دانشگاه مهندسی

گروه برق

پایان‌نامه کارشناسی

گرایش: الکترونیک

عنوان: خود آموز استودیو کدنگار **TMS320C6000**

استاد راهنما: دکتر اصغر طاهری

تاریخ دفاعیه: خرداد ۹۱

مقدمه:

استودیو کدنگار توسعه فرآیند برنامه‌نویسی برای برنامه‌های کاربردی پردازش سیگنال‌های جاسازی شده و بلادرنگ را تولید و تسهیل می‌نماید. استودیو کدنگار توانایی‌های کدنگار IDE که شامل آگاهی کامل برد هدف از DSP توسط میزبان و گسترش آنالیزهای بلادرنگ است، می‌باشد.

این خود آموز با فرض اینکه شما، کدنگار شامل ابزارهای تولیدکد TMS320C6000 به همراه APIها و اتصالات برای DSP/Bios و RTDX می‌باشد دارا هستید، تهیه و تولید شده است. این خودآموز همچنین فرض می‌کند که شما برد مرجع را که شامل ابزار DSP است در رایانه خود نصب نموده اید.

اگر از شبیه ساز DSP به جای برد استفاده می‌کنید شما محدود به گام‌های موجود در قسمت دوم و چهارم هستید.

این خودآموز برخی از نکات کلیدی را معرفی می‌کند. و هدف یک توضیح کلی و فراگیر از هر ویژگی نمی‌باشد. و منظور آن آماده‌سازی شما برای توسعه DSP در استودیو کدنگار می‌باشد.

قراردادهای نوشتاری:

هسته‌ی TMS320C6000 به عنوان C6000 خوانده می‌شود.

کدنگار فایل‌هایی با پسوند s62. و h62. تولید می‌کند که می‌تواند در TMS320c6201 و TMS320C6701 استفاده شوند.DSP/Bios از دستورات شناور که توسط TMS320C6701 پشتیبانی می‌شود، استفاده نمی‌کند.

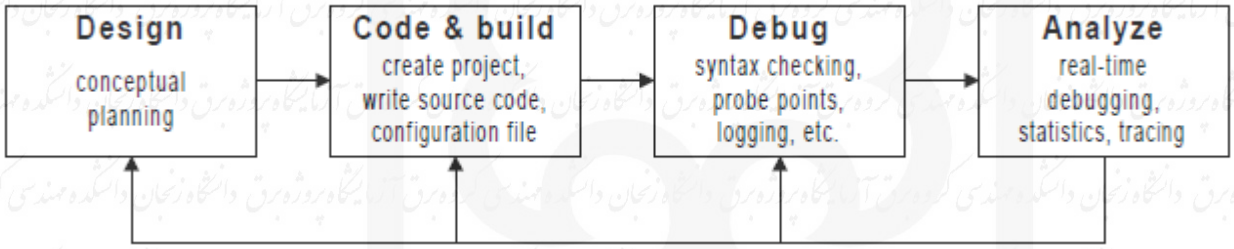
براکت‌ها یک پارامتر نوشتاری را مشخص می‌کنند. اگر شما یک پارامتر اختیاری را انتخاب کنید، می-توانید اطلاعات را داخل براکت مشخص نمایید. تا وقتی که براکت‌ها بولد نشده‌اند اطلاعات وارد ننمایید.

نگاه کلی به استودیو کدنگار

این قسمت یک نمای کلی از فرآیند توسعه نرم‌افزاری ، اجرای کدنگار ، فایل‌ها و متغیرهای استفاده شده در این قسمت می‌باشد. استودیو کدنگار فرآیند توسعه برای برنامه‌نویسانی که کار تولید و آزمایش برنامه های پردازش سیگنال بلادرنگ و جاسازی شده را بر عهده دارند، سرعت و تسهیل می‌بخشد. این برنامه ابزارهایی برای پیکربندی، ساخت، عیب‌یابی، دنباله‌روی و آنالیز برنامه‌ها را تامین می‌کند.

۱.۱ توسعه استودیو کدنگار

استودیو کدنگار ابزار های پایه‌ی تولید کد با یک سری از توانایی‌های آنالیز بلادرنگ و عیب‌یابی را فراهم می‌کند. استودیو کدنگار تمام فازهای سیکل را پشتیبانی می‌کند.



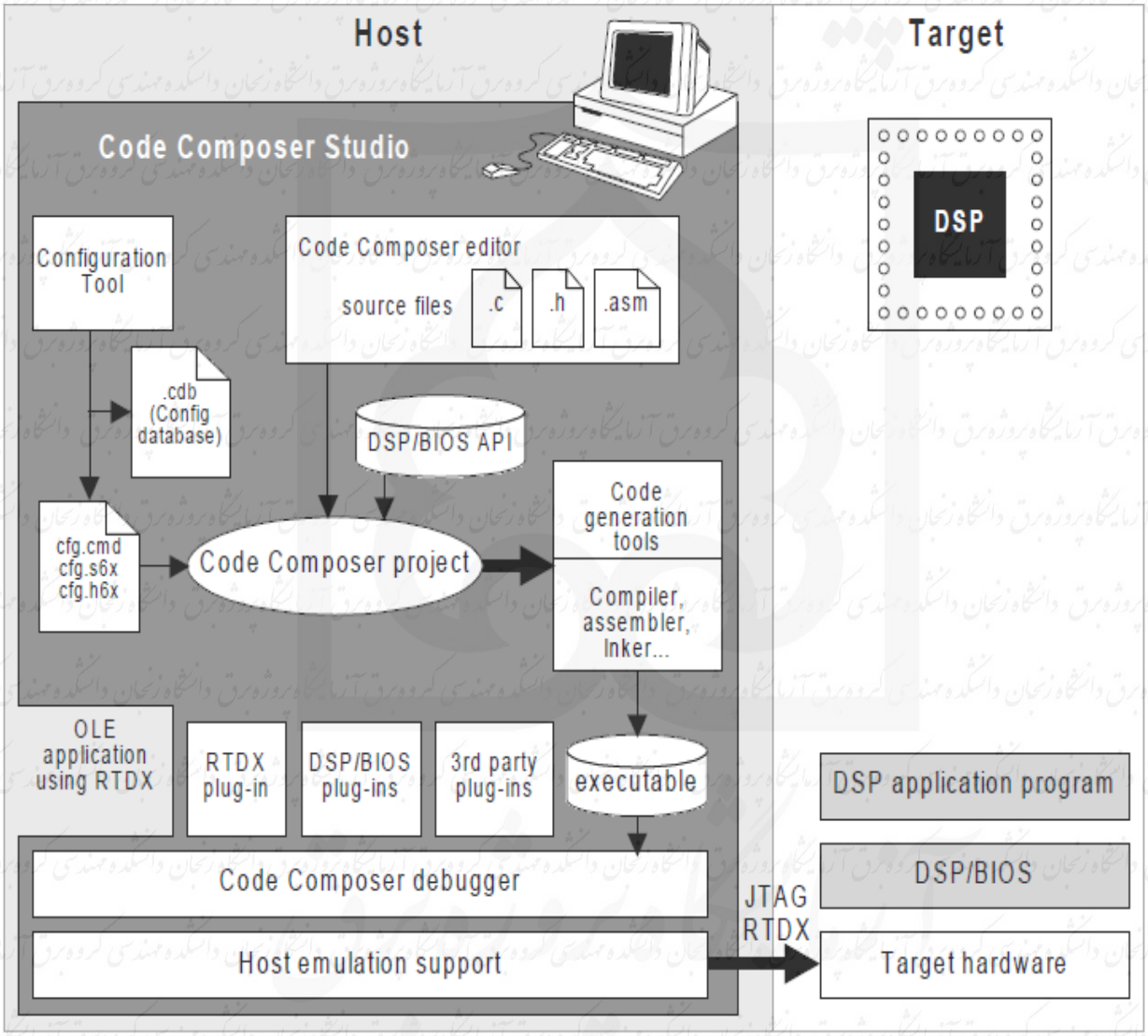
به منظور استفاده ازاین خودآموز بایستی کارهای زیر را انجام دهید:

- ☐ **نصب برد هدف و درایور نرم افزار:** نکات نصب به همراه برد را دنبال کنید و اگر از یک شبیه‌ساز یا بردی که درایور آن توسط این محصول تامین شده است استفاده می‌کنید، می‌توانید به گام‌های راهنمایی نصب این محصول مراجعه نمایید.
- ☐ **نصب استودیو کدنگار:** راهنمایی های نصب را دنبال کنید و اگر از شبیه‌ساز کدنگار و ابزارهای تولید کد TMS320C6000 استفاده می‌کنید از گام دوم در بخش ۲ و ۴ اقدام نمائید.
- ☐ **نصب و اجرای کدنگار:** برنامه‌ی نصب شده اجازه‌ی استفاده از درایور نصب شده روی برد مرجع را می‌دهد.

کدنگار شامل اجزای زیر می‌باشد:

- ☐ اجزای تولید کدTMS320C6000.
- ☐ IDE کدنگار اتصالات DSP/Bios و API
- ☐ اتصالات RTDX، رابط میزبان و API

اتصالات همانند شکل زیر کار می‌کنند:



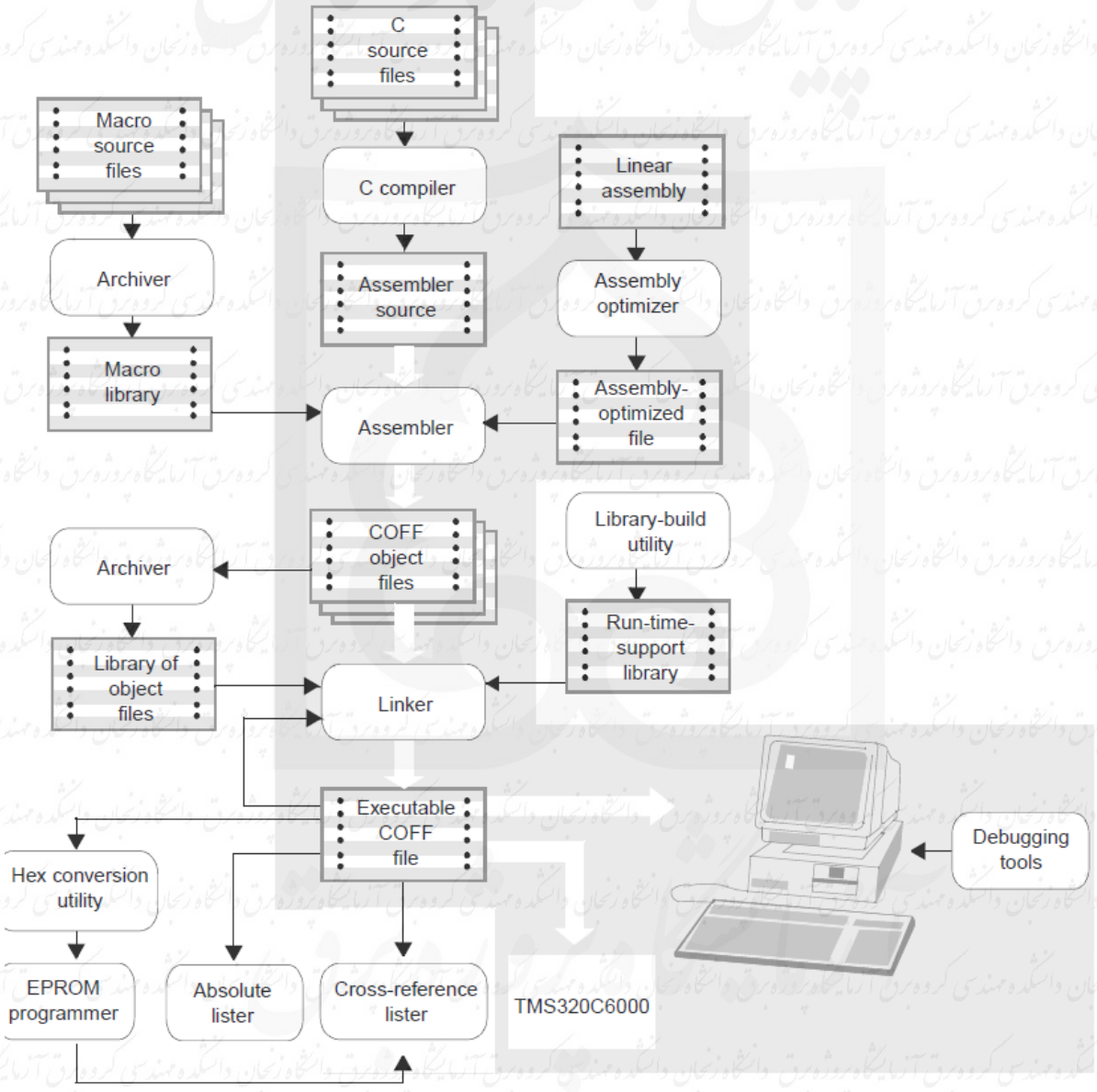
۱-۲ ابزارهای تولید کد:

ابزارهای تولید کد، توسعه کدنگار را فراهم می آورد. شکل صفحه بعد یک جریان پیشرفت نمونه را نشان می‌دهد. مهم‌ترین راه‌های توسعه نرم‌افزاری برای برنامه‌های به زبان C سایه خورده‌اند. قسمت‌های دیگر انتخابی و به منظور تسهیل فرآیند تولید می‌باشند.

لیست زیر ابزارهای شکل را نشان می‌دهند:

- **C Compiler**: سورس کد C را قبول کرده و سورس کد اسمبلی تولید می‌کند.
- **Assembler**: کد اسمبلی را به زبان ماشین ترجمه می‌کند. زبان ماشین به فرمت COFF می باشد.
- **Assembly Optimizer**: به شما اجازه‌ی نگارش کد اسمبلی خطی بدون درگیری و نگرانی در مورد ساختار لوله یا انتخاب ثبات‌ها را می‌دهد. این بخش ثبات‌ها را نصب کرده و بهینه سازی حلقه ای برای تبدیل اسمبلی خطی به اسمبلی کاملا موازی که از خط لوله نرم‌افزار بهره می‌برد، استفاده می‌کند.
- **Linker**: این بخش فایل‌های موضوعی را با یک ماژول اجرایی تلفیق می‌کند.
  همگام با تولید ماژول اجرایی عملیات جابجایی و رفع مراجع خارجی نیز انجام می‌شود.
  ورودی لینکر فایل‌های موضوعی COFF قابل جابجا شدن و کتابخانه‌های موضوعی می‌باشد.
- **Achiever**: به شما اجازه تجمیع گروهی از فایل‌ها به یک آرشیو تکی را می‌دهد که کتابخانه نامیده می‌شود.همچنین اجازه‌ی تصحیح کتابخانه به منظور پاک کردن، جابجایی، استخراج و یا اضافه کردن اعضا را فراهم آورده است.
- **Library-Built Utility**: شما می‌توانید از این بخش برای تولید کتابخانه‌های خود استفاده کنید.
- **Run-Time-Support Library**: حاوی توابع استاندارد ANSI، توابع ریاضی نقطه شناور و توابع I/O که توسط کامپایلر C پشتیبانی می‌شود، می‌باشد.
- **Hex-Conversion Utility**: یک فایل موضوعی COFF را به فرمت TI-tagged، ASCII-hex، Intel، Motorola-s، Tektronix تبدیل می‌کند. شما می‌توانید فایل تبدیل شده را به یک برنامه‌نویس EPROM بارگزاری کنید.
- **Cross-Reference Lister**: از فایل‌های موضوعی برای نشانه‌گذاری Cross-Reference، توضیح آنها و سورس آنها در سورس فایل‌ها استفاده می‌کند.

Absolute-Lister: فایل های موضوعی مرتبط را به عنوان ورودی قبول می‌کند و فایل‌های x.abs به عنوان خروجی تولید می‌کند. فایل‌های abs. برای تولید یک لیست که حاوی آدرس های مستقیم بیشتری نسبت به آدرس‌های انتصابی می‌باشد، استفاده می‌شود. بدون این قسمت تولید چنین لیستی طاقت فرساست و کارهای دستی زیادی نیاز دارد.

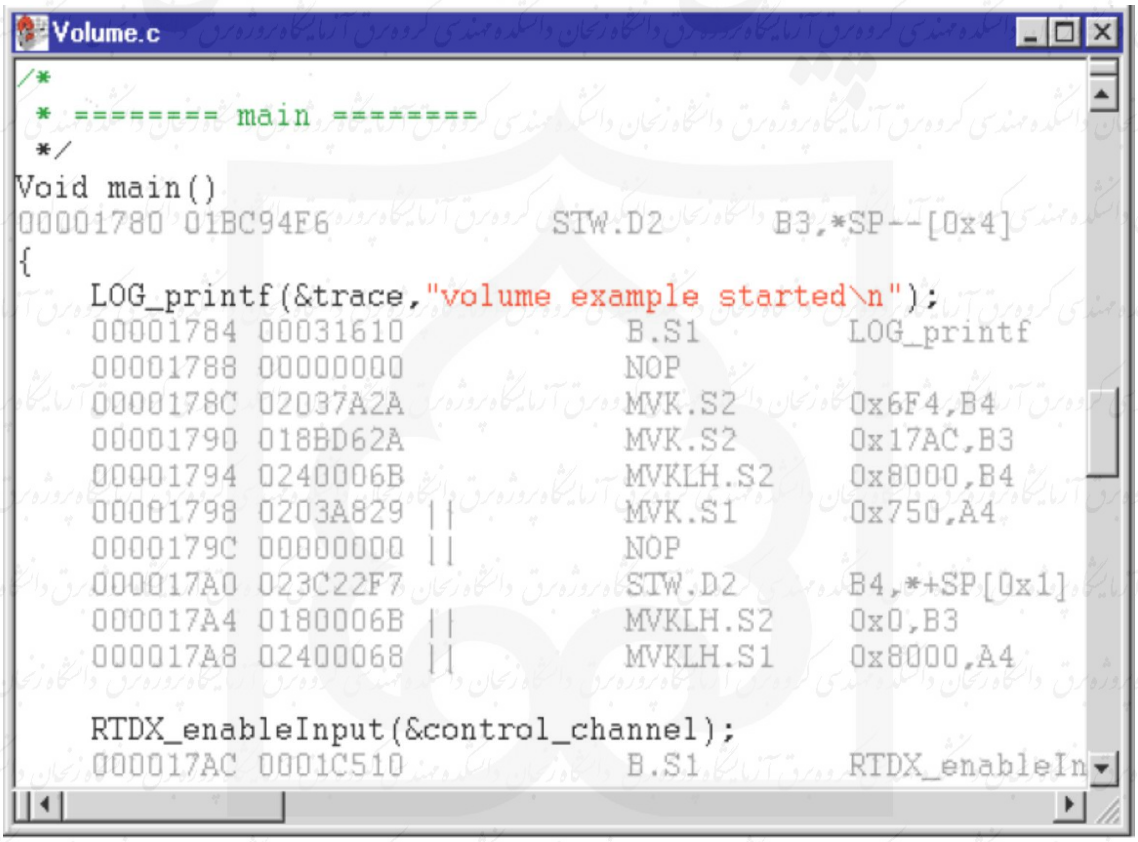


۱-۳ IDE استودیو کدنگار:

این بخش به منظور تصحیح، ساخت و عیب‌یابی برنامه مرجع DSP ساخته شده است.

۱-۳-۱ ویژگی‌های تصحیح کد برنامه:

کدنگار به شما اجازه‌ی تصحیح سورس کد C و اسمبلی را فراهم می‌کند. شما هم‌چنین می‌توانید کد C را به همراه اسمبلی نیز مشاهده فرمایید.

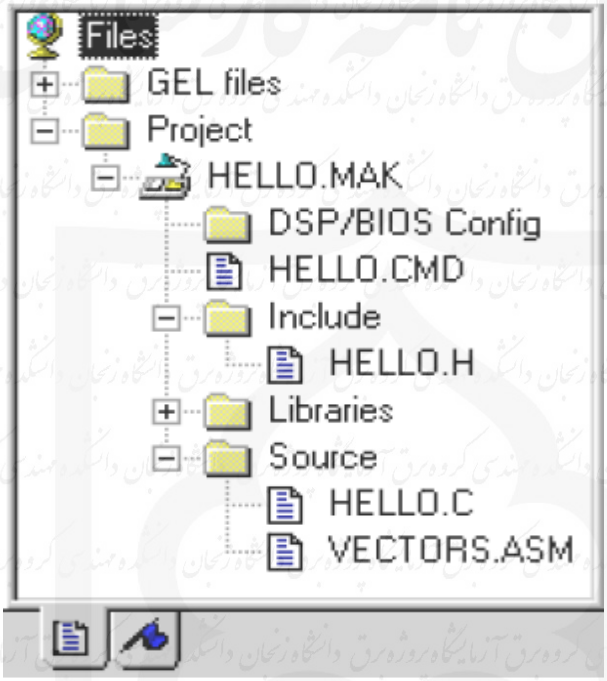


تصحیح‌گر مجتمع برای فرآیندهای زیر استفاده می شود:

- ☐ برجسته کردن کلمات کلیدی، توضیحات و رشته‌ها به صورت رنگی.
- ☐ علامت گذاری بلوک های C در پرانتز ها و براکت ها، پیدا کردن ، تطابق یا انتقال پرانتزها و کروشه ها.
- ☐ اضافه و کم‌کردن فاصله خطوط و مشخص‌کردن نقطه ی سربرگ.
- ☐ پیدا کردن و جایگزینی یک یا چند فایل، پیدا کردن نمونه قبلی و بعدی و جستجوی سریع.
- ☐ Undo و Redo کردن چند عمل.
- ☐ استفاده از کمک حساس به متن.
- ☐ تطابق انتخاب‌های دستورات صفحه کلید.

۱-۳-۲ ویژگی ساخت برنامه کاربردی:

در کدنگار شما می‌توانید یک برنامه را توسط اضافه نمودن فایل‌ها به یک پروژه تولید کنید. فایل پروژه برای ساخت یک برنامه کاربردی استفاده می‌شود. فایل‌های داخل یک پروژه می‌توانند شامل فایل‌های با مرجع C، مرجع اسمبلی ، فایل‌های موضوعی، کتابخانه‌ها، فایل‌های دستوری اتصال (Linker Command) و فایل‌های جامع باشند.



شما می‌توانید از یک پنجره برای تصریح گزینه های مورد استفاده حین کامپایل، اسمبل و اتصال یک پروژه بهره ببرید.

با استفاده از یک پروژه، کدنگار می‌تواند یک ساخت کلی یا یک ساخت تدریجی تولید کند و همچنین می‌تواند فایل‌های تکی را کامپایل کند. همچنین می‌تواند فایل‌ها را به منظور تولید یک درخت وابستگی برای یک فایل جامع برای کل پروژه تولید کند.

امکانات ساخت کدنگار می‌تواند به عنوان ساخت فایل‌ها به شیوه سنتی نیز استفاده شود.

۱-۳-۳ ویژگی عیب‌یابی برنامه:

کدنگار از فعالیت‌های عیب‌یابی زیر پشتیبانی می‌کند.

- ☐ **ایجاد نقطه‌شکست(Breaking point) با یک سری از تنظیمات گام.**

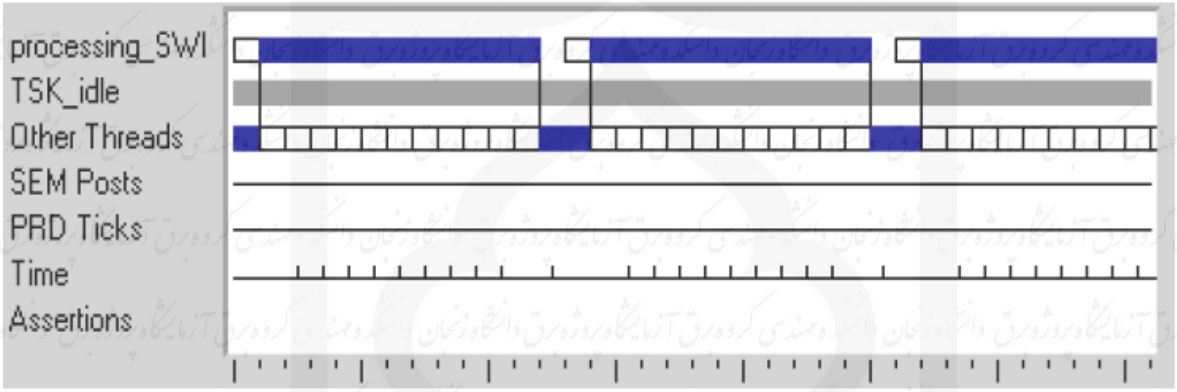
- ☐ **بروزرسانی پنجره‌ها در هر نقطه‌شکست.**
- ☐ **نظارت بر متغیر ها.**
- ☐ **نظارت و تصحیح حافظه‌ها و ثبات‌ها.**
- ☐ **نظارت بر پشته فراخوانی.**
- ☐ **استفاده از ابزار های نقطه‌کاوش (probe point) برای جاری ساختن داده‌ها به برد یا از آن. و بدست آوردن تصاویر لحظه‌ای از حافظه.**
- ☐ **ترسیم سیگنال‌های روی برد.**
- ☐ **مشخص کردن آمار و ارقام اجرایی.**
- ☐ **مشاهده دستورات C و اسمبلی در حال اجرا روی برد مرجع.**

کدنگار از زبان GEL که اجازه‌ی اضافه نمودن توابع به منوی کدنگار که به وفور استفاده می‌شود را نیز پشتیبانی می کند.

۱-۴ اتصالات DSP/Bios:

در طول فاز آنالیز توسعه نرم‌افزار ویژگی‌های خطایابی سنتی برای تشخیص مشکلات ریز که از فعل و انفعالات وابسته به زمان به وجود می‌آیند، غیرفعال می‌باشد.

اتصالات کدنگار توسط پشتیبانی DSP/Bios همچون آنالیزهای بلادرنگ پشتیبانی می‌شود. شما می‌توانید از آنها برای کاوش بصری، دنباله‌روی و نظارت بر یک برنامه DSP با کمترین تاثیر روی اجرای بلادرنگ بهره‌مند شوید. برای مثال، گراف اجرایی نشان داده شده در شکل زیر یک توالی که در آن برنامه‌های مختلف به صورت رشته ای اجرا شدند را نشان می‌دهد.



DSP/Bios API توانایی‌های آنالیز بلادرنگ زیر را نشان می‌دهد:

- ☐ **دنباله‌روی برنامه (Trace):** نمایش رویداد های نوشته شده در گزارشات مرجع و بازخورد دینامیک کنترل جریان در طول اجرای برنامه.
- ☐ **نظارت بر اجرا:** پیگیری آمار و ارقام که استفاده از منابع مرجع را بازتاب می‌کند. همچون بارگذاری پردازنده و زمانبندی رشته‌ای.
- ☐ **جریان‌سازی فایل:** اتصال فایل‌های موضوعی I/O متعلق به برد مرجع به فایل‌های میزبان.
- ☐ DSP/Bios یک زمان‌بند بر پایه‌ی اولویت را که می‌توانید در برنامه استفاده کنید فراهم می‌آورد. این زمان‌بند یک اجرای پریودیک از توابع و رشته‌های چند اولویتی را نشان می‌دهد.

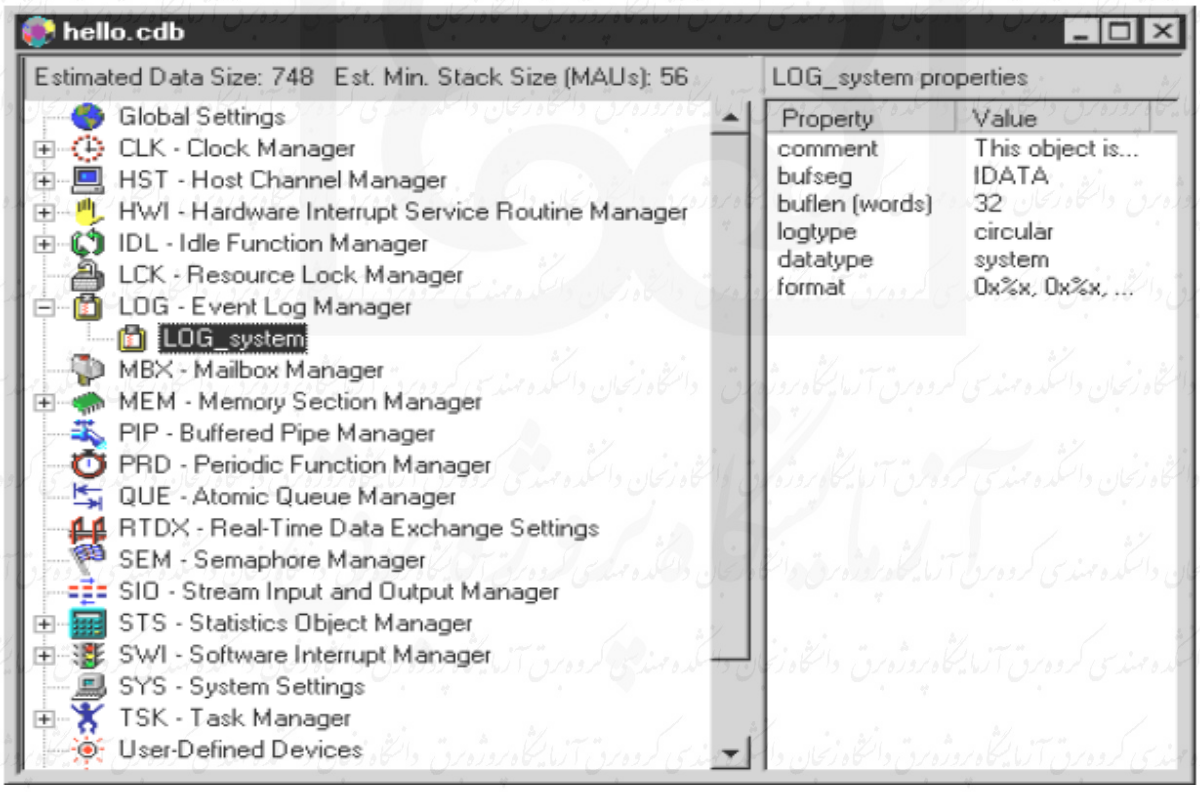
۱-۴-۱ پیکربندی DSP/Bios:

شما می‌توانید فایل‌های پیکربندی در محیط کدنگار که فایل‌های موضوعی مورد استفاده در DSP/Bios API را توصیف می‌کند، ایجاد نمایید. این فایل می‌تواند نگاشت حافظه و نگاشت پیکانی سخت افزار ISR را ساده کند و لذا ممکن است این بخش در استفاده از DSP/Bios API مورد استفاده قرار گیرد.

یک فایل پیکربندی دو نقش را ایفا می‌کند.

- ☐ اجازه‌ی استفاده از پارامترهای جامع در هنگام اجرا را فراهم می‌نماید.
- ☐ این فایل‌ها به عنوان یک تصحیح بصری برای تولید و ایجاد ویژگی‌هایی برای فایل‌های موضوعی در حال اجرا که در فراخونی‌های برد مرجع DSP/Bios API به کار می‌رود، مورد استفاده قرار می‌گیرد. این فایل‌های موضوعی شامل وقفه‌های نرم‌افزاری، خطلوله های I/O و گزارشات رویدادها می‌باشد.

اگر شما یک فایل پیکربندی در کدنگار ایجاد کنید یک پنجره همچون شکل زیر ایجاد می شود:



۸-۷ اجرای برنامه اصلاح شده

۱) Project → Rebuild All را انتخاب نموده و یا روی کلید (بازسازی کلی) در نوار ابزار کلیک نمایید.

۲) File → Load Program را انتخاب نمایید، برنامه‌ی تازه‌بازسازی‌شده را انتخاب نمایید (hostio.out) و آن‌را باز کنید.

۳) Tools → DSP/Bios → Host Channel را انتخاب نمایید. کنترل کانال میزبان فایل‌های موضوعی HST را لیست می‌نماید و به شما اجازه‌ی اتصال آنها به فایل‌های روی کامپیوتر میزبان و شروع و پایان کانال‌های آن داده می‌شود.

Host Channel Control

Channel	Transferr...	Limit	State	Mode	Binding
input_HST	0 B	0 KB	Unbound	Input	<unbound>
output_HST	0 B	0 KB	Unbound	Output	<unbound>

۴) Debug → Run را انتخاب نمایید و یا دکمه‌ی (اجرا) را فشار دهید.

۵) روی کانال input_HST راست کلیک نمایید و گزینه ی Bind را انتخاب کنید.

۶) فایل input.dat را در پوشه‌ی کاری انتخاب کنید و گزینه ی Bind را انتخاب کنید.

۷) روی کانال output_HST راست کلیک نمایید و Bind را انتخاب کنید.

۸) Output.dat را در نام فایل تایپ نمایید و Bind را کلیک نمایید.

Host Channel Control

Channel	Transferr...	Limit	State	Mode	Binding
input_HST	0 B	0 KB	Stopped	Input	C:\ti\myprojects\hostio\input.dat
output_HST	0 B	0 KB	Stopped	Output	C:\ti\myprojects\hostio\output.dat

۹) روی کانال input_HST راست کلیک نمایید و Start را انتخاب کنید.

۱۰) روی کانال output_HST راست کلیک کرده و Start را انتخاب کنید. توجه کنید که ستون Transfered انتقال داده را نشان می‌دهد.

۱۱) وقتی که داده انتقال یافت دکمه‌ی (توقف) در نوار ابزار را کلیک نمایید و یا دکمه‌ی shift F5 را فشار دهید.