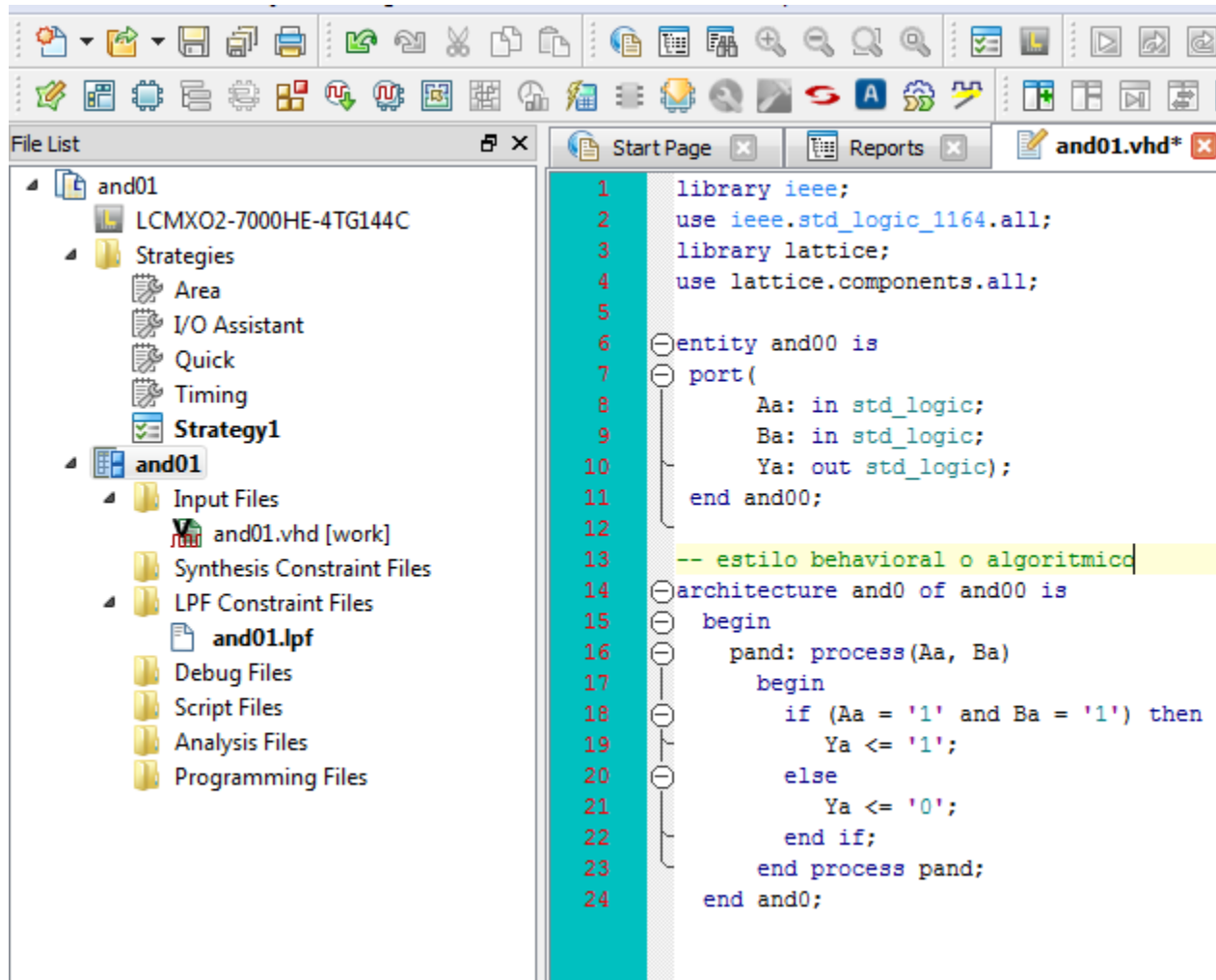


CONTINUAMOS

Iniciando con un proyecto sencillo. La puerta AND.

21.- En este caso la puerta AND se programará mediante el estilo behavioral, para lo cual se hace necesario declarar un “process”. Dentro del process se declara la lógica de la puerta AND. [Aquí nos quedamos en las diapositivas anteriores](#)

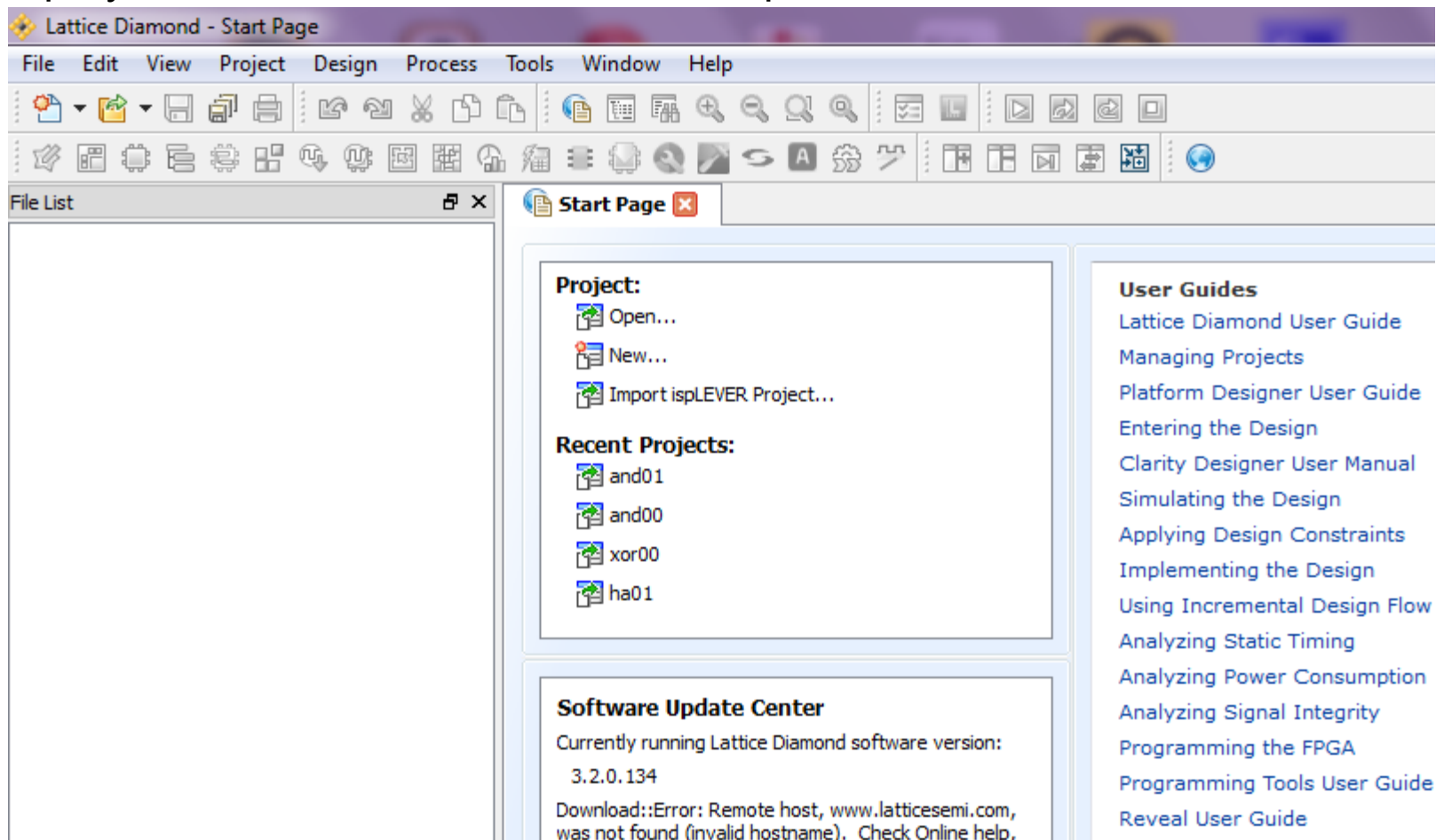


The screenshot displays the Lattice IDE interface. On the left, the 'File List' pane shows a project named 'and01' containing various files like 'LCMXO2-7000HE-4TG144C', 'Strategies', and 'and01.vhd [work]'. The main editor window shows the VHDL code for an AND gate, with line numbers 1 through 24 on the left margin. The code includes library declarations, an entity definition, and a behavioral architecture using a process block.

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  library lattice;
4  use lattice.components.all;
5
6  entity and00 is
7  port(
8      Aa: in std_logic;
9      Ba: in std_logic;
10     Ya: out std_logic);
11 end and00;
12
13 -- estilo behavioral o algoritmico
14 architecture and0 of and00 is
15 begin
16     pand: process(Aa, Ba)
17     begin
18         if (Aa = '1' and Ba = '1') then
19             Ya <= '1';
20         else
21             Ya <= '0';
22         end if;
23     end process pand;
24 end and0;
```

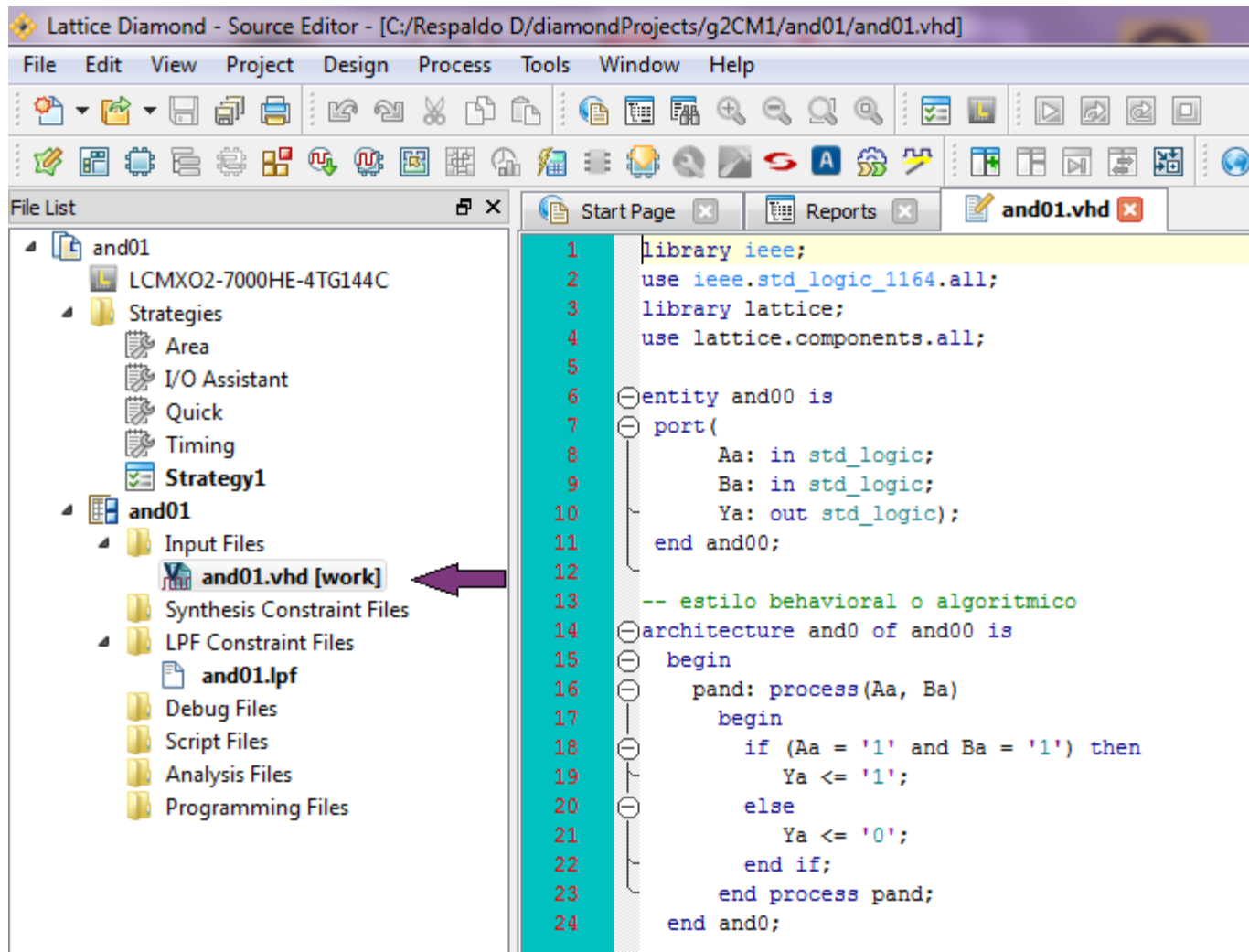
Iniciando con un proyecto sencillo. La puerta AND.

22.- Al abrir el ambiente de desarrollo “Diamond”, después de haber estado trabajando en algún proyecto, aparecerá como se muestra en la siguiente pantalla. En la ventana “Start Page” se ven los proyectos en los que hemos estado trabajando. Selecciones el que deseen, en este caso, como estamos trabajando con el proyecto “and01”, seleccionamos tal opción.



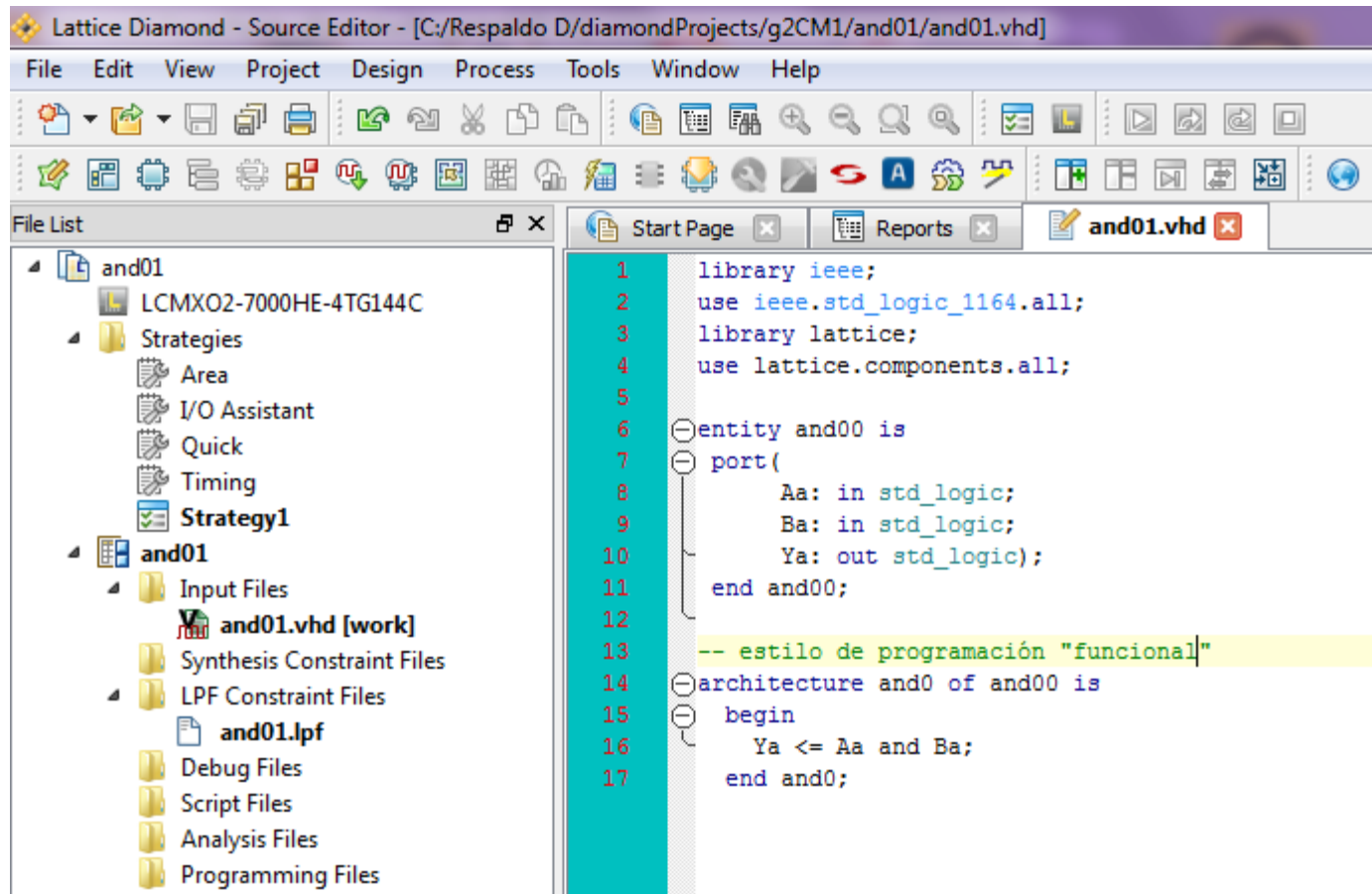
Iniciando con un proyecto sencillo. La puerta AND.

23.- Hacer doble clic en “and01.vhd [work]” indicado con la flecha color púrpura, para tener acceso al código vhd.



Iniciando con un proyecto sencillo. La puerta AND.

24.- Ahora les mostraré, en el mismo proyecto, y en el mismo código, el estilo de programación “funcional” para generar la puerta AND.



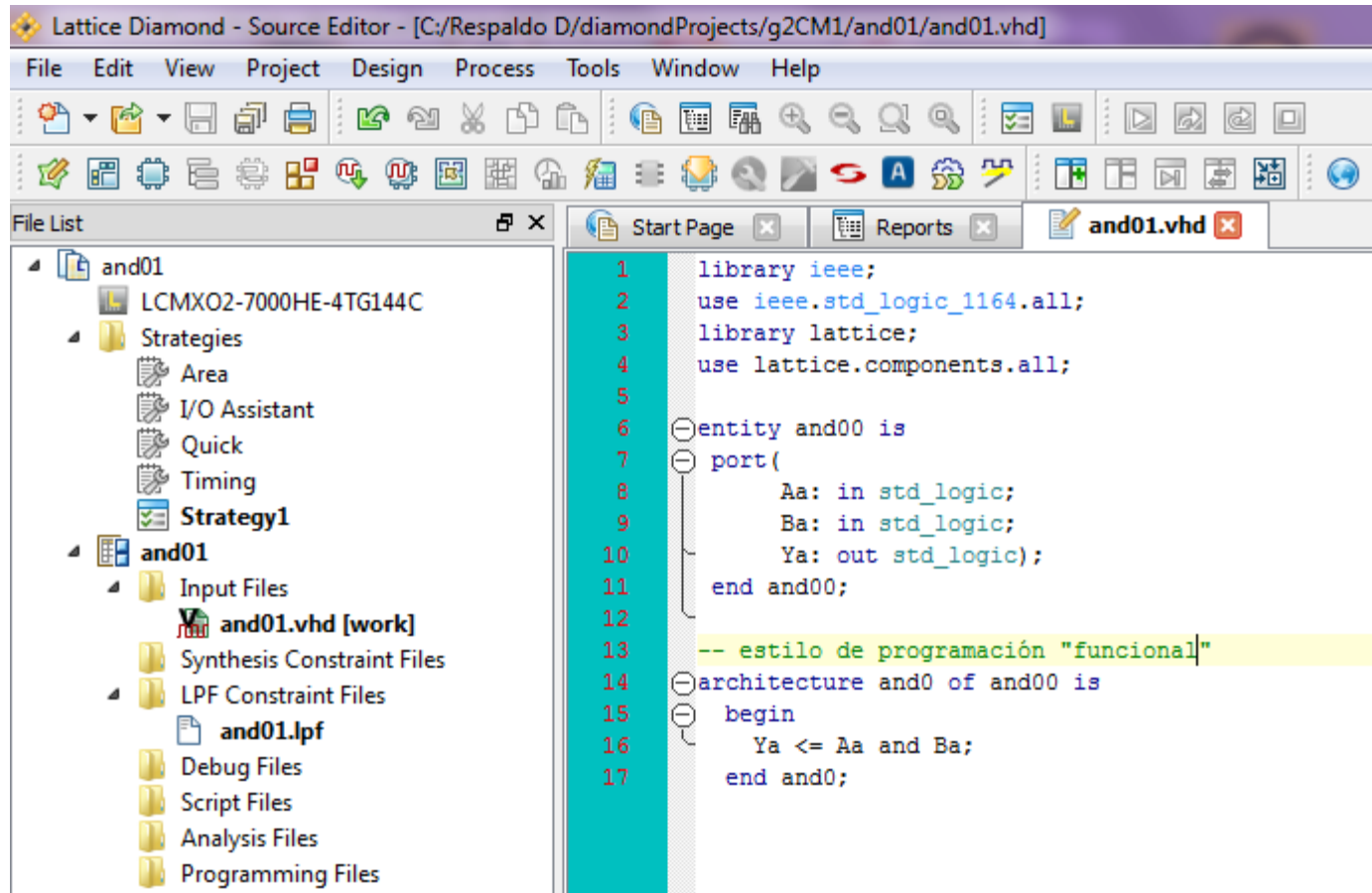
The screenshot shows the Lattice Diamond Source Editor interface. The title bar indicates the file path: [C:/Respaldo D/diamondProjects/g2CM1/and01/and01.vhd]. The menu bar includes File, Edit, View, Project, Design, Process, Tools, Window, and Help. Below the menu is a toolbar with various icons for file operations, design, and simulation. The File List pane on the left shows a project structure for 'and01' under 'LCMXO2-7000HE-4TG144C'. It includes folders for 'Strategies', 'and01', 'Input Files', 'Synthesis Constraint Files', 'LPF Constraint Files', 'Debug Files', 'Script Files', 'Analysis Files', and 'Programming Files'. The 'and01' folder is expanded, showing 'and01.vhd [work]'. The main editor window displays the VHDL code for 'and01.vhd'. The code is as follows:

```
1 library ieee;
2 use ieee.std_logic_1164.all;
3 library lattice;
4 use lattice.components.all;
5
6 entity and00 is
7     port(
8         Aa: in std_logic;
9         Ba: in std_logic;
10        Ya: out std_logic);
11 end and00;
12
13 -- estilo de programación "funcional"
14 architecture and0 of and00 is
15     begin
16         Ya <= Aa and Ba;
17     end and0;
```

Observen que no se necesita declarar process, y en este caso el programa consta de una solo línea. Dicho estilo de programación es eficiente sólo para circuitos lógicos combinacionales.

Iniciando con un proyecto sencillo. La puerta AND.

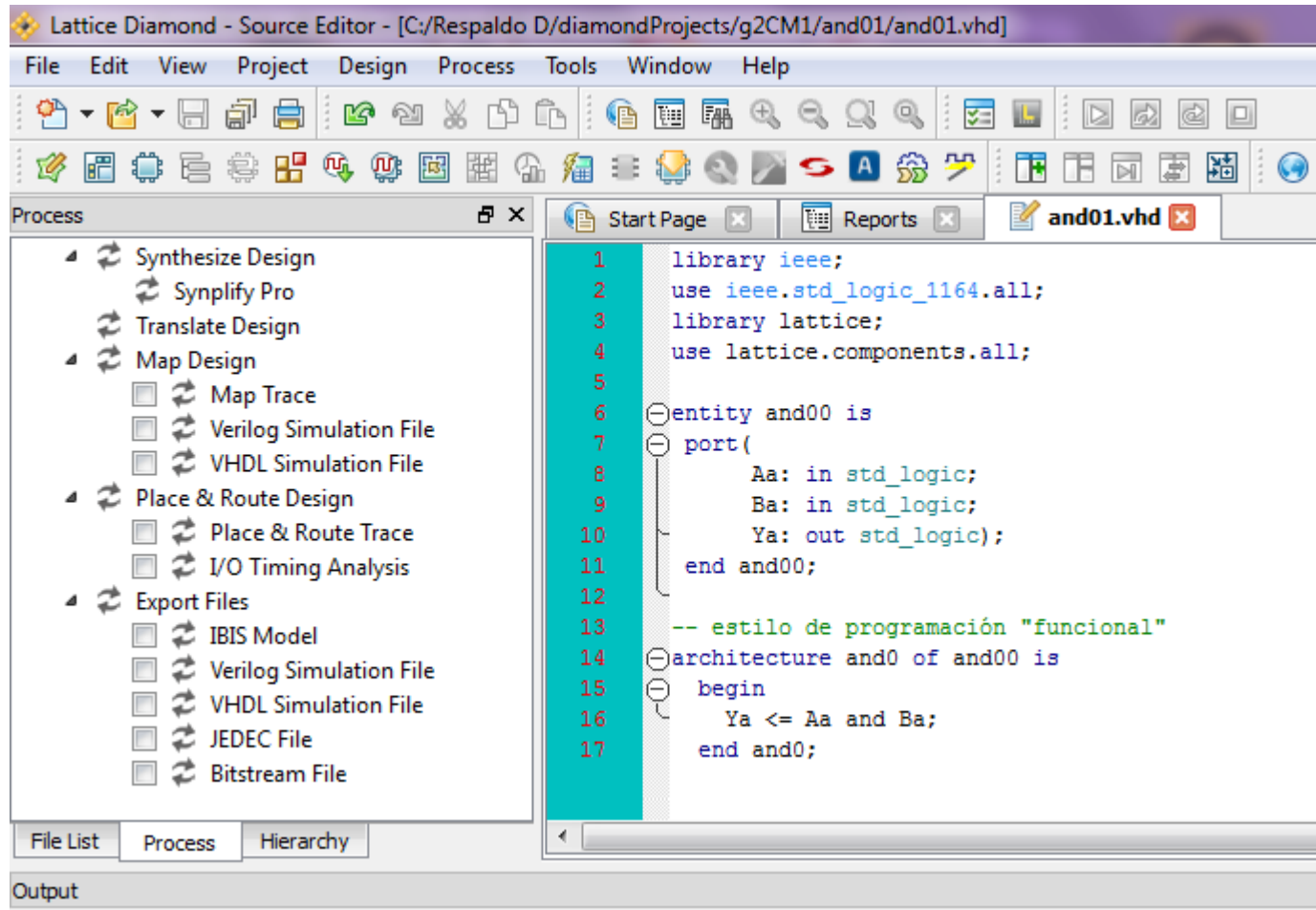
24.- Ahora les mostraré, en el mismo proyecto, y en el mismo código, el estilo de programación “funcional” para generar la puerta AND.



Si gustan, pueden dejarlo de esta forma, o como se muestra en la pantalla 23. Sin embargo a la hora de calificarlo, les pediré que lo compilen e implementen de ambas formas.

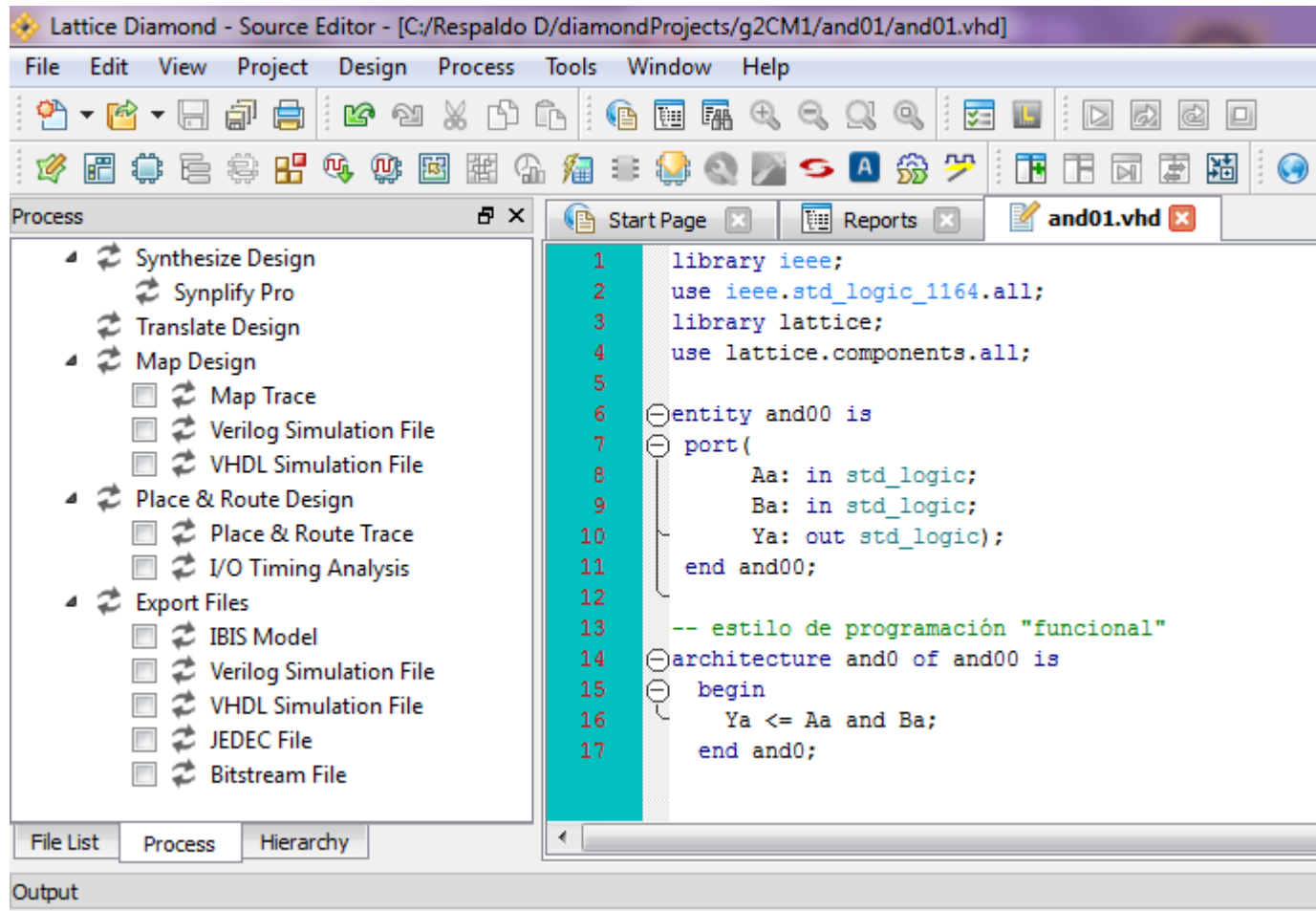
Iniciando con un proyecto sencillo. La puerta AND.

25.- COMPILACIÓN DEL PROYECTO: Ahora vayamos a la ventana de procesos (la de más a la izquierda de la pantalla global, en el ambiente). Seleccionamos la pestaña “Process” como se muestra en la siguiente figura.



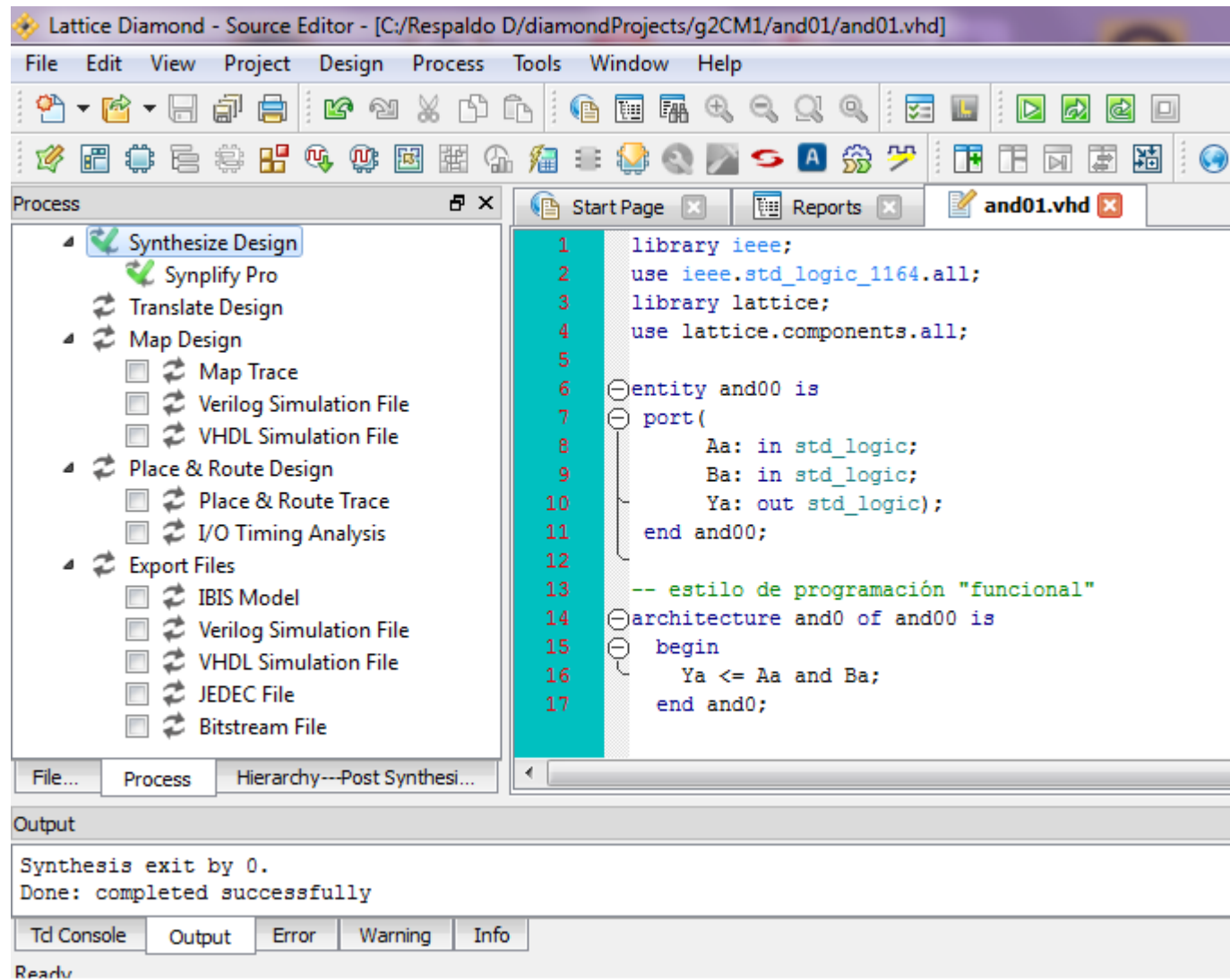
Iniciando con un proyecto sencillo. La puerta AND.

26.- Primeramente hacemos doble clic en “Synthesize Design”



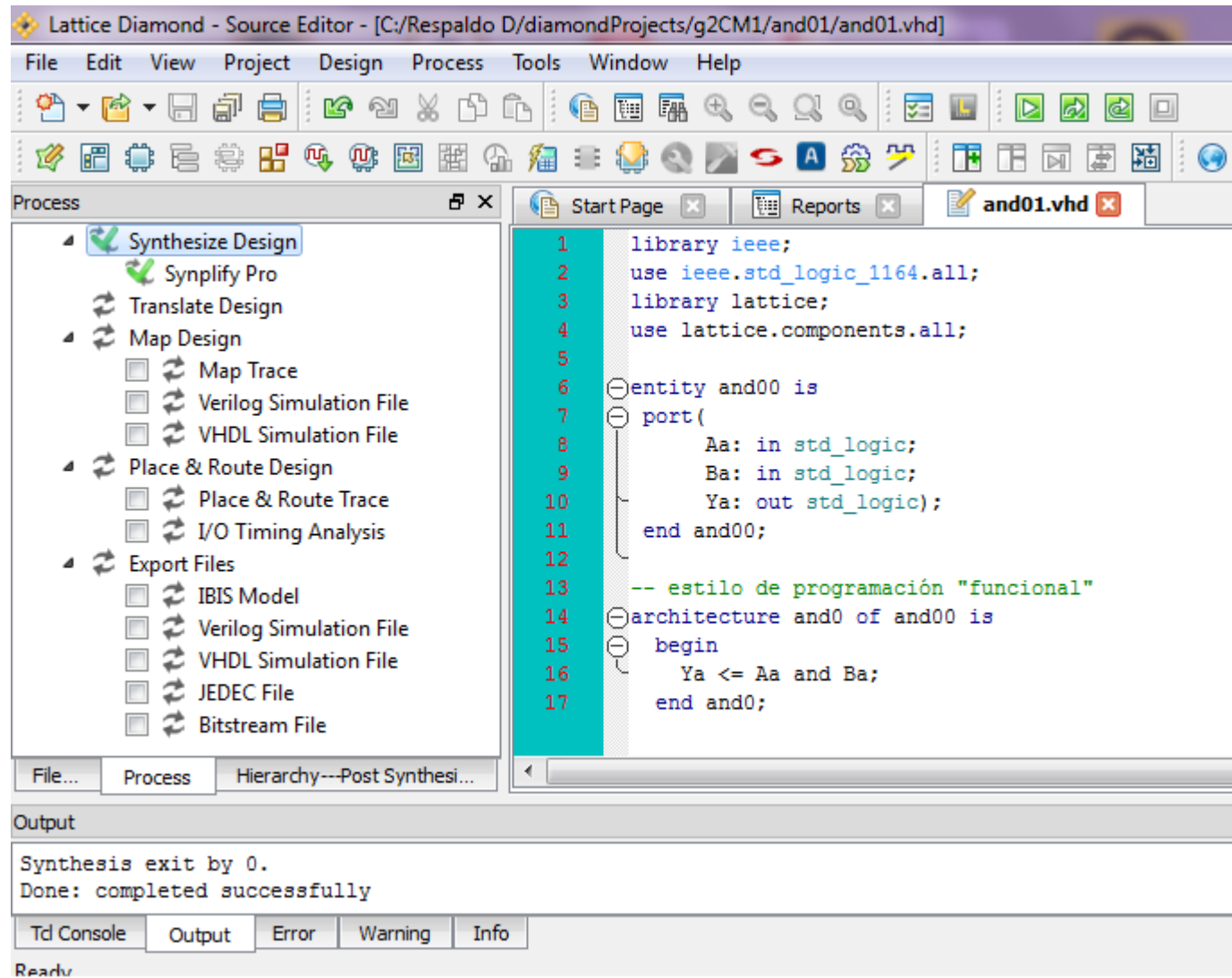
Iniciando con un proyecto sencillo. La puerta AND.

27.- Si todo se ha hecho bien, se mostrará una pantalla como esta



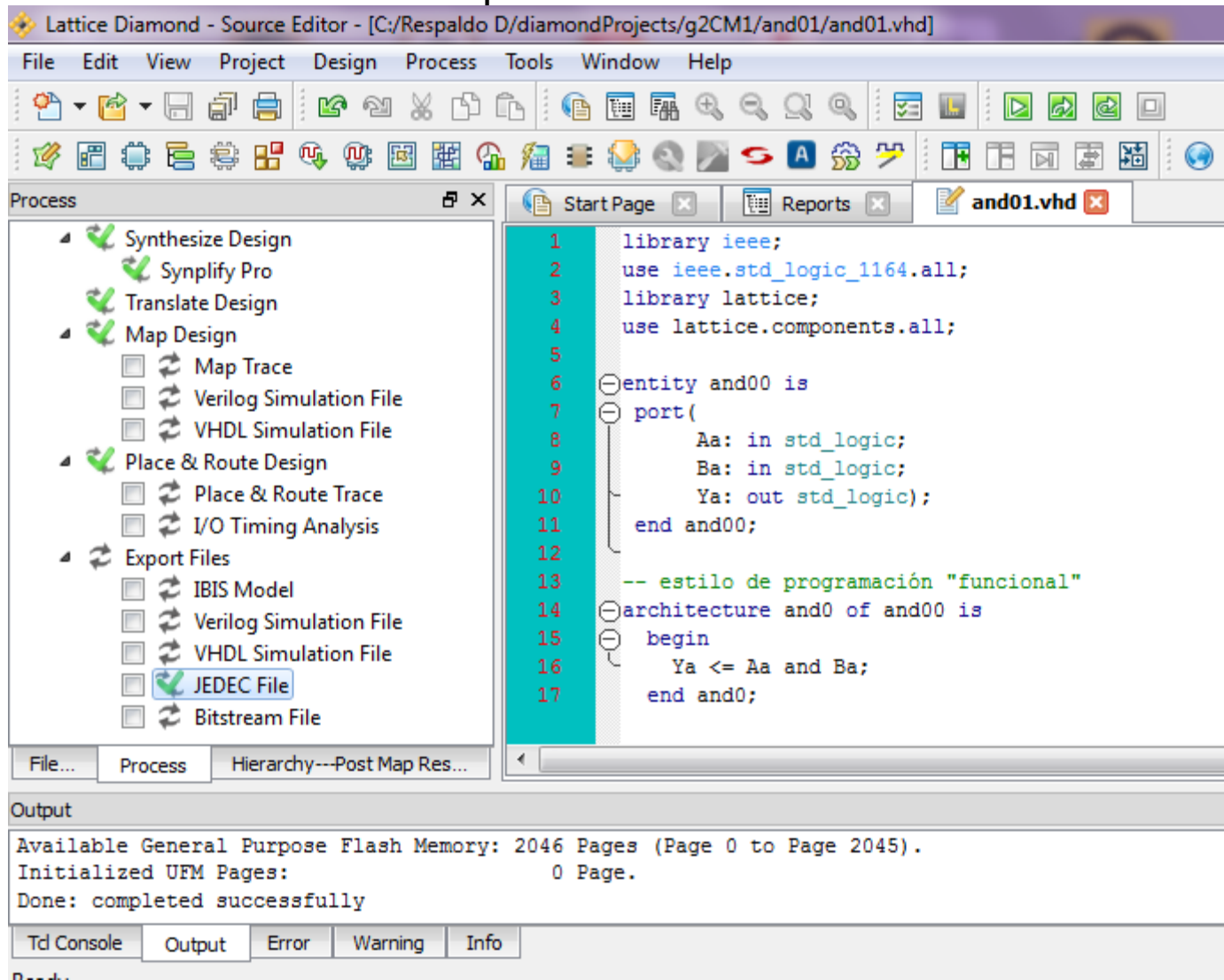
Iniciando con un proyecto sencillo. La puerta AND.

27.- Los procesos “Synthesize Design” y “Synplify Pro” aparecen con marcadores de aceptación color verde. El compilador ha realizado verificación de sintaxis y también ha reservado recursos para el proyecto.



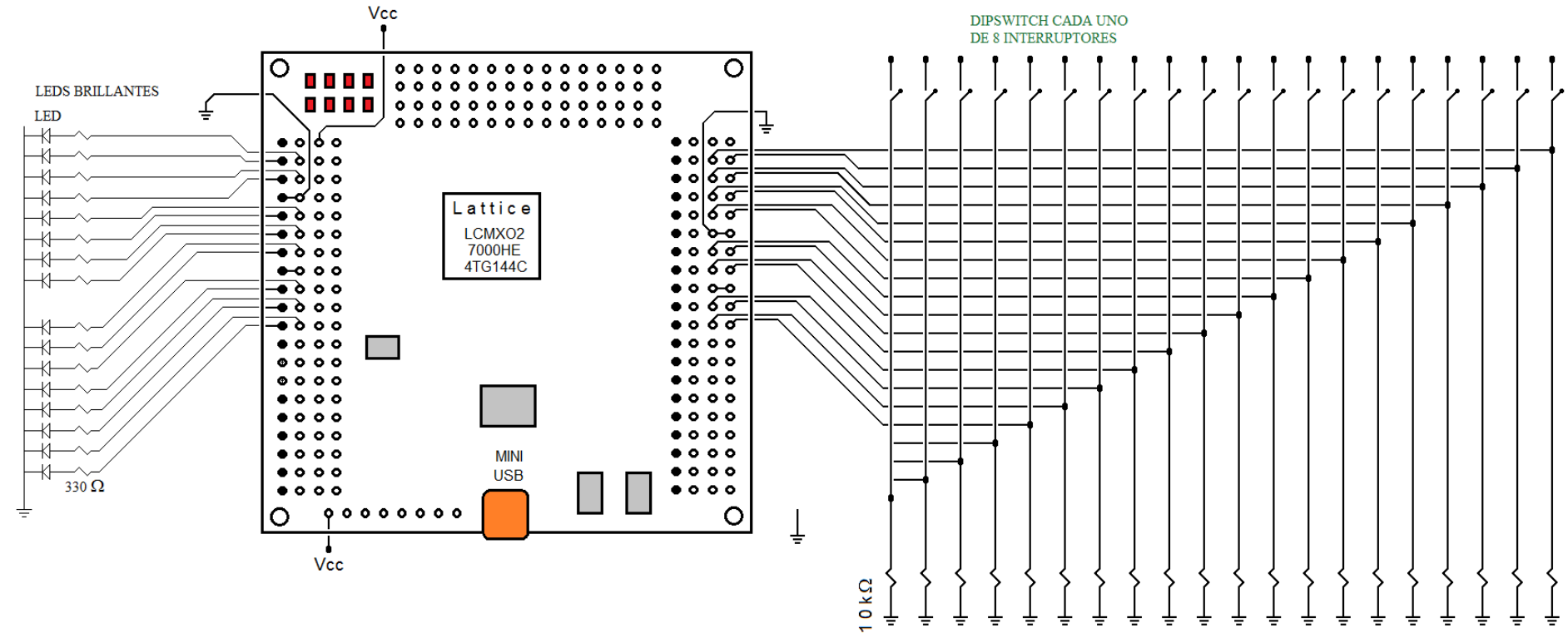
Iniciando con un proyecto sencillo. La puerta AND.

28.- Ahora sigue generar el archivo .jed. Este es el archivo “en lenguaje máquina”, (consta sólo de “0s” y “1s”), que se descargará al FPGA. En la misma ventana de “Process” hacemos doble clic en el proceso “JEDEC File”.



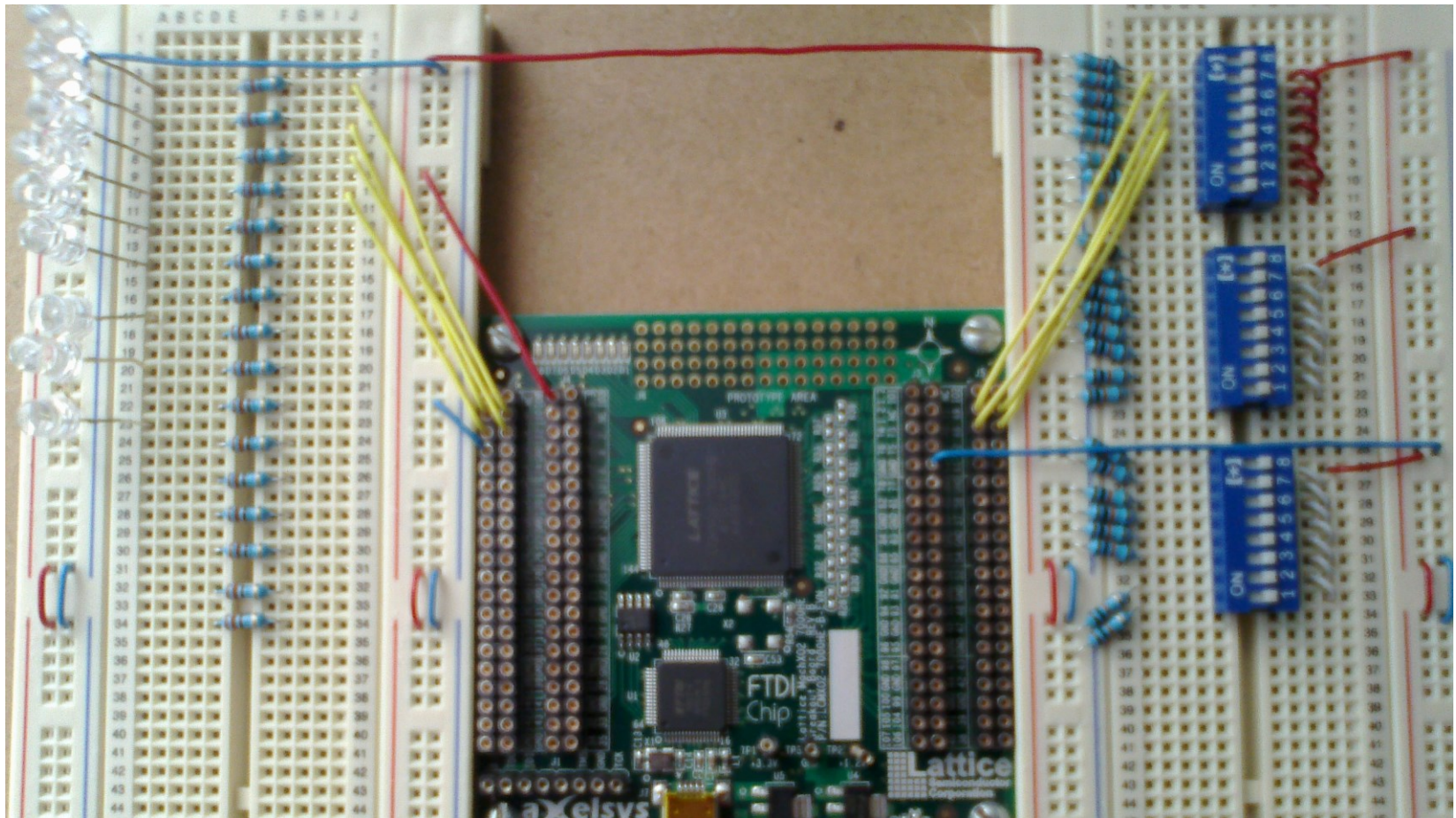
Iniciando con un proyecto sencillo. La puerta AND.

29.- Lo que sigue es asignar las terminales de salida “headers”, que se encuentran conectados a los “dipswitch” y “led”, como se muestra en las siguientes láminas.



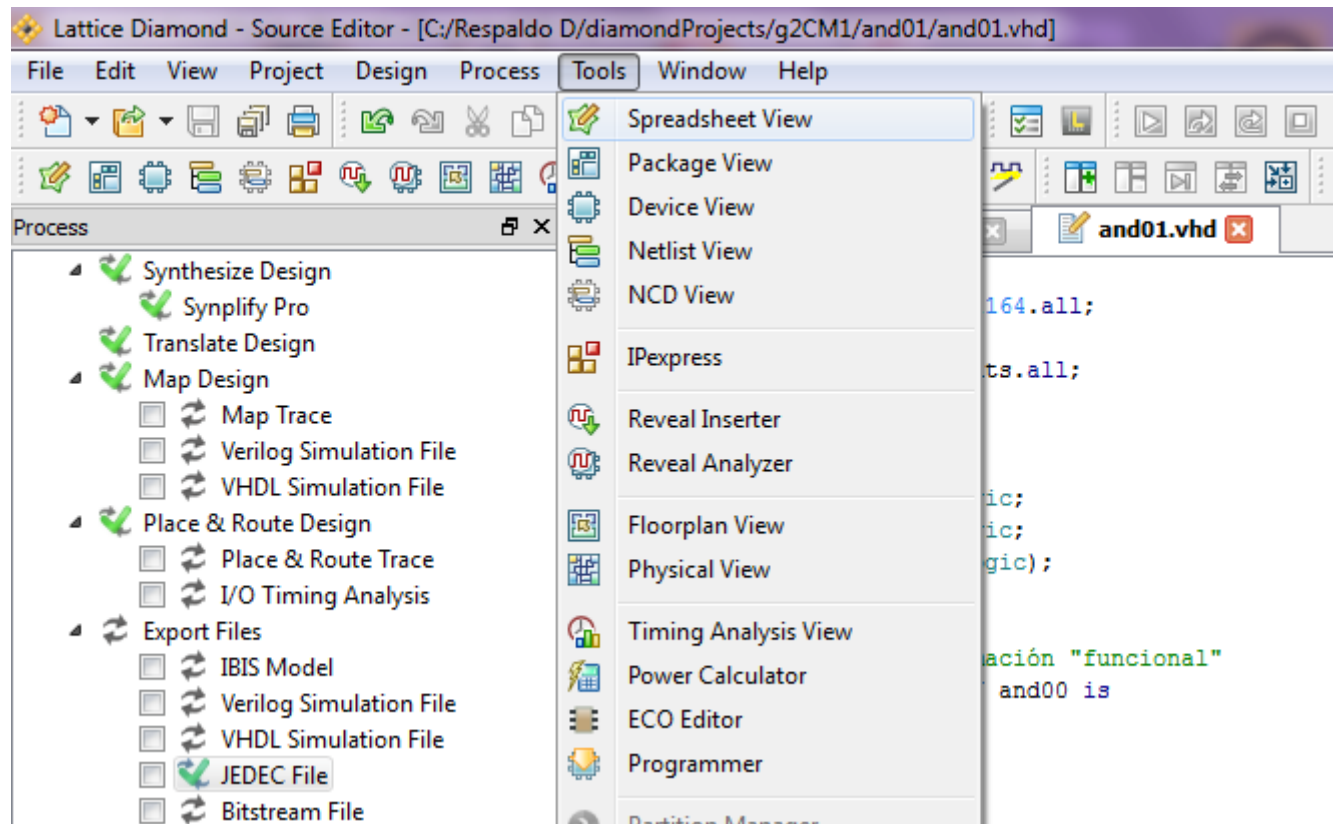
Iniciando con un proyecto sencillo. La puerta AND.

30.- También se muestra la foto de cómo queda alambrado el circuito con la tarjeta



Iniciando con un proyecto sencillo. La puerta AND.

31.- Lo que sigue es asignar pines para la implementación del proyecto. Desde el menú principal del ambiente de desarrollo “Diamond”, seleccionar “Tools” y luego “Spreadsheet View”, como se muestra en la siguiente pantalla.



Iniciando con un proyecto sencillo. La puerta AND.

32.- Después de lo cual se lanza una pantalla como la que se muestra a continuación. Hacer clic en la pantalla inferior rotulada “Pin Assingments”.

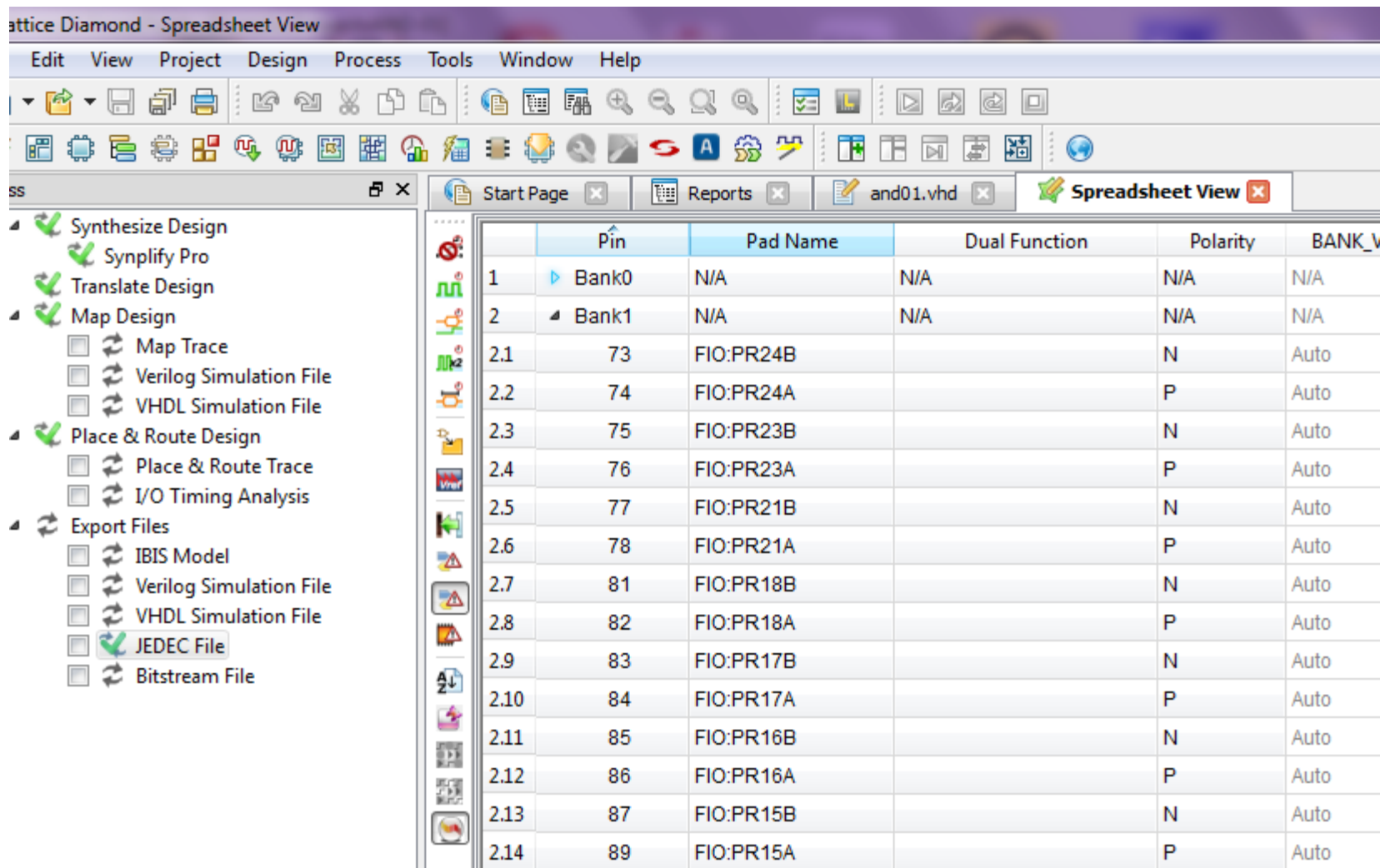
The screenshot displays the Synplify Pro software interface. On the left, the 'Process' pane shows a tree of design steps: Synthesize Design (Synplify Pro), Translate Design, Map Design (with sub-items Map Trace, Verilog Simulation File, and VHDL Simulation File), Place & Route Design (with sub-items Place & Route Trace and I/O Timing Analysis), and Export Files (with sub-items IBIS Model, Verilog Simulation File, VHDL Simulation File, JEDEC File, and Bitstream File). The 'Pin Assignments' window is open in the center, showing a table of pin configurations. The table has columns for Name, Group By, Pin, BANK, BANK_VCC, VREF, IO_TYPE, PULLMODE, and DRIVE. The rows are organized hierarchically: 1 (All ...), 1.1, 1.1.1, 1.1.2, 1.2, and 1.2.1. The 'Pin' column shows values like (34), (35), and (33). The 'BANK' column shows (3). The 'BANK_VCC' column shows Auto. The 'VREF' column shows N/A. The 'IO_TYPE' column shows LVC MOS25(LVC MOS25). The 'PULLMODE' column shows DOWN(DOWN). The 'DRIVE' column shows NA(NA) and 8(8). At the bottom, the 'Output' pane shows 'Performance: 4'. The bottom navigation bar includes tabs for File..., Process, Hierarchy---Post Map Res..., Port Assignments, Pin Assignments, Clock Resource, Route Priority, Cell Mapping, Global Preferences, Timing Preferences, Group, and Misc Prefere.

Name	Group By	Pin	BANK	BANK_VCC	VREF	IO_TYPE	PULLMODE	DRIVE
1	All ...	N/A	N/A	N/A	N/A	N/A		N/A
1.1		N/A	N/A	N/A	N/A	N/A	N/A	N/A
1.1.1		(34)	(3)	Auto	N/A	LVC MOS25(LVC MOS25)	DOWN(DOWN)	NA(NA)
1.1.2		(35)	(3)	Auto	N/A	LVC MOS25(LVC MOS25)	DOWN(DOWN)	NA(NA)
1.2		N/A	N/A	N/A	N/A	N/A	N/A	N/A
1.2.1		(33)	(3)	Auto	N/A	LVC MOS25(LVC MOS25)	DOWN(DOWN)	8(8)

Performance: 4

Iniciando con un proyecto sencillo. La puerta AND.

33.- Extender, con el cursor, la columna “Pin”, con el objeto de visualizar la etiqueta o número de los pines de la tarjeta. Si algún banco de pines no se ha desplegado, hacer doble clic sobre el banco respectivo para visualizar sus pines. Por ejemplo



The screenshot shows the Lattice Diamond Spreadsheet View interface. The left sidebar contains a project tree with the following items:

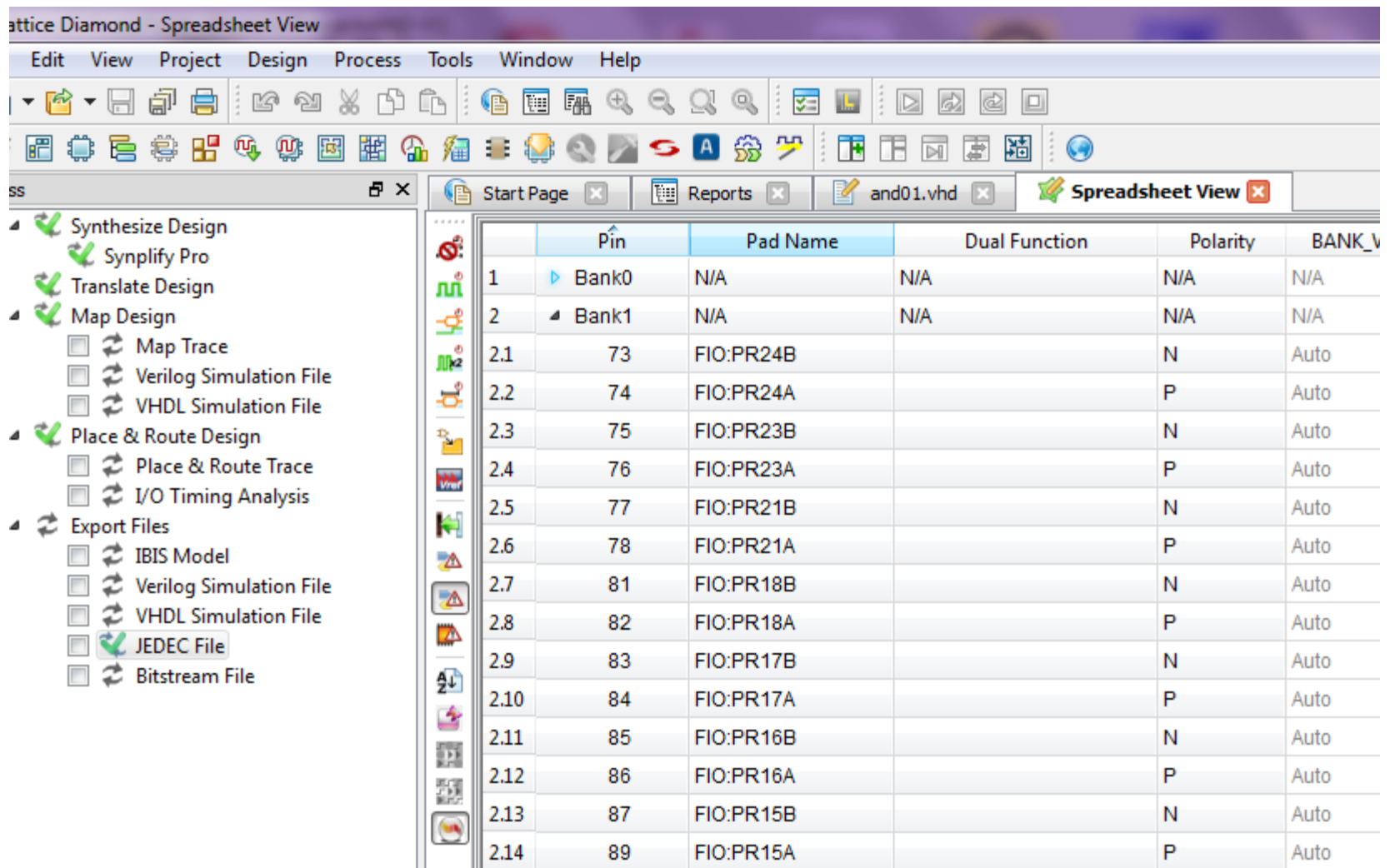
- Synthesize Design
 - Synplify Pro
- Translate Design
- Map Design
 - Map Trace
 - Verilog Simulation File
 - VHDL Simulation File
- Place & Route Design
 - Place & Route Trace
 - I/O Timing Analysis
- Export Files
 - IBIS Model
 - Verilog Simulation File
 - VHDL Simulation File
 - JEDEC File
 - Bitstream File

The main area displays a table with the following columns: Pin, Pad Name, Dual Function, Polarity, and BANK_V. The table shows the configuration for Bank0 and Bank1, with pins 1 through 14 listed.

	Pin	Pad Name	Dual Function	Polarity	BANK_V
1	Bank0	N/A	N/A	N/A	N/A
2	Bank1	N/A	N/A	N/A	N/A
2.1	73	FIO:PR24B		N	Auto
2.2	74	FIO:PR24A		P	Auto
2.3	75	FIO:PR23B		N	Auto
2.4	76	FIO:PR23A		P	Auto
2.5	77	FIO:PR21B		N	Auto
2.6	78	FIO:PR21A		P	Auto
2.7	81	FIO:PR18B		N	Auto
2.8	82	FIO:PR18A		P	Auto
2.9	83	FIO:PR17B		N	Auto
2.10	84	FIO:PR17A		P	Auto
2.11	85	FIO:PR16B		N	Auto
2.12	86	FIO:PR16A		P	Auto
2.13	87	FIO:PR15B		N	Auto
2.14	89	FIO:PR15A		P	Auto

Iniciando con un proyecto sencillo. La puerta AND.

33.- Si algún banco de pines no se ha desplegado, hacer doble clic sobre el banco respectivo para visualizar sus pines. Por ejemplo, en este caso no se han desplegado los bancos “Bank0” y “Bank1”.



The screenshot shows the Lattice Diamond Spreadsheet View interface. The left sidebar contains a project tree with the following items:

- Synthesize Design
 - Synplify Pro
- Translate Design
- Map Design
 - Map Trace
 - Verilog Simulation File
 - VHDL Simulation File
- Place & Route Design
 - Place & Route Trace
 - I/O Timing Analysis
- Export Files
 - IBIS Model
 - Verilog Simulation File
 - VHDL Simulation File
 - JEDEC File
 - Bitstream File

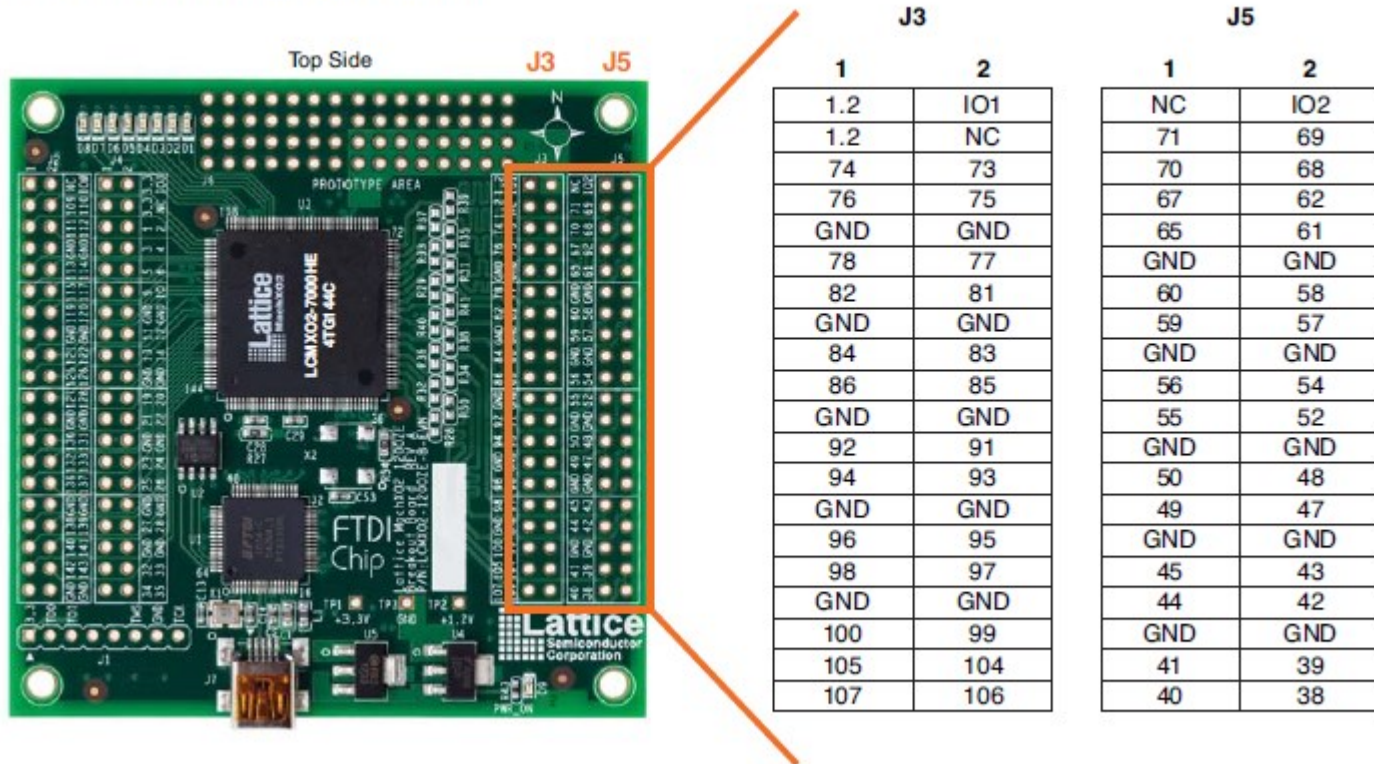
The main area displays a table with the following columns: Pin, Pad Name, Dual Function, Polarity, and BANK_V. The table is divided into two sections for Bank0 and Bank1.

	Pin	Pad Name	Dual Function	Polarity	BANK_V
1	Bank0	N/A	N/A	N/A	N/A
2	Bank1	N/A	N/A	N/A	N/A
2.1	73	FIO:PR24B		N	Auto
2.2	74	FIO:PR24A		P	Auto
2.3	75	FIO:PR23B		N	Auto
2.4	76	FIO:PR23A		P	Auto
2.5	77	FIO:PR21B		N	Auto
2.6	78	FIO:PR21A		P	Auto
2.7	81	FIO:PR18B		N	Auto
2.8	82	FIO:PR18A		P	Auto
2.9	83	FIO:PR17B		N	Auto
2.10	84	FIO:PR17A		P	Auto
2.11	85	FIO:PR16B		N	Auto
2.12	86	FIO:PR16A		P	Auto
2.13	87	FIO:PR15B		N	Auto
2.14	89	FIO:PR15A		P	Auto

Iniciando con un proyecto sencillo. La puerta AND.

34.- En la pantalla 29 se muestran los pines de los “Dip Switch” o interruptores a los que van conectadas ls entradas “Aa” y “Ba”. Además en la siguiente pantalla, tomada del manual de la tarjeta, se muestran los números que asignaremos.

Figure 5. J3/J5 Header Landing Callout



La asignación es



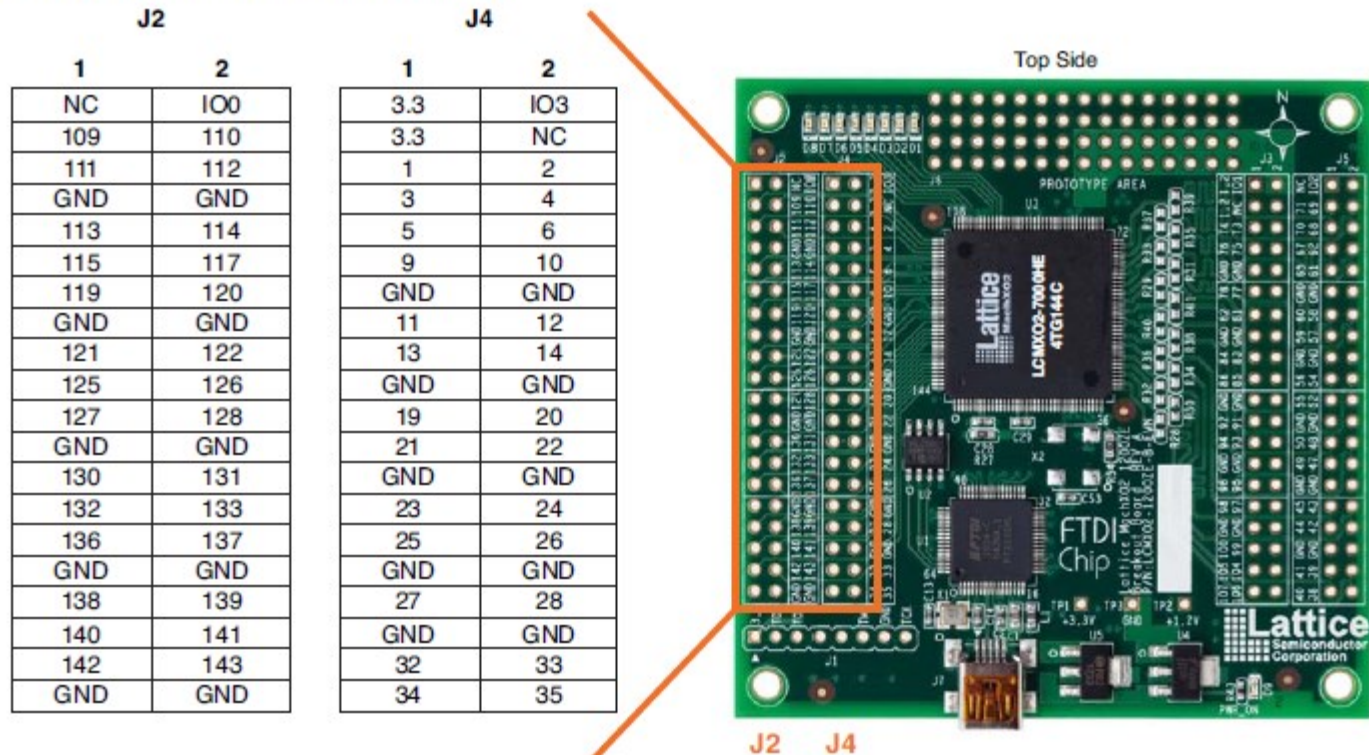
Aa ← pin 71

Ba ← pin 69

Iniciando con un proyecto sencillo. La puerta AND.

35.- Y también se muestra el Pin asignado a la salida “Ya”, a donde se halla conectado un led. Ya ← Pin 109

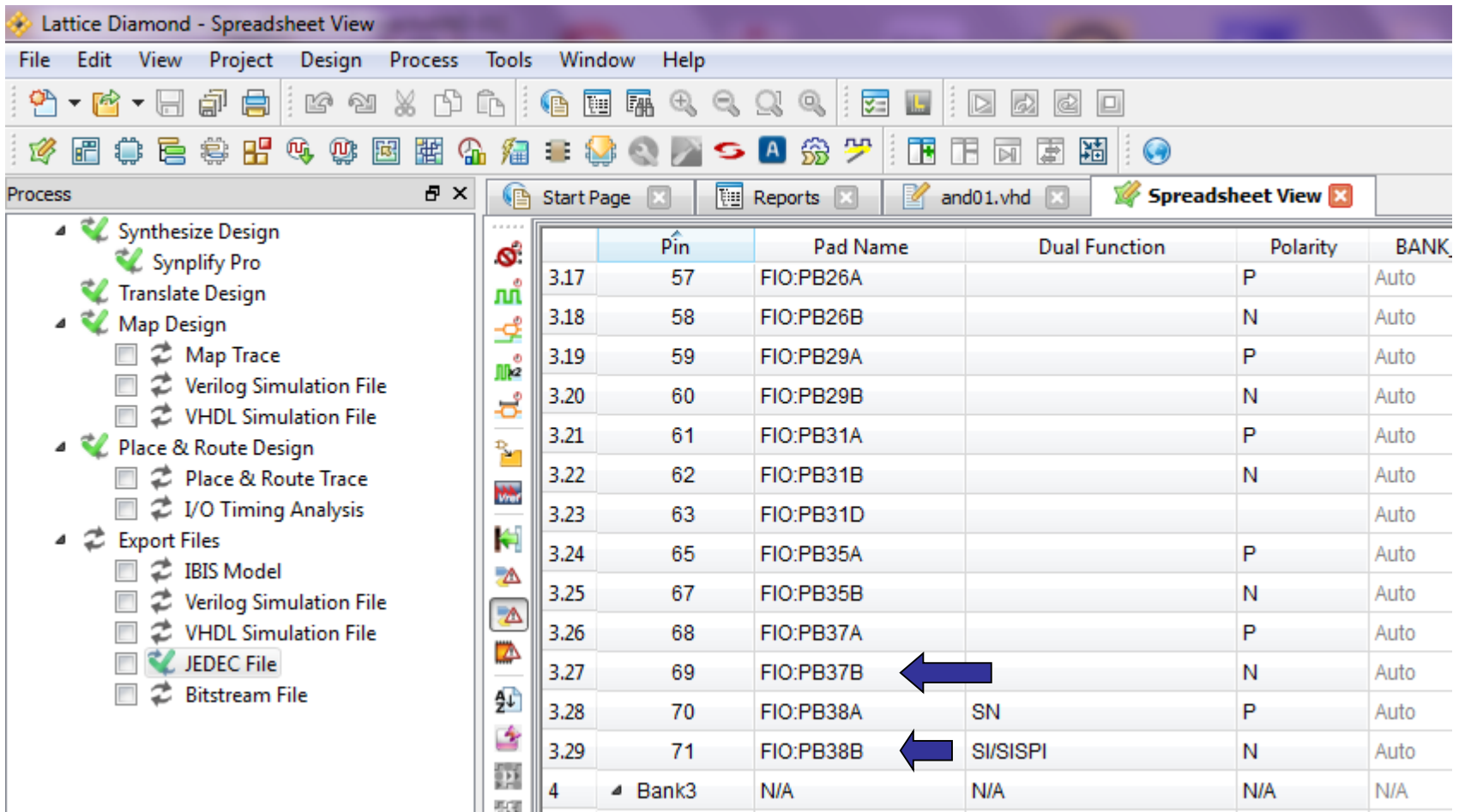
Figure 4. J2/J4 Header Landing Callout



Si ya hemos identificado los pines a los cuales les asignaremos los puertos de entrada y salida, procedemos a hacer esto desde el ambiente “Diamond”.

Iniciando con un proyecto sencillo. La puerta AND.

36.- En el ambiente “Diamond”, desplazarse hacia arriba o hacia abajo, para localizar los pines 69 y 71.



The screenshot shows the Lattice Diamond Spreadsheet View interface. The left sidebar displays the project process flow, with 'Export Files' and 'JEDEC File' selected. The main area shows a table of pin configurations. Two blue arrows point to the rows for pins 69 and 71.

	Pin	Pad Name	Dual Function	Polarity	BANK
3.17	57	FIO:PB26A		P	Auto
3.18	58	FIO:PB26B		N	Auto
3.19	59	FIO:PB29A		P	Auto
3.20	60	FIO:PB29B		N	Auto
3.21	61	FIO:PB31A		P	Auto
3.22	62	FIO:PB31B		N	Auto
3.23	63	FIO:PB31D			Auto
3.24	65	FIO:PB35A		P	Auto
3.25	67	FIO:PB35B		N	Auto
3.26	68	FIO:PB37A		P	Auto
3.27	69	FIO:PB37B		N	Auto
3.28	70	FIO:PB38A	SN	P	Auto
3.29	71	FIO:PB38B	SI/SISPI	N	Auto
4	Bank3	N/A	N/A	N/A	N/A

Iniciando con un proyecto sencillo. La puerta AND.

37.- Seleccionar el renglón donde se halla el Pin 71

Lattice Diamond - Spreadsheet View

File Edit View Project Design Process Tools Window Help

Process Start Page Reports and01.vhd Spreadsheet View

- Synthesize Design
 - Synplify Pro
- Translate Design
- Map Design
 - Map Trace
 - Verilog Simulation File
 - VHDL Simulation File
- Place & Route Design
 - Place & Route Trace
 - I/O Timing Analysis
- Export Files
 - IBIS Model
 - Verilog Simulation File
 - VHDL Simulation File
 - JEDEC File
 - Bitstream File

	Pin	Pad Name	Dual Function	Polarity	BANK_VCC	IO_TYPE	Signal Name	Signal Type
3.17	57	FIO:PB26A		P	Auto			
3.18	58	FIO:PB26B		N	Auto			
3.19	59	FIO:PB29A		P	Auto			
3.20	60	FIO:PB29B		N	Auto			
3.21	61	FIO:PB31A		P	Auto			
3.22	62	FIO:PB31B		N	Auto			
3.23	63	FIO:PB31D			Auto			
3.24	65	FIO:PB35A		P	Auto			
3.25	67	FIO:PB35B		N	Auto			
3.26	68	FIO:PB37A		P	Auto			
3.27	69	FIO:PB37B		N	Auto			
3.28	70	FIO:PB38A	SN	P	Auto			
3.29	71	FIO:PB38B	SI/SISPI	N	Auto			
4	Bank3	N/A	N/A	N/A	N/A	N/A	N/A	N/A
4.1	25	FIO:PL19A		P	Auto			
4.2	26	FIO:PL19B		N	Auto			

Iniciando con un proyecto sencillo. La puerta AND.

38.- A continuación, pararse en el cruce del renglón del Pin 71 y la columna “Signal Name” . Hacer doble clic en dicho cruce. Automáticamente se marca con azul, como se muestra en la siguiente pantalla.

Lattice Diamond - Spreadsheet View

File Edit View Project Design Process Tools Window Help

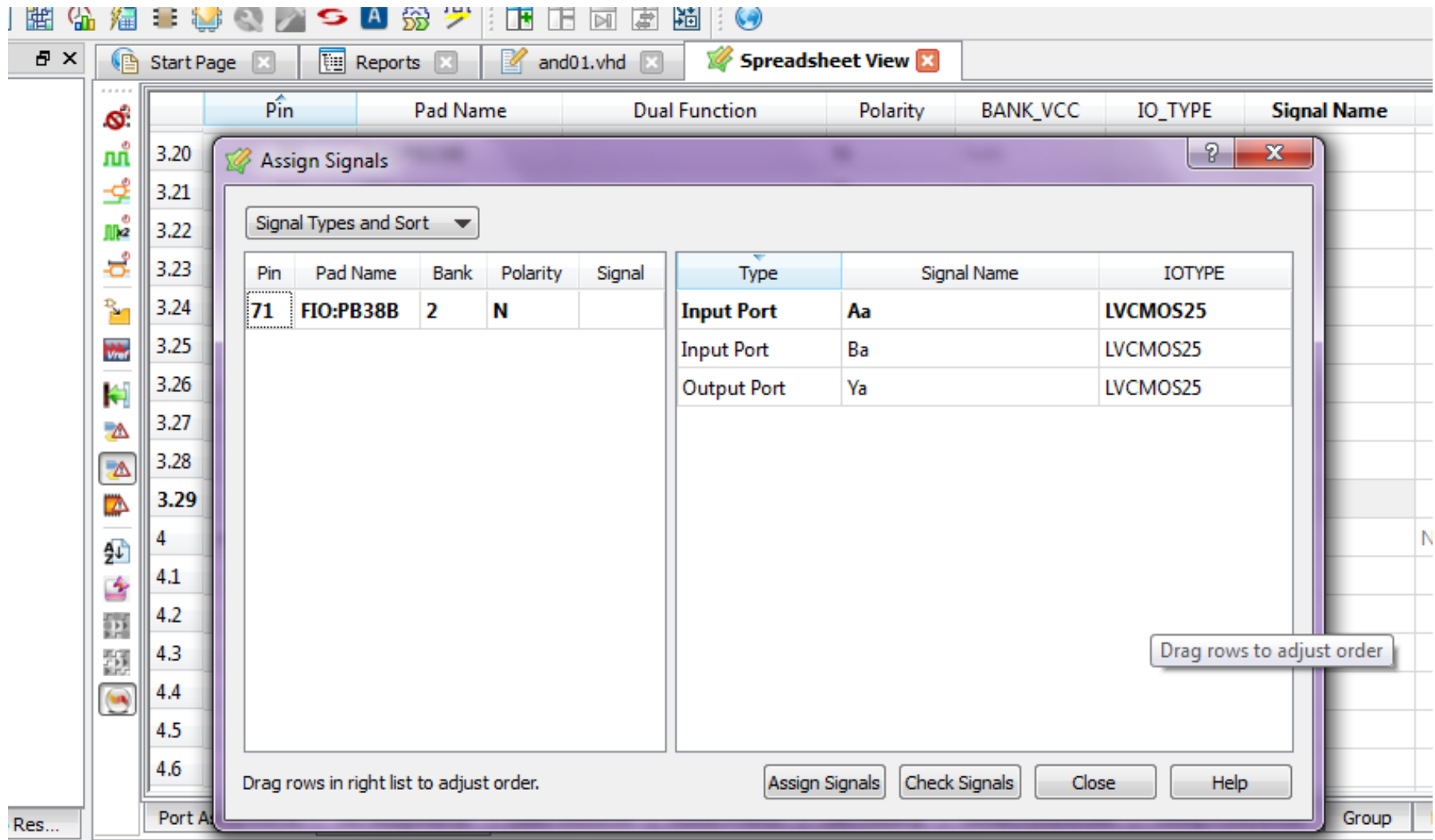
Process Start Page Reports and01.vhd Spreadsheet View

- Synthesize Design
 - Synplify Pro
 - Translate Design
- Map Design
 - Map Trace
 - Verilog Simulation File
 - VHDL Simulation File
- Place & Route Design
 - Place & Route Trace
 - I/O Timing Analysis
- Export Files
 - IBIS Model
 - Verilog Simulation File
 - VHDL Simulation File
 - JEDEC File
 - Bitstream File

	Pin	Pad Name	Dual Function	Polarity	BANK_VCC	IO_TYPE	Signal Name	Signal Name
3.17	57	FIO:PB26A		P	Auto			
3.18	58	FIO:PB26B		N	Auto			
3.19	59	FIO:PB29A		P	Auto			
3.20	60	FIO:PB29B		N	Auto			
3.21	61	FIO:PB31A		P	Auto			
3.22	62	FIO:PB31B		N	Auto			
3.23	63	FIO:PB31D			Auto			
3.24	65	FIO:PB35A		P	Auto			
3.25	67	FIO:PB35B		N	Auto			
3.26	68	FIO:PB37A		P	Auto			
3.27	69	FIO:PB37B		N	Auto			
3.28	70	FIO:PB38A	SN	P	Auto			
3.29	71	FIO:PB38B	SI/SISPI	N	Auto			
4	Bank3	N/A	N/A	N/A	N/A	N/A	N/A	N/A
4.1	25	FIO:PL19A		P	Auto			

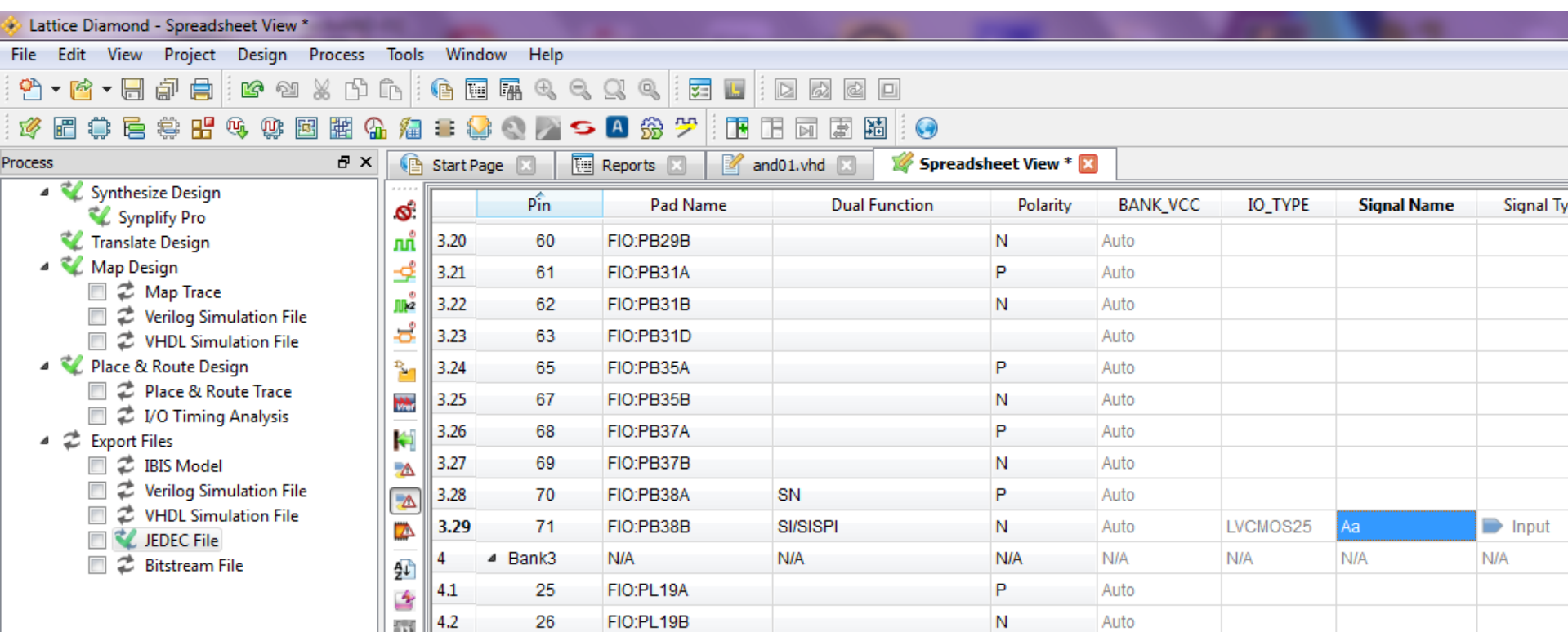
Iniciando con un proyecto sencillo. La puerta AND.

39.- Después de hacer doble clic, se lanza el siguiente asistente de asignación. Seleccionar el renglón "Input Port Aa". Después hacer clic sobre la pestaña "Assign Signals".



Iniciando con un proyecto sencillo. La puerta AND.

40.- Con lo anterior, quedará asignado el pin 71 a la entrada “Aa” de la puerta AND.



The screenshot shows the Lattice Diamond Spreadsheet View for a project named 'and01.vhd'. The interface includes a menu bar (File, Edit, View, Project, Design, Process, Tools, Window, Help), a toolbar, and a left-hand pane with a project tree. The project tree shows the following structure:

- Process
 - Synthesize Design
 - Synplify Pro
 - Translate Design
 - Map Design
 - Map Trace
 - Verilog Simulation File
 - VHDL Simulation File
 - Place & Route Design
 - Place & Route Trace
 - I/O Timing Analysis
 - Export Files
 - IBIS Model
 - Verilog Simulation File
 - VHDL Simulation File
 - JEDEC File
 - Bitstream File

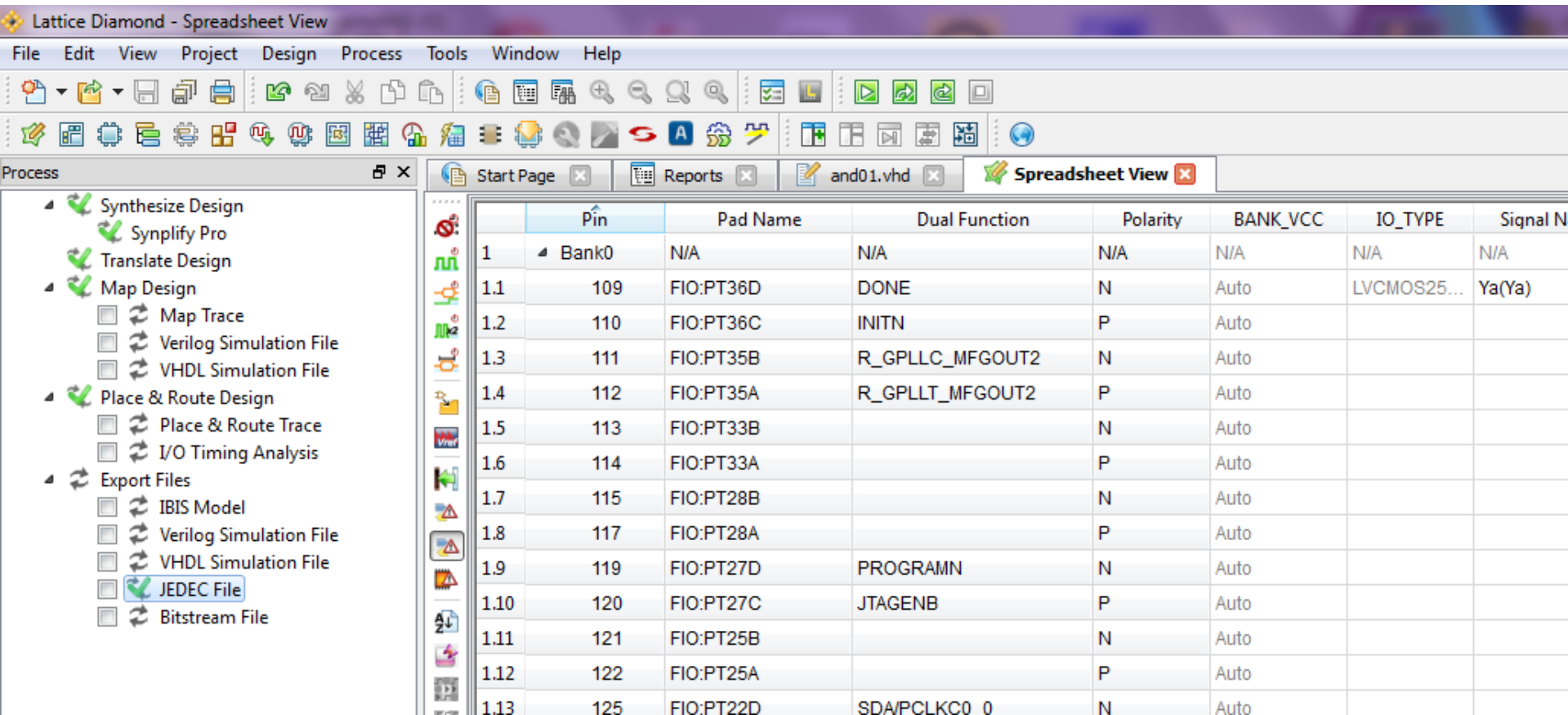
The main spreadsheet table displays pin assignments. The columns are: Pin, Pad Name, Dual Function, Polarity, BANK_VCC, IO_TYPE, Signal Name, and Signal Type. The row for Pin 71 is highlighted in blue.

	Pin	Pad Name	Dual Function	Polarity	BANK_VCC	IO_TYPE	Signal Name	Signal Ty
3.20	60	FIO:PB29B		N	Auto			
3.21	61	FIO:PB31A		P	Auto			
3.22	62	FIO:PB31B		N	Auto			
3.23	63	FIO:PB31D			Auto			
3.24	65	FIO:PB35A		P	Auto			
3.25	67	FIO:PB35B		N	Auto			
3.26	68	FIO:PB37A		P	Auto			
3.27	69	FIO:PB37B		N	Auto			
3.28	70	FIO:PB38A	SN	P	Auto			
3.29	71	FIO:PB38B	SI/SISPI	N	Auto	LVC MOS25	Aa	Input
4	Bank3	N/A	N/A	N/A	N/A	N/A	N/A	N/A
4.1	25	FIO:PL19A		P	Auto			
4.2	26	FIO:PL19B		N	Auto			

Repetir los pasos anteriores para asignar pines a la entrada “Ba” y a la salida “Ya”.

Iniciando con un proyecto sencillo. La puerta AND.

41.- Regresar a la ventana “Process” del ambiente de desarrollo “Diamond”, a la izquierda de la pantalla principal. Hacer doble clic sobre el proceso “JEDEC Files”



The screenshot shows the Lattice Diamond - Spreadsheet View interface. The left sidebar displays the 'Process' tree with the following steps:

- ✓ Synthesize Design
 - ✓ Synplify Pro
- ✓ Translate Design
- ✓ Map Design
 - Map Trace
 - Verilog Simulation File
 - VHDL Simulation File
- ✓ Place & Route Design
 - Place & Route Trace
 - I/O Timing Analysis
- Export Files
 - IBIS Model
 - Verilog Simulation File
 - VHDL Simulation File
 - JEDEC File**
 - Bitstream File

The main area displays the 'and01.vhd' file in 'Spreadsheet View'. The table below represents the data shown in the spreadsheet:

	Pin	Pad Name	Dual Function	Polarity	BANK_VCC	IO_TYPE	Signal N
1	Bank0	N/A	N/A	N/A	N/A	N/A	N/A
1.1	109	FIO:PT36D	DONE	N	Auto	LVC MOS25...	Ya(Ya)
1.2	110	FIO:PT36C	INITN	P	Auto		
1.3	111	FIO:PT35B	R_GPLL C_MFGOUT2	N	Auto		
1.4	112	FIO:PT35A	R_GPL LT_MFGOUT2	P	Auto		
1.5	113	FIO:PT33B		N	Auto		
1.6	114	FIO:PT33A		P	Auto		
1.7	115	FIO:PT28B		N	Auto		
1.8	117	FIO:PT28A		P	Auto		
1.9	119	FIO:PT27D	PROGRAMN	N	Auto		
1.10	120	FIO:PT27C	JTAGENB	P	Auto		
1.11	121	FIO:PT25B		N	Auto		
1.12	122	FIO:PT25A		P	Auto		
1.13	125	FIO:PT22D	SDA/PCL KC0 0	N	Auto		

Con esto último ya ha quedado listo el archivo “jed”, para ser descargado a la tarjeta

En el siguiente conjunto de diapositivas continuaré con el procedimiento de descarga del archivo “.jed” a la tarjeta