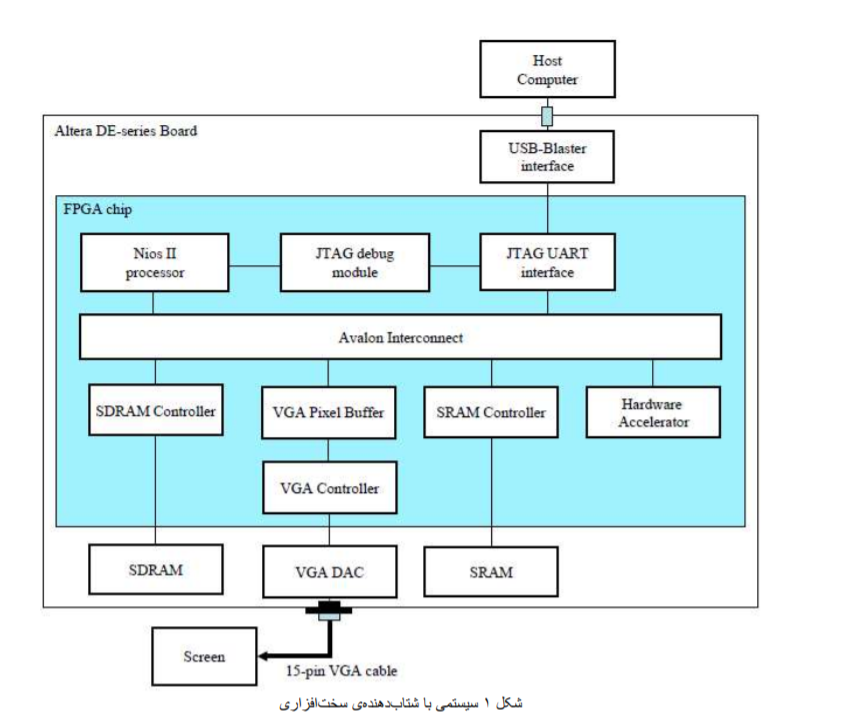
مقدمه:

در این آزمایش می خواهیم صوت را گرفته در بازه های مساوی تقسیم کرده میانگین مجموع مقادیر ان را در ان بازه زمانی محاسبه کرده و نمودار مقادیر محاسبه شده را روی صفحه نمایش نشان دهیم

سپس در زمان پخش ان به ازای مقدار زمان گذشته روی ان بازه از تصویر کشیده شده, تصویر دیگری زیر آن نمایش داده می شود که نشانگر زمان گذشته شده است

این کار ها را یک بار به صورت نرم افزاری سپس به صورت سخت افزاری انجام می دهیم

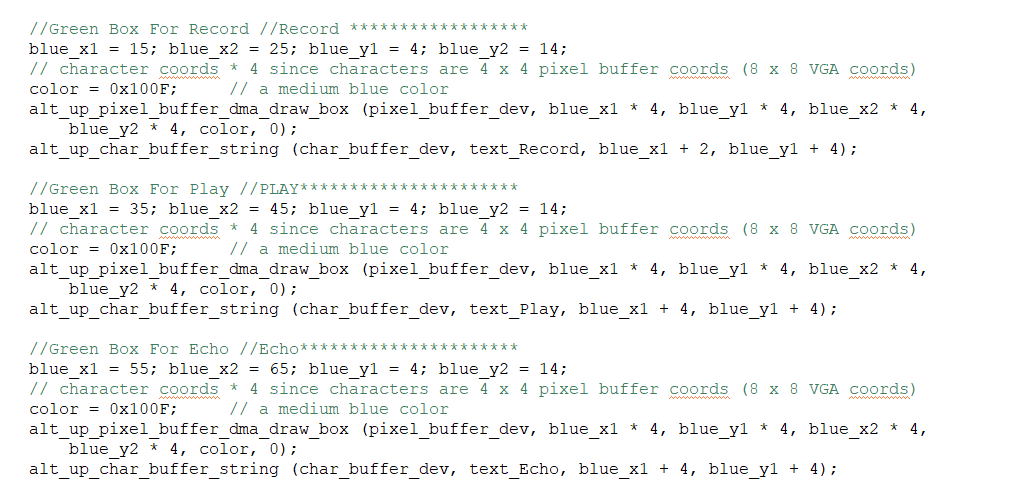
آزمایش:

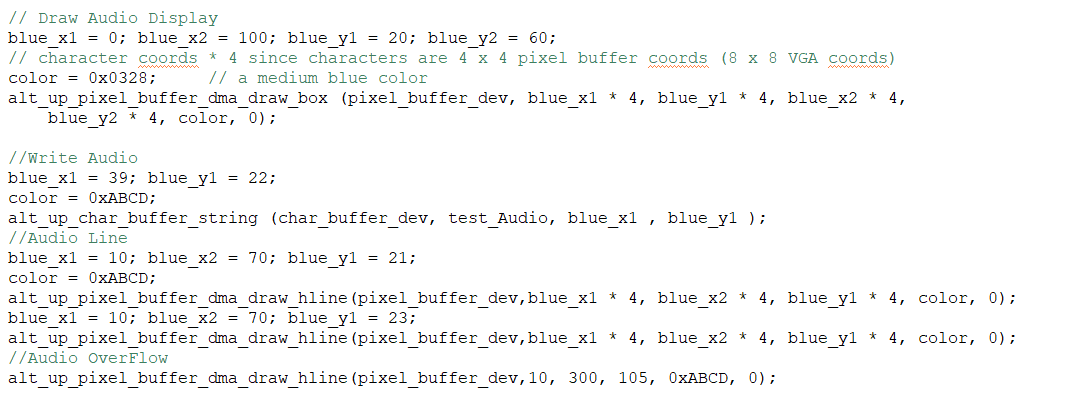


قسمت نرم افزاری:

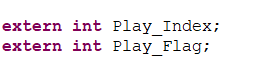
ابتدا 3 باکس که در آزمایش های قبل طراحی شده را در بالای صفحه کشیده و یک خط برای ماکسیموم مقدار نمودار و یک خط برای مینیموم مقدار آن می کشیم و بخش تصویر جدید را مشخص می کنیم

کد های این بخش به صورت زیر است:





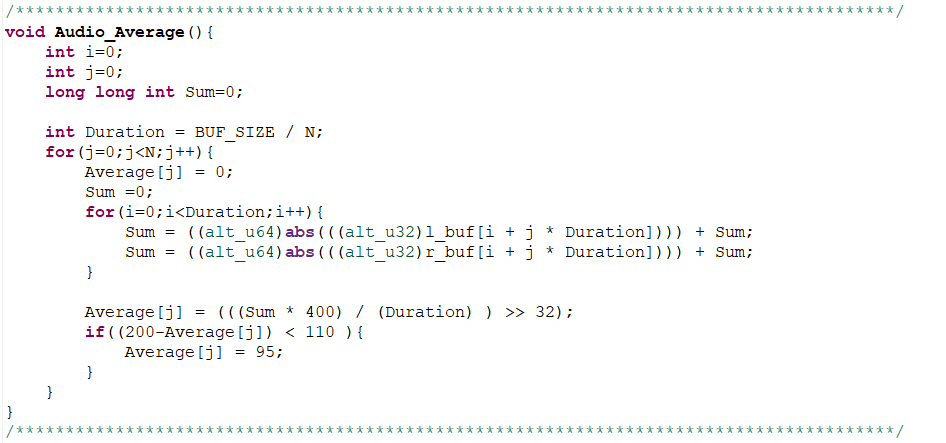
سپس برای بدست آوردن مجموع مقادیر در این بازه ها متغییر های مورد استفاده را تعریف می کنیم:



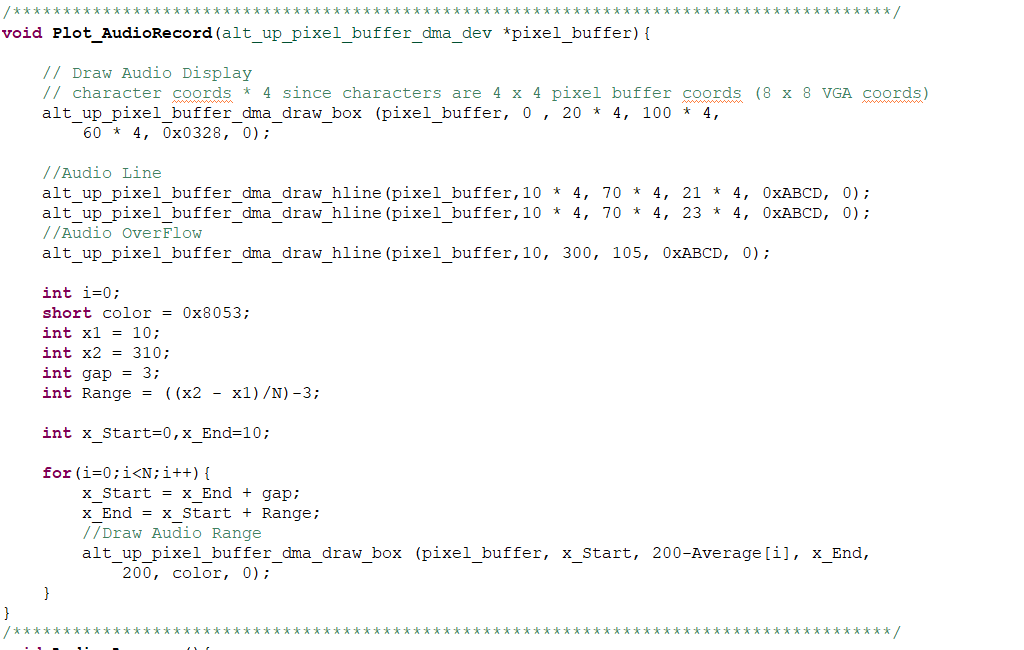
و برای این که مقادیر 32 بیت هستند ما متغییر های 64 بیت تعریف می کنیم تا این مجموع از تعداد بیت ما بیشتر نشود



سپس کد مربوط به محاسبه مجموع این مقادیر را می نویسیم که داریم:

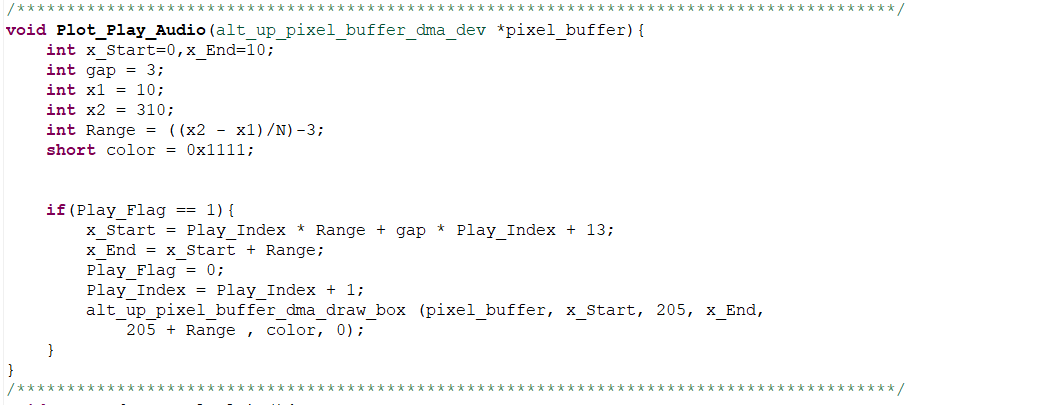


سپس برای کشیدن مقدار مقدار میانگین ان روی صفحه داریم:



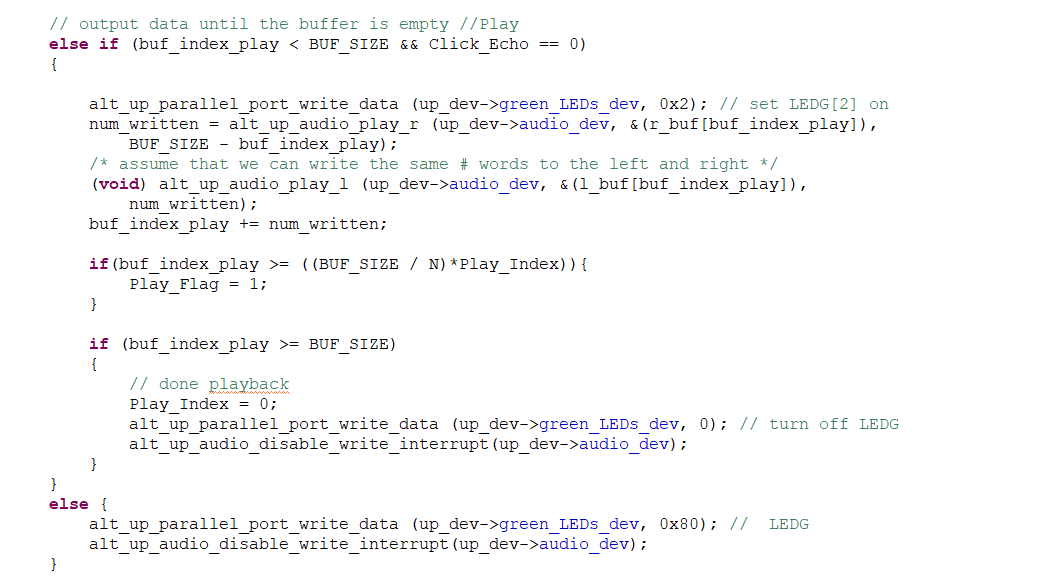
که این تابع بعد از Record کردن صدا, صدا می شود تا میانگین این مجموع ها را گرفته به صورت نمودار روی صفحه نشان دهد و مقدار ماکسیموم ان را نیز مشخص می کنیم تا نمودار روی بقیه صفحه نرود

برای نشان دادن اندازه زمان گذشته بعد از پخش و زدن دکمه Play تابع زیر را هر سری در While کلی کد صدا می کنیم:



که فلگ های هد از فایل اینتراپت مربوط به Audio درست می شود

مانند کد زیر:



بعد از اجرای کد می بینیم که کد کار می کند

برای بدست آوردن زمان از کتابخانه های زیر استفاده می کنیم:



و در کد While آن داریم:



در ان با استفاده از تابع های بالا می توانیم زمان مربوط به اجرای کد را داشته باشیم:

در خروجی داریم:

که ان ها مقدار کلاک گذشته تقسیم بر فرکانس ها که همان زمان گذشته را می دهد است

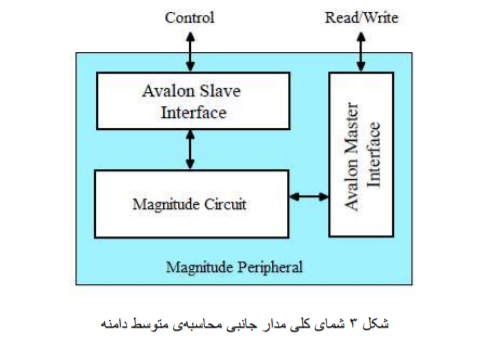
این زمان تقریبا برابر با 4 ثانیه است.

قسمت سخت افزار:

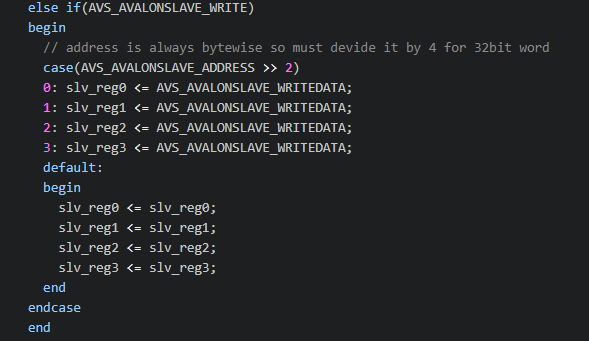
در این قسمت می خواهیم همان کار های قسمت قبل را در سخت افزار انجام دهیم

برای این کار از معماری Slave-master استفاده می کنیم که در ان Slave ما با Nios در ارتباط است و رجیستر های پر شده را می خواند و در صورت نوشتن, می تواند روی ان ها بنویسد.

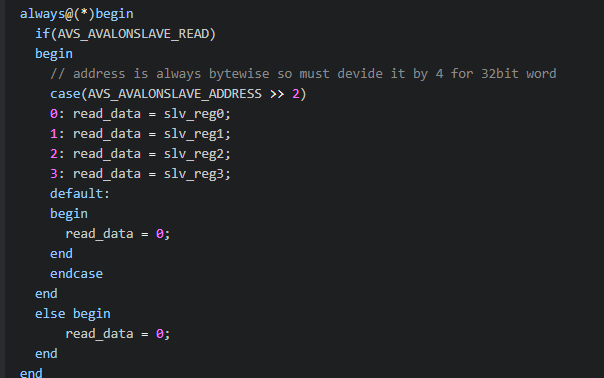
برای این معماری داریم:



ابتدا رابط Slave مربوط به سخت افزار را درست می کنیم که در ان داریم:

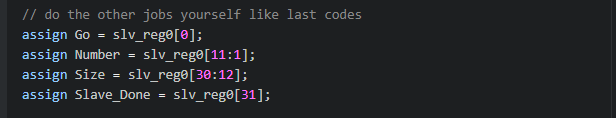
قسمت نوشتن از رجیستر ها به شکل زیر است: 

قسمت خواندن به صورت زیر است:

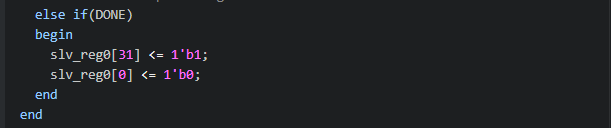


که در ان خواندن بدون کلاک است و سیگنال Wait\_Request 0 است

قسمت رجیستر های ان به شکل زیر است:

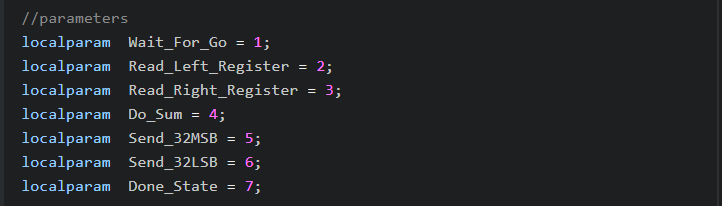


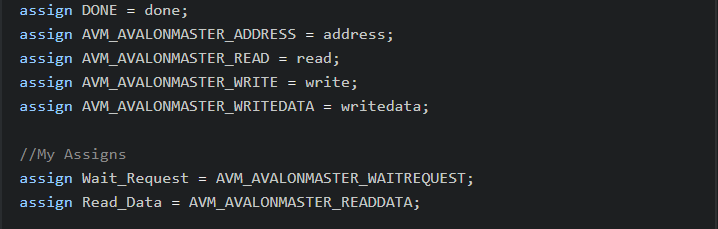
و بعد از تمام شدن به شکل زیر می شود:



سپس برای رابط Master آن داریم:

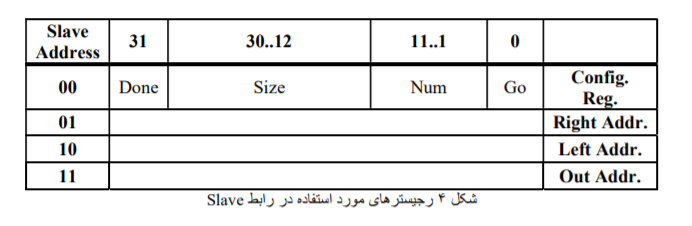
State های قسمت مستر به شکل زیر است:



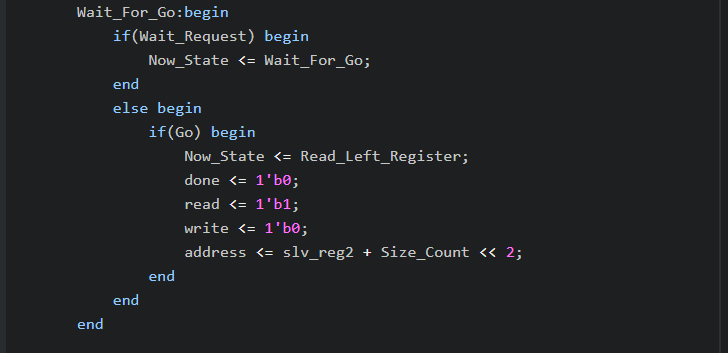
و رابط بین خروجی های اصلی به شکل: 

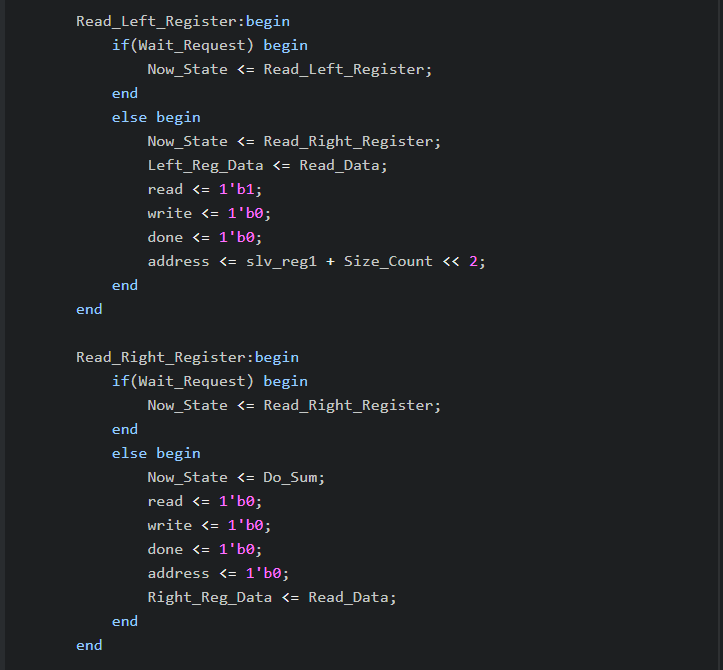
و برای محاسبه مجموع مقادیر ابتدا از رجیستری که ادرس مقدار های راست را دارد حوانده مقادیر را خوانده و جمع می کنیم با مقادیر خوانده شده از چپ.

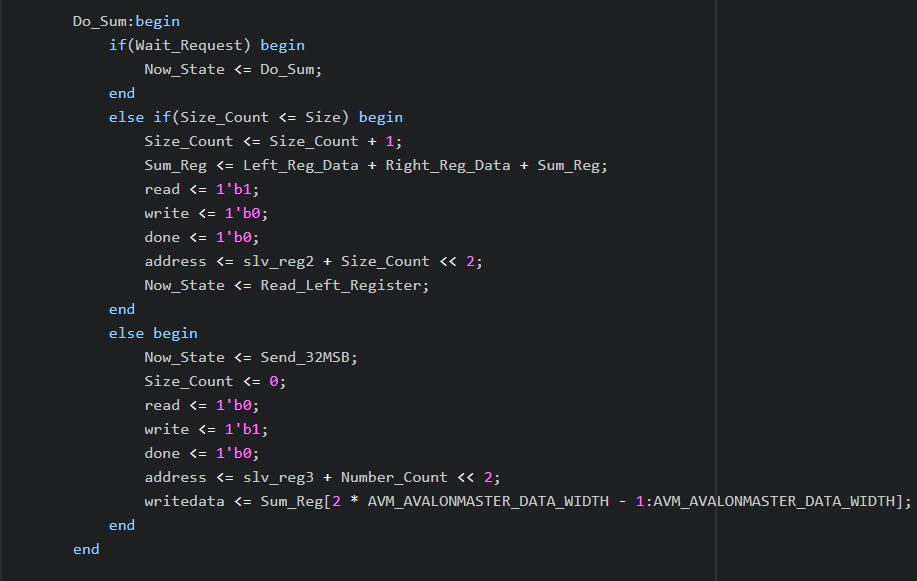
این کار را به تعداد Size انجام می دهیم تا هر یک از بازه های مشخص شده بدست اید سپس برای تعداد Number بار ان را تکرار می کنیم تا تمام بازه های صفحه نمایش بدست اید

رجیستر ها به شکل زیر است: 

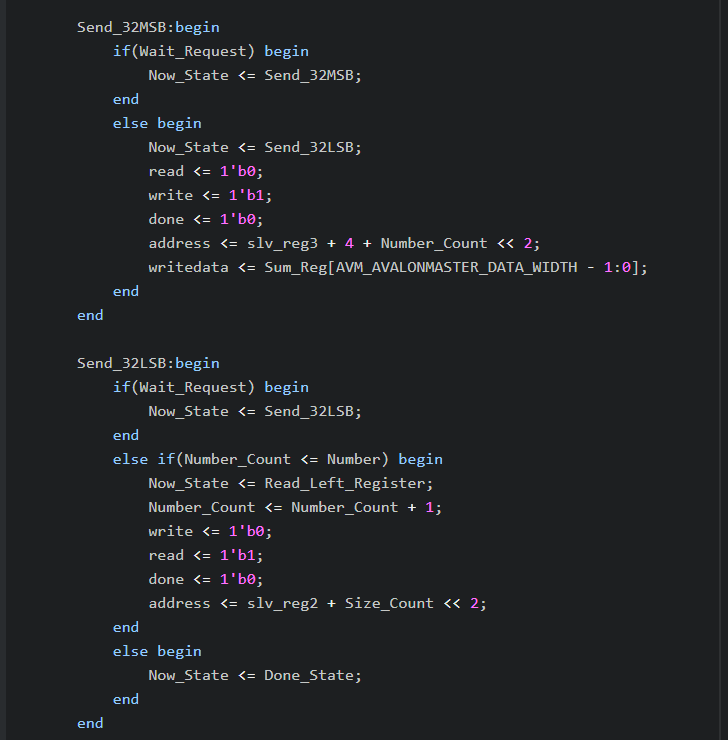
و کد مربوط به هر State به شکل زیر به ترتیب امده است:

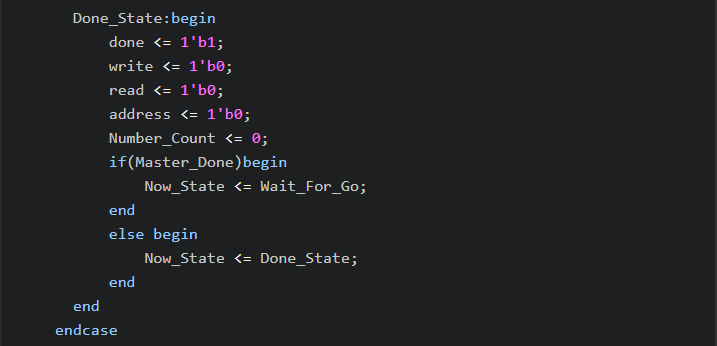






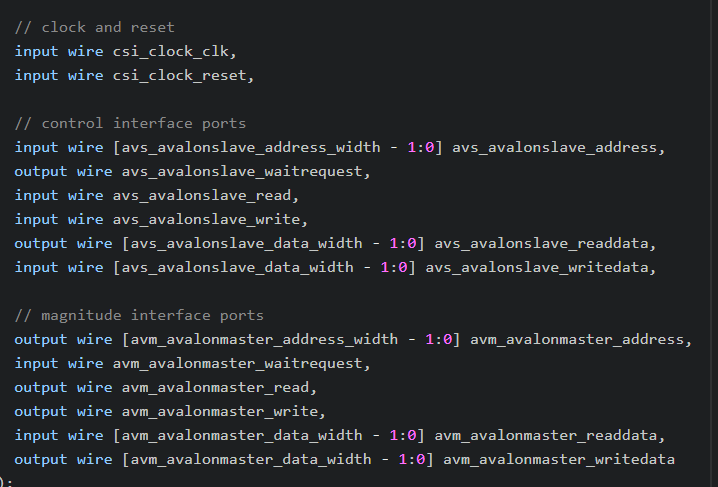
سپس هر یک از این مجموع های بدست آمده را در ادرسی که رجیستر سوم نشان می دهد می نویسد تا تمام بازه ها (Number) تمام شوند و در این حالت به State آخر می رود و سیگنال Done را یک می کند و ان را در رجیستر یک خود می نویسد تا Nios ان را بخواند و از تمام شدن کار خبر می دهد.





و در اینجا کار قسمت Master تمام می شود

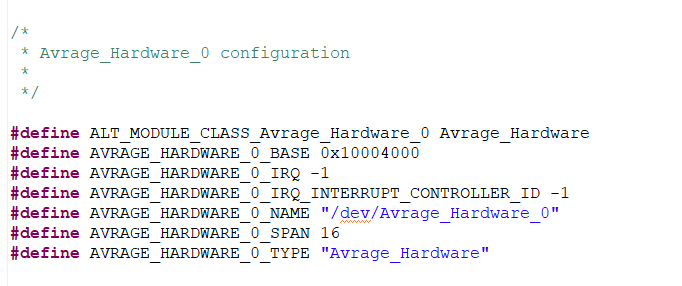
برای ماژول اصلی ما یک فایل درست کرده و از هر دو ی Master و Slave یک Instance گرفته و ان ها را به هم وصل می کنیم تا ماژول کای ما را تشکیل دهد



سپس سخت افزار تولید شده را به Qsys برده و سخت افزار ان را تولید می کنیم و به بقیه مدار وصل می کنیم.

در این حالت Nios یک آدرس به ما می دهد که همان آدرس رجیستر های مربوط به Slave است

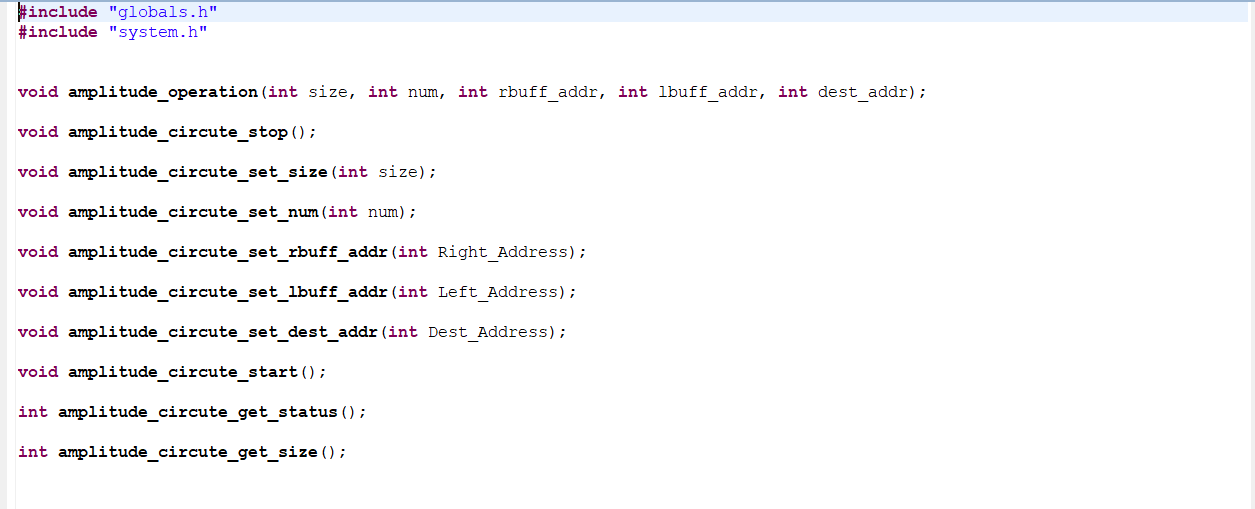
مانند زیر:



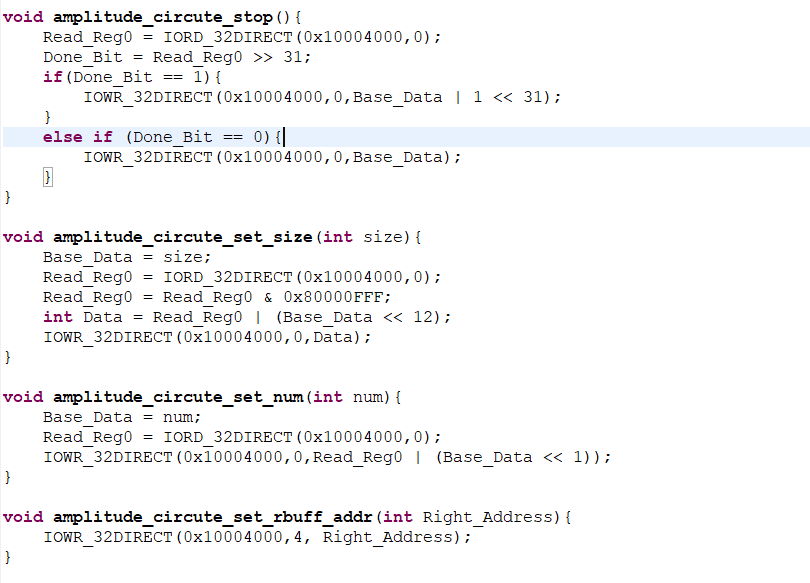
سپس از این آدرس استفاده کرده تا کتابخانه ای بنویسیم که مقادیر را در رجیستر ها نوشته و منتظر بماند تا سخت افزار سیگنال پایان را بدهد

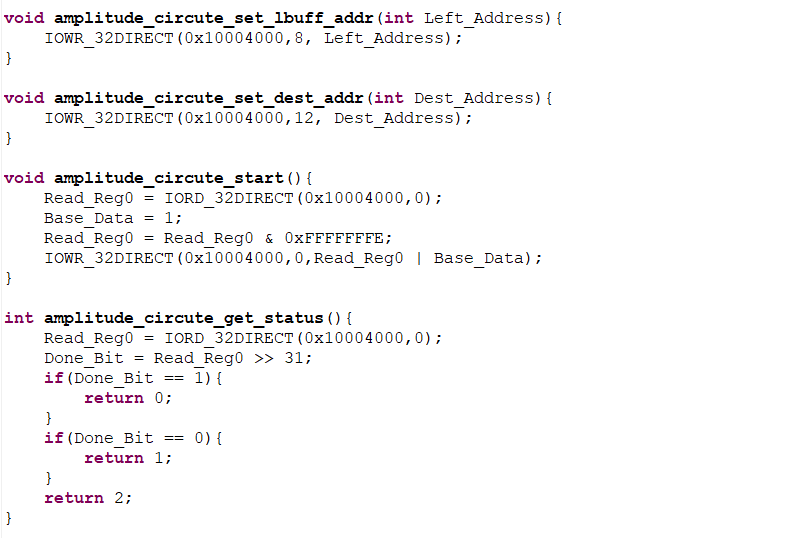
برای این کتابخانه دو فیل جدید مانن زیر می سازیم:



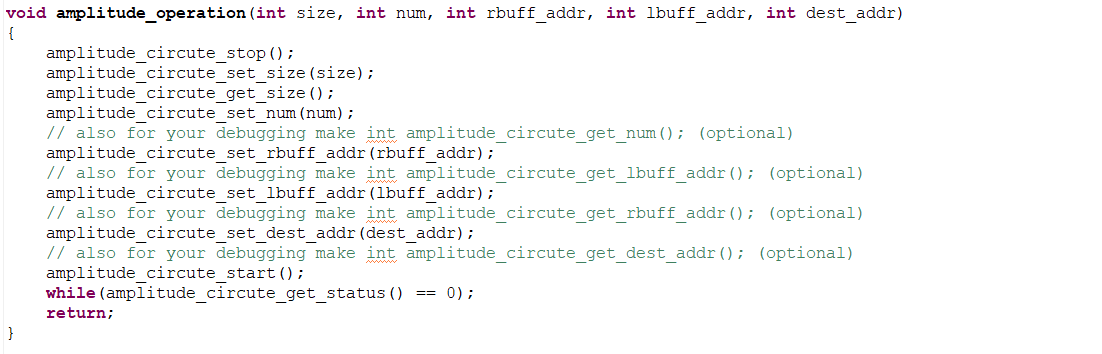


و تعاریف این تابع ها به صورت زیر است:



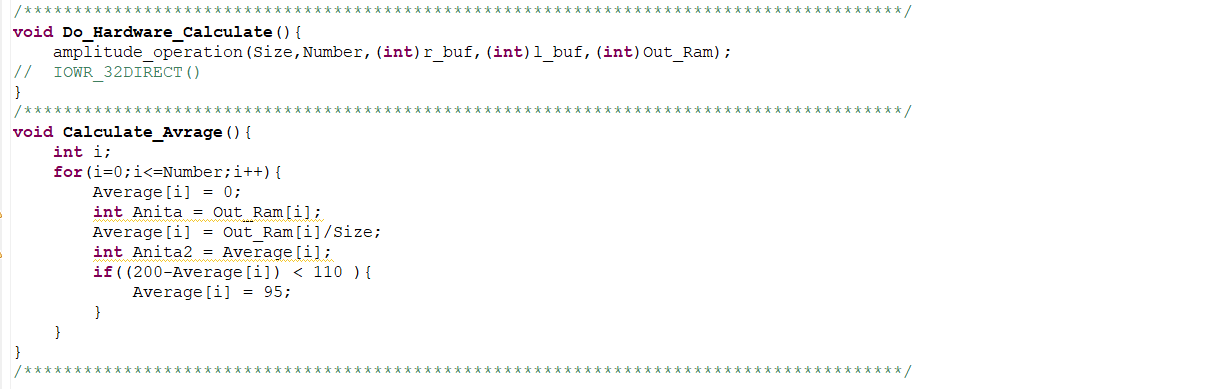


و در اخر تابع زیر در While کلی کد صدا می شود تا سخت افزار کار خود را شروع کند و منتظر می مانیم تا کار تمام شود سپس همانند نرم افزاری مقادیر را روی صفحه می کشم:



سپس در کد While ابتدا تابع بالا را صدا می کنیم تا سخت افزار شروع به کار کند سپس بعد از تمام شدن کار در تابعی دیگر مقادیر میانگین ان را گرفته و برای کشیدن صفحه ان را درست می کنیم.

تابع های زیر برای محاسبه این کار ها است:



و در While اصلی داریم:

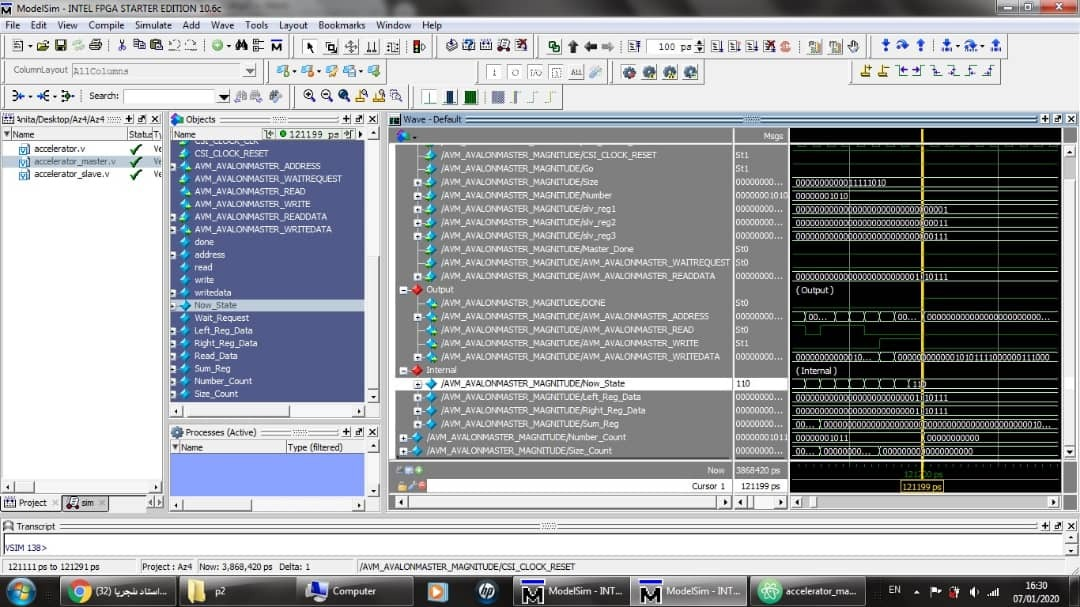


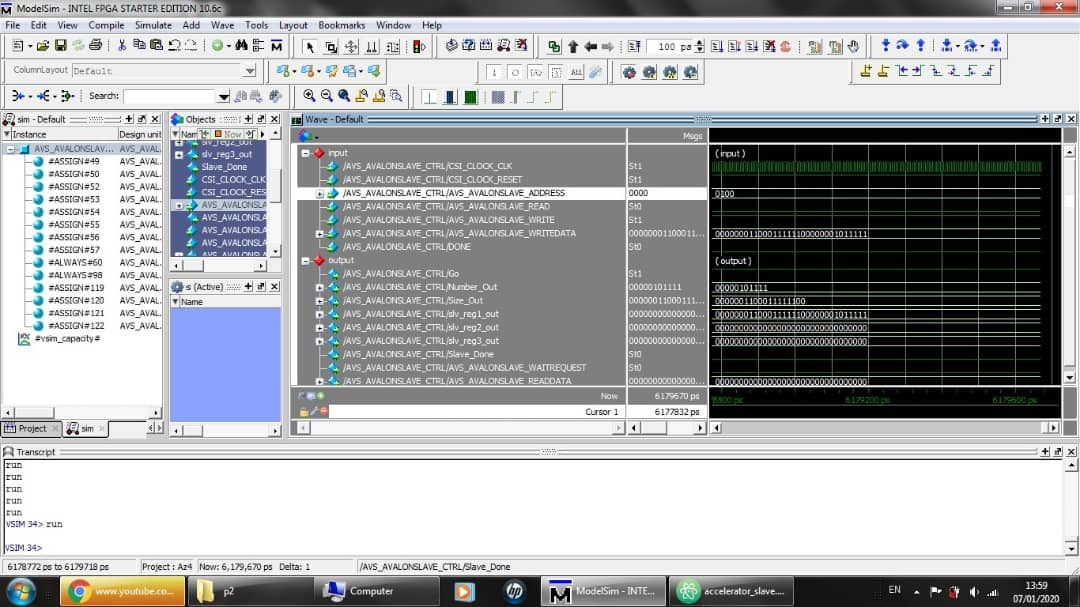
سپس مانند قبل زمان اجرای ان را بدست می آوریم:

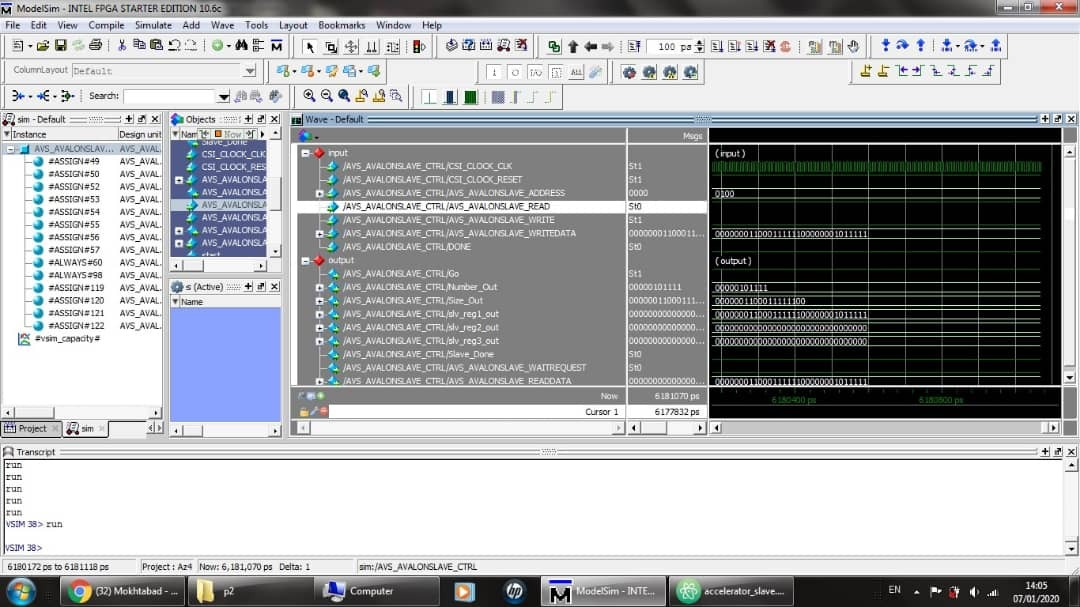
اعداد بدست آماده از قسمت سخت افزار مانند زیر است:

که تقریبا معادل 0.5 ثانیه است.

برای تایید قسمت سخت افزاری ان را در Modelsim تست کرده ایم که شکل های مربوط به آن مانند شکل زیر است:

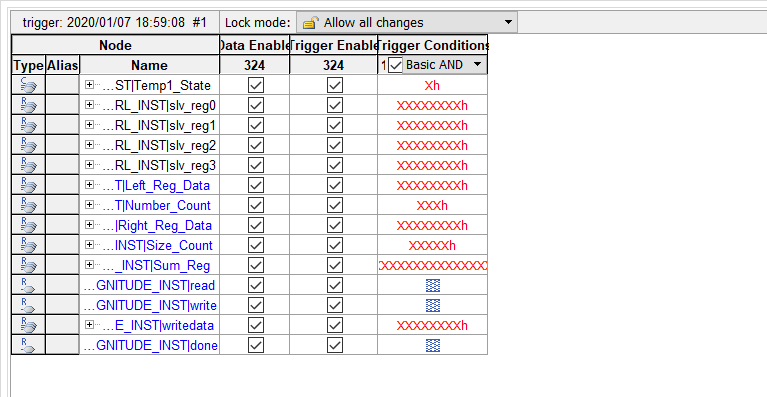






که ان ها نشان می دهند قسمت Slave و قسمت Master هر دو به صورت فانکشنال درست هستند

و در سیگنال تب قسمت Slave ان درست کار می کرده است که داریم:



همه ی کد های سخت افزاری و نرم افزاری به پیوست آمده اند.