

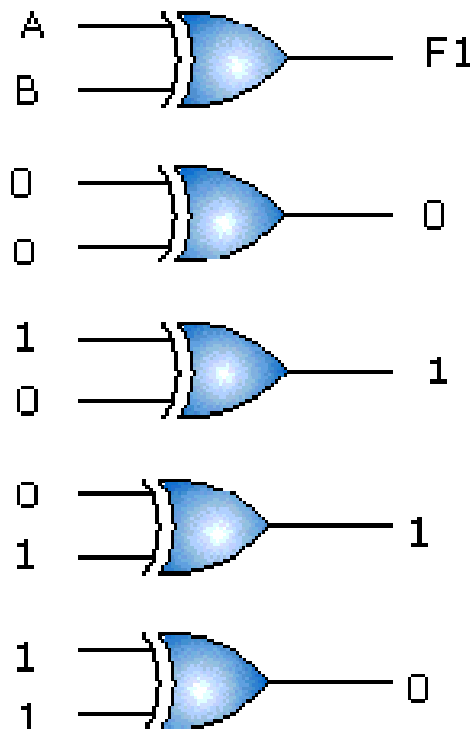
Comparadores

- Un circuito digital comparador realiza la comparación de dos palabras A y B de N bits tomadas como un número entero sin signo e indica si son iguales o si una es mayor que otra en tres salidas $A = B$, $A > B$ y $A < B$. Bajo cualesquiera valores de A y B una y sólo una de las salidas estará a 1, permaneciendo las otras dos salidas a 0.



Comparador de un Bit

- La comparación de dos bits se puede realizar por medio de una puerta XOR o una *XNOR*. La salida del circuito es *1* si sus dos bits de entrada son diferentes y *0* si son iguales.



Entradas: A y B		Salida: C
A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

Comparador de dos Bits

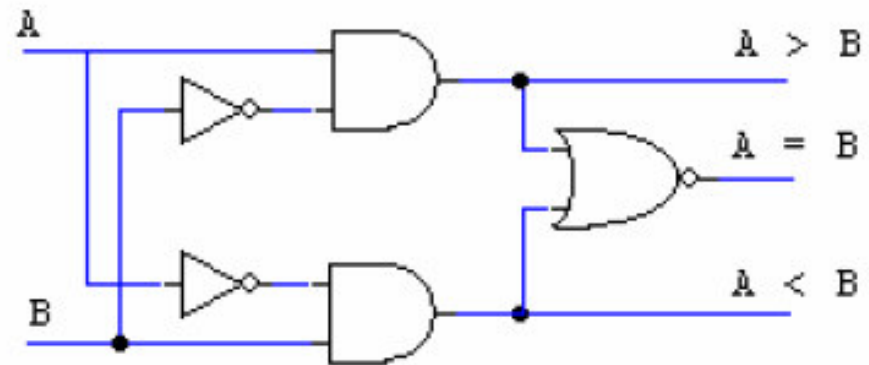
- Para unos operandos A y B de un bit se puede desarrollar un comparador de la siguiente tabla:

A	B	$A = B$	$A > B$	$A < B$
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

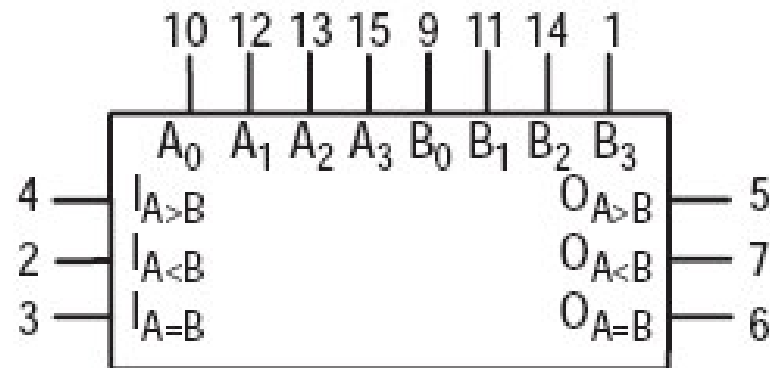
$$(A = B) = \overline{A \oplus B} = \overline{A \overline{B} + \overline{A} B}$$

$$(A > B) = A \overline{B}$$

$$(A < B) = \overline{A} B$$



- En la práctica se encuentran comparadores de números de 4 a 8 bits, que disponen de otras entradas para poder realizar la comparación entre números de más bits.
- Un ejemplo es el 74LS85. Es un comparador de números de 4 bits pero, que además, tiene otras 3 entradas, llamadas entradas de expansión que nos permite conectar varios comparadores en cascada.



$V_{CC} = \text{PIN } 16$
 $\text{GND} = \text{PIN } 8$

Diagrama lógico del 74LS85

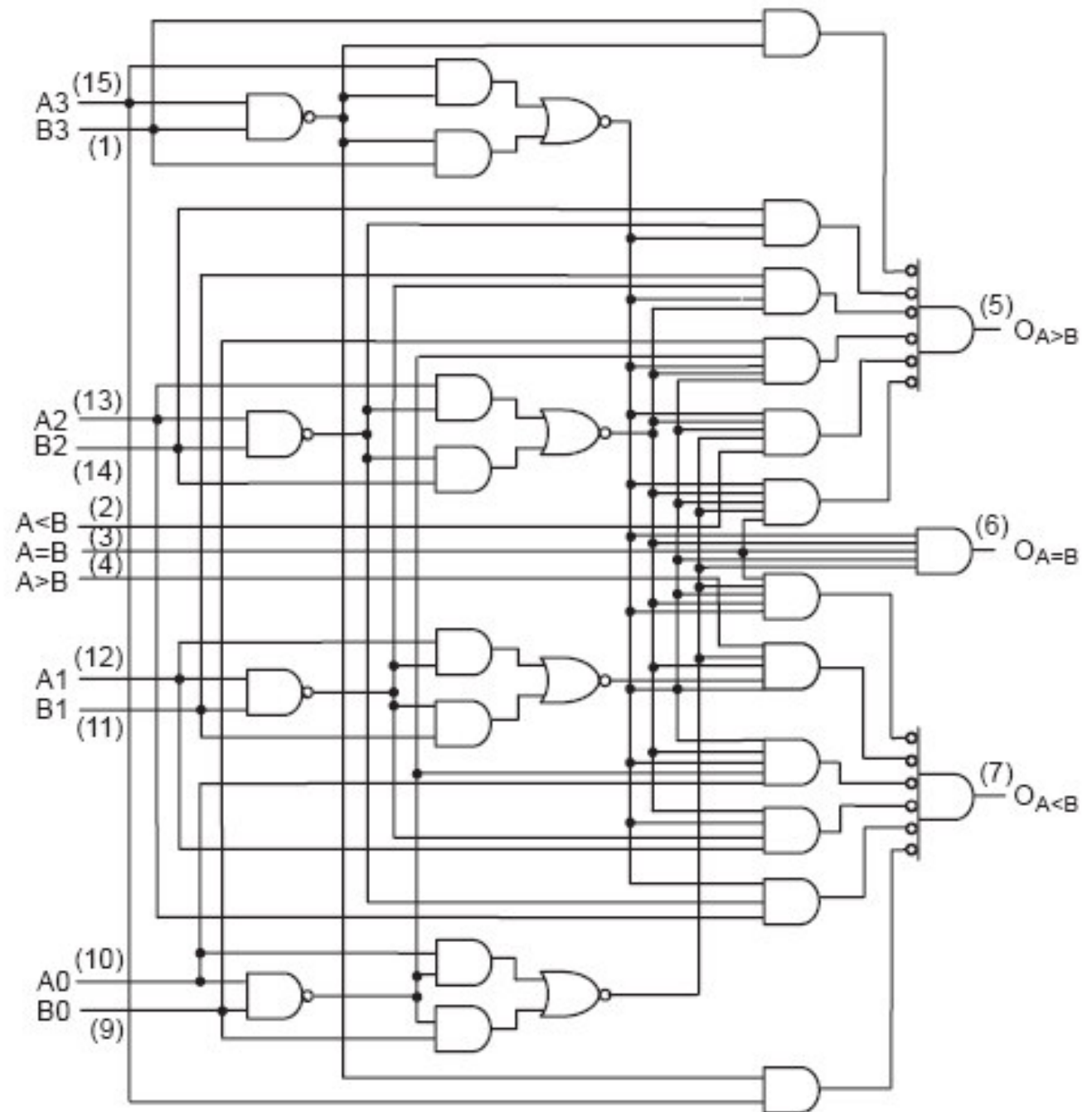


Tabla de verdad del comparador de 4 bits 74LS85.

COMPARING INPUTS				CASCADING INPUTS			OUTPUTS		
A ₃ ,B ₃	A ₂ ,B ₂	A ₁ ,B ₁	A ₀ ,B ₀	I _{A>B}	I _{A<B}	I _{A=B}	O _{A>B}	O _{A<B}	O _{A=B}
A ₃ >B ₃	X	X	X	X	X	X	H	L	L
A ₃ <B ₃	X	X	X	X	X	X	L	H	L
A ₃ =B ₃	A ₂ >B ₂	X	X	X	X	X	H	L	L
A ₃ =B ₃	A ₂ <B ₂	X	X	X	X	X	L	H	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ >B ₁	X	X	X	X	H	L	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ <B ₁	X	X	X	X	L	H	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ >B ₀	X	X	X	H	L	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ <B ₀	X	X	X	L	H	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	H	L	L	H	L	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	L	H	L	L	H	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	X	X	H	L	L	H
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	H	H	L	L	L	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	L	L	L	H	H	L

Comparador para N-Bits

- $A = B$, si $A_i = B_i$ para cada bit i de los operandos A y B .
- $A > B$, si para un bit i $A_i > B_i$ siendo $A_j = B_j$ para todo $j > i$.
- $A < B$, si para un bit i $A_i < B_i$ siendo $A_j = B_j$ para todo $j > i$.

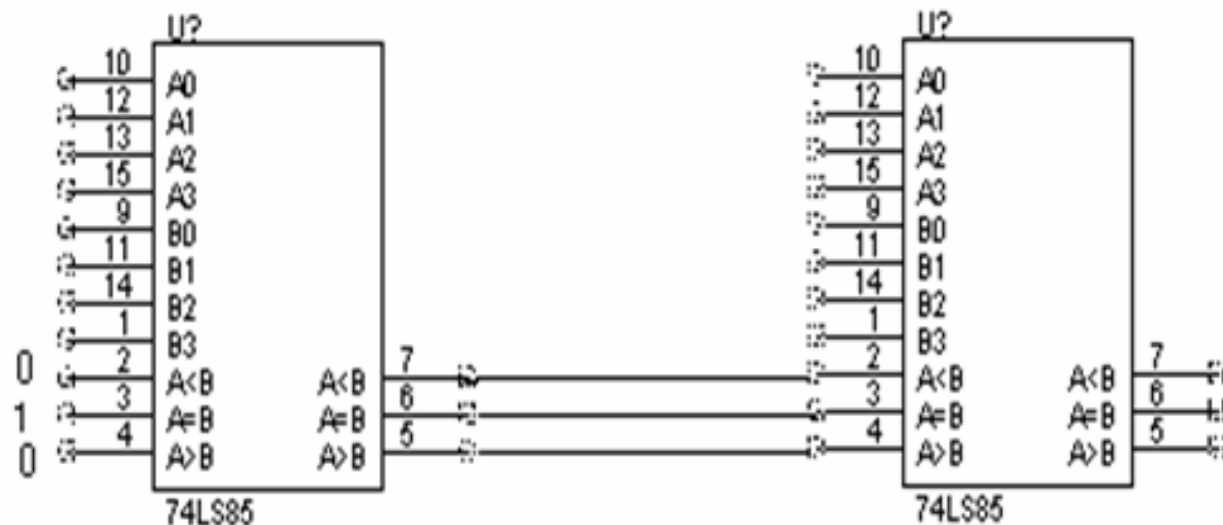
Para operandos A y B de 4 bits, conociendo las expresión de cuando A_i es igual, mayor o menor que B_i , las expresiones lógicas correspondientes quedan así:

$$(A = B) = \overline{A_3 \oplus B_3} \overline{A_2 \oplus B_2} \overline{A_1 \oplus B_1} \overline{A_0 \oplus B_0}$$

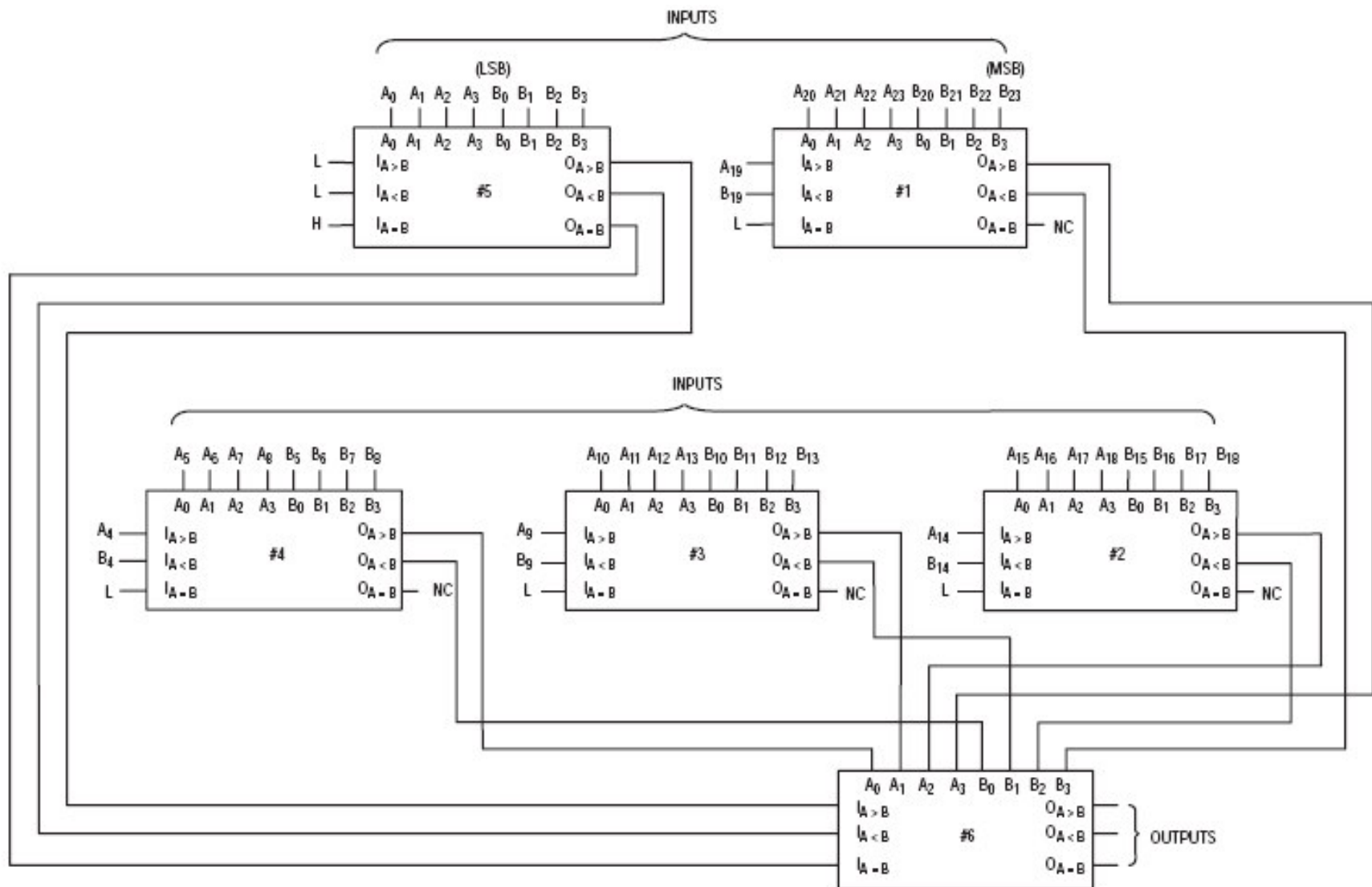
$$(A > B) = A_3 \overline{B_3} + \overline{A_3 \oplus B_3} A_2 \overline{B_2} + \overline{A_3 \oplus B_3} \overline{A_2 \oplus B_2} A_1 \overline{B_1} + \\ + \overline{A_3 \oplus B_3} \overline{A_2 \oplus B_2} \overline{A_1 \oplus B_1} A_0 \overline{B_0}$$

$$(A < B) = \overline{A_3} B_3 + \overline{A_3 \oplus B_3} \overline{A_2} B_2 + \overline{A_3 \oplus B_3} \overline{A_2 \oplus B_2} \overline{A_1} B_1 + \\ + \overline{A_3 \oplus B_3} \overline{A_2 \oplus B_2} \overline{A_1 \oplus B_1} A_0 B_0$$

- Con este comparador podemos no solo comparar números de 4 bits, sino del número de bits que queramos utilizando las entradas de expansión que nos permite conectar varios comparadores en cascada. Un ejemplo seria el siguiente.
- Diseñar un comparador de 2 números de 8 bits utilizando 2 chips 74LS85.



- Utilizando el mismo chip 74LS85, el fabricante nos muestra un comparador de dos números de 24 bits.



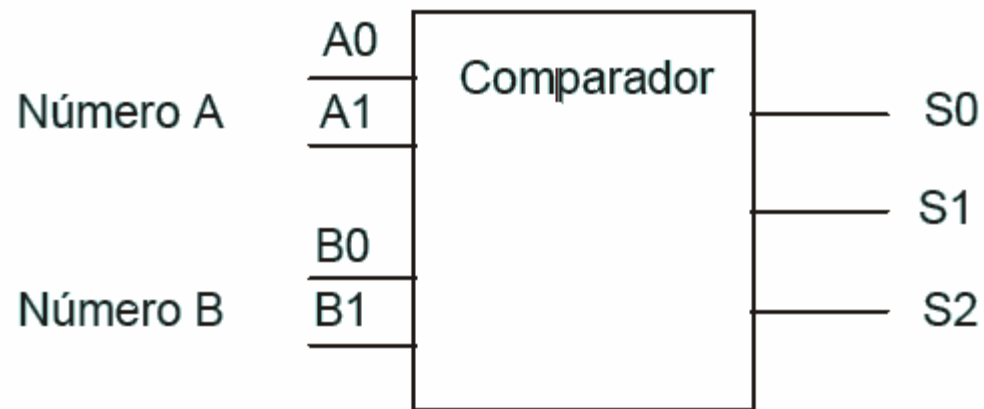
VHDL de un comparador de 4 bits

```
library ieee;  
use ieee.std_logic_1164.all;  
entity compara4 is  
    port (A, B: in std_logic_vector(3 downto 0);  
        AGB, ALB, AEB: out std_logic);  
end compara4;  
architecture comportamiento of compara4 is  
begin  
    AGB <= '1' when ( A > B ) else '0'; -- A > B, salida AGB a 1, resto a 0  
    ALB <= '1' when ( A < B ) else '0'; -- A < B, salida ALB a 1, resto a 0  
    AEB <= '1' when ( A = B ) else '0'; -- A = B, salida AEB a 1, resto a 0  
end comportamiento;
```

Problemas

Problema: 1

El circuito de la figura es un comparador binario de dos numeros (A y B) de dos bits. Las salidas (S0 S1 y S2) toman el valor logico "1" cuando $A > B$, $A < B$ y $A = B$, respectivamente. Obtenga las funciones logicas de cada salida.



Realizamos la tabla de verdad y expresamos las funciones canónicas para las tres salidas.

2^1	2^0	2^1	2^0	A>B	A<B	A=B
A ₁	A ₀	B ₁	B ₀	S ₀	S ₁	S ₂
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	1	0	0
0	1	0	1	0	0	1
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	0	1
1	0	1	1	0	1	0
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	0	1

Funciones resultantes:

$$\begin{aligned} S_0 = & \overline{A_1} \cdot A_0 \cdot \overline{B_1} \cdot \overline{B_0} + A_1 \cdot \overline{A_0} \cdot \overline{B_1} \cdot \overline{B_0} + A_1 \cdot \overline{A_0} \cdot \overline{B_1} \cdot B_0 + \\ & + A_1 \cdot A_0 \cdot \overline{B_1} \cdot \overline{B_0} + A_1 \cdot A_0 \cdot \overline{B_1} \cdot B_0 + A_1 \cdot A_0 \cdot B_1 \cdot \overline{B_0} \end{aligned}$$

$$\begin{aligned} S_1 = & \overline{A_1} \cdot \overline{A_0} \cdot \overline{B_1} \cdot B_0 + \overline{A_1} \cdot \overline{A_0} \cdot B_1 \cdot \overline{B_0} + \overline{A_1} \cdot \overline{A_0} \cdot B_1 \cdot B_0 + \\ & + \overline{A_1} \cdot A_0 \cdot B_1 \cdot \overline{B_0} + \overline{A_1} \cdot A_0 \cdot B_1 \cdot B_0 + A_1 \cdot \overline{A_0} \cdot B_1 \cdot B_0 \end{aligned}$$

$$\begin{aligned} S_2 = & \overline{A_1} \cdot \overline{A_0} \cdot \overline{B_1} \cdot \overline{B_0} + \overline{A_1} \cdot A_0 \cdot \overline{B_1} \cdot B_0 + A_1 \cdot \overline{A_0} \cdot B_1 \cdot \overline{B_0} + \\ & + A_1 \cdot A_0 \cdot B_1 \cdot B_0 \end{aligned}$$

B_1B_0		00	01	11	10
A_1A_0	00				
	01	1			
	11	1	1		1
	10	1	1		

$$S_0 = A_1 \cdot \overline{B_1} + A_0 \cdot \overline{B_1} \cdot \overline{B_0} + \\ + A_1 \cdot A_0 \cdot \overline{B_0}$$

B_1B_0		00	01	11	10
A_1A_0	00		1	1	1
	01			1	1
	11				
	10			1	

$$S_1 = \overline{A_1} \cdot B_1 + \overline{A_0} \cdot B_1 \cdot B_0 + \\ + \overline{A_1} \cdot \overline{A_0} \cdot B_0$$

B_1B_0	00	01	11	10
A_1A_0				
00	1			
01		1		
11			1	
10				1

$$S_2 = \overline{A_1} \cdot \overline{A_0} \cdot \overline{B_1} \cdot \overline{B_0} + \overline{A_1} \cdot A_0 \cdot \overline{B_1} \cdot B_0 + A_1 \cdot \overline{A_0} \cdot B_1 \cdot \overline{B_0} + A_1 \cdot A_0 \cdot B_1 \cdot B_0$$

Problema: 2

Diseñar un circuito que realice la operación aritmética $Z=A+1$ cuando A sea igual a B y la operación $Z=(A-B)-1$ cuando A sea mayor que B, donde A y B son números binarios de 4 bits, siendo siempre $A \geq B$. Implementar el circuito utilizando como base el sumador 74'83 y otros elementos MSI y puertas lógicas.