پروژه فانکشن ژنراتور دیجیتال DDS با AVR

Melec.ir

میکرو دیزاینر الکترونیک

پروژه های الکترونیک

تعریف پروژه

در این پروژه سعی بر آن است با استفاده از میکروکنترلر های سری AVR و زبان برنامه نویسی C ، یک فانکشن ژنراتور دیجیتال با امکانات یک فانکشن ژنراتور آزمایشگاهی طراحی ، اجرا و به عنـوان پروژه ی پایان ترم ارائه شود .

مشخصات این طرح به شرح زیر ست :

پارامتر های الکتریکی		
عنوان	رنج	Resolution
رنج فرکانسی	1 HZ ~ 1 MHZ	1 Hz
رنج دامنه	0 ~ 10 V	1 mV
رنج DC Offset	0 ~± 10 V	1 mV
شکل موج های قابل تهیه	Sin , Squar , ThreeAnglular	
Symmetry	0 ~ %100	٪1

مشخصات بلوکی	
ورودی	5 * 2 Keypad
خروجی	Main OutPut
سیستم نمایشی	16 * 2 LCD

فصل ۰

فانکشن ژنراتور

فانکشن ژنراتور یک سیستم الکترونیکی جهت تولید شکل موج های مختلف با قابلیت کنترل فرکانس ، دامنه ، مقدار DC و... می باشد . در بسیاری از فانکشن ژنراتور های مدرن امکانات دیگری چون کنترل تقارن, Synchronizasion , Sweep تولید شکل موج دلخواه (قابلیت Customize) اتصال به کامپیوتر و... نیز اضافه شده است

انواع فانکشن ژنراتور ها

در یک نگاه کلی میتوان فانکشن ژنراتور ها را به دو نوع انالوگ و دیجیتال تقسیم کرد.

۱. فانکشن ژنراتور انالوگ

در این نوع ، برای تولید موج از قطعات و مدارات خطی چون مدار LC , RC و ... استفاده میکنند . به همین علت معمولا حجم فیزیکی بالایی دارند . علاوه بر ان در این نوع فانکشن ژنراتور ، کنترل به صورت پیوسته و با قطعاتی چون پتانسیو متر ،سلکتور و ... انجام می گیرد . دقت این روش پایین است (حداکثر یک رقم اعشار) و در رنج فرکانسی پایین با مشکل مواجه اند . همچنین جهت بهبود پایداری و دقت ، تدابیر ویژه ای لازم است . تعداد شکل موج های قابل تهیه در این روش محدود . و معمولا THD بالایی دارند .

۲. فانکشن ژنراتور دیجیتال

در فانکشن ژنراتور دیجیتال ، یک اسیلاتور با فرکانس ثابت و پایداری بسیار بالا (مثل اسیلاتور کریستالی) ، فرکانس مرجعی را تولید می کند و مدارات دیجیتال و شکل موج دلخواه با فررکانس مورد نظر را از این فرکانس به دست می آورند .با توجه به ویژگی های مدار منطقی ، یک فانکشن

ژنراتور دیجیتال نسبت به همنوع انالوگ خود ، حجم کمتری دارد . کنترل مولفه های آن به صورت عددی بوده و از این رو بسیار دقیق است . به علت استفاده از اسیلاتور رکیستالی ، دقت و پایداری آن فوق العاده بالاست . ایجد تغییرات (نرم افزاری) در آن بسیار ساده است .با تدابیر خاصی می توان هر شکل موج دلخواهی را ایجاد کرد . خاصیت عددی آن ، اتصال به کامپوتر را ممکن ساخته است . کل مدار آن ، به طور یکپارچه و بر روی یک IC قابل اجرا بوده و از این نظر نسبت به نوع انالوگ مقرون به صرفه تر است .بزرگترین مشکل فانکشن ژنراتور دیجیتال ، فرکانس کاری پایین آن است که حداکثر به ۲۵٪ فرکانس مرجع محدود می شود .

مقایسه ی مشخصات فوق ، فانکشن ژنراتور دیجیتال را در بسیاری از موارد ، نسبت به نوع انالوگ ارجح تر می کند . البته باید در نظر داشت که قابلیت های منحصر به فرد فانکشن ژنراتور آنالوگ برای آن کاربرد های خاصی ایجاد کرده است . مثلا امکان کار در فرکانس های بسیار بالا و کاملا خطی بودن خروجی ، مواردی است که در نوع دیجیتال به سختی و یا با صرف هزینه ی بسار بالا قابل دستیابی است .

کاربرد فانکشن ژنراتور

صرف نظر از نوع فانکشن ژنراتور ، می توان آن را هر جا که یک اسیلاتور نیاز است ، استفاده کرد . اما از آنجا که در بیشتر موارد از اسیلاتور های ثابت نیاز است و با در نظر گرفتن هزینه ی تولید یک فانکشن ژنراتور (در مقایسه با یک اسیلاتور ثابت) ، کاربرد فانکشن ژنراتور معمولا به آزمایشگاه ها و تعمیرگاه ها محدود می شود که در این گونه کاربرد ، فانکشن زنراتور به عنوان یک منبع سیگنال با مشخصات مورد نیاز مدار اصلی ، به طور موقت جانشین اسیلاتور مدار تحت تعمیر یا تحت آزمایش میشود.

مشخصات کاری فانکشن ژنراتور

آنچه به عنوان مشخصات کاری یک دستگاه مطرح می شود ، در واقع شامل رنج کاری و محـدودیت های ان دستگاه از نظر شرایط الکتریکی (ورودی و خروجی) و شرایط فیزیکی می باشد

مشخصات الکتریکی

رنج فرکانسی :

فانکشن ژنراتور دیجیتال بر خلاف نوع آنالوگ ،از لحاظ فرکانس های پایین عملا مشـکلی نداشـته و می تواند حتی چند صدم هرتز را نیز در خروجی ایجاد کند . اما از نظر رنـج بـالایی فرکـانس بسـیار محدود تر از نوع آنالوگ اند . چرا که معمولا از روش تقسیم فرکانسی استفاده می کنند واینکـه ، بـا تکنولوژی ساخت یکسان ، حداثر فرکانس قطعات دیجیتال بسیار کمتر از نوع آنـالوگ مـی باشـد و آنچه فرکانس خروجی را محدود می کند همین عامل است .

در هر حال در این پروژه سعی بر ان است رنج 1Hz تا 1KHz در خروجی تامین شود .

رنج دامنه:

کنترل دامنه در فانکش ژنراتور دیجیتال به روشی مشابه نوع آنالوگ انجام می پذیرد منتهی بادقـت بیشتر تنها عامل محدودکننده ی دامنه، حداکثر ولتاژ تغذیه توان قطعات خروجی و ... می باشـد کـه این موارد در تمام مدارات الکترونیکی عوامل گریز ناپذیرند. دامنه ی این طرح $\pm 10v$ می باشد.

امپدانس ورودی و خروجی:

فانکش ژنراتور دستگاهی نیست که بتوان برای ورودی یا خروجی آن امپدانس خاصی در نظر گرفت چرا که معمولاًدر آن مساله ی تطبیق امپدانس مطرح نمی باشد و معمولاً این گونه درنظر می گیرند که هر چه zi بیشتر باشد (ایده آل بی نهایت) و هر چه zo کمتر باشد (ایده آل صفر) بهتر است.

سایر مشخصات الکتریکی مثل ولتاژ تغذیه، تولرانس الکتریکی، حداکثر توان دریافتی از منبع تغذیه و ... در این پروژه چندان مطرح نیست.

مشخصات فیزیکی

وزن، حجم، شکل و ... از جمله مشخصات فیزیکی یک دستگاه الکتریکی اند که دست کم در این پروژه اهمیت چندانی ندارد.

شرایط محیطی:

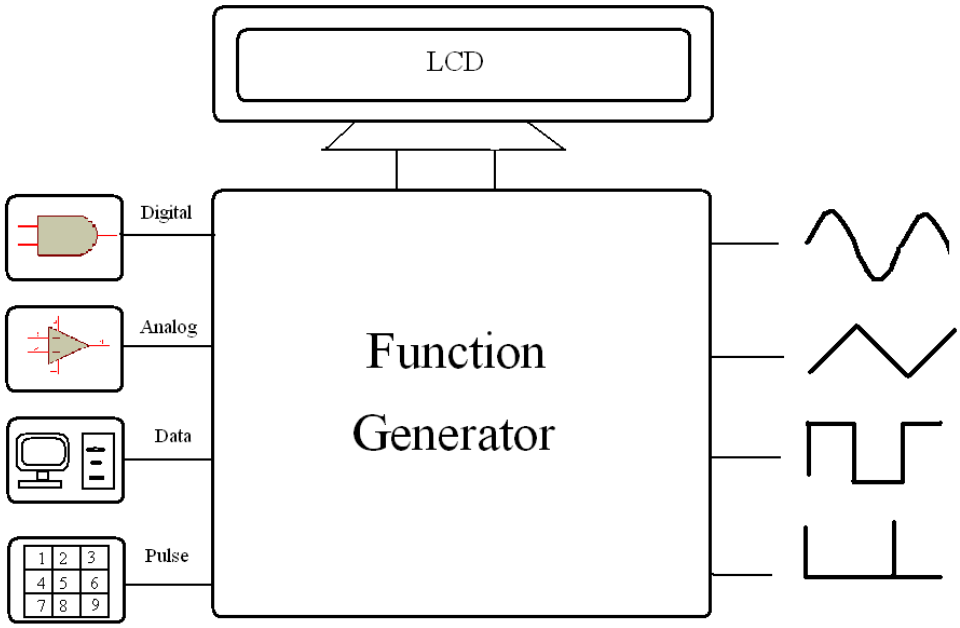
دما، رطوبت، نور، گرد و خاک و ... شرایط محیطی مطرح برای یک مدار الکترونیکی است. یک فانکشن ژنراتور معمولاً برای کار در شرایط آزمایشگاه (یا اتاق) طراحی می شود.

جهت آشنایی بیشتر با فانکشن ژنراتور و ایجاد یک ذهنیت کلی از پروژه Data Sheet یک فانکشن ژنراتور دیجیتالی تجاری درضمیمه ی یک ارائه شده است.

میانگفتار۱

فانکشن ژنراتور دیجیتال

شاید بهتر باشد قبل از ورود به جزئیات طراحی فانکشن ژنراتور ، آن را به عنوان یک بلـوک در نظـر گرفته و مشخصات دقیق تری از آن تعریف کنیم . هر فانکشن ژنراتور تعدادی ورودی و یک یا چنـد خروجی دارد (شکل ۱- آ)



شکل ۱- آ

وضعیت این خروجی ها از نظر مشخصات سیگنال به وضعیت فعلی یا آخرین تغییر خروجی بسـتگی دارد .

در یک فانکشن دیجیتـال ورودی Keypad کـه در Pannel خـود دسـتگاه قـرار دارد ورودی استاندارد محسوب می شود . برای کاهش هزینه نیز فقط یک خروجی اصلی سیگنال بـا مشخصـات

قابل کنترل در نظر گرفته می شود . البته اغلب خروجی دیگری همفاز و هم فرکانس با سیگنال خروجی اصلی در نظر گرفته می شود که شکل موج آن همواره مربعی و دامنه ی آن همواره 5 V بوده و با نام TTL Output شناخته می شود .

با توجه به اینکه فانکشن ژنراتور دیجیتال بر خلاف نوع آنالوگ خود هیچگونه عامل فیزیکی برای نمایش مقادیر خروجی ندارد ، باید یک سیستم Monitoring برای آن در نظر گرفت .بدین منظور می توان یک خط Data به صورت خروجی جهت وصل به کامپیوتر (یا سایر سیستم ها) تعریف کرد واطلاعات را به نمایشگر آنها ارسال کرد .اما معمولا در سیتم های مستقل ، یک نمایشگر کوچک از نوع Seven Segment یا LCD در پانل دستگاه تعبیه شده و اطلاعات ورودی و خروجی را نمایش می دهد .

فصل ۱

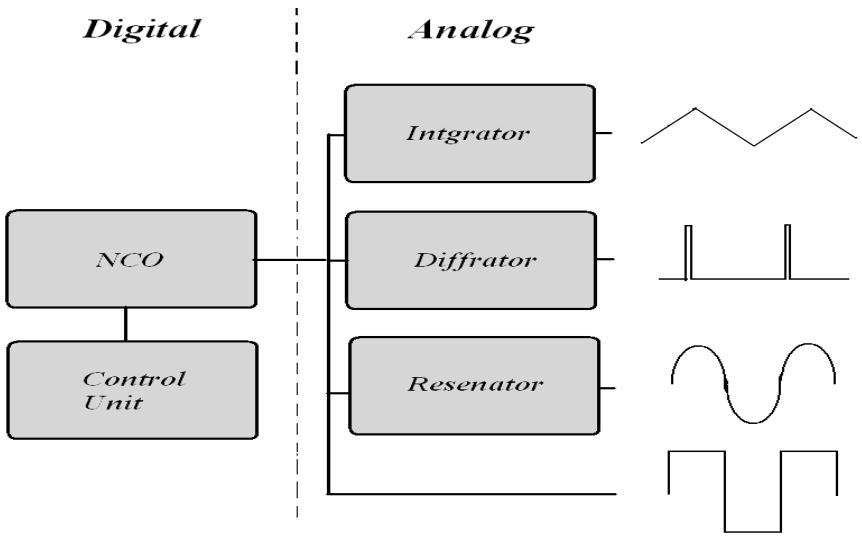
تولید شکل موج به روش دیجیتال

روش های تولید شکل موج به روش دیجیتال

عموماً از دو روش برای تولیدشکل موج، به صورت دیجیتال، استفاده می شود.

۱.۱استخراج مستقیم :

در این روش یک موج مربعی با فرکانس (و گاهی دامنه) قابل کنترل تولیدو با استفاد از مدارات مشتق گیر، انتگرال گیر، مدار تشدید و ... شکل موج موردنظر از آن استخراج می شود.(شکل ۱ = ۱) استخراج شکل موج دلخواه معمولاً توسط مدارات آنالوگ انجام گرفته و تنها کاری که قسمت دیجیتال می کند، کنترل مشخصات خروجی و کنترل قسمت آنالوگ می باشد.



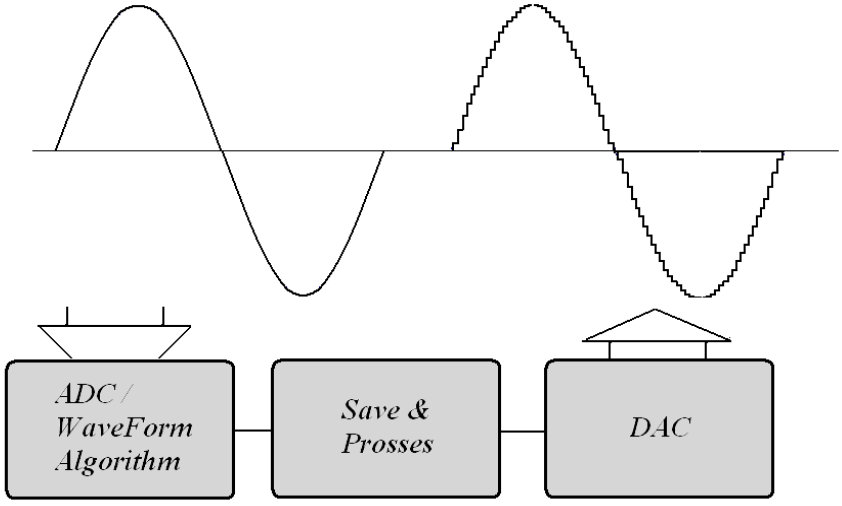
شکل ۱ = ۱

علی رغم برخی قابلیت ها این روش کاربرد چندان ندارد.

۲. DDS :

در این روش مقادیر لحظه ای شکل موج موردنظر در نقاطی با فاصله زمانی مناسب، محاسبه و یـا از یک منبع واقعی نمونه برداری و ذخیره شده و به طور متوالی و با فاصله ی زمانی مناسب به خروجی اعمال شوند. این اعداد در خروجی تبدیل به ولتاژ شده و مقادیر لحظه ای شکل مـوج مـورد نظـر بـا مشخصات قابل کنترل را می سازند

(شکل ۲ – ۱).

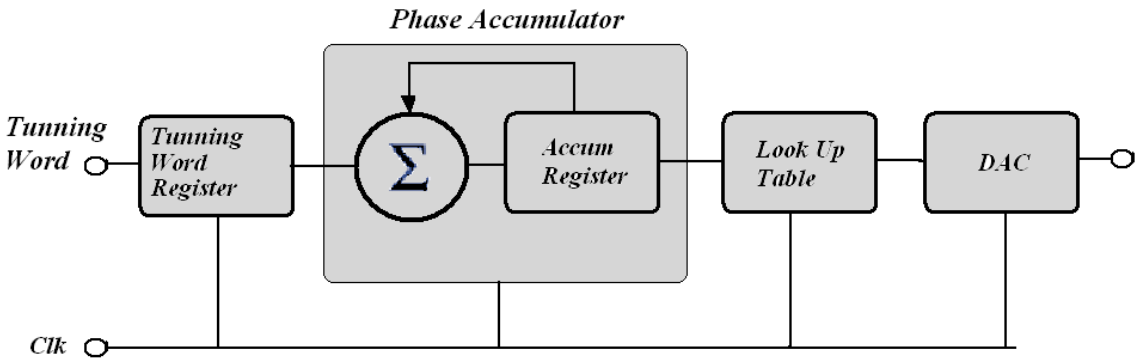


شکل ۲–۱

آنچه امروزه به عنوان فانکشن ژنراتور دیجیتال شناخته می شود. از این روش که اصطلاحاً DDS Direct Digital Sentences : نامیده می شود ، استفاده می کند .

اصول کلی DDS :

شکل ۳-۱ بلوک دیاگرام کلی این طرح را نشان می دهد.



شکل ۳-۱

در این شکل Phase Accumulator عدد قبلی را به طور پیوسته با Tunning Word جمع کرده و حاصل را به Look Up Table اعمال میکند . وقتی حاصل به بیشترین مقدار خود (N ظرفیت سیستم بوده و معمولا ۲۴ در نظر می گیرند)

Look Up Table حافظه ای به صورت M * A است که در آن مقدار هر آدرس متناسب است با مقدار دامنه ی شکل موج به ازای ان آدرس (در حالت استاندارد شکل موج سینوسی و تابع ان به صورت

$$F(n)=ASin(2\pi\frac{n}{N})$$

معادله ی ۱-۱

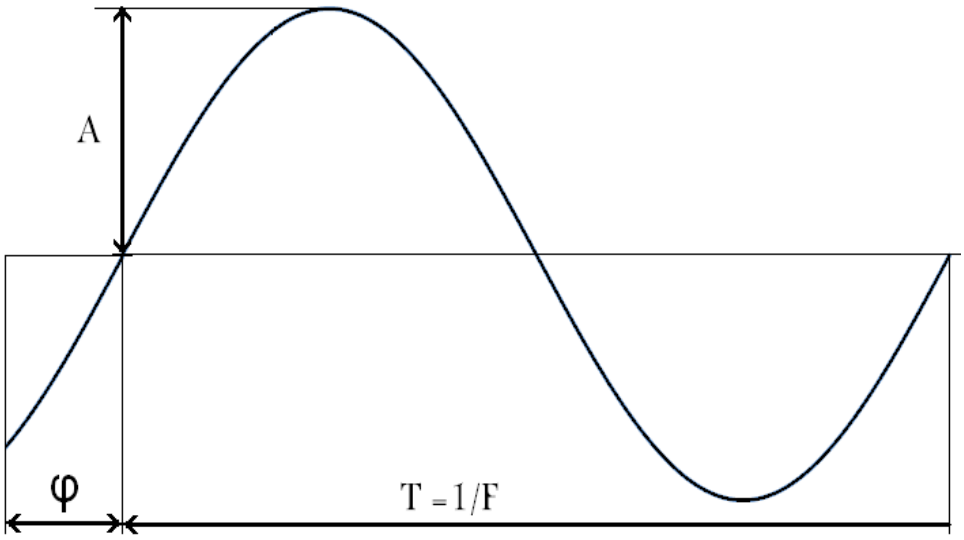
می باشد که در آن A حداکثر عدد دامنه ،معمولا برابر ۲۵۶ است .)

جهت تبدیل مقادیر عددی به ولتاژ متناظر ، از DAC استفاده میشود .

میانگفتار ۲

مشخصات یک شکل موج:

شکل ۱ – ب نشان دهنده ی یک شکل موج سینوسی می باشد.



شکل ۱ – ب

ضابطه ی تابع این شکل موج به شرح زیر است...

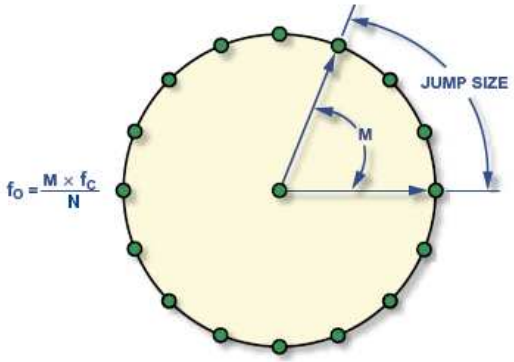
$$v(t) = A\sin(\omega t + \varphi)$$
$$\omega = 2\pi f$$

معادله ی ۱- ب

همان طور که از این معادله وشکل پیداست یک موج سینوسی با سه عامل فرکانس، دامنه و فـاز بـه طور کامل تعریف می شود. تمام شکل موج های استاندارد را می توان بـا همـین سـه عامـل تعریـف کرد.که البته با تعابیر مختلفی از ان یاد می شود. بنابراین فرکانس دامنه و فاز بـه عنـوان مشخصـات شکل موج شناخته می شود بخش های بعدی مقاله تعریف دقیق تر و نحوه ی کنترل و تغییر مقادیر آنها در این پروژه را به تفصیل بررسی می کند.

فصل ۲ : کنترل فرکانس

برای درک بهتر اینکه در روش سنتز دیجیتال، فرکانس چگونه کنتـرل مـی شـود، سـیکل مثلتـاتی شکل ۱ – ۲ ،که مربوط به یک موج سینوسی است ،را در نظر می گیریم.



شکل ۱-۲

نقطه های سبز نشان دهنده ی زاویه هایی است که در ان ها مقدار لحظه ای ولتاژ در حافظه Look Up Table موجود است .

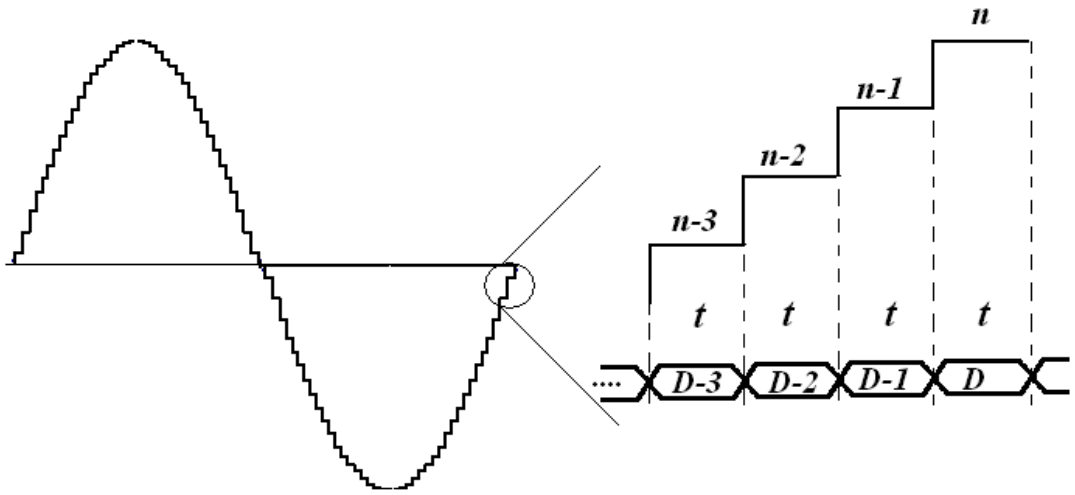
معادلات فرکانس خروجی :

اگر تعداد کل این نقاط N فاصله ی بین دو مقدار متوالی که در خروجی ظاهر می شـود M و فرکانس چرخش f_c باشد خواهیم داشت :

$$F_O = M \frac{F_C}{N}$$

معادله ی ۱ – ۲

توضیحات ارائه شده مربوط به استانداردی است که فقط قادر به تولید شکل موج سینوسی است . جهت درک بهتر تولید سایر شکل موج ها و ورود به بحث کنترل فرکانس در این پروژه قسمتی از شکل ۲ – ۱ را در زیر تکرار می کنیم:



شکل ۲ – ۲

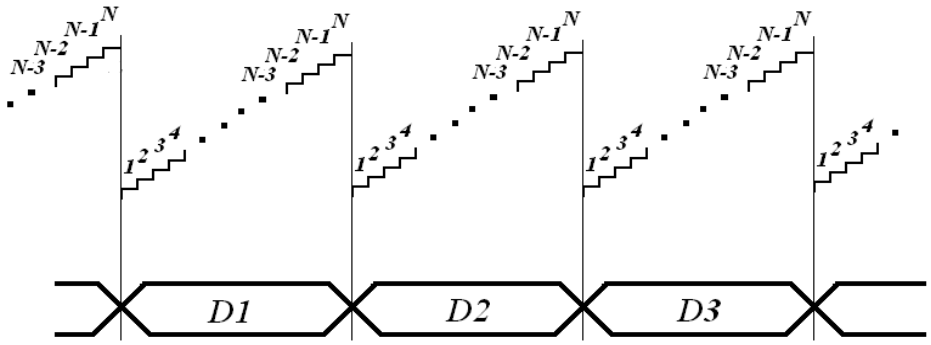
با توجه به این شکل اگر تعداد نمونه های اعمال شده در یک سیکل را N و فاصله ی زمانی t باشد, دوره تناوب موج برابر است با :

$$T = Nt$$ معادله ی ۲ – ۲

N و t هر دومتغیر بوده و به راحتی قابل کنترل اند. در این قسمت به بررسی نحوه ی کنترل و روابط آنها می پردازیم.

کنترل t :

t یک متغیر زمانی است و آنچه در مدارات دیجیتال به عنوان عنصر زمان بندی متغیر مطرح می شود یک Timer با قابلیت Reload یا قابلیت compare است. (مثل تایمرهای میکروکنتر های سری 80xx یا AVR).



شکل ۳ – ۲

زمان بندی t با استفاده از این تایمر و بارگذاری خروجی توسط وقفه ی آن صورت می گیرد.

کنترل t با متغیر V :

با توجه به شکل ۳ – ۲ می توان گفت :

$$t = V \cdot Tx$$

معادله ی ۳ – ۲

که در آن V عدد تایمر (تعداد پله هایی که تایمر می شمارد) وT_x طول زمانی هر پله، برابر یک machin cycle است که با فرکانس کریستال مورد استفاده رابطه ی عکس دارد.

با توجه به ثابت بودن T_x،تا اینجا حداکثر تعداد فرکانس قابل حصور در خروجی، برابر بزرگترین عدد قابل اعمال به Timer است.(مثلا بااستفاده از تایمر ۱۶ بیتی، ۶۵۵۳۶ فرکانس قابل ایجاد است).

جهت جلوگیری از تداخل وقفه ها و hang شدن میکرو، همواره باید یک فاصله ی زمانی مناسب بین دووقفه ی متوالی وجود داشته باشد. به گونه ای که قبل از اتمام روال اجرای وقفه ی اول، وقفه ی بعدی اتفاق نیافتد. بنابراین V_t نباید از یک مقدار خاص کمتر شود. پس:

$$t=(V0+V)\cdot Tx$$

معادله ی ۴ – ۲

که در آن V_0 عددی ثابت و همواره بزرگتر از تعداد mc لازم برای اجرای زیر برنامه ی وقفه است V نیز مقدار متغیر تایمر است.

به این ترتیب تعداد فرکانس های قابل حصول باز هم محدود می شود.

این محدودیت در صورت استفاده از تایمر ۸ بیتی بسیار چشمگیر است . در این مورد جهت افزایش تعداد فرکانس های قابل حصول (با توجه به ثبات t_x و محدودیت V) باید متغیر دیگری واردمعادله ی t کنیم.

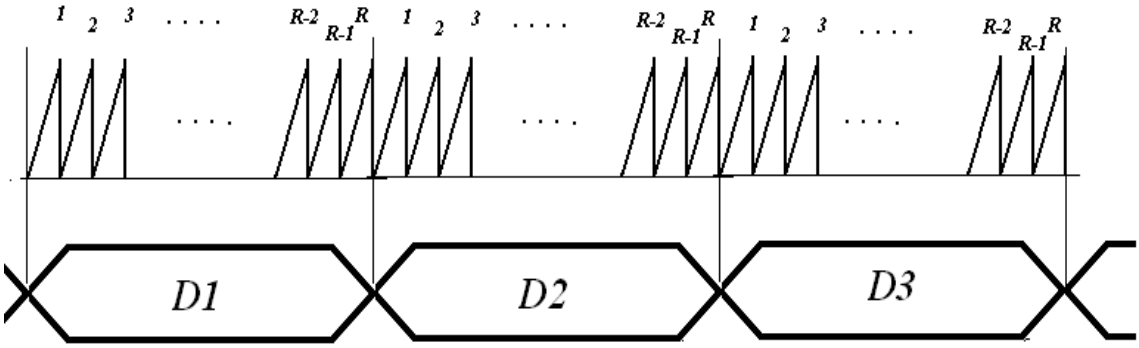
کنترل t با متغیر R :

اگر متغیر R را به گونه ای تعریف کنیم که به ازای هر R بار وقفه ی Timer خروجی یک بار بارگذاری شود (شکل ۴ – ۲) خواهیم داشت:

$$t=R\cdot(V0+V)\cdot Tx$$

معادله ۵ – ۲

برای این کار کافی است که بارگذاری خروجی را شرطی کنیم. تا به همه ی وقفه ها پاسخ ندهد. (وجودR فرکانس های قابل حصول را به طور چشمگیری افزایش می دهد اگر فرض ۱۶ بیتی بودن R را به فرضیات قبلی اضافه کنیم در خروجی حدود $۲^{۲۴}$ فرکانس قابل حصول خواهیم داشت).



شکل ۴ – ۲

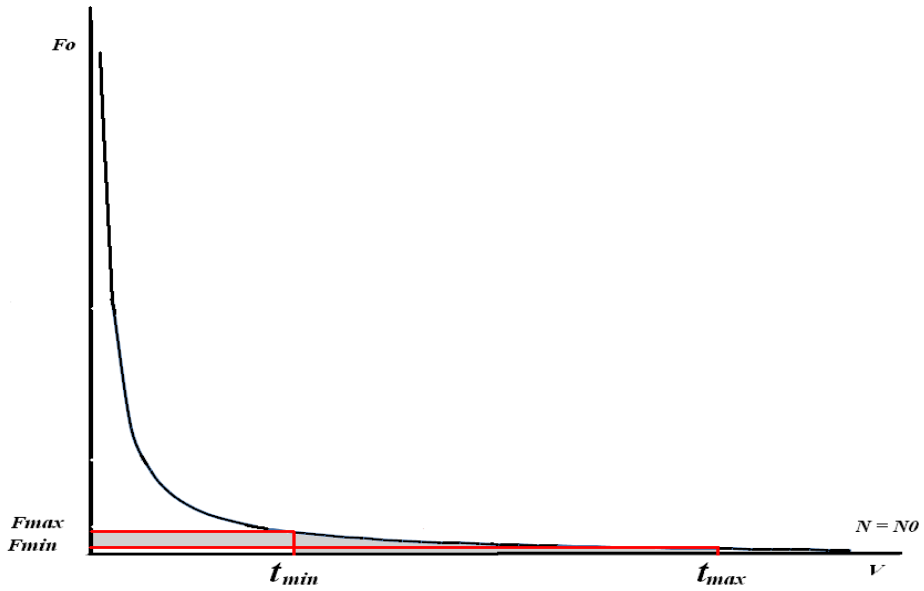
البته در صورت استفاده از تایمر ۱۶ بیتی استفاده از این متغیر لازم نیست .چـرا کـه وارد کـردن R باعث افزایش t و در نتیجه کاهش فرکانس می شود . در حالی که بـا در نظـر گـرفتن متغیـر دیگـر معادله T ، یعنی N ، حتی بدون وجود R نیز تا فرکانس های کمتر از یک هرتز نیز قابل دستیابی است .

کنترل N

لزوم کنترل N :

ممکن است ۶۵۵۳۶ عدد فرکانسی که با کنترل t به دست می آید کافی به نظر برسد اما چند مسئله ی اساسی وجوددارد که در ادامه ی مطلب تشریح می شوند .

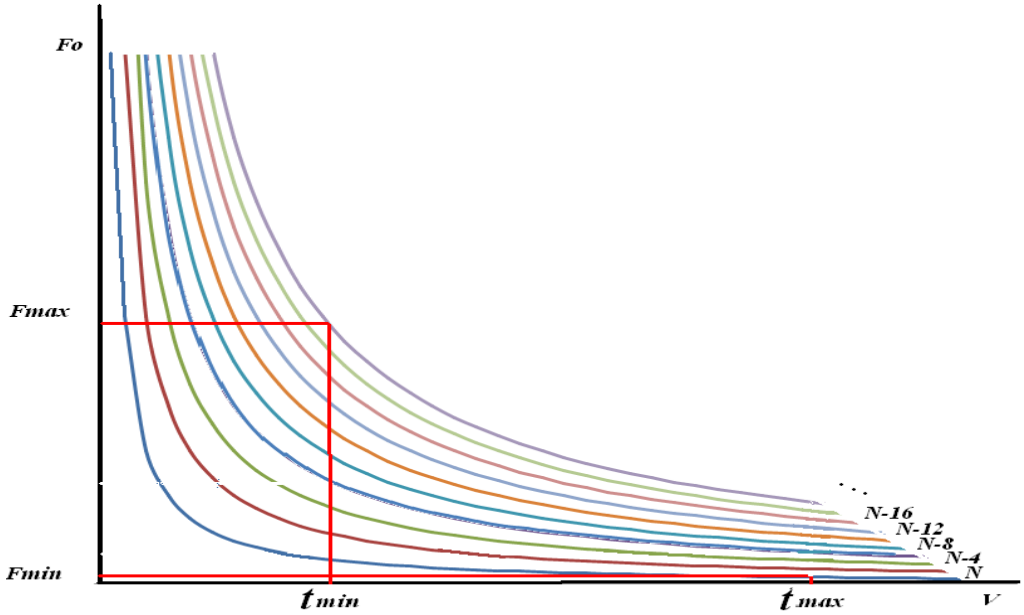
۱. شکل ۵ – ۲ نمودار فرکانس خروجی بر حسب t به ازای N ثابت را نشان می دهد.



شکل ۵ – ۲

مقدار t به بازه ی $[t_{min}, t_{max}]$ محدود است. و این مقادیر محدوده ی کوچکی از رنج فرکانسی مطلوب را در بر می گیرد.(به ازای N=256 بین 1Hz ~500Hz)

همانطور که اشاره شد مشکل محدود بودن t بیشتر برای مقادیر کوچک تر از t_{min} مطرح است چرا که t_{max} برای ایجاد فرکانس های بسیار پائین (حتی در حدود هرتز) به اندازه ی کافی بزرگ است. علاوه بر این، با وارد کردن متغیر R یا وسیع تر کردن محدوده تغیرات آن می توان به فرکانس های بسیار کوچک تر نیز دست یافت اما کاهش مقدار t از t_{min} غیرممکن یا دست کم بسیار مشکل است. بنابراین تنها راه، استفاده از دیگر متغیر معادله ۲-۲ یعنی N است. شکل ۶ – ۲ ، نمودار فرکانس خروجی بر حسب t به ازای N های مختلف را نشان می دهد.

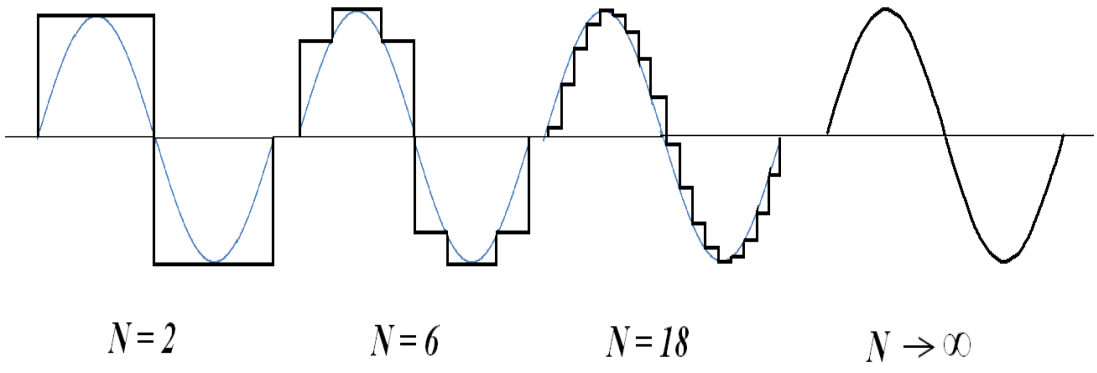


شکل ۶ – ۲

با توجه به این شکل و آنچه در ابتدای فصل در مورد شبیه سازی یک شکل موج بیان شد، می توان نتیجه گرفت که کاهش N باعث افزایش فرکانس (کاهش T) می شود .

محدودیت های N

انتخاب N مقدار برای نیز با محدودیت های خاصی مواجه است .شکل زیر شکل موج های سینوسی ایجاد شده به ازای N های مختلف را نشان می دهد .



شکل۷ – ۲

مشاهده می شود که هر چه N بزرگتر باشد خروجی ایجاد شده به شکل موج واقعی نزدیک تر است. در عمل به علت محدودیت حافظه و قدرت محاسباتی مدار و ... باید N را به مقداری معـین محـدود کرد. با یک محاسبه ی سرانگشتی می توان عدد 256 را برای حداکثر مقدار N در نظر گرفت (نحوه ی به دست آمدن این مقدار در ضمیمه ی ۲ آورده شده است). بنابراین

$$N \leq 256$$ معادله ی ۲-۶

این نکته را نیز باید اضافه کرد که برای شبیه سازی یک سیکل سینوسی، به طوری که قابـل تـرمیم بوده و بتوان THD آن را نادیده گرفت باید حدقل ۶ نمونه در یک پریود اعمال کرد. بنا براین:

$$N \geq 6$$ معادله ی ۲-۷

وبا ترکیب دونامعادله ی فوق می توان نتیجه گرفت:

$$6 \leq N \leq 256$$ معادله ی ۲-۸

برای اینکه تقارن شکل موج به هم نخورد باید تعداد پله ها در همه ی ربع های آن یکسان باشـد بـه عبارت دیگر N باید ضریبی از ۴ باشد.از طرفی برای اینکه تقارن در نقاط Max و Min و دامنـه ی شکل موج به هم نخورد باشد باید مقادیر تابع در نقاط $\pi / 2$ و $3\pi / 2$ همواره بر مقادیر A و A- قرار گرفته و در خروجی ظاهر شوند .بنابر این همواره باید :

$$N = 4k + 2$$ معادله ی ۲-۹

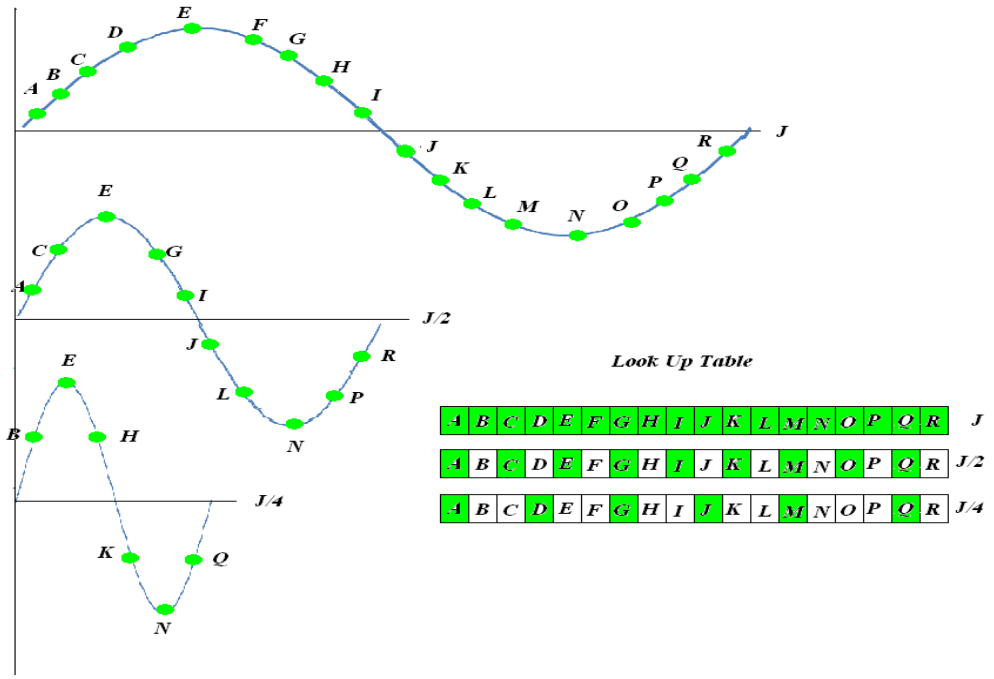
که در آن k تعداد پله ها در ۱/۴ سیکل(بدون در نظر گرفتن $\pi / 2$ و $3\pi / 2$) اسـت.به طوری که :

$$1 \le k \le 63$$ معادله ی ۱۰-۲

روش های کنترل N

افزایش فرکانس با کاهش N به دو روش ممکن است.

۱. شکل موج دلخواه با حداکثر N ممکن ساخته شود و پله ها به صورت غربالی انتخاب شوند. به طوری که مثلاً برای دو برابر کردن فرکانس یک در میان، برای سه برابر کردن فرکـانس دو در میـان برای J+1 برابر کردن فرکانس J در میان انتخاب شود. شکل ۸ – ۲ این موضوع را به خوبی نشـان می دهد .



شکل ۸ – ۲

بنابراین در مورد تعداد پله های اعمال شده در یک فرکانس می توان گفت:

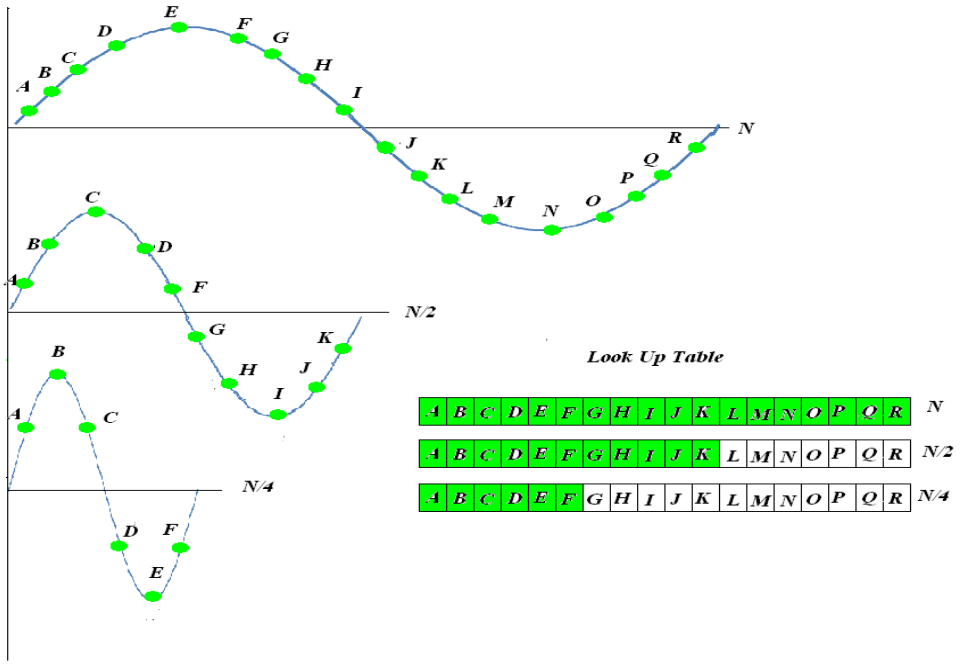
$$N = N_0 \ / \ J$$ معادله ی ۲-۱۱

که درآن N_0 حداکثر تعداد پله ها و J تعداد پله های رد شده است.

البته باید J را طوری انتخاب کرد که معادله ی ۹ - ۲ همواره برقرار باشد . مقادیر ممکن برای J بسیار محدود میشود

این روش به علت عدم نیاز به بازسازی Look Up Table به ازای فرکانس های مختلف (و درنتیجه امکان استفاده از ROM) در DDSهای استاندارد استفاده می شود. و برای رفع محدودیت J ، ظرفیت ROM را به اندازه ی کافی بزرگ انتخاب می کنند .

۲. با توجه به فرکانس مورد نیاز N محاسبه شده و شکل موج مورد نظر با تعداد پله های متغیر تولید شود و روال اعمال پله ها به خروجی همواره ترتیبی (ونه غربالی شود) شکل ۹ – ۲ این موضوع را نشان می دهد.

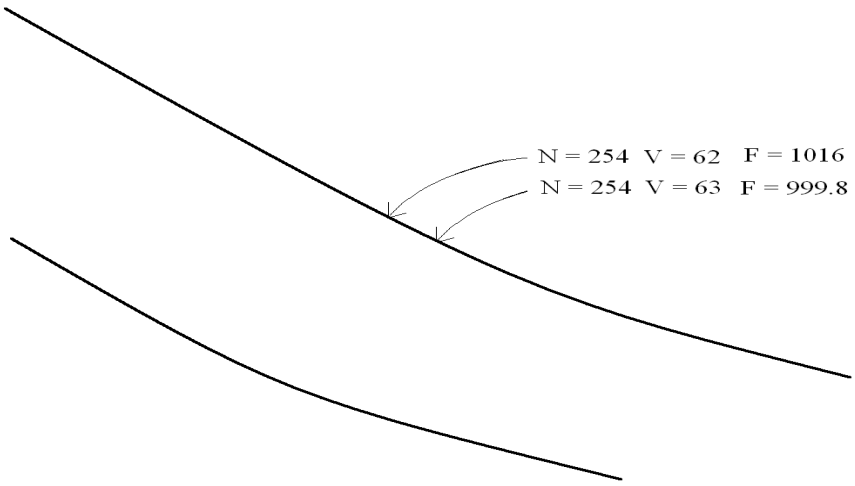


شکل ۹ – ۲

روش دوم به علت عدم تیاز به Look Up Table بسیار بزرگ ، و با توجه به شـرایط و نیـاز هـای پروژه ، مناسب است البته باید در نظر داشت هر بار که عدد جدیدی برای فرکـانس داده مـی شـود، شکل موج باید از نو ساخته شود. بنابراین Look Up Table باید درRAM ایجاد شود .

مشکلات کنترل فرکانس :

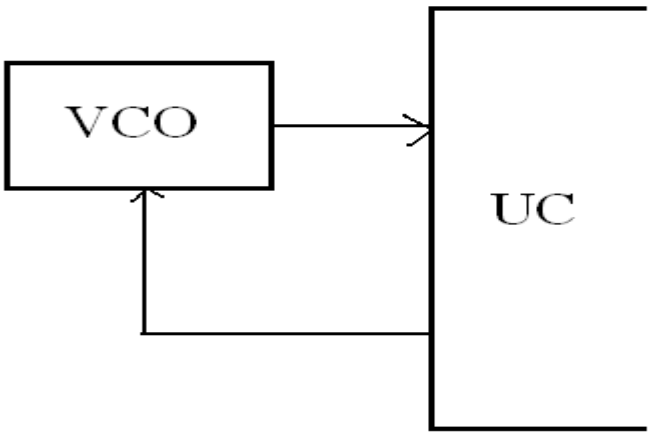
با توجه به اینکه تمام متغیرهای معادله ی t مقادیر صحیحی هستند، T، همواره ضـریب درسـتی از T_x خواهد بود. بنابراین فقط فرکانس هایی قابل حصول خواهند بود که نسبت صحیحی از F_x باشند و سایر فرکانس ها غیرقابل دسترس اند .شکل۱۰ – ۲ با ارائه ی قسمت کوچکی از نمودار شکل ۶ – ۲ این موضوع را به خوبی نشان می دهد.



شکل ۱۰ – ۲

آنچه باعث صحیح بودن نسبت F_x/F_o می شود، دیجیتالی بودن مدار است. مدارات دیجیتال صـرفاً فرکانس clock خود را بر عددی تقسیم می کند و. این عدد هرگز نمی تواند عدد غیرصحیحی باشد. به بیان دیگر، مدارات دیجیتال از ایجاد فرکانس بـا نسـبت غیرصـحیح از فرکـانس بایـد عاجزنـد. در اینجاست که باید مدار آنالوگ مناسبی طراحی و با قسمت دیجیتال کوپل شود.

یکی از راه حل های این شکل، استفاده از یک اسیلاتور با فرکانس قابل تنظیم (مثـل VCO) اسـت که مقدار فرکانس خروجی آن به وسیله ی قسمت دیجیتالی تنظیم می شود. (شکل ۱۱ – ۲)



شکل ۱۱ – ۲

عیب این طرح به علت عدم ثبات کامل فرکانس خروجی و نوسان ذاتی آن حول فرکانس تنظیم شده و نیز تاثیر پذیری از شرایط محیطی و رابطه ی غیر خطی بین فرکانس و پارامتر کنترل کننده و معلوم نبودن مقدار قطعی فرکانس به ازای یک ورودی خاص در این پروژه قابل استفاده نیست . روش دیگر استفاده از مدارات ضرب کننده یا جمع کننده ی فرکانس، فیدبک مثبت دیجیتالی و ... است که در حد این پروژه نمی گنجد. باتوجه به توضیحات فوق، در همین حد قناعت کرده و بحث کنترل فرکانس را در اینجا به اتمام می رسانیم.

چکیده فصل:

با جمع بندی همه ی روابط استفاده شده در این فصل در مورد زمان تناوب می توان گفت

$$F_o = \frac{F_x}{V \cdot N}$$ معادله ی ۱۲-۲

فصل ۳

کنترل دامنه

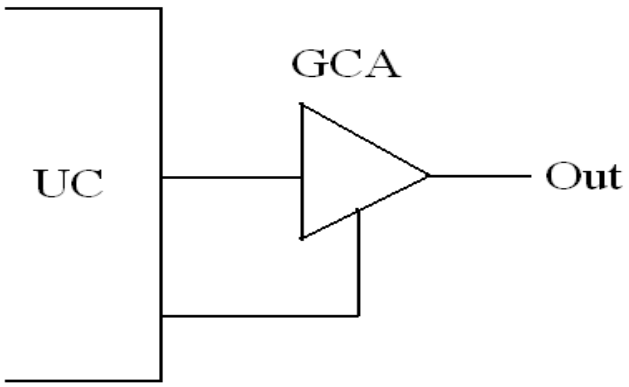
یکی از مهم ترین قابلیت های فانکشن ژنراتور ، که انعطاف پذیری بالایی بـه آن مـی دهـد ، تنظـیم دامنه ی سیگنال آن است . چرا که همه ی مواردی که فانکشن ژنراتور در آن کاربرد می یابد ، دارای دامنه ی یکسانی نیستند . از این رو بررسی نحوه ی کنترل آن در این پروژه الزامی است .

روش های کنترل دامنه :

با توجه به طرح کلی پروژه، دامنه ی سیگنال خروجی به دو صورت قابل کنترل می باشد.

۱. مقادیر عددی موجود در حافظه، قبل از اعمال به خروجی، به یک عدد مناسب و قابل کنتـرل ضرب شود تا دامنه ی مورد نظر را ایجاد کند. به علت محدودیت بزرگترین عدد باینری قابـل اعمـال به خروجی، و در نتیجه امکان ایجاد اعوجاج و تغییر شکل موج و فرکانس دراثر تغییرات دامنـه، ایـن روش با توجه به رنج مطلوب به تنهایی قابل استفاده نیست.

۲. شکل موجی که توسط نرم افزار تولید می شود. همواره دامنه ی ثابتی داشته باشد و دامنه ی خروجی، توسط یک مدار خارجی کنترل شود. (شکل ۱ – ۳)



شکل ۱ – ۳

در این پروژه از تلفیق این دو روش برای کنترل دامنه استفاده شده است . روش اول کاملا به صورت نرم افزاری اجرا شده و نیازی به سخت افزار خاصی نیست . محدودیت روش اول مارا ملزم به استفاده از روش ۲ ، همراه با روش ۱ ، می دارد . بنابر این باید مدار کنترل دامنه ی مناسبی طراحی و اجرا شود .

آنچه که معمولاً به عنوان کنترل کننده ی دامنه مطرح می شود تقویت کننده ای با گین متغیر می باشد. اما قابلیت ویژه ی DAC استفاده شده در این مدار ، توانایی کنترل دامنه ی خروجی را نیز به آن اضافه کرده است. با توجه به دیتاشیت DAC08 همواره :

$$I_{FS} = W \cdot I_R \qquad \text{معادله ۱ – ۳}$$

که در آن I_{FS} جریان Full scale خروجی و I_R جریان موجود در پایه ی ۱۴ یا ۱۵ و Wعدد دیجیتال ورودی است . بنابراین با کنترل I_R (که محدوده ای از صفر تا ۴mA را در بر می گیرد) می توان دامنه ی خروجی را کنترل کرد.

با توجه به اتصال مجازی بودن پایه های ۱۴ و۱۵ ، می توان با وصل یکی از این پایه ها به زمین ، جریان دیگری را با کنترل ولتاژ یک مقاومت کنترل کرد . و آنچه جهت ایجاد ولتاژ متغیر با کنترل دیجیتال مطرح میشود ،یک PWM و یک مدار متوسط گیر است . توضیحات بیشتر راجع به کنترل دامنه نیاز به بررسی مدار آن دارد که در فصل مربوطه ارائه خواهد شد .

فصل ۴

کنترل Offset

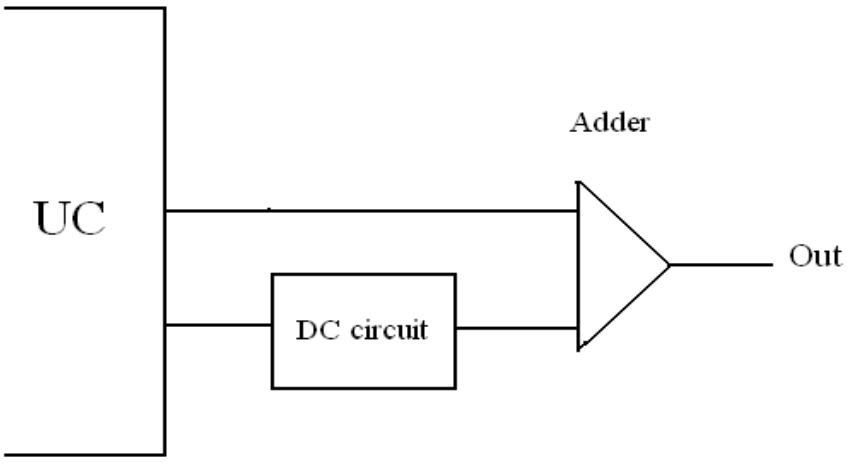
مقدار مولفه ی DC سیگنال خروجی فانکش ژنراتور به عنوان Offset مطرح می شود. کنترل این مولفه در برخی موارد مثلا بررسی پاسخ DC یا بررسی نقه ی بایاس و پایداری ترانزیستور و .. لازم است .

روش های کنترل Offset :

برای کنترل این عامل مل کنترل دامنه دو روش وجود دارد .

۱. شکل موجی که در میکرو شبیه سازی می شود ،قبل از اعمال به خروجی ، به صورت نرم افزاری با یک عدد جمع شود .

۲. شکل موج خروجی میکرو ،نوسط یک مدار جمع گر انالوگ ، با یک مقدار بخصوص جمع شود که این مقدار توسط میکرو کنترل مئ گردد . (شکل ۱ – ۴)



شکل ۱ – ۴

توضیحات بیشتر راجع به کنترل Offset را به بخش " مدار کنترل Offset " موکول می کنیم .

میانگفتار ۳

کنترل فاز

برخلاف دو عامل فرکانس و دامنه، مولفه ی فاز یک مولفه ی نسبی است. یعنی فاز یک سیگنال همواره نسبت به سیگنال مرجع هم فرکانس بیان می شود. بنابراین در یک سیگنال ژنراتور با یک خروجی نمی توان مساله ی کنترل فاز را امواج کرد.

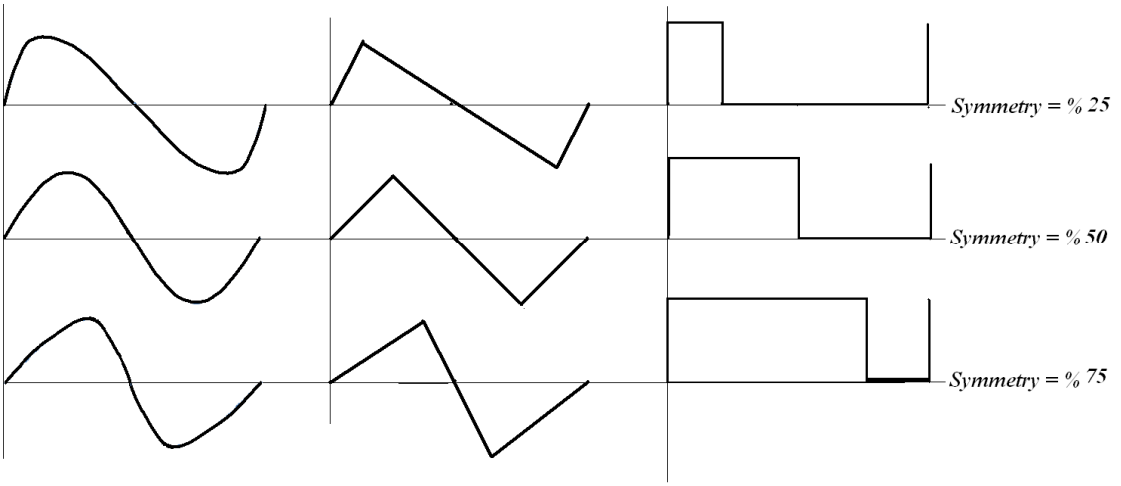
از طرفی ، علاوه بر سه عامل فرکانس، دامنه و فاز که یک سیگنال با یک شکل موج خاص را به طور کامل تعریف می کند مولفه های دیگری نیز هستند که بنابر کاربردهای خاص باید بتوان آن ها را تحت کنترل درآورد.در فصل های بعدی به معرفی و تشریح این عوامل می پردازد .

فصل ۵

کنترل Symmetry

Symmetry یا تقارن بیانگر برابری و همشکلی شیب مثبت و منفی یک شکل موج است در شکل موج هایی با ضابطه ی ناپیوسته مثل مربعی تقارن با عناوین دیگری چون ضریب اتصال، چرخه ی کار علامت- فاصله و ... شناخته می شود.

شکل ۱ – ۵ مقایسه ای بین شکل موج های متقارن و نامتقارن را نشان می دهد.

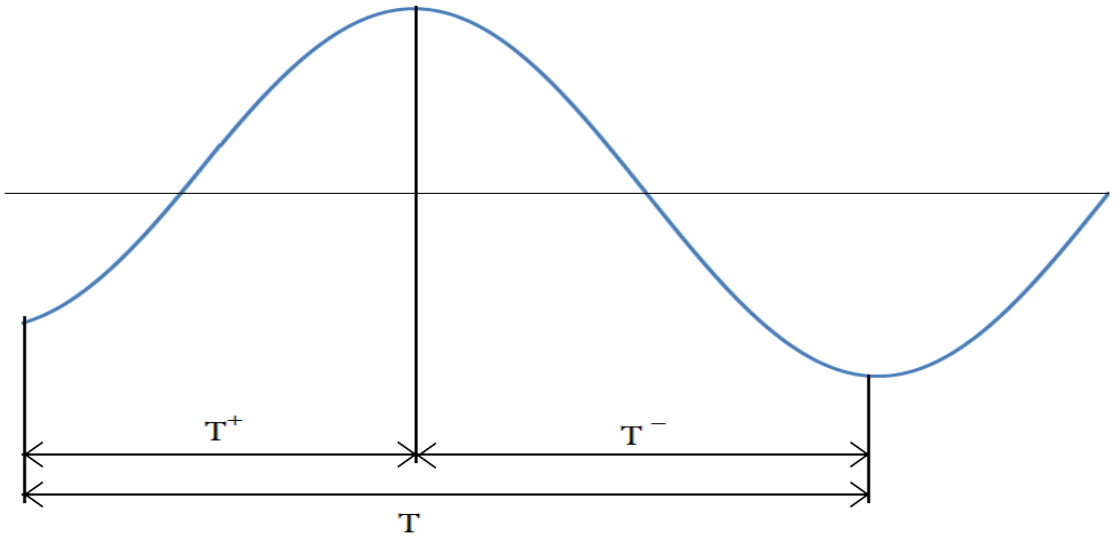


شکل ۱ – ۵

کنترل تقارن ، با توجه به هارمونیک های خاصی که وارد طیف فرکانسی می کند ، معمولا در بررسی باند فرکانسی و رفتار مدارها (الی الخصوص تقویت کننده ها و مدارهای سویچ) اهمیت می یابد .

معادلات Symmetry

تقارن را به زبان ریاضی به صورت نسبت زمان یکی از شیب ها (معمولا شیب مثبت) به کـل زمـان تناوب تعریف می کنند .(شکل ۲ – ۵)



شکل ۲ – ۵

$$S = \frac{T^{+}}{T}$$ معادله ۱ – ۵

با توجه به اینکه مقدار S همواره کوچکتر از واحد است ، معمولا آن را بر حسب درصـد بیـان مـی کنند . مثلا در مورد یک موج متقارن S=٪۵۰ است .

برا ی کنترل تقارن ، باید برای هر یک از شیب های سیگنال مورد نظر ، یک معادله جداگانه تعریف کرد . مثلا در مورد یک موج سینوسی :

$$v_t = \begin{cases} \sin(\omega^{+}t) & -\pi \le t \le \pi \\ \sin(\omega^{-}t) & \pi \le t \le \frac{3\pi}{2} \end{cases}$$ معادله ۲ – ۵

که در آن :

$$\omega^{+} = \frac{\omega}{S}$$ معادله ۳ – ۵

و

$$\omega^{-} = \frac{\omega}{1-S}$$ معادله ۴ – ۵

برای کنترل تقارن در این پروژه ، تنها کاری که باید انجام داد معادل سازی متغیرهای فرمول تقـارن به متغیر های پروژه است . یعنی باید $T \left\| N \right.$ و $T^{+} \left\| N^{+} \right.$ و $T^{-} \left\| N^{-} \right.$ قرار داد .

با توجه به توضیحات فوق بدیهی است که کنترل تقارن نیاز به سخت افزلر خاصی ندارد . زیر برنامـه ی بسیار ساده ی آن نیز در قسمت برنامه نویسی پروژه ارائه خواهد شد .

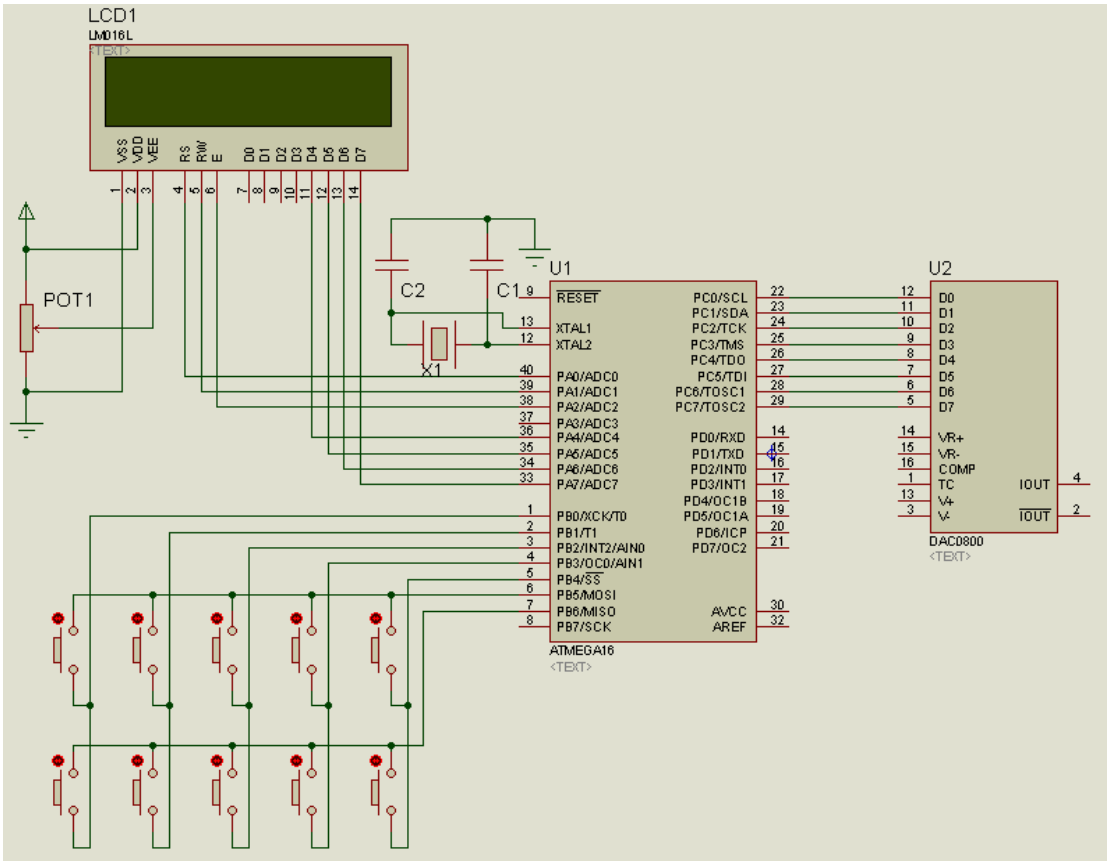
فصل ۶

سخت افزار پروژه

سخت افزار یک پروژه در واقع بستر اجرایی آن است . مدار مورد نظر برای پروژه ، شامل قسمت های مختلفی است که هر یک از آنها برای اجرای اهداف ذکر شده طرح شده است . در این فصل به بررسی مدار طراحی شده برای قسمت های مختلف پروژه و در نهایت طرح مدار کلی وارائه ی PCBمربوطه می پردازیم

مدار پایه :

شماتیک این مدار در نشان داده شده است .



شکل ۱-۶

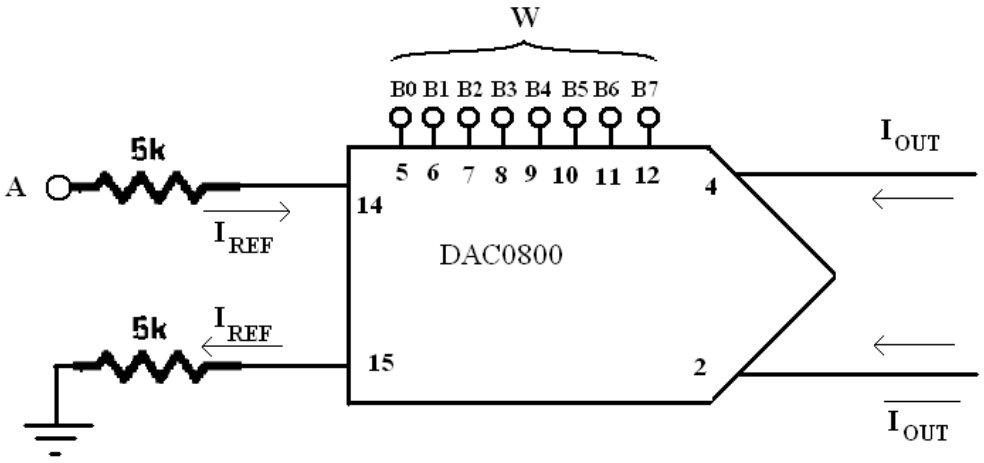
این مدار در واقع طرح پایه ی پروژه بوده و بقیه ی قسمت ها فقط مکمل مدار اند و طرح های متفاوتی می توان برای آن ارائه داد ، یا حتی آنها را حذف کرد . در این مدار میکرو کنترلر ATMEGA16 بستر اجرایی نرم افزار پروژه ،DAC0800 برای تبدیل مقادیر عددی شکل موج به سطوح ولتاژ ،LCD LM016 جهت اعلان مقادیر خروجی و Keypad نیز برای ایجاد تغییات در مقادیر خروجی و کنترل آنهاست . البته طرح های مختلفی برای Keypad قابل اجراست ولی با توجه به نیاز های پروژه و نیز سادگی کنترل , طرح ارائه شده در شکل ۱-۶ (که از نظر بازاری غیر استاندارد است)استفاده شده است .

نکات مهم

۱. همانور که ملاحظه می شود ، Port D بلااستفاده مانده است . بنابر این می توان آن را برای کنترل سایر قسمت های مدار استفاده کرد .
۲. پتانسیو متر Pot1 جهت کنترل کنتراست نمایشگر می باشد
۳. برای دست یابی به فرکنس های هرچه بالاتر باید فرکانس کریستال را تا حد ممکن بالا انتخاب شود . مقدار حدی فرکانس کریستال برای ATMEGA16 برابر 16MHz است . با توجه به Datasheet ارائه شده برای این میکرو ، در صورت استفاده از فرکانس بالاتر از 8 MHz ، نیازی به خازن های C1 ,C2 نمی باشد . همچنین فیوز بیت های میکرو کنترلر باید به صورت زیر برنامه ریزی شود .

مدار کنترل دامنه ی خروجی

با توجه به توضیحات ارایه شده در فصل ۳ دراین پروژه جهت کنترل دامنه باید مدار مناسبی طراحی شود . آنچه به عنوان کنترل کننده ی دامنه مطرح می شود ، معمولا یک تقویت کننده با گین قابل کنترل می باشد . اما DAC0800 به سادگی قابلیت کنترل دامنه ی خروجی مدار را در اختیار ما گذاشته و باعث عدم نیاز به تقویت کننده ی قابل کنترل می گردد . شکل ۲-۶ شماتیک اجرایی DAC0800 رانشان می دهد .



شکل ۲-۶

در این شکل جریان خروجی است که توسط عدد $W= B_0\ B_1\ B_2\ B_3\ B_4B_5\ B_6\ B_7$ و$I_{REF,}$ به صورت رابطه ی زیر قابل کنترل است.

$$I_{out} = I_{REF} \cdot W$$ معادله ی ۱-۶

با توجه به اینکه W در واقع مقادیر لحظه ای خروجی است ، با تغییر I_{REF} به راحتی می توان دامنه ی خروجی را کنترل کرد .

با توجه به شکل ۲-۶ ، I_{REF} مقدار جریانی است که از پایه ی ۱۴ یا ۱۵ جاری می شود .

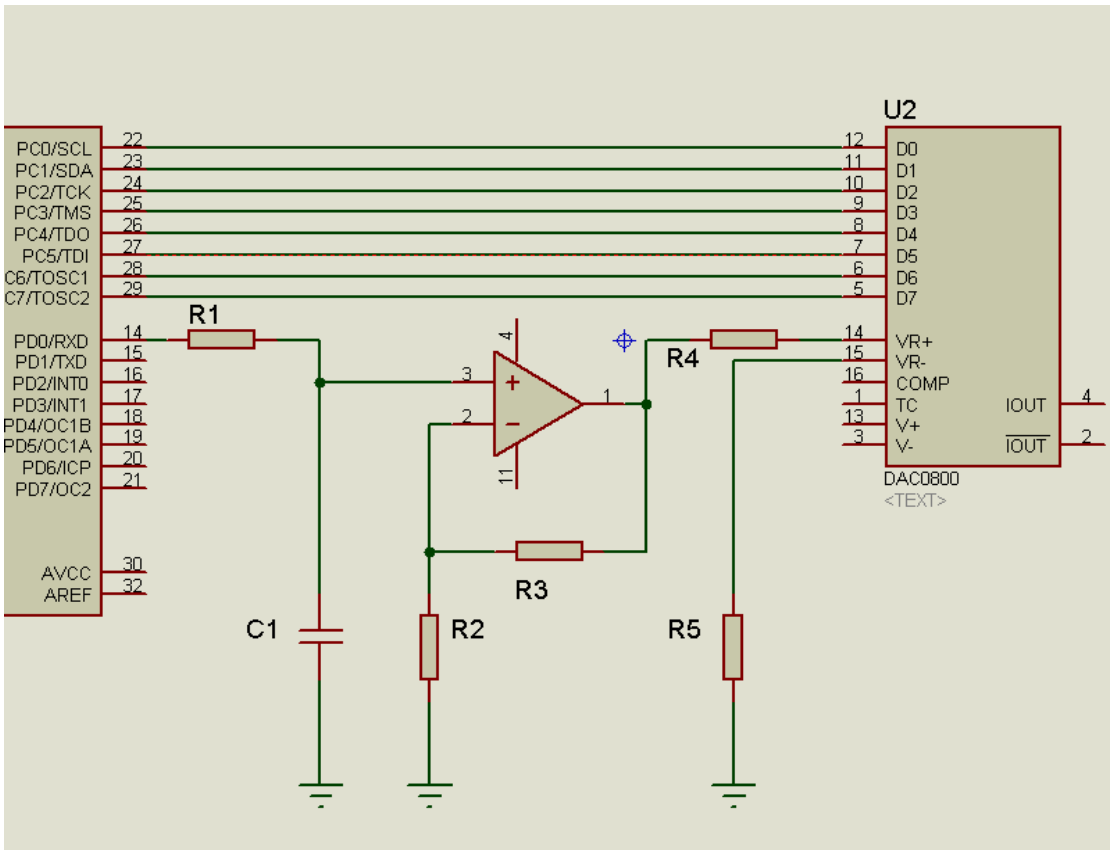
چون این دو پایه به صورت اتصال کوتاه مجازی عمل می کنند (مراجعه شود به فایل دیتا شیت DAC0800) ، با زمین کردن پایه ی ۱۵ پایه ی ۱۴ نیز رمین مجازی خواهد شد . بنابر این می توان گفت :

$$I_{REF} = \frac{V_A}{R_{REF}}$$

معادله ی 2-۶

پس با کنترل ولتاژ نقطه ی A ، I_{REF} و در نتیجه دامنه ی خروجی قابل کنترل خواهد بود . ایجاد یک ولتاژ قابل کنترل ، با ایجاد یک PWM توسط میکرو و متوسط گیری و تقویت آن توسط یک Op-Amp قابل اجرا خواهد بود .

توضیحات ارائه شده به مدار شکل ۳-۷ می انجامد .



شکل ۳-۷

دراین مدار R1 وC1 جهت تبدیل PWM به DC می باشد . با توجه به فرکانس کاری PWM که حدود 62 Khz است, مقدار R1 = 1K وC1 = 10n ثابت زمانی مناسبی ایجد خواهد کرد . مقدار R2 و R3 با وجه به نسبت حداکثر ولتاژ خروجی Op-Amp (برابر ولتاژ تغذیه است) به حداکثر ولتاژ دور C1 به دست می آید

با انتخاب Vcc = 10 V و د رمورد مقادیر R2 و R3 داریم :

$$\frac{V_{O\max}}{V_{\mathrm{Im}\,ax}} = \frac{R_2 + R_3}{R_2}$$

معادله ی ۳-۶

در نهایت :

$$R_2 = R_3$$

معادله ی ۴-۶

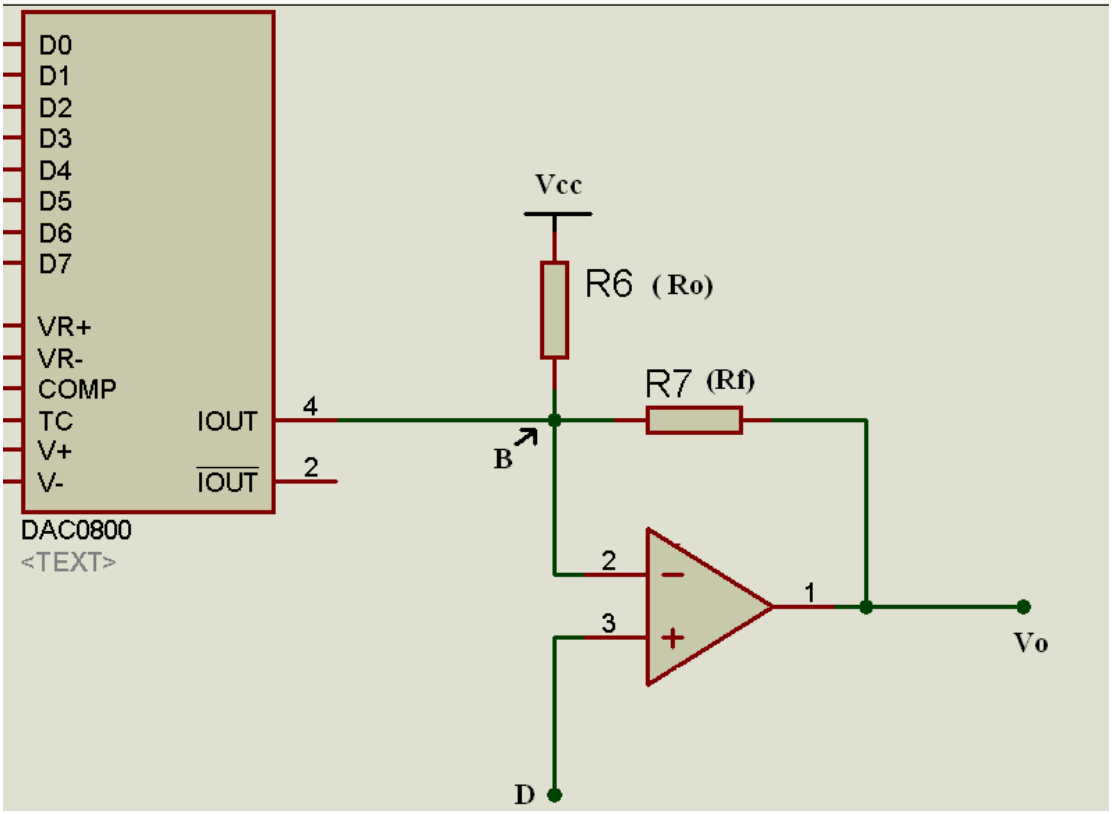
با توجه به جریان بایاس Op-Amp مقدار 1K برای R2 و R3 مناسب خواهد بود .

مدار کنترل Offset

همانور که در فصل ۴ اشاره شد ، ساده ترین راه کنترل Offset ، جمع کردن یک مقدار DC با سیگنال AC. است . شکل ۵-۶ مدار جمع گر مورد نظر را نشان می دهد.

مزیت این مدار، ایزوله بودن کامل دو ورودی نسبت به هم و در نتیجه عدم ایجاد اختلال درکارکردDAC می باشد. همچنین در ورودی DC ، مداری جهت تبدیل PWM به DC مورد نیاز می باشد که در قسمت کنترل دامنه بررسی شد.

مقادیر مقاومت ها بایدبه گونه ای انتخاب شوند که بتوان حداکثر دامنه ی موردنیاز در خروجی و حداکثر تغییرات DC خروجی را با توجه مقادیر موجود در ورودی ایجاد کرد.



شکل ۵-۶

با اعمال یک KCL در نقطه ی B داریم :

$$I_{OUT} + \frac{V_{cc} - V_D}{R_o} + \frac{V_O - V_D}{R_F} = 0$$

معادله ی ۵-۶

و با در نظر گرفتن $R = R_f = R_o$ می توان گفت :

$$V_o = R \cdot I_{OUT} + V_{CC} - 2V_D$$

معادله ی ۶-۶

چون I_{Out} شامل دو مولفه ی I_{AC} وI_{DC} است ، می توان گفت :

$$V_o = V_{AC} + V_{Ofs}$$

معادله ی ۷-۶

که در آن :

$$V_{Ofs} = R \cdot I_{DC} + V_{CC} - 2V_D$$ معادله ی ۶-۸

و با قرار دادن مقادیر متناسب هر مولفه داریم :

$$V_{Ofs} = V_{CC} + 5(\frac{DC}{255}) \cdot R\frac{V_A}{R_{REF}} - 2 \cdot (5\frac{DV}{255})$$ معادله ی ۶-۹

که در آن V_A ولتاژ نقطه ی A (در مدا ر کنترل دامنه) ، DC مقدار DC در Look Up Table ، DV مقدار PWM مربوط به DC ، وVcc ولتاژ تغذیه است

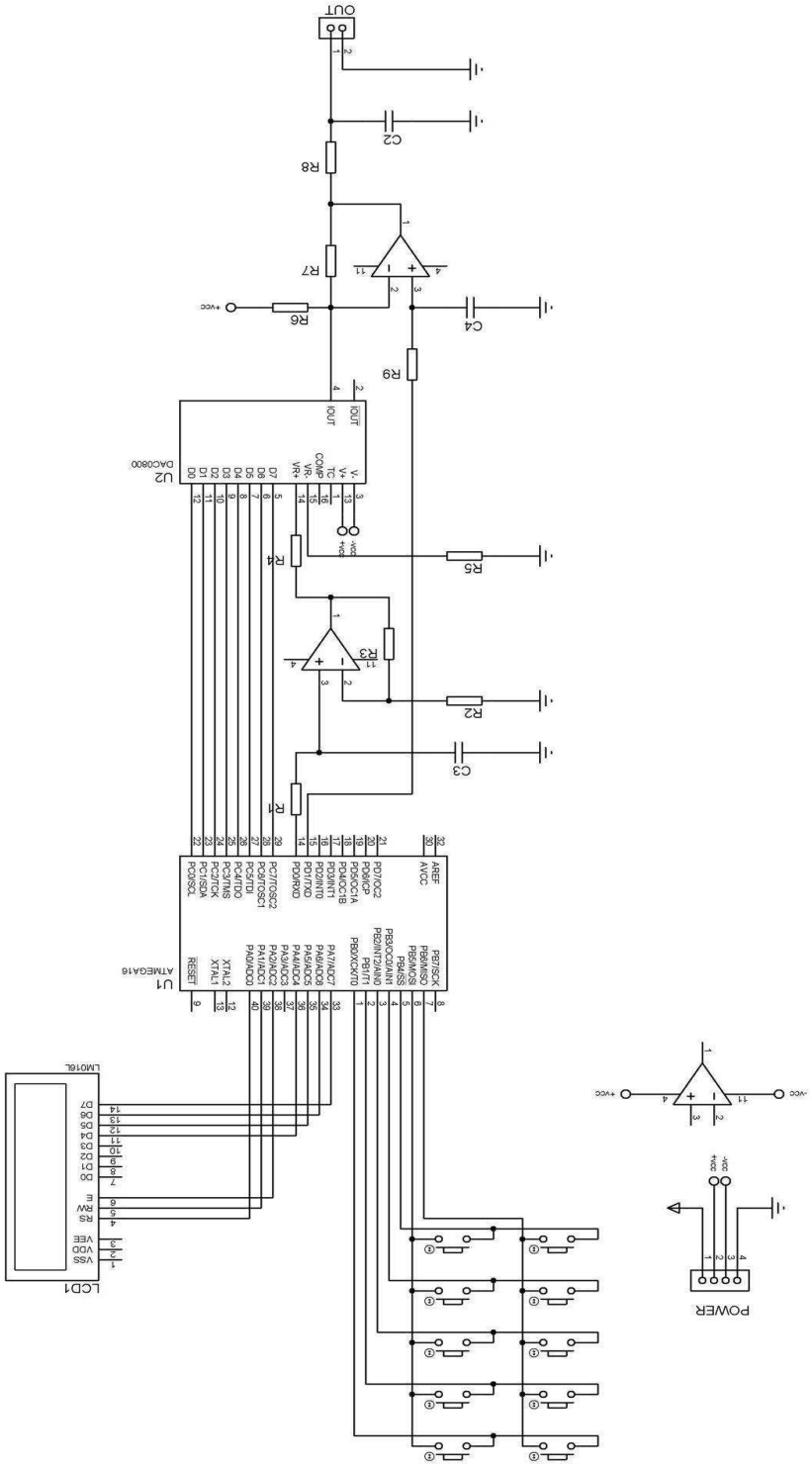
با فرض $R = R_{REF}$ و قرار دادن معادله V_Aدر معادله ی ۶-۹ داریم :

$$V_{Ofs} = 10 + \frac{1}{51} \cdot (DC \cdot \frac{AV}{51} - 2DV)$$ معادله ی ۶-۱۰

کنترل DV و DC جهت ایجاد Offset خواسته شده در قسمت زیر برنامه ی کنترل DC بررسی خواهد شد .

مـد ا ر کـلـی

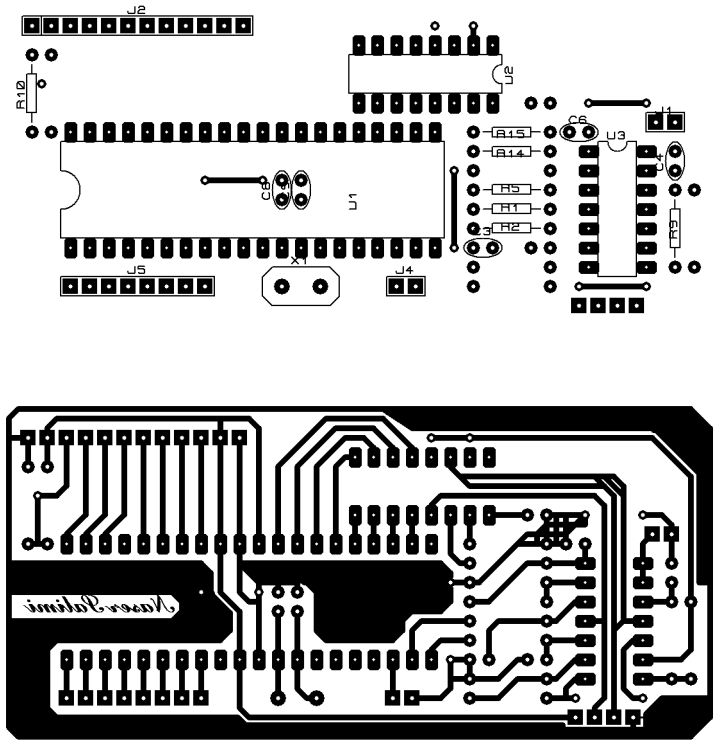
با جمع بندی شکل ها وشماتیک ها ی ارائه شده در قسمت های قبلی ، مدار کلی به صورت شکل ۶-۶ خواهد بود .


OUT
C2
R8
R7
R6
+VCC
C4
R9
U2
DAC0800
IOUT
R5
R4
R2
C3
R3
U1
ATMEGA16
LCD1
LM016L
POWER
-VCC

شکل ۶-۶

PCB :

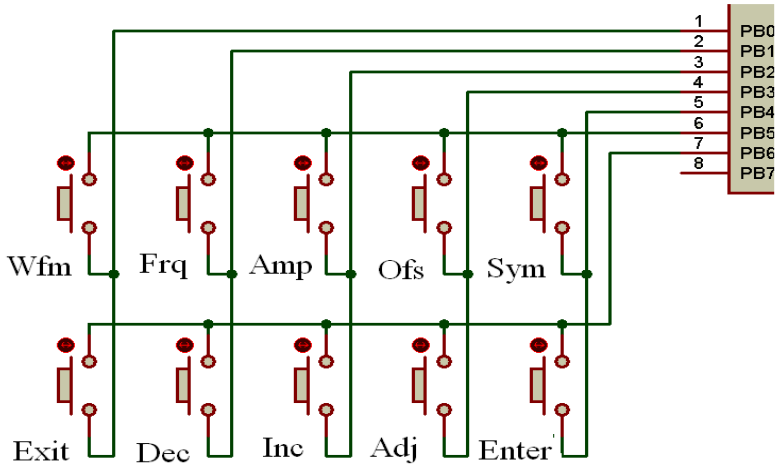
اجرای شماتیک شکل ۶-۶ در محیط Proteus و انتقال آن به محیط PCB ، به شکل زیر می انجامد



شکل ۶-۷

مدار keypad

با توجه به طرح غیر استاندارد مدار keypad ونیز جهت معرفی ورودی مربوطه ، مدار آن را در شکل زیر ارائه می دهیم



شکل ۸-۶

فصل ۷

نرم افزار پروژه

معادله ی کامل ولتاژ خروجی

هدف نهایی یک فانکشن ژنراتور ، ایجاد سیگنالی با مشخصات مورد نظر می باشد . این سیگنال هرچه باشد ،به هر حال یک تابع ریاضی به صورت ... است و تعریف شکل موج های قابل تئلید در این فانکشن ژنراتور نقطه ی شروع برنامه نویسی ماست .

این پروزه قادر به تولید ۵ شکل موج به شرح زیر است :

۷-۱ سینوسی :
$$v = Ofs + A \cdot Sin(2\pi \frac{n}{N})$$

۷-۲ مربعی :
$$v = Ofs \begin{cases} +A & | n < N/2 \\ -A & | n \geq N/2 \end{cases}$$

۷-۳ مثلثی :
$$v = Ofs + \begin{cases} An & | n < N/2 \\ A(1-n) & | n \geq N/2 \end{cases}$$

۷-۴ پالسی :
$$v = Ofs \begin{cases} +A & | n = 0 \\ -A & | n \neq 0 \end{cases}$$

۷-۵ دندان اره ای :
$$v = Ofs \begin{cases} +An & | n < N/2 \\ -An & | n \geq N/2 \end{cases}$$

این معادلات توابع ریاضی بوده و باید جهت مشخصات پروژه و سخت افزار تغییاتی به صورت زیر در آنها ایجاد شود .

۷-۶ سینوسی :

$$v=\begin{cases} DC+AC\cdot Sin(2\pi\frac{N2}{N1}n) & n\le N2 \\ DC+AC\cdot Sin(2\pi\frac{N1-N2}{N1}n) & n>N2 \end{cases}$$

۷-۷ مربعی :

$$v=\begin{cases} DC+AC & n\le N2 \\ DC+AC & n>N2 \end{cases}$$

۷-۸ مثلثی :

$$v=\begin{cases} DC+AC\cdot n & n\le N2 \\ DC+AC\cdot(1-n) & n>N2 \end{cases}$$

۷-۹ پالسی :

$$v=\begin{cases} DC+AC & n=0 \\ DC-AC & n\neq 0 \end{cases}$$

۷-۱۰ دندان اره ای :

$$v=\begin{cases} DC+AC\cdot n & n\le N2 \\ DC-AC\cdot n & n>N2 \end{cases}$$

متغیر های برنامه

همانطور که در بخش قبل گفته شد ، برنامه بر اساس توابع ریاضی خواهد بود . هر تابع ریاضی نیز شامل تعدادی متغییر است . در دنیای برنامه نویسی به طور کلی دو نوع متغیر وجود دارد :

۱. Global Variable : متغیر هایی که در همه ی قسمت های برنامه قابل فراخوانی و تغییر است و به اصطلاح همه ی زیر برنامه ها آن را می شناسد

۲. Local Variable : متغیر هایی که فقط در یک زیر برنامه شناخته می شود.

Local Variable مربوط به هر قسمت در فصل مختص آن توضیح داده خوهند شد. در اینجا Global Variable های استفاده شده در برنامه معرفی می شوند

متغیر های غیر رشته ای			
نام	نوع	آدرس	توضیحات
N1	unsigned char	R5	Look Up Table Value (Frq)
N2	unsigned char	R4	Look Up Table Value (Frq)
n	unsigned char	160	Accumulator Step Counter (Frq)
V1	unsigned int	R7:R6	Accumulator Value (Frq)
V2	unsigned int	R9:R8	Accumulator Value (Frq)
v	unsigned int	161	Accumulator Value Counter (Frq)
AV	unsigned char	R11	PWM value (Amp)
AC	unsigned char	R13	Look Up Table Value (Amp)
DV	unsigned char	R10	PWM value (Ofs)
DC	unsigned char	R12	Look Up Table Value (Ofs)
Type	unsigned char	163	Parameter Selector (Keypad)
Wfm_cnt	unsigned char	182	Current Waveshape Mark (Keypad)
num	long int	164	Inserted Number (Keypad)
minus	bit	R2.0	Minus Number (Keypad)

متغیر های رشته ای			
نام	نوع	آدرس	توضیحات
wave [256]	unsigned char	200	Look up Table
num_st[6]	unsigned char	178	Current Number Characters
Wfm_name_cnt[4]	unsigned char	17E	Current Waveshape Name
cnt [4]	long int	168	Current Values
pnl_buff[32]	unsigned char	183	LCD

define Regesters	
kin	PINB
kout	PORTB
kreg	DDRB
DAC	PORTC
OUT	PORTC

keys marking	
Wfm	1
Frq	2
Amp	3
Ofs	4
Sym	5

waves marking	
Sin	1
Sqr	2
Trg	3
Pls	4
Rmp	5

کنترل فرکانس

بررسی پارامتر V و N ومتغیر های V1 و V2و N1وN2

با توجه توضیحاتی که در فصل ۲ ارائه شد ، بنابر استفاده از متغیر ۱۶ برای V ، نیازی به استفاده از متغیر R نمی باشد .

هرچند که تعریف Look Up Table در فضای RAM باعث ساده تر شدن روند برنامه نویسی و اجرا شده و نیز حجم ROM مورد نیاز برنامه را به طور قابل توجهی کاهش می دهد ، اما حجم محدود RAM خود باعث ایجاد محدودیتهای جدیدی می شود . در DDS استاندارد ، با توجه به حجم بالای ROM مربوط به Look Up Table (معمولا 2^{16}) ، تقریبا هر فرکانسی را می توان با دقت بالا ($\pm 2^{-16}$) از ضرب دو عدد N, V به دست آورد .

اما در این پروژه ، عدم ثبات N از یک طرف و ظرفیت پایین Look Up Table (برابر 2^{8}) از طرف دیگر ، باعث از دست رفتن بسیاری از فرکانس ها می شود .

یک روش ساده برای بازگرداندن مقادیر، استفاده از سیستم زمانبندی دوگانه است . برای درک بهتر قضیه ، عدد T را طوری در نظر بگیرید که با هیچ Nو V صحیحی نتوان آن را ایجاد کرد (مثلا یک عدد اول) در این حالت می توان گفت :

$$\frac{F_x}{N \cdot V} < F < \frac{F_x}{(N+1) \cdot V}$$

معادله ی ۶-۷

در این صورت می توان F را به روش زیر ایجاد کرد :

$$F = \frac{F_x}{N \cdot V_1 + V_2}$$

معادله ی ۷-۷

که در آن همواره :

$$V_2 = \frac{F_x}{F} - N \cdot V_1$$

معادله ی ۸-۷

به عبارت دیگر ، بزگترین عدد ممکن برای N و V1که فرکانس حاصله ی آن کوچکتر از F باشد ،برای آنها انتخاب شده و باقیمانده آن توسط V2 ایجاد می شود . البته در نظر داشته باشید که برای حفظ تقارن شکل موج ،V2 باید در دو پله اعمال شود ،یکی در شیب مثبت و دیگری در شیب منفی . مقدارV در هر کدام از این پله ها نصف V2 است .

برنامه ی ما باید بتواند مقادیرV1 و V2وN را بر حسب حسب فرکانس به محاسبه کند. با مرتب کردن معادله ی ۸-۷ بر حسب فرکانس مورد نظر داریم:

$$N \cdot V_1 + 2V_2 = \frac{F_x}{F}$$

معادله ی ۹-۷

در اینجا نسبت $\frac{F_x}{F}$ برابر تعداد ماشین سیکل هایی است که یک سیکل خروجی را تشکیل می دهند ، بنابر این :

$$mc = N \cdot V_1 + V_2$$

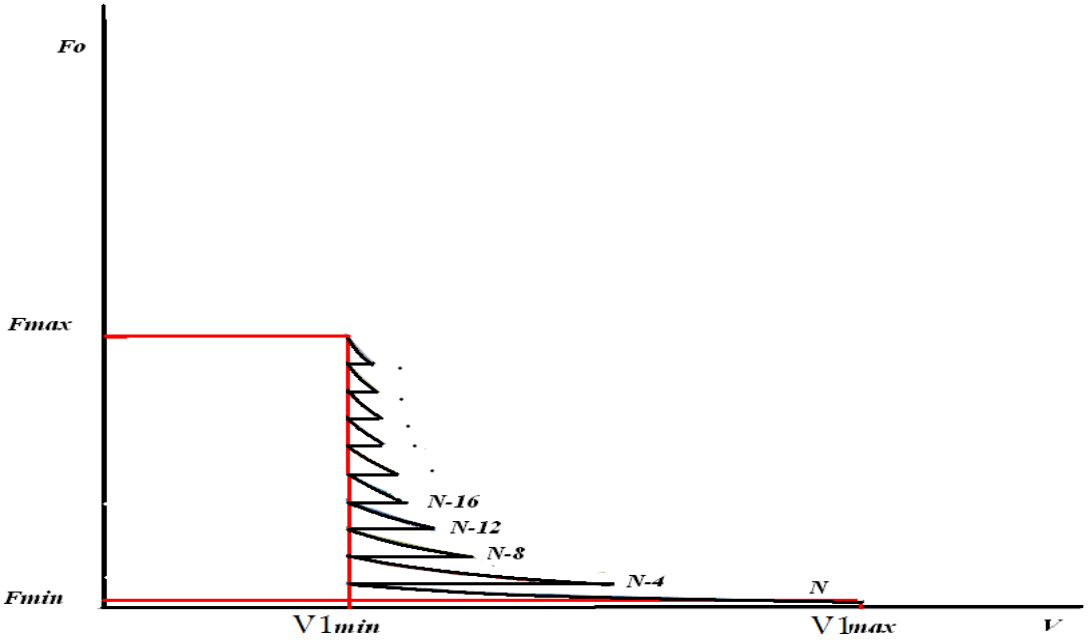
معادله ی ۱۰-۷

بدیهی است که مقادیر مختلفی را می توان برای V_2 وV_1و N انتخاب کرد ولی همواره بهترین گزینه ، انتخاب بزرگترین عدد ممکن برای N است که باعث ایجاد خروجی با کمترین THD. می شود این انتخاب بدین ترتیب است که در ابتدا N در بیشترین عدد قابل اعمال (N=254)قرار می گیرد . سپس مقدار V_1 در داخل یک حلقه محاسبه می شود .اگر این مقدار خارج از رنج خود باشد عدد ۴ از N کم شده وبرنامه به ابتدای حلقه باز می گردد .این روند آنقدر تکرار می شود تا V_1 در رنج خود($0x1F \le V_1 \le 0xFFFF$) قرار گیرد .البته با

توجه به رنج کاری مدار ، مقدار V_1 هرگز به حد بالایی خود نمـی رسـد ($V_1 \le \frac{Fx}{N_{\max} \cdot F_{\min}}$) .

بنابر این نیازی به بررسی محدوده ی بالایی آن نیست. V_2 نیز طبق معادله ی ۸-۷ محاسبه مـی گردد . البته اگر مقدار آن از حداقل ممکن کمتـر باشـد ، V_1 یـک واحـد کـاهش یافتـه و V_2 (و نیز V_1 و N در صورتی که از رنج خود خارج شده باشند) دوباره محاسبه می شـود . ایـن رونـد انقدر ادامه پیدا می کند تا V_1 ، V_2 و N ، هر سه در مقادیر مناسب خود قرار گیرند . البته در این قسمت محدودیت N ($6 \le N \le 256$) در نظر گرفته نشده اسـت ، بنـابر ایـن مقدار F نباید از Fmax بیشتر باشد .($F_{\max} = \frac{Fx}{N_{\min} \cdot V_{\min}}$).

نکته ی جالب توجه اینست که در این روش تعیین مقادیر ، تغییـرات N بـه ازای فرکـانس هـای نزدیک حداقل و تغییرات V_1 حداکثر است . شکل 7-1 محدوده ی تغییرات این متغیر ها را نشـان می دهد .



شکل 7-1

همانطور که شکل نیز نشان می دهد ، متغیر V_1 معمولا در مقادیر پایین قـرار دارد ،زیـرا N در مقادیر بالا قرار گرفته است . چون متغیر V_2 صـرفا جهـت رفـع کمبـود V_1 اسـت ،تغییـرات آن نامرتب می باشد .

نکته ی آخر ودر مورد متغیر N_2 می باشد . قبلا نیز گفتیم که V_2 باید در دو ناحیـه از شـکل موج اعمال شود و برای حفظ تقارن ، فواصل اعمال آن باید مرتب باشـد .از طرفـی محاسـبه ی ایـن فواصل وقت گیر بوده ودر صورتی که در تایمر (یا حلقه ی اصلی) قرار گیرد باعث افزایش V_0 و در نتیجه کاهش حداکثر فرکانس خروجی می شود . بنابر این نقاط اعمال V_2 از قبل محاسـبه شـده ودر تایمر تحت عنوان مورد N_1 و N_2 استفاده قرار می گیرد .

زیر برنامه ی Frequency

همه ی آنچه که در بالا ذکر شد ، در برنامه ی زیر خلاصه می شود :

```
void edit_freq ( float F )
 {
  float MCt ;
  unsigned int Nt  = 258 ;
  V1 = 0 ;
  V2 = 0 ;
  MCt = 16000000 / F ;
   #asm ("C : " )
  while ( V1 < 64 )
   {
    Nt -= 4 ;
    V1 = MCt / Nt ;
   }
 V2 =  MCt - Nt * V1  ;  // = MCt % Nt ;
V2 /= 2 ;
 if ( V2 & V2 < 64 )
   {
    V1 -- ;
    #asm ( " jmp C " ) ;
   }
   N1 = Nt ;
}
```

Accumulator

نقش Accumulator در یک استاندارد ایجاد یک افزایش پیوسته و متناسب با فرکانس ،و اعمال آن به Look Up Table است . Over Flow همواره در عدد معینی صورت گرفته و طول زمانی هر پله همواره ثابت است اما در این پروژه و عملکرد Accumulator تا حدود ی متفاوت است . بدین ترتیب که افزایش ، همواره فقط یک واحد است . عدد سرریز بسته به فرکانس متغیر است و طول زمانی پله ها ثابت نیست .

وقتی زیر برنامه ی Accumulator فراخوانده شد ،بعد از قرار دادن مقادیر ورودی ها ، باید اولین عدد Look Up Table را در خروجی قرار دهد ، شمارنده ی پله را یک واحد افزایش دهد ، اگر شمارنده ی پله برابر شدN1 آن را صفر کند وبه تعدادV2 پله صبرکند (یا کار دیگری انجام دهد) ، اگر شمارنده ی پله برابر شد آن را به تعداد N2پله صبرکند (یا کار دیگری انجام دهد) ، برای تمام مقادیر غیر از N1 و N2 به تعداد V1 پله صبر کند .ورودیKeypad را چک کند و در صورت وجود تغییر در آن از برامه خارج شود ودر غیر این صورت به ابتدای برنامه باز گردد

نکات مهم :

۱. هرچه طول برنامه ی Accumulator کوتاهتر باشد ، V_0 کاهش ودر نتیجه حداکثر فرکانس خروجی افزایش خواهد یافت .
۲. زبان Assembly به علت ایجاد کد های کوچکتر بر سایر زبان ها ارجح تر است
۳. با توجه به اینکه روال اجرای وقفه خود نیاز به چندmc برای فراخوانی دارد ، بهتر است از وقفه ی تایمر برای Accumulator استفاده نشود
۴. در صورت استفاده از حلقه ی اجرایی دائم برای Accumulator باید همه ی وقفه ها غیر فعال شوند تا باعث به هم ریختن فرکانس و شکل موج نشوند .

زیر برنامه ی Accumulator

تمام آنچه ذکر شد به زبان برنامه نویسی C زیر به شکل زیر است :

```
kreg = 0x1F ;
kout = 0x1F ;
while ( ! kin.5 )
{
DAC = wave [ n ] ;
n ++ ;
v = V1 ;
if ( n == N1 )
{
n = 0 ;
v = V1 ;
}
if ( n == N2 )
v = V2 ;

while ( v )
v -- ;
 }
```

وبه زبانAssembly :

(البته بنابر لزوم ایجاد PWM که درزیر برنامه ی دامنه توضیح داده خواهد شد در زیر برنامه ی Accumulator کمی تغییرات ایجاد شده است)

```
void accumulator  ( void )
{

#asm
      LDI R30 , 0x1F
      OUT 0x17 , R30
      OUT 0x18 , R30

      LDI R30 , 0x3
      OUT 0x16 , R30
      MOVW R24 , R28
      CLR R30
      LDI R31 , 2

Began :
      SBIC 0x16 , 5        ; while (! kin.5)
      RJMP LoopEnd
      LD   R0,Z           ; DAC =wave [ n ]
        OUT  0x15,R0        ;
      INC R30            ;n++
      MOVW R28 , R6        ; v = V1
      CP R30 , R5         ;if(n == N1)
      BRNE Loop1
      CLR R30            ;n=0
      MOVW R28 , R8        ;v = V2

      CP R30 , R4         ;if(n == N2)
      BRNE Loop1
      MOVW R28 , R8        ;v = V2
Loop1 :
      SBIW R28 , 1

      CPI R29 , 0
      BRNE PWM1
      CBI 0X15 , 0
      CBI 0X15 , 1
      JMP Began
PWM1 :
      CP R29 , R11
      BRNE PWM2
      SBI 0X15 , 0
PWM2 :
      CP R29 , R10
      BRNE LOOP1
      SBI 0X15 , 1

LoopEnd :
      CLR R31
      CLR R30
      MOVW r28 , R24
      OUT  0x15,R30

#endasm
```

Symmetry

همانطور که قبلا نیز گفته شد Symmetry عددی بر حسب درصد است . بنابر این اولین گام خارج کردن آن از حالت درصدی است .

همچنین با داشتن یکی از مقادیر S ، N^+ یا N^- ، می توان سایر آنها را به دست آورد . با توجه به اینکه محاسبه N^+ از S مسیر طولانی تری می طلبد و نیز تکرار این محاسبات در جاهای مختلف مناسب نمی باشد ، وهمچنین حجم کمتر N^+ نسبت به S و وجود متغیر N_2 که با N^+ متناسب است ، تعیین مقادیر بر حسب این متغیر ارجح است .

در اینجا N^+ معادل N_2 در نظر گرفته شده است .

زیر برنامه ی کنترل Symmetry

```
void edit_sym ( long int S )
{
N2 = N1 * S / 100 ;
}
```

Amplitude

با توجه به توضیحاتی که در فصل ۳ و فصل ۶ ارایه شد ، کنترل دامنه در ۲ مرحله صورت می پذیرد . مرحله ی نرم افزاری آن که با تغییر عدد A ،مربوط به Look Up Table (که ما در اینجا آن را AC نامیده ایم) و مرحله ی کنترل به کمک سخت افزار و تغییر عدد PWM (در اینجاAV) ،دامنه ی خروجی به طور کامل و با Resolution بسیار بالا (برابر ۲^ ۱۵)کنترل می شود .

چون هر دو مقدار AC و AV دارای محدوده ی تغییرات وسیعی می باشند ، اتنخاب مقدار برای آنها ،جهت ایجاد دامنه ی خواسته شده بسیار آسان خواهد بود .تنها محدودیت در مورد متغیر AC است که مثل نظیر فرکانسی آن در Look Up Table (یعنی همان N) هر چه بزرگتر باشد بهتر است . و هرگز نباید از ۳ کمتر شود بنابر این درست مثل حالت تعیین فرکانس بیشترین مقدار را برای AC انتخاب کرده و نسبت باقیمانده را توسط AV ایجاد می کنیم . نحوه ی این انتخاب در بخش فرکانسی مطرح شد و نیازی به تکرار دوباره ی آن نیست

زیر برنامه ی کنترل Amplitude

برنامه ی کنترل دامنه ، به توجه به توضیحات فوق ، به شرح زیر می باشد .

```
void edit_amp ( long int A )
 {
  AC = A *0.0126 ;
  AC ++ ;

  if ( AC <= 10 )
   AC = 10 ;

  AV = ( A * AC ) / 1000000 ;
 }
```

Offset

نحوه ی کنترل Offset دقیقا مثل کنترل دامنه می باشد ، با این تفاوت که اولا Offset دارای مقادیر مثبت و منفی است و ثانیا در فرمول عدد ثابت (Vcc) وجـود دارد . همچنـین بـر خـلاف دامنه مقدار آن می تواند صفر شود .

زیر برنامه ی کنترل Offset

توضیحات ارائه شده در مورد کنترل Offset ما را به زیر برنامه ی زیر رهنمون می سازد

```
void edit_ofs ( long int D )
 {
 DC = D * 0.0127 ;
 DC ++ ;

 if ( minus )
  DC  = 127 + DC ;

 else
  DC  = 127 - DC ;

 }
```

ایجاد شکل موج ها

دلیل اینکه این مبحث تا اینجا به تعویق افتاد این است که با توجه به فرمول های ارائه شده در اول فصل ، تمام متغیر هایی که تا اینجا بررسی شدند ،در آن فرمول ها استفاده شده بودند و طبیعی است که بدون شناخت کامل آنها و تعیین روشی مناسب برای تعیین آنها ، این مبحث گنگ و نامفهوم خواهد بود . حال در اینجا به معرفی زیر برنامه های مربو به زیر برنامه ی ایجاد شکل موج ها شکل موج ها می پردازیم این زیر برنامه ها در واقع کاری به جز قرار دادن مقادیر معادل پارامتر های ریاضی در فرمول و اجرای نرم افزاری آن انجام نمی دهد

زیر برنامه ی ایجاد شکل موج ها

برای شکل موج سینوسی :

```
void wave_sin (void)
 {
  unsigned char Np  ;
  float w  ;
  unsigned char value ;

  N1 ++ ;
  Np = N2 / 2 +1 ;
  w = 1.57 / Np ;

  for( n = Np ; n > 0 ; n -- )
   {
    value = AC * sin ( w * n ) ;

    wave [ n ] =
    wave [ N1 - n ] = DC - value;
   }

 Np = ( N1 - N2 ) / 2   ;
 w = 1.57 / Np ;

  for( n = Np ; n < 2 * Np ; n++ )
   {
    value = AC * sin ( w * n ) ;
    wave[ n ] = DC + value ;
    wave[ N1 - n ] = DC - value ;
   }

N1 -- ;
 }
```

و برای شکل موج مربعی:

```
void wave_sqr (void)
 {
  unsigned char Np  ;

  Np = N2 / 2  + 1 ;

  for( n = 1 ; n < Np ; n++ )
   wave[ n ] = DC + AC ;

  for ( n = N1 ; n > Np ; n-- )
   wave[ n ] = DC - AC ;
 }
```

برای شکل موج مثلثی:

```
void wave_trg (void)
 {
 unsigned char Np  ;
  float w  ;

  N1 ++ ;
  Np = N2 / 2 + 1 ;
  w = 1 / Np  ;

  for( n = Np ; n > 0  ; n++ )
   {
    wave[ n ] = DC + AC * w * n ;
    wave[ N1 - n ] = DC - AC * w * n ;
   }
  for ( n = Np ; n < 2 * Np ; n-- )
   {
   wave[ n ] = DC + AC * w * n ;
   wave[ N1 - n ] = DC - AC * w * n ;
   }
   N1 -- ;
 }
```

برای شکل موج پالسی :

```
void wave_pls ( void )
{
wave[ 0 ] = DC + AC   ;
for( n = 1 ; n <= N1  ; n++ )
  wave[ n ] = DC - AC ;
}
```

برای شکل موج دندان اره ای :

```
void wave_rmp (void)
{
 float w  ;

   N1 ++ ;
 w = 1 / N1  ;

 for(n = 0 ; n <= N1 ; n++ )
  wave[ n ] = AC * n ;

 N1 -- ;
}
```

زیر برنامه ی تعیین شکل موج

در این برنامه با توجه به شکل موج خواسته شده ، زیر برنامه ی ایجاد آن فراخوانی و اجرا می شود

```
void edit_wfm ( unsigned char W )
 {
    switch ( W )
     {
      case 1 :
       wave_sin() ;
       break ;

      case 2 :
      wave_sqr() ;
       break ;

      case 3 :
       wave_trg() ;
       break ;

      case 4 :
       wave_rmp() ;
       break ;

      case 5 :
       wave_pls() ;
       break ;
     }
  Wfm_cnt = W ;
 }
```

اصلاح Look Up Table

ضرورت وجود V2 در ابتدای فصل مورد بررسی قرار گرفت . V2 فقط در مقدار فرکانس خروجی اثر می گذاشت و ثاتیری بر Look Up Table نداشت . با توجه به اینکه وارد شدن این متغیر باعث افزایش تعداد پله ها به مقدار ۲ واحد می شود ، در صورت عدم تطبیق Look Up Table با این افزایش شکل موج خروجی به هم خواهد ریخت . وارد کردن این متغیر در فرمول شکل موج ها نیز بیشتر باعث بغرنج شدن مسئله شود .

یک راه حل ساده برای این موضوع Look Up Table اصلاح است . بدین ترتیب که در صورت وجود V2 دو پله به Look Up Table اضافه شود . یکی از این پله ها باید در شیب مثبت و دیگری در شیب منفی باشد تاتقارن به هم نخورد . با توجه به دامنه ی متغیر شکل موج در Look Up Table بهترین نقطه برای اعمال این دو پله ، نقطه های گذر از صفر است .

با توجه به اینکه پله ی شماره ی ۰ در هیچ کدام از شکل موج ها استفاده نشده ، برای ساده کردن برنامه ی Accumulator و استفاده ی هرچه بیشتر از رجیستر ها ، این برنامه هم ی مقادیر را (اگر V2 موجود نباشد) یک پله به عقب انتقال می دهد و اگر موجود باشد پله ی شماره ی صفر و شماره ی N2 را برابر مقدار کرده و N1 را یک واحد افزایش می دهد

زیر برنامه ی اصلاح Look Up Table

```
 void edit_LookUpTable ( void )
{
  if ( V2 )
  {

 for ( n = N1 ; n > N2 ; n-- )
   wave [ n + 1 ] = wave [ n ] ;

   wave [ 0 ] = DC;
   wave [ N1 / 2 + 1 ] = DC ;

   N1 ++ ;
  }

  else
  {
   for ( n = 0 ; n < N1 ; n++ )
   wave [ n  ] = wave [ n+ 1 ] ;

   N1 -- ;
   V2 = V1 ;
  }

  V1 -= 25 ;
  V1 /=3 ;
}
```

زیر برنامه های مر بوط به ورودی ، خروجی و زیربرنامه های اصلی

با توجه به اینکه ورودی پروژه Keypad بوده و طرح های متفاوتی می توان برای آن ارائه داد ، نرم افزار مربوط به آن نیز متفاوت خواهد بود . Keypad استفاده شده در این پروژه دو ردیف پنج تایی از سویچ های 5mm است که ردیف اول برای تعیین عملیات و ردیف دوم برای تغییر مقادیر استفاده می شود (شکل ۸-۶)

زیر برنامه های مربوط به خروجی

تمام زیر برنامه هایی که مربوط به می شوند به عنوان زیر برنامه ی خروجی مطرح اند .

زیربرنامه های اصلی

زیر برنامه های اصلی جهت ایجاد حالت پیش فرض و اجرای کل زیر برنامه هایی است که تا اینجا معرفی شده اند .

چون این قسمت ها جزو پیکره ی اصلی طرح نبوده و به هر شکلی قابل برنامه نوسیسی و اجرا هستند ، از توضیحات مربوط به آن صرف نظر و فقط به ارائه ی زیر برنامه های مربوط به آن بسنده می کنیم .