

ESIAL

Module PFSI

TD n° 11 - Exceptions

Objectifs:

Comprendre et savoir utiliser les exceptions.

1. Principes.

Décrire brièvement ce qu'est une interruption.

2. Écriture du programme principal.

Écrivez un programme principal ("main program") qui boucle indéfiniment en ajoutant 10h au contenu cntm de la case mémoire d'adresse CNTMA=FF00h.

Ce programme principal **initialise** cntm à **0** et le **remet à zéro** après qu'il ait atteint **6Fh**.

Le programme principal a pour adresse MAIN_PRGA.

3. Écriture du programme d'interruption

A/ **Écrire un programme d'interruption** qui **ajoute 2** au contenu **cnti** de la case mémoire d'adresse CNTIA=FF02h chaque fois qu'il est appelé. Ce programme débute à l'adresse IT_PRGA.

B/ Pourquoi faut-il **sauver les registres utilisés** ?

C/ Pourquoi faut-il **autoriser à nouveau les interruptions** explicitement ?

D/ **Pourquoi** cette machine (comme le PC en mode protégé) **efface** t-elle **IF** lorsqu'elle lance le programme d'interruption ?

D/ Pourquoi le programme d'interruption doit-il **se terminer par RTI** et non par RTS ?

4. Initialisations

A/ Dans cette machine, quel est le numéro INT d'interruption qui correspond à la ligne IRQ2 ?

B/ Complétez le programme principal pour :

- * initialiser la pile à l'adresse STACKA=1000h ;
- * initialiser cntm à 0 ;
- * initialiser la **table des vecteurs d'interruption** pour que IRQ2 lance le programme d'interruption ;
- * **valider les interruptions** (cf. instruction ENI) ;

5. Sauvetage et restitution de l'état, instruction RTI

A/ Entre quelles instructions du programme principal le lancement du programme d'interruption risquerait-il de poser problème si le registre d'état SR n'était pas sauvé ? Pourquoi ?

B/ Pourquoi le retour du programme d'interruption doit se faire avec une instruction **RTI**, et non pas avec une séquence d'instructions (par exemple RTS,...) ?

6. Section critique, instruction DSI

A/ On considère que cette fois programme d'interruption et programme principal agissent sur la même case mémoire (donc CNTMA = CNTIA = FF00h). Vont-ils interagir comme prévu en incrémentant son contenu ?

B / Comment résoudre ce problème ? (cf. instruction **DSI**).

7. Instruction HLT

A/ Que se passe t-il si l'on ajoute l'instruction HLT dans la boucle du programme principal ?

B/ Une machine dans l'état "attente" consomme t-elle généralement plus d'énergie ?

8. Priorité et mémorisation

A/ **Ecrivez** un autre **programme d'interruption qui soustrait 1** de la case mémoire d'adresse CNTMA lorsque IRQ3 est stimulé.

B/ Que se passe t-il lorsque **IRQ2** et **IRQ3** sont stimulés **simultanément** ?

9. Trappe logicielle - Instruction TRP

A/ Appelez le programme d'interruption tel quel en utilisant l'instruction **TRP**.

B/ Pourquoi ne peut-on pas utiliser simplement l'instruction JSR ?

C/ Quel intérêt peut-il y avoir à réaliser des programmes de trappe via un vecteur plutôt que des sous-programmes, bien qu'ils ne sont pas destinés à être déclenchés inopinément par un périphérique mais de manière déterministe par un programme ?

10. Trappe (matérielle)

A/ Que se passe t-il d'après vous lorsqu'un programme effectue une action interdite (e.g. division par zéro) ?

B/ Trouvez d'autres actions interdites susceptibles de déclencher une trappe.

11. Initialisation de l'unité centrale de traitement (CPU)

A/ L'initialisation est-elle une exception ?

B/ Comment est-elle déclenchée ?

C/ Que se passe t-il alors ?

D/ Est-elle vectorisée dans cette machine ?

E/ Pourquoi met-on une instruction de saut à l'adresse FFFA ?

F/ Pourquoi n'utilise t-on pas FFFE ?

G/ Pourquoi utilise t-on FFFA et pas une adresse plus faible, qui permettrait de mettre un programme avant la fin de l'espace d'adressage ?