نمونه سوالات طبقه‌بندی شده
میکروکنترلرهای AVR

همراه با توضیحات تکمیلی

تالیف: الهه شکیبا

انتشارات پندار پارس

دفتر فروش انتشارات پندار پارس

انقلاب، ابتدای کارگرجنوبی، کوی رشتچی، شماره ۱۴، واحد ۱۶

تلفن: ۶۶۵۷۲۳۳۵ - تلفکس: ۶۶۹۲۶۵۷۸ همراه: ۰۹۲۱۴۳۷۱۹۶۴

خرید آنلاین: **www.pendarepars.com** **info@pendarepars.com**

نام کتاب : **نمونه سوالات طبقه‌بندی شده میکروکنترلرهای AVR، همراه با توضیحات تکمیلی**

ناشر : **انتشارات پندار پارس**

تالیف : **الهه شکیبا**

چاپ نخست : **اردیبهشت ۹۵**

شمارگان : **۵۰۰ نسخه**

چاپ، صحافی : **روز**

قیمت : **۱۹۰۰۰ تومان** **شابک : ۱-۰۲-۸۲۰۱-۶۰۰-۹۷۸**

***هرگونه کپی برداری، تکثیر و چاپ کاغذی یا الکترونیکی از این کتاب بدون اجازه ناشر تخلف بوده و پیگرد قانونی دارد ***

درصد اهمیت فصول

فصل اول: ۱۶.۲٪	فصل ششم: ۳۱.۳٪
فصل دوم: ۱۱.۸٪	فصل هفتم: ۷.۳٪
فصل سوم: ۴.۶٪	فصل هشتم: ۴.۸٪
فصل چهارم: ۷.۳٪	فصل نهم: ۴.۶٪
فصل پنجم: ۷.۳٪	فصل دهم: ۴.۸٪

فصل نخست
ساختار داخلی میکروکنترلرهای AVR

۱-۱ تعاریف اولیه از ساختار میکروکنترلر

۱.حافظه‌های ماندگار چگونه حافظه‌هایی هستند؟

حافظه‌های ماندگار (دائمی) به حافظه‌هایی گفته میشود که بتوانند اطلاعات داده شده را نگه دارند و حتی اگر تغذیه آن‌ها قطع شود نباید این اطلاعات پاک شوند.

۲.انواع حافظه‌های ماندگار را نام ببرید و توضیح دهید.

حافظه PROM: این نوع حافظه فقط خواندنی می‌باشد و تنها می‌تواند یک‌بار برنامه‌ریزی شود.

حافظه EPROM: حافظه فقط خواندنی است که اطلاعات توسط مدار واسط، با ولتاژ الکتریکی نوشته و با نور ماورای بنفش مثل نورخورشید توسط پنجره شیشه‌ای که معمولا با یک برچسب پوشیده می‌شود، پاک می‌گردد.

حافظه EEPROM: حافظه فقط خواندنی است که اطلاعات توسط مدار واسط، با ولتاژ الکتریکی نوشته و پاک می‌شود.

۳.دو مدل از حافظه‌های EEPROM را نام ببرید و توضیح دهید.

الف-موازی (Parallel): آدرس‌دهی، نوشتن و خواندن حافظه به صورت موازی انجام می‌گیرد.

ب-سریال (Serial): این حافظه اغلب در بسته بندی‌های کوچک ۸پایه قرار می‌گیرد و آدرس‌دهی حافظه و آدرس‌دهی سخت‌افزاری و همچنین عمل نوشتن و خواندن به صورت سریال و فقط توسط ۲پایه انجام می‌گیرد.

به این نوع ارتباط دهی، I2C گفته می‌شود.

۴.حافظه‌های فرار، چگونه حافظه‌هایی هستند؟

حافظه‌های فرار (غیردائمی) به حافظه‌هایی گفته می‌شود که بتوانند اطلاعات داده شده را نگه دارند اما برخلاف حافظه‌های ماندگار، اگر تغذیه آنها قطع شود اطلاعات آنها نیز پاک می‌شود.

SRAM: این نوع حافظه از نوع Static یا پایدار است. منظور از پایداری این است که نیازی به تازه‌سازی ندارد وتا وقتی که تغذیه آن برقرار است می‌تواند اطلاعات داده شده را حفظ کند.

DRAM: این نوع حافظه از نوع Dynamic یا غیرپایدار است. یعنی CPU باید مدام با یک فاصله زمانی مشخص، دیتای ذخیره شده در این نوع حافظه را بازسازی کند.

این قبیل از حافظه‌ها با ظرفیت‌های بالا ساخته می‌شوند و بیشتر درکنار ریزپردازنده‌هایی با سرعت بالا قرار می‌گیرند.

۵.منظور از پورت یا درگاه چیست؟

منظور از Port یا درگاه این است که یک سری پایه برای ارتباط با دنیای بیرون تراشه فراهم شده است که هم می‌توانند ورودی و هم خروجی باشند.

هر میکروکنترلر با توجه به نوع بسته‌بندی دارای یک الی چندین پورت است و الزاما یک پورت دارای ۸پایه نیست.

۶.چه زمانی از تراشه‌های بافر استفاده می‌شود؟

تراشه‌هایی به نام بافر وجود دارند که ما در راه‌اندازی جریان بیشتر از ۲۰میلی‌آمپر از آن‌ها استفاده می‌کنیم.

مانند: ULN2003 , 74HC244 , 74HC245

۷.منظور از سیکل ماشین چیست؟*

CPU و سایر قسمت‌های میکروکنترلرها برای انجام هر دستور به میزان خاصی زمان نیاز دارند که به آن سیکل ماشین (پالس ساعت) گفته می‌شود.

(فرکانس اسیلاتور/۱)=سیکل ماشین

*: اسیلاتور: نوسان ساز است.

و به تراشه ای اطلاق می‌شود که وظیفه تولید پالس‌های ساعت را برای سامانه پردازشی برعهده دارد.

تعداد پالس‌هایی که نوسان‌ساز در یک ثانیه تولید می‌کند، فرکانس کاری سامانه پردازشی را مشخص می‌کند.

توضیح کامل: در بخش ۱-۹ کتاب

۸.موارد استفاده از تایمر و کانتر چیست؟

برای زمان‌سنجی دقیق از تایمر و برای شمارش پالس‌های بیرونی از کانتر استفاده می‌کنیم. تایمر پالس ساعت خود را از کلاک سیستم دریافت می‌کند و شروع به شمارش می‌کند اما کانتر، پالس ساعت خود را از پایه Tn دریافت و شمارش می‌کند.

بطور مثال برای ساختن یک ساعت دیجیتال به تایمر و برای شمارش شیشه‌های نوشابه عبوری از جلوی سنسور به کانتر نیاز داریم.

۹.منظور استفاده از وقفه و دلیل استفاده از آن چیست؟*

منظور از وقفه تاخیر زمانی نیست.

وقفه به معنی قطع کردن برنامه جاری و سرویس دادن به تابع وقفه است.

برای اینکه میکروکنترلر بتواند علاوه بر برنامه جاری به سایر قسمت‌ها یا المان دیگری سرویس بدهد باید از وقفه استفاده کنیم.

اجزای جانبی CPU مانند: تایمر و کانتر، مبدل ADC، مقایسه کننده آنالوگ، ارتباط سریال، TWI و ... دارای وقفه مخصوص به خود هستند.

*: وقفه ← Interrupt: دستور قطع اجرای برنامه جاری از سوی یک امکان داخلی یا خارجی متصل به ریزپردازنده به منظور گرفتن سرویس از ریزپردازنده

توضیح کامل: درفصل ۵

مبدل ADC: مبدل آنالوگ به دیجیتال

TWI: ارتباط سریال دوسیمه: یک واسط ارتباطی سریال ساده، قدرتمند و انعطاف‌پذیر است که فقط به دو خط باس (گذرگاه) نیاز دارد. این پروتکل به منظور فراهم آوردن لینک ارتباطی بین مدارات مجتمع ارائه شده است.

توضیح کامل: فصل ۱۰

۱۰.رابطه سریال Usart چیست؟*

این ارتباط‌دهی برای تبادل اطلاعات بین دو سیستم بوده و ممکن است این ارتباط بین دو میکروکنترلر یا یک میکروکنترلر با PC و یا یک میکروکنترلر با هر تراشه دیگری که دارای این پروتکل باشد، برقرار شود.

در این ارتباط‌دهی، اطلاعات ارسالی و دریافتی برروی ۲خط صورت می‌گیرد.

*: ارتباط سریال USART: درکل ارسال و دریافت اطلاعات به دو روش سریال و موازی صورت می‌پذیرد.

USART: Universal Synchronous and Asynchronous Serial Receiver and Transmitter

ارتباط سریال Usart یک ارتباط سریال دوطرفه (Full Duplex) است. به این معنا که هر دستگاه هم می‌تواند فرستنده هم گیرنده باشد و ارسال و دریافت اطلاعات بصورت همزمان صورت پذیرد و شامل ارسال و دریافت سریال سنکرون (همزمان) (USRT) و ارسال دریافت آسنکرون (غیرهمزمان) (UART) می‌باشد.

در روش سنکرون در هر بلوک می‌توان تعدادی داده و یا کاراکتر همراه با یک خط کلاک ارسال کرد. گیرنده نیز می‌تواند داده را برروی یک خط توسط کلاک همزمان کننده فرستنده دریافت کند. (مانند ارسال اطلاعات صفحه کلید و موس)

و در روش آسنکرون در هر ارسال یک داده یا کاراکتر، بدون کلاک همزمان کننده ارسال می‌شود.

۱۱.منظور از مبدل ADC و DAC چیست؟ و علت استفاده از مبدل ADC در میکروکنترلهای AVR چیست؟*

برای تبدیل ولتاژ خوانده شده از یک المان یا یک سنسور، از مبدل آنالوگ به دیجیتال استفاده می‌کنیم.

برخی از میکروکنترلهای AVR دارای مبدل ADC با دقت حداکثر ۱۰بیت هستند.

برای اینکه بتوانیم ولتاژ متغیر در خروجی، توسط میکروکنترلر ایجاد کنیم باید از مبدل دیجیتال به آنالوگ استفاده کنیم. مانند:(تولید موج سینوسی)

لازم به ذکر است که میکروکنترلهای AVR فاقد مبدل DAC هستند اما از ویژگی PWM تایمرهای آن میتوان DAC را شبیه سازی کرد.

*: ADC: تراشه‌ای برای خواندن مقادیر آنالوگ از حسگرهای مختلف متصل به پورت‌های ورودی و تبدیل کردن آن به مقادیر باینری

DAC: تراشه‌ای برای تبدیل کردن بیت‌های یک داده خروجی به مقدار آنالوگ آن

PWM: واژه PWM مخفف (Pulse Width Modulation) یا مدولاسیون پهنای پالس می‌باشد که در برخی مواقع به آن (Pulse Duration Mode) نیز می‌گویند. در این مدولاسیون، پهنای پالس تولیدی را می‌توان تحت کنترل داشت بطوری‌که این پهنا در برخی مواقع می‌تواند متاثر از مقدار دامنه یک موج دیگر باشد.

از دیگر کاربردهای PWM می‌توان به کنترل دور موتورهای DC , AC و منابع تغذیه سوئیچینگ و ... نام برد.

این ویژگی که یکی از حالت‌های کار تایمر میکروکنترلر AVR است، برای کنترل ولتاژ برروی پایه‌های خروجی میکروکنترلر و به‌عبارتی کنترل سرعت موتورهای DC است.

۱۲.منظور از سنسور چیست؟*

المان یا وسیله‌ای که یک پارامتر فیزیکی مانند: دما، فشار، رطوبت، گاز، مادون قرمز، میدان مغناطیسی و ... را به پارامتر الکتریکی (ولتاژ یا جریان) تبدیل می‌کند.

*سنسور ← حس‌گر

۱-۲:مقدمه‌ای بر میکروکنترلها

۱۳. تا قبل از سال ۱۹۷۱ طراح سیستم برای ساخت سیستم خود از چه چیزی استفاده می‌کرد؟

با پیشرفت علم و تکنولوژی در الکترونیک تراشه‌هایی به عنوان میکروپروسسورها طراحی و تولید شدند.

تا قبل از سال ۱۹۷۱ میلادی اگر شخص طراح، سیستمی را می‌خواست طراحی کند باید سیستم مورد نظر خود را به شرکت‌های سازنده میکروپروسسور ارائه می‌داد تا طراحی و ساخته شود و یا اینکه مجبور بود با استفاده از آی‌سی‌های دیجیتالی، سیستم مورد نظر خود را طراحی کند.

۱۴.شرکت Zilog با چه هدفی میکروپروسسور Z80 را وارد بازار کرد؟*

از این پس شرکت‌های سازنده میکروپروسسورها، از جمله شرکت Zilog تصمیم به ساخت میکروپروسسوری گرفت که بتوان آن را در اختیار کاربر قرار داد تا به هر صورت ممکن که می‌خواهد سیستم مورد نظر خود را طراحی کند و به همین دلیل میکروپروسسور Z80 را به بازار عرضه کرد ونرم‌افزار کامپایلر به زبان اسمبلی و پروگرامر آن را نیز ارائه داد.

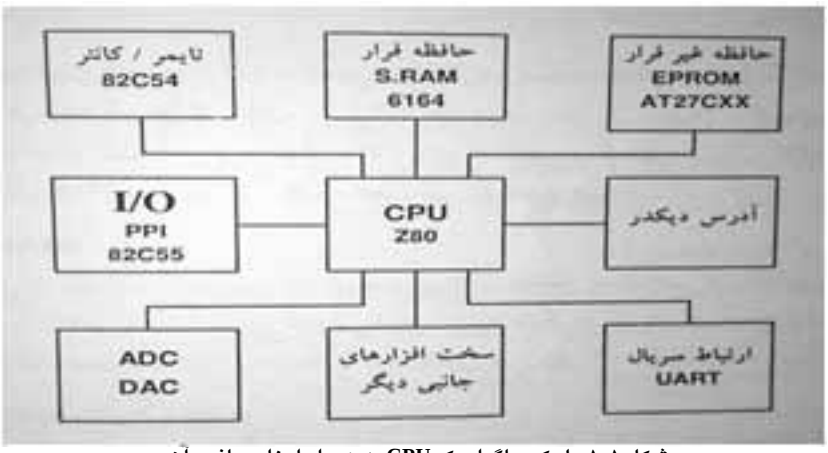
*: کامپایلر: اگر برنامه به یک زبان سطح بالا مثل C و یا Basic نوشته شده باشد، به اسمبلر مربوط به آن کامپایلر گویند.

(اسمبلر: برنامه‌ای که به زبان اسمبلی نوشته شده است، توسط نرم‌افزاری به نام اسمبلر، به رشته‌های صفر و یک قابل فهم توسط ماشین تبدیل می‌شود.)

۱۵.برای استفاده از میکروپروسسور Z80 در یک سیستم چه باید کرد؟

به‌طور کلی اگر یک شخص از میکروپروسسور ۸بیتی Z80 برای سیستمی استفاده کند، باید المان‌های جانبی CPU را نیز علاوه بر سخت‌افزار سیستم موردنظر، در کنار میکروپروسسور Z80 قرار دهد.

همان‌طور که در بلوک دیاگرام شکل ۱-۱ (صفحه بعد) مشاهده می‌نمائید، برای اینکه از یک میکروپروسسور، حتی برای ساده‌ترین سیستم بخواهیم استفاده کنیم، باید از المان‌های جانبی دیگری نیز بهره بگیریم.



شکل ۱-۱: بلوک دیاگرام یک CPU به همراه اجزای جانبی آن

۱۶.معایب استفاده از Z80 به همراه اجزای جانبی چه بود؟

این عمل سبب افزایش قیمت و پیچیده شدن سخت‌افزار پروژه مورد نظر می‌گردد.

۱۷.میکروکنترلرهای خانواده ۸۰۵۱ چه زمانی به بازار وارد شدند و دارای چه امکاناتی بودند؟*

در سال ۱۹۸۱ میلادی شرکت Intel تراشه‌ای را به عنوان میکروکنترلر خانواده ۸۰۵۱ به بازار عرضه کرد.

این میکروکنترلر دارای CPU ۸بیتی، تایمر یا کانتر، وقفه، تبادل سریال، حافظه SRAM و حافظه غیرفرار (FLASH) داخلی می‌باشد.

* حافظه FLASH: کدهای برنامه که توسط پروگرامر برروی میکروکنترلر ریخته شده، در این حافظه قرار می‌گیرد.

این حافظه دارای دو قسمت کاربردی و بارگذاری است.

توضیح کامل: در بخش ۱-۸

۱۸.در اوایل میکروکنترلر ۸۰۵۱ از چه حافظه‌ای استفاده می‌کرد؟

در اوایل میکروکنترلر۸۰۵۱، از نوع حافظه PROM یعنی نوع OTP (One Time Programmable) بهره می‌برد، این نوع میکروکنترلر فقط یک بار قابل برنامه‌ریزی بود.

۱۹.هنگامی که میکرو کنترلر توسعه یافت، از چه حافظه‌ای استفاده کرد و حسن آن چه بود؟

بعدا که این میکروکنترلر توسعه یافت، از حافظه های EPROM استفاده کردند. این نوع میکروکنترلر با شماره 87Cxx آغاز می‌شد و حسن این حافظه در این بود که می‌توانست توسط پنجره‌ی شیشه‌ای که در بالای تراشه قرار داشت، در مجاورت نور ماورای‌بنفش مثل نور خورشید قرار گرفته و بعد از چند دقیقه پاک شود.

۲۰.شرکت سازنده میکروکنترلر ۸۰۵۱ تصمیم گرفت از چه نوع حافظه‌ای بجای EPROM استفاده کند؟

شرکت سازنده میکروکنترلر ۸۰۵۱، تصمیم به استفاده از حافظه‌ای گرفت که بتواند با ولتاژ الکتریکی نوشته و پاک شود (حافظه Flash)، این سری با شماره 89Cxx آغاز می‌شود که امروزه نیز همچنان مورد استفاده قرار می‌گیرد.

۲۱.مدل توسعه یافته میکروکنترلر ۸۰۵۱ چه نام دارد و چه تفاوتی با نوع C دارد؟

شرکت Intel سپس نوع توسعه یافته ۸۰۵۱ را با سری AT89Sxx ارائه کرد.

تفاوت نوع S با نوع C در نحوه‌ی برنامه‌ریزی میکروکنترلر است، زیرا نوع S را می‌توان داخل مدار، پروگرام کرد.

۲۲.پروگرام کردن میکروکنترلر نوع S چگونه انجام می‌شود؟

برای پروگرام کردن نوع S نیازی به خارج کردن میکروکنترلر از مدار نداریم و می‌توانیم در داخل سیستم، توسط یک آی‌سی بافر، عمل پروگرام کردن را از طریق پورت LPT انجام دهیم.

۲۳.خانواده AVR توسط چه شرکتی و در چه سالی وارد بازار شد و AVR مخفف چه کلماتی است؟

سرانجام شرکت Atmel میکروکنترلرهای جدیدی را مانند دیکر شرکتهای سازنده قطعه از جمله شرکت Microchip که میکروکنترلرهای PIC را تولید کرده است، با عنوان خانواده AVR در سال ۱۹۹۷ میلادی به بازار عرضه نموده است. در یک تالار گفتگو به زبان لاتین، برخی بر این باور بودندکه واژه AVR مخفف نام سازندگان اصلی میکروکنترلر از شرکت Atmel با نام‌های Vegard و Alf می‌باشد که حرف R نیز به نوع معماری RISC بکار رفته اشاره دارد. برخی از افراد دیگر، فکر می‌کنند که کلمه AVR مخفف کلمه Advanced Virtual RISC می‌باشد.

البته دقیقا مشخص نیست که AVR مخفف چه کلمه ای می‌باشد. اما ممکن است فقط یک نماد تجاری شرکت Atmel باشد.

۲۴.خانواده AVR به چند سری تقسیم می‌شوند و قابلیت‌های آن چیست؟

خانواده AVR به سه سری تقسیم می‌شوند:

الف-سری AT90S

ب-سری ATtiny

ج-سری ATmega

میکروکنترلرهای AVR قابلیت‌های بیشتری نسبت به میکروکنترلر خانواده ۸۰۵۱ دارند و از نسل جدیدی از حافظه‌های Flash و معماری RISC بهره می‌گیرند.

۲۵.هریک از سری‌های خانواده AVR را به اختصار توضیح دهید.*

الف- سریAT90S : این سری اعضای کلاسیک خانواده AVR را تشکیل می‌دهند و قابلیت‌های کمتری نسبت به دو سری بعدی دارند و کمتر استفاده می‌شوند.

ب- سری ATtiny : این میکروکنترلرها در ابعاد کوچک ۸، ۲۰و ۲۸ پایه هستند و قابلیت‌های خوبی نسبت به سری اول دارند و بیشتر در سیستم‌هایی که نیاز به پورت بالا نیست استفاده می‌شوند. یکی از اعضای ۸پایه این سری ATtiny85 است که دارای امکانات خوبی از جمله مبدل ADC می‌باشد.

ج- سری ATmega : این سری از میکروکنترلرهای AVR امکانات بیشتری نسبت به دو سری قبلی دارند و توجه مخاطبان را به خود جلب نموده‌اند. میکروکنترلهای ATmega128,ATmega32,ATmega8 پرکاربردهای این سری هستند.

(انواع این ۳سری در جداول ۱-۱ و ۱-۲ و ۱-۳ در صفحه بعد ذکر شده‌اند.)

AT90S4434	AT90S8535	AT90S4433	AT90S2323	AT90S1200
AT90S8534	AT90S4414	AT90S8515	AT90S2343	AT90S2313

جدول ۱-۱: میکروکنترلرهای سری AT90S

ATtiny10	ATtiny12	ATtiny15	ATtiny25	ATtiny28	ATtiny85
ATtiny11	ATtiny13	ATtiny22	ATtiny26	ATtiny45	ATtiny2313

جدول ۱-۲: میکروکنترلرهای سری ATtiny

ATmega48	ATmega603	ATmega165	ATmega324
ATmega8	ATmega649	ATmega168	ATmega325

ATmega88	ATmega6490	ATmega169	ATmega329
ATmega64	ATmega16	ATmega128	ATmega3290
ATmega640	ATmega161	ATmega1280	ATmega3250
ATmega644	ATmega162	ATmega1281	ATmega256
ATmega6450	ATmega163	ATmega32	ATmega8532
ATmega670	ATmega164	ATmega323	ATmega8515

جدول ۱-۳: میکروکنترلرهای سری ATmega

*: AT90S: یکی از توانائی‌های خانواده کلاسیک AT90S افزایش حافظه برنامه، حافظه داده و ... است.

ATtiny: بخاطر کوچک بودن و داشتن Cpu قدرتمند، کاربرد فراوانی دارد و برای انجام عملیات ساده، طراحی شده و در مدارهایی که نیاز به حجم کوچک است، بکار گرفته می‌شود.

امکانات جانبی: ADC ۱۰ بیتی، مقایسه کننده آنالوگ، تایمر و کانتر با خروجی PWM، واسط ارتباط سریال USART , 2WIRE و تایمر نگهبان Watch dog

ATmega: علاوه بر دارا بودن توانائی‌های دو سری قبلی، امکانات بیشتری را در خود جای داده و لذا، قدرتمندترین اعضای خانواده AVR محسوب می‌شود و همچنان در حال پیشرفت و گسترش هستند.

۲۶.آیا امکانش وجود دارد که بجای میکروکنترلر ۸۰۵۱ از میکروکنترلر AVR استفاده کنیم؟*

از آنجا که اصل سازگاری از طرف شرکت‌های سازنده قطعه رعایت می‌شود، میکروکنترلهای AVR تمام قابلیت‌های میکروکنترلر ۸۰۵۱ را دارند و از امکانات جدیدی از جمله EEPROM,SPI,TWI,ADC و ... برخوردار هستند.

اما شرکت Atmel میکروکنترلری از سری AVR را طراحی کرده است که بتواند جایگزین مداراتی شود که در آنها میکروکنترلر ۸۰۵۱ بکار رفته است.

بطور مثال میکروکنترلر ATmega8515 پایه‌هایی شبیه به AT89C52 دارد تا بتوان به سادگی آن را در برد میکروکنترلر قدیمی جایگزین کرد.

*: SPI: ارتباط جانبی سریال و یک استاندارد سریال سنکرون است که برای فواصل کوتاه و با سرعت بالا مناسب است.
ADC: مبدل آنالوگ به دیجیتال
TWI: ارتباط سریال دوسیمه
EEPROM: حافظه ماندگار

۲۷.دو نوع معماری برای ساخت میکروکنترلرها را نام برده و توضیح دهید.*

بطور کلی دو نوع معماری برای ساخت میکروکنترلها وجود دارد:

الف-معماری CISC: (Complex Instruction Set Computer)

تاریخچه این نوع معماری به قبل ازسال ۱۹۸۰م برمی‌گردد. در اکثر میکروپروسسورها و میکروکنترلرهای قدیمی از این نوع معماری استفاده شده است. در این معماری تعداد دستورات بیشتر و پیچیده‌تر است اما برنامه‌نویسی آن بخصوص اسمبلی

ساده‌تر شده است و از طرفی سرعت اجرایی دستورات پایین‌تر است.

ب-معماری RISC: (Reduced Instruction Set Computer)

بعد از سال ۱۹۸۰م معماری جدیدی طراحی شد. در این نوع معماری تعداد دستورات کاهش پیدا کرد و ازطرفی سرعت اجرایی دستورات ۱۰برابر نسبت به معماری قبلی افزایش یافت و برنامه‌نویسی به زبان اسمبلی را قدری پیچیده و سخت کرد اما با وجود ساختار بهینه شده میکروکنترلرهای AVR با حافظه‌های ظرفیت بالا و همچنین استفاده از معماری RISC امکان برنامه‌نویسی به زبان‌های سطح بالاتر مانند C و بیسیک فراهم گردید.

*: CISC: کامپیوتر با مجموعه دستورات پیچیده
RISC: کامپیوتر با مجموعه دستورات کاهش یافته

۲۸.تفاوت‌های معماری RISC با معماری CISC را بیان کنید.*

الف-تعداد و اندازه دستورات در RISC کمتر از CISC است در نتیجه سرعت RISC بیشتر است.

ب-در معماری RISC انتقال داده از رجیستر به رجستر و یا حافظه به حافظه صورت نمی‌گیرد.

ج-تعداد رجیسترها در معماری RISC بیشتر است.

د-برنامه‌نویسی به زبان اسمبلی در معماری RISC پیچیده تر از CISC است.

ه-اکثر دستورات در معماری RISC در یک کلاک سیکل اجرا می‌شوند.

و-مصرف توان معماری CISC بیشتر از RISC است.

ی-برنامه های MSDOS در معماری RISC قابل اجرا نیست. همین دلیل باعث شده است تا اکثر میکروپروسسورها (نظیر 80X86 ساخت شرکت Intel) از معماری CISC استفاده کنند.

*: تفاوت ب و ج ← در RISC همه AVRها دارای ۳۲ رجیستر همه منظوره از R0 تا R31 بوده که بطور مستقیم با واحد ALU (واحد محاسبه و منطق) در ارتباط هستند.

تفاوت ه ← و تعداد کمی از دستورالعمل ها در دو پالس ساعت انجام می‌شوند.

۲۹.تفاوت‌های میکروپروسسورها و میکروکنترلرها را توضیح دهید.*

میکروپروسسورها فقط یک پردازنده کوچک هستند و باید المان‌های جانبی نظیر ROM,RAM،تایمر یا کانتر،وقفه و سایر المان‌های جانبی درکنار میکروپروسسور قرار داده شود تا بتوان یک سیستم چند منظوره را طراحی کرد. بطورمثال CPU کامپیوتر یک میکروپروسسور است و PC یک سیستم چندمنظوره است که می‌تواند چندین عمل را اجرا کند. اما میکروکنترلرها دارای المان‌های جانبی محدود در یک بسته‌بندی نظیر FLASH،SRAM،تایمر یا کانتر،وقفه و... هستند و فقط می‌توانند در سیستم‌های تک‌منظوره استفاده شوند. میکروکنترلر بمعنی کنترل‌کننده کوچک است. بطورمثال کنترل دمای یک دستگاه صنعتی توسط میکروکنترلر، یک سیستم تک‌منظوره است. میکروپروسسورها از معماری CISC استفاده می‌کنند اما میکروکنترلرهای پیشرفته ازجمله AVR از معماری RISC استفاده می‌کنند. البته میکروکنترلرهای قدیمی‌تر از جمله ۸۰۵۱ از معماری CISC استفاده می‌کند و اساسی‌ترین تفاوت میکروپروسسورها انعطاف‌پذیری آن‌هاست که میتوان RAMو ROM آن‌ها را افزایش داد اما در میکروکنترلرها میزان ظرفیت حافظه‌ها ثابت است.

*: تفاوت‌های میکروپروسسور و میکروکنترلر:

۱.معنا ۲.المان‌های جانبی ۳.چندمنظوره یا تک‌منظوره بودن

۴.معماری ۵.انعطاف‌پذیری آن‌ها

میکروپروسسور	میکروکنترلر
فقط یک ریزپردازنده کوچک است و نیاز به المان‌های جانبی دارد. مثل: CPU کامپیوتر	دارای المان‌های جانبی است که قابلیت استفاده در سیستم‌های تک‌منظوره را دارد. مثل کنترل دمای یک دستگاه صنعتی
از معماری CISC استفاده می‌کند.	از معماری RISC استفاده می‌کند.
انعطاف‌پذیری بالایی دارند. چون RAM و ROM آنها قابل افزایش است.	انعطاف‌پذیری کمتری دارند چرا که میزان حافظه، ثابت است.

۳۰.آیا می‌توان در حالت کلی میکروکنترلرها را با هم مقایسه کرد؟

اینکه بگوئیم چه میکروکنترلری قوی‌تر و بهتر است، بسیار سخت است و نمی‌توان در حالت کلی میکروکنترلرها را با هم مقایسه کرد.

اما بطور مثال میکروکنترلر PIC به شماره PIC18F452 را می‌توان با میکروکنترلر AVR به شماره ATmega32 مقایسه کرد. قابلیت‌های این دو میکروکنترلر بسیار شبیه هم است.

برای اینکه قضاوت صحیح‌تری داشته باشیم بهتر است ویژگی‌های این دو میکروکنترلر را از برگه اطلاعاتی آنها مطالعه کنیم.

۱-۳: خصوصیات میکروکنترلر ATmega16

۳۱.خصوصیات میکروکنترلر ATmega16 را نام ببرید.*

& قابلیت اجرایی بالا و توان مصرفی پایین

& معماری پیشرفته RISC

- ۱۳۱ دستورالعمل قدرتمند که تنها در یک کلاک سیکل اجرا می‌شوند.
- دارای ۳۲ رجیستر ۸ بیتی همه‌منظوره
- سرعتی حداکثر تا ۱۶ مگاهرتز برای نوع ATmega16

& حافظه غیرفرار برنامه و دیتا

- 16k بایت حافظه Flash با قابلیت خود برنامه‌ریزی

تحمل حافظه Flash: قابلیت ۱۰۰۰۰بار نوشتن و پاک کردن

- ۵۱۲ بایت حافظه EEPROM داخلی قابل برنامه‌ریزی

تحمل حافظه EEPROM: قابلیت ۱۰۰۰۰۰بار نوشتن و پاک کردن

- 1k بایت حافظه SRAM داخلی

- قفل برنامه Flash و EEPROM جهت حفاظت از برنامه نوشته شده

& قابلیت ارتباط دهی JTAG (IEEE Std.)

- حمایت از اشکال زدایی تراشه داخل سیستم

- قابلیت برنامه‌ریزی Lock bits,fuse Bits,EEPROM,FLASH ازطریقJTAG

& ویژگی‌های جانبی

- دو تایمر یا کانتر ۸بیتی با مقسم فرکانسی مجزا و مد مقایسه‌ای

- یک تایمر یا کانتر ۱۶بیتی با مقسم فرکانسی مجزا و مد مقایسه‌ای و مد Capture

- RTC با اسیلاتور مجزا

- دارای ۴ کانال PWM

- ۸ کانال مبدل آنالوگ به دیجیتال ۱۰بیتی

- ۸ کانال Single-ended (سیگنالی که نسبت به زمین سنجیده می‌شود.)

دارای ۷ کانال تفاضلی فقط برای بسته‌بندی نوع TQFP

دارای ۲ کانال تفاضلی با قابلیت برنامه‌ریزی گین 1x , 10x , 200x

- ارتباط سریال دو سیمه (Tow Wire)

- USART سریال قابل برنامه‌ریزی

- ارتباط سریال SPI به‌صورت master/slave

- دارای تایمر Watchdog قابل برنامه‌ریزی با اسیلاتور مجزای داخلی

- مقایسه‌کننده آنالوگ داخلی

& ویژگی‌های خاص میکروکنترلر

- Reset فعال درموقع وصل تغذیه با قابلیت برنامه‌ریزی آشکارساز Brown-Out
- دارای اسیلاتور RC کالیبره شده داخلی
- دارای منابع وقفه داخلی و خارجی
- دارای شش مد Sleep: Idle, ADC Noise Reduction, Power-save, Power-down, standby Extended Standby

& پایه‌های ورودی و خروجی و نوع بسته‌بندی

- ۳۲ خط ورودی و خروجی قابل برنامه‌ریزی
- ۴۰ پایه PDIP، ۴۴ پایه TQFP و ۴۴ پایه MLF

& ولتاژهای عملیاتی

- 2.7v تا 5.5v برای ATmega16L
- 4.5v تا 5.5v برای ATmega16

& فرکانس‌های کاری

- 0 تا 8MHZ برای ATmega16L
- 0 تا 16MHZ برای ATmega16

& مصرف توان با شرائط 1MHZ,3v,25C برای ATmega16L

- در حالت فعال 1.1mA
- در مد توان Idle: 0.35mA
- در مد توان Power-down: کمتر از 1MA

*: SRAM: حافظه فرار پایدار

JTAG: یک استاندارد از طرف موسسه استاندارد IEEE رابط JTAG است:

(Joint Test Access Group)

رابطی است برای تست و برنامه‌ریزی ICهای دیجیتال که طبق استاندار

IEEE 11491 مورد استفاده قرار می‌گیرد و بطور گسترده در ICاهای دیجیتال، بکار گرفته می‌شود. از طریق این ارتباطدهی استاندارد، با فعال کردن فیوزبیت JTAGEN می‌توانیم میکروکنترلر را برنامه‌ریزی کنیم.

مد مقایسه ای (CTC): در این حالت محتوای رجیستر تایمر، دائما توسط سخت‌افزار با رجیستر مقایسه‌ای تایمر مقایسه می‌شود و هرموقع این برابری رخ می‌دهد، محتوای رجیستر تایمر، صفر می‌شود.

توضیح کامل: در فصل ۶

مد CAPTURE (تسخیر یا ضبط کننده): تایمر یک می‌تواند توسط واحد Capture سیگنال‌های خارجی را تشخیص دهد و از زمان جاری با سرعت بالا نمونه بگیرد.

توضیح کامل: در فصل ۶

RTC: تایمر دو دارای یک عملکرد غیرهمزمان است و می‌تواند به طور مجزا با یک اسیلاتور پالس ساعت کار کند. به این عملکرد RTC گفته می‌شود. کریستال پالس ساعت واقعی RTC تایمر دو، مجزا از کریستال کلاک سیستم می‌باشد و بهینه شده و نیازی به قرار دادن خازن‌های بالانس (حذف کننده نویز) نمی‌باشد.

توضیح کامل: فصل ۶

۴کانال PWM: یک حالت کار سریع (تک شیب) برای تعمیر

نوع TQFP: یکی از نوع‌های بسته‌بندی میکروکنترلر AVR است که ۴۴ پایه دارد.

برنامه‌ریزی گین: به عنوان برنامه‌های کاربردی کامپیوتر و به منظور کاهش دستگاه‌های سخت‌افزاری، میتوان از گین‌های تقویت‌کننده قابل برنامه‌ریزی استفاده کرد.

تایمر Watchdog: تایمر نگهبان زبان: میکروکنترلرهای AVR دارای یک تایمر نگهبان زمان هستند، وظیفه این تایمر در صورت فعال شدن توسط برنامه‌نویسی، این است که از عملکرد صحیح برنامه میکروکنترلر می‌توان اطمینان حاصل کرد.

توضیح کامل: در بخش ۱-۱۲

مقایسه کننده آنالوگ داخلی: یک ویژگی میکروکنترلر AVR است که از کاربردهای آن می‌توان به شارژ اتوماتیک باتری، اندازه گیری خازن و مقاومت، تبدیل خروجی زوج مادون قرمز به صفر و یک دیجیتال و ... اشاره کرد.

توضیح کامل: در فصل ۸

RESET: در واقع به نوعی یک وقفه است که می‌تواند توسط هریک از منابع تحریک کننده آن رخ دهد. هنگامی که RESET رخ می‌دهد تمامی رجیسترهای ورودی و خروجی و همچنین دیگر رجیسترهای کنترلی با توجه به مقادیر پیش فرض در نظر گرفته شده، تنظیم می‌شوند.

توضیح کامل: در بخش ۱-۱۱

مدار Brown-out: آشکارساز ولتاژ تغذیه است.

توضیح کامل: در بخش ۱-۱۱

اسیلاتور RC کالیبره شده داخلی: می‌تواند برخی فرکانس‌های ثابت را در شرائط تغذیه و دمای مشخص ایجاد نماید. فرکانس کاری این اسیلاتور به شدت ولتاژ تغذیه و درجه حرارت محیط وابسته می‌باشد.

توضیح کامل: در بخش ۱-۹

مدهای Sleep (خواب): هریک از مدها در عملکردهای متفاوتی کاربرد دارند. هدف از مدهای توان، کاهش توان مصرفی میکروکنترلر است.

توضیح کامل: در بخش۱-۱۰

MLF , TQFP , PDIP: انواع بسته‌بندی میکروکنترلر AVR

انواع دیگر: SSQP , PLCC , SOIC

ATmega16L: میکروکنترلر ATmega16 با توان کمتر

مدهای توان: از انواع مدهای Sleep

۱-۴: فیوزبیت‌های میکروکنترلر ATmega16

۳۲.ویژگی‌های فیوزبیت‌های میکروکنترلر ATmega16 را بیان کنید.*

فضای اختصاص داده شده به فیوزبیت‌ها، از نوع حافظه ماندگار است.

فیوزبیت‌ها برای تنظیمات خاصی استفاده می‌شوند و با پاک کردن میکروکنترلر از بین نمی‌روند و تغییر آنها فقط از طریق پروگرامر امکان پذیر است و برای تنظیم آنها نیاز به برنامه خاصی نداریم و موقع پروگرام کردن توسط ابزار نرم افزار Code Vision AVR آنها را تنظیم و برنامه‌ریزی می‌کنیم.

فیوزبیت‌ها با ۰ برنامه‌ریزی و با ۱ غیرفعال می‌شوند.

برنامه‌ریزی فیوزبیت‌ها باید قبل از قفل کردن تراشه صورت گیرد.

میکروکنترلهای AVR بسته به نوع قابلیتی که دارند، دارای فیوزبیت‌های متفاوتی هستند. (مشخصات فیوزبیت‌های ATmega16 در جداول ۱-۴ و ۱-۵ در صفحه بعد ذکر شده‌اند.)

جدول ۱-۴: بایت بالایی فیوزبیت‌های ATmega16

Fuse High Byte	Bit NO	Description	Default Value
OCDEN(۴)	۷	Enable OCD	۱ (unprogrammed, OCD disabled)
JTAGEN	۶	Enable JTAG	۰ (programmed, JTAG enabled)
SPIEN(۱)	۵	Enable SPI Serial Program and Data Downloading	۰ (programmed, SPI prog enabled)
CKOPT(۲)	۴	Oscillator options	۱ (unprogrammed)
EESAVE	۳	EEPROM memory is preserved through the Chip Erase	۱ (unprogrammed, EEPROM not preserved)
BOOTSZ۱	۲	Select Boot Size (see Table ۱۰۰ for details)	۰ (programmed)(۳)
BOOTSZ۰	۱	Select Boot Size (see Table ۱۰۰ for details)	۰ (programmed)(۳)
BOOTRST	۰	Select reset vector	۱ (unprogrammed)

جدول ۱-۵: بایت پائینی فیوزبیت‌های ATmega16

Fuse LOW Byte	Bit NO	Description	Default Value
BODLEVEL	۷	Brown-out Detector trigger level	۱ (unprogrammed)
BODEN	۶	Brown-out Detector enable	۱ (unprogrammed, BOD disabled)
SUT۱	۵	Select start-up time	۱ (unprogrammed)(۱)
SUT۰	۴	Select start-up time	۰ (programmed)(۱)
CKSEL۳	۳	Select Clock source	۰ (programmed)(۲)
CKSEL۲	۲	Select Clock source	۰ (programmed)(۲)
CKSEL۱	۱	Select Clock source	۰ (programmed)(۲)
CKSEL۰	۰	Select Clock source	۱ (unprogrammed)(۲)

*: CodevisionAVR: یکی از بهترین و قوی‌ترین کامپایلرهای زبان C برای میکروکنترلرهای AVR

توضیح کامل: در فصل ۳

۳۳.با فعال کردن فیوزبیت OCDEN ، چه کاری می‌توان انجام داد؟

موقعیکه فیوزبیت ارتباطدهی JTAG فعال شده باشد و برنامه میکروکنترلر را قفل نکرده باشیم، می‌توانیم با فعال کردن فیوزبیت OCDEN برنامه میکروکنترلر را بطور آنلاین درحین اجرا توسط مدار واسط که از ارتباط سریال JTAG استفاده می‌کند و توسط نرم‌افزار AVR Studio مشاهده کنیم.

به این نوع آنالیز، اِمولاتور(Emulator) یا شبیه‌سازی سخت‌افزاری گفته می‌شود.

فعال کردن این فیوزبیت مصرف توان میکروکنترلر را افزایش می‌دهد.

این فیوزبیت به‌طور پیش‌فرض، غیرفعال است.

۳۴.با فعال کردن JTAGEN، چه کاری می‌توان انجام داد؟

با فعال کردن این فیوزبیت، می‌توان میکروکنترلر را از طریق ارتباطدهی استاندارد JTAG برنامه‌ریزی کرد.

این فیوزبیت به‌طور پیش‌فرض، فعال است.

۳۵.آیا پایه‌های ارتباطدهی JTAG، همیشه می‌تواند فعال باقی بماند؟

&& چون پایه‌های ارتباطدهی JTAG در میکروکنترلر ATmege16 بر روی PC2 تا PC5 قرار دارد در زمانی که ما از این ارتباطدهی استفاده نمی‌کنیم، آن را غیرفعال کنیم، در غیر اینصورت نمی‌توانیم از پایه‌های PC2 تا PC5 استفاده کنیم.

۳۶.با فعال کردن فیوزبیت SPIEN چه کاری می‌توان انجام داد؟

اگر این فیوزبیت فعال باشد، می‌توان میکروکنترلر را از طریق SPI برنامه‌ریزی کرد.

این فیوزبیت در نرم افزار Code Vision AVR قابل دسترس نیست.

فیوزبیت SPIEN بطور پیش‌فرض فعال است.

۳۷.با فعال کردن فیوزبیت CKOPT چه کاری می‌توان انجام داد؟*

با فعال کردن این فیوزبیت، می‌توانیم از حداکثر دامنه نوسان اسیلاتور خارجی استفاده کنیم.

اگر این فیوزبیت فعال باشد، خروجی اسیلاتور بصورت Rail-to-Rail کار می‌کند. یعنی دامنه نوسانات برابر تغذیه میکروکنترلر می‌شود. این حالت در مکان‌هایی که نویز زیادی دارند موثر است، اما از طرفی باعث افزایش توان مصرفی در میکروکنترلر می‌شود.

این فیوزبیت بطور پیش‌فرض غیرفعال است.

*Rail-to-Rail: مثلا اگر منبع تغذیه ۵ ولت باشد، حداکثر دامنه پالس ساعت نیز ۵ ولت می‌شود.

۳۸.با فعال کردن فیوزبیت EESAVE چه کاری می‌توان انجام داد؟

در موقعیکه ما میکروکنترلر را پاک می‌کنیم، EEPROM نیز پاک می‌شود.

اگر بخواهیم در موقع پاک شدن حافظه Flash از محتوای حافظه EEPROM محافظت کنیم، باید این فیوزبیت را فعال کنیم.

این فیوزبیت بطور پیش فرض غیرفعال است.

۳۹.با فعال کردن فیوزبیت‌های BOOTSZ0 , BOOTSZ1 چه کاری می‌توان انجام داد؟*

این دو فیوزبیت میزان حافظه اختصاص داده شده BOOT را تعیین می‌کنند و برنامه‌ریزی آن‌ها طبق جدول ۱-۶ (در صفحه بعد) می‌باشد.

به‌طور پیش‌فرض هردو فیوزبیت فعال هستند.

Bootsz1	Bootsz0	Boot size	Pages	Application Flash Section	Boot loader Flash section	End Application Section	Boot Reset Address (start boot loader section)
۱	۱	۱۹۸ words	۴	$۰۰۰۰ - $۱F۷F	$۱F۸۰ - $۱FFF	$۱F۷F	$۱F۸۰
۱	۰	۳۹۶ words	۸	$۰۰۰۰ - $۱EFF	$۱F۰۰ - $۱FFF	$۱EFF	$۱F۰۰
۰	۱	۷۱۲ words	۱۶	$۰۰۰۰ - $۱DFF	$۱E۰۰ - $۱FFF	$۱DFF	$۱E۰۰
۰	۰	۱۰۲۴ words	۳۲	$۰۰۰۰ - $۱BFF	$۱C۰۰ - $۱FFF	$۱BFF	$۱C۰۰

جدول ۱-۶: تعیین ظرفیت حافظه Boot

*: BOOT: قسمتی از حافظه FLASH بوده و در آن تکه برنامه ای به نام Boot loader قرار می‌گیرد که از آن در موارد اشکال‌زدائی استفاده می‌شود و در طول اجرای برنامه قابل تغییر است.

Application: قسمتی از حافظه FLASH بوده و در آن برنامه کاربردی قرار می‌گیرد و در طول اجرای برنامه نمی‌توان در آن چیزی نوشت.

۴۰.با فعال کردن فیوزبیت BOOTRST چه کاری می‌توان انجام داد؟*

این فیوزبیت برای انتخاب بردار Reset است. اگر غیرفعال باشد بردار Reset از آدرس 0x0000 حافظه خواهد بود اما اگر این فیوزبیت فعال شود به ابتدای آدرسی که فیوزبیت‌های BOOTSZ1 و BOOTSZ0 طبق جدول ۱-۶ تعیین کرده‌اند، پرش می‌کند.

این فیوزبیت به‌طور پیش‌فرض غیرفعال است.

❋: RESET: بعد از RESET یک مدت زمان کوتاه، به اندازه زمان تعیین شده، توسط Start-up طول می‌کشد، سپس بعد از پایداری نوسان‌ساز، برنامه از بردار RESET آدرس 0x0000 شروع می‌شود و در صورتی که فیوزبیت BootRST فعال شود و بردار RESET را توسط فیوزبیت‌های BootSZ0 , BootSZ1 تعیین کرده باشیم، آنگاه بردار RESET از قسمت Boot حافظه FLASH داخلی (طبق ستون آخر جدول ۱-۶) آغاز می‌شود. توضیح کامل: در بخش ۱-۱۱

۴۱.با فعال کردن فیوزبیت BOODEN چه کاری می‌توان انجام داد؟❋

برای فعال کردن مدار Brown-out باید این فیوزبیت فعال شود.

مدار Brown-out آشکارساز ولتاژ تغذیه است که اگر از ۲.۷ یا ۴ ولت کمتر شود، میکروکنترلر را Reset می‌کند.

این فیوزبیت به‌طور پیش‌فرض غیرفعال است.

❋: مدار آشکارساز ولتاژBrown-out فعال می‌شود. ➔BODEN=0

۴۲.با فعال کردن فیوزبیت BODLEVEL چه کاری می‌توان انجام داد؟❋

اگر فیوزبیت BODEN فعال شده باشد و فیوزبیت BODLEVEL برنامه‌ریزی نشده باشد، آنگاه با کاهش ولتاژ VCC کمتر از 2.7v میکروکنترلر Reset می‌شود و اگر فیوزبیت BODLEVEL فعال شود، آنگاه با کاهش ولتاژ VCC کمتر از 4v میکروکنترلر Reset می‌شود.

این فیوزبیت به‌طور پیش‌فرض غیرفعال است.

❋: فیوزبیت BODEN فعال شده باشد. ← مقدار این فیوزبیت صفر باشد.

۴۳.با فعال کردن فیوزبیت‌های SUT1,SUT0 چه کاری می‌توان انجام داد؟❋

این دو فیوزبیت، زمان شروع (Start-up) را در موقع وصل تغذیه طبق جدول ۱-۷ (در صفحه بعد) تعیین می‌کنند.

به‌طور پیش‌فرض SUT0 فعال و SUT1 غیرفعال است.

CKSEL0	SUT1..0	Start-up Time from Power-down and	Additional Delay From reset	Recommended Usage

		Power-save	(Vcc=5.0v)	
0	00	258 CK (1)	4.1 ms	Ceramic resonator, fast rising power
0	01	258 CK (1)	65 ms	Ceramic resonator, slowly rising power
0	10	1K CK (2)	-	Ceramic resonator, BOD enabled
0	11	1K CK (2)	4.1 ms	Ceramic resonator, fast rising power
1	00	1K CK (2)	65 ms	Ceramic resonator, slowly rising power
1	01	16K CK	-	Crystal Oscillator, BOD enabled
1	10	16K CK	4.1 ms	Crystal Oscillator, fast rising power
1	11	16K CK	65 ms	Crystal Oscillator, slowly rising power

جدول۱-۷: تنظیم فیوزبیت‌های Start-up

*: Start-up: مدت زمان شروع کار میکروکنترلر پس از بازنشانی: هنگامیکه CPU از مد Power-down یا Power-save (از مدهای Sleep) خارج می‌شود، پالس ساعت انتخاب شده با کمی تاخیر به میکروکنترلر اعمال می‌شود. این تاخیر را زمان Start-up مینامند، که به منظور پایدار شدن پالس ساعت قبل از شروع به کار است.

طبق جدول ۱-۷ فیوزبیت CKSEL0 به همراه فیوزبیت‌های SUT0 , SUT1 برای تعیین زمان Start-up استفاده می‌شود.

کریستال: یک اسیلاتور الکترونیکی است. قطعه ای با دو پایه که با قرار گرفتن در مدار، سیگنال الکترونیکی با فرکانس بسیار دقیق تولید می‌کند.

۴۴.با فعال کردن فیوزبیت‌های CKSEL3,CKSEL2,CKSEL1,CKSEL0 چه کاری می‌توان انجام داد؟*

توسط این فیوزبیت‌ها نوع و مقدار فرکانس اسیلاتور را تعیین می‌کنیم. بطور پیش‌فرض فیوزبیتCKSEL0 غیرفعال و بقیه فعال هستند یعنی فرکانس1MHZ داخلی انتخاب شده است. اگر بخواهیم فرکانس کاری اسیلاتور داخلی را تنظیم کنیم این فیوزبیت‌ها را طبق جدول۱-۸ (صفحه بعد) تنظیم می‌کنیم و اگر بخواهیم ازکریستال خارجی استفاده کنیم باید این فیوزبیت‌ها را طبق جدول۱-۹ (صفحه بعد) درحالت یک یعنی غیرفعال قرار دهیم. در تنظیم این فیوزبیت‌ها باید دقت نمود. بطورمثال اگر اشتباها تمام این فیوزبیت‌ها را فعال کنیم طبق جدول۱-۹ (صفحه بعد) مد کلاک خارجی انتخاب می‌شود که در این حالت میکروکنترلر نه با نوسان‌ساز داخلی و نه با کریستال خارجی کار نمی‌کند بلکه توسط کلاک خارجی که به پایهXTAL1 اعمال می‌شود، کارمی‌کند. همچنین اگر ازکریستال خارجی برای میکروکنترلر خود استفاده نماییم، باید حتما موقع پروگرامر کردن نیز کریستال به میکروکنترلر وصل باشد ولی درحالت استفاده از نوسان‌ساز داخلی نیازی به قراردادن کریستال بیرونی نداریم.

CKSEL 3..0	Nominal Frequency (MHz)
0001(1)	1.0
0010	2.0
0011	4.0
0100	8.0

میکروکنترلر با مقدار پیش فرض 1MHZ تنظیم شده است.

جدول ۱-۸: تنظیم نوسان‌ساز کالیبره شده داخلی

Device Clocking Option	CKSEL 3..0
External Crystal/Ceramic Resonator	1111 – 1010
External Low-frequency Crystal	1001

External RC Oscillator	1000 – 0101
Calibrated Internal RC Oscillator	0100 – 0001
External Clock	0000

جدول ۱-۹: تعیین منبع کلاک سیستم

*: جدول ۱-۸ ← کالیبره شده: برای کنترل سرعت و دقت پارامترهای دستگاه‌ها

جدول ۱-۹← اسیلاتور RC خارجی: از اسیلاتور با RC خارجی در کاربردهایی که به تغییرات زمان و فرکانس حساسیت نداشته باشیم، استفاده می‌کنیم.

توضیح کامل: در بخش ۱-۹ کتاب

۴۵.اگر فیوزبیت‌ها را اشتباه تنظیم کردیم، روش رفع خطا چگونه است؟

اگر بطور تصادفی فیوزبیت‌ها را اشتباه تنظیم کردید و با قرار دادن کریستال خارجی، میکروکنترلر توسط پروگرامر شناسایی نشد، یک فرکانس 1MHZ توسط یک میکروکنترلر دیگری به پایه XTAL1 میکروکنترلر مذکور اعمال کنید و توسط پروگرامر، فیوزبیت‌ها را صحیح تنظیم نمائید.

۱-۵: پورت‌های ورودی و خروجی میکروکنترلر ATmega16

۴۶.تعداد پایه‌ی پورت‌های ورودی و خروجی میکروکنترلرهای AVR برچه اساسی انتخاب می‌گردد؟

هر میکروکنترلر AVR بسته به نوع قابلیت و بسته‌بندی که دارد، دارای یک سری پایه‌های ورودی و خروجی است.

بطور مثال پورت C میکروکنترلر ATmega8 دارای ۵ پایه می‌باشد.

۴۷.در حالت اولیه Reset، ورودی و خروجی‌های AVR در چه حالتی قرار می‌گیرند؟*

پورت‌های تمام میکروکنترلرهای AVR می‌توانند به صورت ورودی و خروجی عمل کنند.

در حالت اولیه Reset تمام ورودی و خروجی‌ها در حالت Tri-state قرار می‌گیرند.

Tri-state یعنی حالتی که پایه پورت امپدانس بالا می‌باشد.

*Tri-state ← سه حالته

۴۸.در چه حالتی مقاومت بالاکش (Pull-up) فعال می‌شود؟*

تمامی پایه‌های پورت‌ها مجهز به مقاومت بالاکش (Pull-up) داخلی هستندکه می‌توانند در حالت ورودی فعال شوند.

*: Pull-up: مقاومت بالاکش را می‌توان به دلخواه فعال یا غیرفعال کرد.

تمامی پایه‌های پورت‌ها مجهز به مقاومت بالاکش داخلی هستند که می‌توانند در حالت ورودی فعال شوند.

هنگامی‌که پایه بصورت ورودی تعریف می‌شود، در صورتیکه مقاومت بالاکش داخلی فعال نباشد، مقدار خوانده شده از ورودی پایه، قابل اطمینان نمی‌باشد.

بنابراین توصیه می‌شود، هنگام ورودی کردن پایه، مقاومت بالاکش داخلی یا خارجی برای آن در نظر گرفته شود.

۴۹.یک بافر Latch داخلی میکروکنترلرهای AVR، تا چه جریانی را می‌تواند تامین کند؟

از آن‌جایی که جریان‌دهی پورت‌های میکروکنترلرهای قدیمی نمی‌توانستند حتی یک LED را روشن کنند، شرکت‌های سازنده میکروکنترلرهای جدید سعی کرده‌اند که جریان‌دهی پایه‌های پورت‌ها را افزایش دهند.

بافر Latch داخلی میکروکنترلرهای AVR می‌تواند در حالت جریان‌دهی (Source) و جریان‌کشی (Sink) در حالت فعال، جریانی تا 20mA را تامین کند.

البته در حالت حداکثر می‌تواند جریان 40mA را تحمل کند.

بنابراین به راحتی می‌تواند یک LED و یا سون‌سگمنت را جریان‌دهی کند.

۵۰.پورت‌های میکروکنترلرهای AVR دارای چند رجیستر تنظیم‌کننده هستند؟ این رجیسترها را نام برده و به اختصار توضیح دهید.*

بطور کلی تمامی پورت‌های میکروکنترلرهای AVR دارای سه رجیستر تنظیم‌کننده به فرم زیر هستند: