



دانشکده مهندسی برق و کامپیوتر

گروه مهندسی الکترونیک

عنوان پایان نامه جهت اخذ مدرک کارشناسی ارشد در

رشته مهندسی الکترونیک

پیاده سازی الگوریتم توربو کدینگ بر روی FPGA

استادان راهنما

دکتر قادر کریمیان

دکتر جعفر صبحی

استاد مشاور

دکتر میر جواد موسوی نیا

پژوهشگر

کامبیز نظامی بلوچی

زمستان ۹۰

تقدیم بہ خانوادہ عزیزم

تقدیر و تشکر

در اینجا لازم می‌دانم از زحمات بی دریغ و محبت‌های همیشگی خانواده عزیزم تشکر کنم که در سخت‌ترین شرایط همواره پشتیبان من بوده و با حمایت‌ها و راهنمایی‌های موثر خود، مهمترین دلیل برای پیشرفت و ترقی اینجانب بوده‌اند.

همچنین از تمامی اساتید گروه الکترونیک دانشکده برق دانشگاه تبریز، به خاطر زحمات بی‌وقفه‌ای که برای من و کلیه دانشجویان متحمل می‌شوند تشکر می‌کنم به ویژه استاد عزیزم، دکتر قادر کریمیان خسروشاهی که در طول دوره کارشناسی ارشد مطالب بسیار ارزنده‌ای از ایشان فرا گرفتم.

در پایان از تمامی دوستان و هم‌اتاقی‌های عزیزم که در دوره کارشناسی ارشد با آنها آشنا شدم تشکر می‌کنم که همیشه با راهنمایی‌های خود راه گشای اینجانب در هنگام مشکلات بوده‌اند به ویژه: صمد فهیمی، ابراهیم اکبری، سامان عقیلی، علی صباغی‌فر، هومن دهقانی، جعفر طاووسی، میر مجید قاسمی، نوید وفایی، محمد احمد لو، امیر حسین وافی، سید مهدی سیدین، فردین مقیمی، نیما لطفی، علیرضا خلیل زاده، سهند محرمی، افشین وطن خواه، ابوذر غفاری، مسعود ملک زاده، مهرداد احمدی، سلمان نخجیرکان، ابوذر تقی زاده، عباس روشنی، پویا مبادرثانی، علی جوادی، محمد حاج قاسملو.

نام خانوادگی: نظامی بلوچی	نام: کامبیز
عنوان پایان نامه: پیاده سازی الگوریتم توربو کدینگ بر روی FPGA	
استاد راهنمای اول: دکتر قادر کریمیان	استاد راهنمای دوم: دکتر جعفر صبحی
استاد مشاور: دکتر میر جواد موسوی نیا	
مقطع تحصیلی: کارشناسی ارشد	رشته: مهندسی برق
مقطع تحصیلی: کارشناسی ارشد	گرایش: طراحی مدارات مجتمع آنالوگ
دانشگاه: تبریز	دانشکده: مهندسی برق و کامپیوتر
تعداد صفحات: ۱۰۷	
واژه های کلیدی: توربو کدینگ، کد کننده، کانولوشنی، کد گشا، اینترلیور، د اینترلیور، MAP، SOVA، FPGA، Verilog، JSE	
چکیده:	
<p>در این پایان نامه هدف، بررسی، انتخاب و پیاده سازی الگوریتم توربو کدینگ بر روی تراشه های قابل برنامه ریزی میباشد. توربو کدها از زمان معرفی آنها به عنوان یک کد تصحیح خطای پیشرو، مورد توجه بوده اند. به دلیل قدرت اجرایی بالای توربو کدها، آنها توسط استانداردهای مختلفی مانند 3GPP به عنوان یک طرح کدینگ کانال مورد استفاده قرار گرفته اند. توربو کدها تکنیک کدینگ کانال قدرتمندی را برای نسل بعدی ارتباطات ارائه می دهند و پژوهشهای زیادی برای پیدا کردن روش مناسب جهت پیاده سازی توربو کدها انجام گرفته است و با روشهای مختلف سعی در بهینه سازی عمل پیاده سازی کرده اند و مواردی همچون نرخ خطای بیت، زمان تاخیر و سرعت محاسبات را بهبود داده اند.</p>	

فهرست مطالب

أ.....	فهرست مطالب
د.....	فهرست شکل ها
ز.....	فهرست جدول ها
۲.....	۱ آشنایی با کدینگ و پیشینه آن
۲.....	۱-۱ مقدمه
۴.....	۲-۱ آشنایی با نظریه کدینگ
۵.....	۳-۱ کدهای بلوکی
۶.....	۴-۱ کدهای کانولوشنی
۷.....	۵-۱ کدهای به هم پیوسته
۷.....	۶-۱ کدهای توربو
۸.....	۷-۱ کد گشا
۹.....	۸-۱ تشریح گرافیکی کد کننده ها
۱۰.....	۱-۸-۱ دیاگرام درختی
۱۱.....	۲-۸-۱ دیاگرام حالت
۱۱.....	۳-۸-۱ دیاگرام شبکه
۱۴.....	۲ کد کننده و کد گشای توربو
۱۴.....	۱-۲ کد کننده
۱۴.....	۲-۱-۱ کد کننده کانولوشنی
۱۸.....	۲-۱-۲ کد کننده های کانولوشنی RSC

۱۹ ۳-۱-۲ کد کننده‌های توربو
۲۰ ۴-۱-۲ اختتام شبکه
۲۶ ۲-۲ اینترلیورها
۲۷ ۱-۲-۲ اینترلیور تصادفی
۲۷ ۲-۲-۲ اینترلیور بلوکی
۲۷ ۳-۲-۲ اینترلیور QPP
۲۹ ۴-۲-۲ اینترلیور سطری-ستونی
۳۰ ۵-۲-۲ اینترلیور مورب
۳۰ ۶-۲-۲ اینترلیور زوج-فرد
۳۰ ۳-۲ کد گشای توربو
۳۲ ۱-۳-۲ ایده اولیه برای کد گشایی:
۳۳ ۲-۳-۲ کد گشایی به روش ترتیبی
۳۵ ۳-۳-۲ الگوریتم MAP
۳۹ ۴-۳-۲ الگوریتم LOG-MAP
۴۰ ۵-۳-۲ الگوریتم MAX-LOG-MAP
۴۰ ۶-۳-۲ الگوریتم ویتربی
۴۴ ۷-۳-۲ الگوریتم SOVA
۵۸ ۳ طراحی و پیاده سازی الگوریتم توربو کد
۵۸ ۱-۳ طراحی کد کننده توربو
۵۹ ۱-۱-۳ طراحی کد کننده با شیفت رجیستر

۶۱	۲-۱-۳ طراحی کد کننده با LUT
۶۳	۲-۳ طراحی کد گشای MAP
۶۶	۳-۳ طراحی کد گشای SOVA
۷۳	۴-۳ طراحی توربو کد پیشنهادی
۷۷	۵-۳ پیاده سازی توربو کد بر روی FPGA
۷۷	۱-۵-۳ پیاده سازی کد کننده
۷۹	۶-۳ پیاده سازی اینترلیور زوج-فرد
۸۰	۷-۳ پیاده سازی کد گشای SOVA
۸۲	۱-۷-۳ بلوک محاسبات پیش رو
۸۴	۲-۷-۳ بلوک محاسبات پس رو
۸۶	۳-۷-۳ بلوک محاسبه قابلیت اطمینان
۹۱	۴ نتیجه گیری و پیشنهادات
۹۱	۱-۴ نتیجه گیری
۹۲	۲-۴ پیشنهادات
۹۴	منابع

فهرست شکل ها

- شکل ۱-۱ بلوک دیاگرام یک سیستم مخابراتی دیجیتال ۳
- شکل ۲-۱ بلوک دیاگرام کد کننده به هم پیوسته ۷
- شکل ۳-۱ کد کننده توربو مرتبه ۲ ۸
- شکل ۴-۱ بلوک دیاگرام یک کد گشای توربو ۹
- شکل ۵-۱: دیاگرام درختی ۱۰
- شکل ۶-۱: دیاگرام حالت ۱۱
- شکل ۷-۱ دیاگرام شبکه ۴ حالت ۱۲
- شکل ۱-۲ کد کننده کانولوشنی ۴ حالت ۱۵
- شکل ۲-۲ دیاگرام حالت کد کننده ۴ حالت ۱۶
- شکل ۳-۲ دیاگرام شبکه کد کننده ۴ حالت ۱۷
- شکل ۴-۲ دیاگرام شبکه برای ورودی $u=\{110100\}$ ۱۷
- شکل ۵-۲ کد کننده RSC ۴ حالت ۱۸
- شکل ۶-۲ دیاگرام شبکه کد کننده شکل ۵-۲ ۱۹
- شکل ۷-۲ دیاگرام شبکه کد کننده RSC برای ورودی $u=\{1,0,1,0,1,0,0\}$ ۱۹
- شکل ۸-۲ کد کننده توربو با دو کد کننده موازی RSC ۲۰
- شکل ۹-۲ فرآیند کد کردن بدون عملیات اختتام شبکه ۲۲
- شکل ۱۰-۲ اختتام شبکه با بیت‌های دنباله صفر در یک کد کننده کانولوشنی با $m=6$ ۲۳
- شکل ۱۱-۲ اختتام شبکه با m بیت اول داده اصلی ۲۵
- شکل ۱۲-۲ بلوک دیاگرام کد گشای توربو ۳۲
- شکل ۱۳-۲ دیاگرام شبکه کد گشایی به روش ترتیبی ۳۴
- شکل ۱۴-۲ مسیر حرکت برای محاسبه مقادیر α و β ۳۷
- شکل ۱۵-۲ دیاگرام شبکه کد گشایی رشته دریافتی $v=\{0100010000\}$ ۴۳

- شکل ۲-۱۶ کد گشایی به روش ویتربی در مسیر پیش رو ۴۶
- شکل ۲-۱۷ مقادیر محاسبه شده Δ برای مسیر ML ۴۷
- شکل ۲-۱۸ دیاگرام شبکه کد گشای دوم در چرخش اول ۵۰
- شکل ۲-۱۹ دیاگرام شبکه کد گشای اول در چرخش دوم ۵۲
- شکل ۲-۲۰ ساختار کد گشای ویتربی در [۳۰] ۵۳
- شکل ۲-۲۱ ساختار داخلی بلوک TMU ۵۳
- شکل ۲-۲۲ ساختار داخلی بلوک ACS ۵۴
- شکل ۲-۲۳ بلوک دیاگرام پیاده سازی الگوریتم SOVA ۵۵
- شکل ۲-۲۴ بلوک ACS و محاسبه Δ ۵۵
- شکل ۲-۲۵ نمای داخلی بلوک به هنگام ساز قابلیت اطمینان ۵۶
- شکل ۳-۱ کد کننده توربو کانولوشنی ۴ حالت ۵۸
- شکل ۳-۲ فلوجارت یک کد کننده ۴ حالت ۶۰
- شکل ۳-۳ فلوجارت کد کننده با LUT ۶۲
- شکل ۳-۴ فلوجارت کد گشایی به روش MAP ۶۳
- شکل ۳-۵ نمودار BER به E_b/N_0 برای کد گشایی توربو توسط الگوریتم MAP ۶۴
- شکل ۳-۶ فلوجارت محاسبه: الف. γ ب. α ج. β ۶۵
- شکل ۳-۷ نمودار نرخ خطا برای کد گشای توربو در [۳۱] ۶۶
- شکل ۳-۸ بلوک دیاگرام کد گشای ویتربی ۶۷
- شکل ۳-۹ فلوجارت الگوریتم SOVA در مسیر پیش رو ۶۹
- شکل ۳-۱۰ فلوجارت فرآیند محاسبه قابلیت اطمینان ۷۱
- شکل ۳-۱۱ نمودار میزان خطای خروجی کد گشای SOVA در چرخشهای مختلف ۷۲
- شکل ۳-۱۲ نمودار به دست آمده در [۳۲] بر اساس کد گشای SOVA ۷۳
- شکل ۳-۱۳ کد کننده توربو پیشنهادی ۷۴
- شکل ۳-۱۴ کد گشای توربو پیشنهادی ۷۵

- شکل ۳-۱۵ نمودار نرخ خطای بی‌تی برای کد‌گشای توربو پیشنهادی ۷۶
- شکل ۳-۱۶ نمای کلی بلوک کد‌کننده در نرم‌افزار ISE ۷۸
- شکل ۳-۱۷ شماتیک کد‌کننده طراحی شده در ISE ۷۸
- شکل ۳-۱۸ بلوک اینترلیور زوج-فرد پیاده‌سازی شده در ISE ۷۹
- شکل ۳-۱۹ بلوک کد‌گشای توربو پیاده‌سازی شده در ISE ۸۰
- شکل ۳-۲۰ بلوک کد‌گشای پیاده‌سازی شده در ISE ۸۱
- شکل ۳-۲۱ دریافت اطلاعات و بار‌گذاری اولیه در ISE ۸۲
- شکل ۳-۲۲ بلوک محاسبات پیش‌رو ۸۲
- شکل ۳-۲۳ نمای داخلی بلوک محاسبات پیش‌رو در SOVA ۸۳
- شکل ۳-۲۴ نتیجه محاسبات مسیر پیش‌رو استخراج شده از ISim ۸۴
- شکل ۳-۲۵ بلوک محاسبات پس‌رو در ISE ۸۴
- شکل ۳-۲۶ نمای داخلی بلوک محاسبات پس‌رو ۸۵
- شکل ۳-۲۷ محاسبات پس‌رو استخراجی از ISim ۸۶
- شکل ۳-۲۸ بلوک محاسبه قابلیت اطمینان در ISE ۸۶
- شکل ۳-۲۹ مدار شمارنده در بلوک قابلیت اطمینان ۸۷
- شکل ۳-۳۰ محاسبه قابلیت اطمینان برای مسیر ML ۸۷
- شکل ۳-۳۱ واحد محاسبه خروجی کد‌گشا (L_e) ۸۸
- شکل ۳-۳۲ محاسبات مرحله قابلیت اطمینان استخراجی از ISim ۸۸

فهرست جدول ها

- جدول ۱-۲ مقادیر حالت بعدی، خروجی، حالت فعلی و ورودی یک کد کننده ۴ حالتی ۱۵
- جدول ۲-۲ پارامترهای اینترلیور QPP ۲۹
- جدول ۳-۲ نوشته شدن اطلاعات به صورت سطری ۳۰
- جدول ۴-۲ خوانده شدن اطلاعات از حافظه ۳۰
- جدول ۵-۲ خواندن اطلاعات در اینترلیور مورب ۳۰
- جدول ۶-۲ بررسی تعداد خطا بر اساس بیت ورودی، کد تولید شده و کد دریافتی ۳۳
- جدول ۲-LUTV برای یک کد کننده ۸ حالتی و نرخ بیت ۱/۲ ۳۵
- جدول ۸-۲ خروجی محاسبه شده در مسیر برگشت در کد گشای SOVA ۴۸
- جدول ۹-۲ مقادیر محاسبه شده نهایی خروجی کد گشا توسط رابطه (۲-۲۴) ۴۹
- جدول ۱۰-۲ مقادیر محاسبه شده توسط کد گشای دوم ۵۱
- جدول ۱-LUT یک کد کننده ۴ حالتی ۶۱
- جدول ۲-۳ مقادیر BER محاسبه شده در روش پیشنهادی به ازاء چرخش ها و E_b/N_0 های مختلف ۷۶
- جدول ۳-۳ مشخصات سخت افزاری پیاده سازی کد کننده ۴ حالتی SOVA استخراج شده از ISE ۷۷
- جدول ۴-۳ اطلاعات سخت افزاری طراحی ترتیب کننده زوج-فرد ۸۰
- جدول ۵-۳ مشخصات سخت افزاری پیاده سازی کد گشای SOVA استخراج شده از ISE ۸۹

فصل ۱

آشنایی با کدینگ

و

پیشینه آن

آشنایی با کدینگ و پیشینه آن

۱- مقدمه

با گذشت سال‌ها و با توجه به نیاز مبرم به انتقال اطلاعات و ایجاد کانال‌های ارتباطی امن و دقیق، رشد فزاینده‌ای در زمینه ارتباطات ماهواره‌ای، ارتباطات سیار و ارتباطات کامپیوتری رخ داده است و یک سیستم مخابراتی مناسب نقش بسزایی در این میان ایفا می‌کند.

یک سیستم مخابراتی از سه جزء اصلی تشکیل می‌شود: ۱- فرستنده ۲- کانال مخابراتی ۳- گیرنده.

یک فرستنده سیگنال ورودی را پردازش می‌کند تا یک سیگنال مخابراتی متناسب با مشخصات کانال، ایجاد کند. پردازش سیگنال برای انتقال همیشه با مدولاسیون همراه است و می‌تواند شامل کدگذاری هم باشد.

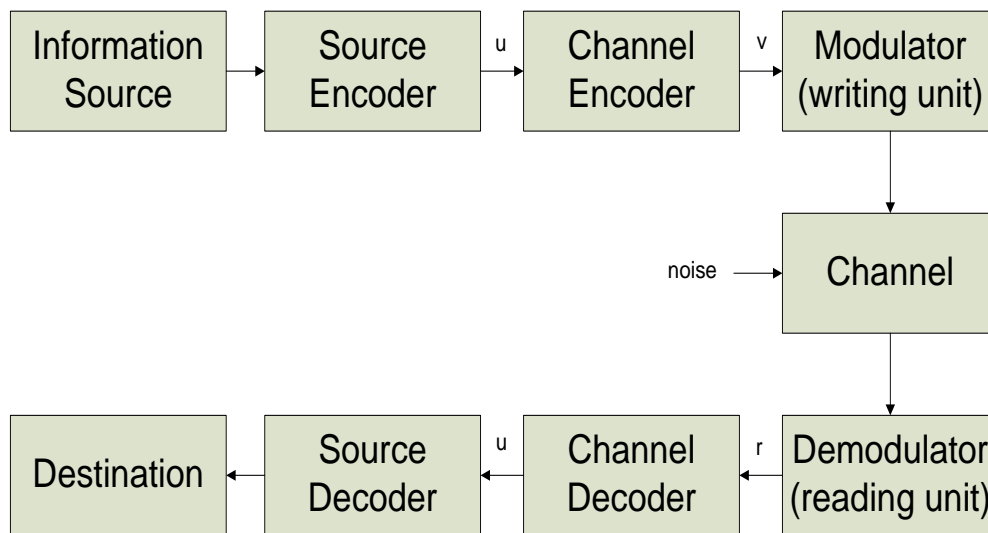
کانال مخابراتی محیطی الکتریکی است که مانند پلی بین منبع و مقصد عمل می‌کند. این کانال می‌تواند یک سیم، یک منبع رادیویی و ... باشد. هر کانال، مقداری تلفات و تضعیف، در انتقال دارد بنابراین عبور از کانال منجر به کاهش توان سیگنال ارسالی، می‌شود.

گیرنده روی سیگنال دریافت شده از کانال عمل کرده و سیگنال مناسب برای استفاده در مقصد را آماده می‌کند. گیرنده شامل تقویت کننده برای جبران تلفات انتقال، دمدولاسیون و کدگشا می‌باشد.

ارسال پیام به دو صورت آنالوگ و دیجیتال صورت می‌پذیرد که در این پایان‌نامه سیستم‌های مخابراتی دیجیتال مد نظر می‌باشند. در سیستم‌های ارتباطی دیجیتال، داده به صورت باینری مبادله می‌شود. این اطلاعات باینری به سیگنال‌های آنالوگ مدوله شده و از طریق کانال‌های ارتباطی ارسال می‌شوند. در این کانال‌ها به دلیل وجود نویز، اطلاعات مخدوش می‌شوند. در سمت گیرنده این اطلاعات مخدوش شده مجدداً توسط دمدولاتور، به حالت باینری تبدیل می‌شود که تا حدودی به اطلاعات اولیه فرستاده شده، شباهت دارد. خطای بیتی که در اطلاعات وجود دارد به نوع کانال ارتباطی بستگی دارد. در ارتباطات دیجیتال جهت کاهش این خطای بیتی و خنثی کردن اثر نویز موجود در کانال از روش‌های موجود برای کد کردن کانال استفاده می‌شود. این کار معمولاً با اضافه کردن بیت‌هایی به رشته بیت اصلی

در فرستنده انجام می‌گیرد. این بیت‌ها باعث می‌شوند تا در گیرنده عمل شناسایی و تصحیح خطا به درستی انجام شود.

یک سیستم مخابراتی نوعی در شکل ۱-۱ آمده است [۱]. ابتدا سیگنال اصلی وارد کد کننده اصلی می‌شود که در این قسمت عمل فشرده سازی سیگنال انجام می‌گیرد. رشته خروجی از این کد کننده، وارد کد کننده کانال می‌شود که بر حسب نوع این کد کننده یک سری بیت به آن اضافه می‌شود. این بیت‌های اضافه شده در گیرنده جهت شناسایی و تصحیح خطاهای احتمالی به کار می‌روند. بنابراین خروجی کد کننده کانال این قابلیت را پیدا کرده است که نویزهای اضافه شده در کانال را در گیرنده شناسایی و اصلاح کند. این خروجی در بلوک مدولاتور به مجموعه‌ای از موج‌های آنالوگ که مناسب جهت ارسال داده می‌باشند، تبدیل می‌شود. اطلاعات پس از عبور از کانال و اضافه شدن نویز به آن توسط بلوک دمدولاتور به حالت اولیه دیجیتال در می‌آید و سپس وارد بلوک کد گشای کانال می‌شود. این بلوک با توجه به مشخصات کد کننده، اقدام به شناسایی و اصلاح خطاهای موجود در پیام می‌کند. خروجی کد گشای کانال یک خروجی نرم^۱ یا سخت^۲ خواهد بود که وارد بلوک کد گشای اصلی می‌شود و در آنجا از حالت فشرده خارج شده و به حالت اولیه برمی‌گردد.



شکل ۱-۱ بلوک دیاگرام یک سیستم مخابراتی دیجیتال

^۱ Soft output

^۲ Hard output

۱-۲ آشنایی با نظریه کدینگ

با توجه به افزایش روز افزون نیاز به انتقال اطلاعات و ایجاد ارتباطات دقیق و با حداقل خطا، تلاش‌های زیادی برای بهینه‌سازی روش‌ها و تکنیک‌های تصحیح خطا صورت گرفته است.

فرکانس خطای احتمالی که بر روی اطلاعات رخ می‌دهد میزان قابلیت سیستم انتقالی را مشخص می‌کند و نرخ خطای بیتی (BER^1) خوانده می‌شود. این نرخ خطا تا حد زیادی به روش کدگذاری کانال بستگی دارد. کدگذاری کانال می‌تواند خطاها را به روش‌های گوناگون کنترل کند [۱]. به طور مثال روش دو طرفه ARQ^2 می‌تواند گزینه مناسبی برای کاهش نرخ خطا باشد. اما این روش منجر به ایجاد تأخیر زیادی می‌شود. بنابراین از روش FEC^3 برای حفاظت اطلاعات در برابر نویز کانال استفاده می‌شود. در روش FEC یک سری داده به اطلاعات اصلی اضافه می‌شود که برای تصحیح خطا به کار می‌روند. این روش نیاز به پهنای باند بیشتری برای انتقال اطلاعات دارد چون در این حالت اطلاعات ارسالی علاوه بر اطلاعات اصلی شامل اطلاعات اضافه شده نیز می‌باشند. روش‌های FEC منجر به مصرف توان بیشتر و افزایش پهنای باند می‌شوند که این امر در برخی کاربردها مشکل‌ساز می‌گردد.

همان‌طور که مشاهده می‌شود روش‌های مختلف دارای مزایا و معایب مختلف می‌باشند و در اینجاست که مهندسین باید با توجه به کاربرد مورد نظر تعاملی بین توان مصرفی، نرخ خطا و ... ایجاد کنند و کدگشا و کدکننده مناسب را انتخاب کنند.

کدکننده‌های موجود که قابل استفاده در زمینه FEC می‌باشند به دو دسته کلی کدهای بلوکی و کدهای کانولوشنی تقسیم می‌شوند.

در کدهای بلوکی هر چقدر که اندازه بلوک بزرگ‌تر باشد نرخ خطا کمتر می‌شود. در کدهای کانولوشنی یک بلوک n رقمی که توسط کدکننده تولید شده است نه تنها به اندازه هر بلوک بلکه به اندازه بلوک‌های قبلی نیز وابسته است. مطالعات نشان داده‌اند که کدهای کانولوشنی در بسیاری از کاربردهای تصحیح خطا بهتر از کدهای بلوکی عمل می‌کنند [۲] و [۳] و [۴].

¹. Bit Error Rate

². Automatic Repeat Request

³. Forward Error Correction

کدهای ذکر شده در بالا در مواردی که با نویز انبوه مواجه می‌شویم مناسب نمی‌باشند و برای مواردی که نویز کانال، تصادفی باشد مناسب هستند. به همین دلیل تکنیک‌های جدیدی برای مقابله با نویز انبوه ارائه شد از جمله استفاده از اینترلیورها^۱ و کدهای پیوسته^۲.

کدهای توربو که توسط برو^۳ و همکارانش [۵] معرفی شدند از نوعی کد کننده به هم پیوسته از نوع RSC^۴ تشکیل شده‌اند که توسط اینترلیور به هم وصل می‌شوند. کدهای توربو به عنوان موفق‌ترین الگوریتم تصحیح خطا شناخته شده‌اند [۵] و [۶] و در سال ۱۹۹۳ به عنوان کاربردی‌ترین الگوریتم در زمینه تصحیح خطا معرفی شدند. کدهای توربو به دو دسته کانولوشنی و بلوکی تقسیم می‌شوند.

۱-۳ کدهای بلوکی

کد بلوکی v [۷] از مجموعه مقادیری با طول ثابت تشکیل شده است. تعداد این کد واژه‌ها که همان تعداد عناصر در بردار v است با n نشان داده می‌شود. هر بلوک شامل دو بخش پیغام، که همان اطلاعات اصلی می‌باشد و با k نشان می‌دهیم، و قسمت اضافه شده به پیغام، که برای تصحیح خطا به کار می‌رود، می‌باشد. به طور مثال در یک کد بلوکی (۱۵و۱۰)، $n=25$ است که ۱۰ عنصر آن خود پیغام و مابقی، داده اضافه شده می‌باشند.

$$v = \{ \underbrace{01101}_{\text{redundant}} \underbrace{1011000101}_{\text{message}} \}$$

فرآیند تولید یک کد بلوکی توسط یک معادله ماتریسی مانند رابطه (۱-۱) نشان داده می‌شود:

فرض کنید که $u = \{u_0, u_1, \dots, u_{k-1}\}$ همان بردار داده باشد که قرار است کد شود. کد واژه خروجی به صورت رابطه (۲-۱) خواهد بود:

$$v = u \cdot g = \{v_0, v_1, \dots, v_{n-1}\} \quad (1-1)$$

$$= \{u_0, u_1, \dots, u_{n-1}\} \times \begin{bmatrix} g_{0,0} & g_{0,1} & g_{0,2} & \dots & g_{0,n-1} \\ g_{1,0} & g_{1,1} & g_{1,2} & \dots & g_{1,n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{k-1,0} & g_{k-1,1} & g_{k-1,2} & \dots & g_{k-1,n-1} \end{bmatrix} \quad (2-1)$$

^۱. Interleaver

^۲. Concatenated Codes

^۳. Berrou

^۴. Recursive Systematic Convolutional

که ۷ همان بردار کد واژه و g ماتریس مولد است که بر طبق اصول مشخص بدست می‌آید [۷].

۱-۴ کدهای کانولوشنی

کدهای بلوکی به خوبی در زمینه تصحیح خطا مورد استفاده قرار گرفتند اما در مواردی هم مشکل‌ساز بودند. برای کدگشایی این کدها نیاز بود تا تمامی کد دریافت شود تا بتوان شروع به کدگشایی کرد. این امر باعث به وجود آمدن تأخیر در کدگشایی این کدها می‌شود. از طرف دیگر عمل کدگشایی در این کدها با مقادیر سخت بود که این امر موجب کاهش قابلیت اعتماد به خروجی می‌شد که برای کدگشایی کانال‌های نویزی مناسب نبود و نکته منفی در به کار بردن این کدها به شمار می‌آمد. برای مقابله با این مشکل الیاس^۱ در سال ۱۹۹۵ کدهای کانولوشنی را معرفی کرد [۸]. کدهای کانولوشنی به جای دسته‌بندی اطلاعات و ارسال آن‌ها، کل اطلاعات را از تعدادی شیفت رجیستر عبور می‌دهند و افزونگی‌های لازم را اضافه می‌کنند. در نتیجه برای شروع عمل کدگشایی نیازی نیست تا کل اطلاعات دریافت شود و در این قسمت تأخیر کمتری خواهیم داشت. مزیت دیگر کدهای کانولوشنی این است که قابلیت کدگشایی با مقادیر نرم را نیز دارا می‌باشند. این مزایا با معرفی الگوریتم کدگشایی ویتربی در سال ۱۹۶۷ [۹] (که بر روی کدهای کانولوشنی اعمال شد و بهبود زیادی در نرخ خطای بیت داد) دو چندان شد و باعث پیشی گرفتن کدهای کانولوشنی از کدهای بلوکی شد. کدهای کانولوشنی توسط ناسا در فضاپیماهای وویجر^۲ و پایونیر^۳ مورد استفاده قرار گرفت [۱۰] و در نسل دوم ارتباطات سیار نیز به خوبی به نیازها پاسخ داد.

کدهای کانولوشنی در دو دسته سیستمی و غیر سیستمی وجود دارند که در نوع سیستمی پیغام اصلی نیز جزئی از کد واژه خروجی می‌باشد. این کار به بهبود در امر تصحیح خطا کمک می‌کند. ساختار کدهای کانولوشنی در فصل ۲ مورد بحث قرار می‌گیرد.

^۱.Elias

^۲.Voyager

^۳.Pioneer

۱-۵ کدهای به هم پیوسته

با توجه به ضعف کدهای بلوکی و کدهای کانولوشنی در مواجهه با نویز انبوه، کدهای به هم پیوسته برای رفع این مشکل معرفی شدند. کدهای به هم پیوسته توسط فورنی^۱ [۱۱] معرفی شدند که قابلیت ایجاد کدهایی با اندازه بلوک بزرگ و قابلیت تصحیح خطای زیادی دارند. روش کار بدین گونه است که از دو مرحله کد کردن استفاده می‌شود. کد کننده داخلی و کد کننده خارجی که توسط یک اینترلیور به هم متصل می‌شوند. اطلاعات ابتدا توسط یک کد کننده خارجی کد شده سپس خروجی این کد کننده به کد کننده بعدی فرستاده می‌شود. این دو کد کننده می‌توانند متفاوت باشند. عمل کدگشایی نیز به طور عکس انجام می‌شود. در شکل ۱-۲ حالت کلی یک کد کننده پیوسته سری نشان داده شده است [۷]. توربو کدها به عنوان دقیق‌ترین و مناسب‌ترین نوع از کدهای به هم پیوسته شناخته شده‌اند.



شکل ۱-۲ بلوک دیاگرام کد کننده به هم پیوسته

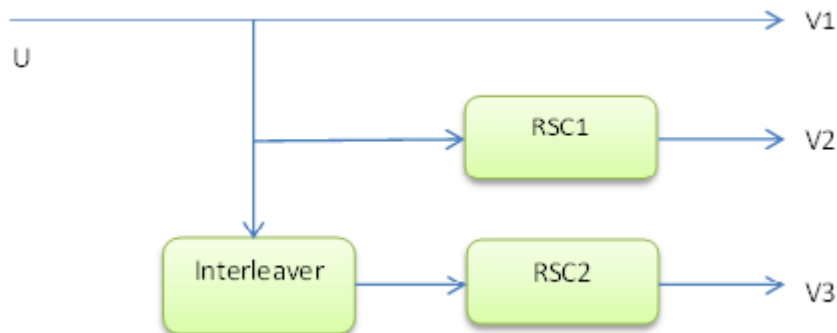
۱-۶ کدهای توربو^۲

کدهای توربو در سال ۱۹۹۳ به عنوان کاربردی‌ترین الگوریتم در زمینه تصحیح خطا معرفی شدند [۵] و [۶]. ارزش و اهمیت بالای این کدها در این است که ارتباطاتی با ضریب اطمینان بالا فراهم می‌کنند. به دلیل قابلیت بالا و بازدهی قابل توجه، توربو کدها به عنوان الگوریتمی مناسب برای کد کردن کانال در استانداردهای 3GPP و IEEE 802.16d و IEEE 802.16e مورد استفاده قرار گرفتند [12]. توربو کدها در کاربردهایی که به توان مصرفی پایین نیاز می‌باشد مورد استفاده قرار می‌گیرند از جمله ارتباطات فضایی و ماهواره‌ای.

¹ Forney

² Turbo Code

ساختار این کد کننده‌ها در شکل ۱-۳ آمده است. همان طور که مشاهده می‌شود از دو کد کننده کانولوشنی برگشتی (RSC^۱) تشکیل شده است که به طور موازی با هم قرار گرفته‌اند و توسط یک اینترلیوکننده از هم جدا شده‌اند. در فصل ۲ ساختار داخلی کدهای توربو مورد بحث قرار می‌گیرد.



شکل ۱-۳ کد کننده توربو مرتبه ۲

در این ساختار نرخ کد برابر $1/3$ می‌باشد که نرخ کد برابر است با تعداد المان ورودی به کد کننده تقسیم بر تعداد المان خروجی از کد کننده. اگر u_k را بیت ورودی به کد کننده در نظر بگیریم خروجی کد کننده ۳ بیت خواهد بود که v_1 همان بیت ورودی، v_2 بیت خروجی از کد کننده RSC1 و v_3 بیت خروجی از کد کننده RSC2 می‌باشد.

۱-۷ کدگشا

پس از معرفی کدهای کانولوشنی در سال ۱۹۵۵ [۸] روش‌های مختلفی برای فرایند کدگشایی کدهای کانولوشنی ارائه شد. همان طور که از شکل ۱-۴ مشخص است قسمت کدگشا در کدهای توربو از دو بلوک کدگشا تشکیل می‌شود که توسط اینترلیوکننده^۲ و داینترلیوکننده^۳ به هم متصل می‌شوند. این دو کدگشا دارای ورودی و خروجی نرم می‌باشند که اصطلاحاً به آن SISO^۴ نیز می‌گویند. خروجی هر کدگشا به عنوان ورودی به کدگشای دیگر می‌رود. به هر بار عمل کدگشایی توسط دو کدگشا یک چرخش^۵ گفته می‌شود.

^۱.Recursive Systematic Convolutional

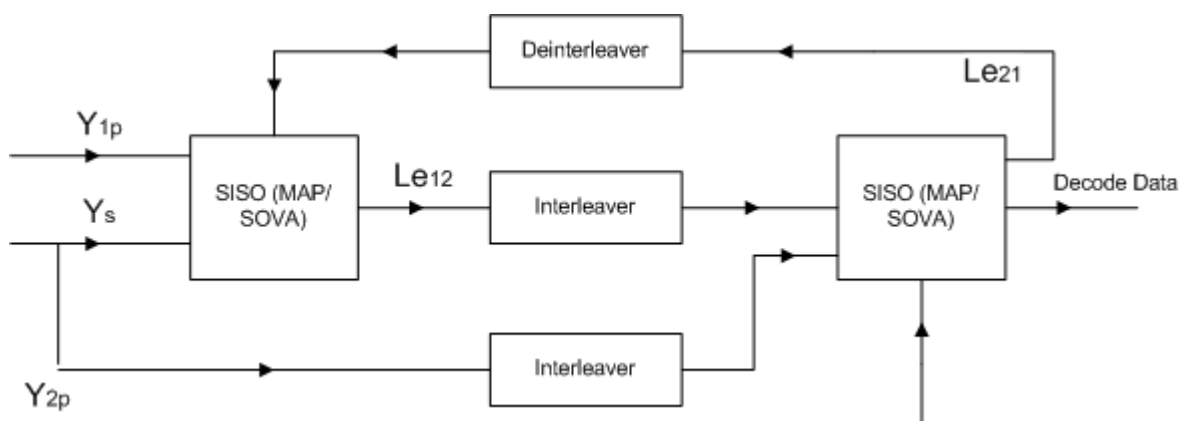
^۲.Interleaver

^۳.Deinterleaver

^۴.Soft Input Soft Output

^۵.Iteration

اکثر الگوریتم‌های معرفی شده در زمینه کد گشایی توربو، این قابلیت را دارند که ورودی نرم و سخت داشته باشند. در کد گشایی با ورودی سخت، ورودی به دو سطح کوانتیزه می‌شود و ورودی نرم به بیش از دو سطح. مقادیر نرم از نظر دقت و قابل اعتماد بودن مطلوب‌تر از مقادیر سخت می‌باشند.



شکل ۴-۱ بلوک دیاگرام یک کد گشایی توربو

این کد گشاهای متناسب با کد کننده‌ها طراحی می‌شوند. در سال ۱۹۶۷ ویتربی الگوریتم ویتربی را به عنوان روشی برای کد گشایی به روش حداکثر شباهت^۱ معرفی کرد [۹]. پس از آن فورنی^۲ و امرا^۳ ثابت کردند که الگوریتم ویتربی بهترین الگوریتم کد گشایی ML برای کدهای کانولوشنی می‌باشد [۱۳] و [۱۴]. در سال ۱۹۷۴ الگوریتم کد گشایی MAP^۴ توسط بل^۵ معرفی شد [۱۵] و برای کد گشایی در توربو کدها مورد استفاده قرار گرفت.

۸-۱ تشریح گرافیکی کد کننده‌ها

سه راه برای به نمایش در آوردن و درک بهتر چگونگی عملکرد کد کننده‌ها به صورت گرافیکی مرسوم است: ۱- دیاگرام حالت ۲- دیاگرام درختی ۳- دیاگرام شبکه. قبل از شروع بحث کد کننده‌ها، به معرفی دیاگرام‌های مذکور می‌پردازیم.

^۱.Maximum Likelihood (ML)
^۲.Forney
^۳.Omura
^۴.Maximum A-posteriori Probability
^۵.Bahl