



درس طراحی در سطح سیستم

تکلیف کامپیوتری ۲: طراحی و سنتز سطح بالا در MATLAB

دانشکده فنی دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

دکتر بیژن علیزاده

نیم سال دوم سال تحصیلی ۱۴۰۳-۱۴۰۲

نگارش: امیرحسین یوسفوند (@usfvnd, amirhyousefvand@ut.ac.ir)

مقدمه:

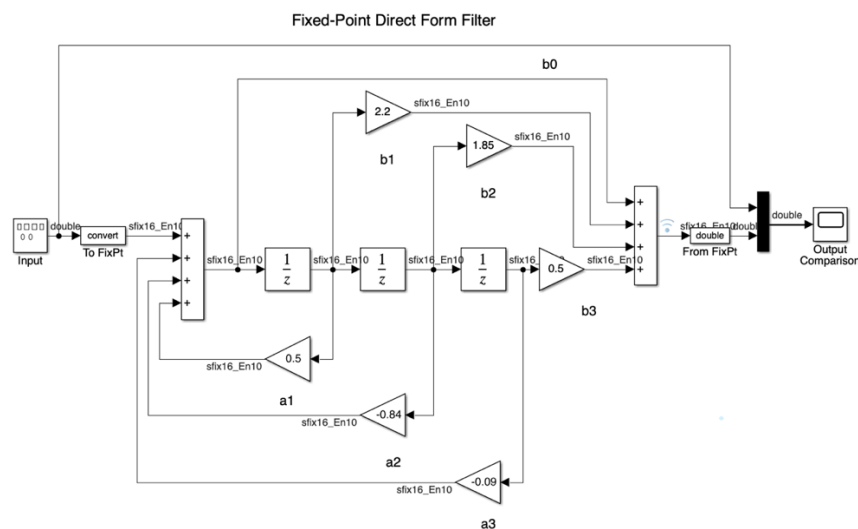
در این تمرین کامپیوتری با ابزارهای مختلف MATLAB برای سنتز در سطح بالا آشنا می‌شویم که ادامه توضیحات مربوط به هر بخش آمده است:

- **Fixed-Point Tool:** این ابزار در محیط MATLAB به شما کمک می‌کند تا مدل‌های خود را برای پردازش با دقت‌های ثابت برنامه‌ریزی کنید. با استفاده از این ابزار، می‌توانید مدل‌های خود را از ممیز شناور به ممیز ثابت تبدیل کرده و تنظیمات مختلف را برای دقت اعمال کنید.
- **HDL Coder:** این ابزار به شما این امکان را می‌دهد که الگوریتم‌ها و مدل‌های MATLAB و Simulink خود را به زبان‌های سخت‌افزاری مانند Verilog و VHDL تبدیل کنید. با استفاده از HDL Coder، می‌توانید سیستم‌های سخت‌افزاری خود را مدل‌سازی کرده و سپس آن‌ها را سنتز و شبیه‌سازی کنید.

توجه: مواردی که باید در گزارش قید شوند با شماره و رنگ **قرمز** در صورت پروژه نوشته شده‌اند.

۱- استفاده از ابزار Fixed Point

در این بخش نحوه استفاده از ابزار fixed point را برای مقایسه دقت میان انواع داده‌های floating point و fixed point در مدل شکل ۱ فرا می‌گیریم. شکل ۱ نشان دهنده یک فیلتر دیجیتال می‌باشد که از المان‌هایی مانند sum, gain, delay تشکیل شده است.



شکل ۱- فیلتر دیجیتال



۱- دو مورد از مزایا و دو مورد از معایب تبدیل floating point به fixed point را بیان کنید.

۱-۱- بررسی فیلتر دیجیتالی

در این بخش به بررسی عملکرد مداری که قرار است در آن تبدیل floating point به fixed point را انجام دهیم می‌پردازیم. برای این کار گام‌های زیر را انجام دهید:

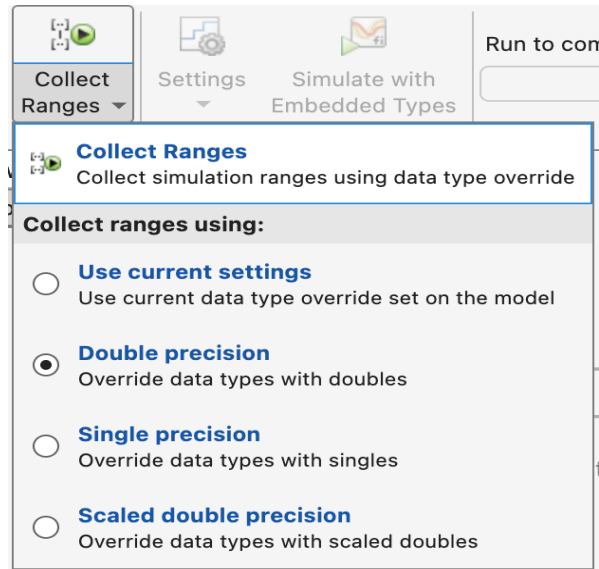
- فایل "fxpdemo_direct_form2.slx" که در اختیار شما قرار گرفته را باز کنید.
- ۲- شبیه‌سازی را با زدن گزینه Run انجام دهید و تصویر output comparison را در گزارش بیاورید. شکل موج خروجی را با توجه به شکل موج ورودی و مدار داده شده توجیه کنید.
- کد زیر را که تولید کننده ۶ ورودی با دامنه‌های مختلف برای فیلتر دیجیتال است را در command window اجرا کنید:

```
simIn(1:6) = Simulink.SimulationInput('fxpdemo_direct_form2');  
  
simIn(1) = simIn(1).setBlockParameter('fxpdemo_direct_form2/Input',...  
    'Amplitude','0.001');  
simIn(2) = simIn(2).setBlockParameter('fxpdemo_direct_form2/Input',...  
    'Amplitude','0.01');  
simIn(3) = simIn(3).setBlockParameter('fxpdemo_direct_form2/Input',...  
    'Amplitude','0.1');  
simIn(4) = simIn(4).setBlockParameter('fxpdemo_direct_form2/Input',...  
    'Amplitude','1');  
simIn(5) = simIn(5).setBlockParameter('fxpdemo_direct_form2/Input',...  
    'Amplitude','10');  
simIn(6) = simIn(6).setBlockParameter('fxpdemo_direct_form2/Input',...  
    'Amplitude','100');  
Simulink.sdi.markSignalForStreaming('fxpdemo_direct_form2/Sum1',1,'on');
```

۲-۱- تغییر مبنای فیلتر دیجیتال به Fixed Point و مقایسه با مبنای Floating Point

حال با استفاده از fixed point tool مبنای فیلتر دیجیتال را به fixed point تغییر می‌دهیم و مقایسه‌ای میان این دو مبنای fixed point و floating point انجام می‌دهیم. برای این کار مراحل زیر را طی کنید:

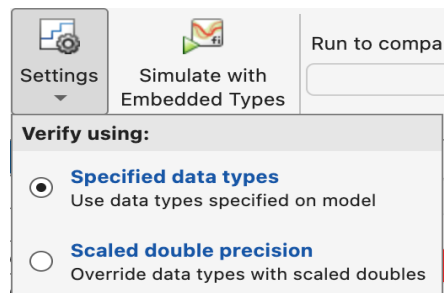
- بر روی صفحه کلیک راست کرده و گزینه fixed point tool را انتخاب کنید.
- از میان workflow های موجود Range Collection را انتخاب کنید.
- در بخش simulation input گزینه simIn را انتخاب کنید تا ورودی‌ها با دامنه‌های مختلف که در بخش ۱-۱ به مدار دادیم بعنوان ورودی‌های شبیه‌سازی قرار گیرند.
- در بخش Signal Tolerances مقدار ۰.۱۵ را در فیلد Relative Tolerance وارد نمایید. این مقدار نشان دهنده حداکثر خطای نسبی یک سیگنال است تا قابل قبول باشد.
- همانند شکل ۲ تنظیمات collect ranges را روی double precision قرار دهید. زمانی که شما این کار را انجام می‌دهید fixed point tool نوع داده‌های درون مدل شما را به double تغییر می‌دهد، مدل را شبیه‌سازی می‌کند و بیشینه و کمینه مقادیر هر یک از المان‌های مدل را محاسبه می‌کند.



شکل ۲- تنظیمات collect ranges

- سپس collect ranges را اجرا کنید. با این کار شبیه‌سازی با روش floating point انجام می‌شود.
- در workflow browser وضعیت underflow و overflow به ازای ۶ ورودی مختلف نمایش داده شده است.

۳- وضعیت underflow و overflow را به ازای هر ۶ ورودی گزارش و تحلیل کنید و بگویید چه رابطه‌ای میان رخ دادن underflow و overflow با دامنه سیگنال ورودی وجود دارد؟
در بخش setting گزینه specified data types را انتخاب کنید (شکل ۳).



شکل ۳- تنظیمات collect ranges

- بر روی گزینه simulate with embedded types کلیک کنید. با این کار شبیه‌سازی با روش fixed point انجام می‌شود و شما می‌توانید نتایج بدست آمده از دو روش را مقایسه کنید.
 - همانطور که دیده می‌شود embedded run تنها برای یکی از ورودی‌ها به ازای tolerance relative که قرار دادیم توانست خروجی مطلوب تولید کند.
 - بر روی خروجی مطلوب کلیک راست کنید و open SDI را انتخاب کنید
- ۴- تصویر صفحه نمایش داده شده را در گزارش بیاورید و نمودارهای نشان داده شده را تحلیل کنید.
- ۵- مرحله ۴ را برای EmbeddeRun_senarion_1 و EmbeddeRun_senarion_6 تکرار کنید.



۲- استفاده از ابزار HDL coder

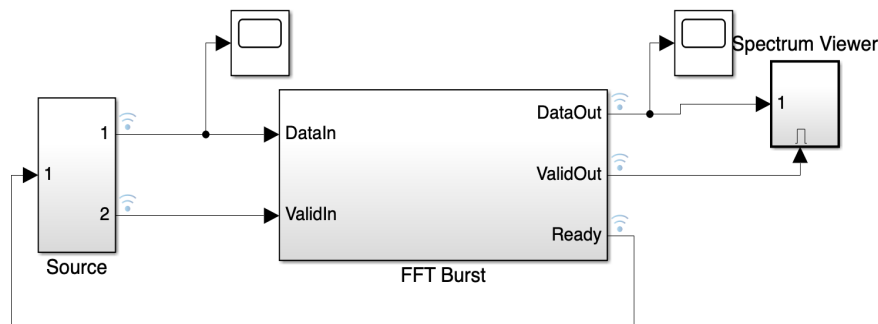
در این بخش ابتدا با مدل FFT داده شده آشنا می‌شویم. سپس به سنتز و تولید کد Verilog ماژول FFT می‌پردازیم. در گام آخر نیز به شبیه‌سازی کدهای تولید شده در HDL coder در MATLAB می‌پردازیم.

۲-۱- بررسی مدل FFT

FFT به معنای تبدیل فوریه سریع است. FFT یک الگوریتم کارآمد است که برای محاسبه تبدیل فوریه گسسته (DFT) یک دنباله یا سیگنال دامنه زمان استفاده می‌شود. تبدیل فوریه گسسته (DFT)، یک تبدیل ریاضی است که سیگنال را از دامنه زمان به دامنه فرکانس تبدیل می‌کند.

الگوریتم FFT با کاهش قابل توجه پیچیدگی محاسباتی، صنعت پردازش سیگنال دیجیتال را متحول کرد. این الگوریتم امکان محاسبه تبدیل فوریه گسسته را به صورت کارآمد فراهم می‌کند و همین امر باعث می‌شود برای برنامه‌های real-time و تجزیه و تحلیل داده‌های بزرگ مناسب و کارآمد باشد.

- فایل با نام "FFTHDLOptimizedExample_Burst.slx" که حاوی محاسبه کننده تبدیل فوریه سریع است را اجرا کنید (شکل ۴).



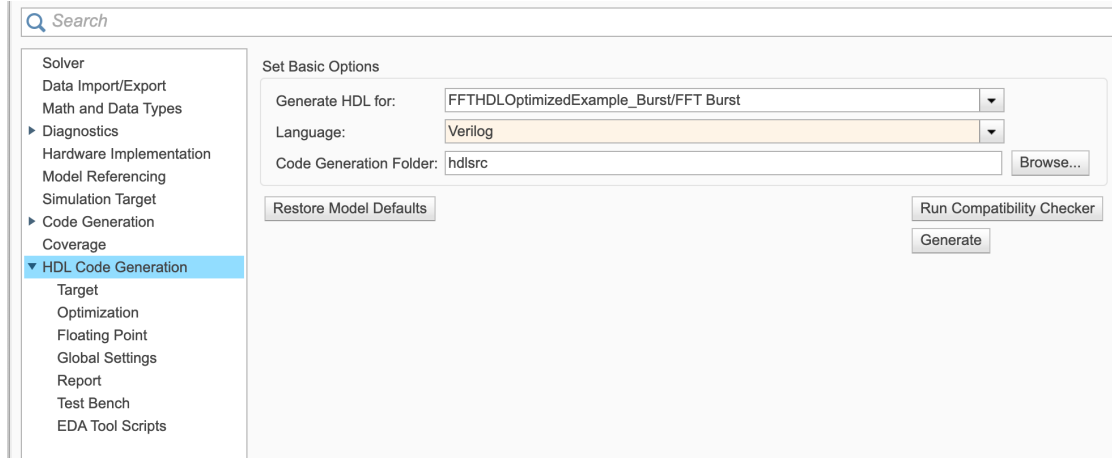
شکل ۴- ماژول FFT

۱- تصویر scope خروجی و ورودی و همچنین spectrum viewer را در گزارش بیاورید.

۲-۲- سنتز مدل و تولید کد Verilog مدل FFT

در این بخش با استفاده از ابزار HDL coder مدل FFT را سنتز می‌کنیم و کدهای Verilog مدل را تولید می‌کنیم. برای این کار مراحل زیر را طی کنید.

- برای تولید کردن کد Verilog سخت‌افزار مربوطه، وارد تب Modeling شوید و گزینه Model setting را انتخاب نمایید.
- در بخش solver، type را fixed step و solver را بر روی discrete قرار دهید.
- بر روی بخش HDL Code Generation کلیک کنید و زبان سخت افزار را Verilog انتخاب کنید (شکل ۵).



شکل ۵- تنظیمات HDL Code Generation

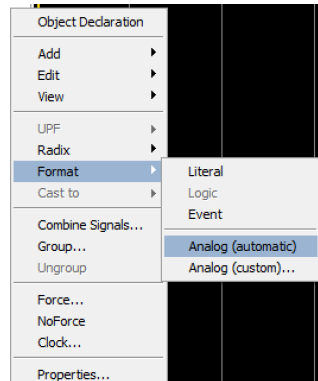
- در قسمت target فرکانس را برابر ۵۰ MHz قرار دهید.
- در بخش optimization قسمت‌هایی برای pipeline و همچنین resource sharing وجود دارد که فعلاً با آن‌ها کاری نداریم.
- در بخش global setting تنظیماتی از قبیل سنکرون بودن یا نبودن reset و... وجود دارد که با آن‌ها نیز کاری نداریم.
- در بخش report تمام موارد را فعال کنید تا پس از سنتز مدار بتوانید آن را تحلیل کنید.
- در بخش test bench گزینه HDL test bench را فعال کنید سپس تنظیمات را اعمال کنید و پنجره را ببندید.
- روی ماژول FFT Burst کلیک راست کرده و در بخش HDL Code گزینه HDL Workflow Advisor را انتخاب کنید.
- در بخش 3.2 Generate HDL code and test bench گزینه Generate test bench را فعال کنید.
- ۲- هر سه مرحله را Run کنید و تصویری از موفقیت آمیز بودن اجرا در گزارش بیاورید.
- پس از مدتی کد verilog سخت افزار تولید می‌شود همچنین گزارش سنتز نمایش داده می‌شود.
- ۳- در گزارش تولید شده بر روی شکل ماژول Latency آن گزارش شده است تصویر این بخش را در گزارش خود بیاورید و مفهوم آن را بیان کنید.
- ۴- همچنین تصویر critical path estimation را در گزارش خود بیاورید و مفهوم آن را بیان کنید.
- ۵- تصاویر تمام گزارش‌های مربوط به high-level resource report را در گزارش بیاورید.

۲-۳- شبیه‌سازی کدهای تولید شده توسط HDL coder

- در این بخش کدهای تولید شده توسط HDL coder را در Modelsim شبیه‌سازی می‌کنیم و نتیجه بدست آمده از شبیه‌سازی را با شکل موج شبیه‌سازی MATLAB مقایسه می‌کنیم. برای این کار مراحل زیر را طی کنید:
- کدهای تولید شده در پوشه hdl_prj قرار دارند. آن‌ها را در نرم افزار modelsim اجرا کنید. برای این کار از test bench تولید شده توسط MATLAB (FFT_Burst_tb.v) استفاده کنید.



۶- تصویر سیگنال‌های `dataIn_re`, `validIn`, `dataOut_re`, `dataOut_im`, `validOut` را در تمام طول شبیه‌سازی در گزارش بیاورید و مشخص کنید هر کدام چه سیگنالی هستند. دقت داشته باشید برای نمایش سیگنال‌های مذکور (به جز سیگنال‌های `valid`) باید فرمت `analog` آن‌ها را مطابق شکل ۶ نمایش دهید.



شکل ۶- نمایش سیگنال بصورت `analog` در `modelsim`

- ۷- سیگنال‌های خروجی شبیه‌سازی `modelsim` را با سیگنال‌های خروجی `Simulink` مقایسه کنید و نتیجه را گزارش کنید.
- ۸- با توجه به مفهوم `Latency`، این پارامتر را از شکل موج‌های شبیه‌سازی استخراج کنید و با مقدار بدست آمده در بخش ۳ مقایسه کنید.
- ۹- همانطور که در کدهای `verilog` تولید شده مشاهده می‌شود واحدی به نام `Butterfly Unit` وجود دارد. درباره نحوه کار این واحد تحقیق کنید و بصورت مختصر توضیح دهید.
- فایل با نام `"FFT_HDLOptimizedExample_Streaming.slx"` که حاوی نوع دیگر محاسبه کننده تبدیل فوریه سریع است را اجرا کنید.
- ۱۰- مراحل ۲ تا ۵ را برای این ماژول نیز انجام دهید.
- ۱۱- تمام نتایج بدست آمده از بخش‌های ۲ تا ۵ برای دو ماژول `Burst` و `Streaming` را مقایسه و تحلیل نمایید.
- ۱۲- درباره تفاوت میان ارتباط `Burst` و `Streaming` تحقیق کنید و بصورت مختصر توضیح دهید.

بارمبندی سوالات

- بخش ۱: ۳۵ نمره
- بخش ۲: ۶۵ نمره

- نام فایل تحویلی به صورت `CA2_Name_#StudentNum.zip` باشد.
- فایل آپلود شده باید شامل گزارش و فایل‌های تولید شده پس از انجام مراحل بالا باشد.
- فایل‌های هر بخش را در یک پوشه جداگانه قرار دهید.
- تمیزی گزارش و مرتب بودن فایل‌های آپلود شده نمره امتیازی دارد.

موفق و سلامت باشید.