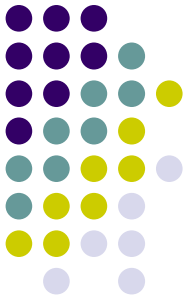
Elementos de Diseño de Sistemas Digitales

Elías Todorovich
G. Bioul, M. Tosini
Arquitectura I - Curso 2011
UNICEN



Algebra de Boole

- Un álgebra de Boole es una estructura algebraica con los siguientes elementos:
  (B, +, •, ¬, 0, 1)
- El elemento de identidad (elemento neutro) con respecto a · es 1 y con respecto a + es 0.
- Los operadores · y + son conmutativos.
- · y + son ambos asociativos:
  (AB) C = A (BC)
  (A+B)+C = A+ (B+C).
- · y + son distributivos uno con respecto al otro:
  A· (B+C) = (A·B)+(A·C)
  A+ (B·C) = (A+B) ·(A+C)
- Para cada valor A existe un valor A' (elemento opuesto) tal que:
  A·A' = 0 y
  A+A' = 1. Éste valor es el complemento lógico de A.

George Boole (1815-1864)
"The Mathematical Analysis of Logic" (1847)
"An Investigation of the Laws of Thought" (1854)

Funciones Booleanas

- $f : (B2)^n \rightarrow B2$

  $f(x1, x2, ..., xn)$

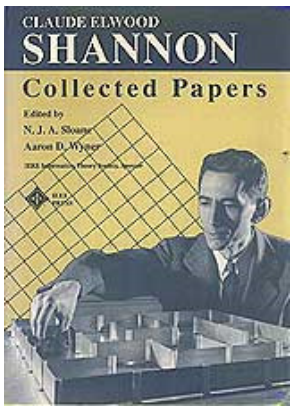
  donde $xi \in \{0, 1\}$

- El conjunto de funciones de $(B2)^n$ en B2 es también un álgebra de Boole.

x1	x2	x3	f(x1, x2, x3)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Del Algebra de Boole a la Electrónica Digital

- En su tesis de maestría en el MIT (1937), *Claude Elwood Shannon*, demostró cómo el álgebra booleana se podía utilizar en el análisis y la síntesis de la conmutación y de los circuitos digitales.
- Algunos consideran que esa tesis es una de las más importantes de la historia. Sin embargo Shannon es más conocido como "el padre de la teoría de la información".

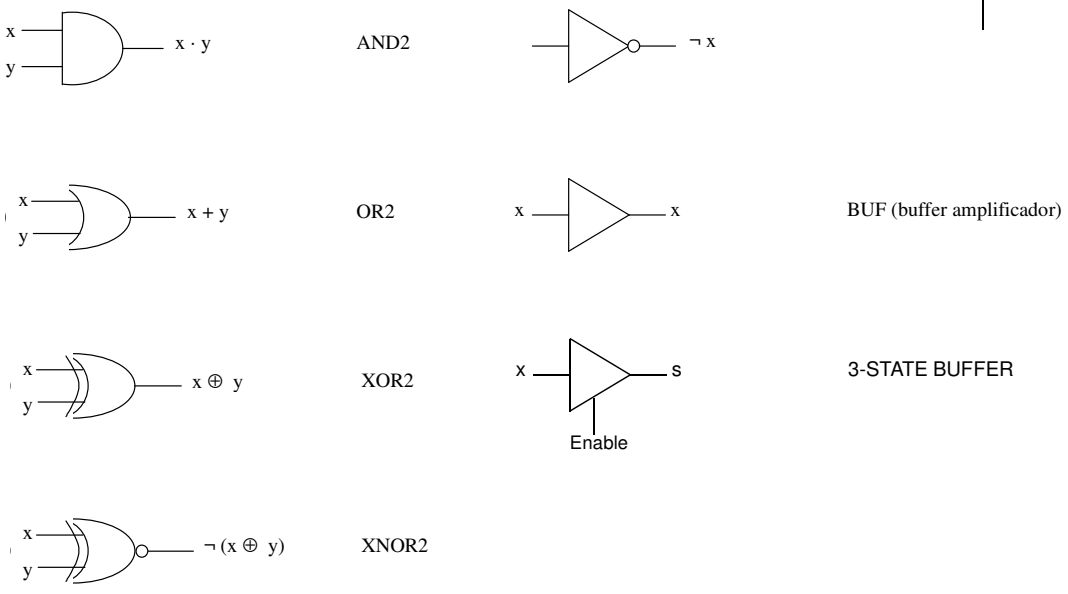


Del Algebra de Boole a la Electrónica Digital

- El cálculo proposicional (CP) y la teoría de conjuntos (TC) son instancias del Algebra de Boole.
- Puertas Lógicas: El Algebra de Boole aplicada a los sistemas digitales tiene las mismas tablas de verdad que en el CP cambiando:
  - Disyunción por OR
  - Conjunción por AND
  - Negación por NOT
- Convención -es una posible:
  - Tensión alta = 1 lógico
  - Tensión baja = 0 lógico
- Funciones Booleanas: Combinando puertas lógicas se pueden construir circuitos lógicos que corresponden a funciones booleanas.

El uso de las puertas lógicas para procesar proposiciones lógicas es el concepto básico que sustenta las computadoras digitales.

Del Algebra de Boole a la Electrónica Digital



Operación Or-Exclusivo en el álgebra de Boole

- **XOR - ⊕ :** $x \oplus y = \neg x \cdot y + x \cdot \neg y$
- Propiedades:

  $x \oplus (y \oplus z) = (x \oplus y) \oplus z$ (asociativo)

  $x \oplus y = y \oplus x$ (conmutativo)

  $x \cdot (y \oplus z) = x \cdot y \oplus x \cdot z$ (distributivo respecto de .)

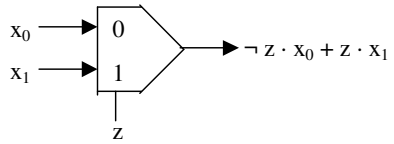
  $x \oplus 0 = x$

  $x \oplus x = 0$

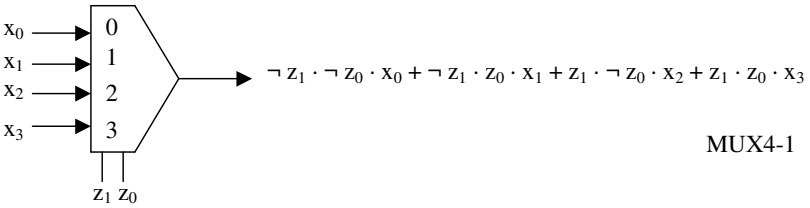
  $x \oplus 1 = \neg x$

  si $x \oplus y = 0 \Rightarrow x = y$

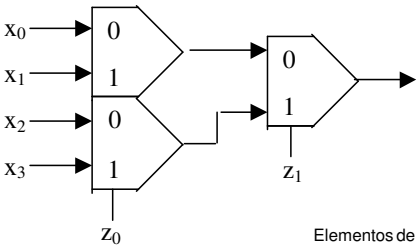
Del Algebra de Boole a la Electrónica Digital



MUX2-1

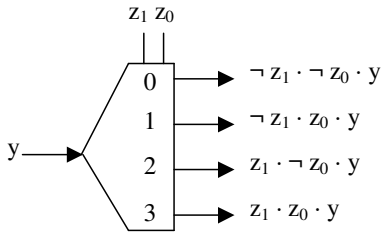


MUX4-1

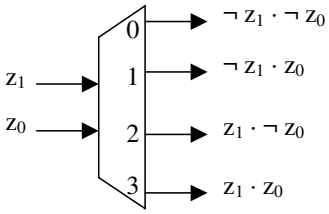


equivalente a MUX4-1

Del Algebra de Boole a la Electrónica Digital



DEMUX1-4



DECODER2-4

Síntesis Lógica

- La síntesis lógica es un proceso por el que se obtiene un circuito en términos de puertas lógicas a partir de una descripción abstracta:
  - Tabla de verdad, ecuación lógica, etc.
  - Descripción RTL en VHDL, Verilog, etc.

Generación de una expresión a partir de una tabla de verdad

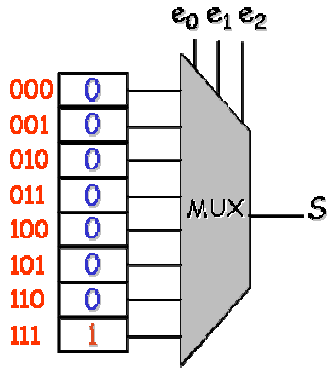
- Una expresión se dice canónica si cada uno de sus términos hace referencia a todas las variables que intervienen en la expresión, ya sea en forma directa o negada.
- Las expresiones canónicas *conjuntivas* son sumas de términos conjuntivos (también llamados minterms).

$Y = \neg X2 \cdot \neg X1 \cdot X0 +$
$\neg X2 \cdot X1 \cdot \neg X0 +$
$X2 \cdot \neg X1 \cdot \neg X0$
cada uno es un minterm.

x1	x2	x3	f(x1, x2, x3)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

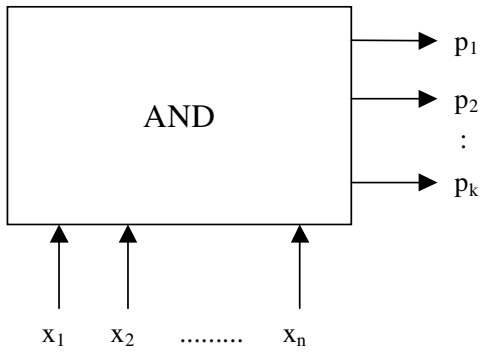
Síntesis mediante multiplexores

- Se emplea un multiplexor con tantas entradas de control como variables de entrada tenga el circuito.
- Cada una de las entradas del MUX (2^n) se conectan a '0' ó a '1' según corresponda al valor de la salida.



Planos de Puertas

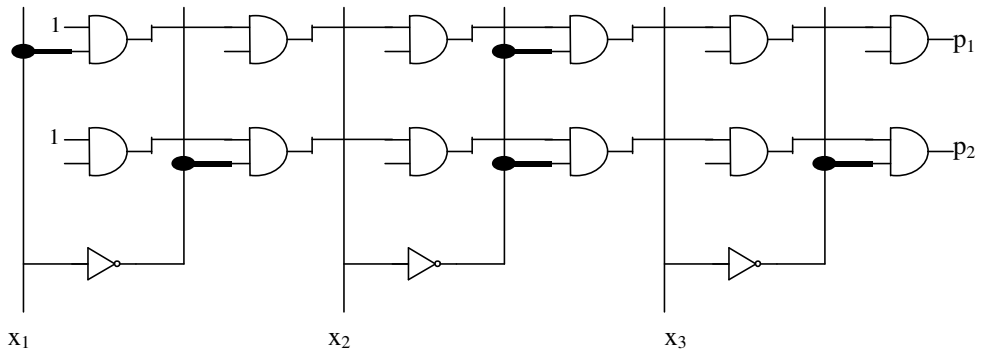
- Para el plano AND, la matriz de compuertas puede tener hasta Xn entradas y hasta pk salidas, donde cada salida pi es el producto de entradas X pudiendo estas estar en forma normal (Xi) o negada (¬ Xi).



Planos de Puertas

- Ejemplo con n = 3 y k = 2:
- Queremos que

p1 = x1 · ¬ x2 p2 = ¬ x1 · ¬ x2 · ¬ x3



Planos de Puertas

- La programación del plano AND se realiza a través de una cadena de bits de tamaño

  2 * #x *#p.

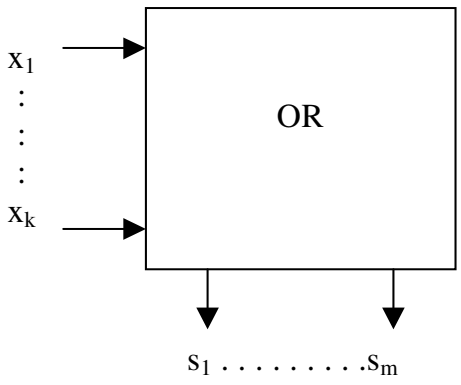
- En donde se representa con un ‘1’ cada posición del plano que debe ser conectada.
- Para el ejemplo:

  1 0 0 1 0 0 -> x1 · ¬ x2

  0 1 0 1 0 1 -> ¬ x1 · ¬ x2 · ¬ x3

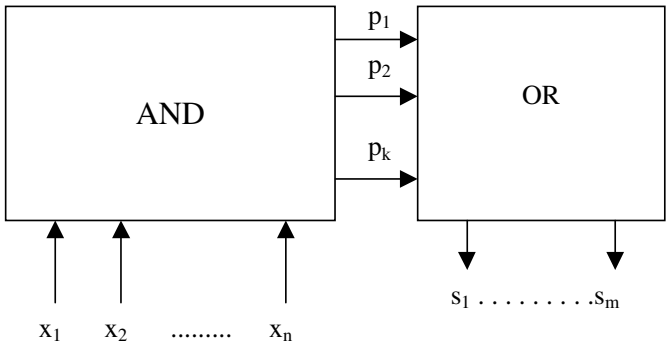
Planos de puertas

- En el plano OR, que es una matriz de compuertas OR interconectadas con k entradas –x1, x2, ..., xk- y m salidas –s1, s2, ..., sm-, cada s_i es la suma (booleana) de algunas variables de entrada, *siempre en forma normal*.



Planos de puertas

- El plano OR en conjunción con un plano AND permite sintetizar funciones booleanas completas mediante componentes denominados PLA (Programmable Logic Array's).



Síntesis mediante Planos de Puertas

- Ejemplo de dos funciones y1 e y2:

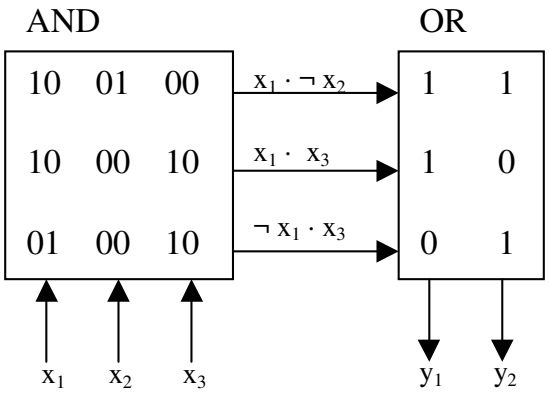
  $y1 = x1 \cdot \neg x2 + x1 \cdot x3$

  $y2 = x1 \cdot \neg x2 + \neg x1 \cdot x3$
- En el plano AND deben implementarse los términos:

  $x1 \cdot \neg x2$

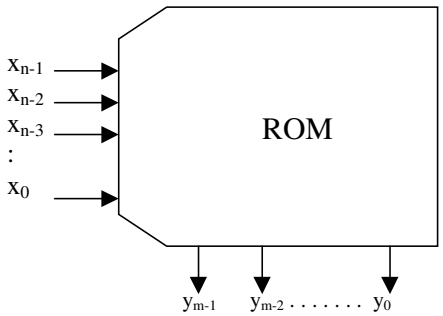
  $x1 \cdot x3$

  $\neg x1 \cdot x3$



Memorias ROM

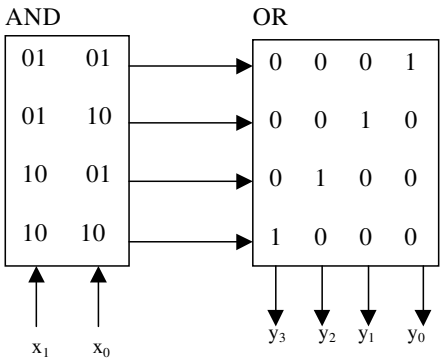
- Una memoria ROM es un dispositivo totalmente cableado. Ante cada combinación de valores de los xi da como resultado un determinado valor en cada salida yi.
- Las ROM son compatible con un plano AND-OR con tantas salidas (yi) como ancho de la celda de datos, y con *n* señales de entrada (xi).



2^n palabras de m bits de ancho

Memorias ROM

- La figura siguiente muestra un ejemplo de implementación de una memoria ROM pasiva de cuatro celdas (n = 2) de 4 bits cada una (m = 4).



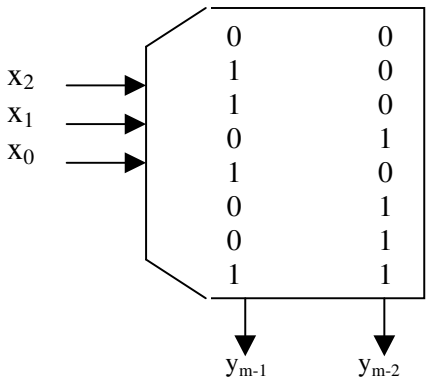
Memorias ROM

- El proceso de síntesis consiste en compilar (en un compilador de circuitos digitales) la secuencia de bits del plano OR.
- La secuencia de bits del plano AND es fija: secuencia de valores de las señales xi desde todos ‘0’ hasta todos ‘1’ (decodificador de direcciones).
- Por ejemplo, para una ROM de 4 palabras de 8 bits cada una con los valores:

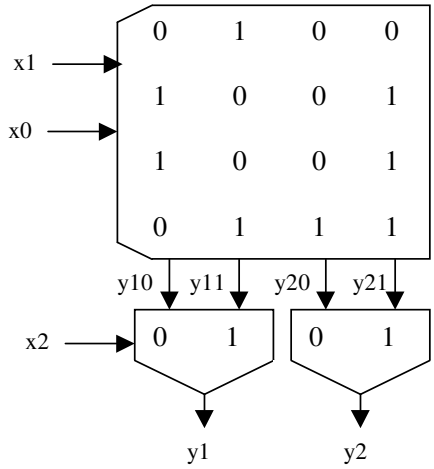
Dirección	Valor
0	AAh
1	FFh
2	03h
3	54h

- Plano AND :
  01 01
  01 10
  10 01
  10 10
- Plano OR :
  10101010
  11111111
  00000011
  01010100

Síntesis mediante memoria ROM



Síntesis mediante ROM+MUX



y11 = y1(x2 = 1)
y10 = y1(x2 = 0)
y21 = y2(x2 = 1)
y20 = y2(x2 = 0)

Generación de una expresión a partir de una tabla de verdad

- Una expresión canónica conjuntiva

$$Y = \neg X2 \cdot \neg X1 \cdot X0 + \neg X2 \cdot X1 \cdot \neg X0 + X2 \cdot \neg X1 \cdot X0$$

¿Se puede minimizar haciendo operaciones permitidas?

x2	X1	X0	f(x0, x1, x2)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Síntesis Lógica

- Técnicas y herramientas de minimización
  - Para tablas de verdad (circuitos combinacionales)
    - Mapas de Karnaugh (1953).
    - Algoritmo de Quine–McCluskey (1956).
    - Espresso heuristic logic minimizer (Brayton et al., 1984).

Mapas de Karnaugh

- Las simplificaciones que permiten los diagramas de Karnaugh se basan en la siguiente identidad:

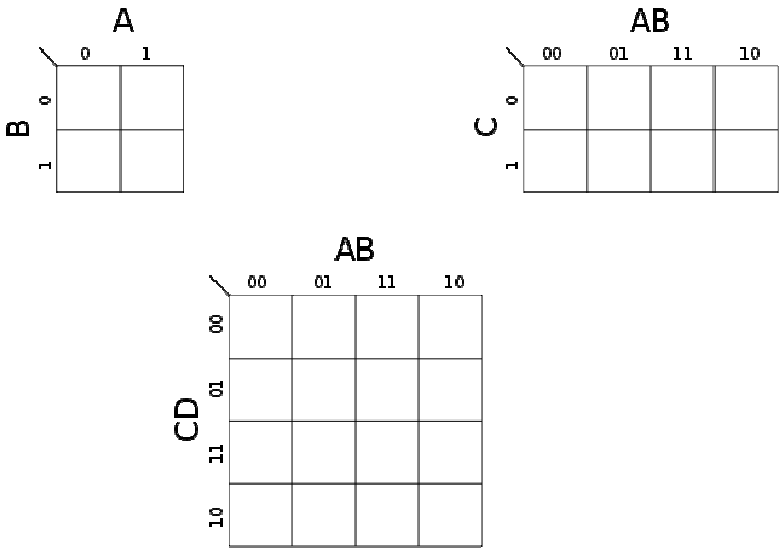
$$A \cdot B \cdot C + A \cdot B \cdot \neg C = A \cdot B \cdot (C + \neg C) = A \cdot B$$

- La ecuación anterior indica que si una variable (la C) aparece negada en un término y no negada en otro que tiene el resto de las variables iguales, puede eliminarse por completo. Los diagramas de Karnaugh ayudan mucho a la localización de estas variables que se pueden suprimir.

Mapas de Karnaugh

- Método manual, gráfico, útil hasta 5 o 6 variables.
- Las variables se ordenan de acuerdo al código de Gray (una sola variable cambia de polaridad entre cuadros adyacentes).
- Para obtener la expresión minimizada:
  - Agrupar los ‘1’ en rectángulos que contengan 1, 2, 4, 8 unos.
  - Cada casilla con un ‘1’ se debe cubrir al menos una vez.
  - Los valores indefinidos (don’t care) pueden aprovecharse para hacer grupos de ‘1’ mas grandes.

Mapas de Karnaugh



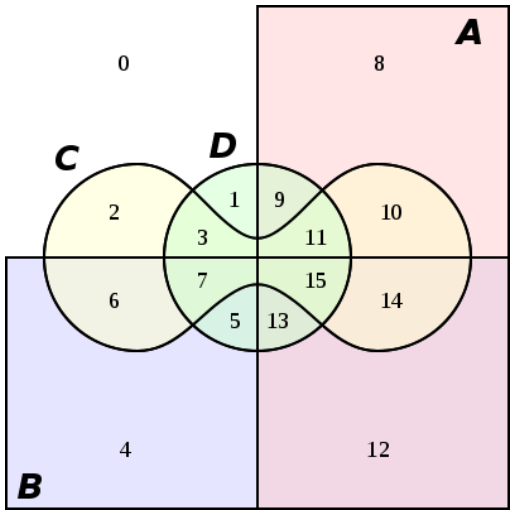
Mapas de Karnaugh

X3 X2 \ X1 X0	00	01	11	10
00	1	1	0	0
01	0	0	0	0
11	1	0	0	1
10	1	1	1	1

$$\neg X2 \cdot \neg X1 + X3 \cdot \neg X0 + X3 \cdot \neg X2$$

Mapas de Karnaugh

- Un mapa de Karnaugh se puede ver como un diagrama de Venn especialmente dibujado:



Mapas de Karnaugh

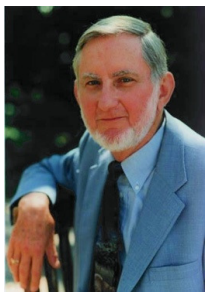
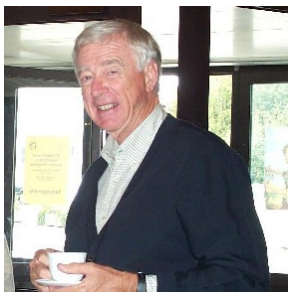
- Al ser el Algebra de Boole un sistema dual, siempre se podrá pensar en la forma dual del método agrupando ‘0’s.

Algoritmo de Quine–McCluskey

- Si bien el algoritmo de Quine–McCluskey se puede implementar en un programa, su complejidad temporal y espacial no son buenas: $3^n/n$ para una función de n variables.
- Resulta que el problema de la minimización es NP-hard y para resolver problemas de muchas variables se **deben** usar métodos heurísticos sub-óptimos.
- Elementos adicionales en la síntesis y minimización de funciones:
  - Funciones no completamente especificadas;
  - Síntesis simultánea de múltiples funciones.

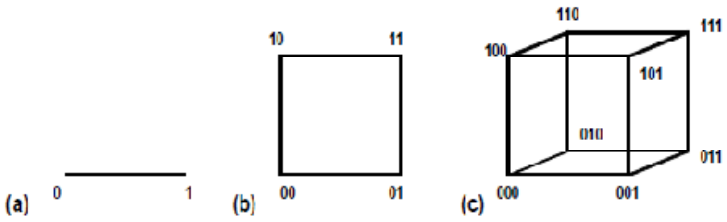
Espresso

- Espresso es el algoritmo estándar de síntesis automática de funciones booleanas.

Espresso

- Tipos de datos: una función booleana de n entradas se puede ver en un espacio n-dimensional con un eje para cada entrada.



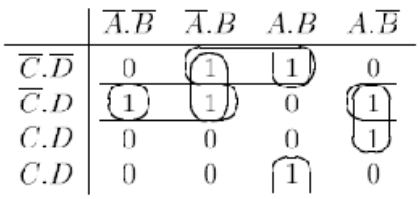
- Una función normalmente ocupara un número de vértices en un cubo n-dimensional.

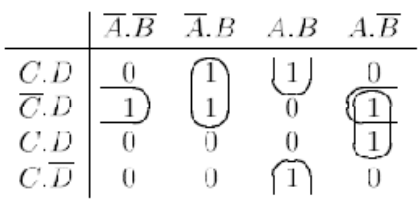
Espresso

- Se pueden seleccionar cubos de menor dimensión que cubran los vértices (minterms) ocupados.
- Un cubo *primo* es aquel que no se puede expandir sin incluir vértices con ‘0’.
- Algoritmo: Espresso es un algoritmo greedy que aplica 3 operaciones iterativamente:
  - Expand
  - Irredundant_cover
  - Reduce

Espresso

- Expand:
  - Los cubos se expanden hasta hacerlos *primos*.
  - El resultado es una cobertura de la función por cubos primos (los cubos no contienen otros primos.)
  - Sin embargo esta operación no basta para encontrar las mejores soluciones.





Espresso

- Irredundant_cover:
  - Se trata de reducir el número de cubos de manera que no haya cobertura redundante. Es lo que se hizo en el ejemplo de la página anterior.
- Reduce:
  - Trata de reemplazar cada cubo por uno menor contenido dentro de él.
- Estas 3 operaciones se repiten comenzando desde diferentes puntos de partida hasta que no se obtengan mejores soluciones.

Espresso

- Espresso esta disponible en la web como código open-source:
  - http://diamond.gem.valpo.edu/~dhart/ece110/espresso/tutorial.html Programa para bajar más instrucciones sobre cómo operarlo.
  - http://embedded.eecs.berkeley.edu/pubs/downloads/espresso/index.htm El código fuente original se puede bajar de la página de Berkeley.

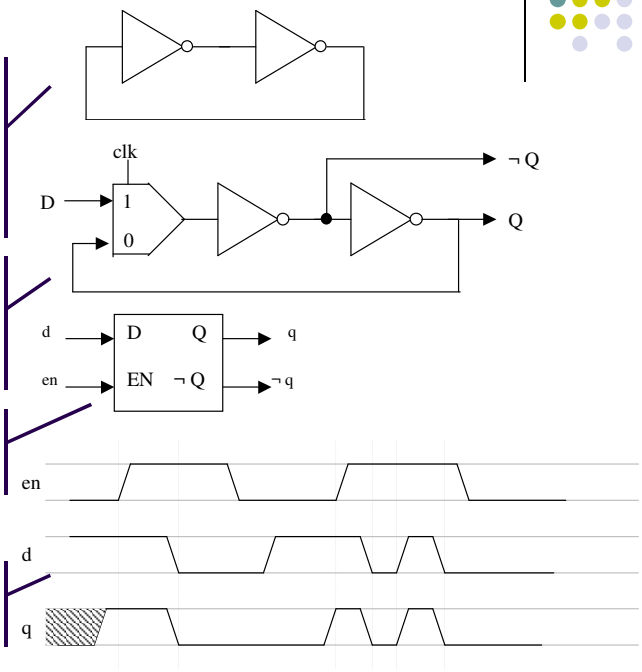
Síntesis de circuitos secuenciales

El elemento de memoria más simple es el biestable, que consta de dos inversores interconectados.

El estado del biestable puede controlarse a través de puertas.

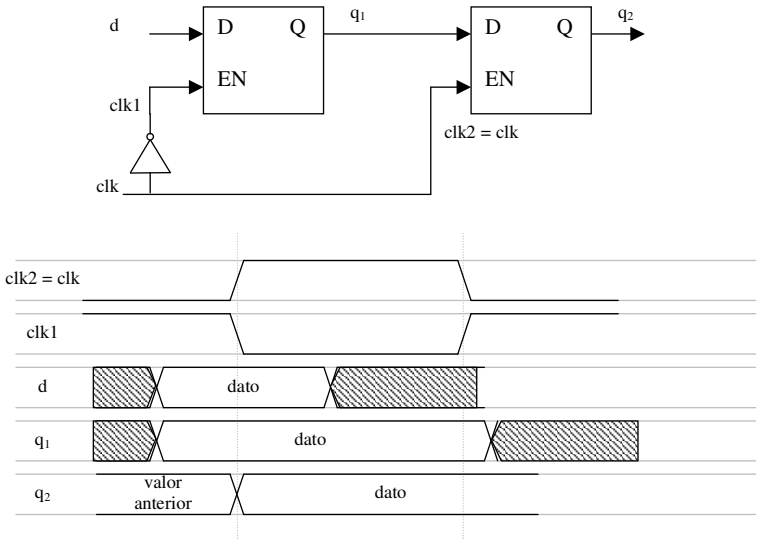
Símbolo esquemático

Cronograma del D-Latch

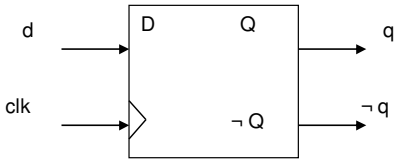


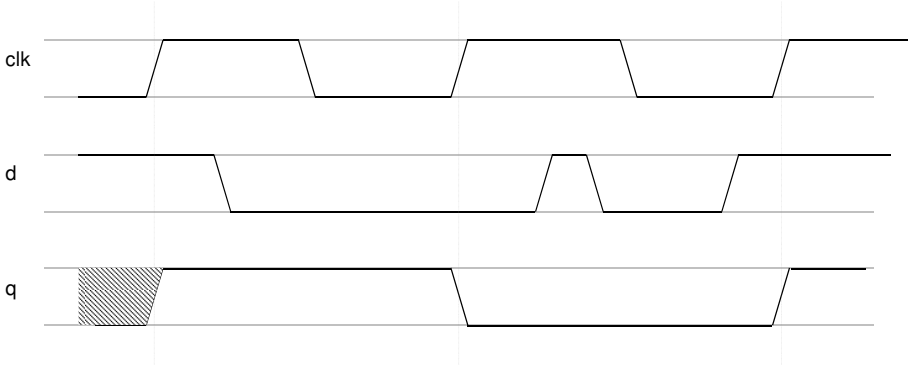
Flip-flop D

- El flip-flop D es equivalente a disponer dos básculas D una a continuación de la otra y sincronizadas de manera que se activen en estados opuestos del reloj.

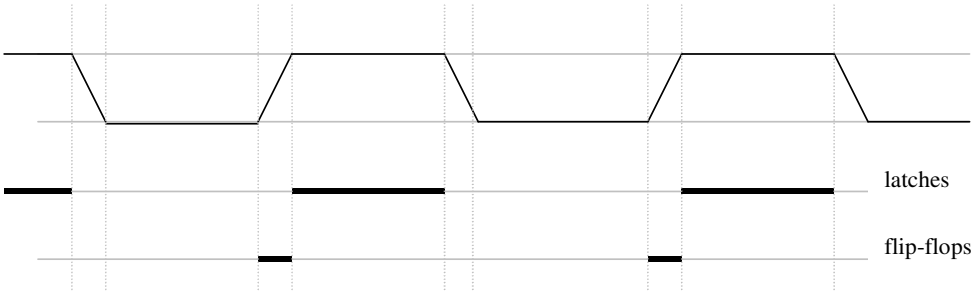


Flip-flop D



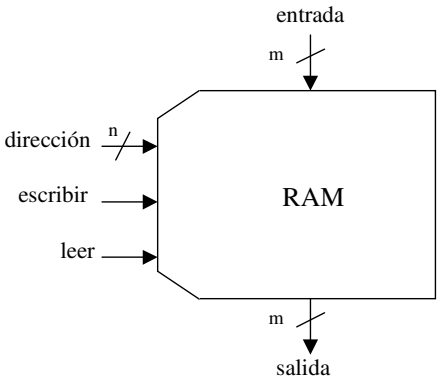


Latch vs. Flip-flop



Zona activa de latches y flip-flpos

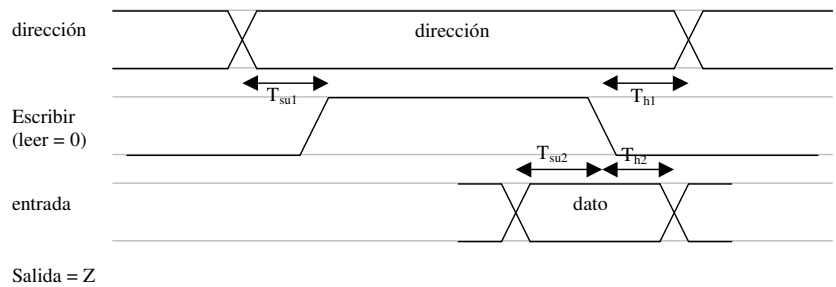
Memoria Activa o RAM



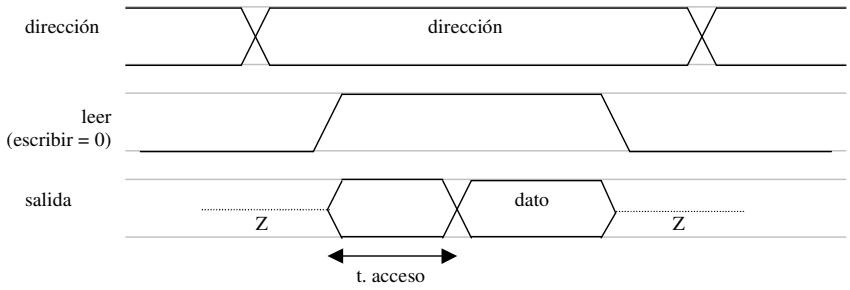
2^n palabras de m bits de ancho

Memoria Activa o RAM

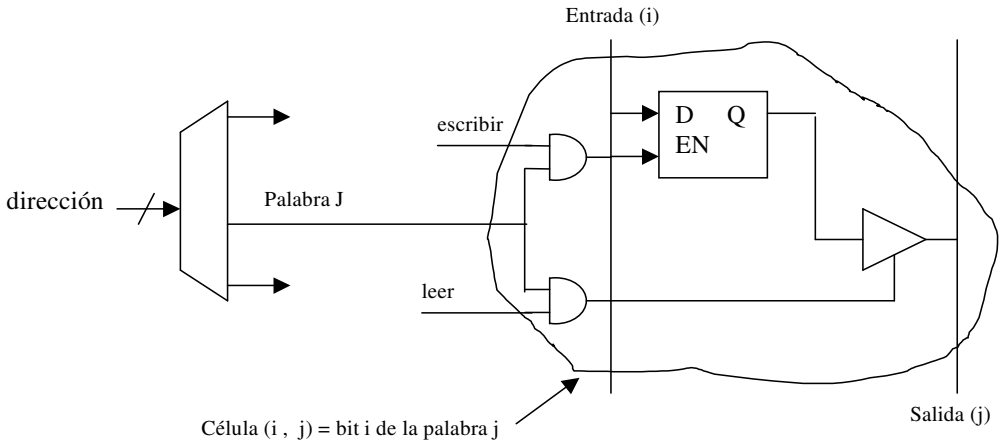
- Escritura



- Lectura

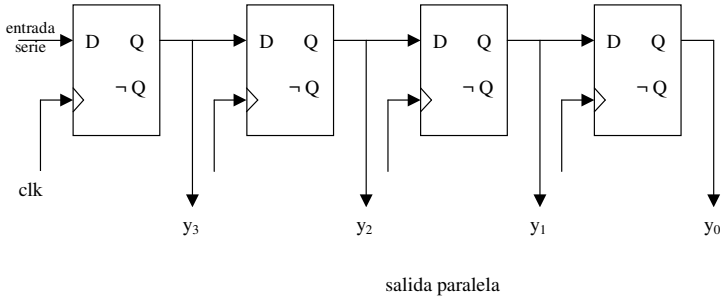


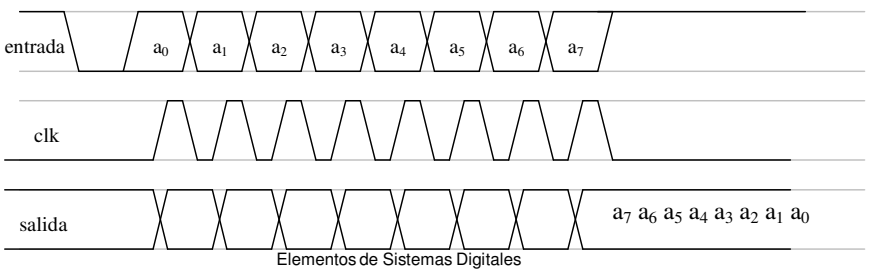
Memoria Activa o RAM



Registros de desplazamiento

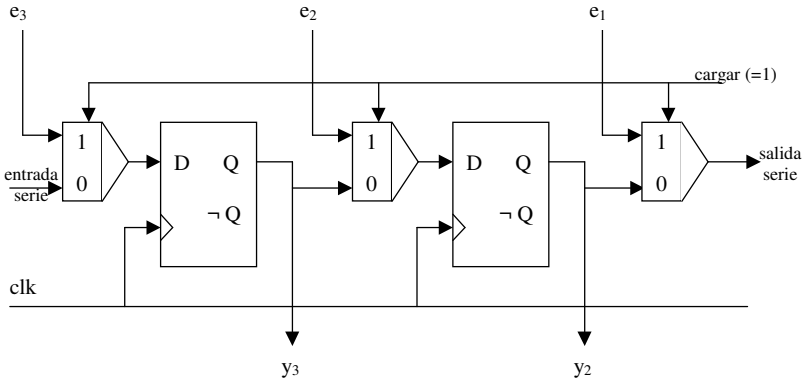
- La aplicación más directa de este tipo de registros es la conversión serie-paralelo:



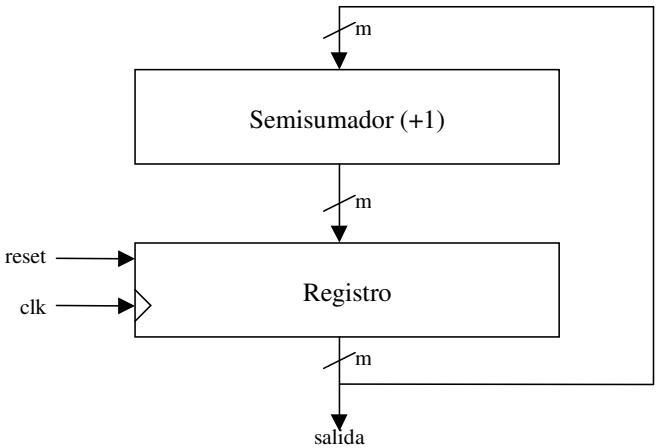


Registro de desplazamiento con entrada paralela

- La aplicación más difundida se da UsART's (transmisores/receptores sincrónicos/asincrónicos universales) usados para comunicaciones seriales.

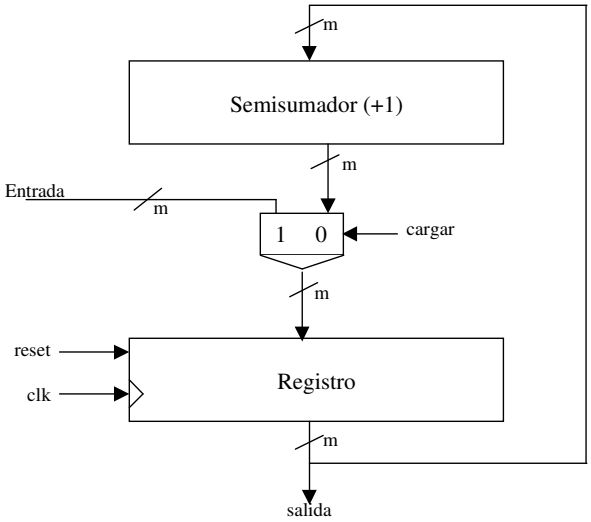


Contadores síncronos con sumadores y registros



Contador binario programable

- Este contador binario programable es igual o similar el contador de programa (PC) de los procesadores actuales.



Elementos de Diseño de Sistemas Digitales

Elías Todorovich
G. Bioul, M. Tosini

Arquitectura I - Curso 2011

UNICEN