

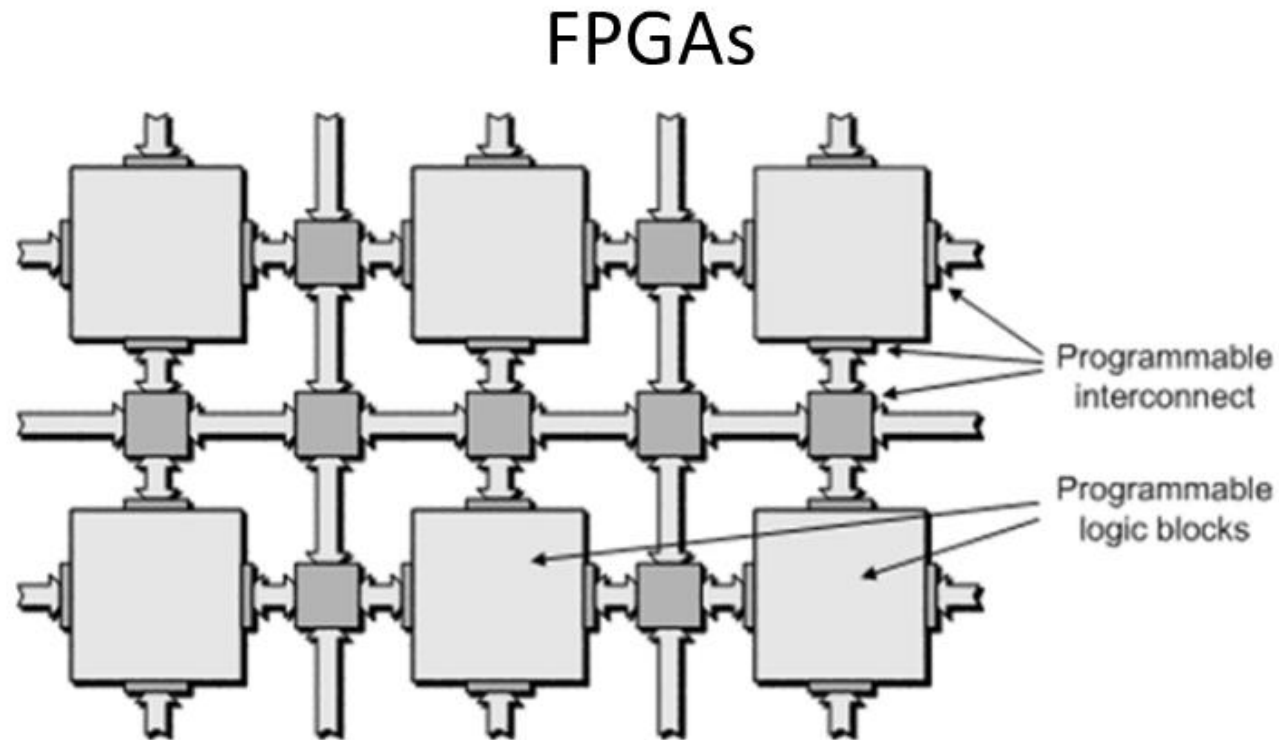
# FPGA BASIC PRINCIPLES & VHDL INTRODUCTION

- ❑ What is FPGA?
- ❑ FPGA basic components
- ❑ FPGA Design Flow.
- ❑ RTL-register transfer level.
- ❑ Parallel processing verse serial processing.
- ❑ **Lab 1: VIVADO-using the tools-creating a new project.**

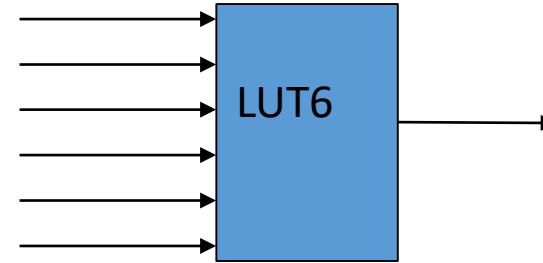
**החומרים לשימוש פנימי בלבד אין להשתמש או להעביר ללא רשות מפורשת בכתב מניר בלולו**

# FPGA-introduction

*FPGA-filed programmed gate array*

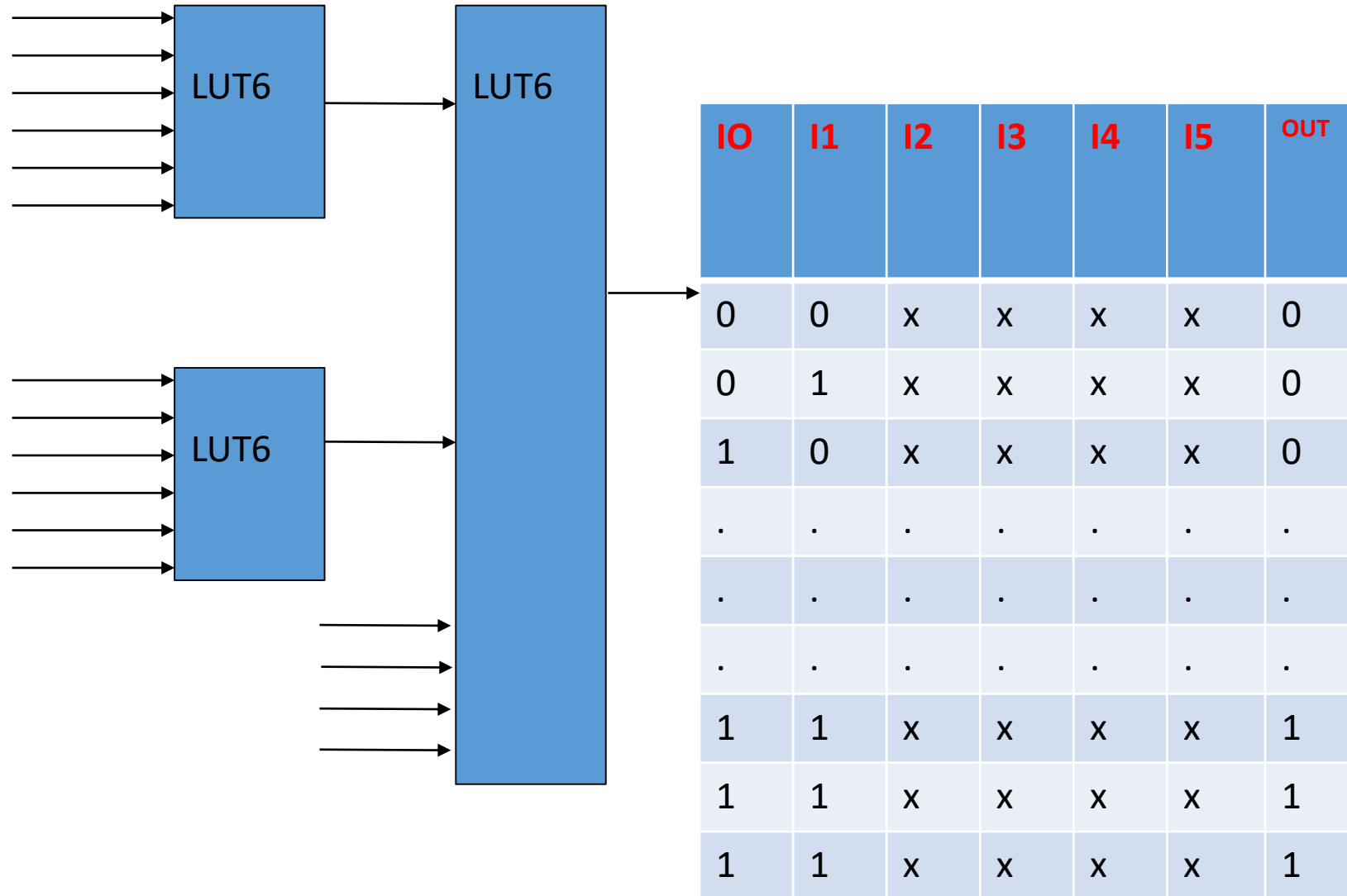


# LUT –comparator example

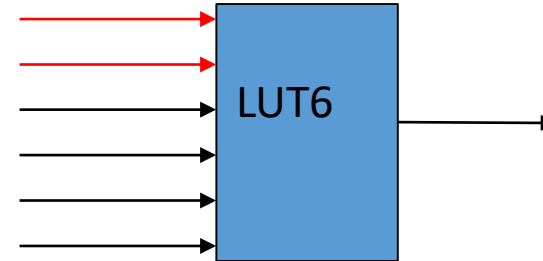


IO	I1	I2	I3	I4	I5	OUT
0	0	0	0	0	0	1
0	0	0	0	0	1	0
0	1	0	0	1	0	1
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
1	1	1	1	0	1	0
1	1	1	1	1	0	0
1	1	1	1	1	1	1

# LUT- COMPARATOR "LEGO"-



# LUT- MULTIPLEXER EXMAPLE

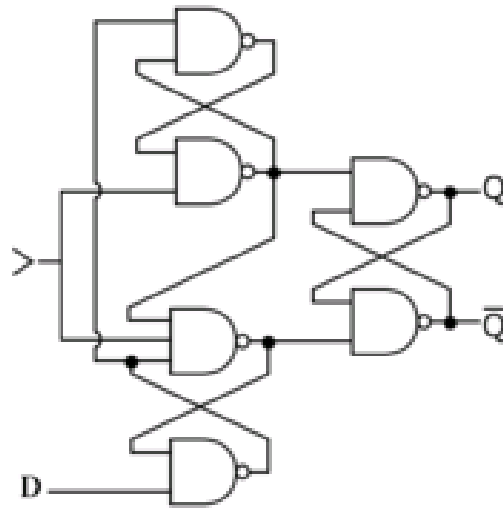


IO(sel1)	I1(sel0)	I2	I3	I4	I5	OUT
0	0	0	0	0	0	0
0	1	0	0	1	0	1
1	0	0	1	1	0	0
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
1	1	0	1	0	1	0
1	1	0	1	1	0	0
1	1	1	1	1	1	1

# FLIP-FLOP

**flip-flop** has two stable states and thereby is capable of serving as one bit of memory.

A flip-flop is usually controlled by one or two control signals and/or a gate or clock signal

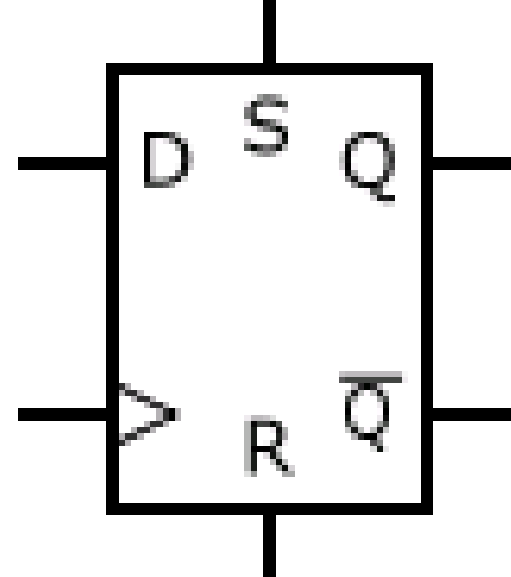


# D-FLIP FLOP

The Q output always takes on the state of the D input at the moment of a rising clock edge (or falling edge if the clock input is active low).. The D flip-flop can be interpreted as a primitive memory cell, delay line.

**Truth table:**

Clock	D	Q
Rising edge	0	0
Rising edge	1	1
Non-Rising	X	$Q_p$
		rev



# VHDL

What is Vhdl?

**VHDL-VHSIC** *hardware description language*

**VHSIC-Very *High-Speed Integrated Circuit***



# ENTITY.

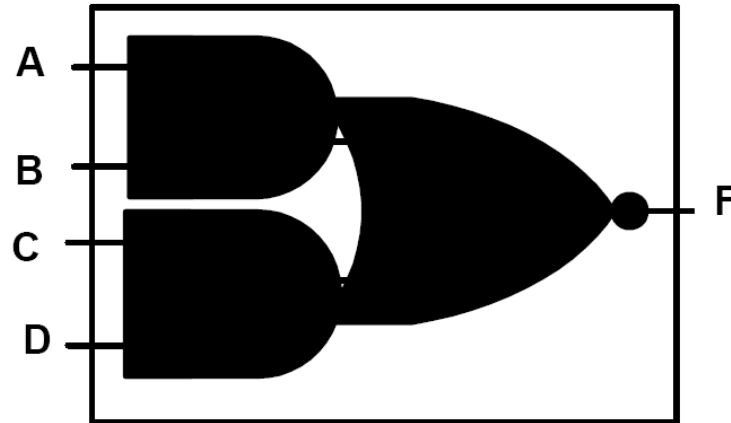
The *entity declaration* represents the external interface to the design entity.



# ARCHITECTURE

The *architecture body* represents the internal description of the design

entity - its behavior, its structure, or a mixture of both. In this example, we want to describe an and-or-invert (**AND\_OR\_INV**) gate in VHDL. If we consider the **AND\_OR\_INV** gate as a single chip package, it will have four input pins and one output pin.



# ENTITY

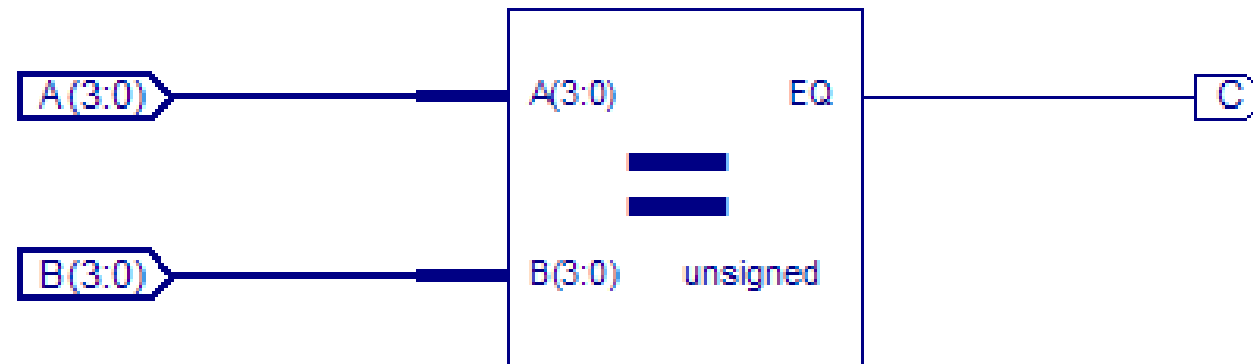
entity **my\_comp** is

Port ( A : in STD\_LOGIC\_VECTOR (3 downto 0);

B : in STD\_LOGIC\_VECTOR (3 downto 0);

C : out STD\_LOGIC);

end **my\_comp**;



# *Process*

architecture *my\_comp* of *my\_comp* is

begin

*Process* (A,B)

*begin*

*if* (A=B) *then*

*C* <= '1';

*else*

*C* <= '0';

*end if*;

*end process*;

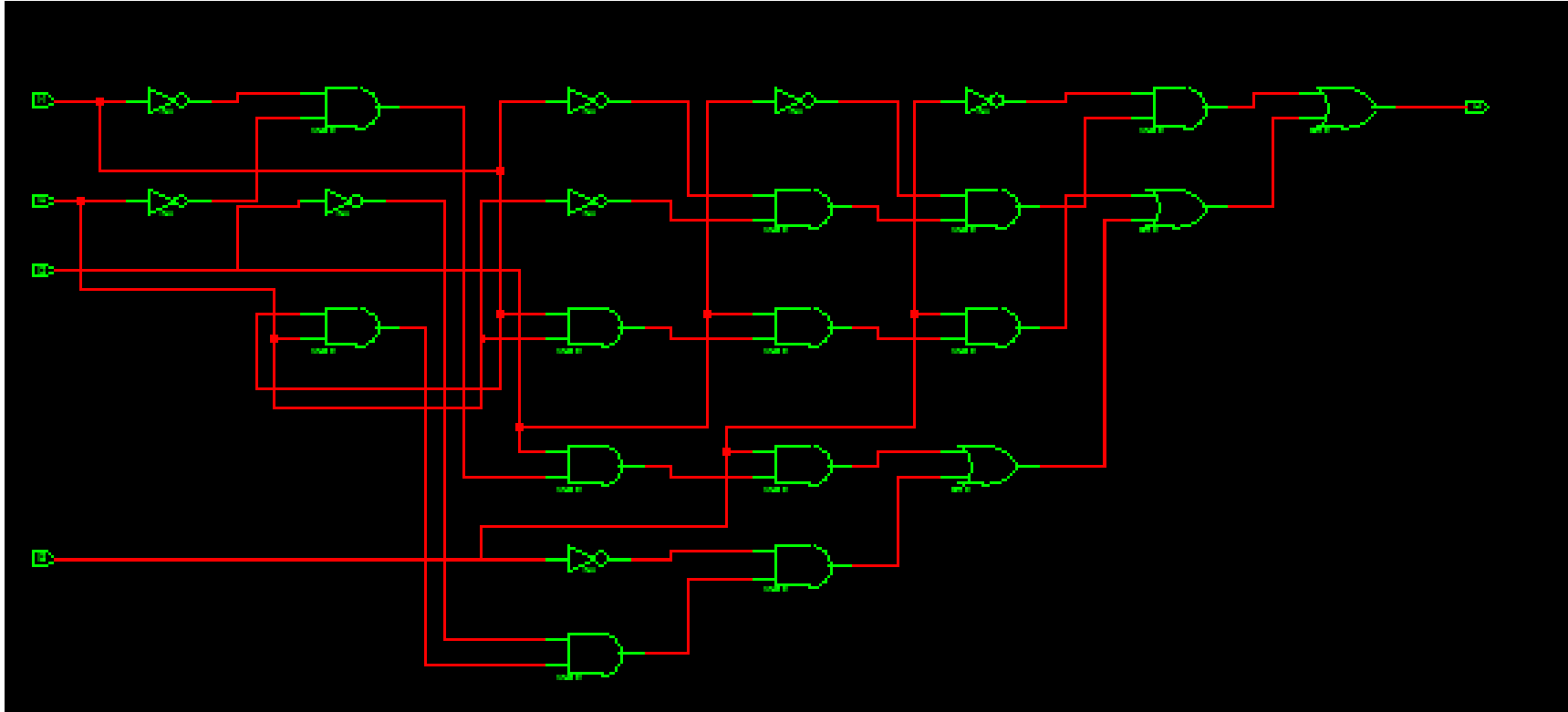
*end my\_comp*;

# Kranugh map-*synthesis*

$i_0 i_1 \backslash i_2 i_3$	00	01	11	10
00	1	0	1	0
01	0	0	0	0
11	1	0	1	0
10	0	0	0	0

$$(i_0 * i_1 * i_2 * i_3) + (i_0 * i_1 * i_2 * i_3) + (i_0 * i_1 * i_2 * i_3) + (i_0 * i_1 * i_2 * i_3)$$

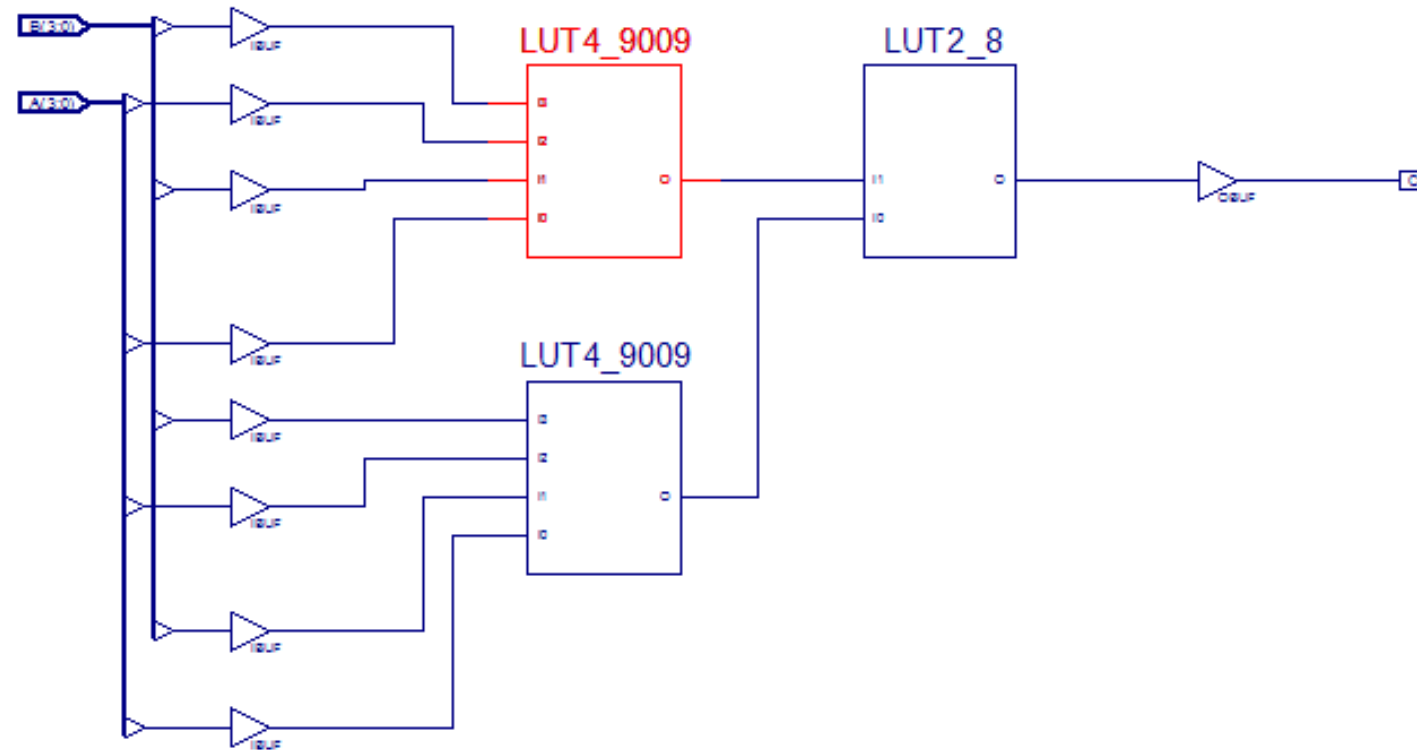
# Kranugh map-*synthesis*



# FPGA-ram based-*synthesis*

I3	I2	I1	I0	O
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

# FPGA-ram based *-synthesis*





## Device utilization –*Synthesis report*

### Device utilization summary:

-----

Selected Device : 3s1600efg320-5

Number of Slices:	2 out of 14752	0%
Number of 4 input LUTs:	3 out of 29504	0%
Number of bonded IOBs:	9 out of 250	3%

# Timing -*Synthesis report*

Timing Summary:

-----

Delay: 7.587ns (Levels of Logic = 4)  
Source: A<1> (PAD)  
Destination: C (PAD)

Data Path: A<1> to C

				Gate	Net
Cell:in->out	fan-out	Delay	Delay	Logical	Name (Net Name)
-----					
IBUF:I->O	1	1.106	0.833		A_1_IBUF (A_1_IBUF)
LUT4:I0->O	1	0.612	0.833		C426 (C4_map33)
LUT2:I0->O	1	0.612	0.681		C454 (C_OBUF)
OBUF:I->O		2.910			C_OBUF (C)

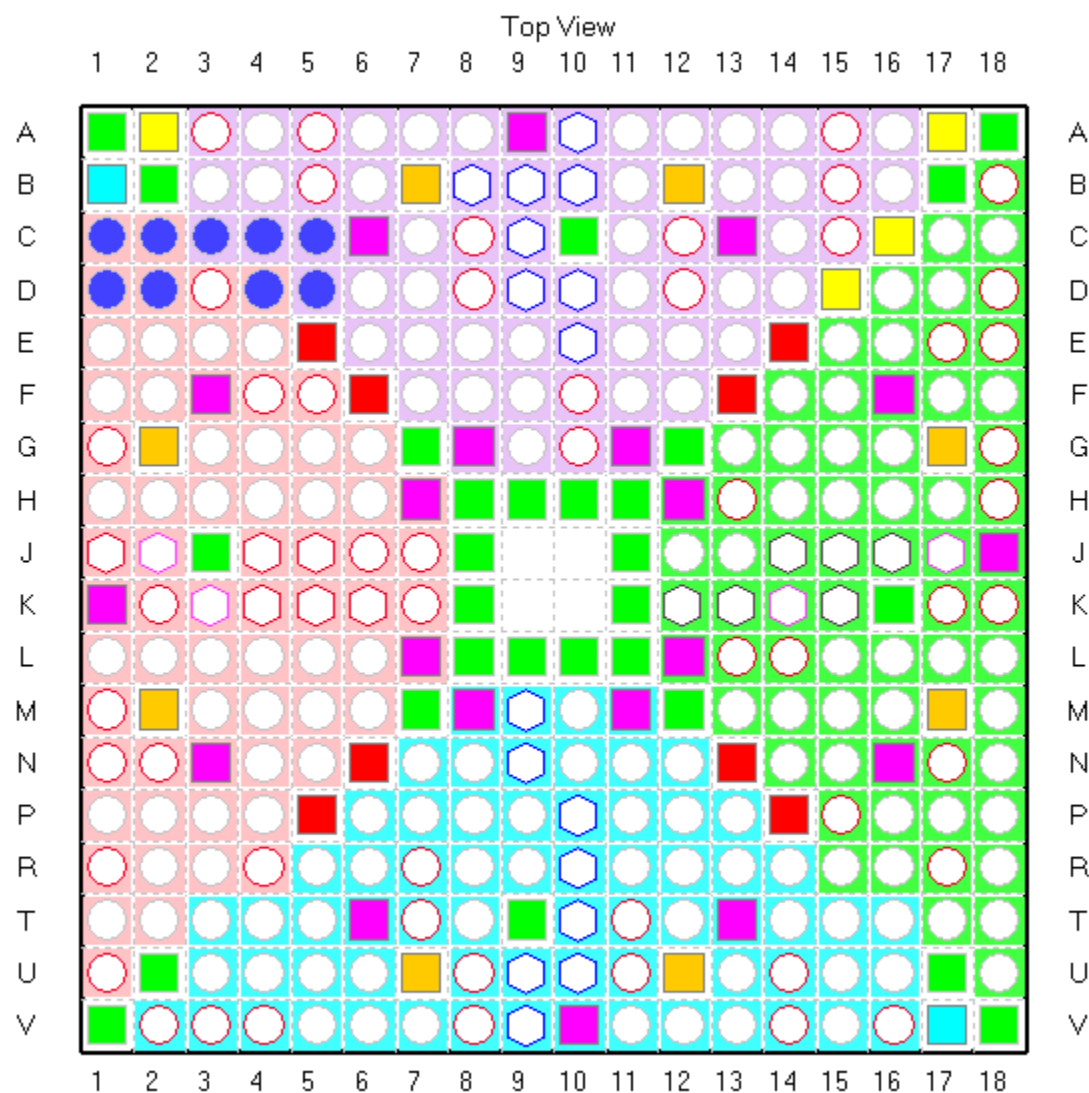
-----

**Total** 7.587ns (5.240ns logic, 2.347ns route)  
**69.1%**logic, **30.9%** route

# Assign package pin-*user constrains*

	I/O Name	I/O Direction	Loc
	A<0>	Input	C4
	A<1>	Input	D4
	A<2>	Input	C5
	A<3>	Input	D5
	B<0>	Input	C1
	B<1>	Input	D1
	B<2>	Input	C2
	B<3>	Input	D2
	C	Output	C3

# Assign package pin-*user constrains*



# mapping

## Design Summary

-----  
Number of errors: 0

Number of warnings: 0

Logic Utilization:

Number of 4 input LUTs: 3 out of 29,504 1%

Logic Distribution:

Number of occupied Slices: 2 out of 14,752 1%

Number of Slices containing only related logic: 2 out of 2 100%

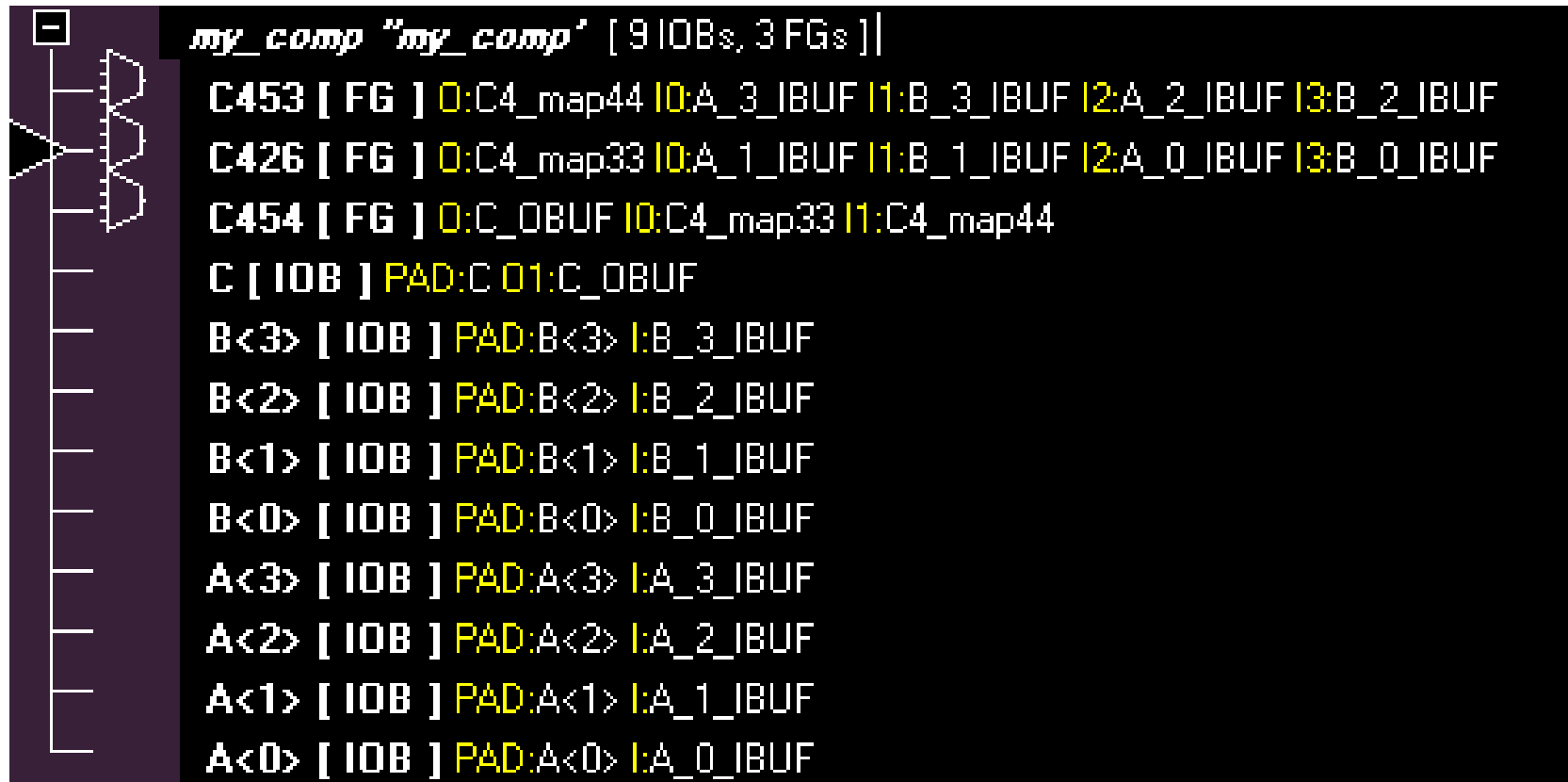
Number of Slices containing unrelated logic: 0 out of 2 0%

\* See NOTES below for an explanation of the effects of unrelated logic

Total Number of 4 input LUTs: 3 out of 29,504 1%

Number of bonded IOBs: 9 out of 250 3%

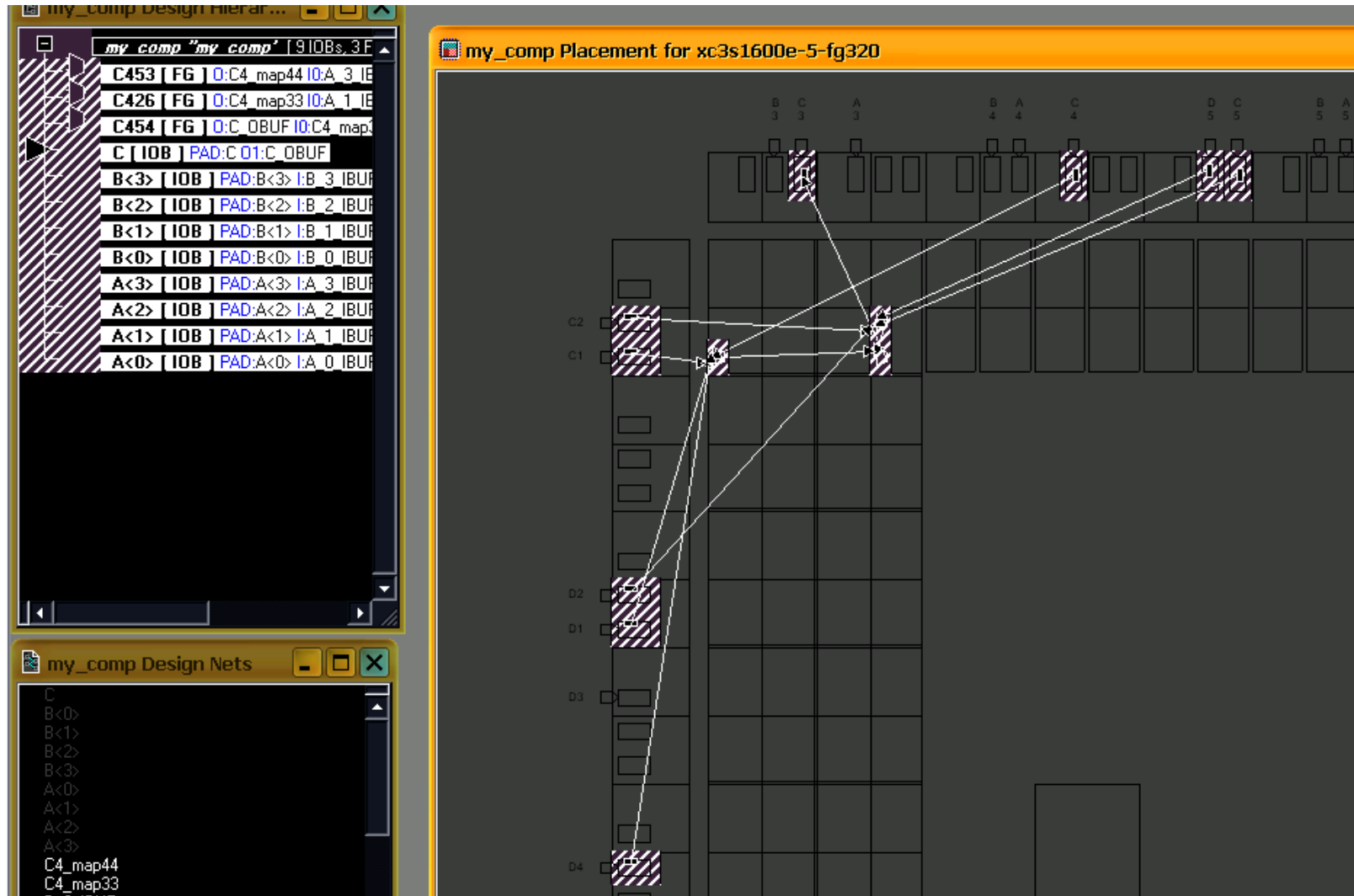
# mapping



# Placing

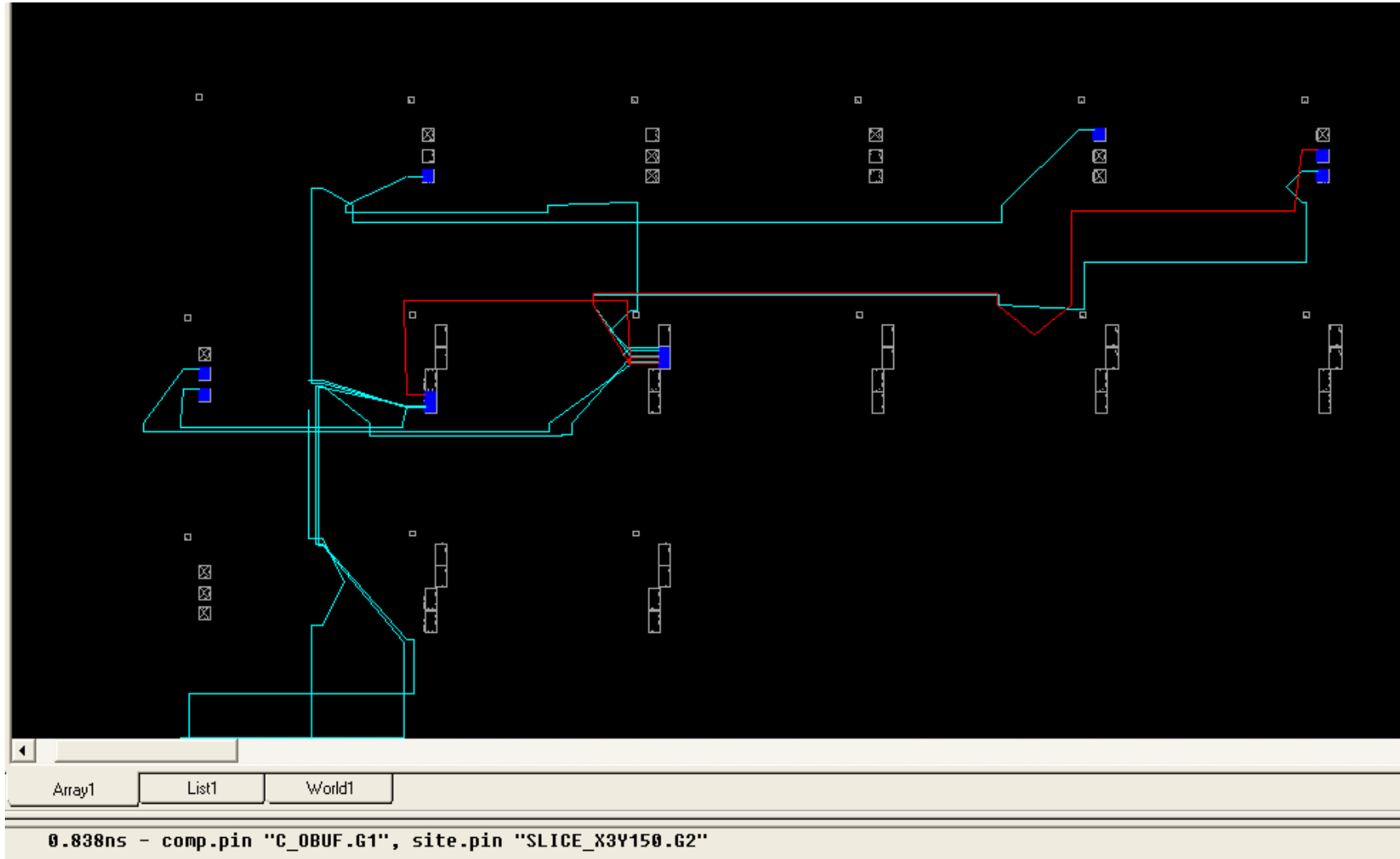


# Placing

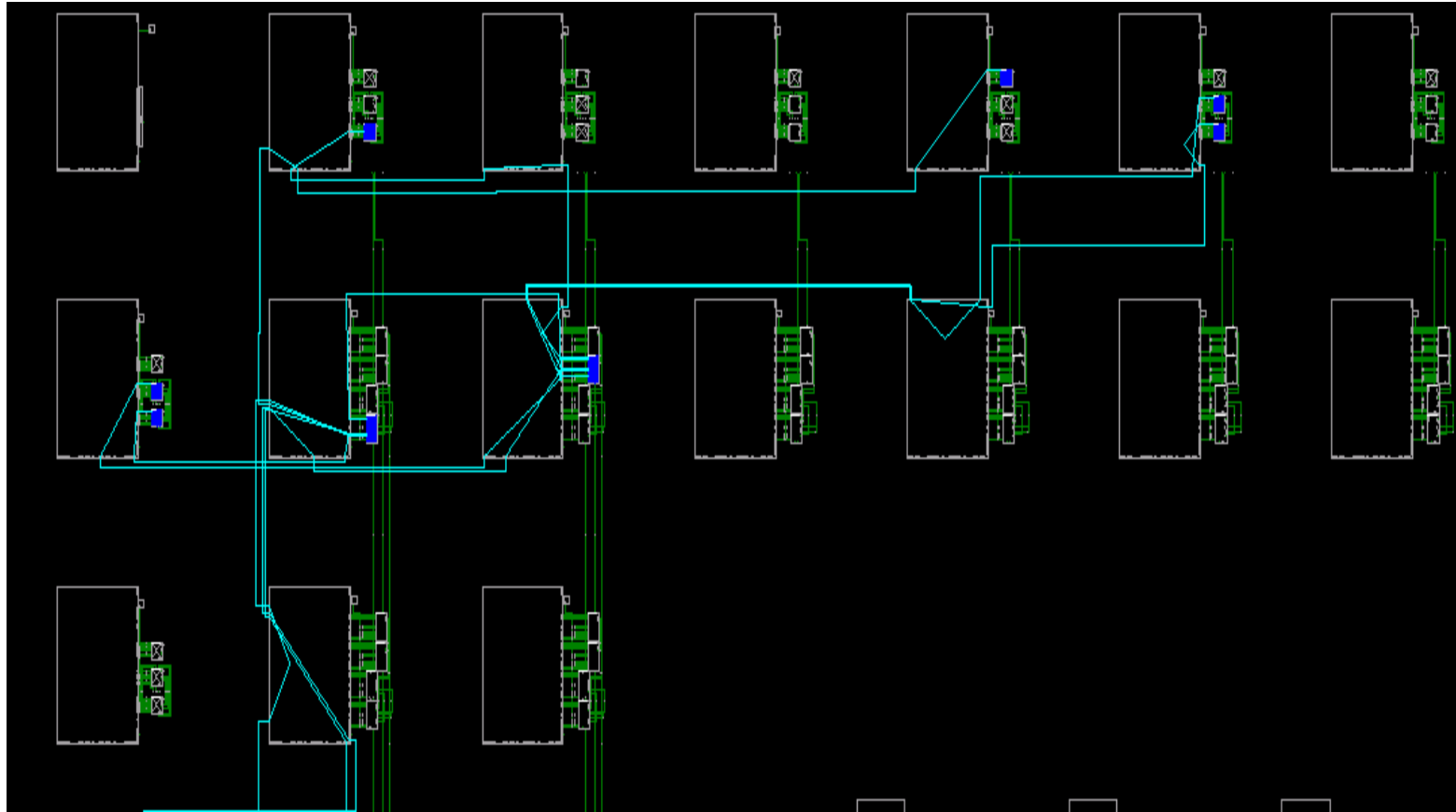




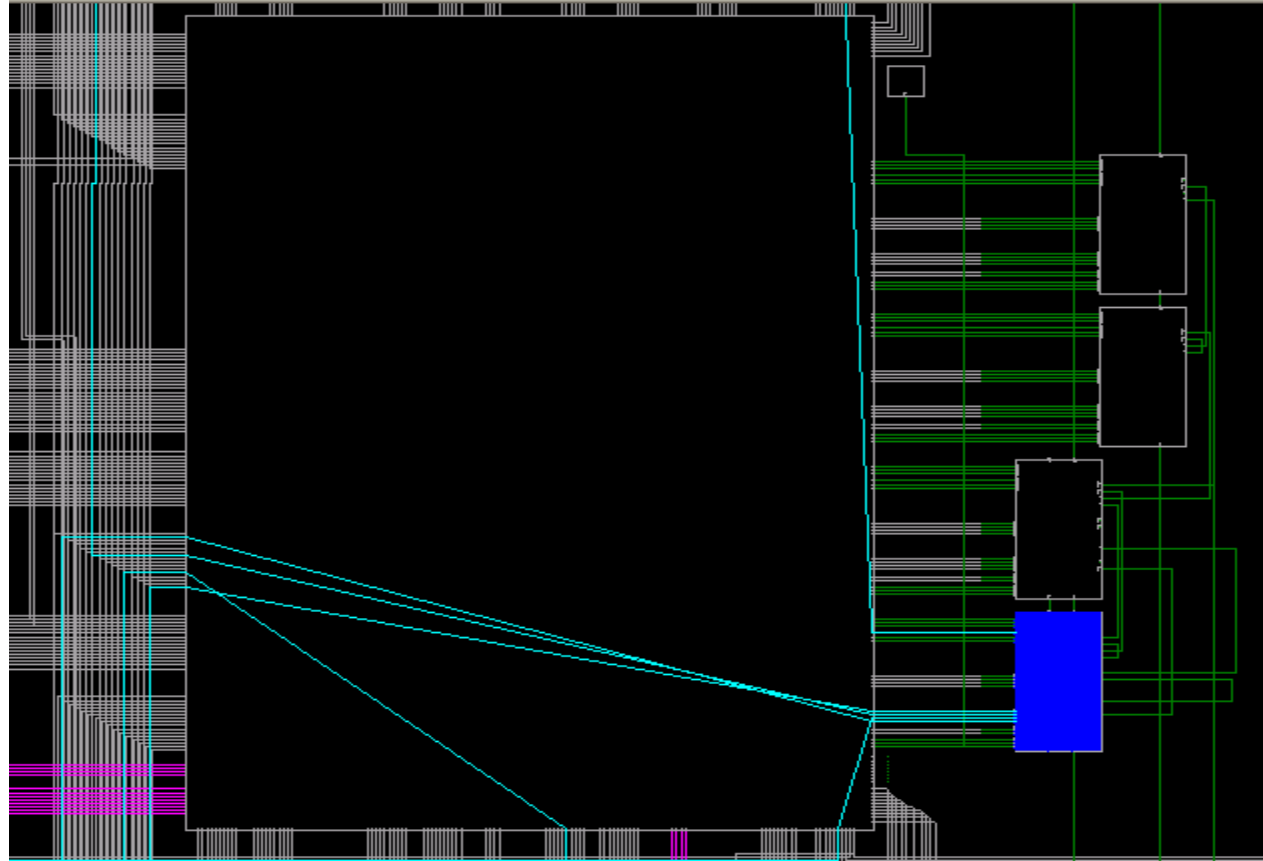
# routing



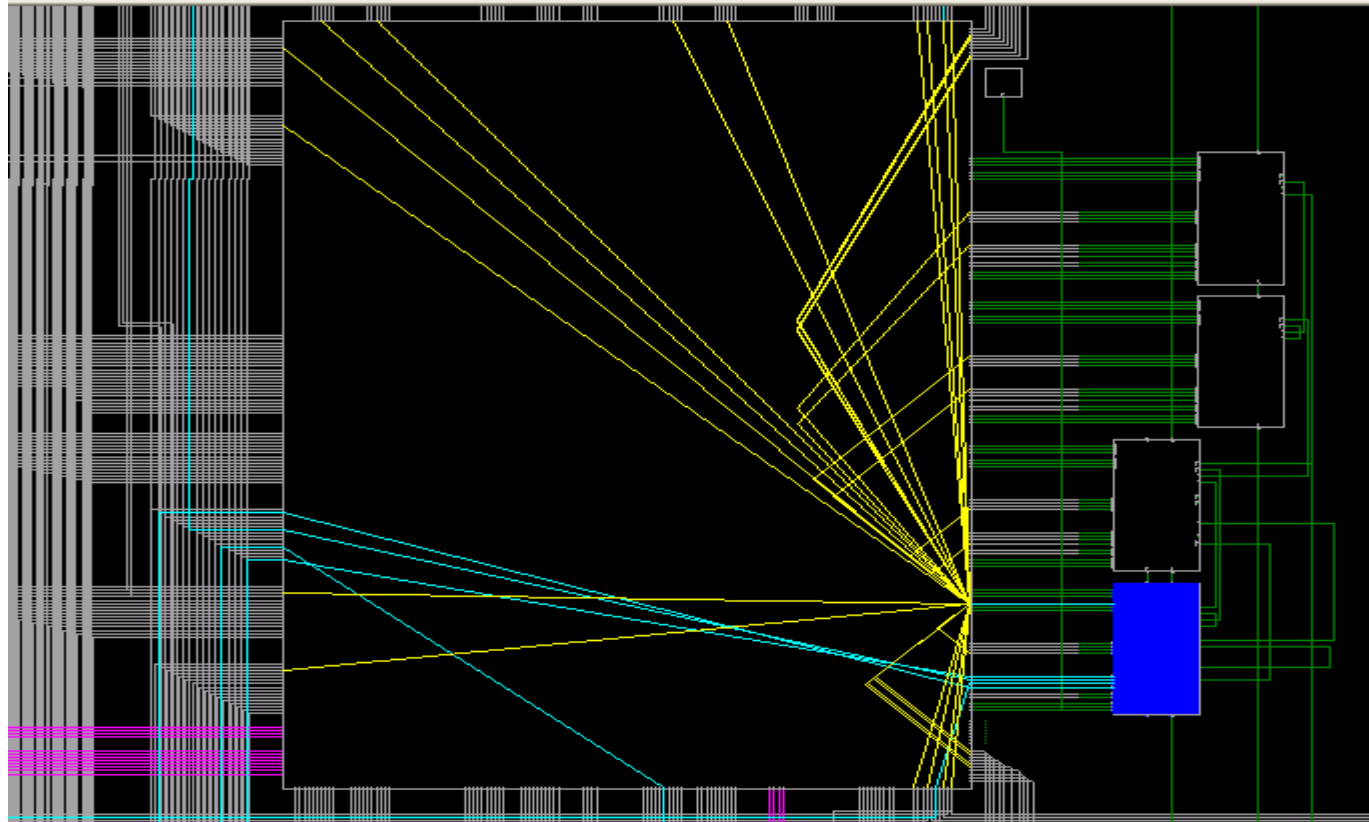
# routing



# Switching matrix

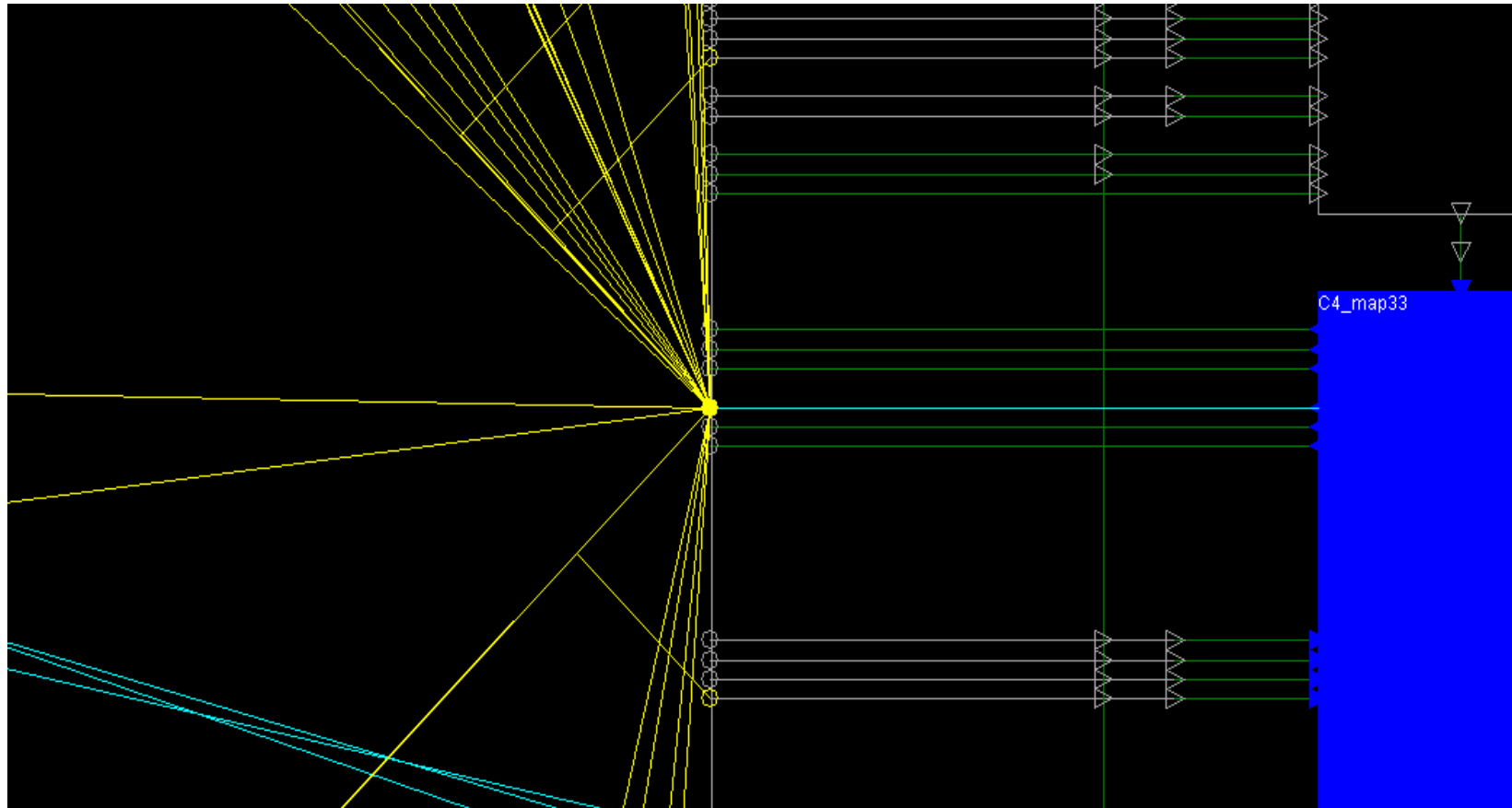


# Switching matrix-optional connections



# Switching matrix-optional connections

zoom in

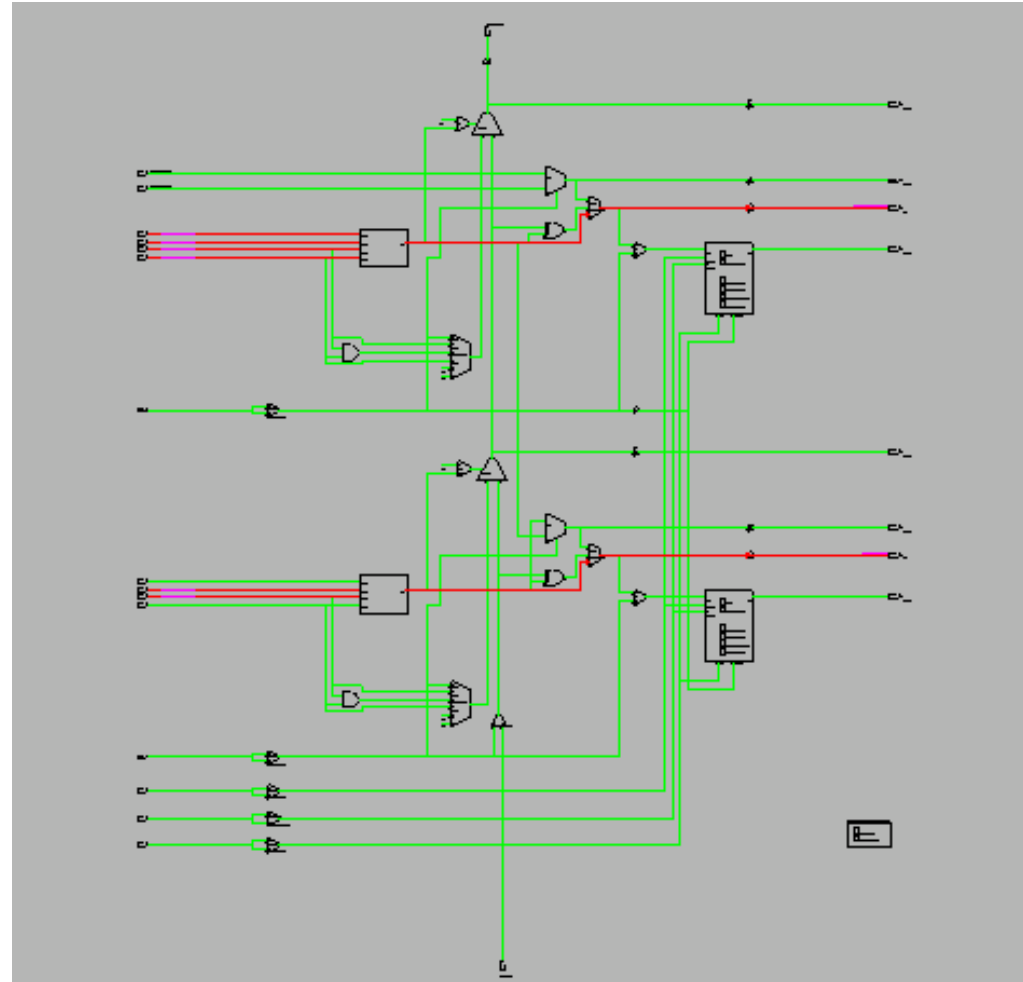


# Delay report

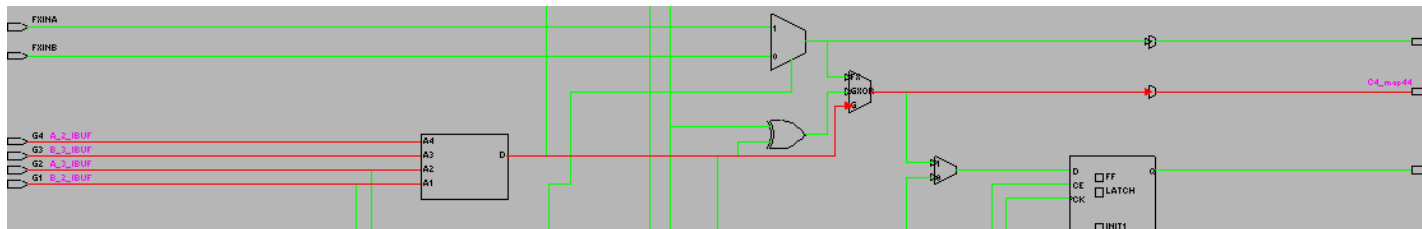
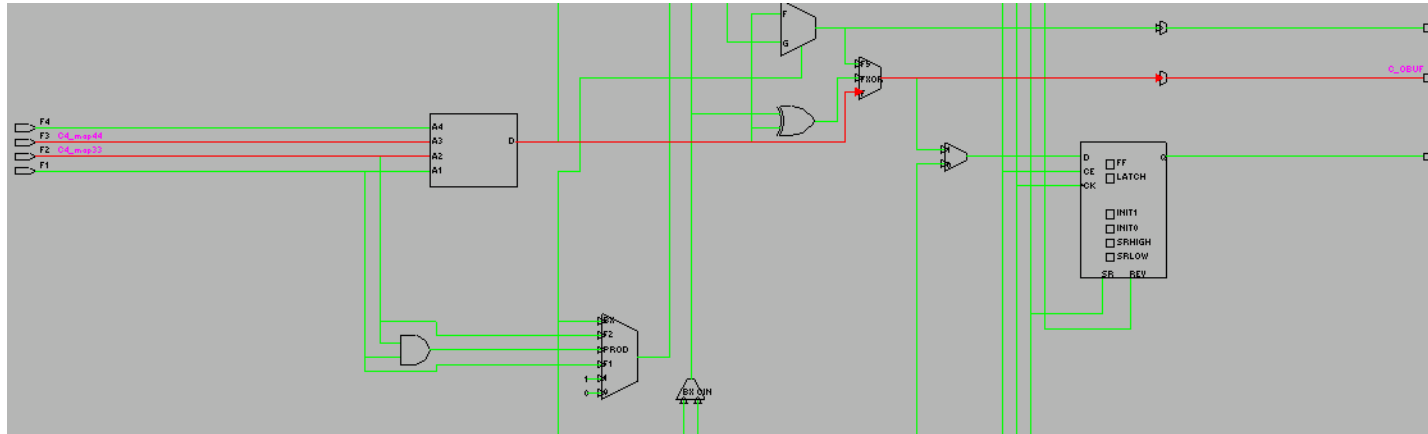
The 20 worst nets by delay are:

+-----+	
Max Delay   Net name	
+-----+	
0.838	A_3_IBUF
0.794	A_1_IBUF
0.770	B_3_IBUF
0.653	B_2_IBUF
0.551	A_2_IBUF
0.538	A_0_IBUF
0.513	C_OBUF
0.491	B_1_IBUF
0.372	C4_map33
0.328	B_0_IBUF
0.020	C4_map44

# Routing-zoom in



# Routing-zoom in





# RTL CODE

```
entity BH_MUX is
    Port ( A : in STD_LOGIC_VECTOR (5 downto 0);
          B : in STD_LOGIC_VECTOR (5 downto 0);
          C : out STD_LOGIC);
end BH_MUX;

architecture Behavioral of BH_MUX is

begin
    process(a,b)
    begin
        if a=b then
            c<='1';
        else
            c <='0';
        end if ;
    end process;

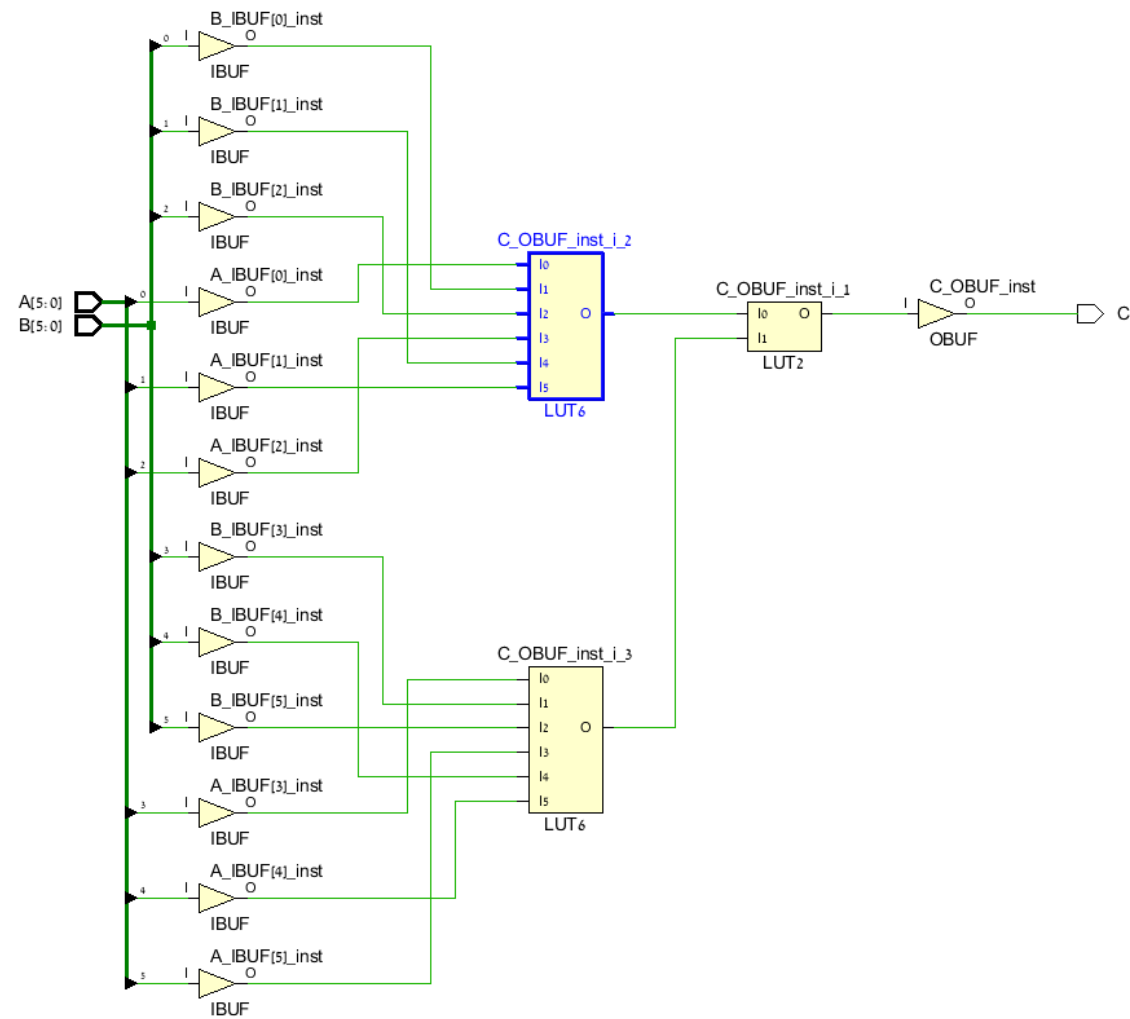
end Behavioral;
```

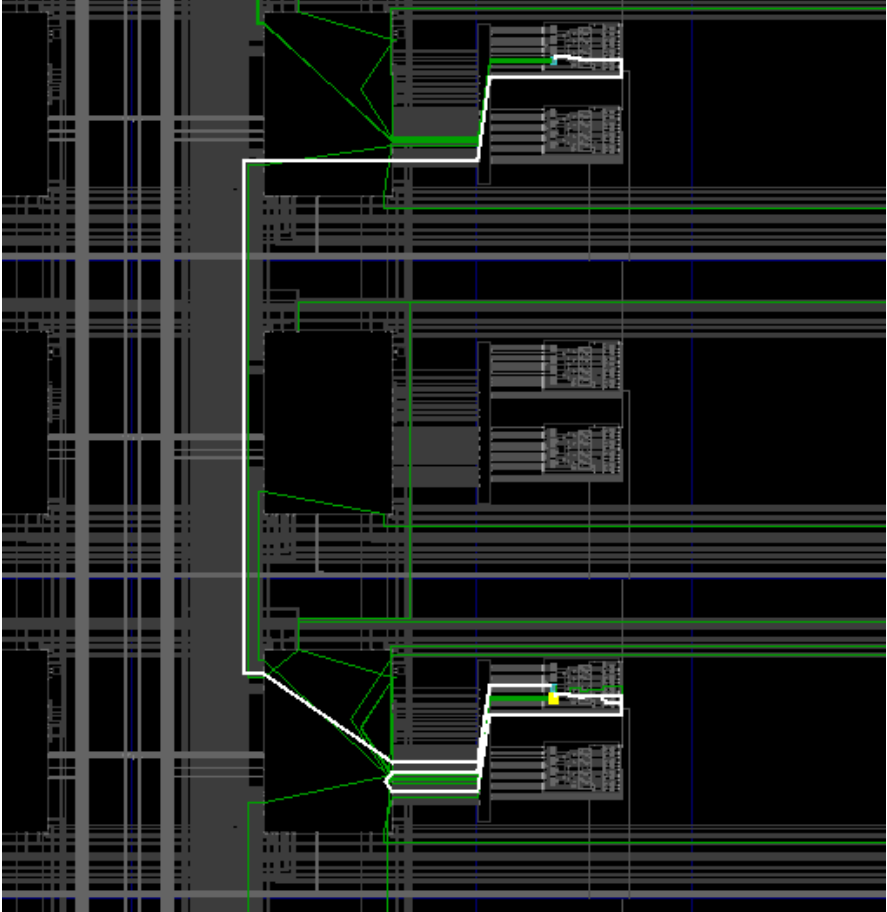
# Synthesis

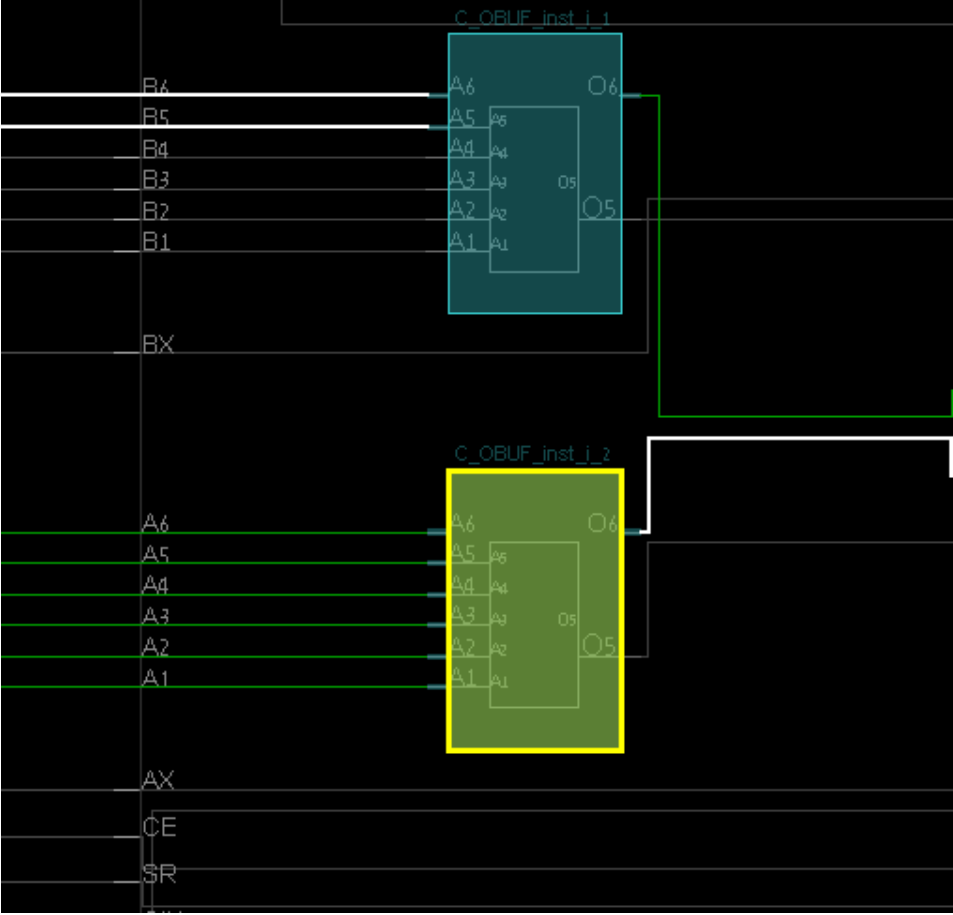
I5	I4	I3	I2	I1	I0	O=I0 & I1 & I2 & I3 & I4 & I5
0	0	0	0	0	0	1
0	0	0	0	0	1	0
0	0	0	0	1	0	0
0	0	0	0	1	1	1
0	0	0	1	0	0	0
0	0	0	1	0	1	0
0	0	0	1	1	0	0
0	0	0	1	1	1	0
0	0	1	0	0	0	0
0	0	1	0	0	1	0
0	0	1	0	1	0	0
0	0	1	0	1	1	0
0	0	1	1	0	0	1
0	0	1	1	0	1	0
0	0	1	1	1	0	0
0	0	1	1	1	1	1
0	1	0	0	0	0	0
0	1	0	0	0	1	0
0	1	0	0	1	0	0
0	1	0	0	1	1	0
0	1	0	1	0	0	0
0	1	0	1	0	1	0
0	1	0	1	1	0	0
0	1	0	1	1	1	0
0	1	1	0	0	0	0
0	1	1	0	0	1	0
0	1	1	0	1	1	0
0	1	1	1	0	0	0

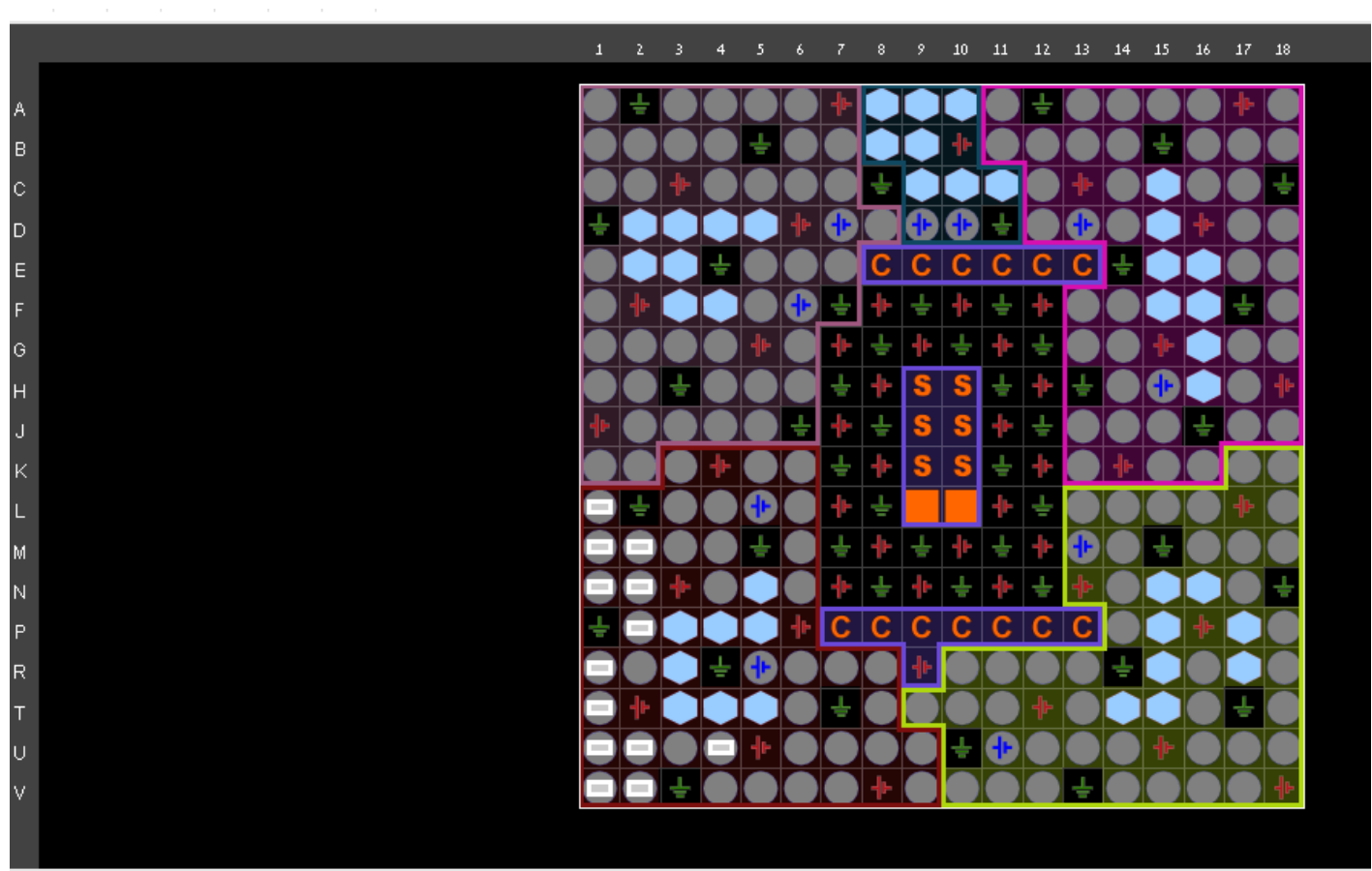
Edit LUT Equation...

ver Nets Cell Pins Truth Table

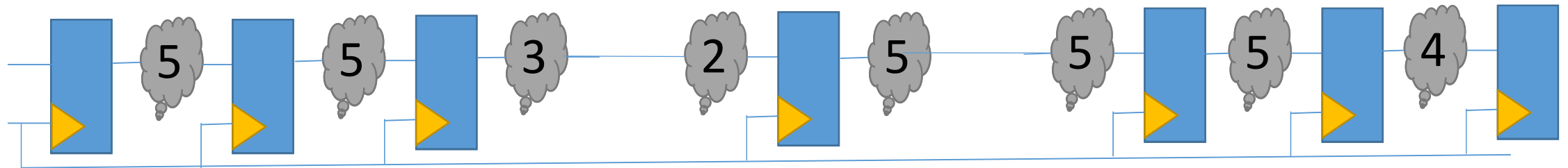






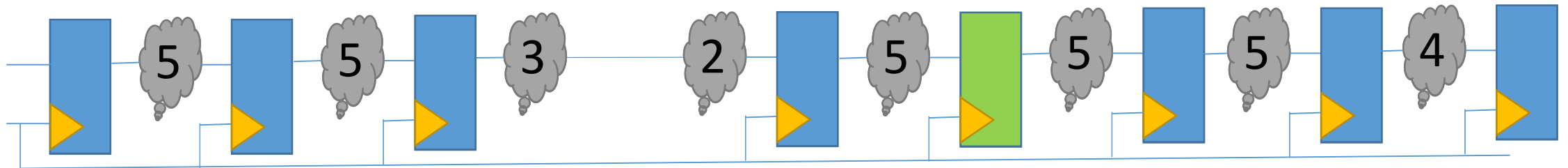


# RTL –Register Transfer Level

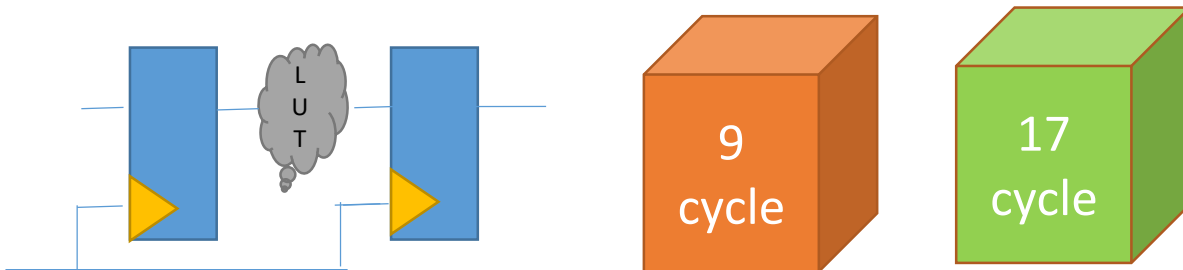
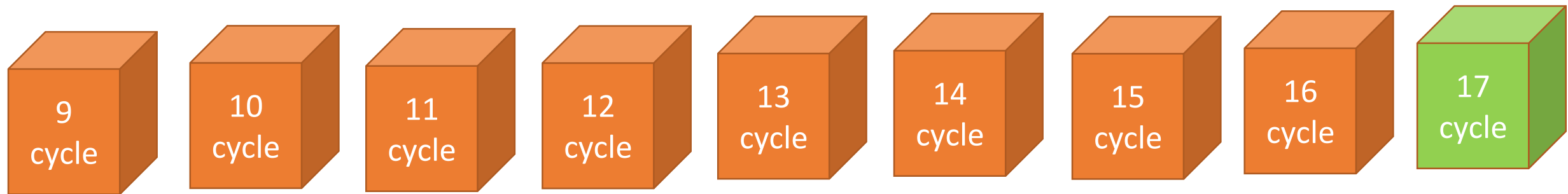
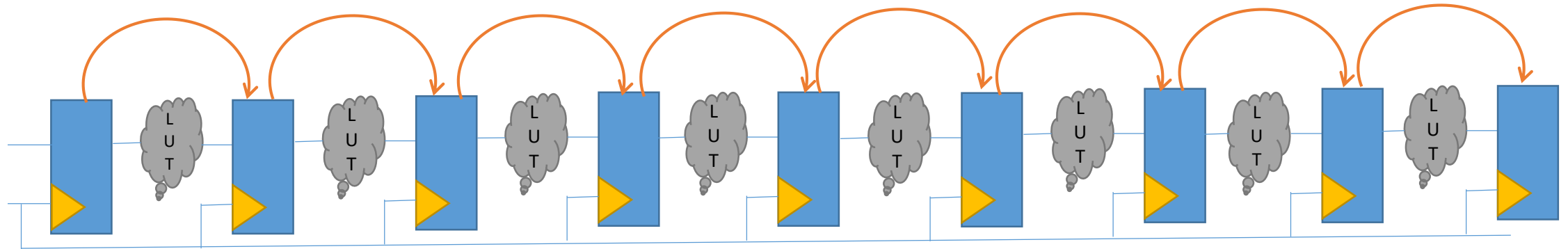


$F_{MAX} = 1 / \text{MAX DELAY} = 100 \text{ MHz}$

$F_{MAX} = 1 / \text{MAX DELAY} = 200 \text{ MHz}$



# PARALLEL PROCESSING VERSUS SERIAL PROCESSING



# Create project



## Quick Start

- [Create Project >](#)
- [Open Project >](#)
- [Open Example Project >](#)

## Tasks

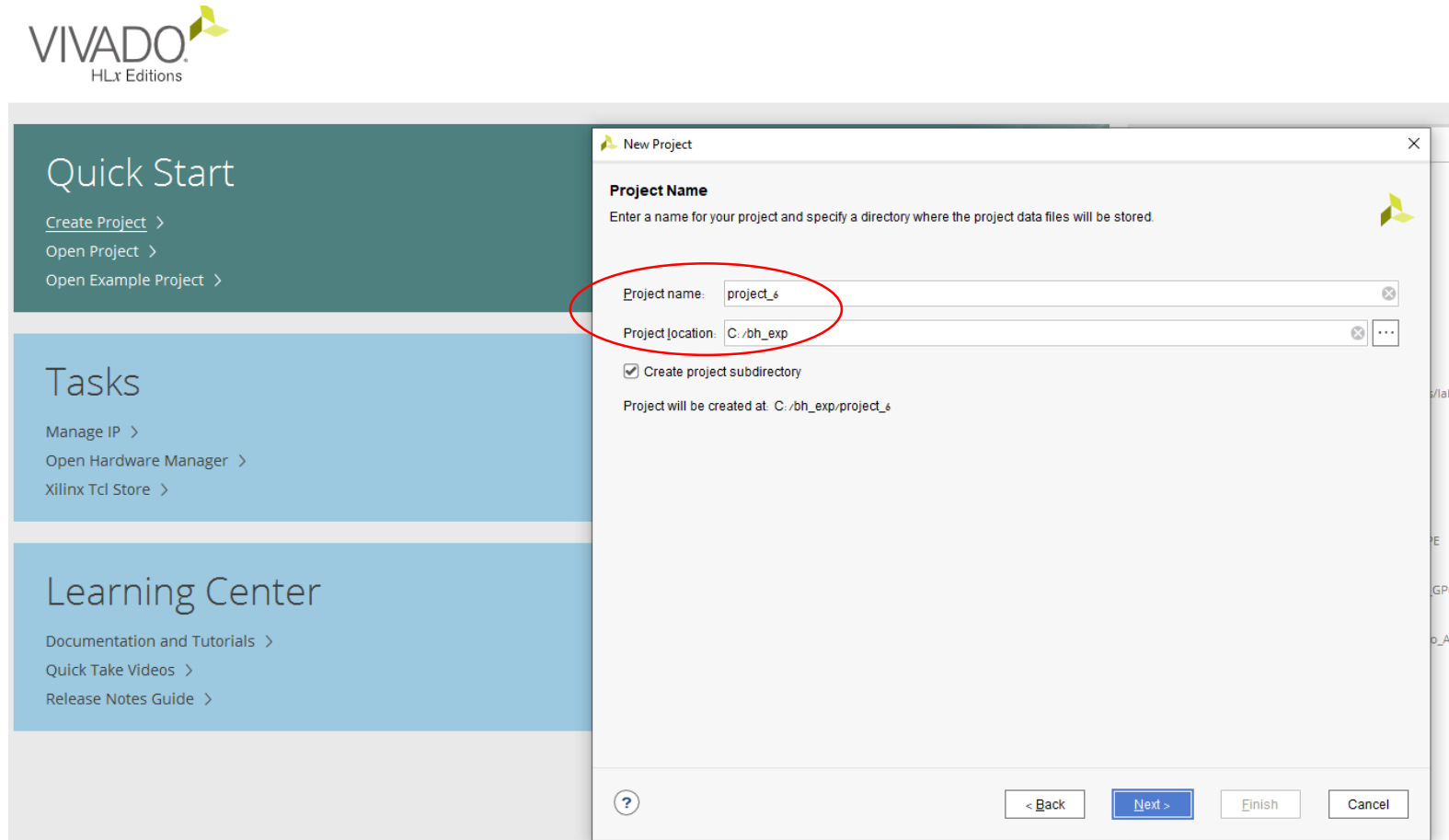
- [Manage IP >](#)
- [Open Hardware Manager >](#)
- [Xilinx Tcl Store >](#)

## Learning Center

- [Documentation and Tutorials >](#)
- [Quick Take Videos >](#)
- [Release Notes Guide >](#)



# Create project –name & path not in Hebrew!!



# Create project –RTL project



## Quick Start

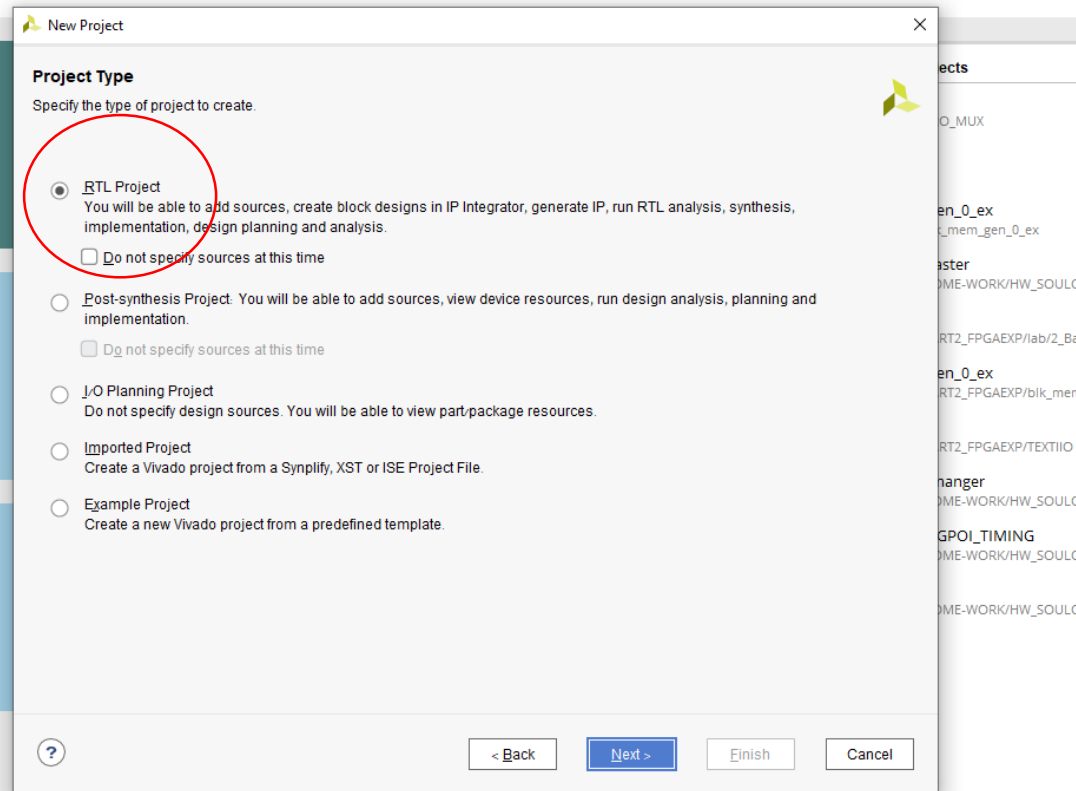
- [Create Project >](#)
- [Open Project >](#)
- [Open Example Project >](#)

## Tasks

- [Manage IP >](#)
- [Open Hardware Manager >](#)
- [Xilinx Tcl Store >](#)

## Learning Center

- [Documentation and Tutorials >](#)
- [Quick Take Videos >](#)
- [Release Notes Guide >](#)



The image shows the 'New Project' dialog box in Vivado. The 'Project Type' section is titled 'Specify the type of project to create.' and lists five options: 'RTL Project', 'Post-synthesis Project', 'I/O Planning Project', 'Imported Project', and 'Example Project'. The 'RTL Project' option is selected, indicated by a red circle around the radio button. Below the 'RTL Project' option, there is a checkbox labeled 'Do not specify sources at this time' which is also checked. The other options are not selected. At the bottom of the dialog, there are four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'. The 'Next >' button is highlighted in blue.

**New Project**

**Project Type**  
Specify the type of project to create.

- ☒ **RTL Project**  
You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.  
☒ Do not specify sources at this time
- ☐ **Post-synthesis Project**  
You will be able to add sources, view device resources, run design analysis, planning and implementation.  
☐ Do not specify sources at this time
- ☐ **I/O Planning Project**  
Do not specify design sources. You will be able to view part/package resources.
- ☐ **Imported Project**  
Create a Vivado project from a Synplify, XST or ISE Project File.
- ☐ **Example Project**  
Create a new Vivado project from a predefined template.

[? < Back](#) [Next >](#) [Finish](#) [Cancel](#)

# Create project –FPGA choosing

New Project

**Default Part**  
Choose a default Xilinx part or board for your project.

Parts | Boards

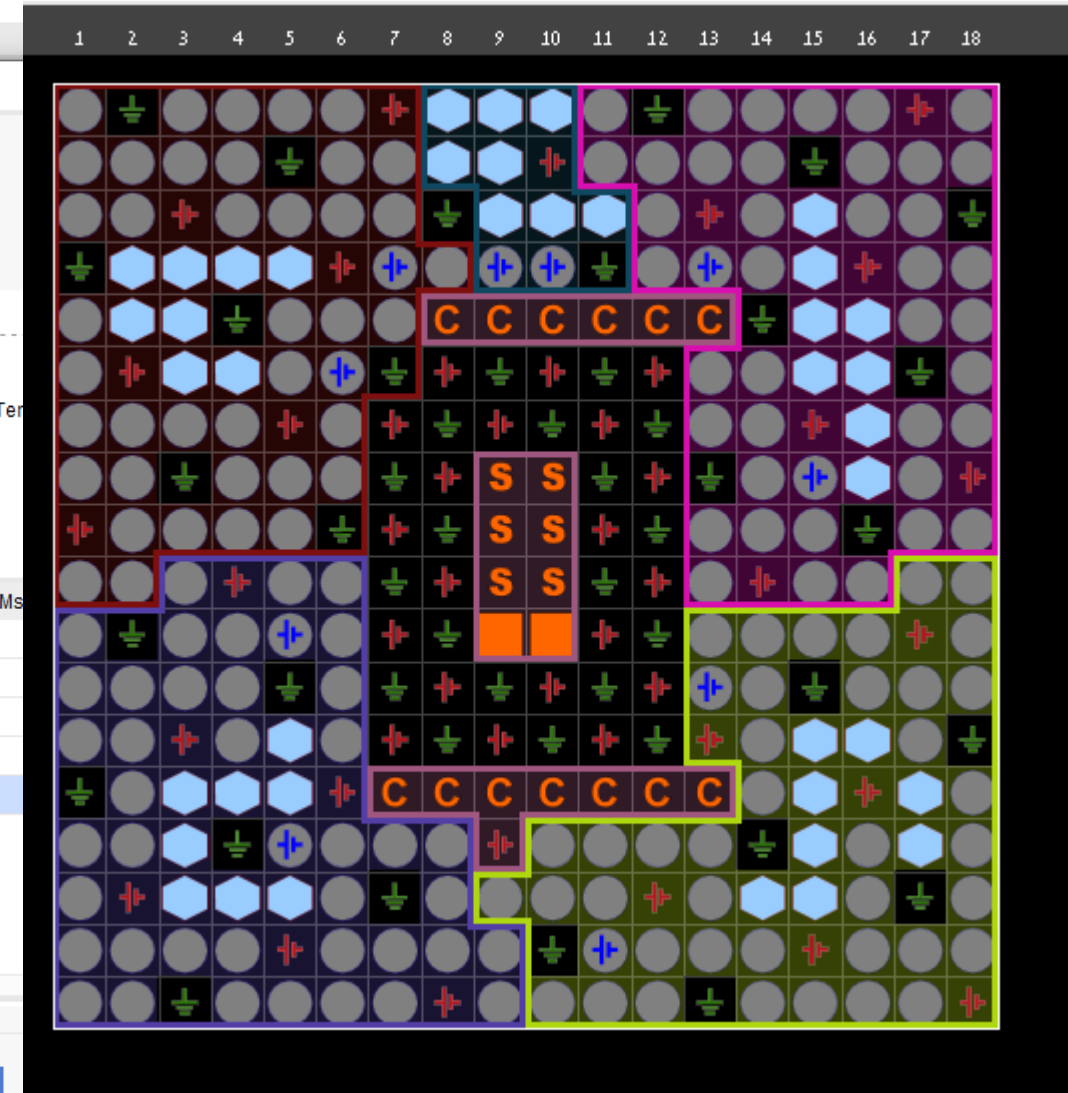
[Reset All Filters](#)

Category: General Purpose Package: csg324  
Family: Artix-7 Speed: -1

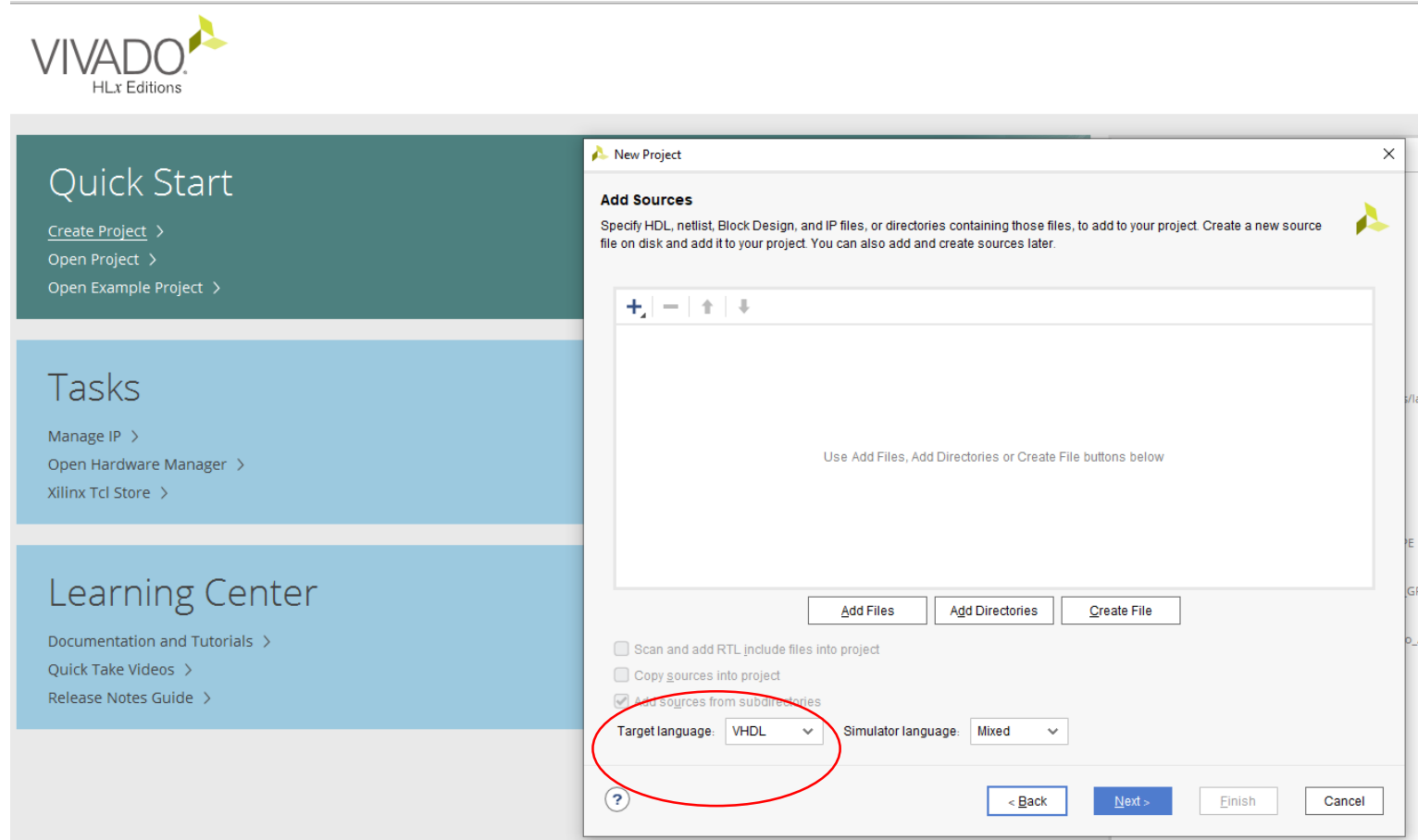
Search: Q-

Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs
xc7a15tcs324-1	324	210	10400	20800	25
xc7a35tcs324-1	324	210	20800	41600	50
xc7a50tcs324-1	324	210	32600	65200	75
xc7a75tcs324-1	324	210	47200	94400	105
xc7a100tcs324-1	324	210	63400	126800	135

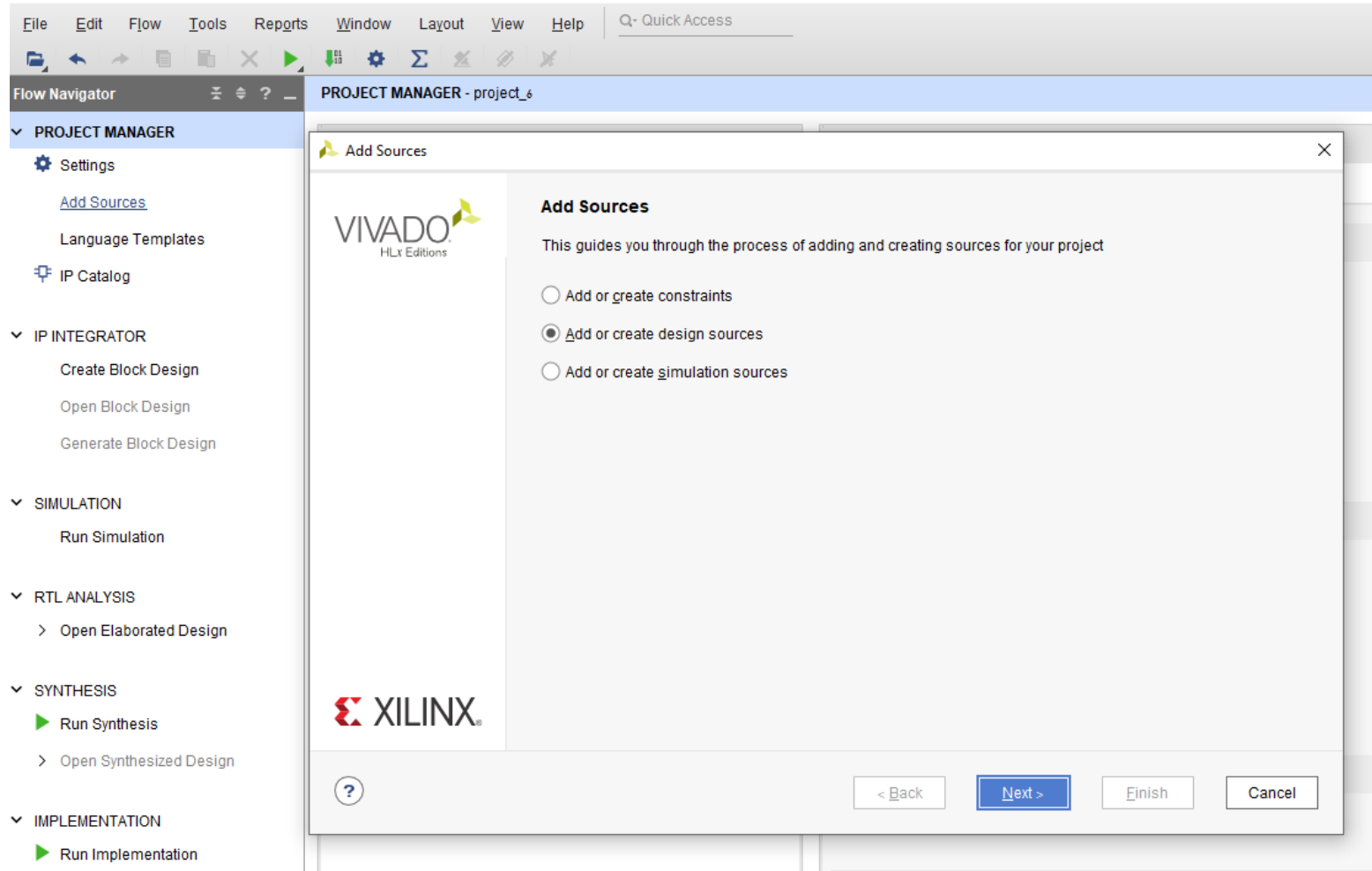
< Back Next >



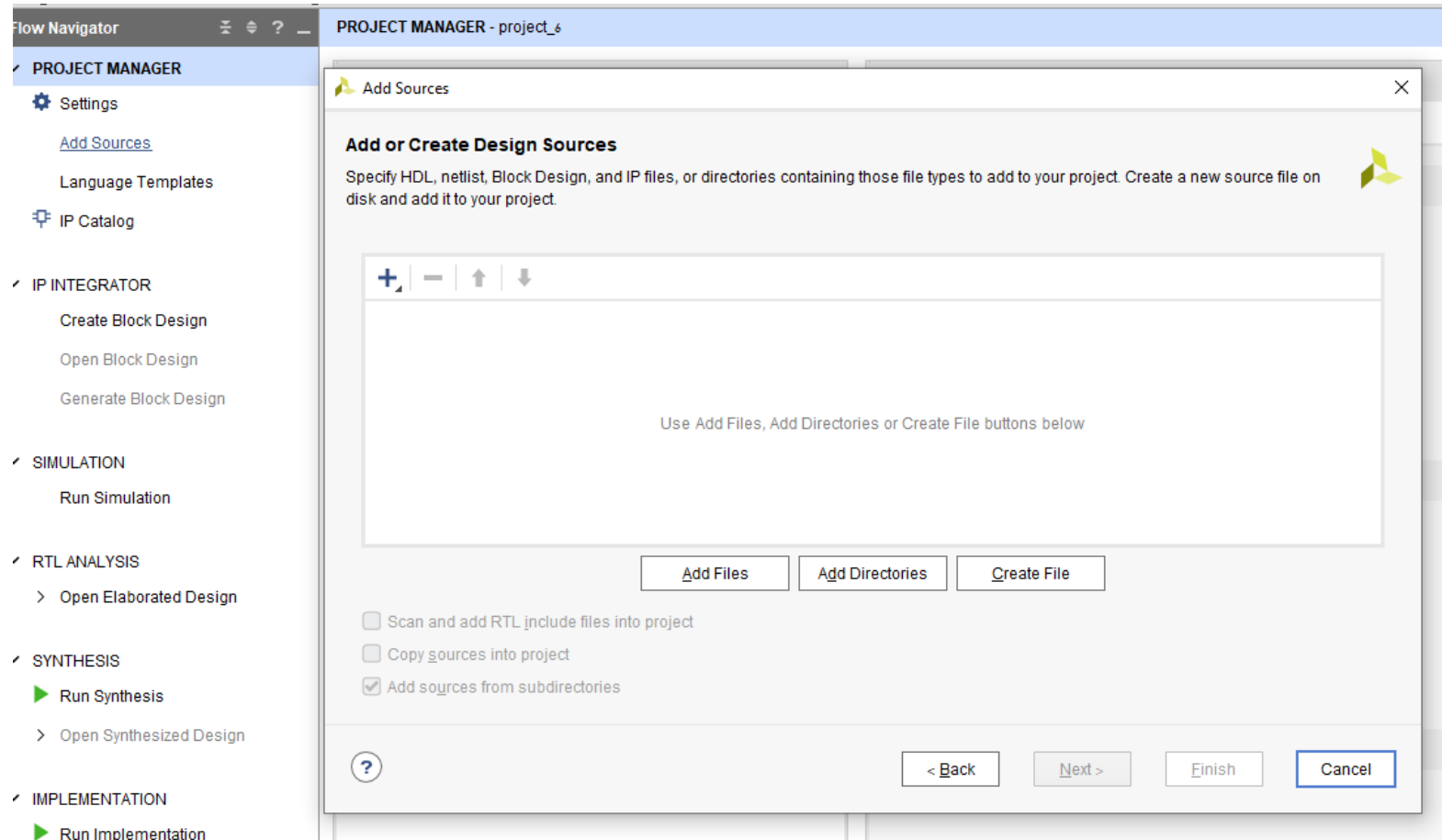
# Create project –next ...next .next...



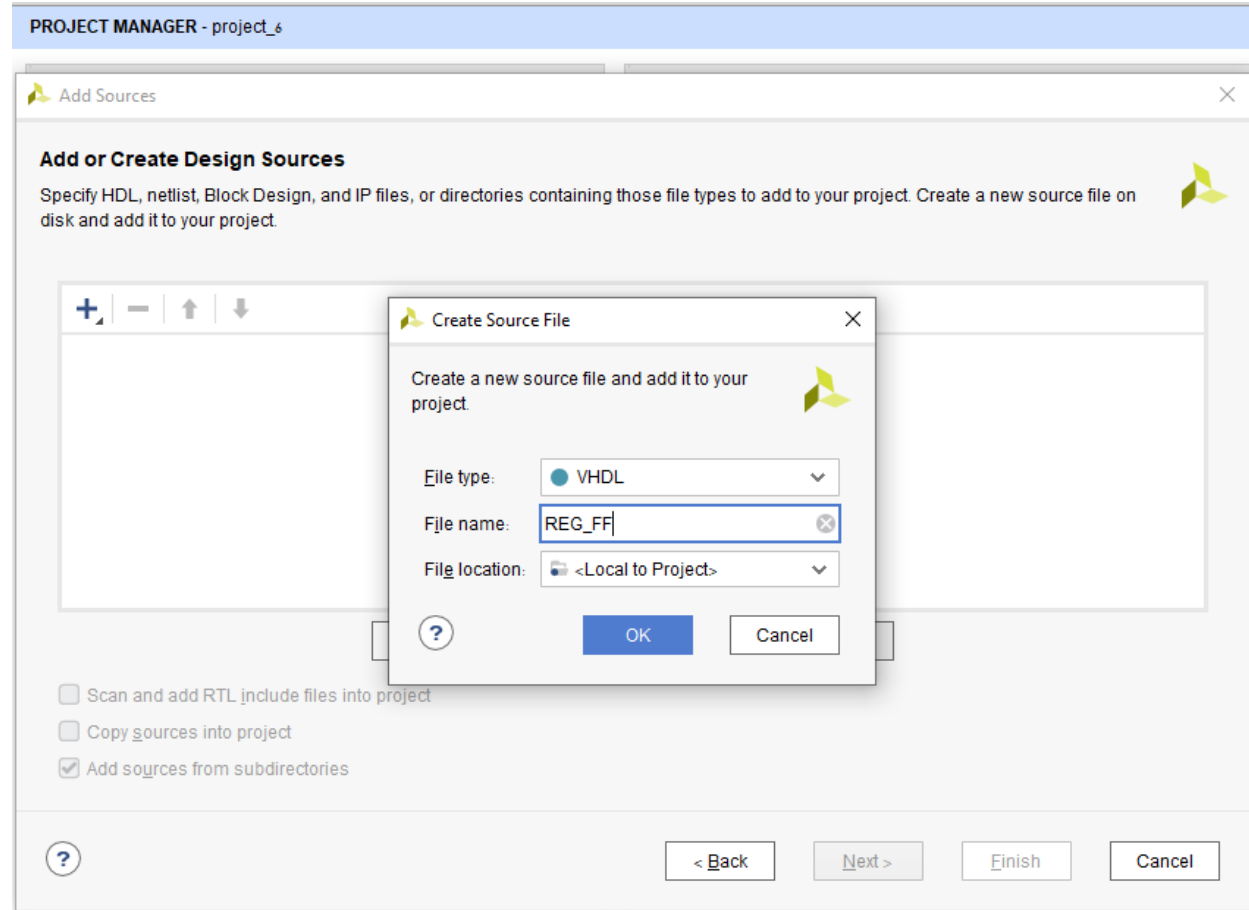
# Adding source



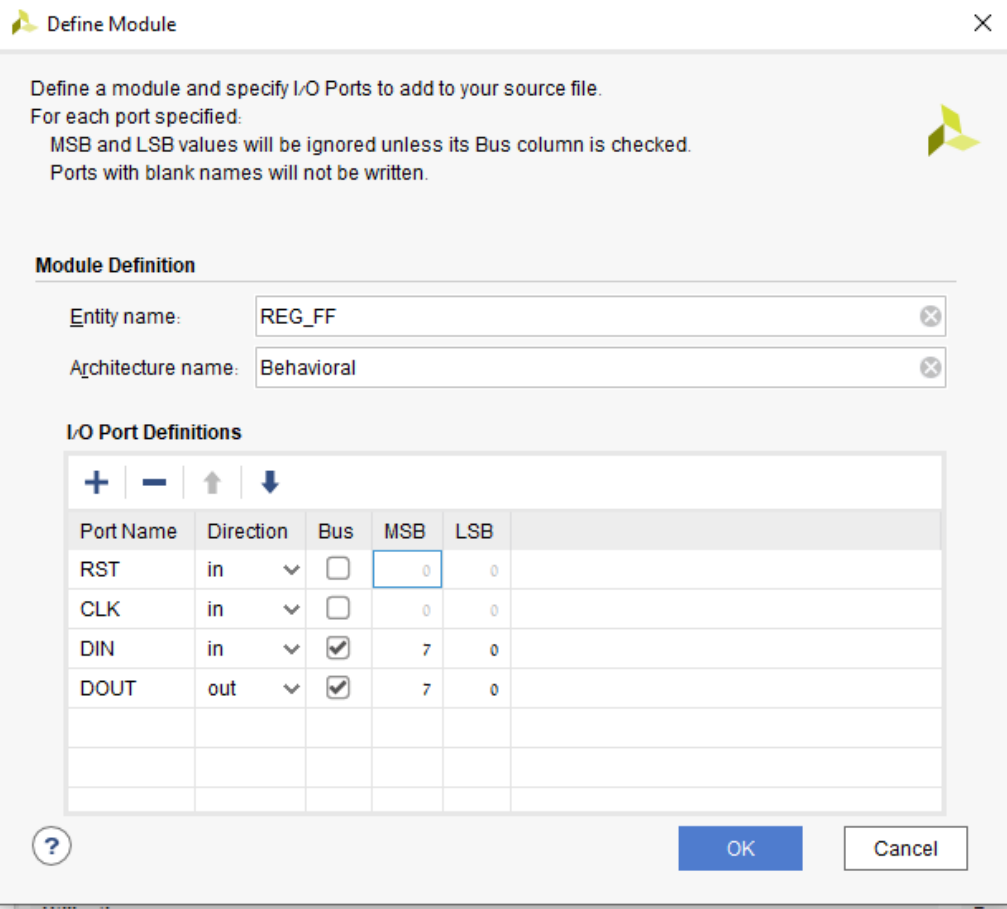
# Create files



# Create sources files



# Create sources files –Entity Definition



Define a module and specify I/O Ports to add to your source file.  
For each port specified:  
MSB and LSB values will be ignored unless its Bus column is checked.  
Ports with blank names will not be written.

**Module Definition**

Entity name: REG\_FF

Architecture name: Behavioral

**I/O Port Definitions**

+ - ↑ ↓

Port Name	Direction		Bus	MSB	LSB
RST	in	▼	<input type="checkbox"/>	0	0
CLK	in	▼	<input type="checkbox"/>	0	0
DIN	in	▼	<input checked="" type="checkbox"/>	7	0
DOUT	out	▼	<input checked="" type="checkbox"/>	7	0

? OK Cancel

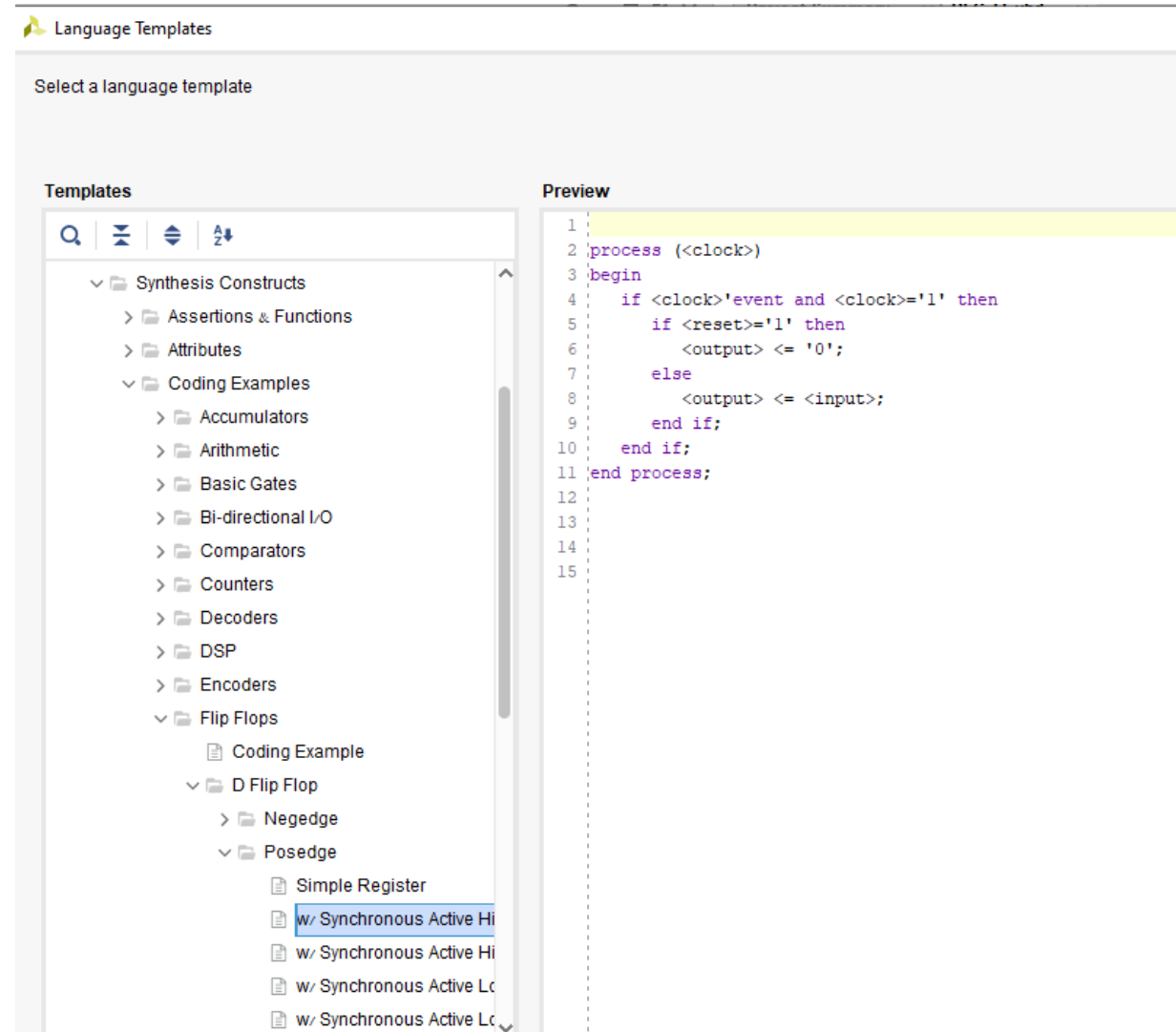
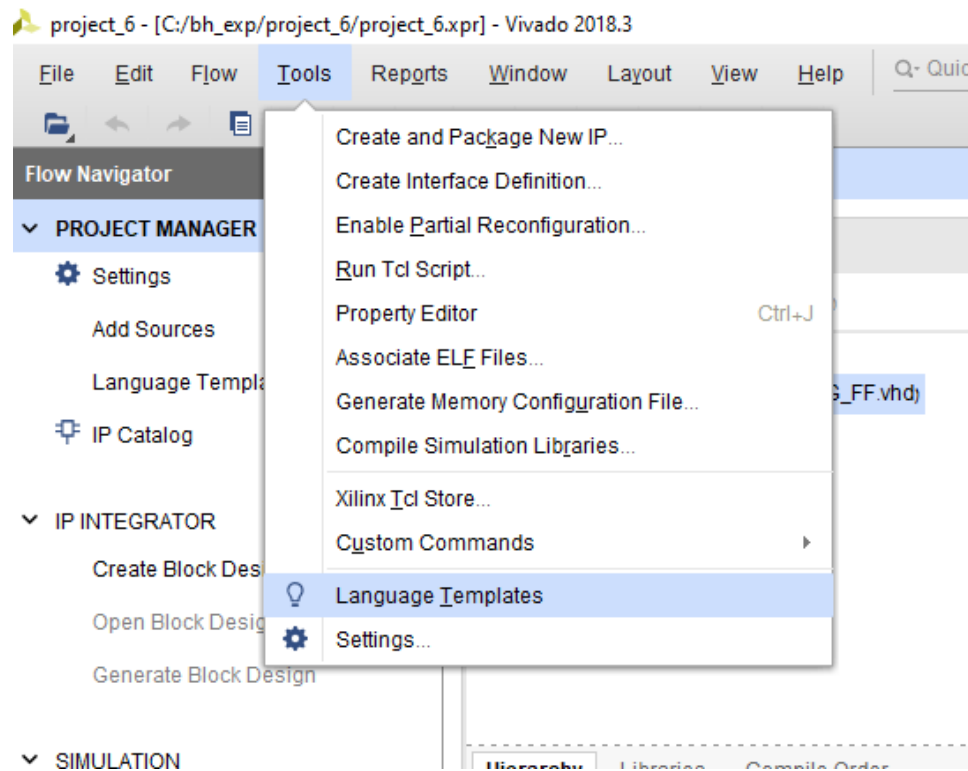


# Create sources files –Entity

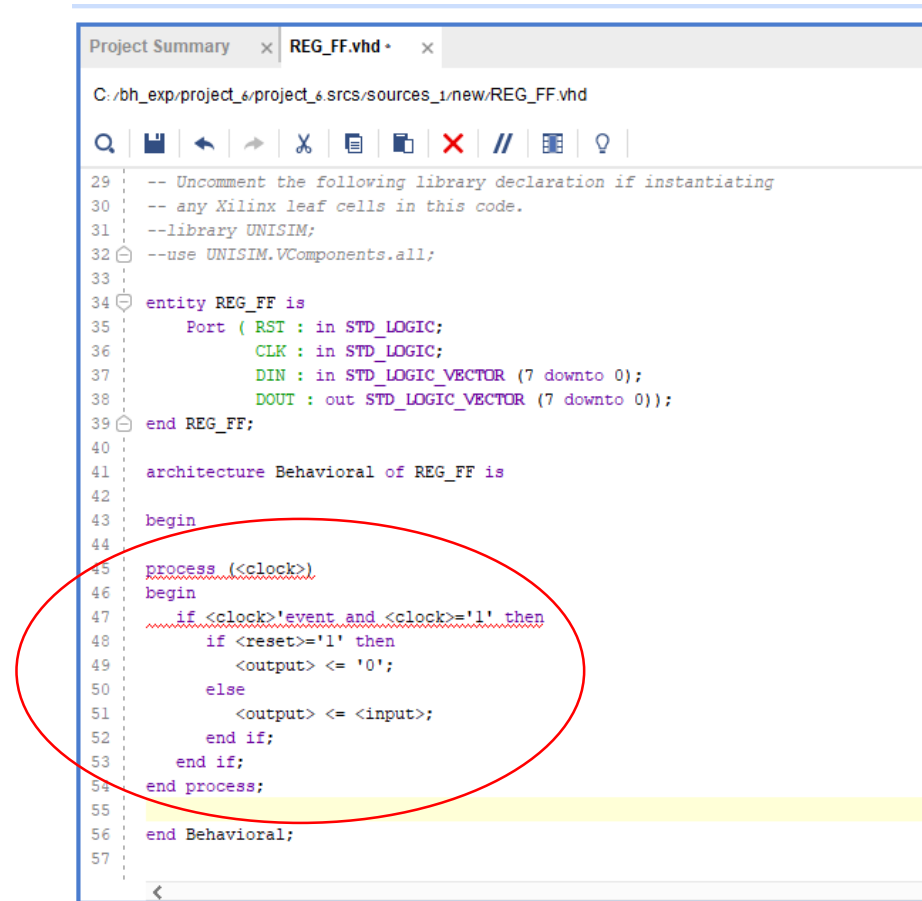
The screenshot displays the Xilinx IDE interface with the following components:

- Flow Navigator (Left Panel):** Contains project management tasks such as Settings, Add Sources, Language Templates, IP Catalog, IP INTEGRATOR (Create/Open/Generate Block Design), SIMULATION (Run Simulation), RTL ANALYSIS (Open Elaborated Design), SYNTHESIS (Run/Open Synthesized Design), and IMPLEMENTATION (Run/Open Implemented Design).
- PROJECT MANAGER - project\_6 (Top Panel):** Shows the project structure with folders for Design Sources, Constraints, Simulation Sources, and Utility Sources. The file **REG\_FF(Behavioral) (REG\_FF.vhd)** is highlighted under Design Sources.
- Source File Properties (Bottom Panel):** Displays details for the selected file:
  - Enabled:** Checked
  - Location:** C:/bh\_exp/project\_6/project\_6.srcs/sources\_1/new
  - Type:** VHDL
  - Library:** xil\_defaultlib
  - Size:** 1.0 KB
  - Modified:** Today at 13:06:23 PM
- REG\_FF.vhd (Right Panel):** Shows the VHDL code for the entity. The code includes:
  - Revision 0.01 - File Created
  - Library IEEE and use IEEE.STD\_LOGIC\_1164.ALL;
  - Uncommented library declarations for IEEE.NUMERIC\_STD.ALL and UNISIM.VComponents.all.
  - Entity declaration: **entity REG\_FF is**
  - Port declarations: **Port ( RST : in STD\_LOGIC; CLK : in STD\_LOGIC; DIN : in STD\_LOGIC\_VECTOR (7 downto 0); DOUT : out STD\_LOGIC\_VECTOR (7 downto 0));**
  - End entity: **end REG\_FF;**
  - Architecture declaration: **architecture Behavioral of REG\_FF is** (highlighted in yellow)
  - Begin statement: **begin**

# Using Language templates –stage 1



# Using Language templates –stage 2



```
29  -- Uncomment the following library declaration if instantiating
30  -- any Xilinx leaf cells in this code.
31  --library UNISIM;
32  --use UNISIM.VComponents.all;
33
34  entity REG_FF is
35      Port ( RST : in STD_LOGIC;
36            CLK : in STD_LOGIC;
37            DIN : in STD_LOGIC_VECTOR (7 downto 0);
38            DOUT : out STD_LOGIC_VECTOR (7 downto 0));
39  end REG_FF;
40
41  architecture Behavioral of REG_FF is
42
43  begin
44
45      process (<clock>),
46      begin
47          if <clock>'event and <clock>='1' then
48              if <reset>='1' then
49                  <output> <= '0';
50              else
51                  <output> <= <input>;
52              end if;
53          end if;
54      end process;
55
56  end Behavioral;
57
```

# Using Language templates –stage 3

```
Project Summary x REG_FF.vhd x
C:/bh_exp/project_6/project_6.srscs/sources_1/new/REG_FF.vhd

29 -- Uncomment the following library
30 -- any Xilinx leaf cells in this
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33
34 entity REG_FF is
35     Port ( RST : in STD_LOGIC;
36           CLK : in STD_LOGIC;
37           DIN : in STD_LOGIC_VECTOR (7 downto 0);
38           DOUT : out STD_LOGIC_VECTOR (7 downto 0));
39 end REG_FF;
40
41 architecture Behavioral of REG_FF is
42
43 begin
44
45     process (<clock>)
46     begin
47         if <clock>'event and <clock>='1' then
48             if <reset>='1' then
49                 <output> <= '0';
50             else
51                 <output> <= <input>;
52             end if;
53         end if;
54     end process;
55
56 end Behavioral;
57
```

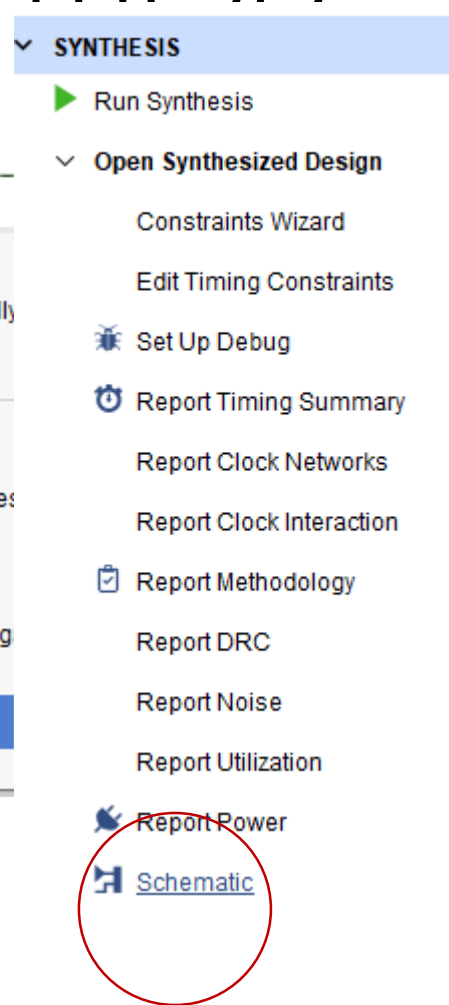
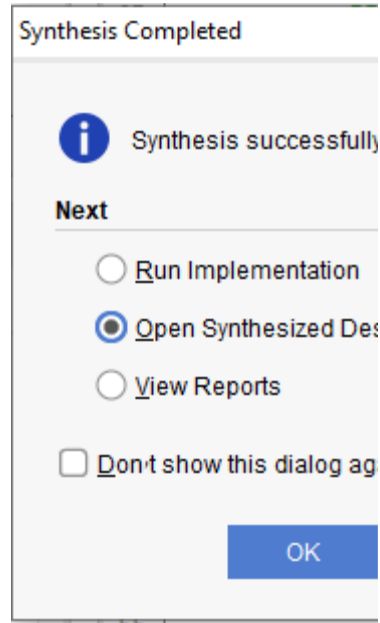
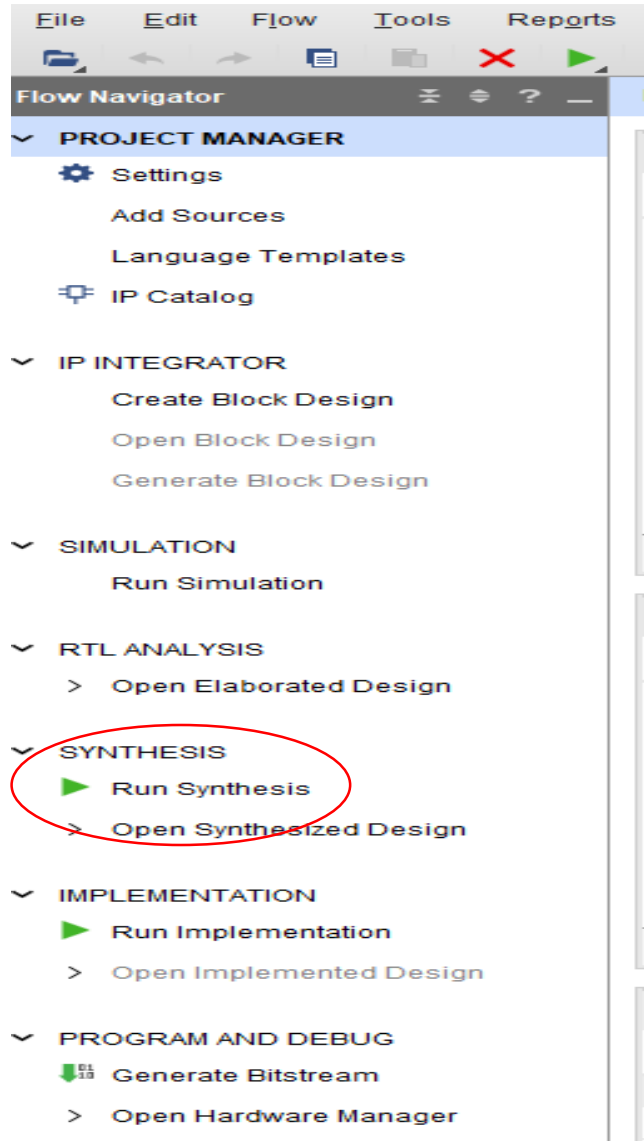
```
Project Summary x REG_FF.vhd x
C:/bh_exp/project_6/project_6.srscs/sources_1/new/REG_FF.vhd

29 -- Uncomment the following library declarat
30 -- any Xilinx leaf cells in this code.
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33
34 entity REG_FF is
35     Port ( RST : in STD_LOGIC;
36           CLK : in STD_LOGIC;
37           DIN : in STD_LOGIC_VECTOR (7 downto 0);
38           DOUT : out STD_LOGIC_VECTOR (7 downto 0));
39 end REG_FF;
40
41 architecture Behavioral of REG_FF is
42
43 begin
44
45     process (CLK)
46     begin
47         if CLK'event and CLK='1' then
48             if RST='1' then
49                 DOUT <= '0';
50             else
51                 DOUT <= DIN ;
52             end if;
53         end if;
54     end process;
55
56 end Behavioral;
57
```

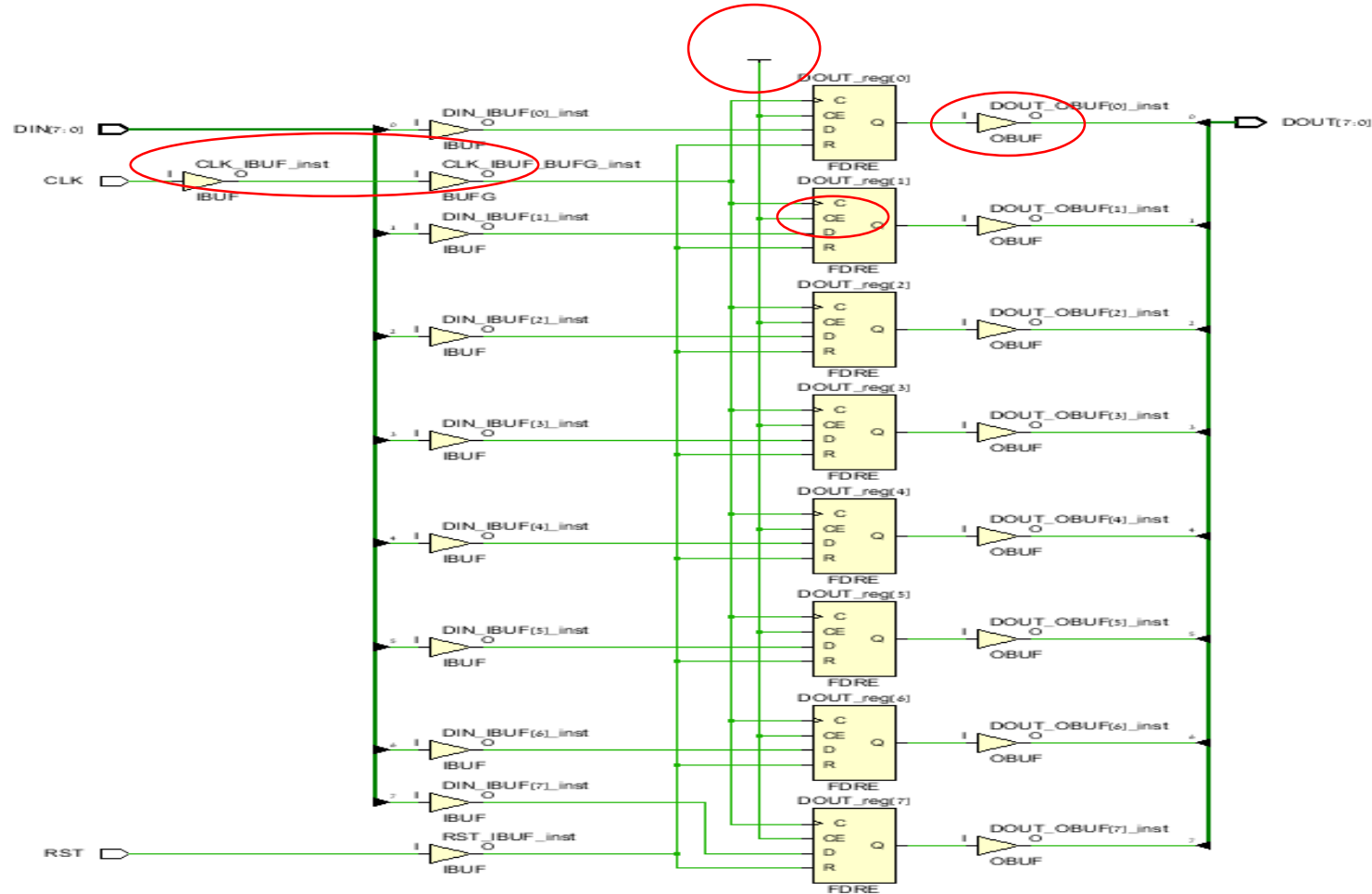
```
Project Summary x REG_FF.vhd x
C:/bh_exp/project_6/project_6.srscs/sources_1/new/REG_FF.vhd

29 -- Uncomment the following library declaration if instantiating
30 -- any Xilinx leaf cells in this code.
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33
34 entity REG_FF is
35     Port ( RST : in STD_LOGIC;
36           CLK : in STD_LOGIC;
37           DIN : in STD_LOGIC_VECTOR (7 downto 0);
38           DOUT : out STD_LOGIC_VECTOR (7 downto 0));
39 end REG_FF;
40
41 architecture Behavioral of REG_FF is
42
43 begin
44
45     process (CLK)
46     begin
47         if CLK'event and CLK='1' then
48             if RST='1' then
49                 DOUT <= (OTHERS=>'0');
50             else
51                 DOUT <= DIN ;
52             end if;
53         end if;
54     end process;
55
```

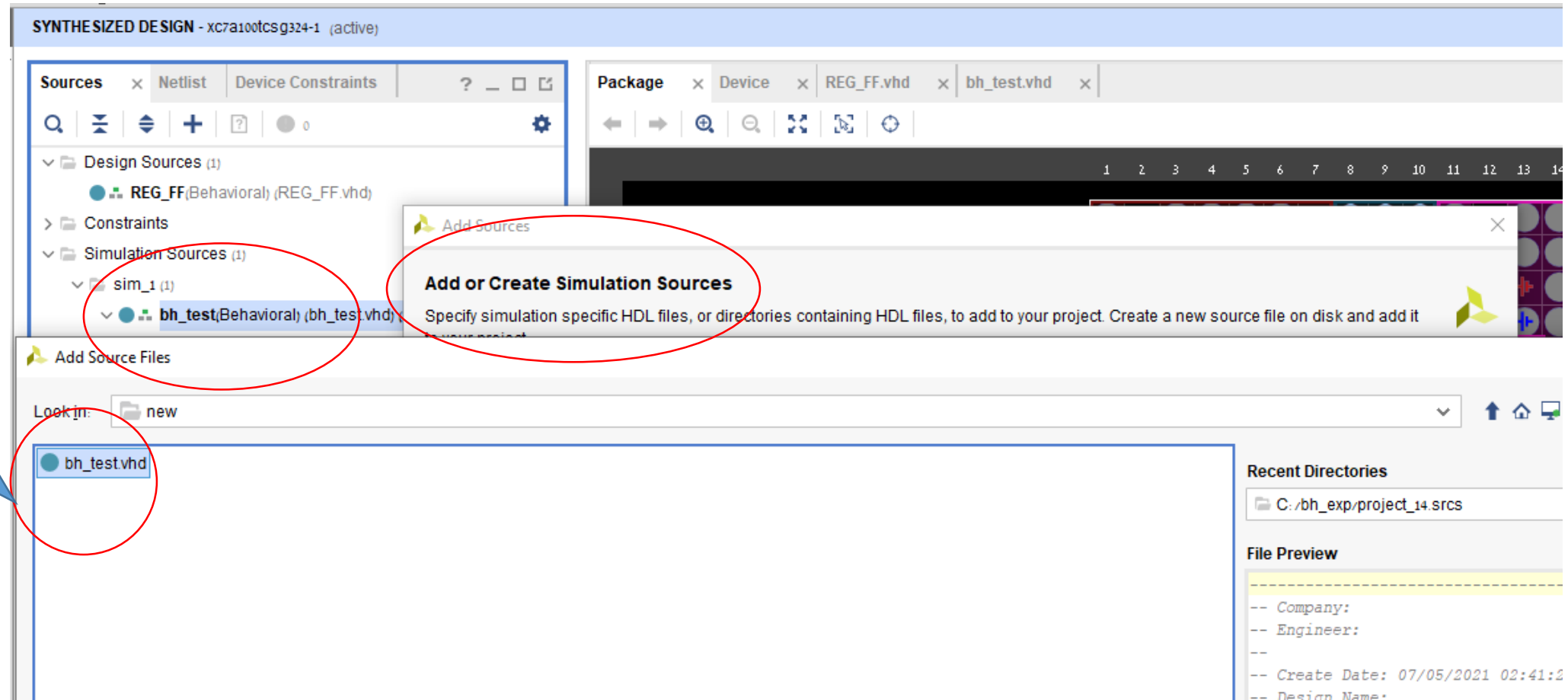
# RUN SYNTHESIS



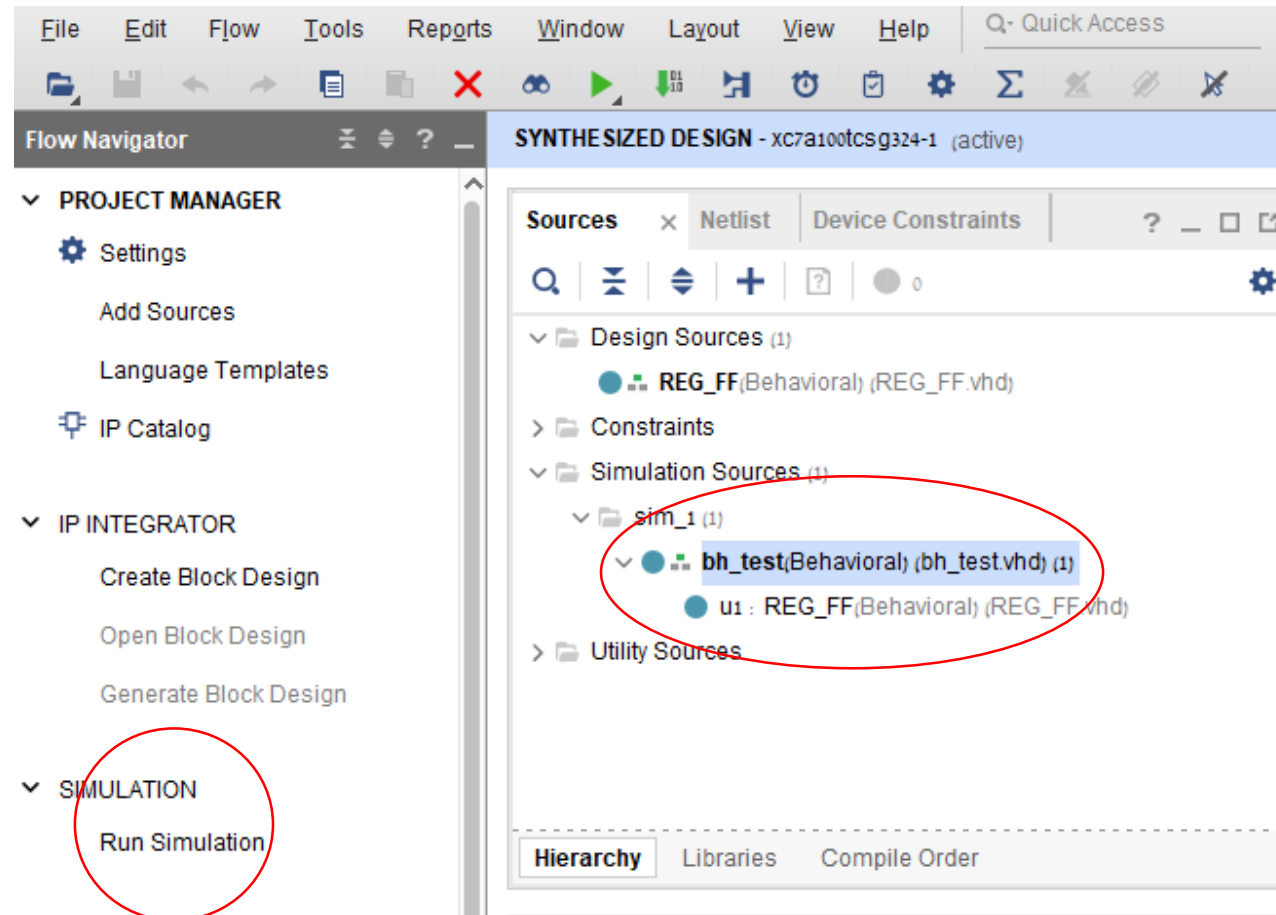
# Schematic



# add source simulation



# Simulation





# Simulation

