

# CUADRO DE ALARMAS

Fundamentos de diseño digital  
Diseño de Sistemas Digitales  
Sistemas digitales

---



# Cuadro de Alarmas



La práctica de “Cuadro de alarmas” consiste en un sistema digital que realiza las siguientes funciones:

- 1) Selecciona alguna de las cuatro entradas **A, B, C**, o **D** (todas de dos bits) a través de la entrada de selección **SEL** (También de dos bits).
- 2) Compara la entrada seleccionada con el valor de **REF** (dos bits).
- 3) Muestra el resultado de esta comparación a través de un display de siete segmentos de la siguiente manera.

DATO > REF	
DATO < REF	
DATO = REF	



# Cuadro de Alarmas

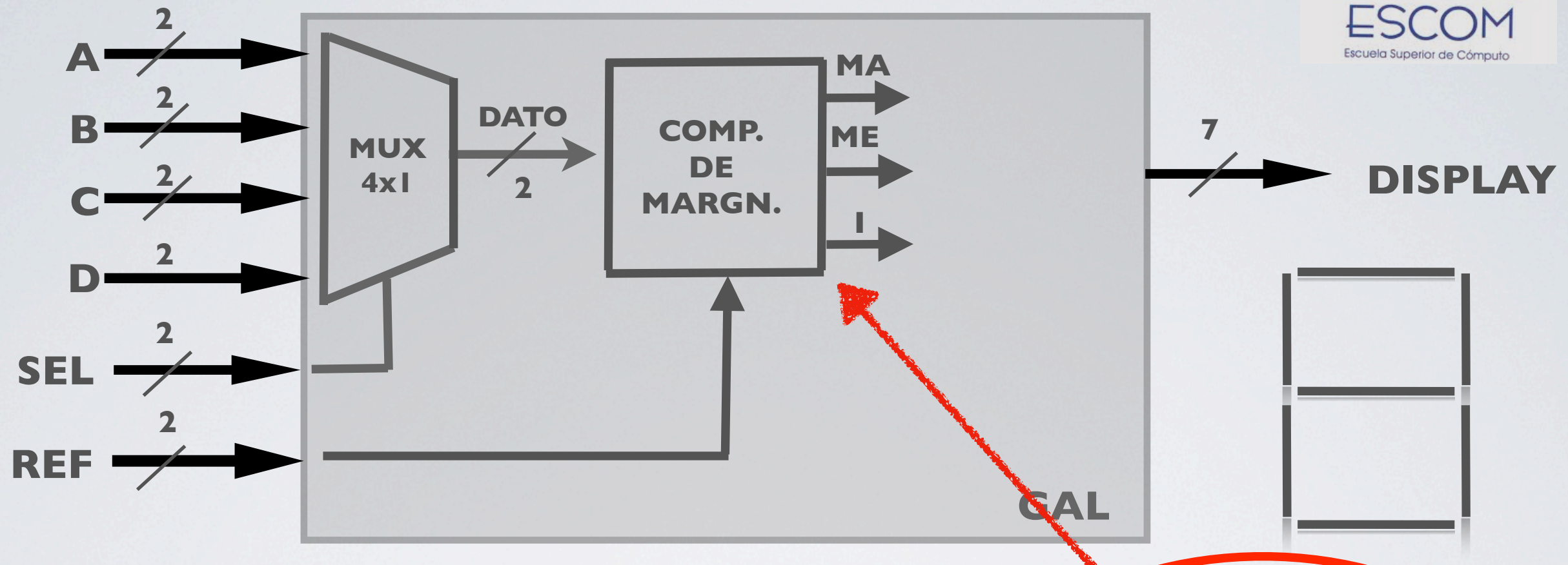


La función de seleccionar una de las cuatros entradas **A**, **B**, **C** y **D** a través de un selector (**SEL**) se puede realizar usando un MUX 4x1 cuya salida será **DATO** (dos bits). La siguiente es su tabla de verdad.

SEL	DATO
00	A
01	B
10	C
11	D



# Cuadro de Alarmas



La función de comparar **DATO** con el valor de entrada **REF** se realiza a través del circuito lógico “Comparador de magnitud”, el cual tiene tres salidas:

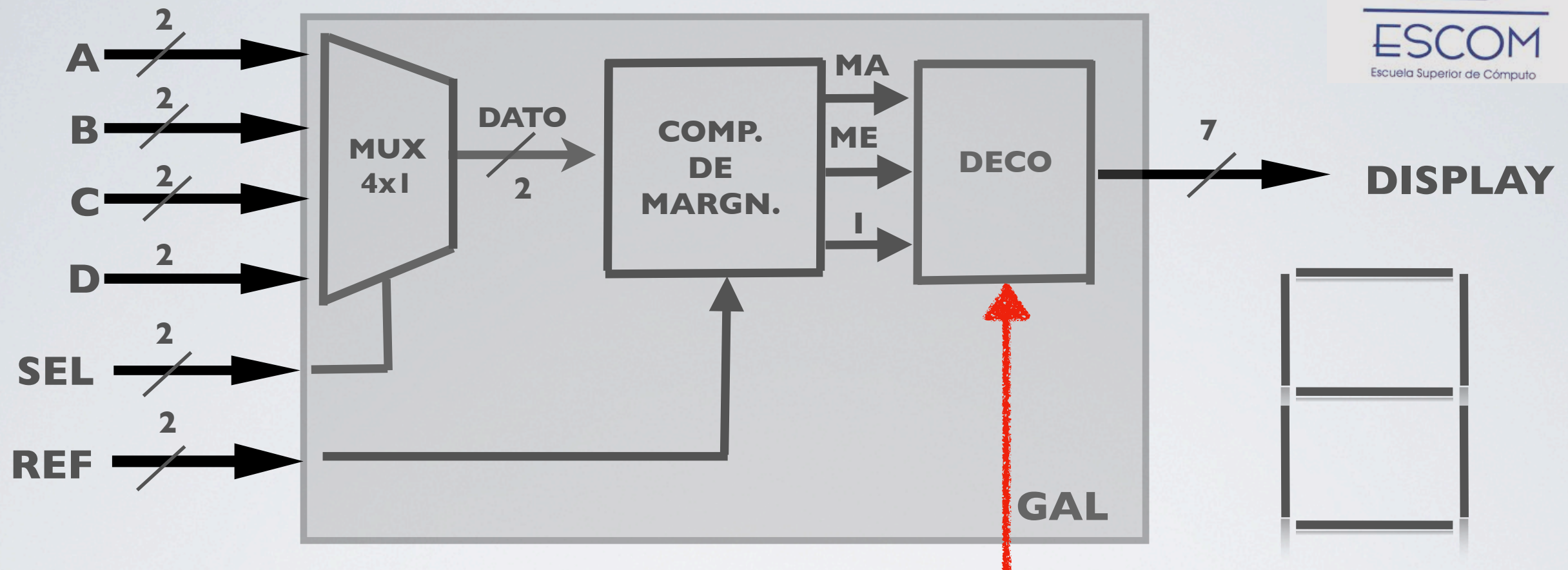
- **MA=1** sólo si **DATO** es mayor que **REF**.
- **ME=1** sólo si **DATO** es menor que **REF**.
- **I=1** sólo si **DATO** es igual que **REF**.

La siguiente es su tabla de verdad.

DATO VS REF	MA	ME	I
DATO > REF	1	0	0
DATO < REF	0	1	0
DATO = REF	0	0	1



# Cuadro de Alarmas



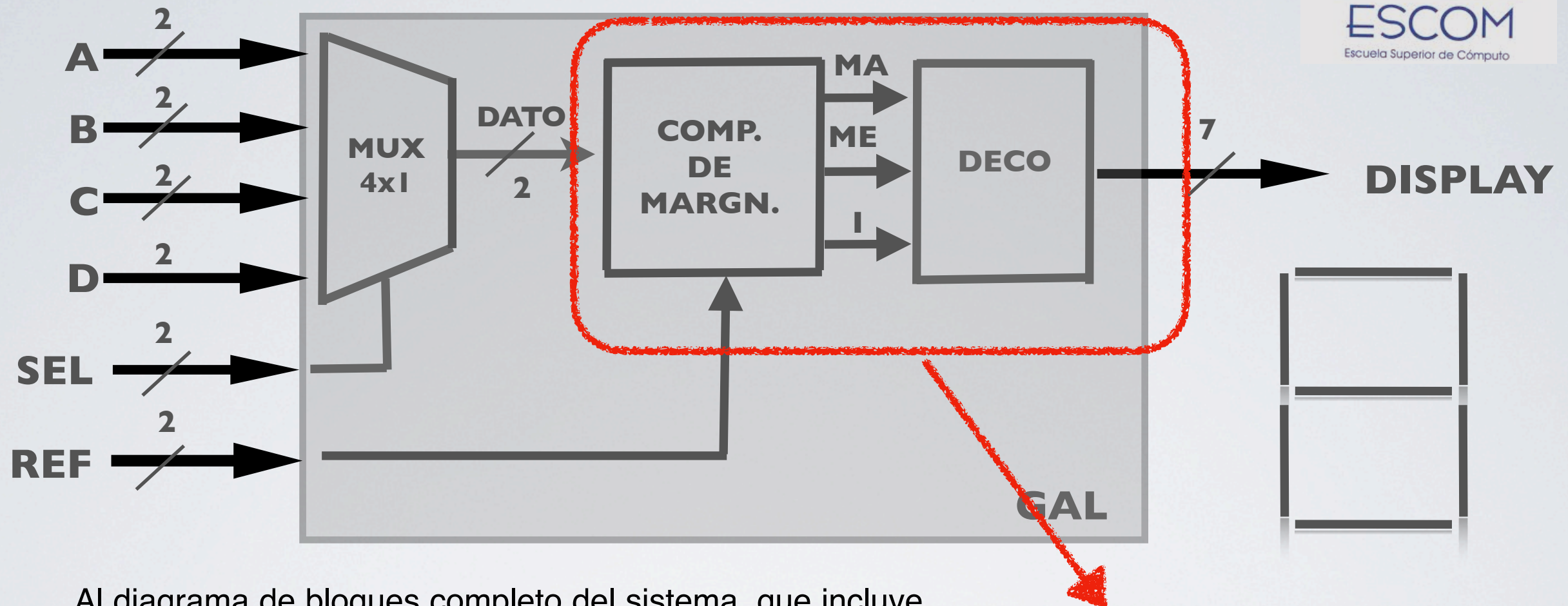
Para convertir la salida del comprador de magnitud en los símbolos  $>$ ,  $<$  e  $=$  se utiliza un decodificador y un display de 7 segmentos. En este caso el decodificador será MAMEI / 7 segmentos y el display será de ánodo común.

La siguiente es su tabla de verdad.

			DISPLAY						
MA	ME	I	A	B	C	D	E	F	G
1	0	0				┌			
						└			
0	1	0				┌			
						└			
1	0	0				┌			
						└			



# Cuadro de Alarmas



Al diagrama de bloques completo del sistema, que incluye la información sobre cada bloque y el flujo de los datos desde la entrada hasta la salida se le llama **Arquitectura**.

Esta arquitectura se puede modificar uniendo los bloques del detector de magnitud y el decodificador, omitiendo las salidas **MA**, **ME** e **I** y asignando directamente a la salida **DISPLAY** la cadena de bits necesaria para visualizar el **>**, **<** e **=**, dependiendo de cada caso, como se muestra en la siguiente tabla de verdad.

	DISPLAY
DATO > REF	3
DATO < REF	7
DATO = REF	11

# Cuadro de Alarmas



Descripción en VHDL de la Arquitectura del Cuadro de Alarmas.

## INSTRUCCIONES CONCURRENTES

```
LIBRARY IEEE;
USE IEEE.TSD_LOG_1164.ALL;

ENTITY CUADRO IS
PORT (A,B,C,D,SEL,REF: IN STD_LOGIC_VECTOR (1 DOWNT0
0);
      DISPLAY: OUT STD_LOG_VECTOR ( 6 DOWNT0 0)
);
END ENTITY;

ARCHITECTURE A_CUADRO OF CUADRO IS
SIGNAL DATO: STD_LOGIC_VECTOR (2 DOWNT0 0);
SIGNAL MA,ME, I: STD_LOGIC;
BEGIN

WHIT SEL SELECT
DATO<= A WHEN "00" ELSE,
      B WHEN "01" ELSE,
      C WHEN "10" ELSE,
      D WHEN OTHERS;

MA <= '1' WHEN (DATO> REF) ELSE '0';
ME <= '1' WHEN (DATO< REF) ELSE '0';
I <= '1' WHEN (DATO= REF) ELSE '0';

DISPLAY <= "0000111" WHEN (MA='1' ) ELSE
           "0110001" WHEN (ME='1') ELSE
           "0110111" ;

END A_CUADRO;
```

## INSTRUCCIONES CONCURRENTES

```
LIBRARY IEEE;
USE IEEE.TSD_LOG_1164.ALL;

ENTITY CUADRO IS
PORT (A,B,C,D,SEL,REF: IN STD_LOGIC_VECTOR (1 DOWNT0
0);
      DISPLAY: OUT STD_LOG_VECTOR ( 6 DOWNT0 0)
);
END ENTITY;

ARCHITECTURE A_CUADRO OF CUADRO IS
SIGNAL DATO: STD_LOGIC_VECTOR (2 DOWNT0 0);
BEGIN

WHIT SEL SELECT
DATO<= A WHEN "00" ELSE,
      B WHEN "01" ELSE,
      C WHEN "10" ELSE,
      D WHEN OTHERS;

DISPLAY <= "0000111"  WHEN (DATO> REF) ELSE
DISPLAY <= "0110001"  WHEN (DATO< REF) ELSE
DISPLAY <= "0110111" ;

END A_CUADRO;
```



# Cuadro de Alarmas



## INSTRUCCIONES SECUENCIALES

```
LIBRARY IEEE;
USE IEEE.TSD_LOG_1164.ALL;

ENTITY CUADRO IS
PORT (A,B,C,D,SEL,REF: IN STD_LOGIC_VECTOR (1 DOWNTO 0);
      DISPLAY: OUT STD_LOG_VECTOR ( 6 DOWNTO 0)
);
END ENTITY;

ARCHITECTURE A_CUADRO OF CUADRO IS
SIGNAL DATO: STD_LOGIC_VECTOR (2 DOWNTO 0);
SIGNAL MA,ME, I: STD_LOGIC;
BEGIN

MUX: PROCESS( SEL)
BEGIN
CASE SEL IS
WHEN "00" => DATO <=A;
WHEN "01" => DATO <=B;
WHEN "10" => DATO <=C;
WHEN OTHERS => DATO <=D;
END CASE;
END PROCESS MUX;

COMP: PROCESS( DATO, REF)
BEGIN
ME<='0';
MA<='0';
I<='0';
IF (DATO> REF) THEN
MA<= '1';
ELSIF (DATO< REF) THEN
ME<= '1';
ELSE
I<= '1';
END IF;
END PROCESS COMP;

PROCESS ( MA,ME,I)
BEGIN
IF (MA='1') THE
DISPLAY <="0000111";
ELSIF (ME='1') THEN
DISPLAY <= "0110001";
ELSE
DISPLAY <= "0110111" ;
END IF;
END PROCESS ;

END A_CUADRO;
```

## INSTRUCCIONES SECUENCIALES

```
LIBRARY IEEE;
USE IEEE.TSD_LOG_1164.ALL;

ENTITY CUADRO IS
PORT (A,B,C,D,SEL,REF: IN STD_LOGIC_VECTOR (1 DOWNTO 0);
      DISPLAY: OUT STD_LOG_VECTOR ( 6 DOWNTO 0)
);
END ENTITY;

ARCHITECTURE A_CUADRO OF CUADRO IS
SIGNAL DATO: STD_LOGIC_VECTOR (2 DOWNTO 0);

BEGIN

MUX: PROCESS( SEL)
BEGIN
CASE SEL IS
WHEN "00" => DATO <=A;
WHEN "01" => DATO <=B;
WHEN "10" => DATO <=C;
WHEN OTHERS => DATO <=D;
END CASE;
END PROCESS MUX;

PROCESS( DATO, REF)
BEGIN
IF (DATO> REF) THEN
DISPLAY <="0000111";
ELSIF (DATO< REF) THEN
DISPLAY <= "0110001";
ELSE
DISPLAY <= "0110111" ;
END IF;
END PROCESS;

END A_CUADRO;
```



# GRACIAS

Fundamentos de diseño digital  
Diseño de Sistemas Digitales  
Sistemas digitales

---