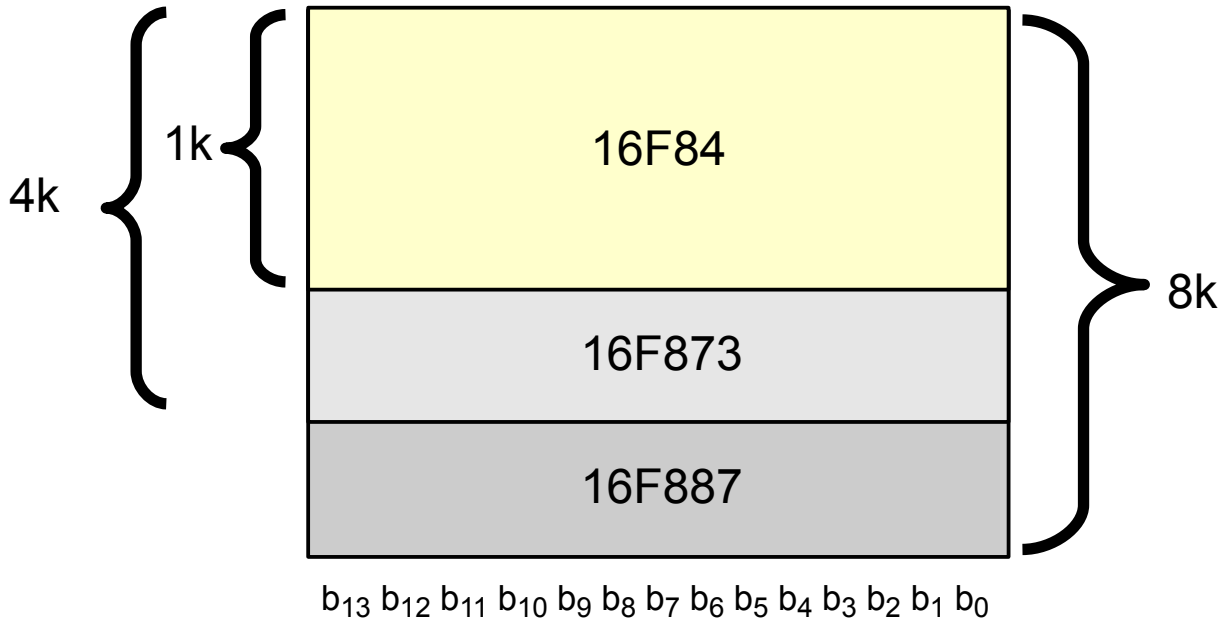


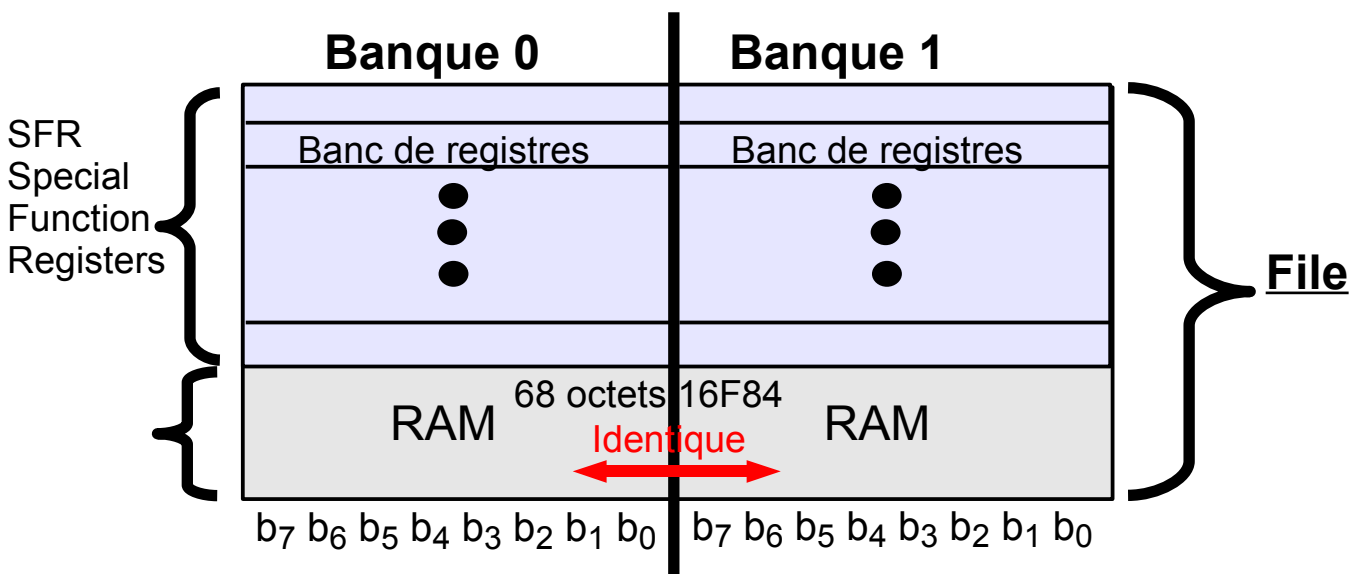
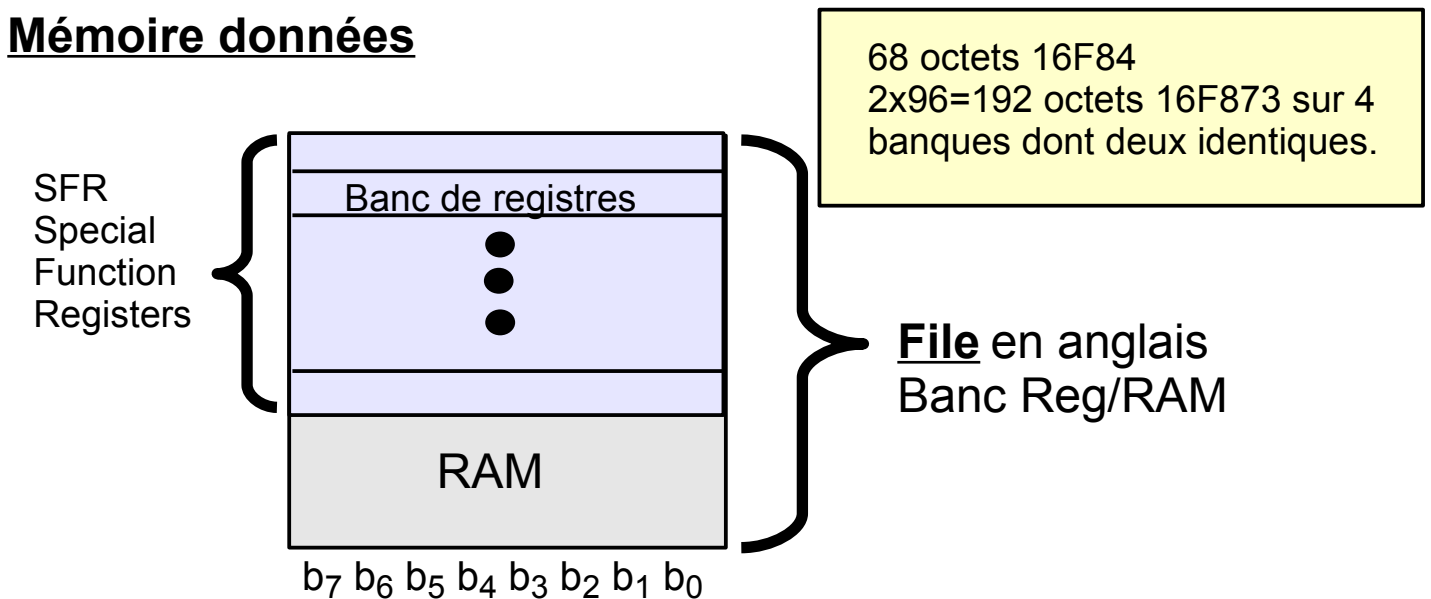
Cours 1- 2 : Architecture interne du PIC 16FXXX

Mémoire programme

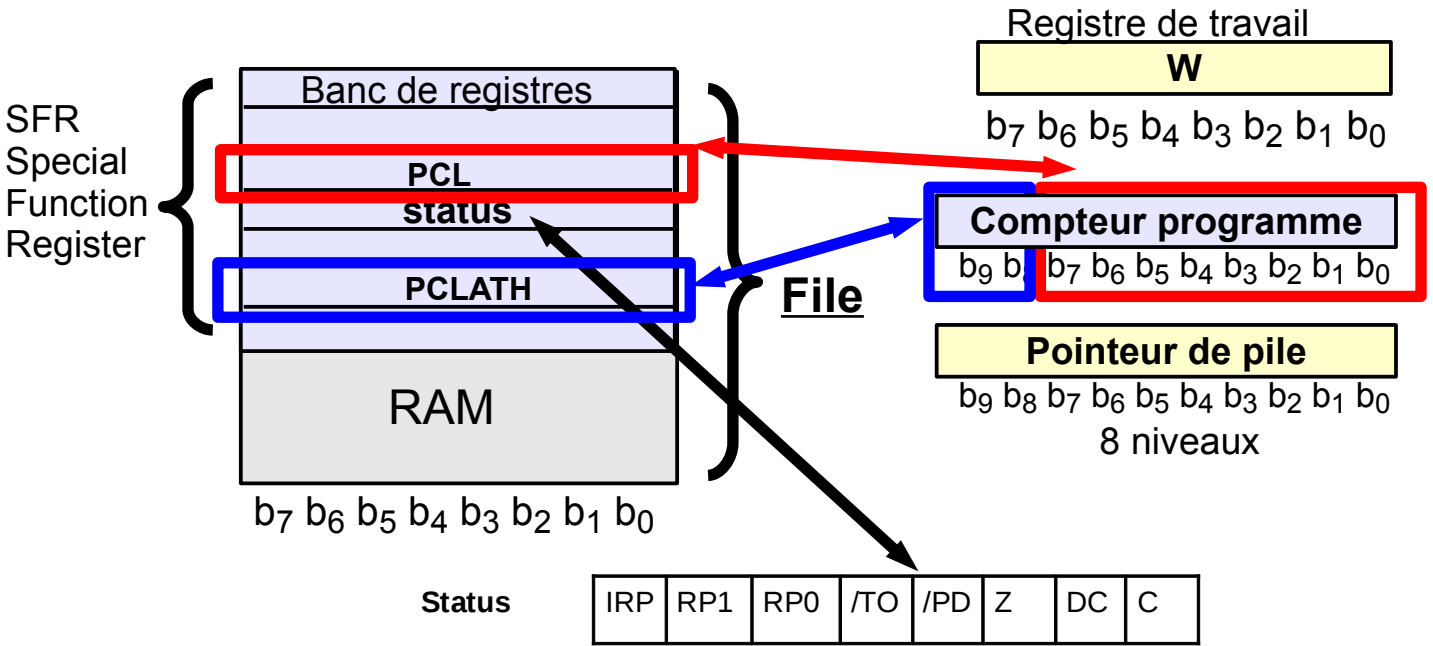


Que valent m et n ?

Mémoire données

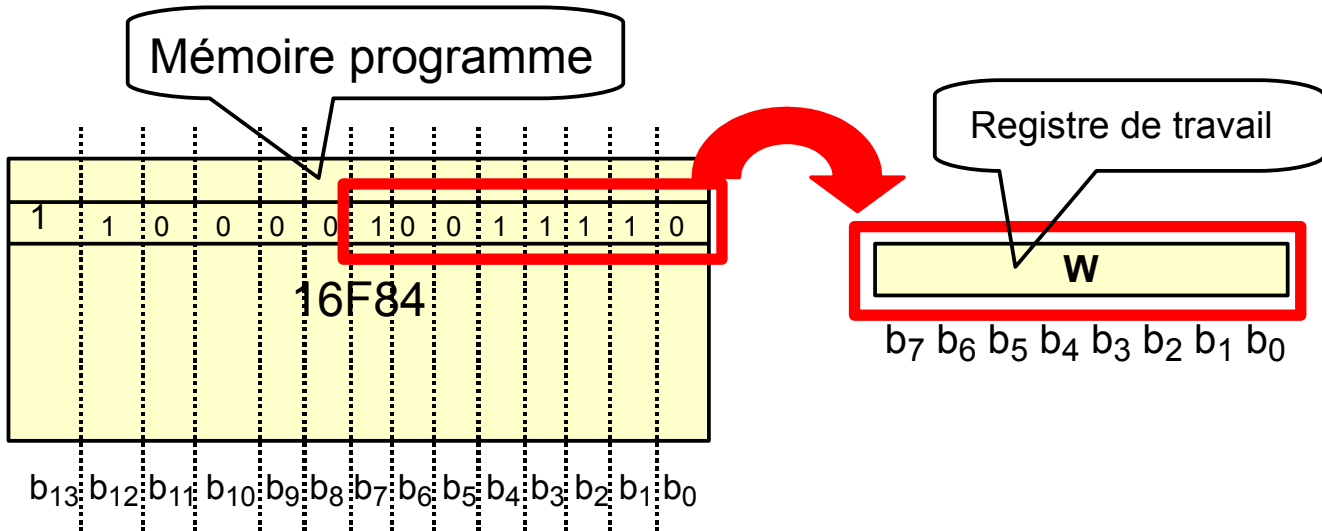
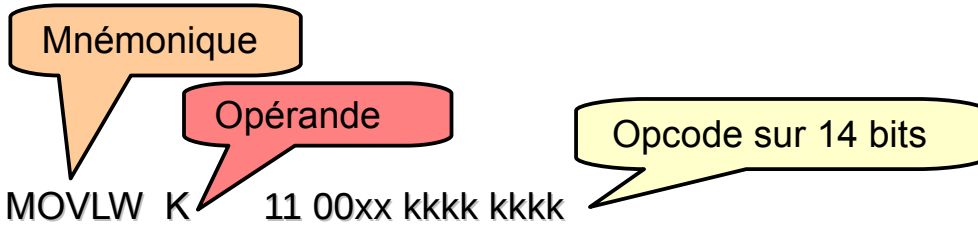


Modèle de programmation du PIC 16F84



Les registres jaunes ne font pas partie du banc de registres

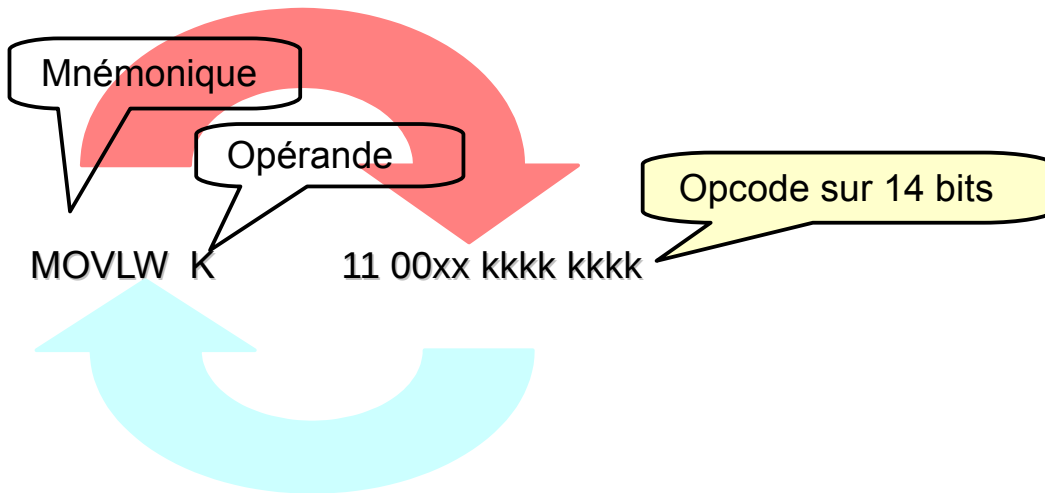
L'instruction MOVLW



La valeur est transférée à partir de la mémoire programme vers le registre de travail.

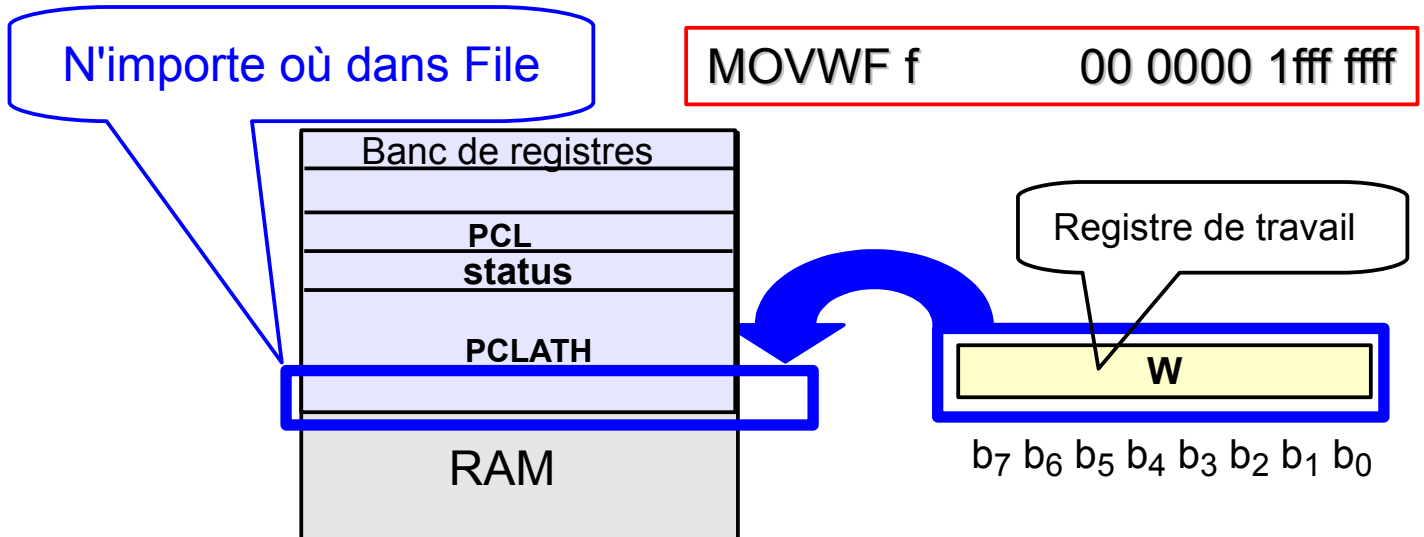
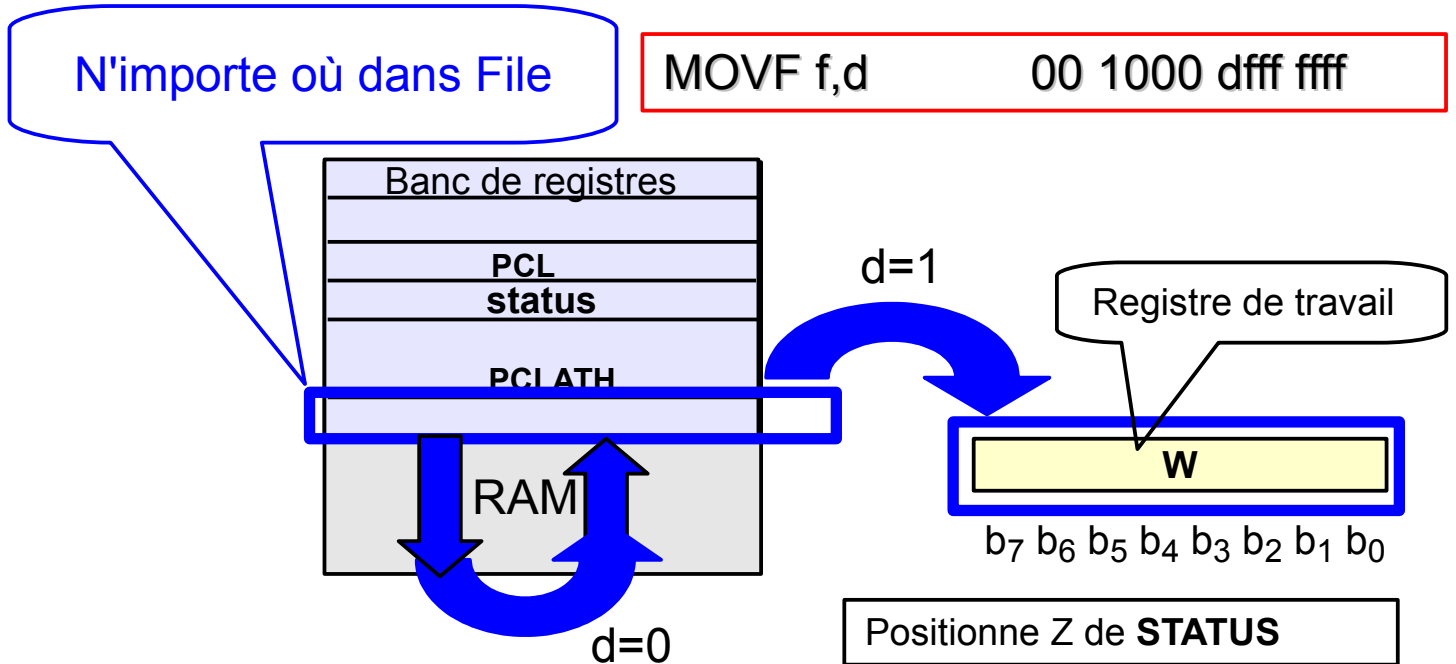
Assembler et désassembler

Assembler

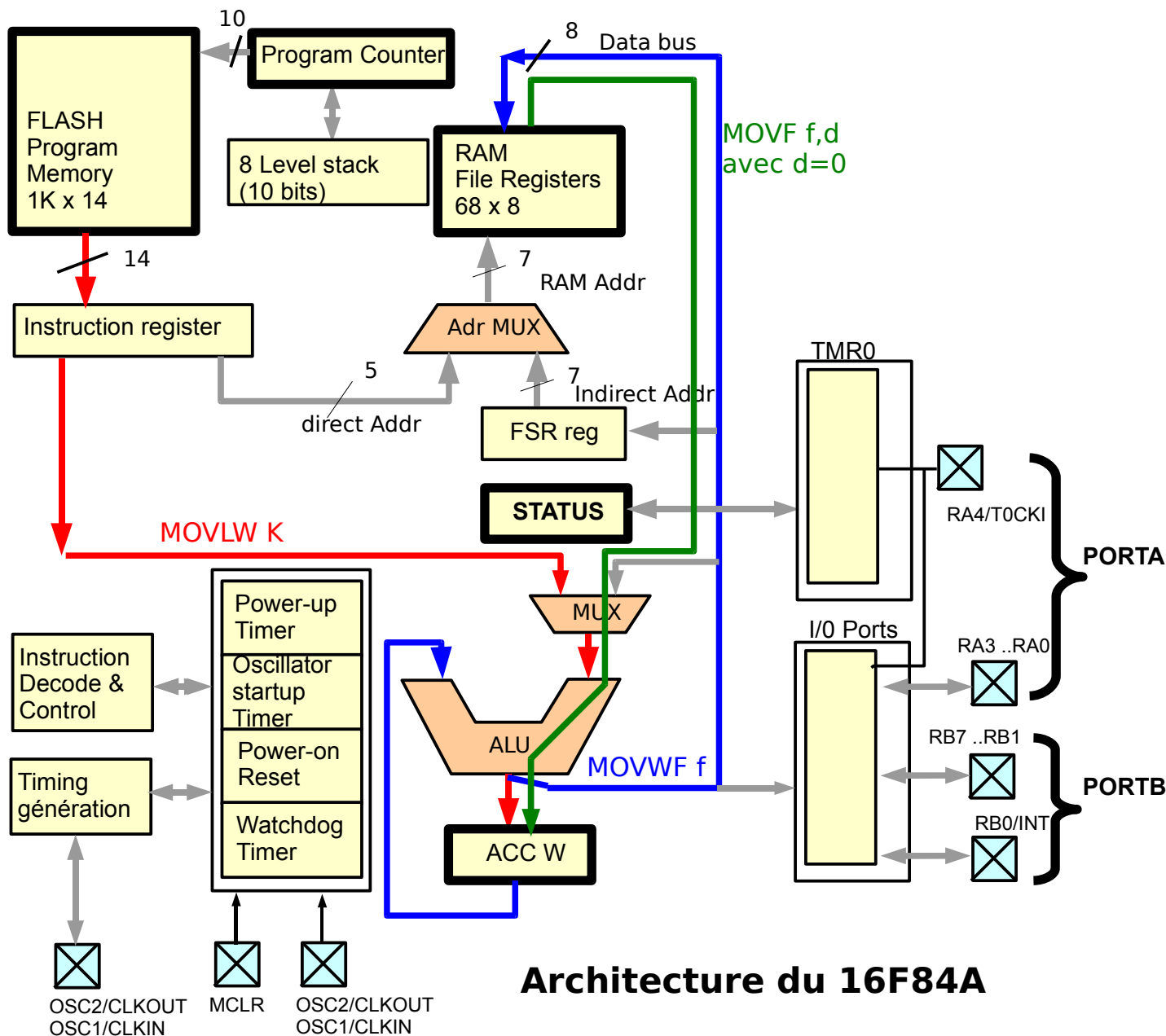


Désassembler

Les deux autres instructions MOV

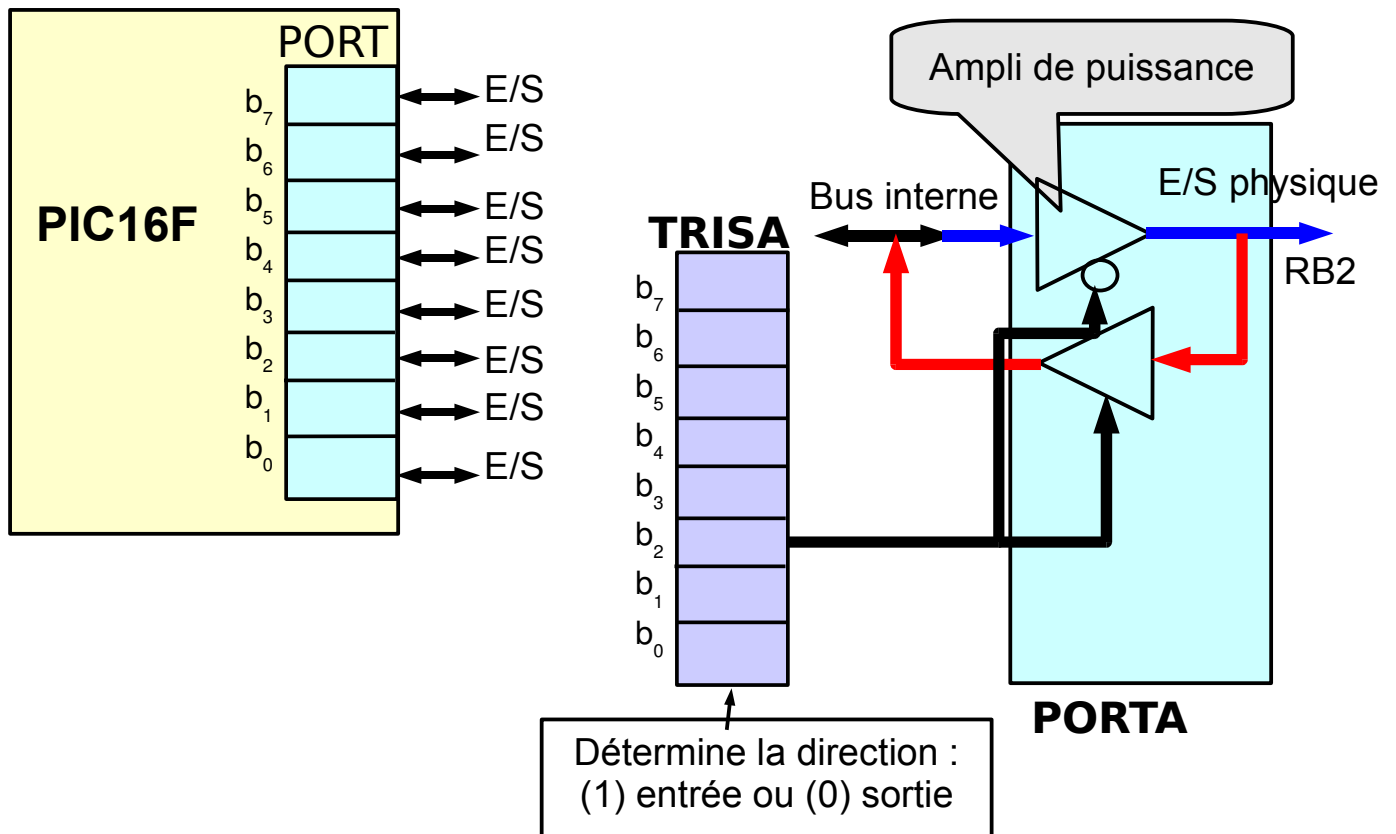


Les instructions MOV sur architecture complète

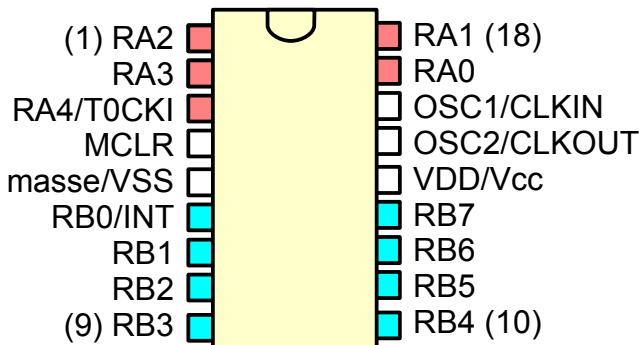


Exécuter un programme avec un compteur programme

Des PORTs pour les entrées et sorties (notées E/S)



Les PORTs du PIC 16F84



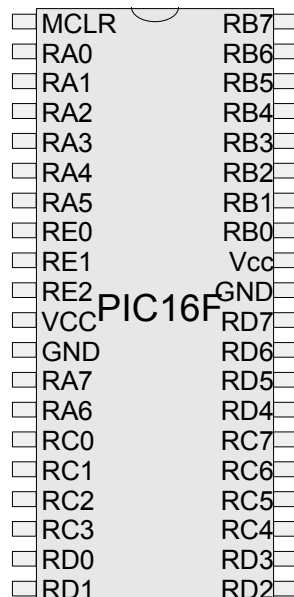
Deux ports :

- PORTA 5 bits $RA_0 \rightarrow RA_4$
- PORTB 8 bits $RB_0 \rightarrow RB_7$

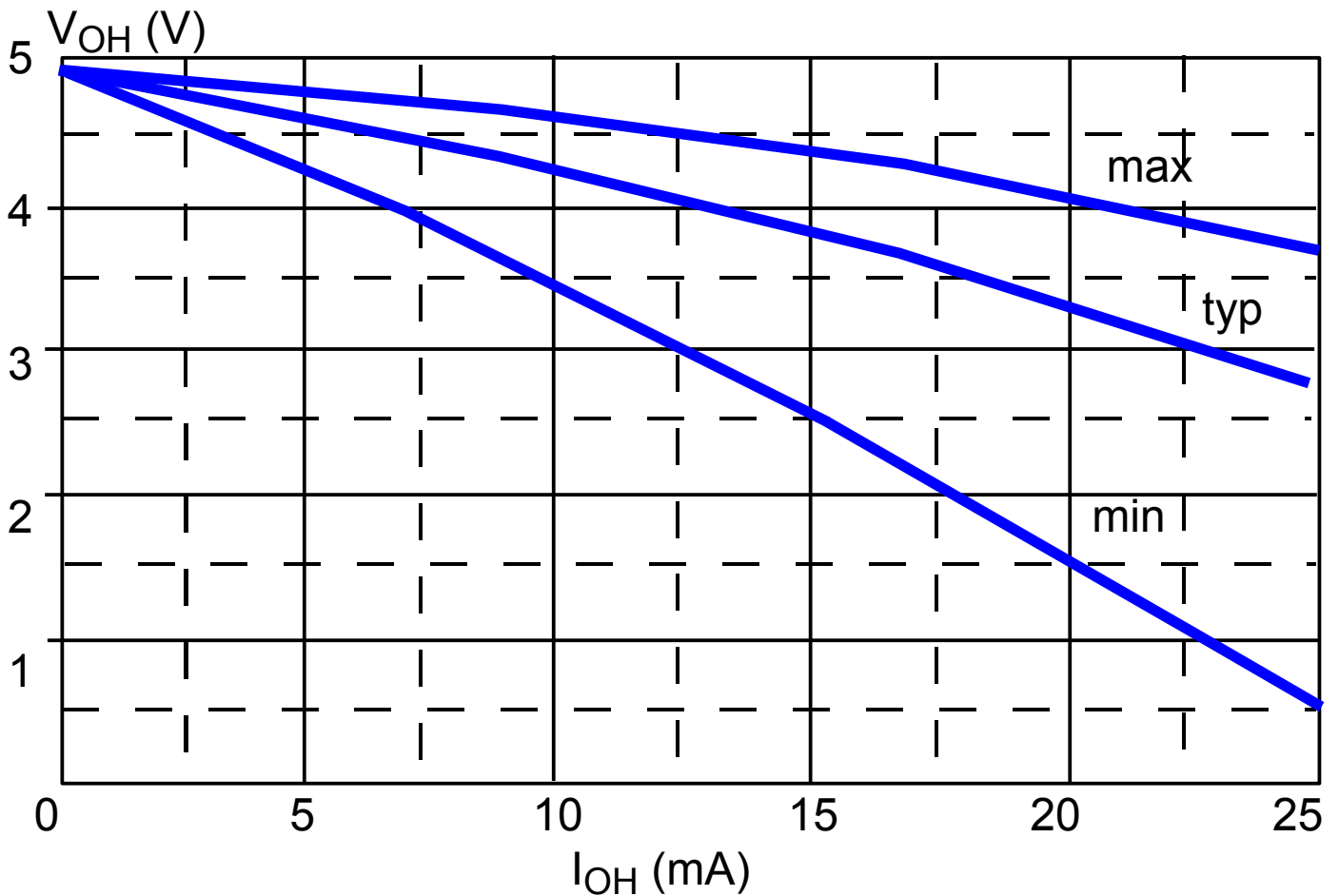
Les PORTs du PIC 16F887

Cinq ports :

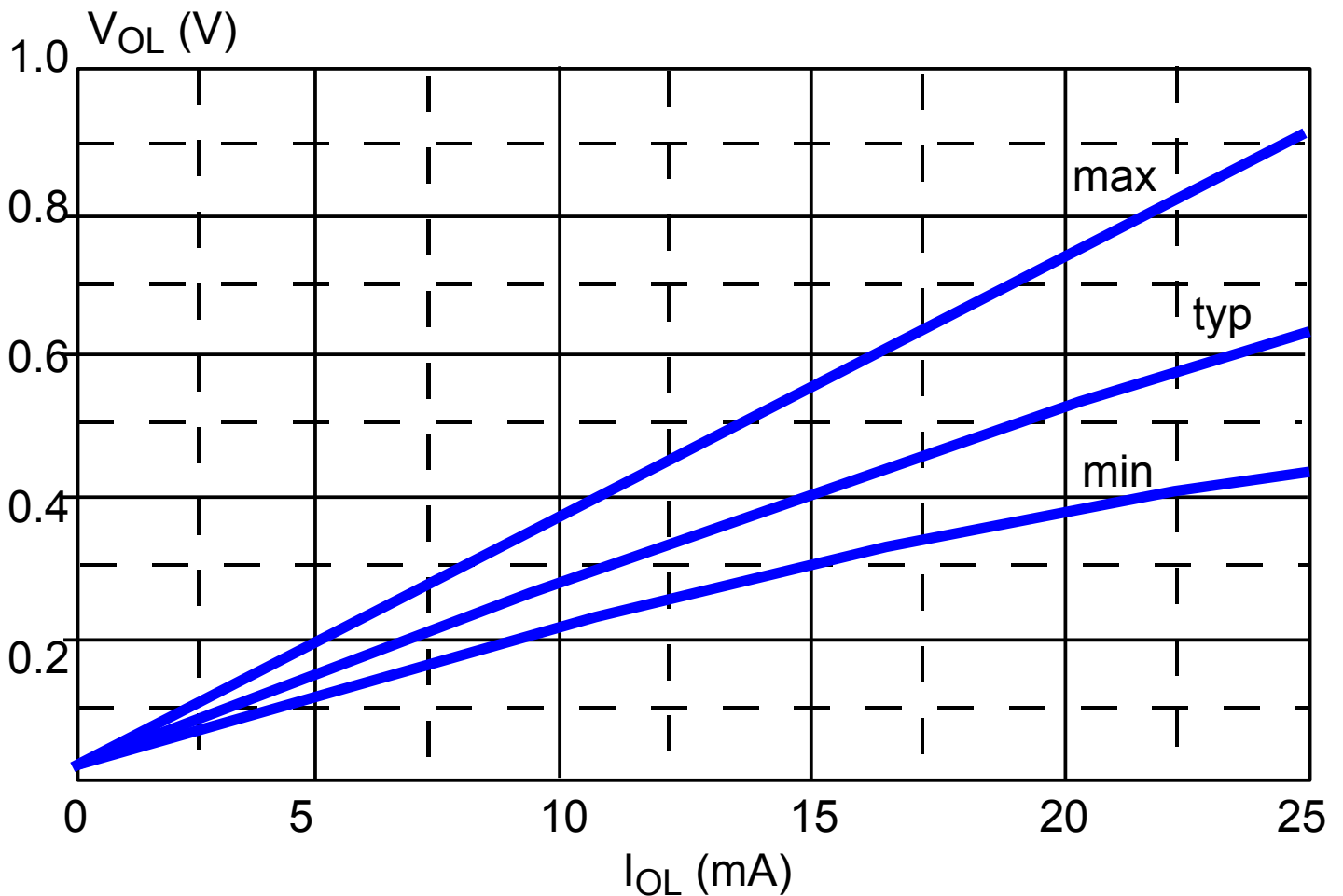
- PORTA 8 bits $RA_0 \rightarrow RA_7$
- PORTB 8 bits $RB_0 \rightarrow RB_7$
- PORTC 8 bits $RC_0 \rightarrow RC_7$
- PORTD 8 bits $RD_0 \rightarrow RD_7$
- PORTE 2 bits $RE_0 \rightarrow RE_2$

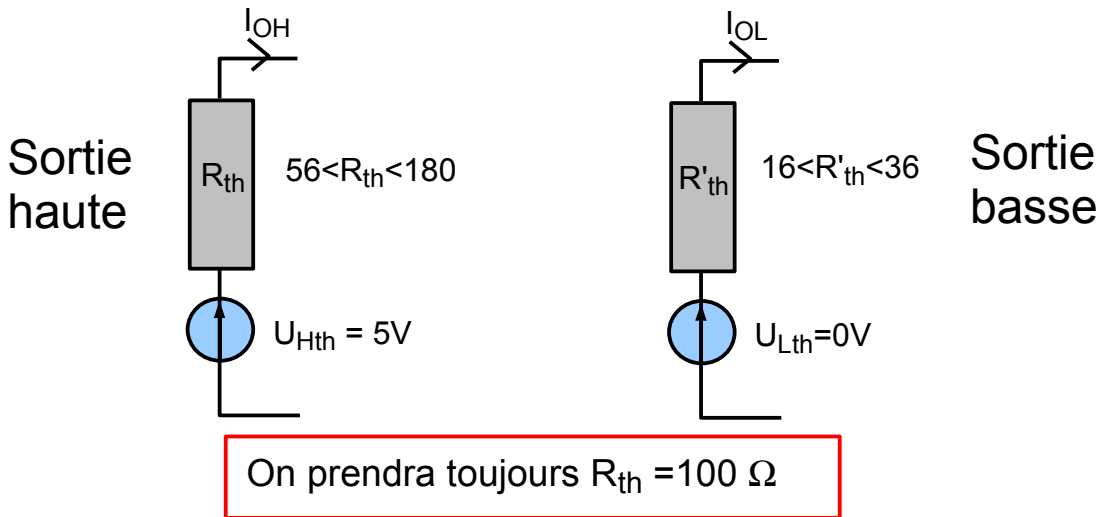


Les modèles électriques des PORTs



Les ports du 16F84 sont supposés délivrer ou recevoir jusqu'à 25mA

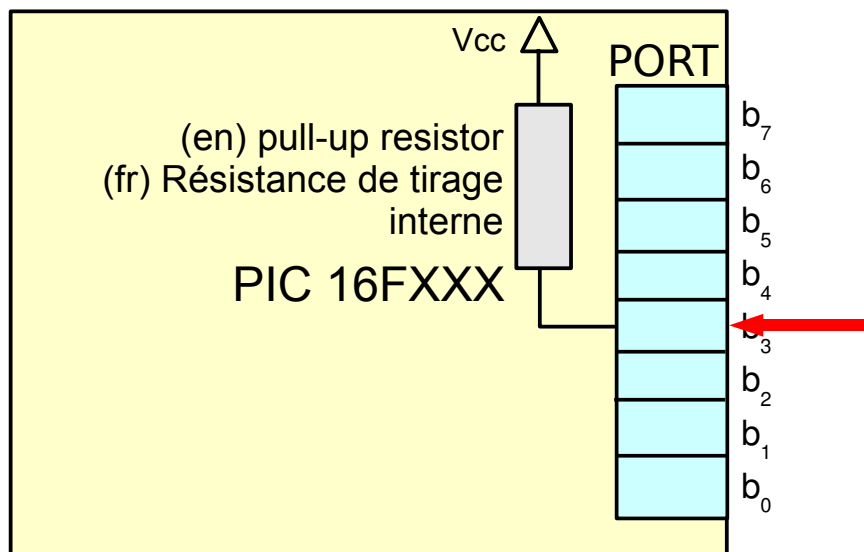




Après avoir déterminé les modèles de Thévenin des PORTs on s'intéresse maintenant aux résistances de tirage.

Les résistances de tirage des PORTs

A quoi cela sert ? On les utilise en général pour les entrées et cela peut éviter d'utiliser ces mêmes résistances en externe.



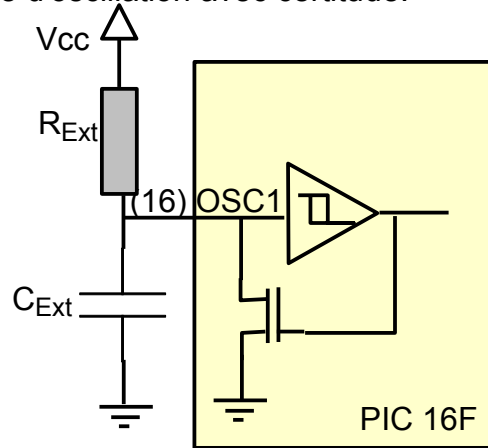
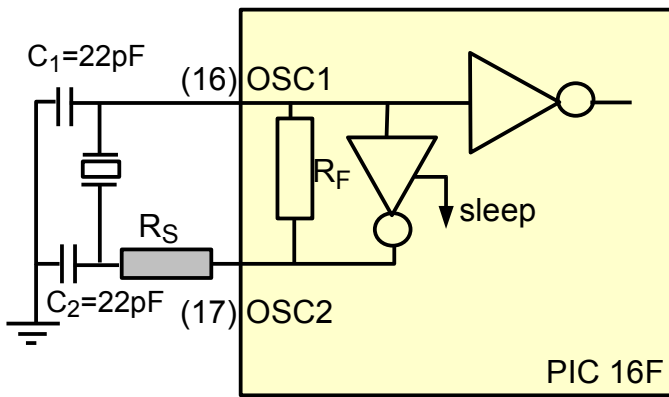
Sur PIC pull-up avec bit RPBU du registre **OPTION**

Cours 3 - 4 Interfaces externes des micro-contrôleurs

I) Horloge

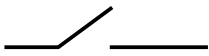
Le 16F84 peut être configuré pour 4 modes d'horloge :

- XT- crystal : mode standard pour 1 – 4 MHz
- HS – High Speed aux alentours de 4 MHz : grande consommation de courant
- LP - Low Power fréquences basses (32,768 kHz jusqu'à 200 kHz)
- RC Resistor Capacitor difficile de prévoir la fréquence d'oscillation avec certitude.

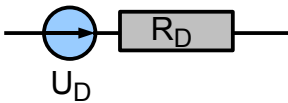


II) Interfacer des LEDs

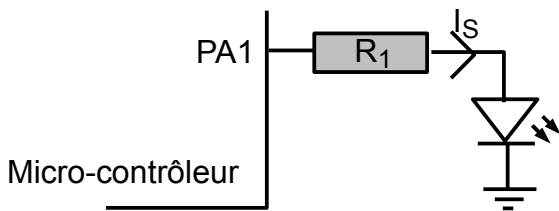
Rappel :
Diode bloquée



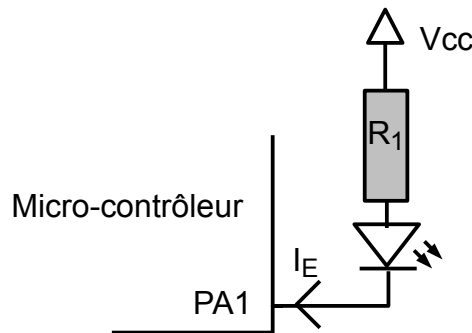
Diode passante



U_D ou U_0 : tension de seuil
Si U_D inconnue prendre 1,8V.
 R_D : résistance dynamique
Si R_D inconnue prendre 0Ω .



Micro-contrôleur comme source de courant



Micro-contrôleur comme puits de courant

ATTENTION les ports ne sont pas forcément symétriques : I_S peut être différent de I_E .

Pour information $V_{OH}=4V$ pour $I_D=10mA$ et $V_{OL} = 0,4V$ pour $I_D=10mA$ pour le PIC 16F84.

Calculer les modèles de Thévenin correspondants.

Si on ne connaît rien on prend $V_D = 1,8V$ et $V_{OL} = 0V$ et $V_{OH}=V_{cc}$.

Calcul de R_1

Source de courant : $R_1 = (V_{OH}-V_D) / I_D$ avec V_D tension directe de la diode et V_{OH} tension haute du port.

Puits de courant : $R_1 = (V_{cc}-V_D-V_{OL}) / I_D$ avec V_{OL} tension basse du port

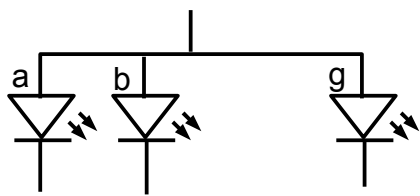
Exemple :

Kingbright led rouge 10mA $I=627nm$ 15mcd à 10mA $V_D = 1,95V$

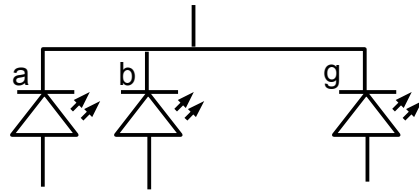
Kingbright led vert 10mA $I=565nm$ 12mcd à 10mA $V_D = 2,09V$

Pour information les résistances utilisées sur EasyPIC5 sont de $1k\Omega$.

Pour généraliser l'utilisation des LEDs, on peut s'intéresser aux afficheurs sept segments



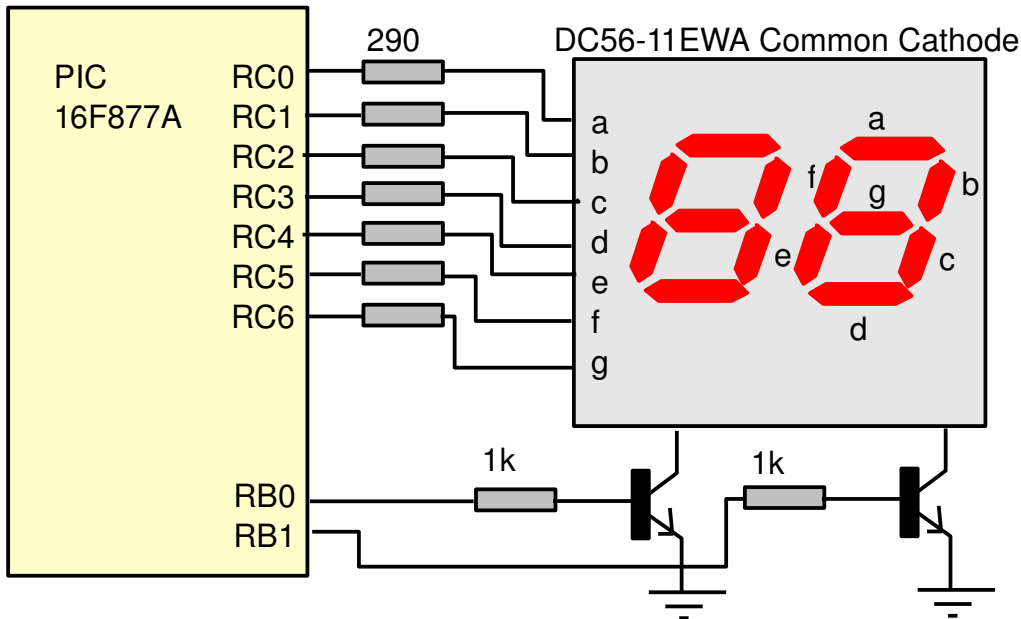
Anode commune



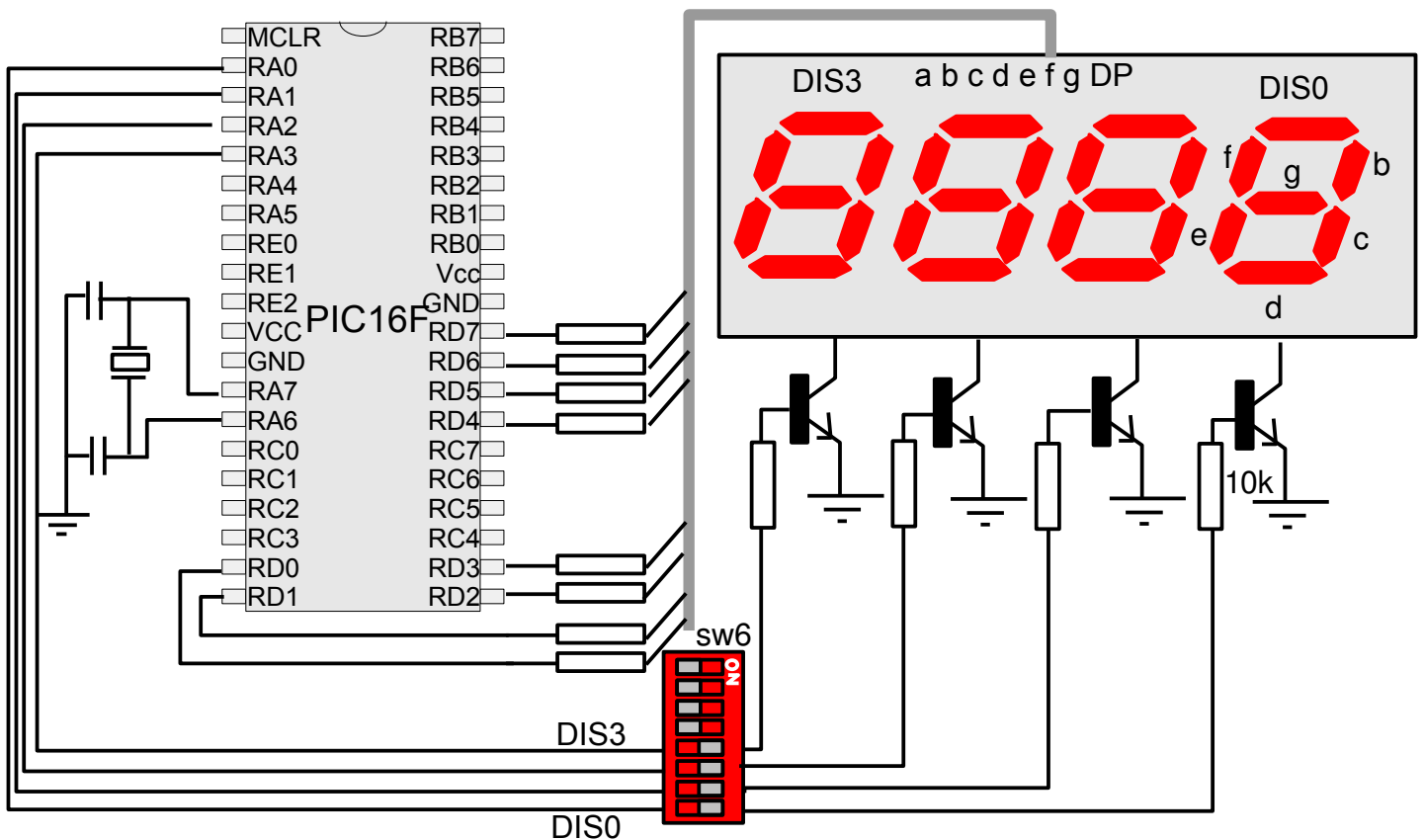
Cathode commune

Comment allume-t-on un afficheur anode commune, un afficheur cathode commune ?

Multiplexage de plusieurs afficheurs :



EasyPIC5

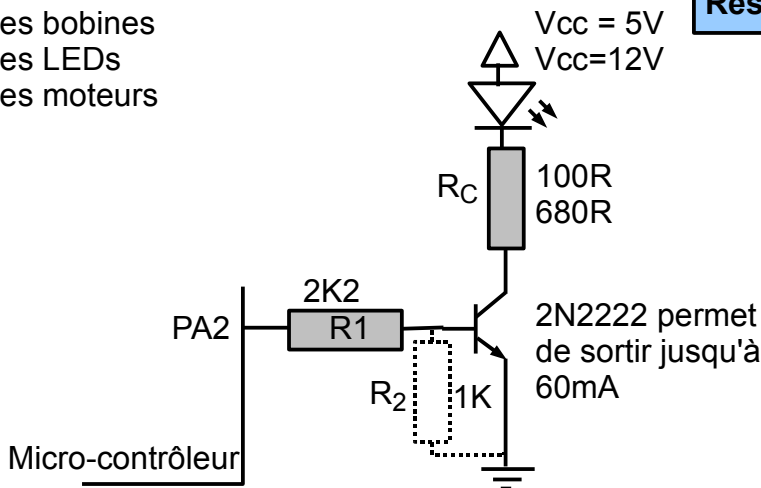


III) Interfacer des sorties puissantes

Pour commander des sorties puissantes il faut ajouter des transistors (BJT Bipolar Junction Transistor). Les sorties puissantes peuvent être :

- des relais
- des bobines
- des LEDs
- des moteurs

Résistance pull-up = résistance de tirage

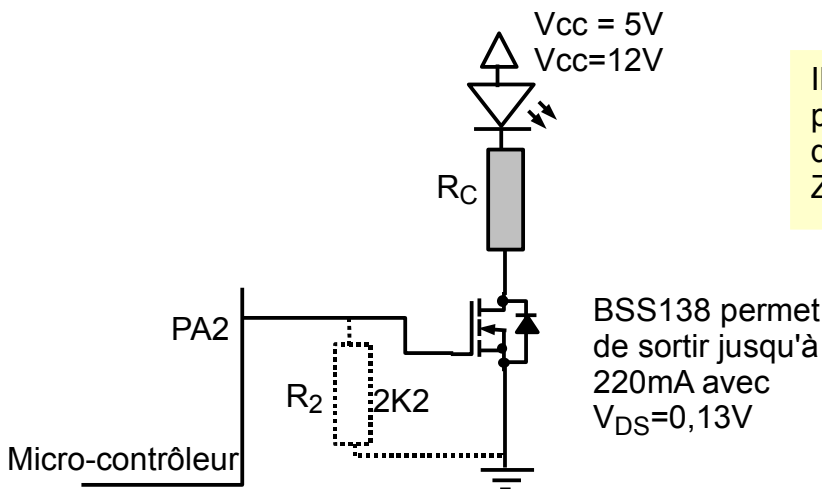


Certains micro-contrôleurs ont une résistance interne pull-up par défaut à la mise sous tension. Le transistor est alors passant. Pour éviter cela on peut ajouter la résistance R2.

Problème des Pull-Up internes pour les entrées
Sur PIC pull-up avec bit RPBUS du registre **OPTION** initialisé à 1 à la mise sous tension mais ????

IV) Interfacer des sorties puissantes avec des FET

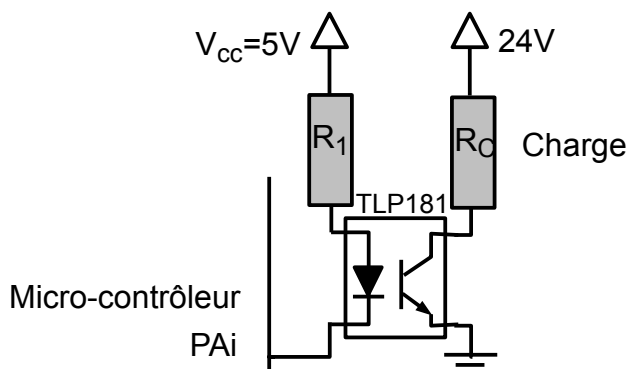
On peut utiliser des transistors à effet de champ à la place des BJT. Nous examinons le cas des MOSFET.



Il existe des familles spécialement faites pour être commandées directement par des PORTs. Voir ZVN4206A et ZVN4306A de chez Zetex.

V) Encore plus de puissance

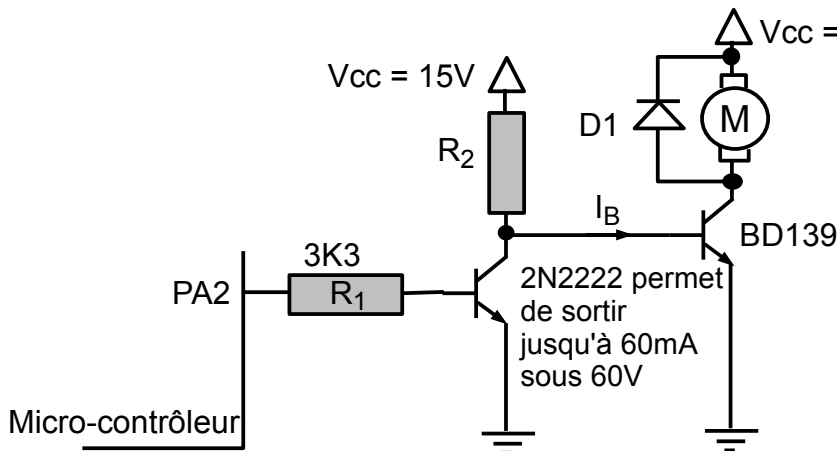
Il faut utiliser des opto-coupleurs. Ils permettent aussi une isolation galvanique ce qui est important si la tension d'alimentation est très différente de celle du micro-contrôleur.



TLP181 :
 $16\text{mA} < I_{D\text{max}} < 20\text{mA}$ avec $V_D < 1,3\text{V}$ à 10mA
 $1\text{mA} < I_C < 10\text{mA}$
Dimensionner R_1 .

VI) Commander des bobines moteurs et relais avec BJT

Il existe des transistors de puissance adaptés. Par exemple le transistor BD139 propose un gain β compris entre 25 et 250. On peut l'utiliser avec un courant $I_C=1,5\text{ A}$ (3A en pointe). Son faible gain nécessite un courant I_B assez important, allant au-delà des possibilités des ports traditionnels (typiquement 10mA). Par exemple pour commuter 1A, il faut un $I_B=40\text{mA}$ dans le pire des cas. Il faut donc un transistor supplémentaire.



Le transistor 2N2222 est bloqué, il circule 40mA dans I_B

1°) Si R_2 est choisi à $330\ \Omega$, quelle puissance passe dans R_2 ?

2°) Si le transistor 2N2222 est saturé ($V_{CEsat}=0,2\text{V}$) quelle puissance passe dans R_2 ?

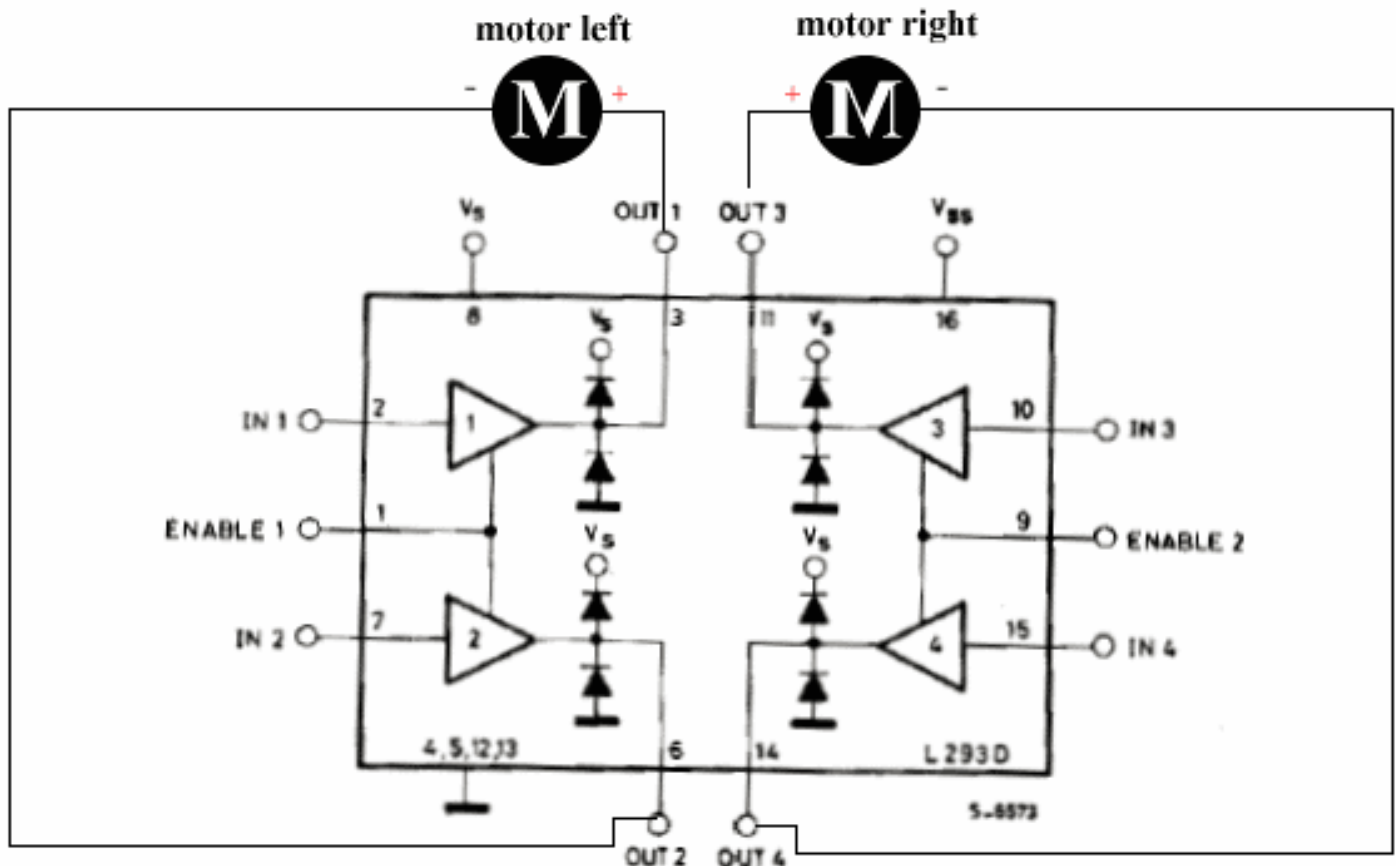
3°) Lorsque le BD139 est saturé, il circule $I_C=1\text{A}$ et l'on a $V_{CEsat}=0,70\text{V}$. Quelle puissance est dissipée dans le BD139 ?

VII) Quelques circuits spécialisés

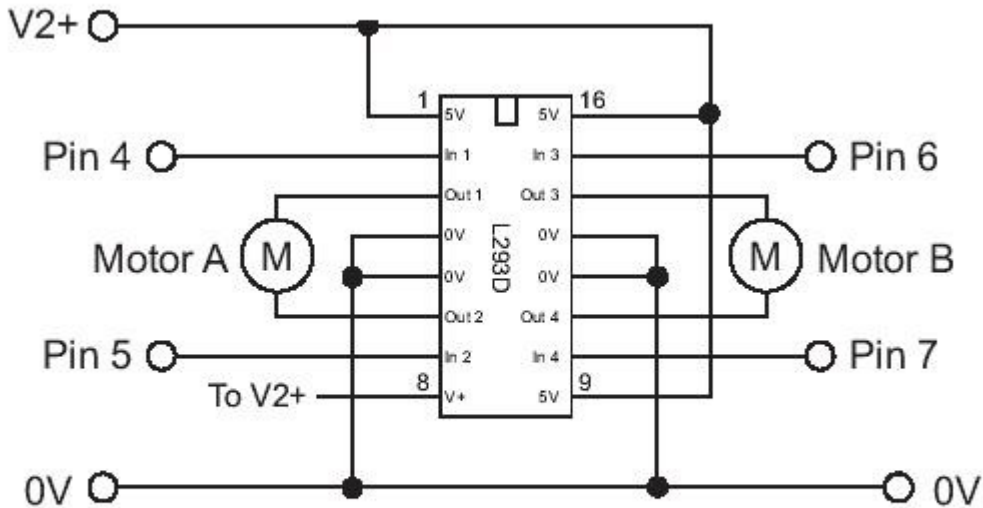
On peut trouver des circuits spécialisés destinés à commander directement des moteurs.

Le circuit L293D de chez Texas

Voici le schéma interne :



How to control DC motors with an H-Bridge IC



Both inputs low - motor halt
 First output high, second output low - motor forward
 First output low, second output high - motor reverse
 Both inputs high - motor halt

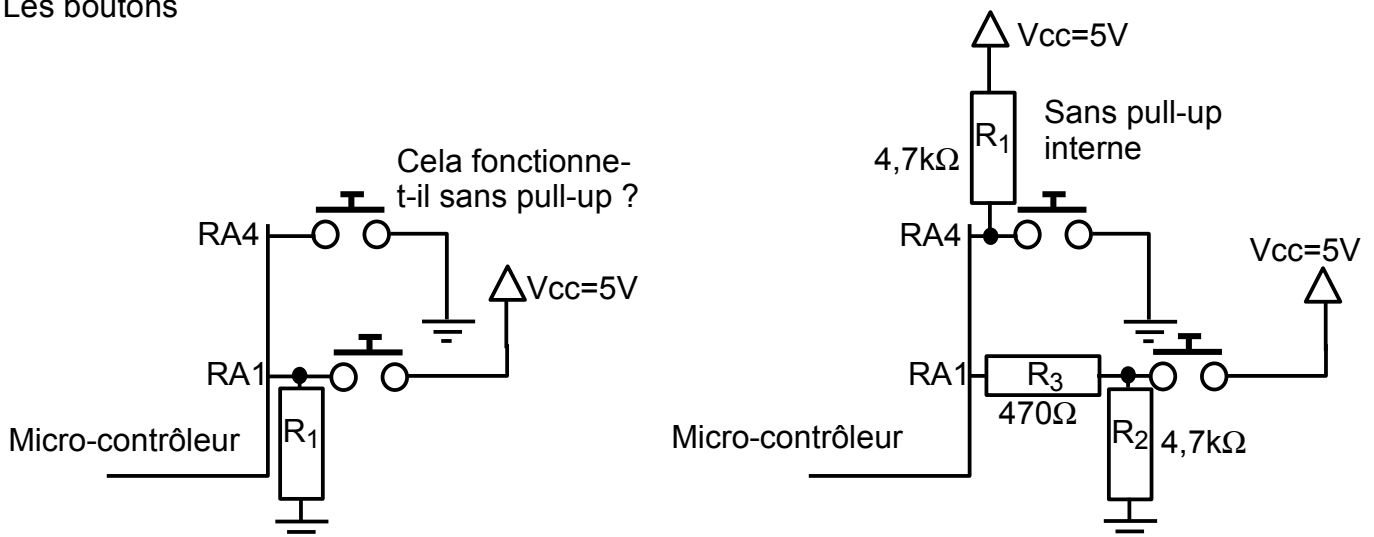
On suppose que

- Pin 4 est reliée à RA3
- Pin 5 est reliée à RA2
- Pin 7 est reliée à RA1
- Pin 6 est reliée à RA0

Comment fait-on avancer un robot motorisé par deux moteurs ? Comment le fait-on reculer ?

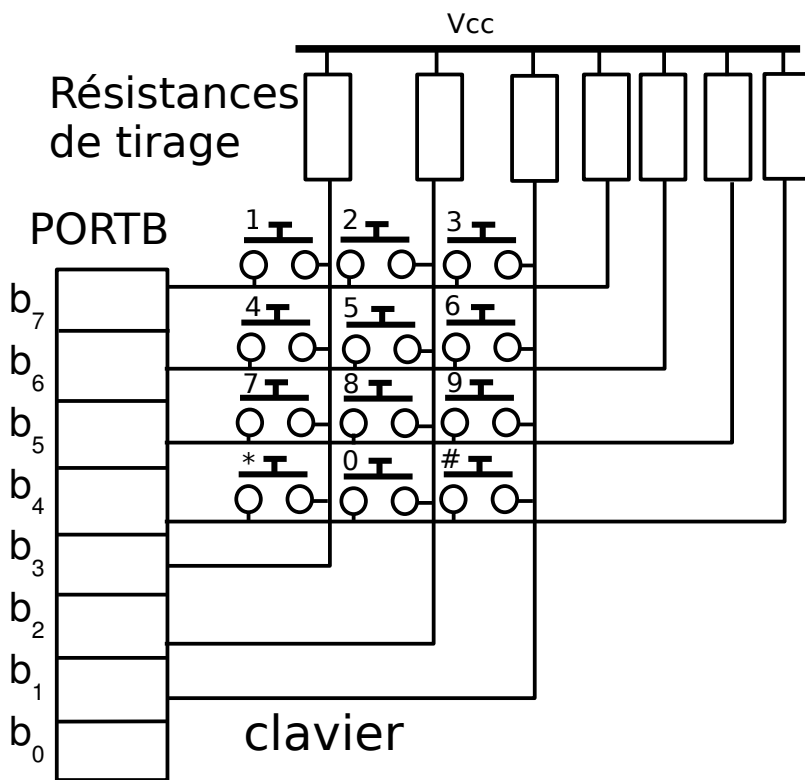
VIII) Interfacer des entrées

- Les boutons

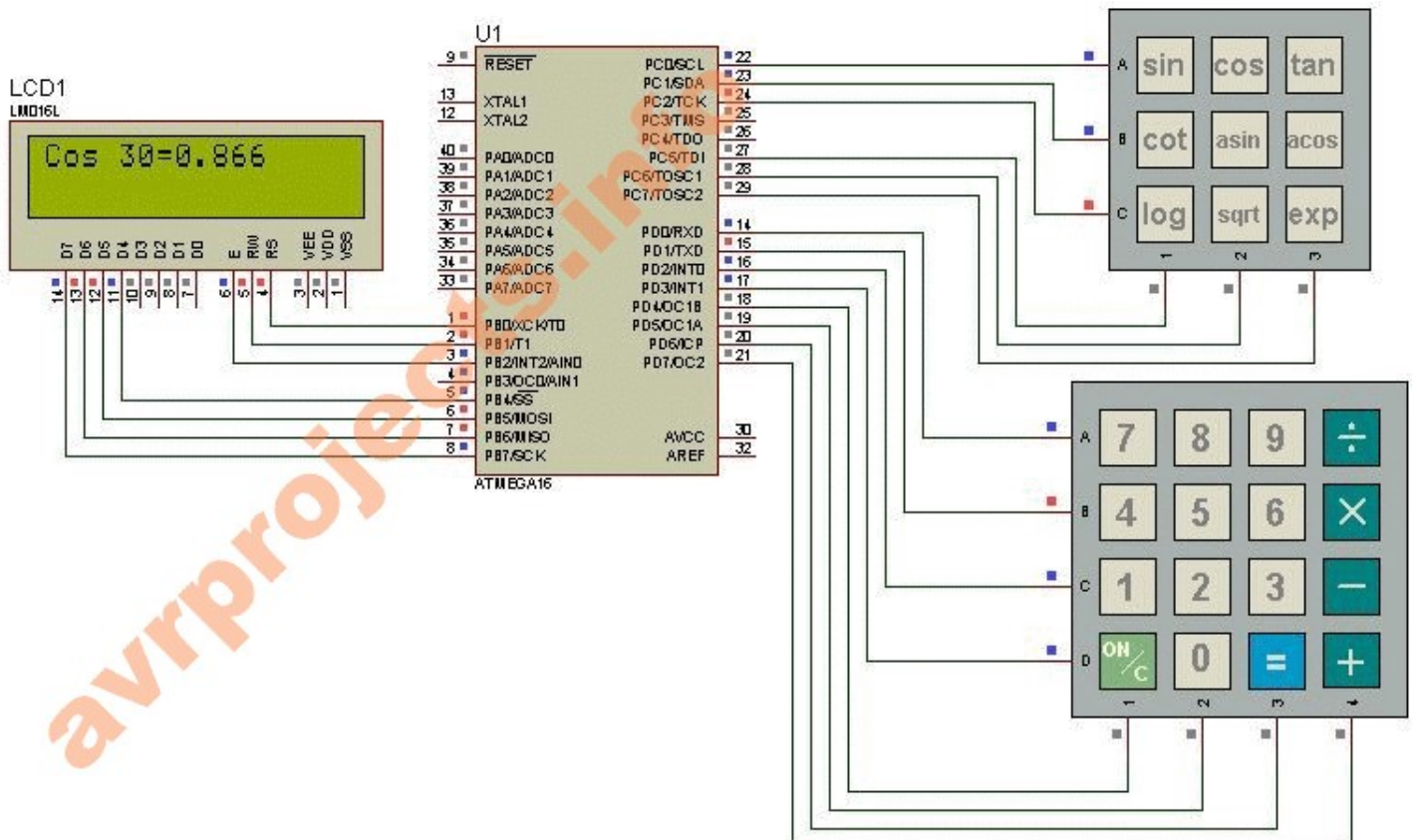


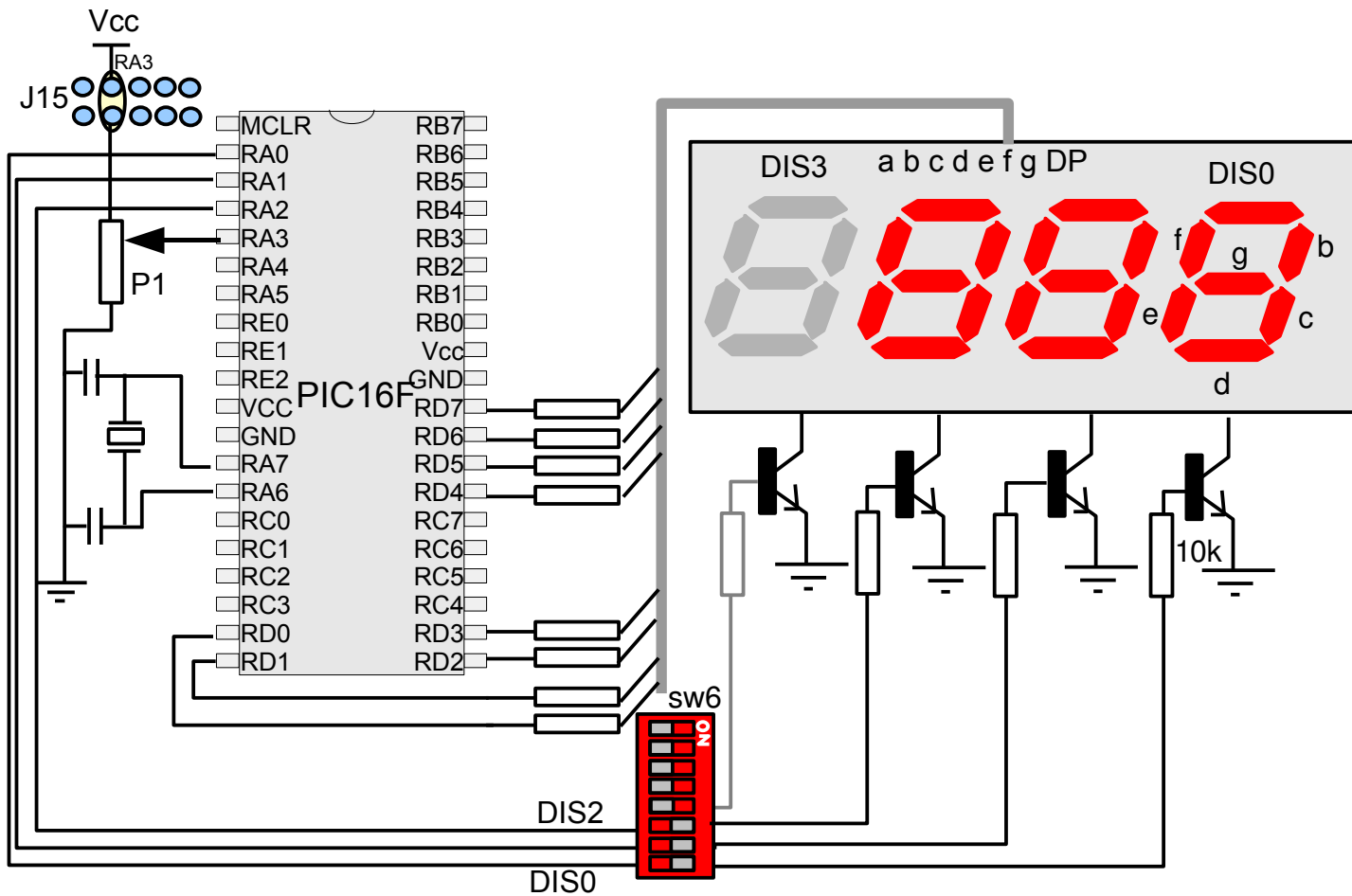
Montage simple qui nécessite cependant un bit par interrupteur. On ne peut pas toujours se le permettre.

Application au clavier



Ajouter interface afficheur LCD

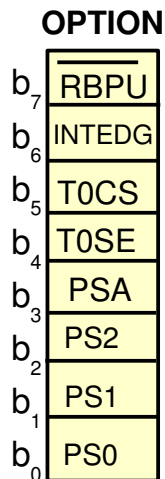




Cours 4-5 : les périphériques internes de la famille 16F

I) Les registres et noms des bits en Mikro C

Documentation technique



Fichier entête Mikro C

```
/**MikroC fichier P18F84A.c ***  
//***** bits de OPTION  
NOT_RBPU    = 0x0007,  
INTEDG      = 0x0006,  
T0CS        = 0x0005,  
T0SE        = 0x0004,  
PSA         = 0x0003,  
PS2         = 0x0002,  
PS1         = 0x0001,  
PS0         = 0x0000,  
...
```

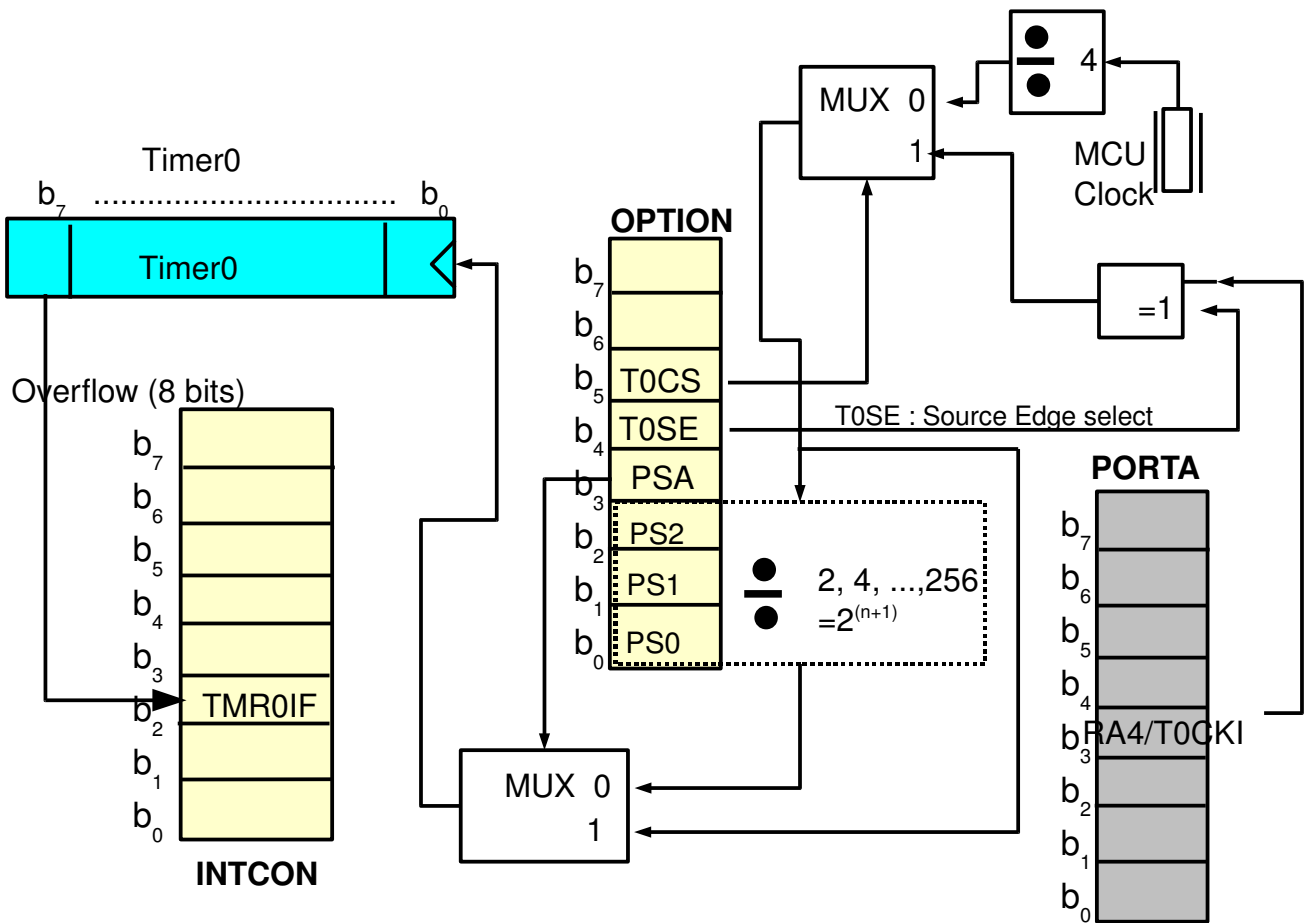
Remarque : les noms de la documentation et les noms du fichier d'entête ne correspondent pas toujours !

Les trois façons de programmer en Mikro C

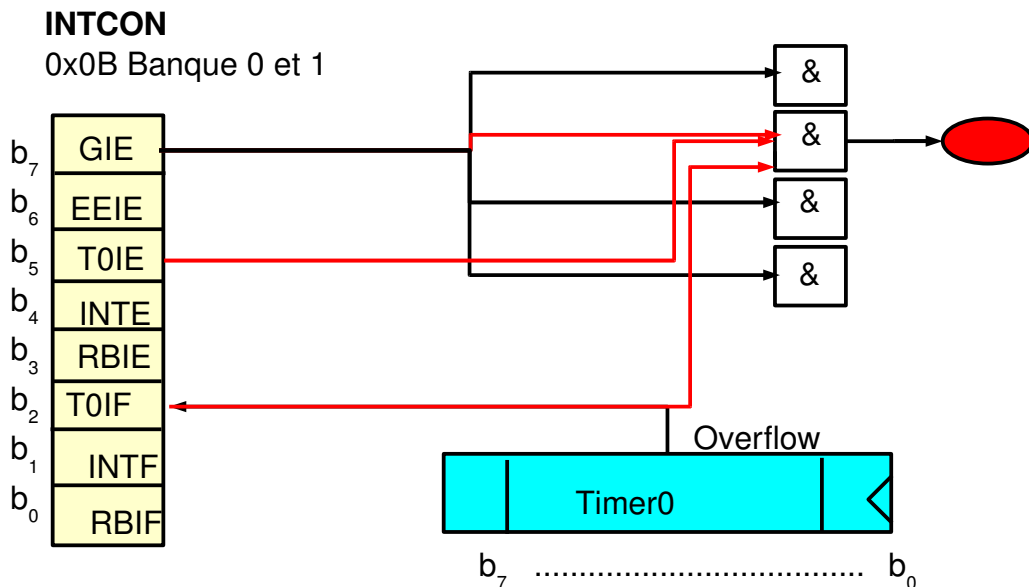
Fichier programme Mikro C

```
Void main( void) {  
....  
/**/ Toujours ****/  
    OPTION_REG.B3 =1;  
/**/ Si on connaît le nom  
    OPTION_REG.PSA =1;  
// **** ou encore  
    PSA_bit = 1;  
...  
}
```

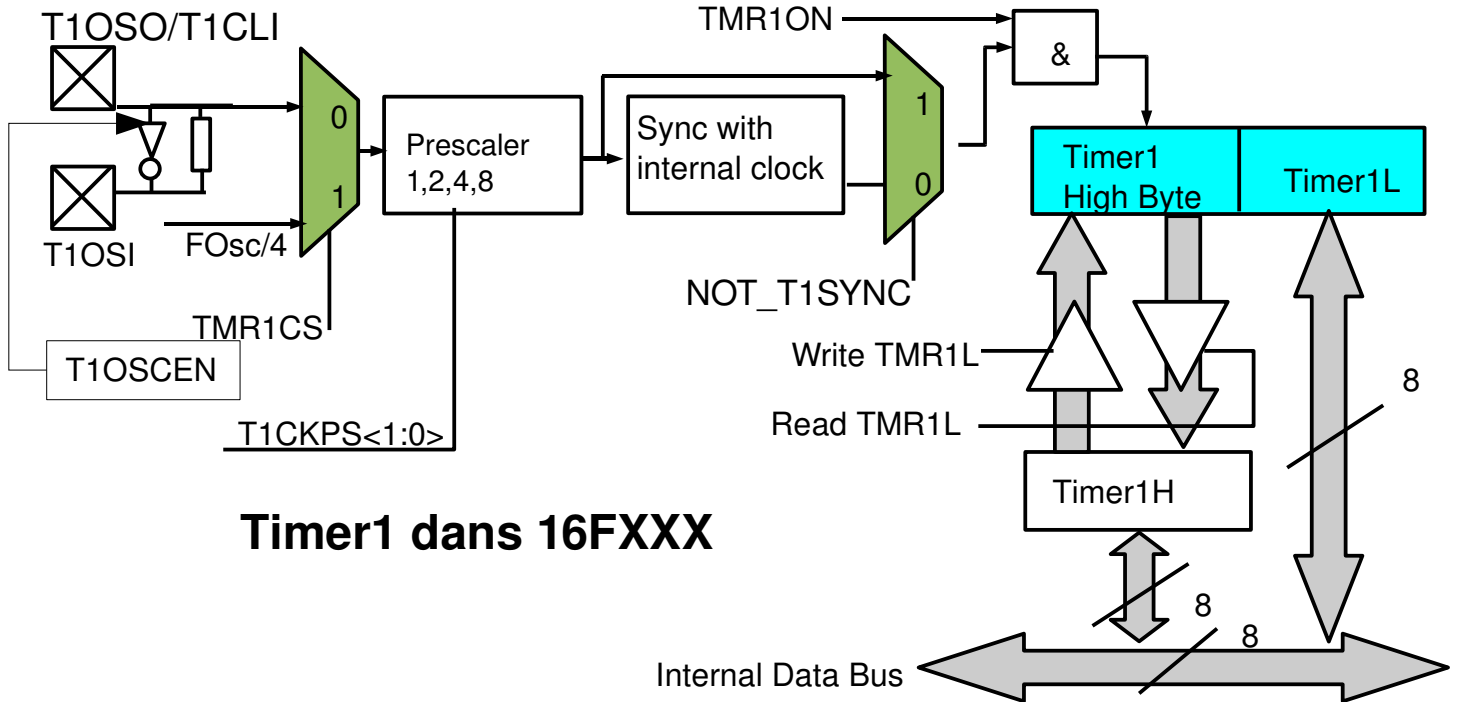
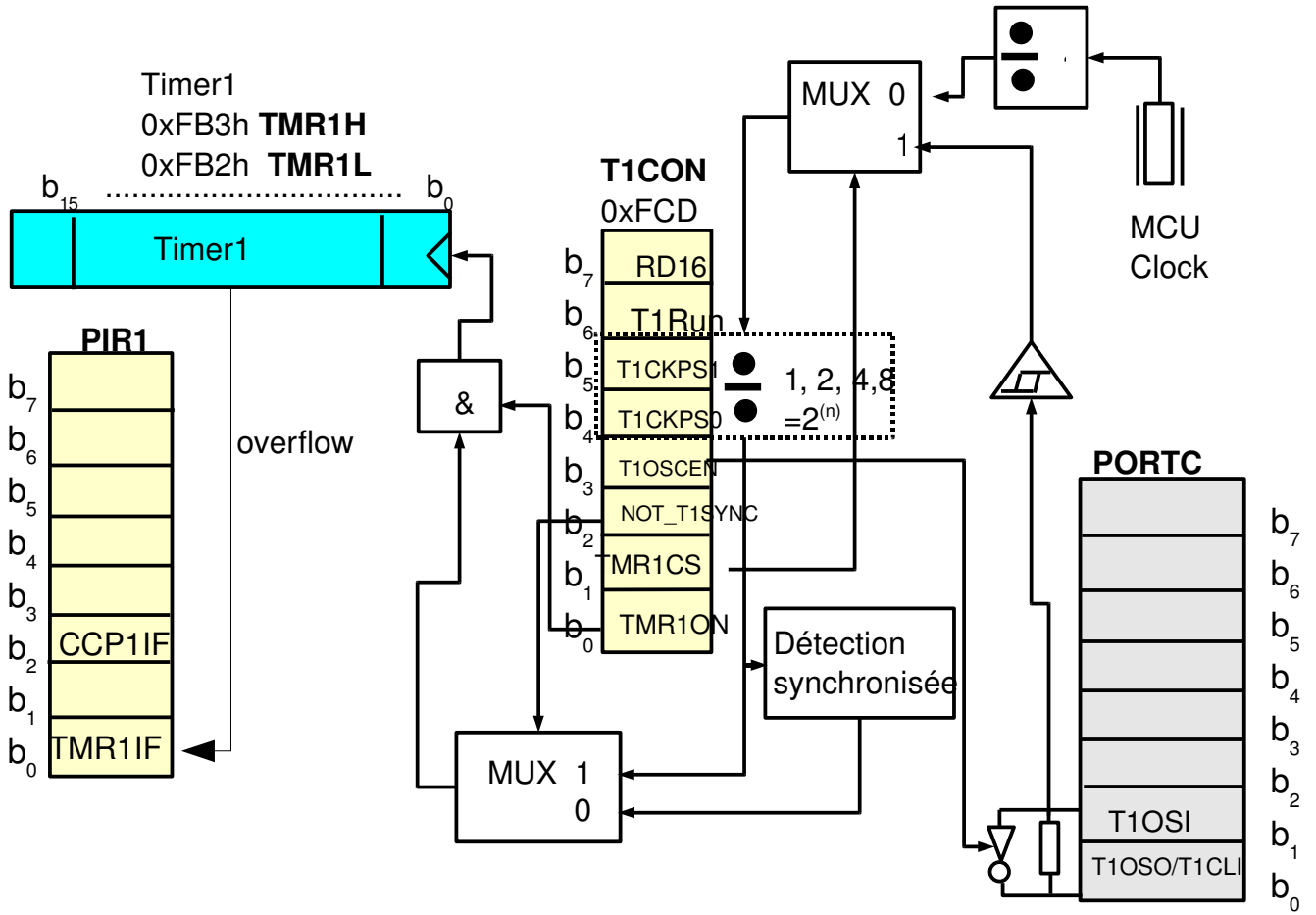
II) Timer0 dans 16FXXX



III) Interruption timer0

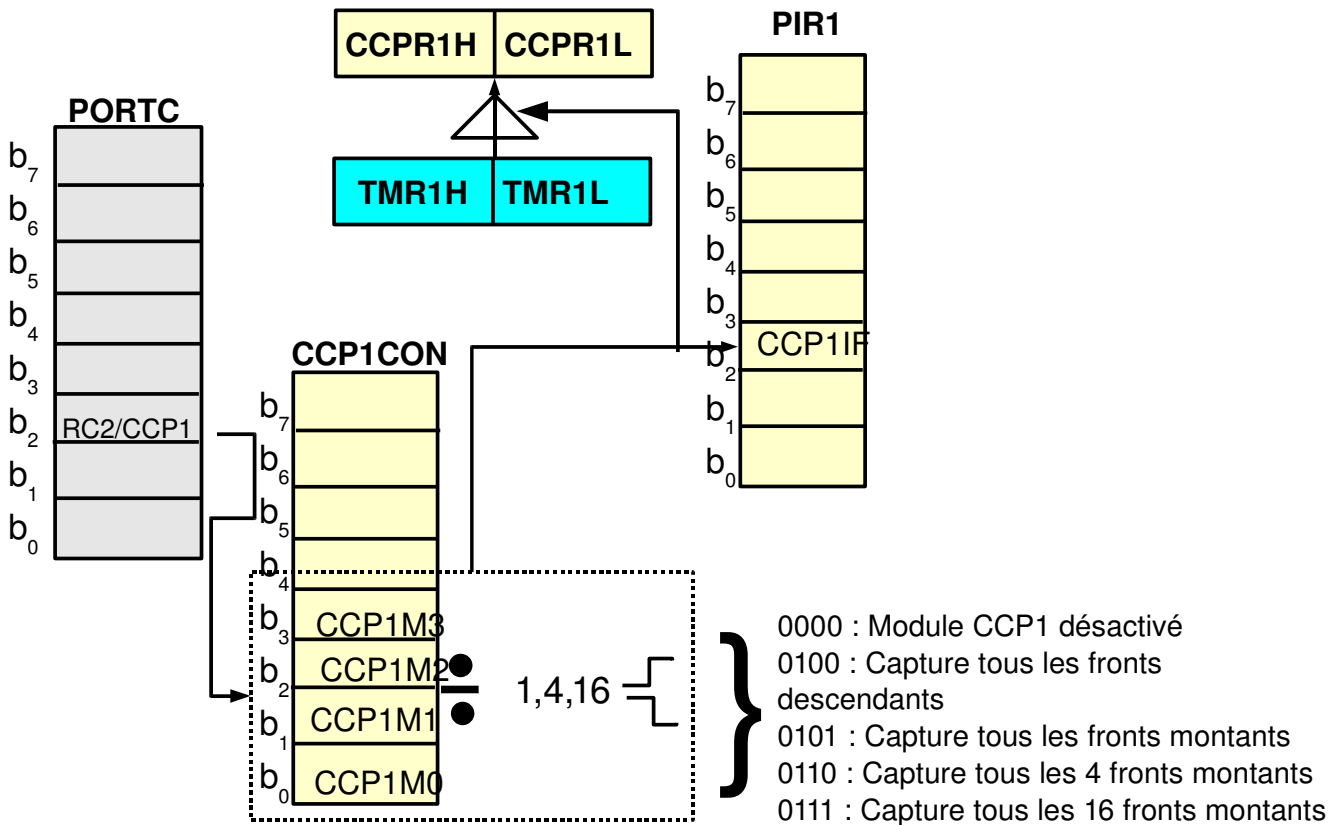


IV) Timer1 dans 18F4550 et 16F877A



Timer1 dans 16FXXX

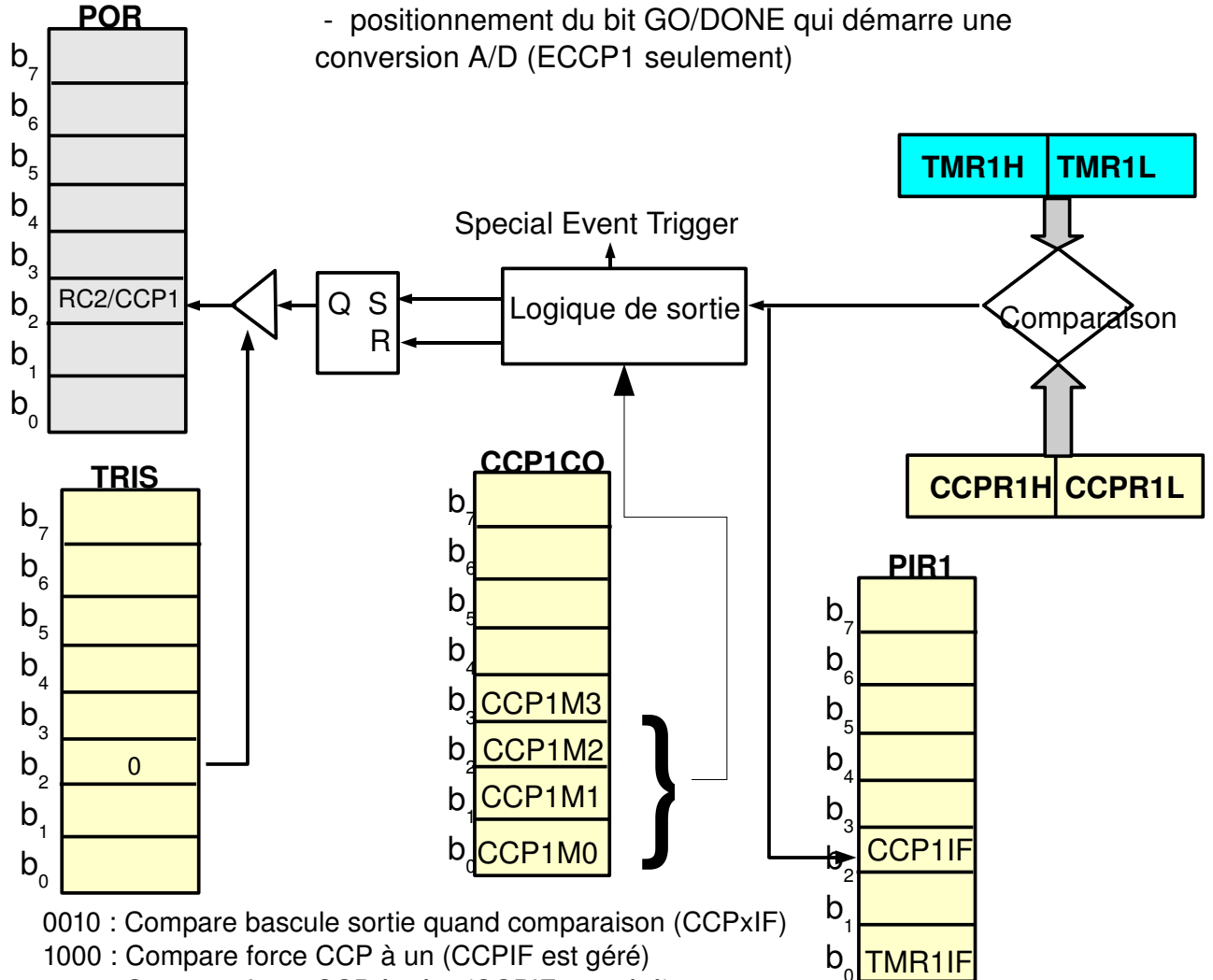
V) CAPTURE POUR 16F877



VI) COMPARAISON PIC 16F877

Special Event Trigger déclenchera:

- un Reset du Timer1 (mais pas un positionnement du drapeau d'interruption de Timer1)
- positionnement du bit GO/DONE qui démarre une conversion A/D (ECCP1 seulement)



0010 : Compare bascule sortie quand comparaison (CCPxIF)

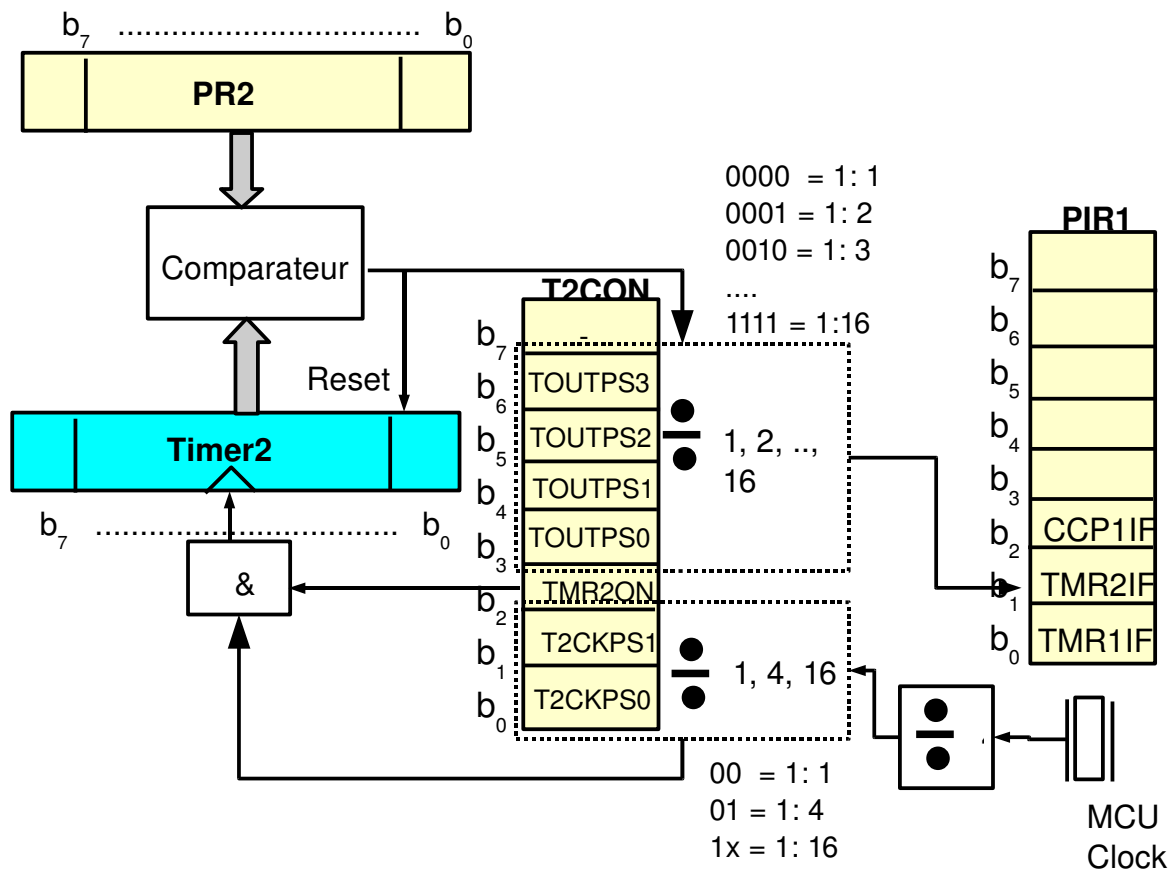
1000 : Compare force CCP à un (CCPIF est géré)

1001 : Compare force CCP à zéro (CCPIF est géré)

1010 : Compare sort rien sur CCP mais sur CCPIF

1011 : Compare force Special Event Trigger reset sur timer1 (CCP1IF est géré)

VII) Timer2 dans 18F4550 et 16F877A



VIII) PWM dans 18F4550 et 16F877A

