

آزمایشگاه مدار منطقی

نویسنده:

علیرضا طباطبائی (9723052)



Amirkabir University of Technology
(Tehran Polytechnic)

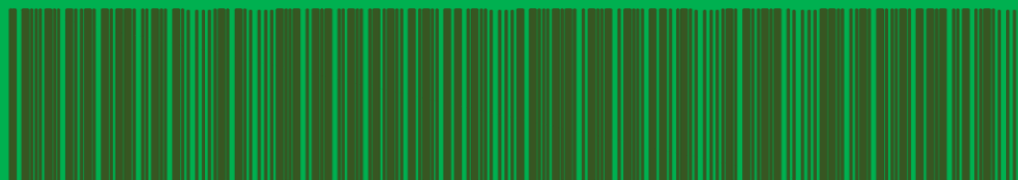
استاد آزمایشگاه:

دکتر علیرضا

ظاهری نوید

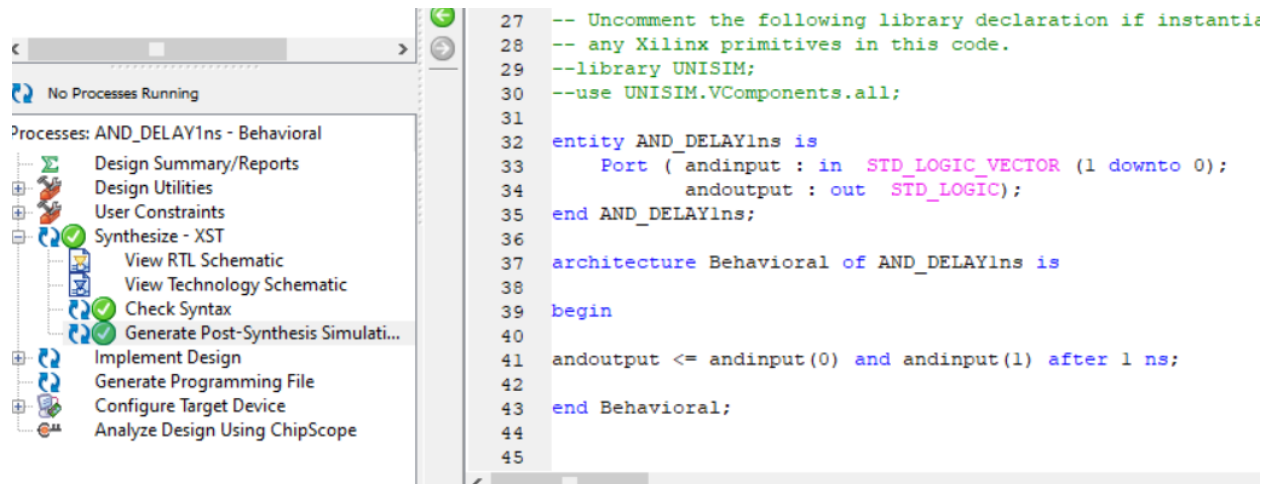


THE PRESENT IS THEIRS, THE FUTURE, FOR



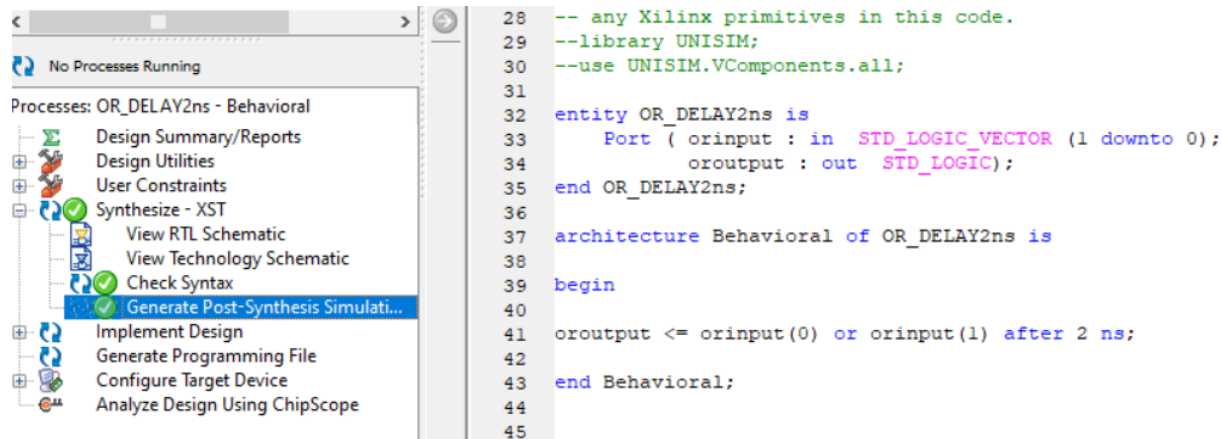
WHICH I REALLY WORK, IS MINE.

تمرین اول: ساخت گیت های AND OR NOT XOR به همراه تاخیر:



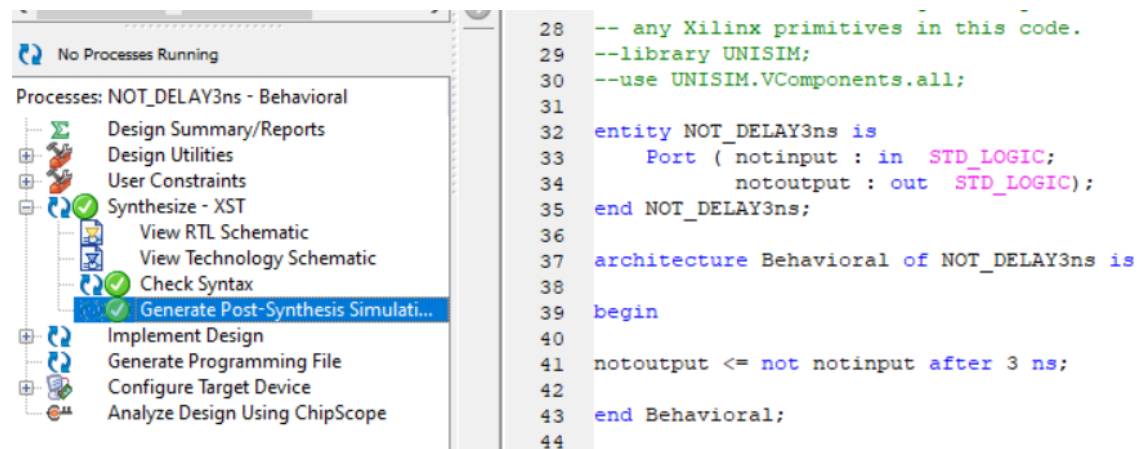
The screenshot shows the Xilinx IDE interface. On the left, the 'Processes' pane for 'AND_DELAY1ns - Behavioral' is visible, listing steps from Design Summary to Analyze Design. The 'Generate Post-Synthesis Simulation' step is highlighted. On the right, the VHDL code for the AND gate is shown:

```
27 -- Uncomment the following library declaration if instantia
28 -- any Xilinx primitives in this code.
29 --library UNISIM;
30 --use UNISIM.VComponents.all;
31
32 entity AND_DELAY1ns is
33     Port ( andinput : in  STD_LOGIC_VECTOR (1 downto 0);
34           andoutput : out STD_LOGIC);
35 end AND_DELAY1ns;
36
37 architecture Behavioral of AND_DELAY1ns is
38
39 begin
40
41 andoutput <= andinput(0) and andinput(1) after 1 ns;
42
43 end Behavioral;
44
45
```



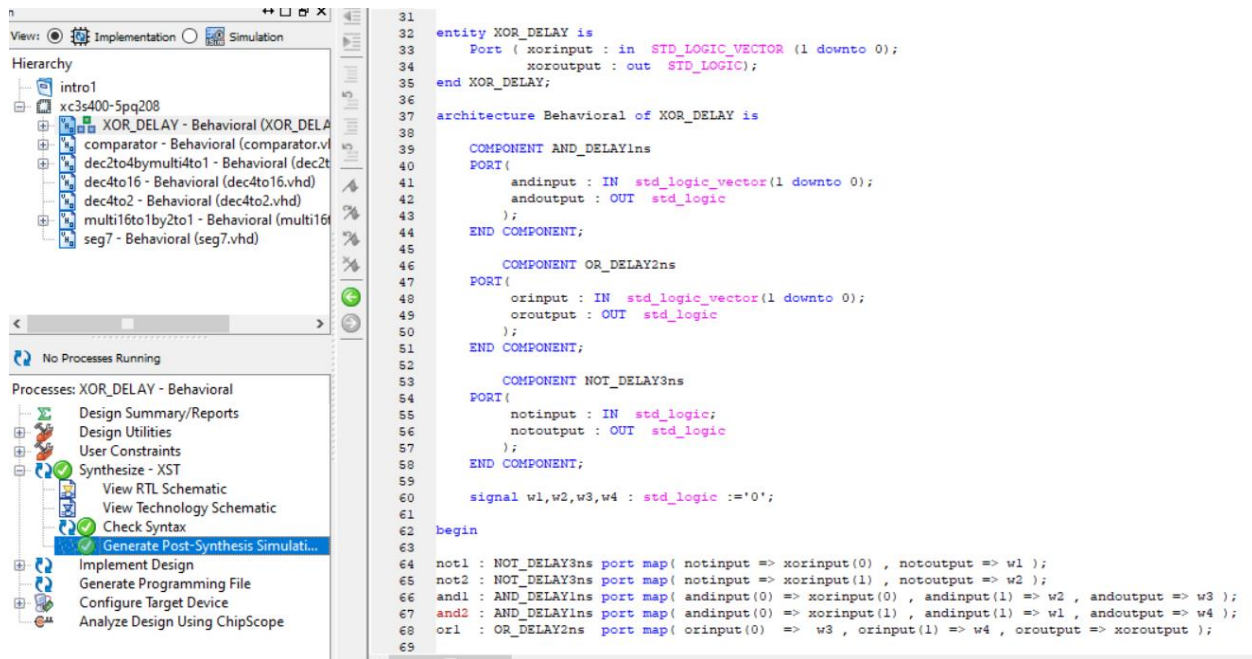
The screenshot shows the Xilinx IDE interface. On the left, the 'Processes' pane for 'OR_DELAY2ns - Behavioral' is visible, with 'Generate Post-Synthesis Simulation' highlighted. On the right, the VHDL code for the OR gate is shown:

```
28 -- any Xilinx primitives in this code.
29 --library UNISIM;
30 --use UNISIM.VComponents.all;
31
32 entity OR_DELAY2ns is
33     Port ( orinput : in  STD_LOGIC_VECTOR (1 downto 0);
34           oroutput : out STD_LOGIC);
35 end OR_DELAY2ns;
36
37 architecture Behavioral of OR_DELAY2ns is
38
39 begin
40
41 oroutput <= orinput(0) or orinput(1) after 2 ns;
42
43 end Behavioral;
44
45
```



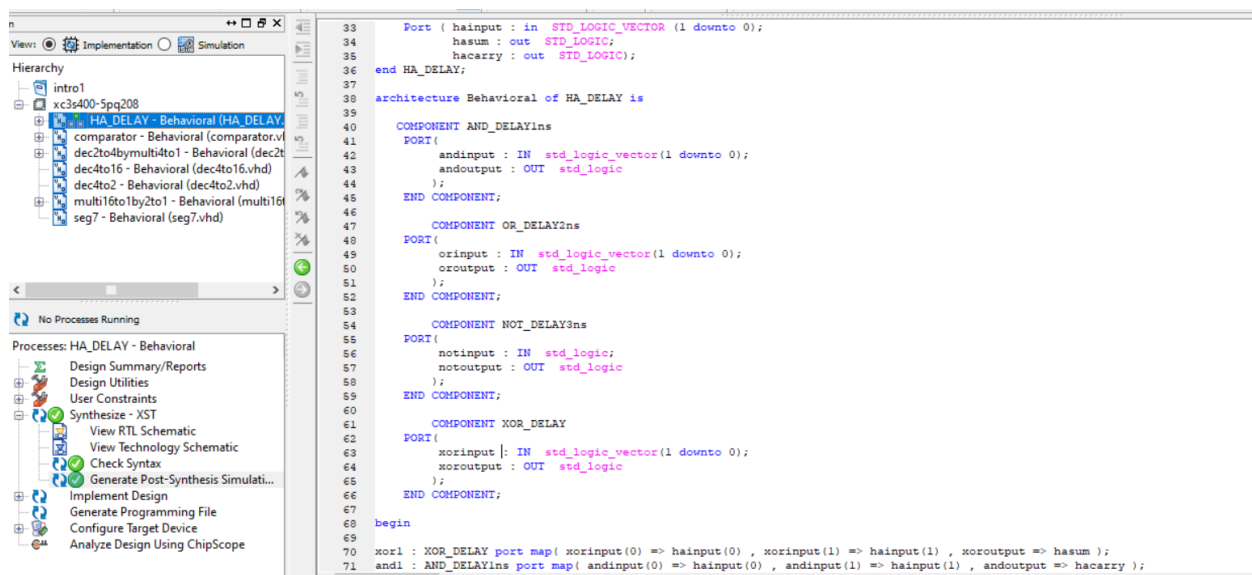
The screenshot shows the Xilinx IDE interface. On the left, the 'Processes' pane for 'NOT_DELAY3ns - Behavioral' is visible, with 'Generate Post-Synthesis Simulation' highlighted. On the right, the VHDL code for the NOT gate is shown:

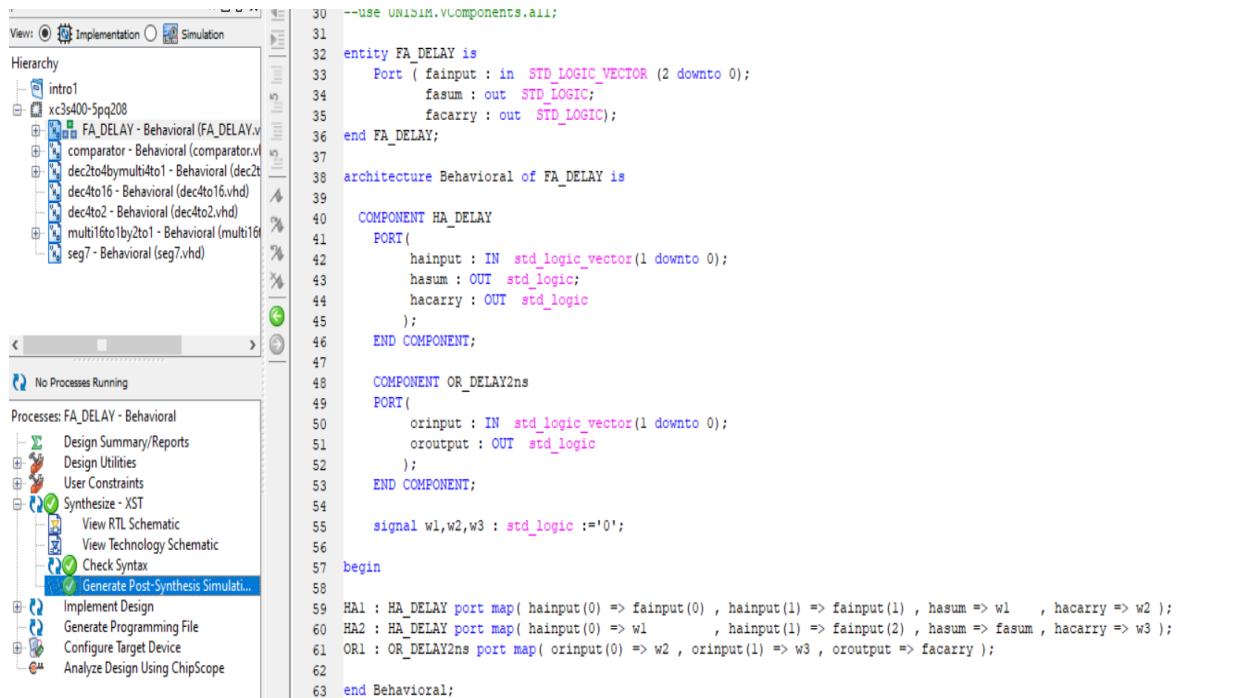
```
28 -- any Xilinx primitives in this code.
29 --library UNISIM;
30 --use UNISIM.VComponents.all;
31
32 entity NOT_DELAY3ns is
33     Port ( notinput : in  STD_LOGIC;
34           notoutput : out STD_LOGIC);
35 end NOT_DELAY3ns;
36
37 architecture Behavioral of NOT_DELAY3ns is
38
39 begin
40
41 notoutput <= not notinput after 3 ns;
42
43 end Behavioral;
44
```



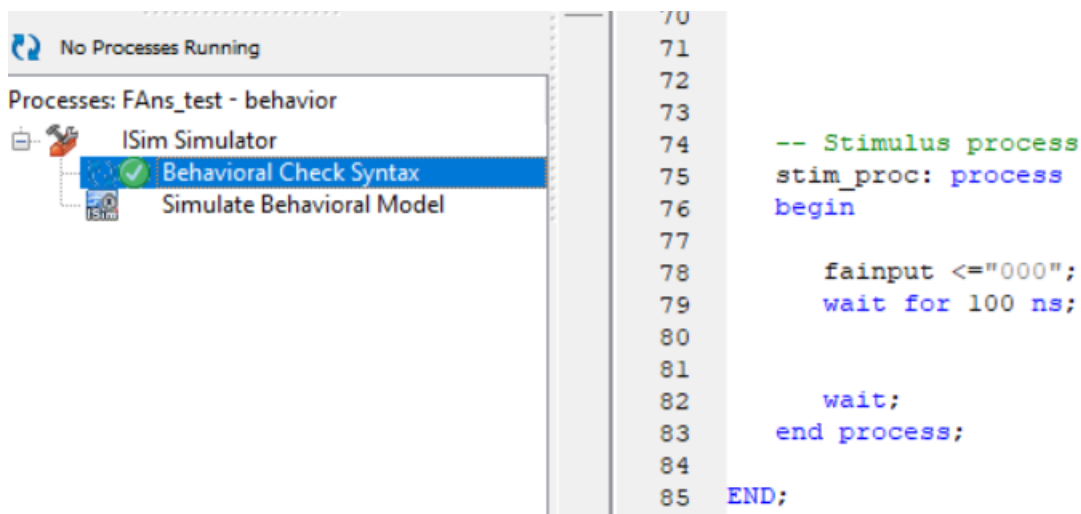
ساخت نیم جمع کننده و تمام جمع کننده با تاخیر:

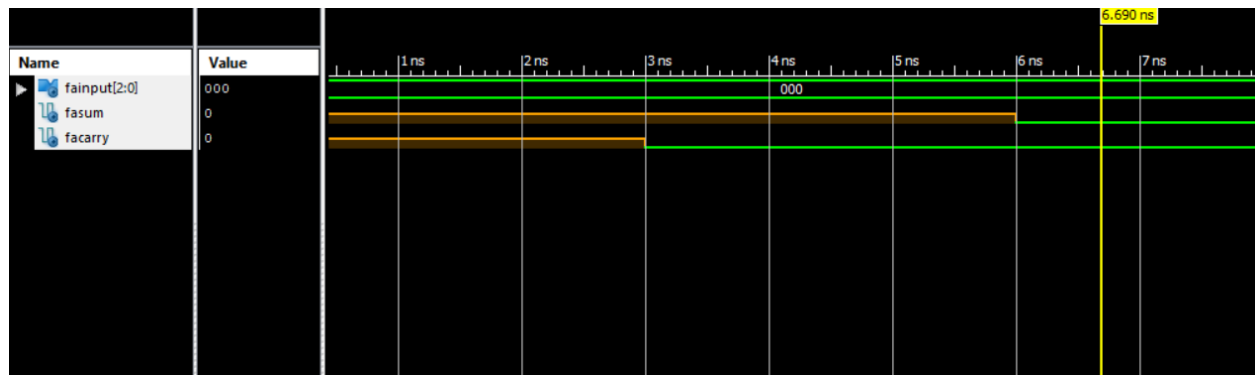
جداول کارنو و ... آن در تکلیف شماره 1 موجود میباشد.





تست پنج های FA برای ورودی 000:





تست پنج های FA برای ورودی 111:

No Processes Running

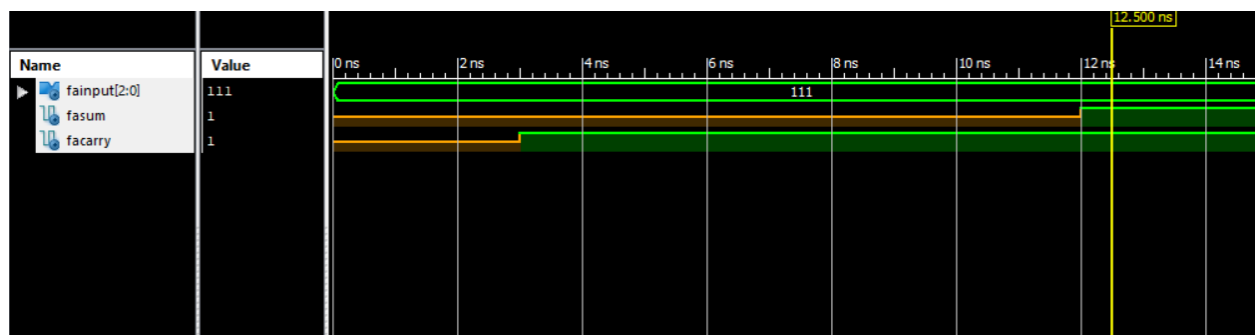
Processes: FAns_test - behavior

- ISim Simulator
- Behavioral Check Syntax
- Simulate Behavioral Model

```

70
71
72
73
74  -- Stimulus process
75  stim_proc: process
76  begin
77
78      fainput <="111";
79      wait for 100 ns;
80
81
82      wait;
83  end process;
84
85  END;
86

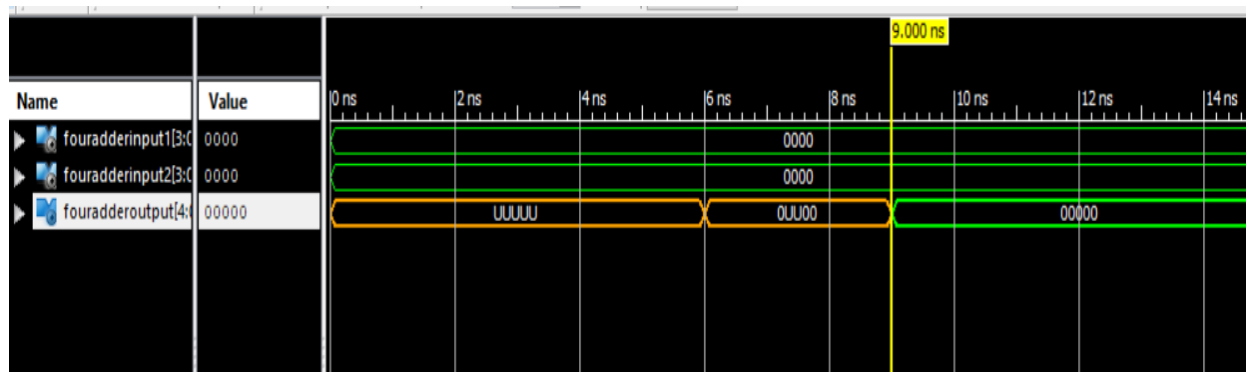
```



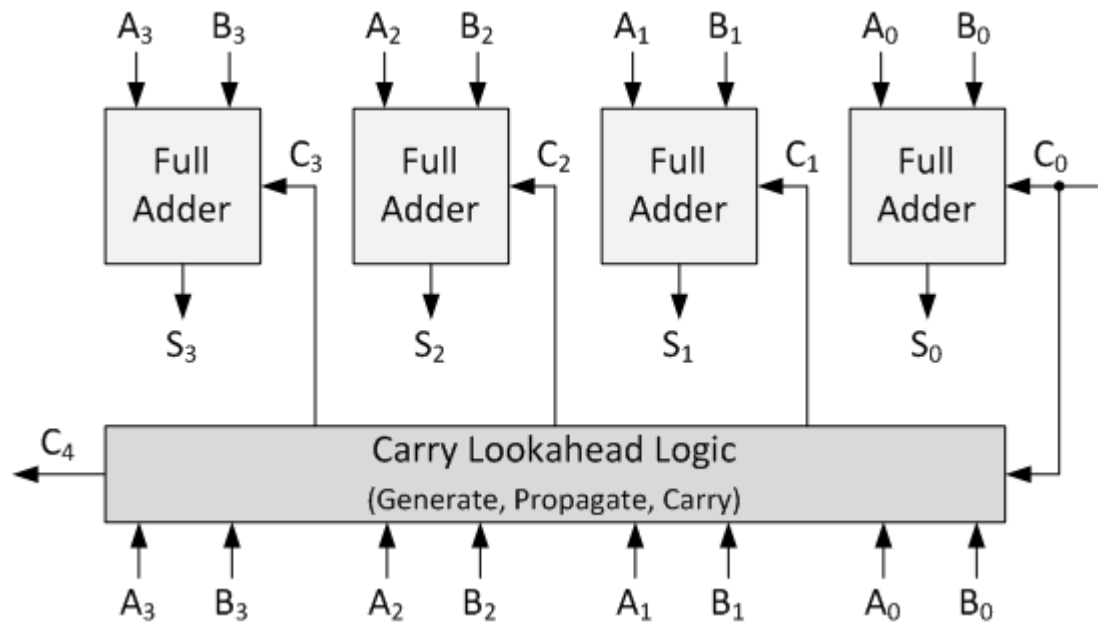
جمع کننده 4 بیتی:

بیشترین تاخیر به ازای جمع 1111 با 0001 بود برابر 18 نانوثانیه و کمترین مربوط به جمع 0000 با 0000 بود برابر 9 نانوثانیه.

روش بدست آوردن: برای بیشترین، باید بیشترین تعداد بیت ها تغییر کند و برای کمترین باید کمترین تعداد بیت ها تغییر کند.



جمع کننده lookahead :



```

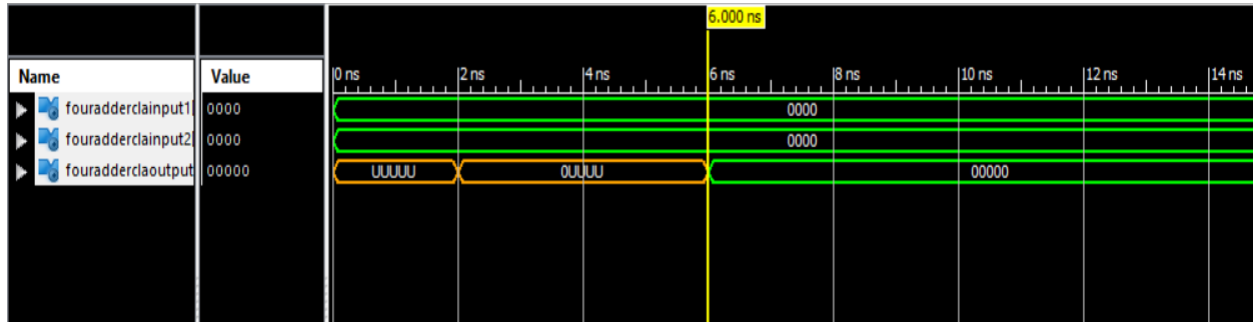
gn
View: Implementation Simulation
Hierarchy
  intro1
  xc3s400-5pg208
  FOURADDERCLA - Behavioral (FOURADDERCLA.vhd)
  FOUR_BIT_ADDER_DELAY - Behavioral (FOUR_BIT_ADDER_DELAY.vhd)
  comparator - Behavioral (comparator.vhd)
  dec2to4bymulti4to1 - Behavioral (dec2to4bymulti4to1.vhd)
  dec4to16 - Behavioral (dec4to16.vhd)
  dec4to2 - Behavioral (dec4to2.vhd)
  multi7to1by2to1 - Behavioral (multi7to1by2to1.vhd)
  seg7 - Behavioral (seg7.vhd)
No Processes Running
Processes: FOURADDERCLA - Behavioral
  Design Summary/Reports
  Design Utilities
  User Constraints
  Synthesize - XST
  View RTL Schematic
  View Technology Schematic
  Check Syntax
  Generate Post-Synthesis Simulation
  Implement Design
  Generate Programming File
  Configure Target Device
  Analyze Design Using ChipScope

architecture Behavioral of FOURADDERCLA is
38
39
40   COMPONENT FA_DELAY
41   PORT
42     fainput : IN std_logic_vector(2 downto 0);
43     fasum : OUT std_logic;
44     facarry : OUT std_logic;
45   );
46   END COMPONENT;
47
48   COMPONENT XOR_DELAY
49   PORT
50     xorinput : IN std_logic_vector(1 downto 0);
51     xoroutput : OUT std_logic;
52   );
53   END COMPONENT;
54
55   signal p1,p3,c1,c2,c3,m1,m2,m3,m4,m5,m6 : std_logic := '0';
56
57 begin
58
59   xor1 : XOR_DELAY port map( xorinput(0) => fouradderclainput1(1), xorinput(1) => fouradderclainput2(1), xoroutput => p1 );
60   xor2 : XOR_DELAY port map( xorinput(0) => fouradderclainput1(2), xorinput(1) => fouradderclainput2(2), xoroutput => p2 );
61   xor3 : XOR_DELAY port map( xorinput(0) => fouradderclainput1(3), xorinput(1) => fouradderclainput2(3), xoroutput => p3 );
62   m1 <= fouradderclainput1(1) and fouradderclainput2(1) after 1 ns;
63   m2 <= p1 and c1 after 1 ns;
64   m3 <= fouradderclainput1(2) and fouradderclainput2(2) after 1 ns;
65   m4 <= p2 and c2 after 1 ns;
66   m5 <= fouradderclainput1(3) and fouradderclainput2(3) after 1 ns;
67   m6 <= p3 and c3 after 1 ns;
68   c1 <= fouradderclainput1(0) and fouradderclainput2(0) after 1 ns;
69   c2 <= m1 or m2 after 2 ns;
70   c3 <= m3 or m4 after 2 ns;
71   fouradderclacoutput(4) <= m5 or m6 after 2 ns;
72
73   FA1 : FA_DELAY port map( fainput(0) => fouradderclainput1(0), fainput(1) => fouradderclainput2(0), fainput(2) => '0', fasum => fouradderclacoutput(0) );
74   FA2 : FA_DELAY port map( fainput(0) => fouradderclainput1(1), fainput(1) => fouradderclainput2(1), fainput(2) => c1, fasum => fouradderclacoutput(1) );
75   FA3 : FA_DELAY port map( fainput(0) => fouradderclainput1(2), fainput(1) => fouradderclainput2(2), fainput(2) => c2, fasum => fouradderclacoutput(2) );
76   FA4 : FA_DELAY port map( fainput(0) => fouradderclainput1(3), fainput(1) => fouradderclainput2(3), fainput(2) => c3, fasum => fouradderclacoutput(3) );

```

مقایسه تاخیر جمع 0000 با 0000 :

در قبلی 9 و در این یکی 6 نانو ثانیه است.



مقایسه تاخیر جمع 0001 با 1111 :

در قبلی 18 و در این یکی همچنان 18 نانو ثانیه است.

