بسمه تعالی

درس طراحی سیستم‌های نهفته مبتنی بر FPGA

آزمایش ۱: طراحی و پیاده‌سازی یک سیستم برای انتقال و پردازش اطلاعات

پردیس دانشکده‌های فنی دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

دکتر بیژن علیزاده

دستیاران آموزشی:

سیامک بیگ محمدی siamackbm@yahoo.com

علیرضا خادم arkhadem@ut.ac.ir

علی عباسی ابیانه abyaneh@outlook.com

پاییز ۱۳۹۷

مدت آزمایش: سه جلسه

اهداف آزمایش:

- ✓ آشنایی با برد DE2
- ✓ آشنایی با پورت سریال RS-232
- ✓ ارسال و دریافت سخت‌افزاری اطلاعات با PC
- ✓ ایجاد یک سیستم کامل دیجیتال با اتصال و کنترل چند ماژول مستقل
- ✓ مباحث پیشرفته‌تر
- ✓ آشنایی با PLL
- ✓ آشنایی با Constraint File
- ✓ آشنایی با Clock Routing

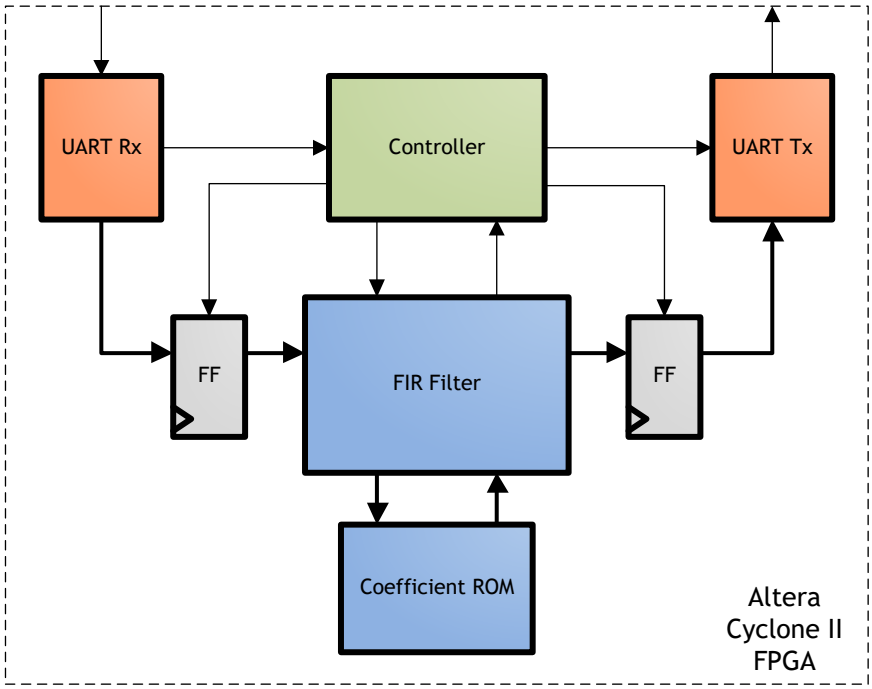
مقدمه

در این آزمایش به طراحی یک سیستم کامل شامل واحد ارسال و دریافت داده، واحد پردازشی و واحد نمایش اطلاعات می‌پردازیم. اطلاعات از طریق PC و پورت سریال به سیستم منتقل می‌شود و پس از پردازش، نتیجه به

PC ارسال می‌گردد. اطلاعات مورد نظر نمونه‌های سیگنال صوتی و پردازش مد نظر اعمال فیلتر دیجیتال بر روی این نمونه‌ها جهت استخراج نمونه‌های صوت فیلتر شده است.

شرح آزمایش

شمای کلی سیستم مورد نظر در این آزمایش در شکل ۱ آمده است. در این سیستم داده از طریق پورت سریال وارد شده و پس از پردازش (اعمال فیلتر FIR) از طریق پورت سریال به کامپیوتر داده می‌شود. کد نرم‌افزاری ارسال و دریافت داده سریال توسط PC به همراه گزارش آپلود شده است. هدف نهایی این آزمایش خواندن یک فایل صوتی، حذف نویز از آن به کمک فیلتر FIR و در نهایت ذخیره‌ی فایل خروجی است.



شکل ۱ شمای کلی آزمایش اول.

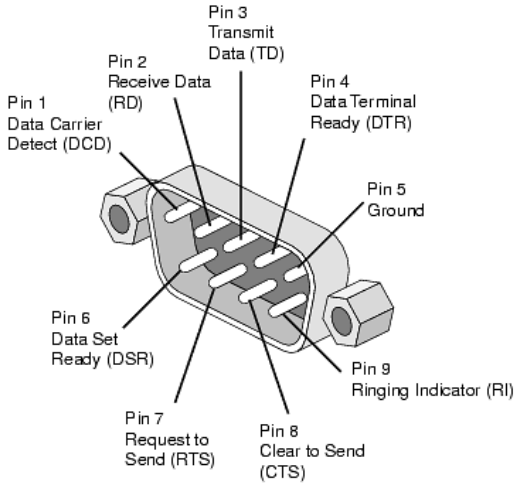
مراحل زیر را به ترتیب انجام دهید:

۱- ارتباط با کامپیوتر از طریق پورت سریال

پورت سریال یکی از ارتباطات خوب و مناسب کامپیوتر با دنیای بیرون به شمار می‌رود که اجازه می‌دهد اطلاعات به صورت دو طرفه انتقال یابد. این پورت با استاندارد و پروتکل RS-232 کار می‌کند. اغلب کامپیوترهای شخصی یک یا دو پورت COM برای واسط RS-232 دارند که معمولاً از یک کانکتور DE9 استفاده می‌کنند و توانایی یک انتقال تمام دوطرفه یا full-duplex را دارند. لازم به ذکر است امروزه با توجه به نیاز به سرعت‌های انتقال بالاتر،

پورت‌های سریال RS-232 جای خود را به سایر استانداردهای ارتباط سریال و خصوصاً USB داده‌اند. با این وجود نه‌تنها استفاده از پورت RS-232 هنوز منسوخ نشده است، بلکه در بسیاری از مواقع به دلایل اقتصادی و فنی استفاده از این پورت ترجیح داده می‌شود.

استاندارد RS-232 و به طور صحیح‌تر TIA/EIA-232، مشخصات الکتریکی، مکانیکی و عملکردی سیگنال‌های ارتباطی را مشخص می‌کند. این استاندارد اولین بار در سال ۱۹۶۲ برای استاندارد سازی ارتباط میان کامپیوترها (DTE[1]ها) و مودم‌ها (DCE[2]ها) و ایجاد امکان اتصال دستگاه‌های تولید شده توسط شرکت‌های مختلف مخابراتی وضع شد، اما کاربردهای گسترده‌تری خارج از این حوزه نیز یافت. در سال ۱۹۸۳ با انتشار کامپیوترهای شخصی توسط IBM، استاندارد RS-232 در این کامپیوترها در قالب کانکتور DE9 به کار گرفته شد. شکل ۲ کانکتور DE9 و شماره‌ی پین‌های آن را نشان می‌دهد.



شکل ۲ کانکتور DE9، پین‌ها و سیگنال‌های متناظر آن.

پایه‌های نشان داده شده در شکل برای ارتباط با مودم استفاده می‌شود و در این آزمایش فقط از ۳ پایه‌ی مهم زیر استفاده می‌کنیم:

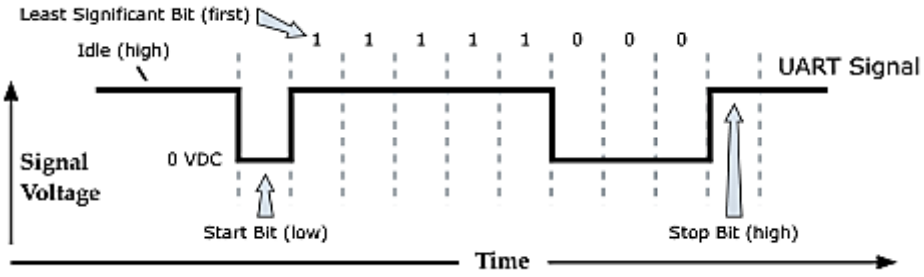
۱- پایه شماره ۲، RxD که همان داده سری دریافتی است.

۲- پایه شماره ۳، TxD که همان داده سری ارسالی است.

[1] Data Terminal Equipment
[2] Data Communicating Equipment

۳- پایه شماره ۵، GND که پایه زمین می باشد.

در ارتباط سریال یک بیت داده در هر لحظه ارسال می‌گردد، پس یک خط برای انتقال در هر جهت کافی است، اما استاندارد RS-232 پروتکل ارتباطی مشخصی را تعیین نمی‌کند. با این وجود در این استاندارد معمولاً از پروتکل UART[۳] استفاده می‌شود. در این پروتکل اطلاعات به صورت بسته های داده ۸ بیتی انتقال داده شده و نیز ابتدا بیت صفرام، سپس بیت ۱ ام و ... ارسال می‌گردند. توجه شود که این ارتباط به صورت آسنکرون می‌باشد، به این معنی که سیگنال کلاک به همراه داده ارسال نمی‌شود و قبل از شروع ارسال یا دریافت، فرستنده و گیرنده روی پارامترهای ارتباطی نظیر سرعت انتقال داده، فرمت داده و غیره بطور یکسان تنظیم می‌شوند. زمانی که روی خط ارتباطی، انتقال داده‌ای نداریم، فرستنده روی خط مقدار ۱ قرار می‌دهد. فرستنده قبل از شروع ارسال هر بایت داده، یک بیت start (با مقدار ۰) روی خط قرار می‌دهد. بعد از بیت شروع، هشت بیت داده با سرعت و فرمتی که قبلاً برای هر دو طرف تنظیم شده است، ارسال و دریافت می‌گردد. در انتها نیز فرستنده یک یا دو بیت stop (با مقدار ۱) به معنای اتمام ارسال ۸ بیت، روی خط قرار می دهد. به عنوان مثال شکل ۳ ارسال 0x1F را نشان می دهد.



شکل ۳ فریم داده‌ی 0x1F در پروتکل UART.

سرعت انتقال اطلاعات با baud rate مشخص می‌شود که نشان می‌دهد چند نمونه داده (که در انتقال UART هر بیت یک نمونه محسوب می‌شود) در ثانیه ارسال شده است. برای مثال baud ۱۰۰۰ یعنی در ثانیه ۱۰۰۰ بیت انتقال می‌یابد. به عبارت دیگر انتقال هر بیت ۱ میلی‌ثانیه طول می‌کشد. کامپیوترهای شخصی معمولاً فقط سرعت‌های مشخصی از انتقال داده را پشتیبانی می‌کنند که برابر ۱۲۰۰، ۹۶۰۰، ۳۸۴۰۰ و ۱۱۵۲۰۰ baud است. در baud ۱۱۵۲۰۰ هر بیت μs ۸/۷ طول می‌کشد و ارسال ۱۱ بیت (شامل یک بیت شروع، ۸ بیت داده و دو بیت خاتمه) حدود ۹۵/۵ میکروثانیه طول خواهد کشید. در این آزمایش به پیاده‌سازی گیرنده و فرستنده‌ی UART

[3] Universal Asynchronous Receiver/Transmitter

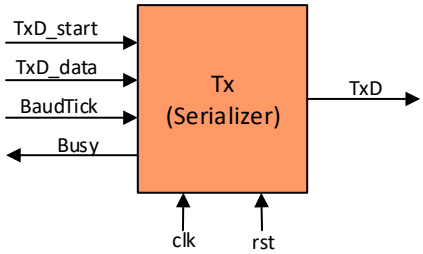
می‌پردازیم و از پورت RS-232 برای ارتباط با کامپیوتر استفاده خواهیم کرد. برای این کار مراحل زیر را انجام دهید:

گام ۱

ابتدا یک ماژول با نام Baud_Rate_Generator برای تولید نرخ baud ۱۱۵۲۰۰ بنویسید. ماژولی بنویسید که با دریافت پالس ساعت متغییر به صورت پارامتر (مثلاً ۵۰ مگاهرتز یا ۱۲۰ مگاهرتز) و شمارش آن به تعداد مورد نیاز، در هر ثانیه ۱۱۵۲۰۰ پالس تیک تولید کند. توجه نمایید خروجی ماژول شما نباید پالس ساعت باشد، بلکه در هر ثانیه از میان ۵۰ میلیون پالس ساعت (در صورت استفاده از کلاک ۵۰ مگاهرتز) در ۱۱۵۲۰۰ پالس مقدار خروجی این ماژول یک و در بقیه پالس‌ها صفر است. با شبیه‌سازی ماژول خود از صحت عملکرد آن مطمئن شوید.

گام ۲

گام دوم نوشتن ماژول فرستنده‌ی UART است. مطابق با شکل ۴ این ماژول داده‌ی ۸ بیتی ورودی را با فعال شدن سیگنال TxD_start به صورت سریال از طریق خط TxD ارسال می‌کند. پارامترهای UART به صورت ۸ بیت داده، ۱ بیت خاتمه و بدون بیت parity انتخاب شده است (این تنظیمات معمولاً به صورت 8n1 نمایش داده می‌شود. بیت parity یک تک بیت است که می‌تواند جهت بررسی صحت دریافت داده در سمت گیرنده استفاده شود). فرستنده در حالتی که در حال ارسال داده است از TxD_start صرف نظر می‌کند، اما خروجی busy را در این مدت ۱ نگه می‌دارد. ماژولی با عنوان Tx بنویسید که با استفاده از یک ماشین حالت و پالس تولید شده در بخش قبل (سیگنال BaudTick)، اطلاعات ۸ بیتی را با نرخ داده‌ی baud ۱۱۵۲۰۰ ارسال نماید.



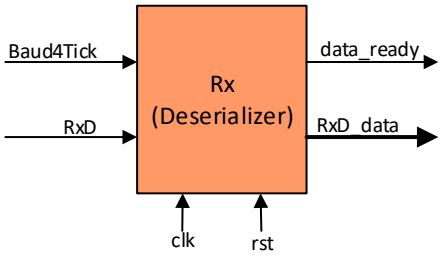
شکل ۴ ماژول فرستنده.

صحت عملکرد ماژول Tx را با شبیه‌سازی تحقیق کنید. سپس TxD_data را بر روی سوئیچ‌های SW[7:0] تنظیم نمایید. در این بخش و بخش‌های بعدی این آزمایش از سیگنال کلاک ۵۰ مگاهرتز استفاده نمایید. TxD_start را

به KEY[0] متصل کنید. پس از مشخص کردن تخصیص پین‌ها[4]، با استفاده از نرم‌افزار RealTerm، TeraTerm یا سایر نرم‌افزارهای ارتباط سریال، عملکرد صحیح ماژول Tx را بر روی برد DE2 تحقیق کنید. چند مقدار را روی سوئیچ‌ها تنظیم کنید و دریافت صحیح اطلاعات توسط PC را بررسی نمایید.

گام ۳

گام بعدی نوشتن ماژول گیرنده است. بلوک دیاگرام کلی ماژول گیرنده در شکل ۵ آمده است. در این ماژول داده از طریق خط RxD دریافت شده، به صورت بایت تبدیل می‌شود و روی باس RxD_data قرار می‌گیرد. هنگام آماده‌شدن داده، data_ready به مدت یک سیکل کلاک یک می‌شود.



شکل ۵ ماژول گیرنده.

مسأله‌ای که می‌تواند مشکل‌ساز شود، دریافت آسنکرون داده‌ها است. گیرنده‌ی UART به منظور سنکرون شدن با داده‌ی ورودی آن را با نرخی بالاتر از نرخ انتقال داده نمونه‌برداری می‌کند (مثلاً چهار برابر نرخ baud). این ماژول زمان تقریبی وسط بیت start را تشخیص می‌دهد و پس از آن با نرخ baud از سیگنال ورودی نمونه‌برداری می‌کند. ماژولی با عنوان Rx بنویسید که این کار را انجام دهد. ورودی Baud4Tick فرکانسی چهار برابر نرخ انتقال داده دارد. سیگنال Buad4Tick با ماژولی مشابه Baud_Rate_Generator (با مقدار شمارش متفاوت) تولید شده و وارد گیرنده می‌شود. ماژول شما می‌تواند نرخ نمونه‌برداری اولیه دیگری را انتخاب کند.

در نهایت ماژولی با نام UART بنویسید که شامل ماژول‌های Rx، Tx و دو ماژول Buad_Rate_Generator باشد و داده دریافتی توسط ماژول Rx را توسط ماژول Tx ارسال کند. همچنین داده دریافتی را روی LEDهای قرمز رنگ نشان دهید. از صحت عملکرد ماژول UART اطمینان حاصل کنید.

[4] Pin Assignment

۲- راه‌اندازی فیلتر FIR

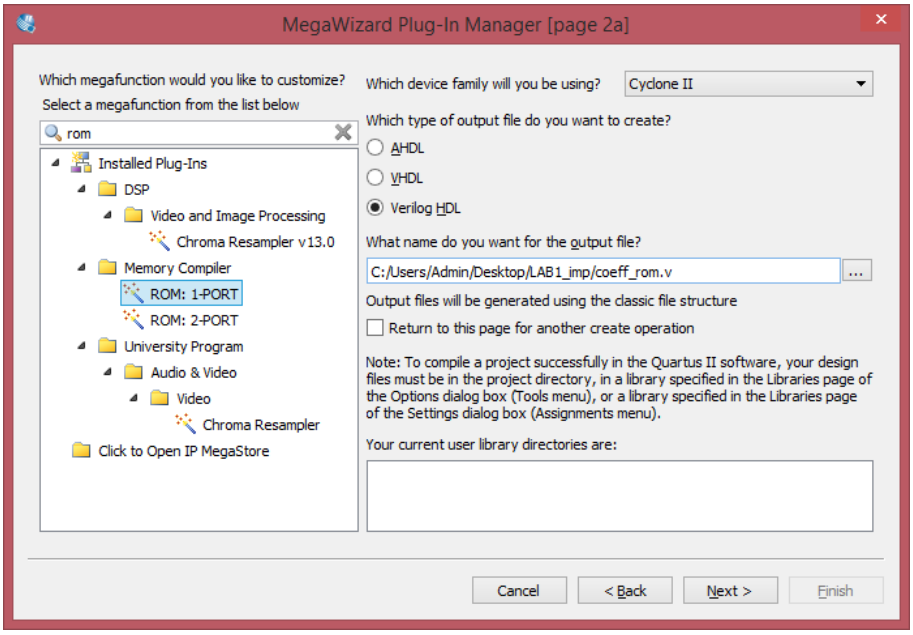
در این بخش فیلتر FIR طراحی شده در تکلیف کامپیوتری ۱ را با پارامترهای عرض بیت ۱۶ و طول فیلتر ۶۴ استفاده می‌کنیم. برای ذخیره‌سازی ضرایب از بلوک‌های حافظه‌ی اختصاصی استفاده خواهیم کرد. برای پیاده‌سازی حافظه در یک FPGA هم می‌توان از LUT[5]های مورد استفاده در پیاده‌سازی لاجیک و هم از بلوک‌های حافظه‌ی تخصیص داده شده در FPGA استفاده کرد. در Cyclone II حافظه‌های اختصاصی از نوع بلوک‌های M4K هستند که هر کدام شامل ۴۰۹۶ بیت حافظه هستند. در این حافظه‌ها امکان پیکربندی با اندازه‌های مختلف (به صورت تعداد کلمات و عرض هر کلمه) وجود دارد که اصطلاحاً به آن Aspect Ratio گفته می‌شود. در این آزمایش از بخشی از یک حافظه M4K با نسبت 16*64 (۶۴ خانه‌ی حافظه‌ی ۱۶ بیتی) استفاده می‌کنیم. گام‌های زیر را به ترتیب انجام دهید:

گام ۱

در گام اول یک حافظه‌ی ROM برای ذخیره‌سازی ضرایب ایجاد می‌کنیم و آن را تست می‌کنیم. بدین منظور مراحل زیر را انجام دهید:

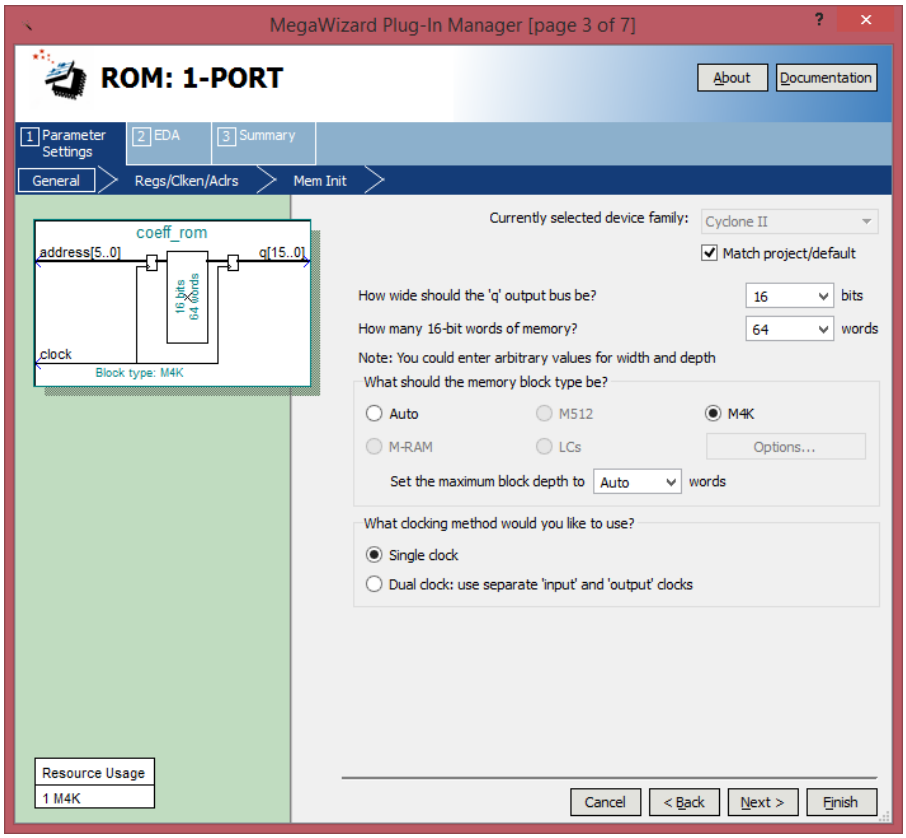
۱) در قسمت Tools> MegaWizar Plug-In Manager ایجاد یک megafunction جدید را انتخاب کنید. در صفحه‌ی دوم در قسمت Select a megafunction from the list below، از گروه Memory Compiler، حافظه‌ی ROM: 1-PORT را انتخاب کنید (شکل ۶). مطمئن شوید فایل خروجی با فرمت Verilog انتخاب شده است.

[5] Look-Up Table



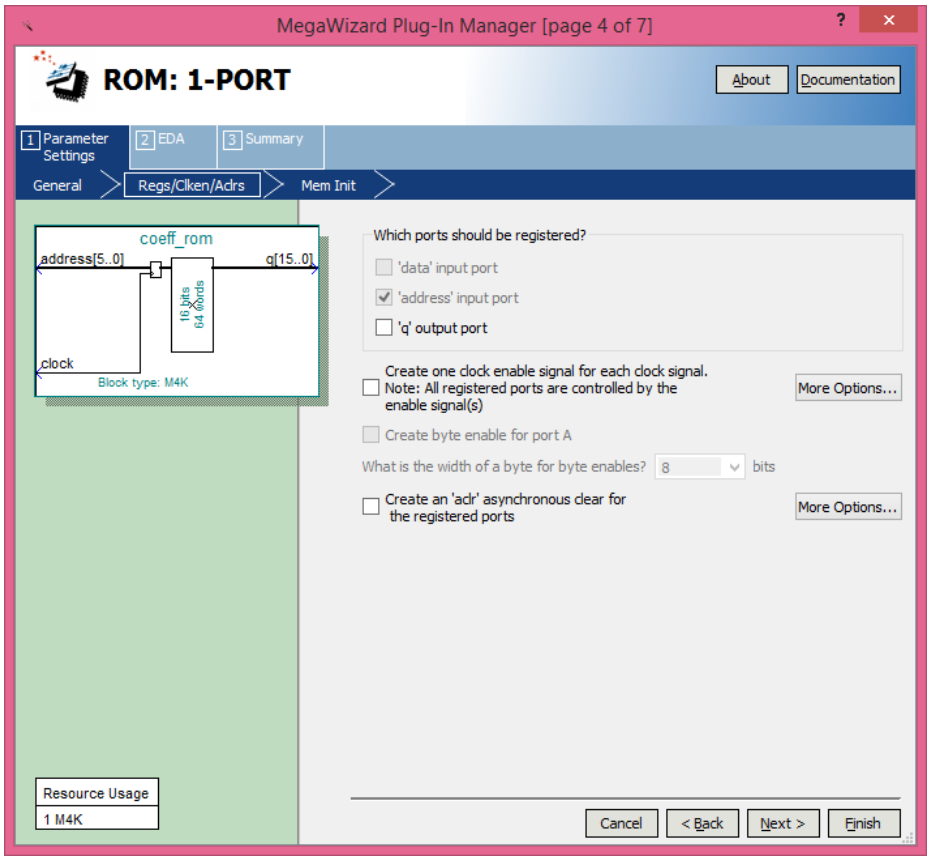
شکل ۶ انتخاب ROM: 1-PORT.

۲) در صفحه‌ی بعد ساختار حافظه را به صورت ۶۴ کلمه‌ی ۱۶ بیتی انتخاب کنید. نوع حافظه را نیز M4K قرار دهید (شکل ۷).



شکل ۷ انتخاب اندازه‌ی حافظه‌ی ضرایب.

۳) در صفحه‌ی چهارم ‘q’ output port را از حالت انتخاب خارج سازید تا رجیستر خروجی حذف گردد. طبعاً اگر در تمرین کامپیوتری اول تأخیر خواندن از مموری را یک سیکل کلاک فرض کرده‌اید، این کار را انجام ندهید. استفاده از رجیستر در ورودی بلوک M4K به دلیل سنکرون بودن این بلوک‌ها اجباری است.



شکل ۸ پیکربندی ورودی/خروجی بلوک حافظه.

۴) ضرایب فیلتر FIR در قالب یک فایل MIF[۶]. (و فایل مشابهی با فرمت hex) به همراه دستور کار در سایت قرار گرفته است. در صفحه‌ی بعد این فایل را انتخاب کنید.

۵) برای بقیه‌ی تنظیمات از مقادیر پیش‌فرض استفاده کنید.

۶) به منظور تست فایل ایجاد شده، آن را در یک ماژول نمونه‌گیری کنید. پورت address را به سوئیچ‌ها و پورت q (خروجی) را به LEDها متصل کنید. پس از پروگرام کردن FPGA فایل MIF (یا فایل معادل

[6] Memory Initialization File

hex) را باز کرده و با بررسی محتوای آدرس‌های مختلف حافظه از تطبیق مقادیر بر این فایل اطمینان حاصل کنید.

گام ۲

با اعمال تغییراتی در کد تکلیف کامپیوتری ۱ خود، حافظه تولیدی را با حافظه موجود در کد تکلیف کامپیوتری ۱ جایگزین کنید. با استفاده از تست‌بنچ مورد استفاده در تکلیف کامپیوتری ۱ از صحت عملکرد فیلتر مطمئن شوید.

نکته مهم: رم تولید شده داده را یک کلاک بعد از درخواست آن به شما تحویل می‌دهد. در صورت نیاز، حتماً طرح فیلتر خود را متناسب با این موضوع تغیر دهید.

۳- راه‌اندازی و تست سیستم کلی

در این بخش سیستم کلی پیاده‌سازی و تست خواهد شد. در حال حاضر تمامی اجزای شکل ۱ را به غیر از واحد Controller در اختیار داریم. مراحل زیر را انجام دهید تا واحد کنترلر نیز آماده گردد:

گام ۱

واحد کنترلر را با یک ماشین حالت پیاده‌سازی کنید به طوری که فرآیند زیر را کنترل کند:

۱) در حالت بی‌کاری کنترلر منتظر می‌ماند تا داده‌ای از ورودی دریافت شود. هر وقت واحد گیرنده (Rx) سیگنال data_ready را منتشر کرد، کنترلر باید داده را به فیلتر FIR انتقال داده و سیگنال کنترلی inputValid را به مدت یک سیکل فعال کند. چون محاسبات به صورت ۱۶ بیتی انجام می‌شود، باید ساختاری پیاده کنید که گیرنده پس از دو بار دریافت داده ۸ بیتی، ورودی فیلتر را فعال کند.

۲) سپس کنترلر منتظر می‌ماند تا محاسبات فیلتر FIR تمام شده و outputValid فعال گردد. کنترلر باید داده‌ی خروجی ۱۶ بیتی را در قالب دو داده‌ی ۸ بیتی بفرستد و برای این کار باید از طریق سیگنال‌های کنترلی TxD_start و busy با واحد فرستنده (Tx) در ارتباط باشد.

۳) پس از اتمام ارسال دو داده‌ی ۸ بیتی کنترلر به حالت اولیه باز می‌گردد و منتظر دریافت ورودی بعدی می‌شود.

گام ۲

حال که تمامی اجزای سیستم آماده شده است، آن‌ها را مطابق شکل ۱ به هم متصل کنید. برای تست این سیستم یک کد MATLAB در اختیارتان قرار خواهد گرفت. عملکرد این کد را در گزارش خود توصیف کنید. پس از انتقال طرح به FPGA و اطمینان از اتصال صحیح پورت سریال، درستی عملکرد سیستم را تحقیق کنید. در محیط MATLAB می‌توانید به صدای اصلی و صدای فیلتر شده گوش دهید. انتظار می‌رود خروجی فیلتر شده شامل نویز فرکانس بالا نباشد (فیلتر مورد استفاده یک فیلتر پایین‌گذر[7] است).

۴- نکات پیشرفته‌تر اما اساسی در طراحی سیستم‌های دیجیتال با FPGAها

در ادامه گام‌های زیر را انجام دهید.

گام ۱

برای آنکه ابزار سنتز بتواند زمانبندی مسیرهای ترکیبی بین فلیپ‌فلاپ‌های مدار را به درستی در هنگام سنتز و جایابی مدار انجام دهد، نیاز به دانستن فرکانس کلاک دارد. کلاک ورودی به مدار شما فرکانس 50 MHz دارد. یک constraint file (با پسوند sdc.) اضافه کنید که در آن ذکر کنید که فرکانس کلاک ورودی ۵۰ مگاهرتز می‌باشد. برای این کار می‌توانید از رابط گرافیکی Quartus کمک بگیرید. بدین منظور از منوی Assignments گزینه TimeQuest Timing Analyzer Wizard… را انتخاب کنید. در بخش Clock می‌توانید محدودیت مد نظر خود را اعمال کنید. محدودیت اعمال شده را در فایل sdc. تولید شده مشاهده و گزارش نمایید و آن را توضیح دهید. چه محدودیت‌های زمانی دیگری را می‌توان توسط این Wizard مشخص کرد؟

گام ۲

علاوه بر بلوک‌های ضرب‌کننده و حافظه، FPGAها جهت تسریع الگوریتم‌ها ساختارهای دیگری نیز به کار می‌برند. از جمله این ساختارها PLLها (Phased Locked Loop) هستند. به کمک این ساختارها می‌توان با داشتن یک کلاک با فرکانس مشخص، کلاکی با فرکانس دیگری تولید کرد. یک PLL به مدار خود اضافه کنید و از کلاک ۵۰ مگا هرتز ورودی به FPGA کلاک مورد نظر خود را تولید کنید. مشابه حافظه‌ها از مسیر Tools و زیرمنوی MegaWizard Plug-In Manager می‌توانید PLL را به صورت یک ماژول اضافه کنید. مدار خود را به کلاک

[7] Lowpass

جدید تولید شده متصل کنید. تلاش کنید بیشترین فرکانس ممکن را از مدار به دست آورید. (با روش هایی که در تمرین کامپیوتری اول گفته شد).

فراموش نکنید که ماژول‌های فرستنده و گیرنده UART به فرکانس کلاک ورودی حساس هستند. دقت نمایید که شمارنده‌های این ماژول را با تغییر پارامتر ماژول‌های Buad_Rate_Generator به درستی تغییر دهید. همچنین جهت اطمینان از قفل شدن حلقه PLL سیگنال lock خروجی آن را به یک LED متصل نمایید.

بررسی کنید که آیا نیازی به اضافه کردن فرکانس جدید مدار (فرکانس خروجی PLL) به constraint فایل می‌باشد یا خیر؟ (چرا؟)

گام ۳

Routing کلاک معمولاً با بقیه سیگنال‌ها متفاوت است (چرا؟). بنابراین باید به ابزار سنتز گفته شود که کدام سیگنال کلاک می‌باشد که آن را بر روی مسیر های از پیش مشخص کلاک در FPGA سنتز نماید.

این کار در Quartus II به صورت خودکار انجام می‌شود (این ابزار سنتز چگونه کلاک را شناسایی می‌کند و این کار را انجام می‌دهد؟ اگر کلاک های شناسایی شده در مدار شما بیشتر از مسیرهای از پیش مشخص شده در FPGA باشد با چه منطقی مناسب‌ترین سیگنال‌ها را برای انتقال روی شبکه‌های کلاک شناسایی می‌کند؟)

حال شما باید این کار را دستی انجام دهید. برای این کار باید حالت خودکار Quartus II را خاموش کنید. (چگونه؟) سپس کلاک مورد نظر (کلاک خروجی PLL) را به عنوان کلاک معرفی کنید. (آیا کلاک ورودی ۵۰ مگاهرتز باید روی مسیر مشخص کلاک route شود؟ چرا؟). جهت راهنمایی به مراجع [1] و [2] مراجعه کنید. در بعضی از FPGAها با توجه به محدودیت‌های موجود در طراحی باید این کار را انجام داد. مثلاً اگر شما کلاک خود را به ۵ PLL متصل کنید شاید نیاز باشد که این کار را بکنید. چرا این کار لازم است؟

نکات مهم:

۱) رعایت استایل کدنویسی ارائه شده در کلاس ضروری است.

۱) نیازی به گزارش تمامی مراحل انجام آزمایش نیست اما اهداف، کلیات، مقدار پارامترهای مختلف، جواب سوالات مطرح شده، شرایط درستی‌سنجی، نتیجه‌ی درستی‌سنجی و سایر نکات مهم را حتماً در گزارش خود قید نمایید.

۲) پیروی از قالب خا صی در گزارش مد نظر نیست اما ترتیب گزارش باید مشابه ترتیب مطالب در د ستور کار آزمایش باشد.

۳) آپلود فایل‌های شبیه‌سازی به همراه فایل گزارش ضروری است.

۴) ماشین های حالت خود را حتما در گزارش رسم کنید.

۵) در بخشی که باید بیشترین فرکانس ممکن را بگیرید نمره آن بخش به صورت خطی محاسبه می شود. به این صورت که کمترین فرکانس قابل قبول 50 MHz است و بیشترین فرکانس ممکن ۱۵۰ مگاهرتز می‌باشد و نمره به صورت خطی (٪۰ برای 50 MHz و ٪۱۰۰ برای 150 MHz) بین این دو تقسیم می‌شود.

مراجع

[1] “How can I assign a PLL output clock to a Global Clock Network?” Online: https://www.altera.com/support/support-resources/knowledge-base/solutions/rd03182011_985.html

[2] “How do I assign a clock in my design to use specific global, regional, dual-regional, or periphery clock networks?” Online: https://www.altera.com/support/support-resources/knowledge-base/solutions/rd09282011_171.html

موفق باشید

۹۷/۷/۲۱