



جزئیات کد پروژه BinaryAdder_4bit

```
module BinaryAdder_4bit (  
    input [3:0] A,  
    input [3:0] B,  
    input C0,  
    output [3:0] S,  
    output C4  
);  
  
    wire C1, C2, C3;
```

در پروژه BinaryAdder_4bit یک ماژول اصلی با نام BinaryAdder_4bit داریم که این ماژول دارای سه ورودی به نام‌های A، B و C0 و دو خروجی به نام‌های S و C4 است. همچنین سه سیگنال واسطه‌ای نیز مشاهده می‌شود که آن‌ها را C1، C2، C3 و از نوع wire

تعریف می‌کنیم. در داخل ماژول BinaryAdder_4bit چهار FullAdder ۱ بیتی قرار دارد. بنابراین، در ماژول BinaryAdder_4bit ورودی‌های A و B را ۴ بیتی تعریف کرده و ورودی C0 را ۱ بیتی در نظر می‌گیریم. از سوی دیگر، خروجی S را ۴ بیتی در نظر گرفته و C4 را ۱ بیتی تعریف می‌کنیم. همان‌طور که گفته شد، در

```
module FullAdder_1bit (  
    input X,  
    input Y,  
    input Cin,  
    output S,  
    output Cout  
);  
  
    wire W1, W2, W3;  
  
    xor G1 (W1, X, Y);  
    xor G2 (S, W1, Cin);  
    and G3 (W2, W1, Cin);  
    and G4 (W3, X, Y);  
    or G5 (Cout, W2, W3);  
  
endmodule
```

داخل ماژول BinaryAdder_4bit از چهار FullAdder استفاده شده است. دو راه پیش رو داریم؛ یا باید هر بار، FullAdder را به‌طور کامل تعریف کنیم یا اینکه یک بار ماژول FullAdder را تعریف کرده و در صورت نیاز، از روش فراخوانی ماژول برای به کار بردن FullAdder استفاده کنیم. نخست نیاز است که ماژولی با عنوان FullAdder_1bit را تعریف کنیم. در این ماژول، سه ورودی ۱ بیتی X، Y و Cin و دو خروجی ۱ بیتی S و Cout داریم. همچنین در این ماژول سه سیگنال واسطه‌ای W1، W2 و W3

از نوع wire تعریف می‌کنیم. ماژول FullAdder_1bit از دو گیت XOR، دو گیت AND و یک گیت OR تشکیل شده است. می‌توان از روش تعریف گیت‌ها و یا روش توصیفی برای این ماژول استفاده کرد که در اینجا از روش تعریف گیت‌ها استفاده شده است. در xor G1 دو مقدار X و Y با هم XOR شده و نتیجه به W1 می‌رود. در xor G2 دو مقدار Cin و W1 با هم xor شده و نتیجه در S که یکی از خروجی‌های ماژول FullAdder_1bit است، قرار می‌گیرد. در and G3 دو مقدار Cin و W1 با هم and می‌شوند و نتیجه در

W2 قرار می‌گیرد. در G4 and دو مقدار X و Y با هم and شده و نتیجه به W3 می‌رود. در نهایت، در or W3، G5 و W2 با هم OR شده و Cout که خروجی دیگر مازول است، حاصل می‌شود.

در ادامه به سراغ FullAdder ها می‌رویم. در مازول BianryAdder_4bit باید چهار بار FullAdder را تعریف کنیم. همان‌طور که گفته شد، از روش فراخوانی برای تعریف آن‌ها استفاده می‌کنیم. FA هایی که قرار است به روش فراخوانی مازول تعریف شوند، به ترتیب با نام‌های مستعار FullAdder_1bit U1، FullAdder_1bit U2، FullAdder_1bit U3 و FullAdder_1bit U4 می‌نویسیم. حال داخل پرانتز ورودی و خروجی‌های مازول فراخوانی‌شده را می‌نویسیم و پیش از هر کدام نقطه می‌گذاریم. سپس، باید تعیین کرد که کدام سیگنال‌های مازول جدید به این مازول فراخوانی‌شده داده شوند. برای نمونه، در FullAdder_1bit U1 ورودی‌های A[0]، B[0] و C0 به FullAdder_1bit داده شده‌اند تا محاسبات لازم را انجام دهند و خروجی‌های S[0] و C1 مشخص شوند. همین روند برای FullAdder_1bit U2 و FullAdder_1bit U3 نیز صدق می‌کند. در پایان،

در FullAdder_1bit U4 ورودی‌های A[3]، B[3] و C3 به FullAdder_1bit داده شده‌اند تا محاسبات انجام گرفته و خروجی S[3] و C4 مشخص شوند.

با سپاس از استاد گرامی، جناب آقای بهروز طاهری

با احترام، رضانی زیارانی

خرداد ۱۴۰۳