



---

---

## Section 45. Data Memory with Extended Data Space (EDS)

---

---

### HIGHLIGHTS

This section of the manual contains the following topics:

45.1	Introduction .....	45-2
45.2	Data Memory Organization .....	45-3
45.3	Extended Data Space .....	45-7
45.4	Data Alignment .....	45-14
45.5	Software Stack .....	45-15
45.6	Interfacing Program and Data Memory Spaces .....	45-15
45.7	Related Application Notes.....	45-16
45.8	Revision History .....	45-17

## 45.1 INTRODUCTION

To cater to the need of large data memory for some applications, like USB and Graphics, the data address space of some of the PIC24F microcontrollers (MCUs) are extended. These PIC24F MCUs with Extended Data Space (EDS) can access up to 16 Mbytes of additional data memory, both internal and external. External memory can be accessed through the Enhanced Parallel Master Port (EPMP). Just like other Harvard architecture devices, PIC24F MCUs also feature separate program and data memory spaces and buses. The PIC24F architecture also allows the direct access of program memory from the data space during code execution.

# Section 45. Data Memory with Extended Data Space (EDS)

## 45.2 DATA MEMORY ORGANIZATION

### 45.2.1 Data Address Space

The PIC24F core has a 16-bit wide data memory space, addressable as a single linear range. The data space is accessed using two Address Generation Units (AGUs), one each for read and write operations. The data space memory map is shown in Figure 45-1.

The 16-bit wide data addresses in the data memory space point to bytes within the Data Space (DS). This gives a DS address range of 64 Kbytes or 32K words, including 2 Kbytes of SFR space. The lower 32 Kbytes (0x0000 to 0x7FFF) of the DS are compatible with the PIC24F MCUs without EDS.

The upper 32 Kbytes of data memory address space (0x8000-0xFFFF) are used as an EDS window. The EDS window is used to access all of the memory region that is implemented in EDS, as shown in Figure 45-2.

The EDS includes any additional internal data memory not accessible by the lower 32 Kbytes of data address space and any external memory through Enhanced PMP. In PIC24F MCUs with EDS, the Program Memory (PM) can also be read from EDS. This is called Program Space Visibility (PSV).

The EDS is organized as pages, with a single page called an EDS page that equals the EDS window (32 Kbytes). A particular EDS page is selected through the Data Space Read register (DSRPAG) or Data Space Write register (DSWPAG). For PSV, only the DSRPAG register is used. The combination of DSxPAG register value and the 16-bit wide data address forms a 24-bit Effective Address (EA). See **Section 45.3.1 “Data Access from EDS”** for details on how EAs are generated for EDS address space for accessing internal extended data memory, external data memory and PSV address space for reading data from PM.

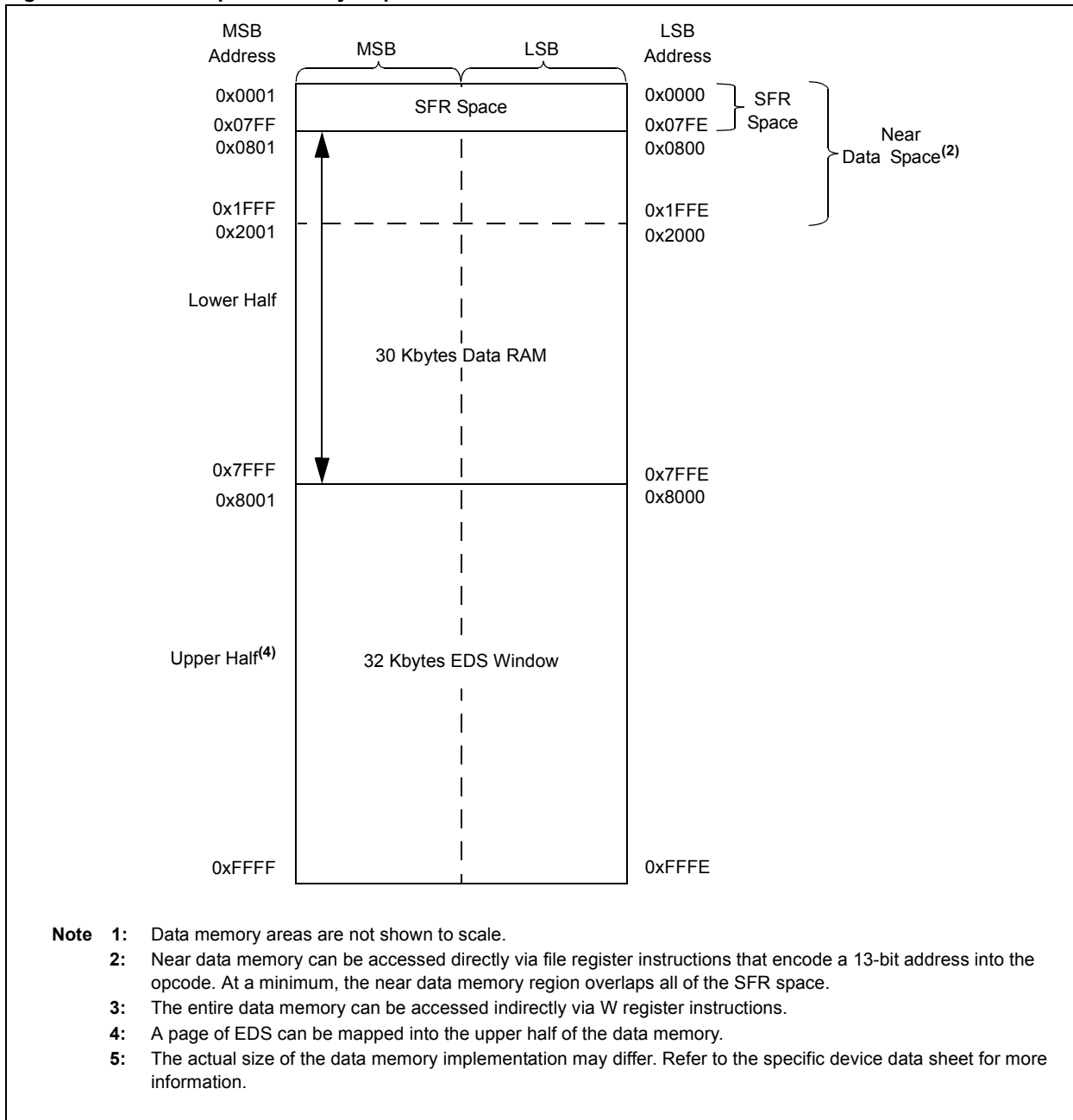
**Note:** Refer to the specific device data sheet for actual implementation of data memory in a specific device.

### 45.2.2 Data Space and Extended Data Space Width

The data memory space is organized in byte-addressable, 16-bit wide blocks. Data is aligned in data memory and registers as 16-bit words, but all the data space and extended data space EAs resolve to bytes. The Least Significant Bytes (LSBs) of each word have even addresses, while the Most Significant Bytes (MSBs) have odd addresses. Figure 45-1 shows the data space memory map.

# PIC24F Family Reference Manual

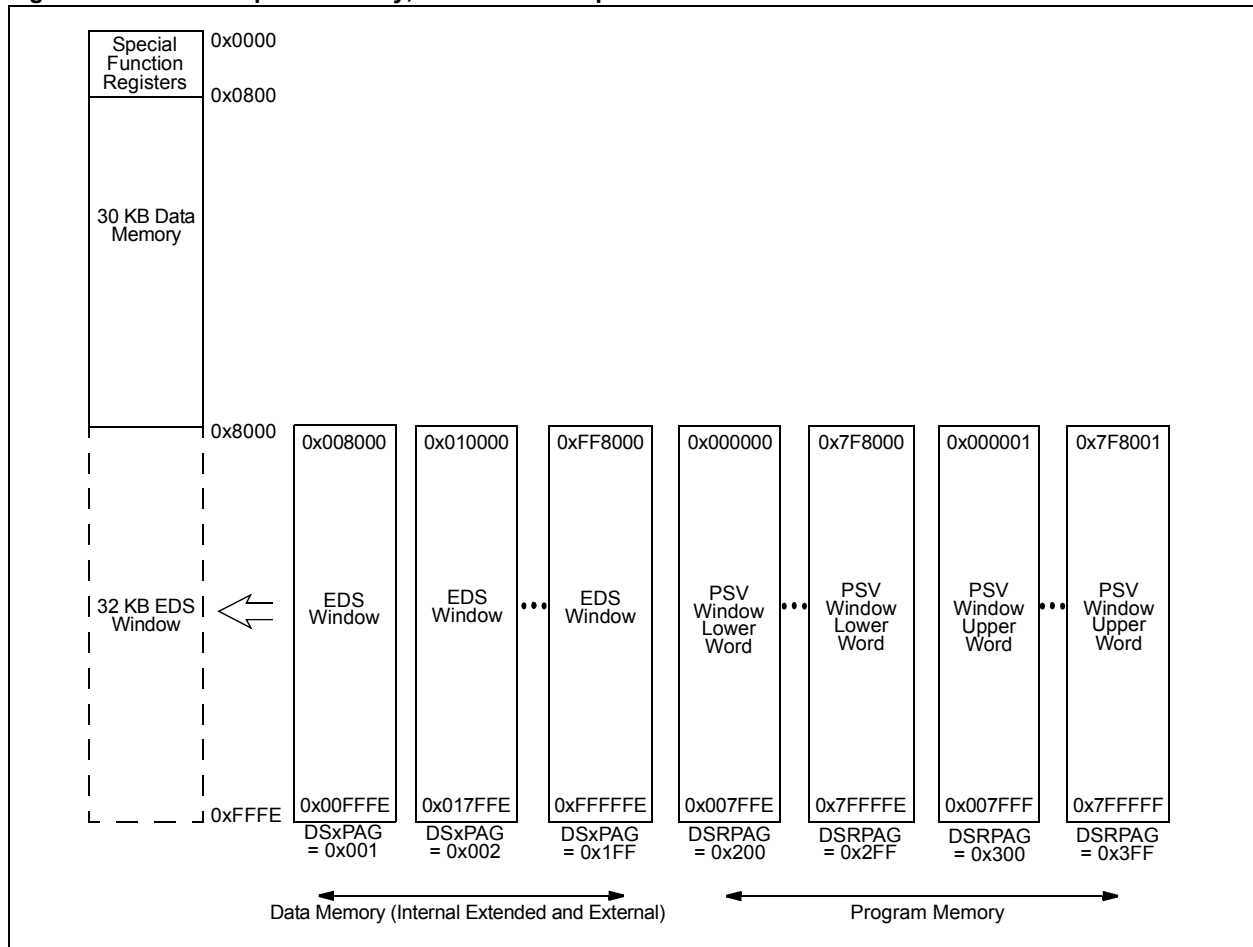
Figure 45-1: Data Space Memory Map for PIC24F Devices<sup>(1,3,5)</sup>



# Section 45. Data Memory with Extended Data Space (EDS)

Figure 45-2 shows the entire DS, EDS and PSV space.

**Figure 45-2: Data Space Memory, EDS and PSV Space for PIC24F Devices**



## 45.2.3 Near Data Memory

An 8-Kbyte address space, between 0x0000 and 0x1FFF, is referred to as near data memory. Near data memory is directly addressable via a 13-bit absolute address field within all the file register instructions.

Near data memory is also addressable by all the Indirect Addressing modes, where the data memory address can be pointed to by any of the 16-bit working registers. The data memory region beyond 0x1FFF (including EDS) is addressable only by the Indirect Addressing modes.

The memory regions included in the near data region will depend on the amount of data memory implemented for each PIC24F family device variant. At a minimum, the near data region will include all of the SFRs. Refer to Figure 45-1 for more details.

# PIC24F Family Reference Manual

## 45.2.4 SFR Space

The first 2 Kbytes of the near data space, from 0x0000 to 0x07FF, are primarily occupied with Special Function Registers (SFRs). These are used by the PIC24F core and peripheral modules for controlling the operation of the device.

SFRs are distributed among the modules that they control and are generally grouped together by module. Much of the SFR space contains unused addresses; these are read as '0'. A diagram of the SFR space that describes where SFRs are actually implemented is shown in Table 45-1. Each implemented area indicates a 32-byte region where at least one address is implemented as an SFR.

**Table 45-1: Implemented Regions of Data Space<sup>(1)</sup>**

SFR Space Address								
	xx00	xx20	xx40	xx60	xx80	xxA0	xxC0	xxE0
0x000	Core			ICN	Interrupts			—
0x100	Timers		Capture	—	Compare	—	—	—
0x200	I <sup>2</sup> C™	UART	SPI		—	—	I/O	
0x300	ADC/CTMU		—	—	—	—	—	—
0x400	—	—	—	—	—	—	—	—
0x500	—	—	—	—	—	—	—	—
0x600	Enhanced PMP	RTC/Comp	CRC	—	PPS			—
0x700	GFX Controller		System	NVM/PMD	—	—	—	—

**Legend:** — = Unimplemented SFRs in this block

**Note 1:** Refer to the specific device data sheet for actual register implementation.

# Section 45. Data Memory with Extended Data Space (EDS)

## 45.3 EXTENDED DATA SPACE

The CPU architecture of PIC24F devices with EDS allows additional data space of up to 16 Mbytes. The EDS is always accessed through the 32-Kbytes EDS window. Hence, a single page in the EDS can have up to 32 Kbytes of memory space. The number of pages that can be addressed depends on read or write operations. An amount of 511 pages (0x001 to 0x1FF) can be accessed while writing and an amount of 1023 pages (0x001 to 0x3FF, including those 511 pages, which can be written as well) can be accessed while reading. While the lower 511 pages (0x001 to 0x1FF), which can be read and written, are for extended data memory, the upper 512 pages (0x200 to 0x3FF) of read-only space are for PSV. Table 45-2 provides the memory map of the entire data memory, including DS, EDS and PSV.

**Table 45-2: Different Address Spaces in PIC24F with EDS**

DSRPAG (Data Space Read register)	DSWPAG (Data Space Write register)	Source/Destination Address while Indirect Addressing	24-Bit EA Pointing to DS, EDS or PSV	Comment	
x <sup>(1)</sup>	x <sup>(1)</sup>	0x0000 to 0x1FFF	0x000000 to 0x001FFF	DS; near data memory <sup>(2)</sup>	
		0x2000 to 0x7FFF	0x002000 to 0x007FFF	DS	
0x001	0x001	0x8000 to 0xFFFF	0x008000 to 0x00FFFE	EDS; 32 Kbytes on each page	
0x002	0x002		0x010000 to 0x017FFE		
•	•		•		
•	•		•		
•	•		•		
0x1FF	0x1FF		0xFF8000 to 0xFFFFFE		
0x200	--- <sup>(3)</sup>		0x000000 to 0x007FFE	0x000000 to 0x007FFE	PSV, lower words of 4M program instructions (8 Mbytes) for read operations only
•			•	•	
•			•	•	
0x2FF			0x7F8000 to 0x7FFFFE	0x7F8000 to 0x7FFFFE	PSV, upper words of 4M program instructions (4 Mbytes remaining, 4 Mbytes are phantom bytes) for read operations only
0x300		0x000001 to 0x007FFF	0x000001 to 0x007FFF		
•		•	•		
•		•	•		
0x3FF		0x7F8001 to 0x7FFFFF	0x7F8001 to 0x7FFFFF		
0x000	0x000	Invalid Address	Invalid Address	Address error trap <sup>(4)</sup>	

**Note 1:** If source/destination address is below 0x8000, the DSRPAG and DSWPAG are ignored.

**2:** This data space can also be accessed by Direct Addressing.

**3:** DSWPAG is 9-bit register, so beyond EDS page, 0x1FF, no write operation is allowed through an EDS mechanism.

**4:** When the source/destination address is above 0x8000, and DSRPAG/DSWPAG is zero, an address error trap will occur.

The amount of physically implemented internal RAM depends on a specific device. For example, PIC24F256DA210 family devices have a total of 96 Kbytes of internal RAM, out of which, 30 Kbytes are physically implemented in the lower 32 Kbytes of DS (where 2 Kbytes are SFR) and an additional 66 Kbytes of RAM are physically implemented in EDS. EDS, page 1 and 2 together, implement 64 Kbytes or 32 Kbytes each, and EDS page 3 implements 2 Kbytes. Enhanced PMP addresses also fall in this EDS. For details on enhanced PMP, refer to “PIC24F Family Reference Manual”, **Section 42. “Enhanced Parallel Master Port (EPMP)”** (DS39730).

**Note 1:** EDS page 0 is not accessible; generates Address Error Trap if accessed.

**2:** There is no auto-increment or decrement from one EDS page to another. DSRPAG and DSWPAG must be loaded with the correct page value when accessing data across multiple EDS pages.

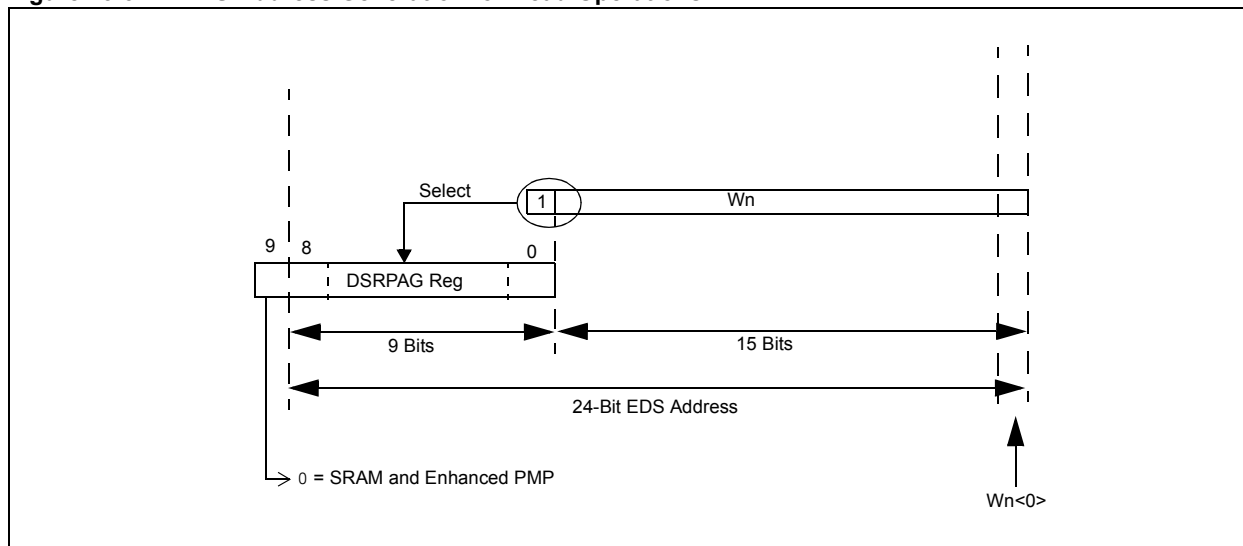
## 45.3.1 Data Access from EDS

The 32-Kbyte EDS window is enabled when bit 15 of the data space address, provided in one of the working registers ( $W_n$ ), is set and the DSRPAG/DSWPAG register holds a valid EDS page address. Each 32-Kbyte EDS window maps to the corresponding address of the selected EDS page. The EDS page can be programmed in the DSRPAG register while reading memory, or in the DSWPAG register while writing to a memory location. In PIC24F MCUs with EDS, the page registers do not update automatically while crossing a page boundary. While developing code in assembly, care must be taken to update page registers while the data memory access crosses the page boundary. The ‘C’ compiler keeps track of the addressing, and increments or decrements the page registers accordingly, while accessing contiguous data memory locations.

### 45.3.1.1 ACCESSING INTERNAL EXTENDED DATA MEMORY

Figure 45-3 and Figure 45-4 illustrate how the EA for the EDS address is generated for data memory read and write operations, respectively.

**Figure 45-3: EDS Address Generation for Read Operations**



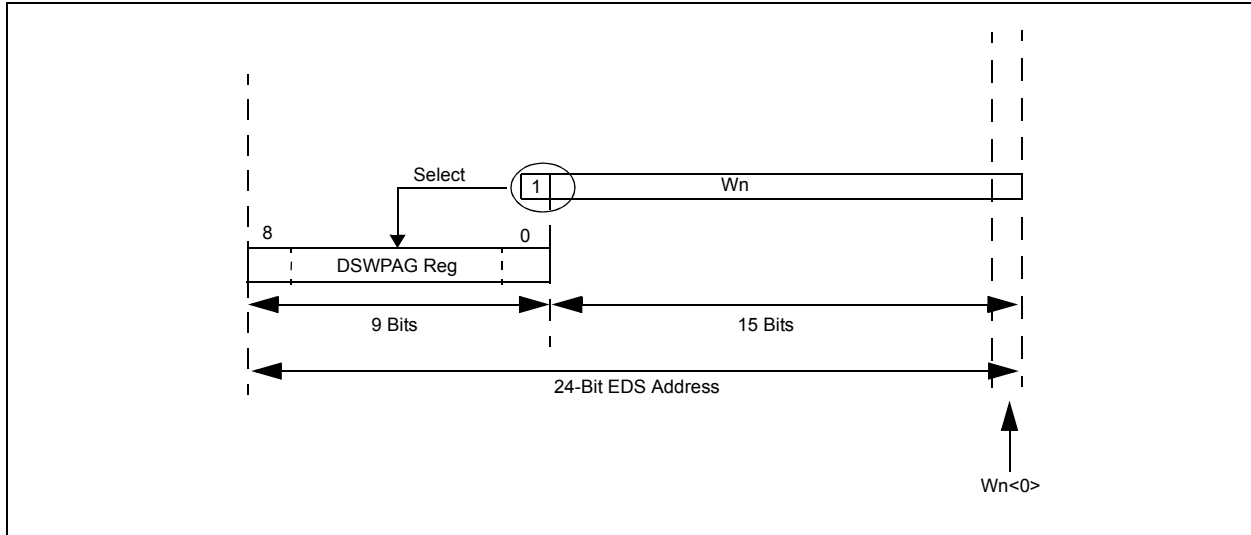
As shown in Figure 45-3, when the Most Significant bit (MSb) of the DS address is ‘1’ and  $DSRPAG\langle 9 \rangle = 0$ , the lower 9 bits of DSRPAG are concatenated to the lower 15 bits of the DS address to form a 24-bit EDS space address for read operations.

**Note:** For byte operations, the Least Significant bit of the DS address,  $W_n\langle 0 \rangle$ , decides which byte is to be accessed. When ‘0’, the LSB is accessed and when it is ‘1’, the MSB is accessed. For word operations,  $W_n\langle 0 \rangle$  has to be maintained as ‘0’ or else an address error will be generated.



# Section 45. Data Memory with Extended Data Space (EDS)

Figure 45-4: EDS Address Generation for Write Operations



As shown in Figure 45-4, when the MSb of the DS address is '1', the lower 9 bits of DSWPAG are concatenated to the lower 15 bits of the DS address to form a 24-bit EDS address for write operations.

## 45.3.1.2 ACCESSING EXTERNAL DATA MEMORY (EPMP)

The extended data space interface also allows the direct addressing of external data memory from the CPU. The accessing of the external data memory is accomplished through EPMP. In this mode of operation, the EPMP module must be enabled (PMPEN = 1) and must be set to Master mode (MODE<1:0> = 11).

When a read or write is performed on the external data memory, all the necessary control signals are generated by the EPMP module. Please refer to "*PIC24F Family Reference Manual*", **Section 42. "Enhanced Parallel Master Port (EPMP)"** for more details.

The recommended PMP configurations to accesses external memory from CPU are:

- For 16-bit port (interface): No address multiplexing, no beginning or middle data wait states and 8/16-bit accesses
- For 8-bit port (interface): no address multiplexing, no beginning or middle data wait states and only 8-bit accesses.

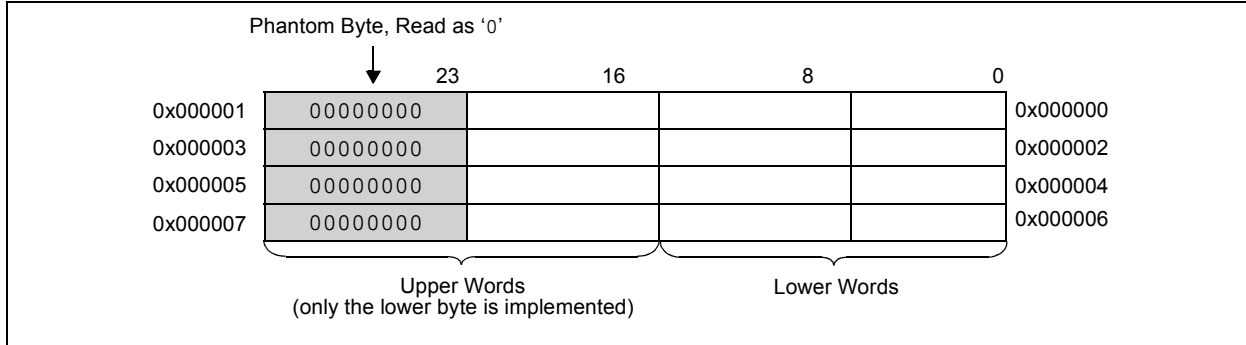
The EA generation for the read and write operations on external memory are illustrated in Figure 45-3 and Figure 45-4, respectively.

- Note 1:** If the external memory space is accessed with the EPMP module disabled or configured in modes other than master, an address error trap will occur.
- 2:** A read from external memory, with no physical memory present, returns garbage data while a write to external memory, without physical memory, will be a dummy write.

## 45.3.1.3 ACCESSING PROGRAM MEMORY (PSV)

Figure 45-5 shows the Program Memory (PM) organization of PIC24F devices. Although the PM space is treated as 24 bits wide, it is more appropriate to consider each address of the program memory as two words, a lower and upper word; the upper byte of the upper word being unimplemented. In PIC24F devices without EDS, the PSV read of the upper word is not supported. PSV read of the upper word of the program memory is supported in PIC24F devices with EDS. The lower byte of the upper word contains the PM data and the upper byte of the upper word always reads zero.

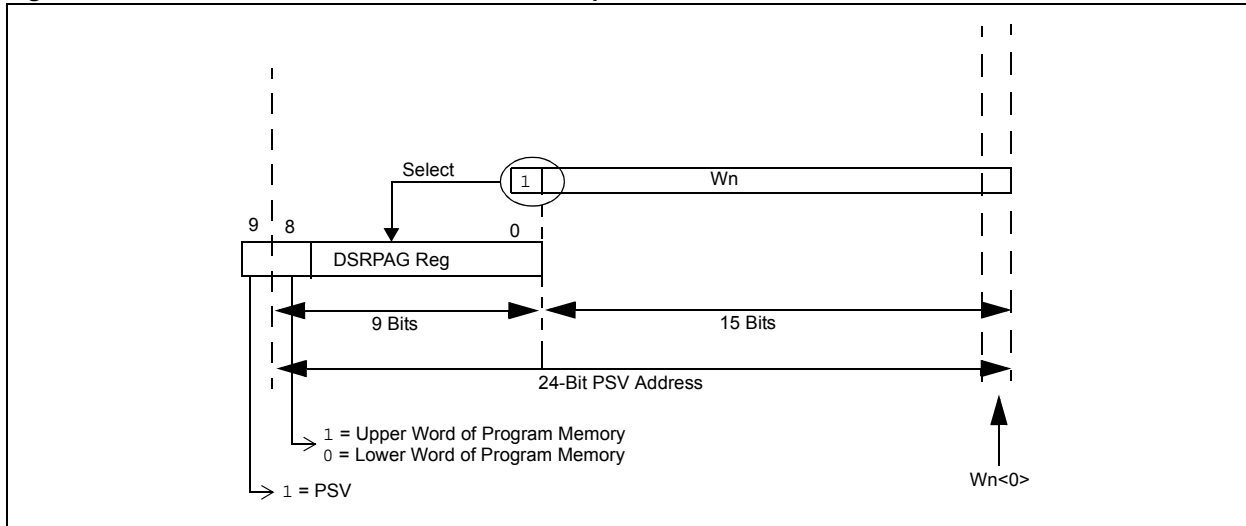
**Figure 45-5: Program Memory Organization**



The entire PM can be accessed through EDS pages, 0x200 to 0x3FF. The EDS pages, 0x200 to 0x2FF, comprise lower words, while the EDS pages, 0x300 to 0x3FF, comprise upper bytes. The data from the desired PM location can be read by selecting the appropriate page. Each page can access 32 Kbytes of data.

Figure 45-6 illustrates how an EA for the PSV address is generated when program memory is accessed for read operations.

**Figure 45-6: PSV Address Generation for Read Operations**



## Section 45. Data Memory with Extended Data Space (EDS)

When the MSb of the DS address is '1' and the DSRPAG<9> is also '1', the 24-bit PSV address is formed by concatenating the lower 15 bits of the DS address and the lower 9 bits of the DSRPAG register. This 24-bit PSV address points to program memory locations. The DSRPAG<8> decides which word is to be addressed; when '0', the lower word is read and when '1', the upper word of the program memory is read.

**Note 1:** Reading the higher byte (phantom byte) of the upper word of program memory returns 0x00.

**2:** PSV is usually the implemented Flash memory in the device; it cannot be written using the EDS mechanism. However, table writes can be used to write data into program memory space.

### 45.3.2 EDS Read

All read operations from EDS space take one extra instruction cycle from those on non-EDS; therefore, a minimum of two instruction cycles are required to complete an EDS read. While EDS reads under a REPEAT instruction, the first two accesses take three cycles each and the subsequent accesses take only one cycle.

EDS reads require two steps. First, an Address Pointer is set up by loading the required EDS page number into the DSRPAG register, and assigning the offset address to one of the W registers. Second, the contents of the pointed EDS location can be read.

Example 45-1 shows how to read a byte, word and double-word from EDS.

#### Example 45-1: EDS Read Code

```
; Set the EDS page from where the data to be read
mov    #0x001 , w0
mov    w0 , DSRPAG    ;page 0x001 is selected for read
mov    #0x800 , w1    ;select the location (0x800) to be read
bset   w1 , #15      ;set the MSB of the base address, enable EDS mode

;Read a byte from the selected location
mov.b  [w1++] , w2    ;read Low byte
mov.b  [w1++] , w3    ;read High byte

;Read a word from the selected location
mov    [w1] , w2      ;

;Read Double - word from the selected location
mov.d  [w1] , w2      ;two words read, stored in w2 and w3
```

Example 45-2 shows how to read a byte, word and double-word from PSV.

## Example 45-2: PSV Read Code

```
;Reading Lower word of program memory
;Set the EDS page from where the data to be read
mov    #0x220 , w0
mov    w0 , DSRPAG    ;page 0x220 is selected for read
mov    #0x0C00 , w1   ;select the location (0x0C00) to be read
bset   w1 , #15      ;set the MSB of the base address, enable EDS mode

;Read a byte from the selected location
mov.b  [w1++] , w2    ;read Low byte
mov.b  [w1++] , w3    ;read High byte

;Read a word from the selected location
mov    [w1] , w2      ;

;Read Double - word from the selected location
mov.d  [w1] , w2      ;two words read, stored in w2 and w3

;Reading Upper word of program memory
;Set the EDS page from where the data to be read
mov    #0x320 , w0
mov    w0 , DSRPAG    ;page 0x320 is selected for read
mov    #0x0C00 , w1   ;select the location (0x0C00) to be read
bset   w1 , #15      ;set the MSB of the base address, enable EDS mode

;Read a byte from the selected location
mov.b  [w1++] , w2    ;read Low byte
mov.b  [w1++] , w3    ;read High byte, always 0x00 (Phantom byte)

;Read a word from the selected location
mov    [w1] , w2      ;

;Read Double - word from the selected location
mov.d  [w1] , w2      ;two words read, stored in w2 and w3
```

# Section 45. Data Memory with Extended Data Space (EDS)

## 45.3.3 EDS Write

All write operations to EDS are executed in a single cycle. Like EDS read, EDS write also requires two steps. First, an Address Pointer is set up by loading the required EDS page number into the DSWPAG register and assigning the memory address to one of the W registers. Second, the addressed location can be written.

Example 45-3 shows how to write a byte, word and double-word to EDS.

### Example 45-3: EDS Write Code

```
; Set the EDS page where the data to be written
mov    #0x002 , w0
mov    w0 , DSWPAG      ;page 0x002 is selected for write
mov    #0x800 , w1      ;select the location (0x800) to be written
bset   w1 , #15        ;set the MSB of the base address, enable EDS
                           ;mode

;Write a byte to the selected location
mov    #0x00A5 , w2
mov    #0x003C , w3
mov.b  w2 , [w1++]      ;write Low byte
mov.b  w3 , [w1++]      ;write High byte

;Write a word to the selected location
mov    #0x1234 , w2    ;
mov    w2 , [w1]      ;

;Write a Double - word to the selected location
mov    #0x2233 , w2
mov    #0x4455 , w3
mov.d  w2 , [w1]      ;2 EDS writes
```

- Note 1:** Use of the REPEAT instruction with read-modify-write operations on EDS is not supported.
- 2:** Use the DSRPAG register while performing read-modify-write operations, such as bit-oriented instructions.

## 45.4 DATA ALIGNMENT

To maintain backward compatibility with PIC<sup>®</sup> MCU devices and to improve the data space memory usage efficiency, the PIC24F instruction set supports both word and byte operations. As a consequence of byte accessibility, all Effective Address calculations are internally scaled to step through word-aligned memory.

The LSb of a 16-bit data address is ignored during word operations. Word data is aligned in the little-endian format with the LSB at the even address (LSb = 0) and the MSB at the odd address (LSb = 1).

For byte operations, the LSb of the data address is used to select the byte that is accessed. Figure 45-7 displays the data alignment for word and byte operations.

**Figure 45-7: Data Alignment**

	15	MSB	8	7	LSB	0	
0x0001	Byte 1			Byte 0			0x0000
0x0003	Byte 3			Byte 2			0x0002
0x0005	Byte 5			Byte 4			0x0004
	Word 0						0x0006
	Word 1						0x0008
	Long Word<15:0>						0x000A
	Long Word<31:16>						0x000C

Data byte reads will read the complete word, which contains the byte using the LSb of any EA to determine which byte to select. The selected byte is placed onto the LSB of the data path. That is, data memory and registers are organized as two parallel, byte-wide entities with shared (word) address decode but separate write lines. Data byte writes only write to the corresponding side of the array or register which matches the byte address.

All Effective Address calculations are automatically adjusted depending on whether a byte or a word access is performed. For example, an address will be incremented by two for a word operation that post-increments the Address Pointer. Similarly, the address will be incremented by one for a byte operation that post-increments the Address Pointer.

All word accesses must be aligned to an even address. Misaligned word data fetches are not supported; therefore, care must be taken when mixing byte and word operations or translating from 8-bit MCU code. If a misaligned read or write is attempted, an address error trap will be generated. If the error occurred on a read, the instruction underway is completed; if it occurred on a write, the instruction will be executed but the write will not occur. In either case, a trap is then executed, allowing the system and/or user to examine the machine state prior to execution of the address Fault. For additional details regarding the interrupts, refer to the “PIC24F Family Reference Manual”, **Section 8. “Interrupts”**.

All byte loads into any W register are loaded into the LSB; the MSB is not modified.

A sign-extend instruction (SE) is provided to allow users to translate 8-bit signed data to 16-bit signed values. Alternatively, for 8-bit unsigned data, users can clear the MSB of any W register by executing a zero-extend (ZE) instruction on the appropriate address.

Although most instructions are capable of operating on word or byte data sizes, it should be noted that some instructions operate only on words.

# Section 45. Data Memory with Extended Data Space (EDS)

---

## 45.5 SOFTWARE STACK

For stack operations, some portion of the data memory of the PIC24F devices needs to be allocated as stack. For additional details on the software stack, refer to “*PIC24F Family Reference Manual*”, **Section 2.3 “Software Stack Pointer”**.

## 45.6 INTERFACING PROGRAM AND DATA MEMORY SPACES

The PIC24F architecture uses a 24-bit wide program space and 16-bit wide data space. The architecture is also a modified Harvard scheme; the program space data can be accessed as if it were present in data space. To use this data successfully, it must be accessed in a way that preserves the alignment of information in both spaces.

Aside from normal execution, the PIC24F architecture provides two methods by which program space can be accessed during operation:

- Using table instructions to access individual bytes or words anywhere in the program space
- Remapping a portion of the program space into the data space using the DSRPAG register

Table instructions allow an application to read or write to small areas of the program memory. This makes the method ideal for accessing data tables that need to be updated from time to time. It also allows access to all bytes of the program word. The remapping method allows an application to access a large block of data on a read-only basis, which is ideal for look-ups from a large table of static data.

For additional details regarding the program and data memory interface, please refer to “*PIC24F Family Reference Manual*”, **Section 4.3 “Data Access from Program Memory”**.

## 45.7 RELATED APPLICATION NOTES

This section lists application notes that are related to this section of the manual. These application notes may not be written specifically for the PIC24F device family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the Data Memory with Extended Data Space (EDS) are:

Title	Application Note #
No related application notes at this time.	

**Note:** Visit the Microchip web site ([www.microchip.com](http://www.microchip.com)) for additional application notes and code examples for the PIC24F family of devices.



# Section 45. Data Memory with Extended Data Space (EDS)

---

## 45.8 REVISION HISTORY

### Revision A (December 2009)

This is the initial released version of this document.

NOTES: