

Les entrées/sorties

Périphériques et contrôleurs

- Les périphériques sont connectés aux contrôleurs d'entrées/sorties
- Les contrôleurs communiquent avec le processeur via les bus

Espace mémoire adressable

- Un processeur peut lire ou écrire dans n'importe où dans l'espace mémoire adressable grâce aux instructions ldr et str.
- La plus grande partie de cet espace correspond à des barrettes de RAM.
- Certaines zones correspondent à des contrôleurs d'entrées/sorties

Exemple d'espace mémoire adressable

- En hexadécimal, la zone 0000 0000-3F00 0000 correspond à une barrette de RAM.
- La zone 4000 0000-4000 000F correspond à un contrôleur C1
- La zone 4000 1000-4000 200F correspond à un contrôleur C2

Structure du controleur C1

- On imagine que le controleur C1 est constitué de 4 registres de 32 bits nommés STATE, OP, PARAM0 et PARAM1.
- L'adresse 4000 0000 correspond au registre STATE du controleur C1.
- L'adresse 4000 0004 correspond au registre OP du controleur C1.
- L'adresse 4000 0008 correspond au registre PARAM0 du controleur.
- L'adresse 4000 000C correspond au registre PARAM1 du controleur.

Commander un controleur d'entrées/sorties

- Il suffit de regarder la documentation du controleur pour savoir comment on le commande !
- Exemple : imaginons que notre controleur est capable d'allumer et d'éteindre 2 leds L0 et L1.
- Registre STATE : état du controleur (LECTURE SEULE)
0==> READY 1 ==> RUNNING 2 ==> ERROR

Connaître l'état du controleur C1

- Comment le processeur peu connaitre l'état du controleur C1 ?

`ldr r0,4000 0000`

r0 contient l'état du controleur

- Si on essaye d'écrire dans l'adresse 4000 0000
il y aura une erreur sur le bus car il est interdit de
modifier ce registre

Envoyer un ordre à C1

- Le registre OP permet d'envoyer un ordre au controleur
 - 1 ==> LIGHT_ON on allume une led
 - 2 ==> LIGHT_OFF on etteind une led
 - 3 ==> SET_COLOR on fixe la couleur d'une led
 - 4 ==> GET_COLOR on récupère la couleur d'une LED
- Pour les commandes 1 et 2 le numéro de led sera dans PARAM0
 - 0==> led L0 1==> led L1
- Pour la commande 3 le numéro de la led est dans PARAM0 et la couleur est écrite dans PARAM1
 - 1==> RED 2==> BLUE 3==> GREEN
- Pour la commande 4 le numéro de la led se trouve dans PARAM0 et la couleur est récupérée dans PARAM1

Exercice

- Ecrire un programme assembleur qui allume la led 1 en bleu

Solution

Allumer en bleu la led L1

```
mov r0,#1
```

```
str r0,4000 0008
```

```
mov r0,#2
```

```
str r0,4000 000C
```

```
mov r0,#3
```

```
str r0,4000 0004
```

```
mov r0,#1
```

```
str r0,4000 0004
```

Difficultés

- Il aurait fallu vérifier que l'état du contrôleur était à READY !
- Imaginons que nos leds mettent une milliseconde pour changer de couleur !
==> L'état du contrôleur va passer de READY à RUNNING durant le changement de couleur de la led.
==> l'état du contrôleur passe de RUNNING à READY lorsque le changement de couleur est effectif.

Que faire ?

- 1) Attendre que la case mémoire 4000 0000 passe à 0
- 2) utiliser la commande SET_COLOR
- 3) Attendre que la case mémoire 4000 0000 passe à 0
- 4) utiliser la commande LIGHT_ON

Attente active

- On veut éviter de lire sans cesse la case mémoire 4000 0000 durant des milliers de cycle !
- Pour éviter l'attente active, on demande que le contrôleur envoie une interruption à chaque changement d'état.
- L'utilisation d'interruption permet d'éviter l'attente active.

Controlleur C2

- On imagine que le controlleur C1 est constitué de 4 registres de 32 bits nommés STATE, OPERATION, PARAM0, PARAM1 et d'un buffer de 4KO nommé BUFFER.
- L'adresse 4000 1000 correspond au registre STATE du controlleur C2.
- L'adresse 4000 1004 correspond au registre OPERATION du controlleur C1.
- L'adresse 4000 1008 correspond au registre PARAM0 du controlleur.
- L'adresse 4000 100C correspond au registre PARAM1 du controlleur.
- Les adresses de 4000 1010 à 4000 200F correspondent au buffer.

Contrôleur C2

- Imaginons que ce controleur commande un disque dur.
STATE : 3 états READY, RUNNING et ERROR
- OPERATION : READ permet de lire un secteur du disque dont le numéro se trouve dans PARAM0
- Lorsque l'état du controleur passe à READY le secteur se trouve dans le buffer

Principe de la lecture d'un secteur du disque

- Attendre que le controleur soit dans l'état READY
- Utiliser le commande READ
- Attendre que le controleur soit dans l'état READY
- Transférer le contenu du buffer vers la RAM

Difficulté

- Le processeur n'est pas efficace pour transférer 4KO de données d'un buffer situé dans le contrôleur vers la RAM !!
- On va utiliser un contrôleur DMA

Controlleur DMA

- Direct Memory Access
- On lui envoie des ordres pour transférer massivement des données d'un contrôleur vers
 - la RAM ou inversement
- Chaque ordre contient 4 paramètres
 - lecture ou écriture de la RAM
 - adresse source
 - adresse destination
 - taille des données à transférer
- A la fin du transfert le contrôleur envoie une interruption au processeur.
- Un contrôleur DMA peut parfois faire 4 transferts simultanés.

Controleur de bus

- Plusieurs entités peuvent être amenées à envoyer des données sur le bus : le processeur et le controleur DMA...
- Il peut y avoir des conflit d'accès.
- Le controleur de bus permet de vérifier qu'une seule entité envoie des données sur le bus à un moment donné.

Conclusion

- Le processeur commande les controleurs d'entrées sorties
- Les interruptions sont indispensables pour éviter l'attente active.
- Il se fait aider par le controleur DMA et le controleur de bus.
- Une horreur à écrire !