

Práctica 1

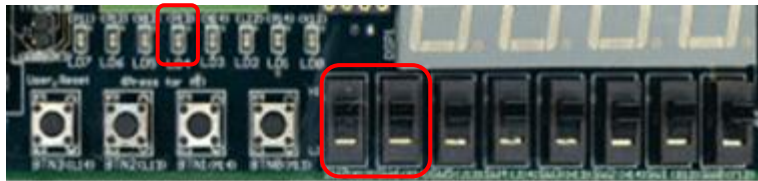
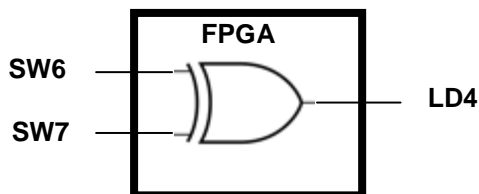
Tutorial

Objetivo

Usando un diseño especialmente simple, seguir con él el flujo básico, descargando el diseño sobre la placa y verificando en ella su funcionamiento.

Circuito utilizado

Se trata simplemente de hacer una función xor con dos interruptores (SW6 y SW7) y volcarla sobre un LED (LD4):



Guía para la realización

Con el fin de tener los ficheros fuente VHDL del diseño ordenados y claramente localizables, por separado del proyecto ISE, usaremos la siguiente estructura de directorios:

```
<directorio del ejercicio> /
    rtl/
    sim/
    ise_<nombre_diseño>/
```

donde *<directorio del ejercicio>* será a su vez *P<número_de_práctica>_<nombre_diseño>*.

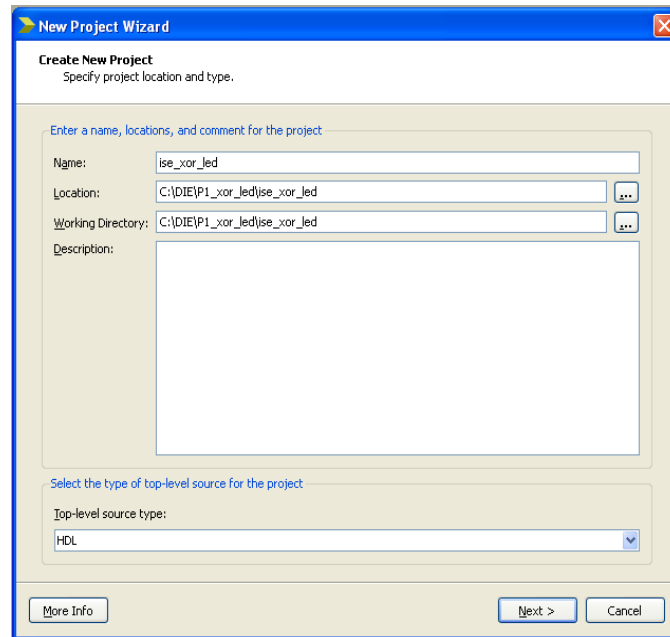
Así pues, crear el directorio “C:\DIE” (o “C:\DCSE”), dentro el directorio “P1_xor_led” y dentro los directorios rtl/ y sim/.

Ejecutar el Project Manager de Xilinx: icono *ISE Design Suite 14.1*.

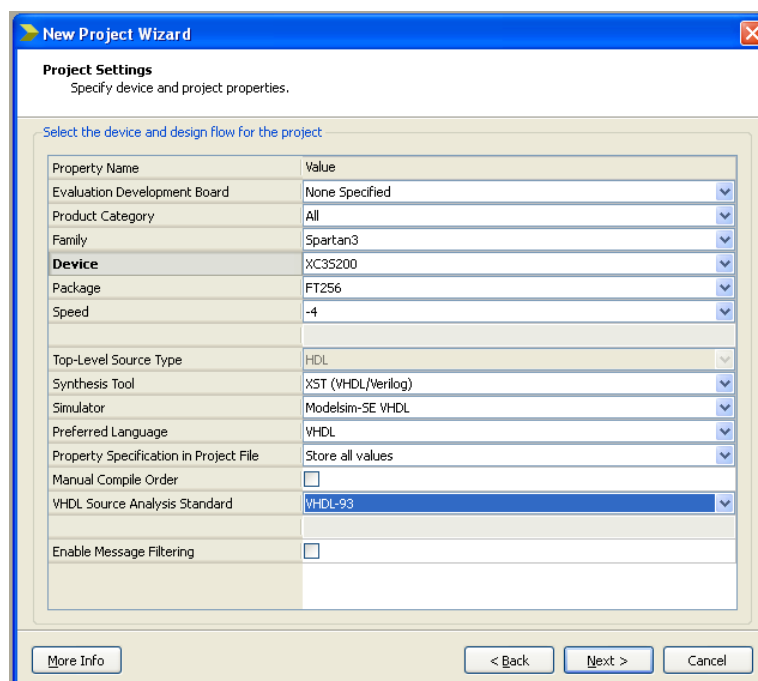


Crear un nuevo proyecto:

- Usar *File / New Project*.
- Seleccionar el directorio donde se ubicará en *Location*, dando al botón "...". Elegir el directorio "P1_xor_led".
- A continuación poner como nombre de proyecto "ise_xor_led".
- Comprobar abajo que el *Top-level source type* es *HDL*.
- El cuadro de diálogo deberá tener un aspecto como el siguiente:



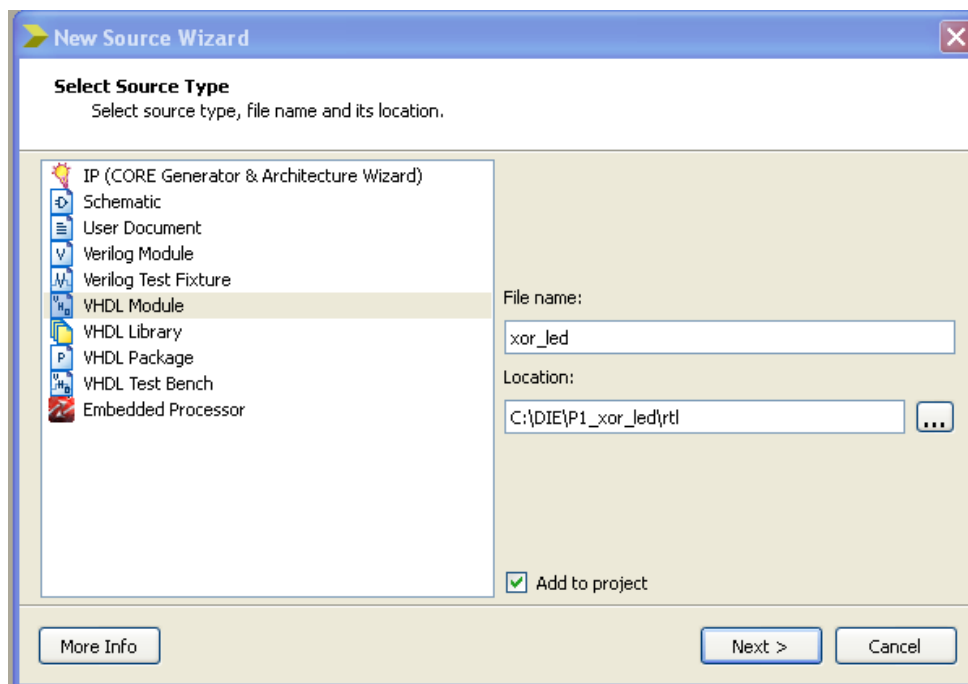
- Dar a *Next*. En la siguiente pantalla comprobar o seleccionar los parámetros de configuración del proyecto de acuerdo con la siguiente figura:



- Dando a *Next* aparece una pantalla con el resumen del proyecto. Dar a *Finish*.

Preparar el RTL:

- Crear un nuevo fichero fuente seleccionando *Project / New Source*.
- Seleccionar a la izquierda “VHDL Module”.
- En *Location* usar el botón “...” para posicionarse en el directorio *rtl/*.
- Elegir como nombre “xor_led” (lo cual dará lugar a la creación de un fichero “xor_led.vhd”).



- A continuación aparece una ventana de configuración de puertos. Como el diseño es muy simple y conviene practicar la memoria con el VHDL, se recomienda saltarse este punto con *Next*. Finalmente, dar a *Finish*.
- En la zona derecha de la herramienta, en la parte de abajo de la ventana de edición, podemos pinchar en la solapa “xor_led.vhd” para acceder a la edición del fichero fuente creado. Disponemos de una plantilla de Xilinx, que habremos de rellenar con el diseño:
 - Añadir a la *entity* la lista de puertos: SW6 y SW7 de entrada y LD4 de salida.
 - Editar la arquitectura con una asignación concurrente que haga que LD4 sea el or exclusivo de SW6 y SW7. (Como en la placa los LEDs se acceden como activos a nivel alto la implementación es directa).
- Guardar el diseño (*File / Save*, Ctrl-S o icono con disquete).

Preparar el testbench para la simulación:

- Crear un nuevo fichero fuente seleccionando *Project / New Source*.
- Esta vez seleccionar a la izquierda “VHDL Test Bench” y a la derecha en *Location* el directorio “sim”. Como nombre para el fichero usar “xor_led_tb”.
- Al dar a *Next* nos permite seleccionar para qué diseño es el testbench. En este caso sólo tenemos una elección, “xor_led”.
- Editar el código fuente del testbench:
 - Eliminar las líneas innecesarias de la plantilla creada por ISE, en particular las relacionadas con el uso de un reloj que no tenemos.
 - Si fuera necesario, editar la declaración del componente *xor_led* y su instanciación.
 - Sustituir el contenido del proceso de estímulos por un único bucle *for* que vaya de 0 a 3, asignando a la pareja (SW7, SW6) el valor *std_logic_vector* correspondiente a cada iteración y comprobando tras 10 ns con un *assert* que la salida es la esperada. Tras el *for* deberá quedar un *wait* para terminar el proceso.
 - Si se ha usado la función *conv_std_logic_vector*, sustituir en la parte superior el uso del package *ieee.numeric_std* por el de *ieee.std_logic_arith*.
- Guardar el diseño.

Simular el diseño funcionalmente:

- Cambiar la vista del panel superior izquierdo (*View:*) a *Simulation*.
- Seleccionar el testbench en la ventana superior izquierda y simular haciendo doble clic en la ventana inferior izquierda sobre *ModelSim Simulator / Simulate Behavioral Model*.
- Comprobar en la ventana de ondas y en la consola de ModelSim que el circuito hace lo esperado. Por defecto ISE encargará a ModelSim que simule 1000 ns, por lo que en la ventana de ondas tendremos que hacer zoom en la zona inicial para ver bien las señales.

Asignar los pines de la FPGA:

- Crear un nuevo fichero fuente, esta vez de tipo *Implementation Constraints File*. Crearlo en el directorio “ise_xor_led”, con nombre “xor_led”, que dará lugar a la creación de un fichero “xor_led.ucf”.
- En la ventana superior izquierda del *Project Navigator* volver a cambiar la vista (*View:*) a *Implementation*. Podremos ver ahora, bajo el diseño “xor_led”, una línea con el fichero “xor_led.ucf”. Pinchar en ella con un solo clic del ratón (un doble clic dispara todo el proceso de implementación).
- Desplegar en el panel de *Processes* la línea *User Constraints* y hacer doble clic sobre *Edit Constraints (Text)*. Aparecerá en el editor el fichero “xor_led.ucf”, en blanco.
- En el manual de la placa podemos encontrar los pines de la FPGA correspondientes a cada interruptor y a cada LED. Usaremos los interruptores SW7 y SW6, y el LED LD4. Podemos indicarle a la herramienta que use los pines adecuados escribiendo lo siguiente en el fichero “xor_led.ucf”:

```
NET "SW7"    LOC = K13;  
NET "SW6"    LOC = K14;  
NET "LD4"    LOC = P13;
```

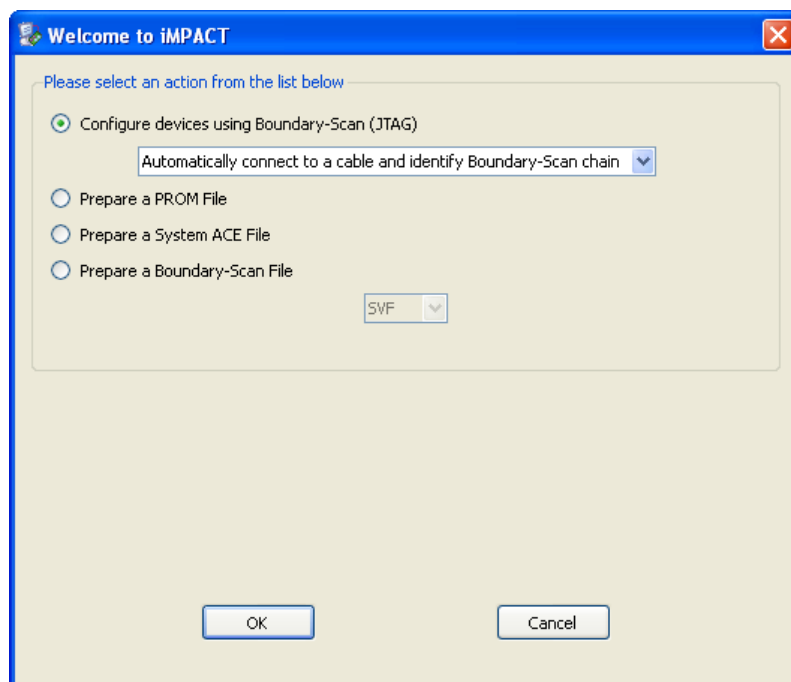
- Nótese que los pines de la FPGA correspondientes a cada interruptor y LED están también serigrafiados en la propia placa.

Realizar la implementación

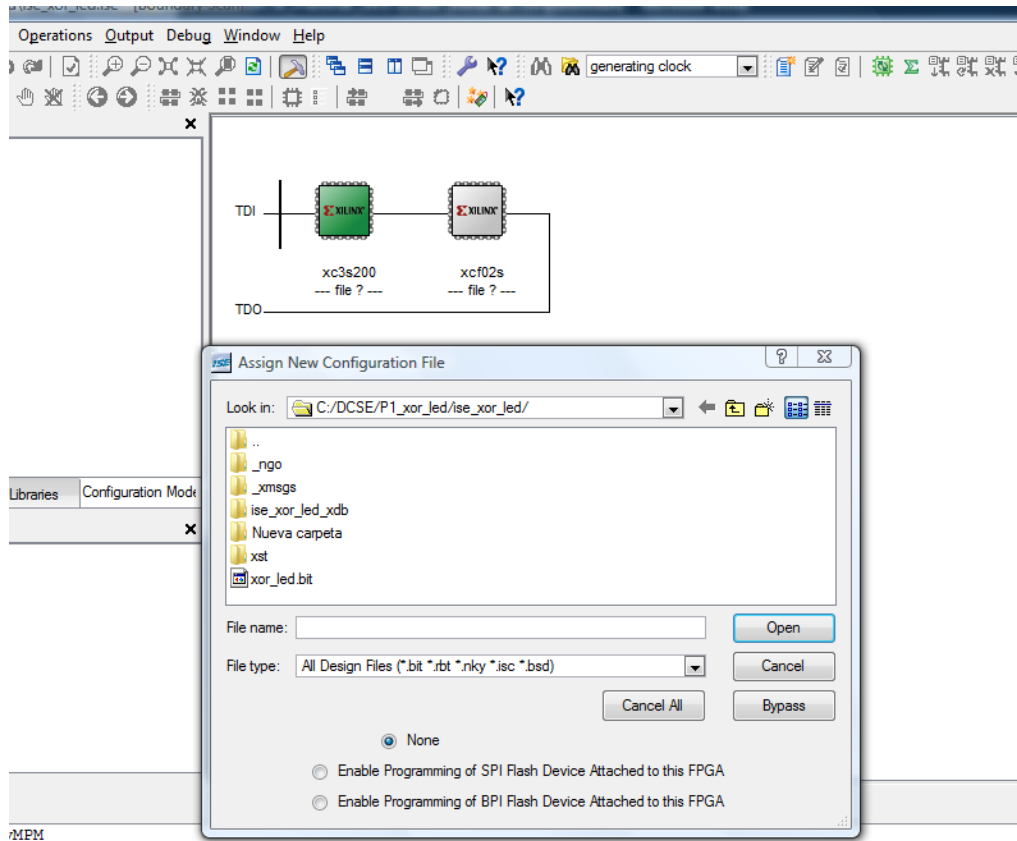
- Seleccionar en el panel *Design / Hierarchy* la línea que comienza por “xor_led”, inmediatamente debajo de la que pone el identificador del modelo de FPGA (“xc3s200-4ft256”).
- Mostrar los subprocesos de implementación haciendo clic sobre el “+” a la izquierda de *Implement Design*.
- Dispararemos la implementación completa haciendo doble clic en *Generate Programming File*.
- Observar cómo evoluciona la implementación, en los iconos de la ventana *Processes* y en la consola. Al finalizar comprobar las solapas *Errors* y *Warnings* de la consola.
- En la ventana principal seleccionar la solapa *Design Summary* y ya dentro de ella *Design Overview / Pinout Report*. Comprobar que en la columna *Signal Name* aparecen nuestras 3 señales en los pines adecuados.

Bajar el diseño a la placa:

- En la ventana *Processes*, hacer doble clic sobre *Configure Target Device / Manage Configuration Project (iMPACT)*. Si es necesario responder afirmativamente a un cuadro de diálogo que puede aparecer antes de abrirse la herramienta *iMPACT*, tras de lo cual se nos pregunta por la acción a efectuar, siendo la acción por defecto adecuada (*Configure devices using Boundary-Scan (JTAG)*):



- Al dar a OK nos sale una pantalla con la cadena JTAG dibujada (aparecen la FPGA y su memoria de configuración), y se nos pide seleccionar un fichero. Comprobar que el path corresponde al directorio *ise_xor_led* o de lo contrario seleccionar este directorio. Hay que prestar atención pues esta herramienta conserva el path usado en otras ocasiones, así que si tenemos más de un directorio con el mismo nombre tendremos que asegurarnos de que el path completo es el correcto. Elegir el fichero “xor_led.bit” y dar a *Open*.



- El fichero “xor_led.bit” queda asignado a la FPGA, y el foco pasa a la memoria de configuración (que toma un color verde). Se nos vuelve a preguntar por un fichero. En este caso seleccionar *Bypass*, pues no queremos programar la memoria de configuración sino la FPGA directamente.
- Finalmente dar a *OK*.
- Hacer clic con el botón derecho sobre el dibujo de la FPGA y seleccionar *Program*. Tras un instante, la FPGA queda programada.
- Comprobar en la placa que la implementación ha sido correcta moviendo los interruptores SW6 y SW7.

Entrega de la práctica

El correcto funcionamiento del diseño sobre la placa “S3BOARD” deberá mostrarse al profesor de prácticas a lo largo del primer día de clase, así como los ficheros editados.

No es necesaria ninguna entrega adicional.