به نام خدا

آزمایشگاه طراحی سامانه‌های رقمی

تابستان ۱۴۰۵

دستیار استاد: علی اثنی‌عشری

دانشکدهٔ مهندسی کامپیوتر

آزمایش هفتم | **طراحی پروتکل UART** | **مهلت ارسال: ۱ شهریور ماه**

- مهلت ارسال پاسخ تمرین تا ساعت ۷:۵۹ تعیین شده است.
- با توجه به محدودیت زمانی آزمایشگاه، طراحی و کدنویسی هر آزمایش باید پیش از جلسه انجام شده و در قالب پیش‌گزارش تحویل داده شود. در جلسهٔ آزمایشگاه فقط پیاده‌سازی طرح روی بورد FPGA انجام می‌شود.
- بارم‌بندی کلی هر آزمایش به شرح زیر است: ۳۵ درصد نمره برای پیش‌گزارش، ۱۰ درصد برای حضور در کلاس، ۳۵ درصد برای پیاده‌سازی روی بورد و ۲۰ درصد برای گزارش کار نهایی در نظر گرفته شده است.
  پیش‌گزارش باید شامل گزارش متنی کامل مراحل انجام آزمایش و پروژهٔ نرم‌افزار Quartus با طراحی قابل سنتز باشد.
  نمرهٔ بخش پیاده‌سازی روی بورد از آزمایش سوم به بعد اعمال می‌شود.
  گزارش کار نهایی نیز باید شامل نسخهٔ نهایی و کامل پروژه، توضیحات دقیق دربارهٔ الگوریتم مورد استفاده و تصمیم‌های خاص اتخاذشده به دلیل محدودیت‌های بورد باشد؛ برای نمونه، محدودیت در وارد کردن یک عدد پنج‌رقمی BCD باید به‌طور کامل توضیح داده شود.
- در صورت غیبت دانشجو در یک جلسه، علاوه بر ۱۰ درصد نمرهٔ حضور، ۳۵ درصد نمرهٔ پیاده‌سازی نیز کسر خواهد شد؛ بنابراین، مجموع نمرهٔ کسرشده برابر با ۴۵ درصد است.
  در شرایط بسیار خاص، دانشجو می‌تواند با هماهنگی قبلی، جلسهٔ غیبت‌شده را در جلسه‌ای دیگر جبران کند. در این حالت، نمایش و اجرای طرح روی بورد FPGA الزامی است.
- پیش‌گزارشی که با تأخیر ارسال شود، فاقد ارزش نمره‌ای خواهد بود. همچنین، به‌ازای هر روز تأخیر در ارسال گزارش کار نهایی، ۳۰ درصد از نمرهٔ آن کسر می‌شود. گزارش کار نهایی حداکثر تا دو روز پس از مهلت تعیین‌شده قابل ارسال است.
- برای نگارش گزارش کار، حتماً از قالب **Course Project Report Template** موجود در **این پیوند** استفاده فرمایید.
- پاسخ‌های خود را به‌صورت تایپ‌شده ارسال کرده و حتماً نکات نگارشی را رعایت فرمایید.
- در سؤال‌هایی که نیازمند برنامه‌نویسی هستند، فایل کد کامنت‌گذاری‌شده و فایل آزمون را نیز همراه با گزارش ارسال فرمایید.
- تمام فایل‌های خود را در یک فایل فشرده با پسوند **zip** قرار دهید. نام فایل باید مطابق قالب زیر باشد:

Hw7_[StudentId]_[firstName]_[lastName]

شرح آزمایش هفتم

هدف از انجام این آزمایش، طراحی و شبیه‌سازی یک واحد UART[1] است. UART یکی از روش‌های رایج ارتباط سریال غیرهمزمان است که در آن بیت‌های داده به‌جای انتقال موازی، به‌صورت پشت سر هم و روی یک خط ارسال می‌شوند. در این روش، برای آنکه گیرنده بتواند آغاز، محتوای اصلی و پایان هر قاب[2] را تشخیص دهد، افزون بر بیت‌های داده، از بیت‌های کنترلی همانند بیت شروع[3]، بیت توازن[4] و بیت توقف[5] استفاده می‌شود.

در قسمت فرستنده، هر بار یک داده ۷ بیتی، همانند یک کد ASCII، به‌صورت سریال ارسال می‌شود. قالب قاب مورد استفاده شامل یک بیت شروع، یک بیت توازن، هفت بیت داده و یک بیت توقف است. بنابراین، برای هر داده، در مجموع ۱۰ بیت روی خط سریال قرار می‌گیرد. در قسمت گیرنده نیز پس از تشخیص بیت شروع، بیت توازن و سپس هفت بیت داده دریافت می‌شوند. در پایان، گیرنده بیت توقف را بررسی کرده و با مقایسه بیت توازن دریافت‌شده با توازن محاسبه‌شده از داده، معتبر یا نامعتبر بودن قاب را مشخص می‌کند.

۱ قالب قاب ارسالی

در این آزمایش، خط سریال در حالت بی‌کار[6] مقدار ۱ دارد. با شروع ارسال، فرستنده ابتدا مقدار صفر را به‌عنوان بیت شروع روی خط قرار می‌دهد. سپس بیت توازن ارسال می‌شود و پس از آن، هفت بیت داده از بیت کم‌ارزش‌تر به بیت پرارزش‌تر روی خط قرار می‌گیرند. در پایان نیز بیت توقف با مقدار ۱ ارسال می‌شود تا پایان قاب مشخص شود.

ساختار قاب مورد استفاده در این آزمایش به‌صورت زیر است:

$$\text{Idle (1)} \rightarrow \text{Start (0)} \rightarrow \text{Parity} \rightarrow D_0 \rightarrow D_1 \rightarrow D_2 \rightarrow D_3 \rightarrow D_4 \rightarrow D_5 \rightarrow D_6 \rightarrow \text{Stop (1)}$$

بیت توازن از طریق عملگر XOR روی هفت بیت داده محاسبه می‌شود. بنابراین رابطه محاسبه بیت توازن به‌صورت زیر است:

$$Parity = D_0 \oplus D_1 \oplus D_2 \oplus D_3 \oplus D_4 \oplus D_5 \oplus D_6$$

برای نمونه، اگر داده ارسالی برابر مقدار زیر باشد:

$$send_data = 7'b1011001$$

انتظار می‌رود پس از پایان دریافت، مقدار داده دریافت‌شده نیز برابر همین مقدار باشد و سیگنال صحت داده مقدار ۱ بگیرد.

[1] Universal Asynchronous Receiver Transmitter
[2] Frame
[3] Start
[4] Parity
[5] Stop
[6] Idle

۲ طراحی فرستنده

فرستنده باید به‌صورت یک ماشین حالت متناهی طراحی شود. وظیفه فرستنده این است که داده ۷ بیتی موازی را دریافت کرده و آن را مطابق قالب قاب مشخص‌شده، به‌صورت سریال روی خط tx قرار دهد. در حالت عادی، فرستنده در وضعیت بی‌کار قرار دارد و مقدار خروجی tx برابر ۱ است. هنگامی که سیگنال new_data فعّال شود، فرستنده باید داده ورودی را نمونه‌برداری کرده و فرآیند ارسال قاب را آغاز کند.

حالت‌های اصلی فرستنده عبارت‌اند از:

۱. حالت Idle: در این حالت، خط tx مقدار ۱ دارد و فرستنده منتظر فعّال‌شدن سیگنال new_data است.

۲. حالت Start: در این حالت، مقدار صفر به مدت یک زمان بیت روی خط tx قرار می‌گیرد.

۳. حالت Parity: در این حالت، بیت توازن محاسبه‌شده از داده روی خط ارسال می‌شود.

۴. حالت Data: در این حالت، بیت‌های داده از D0 تا D6 به‌ترتیب روی خط ارسال می‌شوند.

۵. حالت Stop: در این حالت، مقدار ۱ به‌عنوان بیت توقف ارسال شده و سپس فرستنده به حالت بی‌کار بازمی‌گردد.

سیگنال busy باید در تمام مدت ارسال فعّال باشد و پس از پایان کامل قاب غیرفعّال شود. همچنین طراحی باید به‌گونه‌ای باشد که در هنگام مشغول بودن فرستنده، فعّال‌شدن مجدد new_data باعث تداخل در قاب جاری نشود.

۳ طراحی گیرنده

گیرنده نیز باید به‌صورت یک ماشین حالت متناهی طراحی شود. وظیفه گیرنده این است که خط ورودی rx را به‌صورت پیوسته بررسی کند و پس از مشاهده بیت شروع، بیت‌های بعدی قاب را به‌ترتیب دریافت نماید. تا زمانی که خط rx مقدار ۱ داشته باشد، گیرنده در حالت بی‌کار باقی می‌ماند. مشاهده مقدار صفر روی خط، به‌عنوان آغاز قاب جدید در نظر گرفته می‌شود.

فرآیند دریافت داده شامل مراحل زیر است:

- تشخیص بیت شروع با مشاهده مقدار صفر روی خط rx.
- دریافت بیت توازن و ذخیره آن در یک ثبات[۷] داخلی.
- دریافت هفت بیت داده و ذخیره آن‌ها در ثبات داده.
- دریافت بیت توقف و بررسی مقدار آن.
- محاسبه توازن داده‌های دریافت‌شده و مقایسه آن با بیت توازن دریافتی.
- فعّال‌کردن سیگنال correct_data در صورت صحیح بودن بیت توقف و بیت توازن.
- فعّال‌کردن سیگنال rec_new_data پس از کامل‌شدن فرآیند دریافت قاب.

[7] Register

اگر بیت توازن صحیح نباشد یا بیت توقف مقدار ۱ نداشته باشد، قاب دریافتی نامعتبر در نظر گرفته می‌شود. در چنین حالتی، گیرنده نباید داده را به‌عنوان داده معتبر گزارش کند.

۴ زمان‌بندی و نرخ ارسال

در حالت استاندارد، مقدار نرخ ارسال برابر 115200 bps در نظر گرفته می‌شود. زمان هر بیت از رابطه زیر به دست می‌آید:

$$T = \frac{1}{115200}\text{s}$$

اگر ساعت[8] سامانه مقدار ۵۰ مگاهرتز داشته باشد، تعداد سیکل‌های ساعت لازم برای نگه داشتن هر بیت روی خط به‌صورت تقریبی از رابطه زیر محاسبه می‌شود:

$$\frac{50,000,000}{115200} \simeq 434$$

با این حال، از آنجا که این آزمایش به‌صورت مجازی و با هدف مشاهده سریع‌تر موج‌ها انجام می‌شود، مجاز است برای زمان‌بندی بیت‌ها از یک مشخصه[9] همانند BIT_TICKS استفاده شود. این مشخصه تعیین می‌کند که هر بیت چند سیکل ساعت روی خط باقی بماند. برای مثال، اگر مقدار BIT_TICKS برابر ۲ باشد، هر بیت به مدت دو سیکل ساعت در شبیه‌سازی نگه داشته می‌شود.

۵ ساختار فایل‌های مورد انتظار

برای انجام این آزمایش، طراحی باید در قالب چند فایل مستقل و منظم ارائه شود. حداقل فایل‌های مورد انتظار عبارت‌اند از:

- `UARTSender.v`: شامل طراحی فرستنده و ماشین حالت ارسال.
- `UARTReceiver.v`: شامل طراحی گیرنده، دریافت داده، بررسی توازن و بررسی بیت توقف.
- `UARTTop.v`: شامل اتصال خروجی tx فرستنده به ورودی rx گیرنده.
- `Tester.v`: شامل میزآزمون[10] کامل برای تولید ساعت، اعمال بازنشانی، تولید داده آزمایشی و بررسی خروجی‌ها.

در پیمانه[11] سطح بالا، خروجی سریال فرستنده باید مستقیماً به ورودی سریال گیرنده متصل شود. این اتصال، یک ساختار بازگشت داخلی[12] ایجاد می‌کند. در این حالت، اگر طراحی درست باشد، داده دریافت‌شده باید با داده ارسال‌شده برابر باشد.

[8] Clock
[9] Parameter
[10] TestBench
[11] Module
[12] Loopback

۶ میزآزمون و روش شبیه‌سازی

برای ارزیابی عملکرد مدار، لازم است یک میزآزمون دقیق نوشته شود. میزآزمون باید ورودی‌های مدار را در زمان‌های مشخص مقداردهی کند و سپس خروجی‌ها را در طول شبیه‌سازی بررسی نماید. این میزآزمون باید دست‌کم موارد زیر را انجام دهد:

۱. تولید ساعت با دوره مشخص، برای مثال دوره ۱۰ نانوثانیه.

۲. اعمال بازنشانی اولیه به مدار.

۳. قرار دادن یک داده نمونه، مانند 7'b1011001، روی ورودی فرستنده.

۴. فعّال‌کردن کوتاه‌مدت سیگنال new_data برای آغاز ارسال.

۵. انتظار تا پایان فرآیند ارسال و دریافت قاب.

۶. بررسی مقدار rec_data و مقایسه آن با send_data.

۷. بررسی فعّال‌شدن سیگنال‌های rec_new_data و correct_data.

در میزآزمون باید از دستورهایی مانند $monitor یا $display برای چاپ مقدار سیگنال‌های مهم استفاده شود. همچنین لازم است تصویر موج شبیه‌سازی در گزارش نهایی قرار گیرد تا ترتیب زمانی تغییرات سیگنال‌ها قابل بررسی باشد.
سیگنال‌های پیشنهادی برای نمایش در پنجره Wave عبارت‌اند از:

$$clk,\ rstN,\ new_data,\ send_data,\ tx,\ busy,\ rec_data,\ rec_new_data,\ correct_data$$

۷ آزمون‌های الزامی

برای اطمینان از صحت عملکرد طراحی، دانشجو باید چند سناریوی شبیه‌سازی را اجرا و تحلیل کند. آزمون‌های زیر برای این آزمایش الزامی هستند:

۱.۷ آزمون دریافت صحیح

در این آزمون، یک داده ۷ بیتی مانند 7'b1011001 ارسال می‌شود. انتظار می‌رود پس از پایان دریافت، مقدار rec_data برابر داده ارسالی شود و سیگنال correct_data مقدار ۱ بگیرد.

۲.۷ آزمون خطای توازن

در این آزمون، باید بیت توازن به‌صورت عمدی در میزآزمون یا از طریق یک مسیر کمکی خراب شود. انتظار می‌رود گیرنده قاب را نامعتبر تشخیص دهد و سیگنال correct_data مقدار ۰ بگیرد.

۳.۷ آزمون خطای بیت توقف

در این آزمون، بیت توقف باید به‌صورت عمدی مقدار صفر داشته باشد. در این حالت، حتی اگر داده و بیت توازن صحیح باشند، قاب نباید معتبر در نظر گرفته شود.

۴.۷ آزمون ارسال چند داده پشت سر هم

در این آزمون، حداقل دو داده متفاوت پشت سر هم ارسال می‌شوند. برای مثال می‌توان از دو مقدار زیر استفاده کرد:

$$7'b1011001$$

$$7'b0101101$$

انتظار می‌رود گیرنده هر دو داده را به‌ترتیب صحیح دریافت کند و پس از پایان هر قاب، سیگنال دریافت داده جدید فعّال شود.

۸ گزارش نهایی

گزارش نهایی باید به‌صورت منظم و علمی و با رعایت نکات نگارش نوشته شود و دست‌کم شامل بخش‌های زیر باشد:

۱. توضیح کلی پروتکل UART و قالب قاب مورد استفاده.

۲. توضیح ماشین حالت فرستنده و نقش سیگنال‌های tx و busy.

۳. توضیح ماشین حالت گیرنده و روش بررسی بیت توازن و بیت توقف.

۴. توضیح پیمانه سطح بالا و نحوه اتصال فرستنده به گیرنده.

۵. توضیح میزآزمون و سناریوهای شبیه‌سازی.

۶. تصویر موج شبیه‌سازی در ModelSim.

۷. خروجی متنی Transcript یا خروجی‌های $monitor.

۸. تحلیل نتیجه شبیه‌سازی و مقایسه داده ارسال‌شده با داده دریافت‌شده.