Microprocessor

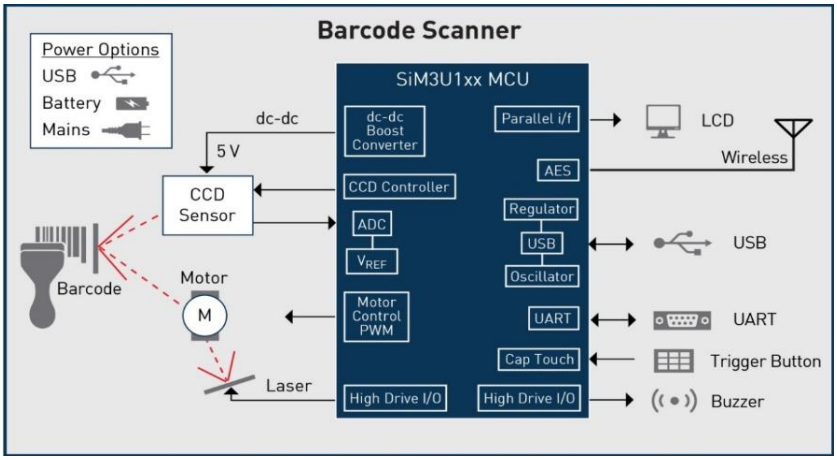
Homework-1

ارمغان سرور 9531807

.1

الف) همانطور که در فایل پیوست اشاره شده است، برای خواندن بارکد، scanner به نوسان کننده لیزری می-تاباند که بارکد را روشن می کند. سپس تصویر با یک سنسور CDD ذخیره شده (این ذخیره سازی به صورت دریافت خط به خط پیکسل ها در هر لحظه از زمان انجام می گیرد) و سپس نور پیوسته وارد یک تبدیل کننده آنالوگ به دیجیتال (ADC) خواهد شد. حال اگر ریزپردازنده ای انتخاب کنیم که بتواند جریان بالایی تامین کند نیاز به ترانزیستورهای قدرت مورد استفاده برای راندن لیزر و موتور از بین می رود. همینطور وقتی میکروکنترلر بتواند برای همگام سازی clock واسطی با سنسور CDD مهیا کند برای طراحان ساده تر خواهد شد. پس به طور کلی میتوانیم این اجزا را به صورت زیر نام ببریم:

Scanner برای تاباندن لیزر - حسگر CCD برای ذخیره تصویر – موتور برای فعال سازی آیینه نوسان کننده – تبدیل کننده آنالوگ به دیجیتال) - نوسانگر و رگولاتور(تنظیم کننده) 5 ولتی USB – 6 عدد پین قابل برنامه ریزی که قادر به فراهم کردن تا 300 میلی آمپر جریان هستند و 16 کانال ورودی سنجش خازنی برای دکمه های لمسی (cap touch) – واحد دریافت و انتقال ناهمگام (UART) که برای ارتباط سریال ناهمزمان به کار می رود.– آیینه ی بازتاب کننده: با تاباندن نور لیزر بر آن، بارکد نمایش داده میشود. – فعال کننده ی یک buzzer برای سیگنال دادن به کاربر هنگام انجام شدن یک scan موفقیت آمیز



ب) برای این کاربرد میکروکنترلر Precision32 SiM3U1xx USB MCUs می تواند یک ساعت همگام و هماهنگ شده را به سنسور انتقال دهد و به سادگی با نرخ نمونه برداری سریعی که CCD (که در قسمت قبل به آن اشاره شد) هماهنگ باشد و همچنین از 3.3 تا 5 ولت کنترل کننده dc-dc boost مهیا کند که باعث کاهش تعداد اجزا می شود.

از طرفی تنظیم کننده ولتاژ که در scannerها قرار داده شده باعث شده است میکروکنترلر بدون انرژی USB عمل کند و نوسانگر داخلی با فرکانس 48MHz که بیش از 25 درصد دقت فراهم می کند و اجازه می دهد عملکرد USB نیازی به کریستال نداشته باشد.(crystal less operation)

این میکروکنترلر طراحی شده است برای فراهم کردن تعدادی از ویژگی های یکپارچه که در دیگر میکرو ها یافت نمیشود مثل: 6 پین قابل برنامه ریزی high-drive که میتواند تا جریان 300 میلی آمپر را فراهم کند ، یک نوسان ساز USB ،تنظیم کننده 5 Vو 16 کانال ورودی سنجش خازنی برای دکمه های لمسی یا لغزنده.

این سطح بالا از ادغام کمک می کند نیاز به اجزای مختلف گسسته از بین رفته و یک دامنه قدرت انعطاف پذیر فراهم شود که موجب کاهش هزینه ی مواد و ساده سازی روند توسعه خواهد شد.

ج) قابلیت دیگر که در این میکروکنترلر(barcode scanner) جاسازی شده است و در بخش الف جزو اجزای این دستگاه به شمار آمد امکان alert دادن به کاربران توسط buzzer است که انجام شدن و نشدن scan را اعلام کند و همچنین مساله امنیت که توسط رمزنگاری توسط HW برای ورودی اسکنر(اسکنرهای wireless) قابل انجام است.(انتقال امن داده ها) از قابلیت های دیگر آن است.

همچنین برای داشتن قابلیت انعطاف پذیری نسبت به تغیرات ، MCUها یک محل از پیش تعیین شده برای لوازم جانبی با انتخاب متناوب فراهم می کنند. تنظیمات از پیش تنظیم شده اغلب منجر به درگیری های پین شده و باعث می شود توسعه دهندگان طراحی خود را تغییر دهند و به بسته بندی بزرگتر و گران تر روی آورند. بنابراین به یک معماری با انعطاف پذیری بیشتری نیاز داریم و میکروکنترلر دوطرفه توسط Silicon Labs ارائه شده است که انعطاف پذیری را برای توسعه دهندگان فراهم می کند تا بتوانند خودشان لوازم جانبی مورد نیاز و قرارگیری پین ها را انتخاب کنند.

داشتن توانایی انتخاب وسایل جانبی مورد نیاز، میتواند مقرون به صرفه تر باشد. با این حال، با استفاده از معماری استاندارد ثابت، چهارUART و سه SPI، 64 پین یا حتی یک پین 100 پین را برای دستیابی به ترکیب دقیق محیطی الزام آور می کند.

د)

ARM Cortex M3 (80 MHz)

80 MHz maximum frequency
Single-cycle multiplication, hardware division support
Nested vectored interrupt control (NVIC) with 16 priority levels

32/64/128/256 kB Flash

8/16/32 kB SRAM w/ 4kB Retention

16-Channel DMA Controller

Watchdog

Serial Wire / JTAG / ETM

ه) برخی از توسعه دهندگان نگران هستند که ممکن است یک معماری crossbar برای برنامه ریزی پیچیده باشد

و Silicon Labs با ایجاد AppBuilder که در حقیقت یک ابزار توسعه نرم افزار رایگان طراحی شده برای ساده سازی اولیه و پیکربندی(IDE) می باشد، این نگرانی را رفع کرده است.

یک AppBuilder مبتنی بر GUI توسعه دهندگان را قادر می سازد به سرعت و به صورت گرافیکی ترکیب محیط خود را انتخاب کنند، بدون نیاز به مطالعه نمودن data sheet ، ویژگی های محیطی را انتخاب کرده و تنظیمات حالت ساعت pinout ها را تنظیم کنند.

از دیگر امکانات آن این است که AppBuilder حتی سورس کد را تولید می کند و بنابراین می تواند با کامپایلر های محبوب مانند Keil، IAR و GCC سازگار باشد.

و) یک نکته مهم در انتخاب میکروکنترلر 32 بیتی، بهینه بودن مصرف توان است و مصرف توان فوق العاده کم به نگرانی اصلی در طیف گسترده ای از برنامه های تعبیه شده تبدیل شده است. در این کاربرد، مهمترین هدف این است که مصرف انرژی در حالت خواب به حداقل برسد. سنسور می تواند در حالت خواب نیز باشد به این صورت که با بیدار شدن ، تعیین کند آیا یک رویداد مانند تشخیص دود، اتفاق می افتد یا خیر و سپس به خواب برگردد. پس به طور کلی در این نوع سیستم، مهم است که حالت خواب با مصرف کم توان داشته باشیم که شامل یک ساعت زمان واقعی (RTC) برای بیدار شدن از طریق تنظیم کننده داخلی، مانند هر 100 میکرو ثانیه است.

همچنین زمان بیدار شدن سریع اهمیت داشته و بهتر است یک پردازنده برای سرعت بخشیدن به تعداد دستورات برای تعیین اتفاق افتادن یک رویداد داشته باشیم.

بعضی از برنامه ها مانند تجهیزات کارخانه هرگز به خواب نمی روند. در این برنامه ها داشتن یک میکروکنترلر با جریان با مصرف کم توان بسیار اهمیت دارد. ترفندهای دیگری نیز هستند که برای صرفه جویی در مصرف توان استفاده می شوند، از قبیل کاهش میزان فرکانس پرواز در پرواز تنها برای استفاده از سرعت پردازش مورد نیاز برای عملی بخصوص. یافتن یک میکروکنترلر 32 بیتی که در حالت خواب، فعال و بیداری کم مصرف باشد، می تواند چالش برانگیز باشد. خانواده ی Precision32 این چالش را با یک سری از قابلیت ها حل می کند که در شکل زیر نمایش داده شده است:

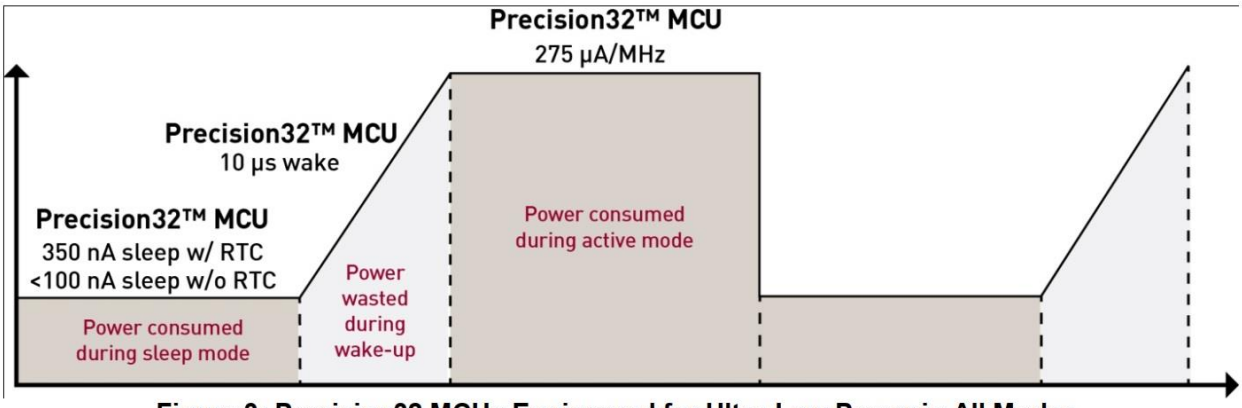


Figure 3: Precision32 MCUs Engineered for Ultra-Low Power in All Modes

میتوان گفت MCUs می توانند در کمتر از 100nA کار کنند و شامل یک آشکارساز و 4 کیلو بایت حافظه رم باشند و ساعت زمان واقعی می تواند برای 250nA اضافی از جریان فعال شود.

امکانی دیگر وجود یک مقایسه کننده آنالوگ برای 400 نانوگرم اضافی است و حتی یک تایمر و شمارنده پالس با مصرف توان پایین فراهم گشته است. علاوه بر این MCUs حاوی PLL ای هستند که می تواند به هر فرکانسی از 1 - 80 MHz قفل شود و کمک کند که خود برنامه نویس به سادگی توان را به صورت دستی بهینه گرداند.

ز) روش های دسته بندی تراشه میکروکنترلر عبارتند از 92 pin LGA ، 64 Pin TQFP ، 64 pin QFN

، 80-Pin TQFP و 40 pin QFN که هر یک حاوی اطلاعات بخصوص برای pinها در جداول موجود در فایل هستند.

ح) محدوده ی دمایی آورده شده در پیوست از 40- تا 85+ درجه سانتی گراد می باشد که با عنوان "دمای محیط کار" نیز آورده شده است.

همچنین دمای اتصال (operating junction) از 40- تا 105 پیشنهاد شده است.

ط) ولتاژ پیشنهادی منبع تغذیه در VDD: از 1.8 تا 3.6 (همان ولتاژ VDD حین برنامه ریزی و غیرفعالسازی رگولاتور)

فعالسازی رگولاتور: از2.7 تا 5.5 ولت

ولتاژ پیشنهادی منبع تغذیه در VIO : از 1.8 تا VDD

.2

SUB AX, 3030H

در این دستور طبق توضیح مقدار immediate مربوط به 3030H درون ثبات ریخته می شود که نیازی به سیکل خواندن از حافظه ندارد.

		خواندن از حافظه(opcode)	نوع عملیات
		کد hex عمل sub	وضعیت باس داده
		خواندن از حافظه	وضعیت باس کنترل

MUL BL

	خواندن از حافظه(opcode)	نوع عملیات
	کد hex عمل mul	وضعیت باس داده
	خواندن از حافظه	وضعیت باس کنترل

ADD BYTE PTR[BX], CH

	نوشتن در حافظه	خواندن از حافظه	خواندن از حافظه(opcode)	نوع عملیات
	نتیجه عملیات sub	محتوای درون آدرس BX	کد hex عمل add	وضعیت باس داده
	نوشتن در حافظه	خواندن از حافظه	خواندن از حافظه	وضعیت باس کنترل

POP AX

در این بخش increment کردن ثبات sp جزو پایه های آن فرض شده که نیازی به سیکل باس ندارد.

حال اگر منظور سوال 8086 باشد که sp در آن در پردازنده قرار دارد داریم:

خواندن از حافظه	خواندن از حافظه(opcode)	نوع عملیات
خواندن داده ی موردنظر از پشته	کد hex عمل pop	وضعیت باس داده
خواندن از حافظه	خواندن از حافظه	وضعیت باس کنترل

و اگر منظور سوال ATMega16 باشد که در آن sp در I/O قرار دارد داریم:

POP AX

	خواندن از حافظه	خواندن از حافظه	خواندن از حافظه(opcode)	نوع عملیات
	خواندن داده ی موردنظر	خواندن اشاره گر پشته	کد hex عمل pop	وضعیت باس داده
	خواندن از حافظه	خواندن از حافظه	خواندن از حافظه	وضعیت باس کنترل

STC

	خواندن از حافظه(opcode)	نوع عملیات
	کد hex عمل stc	وضعیت باس داده
	خواندن از حافظه	وضعیت باس کنترل

LSL

	خواندن از حافظه(opcode)	نوع عملیات
	کد hex عمل lsl	وضعیت باس داده
	خواندن از حافظه	وضعیت باس کنترل

.3

ORG: مشخص می کند دستورات از چه خانه ای در حافظه شروع شوند(شمارنده مکان) در اینجا از آدرس 3030H

مثال:برای زمان هایی که از آدرس ثابت استفاده می کنیم کاربردی ندارد.

EQU: مانند دستور define در C است که دو عملوند دریافت نموده و بیان میکند هر کجا عملوند اول ظاهر شد کامپایلر عملوند دوم را برای آن به حساب آورد. (نام دهی سمبولیک برای مقدار ثابت عددی یا مقداری نسبت به ثبات یا pc) پس هر کجا true توسط کامپایلر مشاهده شود 1 گذاشته می شود.

DW: define word دستوری برای رزرو نمودن 2 بایت در حافظه (دستور به کامپایلر داده می شود و مانند تعریف متغیر است) که با نام pressure تعریف می شود.

DD: define double رزرو نمودن 4 بایت فضا در حافظه که با نام Length تعریف می شود.

DB: define byte رزرو نمودن 1 بایت فضا د ر حافظه که با نام String تعریف می شود.

ترتیب ذخیره بایت ها:

(Processor uses the little-endian byte ordering)

Pressure DW 300

Binary of 300 = 0000 0001 0010 1100

Hex of 300 = 12C

Little endian byte storing order = 2C01 (from low to high value in our number)

Address in memory:

2C: 3030H

01: 3031H

Length DD 11110000111100001111b

From left to right is from low address in memory to high address: 0F0F0F00

Address in memory:

0F: 3032H – 0F: 3033H – 0F: 3034H – 00: 3035H

String DB "CharacterString"

Order of byte storing and Address in memory:

c (3036H) – h (3037H) – a (3038H) – r(3039H) – a(303AH) – c(303BH) – t(303CH) – e(303DH) – r(303EH) – s(303FH) – t (3040H) – e (3041H) – r(3042H) – s(3043H) – t (3044H) – r(3045H) – I(3046H) – n (3047H) – g (3048H)

There is no endian issue here and Characters are stored in order.