

با سلام خدمت دانشجویان عزیز

در خصوص ویدیوی آموزش ISE که توی گروه ارسال کرده بودم، نکته‌ای از قلم افتاد که در این نوشته خدمتون ارائه داده و از شما می‌خواهم اگر تجربه‌ی ارزشمند مرتبط با این موضوع دارید به بنده هم منتقل بفرمایید.

در تغییر نوع شبیه‌سازی از مدل رفتاری (Behavioral) به مدل پاسسیم‌کشی (PAR: Place and Route) دیدیم که خطایی از شبیه‌ساز ISim صادر شد که برای رفع آن ناچار شدیم نوع پورتهای در اعلان کامپوننت را (برخلاف آن چه که در فایل اصلی از نوع Signed تعریف کرده بودیم) به نوع Std_logic_vector تغییر دهیم.

نکته‌ای که اینجا می‌خواهم خدمتتان عرض کنم این است که حسب ظاهر امر، در شبیه‌سازی PAR ناچاراً باید اعلان کامپوننت‌ها مبتنی بر نوع Std_logic_vector انجام و در شبیه‌سازی رفتاری هم ناچاراً باید همان نوع اولیه استفاده شده در تعریف پورتهای (در مثال ما، یعنی نوع Signed) استفاده شود. این مطلب را در فایل بستر آزمایش (Testbench) زیر با توضیحات سبزنگ مشخص کرده‌ام. فایل زیر برای شبیه‌سازی PAR تنظیم شده است. اگر بخواهیم آن را برای شبیه‌سازی رفتاری استفاده کنیم، باید خطوطی که فعلاً سبزنگ (یعنی توضیح) شده‌اند (خطوط ۲۸ تا ۳۹، ۴۴ تا ۴۵، ۵۱ تا ۵۳، ۱۱۵ تا ۱۱۶) را فعال و خطوط فعال (خطوط ۱۳ تا ۲۳، ۴۶ تا ۴۷، ۵۴ تا ۵۶، ۱۱۷ تا ۱۱۸) را سبزنگ (توضیح) کنیم.

```
1  LIBRARY ieee;
2  USE ieee.std_logic_1164.ALL;
3  USE ieee.numeric_std.ALL;
4
5  ENTITY MyTB IS
6  END MyTB;
7
8  ARCHITECTURE behavior OF MyTB IS
9
10     -- Component Declaration for the Unit Under Test (UUT)
11
12     -- The following lines only works in Post-Route Simulation !!!
13     COMPONENT MyAddMuliply
14     PORT (
15         A : IN std_logic_vector(3 downto 0);
16         B : IN std_logic_vector(3 downto 0);
17         clk : IN std_logic;
18         Sum : OUT std_logic_vector(4 downto 0);
19         S : OUT std_logic_vector(3 downto 0);
20         Cout : OUT std_logic;
21         Prod : OUT std_logic_vector(7 downto 0)
22     );
23     END COMPONENT;
24
25
26     -- The following lines only works in Behavioral Simulation !!!
27     --component MyAddMuliply
28     --    Port ( A          : in          Signed (3 downto 0);
```

```

29      B      : in      Signed (3 downto 0);
30
31      Sum   : out      Signed (4 downto 0);
32      Sum   : inout    Signed (4 downto 0);
33
34
35      S      : out      Signed (3 downto 0);
36      Cout   : out      STD_LOGIC;
37      clk    : in      STD_LOGIC;
38      Prod   : out      Signed (7 downto 0));
39 --end component;
40
41
42
43      --Inputs
44      -- signal A : signed(3 downto 0) := (others => '0');
45      -- signal B : signed(3 downto 0) := (others => '0');
46      signal A : std_logic_vector(3 downto 0) := (others => '0');
47      signal B : std_logic_vector(3 downto 0) := (others => '0');
48      signal clk : std_logic := '0';
49
50      --Outputs
51      -- signal Sum : signed(4 downto 0);
52      -- signal S : signed(3 downto 0);
53      -- signal Prod : signed(7 downto 0);
54      signal Sum : std_logic_vector(4 downto 0);
55      signal S : std_logic_vector(3 downto 0);
56      signal Prod : std_logic_vector(7 downto 0);
57      signal Cout : std_logic;
58
59      -- Clock period definitions
60      constant clk_period : time := 20 ns;
61
62 BEGIN
63
64      -- Instantiate the Unit Under Test (UUT)
65      uut: MyAddMulipy PORT MAP (
66          A => A,
67          B => B,
68          Sum => Sum,
69          S => S,
70          Cout => Cout,
71          clk => clk,
72          Prod => Prod
73      );
74
75      -- Clock process definitions
76      clk_process :process
77      begin
78          clk <= '0';
79          wait for clk_period/2;
80          clk <= '1';
81          wait for clk_period/2;
82      end process;
83
84

```

```

85     -- Stimulus process
86     stim_proc: process
87     begin
88         -- hold reset state for 100 ns.
89         wait for clk_period*4;
90         A <= "0111";
91         B <= "0011";
92
93
94         wait for clk_period*4;
95         A <= "1111";
96         B <= "1011";
97
98
99         wait for clk_period*4;
100        A <= "1111";
101        B <= "0011";
102
103
104        wait for clk_period*4;
105        A <= "1100";
106        B <= "0101";
107
108
109        wait for clk_period*4;
110        A <= "0111";
111        B <= "1000";
112
113
114        wait for clk_period*4;
115        -- A <= to_signed(6,4);
116        -- B <= to_signed(-7,4);
117        A <= std_logic_vector(to_signed(6,4));
118        B <= std_logic_vector(to_signed(-7,4));
119
120
121        wait;
122    end process;
123
124 END;
125

```