Universidad de Alcalá

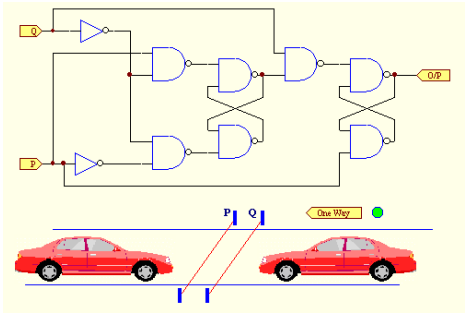
Departamento de Electrónica

Modelado de Sistemas Computacionales

Grado en Ingeniería de Computadores

Tema 1: Modelado de sistemas digitales en FPGAs

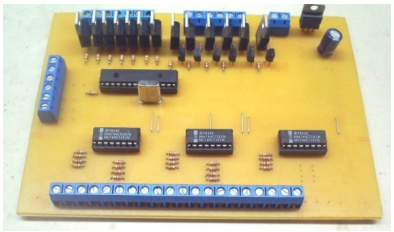
Alternativas al diseño digital



- Sistemas lentos
- Consumos Altos
- Complejidad baja
- Costes “altos”
- No reconfigurables
- Pocos fiables

- Circuitos muy rápidos
- Consumos muy bajos
- Complejidad muy alta
- Costes “bajos”
- Reconfigurables
- Fiabilidad alta

Lógica estándar



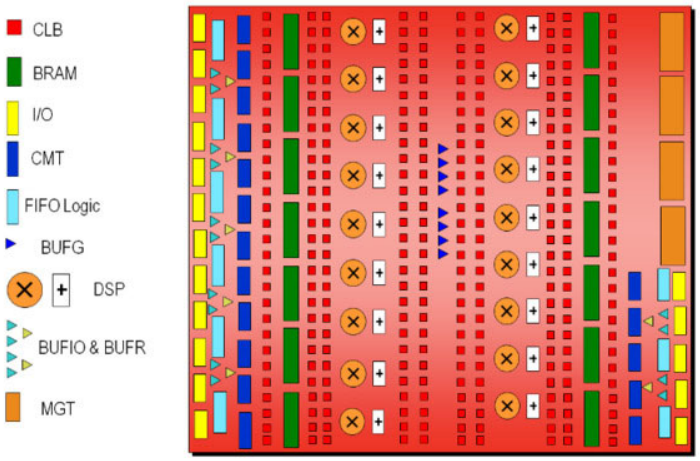
ASICs
(Application Specific Integrated Ciruit)

UAH

Departamento de Electrónica

FPGA: *Field Programmable Gate Array*

- Basadas en arrays bidimensionales de bloques lógicos que se pueden interconectar entre sí y con el exterior.
- Configuración :
  - SRAM: reprogramables (Xilinx)



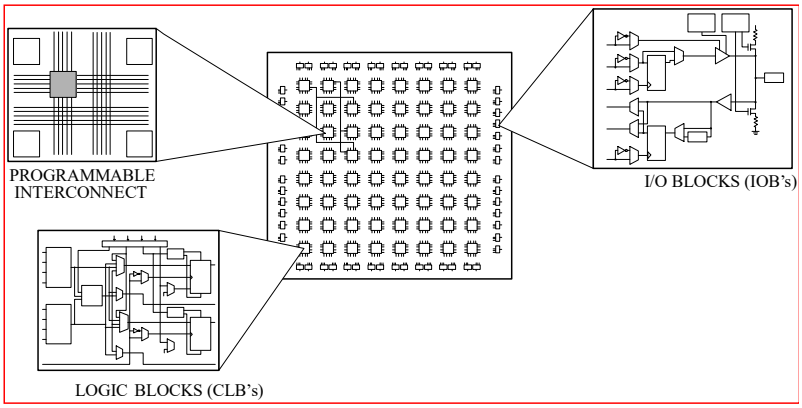
Artix-7 Architecture Overview

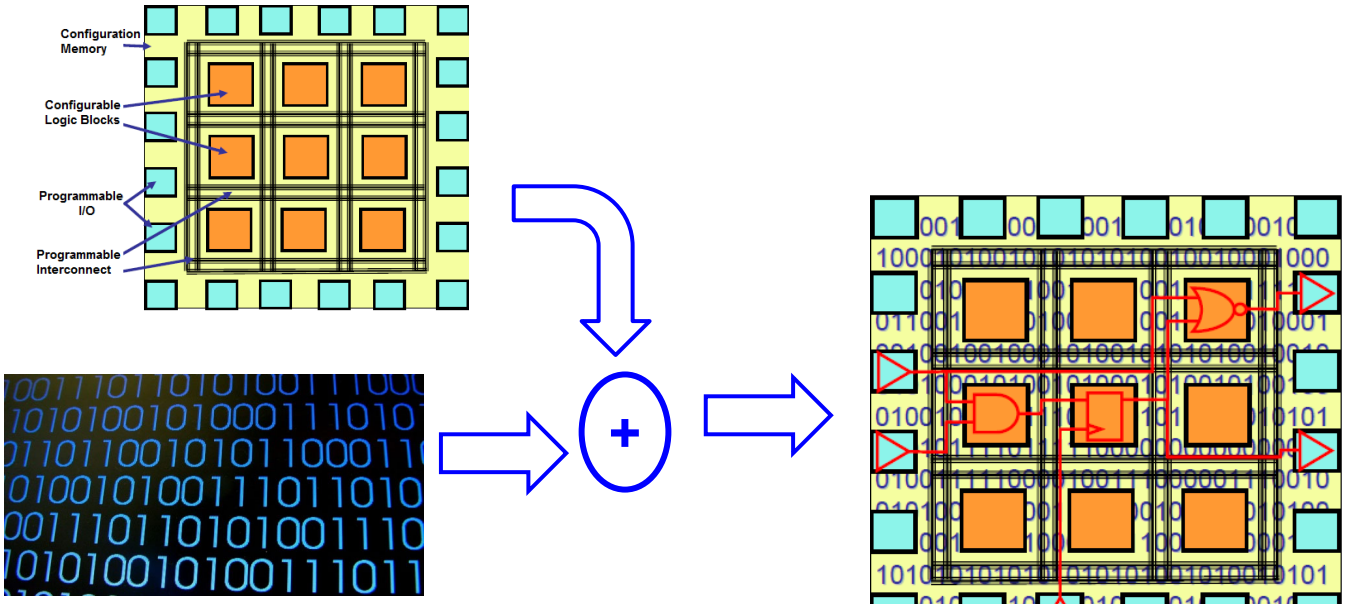
UAH

Departamento de Electrónica

Recursos

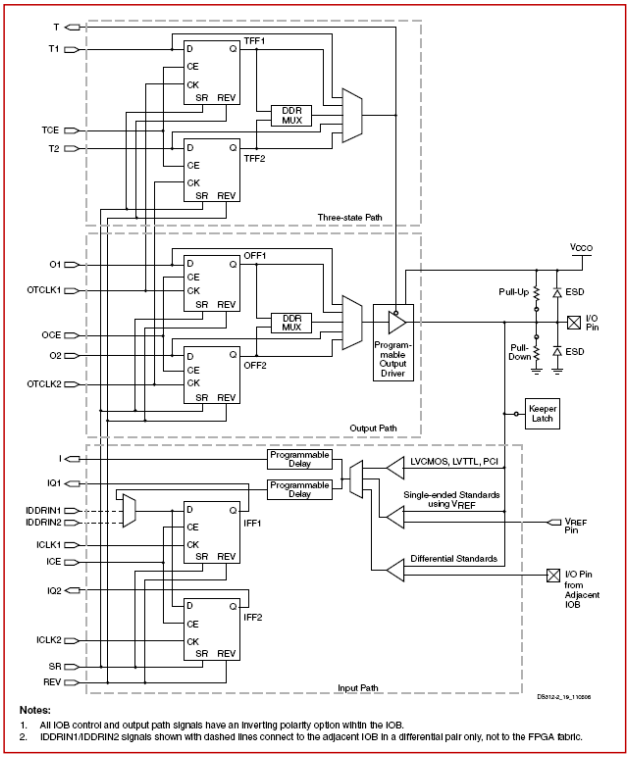
- CLBs : Configurable Logic Blocs
  - Contienen lógica combinatorial y registros
  - IOBs :Input Output Blocs
  - Interface entre la FPGA y el exterior
- Interconexiones programables
- Otros recursos
  - Memoria
  - Multiplicadores
  - Digital Clock Managers
  - Global clock buffers
  - Boundary scan
  - Microprocesadores





Estructura interna de las FPGAs de Xilinx

❑ IOBs

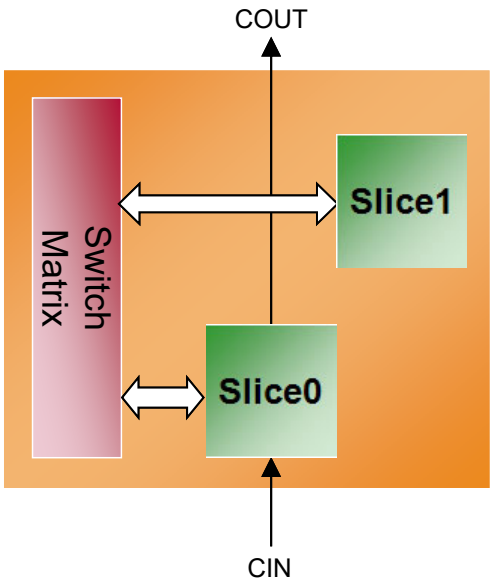


- ❑ Estructurada basada en tres bloques: entrada, salida, y tri-estado.
- ❑ Biestables:
- ❑ Buffer de salida:

Estructura interna de las FPGAs de Xilinx

❑ CLBs.

- ❑ **Cada CLB contiene 2 slices.**
- ❑ **Se conecta a la "switch matrix" para unirse a otros recursos de la FPGA.**

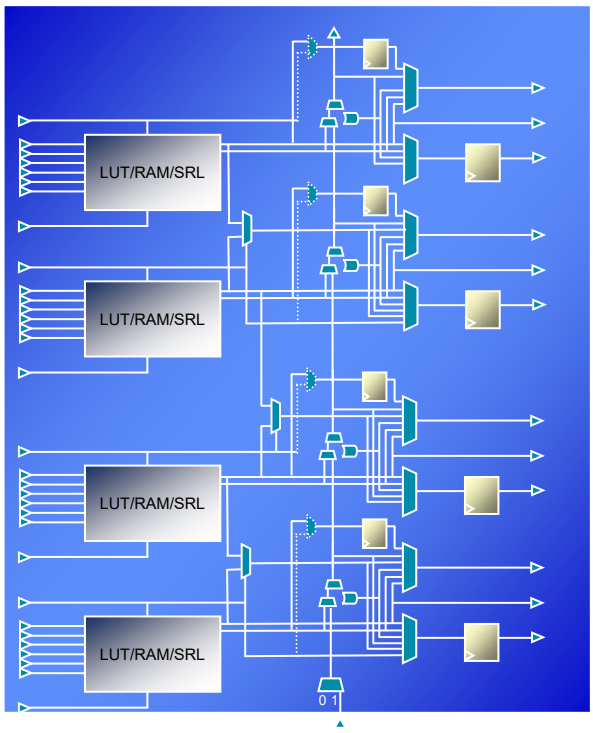


UAH
Departamento de Electrónica

Estructura interna de las FPGAs de Xilinx

❑ Slices.

- ❑ 4 Look Up Tables (LUT) de 6 entradas
- ❑ 4 flip-flop
- ❑ Líneas de propagación de acarreo (carry chain)
- ❑ Varios multiplexores
- ❑ El diseño se debe realizar para utilizar de la mejor forma posible los recursos



Estructura interna de las FPGAs de Xilinx

❑ Memoria RAM distribuida

Single Port	Dual Port	Simple Dual Port	Quad Port
32x2 32x4 32x6 32x8 64x1 64x2 64x3 64x4 128x1 128x2 256x1	32x2**D** 32x4**D** 64x1**D** 64x2**D** 128x1**D**	32x6**SDP** 64x3**SDP**	32x2**Q** 64x1**Q**

- ❑ Utiliza las LUTs
- ❑ El ciclo de escritura es síncrono y el lectura asíncrono
- ❑ Hay varias configuraciones
  - Single port
  - Dual port (**D**)
  - Quad-port (**Q**)
    - 1 read / write port + 3 read-only ports
  - ROM

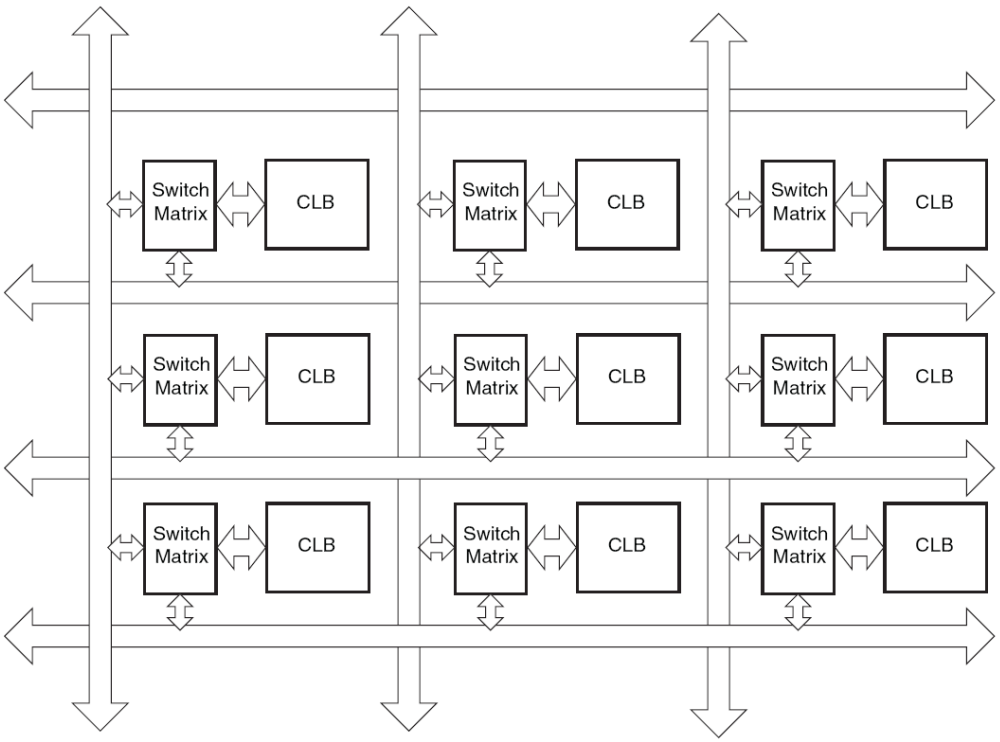
❑ Otros recursos

- ❑ Multiplicadores dedicados
- ❑ Bloques de BRAM/SelectRAM.

UAH
Departamento de Electrónica

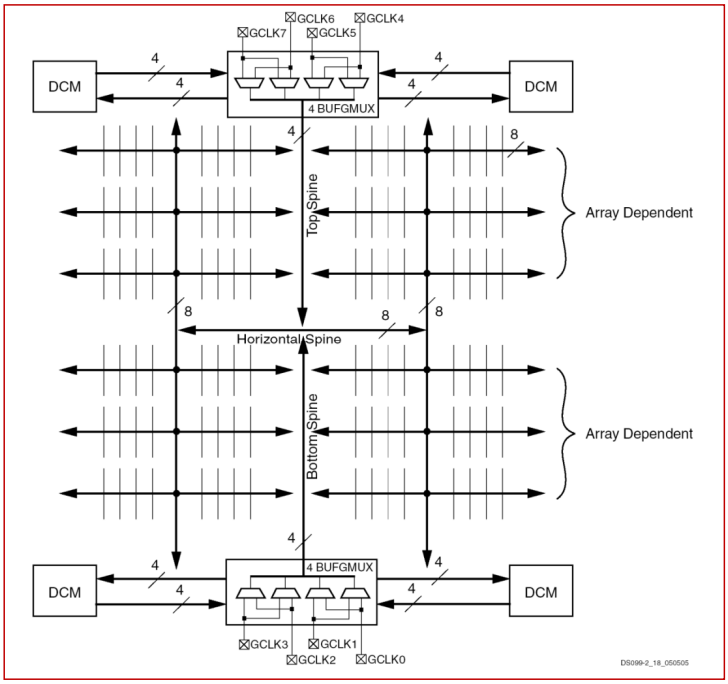
Estructura interna de las FPGAs de Xilinx

❑ Interconexiones



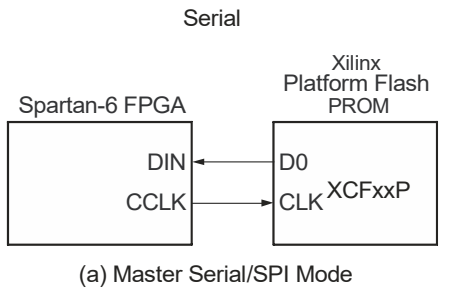
Estructura interna de las FPGAs de Xilinx

- Distribución de la señal de reloj.

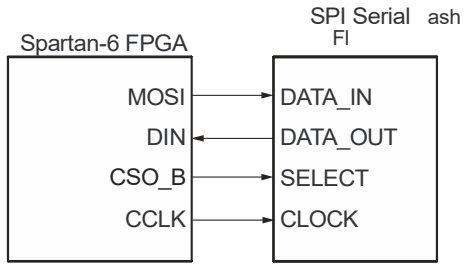


- GCLK0-GCLK7 Entradas de relojes globales.
- BUFGMUX: Multiplexores de relojes globales.
- Distribuyen otras señales de bajo skew y alto fanout:
  - Relojes adicionales.
  - Clock enables.
  - Reset.
  - Control tri-estate.

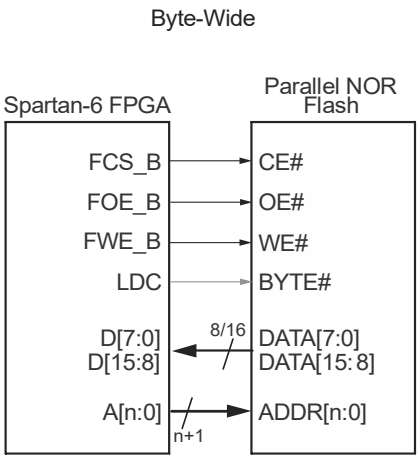
Programación de las FPGAs de Xilinx



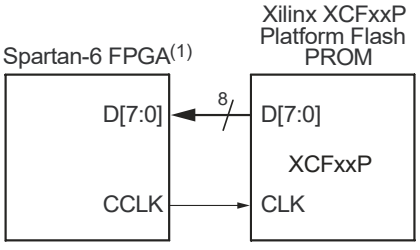
(a) Master Serial/SPI Mode



(b) Master Serial/SPI Mode with SPI Flash



(c) Master SelectMAP/BPI Mode with Parallel NOR Flash

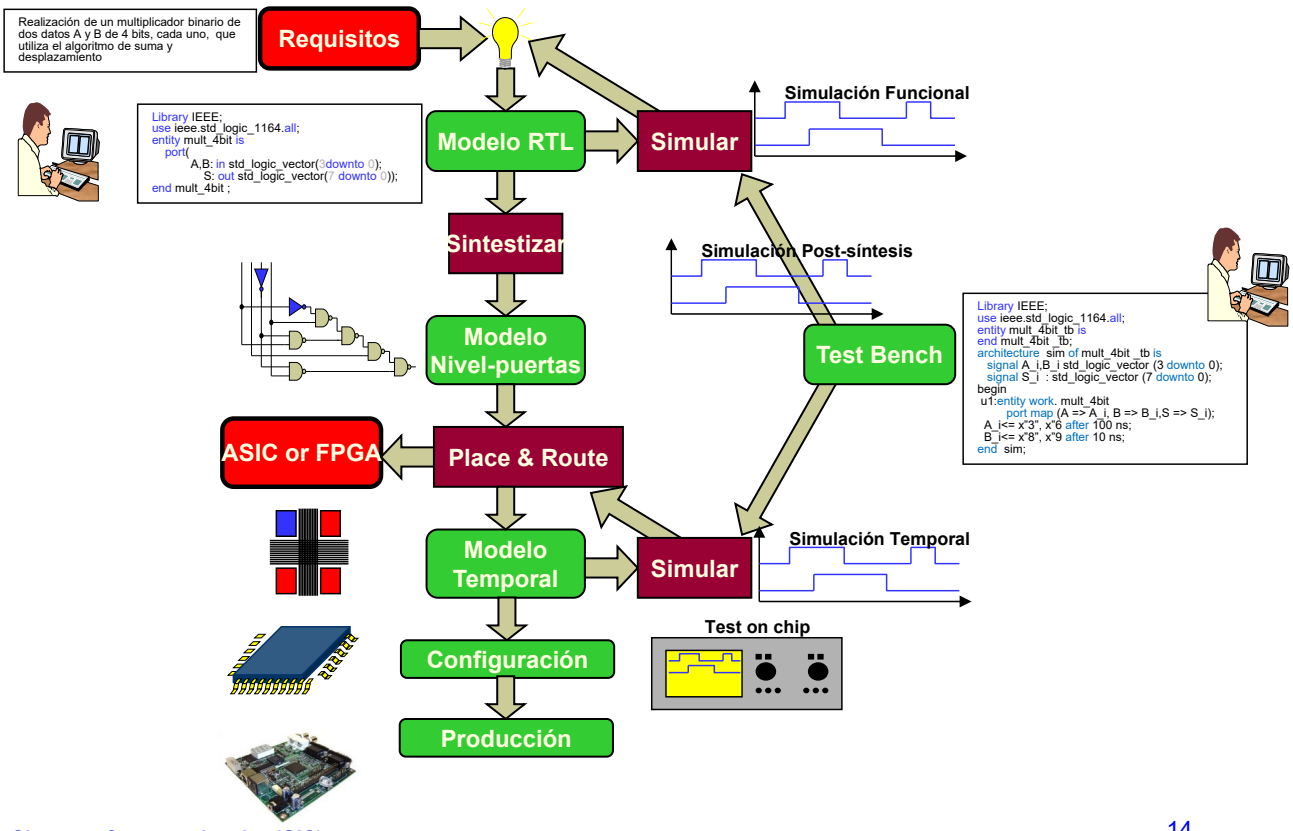


(d) Master SelectMAP/BPI Mode

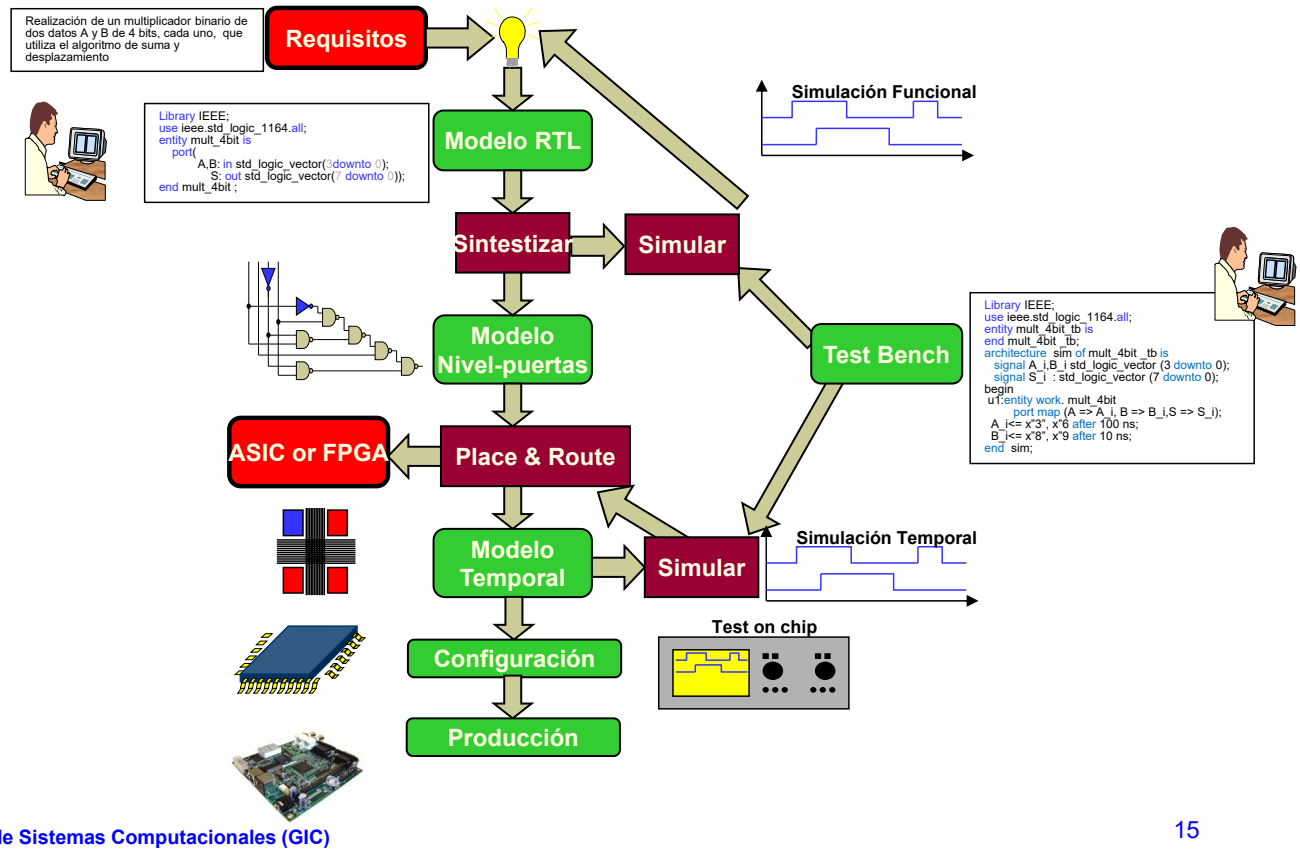
Características

- ☺ Diseño sencillo.
- ☺ Tiempo de diseño corto.
- ☺ Costes fijos bajos.
- ☺ No penalizan los cambios.
- ☹ Tamaño de los diseños limitado.
- ☹ Complejidad de los diseños limitada.
- ☹ Rendimiento limitado.
- ☹ Consumos altos.
- ☹ Coste por unidad alto.

Metodología de diseño



Metodología de diseño



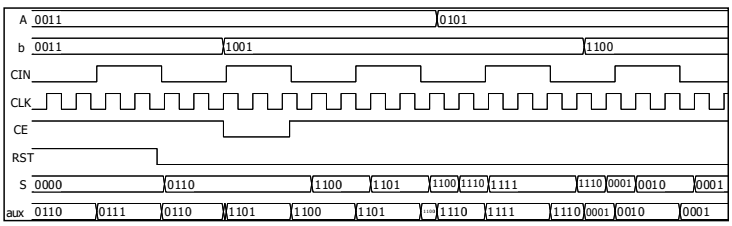
UAH
Departamento de Electrónica

Metodología de diseño

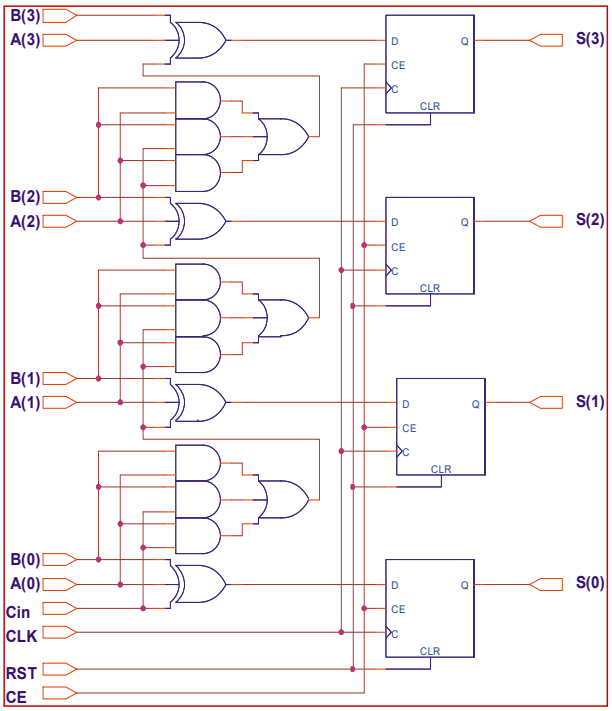
Diseñar un sistema digital que almacene el resultado de la suma de dos números ***A*** y ***B*** de 4 bits más una entrada de acarreo ***Cin***. La suma se almacena siempre que la señal ***CE*** sea un nivel alto. El sistema dispone de una entrada ***RST*** de inicialización asíncrona que lleva todos los bits de la salida a nivel bajo.

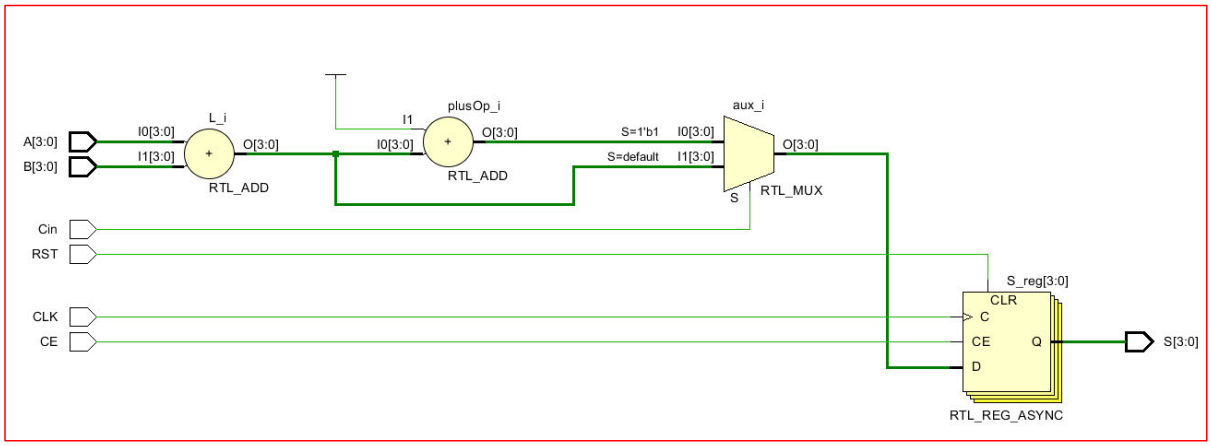
```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity dis is
  port (
    A   : in  std_logic_vector(3 downto 0);
    B   : in  std_logic_vector(3 downto 0);
    Cin : in  std_logic;
    CLK : in  std_logic;
    CE  : in  std_logic;
    RST : in  std_logic;
    S   : out std_logic_vector(3 downto 0));
end dis;

architecture RTL of dis is
  signal aux : std_logic_vector(3 downto 0);
begin
  process (A, B, Cin)
  begin
    if cin = '1' then
      aux <= std_logic_vector(unsigned(A)+unsigned(B)+1);
    else
      aux <= std_logic_vector(unsigned(A)+unsigned(B));
    end if;
  end process;

  process (CLK, RST)
  begin
    if RST = '1' then
      S   <= "0000";
    elsif CLK'event and CLK = '1' then
      if CE = '1' then
        S <= aux;
      end if;
    end if;
  end process;
end RTL;
```

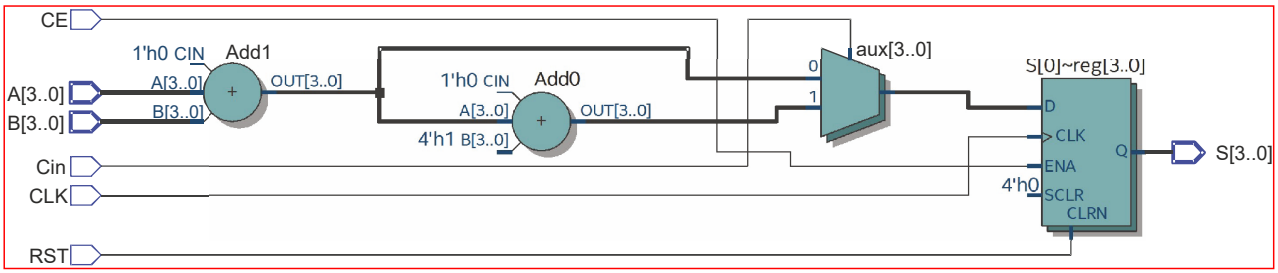
Simulación funcional



❑ Síntesis

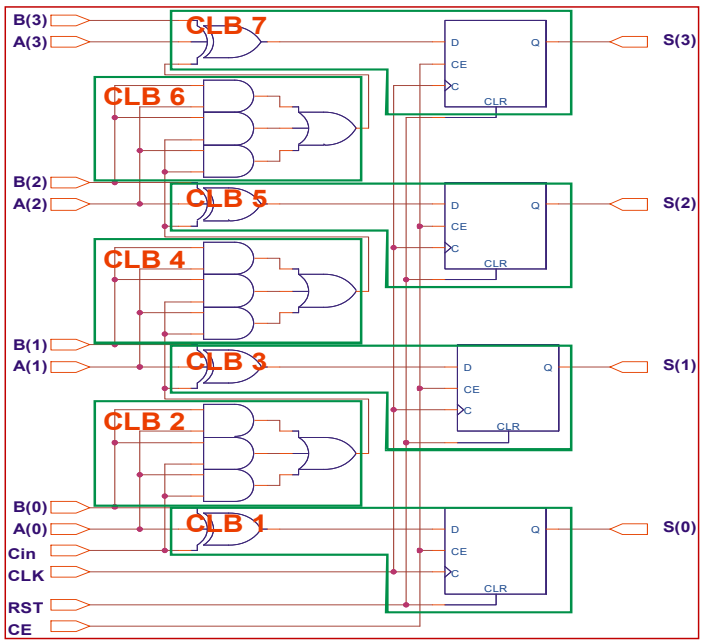




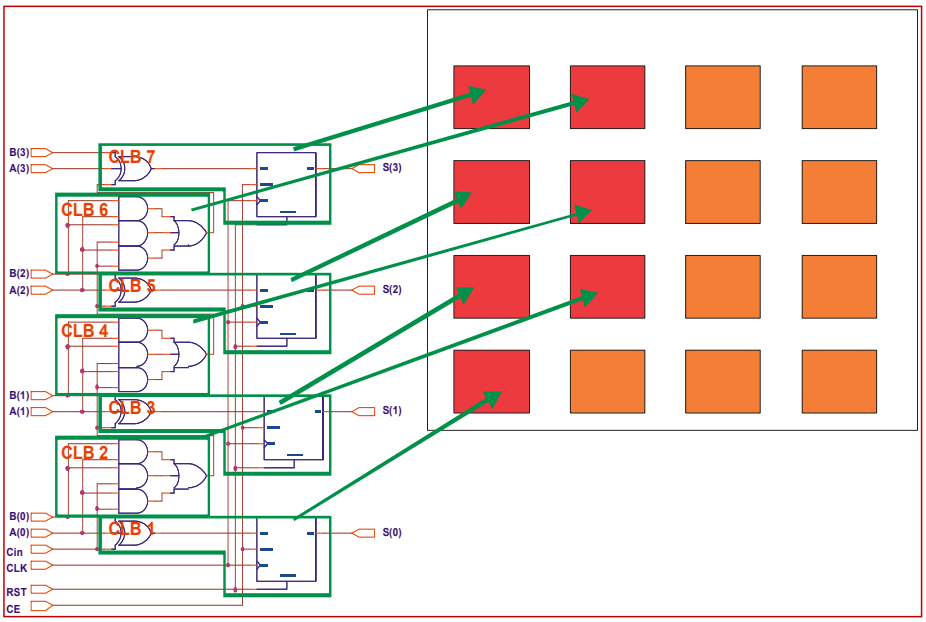


El resultado de la síntesis depende de la herramienta utilizada

❑ Mapping



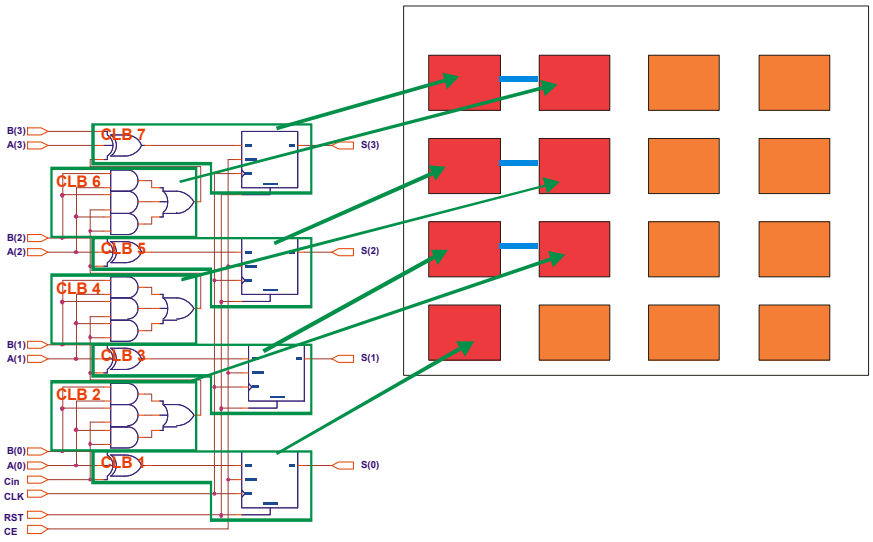
❑ Place



UAH

Departamento de Electrónica

❑ Route



UAH

Departamento de Electrónica

❑ Simulación temporal