

1. Qu'est-ce qu'une interruption? Quand cela se produit-il?

Solution: Une interruption est un évènement généralement imprévu qui interrompt la séquence normale d'exécution des instructions par le microprocesseur.

Les interruptions se produisent quand un périphérique signale un évènement, quand une faute matérielle ou logicielle survient ou quand le programmeur appelle une interruption.

2. D'un point de vue logiciel, que fait généralement un microprocesseur lorsqu'il détecte un signal d'interruption provenant d'un périphérique?

Solution:

1. Le microprocesseur termine l'instruction (ou les instructions) en cours
2. Le microprocesseur vérifie si l'interruption peut être traitée en fonction de sa priorité et si elle est permise.
3. Le microprocesseur sauvegarde l'adresse de retour et toute l'information nécessaire à la reprise normale de l'exécution après le traitement de l'interruption.
4. Le microprocesseur détermine l'adresse de la routine à exécuter afin de traiter l'interruption.
5. Le microprocesseur fait un saut vers la routine qui traite l'interruption

3. Pourquoi un ordinateur se sert-il de la pile lorsqu'une interruption se produit?

Solution: Pour sauvegarder l'adresse de retour de l'interruption et pour sauvegarder les autres informations nécessaires à la poursuite des opérations, après le traitement de l'interruption.

4. Qu'est-ce qu'une interruption masquée?

Solution: Il s'agit d'une interruption dont le traitement est repoussé ultérieurement parce qu'elle est désactivée.

5. À quoi sert la table des vecteurs d'interruption? Cette table est-elle en mémoire vive, dans le microprocesseur, dans une mémoire non-volatile ou sur le disque dur?

Solution: La table des vecteurs d'interruption associe les interruptions à la routine qu'il faudra exécuter afin de les traiter. Située dans la FLASH initialement (peut être déplacée en mémoire RAM), elle a pour index le numéro de l'interruption et pour contenu l'adresse absolue de la routine qui traitera l'interruption.

6. À quoi sert le contrôleur d'interruption programmable de votre ordinateur?

Solution: Le contrôleur d'interruption programmable est un circuit logique qui gère les interruptions : il reçoit et traite les signaux d'interruptions provenant des périphériques d'un côté et, de l'autre côté, échange les signaux adéquats avec le microprocesseur pour gérer les interruptions.

7. Quelles sont les instructions communes à toutes les routines traitant les interruptions (ISR, Interrupt Service Routine)?

Solution: Généralement, la routine d'interruption commence par sauvegarder sur la pile les registres qui seront utilisés dans l'interruption. Ensuite la routine d'interruption contiendra habituellement des instructions gérant l'interruption elle-même (exemple : remettre à 0 le drapeau signalant l'interruption pour que l'interruption puisse se produire de nouveau). La routine d'interruption contiendra aussi très souvent des instructions qui lui demanderont d'accéder au périphérique ayant causé l'interruption lorsqu'il s'agit d'une interruption matérielle. Puis, la routine d'interruption rechargera les registres sauvegardés sur la pile au début de l'interruption. Enfin, une instruction de retour à la fin de la routine d'interruption dépilerà l'adresse de retour et les autres informations nécessaires à la reprise de l'exécution après la routine d'interruption.

8. Qu'est-ce qu'une exception? Pourquoi les exceptions ont-elles une priorité plus grande que les interruptions matérielles et les interruptions logicielles?

Solution: Une exception est un événement anormal ou imprévu survenant lors de l'exécution du code. Habituellement, une exception rend impossible la lecture ou l'exécution de l'instruction en cours (exemple : division par 0). Les exceptions ont des priorités plus grande que les interruptions parce qu'il faut les traiter si elles se produisent lors de l'exécution d'une interruption.

9. Pourquoi utilise-t-on des interruptions logicielles plutôt que des appels de fonction pour appeler des routines du système d'exploitation afin d'accéder aux périphériques?

Solution: Pour permettre au système d'exploitation de changer facilement l'adresse des routines permettant d'accéder au matériel, susceptible de changer d'un ordinateur à l'autre. Pour fournir une interface conviviale entre les fonctions du système d'exploitation qui peuvent changer de place et les applications des programmeurs.

10. Quel est l'effet d'une interruption sur le registre PC?

Solution: PC prend une nouvelle valeur fournie à partir du numéro de l'interruption et de la table des vecteurs d'interruptions.

11. Pour le coeur ARM, où est sauvegardée l'adresse de retour qui sera utilisée lorsque la routine traitant l'interruption sera terminée?

Solution: Sur la pile.

12. À quelles adresses de la mémoire retrouverons-nous l'adresse de la routine d'interruption 9h (assumez un coeur ARM).

Solution: L'adresse sera $9 \times 4 = 36$.

13. Décrivez ce qui se passe, au niveau des lignes dédiées à la gestion des interruptions, lorsqu'un usager appuie sur une touche du clavier?

Solution: Le contrôleur de clavier active un signal d'interruption qui sera reçu par le contrôleur d'interruption. Ensuite, en fonction des autres interruptions et en fonction de ses paramètres d'opération, le contrôleur d'interruption générera un signal d'interruption au microprocesseur. Ce signal sera vu par le microprocesseur qui entamera le traitement de l'interruption après échange d'information et de signaux avec le contrôleur d'interruption.

14. Y a-t-il une différence entre l'instruction permettant de revenir d'une interruption et celle permettant de revenir d'une fonction? Y a-t-il une différence de comportement du microprocesseur dans ces deux cas?

Solution: Il n'y a pas de différence entre les deux instructions pour le processeur ARM : dans les deux cas, BX LR est utilisée. Cependant, LR contient l'adresse de retour lorsqu'on revient d'une fonction et LR contient une valeur égale à 0xFFFFFFFF lorsqu'on veut revenir d'une interruption.

Quand LR n'est pas 0xFFFFFFFF (retour de fonction), PC prends la valeur de LR et le retour est fait.

Quand LR vaut 0xFFFFFFFF (retour d'interruption), le microprocesseur dépile l'adresse de retour ($PC = Mem[SP, \#X]$) et plusieurs autres informations (drapeaux, LR et certains registres).