

セクション 4. プログラムメモリ

ハイライト

本セクションには以下の主要項目を記載しています。

4.1	プログラムメモリのアドレスマップ	4-2
4.2	制御レジスタ	4-4
4.3	プログラム カウンタ	4-6
4.4	テーブル命令を用いたプログラムメモリ アクセス	4-7
4.5	データ領域からの PSV	4-12
4.6	プログラムメモリの書き込み	4-16
4.7	レジスタマップ	4-17
4.8	関連アプリケーション ノート	4-18
4.9	改訂履歴	4-19

Note: ファミリ リファレンス マニュアルの本セクションは、デバイス データシートの内容を補足する事を目的としています。本セクションの内容は、dsPIC33F/PIC24H ファミリの一部のデバイスには対応していません。

本書の内容がお客様のご使用になるデバイスに対応しているかどうかは、最新デバイス データシート内の「プログラムメモリ」の冒頭に記載している注意書きでご確認ください。

デバイス データシートとファミリ リファレンス マニュアルの各セクションは、マイクロチップ社のウェブサイト (<http://www.microchip.com>) からダウンロードできます。

4.1 プログラムメモリのアドレスマップ

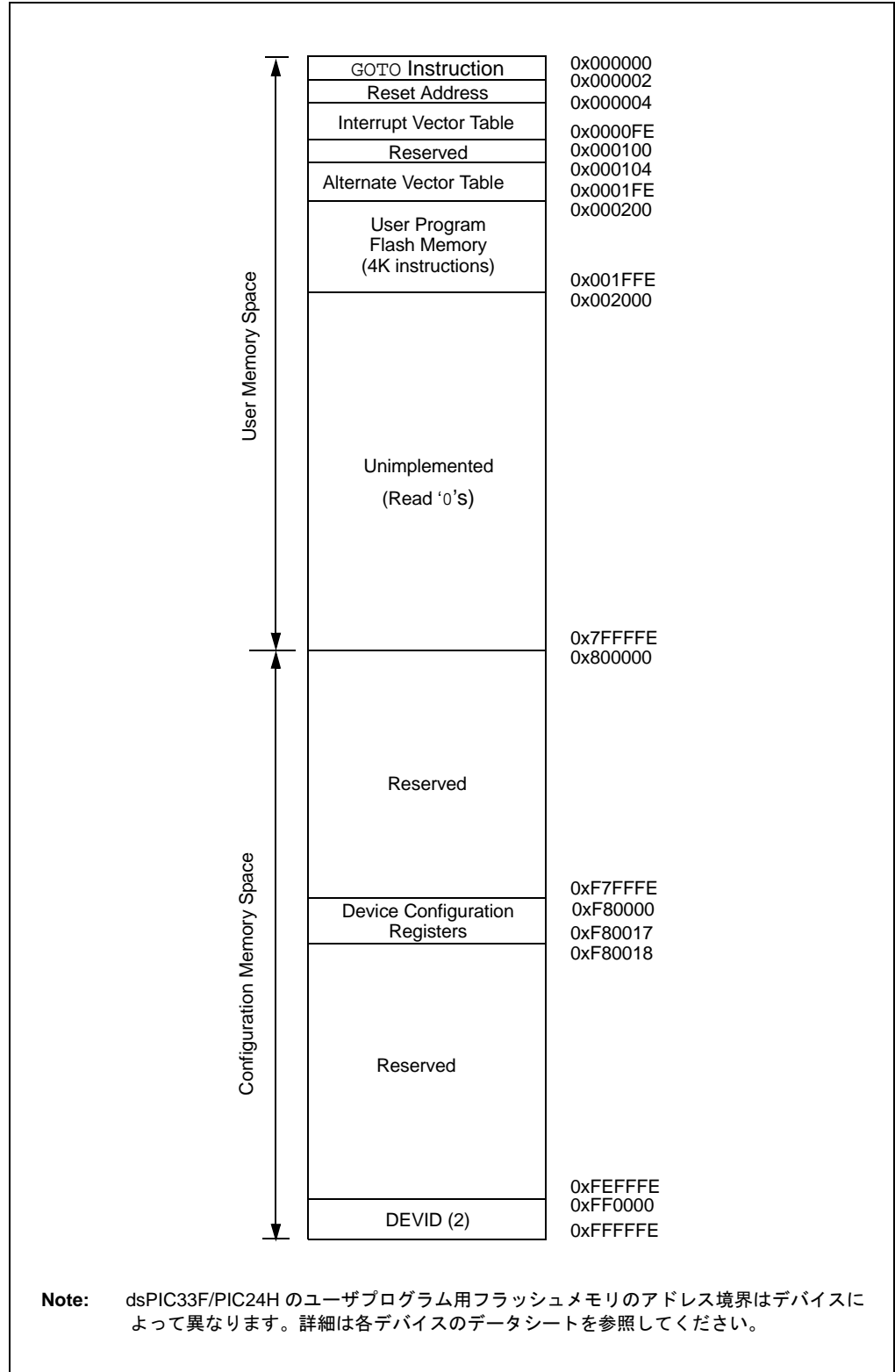
図 4-1 に、dsPIC33F/PIC24H が備える 4M x 24 ビットのプログラムメモリ アドレス領域を示します。プログラム領域へは下記の 3 つの方法でアクセスできます。

- 23 ビット PC (プログラム カウンタ) を使用する
- テーブル読み出し命令 (TBLRD) とテーブル書き込み命令 (TBLWT) を使用する
- プログラムメモリの 32K バイトセグメントをデータメモリ アドレス領域へ割り当てる

プログラムメモリ マップは、ユーザプログラム領域とユーザ設定領域に分割されます。ユーザプログラム領域はリセットベクタ、割り込みベクタテーブル、プログラムメモリを格納します。ユーザ設定領域は、デバイス オプションとデバイス ID 位置を設定するための不揮発設定ビット格納します。

Note: RETURN 命令をプログラムメモリの終端に置くと、ランタイム中に不正アドレス エラートラップが発生します。この問題は、命令プリフェッチが実際には存在しないメモリ位置から次の命令をプリロードしようと試みるために生じます。この問題を解消するには、RETURN 命令の後に 2 命令ワード分の領域を残して、コンパイラが NOP および RESET 命令をプログラムメモリの終端に格納できるようにする必要があります。

図 4-1: プログラムメモリ マップの例



4.2 制御レジスタ

レジスタ 4-1: CORCON: コア制御レジスタ

U-0	U-0	U-0	R/W-0	R/W-0	R-0	R-0	R-0
—	—	—	US	EDT	DL<2:0>		
bit 15							bit 8

R/W-0	R/W-0	R/W-1	R/W-0	R/C-0	R/W-0	R/W-0	R/W-0
SATA	SATB	SATDW	ACCSAT	IPL3	PSV	RND	IF
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-3 **プログラムメモリ関連には使用せず**
CORCON ビットの詳細は**セクション 2. 「CPU」 (DS70204)** を参照してください。

bit 2 **PSV:** データ領域 PSV のイネーブルビット
1 = データ領域内でプログラム領域は可視
0 = データ領域内でプログラム領域は不可視

bit 1-0 **プログラムメモリ関連には使用せず**
CORCON ビットの詳細は**セクション 2. 「CPU」 (DS70204)** を参照してください。

レジスタ 4-2: PSVPAG: PSV ページ レジスタ

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PSV アドレスページ ビット							
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-8 **未実装:** 「0」として読み出し

bit 7-0 **PSV アドレスページ ビット**
8 ビットの PSV アドレスページとワーキング レジスタの下位 15 ビット (15 LSb) を連結する事により、23 ビットの有効プログラムメモリ アドレスを形成します。

レジスタ 4-3: TBLPAG: テーブルページ レジスタ

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15						bit 8	

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
テーブル アドレスページ ビット							
bit 7						bit 0	

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
 -n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-8 **未実装:** 「0」として読み出し

bit 7-0 **テーブル アドレスページ ビット**

8 ビットのテーブル アドレスページとワーキング レジスタを連結する事により、23 ビットの有効プログラムメモリ アドレスとバイト選択ビットを形成します。

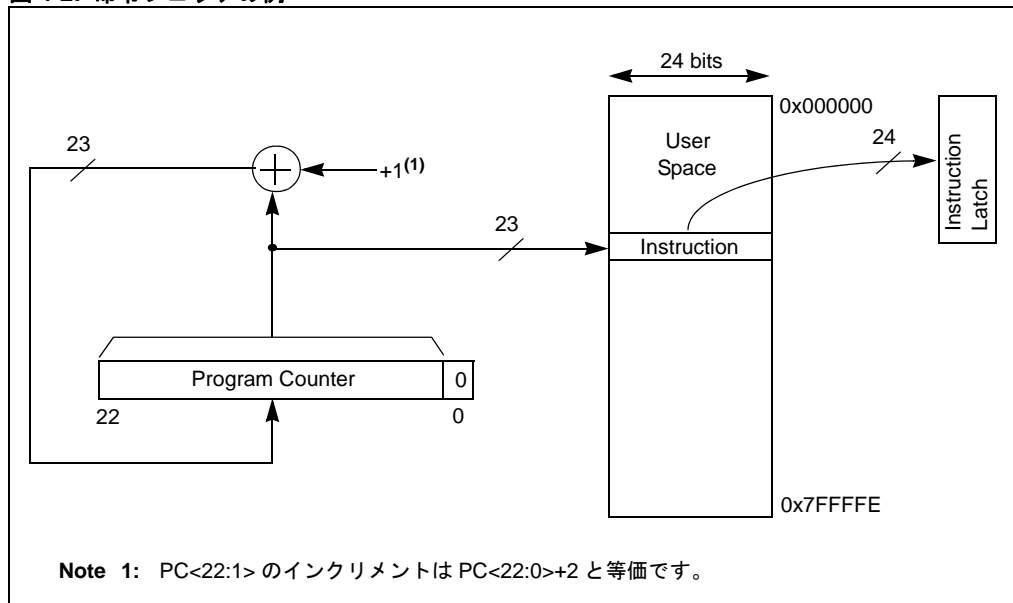
4.3 プログラム カウンタ

プログラムメモリは 24 ビット幅で構成され、上位 8 ビット下位 16 ビットです。16 ビットメモリに対してアドレスが割り付けられているため、プログラム 1 ワードは 2 番地に相当し、偶数 (最下位ビット 0) 番地のみの利用です。これにより、データ領域の 16 ビットアドレッシングとの互換性を確保します。連続する複数の命令ワードは、PC<22:1> によって 4M プログラムメモリ領域内にアドレスされます。各命令ワードは 24 ビット幅を持ちます。

プログラムメモリ アドレス (PC<0>) の LSb は、PSV またはテーブル命令を用いてデータ領域からプログラムメモリへアクセスする際に、バイト選択ビットとして機能します。PC を介する命令フェッチにはバイト選択ビットは不要です。従って PC<0> は常に「0」にセットされます。

図 4-2 に命令フェッチの例を示します。PC<22:1> を 1 つインクリメントする事と PC<22:0> に 2 を加算する事は等価である点に注意してください。

図 4-2: 命令フェッチの例



4.4 テーブル命令を用いたプログラムメモリ アクセス

TBLRDL および TBLWTL 命令を使用すると、データ領域を経由せずにプログラム領域内の任意アドレスの下位 16 ビット (lsb) を直接読み書きできます。これは一部のアプリケーションに便利です。TBLRDH および TBLWTH 命令は、プログラムワードの上位 8 ビットヘデータとしてアクセスできる唯一の方法です。

4.4.1 テーブル命令の概要

バイトサイズまたはワードサイズのデータをプログラム領域とデータ領域間で移動するには、各種のテーブル命令を使用します。テーブル読み出し命令は、プログラムメモリ領域からデータメモリ領域への読み出しに使用します。テーブル書き込み命令は、プログラムメモリ領域へのデータメモリの書き込みに使用します。

Note: テーブル命令を用いた詳細なサンプルコードについては、**セクション 5.「フラッシュプログラミング」** (DS70191) を参照してください。

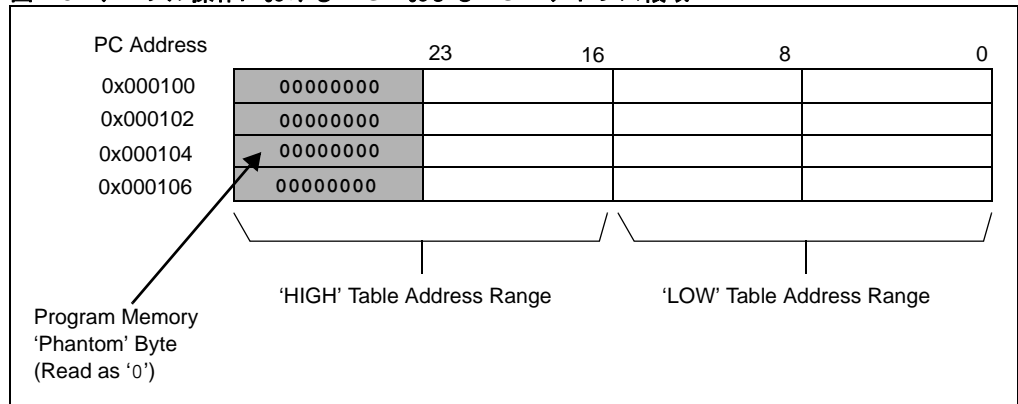
下記の 4 つのテーブル命令を使用できます。

- TBLRDL: テーブル読み出し LOW
- TBLWTL: テーブル書き込み LOW
- TBLRDH: テーブル読み出し HIGH
- TBLWTH: テーブル書き込み HIGH

テーブル命令では、プログラムメモリの 1 つのアドレス領域を 2 つの 16 ビット (ワード幅) アドレス領域として扱う事ができます (図 4-3 参照)。これにより、プログラム領域へバイトとしてアクセスするか、あるいはプログラム領域をワードアドレッシング可能な 16 ビット幅 /64K バイトページに (すなわちデータ領域と同様に) 配置できます。

TBLRDL と TBLWTL はプログラムメモリの下位データワードへアクセスし、TBLRDH と TBLWTH は上位データワードへアクセスします。プログラムメモリは 24 ビット幅しか持たないため、上位データワードの上位バイトはアドレッシング可能ですが、実際には存在しません。このため、このバイトを「幽霊」(phantom) バイトと呼びます。

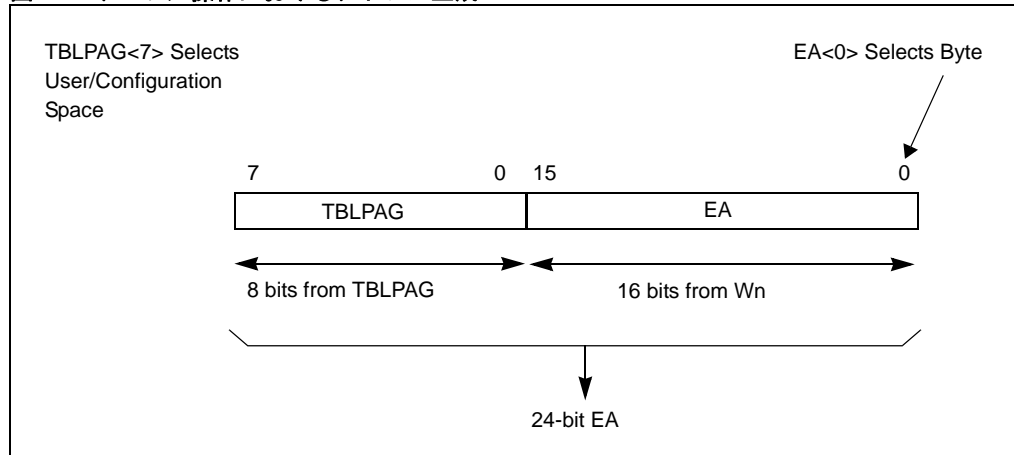
図 4-3: テーブル操作における HIGH および LOW アドレス領域



4.4.2 テーブルアドレスの生成

全てのテーブル命令では、ワーキングレジスタのアドレス値と8ビットのデータテーブルページレジスタ (TBLPAG) を連結して、23ビットの有効プログラム領域アドレスとバイト選択ビットを形成します (図 4-4 参照)。ワーキングレジスタから提供されるプログラム領域アドレスは15ビットです。従ってプログラムメモリ内のデータテーブルのページサイズは32Kワードです。

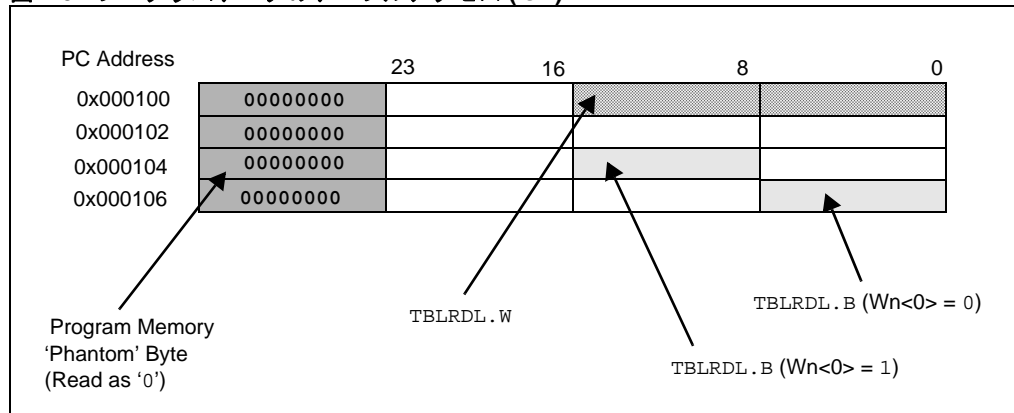
図 4-4: テーブル操作におけるアドレス生成



4.4.3 プログラムメモリの LOW ワードアクセス

TBLRD L および TBLWT L 命令は、プログラムメモリ データの下位 16 ビットへのアクセスに使用します。ワード幅テーブルアクセスでは、ワーキングレジスタ アドレスの LSb を無視します。バイト幅アクセスでは、ワーキングレジスタ アドレスの LSb によって、どちらのバイトを読み出すかが決まります。図 4-5 に、TBLRD L および TBLWT L 命令がアクセスするプログラムメモリ データ領域を示します。

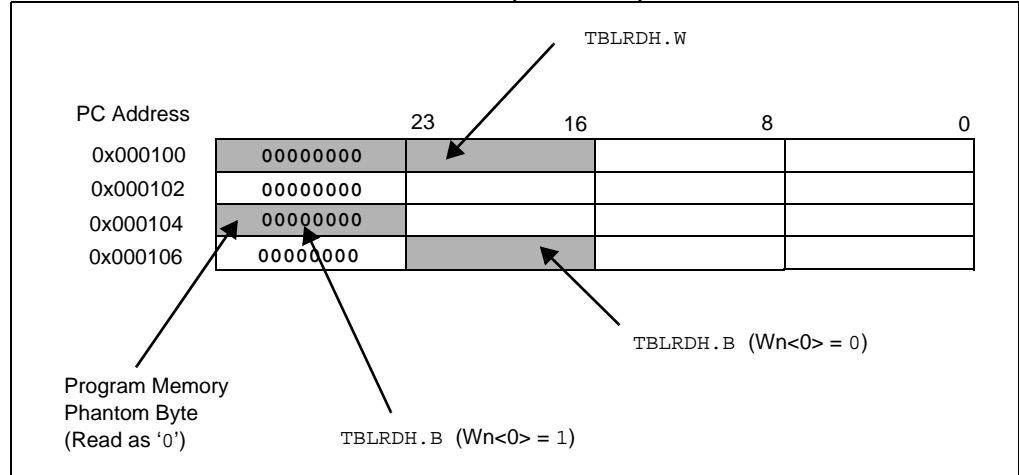
図 4-5: プログラムデータのテーブルアクセス (lsw)



4.4.4 プログラムメモリの HIGH ワードアクセス

TBLRDH および TBLWTH 命令は、プログラムメモリ データの上位 8 ビットへのアクセスに使用します。図 4-6 に、これらの命令がサポートするワードアクセスおよびバイトアクセス モードを示します。ただし、プログラムメモリ データの上位バイトは常に「0」を返します。

図 4-6: プログラムデータのテーブルアクセス (MS バイト)



4.4.5 プログラムメモリ内のデータ保存

多くのアプリケーションでは 16 ビット幅を使用してデータを保存するため、プログラムメモリの上位バイト (PC<23:16>) をデータ用には使用しません。このような場合、プログラムデータの上位バイトに NOP 命令または無効なオペコード値をプログラミングして、保存データの異常実行を回避する事を推奨します。TBLRDH および TBLWTH 命令は、配列プログラム / 検証目的での使用と、圧縮データ保存を必要とするアプリケーション向けの使用を主な用途とします。

4.4.6 テーブル命令を用いたプログラムメモリ アクセスの例

例 4-1 では、テーブル命令を使用して、アドレス 0x12000 から始まるプログラムメモリ ページを消去し、値 0x123456 および 0x789ABC をアドレス 0x12000 および 0x12002 へ書き込みます。

Note: ロック解除シーケンスの詳細はセクション 5.「フラッシュ プログラミング」(DS70191) を参照してください。

例 4-1: テーブル命令を用いたプログラムメモリへのアクセス

```
#define PM_ROW_ERASE 0x4042
#define PM_ROW_WRITE 0x4001
#define CONFIG_WORD_WRITE0X4000

unsigned long Data;

/* Erase 512 instructions starting at address 0x12000 */
MemWriteLatch(0x1, 0x2000,0x0,0x0);
MemCommand(PM_ROW_ERASE);

/* Write 0x12345 into program address 0x12000 */
MemWriteLatch(0x1, 0x2000,0x0012,0x3456);
MemCommand(PM_ROW_WRITE);

/* Write 0x789ABC into program address 0x12002 */
MemWriteLatch(0x1, 0x2002,0x0078,0x9ABC);
MemCommand(PM_ROW_WRITE);

/* Read program addresses 0x12000 and 0x12002 */
Data = MemReadLatch(0x1, 0x2000);
Data = MemReadLatch(0x1, 0x2002);

;*****
;_MemWriteLatch:
;
;W0 = TBLPAG
;W1 = Wn
;W2 = WordHi
;W3 = WordLo
;no return values
_MemWriteLatch:
    mov W0, TBLPAG
    tblwtl W3, [W1]
    tblwth W2, [W1]

    return

;*****
;_MemReadLatch:
;
;W0 = TBLPAG
;W1 = Wn
;return:data in W1:W0
_MemReadLatch:
    mov W0,TBLPAG
    tblrdl [W1],W0
    tblrdh [W1],W1

    return

;*****
;_MemCommand:
;
;W0 = NVMCON
;no return values
_WriteCommand:
    mov W0,NVMCON
    mov #0x55,W0;Unlock sequence
    mov W0,NVMKEY
    mov #0xAA,W0
    mov W0,NVMKEY
    bset NVMCON,#WR
    nop                ;Required
    nop
Loop:btsc NVMCON,#WR;Wait for write end
    bra Loop

return
```

例 4-2 では、プログラムメモリ内にバッファを配置するために `space(prog)` 属性を使用します。バッファへのアクセスには、MPLAB®C30 のビルトイン関数 (`builtin_tblpage`、`builtin_tbloffset` 等) を使用できます。

例 4-2: MPLAB®C30 の内蔵関数を用いたプログラムメモリへのアクセス

```
#include <p33fxxxx.h>

unsigned prog_data[10] __attribute__((space(prog))) = {0x0000, 0x1111, 0x2222,
0x3333, 0x4444, 0x5555, 0x6666, 0x7777, 0x8888, 0x9999};

unsigned lowWord, tbloffset;

int main(void)
{
    TBLPAG = __builtin_tblpage(&prog_data);
    tbloffset = __builtin_tbloffset(&prog_data);
    lowWord = __builtin_tblrld(tbloffset + 6); // Load 0x3333 into lowWord

    for(;;);
    return 0;
}
```

4.5 データ領域からの PSV

dsPIC33F/PIC24H が備えるデータメモリ アドレス領域の上位 32K バイトは、必要に応じて任意の 16K ワード プログラム領域ページへ割り当てることができます。この動作モードは PSV (Program Space Visibility) と呼ばれ、特殊な命令 (TBLRD、TBLWT 等) を必要とせずに、X データ領域からプログラムメモリに保存された定数データへの透過的なアクセスを可能にします。

4.5.1 PSV の設定

コア制御レジスタ (CORCON<2>) 内の PSV ビットをセットすると PSV が有効になります。CORCON レジスタの詳細はセクション 2. 「CPU」 (DS70204) を参照してください。

PSV を有効にすると、データメモリ マップの上位側半分の各データ領域アドレスが、それぞれ 1 つのプログラム アドレスへ直接割り当てられます (図 4-7 参照)。PSV ウィンドウは、24 ビット プログラムワードの下位 16 ビットへのアクセスを可能にします。デバイスの動作信頼性を確保するために、プログラムメモリ データの上位 8 ビットには無効な命令または NOP 命令をプログラミングしておく必要があります。各プログラムメモリ ワードの上位 8 ビットを読み出すには、テーブル命令を使用します。

図 4-8 に PSV アドレスの生成方法を示します。PSV アドレスの下位 15 ビットは、有効アドレスを格納したワーキング レジスタから得られます。ワーキング レジスタの最上位ビット (MSb) は、アドレスの形成には使用しません。この MSb は、プログラム領域からの PSV アクセスを実行するのか、それともデータメモリ領域から通常のアクセスを実行するのかを指定するために使用します。PSV が有効になっている場合に 0x8000 以上のワーキング レジスタ有効アドレスを使用すると、プログラムメモリ領域からデータアクセスが発生します。ワーキング レジスタ有効アドレスが 0x8000 より小さい場合には、全てのデータアクセスはデータメモリから発生します。

有効アドレスを保持するワーキング レジスタの下位 15 ビットと、8 ビットの PSV ページアドレス レジスタ (PSVPAG<7:0>) を連結する事によって、23 ビットのプログラムメモリ アドレスを形成します (図 4-8 参照)。PSV は、プログラムメモリ領域内の値へのアクセスにのみ使用できます。ユーザ設定領域内の値へのアクセスには、テーブル命令を使用する必要があります。

ワーキング レジスタ値の最下位ビット (LSb) はバイト選択ビットとして使用します。これにより、PSV を使用する命令は、バイトモードまたはワードモードのいずれかで動作することができます。

4.5.2 X および Y データ領域での PSV マッピング

dsPIC33F/PIC24H の大部分の製品では、Y データ領域はデータ領域の上位側半分の外側に配置され、従って PSV 領域は X データ領域内へ割り当てられます。X および Y データ領域の割り当ては、アルゴリズム内の PSV の使用方法に影響します。

例として FIR フィルタ アルゴリズムでは、PSV マッピングを用いて係数データを保存することができます。FIR フィルタは、フィルタ入力データの履歴を格納したデータバッファの各値に、フィルタ係数を格納したデータバッファの各値を乗算します。FIR アルゴリズムは、REPEAT ループ内で MAC 命令を使用します。MAC 命令は各繰り返しで履歴入力値と係数値を 1 つずつプリフェッチします。これらの値は次の繰り返しで乗算されます。プリフェッチした値の片方を X データメモリ領域に格納し、他方を Y データメモリ領域に格納する必要があります。

FIR フィルタ アルゴリズムでは、PSV マッピング要件を満たすために、履歴入力データを Y メモリ領域に格納し、フィルタ係数を X メモリ領域に格納する必要があります。

図 4-7: PSV の動作

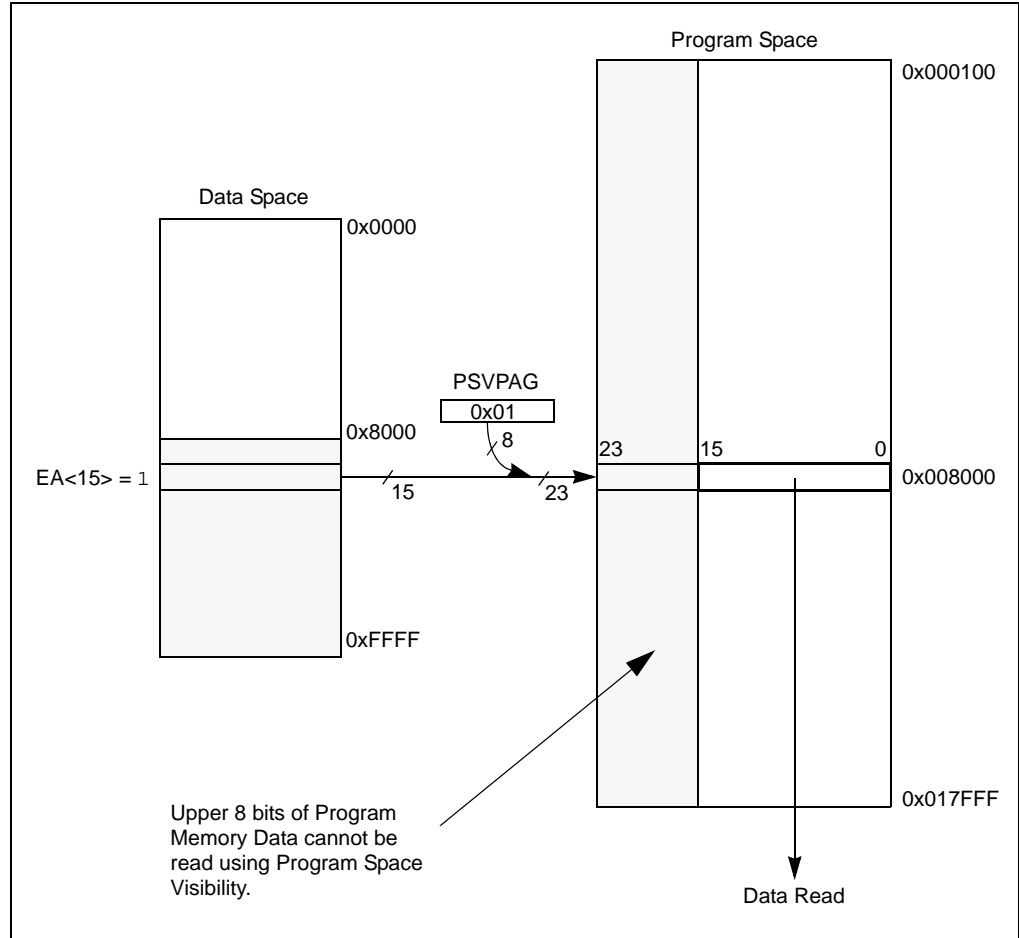
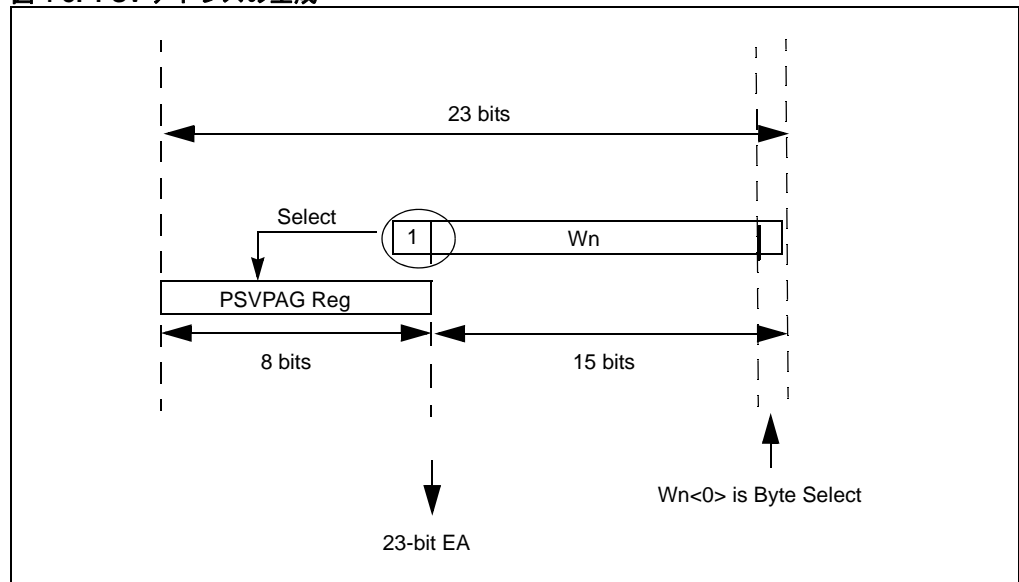


図 4-8: PSV アドレスの生成



4.5.3 PSV の実行時間

PSV を使用する命令の実行には、追加の命令サイクルが 2 つ必要です。例外として、下記の命令の実行には追加命令サイクルが 1 つだけ必要です。

- データ プリフェッチ オペランドを持つ MAC クラス命令
- 全ての MOV 命令 (MOV.D 命令を含む)

追加命令サイクルは、プログラムメモリ バス上で PSV データをフェッチするために必要となります。

4.5.3.1 繰り返しループ内での PSV の使用

REPEAT ループ内で PSV を使用する命令は、プログラムメモリからのデータアクセスに追加の命令サイクルを必要としません。従って実行時間にオーバーヘッドを生じません。ただし、REPEAT ループ内の下記の命令実行では、2 命令サイクルのオーバーヘッドが生じます。

- 最初の命令実行
- 最後の命令実行
- 割り込みによるループ中断前の命令実行
- 割り込みサービス終了後のループ再開時の命令実行

4.5.3.2 PSV と命令ストール

PSV を使用する命令ストールの詳細は**セクション 2.「CPU」**(DS70204) を参照してください。

4.5.4 PSV のサンプルコード

バッファを作成してコンパイラが管理する PSV セクション内でそのバッファへアクセスする方法を例 4-3 に示します。auto_psv 領域がコンパイラが管理する PSV セクションです。このセクションのサイズが 32K を超えると、リンカがエラーを出力します。一連のツールは、プログラム起動時に PSVPAG が正しく設定されるように準備します。既定値では、コンパイラは全ての「const」宣言変数を auto_psv 領域に格納します。

例 4-3: コンパイラが管理する PSV アクセス

```
#include "p33fxxxx.h"

int m[5] __attribute__((space(auto_psv))) = {1, 2, 3, 4, 5};
int x[5] = {10, 20, 30, 40, 50};
int sum;

main()
{
    // Compiler Managed PSV
    sum=vectordot(m,x);
}

int vectordot(int *m, int *x)
{
    int i,sum=0;

    for(i=0;i<5;i++)
        sum+=*m++ * *x++;

    return(sum);
}
```

例 4-4 に、ユーザが管理する PSV セクション内のバッファの配置とアクセスを示します。
psv 領域がユーザが管理する PSV セクションです。

例 4-4: ユーザが管理する PSV アクセス

```
#include "p33fxxxx.h"

const int m[5] = {1, 2, 3, 4, 5};
int m1[5] __attribute__((space(psv))) = {1, 2, 3, 4, 5};
int m2[5] __attribute__((space(psv), address(0xA000))) = {1, 2, 3, 4, 5};
int x[5] = {10, 20, 30, 40, 50};
int sum, sum1, sum2;

main()
{
    int temp;

    // User Managed PSV
    temp=PSVPAG; // Save auto_psv page

    PSVPAG = __builtin_psvpage(&m1);
    CORCONbits.PSV = 1;
    sum1=vectordot(m1,x);

    PSVPAG = __builtin_psvpage(&m2);
    sum2=vectordot(m2,x);

    PSVPAG=temp; // Restore auto_psv page

    // Compiler Managed PSV
    sum=vectordot(m,x);
}

int vectordot(int *m, int *x)
{
    int i,sum=0;

    for(i=0;i<5;i++)
        sum+=*m++ * *x++;

    return(sum);
}
```

プログラムメモリ内に定数データを配置し、PSV データ ウィンドウを介してのこのデータへアクセスするためのアセンブリ プログラムを例 4-5 に示します。

例 4-5: PSV のアセンブリ サンプルコード

```
.include "p33fxxxx.inc"
.section .const,psv
fib_data:
    .word 0, 1, 2, 3, 5, 8, 13

;Start of Code section
.text

.global __reset
__reset:
    ; Enable Program Space Visibility
    bset.b CORCONL, #PSV

    ; Set PSVPAG to the page that contains the fib_data array
    mov #psvpag(fib_data), w0
    mov w0, __PSVPAG

    ; Make a pointer to fib_data in the PSV data window
    mov #psvoffset(fib_data), w0

    ; Load the first data value
    mov [w0++], w1
```

4.6 プログラムメモリの書き込み

dsPIC33F/PIC24H ファミリのデバイスは、ユーザコードの実行用にプログラム フラッシュメモリを内蔵しています。このメモリへは、下記の 2 つの方法でユーザ アプリケーションをプログラミングできます。

- ランタイム自己プログラミング (RTSP)
- インサーキット シリアル プログラミング (ICSP™)

RTSP には TBLWT 命令を使用します。ICSP には SPI インターフェイスとブートローダ ソフトウェアを使用します。RTSP の詳細は**セクション 5.「フラッシュ プログラミング」**(DS70191) を参照してください。ICSP の仕様書はマイクロチップ社のウェブサイト (www.microchip.com) からダウンロードできます。

4.7 レジスタマップ

プログラムメモリに関連するレジスタの概要を表 4-1 に記載します。

表 4-1: プログラムメモリ レジスタ

SFR	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全 リセット				
TBLPAG	—	—	—	—	—	—	—	—	テーブルページ アドレスポイント レジスタ												0000
PSVPAG	—	—	—	—	—	—	—	—	PSV ページ アドレスポイント レジスタ												0000
CORCON	—	—	—	US	EDT	DL<2:0>	SATA	SATB	SATDW	ACCSAT	IPL3	PSV	RND	IF			0000				

凡例： — = 未実装、「0」として読み出し。網掛けしたビットはプログラムメモリ関連には使用しません。

4.8 関連アプリケーション ノート

本セクションに関連するアプリケーション ノートを紹介します。これらのアプリケーション ノートは dsPIC33F/PIC24H デバイスファミリ向けではありません。ただし概念は共通しており、変更が必要であったり制限事項が存在するものの利用が可能です。プログラムメモリ モジュールに関連する最新のアプリケーション ノートは以下の通りです。

タイトル

アプリケーション ノート番号

現在関連するアプリケーション ノートはありません。

<p>Note: dsPIC33F/PIC24H ファミリ向けのその他のアプリケーション ノートとサンプルコードについては、マイクロチップ社のウェブサイト (www.microchip.com) をご覧ください。</p>

4.9 改訂履歴

リビジョン A (2007 年 3 月)

本書の初版

リビジョン B (2007 年 4 月)

本書全体の小規模な更新

リビジョン C (2008 年 7 月)

このリビジョンでの変更内容は次の通りです。

- セクション 4.2「制御レジスタ」および 4.6「プログラムメモリの書き込み」を再編成
- レジスタ：
 - PSVPAG:PSV ページレジスタ (レジスタ 4-2 参照)
 - TBLPAG: テーブルページ レジスタ (レジスタ 4-3 参照)
- 例：
 - MPLAB C30 の内蔵関数を用いたプログラムメモリ アクセス (例 4-2 参照)
 - コンパイラ マネージド PSV アクセス (例 4-3 参照)
 - ユーザマネージド PSV アクセス (例 4-4 参照)
 - PSV アセンブリ サンプルコード (例 4-5 参照)
- レジスタマップ セクションを追加 (例 4-5:「PSV のアセンブリ サンプルコード」参照)
- 上記に加えて、表現および体裁の変更等、本書全体の細部を修正

リビジョン D (2010 年 7 月)

このリビジョンでの変更内容は次の通りです。

- 例 4-1 と例 4-2 のコードを更新
- 文章および体裁の変更等、本書全体の細部を修正

NOTES:

マイクロチップ社製デバイスのコード保護機能に関して次の点にご注意ください。

- マイクロチップ社製品は、該当するマイクロチップ社データシートに記載の仕様を満たしています。
- マイクロチップ社では、通常の条件ならびに仕様に従って使用した場合、マイクロチップ社製品のセキュリティ レベルは、現在市場に流通している同種製品の中でも最も高度であると考えています。
- しかし、コード保護機能を解除するための不正かつ違法な方法が存在する事もまた事実です。弊社の理解ではこうした手法は、マイクロチップ社データシートにある動作仕様書以外の方法でマイクロチップ社製品を使用する事になります。このような行為は知的所有権の侵害に該当する可能性が非常に高いと言えます。
- マイクロチップ社は、コードの保全性に懸念を抱くお客様と連携し、対応策に取り組んでいきます。
- マイクロチップ社を含む全ての半導体メーカーで、自社のコードのセキュリティを完全に保証できる企業はありません。コード保護機能とは、マイクロチップ社が製品を「解読不能」として保証するものではありません。

コード保護機能は常に進歩しています。マイクロチップ社では、常に製品のコード保護機能の改善に取り組んでいます。マイクロチップ社のコード保護機能の侵害は、デジタル ミレニアム著作権法に違反します。そのような行為によってソフトウェアまたはその他の著作物に不正なアクセスを受けた場合は、デジタル ミレニアム著作権法の定めるところにより損害賠償訴訟を起こす権利があります。

本書に記載されているデバイス アプリケーション等に関する情報は、ユーザの便宜のためにのみ提供されているものであり、更新によって無効とされる事があります。お客様のアプリケーションが仕様を満たす事を保証する責任は、お客様にあります。マイクロチップ社は、明示的、暗黙的、書面、口頭、法定のいずれであるかを問わず、本書に記載されている情報に関して、状態、品質、性能、商品性、特定目的への適合性をはじめとする、いかなる類の表明も保証も行いません。マイクロチップ社は、本書の情報およびその使用に起因する一切の責任を否認します。マイクロチップ社の明示的な書面による承認なしに、生命持装置あるいは生命安全用途にマイクロチップ社の製品を使用する事は全て購入者のリスクとし、また購入者はこれによって発生したあらゆる害、クレーム、訴訟、費用に関して、マイクロチップ社は擁護され、免責され、損害をうけない事に同意するものとします。暗黙的あるいは明示的を問わず、マイクロチップ社が知的財産権を保有しているライセンスは一切譲渡されません。

商標

マイクロチップ社の名称と Microchip ロゴ、dsPIC、KEELOQ、KEELOQ ロゴ、MPLAB、PIC、PICmicro、PICSTART、rPIC、UNI/O は、米国およびその他の国におけるマイクロチップ・テクノロジー社の登録商標です。

FilterLab、Hampshire、HI-TECH C、Linear Active Thermistor、MXDEV、MXLAB、SEEVAL、Embedded Control Solutions Company は、米国におけるマイクロチップ・テクノロジー社の登録商標です。

Analog-for-the-Digital Age、Application Maestro、CodeGuard、dsPICDEM、dsPICDEM.net、dsPICworks、dsSPEAK、ECAN、ECONOMONITOR、FanSense、HI-TIDE、In-Circuit Serial Programming、ICSP、Mindi、MiWi、MPASM、MPLAB Certified ロゴ、MPLIB、MPLINK、mTouch、Octopus、Omniscient Code Generation、PICC、PICC-18、PICDEM、PICDEM.net、PICkit、Pictail、PIC³² ロゴ、REAL ICE、rLAB、Select Mode、Total Endurance、TSHARC、UniWinDriver、WiperLock、ZENA は、米国およびその他の国におけるマイクロチップ・テクノロジー社の登録商標です。

SQTP は、米国におけるマイクロチップ・テクノロジー社のサービスマークです。

その他、本書に記載されている商標は各社に帰属します。

© 2010, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.



本書は再生紙を使用しています。

ISBN: 978-1-60932-499-5

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949:2002 ==

マイクロチップ社では、Chandler および Tempe (アリゾナ州)、Gresham (オレゴン州) の本部、設計部およびウェハー製造工場としてカリフォルニア州とイダホのデザインセンターが ISO/TS-16949:2002 認証を取得しています。マイクロチップ社の品質システム プロセスおよび手順は、PIC[®]MCU および dsPIC[®]DSC、KEELOQ[®]コード ホッピング デバイス、シリアル EEPROM、マイクロベリフェラル、不揮発性メモリ、アナログ製品に採用されています。さらに、開発システムの設計と製造に関するマイクロチップ社の品質システムは ISO 9001:2000 認証を取得しています。

各国の営業所とサービス

北米

本社
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
技術サポート :
<http://support.microchip.com>
URL:
www.microchip.com

アトランタ
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

ボストン
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

シカゴ
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

クリーブランド
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

ダラス
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

デトロイト
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

ココモ
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

ロサンゼルス
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

サンタクララ
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

トロント
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

アジア / 太平洋

アジア太平洋支社
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

オーストラリア - シドニー
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

中国 - 北京
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

中国 - 成都
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

中国 - 重慶
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

中国 - 香港 SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

中国 - 南京
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

中国 - 青島
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

中国 - 上海
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

中国 - 瀋陽
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

中国 - 深圳
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

中国 - 武漢
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

中国 - 西安
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

中国 - 厦門
Tel: 86-592-2388138
Fax: 86-592-2388130

中国 - 珠海
Tel: 86-756-3210040
Fax: 86-756-3210049

アジア / 太平洋

インド - バンガロール
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

インド - ニューデリー
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

インド - プネ
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

日本 - 横浜
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

韓国 - 大邱
Tel: 82-53-744-4301
Fax: 82-53-744-4302

韓国 - ソウル
Tel: 82-2-554-7200
Fax: 82-2-558-5932??
82-2-558-5934

マレーシア - クアラルンプール
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

マレーシア - ペナン
Tel: 60-4-227-8870
Fax: 60-4-227-4068

フィリピン - マニラ
Tel: 63-2-634-9065
Fax: 63-2-634-9069

シンガポール
Tel: 65-6334-8870
Fax: 65-6334-8850

台湾 - 新竹
Tel: 886-3-6578-300
Fax: 886-3-6578-370

台湾 - 高雄
Tel: 886-7-536-4818
Fax: 886-7-536-4803

台湾 - 台北
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

タイ - バンコク
Tel: 66-2-694-1351
Fax: 66-2-694-1350

ヨーロッパ

オーストリア - ヴェルス
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

デンマーク - コペンハーゲン
Tel: 45-4450-2828
Fax: 45-4485-2829

フランス - パリ
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

ドイツ - ミュンヘン
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

イタリア - ミラノ
Tel: 39-0331-742611
Fax: 39-0331-466781

オランダ - ドリューネン
Tel: 31-416-690399
Fax: 31-416-690340

スペイン - マドリッド
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

イギリス - ウォーキングサム
Tel: 44-118-921-5869
Fax: 44-118-921-5820