

Los dispositivos de Lógica programable pueden agregar para describir

su funcionamiento el uso de la palabra **TABLE**.

* Primero se define un campo (FIELD) para las variables de entrada y otro para

las variables de salida. Después se especifican una a una las asignaciones de

las salidas ante las diferentes entradas.

* Las condiciones no importa X pueden considerarse para las entradas pero

no para las salidas.

* Una lista de valores de entrada se puede considerar para generar la misma salida.

Para este caso podemos decir que si entrada es decimal podemos definir la salida en binario.

FIELD **entrada** = **[i3..0];** **/\* Campo de entrada\*/**

FIELD **salida** = **[q7..0];** **/\* Campo de salida\*/**

TABLE **entrada** => **salida** **{**

**0=> ‘b’00000001;**

**1 => ‘b’00000010;**

**2 => ‘b’00000100;**

**3 => ‘b’00001000;**

**4 => ‘b’00010000;**

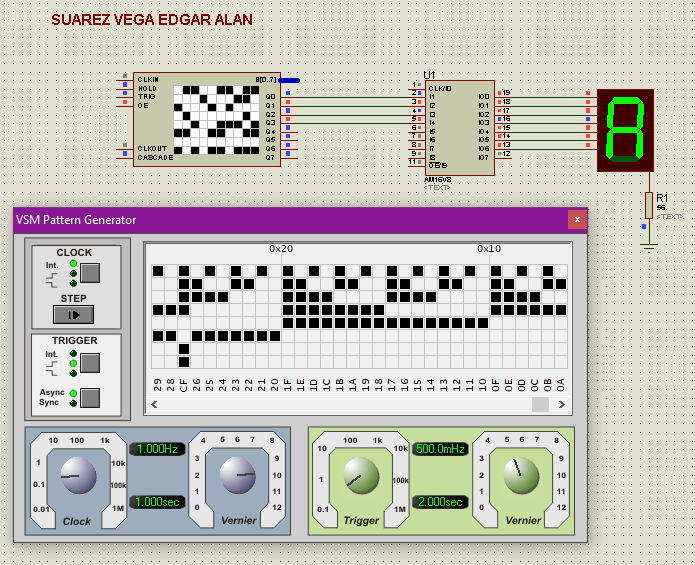
**5 => ‘b’00100000;**

**6 => ‘b’01000000;**

**7 => ‘b’10000000;**

**}**

Para este ejercicio podemos definir un decodificador hexadecimal para un display de 7 segmentos de cátodo común. Esta Tabla muestra las combinaciones en hexadecimal para el display.



Desarrolle el ejercicio en Proteus para que decodifique de 0 a F. **GAL22V10 use en CUPL el código g22v10 para la GAL solicitada.**

FIELD **entrada** = **[i3..0];** **/\* Campo de entrada\*/**

FIELD **salida** = **[q6..0];** **/\* Campo de salida\*/**

TABLE **entrada** => **salida** **{**

**'h'0=> 'h'3f;**

**'h'1 => 'h'06;**

**'h'2 => 'h'5b;**

**'h'3 => 'h'4f;**

**'h'4 => 'h'66;**

**'h'5 => 'h'6d;**

**'h'6 => 'h'7d;**

**'h'7 => 'h'07;**

**'h'8 => 'h'7f;**

**'h'9 => 'h'6f;**

**'h'a => 'h'77;**

**'h'b => 'h'7c;**

**'h'c => 'h'58;**

**'h'd => 'h'5e;**

**'h'e => 'h'79;**

**'h'f => 'h'71;**

**}**

