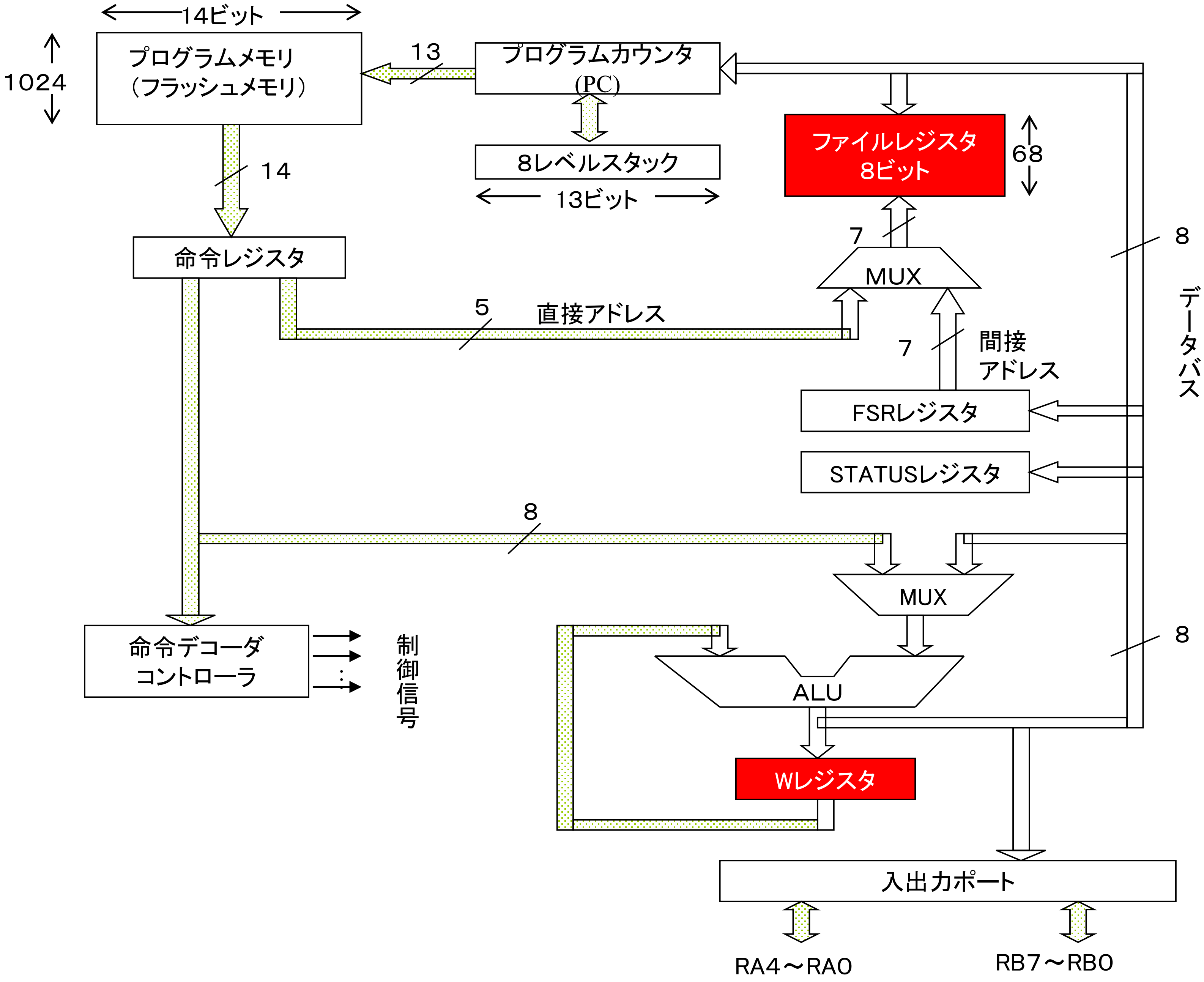
３．演算命令を学ぼう

本稿のWebページ
http://www.mybook-pub-site.sakura.ne.jp/PIC/index.html

このソースファイルを各自打ち込んで下さい.

足し算を実行するプログラムの例です.

```
; ADD

        INCLUDE"p16F84A.inc"
        list p=16F84A

        __CONFIG  _HS_OSC  &  _WDT_OFF  &  _PWRTE_OFF  &  _CP_OFF

MEM1    EQU     0x0C                    ;MEM1 at 0C

        ORG     0
        GOTO    START
        ORG     4

START
        MOVLW   B‘00000001              ; Load 0x01 to W

        ADDLW   B'00000011'             ; W + 0x03 ->W

        MOVWF   MEM1                    ; Move w to MEM1

        MOVLW   B'00000010'             ; Load 0x02 to W

        ADDWF   MEM1, 0                 ; W + MEM1 -> W

        ADDWF   MEM1, 1                 ; W + MEM1 -> MEM1

        GOTO    START

        END
```



PIC16F84Aの構成

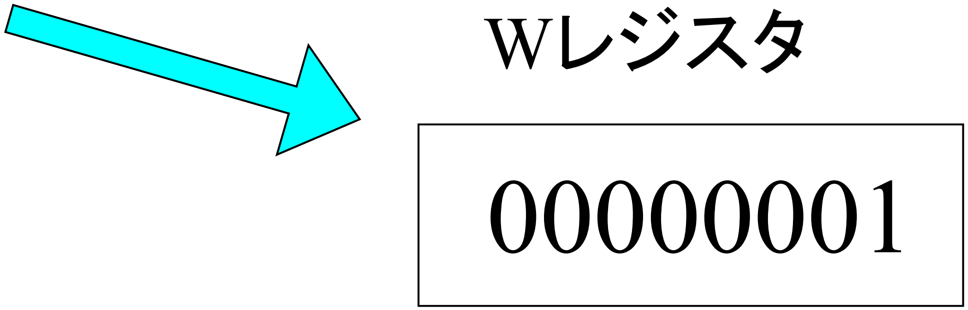
```
; ADD

        INCLUDE"p16F84A.inc"
        list p=16F84A

        __CONFIG   _HS_OSC  &  _WDT_OFF  &  _PWRTE_OFF  &  _CP_OFF

MEM1    EQU     0x0C                    ;MEM1 at 0C

        ORG     0
        GOTO    START
        ORG     4

START
        MOVLW   B'00000001              ; Load 0x01 to W

        ADDLW   B'00000011'             ; W + 0x03 ->W

        MOVWF   MEM1                    ; Move w to MEM1

        MOVLW   B'00000010'             ; Load 0x02 to W

        ADDWF   MEM1, 0                 ; W + MEM1 -> W

        ADDWF   MEM1, 1                 ; W + MEM1 -> MEM1

        GOTO    START

        END
```

電源を入れるとORG 0の次から始まる.

STARTへジャンプする.

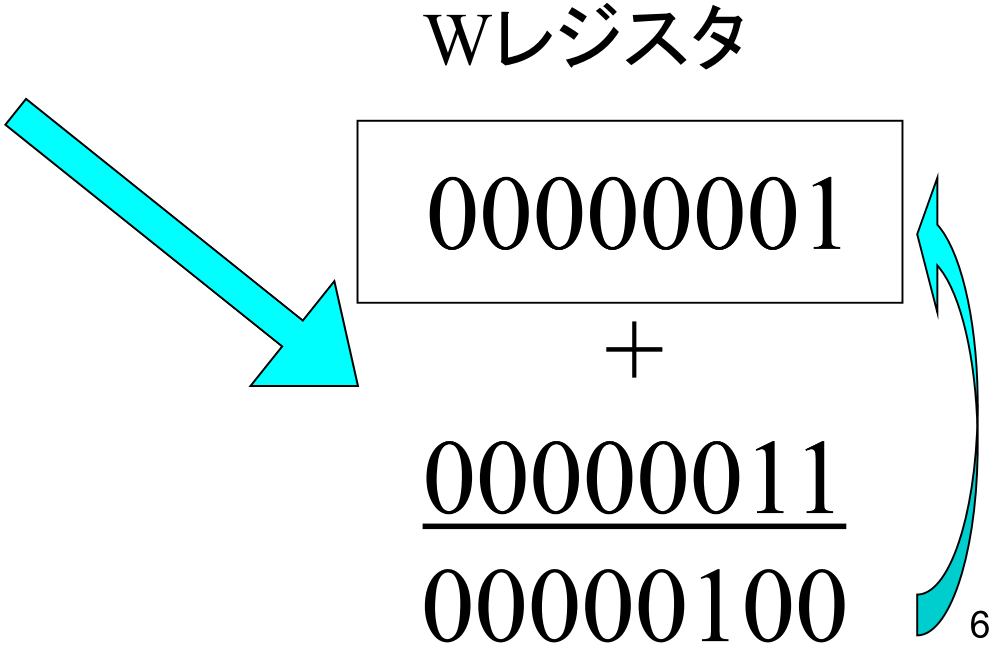
```
; ADD

        INCLUDE"p16F84A.inc"
        list p=16F84A

        __CONFIG  _HS_OSC  &  _WDT_OFF  &  _PWRTE_OFF  &  _CP_OFF

MEM1    EQU     0x0C

        ORG     0
        GOTO    START
        ORG     4

START
        MOVLW   B‘00000001

        ADDLW   B'00000011’

        MOVWF   MEM1

        MOVLW   B'00000010’

        ADDWF   MEM1, 0

        ADDWF   MEM1, 1

        GOTO    START

        END
```

wレジスタに00000001という数を転送せよという命令

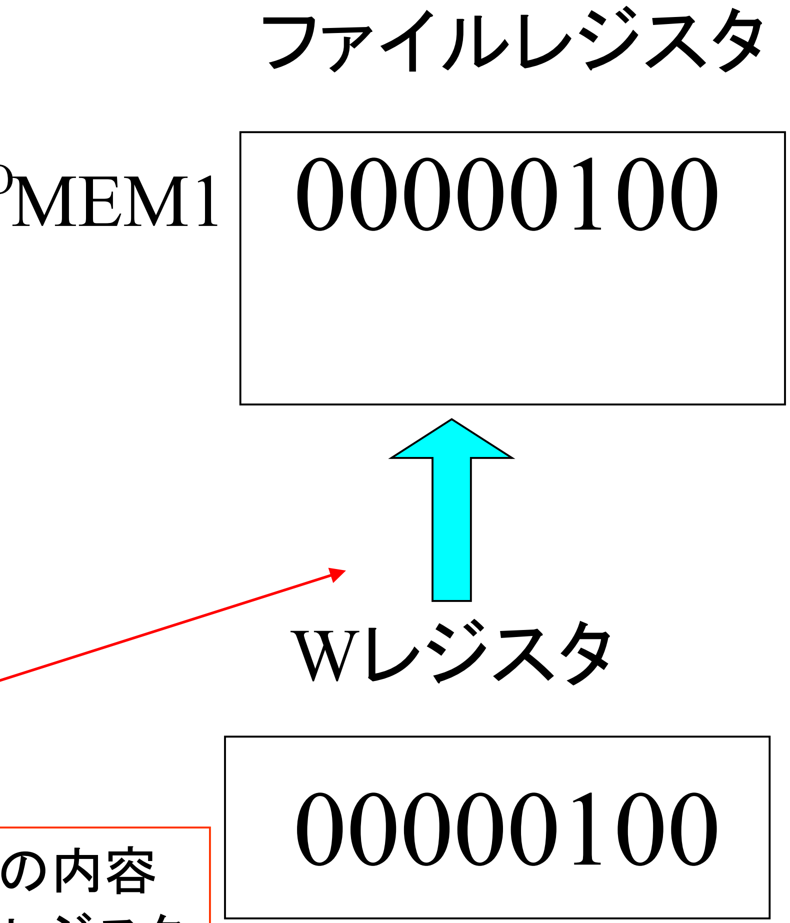


```
; ADD

        INCLUDE"p16F84A.inc"
        list p=16F84A

        __CONFIG  _HS_OSC & _WDT_OFF & _PWRTE_OFF & _CP_OFF

MEM1    EQU     0x0C

        ORG     0
        GOTO    START
        ORG     4

START
        MOVLW   B‘00000001

        ADDLW   B'00000011’

        MOVWF   MEM1

        MOVLW   B'00000010’

        ADDWF   MEM1, 0

        ADDWF   MEM1, 1

        GOTO    START

        END
```

wレジスタの内容に00000011という数を足してその結果をwレジスタに転送せよという命令です.
Add Literal to Wの略です.



```
; ADD

        INCLUDE"p16F84A.inc"
        list p=16F84A

        __CONFIG _HS_OSC & _WDT_O

MEM1    EQU     0x0C

        ORG     0
        GOTO    START
        ORG     4

START
        MOVLW   B‘00000001

        ADDLW   B'00000011’

        MOVWF   MEM1

        MOVLW   B'00000010’

        ADDWF   MEM1, 0

        ADDWF   MEM1, 1

        GOTO    START

        END
```



wレジスタの内容をファイルレジスタのMEM1に転送せよという命令

```
; ADD

        INCLUDE"p16F84A.inc"
        list p=16F84A

        __CONFIG  _HS_OSC  &  _WI

MEM1    EQU     0x0C

        ORG     0
        GOTO    START
        ORG     4

START

        MOVLW   B'00000001

        ADDLW   B'00000011'

        MOVWF   MEM1

        MOVLW   B'00000010'

        ADDWF   MEM1, 0

        ADDWF   MEM1, 1

        GOTO    START

        END
```

ファイルレジスタ

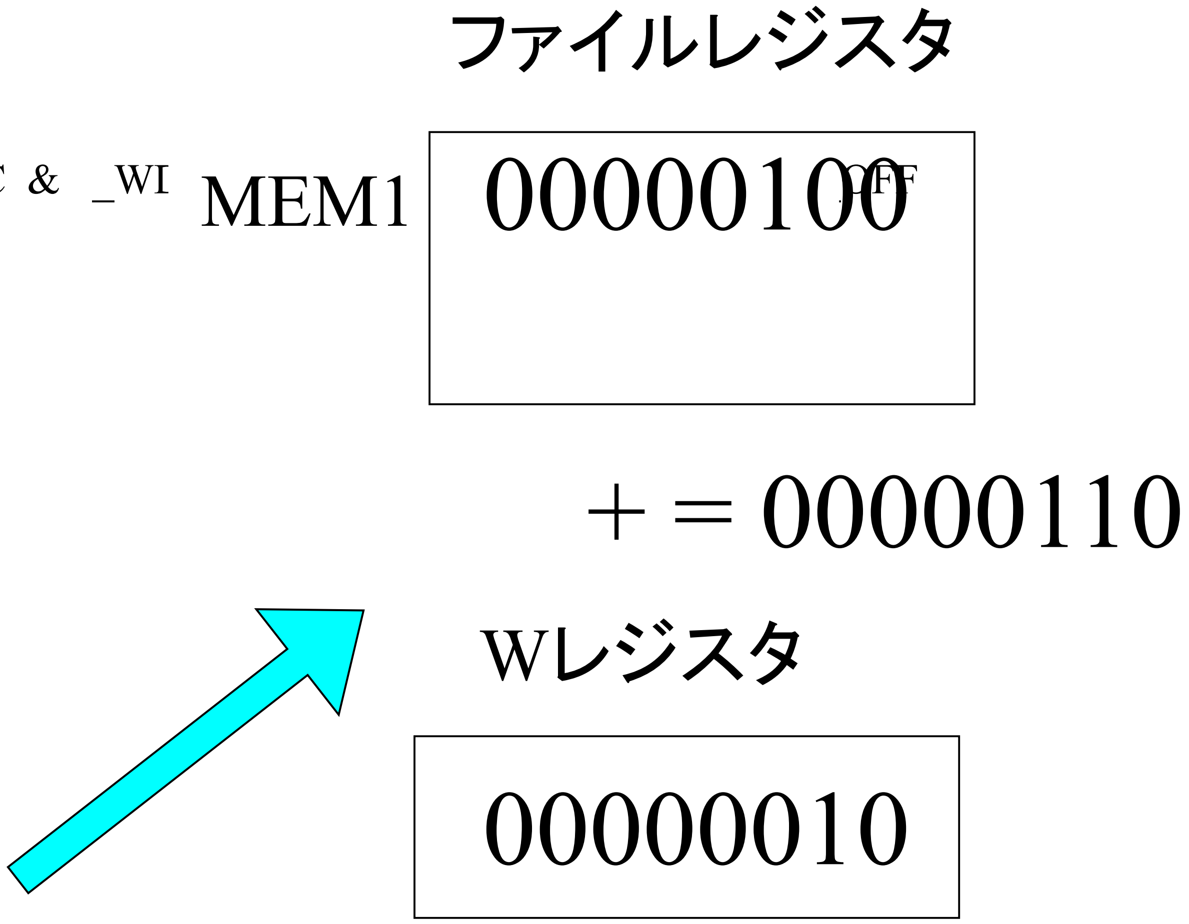
MEM1 | 00000100 (OFF)

Wレジスタ

00000010

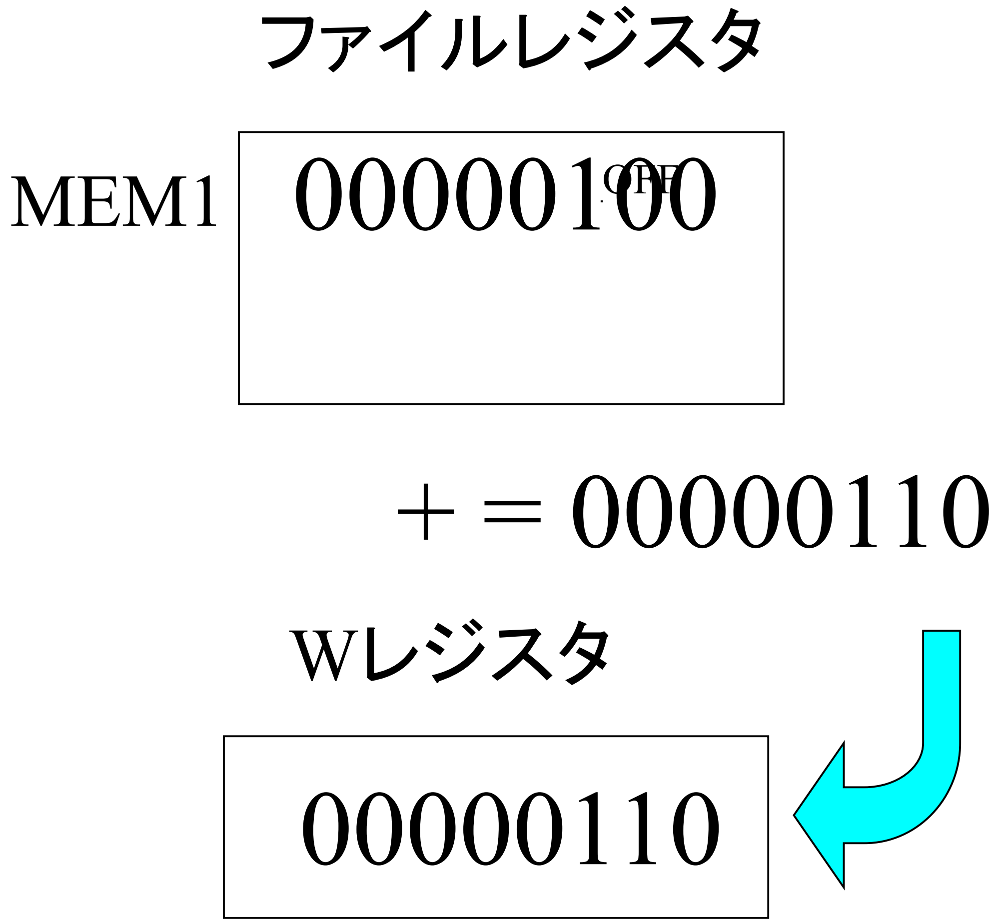
wレジスタに00000010
という数をロードせよと
いう命令

```
; ADD

        INCLUDE"p16F84A.inc"
        list p=16F84A

        __CONFIG   _HS_OSC  &  _WI

MEM1    EQU     0x0C

        ORG     0
        GOTO    START
        ORG     4

START
        MOVLW   B‘00000001

        ADDLW   B'00000011’

        MOVWF   MEM1

        MOVLW   B'00000010’

        ADDWF   MEM1, 0

        ADDWF   MEM1, 1

        GOTO    START

        END
```



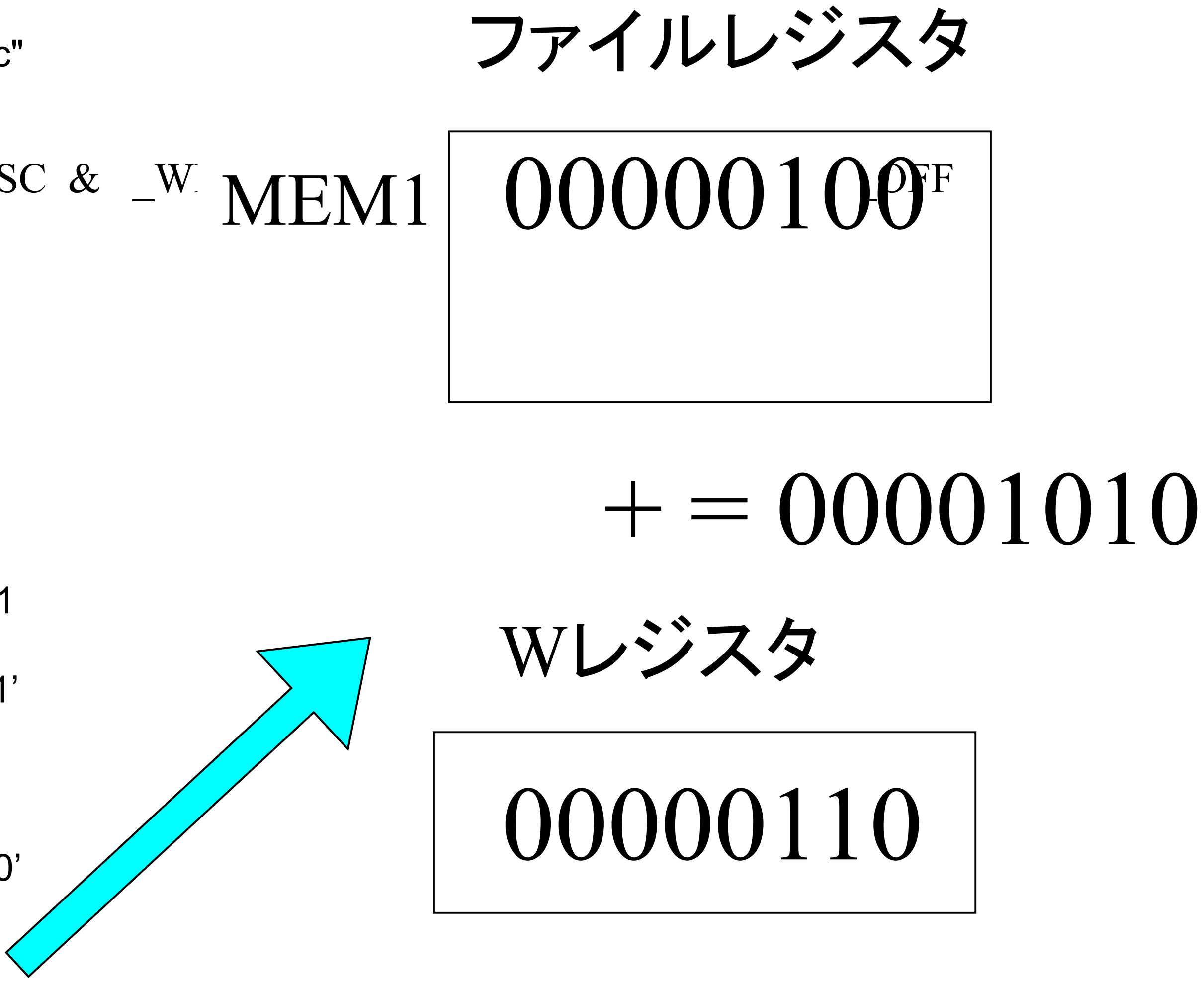
wレジスタの内容とファイルレジスタの内容を足してその結果をwレジスタに転送せよという命令

```
; ADD

        INCLUDE"p16F84A.inc"
        list p=16F84A

        __CONFIG  _HS_OSC  &  _WI

MEM1    EQU     0x0C

        ORG     0
        GOTO    START
        ORG     4

START
        MOVLW   B'00000001

        ADDLW   B'00000011'

        MOVWF   MEM1

        MOVLW   B'00000010'

        ADDWF   MEM1, 0

        ADDWF   MEM1, 1

        GOTO    START

        END
```



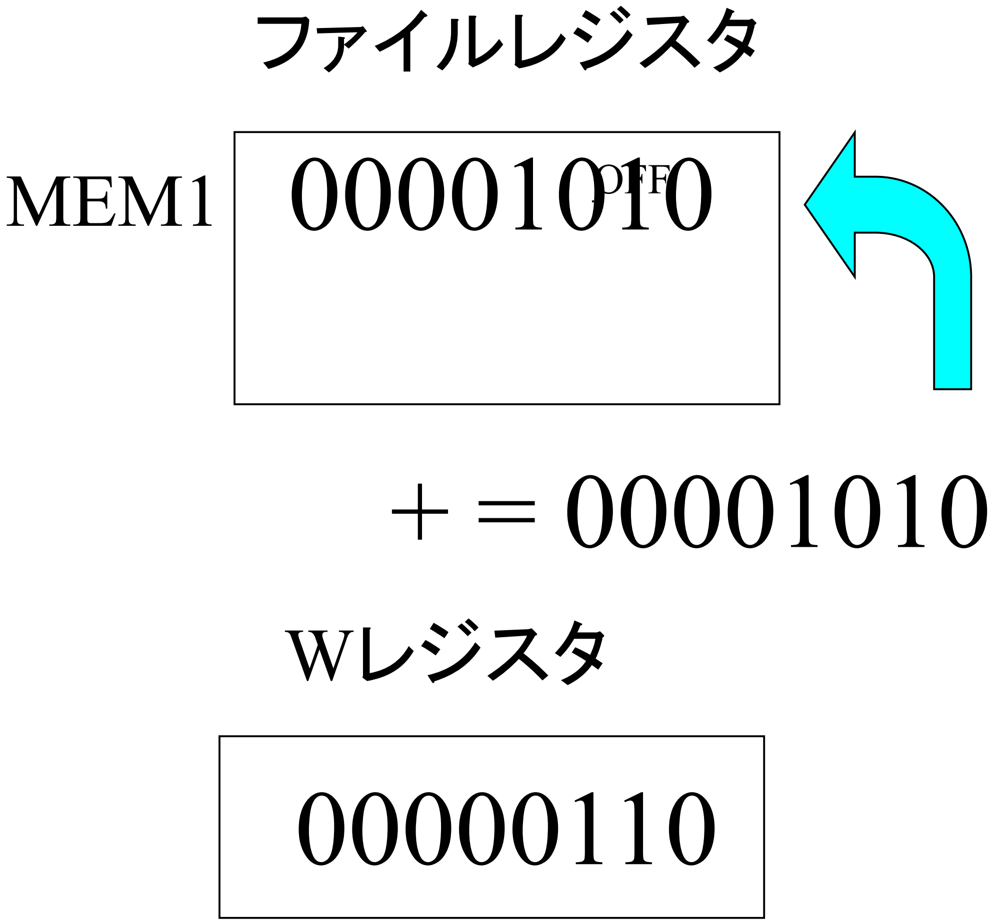
0は行き先がwレジスタであることを示す.

```
; ADD

        INCLUDE"p16F84A.inc"
        list p=16F84A

        __CONFIG  _HS_OSC  &  _W

MEM1    EQU     0x0C

        ORG     0
        GOTO    START
        ORG     4

START
        MOVLW   B‘00000001

        ADDLW   B'00000011’

        MOVWF   MEM1

        MOVLW   B'00000010’

        ADDWF   MEM1, 0

        ADDWF   MEM1, 1

        GOTO    START

        END
```



wレジスタの内容とファイルレジスタの内容を足してその結果をファイルレジスタに転送せよという命令

```
; ADD

        INCLUDE"p16F84A.inc"
        list p=16F84A

        __CONFIG  _HS_OSC & _WI

MEM1    EQU     0x0C

        ORG     0
        GOTO    START
        ORG     4

START
        MOVLW   B‘00000001

        ADDLW   B'00000011’

        MOVWF   MEM1

        MOVLW   B'00000010’

        ADDWF   MEM1, 0

        ADDWF   MEM1, 1

        GOTO    START

        END
```



1は行き先がファイルレジスタ（この場合はMEM1）であることを示す．

```
; ADD

        INCLUDE"p16F84A.inc"
        list p=16F84A

        __CONFIG   _HS_OSC  &   _WI

MEM1    EQU     0x0C

        ORG     0
        GOTO    START
        ORG     4

START
        MOVLW   B‘00000001

        ADDLW   B'00000011’

        MOVWF   MEM1

        MOVLW   B'00000010’

        ADDWF   MEM1, 0

        ADDWF   MEM1, 1

        GOTO    START

        END
```

STARTへジャンプする.

ファイルレジスタ

MEM1 00001010

Wレジスタ

00000110

演習問題 2. MEM1に0xFFを転送し，これに0x01を足し，結果をMEM2に転送するプログラムを作成せよ.

演習問題 3. Wに0x01を転送し，0x02からWの内容を引いて(SUBLW命令)，結果をMEM1に転送せよ.

SUBLW 0x02

0x02からwレジスタの内容を引いてその結果をwレジスタに転送せよという命令です.
Sub W from Literalの略です.

演習問題 4. MEM1に0x01を転送し，これから0x02を引いて，結果をMEM1に転送せよ．また，結果が２の補数表現の-1となっていることを確認せよ．

2の補数表現とは

10進数	2進数 (2の補数表現)
3	00000011
2	00000010
1	00000001
0	00000000
−1	11111111
−2	11111110
−3	11111101

10進数において2の符号を反転させて-2とすることは，2の補数表現では

```
  00000010      1/0を反転させる
  11111101
+        1      1を足す
----------
  11111110
```

とすることで得られる．

10進数における 2 と -1 の足し算は

$$2 + (-1) = 1$$

となる．2の補数表現では

```
 00000010
+11111111
---------
 00000001
```

となる．

10進数における 1 から 3 を引く計算は，まず

```
   00000011
               1/0を反転させる
   11111100
+         1    1を足す
-----------
   11111101
```

と，-3 に対応する 2 の補数表現を求める．つぎに 1 と -3 を足して

```
   00000001
+  11111101
-----------
   11111110
```

と実行されます．

２の補数表現を用いれば，引き算は1/0反転と足し算により実行されます．

演習問題 5. Increment , Decrement命令の動作を確認するプログラムを作成せよ.

INCF　f, d

ファイルレジスタの f に 1 を加えた結果をdに転送する命令です. d = 0のとき転送先はwレジスタ, d=1のとき転送先は f レジスタです.
Increment fileの略です.

例）INCF　　MEM1, 1

ファイルレジスタの MEM1 に 1 を加えた結果をMEM1に転送する命令です.

DECF　MEM1,1

ファイルレジスタのMEM1から 1 を引いた結果をMEM1に転送する命令です.
Decrement fileの略です.

演習問題 6. Complement命令を用いて，２の補数を求めるプログラムを作成せよ.

COMF f, d

ファイルレジスタの f の値を1/0反転した結果をdに転送する命令です. d = 0のとき転送先はwレジスタ, d=1のとき転送先は f レジスタです.
Complement fileの略です.

演習問題２． 解答例

MEM1に0xFFを転送し，これに0x01を足し，結果をMEM2に転送するプログラムを作成せよ．

```
; Problem 2
          INCLUDE"p16F84A.inc"
          list p=16F84A

          __CONFIG  _HS_OSC & _WDT_OFF & _PWRTE_OFF & _CP_OFF

MEM1      EQU       0x0C                ;MEM1 at 0C
MEM2      EQU       0x0C+1              ;MEM1 at 0D

          ORG       0
          GOTO      START
          ORG       4

START
          MOVLW     0xFF                ; Load 0xFF to W
          MOVWF     MEM1                ; Move W to MEM1
          ADDLW     0x01                ; W + 0x01 ->W
          MOVWF     MEM2                ;Move W to MEM2
          GOTO      START

          END
```

演習問題３．解答例

Wに0x01をロードし，0x02からWの内容を引いて(SUBLW命令)，結果をMEM1に転送せよ．

```
; Problem 3

        INCLUDE"p16F84A.inc"
        list p=16F84A

        __CONFIG  _HS_OSC  &  _WDT_OFF  &  _PWRTE_OFF  &  _CP_OFF

MEM1    EQU     0x0C                ;MEM1 at 0C

        ORG     0
        GOTO    START
        ORG     4

START
        MOVLW   0x01                ; Load 0x01 to W
        SUBLW   0x02                ; 0x02 - W ->W
        MOVWF   MEM1                ; Move W to MEM1

        GOTO    START

        END
```

演習問題４． 解答例

MEM1に0x01を転送し，これから0x02を引いて，結果をMEM1に転送せよ．
また，結果が２の補数表現の−1となっていることを確認せよ．

```
; Problem 4

        INCLUDE"p16F84A.inc"
        list p=16F84A

        __CONFIG  _HS_OSC  &  _WDT_OFF  &  _PWRTE_OFF  &  _CP_OFF

MEM1    EQU     0x0C                ;MEM1 at 0C

        ORG     0
        GOTO    START
        ORG     4

START
        MOVLW   0x01                ; Load 0x01 to W
        MOVWF   MEM1                ; Move W to MEM1
        MOVLW   0x02                ; Load 0x02 to W
        SUBWF   MEM1, 1             ;  MEM1- W -> MEM1

        GOTO    START

        END
```

演習問題５．解答例

Increment , Decrement命令の動作を確認するプログラムを作成せよ．

```
; Problem 5
            INCLUDE"p16F84A.inc"
            list p=16F84A

            __CONFIG  _HS_OSC & _WDT_OFF & _PWRTE_OFF & _CP_OFF

MEM1        EQU         0x0C                    ;MEM1 at 0C

            ORG         0
            GOTO        START
            ORG         4

START
            INCF        MEM1,1                  ;MEM1 + 1 -> MEM1
            INCF        MEM1,0                  ;MEM1 + 1 -> W
            DECF        MEM1,1                  ;MEM1 - 1 -> MEM1
            DECF        MEM1,0                  ;MEM1 -1 -> W

            GOTO        START

            END
```

演習問題６．解答例

Complement命令を用いて，２の補数を求めるプログラムを作成せよ.

```
; Problem 6

        INCLUDE"p16F84A.inc"
        list p=16F84A

        __CONFIG  _HS_OSC & _WDT_OFF & _PWRTE_OFF & _CP_OFF

MEM1    EQU     0x0C            ;MEM1 at 0C

        ORG     0
        GOTO    START
        ORG     4

START
        MOVLW   0x01            ; Load 0x01 to W
        MOVWF   MEM1            ; Move W to MEM1

        COMF    MEM1,1          ;Complement MEM1 -> MEM1
        INCF    MEM1,1          ;MEM1 + 1 -> MEM1

        GOTO    START
        END
```

２の補数を求める計算

2004年8月

著者：　古橋武
名古屋大学工学研究科計算理工学専攻
furuhashi@cse.nagoya-u.ac.jp