

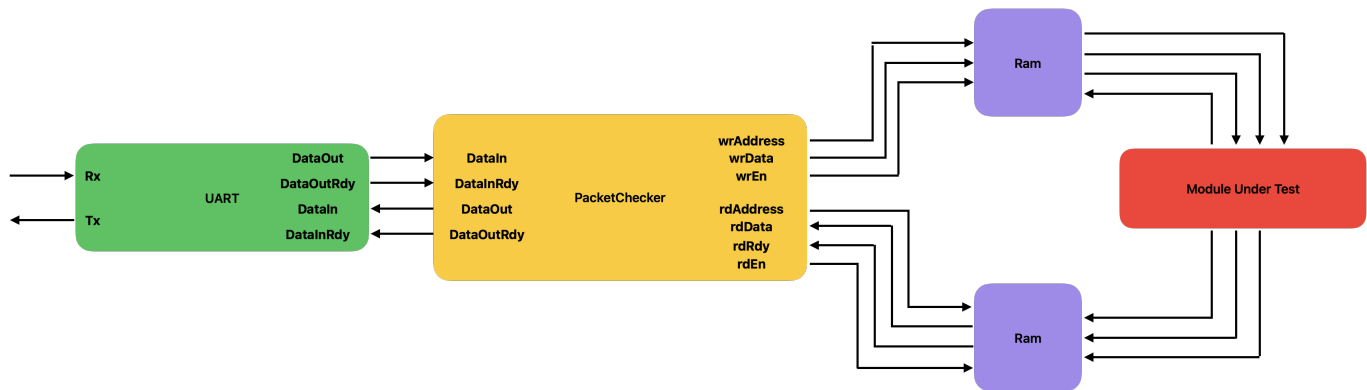


تمرین VHDL سری دوم درس مدارهای منطقی برنامه‌پذیر (FPGA)

تدریس‌یار: سید نیما علوی

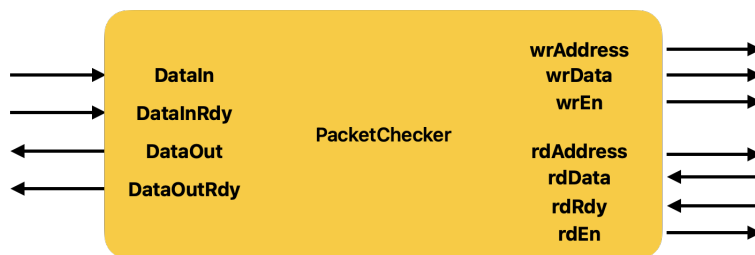
استاد درس: دکتر محمدرضا پورفرد

فرم کلی تاپ‌ماژولی که قرار است در این ترم داشته باشیم بدین صورت می‌باشد.



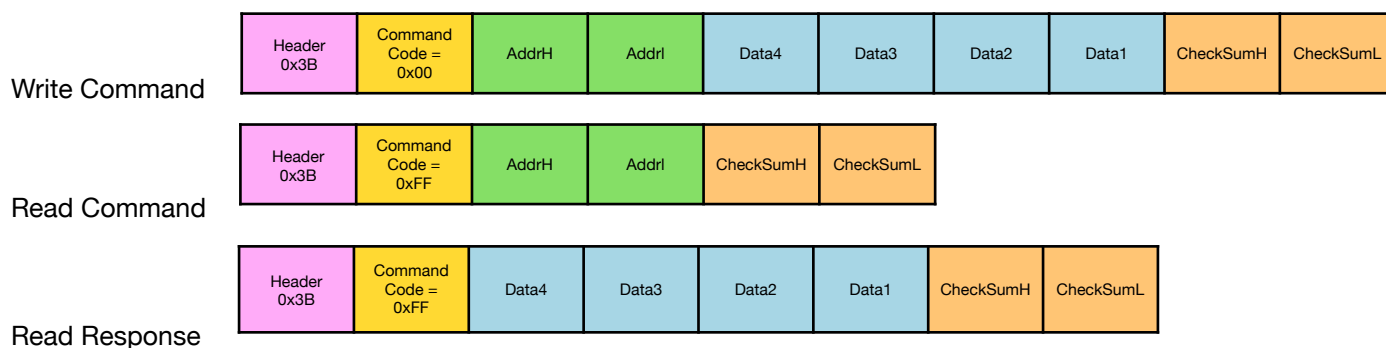
تا به اینجا ما UART را ساخته ایم. و در این تمرین می‌خواهیم ماژول PacketChecker را پیاده‌سازی کنیم.

فرم کلی ماژول ما قرار است بدین صورت باشد:



در این ماژول ما قصد داریم تا پکت‌هایی که از طرف ماژول UART دریافت می‌کنیم را بررسی کرده و در صورت یافتن پکت‌های خاص آنها را به آدرس درست و داده‌ی درست تقسیم بندی کرده و در نهایت به خروجی دهیم. همانطور که مشاهده می‌کنید ماژول بعدی این ماژول RAM می‌باشد، پس هر گاه داده و آدرسی آماده داشتیم با فعال کردن wrEn فرمان نوشتن را به رم صادر می‌کنیم. همین روند به صورت برعکس نیز صادق است، یعنی ما هر گاه توانستیم داده‌ای را بخوانیم rdEn را فعال می‌کنیم و ماژول RAM هرگاه داده را در اختیارمان قرار داد rdRdy را فعال کرده و متوجه می‌شویم که rdAddress و rdData دارای مقادیر معتبری هستند. در نتیجه باید این مقادیر معتبر را به صورت فرمت زیر به خروجی بفرستیم.

به صورت کلی پکت‌های نوشتن و خواندن به صورت زیر می‌باشد.



همانطور که می‌دانیم در ماژول UART هر داده خروجی ۸ بیتی می‌باشد، پس برای آنکه فرم بالا را داشته باشیم باید هر یک از خانه‌های جدول پکت را که ۸ بیتی اند به کمک Rx در ماژول UART بفرستیم. (در اینجا فرض می‌کنیم ماژول UART درست کار می‌کند و داده‌های درستی را در خروجی قرار می‌دهد).

سپس ماژول PacketChecker ما بررسی می‌کند اگر در ابتدا Header را دریافت کرد متوجه می‌شود که فرمت داده جدیدی در راه است و منتظر داده بعدی می‌باشد. حال Command Code را چک می‌کند اگر 0x00 بود می‌فهمیم در مد نوشتن هستیم و اگر 0xFF بود در مد خواندن. و به ترتیب آدرس ۱۶ بیتی و داده ۳۲ بیتی را به ترتیب می‌فرستد.

هنگامی که ما کامند خواندن را اجرا می‌کنیم، ماژول باید بعد از آدرسی که دریافت می‌کند از RAM خوانده و به صورت ReadResponse به UART بدهد تا به کمک پورت Tx به خروجی دهد.

لازم به ذکر است که هر ۸ بیتی که به داخل می‌آیند برای آنکه چک شود خطا داریم یا خیر، ۸ بیت‌ها را باهم جمع کرده و در نهایت با مقدار CheckSum مقایسه می‌کنیم و اگر برابر نبودند خطا را اعلام می‌کنیم.

```
entity PacketChecker is
  generic (
    header      : std_logic_vector(8 - 1 downto 0) := x"00"
  );
  Port (
    clk         : in std_logic;
    rst         : in std_logic;

    dataIn      : in std_logic_vector(8 - 1 downto 0);
    dataInRdy   : in std_logic;
    dataOut     : out std_logic_vector(8 - 1 downto 0);
    dataOutRdy  : out std_logic;

    wrAddress   : out std_logic_vector(16 - 1 downto 0);
    wrData      : out std_logic_vector(32 - 1 downto 0);
    wrEn        : out std_logic;
    rdAddress   : out std_logic_vector(16 - 1 downto 0);
    rdData      : in std_logic_vector(32 - 1 downto 0);
    rdRdy       : in std_logic;
    rdEn        : out std_logic;

    error       : out std_logic
  );
end entity;
```

نکات مهم:

- شما تنها باید مائزول PacketChecker را تحویل دهید.
- کد شما کاملاً باید به صورت سنکرون پیاده‌سازی شود.
- شبیه‌سازی ای که انجام می‌دهید باید به کمک نرم‌افزار مدلسیم باشد پس پیشنهاد می‌شود ویدیوی مربوطه را تماشا کنید.
- نحوه فایل ارسالی شما نیز باید به صورتی باشد که در ویدیو شماره ۲ مدلسیم بیان کردیم. (پوشه‌ها، نام‌ها و ...)
- خوانا نویسی و تمیزی کد شما نیز مهم می‌باشد.
- فایل گزارش نویسی شما باید جامع و کامل باشد؛ به نحوی که هم توضیح هر بخش از کد با جزئیات آورده شود و هم تصاویری از نتایج شبیه‌سازی به همراه تحلیل دقیق آن‌ها در آن قرار بگیرد. (فایل pdf و word هر دو الزامیست).
- همینطور لازم است ویدیویی هم ضبط کنید که در آن به توضیح شفاهی موارد بالا از روی کد و فضای سیمولیشن پرداخته‌اید. این ویدیو را هم در کورسز آپلود و هم در تلگرام برای استاد و بنده ارسال کنید.