Implementing Long Calls

INTRODUCTION

This application note discusses how to implement “long Calls” in the PIC16C5X architecture. The use of long call can simplify the partitioning of the application program with minimal software overhead.

In the PIC16C5X architecture, the program memory page size is 512 words. Depending on the device, the program memory may be as large as 2K words (as in the PIC16C57 or PIC16C58 devices). The program counter (and stack) width range from 9- to 11-bits, depending on the amount of program memory the device has. Table 1 shows the width of the Program Counter (PC) and Stack for the various devices.

TABLE 1: PC AND STACK WIDTH

Device	Width (Bits) Program Counter	Width (Bits) Stack	Program Memory (Words)
PIC16C54 / PIC16C55	9	9	512
PIC16C56	10	10	1K
PIC16C57 / PIC16C58	11	11	2K

The low order 8-bits of the program counter are accessible by the user program. These bits are contained in the PC register. The entire Program Counter is shown in Figure 1.

Since A8 is forced to 0 by `CALL` instructions, the start address of subroutines must be in the first 256 words of each program memory page. Depending on the size and number of called subroutines, this limitation may become a burden to the software developer. The implementation of a “long call” eases this, by allowing the subroutine to be anywhere in the program memory page. The three important concepts, to understand the implementation of the long call are:

1. A `CALL` instruction loads the entire PC onto the Stack
2. A `GOTO` instruction does not affect the Stack
3. A `GOTO` instruction can branch to any location in a program memory page.

Also to select the desired page, the RP1 and RP0 bits (STATUS<6:5>) must be programmed accordingly. These bits do not get loaded into A10:A9 , of the PC, until one of the following occurs:

1. A `CALL` instruction
2. A `GOTO` instruction
3. An instruction that modifies the PC register (PC<:7:0>), such as `ADDWF PC, F`.

So a `CALL` instruction followed by a `GOTO` instruction will always remain in the same page as the intended call. This allows the developer to place “call vectors” at the first 256 words of each page. The instruction at the “call vector” then executes a `GOTO` instruction to the subroutine anywhere in that page. The `RETLW` instruction, of the subroutine, will then POP the stack. The Stack contained the PUSHed PC from the `CALL` instruction.

Figure 2 shows an example of a “long call” sequence in a device with 2K-words.

FIGURE 1: PROGRAM COUNTER STRUCTURE

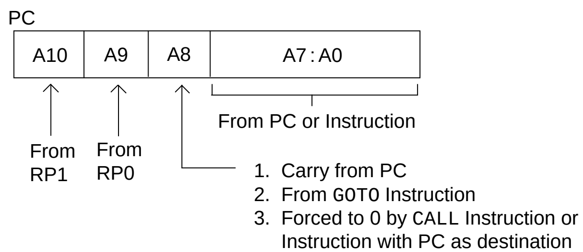
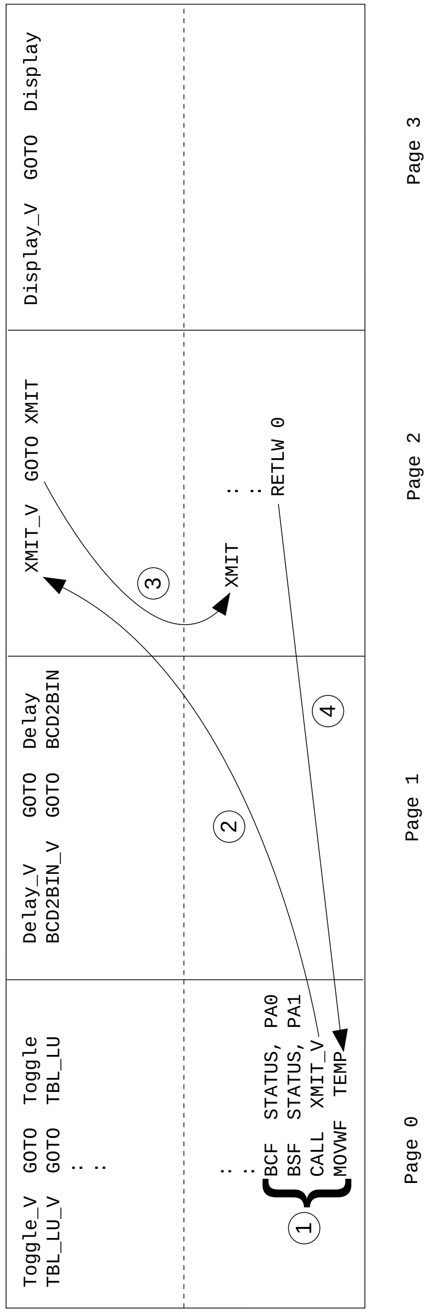


FIGURE 2: EXAMPLE OF A “LONG CALL” SEQUENCE



The flow that occurs in Figure 2 is as follows:

1. Select the program memory page of the desired subroutine and execute the call to that subroutine.
2. The program loads the Stack with the PC+1 address, branches to the selected page and specified address of the “call vector” (must be in the first 256 locations of the page)
3. Executes a GOTO instruction, to have access to the entire program memory page. Then executes the subroutine.
4. Executes the RETLW instruction, which POPs the new PC from the Stack. This causes program execution to continue at the instruction after the CALL instruction.

The use of “long calls” could be used to place all the subroutines in selected page(s), since the entire page can contain the subroutines (not restricted to the top half of the page). The placing of all subroutines in fewer program memory pages can reduce the overhead of specifying the required pages, since they are changed less frequently.

Use of the MPASM assembler can ease in the verification that call vectors and the call routine are in the same program memory page. Example 1 shows the use of assembler directives to print user defined warning or error messages in the listing file. These are shown as the shaded conditional statements. These messages are only printed in the listing file, and no indication of these messages is shown at the completion of assembly.

EXAMPLE 1: USE OF ASSEMBLER DIRECTIVES

```
;
P1_TOP              EQU            0x0000          ; First address in page 0
P2_TOP              EQU            0x0200          ; First address in page 1
P3_TOP              EQU            0x0400          ; First address in page 2
P4_TOP              EQU            0x0600          ; Reset vector address in page 1
RESET_V             EQU            0x07FF
;
     org    P1_TOP
;
     :
     :
     :
;
     org    P3_TOP
;
My_Subroutine_V     GOTO    My_Subroutine    ; Vector for My_Subroutine
     :
     :
     :
;
My_Subroutine                                ; My_Subroutine routine
;
    if ( ( My_Subroutine_V & 0x0600 ) != ( My_Subroutine & 0x0600 ) )
       MESSG   “ERROR - User Defined: CALL VECTOR and CALL routine NOT in same page”
    endif
;
     :
     :
     :
;
My_Subroutine_END    RETURN
;
    if ( ( My_Subroutine_V & 0x0600) != (My_Subroutine_END & 0x0600) )
       MESSG   “Warning - User Defined: Call routine crosses page boundry”
    endif
;
     :
     :
     :
;
       org   RESET_V                         ; Program memory address for the reset vector
;
              GOTO           START           ; Goto the beginning of the program
```

CONCLUSION

The use of “long calls” may ease the development of application programs. For minimal overhead, the application program can execute a subroutine from anywhere in the program memory, and return to the desired location. This eases the development of the application program, by reducing the mapping of subroutine in the first 256 words of each program memory page. The use of “long calls” is possible in any of the PIC16C5X devices, but is most useful in the devices with more than one program memory page. For device with more than one page of program memory, the assembler directives can be used to verify that the subroutines are in the program memory page.

Author: Mark Palmer - Sr. Application Engineer

WORLDWIDE SALES & SERVICE

AMERICAS

Corporate Office
Microchip Technology Inc.
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 602 786-7200 Fax: 602 786-7277
Technical Support: 602 786-7627
Web: http://www.mchip.com/microhip

Atlanta
Microchip Technology Inc.
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 770 640-0034 Fax: 770 640-0307

Boston
Microchip Technology Inc.
5 Mount Royal Avenue
Marlborough, MA 01752
Tel: 508 480-9990 Fax: 508 480-8575

Chicago
Microchip Technology Inc.
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 708 285-0071 Fax: 708 285-0075

Dallas
Microchip Technology Inc.
14651 Dallas Parkway, Suite 816
Dallas, TX 75240-8809
Tel: 214 991-7177 Fax: 214 991-8588

Dayton
Microchip Technology Inc.
35 Rockridge Road
Englewood, OH 45322
Tel: 513 832-2543 Fax: 513 832-2841

Los Angeles
Microchip Technology Inc.
18201 Von Karman, Suite 455
Irvine, CA 92715
Tel: 714 263-1888 Fax: 714 263-1338

New York
Microchip Technology Inc.
150 Motor Parkway, Suite 416
Hauppauge, NY 11788
Tel: 516 273-5305 Fax: 516 273-5335

AMERICAS (continued)

San Jose
Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408 436-7950 Fax: 408 436-7955

ASIA/PACIFIC

Hong Kong
Microchip Technology
Unit No. 3002-3004, Tower 1
Metroplaza
223 Hing Fong Road
Kwai Fong, N.T. Hong Kong
Tel: 852 2 401 1200 Fax: 852 2 401 3431

Korea
Microchip Technology
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku,
Seoul, Korea
Tel: 82 2 554 7200 Fax: 82 2 558 5934

Singapore
Microchip Technology
200 Middle Road
#10-03 Prime Centre
Singapore 188980
Tel: 65 334 8870 Fax: 65 334 8850

Taiwan
Microchip Technology
10F-1C 207
Tung Hua North Road
Taipei, Taiwan, ROC
Tel: 886 2 717 7175 Fax: 886 2 545 0139

EUROPE

United Kingdom
Arizona Microchip Technology Ltd.
Unit 6, The Courtyard
Meadow Bank, Furlong Road
Bourne End, Buckinghamshire SL8 5AJ
Tel: 44 0 1628 851077 Fax: 44 0 1628 850259

France
Arizona Microchip Technology SARL
2 Rue du Buisson aux Fraises
91300 Massy - France
Tel: 33 1 69 53 63 20 Fax: 33 1 69 30 90 79

Germany
Arizona Microchip Technology GmbH
Gustav-Heinemann-Ring 125
D-81739 Muenchen, Germany
Tel: 49 89 627 144 0 Fax: 49 89 627 144 44

Italy
Arizona Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Pegaso Ingresso No. 2
Via Paracelso 23, 20041
Agrate Brianza (MI) Italy
Tel: 39 039 689 9939 Fax: 39 039 689 9883

JAPAN

Microchip Technology Intl. Inc.
Benex S-1 6F
3-18-20, Shin Yokohama
Kohoku-Ku, Yokohama
Kanagawa 222 Japan
Tel: 81 45 471 6166 Fax: 81 45 471 6122

9/22/95