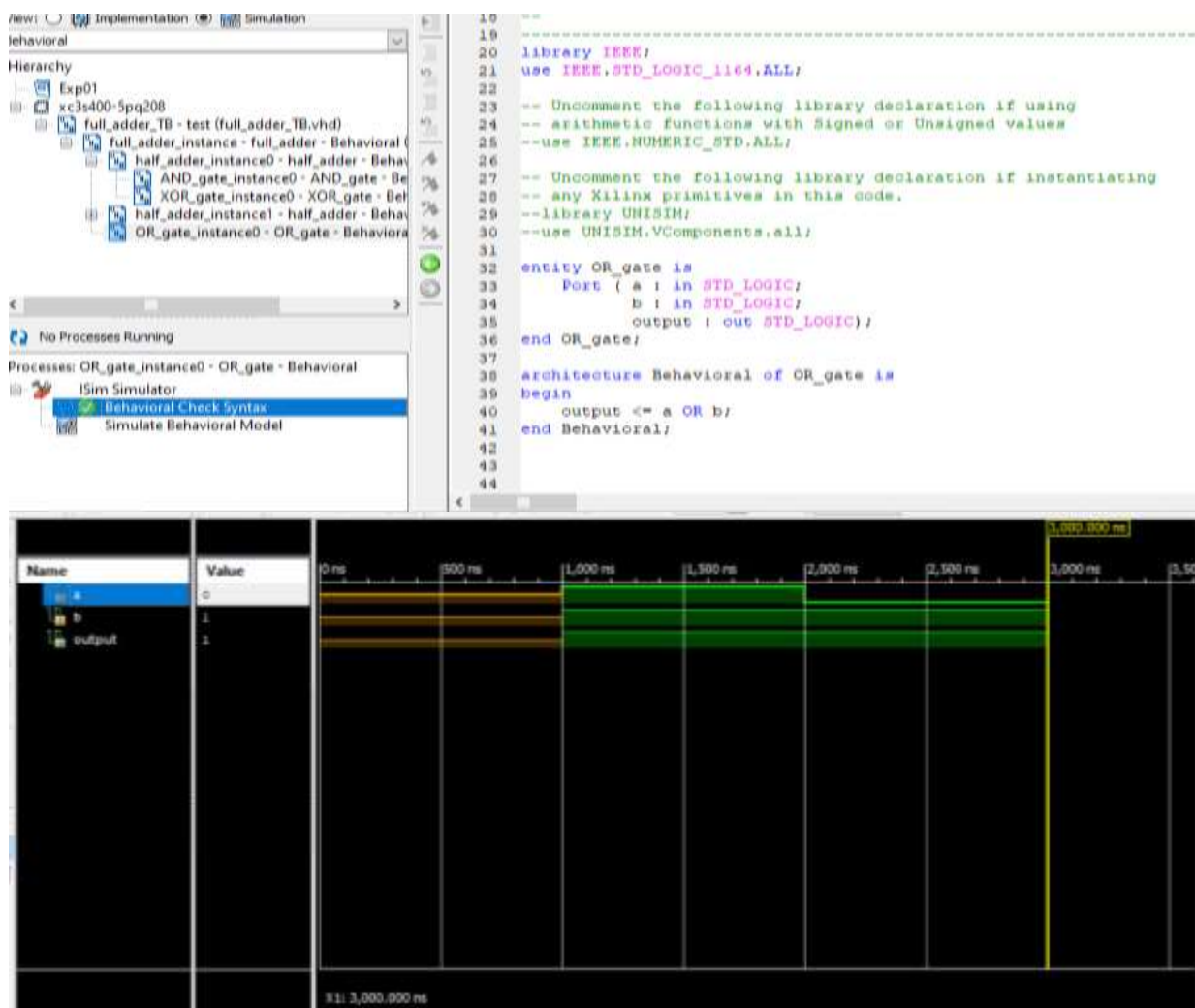


محمد مهدی نظری ۹۹۳۱۰۶۱ - آرمین ابراهیمی صبا ۹۹۳۱۰۸۶

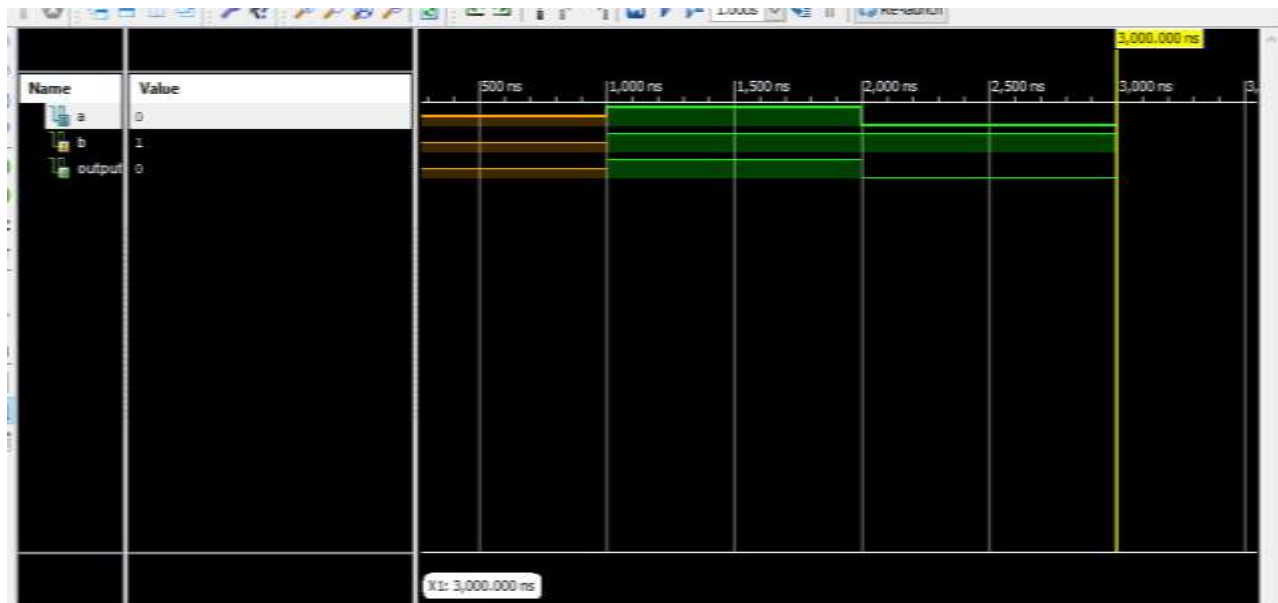
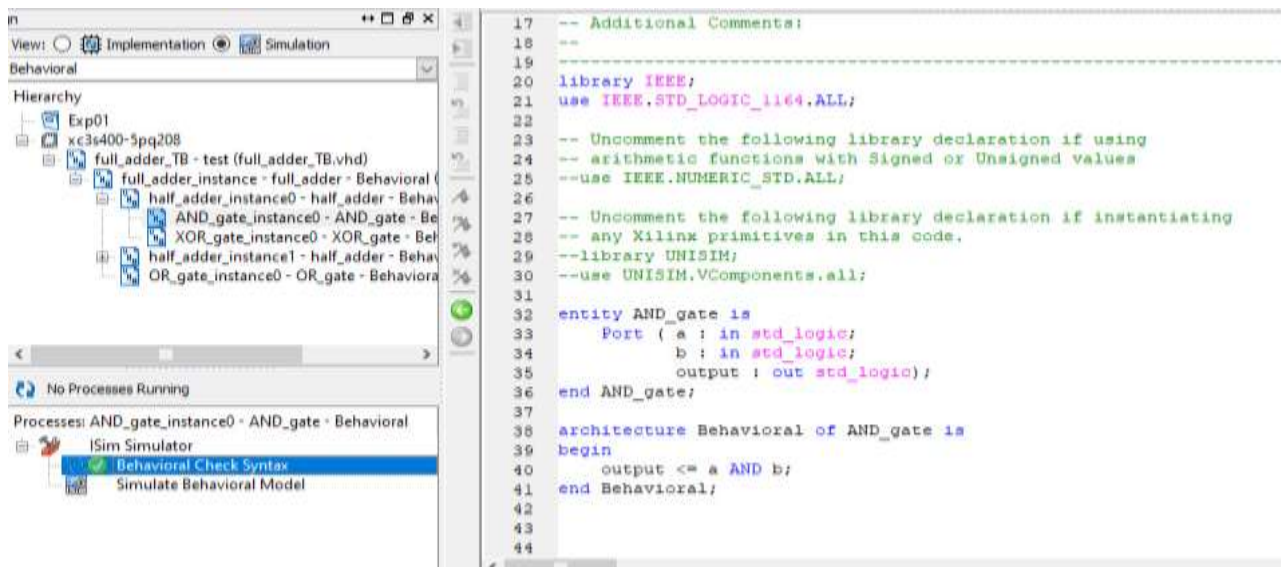
این آزمایش نحوه تولید جمع کننده کامل را با استفاده از نیم جمع کننده و گیت های and و or و xor را میدهد. در هر قسمت کد مربوطه آورده شده که از نظر قواعد نگارشی زبان vhdل یا syntax آن چک شده ( تیک سبز سمت چپ در بخش Behavioral Check Syntax)

در کنار آن نیز تست کوچکی از هر ماژول آورده شده ( بصورت Force Constant ) و در نهایت یک تست بنچ کلی برای جمع کننده نهایی آورده شده. توضیحات نحوه کار و نتیجه مطلوب ماژول ها به ازای ورودی های مختلف در پیش گزارش تشریح شده است.

## ۱ - گیت OR



## ۲ - گیت AND



```

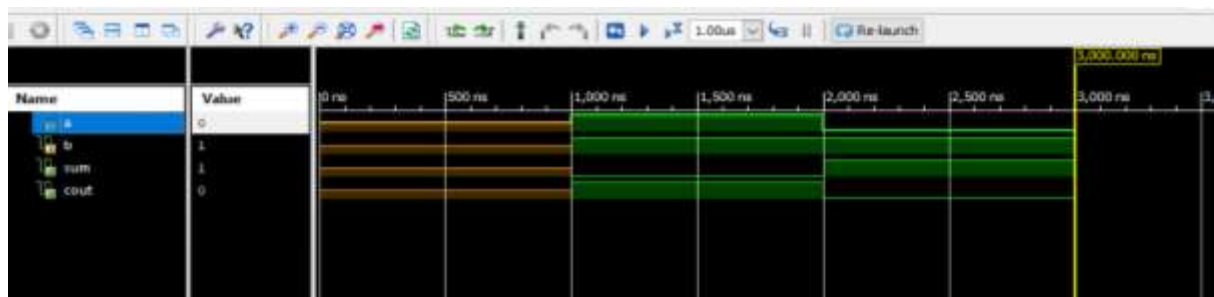
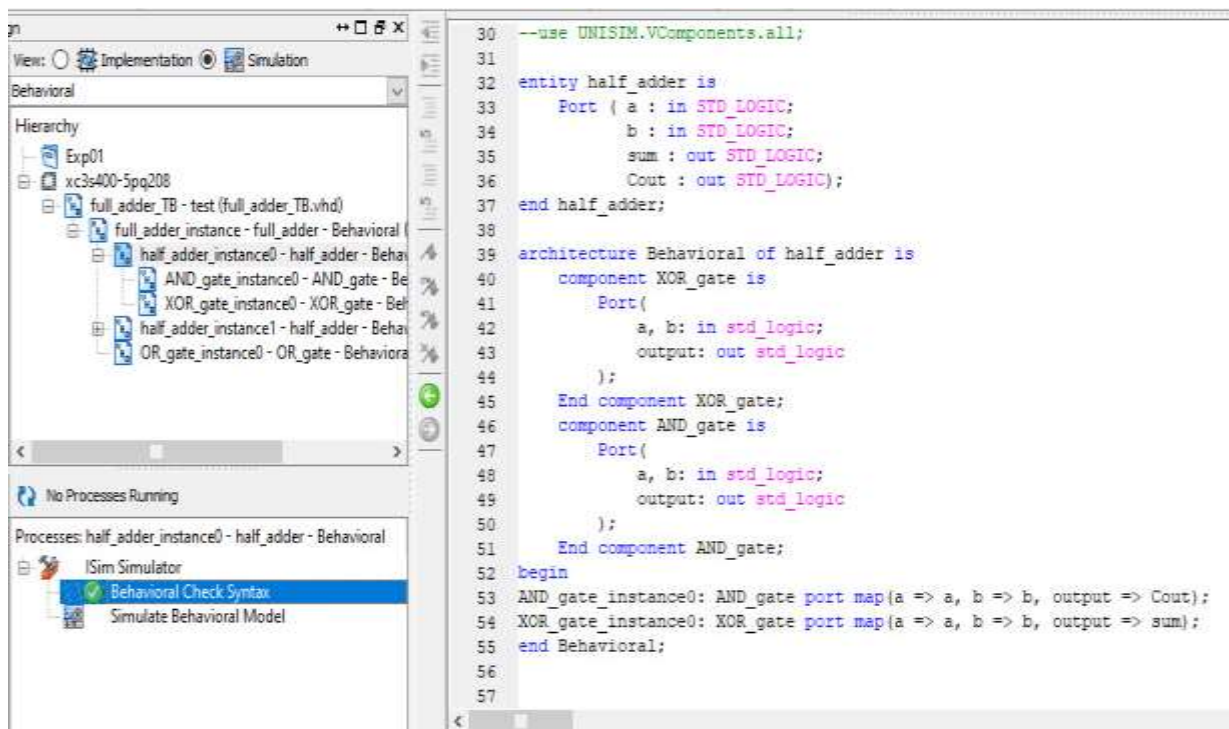
19
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22
23 -- Uncomment the following library declaration if using
24 -- arithmetic functions with Signed or Unsigned values
25 --use IEEE.NUMERIC_STD.ALL;
26
27 -- Uncomment the following library declaration if instantiating
28 -- any Xilinx primitives in this code.
29 --library UNISIM;
30 --use UNISIM.VComponents.all;
31
32 entity XOR_gate is
33     Port ( a : in STD_LOGIC;
34           b : in STD_LOGIC;
35           output : out STD_LOGIC);
36 end XOR_gate;
37
38 architecture Behavioral of XOR_gate is
39
40 begin
41     output <= a XOR b;
42
43 end Behavioral;
44
45
46

```

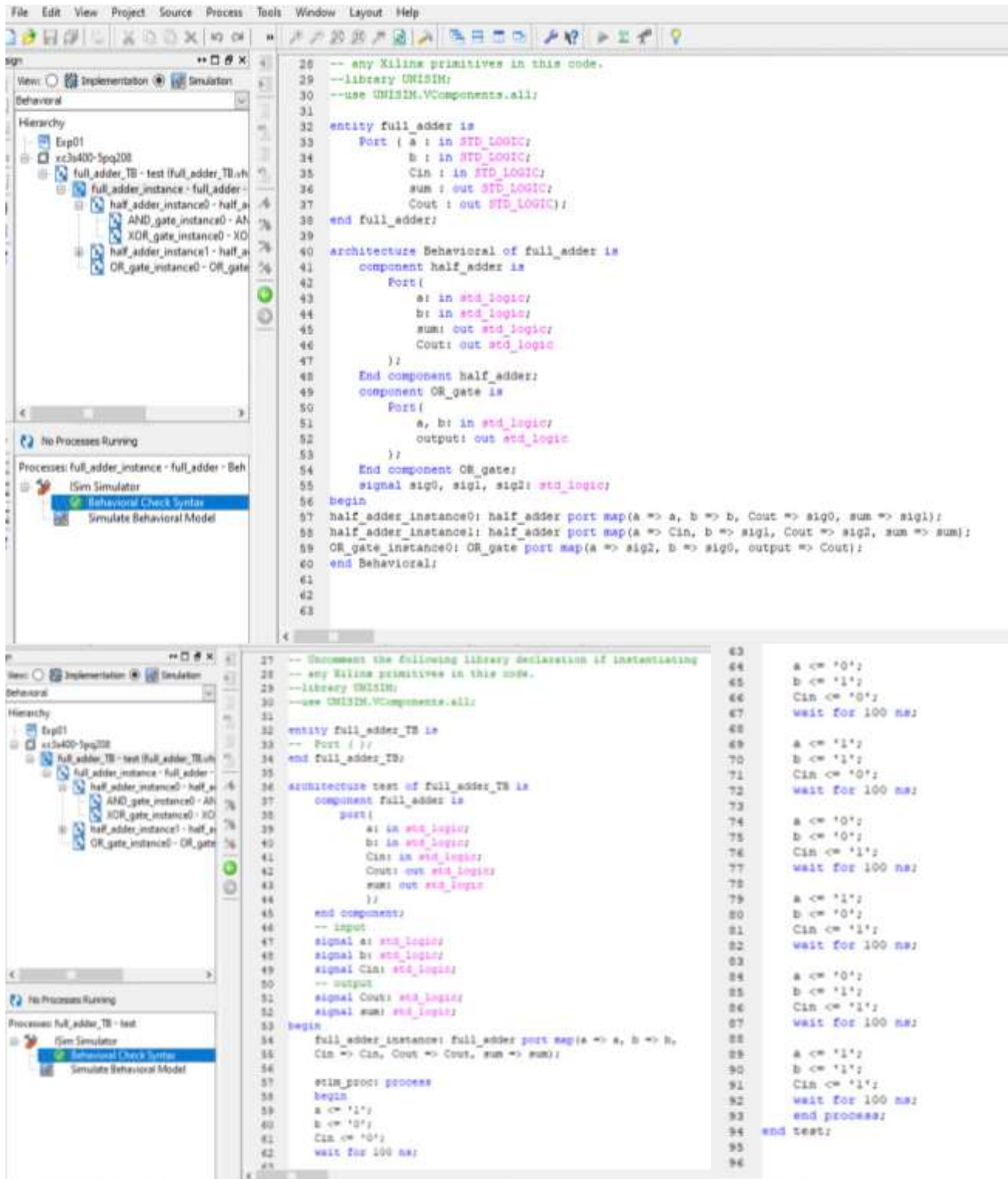


#### ۴ – ماژول half adder

این ماژول از یک گیت xor و یک گیت and تشکیل شده که هرکدام از ورودی ها به هر دو گیت میروند و خروجی گیت and را Cout ، Cout را تشکیل میدهد و خروجی گیت xor ، Sum را. در قسمت کد هم از دو گیت نمونه ساخته شده و در ماژول half adder مورد استفاده قرار گرفته شده .



## ۵- مازول full adder و testbench آن





این ماژول از دو half adder و یک گیت or ساخته شده که ورودی ها به نیم جمع کننده اول میروند و ورودی Cin و خروجی Sum نیم جمع کننده اول به عنوان ورودی به نیم جمع کننده دوم میروند که خروجی Sum آن خروجی Sum جمع کننده کامل را میسازد و در طرف دیگر خروجی های Cout دو نیم جمع کننده با هم به گیت or میروند که خروجی آن خروجی Cout جمع کننده کامل را میسازد. در قسمت کد هم از ماژول half adder و or به ترتیب ۲ و ۱ نمونه ساخته شده و طبق ترتیب گفته شده map شده اند.

در بخش testbench هم چند ورودی به عنوان مثال داده شده و جواب آن را تست میکنیم. مثلاً در ۱۰۰ نانوثانیه اول انتظار داریم به ازای ورودی های ۱ و ۰ و ۰ و ۰ خروجی جمع ۱ و ۰ و ۰ خروجی ۰ بگیریم که طبق شکل پایین صحیح است. یا مثلاً در صد نانوثانیه سوم به ازای ورودی های ۱ و ۱ و ۰ و ۰ انتظار داریم که خروجی جمع ۰ و ۰ و ۰ و ۰ بگیریم که مانند شکل زیر صحیح است.

سایر حالت ها در تست بنچ و پیش گزارش آورده شده اند.

