

درس طراحی سیستم‌های نهفته مبتنی بر FPGA

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

پردیس دانشکده‌های فنی دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

دکتر بیژن علیزاده

دستیاران آموزشی:

siamackbm@yahoo.com

سیامک بیگ محمدی

پاییز ۱۳۹۷

موعد تحویل: ۱۳۹۷/۰۷/۲۰

اهداف تمرین:

- (۱) آشنایی با طراحی و توصیف سخت افزاری یک مدار دیجیتال
- (۲) آشنایی با روش‌های درستی‌سنجی مدارهای دیجیتال
- (۳) آشنایی با روش سنتز یک مدار دیجیتال با ابزار Quartus
- (۴) آشنایی با شبیه‌سازی توسط ابزار Modelsim جهت انجام درستی‌سنجی

مقدمه

۱. فیلترهای FIR

فیلترهای دیجیتال را می‌توان در دو دسته‌ی فیلترهای ^۱FIR (فیلترهای با طول محدود پاسخ ضربه) و فیلترهای ^۲IIR (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 و خصوصاً در سیستم‌هایی که در مقابل انحراف فاز غیر خطی تحمل‌پذیری ندارند؛ کاربرد فراوانی دارند. پاسخ ضربه‌ی یک فیلتر FIR با رابطه‌ی (۱) داده می‌شود:

$$H(z) = \sum_{i=0}^M h[n].z^{-n} \quad (1)$$

¹ Finite Impulse Response

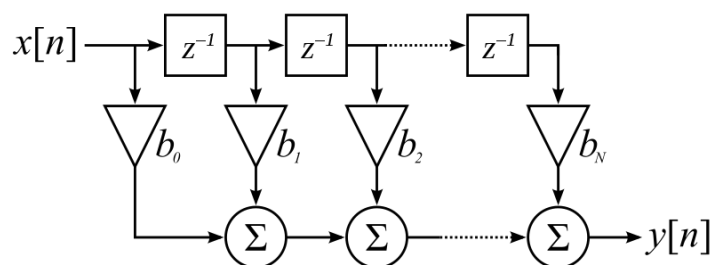
² Infinite Impulse Response

که در آن $h(n)$ پاسخ ضربه (محدود) است. برای توصیف فیلتر FIR معمولاً به جای درجه‌ی این فیلتر (M)، طول پاسخ ضربه‌ی آن ($N = M + 1$) بیان می‌گردد [1].

بدین ترتیب پاسخ فیلتر FIR با فرمول (۲) بیان می‌شود [2]:

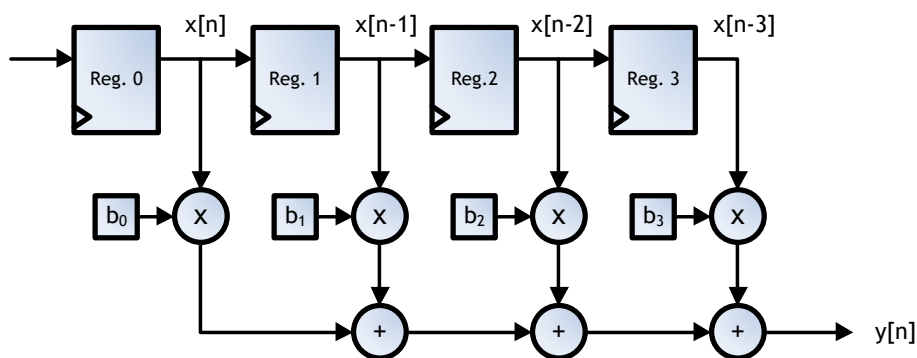
$$y[n] = \sum_{k=0}^M b[k].x[n-k] \quad (2)$$

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



شکل ۱ فیلتر زمان گسسته‌ی 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 درجه‌ی 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 اندازه‌ی 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
$b_0.x[3] + b_1.x[2] + b_2.x[1] + b_4.x[0]$	$x[0]$	$x[1]$	$x[2]$	$x[3]$	۴
$b_0.x[4] + b_1.x[3] + b_2.x[2] + b_4.x[1]$	$x[1]$	$x[2]$	$x[3]$	$x[4]$	۵

۲. نمایش ممیز ثابت (Fixed Point)

در این تمرین و در طول آزمایش‌هایی که در درس سیستم‌های نهفته 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 1101_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 $\text{fix}(4,0)$ 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 -3 و -1.5 به صورت ممیز ثابت، رجیسترها همان مقادیر 1 ، 0 ، 1 و 1 را خواهند داشت. در پیاده‌سازی اعداد به صورت $\text{fix}\langle m,n \rangle$ نکات زیر را مد نظر داشته باشید:

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

۲. رنج دینامیکی: حداکثر بازه عدد قابل بیان در صورت unsigned بودن برابر $[0, 2^{m-n} - 2^{-n}]$ و در صورت signed بودن برابر $[-2^{m-1-n}, 2^{m-1-n} - 2^{-n}]$ است.

در هنگام تصمیم‌گیری در مورد نمایش اعداد به صورت سخت‌افزاری، رعایت نکات فوق لازم است. همچنین لازم به ذکر است پیاده‌سازی جمع و ضرب کننده ممیز ثابت مشابه جمع و ضرب کننده 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 دیجیتال 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 44.1 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 کنید.

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 شکل ۲ در هر سیکل کلاک یک خروجی تولید می‌شود. در پیاده‌سازی فیلترهای FIR به دلیل آنکه معمولاً درجه‌ی آن‌ها زیاد است، از تعداد واحدهای ضرب و جمع‌کننده محدودی اما در چند سیکل استفاده می‌شود. در این تکلیف باید مداری طراحی کنید که بتواند محاسبات را مطابق با شکل ۲ انجام دهد (با طول فیلتر متغیر)، اما لازم است در این طرح فقط از یک ضرب‌کننده و یک جمع‌کننده استفاده نمایید. نیازی نیست طرح شما به دو بخش مسیر داده و کنترلر تقسیم شده باشد اما وجود معماری و سلسله مراتب مشخص و نام‌گذاری صحیح سیگنال‌ها به طوری که عملکرد آن‌ها را مشخص نماید اهمیت بالایی دارد. همچنین استایل کد زنی صحیح بسیار حائز اهمیت است و بخش بزرگی از نمره ره به خود اختصاص داده است. روند کار بدین صورت است که هنگامی که داده در ورودی مدار قرار می‌گیرد سیگنال کنترلی `inputValid` (ورودی) به مدت یک سیکل کلاک به مقدار ۱ ست می‌شود. سپس مدار محاسبات خود را شروع می‌کند و هنگام آماده شدن خروجی (پس از چند سیکل کلاک)، سیگنال کنترلی `outputValid` (خروجی) را به مدت یک سیکل کلاک ۱ می‌کند. در هنگام انجام محاسبات، در صورت ۱ شدن `inputValid` مدار از آن صرف‌نظر می‌کند. در این بخش معماری مناسبی برای این مدار طراحی کنید و آن را در قالب ماشین حالت و عملیاتی که در هر حالت انجام می‌شود، گزارش نمایید. مازول مموری را که ضرایب فیلتر در آن قرار دارد، به صورت جداگانه طراحی کنید به طوری که تأخیر خواندن از آن ۱ سیکل باشد.

۲. (توصیف به کمک Verilog) توجه فرمایید که کدهایی که برای FPGA مورد استفاده قرار می‌گیرند بهتر است که 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. لازم به ذکر است کدهایی که مسیر داده و کنترلر جدا دارند علاوه بر طراحی سخت‌تر، عیب‌یابی سخت‌تری نیز دارند. به همین دلیل سیستم شما ترجیحاً نباید مسیر داده و کنترلر جدا داشته باشد. توجه نمایید که منظور از جدا نکردن مسیر داده و کنترلر به معنی عدم توصیف استاندارد ماشین حالت مدار نیست. کد سخت‌افزاری مربوط به هر مازول 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:

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

تاپ ماژول شما باید پارامتر ورودی خروجی های مشابه داشته باشد (نامگذاری 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, reset, 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 کنند (در آزمایش یک) و هر چه با فرکانس بیشتری بتوانید این فیلتر را بر روی FPGA راه اندازی کنید نمره امتیازی بهتری به شما تعلق می‌گیرد. بنابراین تا حد امکان طراحی خود را Pipeline کنید. روش دیگر برای رسیدن به فرکانس بیشتر تغییر در تنظیمات سنتز و نیز جایابی و مسیریابی (Place and Route) است. طبعاً روند دستیابی به فرکانس بالاتر باید گزارش شود.

۳. **درستی‌سنجی با روش شبیه‌سازی)** یک تست‌بنچ (Testbench) برای فیلتر خود بنویسید و درستی عملکرد مدار خود را با فایل‌های داده شده بررسی کنید. فایل outputs.txt شامل نمونه‌های صحیح خروجی با فرمت fix<38, 30> است. تست‌بنچ باید در محیط Modelsim اجرا شود. تست‌بنچ مورد نظر باید تا حد ممکن کامل و جامع باشد. زمانی فیلتر شما صحیح کار می‌کند که دقیقاً خروجی‌های مد نظر در فایل outputs.txt را تولید نماید.

۴. **درستی‌سنجی با روش Assertion)** با استفاده از assertion‌های System Verilog، درستی عملکرد مدار را برای اندازه‌ی ۵ و عرض بیت ۸ با استفاده از حداقل ۵ assertion تحقیق کنید. به منظور بررسی assertion‌ها در مدار از ابزار شبیه‌سازی QuestaSim استفاده کرده و این موارد را در گزارش خود ذکر کنید. انتخاب assertion‌های مناسب اهمیت دارد.

۵. (سنتز) فیلتر طراحی شده را برای اندازه‌های ۵۰ و ۱۰۰ و عرض بیت ۸ و ۱۶ سنتز کرده تعداد فلیپ‌فلاپ‌ها و المان‌های منطقی استفاده شده در هر حالت را در قالب یک جدول گزارش کنید و مقایسه‌ای بین آن‌ها انجام دهید.

نکات مهم:

- (۱) لطفاً دقت نمایید گزارش شما باید 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) آپلود 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 ۰.۲٪ 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 از 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