

4. Comparadores

Un comparador compara los números de entrada y su salida nos informa si los números son iguales.

4.1. Diseño: Comparador de dos números binarios de 4 bits

Algoritmo

El algoritmo que nos determina si dos números son iguales o no, y en caso de ser distintos, cuál es el mayor, sería el siguiente:

Se comienza comparando los bits de mayor peso. Si son iguales, entonces se comparan los anteriores dos bits de mayor peso, así sucesivamente hasta que una pareja de bits sea diferente. En ese momento, el bit cuyo valor sea 1 será el que identifique al número mayor y ya no habrá que seguir comparando. Este algoritmo se resume gráficamente en la Figura 91

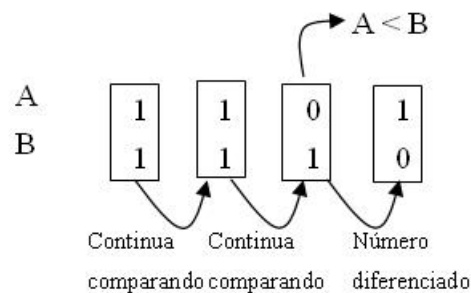


Figura 91

Diseño

El diseño se basará en implementar un circuito para cada pareja de bits de los números A y B, a_i y b_i respectivamente. Dicho circuito dispondrá como entradas, los dos bits (a_i , b_i) y la información de “si se ha diferenciado el número en alguna pareja de bits anteriores”. Las salidas serán dos, una de ellas indicará si se debe continuar con la comparación en los bits de menor peso ya que no ha habido diferenciación anteriormente, y la otra, que indique en caso de que los bits de entrada sean diferentes, cuál de ellos es el mayor. El esquema de entradas/salidas y la Tabla de Verdad correspondiente se presentan en la Figura 92.

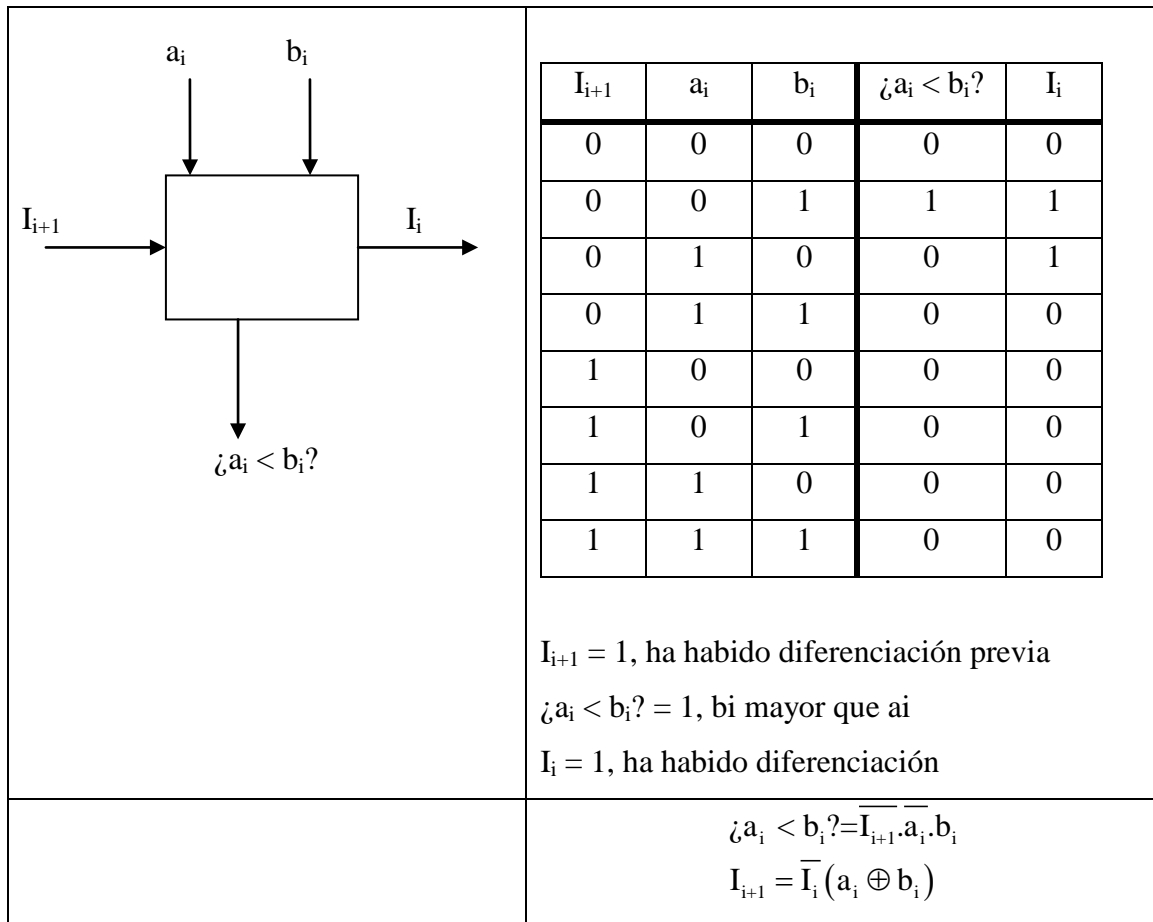


Figura 92

Una vez diseñado el circuito básico comparador de dos bits, bastara con conectar estos circuitos para que realicen la función descrita en la Figura 91. Los comparadores, además de la salida que indica si los dos números son iguales, tienen salidas adicionales que indican cuál de los dos números que se comparan es el mayor. Es decir, existe una salida que indica cuándo A es el mayor ($A > B$) y otra que indica cuándo B es el mayor ($A < B$). Por lo tanto, el circuito completo sería el representado en la Figura 93.

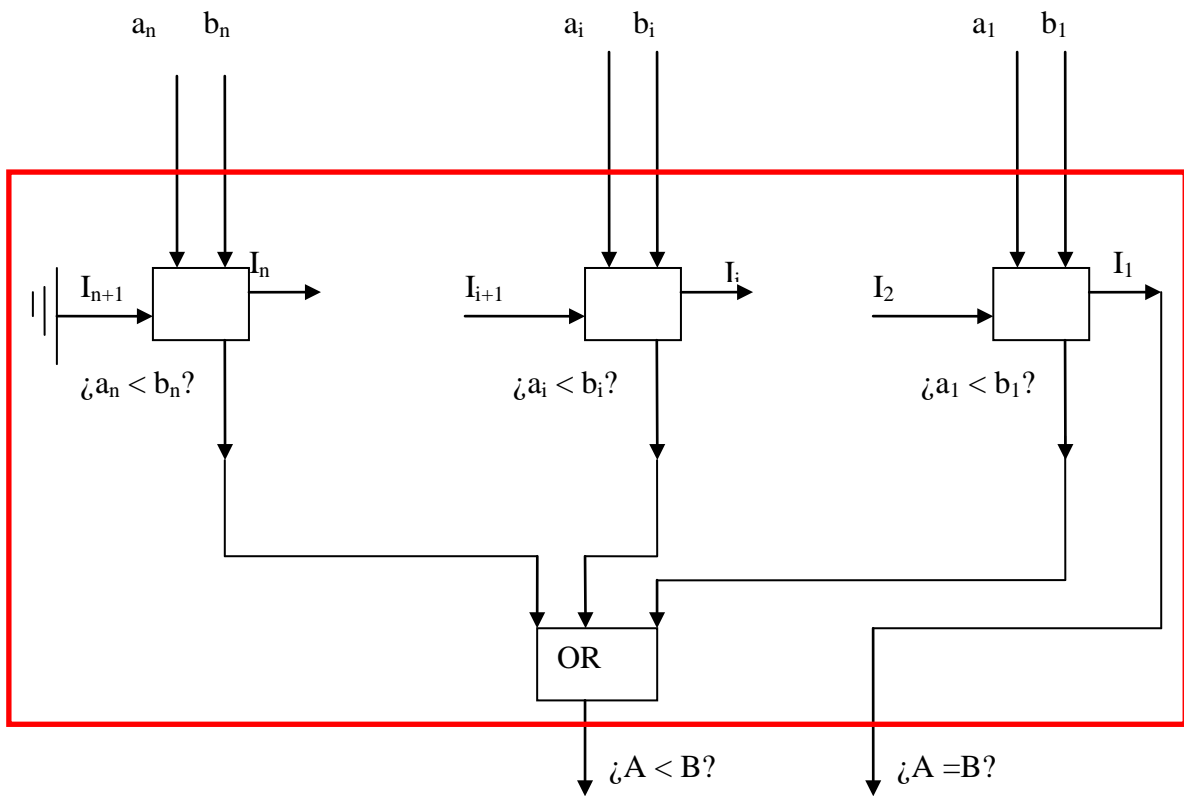


Figura 93

4.2. Circuitos comerciales: 7485

Un ejemplo de MSI comercial es el 7485 disponible en diversas familias. Este dispositivo dispone de tres entradas en cascada que permiten utilizar varios comparadores en cascada para la comparación de cualquier número binario con más de cuatro bits. Según el algoritmo de comparación, la diferenciación entre dos números viene dada por la diferenciación de los bits más significativos, por lo tanto, el comparador que contenga los bits de mayor peso será el que proporcione la salida. La configuración de las entradas en cascada del comparador de menor peso deberá ser tal que dicho comparador deba realizar la comparación entre sus bits de entrada. Por lo tanto, la entrada $A=B$ deberá tener un nivel alto según la tabla de la Figura 95.

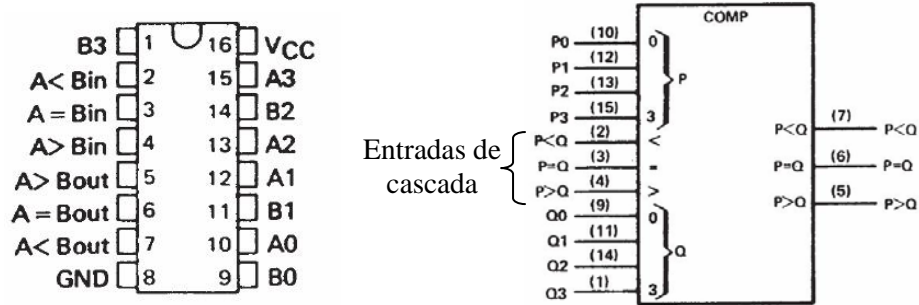


Figura 94

COMPARING INPUTS				CASCADING INPUTS			OUTPUTS		
A3, B3	A2, B2	A1, B1	A0, B0	A > B	A < B	A = B	A > B	A < B	A = B
A3 > B3	X	X	X	X	X	X	H	L	L
A3 < B3	X	X	X	X	X	X	L	H	L
A3 = B3	A2 > B2	X	X	X	X	X	H	L	L
A3 = B3	A2 < B2	X	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 > B1	X	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 < B1	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 > B0	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 < B0	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	L	L	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	H	L	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	X	X	H	L	L	H
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	H	L	L	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	L	L	H	H	L

Figura 95

Diagrama lógico

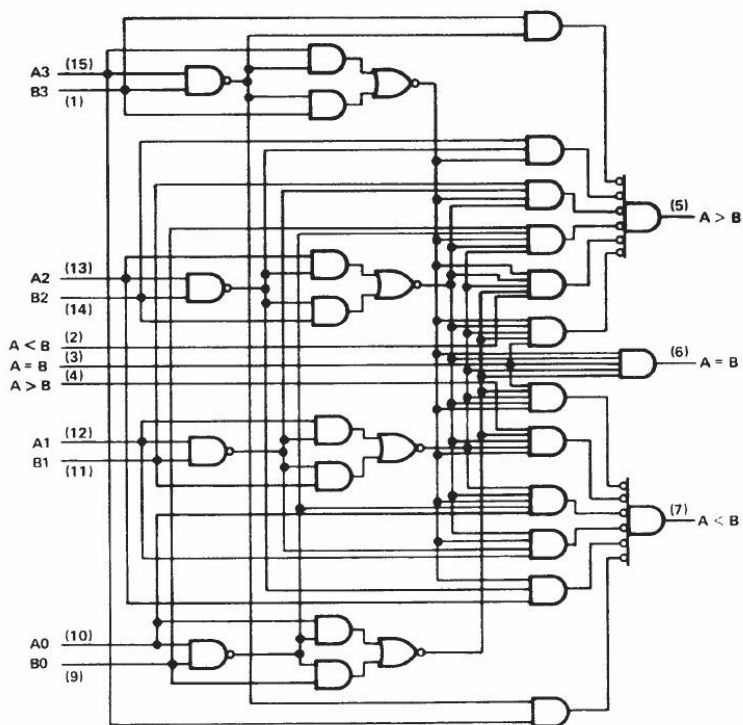


Figura 96

4.3. Descripción VHDL

```
library ieee;
use ieee.std_logic_1164.all;
entity compara is port(
a,b: in std_logic_vector(1 downto 0);
antes_mayor: in std_logic;
antes_menor: in std_logic;
antes_igual: in std_logic
mayor: out std_logic;
menor: out std_logic;
igual: out std_logic);
end compara;
1  architecture archicompara of compara is
2  begin
3  process (a, b, antes_igual, antes_mayor, antes_menor)
4  begin
5  if antes_igual = '0' then igual <= antes_igual;
6  mayor <= antes_mayor;
7  menor <= antes_menor;
8  elsif antes_igual = '1' then
9  if a > b then mayor <= '1';
10  menor <= '0';
11  igual <= '0';
12  elsif a < b then mayor <= '0';
13  menor <= '1';
14  igual <= '0';
15  elsif a = b then mayor <= '0';
16  menor <= '0';
17  igual <= '1';
18  end if;
19  end if;
20  end process;
21  end archicompara;
```