Ciclos Formativos de Grado Medio

# Electrónica

## Capítulo 1: Sistemas Digitales

Marcos García, Pablo Huerta, Carlos Sánchez, Pablo Toharia

Ra-Ma®

# Índice



Marcos García, Pablo Huerta, Carlos Sánchez, Pablo Toharia

Ra-Ma®

# Índice



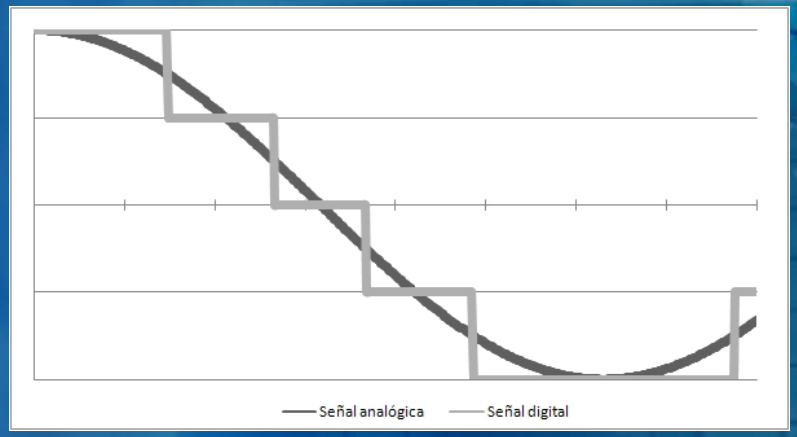
Marcos García, Pablo Huerta, Carlos Sánchez, Pablo Toharia

Ra-Ma®

# Introducción a los sistemas digitales

- **Magnitud:** propiedad física que puede medirse cuantitativamente.
- Dependiendo de la naturaleza de las mismas podemos dividirlas
  - Analógicas: toman valor en un rango continuo.
  - Digitales: toman valor en un rango discreto.

- **Señal**: evolución en el tiempo de una magnitud.




Marcos García, Pablo Huerta, Carlos Sánchez, Pablo Toharia

Ra-Ma®

# Introducción a los sistemas digitales

- Los sistemas digitales procesan información digital.

- Ventajas:
  - Facilidad de diseño.
  - Menor sensibilidad a ruidos.
  - Tolerancia a fallos.
  - Facilidad de almacenamiento.
- Desventajas:
  - Las mayoría de las magnitudes naturales son analógicas (conversores A/D y D/A).

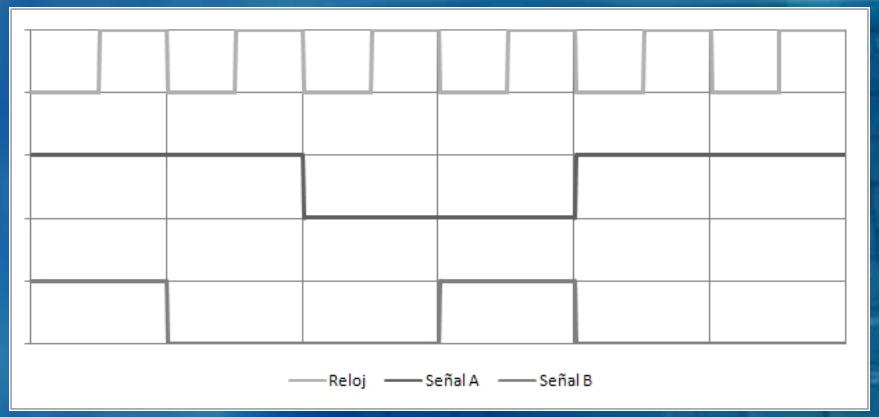
Marcos García, Pablo Huerta, Carlos Sánchez, Pablo Toharia

Ra-Ma®

# Introducción a los sistemas digitales

- Los sistemas digitales actuales procesan información digital **BINARIA** (0 y 1).
- La información se codifica con valores de tensión (VL y VH).

- Los sistemas se analizan mediante cronogramas.

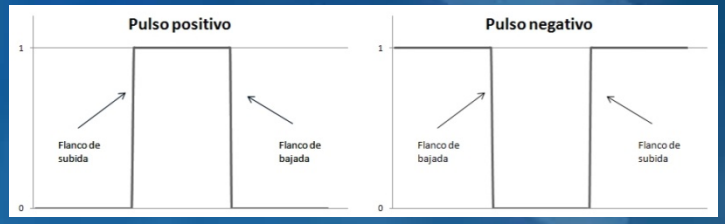



Marcos García, Pablo Huerta, Carlos Sánchez, Pablo Toharia

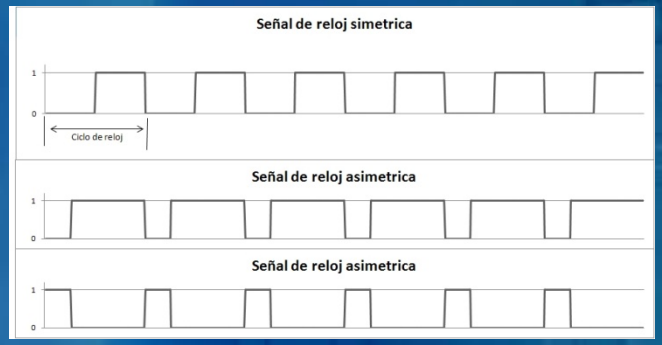
Ra-Ma®

# Introducción a los sistemas digitales

- Flanco: transición entre dos niveles.
- Pulso: la señal entre dos flancos.




- Señal periódica que se utiliza para sincronizar sistemas digitales.




Marcos García, Pablo Huerta, Carlos Sánchez, Pablo Toharia

Ra-Ma®

# Índice



Marcos García, Pablo Huerta, Carlos Sánchez, Pablo Toharia

Ra-Ma®

# El álgebra de Boole

- **Álgebra de Boole:** formalismo matemático utilizado en el análisis y la descripción de circuitos digitales.
- Definición formal del álgebra de Boole a partir de sus elementos:
  - ∀ a ∈ B | a = 0 ó a = 1
  - Operación complemento (’):
    - a’ = 1 → a = 0
    - a’ = 0 → a = 1
  - Operación producto lógico (*):
    - a = 0 y b = 0 → a*b = 0
    - a = 0 y b = 1 → a*b = 0
    - a = 1 y b = 0 → a*b = 0
    - a = 1 y b = 1 → a*b = 1

- Operación suma lógica (+):
  - a = 0 y b = 0 → a+b = 0
  - a = 0 y b = 1 → a+b = 1
  - a = 1 y b = 0 → a+b = 1
  - a = 1 y b = 1 → a+b = 1

Marcos García, Pablo Huerta, Carlos Sánchez, Pablo Toharia

# El álgebra de Boole

- Expresiones lógicas:
  - F(a, b) = a+b' → F(0, 1) = 0+1' = 0+0 = 0
  - F(a, b) = (a+b')*a → F(1, 1) = (1+1')*1 = (1+0)*1 = 1*1 = 1
  - F(a,b,c) = a'*b+c → F(0,0,1) = 0'*0+1 = 1*0+1 = 0+1 = 1
  - ….

# El álgebra de Boole

- Propiedades del álgebra de Boole
  - Propiedad conmutativa:
    - a + b = b + a
    - a * b = b * a
  - Propiedad asociativa:
    - (a + b) + c = a + (b + c)
    - (a * b) * c = a * (b * c)
  - Propiedad distributiva:
    - a + (b * c) = (a + b) * (a + c)
    - a * (b + c) = (a * b) + (a * c)
  - Elemento neutro de la suma lógica:
    - a + 0 = a
    - a * 1 = a
  - Idempotencia:
    - (a')' = a

# El álgebra de Boole

- Teoremas
  - Teorema de identidad:
    a + a' = 1
    a * a' = 0
  - Teorema de idempotencia:
    a + a = a
    a * a = a
  - Teorema de la identidad del 1 y del 0:
    a + 1 = 1
    a * 0 = 0
  - Elemento neutro:
    a + 0 = a
    a * 1 = a
  - Teoremas de absorción:
    a + a * b = a
    a + a' * b = a + b
    a * (a + b) = a
    a * (a' + b) = a * b

Marcos García, Pablo Huerta, Carlos Sánchez, Pablo Toharia

Ra-Ma®

# Índice

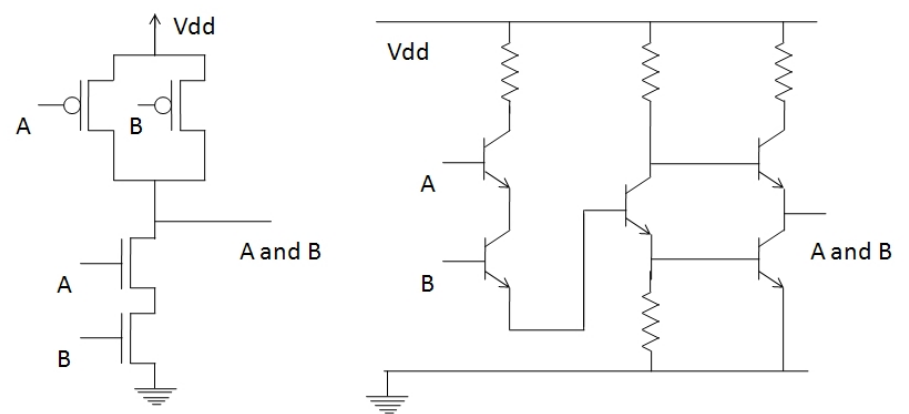


Marcos García, Pablo Huerta, Carlos Sánchez, Pablo Toharia

Ra-Ma®

# Puertas lógicas

- Implementaciones de operadores lógicos y funciones lógicas sencillas:
  - Transistores de efecto campo.
  - Transistores bipolares.



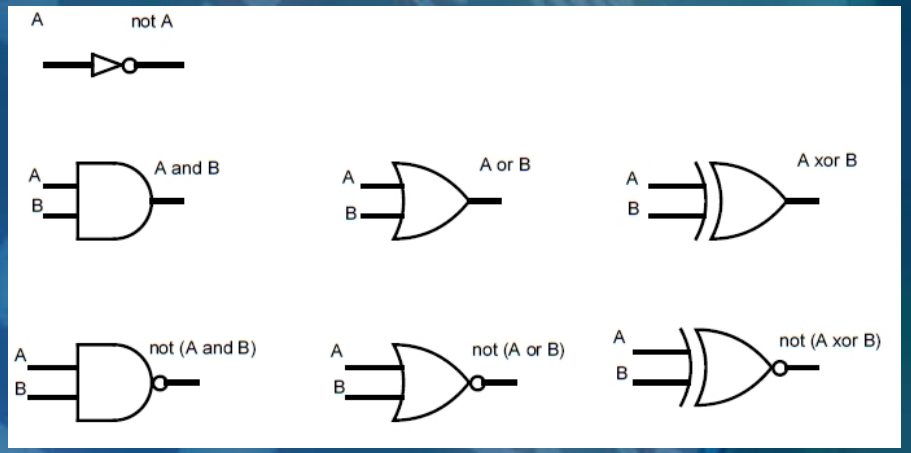

Puerta AND implementada con tecnología CMOS

Puerta AND implementada con tecnología TTL

Marcos García, Pablo Huerta, Carlos Sánchez, Pablo Toharia

Ra-Ma®

# Puertas lógicas

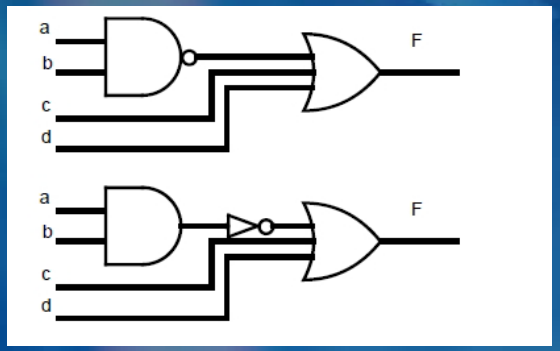


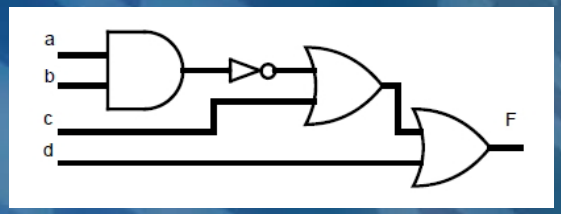

- Caracterizadas por:
  - La función lógica que realizan.
  - El número de entradas.
- Puertas lógicas
  - Puerta O o puerta OR: OR(a, …, n) = a +…+ n.
  - Puerta Y o puerta AND: AND(a, …, n) = a *…* n.
  - Puerta O Negado o puerta NOR: NOR(a, …, n) = (a +…+ n)'.
  - Puerta Y Negado o NAND: NAND(a, …, n) = (a *…* n)'.
  - Puerta O Exclusivo o XOR: XOR(a, b,…, n) = a * b' *…* n' + a' * b*…* n' + a' * b' *…* n.
  - Puerta O Exclusivo Negado o NXOR: XOR(a, b,…, n) = (a * b' *…* n' + a' * b*…* n' + a' * b' *…* n)'.
- Tiene un símbolo asociado que nos permite definir circuitos mediante diagramas de flujo.

Marcos García, Pablo Huerta, Carlos Sánchez, Pablo Toharia

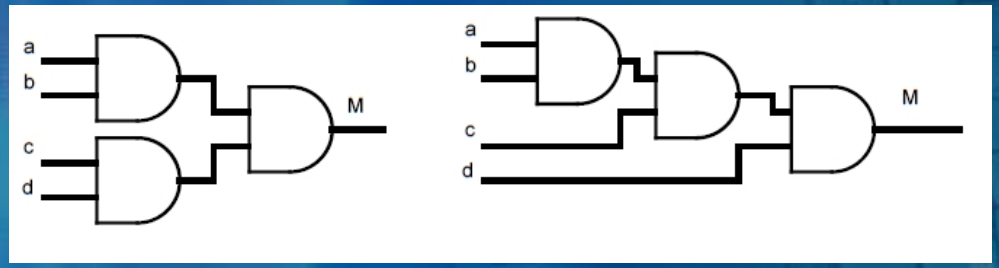
# Puertas lógicas

- F(a, b, c, d) = (a * b)’ + c + d




- M(a, b, c, d) = (a * b) * (c * d) = a * (b * (c * d))




Marcos García, Pablo Huerta, Carlos Sánchez, Pablo Toharia

# Representación de funciones lógicas

- Mediante expresiones booleanas
  - No son únicas.
  - Nos permiten operar de forma que se ajusten a nuestras necesidades
    - Minimizar el número de puertas lógicas.
    - Utilizar sólo puertas lógicas de un tipo.
    - Eliminar rebotes o gliches.
    - …
  - Su diseño *ad-hoc* no es sencillo si la función lógica es complicada.
- Tablas de verdad
  - Únicas.
  - Se pueden obtener fácilmente cuando se desconoce la expresión booleana a partir de una descripción en lenguaje natural.
- Expresiones booleanas en forma normal
  - Únicas.
  - Se pueden obtener fácilmente a partir de una tabla de verdad.

# Representación de funciones lógicas

- Tablas de verdad
  - Indican el valor que debe tomar cada salida para todas las combinaciones de las entradas posibles.
  - Pueden obtenerse a partir de una función lógica o a partir de una descripción detallada en lenguaje natural.

| Entradas | | | Valores intermedios | | Salida | Valores intermedios | | Salida |
|---|---|---|---|---|---|---|---|---|
| a | b | c | b' | b'*c | *a + b' * c* | *(a+b')* | *(a +c)* | *(a+b') * (a+c)* |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |

*F(a, b, c) = a + b' * c = (a + b') * (a + c)*

| Entradas | | | | Salida | Entradas | | | | Salida |
|---|---|---|---|---|---|---|---|---|---|
| A1 | A2 | A3 | A4 | F | A1 | A2 | A3 | A4 | F |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |

Definir un sistema que tomará el valor 1 cuando hubiese un número par de unos en los cuatro bits de entrada

# Representación de funciones lógicas

- Expresiones booleanas en forma normal
  - Se obtienen fácilmente a partir de una tabla de verdad.
  - Tipos
    - Primera forma normal, primera forma canónica o forma normal disyuntiva.
    - Segunda forma normal, segunda forma canónica o forma normal conjuntiva.
  - Primera forma normal
    - Representa la función con una suma de minitérminos
      - Minitermino: producto compuesto por todas las entradas de un sistema, negadas o sin negar.
    - Se construye a partir de la tabla de verdad
      - Identificando las combinaciones de los valores de las entradas que hacen 1 la salida.
      - Por cada una de esas combinaciones se añade un minitérmino.
      - El minitérmino se compone por la variable negada si la entrada toma el valor 0 y por la variable sin negar si ésta toma el valor 1.
  - Segunda forma normal
    - Representa la función con un producto de maxitérminos
      - Maxitermino: suma compuesta por todas las entradas de un sistema, negadas o sin negar.
    - Se construye a partir de la tabla de verdad
      - Identificado las combinaciones de los valores de las entradas que hacen 0 la salida.
      - Por cada una de esas combinaciones se añade un maxitérmino.
      - El maxitérmino se compone por la variable negada si la entrada toma el valor 1 y por la variable sin negar si ésta toma el valor 0.

# Representación de funciones lógicas

| a | b | c | Minitérmino | Expresión | Maxitérmino | Expresión |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | $m_0$ | a' * b' * c' | $M_0$ | a + b + c |
| 0 | 0 | 1 | $m_1$ | a' * b' * c | $M_1$ | a + b + c' |
| 0 | 1 | 0 | $m_2$ | a' * b * c' | $M_2$ | a + b' + c |
| 0 | 1 | 1 | $m_3$ | a' * b * c | $M_3$ | a + b' + c' |
| 1 | 0 | 0 | $m_4$ | a * b' * c' | $M_4$ | a' + b + c |
| 1 | 0 | 1 | $m_5$ | a * b' * c | $M_5$ | a' + b + c' |
| 1 | 1 | 0 | $m_5$ | a * b * c' | $M_6$ | a' + b' + c |
| 1 | 1 | 1 | $m_7$ | a * b * c | $M_7$ | a' + b' + c' |

Marcos García, Pablo Huerta, Carlos Sánchez, Pablo Toharia

# Representación de funciones lógicas

- Síntesis con minitérminos

| Entradas | | | | Salida | $m_i$ | Entradas | | | | Salida | $m_i$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| A1 | A2 | A3 | A4 | F | $m_i$ | A1 | A2 | A3 | A4 | F | $m_i$ |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 8 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 9 |
| 0 | 0 | 1 | 0 | 0 | 2 | 1 | 0 | 1 | 0 | 1 | 10 |
| 0 | 0 | 1 | 1 | 1 | 3 | 1 | 0 | 1 | 1 | 0 | 11 |
| 0 | 1 | 0 | 0 | 0 | 4 | 1 | 1 | 0 | 0 | 1 | 12 |
| 0 | 1 | 0 | 1 | 1 | 5 | 1 | 1 | 0 | 1 | 0 | 13 |
| 0 | 1 | 1 | 0 | 1 | 6 | 1 | 1 | 1 | 0 | 0 | 14 |
| 0 | 1 | 1 | 1 | 0 | 7 | 1 | 1 | 1 | 1 | 1 | 15 |

Definir un sistema que tome el valor 1 cuando haya un número par de unos en los cuatro bits de entrada

$F = m(3, 5, 6, 9, 10, 12, 15) =$

$m_3 + m_5 + m_6 + m_9 + m_{10} + m_{12} + m_{15} =$

$a'b'cd + a'bc'd + a'bcd' + ab'c'd + ab'cd' + abc'd' + abcd$

Marcos García, Pablo Huerta, Carlos Sánchez, Pablo Toharia

Ra-Ma®

# Representación de funciones lógicas

- Síntesis con maxitérminos

Definir un sistema que tome el valor 1 cuando haya un número par de unos en los cuatro bits de entrada

| Entradas | | | | Salida | $M_i$ | Entradas | | | | Salida | Mi |
|---|---|---|---|---|---|---|---|---|---|---|---|
| A1 | A2 | A3 | A4 | F | $m_i$ | A1 | A2 | A3 | A4 | F | $m_i$ |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 8 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 9 |
| 0 | 0 | 1 | 0 | 0 | 2 | 1 | 0 | 1 | 0 | 1 | 10 |
| 0 | 0 | 1 | 1 | 1 | 3 | 1 | 0 | 1 | 1 | 0 | 11 |
| 0 | 1 | 0 | 0 | 0 | 4 | 1 | 1 | 0 | 0 | 1 | 12 |
| 0 | 1 | 0 | 1 | 1 | 5 | 1 | 1 | 0 | 1 | 0 | 13 |
| 0 | 1 | 1 | 0 | 1 | 6 | 1 | 1 | 1 | 0 | 0 | 14 |
| 0 | 1 | 1 | 1 | 0 | 7 | 1 | 1 | 1 | 1 | 1 | 15 |

$F = M(0, 1, 2, 4, 7, 8, 11, 13, 14) =$

$M_0 * M_1 * M_2 * M_4 * M_7 * M_8 * M_{11} * M_{13} * M_{14} =$

$(a + b + c + d) * (a + b + c + d') * (a + b + c' + d) * (a + b' + c + d) * (a + b' + c' + d') *$

$(a' + b + c + d) * (a' + b + c' + d') * (a' + b' + c + d') * (a' + b' + c' + d)$

Marcos García, Pablo Huerta, Carlos Sánchez, Pablo Toharia

Ra-Ma®

# Índice

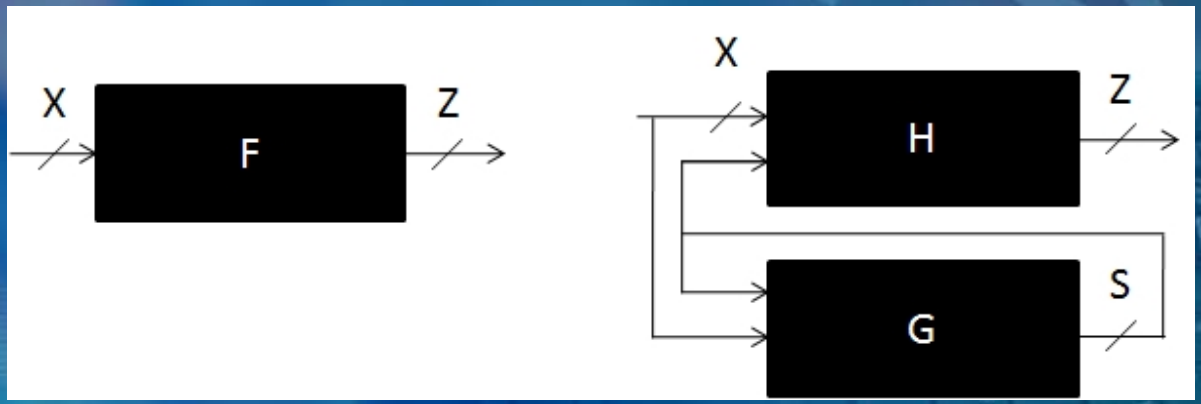


Marcos García, Pablo Huerta, Carlos Sánchez, Pablo Toharia

Ra-Ma®

# Tipos de circuitos lógicos

- Circuitos combinacionales: la salida depende del valor de las entradas.
- Circuitos secuenciales: la salida depende de las entradas y del estado, el estado dependerá del valor de las entradas en todos los instantes anteriores.



# Circuitos combinacionales

- Síntesis de circuitos combinacionales:
  - Definir la tabla de verdad de cada una de las salidas del sistema.
  - Crear una expresión booleana que defina el comportamiento de cada una de las salidas (primera y segunda forma normal).
  - Operar con la función lógica para que se ajuste a los requisitos del sistema.
  - Implementar la función lógica obtenida mediante puertas lógicas.
- Si la complejidad del sistema es elevada, el circuito no se implementa utilizando puertas lógicas
  - El problema se divide en bloques que realizan tareas más o menos simples.
  - Bloques complejos se dividen en bloques más sencillos.

# Circuitos combinacionales

- Bloques combinacionales básicos:
  - Multiplexores
  - Demultiplexores
  - Decodificadores
  - Codificadores
  - Desplazadores
  - Detectores y generadores de paridad
  - Semisumadores y sumadores completos

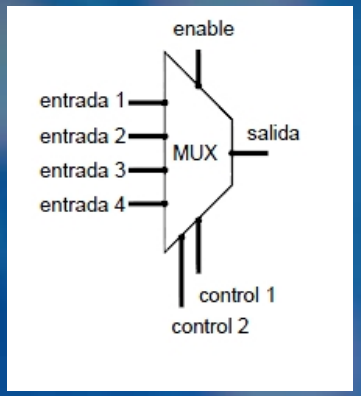
- Dispositivos lógicos programables (PLD)
  - Memorias ROM
  - Dispositivos PLA
  - Dispositivos PAL
  - CPLD
  - FPGA

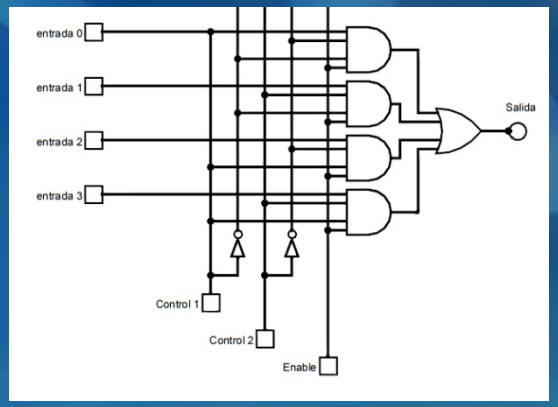
Marcos García, Pablo Huerta, Carlos Sánchez, Pablo Toharia

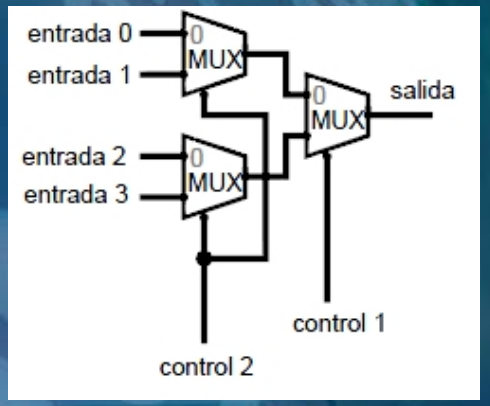
Ra-Ma®

# Circuitos combinacionales

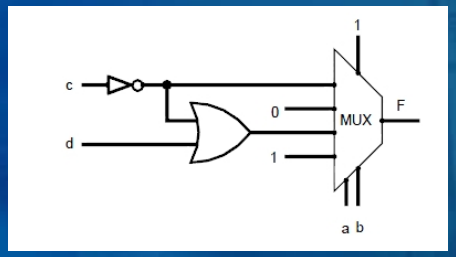
- Multiplexores
  - Selecciona una de las entradas y la pone en la salida.




  - Se pueden sintetizar funciones lógicas a partir de multiplexores.
    - F(a, b, c, d) = a * b + a * b' * (c + d') + a' * b' * d'

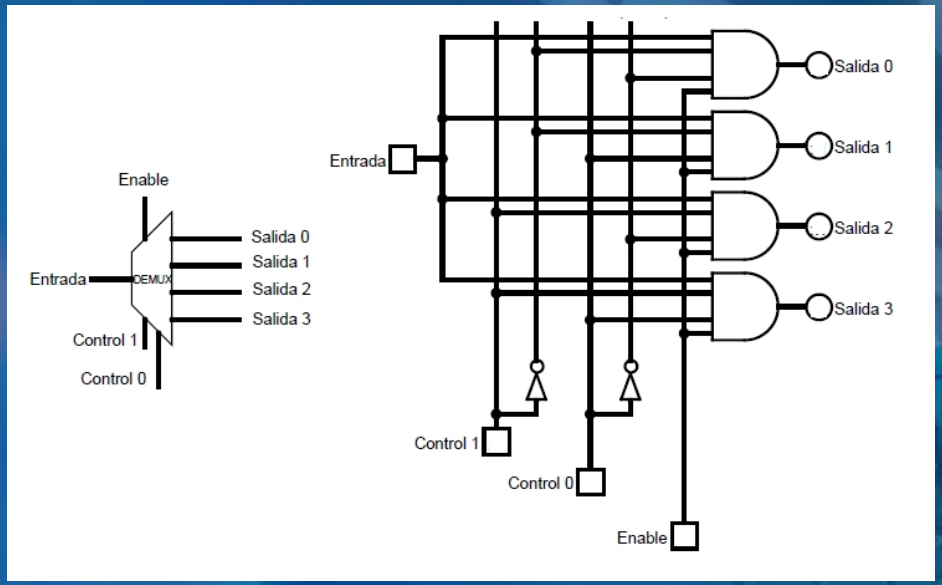



Marcos García, Pablo Huerta, Carlos Sánchez, Pablo Toharia

Ra-Ma®

# Circuitos combinacionales

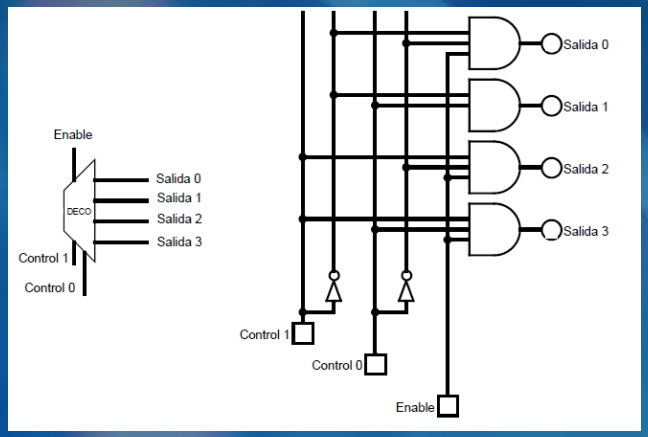
- Demultiplexores
  - Colocan la entrada en la salida escogida.

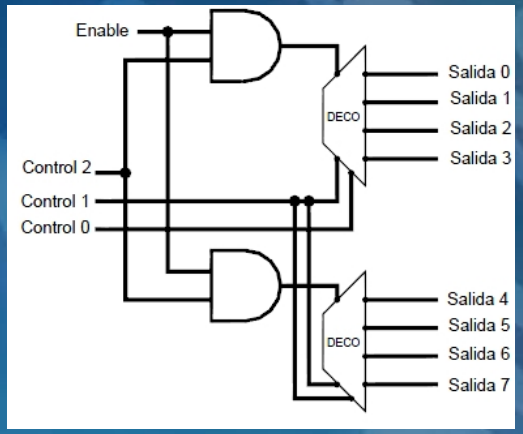



Marcos García, Pablo Huerta, Carlos Sánchez, Pablo Toharia

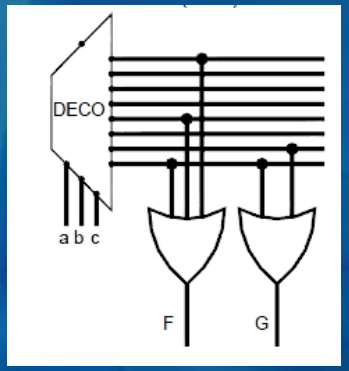
# Circuitos combinacionales

- Decodificadores
  - Activan una de las salidas.



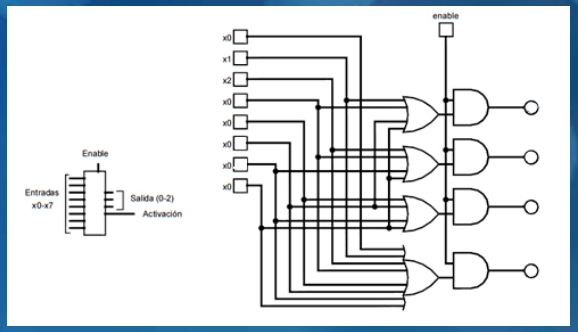

  - Síntesis mediante decodificadores.
    - F(a, b, c)= a * b * c + a * b' * c'+ a' * b' * c' y G(a, b, c)= a * b * c + a * b * c'



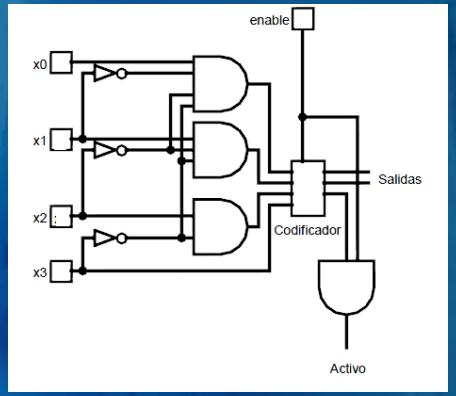

Marcos García, Pablo Huerta, Carlos Sánchez, Pablo Toharia

# Circuitos combinacionales

- Codificadores
  - Indica cual de las entradas está activa.




  - Codificación con prioridad.




Marcos García, Pablo Huerta, Carlos Sánchez, Pablo Toharia

Ra-Ma®