

به نام خدا

درس طراحی سیستم‌های مبتنی بر ASIC/FPGA

دکتر مهدی شعبانی

دانشکده مهندسی برق

دانشگاه صنعتی شریف

تمرین سری پنجم

نیمسال دوم 1402-1403

-
- برای تحویل تمرین، تمامی فایل‌ها از جمله ماژول‌ها و تست‌بنچ‌ها به همراه گزارشی از نحوه‌ی انجام کار به همراه نتایج مربوطه را در سامانه آپلود کنید. برای تحویل کدها، فقط فایل‌های مربوط به ماژول‌ها و تست‌بنچ‌ها را در فولدرهای جداگانه قرار دهید (هر فولدر مربوط به هر سؤال و هر بخش نیز در فولدر جداگانه). برای سؤالات دارای فایل‌های اضافی مثل فایل مموری یا txt آن‌ها را نیز تحویل دهید. از ارسال کل پروژه پرهیز کنید.

- کد تحویل داده شده توسط شما باید قابل سنتز و شبیه‌سازی باشد و در صورت این که کد سنتز و شبیه‌سازی نشود، نمره‌ای به آن تعلق نمی‌گیرد.
 - مشورت و کمک گرفتن از یک‌دیگر، جستجو در اینترنت و کتاب‌ها و.... کاملاً جایز می‌باشد ولی تمرین باید توسط خود شما انجام شود. در صورت مشاهده شباهت غیرعادی نمره سؤال برای همه‌ی افراد کاملاً صفر در نظر گرفته می‌شود.
 - سعی کنید در تمرین برنامه‌نویسی، هم از سطح رفتاری و هم سطح جریان داده استفاده کنید تا به هر دو سطح مسلط شوید.
 - توجه کنید که برای هر سؤال باید یک گزارش از نحوه‌ی انجام کار به همراه بلوک دیاگرام ساختار پیاده‌سازی شده به همراه توضیحات خواسته شده و نتایج شبیه‌سازی بیان و صحت عملکرد با استفاده از تست بنچ تأیید شود.
 - این تمرین برای تمرین کدزنی و شبیه‌سازی شما در نظر گرفته شده است و صرفاً صورت سؤالات طولانی است و هدف اصلی افزایش تسلط شما در کدزنی می‌باشد؛ از انجام این تمرین نهایت لذت را ببرید!
-

1- فرض کنید یک ماژول در اختیار شما قرار داده شده است تا آن را صحت‌سنجی کنید. برای صحت‌سنجی این ماژول باید به عنوان ورودی، سیگنال‌های مختلفی ساخته شود و به آن اعمال شود. ورودی این ماژول 8 بیتی است.

(الف) تست‌بنچی بنویسید شکل موج دندان‌اره‌ای بسازد.

(ب) تست‌بنچی بنویسید شکل موج دندان‌ه مثلثی بسازد.

(ج) با استفاده از یک DDS یک شکل موج سینوسی تولید کنید. فرکانس را دلخواه در نظر بگیرید.

(د) حال با استفاده از فایل، یک شکل موج سینوسی تولید کنید. برای این کار لازم است تا یک سیگنال سینوسی با فرکانس دلخواه با تعداد نمونه‌های 50 هزارنمونه توسط متلب تولید، fixed point شود و روی یک فایل نوشته شود؛ سپس در تست‌بنج این فایل را خوانده و نمایش دهید. در این مرحله لازم است تا وقتی به انتهای فایل رسید، یک پیغام چاپ شود. اطلاعات Fixed-point را مطابق قسمت قبل در نظر بگیرید؛ این اطلاعات را می‌توانید در GUI مربوط به IP Core مشاهده کنید.

(ه) (امتیازی) با استفاده از System Task‌های زبان برنامه‌نویسی ورایلاگ (در صورت وجود) یا سیستم ورایلاگ، یک شکل موج سینوسی با فرکانس دلخواه تولید کنید.

(و) تست‌بنج را به گونه‌ای تغییر دهید که با تغییر یک پارامتر یا رجیستر، شکل‌موج‌های قبلی تولید شود. در واقع با تغییر این رجیستر یا پارامتر یکی از شکل‌موج‌های مثلثی، دندان‌اره‌ای، سینوسی با استفاده از DDS، سینوسی با استفاده از فایل و سینوسی با استفاده از system task ساخته می‌شود و به عنوان ورودی ماژول شما اعمال می‌شود.

2- دو ماژول ضرب کننده دو عدد 8 بیتی برای شما قرار داده شده که دو ورودی به همراه valid آن‌ها به همراه سیگنال‌های reset و enable دارد. سیگنال ریست آن active-low و enable مانند clock-enable عمل می‌کند و ماژول زمانی کار می‌کند که clock-enable آن فعال باشد. این دو ماژول را با نوشتن حداقل 4 سناریوی شبیه‌سازی مختلف، صحت‌سنجی کنید و ایرادات هر کدام از آن‌ها را به تفکیک بیان کنید. در هر سناریو می‌توانید از هر دو ماژول Instance بگیرید و نتایج را مقایسه کنید.

3- فرض کنید ماژولی داریم که سه ورودی یک بیتی را دریافت می‌کند. این سه ورودی سیگنال‌های کنترلی هستند که مهم هست که ترتیب آمدن آن‌ها به چه ترتیب می‌باشد؛ ماژول دارای 6 خروجی است؛ خروجی اول بیان‌کننده اولین سیگنالی است که یک می‌شود، خروجی دوم مربوط به دومین و خروجی سوم مربوط به سومین سیگنالی است که یک می‌شود؛ یعنی اگر ترتیب یک شدن ورودی‌ها به ترتیب 2، 3 و 1 باشد، خروجی اول 2، خروجی دوم 3 و خروجی سوم 1 می‌شود. سه خروجی دیگر نیز مربوط به زمان یک شدن متناظر سه خروجی قبلی می‌باشد که این زمان تعداد clock cycle بعد از یک شدن ریست می‌باشد.

(الف) این ماژول را طراحی کنید و کد آن را بنویسید

(ب) تست‌بنچی بنویسید که در زمان 1000 نانوثانیه، ماژول را از ریست در آورد. ورودی دوم در زمان 2448 نانوثانیه، ورودی اول در زمان 5000 نانوثانیه و ورودی سوم در زمان 8741 نانوثانیه یک شود. فرکانس کلاک را 250 مگاهرتز در نظر بگیرید و نتایج اعمال این شرایط به ماژول خود را گزارش کنید و ماژول خود را صحت‌سنجی نمایید.

(ج) یک سناریوی تست دیگر برای ماژول خود نوشته و کدنوشته شده را verify کنید. در این سناریو لازم است تا ماژول شما بعد از یک شدن دو ورودی و قبل از یک شدن ورودی سوم یک بار ریست شود و سپس دوباره سه ورودی به ترتیب دلخواه یک شوند.

4- قصد داریم رشته بیت های 32 بیتی را برای یک ram ارسال کنیم. این رشته بیت های 32 بیتی را بررسی کرده و در صورت یافتن پکت های خاص، داده 32 بیتی را به آدرس و داده مناسب تقسیم بندی کرده و در نهایت ارسال می کنیم. سیگنال های ارسالی به ram عبارتند از : ram_address, ram_data, ram_en و error. داده 32 بیتی ورودی را به عنوان 8 پکت 4 بیتی در نظر می گیریم :

Header	Address	Data1	Data2	Data3	Data4	check
3-0	11-4	15-12	19-16	23-20	27-24	31-28

ماژول PacketChecker ابتدا 4 بیت اول را بررسی می کند. اگر در ابتدا Header ("1110" = 0xE) را دریافت کند بایت بعدی (8 بیت) را به عنوان آدرس و 2 بایت بعدی (16 بیت) را به عنوان دیتا در نظر می گیرد. در صورتی که به عنوان Header ("1110" = 0xE) را دریافت کند ، سیگنال ram_en یک کلاک 1 می شود و 8 بیت آدرس از طریق ram_address و 16 بیت دیتا از طریق ram_data ارسال می شود.

در غیراینصورت، سیگنال error به صورت تک کلاک 1 می شود.

جهت صحت سنجی فرایند ارسال داده و آدرس به رم، قصد داریم با استفاده از یک تست بنچ داده های 32 بیتی را تولید کنیم و به عنوان ورودی ماژول PacketChecker ارسال کنیم. به این منظور سناریو های مختلف را پیاده سازی کرده و خروجی های ماژول را در سطر های فایل های متناظر ذخیره کنید.

- در هر سطر فایل address، آدرس های مربوط به بایت Address که بر سیگنال ram_address ارسال می شوند را ذخیره کنید. توجه داشته باشید به ازای نمونه ورودی هایی که سیگنال ram_en یک نمی شود، نباید آدرسی ذخیره شود.

- در هر سطر فایل data، به طور مشابه داده های 2 بیتی ram_data را ذخیره کنید.

- در هر سطر فایل error مقادیر 0 و 1 سیگنال error را ذخیره کنید.

- 5-(الف) در مورد پروتکل ارتباطی UART تحقیق و نحوه ی سیگنالینگ فرستنده و گیرنده ی آن را تشریح نمایید.
- (ب) با استفاده از متلب یا پایتون کدی بنویسید که چند ورودی در زمان های مختلف را دریافت کند و سیگنال فرستنده UART را بسازد.
- (ج) کد متلب یا پایتونی بنویسید که سیگنال قبلی را گرفته و اطلاعات اولیه را استخراج کند؛ در واقع در این بخش شما باید کد گیرنده این پروتکل را بنویسید.
- (د) کد وریلاگ منطبق بر کدهای متلب قبلی را پیاده سازی نمایید و با شبیه سازی اولیه، کد خود را صحت سنجی کنید.
- (ه) یک Test Vector، به طول 2048 داده 8 بیتی بسازید و خروجی های مربوط به متلب را در فایلی ذخیره کنید. سپس این بردار تست را به ماژول های hdl اعمال کنید و سیگنال های خروجی ماژول های خود را نیز روی فایلی ذخیره نمایید.
- (و) حال با استفاده از رسم نتایج متناظر روی هم در محیط متلب یا پایتون، تایید کنید که ساختار پیاده سازی شده کاملاً بر کد نرم افزار منطبق است.
- (ز) حال فرستنده و گیرنده خود را به یکدیگر وصل کنید و ساختار تجمیع شده را صحت سنجی کنید. در این تست باید کد نرم افزار و سخت افزار تجمیع شده به گونه ای باشد که ورودی قرار داده شده برای فرستنده، عیناً در خروجی گیرنده ظاهر شود.

موفق باشید