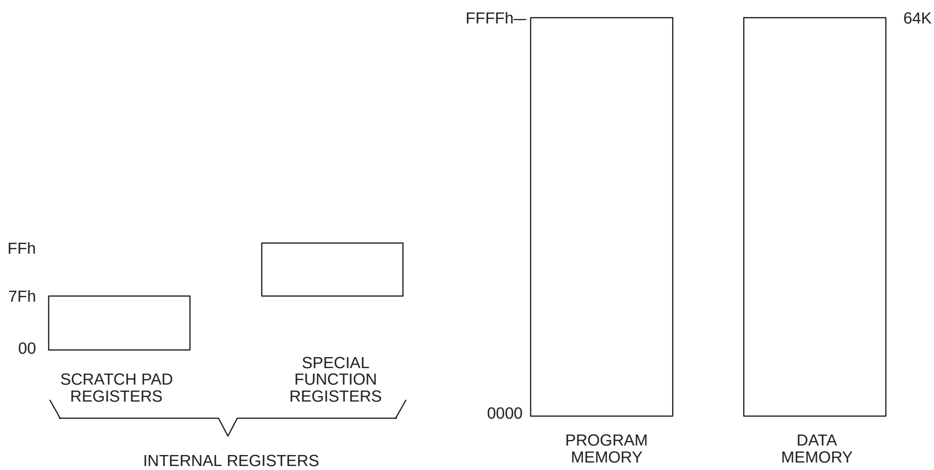
SECTION 4: PROGRAMMER'S GUIDE

The Secure Microcontroller uses nonvolatile RAM technology for both Program and Data memory. It uses NV SRAM in place of ROM by write protecting and decoding memory segments that a user designates as Program memory. The remaining RAM area is used as nonvolatile data storage. One of the advantages of breaking a common RAM into two segments is that a smaller number of memory chips is needed. For example, if a system requires 24K bytes of program memory and 4K bytes of data memory, this all fits within one 32K x 8 SRAM. The Secure Microcontroller can break this RAM into program and data segments, unconditionally write protecting the program area. The process of dividing the common memory space into ROM and RAM is called partitioning. All Secure Microcontrollers are capable of doing this. However, there are differences between original DS5000 series [includes DS5000FP, DS5000(T), and DS2250T] and newer DS5001 series [includes DS5001FP, DS2251T, DS5002FP, DS2252T]. The original DS5000 series could partition one SRAM of up to 32K bytes. It could access a second RAM, but this was restricted to data memory only. The DS5001 series can partition two 32K byte SRAMs, or even one 128K x 8 SRAM. Common elements of the programming model are given below, with individual differences highlighted.

Secure Microcontroller Memory Organization

All Secure Microcontrollers follow the standard 8051 convention of three memory areas. These include Internal registers, Program memory and Data memory. These memory areas are not contiguous and are accessed in different ways. The Secure Microcontroller duplicates all standard 8051 registers and adds several new ones. Secure Microcontrollers have a 64K byte program and 64K byte data space. However, the Secure Microcontrollers provide several ways to access these areas, and these features are what make the family unique. Figure 4–1 shows the memory map of Secure Microcontrollers in general terms. The specific details and access to the memory areas are discussed below.

SECURE MICROCONTROLLER MEMORY MAP Figure 4–1



Internal Registers

The internal register space is divided into two parts. These are Scratchpad Registers and Special Function Registers (SFRs). There are a total of 128 Scratchpad registers, commonly referred to as on–chip RAM. The 128 bytes include four 8–byte banks of working registers (R0–R7). The Scratchpad Registers are located at register addresses 00–7Fh. This area is not located in the Program or Data Memory area and is accessed by different instructions. The Special Function Registers (SFR) are located in the locations between 80h and FFh. SFRs control the on–chip peripherals and memory configurations. Direct addressing should be used to access the SFR locations. If Register–Indirect addressing is used, indeterminate data will be returned. Scratchpad Registers are discussed immediately below, with SFR descriptions following later in this section.

The Scratchpad Registers are general purpose data storage RAM. They are commonly used for temporary storage of a small number of variables when high–speed access is needed. Off–chip RAM (MOVX) is used when the quantity of data is larger than 128 bytes. The Scratchpad Registers are lithium backed and will be preserved in the absence of power.

The Scratchpad area has two additional functions. First, 16 bytes of the Scratchpad area are bit addressable. That is, while each byte has an address of its own, these bits also have individual bit addresses. Certain instructions operate on bits instead of bytes. Although the addresses appear the same, the microprocessor can distinguish a bit address from a byte address by the instruction used. A large number of individual software flags and conditions can be represented using 128 (16*8) individually addressable bits.

A second use of the Scratchpad area is for the programmer's stack. Like the 8051, the Secure Microcontroller uses a Stack Pointer (SP – 81h) SFR to direct stack access into the internal registers. The SP has a default value of 07h. This means that stack storage will begin at location 08h. Each PUSH or CALL instruction will increment the SP. Note that while the SP is located in the SFR area, the stack itself is stored in the Scratchpad area. The Scratchpad Register Memory map is shown in Figure 4–2. *Programmer's note* : with the use of 'C' compilers becoming more frequent, the large memory model should be examined. This compiler model places the stack in off–chip SRAM. Secure Microcontroller based systems usually have an abundance of such SRAM compared to ROM based systems. While off–chip stack results in slower execution time, the stack size becomes virtually unlimited.

SCRATCHPAD REGISTER MAP Figure 4–2

	MSB							LSB
7FH								
2FH	7F	7E	7D	7C	7B	7A	79	78
2EH	77	76	75	74	73	72	71	70
2DH	6F	6E	6D	6C	6B	6A	69	68
2CH	67	66	65	64	63	62	61	60
2BH	5F	5E	5D	5C	5B	5A	59	58
2AH	57	56	55	54	53	52	51	50
29H	4F	4E	4D	4C	4B	4A	49	48
28H	47	46	45	44	43	42	41	40
27H	3F	3E	3D	3C	3B	3A	39	38
26H	37	36	35	34	33	32	31	30
25H	2F	2E	2D	2C	2B	2A	28	28
24H	27	26	25	24	23	22	21	20
23H	1F	1E	1D	1C	1B	1A	19	18
22H	17	16	15	14	13	12	11	10
21H	0F	0E	0D	0C	0B	0A	09	08
20H	07	06	05	04	03	02	01	00
1FH–18H	BANK 3							
17H–10H	BANK 2							
0FH–08H	BANK 1							
07H–00H	BANK 0							

The 8051 instruction set allows efficient (single cycle) access to variables when using the Working Registers. These are a group of four 8–byte banks of Scratchpad RAM. The active Working Registers are referred to as R0–R7. They reside between location 00h and 1Fh, depending on which bank is currently selected. Two bits in the Special Function Register PSW called R1 (PSW.4) and R0 (PSW.3) are used to determine which is the active bank. Once selected, all instructions involving R0–R7 will be directed to the selected group of 8 bytes. This scheme also allows for a fast context switch by simply changing banks. The following Table shows the operation of the Register Bank selection.

PSW.4–3 ; R1–R0
Register Bank Select

Used to select an 8–byte bank of registers to be assigned as R0–R7.

R1	R0	BANK STARTING ADDRESS (R0)
0	0	00h
0	1	08h
1	0	10h
1	1	18h

Program and Data Memory

The Secure Microcontroller divides its main memory between Program and Data segments. Each map consists of a 64K byte area from 0000h to FFFFh. Program memory is inherently read only, since there are no 8051 instructions that write to this segment. Data memory is read and write accessible without restrictions. The CPU automatically routes program fetches to the program area and MOVX instructions to the data memory area. All of these elements are in common with the standard 8051. Secure Microcontroller differences lie in the memory interface, memory map control, and flexibility of the memory resources.

Secure Microcontrollers provide two separate buses for memory access. First is a Byte–wide address/data bus which is new to the 8051 architecture. This bus also provides a switched supply output that make standard SRAM into nonvolatile memory, decoded chip enables, and a R/$\overline{W}$ strobe. Furthermore, the Byte–wide bus allows nonvolatile RAM memory to be divided between Program and Data segments. When using a segment of the RAM as Program Memory, this area can be loaded using the Bootstrap Loader function described later in this book.

Second is an Expanded bus constituted by Ports 0 and 2. This is the standard 8051 compatible memory bus which is available as an option, but is not needed in most cases. Program memory on the Expanded bus must be ROM/EPROM and data memory must be volatile SRAM. If NV RAM is needed on the Expanded bus, then it must be externally backed up and write protected. The Secure Microcontroller makes no special provisions for NV RAM on the Expanded bus.

When discussing memory addressing of Secure Microcontrollers, there are two important terms that are used frequently: Partition and Range. The Partition is the user–selectable address that divides the program segment from the data segment in a common RAM area on the Byte–wide bus. The Partition is a user–adjustable boundary that can be selected during Bootstrap Loading or on the fly by the application software. The Range is the total amount of memory connected to the Byte–wide bus. This is set once during initial programming.

The DS5000 series devices can access between 8K and 64K bytes of NV RAM on the Byte–wide bus. Up to the first 32K bytes are Partitionable into Program and Data segments as described above. The DS5001 series can access between 8K and 128K bytes on its Byte–wide bus with better Partition control. The Memory map control resides in the MCON (address C6h) Special Function Register on DS5000 devices. On DS5001 devices, both the MCON (address C6h) and RPCTL (address D8h) registers are used. Since the memory maps and control have significant differences between these versions, they are described below in separate sections.

DS5000 Series Memory Organization

As mentioned above, the DS5000 series consists of the DS5000FP chip and the DS5000(T) and DS2250T modules. The programming model discussed in this section applies to all of these parts. The DS5000 series Byte–wide bus has 15 address lines, eight data lines, a R/$\overline{\text{W}}$ strobe, and two chip enables to access nonvolatile RAM. In the case of a module, these are already connected and may be thought of as internal or embedded memory. The DS5000 series can use either 8K x 8 or 32K x 8 SRAMs. The user must inform the microcontroller of the selected RAM size using the Range function. The Range bit resides in the MCON SFR at MCON.3 and has a value of 0 when 8K SRAM is used and 1 when a 32K byte SRAM is used. Range is selected during Bootstrap Loading and can not be varied by the application software. The DS5000 device accesses memory on its Byte–wide bus using two chip enables. The first, $\overline{\text{CE1}}$, is Partitionable. That is, the RAM connected to $\overline{\text{CE1}}$, whether 8K or 32K, can be divided between program and data segments. The Partition is user–selected and can be set during Bootstrap Loading and by software. Partitions are generally available on 2K byte boundaries in the DS5000 except for the last which is 4K. The Partition is selected using the MCON SFR described below. $\overline{\text{CE2}}$ is restricted to data memory only. The RAM on $\overline{\text{CE2}}$ should be of the same size as $\overline{\text{CE1}}$. Access to $\overline{\text{CE2}}$ is manual, and functions like a bank switch. Bit 2 (ECE2) of the MCON SFR controls access to $\overline{\text{CE2}}$ and is described below.

Figure 4–3 illustrates the functional memory map of a DS5000 series device. The Partition, Range, ECE2, and the logical address combine to determine whether the DS5000 uses its Byte–wide bus or the Expanded Bus. Nonvolatile RAM access will occur when the logical address lies in one of the shaded regions. These are program addresses below the Partition address, data addresses above the Partition and below the Range address, or data addresses between 0 and the Range when ECE2 is set to a logic 1. Note that when using ECE2 to force data access, the $\overline{\text{CE2}}$ RAM will be selected instead of the $\overline{\text{CE1}}$ RAM. This means that on a DS5000 module or a DS2250 with less than 64K RAM, no data memory exists under $\overline{\text{CE2}}$. The ECE2 has no affect on program memory, which continues from the $\overline{\text{CE1}}$ RAM or the Expanded bus normally.

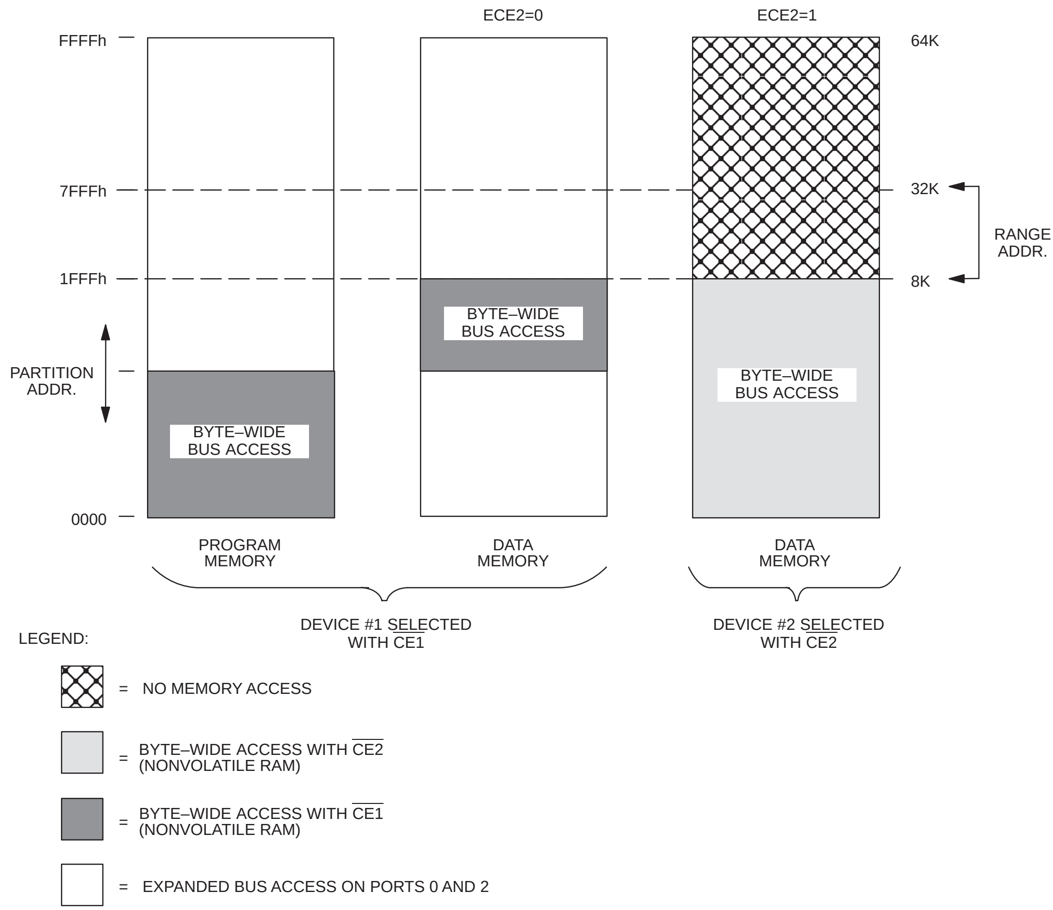
Note that the Partition and Range settings are not automatically linked. This means a user should take care not to select a Partition that is larger than the Range. Naturally when the Range is 32K, the Partition address can be as high as 32K. When a Range of 8K is used, Partition addresses below 8K should be used. Any address that does not map onto the Byte–wide bus will be automatically be routed to the Expanded Bus of Ports 0 and 2. For module users. this means that any address not routed to internal memory will go to the ports. The following examples will help illustrate the decoding.

When the Partition is at 3000h, and the Range at 32K, program memory below 3000h is accessed on the Byte–wide bus. Program memory at or above 3000h is directed to the Expanded bus or Ports 0 and 2. When the Partition is at 5800h and the Range at 32K, data memory at 0000h is accessed on Ports 0 and 2. Data memory at 6000h is located in NV RAM on the Byte–wide bus. When the Partition is at 1000h and the Range at 8K, all memory access above 1FFFh is on the Expanded bus. Below 8K, the Partition rules apply.

IMPORTANT APPLICATION NOTE

The MCON register is a special function register unique to Dallas Semiconductor microcontrollers which contains nonvolatile memory configuration information. This register should be set to the desired value before loading the device via the bootstrap loader. Failure to correctly configure the MCON register can cause the device to operate incorrectly, including symptoms which appear similar to a defective device. Because this register is nonvolatile, incorrect memory settings will be preserved when power is removed. The DS5001FP, DS5002FP, DS2251T, and DS2252T store additional memory configuration information in the RPCTL register, which should also be set to the desired value before loading the device via the bootstrap loader.

DS5000 SERIES MEMORY MAP Figure 4–3



The above memory map covers the standard operating case. There are two conditions that can modify this memory map. The first is the $\overline{\text{EA}}$ pin. The second is the Security Lock. When the $\overline{\text{EA}}$ pin is grounded, the DS5000 will force all memory access to the Expanded bus. This causes the DS5000 to behave like an 8031 regardless of the Partition, Range, or ECE2. The $\overline{\text{EA}}$ should be pulled to +5V for normal operation. The second modifier is the Security Lock. When set, the Security Lock prevents using the Bootstrap Loader to read the contents of the NV RAM. For security purposes, it also prohibits program memory access on the Expanded Bus. Thus all program fetches must be restricted to the Byte–wide bus when locked. The Security Lock overrides the condition of the $\overline{\text{EA}}$ pin as well.

The selection of memory map controls provide unprecedented flexibility to configure a system. However, it is possible to select contradictory settings. The micro will compensate for these as follows. The Partitioning function allows a user to select the quantity of program and data memory. It is possible to select all data and no program in NV RAM by choosing a Partition of 0000h. This is a valid selection. However, using this setting and the Security Lock is a conflict. This condition asks the micro to use all program memory on the Expanded bus, but also to prohibit the use of program memory on the Expanded bus. In this event, special circuits will automatically force the Partition to a location of 7FFFh. This means all 32K memory on the Byte–wide bus is designated program memory. The second contradictory case is to select a Range of 8K, and to choose a Partition of greater than 8K. This will result in the Range as the limiting factor. Addresses above the Range will automatically be deflected to the Expanded bus. No data memory will be allocated in NV RAM for this configuration.

DS5000 Memory Map Control

The Partition and Range can be selected using the Bootstrap Loader discussed in a later section. In addition, the Partition can be selected or modified by the application software and $\overline{CE2}$ is normally software controlled. However, in either case, the MCON SFR is used to choose these settings. The MCON is summarized in the SFR section below, but appears here also.

DS5000 SERIES MCON REGISTER Figure 4–4

Bit Description:

MCON.7–4: **PA3–0**

"Partition Address": Use to select the starting address of Data Memory in Embedded RAM. Program space lies below the Partition address.

Selection:

PA3	PA2	PA1	PA0	Partition Address
0	0	0	0	0000H
0	0	0	1	0800H
0	0	1	0	1000H
0	0	1	1	1800H
0	1	0	0	2000H
0	1	0	1	2800H
0	1	1	0	3000H
0	1	1	1	3800H
1	0	0	0	4000H
1	0	0	1	4800H
1	0	1	0	5000H
1	0	1	1	5800H
1	1	0	0	6000H
1	1	0	1	6800H
1	1	1	0	7000H*
1	1	1	1	8000H*

*A 4K byte increment (not 2K bytes) in the Partition Address takes place between bit field values 1110B and 1111B.

Initialization: Set to all 1's on a No V_{LI} Power On Reset or when the Security Lock bit is cleared to a 0 from a previous 1 state. These bits are also set to all 1's when any attempt is made to have them cleared to all 0's with the SL bit set to a 1 (illegal condition).

Read Access: May be read anytime.

Write Access: PAA bit must = 1 in order to write PA3–0. Timed Access is not required to write to PA3–0 once PAA = 1.

MCON.3: **RA32/8**

"Range Address": Sets the maximum usable address on the Byte–wide bus. RA32/8 = 0 sets Range Address = 1FFFH (8K); RA32/8 = 1 sets Range Address = 7FFFH (32K)

Initialization: Set to a 1 on a No V_{LI} Power On Reset and when the Security Lock bit (SL) is cleared to a 0 from a previous 1 state. Remains unchanged on all other types of resets.

Read Access: May be read normally anytime.

Write Access: Cannot be modified by the application software; can only be written during Program Load mode.

MCON.2: **ECE2**

"Enable Chip Enable 2": Used to enable or disable the $\overline{CE2}$ signal to additional RAM Data Memory space. This bit should always be cleared to 0 in the DS5000–8, DS5000–32, DS2250–8, and DS2250–32 versions.

Initialization: Cleared to 0 only during a No V_{LI} Power On Reset.

Read Access: Read normally anytime.

Write Access: Can be written normally anytime.

MCON.1: **PAA**

"Partition Address Access": Used to protect the programming of the Partition Address select bits. PA3–0 cannot be written when PAA=0. PAA can be written only via the Timed Access register.

Initialization: PAA is cleared on a reset.

Read Access: PAA may be read anytime.

Write Access: The Timed Access register must be used to perform any type of write operation on the PAA bit.

DS5001/DS5002 Memory Organization

As mentioned above, the DS5001/DS5002 series consists of the DS5001FP chip, the DS2251T module, the DS5002FP chip, and the DS2252T module. Note that the DS5002FP is a high security version of the DS5001FP, but has the same memory map and I/O. The programming model discussed in this section applies to all of these parts and any reference to the DS5001 applies to all of them. The DS5001 series Byte–wide bus has 16 address lines, eight data lines, a R/$\overline{W}$ strobe, and a total of eight chip enables to access nonvolatile RAM and peripherals. Chip enables include $\overline{CE1}$ – $\overline{CE4}$ and $\overline{PE1}$ – $\overline{PE4}$. The four chip enables ($\overline{CE1}$–4) are for nonvolatile RAM access. How they are connected depends on the memory mode and the selection of SRAMs. The $\overline{PE}$ signals are generally for memory mapped peripherals, but can be used for more RAM if desired. $\overline{PE1}$ and $\overline{PE2}$ are lithium–backed, $\overline{PE3}$ and $\overline{PE4}$ are not. In the case of a module, $\overline{PE1}$ may be connected to a real–time clock. Memory map control resides in the MCON (C6h) and RPCTL (D8h) registers. The MCON register has selected differences from its DS5000 counterpart. These are documented below. The RPCTL is not present in the DS5000. Also, not all of the bits in this register pertain to memory map control. This section describes the relevant bits and the SFR section below documents the entire register.

The DS5001 series can use multiple 8K x 8 or 32K x 8 SRAMs or a single 128K x 8 SRAM. These parts can operate in either a Partitionable (like DS5000) or non–partitionable mode. The mode is selected via the PM (MCON.1) bit of the MCON register. Note, the DS5001 MCON provides different functions than the DS5000. In a Partitionable mode (PM=0), the DS5001 can use up to 64K x 8 SRAM for program and data on its Byte–wide bus. It can partition this area into program and data segments on 4K boundaries. The 64K memory space would consist of two 32K x 8 SRAMs. Each is accessed by a separate chip enable ($\overline{CE1}$ and $\overline{CE2}$), but the microcontroller automatically decodes which is needed. While the DS5001 can use between one 8K x 8 SRAM and 4 32K x 8 SRAMs, it does not automatically know which configuration is used. The Range function determines how much total memory is connected to the Byte–wide bus. The user must identify the total RAM size using the Range bits RG1 and RG0. RG1 is located at MCON.3 and RG0 is located at RPCTL.0. These Range bits are selected during the Bootstrap Loading process and can not be modified by the application software. The Table below shows the Range values that can be selected when PM=0 (Partitionable).

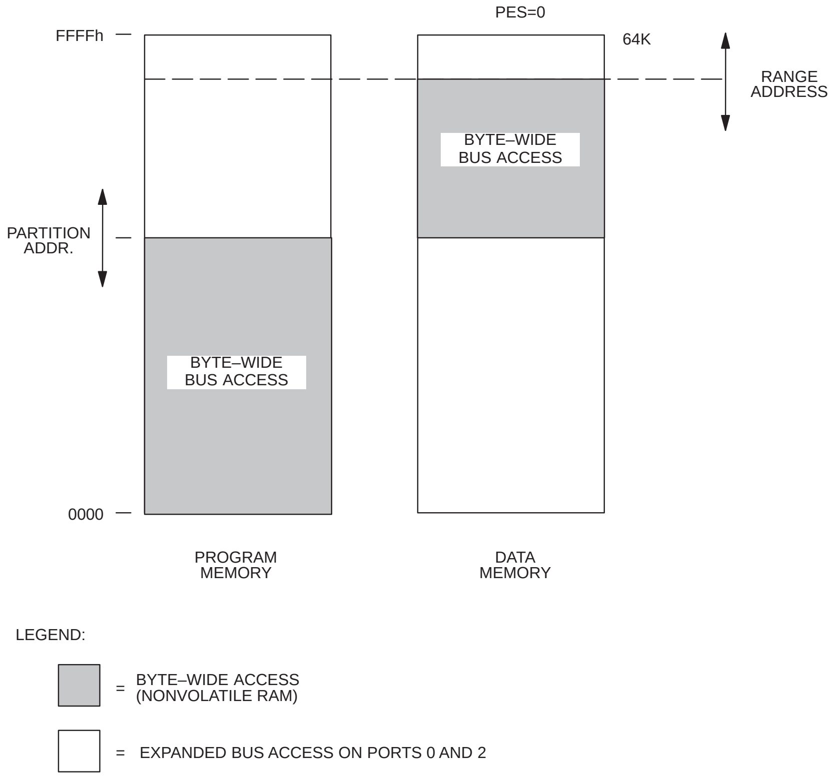
RG1	RG0	RANGE	$\overline{CE1}$ ACCESS	$\overline{CE2}$ ACCESS
1	1	64K	0000–7FFFh	8000–FFFFh
1	0	32K	0000–7FFFh	NA
0	1	16K	0000–1FFFh	2000h–3FFFh
0	0	8K	0000–1FFFh	NA

The total RAM space is partitionable, regardless of which Range is selected. This contrasts with the DS5000 that allowed partitioning of $\overline{CE1}$ only. The Partition table is shown below. PA3–0 are the four MSBs of the MCON register (MCON.7–4). Note that the Partition values do not scale depending on Range. That is, if a Range of less than 64K is selected, then the partition settings above the Range should not be unused. The microcontroller automatically decodes which RAM to enable, and uses the Partition to decide if this is program memory or data memory.

PA3	PA2	PA1	PA0	PARTITION	BYTE–WIDE BUS MEMORY MAP
0	0	0	0	0000h	0K PROGRAM, DATA = RANGE
0	0	0	1	1000h	4K PROGRAM, DATA = RANGE – 4K
0	0	1	0	2000h	8K PROGRAM, DATA = RANGE – 8K
0	0	1	1	3000h	12K PROGRAM, DATA = RANGE – 12K
0	1	0	0	4000h	16K PROGRAM, DATA = RANGE – 16K
0	1	0	1	5000h	20K PROGRAM, DATA = RANGE – 20K
0	1	1	0	6000h	24K PROGRAM, DATA = RANGE – 24K
0	1	1	1	7000h	28K PROGRAM, DATA = RANGE – 28K
1	0	0	0	8000h	32K PROGRAM, DATA = RANGE – 32K
1	0	0	1	9000h	36K PROGRAM, 28K DATA
1	0	1	0	A000h	40K PROGRAM, 24K DATA
1	0	1	1	B000h	44K PROGRAM, 20K DATA
1	1	0	0	C000h	48K PROGRAM, 16K DATA
1	1	0	1	D000h	52K PROGRAM, 12K DATA
1	1	1	0	E000h	56K PROGRAM, 8K DATA
1	1	1	1	FFFFh	64K PROGRAM, 0K DATA

Figure 4–5 illustrates the functional memory map of a DS5001 series device in Partitionable mode. Note that like the DS5000, any access that does not correspond to a Byte–wide bus location is routed to the Expanded bus Ports 0 and 2.

PARTITIONABLE MEMORY MAP FOR DS5001/DS5002 SERIES Figure 4–5

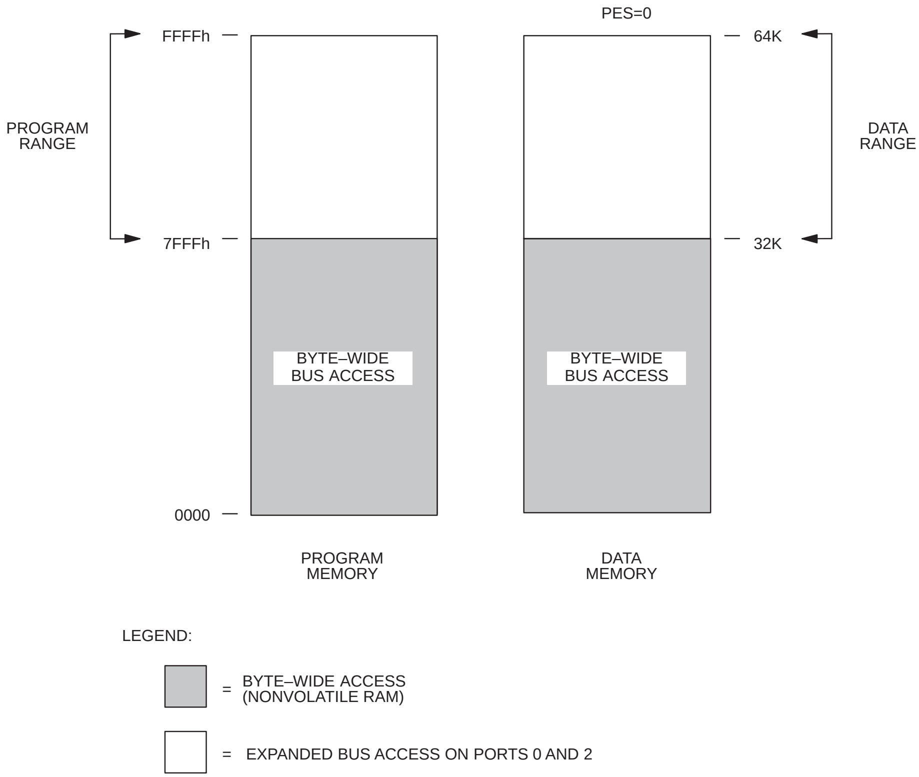


The non–partitionable mode allows the maximum amount of memory to be used on the Byte–wide bus. A non–partitionable mode would be used because the 8051 architecture is restricted to a total of 64K program and 64K data (without bank switching). This means that if the maximum amount of either program or data (or both) is needed, partitioning can not be done. The DS5001/DS5002 series accommodates these situations with four selections of non–partitionable (PM=1) memory control shown below. These are selected using the Range bits when PM=1. Also note the MSEL signal. This is a pin on DS5001/DS5002 series devices that tells the processor whether multiple 32K RAMs or a 128K RAM is being used. When MSEL=0, a single 128K device is used. It is not possible to partition the device when MSEL=0, and the state of the partition bits will be ignored. The four selections are as follows. The non–partitionable memory map is shown in Figure 4–6. Byte–wide bus segments begin at 0000h.

MSEL	RG1	RG0	PROGRAM	DATA	PROGRAM ACCESS	DATA ACCESS
1	0	0	32K	64K	1 @ 32K, $\overline{CE1}$	2 @ 32K, $\overline{CE3}$ and $\overline{CE4}$
1	0	1	64K	32K	2 @ 32K, $\overline{CE1}$ and $\overline{CE2}$	1 @ 32K, $\overline{CE3}$
1	1	0	64K	64K	2 @ 32K, $\overline{CE1}$ and $\overline{CE2}$	2 @ 32K, $\overline{CE3}$ and $\overline{CE4}$
0	1	1	64K	64K	1 @ 128K X 8, for both program and data	

Any address that does not fall into the Byte–wide bus area is routed to the Expanded bus of Ports 0 and 2. This could only occur for the first two settings. Note that a single 128K device is the least expensive in terms of component cost and size. In this case, all memory addressable by the DS5001 is stored in a nonvolatile 128K x 8 SRAM. When the MSEL pin is grounded, the device automatically converts $\overline{CE1}$ to a chip enable, $\overline{CE2}$ to A16, $\overline{CE3}$ to A15, and $\overline{CE4}$ is unused. The MSL bit, accessible only via the bootstrap loader, is used to select whether the the 64KB data or 64KB program segment is addressed by the loader.

NON–PARTITIONABLE MEMORY MAP FOR DS5001, DS5002 SERIES Figure 4–6

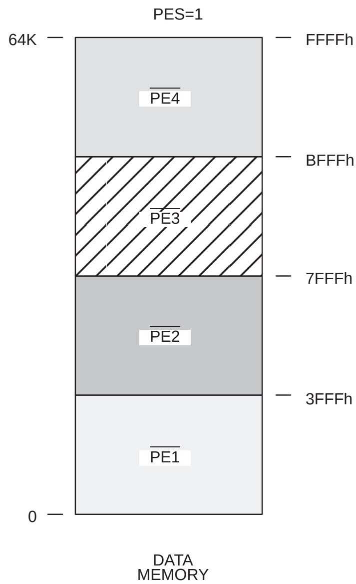


DS5001/DS5002 Memory Mapped Peripherals

The DS5001 series provides four decoded chip enables that can be used for peripheral access or extra RAM on the Byte–wide bus. Application software enables the four $\overline{PE}$ signals, which are decoded on 16K byte boundaries. While they are enabled, they completely use the data memory map and normal data memory is not available on either the Byte–wide or Expanded bus. The PES bit (MCON.2) is set to a logic 1 to access the peripheral space. When PES=1, the appropriate $\overline{PE}$ signal will be activated based on the logical address. Figure 4–7 shows the data memory map while PES=1. PES has an identical effect for either Partitionable or Non–partitionable modes. It has no effect on the program area. Note that the first two Peripheral Enables, $\overline{PE1}$ and $\overline{PE2}$ are lithium backed by the DS5001. This means that when V_{CC} is removed, the device will maintain these chip enables in a logic high, inactive state. $\overline{PE3}$ and $\overline{PE4}$ are not lithium backed making them suitable for UARTs, A/Ds, etc. Lithium backed chip enables are used to access lithium backed memory or peripherals, including the DS1283 real–time clock used in the DS2251T and DS2252T.

On occasion, a memory mapped peripheral is needed that interfaces directly to an 8051 multiplexed bus. When this occurs, MOVX instructions can be forced to use the Expanded bus in any mode with the EXBS bit (RPCTL.5). Setting this bit to a logic one forces all MOVX instructions to the Expanded bus. While EXBS=1, the entire 64K data memory map is accessed in this way. Clearing EXBS will cause the microcontroller to revert to its selected configuration. In most systems, the EXBS bit will not be used.

PERIPHERAL ENABLES IN THE DATA MEMORY MAP Figure 4–7



DS5001/DS5002 Memory Map Control

Like the DS5000, the DS5001/DS5002 uses Special Function Registers to control the memory map. The memory control functions include the Partition, Range, Partition Mode (PM), Expanded Bus Select (EXBS), Peripheral Enable Select (PES) and Access Enable (AE – discussed below). The Partition and Range can be selected using the Bootstrap Loader discussed in a later section. In addition, the Partition can be selected or modified by the application software by writing to the MCON register. PES is normally used by software and is also controlled by the MCON register. The MCON is documented in the SFR summary, but also appears here for convenience. The Range is controlled by a combination of MCON and RPCTL bits. In addition, the EXBS and AE are controlled using the RTPCL register. As not all of the RPCTL bits pertain to memory control, the relevant bits are described below. RPCTL is fully documented in the SFR summary.

DS5001/DS5002 SERIES MCON REGISTER Figure 4–8

PA3	PA2	PA1	PA0	RG1	PES	PM	——

Bit Description:

MCON.7–4: **PA3–0**
Partition Address. When PM=0, this address specifies the boundary between program and data memory in a continuous space.

Initialization: Unaffected by watchdog, external, or power–up resets. Set to 1111B on a No V_{LI} reset.

Read Access: Can be read normally at any time.

Write Access: Timed Access Protected. Also, cannot be written by the application software if set to 0000B by the serial loader. If a 0000B is written via the serial loader and the security lock is set, the Partition will become 1111B. The same will occur if write access is available and application software writes a 0000B. In addition, these bits will be set to 1111B if security lock is cleared.

MCON.3: **RG1**
One of two bits that determine the range of program space. RG0 is located in the RPCTL register.

Initialization: Unaffected by watchdog, external, or power–up resets. Set to 1 on a No V_{LI} reset or a clearing of the security lock.

Read Access: Can be read at any time.

Write Access: Cannot be modified by the application software. Can only be written during program load.

MCON.2: **PES**
Peripheral Enable Select. When this bit is set, the data space is controlled by $\overline{PE1}$ – $\overline{PE4}$. Peripherals are memory–mapped in 16K blocks, and are accessed by MOVX instructions.

Initialization: Cleared by all resets.

Read Access: Can be read at any time.

Write Access: Can be written at any time.

MCON.1: **PM**
Partition Mode. When PM=0, a partitionable, continuous memory map is invoked. When PM=1, one of four fixed allocations is used.

Initialization: Unaffected by watchdog, external, or power–up reset. Cleared on a No V_{LI} reset.

Read Access: Can be read at any time.

Write Access: Cannot be written by the application software. Can only be modified during program load.

DS5001/DS5002 SERIES RPCTL REGISTER BITS AFFECTING MEMORY Figure 4–9

RNR	——	EXBS	AE	IBI	DMA	RPCON	RG0

Bit Description:

RPCTL.5: **EXBS**
The Expanded Bus Select routes data memory access (MOVX) to the Expanded bus formed by ports 0 and 2 when set.

Initialization: Cleared after all resets.

Read Access: Can be read at any time.

Write Access: Can be written at any time.

RPCTL.4: **AE**
Access Enable is used when a software reload is desired without using Program Load mode. When set, the DS5001 will be temporarily configured in a Partitionable configuration with the partition at 4K. This will occur even if the PM=1. When cleared, the prior memory configuration is resumed.

Initialization: Cleared after all resets.

Read Access: Can be read at any time.

Write Access: Can be written at any time, timed access protected.

RPCTL.0: **RG0**
This is a Range bit which is used to determine the size of the program memory space. Its usage is shown above.

Initialization: Unaffected by watchdog, external, or power–up resets. Cleared on a No V_{LI} reset or clearing of the security lock.

Read Access: Can be read at any time.

Write Access: Cannot be modified by the application software. Can only be written during Program Load.

Loading and Reloading Program Memory

Soft Microcontrollers are programmed through a built–in Bootstrap Loader function. This loader is also used to configure the desired options for memory map control. The Secure Microcontroller uses its low power lithium backed circuits to maintain critical settings in the absence of power. For this reason, it is not necessary to set the Partition, Range, etc. after every power–up or reset. Once set, they will remain unless deliberately modified. Bootstrap Loading is discussed in a later section. One of the major advantages of a Secure Microcontroller is the ability to change these settings, and even reload the entire program memory while the device is installed in system. To completely re–program and re–configure a device, the Bootstrap Loader must be invoked. However, the Secure Microcontroller is designed to allow a partial reload of memory without invoking the Bootstrap Loader.

The major advantage of this technique is that it requires no hardware or external switches. Most of the memory can be reprogrammed under application software control. It would commonly be used when the target system connects to a PC through a serial port as part of an application. For example, a data logger that must dump memory periodically. While connected to the PC, it is extremely easy to reload portions of memory using the "Soft Reload".

Application software always has unrestricted read/write access to the nonvolatile RAM designated as data memory. This is the memory that lies above the Partition address and below the Range address (the non–partitionable configuration of the DS5001 will be addressed separately). Data memory is read or written using the MOVX instruction. Only the area designated as program memory can not be altered. The key to doing a "Soft Reload" is to temporarily change the program memory RAM into data memory. Using an SFR, the application software can authorize the Secure Microcontroller to temporarily redefine a portion of the program memory area as data memory. Once this is done, the new code can be received through a serial port (or other means) and written into data memory. When the process is complete and the new memory is verified as correct, software converts the RAM back into write–protected program memory for the duration. As with the memory map control, there are minor differences between the DS5000 series and DS5001/DS5002 series devices in how this is accomplished. Each is described below.

SOFT RELOAD OF A DS5000 SERIES DEVICE

When application software decides that it should reprogram a portion of memory, the software must convert the target area into data memory. The DS5000 will do this when software sets the PAA bit (MCON.1) to a logic 1. PAA is the Partition Access Enable. Setting PAA has two effects. The microcontroller will automatically move the Partition to 0800h and allow write access to the Partition control bits PA3–0 (MCON.7–4). At this time, the software can adjust the Partition, but the new value will not be used until after PAA is cleared. The Partition remains at 0800h as long as PAA=1, regardless of the Partition control bits. This leaves a 2K block of NV RAM (from 0000–0800h) assigned as program memory. Apart from this, no other changes take place and software continues to operate normally. Caution, make certain that the code that controls the PAA resides in this first 2K. When PAA=1, all addresses on the Byte–wide bus greater than 0800h will be viewed as data memory and can not be executed even if they were program memory originally. This gives the software read/write access to the remaining 6K bytes (Range=8K) or 30K bytes (Range=32K) of NV RAM on the Byte–wide bus.

At this time, software can begin reloading the target area of memory. There are two minor variations of this procedure. First, a user's loader routine that resides below 0800h (2K) can reprogram the remainder of memory as needed. This is done be receiving the new code through a serial port or other mechanism and writing it to the RAM at the addresses where it will be executed. Since the RAM is data memory, the write operation is done using MOVX instructions.

The second option is that the user's code below 2K can simply move the Partition to a new value. This is done by writing a new value for PA3–0 in MCON (MCON.7–4) while PAA is still set to a 1, then clearing PAA. The purpose of this would be that the loader routine mentioned in option 1 resides in memory above 2K, but below the target memory area. To gain access, the Partition must be moved to a location that includes this loader routine. Once the Partition is moved to this temporary location, the software loader can reprogram new code as before.

When loading is complete, the Partition must be either restored or set to a new value that is appropriate for the new software. If the PA3–0 bits were not modified, then the PAA bit can simply be cleared. This will cause the old Partition to be restored. If the PAA3–0 were modified during loading or software has grown significantly, then a new Partition is needed. The PA3–0 bits must be written while PAA is set to a 1.

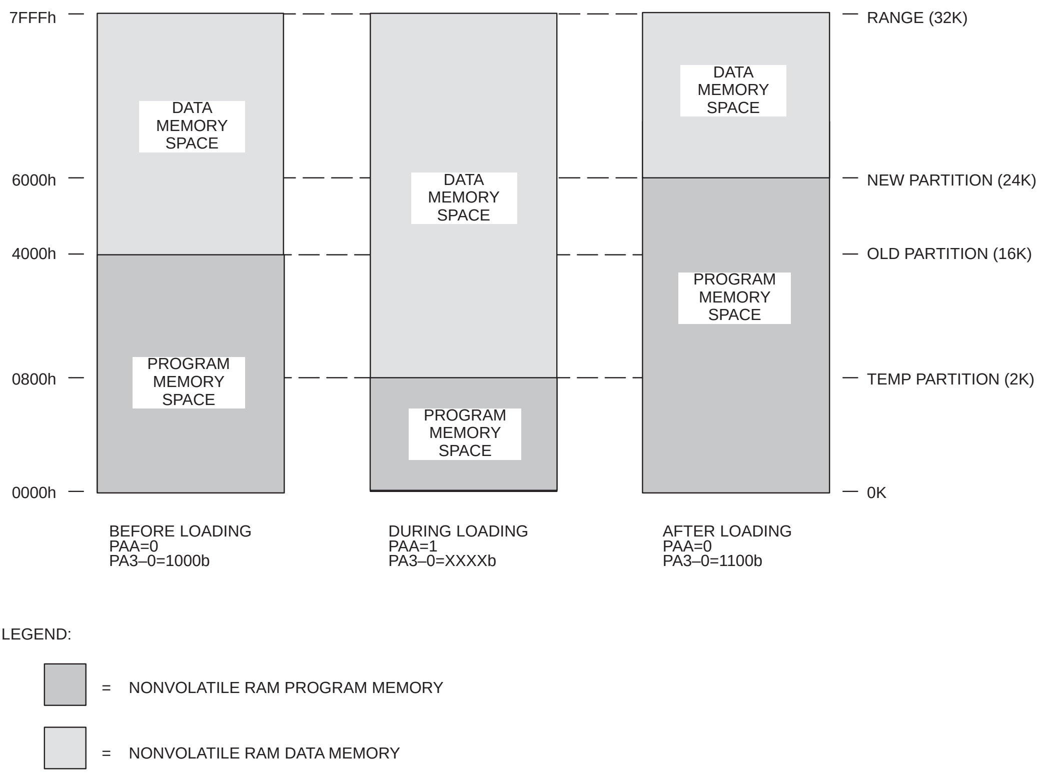
The DS5000FP protects the PAA bit from accidental modification by requiring a Timed Access procedure. Timed Access is designed to prevent an out–of–control program from modifying the PAA bit and crashing the application. Timed Access is discussed in a later section. To summarize the "Soft Reload", the procedure goes as follows:

1. Ensure that current program execution is in the range of 0000h to 0800h.
2. Set the Partition Address Access (PAA) bit using a Timed Access Procedure.
3. Load new contents into program memory at addresses above 0800h using MOVX instructions.
4. Define a new Partition address if necessary and write the appropriate bits into PA3–0 in the MCON SFR.
5. Restore the current Partition by clearing the PAA bit with a Timed Access procedure.
6. Resume operation.

The following illustrates the Soft Reload procedure. The original program requires a partition of 4000h (16K bytes). The new program is larger, requiring a Partition of 6000h (24K bytes). The code that performs these steps is shown below. This routine must be located below 0800h in program memory.

```
MOV     TA, #0AAh                 ; TIMED ACCESS
MOV     TA, #55h                  ; TIMED ACCESS 2
MOV     MCON,  #10001010b         ; SET PAA BIT
.                                 ; USER'S CODE TO LOAD
.                                 ;  RAM USING MOVX
.
.
MOV     TA, #0AAh                 ; TIMED ACCESS
MOV     TA, #55h                  ; TIMED ACCESS 2
MOV     MCON, #11001000b ; LOAD NEW PARTITION AND CLEAR PAA BIT
```

RELOADING PORTIONS OF A DS5000 SERIES DEVICE Figure 4–10



SOFT RELOAD OF A DS5001/DS5002

When application software decides that it should reprogram a portion of memory, the software must convert the target area into data memory. However, a Soft Reload of a DS5001 series device has minor variations from the DS5000 version. First, there is no PAA bit in the DS5001. If the DS5001 is in a Partitionable mode then the user's program must manipulate the Partition control bits PA3–0, placing the Partition to a value that permits the target area to be loaded. Moving the Partition to a new value should convert the target area to data memory allowing read/write access. The user's loader routine then uses MOVX instructions to load the new program contents into memory. This program can be received from a serial port or other mechanism. When the loading procedure is complete, a new Partition (or the old one) must be loaded. Note that the loader routine must reside below the Partition at all times.

In the DS5000 series, the PAA bit was protected by a Timed Access procedure. In the DS5001, the PA3–0 bits are protected directly. The user's program must use a Timed Access procedure to alter these bits. The microcontroller further protects the application by not permitting software to write a 0000b into PA3–0. This would cause a program memory area of 0K. Timed Access is discussed in a later section.

If the device is in a non–partitionable configuration, then an extra step is required. To perform a Soft Reload of the program contents in a non–partitionable mode, the software must convert the micro to a Partitionable mode temporarily. The Access Enable bit (RPCTL.4) will accomplish this. Setting the AE bit to a logic 1 converts the DS5001 into a Partitionable mode for as long as it is set. This means that regardless of the original setting, once AE=1, the memory map is a 64K partitionable mode. The Partition is set to 1000h (4K) when AE=1, so the loader routine must reside in this area. The user can then perform the Soft Reload as discussed above. When loading is complete, the software should clear the AE bit. Note that AE requires software to use a Timed Access procedure to alter it. This method allows a user to alter program memory in a non–partitionable mode. Data memory can be initialized by application software at any time. Since full read/write access is available, no special provisions are needed.

To summarize the "Soft Reload" for a DS5001/DS5002, the procedure goes as follows:

Partitionable mode

1. Write a value to PA3–0 using a Timed Access that gives access to the target area of memory.
2. Load new contents into program memory at addresses above the Partition using MOVX instructions.
3. Define a new Partition address if necessary and write the appropriate bits into PA3–0 in the MCON SFR using a Timed Access.
4. Resume operation.

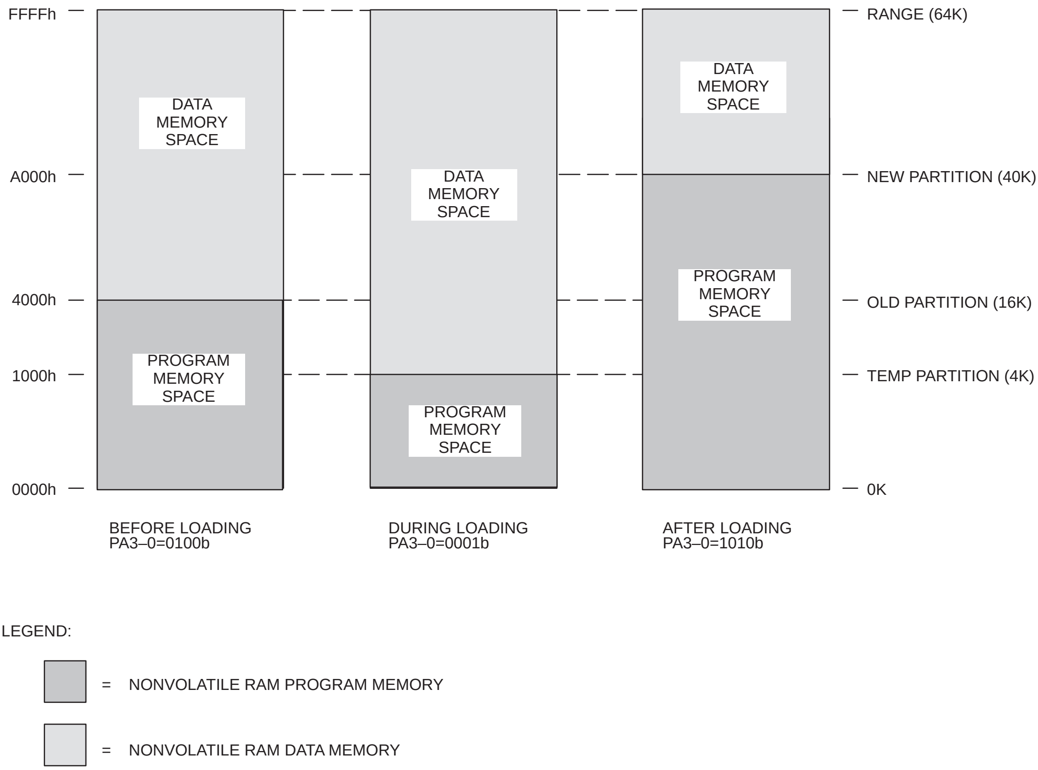
Non–Partitionable mode

1. Set the AE bit to a 1 using a Timed Access procedure.
2. Load new contents into program memory at addresses above the Partition (4K) using MOVX instructions.
3. Clear the AE bit using a Timed Access procedure.
4. Resume operation.

The following illustrates an example where a Soft Reload is performed for a Partitionable mode. The original program requires a partition of 4000h (16K bytes). The new program is larger, requiring a Partition of A000h (40K bytes). A loader routine resides below address 1000h. The code that performs these steps is shown below. Note that the Timed Access procedure is performed, but is described in a later section.

```
MOV    TA, #0AAh                        ; TIMED ACCESS
MOV    TA, #55h          ; TIMED ACCESS 2
MOV    MCON,  #00011000b ; SET PARTITION TO 1000h
|                        ; USER'S CODE TO LOAD
|                        ; RAM USING MOVX
|
|
MOV    TA, #0AAh                        ; TIMED ACCESS
MOV    TA, #55h          ; TIMED ACCESS 2
MOV    MCON, #10101000b  ; LOAD NEW PARTITION OF A000h
```

RELOADING A DS5001/DS5002 SERIES DEVICE Figure 4–11



Special Function Registers

The Secure Microcontroller uses Special Function Registers (SFRs) to control most functions. In many cases, an SFR will contain 8 bits, each of which control a function or report status on a function. The SFRs reside in register locations 80–FFh. They can be accessed using MOV instructions with direct addressing. In addition, some of the SFRs are bit addressable. This can be particularly useful when enabling a function without modifying others in the register since an SFR can contain 8 unrelated control and status functions.

With a few minor exceptions documented below, the Secure Microcontroller provides identical SFRs to a standard 8051, plus extra locations to control unique functions. Modifications to the standard 8051 SFR map are as follows. The PCON register GF1 (PCON.3) and GF0 (PCON.2) have been replaced by the Enable Power Fail Interrupt and the Enable Watchdog Timer bits respectively. In addition, the Secure Microcontroller requires a Timed Access procedure before allowing software to modify the STOP mode bit (PCON.1). This is to prevent errant software from creating a situation that the Watchdog Timer can not recover from. The remaining SFRs are either identical to the 8051 or new to the architecture.

As with the memory map, there are some differences between the DS5000 series and the DS5001 series SFRs. Figures 4–12 and 4–13 show an overview of their respective SFR maps. Following these figures are detailed descriptions. In the case where a particular SFR has differences between the DS5000 and DS5001/DS5002, those differences will be pointed out under the particular register. In some cases, the DS5001 and DS5002 have registers that do not appear in the DS5000. This is also highlighted under the particular register.

DS5000 SERIES SPECIAL FUNCTION REGISTER MAP Figure 4–12

DIRECT BYTE ADDRESS	(MSB)			BIT ADDRESS				(LSB)	SPECIAL FUNCTION REGISTER SYMBOL
0F0H	F7	F6	F5	F4	F3	F2	F1	F0	B
0E0H	E7	E6	E5	E4	E3	E2	E1	E0	ACC
	C	AC	F0	RS1	RS0	OV		P	
0D0H	D7	D6	D5	D4	D3	D2	D1	D0	PSW
0C7H	NOT BIT ADDRESSABLE								TA
	PA3	*PA2*	*PA1*	*PA0*	*RA32/8*	*ECE2*	PAA	*SL*	
0C6H	NOT BIT ADDRESSABLE								MCON
	RWT			PS	PT1	PX1	PT0	PX0	
0B8H	BF	–	–	BC	BB	BA	B9	B8	IP
0B0H	B7	B6	B5	B4	B3	B2	B1	B0	P3
	EA			ES	ET1	EX1	ET0	EX0	
0A8H	AF	–	–	AC	AB	AA	A9	A8	IE
0A0H	A7	A6	A5	A4	A3	A2	A1	A0	P2
99H	NOT BIT ADDRESSABLE								SBUF
	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	
98H	9F	9E	9D	9C	9B	9A	99	98	SCON
90H	97	96	95	94	93	92	91	90	P1
8DH	NOT BIT ADDRESSABLE								TH1
8CH	NOT BIT ADDRESSABLE								TH0
8BH	NOT BIT ADDRESSABLE								TL1
8AH	NOT BIT ADDRESSABLE								TL0
	GATE	$C/\overline{T}$	M1	M0	GATE	$C/\overline{T}$	M1	M0	
89H	NOT BIT ADDRESSABLE								TMOD
	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	
88H	8F	8E	8D	8C	8B	8A	89	88	TCON
	SMOD	$\overline{POR}$	PFW	*WTR*	EPFW	*EWT*	STOP	IDL	
87H	NOT BIT ADDRESSABLE								PCON
83H	NOT BIT ADDRESSABLE								DPH
82H	NOT BIT ADDRESSABLE								DPL
81H	NOT BIT ADDRESSABLE								SP
80H	87	86	85	84	83	82	81	80	P0

* BITS IN ITALICS ARE NONVOLATILE

DS5001/DS5002 SERIES SPECIAL FUNCTION REGISTER MAP Figure 4–13

DIRECT BYTE ADDRESS	(MSB)			BIT ADDRESS				(LSB)	SPECIAL FUNCTION REGISTER SYMBOL
0F0H	F7	F6	F5	F4	F3	F2	F1	F0	B
0E0H	E7	E6	E5	E4	E3	E2	E1	E0	ACC
	ST7	ST6	ST5	ST4	IA0	F0	IBF	0BF	
0DAH	NOT BIT ADDRESSABLE								RPS
	RNR	——	EXBS	AE	IBI	DMA	RPC	*RG0*	
0D8H	DF	DE	DD	DC	DB	DA	D9	D8	RPCTL
	C	AC	F0	RS1	RS0	OV		P	
0D0H	D7	D6	D5	D4	D3	D2	D1	D0	PSW
0CFH	NOT BIT ADDRESSABLE								RNR
0C7H	NOT BIT ADDRESSABLE								TA
	PA3	*PA2*	*PA1*	*PA0*	*RG1*	PES	*PM*	*SL*	
0C6H	NOT BIT ADDRESSABLE								MCON
0C3H	NOT BIT ADDRESSABLE								CRC HIGH
0C2H	NOT BIT ADDRESSABLE								CRC LOW
	RNGE3	RNGE2	RNGE1	RNGE0	——	——	*MDM*	*CRC*	
0C1H	NOT BIT ADDRESSABLE								CRC
	RWT			PS	PT1	PX1	PT0	PX0	
0B8H	BF	——	——	BC	BB	BA	B9	B8	IP
0B0H	B7	B6	B5	B4	B3	B2	B1	B0	P3
	EA			ES	ET1	EX1	ET0	EX0	
0A8H	AF	——	——	AC	AB	AA	A9	A8	IE
0A0H	A7	A6	A5	A4	A3	A2	A1	A0	P2
99H	NOT BIT ADDRESSABLE								SBUF
	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	
98H	9F	9E	9D	9C	9B	9A	99	98	SCON
90H	97	96	95	94	93	92	91	90	P1
8DH	NOT BIT ADDRESSABLE								TH1
8CH	NOT BIT ADDRESSABLE								TH0
8BH	NOT BIT ADDRESSABLE								TL1
8AH	NOT BIT ADDRESSABLE								TL0
	GATE	$C/\overline{T}$	M1	M0	GATE	$C/\overline{T}$	M1	M0	
89H	NOT BIT ADDRESSABLE								TMOD
	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	
88H	8F	8E	8D	8C	8B	8A	89	88	TCON
	SMOD	$\overline{POR}$	PFW	*WTR*	EPFW	*EWT*	STOP	IDL	
87H	NOT BIT ADDRESSABLE								PCON
83H	NOT BIT ADDRESSABLE								DPH
82H	NOT BIT ADDRESSABLE								DPL
81H	NOT BIT ADDRESSABLE								SP
	D7	D6	D5	D4	D3	D2	D1	D0	
80H	87	86	85	84	83	82	81	80	P0/DBB

* BITS IN ITALICS ARE NONVOLATILE

POWER CONTROL REGISTER

Label: PCON **Register Address: 087H**

D7	D6	D5	D4	D3	D2	D1	D0
SMOD	$\overline{\text{POR}}$	PFW	WTR	EPFW	EWT	STOP	IDL

Bit Description:

PCON.7 **SMOD**

"Double Baud Rate": When set to a 1, the baud rate is doubled when the serial port is being used in modes 1, 2, or 3.

Initialization: Cleared to a 0 on any reset.

Read Access: Can be read normally at any time.

Write Access: Can be written normally at any time.

PCON.6 **$\overline{\text{POR}}$**

"Power On Reset": Indicates that the previous reset was initiated during a Power On sequence.

Initialization: Cleared to a 0 when Power On Reset occurs. Remains at 0 until it is set to a 1 by software.

Read Access: Can be read normally at any time.

Write Access: Can be written only by using the Timed Access Register.

PCON.5: **PFW**

"Power Fail Warning": Indicates that a potential power failure is in progress. Set to 1 whenever V_{CC} voltage is below the V_{PFW} threshold. Cleared to a 0 immediately following a read operation of the PCON register. Once set, it will remain set until the read operation occurs regardless of activity on V_{CC}. After PFW is cleared by a read, it will return to a 1 if $V_{CC} < V_{PFW}$.

Initialization: Cleared to a 0 during a Power On Reset.

Read Access: Can be read normally anytime.

Write Access: Not writable.

PCON.4: **WTR**

"Watchdog Timer Reset": Set to a 1 following a Watchdog Timer timeout. If Watchdog Timer Reset is enabled, this will indicate the cause of the reset. Cleared to 0 immediately following a read of the PCON register.

Initialization: Set to a 1 after a Watchdog Timeout Reset. Cleared to a 0 on a Power On Reset. Remains unchanged during other types of resets.

Read Access: May be read normally anytime.

Write Access: Cannot by written.

PCON.3: **EPFW**

"Enable Power Fail Interrupt": Used to enable or disable the Power Fail Interrupt. When EPFW is set to a 1, it will be enabled; it will be disabled when EPFW is cleared to a 0.

Initialization: Cleared to a 0 on any type of reset.

Read Access: Can be read normally anytime.

Write Access: Can be written normally anytime.

PCON.2: **EWT**

"Enable Watchdog Timer": Used to enabled or disable the Watchdog Timeout Reset. The Watchdog Timer is enabled if EWT is set to a 1 and will be disabled if EWT is cleared to a 0.

Initialization: Cleared to a 0 on a No–V_{LI} Power on Reset. Remains unchanged during other types of reset.

Read Access: May be read normally anytime.

Write Access: Can be written only by using the Timed Access register.

PCON.1: **STOP**

"Stop": Used to invoke the Stop mode. When set to a 1, program execution will terminate immediately and Stop mode operation will commence. Cleared to a 0 when program execution resumes following a hardware reset.

Initialization: Cleared to a 0 on any type of reset.

Read Access: Can be read anytime.

Write Access: Can be written only by using the Timed Access register.

PCON.0: **IDL**

"Idle" Used to invoke the Idle mode. When set to a 1, program execution will be halted and will resume when the idle bit is cleared to 0 following an interrupt or a hardware reset.

Initialization: Cleared to 0 on any type of reset or interrupt.

Read Access: Can be read normally anytime.

Write Access: Can be written normally anytime.

TIMER CONTROL REGISTER

Label: TCON **Register Address 088H**

D7	D6	D5	D4	D3	D2	D1	D0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Bit Description:

TCON.7: **TF1**

"Timer 1 Overflow Flag": Status bit set to 1 when Timer 1 overflows from a previous count value of all 1's. Cleared to 0 when CPU vectors to Timer 1 interrupt service routine.

Initialization: Cleared to 0 on any type of reset.

TCON.6: **TR1**

"Timer 1 Run Control": When set to a 1 by software, Timer 1 operation will be enabled.
Timer 1 is disabled when cleared to 0.

Initialization: Cleared to 0 on any type of reset.

TCON.5: **TF0**

"Timer 0 Overflow": Status bit set to 1 when Timer 0 overflows from a previous count value of all 1's. Cleared to 0 when CPU vectors to Timer 0 interrupt service routine.

Initialization: Cleared to 0 on any type of reset.

TCON.4: **TR0**

"Timer 0 Run Control": When set to a 1 by software, Timer 0 operation is enabled. Timer 0 is disabled when cleared to 0.

Initialization: Cleared to 0 on any type of reset.

TCON.3: **IE1**

"Interrupt 1 Edge Detect": Set to 1 to signal when a 1–to–0 transition (IT=1) or a low level (IT=0) has been detected on the $\overline{\text{INT1}}$ pin. Cleared to a 0 by hardware when interrupt processed only if IT1=1.

Initialization: Cleared to 0 on any type of reset.

TCON.2: **IT1**

"Interrupt 1 Type Select": When set to 1, 1–to–0 transitions on $\overline{\text{INT1}}$ will be used to generate interrupt requests from this pin. When cleared to 0, $\overline{\text{INT1}}$ is level-activated.

Initialization: Cleared to a 0 on any type of reset.

TCON.1: **IE0**

"Interrupt 0 Edge Detect": Set to a 1 to signal when a 1–to–0 transition (IT0=1) or a low level (IT0=0) has been detected on the $\overline{\text{INT0}}$ pin. Cleared to a 0 by hardware when interrupt processed only if IT0=1.

Initialization: Cleared to a 0 on any type of reset.

TCON.0: **IT0**

"Interrupt 0 Type Select": When set to 1, 1–to–0 transitions on INT0 will be used to generate interrupt requests from this pin. When cleared to 0, $\overline{\text{INT0}}$ is level–activated.

Initialization: Cleared to a 0 on any type of reset.

TIMER MODE REGISTER

Label: TMOD **Register Address: 089H**

D7	D6	D5	D4	D3	D2	D1	D0
GATE	C/$\overline{\text{T}}$	M1	M0	GATE	C/$\overline{\text{T}}$	M1	M0

Bit Description:

TMOD.7 (Timer 1);
TMOD.3 (Timer 0): **GATE**

"Gate Control": When set to 1 with TRn=1, timer/counter's input count pulses will only be delivered while a 1 is present on the $\overline{\text{INT}}$ pin. When cleared to 0, count pulses will always be received by the timer/counter a long as TRn=1.

Initialization: Cleared to 0 on any reset.

TMOD.6 (Timer 1);
TMOD.2 (Timer 0) **C/$\overline{\text{T}}$**

"Counter/Timer Select": When set to 1, the counter function is selected for the associated timer; when cleared to 0, the timer function is selected.

Initialization: Cleared to 0 on any reset.

TMOD.5, TMOD.4 (Timer 1);
TMOD.1, TMOD.0 (Timer 0): **M1,M0**

"Mode Select": These bits select the operating mode of the associated timer/counter as follows:

M1	M0	
0	0	Mode 0: 8 bits with 5–bit prescale
0	1	Mode 1: 16 bits with no prescale
1	0	Mode 2: 8 bits with auto–reload
1	1	Mode 3: Timer 0 – Two 8–bit timers Timer 1 – Stopped

Initialization: Cleared to 0 on any reset.

SERIAL CONTROL REGISTER

Label:SCON **Register Address: 098H**

D7	D6	D5	D4	D3	D2	D1	D0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Bit Description:

SCON.7, SCON.6: **SM0, SM1**

"Mode Select": Used to select the operational mode of the serial I/O port as follows:

SM0	SM1	MODE	WORD FUNCTION	BAUD LENGTH	CLOCK PERIOD
0	0	Mode 0	SYNC	8–bits	12 t_{CLK}
0	1	Mode 1	ASYNC	10–bits	Timer 1 Overflow
1	0	Mode 2	ASYNC	11–bits	64 t_{CLK} or 32 t_{CLK}
1	1	Mode 3	ASYNC	11–bits	Timer 1 Overflow

Initialization: Cleared to 0 on any type of reset.

SCON.5: **SM2**

"Multiple MCU Comm": Used to enable the multiple microcontroller communications feature for modes 2 and 3. When SM2=1, RI will be activated only when serial words are received which cause RB8 to be set to a 1.

Initialization: Cleared to a 0 on any type of reset.

SCON.4: **REN**

"Receive Enable": When set to 1, the receive shift register will be enabled. Disabled when cleared to 0.

Initialization: Cleared to a 0 on any type of reset.

SCON.3: **TB8**

"Xmit Bit 8": Can be set or cleared to define the state of the 9th data bit in modes 2 and 3 of a serial data word.

Initialization: Cleared to a 0 on any type of reset.

SCON.2: **RB8**

"Rec. Bit 8": Indicates the state of the 9th data bit received while in modes, 2 or 3. If mode 1 is selected with SM2=0, RB8 is the state of the stop bit which was received. RB8 is not used in mode 0.

Initialization: Cleared to a 0 on any type of reset.

SCON.1: **TI**

"Xmit Interrupt": Status bit used to signal that a data word has been completely shifted out. In mode 0, it is set at the end of the 8th data bit. Set when the stop bit is transmitted in all other modes.

Initialization: Cleared to a 0 on any type of reset.

SCON.0: **RI**

"Receive Interrupt": Status bit used to signal that a serial data word has been received and loaded into the receive buffer register. In mode 0, it is set at the end of the 8th bit time. It is set at the mid–bit time of the incoming stop bit in all other modes of a valid received word according to the state of SM2.

INTERRUPT ENABLE REGISTER

Label:IE **Register Address: 0A8H**

D7	D6	D5	D4	D3	D2	D1	D0
EA	–	–	ES	ET1	EX1	ET0	EX0

Bit Description:

IE.7: **EA**

"Enable All Interrupts": When set to 1, each interrupt except for PFW may be individually enabled or disabled by setting or clearing the associated IE.x bit. When cleared to 0, interrupts are globally disabled and no pending interrupt request will be acknowledged except for PFW.

IE.4: **ES**

"Enable Serial Interrupt": When set to 1, an interrupt request from either the serial port's TI or RI flags can be acknowledged. Serial I/O interrupts are disabled when cleared to 0.

IE.3: **ET1**

"Enable Timer 1 Interrupt": When set to 1, an interrupt request from Timer 1's TF1 flag can be acknowledged. Interrupts are disabled from this source when cleared to 0.

IE.2: **EX1**

"Enable External Interrupt 1": When set to 1, an interrupt request from the IE1 flag can be acknowledged. Interrupts are disabled from this source when cleared to 0.

IE.1: **ET0**

"Enable Timer 0 Interrupt": When set to 1, an interrupt request from Timer 0's TF0 flag can be acknowledged. Interrupts are disabled from this source when cleared to 0.

IE.0: **EX0**

"Enable External Interrupt 0": When set to 1, an interrupt from the IE0 flag can be acknowledged. Interrupts are disabled from this source when cleared to 0.

INTERRUPT PRIORITY REGISTER

Label:IP **Register Address: 0B8H**

D7	D6	D5	D4	D3	D2	D1	D0
RWT	–	–	PS	PT1	PX1	PT0	PX0

Bit Description:

IP.7: **RWT**

"Reset Watchdog Timer": When set to a 1, the Watchdog Timer count will be reset and counting will begin again. The RWT bit will then automatically be cleared again to 0. Writing a 0 into this bit has no effect.

Initialization: Cleared to a 0 on any reset.

Read Access: Cannot be read.

Write Access: Can be written only by using the Timed Access register.

All of the following bits are read/write at any time and are cleared to 0 following any hardware reset.

IP.4: **PS**

"Serial Port Priority": Programs Serial Port interrupts for high priority when set to 1. Low priority is selected when cleared to 0.

IP.3: **PT1**

"Timer 1 Priority": Programs Timer 1 interrupt for high priority when set to 1. Low priority is selected when cleared to 0.

IP.2: **PX1**

"Ext. Int. 1 Priority": Programs External Interrupt 1 for high priority when set to 1. Low priority is selected when cleared to 0.

IP.1: **PT0**

"Timer 0 Priority": Programs Timer 0 Interrupt for high priority when set to 1. Low priority is selected when cleared to 0.

IP.0: **PX0**

"Ext. Int. 0 Priority" Programs External Interrupt 0 for high priority when set to 1. Low priority is selected when cleared to 0.

DS5001 CRC REGISTER

Label: CRC **Register Address: 0C1H**

RNGE3	RNGE2	RNGE1	RNGE0	—	—	MDM	CRC

Bit Description:

CRC.7–4 **RNGE3–0**
Determines the range over which a power–up CRC will be performed. Addresses are specified on 4K boundaries.

Initialization: Reset to 0 on a No V_{LI} reset.

Read Access: Can be read at any time.

Write Access: Cannot be written by the application software. Can only be written via the Bootstrap Loader.

CRC.1 **MDM**
When set to 1, the bootstrap loader will attempt to use a modem (UART) on $\overline{PE4}$ if CRC is incorrect. This feature is no longer useful following the obsoletion of the corresponding modem devices.

Initialization: Reset to 0 on a No V_{LI} reset.

Read Access: Can be read at any time.

Write Access: Cannot be written by the application software. Can only be written during Program Load mode.

CRC.0 **CRC**
When set to 1, a CRC check will be performed on power–up or watchdog timeout. CRC will be checked against stored values. An error will initiate Program Load mode. This bit will not be present in the DS5002FP as the device does not support the power–on CRC function.

Initialization: Reset to 0 on a No V_{LI} reset.

Read Access: Can be read at any time.

Write Access: Cannot be written by the application software. Can only be written during Program Load mode.

DS5000 MEMORY CONTROL REGISTER

Label:MCON **Register Address: 0C6H**

D7	D6	D5	D4	D3	D2	D1	D0
PA3	PA2	PA1	PA0	RA32/8	ECE2	PAA	SL

Bit Description:

MCON.7–4: **PA3–0**

"Partition Address": Used to select the starting address of Data Memory on the Byte–wide bus. Program space lies below the partition address.

PA3	PA2	PA1	PA0	Partition Address
0	0	0	0	0000H
0	0	0	1	0800H
0	0	1	0	1000H
0	0	1	1	1800H
0	1	0	0	2000H
0	1	0	1	2800H
0	1	1	0	3000H
0	1	1	1	3800H
1	0	0	0	4000H
1	0	0	1	4800H
1	0	1	0	5000H
1	0	1	1	5800H
1	1	0	0	6000H
1	1	0	1	6800H
1	1	1	0	7000H*
1	1	1	1	8000H*

*A 4K byte increment (not 2K bytes) in the Partition Address takes place between bit field values 1110B and 111B.

Initialization: Set to all 1's on a No V_{LI} Power On Reset or when the Security Lock bit is cleared to a 0 from previous 1 state. These bits are also set to all 1's when any attempt is made to have them cleared to all 0's with the SL bit set to 1 (illegal condition).

Read Access: May be read anytime.

Write Access: PAA bit must = 1 in order to write PA3–0. Timed Access is not required to write to PA3–0 once PAA=1.

MCON.3: **RA32/8**

"Range Address": Set the maximum usable address in on the Byte–wide bus.
RA32/8=0 sets Range Address = 1FFFH (8K)
RA32/8=1 sets Range Address = 7FFFH (32K)

Initialization: Set to a 1 during a No V_{LI} Power On Reset and when the Security Lock bit (SL) is cleared to a 0 from a previous 1 state. Remains unchanged on all other types of resets.

Read Access: May be read normally anytime.

Write Access: Cannot be modified by the application software; can only be written via the Bootstrap Loader.

MCON.2: **ECE2**

"Enable Chip Enable 2": Used to enable or disable the $\overline{CE2}$ signal for the Byte–wide bus data memory. This bit should always be cleared to 0 in the DS5000, DS5000-32, DS2250–8 and DS2250–32 versions.

Initialization: Cleared to 0 only during a No V_{LI} Power On Reset.

Read Access: Read normally anytime.

Write Access: Can be written normally at any time.

MCON.1: **PAA**

"Partition Address Access": Used to protect the programming of the Partition Address select bits. PA3–0 cannot be written when PAA=0. PAA can be written only via the Timed Access register.

Initialization: PAA is cleared on any reset.

Read Access: PAA may be read anytime.

Write Access: The Timed Access register must be used to perform any type of write operation on the PAA bit.

MCON.0: **SL**

"Security Lock": Indicates that the security lock is set when SL=1.

Initialization: Cleared to a 0 on a no V_{LI} power on reset.

Read Access: Read normally any time.

Write Access: Can only be modified by the Lock and Unlock commands of the Bootstrap loader. This bit cannot be modified by the application software or by the Bootstrap loader Write command.

DS5001 MCON REGISTER

Label: MCON **Register Address: 0C6H**

PA3	PA2	PA1	PA0	RG1	PES	PM	SL

Bit Description:

MCON.7–4: **PA3–0**
Partition Address. When PM=0, this address specifies the boundary between program and data memory in a continuous space.

Initialization: Unaffected by watchdog, external, or power–up resets. Set to 1111B on a No V_{LI} reset.

Read Access: Can be read normally at any time.

Write Access:	Timed Access Protected. Cannot be written by the application software if set to 0000B by the serial loader. If a 0000B is written via the serial loader and the security lock is set, the Partition will become 1111B. The same will occur if write access is available and application software writes a 0000B. In addition, these bits will be set to 1111B if security lock is cleared.
MCON.3:	**RG1** One of two bits that determine the range of program space. RG0 is located in the RPCTL register.
Initialization:	Unaffected by watchdog, external, or power–up resets. Set to 1 on a No V_{LI} reset or a clearing of the security lock.
Read Access:	Can be read at any time.
Write Access:	Cannot be modified by the application software. Can only be written via the Bootstrap Loader.
MCON.2:	**PES** Peripheral Enable Select. When this bit is set, the data space is controlled by $\overline{\text{PE1}}$–$\overline{\text{PE4}}$. Peripherals are memory–mapped in 16K blocks, and are accessed by MOVX instructions on the Byte–wide bus.
Initialization:	Cleared by all resets.
Read Access:	Can be read at any time.
Write Access:	Can be written at any time.
MCON.1:	**PM** Partition Mode. When PM=0, a partitionable, continuous memory map is invoked. When PM=1, one of four fixed allocations is used.
Initialization:	Unaffected by watchdog, external, or power–up resets. Cleared on a No V_{LI} reset.
Read Access:	Can be read at any time.
Write Access:	Cannot be modified by the application software. Can only be modified via the Boostrap Loader.
MCON.0:	**SL**
"Security Lock":	Indicates that the security lock is set when SL=1.
Initialization:	Cleared to a 0 on a no V_{LI} power on reset.
Read Access:	Read normally any time.
Write Access:	Can only be modified by the Lock and Unlock commands of the Bootstrap loader. This bit cannot be modified by the application software or by the Bootstrap loader Write command.

PROGRAM STATUS WORD REGISTER

Label:PSW **Register Address: 0D0H**

D7	D6	D5	D4	D3	D2	D1	D0
C	AC	F0	RS1	RS0	OV		P

All of the bits in PSW except parity are read/write and are cleared to 0 on any type of reset. The Parity bit is read only and is cleared to 0 on any type of reset.

Bit Description:

PSW.7: **C**

"Carry": Set when the previous operation resulted in a carry (during addition) or a borrow (during subtraction). Otherwise cleared.

PSW.6: **AC**

"Auxiliary-Carry": Set when the previous operation resulted in a carry (during addition) or a borrow (during subtraction) from the low–order nibble. Otherwise cleared.

PSW.5: **F0**

"User Flag 0": General–purpose flag bit which can be set or cleared as needed.

PSW.4–3: **R1–R0**

"Register Bank Select": Used to select an 8–byte bank of registers within the Data Register space to be assigned as R0–R8 in subsequent instructions. The 8–byte bank starting address selection is as follows:

R1	R0	Data Register Address (R0)
0	0	00H
0	1	08H
1	0	10H
1	1	18H

PSW.2: **OV**

"Overflow": Set when a carry was generated into the high–order bit but not a carry out of the high–order bit as a result of the previous operation, and visa–versa. OV is normally used in 2's complement arithmetic.

PSW.0: **P**

"Parity": Set if the modulo–2 sum of the eight bits of the accumulator is 1 (odd parity); cleared on even parity.

DS5001/DS5002 RPC CONTROL REGISTER

Label: RPCTL **Register Address: 0D8H**

RNR	——	EXBS	AE	IBI	DMA	RPCON	RG0

Bit Description:

RPCTL.7 **RNR**
When internal hardware sets this read–only bit to a 1, a new value may be read from the random number generator register of the DS5001/DS5002 (RNR;0CFh). This bit is cleared when the random number is read, and approximately 160 µs are required to generated the next number.

Initialization: Cleared after all resets. Bit will be set approximately 160 µsec after a reset.

Read Access: Can be read at any time.

Write Access: Cannot be written.

RPCTL.5 **EXBS**
The Expanded Bus Select routes data memory access (MOVX) to the expanded bus formed by ports 0 and 2 when set.

Initialization: Cleared after all resets.

Read Access: Can be read at any time.

Write Access: Can be written at any time.

RPCTL.4 **AE**
Access Enable is used when a software reload is desired without using the Bootstrap Loader. When set, the device will be temporarily configured in a Partitionable configuration with the Partition at 4K. This will occur even if the PM=1. When cleared, the prior memory configuration is resumed.

Initialization: Cleared after all resets.

Read Access: Can be read at any time.

Write Access: Can be written at any time, Timed Access protected.

RPCTL.3 **IBI**
When using the RPC mode, an interrupt may be required for the Input Buffer Flag. This interrupt is enabled by setting the Input Buffer Interrupt (IBI) bit. At this time, the timer 1 interrupt is disabled, and this RPC mode interrupt is used in its place (vector location 1BH). This bit can be set only when the RPCON bit is set.

Initialization: Cleared on all resets, and when the RPCON bit is cleared.

Read Access: Can be read at any time.

Write Access: Can be written when the RPC mode is enabled (RPCON=1).

RPCTL.2 **DMA**
This bit is set to enable DMA transfers when RPC mode is invoked. It can only be set when RPCON=1.

Initialization: Cleared on all resets, and when RPC is cleared.

Read Access:	Can be read anytime.
Write Access:	Can be written when the RPC mode is enabled (RPCON=1).

RPCTL.1 **RPCON**
Enable the RPC 8042 I/O protocol. When set, port 0 becomes the data bus, and port 2 becomes the control signals.

Initialization:	Cleared on all resets.
Read Access:	Can be read at any time.
Write Access:	Can be written at any time.

RPCTL.0 **RG0**
This is a Range bit which is used to determine the size of the program memory space. Its usage is shown above.

Initialization:	Unaffected by watchdog, external, or power–up resets. Cleared on a No V_{LI} reset or clearing of the security lock.
Read Access:	Can be read at any time.
Write Access:	Cannot be modified by the application software. Can only be modified via the Bootstrap loader.

DS5001/DS5002 RPC STATUS REGISTER

Label: RPS **Register Address: 0DAH**

ST7	ST6	ST5	ST4	IA0	F0	IBF	OBF

Bit Description:

RPS.7–4: General purpose status bits that can be written by the microcontroller and can be read by the external host.

Initialization:	Cleared when RPCON=0.
Read Access:	Can be read by DS5001/DS5002 and host CPU when RPC mode is invoked.
Write Access:	Can be written by the DS5001/DS5002 when RPC mode is invoked.

RPS.3: **IA0**
Stores the value of the external system A0 for the last DBBIN Write when a valid write occurs (as determined by the IBF flag).

Initialization:	Cleared when RPC=0.
Read Access:	Can be read by DS5001/DS5002 and host CPU when in RPC mode.
Write Access:	Automatically written when a valid DBBIN Write occurs. Cannot be written otherwise.

RPS.2: **F0**
General purpose flag written by the DS5001/DS5002 and read by the external host.

Initialization:	Cleared when RPC=0.

Read Access:	Can be read by DS5001/DS5002 and host CPU when in RPC mode.
Write Access:	Can be written by the DS5001/DS5002 when in RPC mode.
RPS.1:	**IBF** Input Buffer Full Flag is set following a write by the external host, and is cleared following a read of the DBBIN by the DS5001/DS5002.
Initialization:	Cleared when RPC=0.
Read Access:	Can be read by DS5001/DS5002 and host CPU when in RPC mode.
Write Access:	Written automatically as part of the RPC communication. Cannot be set by the application software.
RPS.0:	**OBF** Output Buffer Full Flag is set following a write of the DBBOUT by the DS5001/DS5002, and is cleared following a read of the DBBOUT by the external host.
Initialization:	Cleared when RPC=0.
Read Access:	Can be read by DS5001/DS5002 and host CPU when in RPC mode.
Write Access:	Written automatically as part of the RPC communication. Cannot be set by the application software.

INSTRUCTION SET

Introduction

The Secure Microcontroller executes an instruction set which is object code compatible with the industry standard 8051 microcontroller. As a result, software tools written for the 8051 are compatible with the Secure Microcontroller, including cross–assemblers, compilers, and debugging tools.

There are a total of 42 instruction types recognized by the Secure Microcontroller. When the instruction uses both source and destination operands, they are specified in the order of "destination, source".

Addressing Modes

There are eight addressing modes. Five of these are used to address operands. The other three are used in instructions which transfer execution of the program to another address (e.g., Branch, Jump, Call).

The modes which address source operands, include Register Addressing, Direct Addressing, Register–Indirect Addressing, Immediate Addressing and Register-Indirect with Displacement. The first three of these can also be used to address a destination operand. Most instructions use operands that are located in the Internal Data Registers.

The addressing modes used for the Control Transfer instructions include Relative Addressing, Page Addressing, and Extended Addressing.

The operation of these addressing modes is summarized below, followed by an example.

<u>Register Addressing</u>

Register Addressing is used on operands contained in one of the eight registers (R7–R0) of the currently selected Working Register Bank. A register bank is selected via a 2–bit field in the PSW Special Function register. All of the Working registers may also be accessed through either Direct Addressing or Register-Indirect Addressing as well. This is due to the fact that the Working registers are mapped into the lower 32 bytes of Internal Data RAM as discussed above.

```
ADD     A, R4        ; Add Accumulator to Working
                     ; register R4
```

<u>Direct Addressing</u>

Direct Addressing is the only mode available for use on operands within the Special Function registers. Addressing of bytes may also be used to access the 128 Internal Data registers.

```
MOV 072H, 074H    ; Load direct register (addr. 072H)
                  ; with direct register (074H)
```

Direct addressing of bits is available on 128 bits located in the Internal Data registers in byte addresses of 20H – 2FH inclusive. Direct bit addressing is also available in Special Function registers located at addresses on 8–byte boundaries starting at 80H (i.e., 80H, 88H, 90H, 98H, ...0F0H, 0F8H).

```
SETB    00H          ; Set addressable bit 00H (D0 in
                     ; Internal Data Reg. 20H)
```

<u>Register Indirect Addressing</u>

Some instructions use Register–Indirect Addressing for accessing operands in other Internal Data registers. This is done by using the contents of Working register R1 or R0 as a pointer to other Internal Data registers.

```
ANL     A, @R0       ; Logical AND of Accumulator with
                     ; Internal Data register; pointed to
                     ; by contents of R0
```

In addition, this addressing is used via the Stack Pointer register (SP) for manipulation of the stack. The stack area is contained in the Internal Data Register area. The PUSH and POP instructions are the only ones which use SP for this addressing mode.

```
PUSH    P0           ; Save the contents of the Port 0
                     ; SFR latch on the stack
```

The R0, R1, and the DPTR registers are used with Register–Indirect Addressing for accessing Data Memory. R1 or R0 in the selected Working Register bank may be used for accessing location within a 256–byte block pointed to by the current contents of the P2 SFR latch (address high byte).

```
MOVX    A, @R1       ; Load the Accumulator with the
                     ; contents of Data Memory
                     ; addressed by the 8–bit contents
                     ; of R1
```

The 16–bit DPTR register may be used to access any Data Memory location within the 64K byte space.

```
MOVX  @DPTR,A   ; Load the Data Memory location
                ; pointed to by the contents of the
                ; DPTR register with the contents
                ; of the Accumulator.
```

Immediate Addressing

Immediate Addressing is used to access constants for use as operands which are contained in the current instruction in Program Memory.

```
ORL   A, #040H  ; Logical OR of the Accumulator
                ; with the constant value of 040H
```

Register–Indirect with Displacement

Register–Indirect with Displacement Addressing is used to access data in look–up tables in Program Memory space. The location accessed is pointed to by the contents of either the DPTR or the PC registers, which are used as a base register added together with the contents of the Accumulator (A), which is used as an index register.

```
MOVC  A, @DPTR+A   ; Load the Accumulator with
                   ; the contents of the Program
                   ; Memory location pointed to
                   ; by the value of the DPTR
                   ; register plus the value
                   ; contained in the Accumulator
```

Relative Addressing

Relative Addressing is used in the determination of a destination address for the Conditional Branch instructions. Each of these instructions includes an 8–bit byte which contains a 2's complement address offset (–127 to +128) which is added to the PC to determine the destination address which will be branched to when the tested condition is found to be true. The PC points to the Program Memory location immediately after the Branch instruction when the offset is added. If the condition is found to be not true, then program execution continues from the address of the following instruction.

```
JZ    –20       ; Branch to the location (PC+2) –
20              ; if the contents of the Accumulator
                ; = 0
```

Page Addressing

Page Addressing is used by the Control Transfer instructions to specify a destination address within the 2K byte block in which the next contiguous instruction resides. The full 16–bit address is calculated by taking the highest-order five bits for the next contiguous instruction (PC+2) and concatenating them with the lowest-order 11–bit filed contained in the current instruction. 11–bit field provides an efficient instruction encoding of a destination address for these instructions.

```
0830  ACALL   100H   ; Call to the subroutine at
                     ; address 0100H + current
                     ; page address
```

In this case the destination address would be 800H + 100H or 900H.

Extended Addressing

Extended Addressing is used in the Control Transfer Instructions to specify a 16–bit destination address within the entire 64K byte addressable range of the Secure Microcontroller.

```
LJMP  0FF80H    ; Jump to address 0FF80H
```

Program Status Flags

All of the Program Status flags are contained in the PSW register. Instructions which affect the states of the flags are summarized below.

INSTRUCTIONS THAT AFFECT FLAG SETTINGS

	FLAGS				FLAGS		
INSTRUCTION	**C**	**OV**	**AC**	**INSTRUCTION**	**C**	**OV**	**AC**
ADD	↕	↕	↕	CLR C	0		
ADDC	↕	↕	↕	CPL C	↕		
SUBB	↕	↕	↕	ANL C, bit	↕		
MUL	0	↕		ANL C, $\overline{\text{bit}}$	↕		
DIV	0	↕		ORL C, bit	↕		
DA	↕			ORL C, $\overline{\text{bit}}$	↕		
RRC	↕			MOV C, bit	↕		
RLC	↕			CJNE	↕		
SETB C	1						

LEGEND:
0 = Cleared to 0
1 = Set to a 1
↕ = Modified according to the result of the operation.