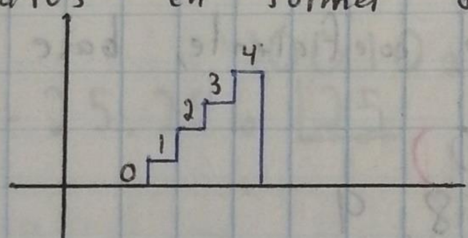
Unidad 1: Introducción a los Sistemas Digitales

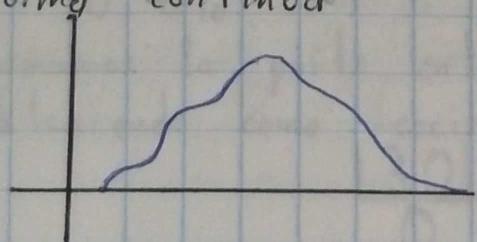
Aprendizajes Esperados:
- Introducción
- Sistemas analógicos y digitales
- Sistemas de numeración: binario y hexadecimal
- Operaciones aritméticas binarias con y sin signo
- Códigos binarios: Gray, ASCII, BCD, Exceso 3 y Paridad

1 - Introducción a Sistemas Digitales

Sistema: Conjunto de elementos interrelacionados para el logro de un objetivo
- Sistema Digital: La representación de los datos en forma discreta



La palabra digital proviene de dígito, que viene a su vez de dedos
- Sistema Analógico: La representación de los datos en forma continua



Ejemplos:
- Número de butacas: Digital (discreto)
- Temperatura: Analógico (continuo

- Plancha: Analógica (continuo)
- Celular: Digital (discreto)
- Velocidad: Analógico (continuo)
- Sonido: Analógico (continuo)
- Horno de microondas: Ambos (mixto o híbrido)
- Reloj de manecillas: Analógico (continuo

Características principales de los sistemas digitales

- Son más precisos
- Al tener más cifras significativas, son más exactos
- Más económico
- Se pueden programar, procesar datos, almacenarlos
- Menor calidad que la analógica
- Una desventaja es que de por sí, la naturaleza es analógica

2.- Sistemas de Numeración

Son sistemas posicionales. Contienen coeficiente, base y exponente

Sistema decimal (Base 10)

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Ejemplo:

$$125.4_{10}$$

$$1 \cdot 10^2 + 2 \cdot 10^1 + 5 \cdot 10^0 + 4 \cdot 10^{-1}$$

$$1 \cdot 100 + 2 \cdot 10 + 5 \cdot 1 + \frac{4}{10} = 125.4_{10}$$

Sistema binario (Base 2)

0, 1

Ejemplo:

$$1101.1_2$$

$$1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1}$$

$$1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 + 1 \cdot \frac{1}{2} = 13.5_{10}$$

Sistema hexadecimal (Base 16)

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

Ejemplo:

$$1A.2_{16}$$

$$1 \cdot 16^1 + 10 \cdot 16^0 + 2 \cdot 16^{-1} = 26.125_{10}$$

Sistema octal (Base 8)

0, 1, 2, 3, 4, 5, 6, 7

Ejemplo:

$$32.4_8$$

$$3 \cdot 8^1 + 2 \cdot 8^0 + 4 \cdot \frac{1}{8} = 26.5_{10}$$

Ejercicio: Convertir de decimal a binario

1.- 18_{10}

18 | 0
9 | 1
4 | 0
2 | 0
1 | 1

10010_2

2.- 25.25_{10}

25 | 1
12 | 0
6 | 0
3 | 1
1 | 1

$.25 \cdot 2 = 0$
$.5 \cdot 2 = 1$

11001.01

3.- 30.6_{10}

Tomamos la parte entera y la dividiremos entre 2, obteniendo como cociente 15 y residuo 0:

30 | 0   15

Repetimos el procedimiento hasta que el cociente no pueda dividirse entre 2

30 | 0
15 | 1
7 | 1
3 | 1
1 | 1

Después tomamos los residuos de abajo hacia arriba:

$$11110$$

Sin embargo, en este caso tenemos decimales. Con ellos se procede multiplicando el decimal por 2, anotando como igual la parte entera del producto, y la decimal se volverá a multiplicar por 2, repitiendo el proceso hasta que no salgan decimales

$$0.6 \cdot 2 = 1$$
$$0.2 \cdot 2 = 0$$
$$0.4 \cdot 2 = 0$$
$$0.8 \cdot 2 = 1$$
$$0.6 \cdot 2 = \ldots$$

Aquí se obtiene un decimal periódico, por lo que se dejó hasta donde se cicla. Para anotar la parte decimal se lee de arriba a abajo

$$11110.1001_2$$

Ejemplo: Convertir 23.4_8 a binario

Para convertir directamente, usaremos 3 bits para representar cada número, ya que $2^3 = 8$

$$2 \quad 3 \; . \; 4$$

$$010 \quad 011 \; . \; 100$$

Finalmente los juntamos simplificando ceros

$$10011.1_2$$

Ejemplo: Convertir 111110.01_2 a octal

Separamos en 3 bits y convertimos

111	110	.	010
7	6	.	2

Juntamos cada parte

76.2_8

Ejemplo: Convertir $A3.5_{16}$ a binario

Se aplica igual que de octal a binario pero con 4 bits

A	3	.	5
10	3	.	5
1010	0011	.	0101

Juntamos: 10100011.0101_2

Ejemplo: Convertir 10110111.10_2 a hexadecimal

Se aplica igual que de binario a octal pero se separa en 4 bits

1011	0111	.	1000
11	7	.	8
B	7	.	8

Juntamos: $B7.8_{16}$

¡Alerta! Para convertir de octal a hexadecimal, lo más sencillo es pasar de octal a binario y de ahí a hexadecimal, igual a la inversa

Ejercicio: Realizar las siguientes conversiones

1- 30.9_{10} a binario

30 | 0
15 | 1
7 | 1
3 | 1
1 | 1

$.9 \cdot 2 = 1$
$.8 \cdot 2 = 1$
$.6 \cdot 2 = 1$
$.2 \cdot 2 = 0$
$.4 \cdot 2 = 0$
$.8 \cdot 2 = ...$

11110.111_2

37.2

2.- 37.2_8 a hexadecimal

3		7	.	2
011		111	.	010

11111.01

0001		1111	.	0100
1		15	.	4

$1F.4_{16}$

3.- $2A.4_{16}$ a binario

2		10	.	4
0010		1010		0100

101010.01_2

Números con signo

Ahora, el bit más significativo será el bit del signo

0 es para positivos

1 es para negativos

Observemos:

$5_{10} = 101_2$ (5)

Bit del signo $\rightarrow$ 0101_2 (+5)

Para los números negativos se utiliza:

- Complemento a 1
- Complemento a 2

Complemento a 1

Se escribe el número como positivo y se cambian los ceros por unos y los unos por ceros

Véase: -5

0101 (+5)

Bit del signo $\rightarrow$ 1010 (-5)

Si deseo saber su valor, sustituyo el bit del signo y vuelvo a complementar

-010

-101

-5

Complemento a 2:

Se escribe el número como positivo, se complementa a 1 y se le suma 1 al bit menos significativo

Véase:

-8

01000	(+8)
10111	(-8 en comp. 1)
11000	(-8 en comp. 2)

Operaciones Aritméticas en Complemento a 1

Se requiere usar una mayor cantidad de bits para procurar no alterar el bit del signo

$$\begin{array}{r} 6_{10} \\ +\ 3_{10} \\ \hline 9_{10} \end{array} \qquad \begin{array}{r} 000110 \\ +\ 000011 \\ \hline 001001 \end{array} = +9_{10}$$

Bit del signo

$$\begin{array}{r} 6_{10} \\ -\ 3_{10} \\ \hline 3_{10} \end{array} \qquad \begin{array}{r} 000110 \\ 111100 \\ \hline \boxed{1}000010 \end{array}$$

Este bit extra se le llama bit de acarreo. Cuando existe, se le suma al bit menos significativo

$$\begin{array}{r} 000010 \\ 1 \\ \hline 000011 \end{array} = +3_{10}$$

Operaciones Aritméticas en Complemento a 2

Se realiza igual que complemento a 1, claro que en complemento a 2, pero aquí, el bit de acarreo se ignora

Unidad 2: Diseño Básico de Circuitos Lógicos

Aprendizajes Esperados:

- Álgebra de Boole
- Análisis de Circuitos Combinacionales
- Universalidad de las compuertas NAND y NOR
- Síntesis de circuitos combinacionales
  - Diseños y descripciones
  - Minimización
  - Mapas de Karnaugh
  - Soma de productos y producto de somas
  - Condiciones de entrada "no importa"
  - Salida múltiple
- Ejemplos de Aplicación

1- Álgebra de Boole

Teoremas

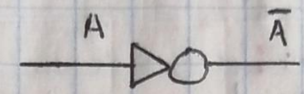
1- $A + A = A$
2- $A \cdot A = A$
3- $A + 0 = A$
4- $A \cdot 1 = A$
5- $A \cdot 0 = 0$
6- $A + 1 = 1$
7- $A + \bar{A} = 1$
8- $A \cdot \bar{A} = 0$
9- $\bar{\bar{A}} = A$
10- $x + y = y + x$
11- $xy = yx$
12- $(x+y)+z = x+(y+z)$
13- $(xy)z = x(yz)$
14- $(x+y)z = xz + yz$
15- $x + \bar{x}y = x + y$
16- $(A+B)' = \bar{A} \cdot \bar{B}$
17- $(A \cdot B)' = \bar{A} + \bar{B}$

Compuertas Lógicas

- Básicas

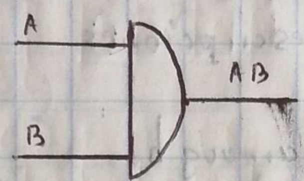
NOT o inversor

A	$\overline{A}$
0	1
1	0



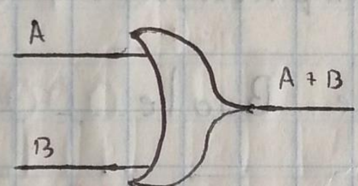
AND

A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1



OR

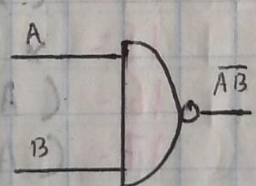
A	B	$A+B$
0	0	0
0	1	1
1	0	1
1	1	1



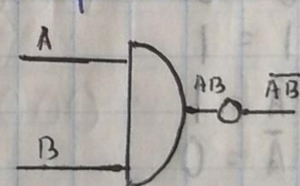
- Compuestas

NAND

A	B	$\overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0

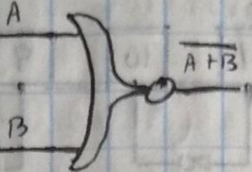


equivalente

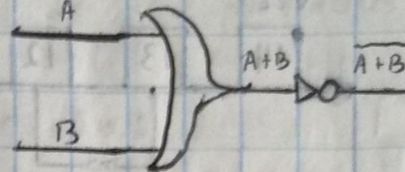


NOR

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

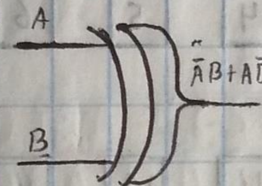


equivalente

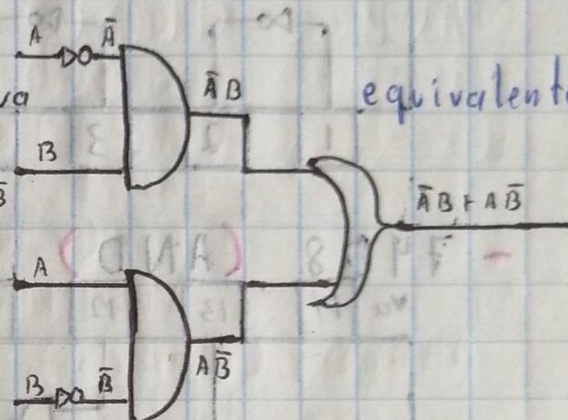


EX OR ú OR Exclusiva

A	B	$\bar{A}B+A\bar{B}$
0	0	0
0	1	1
1	0	1
1	1	0

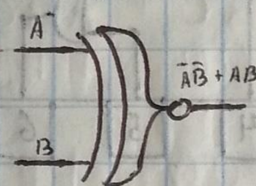


equivalente



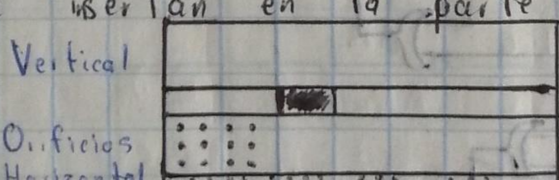
EX NOR

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



Circuitos Integrados

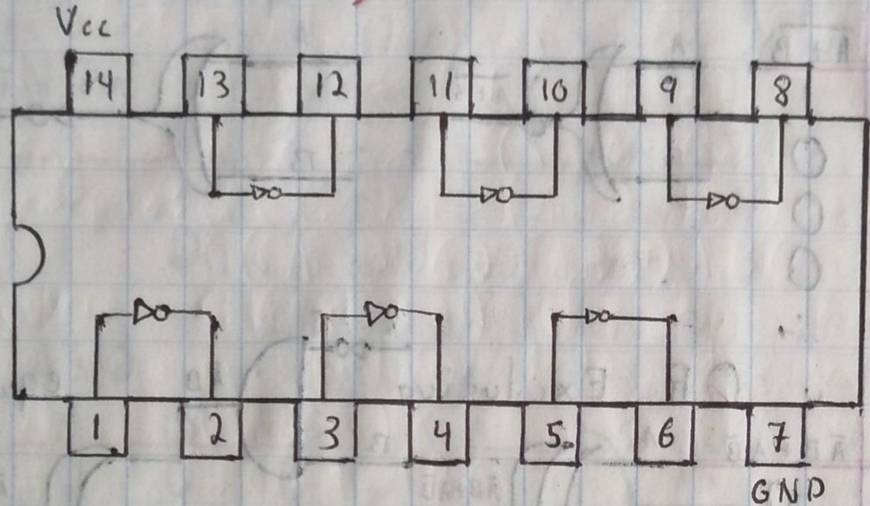
Se insertan en la parte de enmedio del protoboard



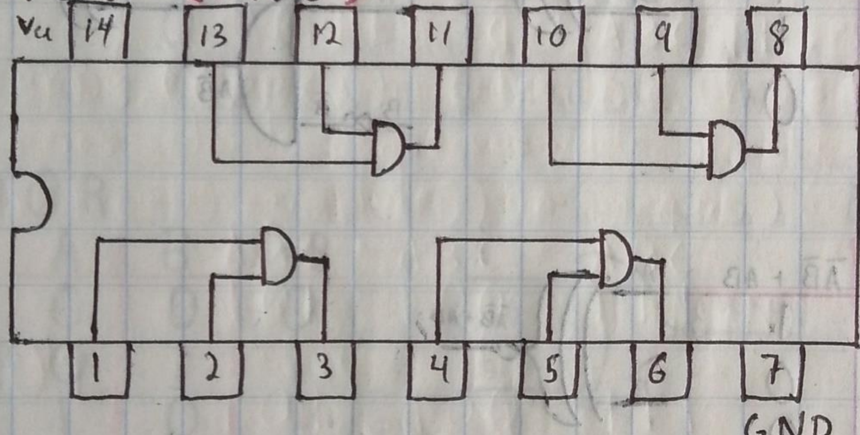
Los orificios están interconectados por la parte posterior

Para que funcione un C. I. (circuito integrado) debe energizarse. Conectar a +5 V y a tierra (ground)
Los pines se enumeran en sentido antihorario con la muesca del lado izquierdo

- 7404 (NOT)



- 7408 (AND)



- 7432 (OR)

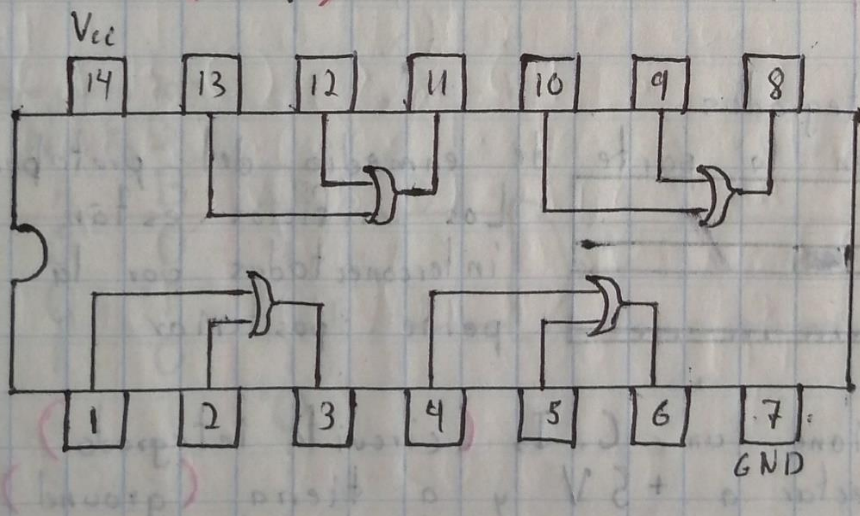


Tabla de Verdad

Muestra las diferentes combinaciones de las entradas y la salida

Por ejemplo, la siguiente expresión:

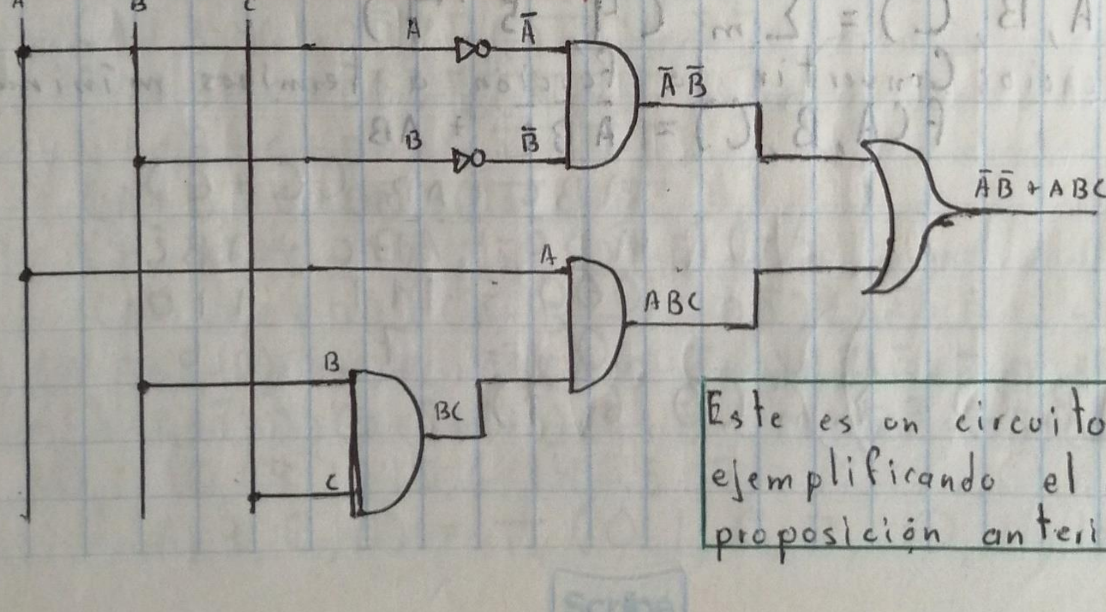
$$\bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC$$

Puede reducirse de la siguiente manera:

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

Y su tabla de verdad es:

Entradas			Salida
A	B	C	$\bar{A}\bar{B} + ABC$
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1



Este es un circuito lógico ejemplificando el de la proposición anterior

Términos mínimos y Términos máximos

Una función lógica se puede expresar en términos mínimos o términos máximos:

- Términos mínimos (Suma de productos):

Ejemplo: Convertir la función a términos mínimos

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

Primero se escribe la función de manera que cada término contenga todas las variables en uso:

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

Después, las variables se transforman a números, donde una variable negada será igual a 0 y una variable sin negar será igual a 1:

$$A\bar{B}C + A\bar{B}\bar{C} + ABC$$
$$101,\ 100,\ 111$$

Dichos números se transforman a decimal

$$101,\ 100,\ 111$$
$$5,\ 4,\ 7$$

Por lo tanto, se escribe que:

$$f(A,B,C) = \sum m\,(4,5,7)$$

Ejercicio: Convertir la función a terminos minimos

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

$$\bar{A}\bar{B}\bar{C} + AB(C+\bar{C})$$
$$\bar{A}\bar{B}\bar{C} + ABC + AB\bar{C}$$
$$000 \quad 111 \quad 110$$
$$0 \quad 7 \quad 6$$

$$f(A,B,C) = \sum m\,(0,6,7)$$

$D = A + B$ $\quad\quad$ $\bar{D} + \bar{C} = \overline{D \cdot C}$

$\overline{(A+B) \cdot C} = \overline{(AC)+(BC)}$

- Términos máximos (Producto de Sumas)

Ejemplo: Convertir la función en términos máximos

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

Igual que en términos mínimos, se escribe la función de modo que cada producto contenga todas las variables:

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

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

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

Después, las variables se transforman a números, donde una variable negada es igual a 1 y una sin negar es igual a 0:

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

0100, 0101, 0001, 0101, 0001

Después se transforman a decimal

0100, 0101, 0001, 0101, 0001

4, 5, 1, 5, 1

Por lo tanto, se escribe que:

$f(A,B,C,D) = \Pi(1,4,5)$

Ejercicio: Convertir la función a términos máximos

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

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

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

$(A+B+C+D\bar{D})[\overline{(A+B)\cdot C}]$

$(A+B+C+D\bar{D})(\overline{AC+BC})$

$(A+B+C+D\bar{D})(\overline{AC}\cdot\overline{BC})$

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

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

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

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

0, 1, 10, 14, 11, 10, 6, 14, 15, 7

$f(A,B,C,D) = \Pi(0,1,6,7,10,11,14,15)$

Ejercicio: Escriba los términos completos

$f(A,B,C) = \sum m(1,5,7)$

$001 \quad 101 \quad 111$

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

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

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

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

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

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

Ejercicio: Escriba los términos completos

$f(A,B,C,D) = \prod(0,3,8,12)$

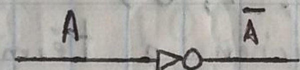
$0000 \quad 0011 \quad 1000 \quad 1100$

$\underbrace{(ABCD)\ (AB\bar{C}\bar{D})\ (\bar{A}BCD)\ (\bar{A}\bar{B}CD)}$

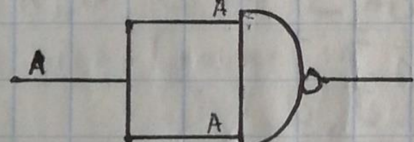
$(A+B+C+D)(A+B+\bar{C}+\bar{D})(\bar{A}+B+C+D)(\bar{A}+\bar{B}+C+D)$

Universalidad de las compuertas NAND y NOR (equivalencias)

• NOT

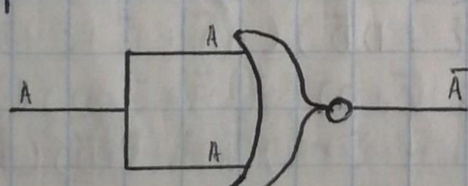


- Equivalencia NAND



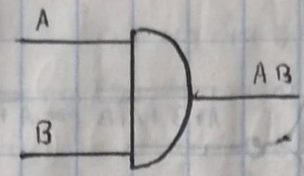
$\overline{A \cdot A} \rightarrow \bar{A} + \bar{A} \rightarrow \bar{A}$

- Equivalencia NOR

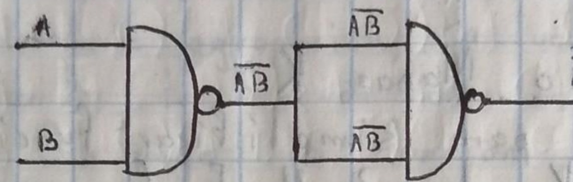


$\overline{A + A} \rightarrow \bar{A} \cdot \bar{A} \rightarrow \bar{A}$

- AND

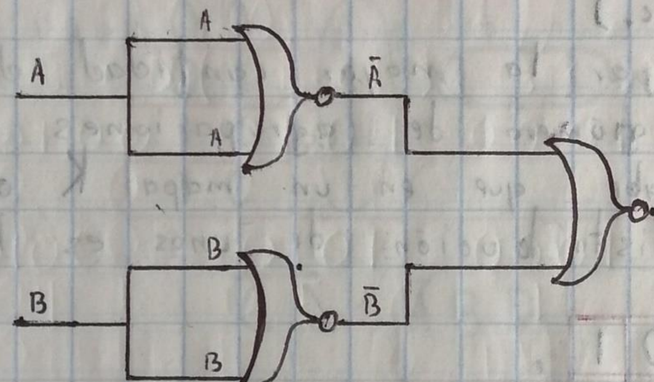


- Equivalencia NAND



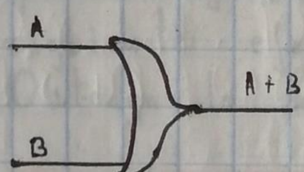
$\overline{\overline{AB}\cdot\overline{AB}} \rightarrow \overline{\overline{AB}}+\overline{\overline{AB}} \rightarrow AB$

- Equivalencia NOR

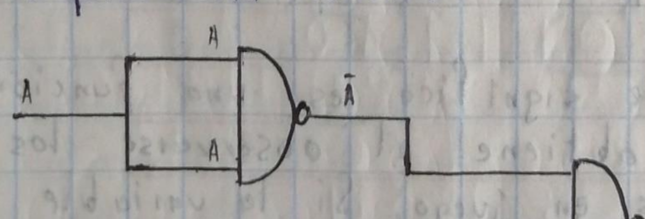


$\overline{\bar{A}+\bar{B}} \rightarrow \bar{\bar{A}}\cdot\bar{\bar{B}} \rightarrow AB$

- OR

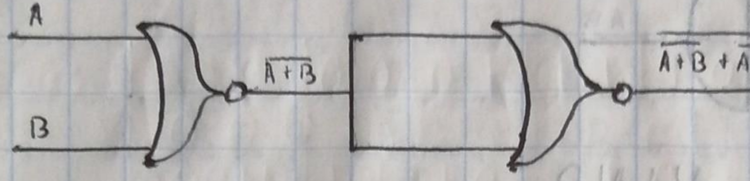


- Equivalencia NAND



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

- Equivalencia NOR



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

Mapas de Karnaugh o Mapas K

Es una herramienta para simplificar funciones lógicas. Hay mapas K de 3, 4, 5, 6 variables. La idea es agrupar unos en potencias de 2 (1, 2, 4, 8, 16, etc.)

El objetivo es agrupar la mayor cantidad de unos con el menor número de agrupaciones

Por ejemplo, si se sabe que en un mapa K de tres variables, la distribución de unos es la siguiente:

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

La agrupación que se trataría de hacer es la marcada con azul, agrupado en la potencia de 2: 4 unos

Esta agrupación, lo que significa es una función booleana, la cual se obtiene al observarse los valores de las variables en juego. Si la variable involucrada posee el valor 0, se tomará como la variable negada, si su valor es 1, será la variable común

Como observamos, en las filas, la variable A se ve involucrada con un 0 y un 1, por lo que esto hace

que se anule la variable A y desaparezca de la función de esta agrupación. Lo mismo ocurre en las columnas con la variable C.
Sin embargo, de vuelta en las filas, la variable B se ve involucrada con 1, lo que hace que en la función de esta agrupación, la variable aparezca normal.
Por lo tanto, la función de este mapa K es B
Ejercicio: Obtenga la función del siguiente mapa

AB \ CD	00	01	11	10
00	1			
01	1	1	1	
11	1			
10	1			

$$\overline{C}\,\overline{D} + \bar{A}BD$$

También pueden agruparse unos de un extremo a otro como en el siguiente ejercicio
Ejercicio: Obtenga la función del siguiente mapa

AB \ CD	00	01	11	10
00				
01	1		1	1
11	1			1
10				

$$B\overline{D} + \bar{A}BC$$

Observamos también, que un mismo uno puede estar incluido en más de una agrupación

Ejercicio: Obtenga la función del siguiente mapa

AB \ CD	00	01	11	10
00	1			1
01				
11	1	1		
10	1	1		1

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

Ejercicio: Obtenga la función del siguiente mapa

AB \ CD	00	01	11	10
00		1	1	
01	1	1		
11		1	1	
10	1	1		

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

Cuando tratamos de reducir una función en términos mínimos, cada cuadro del mapa K es un término mínimo

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

Ejercicio: Reduzca usando mapas K

$f(A, B, C, D) = \sum m(0, 1, 2, 3, 7, 15)$

AB \ CD	00	01	11	10
00	1	1	1	1
01			1	
11			1	
10				

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

Ejercicio: Reduzca usando mapas K

$f(A, B, C, D) = \pi(5, 7, 13, 15)$

AB \ CD	00	01	11	10
00	1	1	1	1
01	1			1
11	1			1
10	1	1	1	1

$\bar{B} + \bar{D}$

Condiciones "no importa"

Las condiciones "no importa" se representan con una x y funcionan como comodín (toman el valor de 1 o 0 para agrupar más unos) como por ejemplo:

AB \ CD	00	01	11	10
00	1	1		
01	1	1	1	
11	x	x	x	x
10	x	x		

$\rightarrow \bar{C} + BD$

Mapas K de 5 variables

El mapa K se construye de la siguiente manera:

	A = 0				A = 1				
BC \ DE	00	01	11	10	10	11	01	00	DE / BC
00	0	1	3	2	18	19	17	16	00
01	4	5	7	6	22	23	21	20	01
11	12	13	15	14	30	31	29	28	11
10	8	9	11	10	26	27	25	24	10

De igual manera, se ven representados los terminos minimos en el mapa

Ejemplo: Obtenga la funcion del siguiente mapa

	A = 0				A = 1				
BC \ DE	00	01	11	10	10	11	01	00	DE / BC
00	1	1							00
01		1	1	1	1				01
11			1			1			11
10	1	1							10

$$\bar{A}\bar{C}\bar{D} + \bar{A}\bar{B}CE + BCDE + \bar{B}CD\bar{E}$$

Ejercicio: Obtenga la funcion del siguiente mapa

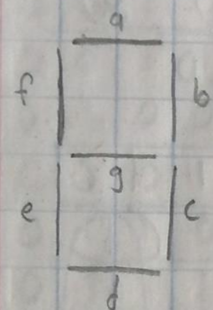
	A = 0				A = 1				
BC \ DE	00	01	11	10	10	11	01	00	DE / BC
00				1	x		1		00
01	1			1	x			1	01
11	1							1	11
10		1					1		10

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

Salida Múltiple

Es cuando un circuito lógico tiene dos o más salidas. Un ejemplo es el decodificador BCD a 7 segmentos

Ejercicio: Diseñe un decodificador de BCD (decimal codificado en binario) a 7 segmentos



A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	x	x	x	x	x	x	x
1	0	1	1	x	x	x	x	x	x	x
1	1	0	0	x	x	x	x	x	x	x
1	1	0	1	x	x	x	x	x	x	x
1	1	1	0	x	x	x	x	x	x	x
1	1	1	1	x	x	x	x	x	x	x

En este ejercicio, se tienen 7 diferentes salidas: a, b, c, d, e, f, g. Usadas para representar los números del 0 al 9. Como el 8 y el 9 necesitan de 4 bits para su representación binaria, se usarán las variables A, B, C, D, pero, con estos 4 bits, también es posible representar números del 10 al 15, los cuales no tendremos en cuenta y les ponemos "x"

Se hará un mapa K para cada salida

AB \ CD	00	01	11	10
00	1		1	1
01		1	1	1
11	x	x	x	x
10	1	1	x	x

$a = C + A + BD + \bar{B}\bar{D}$

AB \ CD	00	01	11	10
00	1	1	1	1
01	1		1	
11	x	x	x	x
10	1	1	x	x

$b = A + \bar{A}\bar{B} + \bar{C}\bar{D} + CD$

AB \ CD	00	01	11	10
00	1	1	1	
01	1	1	1	1
11	x	x	x	x
10	1	1	x	x

$c = A + \bar{C} + CD + \bar{A}B$

AB \ CD	00	01	11	10
00	1		1	1
01		1		1
11	x	x	x	x
10	1	1	x	x

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

AB \ CD	00	01	11	10
00	1			1
01				1
11	x	x	x	x
10	1		x	x

$e = \bar{B}\bar{C}\bar{D} + C\bar{D}$

AB \ CD	00	01	11	10
00	1			
01	1	1		1
11	x	x	x	x
10	1	1	x	x

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

AB \ CD	00	01	11	10
00			1	1
01	1	1		1
11	x	x	x	x
10	1	1	x	x

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

Ahora, se debería construir un circuito con una salida para cada segmento, pero por su complejidad, se omitió

Ejercicio: Diseñe un comparador de 2 números de 2 bits cada uno

A	B	C	D	x > y	x < y	x = y
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	1	0	0
0	1	0	1	0	0	1
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	0	1
1	0	1	1	0	1	0
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	0	1

Se hace un mapa K por cada salida

AB \ CD	00	01	11	10
00	1			
01		1		
11			1	
10				1

$S_0 = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}B\bar{C}D + ABCD + A\bar{B}C\bar{D}$

$BD(\bar{A}\bar{C} + AC) + \bar{B}\bar{D}(\bar{A}\bar{C} + AC)$

$(BD + \bar{B}\bar{D})(\bar{A}\bar{C} + AC)$

AB \ CD	00	01	11	10
00		1	1	1
01			1	1
11				
10			1	

$S_1 = \bar{A}C + \bar{A}\bar{B}D + \bar{B}CD$

AB \ CD	00	01	11	10
00				
01	1			
11	1	1		1
10	1	1		

$S_2 = A\bar{C} + B\bar{C}\bar{D} + AB\bar{D}$

Ahora, se debería construir un circuito con una salida para cada situación, pero por su complejidad, se omitió

Este ejercicio es un ejemplo de un comparador.
Tema que pertenece a la siguiente unidad

Unidad 3: Estructuras Digitales con Lógica Combinacional

Aprendizajes Esperados:
- Decodificadores
  - Decodificador binario
  - Decodificadores de 7 segmentos
- Codificadores
  - Codificadores de prioridad
- Multiplexores y demultiplexores
- Compuertas combinadas XOR y XNOR
- Comparadores
- Sumadores
  - Sumadores medios y completos
  - Sumadores de n-bits
- Dispositivos de tres estados

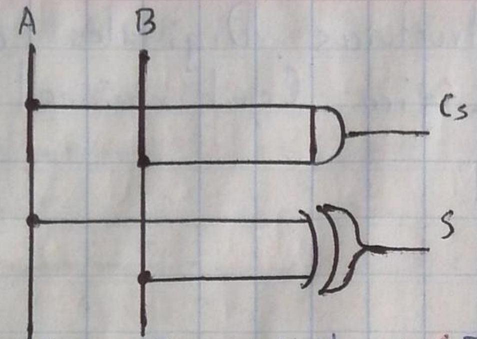
Sumadores

Medio Sumador (Half Adder)

Es un sumador inicial que sumará dos variables, el cual tendrá su salida y un "carry" de salida, por ejemplo:

A	B	C_s	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Como observamos, la función booleana de la salida es una compuerta EX-OR, es decir, $S = \bar{A}B + A\bar{B}$ mientras que la del carry de salida es $C_s = AB$, por lo que su circuito sería:



Sumador Completo (Full Adder)

Este sumador, sumará dos variables y agregará un carry de salida de algún adder anterior, de modo que se tiene:

A	B	C_E	C_S	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Con la información anterior, también deben obtenerse las funciones de la salida y el carry de salida, por lo que se usarán mapas K:

AB \ C_E	0	1
00		1
01	1	
11		1
10	1	

AB \ C_E	0	1
00		
01		1
11	1	1
10		1

$S = \bar{A}\bar{B}C + \bar{A}B\bar{C} + ABC + A\bar{B}\bar{C}$

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

$C_S = AB + BC + AC$

Funciones que nos conducirían a un circuito, que, por complejidad, se omitió

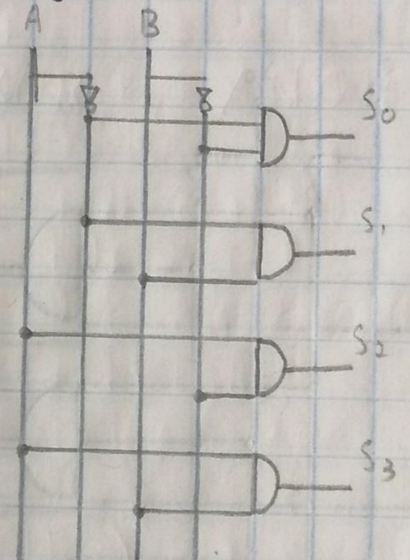
Decodificador

Como su nombre lo dice, decodifica una señal de binario a decimal. Su nombre se indica escribiendo primero el número de entradas y luego el número de salidas. Por ejemplo, un decodificador 2×4 tiene 2 entradas y 4 salidas.

El número de salidas siempre será dado de elevar 2 al número de entradas

Ejercicio: Realice el circuito lógico de un decodificador 2×4

A	B	S_3	S_2	S_1	S_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

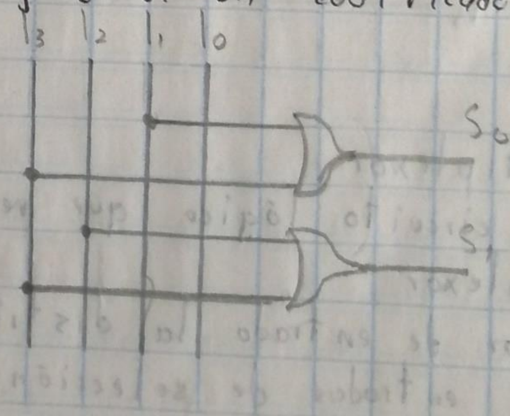


Codificador

Realiza la acción contraria a un decodificador. Éste, a partir de entradas decimales, codificará el pulso a BCD

Ejercicio: Realice el circuito lógico de un codificador 4×2

I_3	I_2	I_1	I_0	S_1	S_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

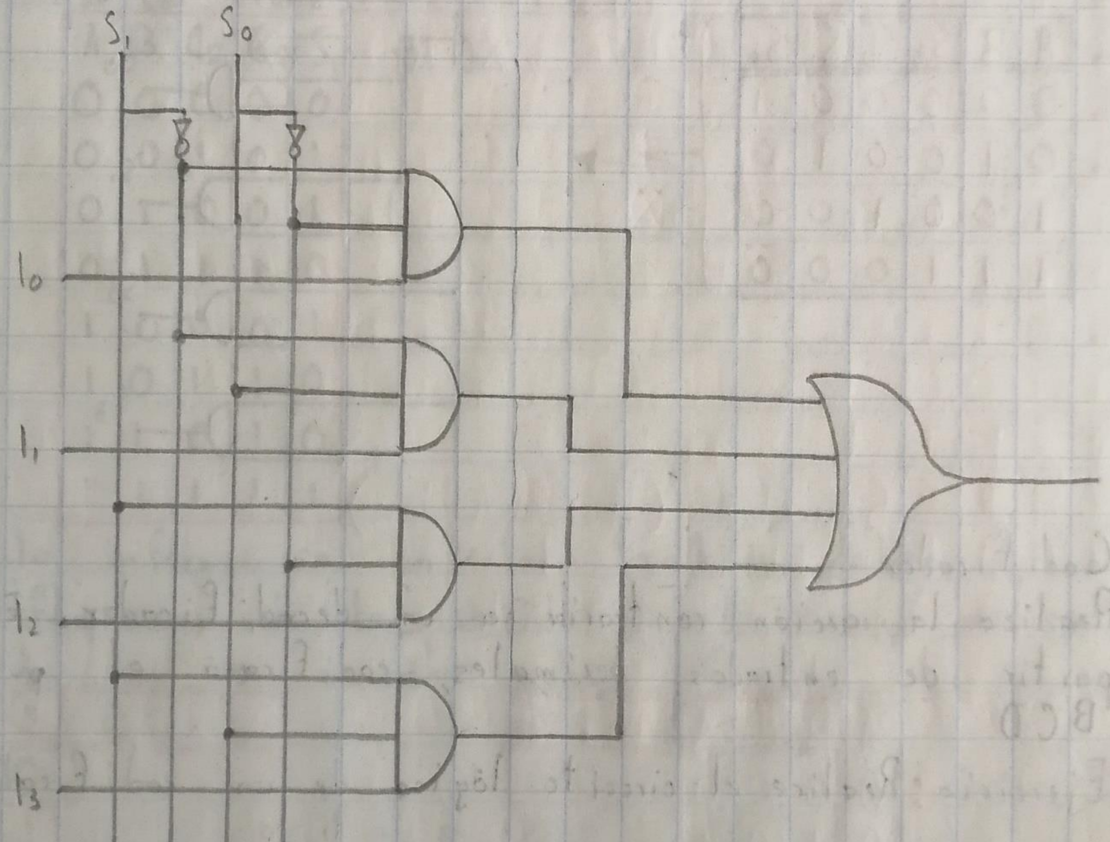


Multiplexor

Es un circuito lógico que funciona como una especie de selector para elegir cuál entrada estará en la salida

Ejercicio: Diseñar un Mux 4 × 1

S_1	S_0	Salida
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3



Demultiplexor

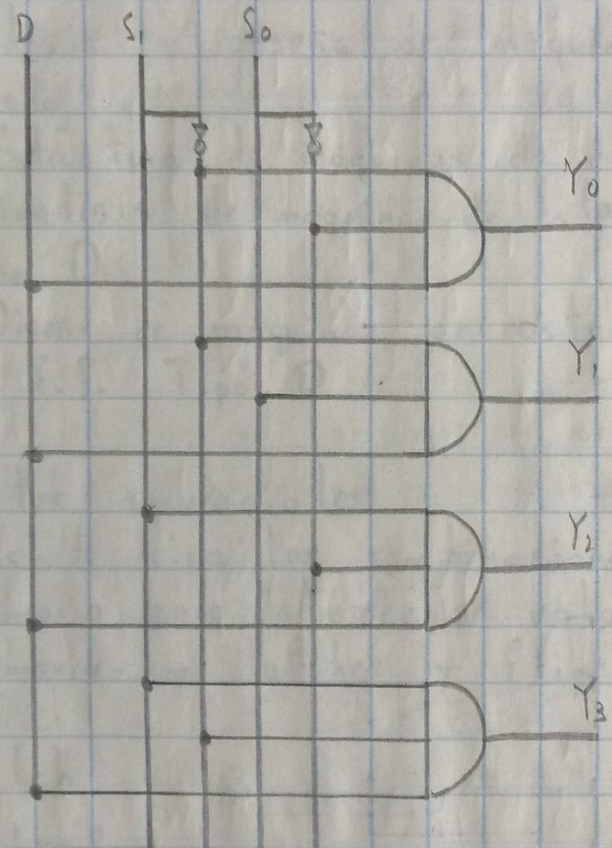
Es un circuito lógico que realiza lo contrario a un multiplexor

El valor de entrado la distribuye en la salida dependiendo de las entradas de selección

Ejercicio: Diseñar un Demux 1×4

Se requieren 2 entradas de selección, porque $2^2 = 4$

S_1	S_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	D
0	1	0	0	D	0
1	0	0	D	0	0
1	1	D	0	0	0



Unidad 4: Estructuras Digitales con Lógica Secuencial

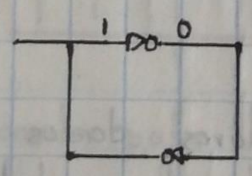
Aprendizajes Esperados:

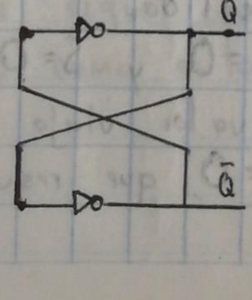
- Cerrojo SR
- Flip - Flop D
- Registros
- Contadores
- Análisis y diseño de máquinas de estado sincronizadas por reloj
  - Estructura de máquinas de estado tipo Moore
  - Análisis de máquinas de estado tipo Moore con F. F. Tipo D
  - Diseño de máquinas de estado tipo Moore y síntesis con F. F. Tipo D

Circuitos secuenciales

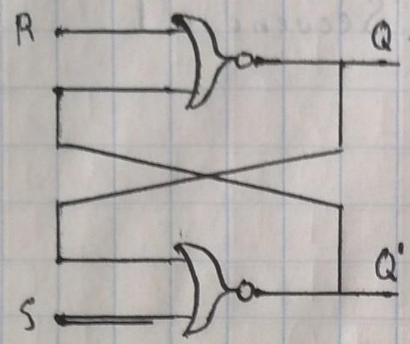
Además de circuitos combinacionales, requieren elementos de memoria para almacenar valores. Estos elementos de memoria son latches y flip - flops

Latch





Latch SR

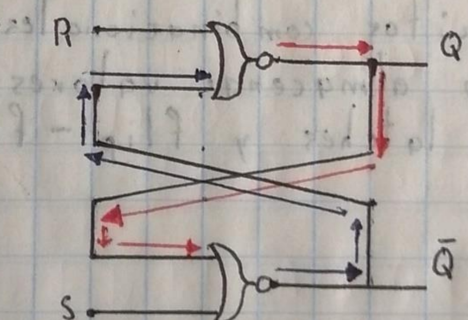


$Q(t) \rightarrow$ Valor anterior
$Q(t+1) \rightarrow$ Valor nuevo

La construcción de la tabla de verdad de este circuito depende una previa, la cual se hace como cualquier tabla para determinar todas las posibles combinaciones de 3 variables de entrada y una de salida

S	R	Q(t)	Q(t+1)
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Para determinar la salida, hay que hacer el recorrido del circuito:



Se inicia desde el punto Q, con los valores dados por la tabla. En este caso, se usarón los valores del primer renglón para obtener la primera salida.
Se llega al primer NOR con valores $Q=0$ y $S=0$, lo que resulta en un 1 para $\bar{Q}$. Este valor viaja al siguiente NOR con valores $\bar{Q}=1$ y $R=0$, que resulta en un 0 para Q

Observamos que Q quedó con valor 0 y $\bar{Q}$ con valor 1, lo cual se espera que siempre ocurra (tener valores inversos) y la pimera salida será 0 por Q = 0

Se hace el mismo recorrido para los valores del segundo renglón. Se llega al primer NOR con Q = 1 y S = 0, resultando en un 0 para $\bar{Q}$. Este valor viaja al siguiente NOR con valores $\bar{Q}$ = 0 y R = 0, resultando en un 1. Obtenemos Q = 1 y $\bar{Q}$ = 0 por lo que se respeta la regla y la segunda salida será 1

De esta manera, es posible llenar la tabla hasta el siguiente punto:

S	R	Q(t)	Q(t+1)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	
1	1	1	

¿Pero qué ocurre con los últimos dos renglones?
Se realiza el recorrido llegando al primer NOR con Q = 1 y S = 1, resultando en un 0 para $\bar{Q}$. Éste viaja al siguiente NOR con $\bar{Q}$ = 0 y R = 1, resultando en un 0 para Q incumpliendo la regla que exige que Q y $\bar{Q}$ sean valores inversos

Cuando pasa esto, se vuelve a hacer el recorrido con estos valores obtenidos una o máximo dos veces y si la regla sigue sin cumplirse, es necesario señalizar el caso, como estos últimos dos renglones

S	R	Q(t)	Q(t+1)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0 *
1	1	1	0 *

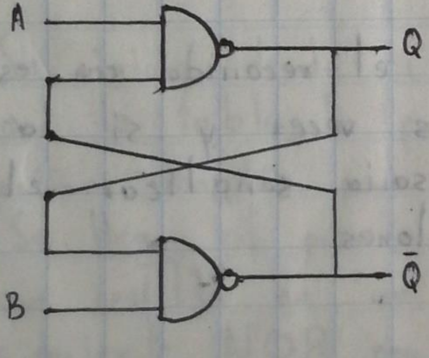
A partir de aquí, se buscará obtener la tabla de manera resumida, es decir, con la información de las variables S y R únicamente.

La tabla es la siguiente:

S	R	Q(t+1)	
0	0	Q(t)	
0	1	0	
1	0	1	
1	1	X	No permitido

El análisis se hace a partir de la tabla anterior. Se observa que para ambos casos donde S y R equivalen a 0, Q(t+1) tendrá el valor de Q(t). En ambos casos donde $S=0$ y $R=1$, $Q(t+1)=0$. Para los casos donde $S=1$ y $R=0$, $Q(t+1)=1$. Los restantes donde S y R son 1, son los casos no permitidos

Ejercicio: Determine la tabla de verdad



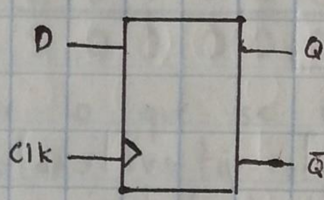
A	B	Q(t)	Q(t+1)
0	0	0	1 *
0	0	1	1 *
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

A	B	Q(t+1)
0	0	X
0	1	1
1	0	0
1	1	Q(t)

Tipos de Flip-Flop

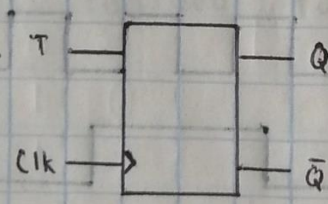
Un flip-flop puede almacenar 1 bit. Si se requieren almacenar 8 bits, se necesitan 8 flip-flops
Los flip-flops son síncronos, es decir, tienen una entrada de reloj (clock). Cuando está en la transición positiva (cuando cambia de 0 a 1) verifica el valor de la entrada

- Flip-flop tipo D (Dato).



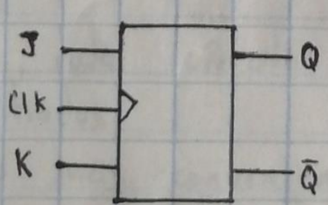
D	Q(t+1)
0	0
1	1

- Flip-flop T (toggle)



T	Q(t+1)
0	Q(t)
1	$\bar{Q}$(t)

- Flip-flop J-K



J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	$\bar{Q}$(t)

Diagramas de Tiempo

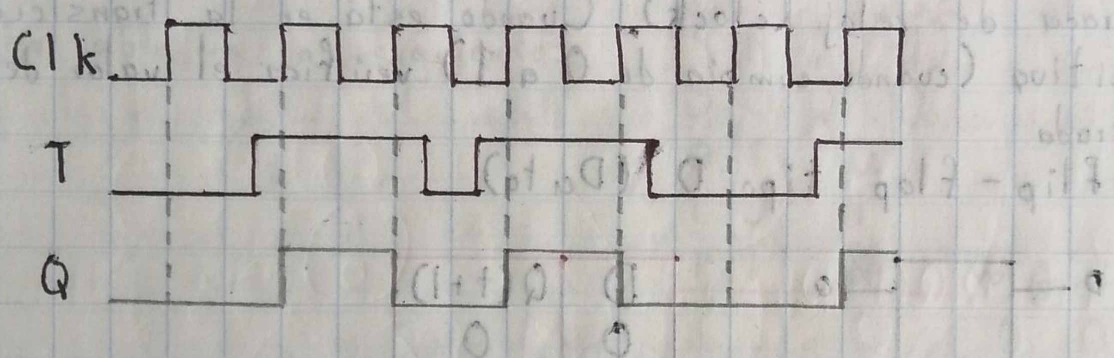
Muestran el funcionamiento de los flip flops

En un mismo renglón, si la señal está arriba, representa un 1, si está abajo es un 0

Las transiciones del reloj ya están dadas por un comportamiento uniforme

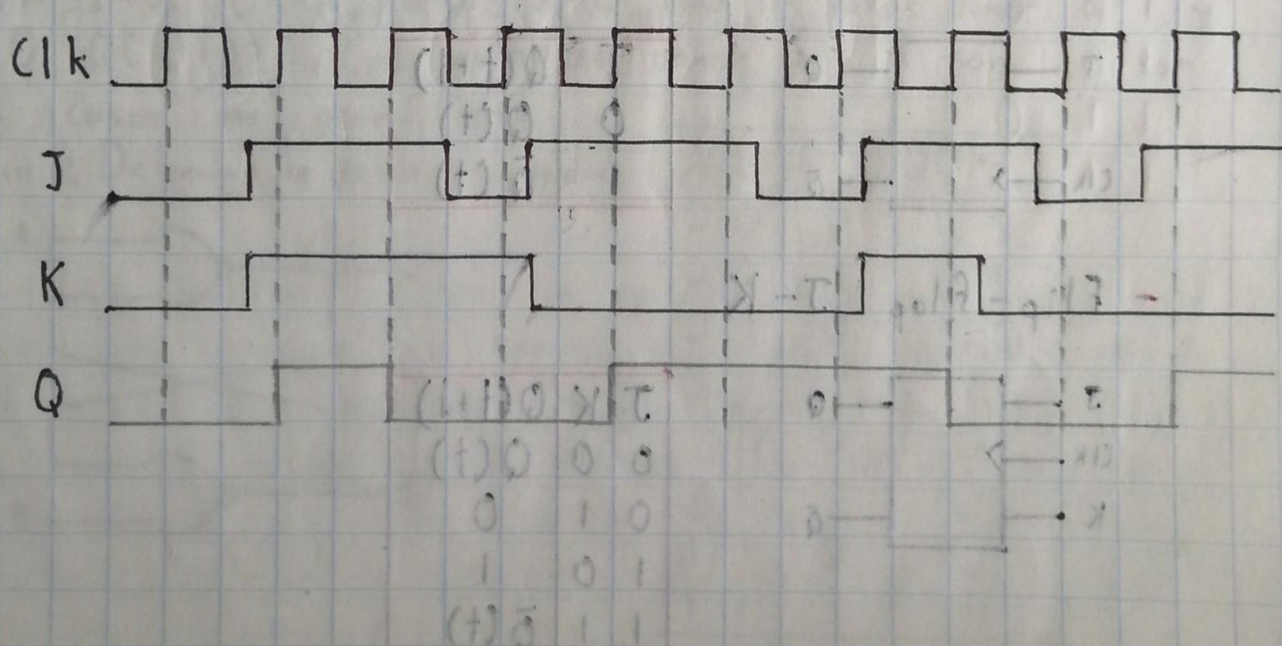
Para obtener la salida, hay que identificar el tipo de flip-flop y seguir las reglas dadas

Ejercicio: Determine la salida del flip-flop T:



A partir de la tabla del flip-flop T y evaluando en cada transición positiva, se obtuvo la salida anterior

Ejercicio: Determine la salida del flip-flop J-K:

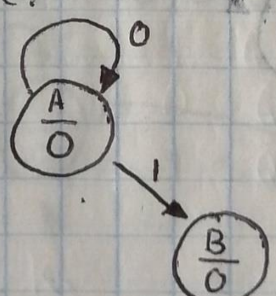


Diagramas de Estado
Representación del funcionamiento de los circuitos secuenciales
- Moore: La salida únicamente depende del estado
- Mealy: La salida depende del estado y de la entrada
Para ilustrar este tipo de diagramas, se realizarán dos ejemplos:
Ejemplo: Dibuje el diagrama de estados de un circuito secuencial de Moore, que detecte la secuencia de 4 unos consecutivos sin solapamiento

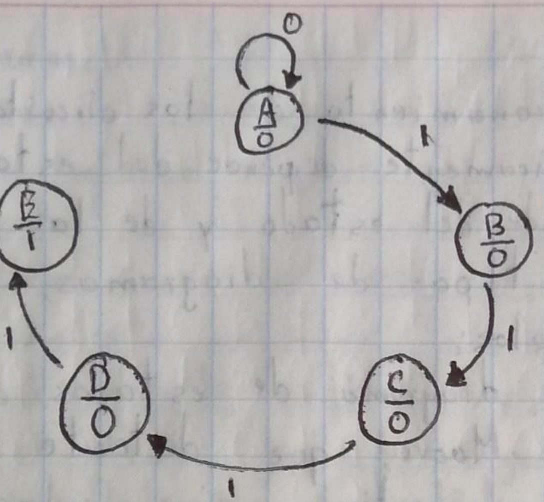
Analizando las instrucciones, el ejercicio pide que a partir de una entrada como la siguiente secuencia, la salida sea:

E	0	1	1	0	0	1	1	1	1	1	1	1	1	0
S	0	0	0	0	0	0	0	0	1	0	0	0	1	0

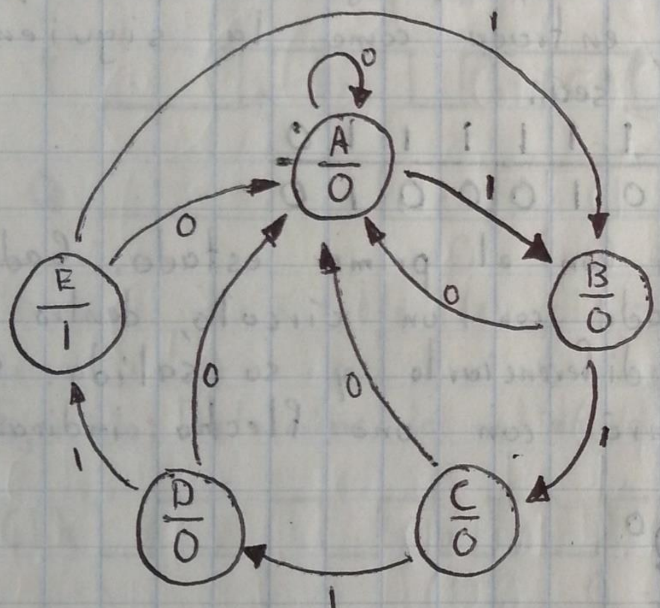
Por lo que se empieza con el primer estado. Cada estado está representado con un círculo, dentro de él, una letra para diferenciarlo y su salida. Cada nueva entrada se hará con una flecha indicando qué entrada fue:



De cada círculo deben salir todas las posibles entradas.
Lo más conveniente para armar el diagrama, es primeramente construir el camino o caminos que guíen a que la salida sea 1

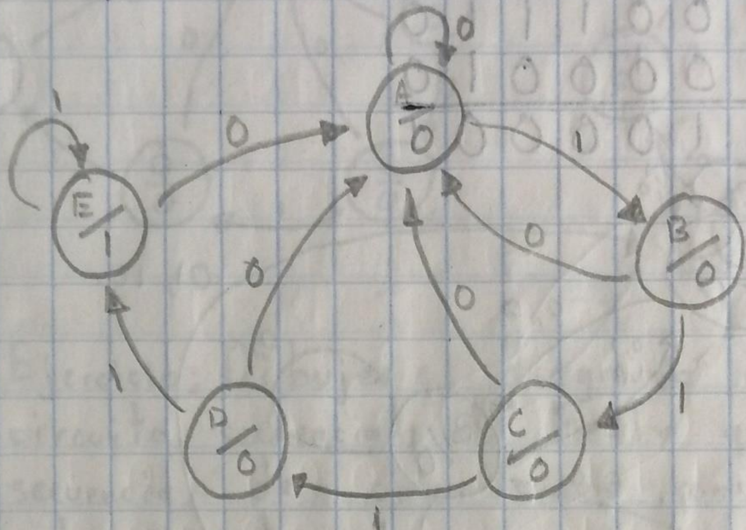


Una vez construido este camino, se debe completar cada estado con las entradas faltantes:



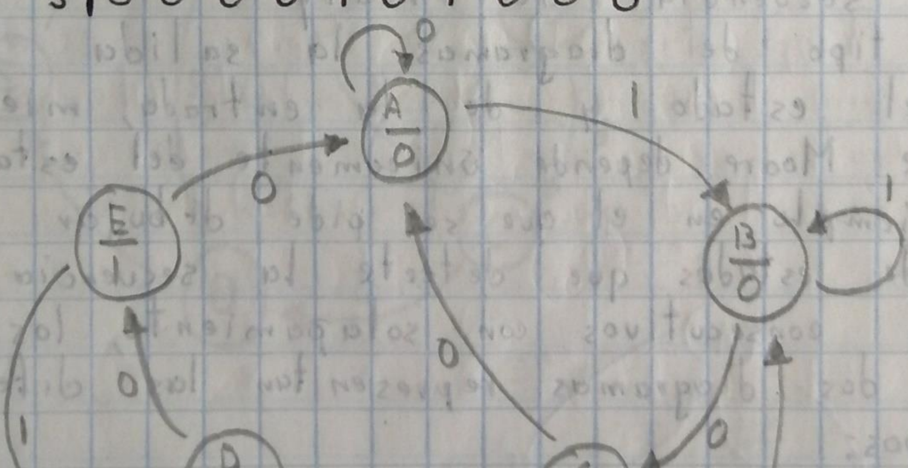
Ejercicio: Dibuje el diagrama de estado de un circuito secuencial de Moore que detecte la secuencia de 4 unos consecutivos con solapamiento

E	0	1	0	1	1	1	1	1
S	0	0	0	0	0	0	1	1



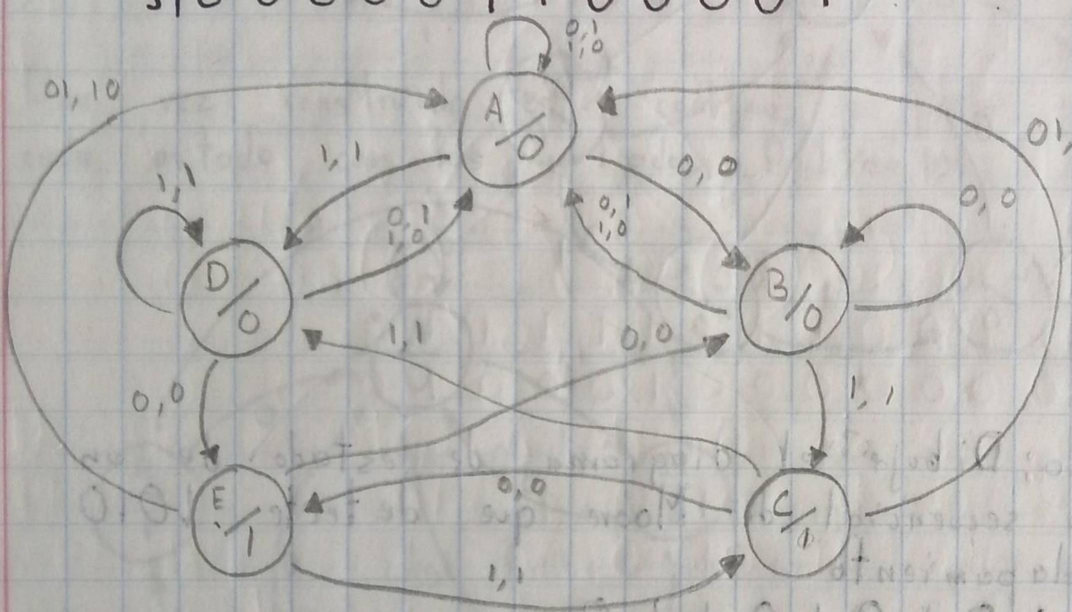
Ejercicio: Dibuje el diagrama de estado de un circuito secuencial de Moore que detecte 1010 con solapamiento

E	1	1	0	1	0	1	0	1	1	0
S	0	0	0	0	1	0	1	0	0	0



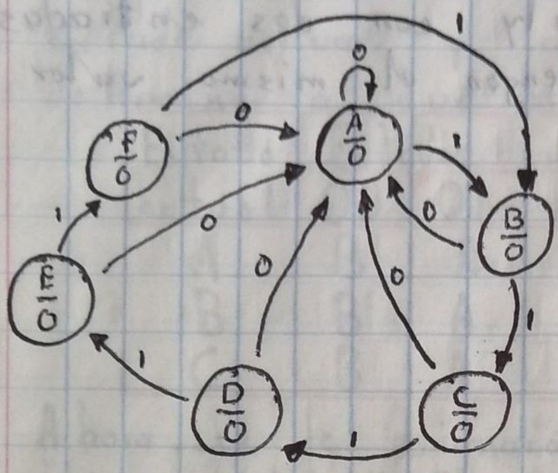
Ejercicio: Dibuje el diagrama de estado de un circuito secuencial de Moore
Se tienen dos entradas y una salida. Esta será 1 cuando ambas tengan el mismo valor pero contrario al anterior

X	0	1	1	0	0	1	0	0	1	1	1	0
Y	1	0	0	0	0	1	0	0	0	0	1	0
S	0	0	0	0	0	1	1	0	0	0	0	1

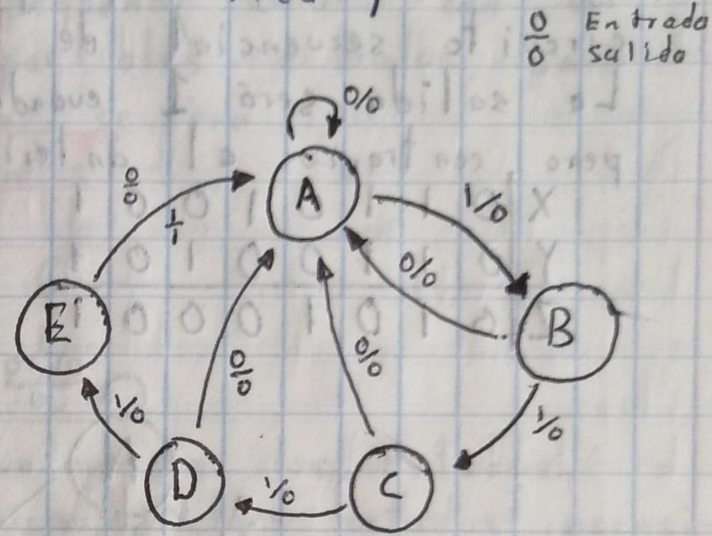


Circuito secuencial de Mealy
En este tipo de diagramas, la salida depende del estado y de la entrada, mientras que el de Moore depende únicamente del estado
Para el ejemplo en el que se pide dibujar un diagrama de estados que detecte la secuencia de cinco unos consecutivos solapamiento, los siguientes dos diagramas representan las diferencias entre ambos:

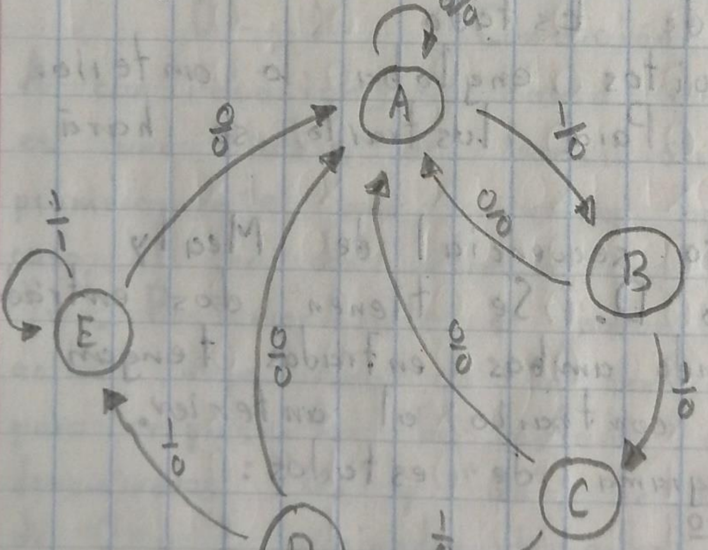
Moore



Mealy

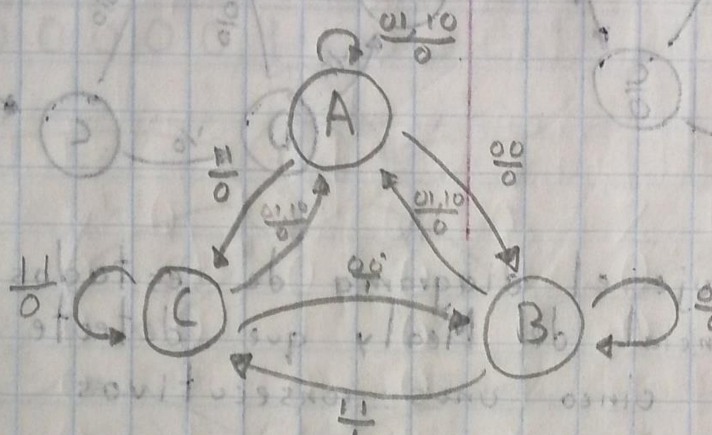


Ejercicio: Dibuje el diagrama de estados de un circuito secuencial de Mealy que detecte la secuencia de cinco unos consecutivos con solapamiento



Ejercicio: Dibuje el diagrama de estados de un circuito secuencial de Mealy con dos entradas. La salida será 1 cuando tengan el mismo valor pero contrario al anterior

X	0	1	1	0	1	0	0	1
Y	0	1	1	0	0	1	0	1
Z	0	1	0	1	0	0	0	1

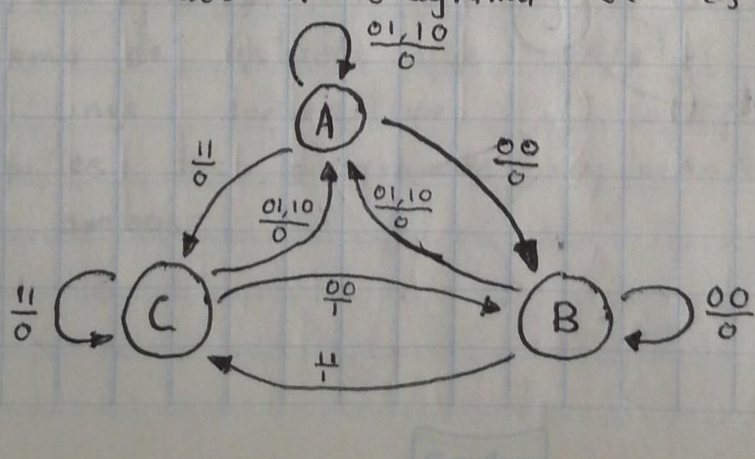


Diseño de Máquinas de Estado

El diseño de estos circuitos engloba lo anterior con algunas adiciones. Para ilustrarlo, se hará un ejemplo

Ejemplo: Diseñe un circuito secuencial de Mealy usando flip flops tipo D. Se tienen dos entradas, la salida será 1 cuando ambas entradas tengan el mismo valor pero contrario al anterior.

Primero se hace el diagrama de estados:



Después se hace una pequeña tabla con el estado actual y las posibles entradas, para determinar a qué estado se irá

Estado actual	Estados siguientes 00	01	10	11
A	B	A	A	C
B	B	A	A	C
C	B	A	A	C

Ahora se le asignarán valores a los estados "binarizándolos". Para representar tres estados necesitamos dos bits y nos sobrará un estado que, como no existe, se tomará como condición no importa

A = 00
B = 01
C = 10
D = 11 - Condición X -

A continuación, se hará una tabla más grande que es una tabla de transición. En ella haremos primeramente una columna para las entradas, en este caso, al ser dos entradas, se harán dos columnas "x" y "y". Posteriormente se harán columnas para el estado actual dependiendo de la cantidad de bits utilizados para representar los estados, en este caso serán dos denominadas "Q_1" y "Q_0". Se harán todas las combinaciones de 0 y 1 originadas de estas columnas. Después se determina el estado siguiente copiando Q_1 y Q_0 pero usando la pequeña tabla y la asignación de valores para representarlos en bits. Los estados que no existan se trabajarán como condiciones no importa. Por estar haciendo un circuito de Mealy se agregará una columna extra para la salida "Z". En los circuitos de

Moore, esta columna se omite porque la salida está dada por el estado. Finalmente se hace una columna con la que obtendremos las entradas de cada flip flop. Para ello, también debemos considerar el tipo de flip flop que se nos pidió. Como en este caso es un tipo D, los valores de las columnas del estado siguiente se copiarán a las columnas de las entradas de flip flops.

La tabla, entonces, quedaría así:

Entradas		E. Actual		E. siguiente		Salida	Entradas F.F.	
x	y	Q_1	Q_0	Q_1	Q_0	Z	D_1	D_0
0	0	0	0	0	1	0	0	1
0	0	0	1	0	1	0	0	1
0	0	1	0	0	1	1	0	1
0	0	1	1	x	x	x	x	x
0	1	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0
0	1	1	0	0	0	0	0	0
0	1	1	1	x	x	x	x	x
1	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0
1	0	1	0	0	0	0	0	0
1	0	1	1	x	x	x	x	x
1	1	0	0	1	0	0	1	0
1	1	0	1	1	0	1	1	0
1	1	1	0	1	0	0	1	0
1	1	1	1	x	x	x	x	x

Con lo obtenido a partir de las columnas de las entradas de flip flops y la salida, se harán mapas K para determinar la función de cada columna. En circuitos de Moore, el mapa K de la salida se omite

$xy \backslash Q_1Q_0$	00	01	11	10
00			x	
01			x	
11	1	1	x	1
10			x	

$$D_1 = xy$$

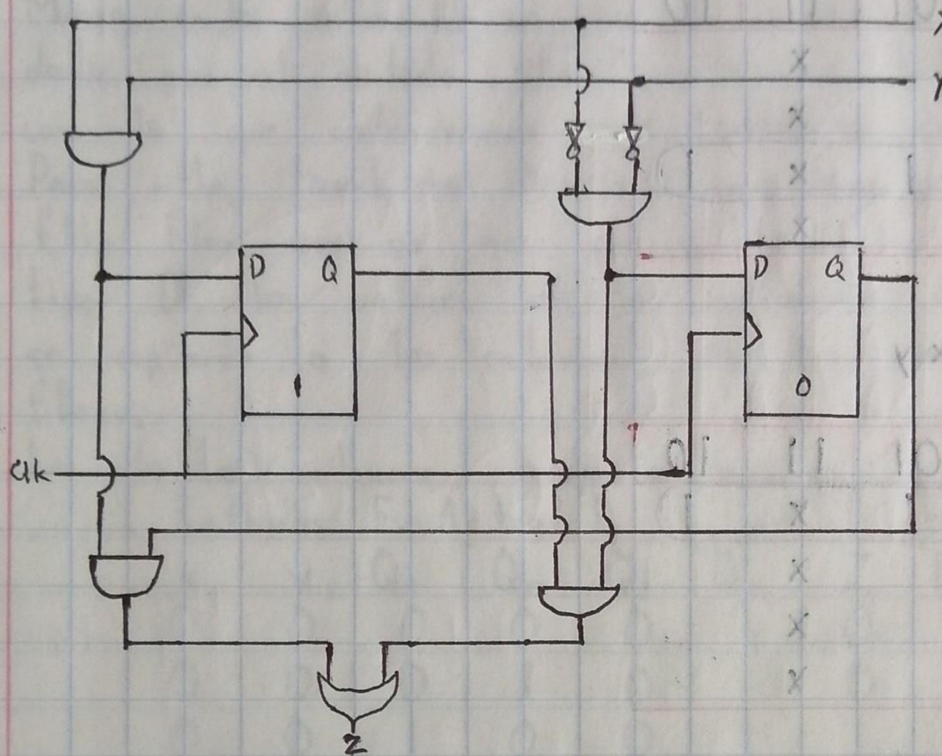
$xy \backslash Q_1Q_0$	00	01	11	10
00	1	1	x	1
01			x	
11			x	
10			x	

$$D_0 = \bar{x}\bar{y}$$

$xy \backslash Q_1Q_0$	00	01	11	10
00			x	1
01			x	
11		1	x	
10			x	

$$Z = xyQ_0 + \bar{x}\bar{y}Q_1$$

Finalmente se hace el circuito con los flip flops, sus entradas, las entradas del circuito y su salida. En un circuito de Moore, sólo se toma la salida de cada flip flop, se conectan a un AND y esta será su salida.
El circuito queda de la manera siguiente:


x
y
D
Q
1
D
Q
0
clk
z