



MICRO INFORMATIQUE 2^{ème} ANNEE



INTRODUCTION

L'objectif de ce cours est la présentation de différentes fonctions avancées des microcontrôleurs:

- Interruptions
- Timers

L'application s'effectue sur le PIC 18f452



LES INTERRUPTIONS

Qu'est ce qu'une interruption?

- Soit un programme qui se déroule normalement. Survient un **événement spécifique**. Le programme principal est **interrompu** (il subit une interruption) et va traiter l'événement. Il reprendra ensuite le programme principal à l'endroit où il avait été interrompu.
- L'interruption est une **rupture de séquence asynchrone**, c'est-à-dire non synchronisée avec le déroulement du programme
- Opposition avec une **rupture de séquence synchrone**, provoquée par le programme lui-même (goto, call, btfss,...).

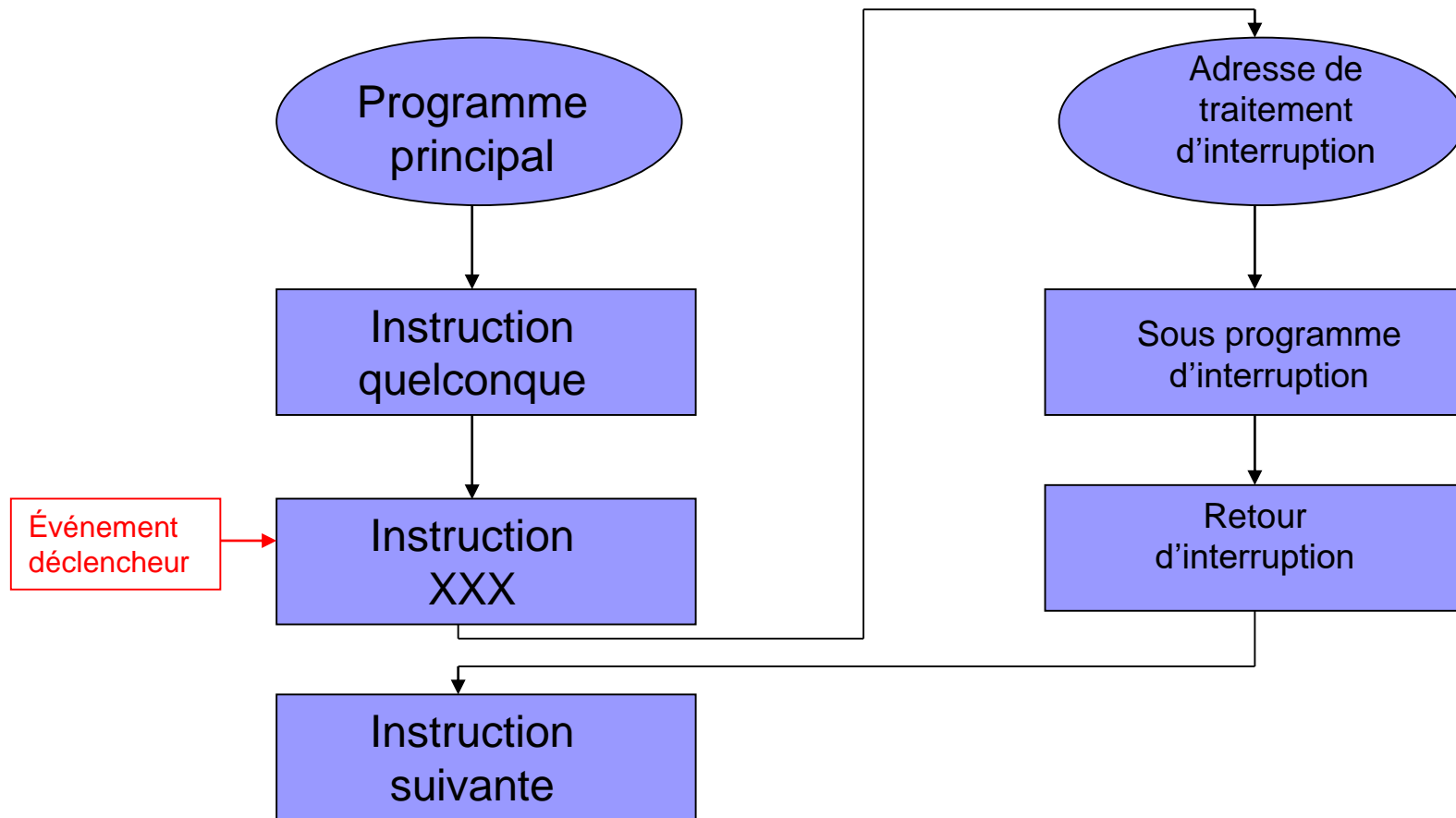
Mécanisme général d'une interruption

- Une routine d'interruption est un sous programme particulier déclenché par un événement spécifique.
- N'importe quel événement ne peut pas déclencher une interruption. Il faut que 2 conditions principales soient remplies:
 - L'événement doit figurer dans la liste des événements susceptibles de provoquer une interruption pour le processeur sur lequel on travaille.
 - Le programmeur doit avoir **autorisé** l'interruption.

Fonctionnement d'une interruption(1)

- Le programme se déroule normalement
- L'événement survient
- Le programme achève l'instruction en cours de traitement
- Le programme saute à l'adresse de traitement de l'interruption
- Le programme traite l'interruption
- Le programme saute à l'instruction qui suit la dernière exécutée dans le programme principal.

Fonctionnement d'une interruption(2)



Mécanisme général d'Interruption sur les PICs (1)

- Le PIC 18f452 dispose de plusieurs sources d'interruptions
- Un système de priorité permet d'attribuer une priorité haute ou basse à une interruption.
- L'adresse de priorité haute est 000008_h ;
l'adresse de priorité basse est 000018_h .

Mécanisme général d'Interruption sur les PICs (2)

- Toute interruption provoquera le saut du programme aux adresses **000008_h** ou **000018_h**
- À ces adresses, le programmeur doit indiquer à quel sous programme il faut sauter en fonction de l'interruption.
- Une interruption ne sauvegarde rien hormis le PC
- Le retour au programme principale s'effectue par l'instruction **RETFIE** en assembleur.
- 10 registres contrôlent les opérations d'interruption:
 - RCON
 - INTCON, INTCON2, INTCON3
 - PIR1, PIR2
 - PIE1, PIE2
 - IPR1, IPR2

LE REGISTRE INTCON

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
Bit 7							Bit 0

Bit 7 **GIE/GIEH**: Global Interrupt Enable

Quand IPEN=0:

1=autorise toutes les interruptions non masquées

0=interdit toute interruption

Quand IPEN=1:

1=autorise toutes les interruptions de priorité haute

0=interdit toute interruption

Bit 6 **PIE/GIEL**: Peripheral Interrupt Enable bit

Quand IPEN=0:

1=autorise toutes les interruptions non masquées

0=interdit toute interruption

Quand IPEN=1:

1=autorise toutes les interruptions de priorité basse

0=interdit toute interruption de priorité basse

Bit 5 **TMR0IE**: TMR0 Overflow Interrupt Enable bit

1=autorise l'interruption due au débordement du Timer0

0=interdit l'interruption due au débordement du Timer0

LE REGISTRE INTCON (suite)

- Bit 4 **INT0IE**: INT0 External interrupt Enable bit
1=autorise l'interruption externe INT0
0=interdit l'interruption externe INT0
- Bit 3 **RBIE**:RB port change interrupt Enable bit
1=autorise l'interruption sur changement du port B
0=interdit l'interruption sur changement du port B
- Bit 2 **TMR0IF**: TMR0 Overflow Interrupt Flag bit
1=demande d'interruption due au débordement du timer 0. Doit être remis à 0 dans le début de SP d'interruption.
0=pas de débordement du timer 0.
- Bit 1 **INT0IF**: INT0 External Input flag
1 = demande d'interruption externe INT0. Doit être remis à 0 dans le début de SP d'interruption.
0 = pas de demande d'interruption externe INT0.
- Bit 0 **RBIF** : RB port change interrupt flag
1 = au moins 1 des E/S RB7, RB6, RB5, RB4 a changé. Doit être remis à 0 dans le début de SP d'interruption.
0 = aucune des E/S RB7, RB6, RB5, RB4 a changé.

Les interruptions externes(1)

- Il existe 4 sources d'interruption externe: RB, RB0/INT0, RB1/INT1 et RB2/INT2
- L'interruption INT0:
 - La demande d'interruption intervient lorsque l'entrée RB0 du PIC passe de 0 à 1 (front montant) si le bit INTEDG0 du registre INTCON2 est à 1 (valeur par défaut). Si le bit INTEDG0 du registre INTCON2 est à 0, la demande est prise en compte sur un front descendant de l'entrée RB0.
 - Lorsque la demande est prise en compte le bit INT0IF du registre INTCON passe à 1.
 - Cette interruption peut être masquée si le bit INT0IE est à 0 ou si le bit GIE/GIEH est à 0.
 - Le bit INT0IF doit être remis à 0 dans le sous programme d'interruption.

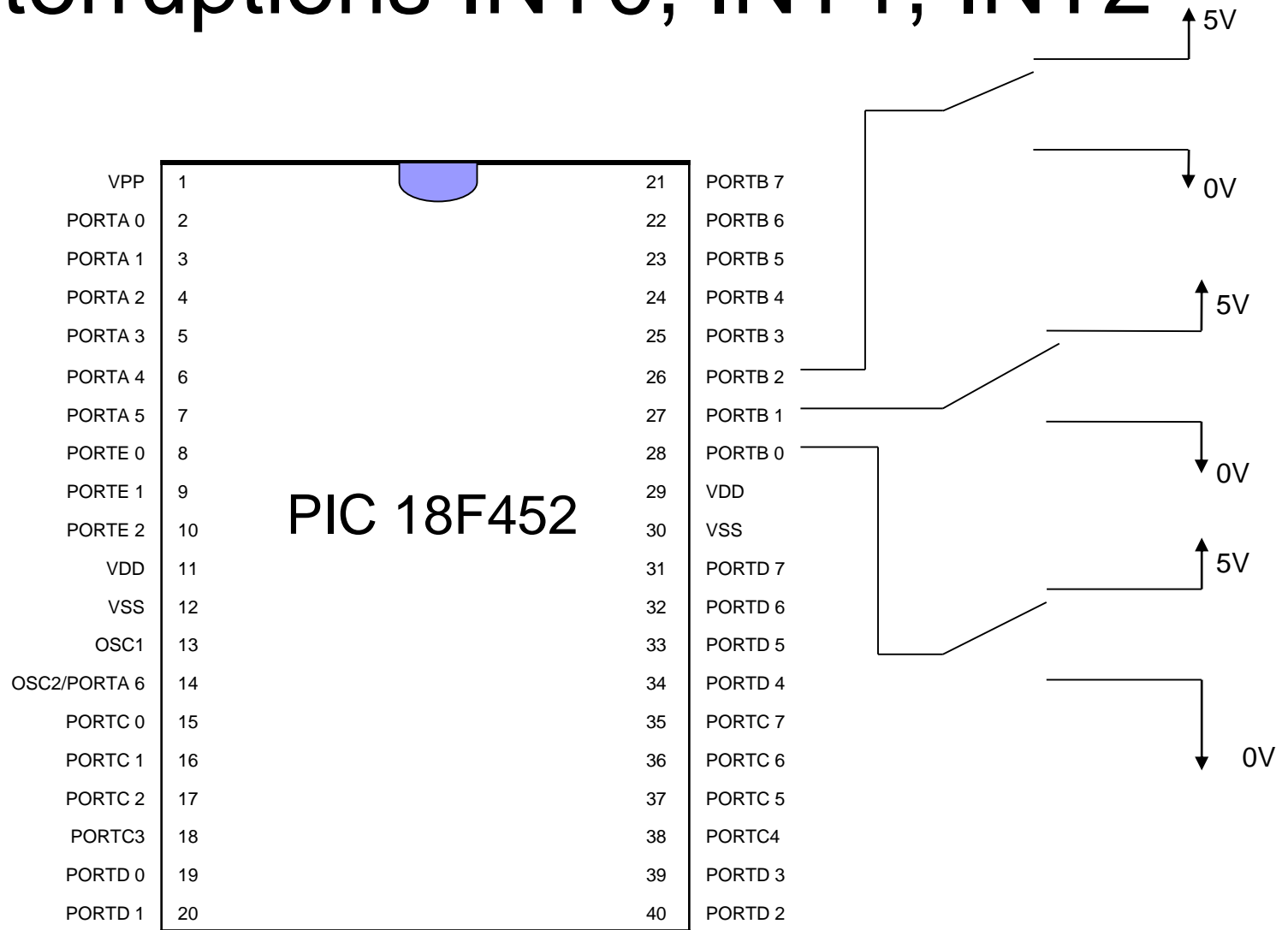
Les interruptions externes(2)

- Les interruptions INT1 et INT2
 - Même principe que INT0 :
 - INTEDG1 ou INTEDG2 bit de front;
 - INT1IE ou INT2IE bit d'autorisation
 - INT1IF ou INT2IF bit de demande d'interruption.
 - Il faut ajouter un bit de priorité
 - INT1IP ou INT2IP = 1 priorité haute (valeur par défaut)
 - INT2IP ou INT2IP = 0 priorité basse
 - Tous ces bits sont situés dans le registre INTCON3

Les interruptions externes(3)

- L'interruption RB change
 - Un changement d'état sur l'une des entrées RB4 à RB7 provoque une demande d'interruption externe par mise à 1 du bit RBIF.
 - Cette interruption doit être autorisée en mettant à 1 le bit RBIE.
 - Le bit de priorité est RBIP (par défaut à 1).

Exemple de câblage des interruptions INT0, INT1, INT2





LES TIMERS

Différents modes de fonctionnement

- Un timer est un compteur d'impulsion.
- 2 type d'impulsions:
 - Impulsion issue d'un circuit extérieur et reçue sur une patte du PIC. C'est le mode **compteur**.
 - Impulsion issue de l'horloge du PIC. Dans ce cas, nous comptons un temps. C'est le mode **timer**.
- Il existe 4 timers sur le PIC 18f452: timer0, timer1, timer2, timer3.

TIMER0

- Le TIMER0 fonctionne en timer ou compteur sur 8 ou 16 bits.
- À chaque impulsion, le registre TMR0 est incrémenté.
- Dès que ce registre est plein, il y a une demande d'interruption sur l'impulsion suivante: passage de FF_h à 00_h sur 8 bits ou de $FFFF_h$ à 0000_h sur 16 bits.

TIMER0

schéma bloc sur 8 bits

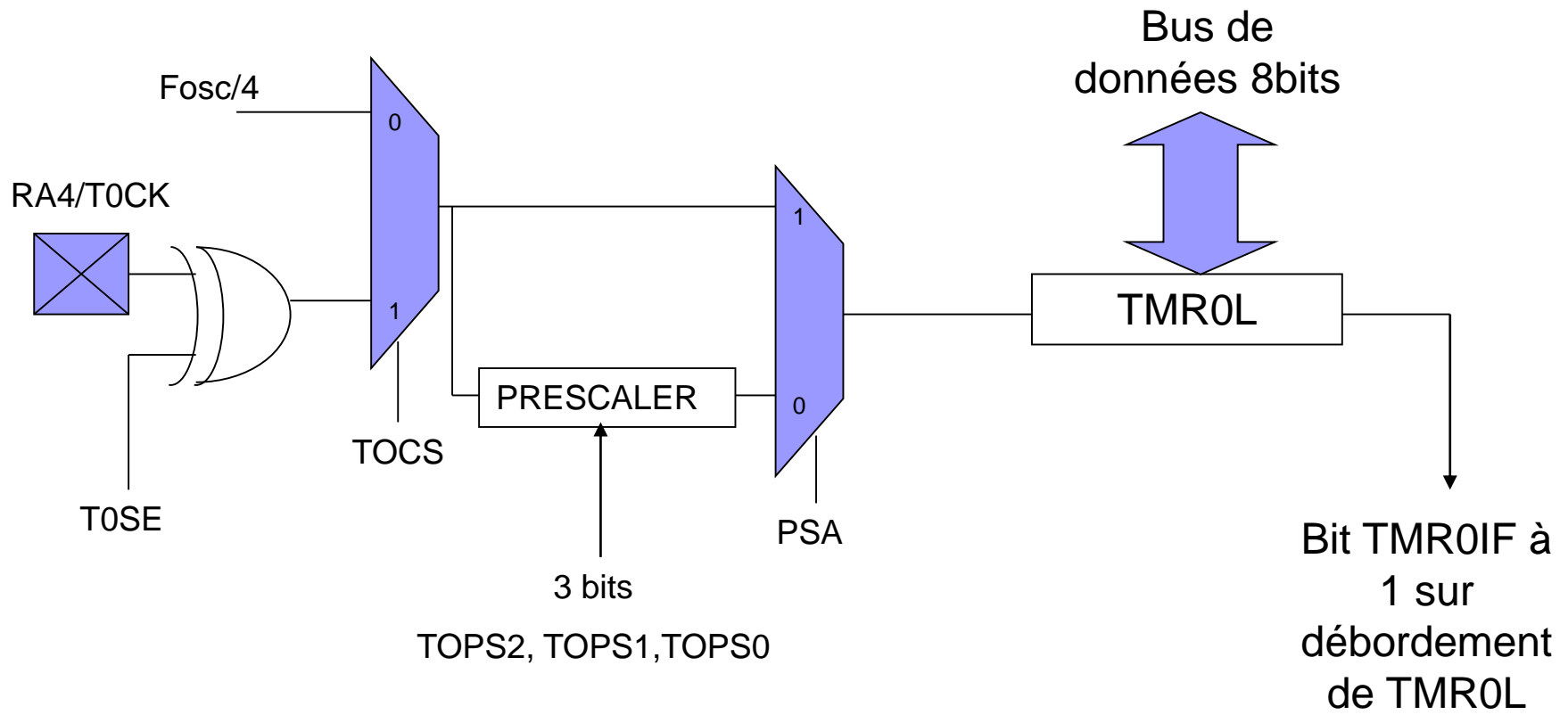
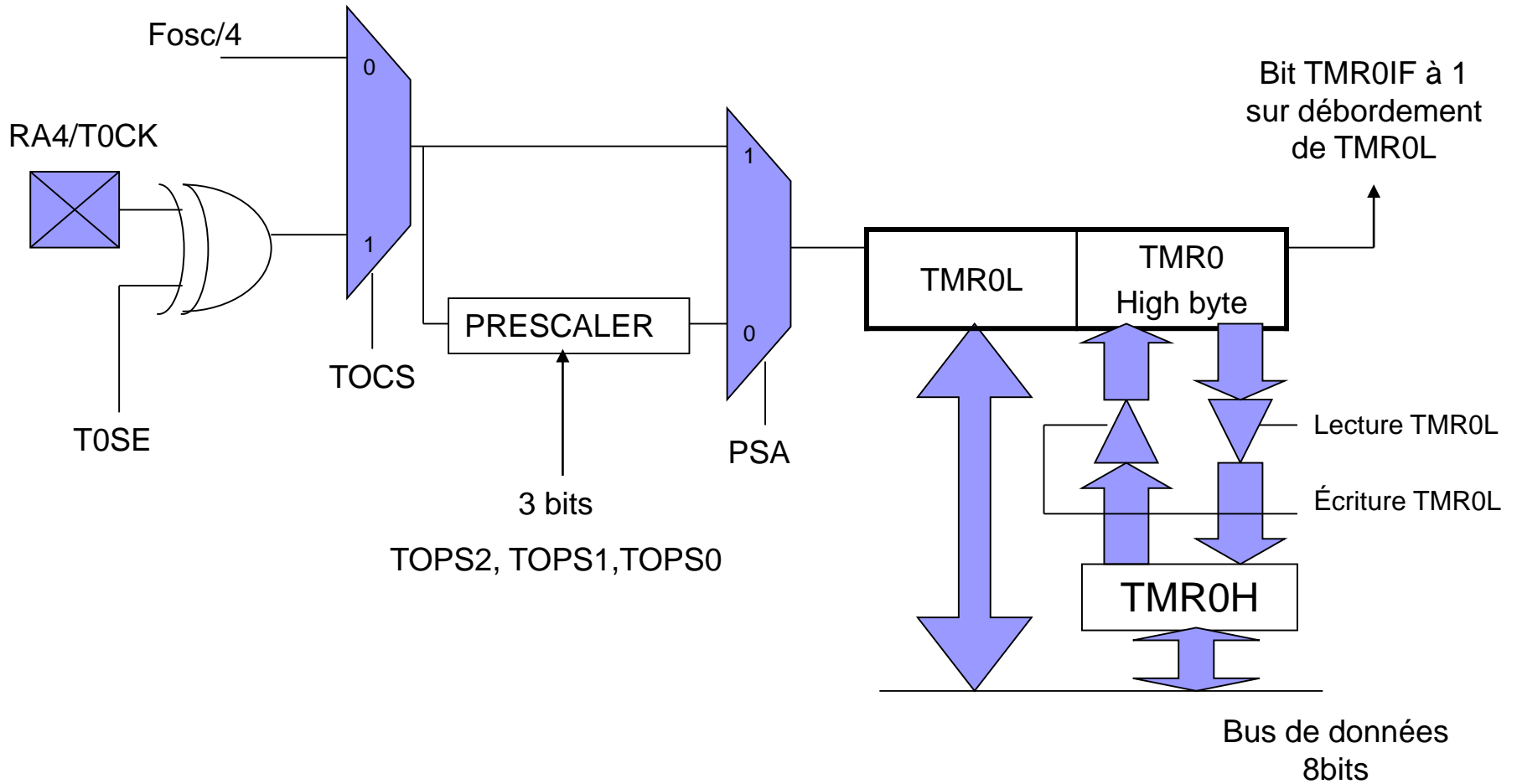


schéma bloc sur 16 bits





TMR0

Interruption

- Lors du débordement du registre TMR0, le bit TMR0IF passe à 1 ce qui génère une demande d'interruption.
- Cette interruption est prise en compte si elle est autorisée: le bit TMR0IE doit être à 1; les autorisations globales doivent aussi être au niveau 1 (GIE ou GIEL).
- En priorité basse, TMR0IP doit être à 0.

TIMER0

Le registre T0CON

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
TMR0ON	T08BIT	TOCS	T0SE	PSA	T0PS2	T0PS1	T0PS0
Bit 7							Bit 0

- bit 7 **TMR0ON:** Timer0 On/Off Control bit
 1 = Timer0 en marche
 0 = Timer0 arrêté
- bit 6 **T08BIT:** Timer0 8-bit/16-bit Control bit
 1 = Timer0 est configuré en 8-bit timer/compteur
 0 = Timer0 est configuré en 16-bit timer/compteur
- bit 5 **TOCS:** Timer0 Clock Source Select bit
 1 = Incrémentation sur T0CKI pin (compteur)
 0 = Incrémentation sur chaque cycle d'horloge CLK0 (timer)

TIMER0

Le registre T0CON (suite)

- bit 4 **T0SE:** Timer0 Source Edge Select bit
1 = Incrémentation sur front descendant de T0CKI
0 = Incrémentation sur front montant de T0CKI
- bit 3 **PSA:** Timer0 Prescaler Assignment bit
1 = Le prescaler n'est pas pris en compte.
0 = Le prescaler est pris en compte.
- bit 2-0 **T0PS2:T0PS0:** Timer0 Prescaler Select bits
111 = 1:256 prescale
110 = 1:128 prescale
101 = 1:64 prescale
100 = 1:32 prescale
011 = 1:16 prescale
010 = 1:8 prescale
001 = 1:4 prescale
000 = 1:2 prescale

TIMER0

Le Prescaler

- Le prescaler permet d'incrémenter le registre TMR0 toutes les 2 ou 4 ou 8 ou...256 impulsions.
- Il est validé par le bit PSA
- Son ratio (2,4,...,256) est déterminé par les bits TOPS2, TOPS1 et TOPS0.
- Exemple:
TOPS2, TOPS1, TOPS0 = 100 => le registre TMR0 est incrémenté toutes les 32 impulsions

TIMER0

Mode 16 bits

- Le registre TMR0H n'est pas l'octet de poids fort du registre compteur TMR0.
- Il est impossible d'écrire ou de lire directement dans l'octet de poids fort du registre TMR0.
- En lecture, TMR0H prend la valeur de TMR0 high byte au moment de la lecture de TMR0L.
- En écriture, TMR0 high byte prend la valeur de TMR0H au moment de l'écriture de TMR0L.
- Cela permet de mettre à jour le registre TMR0 sur 1 cycle.
- **L'écriture dans TMR0 s'effectue d'abord par TMR0H puis par TMR0L**