C) = A'B$$

$$DP(A,B,C) = A$$

$$\begin{aligned}
 UP &= (C \& !B); \\
 CP &= (B \& !A); \\
 DP &= !(A); \\
 M &= !(C \& B \& !B \& !A);
 \end{aligned}$$

$$M(A,B,C) = BC' + AB'$$



# Actividad 4 Miércoles 21

30-Marzo-10

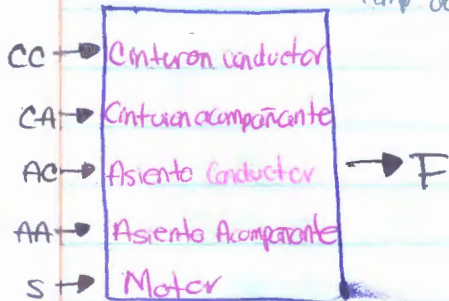
## Ejemplo de cinturón de Seguridad

Se desea diseñar un circuito que avise cuando alguna de las personas de los asientos delanteros **NO** se ha puesto el cinturón (encendiendo un LED, F), siempre que haya alguien en el asiento y el coche este en marcha. Para ello se dispone de 5 sensores:

- Dos en el sistema de enganche de los cinturones, uno para el conductor (CC) y otro para el acompañante (CA). Su salida es un 1 si NO tenemos el cinturón puesto y un 0 en caso contrario.
- Dos sensores mas que no avisan si hay alguien sentado en el Asiento del conductor (AC) o en el del acompañante (AA). Un 1 indica la presencia de alguien en el asiento y un 0 la ausencia.
- Además hay otra señal de control que nos indica cuando el coche está en marcha ( $s=1$ ) y cuando está parado ( $s=0$ )

Sumador de 2 Bin <sup>8bits</sup> Sumador 2sn74283

Comp de 2 Bin <sup>8bits</sup> Cam 2sn7485



30-Marzo-10

25

when  $S=0$   $AL=0$

	S	CC	CA	AL
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	0	1	0	0
9	0	1	0	1
10	0	1	1	0
11	0	1	1	1
12	0	1	0	0
13	0	1	0	1
14	0	1	1	0
15	0	1	1	1
16	1	0	0	0
17	1	0	0	1
18	1	0	1	0
19	1	0	1	1
20	1	1	0	0
21	1	1	0	1
22	1	1	1	0
23	1	1	1	1
24	1	0	0	0
25	1	0	0	1
26	1	0	1	0
27	1	0	1	1
28	1	1	0	0
29	1	1	0	1
30	1	1	1	0
31	1	1	1	1

	S	CC	CA	<del>C</del>	<del>A</del>	AL
0	0	X	X	X	X	0
1	1	0	0	0	0	0
2	1	0	0	0	1	0
3	1	0	0	1	0	0
4	1	0	0	1	1	0
5	1	0	1	0	0	0
6	1	0	1	0	1	1
7	1	0	1	1	0	0
8	1	0	1	1	1	1
9	1	1	0	0	0	0
10	1	1	0	0	1	0
11	1	1	0	1	0	1
12	1	1	0	1	1	1
13	1	1	1	0	0	0
14	1	1	1	0	1	1
15	1	1	1	1	0	0
16	1	1	1	1	1	1

dc6

31/10/10



30-Marzo-10

## Codificador de Prioridad (Priority Encoder)

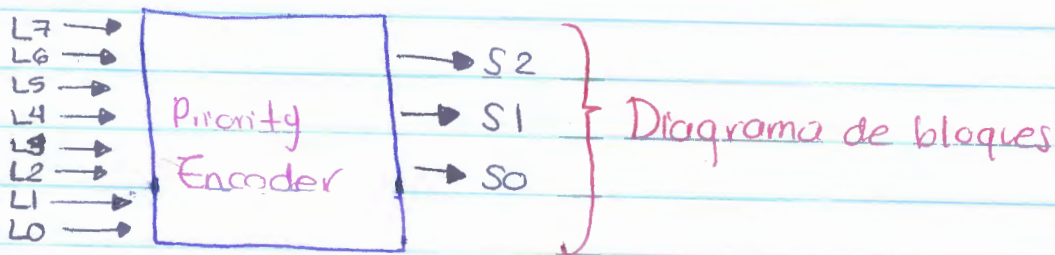
### Definiciones

**Codificador** Proceso en el cual se obtiene un código para cada una de las posibles entradas.

**Código** Combinación de signos que tiene un determinado valor dentro de un sistema establecido.

**Prioridad** Anterioridad de algo respecto de otra cosa, en tiempo, en orden.

Diseña un codificador de prioridad de 8 líneas de entrada a 3 bit's de salida.



Las salidas (S2, S1, S0) debe ser el número en código binario correspondiente a la línea de entrada de mayor valor que contenga un uno.

### Tabla de Verdad

Con ocho entradas se tienen 256 posibles combinaciones. En este ejemplo no es necesario listarlas todas ya que gracias al Don't care (.x.).

En las combinaciones de entrada, se puede representar el comportamiento del codificador de prioridad con solo 9 combinaciones como lo muestra la siguiente tabla.

# Tabla de Verdad

Entradas								Salidas		
L7	L6	L5	L4	L3	L2	L1	L0	S2	S1	S0
1	X	X	X	X	X	X	X	1	1	1
0	1	X	X	X	X	X	X	1	1	0
0	0	1	X	X	X	X	X	1	0	1
0	0	0	1	X	X	X	X	1	0	0
0	0	0	0	1	X	X	X	0	1	1
0	0	0	0	0	1	X	X	0	1	0
0	0	0	0	0	0	1	X	0	0	1
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0

## Archivo en ABEL-HDL

```

MODULE pe
  "Entradas
  x=.x.;
  L7..L0 pin 1..8;
  "Salidas
  S2..S0 pin 19..17 istype 'dc.com';

  Truth_table
  ([L7..L6..L5..L4..L3..L2..L1..L0])->[S2..S1..S0]
  [1.x.x.x.x.x.x.x]->[1.1.1];
  [0.1.x.x.x.x.x.x]->[1.1.0];
  [0.0.1.x.x.x.x.x]->[1.0.1];
  [0.0.0.1.x.x.x.x]->[1.0.0];
  [0.0.0.0.1.x.x.x]->[0.1.1];
  [0.0.0.0.0.1.x.x]->[0.1.0];
  [0.0.0.0.0.0.1.x]->[0.0.1];
  [0.0.0.0.0.0.0.x]->[0.0.0];

  test_vectors
  ([L7..L6..L5..L4..L3..L2..L1..L0])->[S2..S1..S0]
  [1.x.x.x.x.x.x.x]->[1.1.1];
  [0.1.x.x.x.x.x.x]->[1.1.0];
  [0.0.1.x.x.x.x.x]->[1.0.1];
  [0.0.0.1.x.x.x.x]->[1.0.0];
  [0.0.0.0.1.x.x.x]->[0.1.1];
  [0.0.0.0.0.1.x.x]->[0.1.0];
  [0.0.0.0.0.0.1.x]->[0.0.1];
  [0.0.0.0.0.0.0.x]->[0.0.0];
END
  
```

```

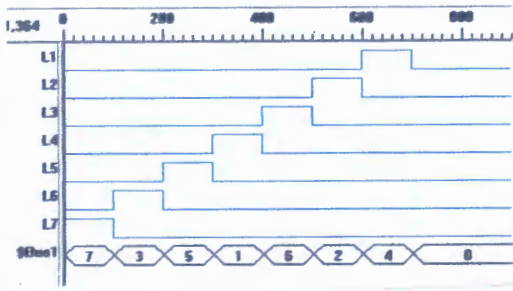
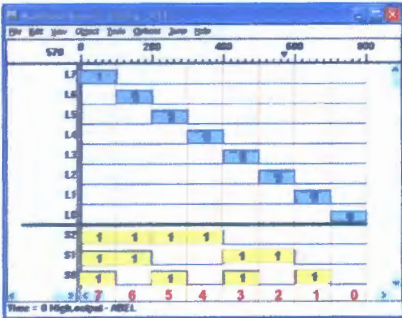
MODULE pe
  "Entradas
  x=.x.;
  L7..L0 pin 1..8;
  "Salidas
  S2..S0 pin 19..17 istype 'dc.com';
  S=[S2..S1..S0];

  equations
  when L7 then S=7;
  when !L7&L6 then S=6;
  when !L7&!L6&L5 then S=5;
  when !L7&!L6&!L5&L4 then S=4;
  when !L7&!L6&!L5&!L4&L3 then S=3;
  when !L7&!L6&!L5&!L4&!L3&L2 then S=2;
  when !L7&!L6&!L5&!L4&!L3&!L2&L1 then S=1 else S=0;

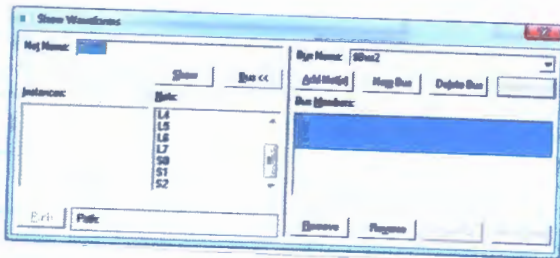
  test_vectors
  ([L7..L6..L5..L4..L3..L2..L1..L0])->[S2..S1..S0]
  [1.x.x.x.x.x.x.x]->[1.1.1];
  [0.1.x.x.x.x.x.x]->[1.1.0];
  [0.0.1.x.x.x.x.x]->[1.0.1];
  [0.0.0.1.x.x.x.x]->[1.0.0];
  [0.0.0.0.1.x.x.x]->[0.1.1];
  [0.0.0.0.0.1.x.x]->[0.1.0];
  [0.0.0.0.0.0.1.x]->[0.0.1];
  [0.0.0.0.0.0.0.x]->[0.0.0];
END
  
```

30-Marzo-10

# Simulación



\$ Bus1 = S2, S1, S0

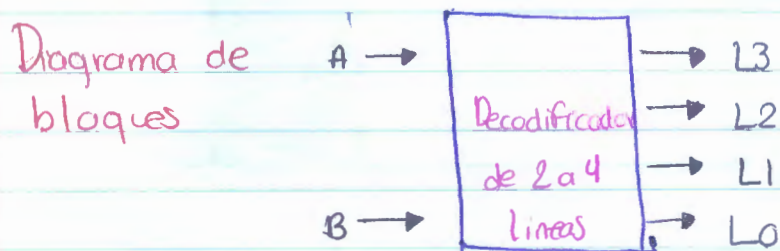


30-Marzo-10

# Decodificadores

## Decodificador de 2 a 4 líneas

Decodificador Proceso que permite pasar de un lenguaje codificado (codigo) a otro legible directamente (línea)



El valor binario de A, B de entrada deberá seleccionar la línea de salida por medio de un nivel alto (uno)

## Tabla de Verdad

m	Entradas		Salidas			
	A	B	L3	L2	L1	L0
0	0	0	0	0	0	1
1	0	1	0	0	1	0
2	1	0	0	1	0	0
3	1	1	1	0	0	0

## Ecuaciones minimas

$$L0(A,B) = A'B'$$

$$L1(A,B) = A'B$$

$$L2(A,B) = AB'$$

$$L3(A,B) = AB$$

30-Marzo-10

## Decodificador de 3 a 8 Lineas

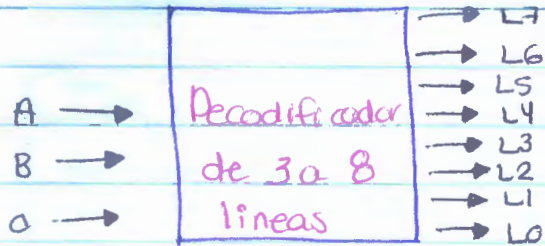


Diagrama de Bloques

## Tabla de Verdad

D	A	B	C	L7	L6	L5	L4	L3	L2	L1	L0
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
2	0	1	0	0	0	0	0	0	1	0	0
3	0	1	1	0	0	0	0	1	0	0	0
4	1	0	0	0	0	0	1	0	0	0	0
5	1	0	1	0	0	1	0	0	0	0	0
6	1	1	0	0	1	0	0	0	0	0	0
7	1	1	1	1	0	0	0	0	0	0	0

## Decodificador de 3 a 8 Lineas

Nivel activo Bajo

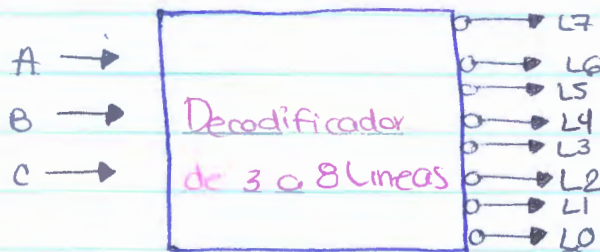


Diagrama de Bloques

30-Marzo-10

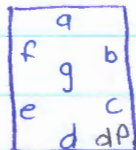
## Tabla de Verdad

D	A	B	C	L7	L6	L5	L4	L3	L2	L1	L0
0	0	0	0	1	1	1	1	1	1	1	0
1	0	0	1	1	1	1	1	1	1	0	1
2	0	1	0	1	1	1	1	1	0	1	1
3	0	1	1	1	1	1	1	0	1	1	1
4	1	0	0	1	1	1	0	1	1	1	1
5	1	0	1	1	1	0	1	1	1	1	1
6	1	1	0	1	0	1	1	1	1	1	1
7	1	1	1	0	1	1	1	1	1	1	1

## Decodificador de BCD a siete Segmentos

BCD Código Decimal (0-9) expresado en Binario (0000 - 1001), cada dígito del decimal se representa por cuatro bits ejemplo 4678(10) 0100 0110 0111 1000 (BCD)

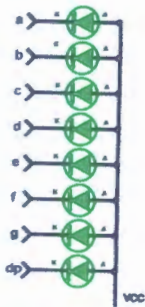
7 Segmentos Se refiere a un DISPLAY (dispositivo para mostrar resultados) compuesto por LED'S (Diodos emisores de luz) distribuidas de tal suerte que se pueden mostrar los dígitos del 0 al 9.



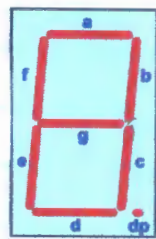
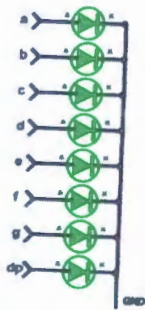
30-Marzo-10

# Display de 7 Segmentos

Anodo Común



Catodo Común

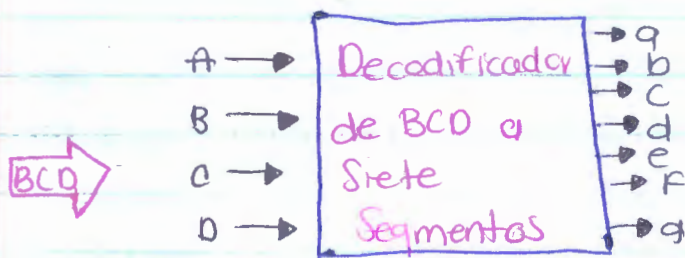


Enciende con un cero

Enciende con un uno

## Decodificador de BCD a Siete Segmentos

Diagrama de bloques



IN	ABCD	a	b	c	d	e	f	g
0	0000	1	1	1	1	1	1	0
1	0001	0	1	1	0	0	0	0
2	0010	1	1	0	1	1	0	1
3	0011	1	1	1	1	0	0	1
4	0100	0	1	1	0	0	1	1
5	0101	1	0	1	1	0	1	1
6	0110	X	0	1	1	1	1	1
7	0111	1	1	1	0	0	0	0
8	1000	1	1	1	1	1	1	1
9	1001	1	1	1	X	0	1	1
10	1010							
11	1011							
12	1100							
13	1101							
14	1110							
15	1111							

30-Marzo-10

# Tabla de Verdad

m	ABCD	a	b	c	d	e	f	g
0	0000	1	1	1	1	1	1	0
1	0001	0	1	1	0	0	0	0
2	0010	1	1	0	1	1	0	1
3	0011	1	1	1	1	0	0	1
4	0100	0	1	1	0	0	1	1
5	0101	1	0	1	1	0	1	1
6	0110	X	0	1	1	1	1	1
7	0111	1	1	1	0	0	0	0
8	1000	1	1	1	1	1	1	1
9	1001	1	1	1	X	0	1	1
10	1010	X	X	X	X	X	X	X
11	1011	X	X	X	X	X	X	X
12	1100	X	X	X	X	X	X	X
13	1101	X	X	X	X	X	X	X
14	1110	X	X	X	X	X	X	X
15	1111	X	X	X	X	X	X	X

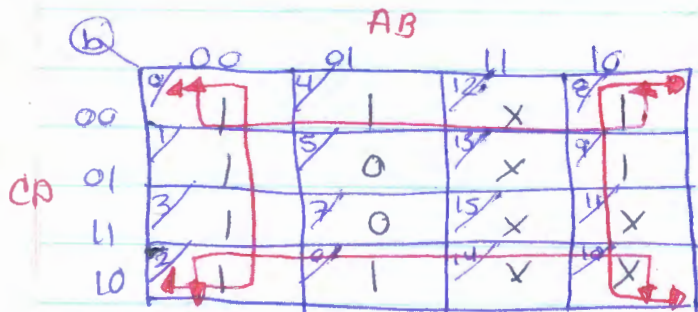
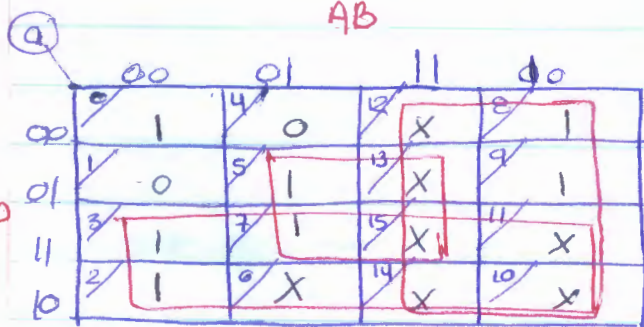
Considerando  
Catodo Común

Código BCD  
0 a 9

[ 0 L L A A

## Ecuaciones minimas

a b c d e f g  
 C- 1 0 0 1 1 1 0  
 O- 1 1 1 1 1 1 0  
 L- 0 0 0 1 1 1 0  
 L- 0 0 0 1 1 1 0  
 A- 1 1 1 0 1 1 1  
 R- 1 1 1 0 1 1 1





30-Marzo-10

## Archivo ABEL-HDL

MODULE BCD7

\* Entradas

A,B,C,D PIN 1,2,3,4:

\* Salidas

a,b,c,d,e,f,g pin 16,17,18,12,13,15,14 istype

com.dc:

S= [A,B,C,D]:

\* Tabla de verdad para Cátodo Común

truth\_table

([S]->[a,b,c,d,e,f,g])

[0]-> [1.1.1.1.1.1.0]:

[1]-> [0.0.1.0.0.0.0]:

[2]-> [1.1.0.1.1.0.1]:

[3]-> [1.1.1.1.0.0.1]:

[4]-> [0.1.1.0.0.1.1]:

[5]-> [1.0.1.1.0.1.1]:

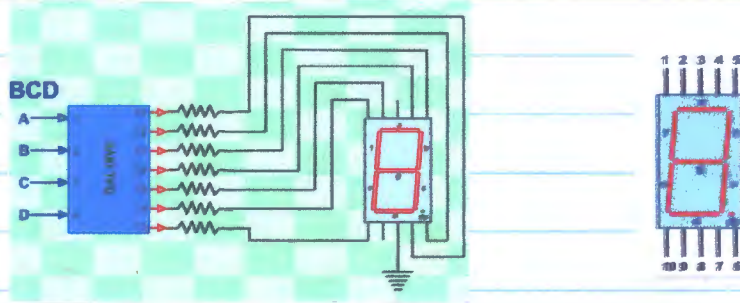
[6]-> [x.0.1.1.1.1.1]:

[7]-> [1.1.1.0.0.0.0]:

[8]-> [1.1.1.1.1.1.1]:

[9]-> [1.1.1..x.0.1.1]:

END

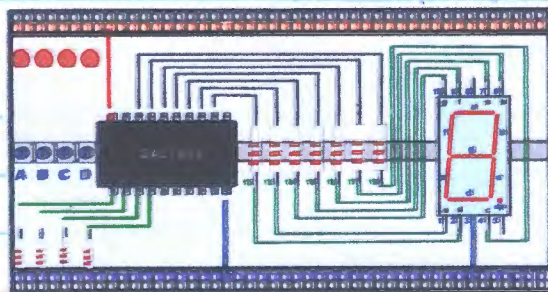


## Proyecto adicional 5

Decodificador de 7

Este del 0-9

y del 10-15 que diga en mensaje



30-Marzo-10

# Convertidores en código

## Códigos Binarios

## Códigos alfanuméricos

Código ASCII

Código EBCDIC

## Códigos numéricos

Decimales BCD, Exceso 3, 2421, 5211

Binarios N(2), Gray

Especiales JOHNSON

## Código ASCII (0-127)

American Standard Code for Information Interchange

## Código ASCII extendido (128-255)

American standard Code for Information Interchange

128	Ç	144	É	160	á	176	⌘	193	±	209	ƒ	225	ß	241	±
129	à	145	è	161	â	177	⌘	194	ƒ	210	ƒ	226	Γ	242	≥
130	á	146	é	162	ó	178	■	195	†	211	⌘	227	κ	243	≤
131	ê	147	ê	163	û	179		196	-	212	⌘	228	Σ	244	∫
132	ë	148	ö	164	ä	180	†	197	†	213	ƒ	229	o	245	∫
133	ì	149	ö	165	Ë	181	†	198	†	214	ƒ	230	μ	246	+
134	í	150	ü	166	°	182	†	199	†	215	†	231	τ	247	∞
135	î	151	à	167	°	183	†	200	⌘	216	†	232	ϕ	248	°
136	ï	152	—	168	ó	184	†	201	ƒ	217	∫	233	⊙	249	.
137	ï	153	Ö	169	-	185	†	202	⌘	218	ƒ	234	□	250	.
138	ï	154	Ü	170	→	186		203	ƒ	219	■	235	ö	251	∫
139	∴	156	É	171	½	187	†	204	†	220	■	236	∞	252	-
140	∴	157	⌘	172	¾	188	†	205	∞	221		237	ϕ	253	∞
141	∴	158	-	173		189	†	206	†	222	■	238	e	254	■
142	Ä	159	ƒ	174	«	190	†	207	†	223	■	239	∞	255	
143	Å	192	L	175	»	191	†	208	⌘	224	n	240	∞		

31-Marzo-10

## Código EBCDIC

Extended Binary Coded Decimal Interchange Code

Posición 0 y 1

Posiciones 4,5,6 y 7	00 Bit para dígitos 2 y 3				01 Bit posiciones 2 y 3				10 Bit para dígitos 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
	0000						·							>	<	#
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 BCD

Decimal expresado en Binario

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

$$825_{(10)} = 1000 \ 0010 \ 0101 \ (BCD)$$

$$m = A^* + B^* + C^* + D^*$$

30-Marzo-10

Exercio 3 o EX3

825<sub>(10)</sub>  
 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

Codigo Ponderado

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

Codigo Reflejado

2421 o AIKEN

825<sub>(10)</sub>  
 1110 0010 1011 (2421)

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

Ponderado

$$m = I \cdot 2 + J \cdot 4 + K \cdot 2 + L \cdot 1$$

Donde m es el valor decimal

5211

(825)<sub>(10)</sub>  
 1110 0011 1000 (5211)  
 1000 0010 0101 (BCD)  
 1011 0101 1000 (EX3)  
 1110 0010 1011 (2421)

Ponderado

$$M \cdot 5 + N \cdot 2 + O \cdot 1 + P \cdot 1$$

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

31-Marzo-10

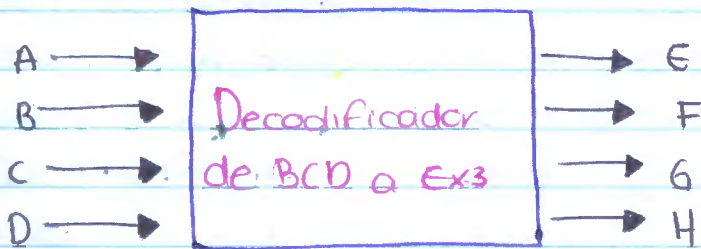
	5211	BCD	Exceso 3	2421
n	WXYZ	ABCD	EFGH	IJKL
0	0000	0000	0011	0000
1	0001	0001	0100	0001
2	0011	0010	0101	0010
3	0101	0011	0110	0011
4	0111	0100	0111	0100
5	1000	0101	1000	1001
6	1010	0110	1001	1100
7	1100	0111	1010	1101
8	1110	1000	1011	1110
9	1111	1001	1100	1111
No usadas	0010	1010	0000	0101
	0100	1011	0001	0110
	0110	1100	0010	0111
	1001	1101	1101	1000
	1011	1110	1110	1001
1101	1111	1111	1010	

0000  
0001  
0010  
0011  
0100  
0101  
0110  
0111  
1000  
1001  
1010  
1011  
1100  
1101  
1110  
1111

últimas 3 primeras  
3 últimas del 5-10

Convierta de un código BCD a un código EX3

Diagrama de Bloques



31-Marzo-10

Tabla de Verdad

Equivatente en EX3

m	BCD				EX3				BCD → EX3	
	A	B	C	D	E	F	G	H	0	3
0	0	0	0	0	0	0	1	1	0(10)	
1	0	0	0	1	0	1	0	0	BCD → EX3	BCD → EX3
2	0	0	1	0	0	1	0	1	1	4 5 8
3	0	0	1	1	0	1	1	0	1(10)	BCD → EX3
4	0	1	0	0	0	1	1	1	BCD → EX3	6 9
5	0	1	0	1	1	0	0	0	2	5 BCD → EX3
6	0	1	1	0	1	0	0	1	BCD → EX3	7 10
7	1	1	1	1	1	0	1	0	3 - 6	BCD → EX3
8	1	0	0	0	1	0	1	1	BCD → EX3	8 11
9	1	0	0	1	1	1	0	0	4 7	BCD → EX3
10	1	0	1	0	X	X	X	X	9	12
11	1	0	1	1	X	X	X	X	9(10)	
12	1	1	0	0	X	X	X	X	Combinaciones no usadas en el BCD	
13	1	1	0	1	X	X	X	X		
14	1	1	1	0	X	X	X	X		
15	1	1	1	1	X	X	X	X		

X = [A, B, C, D]  
 Y = [E, F, G, H]

BCD	EX3
X	Y
0	3
1	4
2	5
3	6
4	7
5	8
6	9
7	10
8	11
9	12

Tabla de verdad

31-Marzo-10

MODULE <b>bcdex</b>	TRUTH_TABLE (X->Y)	TEST_VECTORS (X->Y)
"Convertidor de código"	0->3;	0->3;
"BCD a Ex3"	1->4;	1->4;
" Entradas"	2->5;	2->5;
<b>A,B,C,D</b> PIN 1..4;	3->6;	3->6;
"Salidas"	4->7;	4->7;
<b>E,F,G,H</b> PIN 16..19 ISTYPE 'COM';	5->8;	5->8;
<b>X=[A,B,C,D];</b>	6->9;	6->9;
<b>Y=[E,F,G,H];</b>	7->10;	7->10;
	8->11;	8->11;
	9->12;	9->12;
		END

En ex3 No se pone 'dc, com' solo 'com' porque no respeta la tabla de verdad

### Ecuaciones Minimias

	AB				AB			
	00	01	11	10	00	01	11	10
00	0	4	12	8	0	4	12	8
01	1	5	13	9	1	5	13	9
11	3	7	15	11	3	7	15	11
10	2	6	14	10	2	6	14	10

	AB				AB			
	00	01	11	10	00	01	11	10
00	0	4	12	8	0	4	12	8
01	1	5	13	9	1	5	13	9
11	3	7	15	11	3	7	15	11
10	2	6	14	10	2	6	14	10

31-Marzo-10

### Ecuaciones

$$E = !C!D \& !C \& !A \# !B \& !A;$$

$$F = !(D \& B \# C \& B \# !D \& !C!B);$$

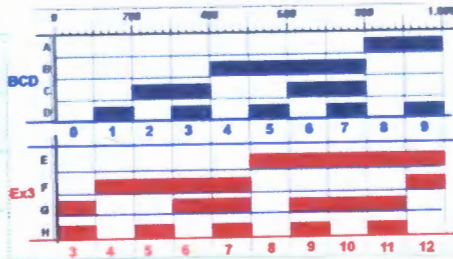
$$G = !(!D \& C \# D \& !C);$$

$$H = !(D);$$

### Distribución de las terminales (Pin Out)

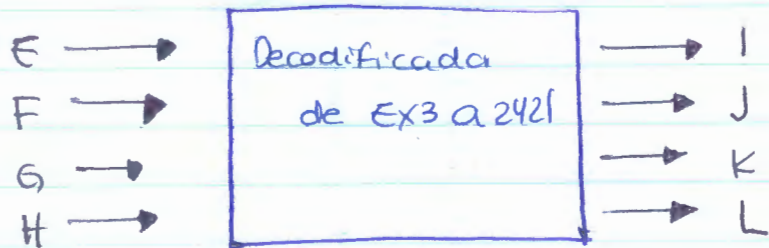


### Simulación



Convierta de un código EX3 a un código 2421

### Diagrama de Bloques





31-Marzo-10

Tabla de verdad

m	EX3				2421				
	E	F	G	H	I	J	K	L	
0	0	0	0	0	X	X	X	X	Combinaciones no usadas en EX3
1	0	0	0	1	X	X	X	X	
2	0	0	1	0	X	X	X	X	
3	0	0	1	1	0	0	0	0	Equivalente en 2421
4	0	1	0	0	0	0	0	1	EX3 → 2421
5	0	1	0	1	0	0	1	0	3      0
6	0	1	1	0	0	0	1	1	0011 (ex3) = 0 (10)
7	0	1	1	1	0	1	0	0	4      1
8	1	0	0	0	1	0	1	1	0100 (ex3) = 1 (10)
9	1	0	0	1	1	1	0	0	5      2 = 2 (10)
10	1	0	1	0	1	1	0	1	6      3 = 3 (10)
11	1	0	1	1	1	1	1	0	7      4 = 4 (10)
12	1	1	0	0	1	1	1	1	8      11 = 5 (10)
13	1	1	0	1	X	X	X	X	9      12 = 6 (10)
14	1	1	1	0	X	X	X	X	10      13 = 7 (10)
15	1	1	1	1	X	X	X	X	11      14 = 8 (10)
									12      15 = 9 (10)

Tabla de verdad

	EX3	2421
X = [E, F, G, H]	X	Y
Y = [I, J, K, L]	0	1
	1	2
	2	3
	3	4
	4	5
	5	6
	6	7
	7	8
	8	9
	9	10
	10	11
	11	12
	12	13
	13	14
	14	15

31-Marzo-10

**MODULE EXAIKEN**

"Convertidor de código

" Ex3 A 2421

" Entradas

E,F,G,H PIN 1..4:

" Salidas

I,J,K,L pin 16..19 Istyle 'com';

X=[E,F,G,H];

Y=[I,J,K,L];

**Truth Table**

(X→Y)

3→0;

4→1;

5→2;

6→3;

7→4;

8→11;

9→12;

10→13;

11→14;

12→15;

**Test\_vectors**

(X→Y)

3→0;

4→1;

5→2;

6→3;

7→4;

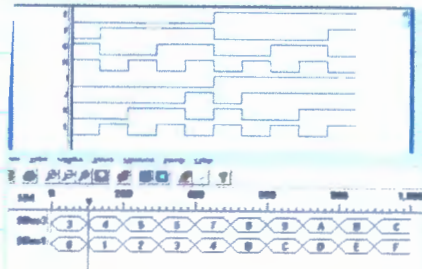
8→11;

9→12;

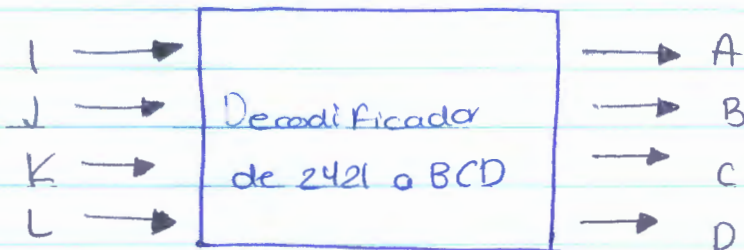
10→13;

11→14;

12→15;



Convierta de un código 2421 a un código BCD  
Diagrama de Bloques



# Tabla de Verdad

m	2421				BCD			
	I	J	K	L	A	B	C	D
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	0
3	0	0	1	1	0	0	1	1
4	0	1	0	0	0	1	0	0
5	0	1	0	1	X	X	X	X
6	0	1	1	0	X	X	X	X
7	0	1	1	1	X	X	X	X
8	1	0	0	0	X	X	X	X
9	1	0	0	1	X	X	X	X
10	1	0	1	0	X	X	X	X
11	1	0	1	1	0	1	0	1
12	1	1	0	0	0	1	1	0
13	1	1	0	1	0	1	1	1
14	1	1	1	0	1	0	0	0
15	1	1	1	1	1	0	0	1

Combinaciones no usadas

Equivalente en BCD

```

MODULE ybcd
"convertidor de 2421 a BCD
"Entradas
I,J,K,L pin 1..4;
"Salidas
A,B,C,D pin 19..16 istype 'dc.com';
M=[I,J,K,L];
N=[A,B,C,D];

```

```

truth table
(M->N)
0->0;
1->1;
2->2;
3->3;
4->4;
11->5;
12->6;
13->7;
14->8;
15->9;
END

```

31-Marzo-10

A	= ( K & J ):	A	= ( K & J & I ):
B	= ( ( K & J ) & ( J & I ) ):	B	= ( ( I & ( K & J ) & ( K & J & I ) & ( L & K & J & I ) ):
C	= ( ( K & I ) & ( K & I ) ):	C	= ( ( K & J & I ) & ( K & J & I ) ):
D	= ( I I ):	D	= ( ( I I & ( K & J & I ) & ( J & I ) ):

5211 a un Código BCD

M →	1		19 →	A
N →	2	5211	18 →	B
O →	3	9	17 →	C
P →	4	BCD	16 →	D

Tabla de Verdad

Combinaciones no usadas

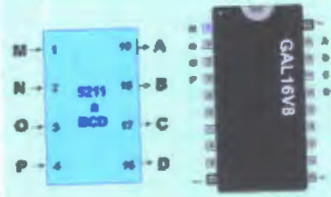
Equivalente en BCD

m	5211				BCD			
	M	N	O	P	A	B	C	D
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	X	X	X	X
3	0	0	1	1	0	0	1	0
4	0	1	0	0	X	X	X	X
5	0	1	0	1	0	0	1	1
6	0	1	1	0	X	X	X	X
7	0	1	1	1	0	1	0	0
8	1	0	0	0	0	1	0	1
9	1	0	0	1	X	X	X	X
10	1	0	1	0	0	1	1	0
11	1	0	1	1	X	X	X	X
12	1	1	0	0	0	1	1	1
13	1	1	0	1	X	X	X	X
14	1	1	1	0	1	0	0	0
15	1	1	1	1	1	0	0	1

31-Marzo-10

5211	BCD
Y	W
0	0
1	1
3	2
5	3
7	4
8	5
10	6
12	7
14	8
15	9

MODULE xyz  
 "convertidor de 5211 a BCD"  
 "Entradas"  
 M,N,O,P pin 1-4;  
 "Salidas"  
 A,B,C,D pin 19-16 istype 'dc.com';  
 Y=(L,K,I);  
 W=(A,B,C,D);



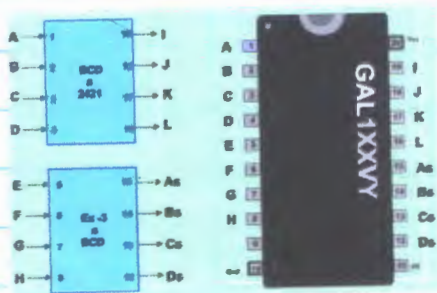
5211 a un código BCD

MODULE xyz  
 "convertidor de 5211 a BCD"  
 "Entradas"  
 M,N,O,P pin 1-4;  
 "Salidas"  
 A,B,C,D pin 19-16 istype 'dc.com';  
 Y=(L,K,I);  
 W=(A,B,C,D);

truth table  
 (Y->W)  
 0->0;  
 1->1;  
 3->2;  
 5->3;  
 7->4;  
 8->5;  
 10->6;  
 12->7;  
 14->8;  
 15->9;  
 END

5211	BCD
Y	W
0	0
1	1
3	2
5	3
7	4
8	5
10	6
12	7
14	8
15	9

## Proyecto Adicional 6



Hacer los dos de BCD-2421 y de E3-3 a BCD para el Lunes 26-Abril

31-Marzo-10

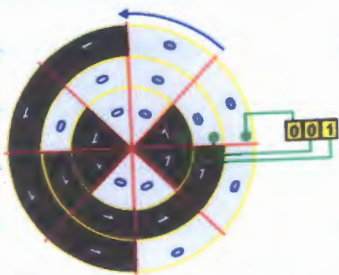
## Codigos binarios

### Codigo Gray

- El codigo Gray es otro tipo de codigo basado en un sistema binario pero de una construccion muy distinta a la de los demas codigos.
- Es un codigo continuo, ciclico y no ponderado.
- Su principal caracteristica es que entre dos numeros consecutivos solo cambia un solo bit.

	A	B	C	N(10)
0	0	0	0	0
1	0	0	1	1
2	0	1	0	3
3	0	1	1	2
4	1	0	0	6
5	1	0	1	7
6	1	1	0	5
7	1	1	1	4

### Codificador de posicion angular (encoder de tres bit's)



31-Marzo-1c

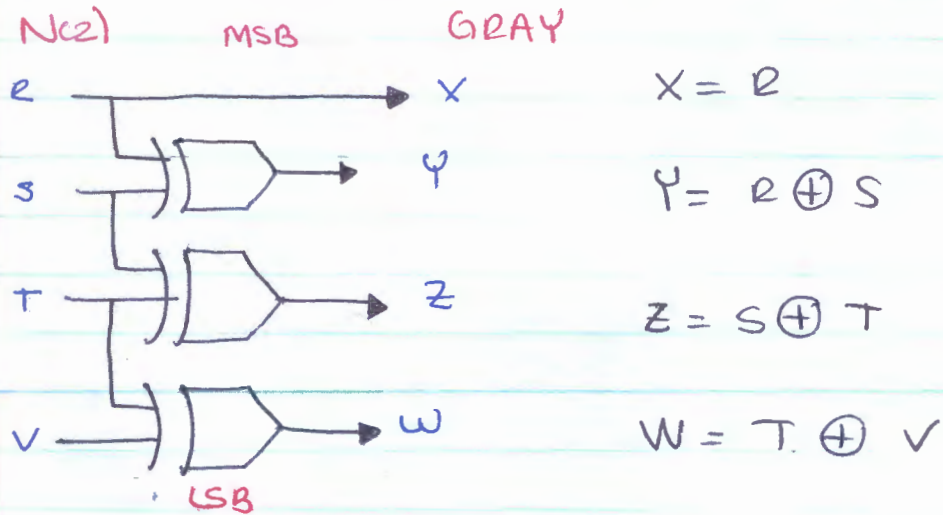
### Mapa de Karnaugh

Entre cuadros adyacentes solo hay un cambio

(FG) AB

	00	01	11	10
00	0000	0100	1100	1000
01	0001	0101	1101	1001
11	0011	0111	1111	1011
10	0010	0110	1110	1010

### Conversion de N(2) a GRAY

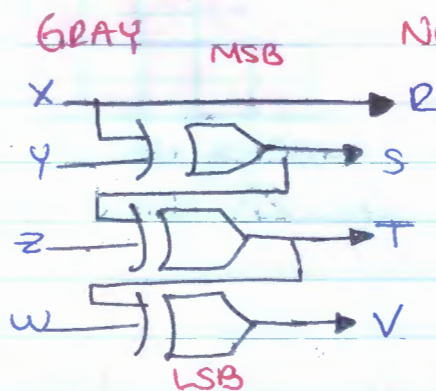


31-Marzo-10

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

Ver dispositivo para las Salidas porque se mueven

### Conversion de GRAY a N(2)



$$r = x$$

$$s = x \oplus y$$

$$t = x \oplus y \oplus z = s \oplus z$$

$$v = x \oplus y \oplus z \oplus w = t \oplus w$$



31-Marzo-10

m	Gray				Binario			
	e	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	1	0	1	0

Ver diapositiva 76  
porque cambian

## Código JOHNSON

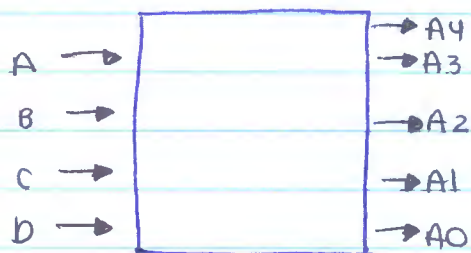
Es un código continuo y cíclico, los números de combinaciones que podemos obtener son  $2^n$ , donde  $n$  es el número de bits del código.

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

31-NOV-10

m	A4	A3	A2	A1	A0
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	0	1	1
3	0	0	1	1	1
4	0	1	1	1	1
5	1	1	1	1	1
6	1	1	1	1	0
7	1	1	1	0	0
8	1	1	0	0	0
9	1	0	0	0	0

m	A	B	C	D	A4	A3	A2	A1	A0
0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	1
2	0	0	1	0	0	0	0	1	1
3	0	0	1	1	0	0	1	1	1
4	0	1	0	0	0	1	1	1	1
5	0	1	0	1	1	1	1	1	1
6	0	1	1	0	1	1	1	1	0
7	0	1	1	1	1	1	1	0	0
8	1	0	0	0	1	1	0	0	0
9	1	0	0	1	1	0	0	0	0
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					



dec

X

# Flip Flops

31-Marzo-10

## Circuitos Logicos Secuenciales

Los bloques basicos para construir los circuitos logicos Secuenciales son los Flip-Flops.

La importancia de los Flip-Flops se debe a su caracteristica de memoria.

Los Flip-Flops tambien se denominan

- Elemento basico de memoria
- Cerrojos o candados
- Multivibradores biestables o binarios

Un Flip Flop es un circuito electronico digital, llamado tambien multivibrador biestable que tiene dos estados estables (0,1)

bi (Del lat. bi-, por bis). elem campos. Significa "do" o "dos veces"

Estable (Del lat. stabilis) adf. Que permanece en un lugar durante mucho tiempo

El Flip-Flop es el elemento de memoria mas pequeno que existe y que es capaz de almacenar un numero binario de un solo bit.

**Memoria:** Dispositivo fisico, generalmente electronico, en el que se almacenan datos e instrucciones para recuperarlos y utilizarlos posteriormente.

31-Marzo-10

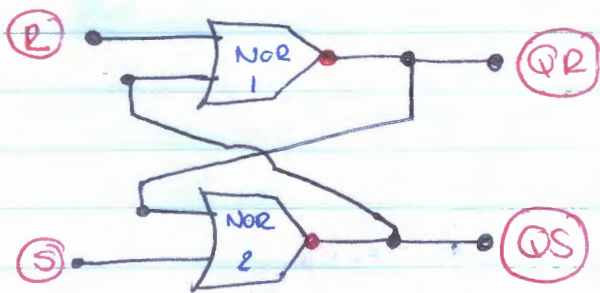
Puede almacenar solo un uno (1) o un cero (0) y permanece indefinidamente (estable) en uno de sus dos estados posibles aunque haya desaparecido la señal de control que provocó su transición al estado actual

## Tipos de Flip Flops

D	Data o Datos	Flip Flop comerciales (estándares)
JK		
RS	Reset Set	Flip Flop basados en compuertas, basados retroalimentados o partiendo de FF's comerciales
T	Toggle	
SC	Set Clear	

El 6AL16V8 tiene 8 FFs tipo D

## Flip Flop RS (Reset, Set)



$$Qr = ?$$

$$Qs = ?$$

$$Qr = (R + Qs)'$$

$$Qs = (S + Qr)'$$

A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0



$$Qr = Qs'$$

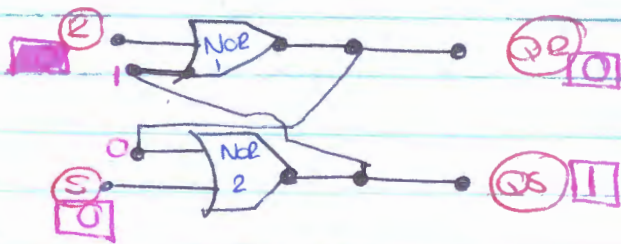
Con cualquier entrada que sea uno la salida es cero

31-Marzo-10

Que pasará con los valores de Salida si cambiamos  $R=0$

Tabla Característica

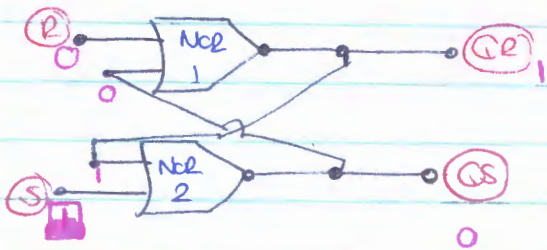
R	S	QR	QS
1	0	0	1
0	0	0	1
0	1		
0	0		



Que pasará con los valores de salida si cambiamos  $S=1$

Tabla Característica

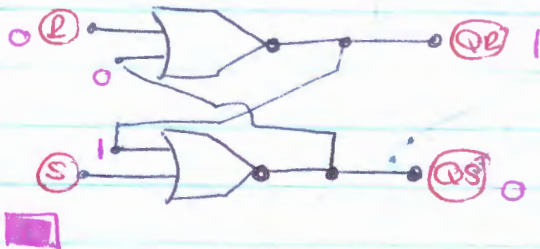
R	S	QR	QS
1	0	0	1
0	0	0	1
0	1	1	0
0	0		



Que pasará con los valores de salida si cambiamos  $S=0$

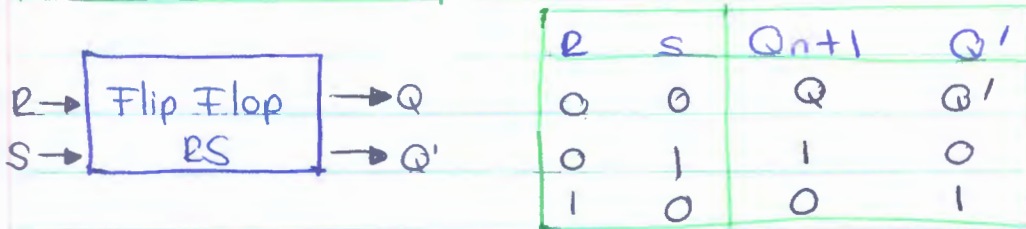
Tabla Característica

R	S	QR	QS
1	0	0	1
0	0	0	1
0	1	1	0
0	0	1	0



31-Marzo-10

Condición	R	S	Q	Q'
Reset	1	0	0	1
Memoria	0	0	0	1
Set	0	1	1	0
Memoria	0	0	1	0



Q<sub>n+1</sub> = el valor proximo

R	S	Q <sub>n+1</sub>	Q'
0	0	Q	Q'
0	1	1	0
1	0	0	1
1	1	0	0

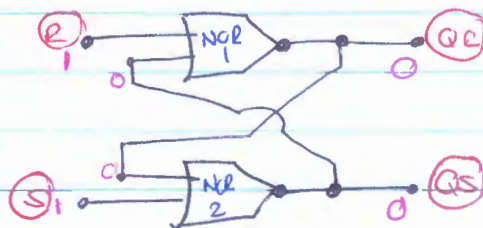
MODULE ffsa  
 "Entradas  
 R,S pin 1,2:  
 "Salidas  
 QR, QS pin 19,18 lstype 'com':  
 equations  
 QR=(R#QS);  
 QS=(S#QR);  
 END

([R,S]->[QR, QS])  
 [0,1]->[.x...x];  
 [0,0]->[.x...x];  
 [1,0]->[.x...x];  
 [0,0]->[.x...x];  
 [1,1]->[.x...x];  
 END

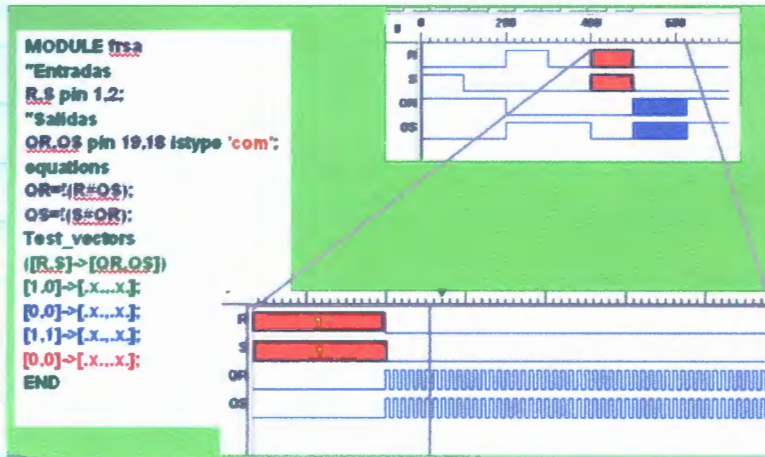
## Flip Flop RS Condición NO estable

Que pasará con los valores de salida si R=1 y S=1

R	S	Q <sub>n+1</sub>	Q'
0	0	Q	Q'
0	1	1	0
1	0	0	1
1	1	0	0



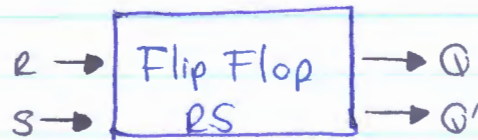
31-Marzo-10



## Flip Flop RS

Tabla característica

R	S	$Q_{n+1}$	$Q'$
0	0	Q	$Q'$
0	1	1	0
1	0	0	1
1	1	0	0



## Flip Flop RS

Tabla característica

R	S	$Q_{n+1}$	$Q'$
0	0	Q	$Q'$
0	1	1	0
1	0	0	1
1	1	?	?

31-Marzo-10

	Entradas de control		Salidas		Flip Flop RS Reset Set
	R	S	$Q_{n+1}$	$Q'$	
Memoria	0	0	$Q_n$	$Q'$	Con la combinacion $r=0$ y $s=0$ el valor proximo de $Q$ ( $Q_{n+1}$ ) se mantiene
Set	0	1	1	0	Con la combinacion $r=0$ y $s=1$ el valor proximo $Q$ ( $Q_{n+1}$ ) es igual a 1 (SET)
Reset	1	0	0	1	Con la combinacion $r=1$ y $s=0$ el valor proximo de $Q$ ( $Q_{n+1}$ ) es igual a 0 (Reset)
X	1	1	X	X	Condicion no estable

## Flip Flop RS

Señal de Sincronia CK

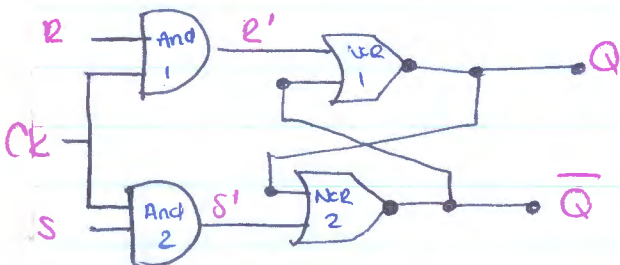
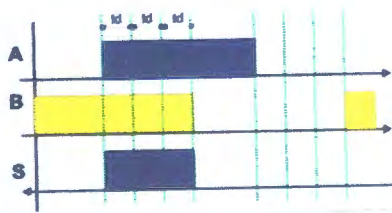
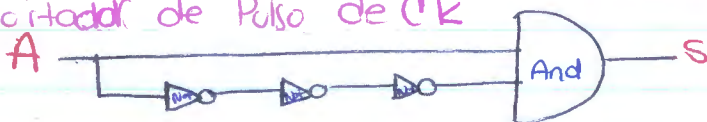


Tabla Caracteristica

CK	R	S	$Q_{n+1}$	$Q'$
0	X	X	Q	$Q'$
1	0	0	Q	$Q'$
1	0	1	1	0
1	1	0	0	1
1	1	1	?	?

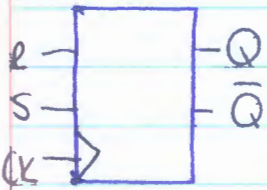
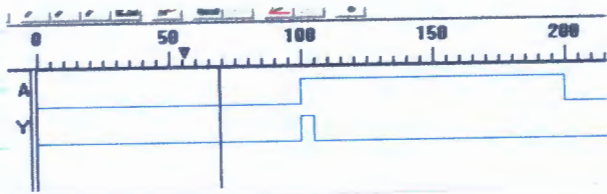
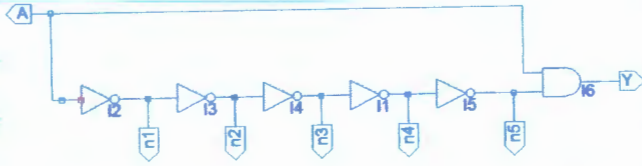
Receptor de Pulso de CK





31-Marzo-10

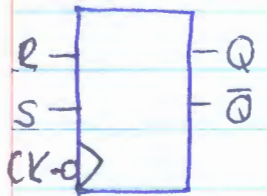
## Recortador de Pulso de CK



CK

0 ↑ 1

Transición  
Positiva

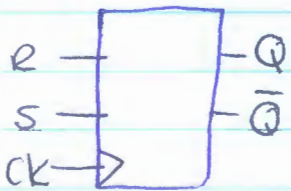


CK

0 1 ↓

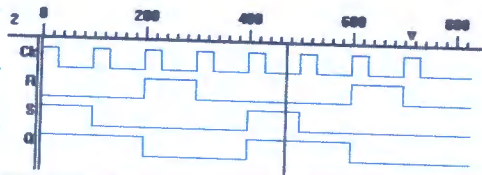
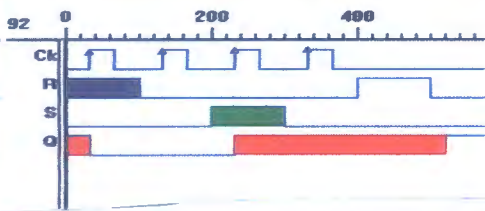
Transición  
Negativa

CK ↑

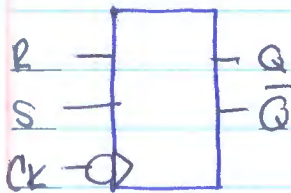


CK	R	S	Q <sub>n+1</sub>	Q'	CK	R	S	Q <sub>n+1</sub>
0	x	x	Q	Q'	0	x	x	Q
↑	0	0	Q	Q'	↑	0	0	Q
↑	0	1	1	0	↑	0	1	0
↑	1	0	0	1	↑	1	0	1
↑	1	1	?	?	↑	1	1	Q

31 - Marzo - 10

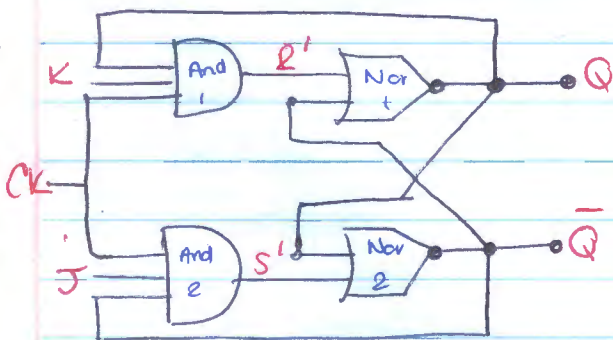


CK ↓



CK	R	S	Q <sub>n+1</sub>	Q
0	X	X	Q	Q'
↓	0	0	Q	Q'
↓	0	1	1	0
↓	1	0	0	1
↓	1	1	?	?

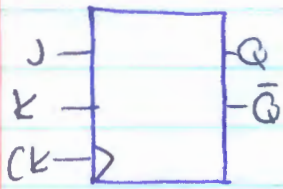
Flip Flop JK



CK	J	K	Q <sub>n+1</sub>	Q'
0	X	X	Q	Q
↑	0	0	Q	Q
↑	0	1	0	1
↑	1	0	1	0
↑	1	1	?	?

Dispositivos Combin

31-Marzo-10



CK	J	K	Q <sub>n+1</sub>	Q'
0	x	x	Q	Q'
↑	0	0	Q	Q'
↑	0	1	0	1
↑	1	0	1	0
↑	1	1	Q'	Q

## Preguntas de Repaso

Que es un Flip Flop?

R=

Cuantos Tipos hay?

R=

Como Funcionan?

R=

Que significado tiene e?

R=

Que significado tiene s?

R=

31-Marzo-10

Cual es la combinacion no recomendada en el RS?

R=

Porque no es recomendada la combinacion R=1 y S=1

R=

Cual es el proposito del ck?

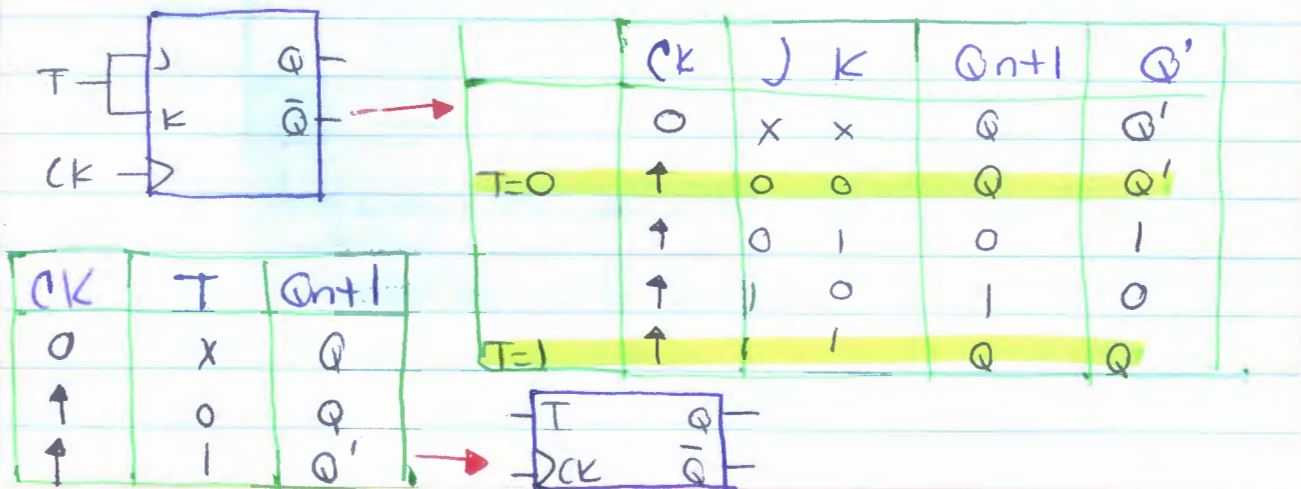
R=

Como debe ser la caracteristica de la señal de ck

R=

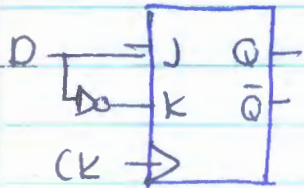
Cual es la diferencia entre el Flip Flop JK y el RS

## Flip Flop T Toggle



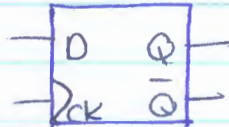
31-Marzo-10

# Flip Flop D Data



	ck	J	K	$Q_{n+1}$	$Q'$
	0	x	x	Q	Q'
T=0	↑	0	0	Q	Q'
D=0	↑	0	1	0	1
D=1	↑	1	0	1	0
T=1	↑	1	1	Q'	Q

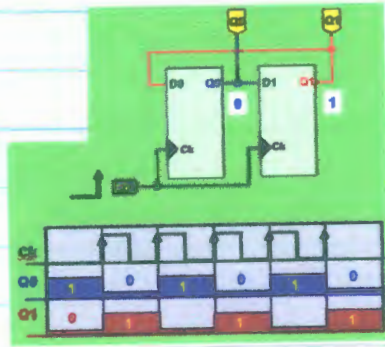
ck	D	$Q_{n+1}$
0	x	Q
↑	0	0
↑	1	1



Reset - Set			Toggle																							
	<table border="1"> <thead> <tr> <th>Estados de control</th> <th>Salida</th> </tr> <tr> <th>R</th> <th>S</th> <th><math>Q_{n+1}</math></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td><math>Q_n</math></td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>X</td> </tr> </tbody> </table>	Estados de control	Salida	R	S	$Q_{n+1}$	0	0	$Q_n$	0	1	1	1	0	0	1	1	X		<table border="1"> <thead> <tr> <th>T</th> <th><math>Q_{n+1}</math></th> </tr> </thead> <tbody> <tr> <td>0</td> <td><math>Q_n</math></td> </tr> <tr> <td>1</td> <td><math>Q_n'</math></td> </tr> </tbody> </table>	T	$Q_{n+1}$	0	$Q_n$	1	$Q_n'$
Estados de control	Salida																									
R	S	$Q_{n+1}$																								
0	0	$Q_n$																								
0	1	1																								
1	0	0																								
1	1	X																								
T	$Q_{n+1}$																									
0	$Q_n$																									
1	$Q_n'$																									
J-K			Data																							
	<table border="1"> <thead> <tr> <th>Estados de control</th> <th>Salida</th> </tr> <tr> <th>J</th> <th>K</th> <th><math>Q_{n+1}</math></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td><math>Q_n</math></td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td><math>Q_n'</math></td> </tr> </tbody> </table>	Estados de control	Salida	J	K	$Q_{n+1}$	0	0	$Q_n$	0	1	0	1	0	1	1	1	$Q_n'$		<table border="1"> <thead> <tr> <th>D</th> <th><math>Q_{n+1}</math></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> </tr> </tbody> </table>	D	$Q_{n+1}$	0	0	1	1
Estados de control	Salida																									
J	K	$Q_{n+1}$																								
0	0	$Q_n$																								
0	1	0																								
1	0	1																								
1	1	$Q_n'$																								
D	$Q_{n+1}$																									
0	0																									
1	1																									

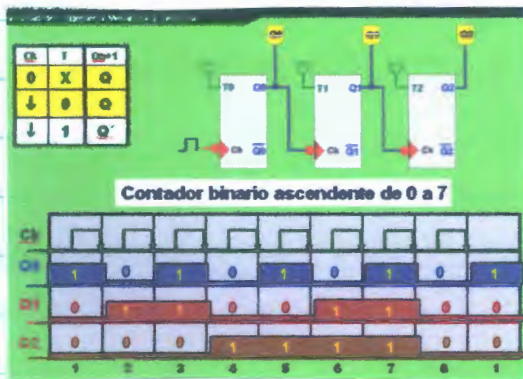
31-Marzo-10

Ck	D	Qn+1
0	X	Q
↑	0	0
↑	1	1



Ck	T	Qn+1
0	X	Q
↓	0	Q
↓	1	Q

Contador binario ascendente de 0 a 7



5-39

31-Marzo-10

## Tabla de excitación

Q	Q <sub>nt+1</sub>	Entradas de control					
		R	S	J	K	T	D
0	0	X	0	0	X	0	0
0	1	0	1	1	X	1	1
1	0	1	0	X	1	1	0
1	1	0	X	X	0	0	1

## Otras entradas de control a los FFS

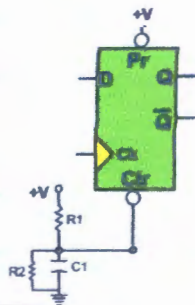
Algunos Flip Flops contienen entradas adicionales como el PRESET y el CLEAR,

Estas entradas son asincrónicas (no requieren de pulso de reloj), uno de los propósitos de estas entradas es el de establecer condiciones iniciales.

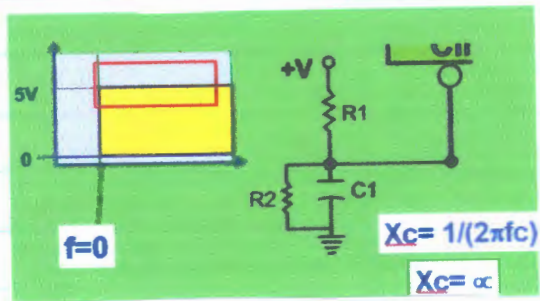
PRESET (Pr) tiene la función de que la salida Q tome el valor de uno ( $Q=1$ )

CLEAR (Clr) la de hacer que Q tome el valor de cero ( $Q=0$ )

## Otras entradas de control a los FFS



31-Marzo-10



$f \neq 0$

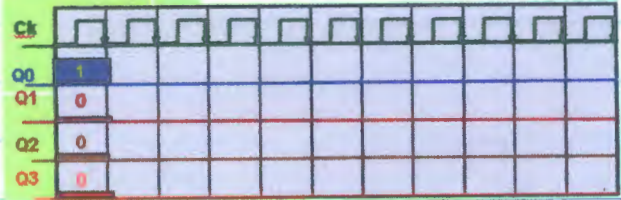
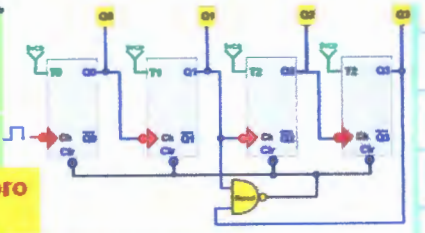


1-Abril-10

CK	T	Q <sub>n+1</sub>
0	x	Q
↓	0	Q
↓	1	Q'

CK	T	Q <sub>n+1</sub>
0	x	Q
↓	0	Q
↓	1	Q'

Hasta cual número va a contar

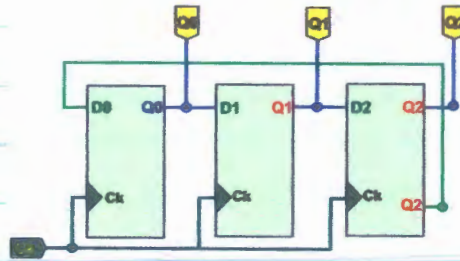


Hasta cual número va a contar

1-Abril-10

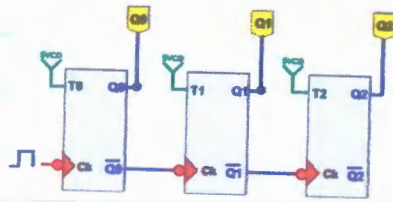
En la libreta obtenga la grafica de los siguientes circuitos:

Ck	D	Q <sub>n+1</sub>
0	X	Q
↑	0	0
↑	1	1



1-Abril-10

CK	T	Q <sub>n+1</sub>
0	X	Q
↓	0	Q
↓	1	Q'



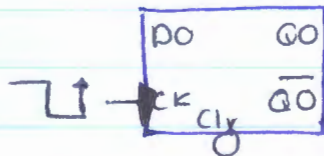
ff2

# Proyecto Adicional 7 P/Lunes Escoger 2

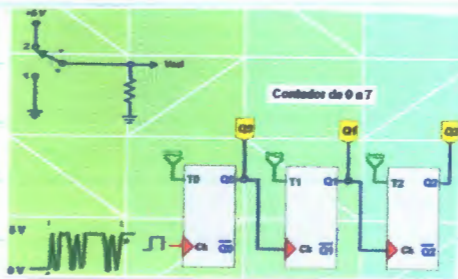
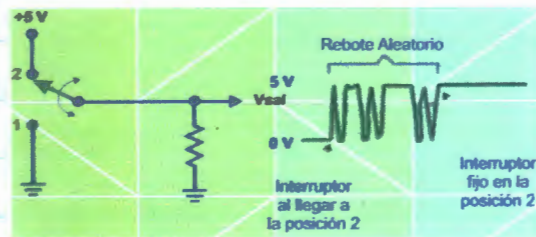
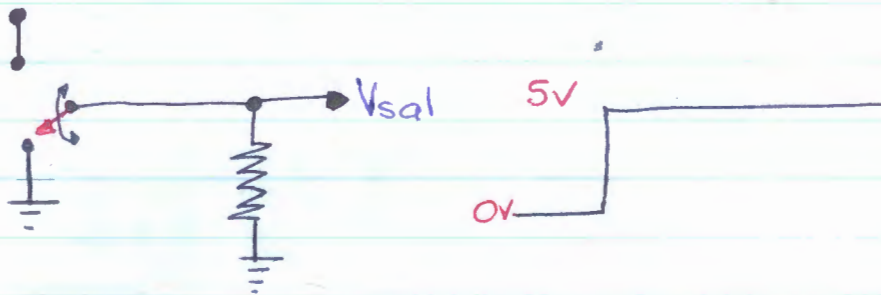
1-Abril-10

## Generación de los pulsos de Sincronía para los Flip Flops

1. Eliminador de rebotes
2. Monoestable
3. Timer 555
4. Comparata retroalimentada (SN7414)
5. Oscilador dentro del GAL



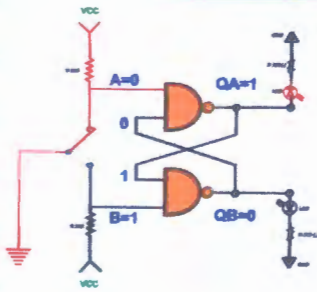
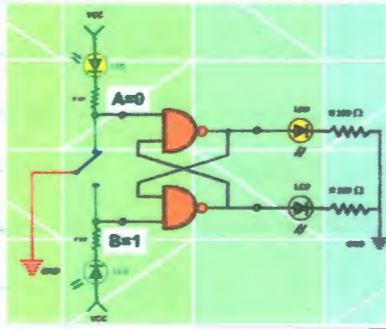
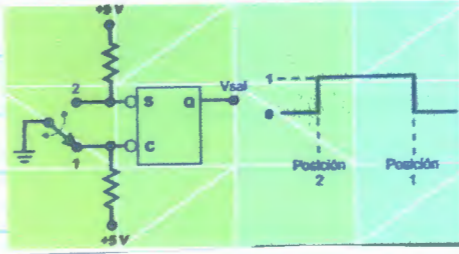
Eliminador de rebote



1-Abril-10

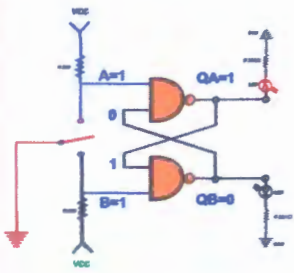
# Flip Flop SC SET CLEAR

Eliminador de Rebote)



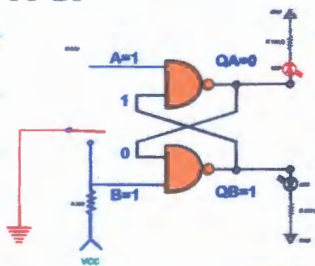
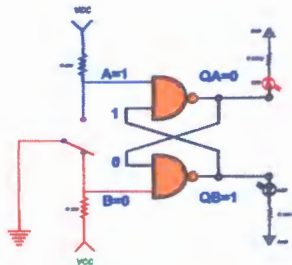
	1	2	3	4	5	6	7	8	9	10	11
SW											
A	0	1	0	1	0	1	0	1	0	1	0
B	1	0	1	0	1	0	1	0	1	0	1
QA	0	0	1	1	0	0	1	1	0	0	1
QB	1	1	0	0	1	1	0	0	1	1	0

# 1-Abbit-10



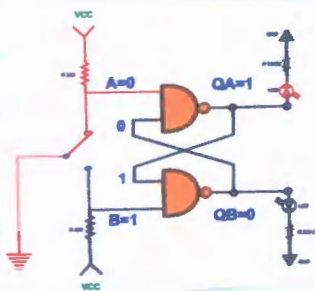
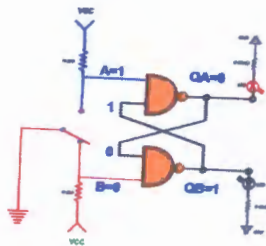
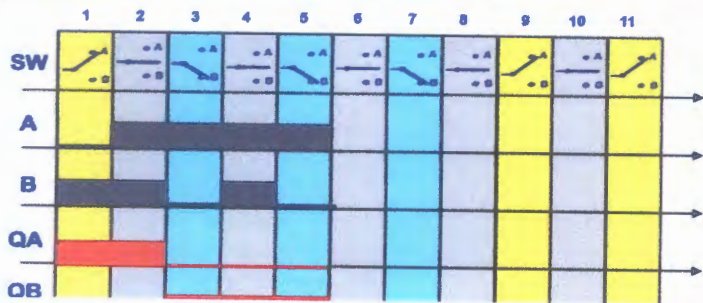
	1	2	3	4	5	6	7	8	9	10	11
SW	↗	↔	↘	↔	↘	↔	↘	↔	↘	↔	↗
A	1	0	1	0	1	0	1	0	1	0	1
B	0	1	0	1	0	1	0	1	0	1	0
QA	1	0	0	0	0	0	0	0	0	0	1
QB	0	1	0	1	0	1	0	1	0	1	0

	1	2	3	4	5	6	7	8	9	10	11
SW	↗	↔	↘	↔	↘	↔	↘	↔	↘	↔	↗
A	1	0	1	0	1	0	1	0	1	0	1
B	0	1	0	1	0	1	0	1	0	1	0
QA	1	0	0	0	0	0	0	0	0	0	1
QB	0	1	0	1	0	1	0	1	0	1	0



	1	2	3	4	5	6	7	8	9	10	11
SW	↗	↔	↘	↔	↘	↔	↘	↔	↘	↔	↗
A	1	0	1	0	1	0	1	0	1	0	1
B	0	1	0	1	0	1	0	1	0	1	0
QA	1	0	0	0	0	0	0	0	0	0	1
QB	0	1	0	1	0	1	0	1	0	1	0

1-Abril-10

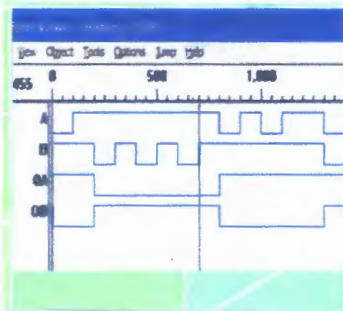


$$QA = \neg(A \& B)$$

$$QB = \neg(B \& A)$$

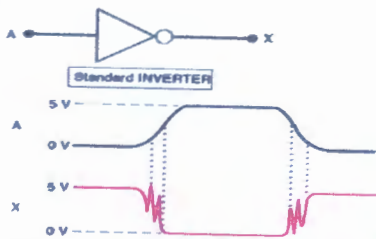
```

MODULE sc
  "entradas
  A,B pin 1,2;
  "salidas
  QA,QB pin 19,18 istype 'com';
  equations
  QA = ~(A&B);
  QB = ~(B&A);
  test vectors
  ((A,B) -> (QA,QB))
  (0,1) -> (1,x,x,x);
  (1,1) -> (1,x,x,x);
  (1,0) -> (1,x,x,x);
  (1,1) -> (1,x,x,x);
  (1,0) -> (1,x,x,x);
  (1,1) -> (1,x,x,x);
  (1,0) -> (1,x,x,x);
  (1,1) -> (1,x,x,x);
  (0,1) -> (1,x,x,x);
  (1,1) -> (1,x,x,x);
  (0,1) -> (1,x,x,x);
  (1,1) -> (1,x,x,x);
  (1,1) -> (1,x,x,x);
  (1,1) -> (1,x,x,x);
  (0,0) -> (1,x,x,x);
  END
  
```

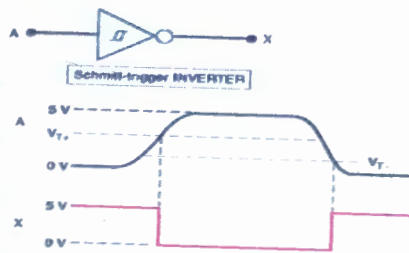




## Rebote al cambio de Sena 1



## Disparador Schmitt



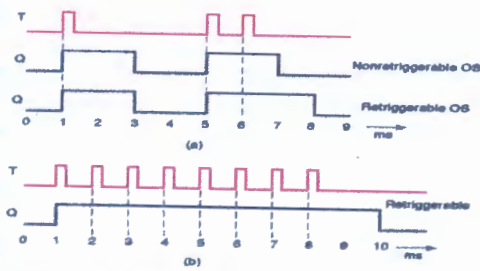
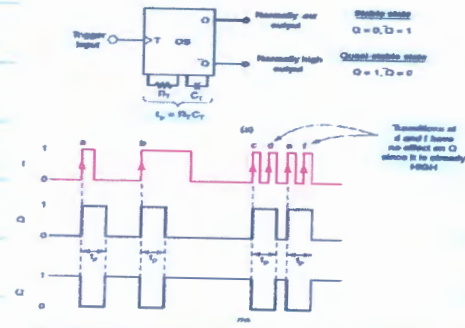
## Multivibradores

- Biestable (Flip Flops)
- Monoestable (un estado estable y otro transitorio)
  - Redisparables
  - No Redisparables
- Astable (sin estabilidad)
  - Time1



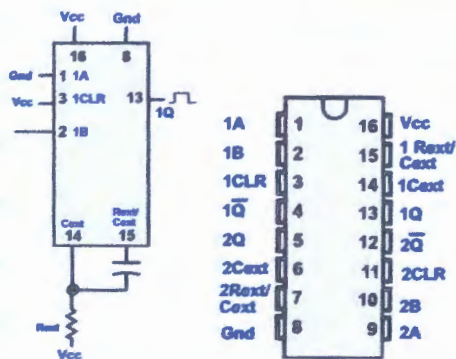
1-Abril-10

# Multivibrador Monestable

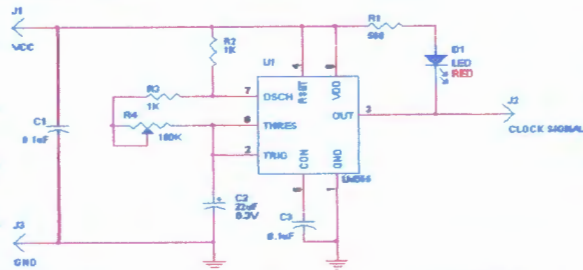


# SN74221 DUAL MONOSTABLE MULTIVRATORS WITH SCMITT-TRIGGER INPUTS

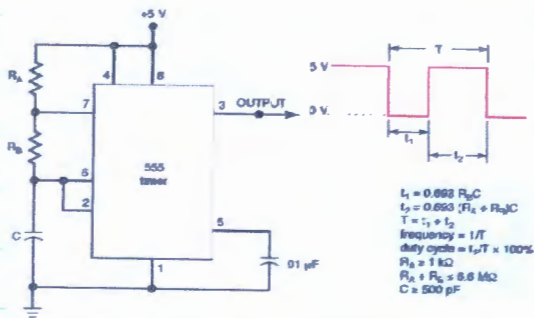
Entradas			Salidas	
CLR	A	B	Q	Q̄
L	X	X	L	H
X	H	X	L	H
X	X	L	L	H
H	L	↑	⌊	⌋
H	↓	H	⌊	⌋
↑	L	H	⌊	⌋



Astable

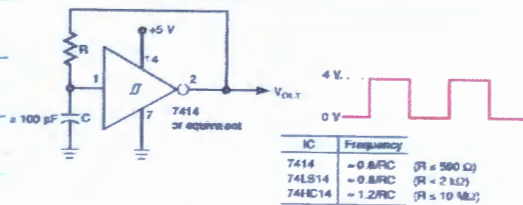


Multivibrador Astable



**Ventaja:** Amplio rango de frecuencias

**Desventaja:** Mala calidad del pulso

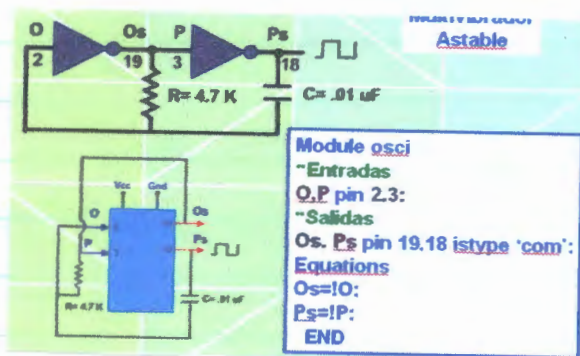


**Ventaja:** Buena calidad del Pulso

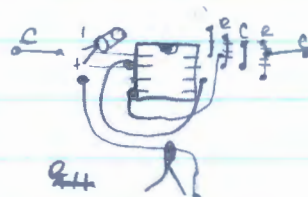
**Desventaja:** Poco rango de frecuencias

L-Abril-10

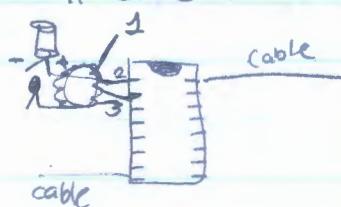
## Multibrador Astable



SN7414

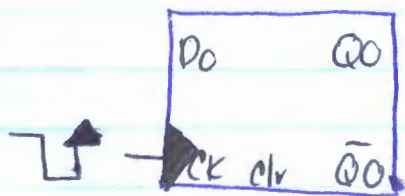


Timer 555

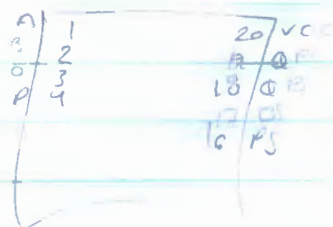


## Generación de los pulsos de sincronía para los Flip Flops

1. Eliminador de rebotes (pag. 123) ✓
2. Monostable (SN74221)
3. Timer 555 (apendice D)
4. Compuerta retroalimentada (SN7414 en presentación)
5. Oscilador dentro del GAL (Pag. 127) ✓



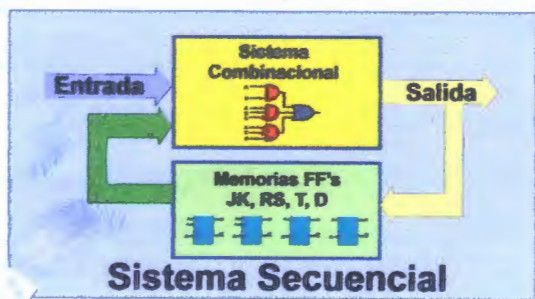
ff3



# Diseño Secuencial

1-Abril-10

Un **sistema secuencial** es aquel sistema en donde los valores de salida no dependen únicamente de las combinaciones de entrada, sino también de la salida misma.



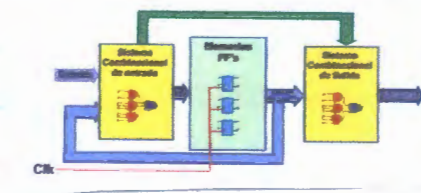
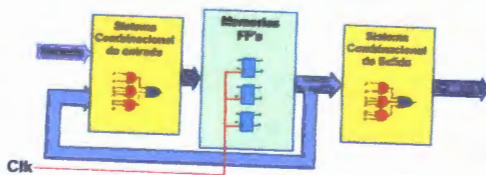
En un semáforo de que depende decidir la luz que sigue en encender?

Cual es la luz que sigue.

- **Asíncrono** que no depende de una señal de sincronía depende solo de un cambio de entrada.
- **Síncrono** en donde los cambios de estado dependen de una señal de sincronía de los Flip Flops llamada reloj, CK o CLK.

## Modelos Secuenciales Síncronos y sus representaciones

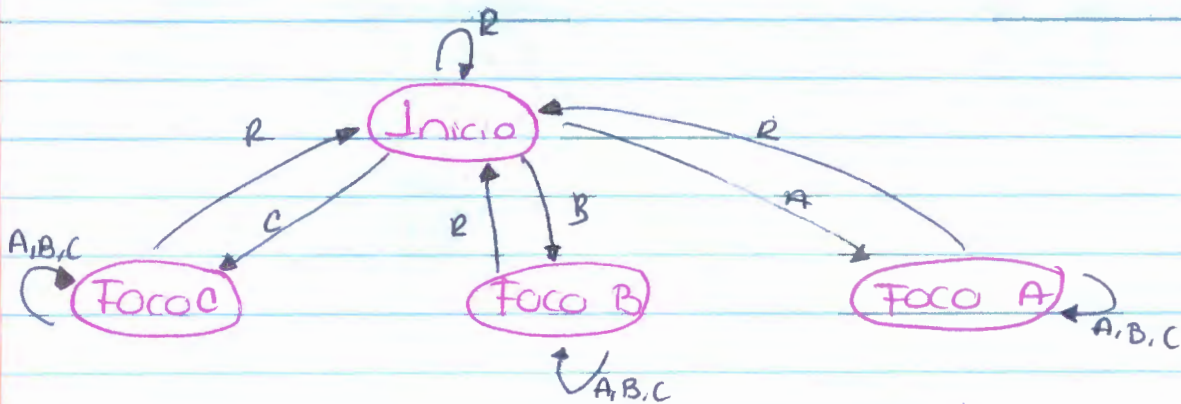
- **Maquina de Moore** en la que la salida solo depende del estado presente
- **Maquina de Mealy** en el que la salida depende tanto de la entrada como del estado presente



1-AbrilHO

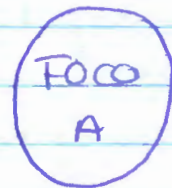
## Diagrama de Transición

Gráficos, Automatas

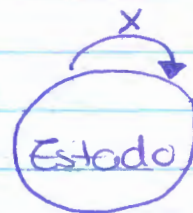


Un diagrama de transición se compone de:

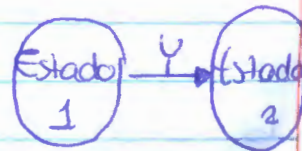
**Estados o Eventos:** Una condición o situación de un objeto, durante la cual satisface una condición, realiza una actividad o está esperando un evento.



**Transición en el mismo estado:** Una entrada **X** cuya estado próximo es el mismo que el anterior.



**Transición de estados:** Una relación entre estados que indica que un objeto que está en el primer estado realizará una acción especificada y entrará en el segundo estado cuando un evento y especificado ocurra y unas condiciones especificadas sean satisfechas.



**Entradas:** Combinaciones que establecen un cambio de eventos.

**Salidas:** Valores combinatoriales que determinan un evento.

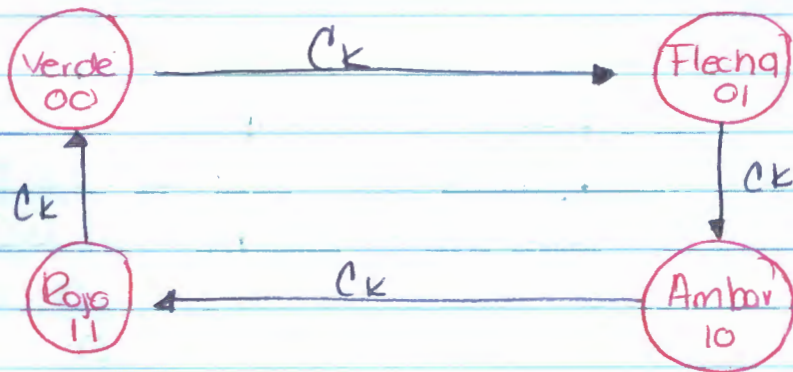
1-Abril-10

## Metodología del Diseño Secuencial

1. Especificar el Sistema (Diagrama de Transición).
2. Determinar la cantidad de Flip Flops.
3. Asignar los valores a los estados.
4. Determinar las entradas y Salidas
  - a) Entradas de sincronía reloj
  - b) Entradas combinacionales
  - c) Salidas Combinacionales
  - d) Salidas registrados (FF's)
5. Construir una tabla de Estados
6. Minimizar
7. Diagrama esquemático.
8. Implementación

### Especificar el Sistema

Para especificar el comportamiento del Sistema se puede hacer uso del diagrama de transición, en donde se indica la secuencia deseada además de las entradas, salidas y estados.



1-Abril-10

### Determinar la cantidad de Flip Flops

La cantidad de Flip Flops depende de la cantidad de los estados utilizados en el diagrama de transición como lo indica la siguiente tabla:

Estados	Cantidad de Flip Flops
2	1
3 o 4	2
5 a 8	3
9 a 16	4
17 a 32	5
33 a 64	6
65 a 128	7
129 a 256	8
257 a 512	9
513 a 1024	10

### Asignar los valores a los estados

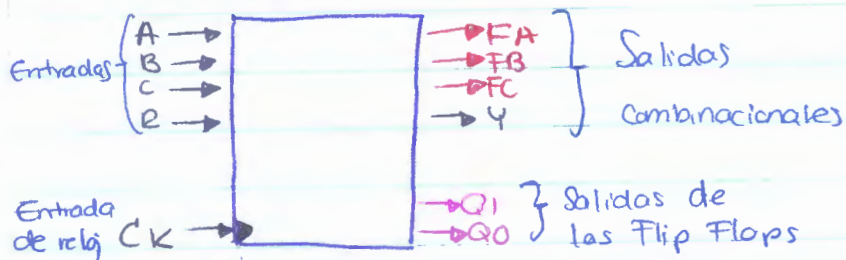
La asignación de valores a los estados puede ser al azar y corresponden a las combinaciones posibles que pueden generar las salidas Q's de los Flip Flops.

Identificar cada estado

Estados	Salidas FF's
	Q1 Q0
01	0 0
Foco A	0 1
Foco B	1 0
Foco C	1 1

## Determinar las entradas y Salidas

En esta parte se recomienda identificar las entradas y salidas del sistema secuencial, usando un diagrama de bloques como lo muestra la siguiente figura



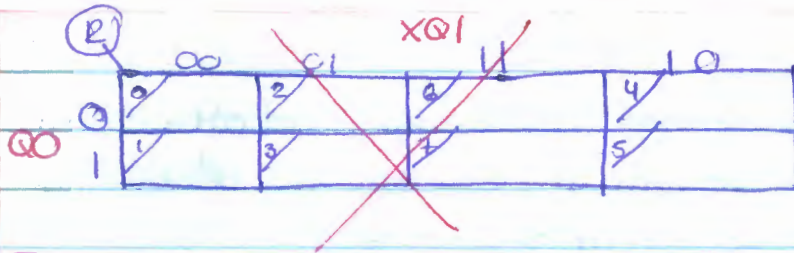
m	Entrada	Estado Presente		Estado Próximo		Entradas de Control		Salidas			
		X	Q1	Q0	Q1+1	Q0+1	D1	D0	V	F	A
0	Verde	0	0	0	0	0	0	1	0	0	0
1	Flecha	0	0	1	0	1	0	1	0	1	0
2	Amber	0	1	0	1	0	1	0	0	0	1
3	Rojo	0	1	1	1	1	1	0	0	0	1
4	Verde	1	0	0	0	1	0	1	1	0	0
5	Flecha	1	0	1	1	0	1	0	0	1	0
6	Amber	1	1	0	1	1	1	0	0	1	0
7	Rojo	1	1	1	0	0	0	0	0	0	1

## Tabla de estados

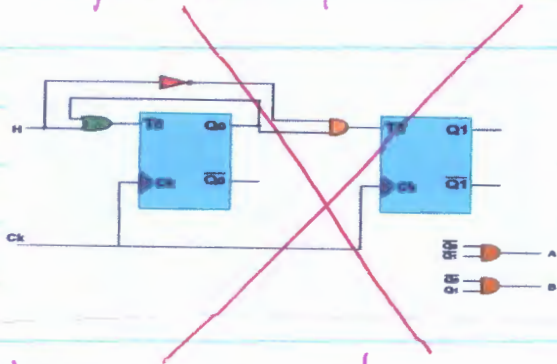
Estado Presente	Estado Próximo	
	x = 0	x = 1
Verde	Verde	Flecha
Flecha	Flecha	Ambar
Ambar	Ambar	Rojo
Rojo	Rojo	Verde



1-Abril-10



### Diagrama esquemático

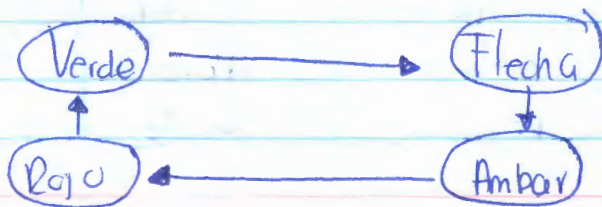


### Implementación

### Ejemplo 1

Diseñe un Sistema Secuencial síncrono que represente la operación de un semáforo de cuatro estados que se presentan en el siguiente orden: Verde, Flecha, Ambar y Rojo que cambie de estado con una señal de transición positiva llamada Ck.

En este diagrama de transición se indica la secuencia del semáforo en donde los cambios de estado se realizarán cada vez que le proporcionemos un pulso de sincronía (Ck) a los Flip Flops.



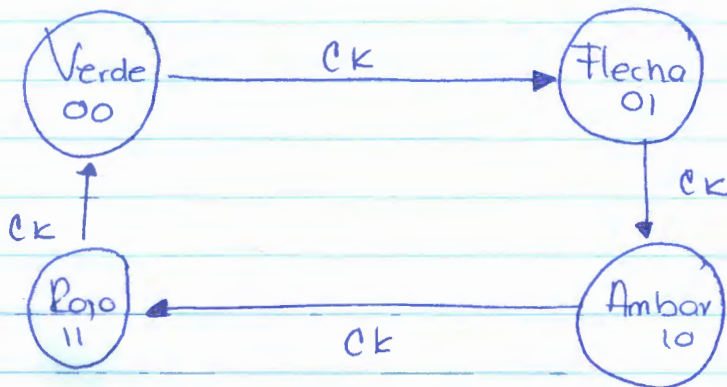
1-Abril-10

## Determinar la cantidad de Flip Flops

Nuestro ejemplo esta compuesto de cuatro estados Verde, Flecha, Ambar y Rojo por lo que requeriremos de 2 Flip Flops y para identificarlos los llamaremos  $Q1$  y  $Q0$

Asignar los valores de estado

Estados	Asignación de valores a los estados	
	$Q1$	$Q0$
Verde	0	0
Flecha	0	1
Ambar	1	0
Rojo	1	1

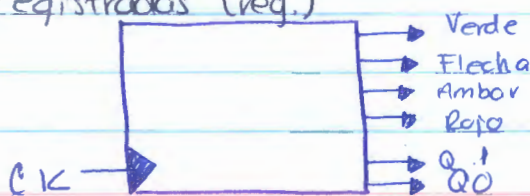


## Determinar las entradas y Salidas

Como se observa en la figura, el sistema tiene una sola entrada llamada  $ck$ .

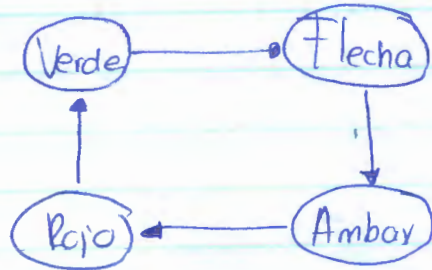
Seis salidas de las cuales Verde, Flecha, Ambar y Rojo son combinacionales.

Ademas de  $Q1$  y  $Q0$  son las salidas de los Flip Flops o tambien llamadas registradas (reg.)



Construir una tabla de estados

Estado Presente	Estado Proximo
Verde	Flecha
Flecha	Ambar
Ambar	Rojo
Rojo	Verde



La Tabla de estados con asignación de valores a los estados

m		Estado Presente		Estado Próximo		Entradas de Control		Salidas			
		01	00	01+1	00+1			V	F	A	R
0	Verde	0	0	0	1			1	0	0	0
1	Flecha	0	1	1	0			0	1	0	0
2	Ambar	1	0	1	1			0	0	1	0
3	Rojo	1	1	0	0			0	0	0	1

Tabla de Excitación

XCITACION		Entradas de control						
Qn	Qn+1	R	S	J	K	T	D	
0	0	x	0	0	x	0	0	
0	1	0	1	1	x	1	1	
1	0	1	0	x	1	1	0	
1	1	0	x	x	0	0	1	

n	Estado Presente	Estado Presente		Estado Próximo		Entradas de Control		Salidas			
		Q1	Q0	Q1+1	Q0+1	T1	T0	V	F	A	R
0	Verde	0	0	0	1	0	1	1	0	0	0
1	Flecha	0	1	1	0	1	1	0	1	0	0
2	Ambar	1	0	1	1	0	1	0	0	1	0
3	Rojo	1	1	0	0	1	1	0	0	0	1

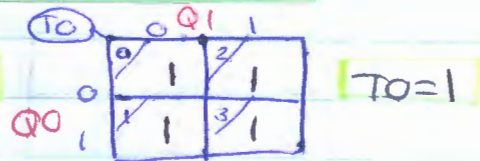
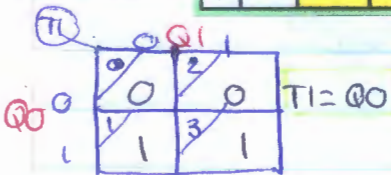
Qn	Qn+1	T
0	0	0
0	1	1
1	0	1
1	1	0

### Tabla de excitación

Qn	Qn+1	T
0	0	0
0	1	1
1	0	1
1	1	0

### Ecuaciones minimas

n	Estado Presente	Entradas		Entradas de Control		Salidas			
		Q1	Q0	T1	T0	V	F	A	R
		0	Verde	0	0	0	1	1	0
1	Flecha	0	1	1	1	0	1	0	0
2	Ambar	1	0	0	1	0	0	1	0
3	Rojo	1	1	1	1	0	0	0	1



$$T1 = Q0$$

$$T0 = 1$$

$$\text{Verde} = Q1'Q0'$$

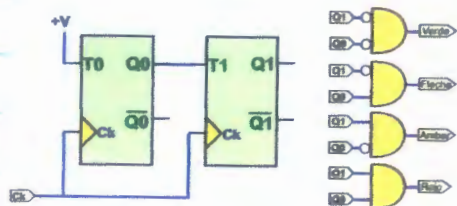
$$\text{Flecha} = Q1'Q0$$

$$\text{Ambar} = Q1Q0'$$

$$\text{Rojo} = Q1Q0$$

1-Abril-10

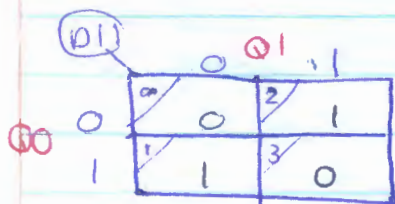
## Diagrama esquemático



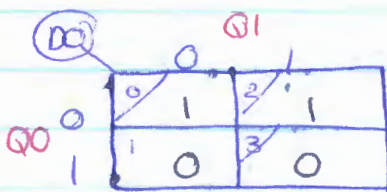
## Para Flip Flop D

n		Estado Presente		Estado Próximo		Entradas de Control		Salidas			
		Q1	Q0	Q1+1	Q0+1	D1	D0	V	F	A	R
0	Verde	0	0	0	1	0	1	1	0	0	0
1	Rojo	0	1	1	0	1	0	0	1	0	0
2	Azul	1	0	1	1	1	1	0	0	1	0
3	Amarillo	1	1	0	0	0	0	0	0	0	1

Qn	Qn+1	D
0	0	0
0	1	1
1	0	0
1	1	1



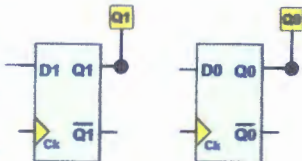
$$D1 = Q1 \oplus Q0$$



$$D0 = Q0'$$

$$D1 = Q1 \oplus Q0$$

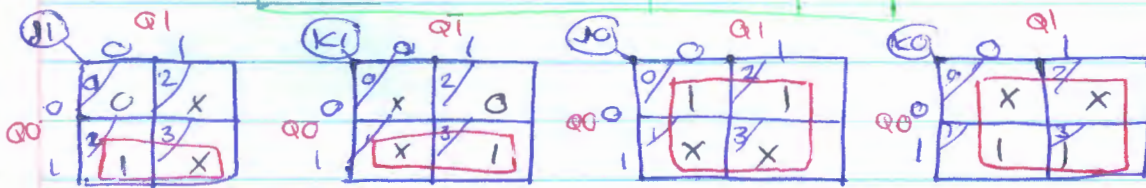
$$D0 = Q0$$



## Para Flip Flop JK

	Estado Presente		Estado Próximo		J1	K1	Entradas de Control	
	Q1	Q0	Q1+1	Q0+1			J0	K0
0	0	0	0	1	0	X	1	X
1	0	1	1	0	1	X	X	1
2	1	0	1	1	X	0	1	X
3	1	1	0	0	X	1	X	1

Qn	Qn+1	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0



$$J1 = Q0$$

$$K1 = Q0$$

$$J0 = 1$$

$$K0 = 1$$

1-Abril-10

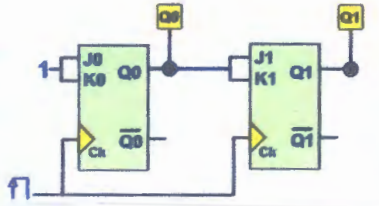
$J1 = K1 = 00$

Ambar =  $0100'$

$J0 = K0 = 1$

Rojo =  $0100$

Verde =  $01'00'$

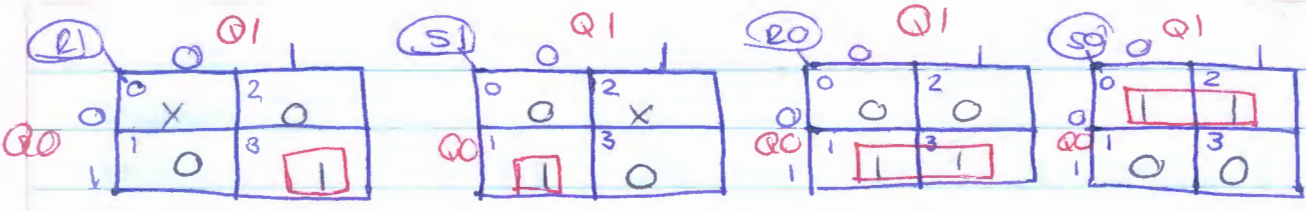


Para Flip Flop RS

	Estado Presente		Estado Próximo		R1	S1	Estados de Control	
	Q1	Q0	Q1+1	Q0+1			R0	S0
0	0	0	0	1				
1	0	1	1	0				
2	1	0	1	1				
3	1	1	0	0				

Qn	Qn+1	R	S
0	0	x	0
0	1	0	1
1	0	1	0
1	1	0	x

1- April - 10

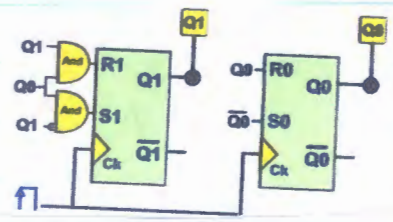


$P1 = Q1Q0$

$S1 = \overline{Q1}Q0$

$P0 = Q0$

$P0 = \overline{Q0}$



# Comparacion

FF T	T1=00	T0=1		
FF D	D1=Q1Q0	D0=Q0		
FF JK	J1=Q0	K1=Q0	J0=1	K0=1
FF RS	R1=Q1Q0	S1=Q1Q0	R0=Q0	S0=Q0



1-Abril-10

## Código en: Abel-Hdl

1. Diagrama de Bloques Entradas y Salidas
2. Sincronización
3. Asignación de valores a los estados
4. Diagrama de transición (state-Diagram, Truth table)
5. Simulación (test-vectors)

## 1. Diagrama de Bloques Entradas y Salidas

Código en Abel-Hdl

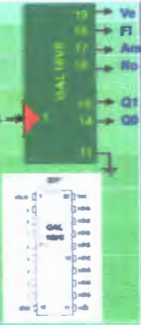
1.- Diagrama de Bloques Entradas y Salidas

Modulo am

"Entrada  
Ck pin 1;

"Salidas Combinacionales  
Ve, Fl, Am, Ro pin 19..16 lotype 'com';

"Salidas Registradas  
Q1, Q0 pin 15, 14 lotype 'reg';



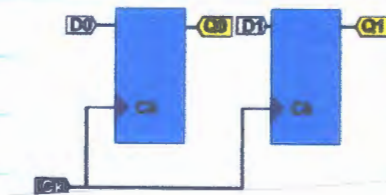
## 2 Sincronización

"sincronización de las FlipFlops

Sinc = [01, 00];

Equations

Sinc.clk = ck;



1-Abril-10

### 3. Asignación de valores a los estados

"Asignación de valores a los estados

declarations

Verde = [0,0];

Flecha = [0,1];

Ambar = [1,0];

Rojo = [1,1];

Estados	Asignación de valores a los estados	
	Q1	Q0
Verde	0	0
Flecha	0	1
Ambar	1	0
Rojo	1	1

### 4. Diagrama de Transición

"Diagrama de transición

State Diagram Sin:

STATE Verde:

Verde: Flecha: Ambar: Rojo:

gato Flecha:

STATE Flecha:

Verde: Flecha: Ambar: Rojo:

gato Ambar:

STATE Ambar:

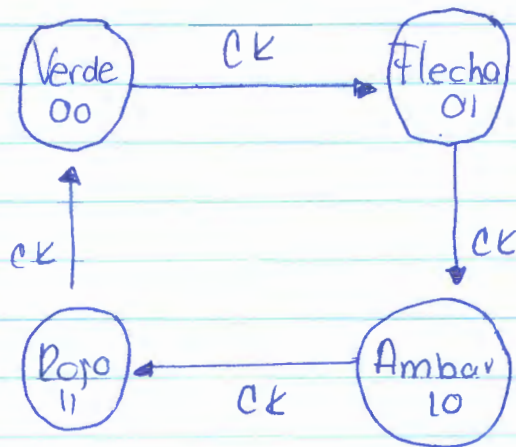
Verde: Flecha: Ambar: Rojo:

gato Rojo:

STATE Rojo:

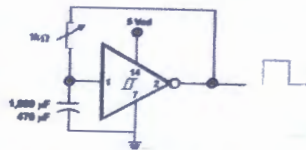
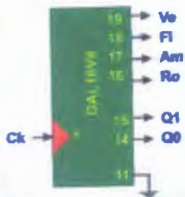
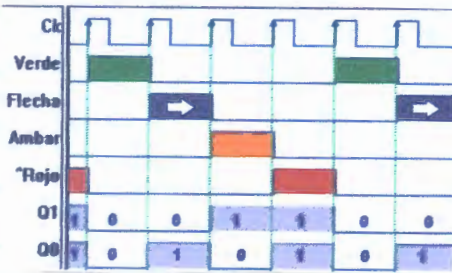
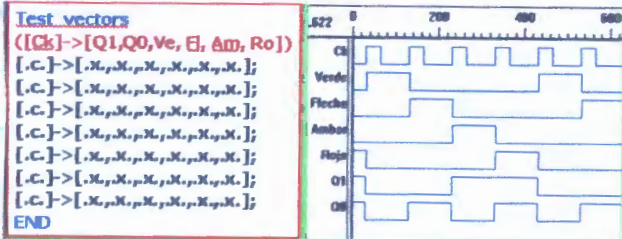
Verde: Flecha: Ambar: Rojo:

gato Verde:

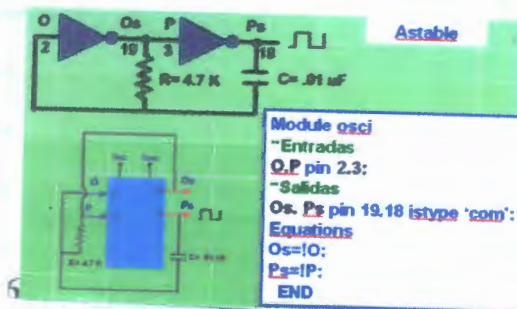


1-Abril-10

## 5. Simulación



## Multibrador Astable



1-Abril-10

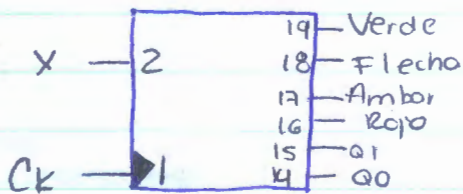
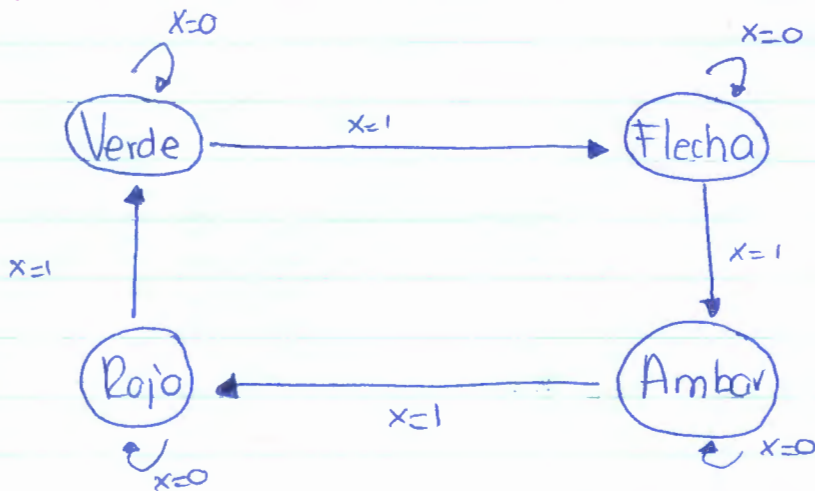
## Ejemplo 2

Diseñe un sistema Secuencial sincrono que represente la operacion de un semaforo de cuatro estados que se presentan en el siguiente orden: Verde, Flecha, Ambar, y Rojo, ademas incluya una Entrada  $x$  de modo que:

Si  $x=0$ , el sistema debera permanecer en el mismo estado

Si  $x=1$ , el sistema debera de cambiar al estado siguiente.

### Diagrama de Transición



### Asignar los valores a los estados

Estados	Asignacion de los valores a los estados	
	Q1	Q0
Verde	0	0
Flecha	0	1
Ambar	1	0
Rojo	1	1

1-Abril-10

## Tabla de Estados

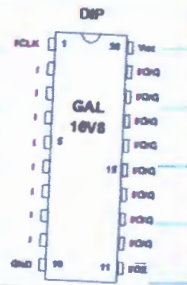
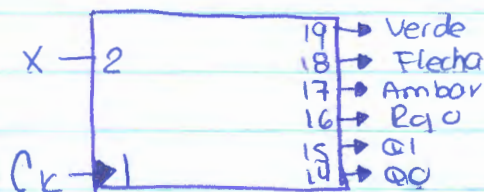
Estado Presente	Estado Proximo	
	x=0	x=1
Verde	Verde	Flecha
Flecha	Flecha	Ambor
Ambor	Ambor	Rojos
Rojos	Rojos	Verde

## Codigo en Abel-Hdl

### 1. Diagrama de Bloques Entradas y Salidas

```

Module sem
  *Entrada
  Ck, X pin 1,2;
  *Salidas Combinacionales
  Ve, Fl, Am, Ro pin 19..16 istype 'com';
  *Salidas Registradas
  Q1, Q0 pin 15,14 istype 'reg';
  
```



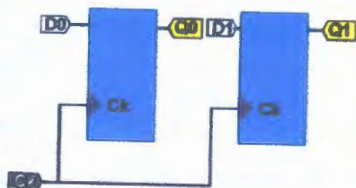
## Sincronización

"Sincronización de los Flip Flops

Sinc = [Q1, Q0];

Equations

Sinc.clk = Ck;



1-Abril-10

## Asignación de valores a los estados

"Asignación de valores a los estados  
declarations

Verde = [0, 0];

Flecha = [0, 1];

Ambar = [1, 0];

Rojo = [1, 1];

Estados	Asignación de valores a los estados	
	Q1	Q0
Verde	0	0
Flecha	0	1
Ambar	1	0
Rojo	1	1

## Diagrama de Transición

\*Diagrama de transición

State Diagram Sinc

STATE Verde:

Ve=1; El=0; Am=0; Ro=0;

If X then Flecha else Verde;

STATE Flecha:

Ve=0; El=1; Am=0; Ro=0;

If X then Ambar;

! X then Flecha;

STATE Ambar:

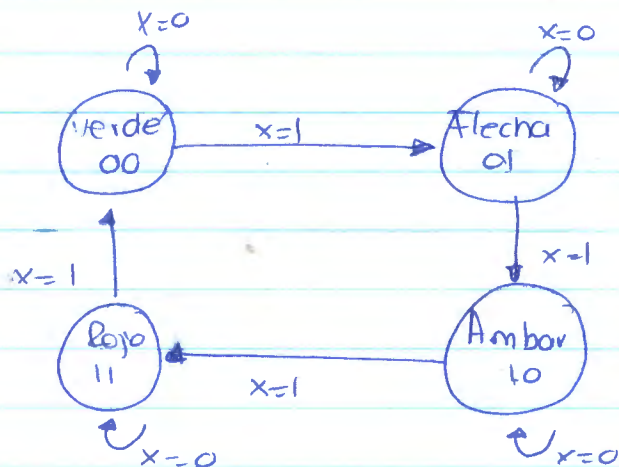
Ve=0; El=0; Am=1; Ro=0;

If X then Rojo else Ambar;

STATE Rojo:

Ve=0; El=0; Am=0; Ro=1;

If X then Verde else Rojo;



## Simulación

Test vectors

((Ck,X) -> [Q1,Q0,Ve, El, Am, Ro])

[.C.,0] -> [.X.,.X.,.X.,.X.,.X.,.X.];

[.C.,1] -> [.X.,.X.,.X.,.X.,.X.,.X.];

[.C.,0] -> [.X.,.X.,.X.,.X.,.X.,.X.];

[.C.,1] -> [.X.,.X.,.X.,.X.,.X.,.X.];

[.C.,0] -> [.X.,.X.,.X.,.X.,.X.,.X.];

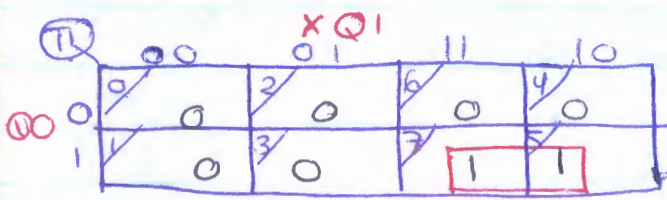
[.C.,1] -> [.X.,.X.,.X.,.X.,.X.,.X.];

[.C.,0] -> [.X.,.X.,.X.,.X.,.X.,.X.];

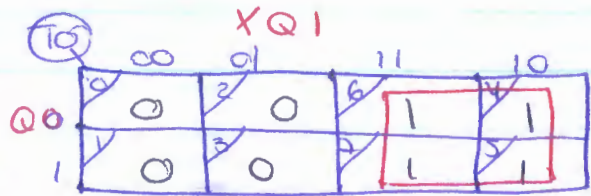
END

1-Abril-10

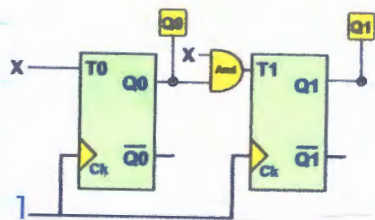
Entrada	Estado Presente		Estado Próximo		Entradas de Control		Salidas				
	X	Q1	Q0	Q1+1	Q0+1	T1	T0	V	F	A	R
0	0	0	0	0	0	0	0	1	0	0	0
0	0	1	0	1	0	0	0	0	1	0	0
0	1	0	1	0	0	0	0	0	0	1	0
0	1	1	1	1	0	0	0	0	0	0	1
1	0	0	0	1	0	1	1	1	0	0	0
1	0	1	1	0	1	1	1	0	1	0	0
1	1	0	1	1	0	1	1	0	0	1	0
1	1	1	0	0	1	1	1	0	0	0	1



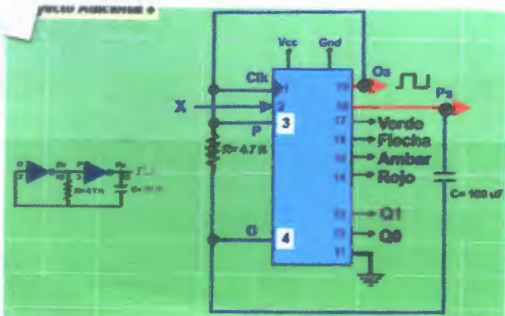
$T1 = X'Q0$



$T0 = X'$



Proyecto adicional 8



Fcs

1-Abril-10

## Metodología del diseño secuencial

1. Especificar el sistema (Diagrama de transición).
2. Determinar la cantidad de Flip Flops
3. Asignar los valores a los estados
4. Determinar las entradas y salidas
5. Construir una Tabla de Estados

## Programación

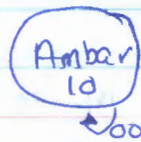
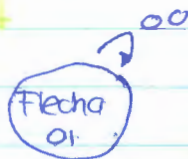
1. Entradas y Salidas
2. Sincronización
3. Asignar valores a los estados
4. Describir la secuencia por estados
5. Simulación Test-Vectors

## Ejemplo 3 Proyecto Adicional 8 P/ Viernes 7 Mayo.

Diseñe un sistema secuencial síncrono que represente la operación de un semáforo de cuatro estados que se presentan en el siguiente orden: Verde, Flecha, Ambar y Rojo, además incluya dos entradas  $x$ ,  $y$  de modo que:

$x$	$y$	Acción
0	0	Permanecer en el mismo estado
0	1	Cambiar al estado siguiente
1	0	Avanzar dos estados
1	1	Retroceder un estado

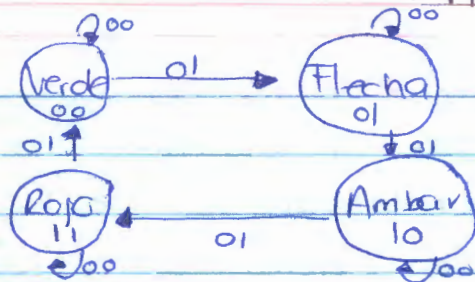
Permanecer en el mismo estado 00



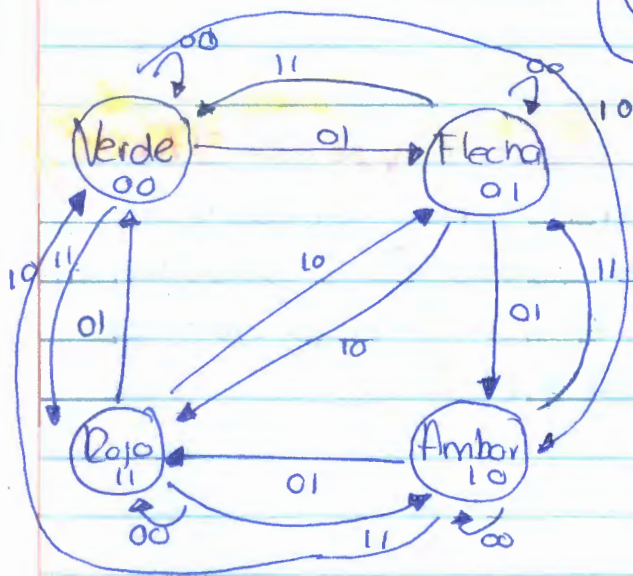
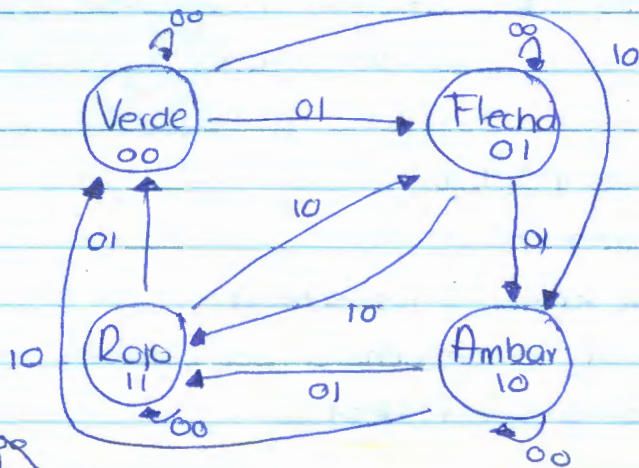


1-Abril-10

Cambiar al estado siguiente 01

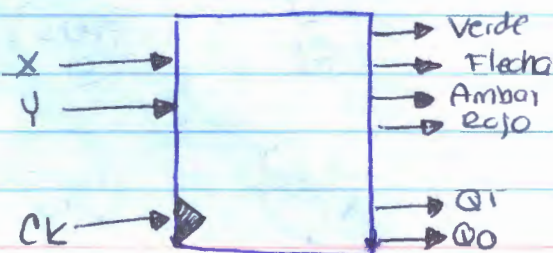


Avanzar dos estados 10



Retroceder un estado 11

Entradas y Salidas



1-Abril-10

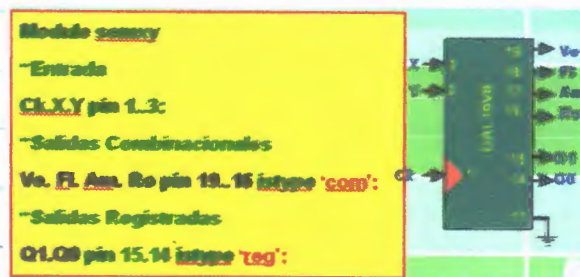
## Tabla de Estados

Estado Presente	Estado Proximo			
	00	01	10	11
Verde	Verde	Flecha	Ambar	Rojo
Flecha	Flecha	Ambar	Rojo	Verde
Ambar	Ambar	Rojo	Verde	Flecha
Rojo	Rojo	Verde	Flecha	Ambar

Estado Presente	Estado Proximo			
	00	01	10	11
E0 Verde	E0	E1	E2	E3
E1 Flecha	Flecha	Ambar	Rojo	Verde
E2 Ambar	Ambar	Rojo	Verde	Flecha
E3 Rojo	Rojo	Verde	Flecha	Ambar

Código en Abel-Hdl

## 1. Diagrama de Bloques Entradas y Salidas



1-Abril-10

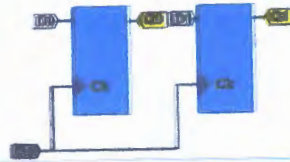
## Sincronización

"Sincronización de los Flip Flops

$$FIME = [Q1, Q0];$$

Equations

$$FIME \cdot clk = Ck;$$



## Asignación de valores a los estados

"Asignación de valores a los estados

$$E0 = [0, 0];$$

$$E1 = [0, 1];$$

$$E2 = [1, 0];$$

$$E3 = [1, 1];$$

Estados	Asignación de valores a los estados	
	Q1	Q0
E0	0	0
E1	0	1
E2	1	0
E3	1	1

Estado Presente		Estado Proximo			
		00	01	10	11
E0	Verde	E0	E1	E2	E3
E1	Flecha	E1	E2	E3	E0
E2	Ambos	E2	E3	E0	E1
E3	Rep	E3	E0	E1	E2

1-Abril-10

State Diagram FINE

State E0:

Y=0; F1=0; A=0; B=0;

if IX & IV then E0;

if IX & Y then E1;

if X & IV then E2;

if X & Y then E3;

Estado Presente	Estado Proximo			
	00	01	10	11
E0 Verde	E0	E1	E2	E3
E1 Flaco	E1	E2	E3	E0
E2 Ancho	E2	E3	E0	E1
E3 Rojo	E3	E0	E1	E2

State E1:

Y=0; F1=1; A=0; B=0;

if IX & IV then E1;

if IX & Y then E2;

if X & IV then E3;

if X & Y then E0;

Estado Presente	Estado Proximo			
	00	01	10	11
E0 Verde	E0	E1	E2	E3
E1 Flaco	E1	E2	E3	E0
E2 Ancho	E2	E3	E0	E1
E3 Rojo	E3	E0	E1	E2

State E2:

Y=0; F1=0; A=1; B=0;

if IX & IV then E2;

if IX & Y then E3;

if X & IV then E0;

if X & Y then E1;

Estado Presente	Estado Proximo			
	00	01	10	11
E0 Verde	E0	E1	E2	E3
E1 Flaco	E1	E2	E3	E0
E2 Ancho	E2	E3	E0	E1
E3 Rojo	E3	E0	E1	E2

State E3:

Y=0; F1=0; A=0; B=1;

if IX & IV then E3;

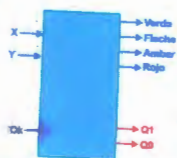
if IX & Y then E0;

if X & IV then E1;

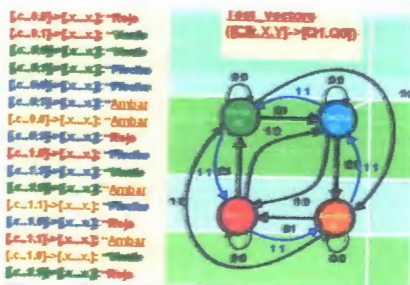
if X & Y then E2;

Estado Presente	Estado Proximo			
	00	01	10	11
E0 Verde	E0	E1	E2	E3
E1 Flaco	E1	E2	E3	E0
E2 Ancho	E2	E3	E0	E1
E3 Rojo	E3	E0	E1	E2

Simulación

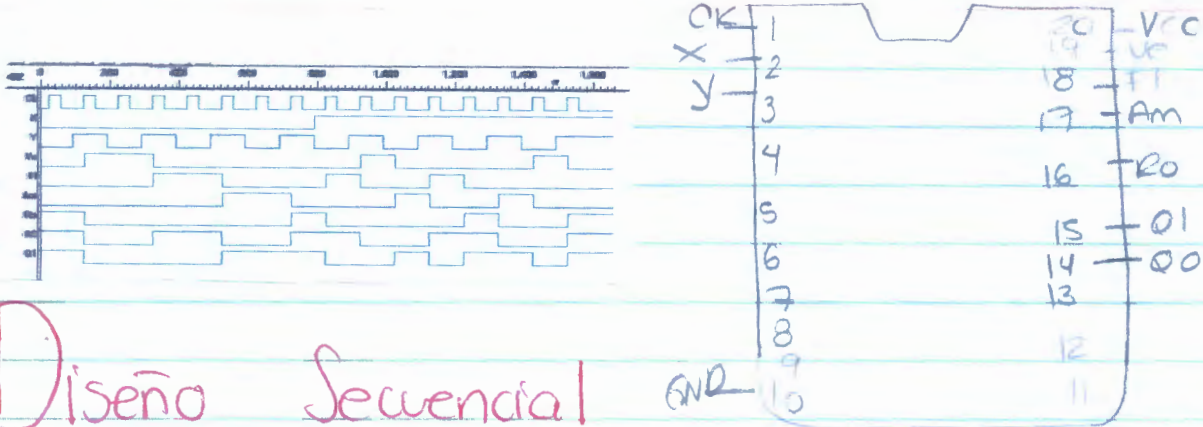


1. Entradas y Salidas
2. Sincronización
3. Asignar valores a los estados
4. Describir la secuencia por estados
5. Test-vectors



Test-Vectors  
(CLK, X, Y) -> (Q1, Q0)

## I-Abri-10

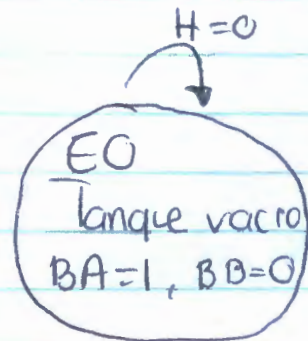
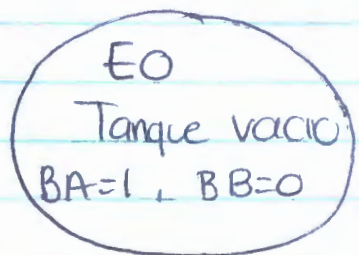


## Diseño Secuencial

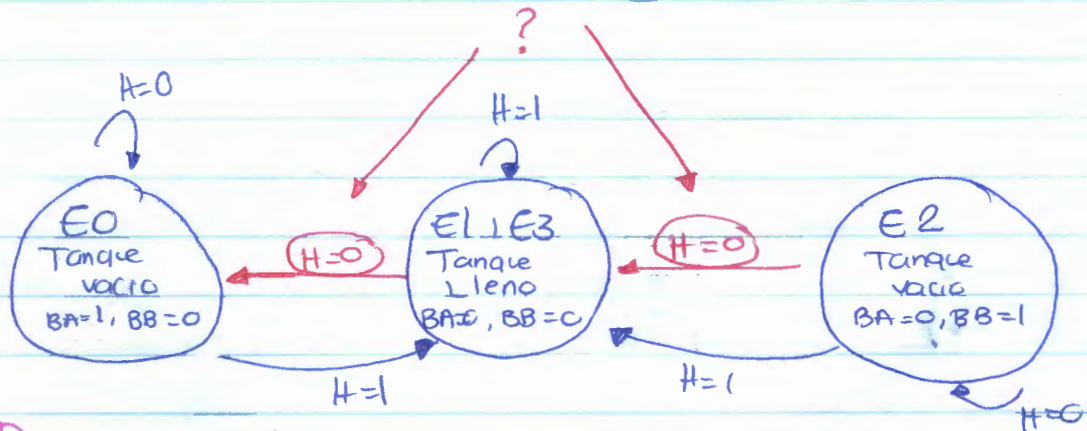
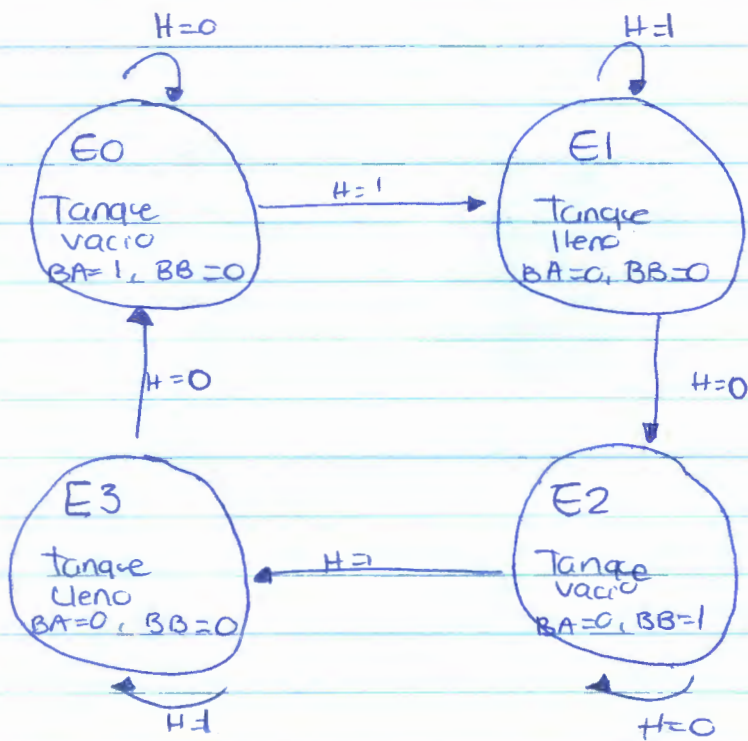
Diseñe un sistema secuencial que controle el llenado de un Tanque con las siguientes características

- El sistema consta de dos bombas llamadas 'A' y 'B'.
- Un sensor de nivel 'H';  $H=1$  Tanque lleno y  $H=0$  Tanque vacío.
- Partiendo de que el Tanque se encuentre vacío ( $H=0$ ), el llenado deberá iniciarse encendiendo la bomba "A" hasta llenar el tanque ( $H=1$ ), y entonces se apaga.
- Si de nuevo se vacía el Tanque ( $H=0$ ), el llenado deberá hacerse encendiendo la bomba "B", hasta llenar el Tanque ( $H=1$ ) y entonces se apaga.
- Si se vacía de nuevo el Tanque, el llenado deberá hacerse con la bomba "A" y así sucesivamente con la finalidad de que las bombas se alternen en su funcionamiento.

## Diagrama de Transición



i-Abrii-10



Determinar la cantidad de Flip Flops

Estados	Cantidad de Flip Flops
2	1
3 o 4	2
5 a 8	3

1-Abril-10

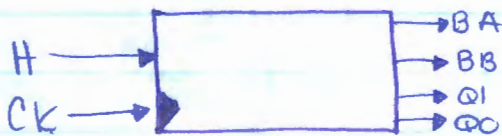
Asignar los valores a los estados

Estados	Asignación de valores a los estados	
	Q1	Q0
E0	0	0
E1	0	1
E2	1	0
E3	1	1

Tabla de estados

Estado Presente	Estado Proximo	
	H=0	H=1
E0	E0	E1
E1	E2	E1
E2	E2	E3
E3	E0	E3

Determinar las entradas y Salidas



Pasos en la programación

1. Entradas y Salidas
2. Sincronización
3. State-diagram o Truth-table
4. Simulación

Archivo EN ABEL-HDL

MODULE DBOMBAS

"Entradas

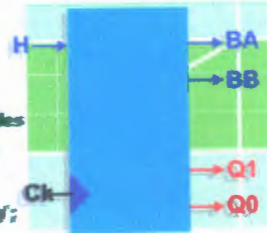
Clk, H Pin 1,2;

" Salidas Combinacionales

BA, BB Pin 19,18 in type 'com';

" Salidas Registradas

Q1, Q0 pin 17,16 in type 'reg';



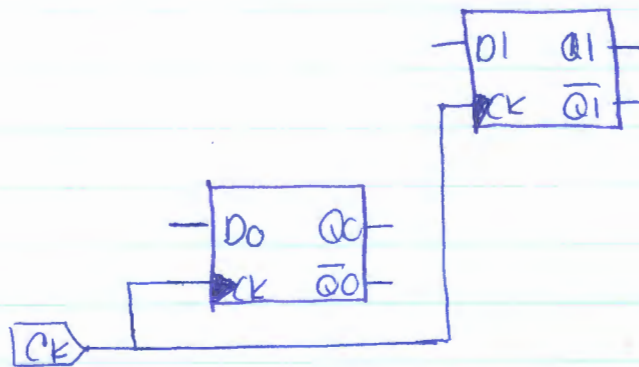
" Conectar el CLK a los dos Flip Flops sincronizar

DECLARATIONS

Sreg = [Q0, Q1];

EQUATIONS

Sreg.clk = CLK



" Asignar valores a los estados

DECLARATIONS

E0 = [0, 0];

E1 = [0, 1];

E2 = [1, 1];

E3 = [1, 0];

Estados	Asignacion de valores a los estados	
	Q1	Q0
E0	0	0
E1	0	1
E2	1	1
E3	1	0



1-Abril-10

Archivo en ABEL-NDL

```
state_diagram Seq;  
state E0:  
  BA=1; BB=0;  
  if N then E1 else E0;  
state E1:  
  BA=0; BB=0;  
  if N then E2 else E2;  
state E2:  
  BA=0; BB=1;  
  if N then E2 else E3;  
state E3:  
  BA=0; BB=0;  
  if N then E3 else E0;
```

Estado Presente	Estado Proximo	BA	BB
E0	E0	E1	
E1	E2	E1	
E2	E2	E3	
E3	E0	E3	

Archivo en ABEL-NDL

```
state_diagram Seq;  
state E0:  
  BA=1; BB=0;  
  if N then E1 else E0;  
state E1:  
  BA=0; BB=0;  
  if N then E1 else E2;  
state E2:  
  BA=0; BB=1;  
  if N then E2 else E3;  
state E3:  
  BA=0; BB=0;  
  if N then E3 else E0;
```

Estado Presente	Estado Proximo	BA	BB
E0	E0	E1	
E1	E2	E1	
E2	E2	E3	
E3	E0	E3	

Programación con Truth-table en modo Secuencial: >

Truth-table

([Entrada, Estado presente] :> [Estado proximo])

1-Abril-10

MODULE dm

CLK pin 12;

BA, BB pin 19, 18 (output 'com');

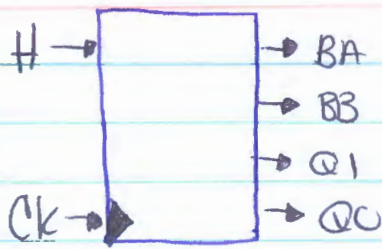
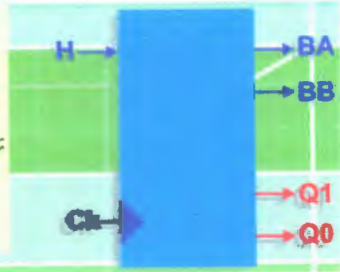
Q1, Q0 pin 17, 16 (TYPE 'REG');

-Sincronización

U1A1 = Q1, Q0;

equations

U1A1 CLK = CLK;



Truth table

(H, Q1, Q0) -> (Q1, Q0)

(0, 0) -> (0, 0)

(0, 1) -> (1, 0)

(1, 0) -> (0, 1)

(1, 1) -> (1, 1)

(1, 0) -> (0, 1)

(1, 1) -> (0, 0)

(1, 0) -> (1, 1)

(1, 1) -> (1, 0)



Estados	Q1	Q0
E0	0	0
E1	0	1
E2	1	0
E3	1	1

Truth table

(Q1, Q0) -> (BA, BB)

(0, 0) -> (1, 0)

(0, 1) -> (0, 0)

(1, 0) -> (0, 1)

(1, 1) -> (0, 0)

Estados	Q1	Q0	BB	BA
E0	0	0	1	0
E1	0	1	0	0
E2	1	0	0	1
E3	1	1	0	0

test\_vectors

(CLK, H) -> (BA, BB)

(1, 0) -> (1, 0)

(1, 0) -> (1, 0)

(1, 0) -> (1, 0)

(1, 0) -> (1, 0)

(1, 0) -> (1, 0)

(1, 0) -> (1, 0)

(1, 0) -> (1, 0)

(1, 0) -> (1, 0)

(1, 0) -> (1, 0)

(1, 0) -> (1, 0)

(1, 0) -> (1, 0)

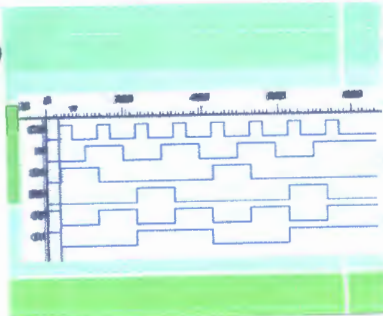
(1, 0) -> (1, 0)

(1, 0) -> (1, 0)

(1, 0) -> (1, 0)

(1, 0) -> (1, 0)

(1, 0) -> (1, 0)



Diseñe un Sistema secuencial usando la Máquina de Moore que controle el llenado de un Tanque con los siguientes características. El sistema consta de tres bombas llamadas "A", "B" y "C", y un sensor de nivel "H" que indica con  $H=1$  Tanque lleno y con  $H=0$  Tanque vacío, el sistema deberá de trabajar bajo la siguiente secuencia

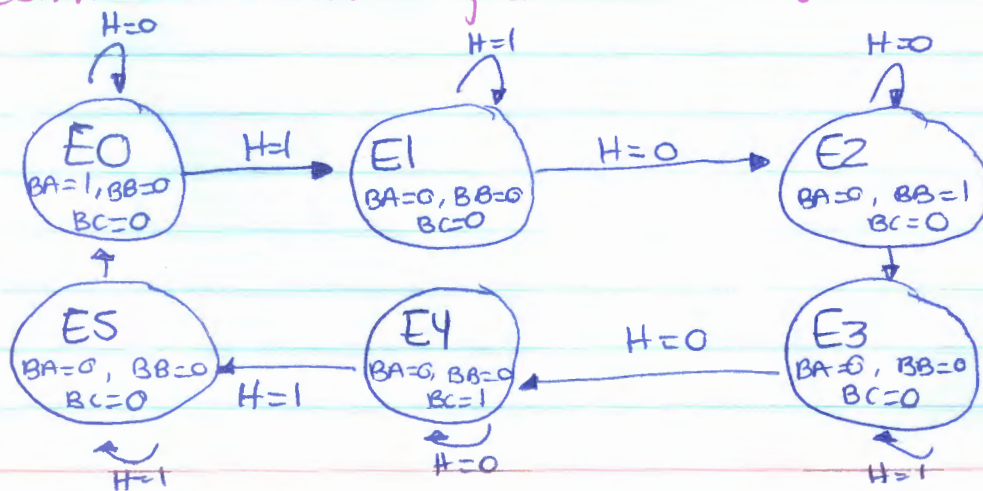
a) Partiendo de que el Tanque se encuentra vacío, el llenado deberá iniciarse con la bomba "A" hasta llenar el tanque ( $H=1$ ) y entonces desconectarla.

b) Si de nuevo se vacía el Tanque ( $H=0$ ), el llenado deberá hacerse con la bomba "B" hasta llenar el Tanque y entonces desconectarla.

c) Si de nuevo se vacía el Tanque ( $H=0$ ), el llenado deberá hacerse con la bomba "C" hasta llenar el Tanque y entonces desconectarla.

d) Si de nuevo se vacía el Tanque ( $H=0$ ), el llenado deberá hacerse con la bomba "A" y así sucesivamente con la finalidad de que las tres bombas se alternen en su funcionamiento.

### Como sería el diagrama de Transición

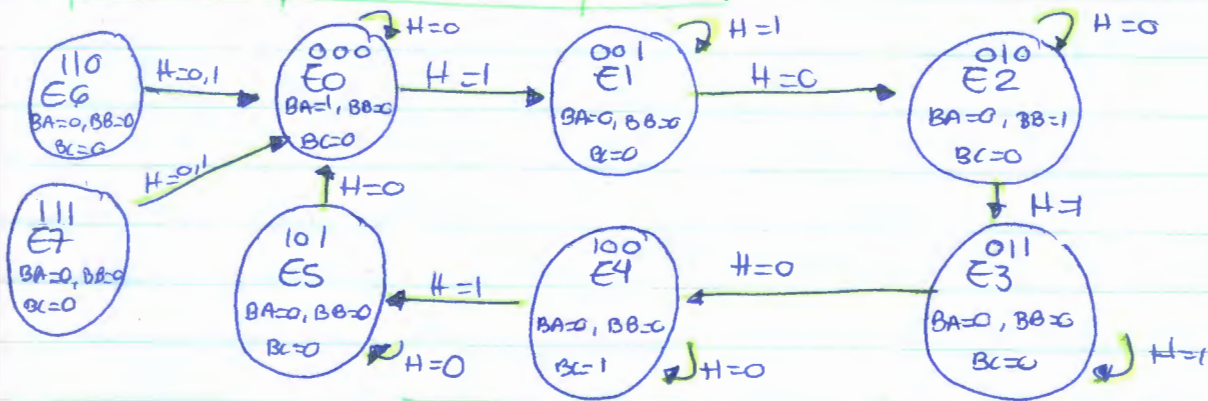


1-Abril-10

### 3 Flip Flops

	Q2	Q1	Q0
E0	0	0	0
E1	0	0	1
E2	0	1	0
E3	0	1	1
E4	1	0	0
E5	1	0	1
E6	1	1	0
E7	1	1	1

Que hacer con E6 y E7?



Estado Presente	Estado Proximo		Salidas		
	H=0	H=1	BA	BB	BC
E0	E0	E1	1	0	0
E1	E2	E1	0	0	0
E2	E2	E3	0	1	0
E3	E4	E3	0	0	0
E4	E4	E5	0	0	1
E5	E0	E5	0	0	0
E6	E0	E0	0	0	0
E7	E0	E0	0	0	0

1-Abril-10



E	Estado presente		Salidas		
	no	si	BA	BB	BC
E0	E0	E1	1	0	0
E1	E2	E3	0	0	0
E2	E2	E3	0	1	0
E3	E4	E5	0	0	0
E4	E4	E5	0	0	1
E5	E5	E5	0	0	0
E6	E6	E6	0	0	0
E7	E6	E6	0	0	0

# Teorema Fundamental

## Detector de nivel

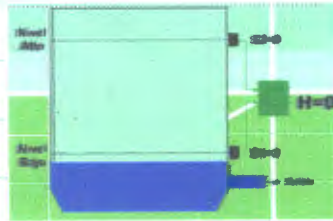
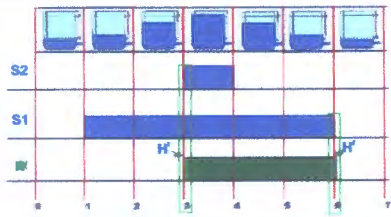
Diseñe un sistema secuencial **asíncrono** para detectar nivel de un tanque que cuenta con dos sensores llamados S1 (Nivel Bajo) S2 (Nivel Alto)

S2 (Nivel Alto)

que contenga una salida H de modo que

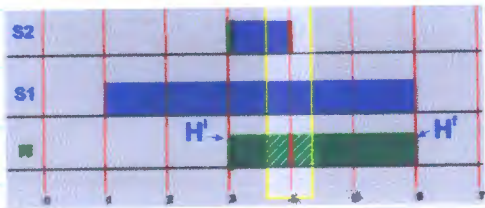
**H=0** Cuando el nivel va de S1 hacia S2 (subida) hasta que llegue a S2.

$H=1$  Cuando el nivel va de  $S_2$  hacia  $S_1$  (bajada) hasta que llegue a  $S_1$  como lo indica el siguiente diagrama de Tiempos.



De que depende que  $H=1$  o  $H^i$  de  $S_2=1$   
 $H^i = S_2$

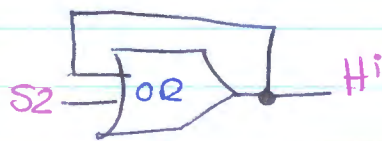
Pero cuando  $S_2=0$ ,  $H=1$ , como lo mantenemos



Obtencion de las ecuaciones

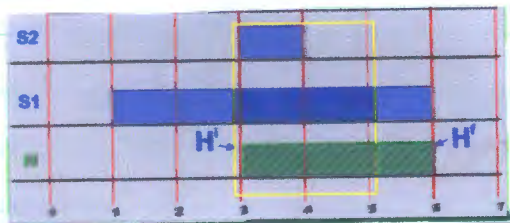
$H^i = S_2$  Pero cuando  $S_2=0$ ,  $H=1$

Para sostener el valor de  $H=1$  usaremos una operacion OR con retroalimentación



$$H^i = S_2 + H^i$$

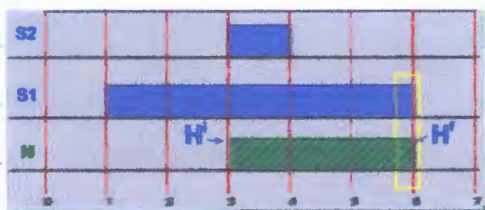
$$H^i = S_2 + H^i$$



1-Abril-10

De que depende que  $H=0$  a  $H=1$  de  $S1=0$

$$H^f = (S1)'$$

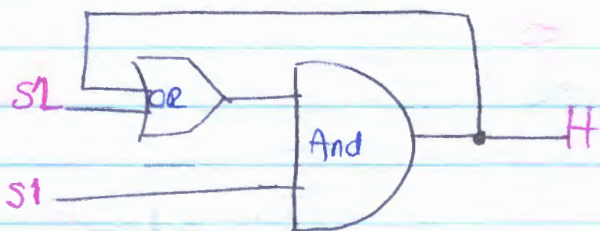


Aplicando el Teorema fundamental

$$H = H^i (H^f)'$$

Obtenemos la ecuación para H

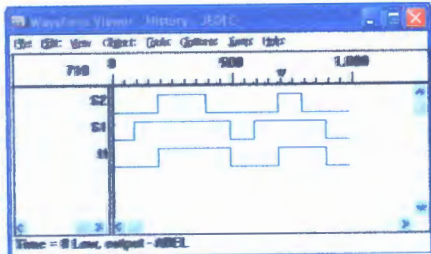
$$H = (S2 + H) S1$$



Archivo en ABEL - HOL

```
MODULE nivel
  "Entradas
  S1, S2 pin 1,2 ;
  "Salida
  # pin 19 istype 'com';
Equations
  H = (S2 # H) & S1;
TEST_VECTORS
((S1 S2) -> (H))
(0,0) -> (0)
(1,0) -> (0)
(1,1) -> (1)
(0,1) -> (0)
(1,0) -> (0)
(1,1) -> (1)
(0,0) -> (0)
(1,0) -> (0)
(1,1) -> (1)
(0,1) -> (0)
(1,0) -> (0)
(1,1) -> (1)
(0,0) -> (0)
END
```

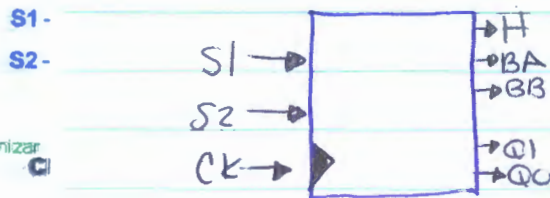
# Simulación



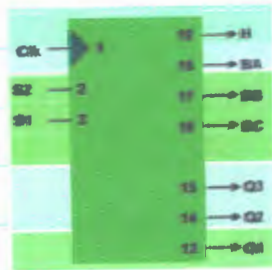
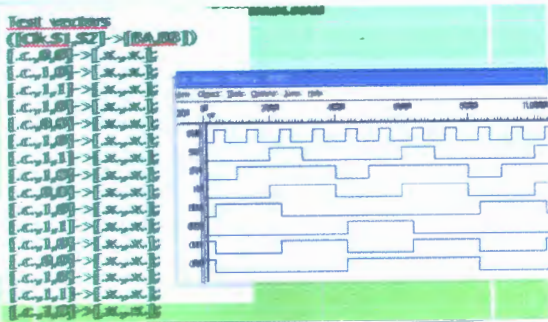
Si agregamos el detector de nivel

```

MODULE DBOMBAS
  "Entradas
  Clk, S1, S2 Pin 1..3;
  " Salidas Combinacionales
  BA, BB, H Pin 19..17 istype'com';
  " Salidas Registradas
  Q1, Q0 pin 16, 15 istype'reg';
  " Conectar el Clk a los dos Flip Flops sincronizar
  DECLARATIONS
  Sreg=[Q0, Q1];
  EQUATIONS
  Sreg.clk=Clk;
  H = (S2 # H) & S1;
  
```



# Simulación



sec. 2



1-Abril-10

## Control de Señal de alerta.

Diseñe un sistema secuencial usando la maquina de Moore que sea el control de una señal de alerta.

Dicha señal consta de 3 luces en forma de triángulo llamadas L0, L1 y L2.

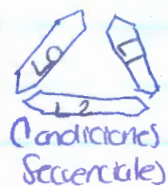
Se requieren de dos diferentes secuencias que son seleccionadas por medio un boton 'S' de modo que:

a) Ambas parten de condiciones iniciales en donde todas las lamparas están apagadas

b) Si  $S=0$  Ocurrira la secuencia Uno A, B, C, A, B, C repetidamente

d) Si  $S=1$  ocurrira la secuencia Dos x, y, z, x, y, z repetidamente

**Nota:** Cuando usted cambia el selector S en medio de una secuencia, las luces continuaran con la secuencia actual hasta terminar (C o z) y pasara a encender todas las luces y posteriormente a condiciones iniciales, de ahí en adelante, continuara con la secuencia que fue seleccionada, correspondiendo al nuevo valor de S.



Secuencia Dos



Cambio de Secuencia

1-Abril-10

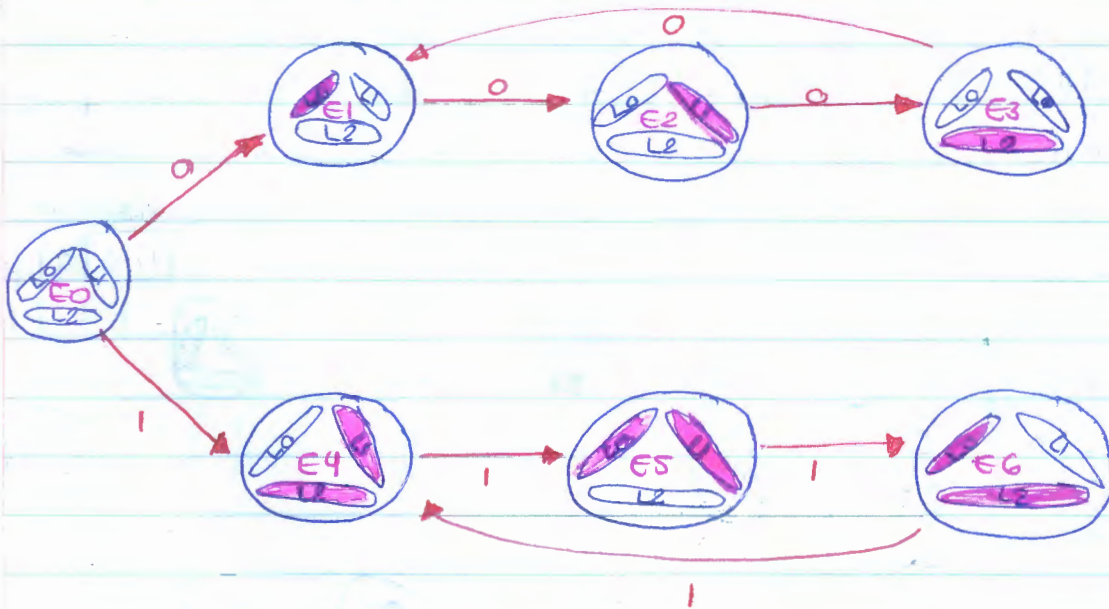
Por ejemplo

Si  $s=0$  y un selector  $S$  se hace 1, en el momento que la secuencia de luces es  $(L_0=0, L_1=1, \text{ y } L_2=0)$ , correspondiente a la condición B de la secuencia Uno, continuara hasta C y de ahí a todas encendidas  $(L_0=1, L_1=1, L_2=1)$  y al siguiente pulso de reloj a condiciones iniciales  $(L_0=0, L_1=0, \text{ y } L_2=0)$  y de ahí en adelante continuara con la secuencia Dos.

a) Ambas parten de condiciones iniciales en donde todas las lamparas están apagadas



b) Si  $s=1$  ocurrira la secuencia Dos (x, y, z, x, y, z, etc.)



1-Abril-10

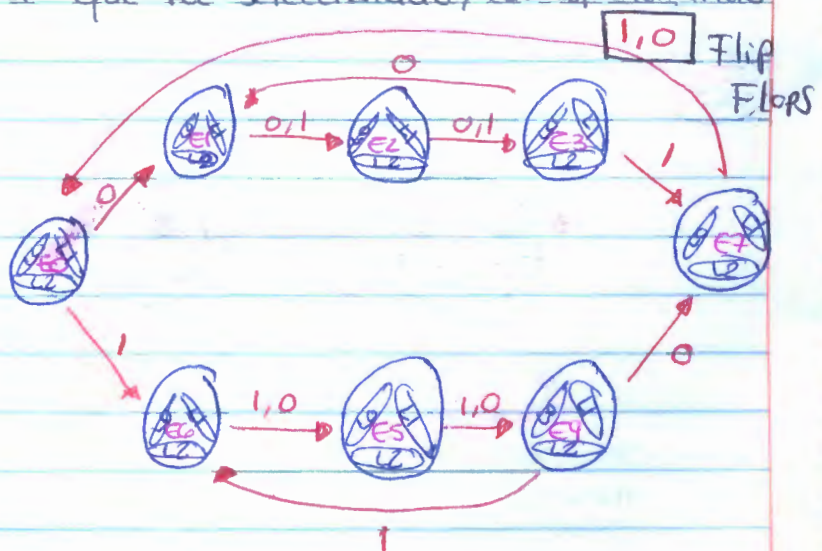
### Tabla de estados

	S=0	S=1
E0	E1	E4
E1	E2	
E2	E3	
E3	E1	
E4		E5
E5		E6
E6		E4
E7		

Cuando usted cambia el selector S en medio de una secuencia, las luces continuaron con la secuencia actual hasta terminar (0 o 2) y pasara a encender todas las luces y posteriormente a condiciones iniciales de ahí en adelante, continuara con la secuencia que fue seleccionada, correspondiendo al nuevo valor de S.

### Tabla de estados

	S=0	S=1
E0	E1	E4
E1	E2	E2
E2	E3	E3
E3	E1	E7
E4	E5	E5
E5	E6	E6
E6	E7	E4
E7	E0	E0

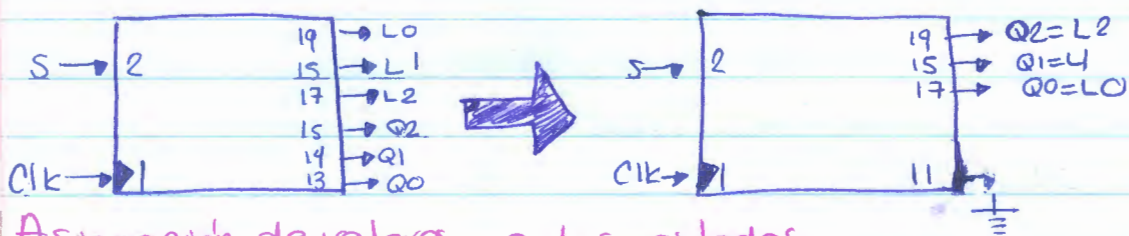


1-Abril-10

### Cuántas Flip Flops

Estados	Cantidad de Flip Flops
2	1
3 o 4	2
5 a 8	3
9 a 16	4

### Cuántas Salidas Combinacionales

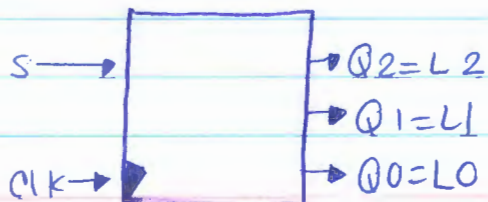


### Asignación de valores a los estados

	L2	L1	L0	Nciol
E0	0	0	0	0
E1	0	0	1	1
E2	0	1	0	2
E3	1	0	0	4
E4	1	1	0	6
E5	0	1	1	3
E6	1	0	1	5
E7	1	1	1	7

### Entradas y Salidas

El sistema solo necesitaría 3 salidas



1-Abril-10

## Archivo ABEL-HDL

MODULE alerta

"Entrada

CLK, S Pin 1,2;

"Salidas Registradas

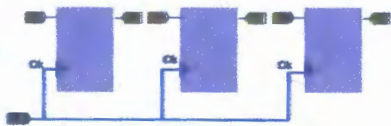
Q2..Q0 pin 19..17 istype 'reg';

"sincronización de los Flip Flops

Sinc = [Q2..Q0];

Equations

Sinc.CLK = CLK;



"Asignación de valores a los estados

Declarations

E0 = [0, 0, 0]

E1 = [0, 0, 1]

E2 = [0, 1, 0]

E3 = [1, 0, 0]

E4 = [1, 1, 0]

E5 = [0, 1, 1]

E6 = [1, 0, 1]

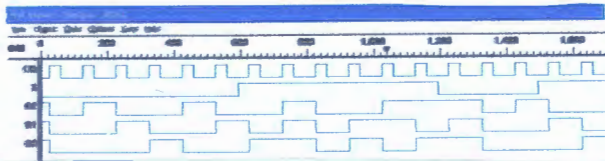
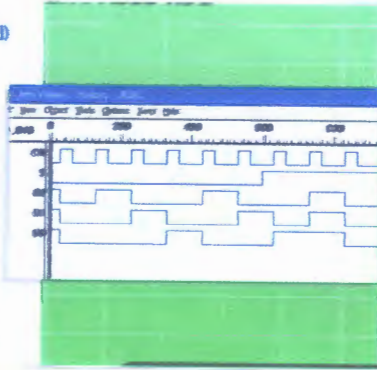
E7 = [1, 1, 1]

1-Abril-10

State diagram Sinc  
State E0:  
if S then E1 else E4;  
State E1:  
goto E2;  
State E2:  
goto E3;  
State E3:  
if S then E7 else E1;  
State E4:  
goto E5;  
State E5:  
goto E6;  
State E6:  
if S then E4 else E7;  
State E7:  
goto E0;

	S=0	S=1
E0	E1	E4
E1	E2	E2
E2	E3	E3
E3	E1	E7
E4	E5	E5
E5	E6	E6
E6	E7	E4
E7	E0	E0

Test vectors  
{(Clk,S)->[Q2,Q1,Q0]}  
[c.,0]->[x.,x.,x.];  
[c.,0]->[x.,x.,x.];  
[c.,0]->[x.,x.,x.];  
[c.,0]->[x.,x.,x.];  
[c.,0]->[x.,x.,x.];  
[c.,0]->[x.,x.,x.];  
[c.,1]->[x.,x.,x.];  
[c.,1]->[x.,x.,x.];  
[c.,1]->[x.,x.,x.];  
[c.,1]->[x.,x.,x.];  
[c.,1]->[x.,x.,x.];  
[c.,1]->[x.,x.,x.];  
[c.,0]->[x.,x.,x.];  
[c.,0]->[x.,x.,x.];  
[c.,1]->[x.,x.,x.];  
[c.,1]->[x.,x.,x.];  
END



Programación con Truth-table en modo secuencial: >

Truth-table

[C Entrada, Estado presente] :> [Estado proximo]

1-Abril-10

# Programación con Truth-table

No es necesaria

la asignación

Programación con Truth table

MODULE luces  
Entrada  
CB.S Pin 1.2:  
salidas Registradas  
O2\_00 pin 19..17  
Inhibe "seg":  
"sincronizacion"  
Sinc-[O2\_00]:  
Emisores  
Sinc.CB=CB:

Truth table  
(S.Sinc) > (Sinc)

	L2	L1	L0	Salida
00	0	0	0	0
01	0	0	1	1
02	0	1	0	2
03	1	0	0	4
04	1	1	0	6
05	0	1	1	3
06	1	0	1	5
07	1	1	1	7

	S=0	S=1
00	E1	E4
01	E2	E2
02	E3	E3
03	E4	E1
04	E5	E5
05	E6	E6
06	E7	E4
07	E0	E0

No es necesario  
La asignación

Programación con Truth table

MODULE luces  
Entrada  
CB.S Pin 1.2:  
salidas Registradas  
O2\_00 pin 19..17  
Inhibe "seg":  
"sincronizacion"  
Sinc-[O2\_00]:  
Emisores  
Sinc.CB=CB:

Truth table  
(S.O2.O1.O0) > (O2.O1.O0)

	L2	L1	L0	Salida
00	0	0	0	0
01	0	0	1	1
02	0	1	0	2
03	1	0	0	4
04	1	1	0	6
05	0	1	1	3
06	1	0	1	5
07	1	1	1	7

	S=0	S=1
00	E1	E4
01	E2	E2
02	E3	E3
03	E4	E1
04	E5	E5
05	E6	E6
06	E7	E4
07	E0	E0

No es necesario  
La asignación

## Secuencia de Luces

**OBJETIVO PARTICULAR** Durante el desarrollo de esta actividad, se obtendrá el diseño de un circuito con un display con LED's destellantes, dicha display tiene 4 LED's que encienden y apagan en una secuencia particular que dependa de una señal de control X.

## Especificaciones

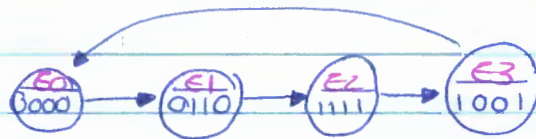
### Secuencia A

Si  $x=0$  Ocurrira la secuencia "A" [Los cuadros blancos indican que la luz esta apagada, los cuadros oscuros indican que la luz esta encendida]

1-Abril-10

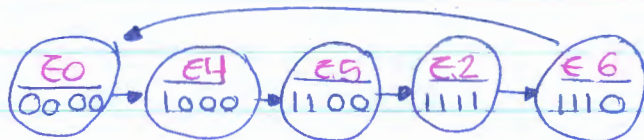
Secuencia 'A'  $x=0$

t	A	B	C	D
0	0	0	0	0
1	0	1	1	0
2	1	1	1	1
3	1	0	0	1
0	0	0	0	0



Secuencia 'B'  $x=1$

t	A	B	C	D
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	1
4	1	1	1	0
0	0	0	0	0



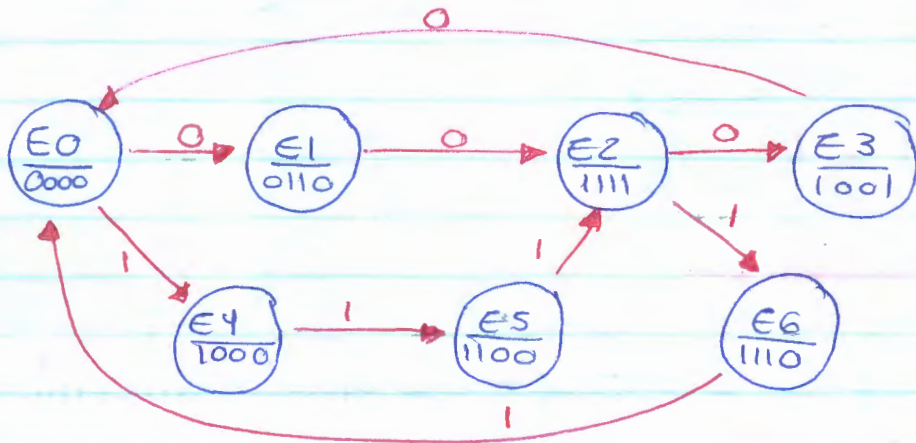
Cuántos eventos iguales tenemos en las dos secuencias

Secuencia A				
	A	B	C	D
E0	0	0	0	0
E1	0	1	1	0
E2	1	1	1	1
E3	1	0	0	1

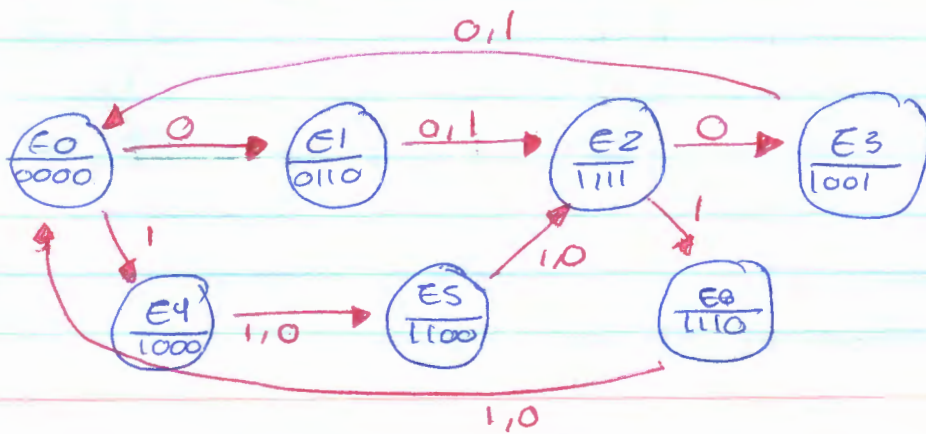
Secuencia B				
	A	B	C	D
E0	0	0	0	0
E4	1	0	0	0
E5	1	1	0	0
E2	1	1	1	1
E6	1	1	1	0



Cuando usted cambia el selector  $X$  en medio de una secuencia, las luces continuaran con la secuencia actual, hasta que se encuentre con un dígito de luces, que tambien este presente en la otra secuencia. De ahí en adelante, la secuencia que fue seleccionada comenzara nuevamente, correspondiendo al nuevo valor de  $X$

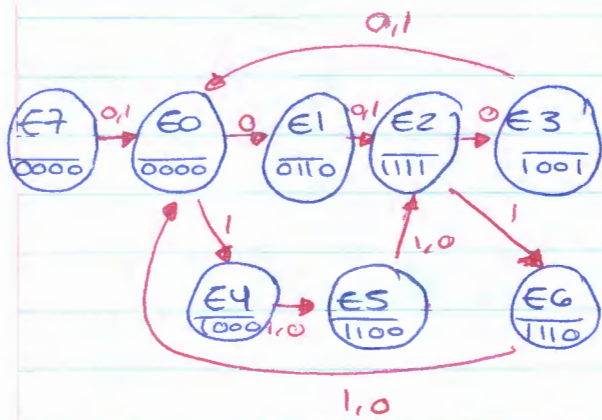


	$X=0$	$X=1$
E0	E1	E4
E1	E2	E2
E2	E3	E6
E3	E0	E0
E4	E5	E5
E5	E2	E2
E6	E0	E0



1-Abril-10

Que hacer con E7



	X=0	X=1
E0	E1	E4
E1	E2	E2
E2	E3	E6
E3	E0	E0
E4	E5	E5
E5	E2	E2
E6	E0	E0
E7	E0	E0

	X=0	X=1	A	B	C	D
E0	E1	E4	0	0	0	0
E1	E2	E2	0	1	1	0
E2	E3	E6	1	1	1	1
E3	E0	E0	1	0	0	1
E4	E5	E5	1	0	0	0
E5	E2	E2	1	1	0	0
E6	E0	E0	1	1	1	0
E7	E0	E0	0	0	0	0

MODULE luc

"Entradas

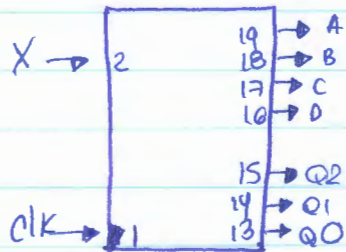
CLK, X pin 1, 2;

"Salidas Combinacionales

A, B, C, D pin 19..16 istype 'com';

"Salidas registradas

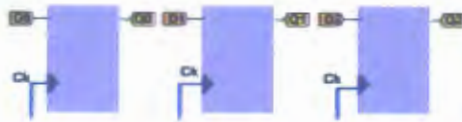
Q2..Q0 pin 15..13 istype 'reg';



$T = [02, 00];$

equations

$T.clk = clk;$



declarations

$E0 = [0, 0, 0];$

$E1 = [0, 0, 1];$

$E2 = [0, 1, 0];$

$E3 = [0, 1, 1];$

$E4 = [1, 0, 0];$

$E5 = [1, 0, 1];$

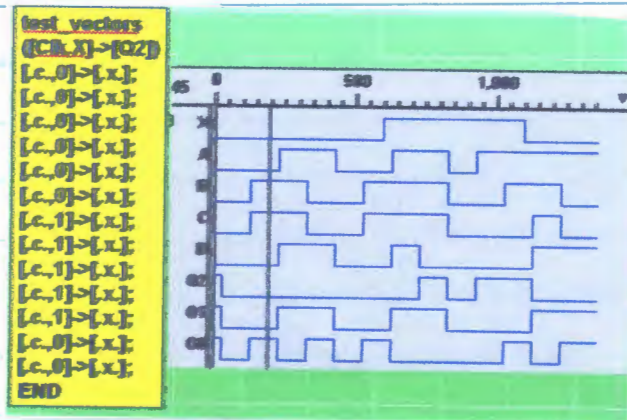
$E6 = [1, 1, 0];$

$E7 = [1, 1, 1];$

	Q2	Q1	Q0
E0	0	0	0
E1	0	0	1
E2	0	1	0
E3	0	1	1
E4	1	0	0
E5	1	0	1
E6	1	1	0
E7	1	1	1

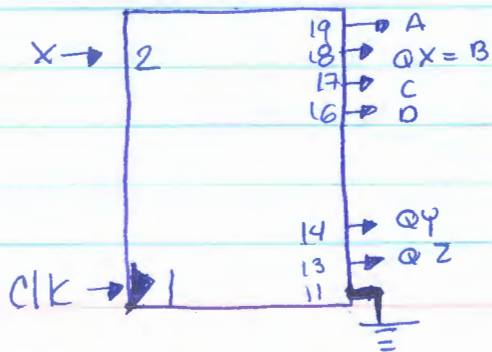


i-Abril-10



Proyecto Adicional 9 P/Lunes 10-Mayo-10

	x=0	x=1	A	B=Q	C	D
E0	E1	E4	0	0	0	0
E1	E2	E2	0	1	1	0
E2	E3	E6	1	1	1	1
E3	E0	E0	1	0	0	1
E4	E5	E5	1	0	0	0
E5	E2	E2	1	1	0	0
E6	E0	E0	1	1	1	0
E7	E0	E0	0	0	0	0



# Detector de Secuencia

2-Abril-10

## Luces traseras de un auto

Diseña un sistema secuencial que controle las luces traseras de un automovil.

Se tienen entradas que son:

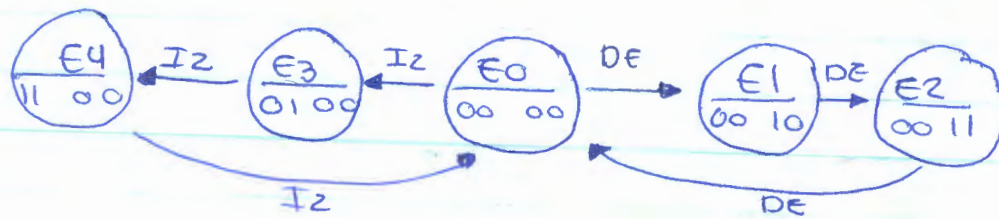
- Pedal de Freno F
- Direccional derecha De
- Direccional izquierda Iz

Ademas de cuatro juegos de luces de salida llamadas O11, O12, D01, D02

Al oprimir el freno F se deberan encender todas las luces no importa si estan activados las direccionales.

Las direccionales tienen tres estados que se repiten en los tiempos mientras la palanca este en esta en la posición.

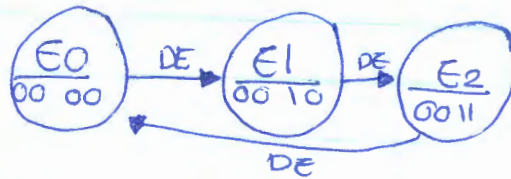
## Diagrama de Transición



Freno



## Diagrama de Transición



Freno



2-Abril-10

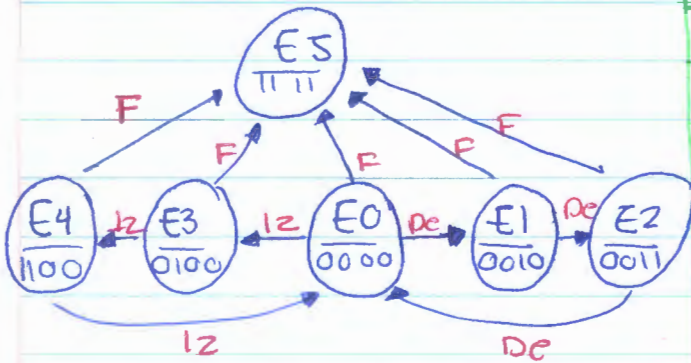


Tabla de estados

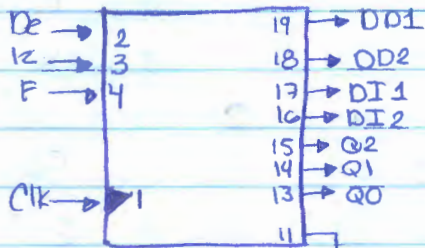
F	0	0	0	0	1
De	0	0	1	1	X
Iz	0	1	0	1	X
E0	E0	E3	E1	X	E5
E1	E0	E3	E2	X	E5
E2	E0	E3	E0	X	E5
E3	E0	E4	E1	X	E5
E4	E0	E0	E1	X	E5
E5	E0	E3	E1	X	E5

- Cual sería el estado siguiente si:  
F=0, De=0, Iz=0
- Cual sería el estado siguiente si:  
F=0, De=1, Iz=1
- Cual sería el estado siguiente si Estando en E1 las entradas son:  
F=0, De=0, Iz=1
- Cual sería el estado siguiente si Estando en E2 las entradas son:  
F=0, De=0, Iz=1
- Cual sería el estado siguiente si Estando en E3 o E4 las entradas son:  
F=0, De=1, Iz=0
- Cual sería el estado siguiente si Estando en E5 las entradas son:  
F=0, De=1, Iz=0  
F=0, De=0, Iz=1

F	0	0	0	0	1
De	0	0	1	1	X
Iz	0	1	0	1	X
E0	E0	E3	E1	E0	E5
E1	E0	E3	E2	E0	E5
E2	E0	E3	E0	E0	E5
E3	E0	E4	E1	E0	E5
E4	E0	E0	E1	E0	E5
E5	E0	E3	E1	E0	E5

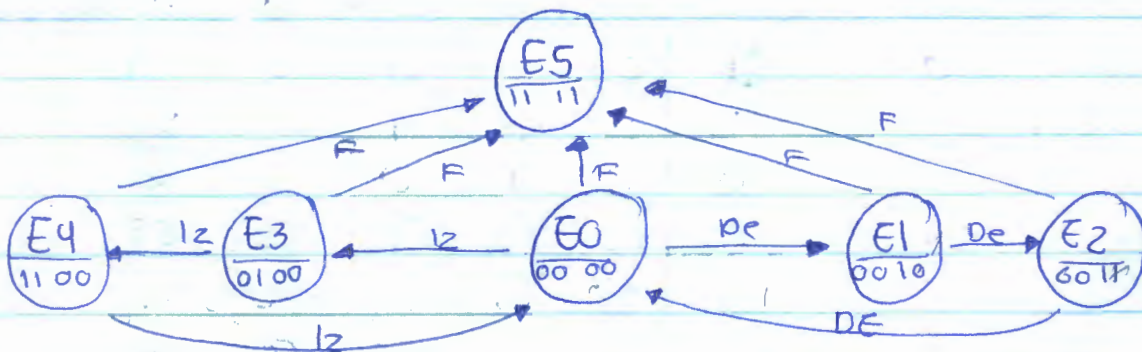
2-Abril-10

## Entradas y Salidas



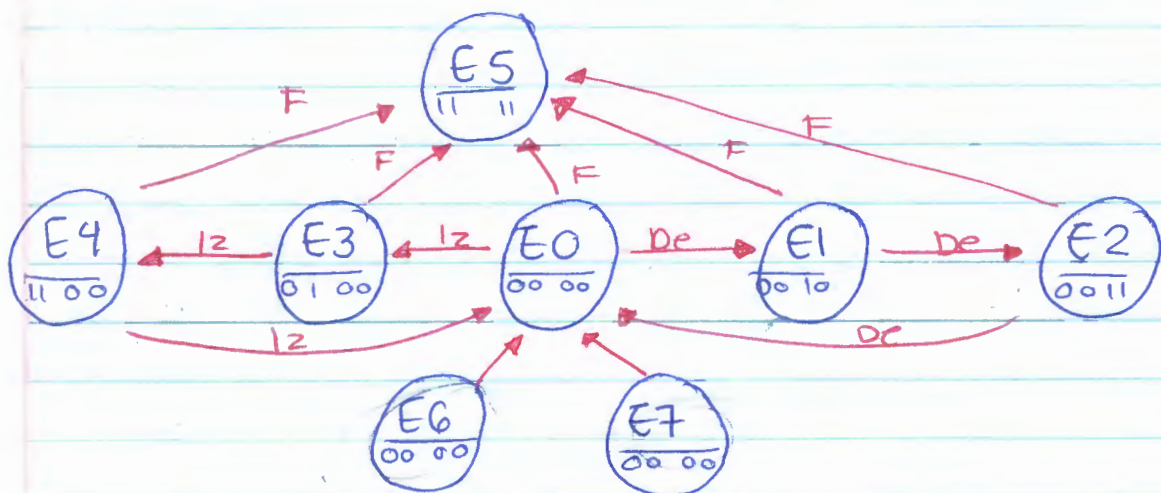
## Asignar Valores a los Estados

F	0	0	0	0	1			
De	0	0	1	1	X	Q2	Q1	Q0
Iz	0	1	0	1	X			
E0	E0	E3	E1	E0	E5	0	0	0
E1	E0	E3	E2	E0	E5	0	0	1
E2	E0	E3	E0	E0	E5	0	1	0
E3	E0	E4	E1	E0	E5	0	1	1
E4	E0	E0	E1	E0	E5	1	0	0
E5	E0	E3	E1	E0	E5	1	0	1



2-Abril-10

F	0	0	0	0	1			
De	0	0	1	1	X	Q2	Q1	Q0
I2	0	1	0	1	X			
E0	E0	E3	E1	E0	E5	0	0	0
E1	E0	E0	E2	E0	E5	0	0	1
E2	E0	E0	E0	E0	E5	0	1	0
E3	E0	E4	E0	E0	E5	0	1	1
E4	E0	E0	E0	E0	E5	1	0	0
E5	E0	E0	E0	E0	E5	1	0	1
E6	E0	E0	E0	E0	E0	1	1	0
E7	E0	E0	E0	E0	E0	1	1	1

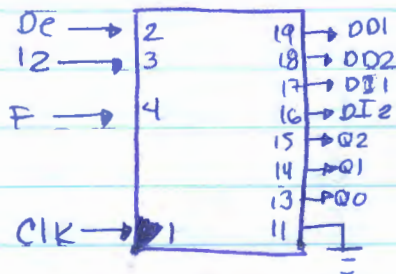


## 1. Entradas y Salidas

### Archivo ABEL-HDL

```

MODULE top
DECLARATIONS
  *Reloj
  CLK PIN 1;
  * variables de Entrada
  De,I,F PIN 2,3,4;
  * variables de Salida Combinacionales
  DD1,DD2,DI1,DI2 PIN 15,16 istype 'com';
  * variables de Salida Registradas FF's
  Q2,Q1,Q0 PIN 15,14,13 istype 'reg';
  
```





2-Abril-10

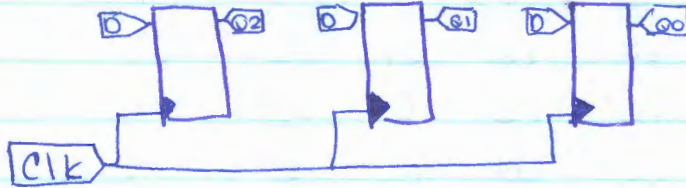
## 2. Sincronización

DECLARATIONS

Sinc = [Q2..Q0];

EQUATIONS

Sinc.clk = CLK;



## 3. Asignar valores a los estados

DECLARATIONS

E0 = [0, 0, 0];

E1 = [0, 0, 1];

E2 = [0, 1, 0];

E3 = [0, 1, 1];

E4 = [1, 0, 0];

E5 = [1, 0, 1];

E6 = [1, 1, 0];

E7 = [1, 1, 1];

	Q2	Q1	Q0
E0	0	0	0
E1	0	0	1
E2	0	1	0
E3	0	1	1
E4	1	0	0
E5	1	0	1
E6	1	1	0
E7	1	1	1

## 4. Tabla de estados

F	0	0	0	0	1
D <sub>e</sub>	0	0	1	1	X
I <sub>2</sub>	0	1	0	1	X
E0	E0	E3	E1	E0	E5
E1	E0	E0	E2	E0	E5
E2	E0	E0	E0	E0	E5
E3	E0	E4	E0	E0	E5
E4	E0	E0	E0	E0	E5
E5	E0	E0	E0	E0	E5
E6	E0	E0	E0	E0	E0
E7	E0	E0	E0	E0	E0

2-Abril-10

### State diagram Sinc

#### State E0:

DD1=0; DD2=0;  
DI1=0; DI2=0;  
IF !F & !De & !Iz Then E0;  
IF !F & !De & Iz Then E3;  
IF !F & De & !Iz Then E1;  
IF !F & De & Iz Then E0;  
IF F Then E5;

#### State E1:

DD1=1; DD2=0;  
DI1=0; DI2=0;  
IF !F & !De & !Iz Then E0;  
IF !F & !De & Iz Then E0;  
IF !F & De & !Iz Then E2;  
IF !F & De & Iz Then E0;  
IF F Then E5;

#### State E3:

DD1=0; DD2=0;  
DI1=1; DI2=0;  
IF !F & !De & !Iz Then E0;  
IF !F & !De & Iz Then E4;  
IF !F & De & !Iz Then E0;  
IF !F & De & Iz Then E0;  
IF F Then E5;

#### State E4

DD1=0; DD2=0;  
DI1=1; DI2=1;  
IF !F & !De & !Iz Then E0;  
IF !F & !De & Iz Then E0;  
IF !F & De & !Iz Then E0;  
IF !F & De & Iz Then E0;  
IF F Then E5;

#### State E5;

DD1=1; DD2=1;  
DI1=1; DI2=1;  
IF !F & !De & !Iz Then E0;  
IF !F & !De & Iz Then E0;  
IF !F & De & !Iz Then E0;  
IF !F & De & Iz Then E0;  
IF F Then E5;

#### State E6;

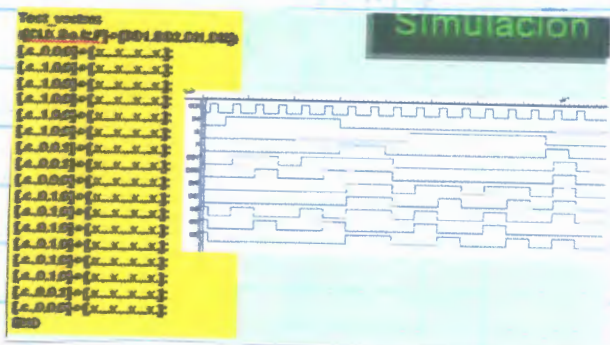
DD1=0; DD2=0;  
DI1=0; DI2=0;  
GOTO E0;

#### State E7:

DD1=0; DD2=0;  
DI1=0; DI2=0;  
GOTO E0;

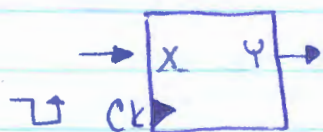
2-Abril-10

## Simulación



## Un detector de Secuencia

Es un circuito el cual tiene una entrada de datos  $x$  en serie y una o varias salidas con las cuales se indica que una o varias secuencias correctas o no.



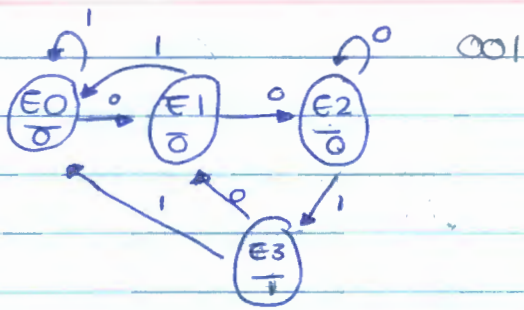
Diseña un sistema secuencial que detecte la secuencia 001.

El detector tendrá una entrada  $x$  que estará sincronizada con la señal de reloj.

A medida que pasan los pulsos del reloj la entrada irá siendo 0 o 1, con lo que se irá "escribiendo" una secuencia en binario (algo como 011000011001000110...).

Cada vez que el circuito detecte la secuencia 001 su salida se pondrá en alta y regresará a condiciones iniciales para detectar de nuevo la secuencia.

2-Abril-140



	0	1	Y
E0	E1	E0	0
E1	E2	E0	0
E2	E2	E3	0
E3	E1	E0	1

MODULE dsccu

"entradas

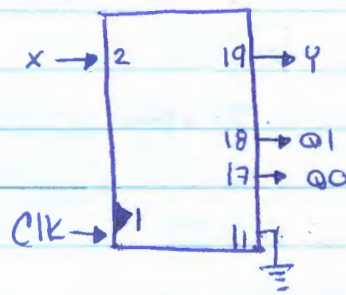
CLK, X pin 1,2;

"Salida combinacional

Y pin 19 istype 'com';

"Salidas registradas

Q1, Q0 pin 18,17 istype 'reg';

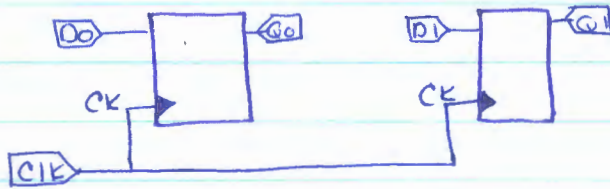


"Sincronización

S = [Q1, Q0];

Equations

S.clk = CLK;



"Asignación de valores a los estados

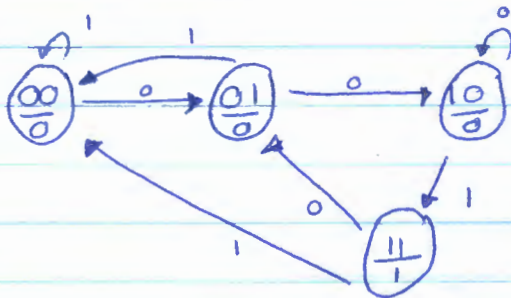
declarations

E0 = [0, 0];

E1 = [0, 1];

E2 = [1, 0];

E3 = [1, 1];



2-Abri-10

state-diagram S

State E0:

Y=0;

IF X then E0 else E1;

State E1:

Y=0;

IF X then E0 else E2;

State E2:

Y=0;

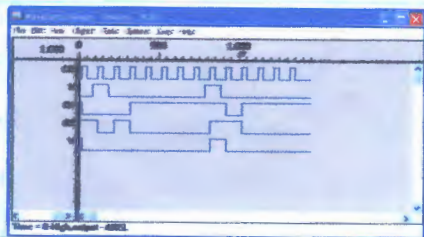
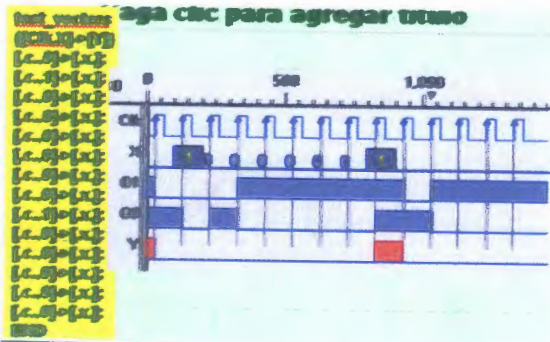
IF X then E3 else E2;

State E3:

Y=1;

IF X then E0 else E1;

	0	1	Y
E0	E1	E0	0
E1	E2	E0	0
E2	E2	E3	0
E3	E1	E0	1

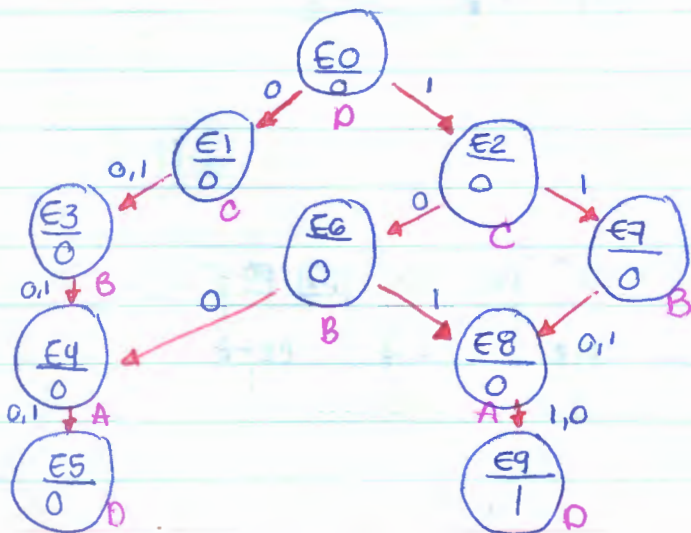
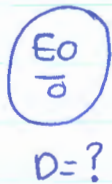
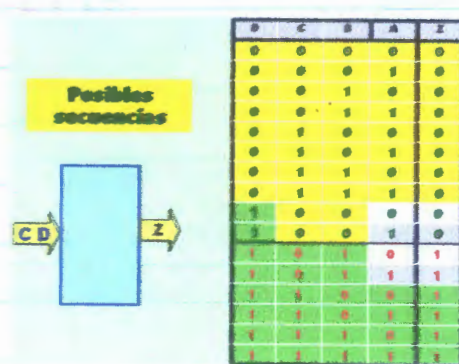


2 Abril 10

## Detector de Secuencia

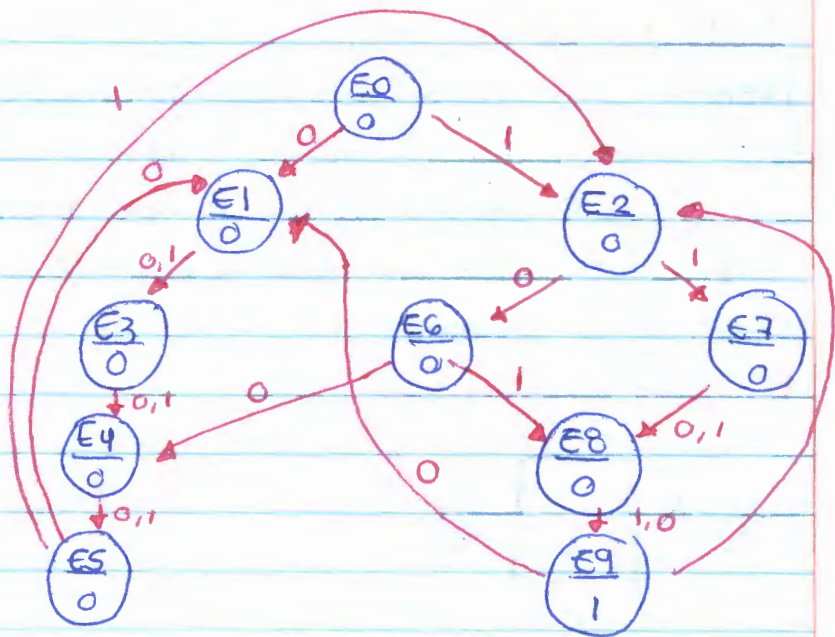
Diseñar un circuito secuencial sincrónico con una sola entrada y una sola salida. Que reconozca números decimales codificados en binario del 0 al 9 (BCD). La salida sea 1 cuando la secuencia es diferente a un número en BCD. El bit más significativo va a la cabeza de la secuencia.

### Posibles Secuencias



2-Abril-10

	0	1
E0	E1	E2
E1	E3	E3
E2	E6	E7
E3	E4	E4
E4	E5	E5
E5	E1	E2
E6	E4	E8
E7	E8	E8
E8	E9	E9
E9	E1	E2



Para lo estados cuantos Flip Flops se requieren

MODULE dcodbcd

"Entradas

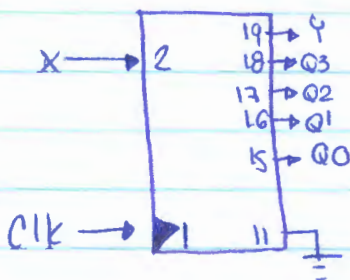
CLK, X pin 1,2;

"Salida

Y pin 19 istype 'com';

"Flip Flops

Q3..Q0 pin 18..15 istype 'reg';

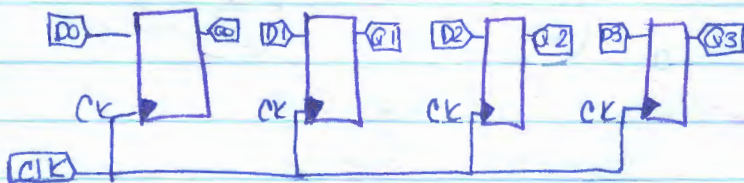


"Sincronización

S = [Q3..Q0];

equations

S.clk = CLK;



2-Abril-10

declaraciones  
asignación de valores a los estados  
E0=[0.0.0]:  
E1=[0.0.1]:  
E2=[0.0.1]:  
E3=[0.0.1]:  
E4=[1.0.0]:  
E5=[1.0.1]:  
E6=[1.1.0]:  
E7=[1.1.1]:  
E8=[1.0.0]:  
E9=[1.0.1]:  
E10=[1.0.1]:  
E11=[1.0.1]:  
E12=[1.1.0]:  
E13=[1.1.1]:  
E14=[1.1.1]:  
E15=[1.1.1]:

State-Diagram S

State E0:

y=0;

IF x then E2 else E1;

State E1:

y=1

goto E3;

State E2:

y=0;

IF x then E7 else E6;

State E3:

y=0;

goto E4;

State E4:

y=0;

goto E5;

	0	1
E0	E1	E2
E1	E3	E3
E2	E6	E7
E3	E4	E4
E4	E5	E5
E5	E1	E2
E6	E4	E8
E7	E8	E8
E8	E9	E9
E9	E1	E2



2-Abril-10

State E5:  
 y=0;  
 IF x then E2 else E1;

State E6:  
 y=0;  
 IF x then E4 else E8;

State E7:  
 y=0

goto E8;

State E8:  
 y=0;

goto E9;

State E9:  
 y=1;

IF x then E2 else E1;

State E10:

y=0;  
 goto E0;

State E11:

y=0  
 goto E0;

State E12:

y=0  
 goto E0;

State E13:

y=0;

goto E0;

State E14:

y=0;

goto E0;

State E15:

y=0;

goto E0;

	0	1
E10	E0	E0
E11	E0	E0
E12	E0	E0
E13	E0	E0
E14	E0	E0
E15	E0	E0

test\_vectors

([CLK] > Y)

[c..0] > x:

[c..1] > x:

[c..1] > x:

[c..1] > x:

[c..0] > x:

[c..0] > x:

[c..1] > x:

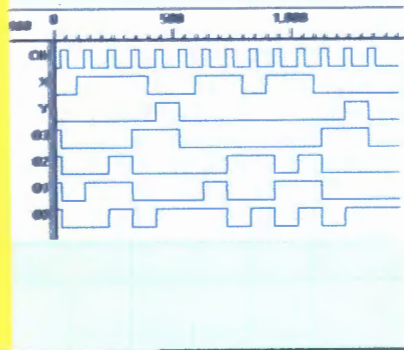
[c..1] > x:

[c..1] > x:

[c..0] > x:

[c..0] > x:

[c..0] > x:



Sec 4

		Estados Próximos					Salidas				Flip Flops		
Estados Faltantes	F	0	0	0	0	1							
	D	0	0	1	1	X							
	I	0	1	0	1	X	DD1	DD2	DI1	DI2	Q2	Q1	Q0
Estados Presentes	E0	E0	E3	E1	E0	ES	0	0	0	0	0	0	0
	E1	E0	E2	E0	ES	1	0	0	0	0	0	1	
	E2	E0	E4	E0	ES	1	1	0	0	0	1	0	
	E3	E0	E4	E2	E0	ES	0	0	1	0	0	1	
	E4	E0	E0	E2	E0	ES	0	0	1	1	1	0	
	ES	E0	E4	E2	E0	E0	1	1	1	1	1	0	
	E6	E0	E0	E0	E0	E0	0	0	0	0	1	1	
	E7	E0	E0	E0	E0	E0	0	0	0	0	1	1	

