

# **Tecnología de Computadores**

## **Práctica 3**



**Universidad  
Rey Juan Carlos**

## 1. Objetivo

En esta tercera sesión de prácticas se pretende que el alumno se familiarice con la minimización de circuitos combinacionales mediante mapas de Karnaugh.

Una vez terminada la práctica el alumno será capaz de:

- Diseñar circuitos combinacionales minimizados en VHDL
- Medir los tiempos de propagación de una señal y obtener el camino crítico de un circuito
- Utilizar el mapa de Karnaugh para simplificar funciones lógicas
- Implementar diseños combinacionales minimizados en una FPGA y comprobar su funcionamiento

## 2. Desarrollo de la práctica

### Minimización de circuitos mediante mapas de Karnaugh.

Como vimos en la práctica anterior, una función lógica dada se puede simplificar aplicando propiedades y teoremas del álgebra de Boole. En esta tercera práctica aplicaremos el concepto de los mapas de Karnaugh visto en clase para minimizar circuitos digitales de forma gráfica. Vamos a partir de la siguiente ecuación:

$$F = \bar{A} \cdot C + A \cdot \bar{B} + \bar{A} \cdot \bar{B} \cdot \bar{C}$$

Para construir un mapa de Karnaugh, primero tenemos que partir de su tabla de verdad. Como vimos, una forma rápida de hacerlo consiste en simular el circuito. Utilice el código VHDL del testbench "*tb\_karnaugh.vhd*" proporcionado para simular dicha función y complete la tabla de verdad. Construya, a partir de dicha tabla, el mapa de Karnaugh.

**Nota:** llame "karnaugh" a la entidad VHDL creada.

Si se realizan correctamente los pasos se obtendrá el siguiente mapa de Karnaugh:

A \ BC	00	01	11	10
0	1	1	1	0
1	1	1	0	0

Agrupando los unos, la función lógica resultante queda:

$$F = \underbrace{\bar{A} \cdot C}_{\text{rojo}} + \underbrace{\bar{B}}_{\text{azul}}$$

Cabe destacar que, si se agruparan los ceros, obtendríamos otra función lógica equivalente a la anterior. Obtenga la función lógica del circuito agrupando los ceros. Compare ambas implementaciones obteniendo el número de puertas lógicas (AND, OR y NOT) y el número de recursos de la FPGA utilizados (LUTs e IOBs). Para ello, incluya primero el fichero de constraints proporcionado ¿Cuál de las dos implementaciones es mejor?

Como se vio en la primera sesión de prácticas, además del número de recursos utilizados, también se pueden obtener otros informes como el de tiempos. Se recuerda que para ello es necesario ejecutar la implementación del circuito, abrir el diseño implementado y luego pulsar en la opción "*Report Timing Summary*". Para que nos salga al menos un tiempo de Setup y otro de Hold para cada ruta entrada-salida ( $A \rightarrow F$ ,  $B \rightarrow F$  y  $C \rightarrow F$ ) debemos cambiar el valor por defecto de la opción "*Maximum number of worst paths per endpoint*". Pondremos 6 en esta casilla (ver figura de la siguiente página), ya que Vivado genera varios caminos alternativos por señal. En el apartado de "*Unconstrained Paths > NONE to NONE*" podremos encontrar los tiempos de Setup y de Hold.

Report Timing Summary

Generate a timing summary to understand if the design met timing.

Results name:

Options | **Advanced** | Timer Settings

**Report**

Path delay type:

☒ Report unconstrained paths

☐ Report datasheet

**Path Limits**

Maximum number of paths per clock or path group:

Maximum number of worst paths per endpoint:

**Path Display**

Display paths with slack less than:  ☒ Use default (1e+30)

Significant digits:

**Command:** `ing_summary -delay_type min_max -report_unconstrained -check_timing_verbose -max_paths 10 -nworst 6 -input_pins -routable_nets -name timing_1`

☒ Open in a new tab

☐ Open in Timing Analysis layout

Si hemos seguido los pasos correctamente podremos consultar los tiempo de *Setup* y de *Hold* como se presenta en la siguiente figura.

Timing x

Unconstrained Paths - NONE - NONE - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock
Path 1	∞	3	2	1	C	F	8.696	5.130	3.566	∞	input port clock
Path 2	∞	3	2	1	C	F	8.696	5.130	3.566	∞	input port clock
Path 3	∞	3	2	1	A	F	8.608	5.122	3.486	∞	input port clock
Path 4	∞	3	2	1	A	F	8.608	5.122	3.486	∞	input port clock
Path 5	∞	3	2	1	B	F	8.257	5.124	3.133	∞	input port clock
Path 6	∞	3	2	1	B	F	8.257	5.124	3.133	∞	input port clock

Setup (6)

Hold (6)

Con esta información podemos calcular el tiempo que tarda en llegar cada una de las señales de entrada hasta la salida del circuito sumando los tiempos de *Setup* y de *Hold* de cada señal de entrada. Esto es lo que se conoce como **tiempo de propagación** de una señal. Unido a este concepto se define lo que se denomina como **camino crítico** de un circuito, que no es más que el camino que genera el peor tiempo de propagación, es decir, el camino por el que tarda más en llegar la señal de entrada hasta la salida del circuito.

Determine el tiempo de propagación de cada señal de entrada para las dos implementaciones del circuito (minimizado agrupando unos y agrupando ceros). ¿Cuál es el camino crítico en cada circuito?

### Conjuntos universales de puertas lógicas.

Como se ha visto en teoría, se puede transformar cualquier circuito combinacional en uno equivalente de modo que se utilicen únicamente puertas lógicas NAND o puertas lógicas NOR ya que forman un conjunto universal de puertas en sí mismo.

En el circuito de ejemplo, si tomamos la función minimizada como suma de productos:

$$F = \bar{A} \cdot C + \bar{B}$$

Podemos, aplicando una doble negación y las leyes de De Morgan, reescribir la ecuación de modo que sea fácilmente implementable mediante puertas lógicas NAND e inversores, quedando:

$$\bar{\bar{F}} = \overline{(\bar{A} \cdot C + \bar{B})} = \overline{(\bar{A} \cdot C)} \cdot \overline{(\bar{B})} = \overline{(\bar{A} \cdot C)} \cdot B$$

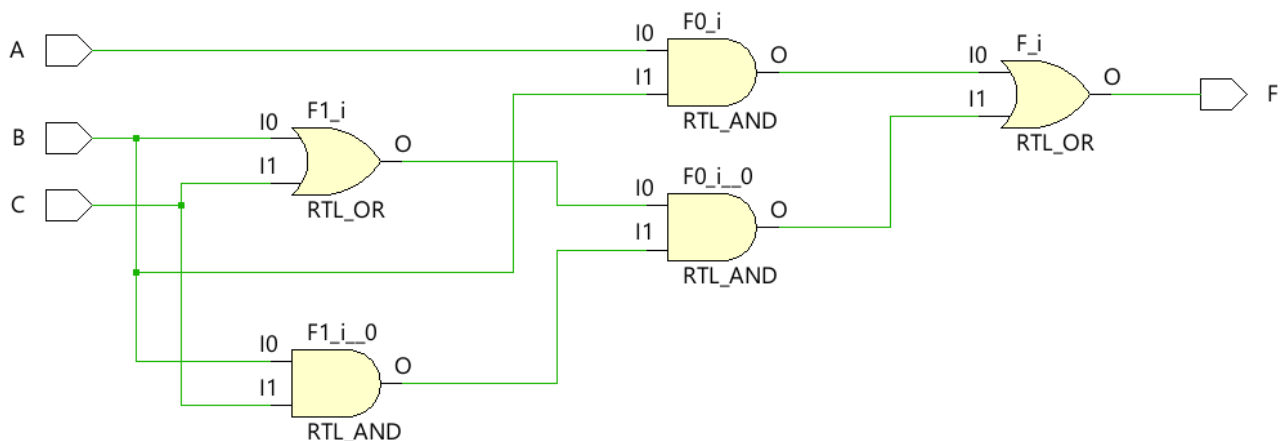
Como vemos, en esta ecuación únicamente aparecen productos lógicos y negaciones, por lo que se puede diseñar usando puertas NAND e inversores.

El proceso análogo se puede aplicar a la función obtenida mediante producto de sumas para generar un circuito que se puede implementar utilizando únicamente puertas NOR e inversores. Obtenga la ecuación lógica de dicho circuito.

### 3. Ejercicios propuestos

A modo de repaso de los conceptos vistos durante la práctica se plantean estos últimos ejercicios:

1. Dado el siguiente circuito combinacional:



- ¿Cuántas LUTs utiliza el circuito simplificado mediante suma de productos?
  - ¿Cuál es el tiempo del camino crítico de cada circuito (inicial y simplificado)?
  - A la vista de estos resultados, determine qué implementación es mejor.
2. Sintetizar un circuito combinacional que tiene como entrada un número X de cuatro dígitos binarios (X3 X2 X1 y X0) y como salida un número Z. La salida Z representa la operación XOR de los cuatro dígitos de X.
- Simplificar el circuito en forma de suma de productos.
  - Obtener el diseño elaborado del circuito y contabilizar número de puertas lógicas AND, OR y NOT.
  - Sintetizar el circuito utilizando únicamente puertas NAND. Contabilice cuántas son necesarias.
  - ¿Cuál es el consumo energético total (*On-Chip Power*) de cada una de las dos implementaciones?
  - A la vista de estos resultados, determine qué implementación es mejor.