

به نام خدا



دانشگاه شهید چمران اهواز

گزارش کار دستور کار شماره 2

ارائه دهندگان:

زهره ابوعلی شمشیری

امین چیت سازان

نام استاد: مهندس سید مجتبی موسوی

مدار تمام تفریق کننده تک بیتی: مداری است که شامل سه ورودی (A و B و Bin) و دو خروجی Bout, D است. این مدار دو عدد تک بیتی را با هم تفریق می کند.

جدول درستی تمام تفریق کننده تک بیتی :

A	B	Bin	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$D = A \text{ Xor } B \text{ Xor } Bin$$

$$Bout = \text{not } A \text{ And } B \text{ Or } (((\text{not}(A \text{ Xor } B \text{ Xor})) \text{ And } Bin))$$

مدار تمام جمع کننده تک بیتی: مداری است که شامل سه ورودی (دو عدد تک بیتی و یک Carry input که برای دریافت و اعمال بیت Carry از مراحل قبل) و دو خروجی S و carry output است. به عبارتی تمام جمع کننده تک بیتی سه ورودی تک بیتی A, B, C in را دریافت کرده و خروجی های Cout و sum را میدهد. در واقع این تمام تجمع کننده دو عدد تک بیتی را با هم جمع می کند.

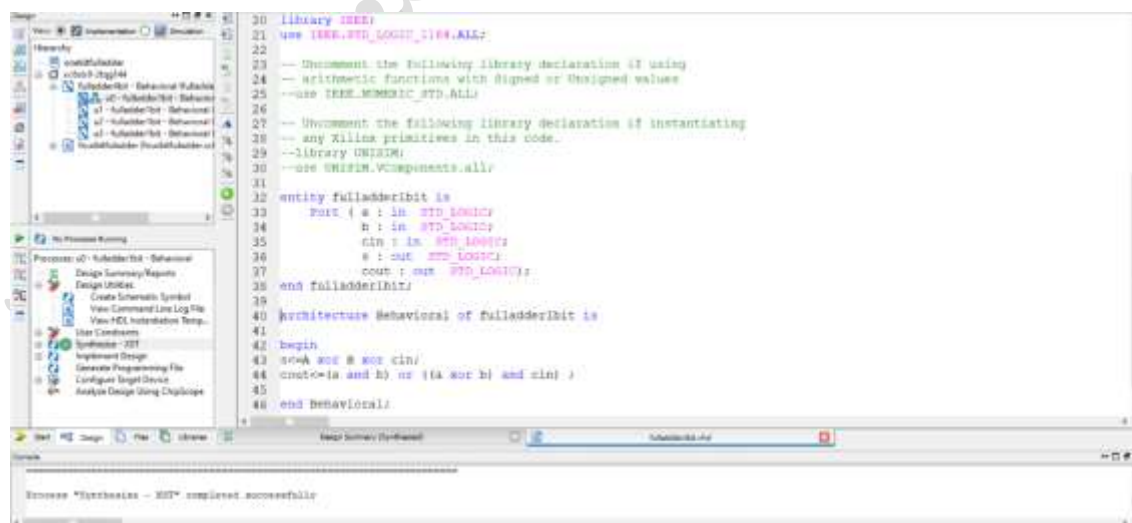
جدول درستی تمام جمع کننده تک بیتی:

A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$S = A \text{ Xor } B \text{ Xor } Cin$

$Cout = A \text{ and } B \text{ Or } (A \text{ Xor } B \text{ and } Cin)$

کد تمام جمع کننده تک بیتی:



```

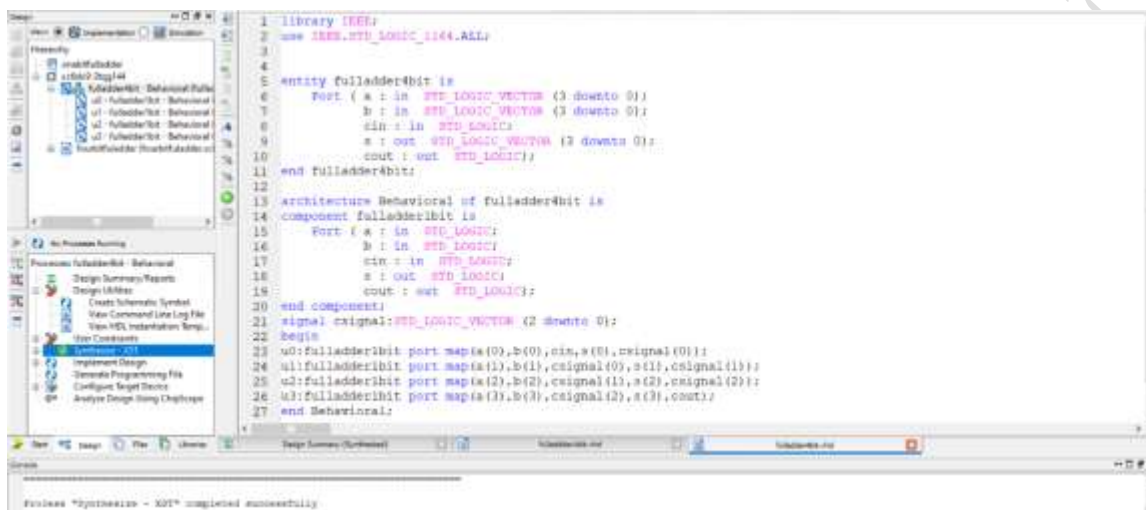
30 library IEEE;
31 use IEEE.STD_LOGIC_1164.ALL;
32
33 -- Uncomment the following library declaration if using
34 -- arithmetic functions with Signed or Unsigned values
35 --use IEEE.NUMERIC_STD.ALL;
36
37 -- Uncomment the following library declaration if instantiating
38 -- any Xilinx primitives in this code.
39 --library UNISIM;
40 --use UNISIM.VComponents.all;
41
42 entity fulladder1bit is
43     Port ( a : in STD_LOGIC;
44           b : in STD_LOGIC;
45           cin : in STD_LOGIC;
46           s : out STD_LOGIC;
47           cout : out STD_LOGIC );
48 end fulladder1bit;
49
50 architecture Behavioral of fulladder1bit is
51
52     begin
53         s <= a xor b xor cin;
54         cout <= (a and b) or (a xor b and cin);
55     end Behavioral;
56

```

تصویر 1- کد VHDL تمام جمع کننده تک بیتی

کد تمام جمع کننده 4 بیتی:

برای نوشتن کد تمام جمع کننده 4 بیتی باید ماژول تک بیتی را در قالب component به ماژول فعلی اضافه کنیم.

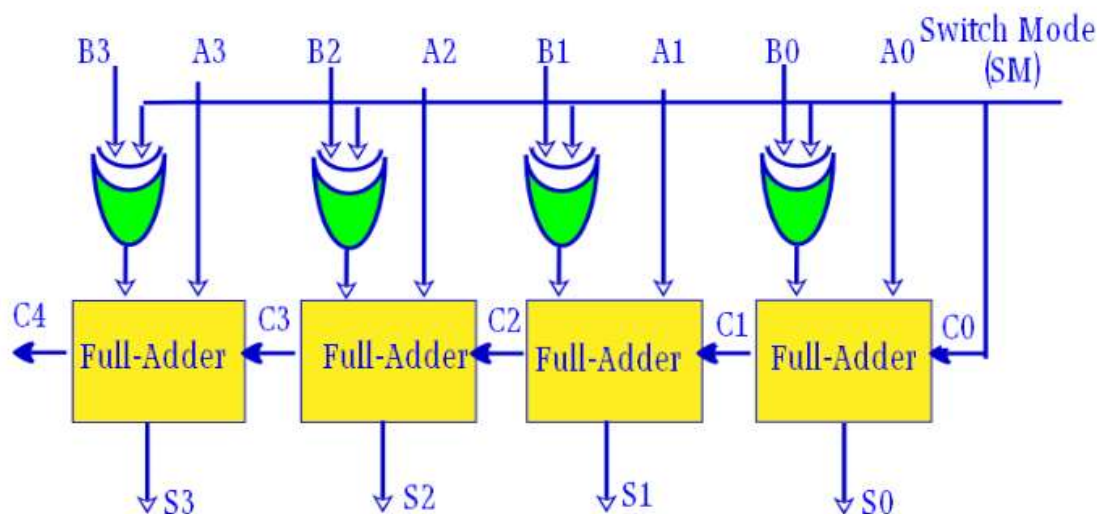


تصویر 2- کد تمام جمع کننده 4 بیتی

تفاوت library, entity, architecture: library مجموعه ای از کامپوننت ها و توابع از پیش تعریف شده است که برای طراحی مدار استفاده میشوند و بنابراین ما در بخش library ما کتابخانه مورد نیاز خود را استفاده میکنیم. در بخش entity ورودی ها و خروجی های مدار مشخص میشوند. برای مثال در کد تصویر 2، a, b, cin ورودی های مدار ما هستند و s, cout خروجی های ما هستند. در بخش architecture توابع یا عملکرد مدار را بیان میکنیم. در کد تصویر 2 در بخش architecture ماژول جمع کننده تک بیتی را در قالب کامپوننت به ماژول فعلی اضافه کردیم و از آن استفاده میکنیم و 4 جمع کننده تک بیتی ایجاد کرده و ورودی و خروجی های آنها را با پورت مپ به ماژول اصلی جمع کننده تک بیتی متصل میکنیم.

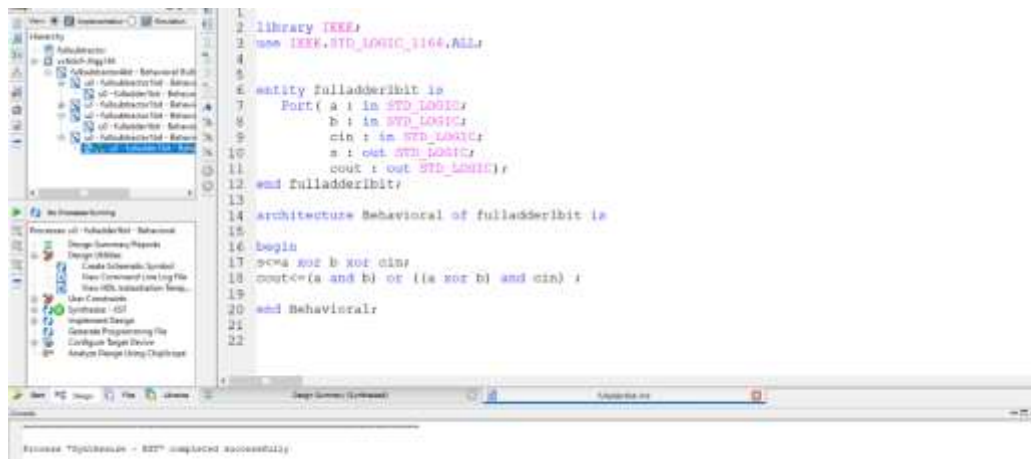
مزیت استفاده از component: استفاده از ماژول های دیگر در ماژول فعلی است.

ترسیم مدار تفریق کننده 4 بیتی با استفاده از جمع کننده: برای رسم این مدار، ما ابتدا یک Switch Mode تعریف میکنیم که C_{in} اولین جمع کننده ما نیز هست. سپس این سوئیچ را با تمام b ها xor میکنیم که نتیجه این ها مشخص میشود که عمل جمع صورت گیرد یا تفریق.

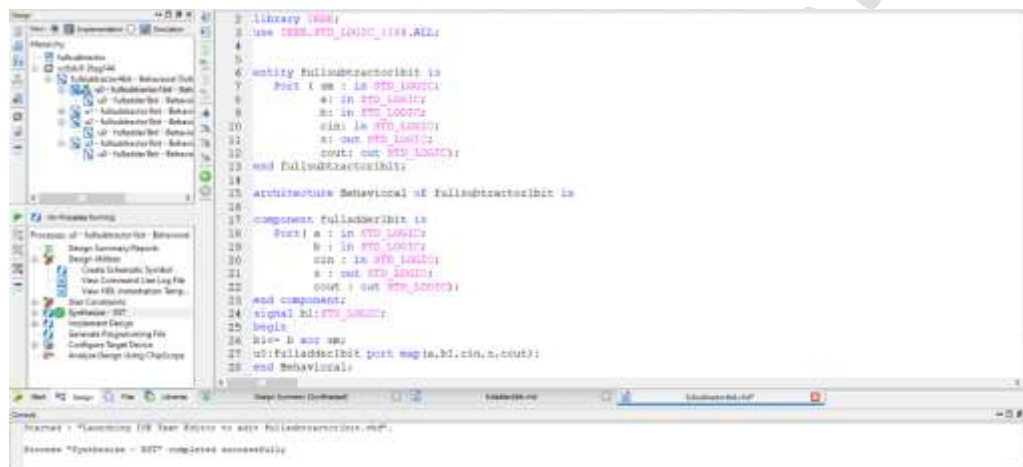


تصویر 3- شماتیک تفریق کننده جمع کننده 4 بیتی

کد تفریق کننده جمع کننده 4 بیتی: ابتدا کد ماژول جمع کننده تک بیتی را مینویسیم (تصویر 4) سپس یک ماژول دیگر برا تفریق کننده جمع کننده تک بیتی میسازیم و ماژول جمع کننده تک بیتی را در این ماژول استفاده میکنیم (با استفاده از component) سپس یک سوئیچ مود تعریف می کنیم و آن را با b xor می کنیم (تصویر 5). سپس یک ماژول دیگر برای تفریق کننده جمع کننده 4 بیتی میسازیم و ماژول تفریق کننده جمع کننده تک بیتی را در آن استفاده می کنیم به گونه ای که به تعداد 4 تا در این ماژول استفاده و پورت مپ میکنیم (تصویر 6)



تصویر 4- کد تمام جمع کننده تک بیتی



تصویر 5 - تفریق کننده جمع کننده تک بیتی



تصویر 6 - تفریق کننده جمع کننده 4 بیتی