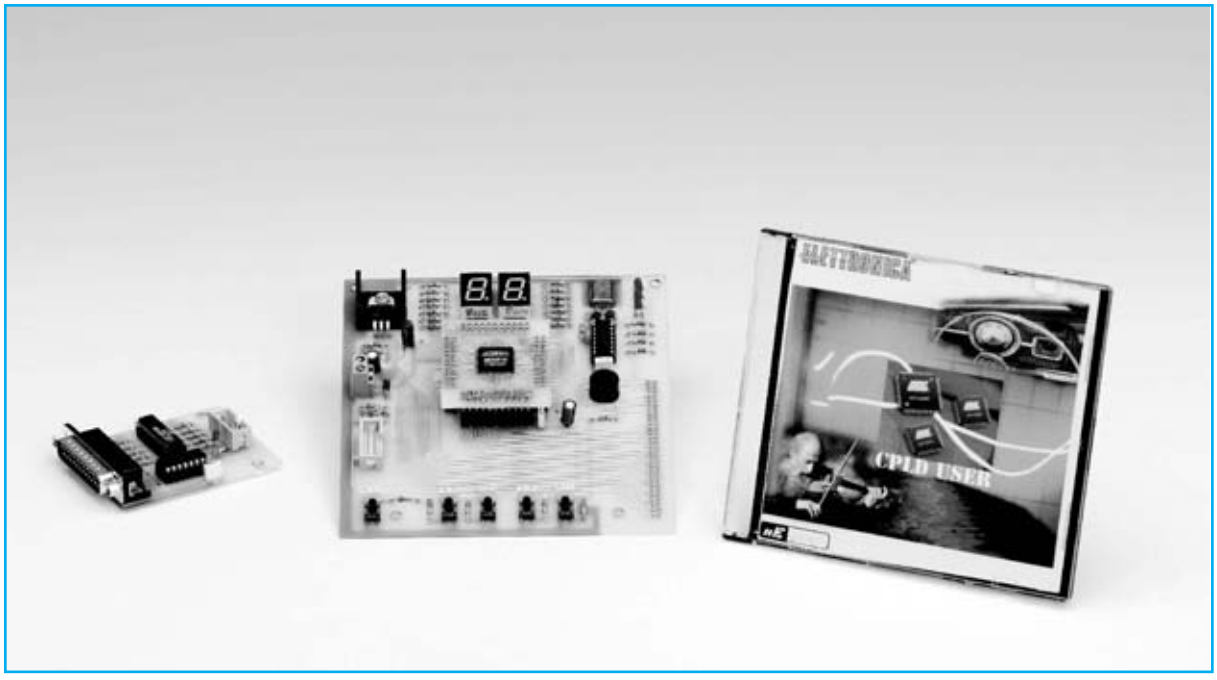


APLICACIONES

Es un hecho constatado que la electrónica digital toma cada vez más ventaja sobre la electrónica analógica. Entre otros dispositivos de desarrollo digital presentamos en la revista Nº269 el Programador de dispositivos CPLD con el fin de reemplazar al lápiz, al papel y al soldador para desarrollar vuestros circuitos digitales. Hoy profundizamos más y os ayudamos a tomar confianza con nuestro dispositivo y con el programa Quartus II de Altera.

Comúnmente el mundo de la electrónica moderna se divide en dos grandes campos, **electrónica analógica** y **electrónica digital**.

Como es conocido, aunque ésta es una primera división, los dos campos son tan vastos que dentro de cada uno hay **numerosas ramas** que abarcan **campos más específicos**.

En la **electrónica analógica**, por ejemplo, nos encontramos la **electrónica de potencia**, **amplificación**, **radiofrecuencia**, **procesamiento de señales**, **audio**, etc.

Igualmente la **electrónica digital** tiene numerosos campos. En esta sección nos ocuparemos de sus fundamentos básicos: La **lógica combinacional**.

EVOLUCIÓN DIGITAL

Si hasta no hace mucho tiempo prácticamente la electrónica digital no existía o era considerada un sector elitista, hoy es prácticamente **imposible** encontrar un circuito que **no incorpore esta tecnología**.

Esto ha sido posible gracias a la **integración** de los **transistores**, el elemento base de cualquier circuito digital, que se ha desarrollado cada vez más siguiendo una famosa ley, denominada **Ley de Moore** (en honor al cofundador de **Intel**), que ya en el lejano **1965** predijo la increíble escalada de la electrónica integrada.

Moore predijo que el **número de transistores integrados** en un **chip individual** se **duplicaría cada año**. La evidencia experimental ha demostrado, y lo sigue haciendo, que el **número de transistores** se **duplica** cada **18-24 meses**, manteniendo en todo caso una **curva exponencial** (ver Fig.1).

Hoy en día se producen muchísimos **microchips** que integran **centenares de millones de transistores**, cada uno de con **tamaño nanométrico** (mil-millonésimas de metro). Las cifras comunes actualmente son **45**, **65**, **80** o **95 nm**.

Hasta hace poco tiempo se expresaba el tamaño en **micrómetros** (millonésimas de metro), de hecho se acuñó el termino de **microelectrónica**. La evolución ha dejado obsoleta esta unidad de medida, ahora los tamaños se expresan en **nanómetros**, de hecho se está implantando el término **nanoelectrónica**.

La velocidad con la que aumenta la disponibilidad de transistores, y por lo tanto la potencia de cálculo, también ha creado un gran **problema**: ¿Cómo puede **explotarse** toda esta **capacidad de cálculo** visto que las capacidades de los diseñadores no pueden crecer proporcionalmente a la complejidad de la lógica?

Como es fácilmente intuible es bastante diferente proyectar un **multiplexor**, un **control** para una **puerta automática** o un **microprocesador**.

PRÁCTICAS con el CPLD

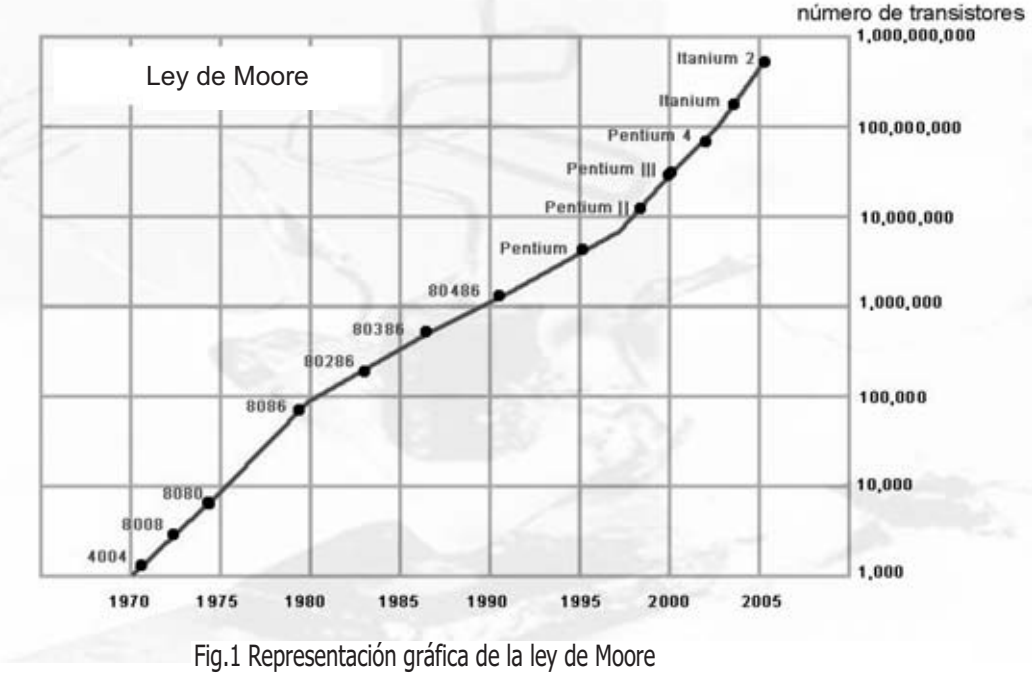
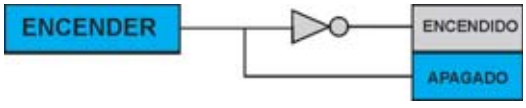


Fig.1 Representación gráfica de la ley de Moore

Para ayudar a los diseñadores se desarrollaron **lenguajes** (como **VHDL**) y **herramientas** (como **Quartus II**) capaces de elevar el **nivel de abstracción** de los circuitos acercando más al l**enguaje humano** la **operativa de diseño** que al lenguaje electrónico de **AND**, **OR**, etc. Así por ejemplo se puede escribir código como el siguiente:

```
if dispositivo_encendido = '1' then
   encender <= '1';
   apagar <= '0';
else
   encender <= '0';
   apagar <= '1';
end if;
```

Este pequeño listado de código puede suponer un **primer contacto** con **VHDL** y **Quartus II**, u otra herramienta equivalente. En este caso cuando la señal "**encender**" esté activa se activará la salida "**encender**" y se desactivará la salida "**apagar**", mientras que cuando esté desactiva la señal "**encender**" se desactivará la salida "**encender**" y se activará la salida "**apagar**".



Obviamente el razonamiento debe extenderse a casos mucho **más complejos**, haciéndose más complejo el código. Por complejo que sea **Quartus II** lo sintetizará **ahorrando muchísimo tiempo** de **desarrollo**, como se puede ver en el ejemplo de la Fig.2.

LÓGICA COMBINACIONAL

Por **lógica combinacional**, **red combinacional** o **circuito combinacional** se entiende, en general, un **circuito digital** con las siguientes características:

- Tiene **una** o **más entradas** y **una** o **más salidas**.

- Representa una **función lógica** cuyos **únicos valores** pueden ser **0** o **1** (**verdadero** o **falso**) (**nivel bajo** o **nivel alto de tensión**).

- Las **salidas** pueden variar exclusivamente en función del valor de las **entradas** y en base a la **función lógica implementada**. **No** hay **realimentaciones** ni elementos de **memoria**, ya que esto supondría diseñar otro tipo de circuitos (**circuitos secuenciales**).

Se habla además de:

- **Lógica positiva**: Cuando las señales se **activan** con **1** (**verdadero**) (niveles **altos** de tensión).

- **Lógica negativa**: Cuando las señales se activan con **0** (**falso**) (niveles **bajos** de tensión).

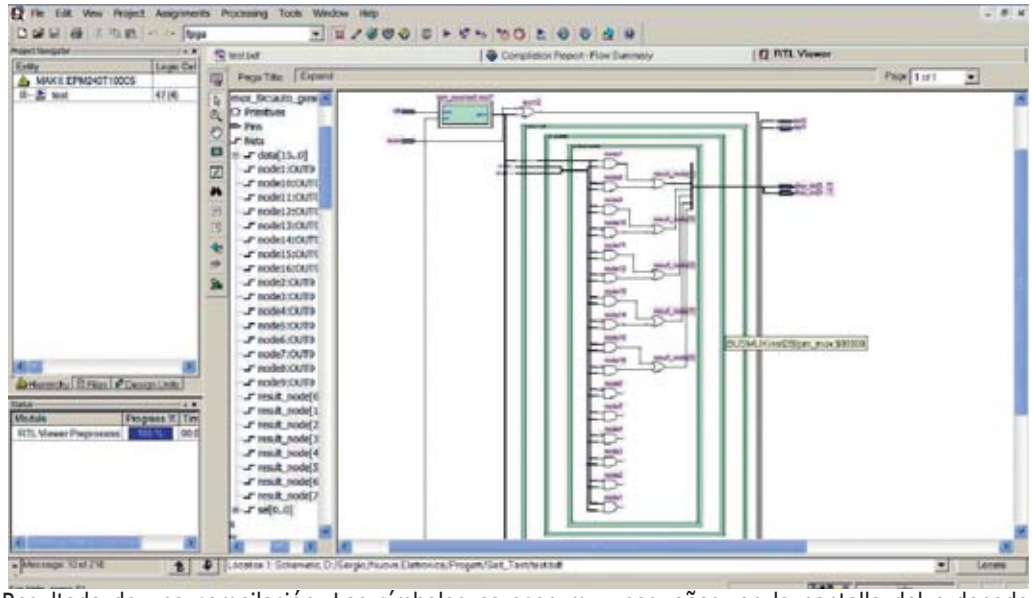


Fig.2 Resultado de una compilación. Los símbolos parecen muy pequeños, en la pantalla del ordenador sus dimensiones son mucho más grandes.

La segunda representación podría parecer a primera vista poco sensata. En realidad en algunas ocasiones es **muy útil**.

Veamos un ejemplo. Supongamos que queremos **encender un diodo LED** con nuestro dispositivo, función que pronto se podrá realizar siguiendo este artículo.

Normalmente, por razones tecnológicas relacionadas con las propiedades del **silicio**, material con el que están construidos el **99%** de los **microchips**, los **transistores** de **canal N** son **más fáciles** de **construir** y soportan **corrientes más elevadas** que los de canal **P**.

Generalmente cuando se **activan** para encender un dispositivo, como un diodo LED o un display, **cortocircuitando** la **salida a masa** (su salida se activa **bajando el nivel**). Por consiguiente el **diodo LED** se **encenderá** cuando la **salida** del dispositivo construido únicamente con transistores de **canal N** esté a **nivel lógico bajo**.

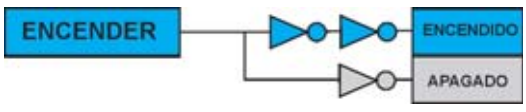
Retomando el tema de la lógica combinacional, para **realizar un circuito** se realizan dos fases:

(1) **Análisis**: Se **expresa** la **función lógica** deseada en una forma adecuada (álgebra booleana, VHDL, etc.).

(**2)** **Síntesis**: La función anteriormente descrita se **implementa** mediante un **circuito digital** constituido por **puertas lógicas** más o menos elementales (AND, OR, NOT, etc.).

El paso de la fase análisis a la fase de síntesis **no es único**. Existen **varios métodos** de resolver el mismo planteamiento ya que en electrónica pueden existir **diferentes formas** de implementar la **misma función lógica**.

Veamos en sencillo ejemplo: La pequeña porción de código anteriormente expuesta y esquematizada puede ser implementada, indudablemente de forma **no optimizada** pero **funcionalmente idéntica**, como se muestra en esta figura:



Antes de la difusión en masa de los circuitos de alta densidad de integración, como por ejemplo los **PLD**, y los correspondientes programas capaces de **sintetizar automáticamente** partiendo de **código escrito**, no era raro encontrar a los diseñadores con **papel** y **lápiz** a dibujando **mapas de Karnough**, escribir **largas funciones booleanas** y **descomponerlas** en elementos simples **NOT**, **AND** y **OR** para realizar el circuito.

Existen bastantes leyes y métodos, por ejemplo las **leyes de Morgan** o el **método de Quine Mc Cluskey**, que **simplifican** y **estandarizan** las **funciones booleanas** con procedimientos sistemáticos que han adoptado los **programas** para conseguir óptimos resultados en poco tiempo.

... Vamos a **pasar a la práctica**. En el siguiente apartado aprenderemos las bases para "movernos" dentro de un **esquema**, en particular dentro del entorno de desarrollo **Quartus II**.

También expondremos **pequeños ejercicios** para que cada uno los resuelva. Las soluciones las presentaremos en los próximos números.

Es bastante aconsejable leer el artículo precedente presentado en la **revista Nº269** y tenerlo a mano. En este artículo se presentó el **Programador para CPLD LX.1685** y el programa **Quartus II**.

Análisis del circuito de Prueba: ESQUEMA

Para tomar práctica con la programación de nuestro dispositivo y el programa **Quartus II de Altera** y así poder empezar a ser **autónomos** proponemos practicar con un ejemplo.

En primer lugar hay que **abrir el programa** y **cargar el proyecto** de **ejemplo** contenido en el CD - **CDR1685** incluido en el KIT **LX.1685**.

Para trabajar más cómodamente es conveniente hacerlo desde el **disco duro**, para lo que hay que **copiar** el directorio "**Self_Test**" del **CD-ROM** al disco duro, por ejemplo en un directorio llamado **C:\CPLD\ Self_Test**.

Después de **abrir Quartus II** hay que ir al menú **File** y hacer click en **Open Project...** (ver Fig.3). A continuación hay que abrir el archivo del proyecto "**test.qpf**" recién copiado (ver Fig.4).

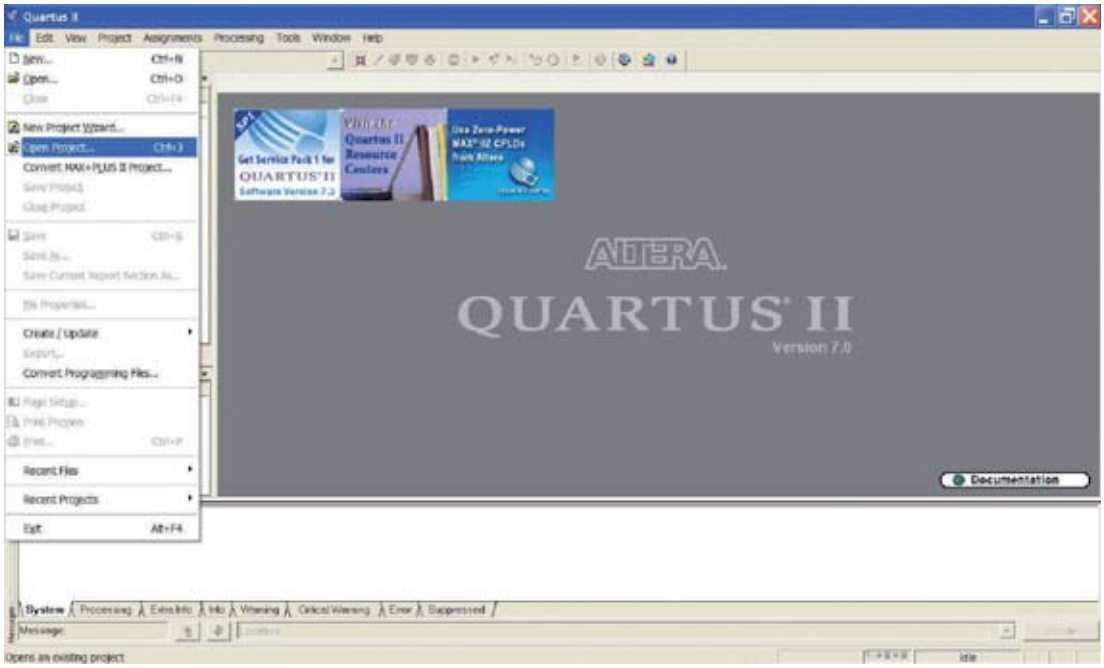


Fig.3 Una vez copiada la carpeta "SelfTest" al disco duro del PC hay que abrir el programa QUARTUS II y hacer click en línea "Open Project" del menú "File".

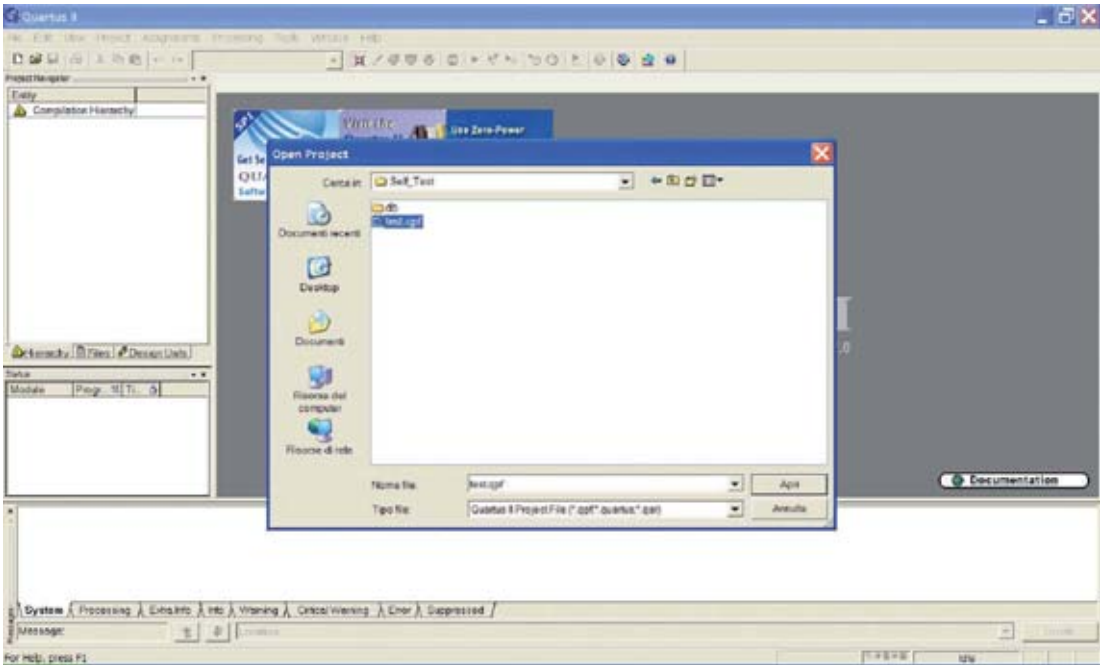


Fig.4 Cuando aparezca esta pantalla para continuar hay que hacer click sobre la línea que contiene el nombre del archivo "test.qpf."

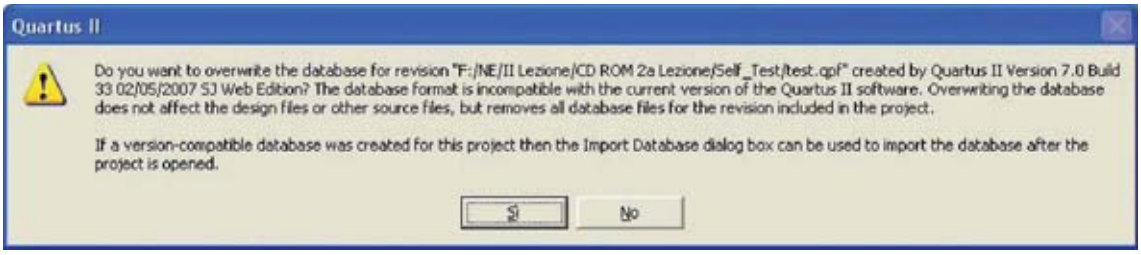


Fig.5 Si aparece esta indicación significa que vuestro PC no dispone de una versión de QUARTUS II actualizada. Se puede utilizar nuestro CD-ROM o Internet para actualizarla.

Fig.6 Si todo funciona correctamente después de la pantalla mostrada en la Fig.4 el programa mostrará este aspecto. Hay que hacer click en el proyecto "test".

Si por cualquier motivo tuvierais una **versión** de Quartus II **diferente** para la que ha sido creado el proyecto de ejemplo aparecería el **mensaje** mostrado en la **Fig.5**. Sólo hay que hacer click en **SÍ**.

Una vez abierto el proyecto hay que hacer click en el **icono** "**test**" situado en la ventana de la izquierda (ver Fig.6). Llegado este punto el **esquema** que ha generado el **archivo test.pof** está **disponible** (ver Fig.7).

En un principio puede parecer complicado pero con un poco de práctica resultará **bastante sencilla** la **programación del hardware**.

Tenemos que controlar cargas como **diodos LED** y **zumbadores** (**buzzers**), además de los **pulsadores** de entrada, que se **activan** a **nivel bajo**.

Para facilitar los razonamientos hemos **negado** tanto las **entradas** como las **salidas** a través de puertas **NOT** y así poder trabajar con la **lógica normal**.

En todo caso, para evitar confusiones, hablaremos de **señal activa** o **no activa**, prescindiendo de la lógica utilizada.

Empezamos el análisis por la parte más simple, los **pulsadores** "**p4**" y "**p5**" que activan la **salida** "**dl1**" mediante una puerta lógica **AND** cuando son accionados simultáneamente.

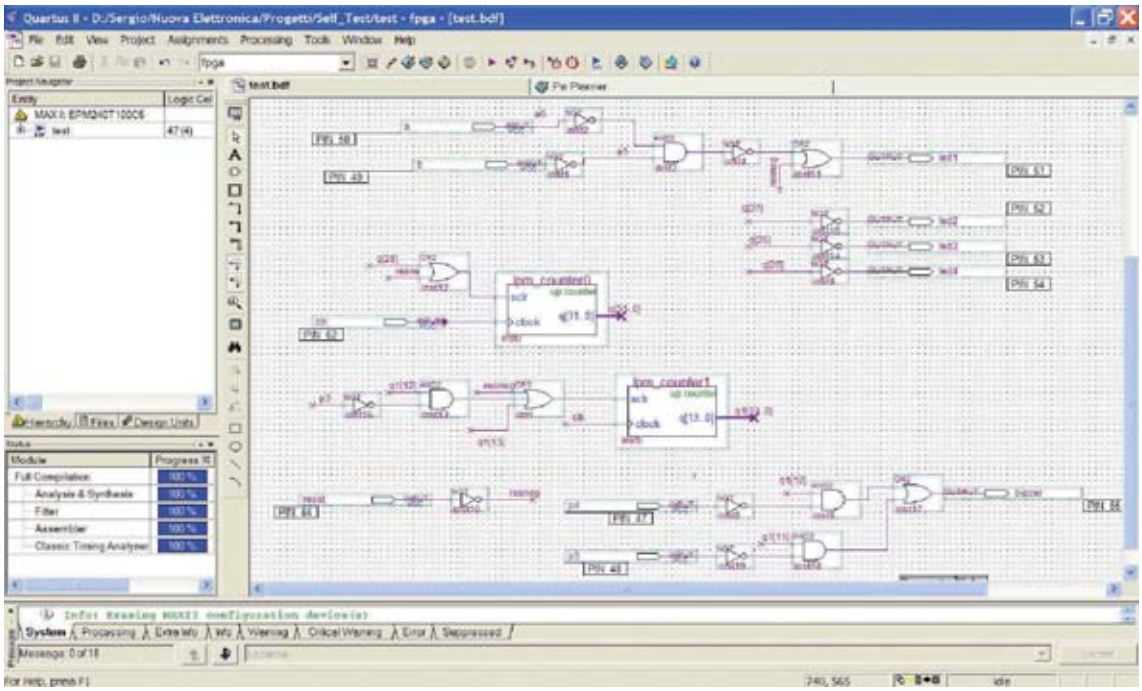


Fig.7 Aquí está reproducido el esquema correspondiente al archivo "test.pof", proporcionado con el Programador CPLD LX.1685 publicado en la revista Nº269.

La puerta **OR** situada en medio sirve para **impedir** el encendido del **diodo LED** cuando se acciona el pulsador **RESET**.

Quienes no recuerden la **tabla de la verdad** de estas **funciones lógicas** pueden consultar el artículo la **revista Nº269**.

Continuando con el análisis, en primer lugar hay que decir que **no todo** el circuito es **combinacional**, ya que hay **registros**, elementos de **memoria** (**lpm_counter0** de 32 bits y **lpm_counter1** de 14 bits) y **divisores de frecuencia**, utilizados para permitir trabajar con **frecuencias adecuadas** en **elementos luminosos** (diodos LED y displays) y **sonoros** (zumbadores) ya que normalmente la **frecuencia del cuarzo** suele ser **mucho más alta** (en nuestro caso **20 MHz**).

Así nuestro **contador binario** con **diodos LED** se incrementa con un **tiempo** igual a:

$$\text{Tiempo (segundos)} = \frac{2 \text{ ^ bits\_contador}}{\text{Frecuencia reloj (Hz)}}$$

NOTA: Para quienes tengan poca familiaridad con las matemáticas precisamos que el símbolo **^** significa que el número **2** debe ser **elevado** al **bits_contador**.

que corresponde a una **frecuencia** de:

$$\text{Frecuencia (Hz)} = \frac{1}{\text{Tiempo (segundos)}}$$

Considerando que el **led4**, que representa el **bit menos significativo** del contador binario implementado, es activado mediante la **señal q[25]** (o bien el **26° bit** del primer contador), tendrá un **período** de:

$$T = \frac{2\text{^}26}{20.000.000}$$

Elevando el número **2** a la **26** obtenemos un valor de **67.108.864**.

Sustituyendo:

$$\text{Tiempo (segundos)} = \frac{67.108.864}{20.000.000} = 3,35$$

Que corresponde a una **frecuencia** de:

$$\text{Hz} = \frac{1}{3,35} = 0,2985$$

Valor que podemos **redondear** a **0,3 Hertzios**.

Por el mismo motivo el **led3** y el **led2** se encenderán, respectivamente, con un **período doble** y **cuádruple** con respecto al **led4**, generando así el efecto de **contador binario**.

De forma similar se procede para la generación de la **frecuencia** del **zumbador**. Analizando el circuito que lo controla se puede aprender mucho.

Como se puede probar experimentalmente al accionar **un pulsador** se emite una **nota**, si se acciona el otro pulsador se emite la misma nota pero **una octava superior**, esto es del **doble** de **frecuencia**.

En efecto, si se observa en el esquema la **salida "buzzer"** está controlada por una **OR** que tiene como entradas las salidas de **dos AND**: El **zumbador** (**buzzer**) se activa gracias a la **OR** cuando **al menos** la salida de **una** de las **dos AND** está **activa**.

Las **AND** no hacen otra cosa que hacer **pasar** la señal **q1[11]** (frecuencia más **alta**) o a la señal **q1[12]** (frecuencia más **baja**) según se accionen "**p3**" o "**p4**".

Se trata de un sencillo **multiplexor simplificado**.

Ejercicio 1

En base en la fórmula utilizada para el **cálculo** del **período** y para la **frecuencia** que hemos expuesto anteriormente, y analizando el esquema, intentar **calcular** el valor de las **dos frecuencias** correspondientes a la utilización de cada uno de los **pulsadores**.

Modificación del ESQUEMA

Una cuestión muy importante a tener en cuenta es que, cuando se escribe **código** o se diseñan **esquemas** para un **PLD**, hay que estar **muy atentos** a las **conexiones físicas** del circuito.

Por ejemplo, definir una señal de **salida** dentro del esquema que en el circuito esté conectada a una **entrada** puede llegar a **destruir** el **terminal** del **PLD**.

También se pueden producir **cortocircuitos** y **destruirse** el **terminal** si, por ejemplo, el **aparato programable** intenta forzar una de sus **salidas** a **nivel alto** y **externamente** en el circuito impreso la conectamos directamente a **masa**.

Estos ejemplos evidencian que hay que estar **muy atentos** para evitar condiciones similares y utilizar mucho las **simulaciones**.

Una vez realizada esta importantísima aclaración ahora podemos **modificar el esquema**. Tanto sobre los **símbolos** como sobre los **cables de conexión** se pueden realizar todas las acciones típicas de entornos Windows: **Editar, Mover**, **cortar**, **pegar**, **borrar**, etc.

Por ejemplo se puede probar a **cambiar** una de las **frecuencias** de trabajo del **zumbador**.

Para ello en primer lugar hacemos click sobre la señal **q1[12]** y con el botón derecho seleccionamos "**Properties**" (ver Fig.9).

Ahora modificamos **q1[12]** por **q1[10]** y seleccionamos **OK**.

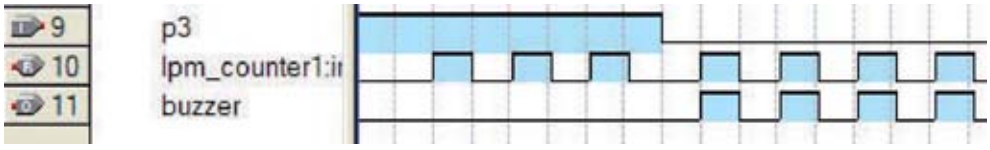


Fig.8 En este gráfico se muestra la salida del contador "q1[11]". Conmuta cíclicamente de nivel lógico 1 a nivel lógico 0 en función de la frecuencia de cuenta. La señal se lleva al zumbador (buzzer) cuando se acciona el pulsador "p3".

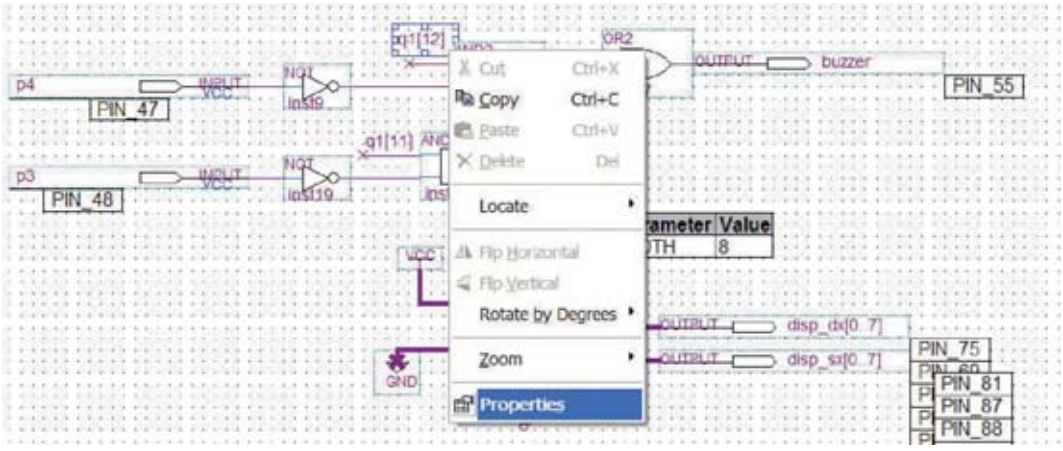


Fig.9 Haciendo click sobre la señal "q1[12]" aparecerá esta ventana. Aquí se ha de hacer click con el botón derecho del ratón sobre "Properties".

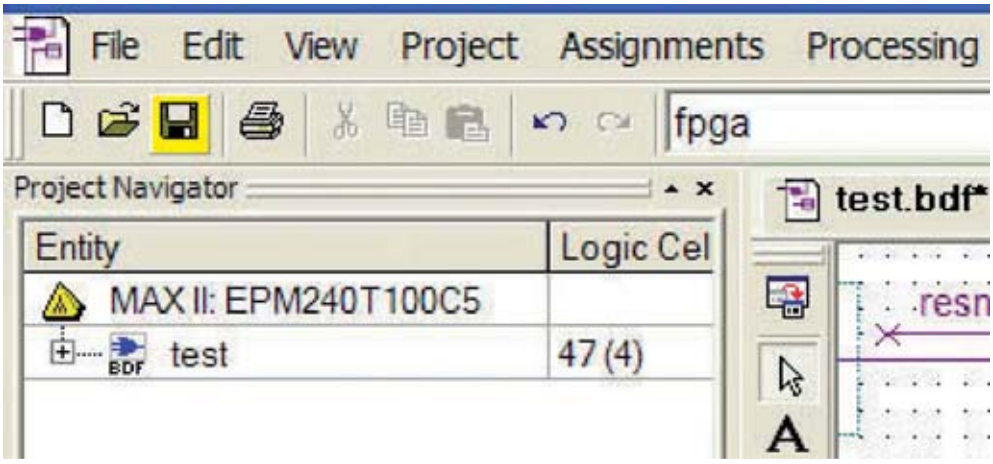


Fig.10 Para salvar el esquema basta con hacer click sobre el icono "Save" (disquete sombreado), situado entre el icono "Open" y el icono "Print".

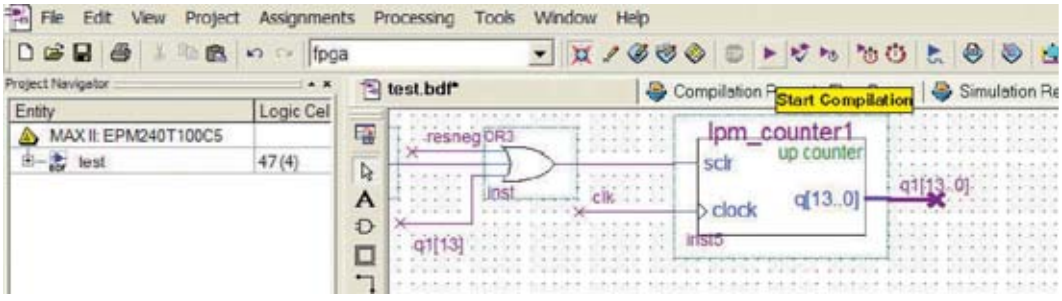


Fig.11 Una vez salvado el esquema se puede realizar la compilación haciendo click sobre el icono "Start Compilation".

A continuación **salvamos** el esquema utilizando el **icono** "**Save**" (ver Fig.10) y lanzamos la **compilación** utilizando el **icono** "**Start Compilation**" (ver Fig.11).

Después de **confirmar** la acción ya podemos **programar** nuestro **CPLD** con el procedimiento descrito detalladamente en la **revista Nº269**.

Desde este momento la **frecuencia** asociada a "**p4**" es **más alta**. Además accionando al **mismo tiempo** los **dos pulsadores** se percibirá como se **suman** ambas **frecuencias**.

INSERCIÓN de NUEVOS SÍMBOLOS

Ahora podemos probar a insertar un **nuevo símbolo**. Para realizar esta acción hay que hacer click con el **botón derecho** del ratón en un **espacio vacío** del **esquema**.

En el cuadro emergente que aparece hay que seleccionar "**Insert**" y luego "**Symbol...**" (ver Fig.12). Se abrirá una ventana con las **librerías disponibles**.

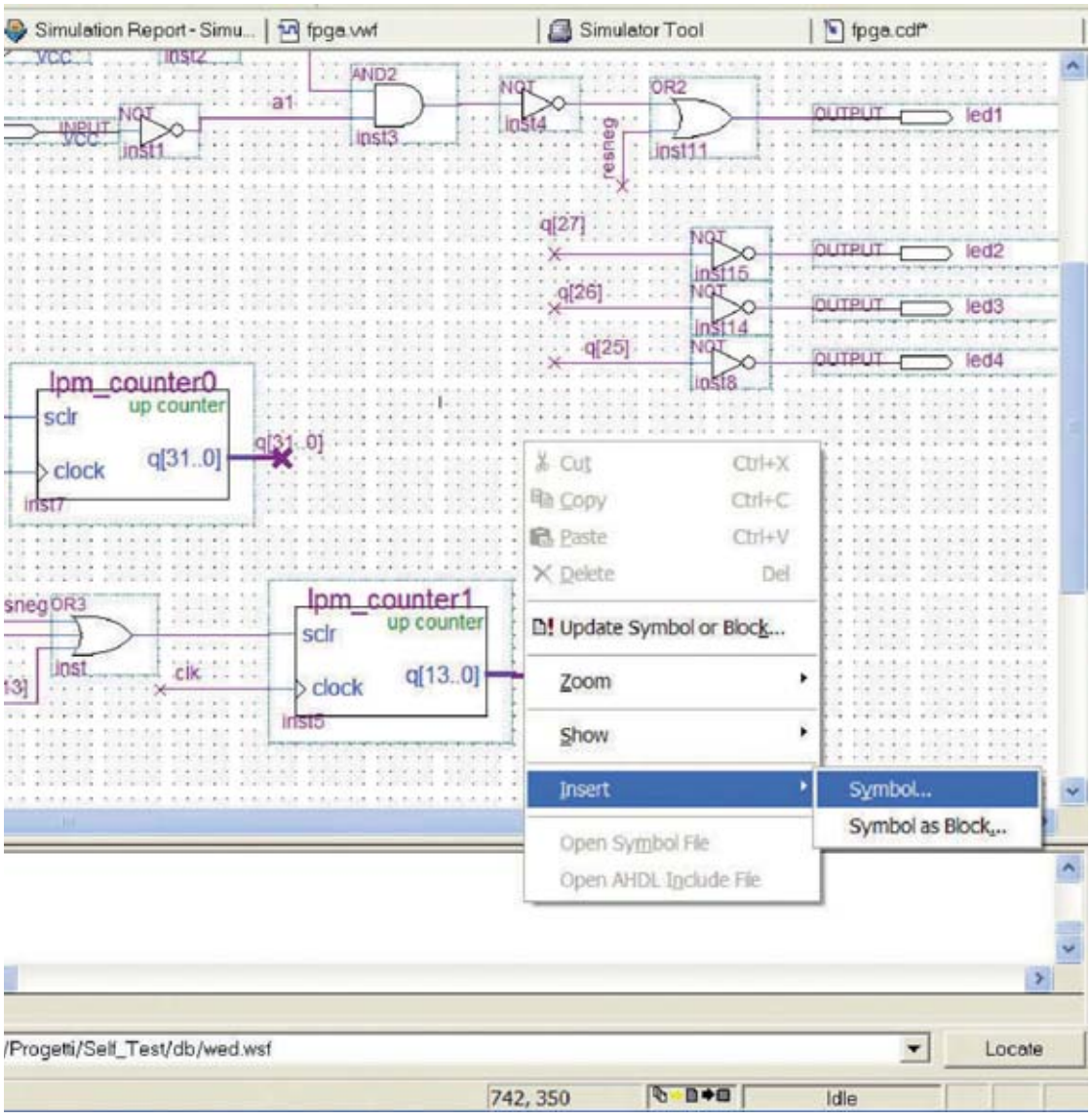


Fig.12 Si se quiere insertar un nuevo símbolo en el esquema hay que hacer click con el botón derecho del ratón sobre un espacio vacío del área de edición, seleccionar "Insert" y luego "Symbol".

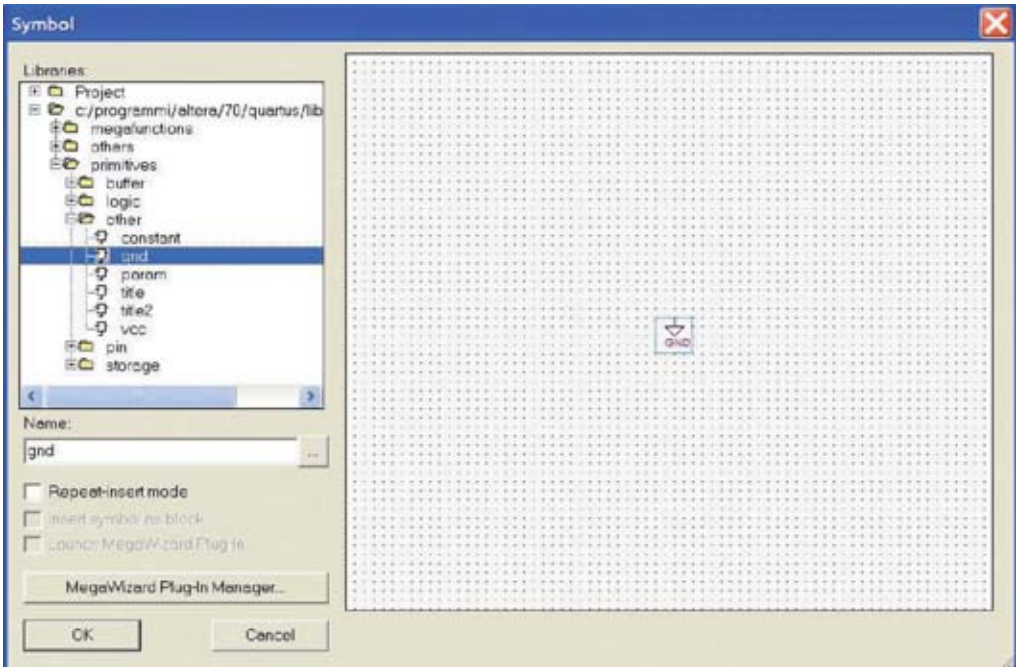


Fig.13 Para insertar un "gnd", es decir un nivel lógico 0, hay que hacer click sobre "others" (rama "primitives"), luego sobre "gnd" y por último en OK. Después ya solo queda posicionar el símbolo.

En la parte izquierda de la pantalla hay que seleccionar la **rama** adecuada dentro del **directorio de instalación** (por ejemplo "**c:\programas\altera\70\quartus\libraries**"), seleccionando "**primitives**".

En estas carpetas están presentes los **símbolos** como **bloques base**.

Por ejemplo, podemos insertar un "**gnd**" (**nivel lógico 0**) haciendo click sobre "**others**" y luego sobre "**gnd**" (ver Fig.13). En la parte derecha se **previsualiza** el **símbolo**.

Ya sólo queda pulsar en **OK** y **posicionar el símbolo** en el lugar deseado del **esquema**.

Siguiendo este procedimiento se puede insertar cualquier otro bloque presente en las librerías.

Si ahora acercamos el **puntero** a la **terminación** del símbolo "**gnd**" recién insertado se notará como el puntero cambia de forma. Se trata de la indicación de **interconexión**.

Haciendo click y arrastrando el ratón se creará un **cable de conexión** (ver Fig.14).

Una vez asignada una **etiqueta** al cable de conexión se pueden **realizar conexiones** utilizando el **nombre** de la **etiqueta asignada**.

Por ejemplo asignando el nombre "**vlow**" al cable conectado al "**gnd**" recién insertado **todas las señales** con la denominación "**vlow**" quedarán automáticamente **conectadas a masa** (**valor lógico 0**).

Ejercicio 2

Modificar el esquema de forma que los **diodos LED** conectados a las señales **q[25]**, **q[26]** y **q[27]** permanezcan **siempre apagados**. Probar la solución.

Ejercicio 3

Insertar el símbolo "vcc" y asignarle un nombre, por ejemplo "vhigh". Modificar el esquema de forma que los **diodos LED** conectados a las señales **q[25]**, **q[26]** y **q[27]** permanezcan **siempre encendidos**. Probar la solución.

Para insertar **bloques lógicos** hay que proceder de forma similar a las acciones realizadas para insertar "**gnd**" y "**vcc**" (ver Figs.12-13), buscando en "**logic**" en lugar de en "**primitives**".

Aquí se encuentran las **funciones lógicas básicas** como **AND**, **OR**, **NAND**, **NOR**, etc. de **2** o **más entradas**.

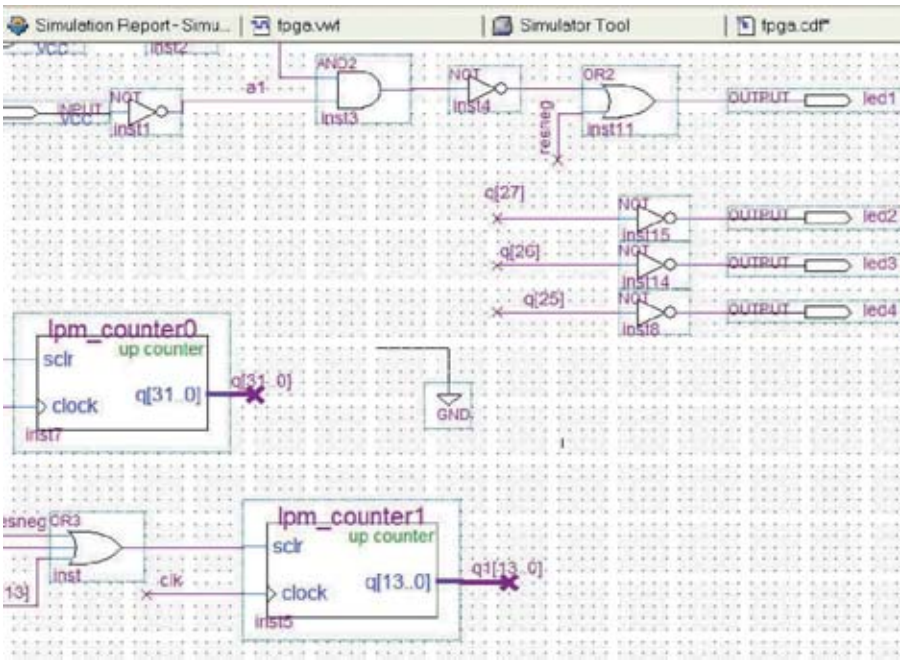


Fig.14 Se puede trazar una línea equivalente a un cable de conexión manteniendo pulsado el botón izquierdo del ratón, como se puede ver en los contadores q(31) y q(13).

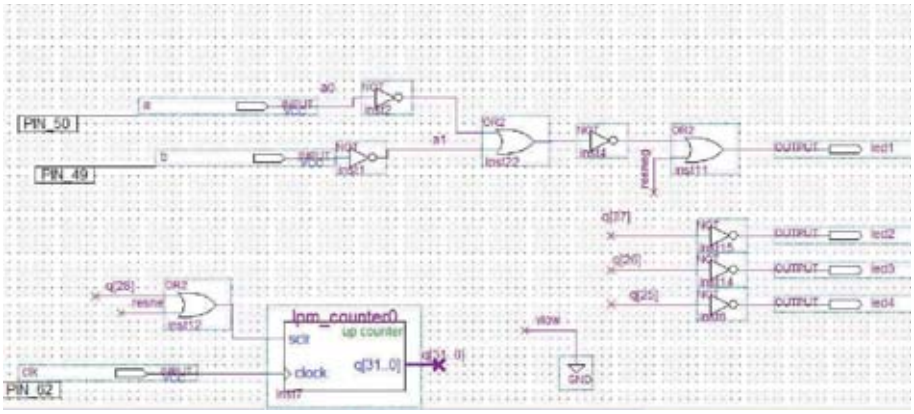


Fig.15 Denominando al extremo de un cable "vlow" se le asignará un nivel lógico 0. Así todas las señales "vlow" automáticamente quedarán conectadas a masa.

Se puede intentar, por ejemplo, **sustituir** la **AND** presente en las entradas **"p4**" y "**p5**" por **otra función** de **dos entradas** y experimentar de forma práctica las consecuencias que se producen en la lógica de encendido de los diodos LED actuando sobre los pulsadores.

Ejercicio 4

Borrar la **AND2** (**inst3**) asociada a los pulsadores "**p4**" y "**p5**" y **sustituirla** por una **OR** de **dos entradas** (**OR2**). Probar los resultados.

NOTA: Para **actualizar** el programa **Quartus II de Altera** se puede descargar de:

https://www.altera.com/support/software/download/sof-download_center.html

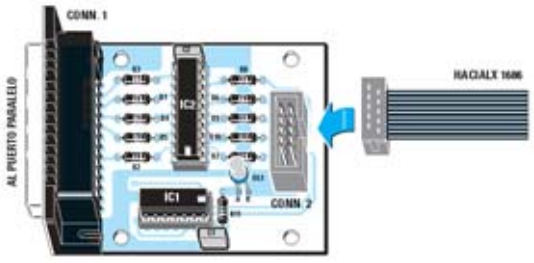


Fig.16 Esquema práctico de montaje de la tarjeta del Programador LX.1685. Esta tarjeta se conecta a través de una manguera de 10 cables a la tarjeta LX.1686.

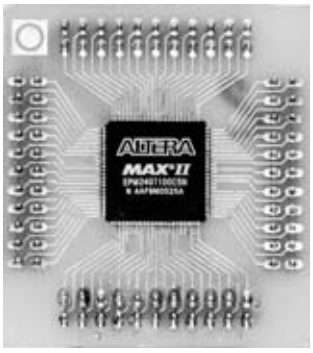


Fig.17 Aquí se muestra el minúsculo integrado SMD de 100 terminales que proporcionamos soldado sobre el circuito impreso KM.1686.

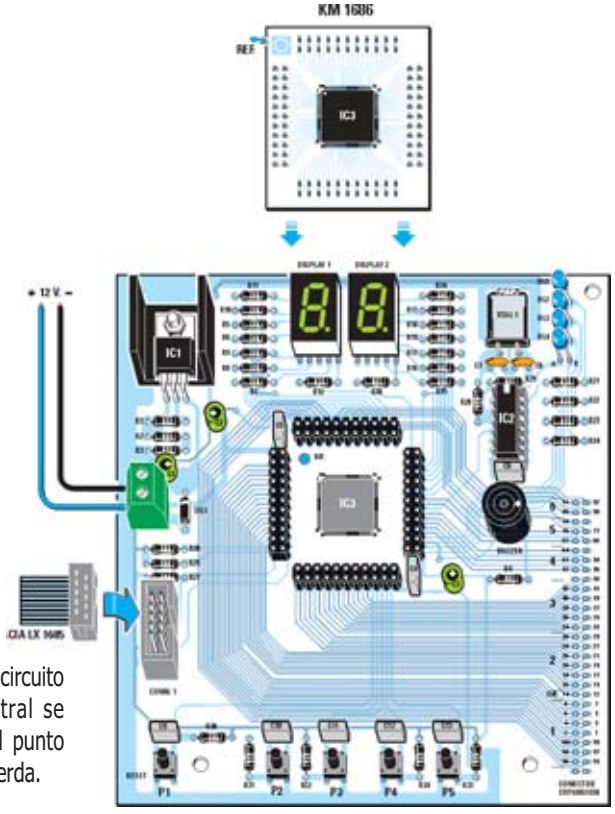


Fig.18 Esquema práctico de montaje del circuito impreso LX.1686. En el conector central se acopla la tarjeta KM.1686, orientando el punto de referencia hacia la parte superior-izquierda.

PRECIO de REALIZACIÓN

LX.1685: Todos los componentes necesarios para realizar el **Programador CPLD** (ver Fig.16), **incluyendo** circuito impreso, integrados, conector para el puerto paralelo, manguera de conexión de 10 hilos para conectar el programador a la Tarjeta de prueba LX.1686 y el **CD-ROM CDR1685** .....

NOTA: El **CD-ROM CDR1685** contiene el programa **Quartus II**, es decir el paquete completo para la escritura del código de programación, para ensamblar y para programar los dispositivos **CPLD**. Además contiene dos programaciones de prueba (**counter.pof** y **test.pof**)

LX.1686: Todos los componentes necesarios para realizar la **Tarjeta de prueba CPLD** (ver Fig.17) **incluyendo** circuito impreso, integrados, cuarzo, dos displays, diodos LED, zumbador, pulsadores y la tarjeta SMD **KM.1686** (ver Fig.15) con el chip CPLD **MAX II EPM240T100C5N** .....**86,30€**

CS.1685: Circuito impreso .....**5,20€**

CS.1686: Circuito impreso .....**24,00€**

ESTOS PRECIOS NO INCLUYEN I.V.A.