

(الف)

برای خواندن بارکد، اسکنر، لیزر را روی یک آینه نوسانگر که توسط موتور تغذیه می شود می اندازد. نور، بارکد را روشن می کند و تصویر بارکد توسط سنسور CCD گرفته می شود. سنسور CCD هر بار یک خط پیکسل را دریافت می کند. سیگنال های نوری رسیده به سطح به یک مبدل ADC داده می شوند. داشتن میکروکنترلر با جریان بالا، نیاز به داشتن ترانزیستور های تغذیه برای درایو کردن لیزر و موتور را از بین می برد. از طرفی داشتن اینترفیس با سنسور CCD برای همگام سازی کلاک میتواند کار را برای طراح آسان تر کند.

اجزای ریزپردازنده عبارتند از:

dc-dc Boost Converter : برای تغذیه سنسور

CCD Controller : برای تغذیه سنسور و همچنین ارتباط با آن

ADC : مبدل آنالوگ به دیجیتال

Motor Control PWM تولید کننده پالس با تکنیک های مدولاسیون

High Drive I/O: برای گرفتن جریان تا به ۳۰۰ میلی آمپر

Cap Touch: برای این که با فشردن دکمه ها بتوان ورودی را trigger کرد، با استفاده از خازن این امکان فراهم شده است.

UART: برای انتقال ناهمگام داده serial

Oscillator: برای تولید کلاک

USB: یکی از راههای تغذیه دستگاه، در کنار برق شهری و باتری

Regulator تنظیم کننده ولتاژ

AES: مازولی جهت رمزنگاری بصورت سخت افزاری

Buzzer: نوعی زنگ برای آگاه کردن کاربر از اسکن موفقیت آمیز بارکد

Parallel i/f: واسط ارتباط موازی. مثلا با نمایشگر LCD

32-256KB Flash: حافظه flash

8-32KB SRAM: حافظه SRAM

GPIO: پورت های ورودی و خروجی برای کاربرد های عام منظوره

(ب)

در همین مثال اسکنر بارکد، پردازنده Precision32 SiM3U1xx USB میتواند به سنسور یک کلاک سنکرون بدهد که بدین وسیله امکان نمونه برداری CCD با سرعت های بالا فراهم شود. و همچنین میتواند توسط کنترلری با ولتاژ ثابت سنسور را تغذیه کند که خود باعث کاهش قابل توجه تعداد اجزای داخلی ریزپردازنده می شود. بعلاوه، در ریزپردازنده هایی که با USB تغذیه می شوند، Precision 32 SiMU1xx USB یک تنظیم کننده ولتاژ از قبل آماده دارد که بتواند برق را بدون معطلی از USB گرفته به تمامی اجزای ریزپردازنده برساند. مدار بازسازی کلاک جهت بالا بردن دقت آن و توانایی استفاده از USB بدون نیاز به کریستال دیگر مزیت این ریزپردازنده به شمار می رود. دیگر ویژگی آن، استفاده از نوعی زنگ برای آگاه کردن کاربر در هنگامی است که اسکن بارکد بطور موفقیت آمیز انجام شده است

(ج)

-جهت انعطاف پذیری بیشتر ریزپردازنده در طرح های مختلف این امکان فراهم آمده تا با معماری Dual-Crossbar ابتدا وسایل جانبی مورد نیاز برای طرح مورد استفاده در ریزپردازنده تعیین شده سپس جاگیری pin ها برای آن وسایل جانبی تعیین شود. به همچنین امکان مسیریابی بهتر بین همان اجزای جانبی و پایه های خروجی این پتانسیل را به ما میدهد که از PCB های کمتری استفاده کنیم.

-امکان on-chip debugging: عیب یابی کردن ریزپردازنده بدون استفاده از منابع دیگر(از قبیل حافظه و رجیسترها)

(د)

32-bit ARM Cortex-M3 CPU

(ه)

لایه دسترسی به سخت افزار یا همان HAL توابع به زبان C را برای تغییر یا خواندن بیت ها جهت برنامه ریزی ریزپردازنده پشتیبانی می کند.

برای برنامه ریزی معماری cross-bar از AppBuilder استفاده می شود تا بدینوسیله دادن مقادیر اولیه به بیت ها و همچنین پیکربندی دستگاه راحت تر صورت پذیرد. نوع clocking مورد استفاده، شخصی سازی کردن pinout ها و تولید کدی که در کامپایلر های متدوال نظیر Kell، IAR و gcc استفاده می شود از ویژگی های این محیط توسعه نرم افزار است.

(و)

جهت کم کردن زیان های زیست محیطی نیاز است تا مصرف انرژی این ریزپردازنده کمینه ممکن باشد. متد مورد استفاده برای بهینه سازی مصرف انرژی بستگی به کاربردی دارد که ریزپردازنده در آن مورد استفاده قرار گرفته است. بعنوان مثال در دستگاه قند خون، با توجه به این که صرفا دفعات کمی در روز از آن استفاده می شود و بیشتر اوقات دستگاه در حالت sleep قرار دارد، مهم ترین مسئله، بهینه سازی مصرف انرژی در حالت sleep می باشد.

از طرفی یک دستگاه سنسور میتواند در حالت sleep باشد ولی نسبت به حالت قبل باید یک مدار کلاک زمان واقعی یا ( ) داشته باشد تا بتواند در بازه های زمانی معین از خواب فعال شود و این امر تا حد امکان به سرعت انجام بگیرد.

و در نهایت در یک دستگاه مورد استفاده در خط تولید کارخانه باید جریان active دستگاه، مینیم توان مصرفی ممکن را به ما بدهد. راهکار های دیگری نیز نظیر کم کردن فرکانس کاری تا حدی که صرفا بتوان آن پردازش و عملیات خاص را انجام داد وجود دارند. نمودار زیر ۳ حالت مختلف مصرف توان برای ریزپردازنده فلان را نشان می دهد:

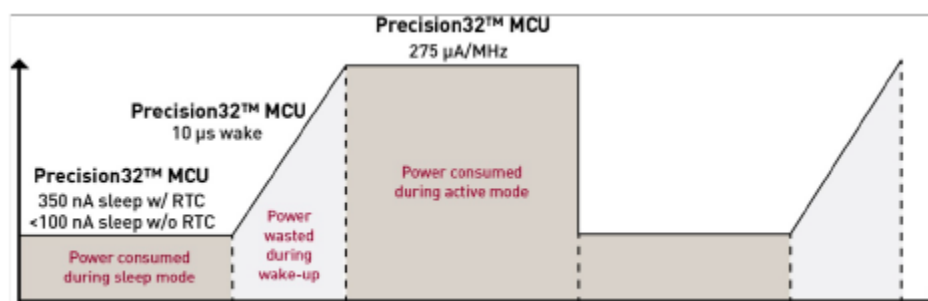


Figure 3: Precision32 MCUs Engineered for Ultra-Low Power in All Modes

(ز)

QFN-TQFP-LGA

(ح)

از منفی ۴۰ تا ۸۵ درجه سانتیگراد

(ط)

در حالت کارکردن regulator، بین ۲٫۷ تا ۵٫۵ ولت

در غیر اینصورت بین ۱٫۸ تا ۳٫۶ ولت

-۲

**SUB AX, 3030H:** MR(opcode + operand read), MR(read 3030H), idle(CPU doing arithmetic)

**MUL BL:** MR(opcode + operand read), idle(CPU doing arithmetic)

**ADD BYTE PTR[BX], CH:** MR(opcode + operand), MR(read content of address SI in memory), MW(write [SI] + CH back in memory)

**POP AX:** symbolic operations:  $AL \leq [SP]$ ;  $AH \leq [SP + 1]$ ;  $SP \leq SP + 2$

MR(opcode + operand), MR(read content of address SP in memory), MR(read content of address SP + 1 in memory), idle(increase SP)

**STC:** MR(opcode), idle

**LSL:** MR(opcode), idle

-۳

**ORG 3030h:** This pseudo-instruction tells assembler to start the machine code of the program from 3030h so it will be set as the origin address of the program stored in RAM.

**EQU True 1:** It defines a constant named 'True' with a corresponding value 1.

**Pressure DW 300:** It defines a variable of type word with 300 (decimal) stored in it

**Length DD 11110000111100001111h:** It defines a variable of type Double-Word(4 bytes = 32 bit) with the corresponding HEX value.

**String DB "CharacterString":** It defines a variable of type byte String, simply a string of bytes, Storing ASCII code of the characters respectively. The order of storing has been stated below.

در ۳ قسمت آخر، نحوه ذخیره سازی Little Endian می باشد بدین ترتیب که چنانچه داده مورد ذخیره سازی را بایت بایت در نظر بگیریم بایت های کم ارزش تر یا به اصطلاح Little End در آدرس های حافظه کمتر قرار می گیرند. بعنوان مثال در مورد Length نحوه ذخیره سازی به این صورت خواهد بود. ستون سمت چپ، آدرس خانه های حافظه متوالی و ستون سمت راست مقادیر محتوای حافظه متناظر با آن آدرس ها میباشند.

Z	0F
Z + 1	0F
Z + 2	0F
Z + 3	00

