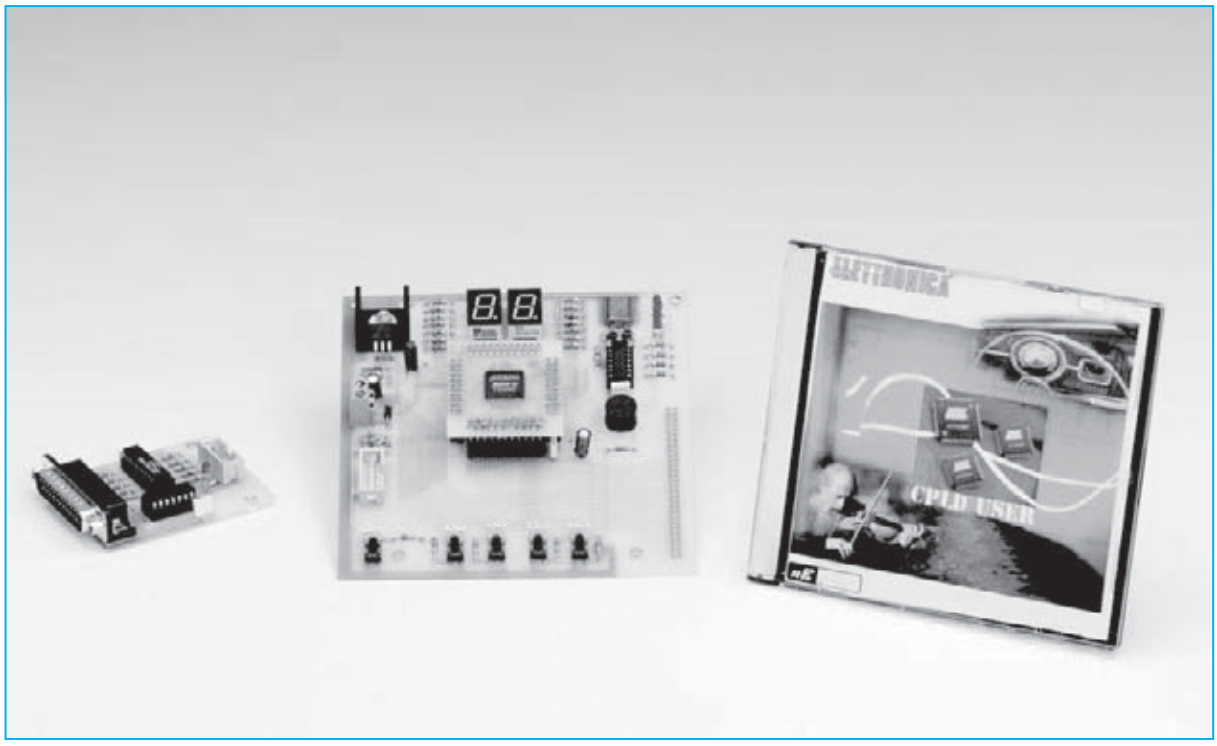


Aún mas APLICACIONES

Vayamos todavía un paso mas adelante con la descripción de las capacidades del software Quartus II, sabiendo que podemos crear un componente personalizado, con los consejos del esquema realizado en número anteriores.

Una de las peculiaridades de este hardware es su **modularidad**, ya que nos permite crear un componente con unas características determinadas, pudiéndolo reutilizar cada vez que queramos.

LA CARRERA CONTRA EL TIEMPO

Como habréis podido observar, el mercado tecnológico ha tenido un gran desarrollo en los últimos años, creándose nuevos productos que suplen a los antiguos en muy poco tiempo.

Si por un lado, el cliente siempre encuentra una oferta más variada y más cercana a sus pretensiones, por otro, hay un mayor esfuerzo por parte de los diseñadores, que deben ser capaces de crear un proyecto, detrás de otro, en un tiempo récord (el llamado **Time-to.market**).

Vosotros, que con este hobby, os encontráis a mitad de camino entre consumidores y diseñadores, sabéis que es imposible poder proyectar un producto, detrás del otro, en tiempos tan cortos, comenzando de cero.

De hecho, hay técnicas que ayudan al proyectista, y que permiten invertir en diseños ya desarrollados para la realización de nuevos productos. De este modo, se puede iniciar de una

base solida y probada que nos permitan desarrollar estos nuevos productos.

Por ejemplo, es muy utilizado el denominado **IP-Reuse** (Intellectual Property= **Propiedad Intelectual**: información restringida y utilizada solo bajo autorización del programador), con la cual se puede reutilizar los **códigos** o los **bloques** creados anteriormente,pudiéndose modificar sus parámetros para los nuevos productos.

Para ello, os proporcionaremos las indicaciones útiles para crear vuestro componente y utilizarlo cada vez que lo necesitéis.

CREACION DE COMPONENTES PERSONALIZADOS

Intentemos construir un componente partiendo del archivo **esquemático** que hemos creado para el decodificador con **output enable**.

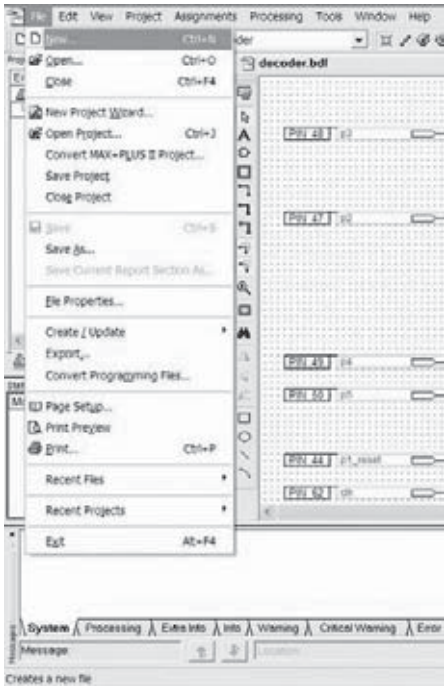
Nota: *os recordamos que este mismo componente se ha utilizado anteriormente en el ejercicio Nº1.*

Primeramente, preparamos el ambiente de trabajo: hay dos opciones:

copiad la carpeta **/Progetti/Decoder_modificato** que encontraréis en el DVD en el directorio de trabajo (por ejemplo la que habéis utilizado otras veces);
también podéis utilizar vuestro proyecto "**decoder**" si la habéis modificado correctamente con la función **output_enable**.

Entrad en el **Quartus II** y abrid el proyecto "**decoder**" que se encuentra, a vuestra elección, o en el directorio que habéis copiado del **DVD**, o en vuestro directorio.
Ahora, cread un nuevo archivo: haced click en el menú "**file>new...**" (ver fig.1), entonces en la ventana que se abra seleccionad "**Block Dia-**

con el QUARTUS II



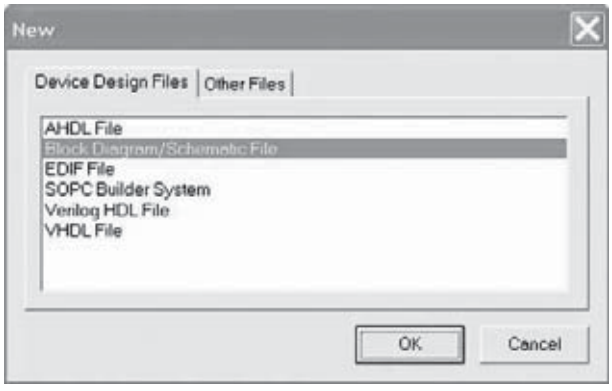


Fig.1 como se puede ver en la imagen reproducida a la izquierda, para crear un archivo, en el menú haced click sobre el archivo, y en la ventana que se abra sobre la opción New.

Fig.2 por tanto, seleccionad la opción "Block Diagram/Schematic File" como se observar en la figura superior, y pulsad en Ok.

gram/Schematic File", luego clickad en "**Ok**", como es visible en fig.2.

En este punto se abrirá un nuevo archivo que os servirá para meter el componente "decoder". Guardadlo, y después, pulsad en el menú "**File>Guardar como...**" (ver fig.3).

En la ventana que se abra debéis escribir el nombre "**decoder_component**" al nuevo archivo, visible en la fig.4, y pulsad "Guardar".

Observad que en la misma ventana la opción aparece default "**Add file to current project**" con ella, el **Quartus II** comprobará que el archivo creado pertenece al proyecto, siendo utilizado durante agrupamiento.

A continuación, regresad al **Top file** "**decoder.bdt**" (Top=dibujo principal), seleccionad el núcleo del decoder (aquello que tramita la función del **decodificador**, es decir, aquella junto al rectángulo verde con la etiqueta "**Decider 2 bit con output enable**"), y copiadlo, pulsando sobre **copiar**, come se puede apreciar en la fig.5.

Después de esto, id nuevamente al archivo "**decoder_com-ponent.bd**f" y "pegadlo" sobre el mismo que acabáis de copiar: podéis comprobar el resultado en la fig.6.

Ahora, debéis dar un nombre a los **input** y a los **output** de vuestro componente, para que de esta manera sea **autónomo** y **compatible**.

Por ejemplo, a los **input: in_a, in_b**; y a los **output**: **out_1, out_2, out_3**.

El resultado obtenido será similar al que hemos reproducido en la fig.7; el archivo estará listo y lo podréis guardar en el menú "**Archivo>Guardar**".

Ahora, deberéis crear un símbolo para este componente: pulsando en el menú "**File>Create/Update>Create Symbol Files for Current File**" (ver fig.8), se abrirá la ventana de la fig.9, indicando que el símbolo ha sido creado correctamente.

Entonces, pulsad "**Ok**".

Desde este momento, tendréis vuestro componente personal "**decoder_component**" en los archivos de proyecto, en la carpeta "**Project**".

A continuación, volved a **al Top file** "**decoder.bdf**", y borrad la parte con la que habéis creado el componente decodificador: el resultado puede verse en la fig.10.

Finalmente, introducid el componente en el lugar de la lógica que habéis borrado, pulsando el botón derecho del ratón sobre el archivo menú "**insert>component**".

En la ventana que se abre, haced click en "**project**" (la librería del proyecto), y por tanto, sobre vuestro "**decoder_component**", como se puede ver en la fig.11.

El resultado debería ser parecido al reproducido en la fig.12.

Observad que el componente tiene exactamente los mismo **input** y los mismo **output**, que vosotros habéis definido, al interior del archivo, del que es símbolo.

Solo queda **interconectar** las **entradas** y las **salidas** del componente a los **pin** físicos de la **PDL**, a través de los **pin** definidos en el **Top**.

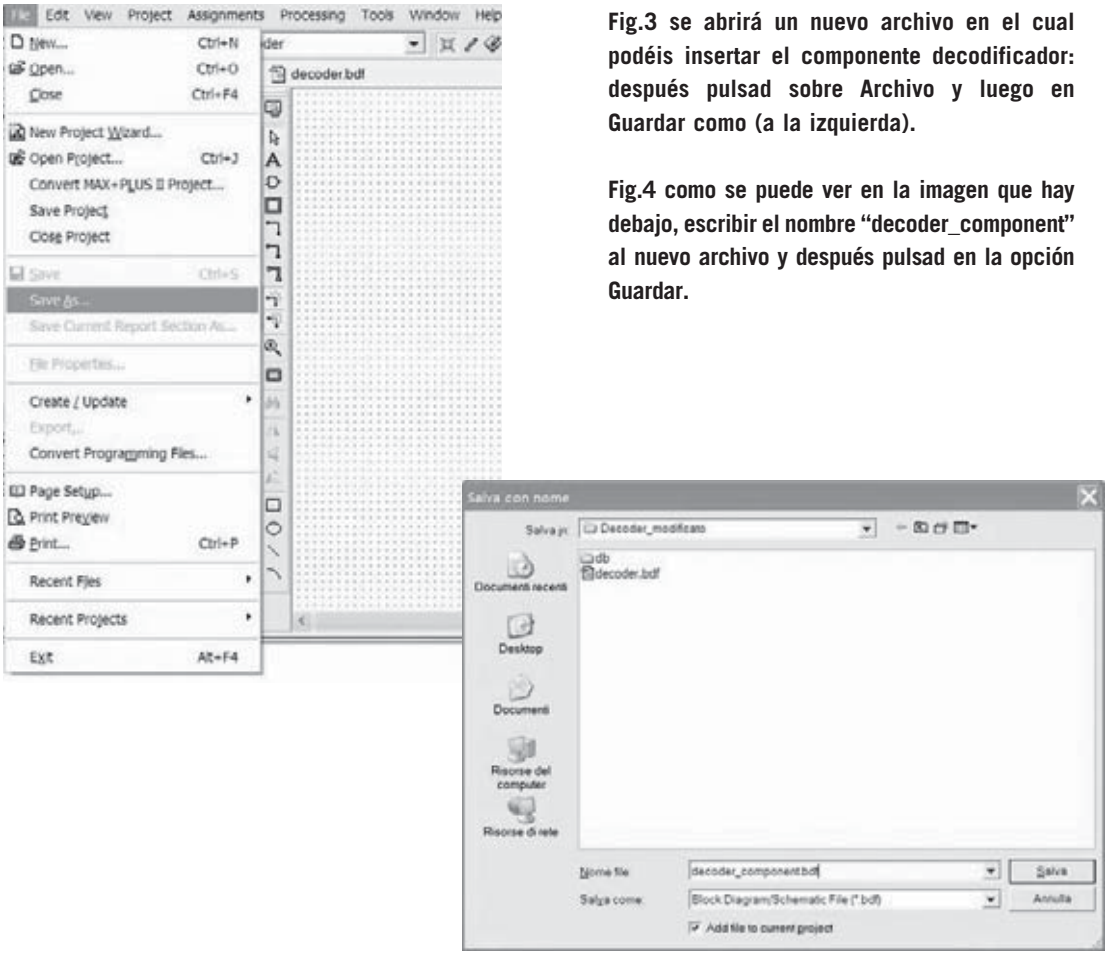
Podéis observarlo en el fig.13.

Entonces, abrir la agrupación y, cargando el nuevo archivo de programación, comprobad que el comportamiento del circuito sea igual que antes.

También, comprobad que el consumo de los recursos y en particular de **Logic Cells** (**LC**), es el mismo, o sea de **4** sobre **240** totales (**2%**), como pueden verse en los **informes** que aparecen al final de la recopilación y en la fig.14.

Lo único diferente es el número que aparece en el recuadro "**Project Navigator**" arriba a la izquierda (ver fig.15): si recordáis al inicio era "**4 (4)**" mientras que ahora es "**4 (0)**".

El primero de sus componentes nos indica cuantos **LC** utiliza este componente, junto a sus subcomponentes, mientras que el que está entreparéntesis nos indica cuantos utiliza exclusivamente (sin contar sus subcomponentes).



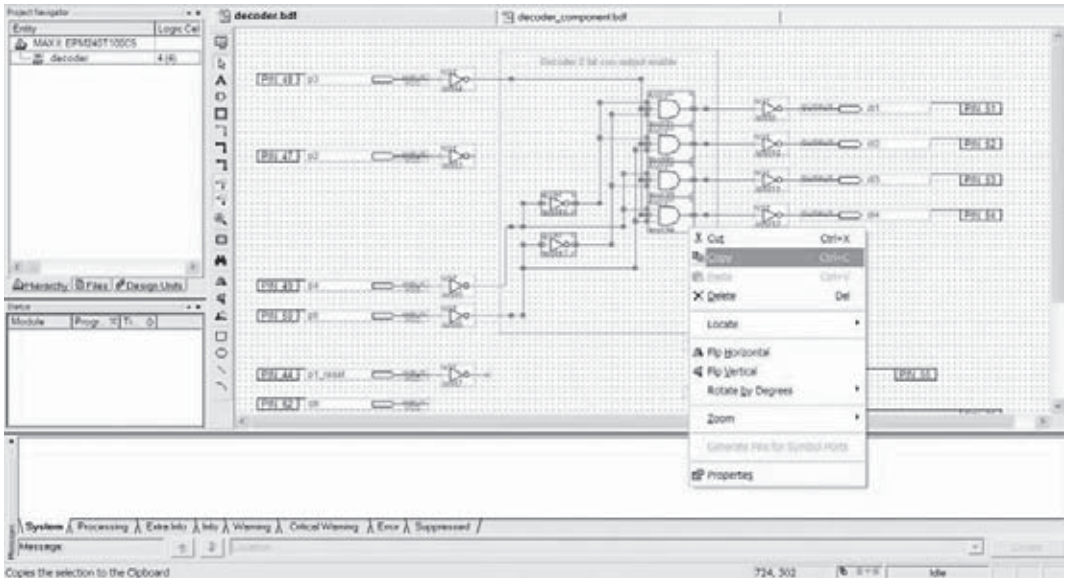


Fig.3 se abrirá un nuevo archivo en el cual podéis insertar el componente decodificador: después pulsad sobre Archivo y luego en Guardar como (a la izquierda).

Fig.4 como se puede ver en la imagen que hay debajo, escribir el nombre "decoder_component" al nuevo archivo y después pulsad en la opción Guardar.

Fig.5 aquí podemos ver la solución del ejercicio N.1, propuesto en revistas anteriores.

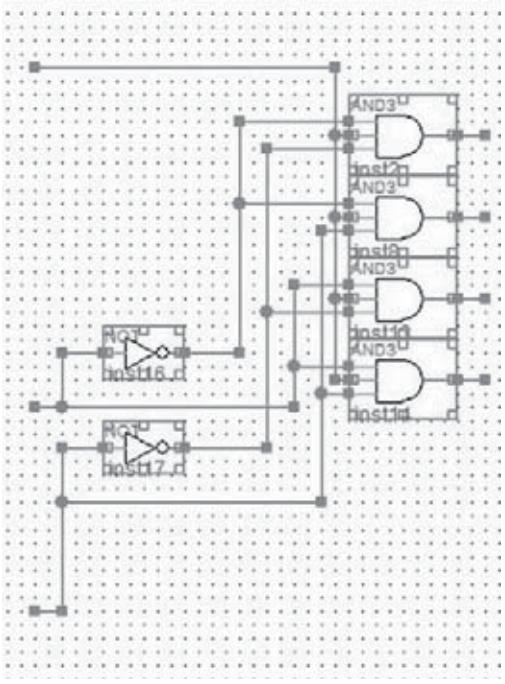


Fig.6 pegad en el nuevo archivo "decoder_component.bdf" que hemos copiado, observando el resultado que hemos reproducido aquí.

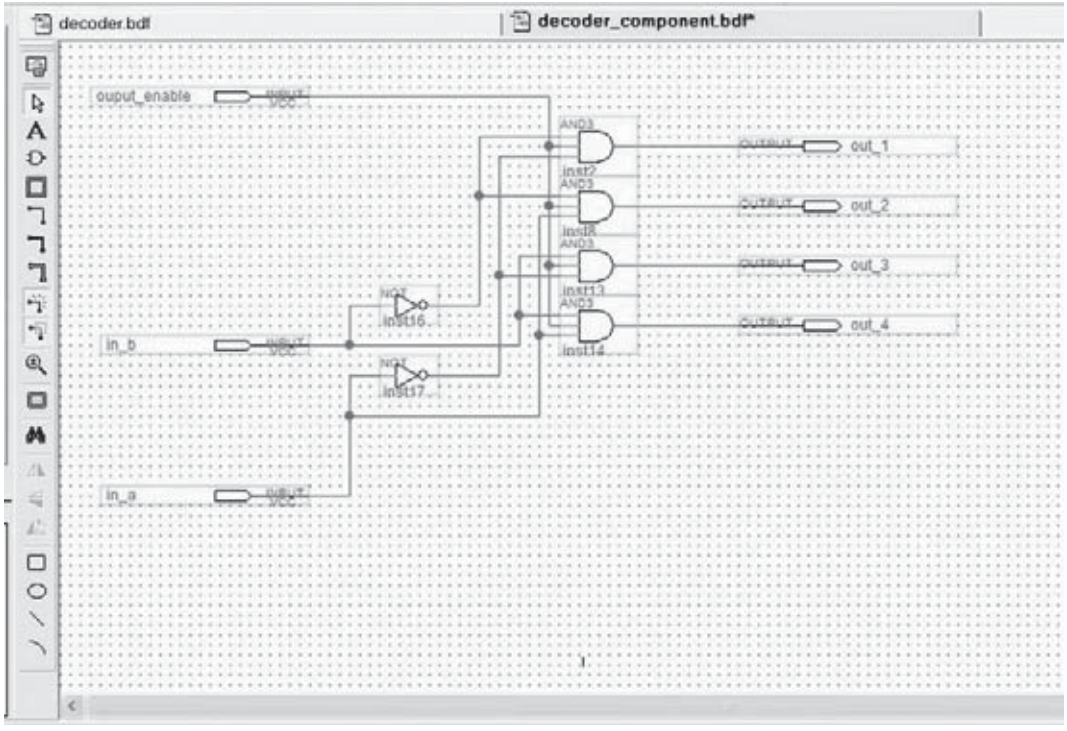
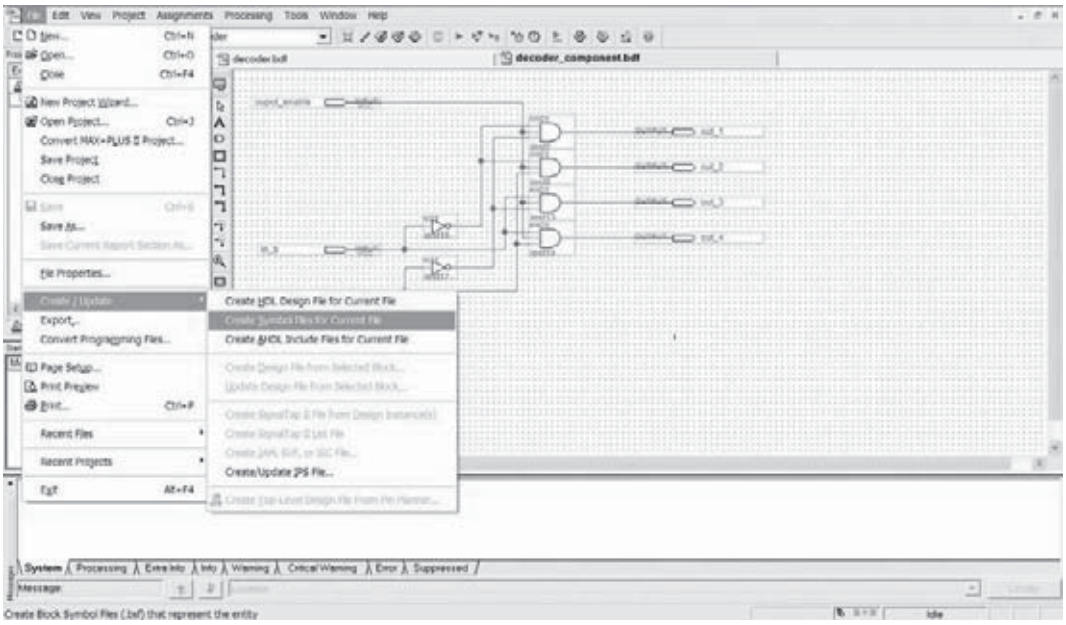


Fig.7 si damos un nombre a los input y a los output del componente, siguiendo las instrucciones dadas en la revista anterior, conseguiréis este mismo resultado, que podréis guardar presionando sobre "Archivo>Guardar".



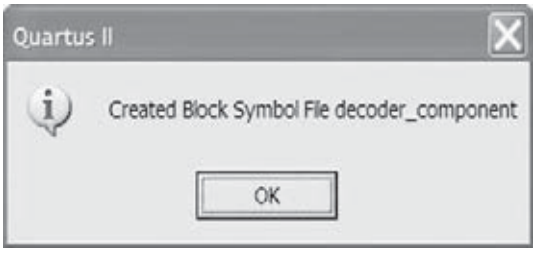


Fig.8 llegados a este punto, podéis crear un símbolo que se asocie a vuestro componente, haciendo click sobre el menú "Archivo>Create/Update> Symbol Files for Current File".

Fig.9 se os abrirá un ventana similar a esta, indicando que el símbolo ha sido creado correctamente. Después pulsad en Ok.

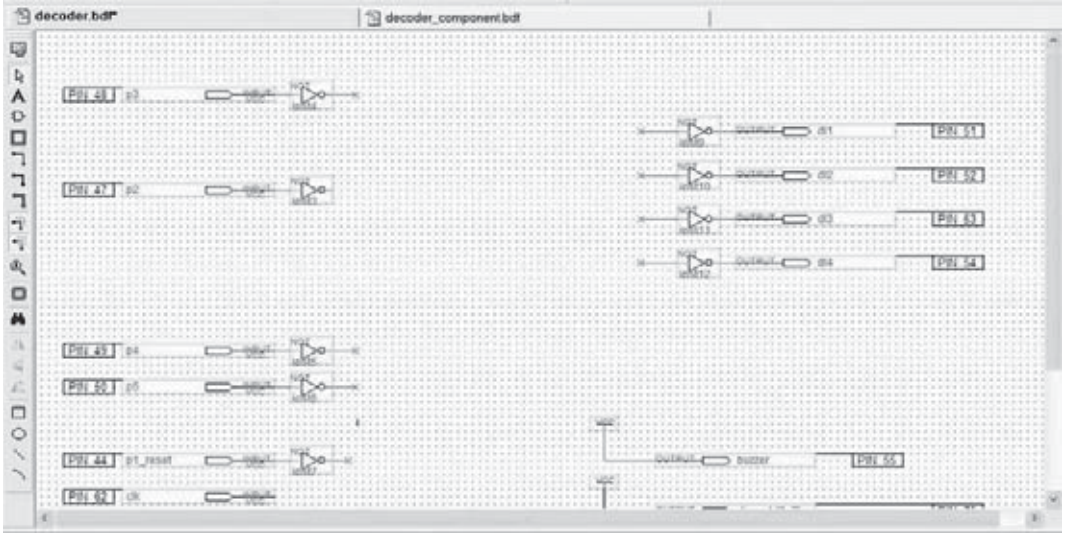


Fig.10 ahora que ya tenéis a vuestra disposición en las librerías de este proyecto vuestro componente personal "decoder-component", regresad al "Top file decoder.bdf", y borrad la parte con la que habéis creado el componente decodificador.

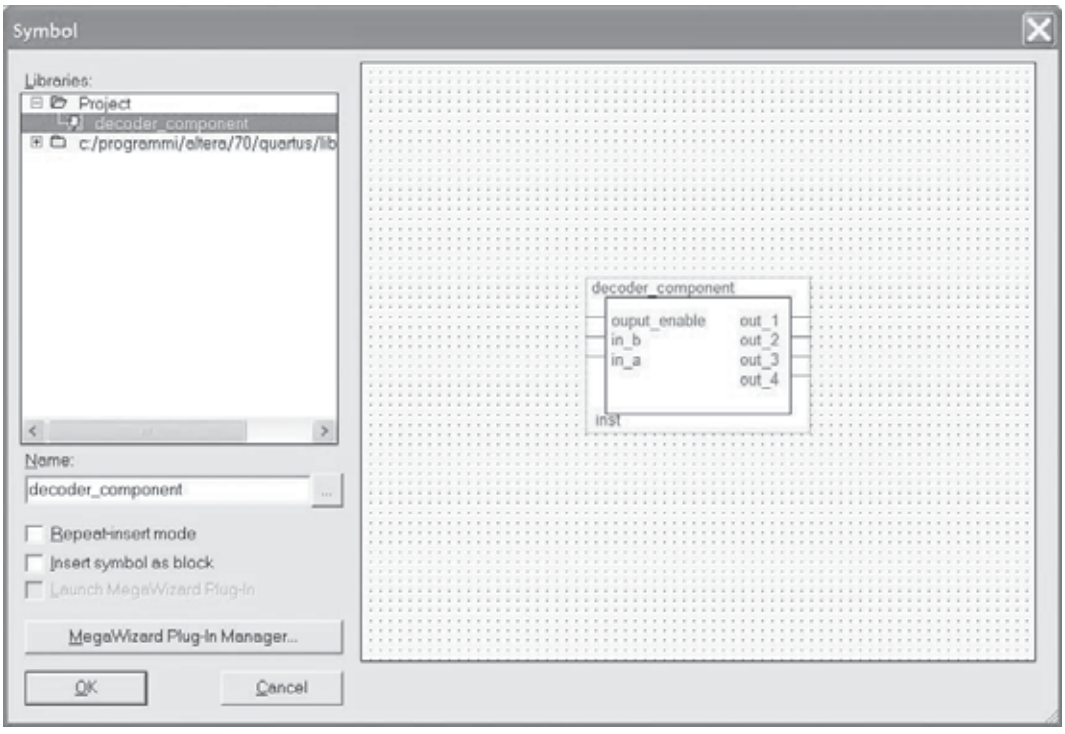


Fig.11 siguiendo las indicaciones del texto podéis visualizar el componente creado: como podéis ver, además de las dos entradas, hay una tercera que habilita en salida el código decodificado.

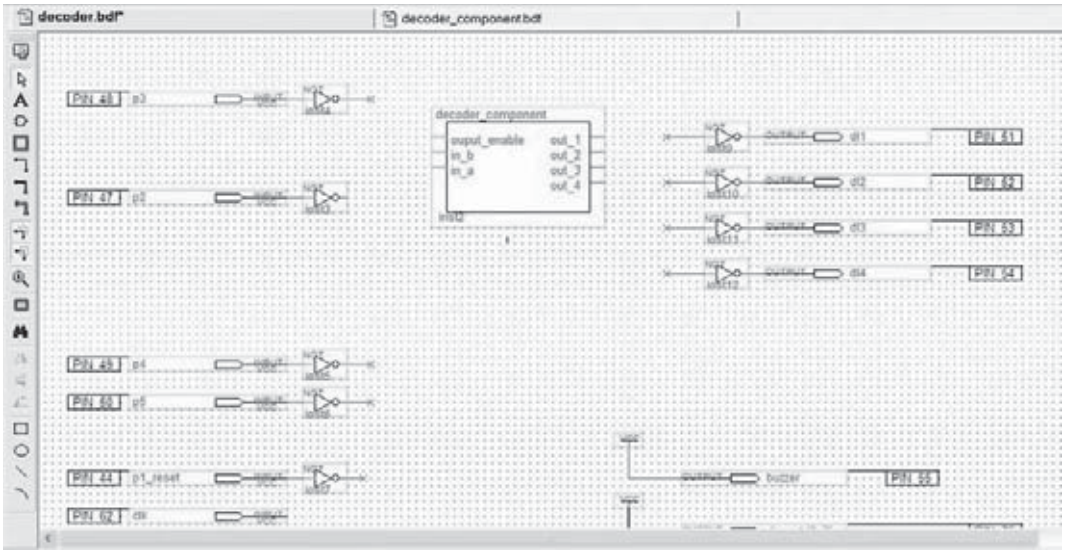


Fig.12 después, si pulsáis, en la ventana que aparece en la fig.11, "project" y sobre "decoder_component" conseguiréis el resultado que aparece aquí arriba.

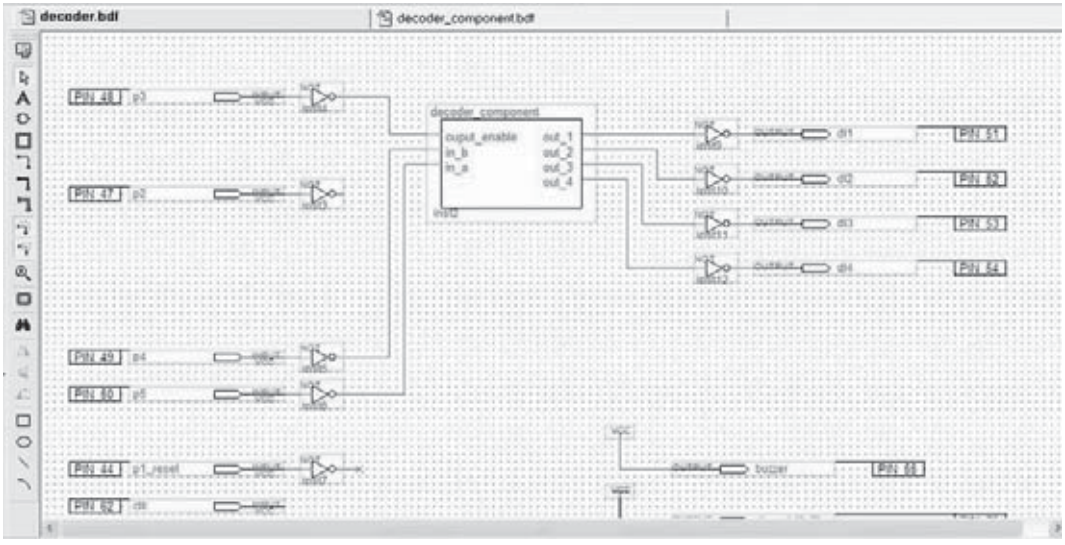


Fig.13 a continuación, interconectad las entradas y las salidas del componente a los pin de la PDL, y abrid el recopilatorio cargando el nuevo archivo de programación.

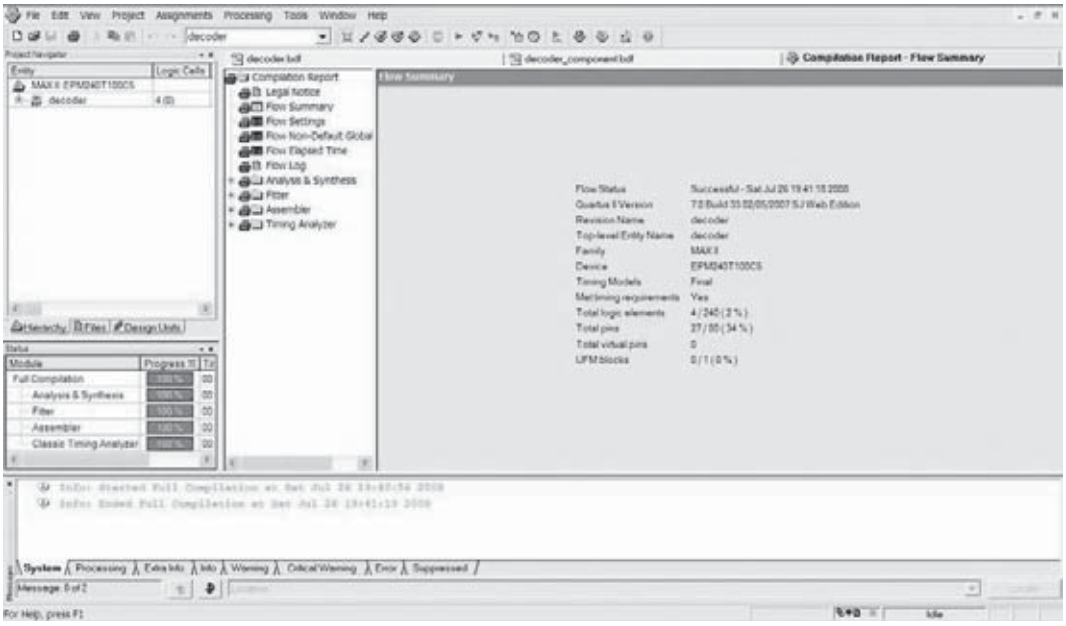


Fig.14 de los "informes" que aparecerán al final del recopilatorio veréis que el consumo de recursos, y en especial de los Logic Cells (LC) no han cambiado.

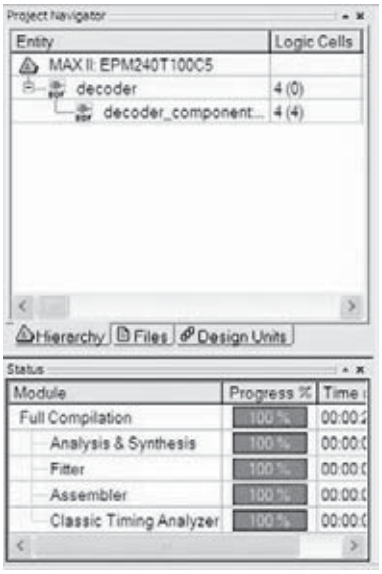


Fig.15 el número que aparece en el recuadro "Project Navigator" que antes era "4(4)", ahora es "4(0)", apareciendo, por otro lado, el "decoder_component".

Fig.16 si pulsáis sobre la carpeta del proyecto veréis, entre otras cosas, los archivos que habréis creado relativo al esquemático y al símbolo.

Antes, había un único componente denominado "**decoder.bdf**", que, además, era el **Top,** y por tanto, las **celdas totales** (**4**).

Ahora, que la jerarquía ha aumentado de profundidad, el **Top** consume **las 4 celdas**, pero **0** pertenece al mismo archivo "**decoder.bdf**", mientras que el subcomponente "**decoder_component**" a su vez consume las **4 celdas** y todas pertenecen e él mismo.

En definitiva, ahora este componente puede ser utilizado todas las veces que deseemos en el proyecto que queramos.

El procedimiento puede ser realizado para crear cualquier componente más o menos complejos, que además nos permitiría incluir otros subcomponentes.

Por ejemplo, podéis decidir si queréis utilizar en otro proyecto tanto el "**decoder_component**" como el "**decoder**", que en este proyecto es el **Top**, pero en otro podría ser utilizado de manera diferente.

Si miráis en la carpeta del proyecto encontraréis los archivos que habéis creado:

"**decoder_component.bdf**", que es el esquemático;
"**decoder_component.bsf**", que es el símbolo (ver fig.16).