بسمه تعالی

درس طراحی سیستم‌های نهفته مبتنی بر FPGA

تکلیف کامپیوتری ۱: طراحی و پیاده‌سازی یک فیلتر FIR با اندازه متغیر به همراه درستی‌سنجی آن

دانشکدگان فنی دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

دکتر بیژن علیزاده

دستیار آموزشی:

علی شایان پور shayanpoorali66@gmail.com

علی قائمی ali.ghaemi@ut.ac.ir

نیمسال اول ۱۴۰۳-۱۴۰۲

موعد تحویل: ۲۴ مهر ۱۴۰۲

اهداف تمرین

- آشنایی با طراحی و توصیف سخت افزاری یک مدار دیجیتال
- آشنایی با روش های درستی سنجی مدارهای دیجیتال
- آشنایی با شبیه سازی توسط ابزار Modelsim جهت انجام درستی سنجی
- آشنایی با روش سنتز یک مدار دیجیتال با ابزار Quartus

مقدمه

۱. فیلترهای FIR

فیلترهای دیجیتال را می‌توان در دو دسته فیلترهای FIR (فیلترهای با طول محدود پاسخ ضربه) و فیلترهای IIR (فیلترهای با طول نامحدود پاسخ ضربه) طبقه‌بندی کرد. از مزایای فیلترهای FIR نسبت به IIR پایدار بودن حتمی آنها و داشتن پاسخ با فاز خطی است. فیلترهای با فاز خطی در سیستم های مخابرات دیجیتال، سیستم‌های پردازش صوت و تصویر، آنالیز طیفی خصوصا در سیستم‌هایی که در مقابل انحراف فاز غیرخطی تحمل‌پذیری ندارند؛ کاربرد فراوانی دارند. پاسخ ضربه یک فیلتر FIR با رابطه ی (۱) داده می‌شود:

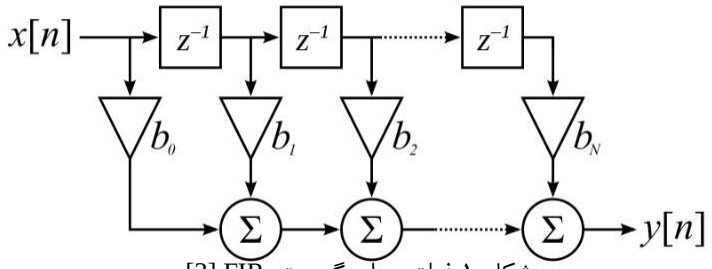
$$H(z) = \sum_{i=0}^{M} h[n].z^{-n} \quad (۱)$$

که در آن h(n) پاسخ ضربه محدود است. برای توصیف فیلتر FIR معمولا به جای درجه این فیلتر (M)، طول پاسخ ضربه ی آن (N = M + 1) بیان می‌گردد.

بدین ترتیب پاسخ فیلتر FIR با فرمول (۲) بیان می‌شود.

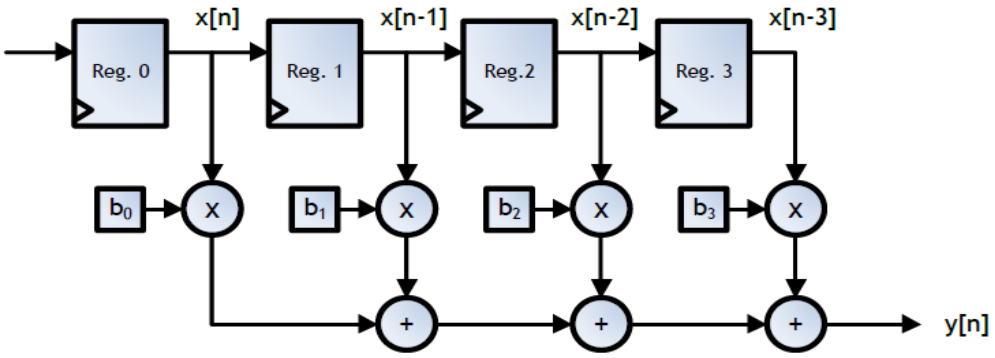
$$y[n] = \sum_{k=0}^{M} b[k].x[n-k] \quad (۲)$$

که در آن b[k] ها ضرایب فیلتر نامیده می‌شوند و برابر مقدار پاسخ ضربه در زمان‌های گسسته ۰ تا M هستند. در این تکلیف به طراحی فیلترهای FIR (تعیین ضرایب فیلتر) نمی‌پردازیم و هدف پیاده‌سازی سخت افزاری فیلتر FIR مطابق با فرمول (۲) است. شکل ۱ ساختار محاسباتی این فیلتر را نشان می‌دهد.



شکل ۱ فیلتر زمان گسسته FIR [3].

در هنگام پیاده‌سازی دیجیتال، به منظور ایجاد تأخیر واحد z^{-1} از رجیسترها استفاده می‌کنیم. به عنوان مثال شکل۲ پیاده‌سازی سخت‌افزاری فیلتر با درجه ۳ (طول ۴) را نشان می‌دهد.



شکل ۲ فیلتر FIR با طول ۴.

با شروع از سیکل صفر پس از ریست شدن رجیسترها، خروجی‌های مدار در هر سیکل مطابق با جدول ۱ خواهند بود. همانطور که مشاهده می‌شود به ازای هر ورودی یک مقدار خروجی تولید می‌شود.

جدول ۱ نمونه خروجی فیلتر با اندازه ۴.

Output	Register values				Clock cycle
	Reg. 3	Reg. 2	Reg. 1	Reg. 0	
0	0	0	0	0	.
b0.x[0]	0	0	0	x[0]	1
b0.x[1]+ b1.x[0]	0	0	x[0]	x[1]	2
b0.x[2]+ b1.x[1]+ b2.x[0]	0	x[0]	x[1]	x[2]	3
b0.x[3]+ b1.x[2]+ b2.x[1]+ b3.x[0]	x[0]	x[1]	x[2]	x[3]	4
b0.x[4]+ b1.x[3]+ b2.x[2]+ b3.x[1]	x[1]	x[2]	x[3]	x[4]	5

۲. نمایش ممیز ثابت (Fixed Point)

در این تمرین و در طول آزمایش‌هایی که در این درس خواهیم داشت، اعداد به صورت ممیز ثابت پیاده‌سازی خواهند شد. اعداد باینری ممیز ثابت، نمایشی دقیقا مشابه اعداد صحیح دارند و تنها فرق آنها وجود ممیز فرضی است. به عنوان مثال نمایش باینری ۱۱۰۱$_2$ را به صورت signed در نظر بگیرید.

$$1101_2 = -1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = -3$$

$$110.1_2 = -1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} = -1.5$$

همانطور که مشاهده می‌شود با تغییر مکان ممیز به سمت چپ، مقدار نصف عدد اولیه ایجاد می‌شود. پیاده‌سازی اول را با fix(4,0) و پیاده‌سازی دوم را با fix(4,1) نشان می‌دهیم. در این نمایش عدد اول تعداد کل بیت‌ها و عدد دوم مکان ممیز یا به عبارتی تعداد بیت‌های اعشاری را نشان می‌دهد. در هنگام پیاده‌سازی سخت‌افزاری دو عدد

۳- و۱.۵- به صورت ممیز ثابت، رجیسترها همان مقادیر ۱، ۱، ۰ و ۱ را خواهند داشت. در پیاده‌سازی اعداد به صورت fix<m, n> نکات زیر را مد نظر داشته باشید:

۱. **دقت اعداد**: دقت اعداد پیاده‌سازی شده به صورت ممیز ثابت برابر میزان ارزش بیت LSB خواهد بود که برابر 2^{-n} است.

۲. **بازه دینامیکی**: حداکثر بازه عدد قابل بیان برای اعداد بدون علامت $[0, 2^{m-n}-2^{-n}]$ و برای اعداد علامت‌دار به صورت $[-2^{m-n-1}, 2^{m-n-1}-2^{-n}]$ است.

در هنگام تصمیم‌گیری در مورد نمایش اعداد به صورت سخت‌افزاری، رعایت نکات فوق الزامی است. همچنین لازم به ذکر است پیاده‌سازی جمع و ضرب‌کننده ممیز ثابت مشابه جمع و ضرب‌کننده اعداد صحیح است. نکات زیر را مد نظر داشته باشید:

۱. خروجی جمع و تفریق دو عدد fix<m,n> به صورت fix<m+1,n> خواهد بود.

۲. خروجی ضرب دو عدد fix<m,n> به صورت fix<2m,2n> خواهد بود.

شرح تمرین

سیگنال صوتی دیجیتال از نمونه‌های صدا تشکیل شده است که با نمونه‌برداری سیگنال آنالوگ با فرکانس مشخصی (معمولا بین ۸ تا ۳۸۴ کیلوهرتز) تولید شده‌اند. این نمونه‌ها اعدادی در بازه (1,1-] هستند و بنابراین به فرمت signed fix<m,m-1> قابل نمایش هستند.

فایل‌های مورد نیاز در سایت درس بارگذاری شده است. فایل input.wav را گوش دهید. در این تمرین قصد داریم نویز فرکانس بالا در این فایل صوتی را با یک فیلتر FIR پایین‌گذر (lowpass) حذف کنیم. در محیط متلب می‌توانید با دستور زیر نمونه‌های فایل input.wav را استخراج کنید:

```
>> [inputs Fs] = audioread('input.wav');
```

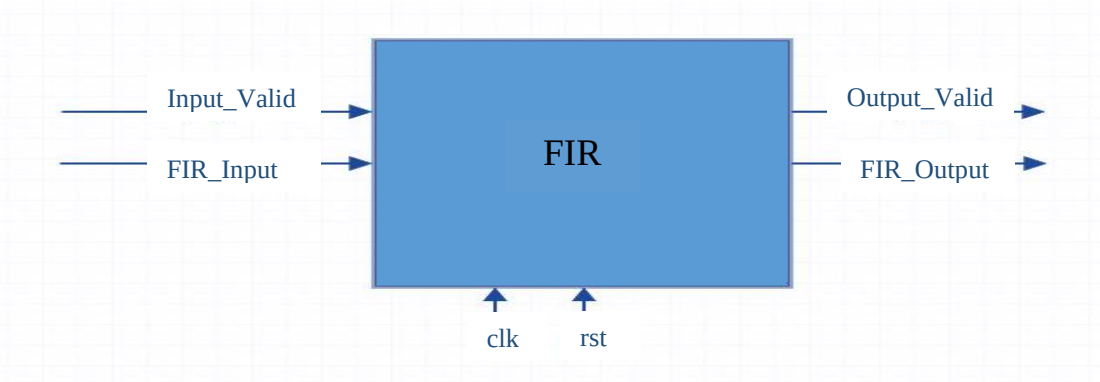
Fs نرخ نمونه‌برداری است که برابر ۴۴.۱ KHz است و inputs آرایه ای تک بعدی از نمونه‌های صدا است. نمونه‌های صوتی موجود در فایل input.wav در فایل inputs.txt با فرمت signed fix<16,15> موجود است.

همانطور که از رابطه (۲) برمی‌آید، یک فیلتر FIR با ضرایب آن مشخص می‌شود. ضرایب فیلتر مورد نظر در فایل coeffs.txt با فرمت signed fix<16,15> قرار دارد. در این آزمایش با استفاده از این ضرایب و نمونه‌های ورودی موجود در فایل inputs.txt انتظار می‌رود نمونه‌های خروجی دقیقا برابر نمونه‌های موجود در فایل outputs.txt

تولید شود. به همراه این فایلها دو فایل دیگر جهت راهنمایی ارائه شده است. فایل readme.txt را مطالعه نمایید. در ادامه مراحل زیر را به ترتیب انجام دهید:

۱. طراحی سخت‌افزاری

در سخت‌افزار شکل ۲ در هر کلاک یک خروجی آماده می‌شود. اما در عمل به دلیل آنکه معمولا درجه فیلترها بالا می‌باشد، به دلیل افزایش تعداد ضرب‌کننده و جمع‌کننده این پیاده‌سازی مناسب نخواهد بود. در این تمرین باید مداری طراحی کنید که محاسبات را مانند شکل ۲ با طول متغیر انجام دهد، اما فقط از یک ضرب‌کننده و جمع‌کننده استفاده کند. در این راستا می‌توانید یک ماشین حالت ساده بنویسید که این موضوع را در دل خود داشته باشد و یا اینکه دو بخش کنترلر و مسیرداده برای طراحی‌تان مشخص کنید. اگرچه وجود بخش‌های کنترلر و مسیرداده اجباری نیست، اما وجود معماری سلسه مراتبی و نام‌گذاری صحیح سیگنال‌ها به طوری که عملکرد آن را مشخص نماید، اهمیت به‌سزایی دارد. همچنین نحوه کدنویسی بسیار مهم می‌باشد و قسمت قابل توجهی از نمره را به خود اختصاص می‌دهد. روند کار بدین صورت است که Input_valid به مدت یک سیکل کلاک یک می‌شود. سپس مدار محاسبات خودرا شروع می‌کند. تا آماده شدن خروجی، مدار به کار خود ادامه می‌دهد. در صورت یک‌شدن مجدد این سیگنال، مدار از آن صرف‌نظر می‌کند. پس از آماده‌شدن خروجی، سیگنال ouput_valid به مدت یک سیکل کلاک یک می‌شود. عرض بیت‌های ورودی و خروجی به صورت پارامتری وارد می‌شوند. ضرایب فیلتر قسمتی از ویژگی یک فیلتر است، پس می‌توانید ضرایب را به صورت مستقیم درون فیلتر قرار دهید. شماتیک کلی فیلتر در شکل ۳ دیده می‌شود.



شکل ۳ نمای کلی فیلتر FIR

۲. توصیف به کمک Verilog

توجه فرمایید که کدهای مورد استفاده برای FPGA، بهتر است در یک فایل ساده و در قالب یک ماشین حالت باشند، زیرا نرم‌افزارها به راحتی حالت ماشین را شناسایی کرده و آن را بهینه‌سازی می‌کنند. کد سخت‌افزاری مربوط به هر ماژول با عملکرد مشخص باید در یک فایل نوشته شود. در این آزمایش یک نکته شدیدا دارای اهمیت است: توصیف‌های کد شما با وریلاگ باید کاملا خوانا باشد. بدین منظور:

- برای تمامی حالت‌های ماشین، اسم مشخص تعیین کنید.
- اتصال ورودی و خروجی ماژول‌ها ترتیبی نباشد و حتما با نام انجام شود.
- نام‌گذاری سیگنال‌ها مناسب باشد به طوری که بر اساس نام عملکرد آنها قابل پیش‌بینی باشد.

ورودی، خروجی و پارامترهای top_module فیلتر طراحی شده، باید تنها شامل موارد زیر باشد.(نام گذاری ورودی، خروجی و پارامترها دلخواه است).

```
module FIR (clk, reset, FIR_input, input_valid, FIR_output, output_valid);
    parameter    LENGTH = 8;
    parameter    WIDTH = 8;
    input        clk, reser, input_valid;
    output       output_valid;
    input        [WIDTH-1:0] FIR_input;
    output       [WIDTH-1:0] FIR_output;
```

لازم به ذکر است پارامتر WIDTH عرض ورودی‌ها، ضرایب و خروجی را تعیین می‌کند، اما عرض خروجی ضرب‌کننده و جمع‌کننده را باید به صورت صحیح انتخاب نمایید. همچنین لازم است توضیحات جامع در مورد عملکرد کنترلی مدار همراه با ترسیم ماشین حالت آن ارائه گردد.

نکته مهمی که باید در طراحی در نظر گرفته شود این است که این فیلتر قرار است که با بالاترین فرکانس ممکن کار کند. در آزمایش ۱ هر چه با فرکانس بیشتری بتوانید این فیلتر را بر روی FPGA راه‌اندازی کنید نمره امتیازی بیشتری به شما تعلق می‌گیرد. بنابراین تا حد امکان طراحی خود را pipeline کنید. بعبارت دیگر، لازم است که شما تشخیص دهید کدام مسیر ترکیبی (combinational) بیشترین تاخیر را دارد. نکته مهم رعایت timing مدار هنگام pipeline کردن و اضافه‌کردن رجیسترها می‌باشد. (راهنمایی: می‌توانید ضرب‌کننده را با دو مرحله pipeline طراحی کنید.) روش دیگر برای رسیدن به فرکانس بیشتر تغییر در تنظیمات سنتز و نیز جایابی و مسیریابی (Place and Route) می‌باشد. روند دستیابی به فرکانس بالاتر را گزارش کنید.

به علاوه، به منظور گزارش مقدار فرکانس بیشینه فیلتر طراحی شده، پس از سنتز توسط ابزارQuartus ، از بخش TimeQuest Timing Analyzer > Slow Model > Fmax Summary استفاده نمایید .

۳. درستی‌سنجی با روش شبیه‌سازی

یک تست‌بنچ (testbench) برای فیلتر خود بنویسید که درستی عملکرد مدار خود را با فایل‌های داده شده بررسی کند. فایل outputs.txt شامل نمونه‌های صحیح خروجی با فرمت <fix<38, 30 است. تست‌بنچ در محیط Modelsim اجرا شده و نتایج کلی خروجی و شکل‌موج‌های مهم را در گزارش خود قرار دهید. زمانی فیلتر شما صحیح کار می‌کند که دقیقا خروجی‌های مد نظر موجود در فایل outputs.txt را تولید نماید.

۴. درستی‌سنجی با روش Assertion

با استفاده از assertion های SystemVerilog، درستی عملکرد مدار را برای اندازه ۵ و عرض بیت ۱۶ با استفاده از حداقل ۵ assertion تحقیق کنید. به منظور بررسی assertion ها در مدار، از ابزار شبیه‌سازی QuestaSim استفاده کرده و این موارد را در گزارش خود ذکر کنید. انتخاب assertion های مناسب اهمیت دارد.
برای این بخش استفاده از assertion های متنوع اهمیت دارد و به استفاده مناسب از ویژگی های assertion و پیاده سازی های خلاقانه تا ۵ درصد نمره امتیازی تعلق خواهد گرفت.

۵. سنتز

فیلتر طراحی شده را برای اندازه‌های ۵۰ و ۱۰۰ و عرض بیت ۸ و ۱۶ سنتز کرده تعداد فلیپ فلاپ‌ها و المان‌های منطقی استفاده شده در هر حالت را در قالب یک جدول گزارش کنید و مقایسه‌ای بین آن‌ها انجام دهید.
به منظور سنتز فیلتر طراحی شده در ابزار Quartus از مشخصات زیر استفاده نمایید.

Device family: Cyclone_II
Available Devices: EP2C35F672C6

نکات مهم:

۱- لطفا دقت نمایید گزارش شما جامع و مانع باشد. عدم گزارش مناسب با کسر نمره مواجه می‌شود.

۲- تایپ کردن گزارش ضروری نیست اما خوانا بودن آن ضروری است.

۳- مراحل ۱ تا ۵ را به صورت جداگانه و به ترتیب گزارش نمایید.

۴- بخش‌های مهم کد را در گزارش بیاورید.

۵- بارگذاری فایلهای شبیه‌سازی به همراه فایل گزارش ضروری است.

۶- سؤالات خود را در سایت و در فروم مربوط به این تکلیف مطرح نمایید.

۷- دقت فرمایید موعد تحویل با تأخیر تا یک هفته پس از تاریخ ذکر شده با احتساب هر روز ۲ درصد کسر نمره می‌باشد و بعد از این تاریخ به هیچ عنوان تمرین تحویل گرفته نخواهد شد.

۸- **این تمرین به هیچ عنوان نباید به صورت گروهی انجام شود.**

۹- این تمرین مبنای کار آزمایش اول است و در جلسه دوم آزمایش اول به آن نیاز خواهید داشت.

مراجع

[1] L. Wanhammer, "*DSP Integrated Circuits*", Academic press,
New York, 1999. [2] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, "*Discrete-Time Signal Processing", 2nd ed* . Upper Saddle River,
NJ: Prentice Hall, 1999.
[3] Wikipedia, The Free Encyclopedia, "*Finite impulse response*".

موفق باشید.