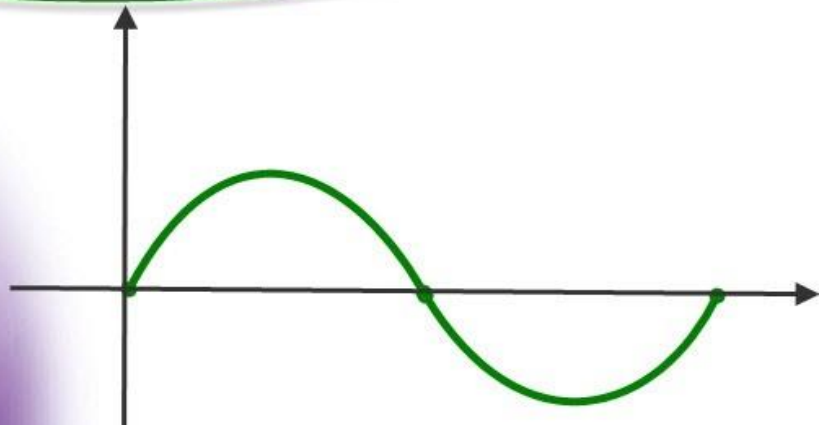


برای دریافت فایل Word پروژه به سایت **ویکی پاور** مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم



برای دریافت فایل Word پروژه به سایت **ویکی پاور** مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

## موضوع پروژه:

پیاده سازی بلادرنگ کدک صحبت استاندارد G.728

بر روی پردازنده TMS320C5402

WikiPower.ir

برای خرید فایل word این پروژه [اینجا کلیک کنید](#).

( شماره پروژه = ۳۹۴ )

پشتیبانی: ۰۹۳۵۵۴۰۵۹۸۶

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

### Abstract

G.728 speech codec is a low delay ITU standard codec which could provide toll quality speech in 16kb/s. It is specially designed for delay sensitive applications like satellite telephony, Internet, and mobile networks.

In this thesis real time implementation of full duplex G.728 encoder and decoder on TMS320C5402 is presented. Using a combinatorial technique for TMS programming, the programming time and complexity have been decreased to 30% comparing with traditional assembly programming. First a fixed point simulation of the codec algorithm has been programmed in C and it is compiled to assembly using CCS (Code Composer Studio) and manually optimized. Then some of the critical functions regarding MIPS, have been programmed in assembly for achieving real time implementation. Finally, implementation results have been presented.

**Keywords:** Speech Coding & Compression, Real Time Implementation, DSP,  
TMS320C5402, DSK Board

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

### چکیده

کدک صحبت استاندارد G.728 ، یک کدک کم تاخیر است که صحبت با کیفیت عالی را در نرخ بیت 16 kbps ارائه می دهد و برای شبکه های تلفن ماهواره ای و اینترنت و موبایل که به تاخیر زیاد حساس هستند ، مناسب است. در این رساله به پیاده سازی بلادرنگ اینکدر و دیکدر G.728 بصورت دوطرفه کامل ( Full Duplex ) بر روی پردازنده TMS320C5402 می پردازیم .

روشی ترکیبی برای برنامه نویسی TMS ارائه می شود که در آن زمان و پیچیدگی برنامه نویسی نسبت به برنامه نویسی دستی به ۳۰٪ کاهش می یابد . در این روش پس از برنامه نویسی و شبیه سازی ممیز ثابت الگوریتم کدک به زبان C ، با استفاده از نرم افزار ( Code Composer Studio ) CCS ، برنامه به زبان اسمبلی ترجمه شده و بهینه سازی دستی در کل کد اسمبلی صورت می گیرد . سپس بعضی از توابع مهم برنامه از نظر MIPS ، بصورت دستی به زبان اسمبلی بازنویسی می شوند تا برنامه بصورت بلادرنگ قابل اجرا گردد . در پایان نتایج این پیاده سازی ارائه می شود .

### کلمات کلیدی

کدینگ و فشرده سازی صحبت ، پیاده سازی بلادرنگ ، DSP ، TMS320C5402 ، برد DSK

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

## فهرست

۴	- مقدمه
	فصل ۱: بررسی و مدل سازی سیگنال صحبت
۶	۱-۱- معرفی سیگنال صحبت
۱۰	۲-۱- مدل سازی پیشگویی خطی
۱۱	۱-۲-۱- پنجره کردن سیگنال صحبت
۱۳	۲-۲-۱- پیش تاکید سیگنال صحبت
۱۴	۳-۲-۱- تخمین پارامترهای LPC
	فصل ۲: روش ها و استانداردهای کدینگ صحبت
۱۵	۱-۲- مقدمه
۱۹	۲-۲- روش های کدینگ
۲۱	۱-۲-۲- کدرهای شکل موج
۲۲	۲-۲-۲- کدرهای صوتی
۲۴	۳-۲-۲- کدرهای مختلط
۲۷	الف- کدرهای مختلط حوزه فرکانس
۲۹	ب- کدرهای مختلط حوزه زمان
	فصل ۳: کدر کم تاخیر LD-CELP
۳۴	۱-۳- مقدمه
۳۶	۲-۳- بررسی کدر کم تاخیر LD-CELP
۳۹	۱-۲-۳- LPC معکوس مرتبه بالا
۴۲	۲-۲-۳- فیلتر وزنی شنیداری
۴۲	۳-۲-۳- ساختار کتاب کد

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

۴۳	۱-۳-۲-۳-جستجوی کتاب کد
۴۵	۴-۲-۳-شبه دیکدر
۴۶	۵-۲-۳-پست فیلتر
	<b>فصل ۴ : شبیه سازی ممیز ثابت الگوریتم به زبان C</b>
۴۹	۱-۴-مقدمه
۵۰	۲-۴-ویژگی های برنامه نویسی ممیز ثابت
۵۳	۳-۴-ساده سازی محاسبات الگوریتم
۵۴	۱-۳-۴-تطبیق دهنده بهره
۵۸	۲-۳-۴-محاسبه لگاریتم معکوس
۵۹	۴-۴-روندنمای برنامه
۶۳	۱-۴-۴-اینکدر
۶۹	۲-۴-۴-دیکدر
	<b>فصل ۵ : پیاده سازی الگوریتم بر روی DSP</b>
۷۴	۱-۵-مقدمه
۷۵	۲-۵-مروری بر پیاده سازی بلادرنگ
۷۶	۳-۵-چیپ های DSP
۷۷	۱-۳-۵-DSP های ممیز ثابت
۷۸	۲-۳-۵-مروری بر DSP های خانواده TMS320
۷۹	۱-۲-۳-۵-معرفی سری TMS320C54x
۸۱	۴-۵-توسعه برنامه بلادرنگ
۸۲	۵-۵-اجرای برنامه روی برد توسعه گر C5402 DSK
۸۴	۱-۵-۵-بکارگیری ابزارهای توسعه نرم افزار
۸۶	۲-۵-۵-استفاده از نرم افزار CCS
۹۴	۳-۵-۵-نتایج پیاده سازی
۹۷	۶-۵-نتیجه گیری و پیشنهاد

- ضمائم

- ضمیمه (الف) : دیسکت برنامه های شبیه سازی ممیز ثابت به زبان C و

برای دریافت فایل Word پروژه به سایت **ویکی پاور** مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

پیاده سازی کدک به زبان اسمبلی

۹۸

- ضمیمه (ب) : مقایسه برنامه نویسی C و اسمبلی

۱۰۳

- مراجع



## برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

### - مقدمه

امروزه در عصر ارتباطات و گسترش روزافزون استفاده از شبکه های تلفن، موبایل و اینترنت در جهان و محدودیت پهنای باند در شبکه های مخابراتی، کدینگ و فشرده سازی صحبت امری اجتناب ناپذیر است. در چند دهه اخیر روشهای کدینگ مختلفی پدیدآمده اند ولی بهترین و پرکاربردترین آنها کدک های آنالیزباسنتز هستند که توسط Atal & Remede در سال ۱۹۸۲ معرفی شدند [2]. اخیرا مناسبترین الگوریتم برای کدینگ صحبت با کیفیت خوب در نرخ بیت های پائین و زیر 16 kbps، روش پیشگویی خطی باتحریک کد (CELP) می باشد که در سال ۱۹۸۵ توسط Schroeder & Atal معرفی شد [8] و تا کنون چندین استاندارد مهم کدینگ صحبت بر اساس CELP تعریف شده اند.

در سال ۱۹۸۸ برنامه ای برای استانداردسازی یک کدک 16 kbps با تاخیراندک و کیفیت بالا در برابر خطاهای کانال آغاز نمود و برای آن کاربردهای زیادی همچون شبکه PSTN، ISDN، تلفن تصویری و غیره در نظر گرفت. این کدک در سال ۱۹۹۲ توسط Chen et al. تحت عنوان LD-CELP معرفی شد [6] و بصورت استاندارد G.728 در آمد [9] و در سال ۱۹۹۴ مشخصات ممیز ثابت این کدک توسط ITU ارائه شد [10]. با توجه به کیفیت بالای این کدک که در آن صحبت سنتز شده از صحبت اولیه تقریبا غیرقابل تشخیص است و کاربردهای آن در شبکه های تلفن و اینترنت و ماهواره ای در این گزارش به پیاده سازی این کدک می پردازیم.

در فصل اول به معرفی و آنالیز سیگنال صحبت پرداخته می شود و در فصل دوم روش ها و استانداردهای کدینگ بیان می شوند. در فصل سوم کدک LD-CELP را بیشتر بررسی می کنیم و در فصل چهارم شبیه سازی ممیز ثابت الگوریتم به زبان C را بیان می نمائیم. و در پایان در فصل ۵ به نحوه پیاده سازی بلادرنگ کدک G.728 بر روی پردازنده TMS320C5402 می پردازیم.



برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

## فصل ۱

### بررسی و مدل سازی سیگنال صحبت

#### ۱-۱- معرفی سیگنال صحبت

صحبت در اثر دمیدن هوا از ریه ها به سمت حنجره و فضای دهان تولید می شود. در طول این مسیر در انتهای حنجره، تارهای صوتی<sup>۱</sup> قرار دارند. فضای دهان را از بعد از تارهای صوتی، لوله صوتی<sup>۲</sup> می نامند که در یک مرد متوسط حدود ۱۷cm طول دارد. در تولید برخی اصوات تارهای صوتی کاملاً باز هستند و مانعی بر سر راه عبور هوا ایجاد نمی کنند که این اصوات را اصطلاحاً اصوات بی واک<sup>۳</sup> می نامند. در دسته دیگر اصوات، تارهای صوتی مانع خروج طبیعی هوا از حنجره می گردند که این باعث به ارتعاش در آمدن تارها شده و هوا به طور غیر یکنواخت و تقریباً پالس شکل وارد فضای دهان می شود. این دسته از اصوات را اصطلاحاً باواک<sup>۴</sup> می گویند.

فرکانس ارتعاش تارهای صوتی در اصوات باواک را فرکانس Pitch و دوره تناوب ارتعاش

تارهای صوتی را پریود Pitch می نامند. هنگام انتشار امواج هوا در لوله صوتی، طیف فرکانس این امواج

<sup>۱</sup> Vocal Cords

<sup>۲</sup> Vocal Tracts

<sup>۳</sup> Unvoiced

<sup>۴</sup> Voiced

## برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

توسط لوله صوتی شکل می گیرد و بسته به شکل لوله، پدیده تشدید در فرکانس های خاصی رخ می دهد که به این فرکانس های تشدید فرمنت<sup>۱</sup> می گویند.

از آنجا که شکل لوله صوتی برای تولید اصوات مختلف، متفاوت است پس فرمنت ها برای اصوات گوناگون با هم فرق می کنند. با توجه به اینکه صحبت یک فرآیند متغیر با زمان است پس پارامترهای تعریف شده فوق اعم از فرمنت ها و پریود Pitch در طول زمان تغییر می کنند به علاوه مد صحبت به طور نامنظمی از باواک به بی واک و بالعکس تغییر می کند. لوله صوتی، همبستگی های زمان-کوتاه، در حدود 1 ms، درون سیگنال صحبت را در بر می گیرد. و بخش مهمی از کار کدکننده های صوتی مدل کردن لوله صوتی به صورت یک فیلتر زمان-کوتاه می باشد. همان طور که شکل لوله صوتی نسبتاً آهسته تغییر می کند، تابع انتقال این فیلتر مدل کننده هم نیاز به تجدید<sup>۲</sup>، معمولاً در هر 20ms یکبار خواهد داشت.

در شکل (۱-۱ الف) یک قطعه صحبت باواک که با فرکانس 8KHz نمونه برداری شده است دیده می شود. اصوات باواک دارای تناوب زمان بلند به خاطر پریود Pitch هستند که نوعاً بین 2ms تا 20ms می باشد. در اینجا پریود Pitch در حدود 8ms یا ۶۴ نمونه است. چگالی طیف توان این قطعه از صحبت در شکل (۱-۱ ب) دیده می شود [3].

اصوات بی واک نتیجه تحریک نویز مانند لوله صوتی هستند و تناوب زمان-بلند اندکی را در بر دارند، همانگونه که در شکل های (۱-۱ ج) و (۱-۱ د) دیده می شود ولی همبستگی زمان کوتاه به خاطر لوله صوتی در آنها هنوز وجود دارد.

<sup>۱</sup>Formant

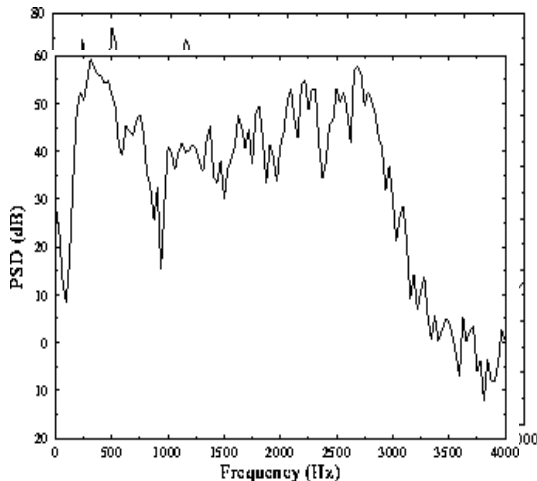
<sup>۲</sup> Update

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

بطور کلی سیگنال صحبت دارای افزونگی<sup>۱</sup> زیادی است که ناشی از عوامل ذیل هستند:

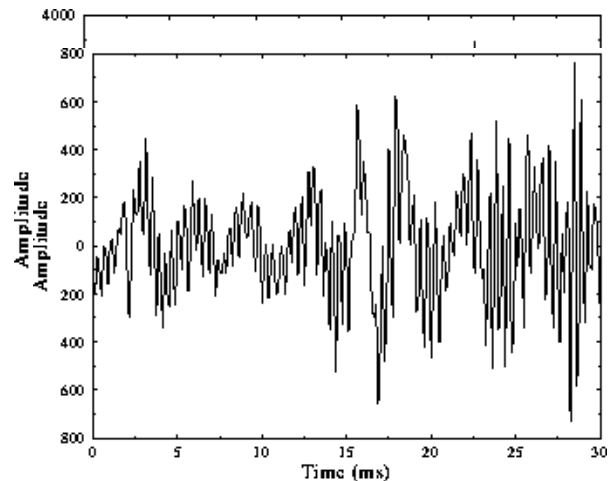
- وابستگی های زمان-کوتاه : این وابستگی ها عمدتاً به کندی تغییرات صحبت با زمان و ساختار

(ب)



(د)

(الف)



(ج)

شکل (۱-۱) : مقایسه اصوات باواک و بی واک. (الف) و (ب) : باواک ، (ج) و (د) : بی واک

نسبتاً منظم فرمت ها مربوط می شوند.

— وابستگی های زمان-بلند : که عمدتاً از طبیعت نیمه متناوب اصوات با واک و تغییرات آرام پریرود Pitch ناشی می شوند.

— تابع چگالی احتمال صحبت : علیرغم پیچیدگی آماری صحبت می توان آن را با توابع چگالی احتمال شناخته شده تقریب زد. شکل لوله صوتی و مد تحریک آن به صورت نسبتاً آرام تغییر می کند و بنابراین صحبت را می توان به صورت شبه ایستان در دوره های کوتاه زمانی (حدود 20ms) در نظر

<sup>۱</sup> Redundancy

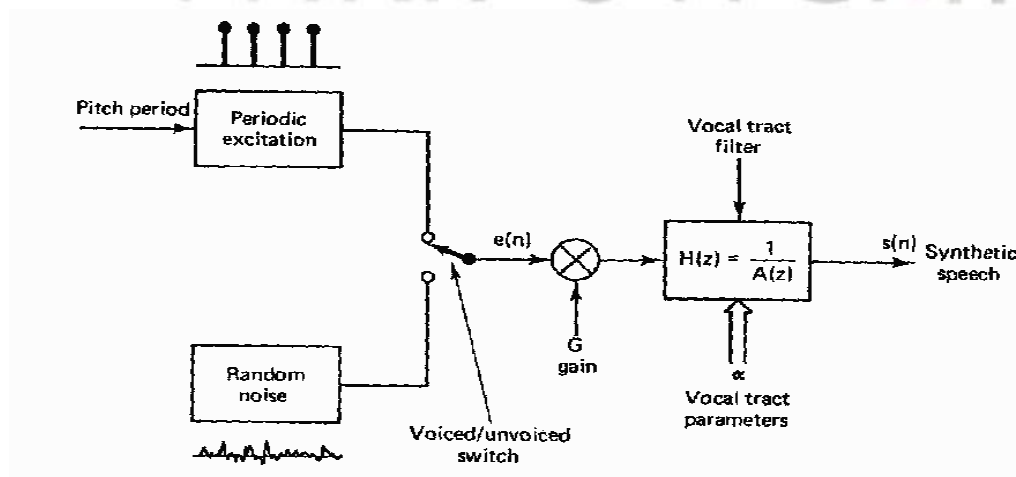
برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

گرفت و با یک فرآیند تصادفی ارگادیک در یک قطعه زمانی کوچک مدل نمود و طیف مشخصی برای آن در این قطعه زمانی بدست آورد.

علاوه بر افزونگی های فوق عامل مهم دیگری که کاهش نرخ داده سیگنال صحبت را ممکن می سازد، طبیعت غیر حساس گوش انسان نسبت به بسیاری از ویژگیهای این سیگنال می باشد.

### ۲-۱- مدل سازی پیشگویی خطی

روش کدینگ پیشگویی خطی (LPC<sup>۱</sup>) مبتنی بر مدل تولید صحبت در کد کننده های صوتی می باشد که در اینجا در شکل (۲-۱) نشان داده شده است. برای استفاده از مدل لازم است که معلوم شود سیگنال با واک است یا بی واک و اگر با واک است پریود Pitch مجاسبه گردد. تفاوت اصلی بین LPC و سایر کدکننده های صوتی در مدل کردن لوله صوتی است. در تحلیل LPC، لوله صوتی به صورت یک فیلتر دیجیتال تمام قطب در نظر گرفته می شود. [4,1].



<sup>۱</sup> Linear Predictive Coding

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

شکل (۲-۱): مدل تولید صحبت در LPC

با شرکت دادن بهره G در این فیلتر داریم:

$$H(Z) = \frac{G}{1 + a_1 Z^{-1} + \dots + a_p Z^{-p}} = \frac{S(Z)}{E(Z)}$$

که در آن p مرتبه فیلتر است. اگر S(n) خروجی فیلتر مدل صحبت و e(n) تحریک ورودی باشد، معادله فوق را در حوزه زمان به صورت زیر می توان نوشت:

$$S(n) = Ge(n) - a_1 S(n-1) - \dots - a_p S(n-p)$$

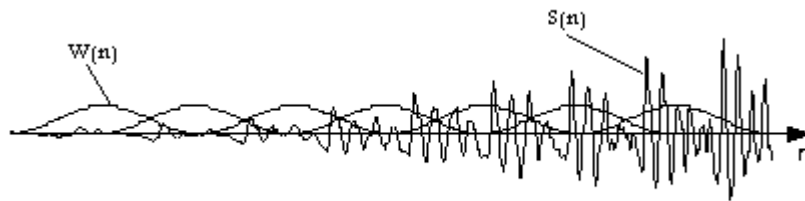
به عبارت دیگر هر نمونه صحبت به صورت ترکیب خطی از نمونه های قبلی قابل بیان است و این دلیل نام گذاری کدینگ پیشگویی خطی (LPC) می باشد.

### ۱-۲-۱- پنجره کردن سیگنال صحبت

روش LPC هنگامی دقیق است که به سیگنالهای ایستان<sup>۱</sup> اعمال شود، یعنی به سیگنالهایی که رفتار آنها در زمان تغییر نمی کند. هر چند که این موضوع در مورد صحبت صادق نیست، اما برای اینکه بتوانیم روش LPC را بکار ببریم، سیگنال صحبت را به قسمت های کوچکی بنام "فریم" تقسیم می کنیم که این فریم ها شبه ایستان هستند. شکل (۳-۱) مثالی از قسمت بندی سیگنال صحبت را نشان می دهد. این قسمت بندی با ضرب کردن سیگنال صحبت S(n)، در سیگنال پنجره W(n) انجام می شود.

<sup>۱</sup> Stationary

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه



شکل (۳-۱) : قسمت بندی سیگنال صحبت

معروف ترین انتخاب برای پنجره ، پنجره همینگ (Hamming) به صورت زیر است:

$$W(n) = 0.54 - 0.46 \cos \frac{2\pi n}{N} \quad , \quad 0 \leq n \leq N-1$$

$$W(n) = 0 \quad , \quad otherwise$$

در اینجا N ، طول پنجره دلخواه به نمونه و عموماً در محدوده 160-320 انتخاب می گردد که

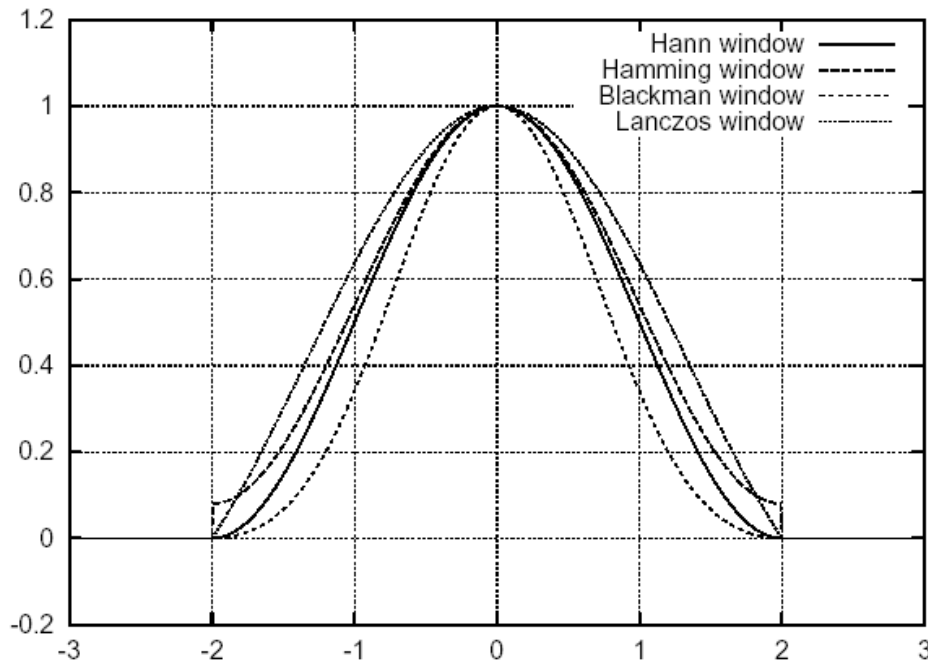
240 یک مقدار نوعی می باشد . در شکل (۱-۴) چند پنجره معروف نشان داده شده است.

معمولاً پنجره های متوالی بر روی هم همپوشانی دارند و فاصله بین آنها را پریود فریم می گویند.

مقادیر نوعی برای پریود فریم 10-30ms می باشد. این انتخاب به نرخ بیت و کیفیت صحبت دلخواه ما

بستگی خواهد داشت. هر چه پریود فریم کوچکتر باشد، کیفیت بهتری خواهیم داشت.

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه



شکل (۴-۱): نمایش چند پنجره معروف

### ۱-۲-۲- پیش تاکید سیگنال صحبت

شکل (۵-۱) یک توزیع طیفی نمونه سیگنال صحبت را برای اصوات باواک نشان می دهد. با

توجه به افت طیف در فرکانس های بالا و ضعیف بودن فرکانس های بالا در طیف صحبت، تحلیل LPC

در فرکانس های بالا عملکرد ضعیفی خواهد داشت. برای تقویت مؤلفه های فرکانس بالا صحبت، آن را

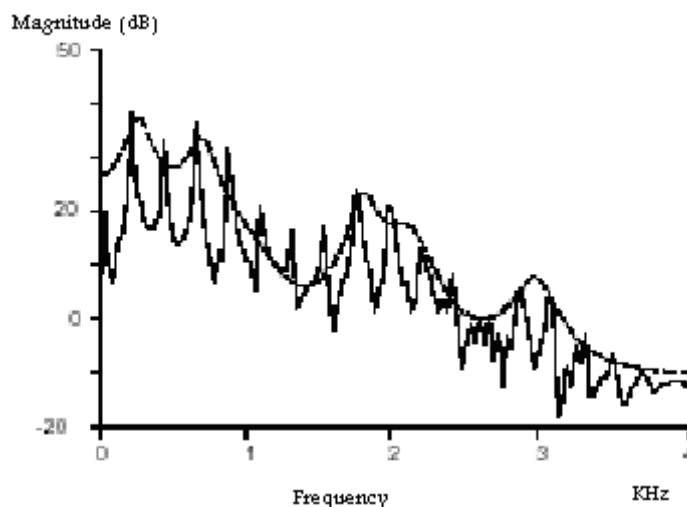
از یک فیلتر بالا گذر با تابع انتقال  $1 - az^{-1}$  که فیلتر پیش تاکید نامیده می شود، عبور می دهیم. مقدار

نوعی ضریب  $a$  معمولاً  $\frac{15}{16} = 0.9375$  در نظر گرفته می شود.

اگر  $S(n)$  سیگنال ورودی باشد، سیگنال پیش تاکید شده  $S'(n)$  خواهد شد:

$$S'(n) = S(n) - 0.9375S(n-1)$$

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آر م سایت و به همراه فونت های لازم



شکل (۵-۱): پوشش طیفی نمونه اصوات باواک

### ۱-۲-۳- تخمین پارامترهای LPC

در اینجا لازم است که پارامترهای مدل LPC یعنی ضرایب  $a_i$  فیلتر و بهره  $G$  تعیین گردند. اگر

$$\hat{S}(n) = -a_1 S(n-1) - \dots - a_p S(n-p)$$

تخمین  $S(n)$  از روی نمونه های قبلی باشد، ضرایب  $a_i$  را چنان تعیین می کنیم که خطای

$$\sum_n [S(n) - \hat{S}(n)]^2$$

روی همه نمونه های موجود مینیمم گردد. این مینیمم سازی ما را به معادلات خطی زیر می رساند:

$$a_1 r(0) + a_2 r(1) + \dots + a_p r(p-1) = -r(1)$$

$$a_1 r(1) + a_2 r(2) + \dots + a_p r(p-2) = -r(2)$$

⋮

$$a_1 r(p-1) + a_2 r(p-2) + \dots + a_p r(0) = -r(p)$$

و یا در فرم ماتریسی



برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

$$R \cdot a = -r$$

در معادلات فوق تعریف زیر را داریم:

$$r(i) = r(-i) = \sum_{n=0}^{N-i-1} S(n)S(n+i)$$

که  $r(i)$ ،  $i$  امین اتوکورلیشن سیگنال می باشد و فرض شده که  $S(n)$  به طول  $N$  پنجره شده است. این فرمولاسیون به روش اتوکورلیشن معروف است و ماتریس  $R$  در آن یک ماتریس Toeplitz می باشد. چنین ماتریسی غیرمنفرد و همیشه معکوس پذیر است و در نتیجه همواره می توانیم جوابی به صورت  $a = -R^{-1}r$  داشته باشیم.

روش دیگری نیز بنام روش کواریانس وجود دارد. در این روش سیگنال صحبت  $S(n)$  پنجره نمی شود و به جای اتوکورلیشن های  $r(i)$ ، کواریانس های  $r(i,j)$  برای عنصر  $(i,j)$  ماتریس  $R$  محاسبه می گردد:

$$r(i,j) = \sum_n S(n+i)S(n+j)$$

در اینجا تضمین نمی شود که ماتریس  $R$  معکوس پذیر باشد و ممکن است که سیستم معادلات فوق جواب نداشته باشد. در این حالت فیلتر LPC ناپایدار می شود. از این رو در اینجا بیش از این به روش کواریانس نمی پردازیم.

راه سوم روش Burg است که امتیاز عدم استفاده از پنجره را در روش کواریانس با امتیاز روش اتوکورلیشن یعنی تضمین پایداری فیلتر، ترکیب می کند. این روش از ساختار مشبک<sup>۱</sup> فیلتر تمام قطب استفاده می کند [1].

<sup>۱</sup> Lattice

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

جواب د ستگاه معادلات فوق را می توان با یکی از روش های کلا سیک آنالیز عددی مثل حذف

گوسی بدست آورد. اما چون R یک ماتریس Toeplitz است می توان از روشی مؤثر بنام روش تکرار

Durbin سود جست که بصورت زیر ضرایب فیلتر را تولید می کند :

$$E(0) = r(0)$$

$$\text{for } i = 1, \dots, p$$

$$K_i = -\frac{r(i) + a_1^{(i-1)}r(i-1) + \dots + a_i^{(i-1)}r(1)}{E(i-1)},$$

$$a_i^{(i)} = K_i$$

$$\text{for } j = 1, \dots, i-1 \quad a_j^{(i)} = a_j^{(i-1)} + K_i a_{i-j}^{(i-1)}$$

$$E(i) = (1 - K_i^2)E(i-1)$$

که در آن  $a_j^{(i)}, j = 1, \dots, i$ ، ضریب  $j$  ام فیلتر در تکرار  $i$  ام و  $E(i)$  خطای پیشگویی مرتبه  $i$  است و

بدین ترتیب ضرایب فیلتر بصورت زیر بدست خواهند آمد:

$$a_j = a_j^{(p)}, j = 1, \dots, p$$

روش تکرار Durbin پارامترهای  $K_i, i = 1, \dots, p$  را که ضرایب انعکاس نامیده می شوند و  $E(p)$  را

بدست می دهد که مربع بهره پیشگویی  $G$  و مورد نیاز فیلتر سنتز می باشد:

$$G^2 = E(p)$$

و چون داریم :

$$E(p) = (1 - K_1^2)(1 - K_2^2) \dots (1 - K_p^2)r(0)$$

می توانیم به جای  $E(p)$ ،  $r(0)$  را کد کرده و ارسال داریم و از آنجا به بهره  $G$  برسیم و این ترجیح داده

می شود زیرا حساسیت  $r(0)$  به نویز کوانتیزاسیون کمتر از  $G$  است.

ضرایب انعکاس  $K_i$  یا PARCOR (برای PARTIAL CORrelation) نقش مهمی در تحلیل LPC دارند و

دارای خواص زیر هستند:

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

۱- ضرایب انعکاس  $K_i$  معادل با ضرایب فیلتر  $a_i$  هستند. به عبارت دیگر می توان  $K$  را به  $a$  و

برعکس تبدیل کرد :

$K$  به  $a$  :

$$\begin{aligned} a_i^{(i)} &= K_i \\ a_j^{(i)} &= a_j^{(i-1)} + K_i a_{i-1}^{(i-1)} \quad i = 1, \dots, p \\ & \quad j = 1, \dots, i-1 \end{aligned}$$

$a$  به  $K$  :

$$\begin{aligned} K_i &= a_i^{(i)} \\ a_j^{(i-1)} &= \frac{a_j^{(i)} - a_i^{(i)} a_{i-j}^{(i)}}{1 - K_i^2} \quad i = p, \dots, 1 \\ & \quad j = 1, \dots, i-1 \end{aligned}$$

۲- برای یک فیلتر پایدار یعنی یک فیلتر LPC که همه قطب های آن داخل دایره واحد باشد داریم:

$$-1 < K_i < 1 \quad , \quad i = 1, \dots, p$$

که این شرط بسیار مهمی است چرا که با اطمینان از اینکه  $K_i$  بین  $-1$  و  $+1$  است حتی بعد از

کوانتیزاسیون ، پایداری فیلتر تضمین خواهد شد. به علاوه محدوده  $(-1, +1)$  کار کوانتیزاسیون را ساده تر

می کند. ولی  $a_i$  ها دارای چنین ویژگی نیستند که پایداری فیلتر را تضمین نمایند و کوانتیزاسیون  $a_i$  ها

می تواند موجب ناپایداری شود.

برای دریافت فایل Word پروژه به سایت **ویکی پاور** مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

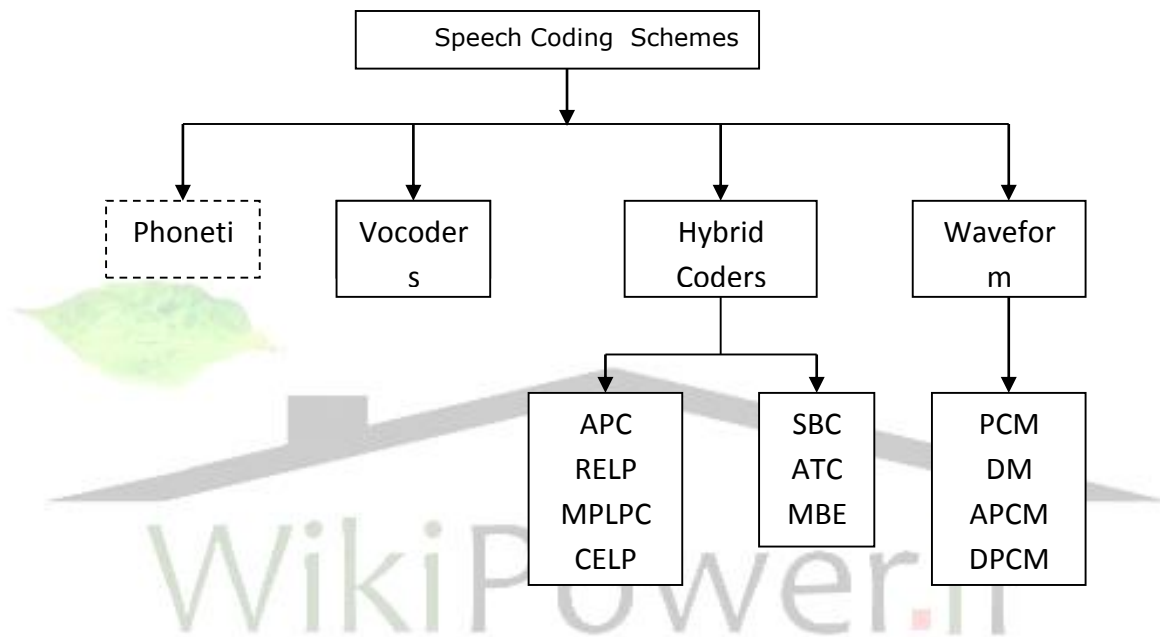


## فصل ۲



برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

روشهای کدینگ صحبت را می توان به چند دسته اصلی که در شکل (۱-۲) نشان داده شده است تقسیم بندی نمود. از این میان سه دسته اصلی که با خط پر نشان داده شده اند موضوع تحقیقاتی بیشتری هستند. این روشها سیگنال صحبت را آنالیز کرده افزونگی های آنها حذف نموده و بخش های غیر زائد صحبت را به روشی کد می کنند که از نظر شنیداری قابل قبول باشد.



شکل (۱-۲): دسته بندی روشهای کدینگ

کدرهای شکل موج<sup>۱</sup> نوعاً نرخ بیت بالایی دارند و صحبت بازسازی شده را با کیفیت خیلی خوب ارائه می دهند. کدرهای صوتی<sup>۲</sup> در نرخ بیت های خیلی پایین کار می کنند و صحبت را از طریق سنتز

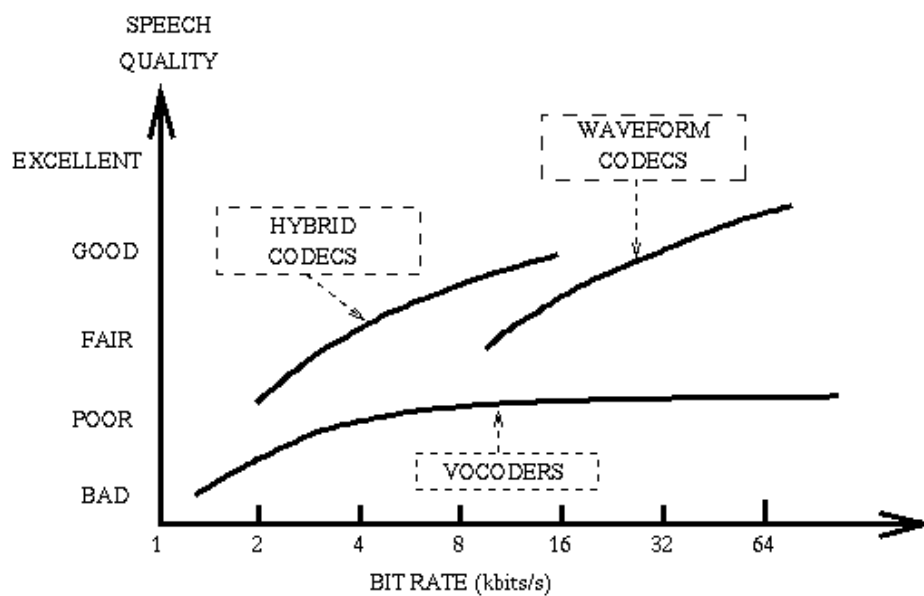
<sup>۱</sup> Waveform Coders

<sup>۲</sup> Vocoders

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

بازسازی می کنند. کدرهای مختلط ترکیبی از تکنیک های صوتی و شکل موج را بکار می گیرند و صحبت با کیفیت خوب را در نرخ بیت های میانی ارائه می دهند.

در شکل (۲-۲) کیفیت صحبت بر حسب نرخ بیت برای سه دسته اصلی کدینگ یعنی کدینگ شکل موج، کدینگ صوتی و کدینگ مختلط نشان داده شده است. همچنین خلاصه ای از کاربردهای روشهای در حال کار و آنهایی که در حال توسعه می باشند در جدول (۱-۲) گردآوری شده است [5].



شکل (۲-۲): مقایسه کیفیت صحبت روشهای کدینگ صحبت

Rate (kbps)	Application	Type of Coder	Year of Operation
64	PSTN (1 <sup>st</sup> Generation)	PCM	1972
32	PSTN (2 <sup>nd</sup> Generation)	ADPCM	1984

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

16	PSTN (3 <sup>rd</sup> Generation)	LD-CELP	1992
16	INMARSAT-B	APC	1985
13	GSM	RPE-LTP	1991
9.6	Skyphone	MPLPC	1990
8	North American Mobile	VSELP	1992
6.4	INMARSAT-M (land mobile)	MBE-CELP	1993
4.8	U.S. Gov.Fed.Standard	CELP	1991
4.8	NASA MSAT-X (mobile satellite)	Vector Adaptive Predictive Coding VAPC	1991

جدول (۱-۲): استاندارد های کدینگ صحبت

## ۱-۲-۲- کدرهای شکل موج

کدرهای شکل موج تلاش می کنند که شکل کلی سیگنال صحبت را حفظ نمایند. این کدرها می توانند هر شکل موجی در باند صوتی را قبول کنند و صرفاً مختص صحبت نیستند. از آنجا که

در این کدرها کدینگ به صورت نمونه به نمونه انجام می شود، عملکرد آنها همانند کوانتیزاسیون بوسیله نسبت سیگنال به نویز (SNR) اندازه گیری می شود. کدینگ های شکل موج برای صحبت، به نرخ بیت های بالای 16kbps محدود می شود و بخاطر سادگی و پیاده سازی آسان، فراگیر شده اند.



## برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

اولین استاندارد جهانی کدینگ صحبت استاندارد 64 kbps PCM G.711 ، با کمپندینگ u-law برای امریکای شمالی و A-law برای اروپا، یک کدر شکل موج بوده و هنوز هم کاربرد زیادی در سیستم های مخابراتی دیجیتال دارد . از آنجا که کیفیت صحبت 64 kbps PCM عالی می باشد ، معمولا مرجعی برای مقایسه دیگر کدرهای صحبت با نرخ بیت پایین تر قرار می گیرد . کدر بعدی که توسط CCITT استاندارد شده PCM تفاضلی تطبیقی (ADPCM) 32 kbps می باشد . کاهش نرخ بیت به نصف در این کدر، از طریق اعمال پیشگویی و کوانتیزاسیون تطبیقی حاصل شده است.

### ۲-۲-۲- کدرهای صوتی (Vocoders)

بر خلاف کدرهای شکل موج ، کدرهای صوتی خیلی به صحبت وابسته هستند و در اصولشان هم سعی ندارند که شکل موج اصلی صحبت را حفظ نمایند . یک کدر صوتی از یک آنالیزکننده و یک سنتزکننده تشکیل شده است . آنالیزکننده از صحبت اصلی یک دسته پارامترها که نمایش دهنده مدل تولید صحبت هستند ، استخراج نموده و آنها را ارسال می کند و در گیرنده ، صحبت با استفاده از این پارامترها بازسازی می شود.

در شکل (۲-۳) مدل تولید صحبت در کدرهای صوتی نشان داده شده است . لوله صوتی بصورت یک فیلتر متغیر با زمان نمایش داده می شود. برای قطعات بی واک صحبت، این فیلتر بایک منبع نویز سفید تحریک می گردد و برای قطعات باواک صحبت بوسیله یک قطار پالس باپریود pitch تحریک می شود.

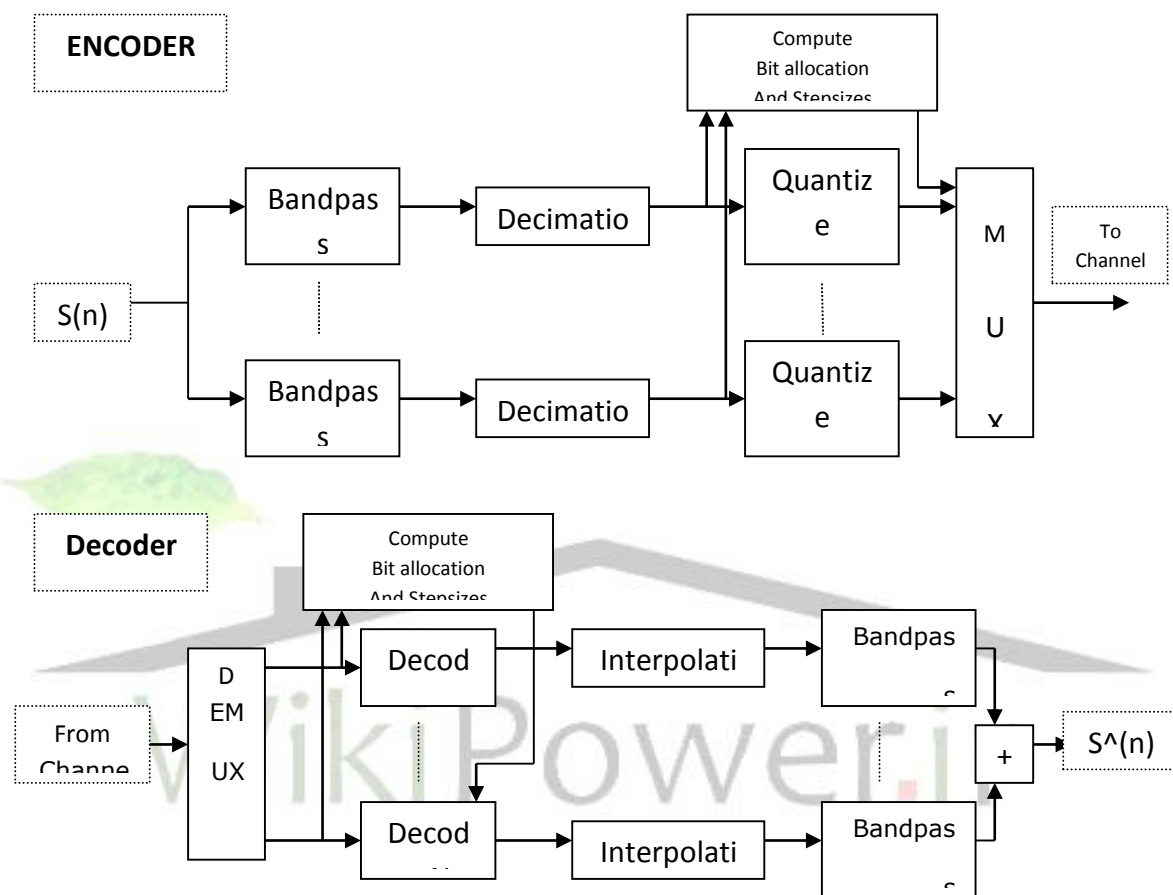
<sup>۱</sup> Adaptive Prediction PCM





برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آر م سایت و به همراه فونت های لازمه

سپس با هم جمع می شوند تا تقریبی از صحبت اصلی را بوجود آورند. در شکل (۲-۴) بلوک دیاگرام اینکدر و دیکدر زیر بانندی نشان داده شده است.

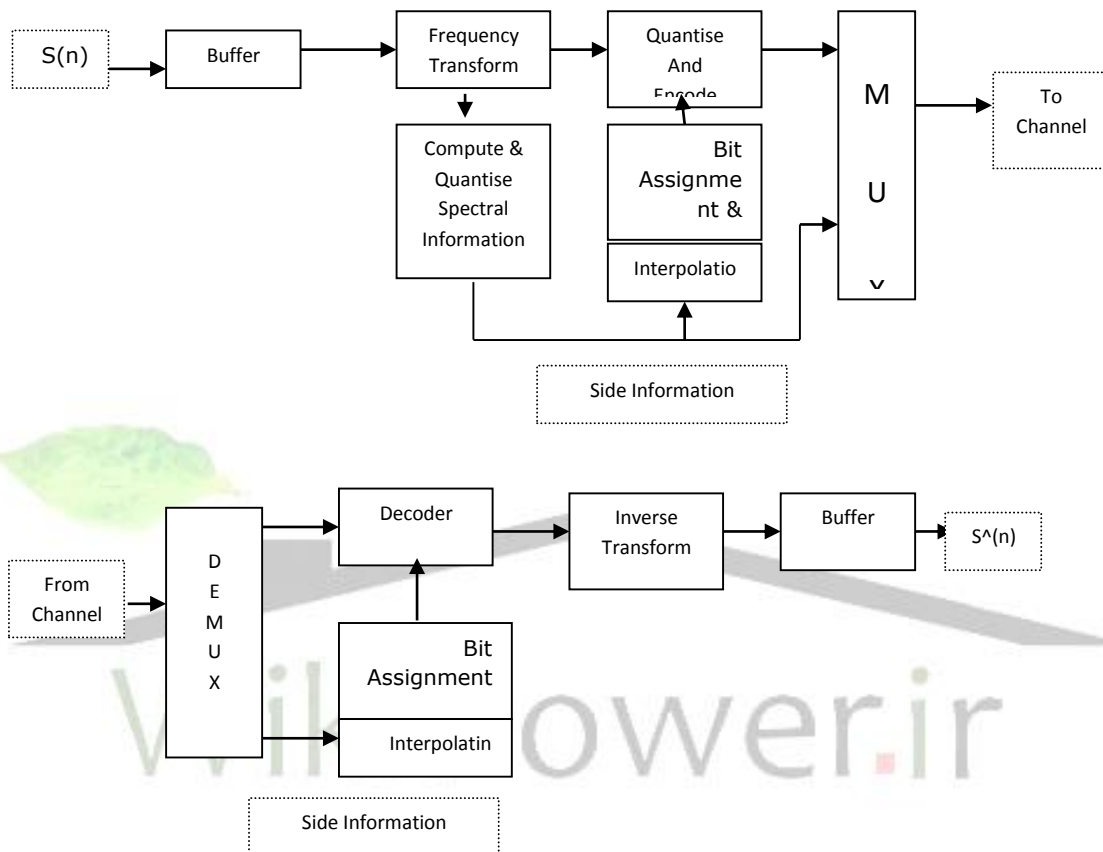


شکل (۲-۴) بلوک دیاگرام یک کدر زیر بانندی نمونه

کدرهای تبدیل تطبیقی (ATC) یک روش پیچیده تر آنالیز حوزه فرکانس هستند که شامل بلوکی از تبدیل های قطعات پنجره شده صحبت ورودی می باشند. هر قطعه با یک دسته از ضرائب تبدیل نمایش داده می شود که جداگانه کوانتیزه و ارسال می گردند. درگیرنده این ضرائب کوانتیزه و تبدیل

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

معکوس شده تا یک کپی از قطعه اصلی را باز سازی نماید. سپس قطعات مجاور به هم متصل شده تا صحبت سنتز شده را شکل دهند. در شکل (۵-۲) بلوک دیاگرام یک کدر ATC نشان داده شده است.



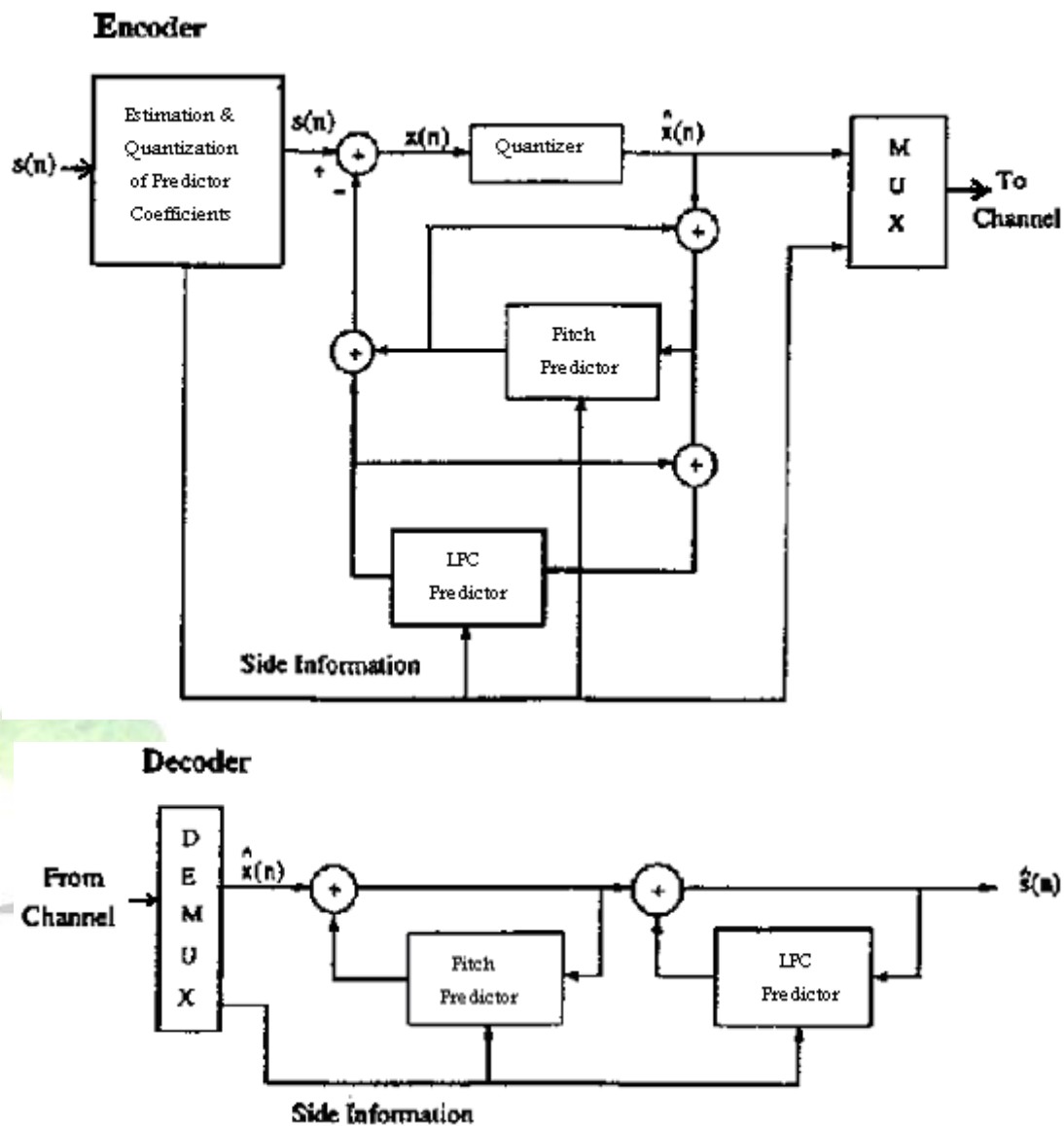
شکل (۵-۲): بلوک دیاگرام یک کدر تبدیل تطبیقی

کدرهای SBC و ATC که در بالا بحث شدند، صحبت با کیفیت بالا در 16kbps تولید می کنند اما کیفیت صحبت در نرخ بیت حدود 8kbps بدلیل عدم حضور پیشگویی Pitch افت می کند بنابراین محدوده عملیاتی آنها را 9.6kbps تا 16kbps در نظر می گیرند.

۱,۱,۱,۱,۱,۱,۱,۱,۲,۳



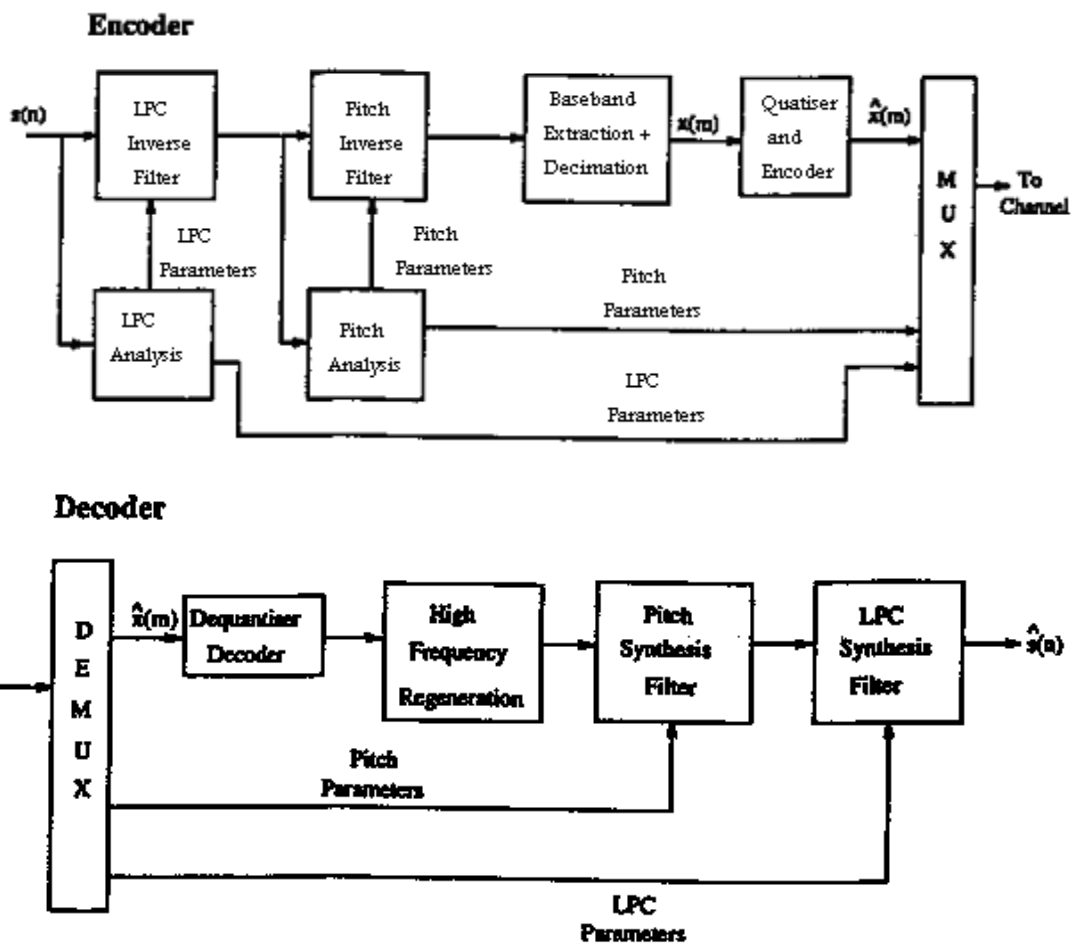
برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آر م سایت و به همراه فونت های لازمه



شکل (۶-۲): بلوک دیاگرام کدر پیشگویی تطبیقی (APC)

همانطور که در شکل (۷-۲) دیده می شود. کدر RELP در اصل یک APC است به جز اینکه تنها بخشی از (بخش فرکانس پایین) سیگنال مانده در آن ارسال می گردد. ایده RELP در این است که می توان فرض کرد، اطلاعات سیگنال مانده در نواحی فرکانس پایین (باند پایه) متمرکز شده اند و بنابراین باکد کردن تنها این قسمت می توان نرخ بیت کدینگ را کاهش داد.

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آر م سایت و به همراه فونت های لازم



شکل (۷-۲): بلوک دیاگرام کدر RELP

از اینرو در اینکدر سیگنال باند پایه بوسیله فیلتر پایین گذر و کوانتیزاسیون استخراج می شود. در دیکدر سیگنال باند پایه با استفاده از روش بازسازی فرکانس بالا [۵] به سیگنال تمام باند، تبدیل می شود. ویژگی اصلی روش RELP، توانایی کار در شرایط خیلی بد نویز زمینه می باشد و عملکرد شنیداری خوب آن محدود به 9.6 kbps و بالاتر است.

روشهای کدینگ صحبت که تا کنون بررسی شده اند بر مبنای آنالیز و سنتز هستند. یعنی سیگنال صحبت آنالیز می شود تا پارامترهای سیگنال بدون افزونگی، از آن استخراج گردد و سپس سیگنال باقیمانده کوانتیزه و ارسال می گردد و در دیکدر عمل عکس صورت می گیرد. این روش کدینگ،



## برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

استخراج پارامترها را از پرو سه کوانتیزا سیون جدا می سازد و بنابراین کنترل بر روی اعوجاج ها به کنترل زیر سیستم های جدا از هم محدود می شود .

برای داشتن کنترل بهتر بر روی تمام پروسه کدینگ، یعنی برای مینیم کردن خطای کلی در سیگنال صحبت سنتز شده ، روشهای آنالیز با سنتز (Abs<sup>۱</sup>) منظور می شوند . در روش Abs در طرف فرستنده یک دیکدر محلی وجود دارد که در آن صحبت سنتز شده برای آنالیز موجود می باشد .

روش های Abs-LPC مختلفی وجود دارد و اولین روش گزارش شده LPC چند پالسی (MPLPC<sup>۲</sup>) می

باشد که در شکل (۲-۸) نشان داده شده است . تعیین محل پالسها و دامنه آنها بوسیله یک پروسه Abs

انجام می شود . یک نمونه از MPLPC در سرویس هوایی Skyphone با نرخ بیت 9.6kbps و کدینگ

FEC کانولو شنال با نرخ  $\frac{1}{2}$  ، بکار گرفته شده است . عیب عمده MPLPC بار محاسباتی نسبتا زیاد آن

است . ترکیبی از MPLPC و RELP که برای پیاده سازی ساده تر باشد توسط Kroon گزارش شده که به

LPC با تحریک پالس منظم [RPELPC<sup>۳</sup>] موسوم است .

تحریک بهینه در این کدر ، پالس هایی با فاصله برابر و دامنه های مختلف می باشد که بار محاسباتی آنرا

کاهش می دهد. این الگوریتم کارآمد از نظر محاسباتی ، برای سیستم موبایل GSM از میان بقیه الگوریتم

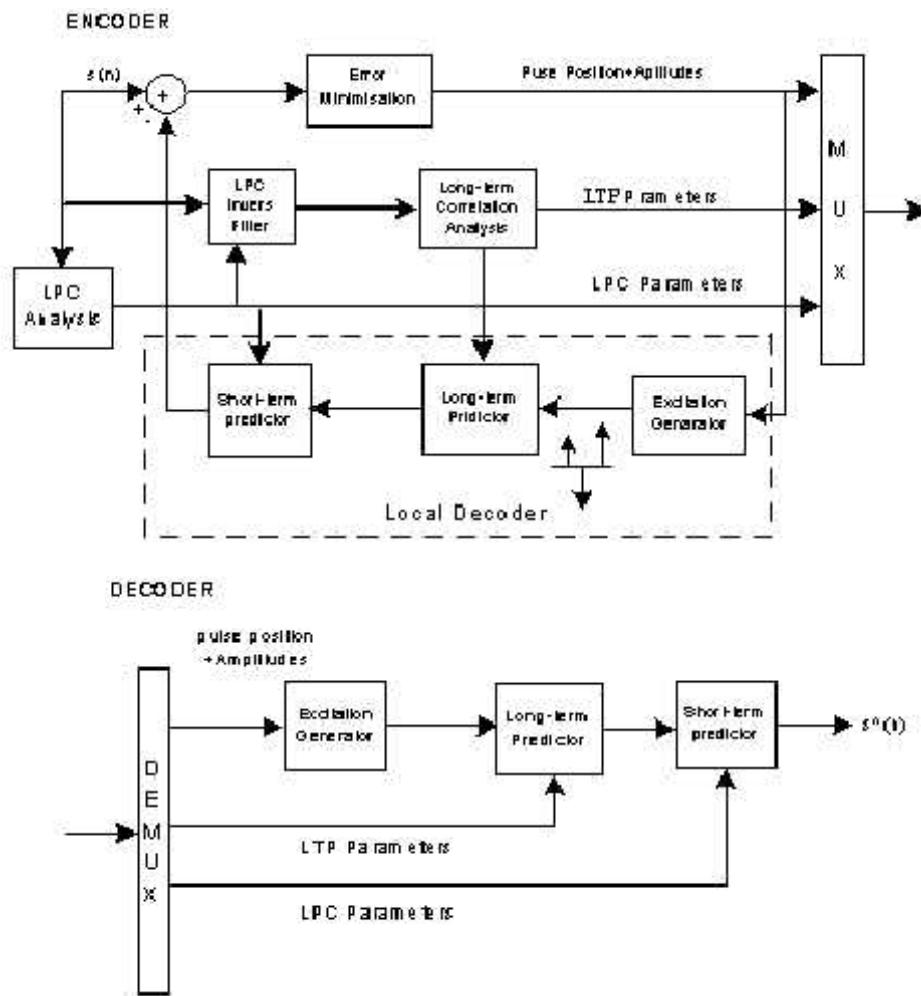
های موجود انتخاب شده است [۵] .

1 Analysis by Synthesis

2 Multi Pulse LPC

3 Regular Pulse Excited LPC

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آر سایت و به همراه فونت های لازم



شکل (۸-۲): بلوک دیاگرام MPLPC استفاده شده در Skyphone

با افزایش قدرت پردازشی چیپ های DSP، سیستم های Abs کاملی که قبلا برای پایه سازی غیر عملی بودند مورد توجه قرار گرفتند، همچون LPC با تحریک کد (CELP<sup>۱</sup>) که برای نرخ بیت های پایین بسیار مناسب است. در CELP تحریک به صورت یک بردار است که از کتاب کد تحریک انتخاب می شود. هر بردار تحریکی که حداقل خطای وزنی را داشته باشد، بعنوان تحریک بهینه انتخاب می گردد. پروسسه Abs در CELP

<sup>۱</sup> Code Excited Linear Prediction

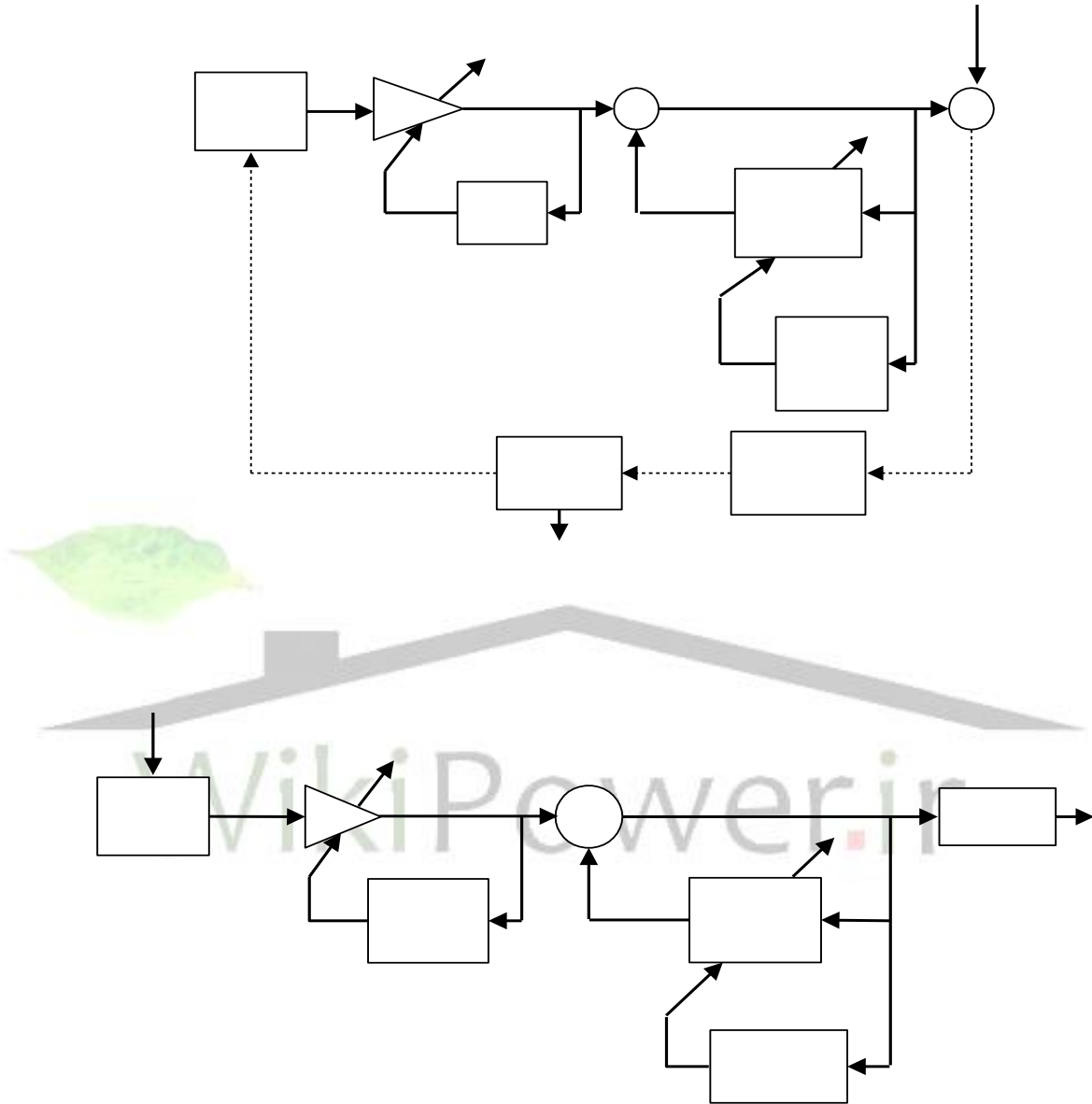
**برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید.** فاقد آرم سایت و به همراه فونت های لازمه

بار محاسباتی زیادی دارد. هر چند که CELP پیچیده است اما قادر به تولید صحبت با کیفیت عالی حتی در نرخ بیت های پایین می باشد. یک گونه از CELP بعد از بررسی روشهای مختلف کدینگ، برای سیستم مخابراتی صدا در وزارت دفاع آمریکا با عنوان 4.8 kbps DOD انتخاب شده تا جانشین کدر قدیمی LPC-10e 2.4kbps شود [۵].

هر چند CELP عمدتاً بر روی نرخ بیت های پایین متمرکز است، برای نرخ بیت های بالا تر هم مناسب بوده و استاندارد CCITT 16kbps LD-CELP نیز نوعی CELP است. بلوک دیاگرام کدر LD-CELP در شکل (۲-۹) نشان داده شده است.

همانطور که در دیاگرام دیده می شود، فیلتر طیفی بصورت برگشتی محاسبه می شود و بنابراین لازم نیست که ضرائب فیلتر از سال شوند. بعلاوه مقیاس کردن یا تنظیم بهره نیز تنها با پیشگویی انجام می شود و نتیجه مستقیم آن این است که نرخ روزآمد شدن تحریک می تواند خیلی بالا باشد (هر ۵ نمونه) و این کدر را قادر می سازد تا تاخیر کدینگ اندک حتی در حدود 2ms داشته باشد. برای یکبار کدینگ در کانالهای نویزی یا بدون نویز، کیفیت صحبت LD-CELP همانند و یا بهتر از استاندارد G. 721 یعنی 32kbps ADPCM گزارش شده است [6].

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم



شکل ( ۲-۹ ) : بلوک دیاگرام کدر LD-CELP

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

## فصل ۳

### کدر کم تاخیر LD-CELP

#### ۳-۱- مقدمه

امروزه با توجه به استفاده روزافزون از شبکه های تلفن و موبایل و نیاز به استفاده موثرتر از ظرفیت شبکه های مخابراتی، کدرهایی که در نرخ بیت متوسط (16 kbps)، صحبتی با کیفیت بالا و حداقل تاخیر ممکن تولید نمایند، مورد توجه خاصی قرار گرفته اند. تاخیر کلی در یک سیستم مخابراتی را می توان ناشی از عوامل زیر دانست:

تاخیر بافر کردن در اینکدر و دیکدر که بعلاوه جمع آوری نمونه های مورد نیاز آنالیز LPC است. تاخیر پردازش در اینکدر و دیکدر که در اثر پردازش روی نمونه های بافر شده می باشد. پردازش نمونه های بافر شده باید در فرصت بافر کردن، کامل شود چرا که باید برای پردازش نمونه های بعدی آماده بود. تاخیر ارسال که در سیستم های ماهواره ای بسیار مهم است و قابل کنترل بوسیله کدر صحبت نمی باشد. بنابراین تاخیر کلی سیستم را می توان مجموع عوامل فوق دانست و معمولا حداکثر تاخیر را

## برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

۴ برابر تاخیر بافر کردن در نظر می گیرند [۵،۶]. در نتیجه با کنترل طول بافر میتوان تاخیر کدر را کاهش داد. بعنوان مثال در کدر LD-CELP طول بافر باندازه ۵ نمونه است که با فرض نمونه برداری 8 KHz تاخیر کلی حداکثر 2.5ms می شود ولی در سیستم GSM با طول بافر 20ms تاخیر کلی حداکثر 80ms خواهد بود.

در سال ۱۹۸۸ CCITT برنامه ای را برای استاندارد سازی یک کدر 16kbps آغاز نمود از ویژگی های آن کیفیت بالا و تاخیر اندک بود و برای آن کاربردهای زیادی همچون شبکه PSTN شبکه ISDN، تلفن تصویری و غیره در نظر گرفته شد. در جدول (۳-۱) لیستی از شرایط مورد نیاز CCITT برای کدر مورد نظر آورده شده است [۵].

شرط تاخیر کمتر از 5ms مسأله ای بود که هیچ یک از کدرهای موجود قادر به برآوردن آن نبودند. از این رو کدر LD-CELP در سال ۱۹۹۱ بدین منظور معرفی شد و توازنست همه شرایط CCITT را برآورده سازد و حتی با 32kbps ADPCM رقابت کند. این کدر در سال ۱۹۹۲ بصورت استاندارد CCITT در آمد و در توصیه نامه (CCITT)ITU-T G.728 ارائه شد [۸].

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

Parameter	CCITT requirements	CCITT objective
Coding Delay	$\leq 5$ ms	$\leq 2$ ms
Quality at BER = 0	Distortion $< 4$ qds	
Quality at BER = $10^{-3}$	No worse than G.721*	
Quality at BER = $10^{-2}$	No worse than G.721	
Tandeming	3 Async Tandem with distortion $\leq 14$ qds	Sync Tandem without distortion accumulation
Signaling Tones	DTMF	

\* G.721 : CCITT standard for 32k bps ADPCM

جدول (۳-۱) : شرایط CCITT برای کدر 16 kbps

### ۳-۲- بررسی کدر کم تاخیر LD-CELP

در کدرهای معمولی CELP که تطبیقی مستقیم هستند، پارامترهای پیشگویی به همراه بردار تحریک به گیرنده ارسال می شوند ولی در LD-CELP فقط سیگنال تحریک ارسال می گردد. ضرائب پیشگویی از طریق انجام آنالیز LPC بر روی نمونه های صحبت قبلی، روز آمد می شوند. بنابراین کدر LD-CELP در اصل یک نسخه تطبیقی معکوس از کدر معمولی می باشد و اساس CELP که جستجوی کتاب کد به روش آنالیز با سنتز است در آن حفظ شده است. در شکل (۳-۱) بلوک دیاگرام ساده شده اینکدر و دیکدر LD-CELP نشان داده شده است.

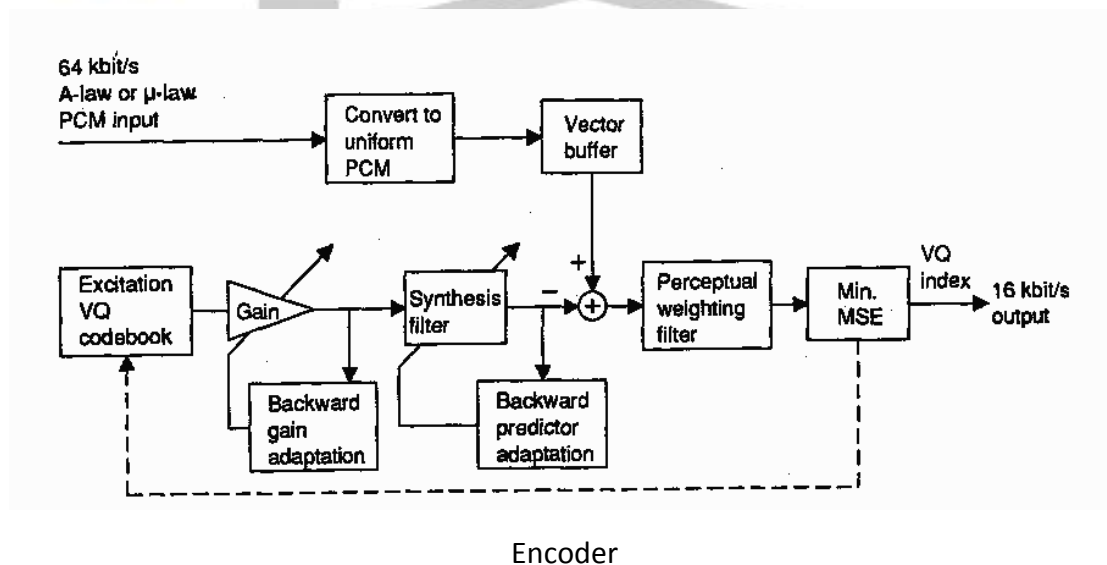
بردار تحریک در اینجا به اندازه ۵ نمونه است و پیشگویی زمان بلند رایج در CELP با پیشگویی زمان کوتاه مرتبه بالای ۵۰، بر روی نمونه های صحبت کوانتیزه که ضرائب آن برای هر ۴ بردار تحریک روز آمد می شوند، تعویض شده است.

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

بهره تحریک با استفاده از پیشگویی خطی تطبیقی مرتبه ۱۰ بصورت لگاریتمی، برای هر بردار روز آمد می شود. ضرائب این پیشگویی لگاریتمی بهره ، هر ۴ بردار یک بار با انجام آنالیز LPC بر روی بهره های لگاریتمی بردارهای تحریک قبلی روز آمد می گردد.

فیلتر مرتبه ۱۰ شنیداری نیز در اینکدر لحاظ شده که هر ۴ بردار تحریک یک بار با انجام آنالیز LPC روی نمونه های صحبت اصلی در اینکدر روز آمد می شود.

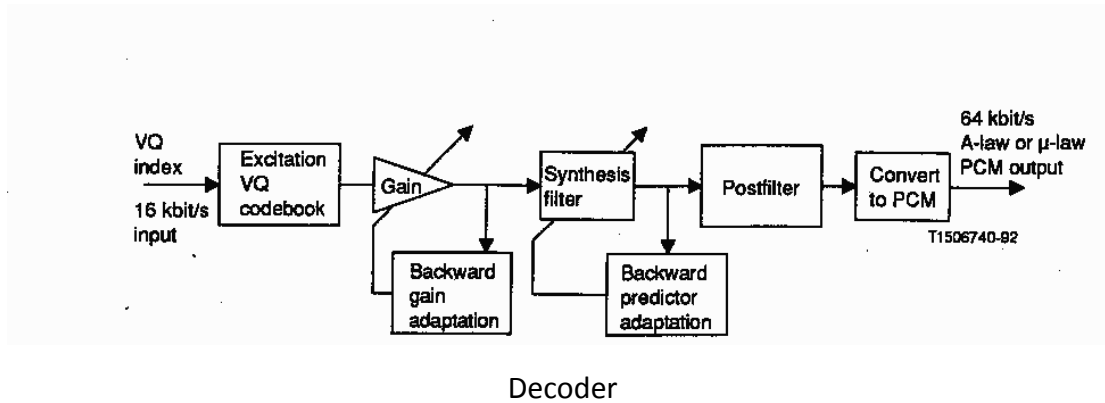
کتاب کد کوانتیزه برداری از کتاب کدهای ۳ بیتی بهره و ۷ بیتی شکل<sup>۱</sup> در 16kbps تشکیل شده و تنها اندیس ۱۰ بیتی این کتاب کد به گیرنده ارسال می شود .



<sup>۱</sup> Shape



برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

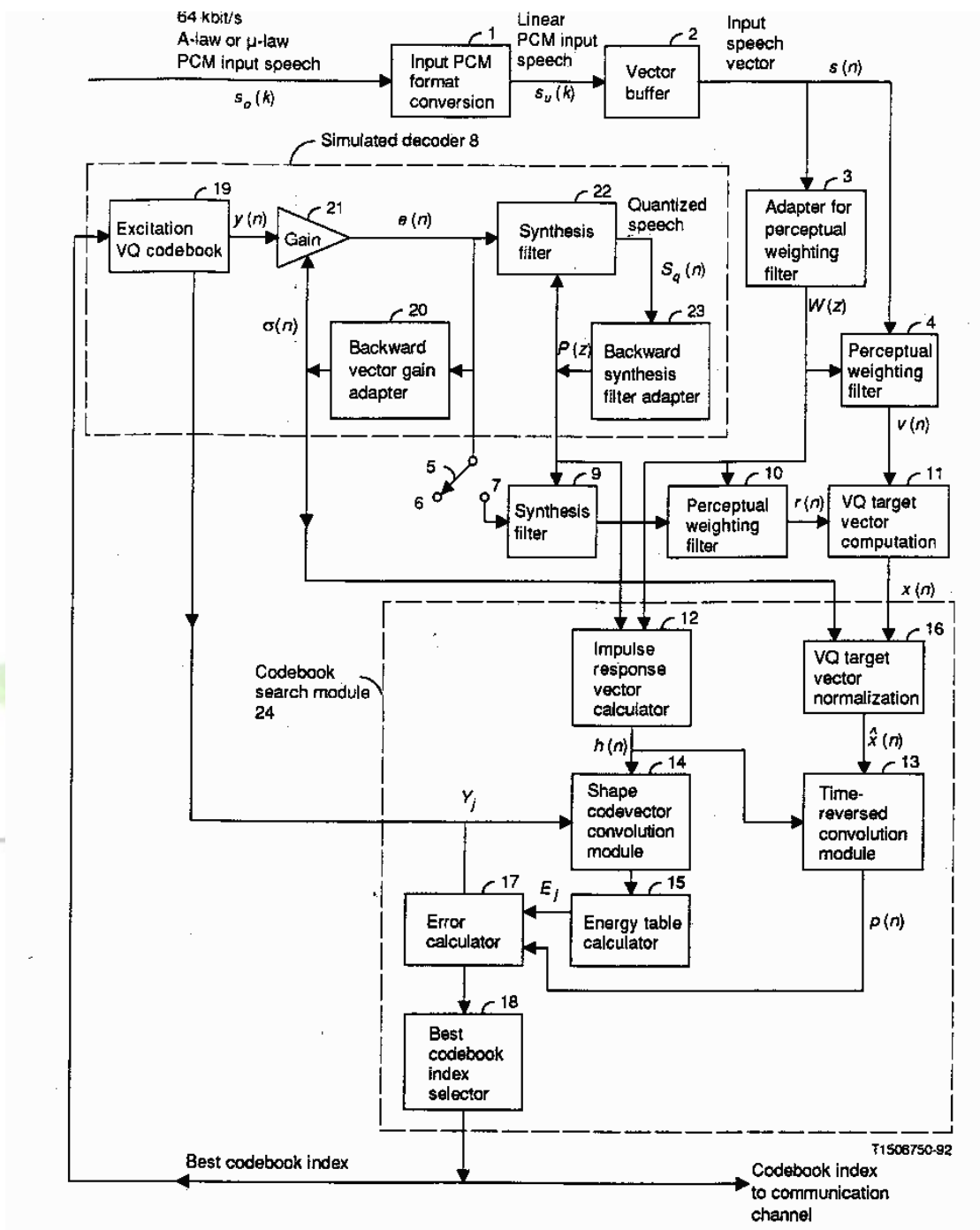


شکل (۳-۱) : بلوک دیاگرام ساده شده LD-CELP 16kbps

در شکل (۳-۲) بلوک دیاگرام اینکدر LD-CELP ، با جزئیات بیشتری نشان داده شده است. در این قسمت به بررسی بیشتر آن می پردازیم .



برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه



شکل (۳-۲): طرح اینکدر LD-CELP

۳-۲-۱- LPC معکوس مرتبه بالا

## برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

از آنجا که پیشگویی زمان بلند (LTP<sup>۱</sup>) pitch ، سیستم را خیلی به خطاهای کانال حساس می سازد ، در الگوریتم LD-CELP , LTP حذف شده و بجای آن از LPC معکوس مرتبه بالا استفاده شده است .

کاربرد LPC مرتبه ۵۰ دارای امتیازهای زیر است :

- حساسیت خیلی پایین کدر به خطاهای کانال
- اطلاعات جنبی دیگر ارسال نمی شوند .
- با حذف LTP ، الگوریتم کمتر به صحبت وابسته می شود و قادر به دریافت داده باند صوتی می شود .

البته استفاده از LPC مرتبه ۵۰ ، باعث افزایش تعداد محاسبات کورلیشن و پیچیدگی زیاد الگوریتم می شود و برای کاهش پیچیدگی از روش پنجره بازگشتی Branwell استفاده شده است [5].

هر چند که این روش باعث کاهش پیچیدگی می شود ولی بخاطر عملیات ضرب باعث بروز مسائلی در پیاده سازی ممیز ثابت می شود از اینرو از یک پنجره مختلط جدید استفاده می شود .

فرض کنید که آنالیز LPC برای هر L نمونه صحبت انجام شود و نمونه های مطابق با سیکل تطبیق فعلی

$\hat{S}_u(m), \hat{S}_u(m+1), \dots, \hat{S}_u(m+L-1)$  باشند و پنجره مختلط به نمونه های قبلی با اندیس کمتر از

m اعمال شود ، در زمان m تابع پنجره مختلط  $W_m(k)$  بصورت زیر تعریف می شود :

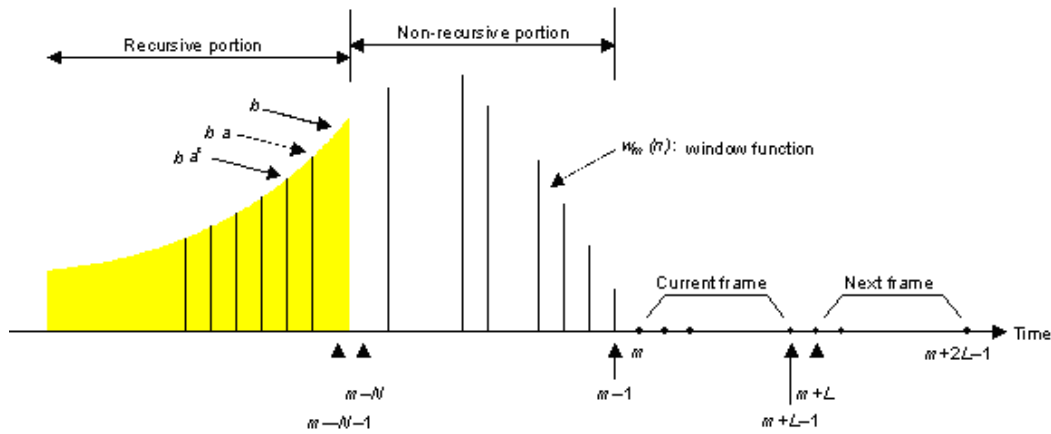
$$W_m(k) = \begin{cases} f_m(k) = b\alpha^{-[k-(m-N-1)]}; k \leq m-N-1 \\ g_m(k) = -\sin[c(k-m)]; m-N \leq k \leq m-1 \\ 0; k \geq m \end{cases}$$

که در آن  $0 < b < 1$  و  $0 < \alpha < 1$  و N طول بخش غیر بازگشتی پنجره هستند .

<sup>۱</sup> Long Term Prediction

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

در شکل (۳-۳) نمایش پنجره مختلط نشان داده شده است [8].



شکل (۳-۳): پنجره مختلط

صحبت پنجره شده بصورت زیر محاسبه می شود:

$$S_m(k) = \hat{S}_u(k)W_m(k) = \begin{cases} \hat{S}_u(k)f_m(k); k \leq m-N-1 \\ \hat{S}_u(k)g_m(k); m-N \leq k \leq m-1 \\ 0; k \geq m \end{cases}$$

فیلتر LPC مرتبه M از M+1 ضریب اتوکورلیشن بصورت زیر استفاده می کند:

$$R_m(i) = \sum_{k=-\infty}^{m-1} S_m(k)S_m(k-i) = r_m(i) + \sum_{k=m-N}^{m-1} S_m(k)S_m(k-i)$$

که در آن  $r_m(i)$  بخش بازگشتی  $R_m(i)$  چنین است:

$$r_m(i) = \sum_{k=-\infty}^{m-N-1} S_m(k)S_m(k-i) = \sum_{K=-\infty}^{m-N-1} \hat{S}_u(k)\hat{S}_u(k-i)f_m(k)f_m(k-i)$$

فرض کنید که مقادیر  $r_m(i)$  در فریم حاضر معلوم هستند و ما می خواهیم، فریم بعدی را محاسبه کنیم

که از  $\hat{S}_u(m+L)$  شروع می شود. صحبت پنجره شده در این حالت:

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

$$S_{m+L}(k) = \hat{S}_u(k)W_{m+L}(k)$$

و بخش بازگشتی  $R_{m+L}(i)$  را می توان بصورت زیر نوشت:

$$r_{m+L}(i) = \sum_{k=-\infty}^{m+L-N-1} S_{m+L}(k)S_{m+L}(k-i)$$

$$r_{m+L}(i) = \sum_{k=-\infty}^{m-N-1} S_{m+L}(k)S_{m+L}(k-i) + \sum_{k=m-N}^{m+L-N-1} S_{m+L}(k)S_{m+L}(k-i)$$

با قرار دادن  $\hat{S}_u(k)$  و تابع پنجره در بخش اول داریم:

$$r_{m+L}(i) = \alpha^{2L}r_m(i) + \sum_{k=m-N}^{m+L-N-1} S_{m+L}(k)S_{m+L}(k-i)$$

و سرانجام ضرایب اتوکورلیشن بصورت زیر محاسبه می شوند:

$$R_{m+L}(i) = r_{m+L}(i) + \sum_{k=m-N}^{m+L-N-1} S_{m+L}(k)S_{m+L}(k-i)$$

بعد از محاسبه ضرایب اتوکورلیشن با  $N=35$  و  $\alpha = \left(\frac{3}{4}\right)^{\frac{1}{40}}$  و  $L=20$ ، از الگوریتم Durbin برای محاسبه

پارامترهای LPC استفاده می کنیم [8]. همچنین ضریب تحصحیح نویز  $\frac{257}{256}$  را برای اصلاح  $R(0)$  بکار می

بریم که معادل اضافه کردن نویز سفید 24dB زیر سطح متوسط صحبت می باشد. این کار باعث پر شدن

چاله های طیفی با نویز سفید و کاهش رنج دینامیک طیفی می شود. قبل از بکارگیری

پارامترهای LPC محاسبه شده، آنها را با ضریب گسترش پهنای باند  $\lambda = \frac{253}{256}$ ، بصورت زیر اصلاح می

کنیم:

$$\hat{a}_i = \lambda^i a_i, i = 0, 1, \dots, 49$$

که در آن  $a_i$  و  $\hat{a}_i$  به ترتیب مقدار اولیه و گسترش یافته پارامترهای LPC می باشند [5].

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

### ۳-۲-۲- فیلتر وزنی شنیداری

فیلتر وزنی شنیداری استفاده شده در LD-CELP (بلوک ۱۰ در شکل (۲-۳))، شبیه فیلتر شکل دهنده نویز به فرم کلی زیر است:

$$W(Z) = \frac{1 + \sum_{i=1}^{10} q_i \gamma_1^i Z^{-i}}{1 + \sum_{i=1}^{10} q_i \gamma_2^i Z^{-i}}$$

که در آن  $\gamma_1$  و  $\gamma_2$  ضرایب تنظیم هستند که به ترتیب برابر 0.9 و 0.6 قرار می گیرند. ضرایب  $q_i$  فیلتر

به همان روش ضرایب فیلتر سنتز، با  $M=10$ ،  $L=20$  و  $N=30$  و  $\alpha = \left(\frac{1}{2}\right)^{40}$  محاسبه می شوند.

همچنین ضریب تصحیح نویز  $\frac{257}{256}$  نیز در اینجا اعمال می شود.

### ۳-۲-۳- ساختار کتاب کد

در شکل (۲-۳) بلوک های ۱۲ تا ۱۸ تشکیل دهنده بلوک جستجوی کتاب کد (۲۴) هستند. این بلوک،

۱۰۲۴ بردار تحریک کتاب کد کوانتیزه برداری (VQ) بلوک ۱۹ را بررسی کرده و اندیس

بهترین بردار که بردار صحبت کوانتیزه متناظر آن، به بردار صحبت ورودی از همه نزدیکتر باشد را

مشخص می سازد. جهت ساده شدن جستجو، کتاب کد ۱۰۲۴ مدخلی ۱۰ بیتی به دو کتاب کد کوچکتر

تقسیم می شود: یک کتاب کد شکل ۷ بیتی که شامل ۱۲۸ بردار کد مستقل و یک کتاب کد بهره ۳ بیتی

شامل ۸ مقدار اسکالر متقارن نسبت به صفر (یعنی ۱ بیت برای علامت و ۲ بیت برای اندازه). بردار

کد نهایی حاصل ضرب بهترین بردار شکل و بهترین سطح بهره خواهد بود.

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

### ۳-۲-۱- جستجوی کتاب کد

بلوک جستجوی ۲۴ در واقع هر بردار کد را در بهره تحریک کنونی  $\sigma(n)$  ضرب کرده و از فیلترهای متوالی سنتز  $F(z)$  و وزن دهنده شنیداری  $W(z)$  عبور می دهد.

فیلتر کردن بردارهای کد  $VQ$  را میتوان بدین صورت بیان کرد: فرض کنید که  $\gamma$ ،  $i$  امین بردار کد در کتاب کد شکل  $\gamma$  بیتی و  $g_i$ ،  $i$  امین سطح در کتاب کد بهره باشد. اگر  $h(n)$  پا سخ ضربه فیلتر متوالی  $H(z)=F(z)W(z)$  باشد، وقتیکه بردار کد مشخص شده با اندیس های  $i$  به  $H(z)$  اعمال شود خروجی فیلتر بصورت زیر خواهد شد:

$$\tilde{x}_{ij} = H\sigma(n)g_i y_j$$

که در آن

$$H = \begin{bmatrix} h(0) & 0 & 0 & 0 & 0 \\ h(1) & h(0) & 0 & 0 & 0 \\ h(2) & h(1) & h(0) & 0 & 0 \\ h(3) & h(2) & h(1) & h(0) & 0 \\ h(4) & h(3) & h(2) & h(1) & h(0) \end{bmatrix}$$

بلوک ۲۴ بدنبال ترکیبی از  $i$  می گردد که خطای میانگین مربعی (MSE) زیر را مینیمم سازد:

که در آن  $\hat{x}(n) = x(n)/\sigma(n)$  بردار هدف نرمالیزه بهره می باشد:

$$D = \sigma^2(n) \left[ \|\hat{x}(n)\|^2 - 2g_i \hat{x}^t(n)H \|\hat{x}(n)\|g_i \|\hat{y}_j\|^2 + \|\hat{H}(n)\hat{y}_j\|^2 \right]$$

چون عبارت های  $D = -2g_i p^t(n)y_j + g_i E_j$  در طول جستجوی کتاب کد ثابت هستند باید:

مینیمم شود که در آن:

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

$$p(n) = H^t \hat{x}(n), E_j = \|Hy_j\|^2$$

توجه کنید که  $E_j$  انرژی زامین بردار شکل است و به بردار هدف  $\hat{x}(n)$  بستگی ندارد. همچنین بردار

شکل  $y_j$  و ماتریس  $H$  تنها به فیلترستز و فیلترز وزنی بستگی دارد که در دوره  $\epsilon$  بردار صحبت ثابت است

و در نتیجه  $E_j$  هم در این دوره ثابت می ماند.

برای کاهش بیشتر محاسبات می توان آرایه های  $b_i = 2g_i$  و  $c_i = g_i^2$  ( $i=0,1,\dots,7$ ) را از قبل محاسبه و

ذخیره ساخت.

در نتیجه:

$$\hat{D} = -b_i P_j + c_i E_j$$

که در آن:

$$P_j = p^t(n) y_j$$

بنابر این برای جستجوی کتاب کد ابتدا باید برای هر بردار شکل  $y_j$ ، بهترین اندیس بهره  $i$  که  $\hat{D}$  را

مینیمم سازد مشخص کرده و سپس این کار را برای همه  $y_j$  ها انجام داده تا در نهایت

$j_{\min}$ ،  $i_{\min}$  مینیمم بدست آید.

۳-۲-۴- شبه دیکدر



## برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

تا کنون کدکننده بهترین اندیس کتاب کد تحریک را از سال داشته ولی هنوز بعضی کارهای آماده سازی برای کدینگ بردارهای بعدی باقی مانده است. ابتدا این اندیس به کتاب کد VQ تحریک اعمال شده تا بهترین بردار تحریک  $y_{jmin}(n)$  پیدا شود. سپس این بردار در بهره تحریک کنونی

ضرب می شود (بلوک ۲۵). بردار تحریک بدست آمده برابر است با:

$$e(n) = \sigma(n)y(n)$$

بردار تحریک از فیلتر سنتز (۲۲) عبور می کند تا بردار صحبت کوانتیزه فعلی  $q(n)$  تولید کند. توجه کنید که بلوک های ۱۹ تا ۲۳ شبه دیکدر (۸) را شکل می دهند. از این رو بردار صحبت کوانتیزه برابر صحبت  $q(n)$  شده شبیه سازی در کانال بدون خطا است.

در شکل (۲-۳) تطبیق دهنده فیلتر سنتز (۲۳)، بردار  $q(n)$  را برای تجدید ضرائب فیلتر سنتز نیاز دارد. همچنین تطبیق دهنده بهره (۲۰)، بردار تحریک  $e(n)$  را جهت تجدید

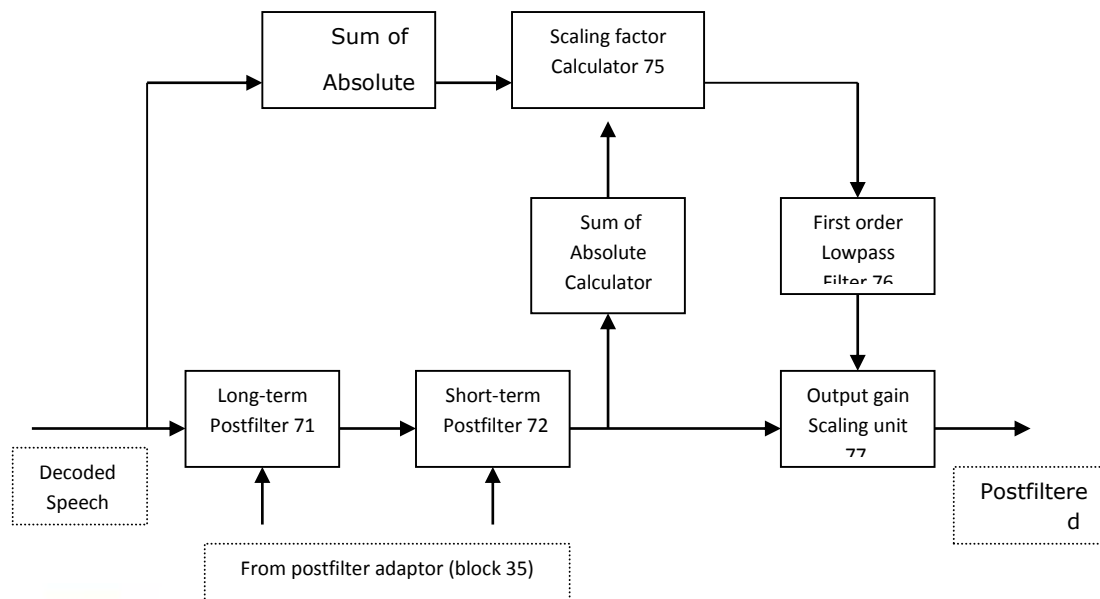
ضرائب log-gain لازم دارد.

### ۳-۲-۵- پست فیلتر<sup>۱</sup>

این بلوک جهت افزایش کیفیت شنیداری، صحبت دیکد شده را فیلتر می کند. جزئیات بیشتر این بلوک در شکل (۴-۳) نشان داده شده است. پست فیلتر در اصل شامل ۳ بخش است: پست فیلتر زمان-بلند ۷۱، پست فیلتر زمان-کوتاه ۷۲ و واحد مقیاس کننده بهره خروجی ۷۷. بقیه بلوکهای شکل (۴-۳) برای محاسبات بلوک ۷۷ هستند.

<sup>۱</sup> Postfilter

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آر سایت و به همراه فونت های لازمه



شکل (۳-۴): طرح بلوکی پست فیلتر

پست فیلتر زمان-بلند (۷۱) که گاهی پست فیلتر pitch نامیده می شود، یک فیلتر شانه ای<sup>۱</sup> است که پیکهای طیفی آن در مضارب فرکانس pitch صحبتی که باید پست فیلتر شود، واقع شده است. پریود pitch را میتوان از صحبت دیکد شده با استفاده از یک آشکارساز pitch استخراج کرد. اگر p پریود pitch (به نمونه) بدست آمده از آشکارساز pitch باشد، آنگاه تابع انتقال پست فیلتر زمان-بلند بصورت زیر میتواند بیان شود:

$$H(z) = g_l(1 + bz^{-p})$$

<sup>۱</sup> Comb

## برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

که در آن ضرائب  $g_i, b_i$  و پریود pitch در تطبیق دهنده پست فیلتر ساخته و در هر  $\frac{1}{2}$  بردار (یک فریم) تجدید می شوند .

پست فیلتر زمان-کوتاه (۷۲) شامل یک فیلتر مرتبه ۱۰ صفر-قطب بدنبال یک فیلتر تمام-صفر مرتبه ۱ است. فیلتر مرتبه ۱۰ صفر-قطب، مؤلفه های فرکانسی بین پیک های فرمنت را تضعیف می کند. در حالیکه فیلتر مرتبه اول تمام-صفر می کوشد تا پاسخ فرکانسی فیلتر صفر-قطب مرتبه ۱۰ را هموار سازد.

فرض کنید که  $a_i, i=1,2,\dots,10$  ضرائب LPC مرتبه ۱۰ و  $K_1$  اولین ضریب انعکاس حاصل از

آنالیز معکوس LPC صحبت دیکد شده باشند آنگاه  $a_i$  و  $K_1$  را میتوان باهم از آنالیز PC

معکوس مرتبه ۵۰ (بلوک ۳۳ در شکل (۳-۲)) بدست آورد. فقط باید تکرار Durbin

مرتبه ۵۰ را در مرتبه ۱۰ متوقف کرده و ضرائب فوق را ذخیره کرد و

سپس تکرار Durbin را از مرتبه ۱۱ تا ۵۰ ادامه داد.

تابع انتقال پست فیلتر زمان-کوتاه بصورت زیر است:

$$H_s(z) = \frac{1 - \sum_{i=1}^{10} b_i z^{-i}}{1 - \sum_{i=1}^{10} a_i z^{-i}} [1 + \mu z^{-1}]$$

$$b_i = \tilde{a}_i (0.65)^i, i = 1, 2, \dots, 10$$

$$a_i = \tilde{a}_i (0.75)^i, i = 1, 2, \dots, 10$$

$$\mu = 0.15 K_1$$

که در آن ضرائب  $a_i, b_i$  در هر فریم یک بار و در اولین بردار آن تجدید می شوند. در حالت کلی بعد از

اینکه صحبت دیکد شده از پست فیلتر زمان-بلند و زمان-کوتاه می گذرد، صحبت فیلتر شده همان

برای دریافت فایل Word پروژه به سایت **ویکی پاور** مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

انرژی اولیه را ندارد و بدین منظور بلوکهای ۷۳ تا ۷۷ کار کنترل اتوماتیک بهره و حفظ انرژی صحبت را انجام می دهند .



برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

## فصل ۴

شبیه سازی ممیزثابت الگوریتم به زبان

### C

۴-۱- مقدمه

شبیه سازی یک الگوریتم بر روی کامپیوتر بو سیله یک زبان سطح بالا مانند C، گام نخست جهت پیاده سازی آن الگوریتم بر روی DSP به شمار می رود. همانطور که در فصل ۵ بیان خواهد شد، با توجه به پیاده سازی کدک G.728 بر روی DSP های ممیزثابت، لازم است که ابتدا الگوریتم کدک را بصورت ممیزثابت شبیه سازی نماییم. در این فصل ابتدا ویژگی های برنامه نویسی ممیزثابت را شرح میدهم. سپس ساده سازی محاسبات برای برنامه نویسی ممیزثابت را بیان نموده و پس از آن بلوک ها و متغیرهای برنامه را به اختصار شرح می دهیم. در قسمت آخر هم روندنمای برنامه اینکدر و دیکدر را توصیف می کنیم.

## برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

۴-۲- ویژگی های برنامه نویسی ممیز ثابت

۴-۲-۱- نمایش اعداد

واحد پایه در پیاده سازی ۱۶ بیتی ، کلمه ۱۶ بیتی است که برای اعداد صحیح علامت دار بین ۳۲۷۶۷ تا ۳۲۷۶۸- تغییر می کند و بصورت مکمل ۲ ذخیره می گردد. برای نمایش اعداد اعشاری یک نقطه اعشاری باید بین دویست کلمه در نظر گرفته شود. مثلاً برای نمایش اعداد بین  $1, +1 -$  بایستی نقطه اعشاری بین بیت های ۱۴ و ۱۵ قرار گیرد. این فرمت خاص Q15 نامیده می شود زیرا ۱۵ بیت در سمت راست نقطه اعشاری وجود دارد.  $Qn$  فرمت بدین صورت تعریف می شود که  $n$  بیت در سمت راست نقطه اعشاری وجود داشته باشد. اعداد صحیح بصورت  $Q0$  نمایش داده می شوند.

برخی داده ها به دقت بیشتری نیاز دارند از اینکه با یک کلمه ۱۶ بیتی نمایش داده شوند. مانند رجیسترهای ضرب و اکومولاتور در چیپ های DSP که می توانند اعداد با دقت از  $1$  تا  $2^{31}$  را نمایش دهند که به دقت مضاعف معروف است.

برخی داده ها رنج وسیعتری دارند از اینکه بتوان با یک فرمت ۱۶ بیتی ثابت نمایش داده شوند. شاید ۱۶ بیت دقت کافی باشد ولی باید از مقیاس کردن دینامیک استفاده شود. این نوع داده ها را می توان بصورت ممیز شناور و با دقت معمولی نشان داد. یعنی اینکه داده با ۲ کلمه نشان داده می شود، اولین کلمه حاوی عددی است که اندازه آن بین ۱۶۳۸۴ و ۳۲۷۶۷ می باشد. این ماننسیس مقدار است و می گوئیم که مقدارش در فرمت نرمالیزه نمایش داده می شود اگر مقدار مثبت باشد، بیت ۱۴ ماننسیس ۱ است. کلمه دوم حاوی تعداد شیفت به چپ ( $NLS^1$ ) هایی است که بکار رفته تا عدد را در مقدار نرمالیزه اش قرار دهد.

<sup>1</sup> Number of Left Shift

## برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

بنابراین کلمه دوم مشخص کننده  $Q$  فرمت مانتیس می باشد. اگر این فرمت برای یک مقدار منفرد بکار رود ممیز شناور اسکالر نامیده می شود. همچنین امکان دارد که یک آرایه از  $n$  مقدار را با  $n+1$  کلمه، با استفاده از ممیز شناور بلوکی نمایش داد. با استفاده از این فرمت، بزرگترین مقدار در آرایه باید به همان صورتی که برای ممیز شناور اسکالر گفته شد، نمایش داده شود. بقیه مقادیر در آرایه باید از همان NLS استفاده کنند و مانتیس آنها لزوماً نباید در فرمت نرمالیزه باشد. گسترش این نمایش بصورت ممیز شناور بلوکی قسمت شده می باشد. در این حالت یک آرایه  $mn$  مقداری بوسیله  $m(n+1)$  کلمه نمایش داده می شود. این آرایه تقسیم می شود به  $m$  زیر آرایه با اندازه  $n$ ، که هر زیر آرایه بصورت ممیز شناور بلوکی با  $n$  کلمه برای اندازه ها و ۱ کلمه برای NLS نمایش داده می شود.

نوع دیگری از نمایش بکار رفته بصورت ممیز شناور با دقت مضاعف است. در این حالت اعداد صحیح با دقت مضاعف بعنوان مانتیس بکار می روند و یک کلمه با دقت معمولی برای NLS استفاده می شود. در مجموع انواع مختلف نمایش ها عبارتند از ممیز ثابت با دقت معمولی، ممیز ثابت با دقت مضاعف برای اکومولاتورها و رجیسترهای ضرب، ممیز شناور اسکالر با دقت معمولی و ممیز شناور بلوکی با دقت معمولی و مضاعف.

### ۴-۲-۲- عملیات حسابی

از آنجا که ضرب دو کلمه ۱۶ بیتی یک عدد ۳۲ بیتی تولید می کند، رجیسترهای ضرب و اکومولاتورها باید حداقل ۳۲ بیتی باشند. در محاسبات مجموع حاصل ضرب ها مانند کانولوشن و فیلتر کردن IIR, FIR ممکن است که اکومولاتور سرریز (overflow) شود که بطور جداگانه مشخص می گردد. در فیلتر

## برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

کردن IIR مجموع حاصل ضربها یا نتیجه عملیات ضرب و انباشت (MAC) بخشی از حافظه فیلتر برای مرحله بعدی است و ۱۶ بیت بالای خروجی در ورودی ضرب کننده استفاده می شود. یک سرریز که موجب می شود تا یک عدد بزرگ مثبت به یک عدد بزرگ منفی تبدیل شود، موجب اختلاف زیادی در خروجی فیلتر می گردد. از اینرو از مد اشباع در فیلترهای IIR استفاده می کنیم. مد اشباع یعنی اینکه اگر کلمه بالایی بزرگتر از ۳۲۷۶۷ یا کوچکتر از -۳۲۷۶۸- باشد آنگاه مقدار آن برش می خورد (به این دو حد محدود می گردد).

الف- شیفت و رند کردن

حاصل ضرب یک مقدار  $Q_n$  فرمت در یک مقدار  $Q_m$  فرمت در رجیستر حاصل ضرب بصورت  $(n+m)$  فرمت و با دقت مضاعف ذخیره می شود. اگر لازم باشد که این نتیجه ذخیره گردد یا با دقت دیگری جمع شود، آنگاه نتیجه باید شیفت داده شود و یا رند گردد تا به دقت مناسب برسد. نکته ای که در عملیات شیفت به راست مکمل ۲ وجود دارد اینست که فرض کنید عدد ۳ را یک بیت به راست می خواهیم شیفت دهیم، نمایش ۱۶ بیتی ۳ بصورت 0000000000000011 است که بعد از شیفت، به 0000000000000001 تبدیل می گردد.

حال اگر عدد ۳- را بخوایم یک شیفت به راست دهیم داریم:

$$-3 = 1111111111111101 \quad \gg 1 \quad 1111111111111110 = -2$$

در این حالت اندازه مقادیر با هم برابر نیستند و در پیاده سازی باید از این نکته آگاه بود.



## برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

همچنین ممکن است که در یک الگوریتم تعداد شیفت به راست از اندازه کلمه بیشتر گردد. مثلاً یک کلمه ۱۶ بیتی باندازه ۱۸ بیت شیفت داده شود که با انجام ۱۸ بار شیفت یک بیتی حاصل ۰ یا ۱- می شود البته بسته به علامت اولیه آن عدد. اما در بعضی از کمپایلرها ممکن است که این کار خطا گرفته شود و این موضوع را باید در نظر داشت.

رند کردن فرایند تبدیل از دقت مضاعف به دقت معمولی در اکومولاتور است. که قبل از ذخیره به صورت ۱۶ بیتی در حافظه، صورت می گیرد. معمولاً با ارزشترین بیت (MSB) در کلمه پایین اکومولاتور چک می شود، اگر این بیت ۱ باشد آنگاه یکی به مقدار کلمه بالایی اضافه می شود و این کلمه بعنوان حاصل رند کردن در نظر گرفته می شود. البته در این حالت باید از سرریز نشدن اکومولاتور مطمئن شد

### ب- تقسیم

تقسیم به اندازه جمع و ضرب زیاد استفاده نمی شود. تنها تقسیمی که بکار می رود بصورت ممیز شناور اسکالر می باشد. صورت و مخرج و خارج قسمت به فرمت نرمالیزه نمایش داده می شوند. NLS خارج قسمت با تفریق NLS مخرج از NLS صورت و جمع با ۱۴ محاسبه می شود. چرا که اگر صورت کمی از مخرج بزرگتر باشد و هر دو دارای  $NLS=0$  باشند خارج قسمت باید دارای  $NLS=14$  باشد تا بصورت صحیح نرمالیزه باشد. اگر ماننسی صورت کوچکتر از ماننسی مخرج باشد آنگاه صورت یکی به چپ شیفت داده می شود و به  $NLS$  آن یکی اضافه می شود سپس  $NLS$  مخرج محاسبه می گردد و این تضمین می کند که ماننسی خارج قسمت بصورت نرمالیزه باشد.

### ۴-۳- ساده سازی محاسبات الگوریتم

## برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

در این قسمت لازم است که برخی از محاسبات الگوریتم را برای پیاده سازی ممیز ثابت مناسب ساخته و بار محاسباتی آنها را کاهش دهیم.

### ۴-۳-۱- تطبیق دهنده بهره

در اینجا لازم است که ابتدا عملیات ممیز شناور تطبیق دهنده بهره را به اختصار شرح دهیم. آرایه حالت بهره GSTATE که با علامت  $\delta$  نشان داده می شود حاوی ۱۰ بهره لگاریتمی بدون افست می باشد. علامت  $\delta(n)$  بهره را برای بردار  $n$  نشان می دهد. خروجی پیشگویی بهره برای بردار  $n$  بصورت زیر است:

$$(۱) \quad \hat{\delta}(n) = -\sum_{i=1}^{10} \alpha_i \delta(n-i)$$

مطابق با شکل (۴-۱) قبل از تبدیل  $\hat{\delta}(n)$  به حالت خطی یک افست بهره 32db باید به آن اضافه شده و آزمایش شود که:

$$(۲) \quad 0 \leq \hat{\delta}(n) + 32 \leq 60$$

تخمین بهره در حالت خطی بصورت زیر خواهد بود:

$$(۳) \quad \sigma(n) = 10^{(\hat{\delta}(n)+32)/20}$$

مقدار  $\sigma(n)$  در ابتدا برای نرمالیزه کردن بردار هدف VQ تحریک بکار می رود. بعد از اینکه جستجوی کتاب کد کامل شد،  $\sigma(n)$  برای مقیاس کردن بهترین بردار کد انتخابی بکار می رود. اگر فرض کنیم که اندیس بهره  $i$  و اندیس شکل  $j$  برای بردار  $n$  انتخاب شده باشد بردار تحریک  $e(n)$  بصورت زیر خواهد بود:

$$(۴) \quad e(n) = \sigma(n) g_i y_j$$

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

که در آن  $y_j$ ،  $z_j$  امین بردار شکل و  $g_i$ ،  $i$  امین سطح بهره در کتاب کد بهره می باشند. آنگاه بردار تحریک  $e(n)$  برای محاسبه  $\delta(n)$  بکار می رود. توان  $e(n)$  بصورت زیر است:

$$(5) \quad P[e(n)] = \frac{1}{5} \sum_{k=1}^5 e_k^2(n)$$

قبل از تبدیل  $P[e(n)]$  به مقیاس لگاریتمی  $P[e(n)]$  را به مقادیر بزرگتر یا مساوی ۱ برش می زنیم یعنی:

$$(6) \quad P[e(n)] \geq 1$$

این از سرریز شدن در تبدیل لگاریتمی جلوگیری می کند و در نهایت  $\delta(n)$  بصورت زیر برای پیشگویی بهره تحریک بعدی حاصل می گردد:

$$(7) \quad \delta(n) = 10 \log_{10} P[e(n)] - 32$$

حالا روشی که از نظر ریاضی معادل روش فوق و مناسب برای پیاده سازی ممیز ثابت می باشد را شرح می دهیم. فرض کنید که  $y_{jk}$   $k$  امین عنصر  $z_j$  امین بردار شکل باشد. از ترکیب معادلات (۵ و ۴) داریم:

$$(8) \quad \begin{aligned} P[e(n)] &= \frac{1}{5} \sum_{k=1}^5 (\sigma(n) g_i y_{jk})^2 \\ &= \sigma^2(n) g_i^2 \left[ \frac{1}{5} \sum_{k=1}^5 y_{jk}^2 \right] \\ &= \sigma^2(n) g_i^2 P[y_i] \end{aligned}$$

با جایگذاری معادله (۸) در معادله (۷) داریم:

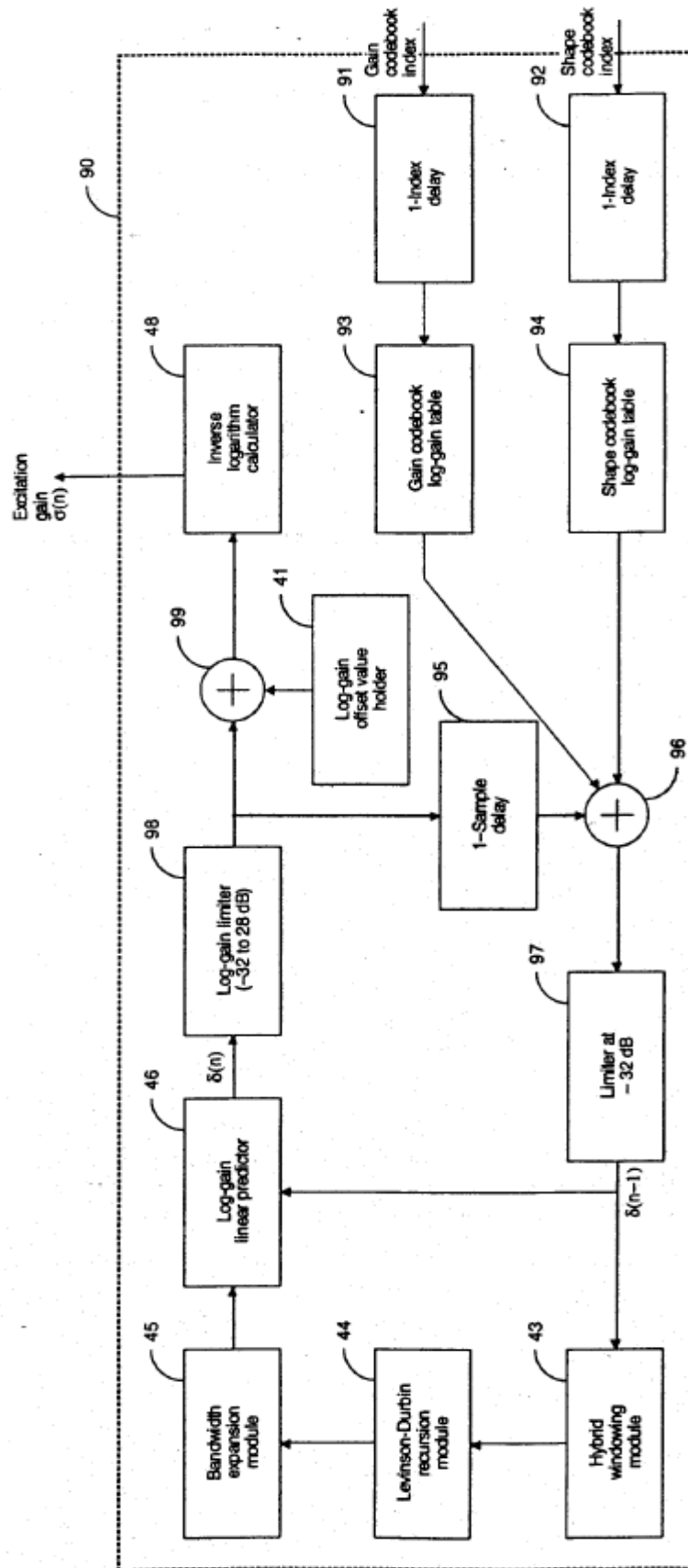
$$\delta(n) = 20 \log \sigma(n) - 32 + 20 \log |g_i| + 10 \log P[y_j]$$

اکنون با استفاده از معادله (۳) داریم:

$$\delta(n) = \hat{\delta}(n) + 20 \log |g_i| + \log P[y_j]$$



برای دریافت فایل Word پروژه به سایت **ویکی پاور** مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه



شکل (۴-۲)

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

روش معادل نشان داده شده در شکل (۴-۲)، ۲ امتیاز مهم بر روش اولیه دارد :

۱- نیاز به محاسبه تابع لگاریتم (بلوک ۴۰ شکل (۴-۱)) که در DSP ها با محاسبه سری های توانی پیاده سازی می شود و سیکل دستور زیادی را برای محاسبه می گیرد، ندارد .

۲- به دلیل حلقه فیدبک طولانی در روش اول (بخاطر تطبیق معکوس ) محاسبات بیشتری در آن انجام می شود و در نتیجه از دقت محاسبات کاسته می شود و بعلاوه در یک پردازنده ممیز ثابت محاسبه تابع لگاریتم همواره با دقت مناسب صورت نمی گیرد .

البته در روش جدید به خاطر محاسبه و ذخیره ۴ بردار بهره و ۱۲۸ بردار شکل به  $128+4=132$  کلمه بیشتر از حافظه ROM احتیاج است که البته مقدار ناچیزی می باشد.

#### ۴-۳-۲- محاسبه لگاریتم معکوس (بلوک ۴۸)

شبه کد ممیزشناور بلوک (۴۸) شکل (۴-۱) بصورت زیر است :

$$GAIN = 10^{(Z/20)}, Z = LOGAIN + GOFF$$

که در آن GOFF ، افسست بهره می باشد. با تبدیل به مبنای ۲ داریم :

$$10^{0.05Z} = 2^{0.05 \log_2(10)Z} = 2^{0.1660964Z}$$

فرض کنید  $X=0.1660964Z$  باشد که در محدوده 0 تا 9.97 تغییر می کند و  $X=[X]+x$  که  $[X]$  جز

صحیح  $X$  و  $x$  بخش کسری آن است. مقدار  $2^{[X]}$  براحتی محاسبه می شود و آنچه که می ماند محاسبه

بخش کسری  $X$  است.

در محاسبه فرض کنید که 0.1660964 در فرمت Q21 نمایش داده شود. این مطابق است با عددی که به

صورت ۱۰ در ۱۶ بیت بالایی و 20649 در ۱۵ بیت پایینی نمایش داده شود. ما هر دو جزء  $Z$  را بطور

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

جداگانه ضرب می کنیم تا به دقت خوبی برای  $X$  برسیم و آنگاه  $[X]$  و  $X$  را جدا می کنیم. در محاسبه نمای جز کسری می دانیم که  $0 < x < 1$  و بنابراین  $1 < 2^x < 2$ . در نتیجه می توان نمایش ثابت زیر را بکار برد:  $x$  بصورت  $Q15$  و  $2^x$ ،  $Q14$  خواهد بود. برای محاسبه  $2^x$  از بسط سری تیلور زیر استفاده می کنیم:

$$2^x = [((C4x + C3)x + C2)x + C1]x + C0 \\ = C4x^4 + C3x^3 + C2x^2 + C1x + C0$$

که در آن ضرایب  $C$  بصورت زیر هستند [10]:

$$C4=323 \quad (Q15)$$

$$C3=1874 \quad (Q15)$$

$$C2=7866 \quad (Q15)$$

$$C1=22702 \quad (Q15)$$

$$C0=16384 \quad (Q14)$$

#### ۴-۴- روند نمای برنامه

با توجه به بلوک دیاگرام طراحی شده برای کدک G.728 (شکل (۳-۱)) و همچنین ملاحظات و تغییرات لازم برای برنامه نویسی ممیز ثابت، برنامه اینکدر به زبان C استاندارد نوشته شده و همراه با برنامه دیکدر در شیه سازی مورد استفاده قرار گرفته و آزمایش شده است. این برنامه ها در ضمیمه (الف) بطور کامل موجود می باشد. در این قسمت ابتدا برخی از متغیرها و بلوک های مهم برنامه را در جدول های (۴-۱) و (۴-۲) معرفی نموده و سپس روندنمای اینکدر و دیکدر را شرح می دهیم.

توصیف	۱-۱-۱-۱-۱ ابعاد (آرایه)	۱-۱-۱-۲ نام متغیر
ضرایب فیلتر سنتز	۱ تا ۵۱	A

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

آرایه موقت ضرایب فیلتر سنتز	۱ تا ۵۱	ATMP
ضرائب مخرج فیلتر وزنی	۱ تا ۱۱	AWP
ضرائب صورت فیلتر وزنی	۱ تا ۱۱	AWZ
آرایه موقت فیلتر وزنی	۱ تا ۱۱	AWZTMP
بردار تحریک	۱ تا ۵	ET
۲ برابر سطوح بهره در کتاب کد	۱ تا ۸	G2
بهره تحریک خطی	۱	GAIN
نقطه وسط سطوح بهره	۱ تا ۷	GB
مربع سطوح بهره	۱ تا ۸	GSQ
ضرائب پیشگویی بهره	۱ تا ۱۱	GP
آرایه موقت ضرایب پیشگویی بهره	۱ تا ۱۱	GPTMP
آرایه موقت بهره	۱ تا ۴	GTMP
بردار پاسخ ضربه $F(z)W(z)$	۱ تا ۵	H
بهترین اندیس کتاب کد برای ارسال	۱	ICHAN
بهترین اندیس بهره کتاب کد	۱	IG
بهترین اندیس شکل کتاب کد	۱	IS
بردار کورلیشن جستجوی کتاب کد	۱ تا ۵	PN
ضرائب اتو کورلیشن	۱ تا ۱۱	R
آرایه موقت ضرائب اتو کورلیشن	۱ تا ۱۱	RTMP



برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

بردار صحبت ورودی	۱ تا ۵	S
بردار صحبت کوانتیزه	۱ تا ۵	ST
حافظه فیلتر سنتز	۱ تا ۵۰	STATELPC
بافر پنجره هیبرید فیلتر وزنی	۱ تا ۲۰	STMP
بافر پنجره هیبرید فیلتر سنتز	۱ تا ۲۰	STTMP
بردار صحبت وزن داده شده شنیداری	۱ تا ۵	SW
بردار هدف VQ	۱ تا ۵	TARGET
آرایه کتاب کد شکل	۱ تا ۶۴۰	Y
انرژی بردارهای کانولوشن شده شکل	۱ تا ۱۲۸	Y2
پاسخ ورودی صفر	۱ تا ۵	ZIR

جدول (۴-۱): متغیرهای برنامه

خروجی (ها)	ورودی (ها)	بلوک
SW	S,AWZ,AWP	۴- فیلتر وزنی شنیداری
ZIR	A,STATELPC,AWZ,AWP	۹ و ۱۰- BLOCKZIR، پاسخ ورودی صفر فیلتر سنتز و فیلتر وزنی شنیداری
TARGET	SW,ZIR	۱۱- محاسبه بردار هدف
H	A,AWZ,AWP	۱۲- محاسبه پاسخ ضربه

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

Y2	H,Y	۱۴ و ۱۵- کانولوشن بردار شکل و محاسبه انرژی
TARGET	TARGET,GAIN	۱۶- نرمالیزه کننده بردار هدف
PN	H,TARGET	۱۳- کانولوشن معکوس زمانی
IG,IS,ICHAN	PN,Y,Y2,GB,G2,GSQ	۱۷ و ۱۸- محاسبه خطا و انتخاب بهترین اندیس کتاب کد
ET	IG,IS,GAIN	۱۹ و ۲۱- کتاب کد تحریک و مقیاس بهره
ST	ET,A,AWZ,AWP,STATEL PC	۹ و ۱۰- تجدید حافظه فیلتر

جدول (۴-۲): بلوک های برنامه

خروجی (ها)	ورودی (ها)	۱-۱-۱-۴- بلوک
R	STMP	۳۶- پنجره هیبرید فیلتر وزنی شنیداری
AWZTMP	R	۳۷- تکرار Durbin فیلتر وزنی شنیداری

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

AWZ,AWP	AWZTMP	۳۸- محاسبه ضرایب وزن داده فیلتر
RTMP	STTMP	۴۹- پنجره هیبرید فیلتر سنتز
ATMP	RTMP	۵۰- تکرار Durbin فیلتر سنتز
A	ATMP	۵۱- گسترش دهنده پهنای باند
R	GTMP	۴۳- پنجره هیبرید بهره
GPTMP	R	۴۴- تکرار Durbin بهره
GP	GPTMP	د - گسترش پهنای باند بهره

جدول (۴-۲): بلوک های برنامه (ادامه)

#### ۴-۴-۱- اینکدر

در شکل (۴-۳) روند نمای کلی برنامه اینکدر ترسیم شده است، در اینجا مراحل اجرای این روند نما و

توصیف کلی بلوکهای استفاده شده در آنرا بیان می کنیم.

در ابتدای برنامه و قبل از وارد شدن به حلقه اصلی برنامه (main-loop)، باید متغیرهای سراسری برنامه

مقدار دهی اولیه شوند، برخی از این متغیرها مانند بردار پاسخ ضربه H باید در ابتدا دارای مقدار

معینی بوده و برخی دیگر مانند ضرایب LPC, A به صفر مقدار دهی اولیه می شود.

## برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

همچنین در ابتدای برنامه فلگ های وضعیت خرابی ILLCOND برای فیلتر سنتز، ILLCONDG برای پیشگویی بهره، ILLCONDP برای پست فیلتر و ILLCONDW برای فیلتر وزنی به FALSE مقدار دهی اولیه می شوند. این فلگ ها در صورتیکه در محاسبه ضرایب هر یک از فیلتر های مربوطه اشکال پدید آید، دارای مقدار TRUE می شوند و در نتیجه ضرایب این فیلتر ها در سیکل معین شده، تجدید نمی شوند. و تا سیکل بعدی از همان ضرایب قبلی استفاده می شود.

سپس متغیر شمارنده برنامه یعنی ICOUNT به صفر مقدار دهی می شود. این شمارنده دارای مقادیر ۱، ۲، ۳، ۴ می باشد و مشخص می کند که در کدام بردار فریم کنونی قرار داریم (هر فریم شامل ۴ بردار است). پس از انجام مقدار دهی های اولیه به حلقه اصلی برنامه اینکدر یعنی main-loop می رسیم.

در ابتدا شمارنده ICOUNT چک می شود که اگر در دور قبل دارای مقدار ۴ بوده است به مقدار ۱ برگردد. سپس یک بردار ۵ نمونه ای از صحبت ورودی از بافر ورودی اینکدر خوانده می شود و به بردار S[1:5] تخصیص داده می شود. در اینجا همه بردار های صحبت ورودی اعم از PCM خطی، u-LAW و یا A-LAW به محدوده [-16384,+16383] تبدیل می شوند. یعنی نمایش Q2 فرمت [-4096,4095.75].

در مرحله بعد اگر ICOUNT=3 باشد و ILLCOND=FALSE باشد بلوک گسترش دهنده پهنای باند ۵۱ اجرا می شود و الا اگر ILLCONDW=FALSE باشد بلوک محاسبه ضرایب فیلتر وزنی ۳۸ اجرا می گردد. سپس بلوک های محاسبه پاسخ ضربه ۱۲ و انرژی کتاب کد ۱۴ و ۱۵ اجرا می شوند.

در مرحله بعدی اگر ICOUNT=2 و همچنین فلگ ILLCONDG=FALSE باشد بلوک گسترش پهنای باند ۴۵ اجرا می شود. در ادامه به شروع پردازش های مربوط به هر بردار می رسیم. ابتدا بلوکهای پیشگویی بهره ۴۶، محدود کننده بهره ۹۸-۹۹ و محاسبه لگاریتم معکوس ۴۸ بطور همزمان و در یک بلوک اجرا

## برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

می شود. سپس blockzir یعنی بلوکهای فیلتر سنتز ۹ و فیلتر وزنی شنیداری ۱۰ برای پاسخ ورودی صفر، اجرا می گردد.

در ادامه به ترتیب بلوکهای فیلتر وزنی شنیداری ۴ و محاسبه بردار هدف ۱۱ و نرمالیزه کننده بردار هدف ۱۶ و کانولوشن معکوس زمانی ۱۳ و جستجوی کتاب کد نرمالیزه ۱۷ و ۱۸ اجرا می گردند. در این مرحله خروجی اینکدر یعنی اندیس کانال ICHAN بدست می آید و در بافر خروجی اینکدر قرار می گیرد.

پس از تحویل اندیس کانال ۱۰ بیتی ICHAN به کانال مخابراتی نوبت به تطبیق معکوس اینکدر می رسد. در ابتدا بلوکهای مقیاس بردار تحریک ۱۹ و ۲۱ و سپس تجدید حافظه فیلترهای سنتز و شنیداری ۹ و ۱۰ اجرا می شود. سپس بلوکهای ۹۶، ۹۴، ۹۳ و ۹۷ برای تجدید بهره اجرا می گردند.

در ادامه برنامه، بردار صحبت کوانتیزه ST[] در بافر STTMP[] ذخیره می گردد. STTMP برای محاسبه ضرایب اتوکورلیشن در فیلتر سنتز بکار می رود. سپس بردار سیگنال S[] در بافر STMP[] ذخیره می گردد تا در محاسبه ضرایب اتوکورلیشن بکار رود. در اینجا پردازش های مربوط به هر بردار پایان می یابد و نوبت به پردازش های مربوط به فریم می رسد.

ابتدا اگر  $ICOUNT=4$  باشد بلوک پنجره هیبرید ۴۹ و تکرار Durbin ۵۰ اجرا می شود.

اگر  $ICOUNT=2$  باشد، بلوک پنجره هیبرید ۳۶ و تکرار Durbin ۳۷ اجرا می گردد.

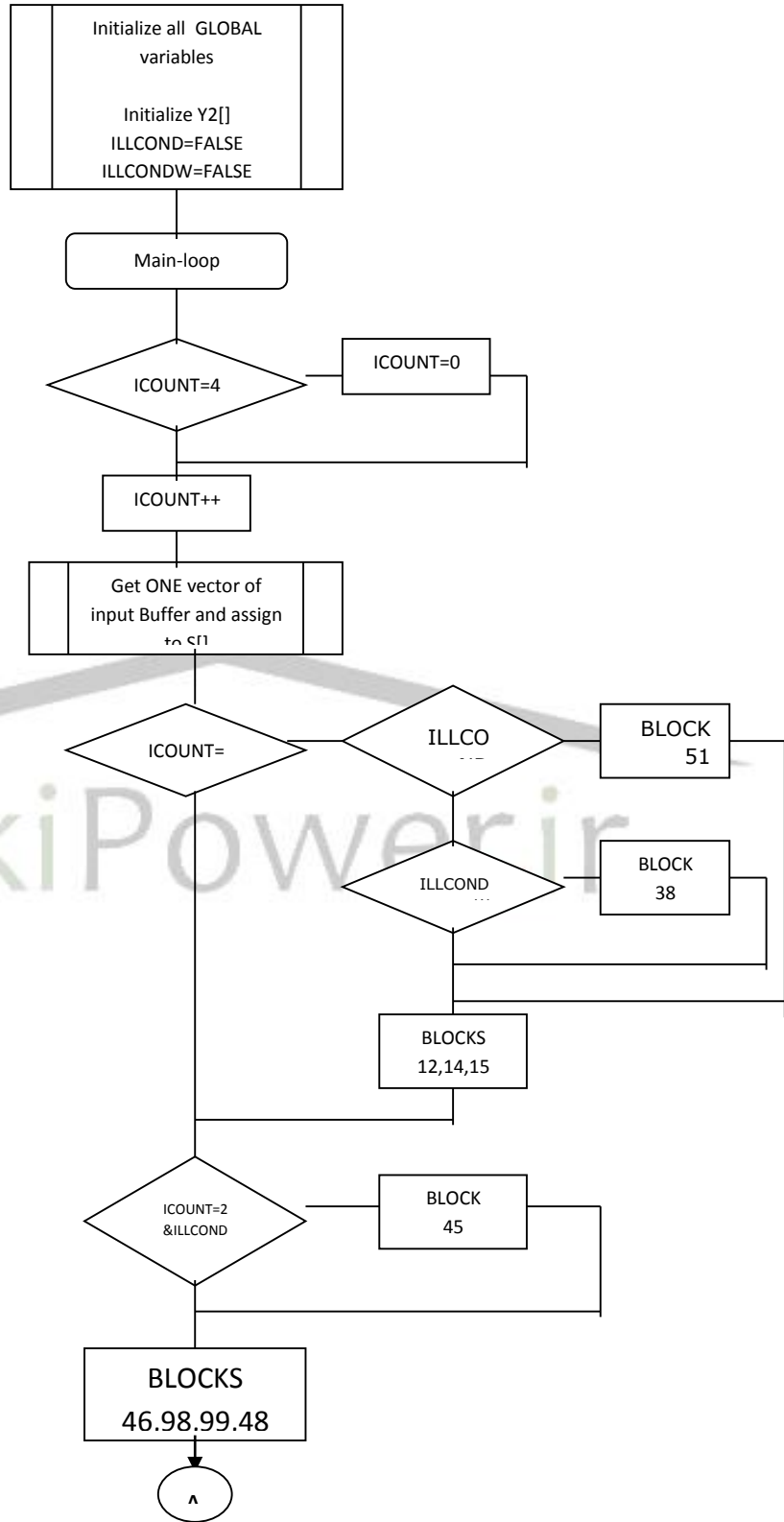
سرانجام اگر  $ICOUNT=1$  باشد ابتدا آرایه موقت پیشگویی بهره، GTMP با حافظه پیشگویی بهره

GSTATE، تجدید شده و سپس بلوکهای پنجره هیبرید ۴۳ و تکرار Durbin ۴۴ اجرا می

شوند. در اینجا پردازش های مربوط به هر فریم هم پایان می یابد و حلقه اصلی برنامه اینکدر کامل می

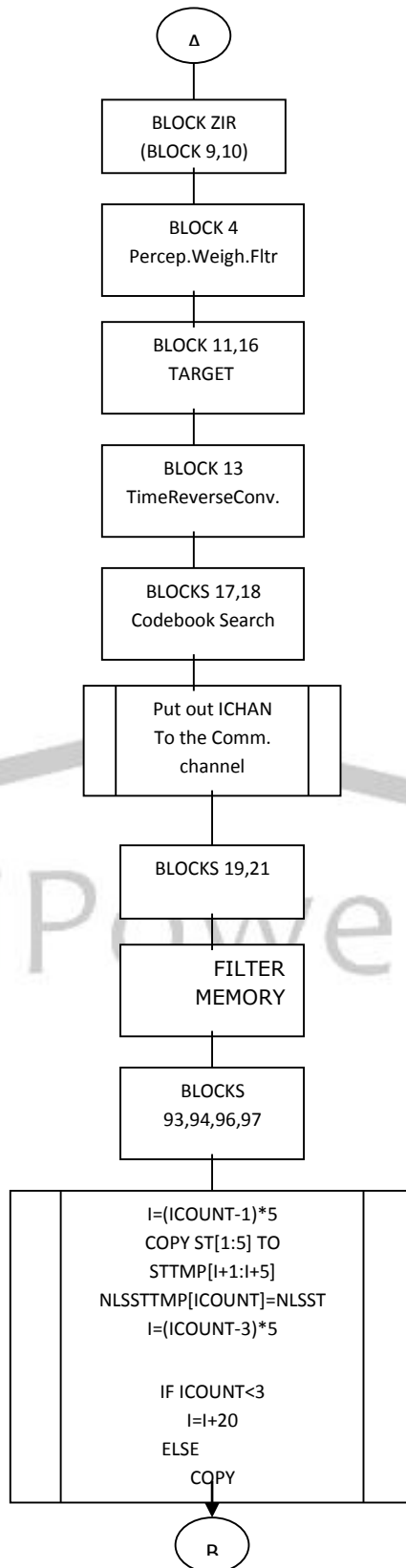
گردد و برنامه به نقطه شروع این حلقه باز می گردد.

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم



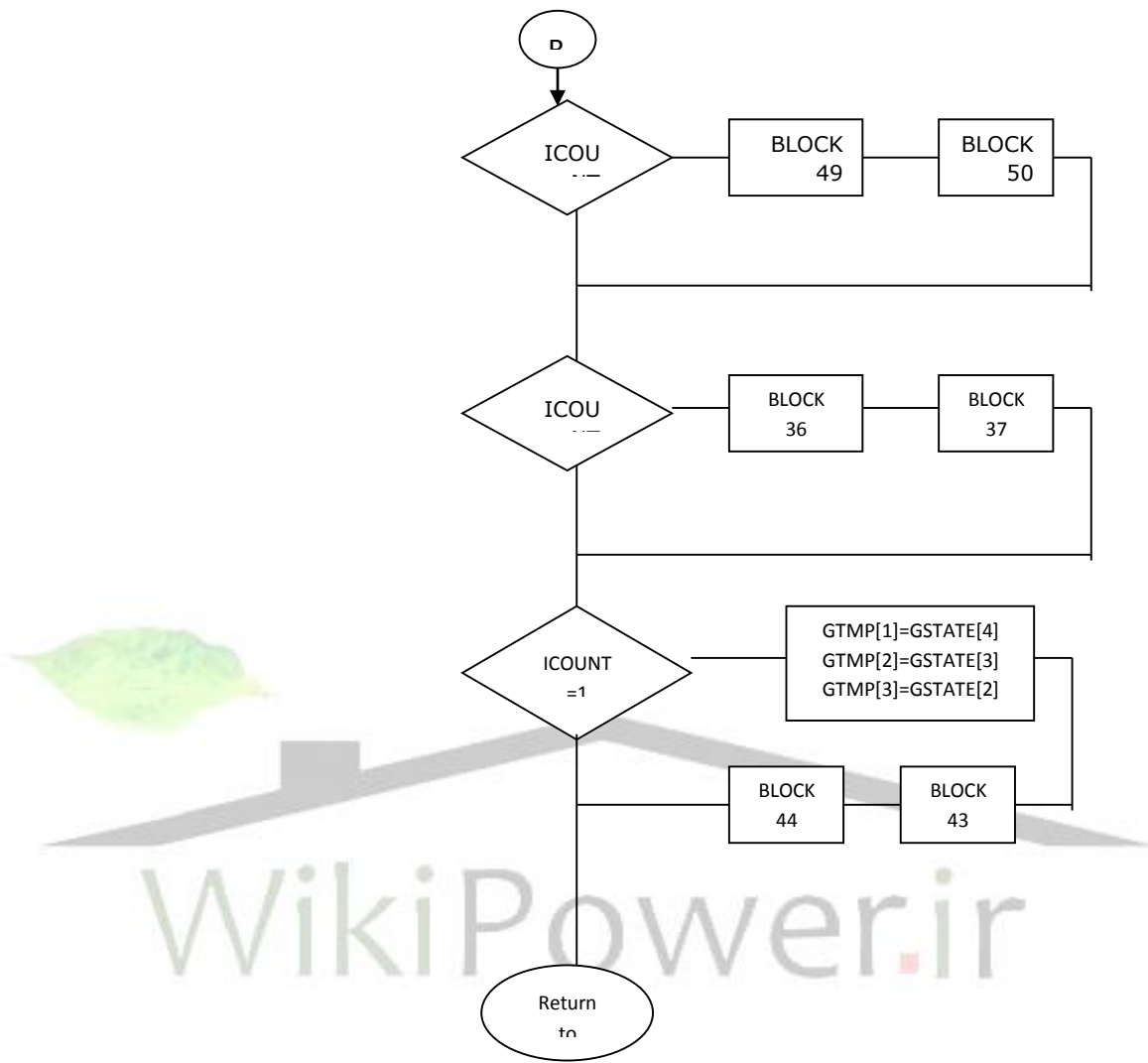
برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

شکل (۴-۳): روندنمای اینکدر



شکل (۴-۳): روندنمای اینکدر (ادامه)

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم



شکل (۴-۳): روندنمای اینکدر (ادامه)



## برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

شکل (۴-۴) روندنمای کلی حاکم بر برنامه دیکدر را نشان می دهد. در این قسمت مراحل اجرای این روندنما را بیان می کنیم.

در ابتدای برنامه، متغیرهای سراسری مقدار دهی اولیه می شوند. همچنین فلگ های وضعیت خرابی فیلتر سنتز ILLCOND، بهره ILLCONDG و پست فیلتر ILLCONDP به FALSE مقدار دهی اولیه می گردند. و سر انجام شمارشگر برنامه ICOUNT به صفر مقدار دهی می شود.

پس از مقدار دهی اولیه برنامه وارد حلقه اصلی دیکدر (main-loop) می شویم، ابتدا شمارشگر برنامه که نشان می دهد هم اکنون در کدام بردار از فریم کنونی هستیم چک می شود که اگر در دور قبل دارای مقدار ۴ بوده به مقدار ۱ برگردد. سپس اندیس کانال ICHAN از بافر ورودی دیکدر خوانده می شود. آنگاه اندیس شکل IS و اندیس بهره IG از ICHAN بدست می آید. حال نوبت به تجدید ضرایب

فیلترها می رسد. اگر ICOUNT=3 باشد و همچنین فلگ خرابی ILLCOND برابر FALSE باشد بلوک گسترش پهنای باندها ۵۱ اجرا می گردد. در مرحله بعد اگر ICOUNT=2 باشد و فلگ ILLCONDG هم برابر FALSE باشد، بلوک گسترش باندها ۴۵ اجرا می شود.

پس از آن پردازش های مربوط به یک بردار شروع می شود. ابتدا بلوک های ۴۶، ۴۸، ۴۹ و ۴۸ مربوط به پیشگویی بهره انجام می شود. سپس بلوکهای ۱۹ و ۲۱ مربوط به بردار تحریک انتخاب شده و بعد از آن بلوک فیلتر سنتز ۳۲ اجرا می شوند.

اگر ICOUNT=1 باشد ضرائب پست فیلتر زمان کوتاه در بلوک ۸۵ تجدید می شود. سپس فیلتر

معکوس LPC مرتبه ۱۰ در بلوک ۸۱ اجرا می شود آنگاه اگر ICOUNT=3 باشد بلوک های استخراج پریود pitch ۸۲، محاسبه ضریب پریود pitch ۸۳ و تجدید ضرائب پست فیلتر زمان بلند ۸۴ اجرا می شوند. سپس بلوکهای پست فیلتر زمان بلند ۷۱ و پست فیلتر زمان کوتاه ۷۲ و محاسبه مجموع قدر مطلق

## برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

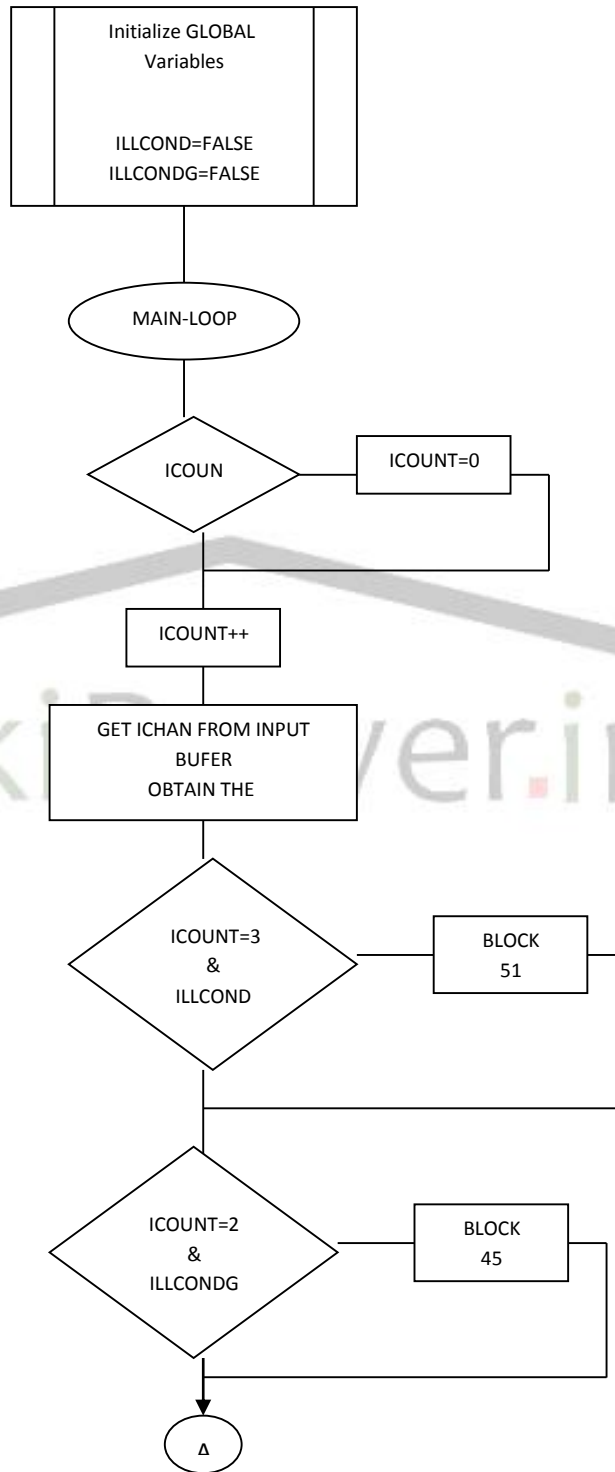
های ۷۴ و ۷۳ و همچنین بلوکهای محاسبه ضریب مقیاس ۷۵ و فیلتر پایین گذر ۷۶ و کنترل بهره خروجی پست فیلتر ۷۷ اجرا می گردند.

پس از آن پیشگویی بهره و حافظه آن در بلوک های ۹۳ و ۹۴ و ۹۶ و ۹۷ تجدید می شود. در مرحله بعد بافر صحبت کوانتیزه و سنتز شده ST یعنی STTMP تجدید می گردد. و در اینجا پردازش های مربوط به یک بردار پایان می یابد و پردازش های مربوط به هر فریم آغاز می شود.

ابتدا اگر  $ICOUNT=4$  باشد بلوک پنجره هیبرید فیلتر سنتز ۴۹ و پس از آن بلوک تکرار Durbin ۵۰ از مرتبه ۱ تا مرتبه ۱۰ اجرا می شود. سپس ضرائب پیشگویی مرتبه ۱۰ در APF ذخیره می شوند تا پست فیلتر بعداً از آنها استفاده نماید. پس از آن عملیات بلوک ۵۰ از مرتبه ۱۱ تا مرتبه ۵۰ ادامه می یابد.

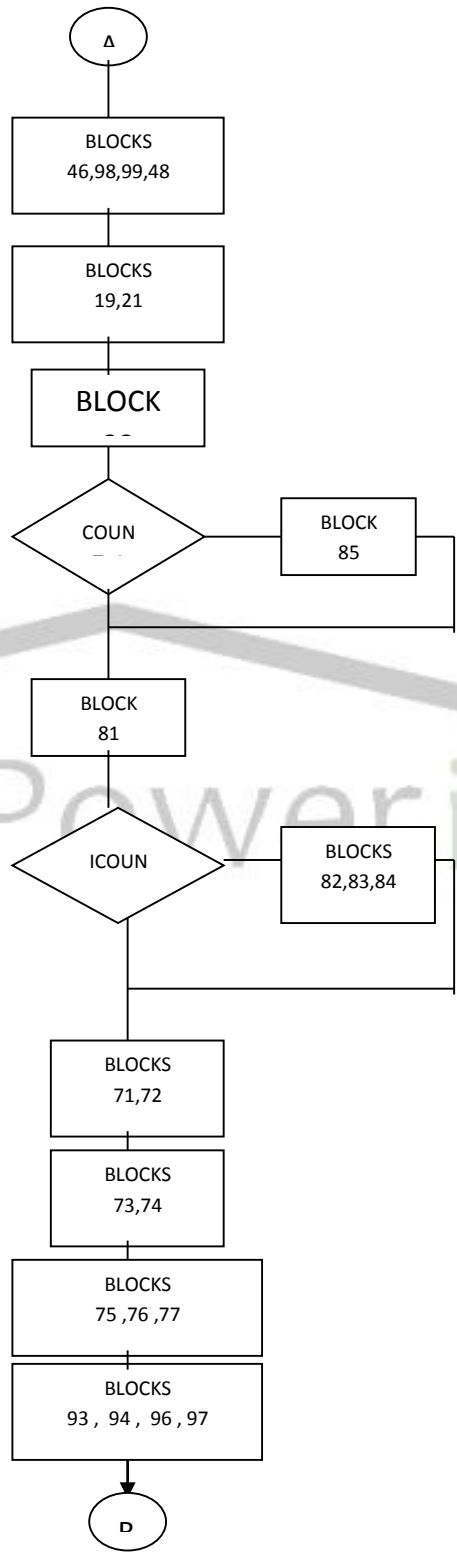
در پایان اگر  $ICOUNT=1$  باشد بافر بهره GTMP بوسیله حافظه GSTATE در یک مرحله تجدید می شود و بلوک های پنجره هیبرید بهره ۴۳ و تکرار Durbin بهره ۴۴ اجرا می گردند. در اینجا پردازش های مربوط به هر فریم در دیکدر پایان می یابد و برنامه به حلقه اصلی (main-loop) باز می گردد.

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم



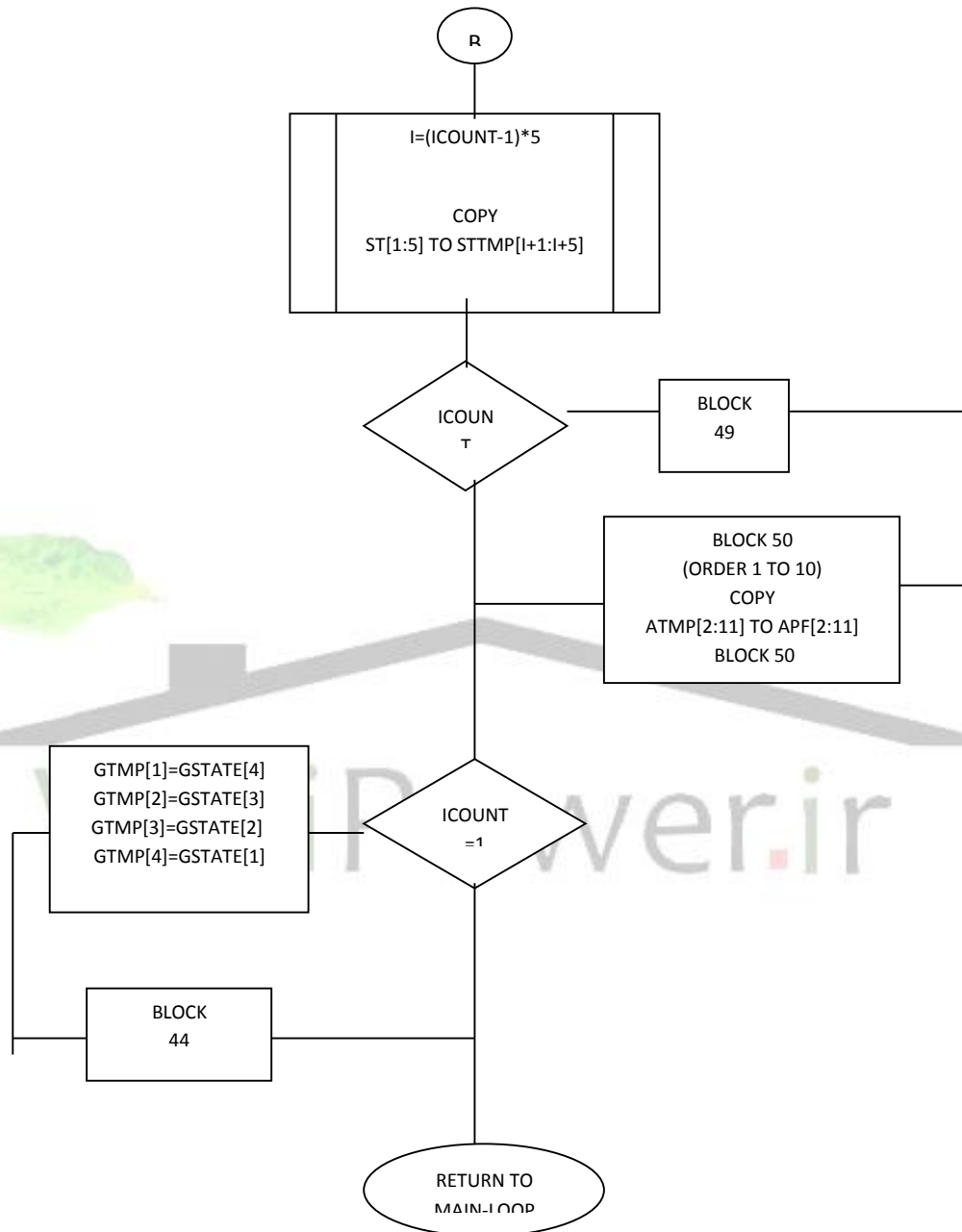
برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

شکل (۴-۴): روندنمای دیکدر



برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

شکل (۴-۴): روندنمای دیکدر (ادامه)



شکل (۴-۴): روندنمای دیکدر (ادامه)

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

## فصل ۵

### پیاده سازی الگوریتم بر روی DSP

WikiPower.ir

۵-۱- مقدمه

پس از شبیه سازی الگوریتم کدک G.728 به زبان C ، نوبت به انتخاب DSP و پیاده سازی برنامه بر روی آن می رسد . در این فصل ابتدا مروری بر نحوه پیاده سازی بلادرنگ داریم . سپس به معرفی DSP های سری TMS می پردازیم. در قسمت بعدی نحوه برنامه نویسی TMS و تبدیل کد برنامه از زبان C به زبان اسمبلی TMS و بهینه سازی آن را به منظور پیاده سازی بی درنگ شرح می دهیم .

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

## ۵-۲- مروری بر پیاده سازی بلادرنگ

یک پروسه بلادرنگ، فرآیندی است که باید در زمان مشخص انجام پذیرد. این حد زمانی ممکن است از مرتبه 200us تا حدود 20 ms برسد. در کدینگ صحبت با فرض نرخ نمونه برداری 8 khz، برای کدری که بصورت نمونه به نمونه کار می کند، پردازش بلادرنگ باید در زمان 0.125 ms انجام شود. اما در کدهای جدید مانند کدک CELP، فرآیند کدینگ، بصورت بلوکی و با طول بلوک اصلی در حدود 20-30 ms (فریم اصلی) و طول بلوک فرعی 4-8 ms (فریم فرعی) صورت می گیرد که باعث افزایش محدودیت زمانی مجاز می شود.

به موازات پیشرفت در الگوریتم های کدینگ صحبت، تکنولوژی پردازنده های DSP هم در چند سال اخیر پیشرفت زیادی داشته و با استفاده از DSP های قدرتمند ممیز ثابت<sup>۱</sup> و ممیز شناور<sup>۲</sup>، امکان پیاده سازی بلادرنگ الگوریتم های صحبت بسیار پیچیده همچون CELP فراهم شده است DSP های ممیز ثابت ارزاتر هستند ولی از نظر برنامه نویسی مشکل تری با شند. هرچند که از نظر قیمت، استفاده از یک DSP ممیز ثابت بهتر به نظر می رسد اما طراحی یک نسخه ممیز ثابت از یک الگوریتم کدینگ

<sup>۱</sup> Fixed Point

<sup>۲</sup> Floating Point

## برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

صحبت که ممیز شناور است کاری دشوار و گاهی غیر ممکن می باشد زیرا باید اثرات مقیاس کردن ، نرمالیزه کردن و سرریز شدن را در نظر گرفت.

دو عامل مهم در انتخاب DSP جهت پیاده سازی الگوریتم کدینگ صحبت عبارتند از سیکل زمانی دستور العمل DSP (سرعت محاسبه) و مناسب بودن دستورالعملهای آن برای پردازش بلوک های اصلی آن الگوریتم . بعنوان مثال در الگوریتم کدینگ CELP بیشتر زمان پردازش DSP با دستوراتی به شکل MAC (ضرب و انباشت) گرفته می شود. برای چنین الگوریتمی DSP هایی که این کار را در یک سیکل دستورالعمل انجام دهند مناسب خواهند بود.

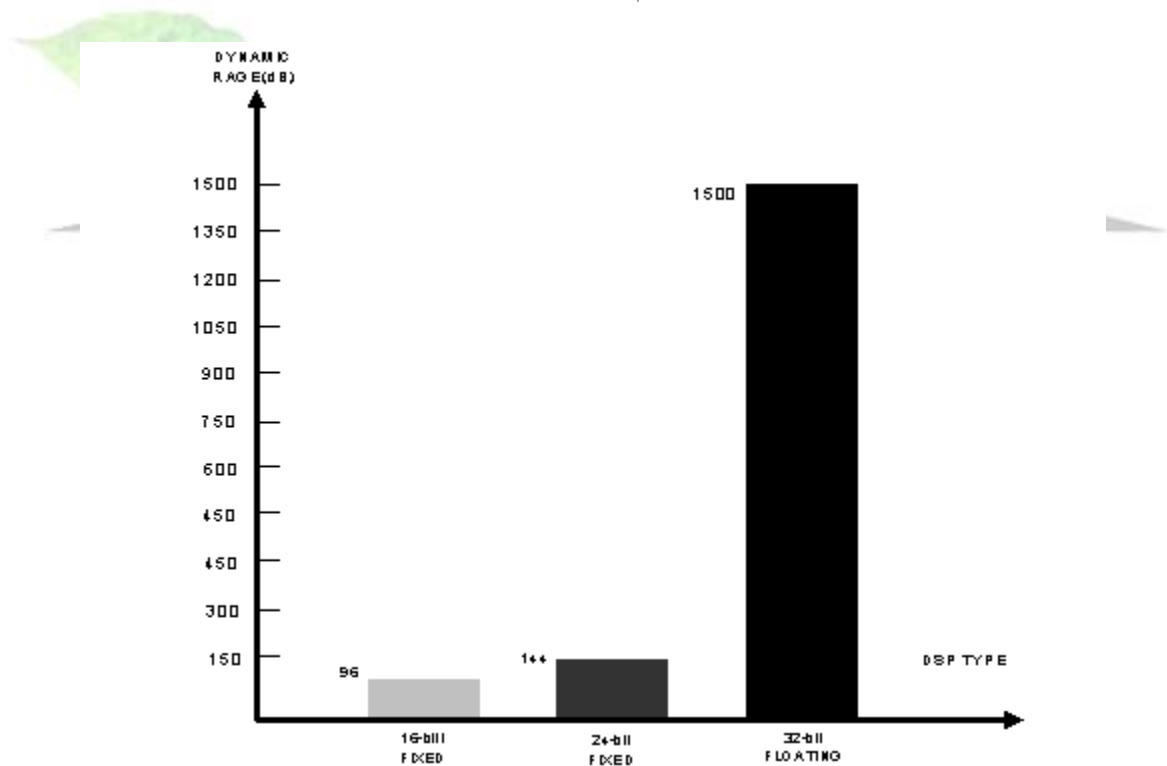
### ۳-۵- چپ های DSP

DSP های برنامه پذیر با توجه به دقت و عملیات حسابی به دو گروه ممیز ثابت و ممیز شناور دسته بندی می شوند . DSP های ممیز ثابت سریعتر و ارزانتر هستند ولی از نظر برنامه نویسی مشکل تر بوده و دقت پایین تری را فراهم می سازند . امتیاز های پیاده سازی ممیز شناور بر پیاده سازی ممیز ثابت زیاد است . دقت در اعداد ممیز شناور بخاطر نرمالیزه شدن اتوماتیک ماتیس بوسیله پردازنده در طول برنامه ثابت می ماند . در حالیکه دقت داده های ممیز ثابت، بسته به اندازه ذخیره سازی آن داده پس از عملیات روی آن، تغییر می کند . بخاطر نرمالیزه شدن اتوماتیک ماتیس در DSP های ممیز شناور ، رند کردن و بریدن اعداد ، خطای کلی کمتری نسبت به ممیز ثابت بوجود می آورد . همچنین دقت ثابت بهمراه توانایی نمایش اعداد خیلی بزرگ یا خیلی کوچک ، باعث می شود مشکلاتی که در پیاده سازی فیلترها از نظر فاصله مکان صفرها و قطب ها وجود دارد ، برطرف گردد.



## برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

همانطور که در شکل (۵-۱) دیده می شود ، رنج دینامیکی وسیع در DSP های ممیز شناور اجازه می دهد که اعداد خیلی کوچک و خیلی بزرگ با دقت بالا نمایش داده شوند [5]. این بخصوص در محاسبات میانی FFT ها و فیلتر های بازگشتی مرتبه بالا لازم می شود. همچنین رنج دینامیکی وسیع موجب حذف نیاز مقیاس کردن میانی در DSP های ممیز ثابت ، جهت جلوگیری از سرریز شدن داده، می گردد. از این رو استفاده از DSP های ممیز شناور ، باعث کاهش اندازه و پیچیدگی برنامه می شود بعلاوه اغلب الگوریتم های DSP ابتدا بر روی کامپیوتر های شخصی و با استفاده از امکانات ممیز شناور شبیه سازی می شوند و در صورت استفاده از DSP های ممیز شناور نیاز به شبیه سازی مجدد ممیز ثابت این الگوریتم ها بر طرف می گردد .



شکل ( ۵-۱) . رنج دینامیک در DSP های ممیز ثابت و ممیز شناور

## برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

در DSP های ممیز ثابت باید توجه خاصی به سرریز شدن و مقیاس کردن داده ها نمود. از طریق چک کردن اعداد و مقیاس کردن مناسب آنها می توان به دقت یکنواختی در تمام پیاده سازی دست یافت، کاهش دقت اغلب در هنگام ضرب دو عدد رخ می دهد چرا که تعداد بیت لازم برای نمایش حاصل ضرب برابر مجموع بیت های آن دو عدد می باشد. این بیت های اضافی باعث از دست رفتن اطلاعات می شود و بایستی حاصل ضرب را با دقت مضاعف ذخیره نمود. در برخی از DSP ها سرریز شدن (overflow) بسادگی با قرار دادن نتیجه برابر با بزرگترین عدد مثبت یا منفی قابل نمایش در DSP، کنترل می شود. ولی معمولا برای اینکه کنترل بهتری روی الگوریتم و کاهش خطای آن داشته باشیم، بایستی قبل از جمع کردن، حاصل ضرب را به اندازه مناسب به سمت راست شیفت دهیم تا از سرریز شدن آکومولاتور جلوگیری شود.

### ۵-۳-۳- مروری بر DSP های خانواده TMS320

شرکت TI بعنوان پیشرو در بازار DSP، اولین چیپ DSP را در سال ۱۹۸۲ معرفی نمود. خانواده TMS320 در حال حاضر به سه گروه اصلی C2000, C5000, C6000 تقسیم بندی می شوند [16].

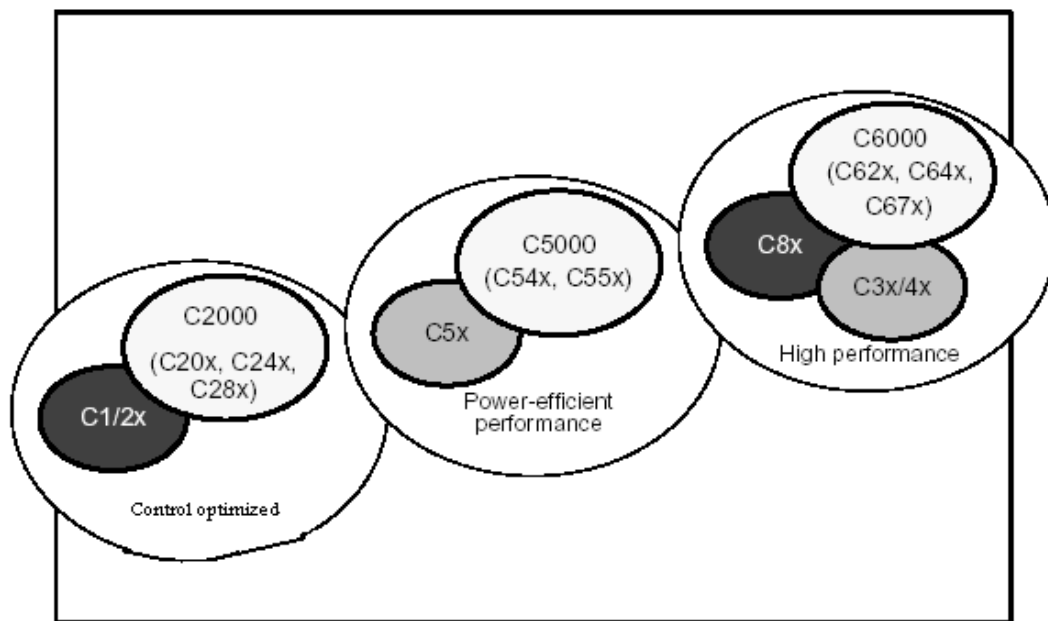
گروه TMS320C6000 از نظر عملکرد و سرعت و راحتی استفاده برای برنامه نویسی سطح بالا بهینه شده اند. این گروه به نسل های ممیز ثابت C64X, C62X و ممیز شناور C67X تقسیم می شوند.

گروه TMS320C5000 از نظر عملکرد و مخصوصا توان مصرفی بهینه شده اند و برای کاربردهای موبایل و اینترنت که مشکل باتری دارند بسیار مناسب هستند. این گروه شامل دو نسل ممیز ثابت C55X, C54X می باشد.

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

گروه TMS320C 2000 برای کاربردهای کنترل دیجیتال در نظر گرفته شده و شامل دو نسل C28X, C24X می باشند.

در شکل زیر نمودار های خانواده TMS320 نشان داده شده است.



شکل (۵-۲): DSP های خانواده TMS320

### ۵-۳-۳-۱- معرفی سری TMS320C54X

DSP های سری C54X دارای سرعت و قابلیت عملیاتی بالایی هستند و از یک معماری باس پیشرفته، CPU با سخت افزار مناسب کاربردهای خاص، حافظه و ادوات جانبی on-chip و دستورالعمل های ویژه استفاده می کنند.

#### ویژگی های C54x:

- معماری پیشرفته چند باسه با یک باس برنامه، ۳ باس داده و ۴ باس آدرس

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

- واحد محاسبه و منطق (ALU) ۴۰ بیتی و دو آکومولاتور ۴۰ بیتی جداگانه
  - ضرب کننده موازی برای عملیات ضرب/ انباشت (MAC) تک سیکلی بدون pipeline
  - فضای حافظه آدرس پذیر ۱۶ بیتی ( 64 kword Data , 64 kword Program )
- 64 kword I/O

در جدول (۱-۵) مشخصات حافظه DSP های C54x ( به kword ) نشان داده شده است [12].

Device	Program ROM	Program/Data ROM	DARAM†	SARAM‡
C541	20	8	5	0
C542	2	0	10	0
C543	2	0	10	0
C545	32	16	6	0
C546	32	16	6	0
C548	2	0	8	24
C549	16	16	8	24
C5402	4	4	16	0
C5410	16	0	8	56
C5420	0	0	32	168

† Dual-access RAM

‡ Single-access RAM

جدول (۱-۵)

کاربردهای زیر را می توان برای سری C54X در نظر گرفت:

- کدینک و دیکدینگ صحبت
- حذف اکو - حذف نویز
- مدولاسیون و دمدولاسیون
- فشردن سازی تصویر و صدا

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

- رمز نگاری صحبت

- تشخیص صحبت و باز سازی صحبت

#### ۵-۴- توسعه برنامه بلادرنگ

پیاده سازی یک نرم افزار بصورت بلادرنگ بر روی سخت افزار از اهمیت بالایی در سیستم های مخابراتی برخوردار است. در پیاده سازی بلا درنگ هزینه تجهیزات همانند پارامترهای کیفی سیستم اهمیت زیادی داشته و بایستی سعی شود که یک الگوریتم به روش بهینه پیاده سازی گردد.

در عمل پیاده سازی بلادرنگ یک الگوریتم بر روی چیپ DSP شامل مراحل زیر می شود: الف) بهبود تئوری الگوریتم. ب) آزمایش الگوریتم بوسیله شبیه سازی کامپیوتری با استفاده از یک زبان سطح بالا مانند C. ج) تبدیل کد سطح بالا به کد اسمبلی DSP مربوطه. د) آزمایش کد بلادرنگ با استفاده از ابزارهای توسعه گر موجود (مانند شبیه سازها) و ه) طراحی سخت افزار مناسب.

در قسمت ج) برای تبدیل کد سطح بالا به کد DSP سه روش وجود دارد. روش اول، استفاده از کراس کمپایلر DSP است که زبان سطح بالای C را به اسمبلی DSP ترجمه می کند. روش دوم، برنامه نویسی دستی و مستقیم الگوریتم با استفاده از مجموعه دستورالعمل های DSP می باشد و روش سوم که عبارتست از استفاده از کراس کمپایلر و همچنین برنامه نویسی دستی در قسمتهایی که از نظر زمان اجرا محدودیت وجود دارد. از آنجا که برنامه نویسی دستی کاری مشکل و وقت گیر می باشد و مخصوصاً وقتی که اندازه برنامه بزرگ باشد آزمایش و اشکال زدایی آن وقت زیادی صرف می کند، معمولاً از روش سوم در برنامه های بزرگ بیشتر استفاده می شود.

## برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

در قسمت (د) نتایج شبیه سازی کامپیوتری با خروجی های معادل DSP مقایسه می شوند. این کار معمولاً برای داده های آزمایش محدودی انجام می شود زیرا که پردازش مقدار زیادی داده در شبیه ساز DSP کار وقت گیر و دشواری می باشد. پس از مقایسه و درستی خروجی DSP و بررسی اجرای نرم افزار بصورت بلادرنگ آنگاه می توان از درستی پیاده سازی بلادرنگ الگوریتم اطمینان حاصل کرد.

### ۵-۵- اجرای برنامه روی برد توسعه گر C5402 DSK

پس از آشنایی با پردازنده های TI، حال نوبت به معرفی برد توسعه گر C5402 DSK و برنامه نویسی آن می رسد. DSK امکان آزمایش و توسعه برنامه ها را بر روی پردازنده C5402 فراهم می سازد و می تواند مرجع خوبی برای طراحی سخت افزاری سیستم باشد [15].

برد DSK شامل اجزای زیر است:

- 100 MHz VC5402 DSP
- 64 kword حافظه خارجی SRAM
- رابط صوتی میکروفن / بلندگو
- رابط تلفن
- رابط داده ناهمزمان RS-232
- رابط JTAG برای امولیشن و رابط Host

در شکل (۵-۳) بلوک دیاگرام و ارتباط های DSK نشان داده شده است.

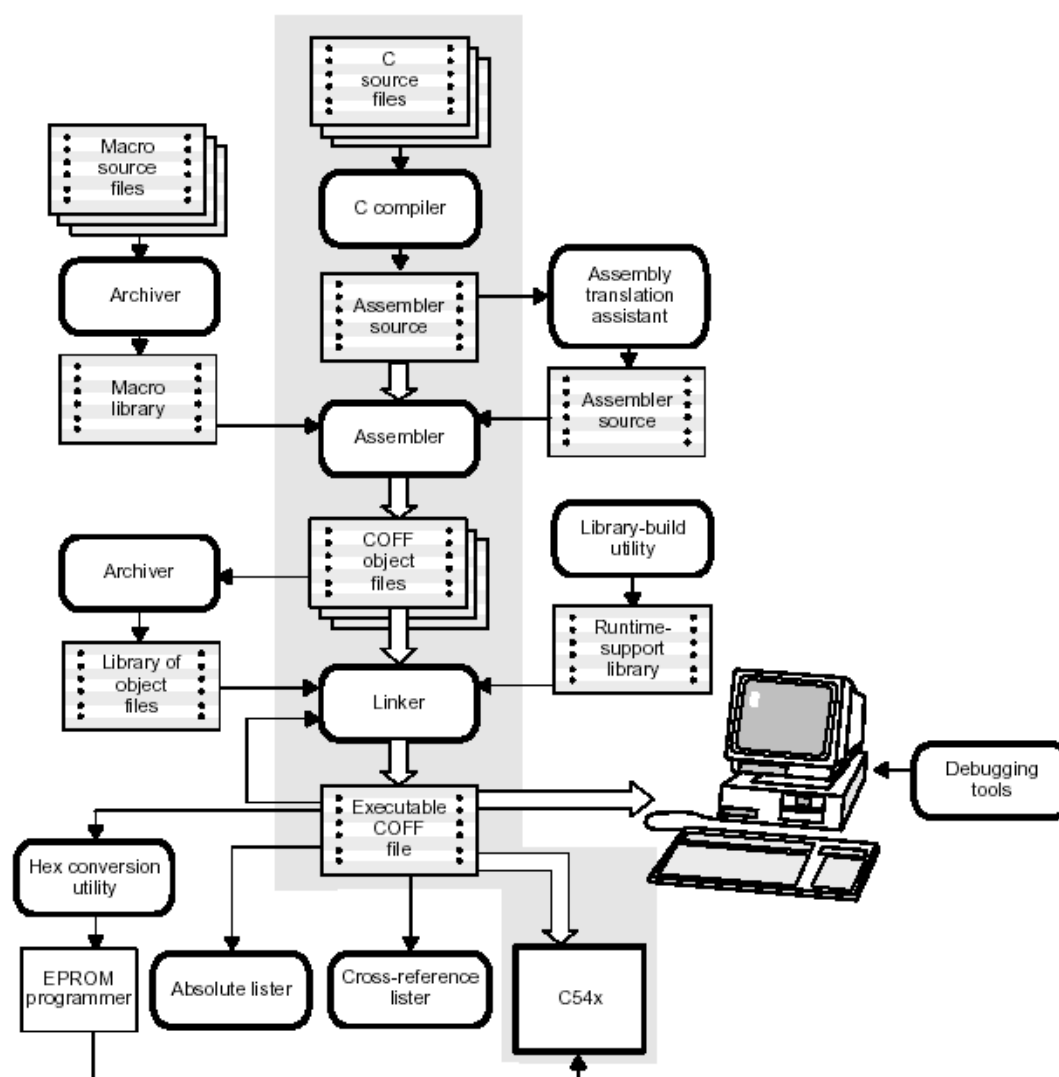


برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آر سایت و به همراه فونت های لازم

### ۵-۵-۱- بکارگیری ابزارهای توسعه نرم افزار

شکل (۴-۵) دیاگرام توسعه نرم افزاری C54x را نشان می دهد. بخش سایه زده مسیری را مشخص می

کند که بیشتر استفاده می شود و بقیه بخش ها انتخابی هستند [13,14].



شکل (۴-۵) : دیاگرام توسعه نرم افزاری TMS320C54x

در این قسمت برخی از ابزارهای نشان داده شده در شکل فوق را بررسی می کنیم :



برای دریافت فایل Word پروژه به سایت **ویکی پاور** مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

- C/C++ Compiler ، کد C/C++ را به کد زبان اسمبلی ترجمه می کند .
- Assembler ، فایل های اسمبلی را به زبان ماشین بصورت یک فایل COFF Object تبدیل می نماید .
- Linker ، Object فایل های تولید شده توسط اسمبلر را بصورت یک ماژول COFF Object قابل اجرا تبدیل می نماید .
- مبدل Hex ، DSP های C54x می توانند فایل COFF را بعنوان ورودی قبول کنند ولی اغلب برنامه ریز های EPROM قادر به این کار نیستند ، از این رو باید فایل COFF به یکی از فرمت های Intel , Motorola , Tektronix , TI-tagged تبدیل شود تا در برنامه ریز EPROM مربوطه ریخته شود .



#### ۵-۵-۲- استفاده از نرم افزار CCS (Code Composer Studio)

نرم افزار CCS محصول شرکت TI است و در قالب یک محیط ویژوال مانند Visual C++ ، امکان برنامه نویسی C ، اسمبلی و تمامی ابزارهای توسعه نرم افزاری شرح داده شده در قسمت قبل را فراهم می آورد. این نرم افزار که به همراه برد DSK می باشد ، از طریق پورت موازی کامپیوتر به برد DSK متصل شده و کار کنترل و مونیتور کردن آن را انجام می دهد .

همه رجیسترها و فضای حافظه و پورت های برد DSK ، توسط این نرم افزار کنترل و برنامه ریزی می شوند و پس از تولید کد قابل اجرا بر روی TMS ، نرم افزار قادر است پردازنده C5402 را برنامه ریزی

## برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

کند. همچنین امکانات اشکال زدایی مانند اجرای برنامه بصورت های مختلف و مشاهده متغیرها توسط CCS فراهم شده است .

بمنظور استفاده از CCS و اجرای برنامه بر روی DSK ابتدا برنامه شبیه سازی ممیز ثابت کدر G.728 را در قالب یک پروژه بنام CODEC به محیط CCS منتقل کردیم. آنگاه پروژه را بدون استفاده از optimizer ، کمپایل و لینک نموده و پس از رفع اشکالات برنامه و ساختن فایل codec.out ، آن را بر روی DSK اجرا کردیم . هدف ما در این مرحله اطمینان از صحت اجرای برنامه و اندازه گیری MIPS مورد نیاز آن بود . از اینرو بخشی از یک فایل صحبت PCM ۱۶ بیتی ( در حدود ۲ ثانیه، 16 kword ) را در بخشی از فضای حافظه بنام input\_ ذخیره نموده و سپس اینکدر و دیکدر را در یک حلقه قرار دادیم . اینکدر نمونه های ورودی را می خواند و بصورت کد شده به دیکدر می داد و دیکدر هم آنها را باز سازی کرده و در بخشی از حافظه بنام output\_ ذخیره می ساخت .

با استفاده از امکان Profiler در نرم افزار CCS می توان از عملکرد هر یک از توابع موجود در برنامه ، MIPS و مقدار حافظه مورد نیاز آنها آگاه شد . شکل (۵-۷) خروجی Profiler را برای ۶۵ دور اجرا (۳۲۵ نمونه) نشان می دهد. در ستون اول نام توابع برنامه را می توان دید که البته بدلیل محدودیت نمایش همه توابع دیده نمی شوند. ستون دوم اندازه کد هر تابع به word و ستون سوم تعداد اجرای آنرا تا کنون نشان می دهد. در ستون ۴ کل سیکل دستورالعمل و ستون ۵ ماکزیمم سیکل دستورالعمل لازم برای اجرای هر تابع را نشان می دهد. در ستون ۶ و ۷ هم به ترتیب مینیمم و میانگین سیکل دستور هر تابع دیده می شود.

برای اجرای برنامه اینکدر و دیکدر بصورت دوطرفه کامل لازم است که عملیات پردازش هر دور برنامه در زمان ۵ نمونه یعنی 0.625 ms به پایان برسد و این در واقع برابر تاخیر الگوریتمی کدک کم تاخیر

## برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

G.728 می باشد. با توجه به قدرت پردازنده C5402 که برابر 100 MIPS است لازم است هر دور اجرای

برنامه از ۶۲۵۰۰ سیکل دستورالعمل بیشتر نباشد. برای محاسبه تعداد سیکل دستورالعمل هر دور برنامه ،

در نقطه ای که برنامه اینکدر و دیکدر به پایان می رسد یک Break Point قرار می دهیم و

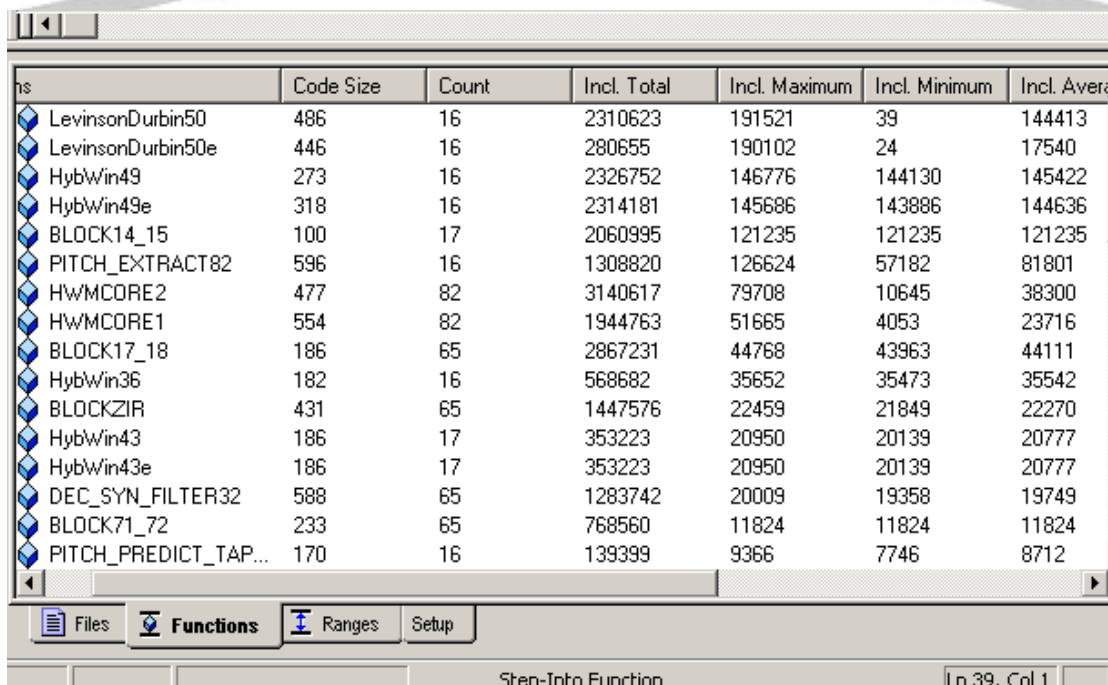
با استفاده از دستور RUN برنامه را تا این نقطه اجرا می کنیم. خروجی CLOCK پردازنده در این حالت

تعداد سیکل دستورالعمل یک دور برنامه را نشان می دهد.

در این حالت چون از optimizer استفاده نشده ، مشاهده می شود که MIPS کلی برنامه چندین برابر

قدرت C5402 (100 MIPS) است و حتی در حالت ماکزیمم به ۴۷۸۹۹۱ برای هر دور اجرا می رسد در

صورتی که با توجه به قدرت C5402 ، نباید از ۶۲۵۰۰ در هر دور فراتر رود .



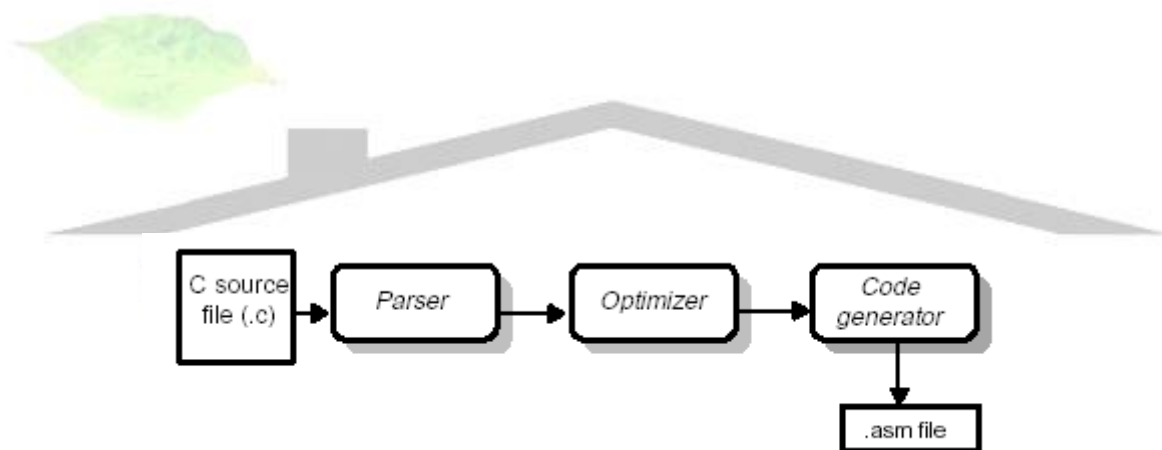
Name	Code Size	Count	Incl. Total	Incl. Maximum	Incl. Minimum	Incl. Average
LevinsonDurbin50	486	16	2310623	191521	39	144413
LevinsonDurbin50e	446	16	280655	190102	24	17540
HybWin49	273	16	2326752	146776	144130	145422
HybWin49e	318	16	2314181	145686	143886	144636
BLOCK14_15	100	17	2060995	121235	121235	121235
PITCH_EXTRACT82	596	16	1308820	126624	57182	81801
HWMCORE2	477	82	3140617	79708	10645	38300
HWMCORE1	554	82	1944763	51665	4053	23716
BLOCK17_18	186	65	2867231	44768	43963	44111
HybWin36	182	16	568682	35652	35473	35542
BLOCKZIR	431	65	1447576	22459	21849	22270
HybWin43	186	17	353223	20950	20139	20777
HybWin43e	186	17	353223	20950	20139	20777
DEC_SYN_FILTER32	588	65	1283742	20009	19358	19749
BLOCK71_72	233	65	768560	11824	11824	11824
PITCH_PREDICT_TAP...	170	16	139399	9366	7746	8712

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

شکل (۷-۵): خروجی Profiler نرم افزار CCS

### - استفاده از Optimizer

بهینه ساز باعث بهبود در سرعت اجرای برنامه و حجم کد تولید شده در کمپایلر می شود. این کار با ساده سازی حلقه ها، مرتب کردن عبارت ها و قراردادن متغیرها در رجیسترها ممکن می شود. کمپایلر C54x قادر به انجام بهینه سازی های مختلفی است و بهینه سازی سطح بالا در optimizer و بهینه سازی سطح پایین در code generator انجام می شود. شکل (۸-۵) جایگاه optimizer را بهتر نشان می دهد.



شکل (۸-۵): روند کمپایل یک فایل

ساده ترین راه استفاده از optimizer این است که در خط فرمان در دستور cl500 از انتخاب -on

استفاده شود که n (0,1,2 & 3)، نشان دهنده سطح بهینه سازی بصورت زیر است:

00-: ساده سازی روندنما

- قرار دادن متغیرها در رجیسترها

- حذف کدهای استفاده نشده

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

- ساده سازی جمله ها و عبارت ها

01- : همه بهینه سازی های 00- بعلاوه :

- حذف تخصیص های استفاده نشده

- حذف عبارت های مشترک محلی

02- : همه بهینه سازی های 01- بعلاوه :

- بهینه سازی حلقه ها

- حذف تخصیص های سراسری استفاده نشده

- حذف عبارت های فرعی مشترک سراسری

03- : همه بهینه سازی های 02- بعلاوه :

- حذف توابعی که صدا نشده

- ساده سازی توابع با مقدار برگشتی استفاده نشده

- inline کردن صدازدن توابع کوچک

- شناسایی مشخصات متغیر ها در سطح فایل

در این مرحله از پیاده سازی، پروژه CODEC را با استفاده از optimizer و با حداکثر قدرت بهینه سازی

03- و بهینه سازی در سطح فایل اجرا کردیم . نتایج بدست آمده در شکل (۵-۹) نشان می دهد که

optimizer کاهش چشمگیری در MIPS و حجم کد برنامه بوجود آورده ولی هنوز هم MIPS برنامه بیش

از دو برابر قدرت C5402 است و حتی در حالت ماکزیمم به ۲۳۳۸۹۲ در هر دور اجرا می رسد.

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

is	Code Size	Count	Incl. Total	Incl. Maximum	Incl. Minimum	Incl. Aver
LevinsonDurbin50	389	16	136754	91283	46	8547
LevinsonDurbin50e	355	16	135902	90998	28	8493
HybWin49	179	16	832522	53104	51360	52032
HybWin49e	172	16	791010	50496	48792	49438
BLOCK17_18	193	65	3052663	47081	46752	46964
PITCH_EXTRACT82	506	16	433522	37380	17695	27095
BLOCK14_15	62	17	544153	32009	32009	32009
HWMCORE2	476	82	1046001	26091	4102	12756
HWMCORE1	512	82	740526	19236	2377	9030
HybWin36	143	16	236629	14898	14743	14789
HybWin43	152	17	181521	10784	10624	10677
HybWin43e	152	17	181521	10784	10624	10677
LevinsonDurbin44e	355	17	59133	10098	28	3478
LevinsonDurbin44	356	17	59017	10080	28	3471
DEC_SYN_FILTER32	459	65	606084	9495	9069	9324
BLOCKZIR	301	65	465758	7314	6908	7165

شکل (۵-۹): خروجی Profiler در حالت استفاده از Optimizer

- برنامه نویسی اسمبلی بصورت دستی همانطور که در قسمت قبل دیدیم، کمپایلر C54x حتی با استفاده از optimizer هم نتوانست اجرای برنامه را به 100 MIPS برساند. این بدین دلیل است که در خانواده C54x، optimizer به حد کافی قوی نیست چراکه ما همین برنامه را با استفاده از optimizer پردازنده C55x کمپایل کردیم و به حدود 40 MIPS برای اجرای آن بر روی C55x نیاز بود.

به هر حال باید اجرای این برنامه را به 100 MIPS برسانیم تا بتوان بصورت بلادرنگ آنرا پیاده سازی کرد. در این مرحله تنها راهی که باقی مانده اینست که بر روی توابع برنامه و MIIPS آنها و نتایج بدست آمده از مرحله قبل بررسی کرده تا توابع و قسمت هایی که به نظر میرسد Optimizer نتوانسته خوب بهینه

## برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

سازد را بصورت دستی برنامه نویسی کنیم. واضح است که در این مرحله باید به زبان اسمبلی C54x تسلط کافی داشت تا بتوان کد اسمبلی تولید شده توسط کمپایلر C54x را بهینه کرد.

در ابتدا ملاحظه می شود که در محاسبات کورلیشن در بعضی از حلقه ها از دستور ضرب/انباشت MAC استفاده نشده و حلقه چند دستوری بوجود آمده است که می توان آنها را با استفاده از این دستور به حلقه تک سیکلی تبدیل کرد. همچنین در بعضی از حلقه های محاسبه انرژی نیز می توان از دستور تک سیکلی مجذور SQURA استفاده نمود. با انجام این اصلاحات MIPS برنامه کاهش یافت ولی هنوز فاصله زیادی با مقدار مورد نیاز ما دارد.

در این مرحله مهمترین توابع برنامه از نظر MIPS - همانطور که در شکل (۵-۹) دیده می شود - همچون  $Block14\_15()$ ،  $Block17\_18()$ ،  $HybWin49()$ ،  $LevinsonDurbin50()$  را بصورت دستی بازنویسی کردیم. در اینجا بعنوان نمونه به  $Block14\_15()$  که کوچکتر است می پردازیم:

همانطور که در کد C این تابع در ضمیمه (ب) دیده می شود، تابع از ۳ حلقه تو در تو تشکیل شده است. حلقه بیرونی  $NCWD=128$  بار، حلقه میانی  $IDIM=5$  بار و حلقه داخلی از ۱ تا ۵ بار اجرا می گردند. سیکل دستورالعمل این تابع بیش از ۳۲۰۰۰ است (شکل (۵-۹)). در ادامه کد اسمبلی تولید شده توسط کمپایلر با حد اکثر بهینه سازی و کد بازنویسی شده بصورت دستی در این ضمیمه آورده شده است. در بازنویسی این بلوک به این نکته توجه شده که دستورالعمل های حلقه داخلی به حد اقل برسد چرا که این حلقه در ضریب  $5 \times 128 = 640$  ضرب می شود.

نتایج این بهینه سازی و برنامه نویسی در شکل (۵-۱۰) به ترتیب صعودی بیشترین MIPS نشان داده شده است. با مقایسه نتایج این شکل و شکل (۵-۹) می توان گفت که MIPS بعضی از توابع همچون

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

LevinsonDurbin() به کمتر از یک سوم و بعضی دیگر همچون Block14\_15() به کمتر از نصف رسیده

است و نتیجه کلی اینکه هر دور اجرای برنامه که شامل اینکدر و دیکدر بصورت دوطرفه کامل (Full

Duplex) می باشد ، کمتر از ۶۲۵۰۰ شده است و می توان آنرا بصورت بلادرنگ اجرا نمود .

Functions	Code Size	Count	Incl. Total	Incl. Maximum	Incl. Minimum	Incl. Aver
codec.out						
LevinsonDurbin50	297	17	36428	26453	39	2142
LevinsonDurbin50e	233	17	35687	26187	28	2099
HybWin49	179	17	296596	18229	17241	17446
HybWin49e	168	17	257603	15961	15019	15153
PITCH_EXTRACT82	504	18	138998	14368	5189	7722
BLOCK14_15	58	19	268014	14106	14106	14106
BLOCK17_18	132	71	588319	8301	8273	8286
HwMCOE2	434	88	325146	7396	1507	3694
HybWin36	143	18	124094	6971	6847	6894
HwMCOE1	460	88	279042	6597	1249	3170
HybWin43	152	18	102510	5782	5665	5695
HybWin43e	152	18	102510	5782	5665	5695
DEC_SYN_FILTER32	459	71	368155	5275	5063	5185
BLOCKZIR	301	71	287752	4134	3932	4052
LevinsonDurbin44e	355	18	23314	3630	20	1295
LevinsonDurbin44	356	18	23180	3612	20	1287
PITCH_PREDICT_...	121	18	31653	1994	1143	1758
FiltrMemUpdt	460	71	126376	1896	1608	1779
BLOCK71_72	166	71	126238	1778	1778	1778
FINDNLS	187	54	67860	1757	1016	1256
BAND_EXP51	64	2	3504	1752	1752	1752

شکل (۵-۱۰) : نتایج برنامه نویسی دستی

### ۵-۳-۵- نتایج پیاده سازی

در این قسمت به نتایج پیاده سازی بلادرنگ کدک G.728 می پردازیم . معمولاً برای ارزیابی یک کدک

صحت از معیار های Subjective و Objective استفاده می شود. معروفترین معیار Subjective ، معیار



## برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

MOS<sup>1</sup> است که دارای ۵ سطح از ۱ (بد) تا ۵ (عالی) می باشد. این معیار به شرایط آزمایش، زبان و شنوندگان وابستگی دارد. معروفترین معیار Objective، معیار SNR است که اعوجاج بین ورودی و خروجی سیستم را در نظر می گیرد و بصورت نسبت توان ورودی به توان خطای بین ورودی و خروجی تعریف می شود.

معمولا در صحبت به دلیل تفاوت انرژی قسمت های باواک و بی واک از معیار SNRseg بصورت زیر استفاده می شود:

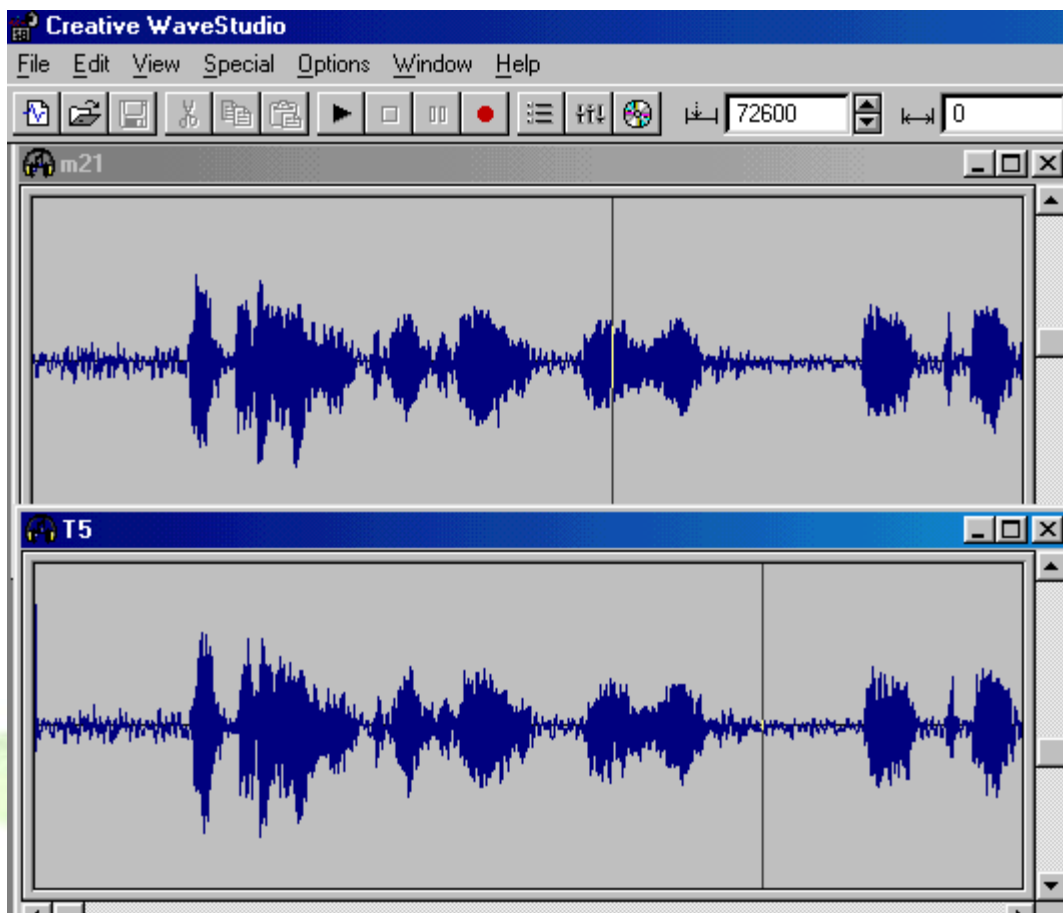
$$SNR_{seg} = 1/M \sum_{m=1}^M 10 \log \frac{\sum_{n=1}^N S_{in}^2(n)}{\sum_{n=1}^N [S_{out} - S_{in}]^2}$$

که در آن N تعداد نمونه های صحبت در یک سگمنت که معمولا ۱۲۸ تا ۲۵۶ نمونه است و M تعداد این سگمنت ها می باشد [7].

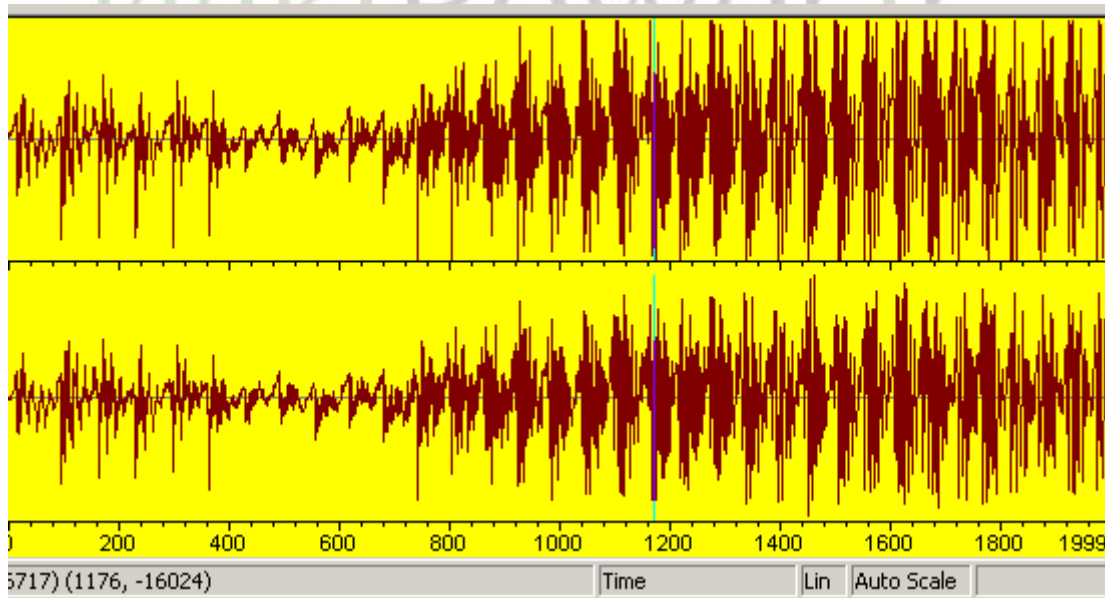
در شکل (۵-۱۱) صحبت ورودی و صحبت سنتز شده خروجی کدک حاصل از شبیه سازی ممیز ثابت و در شکل (۵-۱۲) مقایسه ورودی و خروجی در پیاده سازی بر روی C5402 نشان داده شده است. فایل صحبت جمله " بر سرآتم که گر ز دست بر آید..." با گوینده مرد می باشد. همچنین در جدول (۵-۲) مقادیر SNR محاسبه شده برای سگمنت های ۲۵۶ تایی و کل فایل آورده شده است.

<sup>1</sup> Mean Opinion Score

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه



شکل (۵-۱۱): مقایسه صحبت ورودی (بالا) و صحبت سنتز شده (پایین) در شبیه سازی



شکل (۵-۱۲): مقایسه صحبت ورودی (بالا) و صحبت سنتز شده (پایین) در DSP

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

	SEG256	GLOBAL
SNR (dB)	17.31	17.57

جدول (۲-۵)

در مرجع [7] مقدار SNR برای این کدک از نظر تئوری 22dB محاسبه شده است هرچند که در پیاده سازی ها مثلا مرجع [17] ، SNR کلی کدک برابر 18dB بدست آمده است .

در اینجا نتایج پیاده سازی بر روی پردازنده TMS320C5402 از نظر مقدار حافظه برنامه و حافظه دیتا و MIPS مورد نیاز برنامه در جدول های (۳-۵) و (۴-۵) ارائه می شوند :

Program Memory	Data Memory
8582 word	3632 word

جدول (۳-۵)

	Maximum	Average
Encoder	61.13	54.6
Decoder	38.17	32.8
Full Duplex	99.3	87.4

جدول (۴-۵)

برای مقایسه به نتایج پیاده سازی شرکت DCS بر روی C54X که به حافظه برنامه 9 kword و حافظه دیتا 2.1 kword و پردازش 37 MIPS نیاز دارد [19]، اشاره می کنیم که البته اختلاف MIPS آن با پیاده سازی ما ناشی از برنامه نویسی اسمبلی بصورت کاملا دستی می باشد .

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

## ۵-۶- نتیجه گیری و پیشنهاد

در این گزارش به پیاده سازی بلادرنگ کدک صحبت کم تاخیر G.728 بر روی پردازنده TMS320C5402 پرداختیم. در این راه پس از بررسی روش های کدینگ صحبت و مطالعه الگوریتم کدینگ LD-CELP، این الگوریتم را به زبان C و بصورت ممیز ثابت شبیه سازی نمودیم. سپس برد توسعه گر C5402 DSK و نرم افزار CCS (Code Composer Studio) و امکانات آنرا معرفی و نحوه اجرای برنامه بر روی برد را بیان کردیم. با استفاده از نرم افزار CCS برنامه کدک به زبان C را به کد اسمبلی ترجمه و بر روی TMS اجرا نموده و دیدیم که کمپایلر C54X با حداکثر قدرت بهینه سازی هم نمی تواند برنامه را بصورت بلادرنگ اجرا نماید. از اینرو توابع مهم برنامه از نظر MIPS را بصورت دستی به زبان اسمبلی بازنویسی کردیم. در نتیجه MIPS بعضی از توابع به نصف و بعضی به یک سوم کاهش یافت و برنامه بصورت بلادرنگ قابل اجرا گردید.

بعنوان اولین پیشنهاد می توان با اعمال تغییراتی در ساختار کتاب کد LD-CELP 16 kb/s آنرا به یک کدک با نرخ بیت متغیر تبدیل نمود که جزئیات آن در Annex H توصیه نامه G.728 آورده شده است [18]. چنین کدکی با نرخ بیت متغیر برای شبکه های موبایل و اینترنت مناسب است.

پیشنهاد دیگری که در اینجا مطرح می شود ترکیب این کدک با حذف کننده اکو استاندارد G.165 است. چرا که در سیستم های انتقال صحبت بلوک های فشرده سازی و حذف اکو در کنار هم قرار می گیرند و اگر هردو در یک بلوک پیاده سازی شوند می توان از نظر MIPS و حافظه به نتایج بهتری دست یافت.

برای دریافت فایل Word پروژه به سایت **ویکی پاور** مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

## ضمائم

- ضمیمه (ب): مقایسه برنامه نویسی C و اسمبلی

```

/*****
BLOCK 14,15 :SHAPE CODEVECTOR CONVOLUTION &
ENERGY CALCULATION
inputs: H,Y
output: Y2
/*****/
void BLOCK14_15()
{
    int I,J,K,K1=1;
    long AA0;

    for(J=1;J<=NCWD;J++)
    {
        for(K=1;K<=IDIM;K++)
        {
            K1++;//K1=(J-1)*IDIM+K+1;
            AA0=0;
            for(I=1;I<=K;I++)
                AA0+=H[I]*Y[K1-I];

            AA0>>=14;
            TEMP[K]=AA0;
        }
        AA0=0;
        for(K=1;K<=IDIM;K++)
            AA0+=(TEMP[K]*TEMP[K]);
        AA0>>=15;
        Y2[J]=AA0;
    }
}

```

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

```

/*****/
        کد اسمبلی تولید شده توسط کمپایلر
/*****/

.sect ".text"
.global _BLOCK14_15
.sym _BLOCK14_15,_BLOCK14_15, 32, 2, 0
.func 1757
;-----
; 1757 | void BLOCK14_15()
;-----
;*****
;* FUNCTION DEF: _BLOCK14_15 *
;*****
_BLOCK14_15:
.line 2
.sym _I,0, 4, 1, 16
.sym _J,1, 4, 1, 16
.sym _K,2, 4, 1, 16
.sym _K1,3, 4, 1, 16
.sym _AA0,4, 5, 1, 32
PSHM AR1
FRAME #-6
NOP
.line 3
;-----
; 1759 | int I,J,K,K1=1;
; 1760 | long AA0;
;-----
ST #1,*SP(3) ; |1759|
.line 6
;-----
; 1762 | for(J=1;J<=NCWD;J++)
;-----
SSBX SXM
LD #128,A
ST #1,*SP(1) ; |1762|
SUB *SP(1),A ; |1762|
BC L182,ALT ; |1762|

```

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

```

; branch occurs ; |1762|
L175:
.line 8
;-----
; 1764 | for(K=1;K<=IDIM;K++)
;-----
LD #5,A
ST #1,*SP(2) ; |1764|
SUB *SP(2),A ; |1764|
BC L179,ALT ; |1764|
; branch occurs ; |1764|
L176:
.line 10
;-----
; 1766 | K1++;//K1=(J-1)*IDIM+K+1;
;-----
ADDM #1,*SP(3) ; |1766|
.line 11
;-----
; 1767 | AA0=0;
;-----
LD #0,A
DST A,*SP(4) ; |1767|
.line 12
;-----
; 1768 | for(I=1;I<=K;I++)
;-----
ST #1,*SP(0) ; |1768|
LD *SP(2),A ; |1768|
SUB *SP(0),A ; |1768|
BC L178,ALT ; |1768|
; branch occurs ; |1768|
L177:
.line 13
;-----
; 1769 | AA0+=H[I]*Y[K1-I];
;-----
LD *SP(3),A
SUB *SP(0),A ; |1769|
STLM A,AR1
NOP
NOP

```

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

```

LD *AR1(_Y),T
MVDK *SP(0),*(AR1)
MPY *AR1(_H),A ; |1769|
DADD *SP(4),A,A ; |1769|
ADDM #1,*SP(0) ; |1769|
DST A,*SP(4) ; |1769|
LD *SP(2),A ; |1769|
SUB *SP(0),A ; |1769|
BC L177,AGEQ ; |1769|
; branch occurs ; |1769|
L178:
.line 15
;-----
; 1771 | AA0>>=14;
;-----
DLD *SP(4),A ; |1771|
SFTA A,#-14,A ; |1771|
DST A,*SP(4) ; |1771|
.line 16
;-----
; 1772 | TEMP[K]=AA0;
;-----
MVDK *SP(2),*(AR1)
LD *SP(5),A
STL A,*AR1(_TEMP)
.line 17
ADDM #1,*SP(2) ; |1773|
LD #5,A
SUB *SP(2),A ; |1773|
BC L176,AGEQ ; |1773|
; branch occurs ; |1773|
L179:
.line 18
;-----
; 1774 | AA0=0;
;-----
LD #0,A
DST A,*SP(4) ; |1774|
.line 19
;-----
; 1775 | for(K=1;K<=IDIM;K++)
;-----

```



برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

```

ST #1,*SP(2) ; |1775|
LD #5,A
SUB *SP(2),A ; |1775|
BC L181,ALT ; |1775|
; branch occurs ; |1775|
L180:
.line 20
;-----
; 1776 | AA0+=(TEMP[K]*TEMP[K]);
;-----
MVDK *SP(2),*(AR1)
LD *AR1(_TEMP),T
MPY *AR1(_TEMP),A ; |1776|
DADD *SP(4),A,A ; |1776|
ADDM #1,*SP(2) ; |1776|
DST A,*SP(4) ; |1776|
LD #5,A
SUB *SP(2),A ; |1776|
BC L180,AGEQ ; |1776|
; branch occurs ; |1776|
L181:
.line 21
;-----
; 1777 | AA0>>=15;
;-----
DLD *SP(4),A ; |1777|
SFTA A,#-15,A ; |1777|
DST A,*SP(4) ; |1777|
.line 22
;-----
; 1778 | Y2[J]=AA0;
;-----
MVDK *SP(1),*(AR1)
LD *SP(5),A
STL A,*AR1(_Y2)
.line 24
ADDM #1,*SP(1) ; |1780|
LD #128,A
SUB *SP(1),A ; |1780|
BC L175,AGEQ ; |1780|
; branch occurs ; |1780|
L182:

```

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

```
.line 26
FRAME #6
POPM AR1
RET
; return occurs
.endfunc 1782,000000400h,7
```

```
/*
*****
*/
```

کد اسمبلی تولید شده با برنامه نویسی دستی

```
/*
*****
*/
```

```
.text
.global _BLOCK14_15
_BLOCK14_15:
PSHM    AR0
PSHM    AR1
PSHM    AR2
PSHM    AR3
PSHM    AR4
PSHM    AR5
PSHM    AR6
frame #-2
ssbx    sxm
STM     #_Y2+1,AR4
STM     #0,BK
STM     #127,AR1
stm     _Y,ar2
L1:
STM     #_TEMP+1,AR0
stm     #-1,ar6
stm     #4,brc
```

```
rptb L2
mar *ar2+
mar *ar6+
```

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

```
ldm ar2,a
stlm a,ar5
ldm ar6,a
stl a,*sp(0)
STM #_H+1,AR3
LD #0,A
RPT *sp(0)
MAC *AR5-, *AR3+, A, A

SFTA A,#-14,A
L2: STL A,*AR0+
STM #_TEMP+1,AR3
LD #0,b
rpt #4
squra *ar3+,b
SFTA B,#-15,B
STL B,*AR4+
BANZ L1,*AR1-
frame #2
POPM AR6
POPM AR5
POPM AR4
POPM AR3
POPM AR2
POPM AR1
POPM AR0
RET
```

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

## مراجع

- 1- Panos E. Papamichalis, "Practical Approaches to Speech Coding ", Prentice-Hall Inc. ,1987.
- 2-B.S. Atal & R.Remde "A new model of LPC excited for producing natural-sounding speech at low bit rates", Proc.ICASSP pp.614-617 1982.
- 3-Jason P. Woodard , "Digital Speech Coding" , Mini-Thesis , Department of Electronics & Computer Science, University of Southampton, Jun 1994.
- 4-J.Makhoul "Linear Prediction: A Tutorial Review " Proc. IEEE,vol 63, No.4,pp.561-580 Apr 1975.
- 5- Kondoz A.M," Digital Speech , Coding of low bit rate communication Systems", Chichester Wiley 2000
- 6- Chen,Cox & Lin,"A Low-Delay CELP Coder for the CCITT 16 kb/s Speech Coding Standard",IEEE Jour. On Selected Area in Comm.,vol.10,no.5 ,June 92.
- 7- L.Hanzo,A.Somerville & Jason P. Woodard, "Voice Compression and Communication",IEEE series on Digital & Mobile,2001
- 8- Schroeder & Atal , "Code-Excited linear Prediction (CELP):High quality speech at very low bit rates",IEEE,ICASSP,pp.937-940,1985.
- 9- ITU,"Coding of Speech at 16 kbps using Low-Delay Code Excited  
Linear Prediction ", ITU (CCITT) Recommendation G.728, 1992.
- 10- ITU," G.728 Annex G , 16 kbps Fixed Point specification ", 11/94.
- 11- TI, "TMS320C54x Assembly language tools user guide ", www.ti.com  
June 2001.
- 12- TI, " TMS320C54x DSP reference set, volume 1: CPU", spru131, www.ti.com.
- 13- TI, " TMS320C54x DSP reference set, volume 2: Mnemonic instruction set",  
spru172, www.ti.com .
- 14- TI, "Code Composer user guide",spru 328, www.ti.com.

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

15- TI, "C5402 DSK user guide" ,www.ti.com.

16- TI," DSP product tree" , <http://dspvillage.ti.com/docs/allproducttree.jhtml>.

17- TCTS Lab," The LD-CELP at 16kb/s (ITU-T G.728)",Coding research group homepage ,<http://tcts.fpms.ac.be/coding.htm>.

18- ITU,"G.728 Annex H:variable bit rate LD-CELP operation mainly for DCME at rates less than 16 kb/s",5/99.

19- DCS G.728 C54x Vocoder,

algorithm,<http://wwwd.connect.ti.com/dsp/tpcat/tpcodec.nsf/SoftwareForExternal/EAB728D36C3C916E862569F200542A92>.

