

LES INTERRUPTIONS

1/ Qu'est-ce qu'une interruption

Dans un système piloté par un microprocesseur certains évènements ne surviennent que rarement. Par exemple dans la régulation de four que nous avons déjà évoquée, il est nécessaire de tester le bouton marche/arrêt, si celui-ci est actionné il faut mettre le chauffage à 0. On pourra inclure dans la boucle de régulation le test de ce contact, s'il est mis à 0 le programme fera cesser le chauffage puis le programme tournera dans une boucle testant l'état de ce contact : est-il égal à 1 ? non alors lecture du contact et retour à l'interrogation. Un autre événement encore plus rare peut se produire, le capteur de température est détérioré, la température monte trop haut et le capteur de sécurité s'enclenche, il faut d'urgence arrêter le chauffage, il serait raisonnable d'arrêter immédiatement le chauffage sans attendre la fin d'une temporisation.

Pour répondre à ce besoin, nous disposons des interruptions. Elles peuvent être de deux ordres :

Les interruptions matérielles

Les interruptions logicielles

Les interruptions matérielles, elles sont déclenchées par un état logique appliqué sur certaines broches du microprocesseur. Par exemple le 6800 possède deux broches **NMI** et **IRQ** actives à 0. **NMI** - No Maskable Interrupt – **Interruptions non masquables**, **IRQ** Interrupt request – **Demande d'interruption**. Généralement plusieurs circuits extérieurs sont susceptibles de demander une interruption, leur sortie de demande d'interruption sont câblées en OU logique. Lorsqu'une interruption matérielle survient le microprocesseur doit détecter quel est le circuit qui vient d'effectuer la demande de façon à exécuter le programme adapté. Les circuits périphériques qui peuvent demander une interruption sont munis d'un registre d'état dans lequel un bit signal que le circuit a demandé une interruption. Ce bit d'état est mis à l'état inactif par une lecture du registre d'état par le microprocesseur.

Les interruptions logicielles : certaines instructions peuvent déclencher des interruptions. Le 6800 possède une instruction **SWI** software interrupt, le 68000 en possède un grand nombre, 16 instructions **TRAP**, une **TRAPV**, interruption déclenchée si le bit V est à 1 lorsque le microprocesseur rencontre l'instruction. La division peut déclencher une interruption causée par une tentative de division par 0 etc

2/ En quoi consiste une interruption

Lorsque le microprocesseur rencontre une demande d'interruption, suivant son origine, il va ou non se poser la question ai-je le droit ? si la demande vient de sa broche NMI ou équivalent cette demande est systématiquement acceptée, si ce n'est pas le cas il teste son masque d'interruption et entre en interruption si son masque n'est pas actif. Le masque d'interruption est variable suivant les microprocesseurs, pour le 6800 il y a un bit I dans le registre CCR qui peut être mis à un soit par programme soit parce que le microprocesseur est déjà en interruption. Pour le 68000 qui possède 3 broches de demande d'interruption, ces 3 broches ne sont pas indépendantes l'état appliqué sur ces broches constitue un nombre, pour que l'interruption soit acceptée ce nombre doit être supérieur au nombre stocké dans son registre d'état

Lorsque l'interruption est acceptée le microprocesseur effectue une copie de son registre d'état ou CCR puis va « **sauver son contexte** » dans la pile c'est à dire qu'il va sauver son état : au minimum le registre d'état (ou CCR) et l'adresse de l'instruction qu'il s'apprêtait à exécuter, éventuellement les accumulateurs, l'index, etc tout cela dépendra du microprocesseur et du type d'interruption

C'est le **pointeur de pile** qui oriente la sauvegarde, à la fin de la sauvegarde le pointeur s'est décrémenté et pointe le sommet du reste de la pile prêt à empiler s'il le faut une adresse de retour d'un sous programme ou un nouveau contexte en cas d'interruptions imbriquées.

Après la sauvegarde du contexte, le microprocesseur va charger dans son compteur de programme l'adresse du programme de traitement de l'interruption. Ce programme va d'abord détecter quel circuit a demandé l'interruption puis il va exécuter le programme adéquat la dernière instruction sera un **RTI** Return From Interrupt le microprocesseur va restituer son contexte et le pointeur de pile s'incrémenter.

Lorsque le microprocesseur place l'adresse de l'instruction sauvegardée dans son compteur de programme, le pointeur de pile a retrouvé son état initial et le programme reprend son cours normal. Si le contexte sauvegardé comprenait les accumulateurs (c'est le cas pour le 6800) ou les registres de données pour le 68000 le microprocesseur reprend son programme comme si de rien n'était. Si le programme de traitement n'a pas prévu de laisser des traces en mémoire, rien ne permet de savoir qu'une interruption a été exécutée.

3/ Les vecteurs d'interruption

Il faut entendre vecteurs au sens du moyen de transport. Les vecteurs d'interruption sont les adresses qui vont permettre au microprocesseur d'entrer dans le programme de traitement de l'interruption. Ces adresses sont logées à des adresses imposées par le concepteur du microprocesseur. Prenons l'exemple du 6800 les adresses imposées sont :

FFF8-FFF9	vecteur d'IRQ
FFFA-FFFB	vecteur de SWI
FFFC-FFFD	vecteur de NMI
FFFE-FFFF	vecteur de Reset

Dans le 68000 la notion d'interruption a été étendue à la notion d'exception. Lorsqu'une exception est acceptée, le microprocesseur se fournit un numéro interne ou si la demande est faite par un circuit périphérique de la famille 68000 celui-ci lui fournit un numéro sur le bus de données. Les adresses du 68000 sont de 24 bits mais portent sur 32 bits (2 fois 16 bits) soit 4 octets, le microprocesseur va donc multiplier ce numéro par 4, ainsi deux numéros consécutifs forment ainsi des adresses qui vont de 4 octets en 4 octets. Le nombre ainsi obtenu fournit au microprocesseur l'adresse du vecteur de l'exception en question.

Les circuits périphériques de la famille 68000 sont munis d'un registre interne dans lequel, lors de l'initialisation du système, le programme vient placer un numéro de vecteur, numéro qui sera fourni au microprocesseur lors d'une demande d'interruption.

4/ Le masquage des interruptions.

Le masque d'interruption dans le cas du 6800 se limite au bit I du registre CCR, il est mis à 1 à la mise sous tension, les IRQ sont masquées, c'est à dire que tout état actif appliqué sur le broche IRQ est ignoré. Le programme d'initialisation du système peut alors s'exécuter lorsque tout le système est en état de marche le programmeur peut alors, s'il le souhaite, mettre à 0 ce bit et alors autoriser les demandes d'interruption. Il existe une instruction CLI - Clear du bit I.

Les demandes émanant de la broche NMI sont obligatoirement acceptées par le microprocesseur, elles sont non-masquables. Dès qu'une interruption est acceptée le bit I est mis à 1 mais comme le microprocesseur a fait une copie de son registre CCR et que cette copie est sauvegardée dans la pile, lorsque le contexte est remis en place, systématiquement le bit I reprend sa valeur d'avant l'interruption.

Dans le cas du 68000, le masquage des interruption est effectué par 3 bits du registre d'état I2,I1,I0, ils définissent un niveau, une interruption n'est prise en compte que lorsqu'elle est d'un niveau supérieur au masque. Le niveau d'une interruption est défini par les états actifs qu'il va placer sur les broches I du microprocesseur, c'est le Zéro qui est actif. Ainsi un circuit qui applique sur les entrées IPL2,IPL1,IPL0 les valeurs 001 demande une interruption de niveau 6 si le micro accepte cette demande après la sauvegarde interne de son registre d'état le microprocesseur place son masque au niveau 6, il ne pourra pendant la durée du traitement de l'interruption acceptée prendre en compte que des interruptions de niveau 7

5/ pourquoi masquer une interruption ?

Nous avons déjà vu que les interruptions masquables sont masquées à la mise sous tension, en effet comme les circuits périphériques ne sont pas encore programmés, ils peuvent faire n'importe quoi pendant quelques microsecondes. Il est donc indispensable de bloquer ses demandes tant qu'ils n'ont pas été programmé.

On peut également imaginer des moments dans le programme où on ne doit pas perturber l'écoulement normal des instructions ou également si une interruption d'une urgence supérieure a été addeptée