

فصل چهارم
بخشهاي حافظه

این فصل فضای حافظه قابل آدرس دهی 8XC196KD, 8XC196KC را مورد بررسی قرار می دهد. هر دو نوع میکروکنترلر دارای ۶۴K کیلوبایت فضای حافظه قابل آدرس دهی هستند. این فضای حافظه می تواند برای برنامه (CODE) یا داده (Data) استفاده شود. این جمله در ادامه جمله قبلی نوشته شود. هر مکان حافظه^۱ در برگزیده یک بایت است. شکل ۱-۴ بخشهای اصلی حافظه را نشان می دهد. آدرسهای KC و KD با هم متفاوت هستند چون فضای RAM و OPTROM^۲ میکروکنترلر 8XC196KD دو برابر میکروکنترلر 8XC196KC است.

۴-۱. بخشهای حافظه خارجی

بالای نقشه حافظه و بخشی که درست بالای فایل ثبات قرار دارد و همیشه مختص حافظه خارجی یا I/O است. (شکل ۱-۴ یا جدول ۱-۴ را ملاحظه کنید). 8XC196KC/KD توانایی Interface شدن را با انواع حافظه های خارجی دارد. برای مطالعه جزئیات بیشتر به فصل ۱۳ ارتباط با حافظه خارجی مراجعه کنید.

۴-۲. درگاه ۳ و درگاه ۴

بسته به اینکه بعد از reset شدن، سیگنال EA# چه مقداری داشته باشد. درگاههای ۳ و ۴ می توانند به صورت I/O، گذرگاه (آدرس / داده) یا ترکیبی از این دو عمل کنند. برای اطلاعات بیشتر به فصل ششم درگاه های ورودی - خروجی مراجعه کنید.

جدول ۱-۴. آدرسهای فضای حافظه 8XC196KC/KD

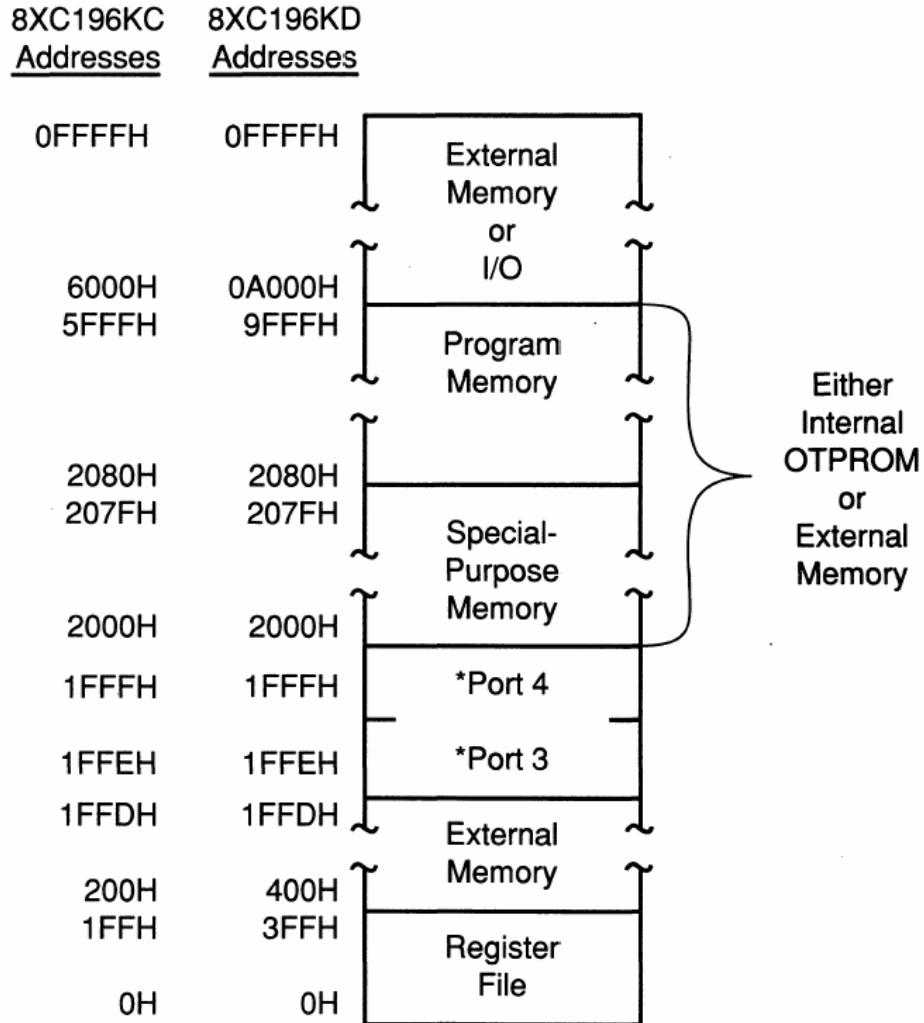
توضیحات	محدوده آدرس 8XC196KC		محدوده آدرس 8XC196KD	
	مبنای شانزده	مبنای دهی	مبنای شانزده	مبنای دهی
I/O یا حافظه خارجی	6000H-0FFFFH	24576-65635	0A000H-0FFFFH	40960-65535
حافظه CODE (برنامه) *	2080H-5FFFFH	8320-24575	2080H-9FFFFH	8320-40956
حافظه تك منظوره ^۳	2000H-207FH	8192-8319	2000H-207FH	8192-8319
درگاه ۳ و درگاه ۴	1FFEh-1FFFFH	8190-8191	1FFEh-1FFFFH	8190-8191
حافظه خارجی	200H-1FFDH	512-8189	400H-1FFDH	1024-8189
فایل ثباتها (شامل SFR ها)	0H-1FFH	0-511	0H-3FFH	0-1023

^۱.Memory Location

^۲.One Time Programmable Read Only Memory

^۳.Special Purpose Memory

* حافظه CODE می تواند در OPTROM داخلی یا فضای حافظه خارجی قرار داشته باشد.



شکل ۴-۱. نقشه فضای حافظه

جدول ۴-۱ آدرسهای حافظه 8XC196KC , 8XC196KD را مقایسه و آدرسهای شروع و پایان را در مبنای دهدهی و مبنای شانزده مشخص می کند.

۴-۳-۱. انتخاب نگاهت حافظه داخلی یا خارجی

برای دست یابی به محدوده آدرس حافظه برنامه یا حافظه تک منظوره می توان از OPTROM داخلی یا حافظه خارجی استفاده کرد ولی نمی توان هم زمان از هر دوی آنها استفاده کرد. مقدار پایه خارجی EA# طی لبه بالا رونده سیگنال RESET# (منظور عمل ریست شدن) مشخص می کند که دستیابی به صورت داخلی است یا خارجی. این مقدار در میکرو latch می شود و تغییر نمی کند تا زمانی که میکرو باردیگر reset شود. اگر EA# در سطح منطقی پایین باشد OPTROM داخلی غیرقابل دسترسی است و تمامی آدرسهای مربوط به این محدوده به حافظه خارجی نسبت داده می شود. ولی اگر EA# در سطح منطقی بالا باشد محدوده این آدرسها به OPTROM داخلی نسبت داده می شود.

۴-۳-۲. حافظه برنامه^۱

حافظه برنامه برای 8XC196KC در آدرس 5FFFH - 2080H و برای 8XC196KD در آدرس 9FFFH - 2080H قرار دارد. زمانی که میکرو ریست می شود، CPU دستوالعمل^۲ را از مکان حافظه 2080H واکشی^۳ کرده و سپس اجرا می کند. (به فصل ۱۱ حداقل سخت افزار مورد نیاز مراجعه کنید).



با اینکه مقدار از پیش تعیین شده مکانهای حافظه ROM/OPTROM ، 0FFH است توصیه می شود که در مکانهایی از حافظه که مورد استفاده قرار نگرفته است 0FFH بنویسید.

۴-۳-۳. حافظه مخصوص^۴

حافظه مخصوص در فضای آدرس 207FH - 2000H قرار دارد. (شکل ۲-۴) این حافظه در برگزیده مکانهای رزور شده، بردارهای PTS و بردارهای وقفه های استاندارد و یک کلید امنیتی^۵ و بایت تعیین وضعیت سیستم^۶ CCB می باشد. جدول ۴-۲ آدرسهای حافظه تک منظوره و آدرسهای شروع و پایان را در مبنای دهدهی و مبنای شانزده مشخص می کند.

^۱.Code Memory

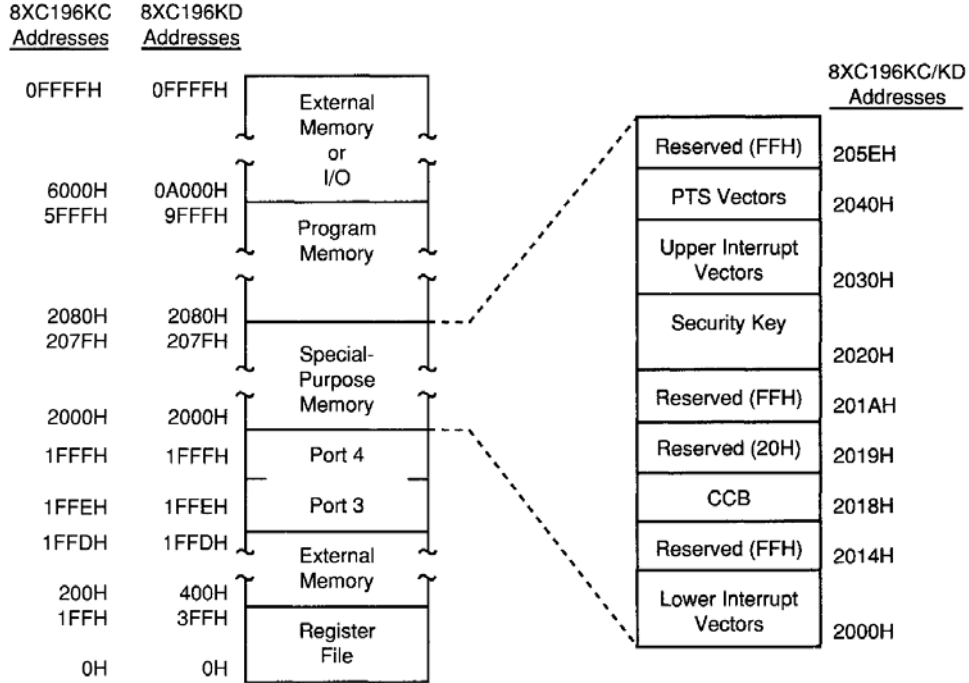
^۲.Instruction

^۳.Fetch

^۴.Special Purpose Memory

^۵ . کلید امنیتی قسمتی از حافظه تک منظوره است (2029H - 2020H) که شامل ۱۶ بایت می شود. برای توضیحات بیشتر به قسمت ۴-۳-۳. همین فصل مراجعه کنید.

^۶.Chip Configuration Byte CCB



شکل ۲-۴. نگاشتی حافظه مخصوص 8XC196KC/KD

۴-۳-۳-۱. مکانهای حافظه رزرو شده

مکانهای حافظه رزور شده متعددی برای آزمایش یا استفاده در محصولات بعدی در نظر گرفته شده است. کاربرد نبایستی در این مکانهای رزور شده چیزی بنویسید و یا بخواند، عملکرد این مکانهای حافظه ممکن است در نسخه های آینده تغییر کند. بنابراین نرم افزارهایی که از این مکانهای حافظه رزور شده استفاده می کنند ممکن است درست عمل نکنند. زمانی که OPTROM، برنامه ریزی می شود، همیشه در آدرس 2019H که رزور شده است مقدار 20H و در بقیه آدرسهای رزور شده 0FFH بنویسید.

۴-۳-۳-۲. بردارهای وقفه^۱

بردارهای وقفه بالایی و پایینی در برگزیده آدرسهای زیر برنامه وقفه هستند. بردارهای PTS شامل آدرس بلوکهای کنترلی PTS می باشد. برای اطلاعات بیشتر به فصل ۵ **وقفه ها** مراجعه کنید.

جدول ۲-۴. آدرسهای حافظه تک منظوره 8XC196KC/KD

محدوده آدرس 8XC196KC/KD	توضیحات
-------------------------	---------

^۱.Interrupt Vector

مبنای دهدی	مبنای شانزده	
8286-8319	205EH-207FH	رزرو شده (بایستی در آن OFFH نوشته شود)
8256-8285	2040H-205DH	بردارهای PTS
8240-8255	2030H-203FH	بردارهای وقفه بالای حافظه تك منظوره
8224-8239	2020H-202FH	کلید امنیتی
8218-8223	201AH-201FH	رزرو شده (بایستی در آن OFFH نوشته شود)
8217	2019H	رزرو شده (بایستی در آن 20H نوشته شود)
8216	2018	CCB
8212-8215	2014H-2017H	رزرو شده (بایستی در آن OFFH نوشته شود)
8192-8211	2000H-2013H	بردارهای وقفه پایین حافظه تك منظوره

۴-۳-۳-۳ . کلید امنیتی^۱

کلید امنیتی از خواندن و نوشتن غیرمجاز OPTROM در 8XC196KC/KD جلوگیری می کند. اگر بیت های قفل امنیتی^۲ در CCR^۳ پاک شوند هر کدام از شانزده بایت کد امنیتی که قابل برنامه ریزی است با بایت متناظر خود در حافظه خارجی مقایسه می شود. اگر داده خارجی با کلید منطبق نباشد، میکرو وارد حالتی که برنامه ریزی شده است نمی شود.

۴-۳-۳-۴ . بایت تعیین وضعیت سیستم CCB

CCB اولین بایتی است که بعد از ریست شدن میکرو از حافظه واکنشی می شود. به دنبال ریست شدن میکرو، بایت CCB در یک ثبات داخلی به نام CCR بار می شود. حفاظت داخلی حافظه^۴، حالت های انتظار داخلی^۵، سیگنال کنترلی گذرگاه، انتخاب پهنای گذرگاه^۶ و حالت عملکرد Powerdown را کنترل می کند. (ضمیمه C ثبات های 8XC196KC/KD را ملاحظه کنید.)

۴-۴ . فایل ثبات

فایل ثبات به دو قسمت بالایی فایل ثبات و پایینی فایل ثبات تقسیم می شود (شکل ۴-۳) فایل ثبات بالایی دربرگیرنده ثبات های

1. Security Key
2. Security Lock Bit
3. Chip Configuration Register
4. Internal Memory Protection
5. Internal Wait State
6. BUSWIDTH

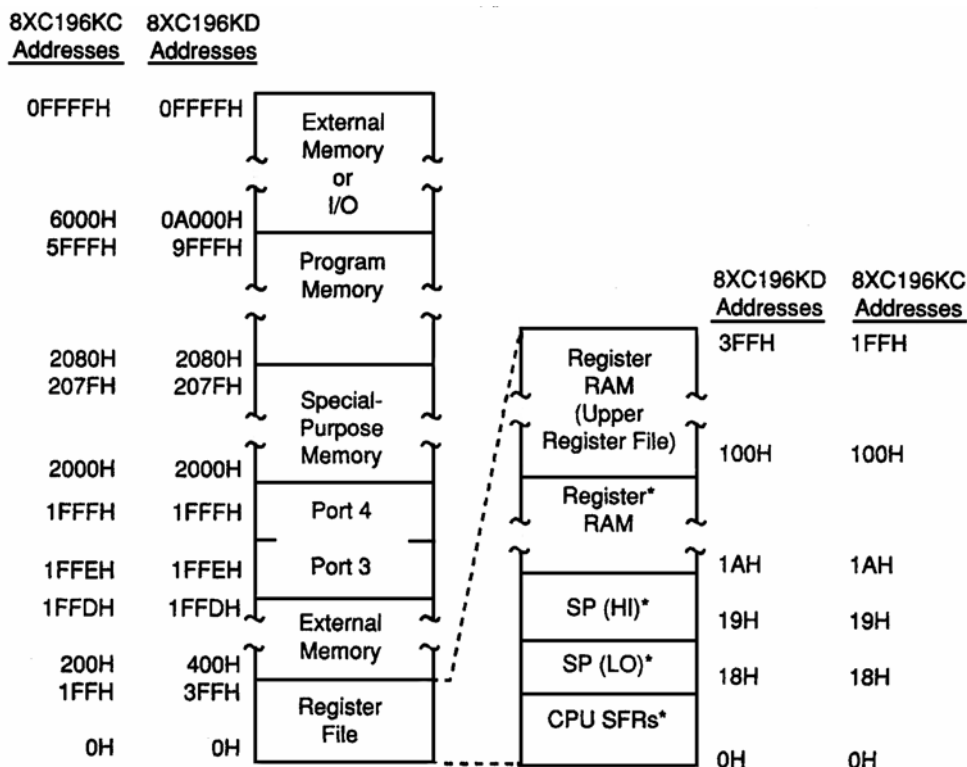
RAM همه منظوره^۱ است. فایل ثبات پایینی دربرگیرنده ثباتهای همه منظوره و اشاره گر پشته SP^۲ و SFR ها است. جدول ۳-۴ آدرسهای حافظه فایل ثبات 8XC196KC و 8XC196KD را مقایسه کرده و آدرسهای شروع و پایان را در مبنای دهدهی و مبنای شانزده مشخص می کند.



فایل ثبات نمی تواند در برگیرنده برنامه (CODE) باشد. تلاش برای اجرای دستورالعمل از مکان حافظه فایل ثبات 0000H-01FFH در 8XC196KC یا 03FFH - 0000H در 8XC196KD موجب می شود که کنترل کننده حافظه به طور خودکار (اتوماتیک) دستورالعملها را از حافظه خارجی واکشی کند. این بخش از حافظه خارجی رزور شده است تا توسط ابزار تولیدی Intel مورد استفاده قرار گیرد.

^۱.General Purpose Register RAM

^۲.Stack Pointer



شکل ۳-۴. نگاشت حافظه فایل ثابت

جدول ۳-۴. آدرسهای حافظه فایل ثابت 8XC196KC/KD

توضیحات	محدوده آدرس 8XC196KC		محدوده آدرس 8XC196KD	
	مبنای شانزده	مبنای دهی	مبنای شانزده	مبنای دهی
فایل ثابت بالایی*	100H-1FFH	256-511	100H-3FFH	256-1023
ثباتهای RAM همه منظور **	1AH-0FFH	26-225	1AH-0FFH	26-255
اشاره گر پشته**	18H-19H	24-25	18H-19H	24-25
SFRها **	0H-17H	0-23	0H-17H	0-23

* : این مورد در برگزیده ثباتهای RAM همه منظور (دست یابی به آن از طریق آدرس دهی غیرمستقیم یا آدرس دهی اندیس دار ممکن است، تا زمانیکه از *Vertical Windowing* (دستری به پنجره به طور عمودی) استفاده نشود است.

** : این سه مورد در فایل ثابت پایینی (دست یابی هم به صورت آدرس دهی مستقیم ثبات و هم به صورت آدرس دهی غیرمستقیم یا آدرس دهی اندیس دار) قرار گرفته است.

۱-۴-۴. ثبات RAM همه منظوره

محل (1AH - 0FFH) در برگرفته ثبات RAM همه منظوره است، محل اشاره گر پشته 18H و 19H است و در صورتی می‌تواند به عنوان ثبات RAM همه منظوره استفاده شود، که عملیات پشته در حال انجام نباشد. RALU برای دسترسی به این قسمت از حافظه می‌تواند از دستیابی مستقیم با استفاده از آدرس دهی مستقیم ثباتی استفاده کند. فایل ثبات بالایی نیز در برگرفته ثبات RAM همه منظوره است (100H-1FFH در 8XC196KC و 3FFH - 100H در 8XC196KD) RALU به طور معمول با استفاده از آدرس دهی اندیس دار و غیرمستقیم به حافظه دسترسی پیدا می‌کند.

Vertical windowing تکنیکی است که RALU را قادر به استفاده از آدرس دهی مستقیم ثباتی در دست یابی RAM به فایل ثبات بالایی می‌کند. فصل ۳ انواع داده‌ها و آدرس‌ها را برای تفاوت‌های بین آدرس دهی مستقیم ثباتی و آدرس دهی اندیس دار ملاحظه کنید.

۲-۴-۴. اشاره گر پشته

محل حافظه 18H و 19H در برگرفته اشاره گر پشته (SP) است. و اشاره گر پشته در برگرفته آدرس پشته است. SP بایستی به آدرس یک کلمه (آدرس زوج) اشاره کند که آدرس آن دوبایت بیشتر از آدرس آغازین مورد نظر باشد. قبل از اینکه CPU یک فراخوانی زیر برنامه^۱ یا زیر برنامه وقفه را اجرا کند، از مقدار SP دو واحد کم می‌کند و سپس آدرس دستورالعمل بعدی را از PC (شمارنده برنامه) به پشته قرار منتقل می‌کند. و سپس آدرس زیر برنامه یا آدرس زیر برنامه وقفه را در PC بار می‌کند. بعد از کامل شدن اجرای زیر برنامه یا زیر برنامه وقفه، دستورالعمل بازگشت (RET) را اجرا می‌کند و مقداری را که در بالای پشته است (آدرس برگشت) را در PC بار می‌کند.

زیر برنامه‌ها ممکن است تودرتو باشند. این به این معنی است که هر زیر برنامه می‌تواند زیر برنامه دیگری را فراخوانی کند. تا زمانی که این فراخوانیهای تودرتو ادامه داشته باشد. CPU محتوای PC را در پشته PUSH می‌کند. پشته به صورت پایین رونده^۲ (کاهشی) است. تنها محدودیت برای عمق تودرتویی زیر برنامه‌ها مقدار حافظه موجود است. زمانی که CPU از هر زیر روال تودرتو بر گردد. آدرس بالای پشته را POP می‌کند و آدرس برگشت بعدی در بالای پشته جایگزین آن می‌شود. زمانی که عملیات پشته در حال اجرا شدن نباشد مکان SP می‌تواند به عنوان ثبات RAM همه منظور مورد استفاده قرار گیرد.

^۱. Subroutine

^۲. منظور این است که با اضافه شدن ورودی به پشته آدرسهای حافظه منظور شده برای پشته کاهش می‌یابد.

۱-۲-۴. مقدار دهی اولیه اشاره گر پشته

برنامه کاربر بایستی آدرسی را که به صورت دو بایت است، در داخل اشاره گر پشته بار کند. آدرس را باید طوری انتخاب کرد که آدرس آن دوبایت بیشتر از آدرس آغازین مورد نظر باشد به دلیل اینکه اشاره گر پشته قبل از PUSH آدرس بازگشتی در پشته به طور خودکار کاهش می یابد. به یاد داشته باشید که پشته به صورت پایین رونده عمل می کند، بنابراین فضای کافی برای بیشترین ورودی به پشته را لحاظ کنید. پشته می تواند هم در فایل ثبات داخلی و هم در RAM خارجی قرار بگیرد.

مثالی که در ادامه آمده است فایل ثبات بالایی 8XC196KD را به عنوان حافظه پشته انتخاب می کند.

LD SP,#400H ;Load Stack Pointer

۳-۴-۴. ثباتهای با وظیفه خاص SFR

CPU، از طریق سه پنجره با دسترسی افقی، می تواند به SFR های که در محل های 00H-17H قرار دارد، دسترسی داشته باشد. (سه پنجره با دسترسی افقی به شماره های ۱۵ و ۱۰ تعریف شده اند) (بخش ۴-۵-۴-۵ همین فصل را ملاحظه کنید). RALU از طریق این SFR ها تمامی واحدهای جانبی به جزء درگاه ۳ و ۴ را کنترل می کند. ضمیمه C "ثباتهای 8XC196KC/KD" هر یک از این ثباتها را بررسی کرده است.

زمانی که یک SFR، به عنوان یک ثبات اندیس، در آدرس دهی اندیس دار یا آدرس دهی غیر مستقیم استفاده می شود در نظر داشته باشید که محتوای SFR ها تغییر می کند و غیر قابل پیش بینی هستند. رخ دادهای خارجی می توانند محتوای SFR ها را تغییر دهند. محتوای بعضی از SFR ها پس از خوانده شدن پاک می شود. عملکرد بسیاری از SFR ها بسته به اینکه آنها در حال خوانده شدن یا نوشته شدن هستند تغییر می کند به همین دلیل هیچ گاه از یک SFR به عنوان عملوند در (خواندن - تغییر دادن - نوشتن) یک دستور العمل استفاده نکنید برای مثال:

XORB AD_RESULT یک کد برنامه نویسی اشتباه است چون AD_RESULT یک SFR است و ممکن است محتوای آن تغییر کند. از SFR های رزرو شده استفاده نکنید؛ یا مقدار صفر در آنها بنویسید یا اینکه آنها را با مقدار از پیش تعیین شده ای که دارند رها کنید. اگر عمل خواندن را بر روی آنها انجام دهید، بیتها و SFR های رزور شده مقدار غیرمشخصی را برمی گردانند.

۵-۴. پنجره با دسترسی افقی^۱

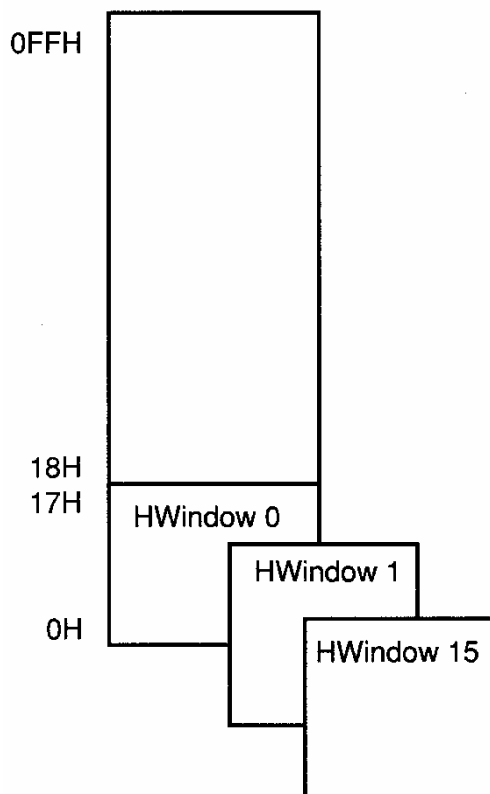
از سه پنجره با دسترسی افقی استفاده می کند (HWindow 0,1,15)، ۲۴ محل حافظه پایینی در فایل ثبات پایینی مختص به این

^۱. HORIZONTAL WINDOWING

سه پنجره است. (شکل ۴-۴ را ملاحظه کنید). هر پنجره با دسترسی افقی (HWindow) یک عمل خواندن و نوشتن برای دست یابی به ترکیب واحدی از SFRها را فراهم می‌کند.

- بعضی از ثباتها به صورت تک بیتی و بعضی دیگر به صورت یک کلمه^۱ (دوبایت) در دسترس هستند.
- بعضی از ثباتها مانند (WSR, 14H) در هر سه پنجره (Hwindow) در دسترس هستند.
- ثباتهای دیگر باید در یک HWindow نوشته شوند و در HWindow دیگری خوانده شوند.

ضمیمه C "ثباتهای 8XC196KC/KD" جزئیات هر ثبات را بررسی می‌کند.



شکل ۴-۴. پنجره‌های با دسترسی افقی

۴-۵-۱. انتخاب یک پنجره افقی یا HWindow از طریق ثبات WSR^۲ می‌توان به HWindow و VWindow^۱ دست یافت (بخش ۴-۶ همین فصل را ملاحظه کنید).

^۱.WORD

^۲.Window Select Register **WSR**

برای انتخاب يك HWindow، شماره پنجره مورد نظر را در بیت‌های WSR.3-WSR.0 بنویسید و بیت‌های WSR.6-WSR.4 را پاک کنید. تنها Hwindow0، Hwindow1، و Hwindow15 موجود هستند و بقیه HWindowها رزرو شده هستند جدول ۴-۴ مقدار مورد نظر WSR را برای انتخاب هر HWindow نشان می‌دهد. ضمیمه C "ثبات‌های 8XC196KC/KD" را برای توصیف کامل WSR مشاهده کنید. اگر به ثبات WSR، Hwindow رزرو شده را نسبت دهید نتیجه غیر قابل پیش‌بینی در پی خواهد داشت تنها راه برای پاک کردن ثبات WSR، ریست کردن میکرو یا استفاده از دستورالعمل POPA است.

جدول ۴-۴. انتخاب‌های HWindow

HWindow	محتویات WSR
0	X000 0000B=00H
1	X000 0001B=01H
15	X000 1111B=0FH

۴-۵-۲. پنجره با دسترسی افقی صفریا Hwindow0 مقدار پیش فرضی که برای Hwindow در نظر گرفته شده است Hwindow0 است. این پنجره امکان خواندن از ۱۹ ثبات و نوشتن در ۲۱ ثبات را فراهم می‌کند. (شکل ۴-۵ را مشاهده کنید) بعضی از ثبات‌ها (برای مثال INT_MASK1 در Hwindow0) هم خوانده می‌شود و هم نوشته می‌شود ولی بقیه ثبات‌های این پنجره را تنها می‌توان خواند یا تنها می‌توان نوشت برای نوشتن در آن دسته از ثبات‌ها که فقط قابل خواندن هستند بایستی پنجره Hwindow15 را انتخاب کرد.

۴-۵-۳. HWindow1

Hwindow1 دسترسی به ۱۲ ثبات را برای خواندن و نوشتن فراهم می‌کند. (شکل ۴-۵ را مشاهده کنید) ویژگی این پنجره تفاوت‌های بین 8XC196KC و 8XC196KD و نسخه‌های اولیه MCS_96 را نشان می‌دهد (در این پنجره ویژگی‌های جدیدی که به 8XC196KC/KD اضافه شده است قرار دارد) بعضی از ثبات‌های این پنجره در Hwindow0 و Hwindow15 نیز قابل دستیابی هستند.

^۱. Vertical Window

	HWINDOW 0 (Read)	HWINDOW 0 (Write)		HWINDOW 1 (Read/Write)
17H	IOS2	PWM0_CONTROL	17H	PWM2_CONTROL
16H	IOS1	IOC1	16H	PWM1_CONTROL
15H	IOS0	IOC0	15H	Reserved
14H	WSR	WSR	14H	WSR
13H	INT_MASK1	INT_MASK1	13H	INT_MASK1
12H	INT_PEND1	INT_PEND1	12H	INT_PEND1
11H	SP_STAT	SP_CON	11H	Reserved
10H	IOPORT2	IOPORT2	10H	Reserved
0FH	IOPORT1	IOPORT1	0FH	Reserved
0EH	IOPORT0	BAUD_RATE	0EH	Reserved
0DH	TIMER2 (HI)	TIMER2 (HI)	0DH	Reserved
0CH	TIMER2 (LO)	TIMER2 (LO)	0CH	IOC3
0BH	TIMER1 (HI)	IOC2	0BH	Reserved
0AH	TIMER1 (LO)	WATCHDOG	0AH	Reserved
09H	INT_PEND	INT_PEND	09H	INT_PEND
08H	INT_MASK	INT_MASK	08H	INT_MASK
07H	SBUF (RX)	SBUF (TX)	07H	PTSSRV (HI)
06H	HSI_STATUS	HSO_COMMAND	06H	PTSSRV (LO)
05H	HSI_TIME (HI)	HSO_TIME (HI)	05H	PTSSSEL (HI)
04H	HSI_TIME (LO)	HSO_TIME (LO)	04H	PTSSSEL (LO)
03H	AD_RESULT (HI)	HSI_MODE	03H	AD_TIME
02H	AD_RESULT (LO)	AD_COMMAND	02H	Reserved
01H	ZERO_REG (HI)	ZERO_REG (HI)	01H	ZERO_REG (HI)
00H	ZERO_REG (LO)	ZERO_REG (LO)	00H	ZERO_REG (LO)

شکل ۴-۵. HWindow0 و HWindow1

۴-۵-۴. HWindow15

به جزء بایتهای با آدرس 0CH-10H، HWindow15 امکان دستیابی به ثباتهایی را میدهد که HWindow0 نیز به آنها دسترسی دارد. بایتهای 0CH-0DH در پنجره HWINDOW0 به شماره ۱۶ بیتی تایمر اختصاص دارد اما همین بایتهای در پنجره HWindow15 به ثبات ۱۶ بیتی $T2CAPTURE$ اختصاص پیدا میکنند. بایتهای 10H-0EH برای ثباتهای $IOPORT0$ ، $IOPORT1$ و $IOPORT2$ در HWindow0، در نظر گرفته شده است. ولی این بایتهای در HWindow15 رزرو شده اند. آن دسته از ثباتهایی که در HWindow0 فقط به صورت خواندنی هستند در HWindow15 فقط نوشتنی هستند و بالعکس.

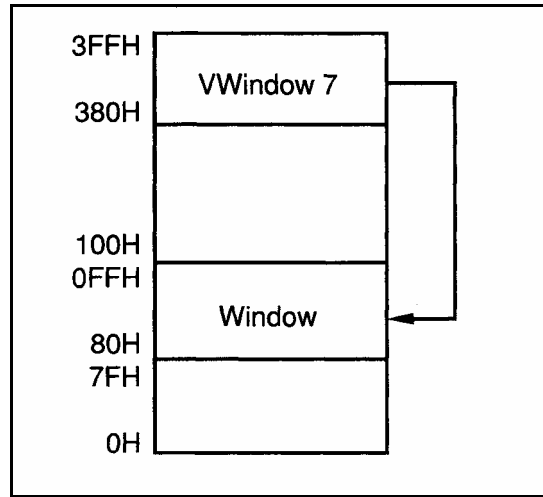
	HWINDOW 15 (Read)	HWINDOW 15 (Write)
17H	PWM0_CONTROL	IOS2
16H	IOC1	IOS1
15H	IOC0	IOS0
14H	WSR	WSR
13H	INT_MASK1	INT_MASK1
12H	INT_PEND1	INT_PEND1
11H	SP_CON	SP_STAT
10H	Reserved	Reserved
0FH	Reserved	Reserved
0EH	Reserved	Reserved
0DH	T2CAPTURE (HI)	T2CAPTURE (HI)
0CH	T2CAPTURE (LO)	T2CAPTURE (LO)
0BH	IOC2	TIMER1 (HI)
0AH	WATCHDOG	TIMER1 (LO)
09H	INT_PEND	INT_PEND
08H	INT_MASK	INT_MASK
07H	SBUF (TX)	SBUF (RX)
06H	HSO_COMMAND	HSI_STATUS
05H	HSO_TIME (HI)	HSI_TIME (HI)
04H	HSO_TIME (LO)	HSI_TIME (LO)
03H	HSI_MODE	AD_RESULT (HI)
02H	AD_COMMAND	AD_RESULT (LO)
01H	ZERO_REG (HI)	ZERO_REG (HI)
00H	ZERO_REG (LO)	ZERO_REG (LO)

شکل ۴-۶. HWindow15

۴-۶. پنجره با دسترسی عمودی^۱

Vertical Windowing تکنیکی است که بخش بالایی فایل ثبات را در محلهای حافظه بالای فایل ثبات پایینی نگاشت می‌کند. دسترسی به این قسمت از حافظه می‌تواند بصورت ۳۲ و ۶۴ و ۱۲۸ بایتی صورت گیرد (که هر کدام از این بلوکها به VWindow معروف است) 8XC196KC ۱۶ عدد VWindow ۳۲ بایتی، ۸ عدد VWindow ۶۴ بایتی و ۴ VWindow ۱۲۸ بایتی دارد. از آن جایی که 8XC196KD نسبت به 8XC196KC دو برابر RAM دارد. بنابراین نسبت به 8XC196KC نیز دو برابر VWindow دارد. شکل ۷-۴ نمونه ای از Vwindow، ۱۲۸ بایتی را نمایش می‌دهد.

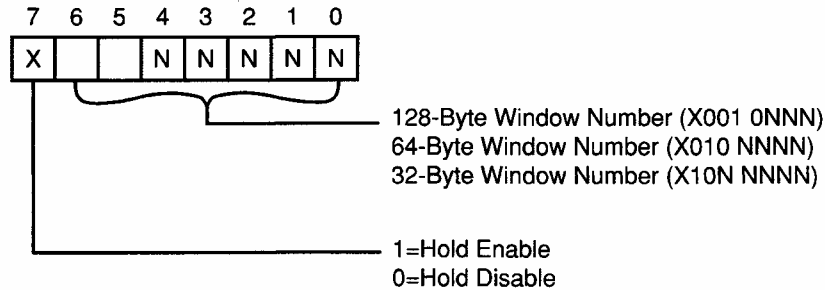
^۱. VERTICAL WINDOWING



شکل ۷-۴. Vertical Windowing

۴-۶-۱. انتخاب يك VWindow

از طریق ثبات WSR می توان HWindow یا VWindow دسترسی پیدا کرد. یک کردن بیت‌های WSR.4 و WSR.5 یا WSR.6 به ترتیب برای انتخاب VWindow ۱۲۸ و ۶۴ و ۳۲ بیتی را انتخاب می‌کند. (شکل ۸-۴ را ملاحظه کنید) برای انتخاب یک شماره VWindow مورد نظر را در بیت‌های پایینی WSR بنویسید. ضمیمه C (ثبات‌های 8XC196KC/KD) توضیح کامل از ثبات WSR با یک جدول از محتویات ثبات WSR برای انتخاب VWindow های مناسب را ارائه کرده است. به عنوان مثال برای انتخاب VWindow مشابه با شکل ۷-۴، مقدار 17H را در ثبات WSR بنویسید.



شکل ۸-۴. تنظیم بیت‌های ثبات انتخاب پنجره

۴-۶-۲. Vertical Windowing و روش‌های آدرس دهی

زمانی که Vertical Windowing فعال می‌شود:
 ▪ یک دستور العمل مستقیم ثباتی که از یک آدرس در فایل ثبات پایینی استفاده می‌کند در واقع به Vwindow در فایل ثبات بالایی دسترسی دارد.

- يك دستورالعمل غیرمستقیم یا اندیس دار كه از يك آدرس در فایل ثبات پایینی یا از VWindow استفاده می کند به مكان واقعي در حافظه دسترسی دارد.
 - اگر $WSR = X1000000$ باشد عملیات shif غیرمستقیم درست عمل نخواهد کرد.
- مثالی که در ادامه آمده است تفاوت بین آدرس دهی مستقیم ثبات و آدرس دهی اندیس دار را زمانی که از Vertical Windowing استفاده می شود را نشان می دهد.

PUAHA		;pushes the contents of WSR onto the stack
LDB	WSR,#17H	;select Vwindow7 ,a 128-byte block
		;The next instruction uses register-direct addressing
ADD	40H,80H	;mem_word(40H)←mem_word(40H) + mem_word(380H)
		;The next instructio uses indirect addressing
ADD	40H,80H[0]	;mem_word(40H)←mem_word(40H) + mem_word(80H +1)
ADD	40H,380H[0]	;mem_word(40H)←mem_word(40H) + mem_word(380H +1)
POPA		;reloads the previous contents into WSR

برنامه های زیر چگونگی تعیین بایت CCB را در برنامه ها به دو زبان اسمبلی و C نمایش می دهد:

```

#include (80c196kd.inc)
; Chip configuration :
; number of wait states = 3
; bus control mode = Standard bus control
; programming protection = none
; power-down mode = enabled
; buswidth control = 8-bit
; hold/holda protocol = disabled
CSEG AT 2018H
DCW 20EDH

CASES
INIT_HOLD_ENABLE :
    ANDB WSR, #7FH
    RET

CSEG AT 2080H
MAIN_HOLD_ENABLE :
    CALL INIT_HOLD_ENABLE
; The following will put the processor in power down mode
    IDLPD #2
    BR$
END
    
```

```
#pragma model(kc)
#include <80c196kd.h>

    /*Chip configuration:
    *number of wait states = 3
    *bus control mode    = Standard bus control
    *programming protection = none
    *power-down mode     = enabled
    *buswidth control    = 8-bit
    *hold/holda protocol = disabled*/

static const unsigned int ccr = 0x20ED;
#pragma locate (ccr = 0x2018)

void init_HoldEnable(void)
{
    wsr = wsr & 0x7f;
}
void main(void)
{
    init_HoldEnable();
    /* The following will put the processor in power down mode*/
    power_down();
    while(1);
}
```
