



بسمه تعالی

درس طراحی سیستم‌های نهفته مبتنی بر FPGA

آزمایش ۳: طراحی سیستم نهفته مبتنی بر پردازنده‌ی Nios II و اضافه کردن دستورات اختصاصی به این پردازنده

دانشکده‌گان فنی دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

دکتر بیژن علیزاده

دستیاران آموزشی:

ali.ghaemi@ut.ac.ir

علی قائمی

mahsa.rasti80@ut.ac.ir

مهسا راستی

پاییز ۱۴۰۲

مدت آزمایش: سه جلسه

اهداف آزمایش:

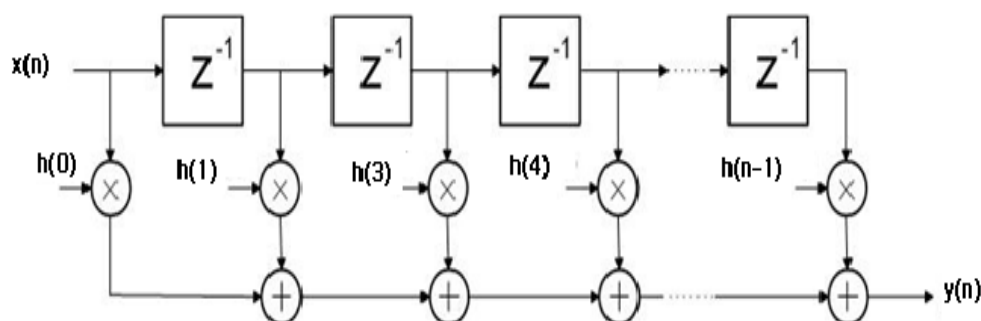
- ✓ آشنایی با طراحی سطح بالای فیلتر FIR در محیط MATLAB.
- ✓ آشنایی با طراحی سخت‌افزاری سیستم نهفته مبتنی بر پردازنده‌ی Nios II.
- ✓ اضافه کردن دستورات اختصاصی در پردازنده Nios II.
- ✓ بررسی مزایا و معایب استفاده از دستورات اختصاصی در یک پردازنده.

مقدمه

در آزمایش دوم با استفاده از سخت‌افزاری که از قبل طراحی شده بود (DE2 Media Computer)، با پردازنده‌ی Nios II و برنامه‌نویسی آن به زبان C آشنا شدیم. در این آزمایش ابتدا با طراحی سطح بالای فیلتر مورد استفاده در آزمایش اول آشنا می‌شویم. این فیلتر در محیط MATLAB طراحی و کد RTL آن استخراج می‌شود. سپس در محیط طراحی سخت‌افزار Qsys، فیلتر طراحی شده را به عنوان دستور اختصاصی به پردازنده Nios II در سیستم DE2 Media Computer اضافه می‌کنیم. در نهایت در محیط Eclipse، عملکرد حذف نویز را در این سیستم به صورت نرم‌افزاری با و بدون استفاده از دستور اختصاصی پیاده‌سازی، درستی‌سنجی و مقایسه می‌کنیم.

شرح آزمایش

در بخش اول این آزمایش به طراحی سطح بالای یک فیلتر پایین گذر FIR (شمای کلی در شکل ۱ قابل مشاهده است) و استخراج کد RTL آن به کمک نرم افزار MATLAB و درستی سنجی کد RTL تولید شده می پردازیم. بخش دوم به اضافه کردن دستورات اختصاصی به پردازنده Nios II اختصاص دارد که ابتدا باید فیلتر مورد نظر را به صورت نرم افزاری و سپس به صورت سخت افزاری با دستور اختصاصی پیاده سازی کنید. برای این منظور، مراحل زیر را به ترتیب انجام دهید.



شکل ۱- فیلتر FIR.

۱- طراحی سطح بالای فیلتر FIR و سنتز آن به کد RTL

اگرچه طراحی سطح بالای فیلترها در درس "طراحی در سطح سیستم" مفصل بحث می شود، در اینجا روند بطور مختصر و مفید توضیح داده شده است. در محیط MATLAB دستور filterDesigner را اجرا کنید. محیط گرافیکی ابزار طراحی و آنالیز فیلتر دیجیتال نمایش داده می شود. در این محیط با استفاده از مشخصات زیر یک فیلتر پایین گذر طراحی کنید. انتظار می رود یک فیلتر با درجه ۶۳ (یعنی با ۶۴ ضریب) طراحی شود (شکل ۲).

Design Method = FIR (Equiripple)

Filter Order = Minimum order

Density Factor = 20


Fs = 48000 Hz

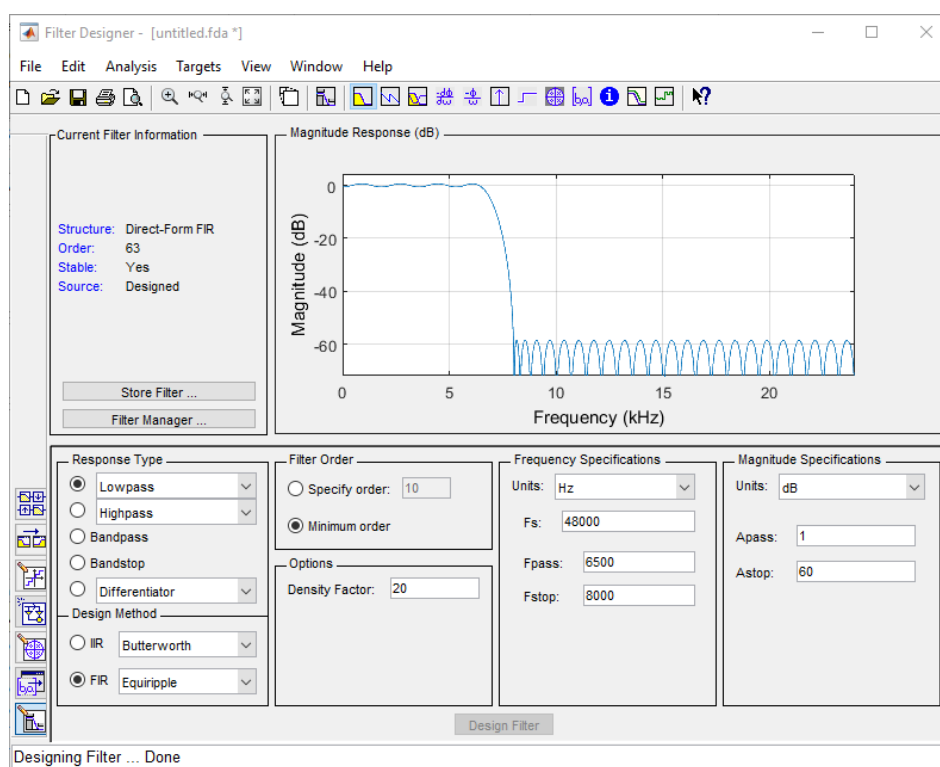
Fpass = 6000 Hz

Fstop = 7500 Hz


Apass = 1 dB

Astop = 60 dB

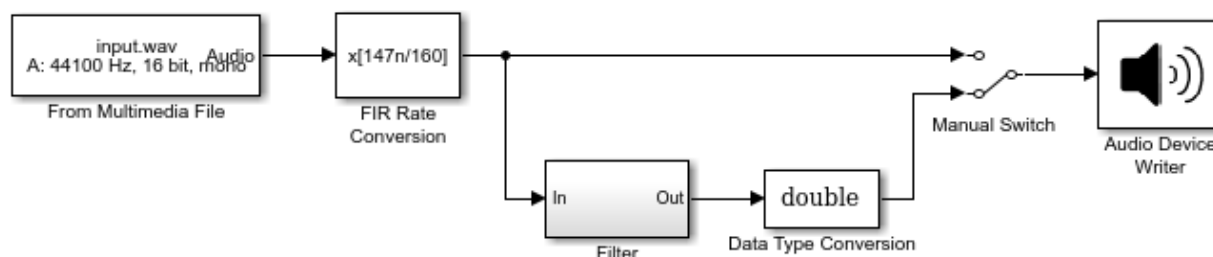
دلیل انتخاب فرکانس نمونه‌برداری ۴۸ کیلوهرتزی، عملکرد Codec صدای موجود در برد DE2 با این فرکانس است (در حالت پیش فرض). برای پیاده‌سازی سخت‌افزاری فیلتر به صورت ممیز ثابت لازم است ابتدا ضرایب آن را کوانتیزه کنید. برای این کار روی آیکون  کلیک کنید. Filter arithmetic را روی Fixed-point تنظیم کنید. در سمت راست سه tab شامل ضرایب، ورودی/خروجی و سیگنال‌های داخلی فیلتر ظاهر می‌شود. ورودی را روی ۲۴ بیت با ۲۳ بیت اعشار تنظیم کنید. همچنین خروجی را روی ۳۲ بیت با ۳۰ بیت اعشار تنظیم کنید (برای این کار باید Filter precision را روی Specify all تنظیم کنید). دلیل تغییر تعداد بیت‌های صحیح، خارج نشدن سیگنال از دامنه دینامیکی قابل نمایش در اثر ضرب و جمع‌های متوالی در فیلتر است. توجه نمایید که با این کار دامنه سیگنال خروجی صدا تقریباً نصف می‌شود. تفاوت حالت Saturate با حالت Wrap در زمان تنظیم مد Overflow چیست؟ سایر تنظیمات را تغییر ندهید. روی Apply کلیک کنید تا ضرایب فیلتر کوانتیزه شود. کوانتیزه کردن ضرایب فیلتر چه مزایا و معایبی دارد؟



شکل ۲- محیط گرافیکی ابزار filterDesigner.

برای درستی‌سنجی عملکرد فیلتر در محیط Simulink، با کلیک روی آیکون  وارد پنل Realize Model شوید و روی دکمه Realize Model کلیک کنید. نوع پردازش ورودی را تغییر ندهید (به صورت frame based باشد) تا امکان پردازش real-time و پخش صدا فراهم شود. یک فایل Simulink باز می‌شود که شامل فیلتر

طراحی شده است. بلوک‌های موجود در شکل ۳ را به مدل اضافه کنید و پس از ذخیره‌سازی آن، به صدای فیلتر شده و نشده گوش داده و از عملکرد صحیح فیلتر اطمینان حاصل کنید. مدل Simulink را به همراه گزارش آپلود کنید. توجه داشته باشید که مقدار FIR rate conversion باید مضربی از ۱۴۷ باشد (مثلاً ۱۴۷۰).



شکل ۳- مدل Simulink.

در ادامه، به محیط filterDesigner برگردید. از منوی Targets گزینه Generate HDL... را انتخاب کنید. در پنجره ظاهر شده زبان Verilog را انتخاب کنید. در برگه Filter Architecture می‌توانید معماری کاملاً موازی (شامل ۶۴ ضرب‌کننده و جمع‌کننده)، کاملاً سریال (شامل فقط یک ضرب‌کننده و جمع‌کننده) و یا حالتی مابین این دو را انتخاب کنید. همچنین معماری بدون استفاده از ضرب‌کننده نیز موجود است. به راهنمای MATLAB مراجعه کنید و عبارت Filter Design HDL Coder را جستجو کنید. راهنمای کار با این ابزار و مثال‌های فراوانی موجود است. با تنظیمات موجود در این ابزار آشنا شوید.

حال فیلتر را در دو حالت کاملاً موازی و کاملاً سریال Generate کنید. بدین ترتیب توصیف سطح بالای فیلتر به کد RTL سنتز خواهد شد. این دو کد را به دقت بخوانید و در گزارش خود معماری آن‌ها را رسم کرده و عملکرد این دو را توضیح داده و آن‌ها را مقایسه کنید.

۱-۱- شبیه‌سازی کد تولید شده توسط MATLAB

در برگه Testbench در زیربرگه Stimuli تیک User defined response را فعال کنید (بقیه ورودی‌ها را غیر فعال کنید).

ابتدا دستورهای زیر را اجرا کنید.

```
>> [inputs Fs] = audioread('input.wav');
>> inputs48k = resample(inputs,160,147);
```

آرایه inputs48k را به عنوان ورودی موردنظر وارد نمایید. فیلتر را در مد کاملاً سریال قرار دهید. در Tab مخصوص Test Bench در زیربرگه Configuration مقدار حاشیه خطا بر حسب بیت را صفر انتخاب کنید (انتظار داریم دقیقاً خروجی مطلوب تولید شود). فیلتر و testbench آن را Generate کنید (مد کاملاً سریال).

۱-۲- پیاده‌سازی

کدهای فیلتر را که در دو حالت کاملاً موازی و کاملاً سریال Generate کردید را به کمک Quartus برای برد DE2 سنتز کنید و میزان استفاده از منابع FPGA را در دو حالت مقایسه کنید. توجه کنید که پین‌های مشترکی را در هر دو سنتز استفاده نمایید. نتایج خود را گزارش کنید.

✓ خواسته‌ها

- فیلترهای کاملاً سریال، کاملاً موازی و سریال-موازی ساخته شده توسط MATLAB را ارسال کنید.
- مدل Simulink در محیط MATLAB را به همراه گزارش آپلود کنید.
- تفاوت فیلترهای کاملاً سریال و کاملاً موازی را گزارش کنید و معماری آن‌ها را رسم کرده و عملکرد این دو را توضیح دهید.
- میزان استفاده از منابع FPGA را در دو حالت مقایسه کنید.

۲- اضافه کردن دستور اختصاصی به پردازنده‌ی Nios II

در آزمایش دوم عملیات فیلترکردن جهت اضافه کردن echo به صدا توسط پردازنده‌ی Nios انجام شد. فیلتر مورد استفاده در این آزمایش فقط دو ضریب داشت و در نتیجه این عملیات، محاسبات کمی می‌طلبید، به‌طوریکه پردازنده قادر بود اضافه کردن echo را به صورت بی‌درنگ^۱ انجام دهد. در این بخش از آزمایش قصد داریم فیلتر

¹ Real-time

حذف نویز مورد استفاده در آزمایش اول را در دو حالت توسط پردازنده‌ی Nios اعمال کنیم: (۱) به صورت کاملاً نرم‌افزاری و (۲) با اضافه کردن دستور اختصاصی و سخت‌افزار مربوط به آن به پردازنده‌ی Nios. در حین کار با نرم‌افزار ممکن است نیاز باشد که داده را cast کنید. به این معنی که بدون تغییر در بیت‌های ذخیره شده مربوط به داده، نحوه تعامل با داده متفاوت شود. به طور مثال داده val را به صورت integer ذخیره کرده‌اید و حال نیاز به تقسیم این داده به عدد اعشاری دارید که می‌تواند خروجی اعشاری بدهد. در این صورت این متغیر به صورت val(float) به متغیر اعشاری تبدیل شده و تقسیم هم به صورت اعشاری انجام خواهد شد. با توجه به اینکه داده‌های ذخیره شده توسط میکروفون، داده‌های ۸ بیتی هستند، ممکن است نتیجه انجام عملیات ضرب و یا جمع بر روی این داده‌ها بیشتر از ۸ بیت نیاز داشته باشد. به همین دلیل رجیستر ذخیره‌سازی نتیجه باید بزرگ‌تر باشد. برای اینکه نتیجه حاصل از عملیات قبل از ذخیره در رجیستر نتیجه، به درستی و با طول مناسب محاسبه شده باشد، باید آن را به داده با طول بیشتر cast کنید.

برای پیاده‌سازی دستور اختصاصی گام‌های زیر را انجام دهید:

گام ۱

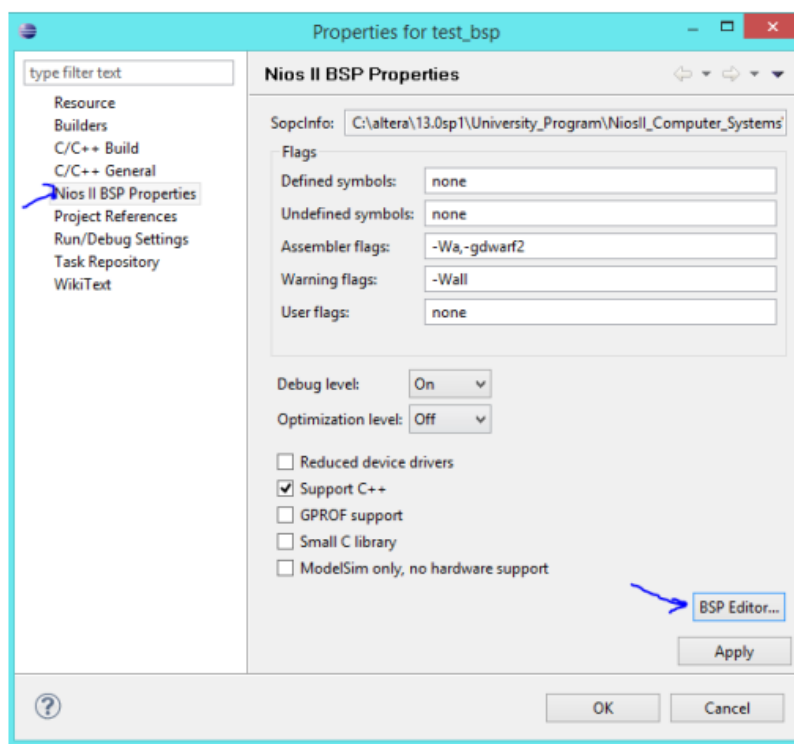
در محیط Quartus، از منوی Tools گزینه‌ی Nios II Software Build Tools for Eclipse را انتخاب کنید. پروژه‌ی جدیدی را بر مبنای فایل sopcinfo. مربوط به سیستم DE2 Media Computer که اطلاعات سخت‌افزاری سیستم را برای SBT جهت تولید اتوماتیک درایورها و سیستم HAL² ارائه می‌دهد، ایجاد کنید. کد نرم‌افزاری آزمایش ۲ را به گونه‌ای تغییر دهید که با فشردن دکمه سوم، به جای echo، عملیات حذف نویز انجام شود.

لازم به ذکر است به دلیل عملیات جمع متوالی برای محاسبات میانی، لازم است از متغیر ۶۴ بیتی (به جای ۳۲ بیتی) استفاده شود (توصیه می‌شود که کد افزوده شده را به صورت یک تابع نوشته و به برنامه اضافه کنید). سیستم DE2 Media Computer را روی FPGA پروگرام کرده و با اجرای نرم‌افزار، مشاهدات خود را از نظر سرعت اجرا و کیفیت خروجی گزارش کنید. زمان انجام محاسبات را اندازه‌گیری و گزارش نمایید. توجه نمایید که نمونه‌های ۲۴ بیتی صدا در بیت‌های MSB از آرایه‌های ۳۲ بیتی r_buf و l_buf ذخیره شده است. بنابراین برای جمع اندازه نمونه‌ها لازم است از متغیر با عرض بیت بالاتر (۶۴ بیتی در نرم‌افزار) استفاده کنید. زمان اجرای الگوریتم محاسبه‌ی دامنه‌ی متوسط را برای مقایسه‌های بعدی اندازه بگیرید و در گزارش ذکر کنید. توجه کنید که این کار حتماً و حتماً با استفاده از کتابخانه‌های مرتبط با زمان انجام شود. کارهایی مانند کرنومتر و ساعت گرفتن قابل

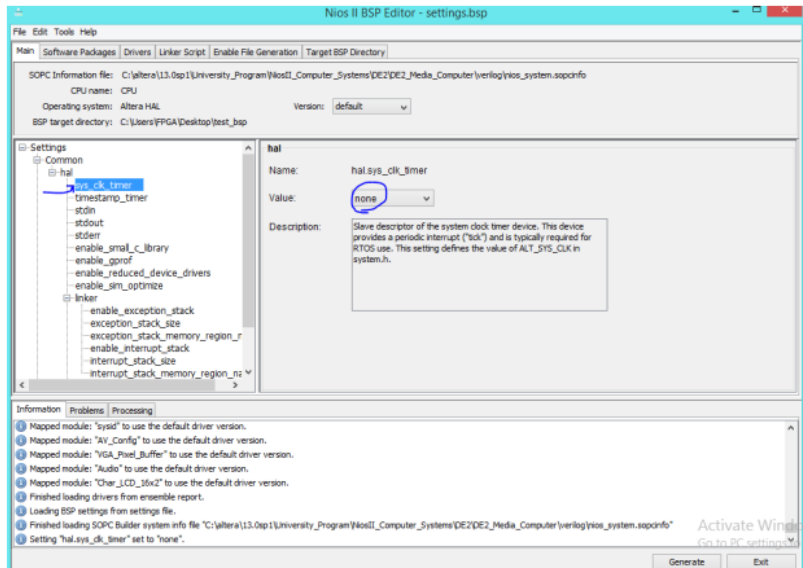
² Hardware Abstraction Layer

قبول نبوده و نمره‌ای به آن‌ها تعلق نمی‌گیرد. همچنین حتما باید روند طی شده برای به دست آوردن زمان به طور کامل گزارش شود (می‌توانید از توابعی که در درس به آن اشاره شد، استفاده کنید). برای اندازه‌گیری زمان می‌توانید از تایمر timestamp استفاده کنید که نحوه استفاده از آن در ادامه آمده است.

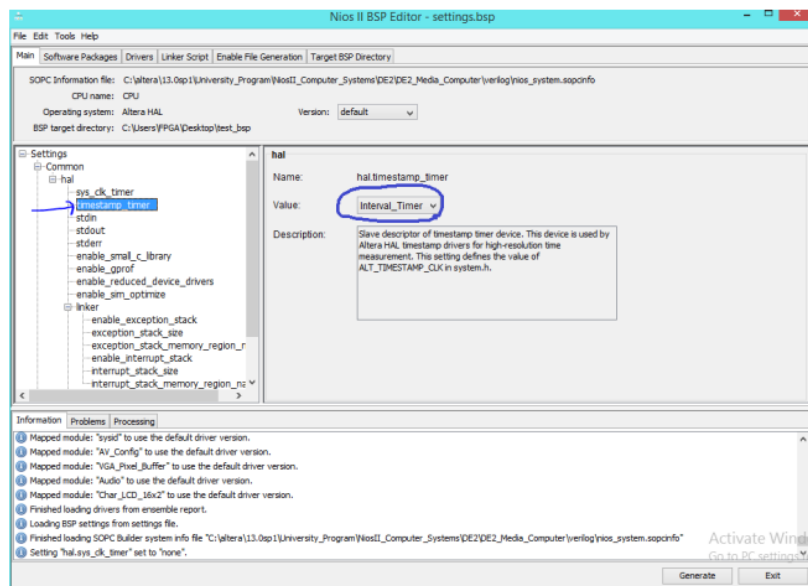
برای استفاده از timestamp timer باید در تنظیمات پروژه bsp خود در نرم‌افزار eclipse مطابق شکل ۴ تا شکل ۶، واحد timer interval را از timer clock system گرفته و آن را به timestamp timer اختصاص دهید. سپس با انتخاب گزینه generate تنظیمات جدید را تولید کنید.



شکل ۴- تنظیمات پروژه bsp.



شکل ۵- تنظیمات پروژه bsp.



شکل ۶- تنظیمات پروژه bsp.

سپس در کد خود کتابخانه "sys/alt_timestamp.h" را include کرده و از توابع زیر استفاده کنید:

alt_timestamp_freq()

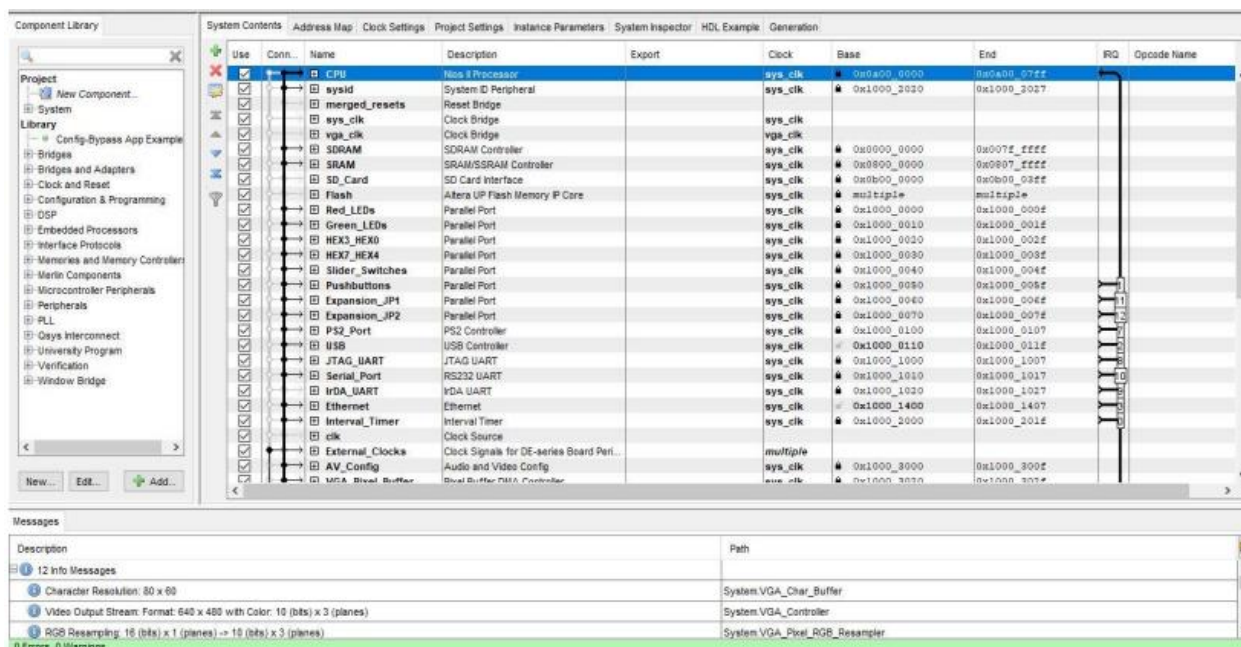
alt_timestamp_start()

alt_timestamp()

راهنمایی: می‌توانید برای دریافت ضرایب در filterDesigner از منوی File گزینه Export نوع Coefficient File فرمت Decimal و یا Hexadecimal فایل ضرایب را با هر فرمتی مثل txt. ذخیره و از آن در روند نوشتن کد نرم‌افزاری استفاده نمایید.

گام ۲

روند اصلی طراحی سخت‌افزاری سیستم در نرم‌افزار Qsys انجام می‌شود. در محیط Quartus، از منوی Tools ابزار Qsys را انتخاب کنید. در سمت چپ پنجره Qsys، کتابخانه‌ای از اجزای قابل استفاده در سیستم نشان داده شده و در سمت راست در هر Tab، مشخصات مختلف سیستم قابل تنظیم است. Tab اصلی، بخش System Contents می‌باشد که اجزای تشکیل‌دهنده سیستم و اتصالات بین آن‌ها را نشان می‌دهد (شکل ۷). توجه کنید که Qsys رفتار نرم‌افزاری سیستم (از جمله مکان قرارگیری دستورات در حافظه) را مشخص نمی‌کند.



شکل ۷- تنظیمات Qsys.

فایل qsys. سیستم DE2 Media Computer را در یک فولدر کپی کرده و آن را با Qsys باز کنید (حتما از این فایل کپی بگیرید). دستور اختصاصی در نهایت به این سیستم اضافه خواهد شد. فیلتر سریال تولید شده توسط نرم‌افزار MATLAB را با کمترین تغییر (تغییرات در گزارش ذکر شوند) به صورت یک دستور Multicycle با تعداد سیکل متغیر برای پردازنده Nios II تعریف کنید. نیازمندی‌های تعریف یک دستور Multicycle چیست؟

برای پاسخ به این سوال و انجام این بخش لازم است اسلایدهای درس را مطالعه نمایید و برای اطلاعات بیشتر به مرجع [۱] را نیز مطالعه کنید. همچنین برای راهنمایی قدم به قدم ساخت انواع Component ها در Qsys، می‌توانید به مرجع [۲] مراجعه نمایید. علاوه بر این نیازمندی‌ها، لازم است داده ورودی این فیلتر را به ۳۲ بیت تغییر دهید؛ اما به صورت داخلی از ۲۴ بیت MSB آن در محاسبات استفاده نمایید. روند کار را در Qsys انجام دهید (نه SOPC Builder). توجه کنید که در مرحله آخر حتماً تعداد عملوندها مقدار ۱ و تعداد سیکل مورد نیاز برای اجرای صحیح سخت‌افزار به درستی تنظیم شوند.

گام ۳

پس از تولید کد سخت‌افزار کلی توسط Qsys، پروژه Quartus مربوط به DE2 Media Computer را باز کرده و دوباره سنترز نمایید. سیستم جدید را روی FPGA پروگرام کنید. به کمک Nios II SBT نرم‌افزار گام ۱ را به گونه‌ای تغییر دهید که دقیقاً همان کار را این بار با دستور اختصاصی انجام دهد (لازم است بر اساس فایل sopcinfo. جدید، BSP را دوباره بسازید). با اجرای این کد، نتایج را (زمان اجرا و میزان استفاده از منابع FPGA) با گام ۱ مقایسه کنید.

راهنمایی: یک Custom instruction macro برای فیلتر فوق ساخته شده که باید از آن استفاده کنید.

توجه نمایید که پس از پروگرام کردن FPGA با فایل sof. تولید شده احتمالاً به دلیل آنکه لایسنس NiosII/f را نداشته‌اید، پنجره‌ای ظاهر خواهد شد که بیانگر موقتی بودن عملکرد کد است. برای آنکه پردازنده درست کار کند، دکمه Cancel در این پنجره را انتخاب نکنید.

نکات مهم

- (۱) بخش مهمی از این آزمایش بدون استفاده از سخت‌افزار DE2 قابل انجام است. مطالعه مرجع [۱] جهت انجام آزمایش ضروری است.
- (۲) در این آزمایش کلیات، مقدار پارامترهای مختلف، مقایسه نتایج، تغییرات انجام شده و سایر نکات مهم را حتماً در گزارش خود قید نمایید.

۳) آپلود کدها به همراه فایل گزارش ضروری است. لازم است کدهای بخش‌های مختلف را در پوشه‌های جداگانه آپلود کنید (همه را در داخل یک فایل zip قرار دهید).

۴) استایل کدنویسی و کامنت‌گذاری صحیح امتیاز مثبت دارد.

۵) پیروی از قالب خاصی در گزارش مد نظر نیست، اما ترجیحاً می‌توانید از قالب ارائه شده برای تکالیف کامپیوتری استفاده نمایید.

مراجع

[1] Intel (Altera), “Nios II Custom Instruction User Guide,” 2017.

Available:

https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug_nios2_custom_instruction.pdf

[2] Intel (Altera), “Making Qsys Components,” 2017.

Available:

ftp://ftp.intel.com/pub/fpgaup/pub/Intel_Material/17.0/Tutorials/making_qsys_components.pdf

موفق و سلامت باشید.