◆پاک کردن اطلاعات حافظه CMOS

چهار شنبه ۲۴ آبان ۱۳۸۵

در کامپیوترهای XTباتوجه به تعداد محدود پارامترها، پیکربندی سیستم بااستفاده از میکروسوئیچ امکان پذیر می باشد اما در سیستم های AT به دلیل بالا بودن تعداد پارامترهای قابل برنامه ریزی توسط استفاده کننده ، حضور یک منبع که هم به راحتی در دسترس باشد و هم جای کمی اشغال نماید ، ضروری به نظر می رسد. این منبع در کامپیوترهای AT حافظه CMOSنامیده می شود .

حافظه CMOS دارای ۶۴ بایت و یا بیشتر ظرفیت می باشد که توسط دو پورت H70 و H71قابل دسترسی می باشد. و از آن برای نگهداری ساعت ، تاریخ و پیکربندی سیستم استفاده می شود. همچنین تعدادی از بیت های این حافظه برای چک کردن پیکربندی سیستم تحت عنوان CHECKSUMاستفاده می شود .

محتویات این حافظه در زمان خاموش بودن سیستم توسط یک باطری پشتیبان ۳/۶ ولتی نگهداری می شود. این باطری ممکن است در داخل و یا در خارج از مادربورد قرار داشته باشد .

البته در بعضی از سیستمها مجموعه باطری و حافظه به صورت یکپارچه ارائه شده که نمونه آن مارک DALLAS می باشد.

از نظر عملکرد و نحوه دستیابی هیچ تفاوتی بین انواع متفاوت CMOS وجود ندارد و همگی با استفاده از دو پورت یادشده قابل دسترسی و برنامه ریزی می باشند .

تغییر در محتویات CMOS بطور معمول از طریق برنامه SETUP امکان پذیر است اما در صورتی که در ست آپ سیستم رمز تعریف شده باشد و رمز مربوطه را هم در اختیار نداشته باشید در اینصورت امکان ورود به برنامه ست آپ و تغییر در پیکر بندی سیستم (اطلاعات CMOS) امکان پذیر نخواهد بود . در این موارد راهی جز پاک کردن محتویات CMOS نداریم . دراین مواقع در احتمال وجود دارد .

(1برای وارد شدن به سیستم رمز تعریف شده باشد.

(2برای وارد شدن به ست آپ رمز تعریف شده باشد.

در حالت اول با توجه به بوت نشدن کامپیوتر کاری از نرم افزارها ساخته نیست و باید اقدام به پاک کردن محتویات CMOS به صورت سخت افزاری نمود .

این کار معمولا با برداشتن باطری پشتیبان ست آپ برای چند دقیقه ، یااتصال کوتاه بر روی جامپر مربوطه (J8) و یا تعویض تراشهCMOS امکان پذیر می باشد . البته این روشها در صورتی کارآمد خواهد بود که شما مجاز به بازکردن کیس کامپیوتر باشید که البته در اکثر مواقع این امکان وجود ندارد.

شایان ذکر است در صورتی که اعمال فوق درست صورت نگیرد احتمال سوختن و خراب شدن CMOSوجود دارد بنابراین تا حد امکان باید از کاربرد این روش اجتناب شود مگر در مواقعی که ضرورت ایجاب نماید.

در حالت دوم با توجه به بوت شدن کامپیوتر نیازی به اقدامات فوق نبوده و می توان با استفاده از نرم افزارهای مناسب اقدام به پاک کردن و یا ذخیره اطلاعات CMOS نماییم . از آنجایی که همیشه نرم افزار مناسب وجود ندارد و یا در دسترس نیست بعنوان یک مهندس نرم افزار باید قادر باشیم تا با استفاده از امکانات موجود بر روی کامپیوتر این کار را انجام دهیم . با استفاده از برنامه DEBUG.EXE که همراه فایلهای سیستم عامل DOS و ویندوز وجود دارد می توان با خواندن و یا نوشتن در پورت های H70 و H71 تغییرات لازم را در محتویات CMOSاعمال نمود

از دو پورت فوق پورت H70 بعنوان گذرگاه آدرس (ADDRESS BUS)و پورت H71 بعنوان گذرگاه داده بکار می رود. الگوریتم کلی کار به این صورت است که CMOS را بعنوان یک آرایه یک بعدی در نظر می گیریم که دارای ۲۵۶ خانه می باشد . برای دسترسی به هر خانه باید ابتدا اندیس (آدرس) خانه را در پورت H70 بنویسیم و بعد از دسترسی به خانه مورد نظر می توان دادهٔ دلخواه را درآن نوشت و یا از آن خواند . دسترسی به داده ها نیز فقط

از طریق پورت H71 امکان پذیر می باشد . آخرین نکته اینکه عمل نوشتن و یا خواندن از

پورت ها به کمک ثبات AL و توسط دستورهای INو OUT مربوط به زبان اسمبلی امکان

پذیر است.

الگوریتم کلی حذف اطلاعات حافظه CMOS

```
void CLR_CMOS(void){
for(int i=0;i<256;i++){
move value of i to register Al ;
Out Register Al to Port 70h ;
Set register Al to zero ;// Al=0
Out Register Al to Port 71h
}
return;
}
```

پیاده سازی این الگوریتم به کمک زبان اسمبلی بسیار آسان می باشد . اما از آنجا که قرار است

این الگوریتم را با استفاده از debug پیاده سازی کنیم قبل از هر چیز باید با تعدادی از

دستورات این نرم افزار آشنا شویم

با اجرای فایل debug.exe می توانید وارد محیط نرم افزار debug شوید محیط اصلی نرم

افزار یک علامت خط تیره می باشد که بعد از اجرای فایل debug.exe ظاهر می شود.

این نرم افزار دارای تعدادی دستور می باشد که با تایپ یک علامت سئوال و فشردن کلید ENTERمی توان لیست فرمانها را مشاهده نمود . در زیر تعدادی از فرمانها را که برای پیاده سازی این الگوریتم به آن نیاز داریم توضیح می دهیم و سایر فرامین را به خواننده واگذار می کنیم تا درصورت نیاز با مراجعه به راهنمای نرم افزار با نحوه کار هریک آشنا شود

عملکرد دستور

امکان نوشتن کداسمبلی از آدرس مشخص شده در صورت مشخص نکردن آدرس از آدرس موجود در IP بعنوان آدرس شروع استفاده می شود . در ابتدای کار IP=100 است که همان شروع فایلهای Com و یا bin می باشد A [address]

برای نمایش / مقداردهی ثبات ها بکار می رود. در صورتی که بدون پارامتر استفاده شود محتویات کلیه ثبات ها را نشان می دهد اما در صورتی که با نام یک ثبات بکار رود علاوه بر نمایش مقدار فعلی ثبات مورد نظر امکان تغییر محتویات آنرا نیز فراهم می آورد R [register] .

برای مشخص نمودن نام و مسیر فایل ورودی / خروجی بکار می رود N [pathname] .

برای اجرای برنامه تا یک آدرس مشخص بکار می رود . چنانچه بدون پارامتر بکار رود برنامه را از خط جاری (ip فعلی) تا انتهای برنامه اجرا می کند G .

برای نوشتن برنامه از آدرس h100 به تعداد بایتهای مشخص شده در ثبات cx بکار می رود .

W

برای خروج از محیط DEBUG بکار می رود Q

حال با فرض بر این که شما وارد محیط debug شده اید مجموعه دستورات لازم برای پیاده سازی الگوریتم فوق را در زیر می آوریم.

مرحله ۱

با اجرای دستور a وارد مود برنامه نویسی اسمبلی شوید در این صورت debug با نشان دادن آدرس xxxx:0100 آمادگی خود را برای دریافت دستورات اعلام می دارد (در عمل debug بجای xxxx سگمنت مربوط به برنامه را نشان می دهد که توجه به این موضوع اصلاً در اینجا اهمیتی ندارد(.

مرحله ۲

دستورات زیر را با دقت تایپ کرده و در انتهای هر خط کلید ENTER را فشار دهید (مواردی که زیر آن خط کشیده شده از طرف نرم افزار نشان داده می شود این آدرسها در واقع

مشخص کننده طول هر دستور و در نهایت طول برنامه می باشد. (

XXXX:100 MOV CL , FF

XXXX:102 MOV AL , CL

XXXX:104 OUT 70 , AL

XXXX:106 MOV AL , 0

XXXX:108 OUT 71 , AL

XXXX:10A LOOP 102

XXXX:10C INT 20

XXXX:10E

)در صورتی که مایل به درک کار دستورات برنامه نمی باشید به مرحله ۳ بروید(

توضیح خط به خط برنامه:

خط اول : یک شمارنده با مقدار ۲۵۵ در نظر گرفته شده که معادل همان متغیر i در الگوریتم است .

خط دوم al : را برابر cl قرار می دهیم.

خط سوم : محتوای al را به پورت h70 ارسال کردیم (در واقع تنظیم گذرگاه آدرس. (

خط چهارم al : را برابر صفر قرار می دهیم.

خط پنجم : محتوای al را به پورت h71 ارسال می کنیم و در واقع داده موجود در al را در

آدرسی که قبلاً مشخص کرده ایم می نویسیم

خط ششم : از آنجا که عملیات مربوط به تنظیم آدرس و نوشتن داده در محل آدرس باید به تعداد لازم تکرار شود بنابراین از دستور LOOP برای ایجاد یک حلقه با تعداد تکرار مشخص شده در CX استفاده کردیم دستور LOOP 102 باعث می شود تا برنامه در هربار اجرا یک واحد از مقدار CX کم کرده و در صورت منفی شدن دستور بعدی و در غیر اینصورت به آدرس مشخص شده پرش می کند .در این برنامه با توجه به مقدار CX خطوط دوم تا پنجم ۲۵۶ بار اجرا می شوند .

خط هفتم : اجرای این وقفه باعث ختم برنامه خواهد شد

مرحله ۳

بعد از فشردن کلید ENTERدرآخرین مرحله مجدداً اعلان DEBUG ظاهر می شود در این مرحله چنانچه نمی خواهید برنامه را ذخیره کنید در جلوی اعلان)همان علامت منها) دستور G را اجرا کنید تا کلیه محتویات CMOS پاک شود. سیستم را RESETنموده و با فشردن کلید DEL وارد ست آپ سیستم شوید و تغییرات لازم را اعمال نمایید . اما در صورتی که تمایل دارید کد برنامه را به صورت یک فایل اجرایی از نوع COM درآورید مراحل بعدی را انجام دهید

مرحله ۴

با اجرای فرمان R CX در مقابل اعلان DEBUG مقدار ثبات CX را به Eتغییر داده و کلید ENTER را فشار دهید .(دقت داشته باشد که کلیه اعداد در DEBUG در مبنای ۱۶ نوشته می شوند و عددE همان طول برنامه شماست یعنی اختلاف ۱۰ 100 – Eدر مبنای ۱۶)

مرحله ۵

با اجرای فرمان N CLR_CMOS.COM در جلوی اعلان DEBUG نام CLR_CMOS.COMرا برای برنامه خود مشخص کنید (به جای این نام می توانید نام دلخواه خودتان را با پسوند COM جایگزین کنید)

مرحله ۶

با اجرای فرمان W در جلوی اعلان DEBUG تعداد ۱۴ بایت کد نوشته شده در فایلی با نامی که در مرحله قبل مشخص کرده اید ذخیره خواهد شد.

حال شما دارای یک فایل اجرایی ۱۴ بایتی خواهید بود که هر وقت بر روی یک کامپیوتر اجرا شود محتویات حافظه CMOS آنرا پاک می کند.

مرحله 7

در این مرحله می توانید همچون مرحله ۳ عمل نموده و یا با اجرای دستور Q از DEBUG خارج شوید و فایلی را که ساخته اید اجرا نموده و سیستم را RESET کنید و وارد ست آپ شوید.