

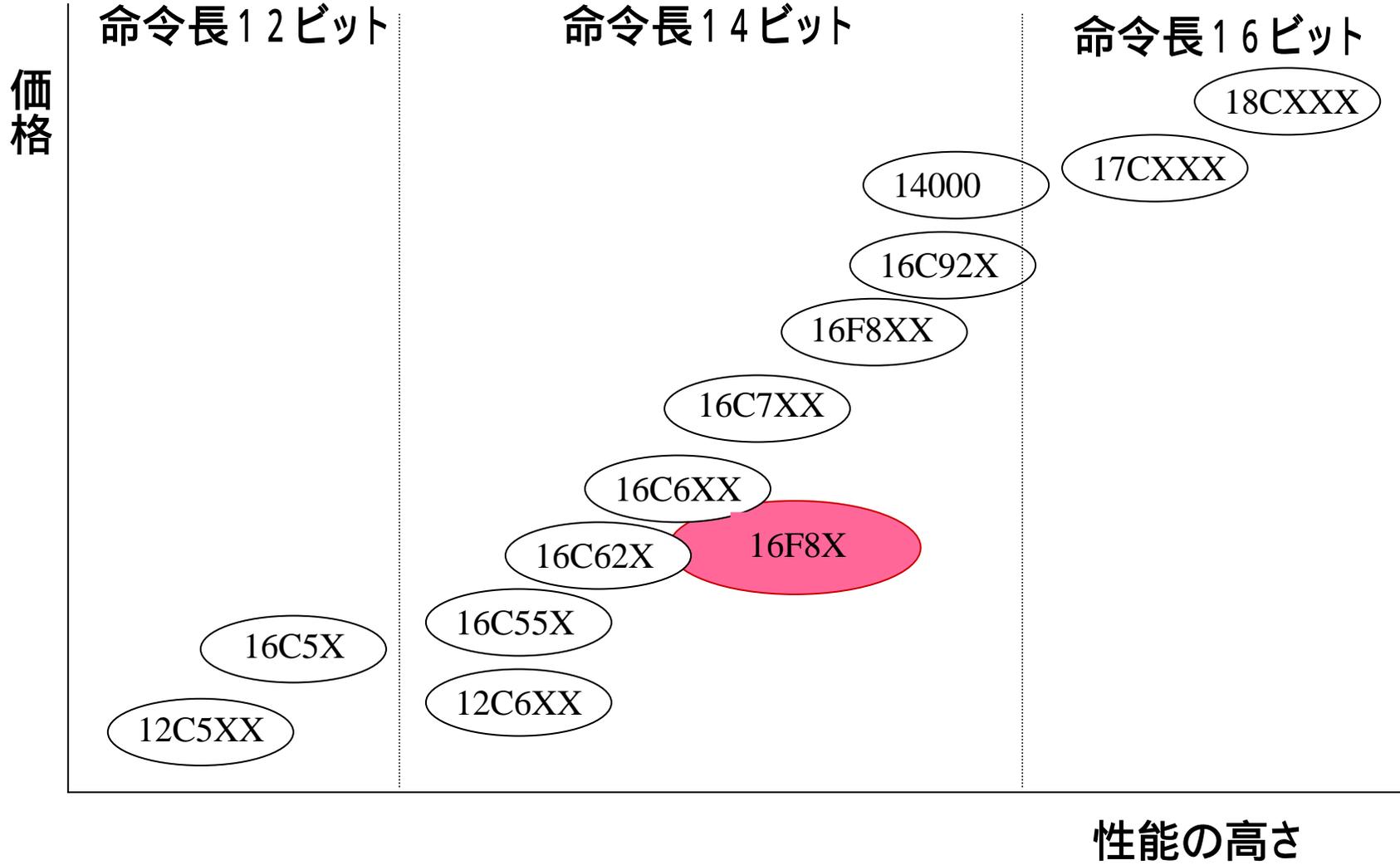
PICプログラム入門

東海北陸地区国立学校等技術職員合同研修

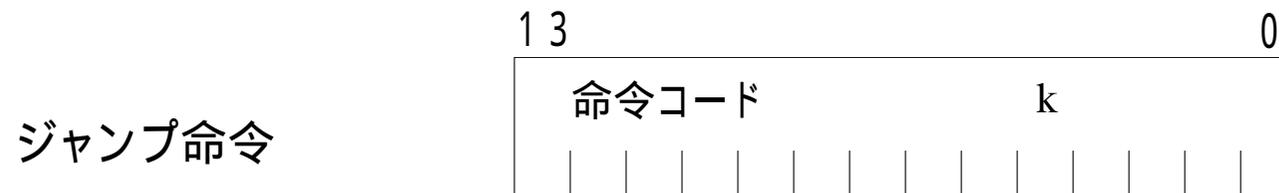
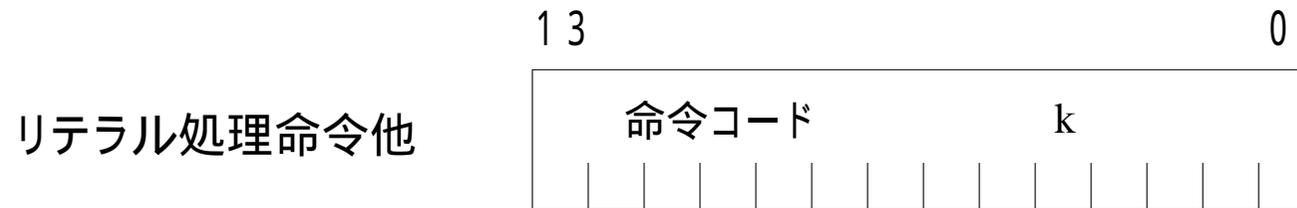
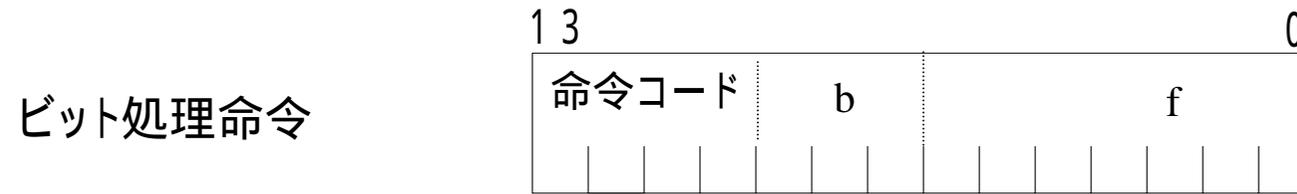
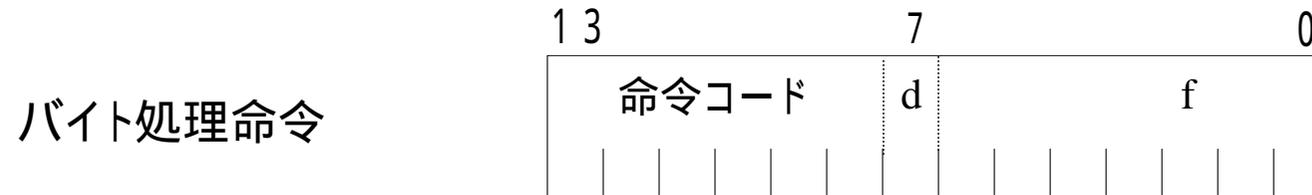
目次

- 1) Pic命令語の説明概略
- 2) レジスタの構成
- 3) プログラム (入力・出力の例題)
- 4) MPLABの使用法 (HEXデータ作成)
- 5) Picへの書き込み・動作チェック
- 6) 応用プログラムも作成
- 7) 実習

PIC一覧



命令アーキテクチャ



PIC製品一覧

表 1.1 PICmicroマイクロコントローラ製品ラインナップ(代表的なもの) (Microchip社カタログより)

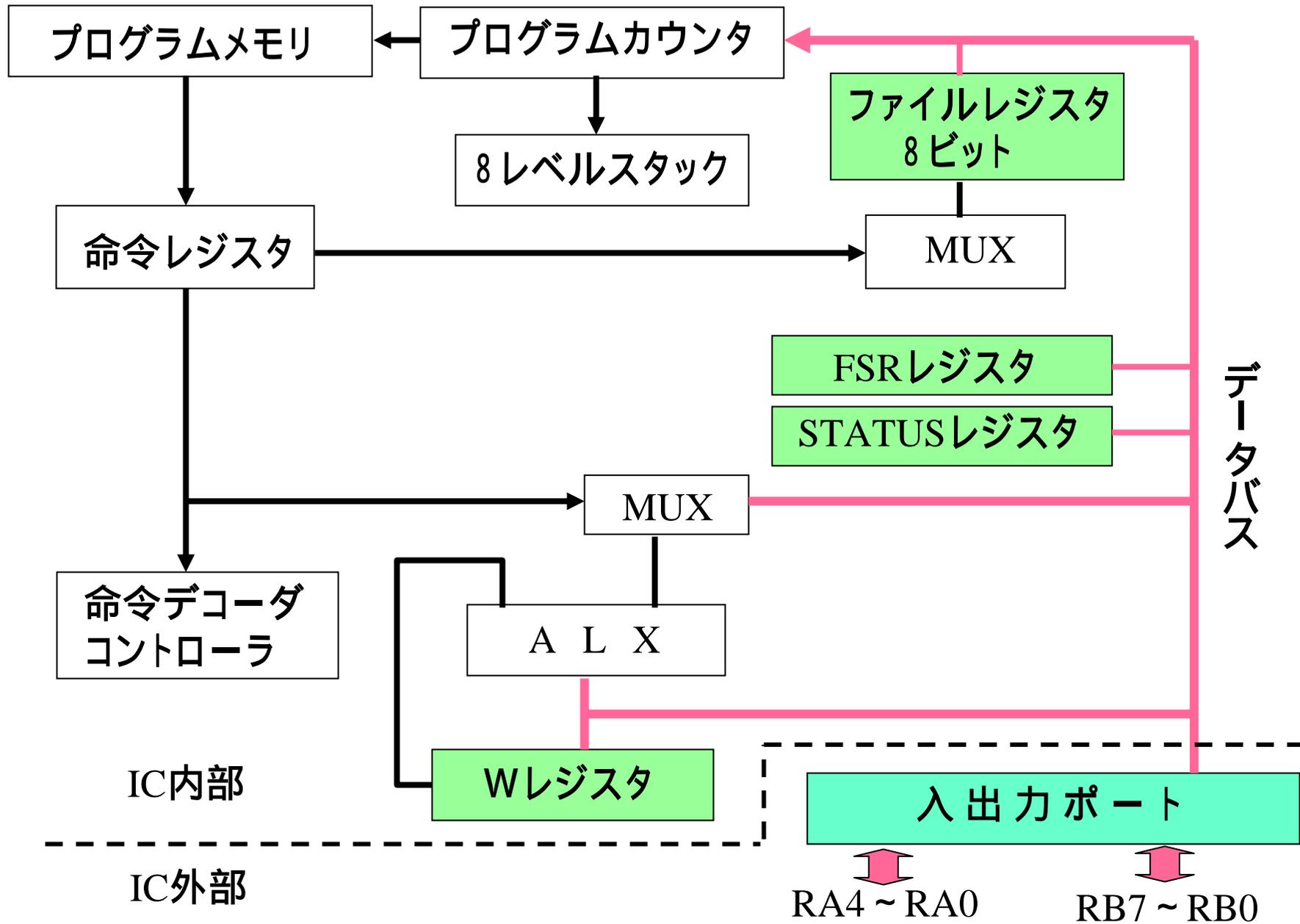
シリーズ	品名	プログラムメモリ [word]	データメモリ [byte]	Flash データメモリ [byte]	入出力 ピン数	A/D コンバータ	アナログ コンパレータ	キャプチャ /コンパレータ /PWM	シリアル ポート SPI/I2C USART	パラレル ポート	タイマ	動作可能 電源電圧 [V]	最大 動作 周波数 [MHz]	命令数	パッケージ ピン数
ベースライン	12C508A	512	25		5						1+WDT	2.5~5.5	4	33	8PDIP
	12C509A	1k	41												
	16C54B	512	25		12					1+WDT	3.0~5.5	20	35		18PDIP
	16C56A	1k													
	16C57C	2k	72		20					3+WDT	2.5~6.0	20			28SDIP
	16C58B		73			12									
ミドルレンジ	12C671	1k	128		5		4					1+WDT	2.5~5.5	4	35
	12C672	2k													
	16C620A	512	80		13	2				1+WDT	3.0~6.0	20	35	18PDIP	
	16C621A	1k													
	16C622A	2k		128											
	16C62A	2k	128		22			1	SPI	3+WDT	2.5~6.0	20	35	28SDIP	
	16C63A	4k	192				2	SPI,SSART							
	16C64A	2k	128		33			1	SPI	3+WDT	2.5~6.0	20	35	40PDIP	
	16C710	512	36		13	4								1+WDT	2.5~6.0
	16C711	1k	68												
	16C715	2k	128		22	5			1	3+WDT	2.5~6.0	20	35	28SDIP	
	16C72														
	16C73B	4k	192		33	8			2	SPI,USART	3+WDT	2.5~6.0	20	35	40PDIP
	16C74B							1							
	16F83A	512	36	64	13	8	(10bit)				1+WDT	2.5~5.5	20	58	18PDIP
	16F84A	1k	68												
16F874	4k	192	128	33	8	(10bit)	2	SPI,USART	1	3+WDT	2.5~5.5	20	58		40PDIP
16F877	8k	368	256												
ハイエンド	17C42A	2k	232		33					4+WDT	2.5~6.0	33	58		40PDIP
	17C43	4k	454					2	USART						
	17C44	8k								4+WDT	2.5~6.0	33	58	40PDIP	
	17C756	16k	902		50	12		4	USARTTx2						3.0~5.5

(注) SPI(Serial Peripheral Interface) I2C(Inter-Integrated Circuit)
 USART(Universal Synchronous Asynchronous Receiver Transmitter) WDT(Watch Dog Timer)

PICの用途

民生機器	OA機器	産業機器
TV、ビデオ ステレオ CDプレーヤ リモコン ビデオゲーム 電子レンジ 洗濯機、乾燥機 掃除機	マウス トラックボール キーボード コピー制御 ディスクドライブ バーコードリーダ FAX	モーター制御 コンプレッサ制御 ロボット制御 カードリーダ 汎用メータ

16F84の構成



メモリマップ

アドレス			アドレス
00h			80h
01h	TMR0	OPTION	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h			87h
08h	EEDATA	EECON1	88h
09h	EEADR	EECON2	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch			8Ch
	ファイル	バンク0に	
	レジスタ	マップされ	
4Fh		ている	CFh
	バンク0	バンク1	

特殊レジスタ一覧

バンク0

アドレス	名称	ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0
00h	INDF	FSRの内容のアドレスのデータメモリ							
01h	TMRO	8ビットリアルタイム・クロック/カウンタ							
02h	PCL	プログラムカウンタ(PC)の下位8ビット							
03h	STATUS	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C
04h	FSR	間接データメモリアドレスポインタ							
05h	PORTA	-	-	-	RA4/TOCKI	RA3	RA2	RA1	RA0
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT
07h		使用しない							
08h	EEDATA	EEDATAEEPROMデータレジスタ							
09h	EEADR	EEADREEEPROMアドレスレジスタ							
0Ah	PCLACH	-	-	-	プログラムカウンタ(PC)の上位5ビットへの書き込みバッファ				
0Bh	INTCON	GLE	EEIE	TOTE	INTE	EBIE	TOIF	INTF	RBIF

バンク1

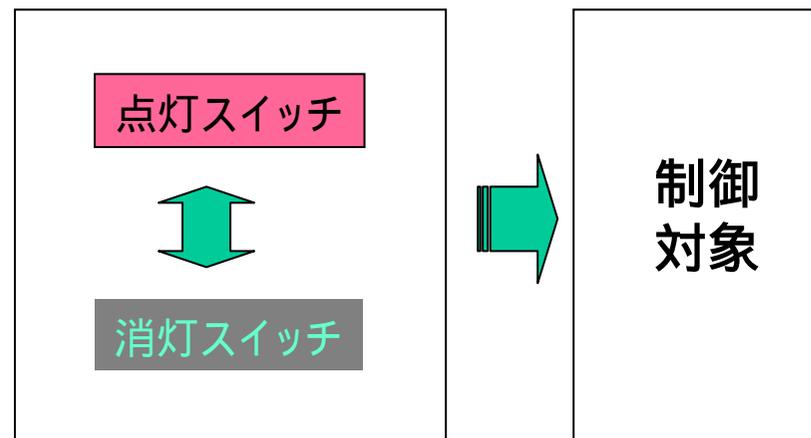
アドレス	名称	ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0
80h	INDF	FSRの内容のアドレスのデータメモリ							
81h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
82h	PCL	プログラムカウンタ(PC)の下位8ビット							
83h	STATUS	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C
84h	FSR	間接データメモリアドレスポインタ							
85h	TRISA	-	-	-	PORTAデータ入出力設定レジスタ				
86h	TRISB	PORTBデータ入出力設定レジスタ							
87h		使用しない							
88h	EECON1	-	-	-	EEIF	WRERR	WREN	WR	RD
89h	EECON2								
8Ah	PCLACH	-	-	-	PCの上位5ビットへの書き込みバッファ				
8Bh	INTCON	GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF

バンク0とバンク1の両方にあるレジスタは同一で、どちらからもアクセスできる

例題構成

例題

- 1) スイッチA 入れたら点灯
- 2) スイッチB 入れたら消灯
- 1) 2) の繰り返しを作成



使用ポート

未使用ポート

入力1 : RA2

入力2 : RA3

出力 : RB1

RB7

RA0

RA1

RA4

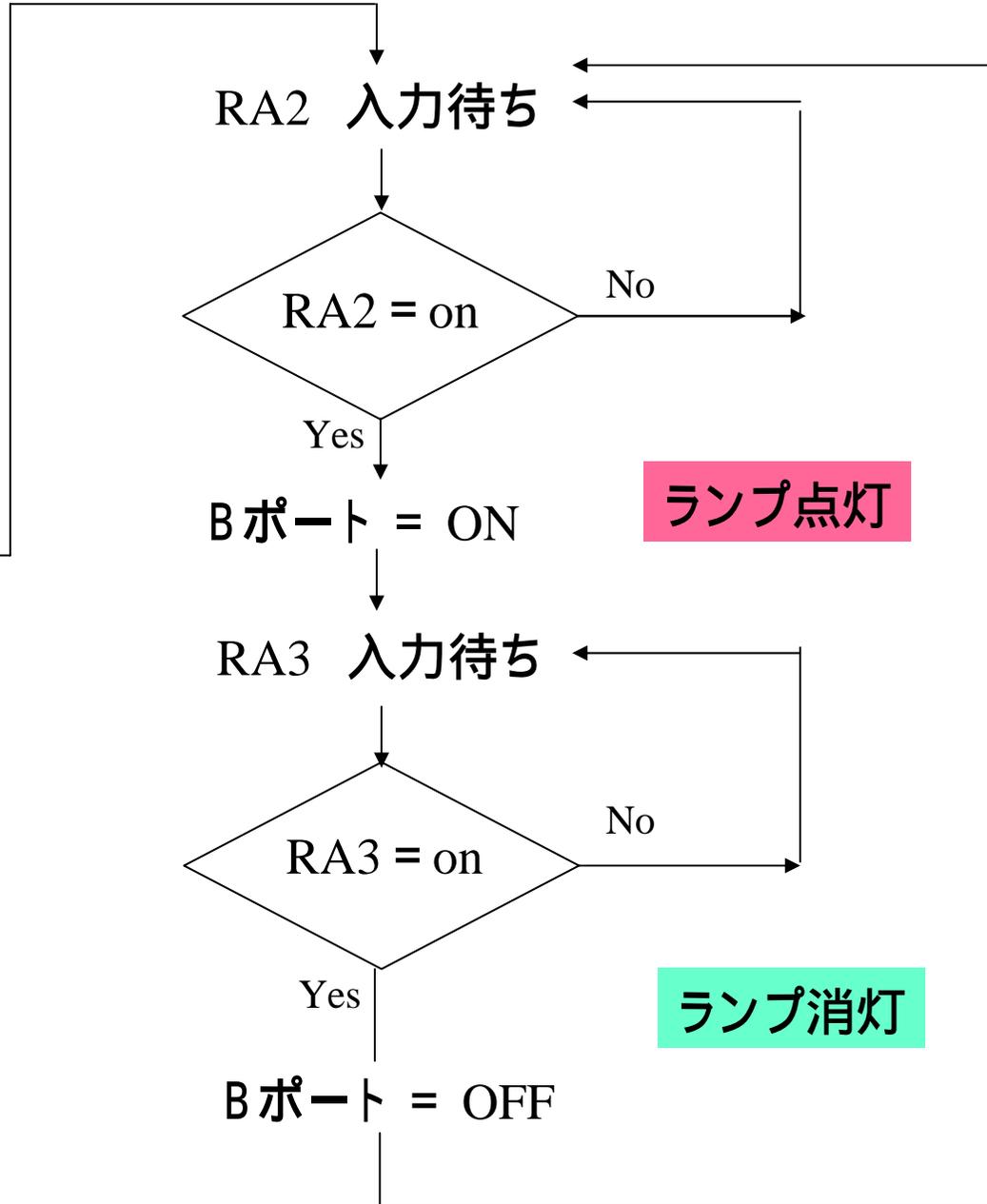
RB0

RA2	1	18	RA1
RA3	2	17	RA0
RA4/T0CKI	3	16	OSC1/CLKIN
MCLR	4	15	OSC2/CLKOUT
Vss	5	14	Vdd
RB0/INT	6	13	RB7
RB1	7	12	RB6
RB2	8	11	RB5
RB3	9	10	RB4

PIC16F84

フローチャート

スタート
↓
初期設定
↓



アセンブラの書き方1/2



MAIN

```
[] BSF STATUS, RP0
>[] CLRF TRISB ;Bポートクリア
>[] MOVLW 0CH ;Aポートの設定データ準備
>[] MOVWF TRISA ;Aポート設定データのセット
>[] BCF STATUS, RP0 ;バンク0に切り替え
>[] CLRF PORTA ;Aポートクリア
>[] CLRF PORTB ;Bポートクリア
```

アセンブラの書き方2/2

ラベル	ニーモニック	オペランド	コメント
-----	--------	-------	------

ラベル : プログラムメモリに対して、アドレスの代わりに用いる

記述ルール

行の先頭より記述

英文字・アンダーバーで始まる半角 3 2 文字以内

大文字・小文字は区別する

ニーモニック : 命令の内容

記述ルール : ラベルとの間を 1 文字以上の空白をあける

オペランド : 命令に対応する f : d : b : k 記述

コメント : コメントを記述

記述ルール : セミコロン“ ; ”以降がコメントとなる

疑似命令

アセンブラに対する制御命令の記載

- INCLUDE命令 : 指定したファイルのソースプログラムを読み込む
- _CONFIG命令 : 基本状態、プロテクト設定をする
- ORG命令 : 機械語を格納するプログラムメモリの先頭番地を指定する
- END命令 : ソースプログラムの終了を示す
- EQU命令 : 数値をラベルとして割り当てる

INCファイル例

保存されている場所

¥MPLAB IDE ¥MCHIP_TOOLS ¥PIC16F84A.INC

W EQU H'0000'

F EQU H'0001'

;----- Register Files-----

INDF EQU H'0000'

TMR0 EQU H'0001'

PCL EQU H'0002'

STATUS EQU H'0003'

FSR EQU H'0004'

PORTA EQU H'0005'

PORTB EQU H'0006'

EEDATA EQU H'0008'

;----- STATUS Bits -----

IRP EQU H'0007' NOT_TO EQU H'0004

RP1 EQU H'0006' NOT_PD EQU H'0003

RP0 EQU H'0005'

Z EQU H'0002' DC EQU H'0001'

C EQU H'0000'

特殊機能

特殊メモリ

IDワード

コンフィグレーション

IDワード

プログラム実行中は読み書きできない、ライターでのみ可能

メモリ領域 2000Hから2003H(4ワード)

プログラム製造年月日

バージョン番号

プログラム管理用番号

メモリチェックサム

コンフィギュレーション・ワード

(PIC16F84)

13

0



アドレス 2007H

CPn	プログラム領域のプロテクト
DP	EPROMのデータメモリのプロテクト
BODEN	ブラウンアウトリセット機能の設定
PWRTE	電源オンリセット機能の設定
MCLRE	MCLRピンの機能の選択
WDTE	ウォッチドッグタイマの設定
FOSC1	発振素子回路の指定
FOSC0	発振素子回路の指定

MPLABの記載例

`__CONFIG _HS_OSC & _WDT_OFF & _PWRTE_ON & _CP_OFF`

<code>_HS_OSC</code>	HSモード
<code>_WDT_OFF</code>	ウォッチドッグタイマーをOFF設定
<code>_PWRTE_ON</code>	電源ONリセット機能をON設定
<code>_CP_OFF</code>	プログラム領域のプロテクトOFF

発振モード	LP	200kHz以下	低電力 / 水晶発振
	XT	4MHz以下	水晶 / セラミック
	HS	4M ~ 20MHz	高周波水晶 / セラミック
	RC	RC発振	1MHz以下

LIST

LIST P=PIC16F84A

MPLABのバージョンにより記述要

例題1/

```
-----  
; ON・OFFスイッチ入力及び出力の練習プログラム  
-----  
INCLUDE      P16F84A.INC  
list P=PIC16F84A  
__CONFIG _HS_OSC & _WDT_OFF & _PWRTE_ON & _CP_OFF  
  
GPR_1 EQU      0Ch      ;定義文  
GPR_2 EQU      0Dh  
GPR_3 EQU      0Eh  
ORG 0          ;リセットベクタ(0番地)を指定  
GOTO          MAIN      ;MAINへ行く
```

例題 2 /

```
MAIN
  BSF      STATUS,RP0    ;バンク 1 に切り替え
  CLRF     TRISB         ;Bポートクリア>Bポート全部を出力設定
  MOVLW   0Ch           ;Aポートの設定データ準備
  MOVWF   TRISA         ;RA2とRA3を入力ポートに設定
  BCF     STATUS,RP0    ;バンク 0 に切り替え
  CLRF     PORTA        ;Aポートクリア
  CLRF     PORTB        ;Bポートクリア
PBS1
  ;RA2 入力待ち (点灯)
  BTFSC   PORTA,2      ;RA2が0なら次の命令スキップ
  GOTO    PBS1
  MOVLW   0FEh         ;Bポートデータ準備(全LED点灯)
  MOVWF   PORTB        ;Bポートデータセット
PBS2
  ;RA3 入力待ち (消灯)
  BTFSC   PORTA,3      ;RA3が0なら次の命令スキップ
  GOTO    PBS2
  MOVLW   0h           ;Bポートデータ準備(全LED消灯)
  MOVWF   PORTB        ;Bポートデータセット
  GOTO    PBS1         ;戻る
END
```


12行目 MAIN

13行目 BSF STATUS,RP0

ビット処理命令 BSF f,d

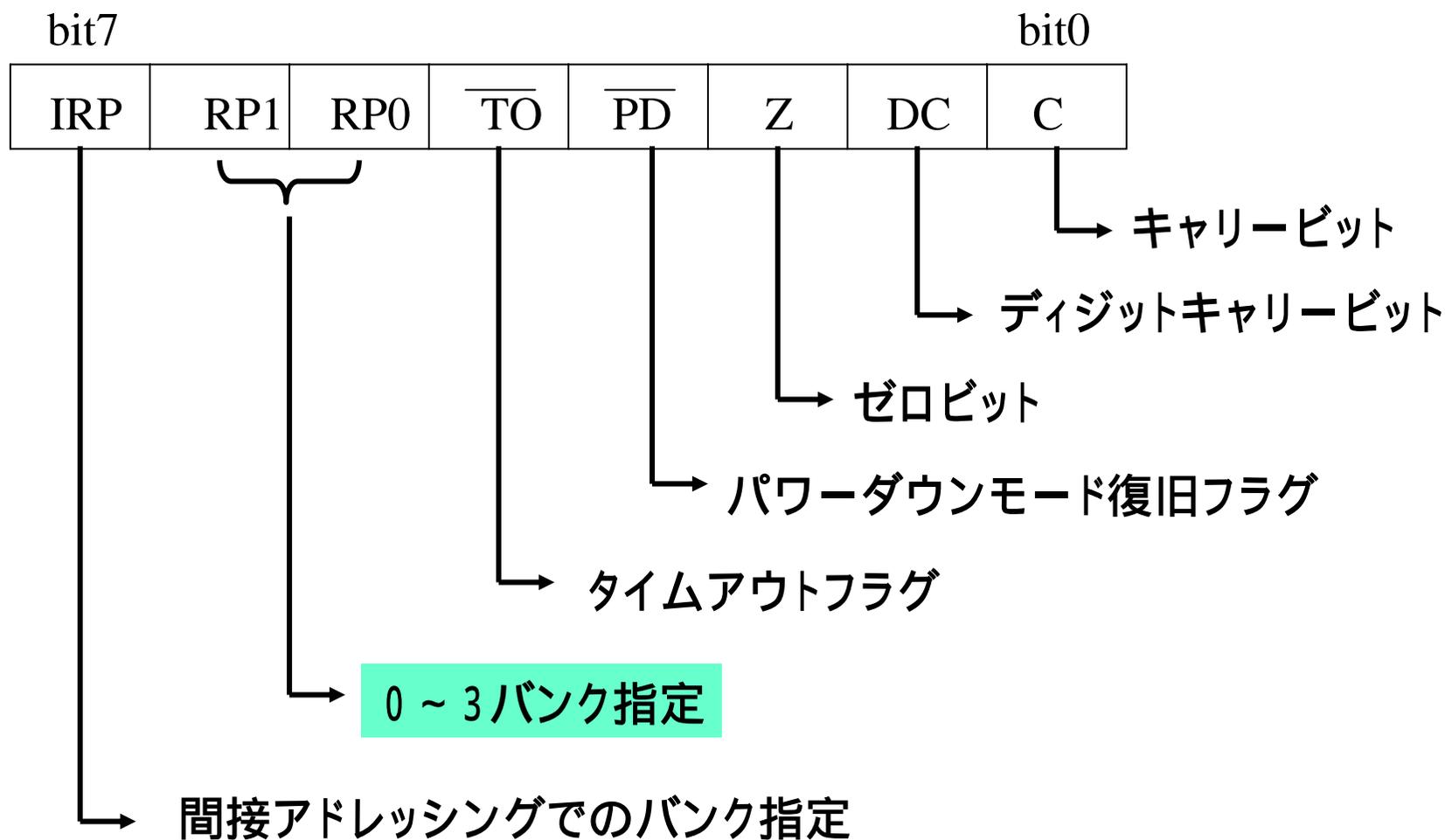
→ f : STATUS
d : RP0

03h レジスタマップ参照
5ビット目を1hにする

← バンク1に変更される

バンク0	アドレス	名称	ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0
	00h	INDF	FSRの内容のアドレスのデータメモリ							
	01h	TMRO	8ビットリアルタイム・クロック / カウンタ							
	02h	PCL	プログラムカウンタ(PC)の下位8ビット							
	03h	STATUS	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C
	04h	FSR	間接データメモリアドレスポインタ							
	05h	PORTA	-	-	-	RA4/TOCKI	RA3	RA2	RA1	RA0
	06h	PORTB	RB7	5B6	RB5	RB4	RB3	RB2	RB1	RB0/INT
	07h		使用しない							
	08h	EEDATA	EEDATAEEPROMデータレジスタ							
	09h	EEADR	EEADREEPROMアドレスレジスタ							
	0Ah	PCLACH	-	-	-	プログラムカウンタ(PC)の上位5ビットへの書き込みバッファ				
	0Bh	INTCON	GLE	EEIE	TOTE	INTE	EBIE	TOIF	INTF	RBIF

STATUSレジスタ



14行目

CLRF TRISB

バイト処理命令 CLRF f

→ f : TRISB

86h番地を0hにする

出力設定になる

バンク1	名称	ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0
80h	INDF	FSRの内容のアドレスのデータメモリ							
81h	OPTION_REG	RBPV	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
82h	PCL	プログラムカウンタ(PC)の下位8ビット							
83h	STATUS	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C
84h	FSR	間接データメモリアドレスポインタ							
85h	TRISA	-	-	-	PORTAデータ入出力設定レジスタ				
86h	TRISB	PORTBデータ入出力設定レジスタ							
87h		使用しない							
88h	EECON1	-	-	-	EEIF	WRERR	WREN	WR	RD
89h	EECON2								
8Ah	PCLACH	-	-	-	PCの上位5ビットへの書き込みバッファ				
8Bh	INTCON	GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF

15行目 MOVLW 0Ch

リテラル処理命令 MOVLW k

→ k: 0Ch

0Ch を Wに入れる

16行目 MOVWF TRISA

バイト処理命令 MOVWF f

→ f: TRISA

85番地に0Chが入る

(Ch=1100)

バンク1	アドレス	名称	ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0
	80h	INDF	FSRの内容のアドレスのデータメモリ							
	81h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
	82h	PCL	プログラムカウンタ(PC)の下位8ビット							
	83h	STATUS	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C
	84h	FSR	間接データメモリアドレスポインタ							
	85h	TRISA	-	-	-	PORTAデータ入出力設定レジスタ				
	86h	TRISB	PORTBデータ入出力設定レジスタ							
	87h		使用しない							
	88h	EECON1	-	-	-	EEIF	WRERR	WREN	WR	RD
	89h	EECON2								
	8Ah	PCLACH	-	-	-	PCの上位5ビットへの書き込みバッファ				
	8Bh	INTCON	GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF

17行目 BCF STATUS, RPO

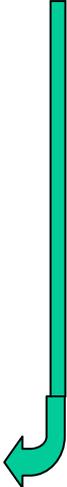
ビット処理命令 BCF f, b

f: STATUS
d: RPO

83h 5ビット目を0

バンク0に変更される

バンク1	名称	ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0
80h	INDF	FSRの内容のアドレスのデータメモリ							
81h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
82h	PCL	プログラムカウンタ(PC)の下位8ビット							
83h	STATUS	IRP	RP1	RP0	T0	PD	Z	DC	C
84h	FSR	間接データメモリアドレスポインタ							
85h	TRISA	-	-	-	PORTAデータ入出力設定レジスタ				
86h	TRISB	PORTBデータ入出力設定レジスタ							
87h		使用しない							
88h	EECON1	-	-	-	EEIF	WRERR	WREN	WR	RD
89h	EECON2								
8Ah	PCLACH	-	-	-	PCの上位5ビットへの書き込みバッファ				
8Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF



18行目 CLRF PORTA

バイト処理命令 CLRF f



f: TRISA

85番地に0hが入る

バンク0									
アドレス	名称	ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0
00h	INDF	FSRの内容のアドレスのデータメモリ							
01h	TMRO	8ビットリアルタイム・クロック / カウンタ							
02h	PCL	プログラムカウンタ(PC)の下位8ビット							
03h	STATUS	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C
04h	FSR	間接データメモリアドレスポインタ							
05h	PORTA	-	-	-	RA4/TOCKI	RA3	RA2	RA1	RA0
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT
07h		使用しない							
08h	EEDATA	EEDATAEEPROMデータレジスタ							
09h	EEADR	EEADREEPROMアドレスレジスタ							
0Ah	PCLACH	-	-	-	プログラムカウンタ(PC)の上位5ビットへの書き込みバッファ				
0Bh	INTCON	GLE	EEIE	TOTE	INTE	EBIE	TOIF	INTF	RBIF



19行目 CLRF PORTB

バイト処理命令 CLRF f



f : TRISB

86番地に0hが入る

バンク0									
アドレス	名称	ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0
00h	INDF	FSRの内容のアドレスのデータメモリ							
01h	TMRO	8ビットリアルタイム・クロック / カウンタ							
02h	PCL	プログラムカウンタ(PC)の下位8ビット							
03h	STATUS	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C
04h	FSR	間接データメモリアドレスポインタ							
05h	PORTA	-	-	-	RA4/TOCKI	RA3	RA2	RA1	RA0
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT
07h		使用しない							
08h	EEDATA	EEDATAEEPROMデータレジスタ							
09h	EEADR	EEADREEPROMアドレスレジスタ							
0Ah	PCLACH	-	-	-	プログラムカウンタ(PC)の上位5ビットへの書き込みバッファ				
0Bh	INTCON	GLE	EEIE	TOTE	INTE	EBIE	TOIF	INTF	RBIF



20行目 P B S 1

ラベル

21行目 BTFSC PORTA , 2

ビット処理命令 BTFSC f, b



f : PORTA
B : 2

0 : スキップ
1 : 次の行

バンク0									
アドレス	名称	ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0
00h	INDF	FSRの内容のアドレスのデータメモリ							
01h	TMRO	8ビットリアルタイム・クロック / カウンタ							
02h	PCL	プログラムカウンタ(PC)の下位8ビット							
03h	STATUS	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C
04h	FSR	間接データメモリアドレスポインタ							
05h	PORTA	-	-	-	RA4/TOCKI	RA3	RA2	RA1	RA0
06h	PORTB	RB7	5B6	RB5	RB4	RB3	RB2	RB1	RB0/INT
07h		使用しない							
08h	EEDATA	EEDATAEEPROMデータレジスタ							
09h	EEADR	EEADREEPROMアドレスレジスタ							
0Ah	PCLACH	-	-	-	プログラムカウンタ(PC)の上位5ビットへの書き込みバッファ				
0Bh	INTCON	GLE	EEIE	TOTE	INTE	EBIE	TOIF	INTF	RBIF

22行目

```

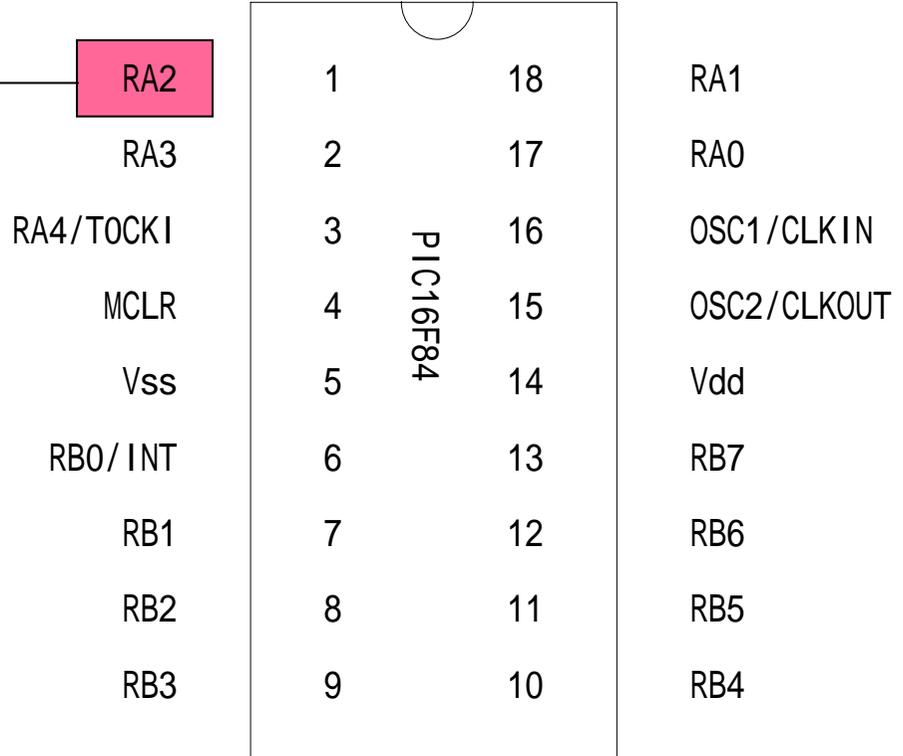
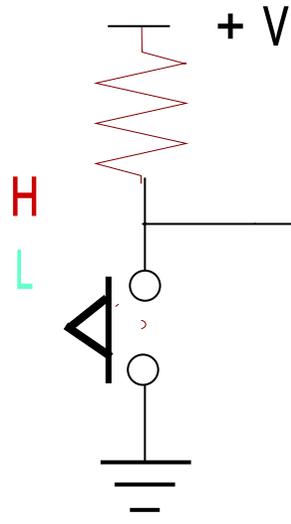
    → PBS1
    → BTFSC
    → GOTO
    → MOVLW
    → MOVWF
  
```

```

    PORTA,2
    PBS1
    0FEh
    PORTB
  
```

;RA2 入力待ち (点灯)
 ;RA2が0なら次の命令スキップ
 ;Bポートデータ準備(全LED点灯)
 ;Bポートデータセット

スイッチ OFF
 スイッチ ON



2 3行目 MOVLW 0FEh

リテラル処理命令 MOVLW k

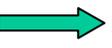


k: 0FEh

FEh を Wに入れる

2 4行目 MOVWF PORTB

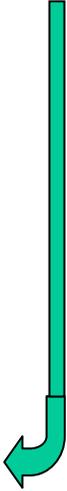
バイト処理命令 MOVWF f



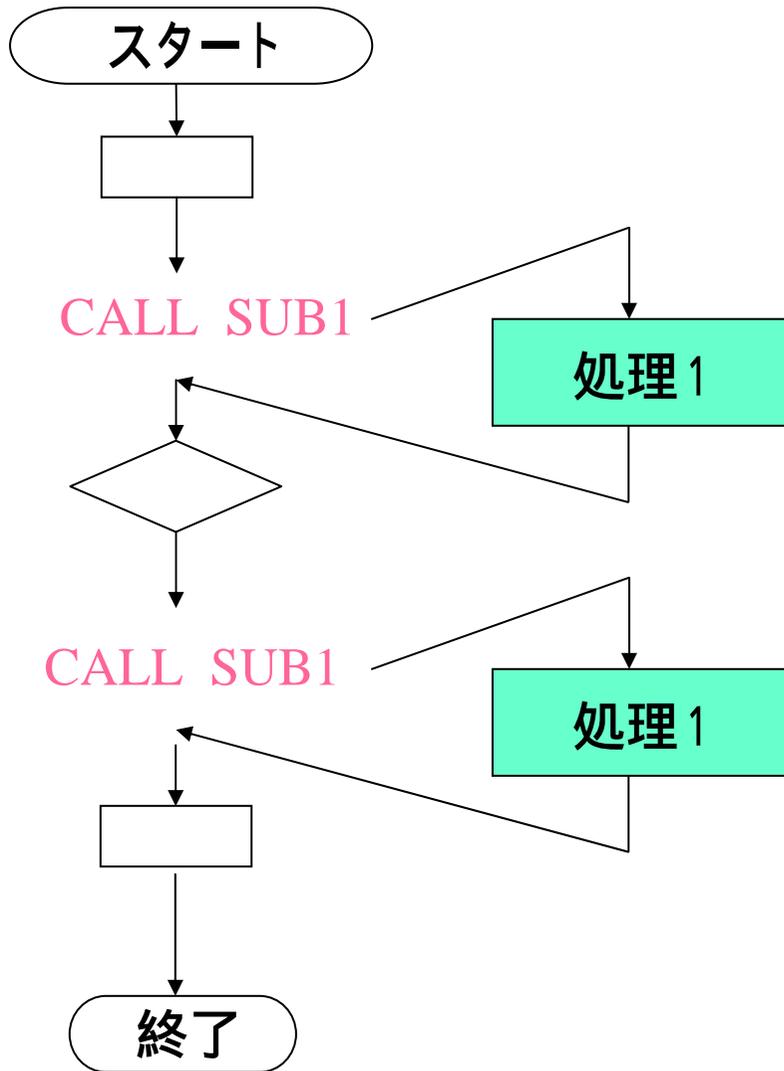
f: PORTB

06番地にFEhが入る

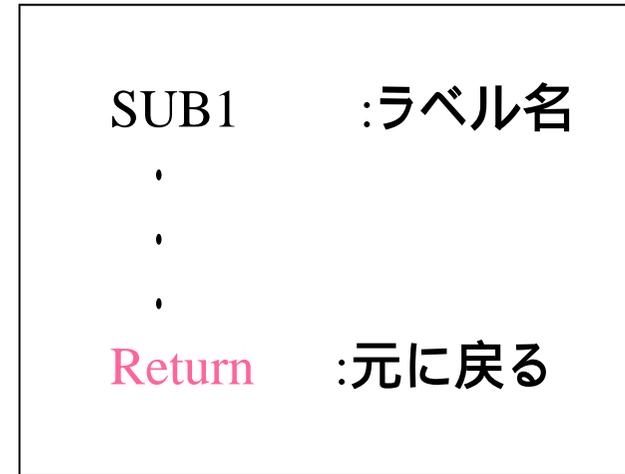
バンク0									
アドレス	名称	ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0
00h	INDF	FSRの内容のアドレスのデータメモリ							
01h	TMRO	8ビットリアルタイム・クロック/カウンタ							
02h	PCL	プログラムカウンタ(PC)の下位8ビット							
03h	STATUS	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C
04h	FSR	間接データメモリアドレスポインタ							
05h	PORTA	-	-	-	RA4/TOCKI	RA3	RA2	RA1	RA0
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT
07h		使用しない							
08h	EEDATA	EEDATAEEPROMデータレジスタ							
09h	EEADR	EEADREEPROMアドレスレジスタ							
0Ah	PCLACH	-	-	-	プログラムカウンタ(PC)の上位5ビットへの書き込みバッファ				
0Bh	INTCON	GLE	EEIE	TOTE	INTE	EBIE	TOIF	INTF	RBIF



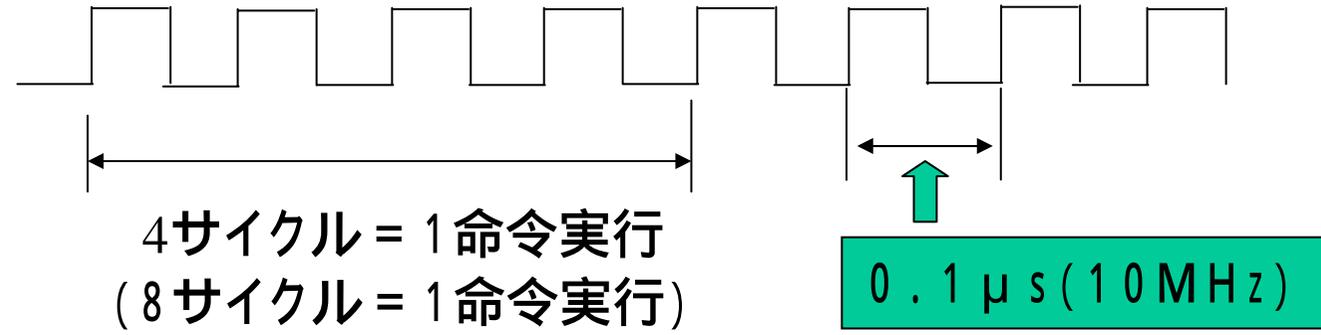
CALL文



処理1の書き方



タイマーの考え方の基本



TIMER

NOP	:0.4 μs	4サイクル必要命令
NOP	:0.4 μs + 0.4 μs = 0.8 μs	
NOP	:0.8 μs + 0.4 μs = 1.2 μs	
RETURN	:1.2 μs + 0.8 μs = 2 μs	8サイクル必要命令



必要に合わせて変更

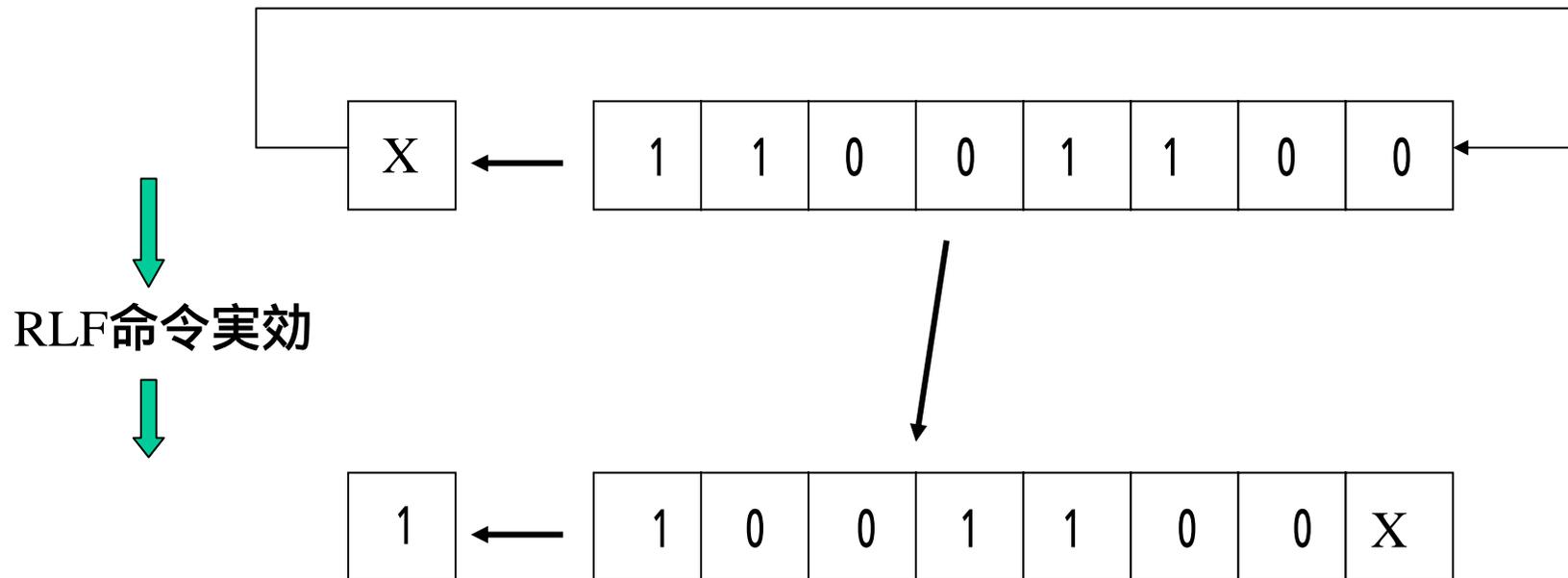
```

MOV LW    0F9h    :0.4秒作成時の数
MOV WF    GRP_1
TIML P1    NOP
DECFSZ    GRP_1  F  :減算を繰り返す
GOTO     TIML P1
RETURN
    
```

RLF命令

バイト処理命令 RLF f, d

Fレジスタの内容をキャリーフラグを含めて左に1ビット移動する



類似命令

RRF命令 : 移動方向が右に1ビット移動

OPTION_REG レジスタ

