

A

1. Un potentiel d'action est l'élément indivisible de transmission d'information dans le cerveau. Il correspond à un état binaire : soit un potentiel d'action est déclenché, soit il ne l'est pas. Il en est de même que le bit pour l'informatique, qui vaut soit 1 soit 0 et qui correspond donc à une information binaire.
2. Opcode : l'opération qui est effectuée ; opérande, sur quoi on fait l'opération. Par exemple : `movlw h'0A'`. Opcode : `movlw` ; opérande : `h'0A'`.
3. On passe par l'adresse quand on utilise le mode direct. En mode direct non étendu, le BSR (bank select register) permet de sélectionner la page mémoire dans laquelle on travaille.

B

- 1 le compteur programme,
- 2 le bus de données,
- 3 le bus d'instructions,
- 4 l'unité arithmétique et logique,
- 5 la mémoire programme,
- 6 les ports entrées-sorties.

C

(5 E)_h = 0101 1110 = a

(6 F)_h = 0110 1111 = b

a - b = a + Cplmt2(b) = ?

Pour faire le complément à deux on rajoute un 0 comme bit de signe à b.

On rajoute aussi un 0 à a pour que a et b aient le même nombre de bits.

La difficulté de l'exercice est de ne pas oublier de rajouter un bit de signe au préalable.

0 01011110 (attention au bit de signe)

+1 10010001 (attention au bit de signe)

=1 11101111

=(1EF)_h

Le résultat demandé est (1EF)_h en codage signé.

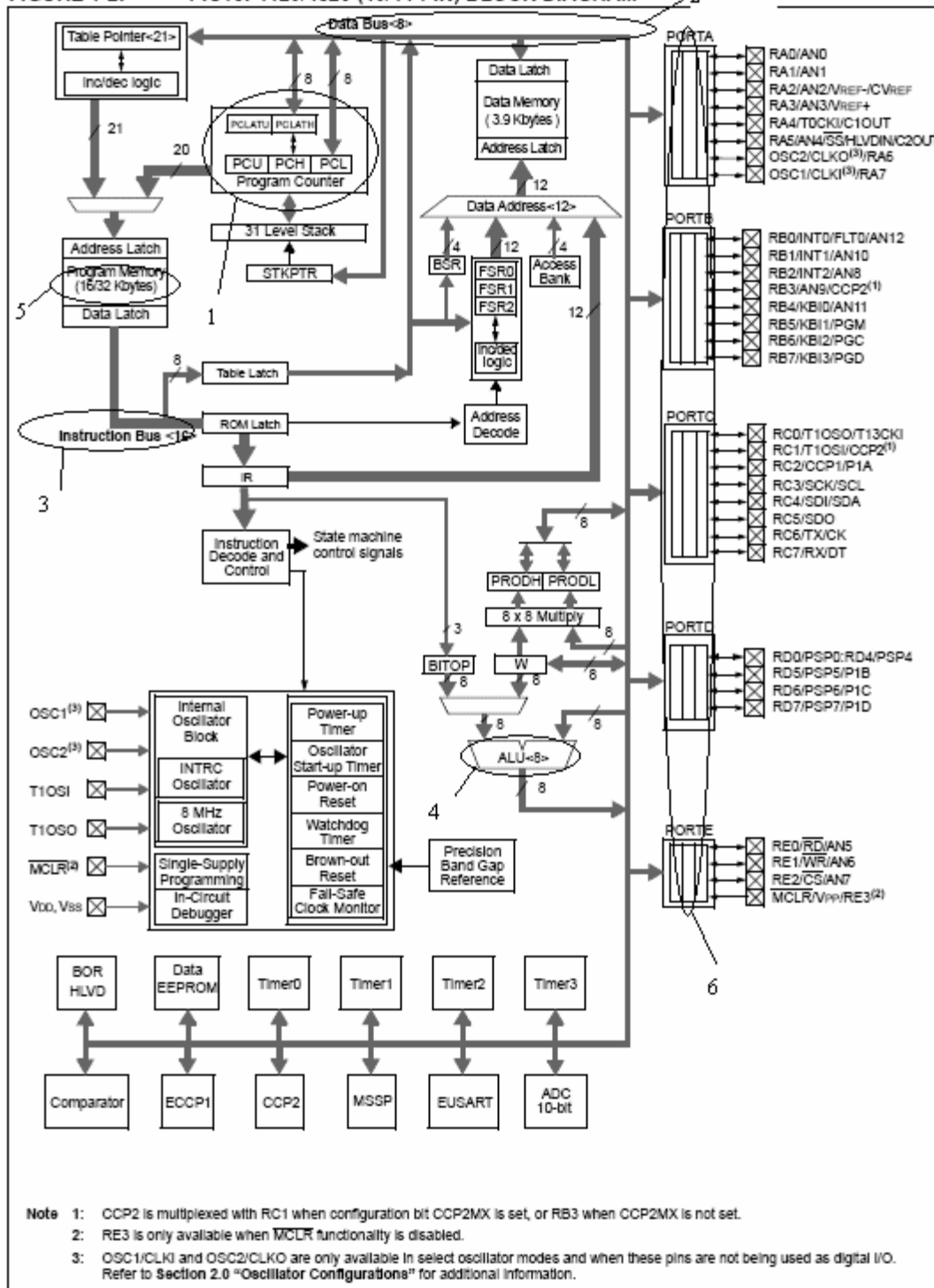
En prenant le complément à 2 du résultat binaire on obtient 0 00010001, donc le résultat de la soustraction est -(11)_h

On vérifie : $-1 \cdot 2^8 + 2^7 + 2^6 + 2^5 + 2^3 + 2^2 + 2 + 1 = -17$

Et (5 E)_h - (6 F)_h = (5*16 + 14) - (6*16 + 15) = 94 - 111 = -17

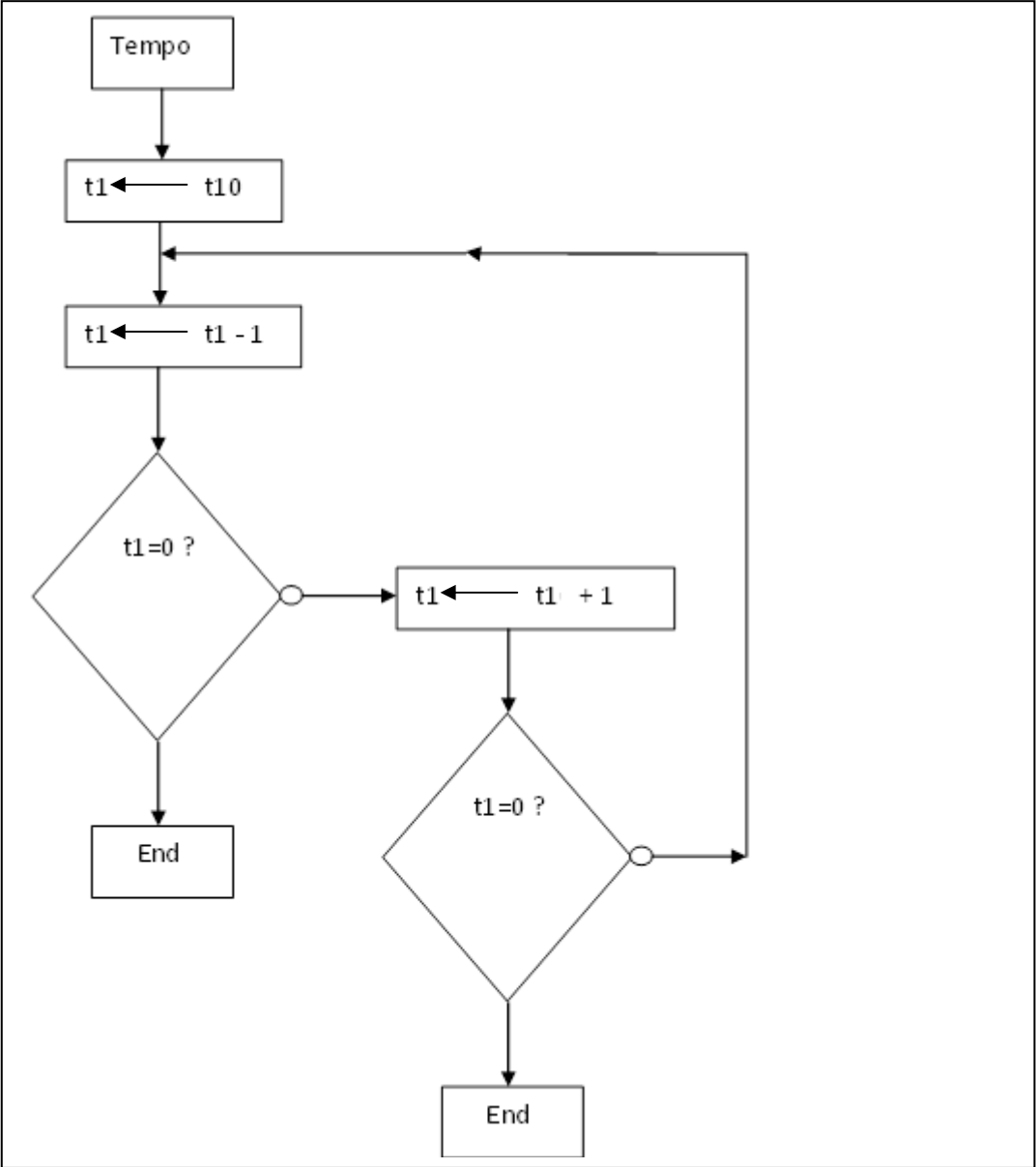
1010 xor 0001 = 1011

FIGURE 1-2: PIC18F4420/4520 (40/44-PIN) BLOCK DIAGRAM



Algorithme

I.



II Nombre de cycles :

| | | |
|--------------|-------|----------|
| movlw | t10 | 1 cycle |
| movwf | t1 | 1 cycle |
| comp1 dcfsnz | t1 | 2 cycles |
| return | | |
| comp2 infsnz | t1 | 2 cycles |
| goto | comp1 | |
| return | | 2 cycles |

$1 + 1 + 2 + 2 + 2 = 8$ cycles donc 8 microsecondes.

III Le rebouclage avant l'étape de décrémentation de t1 n'est jamais effectué ; la valeur de t10 n'a pas d'influence sur la durée du programme.

IV Une temporisation correcte est par exemple :

tempo

| | |
|--------------|-------|
| movlw | t10 |
| movwf | t1 |
| comp1 dcfsnz | t1 |
| return | |
| goto | comp1 |

Problème

- 1) Il y a 4 coups d'horloge par cycle donc la valeur de T_{cy} est $4 * 0.25 * 10^{-6} = 10^{-6}$.
- 2) (a) pour configurer le TIMER0 on utilise le registre T0CON.
(b) pour configurer les interruptions on utilise le registre INTCON.

;;;;;;;;;;;;; Programme principal

```
init    clrf    PORTB;Remise à zéro du port B
        movlw  h'00'
        movwf  TRISB      ;Le port B est défini en sortie

        movlw  h'83'
        movwf  TOCON      ;TIMER0 On, 16bits, Prescaler 16

        rcall  tmr0_init  ;Init TMR0 pour 1s pile
        movlw  h'A0'
        movwf  INTCON     ;Autorisation des interruptions
```

boucle

```
        nop
        goto   boucle     ;Boucle d'attente infinie
```

;;;;;;;;;;;;; Programme d'interruption

prog_int

```
        movwf  W_TEMP
        movff  STATUS, STATUS_TEMP
        movff  BSR, BSR_TEMP
```

```
        btfsc  INTCON,2
```

```
        rcall  tmr0_overflow
```

```
        movff  BSR_TEMP, BSR
        movff  W_TEMP, WREG
        movff  STATUS_TEMP, STATUS
        retfie
```

tmr0_overflow

```
        bcf    INTCON,2
```

```
        movlw  h'0B'
```

```
        movwf  TMR0H
```

```
        movlw  h'DB'
```

```
        movwf  TMR0L
```

```
        movlw  h'01'
```

```
        xorwf  PORTB
```

```
        return
```