دانشکده ی برق و رباتیک

پروژه پایانی کارشناسی ارشد

پیاده سازی فیلترهای دیجیتال بازگشتی LWD با حساسیت و نویز پایین، برروی پردازشگرDSP

دانشجو :

صمد ذبیحی آزادبنی

اساتید راهنما:

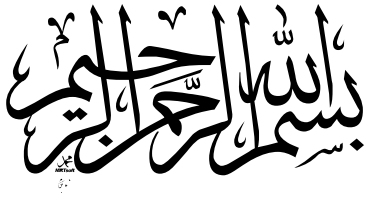
دکتر امیدرضا معروضی

دکتر هادی گرایلو

استاد مشاور:

دکتر علی سلیمانی

تابستان ۹۱



پیوست شماره ۲

دانشگاه صنعتی شاهرود

دانشکده :

گروه :

پایان نامه کارشناسی ارشد آقای/ خانم .....

تحت عنوان: پیاده سازی فیلترهای دیجیتال بازگشتی LWD با حساسیت و نویز پایین، برروی پردازشگر DSP

در تاریخ ..... توسط کمیته تخصصی زیر جهت اخذ مدرک کارشناسی ارشد مورد ارزیابی و با درجه ..... مورد پذیرش قرار گرفت.

اساتید راهنما	امضاء	اساتید مشاور	امضاء
نام و نام خانوادگی :		نام و نام خانوادگی :	
نام و نام خانوادگی :		نام و نام خانوادگی :	

اساتید داور	امضاء	نماینده تحصیلات تکمیلی	امضاء
نام و نام خانوادگی :		نام و نام خانوادگی :	
نام و نام خانوادگی :			
نام و نام خانوادگی :			
نام و نام خانوادگی :			

دانشکده ی برق و رباتیک

پروژه پایانی کارشناسی ارشد

پیاده سازی فیلترهای دیجیتال بازگشتی LWD با حساسیت و نویز پایین، برروی پردازشگرDSP

دانشجو :

صمد ذبیحی آزادبنی

اساتید راهنما:

دکتر امیدرضا معروضی

دکتر هادی گرایلو

استاد مشاور:

دکتر علی سلیمانی

تابستان ۹۱

تقدیم به پدر و مادر عزیزم

تشکر و قدردانی

با سپاس فراوان از زحمات پدر و مادر عزیزم، که حمایتگر من در بسیاری از مسایل زندگی بودند.

تعهد نامه

اینجانب **صمد ذبیحی آزادبنی** دانشجوی دوره کارشناسی ارشد رشته **مهندسی برق الکترونیک گرایش دیجیتال** دانشکده برق و رباتیک دانشگاه صنعتی شاهرود نویسنده پایان‌نامه با عنوان: **پیاده سازی فیلترهای دیجیتال بازگشتی LWD با حساسیت و نویز پایین، برروی پردازشگرDSP** تحت راهنمائی **دکتر امید رضا معروضی** و **دکتر هادی گرایلو** متعهد می شوم.

- تحقیقات در این پایان نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است .
- در استفاده از نتایج پژوهشهای محققان دیگر به مرجع مورد استفاده استناد شده است .
- مطالب مندرج در پایان نامه تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است .
- کلیه حقوق معنوی این اثر متعلق به دانشگاه صنعتی شاهرود می باشد و مقالات مستخرج با نام « دانشگاه صنعتی شاهرود » و یا « Shahrood University of Technology » به چاپ خواهد رسید .
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان نامه تأثیرگذار بوده اند در مقالات مستخرج از پایان نامه رعایت می گردد.
- در کلیه مراحل انجام این پایان نامه ، در مواردی که از موجود زنده (یا بافتهای آنها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است .
- در کلیه مراحل انجام این پایان نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است اصل رازداری ، ضوابط و اصول اخلاق انسانی رعایت شده است .

تاریخ

امضای دانشجو

مالکیت نتایج و حق نشر

- کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج ، کتاب ، برنامه های رایانه ای ، نرم افزار ها و تجهیزات ساخته شده است) متعلق به دانشگاه صنعتی شاهرود می باشد . این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود .
- استفاده از اطلاعات و نتایج موجود در پایان نامه بدون ذکر مرجع مجاز نمی باشد.

چکیده

برای بهره بردن از خواص و ویژگی های الگوریتم‌ها، نیازمند پیاده سازی آن ها در عمل هستیم. در میان الگوریتم های مختلف، الگوریتم های دیجیتال از اهمیت فراوانی برخوردارند. این اهمیت بدلیل برتری‌های سیستم‌های دیجیتال نسبت به آنالوگ است. پیاده سازی الگوریتم‌های دیجیتال در عمل یکی از چالش های مهم فن آوری روز است. در این میان سخت افزارهایی مانند پردازنده های سیگنال دیجیتال ، FPGA و ... دارای کاربردهای وسیعی می باشند.

در این پایان نامه، بروی یکی از مهمترین ساختارهای فیلتر دیجیتال، با نام فیلترهای دیجیتال موج شبکه ای[1] تمرکز شده است. با استفاده از شبیه سازی در محیط نرم افزار MATLAB، مقایسه ای بین این کلاس از فیلترها و چند کلاس مهم دیگر از فیلترهای دیجیتال، از لحاظ سرعت و حساسیت انجام شد.

یک فیلتر LWD آبشاری دو طبقه با استفاده از زبان اسمبلی پردازنده سیگنال TMS320VC5505 پیاده سازی گردید و در محیط نرم افزار CodeComposer Studio شبیه سازی شد. سپس با بهره بردن از دستورات پردازش موازی، برنامه ی اسمبلی بهینه سازی شد و سرعت پردازش برنامه اسمبلی در حالت‌های قبل و بعد از بهینه سازی مقایسه گردید.

با استفاده از ماژول ارزیابی TMS320VC5505[2]، برنامه ی بهینه شده بصورت سخت افزاری پیاده سازی شد و نتایج حاصل از پردازنده سیگنال TMS320VC5505 بروی اسیلوسکوپ نمایش داده شد. این پردازنده سیگنال محصول شرکت Texas Instrument می‌باشد.

[1] LWDF:Lattice Wave Digital Filter
[2] TMS320VC5505 Evaluation Module

فهرست مطالب

عنوان **صفحه**

فهرست اصطلاحات اختصاری ..... **ث**

فهرست جداول ..... **ج**

فهرست شکل‌ها ..... **ح**

فصل اول: فیلترهای دیجیتال

۱-۱- معرفی فیلترهای دیجیتال **۲**

۱-۲- مراحل طراحی فیلتر دیجیتال **۲**

۱-۳- تخمین فیلتر دیجیتال **۵**

۱-۳-۱- توابع تخمین و فواصل طراحی **۵**

۱-۳-۲- نوع تابع تبدیل **۵**

۱-۳-۳- معیارهای تخمین **۶**

۱-۳-۴- روش‌های تخمین **۷**

۱-۳-۴-۱- روش‌های تخمین برای فیلترهای FIR **۷**

۱-۳-۴-۲- روش‌های تخمین برای فیلتر های IIR **۷**

۱-۴- تحقق فیلترهای دیجیتال **۸**

۱-۵- ملاحظات در پیاده سازی فیلتر دیجیتال **۹**

۱-۵-۱- نمایش ضرایب **۱۰**

۱-۵-۱-۱- نمایش مکمل ۲ ..... **۱۰**

۱-۵-۱-۲- نمایش توان های ۲ علامتدار **۱۱**

۱-۵-۲- تاثیر طول کلمات محدود بر فیلترهای دیجیتال **۱۲**

۱-۵-۲-۱- ساختارهایی با حساسیت پایین ۱۴
۱-۵-۲-۲- روش‌های بهینه سازی ۱۴
۱-۵-۲-۳- روش‌های تخمین..... ۱۴

فصل دوم: فیلترهای LWD

۲-۱- معرفی فیلترهای LWD ۱۷
۲-۲- فیلترهای LWD آبشاری..... ۱۹
۲-۳- تابع انتقال فیلتر LWD..... ۱۹
۲-۴- مساله بهینه سازی..... ۲۱
۲-۴-۱- کوانتیزه کردن ضرایب..... ۲۳
۲-۴-۲- حل مساله بهینه سازی..... ۲۴

فصل سوم: شبیه سازی و مقایسه چند ساختار مهم

۳-۱- مقایسه چند ساختار مهم از فیلترهای دیجیتال..... ۲۶
۳-۱-۱- فیلتر LWD مستقیم..... ۲۷
۳-۱-۲- فیلتر LWD آبشاری دو طبقه..... ۲۸
۳-۱-۳- فیلتر WINDOW-BASED FIR ۲۸
۳-۱-۴- فیلتر BUTTERWORTH IIR ۲۹
۳-۲- شبیه سازی ساختارهای معرفی شده..... ۳۰
۳-۲-۱- آنالیز زمانی ۳۳
۳-۲-۲- آنالیز حساسیت..... ۳۴

فصل چهارم: پیاده سازی نرم افزاری

۴-۱- پردازنده‌های سیگنال دیجیتال ۳۸
۴-۲- پردازنده سیگنال TMS320C5000..... ۳۹
۴-۳- ماژول **ارزیابی**TMS320VC5505..... ۴۰
۴-۴- برنامه CODE COMPOSER STUDIO..... ۴۲
۴-۵- زبان اسمبلی ۴۴
۴-۶- دستورات MNEMONIC ۴۴
۴-۷- پیاده سازی فیلتر LWD آبشاری دوطبقه با زبان اسمبلی..... ۴۴
۴-۸- مقایسه نتایج شبیه سازی در محیط MATLAB و محیط CCS V3.3..... ۴۶
۴-۹- بهینه سازی برنامه ی اسمبلی..... ۴۸
۴-۹-۱- استفاده از پردازش موازی..... ۴۸
۴-۱۰- مقایسه سرعت فیلتر قبل و بعد از بهینه سازی ۴۹

فصل پنجم: پیاده سازی سخت افزاری

۵-۱- پیاده سازی سخت افزاری فیلتر بهینه شده..... ۵۲
۵-۲- مدار تبدیل ۵۲
۵-۲-۱- مبدل آنالوگ به دیجیتال ۵۳
۵-۲-۲- مبدل دیجیتال به آنالوگ ۵۴
۵-۳- پیاده سازی سخت افزاری بلادرنگ..... ۵۴
۵-۴- آنالیز حساسیت در پیاده سازی سخت افزاری ۶۱
۵-۴-۱- آنالیز حساسیت فیلتر LWD آبشاری دوطبقه..... ۶۲
۵-۴-۲- محاسبه انحراف خروجی در حالت‌های مختلف..... ۶۸

فهرست منابع ۶۹

فهرست اصطلاحات اختصاری

DSP	Digital Signal Processing
ADC	Analog to Digital Converter
LWDF	Lattice Wave Digital Filter
LTI	Linear-Time invariant
SPT	Signed Power-of-Two
CSD	Canonic Signed-Digit
VLSI	Very Large Scale Integration
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
CCS	Code Composer Studio
USB	Universal Serial Bus
EVM	Evaluation Module
TI	Texas Instruments
MMC	Multi Media Card
EEPROM	Electrically Erasable Programmable Read Only Memo
UART	Universal Asynchronous Receiver
LCD	liquid crystal display
DAC	Digital to Analog Converter
GPIO	General Purpose Input Output

فهرست جداول

عنوان **صفحه**

جدول ۱ ضرایب فیلتر پایین گذر LWD مستقیم ۲۷

جدول ۲ ضرایب فیلتر پایین گذر LWD آبشاری دو طبقه ۲۸

جدول ۳ ضرایب فیلتر پایین گذر Window-based FIR ۲۹

جدول ۴ ضرایب فیلتر پایین گذر Butterworth IIR ۳۰

جدول ۵ نتایج زمان اجرای هر فیلتر در آنالیز زمانی ۳۴

جدول ۶ خطای خروجی فیلترها درآنالیز حساسیت، به ازای طول بیت‌های مختلف برای قسمت اعشاری ضرایب ۳۵

جدول ۷ تعداد کلاک‌های مورد نیازی جهت پردازش هزار نمونه ی قبل و بعد از بهینه سازی ۴۹

جدول ۸ خروجی فیلتر در حالت بدون محدودیت ۶۳

جدول ۹ ضرایب فیلتر در حالت محدودیت ۸ بیت برای ضرایب ۶۴

جدول ۱۰ ضرایب فیلتر در حالت محدودیت ۶ بیت برای ضرایب ۶۵

جدول ۱۱ ضرایب فیلتر در حالت محدودیت ۴ بیت برای ضرایب ۶۶

جدول ۱۲ ضرایب فیلتر در حالت محدودیت ۳ بیت برای ضرایب ۶۷

جدول۱۳ خطای خروجی فیلترها به ازای محدودیت های مختلف برای طول بیت های اعشاری ۶۸

فهرست شکل‌ها

عنوان	صفحه
شکل(۲ـ۱) اتصال موازی بین دو فیلتر تمام گذر	۱۸
شکل(۲ـ۲) اتصال آبشاری فیلتر LWD	۲۰
شکل(۳ـ۱) شکل موج سیگنال ورودی	۳۱
شکل(۳ـ۲) خروجی فیلتر Window-based FIR به ازای ورودی نشان داده شده در شکل (۴ـ۱)	۳۱
شکل(۳ـ۳) خروجی فیلتر Butterworth IIR به ازای ورودی نشان داده شده در شکل (۴ـ۱)	۳۲
شکل(۳ـ۴) خروجی فیلتر LWD آبشاری دو طبقه به ازای ورودی نشان داده شده در شکل (۴ـ۱)	۳۲
شکل(۳ـ۵) خروجی فیلتر فیلتر LWD مستقیم به ازای ورودی نشان داده شده در شکل (۴ـ۱)	۳۳
شکل(۳ـ۶) نمودار خطای خروجی فیلترها نسبت به طول بیت های قسمت اعشاری ضرایب	۳۵
شکل(۴ـ۱) ماژول ارزیابی TMS320VC5505	۴۰
شکل(۴ـ۲) آیکون های مربوط به برنامه ی CCS v3.3	۴۲
شکل(۴ـ۳) محیط نرم افزار CCS v3.3	۴۳
شکل(۴ـ۴) پیاده سازی فیلتر LWD آبشاری دوطبقه با زبان اسمبلی و شبیه سازی آن در CCS v3.3	۴۵
شکل(۴ـ۵) سیگنال ورودی (بالا) و خروجی (پایین) حاصل از شبیه سازی در محیط CCS v3.3	۴۶
شکل(۴ـ۶) نتایج حاصل از شبیه سازی در محیط MATLAB با ، و شبیه سازی در محیط CCS	۴۷

شکل(۵-۱) مدار تبدیل ۵۳

شکل(۵-۲) مدار مبدل آنالوگ به دیجیتال ۵۳

شکل(۵-۳) مدار مبدل مبدل دیجیتال به آنالوگ ۵۴

شکل(۵-۴) ماژول ارزیابی TMS320VC5505 و اتصالات مربوطه ۵۵

شکل(۵-۵) فعال کردن حالت بلادرنگ در برنامه CCS ۵۷

شکل(۵-۶) سیگنال خروجی در برنامه CCS به ازای ورودی مربعی ۵ هرتز ۵۷

شکل(۵-۷) پیاده سازی سخت افزاری فیلتر LWD آبشاری دوطبقه ۵۸

شکل(۵-۸) ورودی و خروجی فیلتر به ازای سیگنال پالس ۵ هرتز ۵۹

شکل(۵-۹) خروجی فیلتر به ازای ورودی سینوسی ۵ هرتز ۵۹

شکل(۵-۱۰) خروجی فیلتر به ازای ورودی سینوسی ۲۰ هرتز ۶۰

شکل(۵-۱۱) خروجی فیلتر به ازای ورودی سینوسی ۸۰ هرتز ۶۰

شکل(۵-۱۲) خروجی فیلتر به ازای ورودی سینوسی ۲۰۰ هرتز ۶۱

شکل(۵-۱۳) ورودی و خروجی فیلتر در حالت بدون محدودیت ۶۴

شکل(۵-۱۴) ورودی و خروجی فیلتر در حالت محدودیت ۸ بیت برای ضرایب ۶۵

شکل(۵-۱۵) ورودی و خروجی فیلتر در حالت محدودیت ۶ بیت برای ضرایب ۶۶

شکل(۵-۱۶) ورودی و خروجی فیلتر در حالت محدودیت ۴ بیت برای ضرایب ۶۷

شکل(۵-۱۷) ورودی و خروجی فیلتر در حالت محدودیت ۳ بیت برای ضرایب ۶۸

فصل اول:

فیلترهای دیجیتال

۱-۱- معرفی فیلترهای دیجیتال

فیلترهای دیجیتال عضو مهمی از کلاس سیستم های LTI هستند که برای تغییر دادن مشخصات حوزه ی فرکانس سیگنال ورودی ساخته می‌شود تا با تغییر آنها مشخصات مطلوب حاصل گردد. مشخصه های یک فیلتر دیجیتال علی بطور کامل توسط پاسخ ضربه ی آن h(t) قابل توصیف است که $H(z)$ را تابع انتقال و $H(e^{jw})$ را پاسخ فرکانسی آن می‌نامند. همچنین رابطه ی بین ورودی و خروجی سیستم در حوزه ی زمان توسط یک معادله ی دیفرانسیل قابل توصیف است [۱].

۱-۲- مراحل طراحی فیلتر دیجیتال

بطور کلی طراحی یک فیلتر دیجیتال در سه مرحله [۲-۷] انجام می‌شود:

۱- تخمین

۲- تحقق

۳- پیاده سازی

مرحله ی تخمین فرایندی است که در آن تابع تبدیلی گسسته در زمان تولید می‌شود که خصوصیات مورد انتظار از فیلتر مورد نظر را دارا است. معمولا این خصوصیات شامل اندازه و فاز و وضعیت احتمالی در حوزه ی زمان است.

روش های موجود برای حل این مساله ی تخمین به دو بخش مستقیم و غیر مستقیم تقسیم می‌شود [۲, ۸]. در روش های مستقیم، طراحی تابع انتقال مستقیما در حوزه ی z صورت می‌گیرد. روش های غیر مستقیم بروی مفهوم تابع انتقال زمان پیوسته ای که معیارهای مورد انتظار را دارد بنا شده است. پس از طراحی تابع انتقال زمان پیوسته ی مربوطه، آنرا به تابع انتقال گسسته در زمان تبدیل می‌کنیم. در طراحی فیلترهای FIR

همیشه از متدهای مستقیم استفاده می‌گردد، در حالی که فیلترهای IIR را می‌توان با استفاده از هر دو روش مستقیم و غیر مستقیم طراحی کرد [۲, ۴, ۸].

تابع انتقال یک سیستم، رفتار یک سیستم را بیان می‌کند و رابطه ی بین ورودی و خروجی آن را توصیف می‌نماید. در هر صورت تابع انتقال به ما جزییاتی در مورد روند محاسبه ی خروجی نمی‌دهد. سنتز یک فیلتر دیجیتال روندی است که در آن ما تابع انتقال را به یک توصیف ساختاری یا یک گراف جریان سیگنال تبدیل می‌کنیم که در آن روند محاسبات و ترتیب محاسبات به وضوح مشخص می‌شود. این مرحله معمولا مرحله ی تحقق نامیده می‌شود. در واقع این ساختار یکه نیست و برای یک تابع انتقال می‌توان بیشمار نمودار جریان-سیگنال تولید کرد. وقتی همه ی محاسبات بصورت دقت-نامحدود[۳] باشد، یعنی محدودیتی از لحاظ تعداد اعداد پس از اعشار و یا طول کلمات حافظه نباشد، اینگونه تحقق بخشیدن‌ها همگی از لحاظ ریاضی یکسان هستند. اما در مورد محاسبات دقت-محدود[۴] این گونه تحقق بخشیدن‌ها مشخصه هایی متفاوت دارند، و تحقق بخشیدن های مختلف، معمولا از لحاظ عددی با هم برابر نیستند [۱].

مانند متدهای تخمین، روش های تحقق نیز می‌توانند به دو گروه مستقیم و غیر مستقیم تقسیم شوند. در روش مستقیم، تابع انتقال و مشخصه های سیستم به فرمی تبدیل می‌شوند که ساختار داخلی سیستم قابل تشخیص باشد. در متدهای غیر مستقیم، فیلتر دیجیتال از یک دسته از فیلتر های آنالوگ مرجع مشتق می‌گردد که این فیلتر های آنالوگ مرجع بصورت ساختاری با فیلتر مورد نظرمان ارتباط دارد [۱].

در پیاده سازی‌هایی که از VLSI استفاده می‌شود، طراح اغلب به تحقق هایی علاقه مند است که پیچیدگی محاسباتی کمتری داشته باشند و در باب محاسبات ممیز محدود حساسیت نداشته و از ساختار و اجزا استاندارد تشکیل شده باشند [۱].

[3] Infinite-precision
[4] Finite-precision

الگوریتم های فیلترهای دیجیتال بر اساس کاربرد می‌توانند به راه های گوناگونی پیاده سازی گردند. این پیاده سازی‌ها عموما به دو گونه ی نرم افزاری و سخت افزاری انجام می‌شود. در حالت نرم افزاری، پیاده سازی با استفاده از یک برنامه ی کامپیوتری است که فیلتر را روی یک کامپیوتر معمولی یا یک پردازنده ی سیگنال شبیه سازی می‌کند. در حالت سخت افزاری، فیلتر به اجزای کوچک تجزیه شده و هر جز توسط یک قطعه ی سخت افزاری پیاده سازی می‌شود. در این نوع پیاده سازی، ضرب کننده قطعه ای بسیار گران است، بنابراین برای صرفه جویی در هزینه ی پیاده سازی، معمولا بجای ضرب کننده‌ها از توالی ای از شیفت های بیتی و جمع‌ها (یا تفریق ها) استفاده می‌کنیم. انتخاب فرم پیاده سازی بستگی به کاربرد دارد. در کاربرد های بلادرنگ که در آن پردازش باید با نرخ بالایی صورت گیرد (برای مثال کاربرد های مخابراتی و لوازم ویدیویی) استفاده از پیاده سازی سخت افزاری اجباری است. در کاربرد هایی غیر بلادرنگ، جایی که داده ابتدا ذخیره شده و سپس پردازش می‌شوند، پیاده سازی نرم افزاری در بسیاری از موارد می‌تواند پاسخگو باشد. علاوه براین، در کاربرد هایی که انعطاف بیشتری مورد نیاز است، پیاده سازی نرم افزاری باید مورد توجه قرار گیرد [۱].

سه مرحله ی بالا از هم مستقل نیستند. در کاربرد های بلادرنگ، سرعت و قیمت سخت افزار معیارهای اصلی اند. در این مورد، یافتن یک جواب بهینه کار بسیار سختی است، زیرا این معیارها تحت تاثیر تمام تصمیماتی است که در طول مراحل طراحی می‌گیریم. بنابراین مراحل طراحی در بالا ممکن است چندین بار تکرار گردد، تا وقتی که فیلتر تمام مشخصات مورد انتظار را بدست آورد. در زیر به نکته هایی اشاره می‌کنیم که باعث می‌شود فیلتر طراحی شده دارای سرعت بالاتر و پیچیدگی سخت افزاری کمتر شود [۱].

۱-۳- تخمین فیلتر دیجیتال

همانگونه که گفته شد، مراحل مختلف طراحی فیلتر بسیار بهم وابسته اند. نوع تابع انتقال انتخابی و فرم تحقق آن بشدت در پیچیدگی فیلتر تاثیرگذار است. از طرف دیگر این انتخاب، تابع تخمین را در مساله ی تخمین مربوطه تعیین می‌کند [۱].

۱-۳-۱- توابع تخمین و فواصل طراحی

در بسیاری از موارد، پاسخ مورد انتظار به وسیله ی پارامتر هایی در حوزه ی فرکانس تعیین می‌شود. معمولا این مشخصه‌ها شامل پاسخ اندازه یا پاسخ فاز یا هر دو هستند. پاسخ اندازه معمولا با تعیین دو ناحیه تعریف می‌شود، یک ناحیه ی فرکانسی به نام باند گذر که در این ناحیه اجزا سیگنال حفظ می‌شود و ناحیه ای دیگر بنام باند توقف که در آن اجزا سیگنال تضعیف می‌گردد. در کاربرد هایی که در آن شکل موج سیگنال باید حفظ شود، پاسخ فاز باید در باند گذر بصورت خطی باشد. در موارد خاص ممکن است شرایط حوزه ی زمان در دست باشد، مانند زمانی که بخواهیم فیلترهای نایکوییست طراحی کنیم که برخی از مقادیر پاسخ ضربه ی آن‌ها محدود به داشتن مقدار صفر است. همچنین کاربرد هایی وجود دارد که در آن شرایط مشخصی برای پاسخ پله مد نظر است [۹،۷].

۱-۳-۲- نوع تابع تبدیل

در مرحله ی تخمین، کلاس مناسبی از فیلترها انتخاب می‌شود که دارای پاسخی نزدیک به پاسخ مورد انتظار باشد. انتخاب اولیه باید بین دو گزینه ی فیلتر های IIR و FIR صورت گیرد. پس از آن، کلاس مناسب را برای فیلتر مورد نظرمان تعیین می‌کنیم. برای داشتن بهره ی بالاتر یا حساسیت کمتر در ساختار های IIR یا FIR، محدودیت های مضاعفی در انتخاب تابع انتقال خواهیم داشت. برای مثال، در فیلترهای LWD، ضرایب

صورت بصورت مشخص توسط ضرایب مخرج تعیین می‌گردند. در این موقعیت، طراحی تابع انتقال و تحقق فیلتر قابل جدا سازی نیستند، و ساختار مورد انتظار فیلتر، کلاس فیلتر مورد نظر را تعیین می‌کند [۱].

۱-۳-۳- معیارهای تخمین

برای یافتن بهترین عضو از کلاس انتخاب شده برای فیلتر، لازم است که بتوانیم خطا را اندازه گیری کنیم تا متوجه شویم که پاسخ تخمینی چقدر به پاسخ مطلوب نزدیک است. متداولترین روش‌ها برای محاسبه ی خطا، روش تخمین چبیشر[۵] یا کمینه-بیشینه[۶] و روش تخمین کوچکترین مجذور[۷] است [۴،۷]. در بسیاری از موقعیت ها، بیشینه ی مقدار مجاز برای خطا مشخص می‌شود. در این موقعیت، مساله ی اصلی یافتن فیلتری با کمترین پیچیدگی است که با معیارهای داده شده مطابقت داشته باشد. پس از آن، کار باقی مانده یافتن بهترین عضو در کلاس انتخاب شده برای فیلتر است که کمترین پیچیدگی را دارد و خطای اندازه گیری شده ی آن پایین است [۱].

از طرف دیگر، مخصوصا در پیاده سازی های اجرا شده با VLSI، یکی از مهمترین فاکتورها قیمت تمام شده برای پیاده سازی فیلتر است. میزان این هزینه وابسته به میزان پیچیدگی ساختار فیلتر می‌باشد. برای مثال، وابسته به تعداد تاخیرها، المان های ذخیره سازی، ضرب کننده‌ها و جمع کننده‌ها و طول کلمات مورد نیاز برای نمایش سیگنال و ضرایب است [۱۰]. هزینه ی بهینه سازی را می‌توان با بهینه سازی ضرایب فیلتر بروی فضای پارامتر های گسسته کاهش داد. یک روش موثر برای کاهش این هزینه آن است که ضرایب فیلتر را طوری انتخاب کنیم که با استفاده از توالی شیفت‌ها و جمع‌ها و تفریق‌ها قابل نمایش باشد. در این حالت به المان های ضرب کننده ی متداول نیاز نیست. بنابراین نتیجه می‌گیریم که مساله ی اصلی یافتن بهترین عضو از

[5] Chebysher
[6] Min-max
[7] Least Square

کلاس انتخابی برای فیلتر است که در آن تعداد جمع کننده‌ها و تفریق کننده‌ها کمینه باشند، بطوری که مشخصات سیستم همچنان با معیار های داده شده مطابقت داشته باشند [۱].

۱-۳-۴- روش های تخمین

۱-۳-۴-۱- روش های تخمین برای فیلترهای FIR

موثرترین الگوریتم برای طراحی یک فیلتر FIR با اندازه ی بهینه و با مشخصات اختیاری، استفاده از الگوریتم چندین معاوضه ی رمز[۸] است. معمول ترین راه برای پیاده سازی این الگوریتم همانی است که توسط پارکس[۹] و مکلین[۱۰] تهیه شده است [۱۱]. توسعه های بعدی صورت گرفته بروی الگوریتم رمز[۱۱] توسط مکلین، پارکس و رابینر[۱۲] صورت گرفته است [۱۲-۱۴]. این الگوریتم برای تهیه ی مستقیم طرح های بهینه شده برای بسیاری از انواع فیلتر های فاز خطی FIR مناسب است. همچنین، این الگوریتم‌ها قابلیت طراحی فیلتر هایی با چند باند گذر و باند قطع مختلف را دارد. بتازگی تلاش های بسیاری برای ایجاد الگوریتم رمز در جهت حل مسایل مختلف تخمین چابیشر صورت گرفته است [۱۵-۱۷]. هر چند که برای همگرایی این الگوریتم هادر جهت تولید یک جواب بهینه تضمینی وجود ندارد.

۱-۳-۴-۲- روش های تخمین برای فیلتر های IIR

در طول ۶۰ سال گذشته، مساله ی بهینه سازی برای فیلتر های انالوگ موضوع فراگیر در تحقیقات شد و طی آن روش های بهینه سازی کارامدی ابدا گردید. این روش های تابع انتقال سیستم های حلقه بسته ی پیوسته در زمان را به خوبی توصیف نمودند. تابع انتقال پیوسته در زمان را در فرم حلقه بسته به خوبی توصیف

[8] Remez Multiple Exchange
[9] Parks
[10] McClellan
[11] Remez
[12] Rabiner

نموند. تابع انتقال پیوسته در زمان را می‌توان به تابع انتقال گسسته در زمان تبدیل نمود. تخمین های موجود برای تابع انتقال را می‌توان برای حل مسایل تخمین فیلتر های دیجیتال بکار برد. در اینجا بصورت خاص بروی طراحی فیلتر پایین گذر تمرکز نمودیم زیرا فیلتر بالا گذر و فیلتر میان گذر و فیلتر میان نگذر را می‌توان با تبدیلات فرکانسی فیلتر پایین گذر بدست آورد [۲،۴].

۱-۴- تحقق فیلتر های دیجیتال

وقتی که مشخصات مورد نیاز فرمولبندی شد، با استفاده از یک روش از روش های بهینه سازی یا تخمین، یک تابع انتقال مناسب بدست می‌آوریم. در این صورت تحقق تابع انتقال بدست آمده تضمین می‌شود. تابع انتقال را نمی‌توان با استفاده از ساختارهای اختیاری تحقق داد. هر کدام از این ساختارها ویژگی های مخصوص به خود را دارند و می‌توانند بسیار متفاوت از یکدیگر باشند. برای انتخاب ساختار مناسب ابتدا باید ویژگی های ساختارهای مختلف را شناخت و سپس مناسبترین ساختار را متناسب با معیار های مورد انتظار از سیستم انتخاب کرد. این ویژگی‌ها عبارتند از: پیچیدگی ساختار، انحراف مقادیر ضرایب، میزان تاثیر گذاری خطای گرد کردن بر عملگر های محاسباتی و ... . از پر استفاده ترین تحقق‌ها می‌توان به موارد زیر اشاره کرد [۲،۸]:

۱- Direct
۲- Direct canonic
۳- State-space
۴- Ladder
۵- Lattice
۶- Parallel
۷- Cascade

در این پایان نامه تمرکز اصلی بروی پیاده سازی فیلترهای آبشاری[۱۳] و برسی ویژگی های آن‌ها است.

۱-۵- ملاحظات در پیاده سازی فیلتر دیجیتال

فیلتر دیجیتال می‌تواند به چندین راه پیاده سازی می‌شوند. پیاده سازی می‌تواند بروی یک کامپیوتر معمولی یا یک پردازنده ی سیگنال، ویا یک سخت افزار مانند چیپ VLSI انجام شود. در بسیاری از موارد، سرعت سخت افزار و منابع سخت افزاری مانند واحد های محاسباتی، حافظه و مساحت چیپ محدود است. در نتیجه، سیگنال های داخلی و خارجی همانند ضرایب فیلتر بصورت کاملا باینری و با طول محدود نمایش داده می‌شوند [۱].

معمولا تعداد بیشمار تابع انتقال می‌توان یافت که معیارهای مورد انتظار را دارا می‌باشند. علاوه بر این، تعداد زیادی ساختار فیلتر دیجیتال می‌توان یافت که برای تحقق یک تابع انتقال داده شده می‌توانند استفاده شوند. این ساختارها معمولا تعیین کننده ی خصوصیات پیاده سازی نهایی خواهند بود، اعم از سرعت، معماری مناسب سخت افزار و دقت محاسباتی مورد نیاز. در پیاده سازی هایی که از پردازنده های ممیز ثابت استفاده می‌شود، معمولا بیشینه ی طول کلمات محدود خواهند بود. بیشینه ی طول کلمات در پردازنده های سیگنال توسط خصوصیاتی مانند طول کلمات گذرگاه داده و بلوک های ضرب کننده تعیین می‌گردد، در حالی که در پیاده سازی های VLSI، بیشینه ی مجاز برای طول کلمات توسط خصوصیاتی مانند تاخیر، مساحت چیپ و توان مصرفی تعیین می‌گردد. طراحی فیلتر دیجیتال همواره نیازمند یک نوع مصالحه بین کارایی فیلتر و هزینه ی پیاده سازی آن است. هزینه ی پیاده سازی در ابتدا بستگی به پیچیدگی ساختار فیلتر دارد. این هزینه در اصل بستگی به تعداد عناصر تاخیر دهنده و ذخیره سازی و ضرب کننده‌ها و جمع کننده‌ها و طول کلمات مورد نیاز برای نمایش ضرایب سیگنال دارد [۱].

[13] Cascade

۱-۵-۱- نمایش ضرایب

عملگرهای محاسباتی در فیلتر های دیجیتال بطور اساسی عملگر های باینری هستند. این عملگرها شامل جمع باینری، ضرب و منفی سازی است. نحوه ی نمایش اعداد استفاده شده در این عملگرها را می‌توان به سه دسته ی اصلی تقسیم کرد. ممیز ثابت، ممیز شناور و ممیز شناور بلوکی. در این پایان نامه هدف ما پیاده سازی فیلتر بصورت ممیز ثابت است، بنابر این ما اینجا تنها به چند الگوی متداول برای نمایش اعداد ممیز ثابت می-پردازیم. اگر طول کلمه به اندازه B+1 بیت برای نمایش اعداد انتخاب شود، بنابراین دقیقا 2^{B+1} عدد مختلف غیر منفی را با این طول کلمه می‌توان نشان داد. در نمایش ممیز ثابت، فرض بر این است که مکان ممیز باینری ثابت است. بسته به نحوه ی نمایش اعداد منفی، سه روش مختلف برای نمایش محاسبات ممیز ثابت وجود دارد [۱۰،۴]. این سه روش عبارتند از اندازه علامت، مکمل ۲، مکمل ۱. ازمیان این چند روش، روش نمایش مکمل ۲ متداولترین روش است، زیرا پردازش و پیاده سازی آن آسان است. بنابراین مزایای بیشتری نسبت به سایر روشها دارد.

۱-۵-۱-۱- نمایش مکمل ۲

عدد مکمل ۲ که با (B+1) بیت نمایش داده می‌شود، با رابطه ی زیر به عدد حقیقی x مربوط می‌گردد:

$$x = \Delta\left(-a_0 + \sum_{n=1}^{B} a_n 2^{-n} \right) \qquad (۱-۲)$$

در رابطه ی (۱-۲) به ازای $n \geq 1$، $a_n \in \{0,1\}$ است. بیت اول a_0 بیت علامت است (به ازای $a_0 = 0$، $0 \leq x \leq \Delta\left(1-2^{-B}\right)$ و به ازای $a_0 = 1$، $-\Delta \leq x \leq 0$ و) مقدار Δ یک عدد مثبت اختیاری است. برای فیلتر های دیجیتال، Δ معمولا مقدار واحد دارد. برای بدست آوردن مقدار منفی یک عدد مثبت مکمل ۲، یک مکمل گرفتن از عدد مثبت مورد نظر و جمع کردن عدد یک در موقعیت کم ارزشترین بیت کافی است و عدد

منفی مربوطه را نتیجه می‌دهد. اندازهی مثبت ترین عدد قابل نمایش، یک عدد واحد از اندازه ی منفی ترین عدد قابل نمایش کوچکتر است. زیرا برای نمایش عدد صفر تنها یک حالت نمایش وجود دارد. حالت نمایش مکمل ۲ خاصیت مطلوب دیگری نیز دارد، بدین صورت که اگر چندین متغیر با هم جمع گردند، در صورتی که جمع نهایی در رنج مطلوب باشد، هیچ سرریزی رخ نخواهد داد. اگر چه جمع میانی ممکن است که خارج از رنج قرار گیرد [۱].

۱-۵-۱-۲- نمایش توان های ۲ علامتدار

در پیاده سازی VLSI، ضرب کننده‌ها المان هایی گران قیمتند. بنابراین توجه عمده ی طراحان به سمت انتخاب ساختارهایی است که در روند محاسباتی خود کمترین نیاز را به ضرب دارند. همانطور که می‌دانیم نمونه های سیگنال ورودی پس از وارد شدن به فیلتر در ضرایب فیلتر ضرب می‌شوند و خروجی فیلتر را ایجاد می-کنند. در برخی از فیلترها ضرب نمونه های سیگنال ورودی در مقادیر ضرایب فیلتر می‌تواند توسط توالی ای از شیفت‌ها و جمع‌ها پیاده سازی شوند. این بدین معناست که در این گونه فیلترها دیگر نیازی به ضرب کننده نیست و بنابراین هزینه ی پیاده سازی فیلتر به نحو چشمگیری کاهش میابد. فیلتر دیجیتالی که تمام ضرایب آن با استفاده از توالی عملگرهای شیفت و جمع پیاده سازی شود را فیلتر بدون ضرب کننده می‌نامند [۱].

نمایش توان های ۲ علامتدار[14] یک عدد بطور رسمی‌به کد عدد علامتدار ریشه ۲ [15] مشهور است. در این زمینه، عدد x، مطابق معادله با $\Delta = 1$ بصورت زیر تعریف می‌شود:

$$x = \sum_{n=1}^{R} s_n 2^{-Pn} \qquad (۲-۲)$$

[14] SPT: Signed Power-of-Two
[15] Radix-2 signed-digit code

در رابطه ی بالا $s_n \in \{-1,1\}$ و $P_n \in \{0,1,...,B\}$ به ازای $n = 1,2,\ldots,R$. نحوه ی نمایش معرفی شده در معادله ی بالا دارای R عدد غیر صفر است. تعداد اعداد غیر صفر در کد عدد علامتدار، وزن نامیده می-شود [۲۴]. کمینه بودن وزن، لازمه ی داشتن فشرده ترین نمایش مربوط به یک کد است. نمایش یک عدد علامتدار کانونی[۱۶] به عنوان فشرده ترین نمایش تعریف می‌گردد. اگر به ازای $n = 2,3,\ldots,R$ ، $s_n s_{n-1} = 0$ باشد، این نمایش شبیه چیزی است که در رابطه ی (۲-۱) گفته شده بود، بغیر از آنکه اکنون x می‌تواند خود مثبت یا منفی باشد، بدون انکه نیازی به یک بیت علامت مشخص باشد [۱].

تعداد ضرب کننده‌ها و جمع کننده های مورد نیاز برای پیاده سازی ضرایب فیلتر را می‌توان با استفاده از روش حذف زیرعبارت[۱۷] یا روش اشتراک زیرعبارت مشترک[۱۸] کوتاهتر از این نیز نمود [۲۵،۲۶،۲۷،۲۸،۲۹،۳۰]. این‌ها تبدیلات عددی ای هستند مربوط به ضرب های چند ثابتی که منجر به ایجاد یک پیاده سازی VLSI کارامد تر از نظر سرعت، توان و مساحت می‌شود. روش حذف زیرعبارت به یک دسته از ضرب های ثابت با مضرب های مشابه اعمال می‌شود. به طور اساسی حذف زیرعبارت فرایندی است در جهت تشخیص ضرب های ثابت و عملگر های اضافی و پیاده سازی شیفت و جمع و تفریق بجای آنها. وقتی که عملیات زایدی یافت شد، عملگر های زاید تنها یکبار اعمال می‌گردند و میان محاسبات مشابه بعدی به اشتراک گذاشته می‌شود .

۱-۵-۲- تاثیر طول کلمات محدود بر فیلتر های دیجیتال

در پیاده سازی های سخت افزاری، یک فیلتر دیجیتال یا با استفاده از پردازنده ی سیگنال پیاده سازی می-شود و یا با استفاده از سخت افزاری خاص که در هر دو مورد رفتار فیلتر پیاده شده بروی سخت افزار دارای رفتاری متفاوت از همتای ایده آل خود است. به دلیل محدود بودن طول کلمات برای ذخیره و نمایش سیگنال های داخلی فیلتر و ضرایب فیلتر، مقداری انحراف در پاسخ اندازه و پاسخ فاز این پیاده سازی‌ها نسبت به پاسخ

[16] CSD: canonic signed-digit
[17] Subexpression elimination
[18] Common subexpression sharing

حالت دقت-نامحدود ایجاد می‌شود. کوانتیزه کردن نتایج حاصل از ضرب نمونه های سیگنال ورودی و ضرایب فیلتر، یک نویز گرد کردن را در خروجی فیلتر ایجاد می‌کند. هرچه طول کلمات محدودتر باشد، میزان نویز گرد کردن بیشتر خواهد بود و مقدار انحراف در پاسخ اندازه و پاسخ فاز فیلتر پیاده شده نسبت به پاسخ حالت دقت-نامحدود شدیدتر است. هزینه، پیچیدگی و سرعت تولید خروجی شدیدا به طول کلمات ضرایب وابسته است. در پیاده سازی های سخت افزاری این نویز باید کمینه باشد تا همچنان مشخصات مربوط به فیلتر در حد قابل قبول بماند [۱].

نوع دیگری از خطاها وجود دارد که از کوانتیزه کردن نتایج میانی سیستم حاصل می‌شود. این گونه خطا را می‌توان بصورت اجزای بسیار همبسته در سیگنال خروجی یا سیگنال های داخلی سیستم دید. این خطاها معمولا شکلی از نوسانات قطعی است که نوسانات پارازیتی نامیده می‌شوند. نوسانات پارازیتی به دلیل وجود اتصالات فیدبک می‌توانند افزایش یابند و حتی بروی رفتار سیستم تاثیر جدی بگذارند. بنابر این مساله ی وجود نوسانات پارازیتی تنها در فیلتر های بازگشتی مشکل زا است زیرا وجود آن در فیلترهای غیر بازگشتی تاثیر چندانی بروی رفتار سیستم ندارند [۱].

میزان کارایی فیلتر به عوامل زیر بستگی دارد [۸]:

- طول کلمات و نوع محاسباتی که برای عملیات فیلترینگ انجام شده
- روشی که برای کوانتیزه کردن ضرایب استفاده شده
- پیکربندی قطب و صفر
- مشخصه های سرریز
- گرد کردن و برش حاصلضرب ها
- ساختار فیلتر
- سیگنال ورودی

در نهایت، نویز گرد کردن و خطاهای کوانتیزه کردن را می‌توان به چند روش کاهش داد:

- استفاده از ساختاری که دارای حساسیت و نویز پایین هستند [۲۵-۲۷]
- بهینه سازی ضرایب فیلتر بروی یک فضای گسسته مشخص [۲۸-۳۹]
- انتخاب تقریبی از فیلتر که در آن حساسیت و نویز کمینه است [۴۰-۴۳]

۱-۵-۲-۱- ساختار هایی با حساسیت پایین

اهمیت پایین بودن حساسیت ساختار فیلتر نسبت به کاهش طول کلمات بخاطر آن است که پاسخ فیلترهایی که چنین ساختارهایی دارند حتی با وجود کوانتیزه شدن ضرایب، بسیار نزدیک به حالت دقت-نامحدود خواهد بود. از مزایای استفاده از این گونه ساختارها آن است که پیچیدگی کلی پیاده سازی کم می‌شود و بهره وری محاسباتی افزایش می‌یابد، زیرا در این حالت ضرایب را می‌توان با تعداد بیت های کمتری پیاده سازی نمود. علاوه بر این، ساختارهای خاصی با این ویژگی وجود دارد که اگر ضرایب فیلتر مربوط به آن در رنج خاصی محدود شده باشد، لزوما فیلتر پایدار خواهد بود. برای چنین ساختارهایی، کوانتیزه کردن ضرایب را میتوان بصورتی اجرا نمود که پایداری کاهش نیابد [۱].

مزایای این ساختارها آن است که با استفاده از حداقل بیت برای نمایش ضرایب، می‌توان فیلتری طراحی کرد که با معیارهای داده شده منطبق باشد. برای پیاده سازی های VLSI، می‌توان بجای استفاده از ضرب کننده های معمولی، تمام مقادیر ضرایب را بصورت توان های ۲ بیان کنیم. این قابلیتی بسیار کاربردی است زیرا جایگزینی شیفت و جمع را بجای ضرب مقدور می‌سازد. در کل برای پیاده سازی های سخت افزاری، بسیار دشوار است که بتوان با محدودیت های موجود بروی طول کلمات، پیاده سازی را با معیار های داده شده منطبق نمود [۱].

۱-۵-۲-۲- روش های بهینه سازی

در طول دو دهه ی گذشته، الگوریتم های زیادی برای بهینه سازی فیلتر های دیجیتال معرفی شدند. عموما الگوریتم های بهینه سازی ضرایب فیلتر را بروی یک فضای گسسته بهینه سازی می‌کنند [۴۴-۴۶]. این الگوریتم‌ها شامل روش های جستجوی محلی [۴۷-۵۰]، الگوریتم های بهینه سازی اتفاقی مانند الگوریتم ژنتیک [۵۱-۵۵]، جستجوی تابو[19] [۵۶،۵۷] و کوانتیزه سازی با استفاده از حساسیت ضرایب [۵۶-۵۸] و تخمین حوزه ی زمان [۵۹] و ... می‌شوند.

در گذشته جستجوی مستقیم با روش هوک و جیوز[20] یکی از پر استفاده ترین الگوریتم‌ها بهینه سازی گسسته بود. امروزه الگوریتم های بهینه سازی تصادفی مانند الگوریتم ژنتیک، الگوریتم جستجوی تابو توجهات بسیاری را به خود جلب نموده است. با وجود آنکه این الگوریتم‌ها دارای حجم پردازش و محاسبات بالایی هستند و هزینه ی اجرای آنها بالاست، هیچ تضمینی وجود ندارد که این روش‌ها باعث ایجاد بهترین بهینه سازی شوند زیرا همه ی آنها بر مبنای احتمال هستند [۱۶].

۱-۵-۲-۳- روش های تخمین

در روش های تخمین، فیلتر دیجیتال به نحوی طراحی می‌شوند که میزان حساسیت و نویز مینیمم شود [۶۰]. سنجش مربوطه می‌تواند پیچیده باشد مانند طول کلمه ی آماری معرفی شده توسط آوانهوس[21] [۶۰] و گسترش یافته توسط کروچیر[22] [۴۲]، و یا ساده باشد مانند محاسبه ی فاصله ی دورترین قطب [۴۰].

[19] Tabu search
[20] Hooke and Jeeves
[21] Avenhaus
[22] Crochiere

فصل دوم:

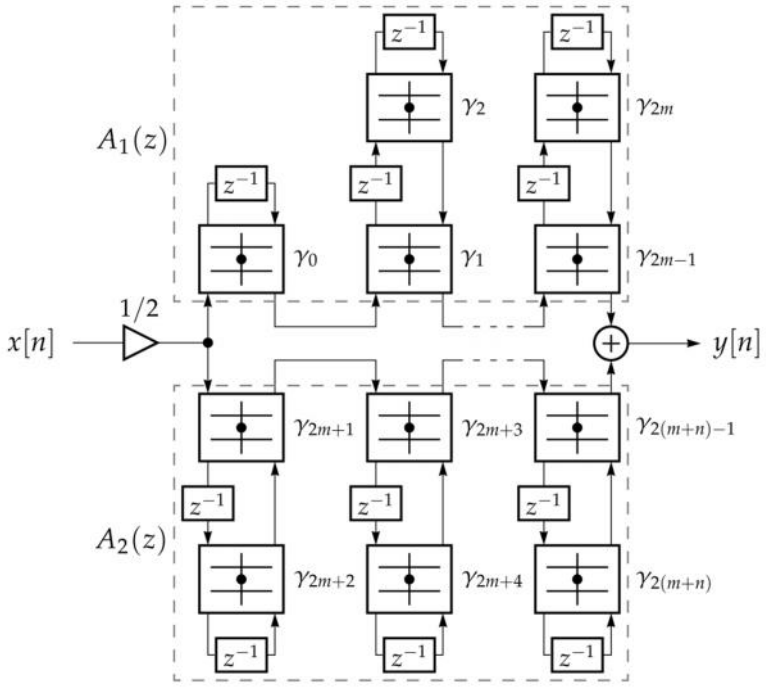
فیلترهای LWD

۳-۱- معرفی فیلترهای LWD [۲۳]

طراحی فیلترهای دیجیتال ساخته شده از ترکیب برش‌های تمام گذر در سه دهه اخیر توجه بسیاری را به خود جلب نموده است. ساختار این گونه فیلترها دارای پیچیدگی کمی است و حساسیت آن نسبت به طول کلمات محدود بسیار کم است. این گونه فیلترها مولفه هایی مفید در بسیاری از کاربردها هستند [۶۱-۶۷]. در گذشته از فیلترهای تمام گذر برای یکسان سازی فاز استفاده می‌شد. استفاده آنها در فیلتر های چند نرخی، فیلتر بانک ها، فیلترینگ دیجیتال قابل تنظیم و دیگر کاربردهای شرح داده شده است. خرده فیلتر های تمام گذر، بلوکی اساسی در ساخت فیلتر های نیم باند بازگشتی است. این گونه فیلترها در کاربردهای تبدیل نرخ نمونه برداری بسیار کارامدند. [۱]

ارتباط موازی بین دو فیلتر تمام گذر در پیاده سازی کانونی مستقیم ، با کمتر از نیمی از ضرب کننده ها قابل پیاده سازی است. علاوه بر آن، این خرده فیلترهای تمام گذر، می‌توانند با استفاده از بخش های مرتبه یک و مرتبه دو به عنوان بلوک های سازنده، ساخته شوند. ساختار اینگونه فیلترها به خوبی از زیر بخش های مجزا تشکیل شده است. این ویژگی‌ها این گونه فیلترها را برای پیاده سازی بروی پردازنده های سیگنال و VLSI مناسب می‌گرداند [۶۸، ۶۹]. تعداد زیادی از ساختار هایی که توسط بخش های تمام گذر اجرا می‌شوند، مشخصات تمام گذر را دارا هستند، بدون در نظر گرفتن کوانتیزه شدن ضرایب. کلاس شناخته شده از این نوع فیلتر ها، کلاس فیلتر های دیجیتال موجی می‌باشد که به یک طبقه اولیه خاص از شبکه های آنالوگ مربوط است. با اتصال موازی دو فیلتر تمام گذر، فیلتر دیجیتال موجی تحقق می‌یابد که فیلتر LWD نیز نامیده می-شود . این ساختار به دلیل خاصیت کنش پذیری، دارای حساسیت بسیار کمی در باند گذر است [۷۰]. شکل (۲-۱) نشان دهنده یک تحقق از فیلتر LWD با مرتبه فرد است.

[23] LWD: Lattice Wave Digital Filter



شکل(۷-۱) اتصال موازی بین دو فیلتر تمام گذر. فیلترهای $A_1(z)$ و $A_2(z)$ فیلترهایی تمام گذر و پایدارند که از بخش هایی آبشارگونه شامل فیلترهای تمام گذر دیجیتال موجی مرتبه ۱ و مرتبه ۲ تشکیل شده اند.

مشهور است که فیلتر های دیجیتالی که بصورت اتصال موازی دو فیلتر دیجیتال تمام گذر هستند، بدلیل رابطه نزدیک آنها با شبکه های پیوسته در زمان بی اطلاف منقضی دوگانه[۲۴]، بصورت ذاتی دارای رنج دینامیکی خوبند. نمایش بلوکی این نوع فیلترها را در شکل (۲-۱) مشاهده می‌نمایید. فتویس[۲۵] و دیگران نشان دادند که فیلتر های تمام گذر موازی مشخصات مطلوب دیگری دارند، مانند مصونیت از نوسانات پارازیتی، به دلیل سرریز یا نوسانات دانه دانه. [۱]

علاوه بر این، تمام فیلتر های یاد شده در بالا می‌توانند بطور طوری طراحی شوند که بطور تقریبی دارای پاسخ فاز خطی در باند گذر خود باشند. این ساختار می‌تواند در کاربردهایی مفید واقع گردد که در آن فیلترهای FIR فاز خطی دارای تاخیر بیش از حدند. [۱]

[24] Doubly terminated
[25] Fetteweis

۳-۲- فیلتر های LWD آبشاری

یکی از بهترین ساختارها برای پیاده سازی فیلترهای دیجیتال بازگشتی، فیلتر LWD است (اتصال موازی دو فیلتر همه گذر). این نوع فیلترها با تعداد زیادی خواص جالب توصیف می‌شوند، مانند حساسیت پایین نسبت به طول کلمات، نویز پایین در گرد کردن، نبود نوسانات پارازیتی. عیب اصلی این نوع فیلترها آن است که اگر تضعیف در باند قطع بالا باشد، برای نمایش ضرایب نیازمند تعداد زیادی بیت خواهیم بود. برای فایق آمدن بر این مساله، یک ساختار شامل یک توالی از فیلترهای LWD را می‌توان استفاده نمود [۷۰].

مهمترین مزیت ساختار آبشاری برای LWD در مقایسه با LWD مستقیم آن است که قطب های ساختار آبشاری بسیار دورتر از دایره واحد هستند. در این حالت تعداد بیت های مورد نیاز جهت نمایش دادن و ضرایب بصورت محسوسی کاهش می‌یابد. هزینه ای که در مقابل این کاهش می‌پردازیم آن است که مرتبه کلی فیلتر کمی افزایش می‌یابد [۷۰].

با انتخاب مناسب تعداد فیلترهای LWD و مرتبه آنها و بهینه سازی آنها، ضرایب آنها با استفاده از تعدادی از توان های ۲ قابل پیاده سازی می‌شود. فیلتر هایی از این دست در پیاده سازی های VLSI کاربرد های بسیاری دارد، زیرا این خاصیت باعث می‌شود که فیلترها را بدون استفاده از ضرب و تنها با بکارگیری عملگرهای شیفت و جمع و تفریق پیاده سازی کرد. این خاصیت از این جهت مهم است که ضرب کننده های متداول بسیار گرانند و استفاده نکردن از المان های ضرب کننده هزینه پیاده سازی را کاهش می‌دهد [۷۰].

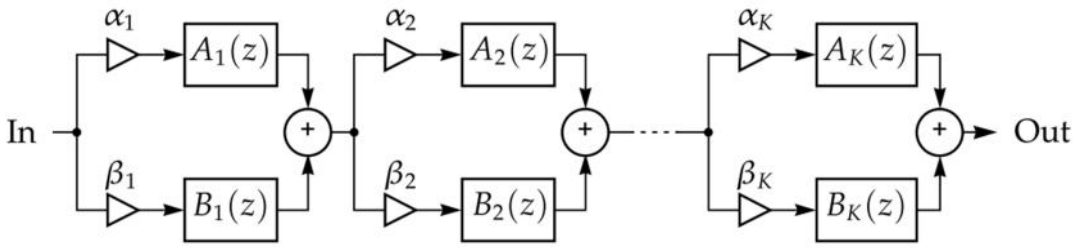
۳-۳- تابع انتقال فیلتر LWD

فرم کلی تابع انتقال مهمترین دسته از فیلترهای LWD بصورت زیر است [۷۰]:

$$H(z)=\prod_{k=1}^{K}H_k(z) \qquad (۱-۲)$$

$$H_k(z) = \alpha_k A_k(z) + \beta_k B_k(z) \qquad (۲-۲)$$

در اینجا A_k و B_kها به ازای k=1, 2, …, K فیلترهایی تمام گذر و پایدارند که مرتبه آنها به ترتیب M_k و N_k است. برای فیلترهای LWD مرسوم و فیلترهای فاز خطی تقریبی K=1 و $\alpha_1 = \frac{1}{2}$ و $\beta_1 = \frac{1}{2}$ است. در حالی که برای فیلترهای LWD آبشاری، K>1 و α_k و β_k می‌توانند مقادیری اختیاری باشند. در زیر ما از اصطلاحات فیلتر LWD و فیلترهای LWD مستقیم بجای هم استفاده می‌کنیم. پیاده سازی تابع انتقال بالا در شکل ۴,۱ نشان داده شده است. در ادامه تاکید اصلی بروی فیلترهای پایین گذر است. هرچند که تابع انتقال فیلترهای میان گذر و بالاگذر نیز به روشی مشابه ساخته می‌شود. در مواردی که از فیلترهای پایین گذر استفاده می‌شود، $M_k = N_k - 1$ است، بنابراین مرتبه فیلتر معادل $M_k + N_k$ فرد خواهد بود. در شکل(۷-۲) اتصال آبشاری فیلترهای A_k و B_k را مشاهده می‌نمایید.



شکل(۷-۲) اتصال آبشاری فیلتر LWD، A_k و B_kها فیلترهایی تمام گذر و پایدارند.

اگر A_k و B_kها بصورت آبشاری از ساختارهای حصیری تمام گذر مرتبه اول و دوم پیاده سازی شود، و M_k و N_k فرض شود که بترتیب فرد و زوج هستند، بنابراین A_k و B_kها توسط ضرایب تطبق بصورت زیر قابل تعریفند:

$$A_k(z) = \frac{-\gamma_0^{(k)} + z^{-1}}{1 - \gamma_0^{(k)} z^{-1}} \prod_{l=1}^{m_k} \frac{-\gamma_{2l-1}^{(k)} + \gamma_{2l}^{(k)}(\gamma_{2l-1}^{(k)} - 1)z^{-2} + z^{-2}}{1 + \gamma_{2l}^{(k)}(\gamma_{2l-1}^{(k)} - 1)z^{-1} - \gamma_{2l-1}^{(k)} z^{-2}} \qquad (۳-۲)$$

و

$$B_k(z)=\prod_{l=1}^{n_k}\frac{-\hat{\gamma}_{2l-1}^{(k)}+\hat{\gamma}_{2l}^{(k)}(\hat{\gamma}_{2l-1}^{(k)}-1)z^{-2}+z^{-2}}{1+\hat{\gamma}_{2l}^{(k)}(\hat{\gamma}_{2l-1}^{(k)}-1)z^{-1}-\hat{\gamma}_{2l-1}^{(k)}z^{-2}} \qquad (۴-۲)$$

بطوری که:

$$n_k=N_k/2 \quad و \quad m_k=(M_k-1)/2 \qquad (۵-۲)$$

اگر A_k دارای قطبی حقیقی در $z=r_0^{(k)}$ و m_k قطب مزدوج مختلط در $z=r_l^{(k)}\exp(\pm j\theta_l^{(k)})$ به ازای $l=1,2,\ldots,m_k$ باشد و B_k دارای n_k قطب مزدوج مختلط در $z=\hat{r}_l^{(k)}\exp(\pm j\hat{\theta}_l^{(k)})$ به ازای $l=1,2,\ldots,n_k$ باشد، در این صورت:

$$\gamma_0^{(k)}=r_0^{(k)} \qquad (۶-۲)$$

$$y_{2l-1}^{(k)}=-\left(r_l^{(k)}\right)^2, y_{2l}^{(k)}=\frac{2r_l^{(k)}\cos(\theta_l^{(k)})}{1+\left(r_l^{(k)}\right)^2} \quad \text{به ازای} \quad l=1,2,\ldots,m_k \qquad (۷-۲)$$

و

$$\hat{y}_{2l-1}^{(k)}=-\left(\hat{r}_l^{(k)}\right)^2, \hat{y}_{2l}^{(k)}=\frac{2\hat{r}_l^{(k)}\cos(\hat{\theta}_l^{(k)})}{1+\left(\hat{r}_l^{(k)}\right)^2} \quad \text{به ازای} \quad l=1,2,\ldots,n_k \qquad (۸-۲)$$

۳-۴- مساله بهینه سازی

قبل از شروع مساله بهینه سازی، ما تابع انتقال یک فیلتر را با $H(\phi,z)$ نشان می‌دهیم، جایی که ϕ برداریست با پارامتر های قابل تنظیم [70]

$$\phi=[\alpha_1,\beta_1,r_0^{(1)},...,r_{m1}^{(1)},\theta_1^{(1)},...,\theta_{m1}^{(1)},\hat{r}_1^{(1)},...,\hat{r}_{n1}^{(1)},\hat{\theta}_1^{(1)},...,\hat{\theta}_{n1}^{(1)},...$$

$$...,\alpha_k,\beta_k,r_0^{(k)},...,r_{mk}^{(k)},\theta_1^{(k)},...,\theta_{mk}^{(k)},\hat{r}_1^{(k)},...,\hat{r}_{nk}^{(k)},\hat{\theta}_1^{(k)}...,\hat{\theta}_{nk}^{(k)}]. \quad (۹-۲)$$

مشخصات اندازه برای فیلتر بصورت زیر خواهد بود

$$1-\delta_P \le \left|H\left(\phi,e^{j\omega}\right)\right| \le 1 \qquad \text{به ازای} \qquad \omega\in\left[0,\omega_p\right] \quad (۱۰-۲)$$

$$\left|H\left(\phi,e^{j\omega}\right)\right| \le \delta_s \qquad \text{به ازای} \qquad \omega\in\left[\omega_p,\pi\right] \quad (۱۱-۲)$$

همچنین معیارها می‌توانند بصورت زیر بیان شوند

$$\left|E\left(\phi,\omega\right)\right| \le 1 \qquad \text{به ازای} \qquad \omega\in\left[0,\omega_p\right]\cup\left[\omega_s,\pi\right] \quad (۱۲-۲)$$

$$E\left(\phi,\omega\right) \le 0 \qquad \text{به ازای} \qquad \omega\in\left[0,\omega_p\right] \quad (۱۳-۲)$$

بطوری که

$$E\left(\phi,\omega\right)=W(\omega)\left[\left|H\left(\phi,e^{j\omega}\right)\right|-D(\omega)\right] \quad (۱۴-۲)$$

که در آن

$$D(\omega)=\begin{cases}1 & \omega\in\left[0,\omega_p\right]\\ 0 & \omega\in\left[\omega_s,\pi\right]\end{cases} \quad (۱۵-۲)$$

و

$$W(\omega)=\begin{cases}1/\sigma_p, & \omega\in\left[0,\omega_p\right]\\ 1/\sigma_s, & \omega\in\left[\omega_s,\pi\right]\end{cases} \quad (۱۶-۲)$$

شرایط مورد نیاز برای فیلترهای LWD که تقریبا دارای فاز خطی هستند به صورت زیر است:

$$\left|\arg[H(\phi,e^{j\omega})]-\tau_0\omega\right|\le\Delta \quad \text{به ازای} \quad \omega\in\Omega_p \qquad (۱۸-۲)$$

در رابطه بالا $\arg[H(\phi,e^{j\omega})]$ نشان دهنده پاسخ فاز آزاد فیلتر و τ_0 انحراف پاسخ فاز سیستم از پاسخ فاز خطی ایده آل است. τ_0 پارامتری قابل تنظیم است. برای سادگی محاسبات، محدودیت های پایداری سیستم در معادلاد مساله بهینه سازی وارد نشد. زیرا محدودیت های پایداری فیلتر به راحتی با یک محدودیت مرزی ساده قابل اعمال است.

۳-۴-۱- کوانتیزه کردن ضرایب

این عملیات بروی کوانتیزه سازی ضرایب در حالت ممیز ثابت تمرکز می‌کند. در فیلتر های دیجیتال عملگر ضرب بسیار پر استفاده است، زیرا ضرایب فیلتر در نمونه های سیگنال ورودی ضرب می‌شود و خروجی را تولید می‌کند. در بسیاری از پیاده سازی ها، جایگزین کردن عملگرهای های شیفت و جمع بجای ضرب بسیار مفید خواهد بود. برای دست یافتن به این هدف، مطلوب است که ضرایب فیلتر به فرم زیر بیان گردند [۱]

$$x=\sum_{n=1}^{R}a_r\,2^{-\mathrm{Pr}} \qquad (۱۷-۲)$$

در رابطه بالا $a_r\in\{-1,1\}$ و P_r اعدادی صحیح اند که بصورت افزایشی مرتب شده اند. نحوه نمایش معرفی شده در معادله بالا دارای R عدد غیر صفر است. هدف یافتن مقادیر پارامترهای بردار ϕ به نحوی که

۱) R یا تعداد توان های ۲ تا حد امکان کم شود

۲) P_r یا تعداد بیت های اعشار تا حد امکان کم شود

۳-۴-۲- حل مساله بهینه سازی

عموما در مسایل بهینه سازی ابتدا چهارچوب اولیه ای برای پاسخ قابل قبول تعیین می‌گردد و آنقدر جواب های مختلف در چهارچوب تعیین شده برسی می‌شود تا آنکه سیستمی نسبتا بهینه یافت شود که جواب قابل قبولی دارد. سپس سیستمی که بهینه ترین وضعیت را در میان حالت های برسی شده داشت، به عنوان سیستم بهینه معرفی می‌شود [۱].

در مسایل بهینه سازی نقطه بهینه مطلق وجود ندارد و الگوریتم بهینه سازی به دفعات تکرار می‌شود تا زمانی که سیستم بهینه تر یافت شود. برای یافتم پاسخ مساله بهینه سازی در زمینه فیلترهای LWD، باید K ، تعداد زیر فیلترها، M_kها و N_kها و پارامترهای بردار ϕ را بنحوی انتخاب کنیم که دو شرط زیر برقرار باشد[۷]:

۱) $H\left(\phi,z\right)$ معیارهای گفته شده در روابط (۲-۱۰) تا (۲-۱۳) را دارا باشد.

۲) ضرایبی که در بردار ϕ قرار دارد به نحوی کوانتیزه شوند که به اهداف مربوط به نحوه نمایش ضرایب دست یابیم.

فصل سوم:

شبیه سازی و مقایسه چند ساختار مهم

۶-۱- مقایسه چند ساختار مهم از فیلترهای دیجیتال

برای توصیف رابطه بین یک ورودی و خروجی می‌توان بیشمار ساختار پیدا نمود که بتواند این توصیف را انجام دهد. هر کدام از این ساختارها نقاط ضعف و قوت خاص خود را دارا می‌باشند. در پیاده سازی های عملی، به دلیل محدودیت های مربوط به سرعت پردازش و سخت افزاری، تنها برخی از این ساختارها می‌توانند در محاسبات دقت-محدود جواب قابل قبولی داشته باشند.

با وجود سرعت روز افزون در پیشرفت تکنولوژی های ساخت سخت افزار های پردازش سیگنال مانند پردازنده های دیجیتال و تراشه های VLSI و ...، حتی جدیدترین سخت افزارها نیز دارای سرعت پردازش و منابع محدودند. در پیاده سازی های عملی نیاز به ساختارهایی است که با وجود محدودیت های موجود در سخت افزار ها، پاسخ قابل قبولی در عمل داشته باشند. بنابراین ساختارهایی که سرعت پردازش بالاتری دارند و حساسیت کمتری نسبت به طول کلمه محدود دارند، از اولویت بالاتری برای پیاده سازی های عملی برخوردارند.

در پیاده سازی فیلترهای دیجیتال، فیلترهایی در عمل جواب بهتری می‌دهند که حساسیت پایینتری نسبت به کاهش طول کلمه ضرایب دارند. برای پیاده سازی های عملی ساختارها LWD دارای ویژگی های مناسبی جهت پیاده سازی های عملی هستند. ضرایب این فیلتر قابل بیان بصورت توان های ۲ است. این ویژگی سبب می‌شود که بتوان بجای ضرب از عملیات های شیفت و جمع استفاده کرد. این مطلب در پیاده سازی های عملی بسیار اهمیت دارد، زیرا ضرب کننده ها المان هایی بسیار گرانی هستند و تعداد ضرب کننده ها در هزینه نهایی پیاده سازی بسیار تاثیرگذار است.

در این فصل تعدادی از ساختارهای مهم فیلترهای دیجیتال، از لحاظ سرعت و پایداری و حساسیت باهم مقایسه می‌شوند. در قسمت اول آنالیز زمانی انجام می‌شود و سرعت تولید خروجی در هریک از ساختارهای مورد

نظر برسی می‌شود. در قسمت دوم آنالیز حساسیت انجام می‌شود و حساسیت هر یک از ساختارها نسبت به کاهش طول کلمات مربوط به بخش اعشاری ضرایب برسی می‌شود.

برای قابل مقایسه بودن نتایج حاصل از این فیلترها، آنها را با معیارها و خصوصیاتی تقریبا مشابه طراحی می‌کنیم. حاشیه باندگذر و حاشیه باندقطع برای این فیلترها بترتیب برابر است با $\omega_p = 0.05\pi$ و $\omega_s = 0.1\pi$. بیشینه ریپل مجاز برای باند گذر 0.5 dB و میزان تضعیف در باند قطع 100 dB است. همانگونه که مشاهده می‌شود، مشخصات گفته شده مربوط به فیلتر پایین گذر می‌شود. فیلترهایی که برای آنالیز های مورد نظر در نظر گرفته شده است به صورت زیر می‌باشند :

۶-۱-۱- فیلتر LWD مستقیم:

ویژگی های این فیلتر در بخش ۲-۱ و فرم کلی تابع انتقال آن در بخش ۲-۳ معرفی شده است. جدول۱ حاوی ضرایب فیلتر LWD مستقیم مربوطه است.

جدول ۷ ضرایب فیلتر پایین گذر LWD مستقیم

$A_k(z)$		$B_k(z)$	
$\gamma_0^{(1)}$	۹۷۴*۲^-۱۰		
$\gamma_1^{(1)}$	-۹۴۰*۲^-۱۰	$\hat{\gamma}_1^{(1)}$	-۹۳۴*۲^-۱۰
$\gamma_2^{(1)}$	۱۰۱۴*۲^-۱۰	$\hat{\gamma}_2^{(1)}$	۱۰۲۰*۲^-۱۰
$\gamma_3^{(1)}$	-۱۰۰۷*۲^-۱۰	$\hat{\gamma}_3^{(1)}$	-۹۶۴*۲^-۱۰
$\gamma_4^{(1)}$	۱۰۰۹*۲^-۱۰	$\hat{\gamma}_4^{(1)}$	۱۰۱۱*۲^-۱۰

۶-۱-۲- فیلتر LWD آبشاری دو طبقه:

ویژگی های این فیلتر در بخش ۲-۱-۳-۲ بیان شده و فرم کلی تابع انتقال آن در بخش ۲-۳ معرفی شده است. مرتبه این فیلتر ۹ و ضرایب آن توسط الگوریتم های بهینه سازی کوانتیزه شده و به فرم $\pm 1 \pm 2^{p_{r1}} \pm 2^{p_{r2}}$ در آمده است. جدول ۲ حاوی ضرایب فیلتر LWD آبشاری دو طبقه مربوطه است.

جدول ۸ ضرایب فیلتر پایین گذر LWD آبشاری دو طبقه

$A_k(z)$		$B_k(z)$	
$\gamma_0^{(1)}$	۱-۲$^{-۳}$-۲$^{-۷}$		
$\gamma_1^{(1)}$	-۱+۲$^{-۴}$+۲$^{-۸}$	$\hat{\gamma}_1^{(1)}$	-۱+۲$^{-۲}$-۲$^{-۴}$
$\gamma_2^{(1)}$	۱-۲$^{-۶}$-۲$^{-۸}$	$\hat{\gamma}_2^{(1)}$	۱-۲$^{-۶}$
$\gamma_0^{(2)}$	۱-۲$^{-۳}$+۲$^{-۵}$		
$\gamma_1^{(2)}$	-۱+۲$^{-۵}$+۲$^{-۷}$	$\hat{\gamma}_1^{(2)}$	-۱+۲$^{-۳}$
$\gamma_2^{(2)}$	۱-۲$^{-۶}$-۲$^{-۸}$	$\hat{\gamma}_2^{(2)}$	۱-۲$^{-۶}$+۲$^{-۸}$

۶-۱-۳- فیلتر Window-based FIR:

برای طراحی این فیلتر از روشی کلاسیک بنام فیلتر FIR فاز خطی پنجره ای[۲۶] استفاده شد. این روش فیلترهایی پایین گذر یا بالا گذر یا میان گذر با مشخصات استاندارد تولید می‌کند. مرتبه مد نظر برای این فیلتر ۹ خواهد بود. برای تولید ضرایب فیلتر از تابع fir1() برنامه MATLAB استفاده شده است. با استفاده از تابع یاد شده و اعمال مشخصات مورد انتظار، ضرایب فیلتر به صورت جدول ۳ است.

برای فیلتر Window-based FIR مرتبه n، فرم کلی تابع انتقال به صورت رابطه (۳-۱) است.

$$B(z) = b(1) + b(2)z^{-1} + \ldots + b(n+1)z^{-1} \qquad (۱-۳)$$

۶-۱-۴- فیلتر Butterworth IIR:

فیلتر های Butterworth IIR نوع خاصی از فیلتر های IIR هستند که با پاسخ اندازه خود مشخص می-شود. نمودار پاسخ اندازه این فیلترها در باند گذر کاملا یکنواخت و هموار است. مرتبه مد نظر برای این فیلتر ۹ بوده است.

[26] Windowed linear-phase FIR digital filter

جدول ۹ ضرایب فیلتر پایین گذر Window-based FIR

	ضرایب فیلتر
b(1)	۰/۰۱۳۳۷۹۰۳۵
b(2)	۰/۰۲۹۱۶۷۹۲۳
b(3)	۰/۰۷۱۱۹۷۸۸۱
b(4)	۰/۱۲۴۶۳۸۰۵۸
b(5)	۰/۱۶۸۷۳۹۷۷۹
b(6)	۰/۱۸۵۷۵۴۶۴۸
b(7)	۰/۱۶۸۷۳۹۷۷۹
b(8)	۰/۱۲۴۶۳۸۰۵۸
b(9)	۰/۰۷۱۱۹۷۸۸۱
b(10)	۰/۰۲۹۱۶۷۹۲۳
b(11)	۰/۰۱۳۳۷۹۰۳۵

برای تولید ضرایب فیلتر از تابع butter() برنامه MATLAB استفاده شده است. با استفاده از تابع یاد شده و اعمال مشخصات مورد انتظار، ضرایب فیلتر به صورت جدول ۴ است.

برای فیلتر Butterworth IIR مرتبه n، فرم کلی تابع انتقال به صورت رابطه (۳-۲) است.

$$H(z) = \frac{b(1) + b(2)z^{-1} + \ldots + b(n+1)z^{-1}}{1 + a(2)z^{-1} + \ldots + a(n+1)z^{-1}} \qquad (۳-۲)$$

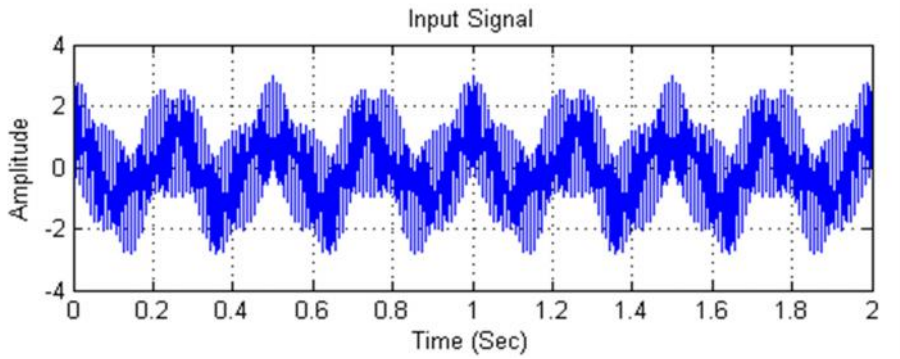
۶-۲- شبیه سازی ساختارهای معرفی شده

فیلترهای گفته شده در بالا را در محیط MATLAB طراحی می‌کنیم. فیلترها را در حوزه زمان پیاده سازی کرده و از سیگنال نشان داده شده در شکل (۳-۱) به عنوان ورودی فیلتر استفاده می‌کنیم. این سیگنال از ترکیب ۳ سیگنال سینوسی با فرکانس های ۴ هرتز، ۱۰۰ هرتز و ۲۱۰ هرتز تولید و با نرخ ۱ کیلوهرتز نمونه

برداری شده است. شبیه سازی مذکور در حالت دقت نامحدود انجام شده است و محدودیتی از نظر طول کلمه وجود ندارد.

جدول ۱۰ ضرایب فیلتر پایین گذر Butterworth IIR

اندیس	a	b
۱	۱	۱/۷۷۸۶۳E-۱۱
۲	-۸/۸۶۴۳۹۷۸۶۷	۱/۷۷۸۶۳E-۱۰
۳	۳۵/۴۱۸۱۹۰۹۷	۸/۰۰۳۸۳E-۱۰
۴	-۸۳/۹۹۴۲۵۶۵۶	۲/۱۳۴۳۵E-۹
۵	۱۳۰/۹۲۰۵۰۹۴	۳/۷۳۵۱۲E-۹
۶	-۱۴۰/۱۳۶۷۳۸۸	۴/۴۸۲۱۴E-۹
۷	۱۰۴/۳۱۷۱۶۴	۳/۷۳۵۱۲E-۹
۸	-۵۳/۳۲۱۸۱۰۲	۲/۱۳۴۳۵E-۹
۹	۱۷/۹۱۰۴۶۴۹۳	۸/۰۰۳۸۳E-۱۰
۱۰	-۳/۵۶۹۶۹۴۶۲۹	۱/۷۷۸۶۳E-۱۰
۱۱	۰/۳۲۰۵۶۵۸۴۱	۱/۷۷۸۶۳E-۱۱

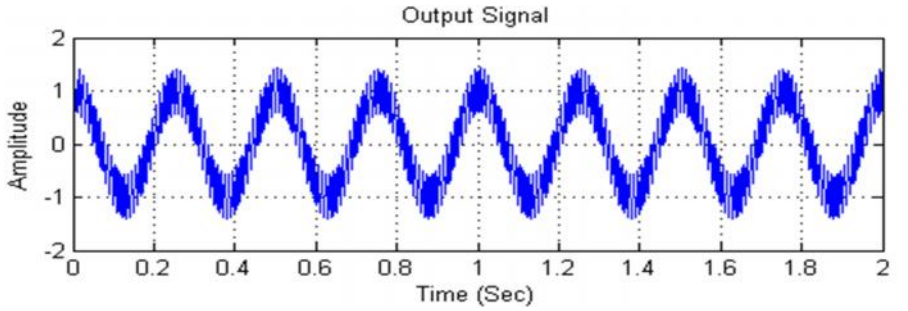


شکل(۱۰-۱) شکل موج سیگنال ورودی. این سیگنال از ترکیب ۳ سیگنال سینوسی با فرکانس های ۴ هرتز، ۱۰۰ هرتز و ۲۱۰ هرتز تولید شده است و با فرکانس ۱ کیلوهرتز نمونه برداری شده است

از آنجا که فیلترهای طراحی شده همگی پایین گذر هستند و به ازای فرکانس نمونه برداری ۱ کیلو هرتز فرکانس قطع باند گذر ۲۶ هرتز و فرکانس قطع باند قطع ۵۰ هرتز است، با اعمال سیگنال شکل (۳-۱) به عنوان

ورودی فیلتر، انتظار می‌رود که فرکانس های بالا تر از فرکانی قطع (یعنی فرکانس های ۱۰۰ و ۲۱۰ هرتز) حذف شده، و تنها فرکانس ۴ هرتز را در خروجی فیلتر دیده شود.

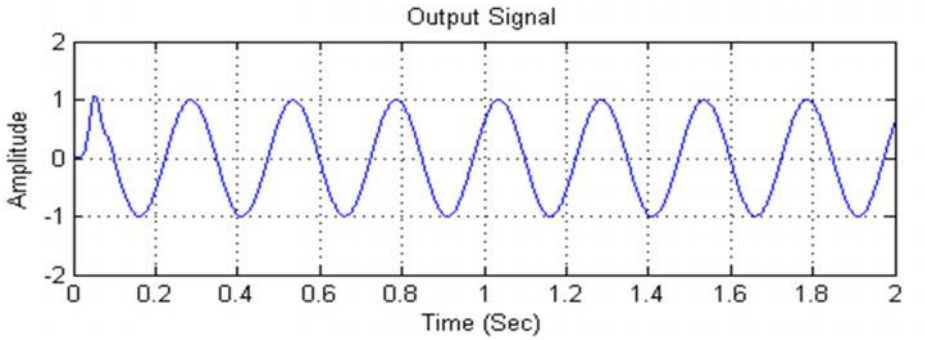
با اعمال ورودی بالا به فیلتر Window-based FIR، خروجی فیلتر بصورت زیر خواهد بود:



شکل(۱۰-۲) خروجی فیلتر Window-based FIR به ازای ورودی نشان داده شده در شکل (۳-۱)

همانگونه که می‌بینیم فیلتر Window-based FIR خروجی مناسبی در مرتبه ۹ ندارد و برای داشتن خروجی مناسبتر، باید مرتبه فیلتر Window-based FIR را بالاتر ببریم.

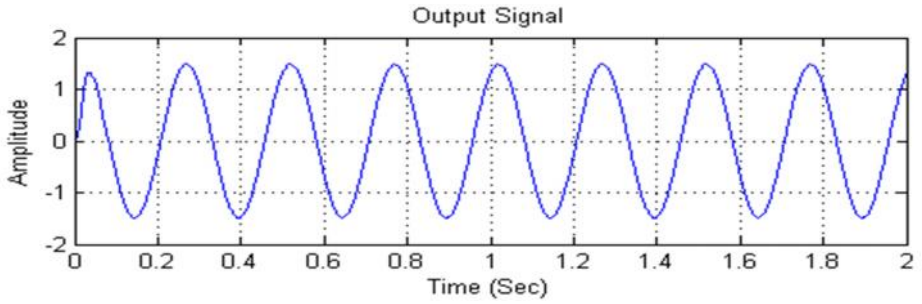
همچنین با اعمال ورودی ورودی شکل (۳-۱) به فیلتر Butterworth IIR، خروجی فیلتر بصورت زیر خواهد بود:



شکل(۱۰-۳) خروجی فیلتر Butterworth IIR به ازای ورودی نشان داده شده در شکل (۳-۱)

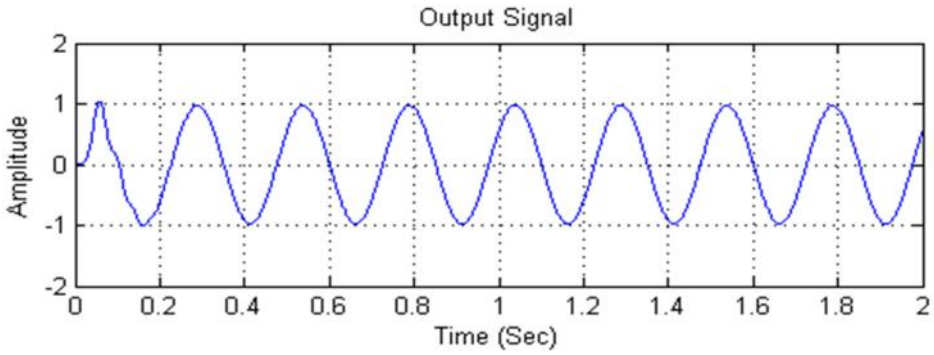
همانطور که مشاهده می‌شود، برعکس فیلتر Window-based FIR، فیلتر Butterworth IIR حتی در مرتبه ۹ نیز عملیات فیلتر کردن را بخوبی انجام می‌دهد. بنابراین فیلتر Butterworth IIR در مرتبه های پایین پاسخی مناسبتر از فیلتر Window-based FIR دارد.

همچنین با اعمال ورودی شکل (۳-۱) به فیلتر LWD آبشاری دو طبقه، خروجی فیلتر بصورت زیر خواهد بود:



شکل(۱۰-۴) خروجی فیلتر LWD آبشاری دو طبقه به ازای ورودی نشان داده شده در (۳-۱)

همچنین با اعمال ورودی شکل (۳-۱) به فیلتر LWD مستقیم، خروجی فیلتر بصورت زیر خواهد بود:



شکل(۱۰-۵) خروجی فیلتر فیلتر LWD مستقیم به ازای ورودی نشان داده شده در شکل (۳-۱)

همانگونه که مشاهده می‌شود، فیلتر LWD آبشاری دو طبقه و فیلتر LWD مستقیم نیز در مرتبه های پایین عملیات فیلتر کردن را به خوبی انجام می‌دهند.

۶-۲-۱- آنالیز زمانی

برای قابل مقایسه بودن نتایج آنالیز زمانی لازم است که همه فیلترها در محیطی مشابه شبیه سازی شوند. در آنالیز زمانی صورت گرفته، محاسبات مربوط به تولید خروجی در حوزه زمان صورت می‌گیرد و این بخش توسط دستورات پایه برنامه MATLAB پیاده سازی شده است. در این آزمون برای تست زمانی فیلترها از تعداد زیادی نمونه ورودی استفاده می‌شود (حدود ۱۰ کیلو نمونه). سیگنال ورودی از ترکیب ۳ سیگنال سینوسی با فرکانس های ۴ هرتز، ۱۰۰ هرتز و ۲۱۰ هرتز تولید شده و با نرخ نمونه برداری ۱ کیلوهرتز نمونه برداری شده است شکل (۳-۱)). آنالیز زمانی با استفاده از عملیات دقت-نامحدود انجام می‌شود و محدودیتی بروی تعداد بیت های اعشاری و صحیح اعمال نشده است.

با شبیه سازی فیلترها و اعمال ورودی مذکور با ۱۰ کیلو نمونه، زمان مورد نیاز برای پردازش ۱۰ کیلو نمونه و تولید خروجی برای هر فیلتر به طور مجزا اندازه گیری و ثبت شد. عدد بدست آمده مقدار زمانی است که هر یک از فیلترها برای پردازش ۱۰ کیلو نمونه نیاز دارد. نتایج ثبت شده در آنالیز زمانی در جدول ۵ قابل مشاهده است.

جدول ۱۱ نتایج زمان اجرای هر فیلتر در آنالیز زمانی

نام فیلتر	زمان اجرا (میلی ثانیه)
فیلتر LWD مستقیم	۱/۹۱۱
فیلتر Window-based FIR	۷۳/۳۹۱
فیلتر Butterworth IIR	۷۱/۱
فیلتر LWD آبشاری دو طبقه	۳۲۱/۳۹۲

با مشاهده نتایج آنالیز زمانی، می‌توان مشاهده کرد که فیلتر LWD مستقیم دارای سرعت اجرای بسیار بالاتری نسبت به سایر فیلترهای شبیه سازی شده است و فیلتر LWD آبشاری دو طبقه به دلیل ساختار دو طبقه ای، دارای سرعت اجرای بسیار پایینتری نسبت به سایر فیلترهای شبیه سازی شده است.

۶-۲-۲- آنالیز حساسیت

برای آنالیز حساسیت یک فیلتر، آنرا در وضعیت دقت-محدود قرار داده سپس میزان انحراف خروجی فیلتر را از حالت دقت-نامحدود محاسبه می‌کنیم. در آنالیز حساسیت مد نظر این پایان نامه، ابتدا طول بیت های بخش اعشاری ضرایب را کاهش می‌دهیم، سپس میزان انحراف خروجی سیستم را از حالت دقت نامحدود محاسبه می‌کنیم. با در نظر گرفتن این انحراف به عنوان خطای خروجی سیستم، انرژی خطا را برای فیلتر مربوطه محاسبه می‌کنیم.

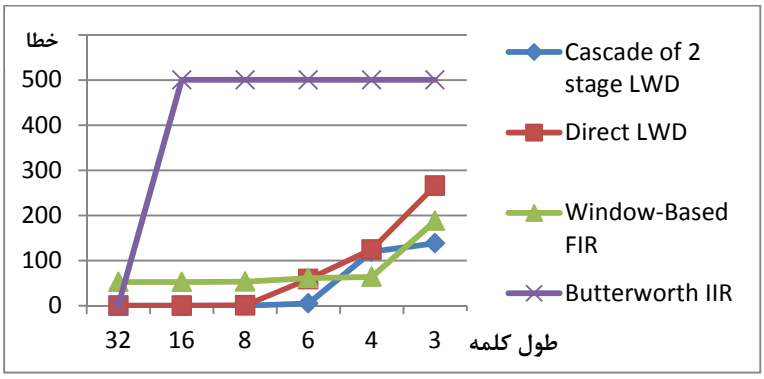
به ازای کاهش طول بیت های بخش اعشاری ضرایب، هر چه میزان انحراف خروجی فیلتر از حالت ممیز-نامحدود بیشتر باشد، خطای سیستم بیشتر خواهد بود. بنابراین در آزمونی مشابه برای چند فیلتر مختلف، هر چه میزان انرژی خطا برای یک فیلتر بیشتر باشد، این امر بیانگر آن است که فیلتر مربوطه دارای حساسیت بیشتری است، متقابلا هر چه میزان انرژی خطا برای فیلتری کمتر باشد، این امر بیانگر آن است که فیلتر مربوطه دارای حساسیت کمتری است.

با پیاده سازی فیلترهای طراحی شده در محیط MATLAB ، ورودی شکل (۳-۱) به فیلترها اعمال گردید. شبیه سازی مربوطه در حالت های مختلفی از دقت-محدود انجام شد. در هر حالت با کاهش هر چه بیشتر بیت های قسمت اعشاری ضرایب، خطای خروجی هر یک از فیلترها مستقلا محاسبه و ثبت شد. خطای خروجی فیلترها به ازای محدودیت ۳۲، ۱۶، ۸، ۶، ۴ و ۳ بیتی برای قسمت اعشاری ضرایب محاسبه شده و در جدول ۶ موجود است.

جدول ۱۲ خطای خروجی فیلترها درآنالیز حساسیت، به ازای طول بیت های مختلف برای قسمت اعشاری ضرایب

طول بیت های قسمت اعشاری ضرایب	۳۲	۱۶	۸	۶	۴	۳
خطای فیلتر LWD آبشاری دو طبقه	۰	۰	۰	۴/۸۰۷۵	۱۱۹/۳۶۳۳	۱۳۸/۳۲۶۴
خطای فیلتر LWD مستقیم	۰	۰	۰/۷۴۶۳	۵۸/۸۵۲	۱۲۴/۴۷۷۹	۲۶۵/۸۲۲۸
خطای فیلتر Window-based FIR	۵۲/۵۰۱۲	۵۲/۴۹۸۷	۵۳/۳۲۴۳	۶۰/۹۴۲۶	۶۳/۷۵	۱۸۷/۸۸
خطای فیلتر Butterworth IIR	۱/۱۲۰۴	۵۰۰/۴	۵۰۰/۴	۵۰۰/۴	۵۰۰/۴	۵۰۰/۴

با رسم نمودار خطای خروجی فیلترها نسبت به طول بیت های قسمت اعشاری ضرایب نمودار را خواهیم داشت.



شکل(۱۰-۶) نمودار خطای خروجی فیلترها نسبت به طول بیت های قسمت اعشاری ضرایب

همانگونه که در نموار شکل(۳-۶) مشاهده می‌شود، حساسیت فیلتر Butterworth IIR بسیار بالاست و به ازای دقت زیر ۲۷ بیت، میزان خطای خروجی بسیار زیاد می‌شود. فیلتر Window-based FIR نیز در مرتبه های پایین پاسخ مناسبی ندارد و در مرتبه ۹، به ازای تمام حالت‌ها خروجی فیلتر دارای خطاست. قبلا بیان شد ساختارهای LWD دارای حساسیت بسیار کمی در مقابل کاهش طول کلمات ضرایب خود دارند. در شکل می-بینیم که فیلتر LWD مستقیم تا دقت ۸ بیت و فیلتر LWD آبشاری دو طبقه تا دقت ۶ بیت، دارای خطای خروجی صفر است. خطای خروجی صفر به این معناست که فیلتر دارای خروجی مشابه خروجی حالت دقت-نامحدود است.

همانگونه که قبلا بیان شد، فیلترهای LWD آبشاری دارای حساسیت کمتری نسبت به فیلترهای LWD مستقیم هستند. بنابراین استفاده از ساختارهای LWD در پیاده سازی های عملی بسیار مناسبتر است، زیرا در دقت های بیتی محدود نیز پاسخی نزدیک به پاسخ حالت ایده آل خواهند داشت.

فصل چهارم:

پیاده سازی نرم افزاری

۱۰-۱- پردازنده های سیگنال دیجیتال

پردازنده سیگنال دیجیتال[۲۷] یک ریزپردازنده بهینه شده برای کاربردهای پردازش سیگنال دیجیتال می-باشد. پردازنده های سیگنال دیجیتال در انجام عملکردهای محاسباتی و پردازش سیگنال های دیجیتال دارای سرعت و کارایی بسیار بالایی هستند. ویژگی بارز پردازنده های سیگنال دیجیتال در آن است که آنها دارای تعدادی دستورالعمل خاص در زمینه پردازش سیگنالند که برخی از این عملکردها در گذشته به وسیله فیلترهای آنالوگ انجام می‌شدند.

بدلیل سرعت روز افزون تراشه های الکترونیکی و کارایی و قدرت تکنولوژی ای سی، پردازنده های سیگنال دیجیتال جای مدارهای آنالوگ را در بسیاری از حوزه ها مخابراتی، اتوماسیون و صنایع الکترونیکی گرفته است [۸۳،۸۹،۹۲]. پردازش سیگنال به کمک پردازنده های سیگنال تحولی عظیمی را در زمینه های فنی ایجاد کرده است. پیاده سازی های مبتنی بر پردازنده های سیگنال دارای انعطاف بسیار زیادی است، زیرا این امکان را به کاربر یا طراح میدهند که پس از پیاده سازی سیستم، پارامترهای سیستم را به راحتی و بدون صرف هزینه های گزاف، برحسب نیاز تغییر دهند.

پردازنده های سیگنال دیجیتال عملا از سال ۱۹۸۰ در زمینه های مختلف کاربرد پیدا کرد. از آن زمان تا کنون الگوریتم های پردازش تصویر، ویدیو و سیگنال روز به روز متنوع تر شدند و از طرفی نیاز های بشر نیز در زمینه های عملی مانند زمینه های نظامی و تجاری بیشتر شده است. پیاده سازی این الگوریتم ها در بسترهای سخت افزاری ضرورتی ایجاد کرد تا بتوانیم از الگوریتم های مختلف پردازش سیگنال در عمل بهره ببریم و کاربردهای بیشتری را پوشش دهیم، مخصوصا کاربردهای بلادرنگ.

[27] DSP: Digital Signal Processor

در میان کاربردهای نام برده شده، پردازنده های سیگنال دیجیتال در زمینه های مخابراتی جایگاه ویژه ای را دارند. امروزه بخش اعظمی از سیستم های مخابراتی بصورت دیجیتال می‌باشد و مخابرات دیجیتال هر روزه در زمینه های کاربردی گسترده تر می‌شود.

۱۰-۲- پردازنده سیگنال TMS320C5000

این پردازنده ساخت شرکت Texas Instruments و از گروه پردازنده های سیگنال دیجیتال ممیز ثابت می‌باشد. این پردازنده ها برای کاربردهای توان پایین طراحی شده اند. توان مصرفی این پردازنده در حالت فعال ۰,۱۵ mW/MHz و در حالت انتظار کمتر از ۰,۱۵ mW است. این پردازنده سیگنال برای کاربردهایی با پردازش های پیچیده در دستگاه های قابل حمل بهینه سازی شده است زیرا مساله اصلی در دستگاه های قابل حمل، توان مصرفی پایین و طول عمر باتری می‌باشد. این پردازنده در کاربرد هایی نظیر وسایل صوتی، حذف نویز، هدفون ها، رادیوها، الات موسیقی، دستگاه های پزشکی، میکروفون بیسیم، تلفن و ... استفاده می‌شود.

معماری پردازنده های سیگنال C55x دارای برتری های زیر است:

- بازدهی بالا
- توان مصرفی پایین
- قابلیت موازی سازی بالا

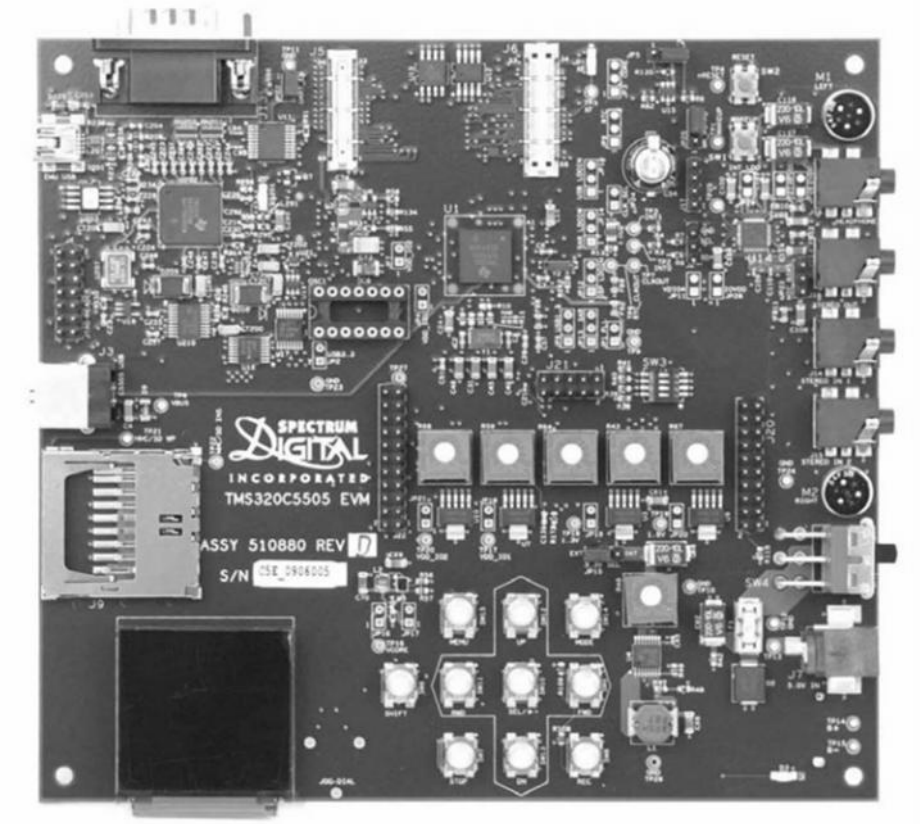
این پردازنده توسط برنامه CCS [۲۸] پشتیبانی می‌شود.

[28] Code Composer Studio

۱۰-۳- ماژول ارزیابی TMS320VC5505

ماژول ارزیابی TMS320VC5505 یا VC5505 EVM یک سطع گسترش مستقل استاندارد است که کاربر را قادر می‌سازد که طرح های خود را برای پردازنده TMS320VC5505 ارزیابی و گسترش دهند. علاوه براین، ماژول ارزیابی یا EVM به عنوان یک مرجع سخت افزاری برای پردازنده سیگنال TMS320VC5505 استفاده می‌شود.

تصویر ماژول ارزیابی TMS320VC5505 را در شکل (۴-۱) مشاهده می‌نمایید.



شکل(۱۴-۱) ماژول ارزیابی TMS320VC5505

همانگونه که در شکل (۴-۱) مشاهده می‌کنید، ماژول ارزیابی TMS320VC5505 دارای ابزارهای گوناگونی است که با استفاده از آن می‌توان قابلیت های مختلف پردازنده را ارزیابی و استفاده نمود. از ویژگی های ماژول ارزیابی می‌توان به موارد زیر ارشاره کرد:

- پردازنده سیگنال TMS320VC5505 محصول شرکت Texas Instruments که فرکانس کلاک آن تا MHz ۱۰۰ می‌رسد
- منبع توان قابل تغییر
- تراشه TLV320AIC3254 برای رمزگذاری و رمزگشایی صدای استریو
- رابط MMC / SD
- پورت USB توسط VC5505
- I^2c EEPROM با سرعت Kbits۲۵۶ و SPI EEPROM با سرعت Kbits۲۵۶
- واسط های EMIF، I2C، UART، SPI توسط رابط های توسعه
- واسط نمونه سازی JTAG خارجی
- کنترل کننده JTAG جاسازی شده
- نمایش دهنده LCD رنگی
- ۱۰ دکمه فشاری قابل برنامه ریزی

ماژول ارزیابی بمنظور کار کردن با نرم افزار CCS شرکت TI طراحی شده است. CCS به وسیله پایه های کابل JTAG یا JTAG جاسازی شده روی برد با ماژول ارزیابی ارتباط برقرار می‌کند. به همراه هر ماژول ارزیابی، یک سی دی حاوی نسخه ای سازگاز از برنامه CCS و درایور های مربوط به برد قرار دارد.

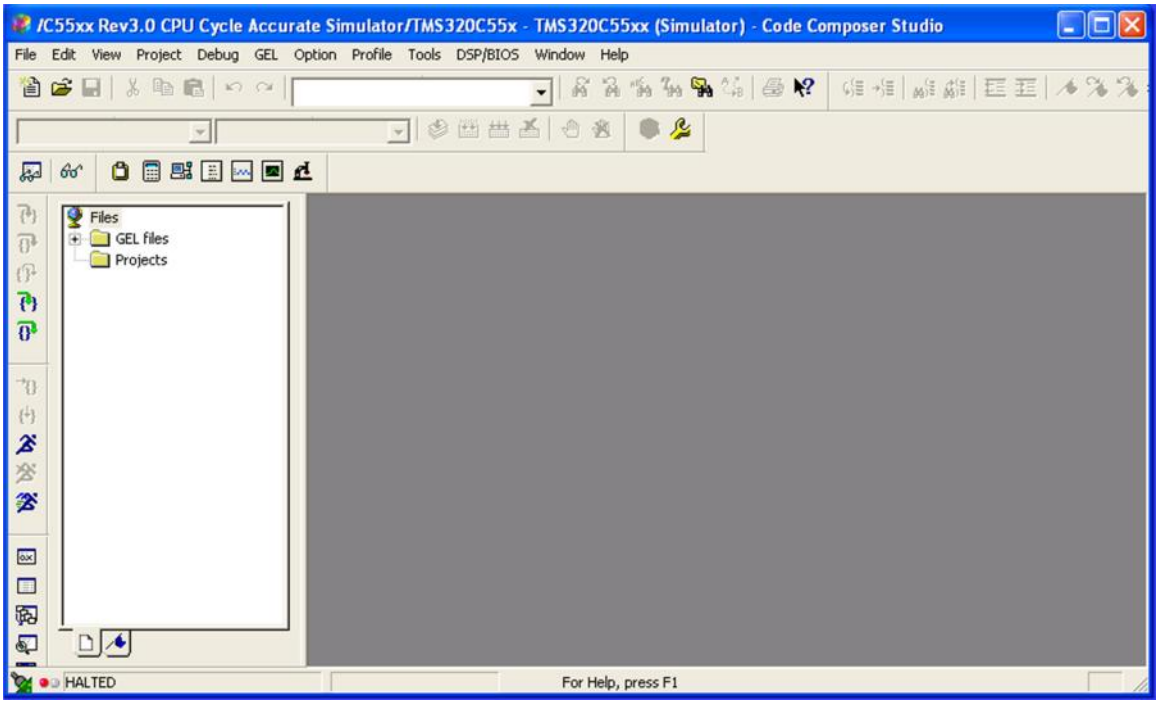
۱۰-۴- برنامه CCS

این برنامه یک محیط توسعه فشرده است برای توسعه کاربرد و توسعه پردازنده های جاسازی شده[۲۹] ساخت شرکت Texas Instruments. در شکل (۴-۲) ایکون های مربوط به نسخه ۳,۳ از برنامه را مشاهده می‌کنید.

شکل(۱۴-۲) آیکون های مربوط به برنامه CCS v3.3

پردازنده های جاسازی شده شرکت Texas Instruments شامل سخت افزار های خانواده DSP ها، ARM و پردازنده های دیگر مانند MSP430 می‌شود. برنامه CCS شامل سیستم عاملی بلادرنگ بنام DSP/BIOS یا DSP/BIOS می‌شود. برنامه CCS از سطح سیستم عاملی خطا یابی و حتی از سطح پایین آن یعنی محیط های مبتنی بر JTAG پشتیبانی می‌کند.

[29] Embedded processors



شکل(۱۴-۳) محیط نرم افزار CCS v3.3

CCS بروی چهارچوب برنامه متن-باز Eclipse ساخته شده است. در نسخه چهارمCCS بروی نسخه ای تغییر یافته از Eclipse بنا شده است. نسخه پنجم CCS از نسخه ای تغییر نیافته از Eclipse استفاده می‌کند و علاوه بر ویندوز، بروی Linux نیز قابل اجراست. نسخه های قبلی CCS از محیط های توسعه اختصاصی استفاده می‌کرد. در شکل (۴-۳) محیط برنامه CCS v3.3 را مشاهده می‌کنید.

نسخه ۳,۳ از برنامه CCS برای کار کردن با ماژول ارزیابی TMS320VC5505 مناسب است. در پیاده سازی انجام شده بروی ماژول ارزیابی TMS320VC5505، از این نسخه از برنامه استفاده شده است.

۱۰-۵- زبان اسمبلی

زبان اسمبلی یک زبان برنامه نویسی سطح-پایین محسوب می‌شود که بسیار به زبان ماشین نزدیک است. این زبان برای برنامه نویسی کامپیوترها، میکروکنترلرها، ریز پزدازنده ها و سایر سخت افزارهای قابل برنامه ریزی

است که در آن هر عبارت بیانگر یک دستور در زبان ماشین است. زبان اسمبلی توسط برنامه ای به نام اسمبلر به زبان ماشین ترجمه می‌شود.

۱۰-۶- دستورات یادمانی

در زبان اسمبلی از دستورات یادمانی[۳۰] برای نشان دادن عملگرهای ماشینی استفاده می‌شود. این عملیات ها در سیستم هدف بصورت یکتا هستند. بر اساس نحوه اسمبل شدن هر دستور، ممکن است که عملوند جزوی از دستور باشد یا نباشد. بیشتر اسمبلرها اجازه استفاده از برچسب ها و سمبل ها را برای آدرس دهی و ثابت ها می‌دهند. این کار باعث می‌شود که بجای استفاده مستقیم عملوندها در کد دستور،عملوندها بصورت سمبلیک قابل تعریف باشند.

۱۰-۷- پیاده سازی فیلتر LWD آبشاری دوطبقه با زبان اسمبلی

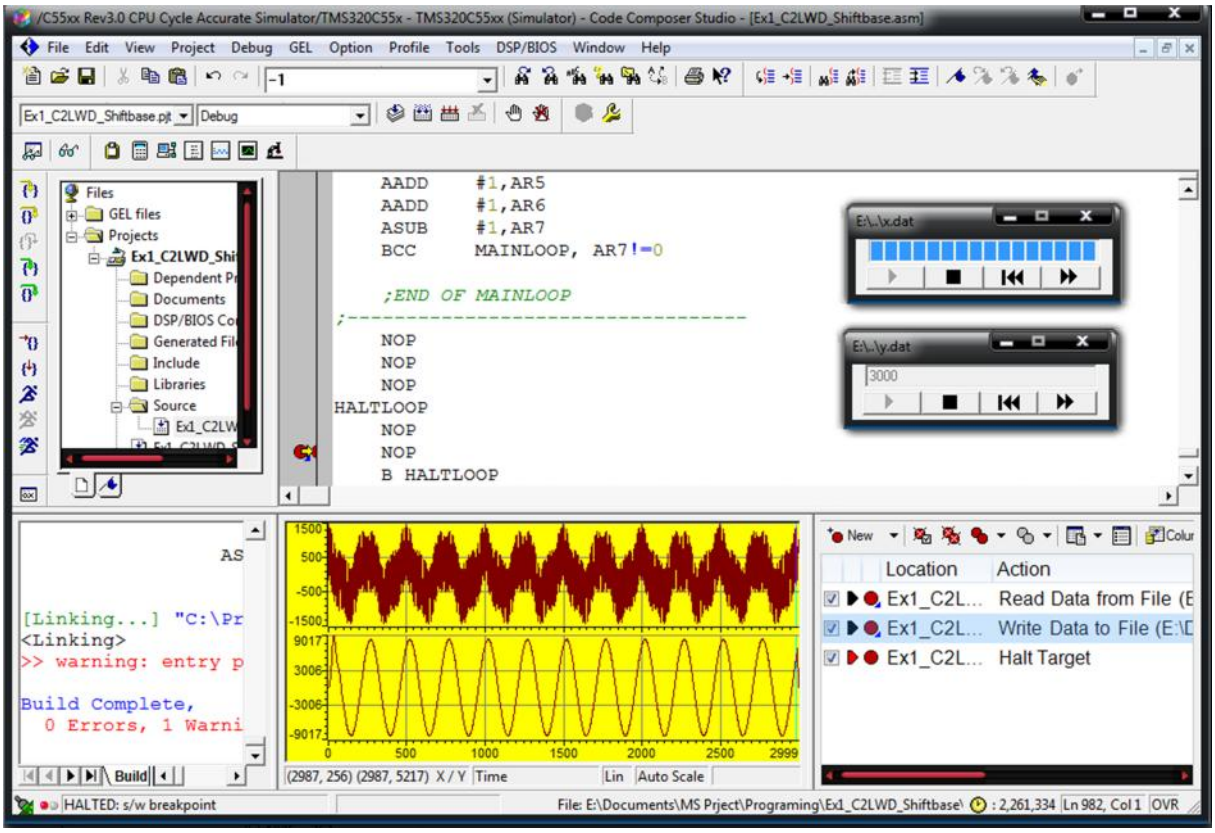
در این پایان نامه الگوریتم فیلتر LWD آبشاری دوطبقه را با استفاده از دستورات اسمبلی مربوط به پردازنده TMS320VC5505 پیاده سازی نمودیم. حاشیه باندگذر و حاشیه باندقطع برای این فیلترها بترتیب برابر است با $\omega_p = 0.05\pi$ و $\omega_s = 0.1\pi$. بیشینه ریپل مجاز برای باند گذر 0.5 dB و میزان تضعیف در باند قطع 100 dB است. همانگونه که مشاهده می‌شود، مشخصات گفته شده مربوط به فیلتر پایین گذر می‌شود.

همانگونه که در مطالب گذشته گفته شد، می‌توان ضرایب فیلتر را به گونه ای کوانتیزه کرد که در هنگام پیاده سازی فیلتر، بجای استفاده از عملگر ضرب از عملگرهای شیفت و جمع استفاده نماییم. در پیاده سازی انجام گرفته این کار انجام شده و برای نشان دادن این قابلیت مربوط به فیلتر LWD ، از هیچگونه عملگر ضربی استفاده نشده و تمام ضرب ها تنها با استفاده از عملگرهای شیفت علامتدار و جمع پیاده سازی شده است.

[30] Mnemonic

الگوریتم پیاده شده مذکور را در محیط برنامه CCS v3.3 شبیه سازی نمودیم. در شکل (۴-۴)، پیاده سازی و شبیه سازی فیلتر LWD آبشاری دوطبقه در محیط برنامه CCS v3.3 نشان داده شده است.

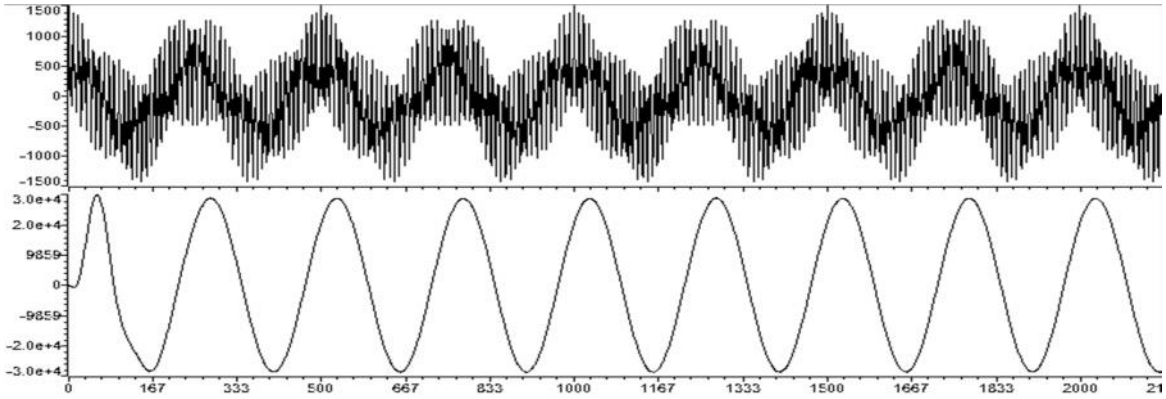
با قراردادن نمونه های سیگنال نشان داده شده در شکل (۴-۱) در یک فایل با پسوند dat. و معرفی فایل مذکور به عنوان سیگنال ورودی، برنامه CCS v3.3 محتویات فایل مذکور را خوانده و از محتویات آن به عنوان نمونه های سیگنال ورودی برای شبیه سازی فیلتر استفاده می‌نماید.



شکل(۴-۱۴) پیاده سازی فیلتر LWD آبشاری دوطبقه با زبان اسمبلی و شبیه سازی آن در برنامه CCS v3.3

از آنجا که فیلتر طراحی شده پایین گذر است، با اعمال سیگنال شکل (۴-۱) به عنوان ورودی فیلتر، انتظار می رود که فرکانس های بالا تر از فرکانی قطع (یعنی فرکانس های ۱۰۰ و ۲۱۰ هرتز) حذف شده، و تنها فرکانس

۴ هرتز را در خروجی فیلتر دیده شود. با انتخاب گزینه Graph از منوی View می‌توان سیگنال ورودی و خروجی حاصل از شبیه سازی را بصورت زیر رسم نماییم.

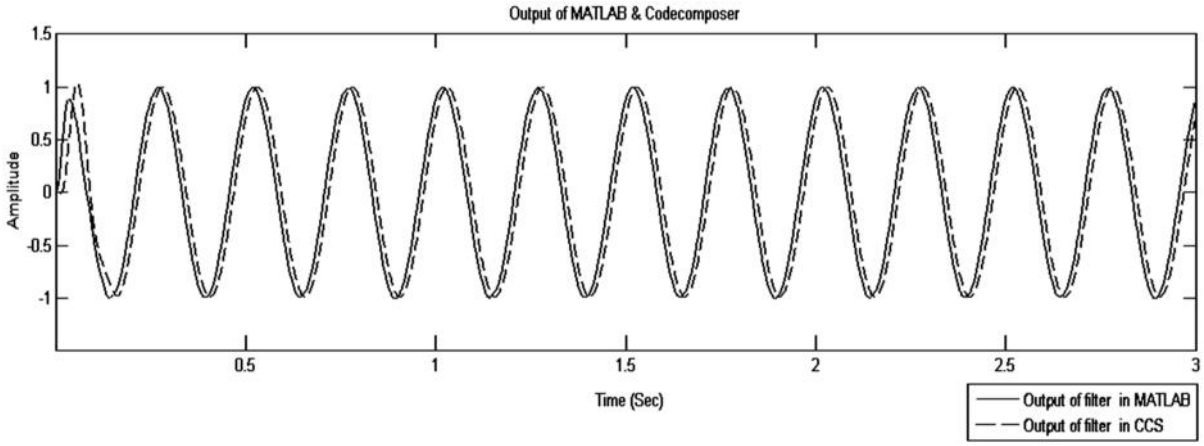


شکل(۱۴-۵) سیگنال ورودی (بالا) و خروجی (پایین) حاصل از شبیه سازی در محیط CCS v3.3

۱۰-۸- مقایسه نتایج شبیه سازی در محیط MATLAB و محیط CCS v3.3

نتایج حاصل از شبیه سازی فیلتر LWD آبشاری دوطبقه در محیط CCS v3.3 را در محیط MATLAB وارد می‌کنیم. نتایج شبیه سازی فیلتر LWD آبشاری دوطبقه در محیط MATLAB (که در به بخش ۴-۲ بیان شده) را به همراه نتایج حاصل از شبیه سازی سازی در محیط CCS v3.3 رسم می‌کنیم. شکل (۴-۶)

به این نکته توجه شود که شبیه سازی در محیط MATLAB در حالت دقت-نامحدود انجام شده است ولی شبیه سازی در محیط CCS v3.3 در بصورت تقریبا حقیقی و در حالت دقت-محدود انجام شده و بخشی از بیت های کم ارزشتر طی روند محاسبات حذف می‌شدند.



شکل(۱۴-۶) نتایج حاصل از شبیه سازی در محیط MATLAB با خط منحنی، و نتایج حاصل از شبیه سازی در محیط CCS با خط چین مشخص شده است.

حذف بیت های کم ارزشتر باعث ایجاد کمی انحراف در پاسخ فاز و پاسخ اندازه سیستم نسبت به حالت پیاده سازی ایده آل می‌شود. از آنجا که پیاده سازی در محیط CCS v3.3 در حالت دقت-محدود بوده، بنابراین مقدار کمی انحراف درپاسخ فاز و پاسخ اندازه سیستم نسبت به حالت شبیه سازی در محیط MATLAB بوجود خواهد آمد. زیرا شبیه سازی در محیط MATLAB تحت شرایط ایده آل انجام گرفته و هیچ بیتی طی محاسبات حذف نشده است.

انحراف پاسخ فاز سیستم، در حوزه زمان باعث ایجاد کمی شیفت زمانی در پاسخ زمانی سیستم می‌گردد که این امر در شکل (۴-۶) قابل مشاهده است.

باید توجه داشت که در پیاده سازی های سخت افزاری همواره محدودیت هایی از نظر طول کلمات ضرایب و تعداد بیت های اعشار و ... وجود دارد. بنابراین همیشه نتایج پیاده سازی های عملی مقداری با نتایج محاسبه شده در حالت ایده آل تفاوت دارد.

تاکید می‌شود که پیاده سازی بالا در محیط CCS v3.3 از عملگر ضرب استفاده نشده و در تمام محاسبات صورت گرفته به جای عملگر ضرب از عملگرهای شیف و جمع استفاده شده است. این کار برای نشان دادن توانایی ها و قابلیت های ارزشمند ساختارهای LWD در پیاده سازی های عملی صورت گرفته است.

۱۰-۹- بهینه سازی برنامه اسمبلی

برای بهینه سازی برنامه اسمبلی دو هدف وجود خواهد داشت:

۱. کاهش منابع مصرف شده
۲. افزایش سرعت اجرا

در پیاده سازی برنامه اسمبلی از همان ابتدا سعی برآن بود که حداقل منابع در پیاده سازی مصرف شود. بنابراین در این بخش تمرکز اصلی بروی افزایش سرعت اجرای الگوریتم است.

برای افزایش سرعت اجرای الگوریتم نیز دو راه باقی می‌ماند، یکی آنکه ساختار برنامه پیاده شده بهینه باشد و دیگری آنکه از پردازش موازی استفاده تماییم. با توجه به آنکه در پیاده سازی صورت گرفته برای فیلتر LWD آبشاری دوطبقه با زبان اسمبلی، ساختار برنامه به نحو بهینه ای پیاده شده ، بنابراین بهترین راه برای افزایش سرعت اجرای الگوریتم، استفاده از پردازش موازی است.

۱۰-۹-۱- استفاده از پردازش موازی

برای انجام پردازش موازی از دستوراتی استفاده می‌شود که قابلیت موازی کاری را با هم دارند. در پیاده سازی صورت گرفته برای فیلتر LWD آبشاری دوطبقه با زبان اسمبلی، ساختار الگوریتم به گونه ای پیاده شد که حداکثر قابلیت را برای انجام پردازش موازی داشته باشد. بنابراین برنامه اسمبلی را دوباره با استفاده از

دستورات پردازش موازی می‌نویسیم و در قدم بعدی، سرعت اجرای برنامه را در دو حالت قبل و بعد از بهینه سازی با هم مقایسه می‌کنیم.

۱۰-۱۰- مقایسه سرعت فیلتر قبل و بعد از بهینه سازی

برای مقایسه برنامه اسمبلی اولیه با برنامه بهینه شده توسط دستورات پردازش موازی، هر دو فیلتر را به ازای تعداد مشخصی از نمونه های ورودی در محیط CCS v3.3 شبیه سازی کرده و سپس تعداد کلاک های مصرفی را در هر دو حالت محاسبه نموده و با هم مقایسه می‌کنیم.

در مقایسه انجام شده، از هزار نمونه سیگنال نشان داده شده در شکل (۴-۱) به عنوان ورودی استفاده شده است. در پیاده سازی بهینه شده از دستورات پردازش موازی استفاده شده است اما در دیگری از پردازش موازی استفاده نشده است. نتایج شبیه سازی ها برای هر دو فیلتر در جدول ۷ جدول زیر نشان داده شده است.

جدول ۷ تعداد کلاک های مورد نیازی جهت پردازش هزار نمونه قبل و بعد از بهینه سازی

پیاده سازی فیلتر LWD آبشاری دوطبقه	تعداد کلاک ها لازم جهت پردازش هزار نمونه
برنامه اسمبلی قبل از بهینه سازی	۷۶۳۲۳۹
برنامه اسمبلی پس از بهینه سازی	۶۹۰۳۱۲

با توجه به جدول ۷ مشاهده می‌شود که با بهینه سازی صورت گرفته و موازی سازی ضرایب، به ازای پردازش هزار نمونه ورودی، ۷۲۹۲۷ کلاک کمتر مورد نیاز خواهد بود. این بدان معناست که با بهینه سازی برنامه، ۱۰ درصد سرعت پردازش نسبت به حالت قبل از بهینه سازی بهبود یافته است.

بنابراین به ازای فرکانس کلاک ۱۰۰ مگا هرتز برای پردازنده سیگنال، پردازش هزار نمونه ورودی ۰,۷۲۹۲۷ میلی ثانیه سریعتر انجام می‌شود. با توجه به سرعت بالای پردازنده های امروزی، ۰,۷۲۹۲۷ میلی

ثانیه زمان بسیار زیادی بحساب می‌آید. بنابراین استفاده از دستورات پردازش موازی نقش بسزایی در افزایش سرعت پردازش فیلتر خواهد داشت.

فصل پنجم:

پیاده سازی سخت افزاری

۱۵-۱- پیاده سازی سخت افزاری فیلتر بهینه شده

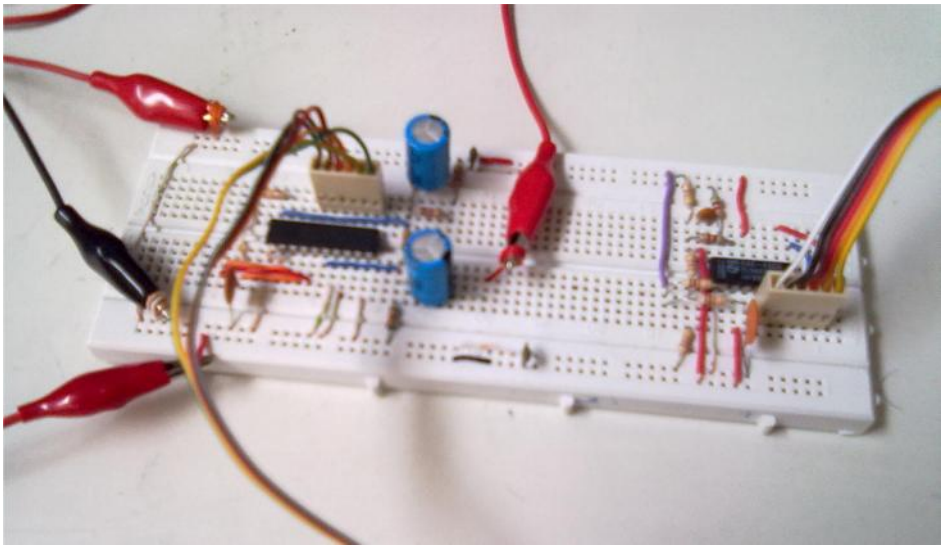
در فصل گذشته فیلتر LWD آبشاری دوطبقه را توسط دستورات زبان اسمبلی اسمبلی مربوط به پردازنده TMS320VC550 پیاده سازی نمودیم و در محیط CCS v3.3 شبیه سازی کرده و نتیجه شبیه سازی را با نتیجه شبیه سازی در MATLAB مقایسه نمودیم. پس از آن برنامه اسمبلی مربوطه را بهینه سازی نموده و به آن قابلیت پردازش موازی دادیم و سپس سرعت پردازش برنامه بهینه شده را با نتایج برنامه فاقد پردازش موازی مقایسه نمودیم.

در فصل جاری برنامه اسمبلی بهینه شده در فصل قبل را بروی ماژول ارزیابی TMS320VC5505 پیاده سازی کرده و سپس سیگنالی آنالوگ را توسط آن فیلتر نمودیم. پیاده سازی در این وضعیت بصورت سخت افزاری بوده و سیگنال های آنالوگ واقعی به سیستم داده می‌شود.

۱۵-۲- مدار تبدیل

برای تبدیل سیگنال آنالوگ به دیجیتال و سیگنال دیجیتال به آنالوگ، مداری را مطابق شکل (۵-۱) طراحی و پیاده نمودیم. این مدار تبدیل سیگنال آنالوگ ورودی را به سیگنالی دیجیتال تبدیل کرده و آنرا به ماژول ارزیابی TMS320VC5505 ارسال می‌کند. وهمچنین مدار تبدیل، سیگنال دیجیتالی ارسال شده توسط ماژول ارزیابی را به سیگنال آنالوگ تبدیل می‌کند.

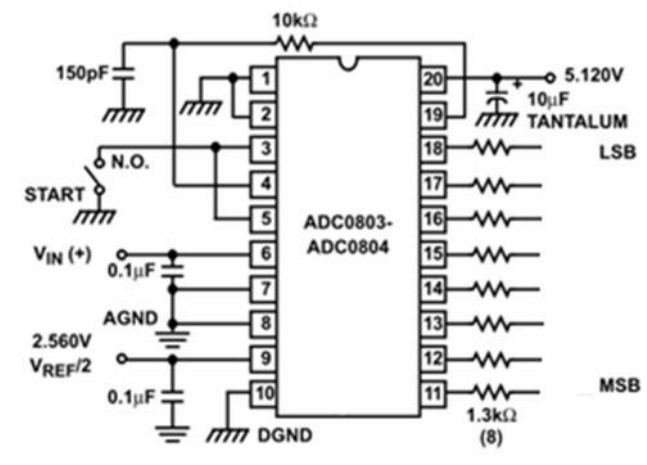
مدار تبدیل از دو بخش تشکیل شده است، یک مبدل آنالوگ به دیجیتال و یک مبدل دیجیتال به آنالوگ. در مدار آنالوگ به دیجیتال از تراشه ADD0804، و در مداردیجیتال به آنالوگ از تراشه DAC-08 استفاده شده است.



شکل(۱۹-۱) مدار تبدیل

۱۵-۲-۱- مبدل آنالوگ به دیجیتال

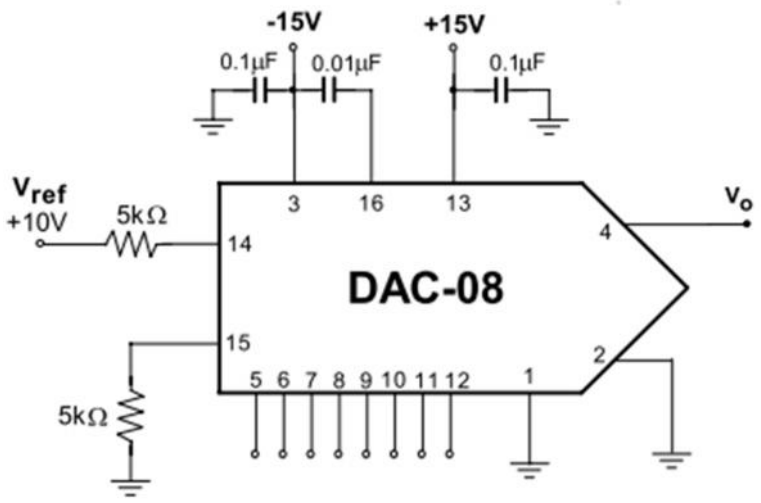
برای تبدیل سیگنال آنالوگ به دیجیتال از مبدل آنالوگ به دیجیتال ADD0804 استفاده شده است. این آی سی محصول شرکت National Semiconductor می‌باشد. تصویر مدار مربوطه را در شکل زیر می‌توان مشاهده نمود.



شکل(۱۹-۲) مدار مبدل آنالوگ به دیجیتال

۱۵-۲-۲- مبدل دیجیتال به آنالوگ

برای تبدیل سیگنال دیجیتال به آنالوگ از مبدل دیجیتال به آنالوگ DAC-08 استفاده شده است. این آی سی محصول شرکت National Semiconductor می‌باشد. تصویر مدار مربوطه را در شکل زیر می‌توان مشاهده نمود.



شکل(۱۹-۳) مدار مبدل مبدل دیجیتال به آنالوگ

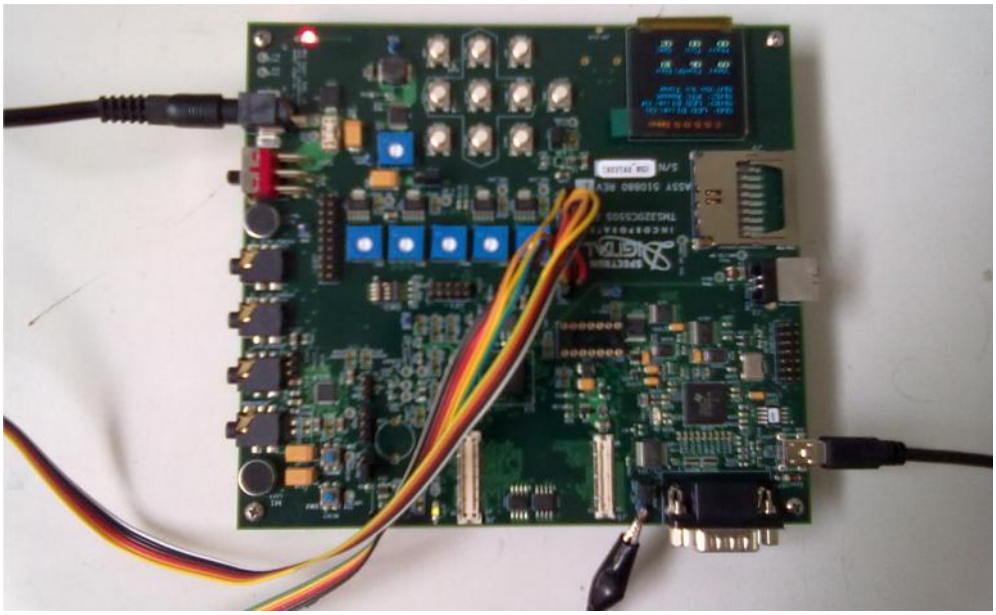
۱۵-۳- پیاده سازی سخت افزاری بلادرنگ

همانگونه که قبلا گفته شد دو نوع پیاده سازی وجود دارد، پیاده سازی نرم افزاری و پیاده سازی سخت افزاری. در کاربرد های بلادرنگ که در آن پردازش باید با نرخ بالایی صورت استفاه از پیاده سازی سخت افزاری اجباری است. در کاربرد هایی غیر بلادرنگ، جایی که داده ابتدا ذخیره شده و سپس پردازش می‌شوند، پیاده سازی نرم افزاری در بسیاری از موارد می‌تواند پاسخگو باشد.

در پیاده سازی سخت افزاری انجام شده، می‌خواهیم سیگنال ورودی توسط سیگنال ژنراتور تولید و همزمان سیگنال ورودی پردازش شود و خروجی فیلتر شده بصورت بلادرنگ تولید شود. سخت افزار مورد نظر برای این کار ماژول ارزیابی TMS320VC5505 است.

ماژول ارزیابی TMS320VC5505 برای ارتباط با کامپیوتر و برنامه CCS از کنترل کننده JTAG جاسازی شده و یک کابل USB استفاده می‌کند. یک سر کابل USB مربوطه را به کامپیوتر متصل کرده و سر دیگر آنرا را مطابق شکل (۵-۴) به ماژول ارزیابی متصل می‌کنیم. از طریق این کابل برنامه CCS قادر خواهد بود که پروژه های کامپایل شده خود را به روی ماژول ارزیابی بار گذاری کند و همچنین وضعیت ثبات‌ها و حافظه پردازنده سیگنال را بصورت همزمان بروی کامپیوتر نشان دهد.

پردازنده TMS320VC5505، سیگنال دیجیتال وردوی را توسط کانکتور J22 و بیت های ثبات GPIO[5:0] دریافت می‌کند. یک کابل اتصال بدین منظور ساخته شده است که مطابق شکل (۵-۴) بیت های سیگنال دیجیتال را بصورت صحیح به پین های روی ماژول ارزیابی انتقال می‌دهد.

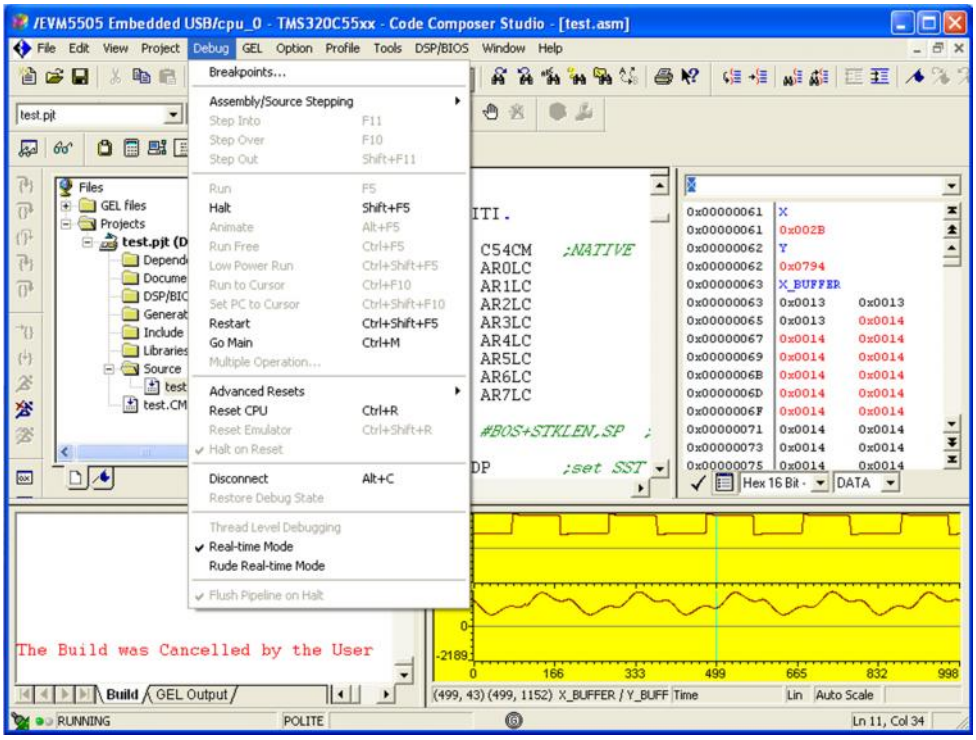


شکل(۱۹-۴) ماژول ارزیابی TMS320VC5505 و اتصالات مربوطه

برای تولید شکل موج سیگنال ورودی، از سیگنال ژنراتور و منبع تغذیه DC استفاده می‌کنیم. با استفاده از منبع تغذیه DC ولتاژی بایاس تولید کرده و سپس سیگنال متناوب تولید شده توسط سیگنال ژنراتور را بروی آن سوار می‌کنیم.

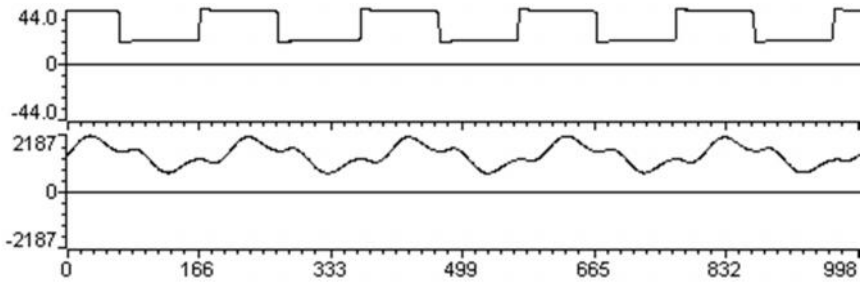
سیگنال آنالوگ مربوطه به مدار تبدیل رفته و در آنجا به سیگنال دیجیتال تبدیل می‌شود. سیگنال دیجیتال تولیدی توسط کابل به کانکتور J22 رفته و به وسیله ثبات GPIO[5:0] توسط پردازنده سیگنال دریافت می‌شود.

با اتصال ماژول ارزیابی به نرم افزار CCS 3.3v، فایل پروژه مربوط به برنامه اسمبلی فیلتر LWD آبشاری دوطبقه را که قبلا بهینه سازی نموده بودیم، باز کرده و کامپایل می‌کنیم. سپس برنامه کامپایل شده را بروی پردازنده TMS320VC5505 بارگذاری می‌کنیم. الگوریتم فیلتر در برنامه اسمبلی به گونه ای پیاده شده است که سیگنال دیجیتال ورودی را با نرخ نمونه برداری ۱ کیلو نمونه برداری کند و آن نمونه‌ها را به عنوان ورودی فیلتر پردازش نماید. با انتخاب گزینه Real-time Mode از منوی Debug، می‌توان نتایج حاصل از پردازش پردازنده سیگنال را به صورت بلادرنگ در برنامه CCS دید و نمودارهای مربوط به ورودی و خروجی را ترسیم کرد. شکل (۵-۵)



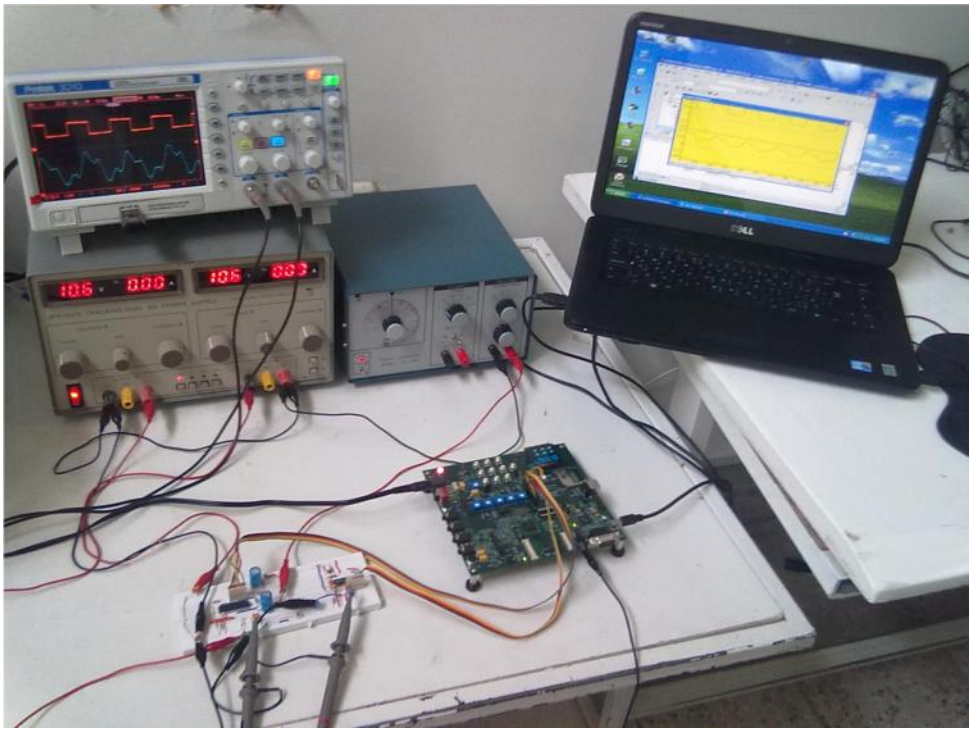
شکل(۱۹-۵) فعال کردن حالت بلادرنگ در برنامه CCS

با اعمال یک سیگنال مربعی با فرکانس ۵ هرتز توسط سیگنال ژنراتور، موج مربعی مربوطه دیجیتالی شده و توسط پردازنده سیگنال با فرکانس ۱ کیلو هرتز نمونه برداری می‌شود. نمونه های ورودی پردازش شده و فیلتر مربوطه بروی آن اعمال می‌شود. با رسم نمونه های سیگنال ورودی و خروجی در برنامه CCS ، نمودار شکل (۵-۶) را خواهیم داشت.



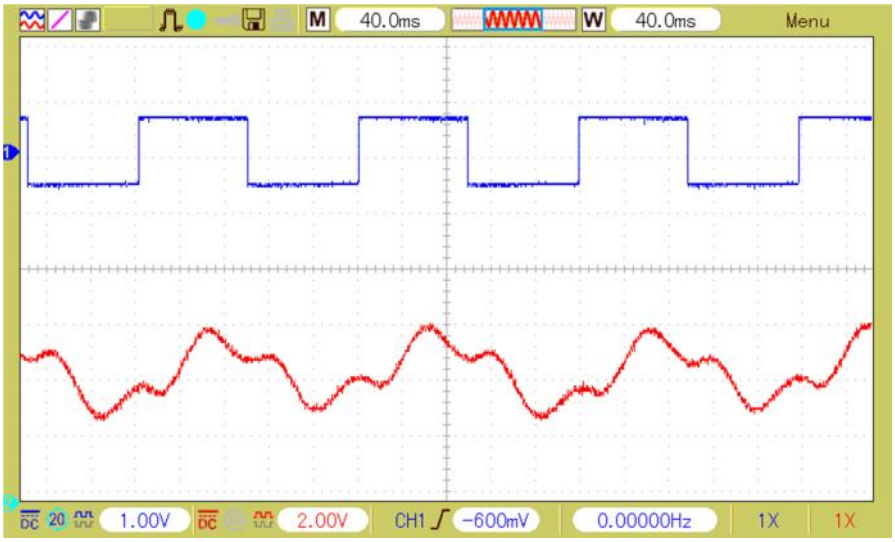
شکل(۱۹-۶) سیگنال خروجی (پایین) در برنامه CCS به ازای ورودی مربعی ۵ هرتز (بالا)

توسط ثبات های GPIO[11:6] وکانکتور J22، سیگنال خروجی فیلتر را به مدار تبدیل می‌فرستیم. مدار تبدیل سیگنال دیجیتال ارسالی توسط ماژول ارزیابی را به سیگنال آنالوگ تبدیل می‌کند. سیگنال های آنالوگ خروجی را می‌توان مطابق شکل (۵-۷) به اسیلوسکوپ داد و توسط آن سیگنالهای ورودی و خروجی را همزمان رسم نمود.



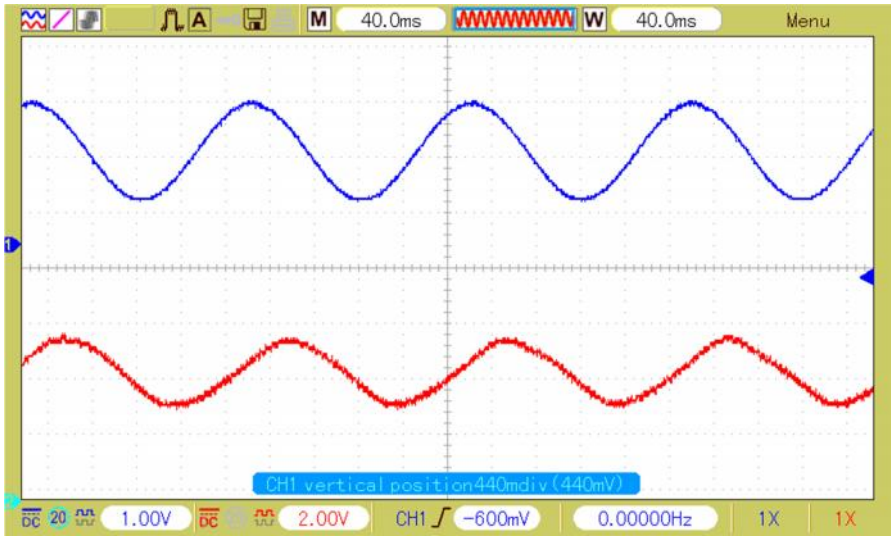
شکل(۱۹-۷) پیاده سازی سخت افزاری فیلتر LWD آبشاری دوطبقه

اسیلوسکوپ سیگنالهای آنالوگ ورودی و خروجی را دریافت می‌کند و از آنجا که عملیات فیلترینگ بصورت بلادرنگ پیاده سازی شده است، ورودی و خروجی فیلتر را بصورت همزمان رسم می‌کند. شکل (۵-۸)



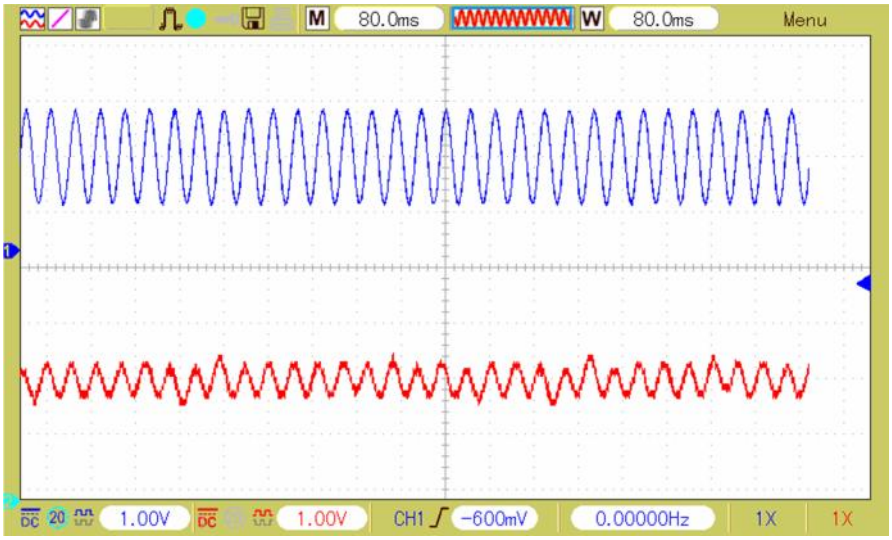
شکل(۱۹-۸) خروجی (پایین) فیلتر به ازای ورودی پالس ۵ هرتز (بالا)

برای دیدن نحوه عمکرد فیلتر پایین گذر در فرکانس های مختلف،و سیگنال هایی سینوسی با فرکانس هایی مختلف را به عنوان ورودی به فیلتر پیاده سازی شده می‌دهیم. ابتدا یک از سیگنال سینوسی ۵ هرتز آغاز می‌کنیم. با اعمال این سیگنال سینوسی به مدار تبدیل، مارژول ارزیابی پس از پردازش ورودی، سیگنال خروجی مربوطه را تولید می‌کند. ورودی و خروجی فیلتر مطابق شکل (۵-۹) می‌شود.



شکل(۱۹-۹) خروجی (پایین) فیلتر به ازای ورودی سینوسی ۵ هرتز (بالا)

و به ازای سیگنال سینوسی ورودی با فرکانس ۲۰ هرتز ورودی و خروجی فیلتر به صورت شکل (۵-۱۰) می‌شود.

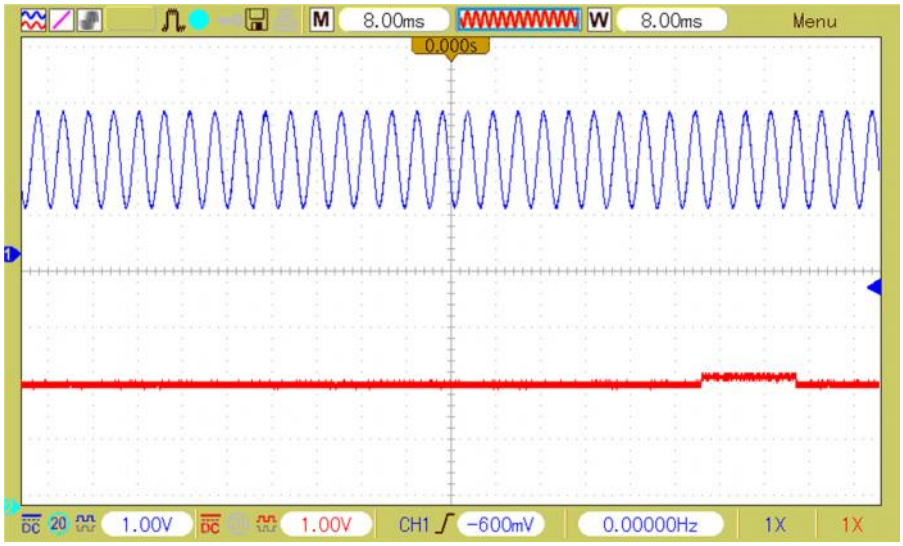


شکل(۱۹-۱۰) خروجی (پایین) فیلتر به ازای ورودی پالس ۲۰ هرتز (بالا)

و به ازای سیگنال سینوسی ورودی با فرکانس ۸۰ هرتز ورودی و خروجی فیلتر به صورت شکل (۵-۱۱) می‌شود.

شکل(۱۹-۱۱) خروجی (پایین) فیلتر به ازای ورودی پالس ۸۰ هرتز (بالا)

و به ازای سیگنال سینوسی ورودی با فرکانس ۲۰۰ هرتز ورودی و خروجی فیلتر به صورت شکل (۵-۱۲) می‌شود.



شکل(۱۹-۱۲) خروجی (پایین) فیلتر به ازای ورودی پالس ۲۰۰ هرتز (بالا)

از آنجا که به ازای فرکانس نمونه برداری ۱ کیلو هرتز، فرکانس قطع باند گذر و فرکانس قطع باند قطع برای فیلتر پایین گذر مربوطه بترتیب ۲۶ هرتز و۵۰ هرتز می‌شود. با مشاهده شکل (۵-۹) تا شکل (۵-۱۲) می‌توان نتیجه گرفت که این فیلتر فرکانس های کوچکتر از ۲۰ را بخوبی عبور می‌دهد. اما با نزدیک شدن به فرکانس ۲۰ هرتز، اندازه سیگنال شروع به تضعیف شدن می‌کند و در فرکانس های بالاتر از ۵۰، این تضعیف بسیار زیاد شده و تقریبا خروجی صفر می‌شود. در فرکانس های بالاتر با افزایش فرکانس، میزان تضعیف باند قطع فیلتر بیشتر و بیشتر می‌شود.

۱۵-۴- آنالیز حساسیت در پیاده سازی سخت افزاری

در ادامه قصد داریم تا آزمون حساسیت را برای فیلتر LWD آبشاری دوطبقه در پیاده سازی سخت افزاری انجام دهیم. برای این کار فیلتر به گونه ای پیاده سازی شده که ضرایب آن بصورت زمان-حقیقی قابل تغییر باشد. در این پیاده سازی سخت افزاری از ماژول ارزیابی TMS320VC5505 و برنامه CCS v3.3 استفاده شده است. مانند بخش قبل فیلتر را اجرا کرده و

با تغییر ضرایب به صورت زمان-حقیقی می‌توان صفرها و قطب‌ها فیلتر را بصورت ب زمان-حقیقی تنظیم نمود و نقطه کار فیلتر را با بابجا کردن صفرها و قطب‌ها بهینه کرد. بنابراین با متغیر تعریف تمودن ضرایب فیلتر در هنگام پیاده سازی فیلتر، می‌توان در هنگام اجرا ضرایب را بدون قطع فرایند پردازش، تغییر داد.

۱۵-۴-۱- آنالیز حساسیت فیلتر LWD آبشاری دوطبقه

در آنالیز حساسیتی که مد نظر داریم، ابتدا حالتی را اجرا می‌کنیم که محدودیتی بروی ضرایب قرار داده نشده است. سپس با کاهش طول کلمات ضرایب به ازای حالت های ۸، ۶، ۴ و ۳ بیت، فیلتر را در هر حالت اجرا کرده و میزان انحراف خروجی در هر حالت را نسبت به خروجی در حالت بدون محدودیت محاسبه می‌کنیم.

از آنجا که فرم کلی ضرایب فیلتر بصورت $\pm 1 \pm 2^{p_{r1}} \pm 2^{p_{r2}}$ است، با کاهش و افزایش توان های Pr1 و Pr2 می‌توان طول کلمات ضرایب را کنترل نمود. به مثال زیر توجه فرمایید:

```
>> dec2bin(( 1-2^(-3)-2^(-7))*256)    %8bit
        ans =11011110
>> dec2bin(( 1-2^(-3)-2^(-6))*256)    %6bit
        ans =11011100
>> dec2bin(( 1-2^(-3)-2^(-4))*256)    %4bit
        ans =11010000
>> dec2bin(( 1-2^(-3)-2^(-3))*256)    %3bit
        ans =11000000
```

برای تولید ورودی فیلتر، یک سیگنال سینوسی با فرکانس ۵ هرتز و دامنه ۱ ولت، و یک سیگنال سینوسی با فرکانس ۱۰۰ و دامنه ۰/۵ ولت را توسط دو عدد سیگنال ژنراتور تولید می‌کنیم. این دوسیگنال سینوسی را

به همراه یک ولتاژ مستقیم ۳ ولتی با هم جمع کرده، و سیگنال مجموع حاصل را به عنوان ورودی به فیلتر مربوطه می‌دهیم.

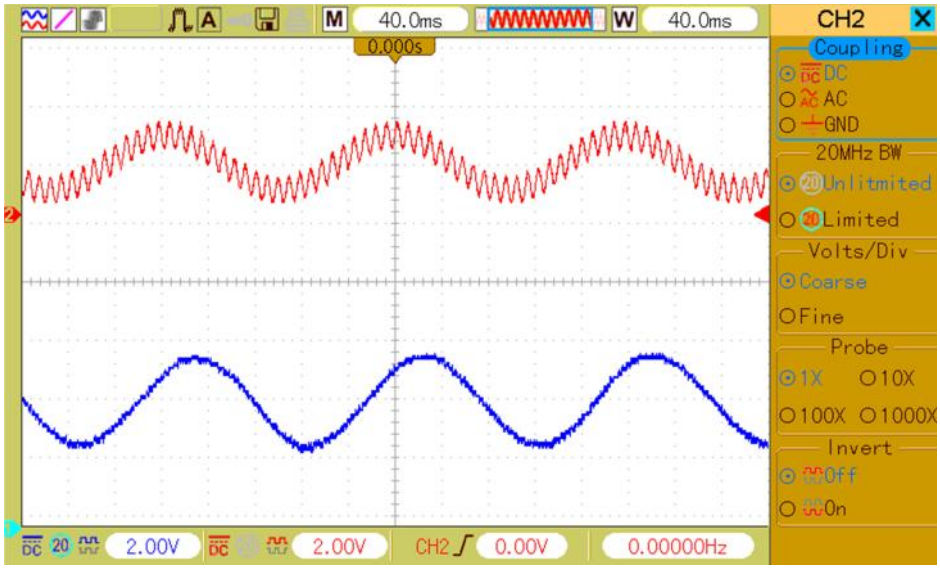
از آنجا که ورودی فیلتر متشکل از سیگنالهایی متناوب سینوسی با فرکانس های ۱۰۰هرتز و ۵ هرتز است، انتظار داریم که فیلتر پایین گذر LWD آبشاری دوطبقه، موج سینوسی با فرکانس بالاتر را در خروجی حذف نماید.

در شروع کار ابتدا فیلتر را در حالت بدون محدودیت اجرا می‌کنیم. ضرایب فیلتر در حالت بدون محدودیت را می‌توان در جدول ۸ مشاهده نمود.

جدول ۸ ضرایب فیلتر در حالت بدون محدودیت

$A_k(z)$		$B_k(z)$	
$\gamma_0^{(1)}$	$1-2^{-3}-2^{-3}$		
$\gamma_1^{(1)}$	$-1+2^{-4}+2^{-8}$	$\hat{\gamma}_1^{(1)}$	$-1+2^{-2}-2^{-4}$
$\gamma_2^{(1)}$	$1-2^{-6}-2^{-8}$	$\hat{\gamma}_2^{(1)}$	$1-2^{-6}$
$\gamma_0^{(2)}$	$1-2^{-3}+2^{-5}$		
$\gamma_1^{(2)}$	$-1+2^{-5}+2^{-7}$	$\hat{\gamma}_1^{(2)}$	$-1+2^{-3}$
$\gamma_2^{(2)}$	$1-2^{-6}-2^{-8}$	$\hat{\gamma}_2^{(2)}$	$1-2^{-6}+2^{-8}$

با اجرای الگوریتم فیلترینگ، و فیلتر کردن ورودی با ضرایب موجود در جدول ۸، ورودی و خروجی فیلتر بصورت شکل (۵-۱۳) خواهد بود.



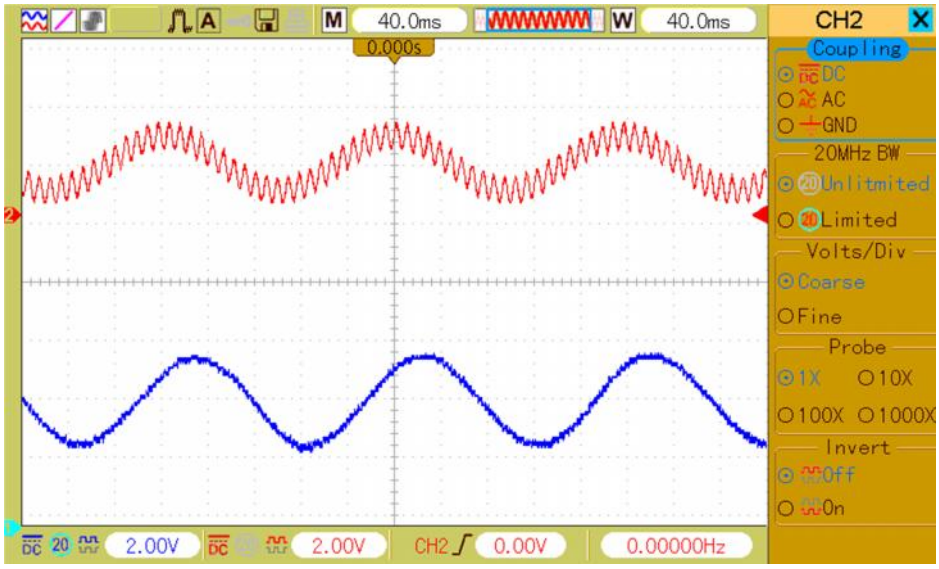
شکل(۱۹-۱۳) ورودی (بالا) و خروجی (پایین) فیلتر در حالت بدون محدودیت

همچنین ضرایب فیلتر در حالتی که محدودیت ۸ بیت برای ضرایب وجود دارد، بصورت جدول زیر می‌باشد:

جدول ۹ ضرایب فیلتر در حالت محدودیت ۸ بیت برای ضرایب

$A_k(z)$		$B_k(z)$	
$\gamma_0^{(1)}$	$1-2^{-3}-2^{-7}$		
$\gamma_1^{(1)}$	$-1+2^{-4}+2^{-8}$	$\hat{\gamma}_1^{(1)}$	$-1+2^{-2}-2^{-4}$
$\gamma_2^{(1)}$	$1-2^{-6}-2^{-8}$	$\hat{\gamma}_2^{(1)}$	$1-2^{-6}$
$\gamma_0^{(2)}$	$1-2^{-3}+2^{-5}$		
$\gamma_1^{(2)}$	$-1+2^{-5}+2^{-7}$	$\hat{\gamma}_1^{(2)}$	$-1+2^{-3}$
$\gamma_2^{(2)}$	$1-2^{-6}-2^{-8}$	$\hat{\gamma}_2^{(2)}$	$1-2^{-6}+2^{-8}$

سپس برای ایجاد محدودیت ۸ بیت بروی ضرایب، ضرایب فیلتر را مطابق جدول ۹ تنظیم می‌کنیم. در این حالت ورودی و خروجی فیلتر بصورت شکل (۵-۱۴) خواهد بود.



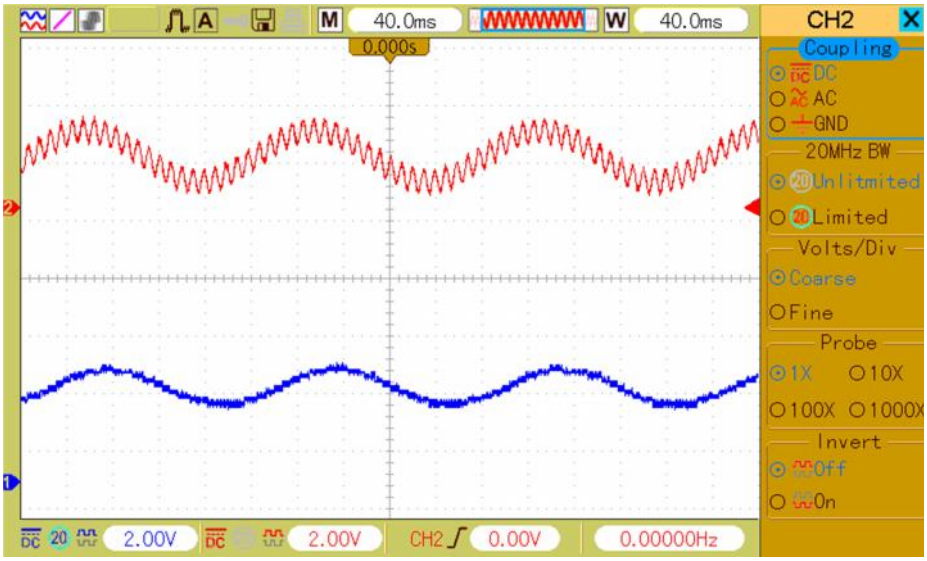
شکل(۱۹-۱۴) ورودی (بالا) و خروجی (پایین) فیلتر در حالت محدودیت ۸ بیت برای ضرایب

همچنین ضرایب فیلتر در حالتی که محدودیت ۶ بیت برای ضرایب وجود دارد، بصورت جدول ۳ می‌باشد:

جدول ۱۰ ضرایب فیلتر در حالت محدودیت ۶ بیت برای ضرایب

$A_k(z)$		$B_k(z)$	
$\gamma_0^{(1)}$	$1-2^{-3}-2^{-6}$		
$\gamma_1^{(1)}$	$-1+2^{-4}+2^{-6}$	$\hat{\gamma}_1^{(1)}$	$-1+2^{-2}-2^{-4}$
$\gamma_2^{(1)}$	$1-2^{-6}-2^{-6}$	$\hat{\gamma}_2^{(1)}$	$1-2^{-6}$
$\gamma_0^{(2)}$	$1-2^{-3}+2^{-5}$		
$\gamma_1^{(2)}$	$-1+2^{-5}+2^{-6}$	$\hat{\gamma}_1^{(2)}$	$-1+2^{-3}$
$\gamma_2^{(2)}$	$1-2^{-6}-2^{-6}$	$\hat{\gamma}_2^{(2)}$	$1-2^{-6}+2^{-6}$

سپس برای ایجاد محدودیت ۶ بیت بروی ضرایب، ضرایب فیلتر را مطابق جدول ۱۰ تنظیم می‌کنیم. در این حالت ورودی و خروجی فیلتر بصورت شکل (۵-۱۵) خواهد بود.



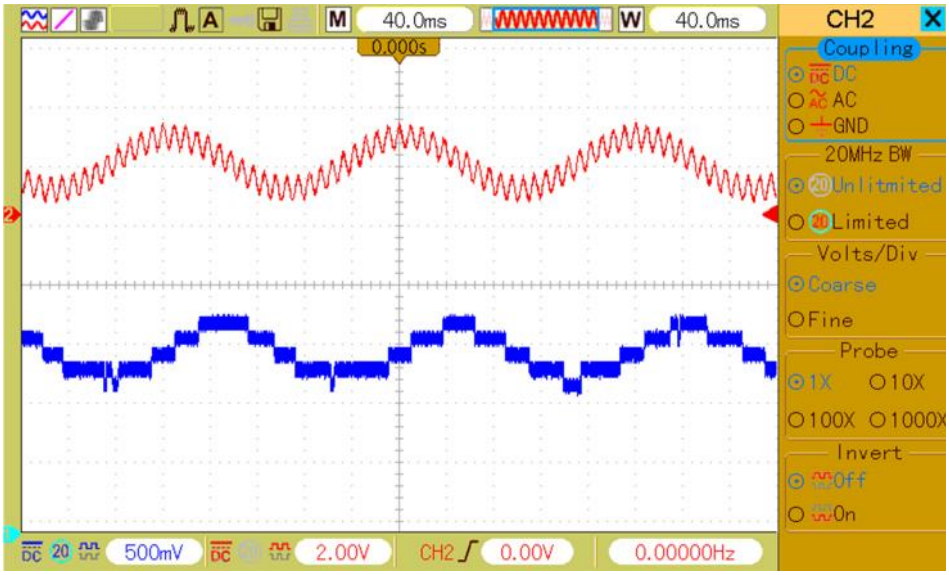
شکل(۱۹-۱۵) ورودی (بالا) و خروجی (پایین) فیلتر در حالت محدودیت ۶ بیت برای ضرایب

همچنین ضرایب فیلتر در حالتی که محدودیت ۴ بیت برای ضرایب وجود دارد، بصورت جدول ۴ می‌باشد:

جدول ۱۱ ضرایب فیلتر در حالت محدودیت ۴ بیت برای ضرایب

$A_k(z)$		$B_k(z)$	
$\gamma_0^{(1)}$	$1-2^{-3}-2^{-4}$		
$\gamma_1^{(1)}$	$-1+2^{-4}+2^{-4}$	$\hat{\gamma}_1^{(1)}$	$-1+2^{-2}-2^{-4}$
$\gamma_2^{(1)}$	$1-2^{-5}-2^{-5}$	$\hat{\gamma}_2^{(1)}$	$1-2^{-4}$
$\gamma_0^{(2)}$	$1-2^{-2}+2^{-3}$		
$\gamma_1^{(2)}$	$-1+2^{-5}+2^{-5}$	$\hat{\gamma}_1^{(2)}$	$-1+2^{-3}$
$\gamma_2^{(2)}$	$1-2^{-5}-2^{-5}$	$\hat{\gamma}_2^{(2)}$	$1-2^{-3}+2^{-4}$

سپس برای ایجاد محدودیت ۴ بیت بروی ضرایب، ضرایب فیلتر را مطابق جدول ۱۱ تنظیم می‌کنیم. در این حالت ورودی و خروجی فیلتر بصورت شکل (۵-۱۶) خواهد بود.



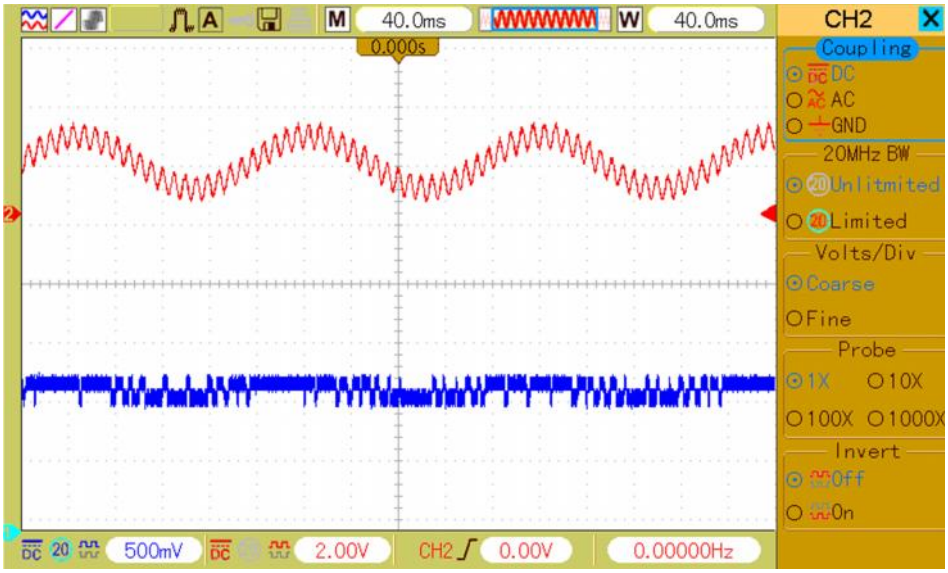
شکل(۱۹-۱۶) ورودی (بالا) و خروجی (پایین) فیلتر در حالت محدودیت ۴ بیت برای ضرایب

همچنین ضرایب فیلتر در حالتی که محدودیت ۳ بیت برای ضرایب وجود دارد، بصورت جدول زیر می‌باشد:

جدول ۱۲ ضرایب فیلتر در حالت محدودیت ۳ بیت برای ضرایب

$A_k(z)$		$B_k(z)$	
$\gamma_0^{(1)}$	$1-2^{-3}-2^{-3}$		
$\gamma_1^{(1)}$	$-1+2^{-4}+2^{-4}$	$\hat{\gamma}_1^{(1)}$	$-1+2^{-1}-2^{-2}$
$\gamma_2^{(1)}$	$1-2^{-4}-2^{-4}$	$\hat{\gamma}_2^{(1)}$	$1-2^{-3}$
$\gamma_0^{(2)}$	$1-2^{-1}+2^{-2}$		
$\gamma_1^{(2)}$	$-1+2^{-4}+2^{-4}$	$\hat{\gamma}_1^{(2)}$	$-1+2^{-3}$
$\gamma_2^{(2)}$	$1-2^{-4}-2^{-4}$	$\hat{\gamma}_2^{(2)}$	$1-2^{-2}+2^{-3}$

سپس برای ایجاد محدودیت ۳ بیت بروی ضرایب، ضرایب فیلتر را مطابق جدول ۱۲ تنظیم می‌کنیم. در این حالت ورودی و خروجی فیلتر بصورت شکل (۵-۱۷) خواهد بود.



شکل(۱۹-۱۷) ورودی (بالا) و خروجی (پایین) فیلتر در حالت محدودیت ۳ بیت برای ضرایب

۱۵-۴-۲- محاسبه انحراف خروجی در حالت های مختلف

با اعمال محدودیت بیشتر بروی طول کلمات ضرایب، میزان انحراف خروجی فیلتر از حالت بدون محدودیت بیشتر می‌شود. انحراف خروجی در هر حالت نشان دهنده میزان خطای فیلتر خواهد بود. با محاسبه میزان انرژی خطا می‌توان عددی بدست آورد که بیانگر میزان انحراف خروجی است. خطای خروجی فیلتر خطای فیلتر LWD آبشاری دو طبقه به ازای محدودیت های ۸، ۶، ۴ و ۳ بیتی محاسبه شده و در جدول ۱۳ موجود است. این خطا به ازای یک دوره سیگنال خروجی محاسبه شده است.

جدول۱۳ خطای خروجی فیلترها به ازای محدودیت های مختلف برای طول بیت های اعشاری

طول بیت های قسمت اعشاری ضرایب	۳۲	۱۶	۸	۶	۴	۳
خطای فیلتر LWD آبشاری دو طبقه	۰	۰	۰	۰/۳۹۱۶	۲۸/۵۸۹۰	۳۳۲/۴۸۸۷

فهرست منابع

[1] Yli-Kaakinen, J. (2002). Optimization of Digital Filters for Practical Implementations, Tampere University of Technology. Doctor.

[2] A. Antoniou, “Improved minimax optimization algorithm and their application in the design of recursive digital filters,” IEE Proc. Part G, Circuits, Devices, Syst., vol. 138, no. 6, pp. 724–730, Dec. 1991.

[3] E. C. Ifeachor and B. W. Jervis, Digital Signal Processing, A Practical Approach. Addison-Wesley, 1993.

[4] L. Rabiner and B. Gold, Theory and Application of Digital Signal Processing. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[5] M. Renfors, On the Design of Efficient Digital Filters Using Graph Theoretic Equivalence Transformations, Dr. Tech. dissertation, Dept. of Electrical Engineering, Tampere University of Technology, Finland, 1982.

[6] T. Saram¨aki, Design of Digital Filters Requiring a Small Number of Arithmetic Operations, Dr. Tech. dissertation, Dept. of Electrical Engineering, Tampere University of Technology, Finland, 1981.

[7] T. Saram¨aki, “Finite impulse response filter design,” in Handbook for Digital Signal Processing, S. K. Mitra and J. F. Kaiser, Eds., chapter 4, pp. 155–277. New York: John Wiley and Sons, 1993.

[8] L. Wanhammar and H. Johansson, Digital filters. Link¨oping University, 1997.

[9] Y. C. Lim, “Efficient special purpose linear programming for FIR filter design,” IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-31, pp. 963–968, 1983.

[10] Y. Neuvo, “Digital filter implementation considerations,” in Handbook for Digital Signal Processing, S. K. Mitra and J. F. Kaiser, Eds., chapter 6, pp. 337– 417. New York: John Wiley and Sons, 1993.

[11] T. W. Parks and J. H. McClellan, “Chebyshev approximation for nonrecursive digital filters with linear phase,” IEEE Trans. Circuit Theory, vol. CT-19, pp. 189–194, 1972.

[12] L. R. Rabiner, “Linear program design of finite impulse response (FIR) digital filters,” IEEE Trans. Audio Electroacoust., vol. AU-20, pp. 280–288, Oct. 1972.

[13] J. H. McClellan and T. W. Parks, “A unified approach to the design of optimum FIR linear phase digital filters,” IEEE Trans. Signal Processing, vol. CT-20, pp. 697–701, Nov. 1973.

[14] J. H. McClellan, T. W. Parks, and L. R. Rabiner, “A computer program for designing optimum FIR linear phase digital filters,” IEEE Trans. Audio Electroacoust., vol. AU-21, pp. 506–526, Dec. 1973.

[15] C.-Y. Tseng, “A numerical algorithm for complex Chebyshev FIR filter design,” in Proc. IEEE Int. Symp. Circuits Syst., San Diego, CA, May 1992, vol. 2, pp. 549–552.

[16] M. Schulist, “Improvements of a complex FIR filter design algorithm,” IEEE Trans. Signal Processing, vol. SP-20, pp. 81–90, 1990.

[17] K. Preuss, “On the design of FIR filters by complex Chebyshev approximation,” IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-37, pp. 702–712, May 1989.

[18] H.-K. Kwan, “On the problem designing IIR digital filters with short coefficient word lengths,” IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-27, no. 6, pp. 620–624, Dec. 1979.

[19] D. R. Bull and D. H. Horrocks, “Primitive operator digital filters,” Proc. Inst. Elect. Eng., Pt. G, vol. 138, pp. 401–411, June 1991.

[20] A. G. Dempster and M. D. Macleod, “General algorithms for reduced-adder integer multiplier design,” Electron. Lett., vol. 31, pp. 1800–1802, Oct. 1995.

[21] R. I. Hartley, “Subexpression sharing in filters using canonic signed digit multipliers,” IEEE Trans. Circuits Syst. II, vol. 43, pp. 677–688, Oct. 1996.

[22] A. Matsuura, M. Yukishita, and A. Nagoya, “A hierarchical clustering method for the multiple constant multiplication problem,” IEICE Trans. Fund., vol. E80–A, pp. 1767–1773, Oct. 1997.

[23] R. P˘asko, P. Schaumont, V. Derudder, S. Vernalde, and D. ˘Dura˘ckov´a, “A new algorithm for elimination of common subexpressions,” IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 18, pp. 58–68, Jan. 1999.

[24] M. Potkonjak, M. B. Srivastava, and A. P. Chandrakasan, “Multiple constant multiplications: Efficient and versatile framework and algorithms for exploring common subexpression elimination,” IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 15, pp. 151–165, Feb. 1996.

[25] T. Saram¨aki and M. Renfors, “A novel approach for the design of IIR filters as a tapped cascaded interconnection of identical allpass subfilters,” in Proc. IEEE Int. Symp. Circuits Syst., Philadelphia, PA, May 1987, pp. 629–632.

[26] P. P. Vaidyanathan and S. K. Mitra, “Low passband-sensitivity filters: Generalized viewpoint and synthesis procedures,” Proc. IEEE, vol. 72, no. 4, pp. 404–423, Apr. 1984.

[27] W. Wegener, “On the design of wave digital filters with short coefficient wordlength and optimal dynamic range,” IEEE Trans. Circuits Syst., vol. CAS-25, pp. 1091–1098, Dec. 1978.

[28] T. Wicks and S. Summerfield, “VLSI implementation of high speed digital filters based on a restricted coefficient set,” in Proc. IEEE Int. Symp. Circuits Syst., Chicago, 1993, pp. 603–606.

[29] N. I. Smith, “A random-search method for designing finite-wordlength recursive digital filters,” IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-27, pp. 40–46, Feb. 1979.

[30] M. Papamarkos and G. Vachtsevanos, “On the design of 1D IIR digital filters with coefficients of finite word length,” Int. J. Circuit Theory Applicat., vol. 19, pp. 217–227, 1991.

[31] K. A. Owenier, “Optimization of wave digital filters with reduced number of multipliers,” Arch. Elektron. ¨Ubertr.tech., vol. 30, pp. 387–393, Oct. 1976.

[32] A. R. Mirzai and S. S. Lawson, “Finite wordlength design of wave digital filters,” Electron. Lett., vol. 22, pp. 851–853, July 1986.

[33] Y. C. Lim and S. R. Parker, “Design of discrete-coefficient-value linear phase FIR filters with optimum normalized peak ripple magnitude,” IEEE Trans. Circuits Syst., vol. 37, pp. 1480–1486, Dec. 1990.

[34] S. S. Lawson and S. Summerfield, “The design of wave digital filters using fully pipelined systolic arrays,” J. VLSI Signal Processing, vol. 2, pp. 51–64, 1990.

[35] A. Jones, S. Lawson, and T. Wicks, “Design of cascaded allpass structures with magnitude and delay constraint using simulated annealing and quasiNewton methods,” in Proc. IEEE Int. Symp. Circuits Syst., Singapore, June 1991, pp. 2439–2442.

[36] L. Gefferth, “Discrete optimization of wave digital filters combining simulated annealing and line search methods,” in Proc. Euro. Conf. Circuit Theory Design, Brighton, UK, 1989, pp. 502–506.

[37] F. Catthoor, J. Vandewalle, and H. De Man, “Simulated-annealing-based optimization of coefficient and data word-lengths in digital filters,” Int. J. Circuit Theory Applicat., vol. 16, pp. 371–390, 1988.

[38] F. Brglez, “Digital filter design with short word-length coefficient,” IEEE Trans. Circuits Syst., vol. CAS-25, pp. 1044–1050, Dec. 1978.

[39] M. S. Anderson, S. Summerfield, and S. S. Lawson, “Realisation of lattice wave digital filters using 3-port adaptors,” Electron. Lett., vol. 31, pp. 628–629, Apr. 1995.

[40] A. Antoniou, C. Charalambous, and Z. Motamedi, “Two methods for the reduction of quantization effects in recursive digital filters,” IEEE Trans. Circuits Syst., vol. CAS-30, pp. 160–167, Mar. 1983.

[41] E. Avenhaus, “On the design of digital filters with coefficients of limited word length,” IEEE Trans. Acoust., Speech, Signal Processing, vol. AU-20, pp. 206–212, Aug. 1972.

[42] R. E. Crochiere, “A new statistical approach to the coefficient word length problem for digital filters,” IEEE Trans. Circuits Syst., vol. CAS-22, no. 3, pp. 190–196, Mar. 1975.

[43] A. G. Dempster and M. D. Macleod, “Variable statistical wordlength in digital filters,” IEE Proc.-Vis. Image Signal Process., vol. 143, pp. 62–66, Feb. 1996.

[44] H. J. Oh and Y. H. Lee, “Design of discrete coefficient FIR and IIR digital filters with prefilter-equalizer structure using linear programming,” IEEE Trans. Circuits Syst. II, vol. 47, pp. 562–565, June 2000.

[45] Y. C. Lim and S. R. Parker, “FIR filter design over a discrete powers-of-two coefficient space,” IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP 31, pp. 583–591, June 1983.

[46] O. Gustafsson, H. Johansson, and L. Wanhammar, “An MILP approach for the design of linear-phase FIR filters with minimum number of signed power-of-two terms,” in Proc. Euro. Conf. Circuit Theory Design, Espoo, Finland, Aug. 28–31 2001, vol. 2, pp. 217–220.

[47] Q. Zhao and Y. Tadokoro, “A simple design of FIR filters with powers-of-two coefficients,” IEEE Trans. Circuits Syst., vol. 35, pp. 566–570, May 1988.

[48] H. Samueli, “A low-complexity multiplierless half-band recursive digital filter design,” IEEE Trans. Acoust., Speech, Signal Processing, vol. 37, no. 3, pp. 442–444, Mar. 1989.

[49] H. Samueli, “An improved search algorithm for the design of multiplierless FIR filters with power-of-two coefficients,” IEEE Trans. Circuits Syst., vol. 36, pp. 1044–1047, July 1989.

[50] X. R. Jiang and S. N. G¨ullo¨uglu, “On discrete optimization of multiplier coefficients of digital filters,” in Proc. Euro. Conf. Circuit Theory Design, 1986, pp. 196–198.

[51] T. Wicks and S. Lawson, “Genetic algorithm design of wave digital filters with a restricted coefficient set,” in Proc. Inst. Elect. Eng. Natural Algorithms in Signal Process., Chelmsford, Essex, Nov. 1993.

[52] P. B. Wilson and M. D. Macleod, “Low implementation cost IIR digital filter design using genetic algorithms,” in Proc. IEE/IEEE Workshop on Natural Algorithms in Signal Processing, Danbury, Chelmsford, U. K., Nov. 1993, pp.

[53] H. Safiri, M. Ahmadi, G. A. Jullien, and W. C. Miller, “A novel approach based on genetic algorithm for pipelining of recursive filters,” in Proc. IEEE Int. Symp. Circuits Syst., Sydney, Australia, May 6–9 2001, vol. 2, pp. 633–636.

[54] D. W. Redmill, D. R. Bull, and E. Dagless, “Genetic synthesis of reduced complexity filters and filter banks using primitive operator directed graphs,” Proc. Inst. Elect. Eng. Circuits, Devices, Syst., vol. 147, pp. 303–310, Oct. 2000.

[55] A. Lee, M. Ahmadi, G. A. Jullien, R. S. Lashkan, and W.C. Miller, “Design of 1-D IR filters with genetic algorithms,” in Proc. IEEE Int. Symp. Circuits Syst., Orlando, FL, 1999, vol. 3, pp. 295 – 298.

[56] D. Karaboga, D. H. Horrocks, N. Karaboga, and A. Kalinli, “Designing digital FIR filters using Tabu search algorithm,” in Proc. IEEE Int. Symp. Circuits Syst., Hong Kong, June 9–12 1997, vol. 4, pp. 2236–2239.

[57] S. Traferro, F. Capparelli, and F. Piazza, “Efficient allocation of power of two terms in FIR digital filter design using tabu search,” in Proc. IEEE Int. Symp. Circuits Syst., Orlando, FL, 1999, vol. 3, pp. 411–414.

[58] H. Shaffeu, M. M. Jones, H. D. Griffiths, and J. T. Taylor, “Improved design procedure for multiplierless FIR digital filters,” Electron. Lett., vol. 27, pp. 1142–1144, June 1991.

[59] D. Li, J. Song, and Y. C. Lim, “A polynomial-time algorithm for designing digital filters with power-of-two coefficients,” in Proc. IEEE Int. Symp. Circuits Syst., May 1993, vol. 1, pp. 84–87.

[60] E. Avenhaus and Sch¨ußler, “On the approximation problem in the design of digital filters with limited wordlength,” Arch. Elek. ¨Ubertragung., vol. 12, pp. 571–572, Dec. 1970.

[61] P. P. Vaidyanathan, P. Regalia, and S. Mitra, “Design of doubly complementary IIR digital filters using a single complex allpass filter, with multirate applications,” IEEE Trans. Circuits Syst., vol. CAS-34, pp. 378–389, Apr. 1987.

[62] J. A. Nossek and H.-D. Schwartz, “Wave digital lattice filters with applications in communication systems,” in Proc. IEEE Int. Symp. Circuits Syst., Newport Beach, CA, May 1983, pp. 845–848.

[63] M. Lang, “Allpass filter design and applications,” in Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, 1995.

[64] M. Gerken, W. Sch¨ußler, and P. Steffen, “On the design of digital filters consisting of a parallel connection of allpass sections and delay elements,” Arch. Elektron. ¨Ubertr.tech., vol. 49, no. 1, pp. 1–11, 1995.

[65] A. Fettweis, “Wave digital filters: Theory and practice,” Proc. IEEE, vol. 74, pp. 270–327, Feb. 1986.

[66] W. Drews and L. Gazsi, “A new design method for polyphase filters using all-pass sections,” IEEE Trans. Circuits Syst., vol. CAS-33, pp. 346–348, Mar. 1986.

[67] T. Saram¨aki and T. Ritoniemi, “Optimization of digital filter structures for VLSI implementation,” Automatika, vol. 34, pp. 111–116, 1993.

[68] M. Renfors and E. Zigouris, “Signal processor implementation of digital allpass filters,” IEEE Trans. Acoust., Speech, Signal Processing, vol. 36, pp. 714–729, May 1988.

[69] L. D. Mili´c and M. D. Lutovac, “Design of multiplierless elliptic IIR filters with a small quantization error,” IEEE Trans. Signal Processing, vol. 47, pp. 469–479, Feb. 1999.

[70] Juha Yli-Kaakinen, T. S. a. (1999). "Design of Very Low-Sensitivity and Low-Noise Recursive Filters Using a Cascade of Low-Order Lattice Wave Digital Filters." IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS 46(7).

Abstract:

To take advantage of the properties and characteristics of algorithms, we need to implement it in practice. Among the different algorithms, the digital algorithms are of great importance, Because of the advantages of digital systems over analog. Digital hardware such as digital signal processors, FPGA and etc., have a wide range of applications.

In this thesis, one of the most important structures of digital filters, wave digital filters was considered. Using Software implementation of filters in MATLAB environment, a comparison between this class of filters and several other important classes of digital filters has been done, in terms of speed and sensitivity.

By using assembly language, a cascade of two-level LWD filter has been implemented for TMS320VC5505 processor. This implementation fulfilled in the Code Composer Studio environment. The assembly program has been optimized, By using parallel processing instructions. Then Run time of the optimized assembly program was compared to the none-optimized assembly program.

Hardware implementation of this filter has been fulfilled by using evaluation module TMS320VC5505. The output signal of the TMS320VC5505 was observed on the oscilloscope. This signal processor is product of Texas Instrument.

Implementation of Low-Sensitivity, Low-Noise LWD recursive digital filters on DSP processors

Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of Master of Science (M.Sc.)

in Electronic Engineering

Department of Electrical Engineering and Robotics

Shahrood University of Technology

By:

Samad Zabihi Azadboni

Supervisor:

Dr. Omidreza Marozi

Dr. Hadi Gerailu

Advisor:

Dr. Ali Soleimani Ivari

Summer 2012