

INTRODUCCIÓN AL SOFTWARE ISE (Integrated Software Environment) DE XILINX

Índice

GUÍA DE INICIO AL SOFTWARE ISE DE XILINX	1
1. Introducción	1
2. Inicio del software ISE (Integrated Software Environment) de Xilinx.	2
3. Realización de sistemas digitales mediante esquemáticos.	2
3.1. Inicio de un nuevo proyecto.....	2
3.2. Realización del diseño mediante esquemáticos	6
3.3. Verificación del diseño. Vector de pruebas y simulación.....	8
4. Diseño de circuitos secuenciales mediante grafos de estados.	9
4.1. Herramienta de edición gráfica de máquinas de estados	9
4.2. Incorporación al proyecto de los archivos VHDL y del vector de pruebas.	13
5. Programación de la FPGA.....	13
5.1. Asignación de entradas y salidas	13
6. Diseño en VHDL de un decodificador BCD/7 segmentos	16

GUÍA DE INICIO AL SOFTWARE ISE DE XILINX

1. Introducción

Actualmente cualquier proceso de ingeniería dispone de un soporte software que asiste al ingeniero de aplicaciones o sistemas en el desarrollo de sistemas complejos. Los sistemas electrónicos reconfigurables del tipo FPGA son un buen ejemplo de la complejidad que se puede alcanzar, esta complejidad no sería abarcable sin la ayuda de un entorno con herramientas que asistan en el proceso de diseño, simulación, síntesis del resultado y configuración del hardware. Un ejemplo de un entorno de este tipo es el software de la empresa Xilinx denominado ISE (*Integrated Software Environment*). Este software constituye un verdadero entorno EDA (*Electronic Desing Automation*). La Figura 1.1 representa el esquema de los componentes más importantes del ISE y la secuencia en que se utilizan.

La interfaz gráfica de usuario (GUI: *Graphic User Interface*) se denomina *Proyect Navigator* y facilita el acceso a todos los componentes del proyecto. Los diseños de usuario se pueden introducir mediante diferentes formatos. Los más utilizados son: los esquemáticos, los grafos de estados y las descripciones hardware en VHDL. Una vez compilados los diseños se puede simular su comportamiento a nivel funcional o a nivel temporal. A nivel funcional no tiene en cuenta los retardos provocados por el hardware y a nivel temporal simula el diseño teniendo en cuenta cómo se va a configurar el hardware.

Esta guía pretende servir de ayuda a los usuarios noveles que utilizan por primera vez este tipo de herramientas.

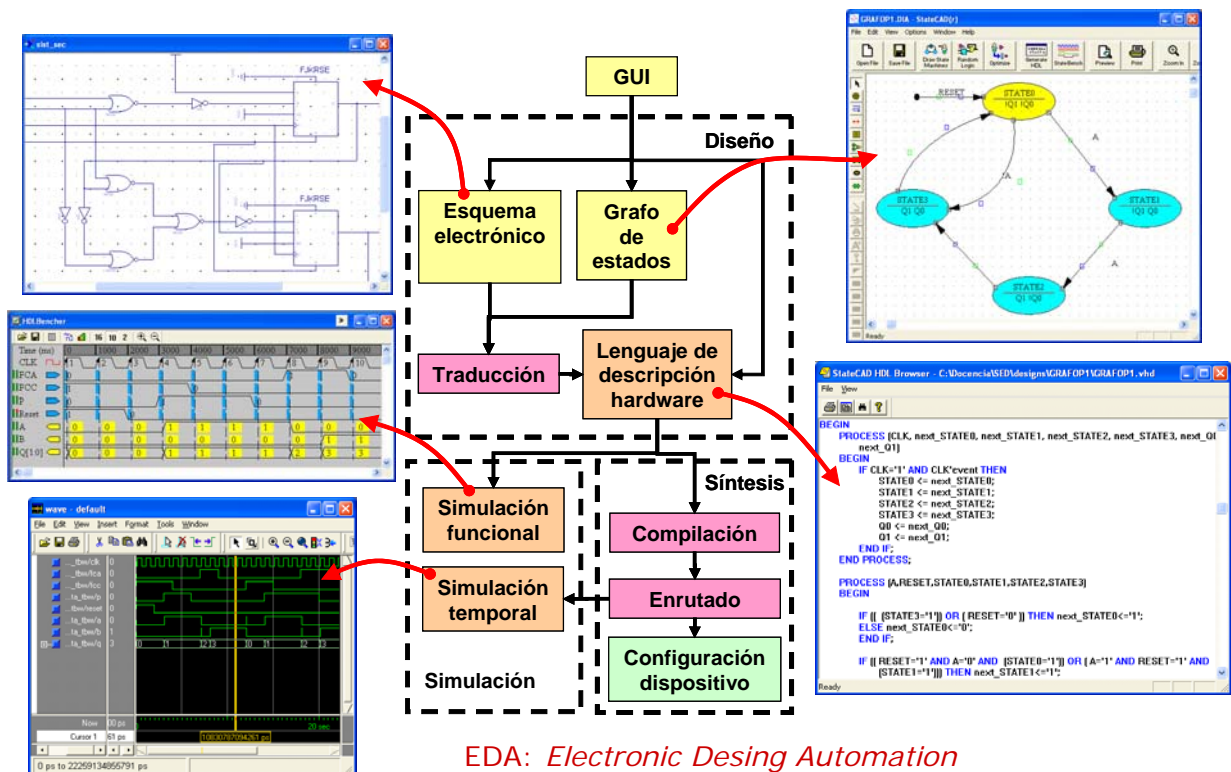


Figura 1.1. Pasos en el desarrollo de aplicaciones con FPGAs.

2. Inicio del software ISE (Integrated Software Environment) de Xilinx.

Se comienza ejecutando el software gestor de proyectos mediante el comando:

Inicio>Programas>Xilinx ISE 6>Project Navigator

A continuación se abre la ventana principal como muestra la Figura 2.1.

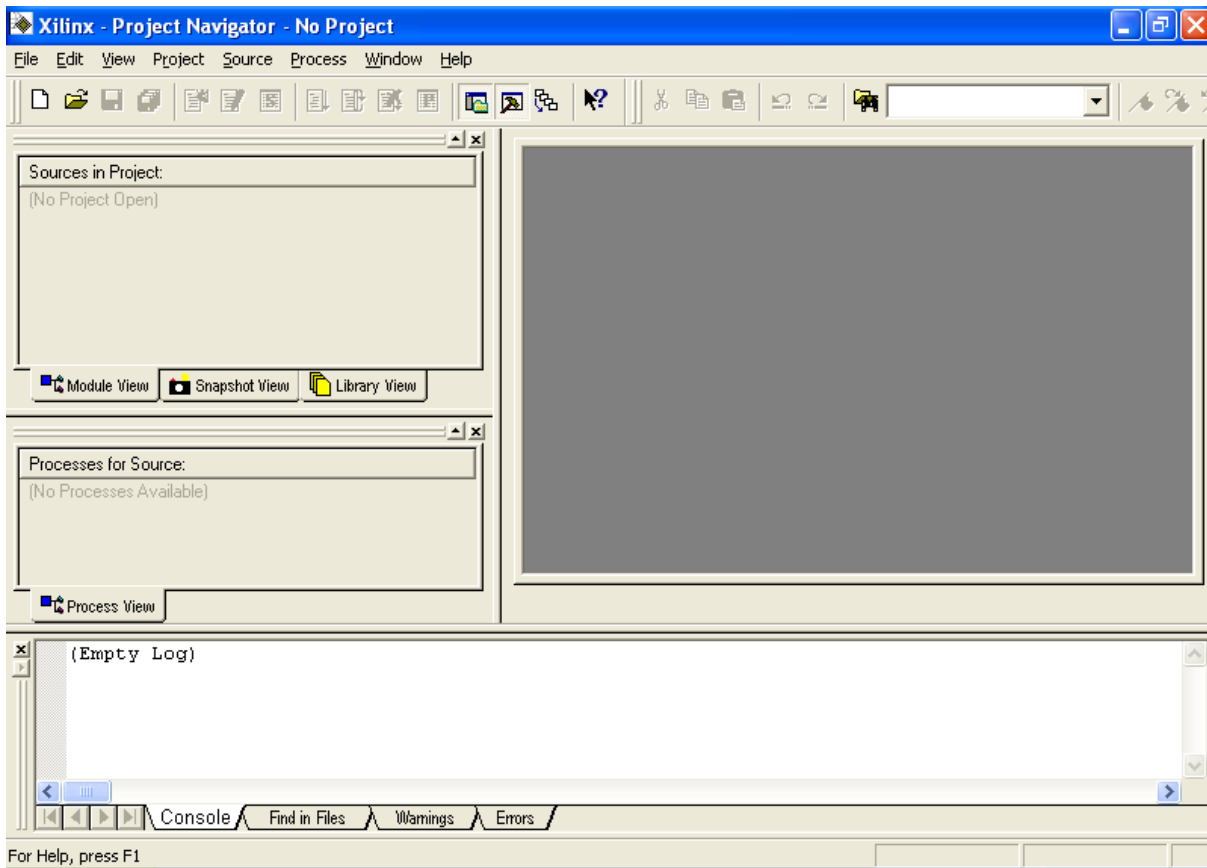


Figura 2.1. Ventana principal del Navegador de Proyectos.

3. Realización de sistemas digitales mediante esquemáticos.

3.1. Inicio de un nuevo proyecto.

En el menú desplegable superior se elige la opción:

File>New Project

A continuación se abre un cuadro de dialogo *New Project* como muestra la Figura 3.1, en el que se asigna un nombre al proyecto, el directorio donde estarán los archivos del mismo y el tipo de módulo para el nivel superior del diseño (en este caso será de tipo esquemático).

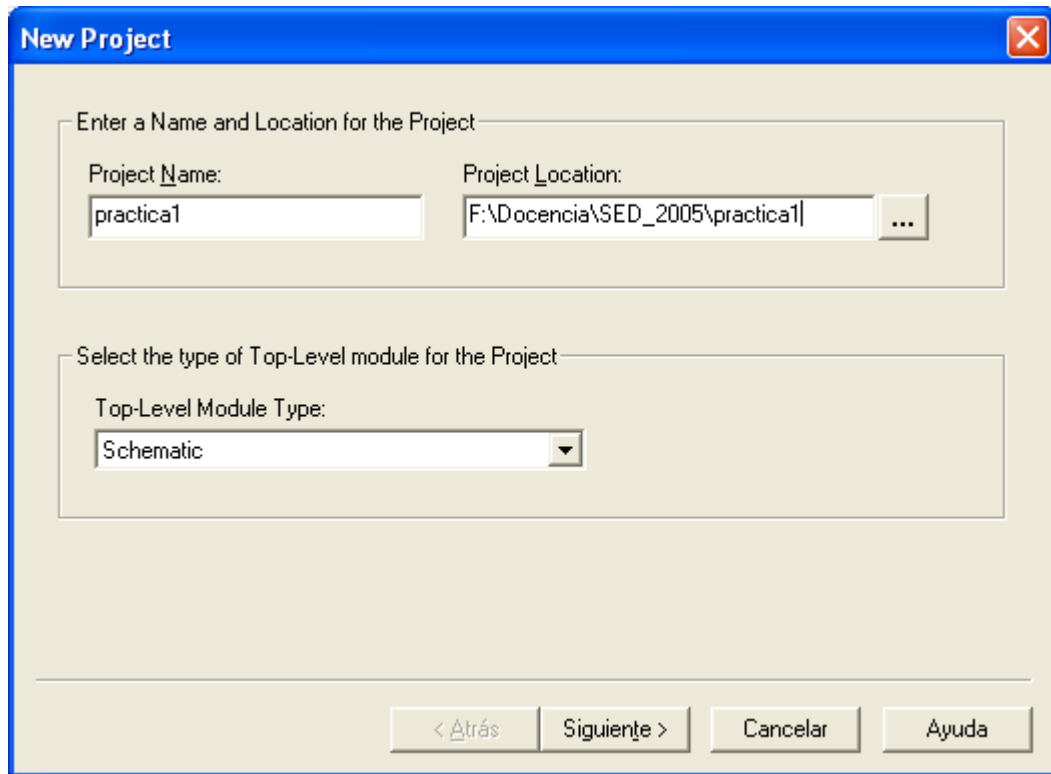


Figura 3.1. Ventana de inicio del asistente para crear proyectos.

A continuación pulsar la tecla *siguiete* y aparecerá el cuadro de dialogo de la Figura 3.2. En él se asigna el dispositivo y sus atributos principales para el que se realizará la síntesis del diseño. Para las demás opciones configurarlas como aparecen en la Figura 3.2.

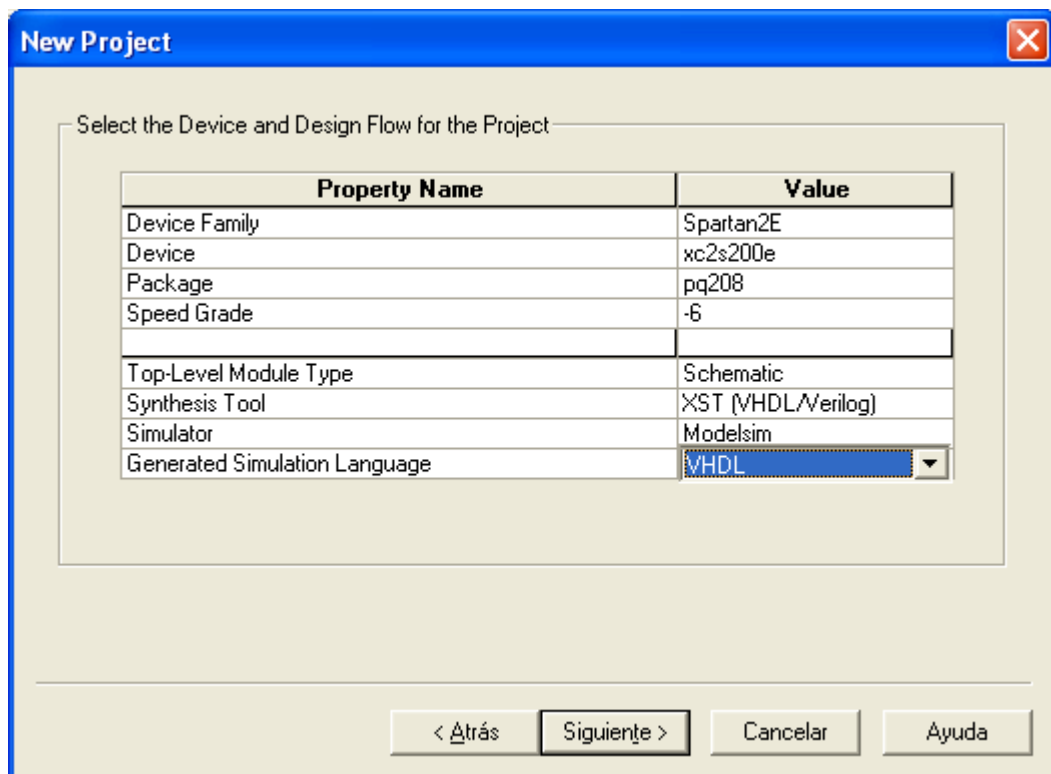


Figura 3.2. Opciones para el dispositivo y el proyecto.

Una vez establecidos los parámetros básicos del proyecto se crea un fichero fuente para el diseño. En este caso se creará un fichero para contener un esquemático tal y como muestra la Figura 3.3.

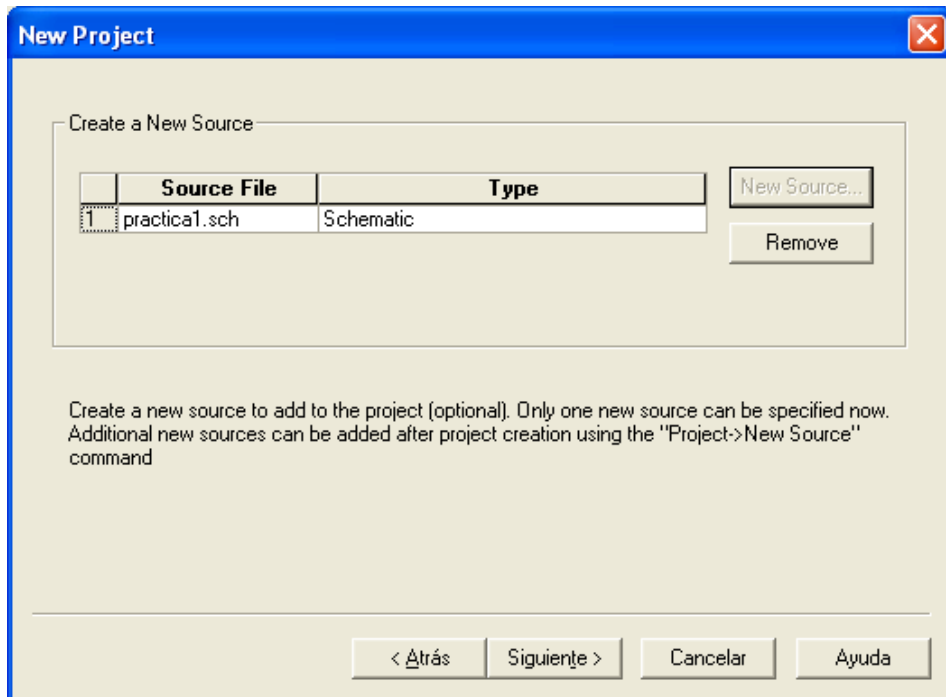


Figura 3.3.

Después de crear el fichero para el esquemático se pulsa la tecla *siguiete* y debe aparecer el cuadro de la Figura 3.4 en el que hay un listado con la información del proyecto. En el apartado *sources* aparece el esquemático que hemos creado.

Llegados a este punto queda configurado e inicializado el proyecto de este ejemplo. En la Figura 3.5 se presenta la apariencia del navegador de proyectos.

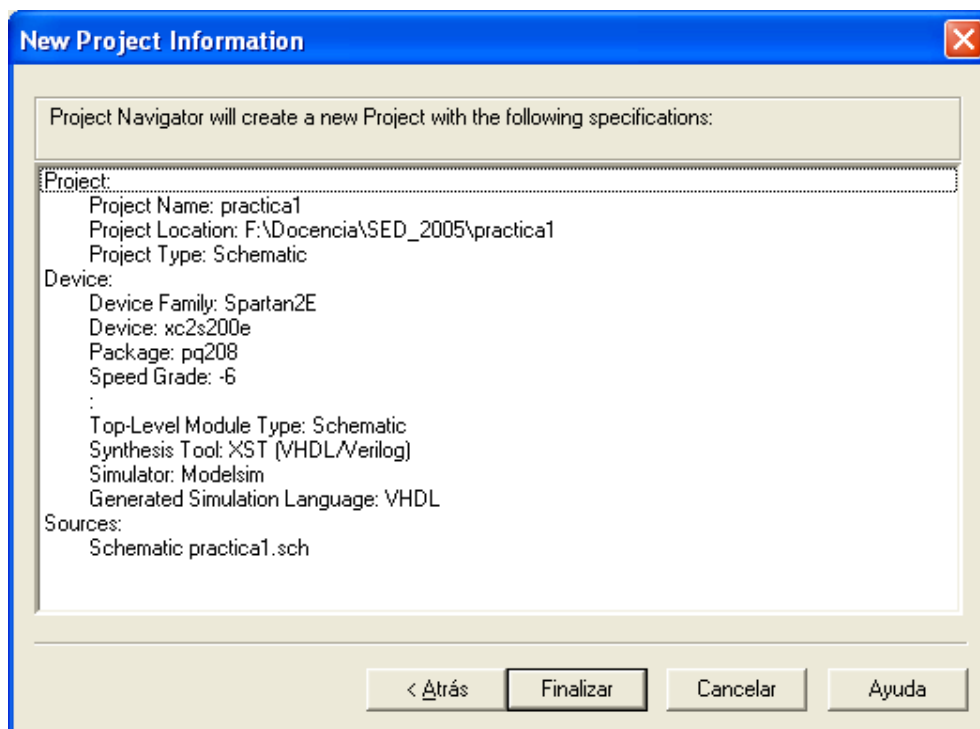


Figura 3.4. Resumen de las propiedades del proyecto.

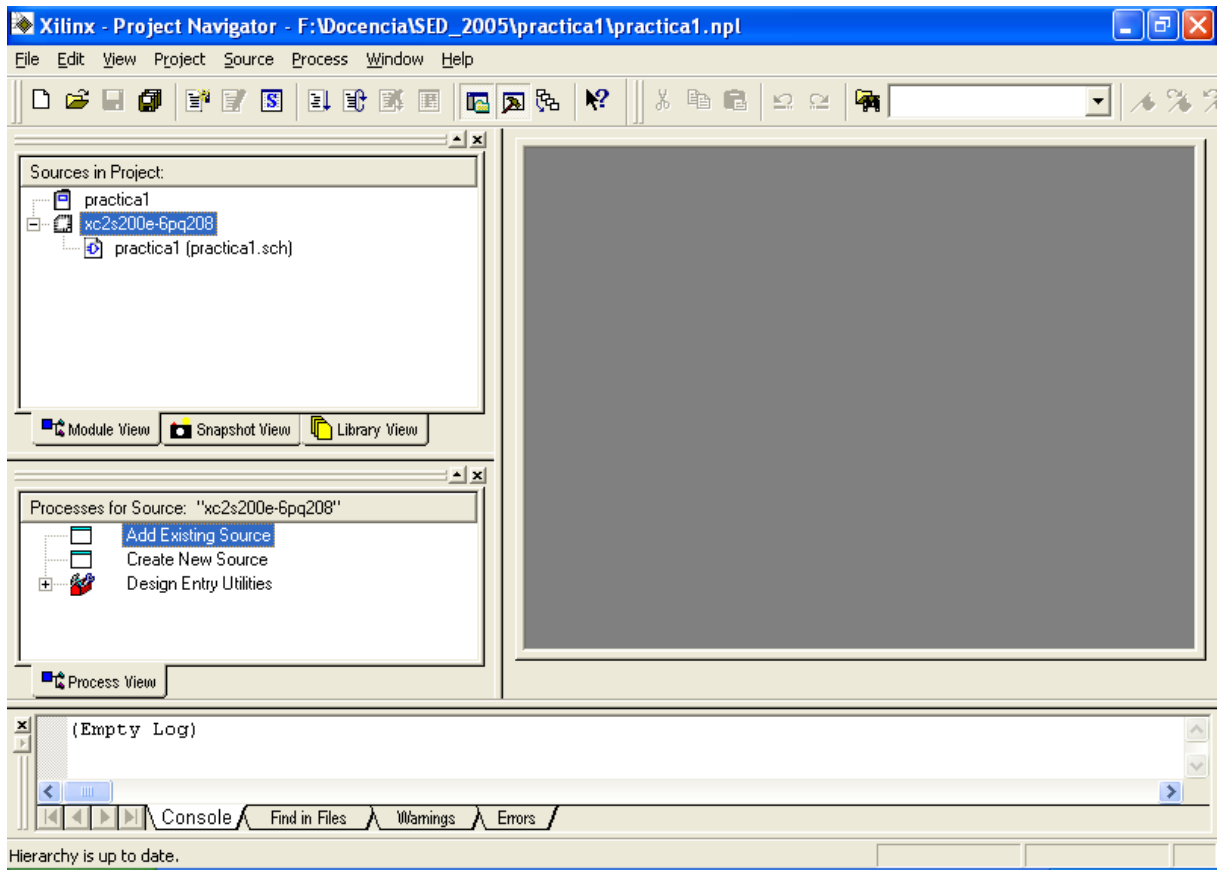


Figura 3.5. Vista general del proyecto creado.

Si no se hubiese añadido el fichero del esquemático en el asistentes del proyecto o si se desea añadir algún otro se puede hacer con la opción **Project>New Source** (Figura 3.6). Es importante recordar que los nombres que se le asignen a los ficheros fuente deben ser con el formato DOS 8.1 y por tanto hay que evitar utilizar caracteres especiales, rutas demasiado largas, etc.

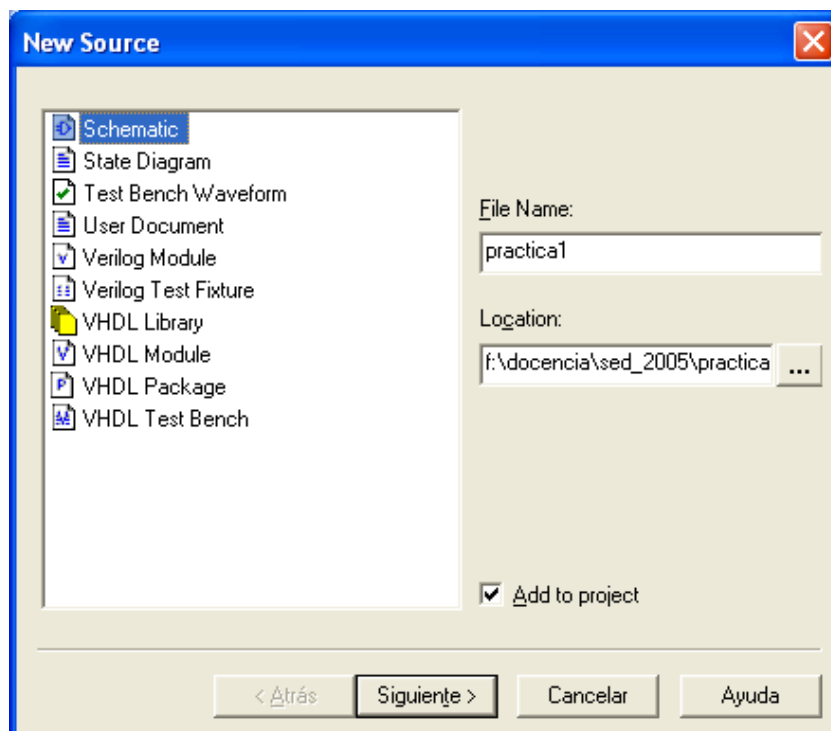


Figura 3.6. Cuadro de diálogo para añadir nuevos ficheros al proyecto.

3.2. Realización del diseño mediante esquemáticos

Una vez inicializado el proyecto haciendo doble clic en el nombre del esquemático de la ventana de fuentes (*Sources in Project*) se abre el editor de esquemáticos de la Figura 3.7.

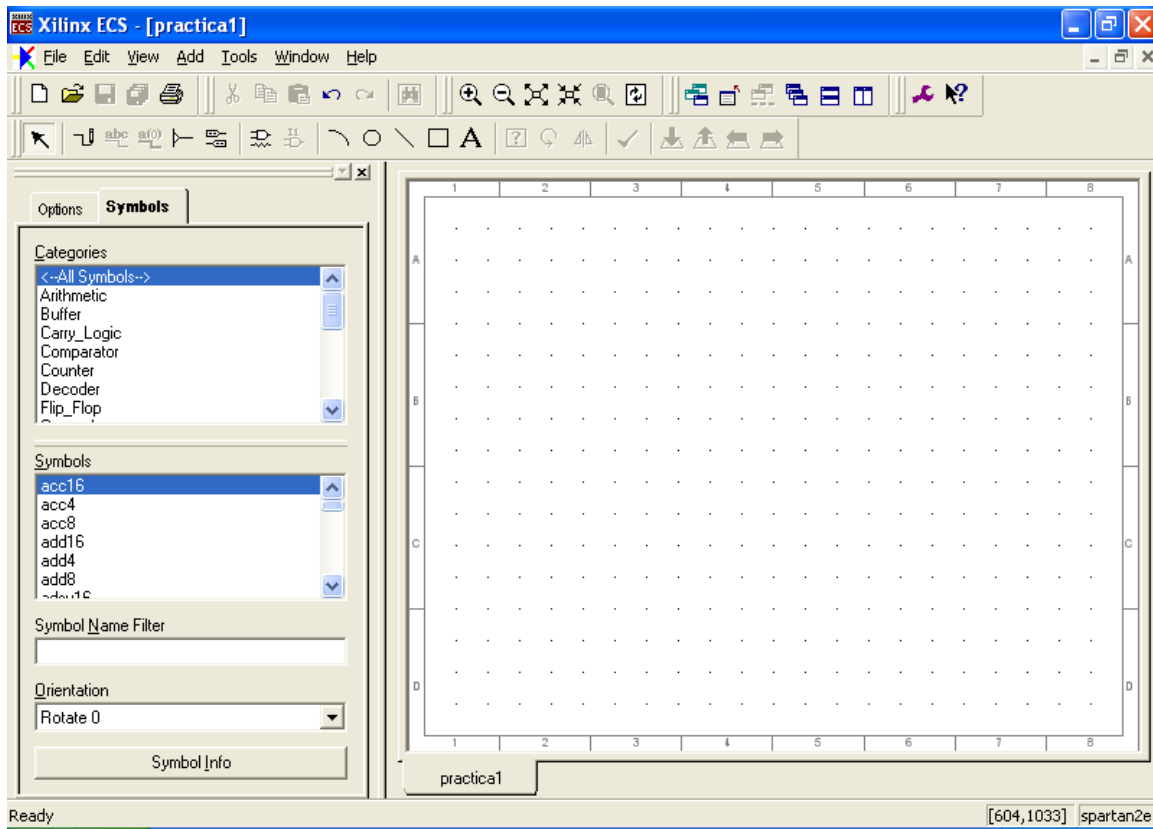


Figura 3.7. Editor de esquemáticos.

En el editor de esquemáticos hay un marco a la izquierda con dos pestañas. En la etiquetada como “*Symbols*” aparecen las bibliotecas de componentes en la ventana “*Categories*”, cuando se selecciona una biblioteca aparece la relación de símbolos que contiene en la ventana inferior.

Para realizar el diseño se van eligiendo los componentes que necesitamos y se pegan en la hoja del esquemático. Para realizar las conexiones de los componentes del esquemático se dispone de las siguientes teclas aceleradoras:



Botón para añadir una conexión.



Botón para etiquetar con un nombre una conexión.



Botón para nombrar un Bus.

Una vez realizadas las conexiones se añaden los puertos de entrada-salida con la opción **Add>IO-Marker** situándolos en los extremos de las conexiones.

Como ejemplo ilustrativo se realiza el esquema correspondiente al sistema secuencial síncrono descrito mediante el grafo de estados de la Figura 3.8. Para implementarlo se utilizan los siguientes componentes:

- 2 biestables J-K (se utilizará el símbolo **FJKRSE** disponible en la biblioteca “Flip_Flop”).
- 4 puertas NOR de 2 entradas (de la biblioteca LOGIC, el símbolo **nor2**).
- 6 puertas inversoras (de la biblioteca LOGIC, el símbolo **inv**).

Las señales que intervienen en el sistema son las siguientes:

- CLK: Señal de reloj que hace evolucionar al sistema.
- A: señal de entrada.
- Reset: señal de puesta inicial del sistema al estado E0
- Q1 y Q0: Señales de salida de los biestables que codifican en binario el estado del sistema (Q0 es la señal de menor peso). Estas señales se agrupan en un bus para visualizar el estado del sistema

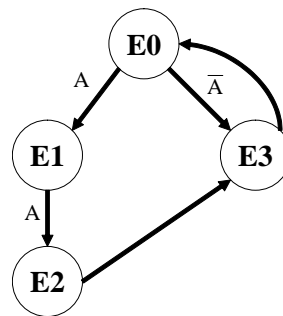


Figura 3.8

El esquemático de este ejemplo debiera resultar algo parecido al de la Figura 3.9. Una vez se tiene el esquemático conviene chequear las reglas de diseño por si hubiese algún error que el programa pueda detectar, para ello ejecutar el comando **Tool>Check Esquematic**.

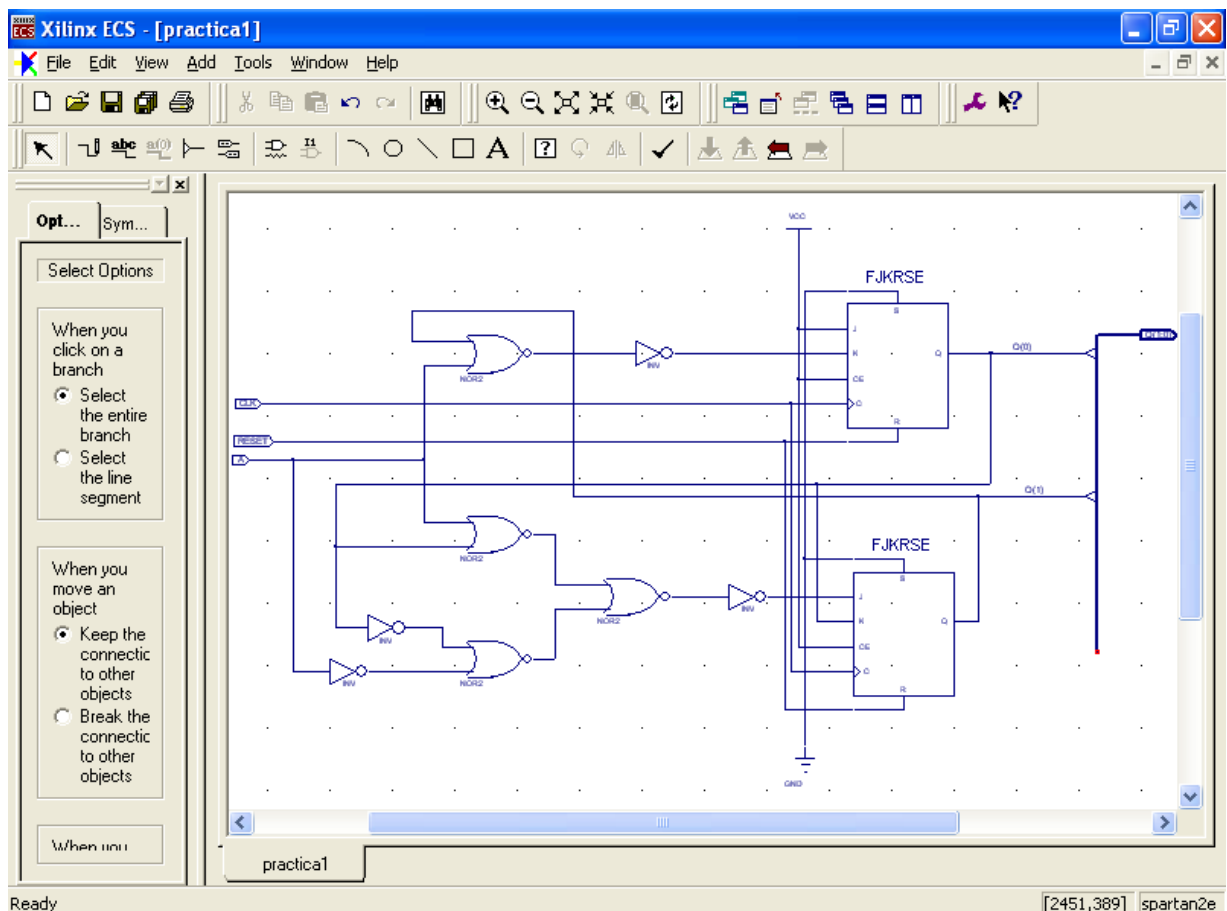


Figura 3.9.

Para realizar el cableado es aconsejable utilizar el comando *Add Wire* pero con la opción *Autorouter* del menú *Options*, de esta forma basta marcar el terminal de inicio y fin de la conexión para que el programa la trace automáticamente. Para las conexiones de los buses utilizar el mismo comando pero activando la opción *Automatically add a bus tap between the bus and the wire*, con esta opción al terminar la conexión en el bus el programa añade automáticamente el conector del bus.

3.3. Verificación del diseño. Vector de pruebas y simulación

Para verificar el diseño es necesario crear estímulos para las entradas. Esto se hace con la herramienta gráfica *Test Bench Waveform*. Para ello primero se añade una nueva fuente al proyecto ejecutando la opción: *Project>Add New Source*. Con esta opción se abrirá el menú de la Figura 3.10 en el que se selecciona el tipo: *Test Bench Waveform*

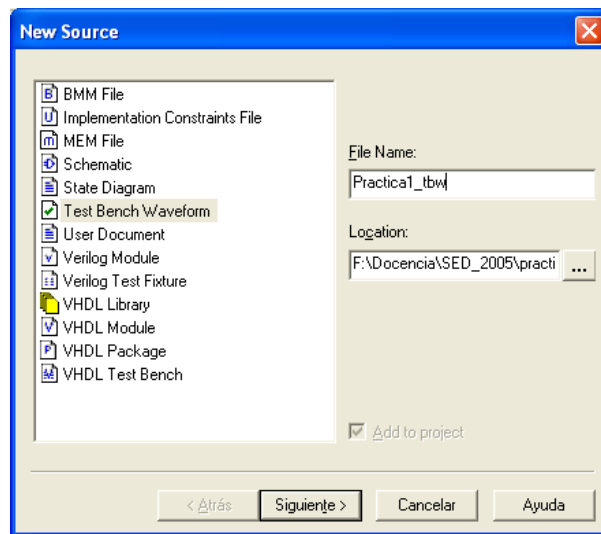


Figura 3.10. Ventana para añadir una nueva fuente de tipo *Test Bench Waveform*.

Una vez creada la nueva fuente el software añade un fichero con extensión *.TBW* al proyecto y se abre el editor en el que aparece una gráfica con las señales conectadas a los puertos de entrada-salida (Figura 3.11). En esta gráfica se crean los estímulos haciendo clic sobre el gráfico de la señal correspondiente. Una vez terminado el fichero de test se simula el diseño mediante el software *ModelSim Simulator* (Figura 3.12).

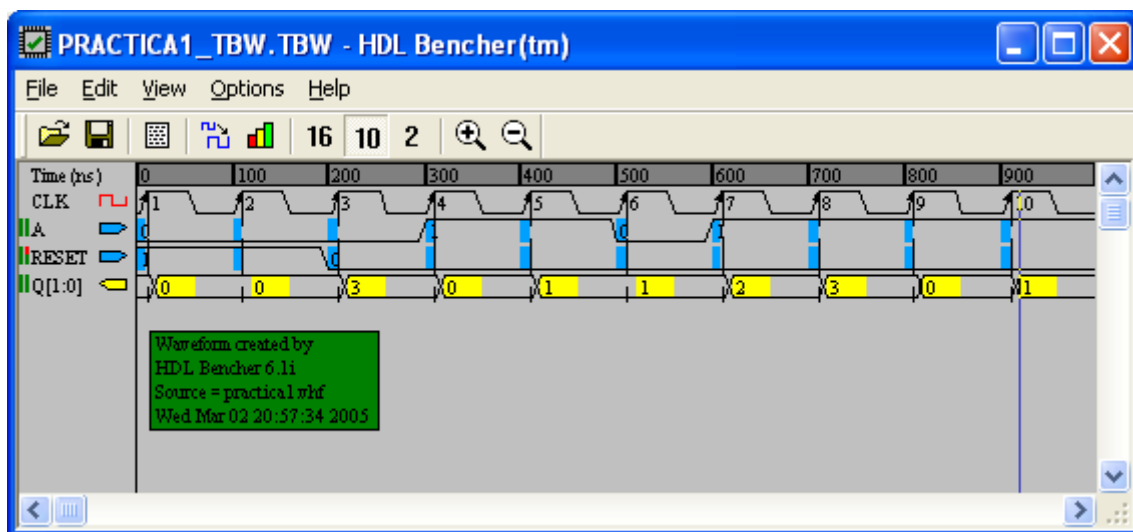
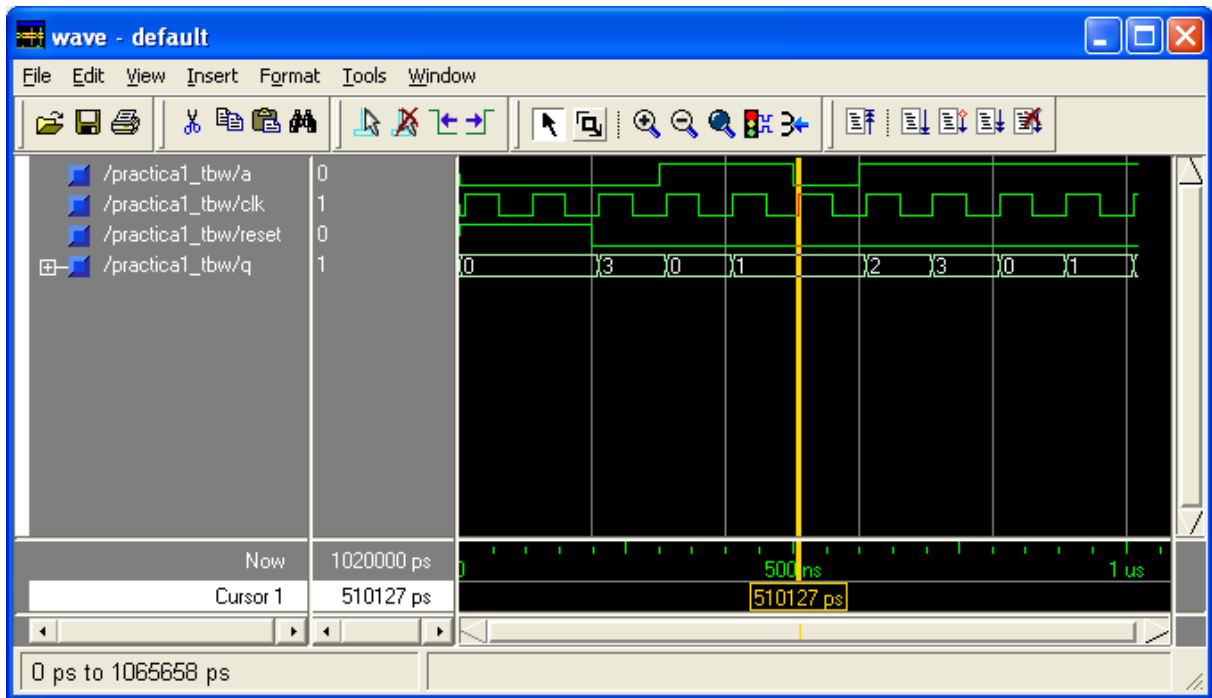


Figura 3.11. Cronograma del *TestBench* para la simulación del comportamiento.


 Figura 3.12. Ventana del cronograma del *ModelSim*.

4. Diseño de circuitos secuenciales mediante grafos de estados.

4.1. Herramienta de edición gráfica de máquinas de estados

El entorno ISE dispone de una potente herramienta denominada *StateCAD* que permite realizar un diseño de forma gráfica. Para ilustrar el uso de esta herramienta se repite el ejemplo del apartado anterior. Se crea un nuevo proyecto vacío y añadimos una nueva fuente del tipo *State Diagram*, Figura 4.1.

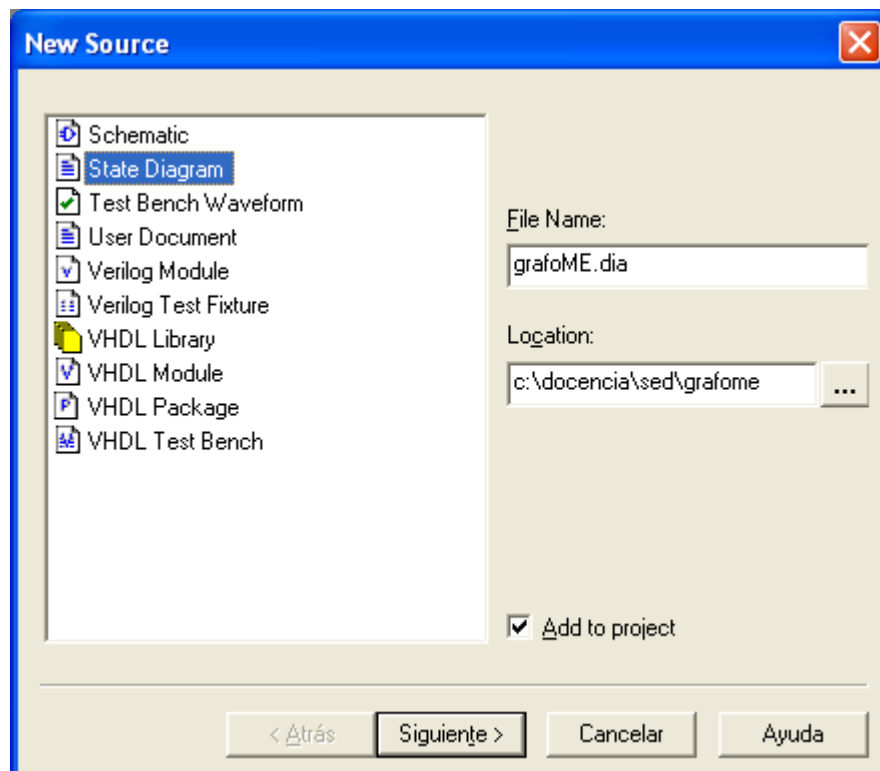


Figura 4.1.

A continuación aparece el cuadro de diálogo de la Figura 4.2. Comprobar que el nombre del archivo es correcto y que tiene la extensión DIA. Una vez creado el archivo se abre el editor gráfico. Ahora se realiza el diseño del grafo que deberá quedar como se muestra en la Figura 4.3.

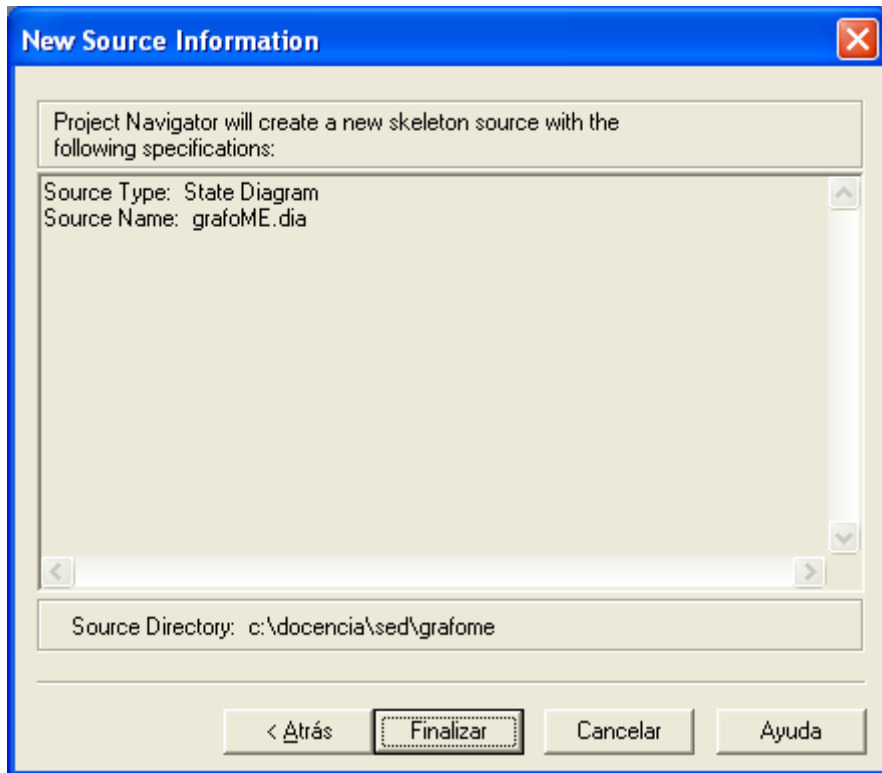


Figura 4.2

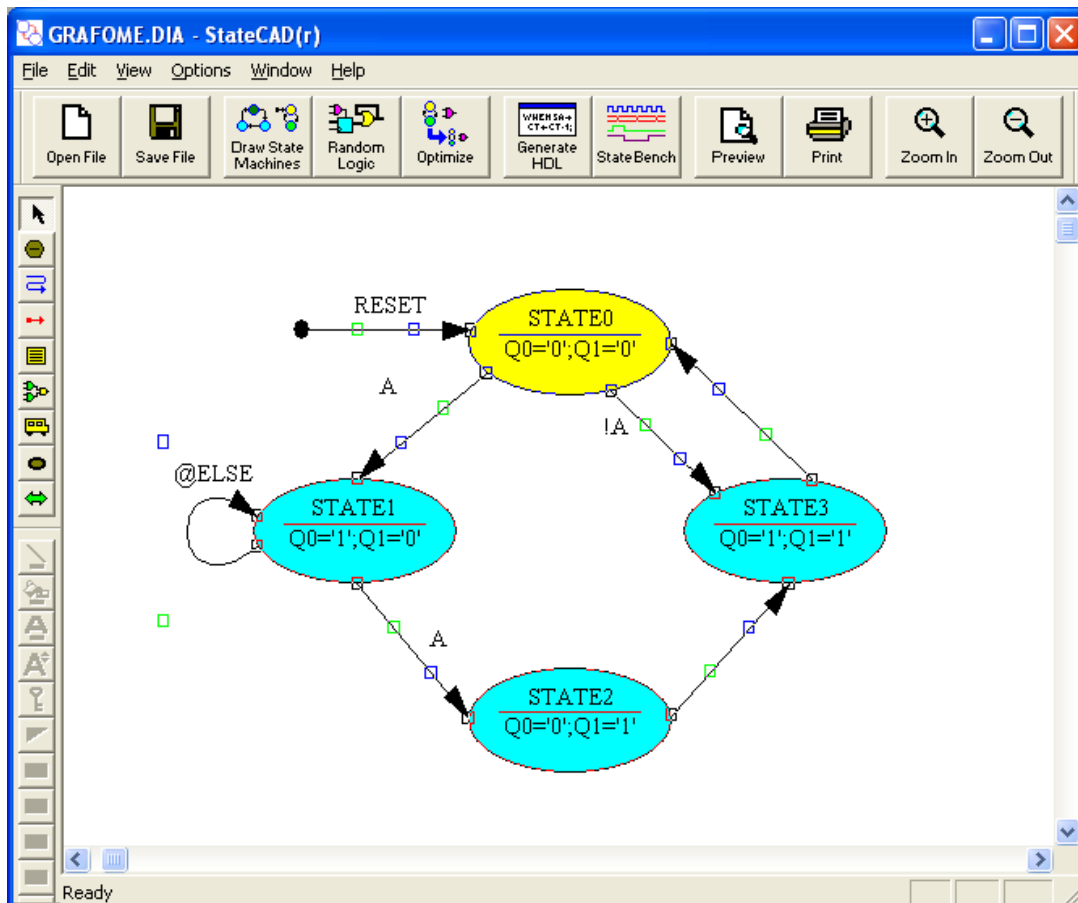


Figura 4.3.

Para dibujar el grafo se utilizan los siguientes botones:



Añadir estado



Añadir transición



Añadir reset



Asistente para crear grafos de estado

Para añadir o cambiar la condición lógica de una transición se hace doble clic sobre ella y aparece el cuadro de la Figura 4.4. Para asignar los valores de las salidas se hace doble clic sobre el estado y se escriben los valores que toman las salidas en dicho estado (Figura 4.5).

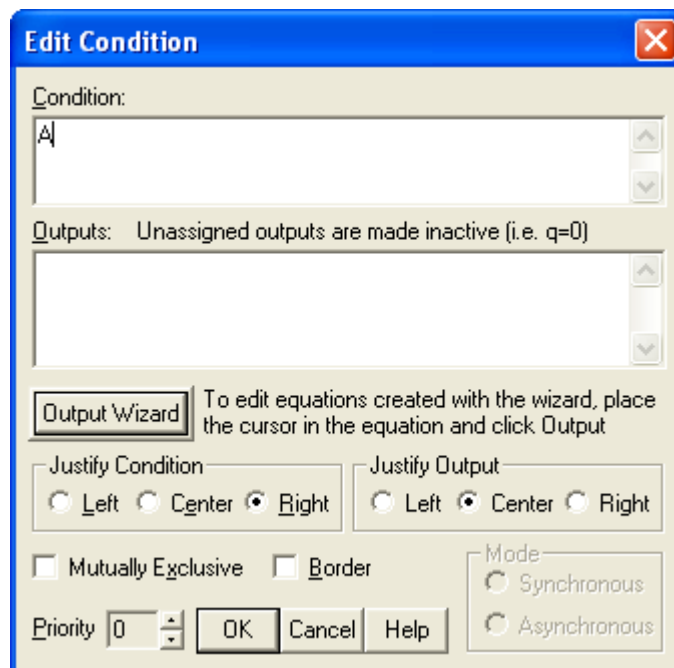


Figura 4.4

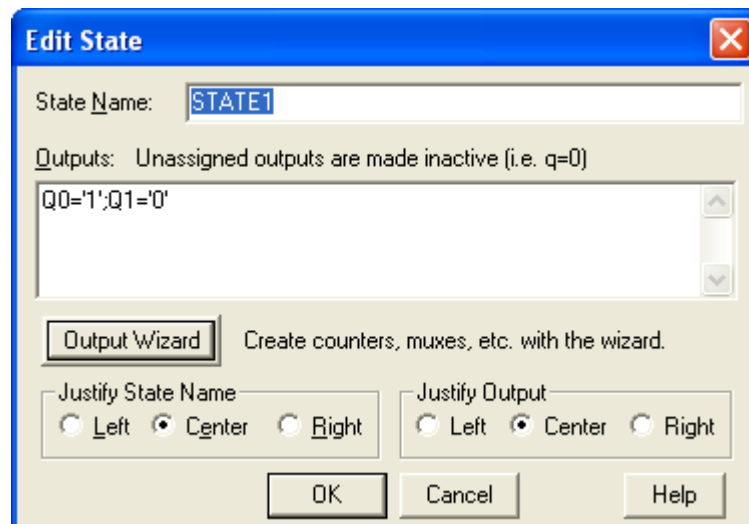
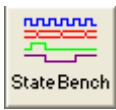


Figura 4.5

Una vez hecho el gráfico se simula con el programa *StateBench* el cual se ejecuta con los siguientes botones:



Botón de llamada al programa *StateBench*.



Botón para ejecutar el *StateBench* de forma automática.

En la Figura 4.6 se muestra el resultado de la simulación.

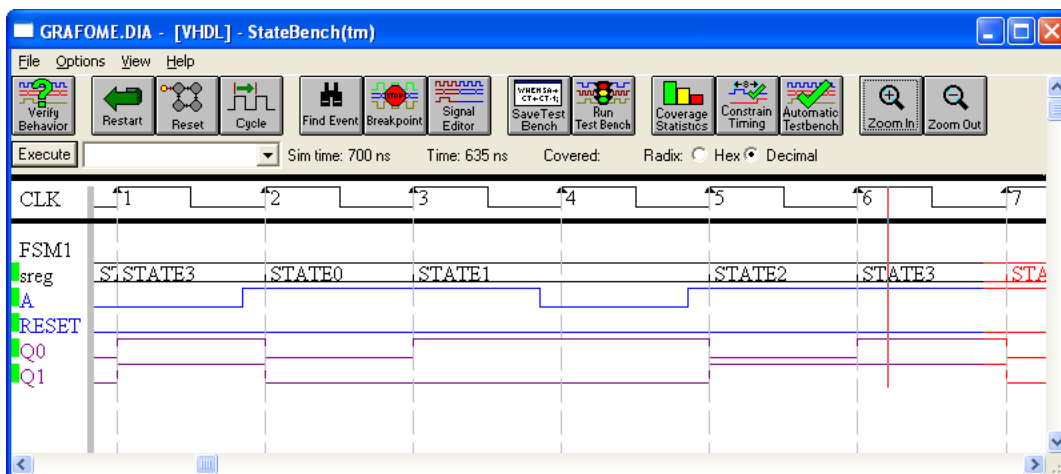


Figura 4.6. Simulación del comportamiento del grafo de estados.

Por último se crea el código VHDL de forma automática mediante el botón:



Botón para crear el código VHDL correspondiente al diseño.

El resultado aparece en un cuadro como el de la Figura 4.7.

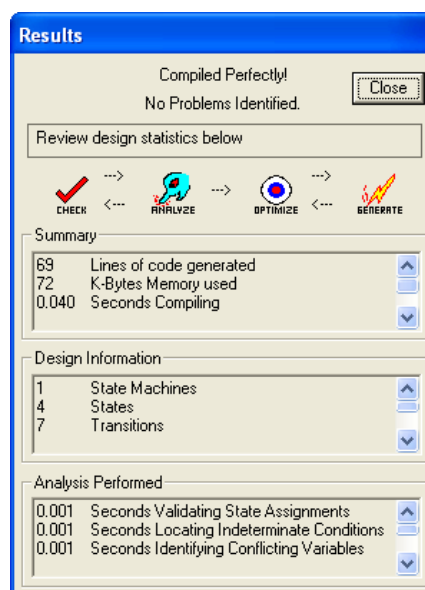


Figura 4.7

4.2. Incorporación al proyecto de los archivos VHDL y del vector de pruebas.

Una vez creado el código VHDL de la máquina de estados se incluye el archivo VHDL en el proyecto y se genera su símbolo. A continuación se crea un nuevo archivo fuente de tipo esquemático y en él se introduce el esquema utilizando el símbolo creado y se añaden los terminales de entrada y salida (Figura 4.8). Una vez terminado se guarda el esquema y se vuelve al gestor de proyectos.

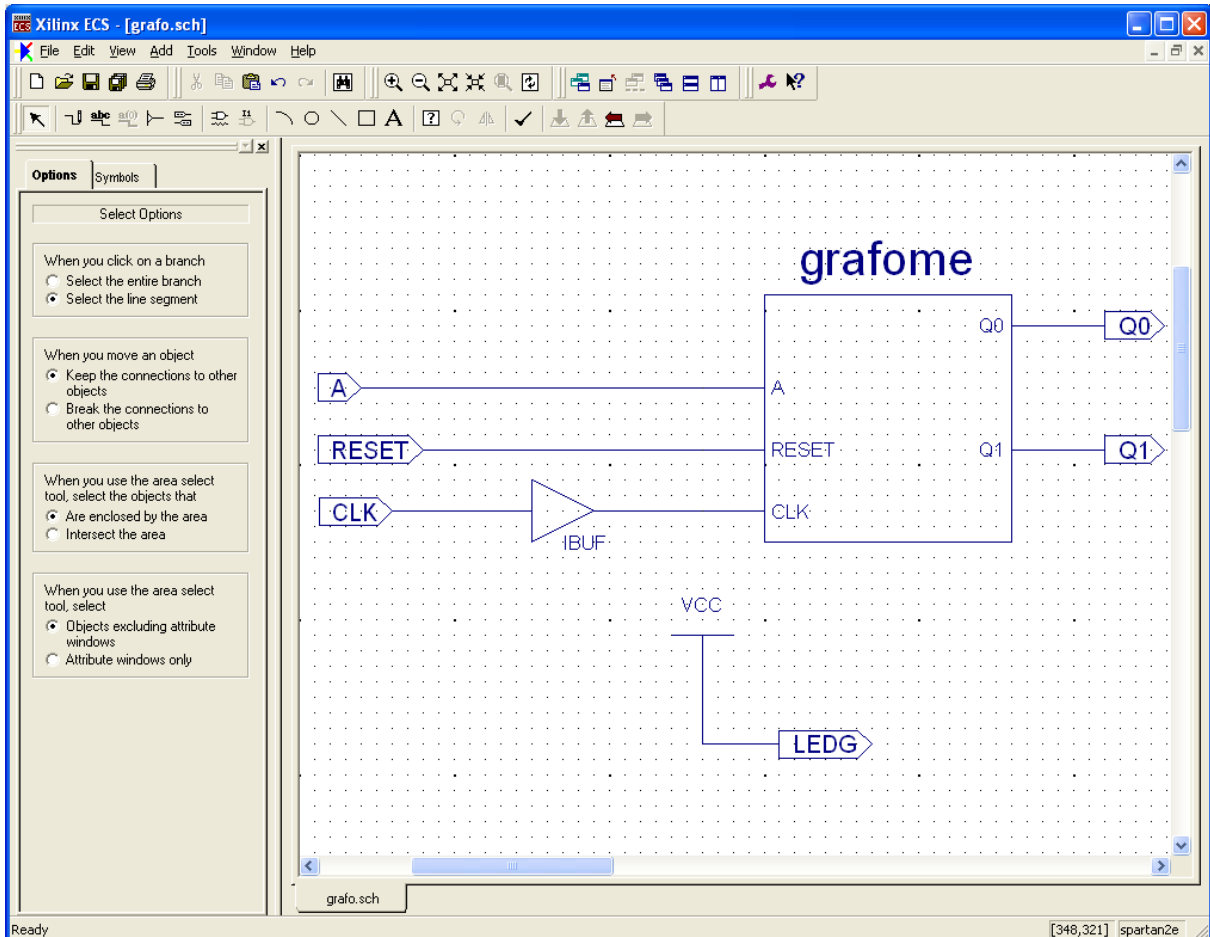


Figura 4.8.

5. Programación de la FPGA.

5.1. Asignación de entradas y salidas

Ahora hay que asignar los pines de la FPGA que correspondan con los periféricos. Como se está utilizando la placa de desarrollo Digilab 2SB y la de periféricos Digilab Digital I/O 4, se asignan los terminales según la Tabla 5.1.

Tabla 5.1.

CLK	BTN1	P3
RESET	SW1	P23
A	SW2	P21
Q0	LED1	P111
Q1	LED2	P109
VCC	LEDG	P45

Para configurar estos valores se crea otro fichero fuente de tipo *UCF* que es un fichero de restricciones donde se configuran los terminales de la FPGA (Figura 5.1).

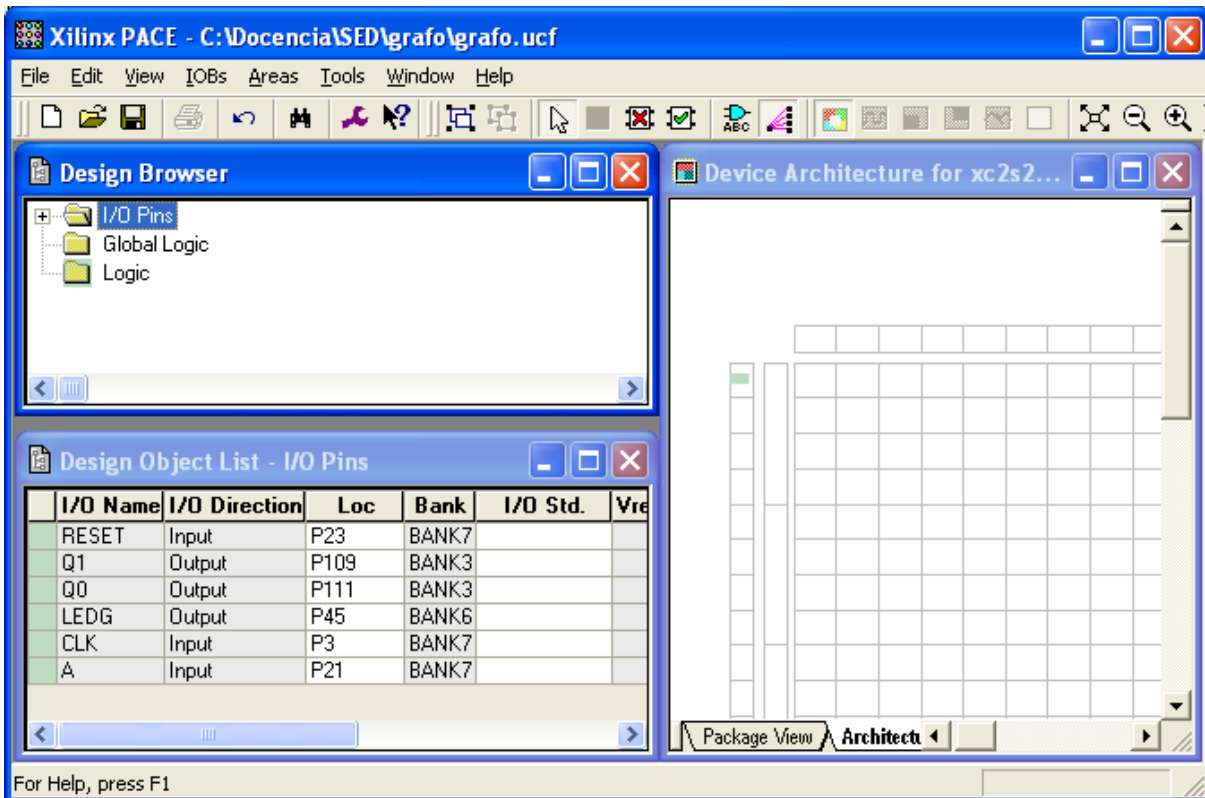


Figura 5.1.

Después de comprobar que el comportamiento del diseño es el esperado se puede hacer una simulación temporal con la opción *Simulate Post-Place & Route VHDL Model* con el ModeSim (Figura 5.2). De esta forma se puede estimar cómo se comportará el sistema una vez se halla configurado la FPGA. En la Figura 5.3 se muestra el contenido del cuadro de gestión del proyecto.

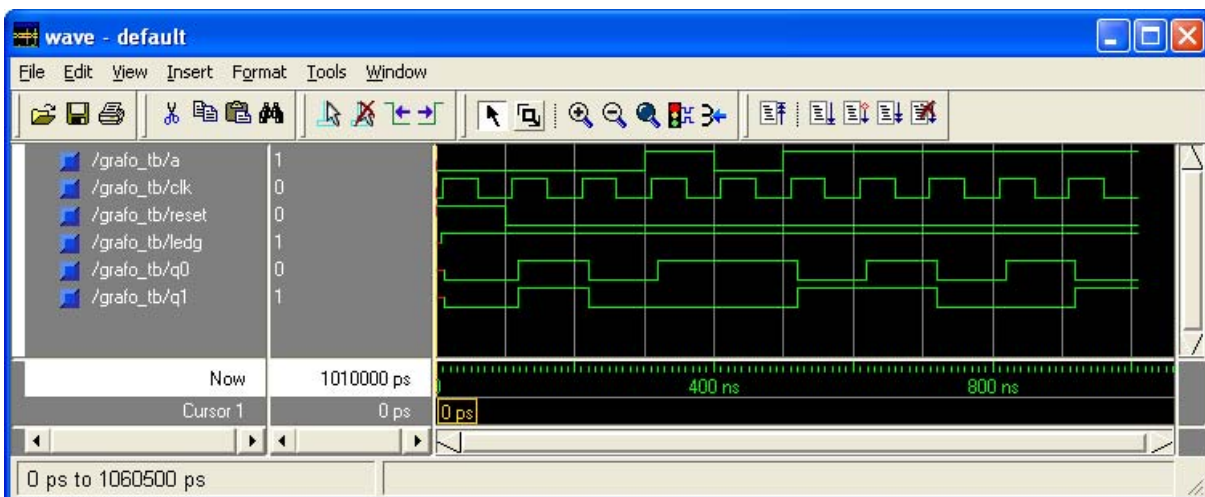


Figura 5.2.

5.2. Configuración de la FPGA

Para configurar la FPGA primero se genera el fichero de programación con la opción *Generate Programming File* y después se configura el dispositivo con la opción *Configure Device (iMPACT)* (Figura 5.3).

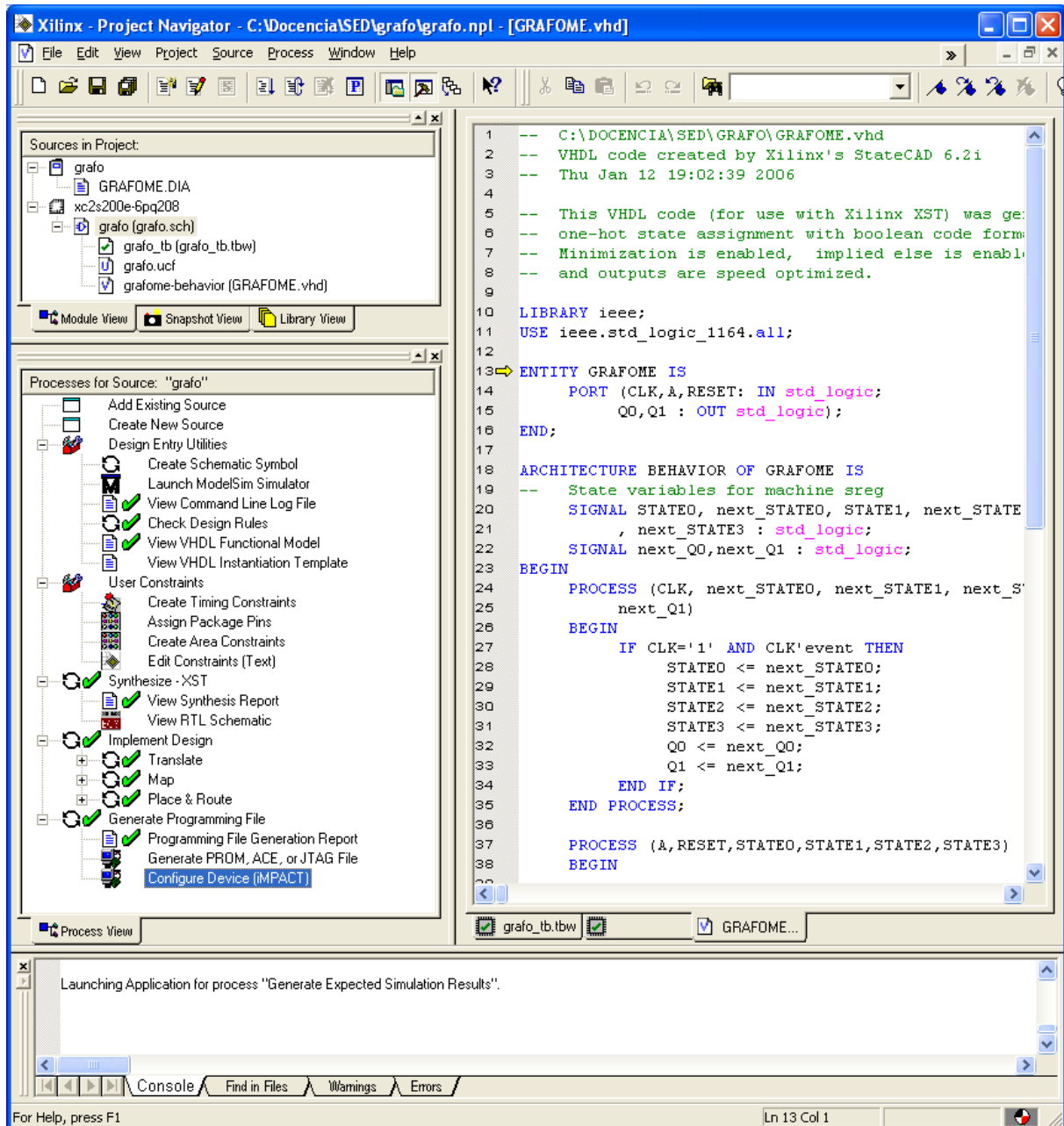


Figura 5.3

6. Diseño en VHDL de un decodificador BCD/7 segmentos

Para terminar el ejercicio se hace un último cambio en el diseño, se trata de añadir un combinacional de salida que cambie la codificación del estado en binario que se hace con los bits Q0Q1 por una codificación en decimal para el display de 7 segmentos. Para ello se crea un nuevo recurso del tipo fichero de VHDL y en él se describe el bloque decodificador (Tabla 6.1). Este código tiene dos partes, una es la definición de la entidad en la cual se declaran los puertos que tiene el bloque decodificador, en la otra se define el comportamiento mediante un proceso. Este proceso tiene como lista de sensibilidad el dato de entrada, y la descodificación se hace mediante una sentencia de selección múltiple.

Tabla 6.1. Descripción de la arquitectura y del comportamiento del decodificador.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity dec7seg is
    Port ( DIN : std_logic_vector(2 downto 0);
          a : out std_logic;
          b : out std_logic;
          c : out std_logic;
          d : out std_logic;
          e : out std_logic;
          f : out std_logic;
          g : out std_logic);
end dec7seg;

architecture Behavioral of dec7seg is
begin
    process(DIN)
    begin
        case DIN is
            when "000" =>
                a<='1';b<='1';c<='1';d<='1';e<='1';f<='1';g<='0';
            when "001" =>
                a<='0';b<='1';c<='1';d<='0';e<='0';f<='0';g<='0';
            when "010" =>
                a<='1';b<='1';c<='0';d<='1';e<='1';f<='0';g<='1';
            when "011" =>
                a<='1';b<='1';c<='1';d<='1';e<='0';f<='0';g<='1';
            when "100" =>
                a<='0';b<='1';c<='1';d<='0';e<='0';f<='1';g<='1';
            when "101" =>
                a<='1';b<='0';c<='1';d<='1';e<='0';f<='1';g<='1';
            when "110" =>
                a<='1';b<='0';c<='1';d<='1';e<='1';f<='1';g<='1';
            when "111" =>
                a<='1';b<='1';c<='1';d<='0';e<='0';f<='0';g<='0';
            when others => null;
        end case;
    end process;
end Behavioral;
```

Con el diseño del decodificador se crea un bloque y se añade al esquemático. Para el correcto funcionamiento del display es necesario invertir las salidas del decodificador ya que los segmentos del display son activos con un '0', ya que las señales van al cátodo de los LEDs.

Para integrar correctamente el nuevo bloque se debe eliminar el actual fichero *UCF* de restricciones y crear otro que contenga la nueva distribución de terminales de entrada-salida según la Tabla 6.2. En la Figura 6.1 se muestra el diseño completo que incluye el nuevo bloque decodificador.

Tabla 6.2. Asignación de terminales de la FPGA.

CLK	BTN1	P3
RESET	SW1	P23
A	SW2	P21
aa		P22
bb		p20
cc		p17
dd		P15
ee		P10
ff		P8
gg		P6

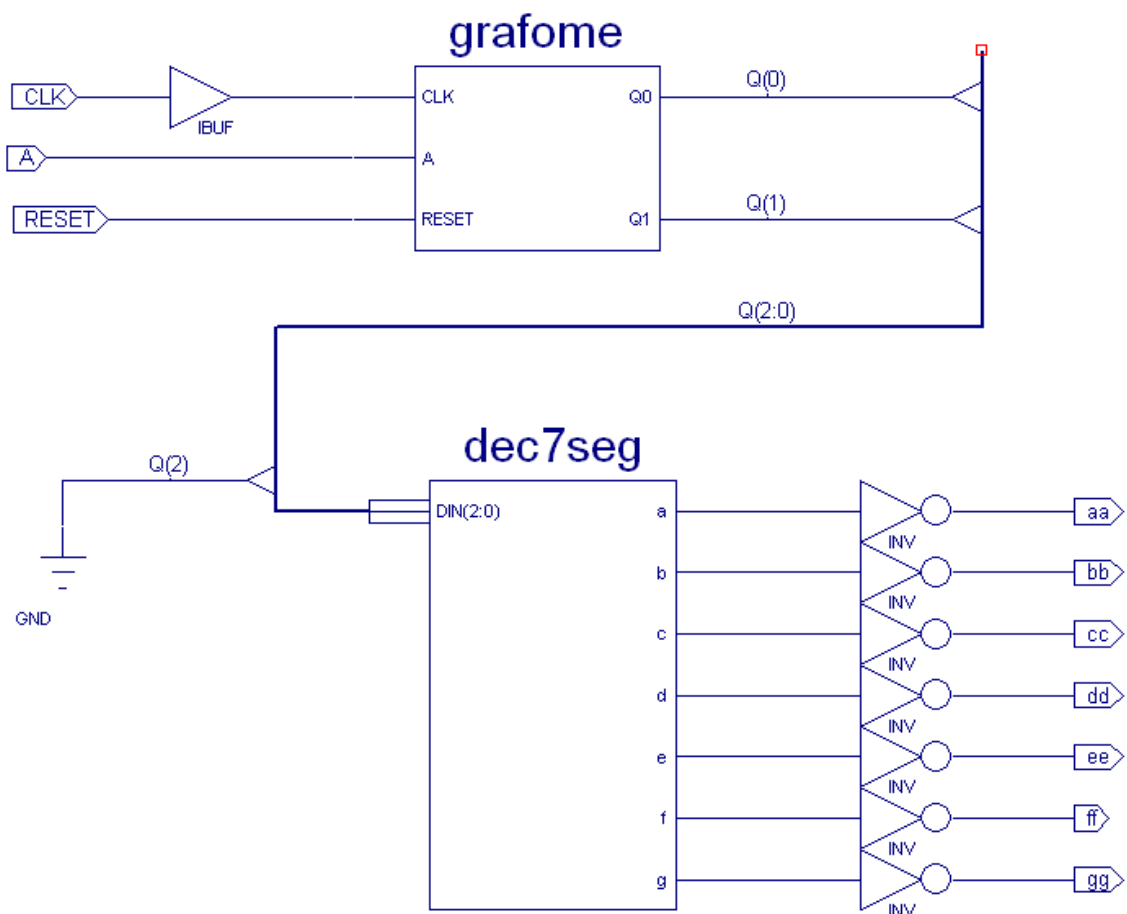


Figura 6.1. Incorporación del módulo decodificador al ejemplo de la máquina de estados.