



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

پرديس دانشكده‌های فنی دانشگاه تهران

دانشكده مهندسی برق و کامپیوتر

دکتر بیژن علیزاده

دستیاران آموزشی:

siamackbm@yahoo.com

سیامک بیگ محمدی

hamid.imani74@gmail.com

حمیدرضا ایمانی

mohsenfathi7@yahoo.com

محسن فتحی

پاییز ۱۳۹۷

مدت آزمایش: دو 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 یک سیستم از قبل طراحی شده با نام 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، آزمون و مقایسه می‌کنیم.

شرح آزمایش

در بخش اول این آزمایش به طراحی سطح بالای یک فیلتر پایین‌گذر FIR و استخراج کد RTL آن به کمک نرم‌افزار MATLAB می‌پردازیم. بخش دوم به اضافه کردن دستورات اختصاصی به پردازنده‌ی Nios II اختصاص دارد. مراحل زیر را به ترتیب انجام دهید:

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

در محیط MATLAB (ترجیحاً نسخه 2017a) دستور filterDesigner را اجرا کنید. محیط گرافیکی ابزار طراحی و آنالیز فیلتر دیجیتال نمایش داده می‌شود. در این محیط با استفاده از مشخصات زیر یک فیلتر پایین‌گذر طراحی کنید. انتظار می‌رود یک فیلتر با درجه ۶۳ (یعنی با ۶۴ ضریب) طراحی شود (شکل ۱).

Design Method = FIR (Equiripple)

Filter Order = Minimum order

Density Factor = 20


Fs = 48000 Hz


Fpass = 6000 Hz

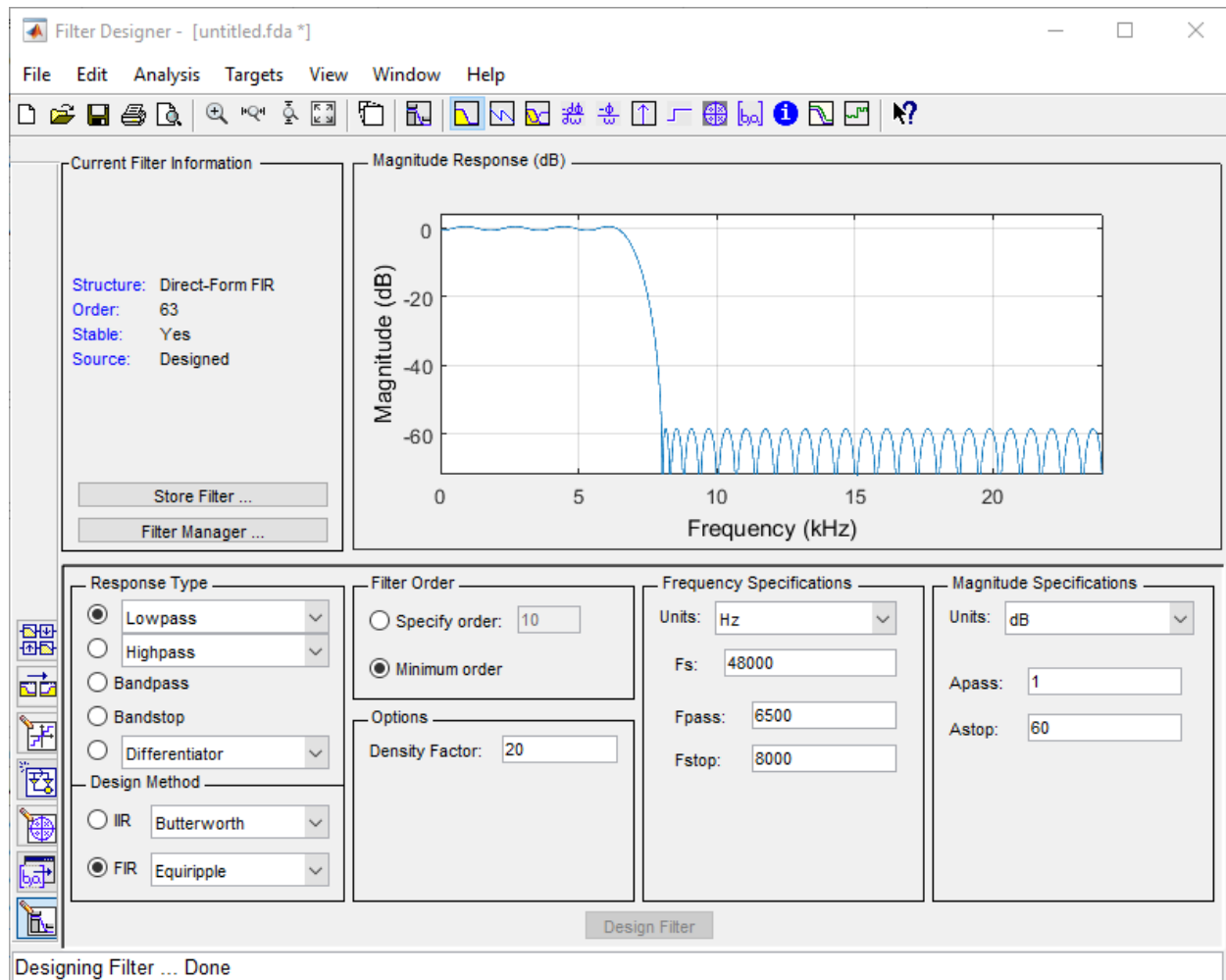
Fstop = 7500 Hz

Apass = 1 dB

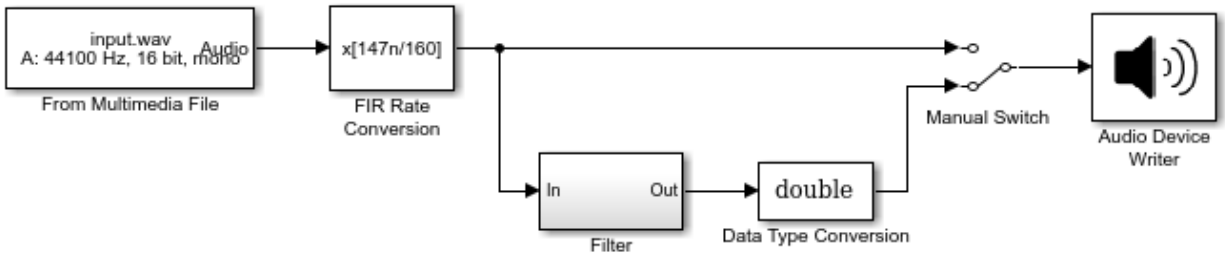
Astop = 60 dB

دلیل انتخاب فرکانس نمونه‌برداری ۴۸ کیلوهرتز، عملکرد Codec صدای موجود در برد DE2 با این فرکانس است (در حالت پیش‌فرض). 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 رنج 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. به جای این کار می‌توانید خروجی را ۳۲ بیتی با ۳۱ بیت اعشار تعریف کنید و در تنظیمات سیگنال‌های داخلی، Overflow Mode را روی Saturate تنظیم کنید (این کار را انجام ندهید). تفاوت این حالت با حالت Wrap چیست؟ سایر تنظیمات را تغییر ندهید. روی دکمه Apply کلیک کنید تا ضرایب فیلتر کوانتیزه شود. کوانتیزه کردن ضرایب فیلتر چه مزایا و معایبی دارد؟

برای آزمون 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 ۲ را به مدل اضافه کنید و پس از ذخیره‌سازی آن به صدای فیلتر شده و نشده گوش دهید و از عملکرد صحیح فیلتر اطمینان حاصل کنید. مدل Simulink را به همراه گزارش آپلود کنید.



شکل ۱ محیط گرافیکی ابزار Filter Designer.



شکل ۲ مدل Simulink.

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. همچنین این دو کد را به کمک Quartus برای برد DE2 سنتز کرده و میزان استفاده از منابع FPGA را در دو حالت مقایسه کنید. توجه کنید که پین‌های مشترکی را در هر دو سنتز استفاده نمایید.

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

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

آرایه inputs48k را به عنوان ورودی مورد نظر وارد نمایید. فیلتر را در مد کاملاً سریال قرار دهید. 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). فایل testbench را به دقت بخوانید و عملکرد آن را توضیح دهید. در محیط Modelsim از صحت عملکرد فیلتر مطمئن شوید (نباید testbench هشدار چاپ کند). برای انجام شبیه‌سازی می‌توانید از اسکریپت‌های تولید شده استفاده کنید. بدین منظور پس از باز کردن

Modelsim در بخش Transcript ابتدا با دستور cd وارد فولدري که فايل‌ها را در داخل آن Generate کرده‌ايد شويد (hdlsrc). سپس با دستور source ابتدا اسکريپت [filter_tb]_compile.do و سپس [filter_tb]_sim.do را اجرا کنيد. دقت کنيد اين شبیه‌سازی کمی زمانبر است (حداکثر ۳۰ دقیقه). بخشی از ابتدا و انتهای شکل موج ورودی/خروجی فیلتر را رسم کنيد (اعداد خوانا باشند).

۲- اضافه کردن دستور اختصاصی به پردازنده‌ی Nios II

در آزمایش دوم عملیات فیلترینگ 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، به طوری که پردازنده قادر بود اضافه کردن echo را به صورت بی‌درنگ^۱ انجام دهد. در این بخش از آزمایش قصد داریم فیلتر حذف نویز مورد استفاده در آزمایش اول را در دو حالت توسط پردازنده‌ی Nios اعمال کنیم: یکی به صورت کاملاً نرم‌افزاری و دیگری با اضافه کردن دستور اختصاصی و سخت افزار مربوط به آن به پردازنده‌ی Nios. گام‌های زیر را انجام دهید:

گام ۱

در محیط Quartus، از منوی Tools گزینه‌ی Nios II Software Build Tools for Eclipse را انتخاب کنيد. پروژه‌ی جدیدی را بر مبنای فايل sopcinfo. مربوط به سیستم DE2 Media Computer که اطلاعات سخت‌افزاری سیستم را برای SBT جهت تولید اتوماتیک درایورها و سیستم HAL^۲ ارائه می‌دهد؛ ایجاد کنيد. 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. سیستم DE2 Media Computer را روی FPGA پروگرام کنيد و با اجرای نرم‌افزار مشاهدات خود را از نظر سرعت اجرا و کیفیت خروجی گزارش کنيد. زمان انجام محاسبات را اندازه‌گیری و گزارش نمایید.

گام ۲

روند اصلی طراحی سخت‌افزاری سیستم در نرم‌افزار Qsys انجام می‌شود. در محیط Quartus، از منوی Tools ابزار Qsys را انتخاب کنيد. در سمت چپ پنجره‌ی Qsys کتابخانه‌ای از اجزای قابل استفاده در سیستم نشان داده

¹ Real Time

² Hardware Abstraction Layer

شده است و در سمت راست در هر Tab مشخصات مختلف سیستم قابل تنظیم است. Tab اصلی بخش System Contents می باشد که اجزای تشکیل دهنده سیستم و اتصالات بین آنها را نشان می دهد. توجه کنید که Qsys رفتار نرم افزاری سیستم از جمله مکان قرارگیری دستورات در حافظه را مشخص نمی کند.

فایل qsys. سیستم DE2 Media Computer را در یک فولدر کپی کرده و آن را با Qsys باز کنید. دستور اختصاصی به این سیستم اضافه خواهد شد. فیلتر سریال تولید شده توسط نرم افزار MATLAB را با کمترین تغییر به صورت یک دستور Multicycle با تعداد سیکل متغیر برای پردازنده Nios II تعریف کنید. نیازمندی های تعریف یک دستور Multicycle چیست؟ برای پاسخ به این سوال به اسلایدهای درس مراجعه کنید و در صورت نیاز به اطلاعات بیشتر مرجع [۱] را مطالعه نمایید. علاوه بر این نیازمندی ها، 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. توجه شود که در اسلایدهای درس روند کار با SOPC Builder گفته شده است. اما جهت آشنایی با نسخه جدید آن یعنی Qsys این روند را در Qsys انجام دهید (نه SOPC Builder).

گام ۳

پس از تولید کد سخت افزار کلی توسط Qsys، پروژه Quartus مربوط به DE2 Media Computer را باز کرده و دوباره سنتر نمایید. سیستم جدید را روی FPGA پروگرام کنید. به کمک Nios II SBT نرم افزار گام ۱ را به گونه ای 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.

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

نکات مهم:

(۱) بخش مهمی از این آزمایش بدون استفاده از سخت افزار DE2 قابل انجام است. مطالعه مرجع [۱] جهت انجام آزمایش ضروری است.

- (۲) نیازی به گزارش تمامی مراحل انجام آزمایش نیست. اما اهداف، کلیات، مقدار پارامترهای مختلف، مقایسه نتایج و سایر نکات مهم را حتماً در گزارش خود قید نمایید.
- (۳) پیروی از قالب خاصی در گزارش مد نظر نیست، اما ترجیحاً می‌توانید از قالب ارائه شده برای تکالیف کامپیوتری استفاده نمایید.
- (۴) آپلود فایل‌های شبیه‌سازی به همراه فایل گزارش ضروری است.

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

موفق باشید

۹۷/۰۸/۲۹