

فصل دوم

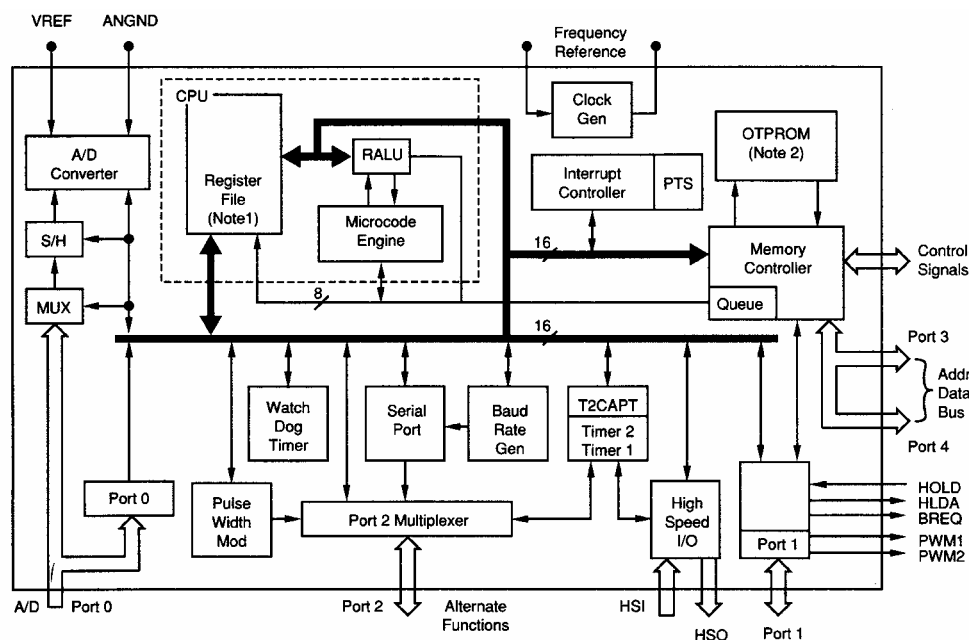
مقدمه ای بر میکرو کنترلرهای 8XC196KC/KD

8XC196KC و 8XC196KD میکروکنترلرهای ۱۶ بیتی CHMOS هستند که برای محاسبات سریع و عملیات ورودی/خروجی سریع طراحی شده‌اند. از اعضای خانواده MCS-96 بوده و از مجموعه دستورالعمل‌ها و معماری مشترکی استفاده می‌کنند. این فصل بررسی گذرای بر معماری و نرم افزار میکروکنترلرهای 8XC196KC/KD خواهد داشت. خانواده MCS-96 به طور معمول در کنترل حلقه بسته^۱ و پردازش سیگنال در سطح متوسط^۲ به کار برده می‌شود. تولیدات خانواده MCS-96 در کاربردهای، کنترل موتورهای^۳، چاپگرها، کنترل ماشین‌ها، دستگاه‌های فتوکپی، ترمزهای ضد قفل^۴، کنترل مطبوع کننده‌های هوا^۵، درایورهای دیسک و ابزار دقیق پزشکی^۶، استفاده می‌شود. شکل ۱-۲ بلوک دیاگرام 8XC196KC و 8XC196KD می‌باشد. هر کدام از این میکروکنترلرها از یک واحد پردازش مرکزی^۷، ۱۶ بیتی بهره می‌برند که از طریق یک گذرگاه^۸ ۱۶ بیتی به یک کنترل کننده وقفه^۹ و یک کنترل کننده حافظه متصل می‌باشد. امتداد این گذرگاه واحد پردازش مرکزی (CPU) را به واحدهای جانبی داخلی متصل می‌کند. به علاوه یک گذرگاه ۸ بیتی CPU برای انتقال بایتهای دستورالعمل از کنترل کننده حافظه^{۱۰} به ثبات دستورالعمل^{۱۱} در RALU منتقل می‌کند.



فایل ثبات در انواع KC و KD متفاوت هستند همانطور که در شکل ۱-۲ ملاحظه می‌شود مقدار این بلوک ۱۰۲۴ بایت در 8XC196KD و ۵۱۲ بایت در 8XC196KC می‌باشد. حافظه داخلی OTPROM نیز در انواع KC و KD متفاوت هستند. تعداد این حافظه ۳۲ کیلو بایت در 8XC196KD و ۱۶ کیلو بایت در 8XC196KC می‌باشد.

- ^۱ . Close Loop
- ^۲ . Mid Range Digital Signal Processing
- ^۳ . Motor Control
- ^۴ . Anti Lock Brakes
- ^۵ . Air Conditioner Control
- ^۶ . Medical Instrumentaion
- ^۷ . CPU (Central Processing Unit)
- ^۸ . Bus
- ^۹ . interrupt Controller
- ^{۱۰} . Memory Controller
- ^{۱۱} . Instruction Register



Notes:

1. 1024 bytes in 8XC196KD, 512 bytes in 8XC196KC.
2. 32K bytes in 8XC196KD, 16Kbytes in 8XC196KC.

شکل ۲-۱. بلوک دیاگرام 8XC196KC/KD

۲-۱. مقایسه 8XC196KD و 8XC196KC با یکدیگر
8XC196KD یک نسخه بسیار سریع از 8XC196KC می باشد که دو برابر 8XC196KC، OPTROM و RAM دارد. جدول (۲-۱) تفاوت های این در میکروکنترلر را با هم مقایسه می کند. 8XC196KD و 8XC196KC از نظر نوع عملکرد و تعداد، واحدهای جانبی یکسانی دارند.

جدول ۲-۱. مقایسه های 8XC196KD و 8XC196KC

ویژگی	8XC196KC	8XC196KD
فضای قابل آدرس دهی حافظه	64 Kbytes	64 Kbytes
RAM داخلی (SFRS ها نیز شامل می شود)	512 bytes	1024 bytes
ROM قابل برنامه ریزی (یکبار)	16 Kbytes	32 Kbytes
پیشینه فرکانس کاری	16 MHZ	20 MHZ

۲-۲. هسته 8XC196KC/KD

هسته 8XC196KC/KD از واحد پردازش مرکزی (CPU)، یک کنترل کننده حافظه و یک کنترل کننده وقفه تشکیل شده است. (شکل ۲-۲ را ملاحظه کنید) و CPU از یک واحد حسابی و منطقی ثبات^۱ و یک فایل ثبات^۲ تشکیل می شود.

۲-۲-۱. کنترل CPU

CPU توسط یک ماشین میکروکد^۳ کنترل می شود که با استفاده از بایت ها و کلمه ها و یا کلمه های مضاعف^۴ فایل ثبات پایینی، یا از طریق دسترسی سیستم به فایل ثبات بالایی، به RALU فرمان انجام عملیاتی را می دهد. دستورالعمل های CPU از صف چهار بایتی^۵ در کنترل کننده حافظه، به ثبات دستورالعمل RALU جابه جا می شود. ماشین میکروکد دستورالعمل ها را رمزگشایی^۶ کرده و مجموعه عملیاتی که برای اجرای دستورالعمل لازم است، اجرا می کند.

۲-۲-۲. فایل ثبات

فایل ثبات به دو قسمت بالایی و پایینی تقسیم می شود فایل ثبات پایینی در برگزیده ۲۴ بایت از ثبات های با وظیفه خاص^۷ (SFR) و ۲۳۲ بایت از ثبات های همه منظوره RAM می باشد. فایل ثبات بالایی تنها در برگزیده ثبات های RAM همه منظوره (۲۵۶) بایت در 8XC196KC و ۷۸۶ بایت در 8XC196KD) می باشد. دستیابی به ثبات RAM همه منظوره می تواند به صورت بایت ها، کلمه ها یا کلمات مضاعف، صورت گیرد.

RALU^۸ به صورت متفاوتی به فایل های ثبات بالایی و پایینی دسترسی دارد. دسترسی به فایل ثبات پایینی همواره به طور مستقیم از طریق روش آدرس دهی مستقیم ثباتی است. (روش های آدرس دهی در فصل سوم را ملاحظه کنید) دستیابی به فایل ثبات بالایی از طریق روش آدرس دهی مستقیم ثباتی تنها زمانی امکان پذیر است که دستیابی پنجره ای به صورت عمودی فعال باشد. دستیابی پنجره ای به صورت عمودی یک تکنیک برای نگاشت بلوک های فایل ثبات بالایی در فایل ثبات پایینی می باشد. فصل چهارم انواع داده ها و آدرسها را برای اطلاعات بیشتر در مورد فایل ثبات و دستیابی پنجره ای ملاحظه کنید.

^۱ . RALU: Register Arithmetic Logic Unit

^۲ . Register File

^۳ . Microcode engine

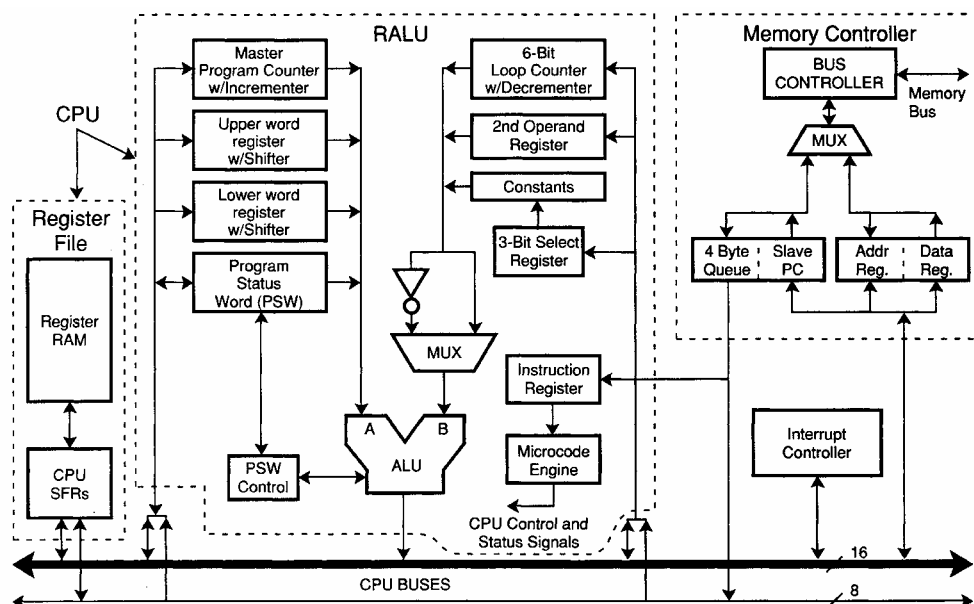
^۴ . double word

^۵ . Four byte queue

^۶ . Four byte queue

^۷ . Special Function Register

^۸ . Register Arithmetic Logic Unit



شکل ۲-۲. بلوک دیاگرام هسته 8XC196KC/KD

۲-۲-۳. واحد حسابی و منطقی ثابت

RALU در برگرفته یک واحد حسابی و منطقی (ALU) ۱۷ بیتی، کلمه وضعیت برنامه^۱ (PSW)، شمارنده برنامه^۲ (PC)، ثابت دستورالعمل، ماشین میکروکد، یک ثابت ثابت^۳، یک ثابت انتخاب کننده سه بیتی، شمارنده حلقه ای^۴ و سه ثابت موقت (ثبات های کلمه بالایی، کلمه پایینی و ثابت عملوند دوم) می باشد. تمام ثبات ها به جز ثبات انتخاب کننده سه بیتی، ۱۶ بیتی یا ۱۷ بیتی می باشند (۱۶ بیتی به علاوه یک بیت علامت) بعضی از این ثبات ها می توانند سرعت ALU را با ساده کردن عملیات کم کننده، کلمات از طریق ورودی های A و B به ALU وارد می شود.

RALU می تواند با ذخیره کردن مقادیر ثابت در ثبات ثابت به محاسبات سرعت می بخشد. به این صورت که هنگام مکمل سازی و افزودن یک واحد یا کاهش یک واحد از بایت یا کلمات این مقادیر ثابت به سادگی در دسترس قرار دارند PSW در برگرفته یک بیت (PSW.1) می باشد که سرویس دهی به تمام وقفه های قابل پوشش را به طور عمومی فعال یا غیرفعال می کند و بیت (PSW.2) نیز سرویس دهنده رابط جانبی^۵ (PTS) را فعال یا غیرفعال می کند و

^۱. Program Status Word

^۲. Program Counter

^۳. Constant Register

^۴. Loop Counter

^۵. Peripheral Transaction Server

شش پرچم بولی^۱ دیگر نشان دهنده وضعیت برنامه کاربر می باشد. جزئیات بیشتر در خصوص ثبات PSW در ضمیمه C آمده است. *شمارنده برنامه (PC)* در برگزیده آدرس دستورالعمل بعدی و نیز سخت افزاری است که به طور اتوماتیک آدرس های بعدی را در آن قرار می دهد. اگر یک پرش، وقفه، فراخوانی یا بازگشت، ترتیب آدرسی را عوض کند، ALU آدرس مناسب را در PC قرار خواهد داد. ثبات های کلمه بصورت ترکیبی در دستورالعمل های ۳۲ بیتی و به عنوان ثبات های موقت برای بسیاری دیگر از دستورالعمل ها مورد استفاده قرار می گیرد. از آنجائیکه ثبات ها خود دارای قابلیت شیفت منطقی هستند، RALU نیز از آنها در عملیاتی که نیاز به شیفت های منطقی دارد استفاده می کند (برای مثال، نرمالیزه کردن، ضرب و تقسیم) ثبات کلمه پایینی تنها زمانی مورد استفاده قرار می گیرد که کمیت های Double-Word در حال شیفت دادن باشد و ثبات کلمه بالایی برای انجام هر شیفتی استفاده می شود. شیفت های متعدد به وسیله شمارنده حلقه ای ۶ بیتی شمرده می شود. ثبات عملوند دوم، زمانی که میکروکد یک دستورالعمل دو عملوندي را اجرا می کند، عملوند دوم را در خود ذخیره می کند. همچنین هنگام ضرب، مضروب^۲ و هنگام تقسیم، مقسوم علیه^۳ را شامل می شود. هنگام تفریق، خروجی این ثبات، مکمل می شود قبل از اینکه به ورودی B در (ALU) وارد شود.

۱-۲-۲-۲. اجرای کد

RALU بیشترین محاسبات را برای 8XC196KC/KD انجام می دهد. این واحد فقط از یک انباره^۴ استفاده نمی کند. بلکه به جای آن به طور مستقیم بر روی فایل ثبات پایینی عمل می کند، که در اصل این فایل ثبات پایینی ۲۵۶ انباره را فراهم می کند. چون در این حالت برنامه به جای یک انباره از طریق چندین انباره اجرا می شود، می توان گفت در 8XC196KC/KD برنامه ها بسیار سریعتر و موثرتر انجام می شود.

برای نمونه، برنامه 80C196 که در ادامه آمده است دو متغیر ۱۶ بیتی (FACTOR-2) و (FACTOR-1) را در هم ضرب می کند و نتیجه ۳۲ بیتی حاصل از آن را در یک متغیر سوم (RESULT) ذخیره می کند.

MOV	AX,FACTOR-1	;AX ← FACTOR-1
MUL	AX,FACTOR-2	;AX ← FACTOR-1 X FACTOR-2
MOA	RESULT,AX	;AX ← Lower Byte
MOV	RESULT+2,DX	;Dx ← Upper Byte

مثال بعدی برنامه معادلی را برای مثال قبل نشان می دهد.

^۱ . Boolean Flag

^۲ . Multiplier

^۳ . Divider

^۴ .Accumolator

MUL RESULT,FACTOR-1,FACTOR-2 ;RESULT ← FACTOR-1 X FACTOR-2

8XC196KC/KD می‌تواند این عملیات را در یک دستورالعمل انجام دهد، به این دلیل که تمام ثباتها در فایل ثبات می‌توانند به عنوان عملوند مستقیماً مورد استفاده قرار گیرند و در این دستورالعمل ۳ ثبات به طور همزمان در یک دستورالعمل ۳ عملوندي استفاده شده است. این فرمت اجازه می‌دهد که یک دستورالعمل، دو ثبات مبدأ و یک ثبات مقصد را بطور همزمان آدرس‌دهی کند. ملاحظه می‌شود که خانواده MCS96 دارای معماری ثبات به ثبات بوده و در آن به انباره نیازی نیست. اکثر عملیات را می‌تواند مستقیماً روی ثباتها انجام دهد.

۴-۲-۲. کنترل کننده حافظه^۱

RALU از طریق کنترل کننده حافظه با تمام حافظه به جزء فایل ثبات در ارتباط است. RALU از طریق کنترل کننده حافظه با فایل ثبات بالایی در ارتباط است (مگر زمانی که از پنجره عمودی استفاده می‌شود، فصل چهارم را ملاحظه کنید). کنترل کننده حافظه شامل ثبات‌های آدرس و داده، یک صف چهار بیتی، یک شمارنده Slave PC و یک کنترل کننده گذرگاه است. کنترل کننده گذرگاه گذرگاه حافظه را راه‌اندازی می‌کند و شامل گذرگاه OPTROM داخلی، گذرگاه RAM داخلی و گذرگاه خارجی آدرس/داده می‌باشد. کنترل کننده گذرگاه، تقاضای دستیابی به حافظه را هم از RALU و هم از صف چهار بیتی دریافت می‌کند. تقاضای صف چهار بیتی دارای اولویت بالاتری است. این صف قابل دسترسی برای کاربر و RALU نمی‌باشد.

زمانی که کنترل کننده گذرگاه تقاضایی را از صف دریافت می‌کند. کد مورد نظر را از آدرسی که شمارنده Slave PC به آن اشاره می‌کند، واکنشی می‌کند. این عمل باعث افزایش سرعت اجرا می‌شود چون دستورالعمل بعدی بلافاصله حاضر است و پردازنده منتظر نمی‌شود تا شمارنده Master PC آدرس را به کنترل کننده حافظه بفرستد. اگر یک پرش (JUMP)، وقفه، فراخوانی یا بازگشت از زیربرنامه ترتیب آدرس را عوض کند، شمارنده Master PC آدرس جدید را در شمارنده Slave PC بار می‌کند. صف پاک می‌شود و پردازش ادامه پیدا می‌کند.



زمانی که از لاجیک آنالایزر برای Debug برنامه استفاده می‌کنید. به یاد داشته باشید که دستورالعمل ابتدا در صف چهار بیتی بار می‌شود و لزوماً بعد از واکنشی بلافاصله اجرا نمی‌شود.

^۱ .Memory Controller

۵-۲-۲. کنترل کننده وقفه

کنترل کننده وقفه قابل برنامه ریزی دارای یک ترتیب اولویت بندی سخت افزاری است که کاربر می تواند این ترتیب را تغییر دهد. این وقفه ها به وسیله زیربرنامه های روتین وقفه که توسط کاربر نوشته می شود سرویس دهی می شود. به علاوه، 8XC196KC/KD یک پردازنده وقفه به صورت سخت افزار میکرو کد شده را به نام سرویس دهنده رابط جانی (PTS) ارائه کرده است. PTS به وقفه ها با یک مجموعه عملیات از قبیل تعیین شده و بصورت سخت افزاری سرویس می دهد. برای مثال انتقال داده، آغاز یک تبدیل A/D خواندن FIFO در واحد ورودی سریع و ثبت رخدادها^۱ در واحد خروجی سریع از این نمونه ها هستند. PTS این سرویس دهی را بسیار سریعتر از زیر برنامه های نرم افزاری وقفه ها انجام می دهد. PTS می تواند تمامی وقفه ها به جزء NMI^۲، تله نرم افزاری^۳ و کد عملیاتی پردازش نشده^۴ را سرویس دهی کند. سیکل های PTS^۵ نسبت به وقفه های استاندارد اولویت بالاتری دارند. و ممکن است زیر برنامه سرویس وقفه را در حالت معلق نگه دارند. برای اطلاعات بیشتری در خصوص چگونگی عملکرد PTS فصل پنجم را ملاحظه کنید.

۳-۲. زمان بندی داخلی

تولید کننده Clock داخلی فرکانس سیگنال XTAL را نصف می کند و دو سیگنال زمانی داخلی PH1 و PH2 را این سیگنال ها زمانی فعال با سطح منطقی بالا هستند. لبه های بالا رونده PH1 و PH2 شکل موج CLKOUT را تولید می کند شکل (۳-۲) چگونگی ساختن CLKOUT را نمایش می دهد تولید می کند. ترکیب تناوب های PH1 و PH2 تعیین کننده واحد زمان مبنا می باشد که به نام **زمان مقرر**^۶ شناخته می شود. در بیشترین فرکانس 8XC196KD که 20MHZ مگا هرتز می باشد. یک زمان مقرر زمانی معادل 100ns نانو ثانیه است. در بیشترین فرکانس 8XC196KC که 16MHZ مگا هرتز می باشد. یک زمان مقرر برابر 125ns نانو ثانیه است. چون 8XC196KC/KD می تواند در فرکانس های مختلفی کار کند در این کتاب زمان های لازم بطور دقیق مشخص نشده و بر حسب زمان مقرر تعیین خواهد شد.

^۱.Event

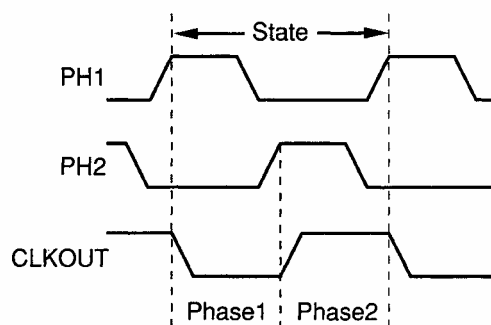
^۲.Non Maskable Interrupt

^۳.Software Trap

^۴.Unimplemented Opcode

^۵.PTS Cycle

^۶.State time



شکل ۲-۳. فازهای Clock داخلی

بیت غیرفعال کننده CLKOUT در 8XC196KC و 8XC196KD به ثبات IOC3 افزوده شده است. یک کردن این بیت موجب غیرفعال شدن سیگنال CLKOUT می‌شود که خود نویز سیستم را کاهش می‌دهد. این ویژگی در نسخه‌های اولیه از 8XC196KC وجود ندارد. به توضیحات ثبات IOC3 به ضمیمه C مراجعه کنید.

۲-۴. واحدهای جانبی داخلی

واحدهای جانبی داخلی توانایی‌های مختلفی را در اختیار کاربران قرار می‌دهد. که در ادامه به بررسی این واحدها خواهیم پرداخت.

۲-۴-۱. درگاه‌های استاندارد I/O

8XC196KC/KD پنج درگاه هشت بیتی دارد. بعضی از آنها فقط ورودی، بعضی دیگر فقط خروجی و بعضی دیگر دو طرفه^۱ هستند و بعضی دیگر چند منظوره هستند. درگاه PORT0 یک درگاه ورودی است که

برای مبدل A/D نیز به عنوان ورودی آنالوگ عمل کند. PORT1 یک درگاه دو طرفه است. پایه‌های این درگاه سیگنال‌های کنترلی گذرگاه و دو خروجی واحد مدولاتور عرض پالس^۲ مالتی پلکس شده‌اند. PORT2 شامل نوع پایه می‌باشد: دو طرفه، ورودی و خروجی، در 8XC196KC/KD چندین عملکرد را بصورت مالتی پلکس شده انجام می‌دهد.

PROT3, PROT4 درگاه‌های درین باز^۳ دو طرفه هستند که پایه‌های آن با گذرگاه آدرس و داده مالتی پلکس شده است. برای اطلاعات بیشتر درخصوص درگاه‌های I/O به فصل ششم مراجعه کنید.

^۱. Bidirectional

^۲. Pulse Width Modulator

^۳. Open Drain

۲-۴-۲. درگاه ورودی و خروجی سریال

درگاه I/O سریال یک درگاه سنکرون و آسنکرون است که در برگرفته یک گیرنده و فرستنده غیر همزمان عمومی^۱ (UART) می باشد. UART در برگرفته یک حالت عملکرد سنکرون و سه حالت آسنکرون است. حالت های عملکرد آسنکرون Full duplex هستند به این معنی که می توانند ارسال و دریافت همزمان داشته باشند. گیرنده واحد سریال در 8XC196KC/KD دو بافر دارد به طوریکه دریافت دومین بایت می تواند قبل از خوانده شدن بایت اولی صورت بگیرد. فرستنده واحد سریال نیز دارای دو بافر است که می تواند ارسال پیوسته ای را به وجود آورد. فصل هفتم اطلاعات بیشتری را درخصوص درگاه سریال در اختیار خوانندگان قرار خواهد داد.

۲-۴-۳. واحد (HSIO) ورودی خروجی سریع

واحد HSIO در برگرفته چهار واحد جانبی می باشد: High Speed Input, Timer2, Timer1, High Speed Output این واحدها همراه با هم، یک سیستم High timer/Counter قدرتمند را شکل می دهند.

۲-۴-۳-۱. تایمرها

تایمر ۱ در هر هشت زمان مقرر یک بار اضافه می شود. تایمر ۲ زمان مبنای واحد HSI است و همچنین می تواند آن را برای زمان مبنای واحد HSO نیز انتخاب کرد. تایمر ۲ می تواند هم با لبه های بالا رونده و هم با لبه های پایین رونده پالس ساعت اعمالی شمارش را انجام دهد این تایمر به عنوان مبنای زمانی واحد HSO، یک شمارنده صعودی یا نزولی، و به عنوان یک تایمر اضافی می تواند مورد استفاده قرار می گیرد.

۲-۴-۳-۲. واحد ورودی سریع HSI

واحد HSI می تواند زمان وقوع رخ داده های خارجی را با دقت هشت زمان مقرر ضبط کند. این واحد دارای چهار پایه ورودی قابل تنظیم و مستقل از یکدیگر می باشد که در زمان وقوع یک حادثه مقدار تایمر ۱ را ثبت^۲ می کند. رخ داده ای که سبب تسخیر شدن مقدار تایمر ۱ می شوند عبارتند از: لبه های بالا رونده، لبه های پایین رونده، لبه های بالا رونده و پایین رونده و هشت لبه بالارونده واحد HSI قابلیت ذخیره کردن تا سقف هشت رخداد (ثبت مقدار تایمر ۱) را دارد. هفت مقدار در هفت محل FIFO^۳ و مقدار هشتم در ثبات نگهدارنده HSI قرار می گیرد.

^۱ . Universal Asynchronous Receiver and Transmitter

^۲ . Capture

^۳ . First Input First Output

۲-۴-۳-۳. واحد خروجی سریع HSO

واحد HSO می‌تواند موجب تحریک رخ داد در زمانی شود که یا تایمر ۲ آن را مشخص می‌کنند. این رخ داده‌ها می‌توانند شروع تبدیل A/D، ریست کردن تایمر ۲، تولید چهار تایمر نرم افزاری و یک کردن یا صفر کردن پایه‌های خروجی HSO را شامل شوند. واحد HSO نوع رخ داد و زمان معین حادث شدن آن رخ داد را در فایل CAM ذخیره می‌کند. این فایل تا سقف هشت فرمان را ذخیره می‌کند که هر فرمان مشخص کننده زمان حادث شدن رخ داد و نوع آن رخ داد است.

۲-۴-۴. مبدل آنالوگ به دیجیتال

مبدل آنالوگ به دیجیتال یک ورودی آنالوگ را به معادل دیجیتال آن تبدیل می‌کند. که دقت آن می‌تواند هشت بیتی یا ۱۰ بیتی باشد. زمان نمونه برداری و تبدیل قابل برنامه‌ریزی است. یک روش آسان برای تبدیل اتوماتیک و ذخیره‌سازی آن استفاده از واحد PTS است.

اصلی‌ترین قسمت‌های مبدل A/D قسمت نمونه برداری و نگه‌داری مالی پلکسر هشت کاناله و یک مبدل آنالوگ به دیجیتال با الگوریتم تخمین موفقیت‌آمیز که دقت ۸ بیتی یا ۱۰ بیتی دارد می‌باشد.

۲-۴-۵. مدولاتور عرض پالس PWM

8XC196KC/KD دارای سه واحد PWM می‌باشد شکل موج خروجی هر کدام از این واحدها سیکل کار^۱ متغییری دارد که می‌تواند در هر ۲۵۶ یا ۵۱۲ زمان مقرر یک بار رخ دهد. از واحد PWM می‌توان در کنترل انواع موتورهای و درایورهای مبدل DC به AC استفاده کرد به عنوان مثال برای کنترل یک موتور DC زمانی که خروجی PWM را فیلتر کنیم یک سطح DC بوجود می‌آید که می‌تواند در ۲۵۶ سطح مختلف با تغییر دادن سیکل کار PWM عوض شود. فصل ۱۰ به بررسی دقیقتر این واحد می‌پردازد.

۲-۴-۶. تایمر نگهبان^۲

تایمر محافظ یک تایمر داخلی است که هنگام عدم موفقیت نرم افزار در انجام عملیات، سیستم را ریست می‌کند. فصل ۱۱ به بررسی این واحد خواهد پرداخت.

۲-۵. حالت‌های عملیاتی ویژه

علاوه بر حالت عملکرد نرمال که برای اجرای برنامه وجود دارد. 8XC196KC/KD در چند حالت عملکرد ویژه دیگر نیز عمل می‌کند. حالت عملکرد Idle و حالت عملکرد کم‌مصرف با غیرفعال

^۱.Duty cycle

^۲.Watch Dog Timer WDT

کردن بخشهای مختلف توان مصرفی را کاهش می‌دهد. حالت عملکرد ONCE، 8XC196KC/KD را به صورت الکتریکی از سیستم ایزوله می‌کند و چندین حالت عملکرد دیگر برای برنامه‌ریزی حافظه OTPROM وجود دارد. فصل ۱۲ حالت‌های عملیاتی ویژه درباره حالت عملکرد Idle، حالت عملکرد کم‌مصرف و حالت عملکرد ONCE جزئیات بیشتر را ارائه می‌کند.

۲-۵-۱. کاهش توان مصرفی

در حالت عملکرد Idle، CPU از اجرای دستورالعملها جلوگیری می‌کند، اما پالس ساعت واحدهای جانبی فعال باقی می‌ماند. توان مصرفی برنامه اجرایی را تا ۴۰٪ کاهش می‌دهد. اگر سخت افزار ریست شود یا منبع متقاضی وقفه فعال شود. میکروکنترلر از حالت عملکرد Idle خارج می‌شود. در حالت عملکرد کم‌مصرف تمام پالس ساعت‌های داخلی در سطح منطقی صفر متوقف می‌شوند و نوسان ساز قطع می‌شود. RAM داخلی و بیشتر واحدهای جانبی مقادیر خود را حفظ می‌کنند اگر Vcc وصل باشد توان مصرفی حد چند میکرو وات کاهش می‌یابد.

۲-۵-۲. تست کردن برد مدارچاپی

حالت عملکرد ONCE به صورت الکتریکی 8XC196KC/KD را از سیستم ایزوله می‌کند. با استفاده از حالت عملکرد ONCE می‌توانید برد مدارچاپی را زمانی که 8XC196KC/KD روی آن سوار شده است، تست کنید.

۲-۵-۳. برنامه‌ریزی حافظه داخلی در 8XC196KC/KD

8XC196KC/KD به هنگام برنامه‌ریزی، حالت‌های عملکرد ویژه‌ای را حمایت می‌کند. از جمله این حالت‌های عملکرد می‌توان به موارد زیر اشاره کرد:

- برنامه‌ریزی اتوماتیک
- برنامه‌ریزی تحت نظر
- برنامه‌ریزی در حال اجرا

۱- حالت عملکرد برنامه‌ریزی اتوماتیک

طوری 8XC196KC/KD را فعال می‌کند که خودش را با استفاده از یک حافظه خارجی برنامه‌ریزی می‌کند یعنی برنامه یک حافظه خارجی را در حافظه داخلی خود می‌نویسد.

۲- حالت عملکرد برنامه‌ریزی تحت نظر

حالت عملکرد برنامه‌ریزی تحت نظر به یک برنامه ریز خارجی اجازه می‌دهد که حافظه داخلی را برنامه‌ریزی می‌کند هنگامی استفاده از این حالت عملکرد شما می‌توانید هر کلمه را به تنهایی در OPTROM برنامه‌ریزی کند و یا آن را تغییر دهید.

۳- برنامه‌ریزی در حال اجرا

برنامه‌ریزی در حال اجرا به شما اجازه می‌دهد که عملیاتی مختلفی از OPTROM به طور جداگانه هنگام اجرای برنامه تحت یک کنترل کامل نرم افزاری، برنامه‌ریزی کنید.

۲-۶. مقدمه‌ای بر نرم افزار

این بخش نگاهی گذرا بر مجموعه‌ای دستورالعمل‌های MCS-96 دارد و به بررسی تفاوت‌ها بین دستورالعمل‌های 8XC196KC/KD و 8096BH می‌پردازد.

ضمیمه A مرجع اطلاعاتی مفیدی درباره مجموعه دستورالعمل‌های 8XC196KC/KD می‌باشد. در این ضمیمه همچنین به توضیحات، کد عملیاتی در مبنای شانزده، طول دستورالعمل، زمان لازم برای اجرای دستورالعمل، و ارتباط بین پرچم‌های کلمه وضعیت برنامه و دستورالعمل‌ها را شامل می‌شود و ضمیمه C نیز توضیح جزئیات PSW و SFRها را ارائه می‌کند. انواع داده‌ها و روش‌های آدرس‌دهی در فصل ۳ توضیح داده شده است.

۲-۶-۱. نگاهی گذرا بر مجموعه دستورالعمل‌های MCS-96

مجموعه دستورالعمل‌های^۱ MCS-96 شامل یک مجموعه کامل از عملیات حسابی و منطقی برای انواع داده‌های ۸ بیتی و ۱۶ بیتی. این مجموعه دستورالعمل‌ها داده‌های ۳۲ بیتی را نیز در عملوندهایی^۲ که شیفت پیدا می‌کنند مانند تقسیم ۳۲ بر ۱۶ و ضرب ۱۶ در ۱۶ پشتیبانی می‌کند. سایر عملیات بر روی یک متغیر ۳۲ بیتی می‌تواند با استفاده از ترکیب عملیات ۱۶ بیتی مورد پردازش قرار گیرد.

برای مثال در برنامه زیر با استفاده از یک رشته عملیاتی ۱۶ بیتی یک جمع ۳۲ بیتی و یک تفریق ۳۲ بیتی را به ترتیب انجام می‌دهند.

AX,CX ; (ADD-2op)	ADD
BX,CX	ADDC
AX,CX ; (SUB-2op)	SUB
BX,DX	SUBC

مجموعه دستورالعمل‌ها، می‌توانند انواع داده‌ها را به یکدیگر تبدیل کنند. دستورالعمل LDBZE یک بایت را به کلمه تبدیل می‌کند. دستورالعمل CLR می‌تواند یک کلمه را با پاک کردن (نوشتن صفر) در کلمه^۳ بالایی به کلمه مضاعف کلمه تبدیل کند. دستورالعمل LDBSE یک SHORT-INTEGER را به یک INTEGER تبدیل

^۱.Instruction Set

^۲.Operand

^۳.WORD

می‌کند و دستورالعمل EXT یک متغیر از نوع INTEGER را به یک متغیر از نوع LONG-INTEGER تبدیل می‌کند.

دستورالعمل‌های MCS-96 برای جمع و تفریق و مقایسه^۱ تفاوتی بین کلمات بدون علامت و متغیرهای صحیح INTEGERS با علامت قائل نمی‌شود. اما دستورالعمل‌های پرش شرطی این اجازه را می‌دهند که با نتایج این عملیات (جمع و تفریق و مقایسه) به صورت کمیت‌هایی علامت دار و بدون علامت برخورد شود.

برای مثال دستورالعمل CMPB برای مقایسه کمیت‌های هشت بیتی علامت دار و بدون علامت استفاده می‌شود. شما می‌توانید از دستورالعمل JH برای عملوندهای بدون علامت یا دستورالعمل JGT برای عملوندهای علامت‌دار استفاده کنید.

سخت‌افزار MCS-96 به طور مستقیم نمی‌تواند عملیات بر روی اعداد حقیقی (اعشاری) را پشتیبانی کند. کتابخانه اعداد اعشاری در (FPAL96)8XC196KC/KD مجموعه دستورالعمل‌هایی که قابلیت عملیات روی اعداد اعشاری را دارند را پشتیبانی می‌کند. این دستورالعمل‌ها توسط زیر برنامه‌هایی که از طرف IEEE برای عملیات بر روی اعداد اعشاری پیشنهاد شده است مورد پردازش قرار می‌گیرد.

دستورالعمل NORML و بیت پرچم ST در PSW باعث افزایش کارآیی اینگونه نرم‌افزارها شده است. دستورالعمل NORML یک متغیر ۳۲ بیتی را نرمالیزه می‌کند. و بیت پرچم ST می‌تواند همراه با پرچم Carry با دقت را زیاد این متغیر را گرد می‌کند.

۲-۶-۲. دستورالعمل‌های اضافه شده به مجموعه

دستورالعمل‌های MCS-96

در این قسمت مجموعه دستورات اضافه شده به اعضاء جدید خانواده MCS-96 بررسی می‌شود. از جمله، دستورات جدید در خانواده‌های 8XC196KD و 8XC196KC مورد بررسی قرار می‌گیرد. ضمیمه A اطلاعات کاملی را در این خصوص ارائه می‌کند. در زیر دستورالعمل‌های جدید در خانواده‌های KD و KC را ملاحظه می‌کنید:

- **BMOV انتقال بلوکی اطلاعات،** این دستور یک مجموعه اطلاعات را از یک محل حافظه به محل دیگر منتقل می‌کند. در طول اجرای این دستورالعمل تقاضای وقفه نمی‌تواند پذیرفته شود.
- **BMOVI** این دستور نیز کاملاً مشابه دستور BMOV است با این تفاوت که در طول اجرای این دستور تقاضای وقفه می‌تواند پذیرفته شود.
- **CMPL** دو متغیر از نوع کلمه مضاعف (۳۲ بیتی بردن علامت) را از نظر دامنه با هم مقایسه می‌کند.

^۱. Compare

- DJNZW مقدار يك عملوند كلمه را يكي كم مي‌کند و چنانچه پس از کاهش مقدار عملوند صفر شود عمليات پرش انجام نمی‌گیرد و در غير اين صورت عمليات پرش انجام می‌گیرد.
- DPTS اين دستورالعمل بيت PSW.2 را صفر می‌کند، به عبارت ديگر واحد سرويس دهنده تبادلات جانبي PTS را غيرفعال می‌کند.
- EPTS اين دستورالعمل بيت PSW.2 را يك می‌کند و واحد سرويس‌دهنده تبادلات جانبي PTS را فعال می‌کند.
- IDLPD بسته به مقدار عملوند اين دستورالعمل می‌تواند سيستم را به حالت عملکرد IDLE يا حالت عملکرد کم‌مصرف ببرد و يا باعث انجام پروسه ريست در سيستم شود.
- POPA به جاي دستور POPF استفاده می‌شود و براي حمايت از ۸ منبع وقفه جديد در نظر گرفته شده اين دستور دو كلمه را از حافظه پشته فراخواني می‌کند. که اين دو كلمه مجموعاً چهار ثبات WSR/INT_MASK1, INT_MASK / PSW را از حافظه فراخواني می‌کند.
- PUSHA اين دستور به جاي دستور PUSHF در نسخه‌هاي قديمي و به منظور حمايت از ۸ منبع وقفه جديد در نسخه‌هاي KD و KC استفاده می‌شود. اين دستور دو كلمه را در حافظه پشته ذخيره می‌کند. اين دو كلمه شامل چهار ثبات WSR / INT_MASK1, INT_MASK / PSW می‌باشد. پس از ذخيره کردن اين دستورالعمل سه ثبات PSW, INT_MASK و INT_MASK1 را پاک می‌کند.
- TIJMP اين دستور ابتدا يك آدرس را از جدول آدرس‌ها انتخاب می‌کند و سپس آدرس محل پرش را محاسبه کرده و به آنجا می‌برد. اين دستور زمان لازم براي دسترسي به يك جدول اطلاعات را کم می‌کند.
- XCH اين دستور مقدار يك عملوند ۱۶ بيتي مبدأ را با مقدار يك عملوند ۱۶ بيتي مقصد جابه‌جا می‌کند.
- XCHB اين دستور مقدار يك عملوند ۸ بيتي مبدأ را با مقدار يك عملوند ۸ بيتي مقصد جابجا می‌کند.

۳-۶-۲. تفاوت‌هاي موجود در مجموعه دستورالعمل‌ها

دستورالعمل‌ها در 8XC196KC/KD بسيار سريعتر از نسخه 8096BH اجرا می‌شود. به عنوان مثال دستورالعمل ضرب تقريباً دو برابر سريعتر انجام می‌شود. ضرب ۱۶ در ۱۶ که در 8096BH، ۲۵ زمان مقرر طول می‌کشد در حالیکه 8CX196KC/KD فقط در ۱۴ زمان مقرر اين عمليات را انجام دهد. بسياري از دستورالعمل‌هاي که يك عملوند يا هيچ عملوندي ندارند، و بيشتر دستورالعمل‌هاي که از داده‌هاي خارجي استفاده می‌کنند، يك يا دو زمان مقرر در 8CX196KC/KD نسبت به 8096BH زودتر اجرا می‌شوند.

دسترس‌ی غیرمستقیم^۱ و شاخص‌دار^۲ به اشاره‌گر پشته^۳ در 8CX196KC/KD با 8096BH1 متفاوت عمل می‌کنند. در 8096BH1 مبنای آدرس محاسبه شده مقدار اشاره‌گر پشته قبل از هنگام^۴ شدن آن است ولی در 8CX196KC/KD مقدار هنگام شده اشاره‌گر پشته مورد استفاده قرار می‌گیرد.

۲-۶-۴. استانداردها و قراردادهای نرم‌افزاری

برای یک پروژه نرم‌افزاری در هر اندازه‌ای که باشد قطعه‌بندی پیکره برنامه و استاندارد کردن آن برای کنترل ارتباط بین این قطعه‌ها ایده خوبی است. این استاندارد بسته به کاربرد نهایی تغییر می‌کند. اما همه استانداردها باید شامل سازوکاری باشند که بتواند پارامترها را به زیربرنامه‌های مختلف منتقل کند و نتایج حاصل آن زیربرنامه‌ها را برگرداند. اتصال قطعه برنامه‌ها به منظور ساختن برنامه اصلی اصطلاحاً لینک کردن نامیده می‌شود. قراردادهای این کار را می‌توانید در راهنمای زبان PLM-96 بیابید.

این قرارداد استاندارد کارآمدی برای زبان اسمبلی و محیط PLM-96 است که سازگاری بین این محیط‌ها را تضمین می‌کند. همچنین به برنامه‌نویس اجازه دستیابی به کتابخانه اعداد اعشاری (FPAL-96) که PLM-96 از آن برای عمل بر روی اعداد حقیقی (اعشاری) استفاده می‌کند را می‌دهد.

۲-۶-۴-۱. استفاده از ثبات‌ها

معماری MCS-96 دارای یک فایل ثبات ۲۵۶ بیتی موسوم به فایل پائینی می‌باشد که بعضی از این ثبات‌ها برای ارتباط با وسایل I/O مورد استفاده قرار می‌گیرند و بعضی دیگر کاربردهای خاص دارند

Zero Register و اشاره‌گر پشته. مابقی بایتهایی فایل ثبات پائینی (۲۳۲ بایت دیگر) می‌توانند متغیرهای استفاده شده در برنامه باشند. (به عنوان انباره از آنها استفاده می‌شود) طبق استاندارد PLM-96 ۸ محل حافظه از این فضا یعنی آدرس‌های بین 1CH تا 23H به عنوان محلهای ذخیره موقت و بنام PLMREG استفاده می‌شود. ثبات‌ها با کاربرد خاص (SFR) می‌توانند بصورت بایتهای یا کلمات باشند.

هنگامی که از یک SFR به عنوان مبدا/یک عملوند یا یک مبنای یک ثبات شاخص برای عملیات شاخص‌دار یا غیرمستقیم استفاده می‌شود، نتایج غیرمنتظره‌ای می‌تواند اتفاق بیفتد، پس باید بسیار دقت کرد. به این دلیل که وقایع خارجی می‌تواند SFRها

^۱. Indirect Addressing Mode

^۲. Indexed

^۳. Stack Pointer SP

^۴. Update

۵. Floating-Point-Library

را تغییر دهد همچنین خواندن بعضی از SFRها باعث پاک شدن آنها می‌شود. این مشخصات (پاک شدن یا تغییر کردن) هنگام استفاده از زبان‌های سطح بالا (مثلاً زبان C) باید مورد توجه قرار گیرد. اگر چه زبانهای سطح بالا از اغلب اوقات اجازه دسترسی به تمام SFRها را به کاربر نمی‌دهند.

۲-۴-۴-۲. آدرس‌دهی عملوندهای ۳۲ بیتی

عملوندهای ۳۲ بیتی (LONG-INTEGERS, DOUBLE-WORDS) از دو کلمه ۱۶ بیتی در حافظه که کنار هم قرار دارند، شکل می‌گیرد. کلمه کم ارزش‌تر در این متغیرها همواره در آدرس پایین‌تر قرار می‌گیرد، حتی اگر این عملوندها در حافظه پشت‌به‌پشت قرار گیرند یعنی کلمه با ارزش‌تر بایستی ابتدا به داخل پشت‌به‌پشت PUSH شود. آدرس کلمه کم ارزش‌تر آن است.

از انواع داده‌های ۳۲ بیتی به عنوان عملوند در عملیات توام با شیفت بطور سخت‌افزاری استفاده می‌شود. همانند تقسیم اعداد ۳۲ بیتی بر اعداد ۱۶ بیتی و ضرب اعداد ۱۶ بیتی در اعداد ۱۶ بیتی برای این عملیات عملوند ۳۲ بیتی بایستی در فایل ثبات داخلی قرار گیرد و آدرس اشاره کننده به این عملوند باید قابل تقسیم بر ۴ باشد.

۲-۴-۴-۳. لینک کردن زیربرنامه‌ها

پارامترها از طریق پشت‌به‌پشت به زیر برنامه انتقال می‌یابند پارامترها هنگامی که در اسکن برنامه به آن برخورد می‌شود به ترتیب برخوردها از اول برنامه تا انتهای برنامه در داخل پشت‌به‌پشت PUSH می‌شوند.

پارامترهای ۳۲ بیتی به صورتی دو مقدار ۱۶ بیتی در داخل پشت‌به‌پشت PUSH می‌شوند بطوریکه ابتدا با ارزش‌ترین قسمت پارامتر به داخل پشت‌به‌پشت PUSH می‌شود. در زیر یک برنامه با پارامترهایش که در استاندارد PLM-96 تعریف شده، ملاحظه می‌شود:

```
example procedure:PROCEDURE (param1,param2,param3),
    BYTE param1      DECLARE
    DWORD param2
    WORD param3
```

زمانی که این زیر برنامه اجرا شود. پشت‌به‌پشت در برگرفته پارامترها به ترتیب زیر خواهد بود.

→ Stack Image

	????? ; param1	
	high word of param2	
	low word of param2	
	param3	
	return address	

← Stack_Pointer

چنانچه زیر برنامه‌ای بخواهد مقداری را به برنامه اصلی برگرداند، این مقدار در PLMREG قرار گیرد. در این وضعیت حافظه PLMREG می‌تواند به عنوان متغیرهای ۸، ۱۶ یا ۳۲ بیتی مورد استفاده قرار گیرد.

قرارداد استاندارد فراخوانی (calling) که از PLM-96 گرفته شده چندین ویژگی کلیدی دارد:

- زیر برنامه می‌تواند از هشت بایت فایل ثبات حافظه، که از PLMREG آغاز می‌شود به عنوان ذخیره موقت در داخل خود زیر برنامه استفاده کند.
- برنامه‌ای که زیر برنامه را فراخوانی می‌کند بایستی فرض کند که زیر برنامه هشت بایت فایل ثبات حافظه را که از PLMREG آغاز می‌شود، تغییر می‌دهد.
- برنامه‌ای که زیر برنامه را فراخوانی می‌کند بایستی فرض کند که زیر برنامه پرچم‌های وضعیت کلمه وضعیت برنامه (PSW) را (Z,N,V,VT,C,ST) را تغییر می‌دهد. به این دلیل که زیر برنامه PSW را در ابتدای ذخیره نمی‌کند و در انتها نیز مقدار آن را بر نمی‌گرداند.
- نتایج عملکرد زیر برنامه همواره به متغیر PLMREG باز می‌گردد.

استاندارد PLM-96 اجازه تعریف زیر برنامه‌های وقفه را می‌دهد که وقتی یک تقاضای وقفه از پیش تعریف شده رخ دهد، این زیر برنامه اجرا می‌شود. اجرای زیر برنامه‌های وقفه‌ای با قوانین اجرای زیر برنامه‌های نرمال، سازگاری ندارد. پارامترها نمی‌توانند به این زیر برنامه‌ها منتقل شوند و نمی‌توانند نتیجه بازگشتی داشته باشند. از آنجایی که زیر برنامه وقفه در هر زمانی باید قابل اجرا باشد، این زیر برنامه بایستی ثبات PSW و مقادیر موجود در حافظه PLMREG را ذخیره کند و بتواند آنها را باز گرداند.

۵-۶-۲. ویژگی‌های حفاظت نرم افزاری

خطاهای سخت‌افزاری و نرم‌افزاری می‌توانند اجرای برنامه را در 8XC196KC/KD با اختلال مواجه کنند. برای رهایی از این خطاها به بررسی چندین ویژگی می‌پردازیم. نخست آنکه، با استفاده از دستورالعمل RST می‌توان بصورت نرم‌افزاری سیستم را ریست کرد. ولی اگر اجرای برنامه با اختلال مواجه شود، استفاده از این دستور عملی نیست.

نکته جالب دیگری که می‌توان به عنوان یک ویژگی برای رهایی از خطاها از آن استفاده کرد، رمز دستورالعمل RST یا 0FFH است. فرض کنید که شمارنده برنامه PC از محدوده مجاز خارج شده و به آدرسی اشاره می‌کند که در فضای حافظه برنامه نیست در این صورت خروجی حافظه برنامه شناور بوده و عدد 0FFH واکشی می‌شود که همان رمز دستورالعمل RST می‌باشد. یعنی هر وقت شمارنده برنامه از محدوده مجاز خارج شود، میکرو ریست خواهد شد و این بسیار خوب است. سخت‌افزار شمارنده نگهبان **WDT** نیز می‌تواند هنگام وقوع خطا سیستم را ریست کند.

توصیه می‌شود قسمتهایی از حافظه برنامه را که مورد استفاده قرار نمی‌گیرد با مقدار 0FFH پر شود. علاوه بر این مقدار می‌توان در این قسمت‌ها از دستورات NOP یا پرش به

زیربرنامه‌های خطا نیز استفاده کرد. این نکته مخصوصاً در قسمتهایی از حافظه که در مجاورت جداول قرار دارند اهمیت دارد. چون احتمال بروز خطا به هنگام دسترسی به جدولها زیاد است. از این رو برای محافظت جدولها را با ۷ دستور NOP احاطه میکنند (چون طولانی‌ترین دستورالعمل در 8XC196KC/KD هفت بایتی است) و سپس از یک دستور RST یا پرش به محل خطا استفاده میشود. از آنجائیکه دستور RST یک بایتی است، چنانچه از این دستور برای محافظت از جدول استفاده کنیم نیازی به دستورات NOP نیست.

به هنگام استفاده از WDT در محافظت از برنامه توصیه میشود که عملیات ریست کردن WDT فقط در یک محل از برنامه صورت گیرد. در این صورت احتمال ریست شدن ناخواسته در WDT کم خواهد شد. بخشی از برنامه که وظیفه ریست کردن WDT را دارد باید از چگونگی اجرای برنامه در سایر بخشها مطلع باشد. این عمل را میتوان با چک کردن مقدار متغیرها انجام داد. ریست کردن نرم‌افزاری WDT را میتوان بسادگی توسط یک شمارنده نرم‌افزاری انجام داد.