APPLICATION NOTE 3563

Using Data Pointers to Read/Write to SRAM

Jun 20, 2005

Abstract: The MAXQ-based microcontroller uses data pointers to read and write to SRAM. This application note describes how to move data from program memory to the SRAM, and how to access the data from SRAM using the data pointers.

Introduction

This application note describes how to write two words of data to SRAM, using data pointers DP[0] and BP[OFFS]. The data is then read back into the MAXQ's accumulators from SRAM using the same data pointers, and a logical "AND" is performed on the two registers.

Getting Started

To begin, you need basic knowledge about the MAXQ architecture, register map, and instruction set, which can be obtained from the MAXQ Family User's Guide or from any MAXQ-based microcontroller data sheet. A good example is the MAXQ2000 data sheet. You also need to reference "Accessing Data Memory" which is in the MAXQ Family User's Guide. Basic familiarity with assembly language in general, and with MAXQ assembler in particular, is assumed.

Data Pointer Overview

There are three data pointers available on a MAXQ-based microcontroller: DP[0], DP[1], and BP[OFFS]. Each of these data pointers can be configured to either word- or byte-access modes by setting the corresponding bit in the Data Point Control (DPC) register.

<table>
<thead>
<tr>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>DP[0]</td>
<td>Data pointer 0</td>
</tr>
<tr>
<td>DP[1]</td>
<td>Data pointer 1</td>
</tr>
<tr>
<td>BP[OFFS]</td>
<td>Frame pointer base</td>
</tr>
<tr>
<td>OFFS</td>
<td>Offset of frame pointer base</td>
</tr>
<tr>
<td>DPC</td>
<td>Data pointer control</td>
</tr>
<tr>
<td>SDPS0</td>
<td>00b selects DP[0] as active pointer</td>
</tr>
<tr>
<td>SDPS1</td>
<td>01b selects DP[1] as active pointer</td>
</tr>
<tr>
<td>WBS0</td>
<td>DP[0] word mode WBS0=1, byte mode WBS0=0</td>
</tr>
<tr>
<td>WBS1</td>
<td>DP[1] word mode WBS1=1, byte mode WBS0=0</td>
</tr>
</tbody>
</table>
Register | Bit Position
--- | ---
DP[0] | DP[0] (16 bits)
BP | BP = BP[Offs] (16 bits)
OFFS | OFFS (8 Bits)
DPC | DPC (16 bits)

The three pointers share a single read/write port on the data memory, and thus the user must knowingly activate the desired pointer before accessing memory. This can be done explicitly using the data select bits (SDPS2:0; DPC.1:0), or implicitly by writing to the DP[n], BP, or OFFS register as shown below. Any indirect memory access using a data pointer will also set the SDPS bits, thus activating the pointer as the active source pointer.

```
move DPC, #2          ;(explicit) selection of BP as the active pointer
move DP[1], DP[1]               ;(implicit) selection of DP[1]; set SDPS1:0=01b
move OFFS, src         ;(implicit) selection of BP; set SDPS1=1
move @DP[0], src         ;(implicit) selection of DP[0]; set SDPS1:0=00b
```

Increment/Decrement Data Pointer

Data pointers can be updated using pre- and post-increment/decrement operator with a register or a virtual NUL destination. Data pointer increment/decrement operation can be done as follows:

```
move NUL, @DP[0]++              ;increment DP[0]
move NUL, @DP[1]--              ;decrement DP[1]
move NUL, @BP[OFFS++]     ;increment Frame Pointer Base + Offset + 1
move @++DP[0], A[1]             ;increment address and store A[1] at new address
;Note: Only the pre-increment/decrement can be ;used ;when writing to memory
move A[1], @DP[1]--             ;Reads value from DP[1] and store in A[1]
then                              ;decrement
;Note: Only the post-increment/decrement can be ;used
;when reading from memory
```

Code Example

1. Move 2 words to SRAM using DP[0]. 5555h is moved to the first word of SRAM, referenced by int_Var1; AAAAh is moved to the second word of SRAM, referenced by int_Var2. Var_1 and Var_2 are read back from SRAM into accumulator A[0] and A[1].

```
int_Var1 EQU 0h         ;address of int_Var1 to the first byte of SRAM
int_Var2 EQU 1h         ;address of int_Var2 to the second byte of SRAM
main:
```
move DPC, #4h ;Set DP[0] to word mode
          ;Must be set active before using
move DP[0], #int_Var1 ;Load address of int_Var1 into DP[0]
move @DP[0], #5555h ;Also activated DP[0]
move DP[0], #int_Var2 ;Load address of int_Var2 into DP[0]
move @DP[0], #0AAAAh ;Writes AAAAh to SRAM at address of int_Var2
move DP[0], #int_Var1 ;Load address of int_Var1 into DP[0]
move A[0], @DP[0] ;Reads from address of int_Var1
move DP[0], #int_Var2 ;Load address of int_Var2 into DP[0]
move A[1], @DP[0] ;Reads from address of int_Var2
move AP, #0 ;Select accumulator 0
end

2. Move 2 bytes to SRAM using DP[1]. 55h is moved to the first byte of SRAM, referenced by int_VAR1; AAh is moved to the second byte of SRAM, referenced by int_VAR2. Var_1 and Var_2 are read back from SRAM into accumulator A[0] and A[1].

int_Var1 EQU 0h ;address of int_Var1 to the first word of SRAM
int_Var2 EQU 1h ;address of int_Var2 to the second word of SRAM
main:
move DPC, #0h ;Set DP[1] to byte mode
          ;Must be set active before using
move DP[1], #int_Var1 ;Load address of int_Var1 into DP[1]
move @DP[1], #55h ;Also activated DP[1]
move DP[1], #int_Var2 ;Load address of int_Var2 into DP[1]
move @DP[1], #0AAAh ;Writes AAAAh to SRAM at address of int_Var2
move DP[1], #int_Var1 ;Load address of int_Var1 into DP[1]
move A[0], @DP[1] ;Reads from address of int_Var1
move DP[1], #int_Var2 ;Load address of int_Var2 into DP[1]
move A[1], @DP[1] ;Reads from address of int_Var2
move AP, #0 ;Select accumulator 0
end

3. Move 2 words to SRAM using BP[Offs]. 5555h is moved to the first word of SRAM, referenced by int_VAR1; AAAAh is moved to the second word of SRAM, referenced by int_VAR2. Var_1 and Var_2 are read back from SRAM into accumulator A[0] and A[1].

int_Var1 EQU 0h ;address of int_Var1 to the first word of SRAM
int_Var2 EQU 1h ;address of int_Var2 to the second word of SRAM
main:
move DPC, #10h ;Set BP[OFFS] to word mode
move BP, #0h ;Sets BP to 0 and activates BP[OFFS]
move OFFS, #int_Var1          ;Load address of int_Var1 into OFFS
move @BP[OFFS], #5555h      ;write to 5555h to SRAM at address of int_Var1
move OFFS, #int_Var2          ;Load address of int_Var1 into OFFS
move @BP[OFFS], #0AAAAh       ;Writes AAAAh to SRAM at address of int_Var2
move OFFS, #int_Var1          ;Load address of int_Var1 into OFFS
move A[0], @BP[OFFS]          ;Reads from address of int_Var1
move OFFS, #int_Var2          ;Load address of int_Var2 into OFFS
move A[1], @BP[OFFS]          ;Reads from address of int_Var2
move AP,   #0              ;select accumulator 0
end

Invalid Data Pointer Instruction

The following are incorrect uses of data pointers:

<table>
<thead>
<tr>
<th>Assembly Instruction</th>
<th>Problem</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>move DP[0], DP[0]</td>
<td>Data pointer initialized to DP[0]. Needs to be reinitialized to DP[1].</td>
<td>move DP[0], DP[0]</td>
</tr>
<tr>
<td>move A[1], @DP[0]</td>
<td></td>
<td>move A[1], @DP[0]</td>
</tr>
<tr>
<td>move A[2], @DP[1]</td>
<td></td>
<td>move A[2], @DP[1]</td>
</tr>
</tbody>
</table>

| move BP[OFFS], #0h   | BP and OFFS must be initialized separately. | move BP, #0h |
| move A[0], @DP[0]++  | Only the pre-increment/decrement can be used. | move A[0], @DP[0]++ |
| move A[0], @++DP[0]  | Only the post-increment/decrement can be used. | move A[0], @++DP[0] |
| move @++DP[0], @DP[0]++ | Unable to increment data pointer and store it using the same data pointer. | move @++DP[0], @DP[0]++ |
| move A[0], @--DP[0]  | You cannot use pre-increment/decrement for a source. | move A[0], @--DP[0] |

Relevant Links

- MAXQ Home Page
- MAXQ Family User's Guide
- Dallas Semiconductor Microcontroller Support Forum

Related Parts