

جهت خرید فایل word به سایت www.kandooen.com مراجعه کنید
یا با شماره های ۰۹۳۶۶۰۲۷۴۱۷ و ۰۹۳۶۶۴۰۶۸۵۷ و ۰۶۶۴۱۲۶۰-۰۵۱۱ تماس حاصل نمایید

طراحی تایمر دیجیتالی

چکیده:

تایمر دیجیتالی که در این پروژه طراحی شده است و معرفی می گردد دارای مشخصات زیر است:

- نمایش مراحل برنامه بر روی سون سگمنت (۲۶ مرحله).
 - حفظ مرحله برنامه در هنگام قطع برق با استفاده از باتری BACKUP.
 - انتخاب شروع از هر مرحله برنامه با استفاده از کلیدهای PROGRAM.
 - کوچک بودن حجم مدار نسبت به نمونه های مشابه دیجیتالی.
- اصولاً تایمر برای شمارش اتفاقات بکار می رود. و تعداد خاصی از این اتفاقات برای ما اهمیت دارد تا در این زمانهای خاص به یک دستگاه فرمان روشن یا خاموش بودن را بدهیم. دراصل تایمر دیجیتالی یک شمارنده است که تعداد پالسهای ورودی را بصورت باینری می شمارد و اگر ما از میان این اعداد موردنظر خودمان را به وسیله یک دیکودر، دیکودر کنیم، به راحتی می توانیم به تعدادی خروجی فرمان دهیم.

مقدمه:

در عصری که ما در آن زندگی می کنیم، علم الکترونیک یکی از اساسی ترین و کاربردی ترین علمی است که در تکنولوژی پیشرفته امروزه نقش مهمی را ایفا می کند.

الکترونیک دیجیتال یکی از شاخه های علم الکترونیک است که منطق زیبای آن انسان را مجذوب خود می کند.

امروزه اکثر سیستمهای الکترونیکی به سمت دیجیتال سوق پیدا کرده است و این امر به علت مزایای زیادی است که سیستمهای دیجیتال نسبت به مدارهای آنالوگ دارند.

مداری که در این پروژه معرفی می گردد یک مدار فرمان میکرویی است که به منظور جایگزینی برای نمونه مکانیکی آن طراحی گردیده است.

برای طراحی و ساخت یک تایمر ماشین لباسشویی، قبل از هر چیز باید ماشین لباسشویی، طرز کار و همچنین عملکرد قسمت های مختلف آن را بشناسیم. برای این منظور در ابتدای به

شرح قسمت های مختلف آن می پردازیم:

اجزای زیر قسمت های مختلف یک ماشین لباسشویی را تشکیل می دهند:

موتور، پمپ تخلیه، المنت گرمکن، شیر برقی، اتوماتیک دما، هیدرو سوئیچ و تایمر.

اگر بخواهیم عملکرد ماشین لباسشویی را بطور خلاصه بیان کنیم، به این صورت است که

ابتدا شیر آب (شیر برقی) باز شده و آب مخزن را پر می کند. سپس در صورت نیاز، گرمکن

آب مخزن را به گرمای مجاز می رساند. سپس موتور شروع به چرخاندن لباسهای کثیف می کند. سپس پمپ، آب کثیف را از مخزن به بیرون از ماشین پمپ می کند. این سلسله عملیات ادامه دارد تا در انتها ماشین بطور اتوماتیک خاموش شده و متصدی دستگاه می تواند لباسهای شسته شده را از دستگاه خارج کند. فرمان تمام اجزای فوق را تایمر می دهد. برای آشنایی با تایمر مکانیکی، مختصری در مورد آن توضیح می دهیم:

این تایمر به این صورت عمل می کند که یک موتور الکتریکی کوچک، یک محور را توسط چرخ دنده هایی می چرخاند و این محور یک سری دیسک های پلاستیکی هم محور را می چرخاند. این دیسک ها بر روی خود دارای برجستگی هایی است و بر روی این برجستگی ها زائده هایی قرار می گیرند که با چرخیدن دیسک، این زائده ها بالا و پایین رفته و پلاتین هایی را بازوبسته می کنند. و این پلاتین ها نیز به نوبه خود یک سری اتصال های الکتریکی قطع و وصل می شوند که می توانند به عنوان فرمان های الکتریکی قسمتهای مختلف لباسشویی به کار روند. شکل زیر نحوه عملکرد این نوع تایمر را نشان می دهد:

تایمرهای مکانیکی دارای عیوب و مزایایی هستند که در زیر به آنها اشاره می شود:
بسیار گران هستند، استفاده از این نوع تایمر باعث پیچیدگی سیم کشی داخل ماشین لباسشویی می شود، بر اثر کارکرد پلاتین های آن اکسیده شده و به خوبی عمل نمی کند.

از مزایای مهم تایمر مکانیکی می توان نویزپذیر نبودن آن را نام برد. قبل از تشریح مدار تایمر دیجیتالی و عملکرد آن، ابتدا کمی در مورد دو عنصر هیدروسوئیچ و اتوماتیک دما که در تمام ماشین های لباسشویی وجود دارد (و کمتر در دستگاههای الکتریکی دیده می شود) توضیح می دهیم:

تایمرهای لباسشویی یک سری مشخصات عمومی دارند که برای همه انواع آن صادق است.

این مشخصات به قرار زیر است:

- نشان دادن مرحله برنامه در هر لحظه.

- حفظ مرحله برنامه در هنگام قطع برق.

- انتخاب شروع برنامه از هر مرحله دلخواه.

- خاموش کردن لباسشویی پس از اتمام به صورت اتوماتیک.

هیدروسوئیچ که مخفف سوئیچ هیدرولیکی است یک عنصر مکانیکی است که پر بودن یا

خالی بودن مخزن لباسشویی از آب را، تشخیص می دهد.

این عنصر از یک مخزن کوچک تشکیل شده که داخل آن یک دیافراگم قرار دارد. این

مخزن دارای یک ورودی هوا است. وقتی هوا تحت فشار معینی به داخل آن برسد، دیافراگم

به جلو حرکت کرده و یک اتصال الکتریکی را قطع و یا وصل می کند.

جهت خرید فایل word به سایت www.kandooen.com مراجعه کنید
یا با شماره های ۰۹۳۶۶۰۲۷۴۱۷ و ۰۹۳۶۶۴۰۶۸۵۷ و ۰۶۶۴۱۲۶۰-۰۵۱۱ تماس حاصل نمایید

علت استفاده از هیدروسوئیچ در ماشین لباسشویی یکی به این دلیل است که وقتی شیربرقی آب را باز کرده و آب وارد مخزن لباسشویی می شود، پس از رسیدن حجم آب بیش از حد مجاز وارد مخزن شود.

دلیل دیگر استفاده از هیدروسوئیچ، وابسته نبودن حجم آب پر شده درون مخزن، به فشار آب ورودی است. اتوماتیک دما هم یک نوع ترموستات الکتریکی است که با قطع و وصل به موقع المنت گرمکن، دمای آب مخزن لباسشویی را طبق انتخاب ما ثابت نگه می دارد.

مدار تغذیه:

در شکل نمای کلی از مدار تغذیه به کار برده شده در این پروژه را نشان می دهیم. که آن را به اختصار شرح می دهیم.

باتری V_1 ولتاژ کمتری نسبت به V_2 دارد پس D_2 هدایت کرده و روشن است و D_1 خاموش است. ما در اینجا از رگولاتور (7805) استفاده کرده ایم که ولتاژ ورودی آن بین 6 تا 10 و کاهنده می باشد که 5 ولت خروجی دارد.

در اینجا به خاطر رسیدن به 5 ولت از $I_c(7805)$ استفاده می کنیم.

مدار داخلی (7805):

یک مدار کلکتور مشترک است که تقویت ولتاژ ندارد و تقویت جریان دارد.

علت استفاده از دیود D_1 در مدار تغذیه:

اگر D_1 در مدار نباشد باتری 9 ولت همیشه در مدار است اما اگر D_1 در مدار باشد وقتی

باتری 9 ولت وارد مدار می شود که ولتاژ تغذیه شهر قطع شود.

علت استفاده از D_2 : برای اینکه ولتاژی از باتری به منبع تغذیه نرود.

مدار تشخیص قطع و وصل بودن برق شهر:

۱- نحوه قرار گرفتن پایه های رگولاتور به صورت زیر است:

۲- مقاومت های بایاس ترانزیستور با مقادیر مشخص شده به کار رفته اند.

۳- علت استفاده از خازن C_1 : یک صافی است، برای اینکه روی میکرو پارازیت نیافتد.

این مدار به منظور رساندن پیامی به میکرو در مدار قراردادده شده تا میکرو را از وضعیت برق شهر مطلع کند.

این مدار یک ولتاژ نمونه از منبع تغذیه اصلی دریافت کرده و اگر جریان برق شهر برقرار باشد خروجی این مدار صفر و در غیراین صورت خروجی مدار ۱ می باشد. که میکرو از روی این اختلاف ولتاژ به بودن یا نبودن برق شهر پی می برد.

این مدار تغذیه دارای یک مدار فرمان است که این مدار فرمان به میکرو متصل می باشد. تا

زمانی که برق شهر رفت، به میکرو فرمان دهد که تمام خروجی ها را خاموش کند.

این مدار تغذیه ۲ ورودی دارد که درحالت seven segment دستگاه خاموش میشود، و میکرو به حالت استندبای می رود.

«مدار قدرت»

این مدار، مدار اپتو کوپلر (بایاس ترایاک) است.

اپتوکوپلرها برای ایزوله کردن مدار فرمان از مدار قدرت بکار می روند به این ترتیب که

فرمان گیت ترایاک توسط یک LED به آن اعمال می شود. بین LED و ترایاک هیچ پایه

مشترکی وجود ندارد.

در صورت مستقیم وصل کردن مدار فرمان به مدار قدرت علاوه بر این اشکالات نويز باعث برق دار شدن مدار فرمان می شود.

برای برطرف کردن این اشکال 2 راه وجود دارد. ۱- استفاده از ترانس پالس ۲- اپتو کوپلر در روش ترانس پالس، به وسیله یک ترانس پالس مدار فرمان از قدرت جدا می شود. به این ترتیب که با اعمال پالس از طرف مدار فرمان در سر دیگر ترانس پالس یک پالس مربعی ایجاد می شود که از آن می توان برای فرمان دادن مدارات قدرت استفاده کرد.

۱- ترانزیستور: از خروجی میکرو جریان کمی می گذرد به خاطر تقویت جریان برای رسیدن به ورودی IC opto IC استفاده می شود.

* مدار پیشنهادی برای راه اندازی تریاک (opto copler) IC توسط اپتو کوپلر

«مدار سنسور آب»:

در این مدار از زوج دارلینگتون استفاده شده برای اینکه ضریب تقویت بالا رود. برای سنس کردن سطح آب می توان از مدار زیر استفاده کرد به این ترتیب که چون آب عنصر خالص نیست پس دارای مقاومتی می باشد. که حدود $300k\pi$ است.

در شرایط عادی وقتی ۰ که بین 2 پایه قطع ما هیچ مقاومتی وجود ندارد ترانزیستور Q_1 و Q_2 که به صورت زوج دارلینگتون برای بالا بردن ضریب تقویت بسته شده و در حالت قطع می باشد. در این حالت خروجی مدار یک ۱ می باشد اما اگر مقاومتی بین دو پایه قطع قرار گیرد: ترانزیستورهای Q_1 و Q_2 به حالت اشباع رفته در این حالت ولتاژ خروجی مدار صفر می شود.

مقاومت R_1 به منظور حفاظت ترانزیستورها از اتصال کوتاه دوسر سنسور به کار برده شده، به این ترتیب که اگر این مقاومت در مدار نباشد هنگام اتصال کوتاه شدن دوسر سنسور تمام ولتاژ دوسر بیس-امتیر ترانزیستورها خواهد افتاد که موجب سوختن و از بین رفتن آنها می شود.

مقاومت R_2 به منظور بایاس ترانزیستورها و کنترل حساسیت مدار به کار رفته است به این ترتیب که هرچقدر مقدار مقاومت R_2 بیشتر باشد حساسیت مدار بیشتر شده و برعکس.

اگر مقاومت R_2 از مدار خارج شود سنسور ما با اشاره دست فعال می شود مانند

(سوئیچهای finger tuch) ...

IC (ADC0804):

یک IC آنالوگ به دیجیتال است که ولتاژ را به کد دیجیتال تبدیل می کند.

سنسور دما یک IC سه پایه است که به ازای هر درجه حرارت 10mv خروجی دارد.

پورت های P_0 مقاومت های بالای IC پورت صفر احتیاج به مقاومت Pull Up دارد که از

مقاومت 10k برای این کار استفاده می شود.

پایه Reset شماره 9 می باشد که به هنگام روشن شدن میکرو آن را Reset می کند و باعث

می شود که برنامه ما از ابتدا اجرا می شود. این مقاومت با یک خازن یا (یک میکرو) سری

شده و به V_{CC} متصل شده است.

کریستال 12mHz میکرو، برای تولید پالس بکار می رود. کلاک میکرو می باشد. P0.0

برای خروجی کلاک برای ATOD می باشد.

مدار متصل به پایه 9 میکرو به صورت زیر است که در آن خازن (میکروفاراد) به V_{CC}

متصل است و مقاومت به کار برده شده 10k بوده و مدار Reset سخت افزاری میکرو می

باشد.

«آشنایی با میکروکنترلرها»

گرچه کامپیوترها تنها چند دهه ای است که با ما همراهند، با این حال تأثیر عمیق آنها بر زندگی ما با تأثیر تلفن، اتومبیل و تلویزیون رقابت می کند. همگی ما حضور آنها را احساس می کنیم، چه برنامه نویسان کامپیوتر و چه دریافت کنندگان صورت حسابهای ماهیانه که توسط سیستمهای کامپیوتری بزرگ چاپ شده و توسط پست تحویل داده می شود. تصور ما از کامپیوتر معمولاً «داده پردازی» است که محاسبات عددی را بطور خستگی ناپذیری انجام می دهد.

ما با انواع گوناگونی از کامپیوترها برخورد می کنیم که وظایفشان را زیرکانه و بطرزی آرام، کارا و حتی فروتنانه انجام می دهند و حتی حضور آنها اغلب احساس نمی شود. ما کامپیوترها را به عنوان جزء مرکزی بسیاری از فرآورده های صنعتی و مصرفی از جمله، در سوپرمارکت ها داخل صندوق های پول و ترازوها، درخانه، دراجاق ها، ماشین های لباسشویی، ساعت های دارای سیستم خبردهنده و ترموستات ها، در وسایل سرگرمی همچون اسباب بازی ها، VCR ها، تجهیزات استریو و وسایل صوتی، در محل کار در ماشین های تایپ و فتوکپی، و در تجهیزات صنعتی مثل مته های فشاری و دستگاههای حروفچینی نوری می یابیم. در این مجموعه ها کامپیوترها وظیفه «کنترل» را در ارتباط با «دنیای واقعی»، برای روشن و خاموش کردن وسایل و نظارت بر وضعیت آنها انجام می

دهند. میکرو کنترلرها (برخلاف میکرو کامپیوترها و ریزپردازنده ها) اغلب در چنین کاربردهایی یافت می شوند. با وجود این که بیش از بیست سال از تولد ریزپردازنده نمی گذرد، تصور وسایل الکترونیکی و اسباب بازیهای امروزی بدون آن کار مشکلی است. در ۱۹۷۱ شرکت اینتل، 8080 را به عنوان اولین ریزپردازنده موفق عرضه کرد. مدت کوتاهی پس از آن، موتور رولا، RCA و سپس MOS Technology و Zilog انواع مشابهی را به ترتیب به نامهای 6800، 1801، 6502، Z80 عرضه کردند. گرچه این مدارهای مجتمع (IC) ها به خودی خود فایده چندانی نداشتند اما به عنوان بخشی از یک کامپیوتر تک بورد (SBC)، به جزء مرکزی فرآورده های مفیدی برای آموزش طراحی با ریزپردازنده ها تبدیل شدند. از این SBC ها که سرعت به آزمایشگاههای طراحی در کالج ها، دانشگاهها و شرکت های الکترونیک راه پیدا کردند می توان برای نمونه از D2 موتور رولا، KIM-1 ساخت MOS Technolog و SDK-85 متعلق به شرکت اینتل نام برد.

میکرو کنترلر قطعه ای شبیه به ریزپردازنده است. در ۱۹۷۶ اینتل 8748 را به عنوان اولین قطعه خانواده میکرو کنترلرهای MCS-48TM معرفی کرد. 8748 با ۱۷۰۰۰ ترانزیستور در یک مدار مجتمع، شامل یک CPU، ۱ کیلو بایت EPROM، ۶۴ بایت RAM، ۲۷ پایه I/O و یک تایمر ۸ بیتی بود. این IC و دیگر اعضای MCS-48TM که پس از آن آمدند، خیلی زود به یک استاندارد صنعتی در کاربردهای کنترل گرا تبدیل شدند جایگزین کردن اجزاء

الکترومکانیکی در فرآورده هایی مثل ماشین های لباسشویی و چراغ های راهنمایی از ابتدای کار، یک کاربرد مورد توجه برای این میکروکنترلرها بودند و همین طور باقی ماندند. دیگر فرآورده هایی که در آنها میتوان میکروکنترلر را یافت عبارتند از اتومبیل ها، تجهیزات صنعتی، وسایل سرگرمی و ابزارهای جانبی کامپیوتر. (افرادی که یک IBM PC دارند کافی است به داخل صفحه کلید نگاه کنند تا مثالی از یک میکروکنترلر را در یک طراحی با کمترین اجزاء ممکن ببینند).

توان، ابعاد و پیچیدگی میکروکنترلرها با اعلام ساخت 8051، یعنی اولین عضو خانواده میکروکنترلرهای MCS-51^{MT} در ۱۹۸۰ توسط اینتل پیشرفت چشمگیری کرد. در مقایسه با 8048 این قطعه شامل بیش از ۶۰۰۰۰ ترانزیستور، ۴ K بایت ROM، ۱۲۸ بایت RAM، ۳۲ خط I/O، یک درگاه سریال و دو تایمر ۱۶ بیتی است. که از لحاظ مدارات داخلی برای یک IC بسیار قابل ملاحظه است، (شکل ۱-۱ را ببینید). امروزه انواع گوناگونی از این IC وجود دارند که به صورت مجازی این مشخصات را دوبرابر کرده اند. شرکت زیمنس که دومین تولید کننده قطعات MCS-51TM است SAB80515 را به عنوان یک 8051 توسعه یافته در یک بسته ۶۸ پایه با شش درگاه I/O ۸ بیتی، ۱۳ منبع وقفه، و یک مبدل آنالوگ به دیجیتال با ۸ کانال ورودی عرضه کرده است. خانواده 8051 به عنوان یکی از جامعترین و

قدرتمندترین میکروکنترلرهای ۸ بیتی شناخته شده و جایگاهش را به عنوان یک میکروکنترلر مهم برای سال های آینده یافته است.

این کتاب درباره خانواده میکروکنترلرهای MSC-51TM نوشته شده است. فصل های بعدی معماری سخت افزار و نرم افزار خانواده MCS-51TM را معرفی می کنند و از طریق مثالهای طراحی متعدد نشان می دهند که چگونه اعضای این خانواده می توانند در طراحی های الکترونیکی با کمترین اجزاء اضافی ممکن شرکت داشته باشند.

در بخشهای بعدی از طریق یک آشنایی مختصر با معماری کامپیوتر، یک واژگان کاری از اختصارات و کلمات فنی که در این زمینه متداولند (واغلب باهم اشتباه می شوند) را ایجاد خواهیم کرد. از آن جا که بسیاری اصطلاحات در نتیجه تعصب شرکت های بزرگ و سلیقه مؤلفان مختلف دچار ابهام شده اند، روش کار ما در این زمینه بیشتر عملی خواهد بود تا آکادمیک. هر اصطلاح در متداول ترین حالت با یک توضیح ساده معرفی شده است.

۲-۱ اصطلاحات فنی

یک کامپیوتر توسط دو ویژگی کلیدی تعریف می شود: (۱) داشتن قابلیت برنامه ریزی برای کار کردن روی داده بدون مداخله انسان و (۲) توانایی ذخیره و بازیابی داده. عموماً یک سیستم کامپیوتری شامل ابزارهای جانبی برای ارتباط با انسان ها به علاوه برنامه

هایی برای پردازش داده نیز می باشد. تجهیزات کامپیوتر سخت افزار، و برنامه های آن نرم افزار نام دارند.

یک سیستم کامپیوتری شامل یک واحد پردازش مرکزی (CPU) است که از طریق گذرگاه آدرس، گذرگاه داده و گذرگاه کنترل به حافظه قابل دستیابی تصادفی (RAM) و حافظه فقط خواندنی (ROM) متصل می باشد. مدارهای واسطه گذرگاه های سیستم را به وسایل جانبی متصل می کنند. حال اجازه بدهید تا هریک از اینها را بطور مفصل بررسی کنیم.

۳-۱ واحد پردازش مرکزی

CPU، به عنوان «مغز» سیستم کامپیوتری، تمامی فعالیتهای سیستم را اداره کرده و همه عملیات روی داده را انجام می دهد. اندیشه اسرارآمیز بودن CPU در اغلب موارد نادرست است زیرا این تراشه فقط مجموعه ای از مدارهای منطقی است که بطور مداوم دو عمل را انجام می دهند: واکنشی دستورالعمل ها، و اجرای آنها. CPU توانایی درک و اجرای دستورالعمل ها را براساس مجموعه ای از کدهای دودویی دارد که هریک از این کدها نشان دهنده یک عمل ساده است. این دستورالعمل ها معمولاً حسابی (جمع، تفریق، ضرب و تقسیم)، منطقی (AND , OR , NOT و غیره)، انتقال داده یا عملیات انشعاب هستند و با مجموعه ای از کدهای دودویی با نام مجموعه دستورالعمل ها نشان داده می شوند.

ثبات دستورالعمل (IR) کد دودویی هر دستورالعمل را در حال اجرا نگه می دارد و شمارنده

برنامه (PC) آدرس حافظه دستورالعمل بعدی را که باید اجرا شود نشان می دهد.

واکشی یک دستورالعمل از RAM سیستم یکی از اساسی ترین اعمالی است که توسط

CPU انجام می شود و شامل این مراحل است: (الف) محتویات شمارنده برنامه در گذرگاه

آدرس قرار می گیرد (ب) یک سیگنال کنترل READ فعال میشود (پ) داده (کد عملیاتی

دستورالعمل) از RAM خوانده می شود و روی گذرگاه داده قرار می گیرد (ت) کد

عملیاتی در ثبات داخلی دستورالعمل CPU انجام می شود و (ث) شمارنده برنامه یک واحد

افزایش می یابد تا برای واکشی بعدی از حافظه آماده شود. مرحله اجرا مستلزم رمزگشایی

کد عملیاتی و ایجاد سیگنالهای کنترلی برای گشودن ثبات های درونی به داخل و خارج از

ALU است. همچنین باید به ALU برای انجام عملیات مشخص شده فرمانی داده شود.

بعلا تنوع زیاد عملیات ممکن، این توضیحات تاحدی سطحی می باشند و در یک عملیات

ساده مثل «افزایش یک واحدی ثبات» مصداق دارند. دستورالعمل های پیچیده تر نیاز به

مراحل بیشتری مثل خواندن بایت دوم و سوم به عنوان داده برای عملیات دارند.

یک سری از دستورالعمل ها که برای انجام یک وظیفه معنادار ترکیب شوند برنامه یا نرم

افزار نامیده می شود، و نکته واقعا اسرارآمیز در همین جا نهفته است. معیار اندازه گیری برای

انجام درست وظایف، بیشتر کیفیت نرم افزار است تا توانایی تحلیل CPU. سپس برنامه ها

CPU را «راه اندازی» می کنند و هنگام این کار آنها گهگاه به تقلید از نقطه ضعف های نویسندگان خود، اشتباه هم می کنند. عباراتی نظیر « کامپیوتر اشتباه کرد» گمراه کننده هستند. اگرچه خرابی تجهیزات غیر قابل اجتناب است اما اشتباه در نتایج معمولاً نشانی از برنامه های ضعیف یا خطای کاربر می باشد.

حافظه نیمه رسانا : RAM و ROM

برنامه ها و داده در حافظه ذخیره می شوند. حافظه های کامپیوتر بسیار متنوعند و اجزای همراه آنها بسیار، و تکنولوژی بطور دائم و پی در پی موانع را برطرف می کند، بگونه ای که اطلاع از جدیدترین پیشرفتهای نیاز به مطالعه جامع و مداوم دارد. حافظه هایی که بطور مستقیم توسط CPU قابل دستیابی می باشند، IC های (مدارهای مجتمع) نیمه رسانایی هستند که RAM و ROM نامیده می شوند. دو ویژگی RAM و ROM را از هم متمایز میسازد: اول آن که RAM حافظه خواندنی / نوشتنی است درحالی که ROM حافظه فقط خواندنی است و دوم آن که RAM فرار است (یعنی محتویات آن هنگام نبود ولتاژ تغذیه پاک می شود) درحالی که ROM غیر فرار می باشد.

اغلب سیستمهای کامپیوتری یک دیسک درایو و مقدار اندکی ROM دارند که برای نگهداری روال های نرم افزاری کوتاه که دائم مورد استفاده قرار می گیرند و عملیات ورودی / خروجی را انجام می دهند کافی است. برنامه های کاربران و داده، روی دیسک

ذخیره می گردند و برای اجرا به داخل RAM بار می شوند. با کاهش مداوم در قیمت هر بایت RAM، سیستمهای کامپیوتری کوچک اغلب شامل میلیونها بایت RAM می باشند.

گذرگاهها: آدرس، داده و کنترل

یک گذرگاه عبارت است از مجموعه ای از سیم ها که اطلاعات را با یک هدف مشترک حمل می کنند. امکان دستیابی به مدارات اطراف CPU توسط سه گذرگاه فراهم می شود: گذرگاه آدرس، گذرگاه داده و گذرگاه کنترل. برای هر عمل خواندن یا نوشتن، CPU موقعیت داده (یا دستورالعمل) را با قراردادن یک آدرس روی گذرگاه آدرس مشخص می کند و سپس سیگنالی را روی گذرگاه کنترل فعال می نماید تا نشان دهد که عمل موردنظر خواندن است یا نوشتن. عمل خواندن، یک بایت داده را از مکان مشخص شده در حافظه برمی دارد و روی گذرگاه داده قرار می دهد. CPU داده را می خواند و دریکی از ثبات های داخلی خود قرار می دهد. برای عمل نوشتن CPU داده را روی گذرگاه داده می گذارد. حافظه، تحت تأثیر سیگنال کنترل، عملیات را بعنوان یک سیکل نوشتن، تشخیص می دهد و داده را در مکان مشخص شده ذخیره می کند.

اغلب، کامپیوترهای کوچک ۱۶ یا ۲۰ خط آدرس دارند. با داشتن n خط آدرس که هر یک می توانند در وضعیت بالا (1) یا پایین (0) باشند، 2^n مکان قابل دستیابی است. بنابراین یک

گذرگاه آدرس ۱۶ بیتی می تواند به $2^{16} = 65536$ مکان، دسترسی داشته باشد و برای یک

آدرس ۲۰ بیتی $2^{20} = 1048576$ مکان قابل دستیابی است. علامت اختصاری K (برای کیلو)

نمایند $1024 = 2^{10}$ می باشد، بنابراین ۱۶ بیت می تواند $64K = 2^6 \times 2^{10}$ مکان را

آدرس دهی کند درحالی که ۲۰ بیت می تواند $1024K = 2^{10} \times 2^{10}$ (یا Meg

۱) را آدرس دهی نماید.

گذرگاه داده اطلاعات را بین CPU و حافظه یا بین CPU و قطعات I/O منتقل می کند.

تحقیقات دامنه داری که برای تعیین نوع فعالیت‌هایی که زمان ارزشمند اجرای دستورالعملها را

در یک کامپیوتر صرف می کنند، انجام شده است نشان می دهد که کامپیوترها دوسوم

وقتشان را خیلی ساده صرف جابجایی داده می کنند. از آن جا که عمده عملیات جابجایی

بین یک ثبات CPU و RAM یا ROM خارجی انجام می شود تعداد خطهای (یا پهنای)

گذرگاه داده در کارکرد کلی کامپیوتر اهمیت شایانی دارد. این محدودیت پهنای، یک تنگنا

به شمار می رود: ممکن است مقادیر فراوانی حافظه در سیستم وجود داشته باشد و CPU از

طریق گذرگاه داده - توسط پهنای گذرگاه داده محدود می شود. به علت اهمیت این

ویژگی، معمول است که یک پیشوند را که نشان دهنده اندازه این محدودیت است اضافه

می کنند. عبارت «کامپیوتر ۱۶ بیتی» به کامپیوتری با ۱۶ خط در گذرگاه داده اشاره می کند.

اغلب کامپیوترها در طبقه بندی ۴ بیت، ۸ بیت، ۱۶ بیت یا ۳۲ بیت قرار می گیرند و توان محاسباتی کلی آنها با افزایش پهنای گذرگاه داده، افزایش می یابد.

توجه داشته باشید که گذرگاه داده همانطور که در شکل ۱-۲ نشان داده شده است، یک گذرگاه دوطرفه و گذرگاه آدرس، یک گذرگاه یک طرفه می باشد. اطلاعات آدرس همیشه توسط CPU فراهم می شود (همانطور که در شکل ۱-۲ با فلش نشان داده شده است). درحالی که داده ممکن است در هر جهت، بسته به اینکه عملیات خواندن موردنظر باشد یا نوشتن، جابجا شود. همچنین توجه داشته باشید که عبارت «داده» در مفهوم کلی بکار رفته است یعنی اطلاعاتی که روی گذرگاه داده جابجا میشود و ممکن است دستورالعمل های یک برنامه، آدرس ضمیمه شده به یک دستورالعمل یا داده مورد استفاده توسط برنامه باشد.

گذرگاه کنترل ترکیب درهمی از سیگنال ها است، که هر یک نقش خاصی در کنترل منظم فعالیت های سیستم دارند. بعنوان یک قاعده کلی، سیگنال های کنترل سیگنال های زمانبندی هستند که توسط CPU برای همزمان کردن جابجایی اطلاعات روی گذرگاه آدرس و داده ایجاد می شوند. اگرچه معمولاً سه سیگنال مثل CLOCK، READ و WRITE وجود دارد، برای انتقال اساسی داده بین CPU و حافظه، نام و عملکرد این سیگنال ها بطور کامل

بستگی به نوع CPU دارد. برای جزئیات بیشتر در این موارد باید به برگه اطلاعات سازندگان مراجعه کرد.

ابزارهای ورودی / خروجی

ابزارهای I/O یا «ابزارهای جانبی کامپیوتر» مسیری برای ارتباط بین سیستم کامپیوتری و

«دنای واقعی» فراهم می کنند. بدون ابزارهای جانبی، سیستمهای کامپیوتری به ماشینهای

درون گرایی تبدیل می شوند که استفاده ای برای کاربران خود ندارند. سه دسته از ابزارهای

I/O عبارتند از ابزارهای ذخیره سازی انبوه، ابزارهای رابط با انسان و ابزارهای کنترل /

نظارت.

ابزارهای ذخیره سازی انبوه

ابزارهای ذخیره سازی انبوه نیز مثل RAM ها و ROM های نیمه رسانا جزو نقش آفرینان

عرصه تکنولوژی حافظه هستند که بطور دائم در حال رشد و بهبود است. آنچنان که از نام

آنها برمی آید این ابزارها مقادیر معتناهی اطلاعات (برنامه یا داده) را نگهداری می کنند و

این حجم از اطلاعات به هیچ وجه در RAM یا «حافظه اصلی» نسبتاً کوچک کامپیوتر جا

نمی گیرد. این اطلاعات پیش از این که در دسترس CPU قرار بگیرد باید به داخل حافظه

اصلی بارشود. دسته بندی ابزارهای ذخیره سازی انبوه برطبق سادگی دستیابی به اطلاعات،

آنها را به دودسته تقسیم می کند ابزارهای آماده کار و ابزارهای بایگانی. در روش ذخیره

سازی آماده کار که معمولاً روی دیسک های مغناطیسی انجام می شود، اطلاعات ذخیره شده در دسترس CPU قرار دارند بدون آن که نیازی به دخالت انسان از طریق اجرای نرم افزار خاصی باشد. در روش ذخیره سازی بایگانی داده هایی نگهداری می شوند که بندرت به کار می روند و باید بصورت دستی در سیستم بار شوند. ذخیره سازی بایگانی معمولاً روی نوارهای مغناطیسی یا دیسک های مغناطیسی انجام می شود. اگرچه دیسک های نوری مثل CD-ROM ها یا تکنولوژی WORM که بتازگی ظهور کرده اند، ممکن است سمت گیری روش ذخیره سازی بایگانی را به علت قابلیت اطمینان، ظرفیت بالا و قیمت پایین خود تغییر دهند.

ابزارهای رابط با انسان

یگانگی انسان و ماشین توسط مجموعه ای از ابزارهای رابط با انسان تحقق می یابد که متداول ترین آنها عبارتند از پایانه های نمایش تصویر (VDT) و چاپگرها. اگرچه چاپگرها ابزارهای صرفاً خروجی هستند که برای چاپ کردن اطلاعات به کار می روند ولی VDT ها در واقع از دو وسیله تشکیل شده اند زیرا شامل یک صفحه کلید به عنوان ورودی و یک CRT به عنوان خروجی می باشند. یک رشته خاص در مهندسی به نام «ارگونومیک» یا «مهندسی فاکتورهای انسانی» به خاطر ضرورتی که در طراحی این ابزارهای جانبی باتوجه به طبیعت انسان احساس می شد، به وجود آمده است و هدف آن وفق دادن مشخصات انسان با

ماشین های مورد استفاده او به شکلی مطمئن، راحت و کارا می باشد. درحقیقت تعداد شرکت هایی که این دسته از ابزارهای جانبی را تولید می کنند بیشتر از شرکتهای تولید کننده کامپیوتر است. در هر سیستم کامپیوتری دست کم سه تا از این ابزارها وجود دارد: صفحه کلید، CRT و چاپگر. از دیگر ابزارهای رابط با انسان میتوان دستگیره بازی، قلم نوری، ماوس میکروفن و بلندگو را نام برد.

ابزارهای کنترل / نظارت

به کمک ابزارهای کنترل / نظارت (وبرخی نرم افزارها و رابط های الکترونیک دقیق) کامپیوترها می توانند کارهای کنترلی زیادی را بی وقفه، بدون خستگی و بسیار فراتر از توانایی انسان انجام دهند.

کاربردهایی نظیر کنترل حرارت یک ساختمان، محافظت ازخانه، کنترل آسانسور، کنترل وسایل خانگی و حتی جوش دادن قطعات مختلف یک خودرو همگی با استفاده از این ابزارها امکان پذیر هستند.

ابزارهای کنترل، ابزارهای خروجی یا عمل کننده هستند. آنها وقتی که با یک ولتاژ یا جریان، تغذیه شوند می توانند برجهان پیرامون خود اثر بگذارند (مثل موتورها و له ها). ابزارهای نظارت، ابزارهای ورودی یا حسگر هستند که با کمیت هایی نظیر حرارت، نور، فشار، حرکت و مانند آن، تحریک شده و آنها را به جریان یا ولتاژی که توسط CPU

خوانده میشود تبدیل می کنند (مثل فتوترانزیستورها، ترمیستورها و سوئیچ ها). ولتاژ یا جریان توسط مدارهای واسطه، به یک داده دودویی تبدیل میشود و یا برعکس و سپس نرم افزار، یک رابطه منطقی بین ورودی ها و خروجی ها برقرار می کند. سخت افزار و نرم افزار موردنیاز برای ارتباط این ابزارها با میکروکنترلرها یکی از موضوعات عمده این کتاب می باشد.

برنامه ها : بزرگ و کوچک

بحث اصلی ما بر سخت افزار سیستمهای کامپیوتری با یک مرور گذرا بر نرم افزار یا برنامه هایی که برای راه اندازی آنها لازم است متمرکز گردیده است. توجه نسبی به سخت افزار در برابر نرم افزار در سال های اخیر بطور حیرت انگیزی دگرگون شده است. درحالیکه نخستین سالهای پیدایش کامپیوتر شاهد برتری چشمگیر هزینه قطعات، تولید و تعمیر و نگهداری سخت افزار کامپیوتر نسبت به هزینه نرم افزار بود، امروزه با تراشه های LSI هزینه های سخت افزاری، کمتر تعیین کننده هستند و کار پرزحمت و متمرکز برای نوشتن، مستندسازی، پشتیبانی، بهنگام کردن و توزیع نرم افزار است که بخش عمده هزینه اتوماسیون یک فرآیند را با استفاده از کامپیوتر تشکیل می دهد.

اکنون هنگام بررسی انواع مختلف نرم افزار است. سه سطح از نرم افزار را بین کاربر و سخت افزار یک سیستم کامپیوتری نشان می دهد: نرم افزار کاربردی، سیستم عامل و زیرروال های ورودی / خروجی.

در پایین ترین سطح، زیرروال های ورودی / خروجی بطور مستقیم سخت افزار سیستم را اداره می کنند، مثل خواندن کارکترها از صفحه کلید، نوشتن کارکترها در CRT، خواندن بلوک های اطلاعات از دیسک و غیره. از آنجا که این زیرروال ها ارتباط بسیار نزدیکی با سخت افزار دارند توسط طراحان سخت افزار نوشته می شوند و (معمولا) در ROM ذخیره می گردند (به عنوان مثال می توان از BIOS در IBM PC نام برد).

برای ایجاد کردن دسترسی نزدیک برنامه نویسان به سخت افزار سیستم، شرایط صریحی برای ورود و خروجی زیر روال های ورودی / خروجی تعیین شده است. کفایت یک نفر ثبات های CPU را مقداردهی اولیه کرده و زیرروال را فراخوانی کند، در اینصورت عملیات موردنظر انجام می شود و نتایج در ثبات های CPU یا RAM سیستم قرار می گیرد.

به عنوان مکملی برای زیرروال های ورودی / خروجی، ROM شامل یک برنامه شروع به کار است که هنگام روشن شدن سیستم یا آغاز بکار مجدد آن بصورت دستی، اجرا می شود. طبیعت غیر فرار ROM در این مورد اهمیت اساسی دارد، زیرا این برنامه باید هنگام

روشن شدن سیستم وجود داشته باشد. «مدیریت اعمال داخلی» مثل بررسی انتخاب ها، مقداردهی اولیه به حافظه، انجام بررسی هایی به منظور عیب یابی و مانند آن، توسط برنامه آغازگر انجام می شود. در پایان یک روال بارکننده خود راه انداز اولین شیار (یک برنامه کوچک) را از دیسک به داخل RAM می خواند و کنترل را به آن می سپارد. سپس این برنامه بخشی از سیستم عامل را که در RAM مقیم است (یک برنامه بزرگ)، از دیسک می خواند و کنترل را به آن می سپارد و به این ترتیب عملیات شروع به کار سیستم کامل می شود و به عبارت دیگر «سیستم خود را توسط راه اندازهای خودش بالا می آورد».

سیستم عامل مجموعه بزرگی از برنامه های همراه با سیستم کامپیوتری است و مکانیسمی را برای دستیابی، مدیریت و استفاده مؤثر از امکانات کامپیوتر فراهم می کند. این توانایی ها در زبان فرمان و برنامه های مفید سیستم عامل وجود دارند و بنوبه خود گسترش نرم افزارهای کاربردی را آسان می کنند. اگر نرم افزار کاربردی، خوب طراحی شده باشد کاربر بدون دانستن سیستم عامل و یا با دانش اندکی درباره آن، با کامپیوتر ارتباط متقابل برقرار می کند. برقرار کردن یک ارتباط مؤثر، معنی دار و مطمئن با کاربر یکی از مهمترین اهداف در طراحی نرم افزار کاربردی است.

میکروها، مینی ها و کامپیوترهای مرکزی

به عنوان یک نقطه شروع، کامپیوترها براساس اندازه و توان آنها با عنوان میکرو کامپیوترها، مینی کامپیوترها و کامپیوترهای مرکزی دسته بندی می شوند. یک ویژگی کلیدی میکرو کامپیوترها اندازه و بسته بندی CPU می باشد که از یک مدار مجتمع واحد - یعنی یک ریزپردازنده تشکیل شده است. از طرف دیگر مینی کامپیوترها و کامپیوترهای مرکزی علاوه بر آن که در برخی جزئیات معماری، پیچیده تر هستند، CPU هایی مشتمل بر چندین IC دارند که از چند IC (در مینی کامپیوترها) تا چندین برد مدار متشکل از IC ها (در کامپیوترهای مرکزی) تغییر می کند و این برای به دست آوردن سرعت های بالا و توان محاسباتی کامپیوترهای بزرگتر ضروری است.

میکرو کامپیوترهایی مثل IBM PC , Apple Macintosh و Commodore Amiga یک ریزپردازنده را به عنوان CPU بکار برده اند. RAM , ROM و مدارهای واسطه به IC های زیادی نیاز دارند و تعداد قطعات اغلب به همراه توان محاسبه افزایش می یابد. مدارهای واسطه از لحاظ پیچیدگی بسته به ابزارهای I/O تفاوت قابل ملاحظه ای دارند. برای مثال راه اندازی بلندگو که در اغلب میکرو کامپیوترها وجود دارد تنها نیازمند یک جفت گیت منطقی است و در مقابل، رابط دیسک معمولاً شامل IC های زیادی است که بعضاً در بسته های LSI قرار دارند.

ویژگی دیگری که میکروها را از مینی ها و کامپیوترهای مرکزی جدا می کند آن است که میکرو کامپیوترها سیستمهایی تک اجرایی و تک کاربر هستند یعنی با یک کاربر ارتباط متقابل دارند و یک برنامه را در یک زمان اجرا می کنند. از طرف دیگر مینی ها و کامپیوترهای مرکزی سیستمهایی چند اجرایی و چند کاربر هستند یعنی می توانند به کاربران و برنامه های زیادی بطور همزمان سرویس دهند. در عمل، اجرای همزمان برنامه ها توهمی است که در نتیجه عمل «برش زمان» توسط CPU بوجود می آید (با این همه سیستمهای چندپردازی از چندین CPU برای انجام همزمان وظایف استفاده می کنند).

مقایسه ریزپردازنده ها با میکرو کنترلرها

پیش از این خاطر نشان شد که ریزپردازنده ها CPU هایی تک تراشه هستند و در میکرو کامپیوترها به کار می روند. پس فرق میکرو کنترلرها با ریزپردازنده ها چیست؟ با این سؤال از سه جنبه می توان برخورد کرد: معماری سخت افزار، کاربردها و ویژگی های مجموعه دستورالعمل ها .

معماری سخت افزار

برای روشن ساختن تفاوت بین میکرو کنترلرها و ریزپردازنده ها، شکل ۱-۲ برای نشان دادن جزئیات بیشتر دوباره رسم شده است (شکل ۱-۶ را ملاحظه کنید).

درحالی که ریزپردازنده یک CPU ی تک تراشه ای است، میکروکنترلر دریک تراشه واحد شامل یک CPU و بسیاری از مدارات لازم برای یک سیستم میکرو کامپیوتری کامل می باشد. علاوه بر CPU میکروکنترلرها شامل RAM , ROM یک رابط سریال، یک رابط موازی، تایمر و مدارات زمان بندی وقفه می باشند که همگی دریک IC قراردارند . البته مقدار RAM روی تراشه حتی به میزان آن دریک سیستم میکرو کامپیوتری کوچک هم نمی رسد اما آنطور که خواهیم دید این مسأله محدودیتی ایجاد نمی کند زیرا کاربردهای میکروکنترلر بسیار متفاوت است.

یک ویژگی مهم میکروکنترلرها، سیستم وقفه موجود در داخل آنهاست. میکروکنترلرها به عنوان ابزارهای کنترل گرا اغلب برای پاسخ بی درنگ به محرکهای خارجی (وقفه ها) مورد استفاده قرار می گیرند. یعنی باید در پاسخ به یک «اتفاق»، سریعا یک فرآیند را معوق گذارده، به فرآیند دیگر پردازند. بازشدن دریک اجاق مایکروویو مثالی است از یک اتفاق که ممکن است باعث ایجاد یک وقفه در یک سیستم میکروکنترلی شود. البته اغلب ریزپردازنده ها می توانند سیستمهای وقفه قدرتمندی را به اجرا بگذارند، اما برای این کار معمولا نیاز به اجزای خارجی دارند. مدارات روی تراشه یک میکروکنترلر شامل تمام مدارات موردنیاز برای بکارگیری وقفه ها می باشد.

کاربردها

ریزپردازنده ها اغلب به عنوان CPU در سیستم های میکرو کامپیوتری بکار می روند. این کاربرد دلیل طراحی آنها و جایی است که می توانند توان خود را به نمایش بگذارند. با این وجود میکرو کنترلرها در طراحی های کوچک با کمترین اجزاء ممکن که فعالیتهای کنترل گرا انجام میدهند نیز یافت می شوند. این طراحی ها در گذشته با چند دوجین یا حتی صدها IC دیجیتال انجام می شد. یک میکرو کنترلر می تواند در کاهش تعداد کل اجزاء، کمک کند. آنچه که مورد نیاز است عبارت است از یک میکرو کنترلر، تعداد کمی اجزاء پشتیبان و یک برنامه کنترلی در ROM. میکرو کنترلرها برای «کنترل» ابزارهای I/O در طراحی هایی با کمترین تعداد اجزاء ممکن مناسب هستند، اما ریزپردازنده ها برای «پردازش» اطلاعات در سیستمهای کامپیوتری مناسبند.

ویژگیهای مجموعه دستورالعمل ها

به علت تفاوت در کاربردها، مجموعه دستورالعمل های مورد نیاز برای میکرو کنترلرها تا حدودی با ریزپردازنده ها تفاوت دارد. مجموعه دستورالعمل های ریزپردازنده ها بر عمل پردازش تمرکز یافته اند و در نتیجه دارای روشهای آدرس دهی قدرتمند به همراه دستورالعمل هایی برای انجام عملیات روی حجم زیاد داده می باشند. دستورالعمل ها روی چهاربیت ها، بایت ها، کلمه ها یا حتی کلمه های مضاعف عمل می کنند. روشهای آدرس

دهی با استفاده از فاصله های نسبی و اشاره گری های آدرس امکان دسترسی به آرایه های بزرگ داده را فراهم می کنند. حالت های افزایش یک واحدی اتوماتیک و کاهش یک واحدی، حرکت گام به گام روی بایت ها، کلمه ها و کلمه های مضاعف را در آرایه ها آسان می کنند. دستورالعمل های رمزی نمی توانند در داخل برنامه کاربر اجرا شوند و بسیاری ویژگی های دیگر از این قبیل.

از طرف دیگر میکروکنترلرها مجموعه دستورالعمل هایی مناسب برای کنترل ورودی ها و خروجی ها دارند. ارتباط با بسیاری از ورودی ها و خروجی ها تنها نیازمند یک بیت است.

برای مثال یک موتور توسط یک سیم پیچ که توسط یک درگاه خروجی یک بیتی انرژی دریافت می کند، روشن و خاموش شود. میکروکنترلرها دستورالعمل هایی برای 1 کردن و

0 کردن بیت های جداگانه دارند و دیگر عملیات روی بیت ها مثل AND , OR یا EXOR کردن منطقی بیت ها، پرش در صورت 1 یا پاک بودن یک بیت و مانند آنها را نیاز انجام می

دهند. این خصیصه مفید بندرت در ریزپردازنده ها یافت می شود زیرا آنها معمولاً برای کار روی بایت ها یا واحدهای بزرگتر داده طراحی می شوند.

برای کنترل و نظارت بر ابزارها (شاید توسط یک رابط تک بیتی)، میکروکنترلرها مدارات داخلی و دستورالعمل هایی برای عملیات ورودی/خروجی، زمانبندی اتفاقات و فعال کردن

و تعیین اولویت وقفه های ناشی از محرک های خارجی دارند. ریزپردازنده ها اغلب به

مدارات اضافی (IC های رابط سریال، کنترل کننده های وقفه، تایمرها و غیره) برای انجام اعمال مشابه نیاز دارند. با این همه در قدرت پردازش محض، یک میکروکنترلر هرگز به ریزپردازنده نمی رسد (اگر در بقیه موارد یکسان باشند)، زیرا بخش عمده «فضای واقعی» IC میکروکنترلر صرف تهیه امکانات روی تراشه می شود البته به قیمت کاهش توان پردازش.

از آنجا که فضاهای واقعی در تراشه برای میکروکنترلرها اهمیت دارند دستورالعمل ها باید بی نهایت فشرده باشند و اساسا در یک بایت پیاده سازی شوند. یکی از نکات در طراحی جادادن برنامه کنترلی در داخل ROM روی تراشه است، زیرا افزودن بیت یک ROM روی تراشه است، زیرا افزودن حتی یک ROM خارجی هزینه نهایی تولید را بسیار افزایش می دهد. به رمزد آوردن فشرده برای مجموعه دستورالعمل های میکروکنترلر اساسی است، در حالی که ریزپردازنده ها بندرت دارای این ویژگی می باشند، روشهای آدرس دهی قدرتمند آنها بعث به رمزد آوردن غیر فشرده دستورالعمل ها می شود.

میکروکنترلر

میکروکنترلرها مانند دیگر فرآورده هایی که پیش از آن برای برطرف کردن موانع کار مورد ملاحظه بودند، توسط دو نیروی مکمل هم یعنی نیاز بازار و تکنولوژی جدید بوجود آمده اند. تکنولوژی جدید همان است که پیش از این ذکر شد، یعنی نیمه رساناهایی با

ترانزیستورهای بیشتر در فضای کمتر که با قیمت پایین تری به صورت انبوه تولید می شوند.
نیاز بازار، تقاضای صنعت و مصرف کنندگان وسایل و اسباب بازی های هوشمند می باشد.
این تعریف گسترده ای است، بهترین مثال شاید داشبورد خودرو باشد که شاهد تغییر «مرکز کنترل» خودرو در طی دهه گذشته بوده است. زمانی راننده ها باید به دانستن سرعت خود اکتفا می کردند، اما امروزه نمایشی از سوخت صرفه جویی شده و زمان تقریبی رسیدن را در اختیار دارند. زمانی دانستن این که یک کمر بند ایمنی در شروع حرکت محکم شده است یا نه کافی بود امروزه به ما «گفته می شود» کدام کمر بند ایمنی ایراد دارد. اگر دری نیمه باز بماند بموقع توسط کلمات به ما اطلاع داده می شود (شاید کمر بند ایمنی لای درگیر کرده باشد).

همه این موارد این مطلب را در ذهن تداعی می کنند که ریز پردازنده ها (و در این مورد میکروکنترلر) به راه حل هایی تبدیل شده اند که به دنبال یک مسأله می گردند. به نظر میرسد که آنها در کاهش پیچیدگی مدارات فرآورده های مصرفی بسیار مؤثر عمل کرده اند بطوری که تولید کنندگان اغلب برای افزودن امکانات اضافی اشتیاق زیادی دارند، فقط به این علت که میکروکنترلرها خیلی راحت برای فرآورده ها قابل طراحی هستند. نتیجه کار، اغلب فاقد سادگی لازم می باشد. بهترین مثال ممکن ظهور فرآورده ها قابل طراحی هستند. نتیجه کار، اغلب فاقد سادگی لازم می باشد. بهترین مثال ممکن ظهور فرآورده های سخنگو

در سالهای اخیر است. این فرآورده ها، چه خودرو و چه اسباب بازی معمولاً مثالهایی از
زیاده روی ها و طراحی های اضافه برنیاز، و شاید گوشه ای از هنر دهه هشتاد هستند. در آن
زمان هم بسیاری معتقد بودند که همین که گرد کهنگی روی این وسایل بنشیند، تنها چیزی
که برای آنها باقی می ماند قابلیت کاری آنها خواهد بود.

میکروکنترلرها پردازنده هایی اختصاصی هستند. آنها به خودی خود در کامپیوترها به کار
نمی روند، بلکه در فرآورده های صنعتی و وسایل مصرفی مورد استفاده قرار می گیرند.
استفاده کنندگان این فرآورده ها اغلب از وجود میکروکنترلرها کاملاً بی اطلاع هستند. از

دید آنها اجزای داخلی وجود دارند اما جزو جزئیات بی اهمیت طراحی به شمار می
روند. برای مثال اجاق های مایکروویو، ترموستات های قابل برنامه ریزی، ترازوهای
الکترونیک و حتی خودروها را می توانید در نظر بگیرید. قسمت الکترونیکی هر یک از این
فرآورده ها عموماً شامل ارتباط میکروکنترلر با کلیدهای فشاری، سوئیچ ها، وسایل
هشداردهنده و لامپ های روی یک تابلو می باشد. در نتیجه به استثناء برخی امکانات
اضافی، طرز استفاده آنها با فرآورده های الکترومکانیکی قبلی تفاوتی نکرده است و
میکروکنترلر آنها از دید استفاده کنندگان مخفی است.

برخلاف سیستمهای کامپیوتری که توسط قابلیت برنامه ریزی و دوباره برنامه ریزی شدن، باز
شناخته می شوند، میکروکنترلرها یک بار برای همیشه و برای یک کاربرنامه ریزی می

شوند. این مقایسه به یک تفاوت اساسی در معماری این دو سیستم منجر می شود. سیستمهای کامپیوتری نسبت RAM به ROM بالایی دارند و برنامه های کاربران در یک فضای نسبتاً بزرگ RAM اجرا می شود درحالی که روال های ارتباط با سخت افزار در یک فضای کوچک ROM اجرا می گردد. از طرف دیگر میکروکنترلرها نسبت ROM به RAM بالایی دارند، برنامه کنترلی آنها که شاید نسبتاً بزرگ هم باشد در ROM ذخیره می شود، درحالی که RAM فقط برای ذخیره موقت مورد استفاده قرار می گیرد. از آنجا که برنامه کنترلی برای همیشه در ROM ذخیره می شود در مرتبه میان افزار قرار می گیرد، یعنی چیزی بین سخت افزار (مدارهای واقعی) و نرم افزار (برنامه هایی در RAM که هنگام خاموش شدن سیستم پاک می شوند). تفاوت بین سخت افزار و نرم افزار تا حدی شبیه به تفاوت بین یک صفحه کاغذ (سخت افزار) و کلمات نوشته شده روی آن (نرم افزار) می باشد. میان افزار را میتوان به صورت فرم های استاندارد که برای یک کاربرد مشخص طراحی و چاپ شده اند در نظر گرفت.

مزیت ها و معایب

وظایفی که میکروکنترلرها انجام می دهند وظایف تازه ای نیستند. آنچه جدید است این است که طراحی ها با تعداد اجزای کمتری از گذشته انجام می شوند. طراحی هایی که در گذشته با استفاده از ده ها و حتی صدها IC انجام می شدند امروزه با یک میکروکنترلر و

اجزایی به تعداد انگشتان دست قابل انجام اند. کاهش تعداد اجزاء که نتیجه مستقیم قابلیت برنامه ریزی و توانایی زیاد میکروکنترلرها در ایجاد یکپارچگی می باشد، معمولاً منجر به زمان طراحی و ساخت کوتاهتر، هزینه تولید پایین تر، مصرف توان کمتر و قابلیت اطمینان بیشتر شود. اعمال منطقی که نیازمند چندین IC می باشند، اغلب توسط یک میکروکنترلر با اضافه کردن یک برنامه کنترلی انجام می شوند.

عیب کار در سرعت است. راه حل میکروکنترلی هرگز در سرعت به پای راه حل های مشابه با اجزای گسسته نمی رسند. در موقعیت هایی که نیاز به پاسخ های بسیار سریع به رویدادها وجود دارد (که البته بندرت چنین کاربردهایی پیدا می شوند) میکروکنترلرها عکس العمل ضعیفی از خود نشان می دهند.

به کار بردن میکروکنترلر برای چنین عملی چندان مرسوم نیست، اما این مکان وجود دارد. نرم افزار باید عملیات نشان داده شده در نمودار گردشگی شکل ۸-۱ را انجام دهد. برنامه زبان اسمبلی 8051 برای این عمل منطقی به صورت زیر می باشد:

```
LOOP:  MOV      C,P1.4  ;READ P1.4 BIT INTO CARRY FLAG
        ANL      C,P1.5  ;AND WITH P1.5
        ANL      C,P1.6  ;AND WITH P1.6
        CPL      C      ;CONVERT TO "NAND" RESULT
        MOV      P1.7,C  ;SEND TO P1.7 OUTPUT BIT
        SJMP     LOOP    ; REPEAT
```

اگر این برنامه در یک میکروکنترلر 8051 اجرا شود بدون شک تابع NAND با سه ورودی تحقق می یابد (این مطلب را می توان با یک ولت متر یا نوسان نگار تحقیق کرد). تأخیر انتشار از یک گذار در ورودی تا استقرار سطح منطقی درست در خروجی دست کم در مقایسه با معادل TTL آن بسیار طولانی است. بسته به نسبت زمانی تغییر در ورودی و تشخیص این تغییر توسط برنامه، تأخیر بین ۳ تا ۱۷ میکروثانیه خواهد بود. (با فرض عملکرد استاندارد 8051 با استفاده از یک کریستال ۱۲ مگاهرتز) در حالی که تأخیر انتشار در معادل TTL از مرتبه ۱۰ نانو ثانیه است یعنی حدود هزار بار کمتر. واضح است که در ایجاد توابع منطقی، سرعت میکروکنترلرها با مدارهای معادل TTL قابل مقایسه نیست.

در بسیاری از کاربردها بویژه آنهایی که با عملکرد انسان سرو کار دارند این که تأخیرها به نانوثانیه اندازه گیری شوند یا میکروثانیه و میلی ثانیه اهمیتی ندارند، مثال گیت منطقی نشان می دهد که میکروکنترلرها می توانند عملیات منطقی را انجام دهند. از این گذشته هرچه طراحی ها پیچیده تر باشند، مزایای طراحی میکروکنترلی، بیشتر خود را نشان می دهد. تعداد کم اجزاء مزیتی است که قبلاً به آن اشاره شد علاوه بر آن عملیات پیش بینی شده در برنامه کنترلی را می توان تنها با تغییر نرم افزار دگرگون کرد و این روش کمترین اثر ممکن را روی چرخه تولید خواهد گذاشت.

مروری بر خانواده MCS-51TM

MCS-51TM خانواده ای از میکروکنترلرهاست که توسط شرکت اینتل به بازار عرضه شده است. دیگر تولیدکنندگان IC نظیر زیمنس، AMD، فوجیتسو و فیلیپس به عنوان تولیدکننده ثانویه، ICهای این خانواده را تحت مجوز اینتل تولید می کنند. هر میکروکنترلر این خانواده از امکاناتی مناسب با یک سری طراحی های مشخص برخوردار است. 8051 یک IC نوعی و اولین عضو این خانواده است که بصورت تجاری مطرح شد.

خلاصه مشخصات این IC از این قرار است:

- ۴K بیت ROM
- ۱۲۸ بیت RAM
- چهار درگاه I/O (ورودی - خروجی) هشت بیتی
- دو تایمر / شمارنده ۱۶ بیتی
- رابطه سریال
- ۶۴K بیت فضای حافظه خارجی برای کد
- ۶۴K بیت فضای حافظه خارجی برای داده
- پردازنده بولی (که عملیات روی بیت ها را انجام می دهد)
- ۲۱۰ مکان بیتی آدرس پذیر

□ انجام عملیات ضرب و تقسیم در ۴ میکروثانیه

دیگر اعضای خانواده MCS-51TM هر یک از امکانات دیگری از قبیل ROM روی تراشه، RAM, EPROM روی تراشه و یا یک تایمر را دارا هستند. در ضمن هر یک از انواع IC های این خانواده یک نسخه مشابه با CMOS کم مصرف نیز دارد (جدول ۱-۲).

عبارت "8051" از روی تسامح به کل خانواده میکروکنترلرهای MCS-51TM اطلاق می شود. هرگاه بحث روی IC خاصی از این خانواده متمرکز شود شماره قطعه مورد نظر ذکر خواهد شد.

بررسی اجمالی پایه ها

در اینجا، معماری سخت افزار 8051 با نگاهی از بیرون به پایه های آن، معرفی می شود (شکل ۲-۲) و در ادامه شرح مختصری از عملکرد هر پایه ارائه می گردد.

همان طور که در شکل ۲-۲ دیده می شود ۳۲ پایه از ۴۰ پایه 8051 به عنوان خطوط درگاه I/O عمل می کنند. معهذاً ۲۴ خط از این خطوط دو منظوره هستند (۲۶ خط در 8052/8051). هر یک از این خطوط می توانند به عنوان I/O یا خط کنترل و یا بخشی از گذرگاه آدرس یا گذرگاه داده به کار روند.

در طراحی هایی که با کمترین مقدار حافظه و دیگر قطعات خارجی انجام می شوند، از این درگاهها به عنوان I/O همه منظوره استفاده می کنند. هر هشت خط یک درگاه می تواند

به صورت یک واحد در ارتباط با وسایل موازی مانند چاپگرها و مبدلهای دیجیتال به آنالوگ بکار رود. و یا هر خط به تنهایی با وسایل تک بیتی مثل سوئیچ ها، LEDها، ترانزیستورها، سیم پیچ ها، موتورها و بلندگوها ارتباط برقرار کند.

درگاه 0

درگاه 0، یک درگاه دو منظوره از پایه 32 تا 39 تراشه 8051 می باشد. این درگاه در طراحی های با کمترین اجزای ممکن به عنوان یک درگاه I/O عمومی استفاده می شود. در طراحی های بزرگتر که از حافظه خارجی استفاده می کنند، این درگاه یک گذرگاه آدرس و داده مالتی پلکس شده می باشد. (به بخش ۶-۲ حافظه خارجی مراجعه کنید)

درگاه 1

درگاه 1 درگاه اختصاصی I/O روی پایه های 1 تا 8 است. پایه های P1.0 تا P1.7 در صورت نیاز برای ارتباط با وسایل خارجی بکار می روند. وظیفه دیگری برای پایه های درگاه 1 در نظر گرفته نشده است، بنابراین آنها گهگاه برای ارتباط با وسایل خارجی بکار می روند. استثناء در IC های 8032/8052 که از P1.0 و P1.1 به عنوان خطوط I/O و یا ورودی تایمر سوم استفاده می شود.

درگاه 2

درگاه 2 (پایه های 21 تا 28) یک درگاه دو منظوره است که به عنوان I/O عمومی و یا بایت بالای گذرگاه آدرس در طراحی با حافظه کد خارجی به کار می رود. این درگاه همچنین در طراحی هایی که به بیش از ۲۵۶ بایت از حافظه داده خارجی نیاز دارند نیز استفاده می شود.

درگاه 3

درگاه 3 یک درگاه دو منظوره روی پایه های 10 تا 17 می باشد. علاوه بر I/O عمومی این پایه ها هر یک وظایف دیگری نیز در رابطه با امکانات خاص 8051 دارند. وظایف خاص پایه های درگاه 3 و درگاه 2 در جدول ۲-۲ خلاصه شده است.

(Program Store Enable) PSEN

8051 چهارسیگنال اختصاص یافته برای کنترل گذرگاه دارد. $\overline{\text{PSEN}}$ یک سیگنال خروجی روی پایه 29 است که حافظه برنامه خارجی (کد) را فعال می کند. این پایه معمولاً به پایه $\overline{\text{OE}}$ یک EPROM وصل می گردد تا خواندن بایتهای برنامه از EPROM امکان پذیر شود.

سیگنال $\overline{\text{PSEN}}$ در طی مرحله خواندن یک دستورالعمل پایین می رود. کدهای دودویی برنامه (کدهای عملیاتی) از EPROM خوانده می شوند، در گذرگاه داده منتقل می گردند و برای رمزگشایی در ثبات دستورالعمل 8051 ذخیره می شوند. هنگام اجرای برنامه از ROM داخلی (8051/8052) $\overline{\text{PSEN}}$ در حالت غیر فعال (وضعیت بالا) باقی می ماند. چرا؟ چون $\overline{\text{PSEN}}$ تنها خواندن را انجام می دهد.

جدول ۲-۲ عملکرد خاص پایه ها

بیت	نام	آدرس بیت	عملکرد خاص
	RXD		دریافت داده برای درگاه سریال
P3.0	TXD	B0H	ارسال داده برای درگاه سریال
P3.1	$\overline{\text{INT0}}$	B1H	وقفه خارجی 0
P3.2	$\overline{\text{INT0}}$	B2H	وقفه خارجی 1
P3.3	T0	B3H	ورودی خارجی برای تایمر/شمارنده 0
P3.4	T1	B4H	ورودی خارجی برای تایمر/شمارنده 1
P3.5	$\overline{\text{WR}}$	B5H	سیگنال فعال ساز نوشتن در حافظه داده خارجی
P3.6	$\overline{\text{RD}}$	B6H	سیگنال فعال ساز خواندن از حافظه خارجی
P3.7	T2	B7H	ورودی خارجی تایمر/شمارنده 2
P1.0	T2EX	90H	تسخیر/reload تایمر/شمارنده 2
P1.1		91H	

(Address Latch Enable) ALE

سیگنال خروجی ALE در پایه 30 برای هر فردی که با ریزپردازنده های اینتل مثل 8085، 8086 یا 8088 کار کرده باشد، آشناست. 8051 بطور مشابهی از ALE برای جداسازی گذرگاه آدرس و داده استفاده می کند. هنگامی که درگاه 0 در حالت خاص خود به عنوان گذرگاه داده و بایت پایین گذرگاه آدرس استفاده می شود سیگنال ALE آدرس را در یک ثبات خارجی در طی نیمه دوم سیکل حافظه یعنی هنگامی که انتقال داده انجام می شود، در دسترس هستند.

سیگنال ALE با فرکانس یک ششم فرکانس نوسان ساز روی تراشه نوسان می کند و می تواند به عنوان یک پالس ساعت همه منظوره در بقیه سیستم بکار رود. اگر 8051 از یک کریستال ۱۲ مگاهرتز، پالس سرعت دریافت کند، ALE با فرکانس ۲ مگاهرتز نوسان می کند. تنها استثناء در طی انجام دستورالعمل MOVX است که یک پالس ALE حذف می شود. این پایه همچنین برای برنامه ریزی پالس ورودی در انواع EPROM دار 8051 مورد استفاده قرار می گیرد.

(External Access) \overline{EA}

سیگنال ورودی \overline{EA} در پایه 31 معمولاً به سطح منطقی بالا (+5V) یا پایین (زمین) وصل می شود. اگر این پایه در وضعیت بالا قرار گرفته باشد 8051/8052 برنامه را از ROM داخلی یعنی 4K یا 8K بایت پایین حافظه اجرا می کند. هنگامی که پایین باشد ROM

داخلی غیر فعال می شود و برنامه ها از EPROM خارجی اجرا می شوند. همچنین نوع

EPROM دار 8051 از خط \overline{EA} برای تغذیه ۲۱ ولت (V_{pp}) در برنامه ریزی EPROM داخلی استفاده می کند.

(Reset)RST

ورودی RST در پایه ۹، آغازگر اصلی 8051 است. هنگامی که این سیگنال حداقل برای

دو سیکل ماشین در وضعیت بالا بماند، ثباتهای داخلی 8051 با مقادیر مناسبی برای یک شروع به کار سازمان یافته بار می شوند.

ورودی های نوسان ساز روی تراشه

همان طور که در شکل ۲-۲ نشان داده شده است. 8051 دارای یک نوسان ساز روی

تراشه است و معمولاً با یک کریستال که به پایه های 18 و 19 متصل می شود، به راه می

افتد. خازنهای پایدار کننده نیز به صورت نشان داده شده، مورد نیاز هستند. فرکانس نامی

کریستال برای اغلب IC های خانواده MCS-51TM، ۱۲ مگاهرتز است، هر چند که

80C31BH-1 می تواند با فرکانسهایی تا ۱۶ مگاهرتز نیز کار کند. نوسان ساز روی تراشه

الزاماً نیازی به یک کریستال ندارد.

اتصالات تغذیه

8051 با یک تغذیه +۵ ولتی کار می کند. اتصال V_{cc} به پایه 40 و V_{ss} (زمین) به پایه 20 وصل می شود.

ساختار درگاه I/O

نوشتن در پایه یک درگاه، داده را در یک ذخیره ساز درگاه بار می کند. در اثر این عمل یک ترانزیستور اثر میدانی (FET) که به پایه درگاه وصل شده است، راه اندازی میشود. قابلیت راه اندازی برای درگاههای 1,2 و 3 به اندازه چهار TTL شاتکی کم مصرف و برای درگاه 0 به اندازه هشت عدد از همین نوع TTL می باشد.

توجه کنید که مقاومت بالابرنده در درگاه 0 وجود ندارد. (مگر هنگامی که به عنوان گذرگاه خارجی آدرس و داده عمل می کند). ممکن است یک مقاومت بالابرنده خارجی بسته به مشخصات ورودی وسیله ای که توسط درگاه راه اندازی می شود، مورد نیاز باشد.

در 8051 دو قابلیت «خواندن ذخیره ساز» و «خواندن پایه» وجود دارد. دستورالعملهایی که عمل بخوان - تغییر بده - بنویس را بکار می برند (مثل CPL P1.5)، برای پرهیز از تشخیص نادرست سطح ولتاژ در مواقعی که پایه بشدت تحت بار قرار دارد (مثل هنگامی که بیس یک ترانزیستور را تحریک می کند)، عمل خواندن را از ذخیره ساز انجام می دهند و

دستورالعملهایی که یک بیت از درگاه وارد می کنند (مثل MOV C, P1.5)، پایه را می خوانند. ذخیره ساز درگاه در این مورد باید شامل 1 منطقی باشد و گرنه EET راه انداز روشن می شود و خروجی را پایین می کشد. Reset کردن سیستم همه ذخیره سازهای درگاه را 1 می کند. پس اگر یک ذخیره ساز درگاه پاک شود (مثل CLR P1.5)، متعاقب آن نمی توان از پایه به عنوان ورودی استفاده کرد، مگر این که ابتدا ذخیره ساز 1 شود. (SET P1.5)

هنگامی که عملکرد خاص این درگاهها در حال انجام است راه اندازهای خروجی به یک آدرس داخلی (درگاه 2)، آدرس/داده داخلی (درگاه 0) یا یک سیگنال کنترلی (درگاه 3) بصورت مقتضی سوئیچ می شوند.

سازمان حافظه

اغلب ریزپردازنده ها یک فضای حافظه مشترک برای داده و برنامه در نظر می گیرند. این کار معقولی است چون برنامه ها معمولاً روی یک سیستم ذخیره شده و برای اجرا به RAM منتقل می گردند. به این ترتیب برنامه ها و داده هر دو در RAM سیستم مقیم می شوند. از طرف دیگر میکروکنترلرها بندرت به عنوان CPU در «سیستم های کامپیوتری» مورد استفاده قرار می گیرند. در عوض، به عنوان جزء مرکزی در طراحی های کنترل گرا به کار می روند

که در این موارد حافظه محدود است، دیسک درایو یا سیستم عامل دیسک وجود ندارد و برنامه کنترلی باید در ROM قرار داده شود.

به همین دلیل 8051 یک فضای حافظه جداگانه برای برنامه (کد) و داده در نظر می گیرد. همان طور که می دانیم، کد و داده هر دو ممکن است داخلی باشند. با این وجود هر دو با استفاده از اجزاء خارجی تا حد ۶۴K بایت حافظه داده قابل توسعه هستند.

حافظه داخلی شامل ROM روی تراشه (فقط در 8051/8052) و RAM داده روی تراشه است. RAM روی تراشه شامل آرایش مناسبی از حافظه همه منظوره، حافظه بی‌تبی آدرس پذیر، بانک های ثابت و ثابت های کاربرد خاص می باشد.

دو ویژگی جالب توجه در 8051 بدین قرار است: (الف) ثباتها و درگاههای ورودی - خروجی بصورت نقشه حافظه هستند و مانند هر مکان دیگر حافظه قابل دسترسی می باشند. (ب) پشته، برخلاف معمول دیگر ریزپردازنده ها که پشته را در RAM خارجی قرار می دهند، در RAM داخلی قرار دارد.

هر مکانی در RAM همه منظوره با استفاده از روش های آدرس دهی مستقیم یا غیر مستقیم قابل دسترسی است. برای مثال برای خواندن محتویات RAM داخلی در آدرس 5FH به داخل انباره این دستورالعمل می تواند به کار رود:

MOV A,5FH

این دستورالعمل یک بایت داده را با استفاده از آدرس دهی مستقیم برای مشخص کردن «مکان مبدأ» (یعنی آدرس 5FH) منتقل می کند. مقصد داده بطور ضمنی در کد عملیاتی دستورالعمل با عنوان A انباره مشخص شده است.

در ضمن RAM داخلی با استفاده از آدرس دهی غیر مستقیم و R0 و R1 نیز قابل دسترسی است. برای مثال دو دستورالعمل زیر همان کاری را می کنند که دستورالعمل بالا به تنهایی انجام می دهد:

```
MOV R0,#5FH
```

```
MOV A,@R0
```

دستورالعمل نخست از آدرس دهی فوری برای انتقال مقدار 5FH به ثبات R0 استفاده می کند. و دستورالعمل دوم آدرس دهی غیر مستقیم را برای انتقال داده ای که «R0 به آن اشاره می کند» به داخل انباره به کار می گیرد.

RAM بیت آدرس پذیر

8051، ۲۱۰ مکان بیت آدرس پذیر دارد، که ۱۲۸ مکان آن در آدرس بایت 20H تا 2FH

قرار داشته و بقیه ثبات های کاربرد خاص هستند که بعداً مورد بحث قرار می گیرند.

اندیشه دستیابی به بیت های منفرد از طریق نرم افزار، یکی از امکانات قدرتمند اغلب

میکروکنترلرها است. بیت ها می توانند توسط یک دستورالعمل، AND,0,1 و یا OR گردند.

اغلب ریزپردازنده ها به یک رشته از دستورالعملهای «بخوان - تغییر بده - بنویس» برای انجام همین اعمال نیاز دارند. بعلاوه در گاههای I/O در 8051 بصورت بیت های آدرس پذیر هستند که ارتباط نرم افزاری را با تک بیت های ورودی و خروجی ساده می کنند.

تعداد ۱۲۸ مکان بیت آدرس پذیر همه منظوره در آدرس بایت 20H تا 2FH وجود دارد.

(۱۲۸ بیت = ۱۶ بایت * ۸ بیت) این آدرس ها بسته به دستورالعمل، به عنوان بایت یا به عنوان

بایت یا به عنوان بیت قابل دسترسی هستند. برای مثال برای ۱ کردن بیت 67H این

دستورالعمل را می توان بکار برد:

«آدرس بیت 67H» با ارزشترین بیت در «آدرس بایت 2CH» است. دستورالعمل بالا تأثیری

روی دیگر بیت ها در این آدرس ندارد. اکثر ریزپردازنده ها همین عمل را به طریق زیر

انجام میدهند:

```
MOV    A, 2CH          ;READ ENTIRE BYTE
ORL    A,#10000000    ;SET MOST-SIGNIFICANT BIT
MOV    2CH,A          ;WRITE BACK ENTIRE BYTE
```

بانک های ثبات

بانک های ثبات در ۳۲ مکان پایین حافظه داخلی قرار دارند. مجموعه دستورالعملهای 8051

هشت ثبات را از R0 تا R7 پشتیبانی می کند و به صورت پیش فرض (پس از reset شدن

سیستم)، این ثبات ها، در آدرس 00H تا 07H قرار می گیرند دستورالعمل زیر محتوای

آدرس 05H را به داخل انباره منتقل می کند:

```
MOV A,R5
```

این دستورالعمل یک دستورالعمل یک بایتی است که از آدرس دهی ثبات استفاده می

کند. البته همین عملیات با یک دستورالعمل دو بایتی با استفاده از آدرس دهی مستقیم در

بایت دوم نیز انجام پذیر است:

```
MOV A,05H
```

دستورالعملهایی که از ثبات های R0 تا R7 استفاده می کنند کوتاهتر و سریعتر از

دستورالعملهای معادلی هستند که از آدرس دهی مستقیم استفاده می کنند. داده هایی که

بطور متناوب استفاده می شوند، بهتر است یکی از این ثبات ها را استفاده کنند.

بانک ثبات فعال با تغییر بیت های انتخاب بانک ثبات در کلمه وضعیت، برنامه، قابل تغییر

است. با فرض آن که بانک ثبات شماره ۳ فعال باشد دستورالعمل زیر محتوای انباره را در

مکان 18H می نویسد:

```
MOV R0,A
```

اندیشه «بانک های ثبات» امکان «سوئیچ متن» را بصورت سریع و کارا فراهم می کند، که به

موجب آن بخشهای جداگانه یک نرم افزار، مستقل از دیگر بخشها از یک مجموعه ثبات

خاص خود استفاده می کنند.

ثبات های کاربرد خاص

ثبات های داخلی در اغلب ریزپردازنده ها توسط مجموعه دستورالعمل ها به صورت ضمنی قابل دسترسی هستند. برای مثال "INCA" در ریز پردازنده 6809 محتوای انباره A را یک واحد افزایش می دهد. عملیات به صورت ضمنی در کد عملیاتی دستورالعمل مشخص شده است. روش دستیابی مشابهی برای ثبات در میکروکنترلر 8051 امکان پذیر است. در حقیقت دستورالعمل "INC A" در 8051 همان عمل را انجام می دهد.

علاوه بر R0 تا R7، ۲۱ ثبات کاربرد خاص (SFR) در ناحیه بالای RAM از آدرس از 80H تا FFH، تعریف نشده اند. تنها ۲۱ آدرس SFR تعریف شده است. (۲۶ آدرس در 8032/8052).

اگرچه انباره (A) همانطور که قبلا نشان داده شد به صورت ضمنی قابل دسترسی است، اکثر SFR ها با استفاده از آدرس دهی مستقیم قابل دستیابی هستند. با دقت در شکل ۶-۲ ملاحظه می شود که برخی SFR ها هم بیت آدرس پذیر و هم بایت آدرس پذیر می باشند. طراحان باید هنگام کار کردن با بیت ها به جای بایت ها مراقب باشند. برای مثال دستورالعمل:

SETB 0E0H

بیت 0 در انباره را 1 می کند ولی دیگر بیت ها را تغییر نمی دهد. شگرد کار تشخیص این

جدول ۲-۳ خلاصه ثبات PSW (کلمه وضعیت برنامه)

بیت	نماد	آدرس	شرح عملکرد بیت
PSW.7	CY	D7H	پرچم نقلی
PSW.6	AC	D6H	پرچم نقلی کمکی
PSW.5	F0	D5H	پرچم 0
PSW.4	RS1	D4H	انتخاب بانک ثابت 0
PSW.3	RS0	D3H	انتخاب بانک ثبات 1
			00H - 07H = بانک 0، آدرس های
			08H - 0FH = بانک 1، آدرس های
			10H - 17H = بانک 2، آدرس های
			18H - 1FH = بانک 3، آدرس های
PSW.2	OV	D2H	پرچم سرریز
PSW.1	-	D1H	رزرو شده
PSW.0	P	D0H	پرچم توازن زوج

بیت توازن

بیت توازن (P) در هرسیکل ماشین برای ساختن توازن زوج انباره، بطورخود کار 1 و یا پاک می شود تعداد بیت های 1 در انباره به علاوه بیت P همواره زوج است اگر برای مثال انباره شامل 10101101B باشد P حاوی 1 خواهد بود (برای تأمین شش عدد 1 که تعداد زوجی از 1 ها است). بیت توازن اغلب در ارتباط با روال های درگاه سریال برای اضافه کردن بیک بیت توازن پیش از فرستادن، و یا بازرسی توازن پس از دریافت به کار می رود.

ثبات B

ثبات B در آدرس F0H به همراه انباره برای عملیات ضرب و تقسیم استفاده میشود. دستورالعمل MUL AB مقادیر هشت بیتی بدون علامت در A و B را ضرب کرده و نتیجه ۱۶ بیتی را در A (بایت پایین) و B (بایت بالا) قرار می دهد. دستورالعمل DIV AB، A را بر B تقسیم می کند و خارج قسمت را در A و باقیمانده را در B می گذارد. ثبات B همچنین به عنوان یک ثبات چرکنویس همه منظوره عمل می کند. این ثبات بیت آدرس پذیر از آدرس F0H تا F7H نیز هست.

اشاره گر پیشته

اشاره گر پیشته (SP) یک ثبات هشت بیتی در آدرس 81H است. این ثبات حاوی آدرس داده ای است که در همان هنگام در بالای پیشته قرار دارد. عملیات پیشته شامل پوش کردن

داده به پشته و پاپ کردن داده از پشته می باشد. پوش کردن به پشته SP را یک واحد

کاهش میدهد. پشته 8051 در RAM داخلی قرار دارد و محدود به آدرسهایی است که با

آدرس دهی غیرمستقیم قابل دسترسی هستند. یعنی ۱۲۸ بایت اول در 8051 / 8031 یا تمام

۲۵۶ بایت RAM روی تراشه در 8052 / 8032.

برای مقدار اولیه دادن دوباره به SP در پشته ای که از آدرس 60H آغاز می شود.

دستورالعمل زیر مورد استفاده قرار میگیرد:

MOV SP, 35FH

در 8031 / 8051 این دستورالعمل پشته را به ۳۲ بایت محدود می کند، چون بالاترین

آدرس در RAM روی تراشه، 7FH است. در این دستورالعمل مقدار 5FH به کاررفته

چون SP پیش از اولین عمل پوش یک واحد افزایش می یابد و 60H می شود.

طراحان ممکن است نخواهند اشاره گر پشته را مقداردهی اولیه کنند و بگذارند مقدار پیش

فرض خود را از هنگام reset شدن حفظ کند. این مقدار یعنی 07H با 8048 که پردازنده

مقابل 8051 است، سازگاری دارد و باعث می شود که اولین نوشتن در پشته، داده را در

آدرس 08H ذخیره کند. اگر نرم افزار مورد استفاده به SP مقدار اولیه ندهد، بانک ثابت 1 (

و شاید 2 و 3) در دسترس نخواهند بود، زیرا این ناحیه از RAM داخلی بخشی از پشته می

باشد.

پشته بطور صریح توسط دستورالعمل های PUSH و POP برای ذخیره و بازیافت موقتی داده ها قابل دسترسی است. یا بصورت ضمنی با فراخوانی زیر روال ها (ACALL , LCALL) و بازگشت (RET , RETI) که دستورالعمل هایی برای ذخیره و بازیابی شمارنده برنامه هستند، می توان به آن دسترسی پیدا کرد.

اشاره گر داده

اشاره گر داده (DPTR) که برای دستیابی به حافظه کد یا داده خارجی استفاده می شود، یک ثابت ۱۶ بیتی است که در آدرس های 82H (بایت پایین، DPL) و 83H (بایت بالا، DPH) قرار دارد. سه دستورالعمل زیر 55H را در مکان 1000H در RAM خارجی می نویسد:

```
MOV A, #55H
```

```
MOV DPTR, #1000H
```

```
MOVX @DPTR, A
```

اولین دستورالعمل از آدرس دهی فوری برای انتقال داده ثابت 55H به انباره استفاده می

کند. دومین دستورالعمل نیز آدرس دهی فوری را این بار برای انتقال آدرس ثابت 1000H

1000H به اشاره گر داده به کار می برد. دستورالعمل سوم از آدرس دهی غیر مستقیم برای

انتقال محتوای A (55H) به مکانی از RAM خارجی که آدرسش در DPTR است

(1000H)، استفاده می کند.

ثبات های درگاه

درگاه های I/O در 8051 عبارتند از درگاه 0 در آدرس 80H، درگاه 1 در آدرس 90H، درگاه 2 در آدرس A0H و درگاه 3 در آدرس B0H. درگاه های 0 و 2 و 3 در صورت استفاده از حافظه خارجی یا برخی از امکانات خاص 8051 (مثل وقفه ها، درگاه سریال و غیره)، برای عملیات I/O در دسترس نیستند. با این همه P1.2 تا P1.7 همواره به عنوان خطوط I/O همه منظوره قابل استفاده می باشند.

همه درگاه بیت آدرس پذیر هستند که امکانات قدرتمندی را برای ارتباطات متقابل فراهم می کند. اگر یک موتور از طریق یک سیم پیچ و سوئیچ ترانزیستوری به بیت 7 درگاه 1 وصل باشد، برای مثال می تواند با یک دستورالعمل 8051 یعنی:

SETB P1.7

روشن شود و با دستورالعمل زیر خاموش گردد:

CLR P1.7

دستورالعمل های فوق از عملگر نقطه برای آدرس دادن به یک بیت در یک مکان بیت آدرس پذیر استفاده می کنند. اسمبلر تغییرات لازم را انجام می دهد، بدین ترتیب دودستورالعمل زیر معادل هستند:

CLR P1.7

CLR

97H

کاربرد نمادهای از پیش تعریف شده اسمبلر (مثل P1) در فصل ۷ بطور مفصل بررسی شده

است. به عنوان یک مثال دیگر ارتباط با وسیله ای شامل یک بیت وضعیت به نام BUSY را در نظر بگیرید که هنگام مشغول بودن وسیله، 1 و هنگام آماده بودن آن پاک می باشد. اگر BUSY بر فرض به بیت 5 درگاه 1 وصل شده باشد، حلقه زیر باعث می شود که سیستم برای آماده شدن وسیله صبر کند:

WAIT : JB P1.5, WAIT

این دستورالعمل یعنی «اگر بیت P1.5، شد، به برجسب WAIT پرش کن» به عبارت دیگر «برگرد و دوباره آن را بررسی کن».

ثبات های درگاه سریال

8051 شامل یک درگاه سریال روی تراشه برای ارتباط با ابزارهای سریال مانند پایانه ها و مودم ها و یا برای ارتباط با دیگر IC هایی که خطوط ارتباط سریال دارند (مبدل های آنالوگ به دیجیتال، ثبات های انتقال RAM های غیرفرار و مانند آن)، می باشد. ثباتی به نام بافر داده سریال (SBUF) در آدرس 99H هر دو داده دریافتی را در دسترس قرار می دهد. حالت های مختلف عملکرد درگاه سریال از طریق ثبات کنترل درگاه سریال در آدرس 98H (SCON) که بیت پذیر نیز هست، قابل برنامه ریزی می باشد. عملکرد درگاه سریال بطور مفصل در فصل ۵ بحث و بررسی شده است.

ثبات های وقفه

8051 دارای یک ساختار وقفه با پنج منبع وقفه و دو سطح تقدم می باشد. وقفه ها پ از reset شدن سیستم غیرفعال می شوند و سپس با نوشتن در ثبات فعال سازی وقفه (IE) در آدرس A8H، فعال می گردند. سطح تقدم از طریق ثبات تقدم وقفه (IP) در آدرس B8H تعیین می شود. هر دو ثابت بیت آدرس پذیر هستند. وقفه ها بطور مشروح در فصل ۶ بررسی خواهند شد.

ثبات کنترل توان

ثبات کنترل توان (PCON) در آدرس 87H شامل بیت های کنترلی مختلفی است که در جدول ۲-۴ خلاصه شده اند.

بیت SMOD نرخ ارسال در گاه سریال را در حالت های 1 و 2 و 3 دو برابر میکند. (به فصل ۵ مراجعه کنید). بیت های 4 و 5 و 6 در PCON تعریف نشده اند و بیت های 2 و 3 بیت های پرچم همه منظوره برای استفاده کاربران می باشند. بیت های کنترل توان یعنی افت تغذیه (PD) و معلق (IDL) ابتدا در همه IC های خانواده MCS-51TM در دسترس بودند اما هم اکنون تنها در نسخه های CMOS منظور

جدول ۲-۳ خلاصه ثبات PCON

بیت	نماد	شرح عملکرد
7	SMOD	بیت دو برابر کننده نرخ ارسال ، هنگام 1 شدن ، نرخ ارسال در حالت های 1 ، 2 یا 3 درگاه سریال دو برابر میشود
6	-	تعریف نشده
5	-	تعریف نشده
4	-	تعریف نشده
3	GF1	بیت 1 پرچم همه منظوره
2	GF0	بیت 0 پرچم همه منظوره
1*	PD	افت تغذیه ، برای فعال کردن حالت افت تغذیه 1 میشود تنها راه خروج ، reset شدن است .
0*	IDL	حالت معلق ، برای فعال کردن حالت معلق 1 میشود ، تنها راه خروج ، یک وقفه یا reset شدن است .

می شوند. ثبات PCON بیت آدرس پذیر نیست.

حالت معلق

دستورالعملی که بیت IDL را 1 کند، آخرین دستورالعملی خواهد بود که پیش از ورود به حالت معلق اجرا می شود. در حالت معلق سیگنال ساعت داخلی به CPU اعمال می شود، اما نه برای کاربردهای وقفه، تایمر و درگاه سریال. وضعیت CPU حفظ و محتویات ثبات ها تثبیت می گردد. پایه های درگاه ها هم سطح منطقی خود را حفظ می کنند. ALE و PSEN نیز بالا می روند.

حالت معلق توسط هر وقفه ای که فعال شود یا با reset شدن سیستم پایان می یابد هر یک از این شرایط بیت IDL را پاک می کند.

حالت افت تغذیه

دستورالعملی که بیت PD را 1 کند آخرین دستورالعملی خواهد بود که پیش از ورود به حالت افت تغذیه اجرا می شود. در حالت افت تغذیه: ۱- نوسان ساز روی تراشه متوقف می شود، ۲- تمامی اعمال متوقف می شوند، ۳- محتویات RAM روی تراشه حفظ می شود، ۴- پایه های درگاه سطح منطقی خود را حفظ می کنند، ۵- ALE و PSEN پایین می روند. تنها راه خروج از این حالت، reset کردن سیستم است.

هنگامی که سیستم در این حالت قرار دارد Vcc به ۵ ولت برگردد.

حافظه خارجی

برای پرهیز از یک تنگنای بالقوه در طراحی، میکروکنترلرها باید قابلیت توسعه را فراتر از منابع و امکانات روی تراشه خود داشته باشند. اگر قرار است امکاناتی توسعه یابد (حافظه، I/O و مانند آن) قابلیت آن باید وجود داشته باشد. معماری MCS-51^{MT} این قابلیت را به صورت ۶۴K بایت فضای حافظه خارجی برای داده فراهم کرده است و در صورت نیاز ROM و RAM اضافی را می توان به آن افزود. IC های ارتباط با ابزارهای جانبی نیز می توانند برای افزایش قابلیت I/O اضافه گردند. اینها جزئی از فضای حافظه داده خارجی با استفاده از نقشه حافظه برای I/O می باشند.

هنگامی که حافظه خارجی مورد استفاده قرار می گیرد درگاه 0 به عنوان یک درگاه I/O قابل استفاده نیست. این درگاه به یک گذرگاه آدرس (A0-A7) و داده (D0-D7) مالتی پلکس شده تبدیل می شود. ALE بایت پایین آدرس را در شروع هر سیکل حافظه خارجی ذخیره می کند. درگاه 2 معمولاً (اما نه همیشه) برای بایت بالای گذرگاه آدرس به کار گرفته می شود.

پیش از بحث پیرامون جزئیات خاص مالتی پلکس کردن گذرگاه های آدرس و داده، ایده کلی در شکل ۷-۲ نشان داده شده است. یک آرایش بدون مالتی پلکس از ۱۶ خط اختصاصی آدرس و ۸ خط اختصاصی داده یعنی کلاً از ۲۴ پایه استفاده می کند. آرایش

مالتی پلکس شده ۸ خط گذرگاه داده را با بایت پایین گذرگاه آدرس مالتی پلکس می نماید. این تعداد با ۸ خط دیگر برای بایت بالای گذرگاه آدرس، کلا ۱۶ پایه می شود. این صرفه جویی در پایه ها باعث می شود که امکانات و توانایی های بیشتری در یک بسته بندی دو ردیفه ۴۰ پایه ایجاد شود.

حال بینیم که آرایش مالیت پلکس شده چگونه کار می کند: در طی نیمه نخست هر سیکل حافظه بایت پایین آدرس در درگاه 0 قرار می گیرد و توسط ALE ذخیره می شود. یک 74HC373 (یا معادل آن) بایت پایین آدرس را در طی سیکل حافظه پایدار نگاه می دارد. در طی نیمه دوم سیکل حافظه درگاه 0 به عنوان گذرگاه داده به کار می رود و داده، بسته به عمل انجام شده خوانده یا نوشته می شود.

دستیابی به حافظه کد خارجی

حافظه کد خارجی یک حافظه فقط خواندنی است که توسط سیگنال PSEN فعال می شود. هنگامی که از یک EPROM خارجی استفاده می کنیم، درگاه های 0 و 2 به عنوان درگاه های I/O همه منظوره در دسترس ما نیستند. اتصالات سخت افزاری برای حافظه EPROM خارجی در شکل ۸-۲ نشان داده شده است.

یک سیکل ماشین در 8051، برابر با ۱۲ تناوب نوسان ساز است. برای نوسان ساز روی تراشه با یک کریستال ۱۲ مگاهرتز هر سیکل ماشین یک میکروثانیه طول می کشد. در طی یک

سیکل ماشین نوعی، ALE و پالس می دهد و دو بایت از حافظه برنامه خوانده می شود.
(اگر دستورالعمل در حال اجرا، یک دستورالعمل یک بیتی باشد از بایت دوم استفاده نمی شود). زمانبندی این عملیات که به واکنشی کد عملیاتی معروف است در شکل ۹-۲ نشان داده شده است .

دستیابی به حافظه داده خارجی

حافظه داده خارجی یک حافظه خواندنی - نوشتنی است که با RD و WR فعال میشود . این دو سیگنال عملکرد دیگر پایه های P3.6 و P3.7 می باشند. تنها راه دستیابی به حافظه داده خارجی دستورالعمل MOVX میباشد که از اشاره گر داده ۱۶ بیتی (DPTR)، R0 و یا R1 به عنوان ثبات آدرس استفاده می کند.

RAM ها می توانند همانند EPROM ها به 8051 مرتبط شوند، با این تفاوت که RD به خط OE تراشه RAM و WR به خط W آن وصل می شود. اتصالات گذرگاه آدرس و داده همانند EPROM ها است. همانطور که گفته شد با استفاده از درگاه های 0 و 2 تا ۶۴K بایت از RAM خارجی داده می تواند به 8051 وصل شود.

یک نمودار زمانبندی برای یک عمل خواندن از حافظه داده خارجی برای دستورالعمل MOVX A, @DPTR در شکل ۱۰-۲ نشان داده شده است. توجه کنید که یک پالس

ALE و یک پالس PSEN در عوض یک پالس روی خط RD برای فعال کردن RAM

حذف شده است.

زمانبندی یک سیکل نوشتن برای A , MOVX @DPTR به همان شکل است. فقط WR

پایین می رود و داده از درگاه 0 خارج می شود. (RD در وضعیتن بالا باقی می ماند).

در سیستم هایی که با کمترین اجزاء ممکن طراحی می شوند، از عملکرد دیگر درگاه 2

(یعنی تأمین بایت بالای آدرس) صرف نظر می شود. زیرا این سیستم ها از حافظه کد خارجی

استفاده نمی کنند و تنها به مقدار کمی حافظه داده خارجی نیاز دارند. اگر حافظه داده

خارجی به صورت حافظه های کوچک صفحه گرا پیکربندی شود توسط آدرسهای هشت

بیتی قابل دسترسی است. اگر بیش از یک صفحه ۲۵۶ بایتی از RAM مورد نیاز باشد چند

بیت از درگاه 2 (یا درگاه های دیگر) می تواند مطابق شکل ۱۱-۲ به 8051 مرتبط شود.

بیت های 0 و 1 درگاه 2 باید برای انتخاب یک صفحه مقداردهی اولیه شوند و سپس یک

دستورالعمل MOVX برای خواندن و نوشتن داده در صفحه استفاده می شود. برای مثال

فرض کنید $P2.0 = P2.1 = 0$ دستورالعمل های زیر می توانند برای خواندن محتویات

RAM خارجی به داخل انباره در آدرس 0050H به کار بروند:

```
MOV R0 , #50H
```

```
MOVX A, @R0
```

برای خواندن آخرین آدرس در این RAM یعنی 03FFH، دو بیت انتخاب صفحه باید 1

شوند. رشته دستورالعمل های زیر برای این منظور می توانند مورد استفاده قرار بگیرند:

SETB P2.0

SETB P2.1

MOV R0 , #0FFH

MOVX A, @R0

یک مزیت این طراحی این است که بیت های 2 تا 7 در گاه 2 به عنوان بیت های آدرس

استفاده نمی شوند در حالی که اگر DPTR به عنوان ثبات آدرس به کار می رفت، مورد نیاز

بودند. در نتیجه P2.2 تا P2.7 برای مقاصد I/O در دسترس قرار می گیرند.

رمزگشایی آدرس

اگر EPROM ها و یا RAM های متعددی به یک 8051 وصل شوند، رمزگشایی آدرس

مورد نیاز خواهد بود. این رمزگشایی مثل رمزگشایی مورد نیاز برای اغلب ریزپردازنده

هاست. برای مثال اگر از EPROM ها و یا RAM های ۸K بایتی استفاده شود، گذرگاه

آدرس باید طوری رمزگشایی گردد که IC های حافظه را در مرزهای ۸K بایت مثل

0000H-1FFFH و 2000H-2FFFH و 3000H-3FFFH و مانند آن انتخاب کند.

عموماً از یک IC رمزگشا مانند 74HC138 برای این کار استفاده می شود و پایه های

خروجی آن به ورودی های انتخاب تراشه (CS) در IC های حافظه وصل می گردد. این

مسئله در شکل ۱۲-۲ برای سیستمی با EPROM های متعدد ۸K بایتی از نوع 2764 و

RAM های ۸K بایتی از نوع 6264 نشان داده شده است. به یاد داشته باشید که به خاطر

خطوط فعال سازی مجزا (PSEN برای حافظه کد و RD و WR برای حافظه داده) 8051

میتواند تا ۶۴K بایت EPROM و ۶۴K بایت RAM را به کارگیرد.

دستورالعمل های فوق از عملگر نقطه برای آدرس دادن به یک بیت در یک مکان بیت

آدرس پذیر استفاده می کنند. اسمبلر تغییرات لازم را انجام می دهد، بدین ترتیب دو

دستورالعمل زیر معادل هستند:

CLR P1.7

CLR 97H

کاربرد نمادهای از پیش تعریف شده اسمبلر (مثل P1) در فصل ۷ بطور مفصل بررسی شده

است. به عنوان یک مثال دیگر ارتباط با وسیله ای شامل یک بیت وضعیت به نام BUSY را

در نظر بگیرید که هنگام مشغول بودن وسیله، 1 و هنگام آماده بودن آن پاک می باشد. اگر

BUSY بر فرض به بیت 5 در گاه 1 وصل شده باشد، حلقه زیر باعث می شود که سیستم

برای آماده شدن وسیله صبر کند:

WAIT: JB P1.5 , WAIT

این دستورالعمل یعنی «اگر بیت P1.5، 1 شد، به برجسب WAIT پرش کن» (به عبارت

دیگر» برگردد و دوباره آن را بررسی کن».

ثبات های تایمر

8051 دارای دو تایمر / شمارنده ۱۶ بیتی برای زمانبندی فاصله های زمانی و یا شمارش اتفاقات است. تایمر 0 در آدرس های 8AH (بایت پایین، TL0) و 8CH (بایت بالا، TH0) و تایمر 1 در آدرس های 8BH (بایت پایین، TL1) و 8DH (بایت بالا، TH1) قرار دارند. عملکرد تایمر توسط ثبات حالت تایمر (TMOD) در آدرس 89H و ثبات کنترل تایمر (TCON) در آدرس 88H تعیین می شود که از این دو، تنها TCON بیت آدرس پذیراست. تایمرها در فصل ۴ بطور مشروح مورد بحث قرار خواهند گرفت.

ثبات های وقفه

8051 دارای یک ساختار وقفه با پنج منبع وقفه و دو سطح تقدم می باشد. وقفه ها پس از reset شدن سیستم غیرفعال می شوند و سپس با نوشتن در ثبات فعال سازی وقفه (IE) در آدرس A8H، فعال می گردند. سطح تقدم از طریق ثبات تقدم وقفه (IP) در آدرس B8H تعیین می شود. هر دو ثبات بیت آدرس پذیر هستند. وقفه ها بطور مشروح در فصل ۶ بررسی خواهند شد.

ثبات کنترل توان

ثبات کنترل توان (PCON) در آدرس 87H شامل بیت های کنترلی مختلفی است که در جدول ۲-۴ خلاصه شده اند.

بیت SMOD نرخ ارسال درگاه سریال را در حالت های 1 و 2 و 3 دو برابر می کند. (به فصل ۵ مراجعه کنید). بیت های 4 و 5 و 6 در PCON تعریف نشده اند و بیت های 2 و 3 بیت های پرچم همه منظوره برای استفاده کاربران می باشند. بیت های کنترل توان یعنی افت تغذیه (PD) و معلق (IDL) ابتدا در همه IC های خانواده MCS-51TM در دسترس بودند اما هم اکنون تنها در نسخه های CMOS منظور می شوند. ثبات PCON بیت آدرس پذیر نیست.

اشتراک در فضای حافظه کد و داده خارجی

از آنجا که حافظه کد فقط خواندنی است. ممکن است درحین تهیه نرم افزار 8051 وضعیت نامناسبی رخ دهد. سؤالی که در اینجا پیش می آید این است که چگونه یک نرم افزار که در سیستم موردنظر نوشته شده از فضای کد فقط خواندنی قابل عیب یابی است؟ یک شگرد معمول آن است که فضای حافظه کد و داده خارجی را به اشتراک بگذاریم. چون PSEN برای خواندن حافظه کد، و RD برای خواندن حافظه داده به کار می روند، یک RAM می تواند با اتصال خط OE خود به AND (NOR با ورودی منفی) PSEN و RD هم فضای حافظه کد، و هم فضای حافظه داده در آن نوشته شود و به عنوان حافظه داده یا کد از آن خوانده شود. بدین ترتیب یک برنامه می تواند به داخل RAM (با نوشتن در آن به عنوان حافظه داده) منتقل گردد و (با دستیابی به آن عنوان حافظه کد) اجرا شود.

۷-۲ امکانات اضافی 8052 / 8032

IC های 8032 / 8052 (و نسخه های CMOS و یا نسخه های EPROM دار)، دو مزیت

بر IC های 8031 / 8051 دارند. اول این که دارای ۱۲۸ بیت اضافی RAM روی تراشه از

آدرس 80H تا FFH میباشند. پس برای این که با SFR ها (که در همان آدرس ها قرار

دارند) تداخل بوجود نیاید، $\frac{1}{8}$ کیلوبایت اضافی RAM تنها از طریق آدرس دهی

غیرمستقیم قابل دستیابی است. یک دستورالعمل مثل:

```
MOV R0, #0F0H
```

در همه IC های خانواده MCS-51^{MT} محتویات ثبات B را به انباره منتقل میکند. دو

دستورالعمل زیر:

```
MOV R0, #0F0H
```

```
MOV A, #R0
```

محتوای آدرس داخلی F0H را در IC های 8032 / 8052 به انباره منتقل میکند اما در IC

های 8031 / 8051 تعریف نشده است.

دومین مزیت 8032 / 8052 یک تایمر ۱۶ بیتی اضافی است یعنی تایمر 2 که از طریق پنج

ثبات کاربرد خاص اضافی برنامه ریزی میشود که در جدول ۵-۲ خلاصه شده اند. برای

جزئیات بیشتر به فصل ۴ مراجعه کنید.

عملیات راه اندازی مجدد، reset

8051 با قرار گرفتن RST در وضعیت منطقی بالا برای دست کم دوسیكل ماشین و سپس پایین آمدن آن می تواند reset شود. RST ممکن است بطور دستی با استفاده از یک سوئیچ و یا هنگام اعمال تغذیه توسط یک مدار RC (مقاومت - مخازن) فعال گردد. شکل (۱۵-۲) دو مدار برای reset کردن سیستم را نشان می دهد.

وضعیت همه ثبات های 8051 پس از reset سیستم در جدول 2-6 خلاصه شده است. شاید مهمترین این ثبات ها شمارنده برنامه باشد که با 0000H بار می شوند. هنگامی که RST

دوباره پایین

جدول ۲-۵ ثبات های تایمر 2

ثبات	آدرس	شرح عملکرد	بیت آدرس پذیر
T2CON	C8H	کنترل	بله
RCAP2L	CAH	تسخیر بایت پایین	خیر
RCAP2H	CBH	تسخیر بایت بالا	خیر
TL2	CCH	بایت پایین تایمر 2	خیر
TH2	CDH	بایت بالای تایمر 2	خیر

جدول ۶-۲ مقادیر ثبات ها پس از reset سیستم

محتویات	ثبات ها
0000H	شمارنده برنامه
00H	انباره
00H	ثبات B
00H	PSW
07H	SP
0000H	DPTR
FFH	درگاه 0-3
XXX00000B	IP (8031/8051)
XX000000B	IP (8032/8052)
0XX00000B	IE (8031/8051)
0X000000B	IE (8032/8052)
00H	ثبات های تایمر
00H	SCON
00H	SBUF
0XXXXXXXB	PCON (HMOS)
0XXX0000B	PCON (CMOS)

خروجی آنالوگ

در ارتباط با دنیای بیرون گاهی لازم است که شرایط آنالوگ را حس و یا ارسال کرد. تولید و کنترل سیگنال خروجی آنالوگ از میکروکنترلر آسان است. در این مثال طراحی، از دو مقاومت، دو خازن، یک پتانسیومتر، یک op-amp LE301 و یک MC1408L8 که یک مبدل دیجیتال به آنالوگ ۸ بیتی است، استفاده شده است. هر دو IC ارزان بوده و براحتی قابل تهیه هستند. هشت داده ورودی به DAC از درگاه 1 در 8031 (شکل ۱۵-۱۰) را ملاحظه کنید) فعال می شوند. مدار، پس از ساخت و اتصال آن به SBC-51، باید با استفاده از فرمانهای نمایش، آزمایش شود. ولتاژ خروجی پایه ۶ را در LM301 (Vo)، هنگام نوشتن مقادیر مختلف در درگاه 1 اندازه گیری و پتانسیومتر 1K را تنظیم کنید. خروجی باید از صفر ولت (P1 = 00H) تا حدود ۱۰ ولت (P1 = FFH) تغییر کند. پس از آن که مدار یک بار بطور صحیح کار کرد، آماده ارتباط نرم افزاری است. برنامه معمول برای آزمایش، مولد موج دندانه اریه ای است که مقداری را به DAC می فرستد، آن را افزایش می دهد و مجدداً می فرستد و همین طور این کار را ادامه می دهد، (سؤال ۳ را در انتهای این فصل ببینید). اما ما این کار را با طرحی پیشرفته تر - یعنی مولد موج سینوسی با کنترل دیجیتالی، انجام می دهیم.

از آنجا که توانایی خرد کردن عدد در 8031 محدود است، تنها راه منطقی برای این مسأله استفاده از جدول جستجو می باشد. به یک جدول با مقادیر ۸ بیتی متناظر با یک دوره

تناوب موج سینوسی نیاز است . مقادیر باید حول ۱۲۷ شروع شده ، تا ۲۵۵ افزایش یابد .

سپس به ۱۲۷ و صفر کاهش یافته ، مجدداً به ۱۲۷ باز میگردد و الگوی یک موج سینوسی را تعقیب می کند .

تقریب منطقی موج سینوسی نیاز به جدولی نسبتاً بزرگ دارد ، بنابراین چگونگی تولید چنین

جدولی مدنظر است . روشهای دستی ، غیرعملی می باشند . آسانترین راه ، نوشتن برنامه ای

بزبان سطح بالا جهت تولید جدول و ذخیره کردن آن در یک فایل است . سپس جدول به

برنامه منبع 8031 وارد شده کار تمام می شود . شکل ۱۶-۱۰ یک برنامه ساده بزبان C

جدول 51.c را که انجام دهنده این عمل است ، نشان می دهد . برنامه یک جدول موج

سینوسی با ۱۰۲۴ ورودی با مقادیری بین صفر تا ۲۵۵ ، تولید می کند . خروجی دریک فایل

خروجی به نام sine 51.src نوشته می شود . قبل از هر ورود ، برای سازگار شدن با کد

منبع 8031 یک «DB» قرار می گیرد .

برنامه موج سینوسی 8031 در شکل ۱۷-۱۰ نشان داده شده است . حلقه اصلی (خطوط 36 تا

40) شامل مواد زیر است : آغاز تایمر 0 هر 1000 مقدار از جدول جستجو با استفاده از

DPTR خوانده شده و سپس در درگاه 1 نوشته می شود . مقدار ثابتی به نام STEP برای

افزایش داخل جدول مورد استفاده قرار می گیرد . STEP به عنوان یک بایت در RAM

داخلی ، در خط 26 تعریف شده است و باید توسط یک فرمان نمایش آغاز شود . در هر

ISR، STEP به DPTR افزوده میشود تا آدرس نمونه برداری بعدی به دست آید. آغاز جدول، ORG، از محل 8400H (خط 69)، بنابراین در مرز 1K آغاز می شود. اگر DPTR پس از 87FFH (پایان جدول) باز هم افزایش یابد، طوری تنظیم میشود که به آغاز جدول بازگردد. از آنجا که جدول بسیار بزرگ است، دستور \$NOLIST اسمبلر، پس از پنج ورود اول (خط 77) به کار می رود تا به خروج فایل فهرستی پایان بدهد. دستور \$NOLIST اسمبلر، در خط 1092 (که نشان داده نشده است)، به کار رفته تا برای پنج ورود آخر رجوع به فهرست را امکان پذیر سازد. فرکانس موج سینوسی با سه پارامتر کنترل می شود: STEP، اندازه جدول و دروه تناوب وقفه تایمر، که این امر در خطوط 16 تا 20 فهرست مشخص شده است.

هدف طرح

فرض کنید که رقمهای BCD در RAM داخلی در محلهای 70H و 71H قرار دارند. با استفاده از وقفه ها، رقمهای BCD را ده بار در ثانیه روی نمایشگرهای LED نمایش دهید. نرم افزار انجام عمل فوق در شکل ۶-۱۰ نشان داده شده است. فهرست، برخی از مفاهیمی را که قبلا مورد بحث قرار دادیم، به تصویر کشیده است. جزئیات ارسال داده به MC14499 در زیر روالهای UPDATE و OUT8 فراهم آمده اند. این مثال در سطحی بالاتر، طراحی کاربردهای راه اندازی شده با وقفه را با توضیح فعالیتهای اصلی و فرعی

تشریح می کند ، وقفه های این مثال با MON51 که از وقفه ها در آن استفاده نشده باشد ، سازگارند . برنامه نمایش در قسمت فرعی ، هنگام اجرای برنامه شکل ۶-۱۰ در سطح وقفه در قسمت اصلی ، اجرا می شود . هنگامی که برنامه شروع شود (مثلا با وارد کردن دستور GO8000 در MON51 ، ضمیمه چ را ببینید) ، شرایط برای بهنگام سازی آغاز وقفه برای نمایشگرهای LED ، آماده می شوند ، و سپس کنترل سریعا به برنامه نمایش باز می گردد . فرمانهای نمایش به صورت معمول اجرا می شوند ، در ضمن ، وقفه ها در قسمت اصلی اتفاق می افتند . بطور مثال اگر فرمان SET نمایش برای تغییر محل های 70H و 71H در RAM داخلی مورد استفاده قرار گیرد ، تغییرات فوراً (در عرض ۰/۱ ثانیه) روی نمایشگرهای LED هفت قسمتی دیده خواهند شد .

به ساختار کلی برنامه توجه کنید . بخشهای زیر بترتیب به چشم می خورند :

□ کنترل های اسمبلر (خطوط 1 تا 3)

□ قالب توضیح (خطوط 4 تا 30)

□ تعریف نمادها (31 تا 38)

□ تعیین نوع ذخیره سازی (خطوط 40 تا 42)

□ جدول پرش برای برنامه و نقاط ورود وقفه (خطوط 44 تا 51)

□ بخش اصلی (MAIN ، خطوط 56 تا 59)

جهت خرید فایل word به سایت www.kandoo.cn.com مراجعه کنید
یا با شماره های ۰۹۳۶۶۰۲۷۴۱۷ و ۰۹۳۶۶۴۰۶۸۵۷ و ۰۶۶۴۱۲۶۰-۵۱۱ تماس حاصل نمایید

□ زیر روال سرویس وقفه خارجی (EXTOISR، خطوط 74 تا 77)

□ زیر روال بهنگام سازی نمایشگر LED (UPDATE، خطوط 89 تا 97)

□ زیر روال خروج بایت (OUT8، خطوط 103 تا 113)

□ کد عدم پیاده سازی وقفه ها (خطوط 118 تا 123)

برنامه اجرا در آدرس 8000H در RAM شماره 6264 در SBC-51، نوشته شده است. از

آنجا که بردار وقفه ها در انتهای حافظه است، برنامه نمایش شامل یک جدول پرش برای

جهت دهی وقفه ها به آدرس آغاز 8000H است، نقطه ورود برنامه برای راحتی 8000H

در نظر گرفته می شود، اما یک دستورالعمل LJMP، کنترل را به برجسب MAIN هدایت

می کند. همه دستورالعمل های مقداردهی اولیه در خطوط 56 تا 58 هستند. بخش MAIN

با پرش به برنامه نمایش پایان می یابد.

Filename: Document1
Directory:
Template: C:\Documents and Settings\hadi tahaghoghi\Application
Data\Microsoft\Templates\Normal.dotm
Title:

Subject:
Author: qq
Keywords:
Comments:
Creation Date: 3/28/2012 4:42:00 PM
Change Number: 1
Last Saved On:
Last Saved By: H.H
Total Editing Time: 0 Minutes
Last Printed On: 3/28/2012 4:42:00 PM
As of Last Complete Printing
Number of Pages: 77
Number of Words: 11,294 (approx.)
Number of Characters: 64,378 (approx.)