

Learning

XMEGA

White

Bascom-AVR , IAR

مسیحی م ح م ی



XMEGA AVR

CLOCK

از جمله اولین نکات این خانواده این است که تا ۳۲ مگاهرتز کلاک را قبول می کنند که در مقایسه با حداکثر ۲۰ مگاهرتزی AVR های معمول، تقریباً دو برابر سرعت را تامین میکنند و همچنین این کلاک ۳۲ مگاهرتز می تواند بصورت اسیلاتور داخلی و بدون نیاز به هیچ کریستالی تامین شود. همچنین برای تنظیم کلاک احتیاج به تغییر فیوزبیت نیست و این کار از طریق تغییر رجیسترهای داخلی انجام می شود. همیشه در ابتدای شروع به کار آی سی ابتدا بصورت خودکار کلاک داخلی ۲ مگاهرتز انتخاب می شود و سپس بوسیله کدهای نوشته شده، نوع کلاک، قابل تغییر است.

یک مکانیزم بسیار جالب در این آی سی قابل فعال سازی است که حتی در صورتی که کلاک خارجی فعال باشد و به هر دلیلی عمل نکند، CPU بصورت اتوماتیک روی کلاک داخلی سوئیچ کند. بنابراین کلیه مشکلاتی که در AVR های معمولی به دلیل تنظیم غلط فیوزبیت ها و یا از کار افتادن کریستال یا اسیلاتور خارجی بوجود می آید، در مورد این آی سی می تواند موضوعیت نداشته باشد. وجود PLL داخلی هم از ویژگی های دیگر این خانواده است که بوسیله آن می توان انواع کلاک ها را از یک منبع ثابت بدست آورد و همچنین یک مکانیزم (Digital Frequency Locked Loop) DFLL هم قابل فعال سازی است که اسیلاتور دائماً مقدار خود را تصحیح می کند و پایداری و صحت کلاک بهبود می یابد.

در مورد راه اندازی Xmega با فرکانس بالاتر از حداکثر ، [overclock Xmega](#) را ببینید.

رجیسترهای متنوع و متعدد کنترلی

به دلیل وجود رجیسترهای کنترلی متعددی که در خانواده XMEGA وجود دارد، امکان نوشتن کدهای بسیار سریعتر و توانمندتری در این خانواده بوجود آمده که حتی در یک کلاک برابر، می تواند سرعت اجرا را بالاتر ببرد.

به عنوان یک مثال، برای تصحیح وضعیت چند بیت یک PORT در حین حفظ وضعیت بقیه بیت ها، در AVR های معمولی باید ابتدا مقدار PORT خوانده شود و بعد از تصحیح مقدار، مجدداً در محل خود نوشته شود. اما در سری XMEGA برای set و reset و not کردن بیت ها، ۳ رجیستر مجزا وجود دارد.

بنابراین بدون آنکه نیازی وجود داشته باشد که مقدار وضعیت فعلی PORT خوانده شود می توان چند بیت آنرا بصورت همزمان set یا reset یا not کرد. امثال چنین امکاناتی می تواند منجر به بهبود راندمان زمان اجرای نرم افزار در طول یک کد طولانی شود و انجام یک کار واحد را برای یک XMEGA که کلاک برابری با یک AVR معمولی دارد، در زمان کمتری میسر کند. در مورد قسمت های دیگر سخت افزار هم کنترل زیادی

از طریق رجیسترهای متعدد داخلی وجود دارد که هرچند شناخت آنرا پیچیده تر می کند، اما دست برنامه نویس را برای انجام عملیات پیچیده و سریعتر، بسیار باز می گذارد.

پورت‌ها

عملکرد پورتها در خانواده XMEGA بسیار کاملتر از AVR های معمولی است. همانگونه که اطلاع دارید، پورتهای AVR یا خروجی هستند که دو وضعیت LOW و HIGH را میتوانند داشته باشند و یا ورودی هستند که [pull up](#) داخلی می تواند فعال یا غیر فعال باشد. اما در XMEGA هم [pull up](#) و هم [pull down](#) قابلیت فعال شدن در ورودی و خروجی را دارند و همچنین وضعیت های [pole totem](#)، [BUS](#)، [Keeper](#)، [OR-Wired](#)، [AND-Wired](#) برای پورتها قابل تعریف است. هر یک از این حالت ها، امکانات مجزایی را برای کار با پورتها در اختیار قرار می دهند که بیشتر آنها در پورتهای AVR عادی قابل پیاده سازی نیست. مثلا اگر چند خروجی در وضعیت OR-Wired قرار داده شوند، این امکان وجود دارد که تمام این پین ها با وجودی که خروجی هستند، به یکدیگر متصل شوند و با قرار دادن [pull down](#) خارجی یا فعال سازی [pull down](#) داخلی، هر خروجی که [high](#) باشد، تعیین کننده وضعیت خروجی کل باشد و سایر خروجی هایی که [low](#) هستند در این وضعیت بی تاثیر باشند. همچنین این امکان وجود دارد که یک منطق NOT برای ورودی یا خروجی فعال شود که مثلا برای وضعیت ورودی، هر منطقی که به پین ورودی اعمال شود به صورت معکوس خوانده شود. یک امکان دیگر پورتهای XMEGA وجود پورتهای مجازی یا [virtual port](#) است که توضیح آن در این مقال نمی گنجد. اما به طور خلاصه به این معناست که یکسری پورت مجازی را به پورتها واقعی نسبت می دهیم و هر عملی که روی پورت مجازی انجام شود، روی پورت معادل آن تاثیر خواهد گذاشت و ۴ پورت مجازی در سری XMEGA وجود دارد.

Totem pole

این وضعیت شبیه عملکرد پورت های AVR های عادی است. به این ترتیب که اگر پورت به عنوان خروجی تعریف شود، LOW و HIGH از یک درایور خروجی تامین می شوند که چه بصورت [source](#) و چه بصورت [sink](#)، جریان را تامین می کند. در وضعیت ورودی هم اگر [pull up](#) فعال نباشد، ورودی حالت [float](#) دارد.

AND-Wired

این وضعیت تا حدودی شبیه پورت های [۸۰۵۱](#) است. به این معنی که در صورت صفر کردن خروجی، یک سوئیچ نیمه هادی روشن می شود و خروجی را [low](#) می کند و در صورت یک شدن آن، باید مقاومت [pull up](#) لازم در خروجی سوئیچ نیمه هادی قرار داشته باشد تا وضعیت [high](#) را ایجاد کند. این مقاومت می تواند بصورت داخلی فعال شود و یا در خارج IC قرار بگیرد. خروجی هایی که در وضعیت AND-Wired قرار

داده شوند، قابلیت وصل شدن به یکدیگر را دارند و آن خروجی یا خروجی هایی که low باشند، خود را به بقیه تحمیل می کنند.

OR-Wired

عکس عملکرد وضعیت قبل را دارد و در صورت یک کردن خروجی، یک سوئیچ نیمه هادی روشن می شود و خروجی را high می کند و در صورت صفر شدن آن، باید مقاومت pull down لازم در خروجی سوئیچ نیمه هادی قرار داشته باشد تا وضعیت low را ایجاد کند.

keeper-Bus

با قرار گرفتن پورت در این وضعیت، این امکان بوجود می آید که وضعیت قبلی پورت زمانی که بصورت خروجی است، در تغییر وضعیت پورت به وضعیت ورودی از نظر low و high بودن همچنان حفظ شود. این کار از طریق فعال سازی هوشمند pull up و pull down بصورت خودکار انجام می شود. به این معنی که اگر وضعیت قبلی خروجی low بوده، pull down را فعال می کند و اگر هم high بوده که pull up را فعال می کند. بنابراین وضعیت پورت از نظر high و low بودن همچنان حفظ می شود.

Virtual port

به دلیل نظام خاص آدرس دهی XMEGA و اینکه بتوان دستوراتی مانند IN و OUT را در مورد پورتهای بکار برد، تعبیه شده است. برای درک بهتر این نوع پورتهای باید به جزئیات آدرس دهی رجیسترهای داخلی XMEGA مراجعه کنید و خواهید دید که آدرس پورتهای در خارج از فضای O memory space/I قرار دارند. بنابراین با map کردن پورتهای واقعی به مجازی و دانستن این موضوع که پورتهای مجازی در فضای O memory space/I قرار دارند، دلیل وجود آنها روشن تر خواهد شد.

تعیین اولویت (priority) وقفه ها

یکی از اشکالات بسیار بزرگ خانواده AVR که از این نظر حتی از سری ۸۰۵۱ هم ضعیف تر می باشد، عدم امکان تعریف اولویت (priority) وقفه هاست. در این خانواده تنها یک اولویت از نوع ابتدایی بین وقفه ها وجود دارد به این معنی که هرگاه دو وقفه با هم از CPU تقاضا شوند، وقفه ای که از نظر آدرس بردار وقفه دارای مقدار کوچکتری است، ابتدا مورد پاسخگویی قرار می گیرد.

اما آن شکلی از تعریف اولویت که در میکروکنترلرهای مانند ۸۰۵۱ وجود دارد، که به CPU این اجازه را می دهد که در وسط اجرای روتین یک وقفه با اولویت پائین تر، به وقفه با اولویت بالاتر پاسخ دهد در AVR های

معمولی وجود ندارد. این موضوع ضعف و خلاء بسیار بزرگی برای AVR محسوب می شود که در برنامه نویسی حرفه ای می تواند بسیار مشکل ساز باشد.

اما تمام این مسائل در سری XMEGA حل شده است و به غیر از اینکه ۳ سطح اولویت برای وقفه ها قابل تعریف است، یک وقفه (Interrupt Non Maskable)NMI هم وجود دارد که اصولاً توسط وقفه های دیگر قابل disable شدن نیست و برای عملیات حساسی مانند از کار افتادن اسیلاتور کریستالی و مانند آن مورد استفاده قرار می گیرد. بنابراین سیستم مدیریت وقفه در خانواده XMEGA بسیار قوی تر از سری های AVR است و از این نظر بسیار کاراتر و توانمندتر عمل می کند.

system Event

، مجموعه ای از ارتباط بین اجزای داخلی است که قدرت و انعطاف زیادی را برای استفاده از امکانات داخلی XMEGA ایجاد می کند. با استفاده از این سیستم می توان بدون درگیر کردن CPU و یا ایجاد هرگونه وقفه ای، تغییرات یک بخش را به عنوان فرمان تغییر و Trigger بخش دیگر به کار گرفت. تایمرها، [ADC](#)، مقایسه کننده آنالوگ، [RTC](#)، پورتهای و کلاک سیستم و فرمان های نرم افزاری می توانند به عنوان منابع ایجاد Event تعریف شوند. همچنین تایمرها، ADC، DAC، DMA و پورتهای و ماژول ارتباط IR می توانند از منبع Event مورد نظر برای تغییرات خود فرمان بگیرند.

به عنوان چند مثال از طریق استفاده از همین سیستم می توان overflow شدن یک تایمر ۱۶ بیتی را به عنوان فرمان clock یک تایمر ۱۶ بیتی دیگر تعریف کرد و بدون استفاده از هر وقفه ای، یک تایمر ۳۲ بیتی ساخت. یا مثلاً می توان سیستم را طوری تنظیم کرد که در مقدار مشخصی از RTC، فرمان تبدیل ADC انجام شود. یا با تغییراتی در یک پین ورودی، خروجی DAC در مقدار جدید قرار بگیرد. در این خانواده ۸ کانال Event در دسترس است که شامل ۸ multiplexer است که با تنظیم آنها می توان هر Event را به عنوان منبع trigger مقصد مورد نظر هدایت کرد. هر کانال Event شامل یک فیلتر دیجیتال است که این امکان را به وجود می آورد که محدوده ای بین ۱ تا ۸ sample را برای پایدار بودن اجباری Event مورد نظر تعریف کنیم تا وجود آن موثر واقع شود.

البته این بحث بسیار گسترده است و نمی توان انتظار داشت که تمام ابعاد آن در اینجا روشن شود. اما بصورت خلاصه وجود System Event باعث ایجاد یک ارتباط سخت افزاری از طریق برنامه نویسی بین اجزای مختلف XMEGA می شود، بصورتی که به عنوان یک مجموعه به هم پیوسته و واحد عمل کنند و این امکان برای یک برنامه نویس توانمند، فرصت مغتنمی محسوب می شود تا بتواند کلربرد های پیچیده و پیشرفته ای را با XMEGA پیاده سازی کند که انجام آن با AVR های عادی مقدور نباشد.

به عنوان کاربردی از Event System که تنها در حد عنوان اشاره می شود، خانواده XMEGA قابلیت و سخت افزار لازم برای اتصال به Quadrature encoder ها(مانند [Shaft encoder](#) هایی که به موتورهای وصل می شوند)، دارا می باشند و تشخیص جهت encoder و مقدار آن و عبور از مبدا را بصورت سخت افزاری ساپورت می کنند.

تایمرها

در خانواده AVR های معمولی، در شماره های ضعیف تر مانند tiny13 فقط یک تایمرهای ۸ بیتی وجود دارد و در شماره های توانمندتر مانند mega128 هم حداکثر ۲ تایمر ۱۶ بیتی و ۲ تایمر ۸ بیتی وجود دارد و حداکثر ظرفیت AVR هم وجود ۴ تایمر ۱۶ بیتی و ۲ تایمر ۸ بیتی در شماره هایی مانند mega2560 است. اما تایمرهای XMEGA همگی ۱۶ بیتی هستند و تعداد آنها هم از ۴ عدد تا ۸ عدد متغیر است. این درحالی است که با استفاده از ظرفیت Event system که قبلا به آن اشاره شد، می توان به تایمرهایی بزرگتر از ۱۶ بیت هم بدون استفاده از وقفه ها دست پیدا کرد.

نکته دیگر اینکه در AVR، کنترلی روی جهت شمارش تایمر وجود ندارد و تایمر در وضعیتهای Normal و CTC و PWM-Fast بصورت افزایشی و در مدهای [Phase corret PWM](#) و [frequency -Phase correct PWM](#) هم بصورت افزایشی و کاهششی متوالی عمل می کند. اما جهت شمارش تایمرهای XMEGA بصورت دلخواه توسط یک بیت کنترلی قابل تغییر است و این امکان بسیار خوبی محسوب می شود.

نکته بسیار مهم دیگر وجود تعداد قابل توجهی خروجی [PWM](#) در خانواده XMEGA است که بین ۱۴ تا ۲۴ خروجی PWM را در شماره های فعلی این خانواده شامل می شود. وجود این خروجی های PWM به علاوه امکان دیگری به نام (AWEX) Extension Advanced Waveform که بعدا توضیح داده می شود، خانواده XMEGA را به ابزار بسیار قدرتمندی برای کاربردهایی مانند کنترل موتور و رباتیک و نظایر آن تبدیل می کند و این در حالی است که تعداد خروجی های PWM در AVR های معمولی بسیار کمتر از این تعداد است (۴ عدد در mega32 و ۸ عدد در mega128 و ۱۶ عدد در mega2560 که حداکثر تعداد در AVR است). ضمن اینکه امکانات کنترل و مدیریت PWM هم در XMEGA قوی تر و کاراتر می باشد.

از جمله کاربردهای این تایمرها، وجود امکان اندازه گیری فرکانس و اندازه گیری عرض پالس بصورت سخت افزاری است که در فرکانس مترها و اندازه گیری های دقیق زمانی بکار می رود و کاربردهای متعدد دیگر که مستلزم آشنایی دقیق با ساختار تایمرهای XMEGA است.

AWeX یا **Advanced Waveform Extension** یک واحد سخت افزاری طراحی شده برای پیاده سازی عملیات کنترل دور موتور و درایورهای قدرت است. وجود ۳ توانایی سخت افزاری **Dead time insertion**، **Pattern generation** و **protection Fault** برای کاربردهای کنترل موتور و مانند آن پیش بینی شده و کارکردن با آن مستلزم آشنایی کامل با مبانی عملکرد درایورهای قدرت می باشد. در مورد **Dead time insertion** می توان این توضیح را ارائه کرد که در یک پل (Bridge) تشکیل شده از سوئیچ های قدرت مانند **Mosfet** یا **IGBT**، بین خاموش شدن هر سوئیچ قدرت و روشن شدن سوئیچ دومی که در همان بازو قرار دارد، باید یک زمان حداقل در حد میکروثانیه تاخیر وجود داشته باشد. در غیر اینصورت قبل از اینکه جریان سوئیچ روشن به صفر برسد، سوئیچ دوم هم روشن خواهد شد و یک جریان بسیار شدید بین ولتاژ مثبت و منفی پل ایجاد می شود و در یک لحظه باعث از بین رفتن سوئیچ های قدرت خواهد شد (به این جریان **shoot through** هم گفته می شود). به این زمان تاخیر به اصطلاح **Dead time** گفته می شود که یک منبع تولید فرمان برای درایور، باید بتواند آنرا بصورت سخت افزاری تامین کند. در واحد **AWeX** خانواده **XMEGA**، چنین امکانی پیش بینی شده که مقدار **Dead time** بصورت نرم افزاری و توسط رجیسترهای کنترلی یکبار تعریف و به سخت افزار اعمال شود.

Pattern generation برای تولید یک **Bit Pattern** مشخص روی پورت خروجی و برای کنترل موتورهای مانند **BLDC** و استپر موتور کاربرد دارد و باید بصورت یک بحث مستقل و کاملاً تخصصی در آینده مطرح شود.

Fault protection هم یک واحد پیش بینی شده برای درایورهای قدرت و در جهت واکنش سریع و مشخص در شرایط وقوع **fault** می باشد. برای کنترل کاربردهای قدرت لازم است این امکان بصورت سخت افزاری پیش بینی شود که در هنگام وقوع **fault** و خطا، واکنش سریع و از قبل معینی انجام شود و مثلاً فرمان خروجی به سوئیچ های قدرت به سرعت قطع شوند و این امر نباید معطل اجرای نرم افزار بماند. چرا که در غیر اینصورت ممکن است به علت جریان کشی بیش از حد به از بین رفتن سوئیچ های قدرت و مواردی از این دست منجر شود. بنابراین واحدی که تحت برنامه ریزی قبلی نرم افزار بصورت سخت افزاری به شرایط خطا پاسخ دهد، از جمله امکاناتی است که در **XMEGA** پیش بینی شده و امکان استفاده از آن را به عنوان یک کنترل کننده و نظارت کننده بر پروسه های در محدوده زمانی میکروثانیه و کمتر، بوجود می آورد.

ADC

تفاوتهای زیادی بین **ADC** داخلی خانواده **XMEGA** با **AVR** های معمولی وجود دارد که شاخص ترین آن دقت و سرعت تبدیل آن است. دقت **ADC** این خانواده ۱۲ بیت است و قابلیت نمونه برداری تا ۲ میلیون نمونه در ثانیه (۲ MSPS) را دارد. از نظر تعداد ورودی آنالوگ هم بین گروه های مختلف **XMEGA** تفاوت

وجود دارد و در گروهی که با پسوند A4 نوشته می شوند، یک کانال با ۱۲ ورودی و در گروه های A3 و A1 هم ۲ کانال مستقل با ۸ ورودی وجود دارند (در گروه های D4 و D3 هم ۱۲x و ۱۶x ورودی وجود دارند).

از جمله نکات دیگر این ADC قابلیت تبدیل ۴ سیگنال آنالوگ بصورت همزمان و با روش Pipeline است و به همین دلیل ۴ سری رجیستر داخلی برای ذخیره سازی این ۴ گروه نتیجه دارد. از نظر ولتاژ مرجع (Reference)، چند امکان برای انتخاب وجود دارد که شامل ولتاژ دقیق ۱ ولت داخلی، $6.1/V_{CC}$ و ولتاژ مرجع خارجی است. در گروه های A1 و A3 دو ورودی مختلف به عنوان ولتاژ مرجع اختصاص داده شده است.

امکان خواندن ورودی ها بصورت تفاضلی، اعمال بهره تقویت ۱ تا ۶۴، و امکان محاسبه مقدار ورودی بصورت علامت دار، از دیگر امکانات ADC این خانواده است.

یکی از موارد مهم قابل ذکر، وجود سنسور دمای داخلی در XMEGA است که نیاز به قرار دادن سنسور دما در خارج از IC را برای سنجش دما برطرف می کند. این امکان وجود دارد که خروجی این سنسور بصورت داخلی به ورودی ADC متصل و مقدار آن محاسبه شود. همچنین امکان اندازه گیری V_{CC} اعمال شده به IC و خروجی DAC و ولتاژ مرجع داخلی هم از طریق ADC وجود دارد.

در عملکرد خانواده XMEGA، یک ساختار function Compare وجود دارد به این معنی که مقدار

ADC بصورت دائمی با یک رجیستر از پیش تعیین شده ۱۲ بیتی مقایسه می شود و می توان وقفه یا

Event را در صورت کوچکتر یا بزرگتر بودن از این رجیستر فعال کرد. و بالاخره اینکه امکان انتقال نتایج

عملیات ADC توسط DMA (که بعدا توضیح داده می شود) در حافظه قابل فعال کردن است و این روش

برای درگیر نکردن CPU و ذخیره سازی سریع اطلاعات در تبدیلات متوالی می تواند بکار برده شود.

DAC

وجود مبدل دیجیتال به آنالوگ ۱۲ بیتی و با سرعت حداکثر ۱ میلیون تبدیل در ثانیه، از جمله مزیت هایی

در خانواده XMEGA است که در خانواده AVR های معمولی به کمری وجود ندارد و زمینه را برای انجام

عملیاتی که با تولید سیگنال های آنالوگ سر و کار دارند، فراهم می کند.

در اینجا ذکر یک توضیح فنی لازم است که هرچند با قرار دادن یک فیلتر پائین گذر در خروجی PWM می

توان سیگنال های آنالوگ را تولید کرد. اما حداکثر فرکانس این سیگنال آنالوگ باید تفاوت قابل توجهی با

فرکانس PWM داشته باشد تا بتواند توسط فیلتر پائین گذر و با دقت بالا، جداسازی شود. حداکثر فرکانس

PWM قابل حصول برای AVR در مد fast PWM و با فرض کلاک ۲۰ مگاهرتز و ۸ بیتی بودن PWM،

برابر ۷۸.۱۲۵ کیلوهرتز است. بنابراین تغییرات سیگنال آنالوگ تولید شده توسط PWM باید بسیار کمتر از

این باشد تا بتواند به خوبی جدا شود. اما وجود DAC، زمینه را برای کاربردهایی مانند تولید صدای خروجی

بصورت استریو و یا تولید سیگنال های آنالوگ با فرکانس نسبتا بالا را برای XMEGA فراهم می کند.

هر واحد DAC دارای دو خروجی مستقل آنالوگ است که هریک رجیستر دیتای خاص خود را دارند. در گروه با پسوندها A3 و A4 یک واحد DAC و در گروه A1 دو واحد DAC (یعنی ۴ خروجی مستقل آنالوگ) وجود دارند.

برای تبدیلات DAC یک ولتاژ مرجع قابل تعیین است که می تواند از Vcc یا ولتاژ مرجع داخلی و یا ولتاژ دلخواه متصل به پایه تعریف شده برای این کار، تامین شود.

خروجی آنالوگ DAC بصورت نرم افزاری قابل کالیبره کردن و تغییر Offset خروجی است و این کار می تواند به کمک اتصال داخلی خروجی DAC به ورودی ADC و خواندن مقدار آن و یا قرائت خروجی آنالوگ از بیرون انجام شود.

خروجی DAC همچنین می تواند بصورت داخلی به مقایسه کننده آنالوگ موجود در XMEGA متصل و ولتاژ آن با ولتاژهای دیگری مقایسه شود که این امکان برای برخی کاربردهای کنترل جریان و درایورهای قدرت، امکان فوق العاده ای محسوب می شود.

برای تولید سیگنال های آنالوگ با فرکانس نسبتاً بالا از طریق قرار دادن متوالی دیتا در ورودی DAC، امکان بهره گیری از DMA داخلی XMEGA وجود دارد و در این شرایط بدون مشغول کردن بیجهت CPU، عملیات تولید سیگنال آنالوگ به خوبی انجام می شود. همچنین عملیات به روز سازی خروجی آنالوگ می تواند تحت مدیریت Event system انجام شود و بدون ایجاد بار نرم افزاری برای CPU و ایجاد هرگونه وقفه ای، وقوع Event تعیین شده، باعث به روز سازی و ایجاد تغییر مقدار در خروجی آنالوگ مورد نظر گردد.

DMA controller

کنترل کننده (DMA Direct Memory Access) برای انتقال اطلاعات بین نواحی مختلف حافظه و رجیسترهای داخلی بکار می رود و بکارگیری صحیح آن، باعث کاهش بلو پردازشی است که می تواند برای انتقال این اطلاعات به CPU تحمیل شود. وجود امکاناتی نظیر DMA و Event system در داخل XMEGA سبب می شود که کارایی و سرعت انجام عملیات در مقایسه با AVR های معمولی به شکل چشمگیری افزایش پیدا کند و در شرایطی که در AVR، انجام هر عملی باید با دخالت مستقیم CPU انجام شود، در XMEGA عملیات مختلف می توانند بصورت کاملاً موازی و بدون مشغول کردن CPU به انجام برسند. بنابراین راندمان اجرای نرم افزار با کمک سخت افزار بسیار بالا می رود و شاید بتوان مثال هایی را مطرح کرد که برای یک XMEGA با کلاک ۲۰ مگاهرتز قابل انجام باشد ولی انجام آن در همان زمان برای یک AVR معمولی، اگر بجای ۲۰ مگاهرتز مثلاً ۱۰۰ مگاهرتز هم کلاک قابل اعمال کردن بود، مقدور نباشد. ۴ کانال DMA مستقل در XMEGA وجود دارد که می توانند اطلاعات را در بلوک هایی با سایز قابل تعریف جابجا کنند و امکان تعریف اولویت (priority) هم بین این کانال ها وجود دارد.

بخش DMA از بخش های نسبتاً پیچیده XMEGA از نظر نرم افزاری محسوب می شود و کار کردن با آن مستلزم تبحر و تسلط کافی به سایر بخش هایی است که DMA روی آن عمل می کند و برای کار با DMA باید یک شناخت جامع از اکثر بخش های XMEGA که با مسئله DMA ارتباط پیدا می کنند، وجود داشته باشد.

(External Bus interface)EBI)

EBI یا External Bus interface واحد سخت افزاری اتصال به وسایل جانبی آدرس پذیر است که بطور خاص تا ۶۴ مگا بایت [SRAM](#) و ۱۲۸ مگابایت [SDRAM](#) را ساپورت می کند. این واحد عملیات Refresh کردن SDRAM را تحت کنترل نرم افزار به عهده می گیرد و از انواع ۴ و ۸ بیتی آن پشتیبانی می کند. یکی از نکاتی که تاکنون ذکر نشده این مطلب است که در ساختار داخلی XMEGA، دو کلاک ۲ برابر و ۴ برابر کلاک CPU وجود دارد که برای بعضی واحدهای داخلی استفاده می شود. EBI بطور خاص از کلاک دو برابر استفاده می کند و بنابراین تا کلاک حداکثر ۶۴ مگاهرتز را پشتیبانی می کند. در پیکر بندی های مختلف واحد EBI، دو یا سه یا چهار پورت بکار گرفته می شوند. در نوع اول با قرار دادن دو [Latch](#) در خروجی پورت J از این پورت سیگنال های ۸ خط دیتا و ۱۶ خط آدرس ساخته می شود و خطوط کنترل هم از پورت H تامین می شوند. در پیکر بندی با استفاده از ۳ پورت، امکان تولید ۱۶ و ۲۴ بیت آدرس با استفاده از یک یا دو Latch وجود دارد و امکان اتصال به SDRAM هم در حالت دیتای ۴ بیتی و بدون استفاده از هر گونه Latch فراهم است. در پیکر بندی با استفاده از ۴ پورت هم امکان تولید ۲۴ بیت آدرس با استفاده از یک Latch و امکان اتصال به SDRAM هم در حالت دیتای ۸ بیتی بدون Latch فراهم است. به غیر از حافظه ها، سایر وسایل جانبی که دارای Data Bus، Address Bus و سیگنال های کنترلی Read Enable و Write Enable و select Chip هستند (مانند [LCD](#) ها، [PPI](#) و ...)، قابلیت اتصال به XMEGA از طریق EBI را دارند.

Engines Crypto

یکی از امکانات فوق العاده و درخشان خانواده XMEGA، وجود سخت افزار لازم برای کد کردن اطلاعات با الگوریتم های (DES) Encryption Standard Data و (AES) Advanced Encryption Standard است. برای توضیح در مورد این دو استاندارد می توانید [Data Encryption Standard](#) و [Advanced Encryption Standard](#) در ویکی پدیا را مطالعه کنید.

اساس کد کردن اطلاعات در الگوریتم DES بر مبنای یک کلید ۵۶ بیتی انجام می شود که با احتساب ۸ بیت به عنوان parity، عملیات روی ۶۴ بیت دیتا و یک کلید به طول ۶۴ بیت انجام می شود. روش کار به این صورت است که دیتای مورد نظر در رجیسترهای R0-R7 و کلید مورد نظر در رجیسترهای R8-R15

قرار می گیرند. در مجموعه دستور العمل های XMEGA، دستور اسمبلی DES به همین منظور پیش بینی شده که با ۱۶ بار اجرای آن و بسته به وضعیت H flag در رجیستر SREG، عملیات Encryption یا Decryption انجام می شود.

برای کد کردن با امنیت بالاتر اطلاعات، یک روش چند مرحله ای با عنوان DS^۳ پیاده سازی می شود که اطلاعات در ۳ مرحله و با ۳ کلید مختلف به ترتیب Encrypte و بعد Decrypte و مجدداً Encrypte می شود که این روش از امنیت بسیار بالاتری نسبت به DES تک مرحله ای برخوردار است و البته به سادگی با XMEGA قابل پیاده سازی است.

برای شکستن اطلاعات کد شده توسط این روش، نیاز به ارقام نجومی محاسبات و تست کلیدهای مختلف است. چنانکه در [این](#) آدرس مشخصات سیستمی ذکر شده که با بکار بردن ۱۸۵۶ پردازنده و صرف هزینه ۲۵۰۰۰۰ دلار، مدت ۹ روز طول کشیده تا بتواند رمز اطلاعات کد شده با روش DES را بشکند. روش AES که نسخه پیشرفته تری برای کد کردن اطلاعات است، بر اساس طول دیتای ۱۲۸ بیت و طول کلید ۱۲۸ یا ۱۹۲ یا ۲۵۶ بیتی، عمل می کند. در روش AES حالت‌های ممکن برای بدست آوردن اطلاعات اصلی به قدری زیاد است که دستیابی به اطلاعات را با امکانات فعلی و در زمان محدود، به مرزهای غیرممکن نزدیک می کند. روش اجرای عملیات AES در XMEGA به این صورت است که بعد از انجام تنظیمات لازم و پرکردن حافظه از دیتا و کد کلیدمورد نظر، بیت مربوط به شروع انجام عملیات در رجیستر مربوطه یک می شود و با پایان Encryption یا Decryption، وقفه مربوط به آن اتفاق می افتد و حاصل عملیات در حافظه قرار می گیرد. امکان انجام عملیات DES و AES بیشتر کاربرد نظامی دارد و از این نظر می تواند در ابزارهای نظامی و مخابراتی به کار گرفته شود.

مقایسه کننده آنالوگ

قبل از توضیح راجع به تفاوت مقایسه کننده آنالوگ در خانواده XMEGA با AVR های معمولی، ذکر این توضیح لازم است که استفاده از این مقایسه کننده ها برای تصمیم گیری های بسیار سریع روی ورودی های آنالوگ به دلیل سرعت بالای مقایسه کننده ها در تشخیص وضعیت ورودی، در کاربردهایی مانند درایورهای قدرت و مدارات کنترل جریان اهمیت زیادی دارد و در IC های [switching regulator](#) و کنترل موتور برای فیدبک گرفتن از ولتاژ یا جریان معمولاً از عملکرد مقایسه کننده های آنالوگ استفاده می شود. مقایسه کننده های آنالوگ در AVR های معمولی به تعداد یک عدد موجود است (به غیر از برخی شماره های بسیار خاص مانند automotive ATMEGA64M1). ورودی مثبت این مقایسه کننده در AVR می تواند از یک پین مشخص ورودی یا یک ولتاژ مرجع داخلی و ورودی منفی هم از یک پین مشخص ورودی یا پین های متصل به ADC فرمان بگیرد. خروجی این مقایسه کننده هم به غیر از تولید وقفه می تواند به عنوان فرمان Capture مربوط به TIMER1 بکار رود.

اما در خانواده XMEGA حداقل دو و حداکثر ۴ مقایسه کننده آنالوگ وجود دارد و امکانات به مراتب قوی تری نسبت به عملکرد مقایسه کننده های این خانواده به شرح زیر اضافه شده است.

• ورودی های متنوع

چندین امکان مختلف برای اتصال به ورودی های مقایسه کننده ها، قابل انتخاب است. در همینجا باید به نکته ای اشاره شود که متاسفانه متون و **document های ATME**L، مملو از اشتباهات چاپی و مفهومی مختلف هستند که می توانند شخص مطالعه کننده را به اشتباه بیاندازند. در مراجعه به متن طولانی و مفصلی که از طرف ATME L برای سری XMEGA A نوشته شده و رجوع به رجیسترهای داخلی این خانواده، مشاهده می شود که برای ورودی مثبت، امکان اتصال به چندین پین ورودی مختلف و خروجی DAC تعریف شده است.

برای ورودی منفی هم امکان اتصال به چند پین ورودی، خروجی DAC، ولتاژ مرجع داخلی و مقدار مقیاس شده Vcc آورده شده است. اما در Application note مربوط به مقایسه کننده داخلی در همین مورد با جمله زیر مواجه می شویم:

positive input pin Only the DAC output is available on the

و این جمله صحیح نیست. از این دست اشکالات در متون ATME L زیاد مشاهده می شود و می توان موارد زیادی را از آن ذکر کرد و این مورد اخیر متاسفانه اشتباه تایپی یا چاپی نیست و یک اشتباه مفهومی است.

• امکان تعریف باند هیستریزس مقایسه کننده

عرض باند هیستریزس مقایسه کننده های داخلی، در سه وضعیت بدون هیستریزس، هیستریزس کم و هیستریزس زیاد قابل تعریف است. عرض باند در حالت اول برابر صفر، در حالت کم حدود ۲۰ یا ۲۵ میلی ولت و در حالت زیاد حدود ۴۰ یا ۵۰ میلی ولت است.

• امکان تعریف تاخیر مقایسه کننده

سرعت عملکرد مقایسه کننده هم در دو وضعیت قابل تعریف است که زمان تاخیر مقایسه کننده را تغییر می دهد. نکته قابل توجه این است که تاخیر مقایسه کننده در خانواده سری A1 و در سرعت بالا حدود ۷۰ نانو ثانیه است. این در حالی است که همین تاخیر برای ATMEGA64 حدود ۵۰۰ نانو ثانیه است و این اعداد، تفاوت سرعت عملکرد XMEGA را در کاربردهایی مانند کنترل جریان یک موتور و مواردی از این دست به خوبی نشان می دهد.

• mode Window

یک امکان ویژه برای پیاده سازی وقفه در محدوده های مختلف ورودی است. در این حالت باید دو مقایسه کننده بکار گرفته شوند و ورودی های مثبت آنها به یکدیگر و به ولتاژ مورد سنجش متصل شوند. دو ورودی منفی هم به ولتاژهای محدوده بالا و پائین Window مورد تعریف، وصل می شوند. با فعال سازی و تنظیم این وضعیت می توان در یکی از ۴ وضعیت بالاتر از محدوده بالایی پنجره (window)، بین دو محدوده ،

خارج از دو محدوده و پائین تر از حد پائین، وقفه ایجاد کرد. همچنین بدون وقفه هم امکان خواندن این وضعیت از رجیسترهای داخلی وجود دارد.

• خروجی مقایسه کننده آنالوگ

امکان فعال سازی پین خروجی مقایسه کننده شماره صفر برای دسترسی بیرونی به وضعیت مقایسه کننده آنالوگ توسط یک بیت رجیسترهای داخلی، فراهم است که وضعیت مقایسه کننده را برای استفاده های خارجی در دسترس قرار می دهد.

علاوه بر همه این موارد، خروجی مقایسه کننده آنالوگ می تواند به عنوان یک **Event** در نظر گرفته شود که با وقوع آن عملیاتی در یک بخش داخلی دیگر و بدون دخالت **CPU** به انجام برسد.

USART و IRCOM

در خانواده **XMEGA** حداکثر ۸ عدد **USART** وجود دارد که در مقایسه با **AVR** های معمولی که اکثراً از یک و حداکثر دو **USART** بهره می برند، تعداد بسیار بالاتری می باشد. از نظر دقت **Baud Rate** هم امکانی در **XMEGA** وجود دارد که **Baud Rate** های بسیار متنوع تری را می تواند در ارتباط تامین کند و طیف گسترده تری را نسبت به **AVR** های معمولی شامل می شود.

USART در خانواده **XMEGA** همچنین امکان ارتباط بصورت **Master SPI** را دارد و این امکان به غیر از سخت افزار مستقلی است که برای ارتباط **SPI** در این خانواده وجود دارد. یکی از امکانات جالب **XMEGA** که در ارتباط با **USART** قرار گرفته، **(IRCOM) (InfraRed Communication Module)** یا ماژول ارتباط مادون قرمز است که سخت افزار لازم برای ارتباط با پروتکل **4.IrDA 1** تا نرخ ارتباط ۱۱۵۲۰۰ بیت در ثانیه را تامین می کند. این واحد سخت افزاری به همراه **USART** مورد نظر، پالس های لازم برای ارسال مادون قرمز را تولید می کند و پالس های دریافتی را هم به اطلاعات مناسب برای اعمال به ورودی **USART**، تبدیل می کند.

DMA هم می تواند با **USART** ارتباط برقرار کند و تبادل اطلاعات را به صورت مستقیم به عهده بگیرد که این امکان در راستای کم شدن بار **CPU** و بالا بردن سرعت کلی اجرای برنامه پیش بینی شده است.

نتیجه یک آزمایش عملی در مورد بخش **CLOCK**

هرچند عملکرد سخت افزارهای مختلف **XMEGA** تا فرکانس ۳۲ مگاهرتز گارانتی شده است. اما از طریق **PLL** داخلی می توان فرکانس کلاک را بالاتر برد و به اصطلاح **CPU** را **over clock** کرد. برای حداکثر مجاز خروجی این **PLL** در یکی از توضیحات **datasheet** عدد ۲۰۰ مگاهرتز (!) ذکر شده است. در یک آزمایش عملی با تنظیم **PLL** مشاهده شد که نرم افزار مورد نظر تا فرکانس ۶۴ مگاهرتز، یعنی ۲۰۰ درصد **over clock** جواب می داد و این نتیجه برای برخی دوستان که مشکل سرعت در مورد کاربردهایی نظیر نمایش تصویر و مانند آن را دارند، احتمالاً می تواند نتیجه امیدوار کننده ای محسوب شود. البته در همین

شرایط مشاهده شد که در بعضی از مقادیر prescaler، خروجی TIMER مورد نظر از کار می افتد و فقط در بعضی شرایط کار می کند. اما هدف از ذکر نتیجه این آزمایش عملی این است که نشان داده شود که فرکانس ۳۲ مگاهرتز لزوماً حداکثر توان پردازشی XMEGA نیست.

RTC

واحد (RTC) Real Time Counter شامل یک شمارنده ۱۶ بیتی است که از یک کلاک مستقل فرمان می گیرد و مقدار حداکثر آن هم قابل تعیین است. از طریق یک Compare register به طول ۱۶ بیت هم امکان تعریف مقدار مشخصی وجود دارد که در صورت برابر شدن RTC با آن یک وقفه تولید شود. منابع کلاک برای RTC می تواند از اسیلاتور ۳۲.۷۶۸ کیلو هرتز خارجی یا دو اسیلاتور ۳۲.۷۶۸ کیلو هرتز یا ۳۲ کیلو هرتز داخلی تامین شود که اسیلاتور خارجی بصورت مستقیم یا تقسیم شده بر ۳۲ و اسیلاتورهای داخلی تنها بصورت تقسیم بر ۳۲ به عنوان منبع کلاک RTC قابل انتخاب است. در ورودی RTC هم یک prescaler قابل برنامه ریزی ۱۰ بیتی وجود دارد که می تواند تا ضریب تقسیم ۱۰۲۴ را برای تقسیم کلاک ورودی انتخاب کند. بنابراین در حداکثر ضریب تقسیم کلاک ورودی، RTC قابلیت تولید زمانی بیش از ۱۸ ساعت را دارد. امکان قرار گرفتن در EVENT SYSTEM هم برای RTC وجود دارد و از سرریز یا شرایط Compare آن می توان به عنوان یک Event استفاده کرد.

Battery backup system و RTC32

ک RTC به طول ۳۲ بیت است که مقدار حداکثر آن قابل تعیین و دارای یک Compare register به طول ۳۲ بیت است. کلاک این واحد فقط از اسیلاتور ۳۲.۷۶۸ کیلو هرتز خارجی تامین می شود و این امکان وجود دارد که یک ضریب تقسیم ۳۲۷۶۸ یا ۳۲ به این کلاک اعمال شود. بنابراین دو کلاک با مقادیر ۱ و ۱۰۲۴ هرتز برای RTC32 قابل انتخاب هستند. با یک محاسبه سرانگشتی می توان حساب کرد که RTC32 به ازای کلاک ۱ هرتز قادر است زمانی بیش از ۱۳۶ سال را ایجاد کند.

یک توانایی خاص موجود در برخی شماره های XMEGA، وجود پایه VBAT برای اتصال یک Battery backup به IC برای استفاده در مواقعی است که تغذیه اصلی IC قطع می شود. در این شرایط بصورت خودکار تغذیه RTC32 و اسیلاتور ۳۲۷۶۸ خارجی از طریق این باتری خارجی تامین می شود و عملیات زمان گیری تنظیم شده برای آن به شکل صحیحی به کار خود ادامه می دهد.

ATXMEGA256A3B یکی از این نمونه هاست که تا این تاریخ از طرف شرکت ATMEL معرفی شده است. در این شماره به جای PF5 که در ATXMEGA256A3 وجود دارد، ورودی VBAT پیش بینی شده و باتری مورد نظر به این پایه متصل و در زمان مقتضی از آن استفاده می شود و بعد از برگشت تغذیه اصلی مجدداً توان از طریق ولتاژ تغذیه تامین خواهد شد. مراجعه به اطلاعات IC مزبور نشان می دهد که مصرف جریان کشیده شده از باتری در هنگام قطع تغذیه اصلی تنها در حد ۰.۵ میکروآمپر است و

این جریان بسیار پائین می تواند استفاده از باتری را جهت تغذیه IC و حفظ عملیات زمان گیری، برای مدتی طولانی تضمین کند.

وقفه های خارجی

برای تولید وقفه های خارجی در خانواده AVR پین های مشخصی در هر شماره وجود دارند که از طریق اعمال ورودی مناسب به آنها، وقفه های خارجی تولید می شوند. این ورودی ها از نظر تعداد کاملاً محدود هستند، چنانکه به عنوان مثال در mega32 دو ورودی و در mega128 حداکثر ۸ ورودی به عنوان ورودی های وقفه خارجی وجود دارند. البته در برخی شماره ها مانند tiny13 یک مکانیزم Pin change interrupt هم وجود دارد که در شرایط تغییر وضعیت پین های یک پورت می تواند وقفه مخصوص به خود را ایجاد کند.

اما پین های تمام پورت ها در خانواده XMEGA قادر به ایجاد وقفه خارجی هستند. برای هر پورت دو وقفه مجزا وجود دارد که تمام پین های پورت این قابلیت را دارند که به عنوان منبع تولید این وقفه ها برنامه ریزی شوند. پین شماره ۲ از هر پورت ویژگی خاص تری برای تولید وقفه دارد که برای جزئیات این مطلب باید به datasheet های مربوطه مراجعه شود. یکی از قابلیت هایی که در همین مورد وجود دارد، امکان فعال سازی وضعیت O/Inverted I برای هر ورودی و خروجی است که به مفهوم فعال سازی یک گیت NOT در هر ورودی و خروجی پورت است. با توجه به اینکه وقفه های خارجی برای ۴ وضعیت لبه بالا رونده، لبه پائین رونده، تغییرات و سطح Low قابل برنامه ریزی هستند، با استفاده از امکان NOT کردن ورودی می توان درخواست وقفه برای سطح High را هم به عنوان پنجمین وضعیت بوجود آورد و بنابراین تمام حالت های متصور برای درخواست وقفه خارجی در XMEGA پشتیبانی می شوند.

Resolution Extension High

امکاری برای افزایش ۴ برابر دقت زمانی در شکل موج خروجی تایمر است. با فعال سازی این واحد، ۱۴ بیت بالا از ۱۶ بیت تایمر توسط کلاکی که فرکانس آن حداکثر می تواند با کلاک CPU برابر باشد، تغییر می کنند و ۲ بیت پائین در یک بخش جدا که فرکانس کلاک آن ۴ برابر است تغییر می کنند. برای فهم بهتر عملکرد این بخش به این نکته توجه کنید که اگر تغییرات تایمر با همان کلاک CPU انجام شود، حداکثر تفکیک پذیری زمانی در شکل موج PWM خروجی، در حد دوره تناوب کلاک CPU است.

اما با استفاده از فعال سازی این بخش و به دلیل اینکه کلاک کلی تایمر در عمل ۴ برابر کلاک CPU می شود، بنابراین در حوزه زمان هم یک تفکیک پذیری و دقت ۴ برابر حاصل می شود. یعنی پله های تغییر شکل موج در حوزه زمان می تواند ۴ برابر دقیق تر شوند. این مسئله می تواند برای تولید شکل موج هایی با دقت ۴ برابر از نظر زمانی مورد استفاده قرار بگیرد که برای فعال سازی آن باید تنظیماتی انجام شود و از جمله در

تنظیمات Prescaler داخلی، کلاک ۴ برابر CPU فعال سازی شود. زیرا با مراجعه به رجیسترهای داخلی می توان مشاهده کرد که Prescaler ها بصورت Default در وضعیت تولید کلاک ۴ برابر قرار ندارند.

SPI

در مورد مزیت های واحد SPI در خانواده XMEGA نسبت به AVR های معمولی می توان به مواردی اشاره کرد. از جمله تعداد SPI های موجود که حداکثر آن در XMEGA سری A1 به تعداد ۴ عدد SPI سخت افزاری روی پورت های C و D و E و F وجود دارند و هر یک بصورت مستقل می توانند عمل کنند. Bit Rate قابل تعریف در ۸ مقدار مختلف می تواند قرار داده شود و این در حالی است که در سری mega64 و mega128 این عدد ۷ مقدار مختلف است. ارتباط با DMA هم از مزیت های بارز SPI در XMEGA است که امکان یک ارتباط سریع و با حجم بالا با حافظه را برای ارسال و دریافت اطلاعات و بدون دخالت CPU فراهم می کند. واحد SPI تنها در مد Slave با DMA ارتباط برقرار می کند، اما برای ارتباط با DMA در مد Master می توان از قابلیت عملکرد USART به عنوان یک SPI در مد Master استفاده کرد و در این صورت ارتباط با DMA در این مد هم میسر خواهد بود.

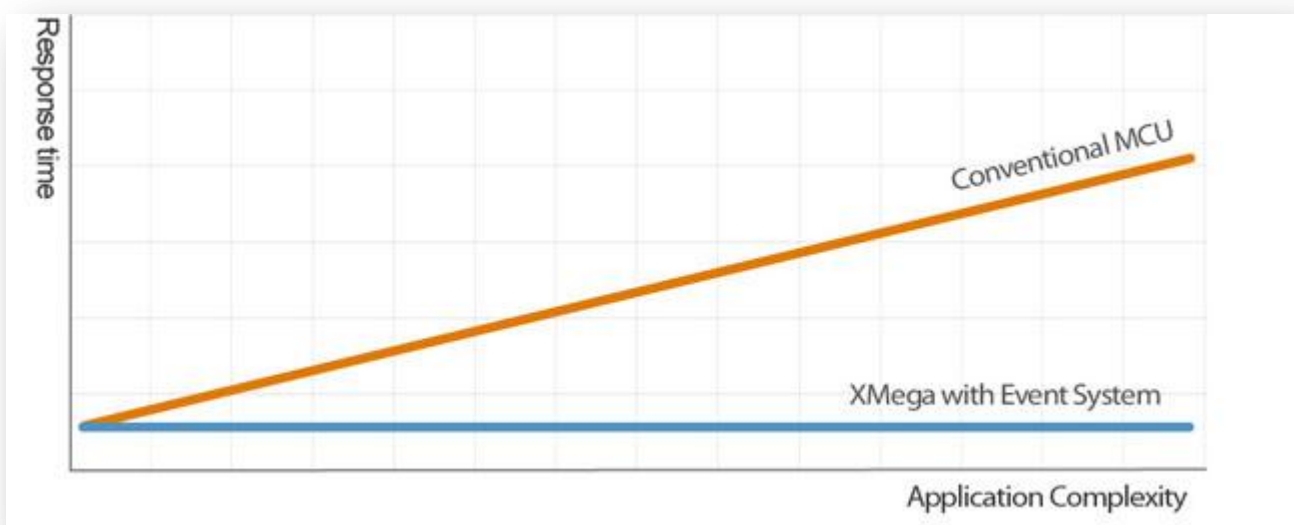
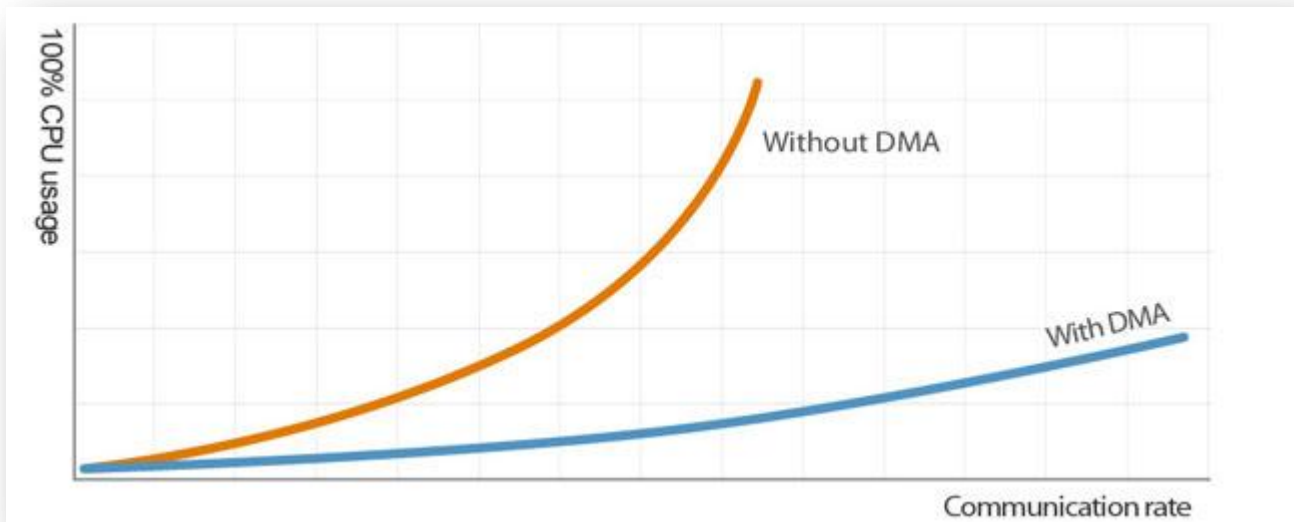
TWI

از نظر تعداد تا ۴ واحد TWI سخت افزاری در سری A1 وجود دارند که روی پورت های C و D و E و F قابل فعال سازی هستند. از نظر عملکرد هم مزیت هایی در خانواده XMEGA وجود دارد که از جمله مهمترین آن پشتیبانی از SMBus علاوه بر پشتیبانی از I2C است. SMBus یا System Management Bus، یک استاندارد ارتباطی است که علیرغم شباهت هایی که با I2C دارد، اما تفاوت هایی هم مانند محدوده ولتاژها، جریان کشی، محدوده فرکانس و پارامتر های زمان بندی بین این دو استاندارد ارتباطی وجود دارد. در [این آدرس](#) توضیحاتی در مورد SMBus و تفاوت های آن با I2C ارائه شده است. در خانواده XMEGA با تنظیم رجیسترهای داخلی، هر دو استاندارد I2C و SMBus برای ارتباط پشتیبانی می شوند.

مزایای XMEGA

دو منحنی مفید برای درک بهتر مزایای XMEGA از سایت atmel آورده شده است. در منحنی اول نشان داده شده که وجود DMA باعث می شود علیرغم افزایش حجم مبادله اطلاعات، CPU از نظر بار پردازشی بسیار دیرتر به مرزهای حداکثر خود نزدیک شود. این مسئله به این دلیل است که وظیفه تبادل اطلاعات از دوش CPU برداشته می شود و به عهده DMA گذاشته می شود و بنابراین برای CPU این فرصت بوجود می آید که به انجام امور دیگر بپردازد.

منحنی دوم تاثیر وجود Event system را نشان می دهد که به علت ارتباط مستقیمی که بین اجزای مختلف و بدون دخالت CPU ایجاد می شود و علیرغم پیچیده تر شدن عملیات انجام شده، زمان انجام آن می تواند افزایش نیابد و بوسیله این امکان سخت افزاری جبران سازی شود. این امکان چیزی فراتر از بالاتر بردن کلاک است و افزایش سرعتی را که در انجام عملیات ایجاد می کند به مراتب مهم تر از افزایش کلاک سیستم است.



محدوده تغذیه

مقدار V_{CC} مجاز برای XMEGA بین ۱.۶ ولت تا ۳.۶ ولت است. اما فرکانس کلاک ۳۲ مگاهرتز تنها از تغذیه ۲.۷ ولت به بالا قابل دستیابی است. در محدوده ۱.۸ ولت تا ۲.۷ ولت حداکثر فرکانس کلاک مجاز بصورت خطی کاهش می یابد و در مقدار $V_{CC} = 8v.1$ این عدد به حداکثر ۱۲ MHz می رسد. در محدوده بین ۱.۶ ولت تا ۱.۸ ولت هم مقدار حداکثر همان ۱۲ MHz باقی می ماند. به دلیل کمتر بودن V_{CC} از مقدار معمول ۵ ولت، محدودیت هایی در مقادیر پورت ها در وضعیت های ورودی و خروجی بوجود می آید. در وضعیت ورودی، حداکثر مقدار مجاز برای اعمال به عنوان ورودی نباید بیشتر از $5v.0 + V_{CC}$ باشد. بنابراین اگر از یک رگولاتور ۳.۳ ولتی برای تغذیه IC استفاده شود، حداکثر مقدار مجاز برابر ۳.۸ ولت است و متصل کردن یک خروجی با مقدار ۵ ولت به ورودی XMEGA، می تواند منجر به آسیب وارد شدن به آن شود. ساده ترین راه برای حل این مشکل استفاده از یک تقسیم مقاومتی و یا در شرایط پیشرفته تر استفاده از IC های Level converter است.

در وضعیت خروجی هم مطابق منحنی های ارائه شده برای XMEGA از طرف کارخانه سازنده و در صورتی که جریانی از خروجی کشیده نشود، مقدار high خروجی پورت برابر V_{CC} و مقدار Low آن برابر صفر خواهد بود. در صورت جریان کشی هم مقدار high کمتر از V_{CC} و مقدار Low بیشتر از صفر خواهد شد (مطابق منحنی های ارائه شده). کمتر بودن مقدار خروجی high از ۵ ولت در اتصال به IC های جانبی، در صورتی که از تغذیه ای مانند ۳.۳ ولت استفاده شود، در اکثر موارد مشکلی بوجود نمی آورد. زیرا اکثر IC هایی که با تغذیه ۵ ولت کار می کنند، ۳.۳ ولت را به عنوان high می شناسند. اما در غیر این صورت و همچنین در صورت استفاده از مقادیر پائین تر V_{CC} ، استفاده از Level converter اجتناب ناپذیر است. (منبع آقای مهندس کی نژاد)

آموزش ۱

در اولین آموزش Xmega به زبان بیسیک ابتدا به سراغ نوع چیپ و تعیین فرکانس کاری مدار می رویم

در این جا نکته ایی که باید یادآور شوم این است که تمامی برنامه ها با نرم افزار بسکام (bascom) ورژن ۲۰۰۱،۰ تست شده و در اختیار دوستان قرار می گیرد که بعضی از این دستورات در ورژن ۱،۱۱،۰۹،۰۸ نیز قابل استفاده است ولی در نسخه ۲۰۰۱،۰ که باید خریداری شود امکانات بسیار زیادی از Xmega تحت پوشش قرار گرفته که در نسخه های قبلی وجود نداشته است.

در برنامه های آموزشی سایت یزد کیت ب میکرو مورد استفاده ATXMEGA128A1 می باشد که در مجموع بهترین ای سی برای آموزش می باشد چون بیشترین امکانات در این ای سی قرار داده شده است.

در ای سی های سری قبلی AVR اگر فرکانس مدار را غیر از ۱ مگاهرتز تعیین می کردید باید علاوه بر نرم افزار در فیوز بیت ها نیز تغییرات لازمه را انجام می دادید و برای افراد مشکلات فراوانی ایجاد می شد اما در سری Xmega این عیب برطرف شده است و با تغییر فرکانس در نرم افزار نیازی به دستکاری در فیوز بیت لازم نمی باشد.

برای ای سی ایکس مگا می توان فرکانس ۲ مگ و ۳۲ مگ و کریستال خارجی و PLL را انتخاب کرد که در ابتدا آموزش از فرکانس ۲ مگاهرتز و ۳۲ مگاهرتز استفاده می کنیم که می توان در خود نرم افزار فرکانس ۳۲ مگ را بر اعداد ۱ و ۲ و ۴ و ۸ و ۱۶ و ۳۲ و ۶۴ و ۱۲۸ و ۲۵۶ و ۵۱۲ تقسیم نمود و فرکانس های مختلف را ایجاد کرد و گزینه های دیگر وجود دارد که کم کم شما را با ان آشنا خواهیم کرد.

دوستان توجه کنند که تمامی دستورات در طول آموزش به طور کامل اما به صورت مقطعی توضیح داده می شود و برای اینکه تمامی افراد بتوانند از آموزش کمال استفاده را ببرند از توضیح بعضی دستورات که کمی سنگین و حرفه ایی تر است در ابتدا جلوگیری می گردد اما در آینده انها را توضیح خواهیم داد.

برنامه نویسی :

در ابتدا برنامه نویسی باید نوع میکرو و فرکانس آن تعیین گردد که در آموزش های ما میکرو ATXMEGA128A1 و فرکانس ۳۲۰۰۰۰۰۰ هرتز می باشد.

```
$regfile = "xm128a1def.dat"  
$crystal = 32000000
```

در قدم دوم باید مقدار فضای لازم برای ذخیره ادرس رجیستر ها در هنگام پرش را برای میکرو تعیین کرد
دراینده مفصل توضیح داده می شود

```
$hwstack = 64  
$swstack = 64  
$framesize = 64
```

سپس باید کتابخانه xmega و فایل های دیگر را در برنامه تعریف نمود.

```
$lib "xmega.lib"  
$external _xmegafix_clear  
$external _xmegafix_rol_r1014
```

حال باید نوع ساخت فرکانس را تعیین نمود . یعنی فرکانس توسط اسیلاتور داخلی ساخته شود یا توسط واحد pll یا توسط کریستال خارجی و گزینه های قابل تنظیم دیگر که فعلا اسیلاتور ۳۲ مگ را فعال می کنیم نکته قابل توجه این است که در این قسمت اسیلاتور فعال شده و میکرو تا پایدار شدن آن صبر می کند و سپس ادامه برنامه را اجرا میکند.

```
Config Osc = Enabled , 32mhzosc = Enabled
```

بعد از فعال کردن اسیلاتور باید مقدار فرکانس را تعیین نمود یعنی میتوان فرکانس ۲ مگ یا ۳۲ مگ و یا فرکانس تقسیم شده از فرکانس ۳۲ مگ را فعال کرد که فعلا فرکانس ۳۲ مگاهرتز را تعیین می کنیم.

```
Config Sysclock = 32mhz , Prescalea = 1 , Prescalebc = 1_1
```

تا این قسمت شما خود میکرو و فرکانس کاری مدار را تعیین نموده اید که مباحث بالا به صورت کامل به شکل زیر می باشد

```
\_____
\               In the name of GOD
\               xm128A1.bas
\               $$$ WWW.yazdkit.com $$$
\_____
```

```
$regfile = "xm128a1def.dat"
$crystal= 32000000
```

```
$hwstack= 64
$swstack= 64
$framesize= 64
```

```
$lib "xmega.lib"
$external _xmegafix_clear
$external _xmegafix_rol_r1014
```

```
Config Osc = Enabled , 32mhzosc = Enabled , 32khzosc = Enabled
```

```
Config Sysclock = 32mhz , Prescalea = 1 , Prescalebc = 1_1
```

در آموزش بعدی نحوه تعیین کردن یک پورت به عنوان خروجی و ورودی را توضیح خواهیم داد و یک برنامه LED چشمک زن را آموزش خواهیم داد.

آموزش ۲

در آموزش [یک](#) با نحوه تعیین میکرو و فرکانس کاری مدار آشنا شده اید.

در این قسمت یک پورت را به عنوان خروجی تعیین می کنیم و کل پایه های پورت را روشن و خاموش می کنیم که اگر به این پایه ها یک led وصل نمایید آن ال ای دی چشمک می زند.

در مجموع در ای سی atxmega128a1 تعداد پورت های آن ۱۱ عدد می باشد که دو پورت آن به صورت ناقص است و تعداد پایه ایی که می توان به عنوان ورودی و خروجی استفاده نمود ۷۸ عدد می باشد.

برای تعیین یک پورت کامل که شامل ۸ پین می باشد از دستور زیر استفاده می کنیم.

برای مثال پورت J به عنوان خروجی تعریف شده است.

```
Config PORTJ = Output
```

با دستور زیر پورت را مقدار می دهیم که مقدار باید بین ۰ تا ۲۵۵ باشد که این مقدار به صورت باینری بر روی خروجی ظاهر می گردد.

مثلا زمانی که مقدار پورت را ۱ می دهیم در خروجی پین $j = 0$ می شود و مابقی پین ها مقدار ۰ را می گیرد.

مقدار ۱ همان ولتاژ ۳٫۳ ولت می باشد و مقدار ۰ همان ولتاژ ۰ ولت می باشد.

```
PORTJ = 0
PORTJ = 1
PORTJ = 255
```

مقداری که به پورت می دهیم می تواند به صورت باینری نیز باشد.

```
PORTJ = &B00000000
PORTJ = &B00000001
PORTJ = &B11111111
```

همچنین می توانیم پورت را به صورت عدد هگز نیز مقدار دهی کنیم.

```
PORTJ = &H00  
PORTJ = &H01  
PORTJ = &HFF
```

برای ایجاد زمان تاخیر در برنامه می توانید از دستورا زیر استفاده کنیم

Wait x – waitms x – waitus x

با دستور **wait** می توان تاخیری در حد ثانیه در برنامه ایجاد کرد و مقدار عددی که در مقابل این دستور قرار می گیرد همان زمان بر حسب ثانیه می باشد . یعنی دستور **wait 12** یک تاخیر ۱۲ ثانیه ایجاد می کند

با دستور **waitms** می توان تاخیری در حد میلی ثانیه در برنامه ایجاد کرد و مقدار عددی که در مقابل این دستور قرار می گیرد همان زمان بر حسب میلی ثانیه می باشد . یعنی دستور **waitms 15** یک تاخیر ۱۵ میلی ثانیه ایجاد می کند و حداکثر مقدار آن از ۱ تا ۶۵۵۳۵ می باشد.

با دستور **waitus** می توان تاخیری در حد میکروثانیه در برنامه ایجاد کرد و مقدار عددی که در مقابل این دستور قرار می گیرد همان زمان بر حسب میکرو ثانیه می باشد . یعنی دستور **waitms 120** یک تاخیر ۱۲۰ میکرو ثانیه ایجاد می کند و حداکثر مقدار آن از ۱ تا ۶۵۵۳۵ می باشد.

```
Wait 12  
Waitms 15  
Waitus 120
```

در برنامه می توانید برچسب های مختلفی را ایجاد کنید و در حین اجرای برنامه به برچسبها پرش نمایید که برچسب باید با حرف شروع شود و می توانید شامل اعداد نیز باشد و در زیر یک برچسب با نام **yazdkit** ایجاد شده است.

Yazdkit :

برای اینکه به برچسب دلخواه پیش نمایید باید از دستور **GOTO** یا **JMP** استفاده نمایید که در نمونه زیر می توان به برچسب **yazdkit** پرش کرد.

```
Goto Yazdkit
```

و حال بر نامه چشمک زن را می توان نوشت که به صورت زیر می باشد.

```
_`_____  
`In the name of GOD  
com $$$yazdkit.`$$$ WWW  
`_____
```

```
dat"."xm128a1def = $regfile  
32000000 = $crystal
```

```
64 = $hwstack  
64 = $swstack  
64 = $framesize
```

```
lib"."xmega $lib  
$external _xmegafix_clear  
$external _xmegafix_rol_r1014
```

```
Enabled = Enabled , 32mhzosc = Config Osc
```

```
= 1 , Prescalebc = 32mhz , Prescalea = Config Sysclock  
1_1
```

```
Output = Portj Config
```

```
:Yazdkit
```

```
0 = Portj
```

```
Waitms 500
```

```
255 = Portj
```

```
Waitms 500
```

آموزش XMEGA - مقدمه

XMEGA سری جدید خانواده AVR است که توسط کمپانی ATMEL عرضه شده است و در عین سازگاری کامل از نظر کدنویسی، دارای توانایی و امکانات بسیار بیشتری نسبت به گروه های S90، Tiny و Mega می باشد. این سری از محصولات جدید با دارا بودن امکاناتی بسیار قوی حتی رقیب قدرتمندی برای میکروکنترلرهای ARM7 محسوب می شود و به همین دلیل قابلیت کاربرد در تولیدات مختلف صنعتی را در سطوح مختلف دارا می باشد.

وجه مشخصه اصلی این خانواده در چند مورد خلاصه می شود:

- ۱- سرعت بالاتر در انجام عملیات که در درجه اول ناشی از حداکثر کلاک قابل اعمال به CPU و سخت افزارهای جانبی است. فرکانس کلاک در این خانواده حداقل ۳۲MHz است که در عمل با Overclock به مقادیر بیشتری هم می توان رسید. عامل دوم وجود امکاناتی مانند Event system و DMA است که راندمان نرم افزار را در یک کلاک برابر به میزان قابل توجهی افزایش می دهند و سبب کاهش بار CPU برای انجام بسیاری از عملیات می شوند.
 - ۲- سخت افزارهای جانبی بسیار غنی مانند ۸ عدد USART و ۴ عدد TWI و ۴ عدد SPI و ADC و DAC با دقت ۱۲ بیت و واحد های رمزنگاری AES و DES و مقایسه کننده های آنالوگ و امکان اتصال به SDRAM خارجی و ۸ عدد تایمر ۱۶ بیتی با ۲۴ خروجی PWM و موارد متعدد دیگری که برای این خانواده اهمیت خاصی را ایجاد کرده است.
 - ۳- مصرف توان بسیار پائین که استفاده از XMEGA را در کاربردهایی که مصرف توان در آن مهم است، کاملاً توجیه پذیر می کند. در یکی از شماره های این خانواده امکان اتصال یک Battery backup خارجی وجود دارد و با قطع تغذیه واحد RTC32 داخلی همچنان به عملیات زمانگیری خود ادامه می دهد.
- بنابراین با توجه به این مزیت ها و شباهت هایی که در عملکرد XMEGA با خانواده های قبلی وجود دارد، بسط و گسترش اطلاعات این میکروکنترلر جدید از اهمیت خاصی برخوردار می باشد.

در ادامه این مباحث به آموزش مواردی از امکانات این میکروکنترلر پرداخته خواهد شد و نمونه کدهایی هم متناسب با هر بخش در اختیار علاقه مندان قرار خواهد گرفت. لازم به ذکر است که این نمونه کدها منطبق با سخت افزار برد آموزشی ارائه شده توسط شرکت نوشته می شوند و روی این برد از نظر عملی قابل اجرا و پیاده سازی خواهند بود.

آموزش XMEGA - حرکت از MEGA به XMEGA

در این بخش به این مسئله پرداخته می شود که چه راهکارهایی برای کسانی که با AVR آشنایی قبلی دارند، برای کار با این خانواده جدید وجود دارد.

از منظر سخت افزار باید به وجوه مشترک و تفاوت های این دو خانواده توجه شود. در عین وجود شباهت هایی در برخی واحدهای سخت افزاری داخلی (مثلا SPI)، تفاوت های اساسی هم در بسیاری از بخش ها از نظر تعداد و عملکرد واحدها وجود دارد. از آنجایی که خانواده XMEGA در واقع نسخه توسعه یافته سری های قبلی است، علاوه بر آنکه وظایف سخت افزارهای قبلی را انجام می دهد، به امکانات جدیدتری هم مجهز شده است. بنابراین نمی توان به اطلاعات قبلی در مورد خانواده AVR بسنده کرد و این اطلاعات باید با موارد جدیدتری تکمیل شود.

به لحاظ برنامه نویسی حداقل ۵ گزینه شامل IAR، GCC-AVR، Codevision، BASCOM و برنامه نویسی به زبان اسمبلی برای هر دو خانواده مطرح است. در مورد آخر به دلیل اشتراک کامل مجموعه دستورات دو خانواده (به غیر از یک دستور)، برنامه نویسی عملاً هیچ تفاوتی بین دو خانواده ندارد (از نظر زبان برنامه نویسی) و یک برنامه نویس اسمبلی برای AVR در واقع بدون هیچ زحمتی می تواند فوراً برای XMEGA هم شروع به برنامه نویسی کند و فقط تفاوت های رجیسترها و حافظه و مانند آن باید مد نظر قرار بگیرد.

در مورد زبان های سطح بالا علیرغم تمایل عمومی که نسبت به کامپایلرهای BASCOM و Codevision (به دلیل ساده تر بودن) وجود دارد، اما برای به فعلیت رساندن حداکثر قابلیت های XMEGA باید از دو گزینه GCC-AVR و IAR استفاده شود. این دو کامپایلر از نظر نوع برنامه نویسی بسیار شبیه هم هستند و حتی در کدهایی که از طرف شرکت Atmel نوشته شده در بسیاری موارد هردو کامپایلر پشتیبانی می شوند. از موارد قابل ذکر این دو کامپایلر این است که از نظر کتابخانه های مختلف بصورت داخلی چندان غنی نیستند، اما کتابخانه های متعددی توسط افراد مختلف نوشته شده که می توان برای این دو کامپایلر از آنها استفاده کرد. در همین رابطه کدهای زیادی برای راه اندازی بخش های مختلف XMEGA توسط شرکت

Atmel نوشته شده و استفاده از این دو کامپایلر، کار راه اندازی XMEGA را با استفاده از این کتابخانه ها و مثال ها بسیار ساده و سریع می کند. از جمله مواردی که در برنامه های آموزشی شرکت نوین تراشه در نظر گرفته شده، استفاده بهینه از همین کتابخانه ها و توابع آماده در مورد XMEGA است که کار بسیار فنی و دقیقی است و نیازمند یک بررسی همه جانبه نسبت به امکانات این خانواده و همچنین کامپایلرهای مذکور است.

آموزش XMEGA - آشنایی با سری A و D

خانواده XMEGA تا این زمان در دو سری کلی A و D عرضه شده اند. سه زیر گروه A1 و A3 و A4 برای سری A و دو زیرگروه D3 و D4 برای سری D وجود دارند. به طور کلی امکانات سری A از سری D قوی تر است و در هر گروه هم شماره های با عدد انتهای کوچکتر دارای امکانات بیشتری هستند. در متن شماره هر IC مانند سری های Mega مقدار Flash آن شماره ذکر می شود. با این توضیحات ATXMEGA384A1 به عنوان قوی ترین شماره این خانواده شناخته می شوند. از نظر Package این شماره ها در ۳ نسخه ۴۴ و ۶۴ و ۱۰۰ پایه عرضه می شوند. سری های D4 و A4 دارای ۴۴ پایه و D3 و A3 دارای ۶۴ پایه و سری A1 دارای ۱۰۰ پایه هستند. برای این خانواده نسخه DIP وجود ندارد و همگی بصورت SMD هستند. برای انتخاب هر شماره باید با توجه به امکانات داخلی IC و نیازهای طراحی و موجود بودن در بازار اقدام شود و قبل از انتخاب یک شماره نسبت به واحدهای سخت افزاری موجود در آن بررسی کاملی به عمل آید.

به عنوان مثال در شماره های سری D اصولاً DMA و DAC وجود ندارد و عدم توجه به این مسئله می تواند انجام یک منظور از پیش تعیین شده را که مستلزم استفاده از این امکانات است با مشکل مواجه کند.

آموزش XMEGA - تغذیه

مقدار ولتاژ تغذیه در خانواده XMEGA متفاوت از ۵ ولت معمول برای AVR است. ناحیه کار برای تغذیه این خانواده در محدوده ۱.۶ تا ۳.۶ ولت است. اما این بدان معنی نیست که حداکثر سرعت عملکرد IC در کل این ناحیه قابل حصول باشد. حداکثر فرکانس کاری در تغذیه ای بزرگتر از ۲.۷ ولت قابل حصول است و در ولتاژهای تغذیه کمتر از این مقدار، مقدار فرکانس مجاز کاهش می یابد. هر چند بنا به منابع Atmel این فرکانس با کاهش تغذیه بصورت خطی کاهش می یابد و در تغذیه ۱.۸ ولت به ۱۲ مگاهرتز می رسد، اما برای شماره های مختلف XMEGA لازم است حتماً به بخش Errata در انتهای Datasheet مراجعه شود تا اگر استثنایی وجود دارد از آن غفلت نشود. به عنوان مثال برای ATXMEGA256A3 ذکر شده که در تغذیه های کمتر از ۱.۷۶ ولت، حداکثر فرکانس به ۸ مگاهرتز محدود می شود و این یک استثناء نسبت به

بقیه موارد است. برای تغذیه XMEGA معمولا از رگولاتورهای ۳.۳ ولتی مانند LF33 و شماره های نظیر آن استفاده می شود. چنین تغذیه ای اتصال XMEGA به وسایل جانبی مانند MMC یا پورت USB را بصورت مستقیم میسر می کند و برای اتصال به وسایل جانبی با تغذیه ۵ ولت هم باید تمهیداتی اندیشیده شود تا احتمالا آسیبی به ورودی های XMEGA وارد نشود.

آموزش XMEGA - پروگرامر

یکی از سوالاتی که در ابتدای کار برای افراد علاقه مند به XMEGA مطرح می شود، شیوه برنامه ریزی این IC است. برنامه ریزی XMEGA از طریق دو پورت PDI و JTAG انجام می شود که JTAG هم در واقع بصورت داخلی با PDI ارتباط برقرار می کند. یکی از بهترین انواع پروگرامر برای این خانواده، AVRISPMKII از محصولات شرکت Atmel است که با اتصال به پورت USB امکان برنامه ریزی خانواده XMEGA را با سرعت بسیار بالا از طریق پورت PDI فراهم می کند. این پروگرامر همچنین قادر است تمام شماره های خانواده AVR را برنامه ریزی کند. از خواص مهم این پروگرامر امکان به روز سازی و Update توسط نسخه های جدیدتر نرم افزار AVRStudio است و به این ترتیب از شماره های جدیدی که در آینده عرضه می شود، پشتیبانی خواهد کرد. نوع دیگری از محصولات Atmel پروگرامر JTAGICEMKII نام دارد که از طریق پورت JTAG با XMEGA ارتباط برقرار می کند. از طریق این پروگرامر می توان عملیات Debugging را در حین اجرای نرم افزار پیاده سازی کرد و به مقادیر داخلی CPU و وضعیت سخت افزارهای داخلی دسترسی پیدا کرد. از نظر قیمت این پروگرامر به میزان قابل توجهی دارای قیمت بالاتری نسبت به نوع قبلی می باشد. به غیر از این دو مورد چند نوع پروگرامر دیگر هم برای برنامه ریزی XMEGA وجود دارد، که بهترین گزینه موجود از نظر قیمت مناسب و در عین حال توانمندی بالا همان AVRISPMKII است.

لازم به ذکر است که برای سهولت استفاده از XMEGA، تعداد محدودی AVRISPMKII تهیه شده تا در اختیار علاقه مندان به کار با این IC جدید قرار بگیرد و کمبودهایی که در بازار برای عرضه این دستگاه بوجود می آید، مانعی را در راه یادگیری و کار عملی با این خانواده جدید ایجاد نکند.

آموزش XMEGA - از کجا شروع کنیم؟

توانایی های XMEGA موجب ایجاد انگیزه زیادی برای آشنایی و تسلط بر این خانواده جدید می شود. اما برای افرادی که به این سمت گرایش دارند، این سوال مطرح می شود که باید از کجا شروع کنند و چه روش

ها و ابزارهایی آنها را سریع تر به مقصد نزدیک می کند. برای پاسخ به این سوال باید به مواردی چند اشاره شود. مورد اول لزوم آشنایی قبلی با خانواده AVR و عملکرد و معماری آن است که کمک شایانی در سرعت پیشرفت کار با XMEGA می نماید. به عبارت دیگر یک پیش نیاز تقریباً لازم برای آشنایی با XMEGA، آشنایی قبلی با AVR های سری قبل از XMEGA است. این آشنایی می تواند راه را برای استفاده از امکانات پیشرفته تر XMEGA هموارتر کند و برای کسانی که سابقه آشنایی با میکروکنترلر ها را ندارند، در ابتدای امر کار با XMEGA توصیه نمی شود.

مسئله بعدی توجه به ابزار های نرم افزاری و سخت افزاری لازم برای شروع به کار با این خانواده است. ۳ ابزار مشخص برای کار با XMEGA باید فراهم باشد:

۱- کامپایلر مورد استفاده به عنوان ابزار برنامه نویسی

۲- سخت افزاری که بتوان به عنوان یک بستر عملیاتی، نرم افزار های نوشته شده را روی آن آزمایش کرد.

۳- پروگرامری که بتوان نرم افزار نوشته شده را به سخت افزار منتقل و روی آن برنامه ریزی کرد.

در مورد اول در بخش "حرکت از MEGA به XMEGA" توضیحاتی داده شده است. به عنوان یک توضیح تکمیلی ذکر این نکته لازم است که برای استفاده از حداکثر ظرفیت میکروکنترلرها لازم است در برخی شرایط توانمندی کار با ریزترین جزئیات رجیسترهای سخت افزاری در اختیار برنامه نویس باشد و استفاده مکرر از Wizard ها یا روش هایی که همه بار درگیری با سخت افزار را به عهده کامپایلر قرار می دهد، روش چندان کارایی نیست و نوعی تنبلی ذهنی و ناتوانی برنامه نویس را به دنبال دارد. بر همین اساس مجموعه ای از برنامه نویسی C و اسمبلی به عنوان کاراترین شکل برنامه نویسی مطرح است و برای XMEGA هم همین شیوه توصیه می گردد.

مسئله بعدی سخت افزار مورد آزمایش است که باید با توجه به پتانسیل های شماره IC مورد نظر طراحی شود. یکی از اشتباهاتی که توسط افراد تازه کار اتفاق می افتد این است که بدون تسلط به جوانب مختلف یک میکروکنترلر جدید، اقدام به تلاش برای ساخت سخت افزار می کنند و وقت خود را بجای پیشرفت در اصل مسئله در حاشیه های آن مصرف می کنند. واقعیت آن است که سری XMEGA به دلیل ابعاد و پیچیدگی های مختلفی که دارد، به اندازه کافی نکته برای بررسی دارد و درگیر شدن در چند جبهه برای کسی که تجربه کافی ندارد، کار چندان معقولی به نظر نمی رسد. البته در صورتی که این توانایی وجود داشته باشد که سخت افزار لازم در همان مرحله اول با ضریب اطمینان بالا و بدون خطا ساخته شود، امری کاملاً پذیرفتنی است. اما عامل پیشتاز بودن و جلو افتادن در زمان هم مسئله مهمی است که نباید از آن غافل شد.

در مورد پروگرامر هم قبلاً توضیح داده شده و بدیهی است که وجود آن برای امکان هرگونه آزمایش عملی امری بدیهی است.

بعد از فراهم شدن این ابزارها با توجه به مبانی تئوری و منابع مکتوب، امکان شروع کار با XMEGA میسر می شود و آنچه که باقی می ماند، حرکت در مسیری است که بر اساس آن بتوان پتانسیل های XMEGA را بصورت عملی روی این بستر سخت افزاری و نرم افزاری پیاده سازی نمود.

آموزش XMEGA - منابع Clock (بخش اول)

منابع تامین پالس ساعت برای CPU و سایر سخت افزارها در این خانواده متعدد هستند. ۵ منبع شامل اسیلاتور ۲ مگاهرتز داخلی، اسیلاتور ۳۲ مگاهرتز داخلی، اسیلاتور ۳۲.۷۶۸ کیلوهرتز داخلی، خروجی PLL و اسیلاتورها یا کریستال های خارجی به عنوان منابع تامین کلاک قابل انتخاب هستند. حداکثر کلاک نامی برای این خانواده از نظر اعمال به CPU در حد ۳۲ مگاهرتز است که تا سرعت اجرای حداکثر ۳۲ میلیون دستورالعمل در ثانیه را تامین می کند. اما در عمل با افزایش کلاک می توان به مقادیر بیشتر هم رسید که به این عمل Clock Over گفته می شود. بر خلاف اکثر شماره های AVR که تغییر منبع کلاک نیازمند برنامه ریزی فیوزبیتهاست، این امر در XMEGA توسط خطوط برنامه انجام می شود و در هر زمان که لازم باشد می توان منبع کلاک را تغییر داد. برای واحد های سخت افزاری داخلی می توان کلاکی متفاوت از کلاک CPU تعریف کرد و این امر از طریق تقسیم کننده های قابل برنامه ریزی که به همین منظور پیش بینی شده اند، میسر است. دو واحد سخت افزاری در XMEGA وجود دارند که یکی تا ۶۴ مگاهرتز و دیگری تا ۱۲۸ مگاهرتز کلاک را می پذیرند. برای استفاده از این ظرفیت باید ابتدا با PLL داخلی یک فرکانس ۱۲۸ مگاهرتز تولید کرد و با تنظیمات لازم دوبار آن را بر دو تقسیم کرد. حال این ۳ فرکانس متمایز یعنی ۱۲۸ و ۶۴ و ۳۲ مگاهرتز هر یک به قسمت مربوط به خود اعمال می شود که ۳۲ مگاهرتز مربوط به CPU است.

باید به این نکته توجه شود که بالاتر بودن کلاک به خودی خود ملاک کاملی برای سرعت XMEGA نیست و وجود واحدهایی مانند Event System و DMA می تواند سرعت اجرای عملیات را به ازای یک کلاک ثابت بسیار بالاتر ببرد و این مسئله ای است که باید در کنار فرکانس کلاک بصورت توأم به آن توجه شود.

به دلیل تغذیه حداکثر ۳.۶ ولت برای XMEGA ، حداکثر ولتاژ تولید شده به عنوان خروجی و حداکثر ولتاژ مجاز به عنوان ورودی بوسیله تغذیه محدود می شود و اتصال به وسایل جانبی با تغذیه ۵ ولت باید با رعایت این مورد انجام بپذیرد. نکته بعدی حالت های متنوع تر ورودی و خروجی نسبت به AVR های معمول است. هر پین یک پورت به عنوان ورودی می تواند در ۴ وضعیت و به عنوان خروجی در ۵ وضعیت مختلف از نظر عملکرد قرار داشته باشد.

از نظر ورودی حالت های زیر قابل فعال شدن هستند:

۱ - High impedance

۲ - فعال شدن Pull up

۳ - فعال شدن Pull down

4-Bus keeper که به معنای فعال سازی خودکار Pull up یا Pull down برای حفظ وضعیت پورت متناسب با حالت خروجی آن است.

از نظر خروجی ۵ وضعیت قابل فعال شدن هستند:

1-Totem pole: در این حالت پین خروجی برای هر دو وضعیت High و Low بصورت مناسب درایو می شود.

2-Pull up +Wired AND: این خروجی ها قابل وصل کردن به یکدیگر هستند. خروجی هایی که High هستند تاثیری در خروجی مشترک پین ها ندارند و خروجی هایی که Low هستند خود را به وضعیت پین های خروجی تحمیل می کنند. اگر حتی یک خروجی Low باشد، پین خروجی صفر می شود. Pull up بصورت داخلی فعال است.

3-Wired AND: مانند وضعیت قبل و بدون فعال بودن Pull up داخلی.

4-Pull down +Wired OR: این خروجی ها قابل وصل کردن به یکدیگر هستند. خروجی هایی که Low هستند تاثیری در خروجی مشترک پین ها ندارند و خروجی هایی که High هستند خود را به

وضعیت پین های خروجی تحمیل می کنند. اگر حتی یک خروجی High باشد، پین خروجی یک می شود. Pull down بصورت داخلی فعال است.

5- Wired OR: مانند وضعیت قبل و بدون فعال بودن Pull down داخلی.

مسئله دیگری که در مورد پورت های XMEGA می توان به آن اشاره کرد، امکان تعریف وقفه خارجی روی تمام پین های پورت هاست که در مقایسه با AVR که فقط پین های خاصی به این امر اختصاص پیدا کرده، مزیت مهمی به شمار می رود.

آموزش XMEGA - تایمرها (بخش اول)

برخلاف AVR های عادی که دارای تایمرهای ۸ بیتی و ۱۶ بیتی هستند، تمام تایمرهای XMEGA بصورت ۱۶ بیتی هستند. از نظر تعداد هم مزیت قابل توجهی در تعداد تایمرهای XMEGA وجود دارد. تعداد تایمرها در سری های مختلف به شرح زیر است:

۸ عدد در سری A1

۷ عدد در سری A3

۵ عدد در سری A4 و D3

۴ عدد در سری D4

یکی از نقاط قوت خانواده XMEGA نسبت به اکثر میکروکنترلرهای ۸ و ۱۶ بیتی، تعداد خروجی های زیاد PWM در شماره های مختلف این خانواده است. تعداد خروجی های PWM در سری های مختلف به شرح زیر است:

۲۴ خروجی PWM در سری A1

۲۲ خروجی PWM در سری A3

۱۶ خروجی PWM در سری A4

۱۸ خروجی PWM در سری D3

۱۴ خروجی PWM در سری D4

از امکانات ویژه این تایمرها نسبت به AVR های قبلی، امکان شمارش بصورت نزولی علاوه بر شمارش صعودی است و این امکان توسط خطوط برنامه قابل انتخاب است. این در حالی است که در تایمرهای AVR، به غیر از وضعیت PWM شمارش تنها بصورت صعودی قابل انجام است و کنترلی روی جهت شمارش وجود ندارد.

آموزش XMEGA - تایمرها (بخش دوم)

یکی از امکانات تایمرها در این خانواده، امکان متوالی کردن دو تایمر بصورت سخت افزاری و ایجاد تایمر ۳۲ بیتی است. وجود واحدهای Capture (که بصورت ۳۲ بیتی هم می توانند عمل کنند)، امکان کاربردهایی مانند اندازه گیری فرکانس یا عرض پالس را میسر می کند. این کار بوسیله نمونه برداری از مقدار تایمر در دو لبه یکسان یا متوالی و محاسبه زمان سپری شده، امکان پذیر می شود.

واحدهای Compare برای تولید خروجی های PWM مورد استفاده قرار می گیرند. برخی تایمرها دارای ۴ واحد Compare و برخی دارای ۲ عدد از این واحد سخت افزاری هستند. خروجی های PWM می توانند در کاربردهایی نظیر کنترل دور موتور و UPS و منابع تغذیه و غیره بکار گرفته شوند. نمونه عملی از راه اندازی یک موتور سه فاز با برد آموزشی، بصورت فیلم کوتاهی در بخش دانلود این سایت قابل مشاهده است. در همین راستا واحدی به نام AWeX در XMEGA وجود دارد که در خروجی تایمر فعال می شود و وظایف یک IC درایور موتور از نظر تولید زمان تاخیر بین روشن و خاموش شدن سوئیچ های قدرت و یا پاسخ به شرایط خطا و مانند آن را به عهده می گیرد.

همچنین واحد سخت افزاری دیگری به نام Extension High Resolution در خروجی تایمر قابل فعال سازی است که تا ۱۲۸ مگاهرتز کلاک را می پذیرد و برای ۴ برابر کردن دقت خروجی تایمر در حوزه زمان مورد استفاده قرار می گیرد. بوسیله این واحد حداکثر کلاک موثر اعمال شده به تایمر بجای ۳۲ مگاهرتز تا ۱۲۸ مگاهرتز قابل افزایش است و این امکان می تواند برای ساخت زمان های بسیار دقیق و با پله تغییراتی در حدود ۷ نانوثانیه مورد استفاده قرار بگیرد.

آموزش-XMEGA-مصرف توان

XMEGA یکی از کم مصرف ترین میکروکنترلرهای موجود در دنیا در مقایسه با امکانات داخلی آن است. از آنجایی که بخش های مختلف سخت افزار، قابلیت فعال یا غیر فعال شدن از طریق نرم افزار را دارند، جریان مصرفی هر واحد سخت افزاری بصورت مجزا در اطلاعات هر خانواده ذکر گردیده است. مراجعه به این اطلاعات نشان می دهد که جریان کشی میکروکنترلر در برخی مدهای SLEEP حتی تا حد 100 nA هم کاهش می یابد و در شرایط عادی کار هم میکروکنترلر می تواند با جریان هایی حتی کمتر از یک میلی آمپر به انجام وظایف خود بپردازد. البته با افزایش فرکانس کار و فعال شدن واحد های سخت افزاری مختلف و همچنین بالا رفتن دما، جریان مصرفی بالا تر می رود.

به دلیل مصرف بسیار کم توان، خانواده XMEGA برای استفاده در وسایل قابل حمل که از باتری و منابع محدود انرژی به عنوان منبع تغذیه استفاده می کنند، بسیار مناسب است. در شماره بخصوص ATXMEGA256A3B، امکان اتصال یک Battery Backup به پایه مخصوص به خود وجود دارد و با قطع منبع تغذیه، RTC داخلی با استفاده از این منبع به کار خود ادامه می دهد. بنابراین تاریخ و زمان صحیح با ادامه به کار RTC محفوظ باقی می ماند و از این نظر نیاز به IC های جانبی RTC که معمولاً در کنار میکروکنترلرها قرار می گیرند، برطرف می شود.

آموزش-XMEGA-منابع Clock (بخش دوم)

اسیلاتور داخلی ۲ مگاهرتز: بعد از Reset شدن میکروکنترلر، این اسیلاتور بصورت خودکار به عنوان منبع کلاک سیستم انتخاب می شود. همچنین مضاربی از فرکانس این اسیلاتور از طریق PLL می تواند به عنوان کلاک انتخاب شود. دقت این اسیلاتور در دمای ۲۵ درجه سلسیوس و تغذیه ۳ ولت در حد ۱ درصد تغییرات است.

اسیلاتور داخلی ۳۲ مگاهرتز: برای استفاده از این اسیلاتور، باید ابتدا بوسیله خطوط برنامه عملیات فعال سازی آن انجام شود و در مرحله بعد بصورت مستقیم یا از طریق PLL به عنوان منبع کلاک سیستم انتخاب شود. دقت این اسیلاتور در دمای ۲۵ درجه سلسیوس و تغذیه ۳ ولت در حد ۱ درصد تغییرات است.

اسیلاتور داخلی ۳۲.۷۶۸ کیلوهرتز: این اسیلاتور بعد از فعال سازی می تواند به عنوان کلاک سیستم انتخاب شود و همچنین بعد از تقسیم شدن بر ۳۲ برای اعمال به بخش RTC انتخاب شود. این اسیلاتور می تواند برای مکانیزم کالیبره کردن خودکار اسیلاتورهای ۲ و ۳۲ مگاهرتز داخلی بکار رود. دقت این اسیلاتور در

دمای ۲۵ درجه سلسیوس و تغذیه ۳ ولت در حد ۱ درصد تغییرات است. امکان کالیبره کردن مقدار خروجی این اسیلاتور بصورت نرم افزاری وجود دارد.

منابع کلاک خارجی: قابلیت اتصال کریستال یا رزوناتور و همچنین اعمال کلاک مستقیم از خارج وجود دارد. پین های XTAL1 و XTAL2 قابلیت اتصال کریستالی در محدوده ۰.۴ تا ۱۶ مگاهرتز را دارند و به دو پین TOSC1 و TOSC2 می توان یک کریستال ساعت با فرکانس 32768 Hz را متصل کرد. هر یک از این دو منبع به همراه کلاک خارجی اعمال شده به ورودی XTAL1 می توانند به عنوان منبع کلاک سیستم تعیین شوند.

PLL: برخی از منابع ذکر شده می توانند به عنوان ورودی به PLL تعیین شوند و مضرری از آن به عنوان کلاک سیستم انتخاب شود. اسیلاتور داخلی ۲ مگاهرتز، اسیلاتور داخلی ۳۲ مگاهرتز (تقسیم بر ۴)، کلاک خارجی و خروجی اسیلاتور کریستالی فرکانس بالا (۰.۴-۱۶ مگاهرتز) می توانند به عنوان ورودی PLL تعیین شوند. ضریب افزایش فرکانس PLL در محدوده ۱ تا ۳۱ برابر است. خروجی PLL نباید در فرکانسی کمتر از 10 MHz تنظیم شود و عدد 200 MHz به عنوان حد بالای آن ذکر شده است. برای تغییر ضریب PLL باید ابتدا این واحد غیر فعال شود و بعد از تغییر ضریب، مجدداً فعال شود.

آموزش-XMEGA-وقفه های خارجی

بر خلاف AVR های معمولی که دارای ورودی های بسیار محدودی برای وقفه های خارجی هستند، کلیه پین های هر پورت در XMEGA قابلیت ایجاد وقفه خارجی را دارند. این وقفه ها می توانند در پاسخ به لبه بالارونده، لبه پائین رونده، تغییرات و سطوح Low و High ایجاد شوند که البته برای مورد آخر باید مکانیزم Not موجود در ورودی پورت فعال شود. این وقفه ها می توانند بصورت سنکرون با کلاک داخلی پورت ایجاد شوند. به غیر از وقفه سنکرون، پین شماره ۲ از هر پورت دارای قابلیت ایجاد وقفه بصورت آسنکرون با کلاک داخلی هم هست و این بدان معنی است که حتی در صورت عدم تغییرات در این کلاک یا غیر فعال بودن آن هم وقفه مورد نظر با توجه به وضعیت ورودی ایجاد می شود. سایر پین های پورت هم امکان پذیرش وقفه بصورت آسنکرون را بصورت محدودتری دارند. از جمله کاربردهای وقفه آسنکرون، خارج کردن میکروکنترلر از وضعیت های SLEEP است. در پین هایی به غیر از پین شماره دو از پورت مورد نظر، تغییرات و سطح Low می توانند بصورت آسنکرون عمل کنند و موجب خروج از وضعیت SLEEP شوند. اما شرط ایجاد وقفه، ثابت ماندن وضعیت آنها تا زمان وقوع کلاک بعدی است و در غیر اینصورت وقفه ای ایجاد نخواهد شد. پین شماره ۲ از هر پورت هم حتی در صورت تغییر وضعیت اولیه و در هر شرایطی منجر به ایجاد وقفه خواهد شد.

در هر پورت دو سطح وقفه قابل تعریف است که هر یک از پین ها می توانند به این وقفه ها نسبت داده شوند. هر یک از این دو سطح وقفه می توانند دارای اولویت مخصوص به خود باشند که در مورد اولویت در بخش وقفه ها توضیح بیشتری داده خواهد شد.

آموزش - XMEGA رمزنگاری

در کاربردهایی مانند سیستم های پولی و بانکی یا نظامی، لازم است اطلاعات حساس در هنگام ذخیره سازی و یا ارسال و دریافت، رمزنگاری شوند تا محتویات واقعی اطلاعات از دید ناظران غیرمجاز مخفی بماند. به این عملیات در اصطلاح Encryption گفته می شود. همچنین برای برگرداندن محتویات کد شده اطلاعات به وضعیت اصلی باید روند معکوس روی دیتا انجام شود تا اطلاعات اولیه بازیابی شوند. به عملیات بازیابی اطلاعات Decryption یا رمزگشایی گفته می شود. پروسه های Encryption و Decryption اصولاً عملیات زمانبری هستند و در بسیاری مواقع لازم است که بصورت Online انجام شوند و همانطور که اطلاعات بصورت متوالی ارسال یا دریافت می شود، این عملیات هم روی محتویات اطلاعات انجام پذیرند.

از جمله ویژگی های بسیار مهم خانواده XMEGA، انجام دو الگوریتم رمزنگاری DES و AES است که انجام آن بصورت سخت افزاری در این خانواده پیش بینی شده است. از نظر سطح امنیت، روش AES در مرتبه بسیار بالاتری نسبت به DES قرار دارد و امنیت آن به حدی است که دستیابی به اطلاعات اصلی تقریباً امکان ناپذیر است و حالات ممکن برای برای رسیدن به اطلاعات اصلی به حدی زیاد است که لازم است هزاران پردازنده برای امتحان همه حالت ها بکار گرفته شوند. روش DES هم از سطح امنیت بسیار خوبی برخوردار است و برای رمزنگاری و رمزگشایی با این روش، یک دستور خاص در مجموعه دستورات XMEGA اضافه شده است. وجود این امکان در XMEGA می تواند ویژگی خاصی را برای استفاده در کاربردهای مخابراتی و نظامی برای این خانواده ایجاد کند و از این جهت خاص به بسیاری از میکروکنترلرهای موجود در دنیا برتری دارد.

آموزش - XMEGA اولویت در وقفه ها

مدیریت وقفه در XMEGA بسیار کامل تر از AVR های معمولی است. در خانواده AVR امکان تعریف اولویت برای وقفه ها پیش بینی نشده است و اگر وقفه ای در حال اجرا باشد، کلیه وقفه های دیگر باید منتظر اتمام اجرای وقفه جاری بمانند. البته بصورت نرم افزاری می توان در روتین وقفه جاری، امکان پذیرش وقفه های دیگر را توسط خطوط برنامه فراهم کرد. اما این یک امر زمان بر است و برای کاربردهای بسیار سریع ممکن است جوابگوی نیازهای برنامه نباشد. در AVR تنها یک نوع اولویت بر اساس آدرس بردارهای وقفه وجود دارد و در هنگام درخواست توام تقاضای وقفه توسط چند منبع، اولویت با منبع درخواست کننده ای

است که آدرس بردار وقفه آن کوچکتر باشد. این نوع تعریف اولویت به هیچ وجه پاسخگوی نیازهای برنامه نویسی در کاربردهای سریع نیست و ضعف عمده ای برای AVR محسوب می شود.

در نقطه مقابل ۳ سطح اولویت در XMEGA قابل تعریف است که هر سطح بالاتر می تواند منجر به متوقف کردن روتین وقفه با سطح پائین تر اولویت شود و بعد از انجام روتین وقفه با اولویت بالاتر، مجدداً به وقفه با اولویت پائین تر برگشت می شود و انجام آن تکمیل می شود. همچنین امکان غیر فعال کردن گروهی از وقفه ها با اولویت یکسان وجود دارد و برای هر سطح از اولویت وقفه ها، کنترل جداگانه ای از نظر فعال بودن در نظر گرفته شده است.

به غیر از این ۳ سطح اولویت که توسط نرم افزار قابل تعریف هستند، Non Maskable Interrupt یا NMI به عنوان وقفه ای با بالاترین اولویت در نظر گرفته شده که در حالت خاصی رخ می دهد که اسیلاتور کریستالی یا خارجی به هر دلیلی از کار بیافتد و کلاک سیستم بصورت خودکار از اسیلاتور داخلی ۲ مگاهرتز تامین شود. این سطح از وقفه از بالاترین اولویت برخوردار است و هریک از ۳ سطح وقفه معمول سیستم را متوقف می کند.

بین وقفه های هم اولویت همچنان مسئله تقدم بر اساس آدرس بردار وقفه برقرار است و اگر چنین وقفه هایی با یکدیگر درخواست شوند، اولویت با وقفه ای خواهد بود که از آدرس بردار وقفه کوچکتری برخوردار است. به این نوع اولویت در اصطلاح Static گفته می شود. علاوه بر این در پائین ترین سطح اولویت وقفه ها امکان فعال سازی بررسی اولویت بصورت Round-Robin وجود دارد. در این روش ملاک کمتر بودن آدرس بردار وقفه، به صورت چرخشی تغییر می کند و اگر به وقفه ای توسط CPU پاسخ داده شد، در نوبت بعدی در انتهای صف پاسخ گویی قرار می گیرد. در صورت درخواست مجدد وقفه پاسخ داده شده به همراه یک وقفه دیگر، به وقفه با آدرس بزرگتر ترتیب اثر داده می شود. در این شرایط احتمال اینکه به وقفه ای پاسخ داده نشود از بین می رود و تمامی وقفه های هم اولویت در صف پاسخگویی قرار می گیرند.

آموزش Event System-XMEGA

Event System یکی از امکانات بسیار مهم در خانواده XMEGA است که از طریق آن یک ارتباط سخت افزاری بدون دخالت CPU بین اجزای داخلی میکروکنترلر برقرار می شود. به واسطه وجود این امکان و به دلیل انجام برخی وظایف بصورت سخت افزاری، به همان میزان در اجرای دستورالعمل ها توسط CPU صرفه جویی می شود. این یک توانایی به غیر از افزایش فرکانس کلاک سیستم است که با کاهش دستورالعمل های لازم برای انجام یک عملیات مشخص، می تواند راندمان اجرای نرم افزار را به میزان قابل توجهی افزایش دهد.

برای درک بهتر این قابلیت فرض کنید در یک AVR سری mega مانند mega64 که دارای ۲ تایمر ۱۶ بیتی است، ایجاد یک تایمر ۳۲ بیتی مورد نظر باشد. یک روش برای انجام این کار این است که وقفه سرریز یکی از تایمرهای ۱۶ بیتی فعال شود و در روتین وقفه سرریز آن تایمر، توسط خطوط برنامه روی یکی از پایه های خروجی یک کلاک ایجاد شود و این کلاک به یکی از پایه های T0 یا T1 اعمال شود. T0 یا T1 در این حالت باید به عنوان منبع کلاک تایمر دوم انتخاب شده باشد.

انجام این عملیات مستلزم اجرای دستورالعمل هایی از طرف CPU و صرف زمان است که موجب LOAD شدن آن برای مدت زمانی می گردد. اما در XMEGA با تنظیم صحیح رجیسترهای داخلی، می توان سرریز یک تایمر را به عنوان منبع کلاک تایمر دیگر تعریف کرد و این عملیات بدون هرگونه صرف زمانی از طرف CPU و بصورت کاملاً سخت افزاری انجام شود.

به این ترتیب با خارج کردن CPU از پروسه انجام عملیات می توان به سرعت های بالاتری در اجرای برنامه دست یافت و زمانی که اجرای چندین عملیات بصورت همزمان مورد نظر باشد، این مزیت به میزان قابل توجهی می تواند منجر به بالا رفتن سرعت اجرای برنامه شود.

۸ کانال مختلف در Event System وجود دارد که می توانند به عنوان منبع تریگر بخش های مختلف انتخاب شوند. منابع زیر به عنوان تولید Event قابل تعریف هستند:

- کلاک وسایل جانبی تقسیم بریک ضریب متغیر

- لبه و سطح پورتهای

- ADC

- مقایسه کننده آنالوگ

- RTC

- تایمرها

کلیه Eventها می توانند با فرمان نرم افزاری هم ایجاد شوند. فرمانهای ناشی از منابع Event می توانند توسط بخشهای زیر مورد استفاده قرار بگیرند:

- تایمرها

- ADC

- DAC

- DMA

- واحد ارتباط سریال مادون قرمز (IRCOM)

برای ارتباط بین دو واحد سخت افزاری باید منبع مورد نظر را به یکی از ۸ کانال Event نسبت داد و در مصرف کننده هم کانال مذکور را به عنوان منبع تریگر مورد نظر انتخاب کرد.

آموزش - XMEGA پورت ها (بخش دوم).

برای تعیین وضعیت پورت ها در AVR های معمولی، به ازای هر پورت ۳ رجیستر وجود دارد. رجیستر DDRx برای تعیین جهت پورت، رجیستر PORTx برای تعیین مقدار خروجی و یا فعالسازی مقاومت Pullup در ورودی و رجیستر PINx برای خواندن مقدار واقعی پین در نظر گرفته شده اند. اما در XMEGA نام عملکرد رجیسترها بصورت متفاوتی است که در اینجا توضیح داده می شود. برای تعیین جهت هر پورت از رجیستری با نام PORTx_DIR استفاده می شود (مثلا). (PORTA_DIR همانند AVR نوشتن یک در هر بیت از این رجیستر منجر به خروجی شدن پین متناظر آن در پورت می شود، اما وضعیت فعالسازی مقاومت Pullup ربطی به این رجیستر ندارد. به غیر از این رجیستر، تعداد ۳ رجیستر دیگر در همین رابطه پیش بینی شده که عملکرد آنها در AVR مشابهی ندارد و برای بالا بردن سرعت انجام عملیات در نظر گرفته شده اند PORTx_DIRSET. رجیستری است که نوشتن یک در هر بیت آن موجب خروجی شدن پین های متناظر در آن پورت می شود، اما وضعیت سایر بیت ها بدون تغییر باقی می ماند. مثلا با اجرای دستور PORTA_DIRSET=0x0F؛ در زبان C فقط ۴ بیت پائین پورت خروجی می شوند و ۴ بیت بالا در هر وضعیتی از نظر ورودی یا خروجی که بودند، باقی می ماند. در همین رابطه دو رجیستر دیگر با نامهای PORTx_DIRCLR و PORTx_DIRTGL وجود دارند که نوشتن یک در بیت های رجیستر اول منجر به ورودی شدن بیت های متناظر و بدون تغییر ماندن جهت سایر بیت ها می شود و برای رجیستر دوم، نوشتن یک منجر به Toggle شدن وضعیت بیت ها از نظر جهت می شود. برای تغییر مقدار خروجی هم ۴ رجیستر برای هر پورت به نام های PORTx_OUT و PORTx_OUTSET و PORTx_OUTCLR و PORTx_OUTTGL وجود دارند که رجیستر اول منجر به تغییر مقادیر کلیه بیت های پورت می شود و در ۳ رجیستر بعدی فقط بیت هایی که در آن یک نوشته شده تغییر می کنند و سایر بیت ها بدون تغییر باقی می ماند.

برای خواندن وضعیت هر پورت هم رجیستر PORTx_IN وجود دارد که عملکردی مشابه رجیستر PINx در AVR های عادی دارد.

مقداردهی به رجیسترهای XMEGA در IAR (بخش اول).

در نمونه کدهایی که تاکنون برای برد آموزشی در بخش دانلود گرفته، برای مقداردهی به رجیسترها از مقادیر عددی استفاده شده است. اما از آنجا که این یک روش بهینه در کدنویسی نیست و با توجه به برگزاری دوره های آموزشی و آشنایی نسبی که اخیراً با این خانواده ایجاد شده، از این به بعد از روش اصولی تری در نمونه کدها استفاده خواهد شد.

برای خواناتر بودن و امکان استفاده مجدد از کدهای نوشته شده و پشتیبانی و نگهداری از نرم افزار، بجای نوشتن مقادیر عددی برای رجیسترها بهتر است از نامگذاری هایی که در فایل Header مربوط به هر شماره وجود دارد، استفاده شود. ۵ نوع نامگذاری مقادیر با پسوند های `_bp` و `_bm` و `_gp` و `_gm` و `_gc` وجود دارد که با ذکر مثال هایی در این مورد توضیح داده می شود.

برای تعیین مقدار یک رجیستر روش اول مقداردهی بصورت عددی است. به عنوان مثال:

```
PORTA_OUT=0X02;
```

روش دوم که بصورت سنتی مورد استفاده قرار می گیرد، استفاده از نامهای معادل بر اساس محل بیت (Bit position) است. این مقادیر برای یک رجیستر ۸ بیتی در محدوده ۰ تا ۷ هستند که نشان دهنده ارزش بیت است و با `_bp` ختم می شوند:

```
PORTA_OUT=( 1 << PIN1_bp);
```

اما روش راحت تر استفاده از مقادیر تعریف شده بر اساس Bit mask است که در آن نیازی به اپراتور شیفت وجود ندارد و در تعریف آن لحاظ شده است. این مقادیر در نامگذاری با `_bm` ختم می شوند :

```
PORTA_OUT=PIN1_bm;
```

برای یک کردن چند بیت متناظر در یک رجیستر می توان از عملیات OR استفاده کرد:

```
PORTA_OUT=PIN1_bm | PIN0_bm;
```

حال اگر تغییر چند بیت در یک رجیستر مورد نظر باشد، به نحوی که سایر بیت ها در وضعیت خود باقی بمانند، از اپراتور `|` و `&=` استفاده می شود:

```
PORTA_OUT |= PIN1_bm;
```

```
PORTA_OUT &= ~(PIN1_bm);
```

دستور اول بیت شماره یک از رجیستر را با حفظ وضعیت بقیه بیت ها یک می کند و دستور دوم همین بیت را صفر می کند.

مقداردهی به رجیسترهای XMEGA در IAR (بخش دوم).

به غیر از تعریف بیت های مجزا در یک رجیستر، برای دسترسی به گروه هایی از بیت ها که وظایف مشخصی دارند، ۳ نوع نام گذاری با پسوندهای `_gm` و `_gc` و `_gp` وجود دارد. به عنوان مثالی از این نوع مقداردهی، برای انتخاب منبع کلاک یک تایمر لازم است به ۴ بیت بصورت همزمان مقداردهی شود. دستور زیر با مقداردهی مستقیم سبب می شود که کلاک وسایل جانبی تقسیم بر ۴ به عنوان منبع کلاک `TCC0` انتخاب شود:

```
TCC0_CTRLA = 0X03;
```

در شکل دوم بجای این دستور می توان از شکل زیر بر اساس تک تک بیت های فعال هم استفاده کرد:

```
TCC0_CTRLA = TC0_CLKSEL1_bm | TC0_CLKSEL0_bm;
```

اما روش اصولی تر آن است که وضعیت مجموعه این ۴ بیت را که با پسوند `_gc` تعریف شده و منظور از آن **Group configuration** است، در یک مرحله در رجیستر مورد نظر قرار داده شود:

```
TCC0_CTRLA = TC_CLKSEL_DIV4_gc;
```

برای فهم بهتر دستورات مذکور می توانیم به تعریف هریک از این مقادیر در فایل **Header** مربوطه مراجعه کنیم:

```
#define TC0_CLKSEL0_bm (1<<0) // Clock Selection bit 0 mask
```

```
#define TC0_CLKSEL1_bm (1<<1) // Clock Selection bit 1 mask
```

```
typedef enum TC_CLKSEL_enum {
    TC_CLKSEL_OFF_gc = (0x00<<0),    // Timer Off
    TC_CLKSEL_DIV1_gc = (0x01<<0),    // System Clock
    TC_CLKSEL_DIV2_gc = (0x02<<0),    // System Clock / 2
    TC_CLKSEL_DIV4_gc = (0x03<<0),    // System Clock / 4
    TC_CLKSEL_DIV8_gc = (0x04<<0),    // System Clock / 8
    TC_CLKSEL_DIV64_gc = (0x05<<0),   // System Clock / 64
    TC_CLKSEL_DIV256_gc = (0x06<<0),  // System Clock / 256
    TC_CLKSEL_DIV1024_gc = (0x07<<0), // System Clock / 1024
    TC_CLKSEL_EVCH0_gc = (0x08<<0),   // Event Channel 0
    TC_CLKSEL_EVCH1_gc = (0x09<<0),   // Event Channel 1
    TC_CLKSEL_EVCH2_gc = (0x0A<<0),   // Event Channel 2
    TC_CLKSEL_EVCH3_gc = (0x0B<<0),   // Event Channel 3
    TC_CLKSEL_EVCH4_gc = (0x0C<<0),   // Event Channel 4
    TC_CLKSEL_EVCH5_gc = (0x0D<<0),   // Event Channel 5
    TC_CLKSEL_EVCH6_gc = (0x0E<<0),   // Event Channel 6
    TC_CLKSEL_EVCH7_gc = (0x0F<<0),   // Event Channel 7
} TC_CLKSEL_t;
```

همانطور که مشاهده می شود، ترکیبات مورد نظر به عنوان یک نوع متغیر تعریف شده و خاصیت این نوع تعریف این است که در تعریف توابع می توان نوع متغیر ورودی را بجای `unsigned char` از نوع `TC_CLKSEL_t` تعریف کرد. در این شرایط اگر مقدار اشتباهی در ورودی تابع اعمال شود، منجر به ایجاد خطا در کامپایلر و تشخیص خطای برنامه نویسی می شود.

سوال بعدی که مطرح می شود این است که برای تغییر چند بیت از یک رجیستر با حفظ سایر بیت های آن باید به چه شیوه ای عمل شود؟ در این مرحله است که مقادیر تعریف شده با پسوند gm که معرف Group mask است، اهمیت خود را نشان می دهد. در این مقادیر تمام بیت های مربوط به یک گروه از بیت ها یک هستند و با استفاده از آن می توان تغییرات مورد نظر را انجام داد. مثلاً در رجیستر PINxCTRL عملاً ۴ گروه مختلف از بیت ها وجود دارند و تغییر یک گروه می تواند با حفظ وضعیت بقیه بیت ها انجام شود. اگر لازم باشد با حفظ وضعیت سایر بیت ها مقاومت Pull down در PA0 فعال شود، می توان از دستور زیر استفاده کرد:

```
PORTA_PIN0CTRL=( PORTA_PIN0CTRL & ~ PORT_OPC_gm) | PORT_OPC_PULLDOWN_gc;
```

دستور اخیر در ابتدا ۳ بیت OPC را با عملیات AND صفر می کند و بعد در محل آن مقدار مناسب را با OR قرار می دهد. با مراجعه به تعریف زیر مفهوم این دستور روشن تر می شود:

```
#define PORT_OPC_gm 0x38 // Output/Pull Configuration group mask
```

آخرین نوع مقادیر که با پسوند gp تعریف شده اند، اهمیت چندانی در برنامه نویسی ندارند و تنها مکان اولین بیت با ارزش کمتر را در گروه بیت ها در رجیستر مشخص می کنند:

```
#define PORT_OPC_gp 3 // Output/Pull Configuration group position
```

آموزش - XMEGA واحد نظارت بر کلاک خارجی

یکی از امکاناتی که در بخش تولید کلاک XMEGA قابل فعال سازی است، واحد نظارت بر کلاک خارجی یا External Clock source failure monitor است. این عملکرد در شرایطی مفید واقع می شود که کلاک سیستم از کریستال یا اسیلاتور خارجی تامین شده باشد. با فعال سازی این واحد که از طریق رجیستر OSC_XOSCFAIL انجام می شود، در صورت از کار افتادن و یا کمتر شدن فرکانس نوسان ساز خارجی از یک حد مشخص، منبع نوسان بصورت خودکار به اسیلاتور داخلی ۲ مگاهرتز سوئیچ می شود. همزمان با انجام این تغییر یک وقفه NMI یا Non Maskable Interrupt ایجاد می شود که از تمام وقفه های دیگر دارای اولویت بالاتری است و قابل mask شدن هم نیست. در صورت فعال سازی این واحد باید روتین مناسب هم برای وقفه آن پیش بینی شود.

حد پائین برای مقدار فرکانس اسیلاتور ۳۲ کیلوهرتز است که در صورت کمتر شدن فرکانس منبع نوسان خارجی از این حد، عمل تغییر منبع نوسان به اسیلاتور داخلی ۲ مگاهرتز و ایجاد وقفه انجام می شود. اگر اسیلاتور خارجی فعال باشد ولی به عنوان منبع کلاک سیستم انتخاب نشده باشد، شرایط Fail منجر به غیر فعال شدن آن می شود. همچنین بعد از فعال کردن واحد نظارت بر کلاک خارجی، امکان غیرفعال کردن آن وجود ندارد و فقط وقوع Reset منجر به غیر فعال شدن آن می شود. در صورت قرار گرفتن CPU در وضعیت SLEEP این واحد متوقف می شود و بعد از خارج شدن از وضعیت SLEEP این واحد بصورت خودکار فعال خواهد شد.

آموزش XMEGA - ADC (بخش اول).

در خانواده XMEGA حداکثر دو واحد ADC روی پورت های A و B وجود دارد که هر واحد دارای ۴ کانال اندازه گیری می باشد. از طریق این ۴ کانال، ۴ اندازه گیری همزمان با مکانیزم Pipeline قابل انجام است که امکان رسیدن به سرعت ۲ میلیون نمونه در ثانیه (۲ MSPS) را ایجاد می کند. برای هر ADC به غیر از ۸ ورودی آنالوگ خارجی، امکان اندازه گیری برخی منابع داخلی هم وجود دارد. منابع داخلی قابل اندازه گیری به شرح زیر هستند:

1 - خروجی سنسور دمای داخلی

2 - VCC/10

3 - Bandgap voltage

4 - خروجی DAC

اندازه گیری هر ۴ کانال باید بصورت علامت دار (Signed) یا بدون علامت (Unsigned) انجام شود. دقت اندازه گیری هم می تواند ۸ یا ۱۲ بیتی تعریف شود که برای هر ۴ کانال مشترک است. ولتاژ مرجع برای اندازه گیری می تواند بصورت داخلی و یا خارجی باشد که دو گزینه $VCC/1.6$ و یک ولتاژ ۱.۰۰V دقیق به عنوان مرجع داخلی و دو ورودی خارجی به عنوان مرجع قابل تعریف هستند.

هر کانال ADC قابلیت تعیین ورودی ها بصورت تکی (Single ended) یا تفاضلی (Differential) را دارد و این انتخاب می تواند بین کانال های یک ADC بصورت مستقل تنظیم شود. در وضعیت تفاضلی امکان تعریف یک ضریب تقویت در ورودی تا حداکثر ۶۴ وجود دارد و در خانواده XMEGA D امکان تعریف یک ضریب تضعیف ۲/۱ هم وجود دارد. به عنوان یک جمع بندی، ۶ نوع اندازه گیری برای ADC به شرح زیر قابل تعریف است:

- 1- Unsigned single ended
- 2- Unsigned Internal
- 3- Signed single ended
- 4- Signed Internal
- 5- Signed differential
- 6- Signed Differential with Gain

آموزش XMEGA - ADC (بخش دوم).

حداکثر کلاک برای سری A برابر ۲ مگاهرتز و برای سری D برابر ۱.۴ مگاهرتز است. این کلاک از طریق تقسیم CLKPER با یک تقسیم کننده برنامه پذیر محقق می شود. شروع یک تبدیل جدید می تواند بوسیله فرمان نرم افزاری و یا Event system انجام پذیرد. همچنین در مد Free Run فرمان تبدیلات متوالی بصورت خودکار و یکی پس از دیگری صادر می شود. از زمان شروع فرمان تبدیل تا آماده شدن نتیجه در کانال مربوطه، ۵ کلاک به ازای دقت ۸ بیت و ۷ کلاک به ازای دقت ۱۲ بیت زمان صرف می شود. اگر طبقه تقویت ولتاژ هم انتخاب شده باشد، یک کلاک به این مقادیر افزوده می شود. در پایان این زمان flag مربوط به وقفه فعال می شود که می تواند منجر به ایجاد وقفه گردد. در همین رابطه بحث Pipe Line بودن ADC اهمیت زیادی پیدا می کند. به این مفهوم که اگر فرمان تبدیل برای هر ۴ کانال بصورت توأم ایجاد شود، یک تقدم زمانی برای کانال های با شماره کوچکتر وجود دارد و به دلیل استفاده مشترک از سخت افزار ADC، بعد از آماده شدن نتیجه کانال اول، با تاخیر یک کلاک نتیجه کانال دوم و به همین ترتیب نتیجه دو کانال بعدی در رجیسترهای متناظر آنان ظاهر می شود. حال سوال اساسی آن است که با این ترتیب چگونه می توان به

سرعت ۲ میلیون نمونه در ثانیه رسید؟ با توجه به مطالب جدیدترین نسخه (AVR1300 که در اکتبر ۲۰۱۰ تصحیح شده)، برای خانواده XMEGA A چنین امکانی وجود دارد و بر اساس جمع بندی کلیه مطالب و انجام آزمایش های عملی، چنین سرعتی در مد Free Run محقق می شود.

یکی از امکانات قابل ذکر در ADC وجود یک مکانیزم Compare است. به این معنی که مقدار هر کانال ADC با یک رجیستر مشخص مقایسه می شود و در صورتی که از آن بیشتر یا کمتر باشد، امکان ایجاد وقفه ای متناسب با شرایط تعریف شده بوجود می آید.

آموزش XMEGA - ADC (بخش سوم).

تابع تبدیل ADC در اندازه گیری علامت دار (Signed) بصورت زیر است:

$$RES = \frac{V_{INP} - V_{INN}}{V_{REF}} * GAIN * TOP$$

در فرمول اخیر VINP ولتاژ اعمال شده به ورودی مثبت و VINN ولتاژ اعمال شده به ورودی منفی است. VREF مقدار ولتاژ مرجع انتخاب شده برای اندازه گیری است. مقدار GAIN می تواند بصورت توانهای ۲ و در محدوده ۱ تا ۶۴ باشد. در XMEGA A Manual برای مقدار TOP عدد ۲۰۴۷ برای دقت های اندازه گیری ۱۲ بیتی ذکر شده است. این در حالی است که در جدید ترین note Application منتشر شده در اکتبر ۲۰۱۰، برای TOP مقدار ۲۰۴۸ در نظر گرفته شده که در همین رابطه مکاتبه ای از طرف نگارنده این مقاله با Atmel انجام شد و در نهایت همان عدد ۲۰۴۷ مورد تأیید قرار گرفت. به همین دلیل انتظار می رود بزودی این تصحیح از طرف Atmel در متون خود اعمال شود. برای دقت اندازه گیری ۸ بیت هم این عدد ۱۲۷ خواهد بود.

تابع تبدیل ADC در اندازه گیری بدون علامت (Unsigned) بصورت زیر است:

$$RES = \frac{V_{INP} + \Delta V}{V_{REF}} * TOP$$

در مورد TOP در فرمول بالا، باز هم تناقضی بین متن Application note و Manual وجود دارد که در این مورد هم با مکاتبه با Atmel عدد ۴۰۹۵ مورد تأیید قرار گرفت. در دقت اندازه گیری ۸ بیتی هم این عدد ۲۵۵ می باشد. ΔV در فرمول بالا دارای مقداری در حدود $V_{REF} * 0.05$ است و به این دلیل در فرمول آورده شده که ترکیب سخت افزاری در مد بدون علامت به گونه ای است که با فرض عدم وجود هرگونه Offset و به ازای ورودی صفر، در خروجی عددی در حدود ۲۰۰ ایجاد خواهد شد. در اندازه گیری بدون علامت با استفاده از این خاصیت می توان مقدار Offset موجود در مدار ADC را اندازه گرفت. زیرا وجود Offset منفی می تواند موجب کاهش خروجی واقعی نسبت به این عدد شود و بنابراین می توان از روی تفاضل خروجی ایده ال نسبت به مقدار واقعی به مقدار Offset دست پیدا کرد.

حداکثر ولتاژ اعمال شده به هر ورودی ADC که منجر به رسیدن نتیجه تبدیل به حداکثر مقدار خود می شود در مد علامت دار به میزان V_{REF} و در مد بدون علامت $V_{REF} - \Delta V$ خواهد بود.

آموزش - XMEGA مقایسه کننده آنالوگ

حداکثر ۴ مقایسه کننده آنالوگ روی پورتهای A و B در XMEGA وجود دارند که امکان مقایسه سریع دو سیگنال آنالوگ را فراهم می کنند. ورودی مثبت مقایسه کننده، امکان اتصال به پین های ۰ تا ۶ پورت و خروجی DAC را دارد. به ورودی منفی، پین های ۰ و ۱ و ۳ و ۵ و ۷ و خروجی DAC و ضربی از V_{CC} در محدوده ۶۴/۱ تا ۱ و ولتاژ داخلی Bandgap قابل اتصال است. از نظر خروجی، پین شماره ۷ از پورت قابلیت اتصال به خروجی یکی از مقایسه کننده ها را دارد. در شکلهای متون Atmel، این پین به اشتباه پین صفر نمایش داده شده که توسط نویسنده این مطلب به اطلاع بخش پشتیبانی آن شرکت رسانده شده است. تغییرات خروجی مقایسه کننده ها می تواند منجر به ایجاد وقفه و Event در لبه بالارونده یا پائین رونده یا هردو لبه گردد. امکان تغییر عرض باند هیستریزیس و تاخیر انتشار مقایسه کننده هم از جمله امکانات قابل ذکر در مقایسه کننده های آنالوگ می باشد.

یکی از امکانات دیگر این مقایسه کننده ها Window mode است که در محدوده ای از ولتاژ ورودی، امکان ایجاد وقفه یا Event را فراهم می کند. در این حالت باید دو مقایسه کننده بکار گرفته شوند و ورودی های مثبت آنها به یکدیگر و به ولتاژ مورد سنجش متصل شوند. دو ورودی منفی هم به ولتاژهای محدوده بالا و پائین Window مورد تعریف، وصل می شوند. بلفعال سازی و تنظیم Window در یکی از ۴ وضعیت بالاتر از محدوده بالایی پنجره، بین دو محدوده، خارج از دو محدوده و پائین تر از حد پائین، وقفه و Event ایجاد می شود.

مقایسه کننده های آنالوگ در ترکیب با خروجی DAC می توانند بسیار سریع تر از ADC برای تشخیص بزرگتر یا کوچکتر بودن سیگنال مورد سنجش نسبت به یک Setpoint مشخص، عمل کنند و این مسئله کاربرد ویژه ای را برای آنها در مدارات سوئیچینگ و کنترل جریان ایجاد می کند.

آموزش - XMEGA سنسور دمای داخلی

یکی از امکانات خوب خانواده XMEGA، وجود یک سنسور دما در داخل IC است که خروجی آن از طریق ADC داخلی قابل اندازه گیری می باشد. ولتاژ خروجی این سنسور به گونه ای تغییر می کند که منحنی مشخصه آن در دمای صفر کلوین از مبدا مختصات عبور می کند. با این فرض و برای تکمیل مشخصه سنسور، یک اندازه گیری در دمای ۸۵ درجه سلسیوس (۳۵۸ درجه کلوین) در زمان پروسه ساخت میکروکنترلر انجام شده و در آدرس مشخصی از حافظه میکروکنترلر ذخیره شده است. با قرائت این مقدار توسط نرم افزار، معادله خط مورد نظر به عنوان مشخصه سنسور بدست می آید و از این طریق سایر دماها هم قابل محاسبه و اندازه گیری هستند. در محاسبه این مشخصه باید به دو نکته توجه شود. اول اینکه این اندازه گیری در مد بدون علامت انجام شده که جمله ΔV که در توضیحات ADC بدان اشاره شد، در محاسبه وجود دارد. مورد بعدی ولتاژ مرجع در اندازه گیری است که از ولتاژ ۱.۰۰ ولت داخلی استفاده شده است. در صورت استفاده از ولتاژ مرجع دیگری به غیر از ۱.۰۰ ولت داخلی، این امر باید در محاسبات لحاظ شود.

از نکات قابل اشاره دیگر این است که کلاک ADC در اندازه گیری سنسور دمای داخلی، دارای محدودیت در مقدار است و نباید از ۱۲۵ کیلوهرتز فراتر رود. این محدودیت کلاک در اندازه گیری سایر منابع داخلی ADC هم وجود دارد و برای بدست آمدن یک نتیجه صحیح باید در اندازه گیری این منابع هم رعایت شود.

حداکثر دو واحد DAC با دو خروجی مستقل (برای هر واحد) در XMEGA وجود دارد. در شرایط ایده آل واحد DAC دارای دقت ۱۲ بیت و سرعت خروجی ۱ MSPS است. البته در Revision های سخت افزاری موجود از XMEGA، دقت بخش های ADC و DAC با کاستی هایی همراه است و نیاز به اصلاحاتی در تولیدات آتی شرکت Atmel وجود دارد. هر واحد DAC می تواند در دو وضعیت تک خروجی (Single channel) و یا دو خروجی (Dual channel) تنظیم شود که حداکثر دقت و سرعت موجود در شرایط استفاده از همان وضعیت اول است و سرعت خروجی در وضعیت دوم کاهش می یابد. ولتاژهای مرجع قابل انتخاب می توانند یکی از گزینه های ولتاژ مرجع داخلی ۱.۰۰V و AVcc و REFA یا REFB باشند. در همین رابطه برای مشاهده شرایط عملکرد صحیح DAC و ADC، مراجعه به بخش Errata در انتهای Datasheet هر خانواده ضروری است و نباید به مطالب موجود در XMEGA A manual اکتفا شود. زیرا با مراجعه به بخش مذکور مشاهده می شود که یکی از شرایط عملکرد صحیح DAC آن است که ولتاژ مرجع آن از مقدار مشخصی کوچکتر باشد و بنابراین گزینه AVcc برای ولتاژ مرجع در عمل قابل استفاده نیست. در غیر اینصورت میزان قابل قبول بودن خروجی به میزان مشخصی برای هر شماره، کاهش می یابد.

عملیات تبدیل DAC می تواند با نوشتن مقدار رجیستر Data مربوطه و یا با فرمان Event system انجام پذیرد. همچنین بخش DAC وقفه یا Event تولید نمی کند، اما از طریق وجود دو Flag برای هر DAC می توان به آمادگی رجیسترهای مربوطه برای نوشتن Data پی برد. این Flag ها می توانند برای فرمان به DMA هم بکار گرفته شوند.

برای استفاده دقیق تر از DAC باید عملیات کایبره کردن آن بصورت نرم افزاری انجام شود. در همین رابطه دو رجیستر به اسامی GAIN و OFFSET وجود دارند که برای مقداردهی صحیح به آنها باید خروجی DAC بصورت داخلی به ورودی ADC متصل شود و با یک الگوریتم مشخص و بر اساس مقدار قرائت شده از ADC، مقادیر این دو رجیستر به شیوه مناسب تغییر داده شود.

در نسخه های موجود XMEGA و با توجه به اطلاعات موجود در بخش Errata، برای رسیدن به حداکثر دقت در DAC باید اولاً از مد Single channel استفاده شود و دوم اینکه ولتاژ مرجع کوچکتر از ۲.۴ ولت و ۰.۶Vcc باشد.

AWeX (Advanced Waveform Extension) یک واحد سخت افزاری طراحی شده برای پیاده سازی عملیات کنترل دور موتور و درایورهای قدرت است. وجود ۳ توانایی سخت افزاری Dead time insertion، Pattern generation و Fault protection برای کاربردهای کنترل دور موتور و منابع تغذیه و UPS و مانند آن پیش بینی شده و کارکردن با آن مستلزم آشنایی کامل با مبانی عملکرد درایورهای قدرت می باشد.

در مورد Dead time insertion می توان این توضیح را ارائه کرد که در یک پل (Bridge) تشکیل شده از سوئیچ های قدرت مانند Mosfet یا IGBT، بین خاموش شدن هر سوئیچ قدرت و روشن شدن سوئیچ دومی که در همان بازو قرار دارد، باید یک زمان حداقل در حد میکروثانیه تاخیر وجود داشته باشد. در غیر اینصورت قبل از اینکه جریان سوئیچ روشن به صفر برسد، سوئیچ دوم هم روشن خواهد شد و یک جریان بسیار شدید بین ولتاژ مثبت و منفی پل ایجاد می شود و در یک لحظه باعث از بین رفتن سوئیچ های قدرت خواهد شد (به این جریان through shoot هم گفته می شود). به این زمان تاخیر به اصطلاح Dead time گفته می شود که یک منبع تولید فرمان برای درایور، باید بتواند آنرا بصورت سخت افزاری تامین کند. در واحد AWeX خانواده XMEGA، چنین امکانی پیش بینی شده که مقدار Dead time بصورت نرم افزاری و توسط رجیسترهای کنترلی یکبار تعریف و به سخت افزار اعمال شود.

Pattern generation برای تولید یک Bit Pattern مشخص روی پورت خروجی و برای کنترل موتورهای مانند BLDC و استپر موتور کاربرد دارد. در این وضعیت محتوای پورت بصورت سنکرون با تایمر و بر حسب یک بافر مشخص، بصورت خودکار تغییر می کند. به این معنی که در شرایط Update تایمر (که می تواند در TOP یا BOTTOM اتفاق بیافتد)، یک وضعیت از پیش تعریف شده روی پورت قرار می گیرد که این وضعیت برای هر پین می تواند High و Low و یا خروجی PWM کانال A تایمر باشد. از طریق این امکان می توان یک سیگنال PWM را در ۸ پین مختلف و بسته به شرایط توزیع نمود.

برای تفهیم بهتر اینکه این امکان چه فایده هایی می تواند داشته باشد، مثال هایی را می توان مطرح کرد. مثلاً اگر فرض کنیم قرار باشد یک استپر موتور بوسیله ۴ بیت پورت کنترل شود، بجای تعریف وقفه ای که بخواهد مقدار پورت را بصورت متوالی تغییر دهد، می توان مقادیر لازم را در بافر پورت قرار داد و در زمان لازم و سنکرون با تایمر، عملیات ایجاد شکل موج مورد نظر انجام خواهد شد و برای اینکار نیازی به وقفه تایمر نیست. همین مسئله می تواند در تابلوهای روان و مانند آن هم با سرعت زیاد و بدون درگیر کردن CPU در وقفه تایمر، بکار گرفته شود.

و یا توزیع انتخابی PWM که می تواند در کاربردهایی (احتمالاً کنترل نور LED های معمولی و RGB و ...) مورد استفاده قرار بگیرد و با استفاده از این امکانات می توان بار CPU را برای انجام برخی از کاربردها کاهش داد و آن را به عهده سخت افزار سپرد.

Fault protection هم یک واحد پیش بینی شده برای درایورهای قدرت و در جهت واکنش سریع و مشخص در شرایط وقوع Fault می باشد. برای کنترل کاربردهای قدرت لازم است این امکان بصورت سخت افزاری پیش بینی شود که در هنگام وقوع fault و خطا، واکنش سریع و از قبل معینی انجام شود و مثلاً فرمان خروجی به سوئیچ های قدرت به سرعت قطع شوند و این امر نباید معطل اجرای نرم افزار بماند. چرا که در غیر اینصورت ممکن است به علت جریان کشی بیش از حد به از بین رفتن سوئیچ های قدرت و مواردی از این دست منجر شود. بنابراین واحدی که تحت برنامه ریزی قبلی نرم افزار بصورت سخت افزاری به شرایط خطا پاسخ دهد، از جمله امکاناتی است که در XMEGA پیش بینی شده و امکان استفاده از آن را به عنوان یک کنترل کننده و نظارت کننده بر پروسه های در محدوده زمانی میکروثانیه و کمتر، بوجود می آورد. این بخش از Event system فرمان می گیرد و این امکان وجود دارد که هر یک از ۸ کانال موجود (بصورت OR) به عنوان فرمان ایجاد شرایط Fault عمل کنند.

منابع :

سایت : <http://www.yazdkit.com/>

سایت : <http://www.novintarashe.com/>