VLSI DESIGN

(Course code:**EECE3051)**

CASE STUDY

Title: Simulation and Implementation of Binary to Gray and Gray to Binary using FPGA Kit in Xilinx vivado software

GROUP MEMBERS

S. Sameera Tasneem(BU21EECE0100100)

S. Sai Lohitha(BU21EECE0100103)

Under the guidance of

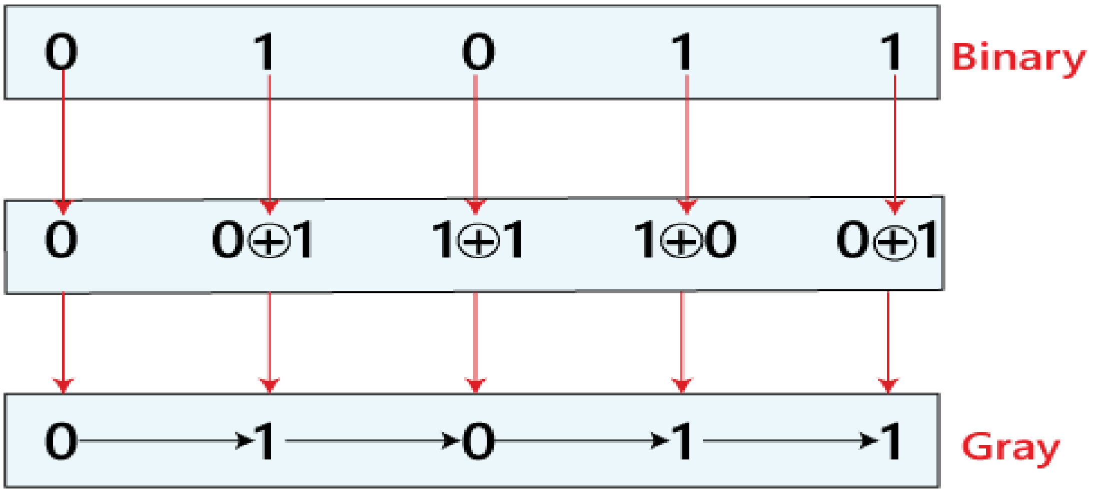
DR. ARUN KUMAR MANOHARAN

Binary to Gray code conversion

- In the Gray code, the MSB will always be the same as the 1'st bit of the given binary number.
- In order to perform the 2nd bit of the gray code, we perform the exclusive-or (XOR) of the 1'st and 2nd bit of the binary number. It means that if both the bits are different, the result will be one else the result will be 0.
- In order to get the 3rd bit of the gray code, we need to perform the exclusive-or (XOR) of the 2nd and 3rd bit of the binary number. The process remains the same for the 4th bit of the Gray code. Let's take an example to understand these steps.

Example

Suppose we have a binary number 01101, which we want to convert into Gray code.

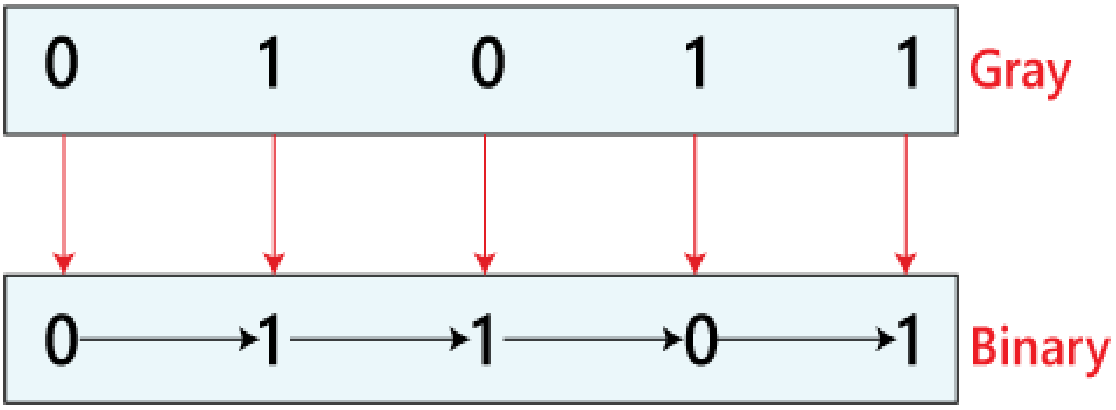


- 1st bit of the Gray code is the same as the MSB of the binary number.
- Next, we perform the XOR operation of the 1'st and the second binary number. The 1st bit is 0, and the 2nd bit is 1. Both the bits are different, so the 2nd bit of the Gray code is 1.
- Now, we perform the XOR of the 2nd bit and 3rd bit of the binary number. The 2nd bit is 1, and the 3rd bit is also 1. These bits are the same, so the 3rd bit of the Gray code is 0.
- Again perform the XOR operation of the 3rd and 4th bit of binary number. The 3rd bit is 1, and the 4th bit is 0. As these are different, the 4th bit of the Gray code is 1.
- Lastly, perform the XOR of the 4th bit and 5th bit of the binary number. The 4th bit is 0, and the 5th bit is 1. Both the bits are different, so that the 5th bit of the Gray code is 1.
- The gray code of the binary number 01101 is 01011.

Gray to Binary Code Conversion

Just like Binary to Gray code conversion; it is also a very simple process. There are the following steps used to convert the Gray code into binary.

- Just like binary to gray, in gray to binary, the 1st bit of the binary number is similar to the MSB of the Gray code.
- The 2nd bit of the binary number is the same as the 1st bit of the binary number when the 2nd bit of the Gray code is 0; otherwise, the 2nd bit is altered bit of the 1st bit of binary number. It means if the 1st bit of the binary is 1, then the 2nd bit is 0, and if it is 0, then the 2nd bit be 1.
- The 2nd step continues for all the bits of the binary number.



Steps to be followed for implementation of Binary to Gray code conversion using FPGA kit

- Open xilin vivado software>create New project>create File>give file name(module name)
- Select

Product category:General Purpose

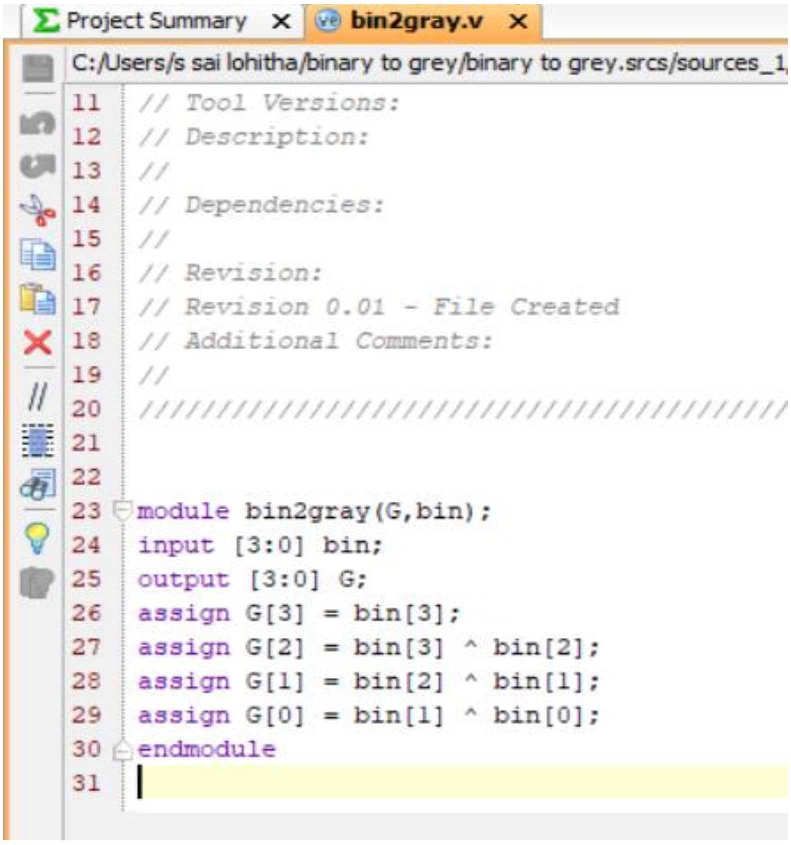
Family:Artix-7

Package:cg324

Speed value:-1

Select Xc7a100tcsg324-1

- Enter the code

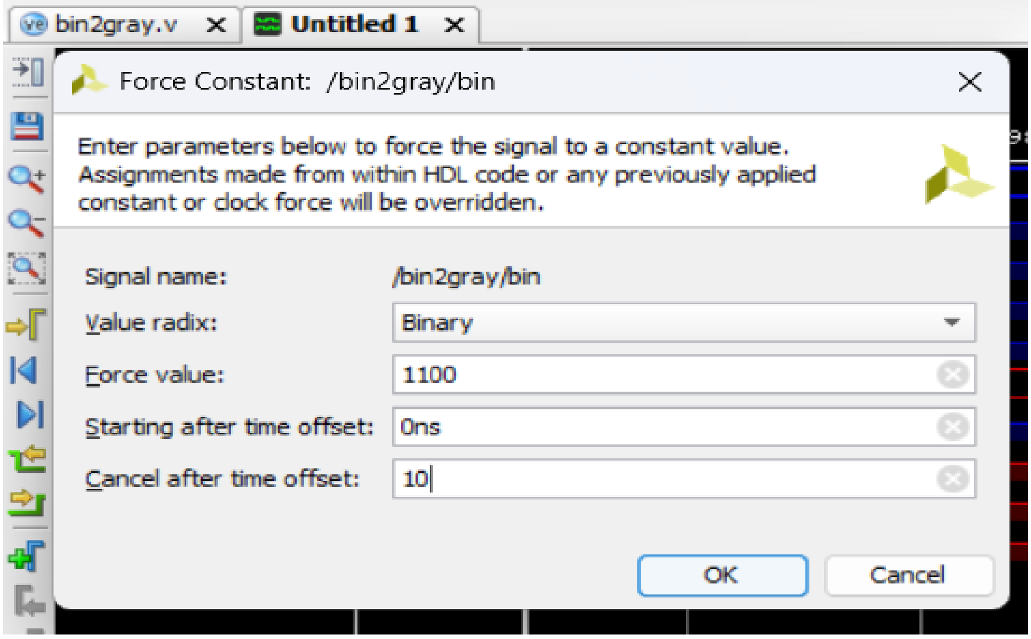


C:/Users/s sai lohitha/binary to grey/binary to grey.srcs/sources_1

```
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////


module bin2gray(G,bin);
input [3:0] bin;
output [3:0] G;
assign G[3] = bin[3];
assign G[2] = bin[3] ^ bin[2];
assign G[1] = bin[2] ^ bin[1];
assign G[0] = bin[1] ^ bin[0];
endmodule
```

- Go to Run simulation>Run behavioral simulation>Give force constant to inputs i.e, binary values



Force Constant: /bin2gray/bin

Enter parameters below to force the signal to a constant value. Assignments made from within HDL code or any previously applied constant or clock force will be overridden.

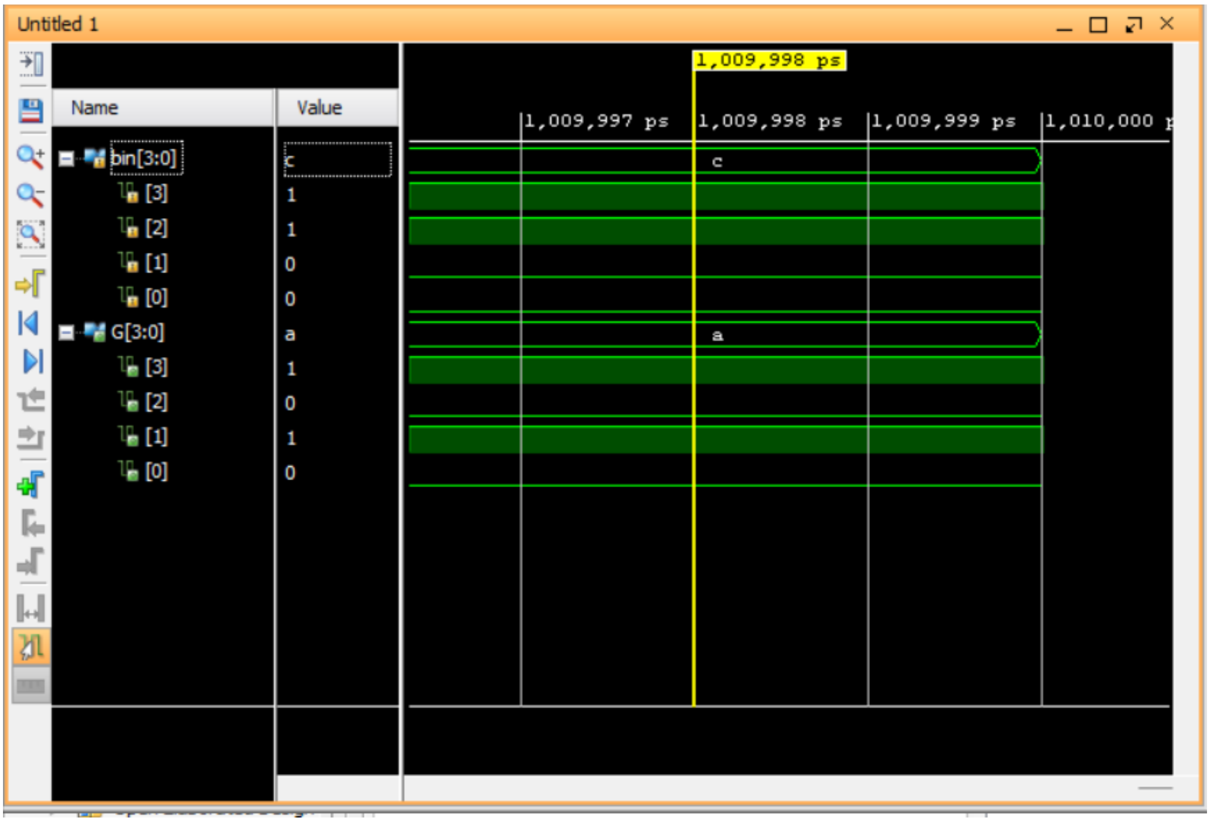
Signal name: /bin2gray/bin

Value radix: Binary

Force value: 1100

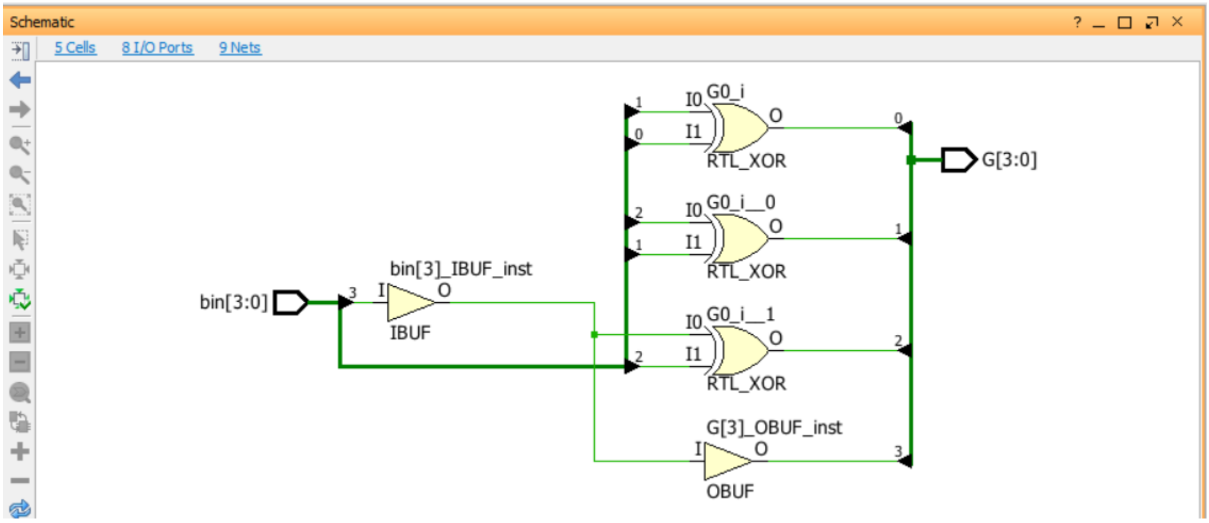
Starting after time offset: 0ns

Cancel after time offset: 10



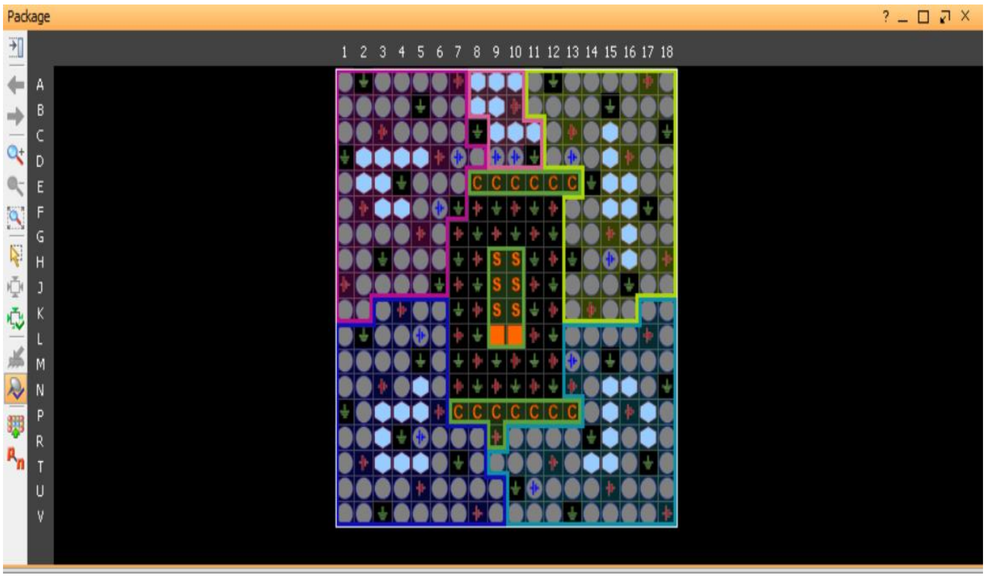
Name	Value
bin[3:0]	c
[3]	1
[2]	1
[1]	0
[0]	0
G[3:0]	a
[3]	1
[2]	0
[1]	1
[0]	0

- In RTL analysis>open elaborated design



- In schematic window, on the top change the “default Layout” to “I/O planning”
- Go to I/O ports[bottom panel]>select scalar ports
- Give I/O std as LMCMOS33
- Go to package pins>Give input as Bin[3]:P4, Bin[2]:P3 , Bin[1]:R3, Bin[4]:T1

and output pins as G[3]:R1,G[2]:V1,G[1]:U3,G[0]:V4



- Select save constraint[below edit button]
- From flow navigator>select Run suynthesis
- After successful synthesis>A pop appears>select Run implementation>click ok
- After finishing, another pop up appears, before selecting any option; connect the FPGA kit to CPU
- After Bit stream generation>choose open implementated design
- In flow navigator>program and debug>open hardware manager

At top[green bar]>select open target>Auto connect

At same bar>Progam device>select the one which is shown>program

Steps to be followed for implementation of Gray to Binary code conversion using FPGA kit

- Open xilin vivado software>create New project>create File>give file name(module name)
- Select

Product category:General Purpose

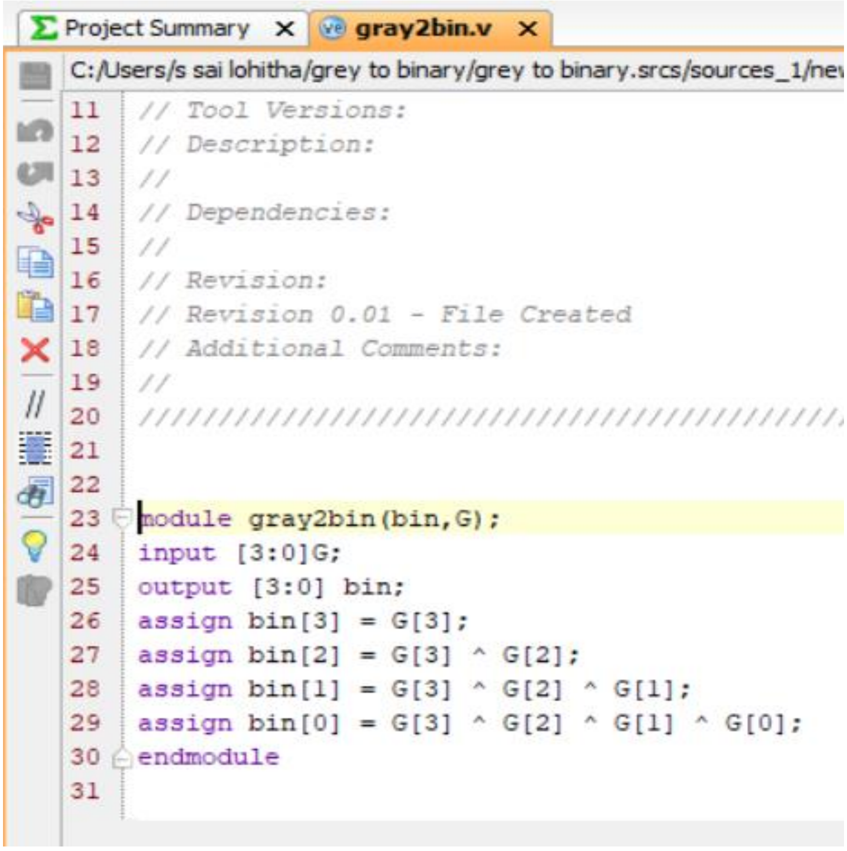
Family:Artix-7

Package:cg324

Speed value:-1

Select Xc7a100tcsg324-1

- Enter the code



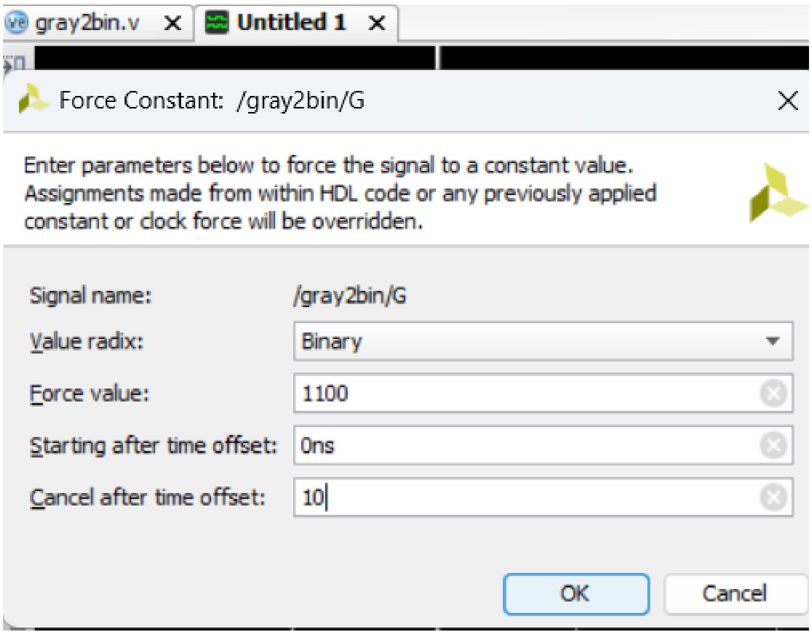
C:/Users/s sai lohitha/grey to binary/grey to binary.srcs/sources_1/new

```
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
///////////////////////////////////////////


module gray2bin(bin,G);
input [3:0]G;
output [3:0] bin;
assign bin[3] = G[3];
assign bin[2] = G[3] ^ G[2];
assign bin[1] = G[3] ^ G[2] ^ G[1];
assign bin[0] = G[3] ^ G[2] ^ G[1] ^ G[0];
endmodule

```

- Go to Run simulation>Run behavioral simulation>Give force constant to inputs i.e, binary values



Force Constant: /gray2bin/G

Enter parameters below to force the signal to a constant value. Assignments made from within HDL code or any previously applied constant or clock force will be overridden.

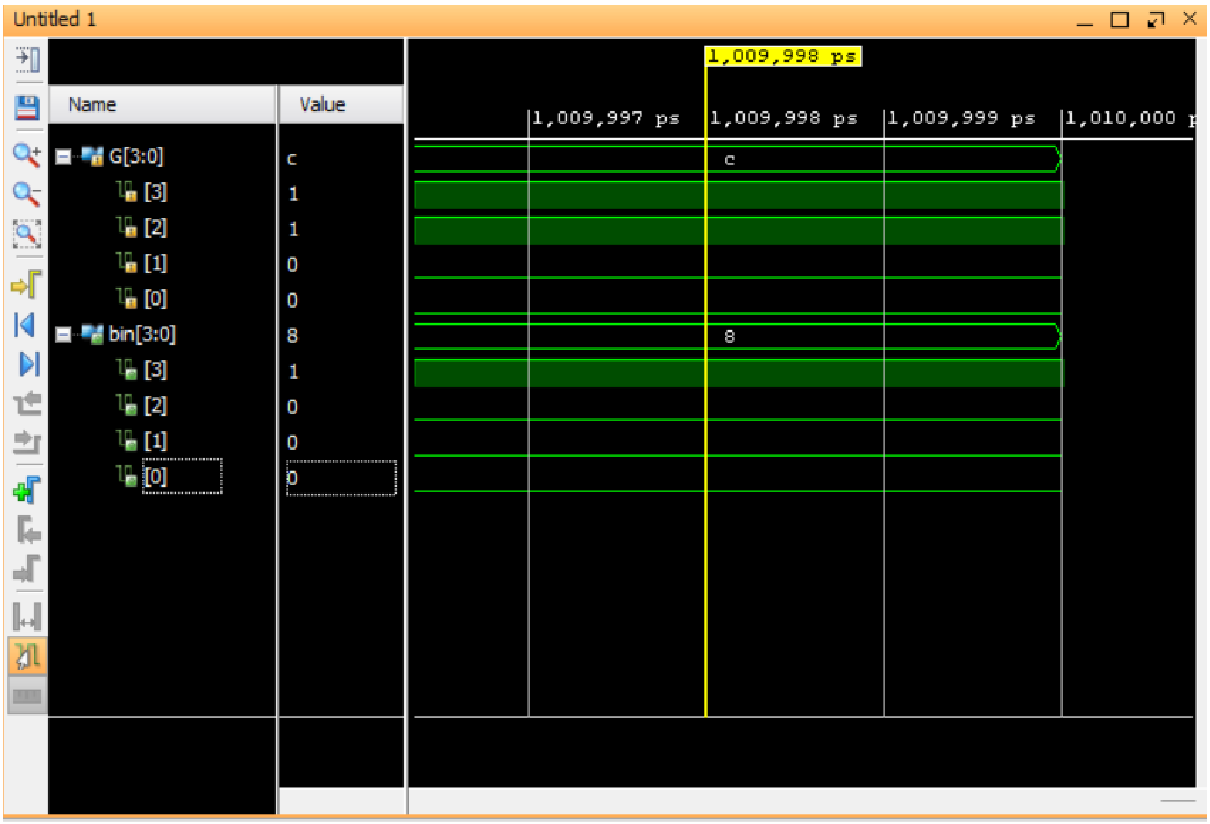
Signal name: /gray2bin/G

Value radix: Binary

Force value: 1100

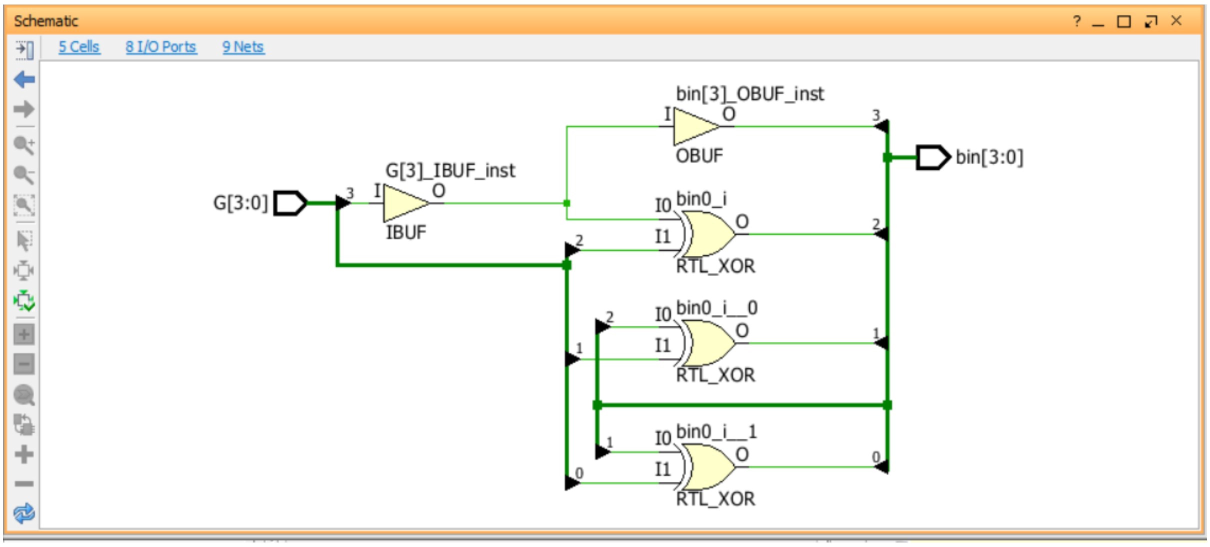
Starting after time offset: 0ns

Cancel after time offset: 10

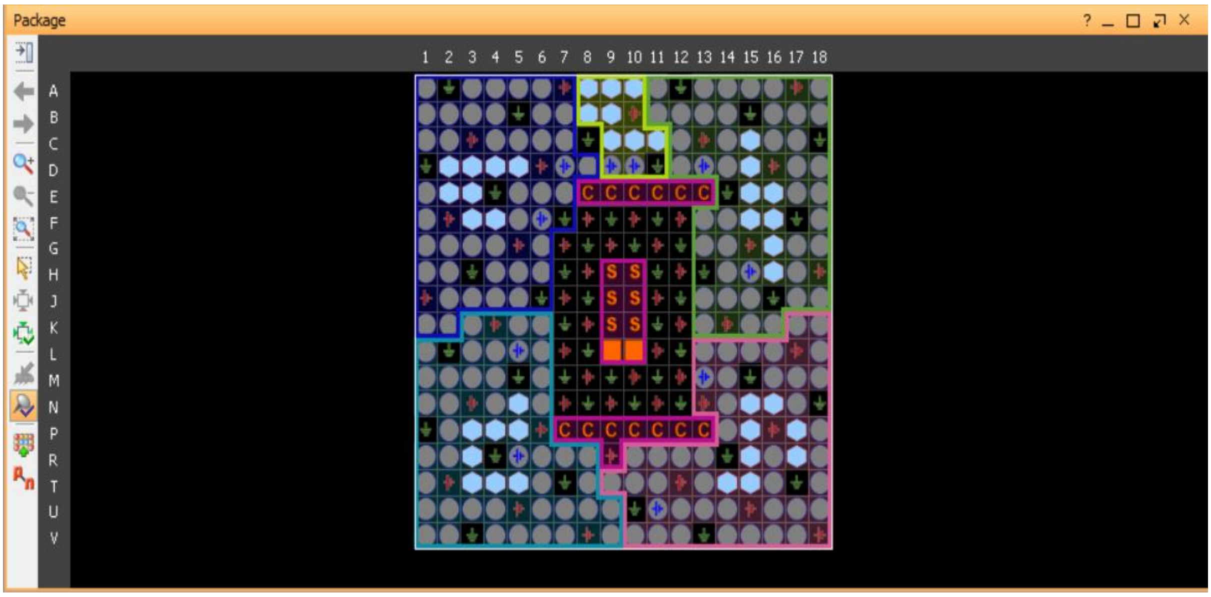


Name	Value
G[3:0]	c
[3]	1
[2]	1
[1]	0
[0]	0
bin[3:0]	8
[3]	1
[2]	0
[1]	0
[0]	0

- In RTL analysis>open elaborated design



- In schematic window, on the top change the “default Layout” to “I/O planning”
- Go to I/O ports[bottom panel]>select scalar ports
- Give I/O std as LMCMOS33
  Go to package pins>Give input as Bin[3]:P4, Bin[2]:P3 , Bin[1]:R3, Bin[4]:T1 and output pins as G[3]:R1,G[2]:V1,G[1]:U3,G[0]:V4



- Select save constraint[below edit button]
- From flow navigator>select Run suynthesis
- After successful synthesis>A pop appears>select Run implementation>click ok

- After finishing, another pop up appears, before selecting any option; connect the FPGA kit to CPU
- After Bit stream generation>choose open implementated design
- In flow navigator>program and debug>open hardware manager

  At top[green bar]>select open target>Auto connect

  At same bar>Program device>select the one which is shown>program

VIDEO LINK:

https://drive.google.com/file/d/1XWrQFxUrEE6ZnwaPCdBDi47GcxUIiLE8/view?usp=drive_link