



سلام بر شما





نام پروژه : مولد ۲۰ هارمونیک

دانشجو : هوشنگ ناصری

شماره دانشجویی : ۱۰۷۰۴

رشته : الکترونیک

استاد راهنما: جناب آقای مهندس انصاری و شعبانی

سال تحصیلی ۸۹ - ۸۸

با تشکر و قدر دانی فراوان از اساتید محترم جناب آقای:

مهندس انصاری

مهندس شعبانی

مهندس زارعی

برای دانلود فایل های پروژه و دیگر آموزش ها به وبسایت ما سر بزنید:

MicroDesigner.ir



فهرست مطالب

صفحه	عنوان
۸	مقدمه:.....
فصل اول: (در مورد میکرو کنترلرها)	
۹	میکروکنترلر چیست:.....
۱۱	ساخت داخلی میکروکنترلر:.....
۱۲	تفاوت میکرو پرسور و میکروکنترلر:.....
۱۳	عیب میکرو کنترلرها.....



۱۳ سیر تکاملی میکرو کنترلرها

۱۵ معماری داخلی میکرو کنترلر

۱۶ معرفی انواع میکرو کنترلرها

فصل دوم: (در مورد سخت افزار پروژه و کار کلی پروژه)

۲۰ مفهوم نام پروژه

۲۰ وظیفه پروژه

۲۰ سخت افزار پروژه

۲۰ Keyboard

۲۱ LCD

۲۲ DAC

۲۳ مبدل آنالوک به دیجیتال میکرو

۲۴ تکنیک کاهش نویز در DAC



شکا ظاهری DAC0808..... ۲۵.....

MICRO..... ۲۶.....

شکل کلی سخت افزار..... ۲۸.....

توضیح کلی انجام کار..... ۲۹.....

علت گرفتن بزرگترین T..... ۳۰.....

نحوه گرفتن T مادر..... ۳۱.....

فرکانس خروجی..... ۳۳.....

فصل سوم: (در مورد نرم افزار و برنامه نویسی پروژه)

برنامه شناخت Keyboard..... ۳۴.....

برنامه وارد کردن دامنه..... ۳۵.....

برنامه وارد کردن فرکانس..... ۳۷.....

برنامه ک م م گرفتن..... ۳۹.....

نحوه استفاده از بست و تایمر برای ایجاد فرکانس خروجی مناسب..... ۴۰.....

قسمت وقفه برنامه..... ۴۲.....

کل برنامه..... ۴۳.....



مثالهایی از این پروژه با دامنه وفر کانس مختلف که باکی بورد به آن دادیم ۵۱

SHAMATUC این پروژه..... ۵۴

PCB این پروژه..... ۵۵

فصل چهارم (Data sheet)

Data sheet dac 0808..... ۵۶

Data sheet LF 351..... ۶۶

Data sheet lcd..... ۷۶

فصل پنجم:

نتیجه گیری..... ۸۰

مراجع..... ۸۲



مقدمه:

نام پروژه: مولد ۲۰ هارمونیک

مفهوم این اسم:

مولد یعنی تولید کننده ۲۰ هم که عدد است و هارمونیک هم به معنی مشتق های یک موج اصلی یا شبیح همان موج اصلی که مقدار دامنه یا فرکانس آن تغییر کرده.

وظیفه این پروژه:

این پروژه یا دستگاه باید ۲۰ موج سینوسی را توسط keyboard بگیرد و با هم جمع لحظه ای کند و در خروجی به یک موج تبدیل کند که ما توسط keyboard فقط مقدار دامنه و فرکانس را وارد می کنیم .



قبل از پرداختن به پروژه با یکسری تعاریف و مقدمات شروع کنیم :

اجرای توابع در میکرو نیازمند برنامه ریزی می باشد. پس نیاز به زبانی قابل فهم برای آی سی ها داریم. اگر بخواهیم برنامه را به صورت کد بنویسیم وقت زیادی می برد و ممکن است دارای خطاهای زیادی باشد پس برای این کار بهتر است از نرم افزار مناسب استفاده کنیم که به آنها کامپایلر یا مبدل گفته می شود امروزه کامپایلرهای مختلفی وجود دارد مانند : زبان اسمبلی ، زبان C ، basic و

مقایسه ی زبان C و زبان اسمبلی:

زبان اسمبلی :

محاسن: سادگی ، سرعت زیاد ، حجم کد تولید شده ی بسیار پایین

معایب: در برنامه های بزرگتر از یک کیلو بایت کار با آن بسیار مشکل است

زبان C:

محاسن: مدیریت برنامه های ۱۰۰۰ خطی نیز در آن صورت می پذیرد

معایب: سرعت نسبت به اسمبلی کمتر ، حجم کد تولید شده بالاتر

که این پروژه بیش از ۹۰٪ به صورت برنامه نویسی است. در نتیجه باید توضیحاتی در مورد میکرو داده شود



فصل اول:

میکروکنترلر چیست :

قطعه ای که این روزها دارد جای خود را در خیلی از وسایل الکتریکی باز میکند. از تلفن گرفته تا موبایل از ماوس لیزری که الان دستتان روی آن است و دارین باهاش کامپیوتر رو کنترل میکنید تا هر وسیله ای که بتوان پیچیدگی رو در اون دید میتونید یک میکروکنترلر رو ببینید .

کلمه میکروکنترلر :

این کلمه از دو کلمه ۱- میکرو ۲-کنترلر تشکیل شده

1- میکرو : میدونین که این یک واحد یونانی است و برابر با 10 به توان منفی ۶ متر است. یعنی یک میلیونوم متر

واحد خیلی کوچیکیه نه...ولی واحدهای خیلی کوچیکتر از این هم داریم که در الکترونیک مورد استفاده قرار میگیرند در قسمتهای بعدی توضیحیهاتی راجع به این واحد ها و موارد استفاده آنها داده میشه .

۲- کنترلر : که همه معنی و مفهومشو میدونین . یعنی کنترل کننده به تعبیری یعنی "مغز" البته بدون تفکر فقط



حالا چرا این کلمات ؟

کلمه میکرو به دو منظور استفاده شده منظور اول و مهم سرعت عمل میکروکنترلر است که میتواند تا یک میلیونیم ثانیه باشد و دستوراتی که به اون میدیم با این سرعت انجام بده به همین خاطر واژه میکرو رو به اون اختصاص دادن البته معنی دوم آن شاید کوچیکی این قطعه باشد که تا یک میلیون متر کوچیک شده شاید باور کردنی نباشه ولی در یک تراشه ممکنه بیش از یک میلیون ترانزیستور به کار رفته باشه. این کلمه وقتی اهمیتش کامل میشه که با واژه کنترلر عجین بشه تا معنیش کامل بشود .

حالا نحوه انجام دادن کار میکروکنترلر را به صورت کلی بررسی میکنیم :

تا حالا همه شما با ماشین حساب کار کردین تا حالا به نحوه کار کردنش فکر کردین شما اطلاعاتتون را که همون عملیات ریاضی هست به وسیله صفحه کلید به اون میدید بعد ماشین حساب این اطلاعات رو بر مبنای دستوراتی که قبلا به اون داده شده پردازش میکنه و جواب را روی lcd نمایش میده. در واقع یک میکروکنترلر برنامه ریزی شده به عنوان مغز ماشین حساب این اطلاعات یا داده رو از صفحه کلید میگیره روشن پردازش انجام میده و بعد بر روی lcd نمایش میده .

کار میکروکنترلر دقیقا مشابه این است میکرو کنترلر بر مبنای یک سری ورودی که به اون داده میشه مثلا این ورودی از یک سنسور دما باشه که درجه حرارت رو میگیره یا از هر چیز دیگه مثل صفحه کلید بر مبنای این ورودی ها و برنامه ای



که قبلا ما به اون دادیم خروجیشو تنظیم میکنه که ممکنه خروجیش یک موتور باشه یا یک lcd یا هر چیز دیگری که با الکتریسیته کار بکند. حالت دیگری هم میتونه باشه که فقط میکروکنترلر بر مبنای برنامه ای که به اون دادیم عمل کند و خروجیش رو فقط بر اساس برنامه بگیرد .

ساختمان دخی میکروکنترلر :

کامپیوتری که الان بر روی اون دارین کار انجام میدین دارای یک پردازنده مرکزیه به نام cpu که از کنار هم قرار گرفتن چندین ملیون ترانزیستور تشکیل شده و بر روی اطلاعات پردازش انجام میده . میکرو کنترلر هم عینا دارای یک پردازنده مرکزی به نام cpu است که دقیقا کار cpu کامپیوتر رو انجام میده با این تفاوت که قدرت و سرعت پردازشش از cpu کمتره که به اون میکروپروسور میگن در بخش بعدی فرق میکرو پروسور و میکروکنترلر را بررسی میکنیم. میکروکنترلر علاوه بر cpu دارای حافظه است که ما برنامه ای که بهش میدیم در اون قرار بگیره در کنار حافظه در میکروکنترلرهای امروزی تایمرها برای تنظیم زمان کانترها برای شمردن کانال های آنالوگ به دیجیتال پورت های برای گرفتن و دادن اطلاعات و امکاناتی دیگر که بعدا مفصل راجع به هر کدام توضیح داده میشه تشکیل شده و همه اینها در یک چیپ قرار گرفته که تکنولوژی جدید اونو تو یک تراشه به اندازه یک سکه قرار داده .



تفاوت میکروپروسسور و میکروکنترلر :

میکروپروسسور همانطور که گفته شد یک پردازنده است و برای کار باید به آن چیپ های حافظه و چیز های دیگری را به اون اضافه کرد این امکان به درد این میخورد که بر حسب کارمان حافظه مناسب و دیگر قطعات را مانند تایمرها و غیره به صورت بیشتری استفاده کنیم ولی مدار خیلی پیچیده میشود و از لحاظ هزینه هم هزینه بیشتر میشود به همین دلیل امروزه از میکروپروسسورها کمتر استفاده میشود اما این روزها میکرو کنترلر های جدید با حافظه های زیاد تعداد تایمر زیاد پورت های زیاد و تنوع بسیار زیاد آنها بر حسب این امکانات دست ما را باز گذاشته است تا دیگر میکروپروسسورها را فراموش کنیم .

آیا میکروکنترلر چیز جدیدی را با خود آورده است ؟

جواب منفی است تمام کارهایی که ما با میکروکنترلر میتوانیم انجام بدهیم با قطعات دیگر هم میتوانیم انجام بدهیم چون ما قبلا هم تایمر داشتیم هم کانتر هم حافظه هم پردازنده و... در واقع میکروکنترلر قطعه ای است با تمام این

امکانات که به صورت یک آی سی آماده شده است و هزینه پیچیدگی و حجم را به نحوه قابل ملاحظه ای کاهش



عیب میکروکنترلر :

میکروکنترلر با این همه مزایا که گفتیم دارای یک عیب کوچیک است. و آن سرعت پایین ! است آیا سرعتی معادل یک

میلیونیوم ثانیه سرعت کمی است ؟ سرعت کمی نیست ولی یک مثال شاید بحثو بهتر باز کند

یک گیت منطقی رو در نظر بگیرین که با توجه به ورودی خروجیشو تنظیم مکنه سرعت عمل این گیت منطقی ۱۰ به

توان منفی ۹ ثانیه است یعنی نانو ثانیه ولی اگر ما بخواهیم این گیت رو با میکروکنترلر کار کنیم سرعتی معادل میکرو

ثانیه داریم پس از لحاظ سرعت برای کاربردهای خیلی محدودی میکروکنترلر مناسب نیست .

خب حالا این میکروکنترلر را با این همه کاربرد کی ساخته؟

حدود ۴ دهه پیش در سال ۱۹۷۱ میلادی شرکت اینتل اولین میکروکنترلر را ساخت و اولین میکروکنترلر را با نام

۸۰۸۰ در اوایل سال ۱۹۸۰ روانه بازار کرد. همین شرکت اینتلی که الان در ساخت cpu یکه تاز دنیاست. اما بعدا این

امتیاز رو به شرکت های دیگری واگذار کرد و شرکت های زیادی در حال حاضر میکروکنترلر های مختلف تولید میکنند.

سیر تکاملی میکرو کنترلرها :



اولین میکرو کنترلرها در اواسط دهه ۱۹۷۰ ساخته شدند. این میکرو کنترلرها در ابتدا پردازنده های ماشین حساب بودند که دارای حافظه برنامه کوچکی از نوع ROM ، حافظه داده از نوع RAM و تعدادی درگاه ورودی/خروجی بودند. با توسعه فناوری سیلیکون ، میکرو کنترلرهای ۸ بیتی قویتری ساخته شدند.

در این میکرو کنترلرها علاوه بر بهینه شدن دستورالعمل ها، تایمر/شمارنده روی تراشه، امکانات وقفه و کنترل بهینه شده خطوط I/O نیز به آنها اضافه شده است. حافظه موجود بر روی تراشه هنوز هم محدود می باشد و در بسیاری موارد کافی نیست. یکی از پیشرفتهای قابل توجه در آن زمان، قابلیت استفاده از حافظه EPROM قابل پاک شدن با اشعه ماورا بنفش، روی تراشه بود.

این قابلیت، زمان طراحی و پیاده سازی محصول را بطور محسوسی کاهش داد و نیز برای اولین بار امکان استفاده از میکرو کنترلر ها را در کاربردهایی که حجم تولید پایینی دارند، فراهم ساخت. خانواده ۸۰۵۱ در اوایل دهه ۱۹۸۰ توسط شرکت اینتل معرفی گردید .

از آن زمان تاکنون ...

8051 یکی از محبوبترین میکرو کنترلرها بوده و بسیاری از شرکتهای دیگر نیز به تولید آن اقدام کرده اند. در حال حاضر مدل های مختلفی از ۸۰۵۱ وجود دارد که در بسیاری از آنها امکاناتی نظیر مبدل آنالوگ به دیجیتال حجم نسبتاً بزرگ از حافظه برنامه و حافظه داده، مدولاتور عرض پالس (PWM) در خروجیها و حافظه فلش (Flash) که امکان پاک کردن و برنامه ریزی مجدد آن توسط سیگنالهای الکتریکی وجود دارد، تعبیه شده است .



میکرو کنترلرها اکنون به سمت ۱۶ بیتی شدن در حرکت هستند. میکرو کنترلر های ۱۶ بیتی، پردازنده هایی با کارایی بالا (نظیر پردازش سیگنالهای دیجیتال) می باشند که در کنترل فرایندهای بلادرنگ و در مواردی که حجم زیادی از عملیات محاسباتی مورد نیاز است، به کار برده می شوند. بسیاری از میکرو کنترلرهای ۱۶ بیتی، امکاناتی نظیر حجم زیاد حافظه برنامه و حافظه داده، مبدل های آنالوگ به دیجیتال چند کانالی، تعداد زیادی درگاه I/O، چندین درگاه سریال، عملکردهای بسیار سریع ریاضی و منطقی و مجموعه دستورالعمل های بسیار قدرتمند با قابلیت پردازش سیگنال را دارا می باشند.

معماری داخلی میکرو کنترلرها :

ساده ترین معماری میکرو کنترلر، متشکل از یک ریز پردازنده، حافظه و درگاه ورودی/خروجی است. ریز پردازنده نیز متشکل از واحد پردازش مرکز (CPU) و واحد کنترل (CU) است .

CPU در واقع مغز یک ریز پردازنده است و محلی است که در آنجا تمام عملیات ریاضی و منطقی انجام می شود. واحد کنترل، عملیات داخلی ریز پردازنده را کنترل می کند و سیگنال های کنترلی را به سایر بخشهای ریز پردازنده ارسال می کند تا دستورالعمل های مورد نظر انجام شوند .

حافظه بخش خیلی مهم از یک سیستم میکرو کامپیوتری است. ما می توانیم بر اساس به کارگیری حافظه، آن را به دو گروه دسته بندی کنیم: حافظه برنامه و حافظه داده . حافظه برنامه، تمام کد برنامه را ذخیره می کند. این حافظه معمولاً از نوع حافظه فقط خواندنی (ROM) می باشد. انواع دیگری از حافظه ها نظیر EPROM و حافظه های فلش



EEPROM برای کاربردهایی که حجم تولید پایینی دارند و همچنین هنگام پیاده سازی برنامه به کار می روند .

حافظه داده از نوع حافظه خواندن / نوشتن (RAM) می باشد . در کاربردهای پیچیده که به حجم بالایی از حافظه

RAM نیاز داریم ، امکان اضافه کردن تراشه های حافظه بیرونی به اغلب میکرو کنترلر ها وجود دارد .

درگاهها ورودی / خروجی (I/O) به سیگنال های دیجیتال بیرونی امکان می دهند که با میکرو کنترلر ارتباط پیدا کند .

درگاههای I/O معمولاً به صورت گروههای ۸ بیتی دسته بندی می شوند و به هر گروه نیز نام خاصی اطلاق می شود. به

عنوان مثال ، میکروکنترلر ۸۰۵۱ دارای ۴ درگاه ورودی / خروجی ۸ بیت می باشد که P3, P2, P1, P0 نامیده می

شوند. در تعدادی از میکرو کنترلر ها ، جهت خطوط درگاه I/O قابل برنامه ریزی می باشد. لذا بیت های مختلف یک

درگاه را می توان به صورت ورودی یا خروجی برنامه ریزی نمود. در برخی دیگر از میکروکنترلرها) از جمله

میکروکنترلرهای ۸۰۵۱) درگاههای I/O به صورت دو طرفه می باشند. هر خط از درگاه I/O این گونه میکرو کنترلرها

را می توان به صورت ورودی و یا خروجی مورد استفاده قرار داد . معمولاً ، این گونه خطوط خروجی ، به همراه

مقاومت های بالا کش بیرونی به کار برده می شوند.

معرفی انواع میکرو کنترلر :

تمام میکروکنترلرها جزء این ۵ قسمت هستند :

8051(1

Pic(2

Avr(3

۴) امروزه میکروکنترلرهای زیادی را تولید می میشود (مثل شرکت های شریک شرکت ARM که انواع



البته مدل های ۶۸۱۱ ساخت شرکت موتورلا و z8 ساخت شرکت زیلوگ حداقل در ایران خیلی کم استفاده می شوند و رقابت اصلی بین سه نوع دیگر است .

تا به امروز هر میکروکنترلری که ساخته شده زیر مجموعه یکی از این ۵ نوع است. البته کارخانه های خیلی زیادی با مارک های مختلف میکرو کنترلر تولید میکنند ولی همه اونها زیر مجموعه یکی از این ۵ قسمت هستند. شما برای هر کدام از این ۵ نوع میکروکنترلر میتونید میکروکنترلر های مختلفی از شرکت های مختلفی را پیدا کنید.(البته در بازار ایرن کمی با مشکل).

اما خوشبختانه همه میکروکنترلر هایی که جزء هر کدام از ۵ نوع بالا باشند از یک برنامه پیروی میکنند. بدین معنا که اگر شما کار با یکی از مدل های آن میکرو را یاد گرفته باشید مثل اینکه کار با تمام میکروکنترلرهای آن نوع را یاد گرفته اید.مثلا شما اگر با یکی از مدل های میکروکنترلر avr مثلا atmega8 را یاد گرفته باشید دیگر با صد ها مدل دیگر میکروکنترلر avr مشکلی ندارید و تقریبا بدون هیچ مشکلی میتوانید با دیگر مدل های این میکرو کار کنید .

اما یه مشکل که در میکروکنترلر ها وجود دارد این است که این ۵ نوع از لحاظ برنامه نویسی به هیچ وجه با هم دیگر سازگاری ندارند . به طور مثال اگر شما میکروکنترلر های avr و ۸۰۵۱ را کامل یاد گرفته باشید حتی ساده ترین برنامه رو روی یک میکروکنترلر pic نمیتوانید اجرا کنید. واین یکی از بزرگترین عیب و مشکل برای یاد گیری میکرو

است .بنابراین از همون اول باید یک انتخاب درست داشته باشید و میکروکنترلر مناسب را برگزینید تا با یادگیری آن

میکروکنترلر بتوانید بعدا به سادگی پروژه های خود را اجرا کنید . البته بسیاری از دوستان هستند که کار با چند

میکروکنترلر را میدوندند و حتما این هم از هوش بالای ایرانی هاست. ولی اگر به صورت خیلی حرفه ای نخواهید وارد این



بحث بشوید باید یکی از این میکروکنترلرها را انتخاب کنید و کار با آن را آغاز کنید. در قسمت بعدی شما را برای این انتخاب کمک خواهیم کرد .

معایب و مزایای میکروکنترلر های مختلف نسبت به هم

از آن جای که ۶۸۱۱ و Z8 خیلی کمتر استفاده میشوند به معرفی سه نوع دیگر میپردازم .

(1) : اول از ۸۰۵۱ که اولین میکروکنترلری بود که به دست بشر ساخته شد شروع میکنیم . همانطور که در مقاله قبلی

گفته شد ابتدا این میکروکنترلر توسط شرکت بزرگ intel ساخته شد . اما بعدا intel این امکان را به دیگر شرکت ها

داد که این میکروکنترلر را تولید کنند و شرکت هایی مانند ATMEL , PHILIPS , SIEMENS , DALLAS

و... به تولید این میکروکنترلر پرداختند یکی از شرکت هایی که به صورت گسترده به تولید این تراشه پرداخت

ATMEL بود که مدل های مختلف میکروکنترلر ساخت این شرکت در سراسر جهان و در ایران به خوبی یافت می شود.

اما اگر بخواهیم به صورت کلی سیر پیشرفت این نوع میکروکنترلر رو در نظر بگیریم اولین میکروکنترلر هایی که ساخته

شد با جدیدترین میکروکنترلرهای ۸۰۵۱ که الان تولید میشود با توجه به این پیشرفت شگفت در تمام زمینه ها که

صنایع دیگر در دنیا دارند پیشرفت زیادی ندارد به طور مثال AT89S5X که میکروکنترلر ۸۰۵۱ جدید ساخت

ATMEL است نسبت به مدل های اولیه ۸۰۵۱ پیشرفت آنچنانی ندارد . امکانات این میکرو نسبت به AVR و PIC

قابل مقایسه نیست . به صورتی که که همین مدل جدید ۸۰۵۱ تقریبا حافظه ای برابر یک صدم (۰.۰۰۱) میکروکنترلر

های AVR را دارد و سرعتش ۴ برابر کمتر از میکروکنترلر های PIC و ۱۲ بار کمتر از میکروکنترلر های AVR

است . از لحاظ امکانات دیگر هم چنین وضعی احساس میشود. اما برای کارهای ساده تر که پیچیدگی زیادی در آن



نباشد به خاطر قیمت بسیار پایینی که این میکروکنترلر دارد بسیار مناسب است . قیمت همین مدل جدید

AT89S5X حول و حوش ۱۰۰۰ تومان است که قیمت بسیار مناسبی است .

این میکرو کنترلر از زبان اسمبلی و C پشتیبانی میکند که زبان برنامه نویسی اصلی آن اسمبلی است که واقعا نوشتن با

این زبان برنامه نویسی نسبت به زبان های برنامه نویسی دیگر هم مشکل تر و هم طولانی تر است. در کل این

میکروکنترلر امروزه دیگر توانای رقابت با AVR و PIC رو ندارد و امروزه رقابت اصلی بین این دو میکروکنترلر است .

۲) میکروکنترلر PIC

واقعا میکروکنترلر خیلی قوی است که بر اساس بعضی آمار ها بیشترین کاربر را به خود اختصاص داده است البته متذکر

شوم که در ایران این آمار به نفع AVR است. این میکروکنترلر ساخت شرکت میکرو چیپ است که PIC رو در مدل

های خیلی زیادی با امکانات مختلف برای کارهای مختلف میسازد . این میکروکنترلر با مدل های مختلف PIC16XXX

و PIC12XXXX که به جای X دوم از چپ به راست حروف C, X, E, F قرار میگیره که هر کدام مفهوم خاصی داره

که چون بحث ما آموزش AVR است از روی اون سریع میگذریم X های بعدی هم اعدادی هستند که نشان دهنده

مدل های مختلف هستند .

۳) میکروکنترلر AVR

به میکروکنترلر AVR میرسیم که به نظر من و خیلی از دیگر بهترین میکروکنترلر موجود در بازار است البته خود من

با AVR و AVR بخصوص AVR خیلی کار کردم ولی تجربه ای با PIC ندارم و قضاوت من ممکنه یه طرفه باشه ولی

این نظر بسیاری از کسانی است که با AVR کار کرده اند . به دلایلی

اول از همه سرعت این میکروکنترلر بسیار بالاست و به قولی دستوراتی که بهش داده میشه در یک سیکل کلاک انجام



میده در صورتی که این سیکل کلاک برای ۸۰۵۱ باید تقسیم بر ۱۲ شود و برای PIC باید تقسیم بر ۴ بنابراین AVR

سریعترین میکروکنترلر موجود در بازار است AVR . از زبان های برنامه نویسی سطح بالا یا به اصطلاح (HIGH

HLL (LEVEL LANGUAGE پشتیبانی میکند که باعث تولید کدهای بیشتری میشود که در کل برنامه نوشته

شده نسبت به برنامه هایی که برای ۸۰۵۱ و PIC نوشته میشود کوتاهتر است. امکانات جانبی این میکروکنترلر بسیار

مناسب است و شما را از خرید بعضی لوازم جانبی مانند چیپ های آنالوگ به دیجیتال , (ADC) مقایسه گر آنالوگ و...

راحت میکند. در ضمن AVR از بسیاری از استانداردهای ارتباطی مانند SPI, UART, I2C, JTAG پشتیبانی

میکند که به راحتی میتوان این میکروکنترلر را با میکروکنترلر دیگر یا وسایل دیگر وصل کرد و با وسایل دیگر به راحتی

ارتباط برقرار کند. قیمت این میکروکنترلر هم به نسبت امکانات فراوانی که دارد بسیار پایین است به طوری که یک

میکروکنترلر AVR تقریباً پیشرفته رو با قیمت حول و حوش ۳ تا ۴ هزار تومان خرید .



فصل دوم

نام پروژه: مولد 20 هارمونیک

مفهوم این اسم:

مولد یعنی تولید کننده 20 هم که عدد است و هارمونیک هم به معنی مشتق های یک موج اصلی یا شبیح همان موج اصلی که مقدار دامنه یا فرکانس آن تغییر کرده.

وظیفه این پروژه:

این پروژه یا دستگاه باید 20 موج سینوسی را توسط keyboard بگیرد و با هم جمع لحظه ای کند و در خروجی به یک موج تبدیل کند که ما توسط keyboard فقط مقدار دامنه و فرکانس را وارد می کنیم .

سخت افزار این پروژه:

1: کی پد

2: LCD

3: DAC

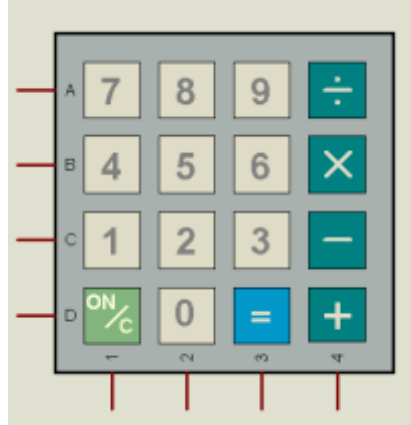
4: میکروکنترلر

این پروژه از **4 قسمت اصلی** سخت افزاری تشکیل می شود.



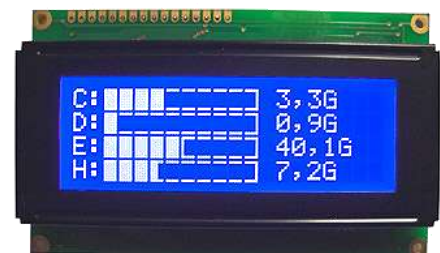
Keyboard:

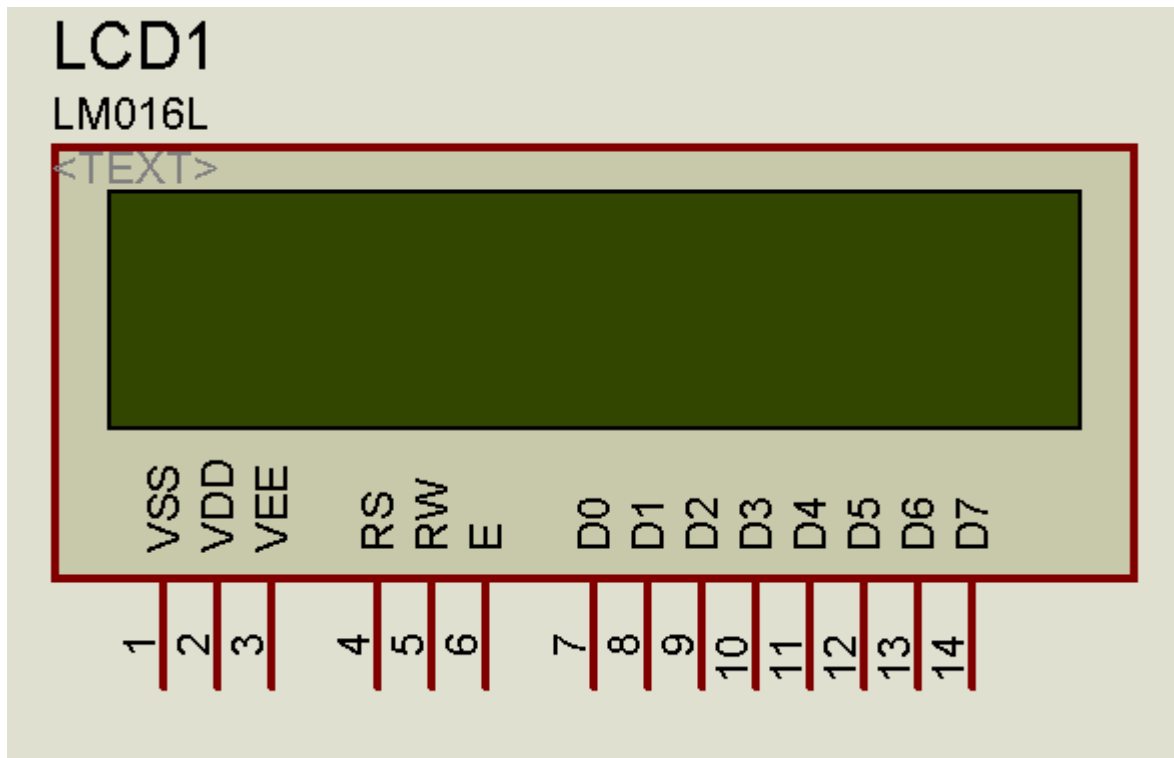
این قسمت این وظیفه را بر عهده دارد که مقدار دامنه و فرکانس را توسط آن وارد میکرو کنیم .
 که بعد میکرو از آن استفاده کند و در محاسبات آن را در نظر بگیرد. که شکل ظاهری آن و برنامه مختص آن در زیر آمده است.



LCD:

این المان این وظیفه را دارد که مقدار دامنه و فرکانس را که توسط Keyboard وارد می کنیم را به ما نشان دهد. که شکل ظاهری آن در زیر آمده است.





:DAC

این المان این وظیفه را دارد که مقدار محاسباتی را که میکرو انجام داده است رابه موج تبدیل کند که ما بتوانیم در خروجی ببینیم . که برای این کار از DAC 0808 استفاده کردم. که شکل ظاهری آن ونحوه نصب پایه های آن در زیر آمده است. در اینجا نیاز است که ساخت یک موج سینوسی را توسط DAC توضیح دهیم

تولید موج سینوسی :

برای تولید موج سینوسی به جدولی نیاز داریم که مقادیرش اندازه سینوس را با زاویه 0 تا 360 درجه بدست دهد. مقادیر برای تابع سینوسی بین -1.0 تا +1.0 به ازای زاویه 0 تا 360 درجه تغییر نمایند . بنابر این مقادیر جدول اعداد صحیحی هستن که مقدار ولتاژ را برای سینوس 0 (تتا) نمایش می دهند. در این روش فقط خروجی اعداد صحیح از mega 32 dac به dac را تضمین می کند جدول زیر زوایای مقادیر سینوس اندازه ولتاژ و مقادیر صحیح را نمایش دهنده اندازه ولتاژ را برای هر زاویه با افزایش ۳۰ درجه نشان می دهد. برای تولید جدول زیر فرض می کنیم حداکثر ولتاژ خروجی dac هنگامی حاصل می شود که همه داده ها ورودی به dac در سطح بالا باشند . بنا براین برای دست یابی به حداکثر مقدار 10 v خروجی معادله زیر را بکار می بریم .



$$V_{out}=5v+(5\times\sin\theta)$$

برای یافتن مقدار ارسالی به dac به ازای زاویه مختلف ولتاژ V_{out} را در 25.60 ضرب می کنیم زیرا 256 مرتبه وجود دارد و V_{out} حداکثر هم 10 v است بنا براین $256 \div 10 = 25.6$ پله به ازای هر ولت وجود دارد .

زاویه θ (درجه)	$\sin\theta$	V_{out} (اندازه ولتاژ) $5v+(5v\times\sin\theta)$	مقادیر ارسالی به dac (دهد هی) (voltage $\text{mag}\times 25.6$)
0	0	5	128
30	0.5	7.5	192
60	0.866	9.33	238
90	1.000	10	255
120	0.866	9.33	238
150	0.5	7.5	192
180	0	5	128
210	-0.5	2.5	64
240	-0.86	0.699	17
270	-1.0	0	0
300	-0.86	0.699	17
330	-0.5	2.5	64
360	0	5	128

مبدل آنالوگ به دیجیتال میکرو:

از خصوصیات های میکروکنترلرهای AVR داشتن یک ADC داخلی است که با چند کانال



آنالوگ مالتی پلکس شده که به هر یک از پایه های پورت اجازه می دهد که به عنوان یک ورودی مبدل آنالوگ به

دیجیتال عمل نماید. مبدل داخلی میکرو دارای وضوح ۱۰ بیتی است و برای تبدیل با این وضوح نیاز به فرکانس کلاکی

بین ۵۰ کیلوهرتز تا ۲۰۰ کیلوهرتز دارد و این کلاک از تقسیم فرکانس کریستال تامین می شود.

ADC دارای دو منبع ولتاژ آنالوگ AV_{CC} و $AGND$ است. که $AGND$ بایستی به

AV_{CC} باشد. زمین وصل شود. ولتاژ مرجع خارجی در صورت وجود باید به پایه ۳۲ وصل شود که این ولتاژ

باید بین ولتاژهای $AGND$ و

پیکره بندی ADC :

$Config\ Adc = Single , Prescaler = Auto , Reference = Av_{cc}$

اگر ولتاژ مرجع مانند برنامه بالا AV_{CC} تعریف شود می توانیم ولتاژی بین ۰ تا ۵ ولت به

ورودی ADC بدهیم خروجی دیجیتال شده ADC در این حالت برای ورودی ۵ ولت می شود

۱۰۲۳. پس اگر به طور مثال ۲ ولت به ورودی دهیم خروجی ADC عدد دیجیتال شده ۲۰۴

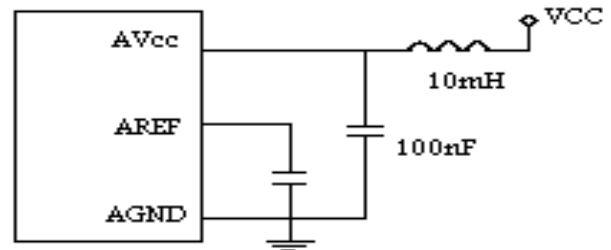
میشود.

تکنیک کاهش نویز ADC :



برای کاهش نویز در زمان نمونه برداری ADC در حالتی که ولتاژ AVCC به عنوان مرجع

انتخاب شده میکرو را به صورت شکل زیر قرار می دهیم.



Application

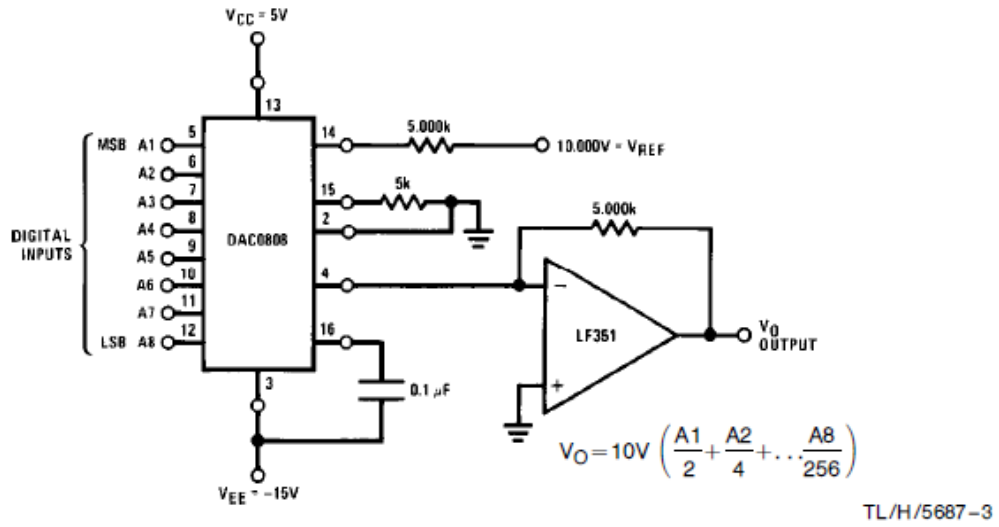
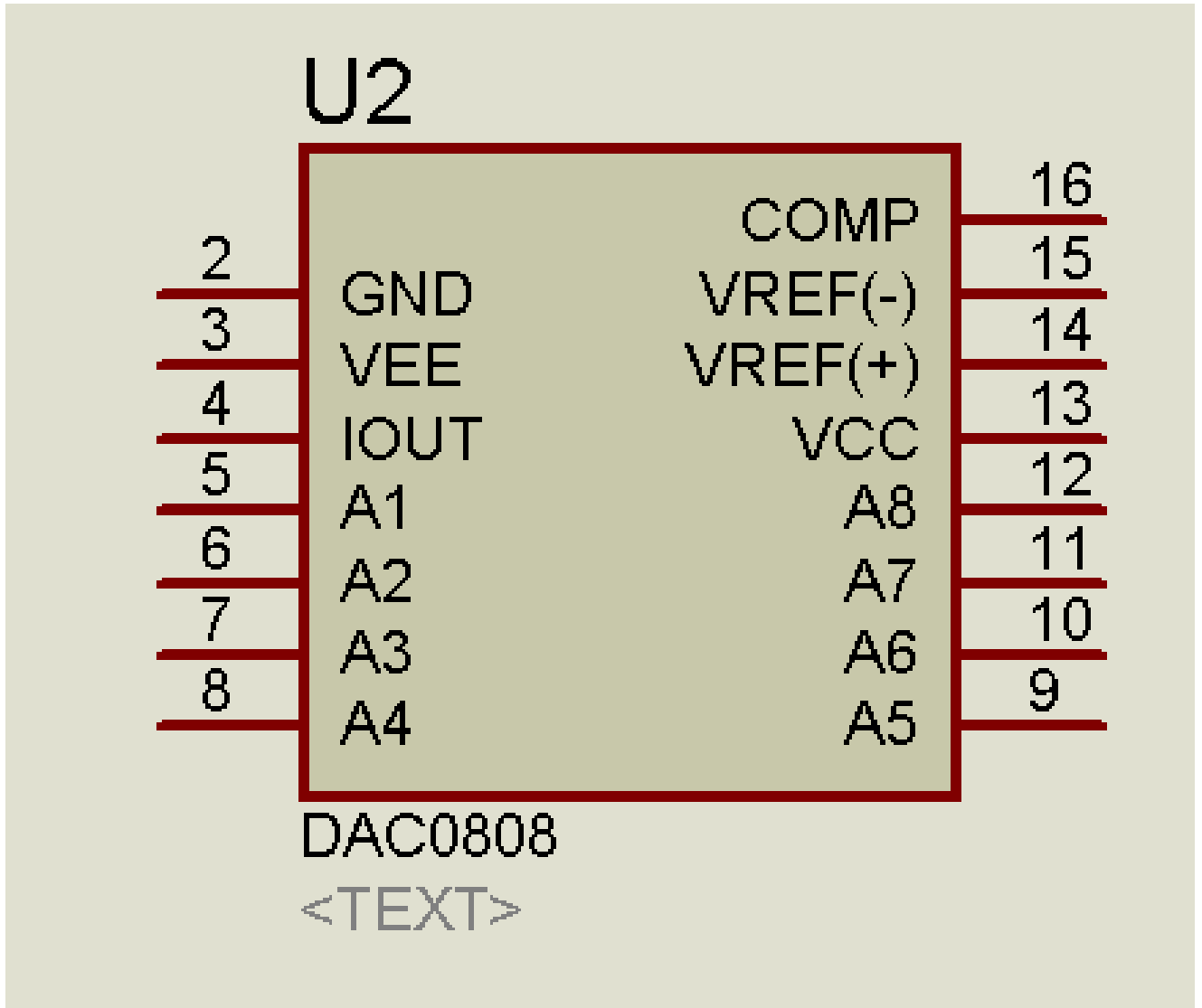


FIGURE 1. +10V Output Digital to Analog Converter (Note 7)

چگونگی وصل میکرو به DAC 0808

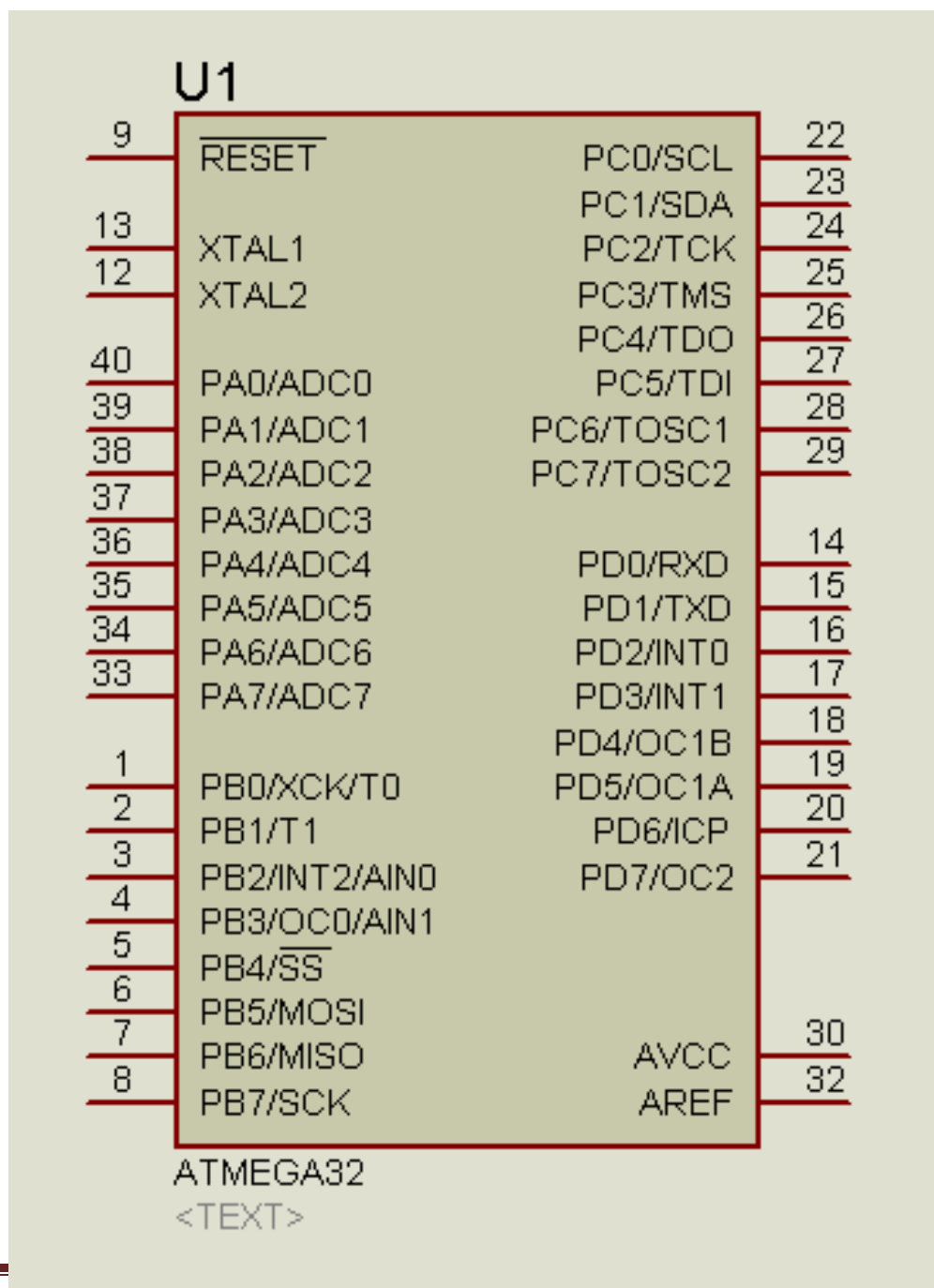


شکل ظاهری DAC808



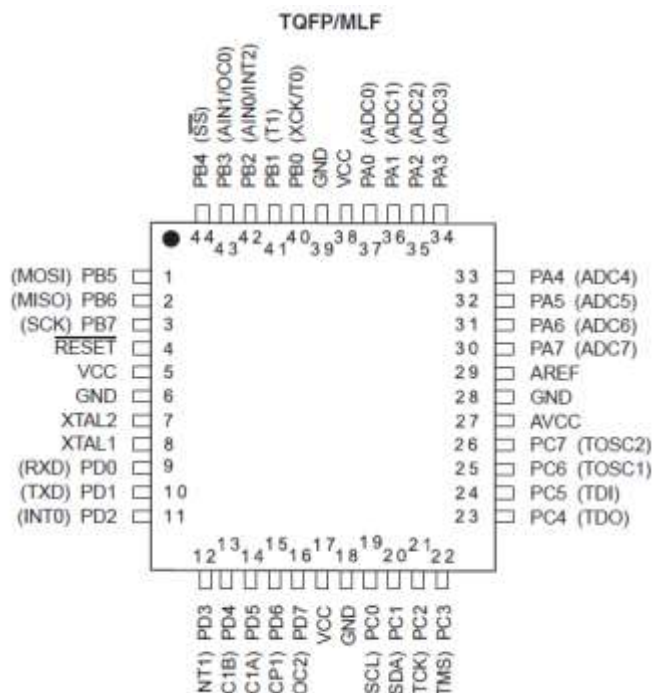
میکروکنترلر:

این المان واحد پردازش مرکزی ما است که مقدار موج خروجی را محاسبه می کند و این مقدار را به DAC می دهد تا DAC آن را در خروجی نشان دهد. که شکل آن در زیر آمده و قسمت اصلی این پروژه است.



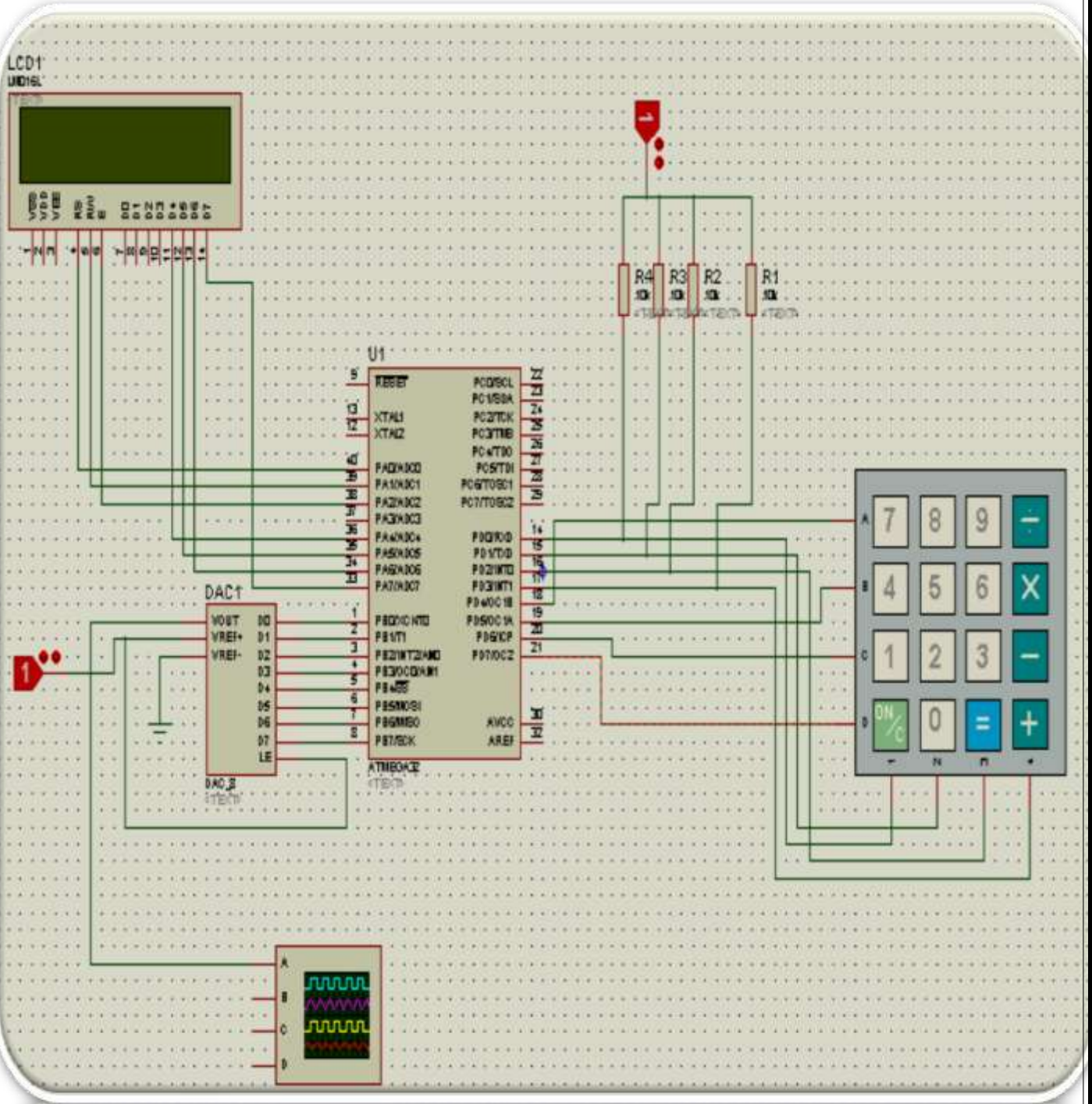


(XCK/T0) PB0	1	40	PA0 (ADC0)
(T1) PB1	2	39	PA1 (ADC1)
(INT2/AIN0) PB2	3	38	PA2 (ADC2)
(OC0/AIN1) PB3	4	37	PA3 (ADC3)
(\overline{SS}) PB4	5	36	PA4 (ADC4)
(MOSI) PB5	6	35	PA5 (ADC5)
(MISO) PB6	7	34	PA6 (ADC6)
(SCK) PB7	8	33	PA7 (ADC7)
\overline{RESET}	9	32	AREF
VCC	10	31	GND
GND	11	30	AVCC
XTAL2	12	29	PC7 (TOSC2)
XTAL1	13	28	PC6 (TOSC1)
(RXD) PD0	14	27	PC5 (TDI)
(TXD) PD1	15	26	PC4 (TDO)
(INT0) PD2	16	25	PC3 (TMS)
(INT1) PD3	17	24	PC2 (TCK)
(OC1B) PD4	18	23	PC1 (SDA)
(OC1A) PD5	19	22	PC0 (SCL)
(ICP1) PD6	20	21	PD7 (OC2)





شکل کلی سخت افزار :



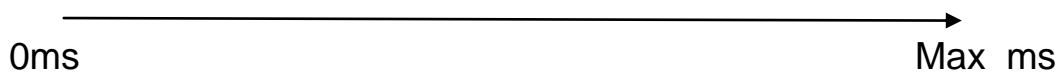


توضیح کلی انجام کار:

ابتدا ما 20 دامنه و 20 فرکانس را توسط Keyboard وارد میکرو می کنیم بعد از این کار میکرو مقدار فرکانس را توسط رابطه $F = 1/t$ به زمان تناوب تبدیل می کند در نتیجه ما 20 زمان تناوب داریم میکرو از بین این 20 زمان تناوب بزرگترین زمان تناوب را انتخاب می کند که علت آن و نحوه انجام این کار بعدا توضیح داده می شود در نتیجه همچنین بستی ایجاد می شود که مانند بست زیر است.

$$A \sin w_1t + B \sin w_2t + C \sin w_3t + D \sin w_4t + E \sin w_5t$$

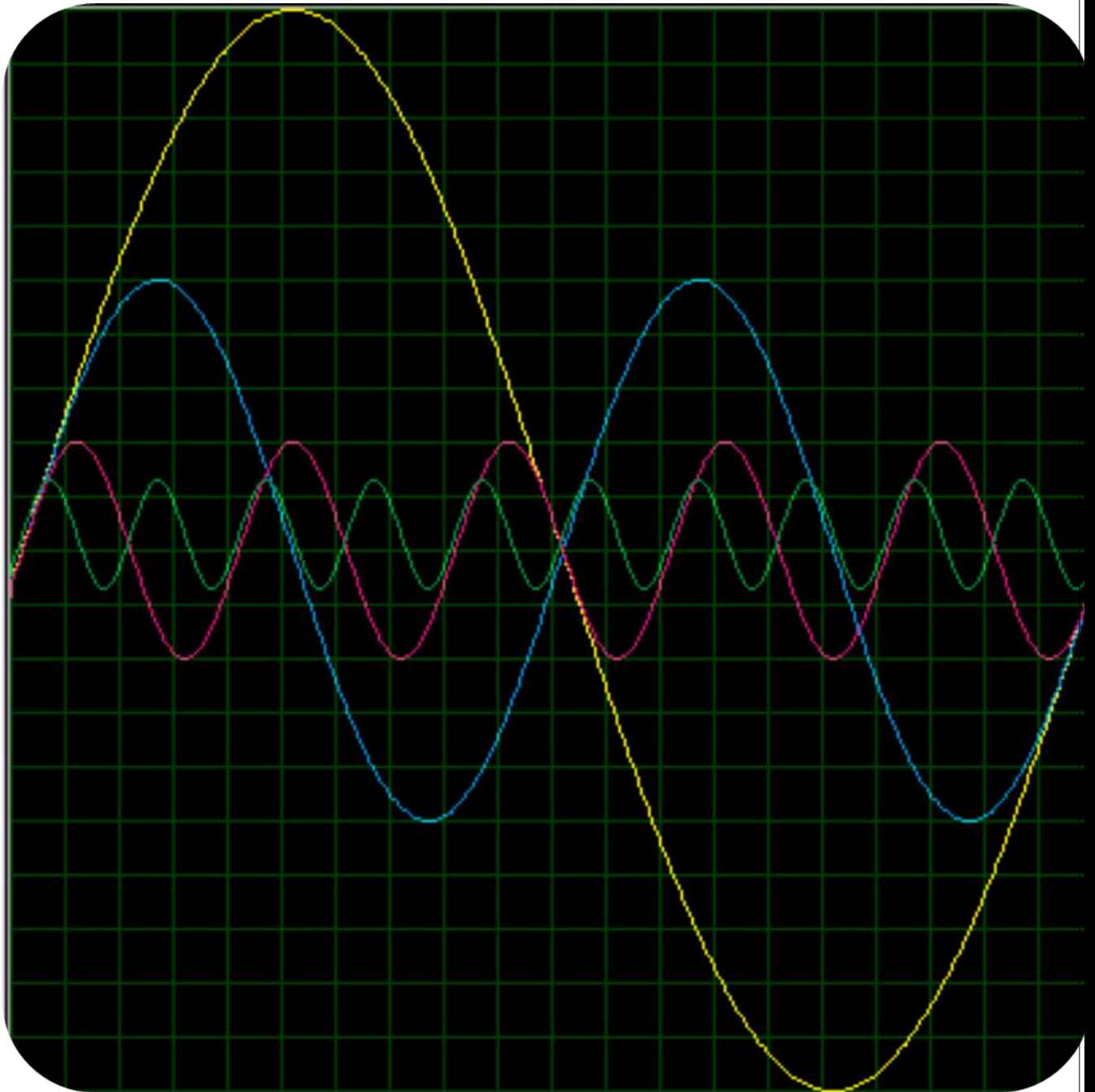
که ما برای ساده کردن تفهیم نحوه کار از 5 بست استفاده می کنیم .
که A,B,C,D,E دامنه موج هستند.



که در اینجا ما کزیموم زمان تناوب همان بزرگترین زمان تناوب یا t مادر است .
بعد از مشخص شدن t مادر ما مقدار t را از صفر شروع می کنیم تا مقدار ما کزیموم می رویم یا به عبارت دیگر آن را به 50 قسمت تقسیم می کنیم و هر دفعه $1/50$ زمان تناوب اصلی را در این بست قرار می دهیم در نتیجه یک مقدار عددی می شود که اگر آن را تا $50/50$ برویم در نتیجه ما 50 مقدار عددی داریم که آنرا به ورودی DAC می دهیم و در خروجی شکل موج خروجی را دریافت می کنیم



علت گرفتن بزرگترین T:





برای فهم بیشتر شکل بالا از 4 تا شکل موج سینوسی با فرکانس مختلف تشکیل شده است که موج زرد، آبی، قرمز و سبز به ترتیب با فرکانس

$$\begin{aligned} F_{\text{زرد}} = 1000\text{HZ} & \longrightarrow T = 1\text{ms} \\ F_{\text{آبی}} = 2000\text{HZ} & \longrightarrow T = 0.5\text{ms} \\ F_{\text{قرمز}} = 5000\text{HZ} & \longrightarrow T = 0.2\text{ms} \\ F_{\text{سبز}} = 10000\text{HZ} & \longrightarrow T = 0.1\text{ms} \end{aligned}$$

اگر ما برای مثال t شکل موج آبی را به عنوان t مادر بگیریم شکل موج زرد به طور کامل در محاسبات شکل نمی گرد و نیم سیکل آن محاسبه می شود در نتیجه ما باید t را که به عنوان t مادر می گیریم t باشد که همه فرکانس ها در آن تشکیل بشود و اگر این طور نباشد شکل موج خروجی درست نمی باشد.

نحوه گرفتن T مادر :

در اصل t مادر همان t ک م م می باشد در نتیجه ما باید بین 20 زمان تناوب ک م م بگیریم که مرحله سخت این پروژه محسوب می شود.

آن چیزی که ما از ک م م به یاد داریم از دوران راهنمایی این بود که اگر بخواهیم بین دو عدد ک م م بگیریم باید آن را تجزیه می کردیم و بعد از این کار ک م م را می گرفتیم .

نحوه محاسبه ک م م به روش دوران راهنمایی:

برای این کار دو مثال زده شده است.

$$\begin{aligned} N1 &= 24 \\ N2 &= 12 \end{aligned}$$

تجزیه 24:

$$24 = 2^3 \times 3$$

تجزیه 12

$$12 = 2^2 \times 3$$

عامل غیر مشترک \times عامل مشترک با بیشترین توان = ک م م

$$24 = 2^3 \times 3 = 24 \text{ ک م م}$$

$$\begin{aligned} N1 &= 25 \\ N2 &= 20 \end{aligned}$$

تجزیه 25:

$$25 = 5^2$$

تجزیه 20:

$$20 = 2^2 \times 5$$



$$k = 2^2 \times 5^2 = 100$$

همان طوری که می بینید گرفتن ک م م به این روش برای میکرو بسیار سخت می باشد در نتیجه من از فرمول دیگری برای این کار استفاده کردم که در زیر آمده است. ب م م

$$k = \frac{N1 \times N2}{b \ m \ m}$$

این فرمول راحتتر است زیرا نیاز به تجزیه ندارد اما مشکل دیگر ما ب م م است که برای رفع آن از این روش استفاده می کنیم آن این است که عدد بزرگتر را به عدد کوچکتر تقسیم می کنیم .

این فرمول راحتتر است زیرا نیاز به تجزیه ندارد اما مشکل دیگر ما ب م م است که برای رفع آن از این روش استفاده می کنیم آن این است که عدد بزرگتر را به عدد کوچکتر تقسیم می کنیم .

1: در دفعه اول باقی مانده این تقسیم صفر می شود در نتیجه ب م م ما مقسوم علیه است

2: اما اگر در تقسیم اول باقی مانده صفر نشد مقسوم علیه را جای عدد اولی می گذاریم و باقی مانده را جای مقسوم علیه می بریم و این کار را آنقدر تکرار می کنیم تا باقی مانده صفر شود.

دو حالت پیش می آید

$$N1=24$$

$$N2=12$$

$$24 \div 12 = 2 \longrightarrow \text{باقی مانده} = 0$$

$$k = (24 \times 12) \div 12 = 24$$

$$N1=25$$

$$N2=20$$



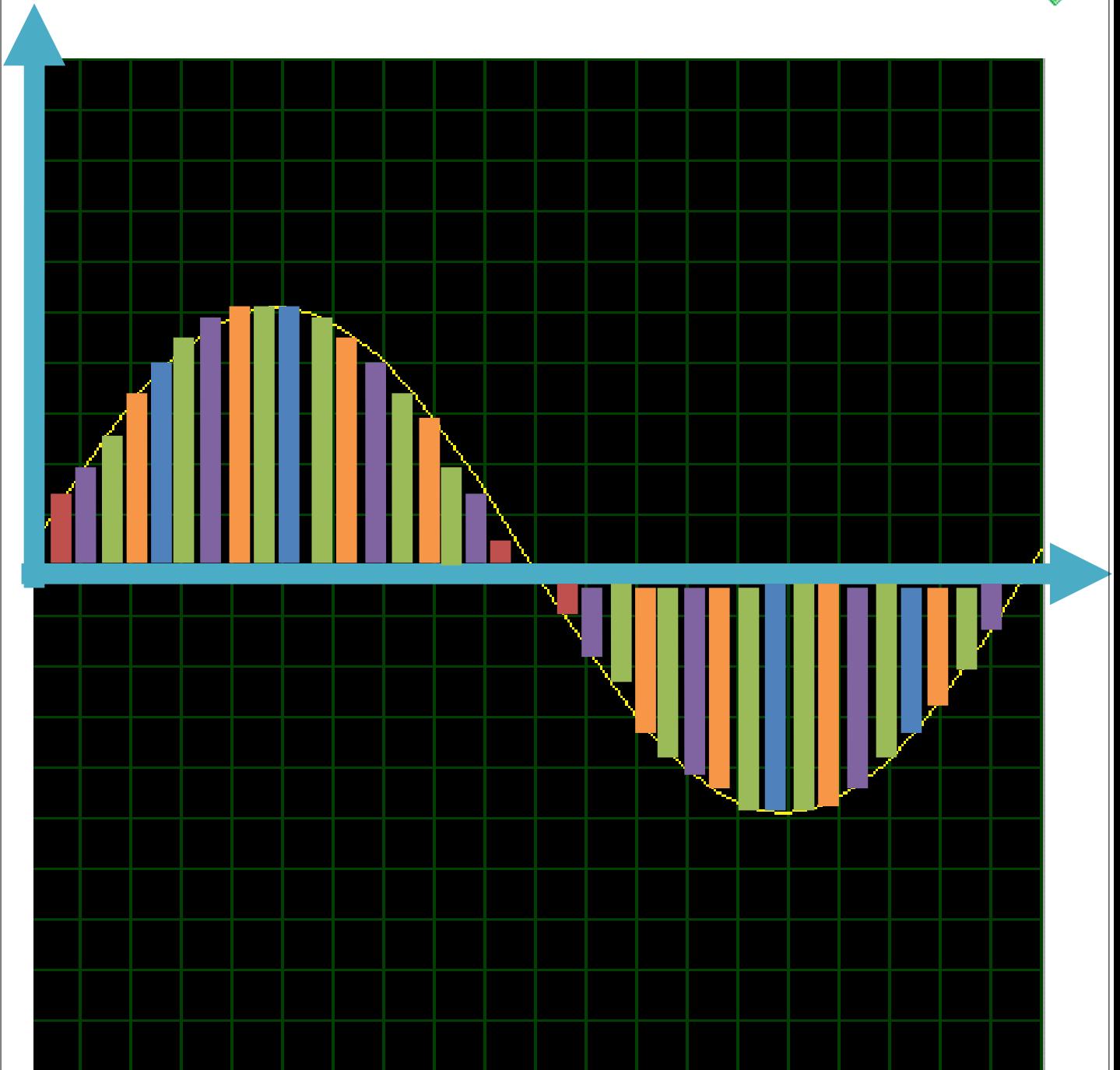
$$25 \div 20 = 1 \longrightarrow \text{باقی مانده} = 5$$

$$20 \div 5 = 4 \longrightarrow \text{باقی مانده} = 0$$

$$k \text{ م م} = (25 \times 20) \div 5 = 100$$

فرکانس خروجی:

مشکل دیگر این پروژه فرکانس خروجی بود چون فرکانسی که در خروجی باید ایجاد میشد باید برابر با فرکانس بزرگترین t یا (ک م م) میشد که این کار را با وقفه و تایمر انجام دادم همان طوری که قبلا گفتیم آن را به ۵۰ قسمت یا بیشتر تقسیم می کنیم در نتیجه برای اینکه فرکانس ورودی با خروجی برابر شود تایمر را راه می اندازیم که هر $1 \div 50$ را بشمارد و بد از رسیدن به مقدار وارد وقفه شود و یک عدد به dac بدهد تا شکل موج خروجی را بینیم.



در این موج ابتدا تایمر مدت زمان قرمز را می شمارد و بعد به وقفه رفته و یک عدد به dac می دهد وبعد مدت زمان بنفش را می شمارد ودوباره به وقفه رفته و یک عدد دیگر اضافه می کند واین روند ادامه دارد تا شکل موج کامل شود .



فصل سوم

قسمت های مختلف برنامه:

برنامه شناخت Keyboard:

```

unsigned char scan_key(void)
{
    unsigned char code[4][4]={{0,4,8,12},{1,5,9,13},{2,6,10,14},{3,7,11,15}};
    unsigned char i,data,num_key,temp,p;
    num_key=0xff;
    temp=0x70;

    for(i=0;i<4;i++)
    {
        PORTD=temp;
        delay_ms(10);
        data=PIND&0X0F;
        delay_ms(10);
        data=PIND&0X0F;
        if(data==0x07)
            num_key=code[0][i];
        if(data==0x0b)
            num_key=code[1][i];
        if(data==0x0d)
            num_key=code[2][i];
        if(data==0x0e)
            num_key=code[3][i];
        temp=((temp>>1)|0x80)&0xf0;
    }
    return num_key;
}
    
```



در این برنامه ابتدا سطر ها را یک در میان صفر می کند و کد متناسب با عددی که فشرده شده است را به میکرو می دهد
و میکرو آن عدد را می شناسد.

وارد کردن دامنه:

```
void inter_amp(void)
{
char key;
char sw=1;
float b=0;
float n=0;
float s=1;
char i=0;
while(i<5)
{
key=scan_key();
delay_ms(20);
if(key!=0xff)
{
if(key==12)
{
sw=0;
}

if(sw==1)
{
if(key>=0 && key<=9)
{
b=key+b*10;
}
}
if(sw==0)
{
if(key>=0 && key<=9)
{
s=s*.1;
n=s*key+n;
}
}
}
}
```



```

if(key==13)
{
    b=0;
    n=0;
    s=1;
}
if(key==14)
{
    i=0;
    b=0;
    n=0;
    s=1;
}

if(key==15)
{
    mux[i]=b+n;
    i++;
    sw=1;
    s=1;
    n=0;
    b=0;
}
}
}
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("enter amp");
lcd_gotoxy(0,1);
ftoa(n+b,dec,buff);
lcd_puts(buff);
}
}

```

می توان گفت که این برنامه برای گرفتن عدد برنامه تکمیل و کاملی است زیرا قادر است اعداد ممیزی را نیز بگرد علاوه بر آن می توان تا عدد حدود 12 رقمی را داد یا عدد 12 رقم بعد از اعشار را به میکرو داد . در ابتدا اگر کلید شماره 12 زده شود میکرو وارد تابع اعشاری می شود و تا 12 رقم بعد از اعشار را می توان نوشت . و اگر عدد 12 زده نشود وارد تابع اعداد صحیح می شود و تا 12 رقم اعداد صحیح را می تواند بگیرد. عدد 13 برای این در نظر گرفته شده تا اگر اشتبا هی صورت گرفت آن را پاک کند و عدد 14 برای این در نظر گرفته شده است که اگر



در بین کار نیاز به ریست بود با زدن این دکمه ریست انجام شود و عدد 15 برای این که عدد مورد نظر به میکرو وارد شود و روی lcd نمایش داده شود .
وارد کردن فرکانس:

```
void inter_w(void)
{
    unsigned int v;
    char key;
    char sw=1;
    float b=0;
    float n=0;
    float s=1;
    char i=0;
    while(i<5)
    {
        key=scan_key();
        delay_ms(20);
        if(key!=0xff)
        {
            if(key==12)
            {
                sw=0;
            }
            if(sw==1)
            {
                if(key>=0 && key<=9)
                {
                    b=key+b*10;
                }
            }
            if(sw==0)
            {
                if(key>=0 && key<=9)
                {
                    s=s*.1;
                    n=s*key+n;
                }
            }

            if(key==13)
            {
                b=0;
            }
        }
    }
}
```




```

        n=0;
        s=1;
    }
    if(key==14)
    {
        i=0;
        b=0;
        n=0;
        s=1;
    }
    if(key==15)
    {
        freq[i]=b+n;
        time[i]=(1/b+n)*10000;
        i++;
        sw=1;
        s=1;
        n=0;
        b=0;
    }
}
    lcd_clear();
    lcd_gotoxy(0,0);
    lcd_putsf("enter w");
    lcd_gotoxy(0,1);
    ftoa(n+b,dec,buff);
    lcd_puts(buff);
}
}

```

وارد کردن فرکانس شباهت بسیاری به وارد کردن دامنه دارد با این تفاوت که در انتهای برنامه 1 تقسیم بر عدد حاصل می شود و برای گرفتن ک م م در یک 10000 ضرب می شود تا عدد بزرگتری شود و گرفتن ک م م راحتتر باشد.



ک م م گرفتن:

```
float result_w(void)
{
double x,y,r,temp,sw,k,u,v,p,w;
char i=0;
for(i=0;i<5;i++)
{
sw=1;
x=time[0];
y=time[i];
p=x*y;
if(x<y)
{
temp=y; // Place your code here
y=x;
x=temp;
};
while(sw)
{
r=x-(x/y*y);
if(r==0)
{
k=(p/y);
time[0]=k;
sw=0;
}
else
{
x=y;
y=r;
sw=1;
}
}
}
}
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("kmm");
lcd_gotoxy(0,1);
ftoa(time[0],dec,buff);
lcd_puts(buff);
delay_ms(50);
```



```
w=time[0];
w=w*.0001;
return w;
}
```

در برنامه بالا عدد اولی را تقسیم بر عدد دومی می کند در صورتی که عدد اولی بزرگتر باشد در غیر این صورت جای عدد اولی را با عدد دومی جا به جا میکند بعد از این کار اگر باقی مانده صفر شد مقسوم علیه را به عنوان ب م م می گیرد در غیر این صورت مقسوم علیه را جای عدد اصلی قرار می دهد و باقی مانده را جای مقسوم علیه قرار می دهد و این کار را تا جای که حاصل تقسیم یعنی باقی مانده صفر شود آن را ادامه می دهد و بعد از این کار آن را روی LCD نمایش می دهد .

نحوه استفاده از بست و تایمر برای ایجاد فرکانس خروجی مناسب :

```
for(i=0;i<50;i++)
{
t=(mux[0]*(sin(2*3.141592654*freq[0]*p*i)))+(mux[1]*(sin(2*3.141592654*freq[1]*
p*i)))+(mux[2]*(sin(2*3.141592654*freq[2]*p*i)))+
(mux[3]*(sin(2*3.141592654*freq[3]*p*i)))+(mux[4]*(sin(2*3.141592654*freq[4]*p*
i)));
t=(t+5)*25.5;
if(t<0)
{
t=0;
}
lcd_clear();
lcd_gotoxy(0,0);
ftoa(t,dec,buff);
lcd_puts(buff);
lcd_gotoxy(0,1);
ftoa(p,dec,buff);
lcd_puts(buff);
delay_ms(10);
dac[i]=t;
t=0;
}
f=1/w;
if(f>1000)
{
f=f/1000;
lcd_clear();
lcd_gotoxy(0,1);
ftoa(f,dec,buff);
lcd_puts(buff);
lcd_gotoxy(0,0);
lcd_putsf("coefficient");
}
```



```

}
over=p*8000000;
over=65536-over;
tim1l=over & 0x00ff;
#asm("sei")
TIMSK=0x04;
ACSR=0x80;
SFIOR=0x00;
tim1h=(over & 0xff00)>>8;
TCNT1L=tim1l;
TCNT1H= tim1h;
TCCR1A=0x00;
TCCR1B=0X01;
while(1)
{
};
/*while(sw)
{
for(i=0;i<50;i++)
{
PORTB=dac[i];
}
key=scan_key();
if(key==15)
{sw=0;
}
}
*/
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("finish");

}

```

این قسمت برنامه حذف شده



در این قسمت برنامه i همان (1÷50) است که آن را در یک for 50 تایی قرار دادیم تا هر دفعه مقدار کل بست را در زمان های متفاوت حساب کند و به dac بدهد و کل آن را در متغییری به نام t می ریزد و در اینجا چون dac نمی تواند منفی را نمایش دهد آن را با 5 جمع کردیم و برای این که حداکثر 10v p-p داشته باشیم آن را در 25.5 ضرب کردیم. و قبل از اینکه از تایمر برای تنظیم فرکانس خروجی استفاده کنیم از یک for برای اینکه آن را در متغییری بریزم استفاده کرده بودم که این باعث می شد تا هر فرکانسی که می دادیم در خروجی 200hz به ما بدهد که علت آن این بود که بدون ترتیب وزمان بندی خاصی آن را به dac می دادیم و این مسئله باعث می شد تا فرکانس خروجی با ورودی تنظیم نبا شد که این قسمت را بعد حذف کردم که برای توضیح بیشتر آن را در برنامه آوردم بعد از این کار از تایمر در مد نرمال استفاده کردم که تایمر یک بود که 16 بیتی بود وقفه آن را تنظیم کردم تا زمانی که تایمر شمارشش به پایان رسید وارد تایع وقفه شود و یک عدد به dac بدهد .

قسمت وقفه برنامه :

```
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
```

```
{
    // Place your code here
    PORTB=dac[step];
    step++;
    if(step==50)

        step=0;

    TCNT1H=tim1h;
    TCNT1L=tim1l;
}
```

در این قسمت برنامه زمانی که شمارش تایمر به پایان می رسد به این تابع که تابع وقفه بود می پرید و یک عدد به پورت b می داد که همان dac بود و زمانی که به انتهای عدد می رسید متغییر را صفر می کند و از ابتدا شروع می کند .



```

float exam;
int step=0;
int tim1h,tim1l;
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
    // Place your code here
    PORTB=dac[step];
    step++;
    if(step==100)
    step=0;
    TCNT1H=tim1h;
    TCNT1L=tim1l;
}

void main(void)
{
    // Declare your local variables here
    char k,j,key;
    char sw=1;
    float t;
    int i;
    float w,m,r;
    float p,s,f;
    int over;
    PORTD=0x00;
    DDRD=0xf0;
    PORTB=0x00;
    DDRB=0xFF;
    // LCD module initialization
    lcd_init(16);
    lcd_clear();
    inter_amp();
    inter_w();
    w=result_w();
    lcd_clear();
    lcd_gotoxy(0,0);
    ftoa(w,dec,buff);
    lcd_puts(buff);
    delay_ms(10);
    p=w/50;
    lcd_gotoxy(0,1);
    ftoa(p,dec,buff);
    lcd_puts(buff);
}
    
```



```
delay_ms(10);
```

```
for(i=0;i<100;i++)
```

```
{
```

```
t=(mux[0]*(sin(2*3.141592654*freq[0]*p*i)))+(mux[1]*(sin(2*3.141592654*freq[1]*p*i)))+(mux[2]*(sin(2*3.141592654*freq[2]*p*i)))+(mux[3]*(sin(2*3.141592654*freq[3]*p*i)))+(mux[4]*(sin(2*3.141592654*freq[4]*p*i))));
```

```
t=(t+5)*25.5;
```

```
if(t<0)
```

```
{
```

```
t=0;
```

```
}
```

```
lcd_clear();
```

```
lcd_gotoxy(0,0);
```

```
ftoa(t,dec,buff);
```

```
lcd_puts(buff);
```

```
lcd_gotoxy(0,1);
```

```
ftoa(p,dec,buff);
```

```
lcd_puts(buff);
```

```
delay_ms(10);
```

```
dac[i]=t;
```

```
t=0;
```

```
}
```

```
f=1/w;
```

```
if(f>1000)
```

```
{
```

```
f=f/1000;
```

```
lcd_clear();
```

```
lcd_gotoxy(0,1);
```

```
ftoa(f,dec,buff);
```

```
lcd_puts(buff);
```

```
lcd_gotoxy(0,0);
```

```
lcd_putsf("coefficient");
```

```
}
```

```
over=p*16000000;
```

```
over=65536-over;
```

```
tim1l=over & 0x00ff;
```

```
#asm("sei")
```

```
TIMSK=0x04;
```

```
ACSR=0x80;
```

```
SFIOR=0x00;
```




```

tim1h=(over & 0xff00)>>8;
TCNT1L=tim1l;
TCNT1H= tim1h;
TCCR1A=0x00;
TCCR1B=0X01;
while(1)
{
};
/*while(sw)
{
    for(i=0;i<50;i++)
    {
        PORTB=dac[i];

    }
    key=scan_key();
    if(key==15)
    {sw=0;
    }
}
*/
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("finish");
}
void inter_amp(void)
{
char key;
char sw=1;
float b=0;
float n=0;
float s=1;
char i=0;
while(i<5)
{
key=scan_key();
delay_ms(20);
if(key!=0xff)
{
    if(key==12)
    {
        sw=0;
    }
}
}
}

```



```

if(sw==1)
{
    if(key>=0 && key<=9)
    {
        b=key+b*10;
    }
}
if(sw==0)
{
    if(key>=0 && key<=9)
    {
        s=s*.1;
        n=s*key+n;
    }
}

if(key==13)
{
    b=0;
    n=0;
    s=1;
}
if(key==14)
{
    i=0;
    b=0;
    n=0;
    s=1;
}

if(key==15)
{
    mux[i]=b+n;
    i++;
    sw=1;
    s=1;
    n=0;
    b=0;
}
}
}
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("enter amp");
    
```



```

    lcd_gotoxy(0,1);
    ftoa(n+b,dec,buff);
    lcd_puts(buff);
}
}
void inter_w(void)
{
    unsigned int v;
    char key;
    char sw=1;
    float b=0;
    float n=0;
    float s=1;
    char i=0;
    while(i<5)
    {
        key=scan_key();
        delay_ms(20);
        if(key!=0xff)
        {
            if(key==12)
            {
                sw=0;
            }

            if(sw==1)
            {
                if(key>=0 && key<=9)
                {
                    b=key+b*10;
                }
            }

            if(sw==0)
            {
                if(key>=0 && key<=9)
                {
                    s=s*.1;
                    n=s*key+n;
                }
            }
            if(key==13)
            {

```



```

        b=0;
        n=0;
        s=1;
    }
    if(key==14)
    {
        i=0;
        b=0;
        n=0;
        s=1;
    }
    if(key==15)
    {
        freq[i]=b+n;
        time[i]=(1/b+n)*10000;
        i++;
        sw=1;
        s=1;
        n=0;
        b=0;
    }
}
    lcd_clear();
    lcd_gotoxy(0,0);
    lcd_putsf("enter w");
    lcd_gotoxy(0,1);
    ftoa(n+b,dec,buff);
    lcd_puts(buff);
}
}
float result_w(void)
{
    double x,y,r,temp,sw,k,u,v,p,w;

    char i=0;
    for(i=0;i<5;i++)
    {
        sw=1;
        x=time[0];
        y=time[i];
        p=x*y;
        if(x<y)
        {

```



```

temp=y; // Place your code here
y=x;
x=temp;
};
while(sw)
{
    r=x-(x/y*y);
    if(r==0)
    {
        k=(p/y);
        time[0]=k;
        sw=0;
    }
    else
    {
        x=y;
        y=r;
        sw=1;
    }
}
}
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("kmm");
lcd_gotoxy(0,1);
ftoa(time[0],dec,buff);
lcd_puts(buff);
delay_ms(50);
w=time[0];
w=w*.0001;
return w;
}
unsigned char scan_key(void)
{
    unsigned char i,data,num_key,temp,p;
    num_key=0xff;
    temp=0x70;
    for(i=0;i<4;i++)
    {
        PORTD=temp;
        delay_ms(10);
        data=PIND&0X0F;
    }
}

```

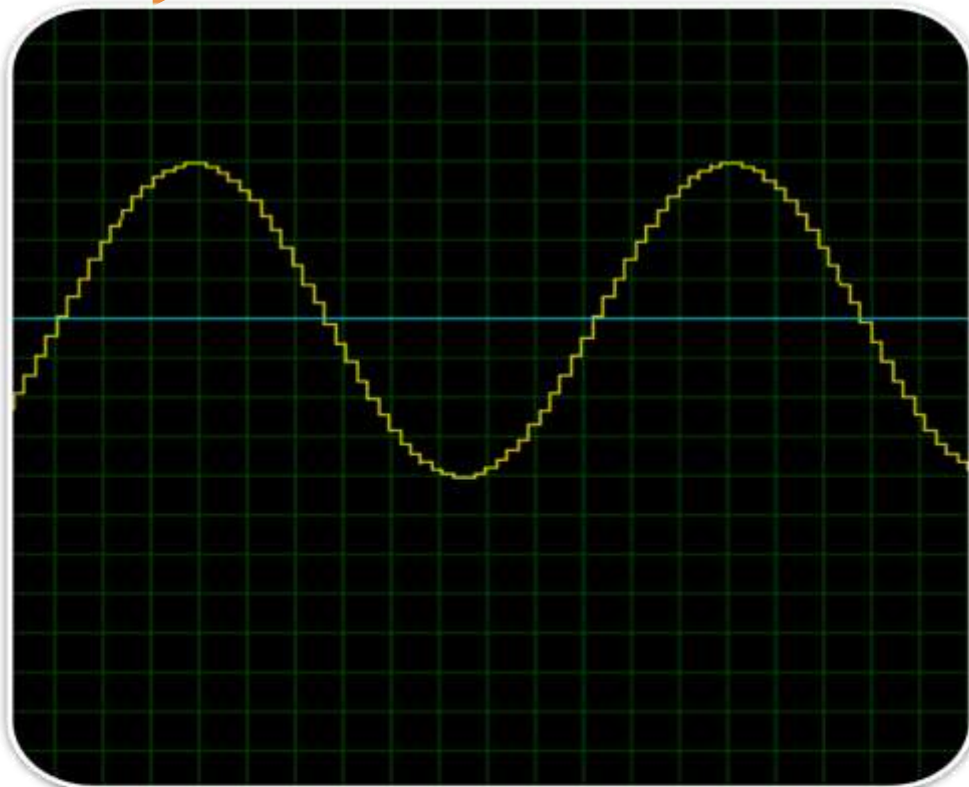


```

delay_ms(10);
data=PIND&0X0F;
if(data==0x07)
num_key=code[0][i];
if(data==0x0b)
num_key=code[1][i];
if(data==0x0d)
num_key=code[2][i];
if(data==0x0e)
num_key=code[3][i];
temp=((temp>>1)|0x80)&0xf0;
}
return num_key;
}
    
```

مثالهایی از این پروژه با دامنه وفر کانس مختلف که باکی بورد به آن دادیم :

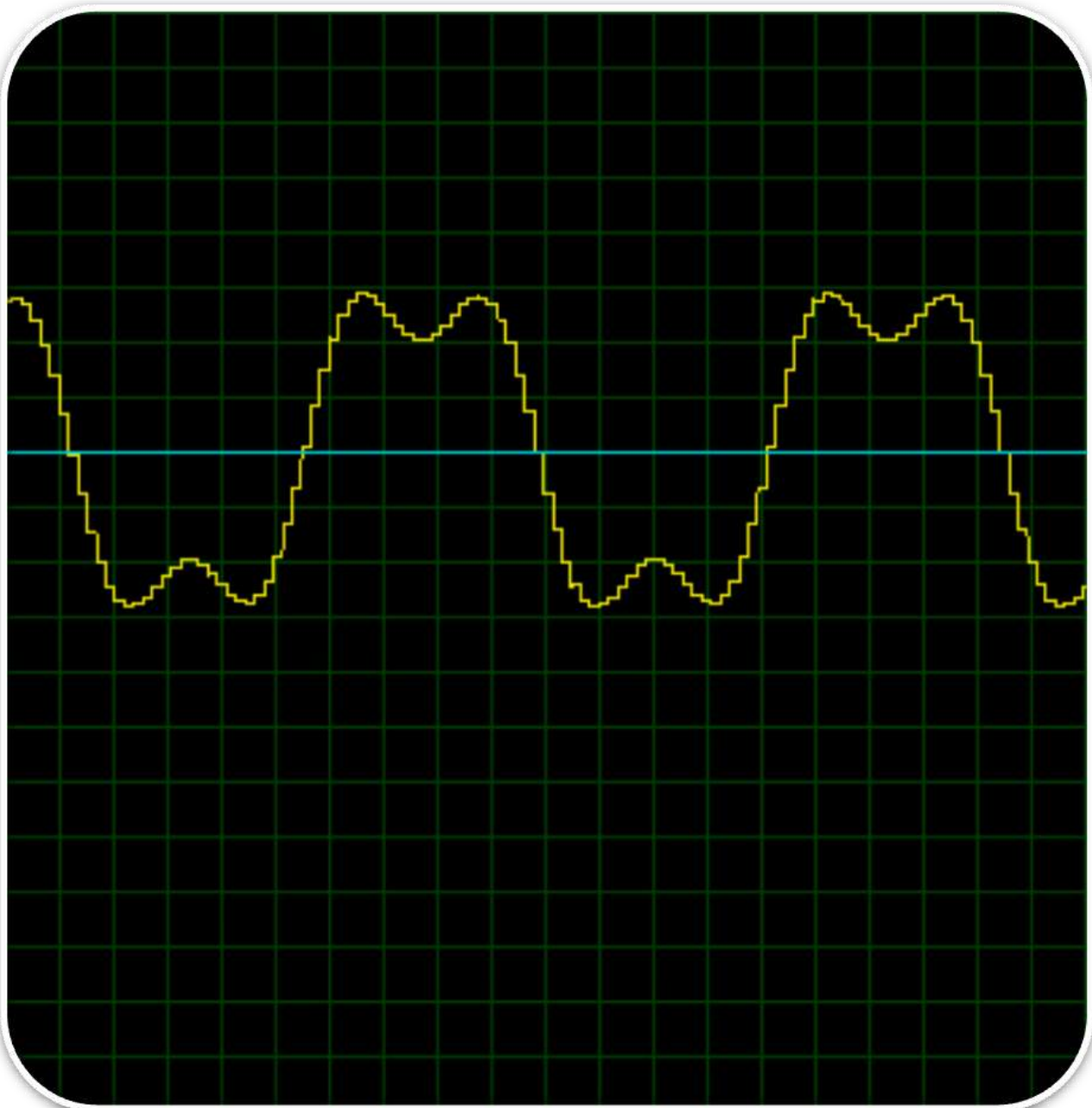
If amp1=1 → f1=1000
 If amp2=1 → f2=1000
 If amp3=1 → f3=1000
 If amp4=1 → f4=1000
 If amp5=1 → f5=1000





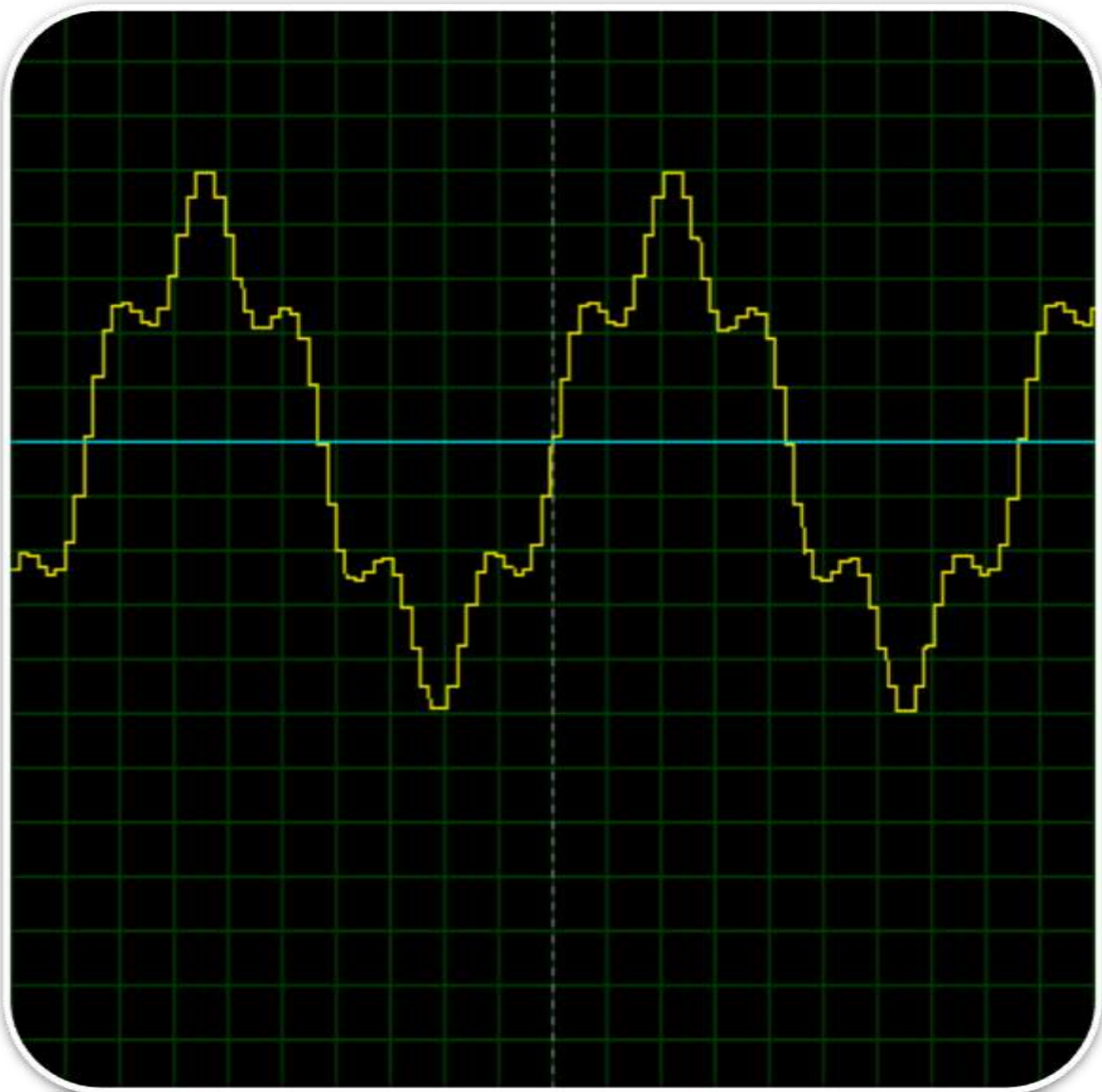
If amp1=1  f1=3000

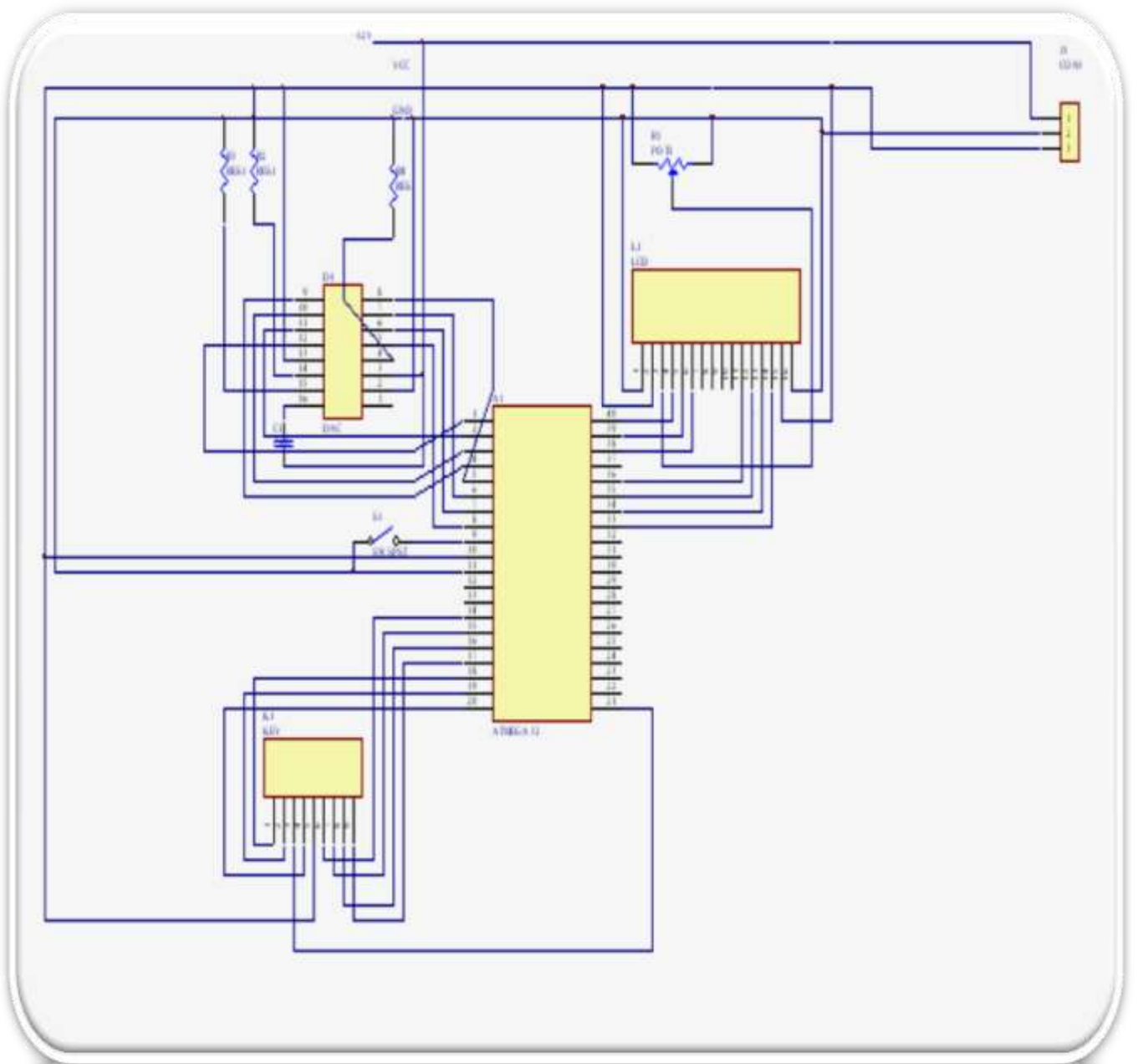
If amp2=3  f2=1000





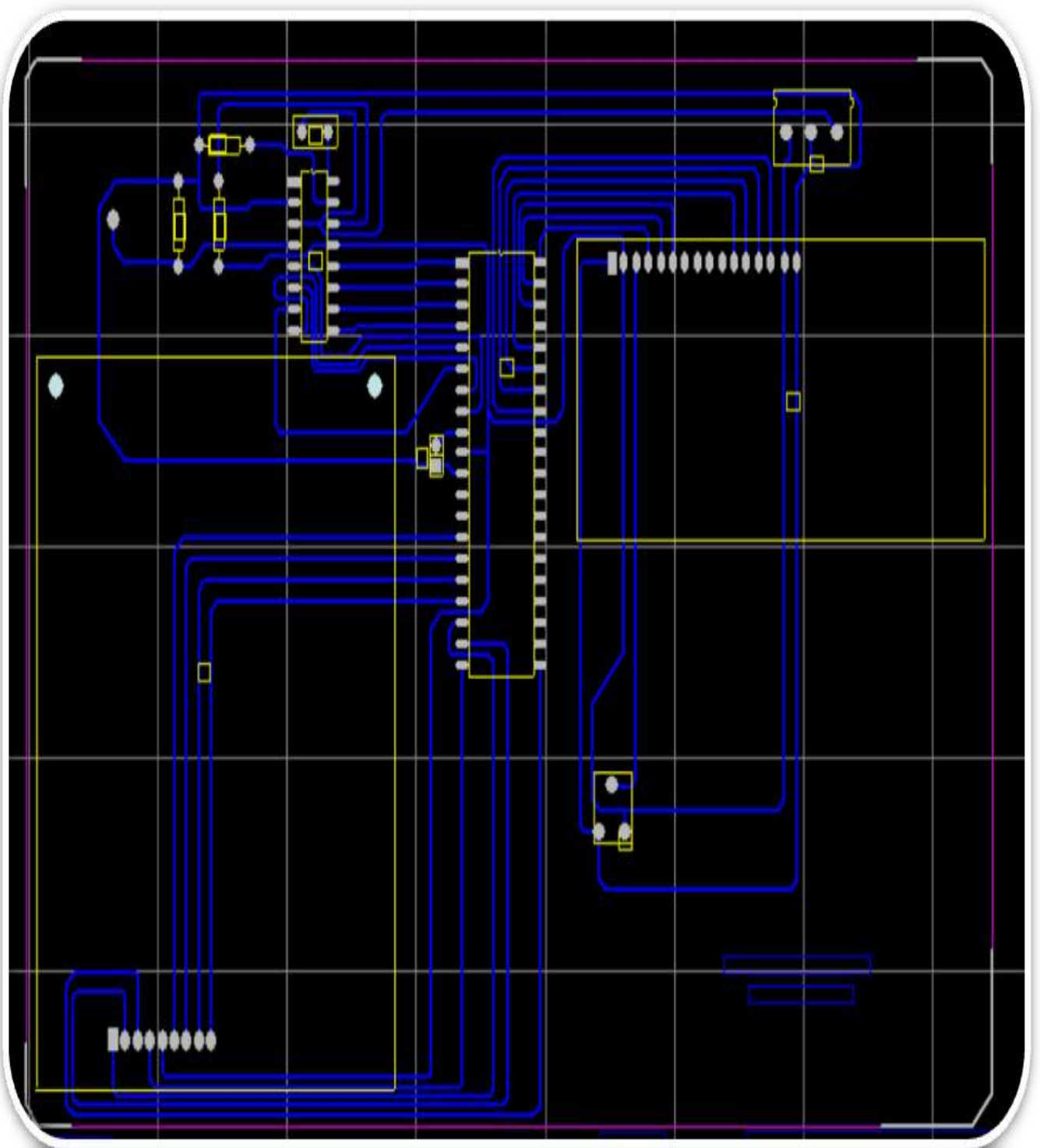
- If amp1=1 → f1=1000
- If amp2=1 → f2=1000
- If amp3=1 → f3=1000
- If amp4=1 → f4=5000
- If amp5=1 → f5=1000







PCB این پروژه :

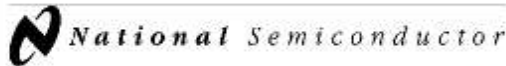




فصل چهارم

:Data sheet dac080

8



January 1995

DAC0808/DAC0807/DAC0806 8-Bit D/A Converters

General Description

The DAC0808 series is an 8-bit monolithic digital-to-analog converter (DAC) featuring a full scale output current settling time of 150 ns while dissipating only 33 mW with $\pm 5V$ supplies. No reference current (I_{REF}) trimming is required for most applications since the full scale output current is typically ± 1 LSB of $255 I_{REF}/256$. Relative accuracies of better than $\pm 0.19\%$ assure 8-bit monotonicity and linearity while zero level output current of less than $4 \mu A$ provides 8-bit zero accuracy for $I_{REF} > 2$ mA. The power supply currents of the DAC0808 series are independent of bit codes, and exhibits essentially constant device characteristics over the entire supply voltage range.

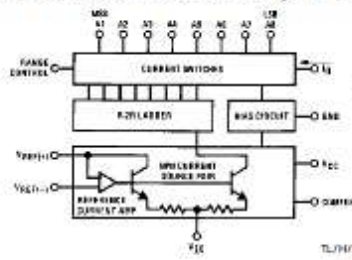
The DAC0808 will interface directly with popular TTL, DTL or CMOS logic levels, and is a direct replacement for the

MC1508/MC1408. For higher speed applications, see DAC0800 data sheet.

Features

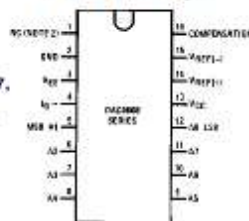
- Relative accuracy: $\pm 0.19\%$ error maximum (DAC0808)
- Full scale current match: ± 1 LSB typ
- 7 and 8-bit accuracy available (DAC0807, DAC0806)
- Fast settling time: 150 ns typ
- Noninverting digital inputs are TTL and CMOS compatible
- High speed multiplying input slew rate: 8 mA/ μs
- Power supply voltage range: $\pm 4.5V$ to $\pm 18V$
- Low power consumption: 33 mW @ $\pm 5V$

Block and Connection Diagrams



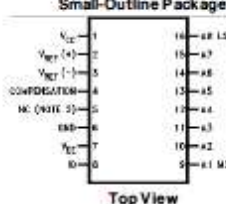
Order Number
DAC0808, DAC0807,
or DAC0806
See NS Package
Number J16A,
M16A or N16A

Dual-In-Line Package



TL/H/5687-2

Small-Outline Package



TL/H/5687-13

Ordering Information

ACCURACY	OPERATING TEMPERATURE RANGE	ORDER NUMBERS			
		J PACKAGE (J16A)*		N PACKAGE (N16A)*	
7-bit	$0^{\circ}C < T_A < +75^{\circ}C$	DAC0807LCJ	MC1408L7	DAC0808LCN	MC1408P8
6-bit	$0^{\circ}C < T_A < +75^{\circ}C$	DAC0806LCJ	MC1408L6	DAC0807LCN	MC1408P7
				DAC0806LCN	MC1408P6

*Note: Devices may be ordered by using either order number.

DAC0808/DAC0807/DAC0806 8-Bit D/A Converters



Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Power Supply Voltage	
V_{CC}	+18 V_{DC}
V_{EE}	-18 V_{DC}
Digital Input Voltage, V_5 - V_{12}	-10 V_{DC} to +18 V_{DC}
Applied Output Voltage, V_O	-11 V_{DC} to +18 V_{DC}
Reference Current, I_{14}	5 mA
Reference Amplifier Inputs, V_{14} , V_{15}	V_{CC} , V_{EE}
Power Dissipation (Note 3)	1000 mW
ESD Susceptibility (Note 4)	TBD

Storage Temperature Range	-65°C to +150°C
Lead Temp. (Soldering, 10 seconds)	
Dual-In-Line Package (Plastic)	260°C
Dual-In-Line Package (Ceramic)	300°C
Surface Mount Package	
Vapor Phase (60 seconds)	215°C
Infrared (15 seconds)	220°C

Operating Ratings

Temperature Range	$T_{MIN} \leq T_A \leq T_{MAX}$
DAC0808LC Series	$0 \leq T_A \leq +75^\circ\text{C}$

Electrical Characteristics

($V_{CC} = 5V$, $V_{EE} = -15V_{DC}$, $V_{REF}/R_{14} = 2mA$, DAC0808: $T_A = -55^\circ\text{C}$ to +125°C, DAC0808C, DAC0807C, DAC0806C, $T_A = 0^\circ\text{C}$ to +75°C, and all digital inputs at high logic level unless otherwise noted.)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
E_r	Relative Accuracy (Error Relative to Full Scale I_O) DAC0808LC (LM1408-8) DAC0807LC (LM1408-7), (Note 5) DAC0806LC (LM1408-6), (Note 5) Settling Time to Within $1/2$ LSB (Includes t_{PLH})	(Figure 4) $T_A = 25^\circ\text{C}$ (Note 6), (Figure 5)			± 0.19 ± 0.39 ± 0.78	% % % ns
t_{PLH} , t_{PHL}	Propagation Delay Time	$T_A = 25^\circ\text{C}$, (Figure 5)		30	100	ns
TC_{IQ}	Output Full Scale Current Drift			± 20		ppm/°C
MSB V_{IH} V_{IL}	Digital Input Logic Levels High Level, Logic "1" Low Level, Logic "0"	(Figure 3)	2		0.8	V_{DC} V_{DC}
MSB	Digital Input Current High Level Low Level	(Figure 3) $V_{IH} = 5V$ $V_{IL} = 0.8V$		0 -0.003	0.040 -0.8	mA mA
I_{IS}	Reference Input Bias Current	(Figure 3)		-1	-3	μA
	Output Current Range	(Figure 3) $V_{EE} = -5V$ $V_{EE} = -15V$, $T_A = 25^\circ\text{C}$	0 0	2.0 2.0	2.1 4.2	mA mA
I_O	Output Current	$V_{REF} = 2.000V$, $R_{14} = 1000\Omega$, (Figure 3)	1.9	1.99	2.1	mA
	Output Current, All Bits Low	(Figure 3)		0	4	μA
	Output Voltage Compliance (Note 2) $V_{EE} = -5V$, $I_{REF} = 1mA$ V_{EE} Below -10V	$E_r < 0.19\%$, $T_A = 25^\circ\text{C}$			-0.55, +0.4 -5.0, +0.4	V_{DC} V_{DC}



Electrical Characteristics (Continued)

($V_{CC} = 5V, V_{EE} = -15V, V_{REF}/R14 = 2mA$, DAC0808: $T_A = -55^{\circ}C$ to $+125^{\circ}C$, DAC0809C, DAC0807C, DAC0806C, $T_A = 0^{\circ}C$ to $+75^{\circ}C$, and all digital inputs at high logic level unless otherwise noted.)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
SRI_{REF}	Reference Current Slew Rate	(Figure 6)	4	8		$mA/\mu s$
	Output Current Power Supply Sensitivity	$-5V < V_{EE} < -16.5V$		0.05	2.7	$\mu A/V$
I_{CC} I_{EE}	Power Supply Current (All Bits Low)	(Figure 3)		2.3 -4.3	22 -13	mA mA
V_{CC} V_{EE}	Power Supply Voltage Range	$T_A = 25^{\circ}C$, (Figure 3)	4.5 -4.5	5.0 -15	5.5 -16.5	V_{CC} V_{EE}
	Power Dissipation All Bits Low	$V_{CC} = 5V, V_{EE} = -5V$		33	170	mW
		$V_{CC} = 5V, V_{EE} = -15V$		106	305	mW
	All Bits High	$V_{CC} = 15V, V_{EE} = -5V$		90		mW
		$V_{CC} = 15V, V_{EE} = -15V$		160		mW

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its specified operating conditions.

Note 2: Range control is not required.

Note 3: The maximum power dissipation must be derated at elevated temperatures and is dictated by T_{JMAX} , θ_{JA} , and the ambient temperature, T_A . The maximum allowable power dissipation at any temperature is $P_D = (T_{JMAX} - T_A)/\theta_{JA}$ or the number given in the Absolute Maximum Ratings, whichever is lower. For this device, $T_{JMAX} = 125^{\circ}C$, and the typical junction-to-ambient thermal resistance of the dual-in-line J package when the board mounted is $100^{\circ}C/W$. For the dual-in-line N package, this number increases to $175^{\circ}C/W$ and for the small outline M package this number is $100^{\circ}C/W$.

Note 4: Human body model, $100 pF$ discharged through a $1.5 k\Omega$ resistor.

Note 5: All current switches are tested to guarantee at least 50% of rated current.

Note 6: All bits switched.

Note 7: Pin-out numbers for the DAC080X represent the dual-in-line package. The small outline package pinout differs from the dual-in-line package.

Typical Application

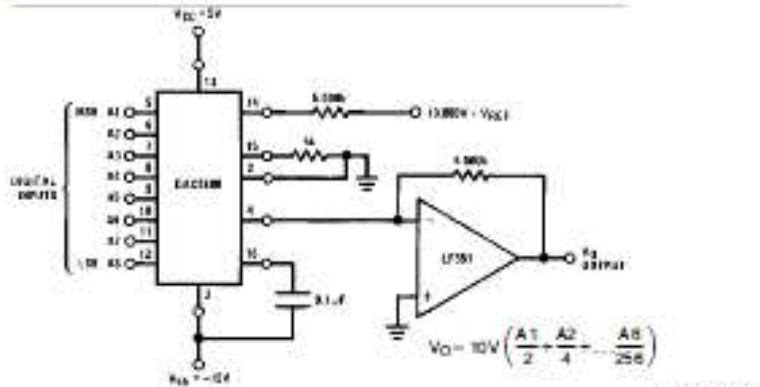


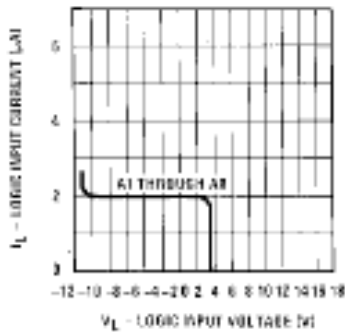
FIGURE 1. +10V Output Digital to Analog Converter (Note 7)



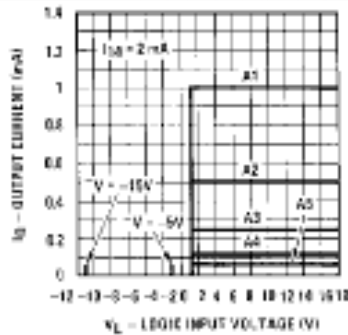
Typical Performance Characteristics

$V_{CC} = 5V$, $V_{EE} = -15V$, $T_A = 25^\circ C$, unless otherwise noted

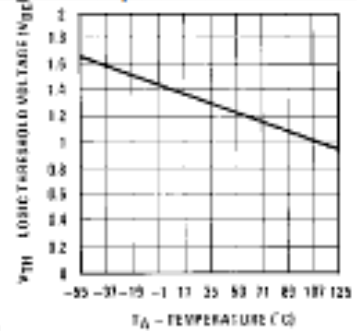
Logic Input Current vs Input Voltage



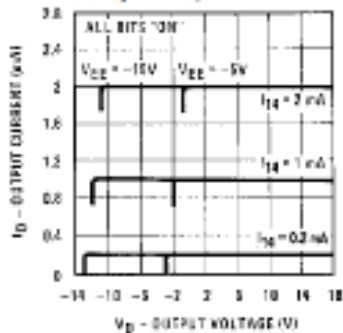
Bit Transfer Characteristics



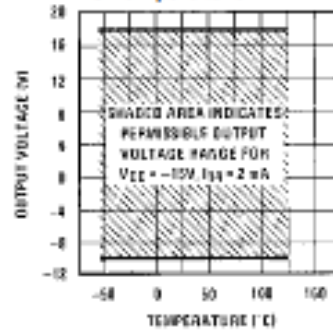
Logic Threshold Voltage vs Temperature



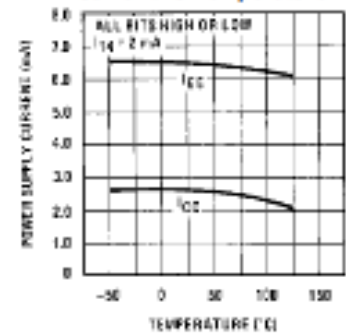
Output Current vs Output Voltage (Output Voltage Compliance)



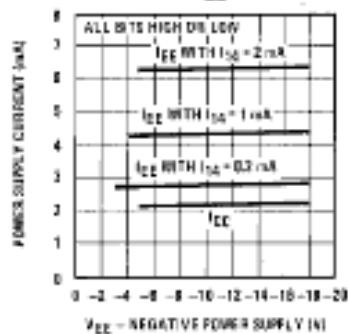
Output Voltage Compliance vs Temperature



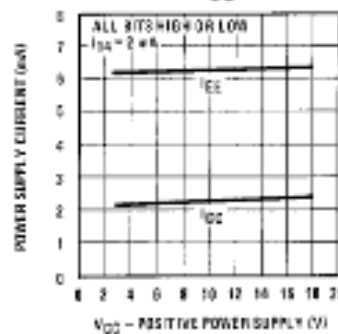
Typical Power Supply Current vs Temperature



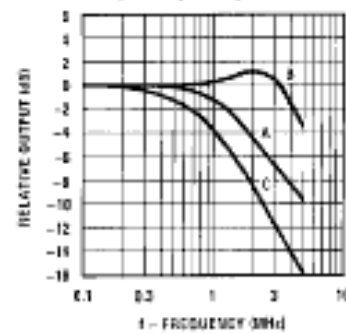
Typical Power Supply Current vs V_{EE}



Typical Power Supply Current vs V_{CC}



Reference Input Frequency Response



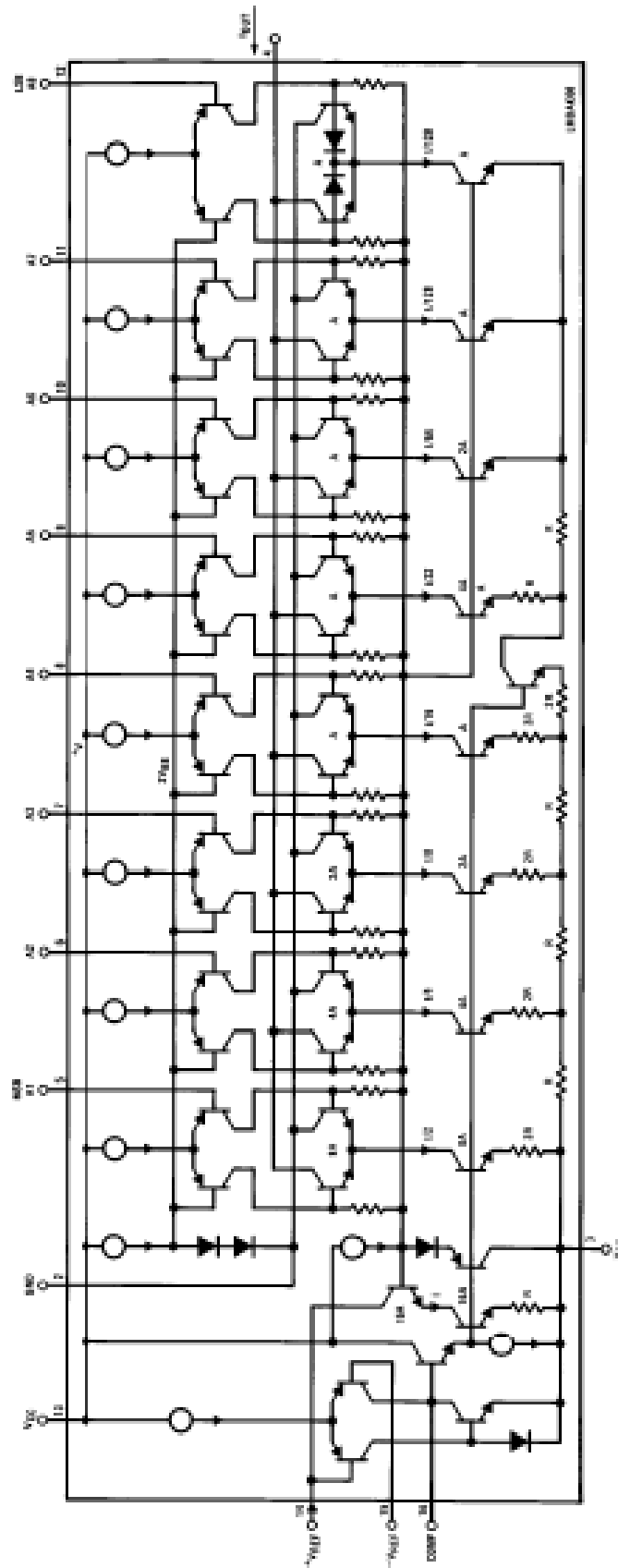
TL1115687-5

Unless otherwise specified: $R_{14} = R_{15} = 1 k\Omega$, $C = 15 pF$, pin 16 to V_{EE} ; $R_L = 50 \Omega$, pin 4 to ground.

Curve A: Large Signal Bandwidth Method of Figure 7, $V_{REF} = 2 V_{p-p}$ offset 1 V above ground.

Curve B: Small Signal Bandwidth Method of Figure 7, $R_L = 250 \Omega$, $V_{REF} = 50 mV_{p-p}$ offset 200 mV above ground.

Curve C: Large and Small Signal Bandwidth Method of Figure 9 (no op amp, $R_L = 50 \Omega$), $R_S = 50 \Omega$, $V_{REF} = 2V$, $V_S = 100 mV_{p-p}$ centered at 0V.

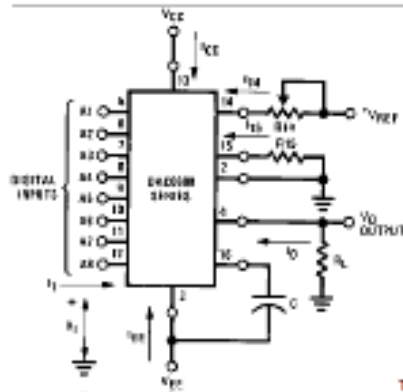


TLV4108B -4

FIGURE 2. Equivalent Circuit of the DAC0808 Series (Note 7)



Test Circuits



V_I and I_I apply to inputs A1 –A8.

The resistor tied to pin 15 is to temperature compensate the bias current and may not be necessary for all applications.

$$I_O = K \left(\frac{A_1}{2} + \frac{A_2}{4} + \frac{A_3}{8} + \frac{A_4}{16} + \frac{A_5}{32} + \frac{A_6}{64} + \frac{A_7}{128} + \frac{A_8}{256} \right)$$

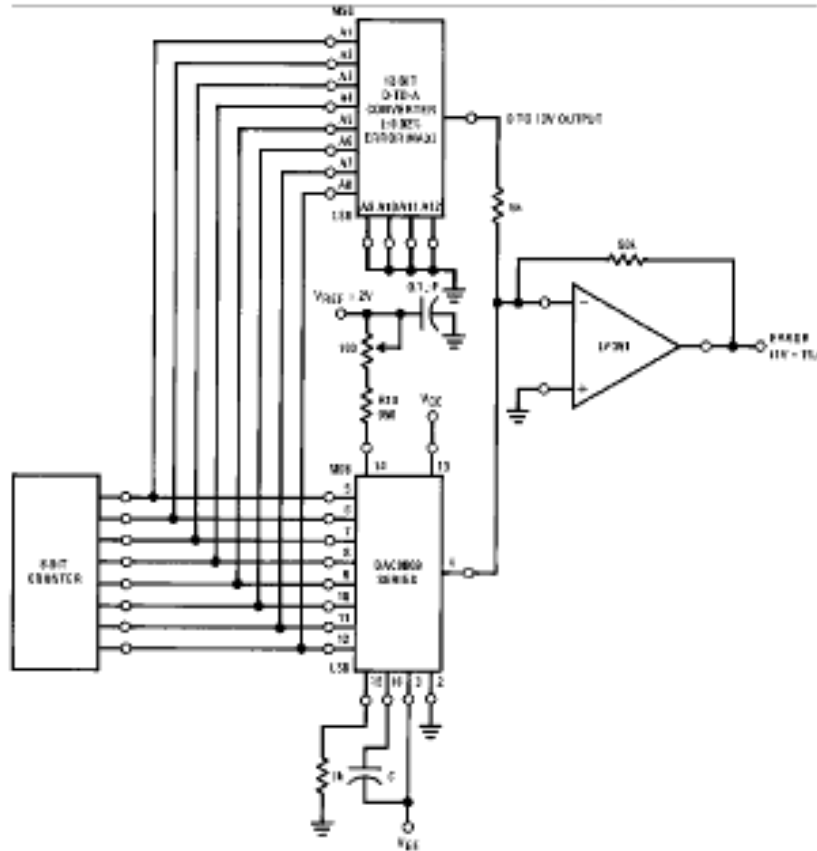
where $K = \frac{V_{REF}}{R_{14}}$

and $A_N = "1"$ if A_N is at high level

$A_N = "0"$ if A_N is at low level

TL/11/5687-6

FIGURE 3. Notation Definitions Test Circuit (Note 7)



TL/11/5687-7

FIGURE 4. Relative Accuracy Test Circuit (Note 7)

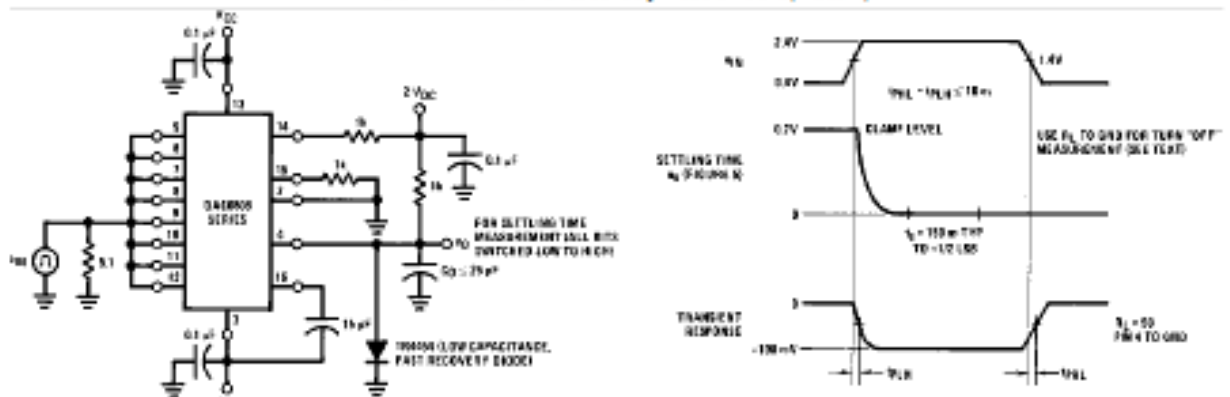
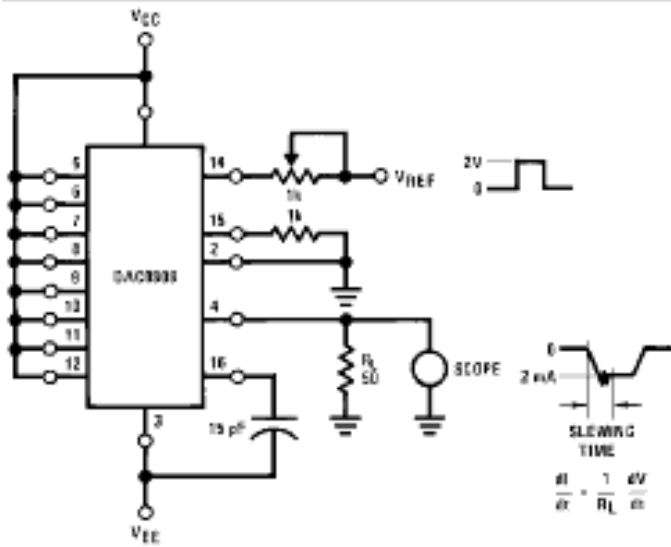


FIGURE 5. Transient Response and Settling Time (Note 7)

TL/11/5687-8

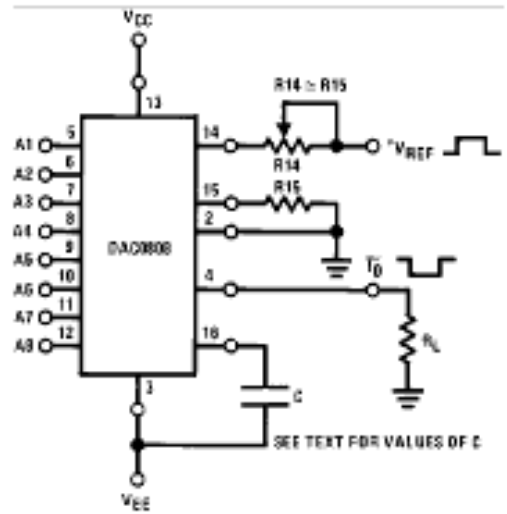


Test Circuits (Continued)



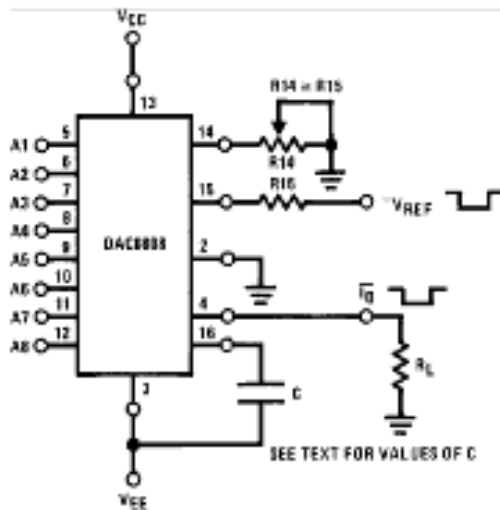
TL/H/5687-9

FIGURE 6. Reference Current Slew Rate Measurement (Note 7)



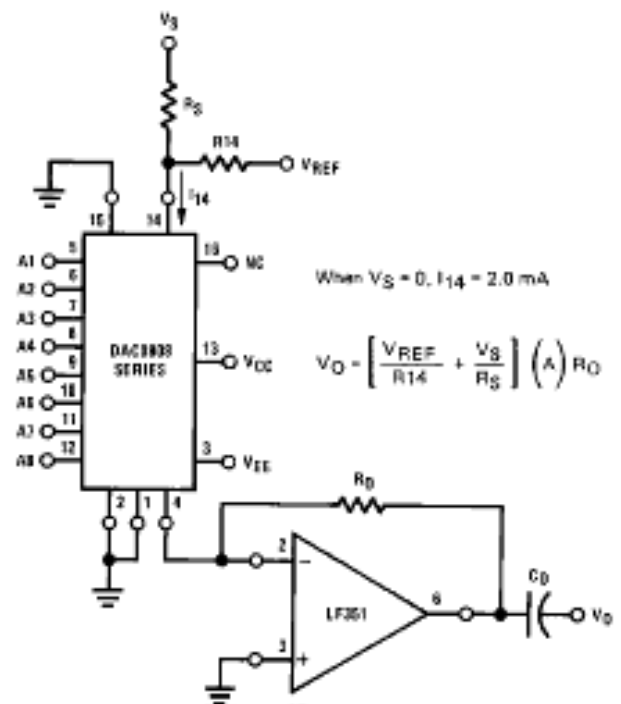
TL/H/5687-10

FIGURE 7. Positive V_{REF} (Note 7)



TL/H/5687-11

FIGURE 8. Negative V_{REF} (Note 7)



TL/H/5687-12

FIGURE 9. Programmable Gain Amplifier or Digital Attenuator Circuit (Note 7)

Application Hints

REFERENCE AMPLIFIER DRIVE AND COMPENSATION

The reference amplifier provides a voltage at pin 14 for converting the reference voltage to a current, and a turn-around circuit or current mirror for feeding the ladder. The reference amplifier input current, I_{14} , must always flow into pin 14, regardless of the set-up method or reference voltage polarity. Connections for a positive voltage are shown in Figure 7. The reference voltage source supplies the full current I_{14} . For bipolar reference signals, as in the multiplying mode,

R_{15} can be tied to a negative voltage corresponding to the minimum input level. It is possible to eliminate R_{15} with only a small sacrifice in accuracy and temperature drift.

The compensation capacitor value must be increased with increases in R_{14} to maintain proper phase margin; for R_{14} values of 1, 2.5 and 5 k Ω , minimum capacitor values are 15, 37 and 75 pF. The capacitor may be tied to either V_{EE} or ground, but using V_{EE} increases negative supply rejection.



Application Hints (Continued)

A negative reference voltage may be used if R14 is grounded and the reference voltage is applied to R15 as shown in Figure 8. A high input impedance is the main advantage of this method. Compensation involves a capacitor to V_{EE} on pin 16, using the values of the previous paragraph. The negative reference voltage must be at least 4V above the V_{EE} supply. Bipolar input signals may be handled by connecting R14 to a positive reference voltage equal to the peak positive input level at pin 15.

When a DC reference voltage is used, capacitive bypass to ground is recommended. The 5V logic supply is not recommended as a reference voltage. If a well regulated 5V supply which drives logic is to be used as the reference, R14 should be decoupled by connecting it to 5V through another resistor and bypassing the junction of the 2 resistors with 0.1 μ F to ground. For reference voltages greater than 5V, a clamp diode is recommended between pin 14 and ground.

If pin 14 is driven by a high impedance such as a transistor current source, none of the above compensation methods apply and the amplifier must be heavily compensated, decreasing the overall bandwidth.

OUTPUT VOLTAGE RANGE

The voltage on pin 4 is restricted to a range of -0.55 to 0.4 V when $V_{EE} = -5$ V due to the current switching methods employed in the DAC0808.

The negative output voltage compliance of the DAC0808 is extended to -5 V where the negative supply voltage is more negative than -10 V. Using a full-scale current of 1.992 mA and load resistor of 2.5 k Ω between pin 4 and ground will yield a voltage output of 256 levels between 0 and -4.980 V. Floating pin 1 does not affect the converter speed or power dissipation. However, the value of the load resistor determines the switching time due to increased voltage swing. Values of R_L up to 500 Ω do not significantly affect performance, but a 2.5 k Ω load increases worst-case settling time to 1.2 μ s (when all bits are switched ON). Refer to the subsequent text section on Settling Time for more details on output loading.

OUTPUT CURRENT RANGE

The output current maximum rating of 4.2 mA may be used only for negative supply voltages more negative than -8 V, due to the increased voltage drop across the resistors in the reference current amplifier.

ACCURACY

Absolute accuracy is the measure of each output current level with respect to its intended value, and is dependent upon relative accuracy and full-scale current drift. Relative accuracy is the measure of each output current level as a fraction of the full-scale current. The relative accuracy of the DAC0808 is essentially constant with temperature due to

the excellent temperature tracking of the monolithic resistor ladder. The reference current may drift with temperature, causing a change in the absolute accuracy of output current. However, the DAC0808 has a very low full-scale current drift with temperature.

The DAC0808 series is guaranteed accurate to within $\pm 1/2$ LSB at a full-scale output current of 1.992 mA. This corresponds to a reference amplifier output current drive to the ladder network of 2 mA, with the loss of 1 LSB (8 μ A) which is the ladder remainder shunted to ground. The input current to pin 14 has a guaranteed value of between 1.9 and 2.1 mA, allowing some mismatch in the NPN current source pair. The accuracy test circuit is shown in Figure 4. The 12-bit converter is calibrated for a full-scale output current of 1.992 mA. This is an optional step since the DAC0808 accuracy is essentially the same between 1.5 and 2.5 mA. Then the DAC0808 circuits' full-scale current is trimmed to the same value with R14 so that a zero value appears at the error amplifier output. The counter is activated and the error band may be displayed on an oscilloscope, detected by comparators, or stored in a peak detector.

Two 8-bit D-to-A converters may not be used to construct a 16-bit accuracy D-to-A converter. 16-bit accuracy implies a total error of $\pm 1/2$ of one part in 65,536 or $\pm 0.00076\%$, which is much more accurate than the $\pm 0.019\%$ specification provided by the DAC0808.

MULTIPLYING ACCURACY

The DAC0808 may be used in the multiplying mode with 8-bit accuracy when the reference current is varied over a range of 256:1. If the reference current in the multiplying mode ranges from 16 μ A to 4 mA, the additional error contributions are less than 1.6 μ A. This is well within 8-bit accuracy when referred to full-scale.

A monotonic converter is one which supplies an increase in current for each increment in the binary word. Typically, the DAC0808 is monotonic for all values of reference current above 0.5 mA. The recommended range for operation with a DC reference current is 0.5 to 4 mA.

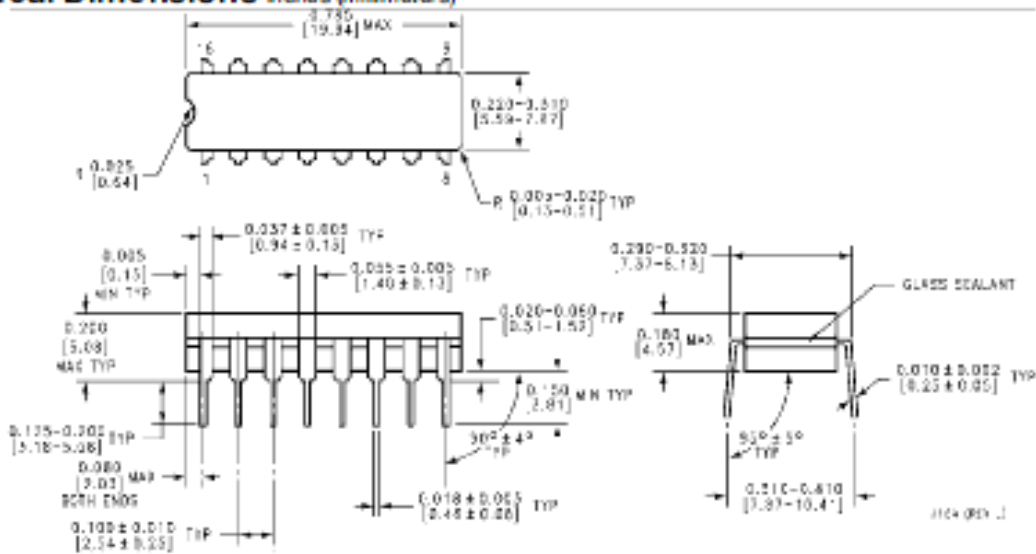
SETTLING TIME

The worst-case switching condition occurs when all bits are switched ON, which corresponds to a low-to-high transition for all bits. This time is typically 150 ns for settling to within $\pm 1/2$ LSB, for 8-bit accuracy, and 100 ns to $1/2$ LSB for 7 and 6-bit accuracy. The turn OFF is typically under 100 ns. These times apply when $R_L < 500\Omega$ and $C_O < 25$ pF.

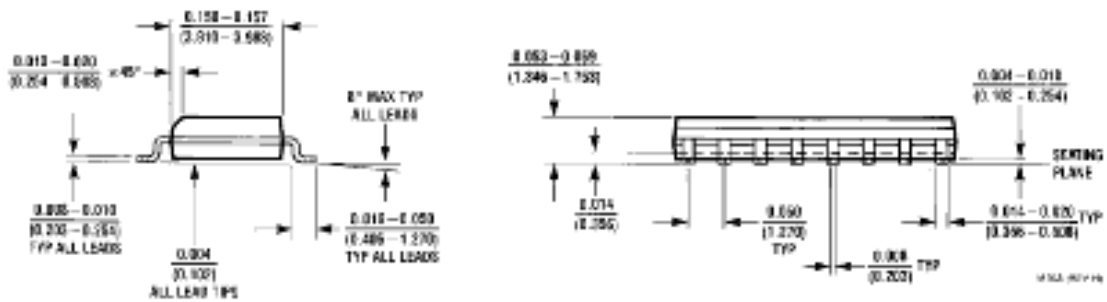
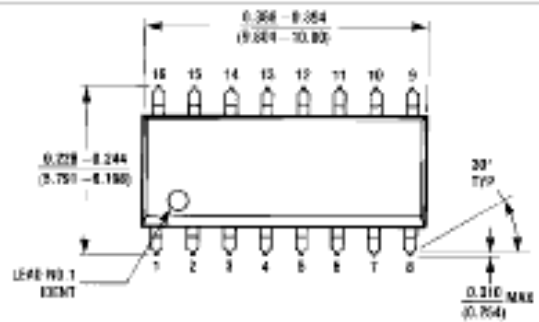
Extra care must be taken in board layout since this is usually the dominant factor in satisfactory test results when measuring settling time. Short leads, 100 μ F supply bypassing for low frequencies, and minimum scope lead length are all mandatory.



Physical Dimensions inches (millimeters)



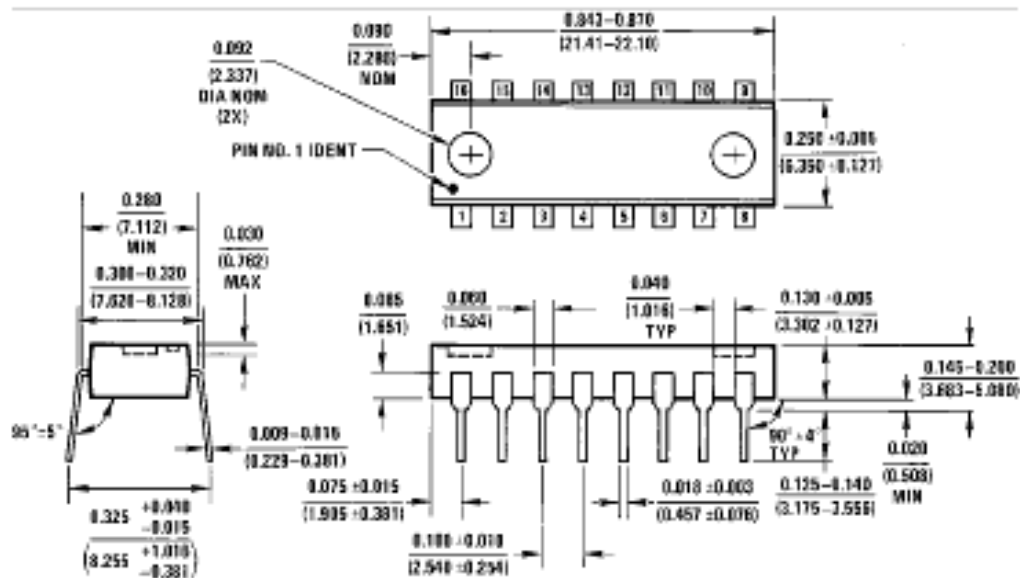
Dual In-Line Package
Order Number DAC0807 or DAC0806
NS Package Number J 16A



Small Outline Package
Order Number DAC0806LCM, DAC0807LCM or DAC0808LCM
NS Package Number M 16A



Physical Dimensions inches (millimeters) (Continued)



N16A-02V D1

Dual-In-Line Package
Order Number DAC0808, DAC0807 or DAC0806
NS Package Number N16A

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

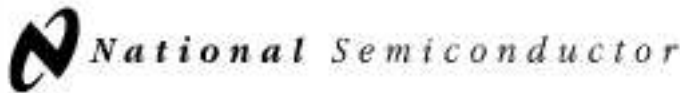


National Semiconductor Corporation
1111 West Bend Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 85
Email: crjwpa@sem2.nsc.com
Deutsch: Tel: (+49) 0-180-530 85 85
English: Tel: (+49) 0-180-532 78 32
Français: Tel: (+49) 0-180-532 93 58
Italiano: Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Street Block,
Crown Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1800
Fax: (852) 2735-9980

National Semiconductor Japan Ltd.
Tel: 81-4-43-299-2109
Fax: 81-4-43-299-2408



December 1995

LF351 Wide Bandwidth JFET Input Operational Amplifier

General Description

The LF351 is a low cost high speed JFET input operational amplifier with an internally trimmed input offset voltage (BI-FET II™ technology). The device requires a low supply current and yet maintains a large gain bandwidth product and a fast slew rate. In addition, well matched high voltage JFET input devices provide very low input bias and offset currents. The LF351 is pin compatible with the standard LM741 and uses the same offset voltage adjustment circuitry. This feature allows designers to immediately upgrade the overall performance of existing LM741 designs.

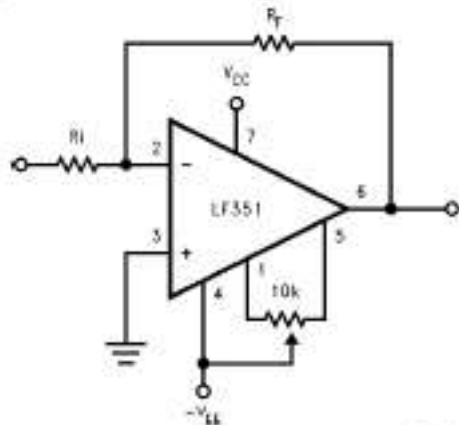
The LF351 may be used in applications such as high speed integrators, fast D/A converters, sample-and-hold circuits and many other circuits requiring low input offset voltage, low input bias current, high input impedance, high slew rate and wide bandwidth. The device has low noise and offset voltage drift, but for applications where these requirements are critical, the LF356 is recommended. If maximum supply

current is important, however, the LF351 is the better choice.

Features

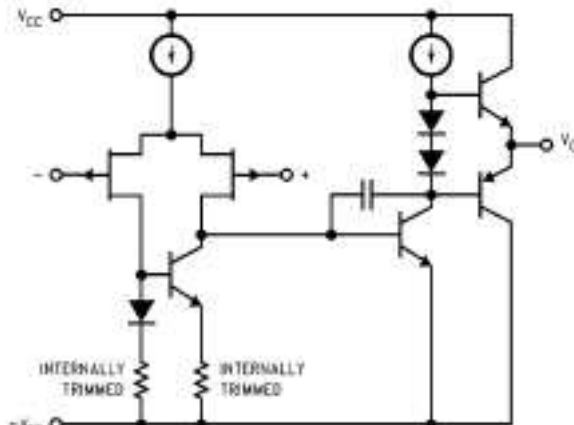
- Internally trimmed offset voltage 10 mV
- Low input bias current 50 pA
- Low input noise voltage 25 nV/√Hz
- Low input noise current 0.01 pA/√Hz
- Wide gain bandwidth 4 MHz
- High slew rate 13 V/μs
- Low supply current 1.8 mA
- High input impedance 10¹²Ω
- Low total harmonic distortion $A_V = 10$, $R_L = 10k$, $V_O = 20$ Vp-p, BW = 20 Hz-20 kHz < 0.02%
- Low 1/f noise corner 50 Hz
- Fast settling time to 0.01% 2 μs

Typical Connection



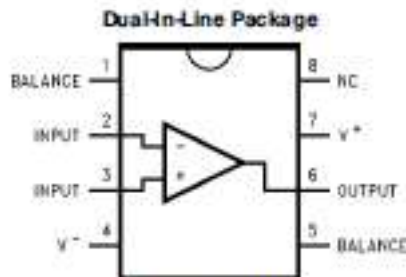
TL/90564B-11

Simplified Schematic



TL/90564B-12

Connection Diagrams



TL/90564B-13

Order Number LF351M or LF351N
See NS Package Number M08A or N08E



Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	±18V
Power Dissipation (Notes 1 and 6)	670 mW
Operating Temperature Range	0°C to +70°C
T _{J(MAX)}	115°C
Differential Input Voltage	±30V
Input Voltage Range (Note 2)	±15V
Output Short Circuit Duration	Continuous
Storage Temperature Range	-65°C to +150°C
Lead Temp. (Soldering, 10 sec.)	
Metal Can	300°C
DIP	260°C

θ _{JA}	N Package	120°C/W
	M Package	TBD
Soldering Information		
Dual-In-Line Package		
	Soldering (10 sec.)	260°C
Small Outline Package		
	Vapor Phase (60 sec.)	215°C
	Infrared (15 sec.)	220°C

See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" for other methods of soldering surface mount devices.

ESD rating to be determined.

DC Electrical Characteristics (Note 3)

Symbol	Parameter	Conditions	LF351			Units
			Min	Typ	Max	
V _{OS}	Input Offset Voltage	R _S = 10 kΩ, T _A = 25°C Over Temperature		5	10 13	mV mV
ΔV _{OS} /ΔT	Average TC of Input Offset Voltage	R _S = 10 kΩ		10		μV/°C
I _{OS}	Input Offset Current	T _J = 25°C, (Notes 3, 4) T _J ≤ 70°C		25	100 4	pA nA
I _B	Input Bias Current	T _J = 25°C, (Notes 3, 4) T _J ≤ ±70°C		50	200 8	pA nA
R _{IN}	Input Resistance	T _J = 25°C		10 ¹²		Ω
A _{VOL}	Large Signal Voltage Gain	V _S = ±15V, T _A = 25°C V _O = ±10V, R _L = 2 kΩ Over Temperature	25	100		V/mV V/mV
V _O	Output Voltage Swing	V _S = ±15V, R _L = 10 kΩ	±12	±13.5		V
V _{CM}	Input Common-Mode Voltage Range	V _S = ±15V	±11	+15 -12		V V
CMRR	Common-Mode Rejection Ratio	R _S ≤ 10 kΩ	70	100		dB
PSRR	Supply Voltage Rejection Ratio	(Note 5)	70	100		dB
I _S	Supply Current			1.8	3.4	mA



AC Electrical Characteristics (Note 3)

Symbol	Parameter	Conditions	LF351			Units
			Min	Typ	Max	
SR	Slow Rate	$V_S = \pm 15V, T_A = 25^\circ C$		13		V/ μs
GBW	Gain Bandwidth Product	$V_S = \pm 15V, T_A = 25^\circ C$		4		MHz
e_n	Equivalent Input Noise Voltage	$T_A = 25^\circ C, R_S = 100\Omega, f = 1000\text{ Hz}$		25		nV/ \sqrt{Hz}
i_n	Equivalent Input Noise Current	$T_J = 25^\circ C, f = 1000\text{ Hz}$		0.01		pA/ \sqrt{Hz}

Note 1: For operating at elevated temperature, the device must be derated based on the thermal resistance, θ_{JA} .

Note 2: Unless otherwise specified the absolute maximum negative input voltage is equal to the negative power supply voltage.

Note 3: These specifications apply for $V_S = \pm 15V$ and $0^\circ C < T_A < +70^\circ C$. V_{OS} , i_n and i_{IS} are measured at $V_{OU} = 0$.

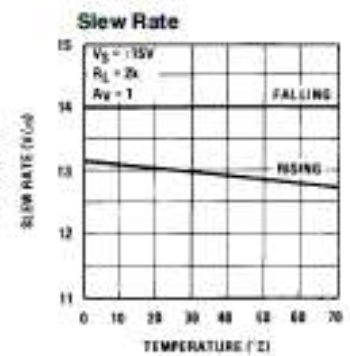
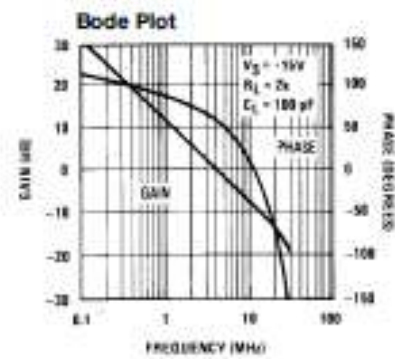
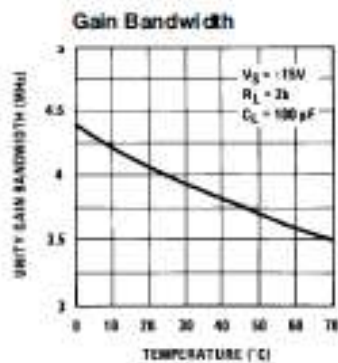
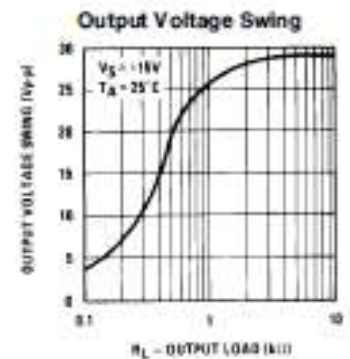
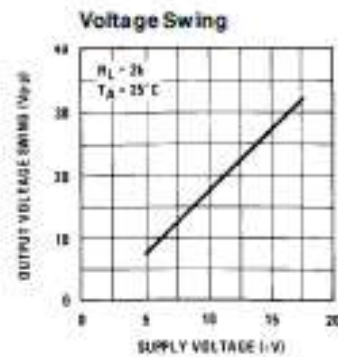
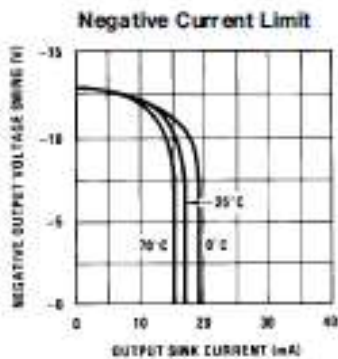
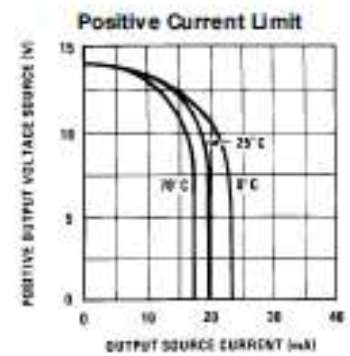
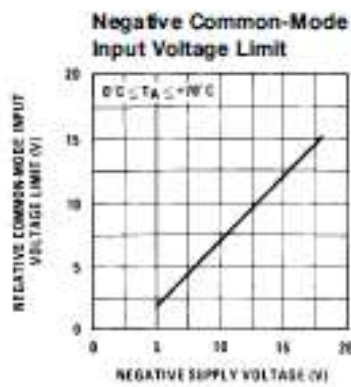
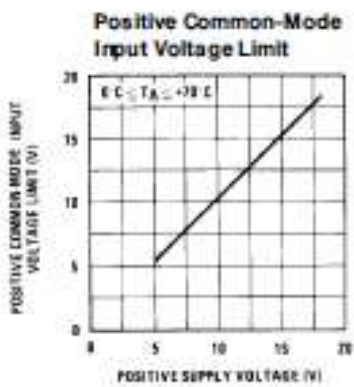
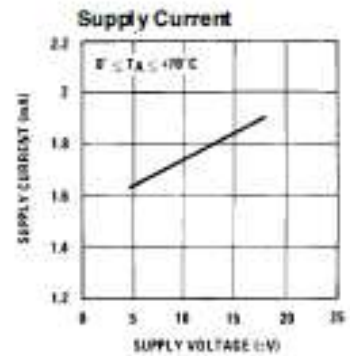
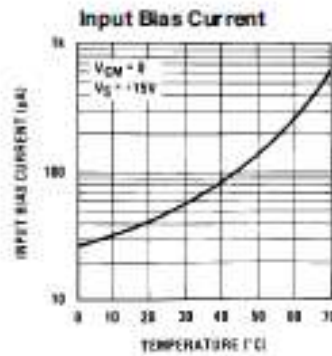
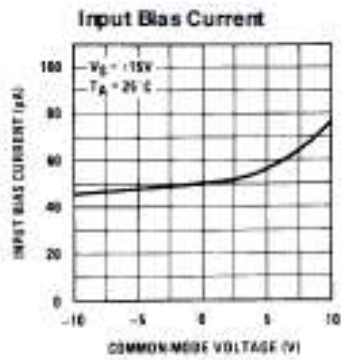
Note 4: The input bias currents are junction leakage currents which approximately double for every $10^\circ C$ increase in the junction temperature, T_J . Due to the limited production test time, the input bias currents measured are correlated to junction temperature. In normal operation the junction temperature rises above the ambient temperature as a result of internal power dissipation, P_D . $T_J = T_A + \theta_{JA} P_D$ where θ_{JA} is the thermal resistance from junction to ambient. Use of a heat sink is recommended if input bias current is to be kept to a minimum.

Note 5: Supply voltage rejection ratio is measured for both supply magnitudes increasing or decreasing simultaneously in accordance with common practice. From $\pm 15V$ to $\pm 5V$.

Note 6: Max. Power Dissipation is defined by the package characteristics. Operating the part near the Max. Power Dissipation may cause the part to operate outside guaranteed limits.



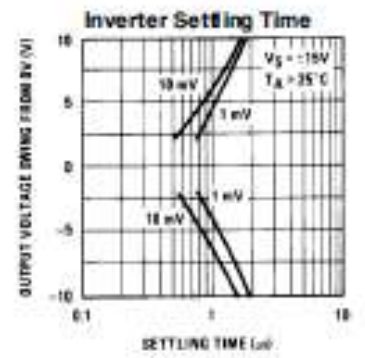
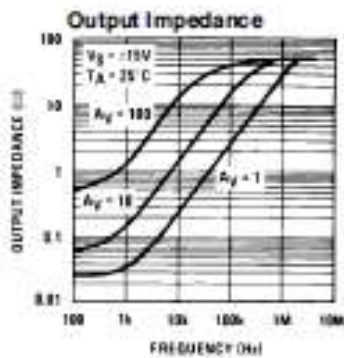
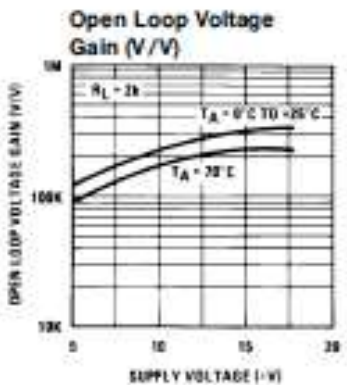
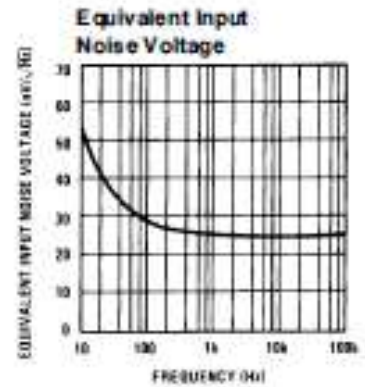
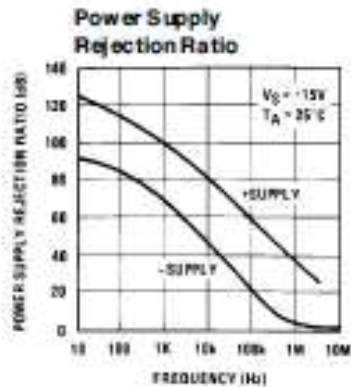
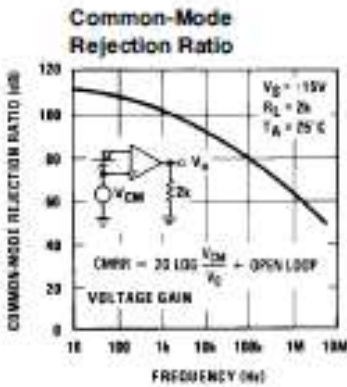
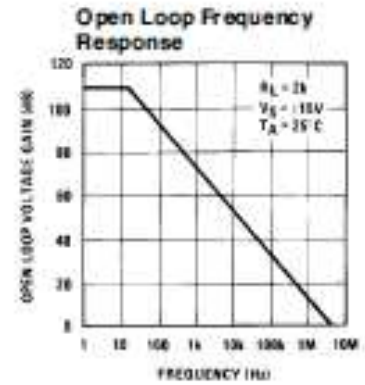
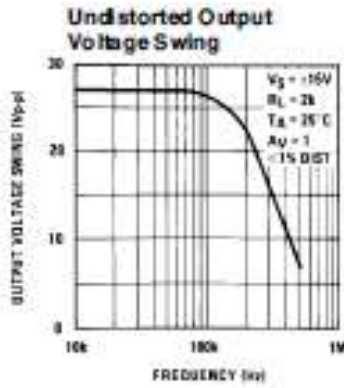
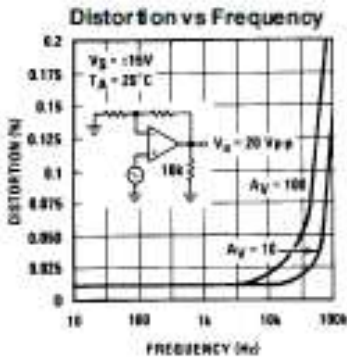
Typical Performance Characteristics



TL/HU/5640-2



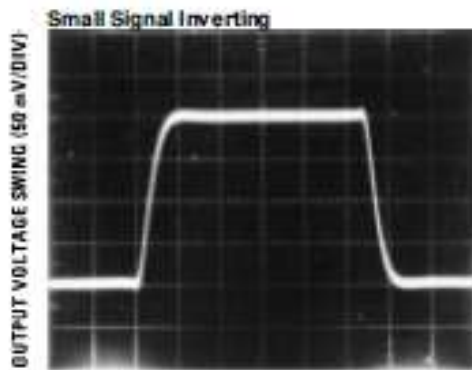
Typical Performance Characteristics (Continued)



TL79568-3

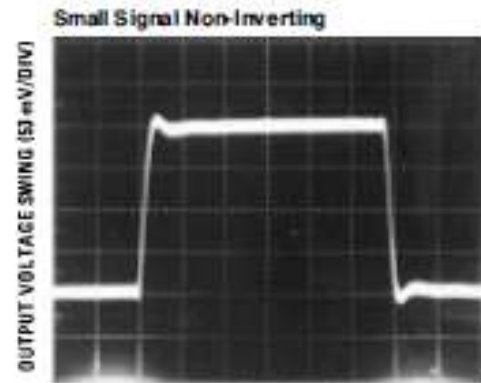


Pulse Response



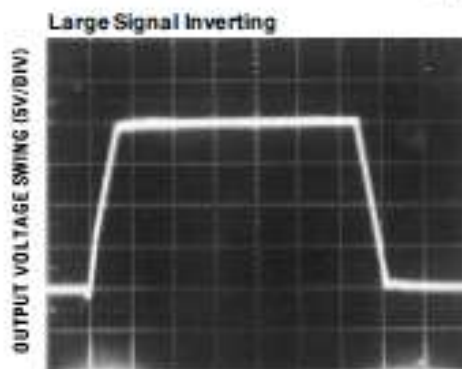
TIME (0.2 μs/DIV)

TL/H/5648-4



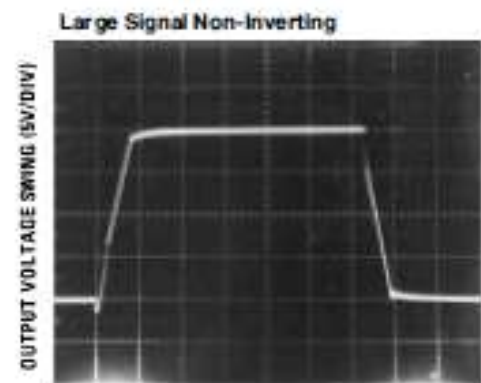
TIME (0.2 μs/DIV)

TL/H/5648-5



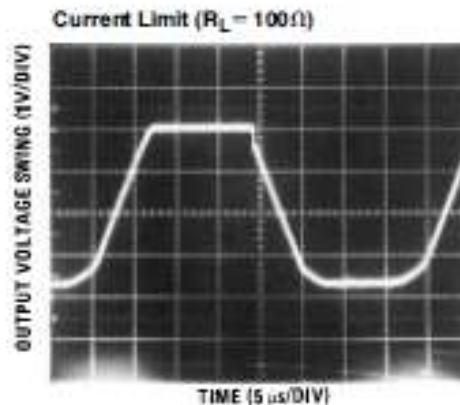
TIME (0.2 μs/DIV)

TL/H/5648-6



TIME (0.2 μs/DIV)

TL/H/5648-7



TIME (5 μs/DIV)

TL/H/5648-8

Application Hints

The LF351 is an op amp with an internally trimmed input offset voltage and JFET input devices (BJFET II™). These JFETs have large reverse breakdown voltages from gate to source and drain eliminating the need for clamps across the inputs. Therefore, large differential input voltages can easily be accommodated without a large increase in input current. The maximum differential input voltage is independent of the supply voltages. However, neither of the input voltages should be allowed to exceed the negative supply as this will

cause large currents to flow which can result in a destroyed unit.

Exceeding the negative common-mode limit on either input will force the output to a high state, potentially causing a reversal of phase to the output.

Exceeding the negative common-mode limit on both inputs will force the amplifier output to a high state. In neither case does a latch occur since raising the input back within the



Application Hints (Continued)

common-mode range again puts the input stage and thus the amplifier in a normal operating mode.

Exceeding the positive common-mode limit on a single input will not change the phase of the output; however, if both inputs exceed the limit, the output of the amplifier will be forced to a high state.

The amplifier will operate with a common-mode input voltage equal to the positive supply; however, the gain bandwidth and slew rate may be decreased in this condition. When the negative common-mode voltage swings to within 3V of the negative supply, an increase in input offset voltage may occur.

The LF351 is biased by a zener reference which allows normal circuit operation on $\pm 4V$ power supplies. Supply voltages less than these may result in lower gain bandwidth and slew rate.

The LF351 will drive a 2 k Ω load resistance to $\pm 10V$ over the full temperature range of 0°C to +70°C. If the amplifier is forced to drive heavier load currents, however, an increase in input offset voltage may occur on the negative voltage swing and finally reach an active current limit on both positive and negative swings.

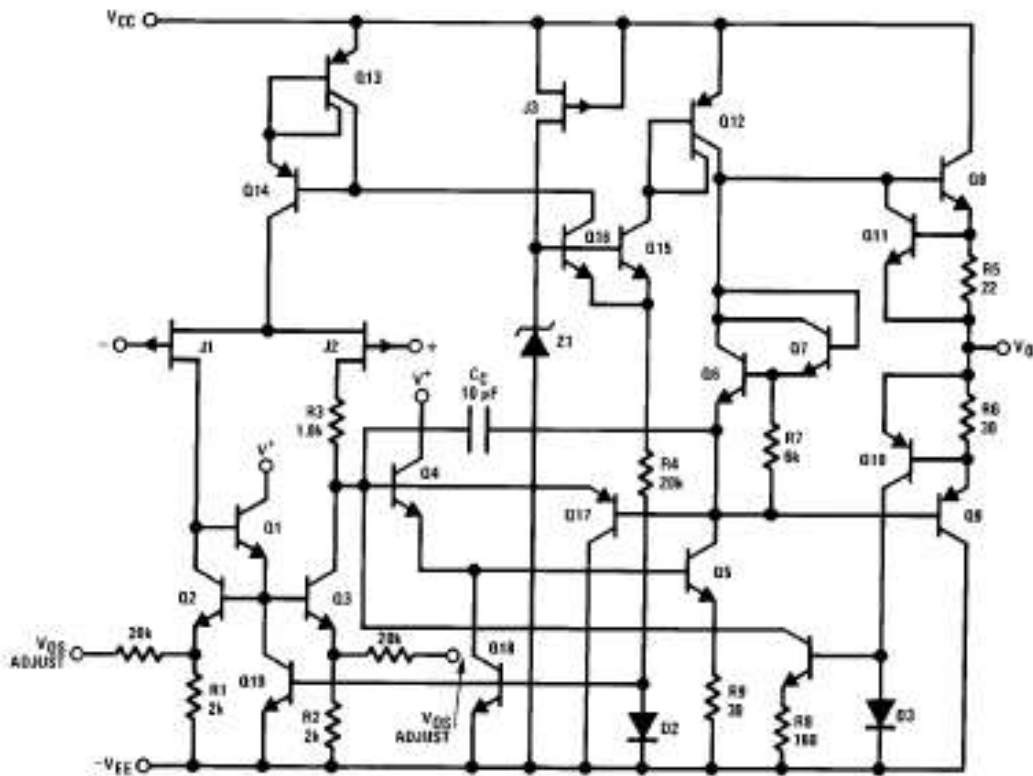
Precautions should be taken to ensure that the power supply for the integrated circuit never becomes reversed in polarity or that the unit is not inadvertently installed back-

wards in a socket as an unlimited current surge through the resulting forward diode within the IC could cause fusing of the internal conductors and result in a destroyed unit.

As with most amplifiers, care should be taken with lead dress, component placement and supply decoupling in order to ensure stability. For example, resistors from the output to an input should be placed with the body close to the input to minimize "pick-up" and maximize the frequency of the feedback pole by minimizing the capacitance from the input to ground.

A feedback pole is created when the feedback around any amplifier is resistive. The parallel resistance and capacitance from the input of the device (usually the inverting input) to AC ground set the frequency of the pole. In many instances the frequency of this pole is much greater than the expected 3 dB frequency of the closed loop gain and consequently there is negligible effect on stability margin. However, if the feedback pole is less than approximately 6 times the expected 3 dB frequency a lead capacitor should be placed from the output to the input of the op amp. The value of the added capacitor should be such that the RC time constant of this capacitor and the resistance it parallels is greater than or equal to the original feedback pole time constant.

Detailed Schematic

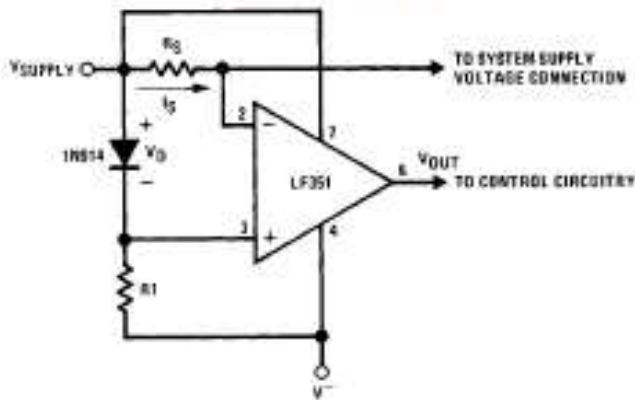


TL1475640-9



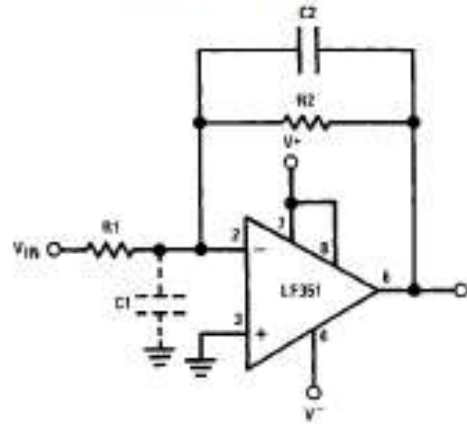
Typical Applications

Supply Current Indicator/Limiter



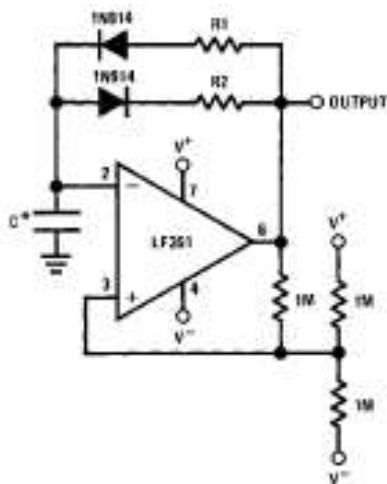
* V_{OUT} switches high when $R_2 I_S > V_D$

Hi- Z_{IN} Inverting Amplifier



Parallel input capacitance $C_1 = (3 \text{ pF for LF351 plus any additional layout capacitance})$ interacts with feedback elements and creates undesirable high frequency pole. To compensate, add C_2 such that: $R_2 C_2 = R_1 C_1$

Ultra-Low (or High) Duty Cycle Pulse Generator



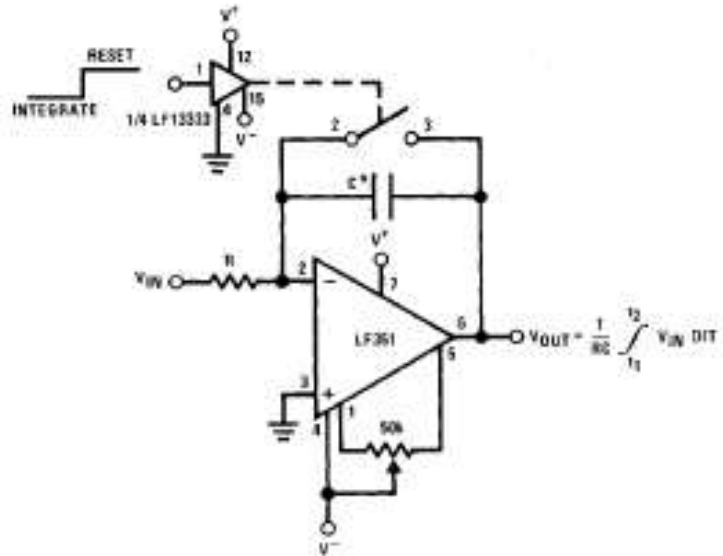
* $I_{OUTPUT HIGH} = R_1 C \ln \frac{4.8 - 2V_S}{4.8 - V_S}$

* $I_{OUTPUT LOW} = R_2 C \ln \frac{2V_S - 7.8}{V_S - 7.8}$

where $V_S = V^+ + |V^-|$

* low leakage capacitor

Long Time Integrator



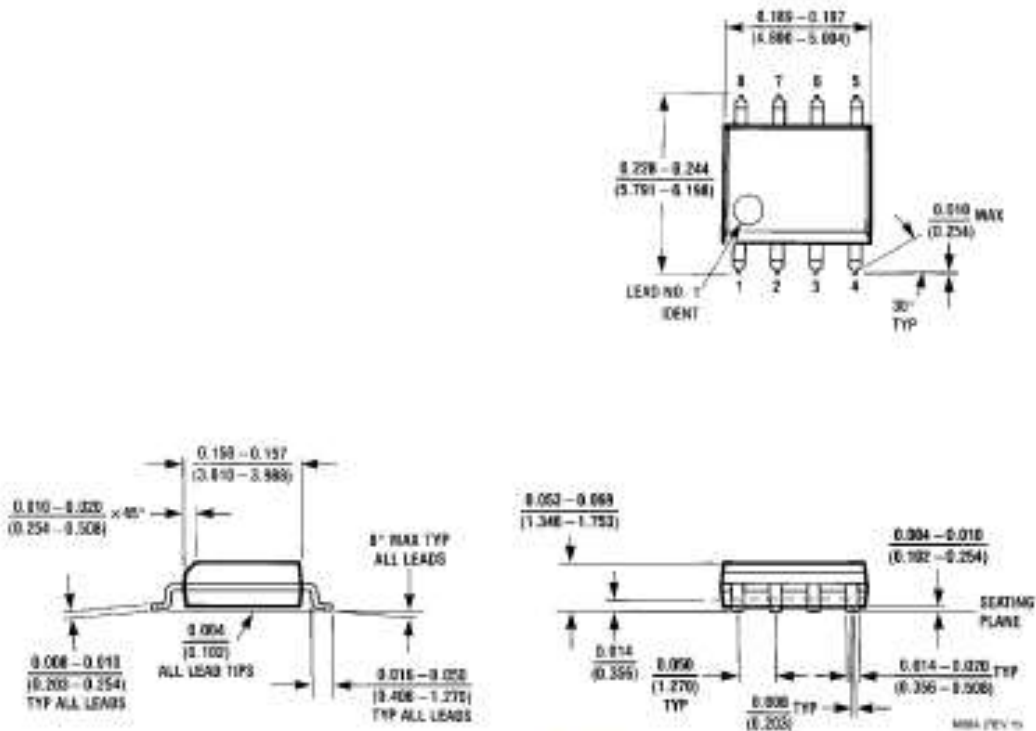
* Low leakage capacitor

* 50k pot used for less sensitive V_{OS} adjust

TL714/5648-10



Physical Dimensions inches (millimeters)

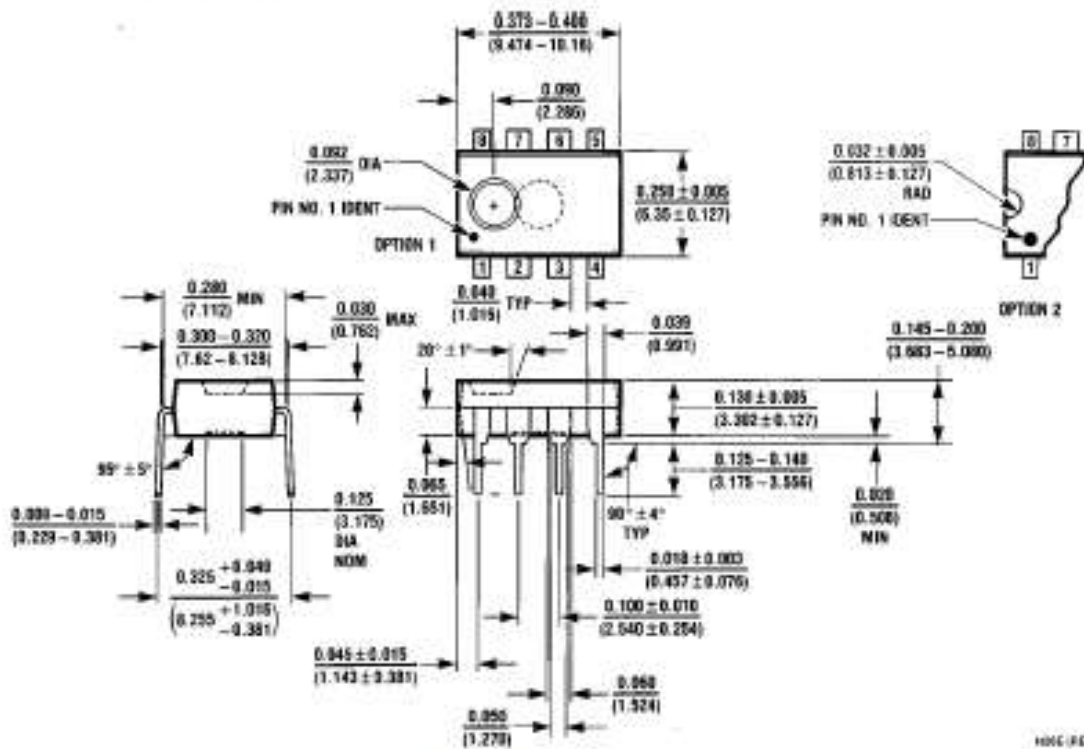


SO Package (M)
Order Number LF35 1M
MS Package Number M08A



LF351 Wide Bandwidth JFET Input Operational Amplifier

Physical Dimensions inches (millimeters) (Continued)



Molded Dual-In-Line Package (N)
Order Number LF351N
NS Package Number N08E

1096 (REV F)

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.



National Semiconductor Corporation
1111 West Bend in Road
Arlington, TX 76017
Tel: (1930) 272-9959
Fax: 1(930) 737-7019

National Semiconductor Europe
Fax: (+49) 0-180-510 85 06
Email: info@sem2.nsc.com
Deutsch: Tel: (+49) 0-180-510 85 05
English: Tel: (+49) 0-180-512 70 32
Français: Tel: (+49) 0-180-512 80 58
Italiano: Tel: (+49) 0-180-514 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Green Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-0-43-299-2100
Fax: 81-0-43-299-2-400



2x16 LCD Module

DATA SHEET

RoHS
Leadfree Soldering

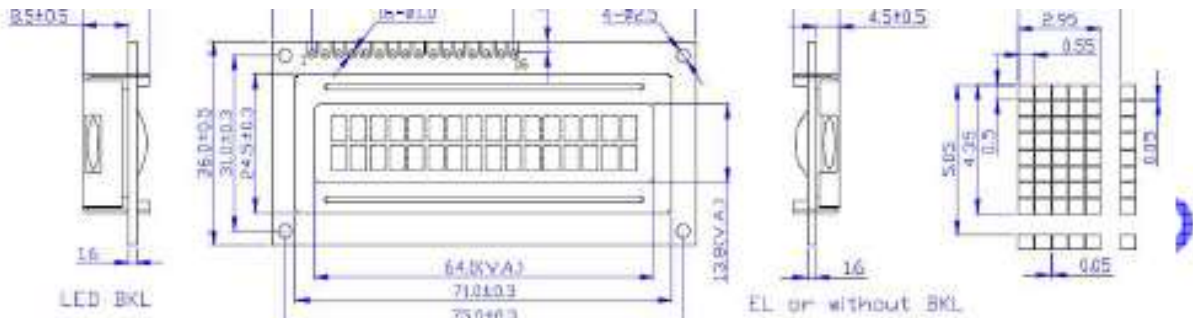


Specifications:

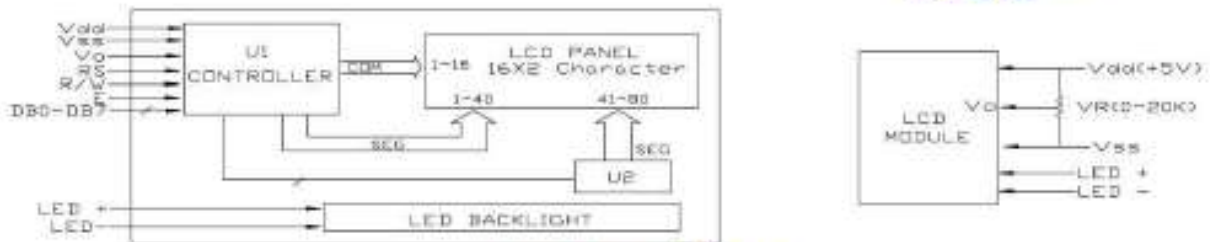
Display Format	: 16 characters (W) x 2 lines (H)
General Dimensions	: 30.0 mm (W) x 36.0 mm (H) x 9.5 mm (T)
Character Size	: 2.95 mm (W) x 4.35 mm (H)
Character Pitch	: 3.65 mm (W) x 5.05 mm (H)
Viewing Area	: 6.70 mm (W) x 13.8 mm (H)
Dot Size	: 0.55 mm (W) x 0.50 mm (H)
Dot Pitch	: 0.60 mm (W) x 0.55 mm (H)
Display Type	: Positive or Negative
LC Fluid	: STN Yellow-Green
Backlight LED	: Optional
Polarizer Mode	: Reflective
View Angle	: 6 o'clock or 12 o'clock
Controller	: S6A0069 or Equivalent
Temperature Range	: 0°C to 50°C (Operating); -20°C to 70°C (Storage)

Diagrams:

4th floor, #10 of Ai-De Air Industry Park,
Gao Qi South 12th Road, Xiamen, Fujian 361006, China
Tel: (86) 592-602-3160 Fax: (86)592-566-5162
Email: sales@olano.net
Website: www.olano.net



Outlined Dimension



Block Diagram

Pin Connections:

Pin Number	Symbol	Function
1	Vss	Ground for Logic
2	Vdd	Power Supply for Logic
3	Vo	Power Supply for LCD
4	RS	Register Selection (H: Data, L: Instruction)
5	RW	Read/Write Selection (H: Read, L: Write)
6	E	Enable Signal
7-14	DB0 - DB7	Data Bus Lines
15	A	BKL +
16	K	BKL -

Electrical Characteristics:

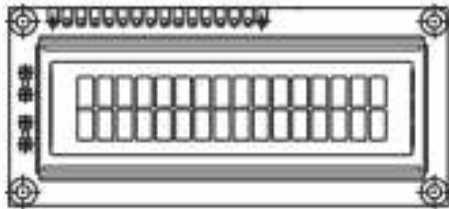
Item	Symbol	Test Condition	Minimum	Typical	Maximum
Operating Voltage (V)	Vdd	Ta=25°C	-	5.0	-
Operating Voltage for LCD (V)	Vlcd	Ta=25°C	-	4.5	-
Current Supply (mA)	Idd	Ta=25°C, Vdd=5.0V	-	2.0	3.0
Voltage Supply for LED (V)	Vf	Ta=25°C, R=6.8Ω	-	4.2	-
Current Supply for LED (mA)	If	Ta=25°C, Vf=4.2V	-	110	-



LCD-016M002B

Vishay

16 x 2 Character LCD



FEATURES

- 5 x 8 dots with cursor
- Built-in controller (KS 0068 or Equivalent)
- + 5V power supply (Also available for + 3V)
- 1/16 duty cycle
- B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED)
- N.V. optional for + 3V power supply

MECHANICAL DATA		
ITEM	STANDARD VALUE	UNIT
Module Dimension	80.0 x 36.0	mm
Viewing Area	66.0 x 16.0	mm
Dot Size	0.56 x 0.66	mm
Character Size	2.96 x 5.56	mm

ABSOLUTE MAXIMUM RATING					
ITEM	SYMBOL	STANDARD VALUE			UNIT
		MIN.	TYP.	MAX.	
Power Supply	VDD-VSS	-0.3	-	7.0	V
Input Voltage	VI	-0.3	-	VDD	V

NOTE: VSS = 0 Volt, VDD = 5.0 Volt

ELECTRICAL SPECIFICATIONS							
ITEM	SYMBOL	CONDITION	STANDARD VALUE			UNIT	
			MIN.	TYP.	MAX.		
Input Voltage	VDD	VDD = + 5V	4.7	5.0	5.3	V	
		VDD = + 3V	2.7	3.0	3.3	V	
Supply Current	IDD	VDD = 5V	-	1.2	3.0	mA	
Recommended LC Driving Voltage for Normal Temp. Version Module	VDD - VO	- 20 °C	-	-	-	V	
		0 °C	4.2	4.8	5.1		
		25 °C	3.8	4.2	4.6		
		50 °C	3.6	4.0	4.4		
LED Forward Voltage	VF	25 °C	-	4.2	4.6	V	
LED Forward Current	IF	25 °C	Array	-	130	260	mA
			Edge	-	20	40	
EL Power Supply Current	IEL	Vol = 110VAC/400Hz	-	-	5.0	mA	

DISPLAY CHARACTER ADDRESS CODE:																
Display Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DD RAM Address	00	01														0F
DD RAM Address	40	41														4F



LCD-016M002B

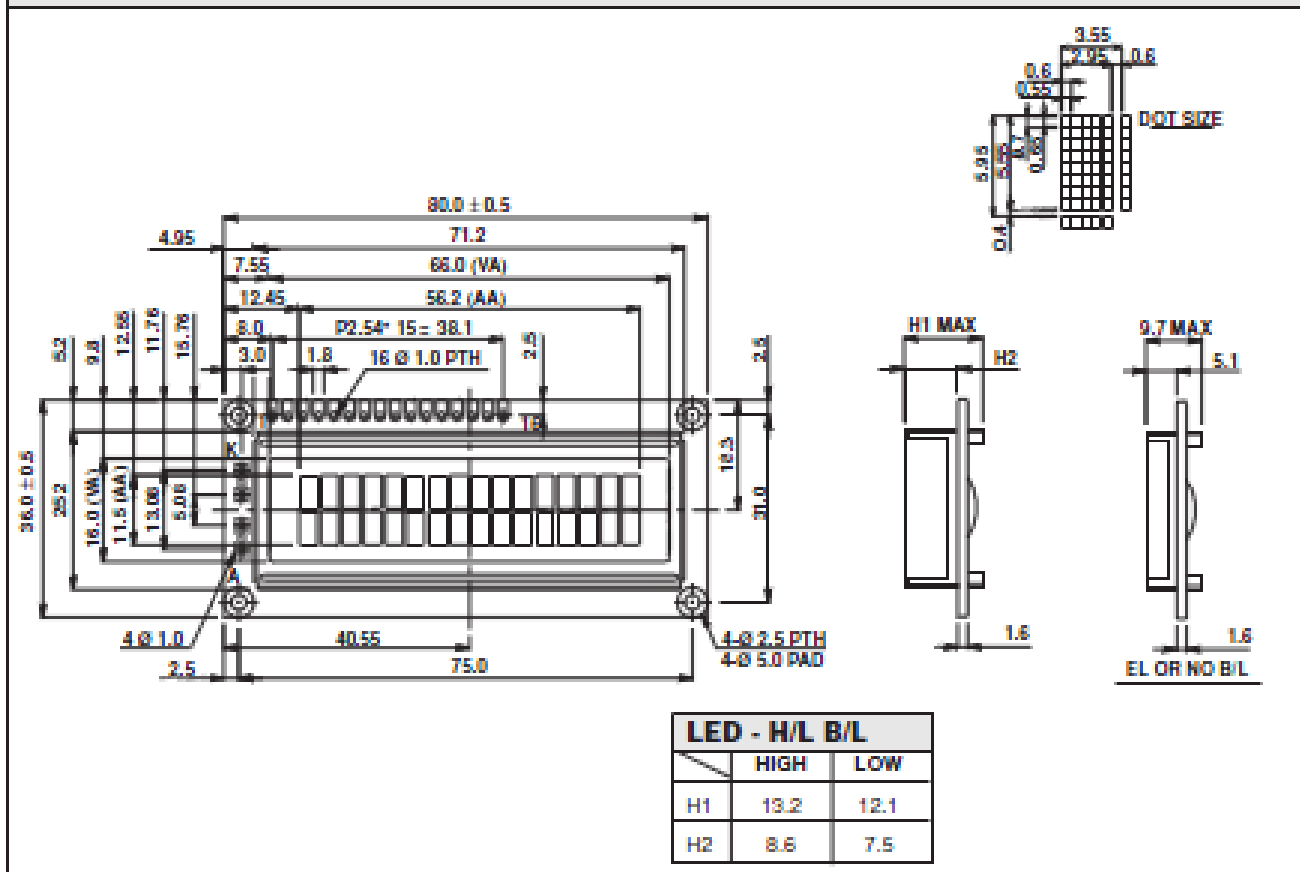


Vishay

16 x 2 Character LCD

PIN NUMBER	SYMBOL	FUNCTION
1	Vss	GND
2	Vdd	+ 3V or + 5V
3	V ₀	Contrast Adjustment
4	RS	H/L Register Select Signal
5	R/W	H/L Read/Write Signal
6	E	H → L Enable Signal
7	DB0	H/L Data Bus Line
8	DB1	H/L Data Bus Line
9	DB2	H/L Data Bus Line
10	DB3	H/L Data Bus Line
11	DB4	H/L Data Bus Line
12	DB5	H/L Data Bus Line
13	DB6	H/L Data Bus Line
14	DB7	H/L Data Bus Line
15	A/Vee	+ 4.2V for LED/Negative Voltage Output
16	K	Power Supply for B/L (OV)

DIMENSIONS in millimeters





فصل پنجم

نتیجه گیری:

در نتیجه ما با جمع لحظه ای چند موج سینوسی می توانیم یک موج غیر سینوسی داشته باشیم که این کار باید با جمع لحظه به لحظه انجام شود که با جمع لحظهای این موج ها می توان یک موج مربعی داشت



کتاب میکرو کنترلر 8051 نوشته محمد علی مزیدی

کتاب میکرو کنترلرها avr نوشته محمد مهدی پرتوی فر وفرزاد مظاهریان (ویراست دوم)

سایت www.datasheetcatalog.com و سایت های الکترونیکی

برای دانلود فایل های آموزشی بیشتر به وب سایت ما مراجعه کنید:

MicroDesigner.ir