

## 2. Nivel de lógica digital y sistemas digitales

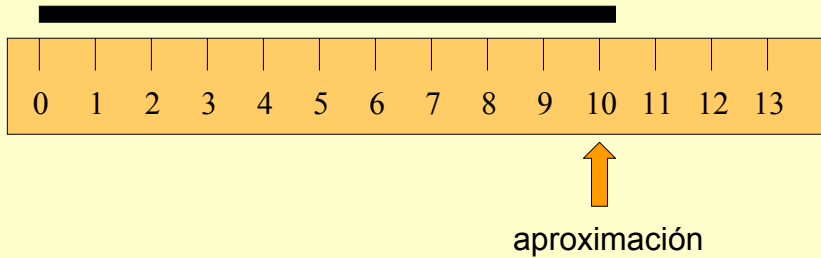
- ❑ **Sistema digital**
- ❑ **Álgebra de Boole y álgebra de Conmutación**
- ❑ **Circuitos combinacionales básicos**
- ❑ **Circuitos secuenciales básicos**
- ❑ **Controladores**

### Sistema digital

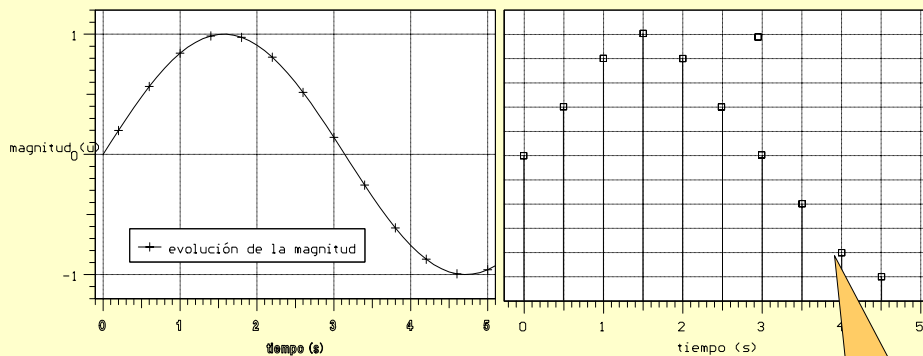
### *Información analógica y digital*

- ❑ La información se encuentra en la forma (aspecto) de las cosas
- ❑ Al medir un aspecto de un fenómeno físico obtenemos una **magnitud**
  - Magnitud → número real  
(magnitud analógico)
  - Si el aspecto cambia con el tiempo, la magnitud también cambia.
  - Las computadoras trabajan con magnitudes discretas

- ❑ Discretización:  
"obtención de magnitudes discretas a partir de magnitudes analógicas"
- ❑ Ejemplo, en una dimensión



Magnitud analógica

1º Muestreo +  
2º cuantización

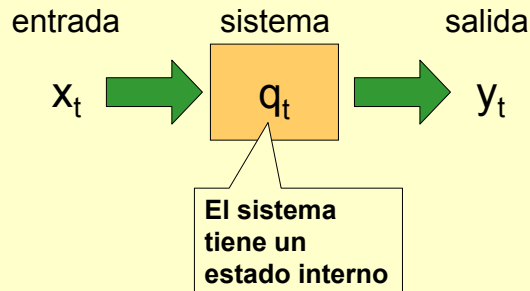
$$x(t) \rightarrow \{x_i\}_{i=0}^I$$



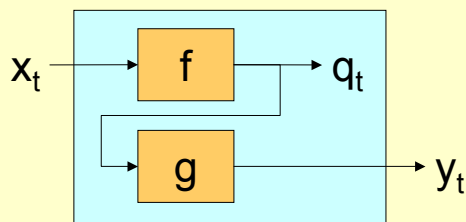
Los valores  
caen en la  
malla

- ❑  $x_i$  pertenece a un conjunto finito de valores  $R$  (alfabeto).
  - Cada  $x_i$  representa un intervalo continuo
  - $R$  es biyectable con los naturales.
  - Hemos pasado de los  $n^{\text{os}}$  reales a símbolos.
- ❑ Esta sucesión  $\{x_i\}$  aproxima bien  $x(t)$  bajo ciertas condiciones.
- ❑ Existen otras codificaciones que minimizan la cantidad de información en bits necesaria para codificar la señal.

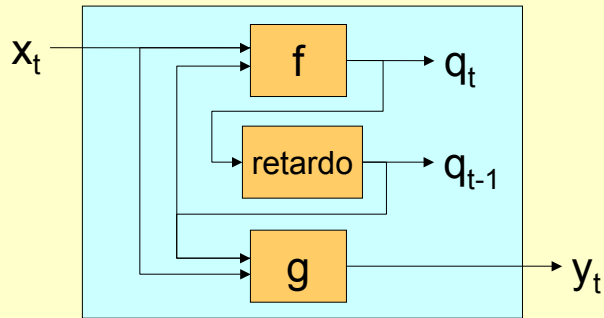
- ❑ Dispositivo donde se genera, procesa, transmite y almacena información en forma de magnitudes físicas discretas, tanto en número como en dimensión.
- ❑ El sistema es binario si las magnitudes se restringen a dos valores.
  - Existen sistemas ternarios, cuaternarios, ...  
En términos del número de valores.



- Los SD binarios son muy interesantes porque se pueden construir en términos de interruptores
  - Interruptor = Dispositivo de conmutación binario.
  - Transistores y válvulas de vacío en zonas de saturación y corte, relevadores, etc...



- Sistema combinacional
  - $q_t = f(x_t)$ , e  $y_t = g(q_t)$ 
    - $f()$  es la función de transición
    - $g()$  es la función de salida
  - $y_t = h(x_t) = (g \circ f)(x_t)$



❑ Sistema secuencial

$$q_t = f(q_{t-1}, x_t), \text{ e } y_t = g(q_{t-1}, x_t)$$

$$y_t = h(q_{t-1}, x_t) = (g \text{ o } f)(q_{t-1}, x_t)$$

- ❑ El considerar  $q_{t-1}$  hace preciso introducir un elemento de retardo (memoria).

## Álgebra de Boole y Álgebra de Conmutación

❑ Álgebra de Boole:  $\mathcal{B} = \langle B, +, \cdot \rangle$  donde

- B es un conjunto
- $+$  y  $\cdot$  son dos operaciones internas a B, con:
  1. B tiene (al menos) dos elementos
  2.  $+$  y  $\cdot$  tienen sendos elementos neutros, 0 y 1.
  3.  $+$  y  $\cdot$  son conmutativas ( $a+b=b+a \dots$ )
  4.  $+$  y  $\cdot$  son asociativas ( $a+(b+c)=(a+b)+c \dots$ )
  5.  $+$  y  $\cdot$  son recíprocamente distributivas ( $a+(b \cdot c)=a \cdot c+a \cdot b \dots$ )
  6. Cada  $x \in B$  tiene un elemento opuesto (complementario)  $\bar{x}$  de modo que:

$$x + \bar{x} = 1, \text{ y } x \cdot \bar{x} = 0$$

**Ejemplos**

- Álgebra de Conjuntos  
sea un conjunto  $A$ , y su potencia  $(2^A)$ ,  
 $\langle 2^A, \cup, \cap \rangle$  es un Álgebra de Boole.

- Álgebra de Boole mínima  
 $\langle \{0, 1\}, +, \cdot \rangle$  es un Álgebra de Boole.

Es la mas sencilla, y su teoría ecuacional presenta el mismo poder expresivo que la lógica de proposiciones.

**Ejemplos**

- Álgebra de Conjuntos  
sea un conjunto  $A$ , y su potencia  $(2^A)$ ,  
 $\langle 2^A, \cup, \cap \rangle$  es un Álgebra de Boole.
  1.  $\emptyset$  y  $A$  son dos elementos de  $2^A$
  2. Los elementos neutros para  $\cup, \cap$  son  $\emptyset$  y  $A$
  3. 4. y 5.  $\cup, \cap$  son conmutativas, asociativas y recíprocamente distributivas.
  6. Cada subconjunto  $S \subseteq A$  tiene un complementario  $C(S)$  de modo que  $S \cup C(S) = A$ , y  $S \cap C(S) = \emptyset$ .

## Ejemplos

- Álgebra de Boole mínima  
 $\langle \{0, 1\}, +, \cdot \rangle$  es un Álgebra de Boole.
  1. Tiene dos elementos.
  2. 0 es e. neutro de  $+$   $\Rightarrow 0+0=0, 1+0=0+1=1$   
 1 es e. neutro de  $\cdot$   $\Rightarrow 1 \cdot 1=1, 0 \cdot 1 = 1 \cdot 0 = 0$
  6. El complementario de 0 debe ser 1, y viceversa, ...
  3. Es conmutativa y asociativa, (fácil de ver)
 Nos falta ver  $1+1$  y  $0 \Rightarrow 0+0=0 \cdot 0$

- Un teorema importante reza  
 $a+a=a$ , y  $a \cdot a=a$ , con lo que:  
 $1+1=1$ , y  $0 \cdot 0=0$ 
  - Con lo anterior, podemos ver que los operadores así definidos cumplen con la
- 4. Asociatividad y
- 5. Distributividad
- El metateorema de la dualidad se aplica también aquí: tenemos doble juego de propiedades y de teoremas.  
 $0 \leftrightarrow 1, \cdot \leftrightarrow +, y \ x \leftrightarrow \bar{x}$

□ Teoremas interesantes:

1.  $a+a=a$ , y  $a \cdot a=a$
2.  $a+1=1$ , y  $a \cdot 0=0$
3.  $\overline{(\overline{a})}=a$
4.  $\overline{1}=0$ , y  $\overline{0}=1$
5.  $\overline{a+b}=\overline{a} \cdot \overline{b}$ , y  $\overline{a \cdot b}=\overline{a} + \overline{b}$
6.  $a+(\overline{a} \cdot b) = a+b$ , y  $a \cdot (\overline{a} + b) = a \cdot b$

## Álgebra de Conmutación

- Una variable de conmutación recorre el dominio  $C=\{0, 1\}$
- Una función de conmutación es una función  
 $C^n \xrightarrow{f} C$ , es decir  $f: x_1, \dots, x_n \mapsto f(x_1, \dots, x_n)$
- Para  $n$  variables, existen  $VR_2^{VR_2^n} = 2^{2^n}$  funciones



## Álgebra de conmutación

- Para  $n=2$ , existen 16 funciones de conmutación:

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

## Álgebra de conmutación

- $\mathcal{F} = \langle F, +, \cdot \rangle$ , siendo  $F$  el conjunto de funciones de  $n$  variables es un álgebra de Boole, y se denomina álgebra de conmutación de  $n$  variables, con operadores  $+$  y  $\cdot$  definidos así:

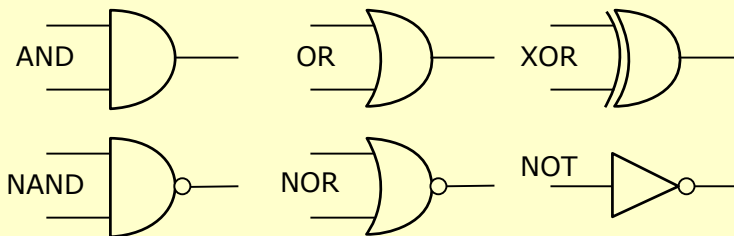
$$(f + g)(\vec{x}) = \begin{cases} 0 & \text{si } f(\vec{x}) = g(\vec{x}) = 0 \\ 1 & \text{de otro modo} \end{cases} \quad \bar{f}(\vec{x}) = \begin{cases} 1 & \text{si } f(\vec{x}) = 0 \\ 0 & \text{si } f(\vec{x}) = 1 \end{cases}$$

$$(f \cdot g)(\vec{x}) = \begin{cases} 1 & \text{si } f(\vec{x}) = g(\vec{x}) = 1 \\ 0 & \text{de otro modo} \end{cases}$$

- Los elementos neutros son **0** y **1**:  
 $\mathbf{0}(x) = 0$ , y  $\mathbf{1}(x) = 1$ .

## Álgebra de conmutación

- ❑ ¿Para qué sirve?  
"permite expresar funciones mediante 0, 1, los operadores +, ·, y el complemento"
- ❑ Basta con un A. de C. de 2 variables.
- ❑ Se puede implementar con circuitos electrónicos

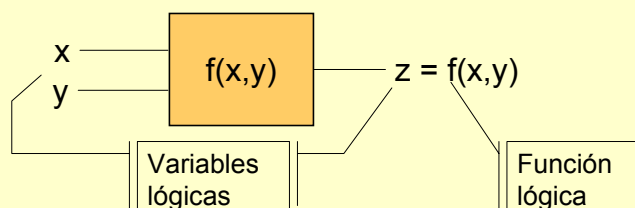


Puertas lógicas más comunes

## Álgebra de Conmutación

### *expresión*

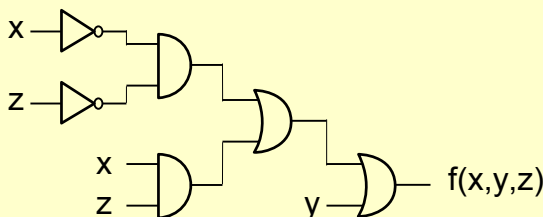
- ❑ Formas de expresar una función de conmutación:
  - Indicando propiedades del sistema (en este caso podrá definir una familia de funciones)
  - Según una tabla de entrada-salida
  - Según una expresión del álgebra de conmutación
  - Según un diagrama de bloques equivalente.



- Función  $f$  de tres variables cuya salida es 1 siempre que la primera y tercera variables sean iguales, o cuando la segunda coincida con la tercera:

| $x$ | $y$ | $z$ | $f^3(x,y,z)$ |
|-----|-----|-----|--------------|
| 0   | 0   | 0   | 1            |
| 0   | 0   | 1   | 0            |
| 0   | 1   | 0   | 1            |
| 0   | 1   | 1   | 1            |
| 1   | 0   | 0   | 1            |
| 1   | 0   | 1   | 1            |
| 1   | 1   | 0   | 0            |
| 1   | 1   | 1   | 1            |

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



- Dado un número entero  $I$ , con  $0 \leq I \leq 2^n - 1$ , y su expresión en binario natural con  $n$  bits  $(i_1, \dots, i_n)$ , se define la función:

- Mínterm, como:

$$m_I(x_1, \dots, x_n) = \begin{cases} 1 & \text{si } x_1 = i_1, \dots, x_n = i_n \\ 0 & \text{de otro modo} \end{cases}$$

- Máxterm, como:

$$M_I(x_1, \dots, x_n) = \begin{cases} 0 & \text{si } x_1 = i_1, \dots, x_n = i_n \\ 1 & \text{de otro modo} \end{cases}$$

- Las funciones mínterm y máxterm funcionan como elementos de una base de un espacio de funciones.

- Una función está en **forma normal** si se escribe como
  - suma lógica (+) ordenada (por I) de minterms (forma disyuntiva), o
  - producto lógico (·) ordenado (por I) de máxterms (forma conjuntiva).
- El teorema de Shannon dice que toda función de conmutación tiene una forma normal conjuntiva y una disyuntiva.
- Cada minterm (o máxterm) dice qué pasa con una combinación de entradas concreta.

| x | y | z | $f^3(x,y,z)$ | minterms                              | maxterms                |
|---|---|---|--------------|---------------------------------------|-------------------------|
| 0 | 0 | 0 | 1            | $\bar{x} \cdot \bar{y} \cdot \bar{z}$ |                         |
| 0 | 0 | 1 | 0            |                                       | $x + y + \bar{z}$       |
| 0 | 1 | 0 | 1            | $\bar{x} \cdot y \cdot \bar{z}$       |                         |
| 0 | 1 | 1 | 1            | $\bar{x} \cdot y \cdot z$             |                         |
| 1 | 0 | 0 | 1            | $x \cdot \bar{y} \cdot \bar{z}$       |                         |
| 1 | 0 | 1 | 1            | $x \cdot \bar{y} \cdot z$             |                         |
| 1 | 1 | 0 | 0            |                                       | $\bar{x} + \bar{y} + z$ |
| 1 | 1 | 1 | 1            | $x \cdot y \cdot z$                   |                         |

$$\begin{aligned}
 f^3(x,y,z) &= (\bar{x} \cdot \bar{y} \cdot \bar{z}) + (\bar{x} \cdot y \cdot \bar{z}) + (\bar{x} \cdot y \cdot z) + \\
 &\quad (x \cdot \bar{y} \cdot \bar{z}) + (x \cdot \bar{y} \cdot z) + (x \cdot y \cdot z) \\
 &= (x + y + \bar{z}) \cdot (\bar{x} + \bar{y} + z)
 \end{aligned}$$

## Circuitos combinacionales básicos

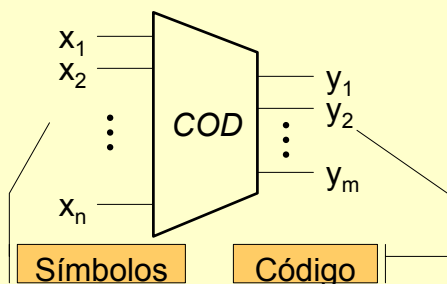
- ❑ **Codificador:** recibe símbolos y emite palabras que codifican el símbolo entrante
- ❑ **Decodificador:** recibe palabras que codifican un símbolo y emite dicho símbolo.
- ❑ **Multiplexor:** proyecta una de las variables de entrada sobre la salida, siguiendo un código.
- ❑ **Demultiplexor:** activa una función de proyección para la entrada, siguiendo un código (como el multiplexor).

17/03/2003

Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003

25

## Circuitos combinacionales básicos *codificador*



- ❑  $\Sigma = \{x_1, \dots, x_n\}$  n símbolos de entrada, y m variables de salida  $\{y_1, \dots, y_m\}$  con el código correspondiente

- ❑ **Funcionamiento:**  
En cada operación se entiende que sólo una de las variables de entrada tiene valor 1, de modo que representa cierto símbolo,  
Las variables de salida presentarán una configuración correspondiente a un código.

17/03/2003

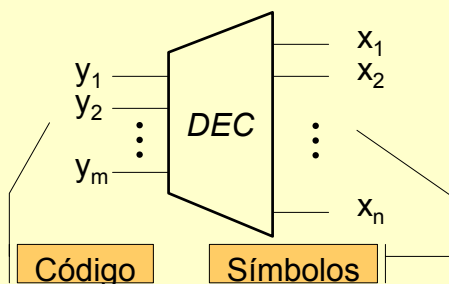
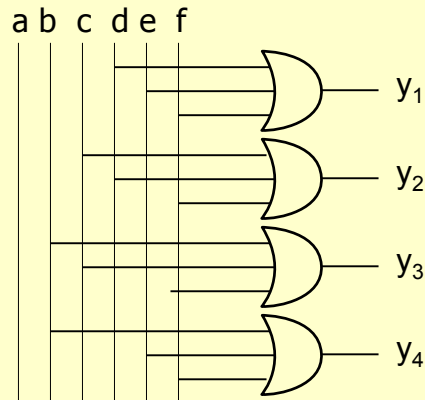
Fundamentos de Informática II (ITI Sistemas) (C) César Llamas Bello, 2003

26

- Supongamos  $\Sigma = \{a, b, c, d, e, f\}$  y el código  $\{(a, 0000), (b, 0011), (c, 0110), (d, 1100), (e, 1001), (f, 1111)\}$ , en versión simple:

| a | b | c | d | e | f | $y_1$ | $y_2$ | $y_3$ | $y_4$ |
|---|---|---|---|---|---|-------|-------|-------|-------|
| 1 | 0 | 0 | 0 | 0 | 0 | 0     | 0     | 0     | 0     |
| 0 | 1 | 0 | 0 | 0 | 0 | 0     | 0     | 1     | 1     |
| 0 | 0 | 1 | 0 | 0 | 0 | 0     | 1     | 1     | 0     |
| 0 | 0 | 0 | 1 | 0 | 0 | 1     | 1     | 0     | 0     |
| 0 | 0 | 0 | 0 | 1 | 0 | 1     | 0     | 0     | 1     |
| 0 | 0 | 0 | 0 | 0 | 1 | 1     | 1     | 1     | 1     |

entradas                      salidas



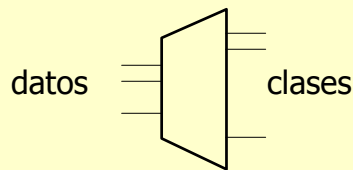
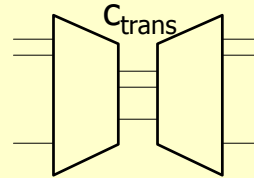
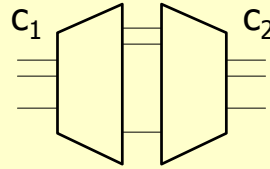
- m variables de entrada  $\{y_1, \dots, y_m\}$  con un código, y  $\Sigma = \{x_1, \dots, x_n\}$  n símbolos de salida.

- Funcionamiento:  
En cada operación se entiende que se recibe a la entrada la codificación de un símbolo: Las variables de salida presentarán 0 salvo aquella salida que represente al símbolo correspondiente a la entrada, que mostrará 1.

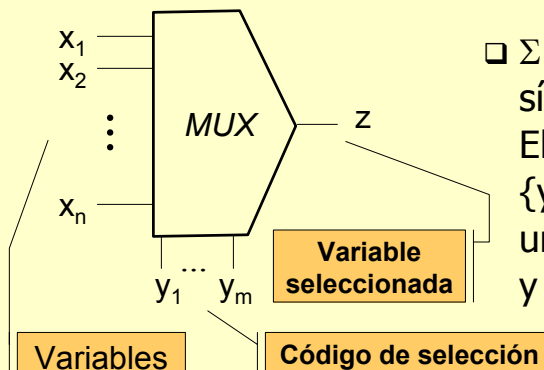
## Circuitos combinacionales básicos *utilidades*

### □ Algunas utilidades:

- Traducir de un código ( $c_1$ ) a otro ( $c_2$ ).
- Codificar la información para transmitirla (codec)
- Clasificar datos de su forma codificada a símbolos.



## Circuitos combinacionales básicos *multiplexor*



- $\Sigma = \{1, \dots, n\}$  n símbolos codificados. El código de entrada  $\{y_1, \dots, y_m\}$  selecciona una de las variables  $x_i$ , y la propaga hacia z.

### □ Funcionamiento:

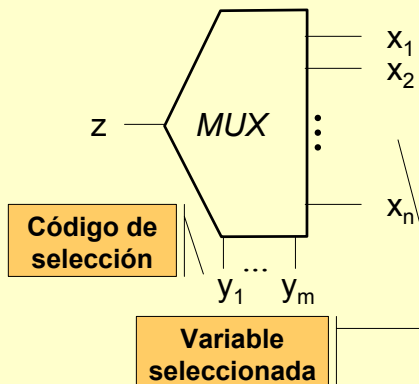
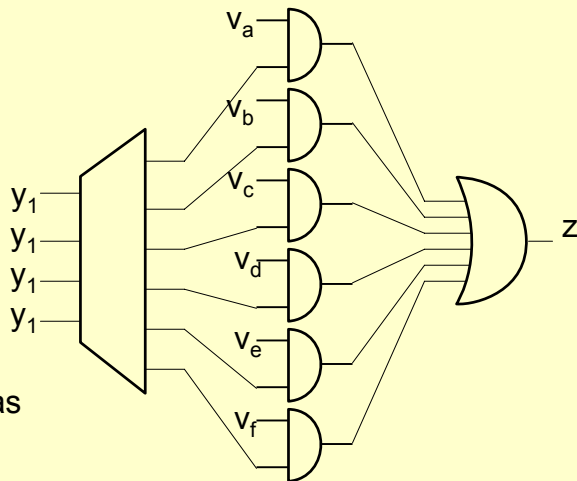
Actúa como un selector, conectando una de las variables de entrada con la salida. La variable se selecciona por el código de selección. El resto de entradas se ignoran.

□  $\{y_1, \dots, y_4\}$  codifican  $\{a, b, c, d, e, f\}$ .

| $y_1$ | $y_2$ | $y_3$ | $y_4$ | $z$   |
|-------|-------|-------|-------|-------|
| 0     | 0     | 0     | 0     | $V_a$ |
| 0     | 0     | 1     | 1     | $V_b$ |
| 0     | 1     | 1     | 0     | $V_c$ |
| 1     | 1     | 0     | 0     | $V_d$ |
| 1     | 0     | 0     | 1     | $V_e$ |
| 1     | 1     | 1     | 1     | $V_f$ |

entradas sin  
 $\{V_a, \dots, V_f\}$

salidas



□  $\Sigma = \{1, \dots, n\}$  n  
símbolos codificados.  
El código de entrada  
 $\{y_1, \dots, y_m\}$  selecciona  
una de las variables  $x_i$ ,  
y recibe valores de z.

□ Funcionamiento:

Actúa como un selector, conectando la variable de entrada con una de las salidas. La variable se selecciona por el código de selección.  
El resto de salidas se puede ignorar.

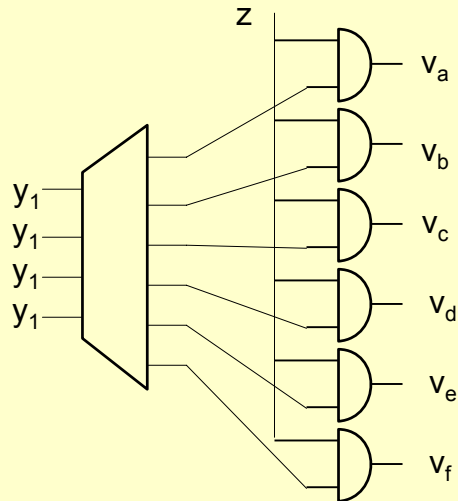


□  $\{y_1, \dots, y_4\}$  codifican  $\{a, b, c, d, e, f\}$ .

| $y_1$ | $y_2$ | $y_3$ | $y_4$ |           |
|-------|-------|-------|-------|-----------|
| 0     | 0     | 0     | 0     | $V_a = z$ |
| 0     | 0     | 1     | 1     | $V_b = z$ |
| 0     | 1     | 1     | 0     | $V_c = z$ |
| 1     | 1     | 0     | 0     | $V_d = z$ |
| 1     | 0     | 0     | 1     | $V_e = z$ |
| 1     | 1     | 1     | 1     | $V_f = z$ |

entradas sin  
z

salidas



□ Algunas utilidades:

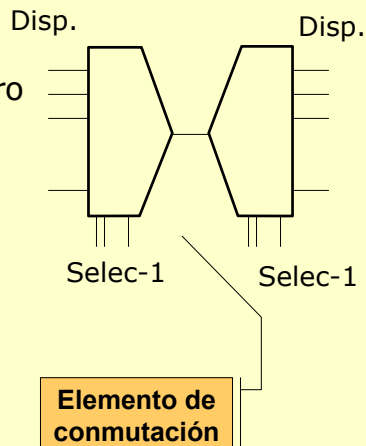
- Compartir una línea de comunicación.
- Conectar un dispositivo con otro

□ Otros multiplexores:

- Multiplexación de un recurso:
  - Tiempo de UCP,
  - Bus de sistema, ...

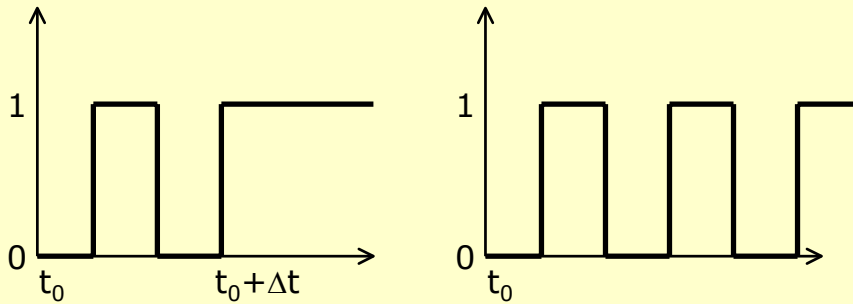
□ Conceptos:

- Multiplexación en tiempo
- Multiplexación en frecuencia
- Multiplexación en un programa



- ❑ Sistemas feed-forward: combinacionales
  - Son sistemas sin recursividad en el tiempo
- ❑ Sistemas feed-back: con realimentación
  - Son sistemas con recursividad en el tiempo:  
Para calcular ciertos valores se requieren valores que se calcularán más adelante
  - A este tipo pertenecen los sistemas que presentan memoria.

- ❑ En los sistemas realimentados, para cierta entrada el sistema es:
  - Estable
    - Constante: si la salida converge a un valor.
    - Acotado: si la salida oscila pero está acotada.
  - Inestable
    - Esencial: si la salida se dispara
    - Oscilante: si la salida oscila y sus oscilaciones crecen.
- ❑ Los sistemas digitales son siempre estables, pues su salida está acotada.



### □ Evolución en el tiempo:

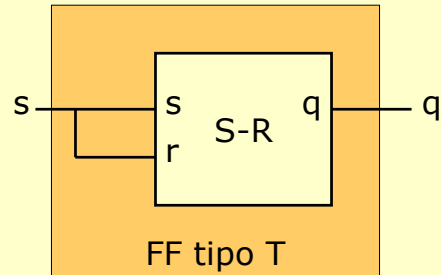
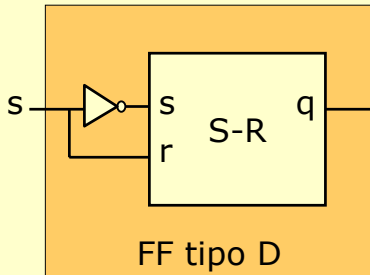
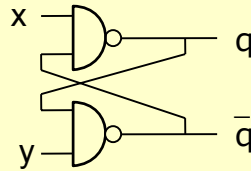
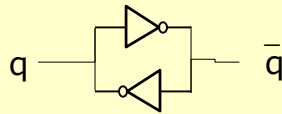
- Asíncrona, si la evolución del sistema ocurre libremente hasta que el sistema se estabiliza.
- Síncrona, si la evolución del sistema procede al ritmo de un reloj o una señal externa.

### □ Biestable: dispositivo secuencial con dos estados (también se denomina flip-flop, FF)

- Puede tener una o dos entradas,
- Su evolución puede ser síncrona o asíncrona.

### □ Tipos básicos:

- Cerrojo (*latch*),
- Set-Reset (*S-R*),
- J-K,
- Maestro-Esclavo
- Oscilador (*toggle* o T)
- Retardo (*delay* o D),
- ... y combinaciones de ellos.

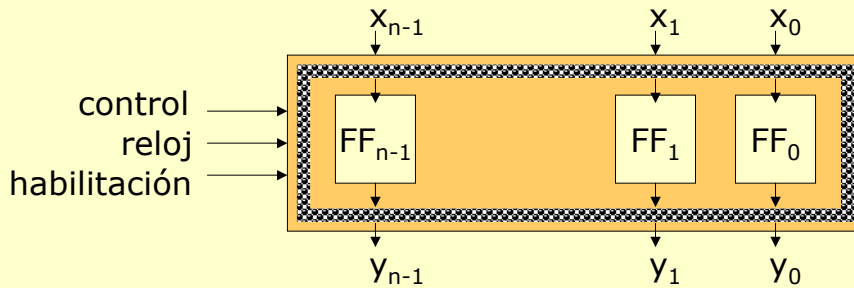


| FF tipo D |               |              |
|-----------|---------------|--------------|
| Entrada   | Salida actual | Nueva Salida |
| 0         | 0             | 0            |
| 0         | 1             | 0            |
| 1         | 0             | 1            |
| 1         | 1             | 1            |

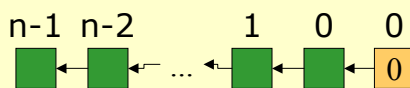
- Bueno para construir registros de almacenamiento de bits y circuitos secuenciales en general.

| FF tipo T |               |              |
|-----------|---------------|--------------|
| Entrada   | Salida actual | Nueva Salida |
| 0         | 0             | 0            |
| 0         | 1             | 1            |
| 1         | 0             | 1            |
| 1         | 1             | 0            |

- Bueno para construir osciladores y contadores.

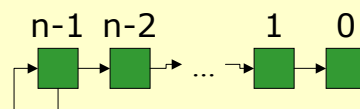


- Un registro es una colección de biestables, interconectados mediante una lógica de control, donde se opera con cadenas de bits.
  - Para almacenarlos,
  - Hacer un desplazamiento lógico o aritmético,
  - Hallar su complemento lógico, ...



| $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ | #  |
|-------|-------|-------|-------|-------|-------|----|
| 0     | 0     | 1     | 1     | 0     | 1     | 13 |
| 0     | 1     | 1     | 0     | 1     | 0     | 26 |

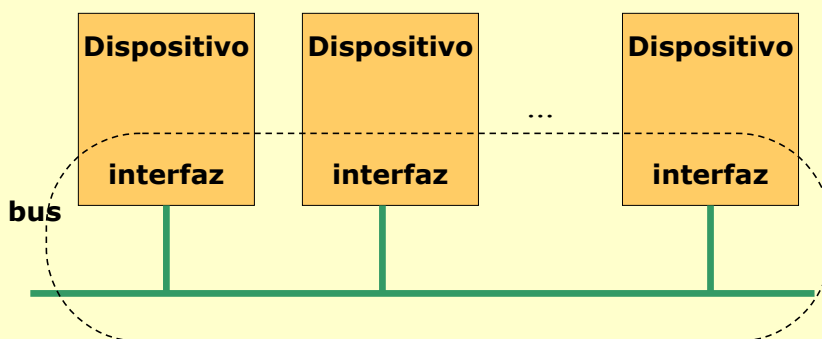
- Desplazamiento aritmético izquierda (x2):
  - Al desplazar a la derecha se aumenta el valor de cada dígito
  - Se insertan ceros



| $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ | #  |
|-------|-------|-------|-------|-------|-------|----|
| 0     | 1     | 1     | 1     | 0     | 1     | 29 |
| 0     | 0     | 1     | 1     | 1     | 0     | 14 |

- Desplazamiento aritmético derecha (/2)
  - Al desplazar a la derecha se disminuye el valor de cada dígito.
  - Se propaga el signo.

- ❑ Dispositivo para conectar varios dispositivos entre sí.
- ❑ Partes habituales de un bus informático
  - Sub-bus de datos (para transmitir datos de un elemento a otro)
  - Sub-bus de direcciones (para direccionar la memoria central)
  - Sub-bus de control y arbitraje (para asignar el uso del bus y señalar el estado y otros eventos)
- ❑ Cada operación es un *ciclo de bus*.



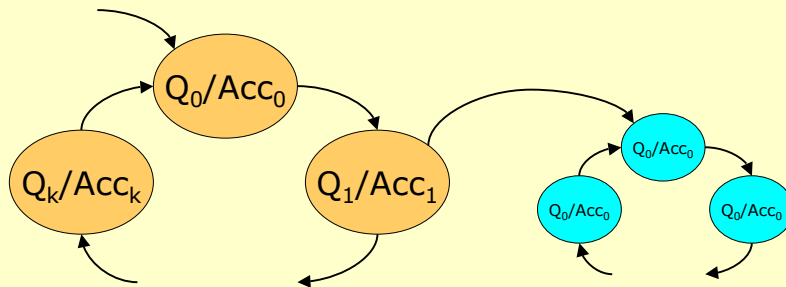
- ❑ Operación:
  - Reconoce la dirección del dispositivo
  - Almacena el estado
  - Proporciona almacenamiento intermedio
  - Proporciona señales de temporización y control

### □ Niveles de especificación del bus

- Mecánico
- Eléctrico y óptico
- Lógico
- Temporización básica
- Transferencia elemental
- Transferencia de bloque

### □ Características del bus

- Tipo de lógica
- Grado de paralelismo
- Sincronización
- Protocolo
- Temporización
- Direccionalidad de los datos
- Estrategia de control
- Velocidad y tasa de transferencia
- Longitud máxima
- Capacidad de conexión
- Soporte



- ❑ El contador evoluciona al ritmo de un reloj y posiblemente según otras señales externas.
  - Módulo del contador: número de etapas de un ciclo completo del contador
  - Permite generar señales de control en cada paso del contador

- ❑ Controlador: máquina de estado finito
  - Si es programable se denomina autómata programable.
- ❑ Se especifican sus estados, sus entradas y sus acciones.
- ❑ Se emplea para aplicaciones de carácter muy controlable:
  - Microcontroladores para dispositivos simples
  - Automatización de robots
  - Microcontroladores para microprocesadores
  - ...