

Nom	BitO	Type de buffer	Fonction
RBO/INT	bit 0	TTL	entrée/sortie ou entrée d'interruption externe sur les fronts (avec pull-up)
RB1	bit 1	TTL	entrée/sortie (avec pull-up)
RB2	bit 2	TTL	entrée/sortie (avec pull-up)
RB3	bit 3	TTL	entrée/sortie (avec pull-up)
RB4	bit 4	TTL	entrée/sortie (avec interruption) (avec pull-up)
RB5	bit 5	TTL	entrée/sortie (avec interruption) (avec pull-up)
RB6	bit 6	TTL/ST	entrée/sortie (avec interruption) (avec pull-up)
RB7	bit 7	TTL/ST	entrée/sortie (avec interruption) (avec pull-up)

Tableau 2 : Résumé des fonctions de la porte B.

temporairement le programme pour exécuter différentes opérations. La ligne RBO peut également donner origine, en présence d'un front montant ou descendant, à une interruption extérieure.

Les interruptions sont activées et désactivées à travers un registre spécial dénommé INITCON, que nous analyserons en détail plus tard.

Enfin, nous vous rappelons que les deux lignes RB6 et RB7 sont utilisées en phase de programmation du dispositif. C'est, en effet, à travers ces deux lignes que se produit le transfert, en forme série, de l'ensemble des données que le microcontrôleur insère dans sa mémoire programme.

Le Timer

Le timer (l'horloge) intégré dans le PIC16F84 (dénommé TMRO), est constitué d'un prescaler (prédiviseur) de 8 bits et du timer lui-même, caractérisé également par 8 bits. Le timer peut fonctionner en deux modes distincts, qui sont déterminés par la valeur du bit D5 dans le registre OPTION, bit que l'on appelle TOCS.

Etudions ces deux modes

Mode TIMER :

On le sélectionne en mettant à 0 le bit TOCS. Dans ce type de fonctionnement, le timer est alimenté par son horloge interne dont la fréquence est égale à celle de l'horloge du microcontrôleur divisée par 4.

Mode COUNTER :

On le sélectionne en mettant à 1 le bit TOCS. Dans ce mode, le timer incrémente son propre comptage à chaque front (de montée ou de descente) présent sur la patte RA4. Pour déterminer si cette incrémentation doit se produire

sur le front positif ou sur le négatif, il faut agir sur le bit D4, toujours dans le registre OPTION, bit que l'on appelle également TOSE. Si ce bit est mis à 0, l'incréméntation est effectuée sur le front positif, et vice versa, s'il est à 1, l'incréméntation est opérée sur le front négatif.

Le timer peut être lu et modifié à tout moment. Il est en effet situé à l'adresse 1 parmi les registres d'utilisation spéciale.

Le prescaler

Le prescaler est un dispositif qui sert à diviser la fréquence qui va piloter le véritable compteur et permet donc d'obtenir des intervalles de temps relativement longs.

Le prescaler peut être connecté, aussi bien au timer TMO, qu'au watchdog (circuit de surveillance). Pour relier le prescaler à l'un ou à l'autre, il vous suffira d'agir sur le bit D3 du registre OPTION, que l'on nomme PSA. Si ce

bit est mis à 0, le prescaler est relié au TMRO, s'il est mis à 1, il sera relié au watchdog. Le rapport de division du prescaler est déterminé à travers trois bits, nommés PS0, PS1 et PS2 du registre OPTION, suivant le tableau 3. Rappelez-vous qu'il n'est pas possible de lire ni d'écrire la valeur du prescaler (c'est-à-dire du comptage qu'il est en train d'effectuer) et que ce registre est remis à 0 à chaque fois que vous effectuez une opération d'écriture dans le registre TMRO.

L'interruption générée par le TMRO

Le timer TMRO est donc un compteur qui incrémente sa propre valeur. Il est piloté par l'horloge, qui commande également le microcontrôleur, ou par des fronts montant ou descendant présents sur l'entrée RA4.

Le timer est très souvent utilisé pour générer des intervalles de temps précis, en jouant justement sur la fréquence de l'horloge et sur le rapport de division introduit par le prescaler. Pour utiliser le timer de cette façon, nous vous conseillons de travailler avec le signal que le timer génère lui-même quand la valeur du registre TMRO passe de FFh à 00h.

En fait, quand le timer arrive à la fin de son comptage, c'est-à-dire à FFh, dans l'incréméntation suivante, le registre TMRO est mis à 00h et le bit D2 du registre INITCON, nommé TOIF, est mis à 1. Ceci détermine également une demande d'interruption au microcontrôleur, qui ira exécuter une routine donnée en réponse à un tel événement.

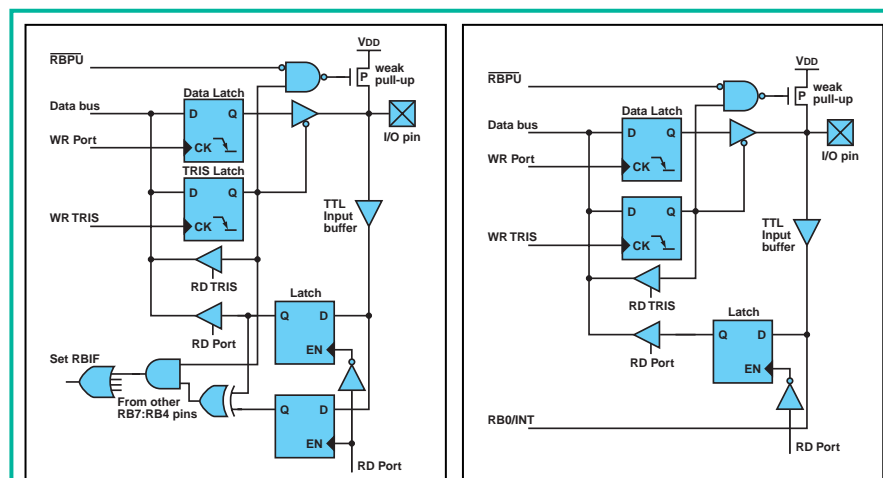


Figure 2 : Cette illustration représente le schéma synoptique des lignes d'entrées RB4 - RB7 (figure 1a) et RBO - RB3 (figure 1b) du port B. Toutes les portes présentent un pull-up interne sélectionnable par software à travers "RBPU" (bit 0 du registre "OPTION"). Les portes RB4 - RB7 peuvent être aussi utilisées comme source d'interruption, déclenchable sur front externe.

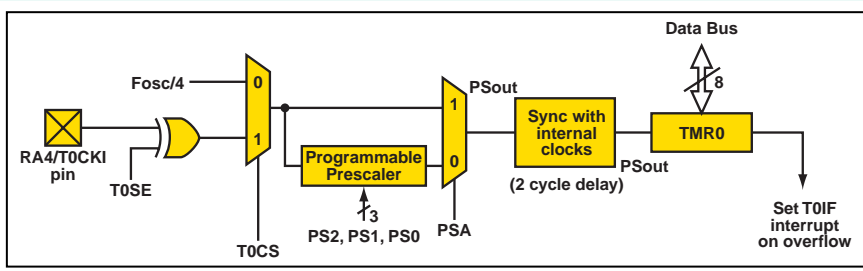


Schéma synoptique du timer TMRO.

Supposons, par exemple, que vous vouliez allumer ou éteindre une LED à intervalles réguliers. Vous devrez charger le prescaler de façon à obtenir l'intervalle de clignotement demandé et faire en sorte qu'à chaque fois qu'une interruption est générée par la fin du comptage effectué par le timer, la routine qui répond à cette interruption aille inverser l'état logique présent sur la patte à laquelle la LED est reliée.

Nous allons voir maintenant, grâce à l'analyse détaillée de toutes les autres sources possibles d'interruption, comment le microcontrôleur répond aux demandes d'interruption.

Les interruptions

Une interruption est un événement qui détermine l'abandon de l'instruction que le microcontrôleur est en train d'exécuter pour aller entreprendre une routine dénommée routine d'interruption. En fait, chaque fois qu'une demande d'interruption est générée par une des quatre sources possibles, le microcontrôleur abandonne temporairement le programme qu'il était en train d'exécuter et saute à une position définie au préalable à l'adresse 0004h de la mémoire programme. De plus, il est possible de lire l'état du registre INITCON afin de déterminer quelle a été l'origine de l'interruption et d'exécuter en conséquence des

opérations en réponse à cette demande.

A la fin de ces opérations, ou plus exactement à la fin du sous-programme d'interruption, il est nécessaire de faire revenir le microcontrôleur à l'exécution abandonnée à cause de l'interruption. Pour permettre cela, avant de sauter à la position 0004h, la valeur du Program Counter, c'est-à-dire du registre qui mémorise l'adresse du positionnement de mémoire de l'instruction qui est en cours d'exécution, est sauvegardée dans une zone particulière de la mémoire nommée "Stack". Au terme de la routine qui "répond" à la demande d'interruption, on récupère dans le Stack, avec l'instruction RETFIE, la valeur du Program Counter et l'on rétablit le flux normal d'exécution du programme.

Le PIC 16F84 a quatre différentes "sources" susceptibles de déterminer une interruption :

- interruption externe sur la patte RBO ;
- interruption générée par le timer TMRO ;
- interruption générée par un changement de valeur logique sur une des pattes RB4 à RB7 ;
- interruption générée par un signal d'écriture de la mémoire EEPROM.

Analyse des sources d'interruption

En ce qui concerne l'interruption générée par la patte RBO, elle est causée par une transition de l'entrée RBO d'un état logique à un autre. Il est possible de faire agir cette interruption soit sur un front montant (passage de 0 à 1), soit sur un front descendant (passage de 1 à 0). L'un ou l'autre de ces modes est sélectionné à travers le bit D6 du registre OPTION. Si ce bit est mis à 1, l'interruption est générée par un front montant, s'il est mis à 0, l'interruption se produit en correspondance avec un front descendant.

L'interruption du timer TMRO est générée, comme nous l'avons déjà vu, lorsque l'opération de fin de comptage effectuée par le compteur du timer se produit.

L'interruption générée par les entrées RB4 à RB7 se produit lorsque l'on relève, sur au moins une de ces pattes, un changement d'état par rapport à la dernière opération de lecture du port qui a été effectuée. Si

PS2	PS1	PS0	Division de fréquence
0	0	0	2
0	0	1	4
0	1	0	8
0	1	1	16
1	0	0	32
1	0	1	64
1	1	0	128
1	1	1	256

Tableau 3 : Détermination du rapport de division du prescaler.

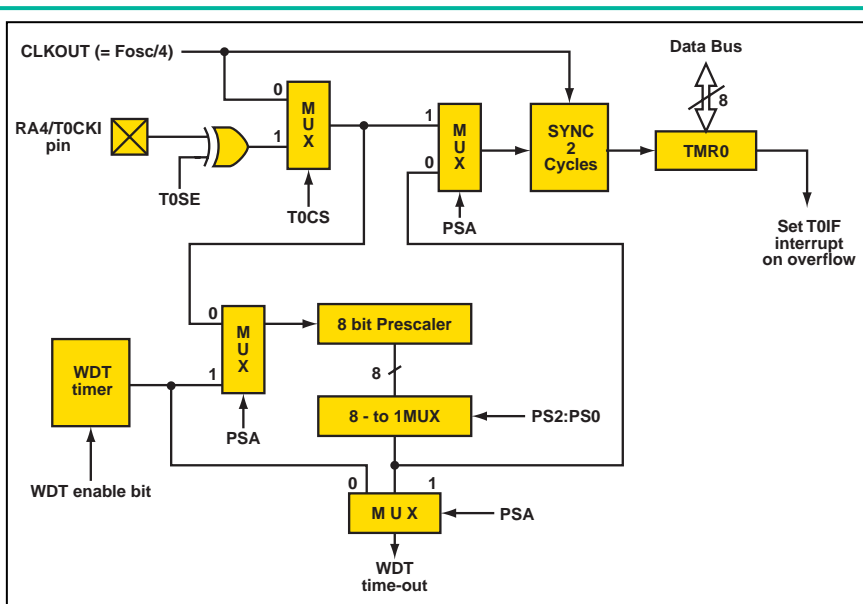


Schéma synoptique du prescaler.