



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

پردیس دانشکده‌های فنی دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

دکتر بیژن علیزاده

دستیار آموزشی:

نگار آقاپور n.aghapour.s@gmail.com

پارسا کاویانی 1kavianiparsa@gmail.com

نیمسال اول ۱۴۰۱-۱۴۰۰

موعد تحویل: ۲۷ مهر ۱۴۰۰

اهداف تمرین

- ✓ آشنایی با طراحی و توصیف سخت‌افزاری یک مدار دیجیتال
- ✓ آشنایی با روش‌های درستی‌سنجی مدارهای دیجیتال
- ✓ آشنایی با شبیه‌سازی توسط ابزار Modelsim جهت انجام درستی‌سنجی
- ✓ آشنایی با Assertion
- ✓ آشنایی با روش سنتز یک مدار دیجیتال با ابزار Quartus

۱. فیلترهای FIR

فیلترهای دیجیتال را می‌توان در دو دسته فیلترهای FIR^1 (فیلترهای با طول محدود پاسخ ضربه) و فیلترهای IIR^2 (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 سیستم‌های مخابرات دیجیتال، سیستم‌های پردازش صوت و تصویر، آنالیز طیفی خصوصاً در سیستم‌هایی که در مقابل انحراف فاز غیرخطی تحمل‌پذیری ندارند؛ کاربرد فراوانی دارند. پاسخ ضربه یک فیلتر FIR با رابطه ی (۱) داده می‌شود:

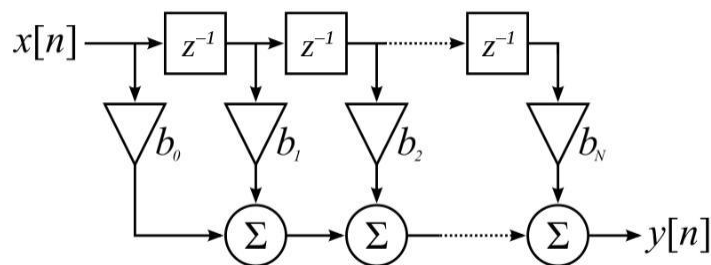
$$H(z) = \sum_{i=0}^M h[n].z^{-n} \quad (۱)$$

که در آن $h(n)$ پاسخ ضربه محدود است. برای توصیف فیلتر FIR معمولاً 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.

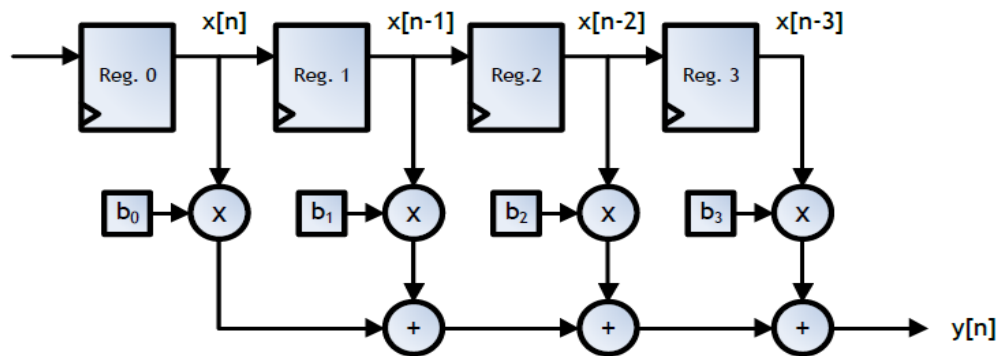
$$y[n] = \sum_{k=0}^M b[k].x[n - k] \quad (۲)$$

که در آن $b[k]$ 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 شکل ۲ پیاده‌سازی سخت‌افزاری فیلتر با درجه ۳ و (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

خروجی	مقدار رجیسترها				شماره سیکل کلاک
	Reg. 3	Reg. 2	Reg. 1	Reg. 0	
0	0	0	0	0	۰
$b_0.x[0]$	0	0	0	$x[0]$	۱
$b_0.x[1] + b_1.x[0]$	0	0	$x[0]$	$x[1]$	۲
$b_0.x[2] + b_1.x[1] + b_2.x[0]$	0	$x[0]$	$x[1]$	$x[2]$	۳
$b_0.x[3] + b_1.x[2] + b_2.x[1] + b_3.x[0]$	$x[0]$	$x[1]$	$x[2]$	$x[3]$	۴
$b_0.x[4] + b_1.x[3] + b_2.x[2] + b_3.x[1]$	$x[1]$	$x[2]$	$x[3]$	$x[4]$	۵

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 دقیقاً مشابه اعداد صحیح دارند و تنها فرق آنها وجود ممیز فرضی است. به عنوان مثال نمایش باینری ۱۱۰۱۲ 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 $\text{fix}(4,0)$ و پیاده‌سازی دوم را با $\text{fix}(4,1)$ 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- و ۱،۵- به صورت ممیز ثابت، رجیسترها همان مقادیر ۱، ۰ و ۱ را خواهند داشت. در پیاده‌سازی اعداد به صورت $\text{fix}\langle m, n \rangle$ 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 $\text{fix}\langle m, n \rangle$ به صورت $\text{fix}\langle m+1, n \rangle$ خواهد بود.

۲. خروجی ضرب دو عدد $\text{fix}\langle m, n \rangle$ به صورت $\text{fix}\langle 2m, 2n \rangle$ خواهد بود.

شرح تمرین

سیگنال صوتی دیجیتالی از نمونه‌های صدا تشکیل شده است که با نمونه‌برداری سیگنال آنالوگ با فرکانس مشخصی (معمولاً بین ۸ تا ۳۸۴ کیلوهرتز) تولید شده‌اند. این نمونه‌ها اعدادی در بازه $[-1, 1]$ هستند و بنابراین به فرمت $\text{signed fix}\langle m, m-1 \rangle$ 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');
```

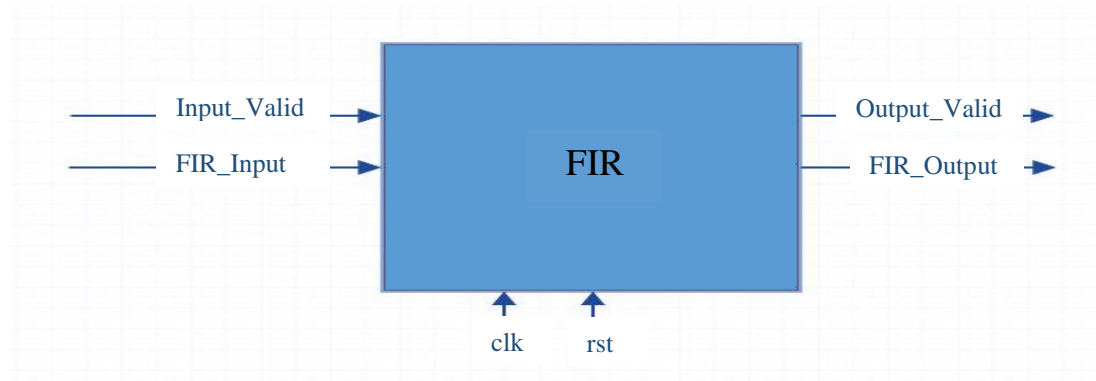
Fs نرخ نمونه‌برداری است که برابر ۴۴,۱ KHz است و inputs آرایه‌ای تک بعدی از نمونه‌های صدا است. نمونه‌های صوتی موجود در فایل `input.wav` در فایل `inputs.txt` با فرمت $\text{signed fix}\langle 16, 15 \rangle$ موجود است.

همانطور که از رابطه (۲) برمی‌آید، یک فیلتر FIR با ضرایب آن مشخص می‌شود. ضرایب فیلتر مورد نظر در فایل `coeffs.txt` با فرمت $\text{signed fix}\langle 16, 15 \rangle$ قرار دارد. در این آزمایش با استفاده از این ضرایب و نمونه‌های ورودی

موجود در فایل inputs.txt انتظار می‌رود نمونه‌های خروجی دقیقاً 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 معمولاً 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 مسیره‌داده برای طراحیتان به‌دست آورید. اگرچه وجود بخش‌های کنترلر و مسیره‌داده 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 خود را شروع می‌کند. تا آماده شدن خروجی، مدار به کار خود ادامه می‌دهد. در صورت یک شدن مجدد این سیگنال مدار از آن صرف نظر می‌کند. پس از آماده شدن خروجی، سیگنال output_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 می‌تواند ضرایب به صورت مستقیم درون فیلتر باشد. 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 شدیداً دارای اهمیت است: توصیف‌های کد شما با وریلاگ باید کاملاً خوانا باشد. بدین منظور:

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

top_module شما باید پارامتر، ورودی و خروجی‌های مشابه داشته باشد (نامگذاری ورودی و خروجی و پارامترها دلخواه است).

```
module FIR (clk, reset, FIR_input, input_valid, FIR_output,
output_valid);
    parameter IN_WIDTH = 16;
    parameter OUT_WIDTH;
    parameter LENGTH;
    input      clk, reset, input_valid;
    output     output_valid;
    input      [IN_WIDTH -1:0] FIR_input;
    output     [OUT_WIDTH -1:0] FIR_output;
```

لازم به ذکر است پارامتر IN_WIDTH عرض ورودی‌ها، OUT_WIDTH عرض خروجی و LENGTH، تعداد ضرایب را تعیین می‌کنند. عرض ورودی ۱۶ بیت است ولی مقدار سایر پارامترها 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 بهتری به شما تعلق می‌گیرد. بنابراین تا حد امکان طراحی خود را pipeline کنید. یعنی لازم است شما تشخیص دهید کدام مسیر ترکیبی (combinational) بیشترین تاخیر را دارد. نکته مهم رعایت timing مدار هنگام pipeline کردن و اضافه کردن 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 جاییابی و مسیریابی (Place and Route) می باشد. روند دستیابی به فرکانس بالاتر باید گزارش شود.

۳. درستی سنجی با روش شبیه سازی

یک تست بنچ (test bench) برای فیلتر خود بنویسید و درستی عملکرد مدار خود را با فایل های داده شده بررسی کنید. فایل outputs.txt شامل نمونه های صحیح خروجی با فرمت `fix<38, 30>` است. تست بنچ در محیط Modelsim اجرا شده و تا حد ممکن کامل و جامع باشد. زمانی فیلتر شما صحیح کار می کند که دقیقاً خروجی های مد نظر موجود در فایل outputs.txt را تولید نماید. همچنین یک فایل MATLAB به نام play_output.m به شما داده شده که می توانید از آن برای شنیدن صدای خروجی استفاده کنید.

۴. درستی سنجی با روش Assertion (امتیازی)

با استفاده از assertion های SystemVerilog، درستی عملکرد مدار را برای اندازه ۶۴ و عرض بیت ۱۶ با استفاده از حداقل ۵ assertion تحقیق کنید. به منظور بررسی assertion ها در مدار، از ابزار شبیه سازی QuestaSim استفاده کرده و این موارد را در گزارش خود ذکر کنید. انتخاب assertion های مناسب اهمیت دارد.

۵. سنتز

فیلتر طراحی شده را برای اندازه های ۵۰ و ۱۰۰ و عرض بیت ۸ و ۱۶ سنتز کنید. برای این کار می توانید از ابزار سنتز Quartus استفاده کنید. سپس تعداد فلیپ فلاپ ها و المان های منطقی استفاده شده در هر حالت را در قالب یک جدول گزارش کنید و مقایسه ای بین آن ها انجام دهید.

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 شوند.

مراجع

- [1] L. Wanhammer, “*DSP Integrated Circuits*”, Academic press, New York, 1999.
- [2] A. V. Oppenheim, R. W. Schaffer, 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*”.
- [4] [Quartus Advanced Fitter Setting](#)

موفق باشید.