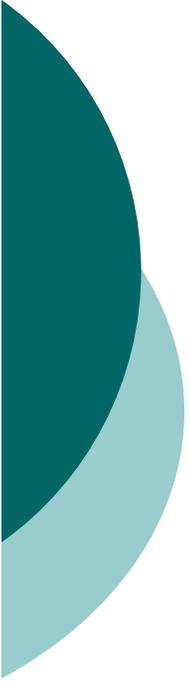


INDICE

- COMPARADOR 1 BIT
- COMPARADOR 4 BITS
 - Fundamentos
 - Familia Lógica
 - Descripción VHDL
- COMPARADOR 16 BITS
- COMPARADOR 24 BITS



OPERACIONES COMPARADOR

○ $A > B$

○ $A = B$

○ $A < B$

○ $A \geq B$

○ $A \leq B$

$A > B$

$A = B$

COMPARACION:

A(1bit) B(1bit)

Sean dos números A y B de 1 bit cada uno, para saber si son =, > o < utilizamos la siguiente tabla de verdad:

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

Si obtenemos la función de cada una de las salidas:

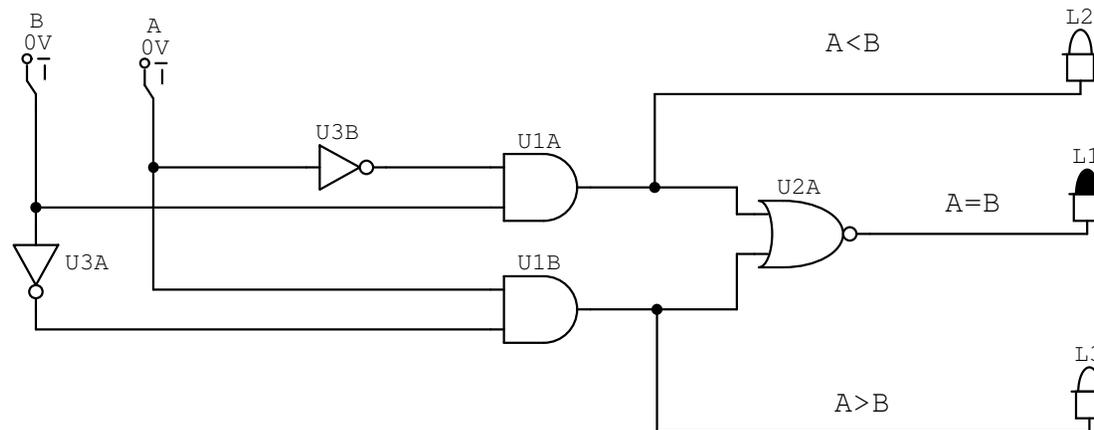
$$F(A=B) = \overline{A \oplus B}$$

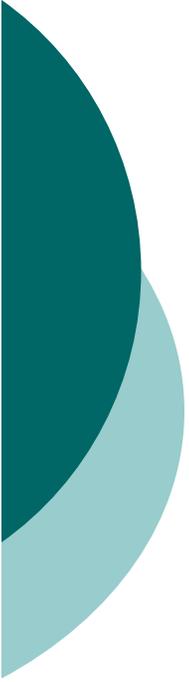
$$F(A>B) = \overline{A} * B$$

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

COMPARACION: A(1bit) B(1bit)

El circuito de un comparador de 1 bit implementado en puertas lógicas será:





COMPARACION:

A(4bit) B(4bit)

FUNDAMENTO

- $A=B$

Deben ser iguales cada uno de sus bits

- $A>B$

Recorremos todos los bits de A y B desde el mas significativo, hasta que A sea 1 y B sea 0

- $A<B$

Recorremos todos los bits de A y B desde el mas significativo, hasta que A sea 0 y B sea 1

COMPARACION:

A(4bit) B(4bit)

Ejemplo1:

A(1010)

B(1001)

1 0 1 0
1 0 0 1

A > B

NOTA:

SE ESTUDIA BIT A BIT Y EN EL MOMENTO QUE A SEA 1 Y B O SE CUMPLE A > B Y VICEVERSA

SI NO SE DIERAN NINGUNA DE ESTAS SERIA LOS NUMEROS IGUALES

Ejemplo2:

A(1010)

B(1010)

1 0 1 0
1 0 1 0

A = B

Ejemplo3:

A(1010)

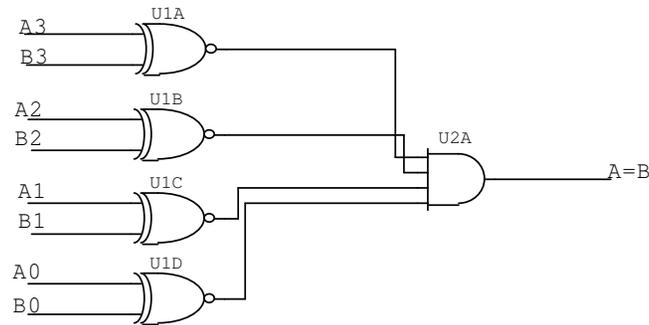
B(1110)

1 0 1 0
1 1 1 0

A < B

COMPARACION: A(4bit) B(4bit)

○ OPERACIÓN $A = B$



LA FUNCION LOGICA DE LA OPERACIÓN $A = B$ ES :

$$F = \overline{A3 \oplus B3} * \overline{A2 \oplus B2} * \overline{A1 \oplus B1} * \overline{A0 \oplus B0}$$



COMPARACION:

A(4bit) B(4bit)

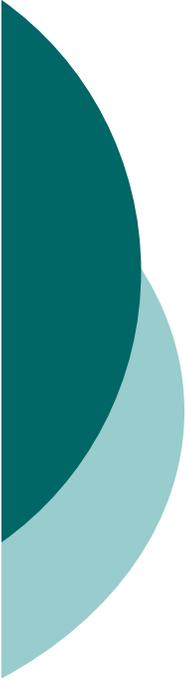
○ OPERACIÓN $A > B$

Partimos de la misma base, es decir, utilizar dos números de 4 bits cada uno.

Para que un numero sea mayor que otro expresado en algebra Booleana sería:

$$F = A_3 * \overline{B_3} + \overline{A_3 \oplus B_3} * A_2 * \overline{B_2} + \overline{A_3 \oplus B_3} * \overline{A_2 \oplus B_2} * A_1 * \overline{B_1} + \overline{A_3 \oplus B_3} * \overline{A_2 \oplus B_2} * \overline{A_1 \oplus B_1} * A_0 * \overline{B_0}$$

Así se va comparando bit a bit teniendo en cuenta que compara de mas significativo a menos significativo



COMPARACION:

A(4bit) B(4bit)

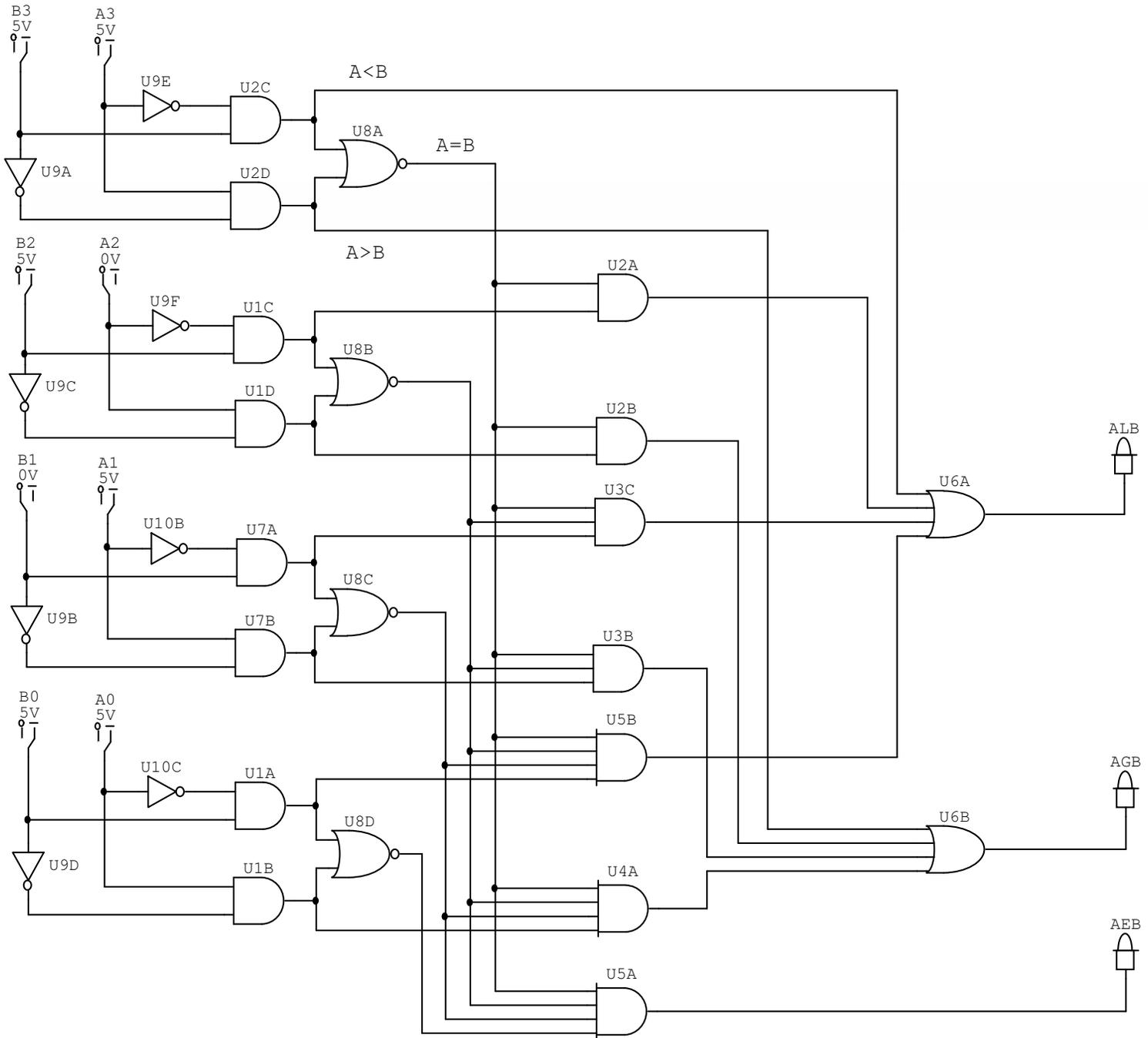
○ OPERACIÓN $A < B$

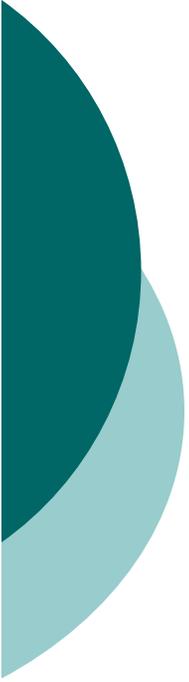
Partimos de la misma base, es decir, utilizar dos números de 4 bits cada uno.

Para que un numero sea mayor que otro expresado en algebra Booleana sería:

$$F = B_3 * \overline{A_3} + \overline{B_3 \oplus A_3} * B_2 * \overline{A_2} + \overline{B_3 \oplus A_3} * \overline{B_2 \oplus A_2} * B_1 * \overline{A_1} + \overline{B_3 \oplus A_3} * \overline{B_2 \oplus A_2} * \overline{B_1 \oplus A_1} * B_0 * \overline{A_0}$$

Así se va comparando bit a bit teniendo en cuenta que compara de mas significativo a menos significativo





VHDL

- Library ieee;
- Use ieee.std_logic_1164.all; --Estas sentencias reconocen el tipo std_logic
- Entity compara is
- Port (A,B: in std_logic_vector(4 downto 1);
- Z: out std_logic_vector(3 downto 1));
- End compara;
- Architecture comportamiento of compara is
- Begin
- Process(A,B)
- Begin
- If (A>B) then
- Z<="100";
- elsif (A<B) then
- Z<="010";
- else
- Z<="001";
- End if;
- End process;
- End comportamiento;

- 
-
- Library ieee;
 - Use ieee.std_logic_1164.all; --Estas sentencias reconocen el tipo std_logic
 - Entity compara is
 - Port (A,B: in std_logic_vector(4 downto 1);
 - AGB,ALB,AEB: out std_logic);
 - End compara;
 - Architecture comportamiento of compara is
 - Begin
 - AGB <= '1' when (A>B) else '0'; -- A>B, salida AGB 1, resto a 0
 - ALB <= '1' when (A<B) else '0'; -- A<B, salida ALB 1, resto a 0
 - AEB <= '1' when (A=B) else '0'; -- A=B, salida AEB 1, resto a 0
 - End comportamiento;

Tabla de verdad

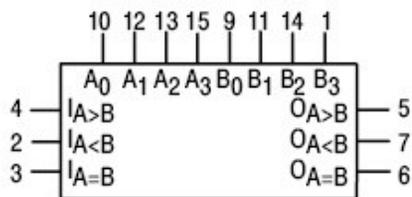
SN74LS85

TRUTH TABLE

COMPARING INPUTS				CASCAING 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

H = HIGH Level
L = LOW Level
X = IMMATERIAL

LOGIC SYMBOL



V_{CC} = PIN 16
GND = PIN 8

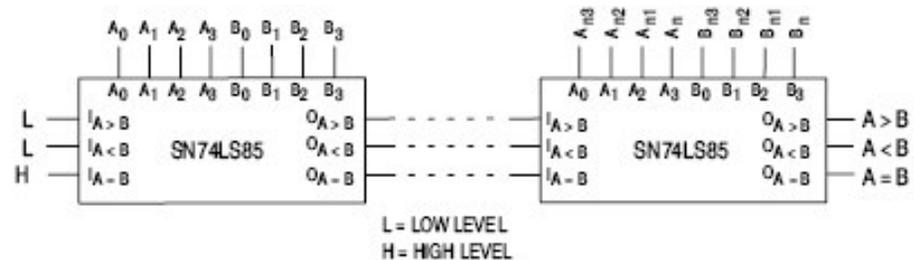
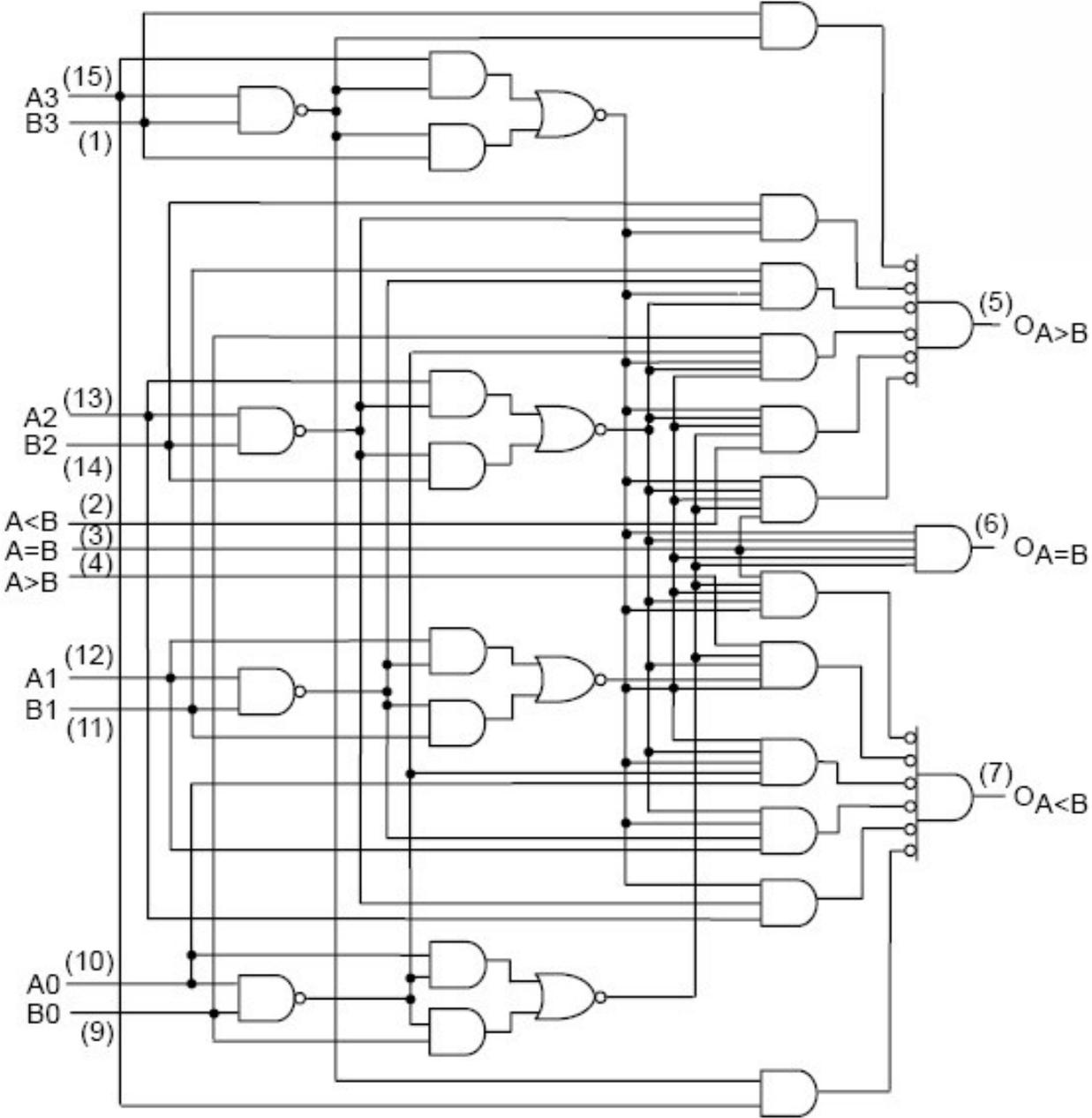
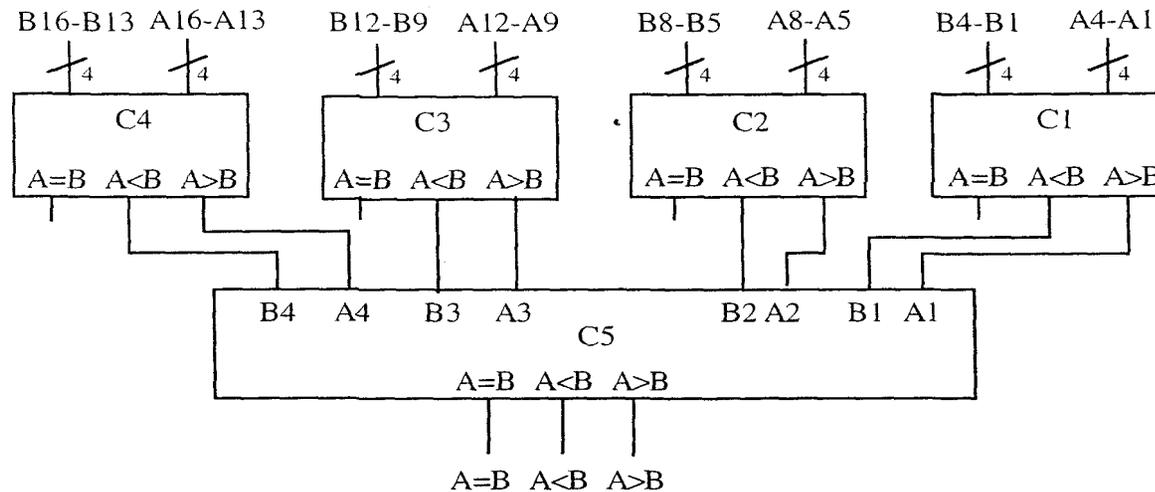


Figure 1. Comparing Two n-Bit Words

LOGIC DIAGRAM



COMPARADOR DE 16 BITS



✦ Para que $A = B$, se necesita que las salidas ($A = B$) de los cuatro comparadores C4-C 1 estén a 1 (todos los bits son iguales en cada grupo de 4 bits).

✦ Para que $A > B$ se necesita que se cumpla una de estas condiciones:

- En C4 la salida ($A > B$) sea 1 {lo que implica a su vez que $A < B$ es 0}
- En C4 las salidas ($A > B$) y ($A < B$) son 0 (equivalente a que la salida ($A = B$) es 1) y en C3 La salida($A > B$) es 1.
- En C4 y C3 Las salidas { $A > B$ } y ($A < B$) son 0 y en C2 la salida ($A > B$) es 1
- En C4, C3 y C2 las salidas ($A > B$) y ($A < B$) son 0 y en C1 la salida ($A > B$) es 1

✦ Para que $A < B$ se realiza el mismo razonamiento que en el punto anterior intercambiando los ($A > B$) y ($A < B$).

COMPARADOR DE 24 BITS

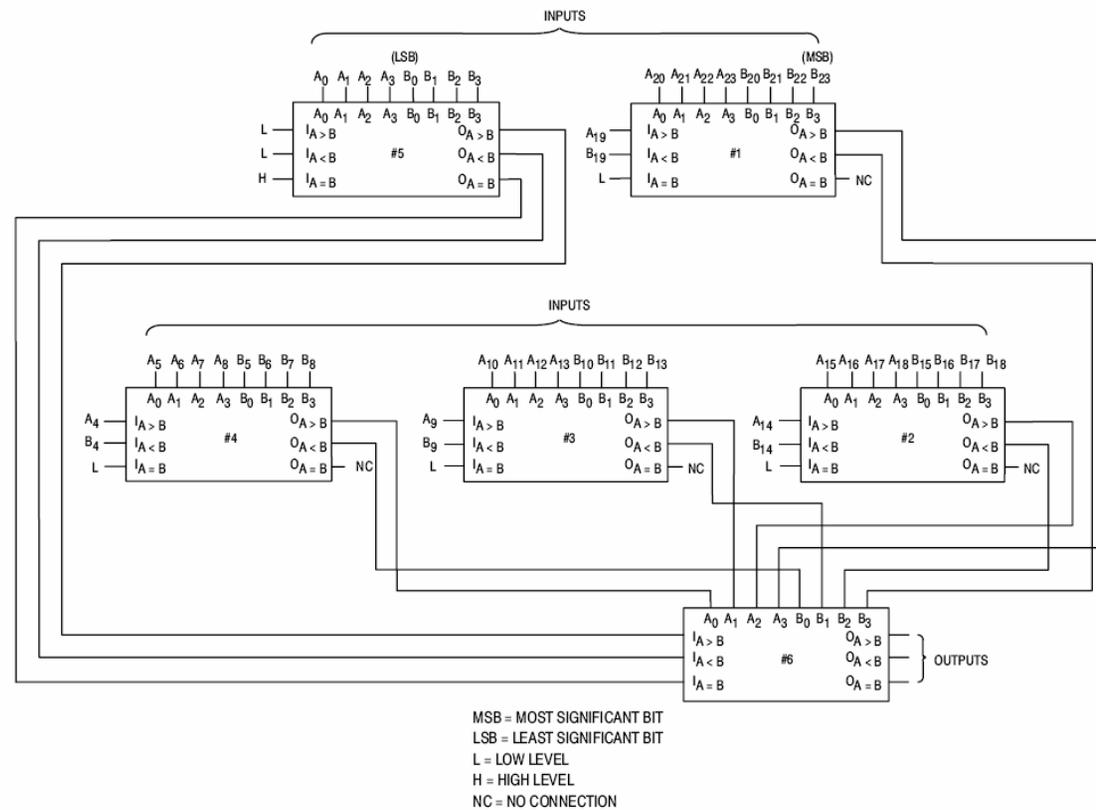
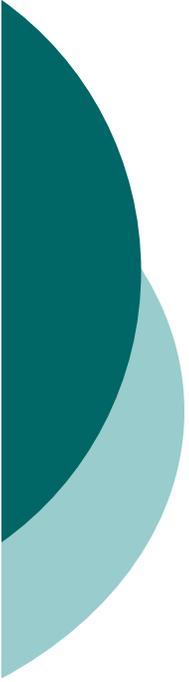


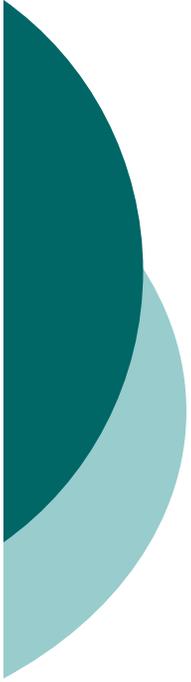
Figure 2. Comparison of Two 24-Bit Words

LAS ENTRADAS I SE UTILIZAN COMO ENTRADAS DE EXPANSION PARA ESTE TIPO DE COMPARADOR QUE NECESITA ENTRADAS EN CASCADA



EJERCICIOS PROPUESTOS

1. Realizar el diseño del siguiente circuito tomando como base el sumador de números binarios de cuatro bits 74'83, utilizando puertas lógicas cuando sea necesario.
Diseñar un comparador de dos números A y B de cuatro bits. El circuito debe generar tres salidas:
O1 ($A = B$), O2 ($A > B$), O3 ($A < B$).
2. Diseñar un circuito comparador COMP13 de números binarios de 13 bits utilizando únicamente cuatro comparadores 74'85 de cuatro bits y un máximo de dos niveles de lógica
3. Se quieren diseñar circuitos digitales que realicen la comparación de dos números binarios con signo X e Y. Se deben obtener tres salidas que indiquen cuando $X = Y$, $X > Y$ ó $X < Y$. Se debe utilizar en lo posible comparadores comerciales como el 74'85, y otros elementos lógicos cuando sea necesario. Indicar en cada caso el razonamiento o las ecuaciones lógicas que llevan al diseño final.



Recordar: los números positivos siempre son mayores que los negativos; entre números positivos el mayor es el de mayor valor absoluto ($5 > 3$), entre números negativos el mayor es el de menor valor absoluto ($-3 > -5$).

- a) Suponer $X (x_3x_2x_1x_0)$ e $Y (y_3y_2y_1y_0)$ de 4 bits en complemento-2.
- b) Suponer $X (S_x x_3x_2x_1x_0)$ e $Y (S_y y_3y_2y_1y_0)$ de cinco bits en formato con bit de signo, donde S_x e S_y son los signos de X y de Y : 0 positivo, 1 negativo. Los otros bits contienen los módulos M_x y M_y de cada número en código binario. Para simplificar un poco el problema, suponer que existe el $+0$ pero no existe el -0 (eso evita el problema de evaluar $-0 = +0$).