



دانشکده مهندسی برق

ساختار کامپیوتر و میکروپروسسور و آزا

پروژه:

طراحی یک مازول ساده شده فرستنده – گیرنده سریال آسنکرون

تهیه کننده:

هلیا شاکری

شماره دانشجویی:

۴۰۰۱۰۳۸۹

ترم ۲-۱۴۰۱

توضیحات اولیه:

کد من با کمک سایت‌هایی که در قسمت منابع نوشته‌ام به دست آمده است ولی مهم‌ترین منبع [این کد](#) است.

اهمیت این کد به ماژول BaudRateGenerator است. در کدهای مختلفی که در اینترنت پیدا کردم، پیچیدگی کد در شمارنده‌ای بود که برای تنظیم bit rate نیاز بود ولی در این کد، با ماژول نام برده، کلاک جدیدی برای هر دو ماژول فرستنده و گیرنده ایجاد می‌شود که برای گیرنده oversampling در منبع ۱۶ برابر بود که به ۸ برابر تغییر دادم.

با استفاده از این ماژول کد فرستنده و گیرنده بسیار ساده‌تر و کمتر از بقیه روش‌ها نوشته شد.

توضیحات درباره ماژول‌های مختلف استفاده شده:

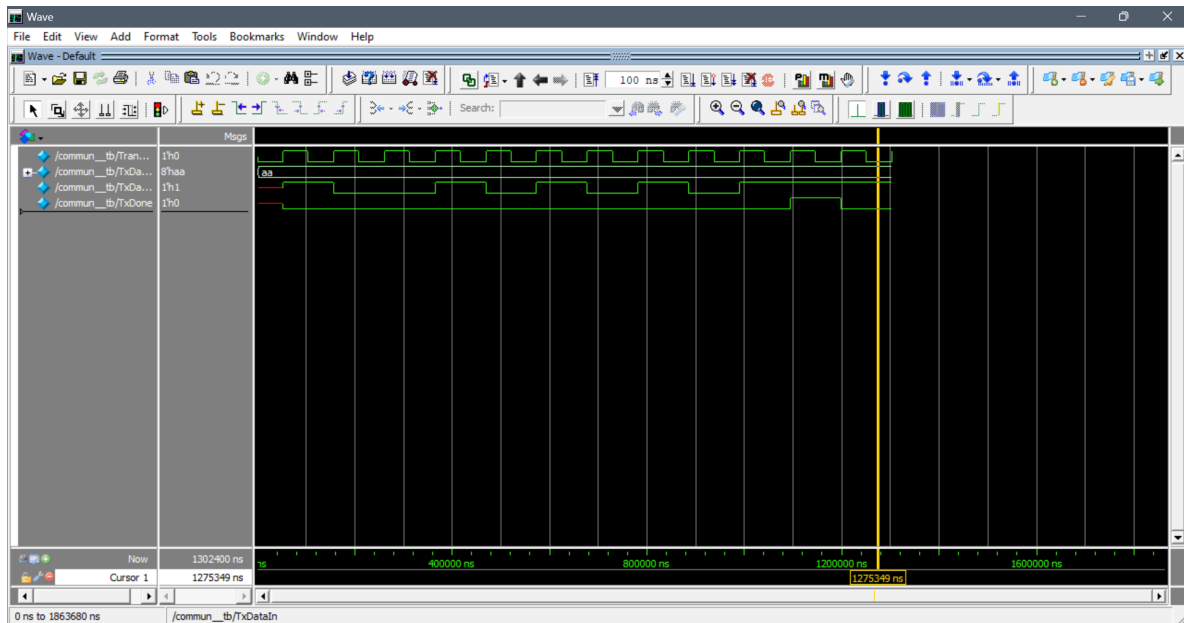
ماژول ClockGen :

در این ماژول برای فرستنده و گیرنده با Baud Rate قابل تنظیم کلاک جدیدی درست می‌شود که باعث می‌شود هر بیت ورودی در فرستنده در یک کلاک جدید فرستنده وارد شود و در گیرنده هر بیت برای ۸ کلاک جدید گیرنده ادامه دارد. برای تنظیم کلاک با baud rate ، ابتدا فرکانس جدید تعریف کرده شد که برابر است با فرکانس اصلی تقسیم بر دو برابر baud rate برای فرستنده و تقسیم بر ۱۶ برابر baudrate برای گیرنده. سپس اندازه رجیستر مورد نیاز برای شمارنده سایکل‌های کلاک اصلی مشخص شده و با استفاده از این اطلاعات کلاک جدیدی تولید می‌شود که فقط وقتی شمارنده به فرکانس جدیدی که به دست آمد رسیده باشد تغییر می‌کند. در این ماژول Baud Rate قابل تغییر است و با استفاده از آن برای دو baud rate مختلف ۱۹۲۰۰ و ۹۶۰۰ در ماژول اصلی دو کلاک تعریف می‌شود که با توجه به متغیر BaudRate یکی انتخاب می‌شود.

ماژول Transmitter :

در این ماژول ابتدا یک FSM تعریف می‌شود که ۴ حالت بیکار، شروع، فرستادن و پایان دارد. حالت بیکار حالت نرمال ماژول است که فقط وقتی ورودی TxDataLoad یک شود تغییر می‌کند و به حالت شروع می‌رود. در این حالت ابتدا بیت آغاز پیام که صفر است فرستاده می‌شود و سپس به حالت فرستادن می‌رود. در این حالت تنها وقتی شمارنده‌ای که برای شمارش بیت‌های پیام ورودی بود به ۷ رسید و بیت MSB ورودی فرستنده فرستاده شد، به حالت پایان می‌رود که در آن بیت پایانی که ۱ است فرستاده می‌شود.

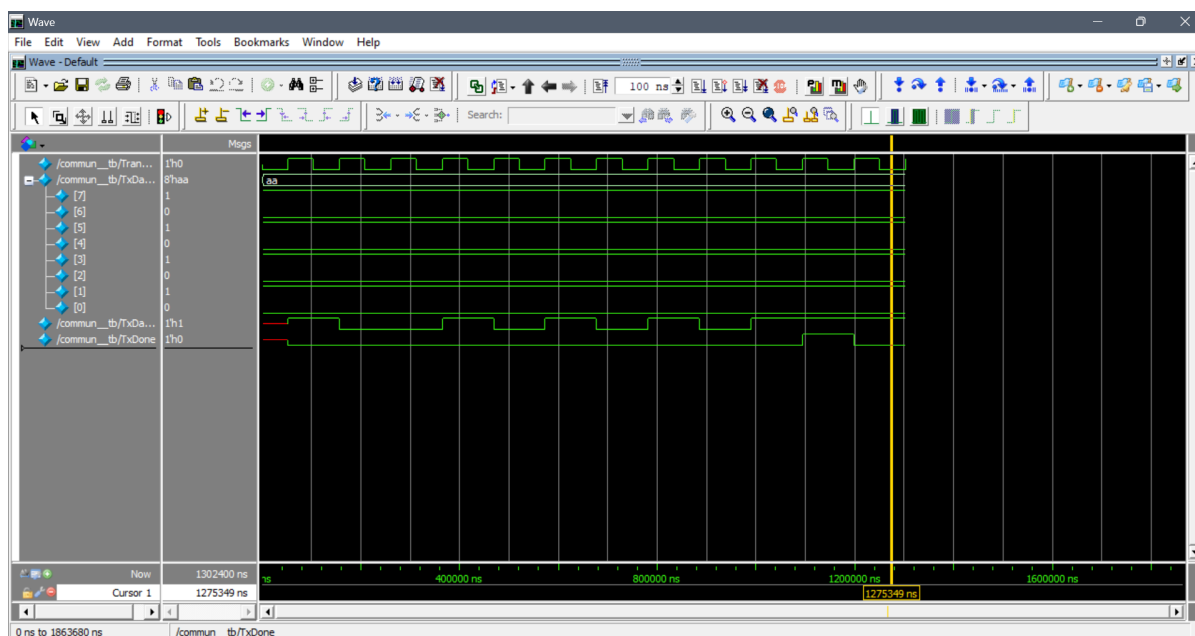
نمونه نتایج testbench برای ماژول Transmitter برای ورودی نمونه دستورکار ۱۰۱۰۱۰۱۰:



ماژول Receiver :

در این ماژول نیز ابتدا FSM تعریف می‌شود با چهار حالت بیکار، شروع، گرفتن و پایان. حالت بی‌کار حالت نرمال ماژول است و به محض اینکه بیت ورودی صفر شود از این حالت به حالت شروع می‌رود. در حالت شروع، باید بررسی شود که آیا بیت اولیه برای ۵ نمونه صفر هست یا نه و اگر نباشد باید RxError را یک کند و آن را نویز در نظر بگیرد و به حالت بی‌کار برگردد. این کار در حالت شروع با شمارش نمونه‌ها انجام می‌شود و سپس به حالت گرفتن منتهی می‌شود. در این حالت، چون oversampling وجود دارد، برای گرفتن هر بیت داده ورودی ۸ زمان وجود دارد. برای اطمینان از پایدار بودن بیت گرفته شده از تقریباً وسط، یعنی پنجمین سایکل کلاک ماژول گیرنده داده گرفته شده و به خروجی داده می‌شود. پس از گرفتن تمام ۸ بیت داده اصلی، وارد حالت پایان می‌شود که در این حالت ابتدا بررسی می‌کند که بیت آخر ۱ هست یا نه و اگر نباشد RxError را ۱ می‌کند و دوباره با حالت بی‌کاری برمی‌گردد. ولی اگر بیت آخر ۱ باشد خروجی را از صفر به ورودی سریالی تغییر می‌دهد و RxDone را ۱ می‌کند.

نمونه testbench ماژول Receiver برای ورودی رندوم:



ماژول Communication :

در این ماژول هر سه ماژول بالا باهم ترکیب می‌شوند و ماژول اصلی می‌باشد. در ابتدا باید تکلیف baud rate و کلاک‌های ماژول‌های فرستنده و گیرنده مشخص شود که همان‌طور که در قسمت توضیح ماژول ClockGen گفتم، دو کلاک برای هر دو ماژول تعریف می‌شود، یکی با baud rate برابر با ۱۹۲۰۰ و یکی با ۹۶۰۰. سپس با توجه به ورودی BaudRate یکی از این دو انتخاب شده و به ماژول‌های فرستنده و گیرنده وارد می‌شود.

در نمونه‌های test bench نشان‌داده شده در بالا، برای تست کردن ماژول فرستنده، ورودی ماژول گیرنده فقط ۱ و برای تست ماژول گیرنده ورودی TxDataLoad ماژول فرستنده صفر است که درست کار کردن ماژول Communication را نیز نشان می‌دهد.

در فایل تحویل داده شده، کدهای هر ماژول به صورت جداگانه است و در فایل ماژول همه ماژول‌های دیگر نیز قرار دارند برای راحتی اجرا. ۲ فایل تست بنچ برای دو ماژول فرستنده و گیرنده جدا نیز قرار داده ام.

1. <https://github.com/hell03end/verilog-uart>
2. <https://github.com/ben-marshall/uart>
3. <https://medium.com/@hmalgewatta/uart-with-verilog-part-one-receiver-2e770c1f35b1>
4. https://electronoobs.com/eng_circuitos_tut26.php
5. <https://nandland.com/uart-serial-port-module/>