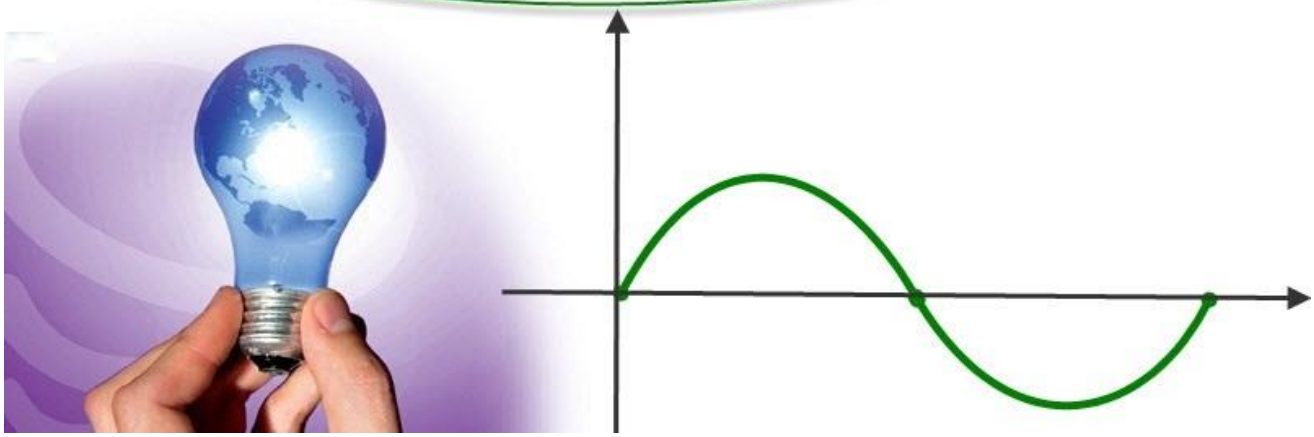
برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه



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

موضوع پروژه:

بررسی و شبیه سازی عملکرد کنترلر CAN، با استفاده از زبان توصیف سخت افزاری VHDL، و پیاده سازی آن بر روی FPGA

WikiPower.ir

برای خرید فایل word این پروژه اینجا کلیک کنید.
(شماره پروژه = ۵۱۶)
پشتیبانی: ۰۹۳۵۵۴۰۵۹۸۶

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

فهرست

عنوان	صفحه
چکیده	۱
فصل اول – مقدمه	۲
۱-۱- مقدمه	۳
۱-۲- معرفی CAN	۴
۱-۳- مقدمه ای بر تراشه های قابل برنامه ریزی	۶
۱-۴- مروری بر زبان های توصیف سخت افزاری	۸
۱-۵- نرم افزارهای طراحی تراشه های FPGA	۹
فصل دوم – مروری بر کارهای انجام شده	۱۰
۲-۱- مقدمه	۱۱
۲-۲- میکروکنترلر مقاوم شده در برابر تشعشع	۱۱
۲-۳- کانولوشن کننده های (Convolelrs) دو بعدی	۱۴
۲-۴- فیلترهای دیجیتال	۱۵
۲-۴-۱- فیلترهای با پاسخ ضربه محدود (FIR)	۱۵
۲-۴-۲- فیلترهای با پاسخ ضربه نامحدود (IIR)	۱۷
۲-۴-۳- فیلترهای Wavelet متقارن	۱۸
۲-۵- تبدیل کسینوسی گسسته و معکوس آن (IDCT,DCT)	۱۹
۲-۶- مبدلهای فضای رنگی ($YC_rC_b \rightarrow RGB, RGB \rightarrow YC_rC_b$)	۱۹
۲-۷- مدولاتور دیجیتال	۲۰
۲-۸- کنترلر گذرگاه USB	۲۱
۲-۹- کنترلر گذرگاه PCI	

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

۲-۱۰-کد کننده گفتار ITU-T G.723.1 ۲۳
۲-۱۱- کد کننده ها کدفایر ۲۴
۲-۱۲- پیاده سازی سخت افزاری الگوریتم های سطح بالای پردازش تصویر با استفاده از پیکر بندی جزئی FPGA در زمان اجرا ۲۴
۲-۱۳- مترجم های زبان های سطح بالا به زبان VHDL ۲۵
۲-۱۴- پیاده سازی یک پردازشگر تصویر قابل پیکر بندی مجدد ۲۶
۲-۱۵- جمع بندی ۲۸
۲۹

فصل سوم – کنترلر گذرگاه CAN ۳۱
۳-۱- مقدمه ۳۲
۳-۲- پایه های تراشه کنترلر CAN ۳۳
۳-۳- بررسی سخت افزار کنترلر CAN ۳۴
۳-۳-۱- شمارنده های خطا در کنترلر CAN ۳۴
۳-۳-۲- ثبات های کنترل ۳۶
۳-۳-۲-۱- ثبات فعال کننده وقفه ها ۳۶
۳-۳-۲-۲- ثبات وضعیت ۳۷
۳-۳-۲-۳- ثبات واسط CPU ۳۸
۳-۳-۲-۴- ثبات پیکربندی گذرگاه ۳۸
۳-۳-۲-۵- ثبات CIK out ۳۹
۳-۳-۳- واحد زمان بندی بیت ۴۰
۳-۳-۳-۱- سرعت نامی نرخ بیت ۴۰
۳-۳-۳-۲- ثبات صفر زمان بندی بیت ۴۱
۳-۳-۳-۳- ثبات یک زمان بندی بیت ۴۱
۳-۳-۴- ثبات ماسک توسعه یافته و استاندارد ۴۲
۳-۳-۵- بسته های پیام
۳-۳-۵-۱- میدان کنترل

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

۳-۳-۵-۲- میدان داوری یا شناسه ۴۲
۳-۳-۵-۳- میدان داده ۴۲
۳-۳-۵-۴- میدان ترکیب بندی ۴۵
۳-۳-۶- ثبات وقفه ۴۵
۳-۴- دریافت و ارسال پیام ۴۵
۳-۴-۱- انواع فریم های اطلاعات قابل مبادله بین گره ها و کنترلر ۴۵
۳-۴-۱-۱- فریم داده ۴۶
۳-۴-۱-۲- فریم دور ۴۶
۳-۴-۱-۳- فریم خطا ۴۶
۳-۴-۱-۴- فریم اضافه بار ۴۸
۳-۴-۲- بررسی کدهای خطا در تبادلات کنترلرCAN ۴۸
۴۹
۵۰
فصل چهارم – خلاصه ای از خصوصیات اصلی زبان VHDL ۵۱
۴-۱- مقدمه ۵۲
۴-۲- شی (object) ۵۲
۴-۳- عملگرهای زبان VHDL ۵۲
۴-۴- توصیف کننده های یک مولفه ۵۳
۴-۵- ساختارهای همزمانی و ترتیبی ۵۵
۴-۶- روشهای توصیف سخت افزار ۵۵
۴-۶-۱- روش توصیف ساختاری ۵۶
۴-۶-۲- روش توصیف فلوی داده (Data Flow) ۵۷
۴-۶-۳- روش توصیف رفتاری ۵۸
۴-۷- کد نویسی قابل سنتز ۵۹
۴-۸- جمع بندی

WikiPower.ir

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

۶۰

فصل پنجم – پیاده سازی کنترلر گذرگاه CAN ۶۱
۵-۱- مقدمه ۶۲
۵-۲-ثبات ارسال و دریافت پیام در کنترلر ۶۲
۵-۳- ثبات ماسک ۶۳
۵-۴- سیستم مقایسه شناسه ها ۶۴
۵-۵- افزایش تعداد بسته های پیام ۶۶
۵-۶- واحد محاسبه کننده کد CRC ۶۷
۵-۷- دیاگرام پایه های کنترلر طراحی شده و پیاده سازی دیکودر آدرس ۷۲
۵-۸- نرم افزار مورد استفاده در پیاده سازی کنترلر CAN ۷۳
۵-۹- جمع بندی ۷۳

فصل ششم – نتایج و جمع بندی ۷۴
۶-۱- مقدمه ۷۵
۶-۲- نتایج حاصل از تست وضعیتهای مختلف کنترلر ۷۵
۶-۳- نتایج حاصل از تست واحد CRC توسعه یافته ۷۸
۶-۴- نتایج حاصل از تست stuff bit ۸۰
۶-۵- ارسال فریم خطا ۸۲
۶-۶- بررسی وضعیت پایه فرکانس خروجی CLK out ۸۳
۶-۷- بررسی عملکرد حالت Sleep , pwd ۸۴
۶-۸- نتایج مربوط به پیاده سازی سخت افزار روی تراشه ۸۵
۶-۹- نتیجه گیری و پیشنهادات برای ادامه کار ۸۶
مراجع ۸۶

I

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

چکیده

یکی از موضوعات مطرح در اتوماسیون صنعتی و روباتیک تبادل اطلاعات بین اجزاء شبکه مانند *CPU* و فرستنده و گیرنده هایی است که نظارت و کنترل اجزاء یک سیستم را بعهده دارند از جمله زیر ساختهای لازم برای تبادل اطلاعات وجود شبکه ها و گذرگاه های تعریف شده و استاندارد برای اتصال اجزاء یک سیستم اتوماسیون صنعتی است شبکه کنترل محلی (*CAN-Control Area Network*) و گذرگاه آن مدتی است که در سیستمهای صنعتی مورد استفاده قرار گرفته است و تراشه های متعددی با عنوان کنترلر گذرگاه *CAN* مورد استفاده قرار می گیرد یکی از این محصولات تراشه ۸۲۵۲۷ اینتل می باشد که اخیرا مورد توجه طراحان شبکه های کنترل محلی قرار گرفته است .

از ابداعات جدید علم الکترونیک که امروزه کاربرد روزافزونی یافته است طراحی و پیاده سازی مدارهای دیجیتال و پردازنده های با کاربرد خاص بر روی تراشه های قابل برنامه ریزی *FPGA* است از مزایای مهم این نوع پیاده سازی طراحی مدارهای با قابلیت پیکربندی مجدد بر اساس خواست طراح است .

علاوه بر این در صورتی که تهیه یک تراشه با کاربرد خاص بنا به دلایل گوناگون از جمله عدم انتقال تکنولوژی مشکل باشد با داشتن و مشخصات کاری آن تراشه به این روش می توان تراشه مورد نظر را بر روی تراشه های قابل برنامه ریزی پیاده سازی نمود.

در این پروژه با استفاده از زبان توصیف سخت افزاری *VHDL* و تراشه های قابل برنامه ریزی به طراحی و پیاده سازی تراشه ۸۲۵۲۷ (کنترلر گذرگاه *CAN*) اقدام شده است در عین حال اصلاحاتی نیز در عملکرد این تراشه لحاظ شده که کارایی آن را بهبود می بخشد نتایج بدست آمده موفقیت این پروژه را در طراحی ، پیاده سازی و بهبود تراشه با انجام تغییرات پیشنهادی نشان می دهد .

WikiPower.ir

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

فصل اول

مقدمه

WikiPower.ir

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

۱-۱- مقدمه

در دو دهه گذشته پیشرفت روز افزون علم الکترونیک تحولات شگرفی را در کار آمدی سطوح فناوری باعث گردیده است به شکلی که این روند رو به افزایش در عرصه های گوناگون از قبیل ارتباطات، پزشکی، اتوماسیون، نظامی و ... کاملا مشهود است.

به عنوان نمونه در اتوماسیون صنعتی و روباتیک واحدهای الکترونیکی که بخش مهمی از سیستم بشمار می روند توانسته اند بهره وری سیستم را فزونی بخشند. از مسایل مطرح در این زمینه می توان طراحی و پیاده سازی شبکه های صنعتی را نام برد . از جمله این شبکه ها، (*Control Area Network*) *CAN* ، شبکه *Profibus* و شبکه *Ethernet* هستند. که هر یک از این شبکه ها در زمینه خاصی کاربرد دارند.

در شبکه های فوق و از جمله شبکه کنترل محلی (*CAN*) نیاز به تراشه های کنترل شبکه است که از نوع تراشه های خاص بوده و انواع متفاوتی از آنها توسط کمپانیهای سازنده به بازار عرضه شده اند. یکی از این محصولات تراشه ۸۲۵۲۷ اینتل می باشد که مورد توجه طراحان شبکه های کنترل محلی قرار گرفته است.

از دیگر ابداعات علم الکترونیک که امروزه کاربرد فراوان دارد طراحی و پیاده سازی مدارهای دیجیتال و پردازنده های با کاربرد خاص بر روی تراشه های قابل برنامه ریزی است. از مزایای مهم این نوع پیاده‌سازی مدارات دیجتال، طراحی مدارهای با قابلیت پیکربندی مجدد بر اساس خواست طراح است.

علاوه بر این در صورتی که تهیه یک تراشه با کاربرد خاص بنا به دلایل گوناگون از جمله عدم انتقال تکنولوژی مشکل باشد با داشتن مشخصات کاری آن تراشه به این روش می توان تراشه مورد نظر را بر روی تراشه های قابل برنامه ریزی پیاده سازی نمود.

در این پروژه با استفاده از یکی از زبانهای توصیف سخت افزاری و تراشه های قابل برنامه ریزی به طراحی و پیاده سازی تراشه ۸۲۵۲۷ (کنترلر گذرگاه *CAN*) اقدام شده است. در عین حال اصلاحاتی نیز در عملکرد این تراشه لحاظ شده که کار آیی آنرا بهبود می بخشد. در ادامه این فصل ابتدا به معرفی گذرگاه *CAN* می پردازیم. پس از آن مروری بر تراشه های قابل برنامه ریزی و در انتها هم مروری بر زبانهای توصیف سخت افزاری خواهیم داشت.

در فصل دوم مروری بر برخی از پیاده سازیها در ارتباط با طراحی و اصلاح پردازنده های عمومی و نیز پیاده سازی پردازشگرهای سیگنال دیجیتال خواهیم داشت. در فصل سوم یکی از پر کاربردترین کنترلرهای گذرگاه *CAN* و پروتکلهای ارتباطی در این شبکه را معرفی خواهیم نمود.در فصل چهارم به

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

معرفی یکی از زبان های توصیف سخت افزار که در این پروژه مورد استفاده قرار گرفته است می پردازیم. در فصل پنجم به پیاده سازی کنترلر معرفی شده می پردازیم. در انتها نتایج حاصل از پیاده سازی را نشان خواهیم داد و به جمع بندی خواهیم پرداخت.

۱-۲- معرفی *CAN*

شبکه کنترل محلی (*Control Area Network*) برای کنترل سیم بندی های ساده تا شبکه های پیچیده قابل استفاده بوده و از جمله موارد کاربرد این شبکه را می توان سیستمهای اتوماسیون صنعتی، وسائل و تجهیزات پزشکی، صنایع خودرو، هواپیما، کشتی سازی و ... را نام برد.

به عنوان مثال در اتومبیل های پیشرفته مانند مرسدس بنز برای متصل نمودن و در عین حال مدیریت بر واحدهای الکترونیکی بخشهای مختلف از قبیل موتور، درها، نمایشگرها و ... این شبکه استفاده می شود. در صنایع حمل و نقل ریلی مثل قطار و مترو نیز این شبکه برای کنترل اجزا مختلف سیستم مورد استفاده قرار می گیرد.

استانداردهای گوناگون با سرعتهای متفاوت که بر اساس قواعد *CAN* عمل می نمایند وضع گردیده است و هر یک از صنایع استفاده کننده از شبکه کنترل محلی یکی از این استانداردها را بکار می برند استفاده می نمایند.

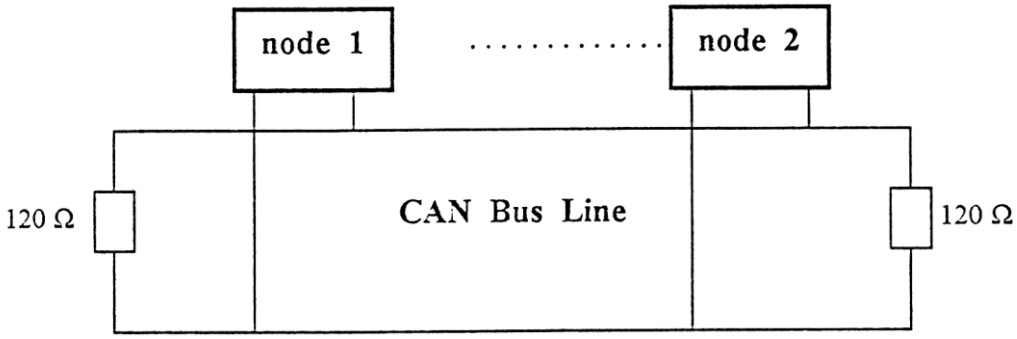
در این شبکه ارتباطات بین بخشهای مختلف به صورت سریال می باشد و اتصال اجزای مختلف این شبکه توسط گذرگاه سریالی که از پروتکلهای استاندارد *CAN* پیروی می کند صورت می گیرد.

بخشهای مختلف این شبکه عبارتند از :

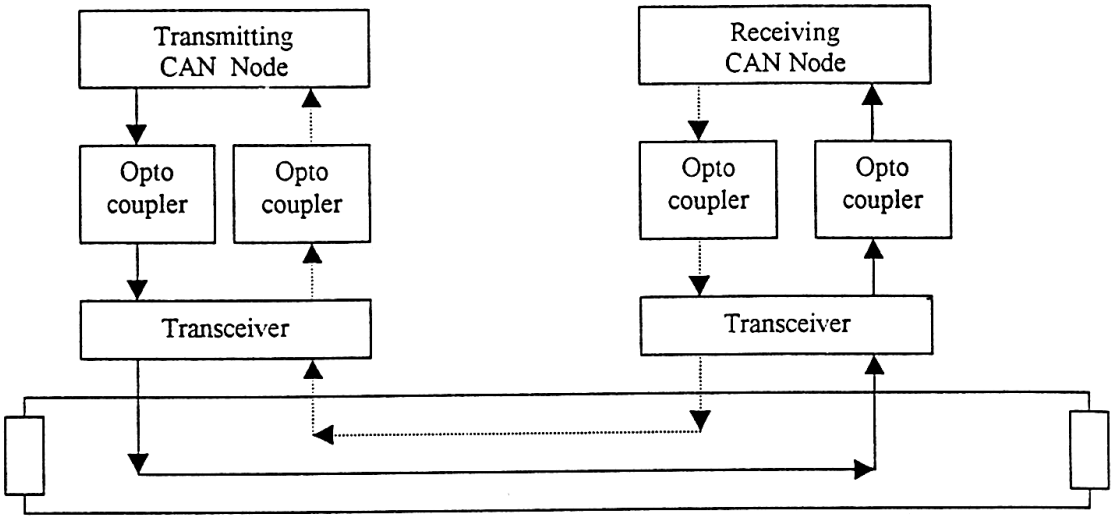
یک کنترلر شبکه، یک میکروکنترلر یا میکروپروسسور و تعداد لازم فرستنده – گیرنده پیام (*Transciver*) و ایزوله کننده های نوری (*Opto cupler*).

میکروکنترلر به منظور راه اندازی و فرمان دادن به کنترلر شبکه بکار گرفته می شود. در بعضی از شبکه‌های بزرگ بیشتر از یک کنترلر وجود دارد. شکل های (۱-۱) تا (۱-۳) نحوه اتصال اجزا مختلف این شبکه را نشان می دهند.

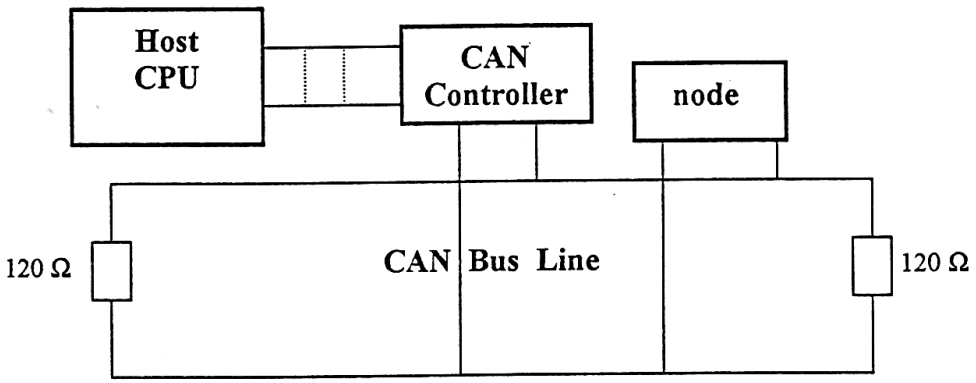
برای دریافت فایل wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه



شکل (۱-۱) نحوه اتصال گره ها در CAN



شکل (۱- ۲) نحوه اتصال فرستنده - گیرنده ها در CAN



شکل (۱- ۳) نحوه اتصال گره ، کنترلر و میکروکنترلر در CAN

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

تراشه های کنترلر زیادی از شرکتهای سازنده نیمه هادی به بازار عرضه شده است. به عنوان نمونه تراشه های ۸۲۵۲۶ و ۸۲۵۲۷ از شرکت اینتل تراشه ۸۲۲۰۰ از کارخانه فیلیپس از جمله تراشه های کنترلر عرضه شده می باشند. کنترلر ۸۲۵۲۷ در صنعت کاربرد بیشتری دارد زیرا این تراشه از بقیه کاملتر است. از دلایل دیگر کاربرد این تراشه شرکت سازنده آن است، به دلیل اینکه محصولات شرکت اینتل در ایران رایج تر می باشد.

کنترلر شبکه توسط گذرگاه سریال (*CAN Bus*) اطلاعات لازم را از گره های شبکه (فرستنده – گیرنده ها) دریافت نموده و در صورت لزوم در اختیار پردازنده سیستم قرار می دهد. همچنین با توجه به فرامین پردازنده سیستم پیامهائی را به گره های شبکه ارسال می کند.

تاریخچه تکامل شبکه کنترل محلی (*CAN*)

استفاده از شبکه داخلی در وسائط نقلیه توسط شرکت بوش (۱۹۸۳).

تصویب و معرفی قواعد مربوط گذرگاه *CAN* (۱۹۸۶).

ساخت و ارائه نخستین کنترلر گذرگاه *CAN* (۱۹۸۷).

تصویب و ارائه نسخه دوم قواعد ارتباطات در گذرگاه *CAN* توسط شرکت بوش (۱۹۹۱).

معرفی پروتکل (قواعد) لایه های بالائی در شبکه مذکور (۱۹۹۱).

تشکیل و معرفی هیات کاربران *CiA* (*CAN in Automation*) (۱۹۹۲).

ارائه قواعد مربوط به لایه کاربردی شبکه توسط *CiA*(۱۹۹۲).

تولید اولین اتومبیل از کارخانه مرسدس بنز که از *CAN* استفاده نمود(۱۹۹۲).

تصویب استاندارد *ISO 11989* مربوط به *CAN* (۱۹۹۳).

برگزاری اولین کنفرانس *CAN* توسط انجمن *CiA* (۱۹۹۴).

ارائه اصلاحات استاندارد *ISO 11898* (۱۹۹۵)..

توسعه قواعد ارتباطات در *CAN* (۲۰۰۰). [۱] [۲] [۳] [۴] [۵].

۱-۳- مقدمه ای بر تراشه های قابل برنامه ریزی

مدارهای مجتمع با کاربرد خاص *ASIC* (*Application Specific Integrated Ciccuits*) به سه دسته تقسیم می شوند:

الف – مدارهای کاملا خاص *Full Custom* .

ب- مدارهای نیمه سفارشی *Semi – Custom* .

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

ج- مدارهای قابل برنامه ریزی *PLD* و *Gate Array* .

در مورد اول طرح مدار در حد گیت ها و ترانزیستورها توسط نرم افزارهای شبیه سازی آماده شده و تست می گردد. این مشخصات در اختیار سازنده قرار می گیرد و سازنده این طرح را به صورت ترانزیستوری در آورده و آن را بر روی سطح سیلیکون خام پیاده می نماید.

در مورد دسته ب همانند حالت قبل مدار با کمک نرم افزار ویژه طراحی شده و سپس نتیجه طراحی جهت ساخت در اختیار سازنده قرار می گیرد با این تفاوت که این بار مدار بر روی ترا شه های نیمه خام گیت ها مثل *AND* و *OR* یا مدارهای دیگر بر روی آنها از قبل ایجاد گردیده و فقط اتصالات بین آنها وجود ندارد که اینها اتصالاتی هستند که با توجه به طرح مشتری بر روی تراشه ایجاد می گردند. یک نمونه این مدارها *Gate Array* می باشد.

مدارهای قابل برنامه ریزی آنهائی ه ستند که بر خلاف د سته اول ساخت آنها در کارخانه سازنده و ق سمتهای تکمیلی آن تو سط خود م شتری انجام می شود در صورتیکه در مدارهای نیمه خام و ق سمت آخر یعنی ایجاد ات صالات تو سط سازنده انجام گرفته و تنها ق سمت میانی یعنی طراحی ات صالات تو سط مشتری قابل انجام است. مدارهای قابل برنامه ریزی شامل انواع *PLA*، *FPGA* و حتی *PROM* ، *EPROM* نیز می شوند.

تراشه های *PGA* در دو دهه گذشته محبوبیت زیادی کسب نموده اند. در این تراشه ها هیچ نوع الگوی ارتباطی از قبل تعیین شده ای وجود ندارد و بنابراین طراح آزادی کامل در جهت ایجاد این اتصالات را دارد. بر اساس نوع برنامه ریزی این تراشه ها در دو دسته تقسیم بندی می شوند :

الف- نوعی که وارد کردن برنامه در آنها بایستی در کارخانه سازنده انجام گیرد *Masked PGA* .

ب- تراشه های قابل برنامه ریزی در محل .

FPGA از لحاظ نوع ساختار ساده ترین و قدیمی‌ترین آنها شامل ماتریسی از *NAND* یا *NOR* های دو وروی می با شد. در ترا شه های د سته (الف) همانند د سته (ب) بخش اول کار یعنی ساخت ترا شه یا مرحله ایجاد اتصالات در کارخانه سازنده انجام می شود سپس طراح با توجه به طرح خود ارتباط بین دریچه ها را تعیین نموده و پس از شبیه سازی مدار و اطمینان از عملکرد صحیح آن اطلاعات مربوط به ات صالات مورد نظر را به صورت یک برنامه کامپیوتری تحویل کارخانه سازنده می دهد تا ما سکهای مورد نیز تهیه گشته و بقیه مراحل تا پایان کار و تکمیل ساخت تراشه توسط سازنده انجام پذیرد در دسته (ب) بخش ایجاد اتصالات هم توسط کاربر و با کمک دستگاه برنامه ریزی ویژه در محل کار طراح قابل انجام است و احتیاج به مراجعه به سازنده برای انجام این کار نیست.

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

بایستی توجه داشت که دسته (ب) در ازای داشتن این مزیت این نقطه ضعف را نیز دارد که بخش مهمی از سطح سیلیکون برای ایجاد امکانات برنامه ریزی در محل استفاده شده است. بنابراین در نوع اخیر از درصد کمتری از سطح سیلیکون برای ایجاد اصل مدار استفاده می شود.یکی از نقاط ضعف عمده تراشه‌های *Masked PGA* در آن است که اگر اشکالی پس از اتمام کار در تراشه ساخته شده کشف شود در آن صورت تمام مراحل تهیه ماسک و ایجاد اتصالات توسط سازنده بایستی دو مرتبه از ابتدا تکرار گردند این موضوع هزینه جبران اشتباه را خیلی بالا می برد نکته دیگری که در استفاده از این نوع تراشه بایستی در نظر داشت این است که تعداد تراشه مورد نیاز باید انقدر زیاد باشد که جبران هزینه های تولید ماسک و غیره را بنماید. همچنین زمان قابل ملاحظه برای انجام این مراحل را توسط کارخانه سازنده نیز بایستی در نظر داشت.

نمونه ای از موارد کاربرد تراشه های *FPGA* :

- رمز گذاری و رمز گشایی داده ها .
- پردازنده های محاسباتی.
- پردازنده های تصویری و صوتی (فشرده سازی، بازسازی، فیلتر و ....)
- کنترلرهای گذرگاه های *PCI* ، *CAN* و ... [۶] [۷]

WikiPower.ir

۱-۴- مروری بر زبان های توصیف سخت افزاری

HDL (*Hardware Description Language*) روش توسعه یافته‌ای از توصیف رفتار سیستم‌های منطقی به وسیله روابط منطقی است. این زبان ها بسیاری از مشخصه های روابط منطقی و روابط حالت را در درون خود دارند. در این قسمت، بیشتر تمرکز ما بر روی *VHDL* است.

VHDL(VHSIC(Very High Speed Integrated Circuit)Hardware Description Language)

امروزه این زبان به عنوان استاندارد صنعتی *MIL SID 454L* معرفی شده است و تمامی طرح های *ASIC* مربوط به وزارت دفاع آمریکا باید طبق این زبان استاندارد نوشته شوند.

این زبان به عنوان قسمتی از پروژه *VHSIC* (مدارهای مجتمع با تکنولوژی مشخصی، تعریف و شبیه سازی نمود. زمانی که یک مدار منطقی بوسیله این زبان تعریف می گردد، می توان ان را به هر پروسه منطقی و یا بر روی ماژول های طراحی شده توسط هر یک از تولید کننده های ابزارهای منطقی انتقال داد. *VHDL (VHSIC HDL)* یک سیستم منطقی را به صورت ساختار بالا به پایین توصیف می کند.

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

برای بد ست آوردن تو صیفی از یک سیستم به صورت ساختار بالا به پایین، سیستم را به صورت مجموعه ای از زیر سیستم ها تقسیم می کنیم که بوسیله یک سری رابط به هم متصل می گردند. هر یک از زیر سیستم های بالایی را می توان به توابع و زیر سیستم های کوچکتر تقسیم کرد. این عمل همچنان ادامه می یابد تا به پایین ترین سطح از سیستم دست بیابیم که در این سطح هر یک از زیر سیستم ها را می توان بوسیله گیتها و بخشهای آماده دیگر طراحی نمود.

به این ترتیب، بدلیل آنکه هر یک از طبقات این ساختار منطقی به صورت یکتا مشخص شده اند.

هر یک از آنها را می توان به تنهایی شبیه سازی نمود و تابع منطقی اجرا شده بوسیله آنها را آزمایش کرده و خطاهای احتمالی را بر طرف نمود. ابتدا صحت عملکرد پایین ترین طبقه این سیستم را آزمایش کرده و با ترکیب زیر سیستم های پایین تر به زیر سیستم های پیچیده تر می ر سیم تا جاییکه به طرح سیستم مورد نظر که در بالاترین طبقه این ساختار وجود دارد برسیم.

پس از انجام این عمل، به مرحله ترکیب می رسیم که در آن کل طرح را پیاده کرده و سپس برای بدست آوردن پارامترهای زمانی آن، عمل شبیه سازی را انجام می دهیم. [۷]

۱-۵- نرم افزارهای طراحی تراشه های *FPGA*

از جمله شرکتهای فعال در زمینه نرم افزارهای طراحی تراشه های *FPGA* می توان از *ALTERA* نام برد که عر ضه کننده مجموعه نرم افزار *MAXPlus* می با شد. این مجموعه به عنوان ورودی خود تو صیف مدار را به زبانهای عمومی *VHDL* و *Verilog HDL* و یا زبان اختصاصی *ALTERA* یعنی *AHDL* می پذیرد. از بین دیگر فعالان این زمینه می توان از شرکتهای *ACTEL* ، *ATMEL* و *Xilinx* نام برد.

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

فصل دوم

مروری بر کارهای انجام شده

WikiPower.ir

۲-۱- مقدمه

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

در این فصل مروری بر استفاده از زبان های توصیف سخت افزاری و تراشه های *FPGA* در طراحی پردازنده ها فیلترها و دیگر مدارات دیجیتالی خواهیم داشت.

بسیاری از پردازنده ها، پردازشگرهای با کاربرد خاص، پردازشگرهای کمکی (*Coprocessor*)، کنترلرهای گذرگاه های مختلف اعم از *USB*، *PCL* و .... را می توان به صورت بهینه بر روی تراشه های *FPGA* پیاده سازی نمود. علاوه بر این می توان در صورتی که نیاز باشد اصلاحاتی بر روی آنها انجام داد و مطابق با شرایط لازم طراحی را تغییر داد. به عنوان نمونه میکرو کنترلر مقاوم شده در برابر تشعشع از جمله پردازنده های اصلاح شده است که در بخش اول به آن اشاره خواهیم نمود. البته لازم است به ساختمان پردازنده ای که می خواهیم آن را اصلاح نمائیم آگاهی کامل داشته باشیم.

علاوه بر این مدارهای ویژه مانند فیلترهای دیجیتال از قبیل *FIR* ، *IIR* ، *Wavelet* ، فیلترهای غیر خطی، وفقی با استفاده از زبان *VHDL* و تراشه های *FPGA* قابل ساخت هستند. اکثر توابعی که در الگوریتم های پردازش سیگنال اعم از صوت و تصویر مانند مبدلهای *DCT* ، *FFT* سیستم های تعیین موقعیت، آشکارسازهای لبه تصویر و ... مورد استفاده قرار می گیرند را می توان به این روش طراحی نمود. کنترلرهای وقفه، *DMA* و انواع کنترلرهای گذرگاه ها نیز به صورت بهینه شده و اصلاح یافته بر روی تراشه های *FPGA* قابل پیاده سازی هستند.

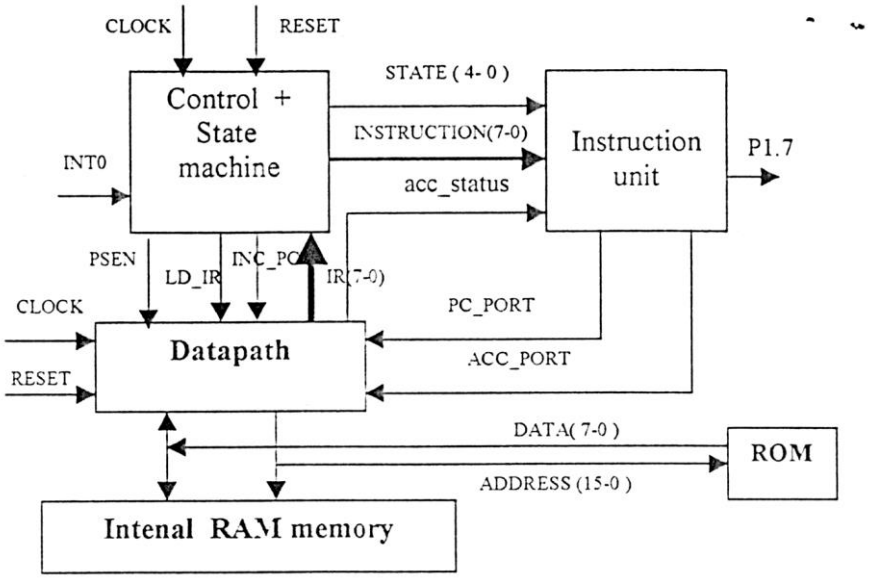
برخی از دانشمندان مترجم هائی تهیه نموده اند که با استفاده از این مترجم ها می توان الگوریتم های پردازش تصویر و صوت را که با استفاده از زبانهای سطح بالا مانند *C* و یا محیط های شبیه سازی مانند مطلب تهیه گشته اند به زبان *VHDL* تبدیل نمود. در بخش آخر به معرفی یکی از این مترجم ها خواهیم پرداخت.

۲-۲- میکرو کنترلر مقاوم شده در برابر تشعشع

مقاومت در برابر خطا(*Fault - Tolerance*) و قابلیت اعتماد بالا از مشخصات ضروری صنایع نظامی می باشند. مدارات دیجیتال بکار گرفته شده در صنایع هوا فضا تحت تاثیر انواع تشعشعات قرار می‌گیرند. یک روش کاهش خطا به منظور مقاوم شدن در برابر تشعشع استفاده از تکنیک کد همینگ (*Hamming*) می باشد.

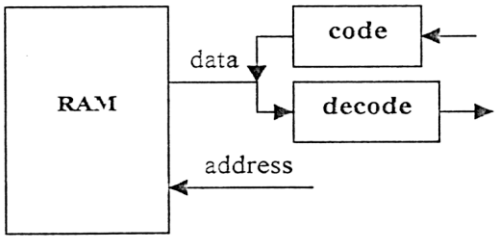
یکی از میکرو کنترلرهای دیجیتالی که بیشتر در این صنایع بکار گرفته می شود خانواده میکرو کنترلر ۸۰۵۱ است. این تراشه را که بلوک دیاگرام ساختمان آن در شکل (۲-۱) نشان داده شده است با استفاده از زبان توصیف سخت افزاری *VHDL* می توان بوسیله تکنیک کد همینگ نسبت به تشعشع مقاوم نمود.

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

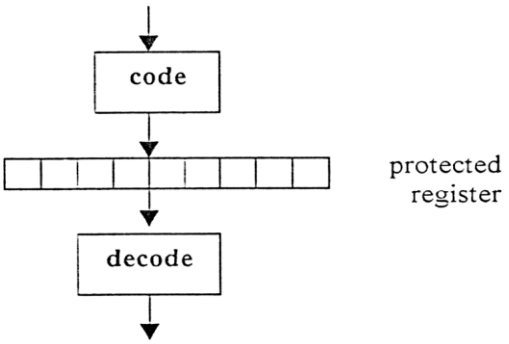


شکل(۲-۱) بلوک دیاگرام ساختمانی میکرو کنترلر ۸۰۵۱

همان طور که در شکلهای (۲-۲) تا (۲-۵) نشان داده شده است این تکنیک به رجیسترهای مسیر داده، واحد کنترل، ماشین حالت و حافظه قابل اعمال می باشد.

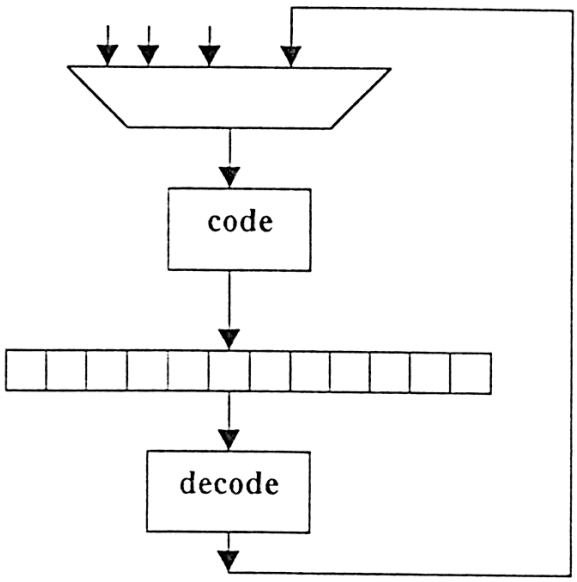


شکل (۲-۲) اعمال همینگ کد به حافظه

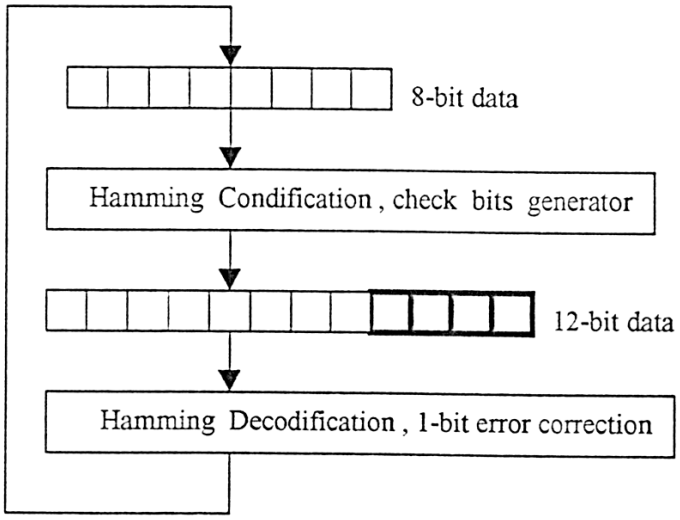


شکل (۲-۳) اعمال کد همینگ به واحد کنترل و ماشین حالت

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



شکل (۲- ۴) رجیسترهای مسیر داده مقاوم شده



شکل (۲- ۵) اعمال تکنیک کد همینگ

نتایج حاصل از اعمال این تکنیک در جدول (۲-۱) نشان داده شده است.

8051-8bit	*Full protected*	*909 CLBs used*

جدول (۲-۱) نتایج حاصل از اعمال همینگ کد

ملاحظه می شود که پیاده سازی ۸۰۵۱ مقاوم شده با استفاده از *CLB* ۹۰۹ (*Configurable Logic Block*) قابل انجام م باشد. این روش را می توان بر روی بسیاری از میکرو کنترلرها و میکروپروسسورها اعمال نمود[۸].

۲-۳- کانولوشن کننده های (*Convolvers*) دو بعدی

کانولوشن کننده های دو بعدی در پردازش تصویر به منظور اعمال فیلتر، تشخیص لبه (*Edge Detection*) ، درون یابی (*Interpolation*) و واضح نمودن (*Sharpening*) بسیار موثر هستند به عنوان نونه، عمل یک کانولوشن کننده که لبه های یک تصویر را برای خروجی بر جسته و واضح (*Sharpen*) می نماید *Edge Enhancement* نامیده می شود.

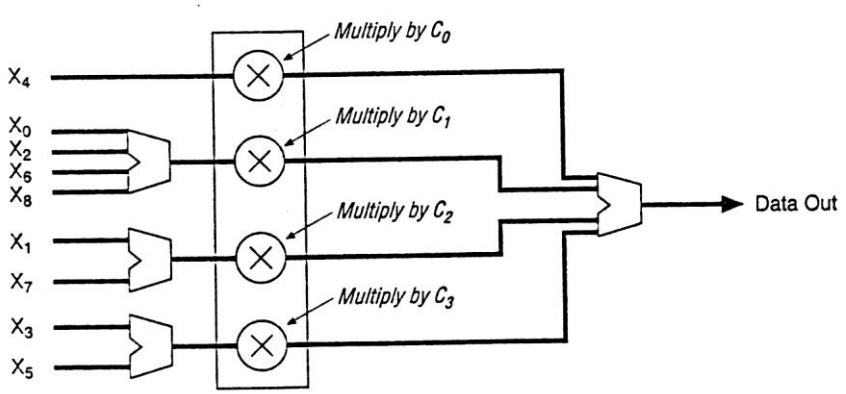
کانولوشن کننده داده های تصویری را در قطعات کوچک (مثلا پنجره های ۳×۳) همانند شکل (۲-۶) مورد پردازش قرار می دهد.

۱	۲	۳
۴	۵	۶
۷	۸	۹

WikiPower.ir

شکل (۲-۶) پنجره ۳×۳ برای عمل کانولوشن

بلوک دیاگرام یک کانالوشن کننده دو بعدی در شکل (۲-۷) نمایش داده شده است . [۹] [۱۰]



شکل (۲-۷) بلوک دیاگرام یک کانولوشن کننده دوبعدی

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

در شکل فوق x نمایشگر یک پیکسل از تصویر می باشد.

۲-۴- فیلترهای دیجیتال

فیلتر های دیجیتال از پر کاربردترین اجزا در پردازش سیگنال های دیجیتال می باشند. کار یک فیلتر حذف قسمتهای نامطلوب یک سیگنال یا استخراج سیگنال هایی در محدوده فرکانسی خاص می باشد. به عبارت دیگر فیلتر فرکانس های مشخصی از یک سیگنال را انتخاب و سپس حذف و یا تغییر می دهد. این کار به منظور کاهش نویز و یا شکل دادن به طیف سیگنال انجام می گیرد.

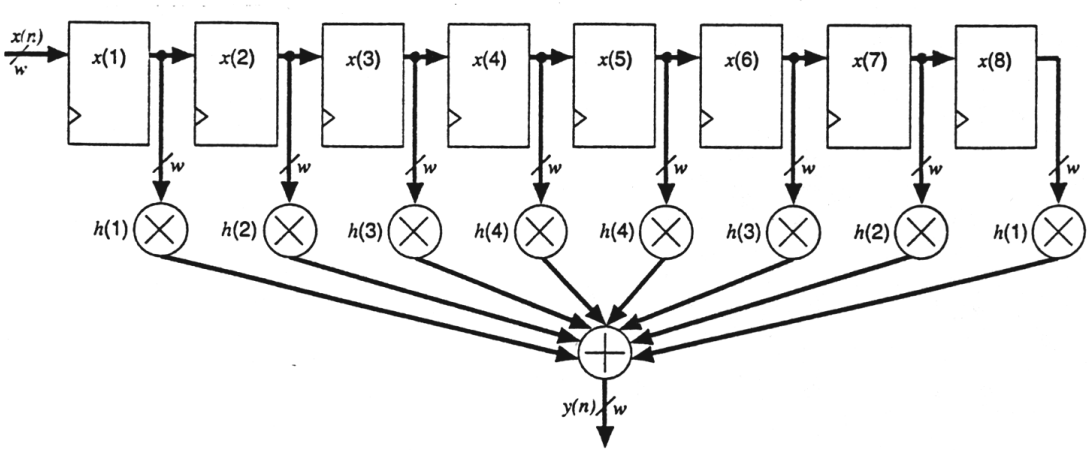
بیشتر فیلترهای قدیمی در کاربردهای *DSP* با بکارگیری پردازنده های *DSP* ویژه پیاده سازی می‌شدند. این پردازنده های *DSP* قادر به انجام عملیات ضرب و ذخیره اطلاعات با سرعت بالا هستند ولی دارای محدودیت در پهنای باند می باشند. فقط تعداد معینی عملیات قبل از ورود نمونه بعدی توسط این پردازنده ها قابل انجام می باشند که در نتیجه محدود کننده پهنای باند است. پردازنده های *DSP* به صورت ذاتی ترتیبی می باشند و بنابراین *DSP* هایی که از یک پردازنده بهره می برند قادر به انجام یک عمل بر روی یک مجموعه داده در هر زمان می باشند. این مسئله باعث محدودیت در فرکانس کلی سیستم می شود. فیلترهای بر پایه *FPGA* با معماری خطی لوله ای موازی پیاده سازی می شوند که باعث افزایش عملکرد کلی سیستم می گردد. پیاده سازی با *FPGA* همچنین امکان ارزیابی دقیق در تمام مراحل الگوریتم را امکان پذیر می سازد. موارد ذکر شده عمده ترین تفاوت های بین یک فیلتر بر پایه *FPGA* با متناظر *DSP* آن می‌باشد.

پیاده سازی فیلترهای دیجیتال با فرکانس نمونه برداری چند مگا هرتز با بکارگیری *DSP* های استاندارد غالبا دشوار است و گران تمام می شود. امکان بالقوه پردازش موازی و برنامه ریزی مجدد *FPGA* ها را به یک راه حل ایده آل تبدیل می کند. قابلیت برنامه ریزی مجدد. تغییر در فیلتر در هر زمان را امکان پذیر می سازد [۱۱].

۲-۴-۱- فیلترهای با پاسخ ضربه محدود(*FIR*)

فیلترهای با پاسخ ضربه محدود (*FIR*) در طراحی فیلتر پایین گذر، کانولوشن تصویر، انتخاب باند، عدم همپوشانی(*Anti Aliasing*)، تخمین، درون یابی و بسیاری دیگر از سیستم های پردازش سیگنال دیجیتال (*DSP*) کاربرد فراوان دارند. تراشه های *FPGA* برای پیاده سازی فیلترهای *FIR* بسیار ایده آل هستند. معماری یک فیلتر *FIR* در شکل (۲-۸) نشان داده شده است.

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



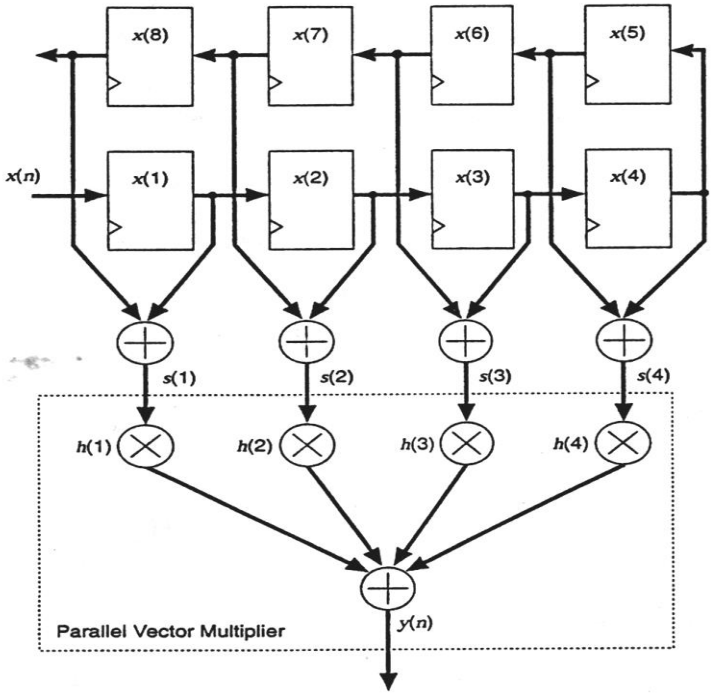
شکل (۲-۸) معماری یک فیلتر FIR

فیلتر صفحه قبل دارای هشت رجیستر هشت بیتی است. خروجی هر رجیستر را با $x(n)$ نشان می‌دهیم. معادله این فیلتر عبارت است از : $y(n)=\sum_{n=1}^{8} x(n)h(n)$

یک فیلتر *FIR* با پاسخ فاز خطی، دارای ضرائب متقارن است. این تقارن اجازه می دهد که $x(n)$ های متقارن با یکدیگر جمع شود قبل از اینکه آنها بوسیله ضرایب ضرب شوند. در این صورت طراحی به صورت شکل (۲-۹) حاصل می شود [۱۲] [۱۳].

WikiPower.ir

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



شکل (۲-۹) معماری یک فیلتر FIR با پاسخ فاز خطی

نتایج حاصل از پیاده سازی در جدول (۲-۲) نشان داده شده است.

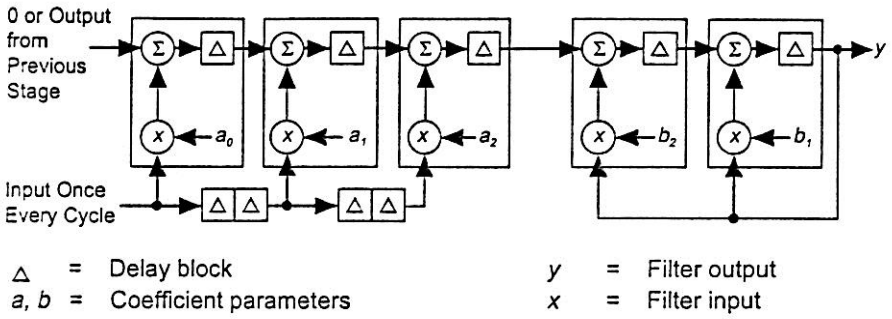
Taps	Device	Speed (MSPS)	Logic Cells
16	EPF8820A	101	468

جدول (۲-۲) نتایج حاصل از پیاده سازی فیلتر FIR

۲-۴-۲- فیلتر با پاسخ ضربه نامحدود (*IIR*)

این فیلتر در طیف وسیعی از سیستم های پردازش سیگنال استفاده می گردد از جمله کاربردهای این فیلتر در پردازش صوت و تصویر دیجیتال، تعیین وضعیت (*Conditioning*) و فیلتر نمودن کانال انتخابی است.

بلوک دیاگرام فیلتر *IIR* در شکل (۲-۱۰) نشان داده شده است.



شکل (۲-۱۰) بلوک دیاگرام فیلتر IIR

فیلتر *IIR*، *biquad* مرتبه دوم با معادلات زیر را می توان برای ساختن فیلترهای *IIR* مرتبه بالاتر استفاده نمود.

$$y_t = \sum_{i=0}^{2} a_i x_{t-i} + \sum_{j=1}^{2} b_j y_{t-j}$$

خروجی فیلتر y

ورودی فیلتر x

پارامترهای ثابت a , b

نتایج حاصل از پیاده سازی در جدول (۲-۳) نشان داده شده است. [۱۴] [۱۵]

WikiPower.ir

Maximum Sample rate (MSPS)	*30*
Percentage of EPF 10k50Device Utilized	*29 %*

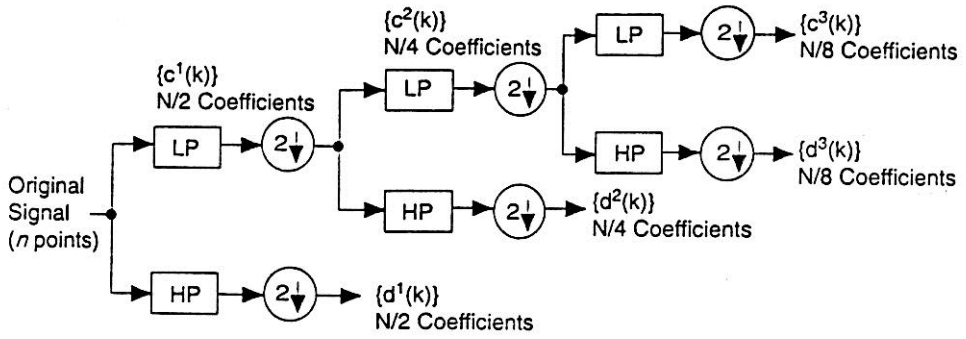
جدول (۲-۳) نتایج حاصل از پیاده سازی فیلتر *IIR*

۲-۴-۳- فیلترهای *Wavelet* متقارن

با استفاده از توابع *Wavelet* سیگنالهای داده به مولفه های فرکانسی متفاوت تجزیه می گردد. این توابع برای نمایش و تقریب سیگنالهای غیر پیوسته کاربرد دارند. ضرایب *Wavelet* را با استفاده از تعداد زیادی فیلتر بالاگذر و پایین گذر می توان محاسبه نمود. مطابق شکل (۲-۱۱) در هر مرحله فیلتر پاین گذر سیگنال را هموارتر می سازد و فیلتر بالاگذر کلیه اطلاعات سیگنال را به صورت مقیاس بندی شده

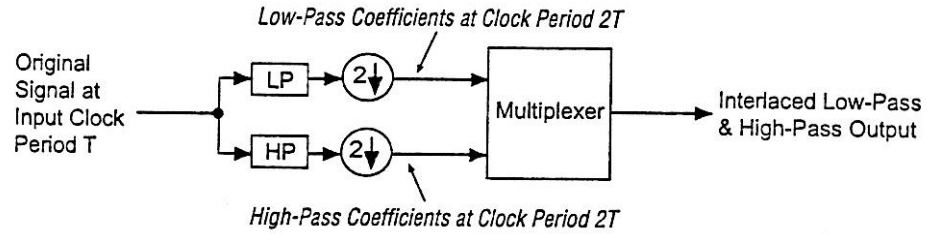
ارائه

می نماید.



شکل (۲-۱۱) بلوک دیاگرام یک فیلتر Wavelet

در مواردی که توابع *Wavelet* برای فشرده سازی تصویر بکار می رود فیلترهای *Wavelet* متقارن مناسب هستند زیرا این فیلترهای در لبه های تصویر اعوجاج کمتر و فشرده سازی بالاتری را فراهم می‌نمایند. یکی از انوع فیلترهای متقارن که کاربرد بیشتری دارد در شکل (۲-۱۲) نشان داده شده است.



شکل (۲- ۱۲) بلوک دیاگرام یک فیلتر Wavelet متقارن

یک سیگنال ورودی در یک فرکانس سات F_0 بکار رفته و خروجیهای بالا گذر و پایین گذر تخمین زده شده در فرکانس ساعت $f_0/2$ محاسبه می گردد سپس یک سیگنال خروجی شامل ضرایب بالا گذر و پایین گذر تولید می نماید [۱۶] [۱۷] .

Data Length	*Device*	F_{max}	*Logic Cells*
16 bit	*EPF10K50-4*	*68 MHZ*	*1776*

۲-۵- تبدیل کسینوسی گسسته و معکوس آن (*IDCT , DCT*)

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

مبدل های کسینوسی گسسته و معکوس آن را در الگوریتمهای فشرده سازی تصویر می توان بکار گرفت. این مبدلها را می توان برای سیستم هایی که به حد اقل عرض باند انتقال و توان مصرفی کم و سطح بالائی از کارآئی را نیاز دارند استفاده نمود. تلفن تصویری، سیستم های چند رسانه ای (مالتی مدیا)، سیستم‌های پخش تصویر با استاندارد های *H.261 , H263 , JPEG , MPEG -1 , MPEG-2* بر اساس تبدیل کسینوسی گسسته عمل می نماید.

نتایج حاصل از پیاده سازی عملگر *DCT* برای یک بلوک تصویری ۸×۸ در جدول (۲-۵) نشان داده شده است [۱۸].

Logic Elements	*Performance (f_{max})*
4386	*17.45MHZ*

جدول (۲-۵) نتایج حاصل از پیاده سازی *DCT*

۲-۶- مبدلهای مختلف مورد استفاده برای ذخیره و نمایش تصاویر رنگی را به یکدیگر تبدیل می کنند. دو قالب متداول عبارتند از:

۱- سبز قرمز آبی (*RGB*)

۲- شدت مولفه رنگ قرمز مولفه رنگ آبی (*YUV*)

قالب *YUV* که در آن *Y* شدت روشنایی نقات تصویر و *U* و *V* به ترتیب مولفه های قرمز و آبی (C_bC_r) یا *Chrominance* می باشند در پخش تصاویر تلویزیونی مورد استفاده هستند در حالیکه فرمت *RGB* در تصاویر کامپیوتری بیشتر استفاده می شوند.

با استفاده از معادلات زیر بندیلات فوق انجام می پذیرد:

$$Y' = 0.257R' + 0.504\ G' + 0.098\ B' + 16$$
$$C_r = 0.439R' - 0.368\ G' - 0.071B' + 128$$
$$Cb = -0.148\ R' - 0.291G' + 0.439B' + 128$$

$$R' = 1.164(Y'-16) + 1.596(Cr - 128)$$
$$G' = 1.164(Y'-16) - 0.813\ (Cr-128) - 0.392\ (Cb - 128)$$
$$B' = 1.164(Y'-16) + 2.017\ (Cb-128)$$

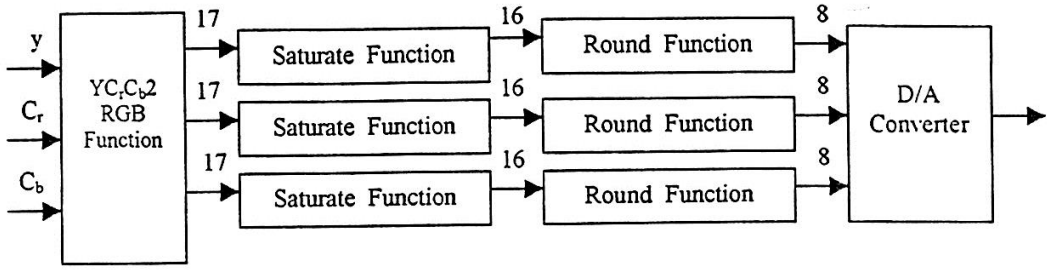
این مبدل در ماشین بینایی و پردازش تصویر استفاده می گردد. نتایج حاصل از پیاده سازی به صورت جدول (۲-۶) می باشد.

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

Function	Clock	Logic Cells
RGB $2YC_rC_b$	100 MHZ	380
YC_rC_b2RGB	100 MHZ	247

جدول (۲-۶) نتایج حاصل از پیاده سازی مبدل $RGB2yc_rc_b$

یک کانال تصویر دیجیتال (*Digital Video*) در شکل (۲-۱۳) نشان داده شده است. با توجه به مقادیر ورودی خروجیهای بلوک اول ممکن است ۱۷ بیتی باشند بنابراین از دو واحد *Round Function , Saturate Function* در ادامه برای نرمالسازی استفاده گردیده است و در نهایت با یک مبدل *D/A* خروجیها به سیگنال تصویر آنالوگ تبدیل می شود.[۱۹]

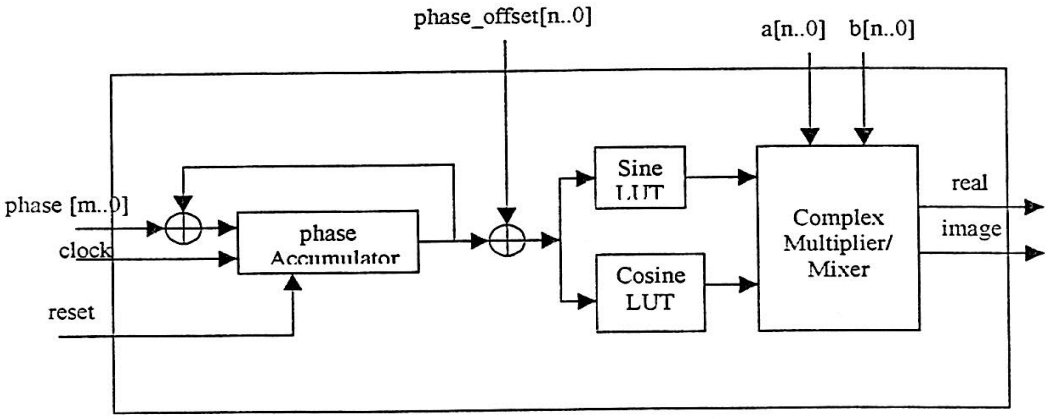


شکل (۲-۱۳) کانال تصویر دیجیتال

WikiPower.ir

۲-۷- مدولاتور دیجیتال

مدولاتور دیجیتال در مدولاسیونهای دامنه فرکانس و فاز مورد استفاده قرار می گیرد. مطابق بلوک دیاگرام شکل (۲-۱۴) در این طراحی دو جدول جستجو (*Look-up table*) برای ایجاد توابع سینوس و کسینوس وجود دارد که توابع $sin\ [2n/N]$ و $Cos\ [2n/N]$ را تولید می نمایند. N تعداد نمونه ها در LUT و N ورودی آدرس به LUT می باشد. ورودی $phase-offset$ زاویه فاز را مدوله می نماید.



شکل (۲-۱۴) بلوک دیاگرام مدولاتور دیجیتال

ضـــرب کننده – مخلوط کننده مختلط دو عدد مختلط با فرمت مکمل ۲ را در یکدیگر ضـــرب می نماید و در طراحی آن یک معماری خط لوله ای (*Pipeline*) استفده شده است و بدین صورت ماکزیمم سرعت فراهم می شود. در این بخش عکلیات صفحه بعد انجام می شود.

$$real + j\,imag = (a + jb) \times (c + jd)$$

$$j = \sqrt{-1}$$

$$real = (a \times c) - (b \times d)$$

$$image = (a \times d) + (b \times c)$$

WikiPower.ir

کل تاخیر مدولاتور از ورودی فاز به خروجی حقیقی (*Real*) شش پالس ساعت می باشند.[۲۰]

۲-۸- کنترلر گذرگاه *USB*

USB مخفف *Universal Serial Bus* می باشد و یک پورت سریال پر سرعت است که برای ارسال داده می تواند از دو مود سرعت کم (*Low Speed*) و سرعت بالا (*Full Speed*) بهره ببرد.

در مورد ســرعت پایین آهنگ ارســال داده ۱/۵ *Mb/s* می باشــد در حالیکه در مود ســرعت بالا نرخ انتقال بیت ها حداکثر تا ۱۲ *Mb/s* می رســد. *USB* بوســیله هدایت کنندگان صــنایع *PC* و مخابرات پیشــنهاد و ســاخته شــد. هدف اصــلی در این پیشــنهاد آن بود که قطعات *Plug and Play* را به خارج کامپیوتر منتقل کرده، نیاز به نصـــب کارت در داخل اســـلات مربوطه در کامپیوتر و ترکیب بندی (*configure*) سیستم را حذف کنند. *USB* امروزه بطور گسترده در کامپیوترهای متحرک بکار می رود.

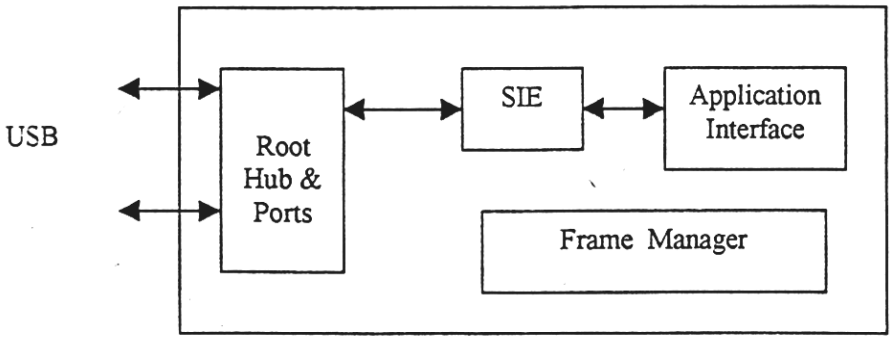
برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

این گذرگاه برای سرویس دادن به طیف گسترده‌ای از قطعات جانبی *PC* مانند صفحه کلید، مونیتور، دوربین‌ها، تلفن، مودم،چاپگر و ... مناسب است.

خصوصیات مهم *USB* بدین شرح است:

- برای ارتباطات سریال بکار می رود.
- تا ۱۲۷ قطعه فیزیکی را حمایت می کند.
- قابلیت اتصال چند گانه: عملیات همزمان را برای چندین قطعه ممکن می سازد.
- امکان انتقالات سنکرون و آسنکرون را از طریق یک سیم ایجاد می کند.
- تکنولوژی قطعات مجتمع.
- حمایت کامل از انتقال داده زمان واقعی(*Real Time*) برای سیگنالهای صدات و تصویر.

این‌گذرگاه دارای یک کنترلر می‌باشد.مطابق با شکل(۲-۱۵)این‌کنترلر از بخشهای زیر تشکیل شده‌است:



شکل (۲-۱۵) بلوک دیاگرام کنترلر گذرگاه USB

واحد *SIE (Serial Interface Engine)* اعمال یک فرستنده – گیرنده از قبیل فرم دهی بسته اطلاعات *(Packet Formation)* و سریال سازی آن را به عهده دارد. علاوه بر این کدینگهای مختلف نیز از وظایف این بخش است.

واحد *Frame Manager* مولد سیگنال شروع فریم *(SOF)* است و مدیریت بسته های داده و *Handshake* را به عهده دارد.

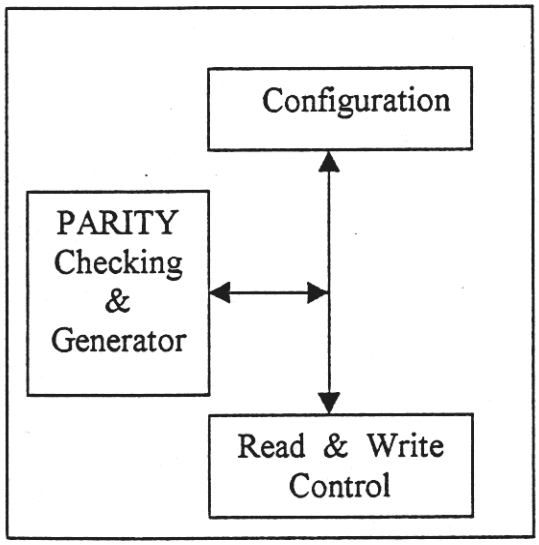
پیاده سازی طرح فوق بر روی قطعه *EPF10K30* انجام گرفته و ۱۲۵۰ سلول منطقی استفاده گردیده است و فرکانس کار آن ۱۲ مگاهرتز می باشد[۲۱] [۲۲].

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

۲-۹- کنترلر گذرگاه *PCI*

بدلیل اهمیت گذرگاه *PCI* بعنوان یک استاندارد کاری برای برقراری ارتباط ادوات و بردهای جانبی با کامپیوترهای شخصی و با همدیگر، امروزه توجه بسیاری به طراحی درایورهای *PCI* در صنعت میکروالکترونیک شده است. یکی از کاربردهای این درایورها زمانی است که طراحان یک سیستم مفصل بخواهند سیستم خود را در یک حجم تولید محدود به بازار ارائه کنند و به این منظور مایل هستند که تولید خود را با استفاده از *FPGA* پیاده سازی نمایند. این مسئله هنگامیکه قبل از تولید انبوه بخواهیم یک محصول را بر روی تراشه های *FPGA* آزمایش نهائی کنیم پیش خواهد آمد. در این حالت باز طراح سیستم به یک بلوک سخت افزاری که بتواند کار کنترل گذرگاه *PCI* را انجام بدهد نیاز خواهد داشت.

کنترلر پیاده سازی شده بر روی تراشه های *FPGA* دارای بلوک دیاگرام شکل (۲-۱۶) خواهد بود.



شکل (۲-۱۶) بلوک دیاگرام کنترلر گذرگاه PCI

بلوک *Configuration* در بر دارنده رجیسترهای داخلی است و ارتباطات خوانده و نوشتن آنها را بر عهده دارد. برخی از وظایف بلوک کنترل خواندن و نشتن عبارت است از: عملیات خواندن و نوشتن حافظه و *I/O* ، بخشی از عملیات دیکود کردن آدرس و .... .

بلوک *Parity* عملیات تولید پریتی برای دستورالعملهای خواندن، چک کردن پریتی و فعال سازی سیگنالهای خطا در صورت بروز خطا، در فاز آدرس و در انتقالات نوشتن را انجام می دهد. پیاده سازی

طرح فوق بر روی قطعه *EPF10K250ABC600* انجام پذیرفته و ۱۲/۰۱ درصد از کل سطح قطعه اشغال گردیده است و فرکانس کاری آن ۲۹/۱ مگا هرتز می باشد [۲۳].

۲-۱۰- کد کننده گفتار *ITU-T G.723.1*

استاندارد *ITU-T G.723.1* که قسمتی از استاندارد ارتباطی صوتی – تصویری *ITU-T H.324* می باشد معرف کد کننده گفتار با نرخ بیتی پایین است. این کد کننده سیگنال گفتار را در دو نرخ بیتی *5.3kbps* و *6.3kbps* با نسبت فشرده سازی *20:1* و *24:1* کد می کند. برخی از خصوصیات کد کننده گفتار *G.723.1* در جدول (۲-۷) نشان داده شده است.

Bit rate	*5.3kbps/6.3kbps*
Sampling Rate	*8KHZ*
Frame Size	*240 Samples*

جدول (۲-۷) برخی از خصوصیات کد کننده گفتار *G.723.1*

برای پیاده سازی این کد کننده از تراشه های با قابلیت پیکر بندی می توان استفاده نمود. نتایج حاصل از پیاده سازی در جدول (۲-۸) وجود دارد. نتیجه بر اساس تعداد سیکل ساعت مورد نیاز جهت کد کردن و دیکد کردن هر قاب سیگنال گفتار می باشد.

5.3 Kbps	*6.3 Kbps*
378749	*466327*

جدول (۲-۸) نتایج حاصل از پیاده سازی کد کننده گفتار

آزمایشها نشان می دهد که کد کننده گفتار که به این شکل طراحی گردیده است دارای کارآیی بالاتری نسبت به کد کننده گفتار با استفاده از *DSP (TMS320C62X)* می باشد. [۲۴].

۲-۱۱- کد کننده کد فایر

به منظور انتقال صحیح اطلاعات، روشهایی جهت آشکار سازی و اصلاح خطا پیشنهاد گردیده است در بین این روشها، کد گذاری و کدبرداری بهترین روش برای تشخیص و تصحیح خطا شناخته شده بدین منظور کدهای مختلفی تو سط ریا ضیدانان مختلف برای اینکار ارائه شده ا ست که هر کدام دارای مزایا و

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

معایب خاص می با شند. کدهای تصحیح خطای بلوکی چرخشی بنا به خصو صیاتی که دارند برای ایجاد یک سیست کنترل خطا به منظور پیاده سازی سخت افزاری مفید می با شند. کد فایر از خانواده کدهای بلوکی چرخشی می باشد.

جهت پیاده سازی از چند جمله ای زیر استفاده می نماییم :

$$1+X^{2}+X^{5}+X^{9}+X^{11}+X^{14}$$

با دریافت ۲۵۶ بیت اطلاعات ورودی و با ساخت و اضافه نمودن ۱۴ بیت اطلاعات افزونی به انتهای آن بردار کد را می سازد. در این پیاده سازی از شیفت رجیستر ۱۴ بیتی استفاده گردیده است که با توجه به چند جمله ای مولد فوق فیدبکهایی نیز در مدار وجود دارد. نتایج حا صل از پیاده سازی در جدول (۲-۹) نشان داده است.[۲۵]

CLB	*Flip Flop*	F_{max}
36	*35*	*40.863*

۲-۱۲- پیاده سازی سخت افزاری الگوریتمهای سطح بالای پردازش تصویر با استفاده از پیکر بندی جزئی *FPGA* در زمان اجرا:

در پردازش بلادرنگ تصاویر با حجم بالایی از داده و عملیات متنوع روبه رو هستیم این عملیات با استفاده از سیستم های همه منظوره قابل انجام نبوده و نیاز به سیستم های خاص منظوره دارد. برای این منظور می توان از الگوریتم های پایه مورفولوژی استفاده نمود.

ریاضیات مورفولوژی یکی از روشهایی است که در زمینه های مختلف پردازش تصویر از قبیل تشخیص لبه و فیلترینگ و ... می توان از آن بهره جست. در این ریاضیات با استفاده از واحدهای عملیاتی پایه و مقدماتی عملیات پیچیده قابل ساخت هستند. برای پیاده سازی سخت افزاری این الگوریتمها روشهای گوناگونی وجود دارد. یکی از این روشها، روش محاسباتی است که ترکیبی از عناصر جمع یا تفریق کننده و مقایسه و تعویض را مورد استفاده قرار می دهد. روش تجزیه آستانه نیز تصویر را به چند زیر بخش تقسیم می نماید. روش دیگری نیز وجود دارد که می توان در پیاده سازی الگوریتمهای سطح بالا، به صورت پیکربندی سخت افزار در زمان اجرا از این الگوریتم های پایه بطور بهینه استفاده نمود.

در اجرای الگوریتمها از دو روش می توان بهره جست:

۱- استفاده از سخت افزارهای خاص منظوره.

۲- پیاده سازی نرم افزاری آنها.

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

امروزه با اســـتفاده از تراشـــه های با قابلیت پیکربندی مجدد بطور همزمان از قدرت ســـخت افزار و انعطاف پذیری نرم افزار استفاده می نماید. به منظور بالا بردن انعطاف پذیری چنین سیستم هایی از ایده پیکر بندی *FPGA* در زمان اجرا اســـتفاده می گردد. این فرایند(*RTR*) در حالت ایده آل باعث می گردد که محدودیت سخت افزاری به طور کلی مرتفع گردد و بتوان هر مدار سخت افزاری با هر حجمی را با یک سخت افزار محدود پیاده نمود.

کم نمودن زمان پیکربندی مجدد بطرز قابل توجهی در کارآئی کل ســـیســـتم تاثیر خواهد گذاشـــت. برای این منظور می توان از پیکربندی جزیی بهره جســـت. بدین معنا که در هر بار پیکربندی مجدد، کل ســـخت افزار فقط تفاوتهای پیکربندی موجود و پیکربندی جدید را بر روی تراشـــه پیاده نمائیم. البته نوع سخت افزار بکار رفته بایستی به شکلی باشد که از لحاظ فنی قابلیت پیکربندی جزئی را دارا باشد. نمونه ای از این تراشه ها *Virtex* از شرکت *Xilinx* می باشد.

نرم افزارهائی وجود دارد که به طراح کمک می نمایند تا او بتواند کنترل خوبی بر جایگذاری و مسیر یابی در داخل *FPGA* داشـــته باشـــد تا بتواند بر ســـخت افزارهائی که در زمان اجرا تغییر می کنند. کنترل خوبی داشته باشد. نمونه ای از این نرم افزارها، نرم افزار *JBit* می باشد.

پس از سنتز و شبیه سازی طراحی انجام شده نتایج به صورت جدول (۲-۱۰) بیان می گردد.

نام تراشه	حجم سخت افزار مصرفی	ماکزیمم فرکانس
XCV100-6-TQ144	*88 %*	*52.110 MKZ*

جدول (۲-۱۰) نتایج حاصل از سنتز

نرم افزاری برای محاســـبه زمان پیکربندی جزئی (*Reconfiguration Time*) وجود دارد. با اعمال فایلهای بیتی دو الگوریتم پایه مورفولوژی *(Erosion , Dilation)* به برنامه مذکور فایل پیکربندی جزئی بدســـت می آید.جدول(۲-۱۱)زیر مقایســـه ای بین حجم های فایلهای پیکربندی جزئی و کامل می باشـــد [۲۶].

حجم فایل پیکربندی جزئی	حجم فایل پیکربندی کامل	الگورتیمهای پایه
۱۵۸۹۶	۹۷۶۵۲	*Erosion*
۱۵۸۹۶	۹۷۶۵۲	*Diation*

جدول (۲-۱۱) مقایسه بین فایل پیکر بندی جزئی و پیکربندی کامل

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

۲-۱۳- مترجم های زبانهای سطح بالا به زبان *VHDL*

در سیستم های پردازش تصویر برخی از توابع وجود دارد که به عنوان مولفه های پایه ای در این گونه سیستم ها بکار می رود. با استفاده از این عملگرها می توان کتابخانه ای ایجاد نمود که در ساده سازی طراحی پروژه های بزرگ بسیار موثر است.

برای پیاده سازی این الگوریتمها از زبان سطح بالای *SA-C* و کمپایلر آن استفاده شده است. این زبان نسخه ای (*a Single – Assignment variant*) از زبان برنامه نویسی *C* می باشد. به کمک این نرم افزار الگوریتمهای پردازش تصویر بیان شده در زبان برنامه نویسی *C* به کدهای *VHDL* قابل سنتز تبدیل می گردد به عنوان نمونه برنامه محاسبه یک کانولوشن (۳×۳) با سرعت ۹/۷ مگا پیکسل در ثانیه که توسط زبان *SA-C* بیان شده است عبارت است از :

```
Unit 20 [:,:] main (unit 8 image [:,:], unit 8
kernel [:,:])
{//***** computes the gradiant for the image
*****
  Unit20 res [:,:] =
      For window win [3,3] in image
          {unit 20 val =
                For elem1 in win dot elem2 in kernel
                    Return (sum (( unit20) elem1×elem2));
     }
  Return (aaray (val));
} return (res);
```

در جدول (۲-۷) پاره ای از این مولفه های که بر روی تراشه های قابل برنامه پذیری پیاده سازی گردیده ا ست ذکر شده ا ست. در این جدول ستون دوم سرعت ساعت برای تراشه ستون سوم سرعت پردازش (با واحد مگا پیکسل در ثانیه) و ستون چهارم زمان انتقال داده و ستون پنجم تعداد بلوکهای منطقی قابل پیکربندی (*CLBs*) مورد نیاز را نشان می دهد.

می توان ترکیبی از توابع جدول فوق را بدلخواه انتخاب و در یک *F PGA* پیاده سازی نمود.

علاوه بر این کمپایلر، کمپایلر دیگری وجود دارد که الگوریتم های پردازش تصـــویر را که در محیط نرم افزار مطلب تهیه گشته اند مستقیما به کدهای *VHDL* تبدیل می نماید. [۲۷] [۲۸] [۲۹].

Benchmark	*Frequency (MHz)*	*Processing (MP/s)*	*W/data transfer*	*CLBs*
Add (Dyadic)	*8.477*	*4.23*	*3.94*	*750*
Add (Monadic)	*9.040*	*9.01*	*7.38*	*647*
Convolution (3x3)	*9.554*	*9.70*	*5.89*	*1,333*
Dilation (3x3)	*8.915*	*9.05*	*3.61*	*1,036*
Erosion (3x3)	*8.981*	*9.13*	*3.62*	*1,036*
Gaussian Filter (3x3)	*8.684*	*8.83*	*6.59*	*725*
La Place Filter (3x3)	*8.625*	*8.77*	*6.62*	*773*
La Place Filter (5x5)	*8.692*	*4.52*	*4.00*	*1,214*
Window Sum of Diffs (MP4)	*10.109*	*2.13*	*2.03*	*477*
Max Filter (4x5)	*9.338*	*9.60*	*7.78*	*835*
Max Value in Image	*12.830*	*12.8*	*10.6*	*320*
Min Filter (3x4)	*8.652*	*8.80*	*7.24*	*743*
Multiply (Dyadic ; 16-bit out)	*8.713*	*4.35*	*4.13*	*800*
Multiply (Monadic ; 16-bit out)	*8.911*	*8.88*	*6.59*	*686*
Prewitt Magnitude	*2.318*	*2.36*	*2.23*	*1,141*
Roberts Magnitude	*3.038*	*3.06*	*2.84*	*940*
Sobel Magnitude	*2.162*	*2.20*	*2.09*	*1,184*
Square Root (fix8.4out)	*6.497*	*6.48*	*5.58*	*660*
Subtract (Dyadic)	*8.960*	*4.45*	*3.67*	*764*
Subtract (Monadic)	*9.492*	*9.43*	*6.88*	*641*
Threshold (1-bit out)	*8.577*	*8.55*	*3.84*	*693*
Wavelet (5x5)	*8.025*	*2.48*	*1.29*	*1,433*

جدول(۲-۷) پاره ای از توابع پایه ای در پردازش تصویر

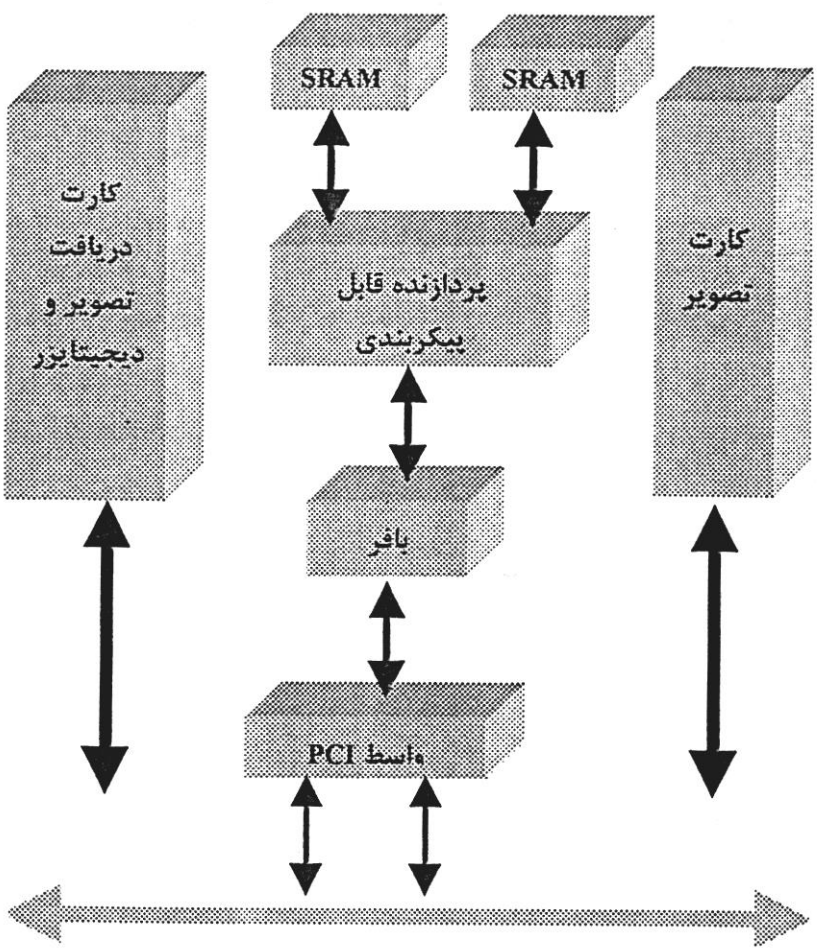
برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

۲-۱۴- پیاده سازی یک پردازشگر تصویر قابل پیکربندی مجدد

برای ایجاد سیستمی که در بسیاری از پیاده سازیهای مربوط به الگوریتم های پردازش تصویر بتوان از آن بهره جست بلوک دیاگرام شکل (۲-۱۷) را در نظر بگیرید. پردازنده قابل پیکربندی توسط کامپیوتر میزبان و از طریق گذرگاه *PCI* برنامه ریزی می گردد و بنابر نیاز به صورت یک پردازشگر خاص مورد استفاده قرار می گیرد. تصویر دریافتی از دوربین ابتدا توسط کارت مبدل آنالوگ به دیجیتال به صورت داده ای که برای پردازش آماده است تبدیل می گردد. از طریق گذرگاه *PCI* در اختیار واحد پردازنده قابل پیکربندی قرار می گیرد. پردازش لازم بر روی داده انجام گرفته و داده خروجی در اختیار کارت تصویر قرار می گیرد. هر یک از عملگرهای پیش بینی شده در جدول (۲-۷) و یا ترکیبی از این عملگرها که بتوانند پردازش تصویر مورد نیاز را فراهم نمایند را می توان در پردازنده قابل پیکربندی تعریف نمود. از جمله کاربردهای این سیستم کنترل خط تولید برخی از کارخانجات است که محصولات تولیدی انها متنوع و زیاد است [۳۰] [۳۱] [۳۲].

WikiPower.ir

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



شکل (۲ - ۱۷) بلوک دیاگرام یک واحد پردازشگر تصویر قابل پیکر بندی مجدد

۲-۱۵- جمع بندی

در این فصل دیدیم که تراشه های *FPGA* برای پیاده سازی انواع مختلفی از فیلترها اعم از فیلترهای *FIR* ، *IIR* ، *Wavelet* و ... بسیار ایده آل هستند. علاوه بر این بسیاری دیگر از فیلترها از قبیل فیلترهای غیر خطی، وفقی، عددی با استفاده از زبان *VHDL* به صورت بهینه قابل طراحی هستند. استفاده از این تراشه ها برای پیاده سازی توابع مربوط به پردازش سیگنالهای صوت و تصویر آن قدر متداول است که کمپایلرهای مخصوص به این کار تهیه گردیده است. به نحوی که با استفاده از این کمپایلرها می توان الگوریتمهای پردازش سیگنال را که با استفاده از نسخه ای خاص از زبان برنامه نویسی

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

C و یا نرم افزار مطلب انجام می پذیرد به صـــورت کد قابل ســـنتز *VHDL* دریافت نموده و آن را بر روی تراشه های *FPGA* پیاده نمود.

WikiPower.ir

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

فصل سوم

کنترلر گذرگاه CAN

WikiPower.ir

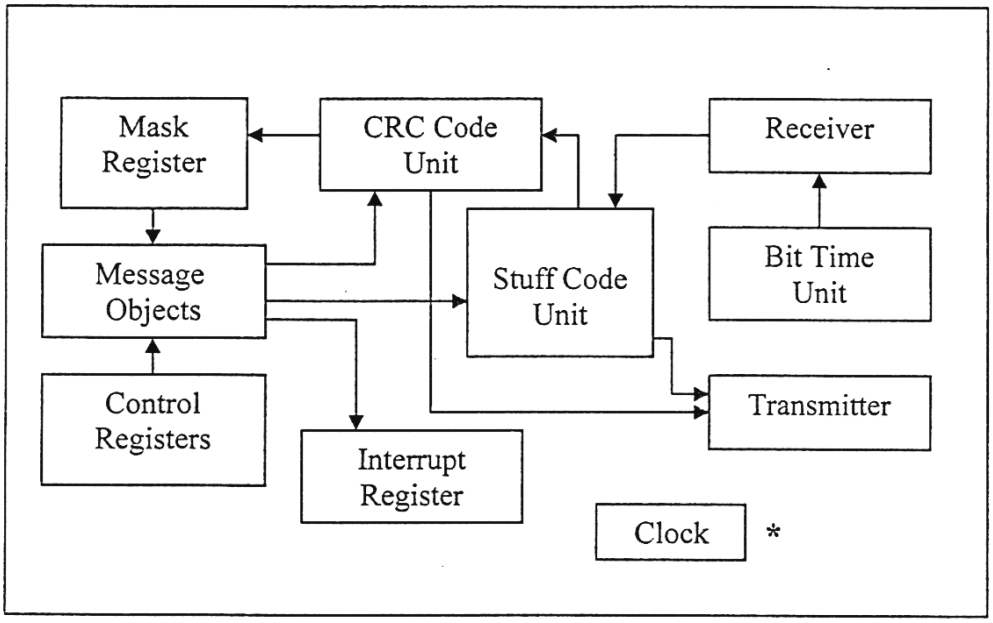
برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

۳-۱- مقدمه

در این فصل با ساختار داخلی تراشه ۸۲۵۲۷ که یکی از کنترلرهای *CAN* می باشد آشنا می شویم.

همانطور که در فصل اول گفته شد این کنترلر به عنوان واسطه ای بین میکرو کنترلر و فرستنده – گیرنده های متصل به شبکه عمل نموده و پیام هائی را بین آنها مبادله می نماید. برای دریافت و ارسال پیام هائی را بین آنها مبادله می نماید. برای دریافت و ارسال پیام ها پروتکلهایی در نظر گرفته شده است که پس از بررسی ساختار داخلی تراشه این پروتکلها و انواع خطاهای مربوط به کنترلر را بیان خواهیم نمود. برای کسب اطلاعات کامل در ارتباط با سخت افزار کنترلر از مراجع [۱] و [۳۰] می توانید استفاده نمائید. اطلاعات کامل در مورد پروتکل های *VHDL* مربوط به این تراشه در مراجع [۲] و [۳۱] و [۳۲] وجود دارد.

مطابق بلوک دیاگرام شکل (۳-۱) برای کنترلر *CAN* بخشهای زیر را در نظر می گیریم:



* واحد ساعت به کلیه واحد ها به جز رجیسترها متصل است .

شکل(۳-۱) بلوک دیاگرام کنترلرCAN

الف – ثبات های کنترلی

در این بخش ثبات هایی وجود دارند که متناسب با بیتهای آنها اعمال کنترلی مربوط به وضعیتهای مختلف گذرگاه، وقفه ها، حالت های گوناگون کارکرد تراشه، ساعت سیستم و .. مشخص شوند.

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

ب- واحد زمان بندی بیت.

در این واحد زمان نمونه برداری از بیت دریافتی و چگونگی این نمونه برداری تعیین می گردد.

ج- ثبات های ماسک توسعه یافته و استاندارد.

این ثبات ها به منظور تعیین بسته پیام خاص استفاده می گردد.

د- ثبات های بسته پیام.

در تراشه مکانهایی برای دریافت و ارسال پیام ها در نظر گرفته شده است در هر واحد بسته پیام دو ثبات امکان دسترسی به پیام، چگونگی دسترسی، نحوه دریافت و ارسال پیام را مشخص می کنند.

ه- ثبات وقفه.

در کنترلر برای نشان دادن منبع وقفه از این ثبات استفاده می گردد.

و- واحد *CRC*.

برای کاهش خطا در ارسال و دریافت پیام کد *CRC* به پیام افزوده می گردد این واحد جهت محاسبه کد *CRC* طراحی گشته است.

ز- واحد *Stuff Code*

به منظور کاهش خطا در ارســـال و دریافت پیام کد *Stuff* به پیام افزوده می گردد این واحد جهت محاسبه کد *Stuff* طراحی گشته است.

WikiPower.ir

۳-۲- پایه های تراشه کنترلر *CAN*

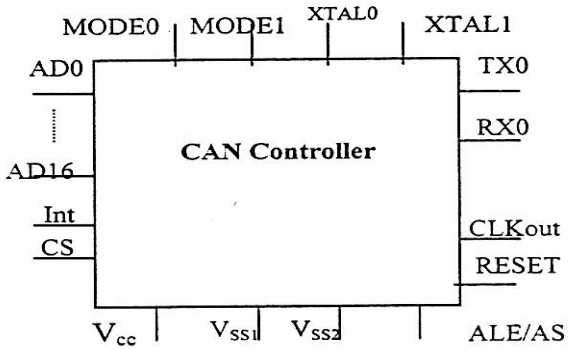
عملکرد پایه های تراشه ۸۲۵۲۷ به شرح زیر می باشند:

RX0/RX1: پایه های ورودی، برای دریافت سریال پیام ها از گذرگاه *CAN*.

TX0/TX1: پایه های خروجی، برای ارسال سریال پیام ها بر روی گذرگاه *CAN*.

CLK out: پایه خروجی، فرکانس این پایه توسط کنترلر قابل برنامه ریزی می باشد.

RESET: پایه ورودی، این پایه توسط میکرو کنترلر برای ری ست نمودن کنترلر تغذیه می گردد.



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

شکل (۳-۲)پایه های کنترلر *CAN*

XTAL1 –XTAL2: پایه ورودی، پالس ساعت مورد نیاز برای کار کردن کنترل را فراهم می نماید.

CS: پایه ورودی، این پایه به عنوان فعال کننده تراشه می باشد که توسط میکرو کنترلرتغذیه می شود.

Vss: پایه زمین تراشه.

Vcc: تغذیه کنترلر *CAN*.

Int: پایه ورودی، وقفه کنترلر است که در موارد مشخصی (همانطور که ذکر شد) وضعیت خود را به میکرو کنترلر اعلام می نماید.

MODE0, MODE1: برای تعیین مود کارکرد کنترلر بکار می رود.

ALE/AS: برای تبادل داده بین کنترلر و *CPU* استفاده می گردد.

این کنترلر می توان در مودهای زیر(برای ارتباط کنترلر و پردازنده میزبان) عمل نماید: (تفاوت این مودها در تعداد خطوط ادرس و داده می باشد)

مود صفر: در این مود گذرگاه داده و آدرس کنترلر به صورت ۸ بیتی مالتی پلکس(اینتل) شده است و بقیه خطوط آدرس و داده به صورت یک پورت ورودی – خروجی عمل می نماید.

مود یک: در این مود گذرگاه داده و آدرس کنترلر به صورت ۱۶ بیتی مالتی پلکس شده است.

مود دو: در این مود گذرگاه داده و آدرس کنترلر به صورت ۸ بیتی مالتی پلکس شده (غیر اینتل) است و بقیه خطوط آدرس و داده به صورت یک پورت ورودی – خروجی عمل می نماید.

مود سه : در این مود گذرگاه داده و آدرس کنترلر به صورت ۸ بیتی غیر مالتی پلکس شده است.

مود سریال: علاوه بر چهار مود موازی فوق یک مود سریال نیز وجود دارد در این مود خطوط ۴ و ۶ گذرگاه ادرس و داده به صورت زیر عمل می نماید.

MOSI: ورودی داده سریال به کنترلر از پردازنده میزبان.

SCLK: پایه ورودی پالس ساعت سریال برای کنترلر که توسط پردازنده میزبان هدایت می گردد.

MISO: خروجی داده سریال از کنترلر به پردازنده میزبان.

۳-۳- بررسی سخت افزار کنترلر *CAN*

۳-۳-۱- شمارنده های خطا در کنترلر *CAN*

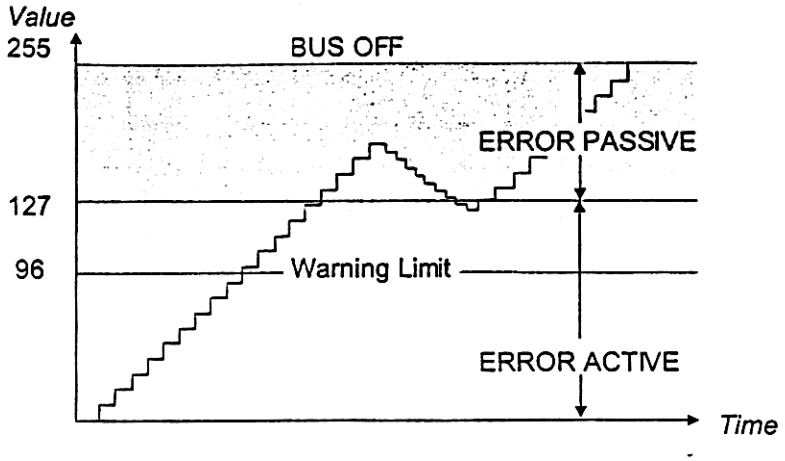
در کنترلر *CAN* دو شمارنده خطا وجود دارد:

۱- شمارنده خطای دریافت (*REC*)

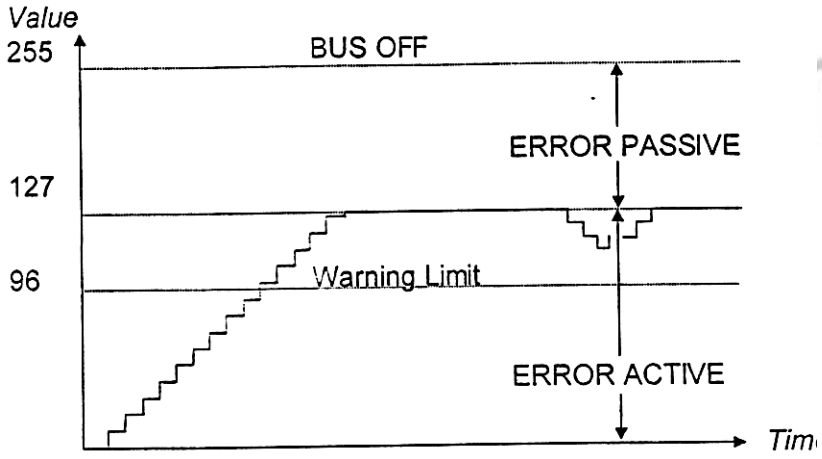
برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

۲- شمارنده خطای ارسال (*TEC*)

مقدار این شمارنده ها با تشخیص خطا افزایش و با ارسال یا دریافت صحیح پیام کاهش می یابند. با توجه به مقدار شـــمارنده ها وضـــعیت گره تغییر داده می شـــود. حالت اولیه یک کنترلر *CAN* خطا فعال (*Error Active*) است. شکل های (۳-۳) و (۳-۴) وضعیتهای مختلف گره را با توجه به شمارنده خطای ارسال و دریافت نشان می دهد.



شکل (۳-۳) شمارنده خطای ارسال



شکل (۳-۴) شمارنده خطای دریافت

همانطور که در شکل های مذکور نشان داده شده است. در صورتی که مقدار هر کدام از شمارنده های خطا کمتر از ۱۲۷ باشد کنترلر در حالت *Active Error* و در صورتی که مقدار آن بیشتر از ۱۲۷ باشد کنترلر در حالت *Passive Error* قرار دارد اگر مقدار شمارنده های خطا برابر عدد ۹۶ باشد کنترلر

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

در وضعیت *Warm* قرار می گیرد. وضعیت *bus off* در صورتی بوجود می آید که یکی از شمارنده های خطا برابر ۲۵۶ باشد.

۳-۳-۲- ثبات های کنترلی

ثبات های کنترلی در این تراشه عبارتند از:

۱- ثبات فعال کننده وقفه ها

۲- ثبات وضعیت.

۳- ثبات واسط *CPU*.

۴- ثبات پیکربندی گذرگاه.

۵- ثبات *CLKOUT*.

۳-۳-۲-۱- ثبات فعال کننده وقفه ها

این ثبات مطابق شکل (۳-۵) از بیتهای زیر تشکیل شده است.

0	*CCE*	*0*	*0*	*EIE*	*SIE*	*IE*	*Init*

شکل (۳-۵) ثبات فعال کننده وقفه ها

۱- بیت فعال کننده تغییر پیکربندی *Change Configuration Enable (CCE)*

با فعال بودن این بیت *CPU* قادر اســـت محتویات ثبات های پیکربندی که عبارتند از ثبات های با آدرس *1FH , 2FH , 3FH , 4 FH , 9FH , AFH* که پیکر بندی تراشه را تعریف می کنند تغییر دهد..

۲- بیت فعال کننده وقفه خطا *Error Interrupt Enable (EIE)*

با فعال بودن این بیت وقفه های خطا (*Warn , Boff*) فعال می گردند اگر این بیت توســـط *CPU* ست گردد و تعداد زیادی خطای گذرگاه *CAN* رخ داده باشد کنترلر *CAN* می تواند به *CPU* وقفه دهد.

۳- بیت فعال کننده وقفه تغییر وضعیت *Status Change Interrpt Enable (SIE)*

اگر یک خطای *CAN* رخ دهد یا یک همل انتقال پیام با موفقیت تکمیل شود و بیت فوق ست باشد یک وقفه تولید خواهد شد.

۴- بیت فعال کننده وقفه *Interrupt Enable (IE)*

با فعال بودن این بیت توسط *CPU* وقفه های کنترلر فعال می گذرند این بیت همراه با بیتهای *EIE* یا *SIE* و وقفه های *TX/RX* مربوط به بسته پیام بکار می رود.

۵- بیت مقدار دهی اولیه *Initialization (Init)*

به دنبال یک ری ست سخت افزاری مقدار دهی اولیه صورت می پذیرد. این بیت توسط *CPU* تنظیم می گردد. علاوه بر این اگر حالت *bus off* بوجود آید کنترلر *CAN* آن را ست می نماید در این صورت همه ارسال (دریافت) پیام ها به (از) گذرگاه *CAN* متوقف می شود در این حالت خروجی *TX* مغلوب (*recessive*) می باشد. برای محاسبه مقدار خطاها ثبات وضعیت را در نظر می گیریم:

۳-۳-۲-۲- ثبات وضعیت

این ثبات در شکل (۳-۶) نشان داده شده است.

Boff	*Warn*	*Wake*	*RXOK*	*TXOK*	*LEC2*	*LEC1*	*LEC0*

شکل (۳-۶) ثبات وضعیت

۱- *B off*

اگر شمارنده خطا در کنترلر به حد ۲۵۶ بر سد کنترلر به حالت گذرگاه خاموش (*Bus off*) می رود. در طول این حالت هیچ پیامی دریافت نمی شود یا ارسال نمی گردد. با غیر فعال کردن بیت *Init* این حالت خاتمه می یابد. به دنبال این عمل شمارنده های خطای ارسال و دریافت *reset* می گردد.

۲- هشدار *Warning Status (Warn)*

در صورتیکه شمارنده خطا در کنترلر برابر ۹۶ گردد حالت هشدار به وجود می آید. در این حالت اگر بیتهای *EIE* و *IE* مربوط به واحد کنترل فعال باشند یک وقفه اتاق خواهد افتاد.

۳- *Wake Up Status Wake*

در یکی از سه حالت زیر این بیت فعال می گردد:

الف- اگر کنترلر در حالت *Sleep* باشد و فعالیت گذرگاه آغاز گردد.

ب- اگر *CPU* حالت *Sleep* را غیر فعال نماید.

ج- بعد از خروج از حالت *power down* ، حالت *sleep* فعال شود.

۴- دریافت (ارسال) موفق پیام *RXOK (TXOK) (Receive (Transmit) Message Successfully)*

اگر یک پیام به شکل صحیح دریافت (ارسال) گردد این بیت ست می شود.

۵- کد آخرین خطا *Last Error Code (LEC)*

این بخش آخرین خطایی را که در سیستم رخ داده است نشان می دهد.

کدهای مربوط به هر نوع خطا به صورت زیر است: ۱- خطای *Stuff* ۲- خطای *CRC*

۳-۳-۲-۳- ثبات واسط *CPU*

RstST	*DSC*	*DMC*	*pwD*	*Sleep*	*MUX*	*0*	CE_n

شکل (۳-۷) ثبات واسط *CPU*

۱- *Rst ST(Reset Stats)*

ست بودن این بیت نشان می دهد که ری ست سخت افزاری کنترلر باس فعال است. اگر ری ست فعال باشد هیچ دسترسی به کنترلر وجود ندارد.

۲- *DSC (Divide System Clock)*

در صورت فعال بودن این بیت ساعت سیستم برابر نصف *CLK* ورودی و در صورت صفر بودن برابر *CLK* است.

۳- *DMC(Divide Memory Clock)*

در صورت فعال بودن این بیت ساعت حافظه سیستم برابر نصف ساعت سیستم و در صورت صفر بودن برابر ساعت سیستم است.

۴- *PWD (Power Down)* فعال کننده حالت *Sleep*

در این مود سیگنال *Clock out* و *CLK* فعال نیست و دسترسی به بسته های پیام وجود ندارد. برای خروج از این حالت ری ست سخت افزاری یا ری ست بیت *PWD* بایستی انجام شود. حالت *Sleep* یا هنگامی که یک رخداد روی گذرگاه *CAN* بوجود آید خاتمه می یابد.

۶- *MUX* بوسیله این بیت پایه وقفه تعیین می گردد.

۷- *CEn (Clock out enable)*

ست بودن این بیت نشان می دهد که سیگنال *Clock out* خروجی فعال گردیده است.

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

۳-۳-۲-۴- ثبات پیکربندی گذرگاه

این ثبات مطابق شکل (۳-۸) از بخشهای زیر تشکیل شده است:

0	*CoBy*	*Pol*	*0*	*DcT1*	*0*	*DcR1*	*DcR0*

شکل (۳-۸) ثبات پیکربندی گذرگاه

۱- *(Comparator Bypass) CoBY*

در صورتی که این بیت ست باشد فقط *Pxo* به عنوان ورودی به کنترلر از گذرگاه می باشد.

۲- *(Polarity) POL*

در صورتی که این بیت فعال باشد منطق یک به عنوان غالب (*Dominant*) و منطق صفر به عنوان مغلوب (*Recessive*) بکار می رود ولی اگر این بیت صفر باشد به عکس منطق فوق عمل خواهد نمود.

۳- *(Disconnect TX1 out put) DcTI*

با ست بودن این بیت خروجی *TX1* غیر فعال می گردد.

۴- *(Disconnect RX1 input) DcR1*

با ست بودن این بیت ورودی *RX1* غیر فعال می گردد.

۵- *(Disconnect RX0 input) DcR0*

با ست بودن این بیت ورودی *Pxo* غیر فعال می گردد.

۳-۳-۲-۵- ثبات *CLK out*

این ثبات از بخشهای زیر تشکیل شده است :

0	*0*	SL_1	SL_0	$CD_۷$

شکل (۳-۹) ثبات *CLK OUT*

ثبات *CLK out* فرکانس سیگنال خروجی *CLK out* را کنترل می نماید. این عمل توسط دو بیت *SL0* و *SL1* انجام می پذیرد. بیتهای $CD_۷$ باعث می گردد توسط سیگنال *CLK out* توسط مقاومت های *Pull UP* برنامه ریزی گردد. این اعمال مطابق با جدول (۳-۱) انجام می شود.

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

CV_n	CLK out
0	CIK
1	CIK/2
10	CIK/3
11	CIK/4
1100	CIK/13
1101	CIK/14
1110	CIK/15

جدول (۳-۱) تعیین فرکانس پایه CLK out

۳-۳-۳- واحد زمان بندی بیت

۳-۳-۳-۱- شرعت نامی نرخ بیت (Nominal Bit Rate)

تعداد بیتهایی که در هر ثانیه بدون همزمانی مجدد بوسیله فرستنده ایده آل ارسال می گردد.

زمان بیت نامی (Nominal BIT Time) Nominal BIT Time = 1/Nominal BIT Rate

زمان بیت (BIT Time) شامل ۴ بخش است:

۱- بخش همزمانی برای همزمان نمودن و آماده نمودن گره های روی باس استفاده می شود.

۲- بخش زمان انتشار (Prop) برای جبران نمودن زمانهای تاخیر فیزیکی داخل شبکه در نظر گرفته شده است.

۳ و ۴- Phase SEG 1& 2 برای جبران خطاهای لبه فاز (Edge Phase) بکار می رود.

نقطه نمونه برداری (Sample Point)

زمانی است که سطح گذرگاه خوانده می شود و به عنوان مقدار واقعی بیت تلقی می گردد. این زمان در انتهای بخش ۱ فاز داده قرار دارد. در CAN یک واحد ثابت به نام Time QUANTUM (tq) در نظر می گیریم که از پریود CLOCK سیستم (tsclk) بدست می آید و مطابق رابطه زیر آن را تعریف می کنیم:

$$Tq = tsclk \times (BRP + 1)$$

BRP: baud Rate Prescaler

BRP یک مقدار ثابت است که در ادامه تعریف می شود.

در این صورت زمان بیت به صورت زیر قابل بیان است.

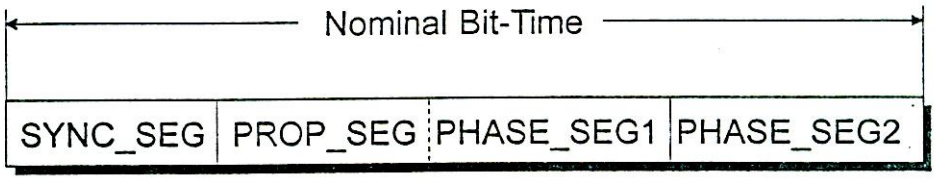
$$Bit\ time = t\ sync + t\ tseg1 + t\ tseg2$$

Tsync برابر یک $t\ q$ است و برای همزمانی و آماده نمودن گره های روی گذرگاه تعریف می شود.

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

T tseg1 مجموع *Prop* و *Phase1* می باشد و *T tseg2* برابر با *Phase 2* است.

واحد زمان بندی بیت دارای ۲ ثبات به شرح زیر می باشد:



شکل (۳-۱۰) زمان بیت نامی

۳-۳-۳-۲- ثبات صفر زمان بندی بیت

ثبات صفر زمان بندی بیت از دو قسمت تشکیل شده است.

BRP (۶ بیت)	*SJW* (۲ بیت)

شکل (۳-۱۱) ثبات صفر زمان بندی بیت

WikiPower.ir

الف – *Baud Rate Prescaler – BRP*

این ثابت که مقدار آن توسط کاربر تعیین می گردد برای محاسبه t_q استفاده می گردد و مقدار مجاز آن $0<BRP<63$ است.

ب- *Synchronization Jump Width – SJW*

وابسته به مقدار *SJW* ، مدت زمان بیت کوتاه یا بلند می گردد.

$$T_{sjw} = (SJW + 1) \times t_q \qquad 0 < SJW < 3$$

۳-۳-۳-۳- ثبات یک زمان بندی بیت

ثبات یک زمان یندی بیت از سه قسمت تشکیل شده است.

TSEG1 (۴ بیت)	*TSEG2*(۳ بیت)	*SPL*

شکل (۳-۱۲) ثبات یک زمان بندی بیت

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

الف- *TSEG1*

برای محاسبه زمان قبل از نقطه نمونه برداری استفاده می گردد.

$2 < TSEG1 < 15$ $\qquad T\,tseg1 = (TSEG1 + 1) \times t\,q$

ب- *TSEG2*

در محاسبه زمان بعد از نقطه نمونه برداری استفاده می گردد.

$1 < TSEG2 < 7$ $\qquad T\,tseg2 = (TSEG2 + 1) \times t\,q$

ج- مود نونه برداری *SPL*

در صورت ست بودن این بیت نمونه برداری در طول زمان بیت (*Bit time*) سه مرتبه انجام می گردد و در غیر این صورت نمونه برداری یک مرتبه در طول زمان بیت انجام می شود.

$T\,tseg2 >= Tsjw$

۳-۳-۴- ثبات های ماسک توسعه یافته و استاندارد

با استفاد از ثبات های فوق هر بیت از شناسه پیام وارده می تواند ماسک گردد در صورتی که صفر در هر بیت نو شته شود برای ان موقعیت صفر یا یک (*don't care*) می توان در نظر گرفت ولی در صورت وجود یک *1* مقدار بیت وارده بایستی با شنا سه پیام مطابقت کند. برای دریافت پیام در شبکه *CAN* دو فرمت متفاوت وجود دارد:

۱- فرمت استاندارد: این فریم دارای شناسه (*Identifier*) ۱۱ بیتی است.

۲- فرمت توسعه یافته : شناسه این فریم ۲۹ بیتی می باشد.

بر این اساس رجیسترهای ماسک نیز به صورت ۱۱ و ۲۹ بیتی می باشند.

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21
ID20	ID19	ID18	ID17	ID16	ID15	ID14	ID13
ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5
ID4	ID3	ID2	ID1	ID0	0	0	0

شکل (۳-۱۳) ساختمان ثبات ماسک

۳-۳-۵- بسته های پیام (*Message Objects*)

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

برای ارتباط فرســـتنده – گیرنده های گذرگاه با کنترلر از یک طرف و *CPU* با کنترلر از طرف دیگر بسته های پیام در نظر گرفته شده اند. در کنترلر *CAN* پانزده بسته پیام که هر کدام از ۱۵ ثبات ۸ بیتی تشکیل می شوند وجود دارند. هر بسته پیام دارای میدان کنترل، میدان داوری، میدان ترکیب بندی پیام و میدان داده است. شکل (۳-۱۴) ساختار یک بسته پیام را نشان می دهد.

۳-۳-۵-۱- میدان کنترل

از دو ثبات کنترل صفر و یک تشکیل شده است که هر یک شامل ۴ میدان دو بیتی می باشند شکل های (۳-۱۵) و (۳-۱۶) این دو ثبات را نشان می دهند.

WikiPower.ir

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

Control 0
Control 1
Arbitration 0
Arbitration 1
Arbitration 2
Arbitration 3
Mess. Conf
Data0
Data1
Data2
Data3
Data4
Data5
Data 6
Data7

شکل (۳-۱۴) ساختار یک بسته پیام

Msgval	TXIE	RXIE	Intpnd

شکل (۳-۱۵) ثبات کنترل صفر

Rmtpnd	TxRqst	MsgLst	NewDat

شکل (۳-۱۶) ثبات کنترل یک

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

میدانهای مختلف ثبات های کنترل به شرح زیر می باشند:

۱- اعتبار دهی پیام *Message Valid (Msgval)*

اگر این بیت ری ست باشد کنترلر به بسته پیام مربوطه دسترسی ندارد ولی اگر این بیت ست باشد بسته پیام ترکیب بندی شده و برای تبادلات ارتباطی آماده است. قبل از اینکه بیت *Init* در ثبات کنترل ری ست شود بیت *Msgval* همه بسته های پیام بدون استفاده باید ری ست شود.

۲- فعال کننده وقفه ارسال و دریافت *(Transmit (Receive) Interrupt Enable)*

اگر بیت *TXIE(RXIE)* ست باشد بعد از ارسال(دریافت)موفق یک فریم وقفه ای تولید خواهد شد.

۳- وقفه معلق *Interrupt Pending (Int Pnd)*

اگر بیت *Int Pnd* یک بسته پیام ست باشد این بسته پیام یک وقفه تولید کرده است که منتظر سرویس و یا در حال سرویس می باشد.

۴- درخواست داده معلق *Remote Frame Pending (Rmt Pnd)*

هنگامی که این بیت ست است نشان می دهد که یک گره دور درخواست داده نموده است ولی به علت اینکه داده تا کنون ارسال نشده است این درخواست معلق و منتظر سرویس است.

۵- درخواست ارسال *Transmit Request (TXR qst)*

با ست بودن این بیت بسته پیام باید ارسال شود. هنگامی که کنترلر *CAN* یک فریم دور دریافت می کند که در آن تقاضای ارسال این بسته پیام شده است این بیت ست می شود.

۶- *Messege Lost Msg Lst*

اگر کنترلر *CAN* یک پیام جدید دریافتی از یک گره را به داخل بسته پیام مربوطه ذخیره نماید این بیت ست می شود.

۷- به هنگام سازی توسط *CPU Updating (CPUUpd) CPU*

CPU با ست نمودن این بیت نشان می دهد که محتویات داده بسته پیام تا زمانی که این بیت ست است نبایستی ارسال شود.

۸- داده جدید *New Data (New Dat)*

اگراین بیت ست باشد کنترلر *CAN* یا *CPU* داده جدیدرادرداخل بخش داده این بسته پیام نوشته اند.

در بسته های پیام با جهت دریافت هر زمان داده جدید به داخل بسته پیام نوشته شد کنترلر آن را ست می کند. در بسته های پیام با جهت ارسال برای نشان دادن اینکه محتویات پیام بهنگام شده است *CPU* باید این بیت را ست نماید.

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

۳-۳-۵-۲- میدان داوری یا شناسه

این بخش که از ۴ بایت تشکیل شده است در بردارنده شناسه مربوط به بسته پیام است.

۳-۳-۵-۳- میدان داده

میدان داده متشکل از هشت بایت می باشد و داده ای را که بایستی در پیام داده قرار گیرد در بر دارد .

۳-۳-۵-۴- میدان ترکیب بندی

این میدان مطابق شکل (۳-۱۷) از سه بخش زیر تشکیل شده است. بخش *DLC* برابر تعداد بایت های داده پیام می باشد. بیت *Xtd* در صورت یک بودن نشان می دهد که پیام ارسالی از این بسته پیام توسعه یافته است و در صورت صفر بودن این پیام استاندارد است. در صورتی که بیت *Dir* یک باشد نشان می دهد که این بسته پیام آماده برای ارسال پیام است و در صورت صفر بودن این بیت بسته پیام، یک بسته پیام دریافتی است.

DLC (۴ بیت)	*Dir*	*Xtd*	*Reserved*

شکل (۳-۱۷) میدان ترکیب بندی یک بسته پیام

۳-۳-۶- ثبات وقفه

IntId (۸ بیت)

شکل (۳-۱۸) ثبات وقفه

مقدار ثبات وقفه، منبع وقفه را نشان می دهد. اگر یک تغییر در وضعیت کنترلر بوجود آمده باشد یک وقفه رخ خواهد داد. در این صورت ثبات فوق مقدار عدد ۱ را در بر دارد. عدد صفر نشان می دهد وقفه ای اتفاق نیفتاده است. در صورتی که ثبات وقفه، مقداری غیر از صفر و یک داشته باشد نشان می

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

دهد که بیت *Intpnd* در یکی از بسته های پیام ست گردیده است. مقدار این ثبات در حالت فوق شماره پیام به اضافه عدد ۲ است. جدول (۳-۲) ، اولویت بسته های پیام و مقدار وقفه را نشان می دهد.

Interrupt	*Register Value*
none	*0*
Status Register	*1*
Message Object 15	*2*
Message Object 1	*3*
Message Object 2	*4*
Message Object 3	*5*
Message Object 12	*EH*
Message Object 13	*FH*
Message Object 14	*10H*

جدول (۳-۲) اولویت بسته های پیام و مقدار ثبات وقفه

۳-۴- دریافت و ارسال پیام.

۳-۴-۱- انواع فریم های اطلاعات قابل مبادله بین گره ها و کنترلر.

الف- فریم داده

ب- فریم دور (*Remote*)

ج- فریم خطا

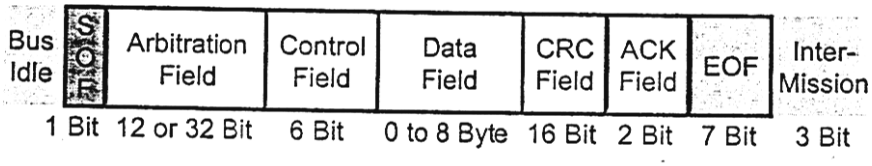
د- فریم اضافه بار (*Over load*)

در ادامه به توضیح مختصری در ارتباط با هر یک از فریم های فوق می پردازیم:

۳-۴-۱-۱- فریم داده (*Data Frame*)

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

این فریم، داده را از یک فرستنده به یک گیرنده حمل می نماید. شکل (۳-۱۹) بخش های یک فریم داده را نشان می دهد.



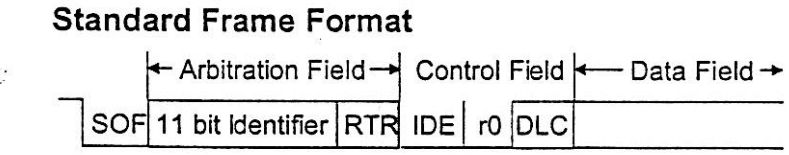
شکل (۳-۱۹) فریم داده

الف- میدان *SOF*

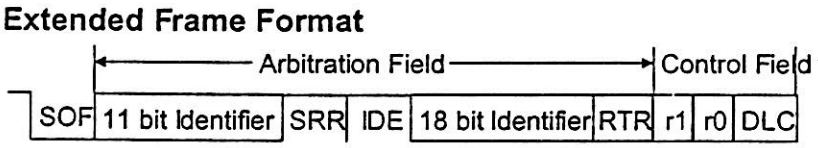
این میدان، یک بیت غالب (*Dominant*) است و برای نشالن دادن شروع یک فریم داده یا فریم دور بکار می رود.

ب- میدان داوری (*Arbitration*)

در فرمت استاندارد این میدان ۱۱ بیت شناشه، ۱ بیت *RTR* را شامل است در حالیکه در فرمت تو سعه یافته ، ۲۹ بیت شنا سه، بیت *SRR* و بیت *RTR* این میدان را ت شکیل می دهند. شکل (۳-۲۰) مطالب فوق را نشان می دهد.



الف



ب

شکل (۳-۲۰) میدان داوری الف – فرمت استاندارد ب- فرمت توسعه یافته

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

بیت *RTR* برای تمایز فریمهای داده و دور از یکدیگر بکار می رود. در فریم های داده این بیت غالب و در فریم دور مغلوب می باشد. بیت *SRR* یک بیت مغلوب است و در فرمت های توسعه یافته بعد از ۱۱ بیت شناسه قرار می گیرد. این بیت جایگزین بیت *RTR* در فرمت استاندارد است بیت *IDE* (توسعه شناسه) در فرمت توسعه یافته یک بیت مغلوب است و در میدان داوری قرار دارد و در فرمت استاندارد یک بیت غالب است و در میدان کنترل (*Control*) قرار می گیرد.

ج- میدان کنترل (*Control*)

در فرمت استاندارد این میدان از کد طول داده (*DLC*) و بیت های *RO* و *IDE* تشکیل شده است در حالیکه در فرمت توسعه یافته به جای بیت *IDE* ، بیت *rl* وجود دارد. کد طول داده تعداد بایتهای داده را در میدان داده نشان می دهد. این کد می تواند از ۰ تا ۷ باشد.

د- میدان داده (*Data*)

حداکثر طول میدان داده ۸ بایت است برای اعلام یک رخداد ، فیلد داده بایستی صفر باشد.

ه- میدان *CRC*

این میدان که از ۱۵ بیت تشکیل شده است برابر با کد *CRC* بخشهائی از فریم داده و دور می باشد .

و- میدان تصدیق (*ACK*)

میدان تصدیق از دو بیت شکاف *ACK (ACK slot)* و حایل *ACK (ACK Delimiter)* تشکیل شده است.واحد فرستنده در میدان *ACK* دو بیت مغلوب ارسال می نماید. حال اگر واحد گیرنده یک پیام معتبر را به شکل صحیح دریافت نماید با فرستادن یک بیت غالب در بازده شکاف *ACK* این موضوع را اعلام می نماید.

ز- انتهای فریم (*EOF*)

بخش انتهای فریم در فریم های داده و دور هفت بیت مغلوب است.

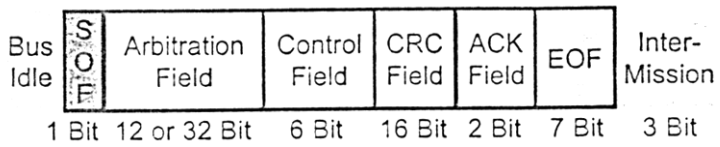
۳-۴-۱-۲- فریم دور (*REMOTE*)

اگر یک گره درخواست دریافت فریم داده با شناسه معینی را داشته باشد این فریم را می فرستد این فریم در هر دو فرمت توسعه یافته و استاندارد از شش فیلد تشکیل یافته است. دو تفاوت در فریم داده و دور وجود دارد.

الف- بیت *RTR* فریم دور، مغلوب است.

ب- در فریم دور میدان داده وجود ندارد ولی کد طول داده موجود است.

شکل (۳-۲۱) بخش های مختلف یک فریم دور را نشان می دهد.

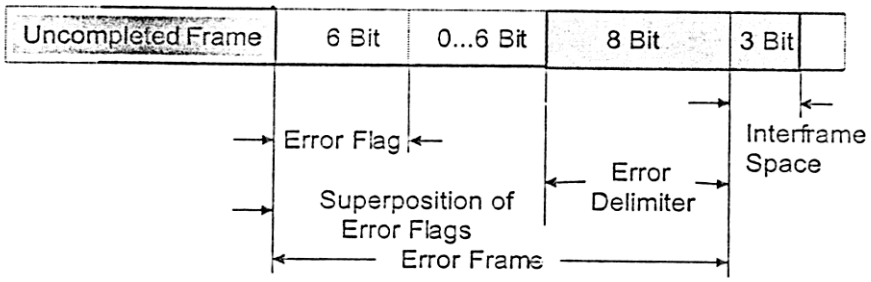


شکل (۳-۲۱) بخش های مختلف یک فریم دور

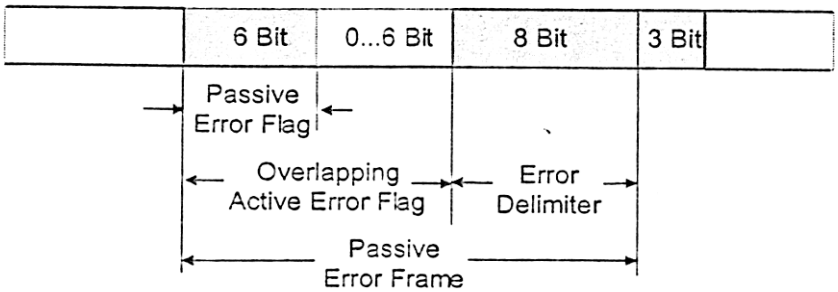
۳-۴-۱-۳- فریم خطا *(Error Frame)*

هر گره برای آ شکار نمودن خطای گذرگاه این فریم را ار سال می نماید در این فریم دو میدان وجود دارد. میدان پرچم خطا (*Error Flag*) که حاصـل از پرچم های خطا مربوط به گره های متفاوت اسـت. میدان حایل (*Delimiter*) خطا، هشـت بیت مغلوب اسـت. یک فریم خطا از یک گره که یک خطا را تشـخیص داده اسـت صـادر می گردد. حایل خطا به گره ها و کنترلر اجازه می دهد که بعد از اینکه یک خطا اعلام گردید به ارتباطات درون گذرگاه ادامه دهند. دو نوع فریم خطا وجود دارد:

۱- فریم خطای فعال، ۲- فریم خطای پسیو



الف

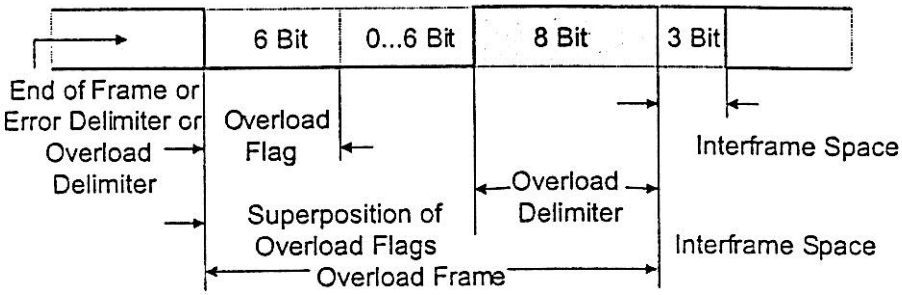


ب

شکل (۳-۲۲) فریم خطا – الف فریم خطای فعال ب- فریم خطای پسیو

۳-۴-۱-۴- فریم اضافه بار (*Over load Frame*)

به منظور فراهم نمودن یک تاخیر اضـافی بین فریمهای قبلی و بعدی داده و دور اسـتفاده می گردد. در این فریم دو میدان وجود دارد. میدان نخ ست شش بیت غالب ا ست و با پرچم خطا مطابقت می کند. حایل سر ریز هفت بیت مغلوب می باشد.



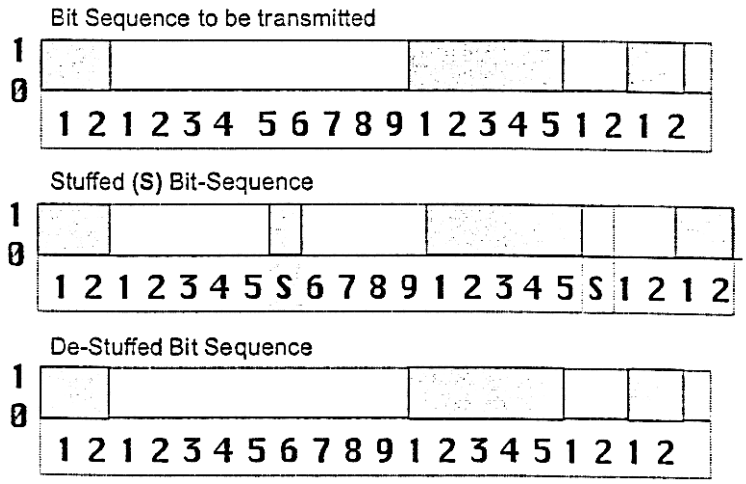
شکل (۳-۲۲) فریم اضافه بار

۳-۴-۲- بررسی کدهای خطا در تبادلات کنترلر *CAN*

۱- *De stuffing Bit Stuffing*

اگر فرسـتنده پنج بیت متوالی با ارزش یکسـان در رشـته ارسـالی مشـاهده نماید یک بیت با مقدار مکمل پنج بیت متوالی به رشـته می افزاید. این بیت *Stuff bit* به طور اتوماتیک بوسـیله گیرنده حذف می گردد. شکل زیر قائده فوق را تشریح می نماید.

WikiPower.ir



Destuffing Stuffing شکل (۳-۲۳)

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

در صورتی که در گیرنده بیشتر از پنج بیت متوالی با ارزش یکسان بین شروع فریم(*SOF*) و حایل *CRC* مشاهده گردد یک *stuff – error* رخ داده است و یک فریم خطا تولید می گردد. حایل *CRC* میدان *ACK* و میدان فریم *EOF* و فریم های سر ریز و خطا با استفاده از روش *Bit stuffing* کد نمی‌گردد.

۲- *CRC*

اگر *CRC* یک پیام دریافتی توسط کنترلر با مقدار *CRC* محاسبه شده برای آن داده توسط کنترلر یکسان نباشد یک خطا رخ داده است. خطای *CRC* و پیاده سازی آن را در فصل ۵ به طور کامل مورد بررسی قرار خواهیم داد.

WikiPower.ir

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

فصل چهارم

خلاصه ای از خصوصیات اصلی زبان VHDL

WikiPower.ir

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

۴-۱- مقدمه

همانطور که در فصل اول بیان گردید زبان *VHDL* یک زبان توصیف سخت افزاری سطح بالا برای سیستم ها و مدارات مجتمع دیجیتال می باشد. در این فصل به معرفی این زبان در حد حوصله خواننده می‌پردازیم. بدیهی است که کتابهای زیاد و پر باری برای آموزش این زبان به رشته تحریر در آمده است به منظور مطالعه بیشتر می توانید مراجع [۳۶] [۳۷] [۳۸] [۳۹] [۴۰] را مورد بررسی قرار دهید.

۴-۲- شیء (*Object*)

در زبان *VHDL* هر شی به یکی از چهار کلاس زیر متعلق می باشد:

۱- سیگنال ها (*Signals*)
۲- متغیرها(*Variables*)
۳- ثابت ها (*Constants*)
۴- فایل ها(*Files*).

سیگنال ها جهت نمایش سیم های سخت افزاری و زمان بندی همراه با آنها بکار می روند. استفاده از سیگنالها و تخصیص به آنها در ساختارهای همزمانی و ترتیبی امکان پذیر است. پورتهای ورودی و خروجی نیز به عنوان سیگنال تلقی می گردند. به منظور نمایش تخصیص مقدار به سیگنال از نماد <= استفاده می گردد.

متغیر ها برای ذخیره مقادیر موقت کاربرد دارند و مفهوم سخت افزاری ندارند. آنها فقط در ساختارهای ترتیبی زبان *VHDL* بکار می روند. تخصیص مقادیر به متغیرها به کمک نماد = : امکان پذیر است.

ثابت ها مقادیر ثابت را در ساختارهای همزمان و ترتیبی نشان می دهند. مقدار یک ثابت نمی تواند تغییر نماید.

فایل ها که در ساختارهای همزمانی و ترتیبی استفاده می شوند شی های ذخیره شده در فایل های محیط میزبان را نشان می دهند.

۴-۳- عملگرهای زبان *VHDL*

همانند زبانهای نرم افزاری در این زبان نیز عملگرهایی وجود دارد که در جدول (۴-۱) این عملگرها ارائه گردیده اند. وظیفه برخی از این عملگرها به صورت زیر است:

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

عملگرهای منطقی در توصیف مولفه های سخت افزاری در سطح گیت زیاد استفاده می گردند. برای مقایسه اسکالرها و آرایه ها عملگرهای نسبی کاربرد دارند. وظیفه عملگر & ایجاد یک آرایه بزرگتر از دو آرایه کوچکتر است.عملگرهای *REM* و *MOD* جهت تعیین باقی مانده تقسیم دو عملوند بکار می روند تفاوت انها در بکارگیری اعداد منفی است. عملگرهای *SHIFT* بردارهایی که نوع آنها در جدول بیان شده است را به تعداد دفعات مشخصی شیفت می دهد.

	عملگرها	نوع عملوند
LOGICAL	*AND , OR NAND , NOR XOR , XNOR*	*BIT Or BOOLEAN*
RELATIONAL	*= / =* *< < =* *> > =*	*ALL Types*
SHIFT	*SLL , SRL ,SLA , SRA, ROL , ORR*	*Left: BIT or BOOLEAN vector Right : INTEGER*
ADDING	*+ - &*	*Numeric Array or Array Element*
SIGN	*+ -*	*Numeric*
MULTIPLYING	*× /*	*INTEGER , REAL*
	MOD REM	*INTEGER*
MISCELLENEOUS	*ABS*	*Numeric*
	××	*Left : Numeric* *Right : Integer*

جدول (۴-۱) عملگرهای زبان *VHDL*

۴-۴- توصیف کننده های یک مولفه

همانطور که گفته شد زبان *VHDL* برای توصیف مولفه های سخت افزاری و سیستم ها استفاده می‌گردد. با توجه به شکلهای (۴-۱) و (۴-۲) در ساده ترین شکل، توصیف یک مولفه در *VHDL* شامل یک توصیف واسط (*Interface*) و یک توصیف معماری (*Architecture*) است.

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

```
ENTITY component_name IS
  input and output ports.
  physical and other parameters.
END component_name;
```

شکل (۴-۱) توصیف واسط یک مولفه سخت افزاری در VHDL

تو صیف وا سط با کلمه کلیدی *ENTITY* آغاز می گردد. در بخش تو صیف وا سط پورتهای ورودی و خروجی مولفه و م شخ صات خارجی آن از قبیل وابستگی های دمائی و زمانی معرفی می گردد. نام مولفه بعد از کلمه *ENTITY* قرار می گیرد و به دنبال آن کلمه کلیدی *IS* واقع می شود.

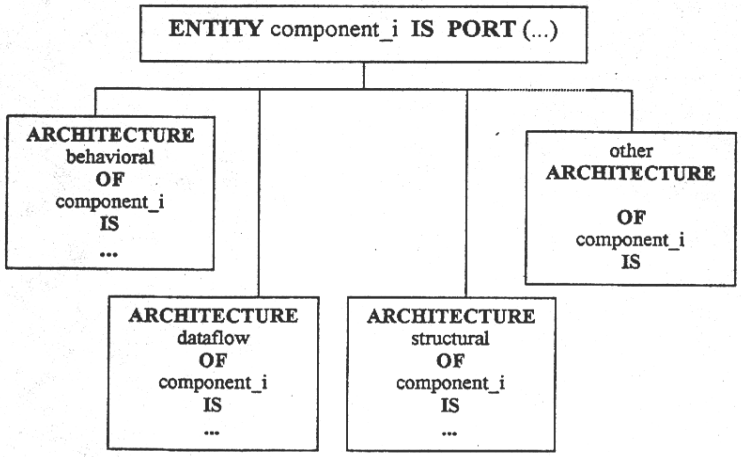
```
ARCHITECTURE identifier OF
component_name IS
   declarations.
BEGIN
   specification of the functionality of the
   component in terms of its input lines and
   influenced by physical and other parameters.
END identifier;
```

شکل (۴-۲) توصیف معماری یک مولفه سخت افزاری در VHDL

یک توصیف معماری با کلمه کلیدی *ARCHITECTURE* آغاز می شود و عملکرد یک مولفه را توصیف می کند. این عملکرد به سیگنالهای ورودی و خروجی و پارامترهای دیگری که در بخش واسط مشخص گردیده است وابسته است. در ابتدای توصیف معماری شناسه و نام مولفه ذکر می گردد. توصیف عملکرد یک مولفه با کلمه *BEGIN* شروع می شود. زبان *VHDL* به حروف کوچک و بزرگ حساس نیست ولی برای وضوح بیشتر می توان حروف بزرگ را برای کلمات کلیدی و تعریفهای استاندارد استفاده نمود.

یک مولفه ممکن ا ست دارای یک تو صیف وا سط و چندین تو صیف معماری با شد. این مو ضوع در شکل (۴-۳) نشان داده شده است.

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

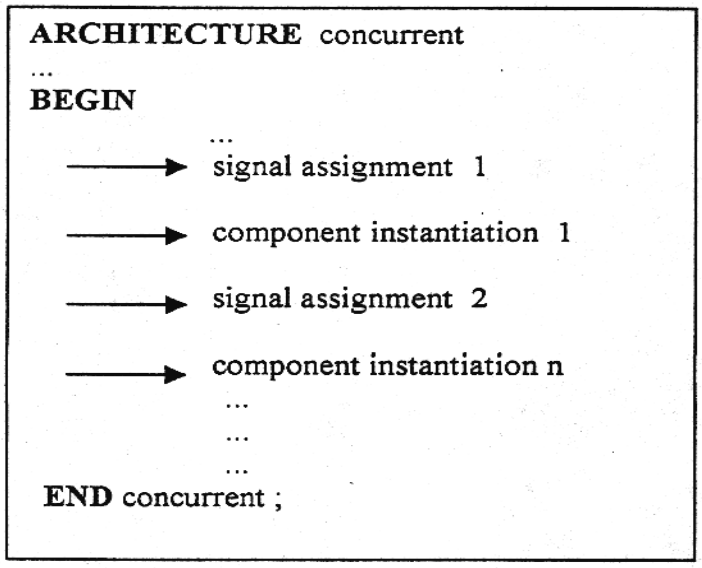


شکل (۴-۳) استفاده از یک توصیف واسط و چندین توصیف معماری برای یک مولفه سخت افزاری

همانطور که در شکل قبل نشان داده شده است یک مولفه با یک توصیف واسط می تواند دارای توصیفهای معماری در سطوح رفتاری، جریان داده ای، ساختاری و..... باشد که با توجه به نیاز یکی از این توصیفها استفاده می گردد. هر یک از این توصیفها در بخش ۴-۶ معرفی گردیده است.

۴-۵- ساختارهای همزمانی و ترتیبی

یکی از خصوصیتهای ضروری هر زبان توصیف سخت افزار همزمانی می باشد. بعد از توصیف مولفه های سیستم، عملکرد سیستم بوسیله مولفه های ارتباطی همزمان از راه پورتهای ورودی و خروجی توصیف می گردد. شکل (۴-۴) نمایش دهنده یک بلوک ساختار همزمان در زبان *VHDL* می باشد.



شکل (۴-۴) یک ساختار همزمان در زبان VHDL

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

در ساختارهای همزمان هنگامی که هر یک از مولفه ها بو سیله یکی از سیگنالهائی که آن مولفه به آن حساس است تحریک می گردد اجرا انجام می شود بطوریکه چند اجرا در چند مولفه می توانند همزمان اجرا شوند. این خصوصیت در زبانهای برنامه نویسی معمول وجود ندارد زیرا مراحل اجرای بخش های مختلف یک برنامه *C* (به عنوان مثال) بصورت ترتیبی است و نمی توان دو مرحله از این برنامه را همزمان اجرا نمود.

در زبان *VHDL* همانند سایر زبانها ساختار ترتیبی مشابه شکل (۴-۵) وجود دارد. جریان برنامه بعد از کلمه *BEGIN* آغاز می گردد و به ترتیب ادامه می یابد.

۴-۶- روشهای توصیف سخت افزار

در این زبان روشهای توصیف مختلفی برای ارائه مفهوم مورد نظر وجود دارد که عبارتند از:

۱- روش توصیف رفتاری (*behavioral description*)
۲- روش توصیف جریان داده (*data flow description*)
۳- روش توصیف ساختاری (*structural description*)

WikiPower.ir

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

```
ARCHITECTURE sequential
....
BEGIN
...
  PROCESS
  ....
  BEGIN
  ...
  ...
  IF THEN ELSE
  ...
  ...
  ...
  FOR LOOP
  ...
  ...
  ...
  END PROCESS;
  ...
END sequential ;
```

شکل (۴-۵) یک ساختار ترتیبی در زبان VHDL

فرق پایه ای روشهای ذکر شده در نزدیکی آنها به سخت افزار است و یا خلاصه بودن توصیف است. در بین روشهای توصیف بیان شده روش توصیف رفتاری بیشتر خلاصه است.

در روش ساختاری در عمل اتصالات جز به جز سیستم ارائه می گردند در حالیکه روش جریان داده و رفتاری در سطح بالاتری مدار را توصیف می کنند.

۴-۶-۱- روش توصیف ساختاری

در این روش سیستم به شکل مجموعه ای از اجزا توصیف می شود. المانهای کوچک در این روش برای ایجاد مدارات بزرگ به هم متصل می شوند. از این روش برای توصیف بلوک های پایه در سطح پایین مدار (*gate level*) و سپس توصیف نحوه اتصال بلوکهای پایه به همدیگر می توان استفاده کرد. در حقیقت در این توصیف مولفه های یک سیستم و ارتباط بین آنها مشخص می گردد. به دلیل اینکه این توصیف با سخت افزار واقعی مطابقت دارد برای طراحان سخت افزار ساده ترین توصیف از لحاظ درک و در عین حال استفاده می باشد.

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

به عنوان ساده ترین مثال، تو صیف ساختاری یک گیت *NAND* در شکل (۴-۶) نشان داده شده است.

```
ENTITY nand2 IS
 PORT ( i1,i2: IN BIT ; o1 : OUT BIT );
END nand2;

ARCHITECTURE structural OF nand2 IS
BEGIN
 o1<= i1 NAND i2 AFTER 5 NS ;
END structrual ;
```

شکل (۴-۶) توصیف ساختاری یک گیت NAND

۴-۶-۲- روش توصیف فلوی داده (*Data Flow*)

یک توصیف مابین توصیف ساختمانی و توصیف رفتاری می باشد. توصیفهای در این سطح، جریان داده بین رجیسترها و باسهای یک سیستم را مشخص می نماید. جریان داده بوسیله سیگنالهای خارجی کنترل می گردند. این سیگنالهای خارجی بو سیله ما شینهای جریان داده دیگر تولید می شوند. بعبارت دیگر نحوه جابجایی اطلاعات بین رجیسترها شرح داده می شود.

این روش بنام (*RTL- Register Transfer Language*) نیز بیان می شود. به عنوان ساده ترین مثال، در یک سیستم دیجیتال برای انتخاب و جابجائی داده در داخل باسها و رجیسترها از مالتی پلکسر استفاده می شود. توصیف جریان داده ای یک مالتی پلکسر به صورت شکل (۴-۷) می باشد.

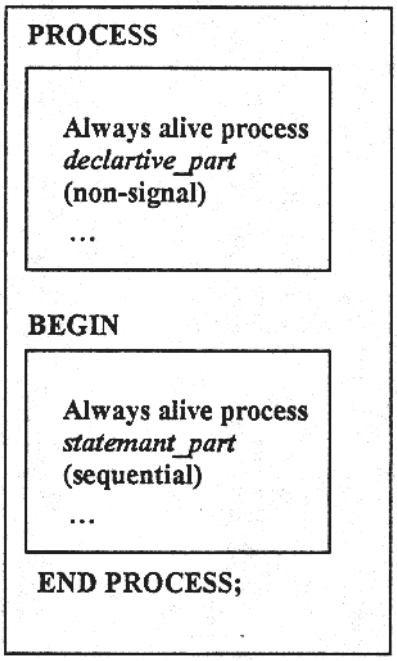
برای دریافت فایل wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

```
ENTITY mux_8_to _1 IS
  PORT ( i7,i6,i5,i4,i3,i2,i1,i0 : IN qit;
  s7,s6,s5,s4,s3,s2,s1,s0 : IN qit; z: OUT qit ) ;
END mux _8_to_1;
ARCHITECTURE dataflow OF mux_8_to _1 IS
BEGIN
WITH  qit_vector ( s7,s6,s5,s4,s3,s2,s1,s0)  SELECT
Z<='0' AFTER 3 NS WHEN "00000000",
i7 AFTER 3 NS WHEN "10000000"| "Z0000000",
i6 AFTER 3 NS WHEN "01000000"| "0Z000000",
i5 AFTER 3 NS WHEN "00100000"| "00Z00000",
i4 AFTER 3 NS WHEN "00010000"| "000Z0000",
i3 AFTER 3 NS WHEN "00001000"| "0000Z000",
i2 AFTER 3 NS WHEN "00000100"| "00000Z00",
i1 AFTER 3 NS WHEN "00000010"| "000000Z0",
i0 AFTER 3 NS WHEN "00000001"| "0000000Z",
'X' WHEN OTHERS;
   END dataflow;
```

شکل (۴-۷) توصیف جریان داده ای یک مالتی پلکسر ۸ به ۱

۴-۶-۳- روش توصیف رفتاری

در این روش مدار به صورت کلی و بدون درگیر شدن زیاد در جزییات توصیف شده و نحوه عملکرد یک سیستم بیان می شود. هدف در اینجا سرعت عمل در شبیه سازی و همچنین اطمینان از صحت طراحی بدون درگیر شدن زیاد در جزییات پیاده سازی می باشد.



شکل (۴- ۸) بلوک دیاگرام یک process statement

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

به عنوان ساده ترین مثال، توصیف رفتاری یک فلیپ فلاپ در شکل (۴-۹) نشان داده شده است.

```
ENTITY flip_flop IS
   PORT ( din,reset,enable,clk: IN BIT;qout: OUT BIT :='0');
END flip_flop;
-
ARCHITECTURE behavioral OF flip_flop IS
BEGIN
 PROCESS ( clk)
  BEGIN
   IF ( clk='0' AND clk'EVENT ) THEN
     IF reset ='1' THEN
        qout<='0';
     ELSE
        IF enable ='1' THEN
           qout <=din;
        END IF;
     END IF;
```

شکل (۴-۹) توصیف رفتاری یک فلیپ فبلپ

WikiPower.ir

۴-۷- کد نویسی قابل سنتز

در زبان *VHDL* کد نویسی قابل سنتز به معنای مدلسازی مفهوم مورد نظر با کدهای *VHDL* است به ق سمی که کدهای نو شته شده قابل سنتز با شند. در نو شتن کدهای قابل سنتز از ساختارهائی که مستقیماً به سخت افزار تبدیل نمی شوند نباید استفاده کرد.

بعضی از ساختارهای فیزیکی که در کد نویسی *VHDL* با هدف سنتز ایجاد مشکل می کنند عبارتنداز:

- تعریف متغیرهای فیزیکی در شبیه سازی دارای مفهوم هستند ولی در سنتز قابل پیاده سازی نمی‌باشند.
- در هنگام استفاده از اعداد صحیح (*I nteger*) در کدنویسی با هدف سنتز، بایستی حتماً محدوده ای برای این اعداد مشخص کرد زیرا در غیر اینصورت عدد صحیح در محدوده ۲۱۴۷۴۸۳۶۴۷ - تا ۲۱۴۷۴۸۳۶۴۷ در نظر گرفته می شود که نتیجه ای جز اشغال سطح زیاد در سنتز در بر ندارد.
- اعداد با ممیز شناور (*Floating point*) در کد نویسی قابل سنتز، نباید بکار برده شوند.

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

- در هنگام استفاده از عبارات حلقه (*Loop Statment*) استفاده از *While* در صورتی مجاز است که با یک مقدار ثابت ارزیابی شود مانند:

```
While (i<6) loop
:
:
End loop,
```

- اگر *process* دارای *sensitivity* نباشد می توان از یک *wait statement* استفاده کرد. البته این *wait* بایستی بلافاصله بعد از *begin* در *process* قرار گیرد.

```
Process
:
begin
  wait until clk' event and clk=`o`;
 :
 :
 End process,
```

۴-۸- جمع بندی

در این فصل به خصوصیات اصلی زبان *VHDL* اشاره نمودیم. توصیفگرهای واسط و معماری یک مولفه معرفی گردید. برای بیان توصیف یک سیستم سه توصیف رفتاری و جریان داده ای و ساختاری وجود دارد. از خصوصیات اصلی زبان *VHDL* وجود ساختارهای همزمان می باشد همانند سایر زبانهای سطح بالا این زبان نیز دارای عملگرهایی می باشد.

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

فصل پنجم

پیاده سازی

کنترلر گذرگاه CAN

WikiPower.ir

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

۵-۱- مقدمه

در این فصل به بررسی چگونگی پیاده سازی بخشهایی از کنترلر *CAN* و پروتکلهای که بطور مستقل قابل پیاده سازی و تست می باشند می پردازیم. توضیحات مربوط به کار تعاملی این واحدها با یکدیگر در فصل بعد همراه با توضیح و تست تعدادی از پروتکلهای *CAN* خواهد آمد. از ارائه نحوه پیاده سازی اجزا ساده تراشه مانند ثبات ها و شمارنده های معمولی خودداری شده است. در ادامه این فصل موارد زیر مورد بررسی قرار می گیرند:

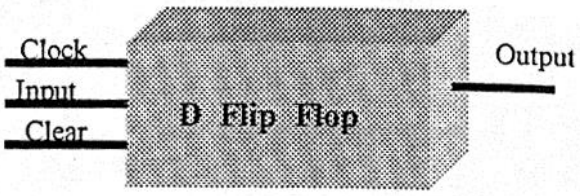
۱- جهت ارسال و دریافت پیام ها در کنترلر به یک رجیستر ارسال و دریافت نیازمند هستیم.
۲- برای پیاده سازی عملکرد سیستم مقایسه شناسه منطق طراحی پیاده شده بیان گردیده است.
۳- همانطور که در بخش پروتکلهای *CAN* مربوط به فصل سوم بیان گردید بر روی پیام های ار سالی و دریافتی در این کنترلر دو تکنیک کدینگ *CRC* اعمال می گردد که نحوه پیاده سازی آنها در این فصل بیان گردیده است.

درفصل مقدمه اشاره کردیم که اصلاحاتی درطراحی این کنترلرلحاظ کرده ایم.این اصلاحات عبارتند از:

۱- کاربر تراشه جهت محاسبه کد *CRC* می تواند آزادی عمل داشته باشد و از بین چند کد *CRC* متفاوت یکی را انتخاب نماید.
۲- تعداد بسته های پیام را افزایش داده ایم تا در صورت بزرگ بودن شبکه و افزایش تعداد گره ها تراشه را با کارائی بهتر در مدار بکار ببریم.

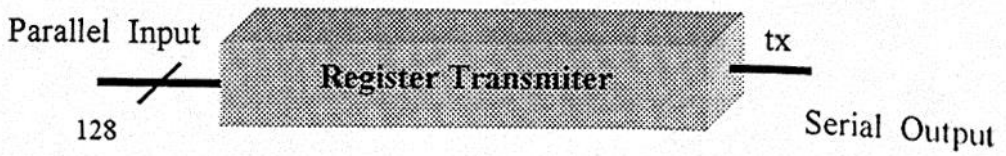
۵-۲- رجیستر ارسال و دریافت پیام در کنترلر

پس از اینکه پیام ارسالی مشخص گردید داده باید بر روی پورت سریال قرار گیرد. برای این منظور از یک شیفت رجیستر که از ۱۲۸ فیلپ فلاپ تشکیل شده است می توان استفاده نمود.



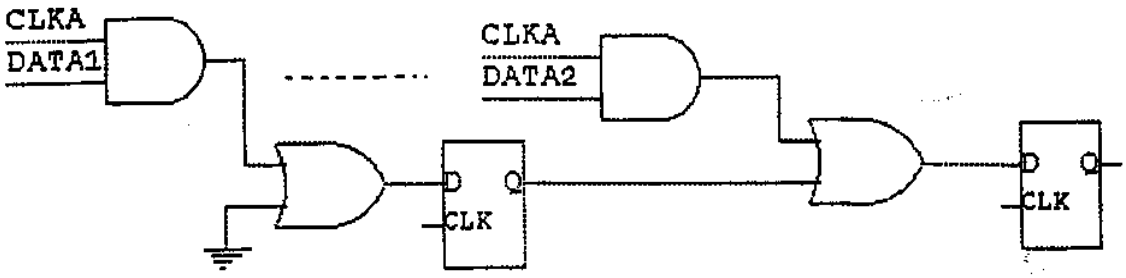
شکل (۵-۱) بلوک نمایشگر فلیپ فلاپ نوع D

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



شکل (۵-۲) بلوک نمایشگر شیفت رجیستر ارسال

قبل از عمل ارسال، کل پیام داده با یک سیکل ساعت در داخل شیفت رجیستر بار می گردد. برای محقق نمودن این بخش طرح شکل (۵-۳) بکار گرفته شد.



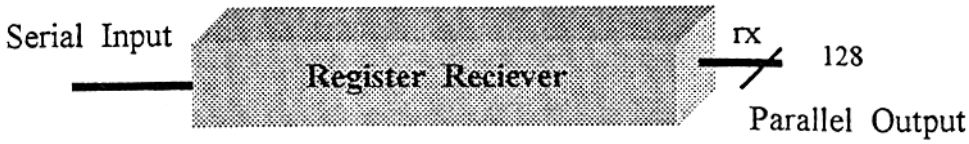
شکل (۵-۳) طرح ساختمانی شیفت رجیستر ارسال

در این طرح یک سیگنال همزمانی برای ورودی داده بطور موازی در شیفت رجیستر وجود دارد (*CLKA*) این سیگنال قبل از فعال شدن اولین پالس ساعت سیستم که برای ارسال بکار می رود فعال می گردد.

در این حالت با اعمال اولین پالس ساعت سیستم ۱۲۸ بیت داده بطور موازی در شیفت رجیستر بار می‌شود. سپس سیگنال *CLKA* غیرفعال شده و با پالسهای ساعت بعدی بیتهای داده یکی یکی به خروجی ارسالی می شود. پس از ۱۲۸ پالس ساعت محتوای شیفت رجیستر صفر خواهد بود.

برای دریافت داده از ورودی یک شیفت رجیستر با ۱۲۸ فیلپ فلاپ طراحی گردیده است که پیام وارده به این واحد وارد می شود و پس از آن بطور موازی داده موجود در شیفت رجیستر در طی یک پالس ساعت در حافظه ذخیره می گردد.

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



شکل (۵-۴) بلوک نمایشگر شیفت رجیستر دریافت

۵-۳- ثبات ماسک

در کنترلر گذرگاه *CAN* پس از دریافت یک پیام داده، به منظور شناسائی شناسه پیام، یک ثبات ماسک بکار می رود. نحوه عملکرد این واحد در فصل سوم بیان گردید.

شکل (۵-۵) بلوک نشان دهنده ثبات ماسک

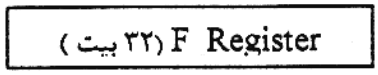
۵-۴- سیستم مقایسه شناسه ها

سیستم مقایسه شناسه ها شناسه پیام ورودی را با ۳۰ شناسه مربوط به بسته های پیام (با در نظر گرفتن ثبات ما سک) بطور همزمان مقایسه می نماید. برای این منظور از ساختاری مشابه با حافظه های تداعی گر استفاده می نمائیم. به منظور پیاده سازی این سیستم طرحی با منطق شکل (۵-۹) استفاده می گردد. در این طرح *K* محتویات بیتهای ثبات ماسک و *A* بیتهای شناسه پیام وارده و *F* بیتهای شناسه بسته پیامی است که کنترلر در حال بررسی آن می باشد [۴۱].

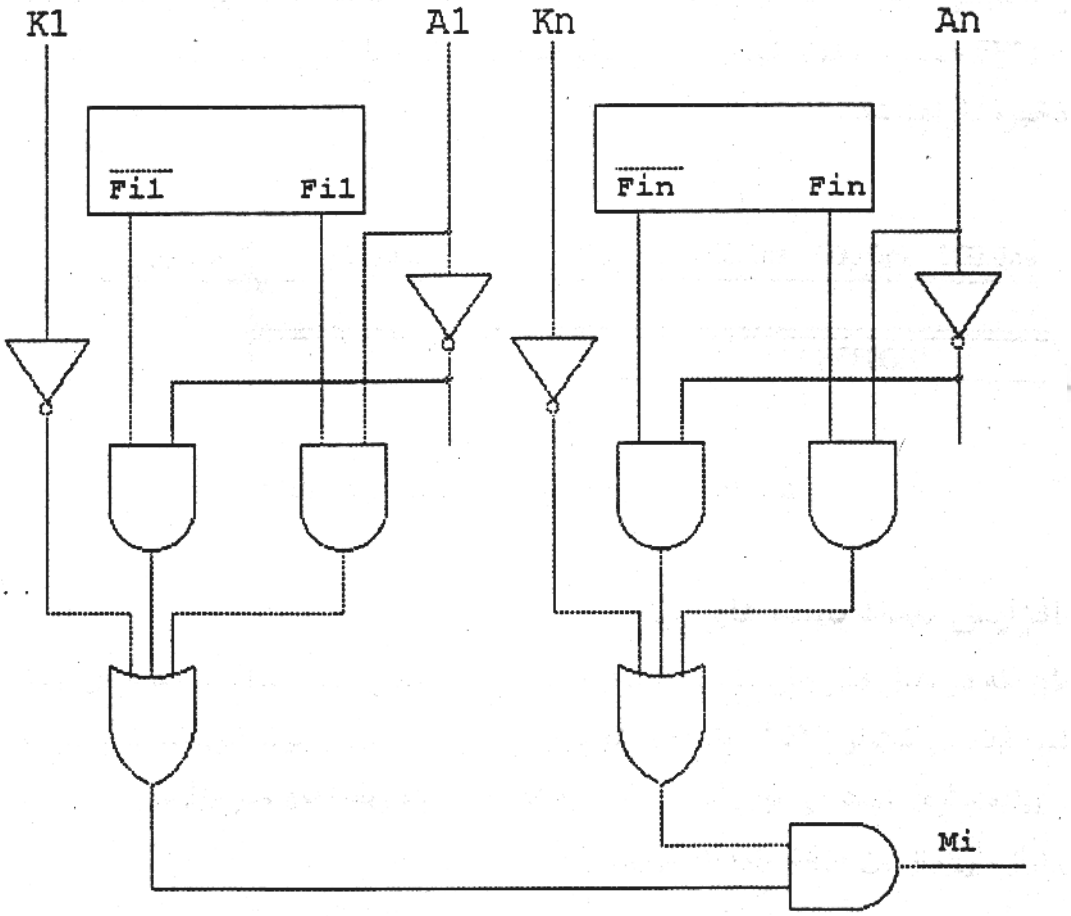
برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

شکل (۵-۶) ثبات ماسک

شکل (۵-۷) ثبات شناسه پیام داده ورودی



شکل (۵-۸) ثبات شناسه یک بسته پیام



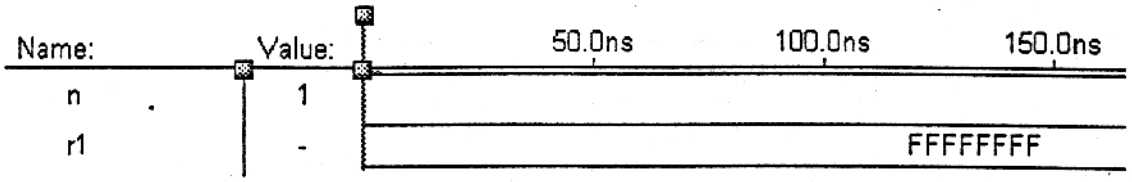
شکل (۵-۹) طراحی سیستم مقایسه شناسه ها

با توجه به اینکه شناسه پیام ۳۲ بیتی می باشد بنابراین ۳۲ واحد از بلوک فوق برای مقایسه شناسه هر بسته پیام در طراحی بکار می رود.

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

تست اول

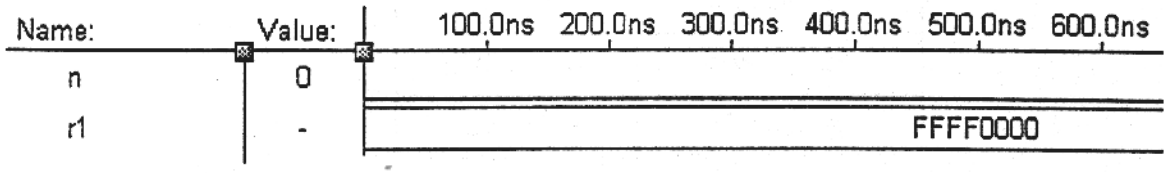
برای نمونه اگر فرض کنیم پیام داده دارای شناسه با مقدار *FFFFFFFH* باشد و مقدار ثبات ماسک و شناسه بسته پیام (*i*) نیز همین مقدار باشد پیام در این بسته پیام ذخیره می گردد. زیرا مقدار خروجی *Mi* (*n*) برابر با یک بوده و این بسته پیام، آمادگی دریافت پیام را دارد. *R1* شناسه پیام وارده می باشد.



شکل (۵-۱۰) نتیجه حاصل از تست اول سیستم مقایسه شناسه

تست دوم

به عنوان آزمایشی دیگر شناسه پیام برابر مقدار *FFFFOOOOH*، شناسه بسته پیام و فیلتر ماسک برابر با *FFFFFFFH* ه ستند. مقدار خروجی فیلتر برابر صفر بوده که ن شان می دهد این پیام در ب سته پیام مذکور ذخیره نخواهد شد.



شکل (۵-۱۱) نتیجه حاصل از تست دوم فیلتر Mask

۵-۵- افزایش تعداد بسته های پیام

همانطور که در فصل قبل بیان شد در کنترلر برای ذخیره نمودن پیام ها، بسته های پیام در نظر گرفته شده اند. در کنترلر ۸۲۵۲۷ پانزده ب سته پیام وجود دارد. در صورتیکه این کنترلر بر روی *FPGA* پیاده سازی گردد می توان این تعداد را به اندازه دلخواه زیاد نمود و در مواردی که به بسته پیام بیشتری نیاز داریم از این مزیت بهرمند گردیم.

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

در کنترلر طراحی شـــده تعداد بســـته ها را به ۳۰ افزایش داده ایم. بنابراین گذرگاه آدرس به جای هشت بیت دارای نه بیت خواهد بود.

نقشه حافظه پیام ها

در کنترلر ۸۲۵۲۷ کل فضای حافظه ۲۵۶ بایت است. ولی در کنترلر طراحی شده با توجه به اینکه ۳۰ بسته پیام وجود دارد و فضای حافظه ۵۱۲ بایت خواهد بود و نقشه حافظه برای بسته های پیام مطابق شکل (۵-۱۲) خواهد بود.

Message 1	10-IEH
Message 2	20-2EH
Message 3	30-3EH
Message4	40 –4 EH
Message5	50 –5EH
.....	.....
.....	.....
.....	.....
Message 26	1AO-1AEH
Message 27	1BO-1BEH
Message 28	1CO-1CEH
Message 29	1DO-1DEH
Message 30	1EO-1EEH

شکل (۵-۱۲) نقشه پیام ها در کنترلر طراحی شده

۵-۶- واحد محاسبه کننده کد CRC

کد (Cyclic Redundancy Checks) یکی از پر اســـتفاده ترین کدهای بررســـی خطا در انتقال سریال ناهمگام (آسنکرون) است. معمولاً این روش برای بلوکهای بزرگ داده که شامل صدها بایت است استفاده می شود. بایتهای کد CRC به آخر جریان داده متصل می گردد و همراه داده منتقل می گردد.

در مقصد بوسیله سخت افزار یا نرم بایتهای CRC و خود داده تست می شوند. این روش همچنین برای بررسی خطا هنگام ذخیره داده در دیسک کامپیوتر بکار می رود.

بایتهای CRC از فرمول زیر محاسبه می گردد:

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

$$Q(x)=M(x)\times X^n/G(x)+R(x)$$

چند جمله ای $M(x)$ جریان بیتها را نشان می دهد. این چند جمله ای در X^n که n تعداد بیتهاست ضرب می شود.

حاصلضرب این دو بایت بر یک چند جمله ای مولد نامیده می شود تقسیم می شود که خارج قسمت $Q(x)$ و باقیمانده $R(x)$ را تولید می کند. چند جمله ای باقیمانده $R(x)$ کد CRC را مشخص می کند.

به عنوان نمونه کد CRC برای جریان داده ۱۶ بیتی $4D92H$ یا 0100110110010010 به صورت زیر محاسبه می گردد:

برای بدست آوردن $M(x)$ ابتدا بیتها را برعکس می نماییم. این سری بیت به صورت زیر به ضرایب چند جمله ای تبدیل می شود که با حذف عناصر صفر آن داریم.

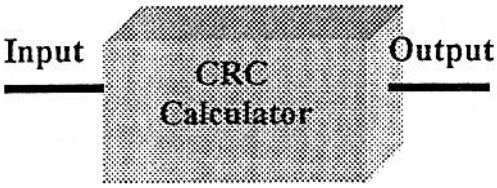
$$M(0100\ 1001\ 1011\ 0010)= X^{14}+X^{11}+X^{8}+X^{7}+X^{5}+X^{4}+X^{1}$$

سپس چند جمله ای حاصل در X^{16} ضریب کرده و آن را بر چند جمله ای مولد تقسیم می نمائیم. این تقسیم در شکل (۵-۱۴) نشان داده شده است [۴۲] [۴۳].

نتیجه عمل تقسیم را معکوس نموده و کد CRC به صورت زیر محاسبه می گردد.

$$CRC\ Code=0101\ 0101\ 1010\ 1101$$

به منظور پیاده سازی واحد محاسبه کننده کد CRC بصورت سخت افزاری از منطق طراحی شکل (۵-۱۵) استفاده می نمائیم.



شکل (۵-۱۳) بلوک نمایشگر واحد محاسبه کننده کد CRC

در این طراحی یک شیفت رجیستر و گیت های منطقی XOR بکار رفته است. در صوریتکه رشته داده به این واحد اعمال گردد محتویات رجیسترها در N امین پالس ساعت برابر مقدار کد CRC مربوط خواهد بود. N عبارت از مجموع طول رشته داده و درجه چند جمله ای مولد است. درجه چند جمله ای مولد مشخص کننده تعداد فلیپ فلاپها در شیفت رجیستر می باشد. به ازای هر جمع در این عبارت یک گیت XOR وجود دارد.

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

$$
\begin{array}{r|l}
 & X^{14}+X^{13}+X^{12}+X^{8}+X^{5}+X \\
\hline
X^{16}+X^{15}+X^{2}+1 & X^{30}+X^{27}+X^{24}+X^{23}+X^{21}+X^{20}+X^{17} \\
 & X^{30}+X^{29}+X^{16}+X^{14} \\
\hline
 & X^{29}+X^{27}+X^{24}+X^{23}+X^{21}+X^{20}+X^{17}+X^{16}+X^{14} \\
 & X^{29}+X^{28}+X^{15}+X^{13} \\
\hline
 & X^{28}+X^{27}+X^{24}+X^{23}+X^{21}+X^{20}+X^{17}+X^{16}+X^{15}+X^{14}+X^{13} \\
 & X^{28}+X^{27}+X^{14}+X^{12} \\
\hline
 & X^{24}+X^{23}+X^{21}+X^{20}+X^{17}+X^{16}+X^{15}+X^{13}+X^{12} \\
 & X^{24}+X^{23}+X^{10}+X^{8} \\
\hline
 & X^{21}+X^{20}+X^{17}+X^{16}+X^{15}+X^{13}+X^{12}+X^{10}+X^{8} \\
 & X^{21}+X^{20}+X^{7}+X^{5} \\
\hline
 & X^{17}+X^{16}+X^{15}+X^{13}+X^{12}+X^{10}+X^{8}+X^{7}+X^{5} \\
 & X^{17}+X^{16}+X^{3}+X \\
\hline
 & X^{15}+X^{13}+X^{12}+X^{10}+X^{8}+X^{7}+X^{5}+X^{3}+X
\end{array}
$$

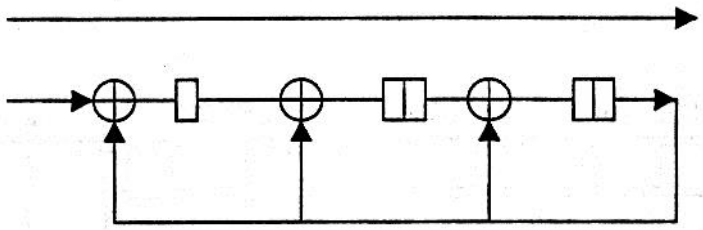
شکل (۵-۱۴) عمل تقسیم مربوط به محاسبه کد CRC

به عنوان مثال چند جمله ای مولد با عبارت زیر را در نظر بگیرید:

$$X^5+X^4+X^2+1$$

با توجه به اینکه درجه چند جمله ای از مرتبه پنج می باشد بنابراین پنج فلیپ فلاپ وجود دارد.

اگر رشته داده ۱۰ بیتی باشد بایستی بعد از اعمال رشته داده پنج بیت صفر نیز به این واحد اعمال گردد محتویات رجیسترها در پانزدهمین پالس ساعت برابر مقدار کد *CRC* داده اصلی می باشد.



شکل (۵-۱۵) بلوک ساختمانی واحد محاسبه کننده CRC با چند جمله ای مولد

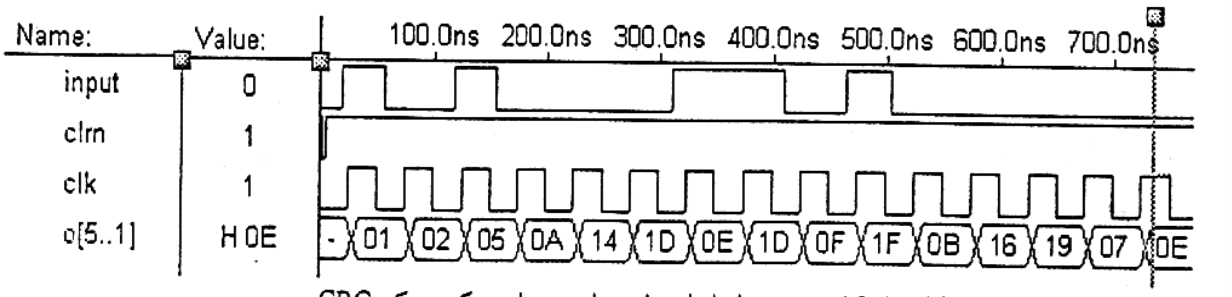
$$X^5 + X^4 + X^2 + 1$$

شیفت رجیستر

عملگر XOR منطقی

جهت حرکت داده در شیفت رجیستر

برای نمونه داده ای با مقدار *101000110100000* به واحد *CRC* اعمال می نمائیم. کد *CRC* آن در پانزدهمین پالس ساعت حاصل می گردد. نتیجه حاصل از پیاده سازی در شکل (۵-۱۶) نشان داده شده است.



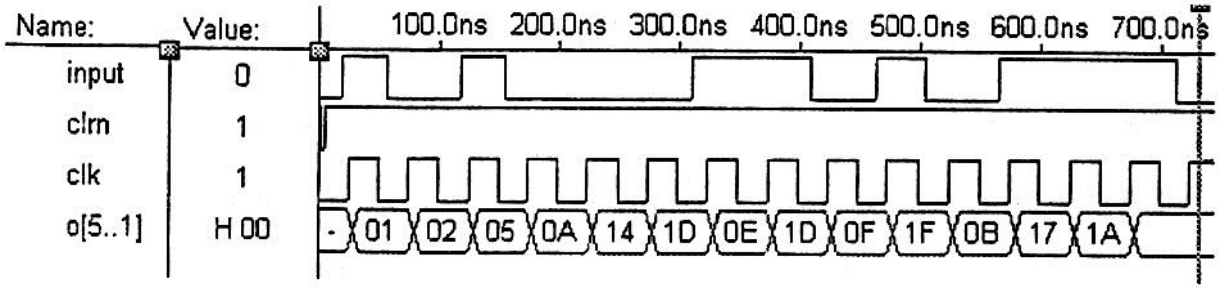
شکل (۵-۱۶) نتیجه حاصل از پیاده سازی محاسبه کننده کد CRC با چند جمله ای مولد $X^5 + X^4 + X^2 + 1$

رشته داده: *1010001101*

کد *CRC*: *0E H*

هنگامیکه کنترلر یک رشته داده را همراه با کد *CRC* آن دریافت می کند بایستی صحت داده دریافتی را بررسی نماید. در صورتیکه داده دریافتی و کد *CRC* آن بدون خطا دریافت شده باشند برای بررسی صحت رشته دریافتی از همان مدار مولد *CRC* شکل (۵-۱۴) می تواند استفاده نمود. اگر داده و کد *CRC* آن را به واحد محاسبه کننده *CRC* اعمال نماییم محتویات شیفت رجیستر در *N* امین پالس ساعت بایستی برابر صفر شود.

بعنوان نمونه اگر داده ۱۰ بیتی مثال قبل و ۵ بیت *CRC* آنرا بصورت یک رشته ۱۵ بیتی (*101000110101110*) را به مدار شکل (۵-۱۵) اعمال کنیم در پانزدهمین پالس ساعت محتویات شیفت رجیستر برابر صفر خواهد بود. نتیجه حاصل از این آزمایش در شکل (۵-۱۷) نشان داده شده است.



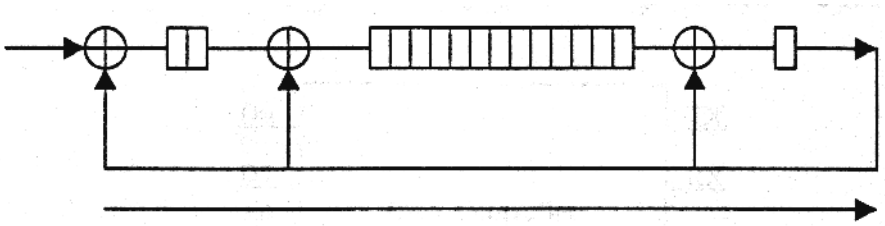
شکل (۵- ۱۷) نتیجه حاصل از اعمال داده همراه با کد CRC آن به
محاسبه کننده کد CRC
با چند جمله ای مولد $X^5 + X^4 + X^2 + 1$

علاوه بر چند جمله ای مولد قبل چند جمله ایهای مولد دیگری نیز وجود دارد که می توان به موارد زیر اشاره نمود:

$X^{16}+X^{15}+X^2+1$
$X^{16}+X^{12}+X^5+1$
$X^{16}+X^{14}+X^1+1$
$X^{15}+X^{14}+X^{10}+X^8+X^7+X^4+X^3++1$

بایستی متذکر شد که تعداد بیت کد *CRC* برابر با درجه چند جمله ای مولد می باشد و هر چه تعداد بیت *CRC* بیشتر باشد قدرت تشخیص خطای کد *CRC* بالاتر خواهد بود.

به عنوان نمونه ای کاملتر پیاده سازی چند جمله ای مولد $X^{16}+X^{15}+X^2+1$ به صورت شکل (۵-۱۸) است.

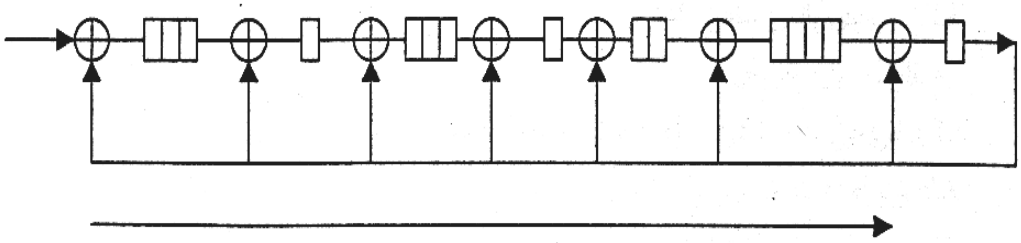


شکل (۵-۱۸) بلوک ساختمانی واحد محاسبه کننده CRC با چند جمله ای مولد
$X^{16} + X^{15} + X^2 + 1$

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

چند جمله ای مولدی که در کنترلر گذرگاه *CAN*، ۸۲۵۲۷ مورد استفاده قرار می گیرد عبارت است از $X^{15}+X^{14}+X^{10}+X^{8}+X^{7}+X^{4}+X^{3}+1$

که طراحی آن مطابق شکل (۵-۱۹) است.



شکل (۵-۱۹) طراحی واحد محاسبه کننده کد CRC در کنترلر گذرگاه CAN ، ۸۲۵۲۷ با چند جمله ای مولد

$$X^{15} + X^{14} + X^{10} + X^{8} + X^{7} + X^{4} + X^{3} + 1$$

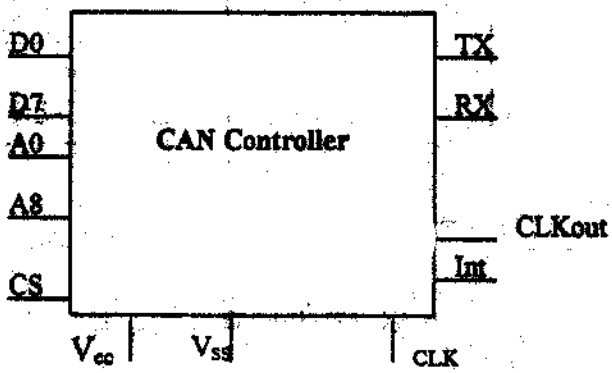
در کنترلر *CAN* طراحی شـــده این امکان وجود دارد که کاربر بتواند چند جمله ای مولد را خود انتخاب نماید. در این طراحی چهار چند جمله ای مولد در نظر گرفته ایم. این چند جمله ایها عبارتند از:

$$X^{16}+X^{15}+X^{2}+1$$
$$X^{16}+X^{12}+X^{5}+1$$
$$X^{16}+X^{14}+X^{1}+1$$
$$X^{15}+X^{14}+X^{10}+X^{8}+X^{7}+X^{4}+X^{3}+1$$

کاربر مطابق با فلوچارت شـکل (۵-۶) ارائه شـده در فصـل شـشـم می تواند چند جمله ای مولد را انتخاب نماید [۴۴]

۵-۷- دیاگرام پایه‌های کنترلر *CAN* طراحی شـده در این پروژه در شـکل(۵-۲۰) نشـان داده شـده است.

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



شکل (۵-۲۰) دیاگرام پایه های کنترلر CANطراحی شده

DO-D7: گذرگاه داده بین کنترلر و پردازنده میزبان.

A0-A8: گذرگاه آدرس بین کنترلر و پردازنده میزبان

CLK: ورودی پالس ساعت کنترلر

CLKout: پایه خروجی

TX: خروجی داده کنترلر برای اتصال به گذرگاه سریال

RX: ورودی داده کنترلر برای اتصال به گذرگاه سریال

Cs: ورودی انتخاب کنترلر توسط پردازنده میزبان

Int: خروجی وقفه

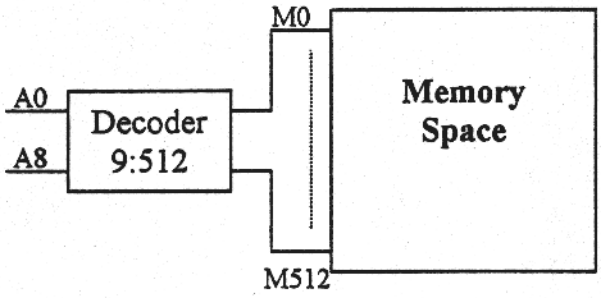
در طراحی این کنترلر نکات زیر در نظر گرفته شده اند:

الف- برخلاف تراشه ۸۲۵۲۷ که دارای ۴ حالت کاری موازی (تفاوت این حالتها فقط در تعداد خطوط آدرس و داده می باشد) و یک حالت ارتباط سری بین *cpu* و کنترلر می باشد، در طراحی این پروژه فقط حالت کاری موازی ۸ بیتی انتخاب گردید. دلیل این انتخاب معمول نبودن استفاده از حالت سری بدلیل سرعت کم و کافی بودن گذرگاه داده ۸ بیتی هنگامیکه میکروکنترلر ۸۰۵۱ یا تراشه ۸۰۱۹۶ بعنوان پردازنده میزبان استفاده می شوند، می باشد. ضمناً اگر بخواهیم از گذرگاه داده ۱۶ بیتی یا حتی ۳۲ بیتی ا ستفاده نماییم این کار با افزایش تعداد پایه های ترا شه *FPGA* انتخابی و بدون نیاز به *Multiplexing* آدرس و داده (آنطور که در ۸۲۵۲۷ انجام شده است) به راحتی قابل پیاده سازی می باشد.

ب- استفاده از ۹ خط آدرس *A0* تا *A7* بجای ۸ خط آدرس بدلیل افزایش تعداد بسته های پیام از ۱۵ به ۳۰.

عمل انتخاب هر یک از ۵۱۲ مکان حافظه *RAM* داخلی کنترلر توسط یک دیکودر ۹ به ۵۱۲ انجام می شود که بدلیل سادگی پیاده سازی از توضیح بیشتر خودداری شده است.

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



شکل (۵-۲۱) واحد دیکودر آدرس در کنترلر

۵-۸- نرم افزار مورد استفاده در پیاده سازی کنترلر *CAN*

همانطور که در فصل مقدمه اشاره کردیم نرم افزارهای زیادی برای برنامه نویسی زبان های توصیف سخت افزار وجود دارد یکی از این نرم افزارها، *Model sim* می باشد که به منظور شبیه سازی طرح سخت افزاری بکار می رود. برنامه نوشته شده در این محیط با استفاده از نرم افزار *Leonardo* به کدهای قابل سنتز تبدیل می گردد.

از نرم افزارهای دیگری که وجود دارد می توان به نرم افزار *Maxplus* اشاره نمود. این نرم افزار مخصوص تراشه های شرکت *ALTERA* می باشد. برنامه نوشته شده در این محیط قابل سنتز بوده و قابل اجرا بر روی تراشه های قابل برنامه ریزی می باشد. در این نرم افزار برنامه هایی با هر یک از زبانهای *VHDL,Verilog* و *AHDL* قابل اجرا هستند. زبان *Verilog* نیز یکی دیگر از زبانهای توصیف سخت افزار می باشد. زبان *AHDL*، زبان مخصوص خود شرکت *ALTERA* است.

استفاده از تراشه های شرکت *ALTERA* دارای مزایای زیر می باشد:

- تراشه های این شرکت ارزان است.
- به راحتی قابل دسترس می باشد.
- نسخه های متعدد نرم افزار *Maxplus* به بازار عرضه شده است.
- برنامه نوشته شده در این محیط قابل سنتز می باشد.

در این پروژه به منظور پیاده سازی کنترلر گذرگاه *CAN* از نرم افزار *Maxplus* استفاده گردید.

۵-۹- جمع بندی

در این فصل پیاده سازی بخشهایی از کنترلر که بطور مستقل قابل پیاده سازی و تست بودند بیان گردید نتایج موفقیت آمیز حاصل از تست هر بخش نیز بلافاصله ارائه و بررسی گردید. در هر قسمت اصلاحاتی را که در این طراحی اضافه گردیده بود بیان کردیم. در فصل بعد نتایج حاصله از تست تراشه

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

۸۲۵۲۷ شبیه سازی شده که از کنار هم گذاشتن اجزا ارائه شده در این فصل و اجزا تکمیلی دیگر بدست آمده است ارائه می گردد.

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

فصل ششم

نتایج وجمع بندی

WikiPower.ir

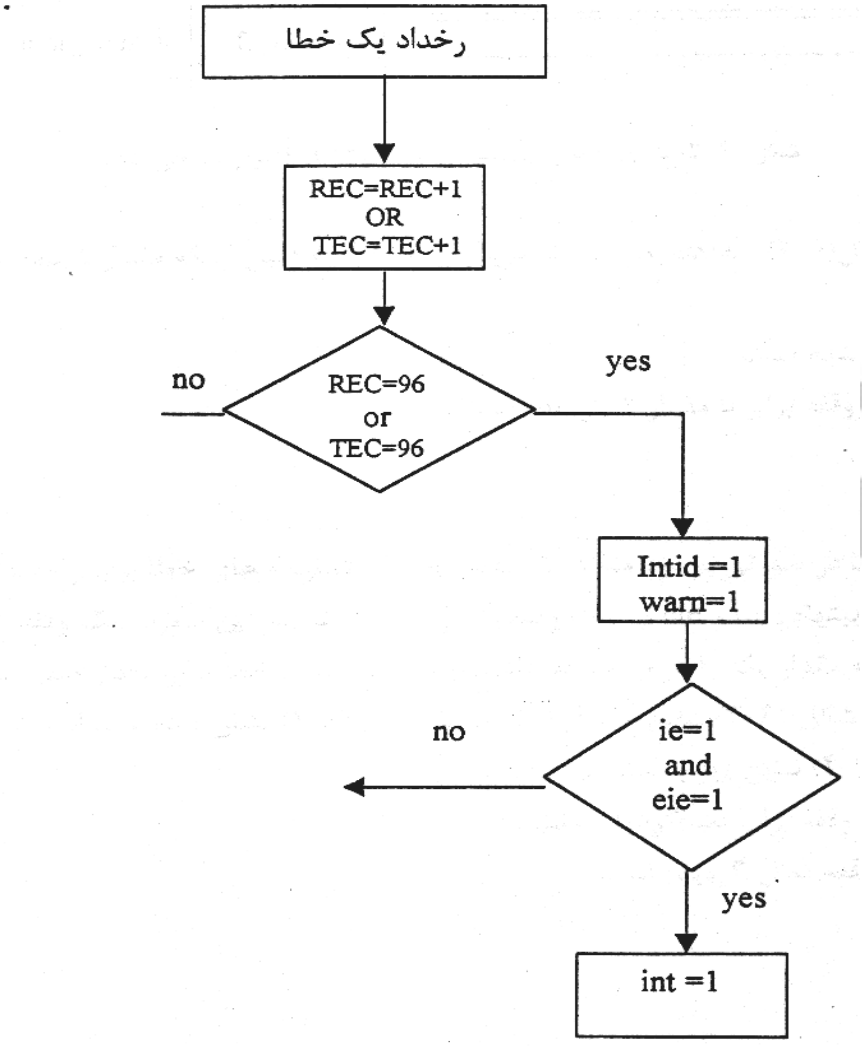
برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

۶-۱- مقدمه

در این فصل به ارائه نتایج حاصل از آزمایشهای نهائی بر روی تراشه پیاده سازی شده خواهیم پرداخت. برای بررسی عملکرد تراشه در حالت های مختلف کاری آزمایش هایی در نظر گرفته شده است و نتایج حاصل از آنها نشان داده می شود. علاوه بر این با استفاده از قراردادهای *CAN* نتیجه کلی که عبارت از نوع قطعه بکار رفته و تعداد سلول منطقی مصرف شده در این طراحی نیز بیان می شود.

۶-۲- نتایج حاصل از تست وضعیتهای مختلف کنترلر

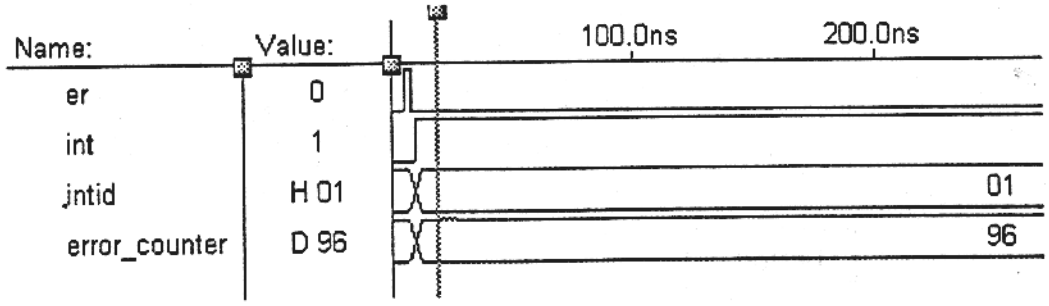
برای بررسی وضعیتهای مختلف کنترلر ابتدا وضعیت *Warn* را در نظر می گیریم. فلوچارت شکل‌های (۶-۲) نشان داده شده است.



شکل (۶-۱) فلوچارت وضعیت warn درکنترلر گذرگاه CAN طراحی شده

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

دیدیم که پس از وقوع یک خطا در کنترلر شـمارنده خطای دریافت و ارسـال یکی افزوده می گردد. حال اگر مقدار یکی از این شمارنده ها برابر ۹۶ گردد وضعیت *warn* ایجاد می گردد و بیت مربوط به آن در رجیستر وضعیت ست می گردد. در صورتیکه بیتهای *ie* و *eie* در ثبات کنترل توسط پردازنده ست گردیده باشـد پایه وقفه کنترلر فعال گردیده و یک وقفه تولید خواهد شـد. علاوه بر این به دلیل اینکه بر اساس فلوچارت شکل (۶-۱) نتایج حاصل از تست در شکل (۶-۲) نشان داده شده است.



شکل (۶-۲) نتایج حاصل از تست وضعیت warn کنترلر طراحی شده

همانطور که از شکل (۶-۲) م شاهده می گردد پس از وقوع خطا به دلیل اینکه مقدار شمارنده خطا برابر ۹۶ گردیده است.

۱- یک وقفه تولید شده است

۲- مقدار رجیستر وقفه برابر با مقدار ۱ گردیده است.

وضعیت *busoff*

وضـعیت *busoff* در صـورتی بوجود می آید که مقدار یکی از شـمارنده های خطا برابر مقدار ۲۵۶ گردد. در این صورت اگر بیتهای *ie* و *eie* از ثبات وضعیت برابر یک با شد در این صورت یک وقفه تولید خواهد شد و رجیستر وقفه مقدار یک را در بر خواهد داشت و شمارنده های خطا برابر مقدار صفر خواهند شد.

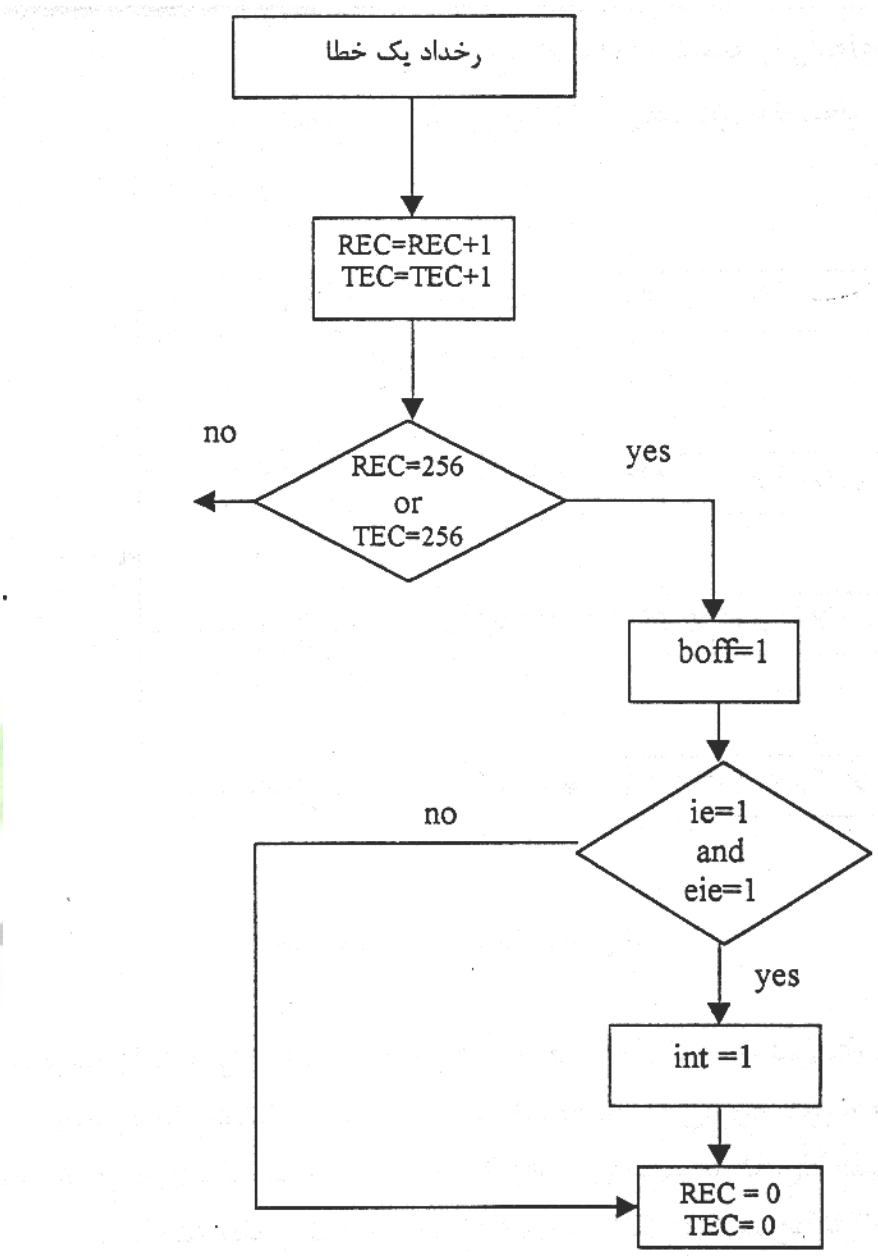
بر اساس فلوچارت شکل (۶-۳) نتایج حاصل از تست در شکل (۶-۴) نشان داده شده است.

با توجه به شکل (۶-۴) نتایج زیر بدست می آید:

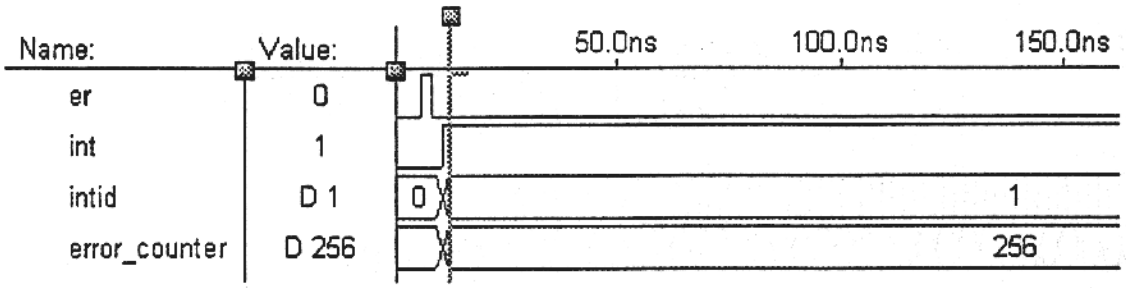
۱- مقدار رجیستر وقفه برابر یک گردیده است.

۲- پایه خروجی وقفه فعال گردیده است.

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



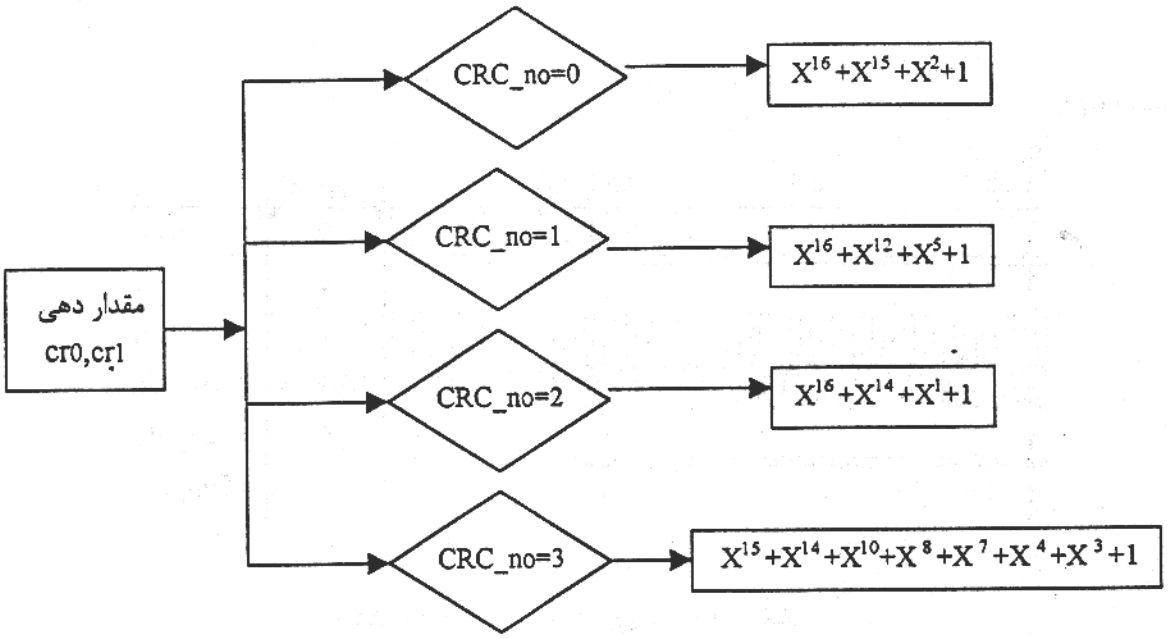
شکل (۶-۳) فلوچارت وضعیت bus off درکنترلر گذرگاه CAN طراحی شده



شکل (۶-۴) نتایج حاصل از تست وضعیت bus off کنترلر طراحی شده

۶-۳- نتایج حاصل از تست واحد *CRC* توسعه یافت

برای تست این واحد فلوچارت شکل (۶-۵) را در نظر می گیریم.

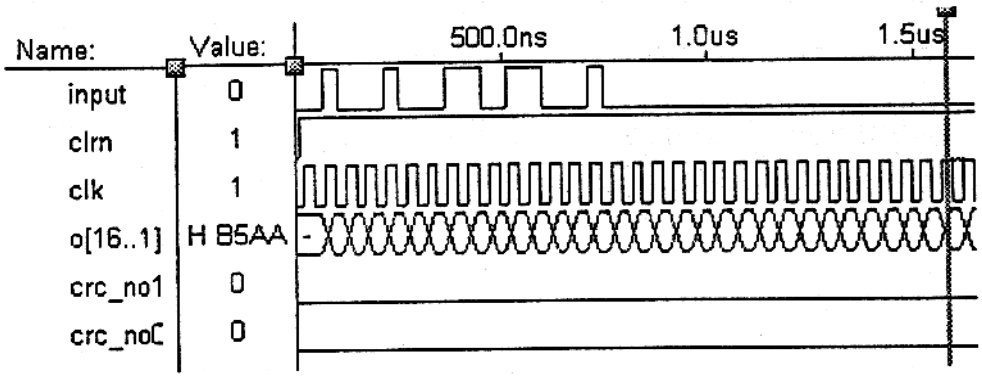


شکل (۶-۵) فلوچارت تست واحد CRC

به عنوان آزمایش یک رشته داده به واحد *CRC* اعمال نموده ایم و نتایج حاصل که عبارت از کدهای *CRC* می باشد به صورت خروجی *O[16..1]* نشان داده شده است. همانطور که در فلوچارت شکل (۶-۵) نشان داده شده است کاربر با تنظیم دو بیت *CRC-no* می تواند چند جمله ای مورد نظر خود را انتخاب نماید. نتایج حاصل از این تست در شکل های (۶-۶) تا (۶-۹) نشان داده شده است.

تست اول

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



شکل (۶-۶) نتایج حاصل از تست واحد CRC
با چندجمله ای مولد $X^{16}+X^{15}+X^{2}+1$

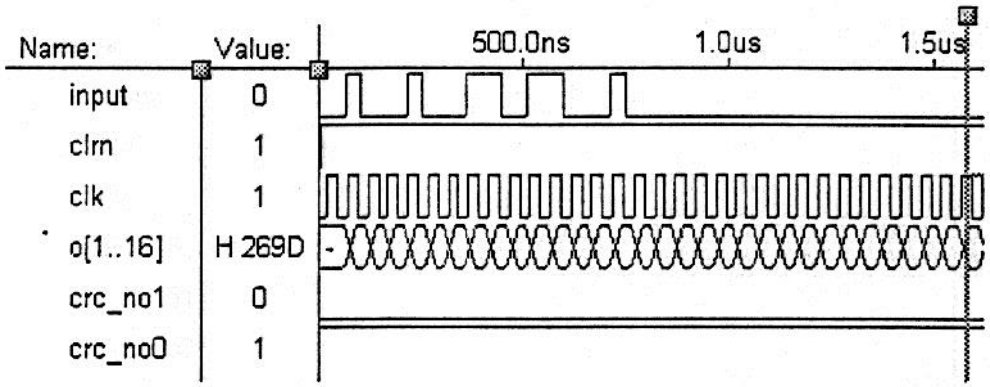
نتایج تست اول:

رشته داده ورودی: *100100110110010*

بعد از گذشت تعداد ۱۶ پالس ساعت محتویات شیفت رجیستر برابر کد *CRC* می باشد.

کد *CRC*: *B5AAH*

تست دوم



شکل (۶-۷) نتایج حاصل از تست واحد CRC
با چندجمله ای مولد $X^{16}+X^{12}+X^{5}+1$

نتایج تست دوم:

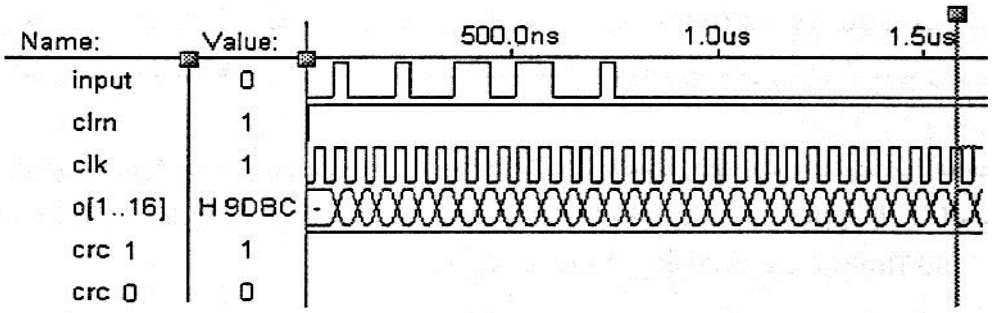
رشته داده ورودی: *100100110110010*

بعد از گذشت تعداد ۱۶ پالس ساعت محتویات شیفت رجیستر برابر کد *CRC* می باشد.

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

کد *CRC*: *269D H*

تست سوم



شکل (۶-۸) نتایج حاصل از تست واحد CRC
با چندجمله ای مولد $X^{16}+X^{14}+X^{1}+1$

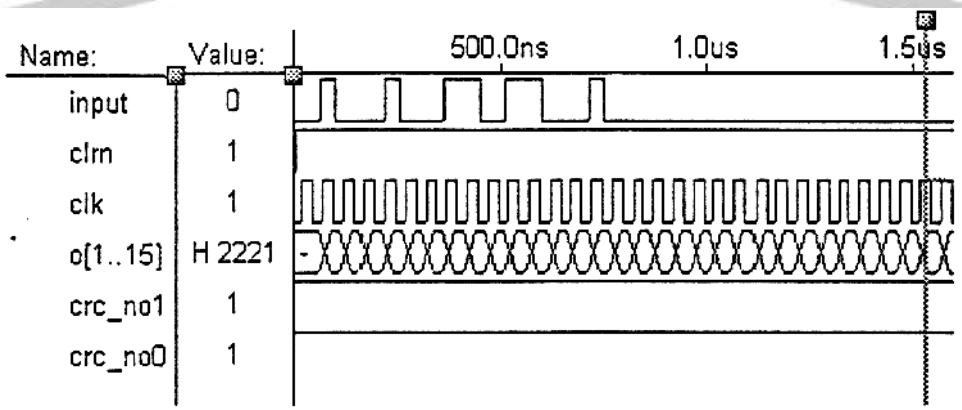
نتایج تست سوم:

رشته داده ورودی: *100100110110010*

بعد از گذشت تعداد ۱۶ پالس ساعت محتویات شیفت رجیستر برابر کد *CRC* می باشد.

کد *CRC*: *9DBC*

تست چهارم



شکل (۶-۹) نتایج حاصل از تست واحد CRC
با چندجمله ای مولد $X^{15}+X^{14}+X^{10}+X^{8}+X^{7}+X^{4}+X^{3}+1$

نتایج تست چهارم

رشته داده ورودی: *100100110110010*

بعد از گذشت تعداد ۱۵ پالس ساعت محتویات شیفت رجیستر برابر کد *CRC* می باشد.

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

کد *CRC*: *2221H*

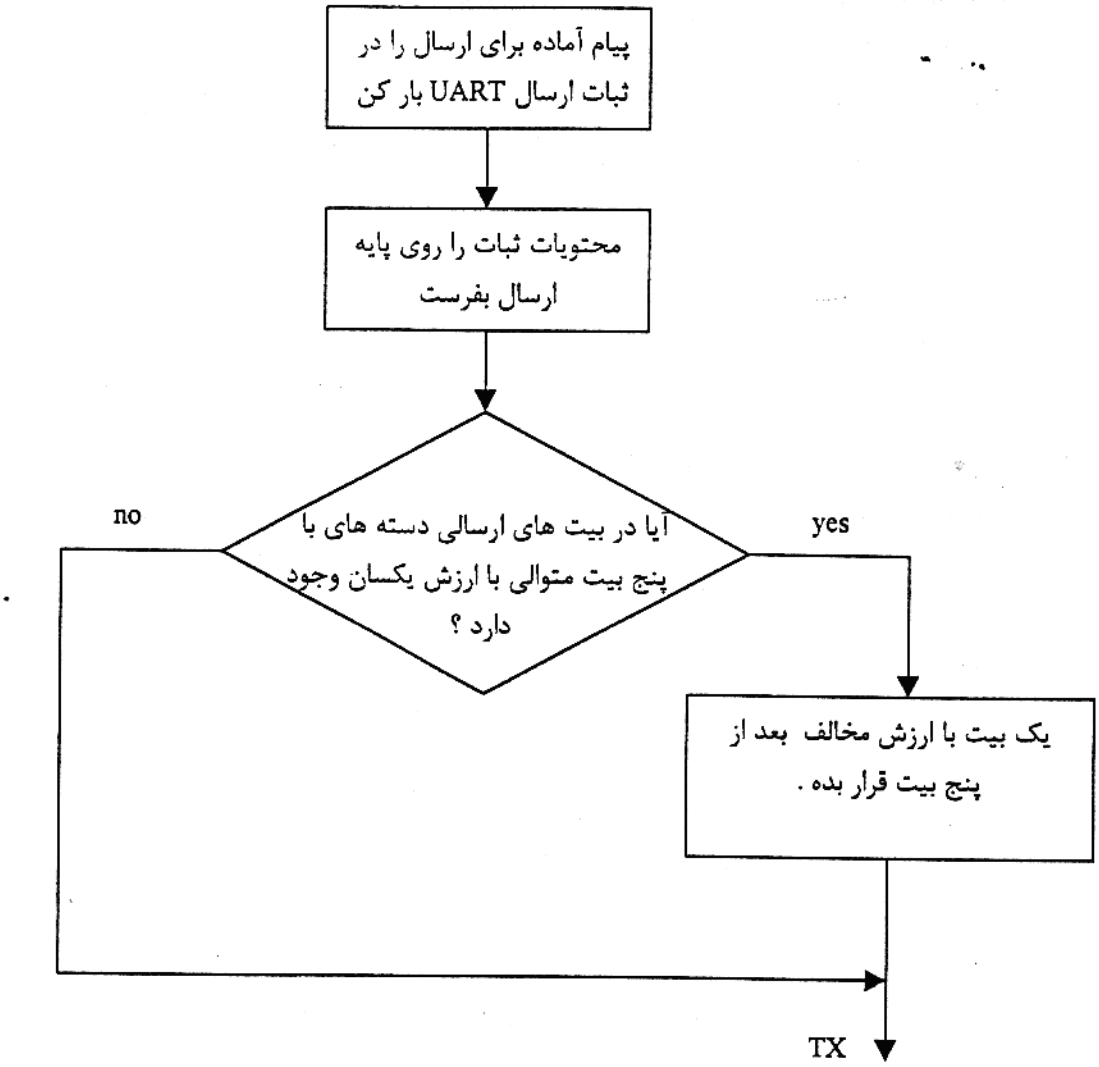
۶-۴- نتایج حاصل از تست *Stuff bit*

در فصل سوم مالد کد *Stuff bit* را بررسی نمودیم. دیدیم که هرگاه در یک رشته داده آماده برای ارسال پنج بیت با ارزش یکسان وجود داشته باشد یک بیت با ارزش مخالف با آن پنج بیت به رشته اضافه می گردد.

با توجه به این موضوع می توان فلوچارت شکل (۶-۱۰) را برای توصیف عملکرد این واحد در نظر گرفت. نتیجه حاصل از آزمایش در شکل (۶-۱۱) نشان داده شده است. رشته داده ورودی (*DATA*) و بعد از اعمال *Stuff bit* به صورت *DATA-Stuff* در می آید.

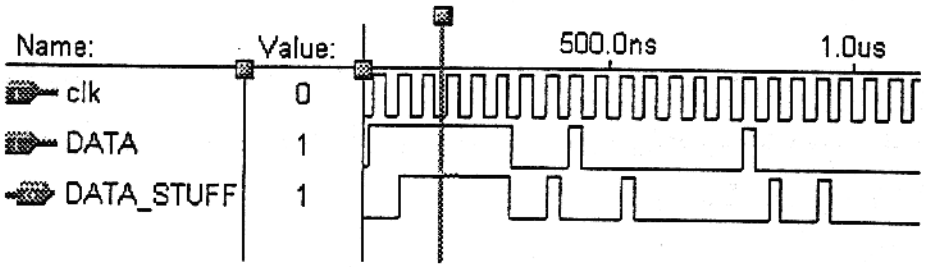
WikiPower.ir

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



شکل (۶-۱۰) فلوچارت مربوط به تست واحد Stuff bit

با در نظر گرفتن فلوچارت واحد *Stuff* نتایج شکل (۶-۸) بدست خواهد آمد.



شکل (۶-۱۱) نتیجه حاصل از تست واحد Stuff bit

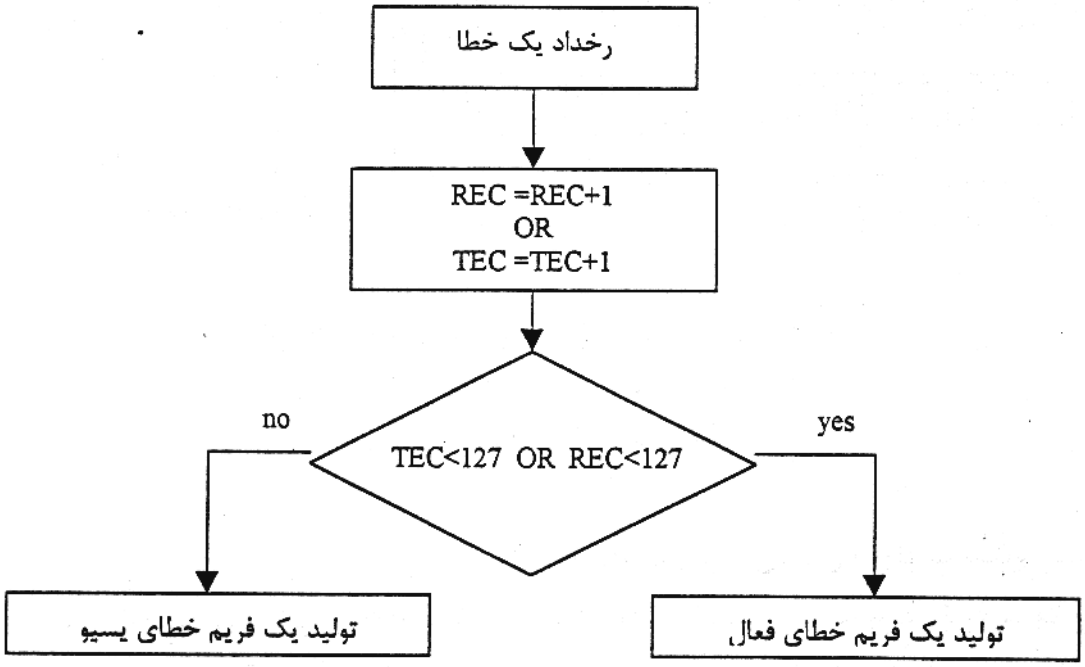
برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

نتایج زیر از شکل (۶-۱۱) بدست می آید:

رشته داده ورودی: *11110100100000101*

۶-۵- ارسال فریم خطا

در صورتیکه در سیستم یک خطا رخ دهد یک فریم خطا تولید خواهد شد. همانطور که قبلاً بیان نمودیم فریم خطا از شش بیت غالب یا مغلوب تشکیل شده است و دارای هشت بیت حایل مغلوب می باشد. با توجه به اینکه در اکثر موارد شبکه بصورت *Wire AND* ترکیب بندی می گردد وضعیت مغلوب همان حالت یک می باشد.

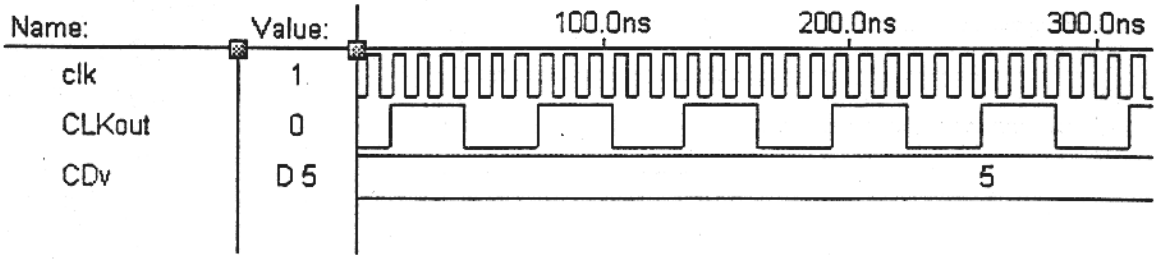


شکل (۶-۱۲) فلوچارت مربوط به تولید یک فریم خطا توسط کنترلر

در صورتیکه کنترلر در وضعیت خطا فعال باشد بخش اول فریم خطا شش بیت غالب است و در صورتیکه در حالت خطا پسیو باشد در بخش اول شش بیت مغلوب خواهیم داشت. برای بیان موارد فوق فلوچارت شکل (۶-۱۲) را در نظر بگیرید.

به منظور تست تولید خطا در کنترلر با توجه به فلوچارت فوق نتایج شکل (۶-۱۳) بدست می آید.

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

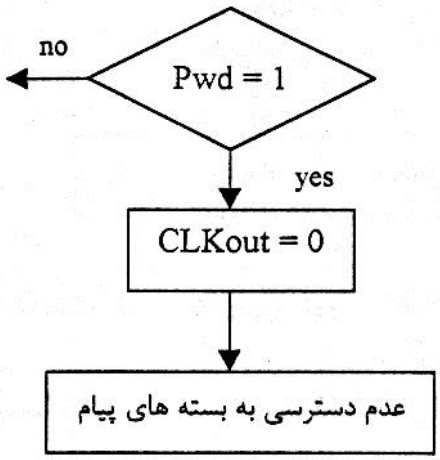


شکل(۶- ۱۴) نتایج حاصل از مقادیر مختلف بیتهای CD_v و فرکانس خروجی CLKout

همانطور که از تست بالا مشاهده می شود به ازای مقادیر مختلف بیتهای *CDv* فرکانسهای متناسب با آن بیتها در خروجی *CLKou* تولید می گردد.

۶-۷- بررسی عملکرد حالت *pwd* و *Sleep*

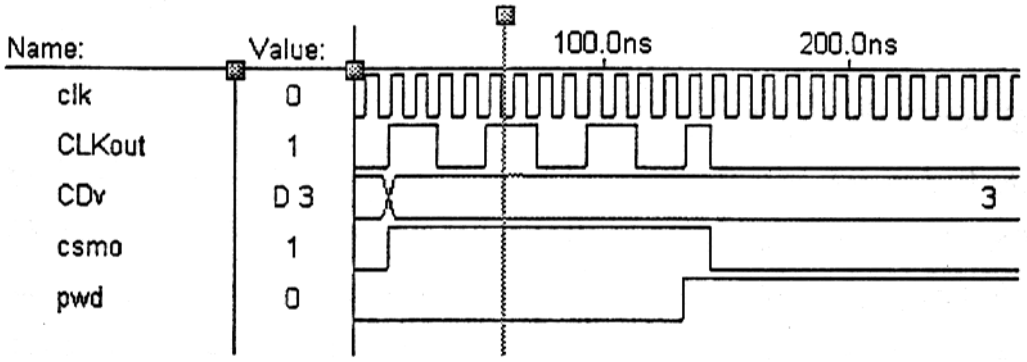
در صورتی که کنترلر در مود *pwd* قرار گیرد فرکانس خروجی پایه *CLKout* تراشه برابر صفر بوده و به هیچ کدام از بسته های پیام دسترسی وجود ندارد. برای توصیف این حالت فلوچارت شکل (۶-۱۵) را در نظر بگیرید.



شکل (۶- ۱۵) فلوچارت مربوط به حالت pwd کنترلر

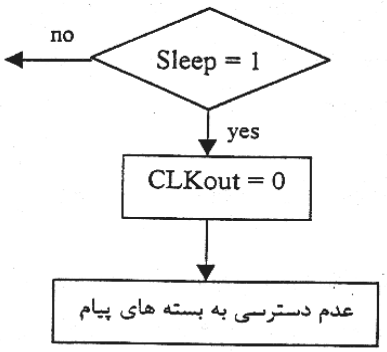
نتایج حاصل از تست این حالت کنترلر در شکل (۶-۱۶) نشان داده شده است. کنترلر طراحی شده در صورتی که در این حالت قرار گیرد سیگنال *CSMo* برابر صفر بوده و امکان دسترسی به هیچ یک از بسته های پیام وجود ندارد.

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



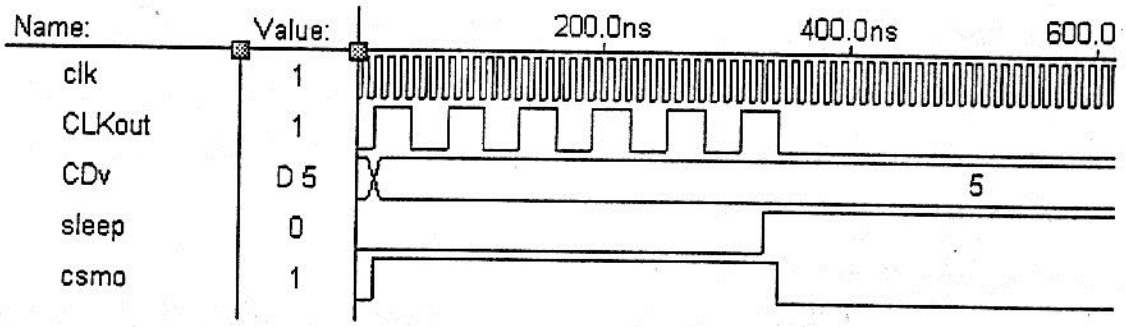
شکل (۶- ۱۶) نتیجه حاصل از حالت pwd کنترلر

در حالت *Sleep* نیز همانند حالت *pwd* فرکانس پایه خروجی *pwd* برابر صفر بوده و به هیچ یک از بسته های پیام دسترسی وجودندارد که فلوچارت شکل (۶-۱۷) نشان دهنده این حالت می باشد.



شکل(۶-۱۷) فلوچارت بیان کننده حالت Sleep کنترلر

نتایج حاصل از وضعیت *sleep* نیز در شکل (۶-۱۸) نشان داده شده است.



شکل (۶- ۱۸) نتیجه حاصل از حالت sleep کنترلر

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

۶-۸- نتایج مربوط به پیاده سازی سخت افزار روی تراشه

پیاده سازی کنترلر بر روی قطعه *EPF10K130* انجام پذیرفت و ۵۵۷۰ سلول منطقی از مجموعه سلول موجود بر روی تراشه برای این پیاده سازی استفاده گردید. کنترلر با فرکانس کاری حداکثر ۲۰ مگا هرتز می تواند عمل نماید. بعضی از اجزا داخلی کنترلر دارای فرکانس کاری بالاتری هستند اما بخشهایی مانند واحدهای محاسبه کننده کدهای *CRC* و *Stuff* حداکثر با فرکانس ۲۰ مگا هرتز می توانند عمل نمایند که در عملکرد کلی کنترلر تاثیر می گذارد و در نتیجه فرکانس کاری کل کنترلر برابر ۲۰ مگا هرتز خواهد بود.

۶-۹- نتیجه گیری و پیشنهادات برای ادامه کار

در این تحقیق با مفهوم *CAN* آشنا شدیم. کنترلر قواعد دریافت و ارسال داده در این شبکه را بررسی نمودیم. همانطور که گفته شد تراشه های قابل برنامه ریزی و زبان توصیف سخت افزار *VHDL* برای پیاده سازی طرحهای سخت افزاری دیجیتالی بسیار مناسب هستند. به ارائه نحو پیاده سازی کنترلر *CAN* بر روی *FPGA* پرداختیم. اطلاحاتی در عملکرد کنترلر لحاظ نمودیم که نتایج ارائه شده موفقیت این اصلاحات را نشان می دهد.

به عنوان پیشنهاد کار برای آینده می توان به موارد زیر اشاره نمود:

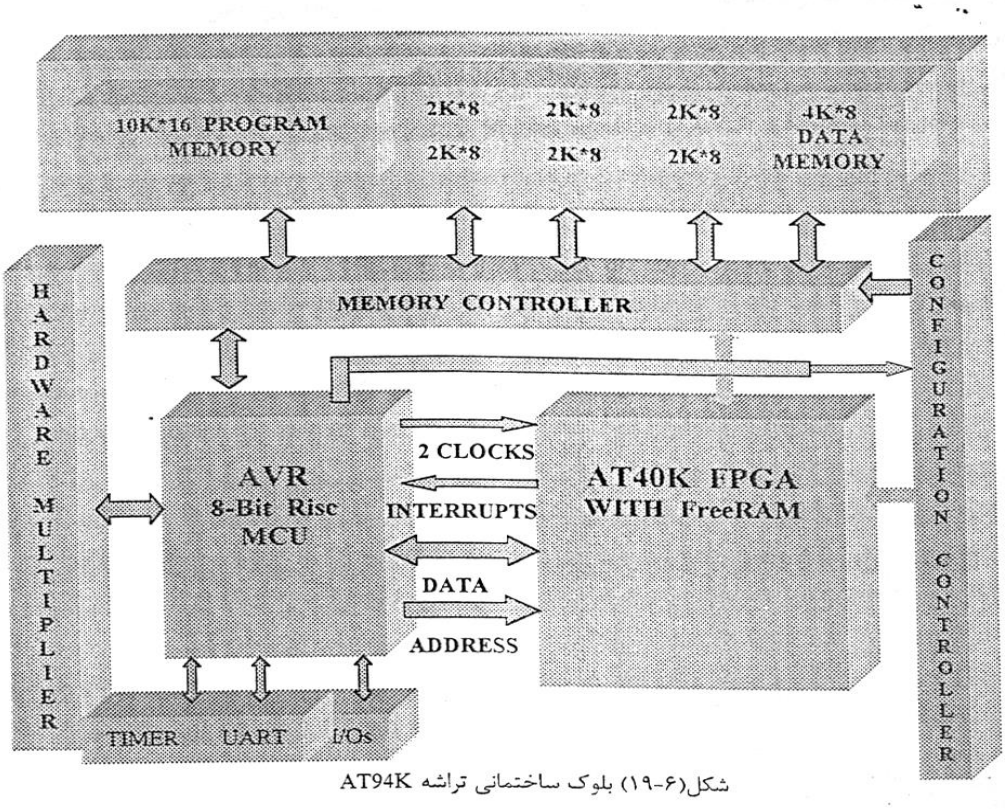
- همانطور که بیان شد کنترلر *CAN* برای هدایت و برنامه ریزی به یک پرو سسور یا میکر کنترلر نیاز دارد. با استفاده از توصیف رفتار این دو تراشه یک کنترلر طراحی نماییم که به پروسسو نیاز نداشته باشد.
- کد *CRC* که در کنترلر مورد استفاده قرار گرفته است یک کد تشخیص خطا می باشد. به منظور افزایش عملکرد کنترلر از یک کد با قابلیت تصحیح خطا استفاده نماییم.
- با توجه به اینکه هر یک از گره های شبکه *CAN* یک فرستنده- گیرنده با امکانات مثل مولد *CRC* یا *Stuff-bit* و .... می باشند، می توان گفت که اعمال یک *Transceiver* شبکه *CAN* زیر مجموعه ای از اعمالی است که کنترلر *CAN* انجام می دهد. تراشه ای..... نمونه ای از این *Transceiver*ها هستند. به راحتی می توان با خلاصه نمودن و حذف بعضی از عملیات کنترلر طراحی شده هر یک از تراشه های فوق را نیز طراحی نمود.
- پیاده سازی کنترلر با استفاده از ادوات *FPSLIC*

FPSLIC (Field programmable System Level Integrated Circuits)

خانواده *AT94K* از جمله ادوات *FPSLIC* اســـت که دارای یک هســـته میکر کنترلر ۸ بیتی ۴۰*K* *FPGA Gate* ۳۶*K* بایت *SRAM* است. شکل (۶-۱۹) بلوک ساختمانی این تراشه را نشان می دهد.

این ساختار (*Platform*) برای پیاده سازی سیستمهای با حداقل حجم و ساخت نمونه های اولیه و آزمایشـــگاهی (*prototype*) قابل اســـتفاده می باشـــد. بنابراین معماری این خانواده برای طراحی سیستمهای روی تراشه (*System on Chip*) بسیار مناسب می باشند.

ساختار بخش *FPGA* در ترا شه *AT94K* م شابه ساختار ترا شه *AT40K FPGA* می با شند. یک مزیت خاص در معماری این تراشه قرار دادن سلول های حافظه آزاد (*Free RAM*) می باشد که در کنار هر آرایه سلولی ۴×۴ گیتی یک سلول *Free- RAM* قرار داده شده است. استفاده از این سلولها باعث می شود که تبادل اطلاعات از طریق گذرگاه با حافظه سیستم کاهش یابد و همچنین طراحی *FPGA* سریع تر و در حجم کمتر انجام گردد.



شکل(۶-۱۹) بلوک ساختمانی تراشه AT94K

بخش *FPGA* از طریق گذرگاه داده ۸ بیتی به هســـته *AVR* متصـــل می گردد. این دو قســـمت نیز با حافظه نگا شته شده بر روی ترا شه بطور جداگانه ارتباط دارند. *SRAM* موجود به سه بخش زیر تق سیم می گردد.

- ۲۰*K* بایت حافظه برنامه.

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

- ۴K بایت حافظه داده

-۱۲K بایت حافظه که می تواند بطور دینامیکی به شـــکل حافظه داده یا برنامه مورد اســـتفاده قرار گیرد. علاوه بر این قطعه فوق دارای بخشهای تایمر و پورتهای ورودی – خروجی و UART می باشد.

در صورتیکه کنترلر CAN بر روی بخش FPGA ترا شه AT94K پیاده سازی گردد از بخش AVR ترا شه به منظور هدایت و راه اندازی و برنامه ریزی کنترلر می توان ا ستفاده نمود. این پیاده سازی دارای مزیتها ی زیر است.

ارتباط کنترلر و پردازنده راه انداز از طریق گذرگاه داخلی انجام می پذیرد در نتیجه این عمل

۱- اثر نویز بر روی کل سیستم کم می گردد.
۲- سرعت کار سیستم افزایش می یابد.
۳- دو تراشـــه در داخل یک تراشـــه قرار گرفته اند در نتیجه حجم کل ســـخت افزار کاهش می یابد. [۴۵].

۴- مراجع

[1]"82527 Serial Communications Controller", Intel, 1996.

[2] "Data Link Layer", available at Am Weichselgarten 26, D-91058 Erlangen, headquarters@ can – cia. De.

[3]"Milsone of CAN history", available at: http://www.CiA . Com.

[4] "CANopen,an overview",available at: http://www. CiA .com.

[5] "CAN in passenger cares", avaigable at: http:// www. CiA. Com.

[۶] عباس وفائی مبانی تراشه های قابل برنامه ریزی دانشگاه اصفهان، ۱۳۸۰. -5

[7] Ghosh . s, "Hardware Description Language, Concepts and principles", IEEE Press, ۱999. -٦

-٧

[8]DeLima,F.G.E.,Carro,L.,Lubaszewski,M.,Reis,R., Velazco,R.,"Designing a Radiation Hardened 8051 – Like Micro-Controller",proceedings of the 13th

Symposium on Integrated Circuits and Systems Design , Grenoble- France (SBCCI'00).

[9] "Implementing Logic with the Embedded Array in FLEX 10K Devices", available at: http://www . altera . com.

[10] Kazizimierz Wiatr and Ernest Jamro,"Implementation Image Data Convolutions operations in FPGA Rconfigurable Structures for Real – Time Vision Systems",proceedings of the The Interntional Conference on Information Technology: Coding and Computing.

[۱۱] ستار میرزا کوچکی، شهرام طلاکوب" پیاده سازی سخت افزاری یک فیلتر دیجیتال *FIR* معکوس شده". دهمین کنفرانس برق، تبریز، اردیبهشت ۸۱ .

[12] Oppenhim, A. V. And Schafer, R.W."Discrete-Time Signal processing", prentice – Hall, 1998. –۸

[13]"Impementing FIR Filters in FLEX Devices", available at http:// www. altera. cam.

[14] Lorca, F.G, Kessal, L., Demigny, D. "Efficient ASIC and FPGA Implementations of IIR Filters for real time edge detection", proceedings of The 1997 International Conference on Image processing, Cergy pontoise, France (ICIP'97)

[15] "Biquad IIR Filter", available at http://www. Altera. com.

[16] Mari`A.Trenas, Juan Lo`pez and Emilio L.Zapata, "FPGA Implementation of Wavelet packet Transform With Reconfigurable Tree Structure",proceedings of The 26 th EUROMICRO Conference, Universidad de Malaga, 2000.

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

[17] "Biorthogonal Wavelet Filter Megafunction", available at http:// www. altera. com. –۹

[18] "Discrete Cosine Transform Megafunctions", available at http:// www. altera . com.

[19] "RGB2YC$_r$C$_b$8YC$_r$C$_B$2RGB Converter", available at :http:// www . altera. com.

[20]" Digital Modulator Megafunction", available at: http://www altera. com.

10- *[21] "New Bus Architectures: How CardBus Fits with IEEE 1394, USB, and PCI and Others."Intel Inc.1998, available online at: http:// :www. PC-CARD. Com/papers/new.*

11- *[22]"USB Host Controller Megafunction", available at http://www.altera.com.*

۱۲- [۲۳] سید مهدی فخرائی، فرشید رئیسی، مهدیه مهران،؛ پیاده سازی کنترلر گذرگاه *PCI* بر روی *FPGA*" پایان نامه کارشناسی ارشد، دانشگاه صنعتی خواجه نصرالدین طوسی، زمستان ۷۹.

۱۳- [۲۴] مهران شتابی، احمد اکبری، "پیاده سازی کد کننده های گفتار با استفاده از سیستم های با قابلیت پیکربندی مجدد"، یازدهمین کنفرانس برق، شیراز، اردیبهشت ۸۲.

۱۴- [۲۵] مهدی قویدل جلیسه، حسن حاج قاسم، محمد ابراهیم نژاد سلمانی، " طراحی و پیاده سازی *ASIC* کد کننده و کد بردار کد فایر"، یازدهمین کنفرانس برق، شیراز، اردیبهشت ۸۲ .

۱۵- [۲۶] محمود فتحی، صالح یوسفی. " پیاده سازی سخت افزاری الگوریتمهای سطح بالای پردازش تصویر با استفاده از پیکربندی جزئی *FPGA* در زمان اجرا"، یازدهمین کنفرانس برق، شیراز، اردیبهشت ۸۲ .

۱۶-

17- *[27] Rinker,R.,Hammes,J,"Compiling Image processing Applications to Reconfigurable Hardware",Proceedings of the IEEE International Conference on Application Specific Systems, ASAP 2000.*

18- *[28] Bios,G., Bosi,B.," High performance Reconfigurable Coprocessor for Digital Mentor Graphics User s Group, Oregan,Oct. 1997*

19- *[29] Haldar,M., Nayka, A., Choudhary,A., "FPGA Hardware Synthesis From*

20- *Matlab", 14th International Conference on VLSI Design, Bangalore, India,*

21- *January 03-07,2001.*

22- *[30] Lbrra, A.,Femandez, C., Alvarez, B. Femandez-Merono J.M.."FPGA Solution for Low Cost Applications of Real – Time Automated Visual Inspection (RT-AVI)*

23- *Systems", Dedicated Systems Magazine –2001 Q2 (http://www. dedicated –systems .com).*

24- *[31] Woods,r., "Applying an XC 6200 to Real-Time Image Processing", IEEE Design and Test of Computers, Vol.15, No.1, Jan-March 1998.*

25- *[32] Fawcett, B.K., Watson, J.,"Reconfigurable processing with Field programmable*

26- *Gate Arrays", International Conference on Application-specific System,*

27- *Architectures, and processor, Chicago, IL, 1996.*

28- *[33] "Embedded Microcontrollers", Intel. 1996.*

29- *[34] "CAN Specification 2.0 part B", available online at http://www.CAN-Cia.de.*

30- *[35] "CAN Specification 2.0 part A", available online at http://www.CAN-Cia.de.*

31- *[36] Navabi, Z., "VHDL Analysis and Modeling of Digital Systems" , 2nd edition, McGraw – Hill, 1998.*

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

32- *[37] Ashenden, P.J., "The designer s Guide to VHDL" , 2nd edition, Academic Press, 2002.*

33- *[38] Cohen, B., "VHDL Coding Styles and Methodologies " ,2nd edition, Kluwer Academic Publishers, 1999.*

34- *[39] Armstrong, J.R. Gray , F.G. , " VHDL Design Repersentation and Synthesis" , 2nd edition , Prentice – Hall PTR , 2002.*

35- *[40] Pellerin, D. and D.Taylor, D., "VHDL Made Easy" , Prentice Hall Inc, 1997.*

36- *[41] Mano, M.M., " Computer System Architecture " 3rd edition , Prentice-Hall, 1993.*

37- *[42] Mazidi, M.A. " The 80X86 IBMPC & compatible computers" , 2 nd edition , prentice- Hall, VolumeII, 1998.*

38- *[43] Anderw Tanebaum, S." computer Networks" , 3rd edition, Prentice-Hall, 1996.*

39- *[44] Nair, R ., Ryan G., Farzaneh, F., "A symbol Based Algorithm for Hardware Implementation of cyclic Redundancy Check (CRC)" , 1997 VHDL International User s Forum , Arlington, 1997.*

[45] Atmel Inc. " Configurable Lagic Data Book" , 2001.

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

ABSTRACT:

Information exchange between network elements such as CPU and monitoring transceivers is an important aspect in industrial automation and robotics Infrasuctures like standard networks and buses are necessary for communications of the elements on these systems . control Area network (CAN) and it's bus (CAN BUS) have recently been introduced and used in industrial automation systems and many ASIC chips have been put in the market as CAN controllers . One such chip which is used by CAN designers as interface between CPU and transceivers on a CAN bus is the Intel 82527 .

A new approach to design and implementation of digital circuits and Aspic's which has become very popular nowadays is the use of Filed programmable Gate Arrays (FPGA) Having behavioral description of a digital IC (which may be not possible to obtain it from the market) one could design and implement it using a hardware descriptive language.

In this project , design and implementation of 82527 CAN bus controller FPGA chips has been accomplished using FPGA and hardware description language VHDL.

Some modifications has been done to improve the functionality of the chip. Results suggest that implementation and proposed modifications have both been successful.