

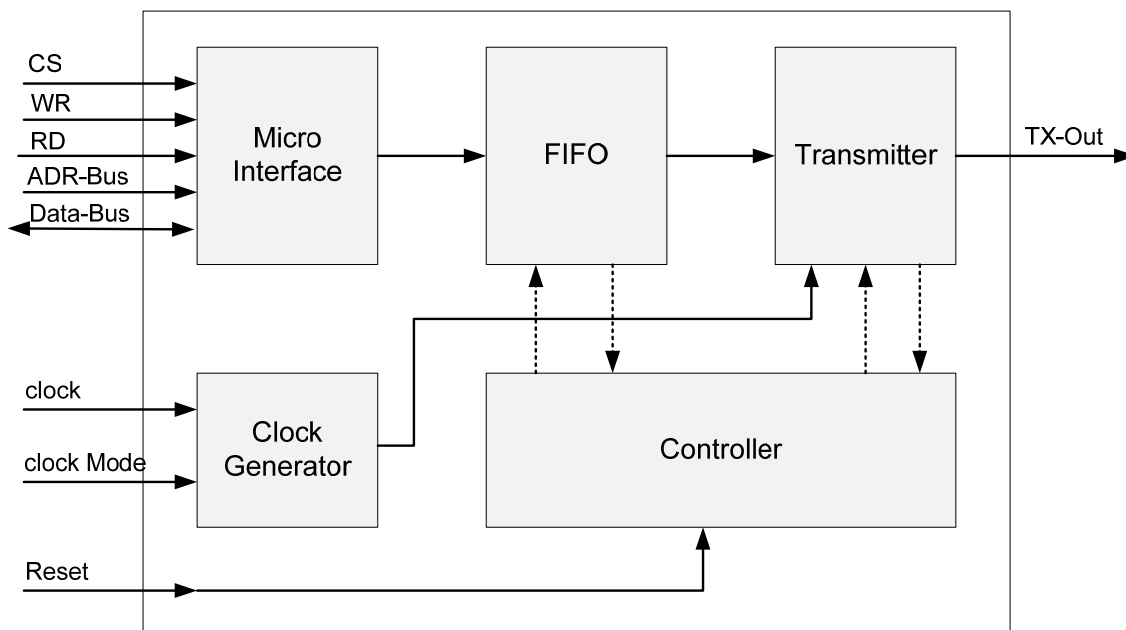


## ۱. مقدمه

هدف از انجام این پروژه، آشنایی کامل با چگونگی طراحی و پیاده سازی یک سیستم دیجیتال با استفاده از زبان توصیف سخت افزاری VHDL می باشد. سیستم پایه ای که برای این پروژه در نظر گرفته شده است، یک واسطه سریال (UART Interface) می باشد. در این واسطه داده های هشت بیتی از طریق Data Bus خوانده شده و سپس این داده ها به صورت سریال بر روی خروجی ارسال می گردد.

## ۲. آشنایی با واسطه سریال

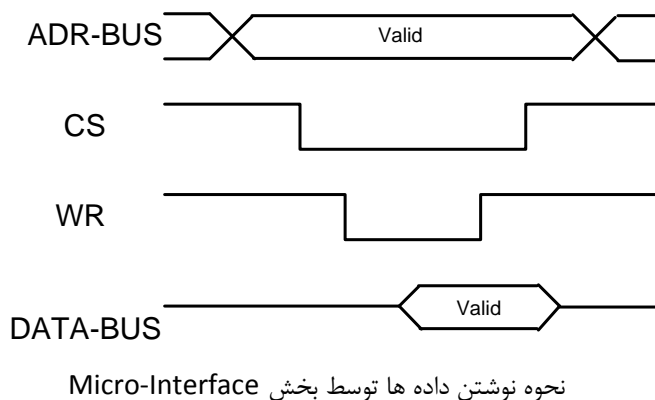
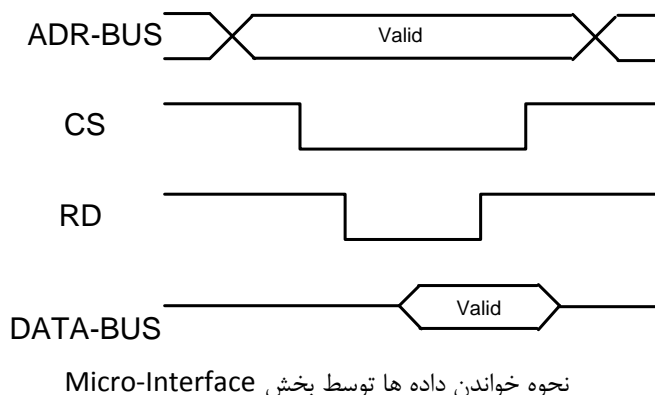
شکل زیر بلوک دیاگرام بخش فرستنده یک واسطه سریال می باشد.



همانطور که در شکل نیز آورده شده است، این ماجول از پنج بخش مختلف تشکیل شده است:

۱- بخش Micro-Interface : در این بخش ارتباط واسط با یک پردازنده پیاده سازی می شود. همانطور که می دانید واسط سریال یک آی سی جانبی می باشد که برای استفاده از آن نیاز به یک پردازنده می باشد که از طریق آن دستورات مورد نیاز را دریافت نموده و آنها را اجرا نماید. از طریق این بخش داده هایی که باید توسط واسط سریال ارسال گردند به آن داده می شود. همچنین وضعیت واسط از طریق این بخش خوانده شده و در اختیار پردازنده قرار می گیرد. به عنوان مثال وضعیت پر/خالی بودن صف ارسال داده ها از این طریق خوانده می شود. به طور کلی این بخش دارای دو وظیفه مهم است: الف- انتقال وضعیت پر/خالی بودن صف انتقال اطلاعات ب- نوشتن داده های مورد نیاز به درون واسط جهت ارسال از طریق ارتباط سریال

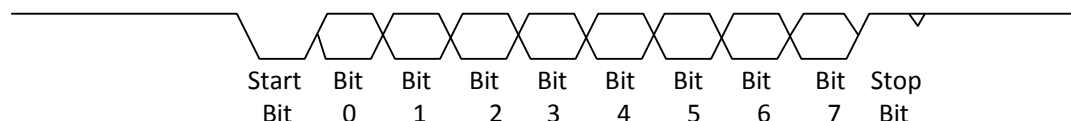
از آنجا که این ارتباط یک ارتباط Asynchronous می باشد، لذا برای مبادله اطلاعات به سیگنال های CS، RD، WR، Address-Bus و Data-Bus نیاز می باشد. Timing کار با این سیگنال ها برای خواندن و یا نوشتن داده ها به صورت زیر می باشد:



آدرس های اختصاص یافته به بخش های مختلف واسط سریال به صورت زیر می باشد. از آدرس "110" وضعیت پر/خالی بودن صف داده ها خوانده می شود. اگر مقدار خوانده شده از این آدرس صفر بود یعنی صف خالی می باشد و اگر مقدار خوانده شده یک بود یعنی اینکه صف پر می باشد. از آدرس "101" نیز برای نوشتن داده ها درون صف به منظور ارسال به صورت سریال استفاده می گردد.

۲- بخش FIFO یک صف سخت افزاری به طول ۱۶ بایت می باشد. صف سخت افزاری چیزی جز یک حافظه RAM نیست که برای کار با آن از قاعده First-In-First-Out استفاده می شود. این بخش داده های گرفته شده از پردازنده (که از طریق بخش Micro-Interface صورت گرفته است) را دریافت نموده و در صف قرار داده و سپس داده های درون صف را یکی-یکی خوانده و از طریق بخش Transmitter به صورت سریالی به بیرون ارسال می نماید.

۳- بخش Transmitter داده های خوانده شده از صف بخش FIFO را به صورت سریال به بیرون ارسال می نماید. این بخش دائماً FIFO را مانیتور نموده و در صورتیکه FIFO خالی نباشد، یک داده از آن خوانده و به بیرون ارسال می نماید. با اتمام ارسال داده، مجدداً Transmitter وضعیت FIFO را مانیتور نموده و اگر FIFO خالی نباشد، یک داده دیگر خوانده و آنرا ارسال می نماید. طبیعتاً اگر FIFO خالی باشد، بخش Transmitter کاری انجام نخواهد داد و داده ای را روی خروجی اش قرار نمی دهد. فرمت استاندارد ارسال یک بایت بر روی سریال به صورت زیر می باشد:



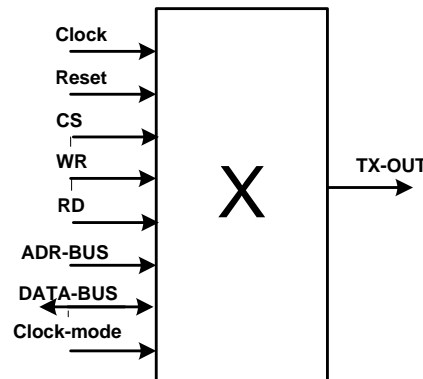
همانطور که در شکل مشخص است، خروجی TX-OUT همواره در حالت Idle به صورت یک می باشد و زمانی که داده ای برای ارسال دارد در ابتدا یک بیت داده صفر تحت عنوان Start-Bit و سپس هشت بیت داده را به ترتیب و بدون فاصله ارسال می نماید و در انتهای آن یک بیت داده یک به عنوان Stop-Bit ارسال می نماید. این سناریو برای ارسال هر داده هشت بیتی تکرار می گردد. طول هر بیت اطلاعات ارسال شده بر روی TX-OUT به فرکانس کاری ما بستگی دارد. برای این پروژه دو فرکانس کاری 115200 و 57600 هرتر کافی می باشد. پس بسته به فرکانس مورد استفاده، طول بیت های TX-OUT برابر پریود یکی از دو کلاک فوق می باشد.

۴- بخش Clock-Generator وظیفه ایجاد کلاک مورد نیاز سیستم را دارد. این بخش، ورودی Clock با فرکانس 115200 Hz را دریافت نموده و اگر ورودی clock mode برابر یک باشد فرکانس ورودی clock تقسیم بر دو شده و به عنوان کلاک سیستم، برای ارسال اطلاعات استفاده می شود و در غیر این صورت، از همین کلاک به عنوان کلاک سیستم استفاده می گردد.

۵- بخش controller وظیفه کنترل بخش های مختلف واسط سریال را بر عهده دارد. این بخش مشخصاً از یک یا چند State-Machine تشکیل شده است. طراحی این State-Machine ها با توجه به باقی بخش ها می تواند از یک طراحی به طراحی دیگر متفاوت باشد.

### ۳. تعریف پروژه

پروژه مورد نظر این درس دارای ورودی/خروجی های زیر می باشد:



کد Entity ماحول توصیف شده فوق، در زیر نشان داده شده است.

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

--

ENTITY X IS
    PORT
    (
        clock           : IN STD_LOGIC;
        reset           : IN STD_LOGIC;
        clock_mode      : IN STD_LOGIC;
        cs              : IN STD_LOGIC;
        wr              : IN STD_LOGIC;
        rd              : IN STD_LOGIC;
        address_bus     : IN STD_LOGIC_VECTOR(2 DOWNTO 0);
        data_bus        : INOUT STD_LOGIC_VECTOR(7 DOWNTO 0);
        transmit_out    : OUT STD_LOGIC
    );
END X;
```

در این پروژه، داده های هشت بیتی توسط بخش Micro-Interface از برنامه Test-Bench خوانده می شوند. ورودی clock، یک سیگنال کلاک 115200 Hz می باشد که از طریق Test-Bench ایجاد می گردد.

همچنین در Test-Bench باید clock-Mode نیز تعیین گردد که می تواند صفر یا یک باشد.

داده های هشت بیتی، پس از خوانده شدن از طریق Micro-Interface به FIFO انتقال داده شده و بخش Transmitter داده های را یکی یکی از FIFO خوانده و با توجه به فرکانس کاری تعیین شده توسط بخش clock-Generator، به صورت سریال (با ملاحظات Start-Bit و Stop-Bit) به خروجی می فرستد. توجه نمایید که وجود FIFO در این واسطه کاملاً ضروری می باشد زیرا نرخ ایجاد داده ها با نرخ مصرف آنها کاملاً

متفاوت می باشد و ممکن است در زمان هایی مثلاً شش داده برای ارسال در FIFO وجود داشته باشد یا در مواقعی هیچ داده ای در FIFO نباشد. زمانیکه FIFO خالی می باشد، بخش Transmitter نباید کار کند و خط TX\_OUT به صورت یک (حالت Idle) باقی می ماند.

#### ۴. نحوه تحویل پروژه

در هنگام تحویل پروژه لازم است گزارشی از نحوه طراحی و پیاده سازی ماحول مورد نظر، تاپ شده تحویل داده شود. در این گزارش تمامی Data-path و Controller های ماحول ترسیم شده و لازم است به طور دقیق شرح داده شوند. انجام پروژه به صورت انفرادی خواهد بود و هر نفر ملزم به ارائه تمامی خواسته های پروژه می باشد.

موفق باشید.