

توضیحات مهم

- برای سوالات تشریحی یک فایل pdf بارگذاری نمایید.
- فایل‌های مربوط به هر سوال برنامه‌نویسی را بدون قرار دادن در پوشه zip کرده و در [کوئرا](#) بارگذاری کنید.
- فرمت‌های فشرده دیگر مانند rar و gzip قابل داوری نیست.
- در فایل‌های مربوط به یک سوال، دو ماژول با نام یکسان تعریف نکنید.
- نام ماژول‌ها و پورت‌ها را دقیقاً مانند صورت سوال تعریف کنید.
- در تعریف ماژول‌ها غیر از مواردی که صراحتاً ذکر شده‌است از تاخیر استفاده نکنید.
- ابهامات را در کوئرا مطرح نمایید.

۱- نمودار زمانی کد زیر را با شبیه‌سازی در Modelsim رسم کرده و گزارشی از چگونگی تغییرات و عملکرد سیگنال‌های کد، همراه با تصویر شکل موج حاصل بنویسید. شکل موج باید شامل تمام سیم‌ها و رجیسترها باشد. در صورت مشاهده Race آن را ذکر کنید و دلیل آن را توضیح دهید.

```
module test_wave;
    wire out;
    reg a, b, c, d;
    wire e, f, g;
    or #(2,3) (e, a, b);
    and #(3,4) (f, e, c);
    nor #(3,5) (g, f, d);
    xor #(4,6) (out, e, g);
    initial
    begin
        a = 1'b1;
        c = 1'b1;
        #3 c = 1'b0;
        #2 c = 1'b1;
        #8 a = 1'b0;
        b = 1'b0;
        #6 d = 1'b0;
    end
endmodule
```

۲- برای قطعه کد زیر مقدار C را در بلاک initial به‌گونه‌ای مقداردهی کنید که شکل موج A با شکل موج B متفاوت شود. سپس تصویر شکل موج‌ها را به همراه توضیحات خود در فایل PDF درج نمایید.

```
module top;
    reg C;
    wire #5 A;
    assign #10 A = C;
    wire #10 B;
    assign #5 B = C;
    initial
    begin

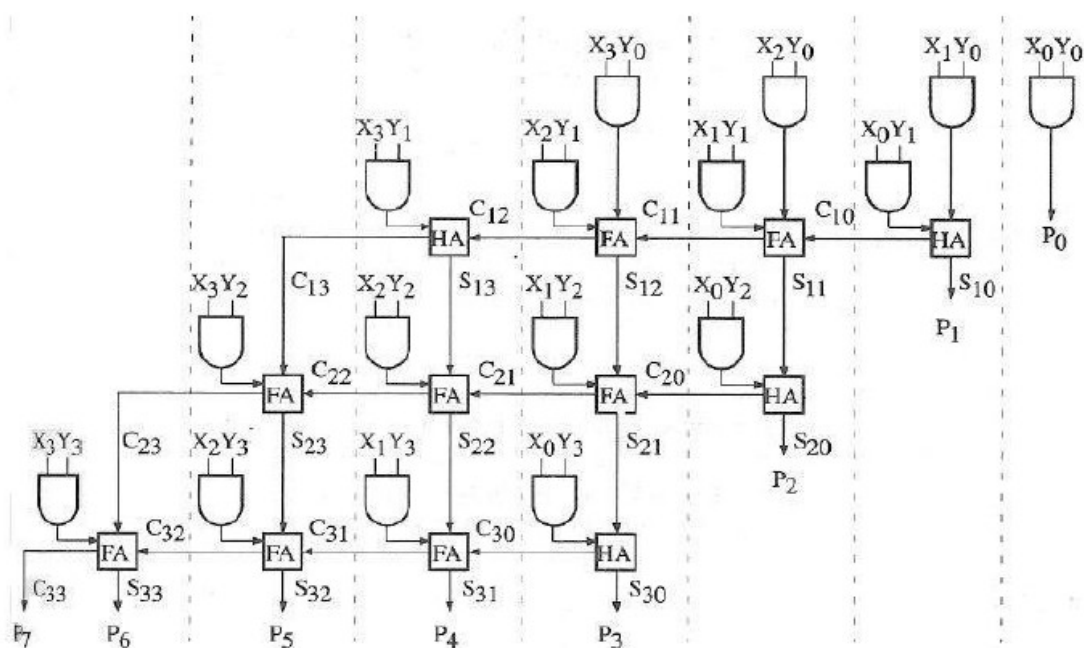
    end
endmodule
```

۳- با استفاده از کدهایی که برای half adder و full adder در اختیاران قرار داده شده است و گیت‌های پایه یک ضرب‌کننده بدون علامت n بیتی را با توجه به مدار داده شده طراحی کنید.

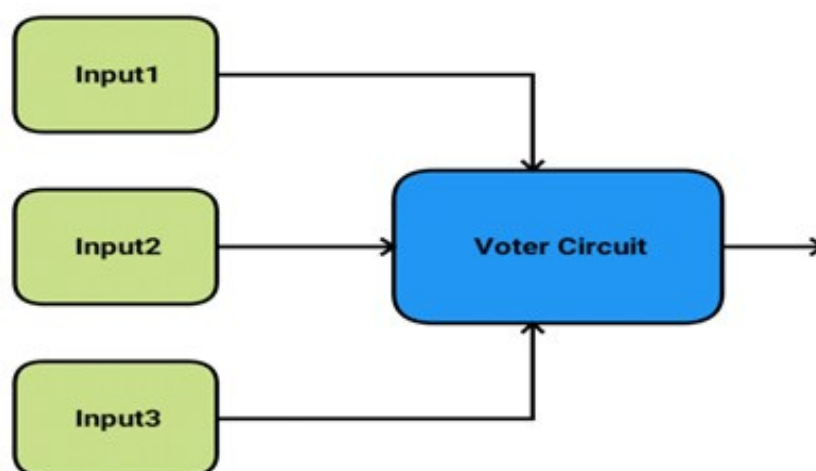
```
module unsigned_multiplier(result, a, b);
    parameter N = 4; //a and b each have N bits so the result should have 2*N bits
```

استفاده از generate block بلامانع است و استفاده از توصیف رفتاری مجاز نیست.

(شکل مربوط به ضرب‌کننده ۴ بیتی بدون علامت است)



۴- همواره احتمال عملکرد اشتباه یک سخت‌افزار به دلایل مختلف مثل نوسان منبع تغذیه، برخورد ذرات آلفا و ... وجود دارد. یکی از راه‌ها برای کاهش اثرات این مشکل استفاده از سه سخت‌افزار یکسان به جای یکی است چرا که احتمال خراب شدن دو سخت‌افزار به طور همزمان بسیار کمتر از احتمال خراب شدن یکی از آن‌ها است. در این حالت بین سه خروجی تولید شده از سه سخت‌افزار یکسان، موردی انتخاب می‌شود که حداقل ۲ بار تکرار شده باشد و در صورتی که چنین خروجی وجود نداشته باشد خروجی error فعال می‌شود. به ماژولی با این عملکرد، voter را رای‌گیر گفته می‌شود. برای مثال اگر ما از یک واحد ضرب‌کننده در طراحی خود استفاده کنیم در صورت رخداد اشکال، خروجی ما دچار اشکال می‌شود اما اگر از سه مدار مشابه استفاده کنیم اگر یکی از خروجی یکی از ضرب‌کننده‌ها دچار مشکل شد و دو ضرب‌کننده دیگر مقدار درست را محاسبه کردند در نهایت رأی‌گیر مقدار درست را انتخاب می‌کند (بیشترین ورودی تکرار شده از بین ورودی‌ها) و در نهایت مقدار درست را بر روی خروجی قرار می‌دهد. شکل این طراحی در ادامه قرار داده شده است.



شکل ۱: دیاگرام مربوط به روش بالا

برای این سوال یک رأی‌گیر را سطح Gate طراحی کنید به این صورت که رأی‌گیر از بین سه ورودی ۲ بیتی بیشترین مقدار تکرار شده (Majority) را انتخاب و بر روی خروجی خود قرار دهد. ماژول خود را مشابه مثال زیر نامگذاری کنید. این رأی‌گیر دارای یک سیگنال error نیز هست تا در صورت مشاهده سه مقدار متفاوت آن را یک کند.

```

module voter (in1,in2,in3,out,error);
  input [1:0] in1,in2,in3;
  output [1:0] out;
  output error;
endmodule
  
```