دانشکده برق و رباتیک

پایان نامه کارشناسی ارشد برق - کنترل

طراحی و ساخت شبیه سازسخت افزاری برای سیستمهای خطی و غیرخطی چندورودی- چندخروجی

ارائه دهنده:

اسماعیل طوسی

استاد راهنما:

دکتر علی سلیمانی

((تقدیم به پدر و مادر مهربانم

و

همسرعزیزم))

تقدیر و تشکر

گرچه دانسته هایم اندک است اما میخواهم آن را بشناسانم تا دیگری، بهتر از من، حقیقت را کشف کند و کاری را پی گیردکه به رفع اشتباه من بینجامد. و با این همه شادمان خواهم شدکه علت کشف آن حقیقت بوده ام.

((آلبرشت دورر))

انجام این پایان نامه میسرنبود مگر با لطف و کرم خداوند متعال، از این رو سربندگی و خضوع بر آستانش می سایم . حلاوت حکمت استادان بسیاری را درجان و از جان دوستتر دارم، هم آنان که فروغ دانششان فروزانی مسیرم بوده، با چشم ادب بر دستان موسایی آنان بوسۀ فروتنی می زنم. و از زحمات آنان در دورۀ تحصیل خود، بخصوص از زحمات بیدریغ استاد راهنمای پایان نامه خود جناب آقای دکترعلی سلیمانی کمال تشکر و قدردانی را دارم.

چکیده:

در این پایان نامه طراحی و ساخت یک نوع شبیه ساز سخت افزاری مورد بررسی قرار می گیرد. از آنجاییکه طراحی کنترل کننده برای سیستمهای SISO و MIMO در نرم افزارهایی مانند Matlab به سادگی میسر می باشد، می توان در محیط Simulink با بلوک دیاگرامها و تابعهایی که در نظرگرفته شده است سیستم مورد نظررا پیاده کرده و با اعمال ورودیهای واقعی، کنترل کننده مورد نیاز را بصورتPID ، فازی، عصبی یا الگوریتم ژنتیک طراحی کرد.

اما اگرقرار باشد سیستم بصورت واقعی پیاده شود، چون سیستم از یک محیط نرم افـزاری بـه یـک محیط سخت افزاری منتقل می شود، با توجه به تقریبهایی که وجود دارد جواب مطلوبی به دسـت نمی آید، لذا باید پارامترهای کنترل کننده را باز سازی و اصلاح کرد. ولـی درخیلـی از مـوارد روی سیستم واقعی این کار غیر ممکن به نظر می رسد، لذا بهتر است یک سیستمی شبیه سیستم واقعی ساخته شود و کنترل کننده قبل از اینکه به سیستم واقعی اعمال شود روی این سیستم تست شود و اصلاحات لازم صورت گیرد و بعد از حصول اطمینان از جواب صحیح سیستم، کنتـرل کننـده بـه سیستم واقعی اعمال گردد. بنابراین در این پایان نامه شبیه سازی بـصورت سـخت افـزاری سـاخته شده است که می توان سیستم مورد نظر را در یـک بـورد دیجیتـالی پـردازش کـرده و بـا اعمـال ورودیهای واقعی به شبیه ساز، خروجیهای واقعی نیز از آن دریافت کرد و کنترل کننده هایی را کـه در یک محیط نرم افزاری مانند Matlab طراحی شده اند، تـست و اصـلاح کـرد و بعـد از حـصول اطمینان از عملکرد آن روی سیستم تحت کنترل پیاده کرد.

فهرست

عنوان **صفحه**

فصل اول

مقدمه..... ۱

فصل دوم

بررسی شبیه سازهای زمان واقعی برای سیستمهای دو ورودی – دو خروجی..... ۳

۲-۱ شبیه ساز سخت افزاری..... ۵

۲-۲ استفاده ازشبیه سازی زمان واقعی در یک آزمایشگاه سرعت ۷

۲-۳ کاربردشبیه سازی..... ۸

۲-۴ شبیه سازی (AE ۴۵۲۴) ۹

۲-۴-۱ آزمایش (AE ۴۵۲۵)..... ۱۱

۲-۴-۲ تبدیل کد شبییه ساز ۱۴

۲-۴-۳ مدل خطی استخراج شده ۱۵

فصل سوم

سیستم های کنترل دیجیتال ۱۹

۳-۱ تعریف ۲۰

۳-۲ مزایای کنترل دیجیتال یک فرآیند در مقایسه با کنترل آنالوگ..... ۲۰

۳-۳ معایب کنترل دیجیتال ۲۰

۳-۴ روش های گسسته کردن سیستم زمان پیوسته..... ۲۲

۳-۴-۱ روش تفاضل معکوس ۲۳

۳-۴-۲ روش تبدیل دوخطی ۲۴

۳-۴-۳ روش تغیر نا پذیری پله ۲۴

۳-۵ پریود نمونه برداری ۲۷

۲ گم شدن اطلاعات ۲۸

۳ کم شدن دقت ۲۸

۳-۵-۱ انتخاب پریود نمونه برداری بصورت عملی ۲۸

۳-۶ تبدیل Z,وتحقیق آن..... ۲۹

۳-۷ فیلتر..... ۳۰

۳-۷-۲ انواع اساس فیلترها ۳۱

۳-۷-۳ فیلتر ایده آل..... ۳۲

۳-۸ معرفی انواع فیلتروپارامترهای آن..... ۳۳

۳-۹ آشنایی بافیلترهای پسیو واکتو ۳۴

۳-۹-۲ فیلترهای پایین گذراکتیو..... ۳۵

۳-۹-۳ فیلتر بالا گذر اکتیو..... ۳۵

۳-۹-۴ فیلتر های میان گذر اکتیو..... ۳۵
فصا چهارم
بررسی پردازنده (میکروکنترلر)سیستم شبیه ساز سخت افزاری مورد استفاده..... ۳۷
۴-۱ فهرستی از ویژه گی های(ATmega ۱۲۸)..... ۳۹
۴-۲ خصوصیات جانبی ۴۰
۴-۳ خصوصیات ویژه میکرو کنترلر ۴۰
۴-۴ ولتاژهای عملیاتی (کاری)..... ۴۰
۴-۵ خطوط I/0 وانواع بسته بندی..... ۴۱
۴-۶ ساختار پایه ها ۴۱
۴-۷ مدل آنالوگ به دیجیتال ADC..... ۴۳
۴-۷-۱ خصوصیات ویژه..... ۴۳
۴-۷-۲ عملکرد..... ۴۶
۴-۷-۳ شروع یک تبدیل ۴۷
۴-۷-۴ presalingوزمان بندی تبدیل ۴۷
۴-۷-۵ کانال های ورودی ADC..... ۵۰
۴-۷-۶ ولتاژ مرجع ADC..... ۵۰
۴-۷-۷ حذف نویز ADC..... ۵۲
۴-۷-۸ مداروردی آنالوگ ۵۲
۴-۷-۹ تکنیک های حذف نویز..... ۵۳
۴-۷-۱۰ تعریف دقت ADC..... ۵۴
۴-۷-۱۱ خروجی تبدیلADC..... ۵۶
۴-۸-۱ رجیسترها..... ۶۲
۴-۸-۲ تعریف ها ۶۳
۴-۸-۳ منابع کلاک..... ۶۳
۴-۸-۴ واحد شمارش ۶۴
۴-۸-۵ توضیح در مورد سیگنال ها ۶۴
۴-۸-۶ واحد مقایسه خروجی ۶۵
۴-۸-۷ شدت عمل مقایسه خروجی ۶۶
۴-۸-۸ بلوکه کردن تطابق match شده از طریق نوشتن در TCNT2..... ۶۶
۴-۸-۹ واحد Comparematchompu وتولیدشکل موج۱مدهای عملکرد..... ۶۸
۴-۸-۱۰ مدهای عملکرد..... ۶۲
۴-۸-۱۱ مد نرمال ۶۸
۴-۸-۱۲ مد پاک کردن Timerدر تطابق مقایسه گر (CTC).. .... ۶۹
۴-۸-۱۳ مد سریع PWM ۷۰

۴-۸-۱۴ مد Phase corret pwm ۷۰
۴-۸-۱۵ دیاگرام زمان بند ی ۷۴
۴-۸-۱۶ توصیف رجیستر timer/counter هشت بیتی ۷۵
فصل پنجم
سخت افزار پروژه ۸۱
۵-۱ ساختار شبیه ساز سخت افزاری ۸۴
۵-۱ منبع تغذیه ۸۴
۵-۳ مدار Reset میکرو ۸۴
۵-۴ عمل فیلترینگ ۸۵
۵-۵راه اندازی نرم افزار میکرو ۸۶
فصل ششم
پیاده سازی یک سیستم دو ورودی-دو خروجی روی شبیهساز سخت افزاری ۹۲
۶-۱ مدل هواپیما ۹۵
۶-۲ معادلات فضای حالت سیستم به شکل زیر میباشد ۹۵
۶-۳ برنامه شبیه ساز ۱۰۲
ضمیمه الف ۱۰۶
ضمیمه ب ۱۱۰

فهرست شکلها

شکل **صفحه**

شکل ۲-۱بلوک دیاگرام کلی محیط طراحی ۵
شکل ۲-۲ بلوک دیاگرام کلی شبیه ساز سخت افزار ۶
شکل ۲-۳ توابع تشکیل دهنده ی شبیه ساز زمان واقعی ۷
شکل ۲-۴ شبیه ساز با دو پنجره ی نمایشگر ۱۰
شکل ۲-۵ موقعیتهای تولید شده در شبیه ساز ۱۱
شکل۲-۶ بلوک دیاگرام پایدار ی نیروی جل برنده وابسته به طول جغرافیایی ۱۲
شکل ۲-۷بلوک دیاگرام افزایش پایدار ی هدایت افقی ۱۲
شکل ۲-۲ بلوک دیاگرام وابستگی طول جغرافیایی ۱۳
شکل ۲-۹ بلوک دیاگرام وابستگی هدایت افقی..... ۱۳
شکل ۳-۲ اثرات افزایشوکاهش پریود نمونه برداری ۲۸
شکل ۳-۳ پیاده سازی تابع تبدیل Z به روش مستقیم نوع I I..... ۳۰
شکل ۳-۴ باند عبور یک فیلتر ایده آل..... ۳۲
شکل ۳-۵ نمایش عمودی یک فیلتر ۳۲
شکل ۳-۶ یک شبکه ساده RC بانقطه قطع ۳- دسی بل دراهرتر ۳۴
شکل ۳-۷ فیلترهای پایین گذر (a)نوع π (b)نوع T..... ۳۵
شکل۳-۸ فیلترپیو متصل شده به منبع سیکنال با مقاومت داخلی خیلی کم که به امپدانس منتهی میگردد..... ۳۵
شکل ۳-۹ یک نوع فیلتر بالا گذر اکتیو –تقویت کنندهی عملیاتی ۳۵
شکل ۴-۱ نمودار مقایسه ی افزایش ۳۹
شکل ۴-۲ پایه های خروجی ۴۱
شکل ۴-۳ بلوک دیاگرام تبدیلی آنالوگ به دیجیتال ۴۵
شکل ۴-۴ ADCPrescaler..... ۴۷
شکل۴-۵ دیاگرام زمان بند ADC اولین تبدیل(مد تبدیل single)..... ۴۸
شکل ۴-۶ دیاگرام زمان بندی ADC تبدیل single..... ۴۹
شکل ۴-۷ دیاگرام زمان بندی ADCتبدیلی در مد..... ۴۹
شکل ۴-۸ مدار ورودی آنالوگ..... ۵۳
شکل ۴-۹ اتصالات توان ADC..... ۵۴
شکل ۴-۱۰ خطای آفت ۵۵
شکل ۴-۱۱ خطای بهره ۵۵
شکل۴-۱۲ رنج تغییرات اندازه گیری شده ۵۷
شکل ۴-۱۳ بلوک دیاگرام Timer/conter هشت بیتی ۶۲
شکل۴-۱۴ بلوک دیاگرام کانتر..... ۶۴

شکل ۴-۱۵ بلوک دیاگرام قسمت مقایسه گر خروجی..... ۶۵
شکل۴-۱۶ شماتیک واحد..... ۶۷
شکل۴-۱۷ مدCTC دیاگرام زمان بندی..... ۷۰
شکل۴-۱۸ مد سریع PWM دیاگرام زمان بندی..... ۷۱
شکل۴-۱۹ مد Phase correct PWM دیاگرام زمان بندی..... ۷۳
شکل۴-۲۰ دیاگرام زمان بندی Timer/conter بدون Prescaler ۷۴
شکل۴-۲۱ دیاگرام زمان بندی با Prescaler ۷۴
شکل۴-۲۲ نشانگر یک شدن Ocf2 در همه مدها به جزء مد CTC ۷۵
شکل۴-۲۳ دیاگرام زمان بندی تایمر – کانتر در مد تطابق..... ۷۵
شکل ۵-۱ بلوک دیاگرام کلی سیستم پیشنهادی..... ۸۲
شکل۵-۲ بلوک دیاگرام سخت افزار پروژه ۸۳
شکل۵-۳ منبع تغذیه سیستم ۸۴
شکل ۵-۴ مدار Reset ۸۵
شکل ۵-۵ فیلتر اکتیو پایین گذر..... ۸۶
شکل ۵-۶ پردازنده Atmega128 ..... ۹۱
شکل ۵-۷ برد آماده شده شبیه ساز..... ۹۲
شکل ۵-۸ سیستم شبیه ساز همراه با پروگرامر و سیستم ناظر..... ۹۳
شکل ۶-۱ نمایش پارامترها روی مدل..... ۹۵
شکل ۶-۲ اثر ورودی یک روی خروجی یک درحالت زمان- پیوسته..... ۱۰۰
شکل۶-۳ اثر ورودی یک روی خروجی یک درحالت زمان- گسسته..... ۱۰۰
شکل ۶-۴ اثر ورودی دو روی خروجی دو درحالت زمان- پیوسته..... ۱۰۱
شکل۶-۵ اثر ورودی دو روی خروجی دو درحالت زمان – گسسته..... ۱۰۱
شکل۶-۶ اثر ورودی دو روی خروجی یک درحالت زمان – پیوسته..... ۱۰۲
شکل ۶-۷ اثر ورودی دو روی خروجی یک درحالت زمان –گسسته..... ۱۰۲
شکل ۶-۸ اثر ورودی یک روی خروجی دو درحالت زمان- پیوسته..... ۱۰۳
شکل ۶-۹ اثر ورودی یک روی خروجی دو درحالت زمان – گسسته..... ۱۰۳
شکل ۶-۱۰ فلوچارت برنامه..... ۱۰۴

فصل اول

مقدمه

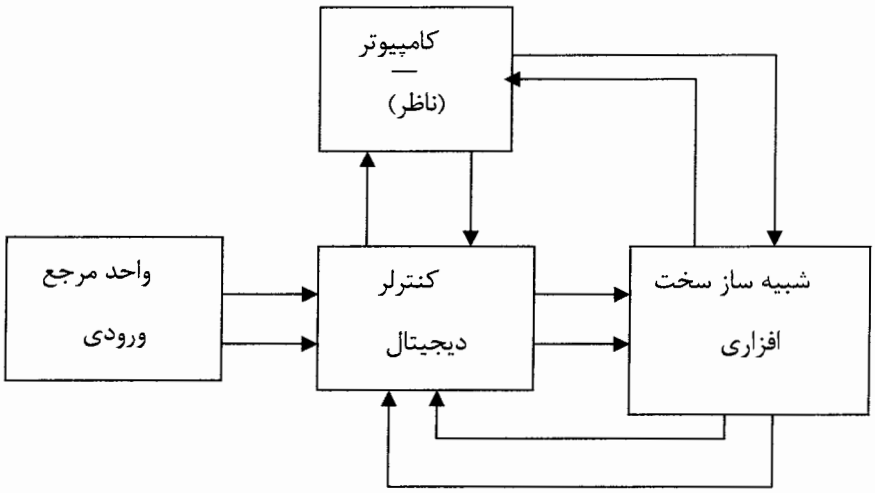
ششم یک نمونه به عنوان مثال مورد بررسی قرار گرفته و خروجیها درحالت شبیه سازی نرم افزاری با شبیه سازی سخت افزاری بررسی شده است.

فصل دوم

بررسی شبیه سازهای زمان واقعی برای سیستم های دو ورودی-دوخروجی

در این فصل محیط شبیه سازهای سخت افزاری مورد بررسی قرار می گیرد در این محیط شبیه ساز به صورت زمان واقعی شبیه سازی شده و برنامه های مورد نظر به برد شبیه ساز ارسال و اطلاعات لازم اعمال می گردد طراحی و ساخت کنترلر برای سیستمهایsiso وmimo در نرم افزارهایی چون matlab به خوبی عملی می باشد و می توان در محیط simulink مطلب با استفاده ازبلوک دیاگرامها و تابعهایی که پیش بینی شده سیستم مورد نظر را پیاده کرده و ارتباط ها را به سادگی برقرار کرد و ضمن پردازش سیستم و اعمال ورودیهای مورد نظر خروجی را دریافت کرد و با استفاده از آن کنترل کننده مورد نیاز را به صورت PID ، فازی ، عصبی یا الگوریتم ژنتیک طراحی کرد و در این محیط به جواب مورد نظررسید .

اما اگر قرار باشد این سیستم و کنترل کننده به صورت واقعی پیاده شوند، اگر سیستم به صورت دیجیتالی مد نظر باشد باید در یک برد پروسسوری محقق گردد با اجرای این عمل چون سیستم از یک محیط نرم افزاری به یک محیط سخت افزاری منتقل می شود ، خواه ناخواه تقریب یا تقریبهایی در ورودیها ، خروجیها ممکن است لازم باشد. باوجود این تقریبها لازم است کنترلری که در محیط MATLAB طراحی شده اصلاح و بازسازی شود .اگر چنانچه لازم باشد این کنترل کننده باز نگری واصلاح شود در خیلی از موارد نمی توان این اصلاحات را روی سیستم واقعی انجام داد . لذا لازم است قبل از اینکه کنترلر به سیستم واقعی اعمال گردد آنرا به سیستمی مشابه سیستم واقعی اعمال کرد و در صورت موفقیت و انجام تست های لازم و بررسی احتمال ها و تلرانس ها و نویزهای اعمالی، به سیستم واقعی اعمال گردد. برای این منظور لازم است، سیستم مورد نظر شبیه سازی و جانشین سیستم واقعی گردد و کنترلر بصورت زمان واقعی تست و اصلاح شود . از آنجایی که ممکن است لازم باشد روند اجرا و عملکرد توسط سیستمی تحت نظارت و بررسی قرار گیرد . سیستم ناظری برای مجموعه منظور می گردد . با این توضیح بلوک دیاگرام کلی به صورت شکل ۲-۱ خواهد بود .



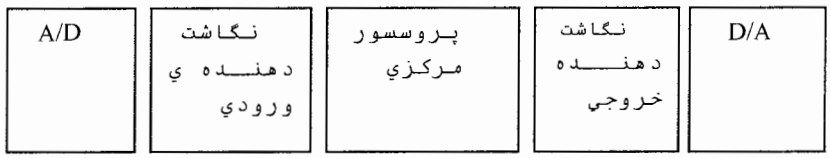
شکل ۲-۱- :بلوک دیاگرام کلی محیط طراحی کنترلر

علاوه بر موارد مطرح شده با توجه به ساختار محیط مورد نظر می توان از آن به عنوان یک وسیله آزمایشگاهی جهت آشنایی با مفاهیم کنترل دیجیتال و سیستمهای چند ورودی- چند خروجی استفاده کرد . و همچنین به عنوان محیطی جهت تست کنترل کننده های معمولی ، تطبیقی ، مقاوم ، فازی و غیره و همچنین بکارگیری الگوریتمهای شناسایی استفاده کرد.

۲-۱- شبیه ساز سخت افزاری :

این شبیه سازها ممکن است با توجه به ساختار سیستم واقعی با سیستم مورد نظر ، با آمپلی فایر قدرت ، مدارات الکترونیکی ،کامپیوتر آنالوگ و یا کامپیوتر دیجیتال ساخته شوند .

هر کدام از این روش ها مزایا ومعایب خاص خود را دارد در این بحث شبیه ساز انتخاب شده از نوع کامپیوتر دیجیتال می باشد مزیت عمده ای که این نوع شبیه ساز دارد قابلیت انعطاف آن و عیب عمده سرعت کم و خطای موجود در شبیه سازی می باشد ، و با طراحی مدارات جانبی لازم می‌توان این عیبها و نقصها را کم نمود . بلوک دیاگرام کلی این شبیه ساز بصورت شکل ۲-۲ می‌باشد.



شکل ۲-۲- بلوک دیاگرام کلی شبیه ساز سخت افزاری

در این بلوک دیاگرام وظیفه بلوک A/D تبدیل سیگنال آنالوگ به دیجیتال و وظیفه بلوک نگاشت دهنده ورودی، تبدیل عدد دیجیتال خوانده شده به یک عدد بین صفر تا پنج و همچنین حذف آفست مربوط به سطح سیگنال می‌باشد و پروسسور مرکزی محاسبات لازم را بر عهده دارد . و جهت اینکه نتایج به D/A اعمال گردد، بایستی اعداد صفر تا ۵ به اعداد صفر تا ۲۵۵ نگاشت گردد و همچنین بایستی در بعضی از موارد به سیگنال خروجی مقداری آفست داد و ارزش آن را کم یا زیاد نمود. بلوک نگاشت دهنده خروجی این وظایف را انجام می‌دهد و مقدار صفر تا ۲۵۵ حاصله به D/A اعمال شده و یک سیگنال آنالوگ بین صفرتا پنج ولت درخروجی ظاهرخواهد شد.

این شبیه ساز با استفاده از میکرو کنترلر Atmega128 و با قطعات و عناصر جانبی ساخته شده و قادر است یک تابع دو ورودی- دو خروجی را شبیه سازی نماید . از آنجاییکه که تمامی محاسبات و اعمال در زمان واقعی انجام می‌گیرد سیستم دارای خطاهای زیر می‌باشد که بایستی به آنها توجه نمود .

الف : تاخیر های ناشی از تبدیل آنالوگ به دیجیتال ، تبدیل دیجیتال به آنالوگ و محاسبات ریاضی لازم.

ب : عدم دقت A/D و D/A

ج : نویز و اثرات آن

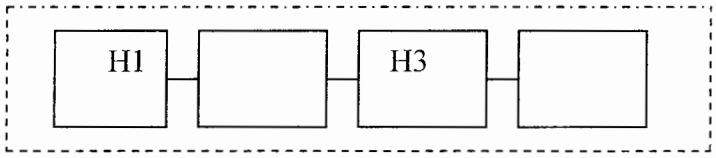
همانطورکه مشاهده می شود شبیه ساز فوق از بلوکهای مجزایی تشکیل شده است،اگر چنا نچه برای بلوکهایA/D و D/Aمحاسبات ریاضی که بصورت تأخیرظاهرمی شوند ، توابعی استخراج نماییم تحت شرایطی می توان سیستم را بصورت واقعی تری شبیه سازی کرد. برای این منظورتوابع H1 ، H2 ، H3و H4 که بصورت شکل ۲-۳ قرارگرفته اند در نظر گرفته می شوند.

تابع تبدیلA/D=H1

تابع تبدیل محاسبات ریاضی=H2

تابع تبدیل مجهول=H3

تابع تبدیل D/A=H4



شکل۲-۳- توابع تشکیل دهندهٔ شبیه سازمان واقعی

بایستی تابع H3 را به نحوی تعیین نمود که حاصل سری شدن توابع H1 , H2 , H3 وH4 برابر تابع سیستم مورد نظر گردد . که البته تعیین در حالت کلی ساده نمی باشد و با تقریب می توان آن را بدست آورد . چون محاسبه تابع H3 که برای جبران تأخیر ناشی از مبدلهای A/D , D/A و محاسبات ریاضی سیستم درنظر گرفتته شده است موضوع مورد بحث این پایان نامه نمی باشد ، لذا برای جبران زمان تأخیر مورد نظر از یک پردازنده با سرعت پردازش بالا استفاده میشود. حال به بررسی یک نمونه شبیه ساز سخت افزاری که در حال حاضر در مراکز آزمایشگاهی مورد استفاده قرار می‌گیرد می پردازیم.

۲-۲- استفاده از شبیه سازی زمان واقعی در یک آزمایشگاه سرعت

در دهه های گذ شته یک ابزار مهم برای طراحی سیستم کنترل پرواز ، شبیه سازی کامپیوتری بود ، استفاده از یک شبیه ساز، یک مهندسی سالم ، سریع و راحت در اختیار قرار می‌دهد . و روشی برای تست و طراحی کنترلر جدید می‌باشد.

چند سال قبل تعدادی از دانشگاه‌ها استفاده از طراحی کامپیوتری را در ارتباط با طراحی کنترل کننده ، شروع کردند . انجمن تکنولوژی مدرسه مهندسین فضانوردی در گرجستان یکی از بزرگترین سطوح دوره‌های آزمایشگاهی مهندسین فضانوردی می باشد این انجمن با سه آزمایشگاه جدید برای بکار بردن کامپیوتر برای تکنولوژی شبیه سازی پرواز توسعه داده شده است . این روش به دانشجویان اجازه می‌دهد کنترل کننده های گوناگونی را با یک شبیه ساز پرواز با صحت بالا به جای تنها مطالعه روی مدل تئوری ، طراحی و تست کنند.

شبیه ساز کامپیوتری یک روش با ارزش و قابل دسترسی برای مهندسی کنترل می‌باشد. شبیه سازی اغلب در مقایسه با تست روی یک سخت افزار واقعی ارزانتر و ساده تر می‌باشد .

هنگام طراحی کنترل کننده جدید ، تست کنترل کننده ، روی یک شبیه ساز مقدماتی آسانتر می‌باشد . در این روش اگر مشکلی وجود داشته باشد طراحان می‌توانند سریعتر ، با هزینه کمتر بدون خسارت و تجهیزات کم، آن مشکل را برطرف سازند . بعلاوه شبیه ساز سخت افزاری به مهندسین کنترل این شانس را می‌دهد که روشهای جدیدی را برای کنترل کننده مطمئن ایجاد کنند .

۲-۳-کاربرد شبیه سازی :

درچند سال گذشته دانشگاه های زیادی استفاده از شبیه سازی را در کنترل کلا سیک به عنوان ابزار تدریس شروع کردند . این روش به دانشجویان اجازه می دهد ، برای سیستم ها و تجهیزات خود و تست آنها کنترل کننده ای طراحی کنند . همچنین آنها توانایی تنظیم پارامترها را در طراحی شان دارند و تأ ثیر آن را روی سیستم شبیه سازی شده می بینند. بعلاوه دانشجویان همه متغیرهای حقیقی را که می تواند روی سیستم تاثیر بگذارد ، می توانند آزمایش کنند . شبیه سازی کامپیوتری در آزمایشگاههای کنترل در راههای متنوعی استفاده می شود . دانشگاه جان هاپکینز پیشنهاد یک دوره طراحی سیستم های کنترلی را داد که بوسیله پابلوایگلسیاس تدریس می شد .

پروژه نهایی در این کلاس برای پرواز، برای هر گروه دانشجویی یک مدل هلیکوپتر بود که برای طراحی کنترل کننده توسط آنها استفاده می شد . با توجه به مطالب، برای مدل واقعی پرواز، دانشجویان می توانند کنترل کننده واقعی را قبل از مبادرت به پرواز مدل واقعی هلیکوپتر ، روی یک شبیه ساز هلیکوپتر تست کنند . گروه مهندسی برق در دانشگاه سنگاپور یک آزمایشگاه شبیه سازی را با به هم پیوستن دستگاهها ایجاد کرد . این شبیه ساز قابلیت انجام کنترل دستی ، تناسبی - انتگرالی- مشتقی ، فضای حالت و فازی را دارد .

زیرا قابلیت های این شبیه ساز یک روش عمومی برای تست کنترل کننده جدید را مهیا می کند .دانشگاه ایالت پنسیلوا نیا از یک مرحله شبیه ساز موقعیت برای یک کنترل فید بک سطح بالا در تکنولوژی مهندسی الکترومکانیک استفاده می کند. این وسیله تابع انتقال مراحل مختلف را شبیه سازی می کند که ممکن است با آن مواجه شود . پارامترهای تابع انتقال را می توان به آسانی تغییر داد و به شبیه ساز اجازه داد تا مراحل مختلفی را جایگزین کند، این شبیه ساز می تواند برای

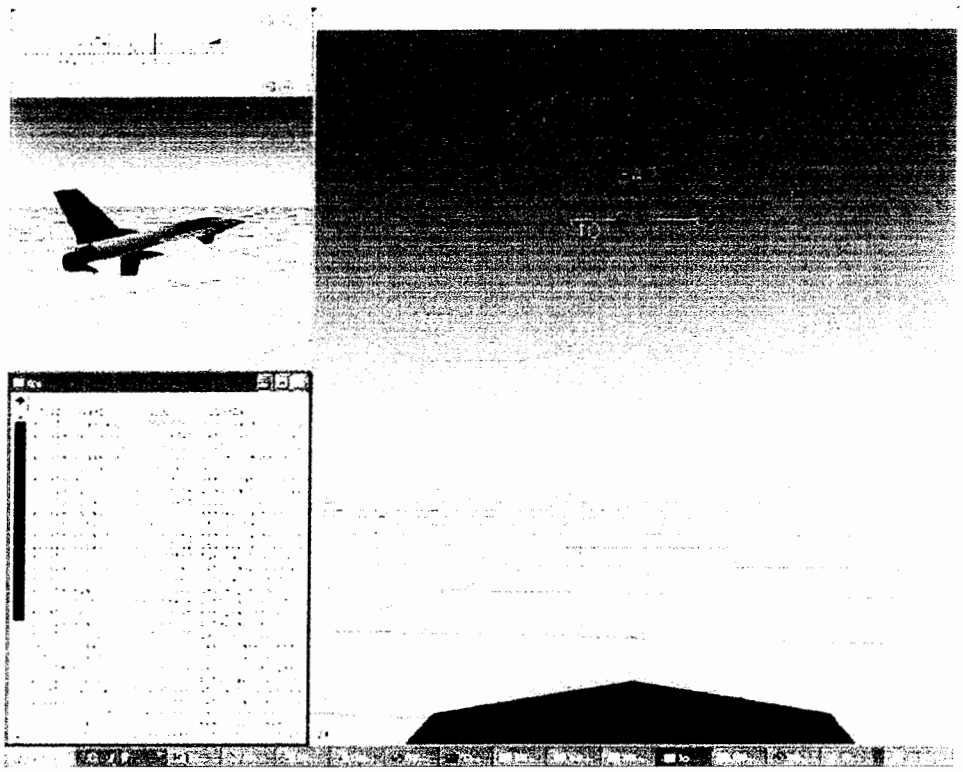
تست نتیجه کنترل کننده PID استفاده شود . بعلاوه خطای حالت ماندگار و مرحله پایدار می تواند ارزیابی شود.

۲-۴-شبیه ساز (AE4524)

دربهار ۲۰۰۲ سه تجربه جدید به تجربه مدرسه مهندسی هوا فضا در دوره لیسانس آزمایشگاه کنترل اضافه شد . شبیه ساز استفاده شده روی یک هواپیما ی F-۱۶ نمایش داده می شود. زیرا پرواز بدون افزایش پایداری خیلی مشکل است . دانشجویان امکان پرواز، با شبیه ساز با چندین سیستم کنترل متفاوت را که خودشان طراحی می کنند را دارند . شبیه ساز قادر است سه پنجره متفاوت با پنج انتخاب متفاوت قابل دسترس را در یک زمان نمایش دهد . عامل نمایش اتاق خلبان نامیده می شود ، زمین و جسم شناور را حول محورها دنبال می کند . اتاق خلبان نمایش دهنده مرکزی می باشد . مثل این که گرداننده خلبانی است که جهت جلو را در طول محور بدنه هواپیما نشانه گرفته است و مبدا ء زمین ثابت است و بطور متناوب برای ماندن هواپیما در یک فاصله معین با دنبال کردن زمین ، نشانگر شناور عوض می شود . شبیه ساز محورهای مختلف سیستم و پارامترهایی از یک انتخاب شش تایی را که تعدادی از اینها ممکن است در یک زمان نمایش داده شوند قادر است نمایش دهد . بردارهایی نیز برای سرعت ، سرعت زاویه ای ، زاویه حرکت و مسیر می تواند روی صفحه نمایش نشان داده شود . برای هر یک از این بردارها یک رنگ متفاوت در نظر گرفته می شود . بعلاوه شبیه ساز پرواز شامل یک فهرست گروهی می باشد که به اپراتور اجازه مشاهده اندازه پرواز و کنترل پارامترها در زمان واقعی را می دهد همچنین یک سیستم خلبان اتوماتیک وجود دارد که می تواند هواپیما را روی خط نگه دارد اساس کار خلبان اتوماتیک این گونه است که اجازه می دهد اپراتور بر اساس سرعت و ارتفاع ، مسیر را مشخص کند ارزش زیاد آن برای این است که می توان زمان پرواز را تغییر داد این شبیه ساز به اپراتور اجازه می دهد که کنترل کننده طراحی شده را روی یک مسیر پرواز اصلی تست کند و همچنین اجازه می دهد هواپیما به هر طریقی که میل دارد با توجه به مجموعه اطلاعات با اطمینان پرواز کند .

علاوه بر این پارامترهای کنترلی می توانند در طول پرواز تغییر کنند ، که اجازه دهند کنترل کننده طراحی شده سریع و آسان تست شود . شکل ۲-۴ نمایی از شبیه ساز با دو پنجره نمایشگر را

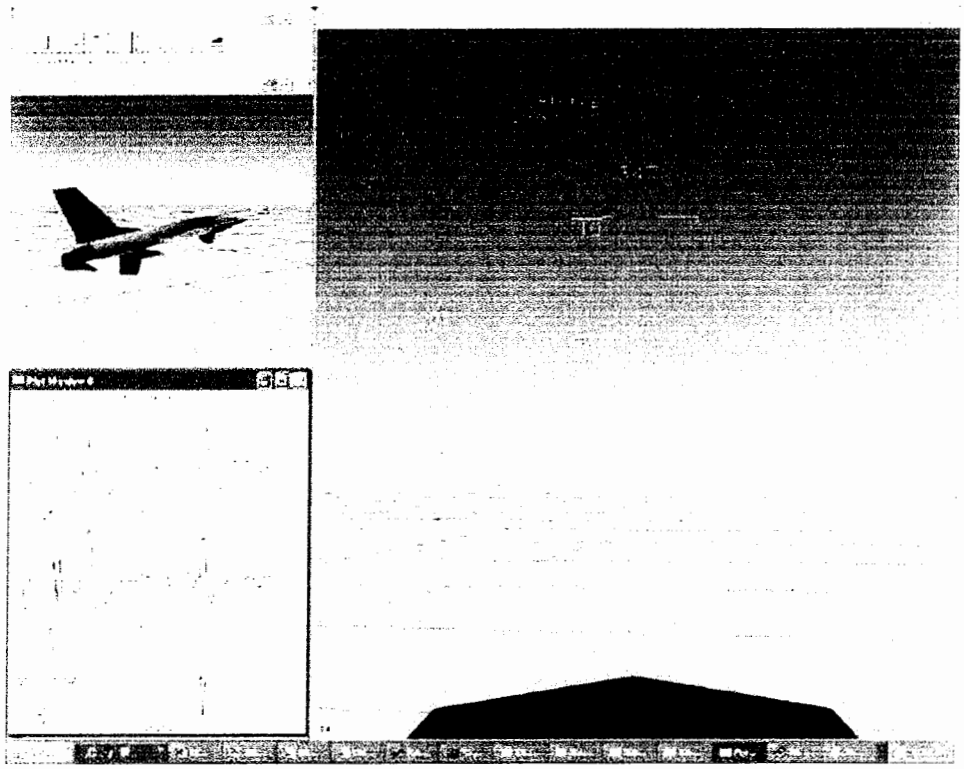
نشان می دهد . یکی در اتاقک خلبان و دیگری در نمایشگر دنبال کننده وجود دارد. نمایشگر دنبال کننده ، محورها را نیز روی آن نشان می دهد .



شکل۲-۴- شبیه ساز با دو پنجره نمایشگر

علاوه بر این یک پنجره پارامتری در سمت چپ ، پایین شکل می توان دید شبیه ساز می تواند اطلاعات را جمع کند و موقعیت را مشخص کند تا حدود سه پنجره در یک زمان می تواند بازشود موقعیت های تولید شده زمان واقعی ، اجازه می دهند اپراتور گرافیک اتفاق های رخ داده را مشاهده کند بعد از مورد قبول واقع شدن داده های جمع شده می تواند آنها را در فایل متنی ذخیره کند.

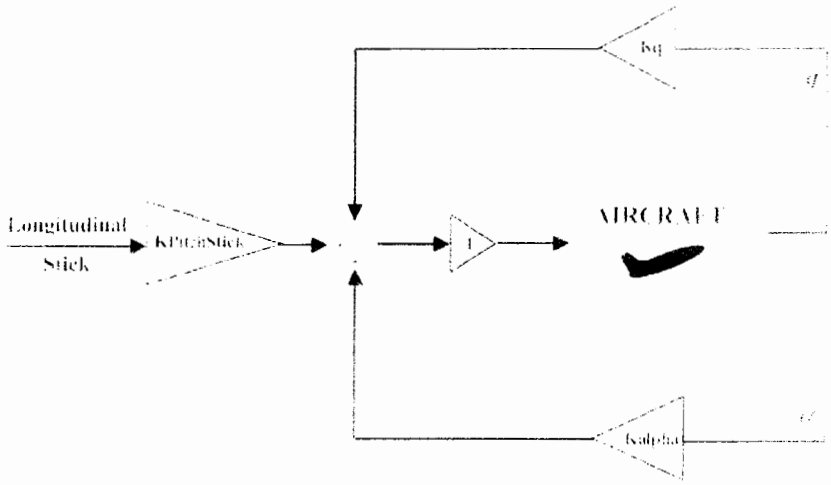
مثال این موقعیت در شکل ۲-۵ نشان داده شده است .



شکل۲-۵- موقعیتهای تولید شده درشبیه ساز

۲-۴-۱- آزمایش AE4525

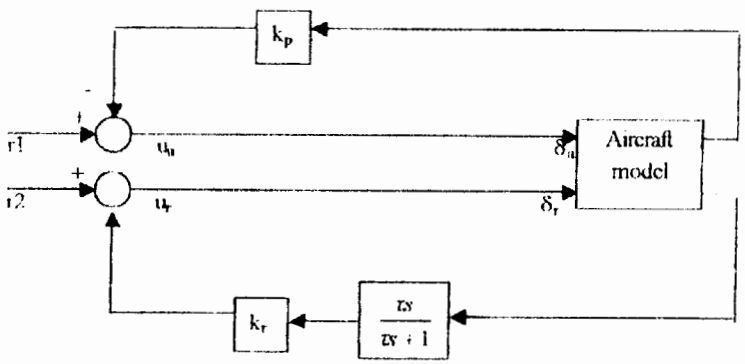
موقعیکه آزمایش انجام شد، هواپیما ابتدا در یک حلقه باز (ناپایدار) دیده می شود که برای پرواز مناسب ، به سیستم کنترل پرواز نیاز دارد . در ابتدای این آزمایش دانشجویان سیستم کنترل وابسته به طول جغرافیایی با زاویه حمله و زاویه اوج و فیدبک سرعت را طراحی و اجرا می کنند. بخش دوم آزمایش ، اندازه گیری بخش هدایت افقی سیستم کنترل فیدیک ، زاویه چرخش و انحراف از مسیر سرعت می باشد . آزمایش نهایی ، طراحی و اجرای یک سیستم کنترلی و افزایش پایداری حقیقی می باشد، با شبیه سازی دانشجویان قادرهستند پاسخ شبیه سازی هواپیما را تنظیم کنند . بعد از طراحی جدید ، سیستم کنترل اجرا شده و دانشجویان یک پرواز مخصوص را تست ، و اطلاعات را جمع آوری می کنند. آزمایش اول وابستگی ساختمان سیستم کنترل، به طول جغرافیایی بود . شکل ۲-۶ بلوک دیاگرام این سیستم کنترل را نشان می دهد.



شکل۲-۶- بلوک دیاگرام پایداری نیروی جلو برنده وابسته به طول جغرافیایی

در این شکل زاویه حمله و سرعت چرخش فیدبک متغیر هستند . قبل از رسیدن به آزمایشگاه دانشجویان انتظار داشتند ، برای محاسبه گین ضروری از تکنیک روت لوکاس استفاده کنند . سپس در کلاس دانشجویان بایستی پرواز حلقه باز ، با انجام سیستم کنترل را بفهمند .در طول پرواز آنها بایستی اطلاعات مربوط به زاویه حمله ، سرعت اوج و سمت بالا را برای آنالیز بیشتر ثبت کنند . در پایان آزمایش انتظار می رود که دانشجویان بتوانند نقشه روت لوکاس را برای هر دو گین رسم کنند و موقعیت نهایی ریشه ها ، گین و فاز مارجین سیستم حلقه بسته را محاسبه کنند و اهمیت گین و فاز مارجین را بفهمند . بعلاوه دانشجویان کار هواپیما را بر اساس داده های جمع شده تحلیل می کنند، تا کیفیت سرعت، برای استفاده جنگ افزاری را پیش بینی کنند .

آزمایش دوم طراحی ، ساخت و پایداری یک سیستم افزایشی می باشد . بلوک دیاگرام شکل ۲-۷ نشان می دهد که سرعت چرخش و انحراف در حلقه فیدبک قرار دارند.

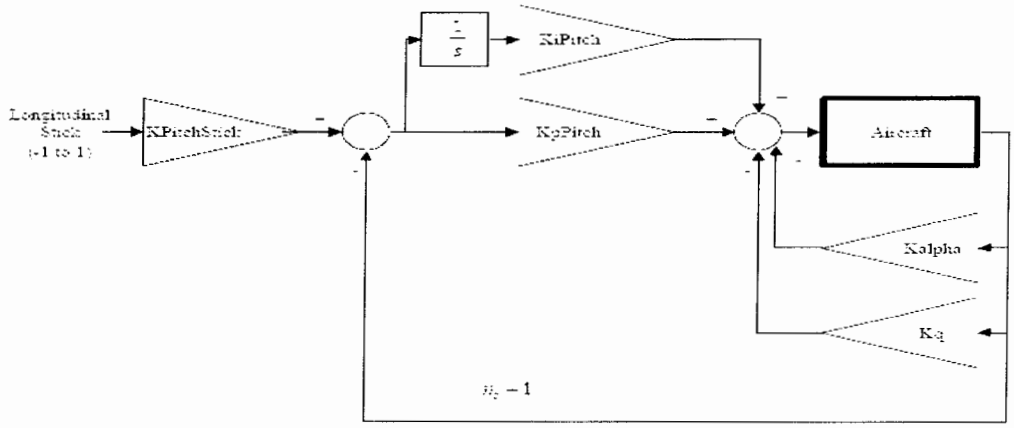


شکل۲-۷-بلوک دیاگرام افزایش پایداری هدایت افقی

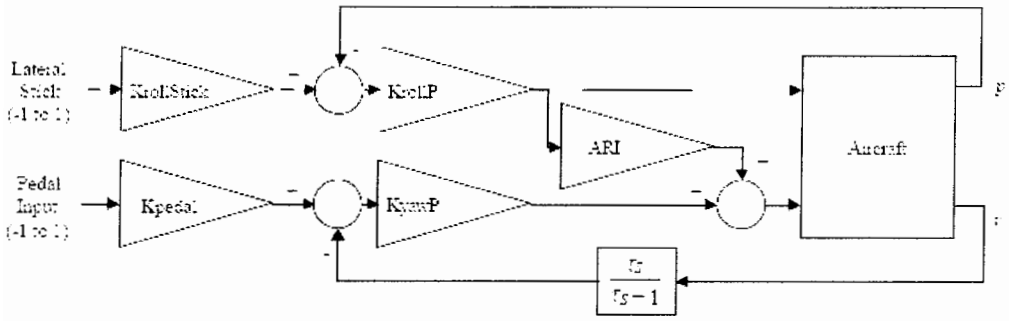
قبل از اینکه گین را محاسبه کنند و سپس پرواز را شبیه سازی کنند هر دو را با هم بدون سیستم کنترل اجرا می کنند . در این حالت داده ها برای سرعت انحراف و سرعت چرخش ، به خوبی سکان هواپیما و قسمت متحرک هواپیمارا حمایت می کنند .از این آزمایش نسبت به قبلی انتظار بیشتری می باشد . به هر حال اندازه داده ها وابسته به سمت جانبی می باشد .

آزمایش سوم مربوط به طراحی یک پایدار ساز و کنترل سیستم افزایشی (SCAS[1]) که شامل طول جغرافیایی و هدایت جانبی اجزاء یک سیستم کنترل پرواز کامل می باشد .

شکل ۲-۸و۲-۹ بلوک دیاگرام وابستگی طول جغرافیایی و هدایت جانبی را نشان می دهند .



شکل۲-۸- بلوک دیاگرام وابستگی طول جغرافیایی



شکل۲-۹- بلوک دیاگرام وابستگی هدایت افقی

همانطور که در شکل می توان دید پارامترهای عمده ای وجود دارد که لازم است تعیین شوند .

[1] Stability and control augmentation system

به هرحال چیزهای محدودی با استفاده از شبیه ساز کشف می شوند و اغلب صحیح می باشند. در این آزمایش دانشجویان برای رسیدن به بهترین گین تعدادی از این گینها را می توانند اصلاح کنند و تاثیر این تغییرات را روی سیستم ببینند .

برای تجزیه و تحلیل این آزمایش دانشجویان انتظار دارند، مقادیر بدست آمده را بفهمند. و نتایج تجزیه و تحلیل خطی دینامیکی وسیله را چگونه انتخاب کنند . آنها همچنین می خواهند شبیه سازی این مدل خطی را با نتایج داده های بدست آمده مقایسه کنند . بعلاوه دانشجویان سرعت هواپیما را برای طراحی بدست می آورند و پردازش می کنند .

۲-۴-۲ تبدیل کد شبیه ساز:

برای بهبود آزمایش تصمیم گرفته شد که مدل خطی سیستم به آزمایش اضافه شود . پیش از این مدل خطی ، برای یافتن کنترل کننده هواپیما و شبیه سازی بوسیله برایان استیون و فرانکی استفاده می‌شد . این مدل برای یک ارتفاع و سرعت پرواز خطی مناسب است ،۵۰۰ پا از سطح زمین. مدل خطی در دسترس باید به دانشجویان اجازه دهد تا یک مدل خطی واقعی برای پرواز طراحی کنند. به هر حال برای آماده کردن مدل خطی معادله حرکت ، برای شبیه سازی احتیاج به دو باره نویسی و خلاصه کردن دارد . برای شروع ، یک مدل کروی زمین استفاده می‌شود که شامل بدنه ، اتمسفر ، اینرسی و ساختمان ثابت شده زمین است . این به جای یک زمین معلق که فقط شامل سه مشخصه : محل ، بدنه و آتمسفر می‌باشد استفاده می‌شود، نتیجه تاثیر زمین ، شکل محل و اینرسی بطور همزمان است . آنها برای جهت یابی شمال ، شرق و پایین سیستم استفاده می‌شوند. ساختمان بدنه با قسمتهای عمده اصلی هواپیما جایگزین می‌شود دماغه با محور X ها نشان داده می شود، محور Yها سمت راست را نشان می‌دهد و محور Z رو به پایین می‌باشد .

محور اتمسفر، سیستم جهت‌یابی است که وابسته به اتمسفر در جهت محور X ها در امتداد سرعت باد می‌باشد بعلاوه ، بسیاری از متغیرهای نوشته شده به شکلهای مختلف رابطه دارند . بدست آوردن مدل خطی خلاصه شده احتیاج به متغیر وابسته گذا شته شده در ساختمان بدنه دارد.

۲-۴-۳-مدل خطی استخراج شده :

مدل خطی استخراج شده و سپس به شبیه ساز اضافه می‌شود . و در یک خط فرمان مـی‌تـوان آن را تایپ کرد ، این مدل خطی سپس در یک فایل ذخیره می شود .

مدل خطی آماده شـده بـه فـرم $\dot{X} = AX + BU$ مـی باشـد کـه در آن X حالـت ، U ورودی سیستم ، A ماتریس حالت و B ماتریس ورودی می باشد .

مدل خطی شامل هشت حالت می‌باشد

سرعت زاویه ای: (p,q,r)

سرعت: (u,v,w)

زاویه چرخش و زاویه اوج p و θ

معادله برای شش حالت بوسیله اولین تحریک هر حالت در طول آن محور تعیین می شود. مقدار هر حالت مشتق گرفته می شود و سپس با نسبت حالت تحریک شده جمع می شود وسپس در ماتریس A ثبت می شود.

با استفاده از دنبال کردن معادله حرکت مقادیر به صورت زیر تعیین می شوند .

$$P = \phi^{'} - \psi^{'} COS\phi \qquad (۱-۲)$$

$$q = \theta^{'} * \cos(\phi) + \psi^{'} \sin(\phi)\cos(\theta) \qquad (۲-۲)$$

$$r = -\theta^{'} \sin(\phi) + \psi^{'} \cos(\phi)\cos\theta \qquad (۳-۲)$$

درشبیه سازی برای محاسبه ماتریس B ازروش تحریک هواپیما استفاده می شود در این شبیه ساز چهار ورودی وجود دارد :قسمت متحرک بالهای هواپیما ، سکان هواپیما (وسیله هدایت)، دریچه کنترل بنزین و بالابر . تابع مطلب همچنین اصلاح ماتریسهای A وB را فراهم می کند تا اینکه آنها بتوانند به آسانی طول جغرافیایی را از هم جدا کنند و مسیر ترکیب شده را اصلاح کننـد . بعـلاوه یک جمع کننده برای عوض کردن ترکیبات سرعت w,v,u برای سرعت نهـایی اضـافه شـده اسـت بخصوص A , B .

مدل خطی نهایی شامل حالتهای مقابل می باشد . $v, \alpha, \theta, q, \beta, \phi, p, r$

دراین شکل ابتدا چهار ردیف وستون شامل طول جغرافیایی و مسیر هدایت در ماتریسAیافت می شود. B و A ماتریسهای وابسته به طول جغرافیایی که در کتاب شبیه سازی کنترل و شبیه سازی هواپیما به وسیله استیون و لویس آمده است.

A و B ماتریسهای طول جغرافیایی:

$$A=\begin{bmatrix} -0.019311 & 8.8157 & -32.17 & -0.57499 \\ -2.5398*10^{-4} & -1.0189 & 0 & 0.90506 \\ 0 & 0 & 0 & 1 \\ 2.5398*10^{-12} & 0.8225 & 0 & -1.0774 \end{bmatrix}$$

و

$$B=\begin{bmatrix} 9.95391 \\ -0.12319 \\ 0 \\ -10.059 \end{bmatrix}$$

B,A ماتریسهای هدایت افقی:

$$A=\begin{bmatrix} -0.13150 & 0.14858 & 0.32434 & -0.93964 \\ 0 & 0 & 1 & 0.33976 \\ -10.614 & 0 & -1.1793 & 1.0023 \\ 0.99655 & 0 & -0.0018174 & -0.25855 \end{bmatrix}$$

و

$$B=\begin{bmatrix} 0.006903 & 0.018849 \\ 0 & 0 \\ -5.910507 & 1.202467 \\ -0.122219 & -0.61392 \end{bmatrix}$$

ماتریسهای طول جغرافیایی با یک سرعت سیر هوایی ۵۰۲ فیت بر ثانیه (feet/second) (هر فیت ۱۲ اینچ) از سطح دریا حساب می شوند . ماتریسهای افقی شبیه ارتفاع با سرعت ۲۰۵فیت بر ثانیه (feet/second) تهیه می شوند ماتریسهای بدست آمده با مدل خطی جدید مرتبط می باشند.

A و B ماتریسهای وابسته به طول جغرافیایی :

$$A=\begin{bmatrix} -0.0371 & 9.0006 & -32.17 & -0.5676 \\ -2.56*10^{-4} & -1.0651 & 0 & 0.9008 \\ 0 & 0 & 0 & 1 \\ 0 & 0.8186 & 0 & -1.0738 \end{bmatrix}$$

و

$$B=\begin{bmatrix} 9.9667 \\ -0.1287 \\ 0 \\ -9.6176 \end{bmatrix}$$

A , B ماتریسهای جهت افقی:

$$A=\begin{bmatrix} -0.1370 & 0.1495 & 0.3175 & -0.9415 \\ 0 & 0 & 1 & 0.3189 \\ -11.7284 & 0 & -1.1862 & 0.9697 \\ 1.003 & 0 & -0.0071 & -0.2502 \end{bmatrix}$$

و

$$B=\begin{bmatrix} 0.0072 & 0.0196 \\ 0 & 0 \\ -5.9271 & 1.1922 \\ -0.1323 & -0.6066 \end{bmatrix}$$

مقادیر ماتریسها کاملاً شبیه هم هستند . اختلاف اندک ناشی از ارتفاع واقعی نمـی توانـد دقیقـاً یکسان شود و مشکلاتی در نگهداری هواپیما در سطح دقیق پرواز با مدل خطی بدست آمده بوجود خواهد آمد. تغییر اندک ، در جهت یابی تاثیر قابل توجهی خواهد گذاشت.

استفاده از شبیه سازها برای تست کنترل کننده های جدید یک مرحله مهم در طراحی کنترل می باشد . دانشگاههای زیادی این مرحله را جدیداً در کلاسهای کنترل مطرح کرده اند تا اینکه این موضوع را در معرض نمایش قرار دهند . در گرجستان دانشجویان انجمن تکنولوژی PE4525 امکان استفاده از شبیه ساز پرواز برای تست کنترل کننده های چند متغیره را که آنها طراحی کرده اند دارند . در سپتامبر ۲۰۰۳ دانشجویان همچنین شانس تولید مدل های خطی برای هواپیما با شرایط پرواز گوناگون را قبل از اینکه آنها کنترل کننده ها را طراحی کنند شروع کردند .

فصل سوم

سیستم های کنترل دیجیتال

بسیاری از سیستم های کنترل صنعتی کامپیوترها را به عنوان جزء مکمل عملیات خود بکار می برند . روندهای اخیر در زمینه کنترل دیجیتال سیستم های دینامیکی درمقایسه با کنترل آنالوگ ، اساسا بخاطر پیشرفتهای اخیر در کامپیوترهای دیجیتال و مزایای حاصل از کار کردن با سیگنالهای زمان - گسسته در مقایسه با سیگنالهای زمان- پیوسته حاصل شده است . دسترس پذیری میکروپروسسورها و میکرو کامپیوترهای ارزان قیمت ، روند جدیدی حتی برای سیستم های کنترل با مقیاس کوچک نیز به وجود آورده است که در آن برای بدست آوردن عملکرد بهینه از کامپیوترهای دیجیتالی استفاده می کنند. هدف اصلی این فصل بررسی تحلیل و طراحی سیستم های کنترل زمان – گسسته می باشد.

۳-۱-تعریف

شمای کنترلی را که در آن برای انجام پردازش سیگنال با شیوهٔ مطلوب، کامپیوتر دیجیتا لی در حلقه کنترل گنجانیده شود کنترل دیجیتال نامند .

۳-۲- مزایای کنترل دیجیتال یک فرآیند در مقایسه با کنترل آنالوگ :

۱) پردازش داده ها در کنترل کننده های دیجیتال سر راست است ، محاسبات کنترلی پیچیده را می توان به سهولت انجام داد .

۲) برنامه های کنترل (مشخصه های کنترل کننده) را در صورت نیاز می توان به سادگی تغییر داد.

۳) از دیدگاه نویز داخلی و اثرات رانش ، کنترل کننده های دیجیتالی به مراتب برتر از کنترل کننده های آنالوگ می باشند .

۳-۳-معایب کنترل دیجیتال :

۱) فرآیندهای نمونه برداری و کوانتیزه کردن به خطاهای بیشتری می انجامند که عملکرد سیستم را تنزل می دهند .

۲) طراحی کنترل کننده های دیجیتال برای جبران چنین تنزلی بمراتب پیچیده تر از طراحی کنترل کننده های آنالوگ در یک سطح عملکرد معادل می باشد .

کوانتیزه کردن :

کوانتیزه کردن یعنی تبدیل یک سیگنال پیوسته یا آنالوگ به دسته ای از حالتهای گسسته .

در این رابطه تبدیل تابع تبدیل یک سیستم از فضای پبوسته به فضای گسسته نیـز مـورد نظـر است .

خطای کوانتیزه کردن :

از آنجائیکه تعداد بیتها در کلمه دیجیتال محدود است تبدیل A/D به یک دقت محدود منجر می شود . یعنی خروجی دیجیتال فقط می تواند تعداد محدودی از مقادیر را اختیار کند و بنـابراین یک عدد آنالوگ باید به بزرگترین سطح دیجیتال گرد شود از این رو هر تبدیل A/D شامل خطای کوانتیزه کردن است . چنین خطـای کـوانتیزه کـردن بـین صـفر و $\pm 1/2Q$ ($Q = \frac{fsR}{2^n}$ ، fsr گستره مقیاس کامل است) تغییر می کند . این خطا به ظرفیت سطح کوانتیزه کردن بستگی دارد و با کوچکتر گرفتن سطح کوانتیزه کردن (یعنی با افزایش تعداد بیت n) می توان آنـرا هـر قـدر کـه خواست کوچکتر کرد . در عمل برای تعداد بیتها n ، حداکثری وجود دارد و بنابراین همیشه خطای ناشی از کوانتیزه کردن وجود دارد .

برای تعیین اندازه مطلوب سطح کوانتیزه کردن (یا تعـداد حالتهـای خروجـی) در یـک سیـستم کنترل دیجیتال ، باید درک خوبی از ارتباط میان انـدازه سـطح کـوانتیزه کـردن و خطـای حاصـل داشت. پراش نویز کوانتیزه کردن معیار مهمی از خطای کوانتیزه کردن است . زیـرا پـراش بـا تـوان متوسط همراه با نویز متناسب است . خطای حاصل از صـرفنظر کـردن از رقمهـای باقیمانـده را در محاسبات عددی خطای گرد کردن گویند . از آنجائیکه فرآیند کوانتیزه کردن یـک فرآینـد تقریـب زنی است که در آن کمیت آنالوگ با یک عدد دیجیتال محدود تقریب زده می شود کوانتیزه کـردن یک خطای گرد کردن است ، واضح است هر قدر سطوح کوانتیزه ظریف تر باشد خطای گرد کردن کوچکتر خواهد بود . برای بدست آوردن معادلهای زمان – گسسته یک سیـستم زمـان – پیوسـته چندین روش به طور معمول در دسترس است . معادل زمان- گسسته یک سیستم زمان – پیوسته تقریباً باید دارای مشخصات دینامیکی یکسان با سیستم زمان – پیوسته اصلی باشد . یعنـی در بـه دست آوردن معادل زمان – گسسته یک سیستم زمان- پیوسته مطلوب آن است که مشخصه های پاسخ گذرا و پاسخ فرکانسی هر قدر ممکن است به مشخصه های سیستم زمـان – پیوسـته اصـلی نزدیک تر باشد . لکن تحقق این امر ممکن است مقدور نباشد . در واقع ، در به کارگیری یک روش

معین گسسته کردن می توان مشخصه های پاسخ ضربه یکسان را به دست آورد در حالی که امکان ندارد در مشخصه های پاسخ فرکانسی همانندهی نسبتاً خوبی را بدست آورد یا برعکس .

در حالت کلی ، مشخصه های سیستم زمان – گسسته به فرکانس نمونه برداری و به روش خاص گسسته کردن بستگی دارد . با پایین آوردن فرکانس نمونه برداری همانندهی سیستم زمان گسسته کاهش می یابد . در بیشتر موارد عملی در گسسته کردن یک سیستم زمان- پیوسته ، طراح ممکن است بخواهد یک یا چند تا از خواص زیر در میان خواص دیگر را حفظ کند تعداد قطبها و صفرها ، مشخصه های پاسخ ضربه یا مشخصه های پاسخ پله ، بهرهٔ تقویت dc ، حاشیه های فاز و بهره یا پهنای باند ، برخی سیستم های گسسته شده از لحاظ مشخصات پاسخ ضربه تطابق خوبی با سیستم زمان- پیوسته ، اصلی دارند ولی برخی دیگر ندارند . بویژه ، در بیشترین حالتها تطبیق مشخصه های پاسخ فرکانسی یک سیستم زمان – گسسته با مشخصه های سیستم زمان – پیوسته اصلی مشکل است . در گسسته کردن سیستم زمان – پیوسته ، طراح باید تصمیم بگیرد که کدام مشخصه های دینامیکی در یک وضعیت خاص اهمیت بیشتری دارند ، سپس هدف باید حفظ آن مشخصه های مهم با انتخاب یک روش گسسته کردن مناسب باشد.درموردمشخصه های پاسخ فرکانسی، تذکراین مطلب حائز اهمیت است که اگر یک سیستم زمان- پیوسته گسسته شود و فرکانس نمونه برداری آنقدر پایین باشد که قضیه نمونه برداری برآورده نشود ، سیستم حاصل ممکن است پدیده نامطلوبی را نشان دهد که در سیستم زمان- پیوسته اصلی موجود نباشد . (مانند تا شدن فرکانسی) . به خاطر بسپارید که گسسته کردن سیستم های زمان – پیوسته ، نه تنها روش خاص به کار رفته بلکه فرکانس انتخاب شده نمونه برداری در مشخصه های دینامیکی سیستم حاصل تأثیر می گذارد . در صورتیکه فرکانس نمونه برداری به قدر کافی بالا باشد ، سیستم زمان -گسسته معادل ، تقریب خوبی برای سیستم زمان – پیوسته داده شده به دست خواهد داد . اما یک فرکانس نمونه برداری آهسته به تقریب ضعیفی می انجامد . وقتی که یک سیستم زمان – پیوسته گسسته سازی و کوانتیزه گردد یک سیستم دیجیتال معادل به دست می آید . چنین سیستم دیجیتالی را می توان به وسیله نرم افزار تحقق داد.

۳-۴-روشهای گسسته کردن سیستم زمان پیوسته :

روشهای مختلفی برای تبدیل از فضای پیوسته به فضای گسسته وجود دارد که در این بین می توان از روش تفاضل معکوس، تفاضل مستقیم، تبدیل دوخطی، تبدیل دو خطی با پیش تاب دادن فرکانسی ، تغییر نا پذیری پله، تغییر نا پذیری ضربه و نگاشت قطب- صفر تطبیق یافته نام برد. در ادامه بعضی از آنها توضیح داده شده اند.

۳-۴-۱-روش تفاضل معکوس:

$$\frac{y_{(s)}}{x_{(s)}} = \frac{1}{Rcs+1} = \frac{a}{s+a}$$

$$\Rightarrow \frac{dy}{dt} = -ay + ax \Rightarrow \int_{\circ}^{t} \frac{dy(t)}{dt}dt = -a\int_{\circ}^{t} y(t)dt + a\int_{\circ}^{t} x(t)dt$$

$$t = kt$$

$$\int_{\circ}^{kt} \frac{dy\,(t)}{dt}\,dt \;=\; -a\int_{\circ}^{kt} y\,(t)\,dt \;+\; a\int_{\circ}^{kt} x\,(t)\,dt$$

یا

$$y(kt) - y(0) = -a\int_{\circ}^{kt} y(t)dt + a\int_{\circ}^{kt} x(t)dt \qquad (۱-۳)$$

با تغییر KT به $(k-1)T$

$$y((k-1)T) - y_{(0)} = -a\int_{\circ}^{(k-1)t} y(t)dt + a\int_{\circ}^{(k-1)T} x(t)dt \qquad (۲-۳)$$

از معادله (۳-۱) و (۳-۲) نتیجه می شود

$$y(KT) - y((k-1)T) = -a\int_{(k-1)T}^{kT} y(t)dt + a\int_{(k-1)T}^{kT} x(t)dt \qquad (۳-۳)$$

با استفاده از انتگرال گیری با روش تفاضل معکوس :

$$\int_{(k-1)t}^{kt} dt = y(KT)T$$

$$\int_{(k-1)T}^{kT} x(t)dt \approx x(kT)T$$

می توان نتیجه گرفت :

$$y(kT) = y((k-1)T) - aT[(y(kT) - x(kT)] \qquad (۴-۳)$$

تبدیل Z معادله به شکل زیر می شود .

$$y(z)=z^{-1}y(z)-aT\left[y(z)-x(z)\right]$$

از معادله قبل نتیجه می شود :

$$y(z)=z^{-1}y(z)-aT\left[y(z)-x(z)\right]$$

$$\Rightarrow \frac{y(z)}{x(z)}=G_D(z)=\frac{aT}{1-Z^{-1}+aT}=\frac{a}{\frac{1-z^{-1}}{T}+a} \qquad (۵-۳)$$

ملاحظه می شود که اگر $S=\frac{1-Z^{-1}}{TZ^{-1}}$ تبدیل شود سیستم زمان پیوسته به یک سیستم زمان گسسته تبدیل می شود .

۳-۴-۲-روش تبدیل دو خطی :

روش تبدیل دو خطی روش انتگرال گیری ذوزنقه یا روش تبدیل توستین نیز گفته می شود .

در این روش

$$\int_{(k-1)T}^{kT} y(t)dt \approx \frac{1}{2}\left[\left(y[kt)+y((k-1)T)\right)\right]T$$

و

$$\int_{(k-1)T}^{kT} x(t)dT \approx \frac{1}{2}\left[x(kT)+x((k-1)T\right]T$$

با استفاده از روش تبدیل دو خطی معادله (۳-۳) رامی توان به صورت زیر نوشت :

$$y(kT)=y((k-1)T)-\frac{aT}{z}\left[y(kT)+y((k-1)T)+\frac{aT}{z}[x[kT)+x((k-1)T)\right] \qquad (۶-۳)$$

تبدیل z معادله به شکل زیر می باشد :

$$y(z)=z^{-1}y(z)-\frac{aT}{z}\left[y(z)+z^{-1}y(z)\right]+\frac{aT}{z}\left[x(z)+z^{-1}x(z)\right]$$

$$\frac{y(z)}{x(z)}=G_D(z)=\frac{\frac{aT}{z}\left(1+z^{-1}\right)}{\left(1-z^{-1}\right)+\frac{aT}{z}\left(1+z^{-1}\right)}=\frac{a}{\frac{z}{T}\frac{1-z^{-1}}{1+z^{-1}}+a} \qquad (۷-۳)$$

۳-۴-۳-روش تغییر ناپذیری پله :

سیستم زمان پیوسته $G(s)$ را در نظر بگیرید که در آن

$$G(s)=\frac{a}{s+a}$$

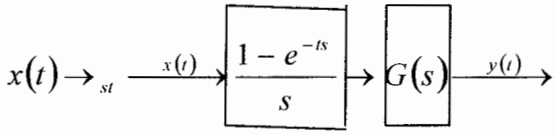
فیلتر زمان گسسته $G_D(z)$ معادل فیلتر زمان –پیوسته $G(s)$ به دست آمده با روش تغییر ناپذیری پله فیلتری است که پاسخ پله فیلتر زمان – پیوسته اصلی در لحظه های نمونه برداری یکسان باشد . یعنی

$$Z^{-1}\left[G_D(z)\frac{1}{1-Z^{-1}}\right]=L^{-}\left[G(s)\frac{1}{s}\right]_{t=kT} \qquad (۸-۳)$$

که در آن $G_D(z)\left[1/(1-z^{-1})\right]$ نمایشگر پاسخ پله $G_D(z)$ و $G(s)/s$ نشان دهنده پاسخ پله $G(s)$ است با گرفتن تبدیل Z این معادله اخیر ، به دست می آید .

$$G_D(z)\frac{1}{1-z^{-1}}=Z\left\{L^{-}\left[\frac{G(s)}{s}\right]\right\}=Z\left[\frac{G(s)}{s}\right] \qquad (۹-۳)$$

می توان ملاحظه کرد که سمت راست معادله اخیر تبدیل Z فیلتر $G(s)$ است که پیش از آن یک نمونه بردار ضربه ای و یک افزاره نگهدار قرار دارد . بنابراین ، می توان چنین فرض کرد که سیگنال ورودی زمان – پیوسته $x(t)$ نخست به یک افزاره نمونه بردار و نگهدار ساختگی طبق شکل ۳-۱ داده می شود ، سپس فرض می کنیم که خروجی نگهدار به فیلتر زمان - پیوسته $G(s)$ خورانده می شود نسبت تبدیلهای Z خروجی $y(t)$ و ورودی $x(t)$ فیلتر زمان گسسته ، $G_D(z)$ را به آن می دهد .



شکل۳-۱-بلوک دیاگرام نمونه برداروونگهدار

با توجه به روشهای بررسی شده می توان نتیجه گرفت که پاسخ میان لحظه های نمونه برداری برای هر روش گسسته کردن به کار رفته متفاوت می باشد .

علاوه بر این ، هیچ یک از این سیستم های زمان – گسسته معادل نمی تواند همانندهی کامل داشته باشند . در فرکانس $\omega=\frac{1}{2}\omega_s=\frac{\pi}{T}$ یا نزدیک آن اختلافهای قابل ملاحظه در پاسخ میان

سیستم زمان پیوسته اصلی و معادلهای زمان- گسسته آن وجود دارد . ($\frac{1}{2}\omega_s$ بالاترین فرکانسی است که در پاسخ زمان- گسسته یا سیستم کنترل دیجیتال در نظر گرفته می شود)

اگر طراحی یک سیستم کنترل زمان – گسسته از طریق روشهای طراحی زمان – پیوسته اقدام شود و یک سیستم زمان – گسسته معادل به جای یک سیستم زمان – پیوسته (آنالوگ) به کار رود در آن صورت مصلحت است که قبل از اینکه طراحی کامل شود طراح چند شکل متفاوت سیستم زمان- گسسته معادل را آزمایش کند . از آنجایی که روش نگاشت قطب – صفر تطبیق یافته روش تبدیل دو خطی ، روش تبدیل دو خطی با پیش تاب دادن فرکانسی معمولاً نتایج رضایت بخشی می دهند یکی از این روشها در آزمایش اول فرآیند طراحی می تواند به کار برده شود . بنابراین ازروش زیربرای تبدیل سیستم زمان پیوسته به سیستم زمان گسسته ومحاسبه ماتریسهای معادل زمان گسسته استفاده می شود.

برای سیستم زمان- پیوسته درفضای حالت داریم:

$$\begin{cases} \dot{X}(t) = AX(t) + BU(t) \\ Y(t) = CX(t) + DU(t) \end{cases} \qquad (۱۰-۳)$$

$$Y(t) = CX(t) + DU(t)$$

برای معادله (۱)پاسخ برابراست با:

$$X(t) = e^{At}X(0) + \int_0 e^{A(t-\tau)}BUd\tau \qquad (۱۱-۳)$$

اگرUرایک ورودی ثابت(Step)درنظربگیریم داریم:

$$X(t) = e^{At}X(0) + \int_0^t e^{A(t-\tau)}Bd\tau.U \qquad (۱۲-۳)$$

حال اگر $X(KT)$ راداشته باشیم برای $X(KT+T)$ داریم:

$$X(KT+T) = e^{At}X(KT) + \int_{KT}^{KT+T} e^{A(KT+\tau-C)}Bd\,\tau.U(KT) \qquad (۱۳-۳)$$

یا:

$$X[K+1] = e^{AT}X[K] + \int_0^T e^{A\tau'}Bd\,\tau'.U[K] \qquad (۱۴-۳)$$

و

$$Y[K] = C[K] + DU[K] \qquad (۱۵-۳)$$

بطورکلی معادل گسسته یک سیستم پیوسته باتغییرمتغیر $\tau' = KT + T - \tau$ به شکل زیرمیباشد:

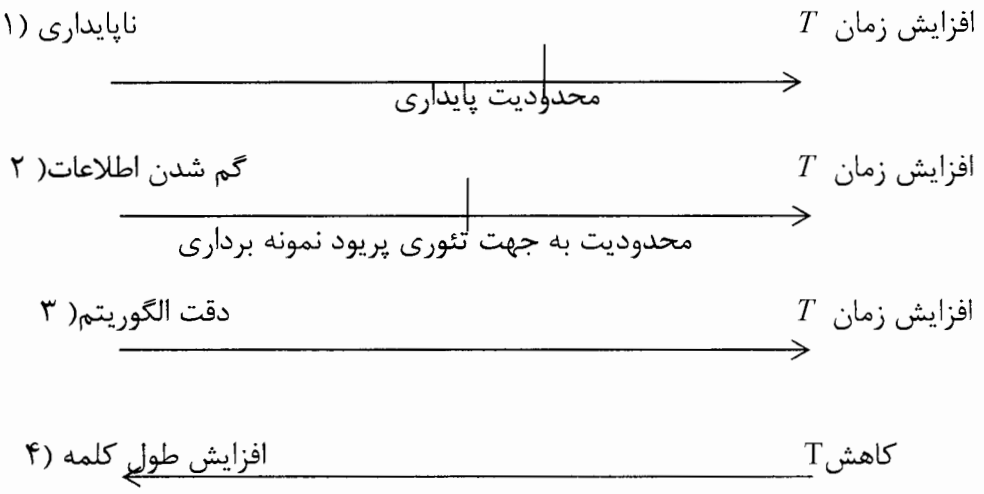
$$\begin{bmatrix} A' = e^{AT'} \\ B' = \int_0^{T'} e^{A\tau'} B d\tau \\ C' = C \\ D' = D \end{bmatrix} \qquad (۱۶-۳)$$

۳-۵- پریود نمونه برداری :

از مواردی که بایستی در انتخاب آن دقت نمود پریود نمونه برداری است . انتخاب بزرگ آن سرعت محاسبات و هزینه سخت افزار را کاهش می دهد اما محدودیتهای دیگری نیز در افزایش زمان نمونه برداری وجود دارد . نوع پروسس، نوع الگوریتم، لوازم کنترل و مشخصات ورودی و نویز روی ماکزیمم مقدار T اثر می گذارند .

از طرف دیگر T بزرگ می تواند باعث ناپایداری سیستم شود . هر چه T کوچکتر انتخاب شود سیستم پایدارتر می گردد . همینطور با افزایش T حساسیت به تغییر پارامترهای سیستم دینامیکی افزایش می یابد . بنابراین عامل دیگری در انتخاب T کوچک ، کم کردن حساسیت نسبت به پارامترها می باشد. عامل مهم دیگر در انتخاب T کوچک توانایی سیستم حلقه بسته در رفع اغتشاشهای ناخواسته است . اگر فرکانس نمونه برداری نسبت به فرکانسهای موجود در اغتشاشهای ناخواسته خیلی کوچک باشد سیستم حلقه بسته نسبت به این اغتشاشها اساساً مانند سیستم حلقه باز رفتار می نماید . اگرخواسته شود کنترل کننده ها در رفع این اغتشاشها مؤثر باشند باید حتی الامکان فرکانس نمونه برداری را به بالاترین فرکانسهای نویز نزدیک کرد . شکل ۳-۲ سه عامل اصلی مخرب افزایش T را نشان می دهد که به توضیح مختصر هر یک پرداخته خواهد شد.:

۱) ناپایدار ی : بعلت وجود زمان تبدیل و محاسبات الگوریتمهای دیجیتالی ، زمان مرده ای بوجود می آید که تشابه بین سیگنال آنالوگ و دیجیتال را از بین می برد . این زمان بخاطر اینکه شیفت فازی بوجود می آورد باعث ناپایداری می شود .



شکل ۳-۲- اثرات افزایش و کاهش پریود نمونه برداری

۲) گم شدن اطلاعات : قضیه معروف نایکوئیست می گوید جهت اینکه بتوان یک سیگنال دیجیتال را بازسازی نمود اینست که فرکانس نمونه برداری دو برابر ماگزیمم فرکانس سیگنال باشد. لذا اگرفرکانس نمونه برداری کمتر از دو برابر فرکانس سیگنال باشد اطلاعاتی از سیگنال حذف خواهد شد .

۳) کم شدن دقت : تعدادی از کنترلرهای دیجیتال و الگوریتمها از الگوریتمهای پیوسته آنالوگ با گسسته نمودن آنها مشتق شده اند . تبدیل یک الگوریتم از پیوسته به دیجیتال خطاها را افزایش می دهد . بطور معمول این خطاها هنگامی که زمان نمونه برداری افزایش می یابد زیاد می شود . با توجه به موارد فوق در افزایش پریود نمونه برداری محدودیت وجود دارد همچنین نمی توان آنرا خیلی کوچک انتخاب کرد . کران پایین این پریود را اولاً زمان پردازش و ثانیاً محدودیت طول کلمه تعیین خواهد کرد همانطور که T کوچکتر می شود نمونه ها مقادیری نزدیک به هم پیدا می کنند . بخاطر حفظ اختلاف بین نمونه ها طول کلمه بیشتری لازم است که اگر این مورد رعایت نشود سیستم ناپایدار خواهد شد .

۳-۵-۱- انتخاب پریود نمونه برداری بصورت عملی

۱- فرض کنید یک پروسس ثابت زمانی $Tdom$ را داشته باشد بایستی پریود نمونه برداری رابطه زیر را برآورده نماید .

$$T\langle Tdom/10 \qquad (۱۷-۳)$$

۲- فرض کنید پروسسی ، تابع تبدیلی بفرم زیر داشته باشد .

$$G(s)=e^{-T1.S}/(1+T_2.S) \qquad (۱۸-۳)$$

T بایستی در رابطه زیر صدق کند .

$$T\langle T_1/4 \qquad (۱۹-۳)$$

۳- فرض کنید پروسسی تحت کنترل حلقه بسته زمان نشست T_s و فرکانس طبیعی ω_n را داشته باشد لذا بایستی روابط زیر بر آورده شوند .

$$T\langle Ts/10 \qquad (۲۰-۳)$$

$$\omega\rangle 10\omega_n \qquad (۲۱-۳)$$

۳-۶- تبدیل Z و تحقق آن :

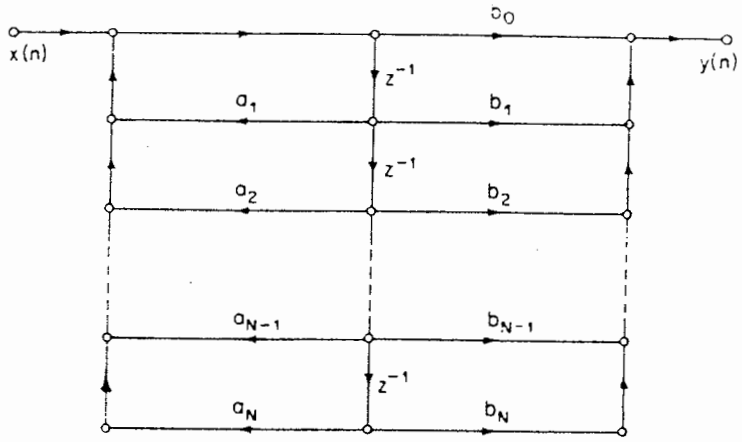
همانطور که در توابع پیوسته از تبدیل S برای تجزیه و تحلیل سیستمها استفاده می شود برای تجزیه و تحلیل سیگنالهای گسسته بایستی از تبدیل Z استفاده کرد . در اینجا بحثی از تبدیل Z ننموده و تنها نتایج و نحوه کاربرد مطرح می گردد . برای اطلاعات بیشتر می توان از مراجع مربوطه استفاده نمود . برای تجزیه و تحلیل سیستم ابتدا تبدیل S سیستم مورد نظر را استخراج نموده و سپس با استفاده از روشهای مرسوم معادل تبدیل Z آنرا استخراج کرد و سپس آنرا به روش دیجیتالی محقق ساخت . برای تحقق این چنین سیستمهایی که از نوع IIR می باشند روش مستقیم نوع II انتخاب می گردد. فرض می شود $H(Z)$، تابع تبدیلی به فرم رابطه ۳-۲۲ داشته باشد .

$$H(z)=\frac{\sum_{K=0}^{M} b_k.Z^{-k}}{1-\sum_{K=1}^{N} a_k.Z^{-K}} \qquad (۲۲-۳)$$

که معادل آن در حوزه زمان گسسته بفرم رابطه ۳-۲۳خواهد بود .

$$Y(n)=\sum_{K=1}^{N} a_k\, Y[n-k]+\sum_{K=0}^{M} b_k\, X[n-k] \qquad (۲۳-۳)$$

بلوک دیاگرام تحقق این روش بصورت شکل ۳-۳ می باشد .



شکل۳-۳- پیاده سازی تابع تبدیل Z به روش مستقیم نوع II

در این تحقیق با در اختیار داشتن تابع تبدیل سیستم در فضای پیوسته ، چون برای پردازش سیستم در پردازنده ، پارامترهای گسسته سیستم مورد نیاز می باشد لذا سیستم مورد نظر را به حالت گسسته تبدیل می کنیم . برای بدست آوردن تابع تبدیل Z آن راههای گوناگونی وجود دارد و ما بدین منظور از نرم افزار مطلب کمک می گیریم اما قبل از آن بایستی مسائل زیر روشن باشند .

۱- پریود نمونه برداری چقدر باشد .

۲- تقریب تابع تبدیل تا چه حدی قابل قبول است .

۳- آیا پاسخ سیستم در حالت پیوسته و گسسته یکی است .

۳-۷-فیلتر :

در این طرح چون از PWM به عنوان مبدل آنالوگ به دیجیتال استفاده شده است ، از طرفی چون در خروجی PWM فرکانس پالس PWM مشاهده میشود ، لذا برای حذف فرکانس PWM و مشاهده فرکانس سیستم مورد نظر در خروجی از یک فیلتر پایین گذر استفاده می شود . فیلترها یکی از موارد اساسی در سیستم ها می باشند که باید طراحی شوند . همه کسانیکه با مدارات الکترونیکی سر و کار دارند کما بیش به اهمیت مبحث فیلتر واقفند . اصولاً کمتر مداری است که نوعی فیلتر در آن بکار نرفته باشد ، فیلترها علاوه بر تصحیح شکل موج با انتخاب برخی فرکانسها از ورود سیگنالهای ناخواسته (نویز) نیز جلوگیری می نمایند . در پاره ای از سیستمها ، فیلترها سازنده شکل موجهای خاصی هستند . با وجود نیاز مبرم به فیلتر در شبکه های الکترونیکی ، مخابراتی ، کامپیوتر و حتی سیستمهای قدرت و علی رغم موجود بودن المانهای مورد نیاز فیلتر در بازار

بواسطه پیچیدگی محاسبات ریاضی و تنوع انواع مورد استفاده در سیستمهای مختلف عملاً طراحی فیلتر مشکل و برای عده ای غیر ممکن می نماید .

۳-۷-۱- مقدمه ای بر فیلترها :

هنوز هم خیلی از سازندگان وسایل الکترونیکی بدرستی با طرز کار و محاسبه فیلترها آشنایی کامل ندارند . در مطالب ارائه شده سعی بر آن است تا طرز عمل انواع فیلترهای معمول که عمدتاً مهندسین با آنها مواجه هستند شرح داده شود .

تصور وسیله ای الکترونیکی مثل وسایل صوتی ، وسایل با فرکانس بالا ، تلویزیون و نظایر آن بدون داشتن نوعی از انواع فیلتر مشکل بنظر می رسد . فیلتر عبارت است از شبکه هایی که دارای تعدادی امپدانس است که مجموعاً تشکیل یک سیستم را می دهند و رفتار این سیستم بستگی به مقاومتها ، خازنها و سلف هایی که از آنها ساخته شده است دارد . شبکه ها را می توان بسته به شکل ترکیب مدارات آنها به شبکه های $T-\pi-L$ طبقه بندی نمود. فیلترها را می توان براساس پارامترهای فیزیکی به دو نوع پسیو (فیلترهای $Stripline, Rc, RLc, lc, LR$ - تشدید کننده های سرامیکی) و اکتیو تقسیم کرد .

۳-۷-۲- انواع اساسی فیلترها :

۱) فیلتر پایین گذر: در این فیلتر تمام سیگنالهای DC تا فرکانس مشخصی ، که فرکانس قطع نامیده می شود عبور داده می شوند . فرکانسهای بیشتر از فرکانس قطع تضعیف شده یا بکلی قطع می شوند .

۲) فیلتر بالا گذر: در این فیلتر تمام امواج با فرکانسهای پایین تر از فرکانس قطع تضعیف یا حذف می شوند .

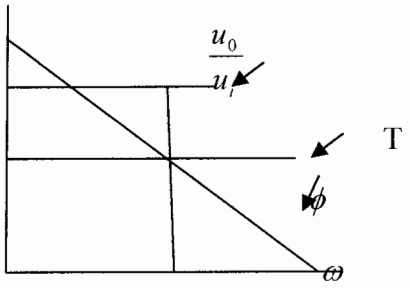
۳) فیلتر میان گذر: در این فیلترها تمامی فرکانسها بین دو فرکانس مشخص که به ترتیب فرکانس قطع بالا و فرکانس قطع پایین نامیده می شوند عبور می کنند .

۴) فیلتر میان حذف : در این فیلتر تمام فرکانسهای بین دو فرکانس ، حذف شده و فرکانسهای خارج از این محدوده عبور داده می شوند .

۵) فیلتر تمام گذر : این نوع فیلتر تمام سیگنالها را با هر فرکانسی عبور می دهد ، اما یک انتقال فاز بصورت تابعی از پارامترهای شبکه ایجاد می نماید .

۳-۷-۳- فیلتر ایده آل :

فیلتر ایده آل فیلتری است که تمامی سیگنالهای با فرکانس ما بین دو فرکانس معین را از خود عبور دهد و بقیه سیگنالها را کاملاً حذف نماید .



شکل ۳-۴- :باند عبوریک فیلترایده آل

فیلتر ایده آل دارای تأخیر زمانی تعریف شده ای بین ورودی و خروجی خود میباشد که به فرکانس بستگی ندارد . خط افقیT در شکل ۳-۴ بیانگر این تأخیر زمانی است .

از روی دو خط افقی T و $\frac{u_0}{u_i}$ می توان زاویه انتقال فاز ϕ را به شکل خطی راست مشاهده نمود تأخیر زمانی T را از روابط زیر می توان محاسبه نمود .

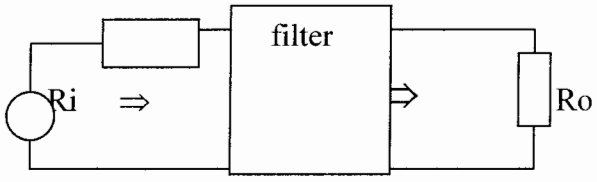
$$T=\frac{B}{\omega} \quad \text{یا} \quad T=\frac{\phi}{f} \qquad (۲۴-۳)$$

در این رابطه ϕ زاویه انتقال فاز برحسب درجه B زاویه انتقال فاز برحسب رادیان ، f فرکانس و $\omega=2\pi f$ می باشد .

برخی تئوریهای شبکه :

شکل ۳-۵ نمایش عمومی یک فیلتر می باشد که در مدارات مختلف بکار می رود .

دو مقاومت R_o, R_i ارتباط مهمی با عملکرد فیلتر دارند ، اما اگر در ورودی و خروجی فیلتر ازمدارات بافراستفاده شود،این مقاومتها اثری روی کارفیلتر نخواهند داشت .



شکل ۳-۵-نمایش عمومی یک فیلتر

هنگام طراحی فیلتر بدون استفاده از مدارات بافر مقادیر R_o, R_i باید معلوم باشند .

۳-۸- معرفی انواع فیلتر و پارامترهای آن :

هر فیلتر دارای خاصیت مخصوصی است که توسط پارامترهای آن بیان می گردد . در این قسمت درباره این گونه پارامترها بحث خواهد شد .

خواص فیلتر توسط تعدادی از پارمترها مشخص می شود یکی از این پارامترها مشخصه یا منحنی پاسخ فرکانسی است طراح با توجه به آنچه نیاز می باشد میزان شیب منحنی فیلتر و ریپل باند عبور را انتخاب می کند . این مورد در فیلتر بالا گذر یا باند گذر تفاوتی نمی کند .

هر نوع فیلتری را می توان به یک فیلتر پایین گذر با فرکانس قطع 1hz تبدیل کرد . برای این منظور مدار مورد نظر باید به فیلتر پایین گذر نرمالیزه تبدیل گردد. سپس می توان آن را با منحنی های استاندارد موجود برای فرکانس قطع 1hz مقایسه نمود . زمانی که پارامترهای مورد نظر انتخاب شد ، فیلتر برای فرکانس مطلوب بسادگی قابل محاسبه است. در مورد نوع فیلتر یکی از موارد زیر میتواند اختیار گردد .

۱) باتروث *Batterwoth*

۲) بسل *Bessel*

۳) چبی شف *Chebyshev*

۴) گذرا *Transitin*

۵) فاز خطی *Linear – Phase*

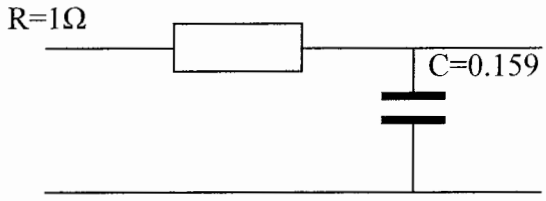
۶) تنظیم شده سنکرون *Synchronous – Tuned*

۷) تابع الیپتیک *elliptic – function*

به غیر از نوع آخر (تابع الیپتیک) تمامی انواع ذکر شده دارای مشخصه فرکانسی نرمالیزه شده برای نقطه قطع $3dB-$ در فرکانس 1hz هستند .

پارامترهای فیلتر :

بعنوان یک مثال در مورد نحوه عمل یک فیلتر ، ساده ترین نوع در نظر گرفته می شود که عبارتست از یک شبکه RC که در شکل ۳-۶ نشان داده شده است .



شکل ۳-۶- :یک شبکه سادهRCبانقطه قطع۳- دسی بل در۱هرتز

این شبکه به یک امپدانس بی نهایت و به یک منبع ولتاژ با مقاومت داخلی صفروصل شده است. خازن یک المان وابسته به فرکانس است و باعث تغییر فاز می گردد و تابع تبدیل این فیلتر عبارتست از :

$$T_{(jw)} = \frac{1}{1 + jwcR} \qquad (۲۵-۳)$$

قدر مطلق تابع عبارتست از :

$$\left|J_{(j\omega)}\right| = \frac{1}{\sqrt{1 + (\omega cR)^2}} \qquad (۲۶-۳)$$

تغییر فاز حاصل عبارتست از :

$$\phi = -ar\tan(\omega Rc) \qquad (۲۷-۳)$$

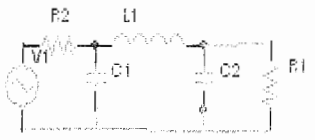
۳-۹- آشنایی با فیلتر های پسیو و اکتیو :

یک فیلتر میتواند از اجزاء پسیو نظیر مقاومت ، خازن و سلف ساخته شود یا می تواند شامل المانهای اکتیوی باشد که جایگزین سلفها می گردند .

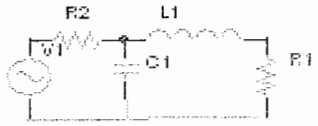
۳-۹-۱-فیلترهای پایین گذر پسیو :

۱) فیلتری که امپدانسهای ترمینالهای خروجی و ورودی آن مساوی هستند .

این نوع فیلترها معمولاً به شکل π یا T ساخته می شوند .



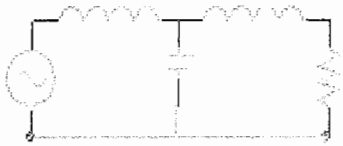
(a



(b

شکل ۳-۷- فیلترهای پایین گذرپسیو: (a) نوع π (b) نوع T

۲)فیلتری که به یک مولد سیگنال با مقاومت داخلی خیلی کوچک متصل شده است . و به یک امپدانس منتهی می گردد .



شکل ۳-۸-فیلترپسیومتصل شده به منبع سیگنال بامقاومت داخلی خیلی کم که به امپدانس منتهی میگردد.

۳-۹-۲- فیلترهای پایین گذر اکتیو :

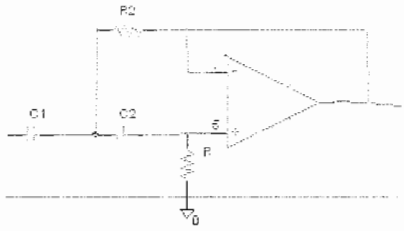
سه نوع فیلتر پایین گذر اکتیو وجود دارد :

!) فیلتر نوع ولتاژ پیرو

۲) فیلترهای دو قطبی دارای ضریب و تقویت

۳) فیلتر متغیر حالت

۳-۹-۳- فیلتر بالا گذر اکتیو :



شکل ۳-۹-یک نوع فیلتر بالاگذراکتیوباتقویت کننده عملیاتی

۳-۹-۴- فیلترهای میان گذر اکتیو :

در این طرح ابتدا باید ظرفیت خازن با توجه به مقادیر استاندارد انتخاب شده و سپس مقاومتها محاسبه شوند .یکی ازمشکلات این طرح پیچیدگی زیادمحاسبه فیلترهای میان گذراست.

لذاازنرم افزارجهت طراحی فیلتراستفاده می شود.

فیلتر با تروث :

شهرت این فیلتر مدیون پاسخ هموار دامنه در باند عبور و شیب مناسب دامنه بعد از فرکانس قطع میباشد و یکی از نقاط ضعف این فیلتر غیر خطی بودن مشخصه فاز آن می باشد . شیب دامنه بعد از فرکانس قطع با دقت خوبی با رابطه 6n دسی بل براکتاو بیان می گردد . که n مرتبه فیلتر می باشد .فیلتر باتروث دارای خصوصیات میانه ای بین فیلتر بسل (کم شیب بودن دامنه در نقطه قطع در مقابل پاسخ خطی فاز) و فیلتر چپی شف (شیب تند دامنه در نقطه قطع ، پاسخ فاز ضعیف وریپل در باند عبور) می باشد . برای کاربردهایی که به یک باند عبور هموار و صاف و شیب تند دامنه در نقطه قطع احتیاج باشد بدون شک فیلتر باتروث بهترین انتخاب است .

فصل چهارم

بررسی پردازنده(میکروکنترلر) سیستم شبیه ساز سخت افزاری مورد استفاده

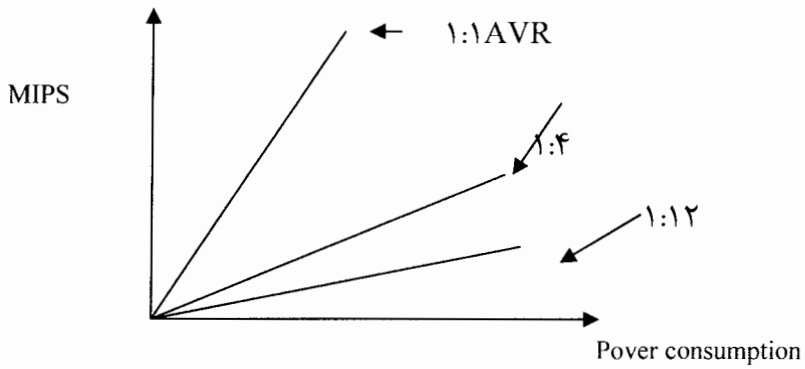
در بخش پردازنده احتیاج به یک میکرو پروسسور می باشد که بتواند خواسته های ما را بر آورده کند برای نیل به این هدف و کاهش هزینه و کار آیی بالا از یک میکروکنترلر جدید ، متعلق به شرکت ATMEL به نام میکروکنترلر AVR از نوع *ATmega*128 استفاده شده است .

این میکروکنترلرهای هشت بیتی به علت وجودکامپایلرهای قوی به زبان HLL کارآیی بالایی دارند . امروزه زبانهای سطح بالا به سرعت در حال تبدیل شدن به زبان برنامه نویسی استاندارد برای میکروکنترلرها می باشند. ولی در اکثرکاربردها کدهای بیشتری را نسبت به زبان برنامه نویسی اسمبلی تولید می کنند .ATMEL با ایجاد تحولی در معماری ، جهت کاهش کد به مقدار مینیمم روشی را معرفی کرد که نتیجه این تحول میکروکنترلرهای AVR هستند که علاوه برکاهش و بهینه سازی مقدار کدها به طور واقع عملیات را تنها در یک کلاک سیکل توسط معماری RISC انجام می دهند و از ۳۲ رجیستر همه منظوره (ACCUMULATORS) استفاده می کنند که باعث شده ۴ تا ۱۲ بار سریعتر از میکروهای مورد استفاده کنونی عملیات پردازش را انجام دهند.

AVR ها به عنوان میکروهایRISC با دستورات فراوان طراحی شده اند که باعث می شود حجم کد تولید شده کم و سرعت بالاتری بدست آید.

اکثر میکروها کلاک اسیلاتور به سیستم را با نسبت۱:۴یا۱:۱۲تقسیم می کنندکه خود باعث کاهش سرعت می شود ، ولی با انجام عملیات تک سیکل در AVRکلاک اسیلاتور با کلاک داخلی سیستم یکی می شود و هیچ تقسیم کننده ای در داخل میکرو قرار ندارد که ایجاد اختلاف فاز کلاک کند. بنابراین AVR ها ۴تا ۱۲برابر و مصرف آنها نیز ۴تا۱۲برابر نسبت به میکروکنترلرهای مصرفی کنونی کمتر است.

نمودار زیر افزایش MIPS را به علت انجام عملیات تک سیکلAVR(نسبت۱:۱) در مقایسه با نسبتهای۱:۴و ۱:۱۲در دیگرمیکرو ها نشان میدهد.



شکل۴-۱-نمودارمقایسه افزایشMIPS/Power consumption

۴-۱-فهرستی از ویژگی های *ATmega*128

- کارآیی بالا و توان مصرفی پایین
- از معماری AVR RISC استفاده می کند .
- دارای ۱۳۳ دستورالعمل با کارآیی بالا که اکثراً تنها در یک سیکل کلاک اجرا می شود .
- 8*32 رجیستر کاربردی
- سرعتی تا *16MIPS* در فرکانس *16MHZ*
- حافظه برنامه و داده غیر فرار
- 128 کیلو بایت حافظه *FLASH* داخلی قابل برنامه ریزی
- پایداری حافظه *FLASH* : قابلیت 1000 بار نوشتن و پاک کردن *(WRITE / ERASE)*
- 4 کیلو بایت حافظه *SRAM* داخلی
- قابیلت آدرس دهی *64KB* (کیلوبایت) حافظه خارجی
- 4 کیلو بایت حافظه *EEPROM* داخلی قابل برنامه ریزی
- پایداری حافظه *EEPROM*: قابلیت 100,000 بار نوشتن و پاک کردن *(WRITE / ERASE)*
- قفل برنامه *FLASH* و حفاظت داده *EEPROM*
- قابلیت ارتباط *JTAG* (مطابق استاندارد *IEEE*)
- قابلیت برنامه ریزی برنامه *FLASH* ، *EEPROM* ، *BIT* ، *FUSE* ، *LockBITS* از طریق *JTAG*

۴-۲- خصوصیات جانبی

- دارای دو تـایمر کـانتر (*TIMER / Counter*) ، ۸ بیتـی بـا Pr*escaler* مجـزا و دارای مدهای *Compare*

- دارای دو تایمر کانتر (*TIMER / Counter*) ،۱۶ بیتی با Pr*escaler* مجـزا و دارای مـدهای *Compare* ، *CAPTURE*

- ۲ کانال *PWM*، ۸ بیتی

- ۶ کانال *PWM* با قابلیت وضوح 2 تا 16 بیتی

- ۸ کانال مبدل آنالوگ به دیجیتال ده بیتی

-۸ کانال مبدل Single-ended

- ۷ کانال تفاضلی *ADC*

-دارای دو کانال با کنترل گین $200X,10X,1X$

- یک مقایسه کننده آنالوگ داخلی

-Watch dog قابل برنامه ریزی با اسیلاتور داخلی

- ارتباط سریال *SPI* برای برنامه ریزی داخل مدار (Programming insystem)

- قابلیت ارتباط سریال SPI به صورت Master یا Slave .

- قابلیت ارتباط با پروتکل سریال دو سیم Two-wire

- دو Usart قابل برنامه ریزی

۴-۳- خصوصیات ویژه میکرو کنترلر

- Power – Onreset circuit

- Brown – Out detection قابل برنامه ریزی

- انتخاب نرم افزاری فرکانس کلاک سیستم

- دارای شش حالت Sleep (Idle , Power – Save , Standby , ADC Noise Reduction , Power - Down , Extended standby)

- منابع وقفه (Interupt) داخلی و خارجی .

- دارای اسیلاتور RC داخلی کالیبره

- عملکرد کاملاً ثابت

- توان مصرفی پایین و سرعت بالا توسط تکنولوژی CMOS

۴-۴- ولتاژهای عملیاتی (کاری)

2.7V تا 5.5 برای (Atmega128L)

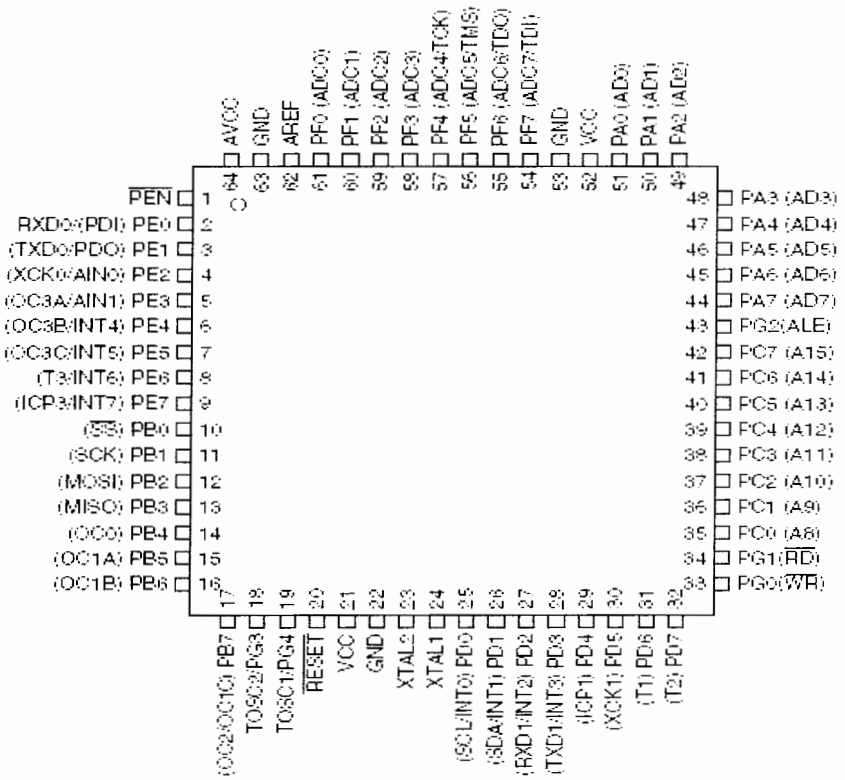
4.5V تا 5.5v برای (Atmega128)

۴-۵- خطوط I/O و انواع بسته بندی

53 خط ورودی / خروجی (I/O) قابل برنامه ریزی

64 – lead TQFPو64 – pad MLF

۴-۶- ساختار پایه ها



شکل ۴-۲- پایه های خروجی Atmega 128

توصیف پایه ها

VCC:

منبع تغذیه ولتاژ دیجیتال

GND :زمین

پورت A (PA7... PA0) :

پورت A یک پورت I/O دو طرفه ۸ بیتی با مقاومتهای $Pull-up$ (که برای هر بیت در نظر گرفته شده است) می باشد .

بافرهای خروجی پورت A ، دارای ویژگیهای متفاوتی می باشند .

زمانی که میکرو ریست می شود حتی اگر کلاک فعال نباشد پایه های پورت A ، tri – stated می باشند یعنی سه حالت را می توانند قبول کنند .

پورت B (PB7 ... PB0)

پورت B یک پورت I/O دو طرفه با مقاومتهای $Pull-up$ درونی (که برای هر بیت در نظر گرفته شده است) می باشد .

زمانی که میکرو ریست می شود پایه های پورت B سه حالت می توانند باشند حتی اگر هیچ کلاکی فعال نباشد .

پورت C (PC7 ... PC0)

پورت C یک پورت I/O دو طرفه با مقاومتهای $Pull-up$ درونی می باشد .

زمانی که میکرو در حالت ریست قرار می گیرد ، پایه های پورت C سه حالت می توانند قبول کنند . حتی اگر هیچ کلاکی فعال نباشد .

پورت D (PD7 ... PD0)

پورت D یک پورت I/O دو طرفه می باشد با مقاومتهای $Pull-up$ درونی زمانی که میکرو در شرایط ریست قرار می گیرد ، پایه های پورت D سه حالت را می توانند قبول کنند . حتی اگر هیچ کلاکی فعال نباشد .

پورت E (PE7 ... PE0)

پورت E یک پورت I/O دو طرفه با مقاومتهای $Pull-up$ درونی می باشد .

پایه های پورت E ، زمانی که میکرو در حالت ریست قرار دارد سه حالت می توانند قبول کند حتی اگر هیچ کلاکی فعال نباشد .

پورت F (PF7 ... PF0)

پورت F به عنوان ورودی های آنالوگ برای تبدیل A/D بکار مـی رود. اگـر پـورت F بـه عنـوان ورودی آنالوگ بکار نرود می توان از آن به عنوان پورت I/O دو طرفه استفاده کرد. اگر میکـرو در شرایط ریست قرار بگیرد حتی اگر هیچ کلاکی فعال نباشد ، پورت F می تواند در سـه حالـت قـرار بگیرد .

اگر رابط JTAG فعال باشد مقاومتهای $Pull-up$ در (TDI) PF7 ،

PF4(TDX),PF5(TMS) فعال خواهند شد حتی اگر ریست اتفاق بیافتد .

پورت G (PG7 ... PG0)

پورت G یک پورت I/O با پنج بیت با مقاومتهای $Pull-up$ درونی می باشد . پـورت G مـی تواند در سه حالت قرار بگیرد زمانی که میکرو در شرایط ریست باشد حتی اگر هـیچ کلاکـی فعـال نباشد .

: RESET

ورودی ریست که با صفر فعال می باشد .

: XTAL1

ورودی تقویت کننده اسیلاتور Inverting و ورودی کلاک مدار

: XTAL2

خروجی تقویت کننده اسیلاتور Inverting

: AVCC

AVCC ، پایه مرجع آنالوگ برای تبدیل A/D می باشد .

: PEN

PEN ، یک پایه قفل برنامه ریزی و برای SPI می باشد .

۴-۷- مبدل آنالوگ به دیجیتال ADC:

۴-۷-۱- خصوصیات ویژه

دارای ۱۰ بیت قدرت تشخیص

$\pm 2lSB$ دقت قطعی

$65-260\mu S$ زمان تبدیل

دارای ۸ کانال مبدل ورودی **Single – Ended**

دارای ۷ کانال تفاضل ورودی

دارای ۲ کانال تفاضل ورودی با گین های **10X و 200X**

دارای تنظیمات اختیاری برای خروجی **ADC** بین 0-vcc می باشد .

ولتاژ 2.56v برای ولتاژ مرجع (refrence) **ADC** قابل گزینش است .

دارای مد تبدیل **Single** یا مد **Free running**

دارای وقفه در تبدیل **ADC**

دارای مد **Sleep** برای حذف نویز

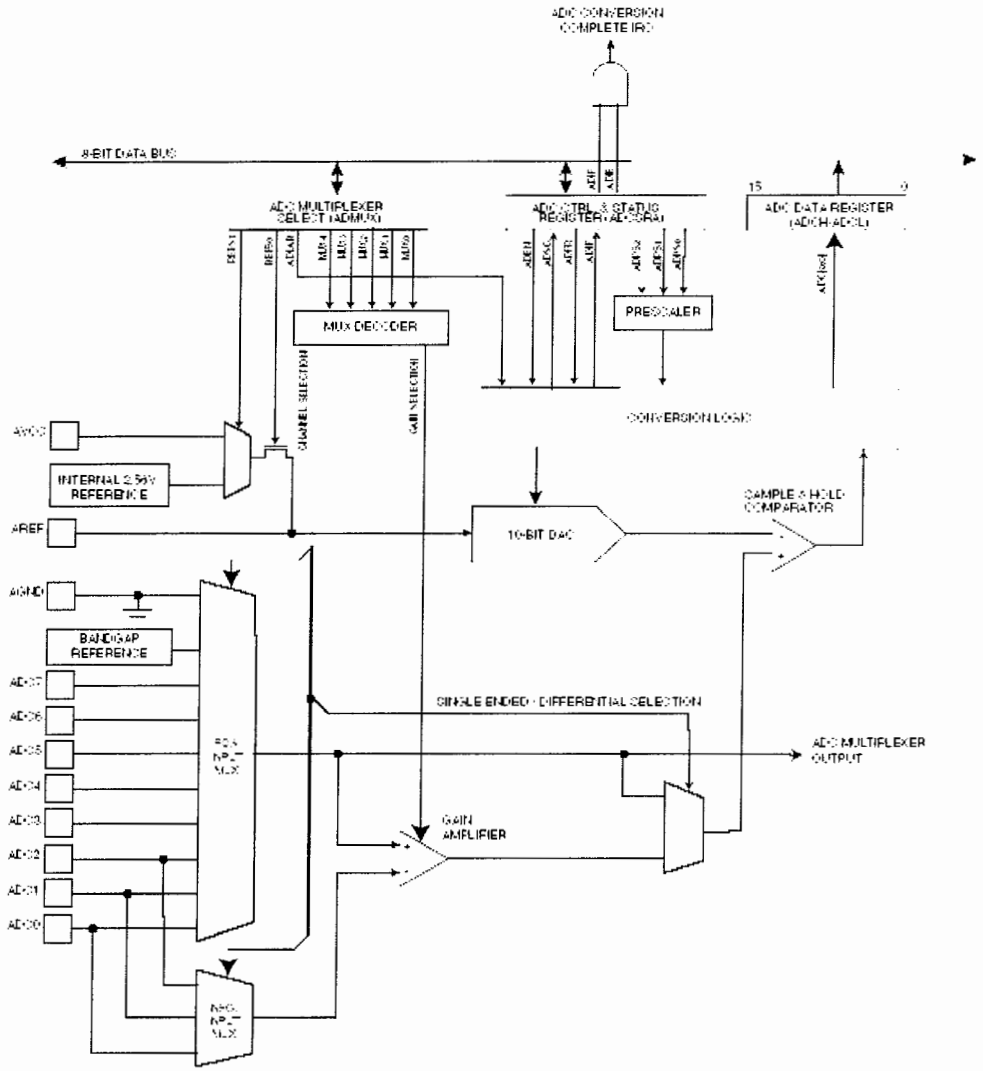
از خصوصیات ویژه ATMega 128 این است که دارای 10 بیت تقریباً متوالی برای **ADC** می باشد . **ADC** به یک کانال 8 تایی ترکیبی که 8 تا ولتاژ ورودی **Single – Ended** – که از طریق پورت F وارد می شود متصل است . این قطعه همچنین ترکیب 16 ولتاژ ورودی تفاضلی را پشتیبانی می کند .

دو ورودی تفاضلی (ADC1 – ADC0 – ADC3 – ADC2) با یک بهره قابل برنامه ریزی مجهز است بطوریکه ولتاژ ورودی A/D با بهره های 0dB (1X) ، 20dB(10x) یا 46dB(200x) قابل تقریب می باشند .

7 کانال آنالوگ ورودی تفاضلی تقسیم می شود به یک ترمینال منفی و پایه های ورودی دیگر به عنوان ورودی مثبت انتخاب می شوند .

اگر گین 1X یا 10X استفاده شود ، 8 بیت قدرت تشخیص را می توان انتظار داشت و اگر گین 200X استفاده شود 7 بیت قدرت تشخیص را می توان انتظار داشت .

ADC شامل : ۱ نمونه بردار ، ۲ مد نگه دار می باشد تا اطمینان یابد که ولتاژ ورودی به ADC در طول تبدیل در یک مقدار ثابت نگه داشته می شود بلوک دیاگرام ADC در شکل۴-۳ نشان داده شده است .



شکل ۴-۳-شماتیک بلوک دیاگرام تبدیل آنالوگ به دیجیتال

ADC دارای یک منبع تغذیه جدا می باشد با نام AVCC . VCC نباید بیشتر از $\pm 0.3v$ نسبت به VCC فرق داشته باشد . به شکل ۴- ۹ نگاه کنید که چطور AVCC و VCC به هم وصل شده اند . ولتاژ مرجع را نیز می توان در ساختار خارجی قطعه و از طریق پین AREF از طریق خازن برای حذف نویز ایجاد کرد .

۴-۷-۲- عملکرد

ADC ولتاژ ورودی آنالوگ را به یک مقدار دیجیتالی 10 بیتی تبدیل می کند که کمترین مقدار خروجی ADC مقدار GND یعنی صفر می باشد و ماکزیمم مقدار خروجی ADC ، مقدار معادل ولتاژی است که به پایه AREF منهای 1LSB می باشد بطور اختیاری می توان پایه AREF را به ولتاژ مرجع درونی (2.56V) و یا AVCC وصل کرد (با نوشتن در بیت های REFSn در ریجیستر ADMUX) ولتاژ مرجع را نیز می توان بوسیله یک خازن خارجی با اتصال به پایه AREF برای بهبود دادن به نویز مدار ، تغییر داد .

کانال ورودی آنالوگ و بهره تفاضلی با نوشتن در بیت های MUX در ریجیستر ADMUX انتخاب می شود . اگر کانال های تفاضلی انتخاب شوند گین تفاضلی اختلاف ولتاژ بین جفت کانال ورودی انتخاب شده را با گین انتخابی تقویت می کند . و ابتدا تقویت انجام می شود و بعد وارد مرحله ADC می شود اگر کانالهای **Single – Ended** استفاده شوند ، گین تقویتی کاملا کنار گذاشته می شود . ADC با یک کردن بیت ADC enable، فعال می شود که این بیت ADEN در ریجیستر ADCSRA می باشد .

ADC تا زمانی ADEN یک نشده است هیچ توانی را مصرف نمی کند . بنابراین توصیه می شود که ADC قبل از اجرای مدهای Power Saving Sleep خاموش شود .

خروجی ADC که 10 بیت می باشد در ریجیسترهای دیتایADCL,ADCH.ADC قرار می گیرد . با توجه به این نکته که خروجی از سمت راست چیده می شود ولی می توان طوری تنظیم کردکه از سمت چپ قرار بگیرند. (با یک کردن بیت ADLAR در ریجیستر ADMUX)

اگر خروجی یک بیت به چپ شیفت پیدا کند و بیشتر از 8 بیت هم دقت لازم نباشد خواندن ADCH تنها کافی است در غیر این صورت ابتدا باید ADCL خوانده شود و سپس ADCH خوانده شود . برای اینکه اطمینان حاصل شود که محتویات رجیسترهای دیتا به همان تبدیل متعلق است ، به عبارت دیگر اگر ADCL خوانده شود و یک تبدیل آنالوگ به دیجیتال قبل از خواندن ADCH کامل شود با تغییر نکردن محتویات رجیستر ، نتیجه خروجی ADC ناقص است و بعضی از بیت ها از دست میرود ولی وقتی که ADCH هم خوانده می شود ، ADC به رجیسترهای ADCL و ADCH دو باره دسترسی پیدا می کند .

ADC وقفه های خاص خودش را دارد و زمانی فعال می شوند که یک تبدیل کاملاً انجام شود . زمانی که ADC به رجیسترهای دیتا دسترسی پیدا می کند، وقفه فعال خواهد شد حتی اگر بیتی از خروجی از دست برود .

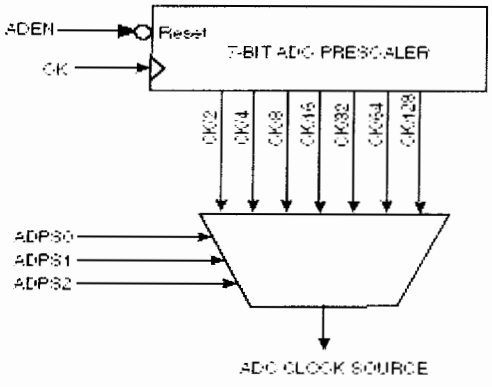
۴-۷-۳- شروع یک تبدیل

با نوشتن منطق یک، در بیت شروع تبدیل ADC (ADSC) عمل ADC انجام می شود . و این بیت در طول عملیات ADC ، high باقی می ماند و زمانی که این عمل ADC تمام شود توسط سخت افزار پاک می شود .

اگر به یک کانال دیتای دیگری ، مقدار آنالوگ داده شود، زمانی که هنوز عملیات ADC قبلی تمام نشده است ، قبل از اجرای تغییرات در کانال عملیات ADC تمام خواهد شد .

در مد Free running ، ADC دائماً در حال نمونه برداری و تغییر دادن مقدار رجیستر دیتا ADC می باشد . مد Free running با high کردن بیت ADFR در رجیستر ADCSRA فعال می شود . اولین تبدیل زمانی انجام می شود که بیت ADSC،در رجیستر ADCSRA یک شود . در این مد، ADC عملیات تبدیل را به طور متوالی انجام می دهد خواه پرچم وقفه ADC ، 1 شده باشد خواه نشده باشد .

۴-۷-۴- Prescaling و زمان بندی تبدیل :



شکل ۴-۴- ADC Prescaler

برای بدست آوردن قدرت تشخیص ماکزیمم، لازم است که فرکانس کلاک ورودی مدار بین 200KHZ تا 50KHZ انتخاب شود .

اگر یک قدرت تشخیص و دقتی کمتر از 10 بیت لازم باشد ، فرکانس کلاک ورودی مـی توانـد بیشتر از 200KHZ باشد تا سرعت نمونه برداری افزایش یابد .

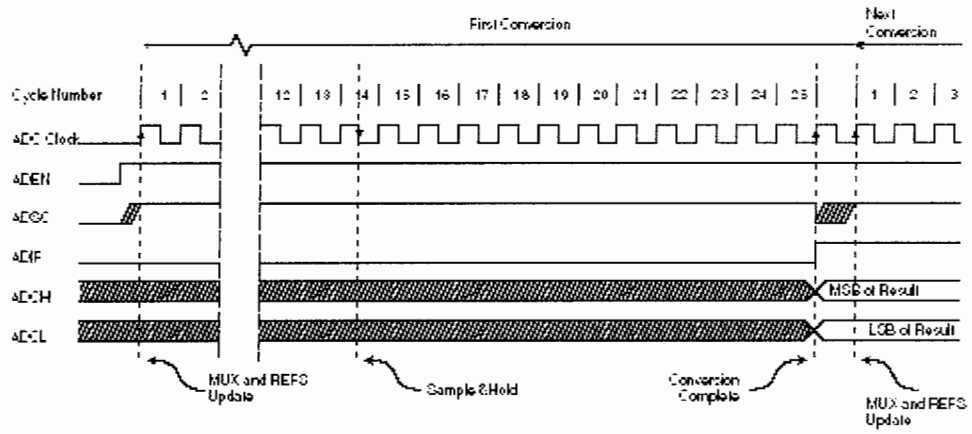
Prescaling با high کردن بیت های ADPS در رجیستر ADCSRA فعال می شود .

Prescaler از لحظه ای که بیت ADEN در رجیستر ADCSRA راه انداخته شود عمـل تبـدیل با لبه بالا رونده سیکل کلاک ADC انجام می شود، و یک تبدیل نرمال 13 سیکل کلاک می گیرد .

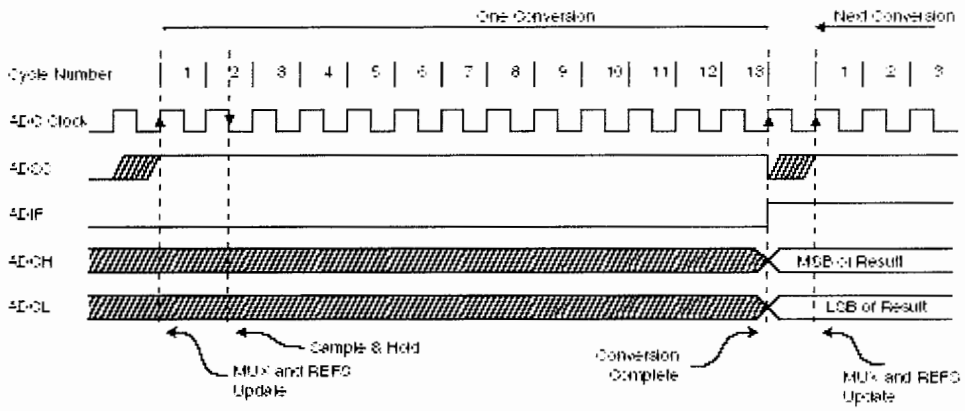
اولین تبدیل بعد از اینکه ADC فعال شود (بیت ADEN در رجیستر ADCSRA ، high شود). 25 سیکل کلاک را به منظور مقدار دهی اولیه به مدار آنالوگ لازم دارد .

نمونه برداری و حفظ دیتا بعد از شروع یک تبدیل نرمال 1.5 سیکل کلاک ADC را می گیرد و 13.5 سیکل کلاک ADC را بعد از شروع اولین تبدیل می خواهد .

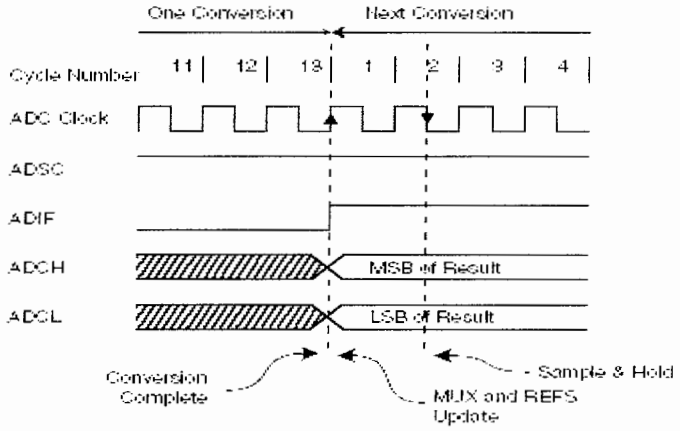
زمانی که یک تبدیل کامل شد ، خروجی در رجیسترهای دیتای ADC نوشته می شود و ADIF ، high می شود . در مد تبدیل ، ADSC همزمان پاک می شود و نرم افزار ممکن اسـت ADC را دوباره high کند و تبدیل دیگری همزمان با اولین لبه بالا رونده کلاک ADC انجـام شـود . در مـد Free running بلافاصله بعد از اینکه یک تبدیل کامل شد تبدیل دیگری انجام مـی شـود البتـه تـا زمانی که ADSC , high باقی بماند . (به جدول ۴-۱ نگاه کنید) .



شکل ۴-۵- دیاگرام زمان بندی ADC ، اولین تبدیل (مد تبدیل Single)



شکل ۴-۶- دیاگرام زمان بندی ADC ، تبدیلsingle



شکل ۴-۷- دیاگرام زمان بندی ADC ، تبدیل در مد Free Running

زمان تبدیل (سیکل ها)	نمونه برداری و حفظ دیتا (سیکل های کلاک از شروع تبدیل)	وضعیت
25	14.5	اولین تبدیل
13	1.5	تبدیل نرمال ، Signal - ended
13/14	1.5/2.5	تبدیل نرمال ، تفاضلی

جدول۴-۱-زمان تبدیلADC

۴-۷-۵- کانالهای ورودی ADC

زمانی که کانالهای مختلفی انتخاب می شود و کانالهای ورودی تغییر می کند کاربر باید مواظب باشد تا اطمینان حاصل کند که کانال را درست انتخاب کرده است . در مد تبدیل Single ، همیشه کانال قبل از شروع تبدیل انتخاب می شود .

یک سیکل کلاک ADC البته بعد از High شدن ADSC ممکن است کانال انتخابی را تغییر دهد . هر چند ساده ترین راه این است که منتظر بماند تا تبدیل قبل از انتخاب کانال دیگری کامل شود . در مد Free Running همیشه کانال قبل از شروع اولین تبدیل انتخاب می شود . یک سیکل کلاک ADC البته بعد از High شدن ADSC ممکن است کانال انتخابی را تغییر دهد . هر چند ساده ترین راه این است که منتظر بماند تا اولین تبدیل انجام شود و سپس کانال را انتخاب کند از آنجایی که تبدیل بعدی به صورت خودکار انجام می شود خروجی نهایی با توجه به کانالی که زودتر انتخاب شده منعکس می شود .

و تبدیل بعدی خروجی را با توجه به کانال انتخابی بعدی نمایش می دهد . زمانی که از کانالهایی دارای گین تفاضلی استفاده می شود خروجی اولین تبدیل ممکن است از دقت و صحت نامطلوبی برخوردار باشد که این ناشی از مدت زمان لازم برای حذف آفست مدار به طور خودکار می باشد .

۴-۷-۶- ولتاژ مرجع ADC

ولتاژ مرجع ADC بیانگر تغییرات تبدیل ADC می باشد . کانالهای Single – Ended که ولتاژ ورودی آنها از ولتاژ V_{REF} تجاوز کند ، خروجی در بازه OXFF قرار می گیرد . V_{REF} را می توان V_{CC} و یا 2.56V و یا مقداری که به پایه خروجی AREF وصل شده انتخاب کرد .

ولتاژ مرجع 2.56V درونی میکرو از V_{BG} (اختلاف ولتاژ مرجع داخلی میکرو) در طی یک تقویت درونی میکرو تأمین می شود .

در یک حالت دیگر پایه AREF را می توان مستقیماً به پایه ADC وصل کرد و با وصل کردن پایه AREF به یک خازن می توان این پایه را از نویز ایمن کرد . توجه کنید که V_{REF} یک منبع high امپدانس است و یک بار خازنی باید وصل شود . اگر کاربر یک منبع ولتاژ ثابت به پایه AREF وصل کند ، کاربر دیگر از ولتاژ اختیاری دیگری به عنوان مرجع استفاده نمی کند چرا که مرجع

های دیگری به پایه خروجی اتصال کوتاه شوند و اگر هیچ ولتاژ به پایه AREF وصل نشود کاربر و یا میکرو یک ولتاژی بین 2.56V و V_{CC} را به عنوان ولتاژ مرجع برمی گزیند البته خروجی اولین تبدیل بعد از انتخاب ولتاژ مرجع ممکن است غلط باشد و کاربر ترجیح دهد که این خروجی را حذف کند . اگر کانالهای تفاضلی انتخاب شوند ولتاژ مرجع انتخابی نباید به V_{CC} محدود شود به جدول ۴-۲ نگاه کنید .

ویژگیهای ADC – دیتای اولیه

Symbol	Parameter	Condition	Min[1]	Typ[1]	Max[1]	Units
	Resolution	Gain = 1x			10	Bits
		Gain = 10x			10	Bits
		Gain = 200x			10	Bits
	Absolute Accuracy	Gain = 1x V_{REF} = 4V, V_{CC} = 5V ADC clock = 50 - 200 kHz		17		LSB
		Gain = 10x V_{REF} = 4V, V_{CC} = 5V ADC clock = 50 - 200 kHz		17		LSB
		Gain = 200x V_{REF} = 4V, V_{CC} = 5V ADC clock = 50 - 200 kHz		7		LSB
	Integral Non-Linearity (INL) (Accuracy after Calibration for Offset and Gain Error)	Gain = 1x V_{REF} = 4V, V_{CC} = 5V ADC clock = 50 - 200 kHz		1.5		LSB
		Gain = 10x V_{REF} = 4V, V_{CC} = 5V ADC clock = 50 - 200 kHz		2		LSB
		Gain = 200x V_{REF} = 4V, V_{CC} = 5V ADC clock = 50 - 200 kHz		3		LSB
	Gain Error	Gain = 1x		1.5		%
		Gain = 10x		1.5		%
		Gain = 200x		0.5		%
	Offset Error	Gain = 1x V_{REF} = 4V, V_{CC} = 5V ADC clock = 50 - 200 kHz		2		LSB
		Gain = 10x V_{REF} = 4V, V_{CC} = 5V ADC clock = 50 - 200 kHz		3		LSB
		Gain = 200x V_{REF} = 4V, V_{CC} = 5V ADC clock = 50 - 200 kHz		4		LSB
	Clock Frequency		50		200	kHz
	Conversion Time		65		260	µs
AVCC	Analog Supply Voltage		V_{CC} - 0.3[2]		V_{CC} + 0.3[2]	V
V_{REF}	Reference Voltage		2.0		AVCC - 0.5	V
V_{IN}	Input Voltage		GND		V_{CC}	V
V_{DIFF}	Input Differential Voltage		-V_{REF}/Gain		V_{REF}/Gain	V
	ADC Conversion Output		-511		511	LSB
	Input Bandwidth			4		kHz

Symbol	Parameter	Condition	Min[1]	Typ[1]	Max[1]	Units
V_{INT}	Internal Voltage Reference		2.3	2.56	2.7	V
R_{REF}	Reference Input Resistance			32		kΩ
R_{AIN}	Analog Input Resistance		55	100		MΩ

جدول۴-۲- ویژگیهای ADC

۴-۷-۷- حذف نویز ADC

از خصوصیات ویژه ADC حذف نویز میکرو می باشد بطوریکه عمل تبدیل حین مد Sleep انجام می گیرد تا نویز ناشی از حافظه مرکزی cpu و پورت های I/O را حذف کند . برای حذف نویز می توان از ADC.Noise Reduction و مد Idle استفاده کرد . برای استفاده از این ویژگیها باید به طرق زیر عمل کرد :

۱. باید مطمئن شد که ADC فعال است و عمل تبدیل انجام نمی گیرد و مد تبدیل Single باید انتخاب شود و همینطور وقفه های تبدیل ADC باید همه فعال باشند .

۲. باید مد ADC.Noise Reduction (یا مد Idle) را فعال کرد.

۳. اگر هیچ وقفه ای از کامل شدن عمل تبدیل ADC اتفاق نیفتد وقفه ADC به cpu پیغام می فرستد و با اجرای تبدیل ADC روال وقفه پایان می پذیرد .

اگر هیچ وقفه دیگری قبل از اینکه ADC کامل شود به cpu پیغام بفرستد این وقفه اجرا خواهد شد و وقفه اجرای کامل ADC زمانی که ADC کامل انجام شود ایجاد خواهد شد و cpu در مد active باقی می ماند تا زمانی که دستور sleep دیگری اجرا شود .

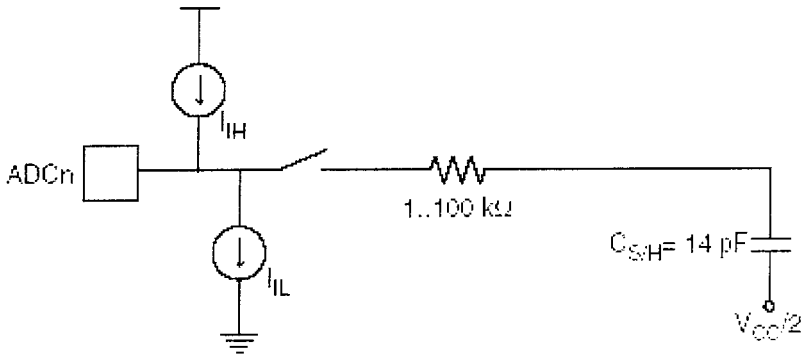
توجه کنید که عمل ADC به طور خودکار خاموش نمی شود زمانی که مدهای sleep دیگری به جز مد Idle و مد ADC.Noise Reduction اجرا می شود .

میکرو ترجیح می دهد تا منطق بیت ADEN را قبل از اجرای این مدهای sleep صفر کند تا از مصرف توان زیاد خودداری کند . اگر ADC در این مدهای sleep فعال باشد و میکرو بخواهد تبدیلی تفاضلی را اجرا کند ترجیح می دهد تا ADC را خاموش کند و بعد از خارج شدن از مد sleep بلافاصله عمل تبدیل را ادامه می دهد تا یک خروجی درست را ایجاد کند .

۴-۷-۸- مدار ورودی آنالوگ

مدار ورودی آنالوگ برای کانال های Single – ended در شکل ۴-۸ نشان داده شده است یک منبع آنالوگ اعمال شده به ADCn را به یک پایه خازنی که با یک مقاومت سری شده وصل کرده هر چند این کانال انتخابی به عنوان ورودی ADC انتخاب شده باشد . اگر کانال انتخاب شود، منبع باید از طریق مقاومت سری شده ،خازن S/H را ولتاژ بدهد . ADC با یک مقاومت خروجی تقریباً $10K\Omega$ و یا کمتر برای سیگنالهای آنالوگ بهینه شده است اگر از چنین منبعی استفاده شود زمان

نمونه برداری ناچیز می شود و اگر یک منبع با امپدانس بالا استفاده شود زمان نمونه برداری به این بستگی دارد که منبع چه مدت زمانی را برای شارژ خازن S/H لازم دارد . میکرو توصیه می کند که از یک منبع با امپدانس پائین استفاده شود و فرکانس سیگنال آنالوگ ورودی پائین باشد یعنی تغیراتش آرام باشد و این باعث می شود تا انتقال شارژ به خازن S/H کمتر نیاز باشد .



شکل ۴-۸- مدار ورودی آنالوگ

۴-۷-۹- تکنیک های حذف نویز

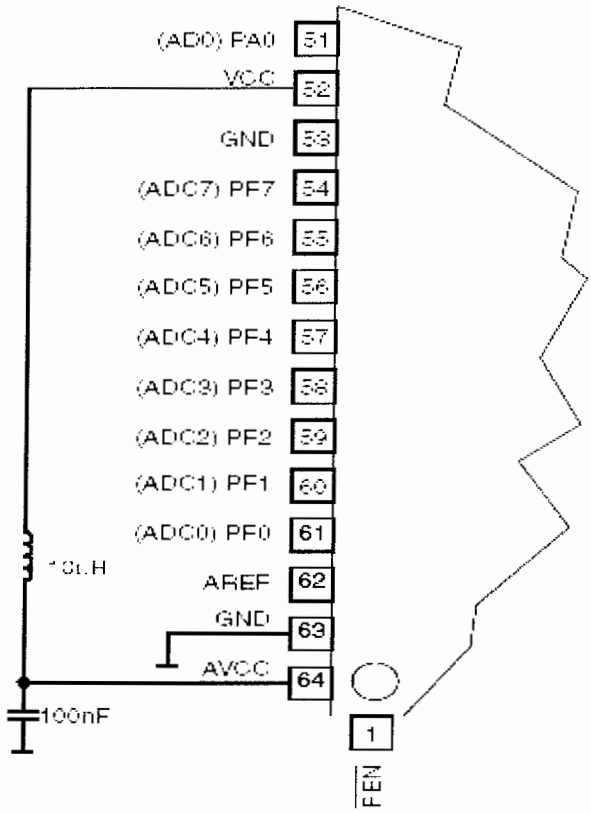
مدارات دیجیتال داخلی و خارجی میکرو ، EMI ای که ممکن است بر روی دقت اندازه گیری آنالوگ تأثیر بگذارد تولید می کند و اگر دقت تبدیل نامطلوب باشد باید از مقدار نویز با استفاده از تفکیک های زیر کم کرد :

۱. مسیرهای آنالوگ را می توان کوتاه کرد . از تماس نداشتن مسیرهای آنالوگ و زمین های آنالوگ اطمینان پیدا کرد و آنها را از مسیرهای دیجیتال فرکانس بالا دور کرد .

۲. باید پایه AV_{CC} میکرو با یک فیلتر پایین گذر به V_{CC} وصل شود که می توان آنرا در شکل ۴-۹ نیز دید .

۳. از مدهای Sleep و حالت ADCNoise Reduction برای کاهش نویز القاء شده توسط CPU استفاده شود .

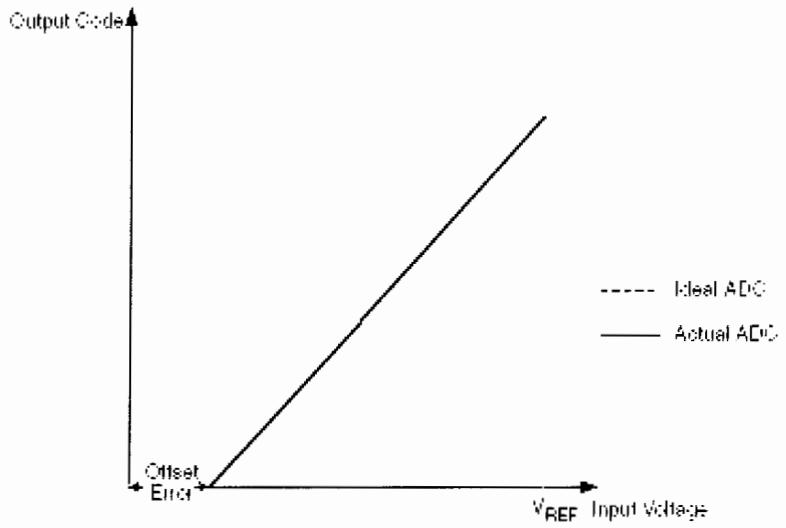
اگر هر کدام از Pin های پورت ADC به عنوان خروجی دیجیتال استفاده می شوند لازم است که این پین ها در حین عمل تبدیل ADC با هم اتصال پیدا نکنند .



شکل ۴-۹- اتصالات توان ADC

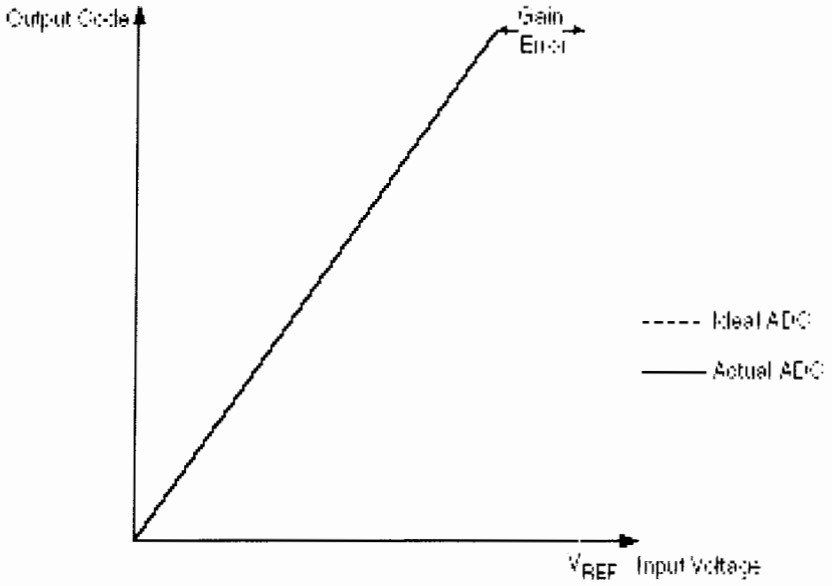
۴-۷-۱۰- تعریف دقت ADC

عمل تبدیل ADC یک سیگنال Single – ended را که بین GND و V_{REF} تغییر می کند. اگر خروجی n بیت باشد سیگنال آنالوگ 2^n حالت (LSBS) دربیاید . کوچکترین کد صفر و بالاترین کد خوانده شده $2^n - 1$ می باشد. پارامترهای مختلف انحراف از یک رفتار ایده آل را شرح می دهد. **آفست :** انحراف از تغییر اول (0x000 به 0x001) مقایسه شود با تغییر (در 0.5LSB) که مقدار ایده آل 0LSB می باشد .



شکل۴-۱۰- خطای آفست

خطای بهره : بعد از تعدیل کردن آفست ، خطای بهره به عنوان انحراف از آخرین تغییر (0X3FE به 0X3FF) پیدا می شود که با تغییر ایده آل (ماکزیمم کمتر از 5LSB.) مقایسه می شود . مقدار ایده آل 0LSB می باشد .



شکل ۴-۱۱- خطای بهره

خطای تبدیل : در طول تبدیل ولتاژ ورودی به یک عدد و یا کد ، یک رنج از ولتاژ ورودی (به پهنای 1LSB) به کد متناظر با همان ولتاژ تبدیل می شود . خطا همیشه $\pm 0.5LSB$ می باشد .

دقت مطلق : ماکزیمم انحراف از یک تغییر واقعی با یک تغییر ایده آل برای هر کد مقایسه می شود این به خاطر ترکیب تأثیرات آفست ، خطای بهره ، خطای تفاضلی غیر خطی ، خطـای تبـدیل می باشد که مقدار ایده آل $\pm 0.5LSB$ می باشد .

۴-۷-۱۱- خروجی تبدیل ADC

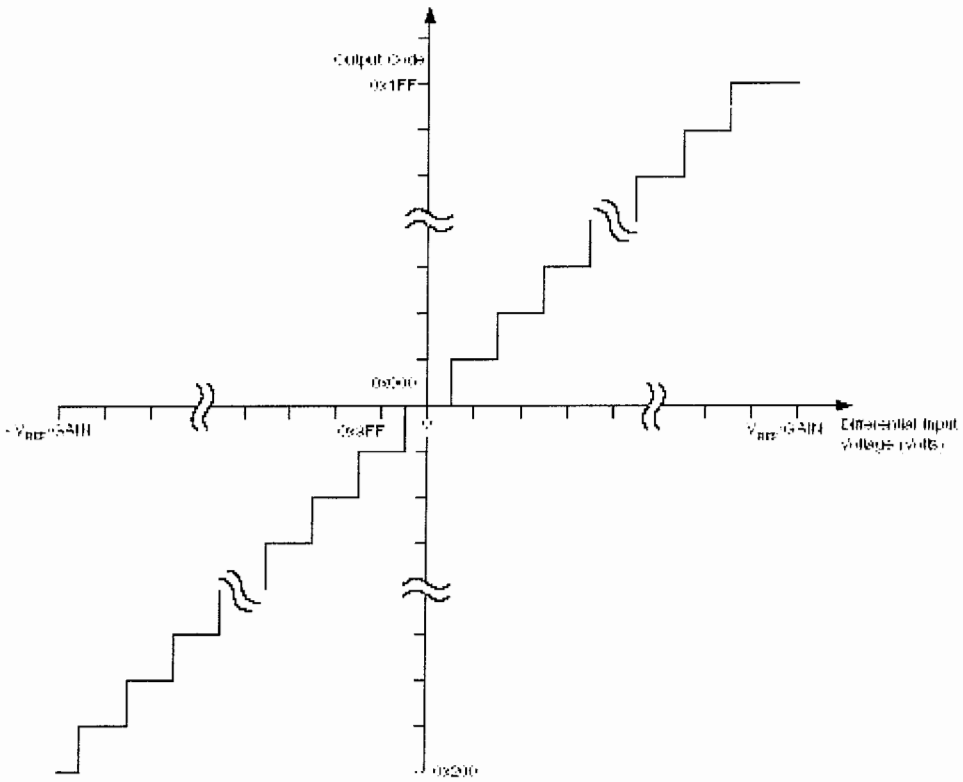
بعد از اینکه تبدیل کامل شد (یعنی ADIF یک شد) خروجی تبدیل در رجیسترهای خروجـی ADC (ADCL , ADCH) دیده می شوند :

$$ADC = \frac{V_{in}.1024}{V_{REF}} \qquad (۱-۴)$$

با توجه به این نکته که V_{in} به عنوان ورودی ولتاژ اعمالی در Pin ورودی انتخاب شده است و V_{REF} ولتاژ مرجع انتخابی است به جدول ۴-۴ نگاه کنید . واگر از کانالهای تفاضلی استفاده شود خروجی بصورت زیر می شود :

$$ADC = \frac{(V_{POS} - V_{NEG}).GAIN.512}{V_{REF}} \qquad (۲-۴)$$

با توجه به این نکته که V_{POS} ولتاژ مثبت در Pin ورودی مثبت و V_{NEG}درPin ورودی منفی GAIN و بهره انتخابی و V_{REF} ولتاژ مرجع می باشد خروجی در دو فرم مکمل از 0X200 (-512d) تا 0x1FF(+511d) نمایش داده می شود . توجه کنید که میکرو برای سریع پی بردن به پلاریته خروجی کافی است که فقط مقدار MSB خروجی را بخواند (ADC9 در ADCH) اگر این بیت یک باشد خروجی منفی و اگر این بیت صفر باشد خروجی مثبت است . شکل ۴-۱۲ کد گذاری رنج ورودی تفاضلی را نشان می دهد . جداول ۴-۳ نیز نتیجه کدهای خروجی را نشان می دهد ، اگر جفت کانال ورودی تفاضلی باشد (ADCM , ADCN) با بهره GAIN و ولتاژ مرجع V_{REF} انتخاب شده است .



شکل ۴-۱۲- رنج تغییرات اندازه گیری شده

V_{ADCn}	Read code	Corresponding decimal value
$V_{ADCm} + V_{REF}/GAIN$	0x1FF	511
$V_{ADCm} + 511/512\ V_{REF}/GAIN$	0x1FF	511
$V_{ADCm} + 511/512\ V_{REF}/GAIN$	0x1FE	510
...	...	...
$V_{ADCm} + 1/512\ V_{REF}/GAIN$	0x001	1
V_{ADCm}	0x000	0
$V_{ADCm} - 1/512\ V_{REF}/GAIN$	0x3FF	-1
...	...	...
$V_{ADCm} - 511/512\ V_{REF}/GAIN$	0x201	-511
$V_{ADCm} - V_{REF}/GAIN$	0x200	-512

جدول ۴-۳-ارتباط بین ولتاژ ورودی و کدهای خروجی

مثال ، ADMUX=0XED(ADC3-ADC2.10Xgain ,2.56 refrence و خروجی از سمت چپ مرتب می شود)

ولتاژ در ADC3 , 300mv است و در ADC2 ، 500 mv است .

$$ADCR = 512 \times 10 \times (300 - 500)/2500 = -400 = 0X270$$

نهایتاً ADCL ، 0X00 و ADCH ، 0X9C خواهد بود .

با صفر کردن بیت ADLAR خروجی از سمت راست چیده می شود ،

ADCH = 0X02 و ADCL=0X70

انتخاب مالتی پلکسر ADC ، رجیستر ADMUX

Bit	7	6	5	4	3	2	1	0	
	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

بیت 6;7 – REFSI : بیت های انتخاب مرجع

این بیت ها ولتاژ مرجع را برای ADC همانطور که در جدول ۴-۴ نشان داده شده است انتخاب می کنند . اگر این بیت ها در طول یک تبدیل تغییر کنند ، تغییرات تأثیری نخواهند داشت تا زمانی که تبدیل کامل شود (ADIF در ADCRA . high است) اگر چه به پایه AREF میکرو ولتاژی را اعمال کنیم ولتاژ refrence درونی مدار استفاده نخواهد شد .

REFS1	REFS0	Voltage Reference Selection
0	0	AREF, Internal Vref turned off
0	1	AVCC with external capacitor at AREF pin
1	0	Reserved
1	1	Internal 2.56V Voltage Reference with external capacitor at AREF pin

جدول ۴-۴- انتخاب ولتاژ مرجع برای ADC

بیت **5** ام – **ADLAR (ADCLeft Adjust Result)**

بیت ADLAR بر روی نمایش خروجی تبدیل ADC، در رجیستر دیتای ADC تأثیر می گذارد. اگر ADLAR یک باشد خروجی از طرف چپ مرتب می شود در غیر این صورت خروجی از طرف راست مرتب می شود .تغییر در بیت ADLAR بر روی ADC Data Register فوراً تأثیر می گذارد با وجود اینکه عمل تبدیل به طور مداوم انجام می شود .

بیت **4** ام : **:0:0-mux4**

کانال آنالوگ و بیت های انتخاب بهره ، مقدار این بیت ها انتخاب می کنند که کدام یک از ترکیب ورودیهای آنالوگ به ADC متصل هستند . این بیت ها همچنین بهره کانال های تفاضلی را انتخاب می کنند . این بیت ها در طول یک تبدیل تغییر می کند و این تغییرات تا زمانی که تبدیل کامل نشده است تأثیر نمی گذارد. (ADIF در ADCSRA یک است)

کنترل ADC و موقعیت رجیستر ADSRA – A

Bit	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

بیت 7 ام – ADEN : فعال کننده ADC :

با یک کردن این بیت ADC فعال می شود و با صفر کردن این بیت ، ADC غیر فعال می شود با خاموش کردن ADC در حین تبدیل عمل ADC پایان می پذیرد .

بیت 6 ام – ADSC :

ADC عمل تبدیل را شروع می کند .

در مد تبدیل Single با یک کردن این بیت عمل تبدیل شروع میشود .

در مد Free Running ، با یک شدن این بیت اولین عمل تبدیل آغاز می شود . در اولین تبدیل بعد از اینکه ADSC ، یک شد ، ADC فعال می شود یا اگر ADSC در همان زمانی که ADC فعال شده است ، مقدار دهی شود 25 سیکل کلاک ADC را می گیرد در حالی که مقدار نرمال سیکل کلاکی که لازم است 13 تا می باشد. اولین تبدیل دستورات ADC را اجرا می کند و در حالیکه صفر شدن این بیت تأثیری نمی گذارد .

بیت **5 ام – ADFR** : انتخاب مد Free Running ، ADC زمانی که این بیت یک می شود ، ADC در مد Free Running کار می کند ADC نمونه برداری می کند . با صفر کردن این بیت ADC از مد Free Running بیرون می آید .

بیت 4 ام – ADIF : پرچم وقفه ADC :

این بیت زمانی که یک تبدیل کامل می شود و رجیسترهای دیتا مقدار جدیدی می گیرند ، یک می شود . اگر بیت ADIE , I-bit در SREG یک شود وقفه کامل شدن تبدیل ADC عمل می

کند . ADIF توسط سخت افزار زمانی وقفه های مشابه را اجـرا مـی کنـد . ADIF نیـز بـه همـین ترتیب با یک کردن پرچم پاک می شود .

توجه داشته باشید که اگر عمل خواندن ، نوشتن یا ویرایش ADCSRA را انجام مـی دهیـد در طول این مدت وقفه می تواند غیرفعال باشد . همچنـین اگـر از دسـتور العمـل هـای SBI و CDI استفاده شود همین اتفاق می افتد .

بیت 3 ام – ADIE : فعال کننده وقفه ADC

زمانی که این بیت یک شده است ، و I-bit در رجیستر SREG نیز یک شده است وقفه کامل شدن تبدیل ADC فعال می شود .

بیت های 0:ADPS2-0:2 : بیت های انتخاب Prescaler . ADC این بیت ها ، ضریب فرکانس بین XTAL و کلاک ADC ورودی را تعیین می کند .

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

جدول۴-۵- انتخابهایADC Prescaler

رجیسترهای دیتای **ADCH , ADCL – ADC**

ADLAR = 0:

Bit	15	14	13	12	11	10	9	8	
	–	–	–	–	–	–	ADC9	ADC8	ADCH
	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

ADLAR = 1:

Bit	15	14	13	12	11	10	9	8	
	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
	ADC1	ADC0	–	–	–	–	–	–	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

زمانی که یک تبدیل ADC کامل می شود ، خروجی را در این دو رجیستر می توان دید ، اگر از کانالهای تفاضلی استفاده شود خروجی در دو شکل مکمل هم نشان داده می شود .

زمانی که ADCL خوانده می شود رجیستر دیتای ADC تا زمانی که ADCH خوانده شود مقدار جدیدی نمی گیرد . نتیجتاً اگر خروجی از طرف چپ مرتب شود،بیشتر از 8 بیت لازم ندارد ، و خواندن ADCH کافی است در غیر این صورت ابتدا ADCL باید خوانده شود و سپس ADCH ، بیت ADLAR در رجیستر ADMUX و بیت های MUXN در رجیستر ADMUX بر مسیری که از رجیسترها بخواند تأثیر می گذارد .

اگر ADLAR یک شود خروجی از چپ مرتب می شود و اگر ADLAR پاک شود خروجی از طرف راست مرتب می شود .

ADC9 : 0 : خروجی تبدیل ADC :

این بیت ها خروجی تبدیل را نشان می دهد .

DAC –۸–۴

برای بررسی PWM ابتدا باید Counter / Timer ها را مرورکرد.

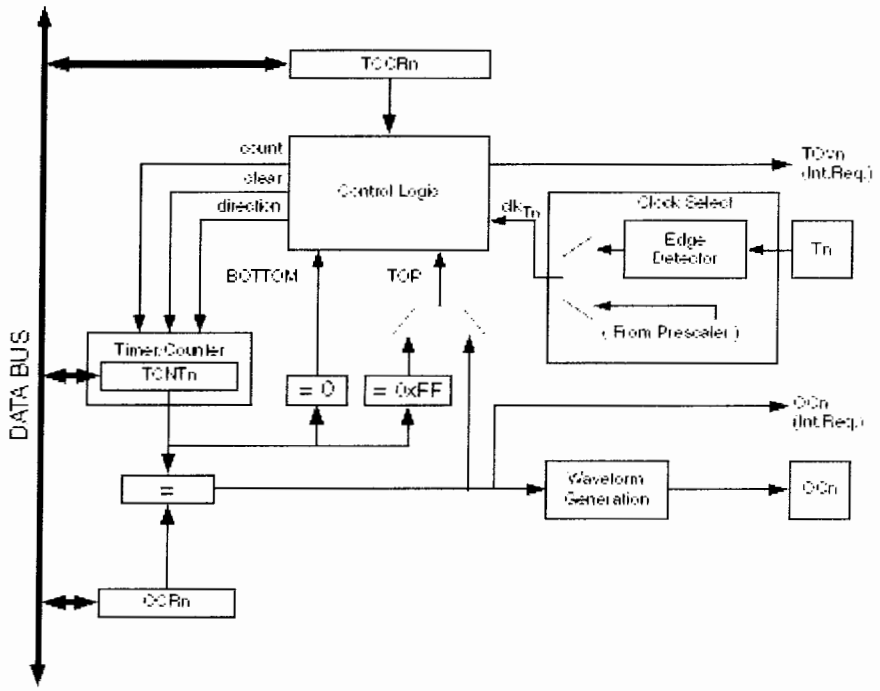
مهمترین ویژگی های تایمر و کانتر و PWM به صورت زیر می باشد :

- **کانتر دارای کانال Single**
- **تایمر بعد از انجام مقایسه و بعد از مطابقت پاک می شود (به صورت خودکار مقدار می گیرد) .**
- **آشکار سازی پهنای پالس دقیق خازنی (PWM)**
- **تولید کننده فرکانس**

- **10 بیت کلاک Prescaler**
- **سرریز و مقایسه تطابق وقفه (TOV2 , OCF2)**

مرور جزئی :

یک بلوک دیاگرام ساده شده از Timer / Counter هشت بیتی در شکل ۴-۱۳ نشان داده شده است .



شکل ۴-۱۳- بلوک دیاگرام Timer / Counter . 8 بیتی

۴-۸-۱- رجیسترها

Timer / Counter ، (TCNT2) و رجیستر مقایسه گر و خروجی (OCR2) ، رجیستر 8 بیتی هستند . سیگنالهای درخواست وقفه (در شکل با int Reg نمایش داده شده اند) .

در رجیستر پرچم وقفه Timer قابل رؤیت می باشد . (TIFR) همه وقفه ها با رجیستر وقفه MASK پوشش داده می شوند. Timer / Counter از طریق Prescaler یا به وسیله یک منبع کلاک خارجی در پایه T2 یا از طریق کلاک داخلی می تواند کلاک بخورد .

می توان تعیین کرد با هر لبه کلاک Timer / Counter یا به مقدار واقعی اش یکی اضافه کند و یا یکی کم کند . Timer / Counter زمانی که هیچ کلاکی نخورد غیر فعال است .

OCR2 در تمام مدت عملیات Timer / Counter با مقدار اندازه گیری توسط Timer / Counter مقایسه می شود . خروجی مقایسه شده رامی توان به عنوان امواج ژنراتور برای تولید یک PWM یا فرکانس خروجی متغیر در پایه OC2 به کار برد . در صورت تطابق یا match شدن خروجی بعد از انجام مقایسه ، پرچم مقایسه یا OCF2 یک خواهد شد به طوری که برای تولید یک درخواست وقفه مقایسه گر خروجی می توان استفاده کرد .

۴-۸-۲- تعریف ها :

خیلی از رجیسترها و بیت های مرجع در این پوشه در یک شکل عمومی نوشته می شوند وقتی که از یک بیت یا رجیستر در یک برنامه استفاده می شود شکل مختصر آن بیت یا رجیستر را باید استفاده کرد مثل i.e , TCNT2 به جای استفاده از Timer / Counter)

به تعاریف جدول زیر دقت کنید .

BOTTOM	کانتر زمانی به مقدار BOTTOM می رسد که مقدار آن 0x00 شود
MAX	کانتر زمانی به مقدار MAX خود می رسد که مقدار آن 0XFF شود
TOP	کانتر زمانی به مقدار TOP می رسد که مقدار آن با ارزش ترین مقدار شمارش برابر شود . ارزش TOP را می شود با یک مقدار ثابت PXFF (مقدار ماکزیمم یا مقدار ذخیره شده در رجیستری OCR2 تعیین گردد . مقدار دهی TOP به مد عملکرد بستگی دارد .

جدول۴-۶- تعاریف .

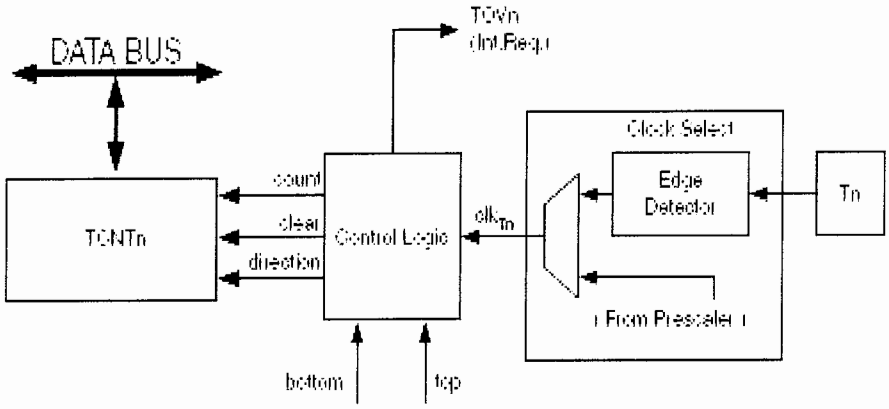
۴-۸-۳- منابع کلاک . Timer / Counter

Timer / Counter می تواند از کلاک داخلی مدار و کلاک خارجی استفاده کند .

منابع کلاک Timer / Counter بوسیله منطق انتخاب کلاک تعیین می گردد که از طریق انتخاب کلاک (CS22:0) که در رجیستر کنترلی Timer / Counter (TCCR2) قرار دارد .

۴-۸-۴- واحد شمارش

مهمترین قسمت ، 8 بیت Timer / Counter واحد شمارش دو طرفه قابل برنامه ریزی می باشد شکل ۴-۱۴ یک بلوک دیاگرام از کانتر و قسمت های اطراف کانتر را نشان می دهد .



شکل ۴-۱۴- بلوک دیاگرام کانتر

۴-۸-۵- توضیح در مورد سیگنالها :

Count : TCNT2 ، یا یکی یکی کم می کند و یا زیاد می کند .

Direcetion : بین افزایش و یا کاهش یکی را انتخاب می کند .

Clear : TCNT2 را پاک می کند (همه بیت ها را یک می کند)

Clx_{TN} : کلاک Timer / Counter اشاره می کند به Clx_{TO}

TOP : نشان می دهد که TCNT2 به مقدار ماکزیمم خود رسیده است .

Bottom : نشان می دهد که TCNT2 به مقدار min خود رسیده است .

با توجه به مد عملکرد کانتر ، کانتر در هر کلاک تایمر (Clx_{T2}) پاک می شود یا مقدارش اضافه می شود و یا کم می شود . و زمانی که هیچ کلاکی به میکرو اعمال نشود (یعنی cs22:0=0 باشد) تایمر متوقف می شود . هر چند که مقدار TCNT2 از طریق CPU قابل دسترسی باشد، خواه Clx_{T2} وجود داشته باشد و خواه وجودنداشته باشد .

ترتیب شمارش از طریق یک کردن بیت های WGM01 , WGMOO که در رجیسترهای کنترلی Timer / Counter (TCCR2) قرار داد تعیین می شود .

اتصالات نزدیک به همی وجود دارد که چطور کانتر رفتار کند . (بشمارد) (چطور شکل موج ها در پایه خروجی OC2 تولید شوند) .

پرچم سرریز Timer / Counter ، TOV2 ، مطابق با مد عملکرد

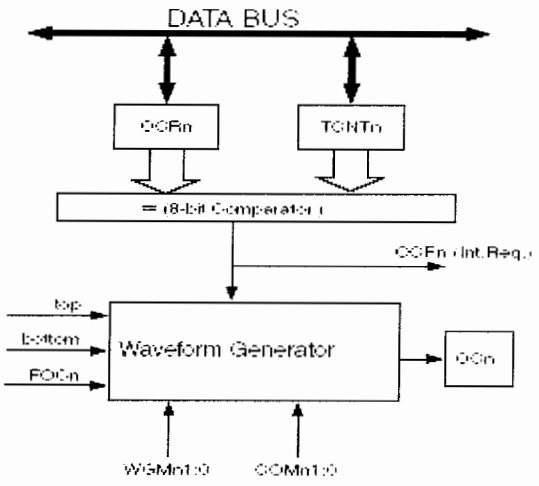
Timer / Counter یک می شود . TOV2 می تواند برای تولید یک وقفه CPU نیز استفاده شود

۴-۸-۶- واحد مقایسه خروجی

هشت بیت مقایسه گر بدون توقف و بطور متوالی TCNT2 رابا رجیستر مقایسه کننده خروجی (OCR2) مقایسه می کند . زمانی که TCNT2 با OCR2 برابر شوند مقایسه گر اعلام تطابق می کند . تطابق در سیکل بعدی کلاک تایمر، پرچم مقایسه گر خروجی (OCF2) را یک خواهد کرد . اگر پرچم مقایسه کننده خروجی فعال باشد (یعنی OCIE2=1 باشد و پرچم وقفه سراسری در SERG یک باشد) ، باعث می شود تا یک وقفه مقایسه گر خروجی ایجاد شود . پرچم OCF2 ، زمانی که وقفه اجرا می شود بطور خودکار مقدار این پرچم پاک می شود . به همین ترتیب ، پرچم OCF2 از طریق نرم افزار با یک کردن موقعیت بیت I/O پاک می شود .

ژنراتور شکل موج از سیگنال match برای ایجاد خروجی (البته با توجه به مد عملکرد میکرو) استفاده می کند . سیگنالهای max , bottom به عنوان شکل موجهای ژنراتور برای رسیدگی به موارد خاص مقادیر زیاد در بعضی از مدهای عملکرد به کار می رود .

شکل ۴-۱۵بلوک دیاگرام قسمت مقایسه گر خروجی را نشان می دهد .



شکل ۴-۱۵ بلوک دیاگرام قسمت مقایسه گر خروجی

رجیستر OCR2 بافر دوطرفه است . زمانی که از هر یک از مدهای PWM ، timer در حالت عادی و هنگام پاک شدن در مدهای مقایسه (CTC) بافر دوحالته غیرفعال است .

بافر دو حالته مقدار جدید رجیستر مقایسه کننده (OCR2) را با هـر یـک از مقـادیر Bottom , Top از توالی شمارش تطبیق می دهد . عمل همزمان سازی از ایجاد تعداد بیت های فرد پالـسهای نامتقارن PWM جلوگیری می کند . ممکن است که دسترسی به رجیستر OCR2 پیچیده به نظـر برسد ولی اینطور نیست و زمانی که بافر دو طرفه است ، CPU به بافر رجیـستر OCR2 دسترسـی دارد . و اگر بافر دو حالته فعال نباشد ، CPU به OCR2 به سرعت دسترسی پیدا خواهد کرد .

۴-۸-۷- شدت عمل مقایسه خروجی

در مدهای غیر از مد تولید شکل موج PWM ، خروجی match شده مقایسه گر ، با یک کـردن بیت FOC2 (شدت مقایسه خروجی) شدت داده می شود . شدت دادن تطابق مقایسه گـر ، پـرچم OCF2 را یک نخواهد کرد و تایمر را پاک نمی کند و همچنین دوباره بارگذاری نمی کند . ولی پایه OC2 مقدارش تغییر خواهد کرد اگر که خروجی مقایسه گـر match شـود (بیـت هـای یـک شـده COM 21:O تعیین می کند حال خواه پایه OC2 یک شده باشد و یا پاک شـده باشـد و یـا مقـدار قبلی اش تغییر کرده باشد) .

۴-۸-۸- بلوکه کردن تطابق match شده از طریق نوشتن در TCNT2

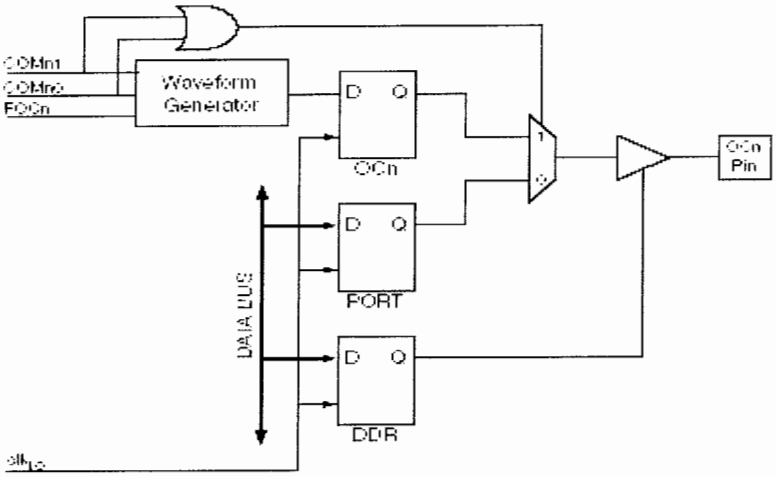
همه CPU ها ، عملکردهارا که در رجیستر TENT2 نوشته اند ، بلوکه خواهند کرد . هر تطـابق match شده را که در سیکل کلاک تایمر بعدی اتفاق می افتد حتی اگر تـایمر متوقـف شـود . ایـن خصوصیت باعث می شود تا OCR2 همان مقدار TCNT2 را بگیرد و بدون اتفاق افتادن یک وقفه ، البته زمانی که کلاک Timer / Counter فعال است .

۴-۸-۹- واحد Compare match out put

در مد Compare match out put بیت های COM 21:O دارای دو تابع هستند . تولید کننـده شکل موج از بیت های COM 21:O برای تعریف(out put Compare) که در Compare match بعدی استفاده می کند .

همچنین بیت های COM 21:O منبع خروجی پایه OC2 را کنترل می کند . شکل ۴-۱۶ یک شما تیک ساده از منطـق تـأثیر پذیرفتـه بـا یـک شـدن بیـت COM 21:O را نـشان مـی دهـد . رجیسترهای I/O ، بیت های I/O و پایه های I/O در شکل به وضوح پیداسـت . تنهـا رجیـسترهای کنترل پورت I/O (DDR) که بوسیله بیت های COM 21:O تأثیر می گیرند در شکل نـشان داده

شده است . وقتی که به موقعیت OC2 مراجعه می شود ، مرجع درونی ، OC2 است نه پایه OC2 و اگر مدار Resetشود رجیستر OC2 ، صفر می شود .



شکل ۴-۱۶- شماتیک واحد Compare match out put

تابع عمومی پورت I/O بوسیله OC2 از طریق ژنراتور شکل موج ، با یک شدن هر کدام از بیت های COM 21:O باطل می شود هر چند مسیر پایه OC2 (ورودی یا خروجی) از طریق رجیستر Data Direction (DDR) کنترل شود . بیت رجیستر Data Direction برای پایه (DDR – OC2) OC2 به عنوان خروجی باید یک شود البته قبل از این مقدار OC2 در روی پایه قرار می گیرد .

پورت ، تابعی که به مد تولید شکل موج وابسته است را باطل می کند . طراحی منطق پین مقایسه گر خروجی ، OC2 را مقدار دهی می کند البته قبل از اینکه خروجی فعال شود .

توجه کنید که بعضی از بیت های یک شده COM 21:O برای مدهای عملکرد خاصی ذخیره می شود .

۴-۸-۱۰- مد Compare out put و تولید شکل موج

در حالت عادی تولید کننده شکل موج به طور متفاوت از بیت های COM 21:O مدهای PWM وCTC استفاده می کند . در همه مدها اگر COM 21:O = O شود این نشان می دهد که تولید کننده شکل موج که هیچ عملی در رجیستر OC2 در Compare match بعدی اجرا نکرده است. برای عملکردهای خروجی مقایسه در هرکدام از مدهای PWM به جدول ۴-۸ مراجعه شود . برای مد سریع PWM به جدول ۴-۹ مراجعه شود و برای فاز صحیح PWM به جدول ۴-۱۰ مراجعه

شود . یک تغییر در موقعیت بیت های COM 21:O در اولین Compare match بعد از اینکه بیت ها ثبت شده اند اتفاق می افتد . برای هرکدام از مدهای PWM ، عملکرد برای یک تأثیر فوری می تواند شدت پیدا کند (با استفاده از بیت های FOC2)

۴-۸-۱۱- مدهای عملکرد

مد عملکرد ، i-e رفتار Timer / Counter و پایه های خروجی مقایسه گر با ترکیب مد تولید شکل موج (wGm2:0) و مد Compare out put (بیت های COM 21:O) تعریف می شوند. بیت های مد Compare out put بر روی توالی شمارش تأثیری نمی گذارد ، تا زمانی که بیت های مد تولید شکل موج ، عمل کنند بیت های COM 21:O کنترل می شوند ، حال خواه خروجی PWM تولید شده معکوس شده باشد و یا نشده باشد . برای مدهایی به غیر از PWM ، بیت های COM 21:O کنترل می کنند . حال خواه خروجی یک شده باشد و یا پاک شده باشد و یا در Compare match تغییر وضعیت بدهد .

۴-۸-۱۲- مد نرمال

ساده ترین عملکرد ، مد نرمال (WGM21:0=0) می باشد . در این مد همیشه شمارش به صورت افزایش می باشد و کانتر پاک نشده است و زمانی کانتر به 8 بیت ماکزیمم رسید (TOP = OXFF) کانتر به سادگی سرریز می شود و دوباره از مقدار OXOO شروع می شود در یک عملکرد عادی پرچم سرریز Timer / Counter (TOV2) در همان سیکل کلاک تایمر یک خواهد شد . همانطور که TCNT2 صفر می شود .

در این حالت پرچم TOV2 به عنوان بیت نهم رفتار می کند و با این تفاوت که این بیت همواره یک است و پاک نمی شود . هر چند از یک وقفه سرریز تایمر ترکیب شده است که پرچم TOV2 را پاک می کند . قدرت تشخیص Timer از طریق نرم افزار قابل افزایش است .

هیچ مورد خاصی برای شامل شدن در مد نرمال وجود ندارد و یک مقدار جدید را می توان هر زمانی در کانتر نوشت . هر وقت که مقدار کانتر به مقدار TOP برسد با استفاده از پرچم OCF2 یک وقفه ایجاد می شود، اگر وقفه فعال شود روال خروجی وقفه برای اینکه TOP مقدار جدیدی بگیرد استفاده می شود .هر چند تغییرات TOP به محدوده Bottom بستگی دارد ، زمانی که کانتر با یک Prescaler پائین و یا اصلاً بدون Prescaler فعال می شود باید توجه شود که مد CTC خصوصیت

بافر دو طرفه را ندارد . اگر مقدار جدید ثبت شده در OCR2 کمتر از مقدار جاری در TCNT2 باشد کانتر تطابق مقایسه را از دست خواهد داد . سپس کانتر تا مقدار ماکزیمم خود خواهد شمرد که مقدار ماکزیمم آن OXFF است و مقدار ایده آل آن 0x00 می باشد . قبل از اینکه Compare match اتفاق بیافتد . برای تولید یک شکل موج خروجی در مد CTC ، خروجی OC2 ، تغییر وضعیت می دهد ، در هر Compare ، با تغییر وضعیت تمام بیت های مد Compare match (COM21:0=1) مقدار OC2 در خروجی ظاهر خواهد شد مگر اینکه مسیر دیتا برای پایه مورد نظر به عنوان خروجی تعریف شود . شکل موج تولید شده زمانی که OCR2 مقدارش صفر شود یک فرکانس man دارد به طوریکه $f_{oc2} = f_{clk}$

فرکانس شکل موج خروجی از طریق تساوی زیر تعریف می شود .

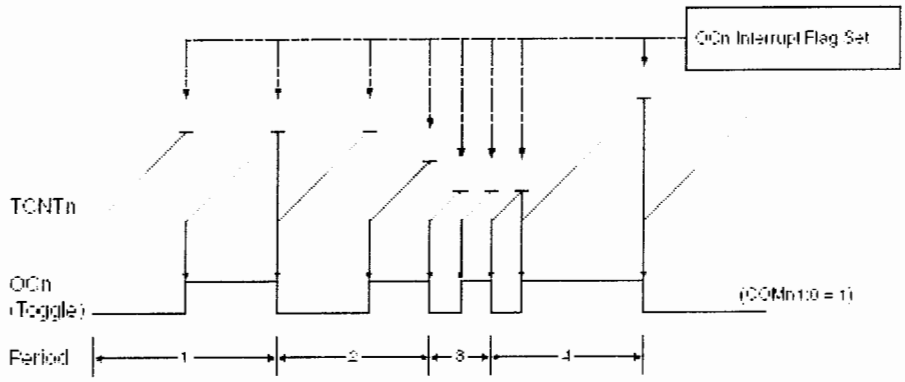
$$f_{ocn} = \frac{f_{clk} - I/O}{2.N.(1 + 0CR_n)} \qquad (۳-۴)$$

واحد out put Compare برای تولید وقفه در بعضی از زمان های داده شده استفاده می شود . استفاده از مقایسه گر خروجی ، برای تولید شکل موج در مد نرمال توصیه نمی شود زیرا این کار فضای زیادی از زمان CPU را اشغال می کند .

۴-۸-۱۳- مد پاک کردن Timer در تطابق مقایسه گر (CTC)

در مد CTC (WGM21:0=2) رجیستر OCR2 استفاده می شود تا قدرت تشخیص کانتر با دست قابل تغییر باشد . در مد CTC ، زمانی که مقدار کانتر TCNT2 با مقدار OCR2 یکی می شود کانتر پاک می شود و با صفر برابر می شود . OCR2 تعیین می کند مقدار TOP برای کانتر را و همچنین قدرت تشخیص آن را تعیین می کند این مد باعث کنترل بیشتری از فرکانس خروجی Compare match می شود و همچنین عملکرد شمارش را ساده می کند .

دیاگرام زمان بندی مد CTC در شکل ۴-۱۷ نشان داده شده است ، مقدار کانتر (TCNT2) افزایش می یابد تا زمانی که TCNT2 . OCR2 . match شوند و سپس مقدار کانتر (TCNT2) پاک می شود.



شکل۴-۱۷-مدCTC،دیاگرام زمان بندی

متغیر N نمایانگر PRESCALER است (1 ، 8 ، 64 ، 256 ، 1024)

در مد نرمال عملکرد پرچم TOV2 در همان سیکل کلاک تایمر یک است بطوری که کانتر از max ام تا صفر را می شمرد .

۴-۸-۱۴- مد سریع p w m

مدولاسیون پهنای پالس سریع مد PWM (بطوریکه (WGM21:0=3 است)

یک شکل موج PWM با فرکانس بالا ایجاد می کند .

PWM سریع از دیگر انتخابات PWM فرق می کند عملکرد شیب دار single کانتر در این مد از مقدار Bottom شروع می کند .

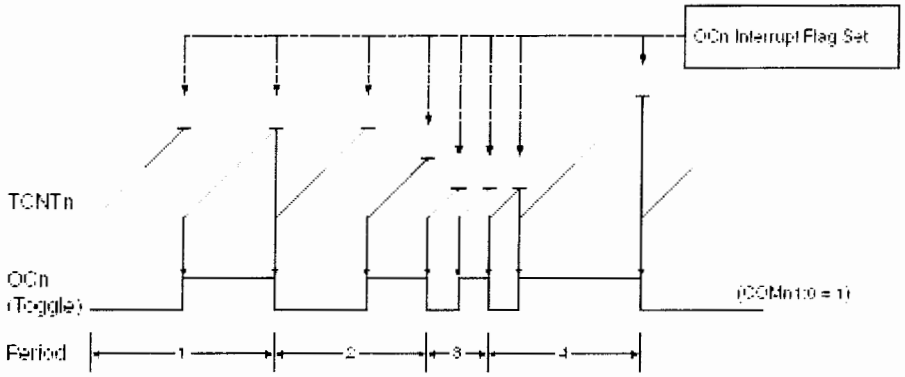
در مد non – inverting ، Compare out put ، OC2 در تطابق مقایسه گر بین OCR2 , TCNT2 پاک می شود و در مقدار Bottom دوباره یک می شود .

در مد inverting تطابق مقایسه ای خروجی در زمان تطابق یک می شود و در مقدار Bottom پاک می شود . ناشی از عملکرد شیبدار single ، فرکانس عملکرد مد سریع PWM دو برابر فرکانس مد PWM با فاز درست است که عملکرد آن بصورت شیب دوطرفه می باشد و این فرکانس بالا باعث می شود که مد سریع PWM برای تنظیم توان ، یکسوسازی ، و عملکرد DAC مناسب باشد .

کار با فرکانس بالا باعث کوچکتر شدن عناصر خارجی مدار می شود (مثل خازن ها و سلف ها) در نتیجه سیستم هزینه کمتری می برد . در مد سریع PWM ، کانتر تا زمانی که مقدارش با مقدار Match , MAX شود می شمارد . سپس کانتر در سیکل کلاک تایمر پاک می شود .

دیاگرام زمان بندی برای مد PWM سریع در شکل ۴-۱۷ نشان داده شده است . مقدار TCNT2 در دیاگرام زمان بندی به شکل یک نمودار ستونی نشان داده شده است . برای شروع عملکرد شیبدار Single .

دیاگرام شامل خروجی های inverting , Non – inverting ، PWM می باشد . خط افقی کوچکی که در روی TCNT2 قرار دارد ، تطابق قیاسی بین OCR2 , TCNT2 را نشان می دهد .



شکل۴-۱۸مدسریعPWM،دیاگرام زمان بندی

پرچم سرریز Timer / Counter (TOV2) ، هر زمانی که کانتر به مقدار max ام رسید اگر وقفه فعال شود . روال وقفه زمانی مورد استفاده قرار می گیرد که ارزش مقایسه مقدارش تغییر کند .

در مد سریع PWM ، واحد مقایسه یک شکل موج PWM در پایه OC2 ایجاد می کند که با قرار دادن بیت ها COM 21:0=2 یک PWM به صورت Non – inverting ایجاد می شود و با قرار دادن بیت های COM 21:0=3 شکل موج خروجی PWM به صورت inverting خواهد بود (به جدول ۴-۹ نگاه کنید) . تنها زمانی مقدار واقعی OC2 در خروجی ظاهر می شود که مسیر دیتای پایه میکرو به عنوان خروجی تعریف شده باشد . شکل موج PWM ، با صفر یا یک کردن رجیستر OC2 در مقایسه بین OCR2 و TCNT2 ایجاد می شود . فرکانس PWM خروجی از طریق رابطه زیر قابل محاسبه می باشد :

$$Focpwm = \frac{f_{clk} - J/o}{N.256} \qquad (۴-۴)$$

که متغیر N نمایانگر Prescaler می باشد (1 ، 8 ، 64 ، 256 ، 1024) مقادیر بزرگ رجیستر OCR2 نمایانگر موارد خاص می باشد زمانی که یک شکل موج PWM در خروجی در مد سریع ایجاد شود . اگر بیت های OCR2 برابر با Bottom شوند .

اگر OCR2 برابر با مقدار max شود خروجی بطور پیوسته high یا low می شوند . (وابسته به پلاریته خروجی که از طریق بیت های COM 21:0 یک می شود) .

فرکانس شکل موج خروجی (با dutycycl , 50%) در مد سریع PWM از طریق یک کردن OC2 برای تغییر وضعیت مقدارش در هر Compare match(COM 21:0=1) قابل دستیابی است .

شکل موج تولید شده ، زمانی که OCR2=0 است یک فرکانس ماکزیمم $f_{oc2} = \frac{f_{c|x-j|o}}{2}$ را خواهد داشت . این ویژگی شبیه تغییر وضعیت OC2 در مد CTC می باشد . بجز در وضعیت بافر دو طرفه از واحد تطابق خروجی که در مد سریع PWM ، فعال است .

۴-۸-۱۵- مد PWM Phase Correct

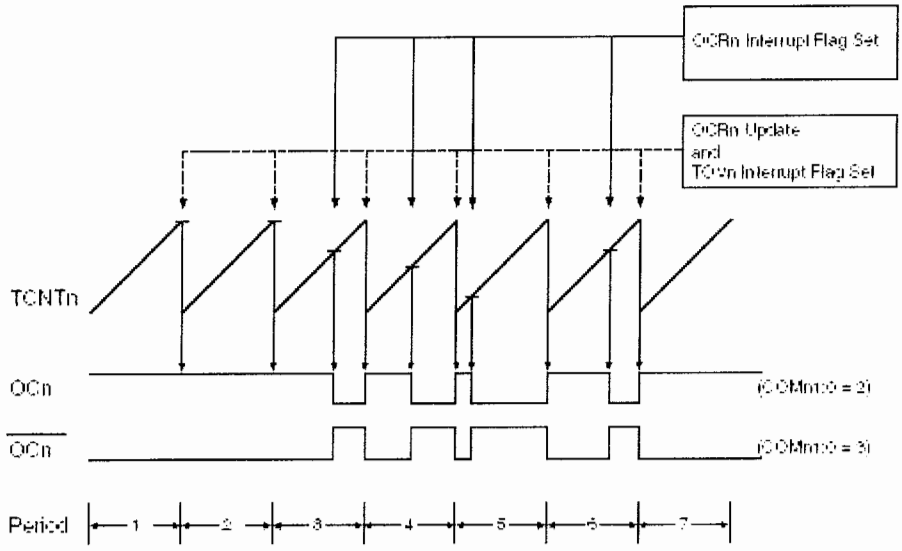
مد PWM Phase Correct (WGM21:0=3) از یک تجربه بالائی از تولید شکل موج PWM برخوردار است . مد PWM Phase Correct بر عملکرد شیب دو طرفه بنا شده است . شمارنده بطور مداوم از Bottom تا MAX می شمارد و دوباره از MAX تا Bottom .

در مد مقایسه گر خروجی Non – inverting ، مقایسه گر خروجی (OC2) در هنگام تطابق مقایسه بین TCNT2 , OCR2 هنگامی که شمارنده بطور افزایشی می شمارد ، OC2 پاک می شود و هنگامی که با شیب منفی می شمارد ، OC2 یک می شود .

در مد inverting خروجی معکوس می شود . فرکانس ماکزیمم در عملکرد بصورت شیب دو طرفه نسبت به عملکرد با شیب یک طرفه کمتر است . بطوریکه عملکرد به صورت شیب دو طرفه برای کنترل موتور بیشتر ترجیح داده می شود .

تجزیه تحلیل PWM در مد PWM Phase Correct با 8 بیت در نظر گرفته شده است. در مد PWM Phase Correct ، کانتر به صورت افزایشی تا زمانی می شمارد که مقدار کانتر با مقدار MAX ، یکی شود و زمانی که کانتر به مقدار MAX رسید، وضعیت شمارش عوض می شود . مقدار TCNT2 در یک سیکل کلاک تایمر برابر MAX می شود . دیاگرام زمان بندی برای مد PWM Phase Correct در شکل ۴-۱۸ نشان داده شده است .

مقدار TCNT2 در دیاگرام زمان بندی به عنوان یک هیستوگرام به شرح عملکرد شیب دو طرفه نشان داده شده است . دیاگرام شامل خروجی های inverting PWM و Non inverting –PWM می باشد . خط افقی کوچکی در روی نقطه ماکزیمم TCNT2 قرار دارد نمایانگر تطابق مقایسه ای بین OCR2 و TCNT2 می باشد .



شکل ۴-۱۹-مدPhase correct PWM،دیاگرام زمان بندی

پرچم سرریز تایمر– کانتر (TOV2) هر زمانی که کانتر به مقدار Bottom برسد ، یک می شود . پرچم وقفه ، هر زمانی که کانتر به مقدار Bottom برسد یک وقفه ایجاد می کند .

در مد Phase Correct PWM ، واحد مقایسه ، شکل موج PWM را در روی پایه OC2 ایجاد می کند . با قرار دادن بیت های COM21:0 برابر با یک، PWM در حالت Non inverting قرار می گیرد . و اگر بیت های COM21:0 برابر با 3 شود ، خروجی PWM به صورت inverting می شود . مقدار OC2 تنها زمانی در پایه خروجی ظاهر می شود که پایه مسیر دیتا به عنوان خروجی تعریف شود . شکل موج PWM از طریق صفر و یک کردن ایجاد می شود و این صفر و یک کردن رجیستر OC2 از طریق مقایسه بین OCR2 , TCNT2 ایجاد می شود .

فرکانس خروجی در مد Phase Correct PWM از طریق رابطه زیر قابل محاسبه می باشد :

$$f_{OCNPCPWM} = \frac{f_{clk-I/O}}{N.510} \qquad (۵-۴)$$

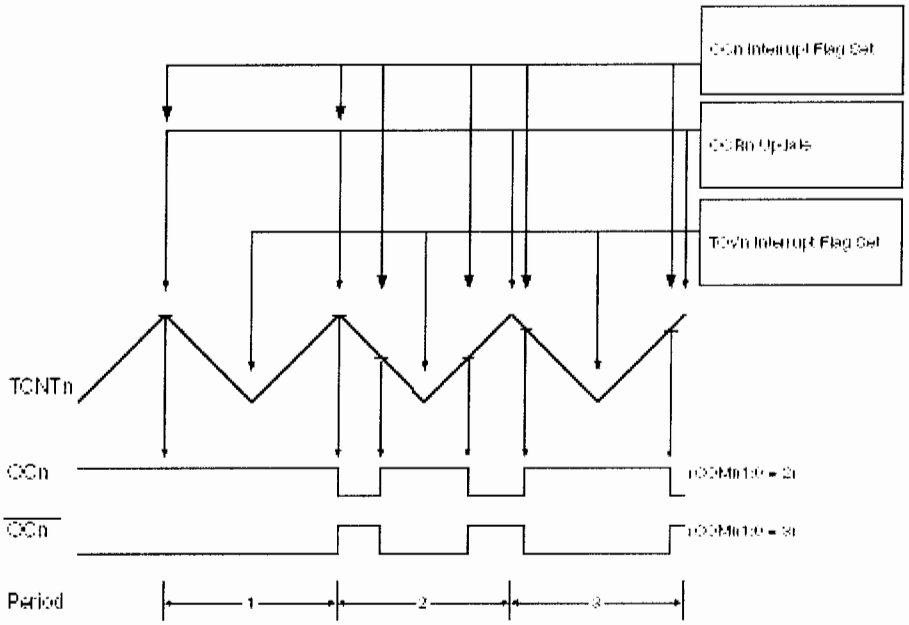
که متغیر N نمایانگر ضریب Prescaler می باشد (1 ، 8 ، 64 ، 256 ، 1024) .

در مد Non – inverting : اگر OCR2 برابر با مقدار Bottom شود خروجی به تدریج صفر می شود و اگر MAX شود خروجی بتدریج high می شود و برای مد inverting دقیقاً خلاف این اتفاق می افتد .

۴-۸-۱۶- دیاگرام زمان بندی Timer / Counter

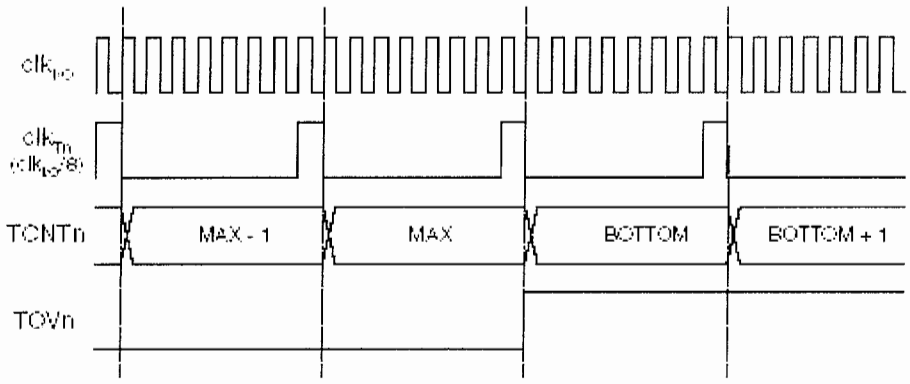
شکل های زیر شامل اطلاعاتی هستند مثلاً اینکه چه زمانی پرچم های وقفه یک می شوند .

شکل ۴-۱۹ شامل زمان بندی دیتا بر مبنای عملکرد Timer / Counter می باشد .

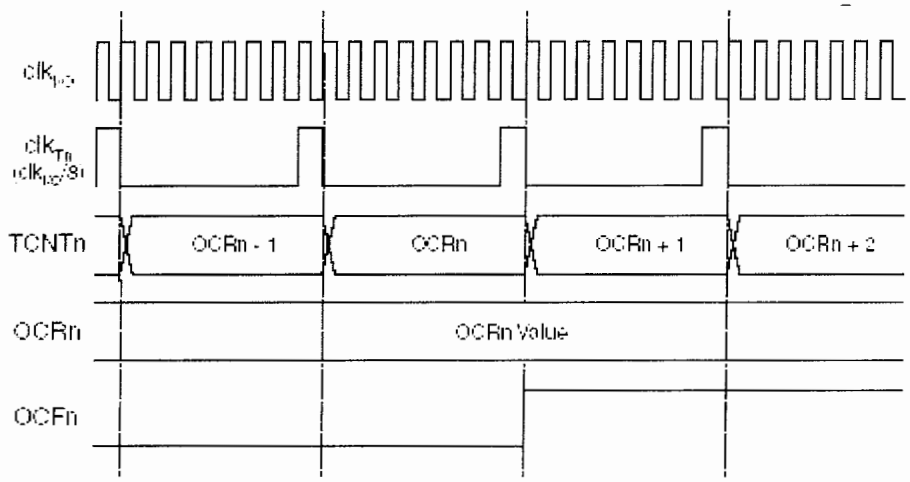


شکل۴-۱۹-دیاگرام زمانبندیtimer/conter بدونPrescaler

شکل ۴-۲۰همان دیتای زمان بندی را نشان می‌دهد ولی با فعال کردن prescaler



شکل ۴-۲۰- دیاگرام زمان بندی با $prescaler(\frac{f_{clk-I/o}}{8})$

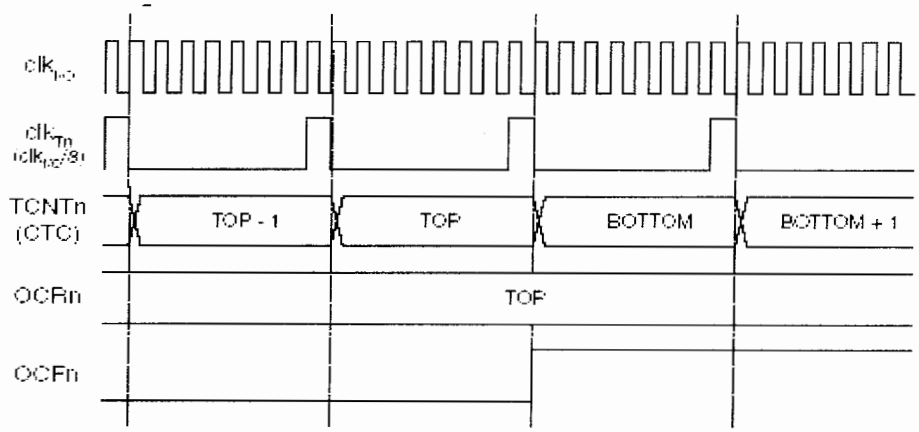


شکل ۴-۲۱- نشانگر یک شدن OCF2 در همه مدها بجز مد CTC می‌باشد .

شکل ۴-۲۱ دیاگرام زمان بندی timer/counter یک شدن OCF2 با وجود

$$(\frac{f_{clx-I/o}}{8})prescaler$$

شکل ۴-۲۲ نمایانگر یک شدن OCF2 و پاک شدن TCNT2 در مد CTC می‌باشد .



شکل ۴-۲۲- دیاگرام زمان بندی timer/counter ، تایمر – کانتردر مد تطابق مقایسه‌ای با وجود

$$(\frac{f_{clx-i/o}}{8})prescaler$$

۴-۸-۱۷- توصیف رجیستر ، timer/counter ، هشت بیتی

کنترل timer/counter از طریق رجیستر TCCR2

Bit	7	6	5	4	3	2	1	0	
	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	TCCR2
Read/Write	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

بیت 7 ام – FOC2 : شدت تطابق مقایسه‌ای خروجی

بیت FOC2 تنها زمانی بیت WGM20 فعال است که مد pwmnon-inverting فعال باشد . برای اطمینان از سازگاری میکرو با قطعات بعدی ، اگر TCCR2 مطابق مد pwm بار شود بیت FOC2 باید صفر شود . زمانی که بیت FOC2 یک می شود ، تطابق مقایسه‌ای فوری در واحد تولید شکل موج شدت می‌گیرد . خروجی OC2 براساس تغییرات بیت های COM21:0 تغییر می‌کند .

توجه شود که بیت FOC2 به عنوان یک سیگنال عمل می‌کند . بنابراین این بیت نمایانگر ارزش بیت های COM21:0 است که تأثیر شدت مقایسه را تعیین می‌کند .

سیگنال FOC2 هیچ وقفه‌ای را ایجاد نمی‌کند . بیت FOC2 همیشه به عنوان مقدار صفر خوانده می‌شود .

بیت 3,6- 0 : WGM21 ، مد تولید شکل موج

این بیت ها توالی شمارش کانتر را کنترل می‌کنند و همچنین برای تعیین نوع تولید شکل موج استفاده می‌شود .

مدهای پشتیبانی عملکرد از طریق واحد timer/counter بصورت زیر می‌باشد :

مد نرمال ، پاک کردن timer در مد CTC ، و دو نوع مد PWM . جدول ۴-۷ را ببینید .

Mode	WGM21 (CTC2)	WGM20 (PWM2)	Timer/Counter Mode of Operation	TOP	Update of OCR2 at	TOV2 Flag Set on
0	0	0	Normal	0xFF	Immediate	MAX
1	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	1	0	CTC	OCR2	Immediate	MAX
3	1	1	Fast PWM	0xFF	TOP	MAX

جدول۴-۷-توصیف بیت مد تولید شکل موج

توجه : اسمهای تعریف بیت CTC2 و PWM2 در حال حاضر غیر قابل استفاده می باشند . از تعریف های WGM21 : 0 استفاده شود . موقعیت و عاملیت این بیت ها با نگارش قبلی TIMER سازگار است .

بیت 5 : 4- 0 : COM21 : مد تطابق مقایسه ای خروجی

این بیت ها رفتار پایه OC2 را کنترل می کند . اگر یکی یا دو تا از بیت های COM 21:0 یک شوند ، خروجی OC2 ، تابعیت پورت ، پایه های I/O را باطل می کند . توجه کنید که بیت DDR

که متناظر با پایه OC2 می باشد به منظور فعال کردن خروجی باید یک شود . زمانی که OC2 بـه پایه وصل می شود تابع بیت های COM21:0 به یک شدن بیت های WGM21:0 بستگی دارد .

جدول ۴-۸ تابعیت بیت 0 : COM21 را نشان می دهد ، زمانی که بیـت هـای WGM21:0 بـا یک نرمال یا یک مد CTC(non – PWM) مطابق می شوند .

COM21	COM20	Description
0	0	Normal port operation, OC2 disconnected.
0	1	Toggle OC2 on compare match
1	0	Clear OC2 on compare match
1	1	Set OC2 on compare match

جدول ۴-۸- مد تطابق خروجی ، مد non – pwm

جدول ۴-۹ تابعیت بیت COM21:0 را نشان می دهد ، زمانی که بیت های WGM21:0 با مـد سریع PWM مطابق شوند .

COM21	COM20	Description
0	0	Normal port operation, OC2 disconnected.
0	1	Reserved
1	0	Clear OC2 on compare match, set OC2 at TOP
1	1	Set OC2 on compare match, clear OC2 at TOP

جدول ۴-۹- مد تطابق مقایسه ای ، مد سریع PWM

توجه : ۱ . یک مورد خاص زمانی اتفاق می افتد که مقدار OCR2 با مقدار TOP یکی شود بیت COM21 ، یک شود . در این مورد ، تطابق مقایسه نادیده گرفته می شود ، ولی یک شدن در مقدار TOP انجام می شود .

جدول ۴-۱۰ تابعیت بیت COM21:0 را نشان می دهد ، زمانی که بیت های WGM21:0 با مد Phase Correct Pwm مطابق شوند .

COM21	COM20	Description
0	0	Normal port operation, OC2 disconnected.
0	1	Reserved
1	0	Clear OC2 on compare match when up-counting. Set OC2 on compare match when downcounting.
1	1	Set OC2 on compare match when up-counting. Clear OC2 on compare match when downcounting.

جدول ۴-۱۰ مد تطابق مقایسه ، مد PHASE CORRECT PWM

توجه : یک مورد خاص زمانی اتفاق می افتد که مقدار OCR2 با مقدار TOP برابر شود و بیت COM21 نیز یک شود در این حالت تطابق مقایسه نادیده گرفته می شود ، ولی صفر و یک کردن در مقدار TOP انجام می گیرد .

بیت CS22 – O:2. : انتخاب کلاک

سه بیت انتخاب کلاک منبع مورد استفاده برای timer / counter را انتخاب می کند .

CS22	CS21	CS20	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	$clk_{I/O}$/(No prescaling)
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)

جدول ۴-۱۱- توصیف بیت انتخاب کلاک

اگر مدهای pin بیرونی برای timer / counter استفاده شوند انتقال به پایه T2 به کانتر کلاک خواهد داد حتی اگر پایه به عنوان خروجی تعریف شده باشد . این ویژگی باعث می شود تا از طریق نرم افزار افزایش شمارش قابل کنترل باشد .

رجیستر TCNT2 ، تایمر – کانتر

Bit	7	6	5	4	3	2	1	0	
	TCNT2[7:0]								TCNT2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

رجیستر TIMER / COUNTER مسیر دسترسی را می دهد هم برای خواندن و هم برای نوشتن در واحد ۸ بیتی کانتردراختیارقرارمی دهد .

نوشتن در بلوک های رجیستر TCNT2 : تطابق مقایسه در کلاک تایمر بعدی را جابجا می کند . تغییر دادن مقدار کانتر (TCNT2) زمانی که کانتر فعال است این ریسک را در بر دارد که ممکن است یک قیاس بین TCNT2 و OCR2 از بین برود .

رجیستر مقایسه خروجی OCR2

Bit	7	6	5	4	3	2	1	0	
	OCR2[7:0]								OCR2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

رجیستر مقایسه خروجی یک رجیستر ۸ بیتی می باشد که بطور مداوم با مقدار TCNT2 مقایسه می شود . یک تطابق (match) ، یک وقفه مقایسه خروجی را ایجاد می کند و یا یک شکل موج در پایه OC2 خروجی ایجاد می کند .

رجیستر پوشش وقفه TIMSK-timer/counter

Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

بیت 7 ام – OCIE2 : فعال کننده وقفه تطابق مقایسه گر خروجی timer/counter2 :

زمانی که بیت OCIE2 یک می شود و I-bit در رجیستر حالت یک باشد ، وقفه تطابق مقایسه timer / counter2 فعال است .

وقفه متناظر ، اگر یک تطابق مقایسه timer / counter2 اتفاق افتد ، اجرا می شود زمانی که بیت OCF2 در رجیستر پرچم وقفه timer/counter(TSFR) یک باشد .

بیت 6 ام -TOIE2 : فعال ساز وقفه سرریز timer/counter2 :

زمانی که بیت TOIE2 یک می شود و I-bit در رجیستر حالت یک باشد ، وقفه سرریز timer/counter2 فعال می شود .

وقفه متناظر زمانی فعال می شود که سرریز در timer/counter2, i.e اتفاق بیافتد ، زمانی که بیت TOV2 در رجیستر پرچم وقفه timer/counter2 (TIFR) یک شود .

رجیستر پرچم وقفه TIFR-timer/counter2

Bit	7	6	5	4	3	2	1	0	
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	TIFR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

بیت 7 ام – OCF2 : پرچم مقایسه خروجی ۲

زمانی که یک تطابق مقایسه بین timer / counter2 و دیتا در OCR2 (رجیستر مقایسه خروجی ۲) اتفاق افتد ، بیت OCF2 یک می شود .

OCF2 بصورت سخت افزاری زمانی که وقفه متناظر اجرا شود، پاک می شود . و بطور متناوب OCF2 با یک کردن پرچم پاک می‌شود . زمانی که 1-bit در SREG ،OCIE2 و OCF2 یک می‌شوند، وقفه تطابق مقایسه اجرا می‌شود .

بیت 6 ام -TOV2 : پرچم سرریز شود timer/counter2 :

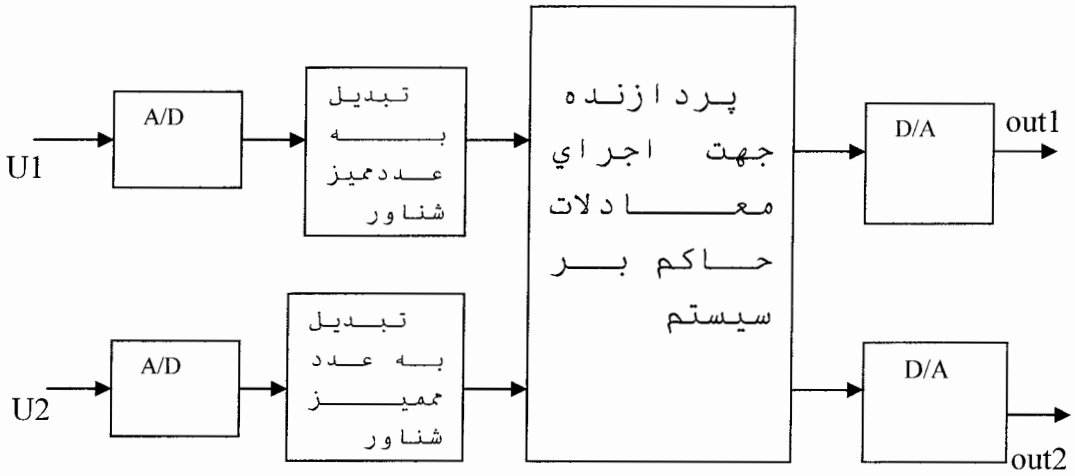
زمانی که timer/counter2 سرریز شود بیت TOV2 یک می‌شود . TOV2 بصورت سخت افزاری پاک می‌شود زمانی که وقفه متناظر اجرا شود . بطور متناوب ، TOV2 با یک کردن پرچم پاک می‌شود . زمانی که I-bit. SERG ، TOIE2 (فعال کننده وقفه سرریز timer/counter2) و TOV2 یک شوند ، وقفه سرریز timer/counter2 اجرا می‌شود . در مد PWM ، این بیت زمانی یک می‌شود که مسیر شمارش timer/counter2 تغییر کند به $00 .

فصل پنجم

سخت افزار پروژه

۵-۱- ساختار شبیه سازسخت افزاری:

سیستم زمان - پیوسته ای که باید در محیط شبیه ساز سخت افزاری شبیه سازی شود ابتدا در محیط مطلب با زمان نمونه برداری مطلوب به یک سیستم زمان – گسسته که جواب آن با سیستم زمان– پیوسته اولیه یکی باشد، تبدیل شده و به برد شبیه ساز سخت افزاری اعمال می شود ورودیهای سیستم نیز با همان زمان نمونه برداری تبدیل سیستم زمان – پیوسته به سیستم زمان- گسسته ، به ورودی A/D شبیه ساز سخت افزاری اعمال شده و پردازنده ضمن دریافت ورودیها و انجام محاسبات ، نتیجه را به ورودی D/A اعمال می کند . در این شبیه ساز، مبدل دیجیتال به آنالوگ(D/A) ، همان PWM موجود در میکروکنترلر استفاده شده در این شبیه ساز میباشد . خروجی زمان – پیوسته سیستم پردازش شده ، از خروجی PWM دریافت می شود ولی در خروجی PWM فقط فرکانس PWM دریافت می شود ، لذا برای دریافت فرکانس سیستم از فیلتری استفاده می شود که فرکانس PWM را حذف کرده و فرکانس سیستم پردازش شده را عبور دهد. لذا بلوک دیاگرام کلی سیستم پیشنهادی بصورت شکل ۵-۱ می باشد.

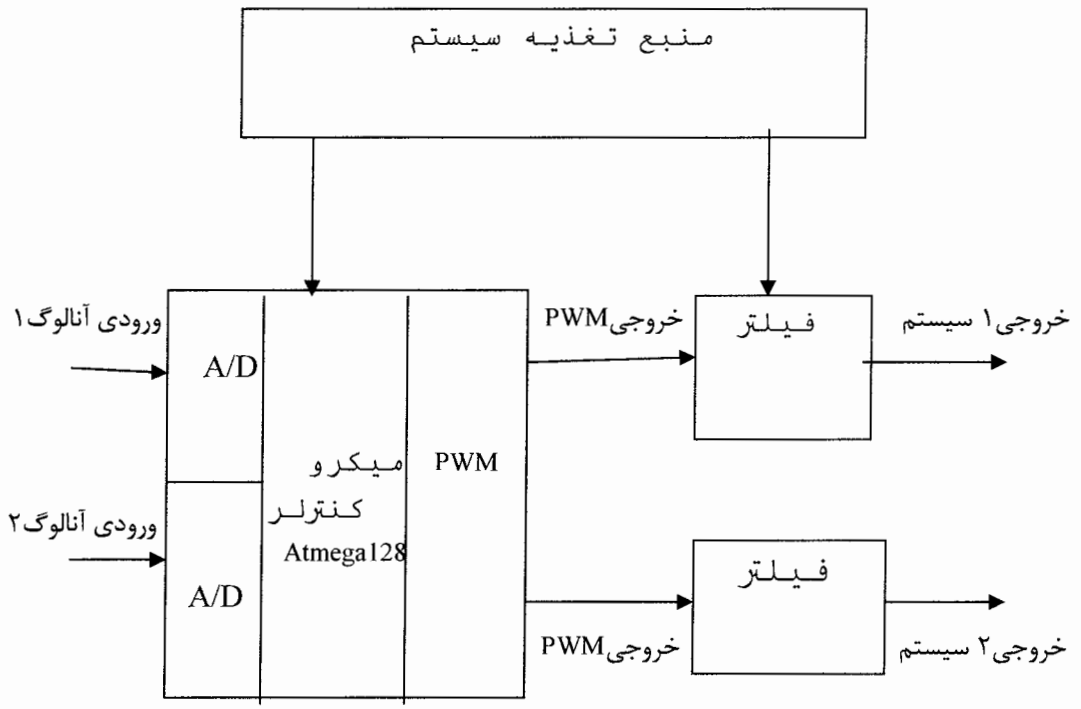


شکل۶-۷-بلوک دیاگرام کلی سیستم پیشنهادی

برای پیاده سازی شبیه ساز مورد نظر در بخش پردازنده از یک میکروکنترلر جدید به نام میکروکنترلر Atmega128 استفاده شده است . علت استفاده از این میکروکنترلر به این دلیل می باشد که این میکروکنترلر نسبت به میکروکنترلرهای موجود از سرعت بالایی برخوردار بوده و به علت دارا بودن کریستال داخلی ، مبدل آنالوگ به دیجیتال و مبدل دیجیتال به آنالوگ به طور قابل

ملاحظه ای از حجم سخت افزار می کاهد . در این پروژه تجزیه و تحلیل سیستم مورد نظر از یک محیط نرم افزاری به یک محیط سخت افزاری منتقل شده است که می توان با مشاهده خروجیهای واقعی ، تغییرات مورد نیاز را بدون اینکه هزینه ای را به ما تحمیل کند انجام دهیم . این روش در مقایسه با تست کردن روی سیستم واقعی ساده تر و ارزانتر می باشد . در این روش اگر مشکلی وجود داشته باشد طراحان می توانند سریعتر و با هزینه کمتر ، بدون خسارت و تجهیزات کم ، آن مشکل را بر طرف سازند . اجزای مورد نیاز برای این شبیه ساز سخت افزاری عبارتند از: منبع تغذیه ، مبدل آنالوگ به دیجیتال ، مبدل دیجیتال به آنالوگ و پردازنده .

دراین مورد خاص از یک میکرو کنترلر Atmega128 استفاده شده است ، که ضمن دارا بودن سرعت بالا برای پردازش سیستم ، به علت دارا بودن مبدل آنالوگ به دیجیتال ، مبدل دیجیتال به آنالوگ و نوسان ساز داخلی از حجم سخت افزار می کاهد . از یک میکرو کنترلر Atmega 128 استفاده شده است . بنابراین بلوک دیاگرام سخت افزار پروژه بصورت شکل ۵-۲ می باشد:

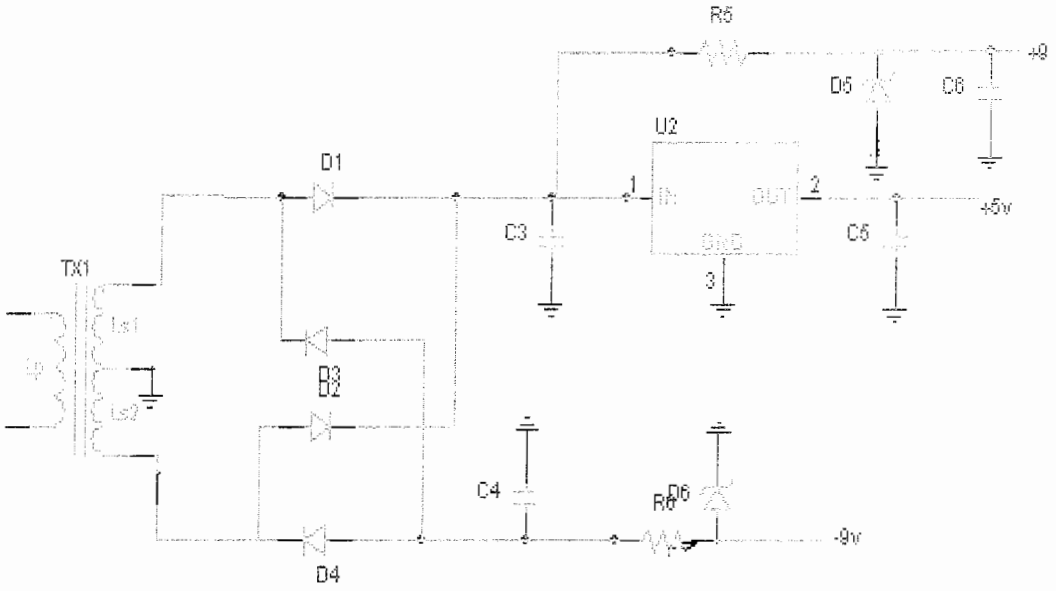


شکل۵-۲- بلوک دیاگرام سخت افزار پروژه

در طراحی سخت افـزار منبـع تغذیـه ، مـدار ریـست ، فیلتـر ، تغذیـه آنـالوگ بـه دیجیتـال و کانکتورهای ارتباطی بایستی مد نظر قرار گیرد که در ادامه هر یک به اختصار تشریح می گردد.

۵-۲- منبع تغذیه

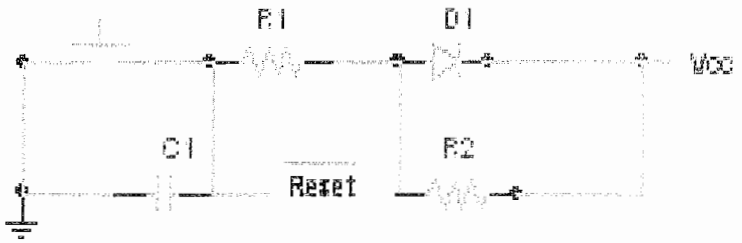
برای ساختن یک منبع تغذیه مانند شکل ۵-۳ ابتدا یک ترانس 7.5V لازم مـی باشـد ، تـا بـرق شهر را به 7.5V تبدیل کند و بعد خروجی به یک پل دیودی وصل می شود ، تا ولتـاژ سینوسـی را یکسو کند و بعد از طریق یک خازن $2200\mu f$ این ولتاژ صاف و به وسیله یک تثبیت کننده ، ولتاژ تثبیت شده و به سیستم اعمال میگردد . حال می‌توان از خروجی این تغذیه به عنوان VCC میکرو استفاده کرد .



شکل ۵-۳- منبع تغذیه سیستم

۵-۳- مدار Reset میکرو

برای شروع به کار مجدد میکرو احتیاج به یک مدار Reset می باشد . که برای این کار یک کلید فشاری با مقاومت 100Ω سری و با یک خازن $10\mu f$ موازی می‌گردد و یـک طـرف کلیـد فـشاری زمین می‌شود و سر مقاومت 100Ω را به یک دیود معمولی و یک مقاومت 10Ω که با هـم مـوازی شده اند وصل می شود و سر دیگر مقاومت و دیود به VCC وصل می‌گردد و خروجی مدار ریست از خازن $10\mu F$ گرفته می شود .



شکل ۵-۴- مدار Reset

۵-۴- عمل فیلترینگ

اغلب میکروکنترلرها خروجی آنالوگ ندارند بلکه با فیلتر نمودن خروجی PWM آنها سیگنال مورد نظر را می توان استخراج نمود . در این میکروکنترلر خروجی PWM موج مربعی با دامنه صفر تا پنج ولت و فرکانسی متناسب با فرکانس کاری میکروکنترلر می باشد. با اعمال اعداد مختلف به ورودی PWM آنچه که تغییر می کند نسبت یک به صفر سیگنال است . با فیلتر نمودن این سیگنال وحذف فرکانس PWM سیگنال اصلی استخراج میگردد . فرکانس قطع این فیلتر باید به نحوی باشد که فرکانس PWM را کاملا حذف و فرکانس سیگنال ورودی را عبور دهد . با تخمین حد اکثر فرکانس سیگنال ورودی ، می توان فرکانس قطع بالای فیلتر را تعیین و به طراحی آن پرداخت . لذا برای فیلتر نمودن فرکانس PWM وفرکانسهای مزاحم از فیلتر پایین گذری در خروجی استفاده می شود . یک فیلتر پایین گذر مرتبه دوم تابع انتقالی بصورت زیر دارد.

$$V_2(s)/V_1(s) = K/(S^2 + a.S + b) \qquad (۱-۵)$$

برای تحقق این فیلتر از عناصر اکتیو استفاده می شود . شکل ۵-۴ این چنین فیلتری را نشان می دهد . با کسکید نمودن این فیلتر می توان فیلتر مرتبه چهار نیز ساخت . با تجزیه و تحلیل معادله فوق مقادیر K و a و b بصورت زیر بدست می آیند .

$$K = 1/(R_1.R_2.C.C_1) \qquad (۲-۵)$$

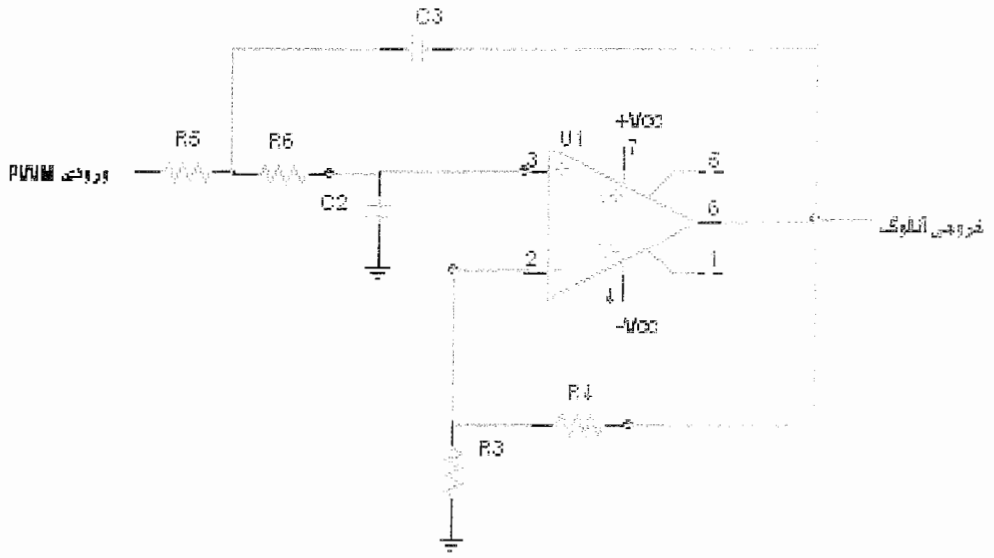
$$a = (1.\mu)/(R_2.C_1) + 1/(R_1.C) + 1/(R_2.C) \qquad (۳-۵)$$

$$b = 1/(R_1.R_2.C.C_1) \qquad (۴-۵)$$

$$\mu = 1 + R_4 / R_3 \qquad (۵-۵)$$

در طرح مورد نظر فرکانس قطع (f) را حدود ۳۰ هرتز انتخاب نموده و با استفاده از نمودارهای مربوطه و با انتخاب K و C مقادیر R_4, R_3, R_2, R_1, C_1 برای بهترین حالت انتخاب می شوند .روش دیگر برای طراحی فیلتر استفاده از نرم افزار Sedif می باشد . در این نرم افزار اطلاعات نوع فیلتر

از طراح خواسته می شود سپس شکل مدار فیلتر و پارامترهای آن مشخص می شود . در طرح مورد نظر فیلتر استفاده شده به شکل زیر می باشد.



شکل ۵-۵- فیلتر اکتیو پایین گذر

۵-۵- راه اندازی نرم افزارمیکرو

پس از تکمیل سخت افزار ، تست و راه اندازی از اهمیت خاصی برخوردار است . بدین جهت بایستی سخت افزار را مرحله به مرحله تست و آزمایش نمود . پس از اطمینان از صحت قسمتهای مختلف ، می توان نرم افزار نهایی را مورد آزمایش قرار داد .

زبان برنامه نویسی می تواند اسمبلی ، C یا بیسیک باشد . با اسمبلی نمی توان براحتی عملیات جمع ، تفریق ، تقسیم وضرب را انجام داد .از آنجاییکه کامپایلر مناسب با زبان C در اختیار نبود از زبان برنامه نویسی Bascom استفاده شد . پروگرمر مورد استفاده STK 400 با مشخصات زیر می باشد. پروگرامر STK 400 توسط گروه مهندسی APPLE II و جهت برنامه ریزی انواع AVR و همچنین تراشه های 89s8252 ، 89s52 و 89s51 طراحی شده است. پروگرامر مورد نظر قابلیت پروگرام کردن میکرو کنترلرهای AVR از جمله ATmega128 را دارد.

مرحله اول (پایه‌های خروجی pulse بخورد)

ابتدا برای تست میکرو و اطمینان از پروگرام شدن میکرو برنامه کوچکی به میکرو داده می شود که یکی از پایه‌های I/O میکرو را صفر و یک کند .

حال برای اینکه یکی از پایه‌های I/O میکرو روشن / خاموش شود، در محیط برنامه نویسی که برنامه آن را خواهید دید ، این پایه را Reset/Set کرده و به جای Reset /Set از دستور Toggle نیز می توان استفاده کرد ، دستور Toggle وضعیت پایه میکرو را عوض می‌کند . با یک LED تغییر وضعیت پایه را می توان مشاهده نمود . آشنایی با دستورات بکار رفته در ضمیمه الف آمده است.

```
'regfile="M103def.dat"                    'MEGA 128 MCU$
Crystal = 1000000$
Baud = 9600                               '8MHZ internal osc$
Dim D As Byte
Config pinb .0=output
Do
Set portb.0.0
Waitms 10                                 ' delay
Reset portb.0
Waitms 10                                 ' delay
Loop
End                                       'end program
```

در این برنامه پایه portb.0 به عنوان خروجی تعریف شده است و بعد از هر بار تغییر وضعیت در portb.0 , 10ms تأخیر در نظرگرفته شده است ، تا بتوان به راحتی خروجی را مشاهده کرد.

مرحله دوم تست و راه اندازی مبدل آنالوگ به دیجیتال

در این مرحله تست ، ADC مدار چک شده و یک برنامه به میکرو داده می شود بطوریکه مقدار آنالوگ را از پایه PF0 میکرو می‌گیرد و به دیجیتال تبدیل می‌کند دستوری که مقدار آنالوگ را می‌گیرد GetAdc (Channel) است و مقدار دیجیتال خروجی بصورت 10 بیتی می باشد که باید آن را به یک 8 بیت و یک 2 بیت تفکیک کرد و خروجی دو پورت را اشغال می‌کند . همانطور که توضیح داده شد باید کانالهایی که استفاده شده اند تعریف گردند، یعنی اگر پورت D به عنوان خروجی استفاده می شود باید تعریف شود که این پورت یک پورت خروجی می‌باشد .

در ابتدا به جای دادن یک موج سینوسی به PF0 (ورودی آنالوگ میکرو) یک ولتاژ از طریق پتانسیومتر به پایه مورد نظرداده شده و مقدار دیجیتال خروجی همان طور که قبلاً ذکر شده از طریق رابطه زیر قابل محاسبه می‌باشد .

$$\text{خروجی} = \frac{input \times 1024}{5} \qquad (۶-۵)$$

در این برنامه پورت F به عنوان ورودی آ نالوگ در نظر گرفتـه شـده اسـت . و از پایـه صـفر آن استفاده شده است . خروجی A/D به پورتهای C و E اعمال می شود و به عنوان خروجی اسـتفاده میشوند . در مرحله بعد به جای یک ورودی DC یک موج سینوسی با فرکانس پائین در حـد - 40 50HZ به PF0 داده،که دامنه این موج سینوسی از 0-5V تغییر می‌کند و به همین ترتیب خروجـی بین 0-3FF تغییر خواهد کرد . توجه کنید که در این مدار باید پایه V_{REF} به VCC وصل شود زیرا که منطق DAC یا PWM بر این بنا شده است کـه V_{REF} بـه VCC وصـل اسـت البتـه در ADC احتیاجی به DAC وجود ندارد ولی بعدا نیاز می شود . همین طور با یک فیلتـر پـائین گـذر بایـد AVCC را به VCC وصل کرده . دقت کنید برای مشاهده خروجی یـک تـأخیر 100ms در برنامـه قرار داده شود . چیزی که جالب توجه است این است که مدار سخت افزار زیادی نـدارد و براسـاس data sheet آن به راحتی می‌توان عملکرد ADC و DAC و قابلیت‌های دیگر میکرو را دید .

```
"Regfile = "M103def.Dat$
Crystal = 1000000                                    ' $Baud = 1200$
Config Adc = Single , Prescaler = Auto
Start Adc
Dim W As Word , Channel As Byte,A As Byte , B As Word
Config Porte = Output
Channel = 0
Do
W = Getadc(Channel)
A=W And &H00FF
B=W And & HFF00
B=High(B)
Portc=A
Porte=B
 Waitms 100
Loop
End
```

مرحله سوم تست و راه اندازی مبدل دیجیتال به آنالوگ (DAC)

طرز کارکرد PWM قبلاً در فصل چهار بطورکامل شرح داده شده است . در این جا برای تست PWM میکرو یک عدد ثابت برای میکرو تعریف می‌شود و شمارنده میکروکنترلر با توجه به برنامه‌ای که به میکرو داده می شود فعال می‌شود و بعد از رسیدن به عددی که در برنامه مشخص شده است خروجی PWM بر اساس تعریفی که شده است صفر یا یک می‌شود و خروجی را می‌توان در پایه OC2 مشاهده کرد و در مرحله بعد با استفاده از حلقه FOR خروجی PWM را یکی یکی اضافه کرده بطوریکه خروجی تا مقدار 1023 می‌شمرد (زیرا PWM ده بیتی می باشد) و خروجی یک شیب خواهد شد و بعد از اینکه تا 1023 شمرد دو باره صفرمی شود . همانطور که قبلاً اشاره شد فرکانس PWM برای یک PWM , 10 بیتی بصورت زیر محاسبه می شود :

PWM FREQUNCY=Foce / (2046 * Prescale)

و از آنجایی که هر چه فرکانس بیشتر باشد تا سرعت نمونه برداری هم بیشتر باشد بهتراست prescale = 1 باشد .

برنامه PWM بصورت زیر است .

```
"Rrgfile="M103def.Dat$
Crystal = 1000000$
Dim C As Word
Config Timer2 = Pwm , Pwm = 10 , Compare A Pwm = Clear Down , Compare B
_pwm
Clear down , prescale = 1=
Config portb=output
Enable interrupts
Enable ocla
C=0
D=0
Pwmla = c
Print c
If c=1024 then c=0
Incr c
Loop
End program'
```

خروجی PWM بصورت یک موج مربعی با زمان وظیفه متناسب با عدد قرار داده شده در رجیستر PWM است . اعداد کوچکتر زمان وظیفه کمتر و اعداد بزرگتر زمان وظیفه بزرگتر دارند. برای تبدیل این اعداد به یک مقدار آنالوگ فیلتر لازم است .

مرحله چهارم (ADC-DAC)

در این مرحله یک شکل موج سینوسی به پایه PF0 داده شده و خروجی را بصورت دیجیتال در پورت D , E می توان دید و مقدار ورودی آنالوگ را در یک متغیر از نوع WORD به نام W قرار داده و PWMla = W مشخص کرده و بدین ترتیب ورودی سینوسی را می توان در خروجی PWM دید . البته برای وضوح کار ، یک فیلتر در خروجی قرار داده میشود که در مورد آن توضیح داده شده است . و فرکانس کار میکرو را نیز 8MHZ قرار داده که برای این کار از طریق programer عمل کرده، تا فرکانس نمونه برداری سریعتر انجام گیرد . و بدین طریق یک موج سینوسی در خروجی مشاهده می شود . برنامه این مرحله درزیر آمده است .

این برنامه ترکیبی از دو برنامه ADC و PWM می‌باشد . و باید توجه شود که در این برنامه هم هیچ تأخیری نباید قرار داد تا خروجی را بتوان بطور واضح مشاهده کرد .

```
"Regfile="M103def.Dat$
Crystal=1000000$
Baud=1200$'
Config Adc = Single , Prescaler = Audo
Start Adc
Dim  W As Word , Channel As Byte , A As Byte , B As Word
Config Porte=Output
Config Portd=Output
Channel = 0
Do
W=Getadc (Channel)
A=W And & H00f
B=W And & Hff00
B=High (B)
Portd= A
Porte= B
Waitms 100
Config Timer2 = Pwm , Pwm = 10 , Compare A Pwm = Clear Down  , Compare B
_Pwm
```

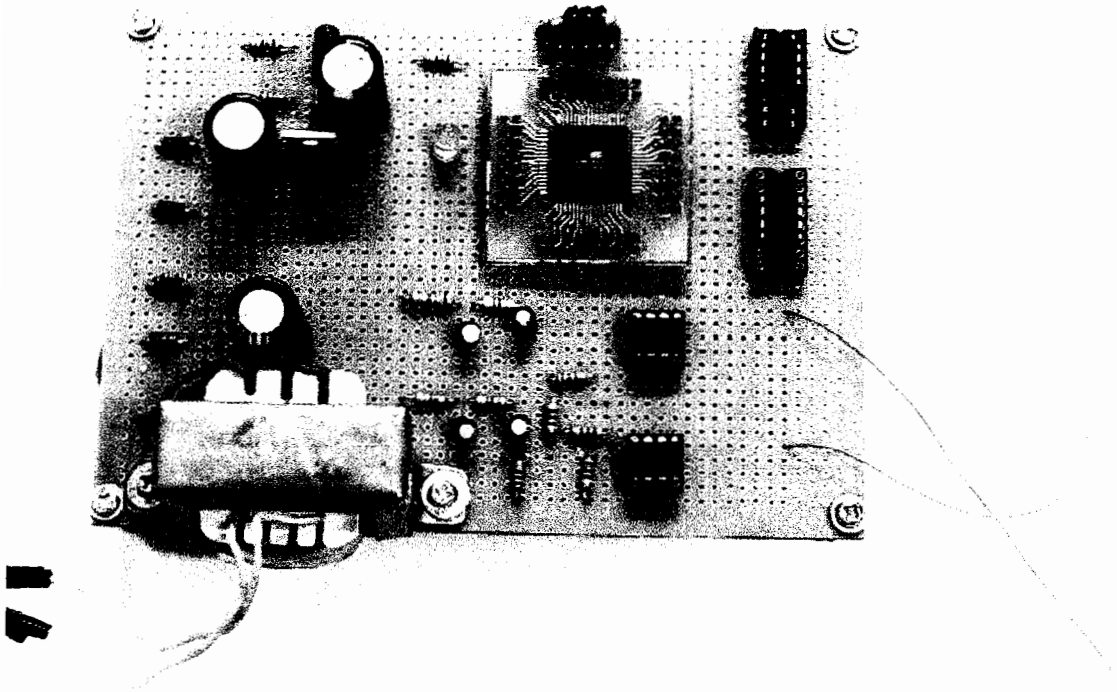
```
Clear Down , Prescale=1
Config Portb = Output
Enable Interrupts
Enable Ocla
Pwmla = W
Wait 1
Loop
End
```

درحال حاضر سخت افزار طراحی و قسمتهای مختلف آن با نرم افزارهای خاص تست شده اند حال باید سیستم گسسته شده مورد نظر و سپس ورودیها را با تأخیر زمانی برابر با زمان نمونه برداری به شبیه ساز اعمال کرده و خروجیها را دریافت کرد .

شکل ۵-۶ پردازنده به کار رفته در مدار شبیه ساز را نشان می دهد.

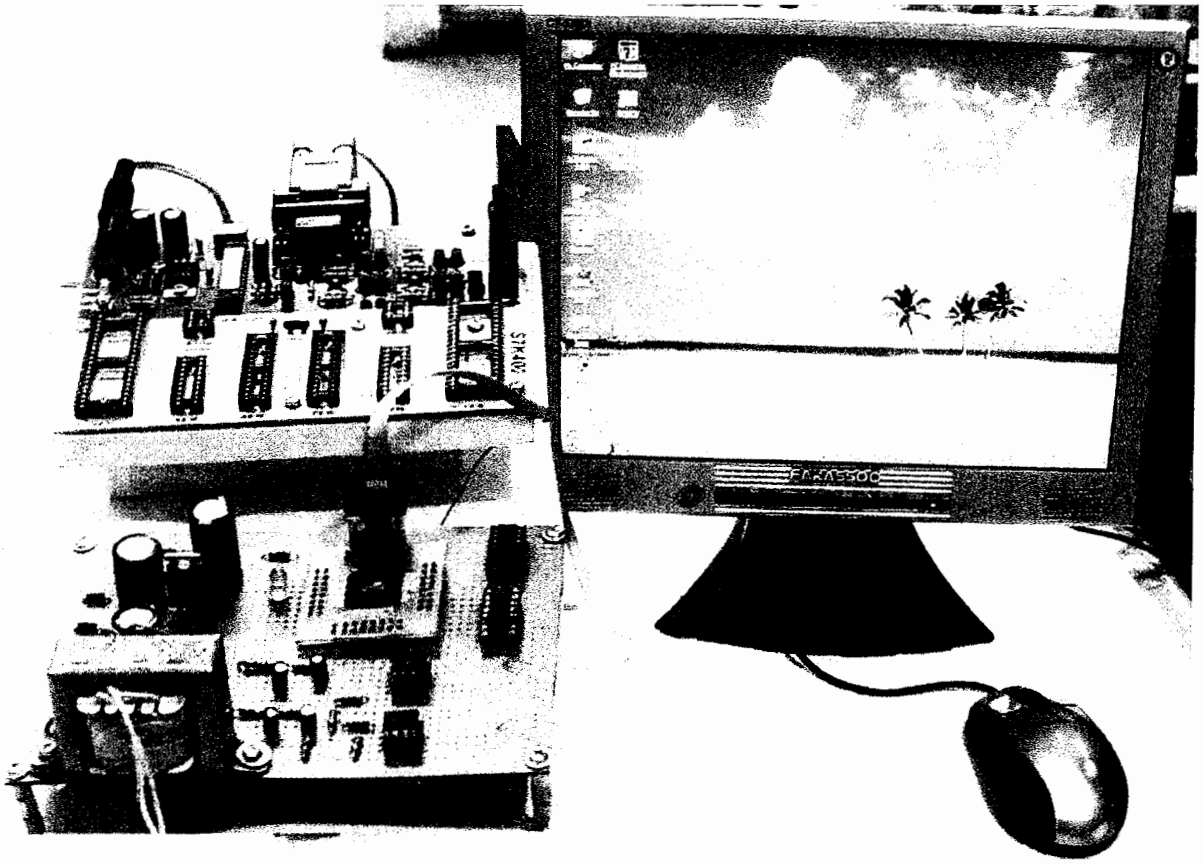
شکل ۵-۶- پردازنده Atmega128

شکل ۵-۷- برد آماده شده شبیه ساز همراه با منبع تغذیه را نشان می دهد.



شکل ۵-۷- برد آماده شده شبیه ساز

شکل ۵-۸ سیستم شبیه ساز و پروگرامر وسیستم ناظر را نشان می دهد.



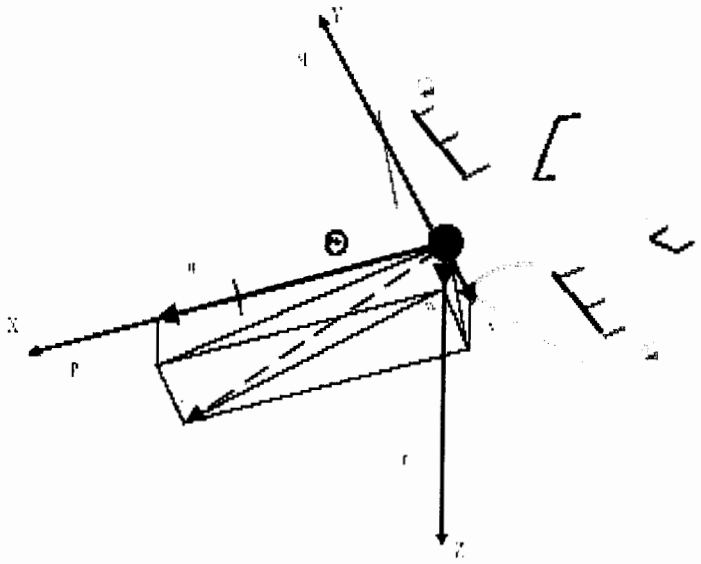
شکل ۵-۸- سیستم شبیه ساز همراه با پروگرامر و سیستم ناظر

فصل ششم

پیاده سازی یک سیستم دو ورودی– دو خروجی روی شبیه سازسخت افزاری

در این فصل یک سیستم دو ورودی – دو خروجی را به عنوان نمونه روی شبیه ساز مورد نظر تست می کنیم . سیستمی که به عنوان مثال در نظر گرفته شده است سیستم خلبان اتوماتیک یک هواپیما می باشد . سیستم مورد نظر یک سیستم خطی که دارای دو ورودی – دو خروجی می باشد ورودیها شامل $\delta_f(t)$: زاویه پیچش ، $\delta_E(t)$: زاویه بالا برنده و برای حالتها $\alpha(t)$ زاویه حمله ، $\theta(t)$ زاویه اوج ، $q(t)$ سرعت اوج پرواز ، $u(t)$ سرعت افقی و γ زاویه مسیر پرواز می باشد . مدل هواپیمای مورد نظر به همراه نمایش پارامترها در شکل ۶-۱ آمده است

۶-۱-مدل هواپیما



شکل۶-۱-نمایش پارامترها روی مدل

۶-۲- معادلات فضای حالت سیستم به شکل زیر میباشد:

$$\begin{cases} \dot{q} = -0.8q(t) - 0.0006\,u(t) - 13.2\alpha(t) - 19\,\delta_E(t) - 2.5\delta_f(t) \\ \dot{u}(t) = -0.014u(t) - -16.64\alpha(t) - 32.2\theta(t) - 0.66\delta_E(t) - 0.5\delta_f(t) \\ \dot{\alpha}(t) = q(t) - 0.0001u(t) - 1.65\alpha(t) - 0.16\delta_E(t) - 0.6\delta_f(t) \\ \dot{\theta}(t) = q(t) \end{cases} \qquad (۱-۶)$$

این معادلات را میتوان در فرم فضای حالت ، به شکل زیر نوشت

$$\begin{bmatrix}\dot{q}\\ \dot{u}\\ \dot{\alpha}\\ \dot{\theta}\end{bmatrix}=\begin{bmatrix}-0.8 & -0.0006 & -13.2 & 0\\ 0 & -0.014 & -16.64 & -32.2\\ 1 & -0.0001 & -1.65 & 0\\ 1 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}q\\ u\\ \alpha\\ \theta\end{bmatrix}+\begin{bmatrix}-19 & -2.5\\ -0.66 & -0.5\\ -0.16 & -0.6\\ 0 & 0\end{bmatrix}\begin{bmatrix}\delta_E\\ \delta_f\end{bmatrix} \qquad (۲-۶)$$

$$\begin{bmatrix}\theta\\ \gamma\end{bmatrix}=\begin{bmatrix}0 & 0 & 0 & 1\\ 0 & 0 & -1 & 1\end{bmatrix}+\begin{bmatrix}q\\ u\\ \alpha\\ \theta\end{bmatrix}+\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\delta_E\\ \delta_f\end{bmatrix} \qquad (۳-۶)$$

روش محاسبه پارامترهای زمان – گسسته معادلات فوق:

برای سیستم پیوسته درفضای حالت داریم:

$$\begin{cases}\dot{X}(t)=AX(t)+BU(t)\\ Y(t)=CX(t)+DU(t)\end{cases} \qquad (۴-۶)$$

برای معادله (۶-۴)پاسخ برابراست با:

$$X(t)=e^{At}X(0)+\int_0^t e^{A(t-\tau)}BUd\tau \qquad (۵-۶)$$

اگرUیک ورودی ثابت(Step)گرفته شود:

$$X(t)=e^{At}X(0)+\int_0^t e^{A(t-\tau)}Bd\tau.U \qquad (۶-۶)$$

حال اگر $X(KT)$ راداشته باشیم برای $X(KT+T)$ داریم:

$$X\left(KT+T\right)=e^{At}X\left(KT\right)+\int_{KT}^{KT+T}e^{A\left(KT+T-\tau\right)}Bd\,\tau.U\left(KT\right) \qquad (۷-۶)$$

یا:

$$X\left[K+1\right]=e^{AT}X\left[K\right]+\int_{0}^{T}e^{A\tau'}Bd\,\tau'.U\left[K\right] \qquad (۸-۶)$$

و

$$Y[K]=C[K]+DU[K] \qquad (۹-۶)$$

بطورکلی معادل گسسته یک سیستم پیوسته باتغییرمتغیر $\tau'=KT+T-\tau$ به شکل زیرمیباشد:

$$\begin{cases} A'=e^{AT} \\ B'=\int_{0}^{T}Bd\tau \\ C'=C \\ D'=D \end{cases} \qquad (۱۰-۶)$$

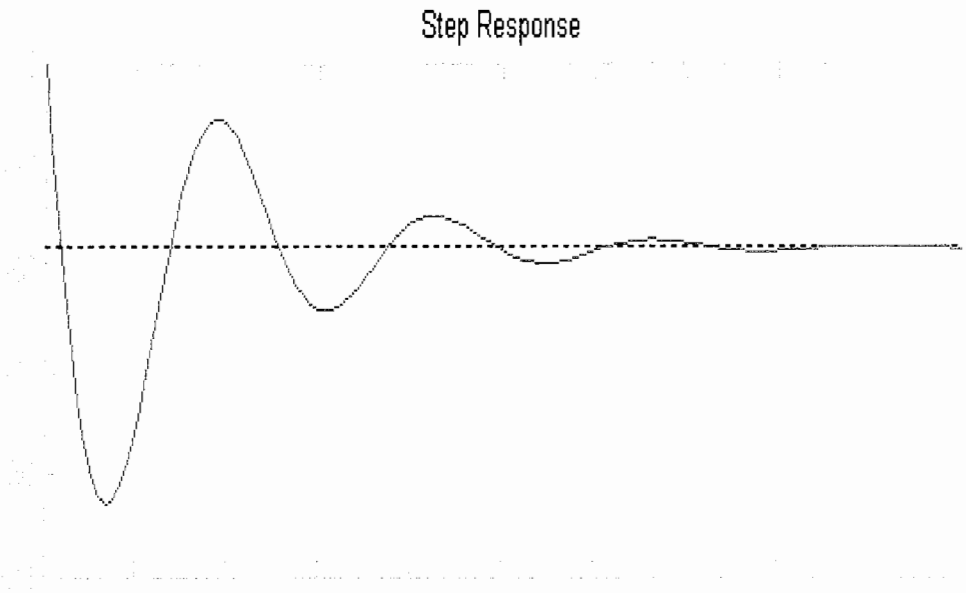
با توجه به روابط بالا مقادیر ماتریسها را درحالت زمان- گسسته در محیط مطلب می توان از برنامه زیر با زمانهای نمونه برداری تجربی محاسبه کرد.

```
clc
clear
T=0.01;
a=[-.8 -.0006 -13.2 0;0 -.014 -16.64 -32.2;1 -0.0001 -1.65 0;1 0 0 0];
ad=expm(a*T);
b=[-19 -2.5;-.66 -.5;-0.16 -0.6;0 0];
syms x
bds=int(exp(a*x)*b,0,T);
bd=double(bds);
c=[0 0 0 1;0 0 -1 1];
cd=[0 0 0 1;0 0 -1 1];
```

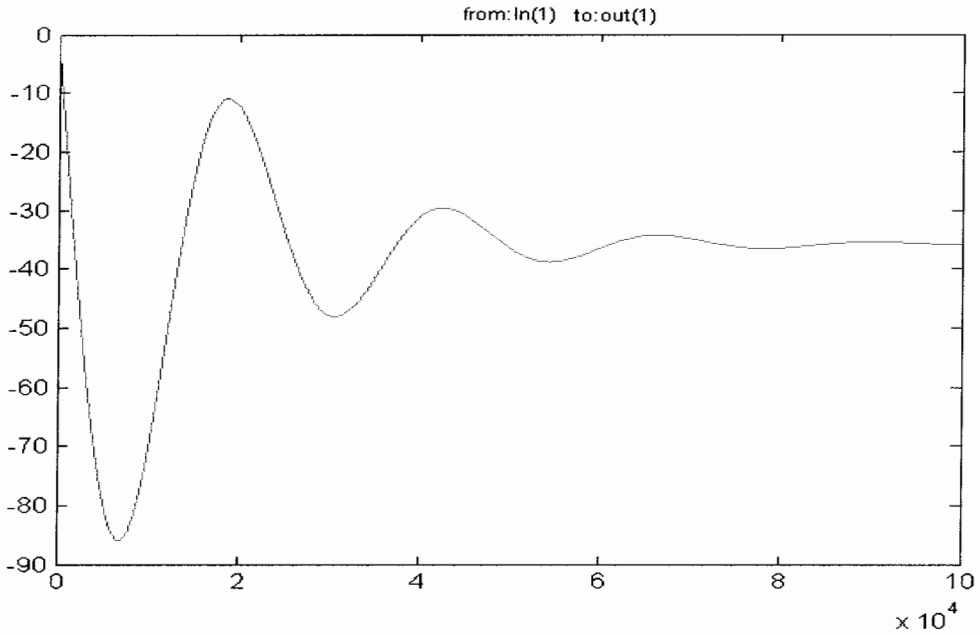
```
d=[0 0;0 0];
dd=[0 0;0 0];
step(a,b,c,d)
r(:,1)=[0;0;0;0];
u=[1;0];
 [as,bs]=c2d(a,b,T)
for i=1:1000/T
r(:,i+1)=as*r(:,i)+bs*u ;
y(:,i)=c*r(:,i)+d*u ;
end
y
figure
plot(y(1,:))
figure
plot(y(2,:))
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
k(:,1)=[0;0;0;0];
u2=[0;1];
 [as,bs]=c2d(a,b,T)
for i=1:1000/T
k(:,i+1)=as*k(:,i)+bs*u2 ;
y(:,i)=c*k(:,i)+d*u2;
end
y
figure
plot(y(1,:))
figure
plot(y(2,:))
```

با توجه به برنامه فوق خروجیهای مدار درحالت زمان- پیوسته و زمان- گسسته در نمودارهای زیر آمده است. از آنجا که سیستم مورد نظر دارای دو ورودی و دو خروجی می باشد با دادن یک ورودی پله به سیستم مورد نظر می توان خروجیهای سیستم را در حالتهای زیر مشاهده کرد . در شکل ۶-۲ خروجیهای سیستم در حالت زمان- پیوسته نشان داده شده است .ابتدا در حالتی که ورودی یک ، تابع پله و ورودی دو ، صفر باشد ، خروجیهای سیستم رسم می شود . سپس در حالتیکه ورودی یک صفر باشد و ورودی دو یک باشد خروجیهای سیستم در حالت زمان- پیوسته رسم می شوند . سپس خروجیهای سیستم در حالت زمان- گسسته با زمان نمونه برداری 0.01s رسم می شود.

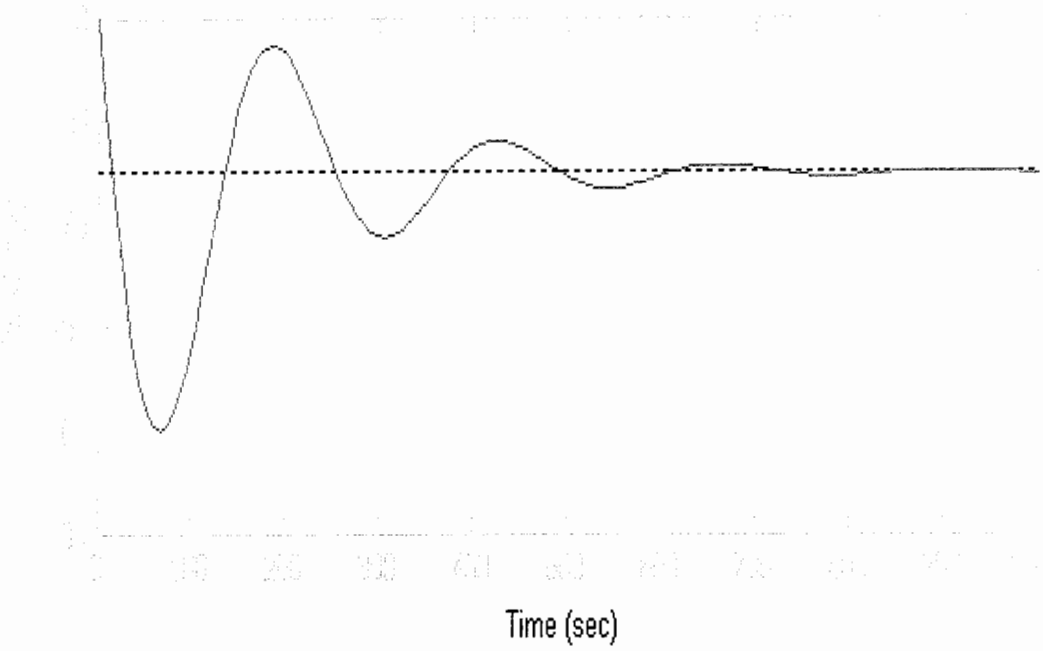
اثر ورودی تابع پله روی خروجیهای سیستم در حالت زمان- پیوسته و حالت زمان- گسسته ، با زمان نمونه برداری $T=0.01s$:



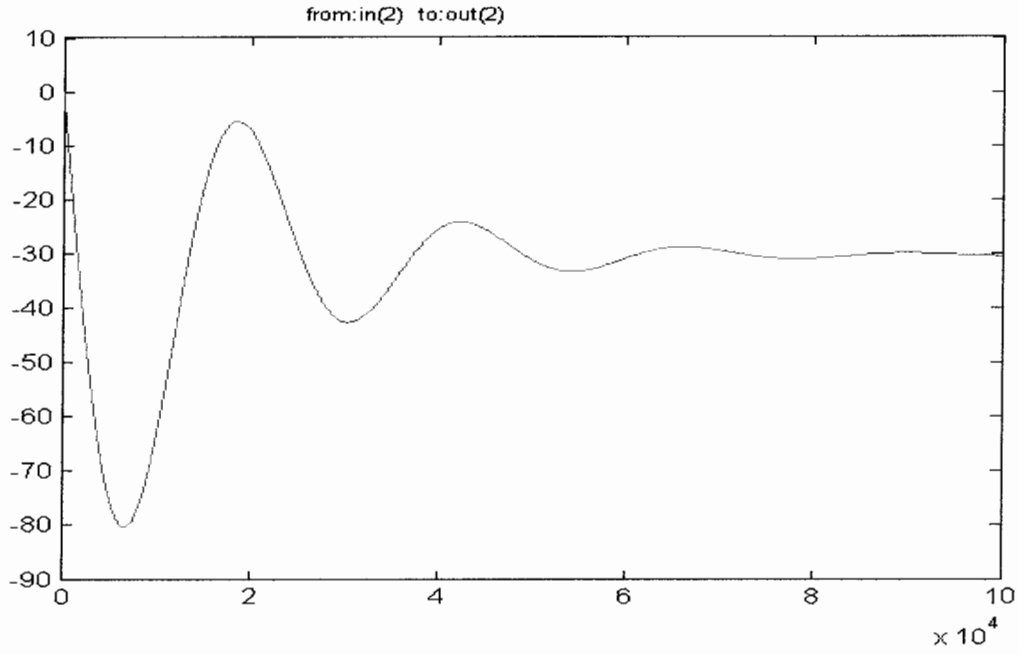
شکل۶-۲- اثر ورودی یک روی خروجی یک در حالت زمان - پیوسته



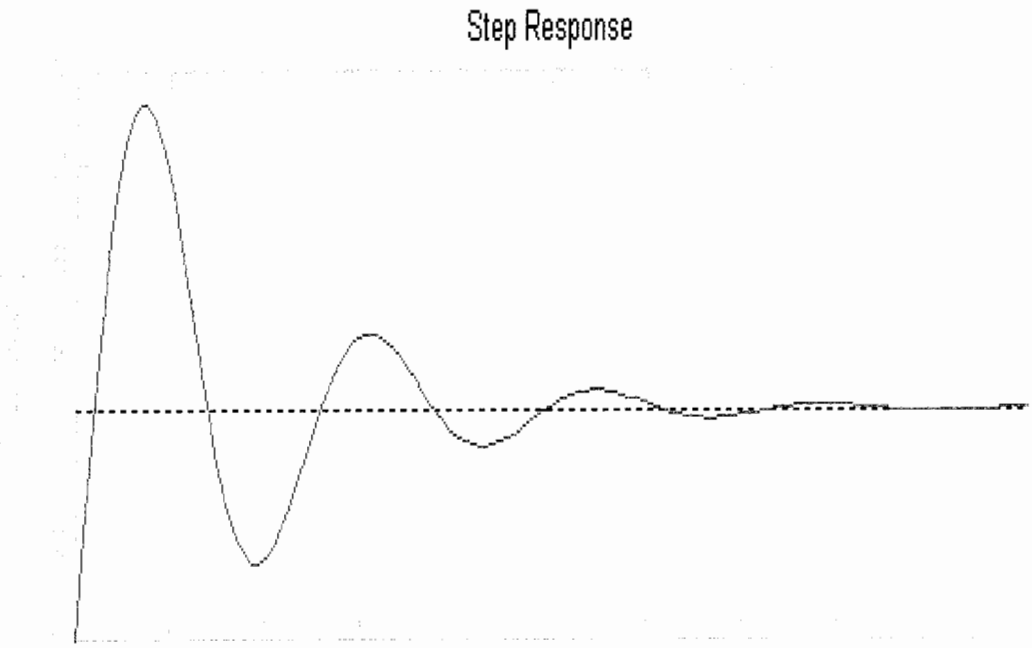
شکل۶-۳- اثر ورودی یک روی خروجی یک در حالت زمان - گسسته



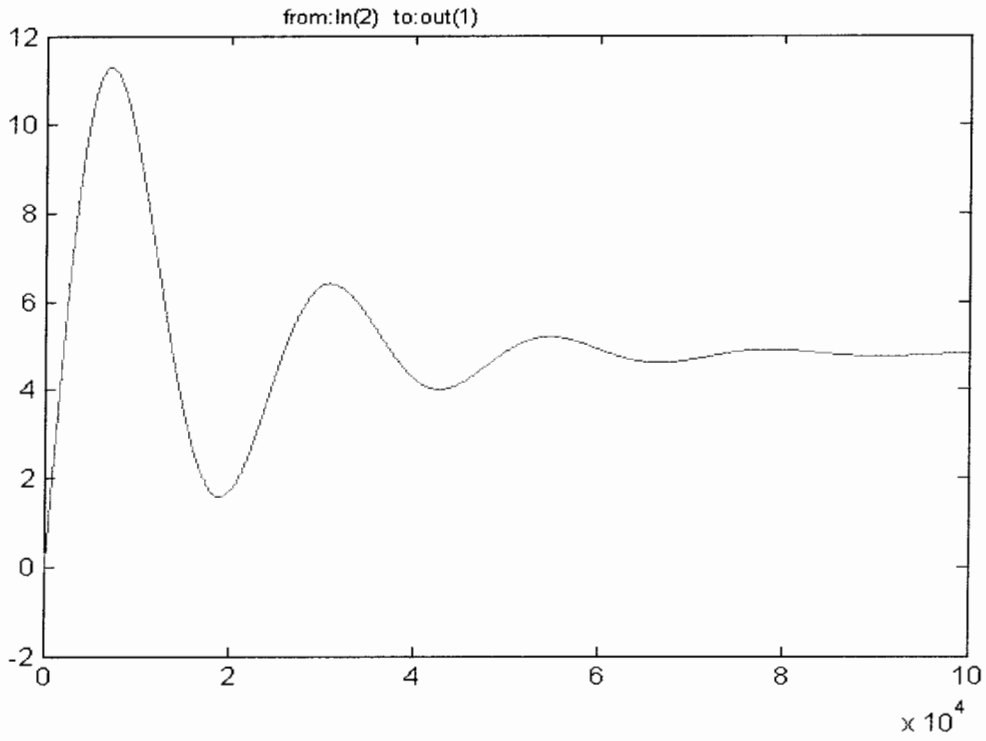
شکل۶-۴- اثر ورودی دو روی خروجی دو در حالت زمان - پیوسته



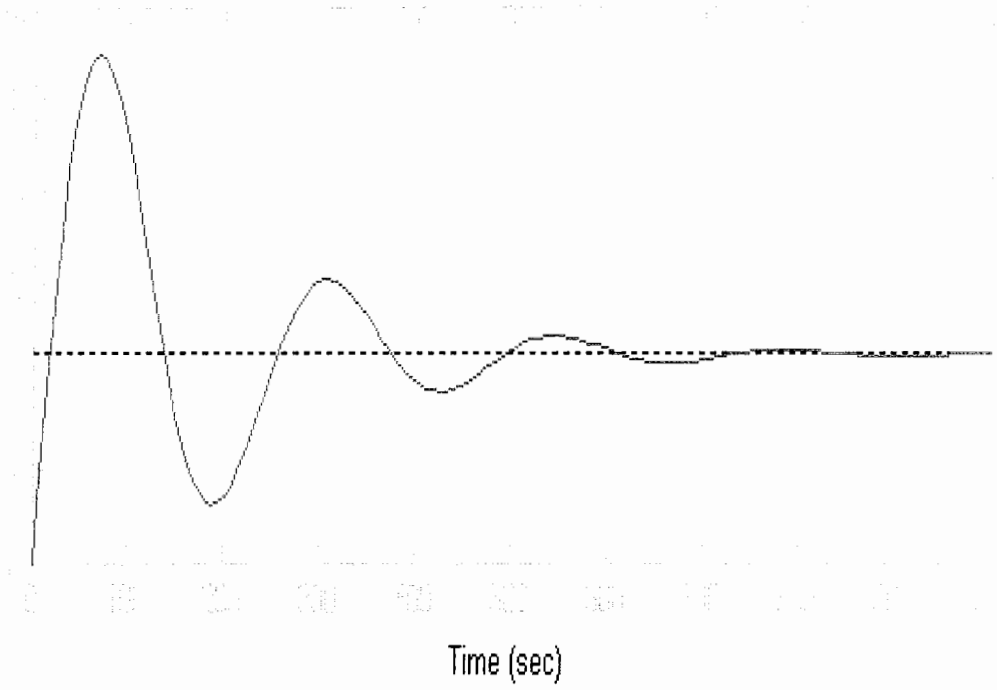
شکل۶-۵- اثر ورودی دو روی خروجی دو در حالت زمان - گسسته



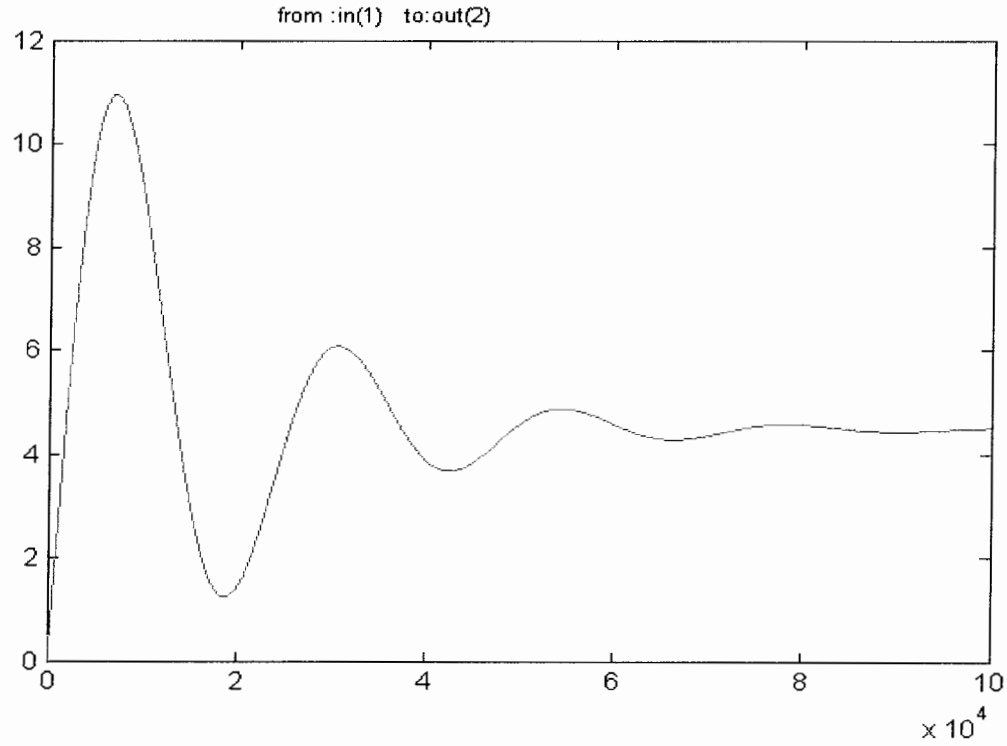
شکل ۶-۶- اثر ورودی دو روی خروجی یک در حالت زمان - پیوسته



شکل ۶-۷- اثر ورودی دو روی خروجی یک در حالت زمان - گسسته



شکل ۶-۸- اثر ورودی یک روی خروجی دو در حالت زمان - پیوسته

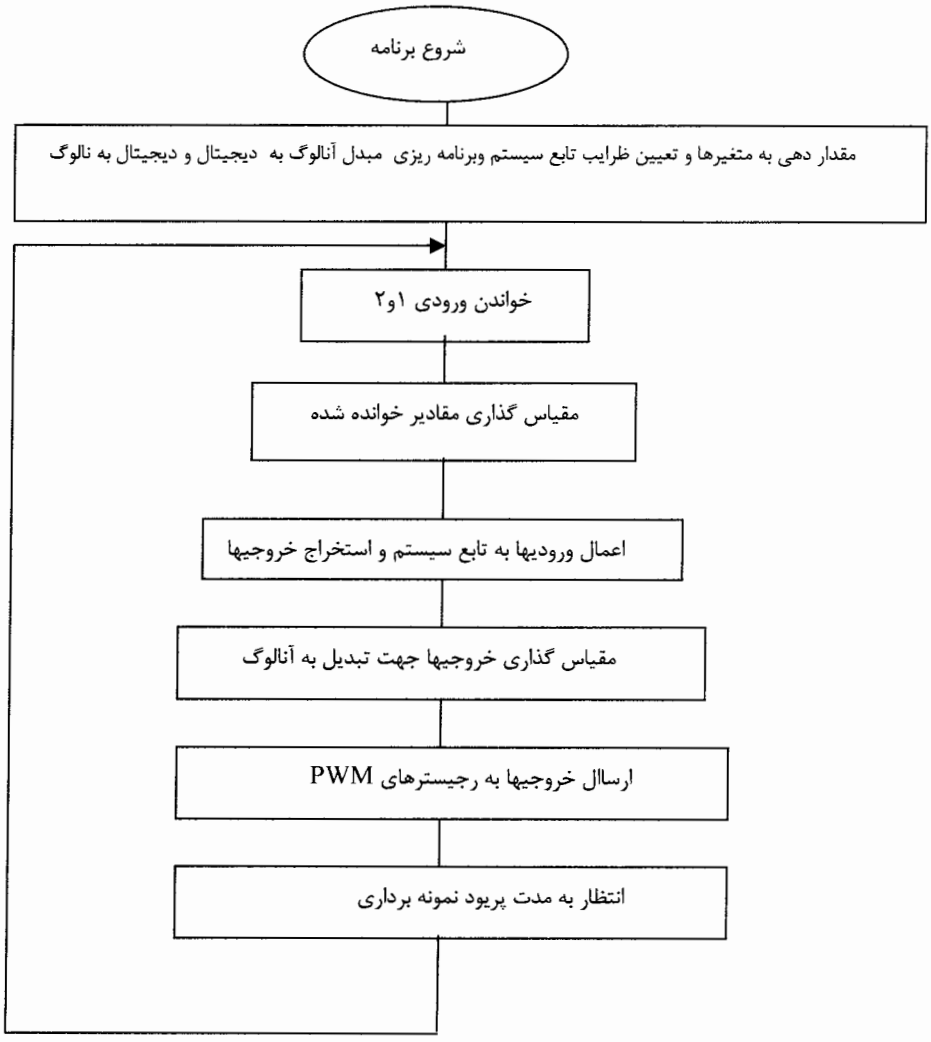


شکل ۶-۹- اثر ورودی یک روی خروجی دو در حالت زمان - گسسته

۶-۳- برنامه شبیه سازی

نرم افزار استفاده شده برای برنامه نویسی، نرم افزار BASECOM می باشد ، استفاده از این نـرم افزار این امکان را فراهم می کند که سیستمهای غیرخطی، حد اکثر چهار ورودی – چهار خروجـی را نیز بتوان با سیستم مورد نظر ، تنها با اضافه کردن چند سطر برنامه، بدون اضـافه کـردن سـخت افزار شبیه سازی کرد ، که این خارج از بحث ما می باشد. حال معادلات حالـت ۶-۲ و ۶-۳ کـه بـا معادلات ۶-۱۰ به یک سیستم گسسته تبدیل شده اند ، شبیه سازی می شوند.

فلوچارت برنامه شبیه سازی:



شکل ۶-۱۰ فلوچارت برنامه

برنامه شبیه سازی سخت افزاری در ضمیمه ب آمده است.

نتیجه:

نتایج نشان می دهند توسط محیط شبیه ساز ساخته شده می توان خروجیهای واقعی سیستم را محاسبه کرده، و قبل از اعمال آن روی سیستم واقعی، اصلاحات و تغییرات لازم را روی کنترل کننده طراحی شده، ایجاد کرد تا از ضررهای احتمالی و اتلاف وقت جلوگیری کرد. علاوه براین:

الف) با توجه به نوع نرم افزار استفاده شده می توان سیستمهای غیرخطی را نیز بصورت زمان واقعی شبیه سازی کرد.

ب)می توان سیستمهای چند ورودی - چند خروجی را نیز شبیه سازی کرد (درشبیه ساز طراحی شده میکرو می تواند تعداد ورودیها تا هشت ورودی را داشته باشد اما تعداد خروجیها چهار خروجی بیشتر نمی تواند باشد.)

ج) درصورت نیاز با تهیهٔ بردهای جانبی می توان از کامپیوتر نیز به عنوان پردازنده استفاده کرد.

د) روش مطرح شده نسبت به نرم افزارهای دیگرمانند مطلب به نمونه واقعی نزدیکتراست.

پیشنهاد

با توجه به مطالب مطرح شده و امکانات سیستم مورد نظر علاوه بر اینکه می توان سیستمهای مختلف را شبیه سازی کرد می توان با طراحی و ساخت یک کنترل کننده از نوع PID ، فازی ، عصبی ، یا ا لگوریتم ژنتیک عملکرد کنترل کننده مورد نظر را بصورت واقعی مورد بررسی قرار داد و اشکالات موجود را رفع و به کنترل کننده واقعی نزدیکتر شد.

((پایان))

((والسلام))

ضمیمه الف:

برخی از دستورات و توابع محیط برنامه نویسی Bascom :

بدنه یک برنامه در محیط Bascom :

بدنه یک برنامه در محیط Bascom شامل تعیین میکرو مورد استفاده، کریستال و پایان و گزینه های اختیاری دیگری است که در زیر معرفی شده اند.

معرفی میکرو:

$REGFILE=VAR

برای شروع یک برنامه در محیط BASCOM ابتدا بایستی میکرو مورد نظر تعریف گردد. VAR نام چیپ مورد استفاده است.

کریستال:

برای مشخص کردن فرکانس کریستال استفاده شده بر حسب هرتز از دستور زیر استفاده می نماییم.

$CRYSTAL=X

X فرکانس کریستال استفاده شده بر حسب هرتز است.

اسمبلی و بیسیک :

در صورت نیاز برای نوشتن برنامه اسمبلی در بین برنامه بیسیک از دستور زیر استفاده می شود.

$ASM
ASSEMBLY PROGRAMME
$ENDASM

یاداشت :

گاهی نیاز است یاداشتهایی برای اطلاعات بیشتر در برنامه اضافه کنید.

REM

آدرس شروع برنامه ریزی حافظه FLASH:

گاهی نیاز است که برنامه خود را از آدرسی دلخواه در حافظه FLASHROM قرار دهید.

$ROMSTART=ADDRESS

تعیین کلاک:

با این دستور در بعضی از میکروهای سری MEGA AVR به صورت نرم افزاری می توان کـلاک سیستم را تغییر داد.

CLOCKDIVISON=var

پایان برنامه:

END

این دستور در انتهای برنامه قرار می گیرد و اجرای برنامه را متوقف می کند.

دستور CONST :

برای تعریف یک ثابت از این دستور استفاده می شود.

مثال:

CONST A = 5

دستور ALIAS :

از این دستور برای تغییر نام متغیر استفاده می شود.

دستور CHR :

از این دستور برای تبدیل متغیر عددی یا یک ثابت به کاراکتر استفاده می شود.

دستور INSTR:

این دستور محل و موقعیت یک زیر رشته را در رشته دیگر مشخص می کند.

دستور INCR:

INCR var

این دستور یک واحد به متغیر عددی var می افزاید.

دستور DECR :

DECR var

این دستور متغیر var را یک واحد کم می کند.

دستور LOW :

این دستور LSB یک متغیر را بر می گرداند

دستور HIGH :

این دستور پر ارزشترین بایت یک متغیر را بر می گرداند.

دستورالعملهای حلقه و پرش:

دستور GOTO و JMP

GOTO label
JMP label

با این دستورات می توان به برچسب label پرش کرد.

دستورالعمل DO – LOOP :

فرم کلی دستور DO...LOOP به صورت زیر می باشد.

DO
Statements
LOOP[UNTIL expression]

دستورالعمل FOR – NEXT :

فرم کلی آن به صورت زیر می باشد.

FOR var = start TO end
Statements
Next var

دستورالعمل WHILE – WEND :

این دستور تشکیل یک حلقه تکرار می دهد که تکرار این حلقه زمانی ادامه می یابد که عبارت بکار برده شده نتیجه FALSE را حاصل کند و یا دارای مقدار صفر شود.

WHILE Condition
Statements
Wend

دستورالعمل IF :

در حالتهای زیر عبارت statements می تواند یک دستورالعمل ساده یا چند دستورالعمل مرکب باشد.

If Expression Then statement

یا

If Expression Then
Statement1
Else
Statement2
End If

دستور EXIT :

با این دستور می توانید فقط از یک ساختار یا حلقه خارج شوید و ادامه برنامه را بعد از ساختار یا حلقه ادامه دهید.

دستور DELAY :

این دستور برای مدت کوتاهی به مقدار $1000\mu s$ در اجرای تاخیر ایجاد می کند.

دستور WAITus :

برای ایجاد تاخیر در برنامه از این دستور استفاده می شود.

WAITus microseconds

دستور WAITms :

برای ایجاد تاخیر در برنامه از این دستور استفاده می شود.

WAITms milliseconds

دستور WAIT :

برای ایجاد تاخیر در برنامه از این دستور استفاده می شود.

WAIT seconds

ضمیمه ۲:

برنامه شبیه سازی :

```
regfile "M103def.dat"                    'MEG128 Mcu$ '
crystal = 1000000$
Config Adc = Single , Prescaler = Auto
Stop Adc
Start Adc
Dim A11 As Single , A12 As Single , A13 As Single , A14 As Single
Dim A21 As Single , A22 As Single , A23 As Single , A24 As Single
Dim A31 As Single , A32 As Single , A33 As Single , A34 As Single
Dim A41 As Single , A42 As Single , A43 As Single , A44 As Single
Dim B11 As Single , B12 As Single , B21 As Single
Dim B22 As Single , B31 As Single , B32 As Single
Dim B41 As Single , B42 As Single
Dim C11 As Single , C12 As Single , C13 As Single , C14 As Single
Dim C21 As Single , C22 As Single , C23 As Single , C24 As Single
Dim D11 As Single , D12 As Single , D21 As Single
Dim D22 As Single
Dim U10 As Integer , U20 As Integer
Dim U1 As Single , U2 As Single , U1a As Single , U2a As Single
Dim Y1 As Single , Y2 As Single , Y1a As Single
Dim Y1o As Integer , Y2o As Integer , Y2a As Single
Dim X11 As Single , X10 As Single
Dim X21 As Single , X20 As Single
Dim X31 As Single , X30 As Single , X40 As Single
Dim X12 As Single , X13 As Single , X14 As Single
Dim X15 As Single , X16 As Single , X17 As Single , Xta1 As Single
Dim Xta2 As Single , Xta3 As Single , Xta4 As Single
Dim X22 As Single , X23 As Single , X24 As Single
Dim X25 As Single , X26 As Single , Xtb1 As Single
Dim X27 As Single , X41 As Single , X42 As Single , X43 As Single
Dim Xtb2 As Single , Xtb3 As Single , Xtb4 As Single
```

```
Dim X32 As Single , X33 As Single , X34 As Single
Dim X35 As Single , X36 As Single , X37 As Single , Xtc1 As Single
Dim X44 As Single , X45 As Single , X46 As Single , X47 As Single
Dim Xtc2 As Single , Xtc3 As Single , Xtc4 As Single
Dim Xtd1 As Single , Xtd2 As Single , Xtd3 As Single , Xtd4AsSingle
Dim Y11 As Single , Y12 As Single , Y13 As Single
Dim Y14 As Single , Y15 As Single , Y16 As Single , Ya1 As Single
Dim Ya2 As Single , Ya3 As Single , Ya4 As Single
Dim Y21 As Single , Y22 As Single , Y23 As Single
Dim Y24 As Single , Y25 As Single , Y26 As Single , Yb1 As Single
Dim Yb2 As Single , Yb3 As Single , Yb4 As Single
Config Timer1 = Pwm , Pwm = 8 , Compare A Pwm = Clear Up ,
Compare B Pwm = Clear Down
A11 = 0.9992
A12 = -0.0000
A13 = -0.0132
A14 = 0.0000
A21 = 0.0000
A22 = 1.000
A23 = -0.0166
A24 = -0.0322
A31 = 0.0010
A32 = -0.0000
A33 = 0.9983
A34 = 0.0000
A41 = 0.0010
A42 = -0.0000
A43 = -0.0000
A44 = 1.0000
B11 = -0.0198
B12 = -0.0036
B21 = -0.0198
B22 = -0.0036
```

```
B31 = -0.0198
B32 = -0.0036
B41 = -0.0198
B42 = -0.0036
C11 = 0
C12 = 0
C13 = 0
C14 = 1
C21 = 0
C22 = 0
C23 = -1
C24 = 1
D11 = 0
D12 = 0
D21 = 0
D22 = 0
X10 = 0
X20 = 0
X30 = 0
X40 = 0
Do
Waitms 1
U10 = Getadc(0)
Waitms 1
U20 = Getadc(1)
U1a = U10 * 256
U1 = U1a / 5
U2a = U20 * 256
U2 = U2a / 5
X12 = A11 * X10
X13 = A12 * X20
X14 = A13 * X30
X15 = A14 * X40
```

X16 = B11 * U1
X17 = B12 * U2
Xta1 = X12 + X13
Xta2 = X14 + X15
Xta3 = X16 + X17
Xta4 = Xta1 + Xta2
X11 = Xta3 + Xta4
X22 = A21 * X10
X23 = A22 * X20
X24 = A23 * X30
X25 = A24 * X40
X26 = B21 * U1
X27 = B22 * U2
Xtb1 = X22 + X23
Xtb2 = X24 + X25
Xtb3 = X26 + X27
Xtb4 = Xtb1 + Xtb2
X21 = Xtb3 + Xtb4
X32 = A31 * X10
X33 = A32 * X20
X34 = A33 * X30
X35 = A34 * X40
X36 = B31 * U1
X37 = B32 * U2
Xtc1 = X32 + X33
Xtc2 = X34 + X35
Xtc3 = X36 + X37
Xtc4 = Xtc1 + Xtc2
X31 = Xtc3 + Xtc4
X42 = A41 * X10
X43 = A42 * X20
X44 = A43 * X30
X45 = A44 * X40

X46 = B41 * U1
X47 = B42 * U2
Xtd1 = X42 + X43
Xtd2 = X44 + X45
Xtd3 = X46 + X47
Xtd4 = Xtd1 + Xtd2
X41 = Xtd3 + Xtd4
Y11 = C11 * X10
Y12 = C12 * X20
Y13 = C13 * X30
Y14 = C14 * X40
Y15 = D11 * U1
Y16 = D12 * U2
Ya1 = Y11 + Y12
Ya2 = Y13 + Y14
Ya3 = Y15 + Y16
Ya4 = Ya1 + Ya2
Y1 = Ya3 + Ya4
Y21 = C21 * X10
Y22 = C22 * X20
Y23 = C23 * X30
Y24 = C24 * X40
Y25 = D21 * U1
Y26 = D22 * U2
Yb1 = Y21 + Y22
Yb2 = Y23 + Y24
Yb3 = Y25 + Y26
Yb4 = Yb1 + Yb2
Y2 = Yb3 + Yb4
X10 = X11
X20 = X21
X30 = X31
Y1a = Y1 / 256

```
Y1o = Y1a * 5
Y2a = Y2 / 256
Y2o = Y2a * 5
Config Timer1 = Pwm , Pwm = 8 , Compare A Pwm = Clear Up,
Config Timer1 = Pwm , Pwm = 8 , Compare B Pwm = Clear Down ,
Prescale = 8
Pwm1a = Y1o
Pwm1b = Y2o
Loop
End
```

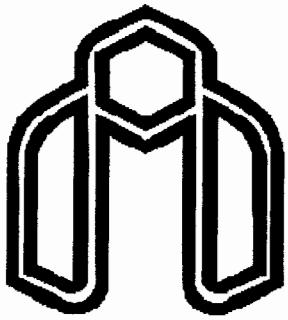
منابع:

[1],[2]use of real Time simulation in a laboratory course
Eric N . Johnson 2003
[3]Testing Embeded control systemes using Hard ware – in the loop Simulation and Tmporal Logic
Marco A.A.Sanvido
[4]Observability in Multiprocessor Real –Time Systems with HardWare/SoftWare Co-Simulation
Mohammed El Shobaki
[5]hard ware in the loop simulation using real time CD++
Lidan li – Gabrial .wainer
[6] from hibrid system simulation to real time implementation
Email:ramine . nikokhah @ inria . fr
[7]طراحی و ساخت کنترلر دیجیتال و شبیه ساز زمان واقعی برای سیستمهای سه ورودی سه خروجی
دکتر علی سلیمانی
[8] مهندس علی کاهه، میکروکنترلرهای AVR
[9] سیستمهای کنترل دیجیتال ، مولف کاتسو هیکو اوگاتا
[10] اینترنت ، ، میکروکنترلر Atmega 128 (www. Atmel .com)
[11] LQG LTR controller design for an aircraft model
Balazs kulcsar
[12] stevens , Briane L . and frankl .Lewis. Aircraft control and simulation . New york john wiley & sons , Inc 1992

Abstract:

In This Thesis , Design and making a kind of hard ware Simulation will be considered .since controller Design for SISO and MIMO System in the soft ware such as matlab is simply possible , so we can unload the considered system in the simulink's environment with diagram's block and tthe functions that are considered and with applying real entrance and receiveing considered exit , we can plan the required controller in the form of PID , FUZZY, NEURAL OR GENETICS ALGORITM.

But if it determines that the system unload really , because the system transferred from a soft ware environment to a hard ware one , so with considering the approximations that are , the desired respond will not obtain , there for we mustreform the controller parameters . but in many cases it seems impossible on many real systems , so it is better to make a system look like a real systemand before controller applies on a real system , it must examine on this system and necessary reform must do , the controller must apply to a real system after attaining assurance of the correct system's respond . so in this thesis , equalization is made in a form of soft ware , which can proceed the considered system in the digital board and with applying real enterance to equalization we can also receive real exits and to examine and reform the controllers that are planned in asoft ware environment such as matlab and to unload after attaining assurance from its function on the controlled system .



Shahrood University of Technology
Department of Electrical and Robotic Engineering

Master Thesis in Control engineering

Design and Making a Hard ware Simulation for Multi Input , Multi Output , Linear And unlinear systems

By: **Esmaeil Toosie**
Supervisor:
Dr.Ali Solimani