



بسمه تعالی



## درس طراحی در سطح سیستم نکاتی در مورد استفاده از HDL Coder

پردیس دانشکده‌های فنی دانشگاه تهران

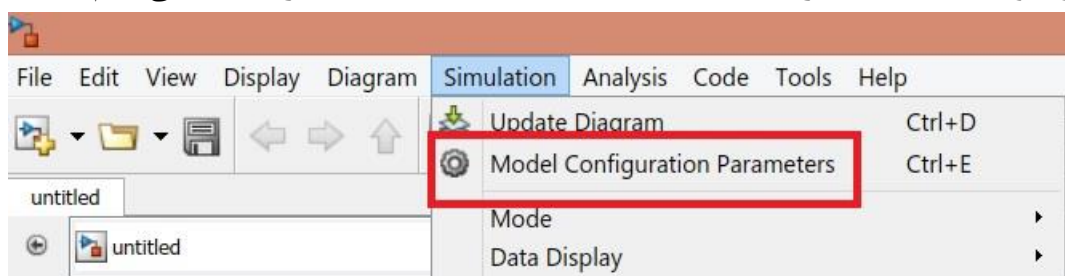
دانشکده مهندسی برق و کامپیوتر

دکتر بیژن علیزاده

نیمسال دوم سال تحصیلی ۹۸-۱۳۹۷

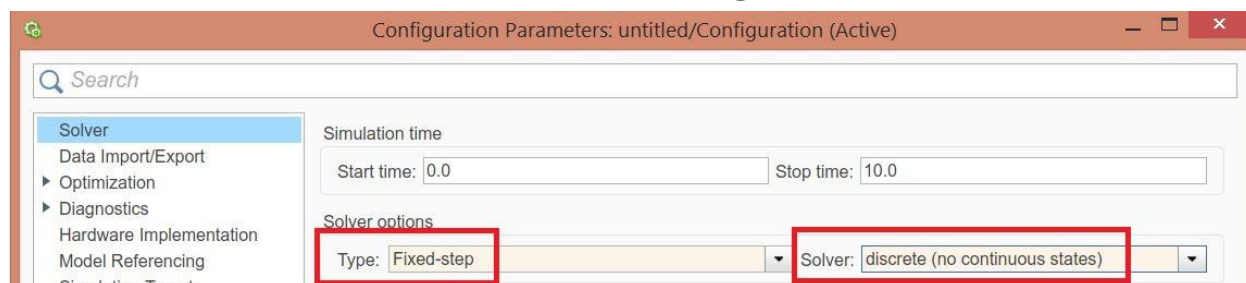
نگارش: آرمان واسعی ([arman.helli@yahoo.com](mailto:arman.helli@yahoo.com))

در ابتدای استفاده از محیط سیمولینک برای تولید کد HDL حتما باید تنظیمات زیر را انجام بدهیم:  
در ابتدا از منوی Simulation گزینه ی Model Configuration Parameters را انتخاب می‌کنیم.



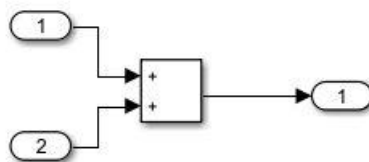
شکل ۱ - تنظیمات شبیه‌سازی

بعد تنظیمات Solver را به گونه‌ی زیر تغییر می‌دهیم. Solver نقش شبیه‌سازی ورودی و محاسبه‌ی خروجی طبق ورودی را دارد و چون باید کد HDL تولید کنیم، پس ورودی‌ها باید به صورت گسسته باشند و تنظیمات زیر نقش تعیین گسسته بودن مدل شبیه‌سازی را ایفا می‌کنند:

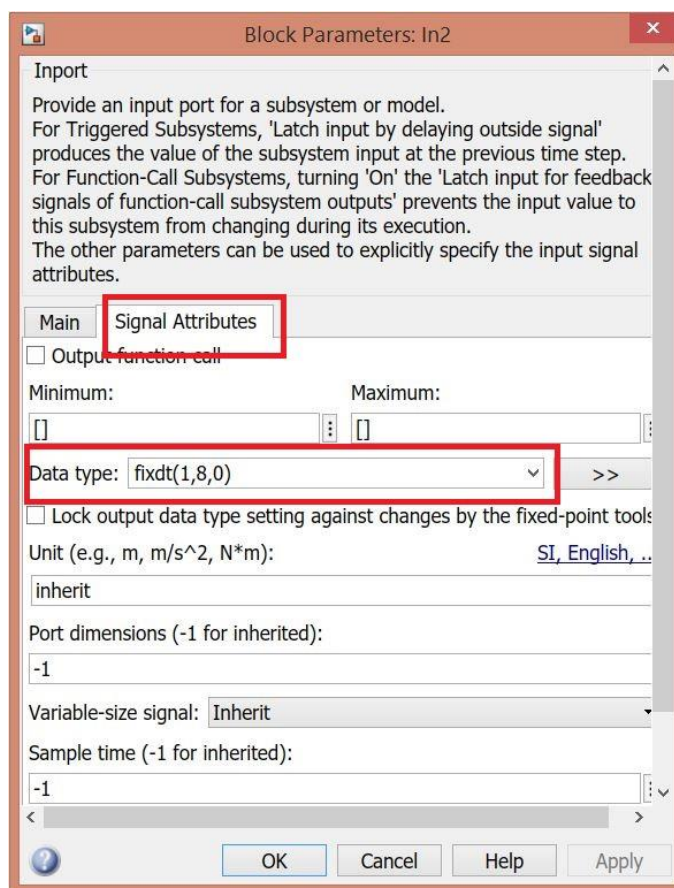


شکل ۲ - تنظیمات Solver

برای سیستم‌هایی با انجام عملیاتی مانند جمع و بهره (مانند فیلترهای FIR) که بلوک‌های مربوطه به شکل Fixed Point توسط HDL Coder پشتیبانی می‌شوند صرفا نوع ورودی مورد نظر را با دبل کلیک کردن بر روی ورودی‌ها انتخاب می‌کنیم. سیستم ساده زیر را در نظر بگیرید:



شکل ۳ - سیستم ساده در نظر گرفته شده برای تولید کد HDL  
 روند تعیین نوع ورودی در شکل زیر مشاهده می‌شود.  $\text{fixdt}(1,8,0)$  به این معنی است که ۱ علامت‌دار بودن نوع داده را تعیین می‌کند، ۸ به معنای تعداد بیت ورودی است و ۰ نیز تعداد بیت‌های اعشاری را نشان می‌دهد.



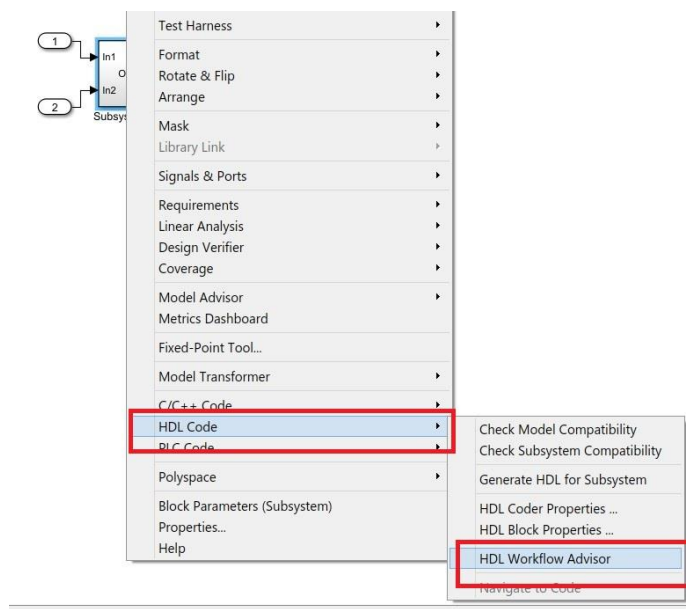
شکل ۴ - تنظیمات نوع ورودی

همچنین نوع خروجی را نیز از نوع `Inherit: auto` انتخاب می‌کنیم که خود HDL Coder با ملاحظات ریاضی و منطقی تعداد بیت‌های آن را تعیین کند.

سپس در محیط خط فرمان متلب دستور زیر را برای انتخاب ابزار سنتز وارد می‌کنیم:  
`Hdlsetuptoolpath('ToolName','Altera Quartus II','ToolPath','C:\intel\FPGA_lite\16.1\quartus\bin64\quartus.exe');`  
 در مثال بالا نوع سنتز طول Quartus II بود ولی می‌توان ابزارهای دیگری مانند Vivado یا ISE را نیز انتخاب کرد. منتها باید دقت شود که پارامترهای `ToolPath` و `ToolName` همیشه به شکل بالا در دستور نوشته می‌شوند.

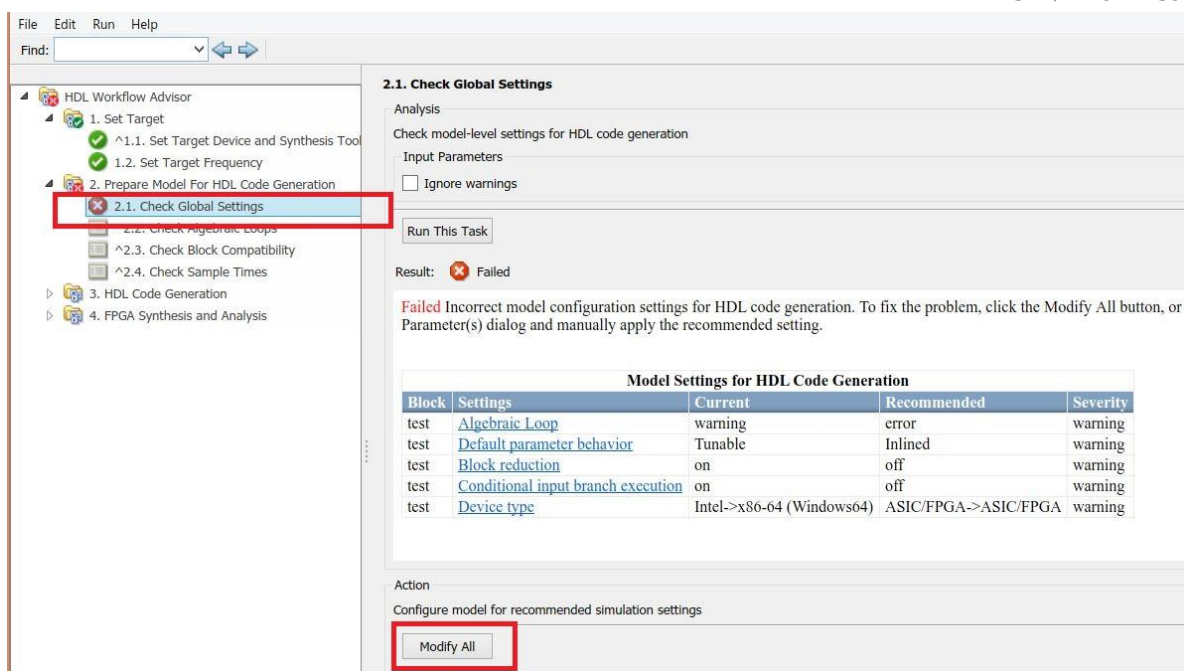
و ترتیب پارامترها نیز مهم است. همچنان در انتخاب آدرس باید در نهایت آدرس کامل فایل‌های .exe یا .bat که فایل‌های اجرایی ابزار سنتز هستند، نوشته شود.

سپس با انتخاب یک Subsystem از بلوک‌های انتخاب شده و با کلیک راست کردن بر روی زیرسیستم طبق شکل زیر روندهای متوالی نشان داده شده را طی می‌کنیم:



شکل ۵ - انتخاب HDL Workflow Advisor

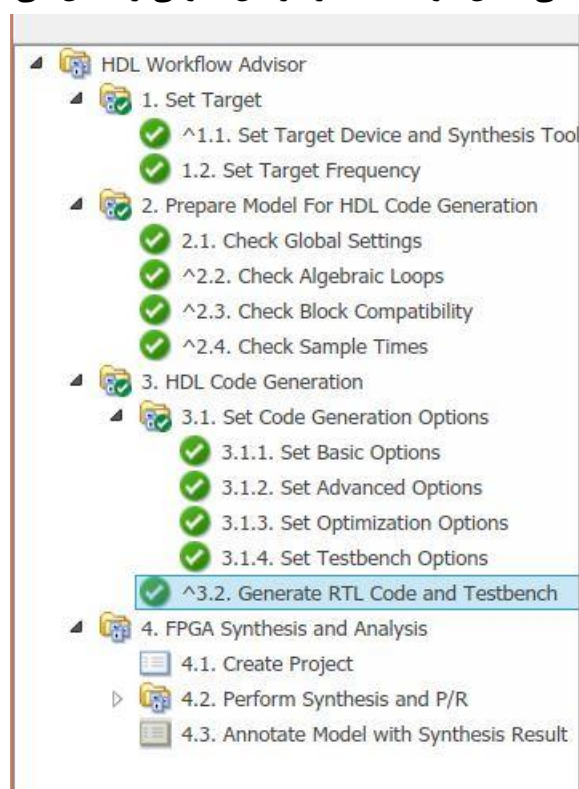
سپس نوع FPGA و همچنین فرکانس کاری مورد نظر را انتخاب می‌کنیم. با اجرای مراحل بالا به خطای زیر برخورد خواهیم کرد:



شکل ۶ - خطای مربوط به تنظیمات پیش فرض

این خطا در واقع نوع واکنش ابزار سنتز به مواردی همچون حلقه‌ی جبری یا فعال بودن تنظیماتی مانند کاهش بهینه بلوک‌ها را نشان می‌دهد. با کلیک بر گزینه‌ی Modify All تنظیمات پیش‌فرض به تنظیمات پیشنهاد داده شده توسط خود HDL Coder تغییر داده می‌شوند و با بار دیگر اجرای این مرحله، این خطا برطرف خواهد شد. مراحل بعدی نیز در صورت وجود نداشتن حلقه‌ی جبری و فیدبک و تنظیمات صحیح Solver که در بالا اشاره شد نیز بدون هیچ مشکلی طی می‌شوند.

مرحله ۳ نیز به تنظیمات نوع کد تولیدی (Verilog یا VHDL) و همچنین پایپ لاین شدن و به اشتراک گذاشتن منابع اختصاص دارد. شکل زیر طی شدن موفقیت آمیز مراحل متوالی را نشان می‌دهد:



شکل ۷ - طی شدن موفقیت آمیز تمامی مراحل

برای صرفاً تولید کد HDL اجرای مراحل تا مرحله 3.2 کافی است. حال نگاهی به کد HDL تولید شده برای سیستم مذکور در بالا می‌اندازیم:

```

module Subsystem
(
    In1,
    In2,
    Out1);

    input  signed [7:0] In1; // int8
    input  signed [7:0] In2; // int8
    output signed [8:0] Out1; // sfix9

    wire signed [8:0] Add_1; // sfix9
    wire signed [8:0] Add_2; // sfix9
    wire signed [8:0] Add_out1; // sfix9

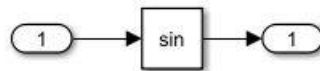
    // <S1>/Add
    assign Add_1 = {In1[7], In1};
    assign Add_2 = {In2[7], In2};
    assign Add_out1 = Add_1 + Add_2;

    assign Out1 = Add_out1;

endmodule // Subsystem

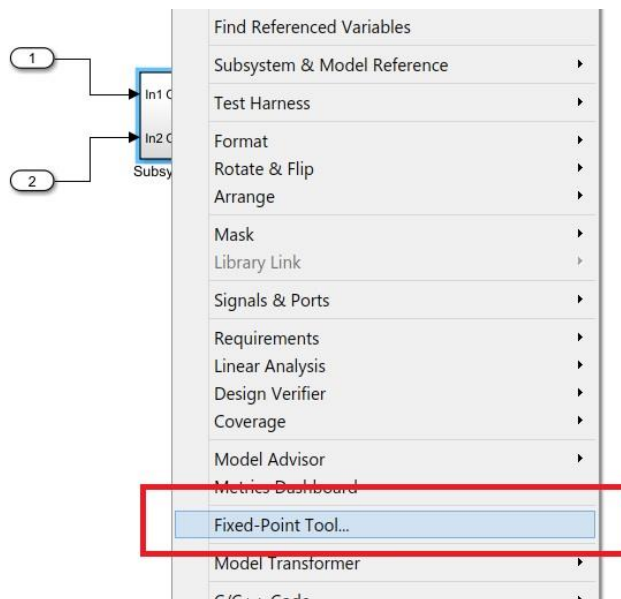
```

همانطور که در کد تولید شده مشاهده می‌شود، HDL Coder نوع خروجی را ۹ بیت در نظر گرفته است که با توجه به ورودی ۸ بیتی انتخاب شده کاملاً صحیح است. اما بلوک‌هایی نیز وجود دارند که فقط از اعداد floating point پشتیبانی می‌کنند مانند توابع مثلثاتی. سیستم زیر را در نظر بگیرید:



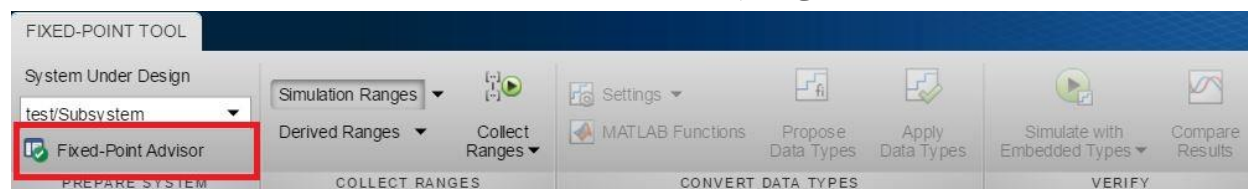
شکل ۸ - سیستم در نظر گرفته شده دوم برای تولید کد HDL

در ابتدا نوع ورودی را از جنس `fixdt(1,16,13)` انتخاب می‌کنیم. (دلیل انتخاب این بازه آن است که چون ورودی باید به رادیان باشد و زاویه مورد نظر ما بین  $\pi$  و  $-\pi$  است، پس در یک عدد ۱۶ بیتی، صرفاً ۳ بیت علامت و صحیح برای نمایش این بازه کفایت می‌کند). سپس باید از ابزار Fixed-Point Tool استفاده کنیم که پنجره آن با کلیک راست کردن روی زیرسیستم باز می‌شود:



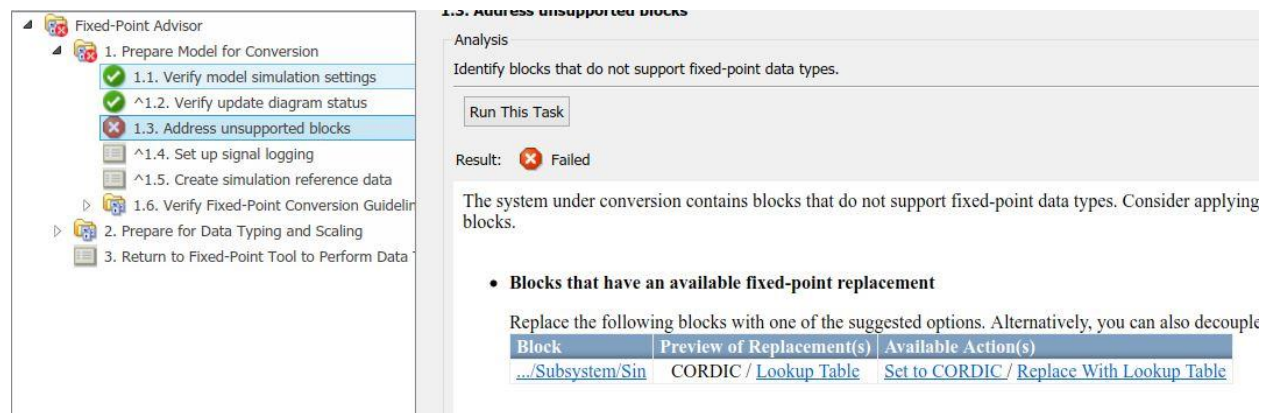
شکل ۹ - ابزار Fixed-Point Tool

ابتدا باید سیستم موردنظر را برای تبدیل شدن از floating point به fixed point آماده کرد. بدین شکل که ابتدا بر روی Fixed-Point Advisor کلیک می‌کنیم:



شکل ۱۰ - ابزار Fixed-Point Advisor

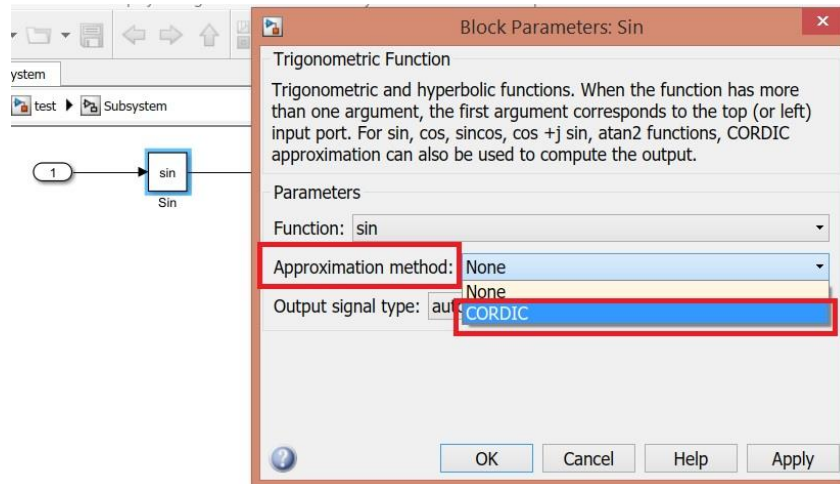
سپس مراحل متوالی را به ترتیب اجرا می‌کنیم. اولین خطایی که به آن برخورد خواهیم کرد، خطای زیر است:



شکل ۱۱ - خطای ناشی از عدم تطابق ورودی با نوع بلوک

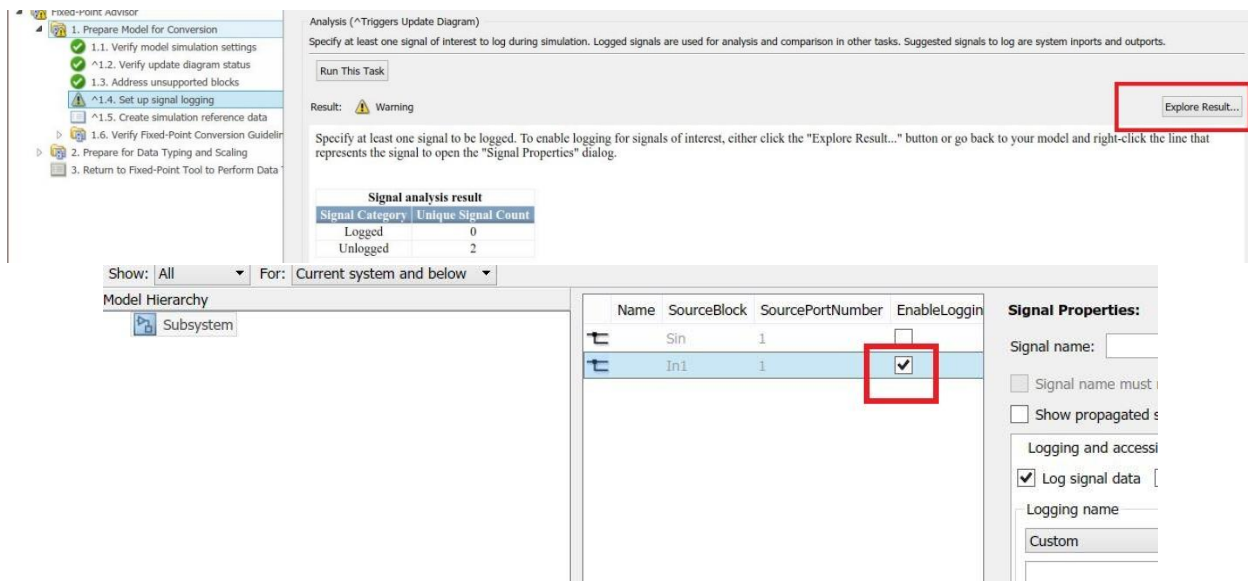
خطای بالا ناشی از عدم تطابق بلوک sin با داده‌هایی از نوع fixed point است. در ابتدا به تنظیمات این بلوک رفته و نوع روش محاسباتی بلوک را به CORDIC تغییر می‌دهیم:





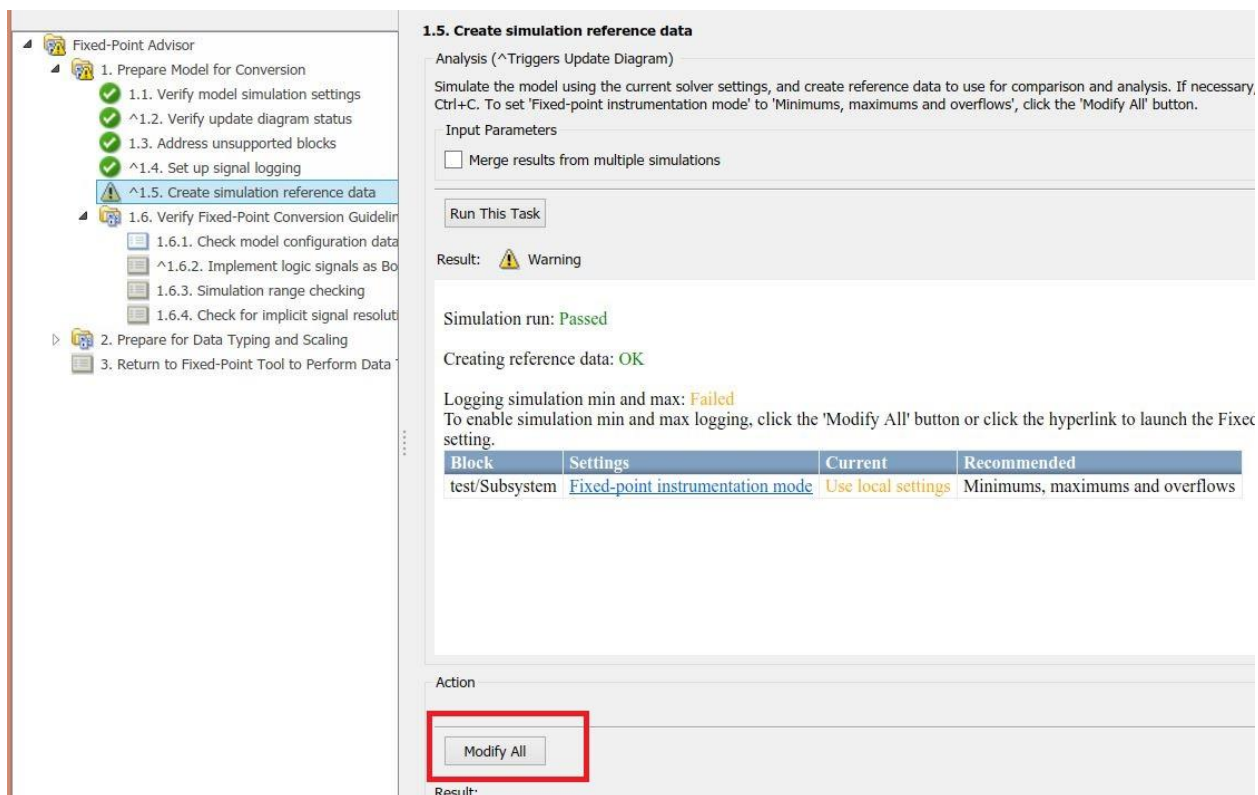
شکل ۱۲ - تغییر روش محاسبات بلوک sin

با انتخاب این گزینه همچنین تعداد تکرارهای مبتنی بر الگوریتم CORDIC را نیز می‌توان انتخاب کرد. هشدار بعدی مربوط به log کردن حداقل یک سیگنال است که برای شبیه‌سازی سیستم توسط خود نرم‌افزار به کار می‌رود. بهتر است سیگنال ورودی را برای این کار انتخاب کنیم. طریقه log کردن سیگنال نیز در شکل زیر نشان داده شده:



شکل ۱۳ - log کردن سیگنال‌ها

بعد از طی شدن مراحل بالا به هشدار زیر که باز به نوع برخورد با سیگنال مرجع برای شبیه‌سازی است، برخورد خواهیم کرد و سپس باز هم با استفاده از دکمه Modify All و اجرای دوباره این مشکل را برطرف می‌کنیم.



شکل ۱۴ - هشدار مربوط به مرجع شبیه‌سازی

در روندهای متوالی مرحله ۱،۶ نیز باز ممکن است به هشدار بالا برخورد کنیم که آن‌ها هم به طریق ذکر شده برطرف می‌شوند.

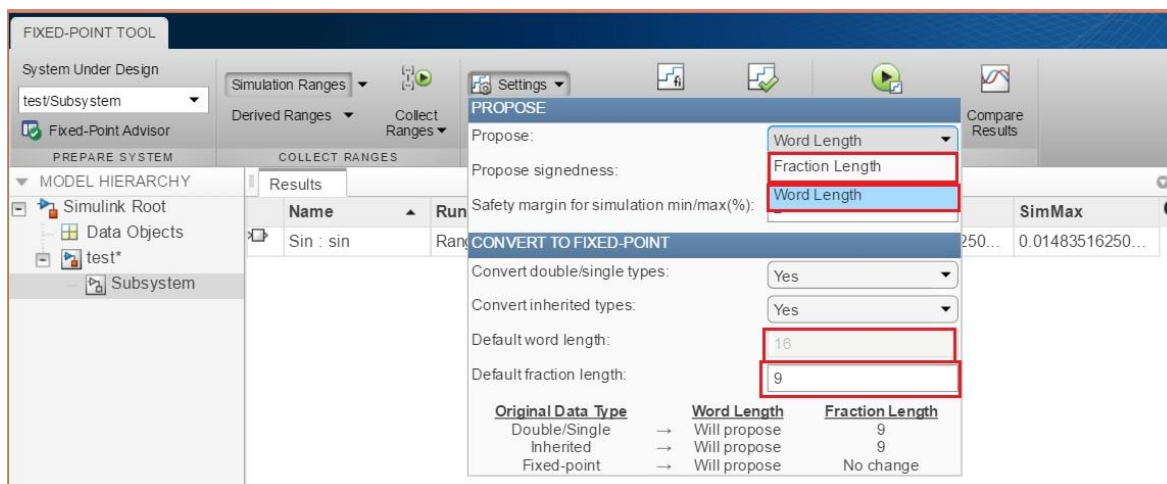
حال بعد از اتمام مراحل بالا با انتخاب Collect Range بازه نتایج شبیه‌سازی داخلی خود MATLAB را بدست می‌آوریم:



شکل ۱۵ - جمع‌آوری نتایج شبیه‌سازی

سپس باید نوع داده‌های مورد نظر از تعداد بیت‌ها و همچنین تعداد بیت‌های اعشاری را انتخاب کنیم:





شکل ۱۶ - انتخاب نوع داده ها

با انتخاب Fraction Length به عنوان پارامتر ثابت انتخاب طول Word برای ما باز می‌شود و سپس با انتخاب Word Length به عنوان پارامتر ثابت می‌توانیم تعداد بیت اعشاری را انتخاب کنیم. چون در اینجا طول Word مد نظر است پس آن را به عنوان هدف انتخاب می‌کنیم.

سپس با انتخاب Propose Data Types هدف را نهایی خواهیم کرد و بعد با انتخاب تیک نشان داده شده نوع داده مورد نظر را تایید نهایی و با کلیک بر Apply Data Types کار را تکمیل می‌کنیم. همانگونه که در شکل زیر نشان داده شده است، نوع خروجی خود به خود به  $\text{fixdt}(1,16,14)$  تغییر می‌کند و چون سینوس یک زاویه بین 1 و -1 می‌باشد، پس نوع خروجی نیز درست مشخص شده است.

Name	Run	CompiledDT	SpecifiedDT	ProposedDT	Accept	SimMin	SimMax
In1	FPA_Reference		$\text{fixdt}(1,16,13)$	$\text{fixdt}(1,16,13)$	<input checked="" type="checkbox"/>		
Out1	FPA_Reference		Inherit: auto	n/a			
Sin : sin	FPA_Reference	$\text{fixdt}(1,16,14)$		n/a		0.0004272460...	0.0004272460...

شکل ۱۷ - نحوه تثبیت نوع داده ها

مابقی مراحل هم که برای تولید کد HDL است همانند مثال قبلی می‌باشد.