

## RÉSUMÉ EXCEPTIONS

Une exception est une **exception au déroulement normal d'un programme**.

### INITIALISATION

Par exemple, l'**initialisation** du CPU (requis par la ligne ReSeT du bus), est une exception qui a pour effet d'arrêter le programme en cours et de lancer (sans retour) un programme d'initialisation depuis une adresse de démarrage fixe (FFFAh ici). Certaines machines disposent d'une instruction qui l'initialise (ici **RST**).

### INTERRUPTION

Un périphérique peut matériellement requérir l'**interruption** du programme principal en cours pour lancer une sorte de sous-programme spécifique à son service (dit **programme de service d'interruption ou d'exception**).

Cette exception déclenchée par un périphérique s'appelle **interruption matérielle**.

A la fin de ce programme de service d'exception, le programme principal reprend là où il a été interrompu (comme c'est le cas pour un sous-programme) dans le même état SR que si l'interruption n'avait pas eu lieu..

Une différence essentielle entre un appel à un **programme de service d'exception** et un sous-programme normal est donc que non seulement le **PC** mais aussi le **registre d'état SR** sont sauves sur la **pile**, afin de pouvoir restituer son état au retour.

### DÉCODAGE DES REQUÊTES D'INTERRUPTION

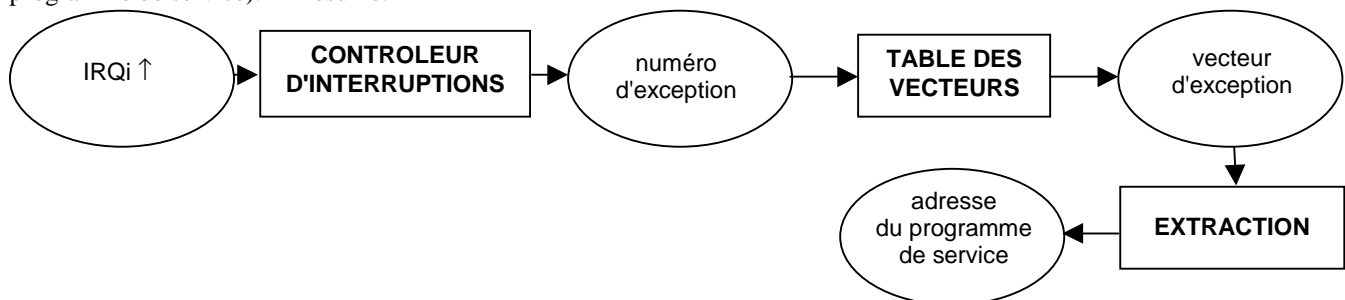
Dans un système à **interruptions décodées** (comme ici), il y a une **ligne de requête d'interruption** IRQ<sub>i</sub> n°i (Interrupt ReQuest) par périphérique. Ces lignes font normalement partie du bus système (15 dans le l'ex bus ISA du PC, 32 ici).

Lorsqu'un signal (ici un front ↑) est envoyé par un périphérique sur sa ligne de requête d'interruption IRQ<sub>i</sub> au **contrôleur d'interruption**, ce dernier demande au CPU de déclencher l'interruption de **numéro d'exception** INT=f(i) (ici INT= i + 32).

### VECTEURS D'EXCEPTION

Pour une **exception vectorisée** comme une interruption, l'**adresse du programme de service d'exception** n°INT se trouve dans le **vecteur d'exception** n°INT de la **table des vecteurs d'exception**, placée en mémoire centrale à une adresse connue (ici 0000h).

La composition du **vecteur d'exception** complet dépend des machines. (Ici il a deux mots; le premier est l'adresse de programme de service). En résumé:



### MASQUAGE DES EXCEPTIONS

Un drapeau du registre d'état SR (bit IF "Interrupt Flag" dans le PC et ici) permet d'inhiber (on dit **masquer**) les exceptions (ici lorsqu'il est à 0).

Des instructions (ici **ENI** et **DSI**) permettent de mettre IF à 1 ou 0 et donc de respectivement valider ou inhiber les exceptions matérielles comme les interruptions.

Les exceptions sont automatiquement inhibées à l'initialisation (IF à 0) afin de garantir qu'il n'y aura pas de requête d'exception lancée alors que les vecteurs n'ont pas encore été remplis ...

Les exceptions sont automatiquement inhibées (IF se met à 0) au lancement d'un programme de service afin d'éviter qu'il ne soit lui-même intempestivement interrompu par une requête plus tardive ou moins prioritaire.

## ATTENTE D'INTERRUPTION

L'instruction d'attente (ici HLT) permet de mettre le CPU en état d'attente d'interruption: il s'arrête alors sur **HLT** en consommant moins de puissance électrique et donc en chauffant moins. Il répond aussi plus vite à la requête.

Il ne repart qu'à la prochaine interruption pour exécuter son programme de service, puis en revenir pour exécuter l'instruction qui suit HLT et continuer.

(Ici, l'état d'attente est indiqué par l'indicateur WF "Wait Flag" du registre d'état SR).

## TRAPPE MATÉRIELLE

Une **exception déclenchée par le CPU** est appelée **trappe**.

**Certaines trappes sont matérielles** et inopinées, déclenchées par un problème dans le CPU comme:

- tentative de division entière par zéro,
- tentative d'extraction de racine carrée d'un nombre négatif,
- tentative d'exécution de code d'instruction inexistante,
- débordement de la pile etc..

Chaque trappe matérielle a un n° d'exception INT fixe prévu par la documentation de la machine.

## TRAPPE LOGICIELLE

On peut appeler un programme de service d'exception de manière déterministe comme un sous-programme en prenant certaines précautions; ceci s'appelle une **trappe logicielle** ou interruption logicielle. L'instruction **TRP** n permet de lancer le programme de service d'exception n° n.

Ceci permet d'implémenter les **appels système** (comme par exemple sortir une chaîne de caractère). Chaque appel système aura toujours le même numéro (défini par la documentation du système) même si au cours des diverses versions du système l'adresse de son programme change.

### Instructions

COMMENTAIRE	INSTRUCTION	ACTIONS RÉSUMÉES
initialisation logicielle	<b>RST</b>	FFFAh → PC
Appel au sous-programme ssp	<b>JSR ssp</b>	adresse de retour → pile ; adresse du sous-prog ssp → PC;
Retour de sous-programme	<b>RTS</b>	pile → PC
lancement du programme de service d'exception n° n	<b>TRP n</b>	SR → pile; PC → pile; adresse du prog. de service du vecteur n°n → PC;
Retour de programme de service d'exception	<b>RTI</b>	pile → PC; pile → SR;
validation des exceptions	<b>ENI</b>	1 → IF
inhibition des exceptions	<b>DSI</b>	0 → IF
mise en attente d'interruption	<b>HLT</b>	1 → WF

### Exceptions matérielles

COMMENTAIRE	SIGNAL	ACTIONS RÉSUMÉES
initialisation matérielle	<b>ReSeT=1</b>	FFFAh → PC;
requête d'interruption matérielle par ligne IRQi	<b>IRQi ↑</b>	SR → pile; PC → pile; adresse du prog. de service du vecteur n°n = i+32 → PC;
trappe matérielle n°n	<b>problème CPU n°n</b>	SR → pile; PC → pile; adresse du prog. de service du vecteur n°n → PC;