Memory Partitions 4

CHAPTER 4
MEMORY PARTITIONS

This chapter describes the addressable memory space within the 8XC196KC and 8XC196KD. Both devices have 64 Kbytes of addressable memory space, most of which is available for either program or data memory. Each memory location holds one byte.

Figure 4-1 shows the major memory partitions. The addresses differ because the 8XC196KD has twice as much RAM and One-Time-Programmable Read-Only Memory (OTPROM, a version of EPROM) space as the 8XC196KC.

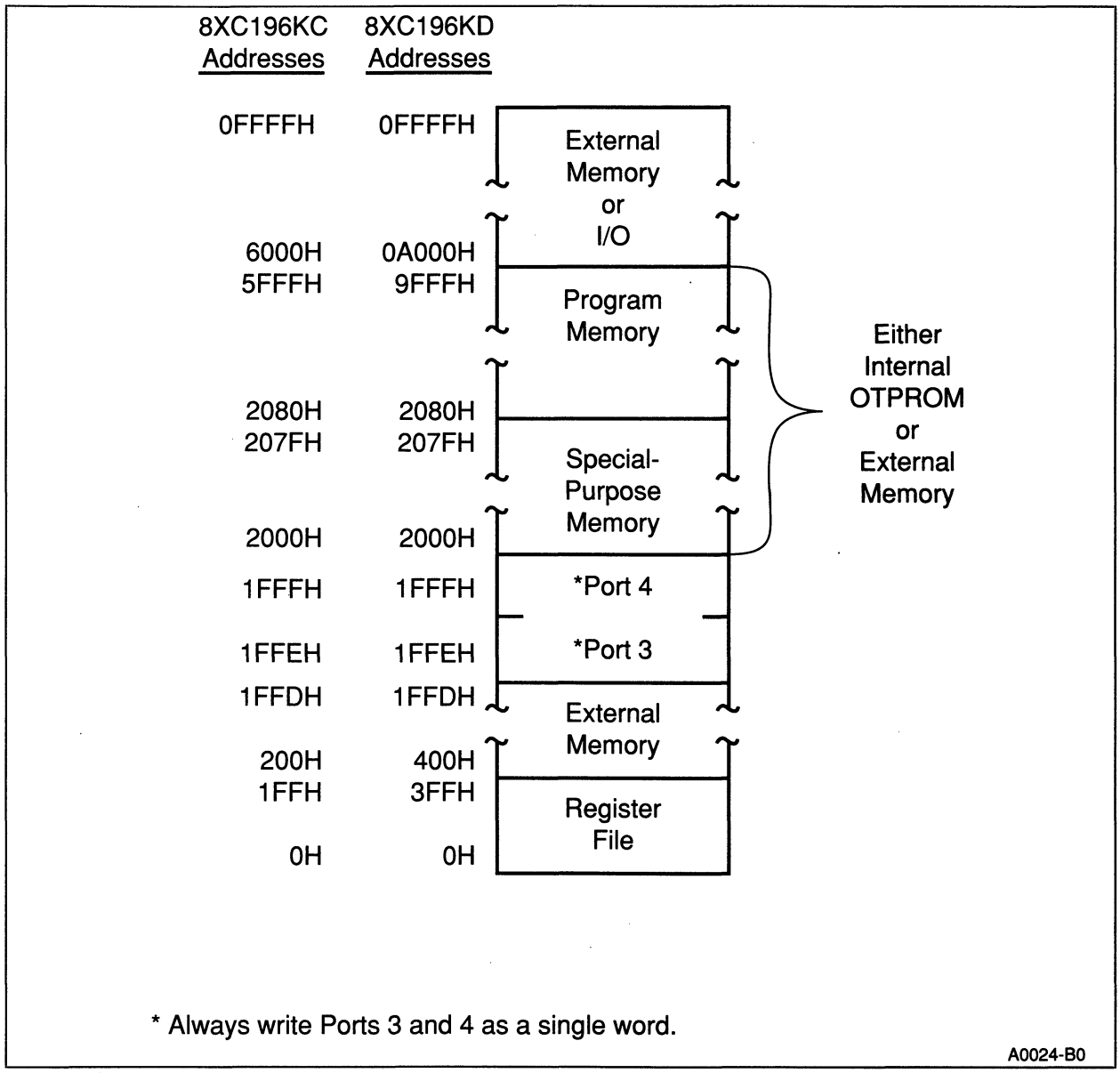

Figure 4-1. Top-Level Memory Map

Table 4-1 compares the 8XC196KC and 8XC196KD memory addresses and provides beginning and ending addresses in both hexadecimal and decimal. The sections that follow describe each memory partition.

Table 4-1. 8XC196KC/KD Top-Level Memory Addresses

Description	8XC196KC Address Range		8XC196KD Address Range	
	Hexadecimal	Decimal	Hexadecimal	Decimal
External Memory or I/O	6000H–0FFFFH	24576–65635	0A000H–0FFFFH	40960–65535
Program Memory (Note 1)	2080H–5FFFH	8320–24575	2080H–9FFFH	8320–40956
Special-Purpose Memory (Note 1)	2000H–207FH	8192–8319	2000H–207FH	8192–8319
Ports 3 and 4 (Note 2)	1FFEH–1FFFH	8190–8191	1FFEH–1FFFH	8190–8191
External Memory	200H–1FFDH	512–8189	400H–1FFDH	1024–8189
Register File (includes SFRs)	0H–1FFH	0–511	0H–3FFH	0–1023

NOTES:

1. Located in either internal OTPROM or external memory.
2. Ports 3 and 4 are word-addressable only.

4.1. EXTERNAL MEMORY PARTITIONS

The top of the memory map and the partition directly above the Register File are always assigned to external memory or I/O (see Figure 4-1 or Table 4-1). The 8XC196KC/KD can interface with a variety of external memory devices. See Chapter 13, "Interfacing with External Memory," for details.

4.2. PORTS 3 AND 4

Ports 3 and 4 are read and written as a word at memory location 1FFEH (Port 3 is the low byte; Port 4 is the high byte). These ports function as either I/O ports, the system address/data bus, or a combination of both, depending upon the value of the EA# signal at reset. See Chapter 6, "I/O Ports," for more information.

4.3. PROGRAM AND SPECIAL-PURPOSE MEMORY

Program memory and special-purpose memory occupy a 16-Kbyte memory partition in the 8XC196KC and a 32-Kbyte memory partition in the 8XC196KD (see Figure 4-1 or Table 4-1). These partitions can reside in either internal OTPROM or external memory. Chapter 14, "Programming the Nonvolatile Memory," provides information about programming the OTPROM.

4.3.1. Selecting Internal or External Memory Mapping

An access to the program memory or special-purpose memory address range is directed either to internal OTPROM or external memory, but not both. The value of the External Access (EA#) pin during the rising edge of the RESET# signal determines whether the access is internal or external. This value is latched into the device and cannot be changed unless the device is reset. If EA# is low, the internal OTPROM is inaccessible and all accesses to this address range are directed to external memory. If EA# is high, all accesses to this address range are directed to the internal OTPROM.

NOTE

The EA# input must be tied low for CPU-only devices to function properly.

4.3.2. Program Memory

Program memory is located at addresses 2080H–5FFFH in the 8XC196KC and addresses 2080H–9FFFH in the 8XC196KD. When the device is reset, the CPU fetches and then executes the instruction from location 2080H. (See Chapter 11, "Minimum Hardware Considerations," for more information about reset.)

NOTE

Since the default value in ROM/OTPROM locations is 0FFH, we recommend that you write 0FFH to unused program memory locations.

4.3.3. Special-Purpose Memory

Special-purpose memory is located at addresses 2000H–207FH (see Figure 4-2). It contains several reserved memory locations, vectors for both the Peripheral Transaction Server (PTS) and standard interrupts, a security key, and the Chip Configuration Byte (CCB). Table 4-2 lists the special-purpose memory addresses and provides beginning and ending addresses in both hexadecimal and decimal.

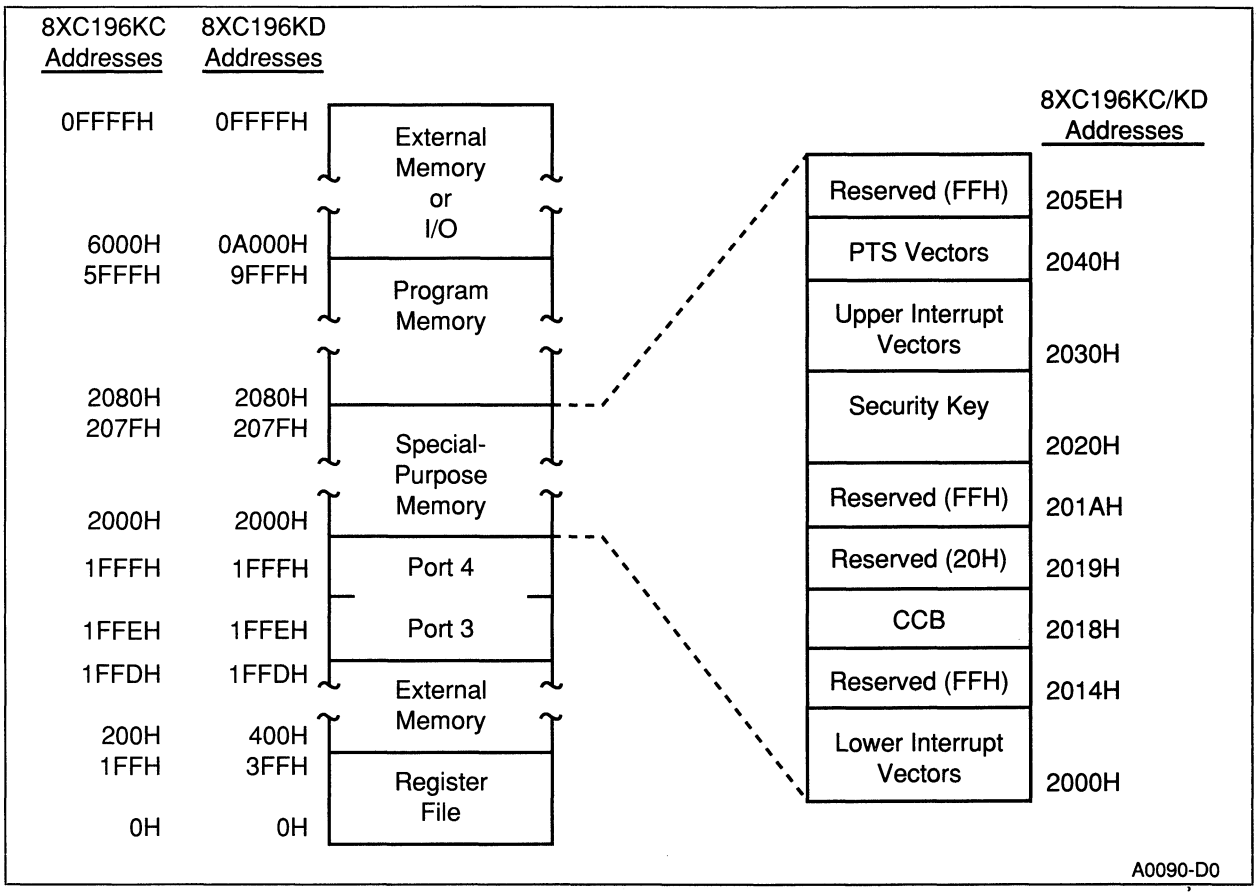

Figure 4-2. 8XC196KC/KD Special-Purpose Memory Map

4.3.3.1. RESERVED MEMORY LOCATIONS

Several memory locations are reserved for testing or future product use. The user program must not read or write these reserved memory locations. The contents and/or function of these locations may change in future revisions, so software that uses reserved locations may not function properly.

When programming the OTPROM, always write 20H to reserved address 2019H and 0FFH to all other reserved addresses.

4.3.3.2. INTERRUPT VECTORS

The Upper and Lower Interrupt Vectors contain the addresses of the interrupt service routines. The Peripheral Transaction Server (PTS) vectors contain the addresses of the PTS control blocks. See Chapter 5, "Interrupts," for more information.

Table 4-2. 8XC196KC/KD Special-Purpose Memory Addresses

Description	8XC196KC/KD Address Range	
	Hexadecimal	Decimal
Reserved (must contain 0FFH)	205EH–207FH	8286–8319
PTS Vectors	2040H–205DH	8256–8285
Upper Interrupt Vectors	2030H–203FH	8240–8255
Security Key	2020H–202FH	8224–8239
Reserved (must contain 0FFH)	201AH–201FH	8218–8223
Reserved (must contain 20H)	2019H	8217
CCB	2018H	8216
Reserved (must contain 0FFH)	2014H–2017H	8212–8215
Lower Interrupt Vectors	2000H–2013H	8192–8211

4.3.3.3. SECURITY KEY

The security key protects the 8XC196KC/KD against unauthorized reading and writing of OTPROM. If the security lock bits in the Chip Configuration Register (CCR.6 and CCR.7) are cleared, each byte of the 16-byte programmable security code is compared to a corresponding byte in external memory. If the external data does not match the security key, the device does not enter into the requested programming mode. See Chapter 14, "Programming the Nonvolatile Memory," for more information about the security key.

4.3.3.4. CHIP CONFIGURATION BYTE

The Chip Configuration Byte (CCB) is the first byte fetched from memory after a device reset. The reset sequence loads the CCB into an internal register called the Chip Configuration Register (CCR). The CCR controls internal memory protection, internal READY mode, bus control signals, bus-width options, and Powerdown mode (see Appendix C, "8XC196KC/KD Registers").

4.4. REGISTER FILE

The Register File is divided into an upper and a lower Register File (see Figure 4-3). The upper Register File contains general-purpose register RAM. The lower Register File contains general-purpose register RAM, the Stack Pointer (SP), and CPU Special Function Registers (SFRs). Table 4-3 compares the 8XC196KC and 8XC196KD Register File memory addresses and provides beginning and ending addresses in both hexadecimal and decimal.

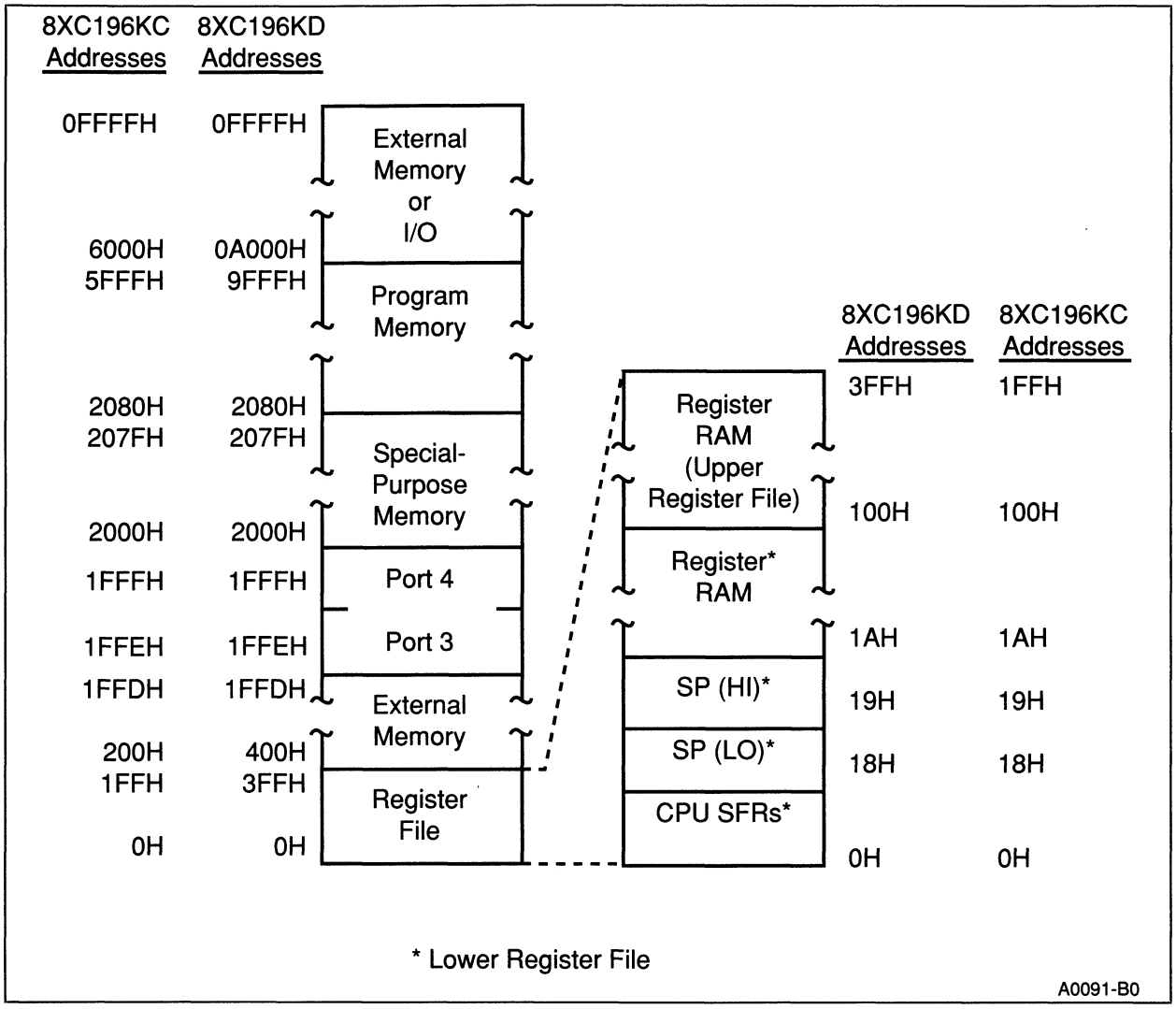

Figure 4-3. Register File Memory Map

NOTE

The Register File must not contain code. An attempt to execute instructions from Register File memory locations (0H–01FFH in the 8XC196KC or 0H–03FFH in the 8XC196KD) causes the memory controller to automatically fetch instructions from external memory. This section of external memory is reserved for use by Intel development tools.

Table 4-3. 8XC196KC/KD Register File Memory Addresses

Description	8XC196KC Address Range		8XC196KD Address Range	
	Hexadecimal	Decimal	Hexadecimal	Decimal
Upper Register File (Note 1)	100H–1FFH	256–511	100H–3FFH	256–1023
General-purpose register RAM (Note 2)	1AH–0FFH	26–255	1AH–0FFH	26–255
Stack Pointer (SP) (Note 2)	18H–19H	24–25	18H–19H	24–25
Special Function Registers (SFRs) (Note 2)	0H–17H	0–23	0H–17H	0–23

NOTES:

1. Contains general-purpose register RAM (accessed by indirect or indexed addressing unless vertical windowing is used).
2. Located in the lower Register File (accessed by either register-direct, indirect, or indexed addressing).

4.4.1. General-Purpose Register RAM

Locations 1AH–0FFH contain general-purpose register RAM. The stack pointer locations, 18H and 19H, can also be used as general-purpose register RAM when stack operations are not being performed. The RALU can access this memory directly, using register-direct addressing. The upper Register File also contains general-purpose register RAM (100H–1FFH in the 8XC196KC and 100H–3FFH in the 8XC196KD). The RALU normally accesses this memory using indirect or indexed addressing. *Vertical windowing* is a technique that enables the RALU to use register-direct addressing to access the RAM in the upper Register File. (See Chapter 3, "Data Types and Addresses," for a description of differences between register-direct and indexed addressing.) Vertical windowing provides fast context switching of interrupt tasks and faster program execution. See "Vertical Windowing" on page 4-12.

4.4.2. Stack Pointer (SP)

Memory locations 18H and 19H contain the stack pointer (SP). The SP contains the address of the stack. The SP must point to a word (even) address that is two bytes greater than the desired starting address. Before the CPU executes a subroutine call or interrupt service routine, it decrements the SP twice and then copies (PUSHes) the address of the next instruction from the program counter onto the stack. It then loads the address of the subroutine or interrupt service routine into the program counter. After it completes the subroutine or interrupt service routine, the CPU executes a return from subroutine instruction (RET) and loads (POPs) the contents of the top of the stack (that is, the return address) into the program counter.

Subroutines may be nested. That is, each subroutine may call other subroutines. The CPU pushes the contents of the program counter onto the stack each time it executes a subroutine call. The stack grows downward as entries are added. The only limit to the nesting depth is the amount of available memory. As the CPU returns from each nested subroutine, it pops the address off the top of the stack and the next return address moves to the top of the stack.

When stack operations are not being performed, the SP locations may be used as general-purpose register RAM.

4.4.2.1. INITIALIZING THE STACK POINTER

The user program must load a word-aligned (i.e., even) address into the stack pointer. Select an address that is two bytes greater than the desired starting address because the stack pointer is automatically decremented before the CPU pushes the first byte of the return address onto the stack. Remember that the stack grows downward, so allow sufficient room for the maximum number of stack entries. The stack may be located in either the internal Register File or external RAM.

The following example initializes the top of the 8XC196KD's upper Register File as the stack.

```
LD SP, #400H                    :Load stack pointer
```

4.4.3. Special Function Registers

Locations 00H–17H provide access to the CPU Special Function Registers (SFRs) through three *horizontal windows* (HWindow 0, 1, and 15). (See "Horizontal Windowing" on page 4-8.) The RALU has direct control of all peripheral modules, except Ports 3 and 4, through the SFRs. Appendix C, "8XC196KC/KD Registers," describes each register in detail.

When using an SFR as a base or index register for indirect or indexed operations, be aware that the contents of the SFRs are not always predictable. External events can change the contents of SFRs, and some SFRs are cleared when read.

The functions of many SFRs change depending upon whether they are being read from or written to. For this reason, never use an SFR as an operand in a read-modify-write instruction (e.g., XORB, AD_RESULT).

Do not use reserved SFRs; write zeros to them or leave them in their default state. When read, reserved bits and SFRs will return undefined values.

4.5. HORIZONTAL WINDOWING

The 8XC196KC/KD uses a *horizontal windowing* scheme that swaps three 24-byte memory blocks (HWindow 0, 1, and 15) within the lowest 24 bytes of the lower Register File (see Figure 4-4). Each HWindow provides read or write access to a unique combination of SFRs. Some registers are accessed as a single byte; others are accessed as a word (two bytes). Some registers, such as the Window Select Register (WSR, 14H), are accessible in all three HWindows. Others must be written in one HWindow and read in another. Appendix C, "8XC196KC/KD Registers," provides detailed descriptions of each register.

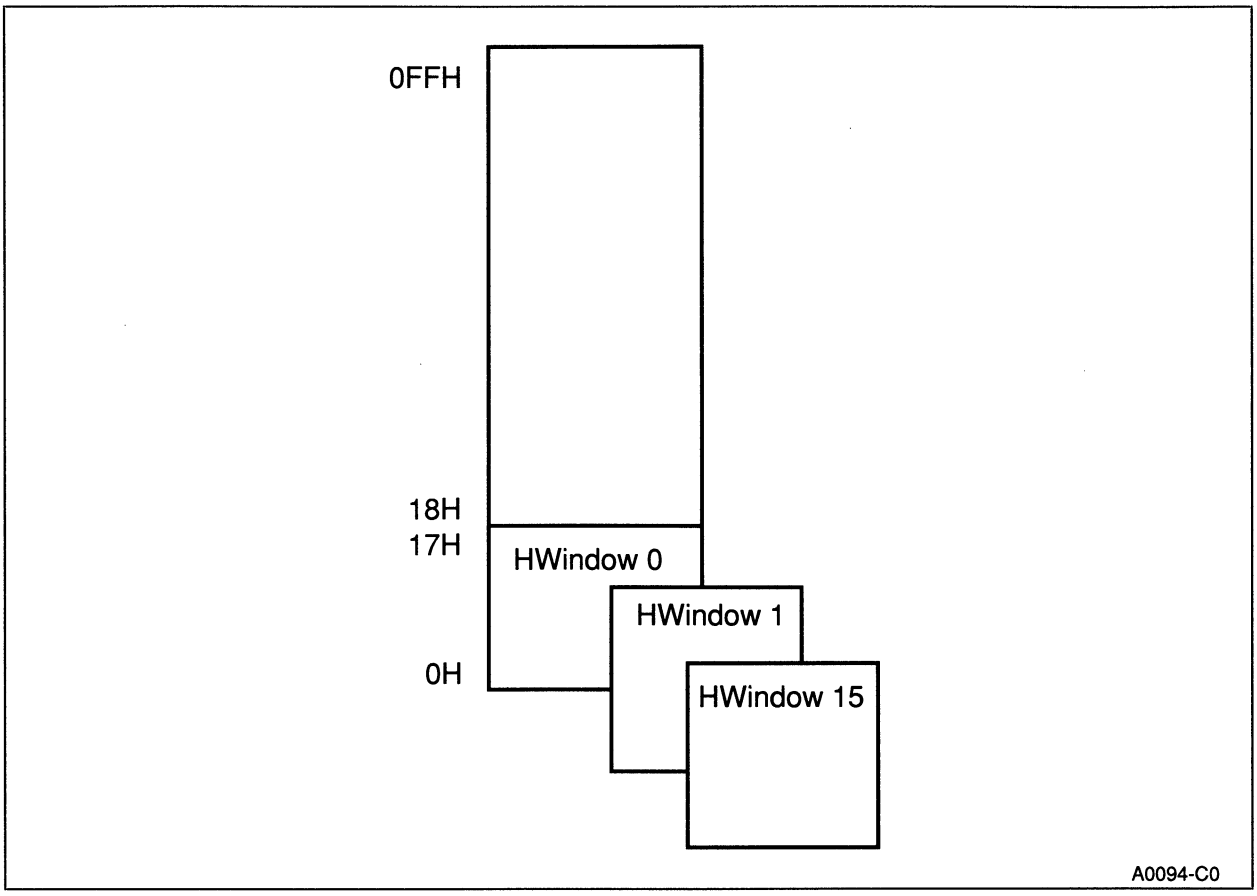

Figure 4-4. Horizontal Windowing

4.5.1. Selecting an HWindow

The Window Select Register (WSR, 14H) provides access to HWindows and VWindows (see "Vertical Windowing" on page 4-12). To select an HWindow, write the number of the desired window into WSR.0–WSR.3 and clear WSR.4–WSR.6. Only HWindows 0, 1, and 15 are available. All other HWindows are reserved. Table 4-4 shows the appropriate WSR contents for each HWindow selection. See Appendix C, "8XC196KC/KD Registers," for a complete description of the WSR.

Table 4-4. HWindow Selections

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

4.5.2. HWindow 0

HWindow 0 is the default window. It provides read access to 19 registers and write access to 21 registers (see Figure 4-5). Some registers (e.g., INT_MASK1) can be both read and written within HWindow 0. Others (e.g., IOS1) can be either read or written, but not both, within HWindow 0. For these registers, select HWindow 15 to perform the complementary function.

4.5.3. HWindow 1

HWindow 1 provides read and write access to 12 registers (see Figure 4-5). Most support those features that differentiate the 8XC196KC and 8XC196KD from other, earlier members of the MCS-96 family. Some registers are also accessible in both HWindow 0 and HWindow 15.

	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	PTSSEL (HI)
04H	HSI_TIME (LO)	HSO_TIME (LO)	04H	PTSSEL (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)

Figure 4-5. HWindow 0 and HWindow 1

4.5.4. HWindow 15

HWindow 15 provides access to the same registers that HWindow 0 does, except for bytes 0CH–10H. Bytes 0CH–0DH access the 16-bit TIMER2 register in HWindow 0, but they access the 16-bit T2CAPTURE register in HWindow 15. Bytes 0EH–10H access the IOPORT0, IOPORT1, and IOPORT2 registers in HWindow 0, but they are reserved in HWindow 15. Those registers that are read-only in HWindow 0 are write-only in HWindow 15, and vice versa.

	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)

Figure 4-6. HWindow 15

4.6. VERTICAL WINDOWING

Vertical windowing maps sections of the upper Register File into the upper locations of the lower Register File, in 32-, 64-, or 128-byte increments known as VWindows. The 8XC196KC has sixteen 32-byte VWindows, eight 64-byte VWindows, and four 128-byte VWindows. Since the 8XC196KD has twice as much RAM as the 8XC196KC, it also has twice as many VWindows.

Figure 4-7 is an example of a 128-byte VWindow.

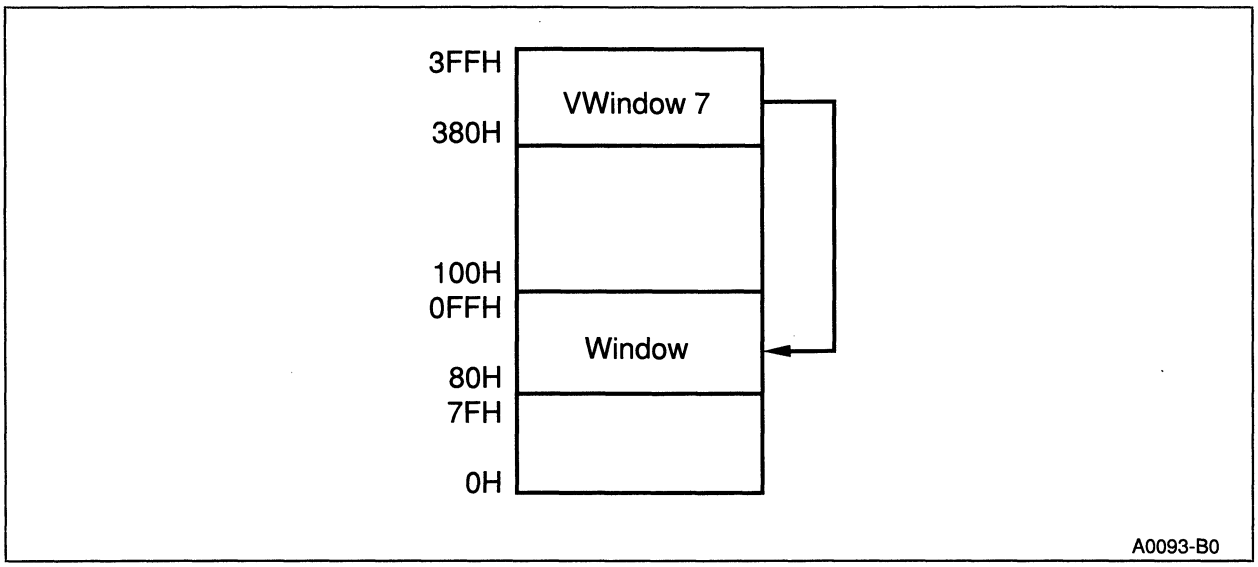

Figure 4-7. Vertical Windowing

4.6.1. Selecting a VWindow

The Window Select Register (WSR, 14H) provides access to HWindows and VWindows. Set WSR.4, WSR.5, or WSR.6 to select a 128-, 64-, or 32-byte VWindow respectively (see Figure 4-8). Write the VWindow number into lower WSR bits. Appendix C, "8XC196KC/KD Registers," provides a complete description of the WSR, with tables that list the appropriate WSR contents for each VWindow selection. (To select the vertical window shown in Figure 4-7, load 17H into the WSR.)

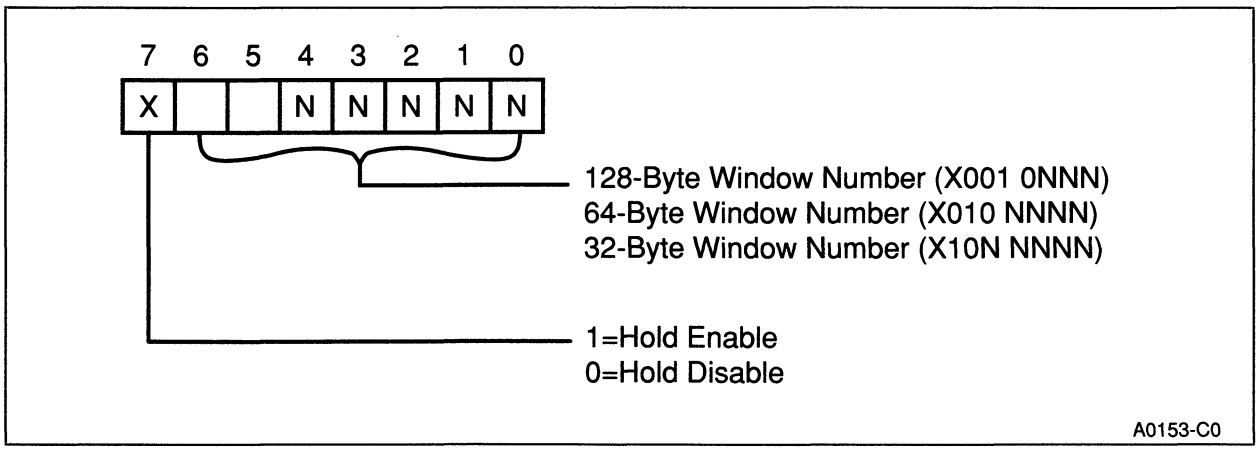

Figure 4-8. Window Select Register Bit Settings

4.6.2. Vertical Windowing and Addressing Modes

When vertical windowing is enabled:

- a register-direct instruction that uses an address within the lower Register File window actually accesses the VWindow in the upper Register File;
- an indirect or indexed instruction that uses either an address within the lower Register File window or the VWindow accesses the actual location in memory.

NOTE

Indirect shift operations will not perform correctly if WSR = X100 0000.

The following sample code illustrates the difference between register-direct and indexed addressing when using vertical windowing.

```
PUSHA                   ;pushes the contents of WSR onto the stack
LDB WSR, #17H           ;select VWindow 7, a 128-byte block
                        ;The next instruction uses register-direct addr
ADD 40H, 80H            ;mem_word(40H)←mem_word(40H) + mem_word(380H)
                        ;
                        ;The next two instructions use indirect addr
ADD 40H, 80H[0]         ;mem_word(40H)←mem_word(40H) + mem_word(80H + 0)
                        ;
ADD 40H, 380H[0]        ;mem_word(40H)←mem_word(40H) + mem_word(380H + 0)
                        ;
POPA                    ;reloads the previous contents into WSR
```