

خطا و خطایابی در طراحی سخت افزار

به کوشش امیرحسین آدینه

راه های ارتباطی



برای ارتباط با من راه های زیر هست!

لینکدین: [Amir.H Adineh](https://www.linkedin.com/in/Amir.H.Adineh)

اینستاگرام: [En_adineh](https://www.instagram.com/En_adineh)

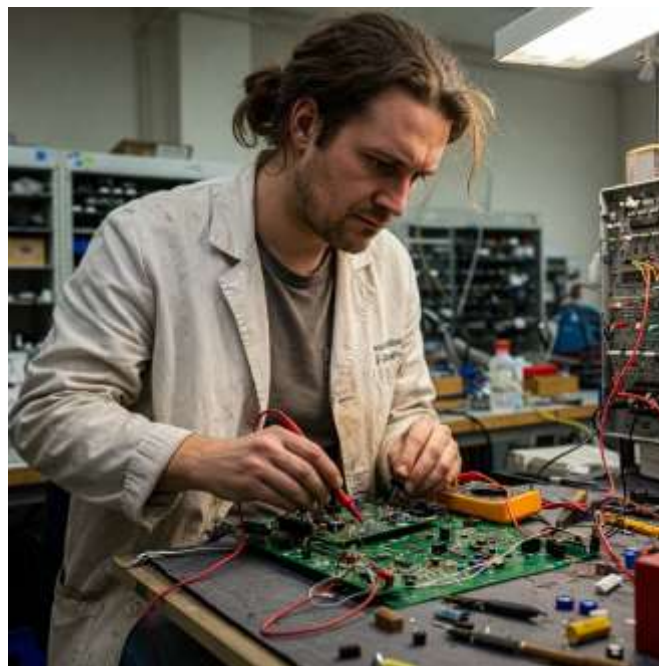
تلگرام: [En_adineh](https://www.telegram.me/En_adineh)

وبسایت گروه ما: www.mirmadar.ir

جهت آموزش: www.PCBNETS.ir

ارسال نامه : اصفهان صندوق پستی ۸۱۷۸۵-۳۵۶

همه چیز از آنجا شروع شد....



مشکلی جدی هست...



مشکل چیست؟!

مشکل خطا یا خطاهایی هستند که ممکن است در هرکدام از این مراحل رخ داده باشد:

❖ طراحی سخت افزار

❖ طراحی نقشه های شماتیک

❖ پیاده سازی PCB

❖ خرید قطعات

❖ تولید و مونتاژ PCB

❖ فرایند راه اندازی و تست محصول

مفهوم باگ در سخت افزار

این اصطلاح اولین بار توسط خانم گریس موری هاپر به کار گرفته شده است.

خانم گریس هاپر به عنوان یکی از اولین برنامه نویسان زن شناخته می شود.

زمانی که هاپر در دانشگاه هاروارد در حال گره گشایی یا دیباگ کردن (Debugging) کامپیوتر Mark II بود، از این لغت به معنای خطا و اشکال استفاده کرد.

زیرا زمانی که هاپر به همراه شماری از متخصصین در حال بررسی کامپیوتر Mark II بودند، متوجه شدند که اختلال در عملکرد این کامپیوتر به دلیل وجود یک حشره در آن ایجاد شده است.

بنابراین در آنجا اصطلاح Bug به معنای نقص یا گره نرم افزاری را بکار بردند.

مفهوم باگ در سخت افزار

اولین تصویر "اشکال کامپیوتری" در حین آزمایش در دانشگاه هاروارد.

9 سپتامبر 1947، بین نقاط رله شماره 70، پانل F، محاسبات رله ایکن Mark II پیدا شد.

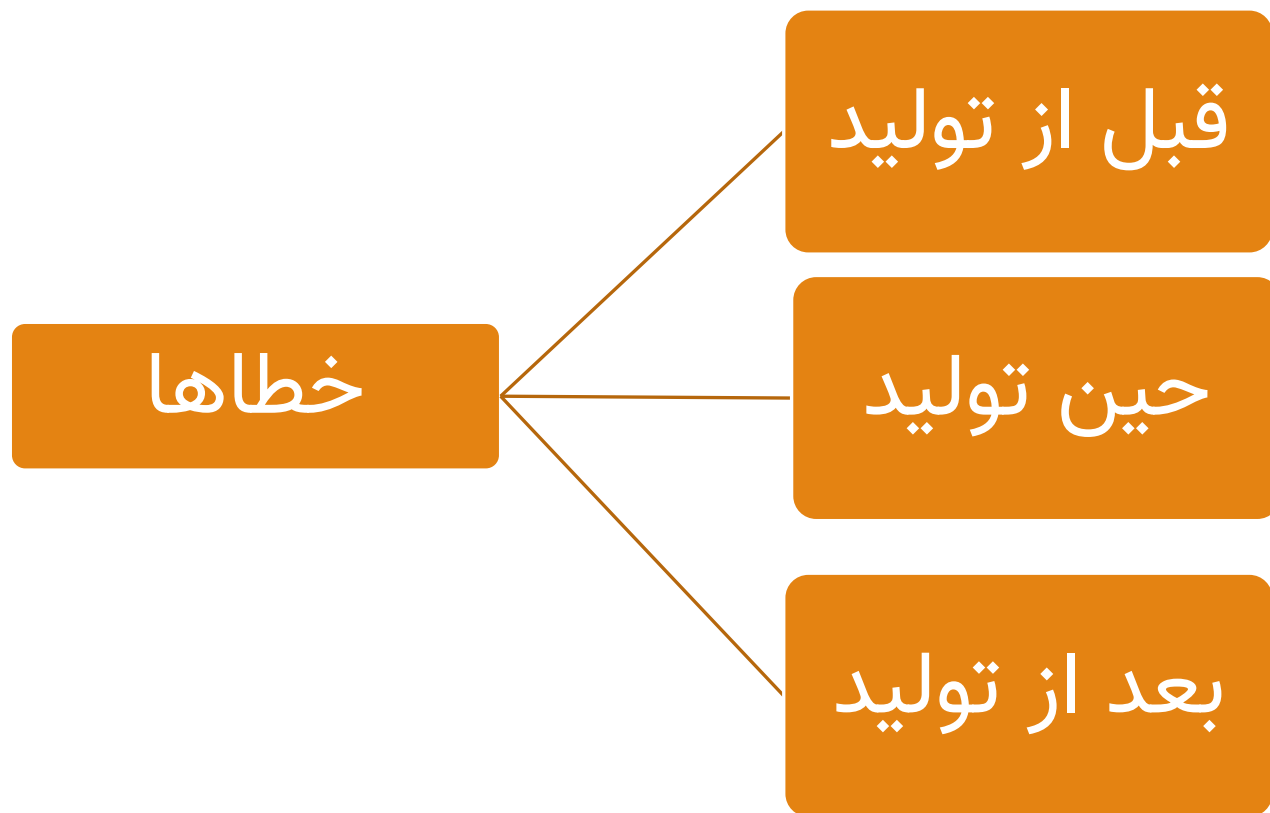
اپراتورها یک لاگ با این مضمون به کامپیوتر چسبانند:

"اولین مورد واقعی از باگ پیدا شد."

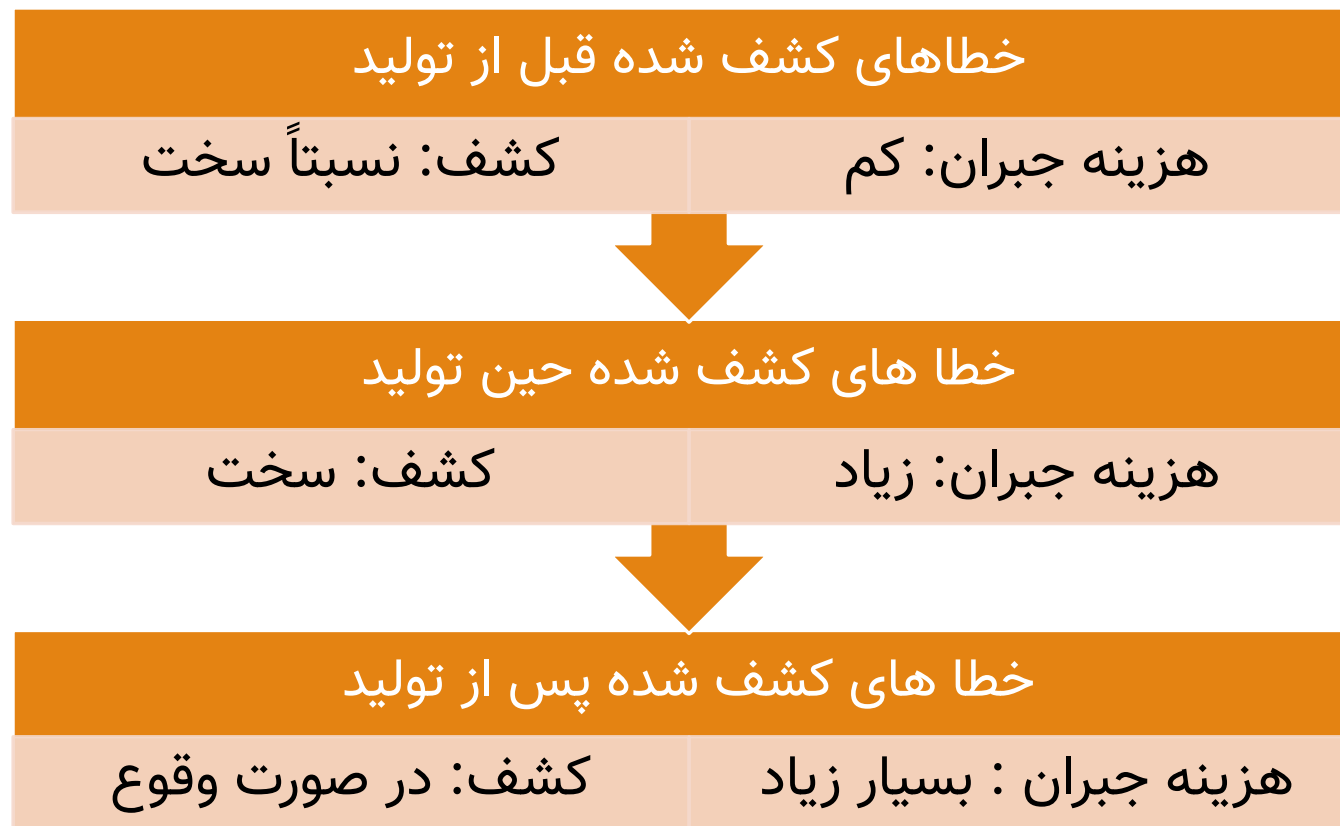
(اصطلاح «اشکال زدایی یا دیباگ» قبلاً وجود داشت؛ بنابراین، یافتن یک باگ واقعی یک اتفاق سرگرم کننده بود.)



طبقه بندی خطا در مراحل تولید سیستم ها



مقایسه تبعات خطا در مراحل مختلف تولید محصول



چه کنیم؟

تا زمانی که یک طرح تولید نشده است، در صورت کشف خطا خسارتی جز صرف زمان مطرح نیست!

اما در صورت تولید، اگر در حد نمونه آزمایشگاهی باشد خسارت هایی را متحمل خواهیم شد، اما در محصول نهایی خسارات ناشی از بروز خطا بسیار زیاد خواهد بود.

دیکتاتور باشید و توطئه های شوم را در نطفه نابود کنید!

برخی از اشتباهات رایج در انتخاب سخت افزار

❖ تخصیص تراشه ها با کاربری نامناسب

❖ در نظر نگرفتن جوانب عملکرد (کم مصرف ، کارآیی بالا ، ارزان و....)

❖ عدم تناسب سخت افزار و پردازش نرم افزاری (مثال: باگ سامانه پدافندی پاتریوت)

❖ اصرار به راه حل های ارزان و ساده (مانند آردوینو) در طرح های پیچیده

یک تجربه

❖ یک پرونده شکایت به من ارجاع شد. مالک یک جرثقیل بتن ریزی به علت نقص در عملکرد ریموت دچار حادثه شده بود و جرثقیل او از کنترل خارج شده بود و

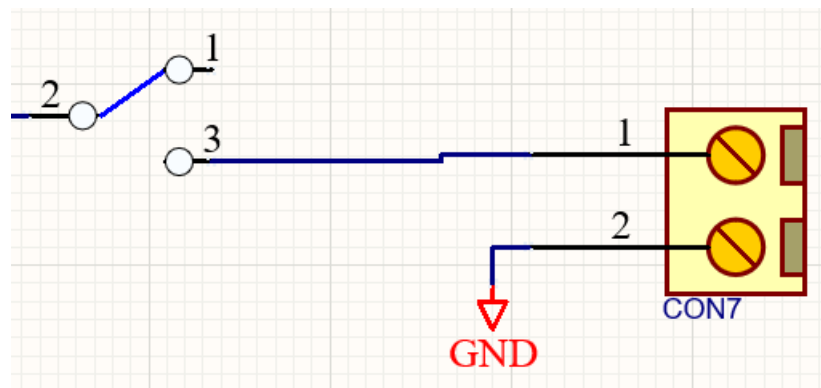
پس از بررسی متوجه شدم که در ریموت بجای پردازنده های دارای گواهی ایمنی از AVR و رله های معمولی برای کنترل عملگر های هیدرولیکی استفاده شده بود و جهت قطع ارتباط نیز تمهیدی اندیشیده نشده بود. متأسفانه طراح این ریموت به پرداخت غرامت سنگینی محکوم شد.

برخی از خطاهای رایج در حین طراحی شماتیک

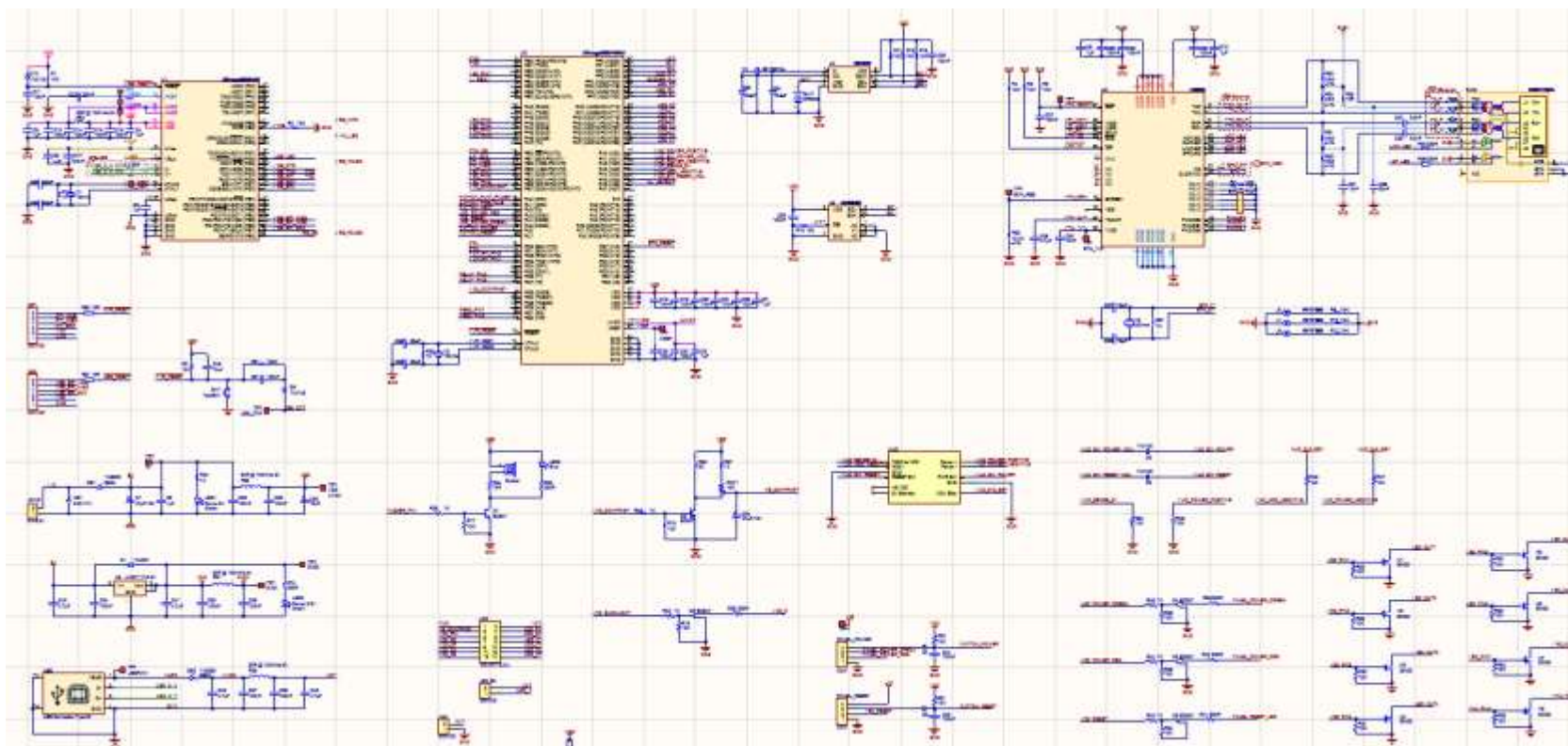
- ❖ کم توجهی به پایه های مهم مانند Reset، فعال ساز، فیدبک و....
- ❖ ترتیب اشتباه در اتصال پایه ها (قربانی همیشگی: پورتهای سریال که نیاز به اتصال Cross دارند).
- ❖ اشتباه در مقادیر قطعات یا اتصالات مانند آدرس های باس ها
- ❖ مقادیر اشتباه در عناصر مدار (اغلب ناشی از کپی کردن!)
- ❖ دقت نکردن در انتخاب قطعه و پیشوند و پسوند های آن
- ❖ ترسیم عجولانه و نامفهوم نقشه و عدم درج اطلاعات کافی

برخی از خطاهای رایج در حین طراحی شماتیک

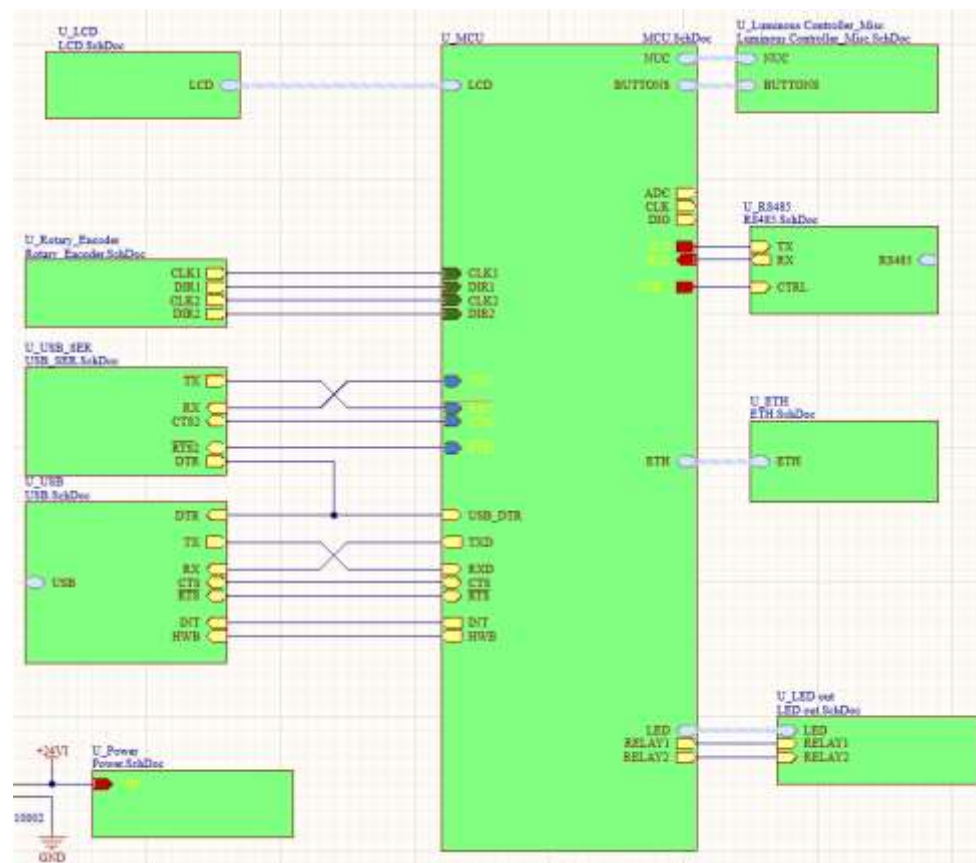
❖ رعایت نکردن Grid در ترسیم شماتیک



یک شماتیک که نامفهوم و گنگ ترسیم شده



همان شماتیک پس از بهینه سازی و مرتب کردن



خطاهای رایج در طراحی کتابخانه

- ❖ شماره گذاری اشتباه پایه ها (به خصوص در مدل هایی که دستی طراحی می شوند).
- ❖ فواصل و اندازه های اشتباه در فوتپرینت؛ مثلاً بین هدر mm1.25 بجای mm1.27
- ❖ دقت نکردن به Land pattern و Pinout ارایه شده در دیتا شیت.
- ❖ طراحی وارونه قطعات (رله ها - کانکتور ها و قطعات THT)
- ❖ پدهای نامتناسب

منشاء برخی خطاهای رایج در طراحی PCB

- ❖ منتقل نشدن قطعات و نت لیست به طور صحیح
- ❖ پین ها و نت هایی که به هر علت طور که دستی اصلاح می شوند
- ❖ قرار دادن قطعه ای در حریم قطعه ای دیگر (یا زیر آن)
- ❖ استفاده از فوتپرینت نادرست یا آینه ای کردن فوتپرینت (با دستور flip)
- ❖ نامرتب قرار دادن متن ها و چاپ راهنما (در عملکرد تاثیر ندارد)
- ❖ رعایت نکردن گرید در شماتیک و اتصالات ناقص در PCB
- ❖ رعایت نکردن استاندارد ها یا محدودیت های شرکت سازنده برد!

یک مثال...

Ordering Information

RT9166/A-□□□□

Package Type

VL : SOT-23-3 (L-Type) (RT9166 Only)

X : SOT-89

XL : SOT-89 (L-Type)

G : SOT-223

GL : SOT-223 (L-Type)

Z : TO-92

L : TO-252

Lead Plating System

P : Pb Free

G : Green (Halogen Free and Pb Free)

Output Voltage

12 : 1.2V

13 : 1.3V

:

45 : 4.5V

1B : 1.25V

600mA Output Current

300mA Output Current

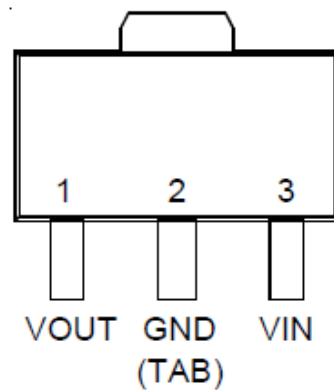
در یک طرح تراشه RT9166/A33X مد نظر بود.

از تراشه RT9166/A33XL استفاده شده بود.

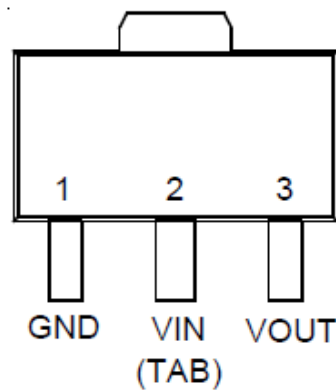
در نگاه اول گفته می شود که گزینه دوم احتمالاً بهتر است!

پکیج ها هم که مانند هم هستند....

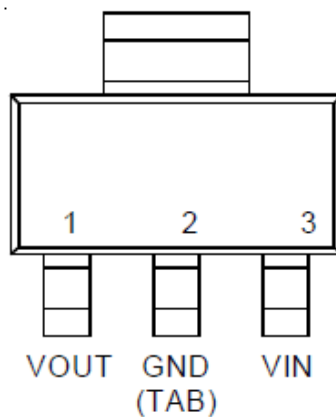
RT9166/A



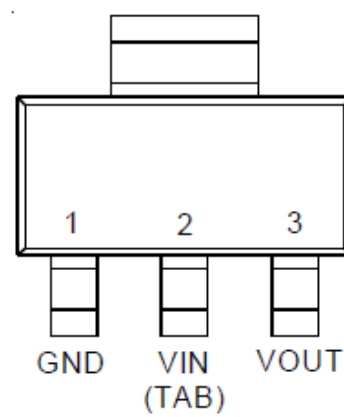
SOT-89



SOT-89 (L-Type)



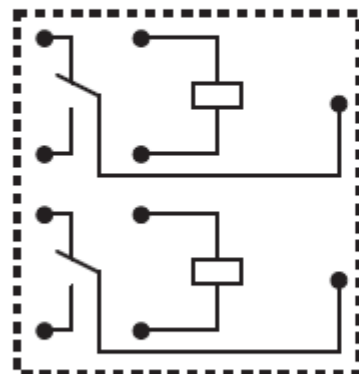
SOT-223



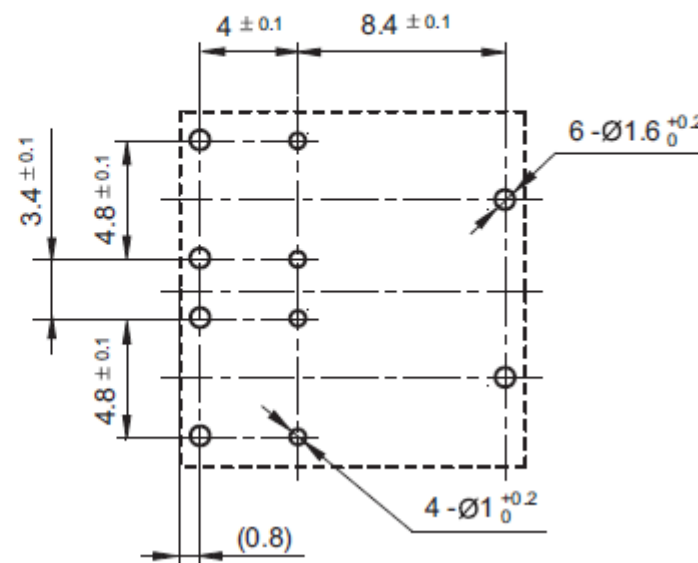
SOT-223 (L-Type)

در برخی دیتاشیت ها نمایش وارونه است!

Wiring Diagram (Bottom view)



PCB Layout (Bottom view)

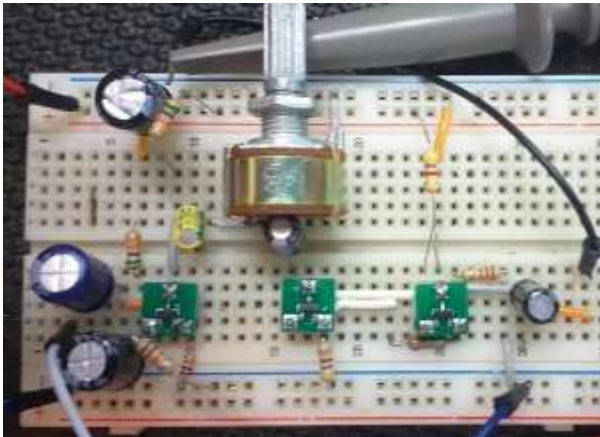


چطور از ایجاد باگ حین طراحی اجتناب کنیم؟

❖ اگر مداری را از جایی دیگر یا پروژه های قبلی کپی کرده اید، مجدداً مقادیر و اتصالات را بررسی کنید، خطاهای زیادی در لابلای این کپی کردن ها لانه میکنند!

❖ اگر برای اولین بار مدل های یک قطعه را طراحی میکنید، بهتر است که یک نمونه فیزیکی از قطعه را در اختیار داشته باشید (به خصوص عزیزان دورکار و فریلنسر)

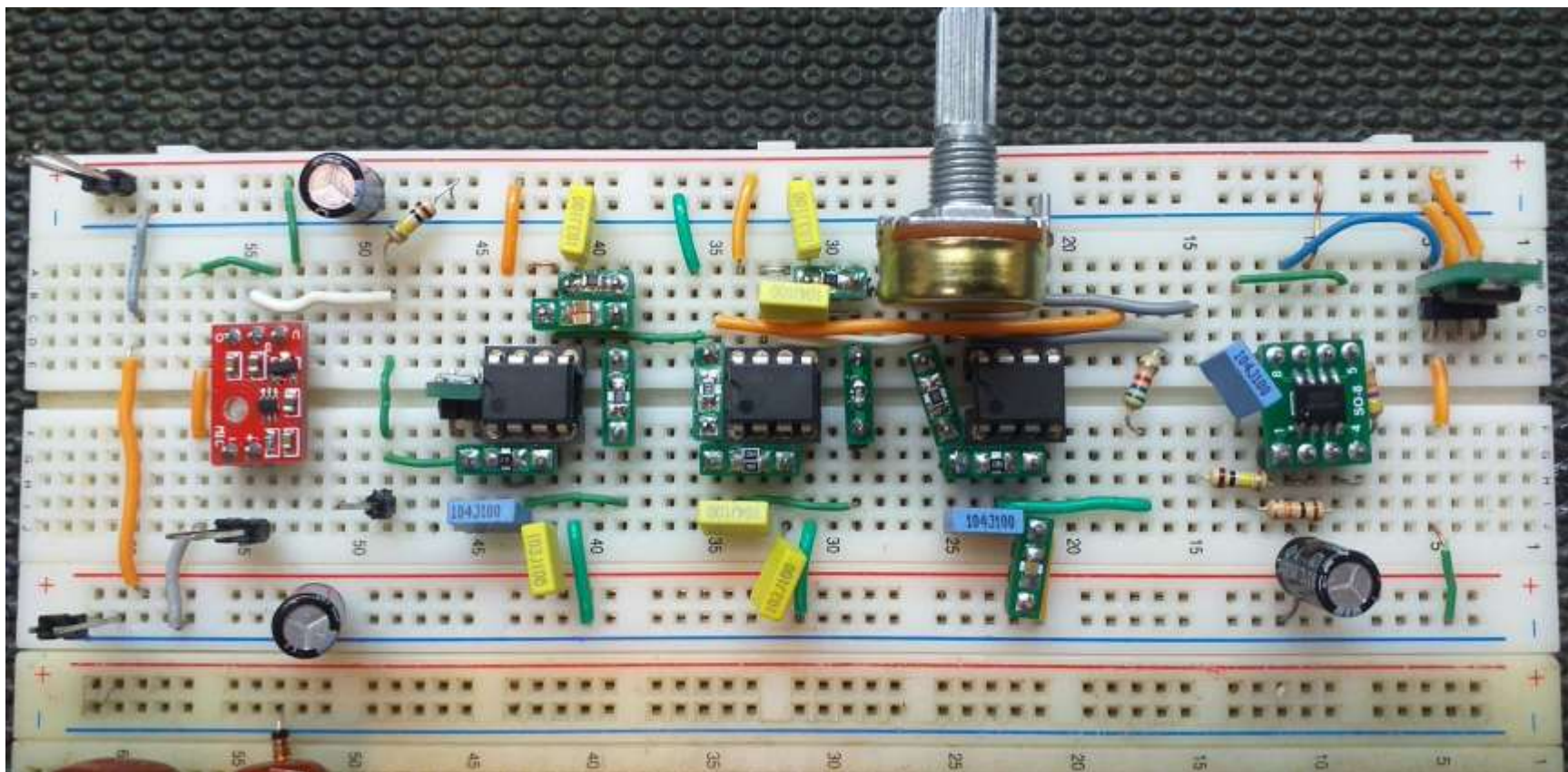
❖ اگر با یک IC تابحال کار نکرده اید، شایسته است یک برد تجربی Experimental یا Breakout طراحی کنید و نمونه سازی کنید.



طرح نهایی (پس از آخرین تست)



بررسی یک مدار روی برد



چطور از ایجاد باگ حین طراحی اجتناب کنیم؟



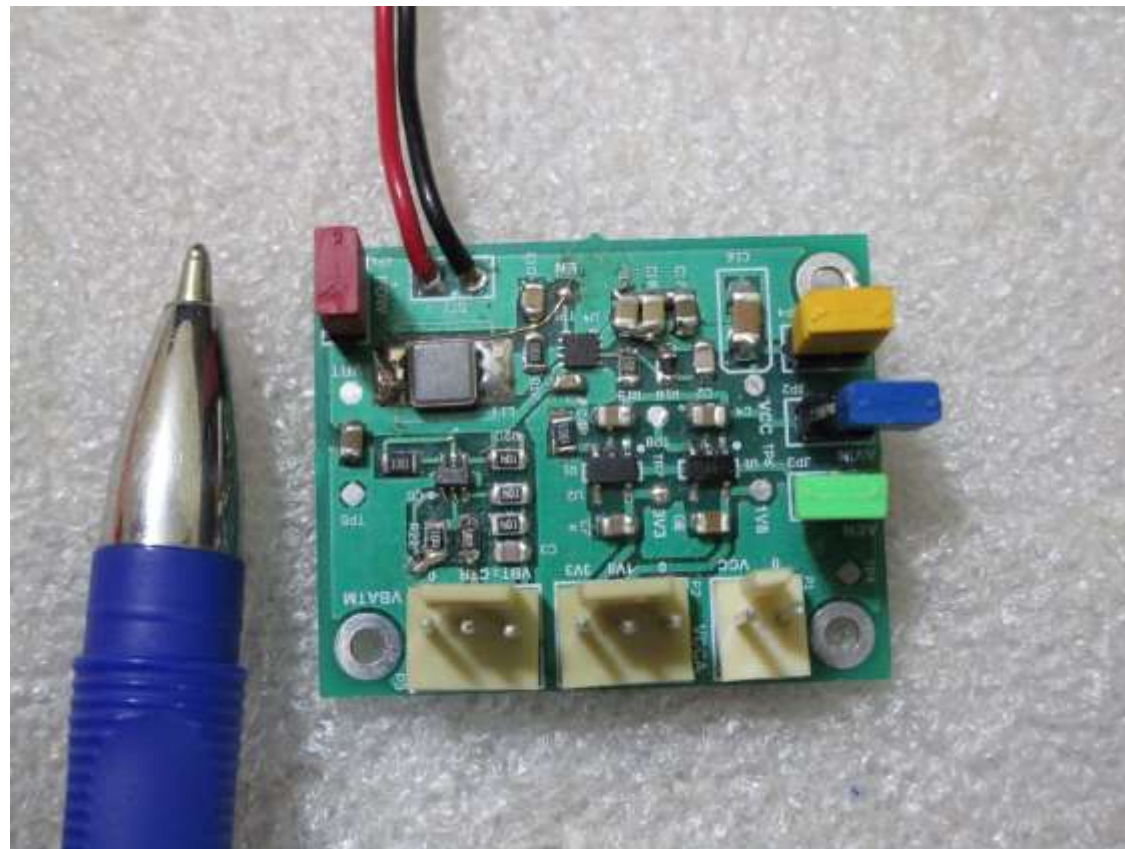
❖ قبل از شروع طراحی حتماً ضوابط و ویژگی های طرح، قابلیت های فنی و استانداردهای مورد نیاز را بررسی کنید و در این خصوص حتماً اطلاعات کافی را از کارفرما یا مدیر خود دریافت کنید!

❖ از نرم افزارهای DRC و تحلیلگر (مانند CST , Hyperlynx و) استفاده کنید.

❖ در حین طراحی یا پس از طراحی طرح خود را توسط شخصی غیر از طراح مورد بررسی و تحلیل قرار دهید.

❖ نمونه آزمایشگاهی بسازید و تست کنید

طرح تجرّبی اولیه (چند باگ کشف شده)



اگر پس از ساخت به مشکل برخورد کردیم ...



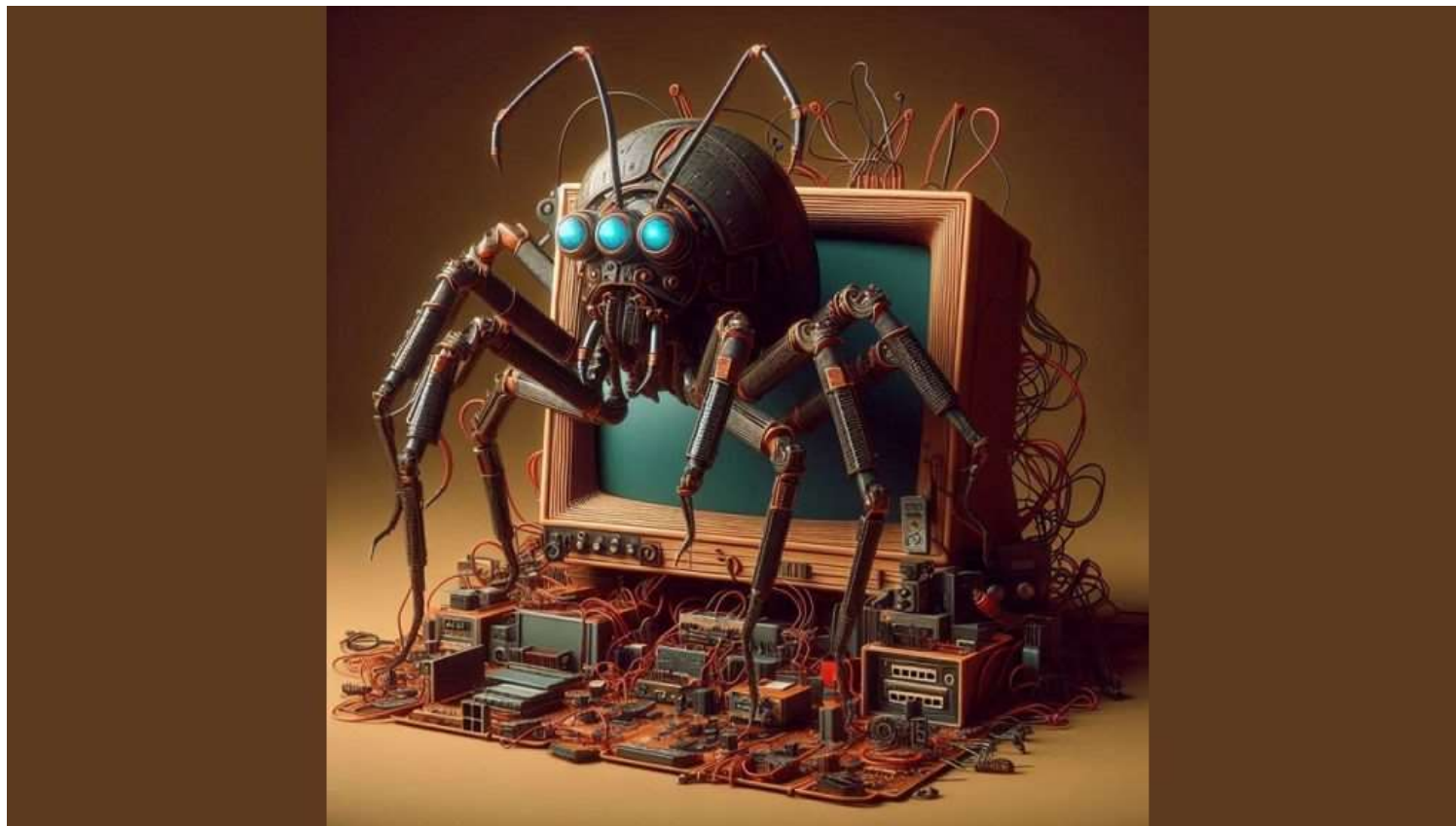
❖ خونسرد باشد و سعی کنید ابتدا طرح را تحلیل کنید.
سپس مقادیر و شماره قطعات و اتصالات را با طرح تطبیق
دهید. پیشوند و پسوند قطعات را حتما بررسی کنید و
انطباق دهید

❖ در پردازنده ها مطمئن شوید که TX و RX جابجا نیست!

❖ پایه های حساس مانند ریست و فیدبک و فعال ساز را با
اسیلوسکوپ بررسی کنید.

❖ مراقب تله های ذهنی باشید! سعی کنید از اشخاص با
تجربه در محل کار یا [فروم](#) ها کمک بگیرید.

برخی از باگ های معروف تاریخ



سال ۱۹۹۱ : ناتوانی سامانه موشکی پاتریوت در مقابل سامانه SCUD

در طول جنگ خلیج در دهه ۹۰ میلادی، ارتش عراق از سامانه پرتاب موشک SCUD که توسط روسیه ساخته شده بود، استفاده می‌کرد و آمریکا هم از سامانه ضد موشک پاتریوت بهره می‌برد.

شب ۲۵ فوریه سال ۱۹۹۱، سامانه موشکی پاتریوت در ظهران عربستان از رهگیری موشک عراقی عاجز می‌ماند و موشک به سنگر نیروهای آمریکایی برخورد کرده و ۲۸ سرباز آمریکایی کشته و ۹۸ سرباز مجروح می‌شوند.

سال ۱۹۹۱ : ناتوانی سامانه موشکی پاتریوت در مقابل سامانه SCUD

خطا چه بود؟

محاسبه زمان رسیدن موشک به هدف، بر اساس سرعت موشک و زمان رویت توسط آخرین رادار صورت می‌گیرد. مقدار سرعت به صورت یک عدد طبیعی و یک عدد اعشاری و مقدار زمان به صورت یک عدد طبیعی یا یک عدد صحیح پیوسته در سیستم قرار می‌گیرد.

الگوریتم محاسبه زمان رسیدن موشک به هدف، نیاز به بررسی مقدار زمان و سرعت به طور همزمان دارد. در حالی که رجیسترهای کامپیوترهای پاتریوت ۲۴ بیتی هستند. بنابراین و با توجه به این که زمان بر حسب دهم ثانیه مطرح میشد، هنگامی که دقت محاسبه زمان خیلی بالا می‌رفت، سیستم‌های پاتریوت دچار مشکل می‌شد و راه‌حلی که برای این مشکل در نظر گرفته شده بود، ریستارت شدن و راه‌اندازی مجدد سیستم بود که SCUD از همین راهکار استفاده کرد.

سال ۱۹۹۸: سقوط Mars Climate Orbiter در مریخ

در سال ۹۸ میلادی، دو کاوشگر ناسا Mars Climate Orbiter و Mars Polar Lander پرتاب شدند. قرار بود Mars Climate Orbiter در مدار مریخ بچرخد و Mars Polar Lander در یکی از قطب‌های مریخ بنشیند. مشکل زمانی بوجود آمد که Mars Climate Orbiter وقتی می‌خواست در مدار خود قرار بگیرد، در اتمسفر مریخ بسیار پایین آمد و سقوط کرد.

یکی از پیمانکاران ناسا از سیستم اندازه‌گیری امپریال استفاده می‌کرد ولی ناسا از سیستم اندازه‌گیری متریک بهره می‌برد و همین باعث می‌شود کاوشگر در تشخیص فشار هوا اشتباه کند. کارها به صورت جداگانه درست بود اما در کنار هم خیر.

این خطا 330 میلیون دلار هزینه داشت.... نرخ مبادله دلار امروز چقدر است؟!

نقص در نرم افزار دستگاه پرتودرمانی Therac-25



نقص در نرم افزار دستگاه پرتودرمانی Therac-25

دستگاه پرتودرمانی Therac-25 توسط آژانس انرژی اتمی کانادا و با مشارکت یک شرکت فرانسوی بین سالهای 1985 تا 1987 توسعه داده شد و برای بیماران مبتلا به سرطان استفاده می‌شد.

این دستگاه با دو حالت عمل می‌کرد:

- حالت اول، که باید از یک پرتوی الکترونی با قدرت کم (ذرات بتا) را استفاده می‌شد
 - حالت دوم که برای درمان بیماری‌های شدیدتر بود و در آن اشعه X با قدرت بسیار بالا کاربرد داشت.
- در طی دو سال، تعدادی از افرادی که با این دستگاه تحت درمان قرار گرفته بودند، دچار مسمومیت پرتویی شدند و تعدادی هم جان خود را از دست دادند.

نقص در نرم افزار دستگاه پرتودرمانی Therac-25

علت نقص؟

نرم افزار تامین انرژی این دستگاه از مدل قبلی این دستگاه که تک حالت بود، اقتباس شده بود و نتیجه ای که بر اساس تغییر سریع بین دو حالت به وجود می آمد مشخص نبود.

وقتی این اتفاق می افتاد، نرم افزار دستورالعمل هر دو حالت را ارسال می کرد و هر کدام که زودتر می رسید اجرا می شد. (نوعی خطا در سیستم عامل بلادرنگ)

این مشکل باعث برخورد اشعه X به بیمارانی شد که به طور کامل محافظت نشده بودند. حاصل این اتفاق مسمومیت پرتویی شش نفر بود که پنج نفرشان جان خود را از دست دادند.

در کپی کردن هر نوع کد ، نقشه و داده دقت کنید و با جان مردم بازی نکنید!

سال 2000: باگ Millennium مشکل هزاره

در گذشته طراحان نرمافزار عادت داشتند تا از یک عدد دو رقمی برای نمایش تاریخ در نرمافزارهای خود استفاده کنند، مثلاً برای نمایش ۱۹۹۸ از ۹۸ استفاده می‌کردند. به همین علت کامپیوترها در آغاز سال ۲۰۰۰ نمی‌دانستند که ۰۰ ، ۱۹۰۰ است یا ۲۰۰۰.

بر اساس گزارش‌ها هزینه برطرف کردن این مشکل و اصلاح آن (چهار رقمی کردن عدد نمایش‌دهنده تاریخ) بالغ بر ۴۰۰ میلیارد یورو بوده است. مشابه این را در سال 1400 داشتیم!

سپاس از همراهی شما



فروم سیسوگ رو فراموش نکنید!
موکدا در کپی کردن طرح هایتان خیلی دقت کنید...

جهت ارتباط با من کد رو به رو را اسکن کنید!