ماژول گیرنده برای UART

در این قسمت شما باید یک ماژول دریافت‌کننده به شکل زیر بنویسید:

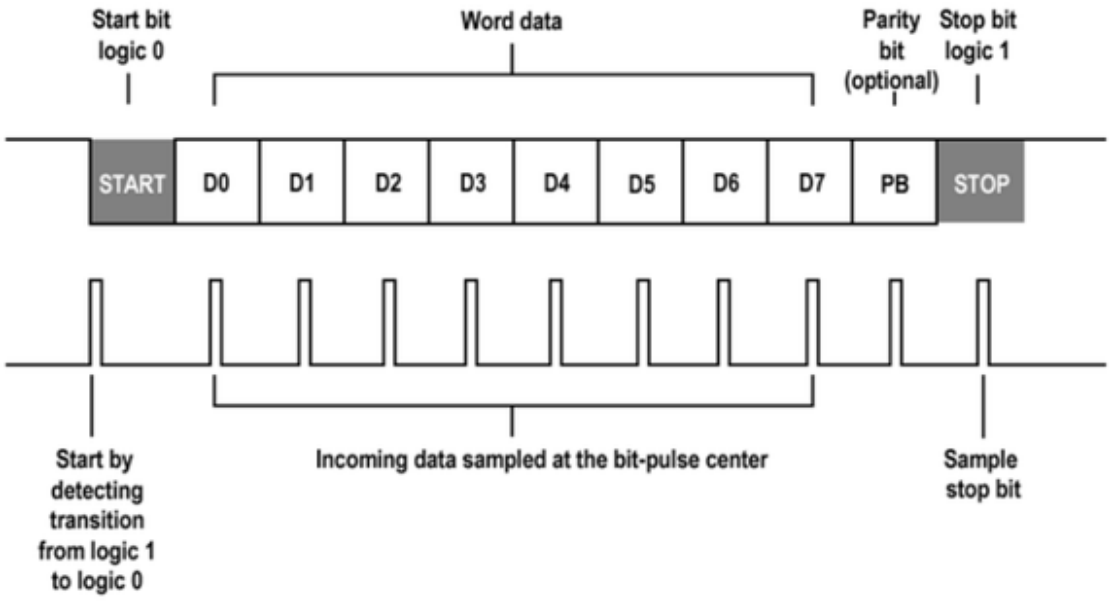
```
module Receive #(parameter clockperbit)(rxdata, rxfinish, rx, clock, reset

    input clock;
    input reset;
    input rx;
    output reg rxfinish;
    output reg [7:0] rxdata;
```

پارامتر clockperbit: این پارامتر مشخص می‌کند هر بیت داده به اندازه چند کلاک دستگاه روی ورودی rx می‌ماند. مقدار clockperbit به کلاک دستگاه و baudrate بستگی دارد و با تغییر آن‌ها این مقدار نیز تغییر می‌کند.

ورودیrx:ورودی یک بیتی است که وقتی در حالت idle هست اگر مقدار آن از ۱ به ۰ تغییر کند به منزله شروع ارسال داده از سمت فرستنده است.

هر ماژول برای دریافت داده در زمانهای خاصی را می‌خواندو پس از اتمام دریافت داده، داده ۸ بیتی را در خروجی قرار می‌دهداین زمان‌های خاص به نام sampling معروف هستند و وسط هر واحد زمانی قرار دارندمانند شکل زیر:



خروجی rxfinish: سیگنال خروجی ماژول است که مشخص می‌کند این ماژول داده را کامل دریافت کرده و مقداری که در خروجی قرار داده معتبر است

خروجی rxdata: داده ۸ بیتی که از پورت rx به صورت سریال دریافت شد روی این خروجی قرار می‌گیرد

ورودی reset وقتی مقدار این ورودی ۱ می‌شود دستگاه باید reset شودبه این معنا که اگر در حال خواندن داده است به حالت اولیه برگردد و منتظر شود مقدار rx از ۱ به صفر تغییر کند تا دوباره داده را از ابتدا بخواند.

طراحی با استفاده از بلوک initial و for در این تمرین غلط است و نمره‌ای نمی‌گیرد. همچنین استفاده از تاخیر و فراخوانی سیستمی time نیز طراحی صحیحی نیست و نمره‌ای نمی‌گیرد.

راهنمایی: در طراحی ماژول خود از یک counter استفاده کنید. به این شکل که تعداد poseedge های کلاک ورودی را بشمارید و هر گاه به وسط بازه زمانی مربوط به یک بیت رسیدید یبیت مربوط به آن را بخوانید.

ماژول فرستنده برای UART

ماژول فرستنده را به شکل زیر طراحی کنید:

```
module Transmit#(parameter clockperbit)
(txdata, txdone, send, tx, reset, clock);
input reset;
input clock;
input [7:0] txdata;
input send;
output reg tx;
output reg txdone;
```

پارامتر clockperbit: این پارامتر مشخص می‌کند هر بیت داده به اندازه چند کلاک دستگاه باید روی خروجی tx بماند. مقدار clockperbit به کلاک دستگاه و baudrate بستگی دارد و با تغییر آن‌ها این مقدار نیز تغییر می‌کند.

ورودی txdata: داده‌ای که قرار است فرستاده شود.

خروجی tx: روی این سیم در حالت idle مقدار ۱ قرار دارد و اگر بخواهیم داده‌ای را ارسال کنیم ابتدا آن را برای یک واحد زمانی ۰ می‌کنیم سپس داده را یک بیت یک بیت ارسال می‌کنیم. و در انتها به اندازه یک واحد زمانی tx را ۱ قرار می‌دهیم.

خروجی txdone: این سیگنال خروجی مشخص می‌کند که دستگاه ارسال داده را تمام کرده و آماده است که داده بعدی را ارسال کند. مقدار txdone تا وقتی دستگاه شروع به ارسال داده بعدی نکرده ۱ می‌ماند.

ورودی send: وقتی این سیگنال ورودی ۱ می‌شود دستگاه باید شروع به ارسال داده کند.

ورودی reset: وقتی مقدار این ورودی ۱ می‌شود دستگاه باید reset شود به این معنا که به حالت اولیه برود و منتظر بماند تا سیگنال send یک شود تا داده بعدی را ارسال کند.

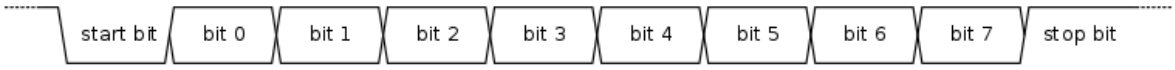
طراحی با استفاده از بلوک initial و for در این تمرین غلط است و نمره‌ای نمی‌گیرد. همچنین استفاده از تاخیر و فراخوانی سیستمی time نیز طراحی صحیحی نیست و نمره‌ای نمی‌گیرد.

راهنمایی: در طراحی ماژول خود از یک counter استفاده کنید. به این شکل که تعداد poseedge های کلاک ورودی را بشمارید و هر گاه به اندازه‌ی پارامتر ورودی clockperbit شد داده بعدی را ارسال کند.

UART کامل

یوآرت چیست؟

یوآرت(universal asynchronous receiver-transmitter) یک سخت‌افزار برای ارتباط ناهمگن **سریال** است که در آن سرعت و فرمت ارسال داده قابل تنظیم است. هر دستگاه UART یک پورت ورودی rx برای دریافت داده و یک پورت خروجی tx برای ارسال داده دارد. هر دستگاه UART یک بایت داده را دریافت می‌کند و تک تک بیت‌های آن را به صورت سریال با پروتکل مشخصی روی پورت tx ارسال می‌کنددستگاه دیگری در مقصد می‌تواند از پورت rx این بیت‌ها را دریافت کند و آن را دوباره به یک بایت داده تبدیل کند. فرم کلی دریافت و ارسال داده در ماژول UART به شکل زیر است:



در حالت idle (زمانی که هیچ داده‌ای در حال فرستادن نیست) خروجی ۱ است. زمانی که می‌خواهیم شروع به ارسال داده کنیم، به اندازه یک واحد زمانی خروجی را صفر میکنیم (start bit) و سپس بیت‌های داده را (۵ تا ۹ بیت که تعداد آن از قبل مشخص و قرارداد شده است) یکی یکی ارسال میکنیم. هر بیت به اندازه یک واحد زمانی روی سیم می‌ماند و سپس بیت بعدی ارسال می شود. بعد از اتمام بیت‌های داده، خروجی را برای ۱، ۱/۵ یا ۲ واحد زمانی یک نگه‌می‌داریم(stop bit). طول stop bit نیز قرارداد شده و از قبل مشخص است.

مفهوم baud rate و واحد زمانی:

نرخ ارسال داده یا baud rate به تعداد بیتی می‌گویند که در یک ثانیه منتقل می‌شود. واحد زمانی یا clock per bit به تعداد کلاکی می‌گویند که یک بیت روی سیم انتقال می‌ماند. برای مثال اگر کلاک یک سیستم 9.6MHz باشد و baud rate برابر ۹۶۰۰ باشد، هر بیت باید به اندازه ۱۰۰۰ کلاک روی سیم بماندبنابراین هر واحد زمانی در این سیستم با این نرخ ارسال داده ۱۰۰۰ کلاک است.

پیاده سازی

طول داده در این پروتکل می‌تواند بین ۵ تا ۹ بیت باشد. اما باید از پیش مشخص باشد و بین دو دستگاه قرارداد شود. همچنین می‌تواند شامل بیت parity باشد یا نباشد. در این تمرین ما از شما می‌خواهیم که یک UART

طراحی کنید که در هر ارسال یا دریافت، ۸ بیت داده بدون parity انتقال دهد. همچنین باید قابلیت ارسال و دریافت همزمان داده را داشته باشد. تعداد stop bit ها نیز می‌تواند متفاوت باشد ولی مانند طول داده باید از قبل مشخص و قرارداد شده باشد. در این تمرین تعداد stop bit را ۱ در نظر می‌گیریم. برای پیاده‌سازی این ماژول ابتدا باید دو ماژول فرستنده و گیرنده(سؤال ۲ و ۳ این تمرین) را پیاده‌سازی کنید. سپس با کمک این دو ماژول، ماژول UART را پیاده‌سازی کنید.(بنابراین، پس از مطالعه مطالب بالا، ابتدا سؤال ۲ و ۳ را حل کنید و سپس سراغ این سؤال بیایید.)

طراحی با استفاده از بلوک initial و for در این تمرین غلط است و نمره‌ای نمی‌گیرد. همچنین استفاده از تاخیر و فراخوانی سیستمی time نیز طراحی صحیحی نیست و نمره‌ای نمی‌گیرد.

ماژول UART

ماژول UART را به شکل زیر طراحی کنید:

```
module UART #(parameter clkperbit)
(rx, send, clock, reset, tx, txdata, rxdata, rxfinish, txdone);
input rx , clock, reset, send;
output tx, rxfinish, txdone;
output[7:0] rxdata;
input[7:0] txdata;
```

پارامتر clockperbit: این پارامتر مشخص می‌کند هر بیت داده چند کلاک باید روی سیم بماند.

ورودی txdata: داده‌ای که می‌خواهیم ارسال کنیم.

خروجی rxdata: داده‌ی ۸ بیتی که از طریق پورت rx دریافت شده.

ورودی reset: با یک شدن این سیگنال دستگاه باید به حالت اولیه برگردد و از ابتدا ارسال یا دریافت داده را انجام دهد.

ورودی rx: پورت ورودی سریال

خروجی tx: پورت خروجی سریال که از طریق آن داده‌ها را می‌فرستیم.

ورودی send: با یک شدن این سیگنال باید داده‌ای که در txdata قرار دارد از طریق پورت سریال tx فرستاده شود.

خروجی rxfinish: مشخص می‌کند که دریافت داده از پورت rx به انتها رسیده.

خروجی txdone: در صورتی که ۰ باشد به این معنی است که دستگاه در حال فرستادن داده است. و اگر ۱ باشد به این معنی است که ارسال داده پایان یافته و دستگاه آماده است که داده بعدی را ارسال کند.

سلول