



دانشگاه صنعتی اصفهان

دانشکده برق و کامپیوتر

گزارش پروژه‌ی درس زبان توصیف سخت‌افزار و مدارات

# پیاده سازی فیلتر FIR برای حذف نویز

استاد

دکتر مجید نبی

دانشجویان

محمد بصیری ۹۵۲۳۷۷۳

مهدی صمیمی ۹۵۲۹۵۸۳

آرش ماری اورپاد ۹۵۳۲۶۰۳

علی ربانی ۹۵۲۶۹۶۳

محمد مهدی اعرابی ۹۵۲۲۶۱۳

تابستان ۱۳۹۷

## فهرست

چکیده	۳
۱- در مورد FIR Filter	۴
۲- طرح مسئله	۵
۳- فاز اول	۷
۴- فاز دوم	۱۰
۴- ۱- بهینه سازی از نظر منابع مصرفی	۱۰
۴- ۲- بهینه سازی از نظر سرعت و فرکانس کاری	۱۲
۵- فاز سوم	۱۷
۵- ۱- بهینه سازی فضا	۱۷
۵- ۲- بهینه سازی سرعت	۱۹
۶- نتیجه گیری و جمع بندی	۲۰

## چکیده

در این گزارش ابتدا به معرفی فیلتر FIR و کاربرد آن در حذف نویز صوتی پرداخته ایم. سپس طرح مسئله و صورت پروژه را مطرح می‌کنیم که شامل توضیحات در مورد نحوه ی طراحی، نمودار ها و فایل ورودی برای تست کردن ماژول های بدست آمده می باشد.

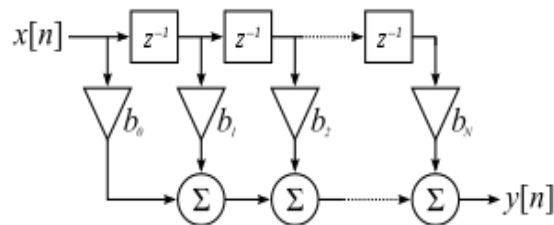
پس از طرح مساله در فاز اول فیلتر را به صورت اولیه پیاده سازی کردیم. در هر فاز گزارش سنتز و نمودار های مربوط ضمیمه شده است. در فاز دوم ماژول طراحی شده برای فاز اول را از دو جهت بهینه سازی کردیم : اول از نظر میزان مصرف گیت ها و حجم ماژول و دوم از نظر سرعت فرآیند فیلتر یعنی در قسمت دوم بدون توجه به میزان گیت های مورد استفاده سعی بر این شده تا حداکثر فرکانس کلاک را برای پردازش ورودی داشته باشیم.

در فاز سوم بدون بهره گیری از فیلتر طراحی شده در دو بخش اول با هسته ی FIR Compiler V.5 به پردازش ورودی پرداختیم در فاز سوم نیز مانند فاز دو، دو معیار سرعت پردازش و حجم منابع مصرفی را مدنظر قرار دادیم با این تفاوت که در فاز دو هدف طراحی با این معیار ها و در فاز سوم ، گرفتن خروجی برای مقایسه با فاز دوم بود.

در نهایت نتایج بدست آمده در هر سه فاز را در بخش آخر گزارش بیان کردیم که شامل تحلیل نتایج حاصل و منابع مصرفی فاز ها با یکدیگر می باشد.

## ۱- در مورد FIR Filter

نام "FIR" مخفف Finite Impulse Response است که به معنای "پاسخ ضربه محدود" است. از فیلترهای دیجیتال و فقی FIR بطور وسیعی برای کاربردهایی نظیر شناسایی سیستم، متعادل کننده ها، حذف نویز فعال حذف پژواک آکوستیکی، رمز سیگنال صحبت،... استفاده می گردد.



شکل ۱ - ۱

شکل ۱ - ۱ یک فیلتر FIR در سطح گیت نشان میدهد بسته به مرتبه فیلتر تعدادی ضریب وجود دارد که ورودی ابتدا در آن ها ضرب شده سپس تمامی مقادیر حاصل با هم جمع میشوند و خروجی ساخته می شود. یک فیلتر FIR با ضرایب و سایر مشخصات خاص طراحی می شود که می تواند در دامنه زمانی یا دامنه فرکانسی باشد.

## ۲- طرح مسئله

در این پروژه مسئله طراحی فیلتر FIR در فاز اول، بهینه سازی آن در فاز دوم و نهایتاً استفاده از هسته پیش طراحی شده زایلینکس به نام FIR Compiler در فاز سوم است. همه ی طراحی ها باید با استفاده از زبان وریلاگ توصیف شوند. در هر فاز به فیلتر طراحی شده می بایست یک فایل صوتی به مدت ده ثانیه ورودی داده شود. این فایل شامل نویزی است که هدف فیلتر حذف این نویز است. برای طراحی فیلتر جدولی به نام "جدول ضرایب" وجود دارد که داده ی ورودی در آن ضرب و سپس همه ی مقادیر با هم جمع میشوند تا خروجی بدست آید. جدول شامل ۱۹ ضریب و فیلتر از مرتبه ۱۸ است. همچنین در هر بخش نیز فایل تستی باید نوشته شود که داده ی ورودی را به ماژول بدهد.

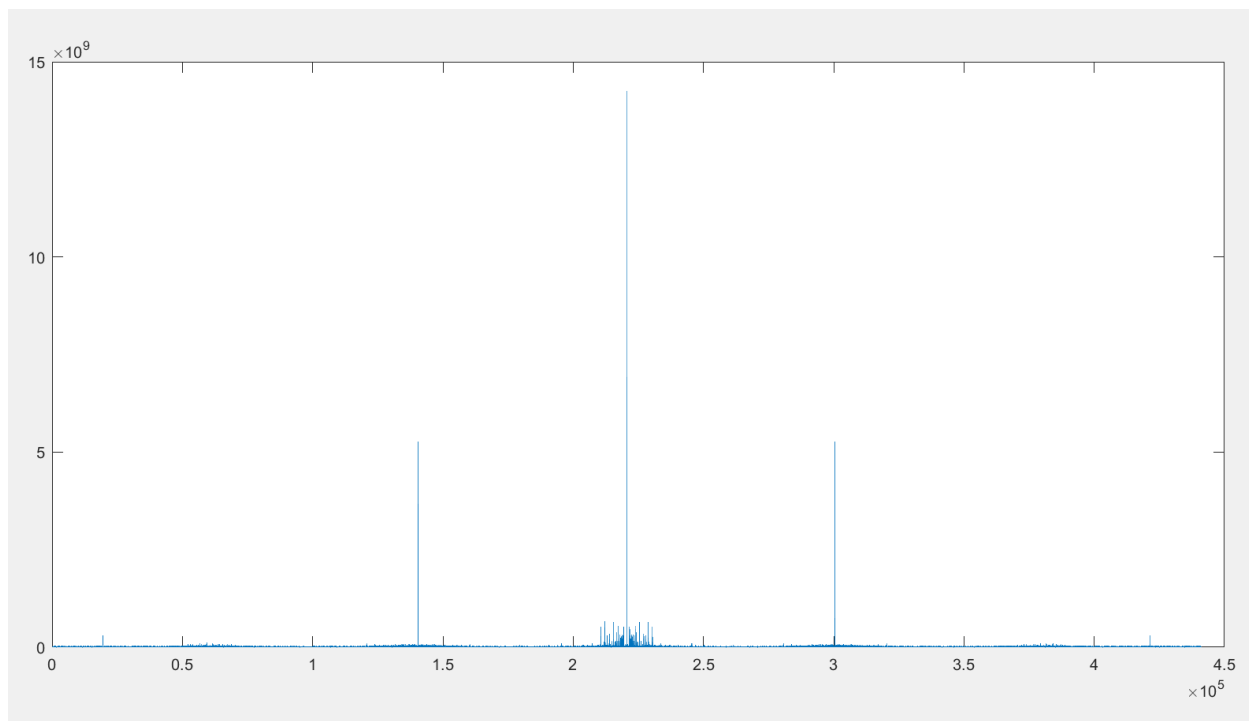
در فاز اول مسئله صرفاً طراحی اولیه از این فیلتر است به این صورت که عملیات بالا که در مورد فیلتر توضیح داده شد به صورت ساده و اولیه با زبان وریلاگ توصیف شود.

در فاز دوم مسئله بهینه سازی فیلتر است یعنی باید پروژه ی فاز اول را از چند نظر بهبود بخشید مثلاً استفاده از پایپ لاین یا اشتراکی کردن منابع (Resource sharing). سپس به این مسئله پرداخت که طراحی جدید چقدر از طراحی قدیمی سریع تر است یا چقدر منابع کمتری را در اختیار می گیرد.

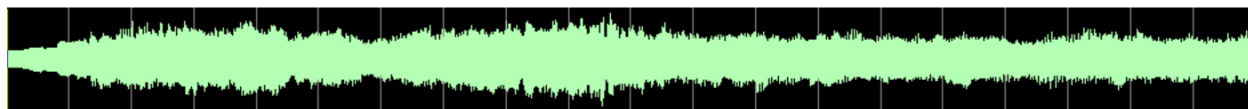
در فاز سوم مسئله بر خلاف دو فاز قبل که طراحی فیلتر بود استفاده از هسته ی از پیش آماده شده ی شرکت زایلینکس با نام FIR Compiler است. در این قسمت می بایست اطلاعاتی که برای طراحی داده شده (مرتبه فیلتر، ضرایب، نرخ نمونه برداری، فرکانس کاری و ...) را به هسته ورودی داده و سپس به فیلتر کردن داده ها پرداخت.

$b_0 = 26$	$b_5 = 8259$	$b_{10} = 18666$	$b_{15} = 2424$
$b_1 = 270$	$b_6 = 12194$	$b_{11} = 15948$	$b_{16} = 963$
$b_2 = 963$	$b_7 = 15948$	$b_{12} = 12194$	$b_{17} = 270$
$b_3 = 2424$	$b_8 = 18666$	$b_{13} = 8259$	$b_{18} = 26$
$b_4 = 4869$	$b_9 = 19660$	$b_{14} = 4869$	

شکل ۲ - ۱: جدول ضرایب برای فیلتر مربوط به پروژه از مرتبه ۱۸



شکل ۲-۲: نمودار حوزه فرکانس برای فایل صوتی نویز دار (ورودی): همانگونه که مشاهده می شود، دو خط عمودی سمت چپ و راست نشان دهنده نویز تک فرکانسی است که در طی پروژه حذف خواهد شد.



شکل ۲-۳: نمودار آنالوگ برای فایل صوتی نویزدار

### ۳ - فاز اول

در این فاز اقدام به ساخت یک FIR Filter به آسان ترین روش ممکن می کنیم. یک آرایه از رجیستر ها به نام reg\_data که ۲۰ عضو دارد و هر کدام ۱۶ بیت هستند در نظر میگیریم و داده های ورودی را در هر کلاک داخل reg\_data [0] می ریزیم و در هر کلاک reg\_data [i] به reg\_data [i+1] انتقال میدهم ( همانند یک شیفت -رجیستر ) و در هر کلاک مقدار ضرب هر reg\_data [i] در ضریب مشخص شده ی خود ( ضریب داده برای این فیلتر ) ضرب کرده و سپس همه ی حاصل ضرب ها را باهم جمع می کنیم و خروجی این ماژول همین حاصل جمع حاصل ضرب هاست.

بدین صورت در هر کلاک یک داده توسط ماژول گرفته می شود و در ابتدای کلاک بعدی این داده در خروجی نمایش می دهد. کلاک بدست آمده برای آی سی اف پی جی ای نوشته شده در گزارش (XC6SSLX9-3TQG144) برابر 601 MHZ بدست می آید که فرکانس بسیار زیادی است حال این سوال پیش می آید آیا این جواب درست است ، در حالی که فرکانس هسته پیش طراحی شده زایلینکس (FIR Compiler) حدودا 400 MHZ است؟

اگر نمودار خروجی ها را نیز ترسیم کنیم صحت عملکرد فاز اول به ضوح مشخص است و تمام نویز حذف می شود پس اشکال کار کجاست؟

حال نگاهی دقیق تر به کد می اندازیم ، برای پیدا کردن مشکل ابتدا باید ببینیم بیشترین زمان در هر پردازش روی خروجی مربوط به کدام قسمت از ماژول و به طور کلی تر کدام قسمت از کد است؟ کد ما در هر کلاک دو قسمت را اجرا می کند:

الف) یک دیتا از ورودی خوانده و این داده را در خانه اول آرایه قرار میدهد و کل آرایه را شیفت می دهد  
ب) همه ی داده ها را در ضرایب خود ضرب می کند و سپس همه را باهم جمع می کند.

بدون شک بیشترین زمان پردازش مربوط به قسمت ب است.

فرض ما این است که در این کد زمان پردازش قسمت ب بیشتر از زمانی است که هسته ی زایلینکس آن را در نظر گرفته و این کلاک برای این پردازش بسیار زیاد است و در این زمان کم حتما داده ی خروجی اشتباه است. در کد قسمت ب از عبارت (\*) @ always استفاده شده است، که به تمامی ورودی های always حساس باشد این قطعه کد باید در هر کلاک حداقل یک بار اجرا شود تا نتایج را روی خروجی پیاده سازی کند ، حال اگر (\*) @ always را با @ (posedge clk) always عوض کنیم (بدون در نظر گرفتن درستی و یا نادرستی در روند

طراحی)، این قسمت از کد در هر کلاک فقط یک بار اجرا می شود پس انتظار ما این است که فرکانس کلاک برای این حالت بیشتر یا مساوی حالت قبلی بشود ، اما نتایج بدست آمده در نرم افزار ISE ، فرکانس 40 MHz را به ما نمایش می دهد.

نتیجه : فرکانس بدست آمده به شدت کاهش یافت در صورتی که باید افزایش پیدا می کرد، پس در حقیقت در کد اولیه ، نرم افزار زایلینکس به اشتباه وابستگی ای بین کد قسمت ب و کلاک قائل نشده بود یعنی در هر کلاک ممکن است پردازش قسمت ب اجرا بشود و یا نشود پس کلاک کلی نباید بیشتر از ۴۰ مگاهرتز باشد ، که با کمی محاسبه کلاک ماژول نزدیک ۳۵ مگاهرتز بدست می آید ، زیرا:

$$\frac{1}{f} = \frac{1}{40} + \frac{1}{600}$$

### رفع مشکل ( کد فاز ۱ ورژن ۵ )

بهترین راه برای حل این مشکل سنکرون کردن تمام مدار است که در ۴ کلاک پردازش روی داده ی ورودی انجام میگیرد ، در کلاک اول داده ورودی عضو صفر آرایه reg\_data قرار میگیرد و همه ی داده های دیگر reg\_data شیفت داده می شوند ، در کلاک دوم همه ی عملیات های ضرب ( ۱۹ عمل ضرب ) به صورت موازی انجام می شوند و هرکدام به روی عضو نظیر خود روی آرایه y\_hasel\_zarb ذخیره می شوند، در کلاک سوم همه ی عملیات های جمع انجام می شوند که ۱۸ عمل جمع متوالی است (اعضای آرایه y\_hasel\_zarb را بایکدیگر جمع می کند) ، در کلاک چهارم داده بخش پر ارزش حاصل عملیات های جمع انجام شده بر روی خروجی ریخته می شود.

#### HDL Synthesis Report

##### Macro Statistics

# Multipliers	: 19
16x16-bit multiplier	: 19
# Adders/Subtractors	: 19
2-bit adder	: 1
36-bit adder	: 18
# Registers	: 5
16-bit register	: 1
2-bit register	: 1
304-bit register	: 1
36-bit register	: 1
684-bit register	: 1

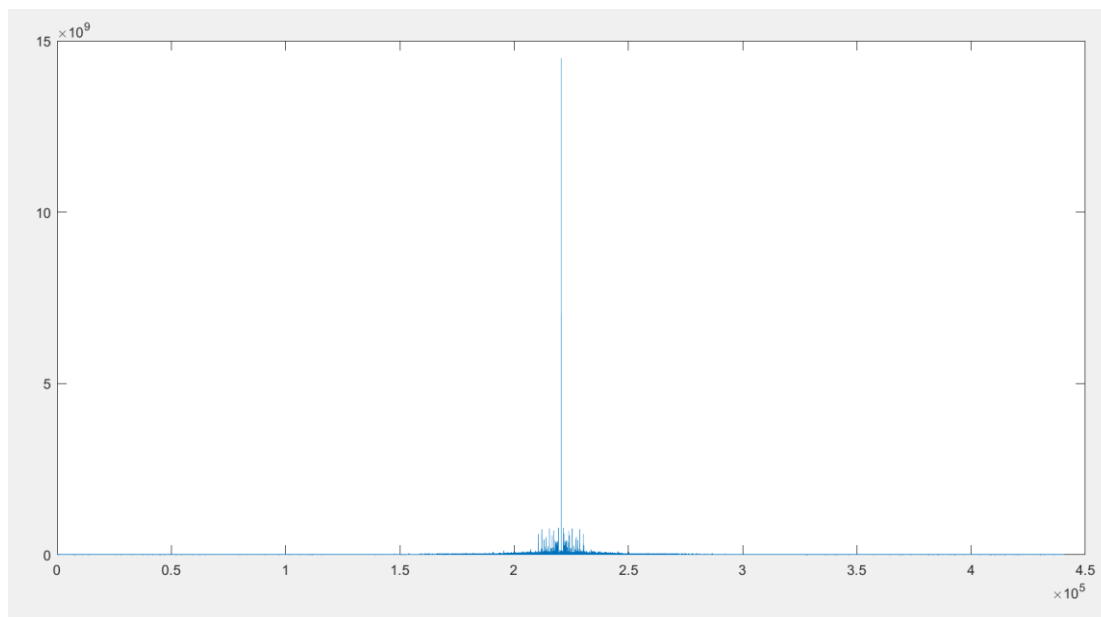


در شکل ۳-۱ اطلاعاتی در مورد میزان مصرف لاجیک ها پس از عملیات سنتر مانند تعداد جمع کننده ، رجیستر و ... را می توانید مشاهده کنید.

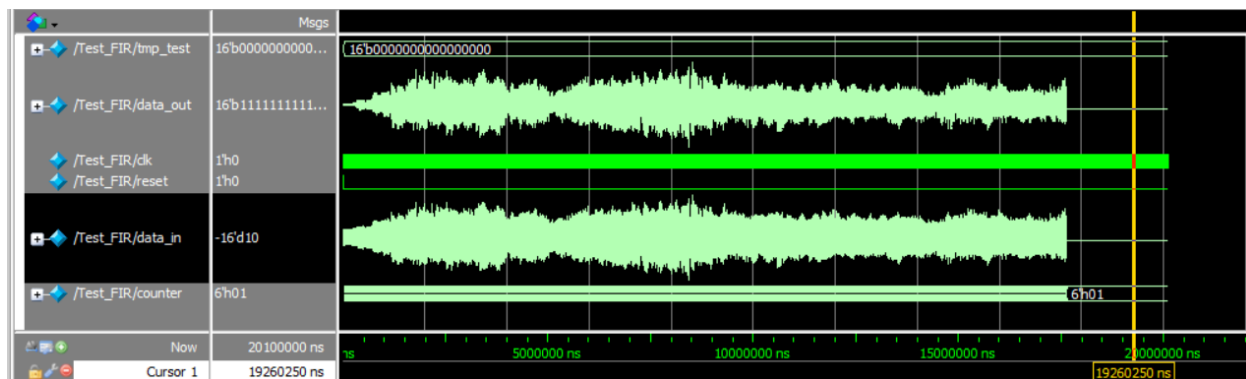
Device Utilization Summary (estimated values)				<a href="#">[-]</a>
Logic Utilization	Used	Available	Utilization	
Number of Slice Registers	405	11440	3%	
Number of Slice LUTs	1115	5720	19%	
Number of fully used LUT-FF pairs	383	1137	33%	
Number of bonded IOBs	49	102	48%	
Number of BUFG/BUFGCTRLs	1	16	6%	
Number of DSP48A1s	16	16	100%	

شکل ۳-۲: جدول Device Utilization

در ادامه نمودار های مربوط به فاز یک آورده شده است ، توضیح هر نمودار در زیر آن نوشته شده است.



شکل ۳-۳: نمودار حوزه فرکانس کد فاز یک: همانگونه که مشاهده می شود دو خط نشانگر نویز حذف شده اند.



شکل ۳- ۴: نمودار آنالوگ حاصل از خروجی فاز اول

## ۴ - فاز دوم

### ۴- ۱ - بهینه سازی از نظر منابع مصرفی

همان طور در در فاز اول دیده شد، برای به دست آوردن خروجی در هر کلاک پالس ۱۹ عمل ضرب و ۱۹ عمل جمع صورت می گرفت که این قضیه به معنای استفاده از ۱۹ ضرب کننده و ۱۹ جمع کننده می باشد. در فاز دوم پروژه جهت بهینه سازی فیلتر ، دو راه پیش روی خود داشتیم ؛

الف) استفاده از خط لوله یا پایپ لاین ب) استفاده از منابع کمتر یا ریسورس شیرینگ

در مورد قسمت ب همان طور در کد نیز مشخص است در هر کلاک پالس تنها یک عمل ضرب و یک عمل جمع انجام می شود که در واقع به معنای استفاده از یک ضرب کننده و یک جمع کننده به جای ۱۹ مورد از آنها در فاز اول است ، ولی بر خلاف فاز اول که در هر کلاک پالس می توانستیم خروجی مورد نیاز را دریافت کنیم، در این فاز هر خروجی پس از ۱۹ کلاک پالس آماده می شود که به معنی کاهش فرکانس کاری ماژول است.

در فاز اول اعمال انجام شده توسط مدار در هر کلاک پالس شامل ۱۹ عمل ضرب، ۱۹ عمل جمع و ... می باشد در حالی که در فاز دوم در هر کلاک پالس فقط یک عمل ضرب و یک عمل جمع صورت می گیرد. بنابراین انتظار می رود که طول پالس ساعت برای فاز اول بیشتر از فاز دوم در قسمت ریسورس شیرینگ باشد که با توجه به گزارش همین طور نیز می باشد (حداکثر فرکانس پالس ساعت برای این بخش 130.625 مگاهرتز بدست آمده است).

# HDL Synthesis Report

## Macro Statistics

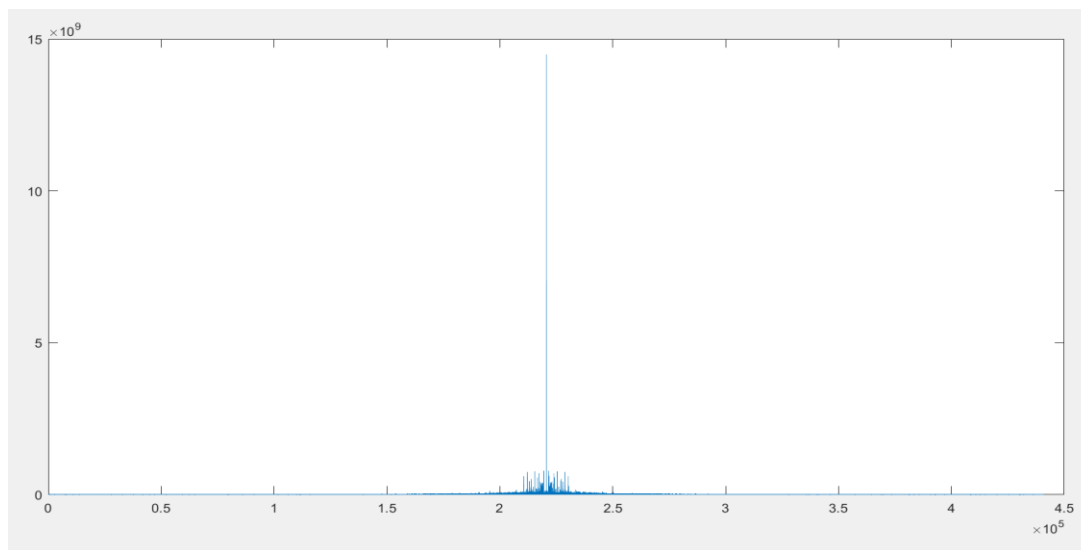
# RAMs	: 1
32x16-bit single-port Read Only RAM	: 1
# Multipliers	: 1
16x16-bit multiplier	: 1
# Adders/Subtractors	: 2
36-bit adder	: 1
6-bit adder	: 1
# Registers	: 4
16-bit register	: 1
304-bit register	: 1
36-bit register	: 1
5-bit register	: 1
# Multiplexers	: 2
16-bit 19-to-1 multiplexer	: 1
5-bit 2-to-1 multiplexer	: 1

شکل ۴ - ۱ - ۱

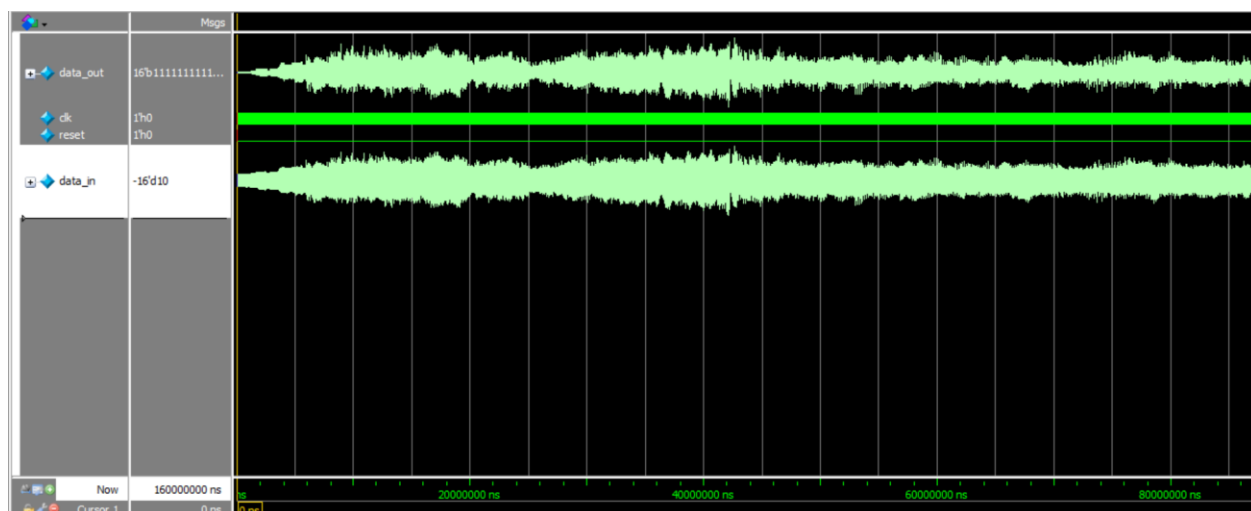
در شکل ۴ - ۱ - ۱ اطلاعاتی در مورد میزان مصرف لاجیک ها پس از عملیات سنتر مربوط به قسمت اول فاز دوم را ملاحظه می کنید مقایسه اطلاعات این شکل به مورد مشابه در فاز اول در بخش نتیجه گیری آورده شده است.

Device Utilization Summary (estimated values)				[1]
Logic Utilization	Used	Available	Utilization	
Number of Slice Registers	334	11440	2%	
Number of Slice LUTs	423	5720	7%	
Number of fully used LUT-FF pairs	312	445	70%	
Number of bonded IOBs	34	102	33%	
Number of BUFG/BUFGCTRLs	1	16	6%	
Number of DSP48A1s	1	16	6%	

شکل ۴ - ۱ - ۲ : جدول Device Utilization برای قسمت اول



شکل ۴-۱-۳: نمودار حوزه فرکانس بخش اول فاز دوم: همانگونه که مشاهده می شود دو خط نشانگر نویز حذف شده اند.



شکل ۴-۱-۴: نمودار آنالوگ حاصل از خروجی فاز دوم قسمت اول

## ۴-۲- بهینه سازی از نظر سرعت و فرکانس کاری

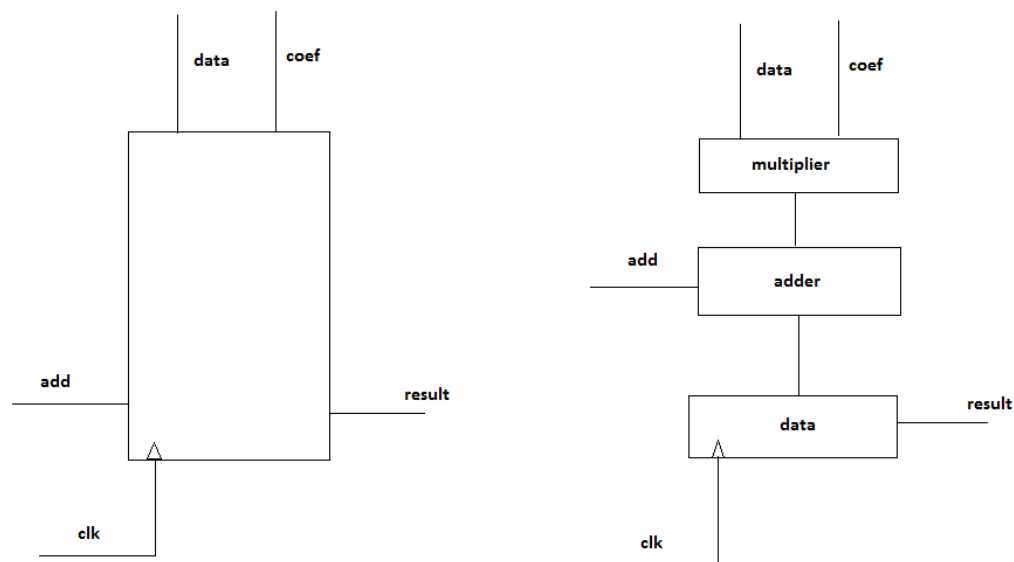
فرمول برای محاسبه خروجی:

$$f(x_i) = \sum_{n=0}^{18} (c_n * x_{i-n})$$

نحوه عملکردن پایپ لاین به این صورت است که در هر بار دریافت ورودی تمام حاصل ضرب های این ورودی در ثابت های تعیین شده بدست می آید بنابراین پس از ورود  $x_i$  تمام مقادیر  $c_0x_i$  و  $c_1x_i$  و  $c_2x_i$  و ... و  $c_{18}x_i$

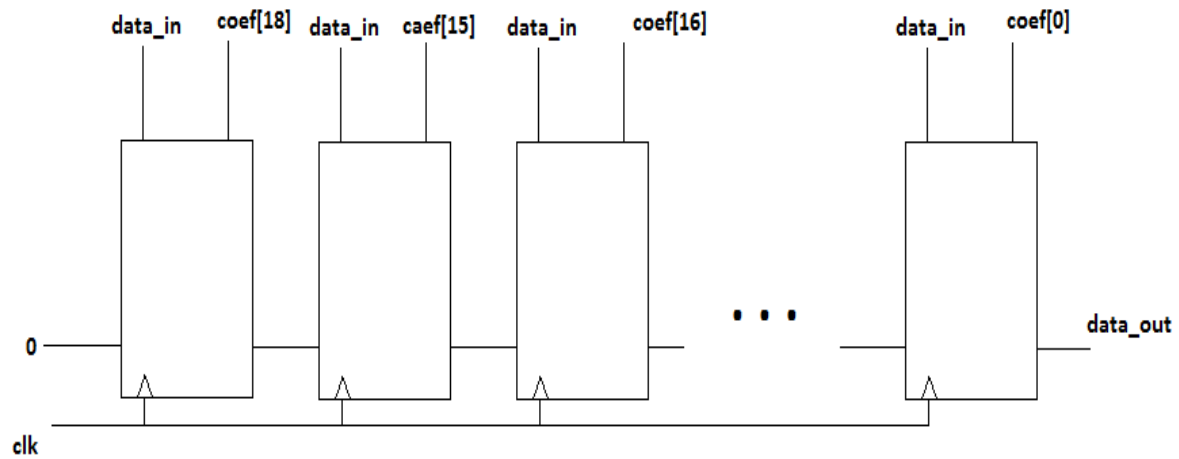
پس از زمان انجام یک ضرب در دسترس است و ما مقدار  $c_0x_i$  را برای محاسبه  $f(x_i)$  و مقدار  $c_1x_i$  را برای محاسبه  $f(x_{i+1})$  و  $\dots$  و مقدار  $c_{18}x_i$  را برای محاسبه  $f(x_{i+18})$  نیاز داریم. ۴ - ۵ نحوه کار این پایپ لاین را نشان می دهد

ابتدا ماژول زیر را طراحی می کنیم:



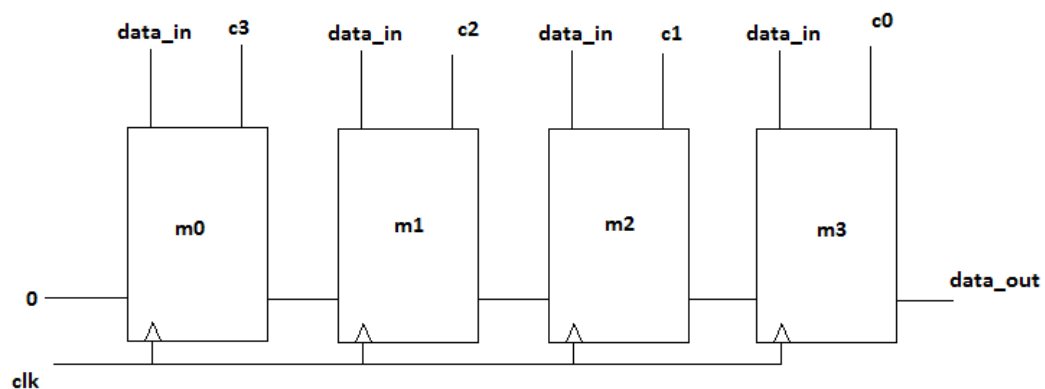
شکل ۴ - ۲ - ۱

حال کافی است با ۱۹ بار instance گرفتن از ماژول این فیلتر را بسازیم. (شکل ۴ - ۲ - ۲):



شکل ۴ - ۲ - ۲

و اینگونه در هر کلاک یک data خارج می شود. اگر داده  $x_i$  در یک کلاک وارد شود مقدار  $f(x_i)$  در کلاک بعدی از آن خارج می شود. در هر کلاک ۱۹ عملیات ضرب موازی و ۱۹ عملیات جمع موازی انجام می شود این در حالی است که در حالت بدون پایپ لاین در هر کلاک ۱۹ عملیات ضرب موازی و ۱۹ عملیات جمع متوالی بعد از آن است که باعث کمتر شدن فرکانس کلاک میشود. برای درک بهتر این فیلتر را با ۴ instance درست می کنیم (شکل ۴ - ۲ - ۳)؛



شکل ۴ - ۲ - ۳

نتایج بدست آمده به صورت زیر است ؛

کلاک	Data_in	مقدار m0	مقدار m1	مقدار m2	مقدار m3	Data_out
۱	$x_0$	$C3X0$	–	–	–	–
۲	$x_1$	$C3X1$	$C3X0 + C2X1$	–	–	–
۳	$x_2$	$C3X2$	$C3X1 + C2X2$	$C3X0 + C2X1 + C1X2$	–	–
۴	$x_3$	$C3X3$	$C3X2 + C2X3$	$C3X1 + C2X2 + C1X3$	$C3X0 + C2X1 + C1X2 + C0X3$	–
۵	$x_4$	$C3X4$	$C3X3 + C2X4$	$C3X2 + C2X3 + C1X4$	$C3X1 + C2X2 + C1X3 + C0X4$	<b>F(X3)</b>
۶	$x_5$	$C3X5$	$C3X4 + C2X5$	$C3X3 + C2X4 + C1X5$	$C3X2 + C2X3 + C1X4 + C0X5$	<b>F(X4)</b>

در ادامه جدول مصرف لاجیک ها و نمودار های مربوط به قسمت دوم فاز اول آورده شده است. ماکزیمم فرکانس پالس ساعت بدست آمده برای این فاز 306.063 مگاهرتز می باشد).

#### HDL Synthesis Report

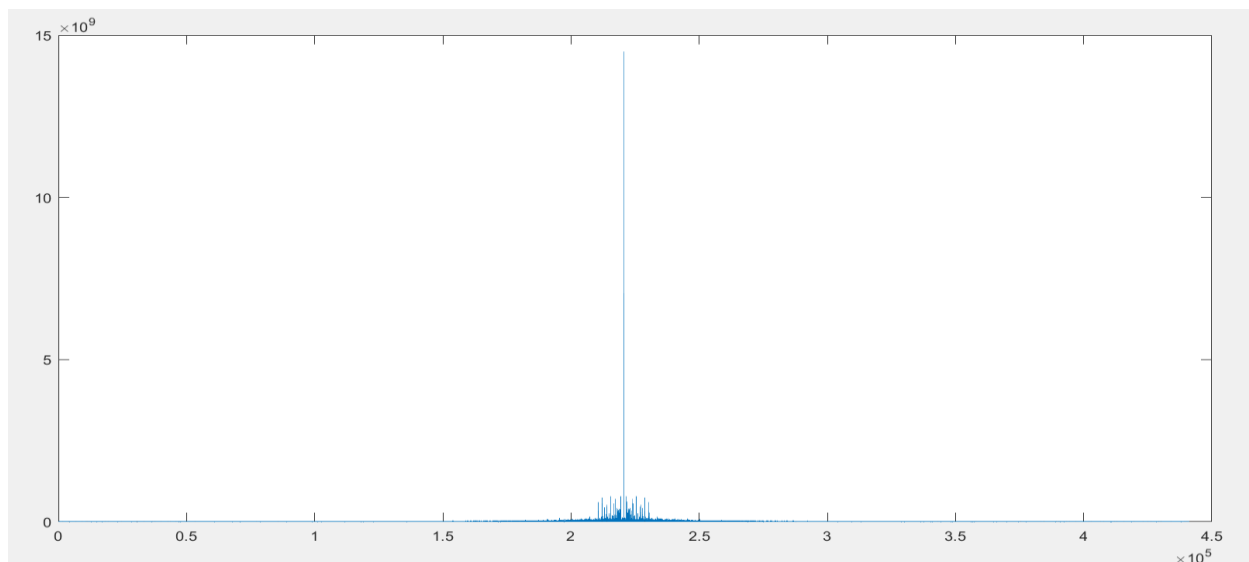
##### Macro Statistics

# Multipliers	: 10
16x16-bit multiplier	: 10
# Adders/Subtractors	: 18
36-bit adder	: 18
# Registers	: 1
664-bit register	: 1

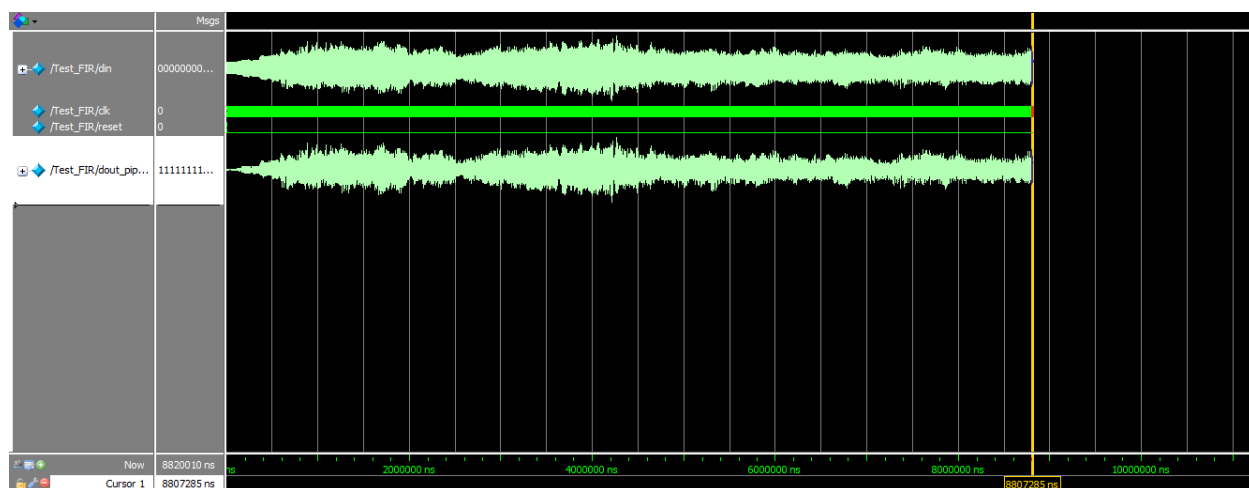
شکل ۴ - ۲ - ۴ : گزارش منابع مصرفی فاز دوم قسمت دوم

Device Utilization Summary (estimated values)				[1]
Logic Utilization	Used	Available	Utilization	
Number of Slice Registers	533	11440	4%	
Number of Slice LUTs	566	5720	9%	
Number of fully used LUT-FF pairs	511	588	86%	
Number of bonded IOBs	34	102	33%	
Number of BUFG/BUFGCTRLs	1	16	6%	
Number of DSP48A1s	10	16	62%	

شکل ۴ - ۲ - ۵ : جدول Device Utilization برای قسمت دوم



شکل ۴ - ۲ - ۶: نمودار حوزه فرکانس بخش اول فاز دوم: همانگونه که مشاهده می شود دو خط نشانگر نویز حذف شده اند.



شکل ۴ - ۲ - ۷: نمودار آنالوگ حاصل از خروجی فاز دوم قسمت دوم



## ۵ - فاز سوم

در فاز سه شبیه سازی پروژه با استفاده هسته FIR Compiler v.5 و در دو سطح بهینه سازی انجام شد. این هسته طراحی شده توسط کمپانی Xilinx است و نمای کلی آن بدین شکل است:



شکل ۵ - ۱ : شمای کلی هسته FIR Compiler v.5

din: داده ورودی ۱۶ بیتی سریال که طبق نرخ نمونه برداری (در این پروژه ۴۴۱۰۰ هرتز) وارد ماژول می شود.

clk: پالس ساعت (در این پروژه ۵۰ مگاهرتز)

dout: خروجی ۱۶ بیتی سریال که طبق نرخ نمونه برداری (در این پروژه ۴۴۱۰۰ هرتز) از ماژول خارج می شود.

rfd: سیگنال خروجی یک بیتی که هر گاه ماژول آماده دریافت ورودی باشد، یک می شود.

rdy: سیگنال خروجی یک بیتی که هر گاه ماژول یک خروجی را آماده کند، یک می شود.

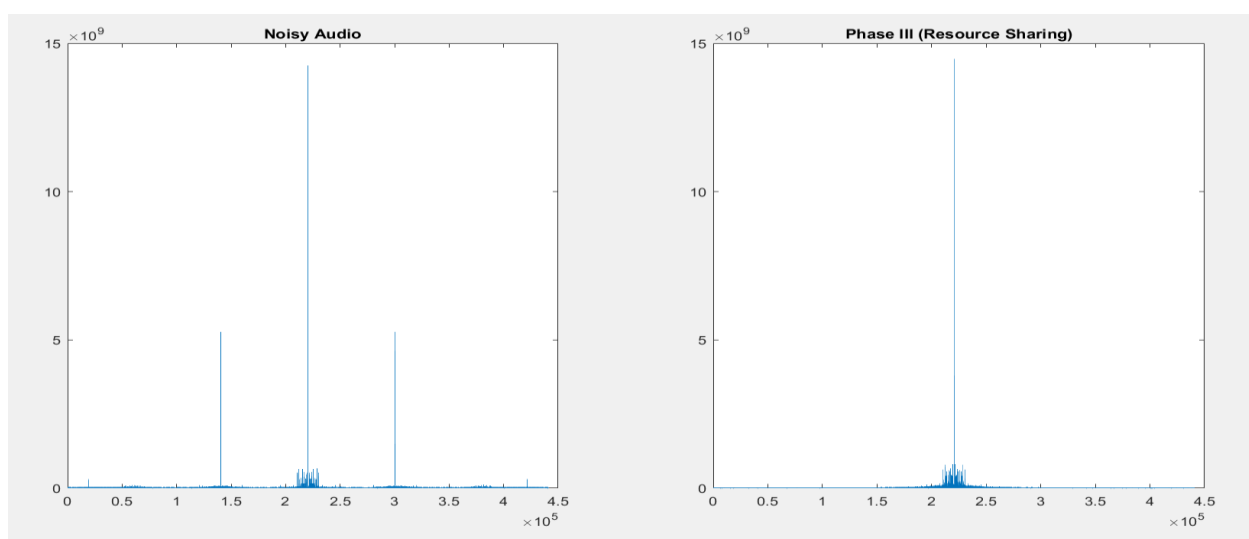
در این فاز دو کور به شکل های زیر ساخته شده و مورد استفاده قرار گرفتند:

### ۵ - ۱ - بهینه سازی فضا

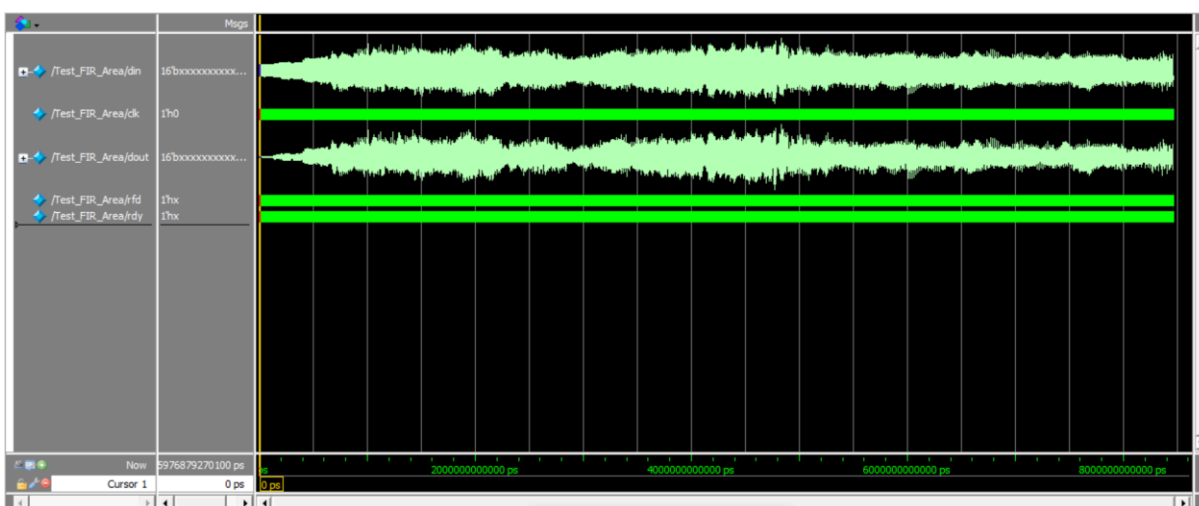
در این هسته بهینه سازی فضا و منابع مورد استفاده در FPGA مورد توجه قرار گرفت. پس از خروجی گیری ماکزیمم پالس ساعت قابل اعمال به این هسته، 346.929MHz به دست آمد. سایر نتایج اعم از خروجی و منابع استفاده شده در FPGA را در شکل های زیر مشاهده می کنید.

Device Utilization Summary (estimated values)				
Logic Utilization	Used	Available	Utilization	
Number of Slice Registers	148	11440	1%	
Number of Slice LUTs	141	5720	2%	
Number of fully used LUT-FF pairs	135	154	87%	
Number of bonded IOBs	35	102	34%	
Number of BUFG/BUFGCTRLs	1	16	6%	
Number of DSP48A1s	1	16	6%	

شکل ۵ - ۱ - ۱ : جدول Device Utilization فاز سوم در حالت بهینه سازی Resource Sharing



شکل ۵ - ۱ - ۲ : نمودار حوزه فرکانس حاصل از فاز سوم در حالت بهینه سازی Resource Sharing



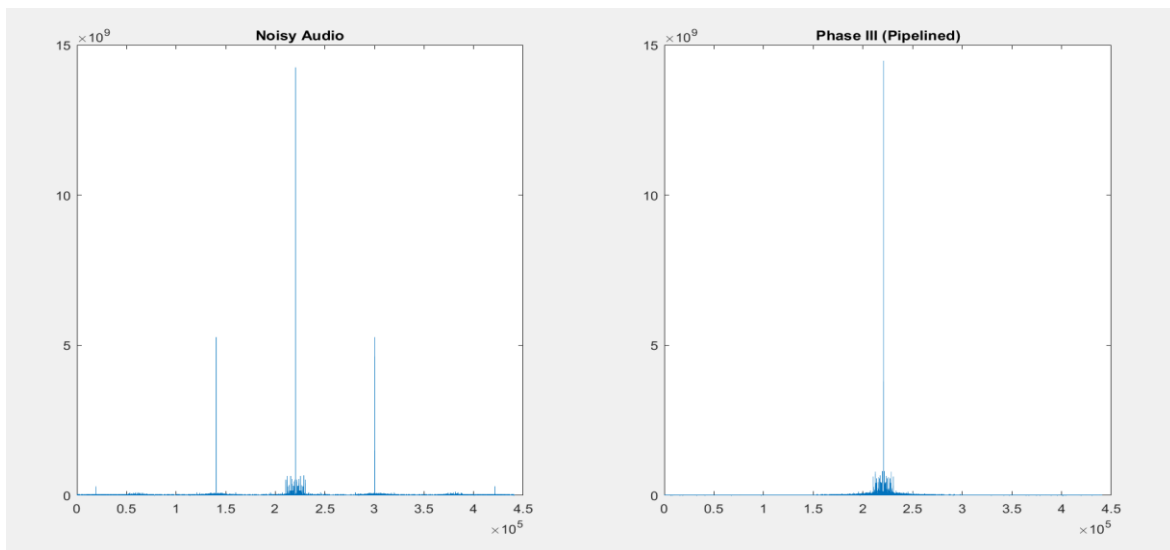
شکل ۵ - ۱ - ۳ : نمودار آنالوگ خروجی حاصل از فاز سوم در حالت بهینه سازی Resource Sharing

## ۵ - ۲ - بهینه سازی سرعت

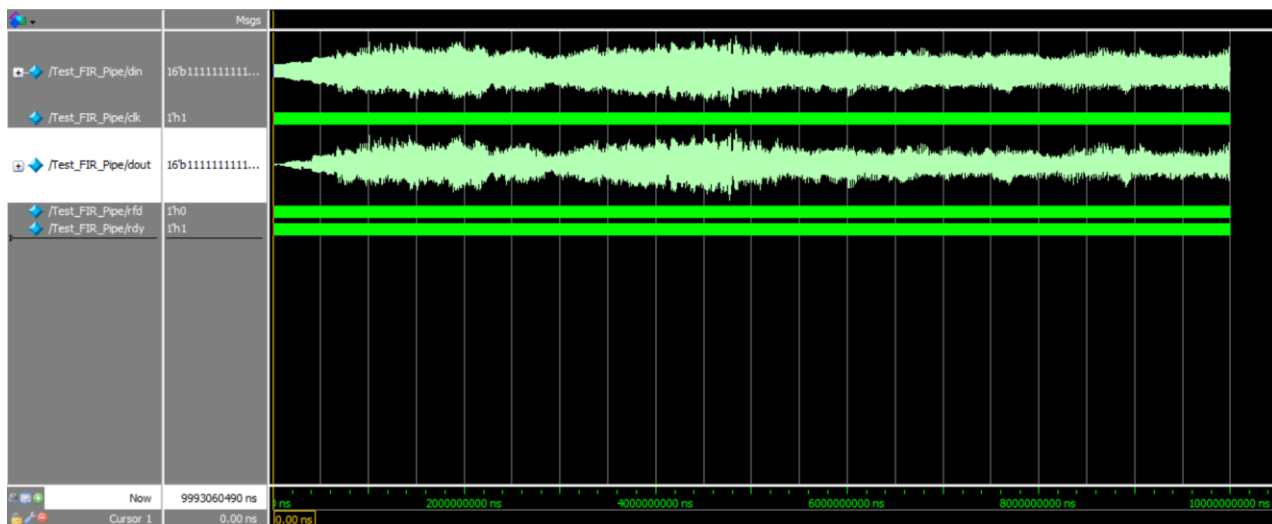
در این هسته بهینه سازی فضا و منابع مورد استفاده در FPGA مورد توجه قرار گرفت. پس از خروجی گیری ماکزیمم پالس ساعت قابل اعمال به این هسته، 452.396MHz به دست آمد. سایر نتایج اعم از خروجی و منابع استفاده شده در FPGA را در شکل های زیر مشاهده می کنید.

Device Utilization Summary (estimated values)				[...]
Logic Utilization	Used	Available	Utilization	
Number of Slice Registers	148	11440	1%	
Number of Slice LUTs	141	5720	2%	
Number of fully used LUT-FF pairs	135	154	87%	
Number of bonded IOBs	35	102	34%	
Number of BUFG/BUFGCTRLs	1	16	6%	
Number of DSP48A1s	1	16	6%	

شکل ۵ - ۲ - ۱ : جدول Device Utilization فاز سوم در حالت بهینه سازی Pipelining



شکل ۵ - ۲ - ۲ : نمودار حوزه فرکانس حاصل از فاز سوم در حالت بهینه سازی Pipelining



شکل ۵ - ۲ - ۳ : نمودار آنالوگ خروجی حاصل از فاز سوم در حالت بهینه سازی Pipelining

## ۶ - نتیجه گیری و جمع بندی

با توجه به مواردی که در بخش طرح مسئله به آن اشاره کردیم در این بخش نمودار و اطلاعات بدست آمده از سه بخش را با هم مقایسه می کنیم.

در فاز اول با طراحی ساده که در ورژن ۵ داشتیم به فرکانس 110.082 MHz رسیدیم که در بخش دوم با روش Pipelining توانستیم به فرکانس کاری 306.063 MHz برسیم ، که نحوه طراحی در بخش مربوط ذکر شد. در فاز سوم از هسته ی پیش طراحی شده استفاده کردیم که سرعت در حالت Pipelining به 452.396MHz رسید. همچنین در فاز اول مصرف منابع بطور خلاصه چنین بود :

Multipliers : 19

Adders/Subtractors : 19

Registers : 5

اما در فاز دوم قسمت بهینه سازی منابع مصرفی مصرف منابع بطور خلاصه چنین بود :

RAMs : 1

Multipliers : 1

Adders/Subtractors : 2

Registers : 4

Multiplexers : 2

که تغییر قابل ملاحظه ای در منابع مصرفی محسوب می شود اما این تغییر فرکانس کاری را به 130.625 MHz رساند. در فاز سوم قسمت Resource Sharing فرکانس کاری 346.929MHz بود.