

- **Análisis**

- Algoritmo de análisis, para un circuito lógico combinacional

- **Síntesis. Conceptos**

- Circuitos combinacionales bien contruidos

- Circuitos combinacionales mal contruidos

- Criterios de optimización

- Definiciones y propiedades para síntesis mínima

- **Síntesis. Herramientas**

- Aplicaciones del mapa de Karnaugh

- Determinación de IP

- Determinación de suma mínima

- **Concepto de indeterminación**

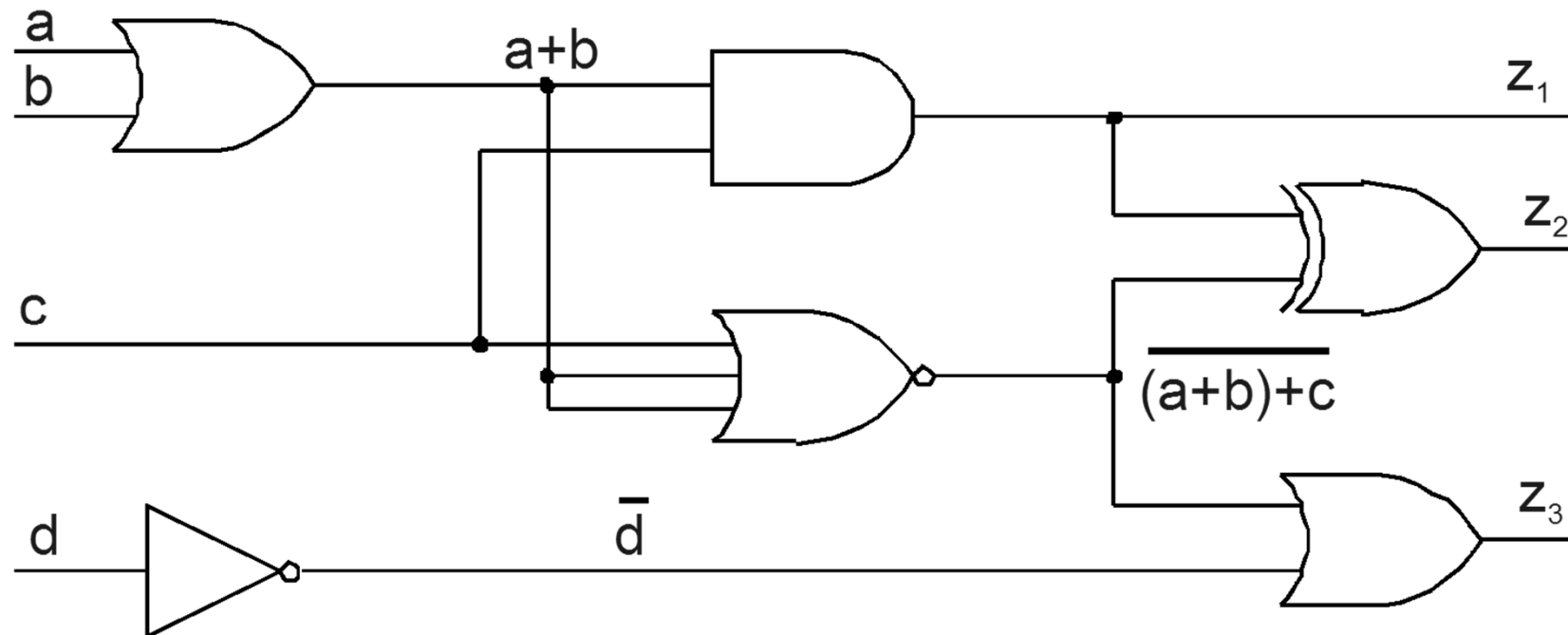
Análisis

- **Algoritmo de análisis, para un circuito lógico combinacional**

Analizar un circuito combinacional consiste en determinar, a partir de su estructura, cual será su comportamiento para todas las posibles entradas, o lo que es lo mismo, determinar la función lógica que implementa.

- **Procedimiento de análisis lógico**

- a) Identificar las entradas, las salidas y las variables intermedias.
- b) Identificar los niveles de puertas.
 - b.1 Constituyen el nivel 1, aquellas puertas cuyos terminales de entrada estén conectados únicamente a las entradas del circuito.
 - b.2 Constituyen el nivel $n+1$, aquellas puertas que tienen al menos uno de sus terminales de entrada conectado a las salidas del nivel n , y el resto a las salidas de los niveles precedentes, o a las entradas del circuito.
- c) Expresar las salidas de cada nivel en función de sus entradas
 - c.1 Comenzar por el nivel $n=1$.
 - c.2 Especificar las salidas del nivel n en función de las entradas a dicho nivel.
 - c.3 Incrementar el nivel de análisis.
 - c.4 El procedimiento termina cuando las salidas del sistema están expresadas en función de las entradas.



$$z_1 = (a + b) c$$

$$z_2 = z_1 \oplus \overline{(a + b) + c} = (a + b) c \oplus \bar{a} \bar{b} \bar{c}$$

$$z_3 = \overline{(a + b) + c} + \bar{d} = \bar{a} \bar{b} \bar{c} + \bar{d}$$

Síntesis. Conceptos

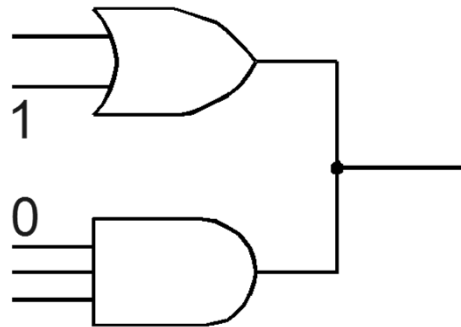
- **Circuitos combinacionales bien contruidos**

Normas de los circuitos bien contruidos

- a) Una sola línea (cable) o puerta lógica con señales de entrada primarias constantes o variables (binarias), está bien contruido
- b) Un circuito formado por la yuxtaposición de dos circuitos disjuntos bien contruidos $C1$ y $C2$ (es decir $C1$ y $C2$ colocados lado a lado), está bien contruido
- c) Dados dos circuitos disjuntos (desconectados) bien contruidos $C1$ y $C2$, el circuito que se obtiene al conectar una línea de salida de $C1$ a una línea de entrada de $C2$ o a una nueva salida primaria, está bien contruido
- d) Si C está bien contruido, el circuito que se obtiene al conectar dos entradas primarias de C para formar una sola entrada primaria también estará bien contruido.

- **Circuitos combinacionales mal contruidos**

Dos o más salidas interconectadas pueden tener valores distintos, con lo que el resultado es impredecible



La salida de un circuito bien construido, conectada a una de sus entradas provoca una realimentación que puede provocar fenómenos de oscilación.



- **Criterios de optimización**

El objetivo a conseguir es implementar de forma mínima una expresión de conmutación de manera que sean mínimos

- Coste real

- Espacio ocupado

- Tiempo de respuesta

- Consumo de potencia

Tenemos las siguientes restricciones físicas al diseño:

- Máximo fan-in y fan-out

- Reglas de interconexión

- Diagnosticabilidad y testabilidad

Nuestros objetivos de diseño en esta asignatura serán:

- Minimizar el número de puertas

- Minimizar el número de conexiones

- Minimizar el número de niveles

- Definiciones y propiedades para síntesis mínima

Dadas $f(x)$ y $g(x)$ dos funciones de conmutación, se dice que $f(x)$ **implica** a $g(x)$ (se denota $f(x) \subseteq g(x)$), y que $g(x)$ **cubre** a $f(x)$, si verifica:

$$\forall x / f(x) = 1 \Rightarrow g(x) = 1$$

Dadas $f(x)$ y $g(x)$ dos funciones de conmutación, se dice que $f(x)$ **implica estrictamente** a $g(x)$ (se denota $f(x) \subset g(x)$), si verifican:

$$\forall x / f(x) = 1 \Rightarrow g(x) = 1$$

$$\exists x / f(x) = 0 \text{ y } g(x) = 1$$

Definimos **implicante** de una función de conmutación a aquellos términos producto que implican dicha función

Definimos **implicante primo** a todo implicante tal que ningún término que lo cubra es implicante

Definimos **cubrimiento** de una función a la suma de todos sus implicantes

Definimos **suma irredundante** a la suma de todos los implicantes tales que si se elimina uno de ellos, la expresión resultante no cubre la función

Definimos **implicante primo esencial** a todo implicante primo que aparece en todas las sumas irredundantes

Definimos **suma mínima** a la suma irredundante con el menor número de implicantes primos

Definimos **suma estrictamente mínima** a la suma mínima con el menor número de literales

- **Propiedad para diseño mínimo**

La suma estrictamente mínima asociada a una función, proporciona una realización AND-OR en dos niveles, que cumple los objetivos de diseño óptimo.

Síntesis. Herramientas

- Mapa de Karnaugh para determinación de IP

Dada una función booleana representada en forma de mapa de Karnaugh

Definimos *casillas adyacentes* a aquellas que sólo se diferencian entre sí, en el valor de una de las variables de la función (en el mapa de Karnaugh estarán juntas gráficamente)

Definimos *grupo permitido* de [1's] a aquel en forma de cuadrado ó rectángulo, que contenga un número de 2^n casillas, y que cada casilla que lo forma sea adyacente con otras n casillas de dicho grupo

Grupos de $2^0=1$ [1's] adyacentes con 0 [1's]

Grupos de $2^1=2$ [1's] adyacentes con 1 [1's]

Grupos de $2^2=4$ [1's] adyacentes con 2 [1's]

Grupos de $2^3=8$ [1's] adyacentes con 3 [1's]

.....

A cada grupo de los formados corresponderá a un **término producto**, que contendrá todas las variables que permanezcan constantes en todas las casillas apareciendo complementadas para el caso de tener valor 0 y sin complementar en caso de tener valor 1

Serán **implicantes** todos los términos producto obtenidos a partir de un *grupo permitido* de [1's]

Serán **implicantes primos** aquellos implicantes que no puedan ser incluidos en otro *grupo permitido* de [1's] mayor

Serán **implicantes primos esenciales** aquellos implicantes primos que son imprescindibles para cubrir por completo todos los [1's] de la función

		x_1x_2			
		00	01	11	10
x_3x_4	00		1	1	
	01	1	1	1	
	11	1	1		
	10	1	1		

Algunos implicantes

$\bar{x}_1x_3\bar{x}_4$ $\bar{x}_1x_2\bar{x}_3$ $x_2\bar{x}_3\bar{x}_4$ $x_2\bar{x}_3x_4$

		x_1x_2			
		00	01	11	10
x_3x_4	00		1	1	
	01	1	1	1	
	11	1	1		
	10	1	1		

Implicantes primos

$$x_2\bar{x}_3 \quad \bar{x}_1x_4 \quad \bar{x}_1x_3 \quad \bar{x}_1x_2$$

Implicantes primos esenciales

$$x_2\bar{x}_3 \quad \bar{x}_1x_4 \quad \bar{x}_1x_3$$

Implicantes primos NO esenciales

$$\bar{x}_1x_2$$

- **Mapa de Karnaugh para obtención de la suma mínima**

- a) Obtener los implicantes primos esenciales (IP)
- b) La suma de menor número de IP es mínima
- c) Si hay varias sumas mínimas, la que tenga el menor número de literales es estrictamente mínima

Si aplicamos las reglas anteriores pero considerando los 0 en lugar de los 1, obtendremos la suma mínima de la función complementaria

Definimos celda *distinguida*, como aquella que sólo pertenecen a un único IP

		x_1x_2			
		00	01	11	10
x_3x_4	00		1	1	
	01	1	1	1	
	11	1	1		
	10	1	1		

Suma mínima

$$f(x_1, x_2, x_3, x_4) = x_2\bar{x}_3 + \bar{x}_1x_4 + \bar{x}_1x_3$$

Concepto de Indeterminación

- **Indeterminación**

Decimos que una función combinacional presenta una indeterminación en su salida para una combinación de valores de entrada, si el valor de la función para dicha combinación no está especificado.

Causas:

- a) Porque en la definición de la función se establece explícitamente la inespecificación.
- b) Porque la combinación de valores que provoca la indeterminación no pueden darse nunca en las entradas.

Las indeterminaciones son asignadas interesadamente, a valor 0 ó a valor 1 de forma que la implementación de la función sea la mínima posible.

Una vez implementada la función ya no presentará indeterminaciones. Al aplicar sobre la función la combinación que provocaba la indeterminación, obtendremos a la salida el valor asignado a dicha combinación en tiempo de diseño.