GIF-1001 Ordinateurs: Structure et Applications
Solutions: Bus et addressage

---

1. Qui contrôle le bus d'adresse? Le bus de données? Le bus de contrôle?

   Solution: Bus d'adresse: CPU ou contrôleur de DMA avec l'aval du CPU.

   Bus de données: La personne désignée par le bus d'adresse et le bus de contrôle.

   Bus de contrôle: CPU ou contrôleur de DMA avec l'aval du CPU

2. Quand le microprocesseur de votre ordinateur ira-t-il lire une instruction de la mémoire? Quand le microprocesseur de votre ordinateur ira-t-il lire ou écrire une donnée de la mémoire?

   Solution: Le microprocesseur lira des instructions de la mémoire dès le démarrage de l'ordinateur. Il lira ensuite les instructions une après l'autre à moins de rencontrer une instruction qui lui dise de faire autrement.

   Le microprocesseur lira ou écrira une donnée de la mémoire s'il rencontre une instruction lui demandant de le faire.

3. Si un bus d'adresse a 5 lignes, un bus de donnée a 8 lignes et un bus de contrôle a deux lignes (lecture et écriture), combien de bits peuvent être adressés et lus/écris par ces bus?

   Solution: 2^5adresses $\times$ 8bits/adresse $= 2^8 = 256$

4. Quelle est la séquence d'opération effectuée par le CPU pour
   (a) lire une donnée en mémoire?

   Solution: 1) Changer les lignes d'adresse, 2) Activer le signal de lecture de la mémoire, 3) Lire les données sur le bus de données

   (b) lire une instruction en mémoire?

   Solution: 1) Changer les lignes d'adresse, 2) Activer le signal de lecture de la mémoire, 3) Lire l'instruction sur le bus de données

   (c) écrire une donnée en mémoire?

   Solution: 1) Changer les lignes d'adresse, 2) Mettre des données sur le bus de données, 3) Activer le signal d'écriture de la mémoire

5. Qu'est-ce qu'un ALU et quel est sont rôle?

> **Solution:** ALU = "Arithmetical and Logical Unit". L'ALU est la partie du CPU qui exécute les opérations mathématiques et logiques.

6. Qu'est-ce qu'un registre?

> **Solution:** Un emplacement mémoire du CPU à accès très rapide et étroitement reliée au fonctionnement de ce dernier.

7. Pourquoi un port d'entrée utilise-t-il des buffers "tri-state"?

> **Solution:** Il faut que les données passent du bus de données à une mémoire (ou une entrée/sortie avec un port) ou vice versa seulement lorsqu'on le désire (lorsque les lignes d'adresse ont une certaine valeur!). Sinon, il y a des collisions sur le bus de données ou la valeur de la mémoire change dès que le bus de donnée change.

8. Soit une instruction de langage machine qui permet de transférer (écrire) la valeur $E7_h$ à l'adresse mémoire 148_h. Au moment où cette instruction est exécutée, indiquez la valeur binaire présente sur le bus d'adresse, sur le bus de données et indiquez le signal de contrôle actionné sur le bus de contrôle.

> **Solution:** Adresse : 148_h. Donnée : $E7_h$. Contrôle : Écriture.

9. À quoi sert le décodeur d'adresse sur le bus d'un système ordinateur?

> **Solution:** Le décodeur d'adresse sert à déterminer la mémoire ou le périphérique qui sera activé en fonction de l'adresse choisie.

10. Supposons le système illustré en figure 1.

    Dans ce système, un bus d'adresse de 5 lignes permet d'accéder à 32 adresses différentes. Les adresses 8 à 15 accèdent à la mémoire 1, car A3 active cette dernière. Les adresses 16 à 23 accèdent à la mémoire 2, car A4 active cette dernière. Les mémoires, le CPU et le bus de donnée ont 8 bits. Par ailleurs, il existe deux lignes formant le bus de contrôle : "Read memory" et "Write memory". Ces lignes permettent de lire ou écrire en mémoire.

    Le CPU possède au moins les registres R1, R2, IR, PC et MAR. Les registres R1 et R2 sont équivalents à des variables 8 bits utilisées pour différentes opérations arithmétiques, logiques ou autres. Le registre IR contient l'instruction en cours d'exécution. Le registre PC détermine quelle sera la prochaine instruction exécutée par le CPU. Il est incrémenté automatiquement à la fin d'une instruction, sauf pour l'instruction JMP. Finalement, le registre MAR fixe la valeur

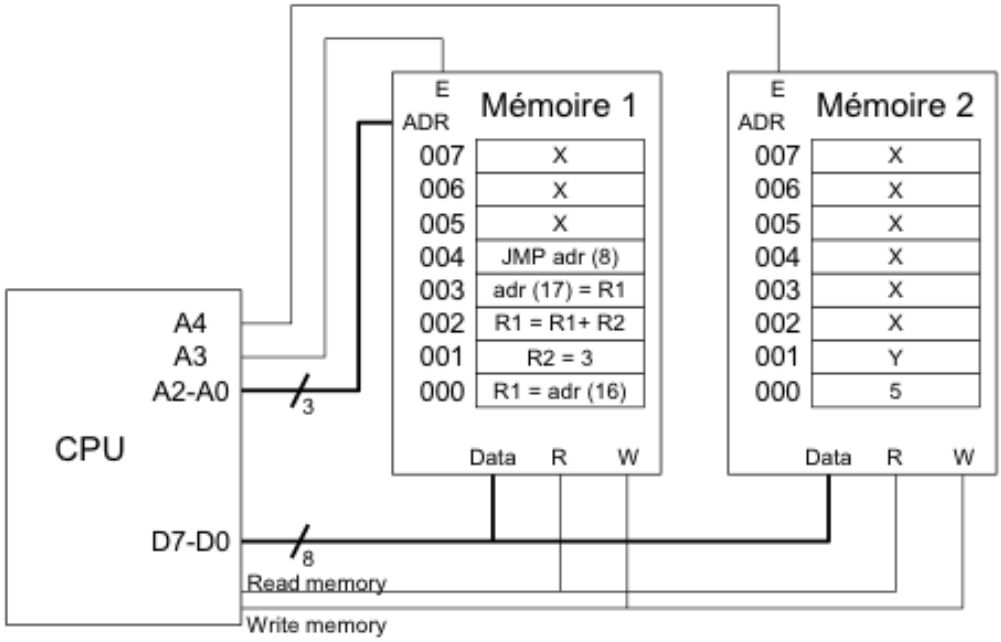

Figure 1: Système pour la question 10

des lignes d'adresse du CPU. Si ce registre contient 5 par exemple, A0 et A2 seront à "1" et A1, A3 et A4 seront à "0".

Le CPU est capable d'exécuter plusieurs instructions : il peut mettre le contenu d'une case mémoire dans un registre et vice versa, il additionner des registres, il peut mettre des constantes dans un registre, il peut effectuer des sauts inconditionnels (l'instruction JMP change la valeur du PC, i.e. JMP 12 équivaut à PC = 12!), etc.

Sachant que la valeur initiale de PC est 8 et que le CPU exécute une instruction par coup d'horloge, décrivez ce que fera cet ordinateur dans les prochains coups d'horloge. Dites quelles seront les valeurs des bus (contrôle, données, adresse), des registres et des cases mémoire.

Pour vous aider, voici les premières étapes:

Fetch instruction 1 :

- PC vaut 8
- MAR est mis à 8
- La ligne Read Memory est activée (pour aller chercher une instruction!)
- L'instruction `R1 = adr(16)` apparaît sur le bus de données et elle est mise dans IR.

Le CPU exécute l'instruction 1 :

- Le MAR est mis à 16
- La ligne Read Memory est activée (pour aller chercher une donnée!)
- 5 apparaît sur le bus de données
- 5 est mis dans R1
- PC est incrémenté, il vaut 9 maintenant.

Solution: Fetch instruction 2 :

- PC vaut 9
- Le MAR est mis à 9
- La ligne Read Memory est activée (pour aller chercher une instruction!)
- L'instruction `R2 = 3` apparaît sur le bus de données et elle est mise dans IR.

Le CPU exécute l'instruction 1 :

- Le registre R2 est mis à jour.
- PC est incrémenté, il vaut 10 maintenant.

Fetch instruction 3 :

- PC vaut 10.
- Le MAR est mis à 10.
- La ligne Read Memory est activée (pour aller chercher une instruction!).
- L'instruction `R1 = R1 + R2` apparaît sur le bus de données et elle est mise dans IR.

Le CPU exécute l'instruction 3 :

- Le registre R1 est mis à jour ($R1 = 5 + 3 = 8$).
- PC est incrémenté, il vaut 11 maintenant.

Fetch instruction 4 :

- PC vaut 11.
- Le MAR est mis à 11.
- La ligne Read Memory est activée (pour aller chercher une instruction!).
- L'instruction `adr(17) = R1` apparaît sur le bus de données et elle est mise dans IR.

Le CPU exécute l'instruction 4 :

- Le MAR est à 17.
- R1 est mis sur le bus de données.
- La ligne Write Memory est activée.
- L'adresse 001 dans la Mémoire 2 (le Y) vaut maintenant 8.

- PC est incrémenté, il vaut 12 maintenant.

Fetch instruction 5 :

- PC vaut 11.
- Le MAR est mis à 11.
- La ligne Read Memory est activée (pour aller chercher une instruction!).
- L'instruction `adr(17) = R1` apparaît sur le bus de données et elle est mise dans IR.

Le CPU exécute l'instruction 5:

- Le MAR est à 17.
- R1 est mis sur le bus de données.
- La ligne Write Memory est activée.
- L'adresse 001 dans la Mémoire 2 (le Y) vaut maintenant 8.
- PC est incrémenté, il vaut 12 maintenant.

Fetch instruction 6 :

- PC vaut 12.
- Le MAR est mis à 12.
- La ligne Read Memory est activée (pour aller chercher une instruction!).
- L'instruction `JMP adr(8)` apparaît sur le bus de données et elle est mise dans IR.

Le CPU exécute l'instruction 6 :

- PC vaut 8 maintenant.
- Puisque PC vaut 8, on retourne à Fetch instruction 1 : et on boucle jusqu'à la fin des temps, jusqu'à un bris matériel, ou plus probablement, jusqu'à ce que l'alimentation soit éteinte!