2. 転送命令を学ぼう

2004年8月に本講義ノートをWebにアップして以来，とても多くの方の訪問を受けてきました．内容が一部古くなっていたので，2012年5月時点の情報に書き改めました．主な変更点は以下の通りです．

第0章に本講座の準備のための章を設け，以下の更新をしました．

1．プログラム開発環境(MPLAB IDE)をv8.84に更新しました．
2012年5月時点での最新バージョンはMPLAB X IDE v1.10ですが，アセンブラの勉強のためには，旧バージョンのv8.84が良さそうです．（筆者がXを使いこなしていないこともあります．）
2．ブレッドボード上に回路を製作し，その詳細を記しました．
ハンダ付けをほとんど必要とせずにマイコン回路を製作できます．
3．部品を全てネットで購入して，その仕様と入手先を記しました．
入手が容易な部品ばかりでマイコン回路を製作しました．
4．In-Circuit Debugger/ProgrammerにPICkit3を用いました．
ブレッドボード上のマイコンへのプログラミングにPICkit3を使用した例を記しました．

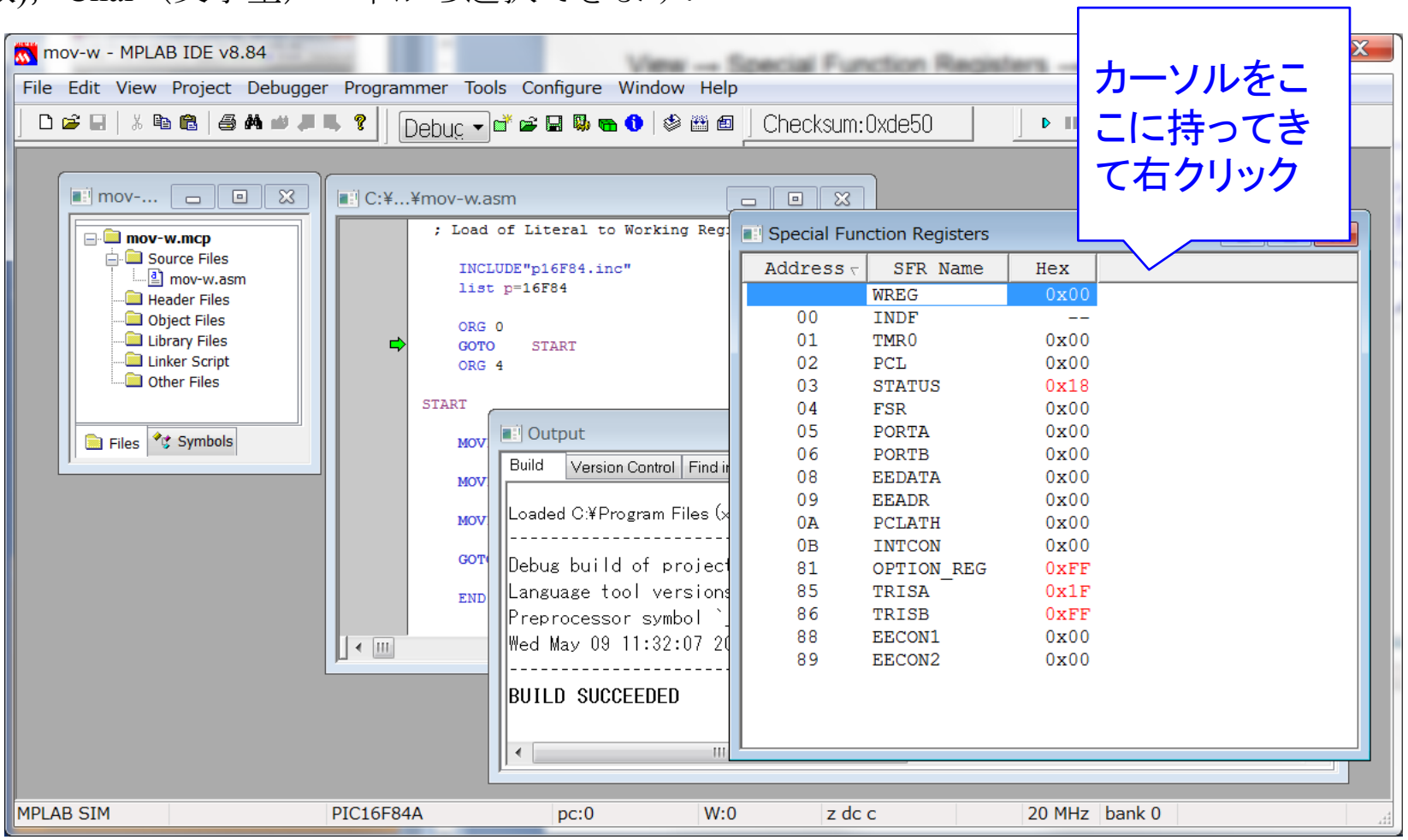
まず，このソースファイルを打ち込んで下さい．
MPLAB IDE v8.84の立ち上げ方，Project Fileの作り方，アセンブラのソースファイルの作り方，ビルドの仕方，シミュレーションの仕方は第0章を参照してください．　注意 CONFIGの前のアンダーバーは2つあります．

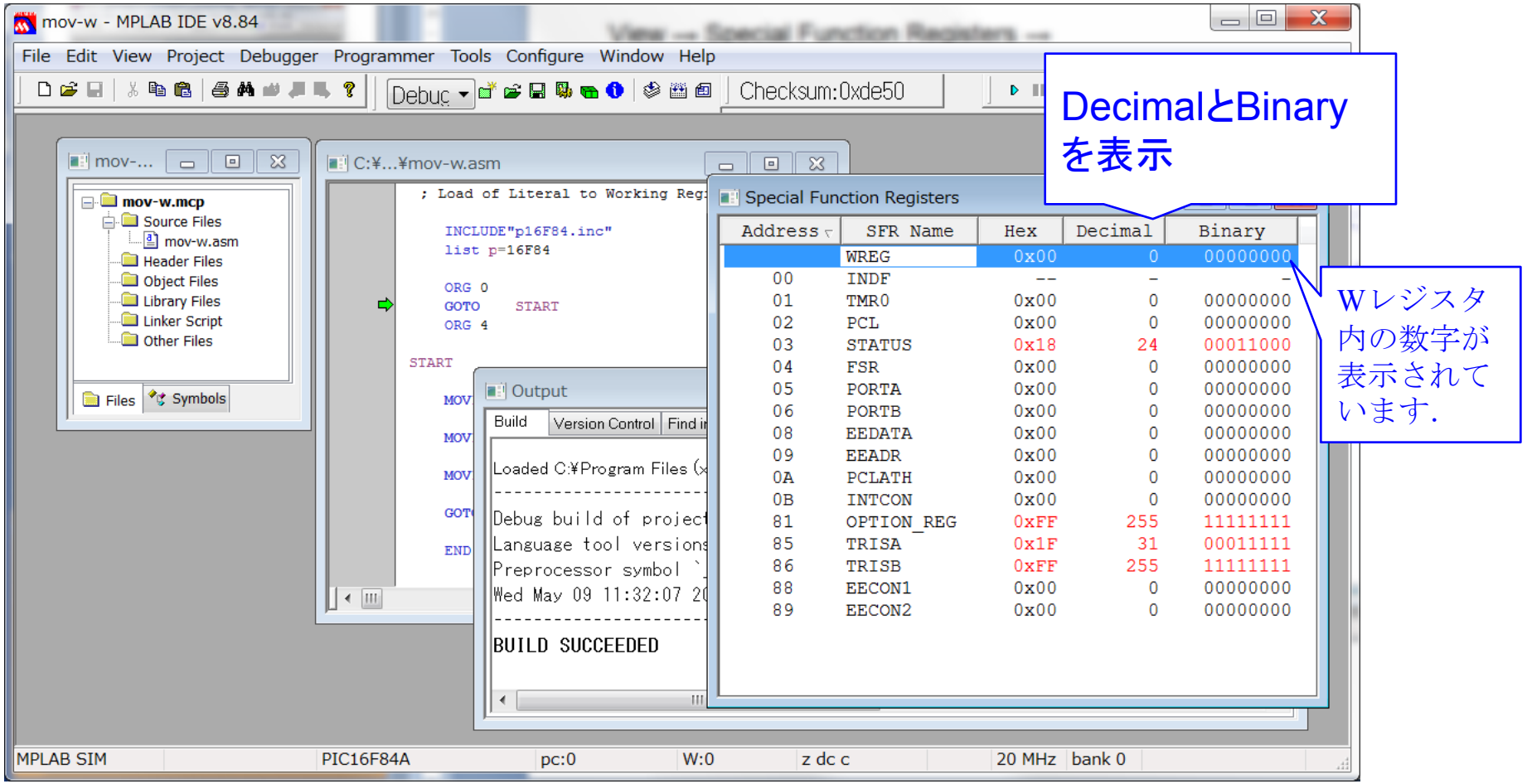
```
; Load of Literal to Working Register

        INCLUDE"p16F84A.inc"
        list p=16F84A

        __CONFIG  _HS_OSC  &  _WDT_OFF  &  _PWRTE_OFF  &  _CP_OFF

        ORG     0
        GOTO    START
        ORG     4

START

        MOVLW           B'10011010'         ; Binary number

        MOVLW           0xAB                ; Hexadecimal number

        MOVLW           D'255'              ; Decimal number

        GOTO            START

        END
```

これはWレジスタにLiterarl（リテラル，文字）を転送させるプログラムです．最初に2進数の10011010を転送し，次に16進数のABを転送し，最後に10進数の255を転送します．
MOVLWはMove Literal to Working registerの略です．

シミュレーションの仕方

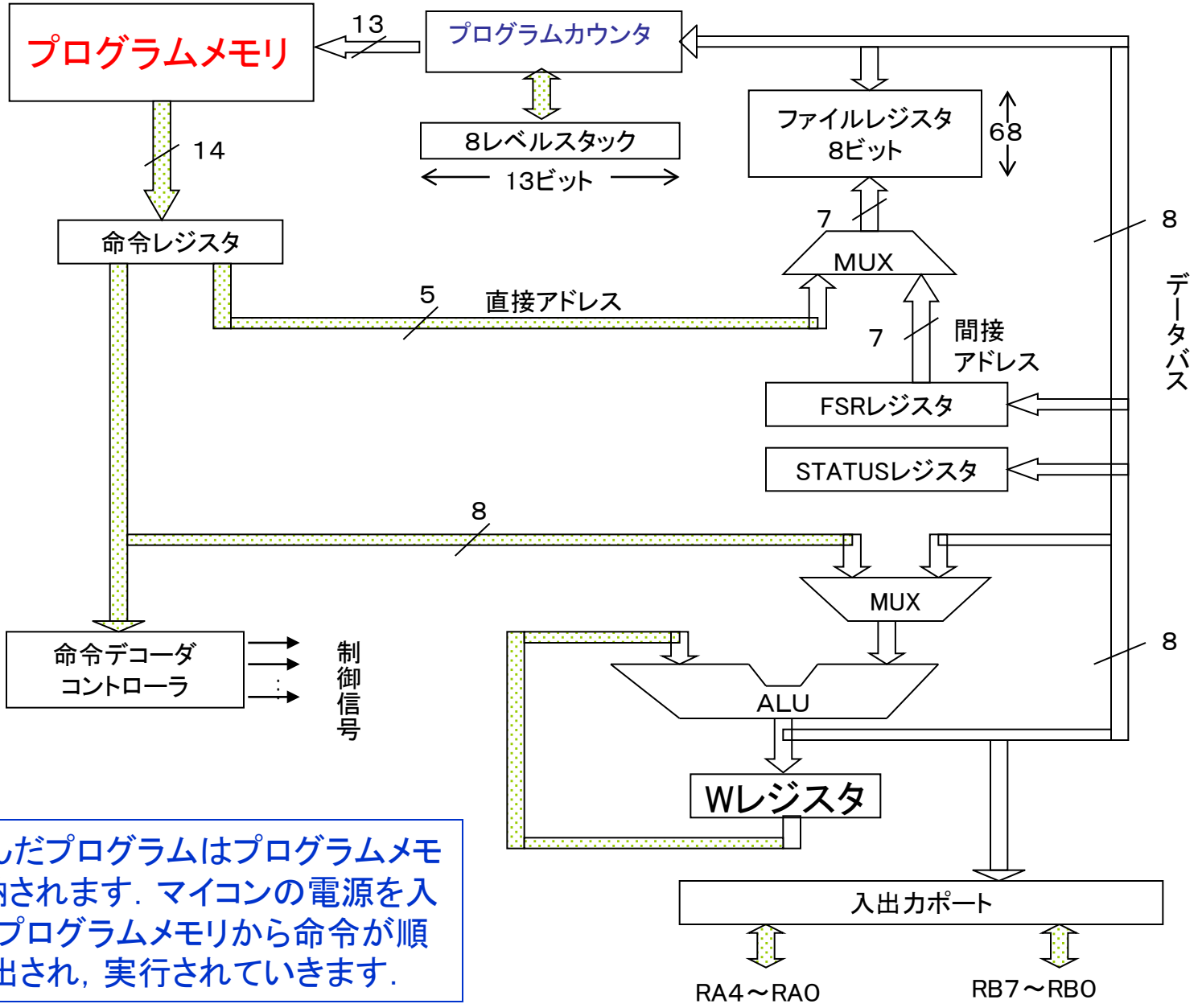
プログラムを打ち込み終わったら，Debugger → Select Tool → MPLAB SIM → Project → Makeと選択すると，プログラムに文法上の間違いがなければ，BUILD SUCCEEDEDというメッセージが出ます．次にView → Special Function Registersを選択して下さい．下図のようにレジスタ群の内容を示すウィンドウが開かれます．図示の位置にカーソルを持ってきて右クリックすると，レジスタ内容の表示型をHex(16進数)，Binary(2進数)，Decimal(10進数)，Char（文字型）の中から選択できます．





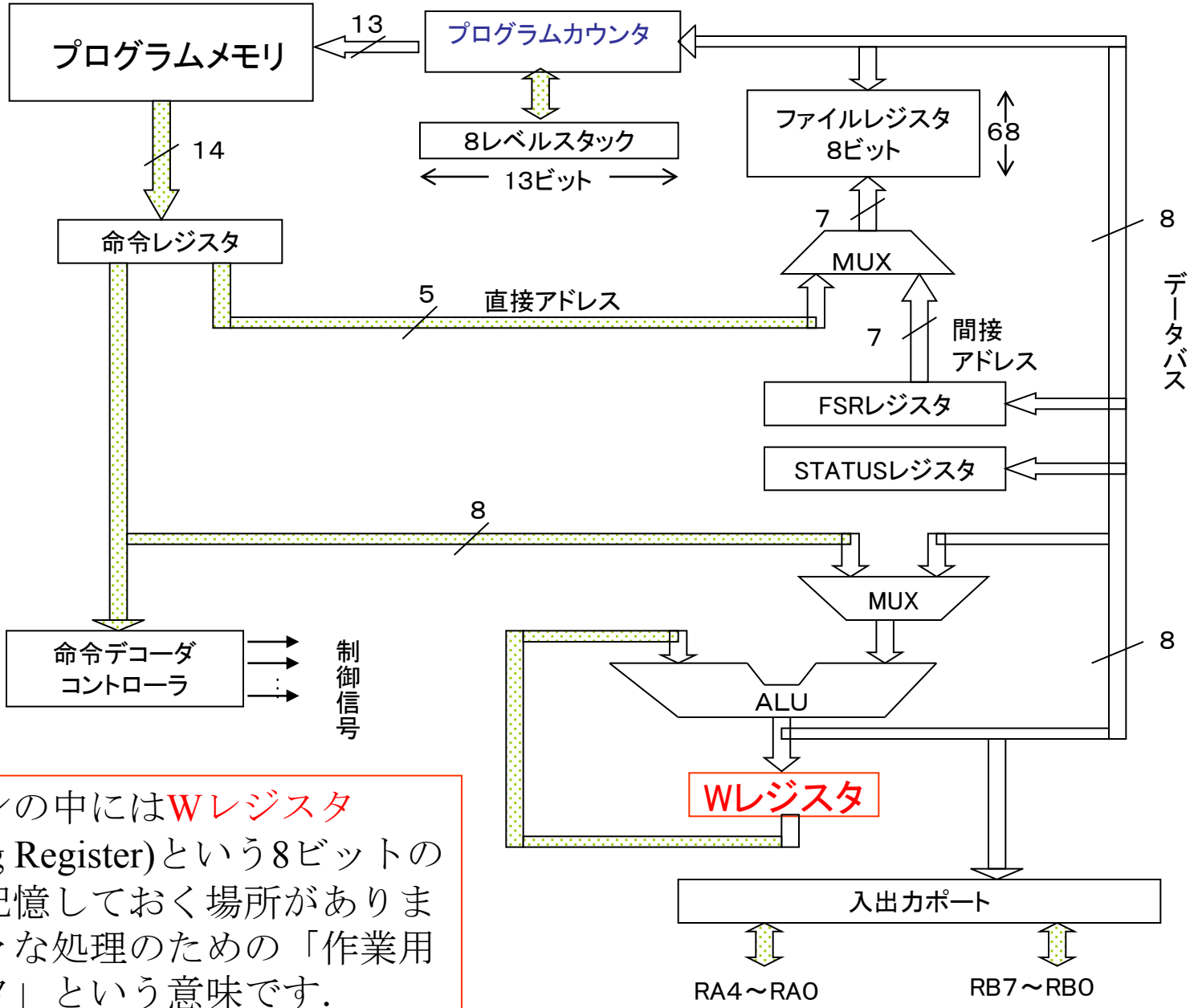
ファンクションキーのF7を押して，プログラムを一行ずつ実行して，WREG (Wレジスタ）の中の数字がどう書き換えられていくかを確認してください.

PIC16F84AのData SheetのBLOCK DIAGRAMの抜粋です.



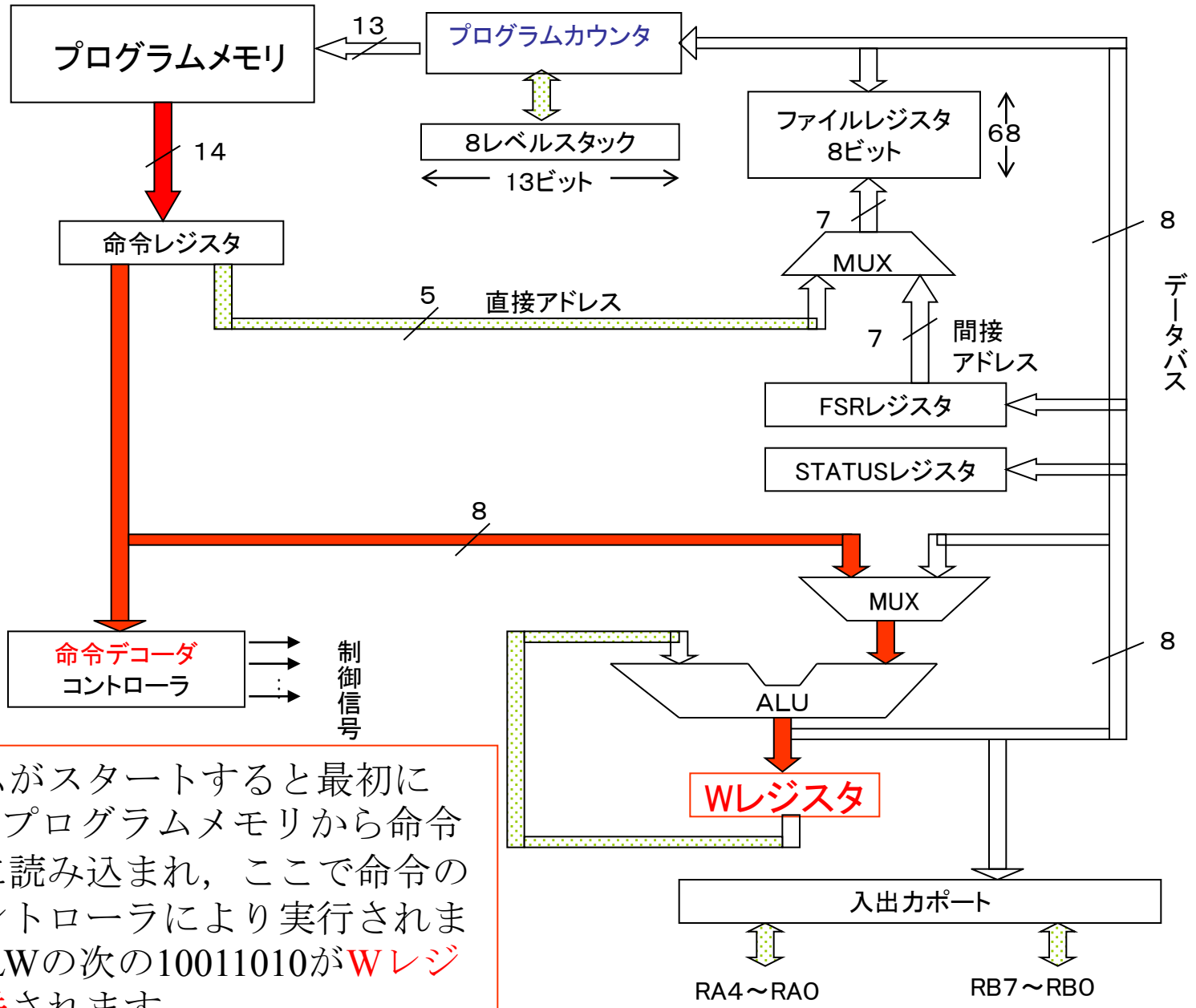
打ち込んだプログラムはプログラムメモリに格納されます. マイコンの電源を入れると, プログラムメモリから命令が順次読み出され, 実行されていきます.

PIC16F84AのData SheetのBLOCK DIAGRAMの抜粋です.



マイコンの中にはWレジスタ(Working Register)という8ビットの数字を記憶しておく場所があります．様々な処理のための「作業用レジスタ」という意味です.

MOVLW　B‘10011010’　の実行



プログラムがスタートすると最初にMOVLWがプログラムメモリから命令デコーダに読み込まれ，ここで命令の解読，コントローラにより実行されます．MOVLWの次の10011010がWレジスタに転送されます．

```
; Load of Literal to Working Register

        INCLUDE"p16F84A.inc"
        list p=16F84A

        __CONFIG  _HS_OSC  &  _WDT_OFF  &  _PWRTE_OFF  &  _CP_OFF

        ORG     0
        GOTO    START
        ORG     4

START

        MOVLW           B'10011010'

        MOVLW           0xAB

        MOVLW           D'255'

        GOTO            START

        END
```

Configuration設定です．マイコンのシステムクロックを設定します．

MicrochipのWebページからダウンロードできるPIC16F84AのData Sheetの21ページにConfigurationの説明があります．

OSC: Oscillator

RC: Resistor/Capacitor
内蔵の発振回路，4MHz程度まで
発信周波数は不安定

HS: High Speed Crystal/Resonator
3.5MHz 以上で推奨
本回路では10MHzのセラミック発振子を外付けしてあるので，HS設定とする．

XT: Crystal/Resonator
4 MHz 以下

LP: Low Power Crystal
低消費電力, 200kHz 以下

WDT: Watch Dog Timer

PWRTE: Power up Timer Enable

CP: Code Protection

本資料の使い方の範囲内では全てオフで差し支えない．

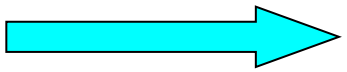
```
; Load of Literal to Working Register

        INCLUDE"p16F84A.inc"
        list p=16F84A

        __CONFIG   _HS_OSC  &   _WDT_OFF  &   _PWRTE_OFF  &   _CP_OFF

        ORG     0
        GOTO    START
        ORG     4

START

        MOVLW           B'10011010'

        MOVLW           0xAB

        MOVLW           D'255'

        GOTO            START

        END
```

現実のマイコンでは電源を入れるとここから始まります.

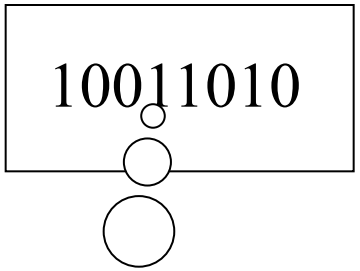
STARTへジャンプする.

```
; Load of Literal to Working Register

        INCLUDE"p16F84A.inc"
        list p=16F84A

        __CONFIG  _HS_OSC  &  _WDT_OFF  &  _PWRTE_OFF  &  _CP_OFF

        ORG     0
        GOTO    START
        ORG     4

START

        MOVLW           B'10011010'

        MOVLW           0xAB

        MOVLW           D'255'

        GOTO            START

        END
```

wレジスタに100110101という数を転送せよという命令です.



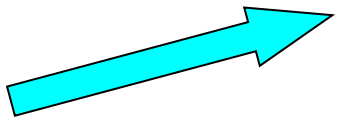
Wレジスタ



B’10011010’　2進数は
= D‘154’　10進数
= H‘9A’　16進数です.
表現が違うだけでどれも同じ数字です.

```
; Load of Literal to Working Register

        INCLUDE"p16F84A.inc"
        list p=16F84A

        __CONFIG  _HS_OSC  &  _WDT_OFF  &  _PWRTE_OFF  &  _CP_OFF

        ORG     0
        GOTO    START
        ORG     4

START

        MOVLW           B'10011010'

        MOVLW           0xAB

        MOVLW           D'255'

        GOTO            START

        END
```

wレジスタにABという数を転送せよという命令です.



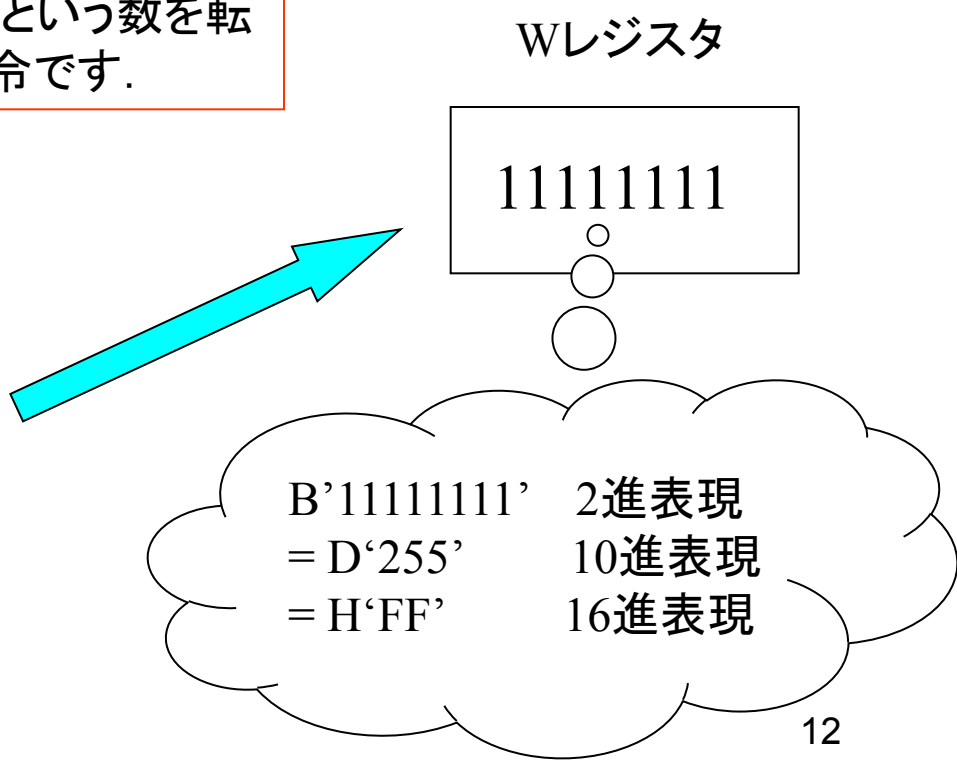
Wレジスタ

10101011

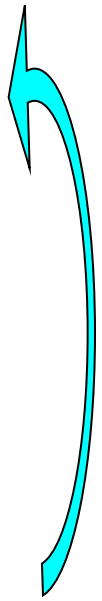
B’10101011’　2進表現
= D‘171’　10進表現
= H‘AB’　16進表現

```
; Load of Literal to Working Register

        INCLUDE"p16F84A.inc"
        list p=16F84A

        __CONFIG  _HS_OSC  &  _WDT_OFF  &  _PWRTE_OFF  &  _CP_OFF

        ORG     0
        GOTO    START
        ORG     4

START

        MOVLW           B'10011010'

        MOVLW           0xAB

        MOVLW           D'255'

        GOTO            START

        END
```

wレジスタに255という数を転送せよという命令です.



```
; Load of Literal to Working Register

        INCLUDE"p16F84A.inc"
        list p=16F84A

        __CONFIG  _HS_OSC  &  _WDT_OFF  &  _PWRTE_OFF  &  _CP_OFF

        ORG     0
        GOTO    START
        ORG     4

START

        MOVLW           B'10011010'

        MOVLW           0xAB

        MOVLW           D'255'

        GOTO            START

        END
```

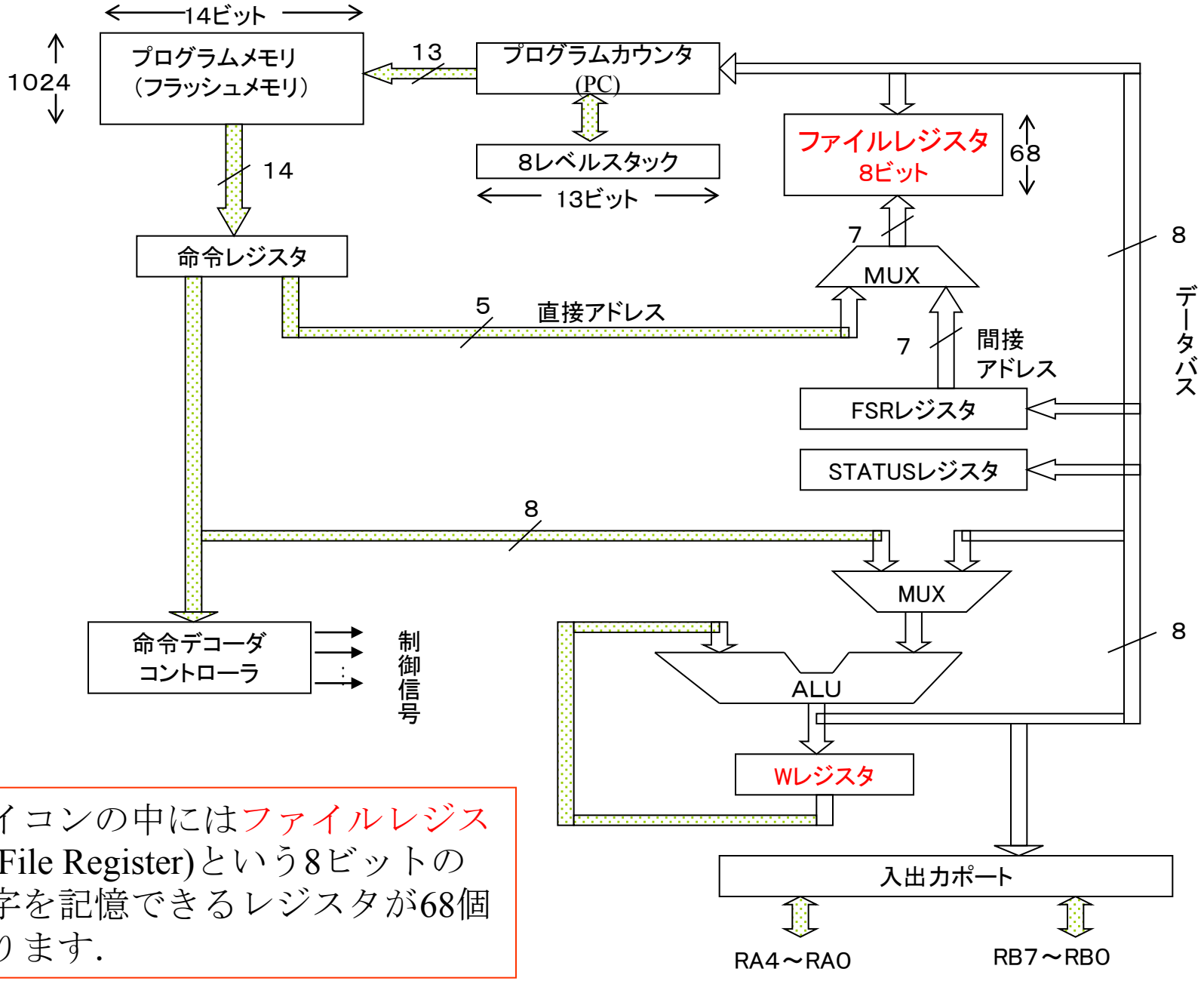


Wレジスタ

11111111

STARTへジャンプする.

次に，ワーキング（W）レジスタとファイルレジスタ間でのデータのやりとりを体験してください．



マイコンの中にはファイルレジスタ(File Register)という8ビットの数字を記憶できるレジスタが68個あります.

PIC16F84Aの構成

0C～4Fの番地を持つ68個の８ビットレジスタがある.

8ビットの数字を68個までしまっておける「引き出し」のようなもの.

データの「一時保管所」のような意味です.

0x0C

0x4F

ファイルレジスタ

Wレジスタはたった１個の８ビットレジスタからなる.

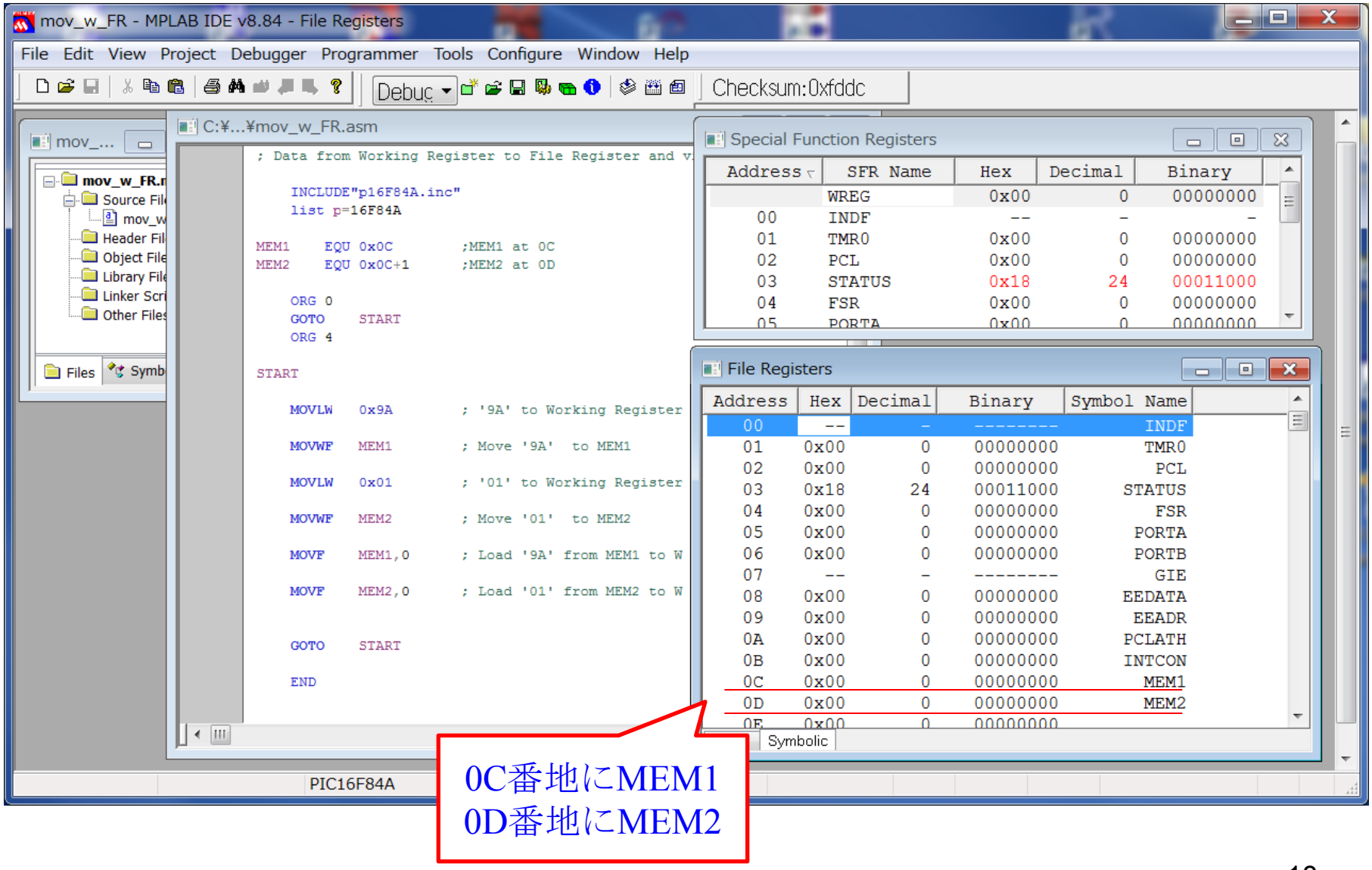
wレジスタ

```
; Data from Working Register to File Register and vice versa

            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              ;MEM2 at 0D

            ORG       0
            GOTO      START
            ORG       4

START
            MOVLW     0x9A                ; '9A' to Working Register

            MOVWF     MEM1                ; Move '9A'  to MEM1

            MOVLW     0x01                ; '01' to Working Register

            MOVWF     MEM2                ; Move '01'  to MEM2

            MOVF      MEM1,0              ; Load '9A' from MEM1 to W

            MOVF      MEM2,0              ; Load '01' from MEM2 to W


            GOTO      START

            END
```

このプログラムを打ち込んで下さい.

シミュレータでプログラムをステップ実行しながら，w, MEM1, MEM2レジスタの中の数字がどう変わるか確認して下さい.

MEM1, MEM2の中の数字は，View → File Registersにより見ることができます．



```
; Data from Working Register to File Register and vice versa

            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              ;MEM2 at 0D

            ORG       0
            GOTO      START
            ORG       4

START
            MOVLW     0x9A                ; '9A' to Working Register

            MOVWF     MEM1                ; Move '9A'  to MEM1

            MOVLW     0x01                ; '01' to Working Register

            MOVWF     MEM2                ; Move '01'  to MEM2

            MOVF      MEM1,0              ; Load '9A' from MEM1 to W

            MOVF      MEM2,0              ; Load '01' from MEM2 to W


            GOTO      START

            END
```

0C番地のファイルレジスタにMEM1という名前をつけ，0C+1 (= 0D)番地のファイルレジスタにMEM2という名前を付けます．

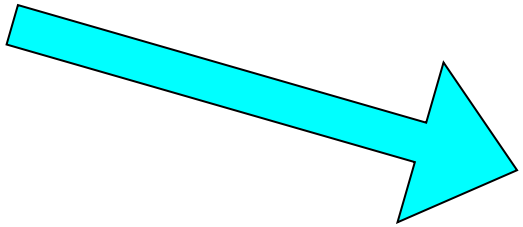
```
; Data from Working Register to File Register and vice versa

            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              ;MEM2 at 0D

            ORG         0
            GOTO        START
            ORG         4

START
            MOVLW       0x9A                ; '9A' to Working Register

            MOVWF       MEM1                ; Move '9A'  to MEM1

            MOVLW       0x01                ; '01' to Working Register

            MOVWF       MEM2                ; Move '01'  to MEM2

            MOVF        MEM1,0              ; Load '9A' from MEM1 to W

            MOVF        MEM2,0              ; Load '01' from MEM2 to W


            GOTO        START

            END
```

電源を入れるとORG 0の次から始まります.

STARTへジャンプする.

```
; Data from Working Register to File Register and vice versa

            INCLUDE"p16F84A.inc"
            list p=16F84A

            __CONFIG  _HS_OSC  &  _WDT_OFF  &  _PWRTE_OFF  &  _CP_OFF

MEM1        EQU       0x0C
MEM2        EQU       0x0C+1

            ORG       0
            GOTO      START
            ORG       4

START
            MOVLW     0x9A

            MOVWF     MEM1

            MOVLW     0x01

            MOVWF     MEM2

            MOVF      MEM1,0

            MOVF      MEM2,0


            GOTO      START

            END
```

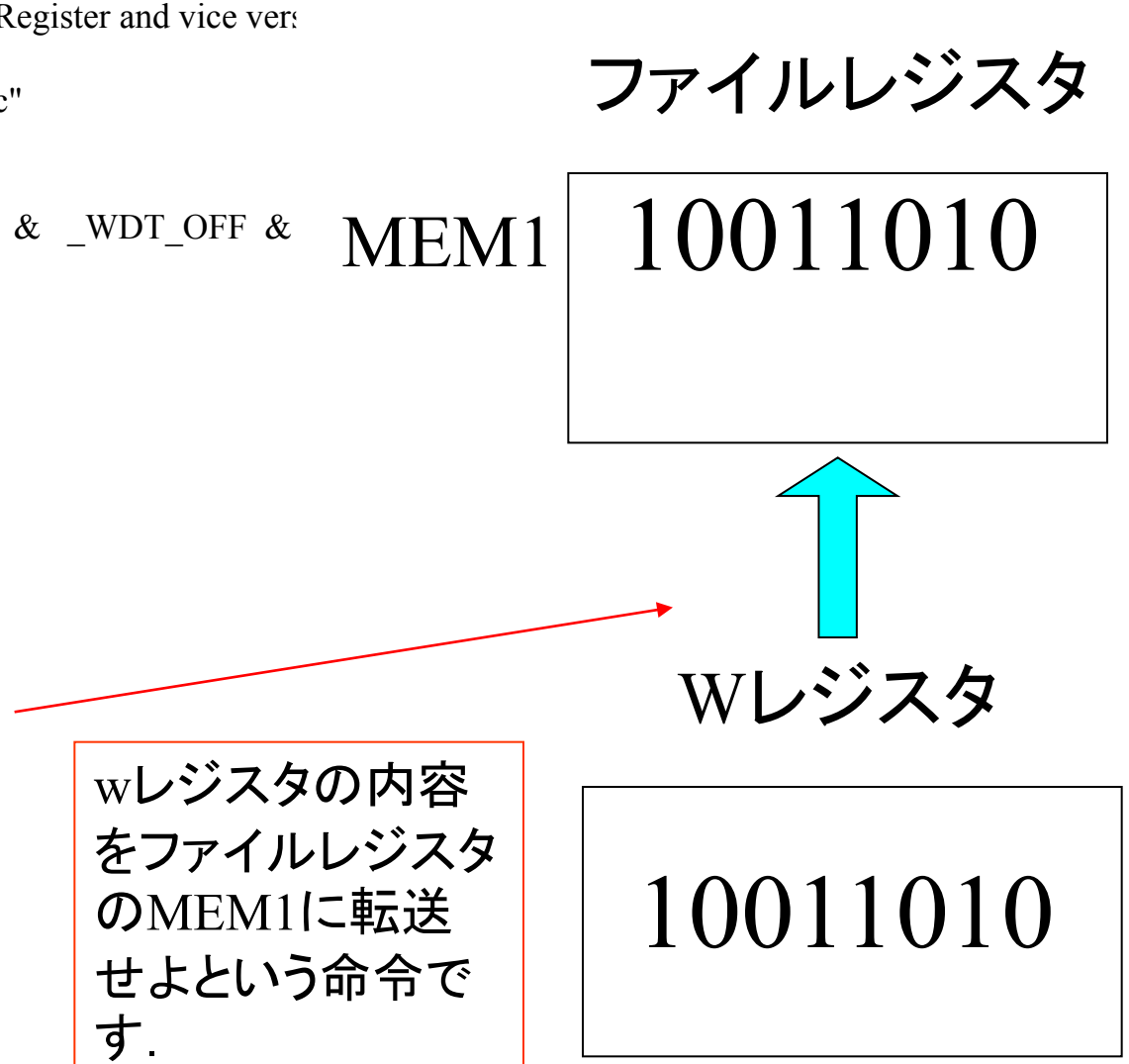
wレジスタに
100110101という数を
転送せよという命令



Wレジスタ

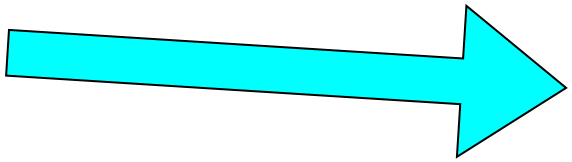
10011010

```
; Data from Working Register to File Register and vice ver
                INCLUDE"p16F84A.inc"
                list p=16F84A

                __CONFIG  _HS_OSC  &  _WDT_OFF  &

MEM1            EQU             0x0C
MEM2            EQU             0x0C+1

                ORG             0
                GOTO            START
                ORG             4

START
                MOVLW           0x9A

                MOVWF           MEM1

                MOVLW           0x01

                MOVWF           MEM2

                MOVF            MEM1,0

                MOVF            MEM2,0


                GOTO            START

                END
```



```
; Data from Working Register to File Register and vice ver
            INCLUDE"p16F84A.inc"
            list p=16F84A

            __CONFIG  _HS_OSC  &  _WDT_OFF  &

MEM1        EQU       0x0C
MEM2        EQU       0x0C+1

            ORG       0
            GOTO      START
            ORG       4

START
            MOVLW     0x9A

            MOVWF     MEM1

            MOVLW     0x01

            MOVWF     MEM2

            MOVF      MEM1,0

            MOVF      MEM2,0


            GOTO      START

            END
```

wレジスタに00000001という数を転送せよという命令



ファイルレジスタ

MEM1 | 10011010

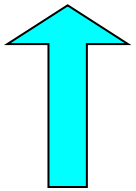
Wレジスタ

00000001

```
; Data from Working Register to File Register and vice ver
            INCLUDE"p16F84A.inc"
            list p=16F84A

            __ CONFIG  _HS_OSC  &  _WDT_OFF  &

MEM1        EQU        0x0C
MEM2        EQU        0x0C+1

            ORG        0
            GOTO       START
            ORG        4

START
            MOVLW      0x9A

            MOVWF      MEM1

            MOVLW      0x01

            MOVWF      MEM2

            MOVF       MEM1,0

            MOVF       MEM2,0


            GOTO       START

            END
```

ファイルレジスタ

MEM1	10011010
MEM2	00000001



Wレジスタ

00000001

wレジスタの内容をファイルレジスタのMEM2に転送せよという命令

```
; Data from Working Register to File Register and vice ver
            INCLUDE"p16F84A.inc"
            list p=16F84A

            __CONFIG  _HS_OSC  &  _WDT_OFF  &

MEM1        EQU       0x0C
MEM2        EQU       0x0C+1

            ORG       0
            GOTO      START
            ORG       4

START
            MOVLW     0x9A

            MOVWF     MEM1

            MOVLW     0x01

            MOVWF     MEM2

            MOVF      MEM1,0

            MOVF      MEM2,0


            GOTO      START

            END
```

ファイルレジスタ

MEM1 10011010

MEM2 00000001

Wレジスタ

10011010

MEM1の内容をwレジスタに転送せよという命令

MOVF　f, d
レジスタfの中身をdへ転送せよと言う命令
d＝0はWレジスタ

```
; Data from Working Register to File Register and vice ver
            INCLUDE"p16F84A.inc"
            list p=16F84A

            __CONFIG  _HS_OSC  &  _WDT_OFF  &

MEM1        EQU       0x0C
MEM2        EQU       0x0C+1

            ORG       0
            GOTO      START
            ORG       4

START
            MOVLW     0x9A

            MOVWF     MEM1

            MOVLW     0x01

            MOVWF     MEM2

            MOVF      MEM1,0

            MOVF      MEM2,0


            GOTO      START

            END
```

ファイルレジスタ

MEM1 10011010

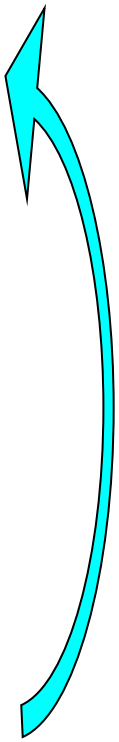
MEM2 00000001

Wレジスタ

00000001

MEM2の内容をwレジスタに転送せよという命令

```
; Data from Working Register to File Register and vice ver
            INCLUDE"p16F84A.inc"
            list p=16F84A

            __CONFIG  _HS_OSC  &  _WDT_OFF  &

MEM1        EQU       0x0C
MEM2        EQU       0x0C+1

            ORG       0
            GOTO      START
            ORG       4

START
            MOVLW     0x9A

            MOVWF     MEM1

            MOVLW     0x01

            MOVWF     MEM2

            MOVF      MEM1,0

            MOVF      MEM2,0


            GOTO      START

            END
```



STARTへジャンプする.

ファイルレジスタ

MEM1	10011010
MEM2	00000001

Wレジスタ

00000001

演習問題１．MEM1に0xAA，MEM2に0xBBを転送し，これらを交換するプログラムを作成せよ.

演習問題１．MEM1に0xAA，MEM2に0xBBを転送し，これらを交換するプログラムを作成せよ．

```
; Problem 1

            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                ;MEM2 at 0D
MEM3        EQU       0x0C+2                ;MEM3 at 0E

            ORG       0
            GOTO      START
            ORG       4

START

            MOVLW     0xAA                  ; ‘AA' to Working Register
            MOVWF     MEM1                  ; Move ‘AA' to MEM1
            MOVLW     0xBB                  ; ‘BB' to Working Register
            MOVWF     MEM2                  ; Move ‘BB'  to MEM2

            MOVF      MEM1, 0               ;Load MEM1 to W
            MOVWF     MEM3                  ;Move W to MEM3
            MOVF      MEM2,0                ;Load MEM2 to W
            MOVWF     MEM1                  ;Move W to MEM1
            MOVF      MEM3,0                ;Load MEM3 to W
            MOVWF     MEM2                  ;Move W to MEM2

            GOTO      START
            END
```

MEM1に0xAA，MEM2に0xBBを転送

MEM1の内容を一端MEM3に転送し，MEM2の内容をMEM1に移した後に，MEM3の内容をMEM2に転送

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