

به نام خدا

موضوع : پیاده سازی بخش SPI (Master) آی سی ADF4350

نام و نام خانوادگی : جواد علی شعار      شماره دانشجویی : 401611193

مقدمه : مراحل کار به شرح زیر است.

1- مطالعه مشخصات SPI و اینکه در این آی سی بایستی مرحله نوشتن با طول 32 بیت با فرکانسی انجام شود که از جدول زمانی آن تعیین خواهد شد . مرحله خواندن وجود ندارد. در واقع یک SPI سه سیمه می باشد. برنامه ریزی آی سی با نوشتن در در شش رجیستر به طول 32 بیت کامل می شود.

2- جهت اجرای این قسمت ، از منابع آموزشی آقای ثقیی استفاده شده است . در پیاده سازی ابتدا پروتکل SPI به صورت کامل معرفی می شود. ثانیاً تایمینگ ای پروتکل SPI شرح داده می شود. که اساس کار و نقطه عطف کار می باشد زیرا کلیه پارامترهای زمانی و تعیین فرکانس انتقال داده تعیین می شود. در این قسمت تعیین زمان های با حاشیه اطمینان جهت اجرای طرح بسیار حایز اهمیت است. بدون درک این بخش ادامه کار بیهوده است. ثالثاً اجرای کد از پروسس به همراه دستورات شرطی استفاده شده است. نکته اساسی در این پیاده سازی استفاده از شمارنده (Counter) جهت ایجاد تاخیر های زمانی است که به نظر این نکته کلید اجرای پروتکل های زمانی مانند SPI می باشد.

3- در کد تغییراتی ایجاد شده است زیرا شکل خروجی آن با شکل تایمینگ SPI یکسان نبود. در شکل موج زمان انتقال داده در کاتالوگ با شکل موج برنامه متفاوت بود که تصحیح شد.

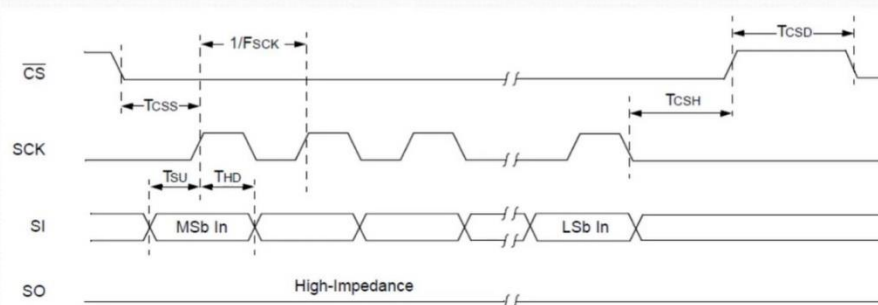
4- برنامه در Vivado نوشته و شبیه سازی شده است. این برنامه ویژگی جالب Auto complete را دارد و در لحظه برنامه را از لحاظ غلط املایی بررسی می کند.

5- ابتدا جدول تایمینگ را در صفحه بعد خواهید دید و با توجه به آن جدول تایمینگ آی سی ADF4350 را با آن هماهنگ نموده و پارامترهای برنامه را استخراج می کنیم.

## تایمینگ پکت SPI

● مثال: پارامترهای زمانی سیکل نوشتن

Sym	Characteristic	Min	Typ	Max	Units
F <sub>SCK</sub>	SPI Clock Frequency	DC	—	14	MHz
T <sub>DUTY</sub>	SCK Duty Cycle	45	—	55	%
T <sub>CSS</sub>	CS Setup Time	50	—	—	ns
T <sub>CSH</sub>	CS Hold Time	50	—	—	ns
T <sub>CSD</sub>	CS Disable Time	20	—	—	ns
T <sub>SU</sub>	Data Setup Time	10	—	—	ns
T <sub>HD</sub>	Data Hold Time	10	—	—	ns
T <sub>V</sub>	Output Valid from Clock Low	—	—	10	ns
T <sub>DIS</sub>	Output Disable Time	—	—	10	ns



در بالا تایمینگ آموزشی را ملاحظه می کنید کد در برنامه ارایه شده است و شمارنده ها بر اساس آن برنامه ریزی شده است. حال پارامترهای ADF4350 را در پایین ملاحظه می کنید. با توجه به آموزش داده شده پارامترها را برای برنامه ADF4350 را بدست می آوریم.

Table 2.

Parameter	Limit (B Version)	Unit	Test Conditions/Comments
t <sub>1</sub>	20	ns min	LE setup time
t <sub>2</sub>	10	ns min	DATA to CLK setup time
t <sub>3</sub>	10	ns min	DATA to CLK hold time
t <sub>4</sub>	25	ns min	CLK high duration
t <sub>5</sub>	25	ns min	CLK low duration
t <sub>6</sub>	10	ns min	CLK to LE setup time
t <sub>7</sub>	20	ns min	LE pulse width

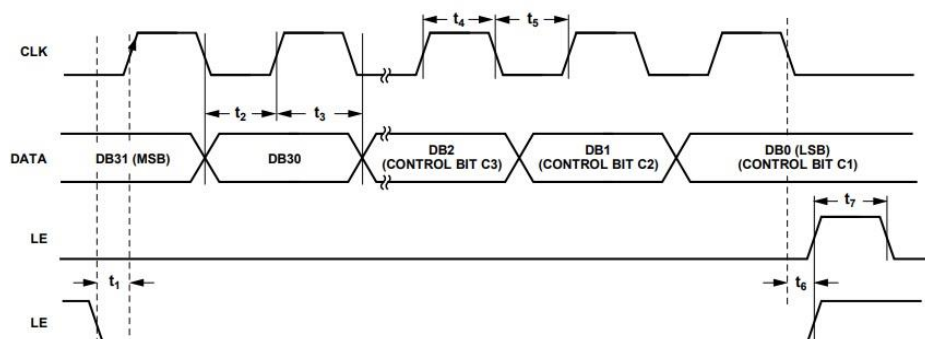


Figure 2. Timing Diagram

0735-002

- فرکانس کلاک FPGA را برابر 100 MHz و پریود زمانی آن 10 ns است. بنابراین دقت شمارنده و تاخیر های زمانی در حدود 10 ns است. حال ماکزیمم فرکانس آن را محاسبه می کنیم.

$$T_{SCK} > t_4 + t_5 = 25 \text{ ns} + 25 \text{ ns} = 50 \text{ ns (min)} \Rightarrow F_{SCK} < 20 \text{ MHz (max)}$$

برای راحتی کار دقت زمانی فرکانس انتقال 10 MHz با پریود زمانی 100 ns انتخاب می کنیم.

$$T_{SCK} = 100 \text{ ns} \Rightarrow F_{SCK} = 10 \text{ MHz}$$

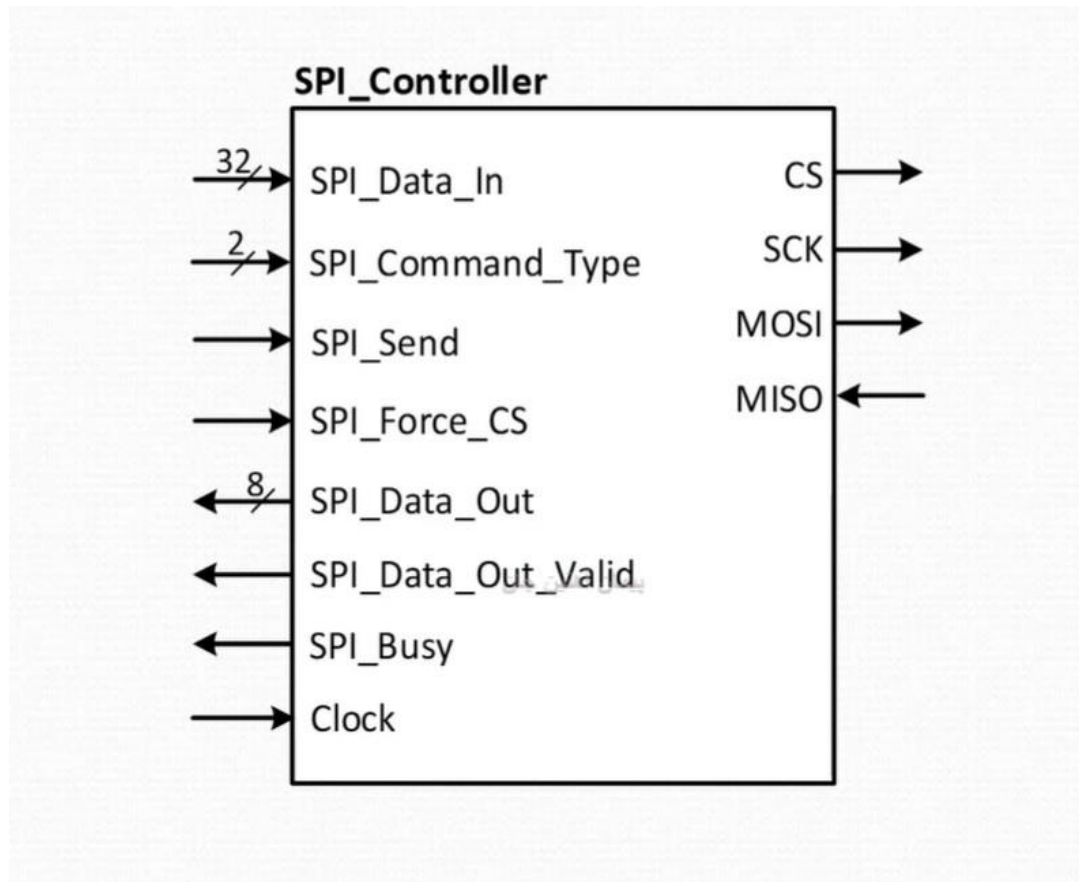
- جهت تولید فرکانس  $F_{SCK}$  از کلاک از یک کانتر چهار بیتی SCK\_Clock\_Divider استفاده می شود. دقت کنید در داخل پروسس در دستور شرطی این کلاک انتقال داده ساخته می شود.

- تاخیر بعدی  $T_{CSS}$  که همان LE setup time است که منیم 20 ns است و این مدت زمان بین لبه پایین رونده CS و لبه بالا رونده SCK است که در برنامه این مقدار بیشتر از این مقدار خواهد بود. این تاخیر توسط کانتر سه بیتی CS\_Disable\_Counter تولید می شود.

- برای رعایت نمودن زمان DATA to CLK setup time = 10 ns و DATA to CLK hold time = 10 ns بدلیل اعمال داده در دستور شرطی که کلاک داخلی SCK ساخته می شود همزمان با داده به آن اعمال می شود.

- تاخیر زمانی  $T_{CSH}$  که همان  $t_6 = 10 \text{ ns}$  است و برابر با زمان که به آخرین پالس لبه پایین رونده SCK اضافه می شود تا نوشتن کامل شود. زمان  $t_7$  که  $T_{CSD}$  توسط برنامه نویس جهت نوشتن بایستی رعایت شود.

بعد از مشخصات کلاک و کانترهای مورد نیاز به مرحله بعدی که تعیین پورت ها رسیده است. در طراحی که توسط آقای ثقفی شده دارای امکانات کاملی است که باعث پیچیده شدن مدار شده است. اگر مدار نوشتن 32 بیتی بسیار ساده خواهد بود. بنابراین ابتدا طراحی پیشرفته SPI را پیاده سازی کرده و آن را در حالت ساده شده در SPI\_ADF4350 به کار می بریم.



پورت های سمت راست پورت های انتقال full duplex spi می باشد.

CS = LE : که هنگام انتقال داده low می شود.

SCK : فرکانس انتقال داده است در مدت انتقال داده فعال می شود.

MOSI : خروجی SPI Master است.

MISO : ورودی SPI Master است.

پورتهای سمت چپ به ترتیب زیر است.

SPI\_Data\_In : ورود داده که می تواند به صورت 8 و 16 و 24 و 32 بیت باشد که در پروژه ما 32

بیتی است که توسط خط SPI\_Command\_Type تعیین می شود.

SPI\_Command\_Type : یک ورودی دوبیتی است که به ازای :

0 : انتقال 8 بیتی است.

1 : انتقال 16 بیتی است.

2: انتقال 24 بیتی است.

3 : انتقال 32 بیتی است.

در انتقال بیش از 8 بیتی می توان هم حالت write و read برنامه نویسی شود. در SPI\_ADF4350 انتقال 32 بیتی در حالت write است. در کارهای دیگر می تواند هشت بیت داده دستور به slave ارسال می شود و بقیه بیت ها داده های ارسالی از slave به Master است.

SPI\_Send : دستور شروع انتقال بیت ها را صادر می کند.

SPI\_Force\_CS : برای خواندن داده های بیش از 32 بیت تعبیه شده است.

SPI\_Data\_In : داده ها به صورت هشت بیت در حالت read دریافت می شود و سیگنال SPI\_Data\_Out\_Valid : نشان دهنده آماده بودن داده ورودی برای خواندن است.

SPI\_Busy : نشان دهنده انتقال بیت ها است و داده های جدید نمی تواند بوسیله SPI منتقل شود.  
Clock : سیگنال کلاک FPGA می باشد.

با توضیحاتی که ارائه شد با تنظیمات ساده می توان با این برنامه داده 32 بیتی را در ADF4350 نوشت.

**تنظیمات برنامه :**

1 . مقدار Command\_Type = 3 داده 32 بیتی است.

2. سیگنال Force\_CS = 1 و برنامه را برای مقدار 32 بیتی دو بار تکرار می کنیم.

3. برای تست مقدار 32 بیتی "FFAA5533" را به SPI اعمال می کنیم به شکل موج CS و MOSI و SCK که مطابق کاتالوگ می باشد. این تست نشان می دهد مدار درست کار می کند.

