**ازمایش چهارم**

**موضوع: انتقال سریال یک بایت از دیپ سوییچ به tx**

تاریخ آزمایش: 9/8/1402

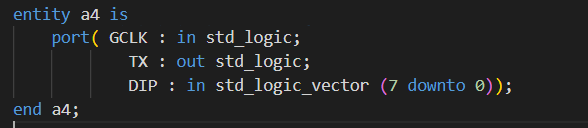
استاد: مهندس جوادی

جواد فرجی (99522005)

محمد رحمانی (97521288)

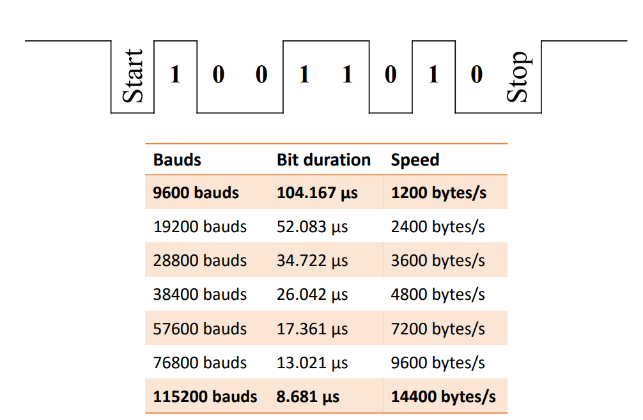
**ورودی و خروجی:**

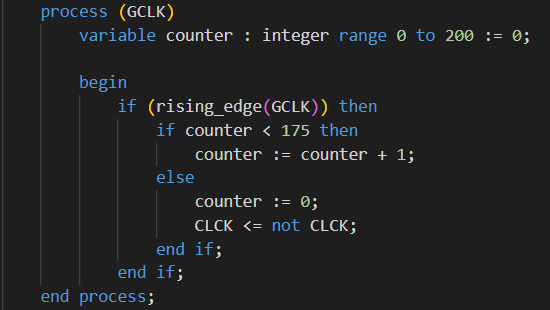
کلاک GCLK به عنوان کلاک اصلی. خروجی TX که قرار است به صورت سریال باشد و 8 بیت DIP که ورودی دیپ سوییچ است.



پروسس ها:

* ساخت کلاک خروجی برای تشخیص توالی بیت ها:

**برای این که بتوانیم به ترتیب بیت ها را بخوانیم و تشخیص دهیم نیاز داریم که یک سری استاندارد ها را رعایت کنیم. یکی از این استاندارد ها اندازه کلاک است که در جلسه توضیح داده شده و باید کلاک خروجی جدیدی بسازیم. برای این کار نیاز به یک پروسس داریم که کلاک جدیدی بر اساس کلاک کنونی بسازد. در اینجا هر 175 بار که کلاک GCLK از 0 به 1 تریگر میشود، یک بار مقدار کلاک CLCK تغییر میکند. که یک سیگنال در بدنه اصلی برنامه است.**



* **انتقال سریال بیت ها در یک پروسس**

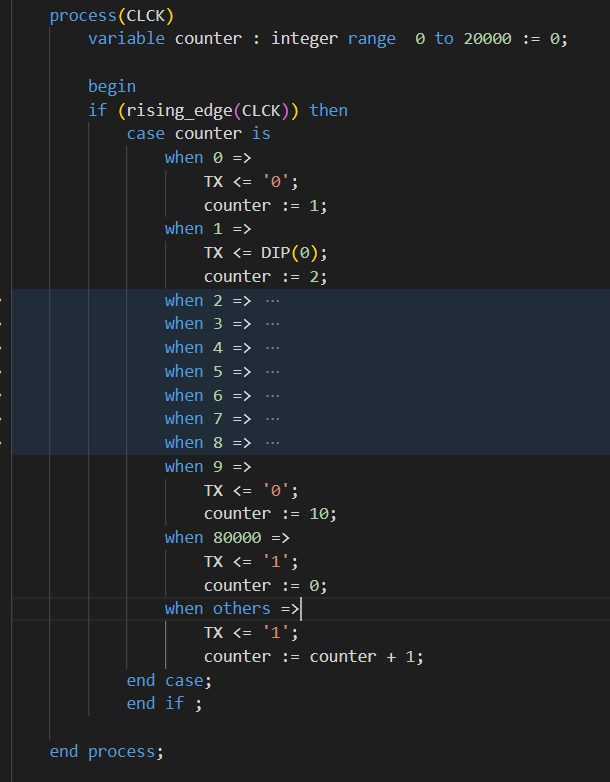
**برای این کار باید مطابق استاندارد های انتقال سریال، یک بار بیت خروجی را 0 کنیم که نشان دهنده استارت است. بعد از آن باید 8 بیت به خروجی بدهیم. خروجی 9 ام میتواند استاپ باشد که در این صورت باید 0 باشد و یا میتواند بیت parity باشد که در اینجا استفاده نکردیم.**

**نکته: در حالت عادی بیت خروجی باید 1 باشد. به غیر از زمان هایی که نیاز به انتقال سریال داشته باشیم. استارت و استاپ نشان دهنده اغاز و پایان انتقال اطلاعات هستند.**

**در این پروسس هم از این روش استفاده شده. با استفاده از سوییچ کیس بر روی متغیر counter چند حالت را بررسی میکنیم.**

**در حالت 0 و 9 استارت و استاپ داریم. در دو حالت اخر نیز شمارنده تا 80000 می شمارد که هر یک ثانیه یک بار اطلاعات منتقل شوند.**

**در 8 حالت میانی، در هر کلاک یکی از مقادیر بیت های دیپ سوییچ را به ترتیب به خروجی داده ایم.**

****

* نمایش اطلاعات خروجی:

**در اینجا نیاز داریم اطلاعات سریالی که در خروجی tx آمده را نمایش دهیم. برای این کار از یک مبدل استفاده شده که خروجی را به usb منتقل کنیم تا بتوانیم با استفاده از نرم افزار docklight این خروجی سریال را مشاهده کنیم.**

**مپ کردن خروجی ها به روی fpga:**

برای مپ کردن روی برد های fpga، با استفاده از داکیومنت موجود، این خطوط را داخل فایل ucf قرار میدهیم:

NET "GCLK" CLOCK\_DEDICATED\_ROUTE = FALSE;

NET "GCLK" LOC = P184;

NET "TX" LOC = P40;

NET "DIP[0]" LOC = P171;

NET "DIP[1]" LOC = P169;

NET "DIP[2]" LOC = P168;

NET "DIP[3]" LOC = P167;

NET "DIP[4]" LOC = P166;

NET "DIP[5]" LOC = P165;

NET "DIP[6]" LOC = P162;

NET "DIP[7]" LOC = P161;

**اجرای برنامه بر روی برد:fpga**

1. Synthesize
2. Implement design
3. Generate programming

در این سه مرحله گزینه run را میزنیم و در صورتی که مشکل خاصی در برنامه وجود نداشته باشد و به باگ نخوریم به مرحله بعد می‌رویم.

1. Impact

با استفاده از این برنامه، فایل باینری ساخته شده را به programmer انتقال میدهیم و programmer این برنامه را روی بردهای fpga اجرا میکند.