

## توضیحات مهم

- برای سوالات تشریحی یک فایل pdf بارگذاری نمایید.
- 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 شده است از تاخیر استفاده نکنید.
- ابهامات را در کوئرا مطرح نمایید.

۱- الف) توابع (Functions) و وظایف (Tasks) در وریلاگ چه تفاوت‌هایی دارند؟ (۴مورد)

ب) آیا در توابع وریلاگ امکان پیاده‌سازی توابع بازگشتی وجود دارد؟ اگر بله توضیح دهید.

پ) اگر یک task را بدون حالت automatic ایجاد کنیم در صورت فراخواندن همزمان آن از مکان‌های مختلفی در برنامه چه مشکلی می‌تواند ایجاد شود؟ توضیح دهید.

۲- الف) یک تابع وریلاگ (function) بنویسید که با دریافت یک وکتور N بیتی، مکمل دوم آن را محاسبه کند. تابع را به نحوی پیاده‌سازی کنید که صدا کردن آن به شکل `complement_two(input_vector, length)` صورت بگیرد که در آن `input_vector` عدد داده‌شده و `length` تعداد بیت‌های آن است. می‌توانید N را به عنوان یک پارامتر گلوبال فرض کرده و از آن استفاده کنید. پیاده‌سازی خود را با یک روش بیت‌به‌بیت و با استفاده از حلقه صورت دهید.

ب) ماژول تستی بنویسید که به ازای ورودی‌های مختلف عملکرد تابع نوشته‌شده را تست کند. کدها و شکل موج‌های به‌دست آمده را گزارش کنید.

۳- ورودی ۸ بیتی A را در نظر بگیرید. ماژولی با استفاده از توصیف رفتاری بنویسید که رفتار آن به این نحو باشد که شماره پرارزش‌ترین بیت A که ۱ می‌باشد را در خروجی Y نشان دهد، در صورتی که A صفر باشد باید خروجی NONE یک شود. برای مثال در صورتی که ورودی A مقدار ۷ و یا ۴ را داشته باشد خروجی Y باید مقدار ۲ را نمایش بدهد. (توجه کنید در مدار شما هیچ latch ای به وجود نیاید در غیر این صورت حتی در صورت کسب امتیاز کامل داوری نمره از دست خواهید داد). تعریف مدارتان باید به شکل روبرو باشد.

```
module first_one(A, Y, NONE);
    input wire [ 7 : 0 ] A;
    output reg [ 2 : 0 ] Y;
    output reg NONE;
```

(ادامه در صفحه‌ی بعد)

۴- یکی از عملیات‌های پرهزینه در رمزنگاری، عمل ضرب با اندازه‌های ورودی بزرگ می‌باشد. در صورتی که این ضرب‌ها به صورت ترکیبی و در یک کلاک انجام شوند، فرکانس مدار به شدت افت می‌کند. به این منظور، سازندگان بوردهای FPGA، واحدهای از پیش طراحی شده‌ای را بر روی مدارات خود قرار می‌دهند (برای مثال DSP48E در بوردهای شرکت Xilinx) که می‌توانند ضرب‌های با اندازه کوچک (مثلاً ۱۷ در ۲۴ بیت) را با فرکانس بالایی به صورت ترکیبی و در یک کلاک انجام دهند. بنابراین راهکار اصلی برای انجام ضرب‌های بزرگ، شکستن آن‌ها به ضرب‌های کوچکتر و انجام آنها با استفاده از این واحدها می‌باشد.

در این تمرین شما باید با استفاده از یک ضرب‌کننده ۳۲ بیتی ترکیبی که کد آن در اختیارتان قرار می‌گیرد یک ضرب‌کننده با طول ورودی  $N$  (مضربی از ۳۲) و خروجی به طول  $2N$  بیت را پیاده‌سازی می‌کنید. به این نکته دقت کنید که تنها مجاز به یک بار استفاده از ضرب‌کننده ۳۲ بیتی در کد خود هستید و طبیعتاً در هر کلاک یک ضرب انجام می‌دهید تا همه ضرب‌ها بر روی آن انجام شوند. از زمانی که ورودی `start` یک شود، ورودی‌های ضرب معتبر هستند و می‌توانید ضرب را آغاز کنید و به محض آماده شدن خروجی، سیگنال `ready` را ۱ کنید.

سرخ‌خط مازول طراحی شده به شکل زیر است.

**مهم:** توجه داشته باشید که کد شما باید قابل سنتز باشد

```
module multiplier(clock, start, in1, in2, ready, out);
    parameter N = 128;

    input clock, start;
    input [N-1:0] in1, in2;

    output reg ready;
    output reg [2*N-1:0] out;
```