
6.2 FLOORPLAN

En esta fase se imponen una serie de restricciones físicas que afectan directamente al proceso de P&R, y que permiten delimitar la zona de la FPGA que se queda libre. En el ejemplo de aplicación, la lógica correspondiente al procesador LEON, la situación física en la FPGA de la interfaz AMBA diseñada y la zona dinámica.

La planificación que sigue el P&R se realiza con la herramienta *Floorplanner*. Mediante dicha herramienta se pueden imponer restricciones de área y realizar el *placement* manual de ciertas partes del diseño

El objetivo es conseguir en una zona de la FPGA los módulos de lógica diseñados, y en otra, la lógica del diseño principal que se conecta con dichos módulos. Es necesario para en la fase de *routing*, rutar de forma automática las líneas que constituyen la interfaz, y el resto de líneas.

Si no se realizan las restricciones de área sobre los dos grupos de lógica, diseño principal y lógica adicional, separándolas a cada lado en la FPGA, los resultados que se obtendrían en la fase de *routing* no serían correctos. Podrían obtenerse líneas rutadas que no pertenecen a la interfaz en la zona dinámica.

La asignación de restricciones de agrupación de lógica y área apenas penaliza la velocidad máxima del diseño obtenida, como se comprueba tras realizar la implementación en 6.6 RESULTADOS.

Haciendo uso de la herramienta *Floorplanner*, se diseña un *floorplan* que debe de seguir las siguientes directrices:

- Realizar agrupaciones de lógica, distinguiendo entre:
 - Lógica adicional diseñada para la interfaz (*dummy logic*). Si se han usado las opciones de la herramienta de síntesis mencionados en el diseño de alto-nivel, suele aparecer agrupada por defecto en la estructura jerárquica.
 - Lógica perteneciente a la zona fija (diseño principal) que tiene como entradas o salidas señales que forman parte de la interfaz (*other side logic*). Para poder agruparla será necesario seleccionarla con la herramienta de búsqueda de *Floorplanner*.
 - Resto de lógica del diseño de la FPGA

- Se asignan restricciones de área a los grupos de lógica anteriores.
 - A los grupos de lógica adicional dentro de la zona dinámica de la FPGA (zona que se pretende dejar libre para reconfiguración).

-
- A la lógica de la interfaz del diseño principal, en la zona fija de la FPGA
 - Al grupo que engloba al resto de lógica, un área que ocupe toda la zona fija.
-
- Se deben de incorporar las restricciones que existan en el diseño para pins de E/S, intentando usar IOBs de la zona fija. De otra forma aparecen líneas rutadas por la zona dinámicas que enlazan con éstos.
 - Prohibición del resto de recursos que quedan libres, entre ellos, CLBs de la zona dinámica no utilizados e IOBs que la rodean.

El resultado del *floorplan* se encuentra en el fichero *.fnf* que tras el proceso automático de *placement* almacenará la información de localización de cada elemento lógico. Esto será aprovechado durante la fase de rutado de la interfaz.

6.2.1 FLOORPLAN: LEON

Se aplican las directrices marcadas anteriormente sobre el diseño del procesador LEON y que permiten delimitar la zona de la FPGA en que no se va a colocar lógica y la situación física en la FPGA de la interfaz AMBA diseñada.

Al incorporar al flujo de diseño un *floorplan*, las restricciones que se han impuesto en el fichero *.ucf* quedan deshabilitadas. Es importante por tanto incorporar estas restricciones al *floorplan* para que se mantengan la asignación de pins correcta.

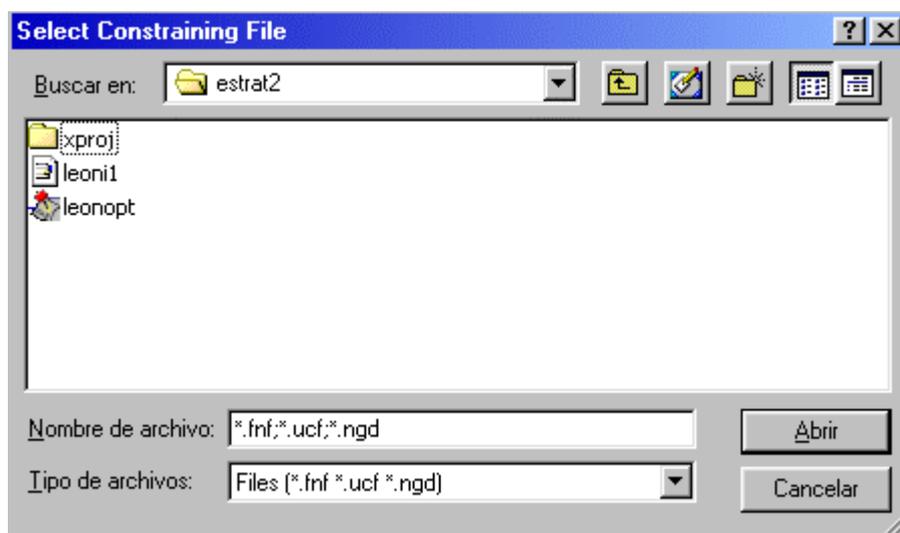


Figura 6-7: Menú de inserción de restricciones UCF

Antes de realizar el *floorplan*, se agrupa la lógica de todo el diseño. Para agrupar la lógica perteneciente al diseño principal que está conectada con las líneas que componen la interfaz será necesario seleccionarla con la herramienta de búsqueda de *Floorplanner*.

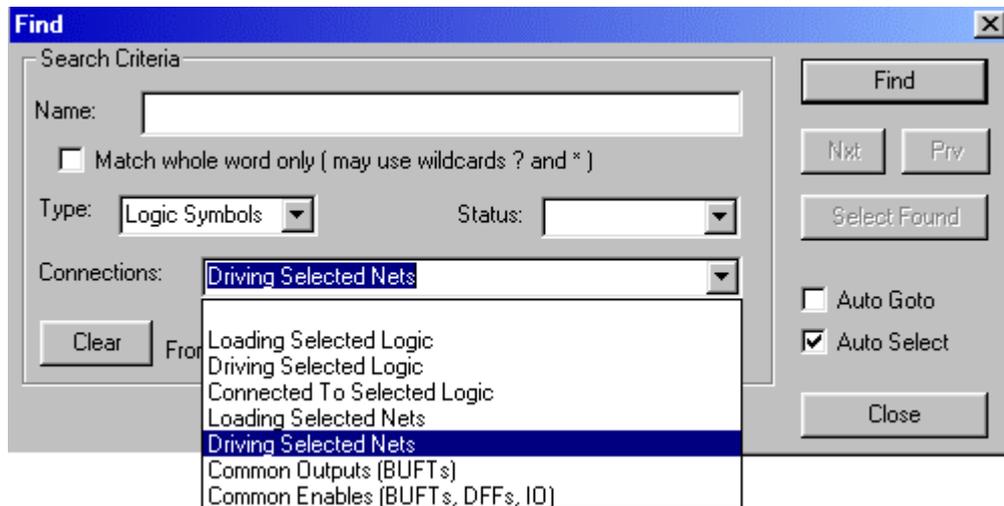


Figura 6-8: Herramienta de búsqueda *Floorplanner*

Existe el condicionante mencionado anteriormente acerca de la memoria RAM que se debe ocupar en la zona dinámica de la FPGA debido a escasez de recursos. Es necesario ocupar dos bloques RAM en la zona dinámica, y para controlar a la herramienta de P&R, se realiza el tratamiento como si fuera un grupo de lógica adicional.

Se forman los siguientes grupos:

- procesador LEON
- lógica conectada con la interfaz del procesador (*other side logic*)
- memoria RAM configurada para funcionar como PROM
- interfaz APB, a interfaz AHB esclavo, e interfaz AHB maestro de la zona dinámica

En la figura 6-9 se muestra la estructura jerárquica que existe tras la agrupación.

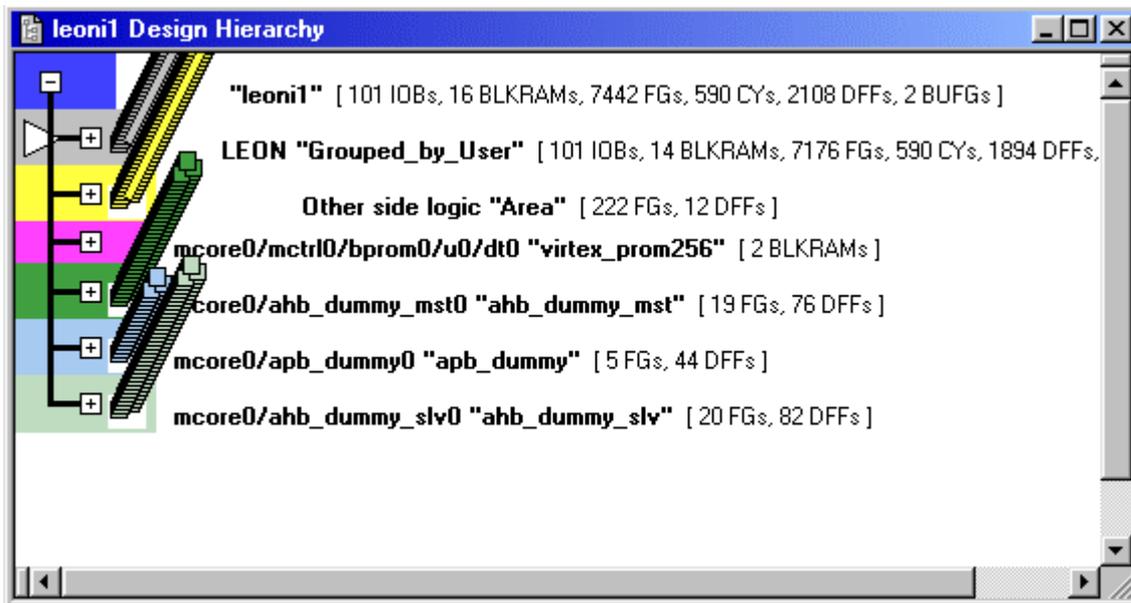


Figura 6-9: Grupos lógicos formados en el LEON

En el *floorplan* se configuran las siguientes restricciones sobre el mapa de recursos de la FPGA:

- Se incorporan las restricciones de E/S que existen en el fichero *.ucf* previo.
- Al grupo de lógica que engloba al procesador se le impone una restricción de área. De esta forma se fija el área de FPGA que será fija y el resto zona dinámica. El área de forma rectangular que se convierte en fija se sitúa en la parte de la FPGA en que existen más IOBs ocupados por las restricciones de la zona anterior.
- El grupo de memoria PROM instanciada se coloca manualmente en el lado de dinámico de la FPGA. Esto permite que el resto de RAM utilizada para los registros del procesador y caché sea colocada de forma automática por la herramienta de P&R en el otro lado de la FPGA, ocupándola por completo. De otra forma la herramienta de P&R podría desagrupar los bloques de RAM que forman los registros y colocarlos en ambas columnas de la FPGA.

- Se asignan restricciones de área a los componentes que constituyen la interfaz APB y AHB dentro de la zona que se quiere dejar libre de lógica. En la elección de la zona se consideran los lados de la FPGA que están menos ocupados por las restricciones de entrada/salida.
- Se anotan las restricciones impuestas para cada grupo de lógica, con la notación CLB_R?C?:CLB_R?C?, pues serán utilizadas en la siguiente fase junto con el fichero *floorplan* (.fnf) obtenido tras el *placement*.
- Por último, se imponen restricciones de prohibición de recursos, que permiten delimitar la zona de la FPGA que se queda libre, así como prohibir el uso de determinados IOBs y bloques de RAM reservados para su utilización en los módulos que se incorporen dinámicamente.

Tras estas acciones, el *floorplan* que resulta se muestra en la siguiente figura:

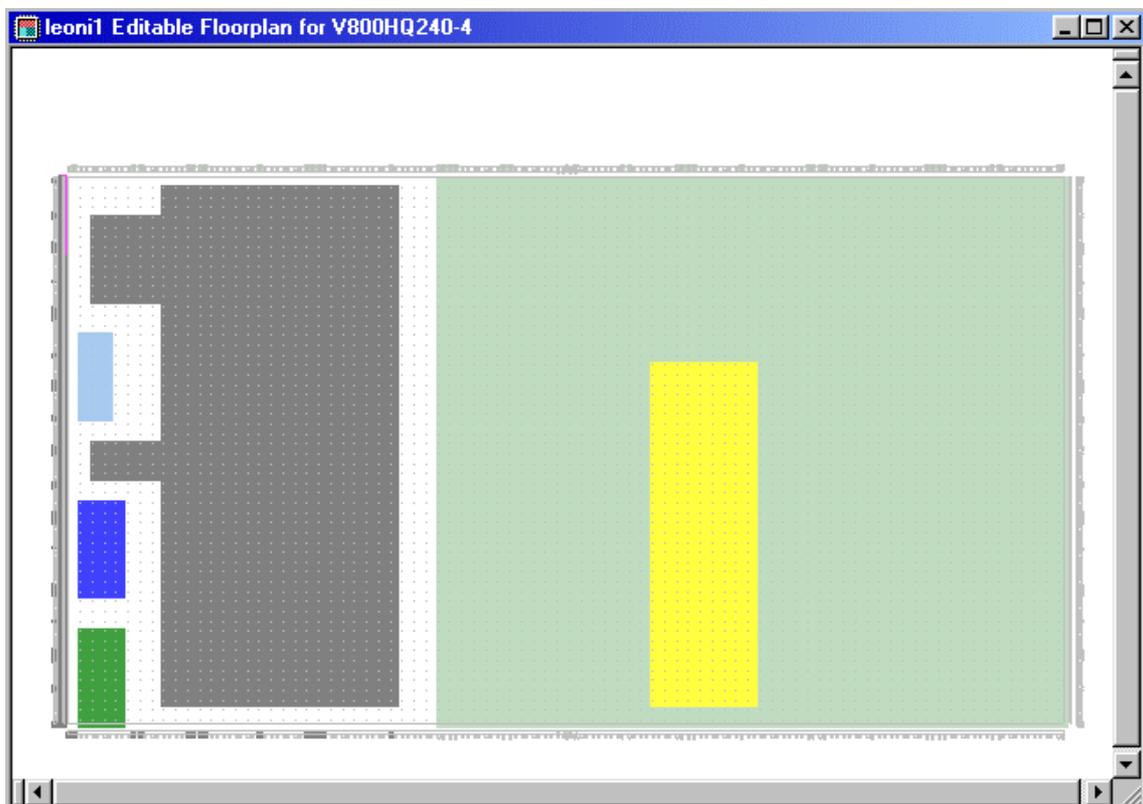


Figura 6-10: *Floorplan*

6.2.2 PLACEMENT

Una vez realiza el *floorplan*, se incorpora al flujo de diseño ejecutando de nuevo la herramienta *map*.

Después se usa la herramienta *par*, pero habilitando sólo la opción de *placement* automático. El resultado es el diseño completamente colocado pero sin ninguna línea rutada.

```
map -p xcv800-4-hq240 -o map.ncd -fp leoni1.mfp leoni1.ngd  
leoni1.pcf
```

```
par -w -ol 2 -r map.ncd leoni1.ncd leoni1.pcf
```

La siguiente figura muestra el resultado del *placement* automático realizado por la herramienta *par*, en *FPGA Editor*:

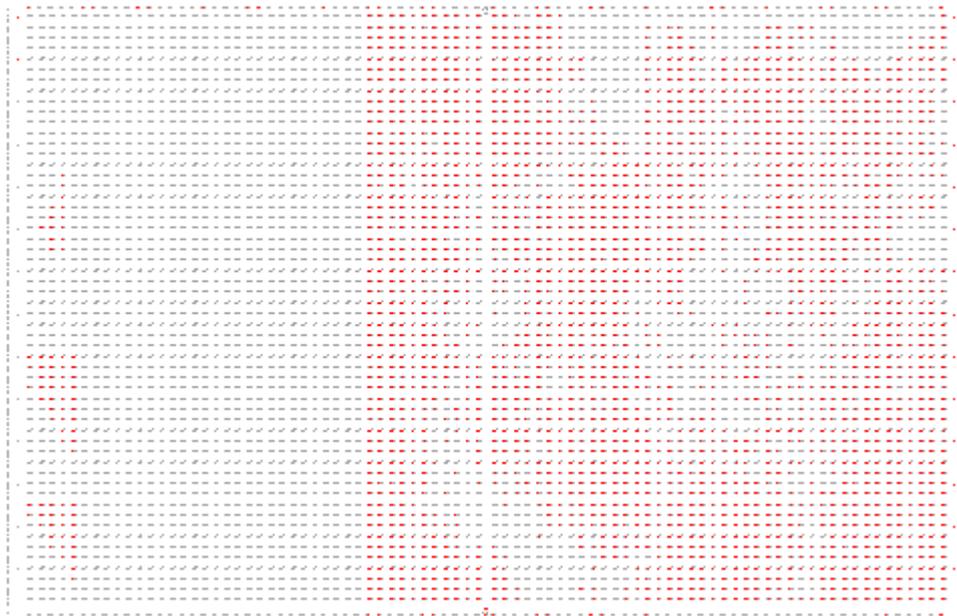


Figura 6-11: *Placement* LEON (*FPGA Editor*)

Para comprobar que la lógica perteneciente a los distintos grupos ha sido colocada respetando las restricciones de área fijadas, se puede observar el *placement* con *Floorplanner*.

Hay que observar los resultados, pues puede ocurrir que las áreas asignadas sean demasiado reducidas para la lógica que se quiere colocar y no se hayan respetado las restricciones.

Además al utilizar la herramienta *Floorplanner* para observar el *placement*, se produce una actualización del fichero *floorplan*, indicando en que CLB ha sido colocado cada elemento de la lógica, información que será necesaria en la fase de *routing* para poder rutar las líneas de la interfaz.

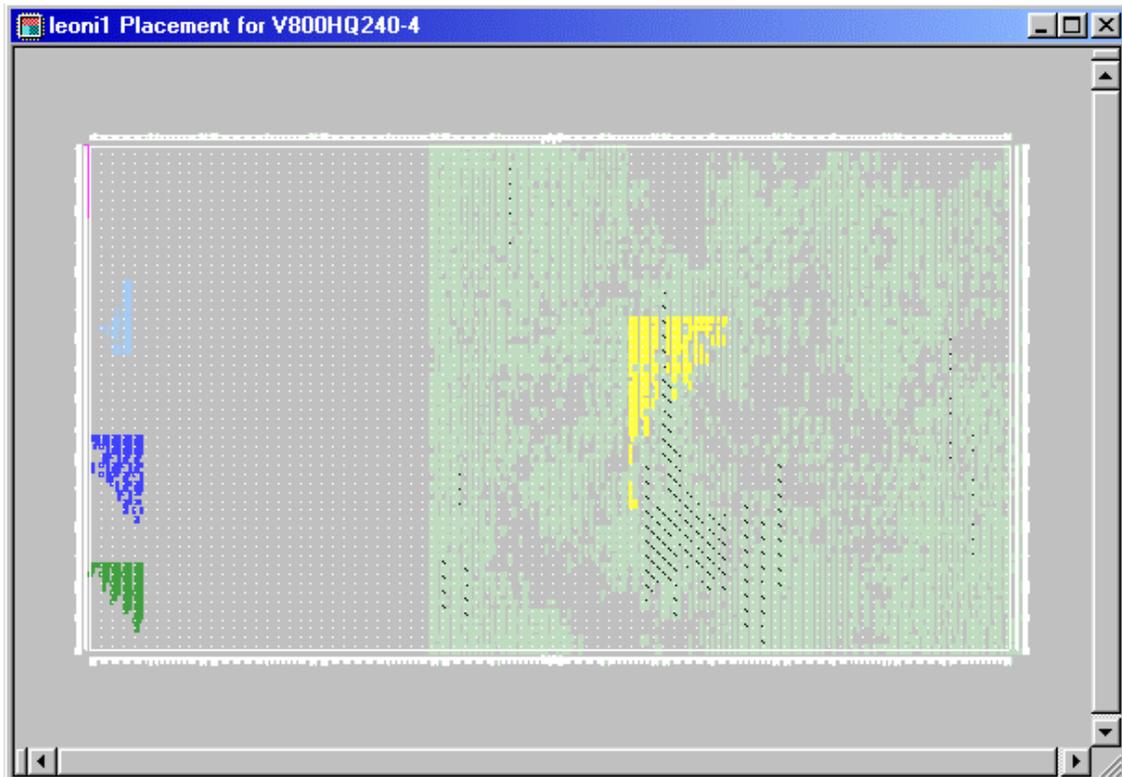


Figura 6-12: *Placement* LEON (*Floorplanner*)

Se puede intentar realizar con *floorplanner* el *placement* manual de la lógica que forma los distintos grupos, para saber a priori la posición en que se colocan. Pero al realizar el mapeado a CLBs, la herramienta *map* y posteriormente *par*, no respetan las restricciones de *placement* en una determinada posición y los dispersa por toda la FPGA.

Se ha comprobado que funciona siempre correctamente con restricciones de área, pero no con el *placement* exacto de un determinado elemento lógico (BEL).