

به نام خدا

آموزش VHDL

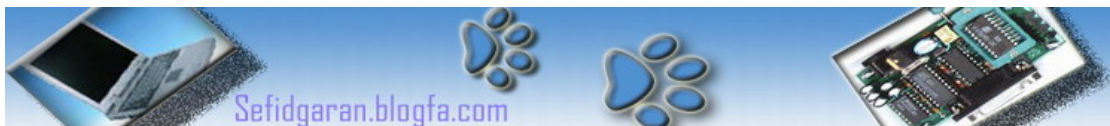
(قسمت چهارم)

Very High Speed Integrated Circuit Hardware Description Language (Part 4)

تحلیل یک برنامه VHDL که در درس قبلی به عنوان تمرین مطرح شد و آشنایی با چند
دستور دیگر:

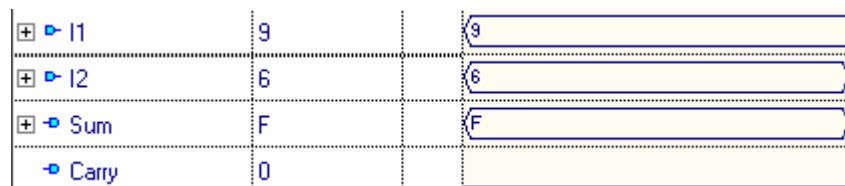
برنامه یک Full Adder چهار بیتی :

```
library IEEE ;
use IEEE.STD_LOGIC_1164.all ;
entity Adder is
port(
I1 : in STD_LOGIC_VECTOR(3 downto 0) ;
I2 : in STD_LOGIC_VECTOR(3 downto 0) ;
Carry : out STD_LOGIC;
Sum : out STD_LOGIC_VECTOR(3 downto 0)
) ;
end Adder;
architecture FullAdder of Adder is
signal c1,c2,c3:std_logic ;
begin
c1<=i1(0) and i2(0) ;
sum(0)<=i1(0) xor i2(0) ;
c2<=(i1(1)and i2(1)) or (i1(1)and c1) or (i2(1)and c1) ;
sum(1)<=i1(1) xor i2(1) xor c1 ;
c3<=(i1(2)and i2(2)) or (i1(2)and c2) or (i2(2)and c2) ;
sum(2)<=i1(2) xor i2(2) xor c2 ;
carry<=(i1(3)and i2(3)) or (i1(3)and c3) or (i2(3)and c3) ;
sum(3)<=i1(3) xor i2(3) xor c3 ;
end FullAdder ;
```



نکته ای درمورد نوع داده ی STD_LOGIC :

شما می توانید به جای نوع داده ی Bit از آن استفاده کنید زیرا مزایایی دارد که در طول دروس بعدی متوجه آنها خواهید شد پس از این به بعد همیشه به جای Bit از STD_LOGIC استفاده خواهیم کرد. در برنامه فوق دو ورودی داریم از نوع Bus و یک خروجی از نوع Wire و یک خروجی از نوع Bus. شکل زیر حاصل تست برنامه جمع کننده فوق در Wave Form نرم افزار Active VHDL است:



می دانید که حاصل جمع دو بیت را می توانیم توسط XOR بدست آوریم و مقدار Carry هم با And کردن دو بیت بدست می آید اگر دوست دارید با این مسئله در گیر شوید برنامه Full Adder هشت بیتی را بنویسید. فکر نمی کنم با خواندن برنامه بالا چیزی دستگیرتان شود فقط باید خودتان زحمت بکشید. البته برنامه فوق یک برنامه Structural است و به ساده ترین روش نوشته شده است.

این درس همین جا تمام شد و قسمت اعظم کار با شماست یعنی برنامه ی فوق را خط به خط تریس کنید تا کاملا متوجه شوید (البته دوستانی که ضعیفند)

می توانید برنامه بالا رو راحت کپی کنید در نرم افزار Active VHDL سپس کامپایل کرده و به شکل Wave که در بالا آوردم برسید. به متغیرها همونطور که من عدد دادم عدد بدین و اونو اجرا کنید. اگر با این نرم افزار آشنا نیستید الان فرصت خوبی برای آشنایی!

بدرود