

ESIAL

Module PFSI

TD N° 4 - CHEMINS DE DONNÉES - INTERFAÇAGE I/O SUR BUS

Objectifs:

- Savoir lire une carte d'adressage;
- Comprendre les principes de l'interfaçage d'un bus.

1. Espace d'adressage

On donne ci-après (cf. tableau 1) la carte ("map") de l'espace d'adressage d'une machine à mots de 16 bits pour les adresses et les données.

On suppose que la machine est "big endian" (comme la plupart des RISC et donc le Power PC, c'est à dire que l'adresse d'un octet décroît avec son rang dans le mot, contrairement à une machine "small endian" comme le Pentium des PCs).

Calculer l'adresse en hexadécimal des données suivantes puis les reporter sur la carte:

- a/ dernier mot de la mémoire vive;
- b/ premier mot de la mémoire morte;
- c/ mot de sortie du port parallèle.
- d/ octet d'entrée du port parallèle;

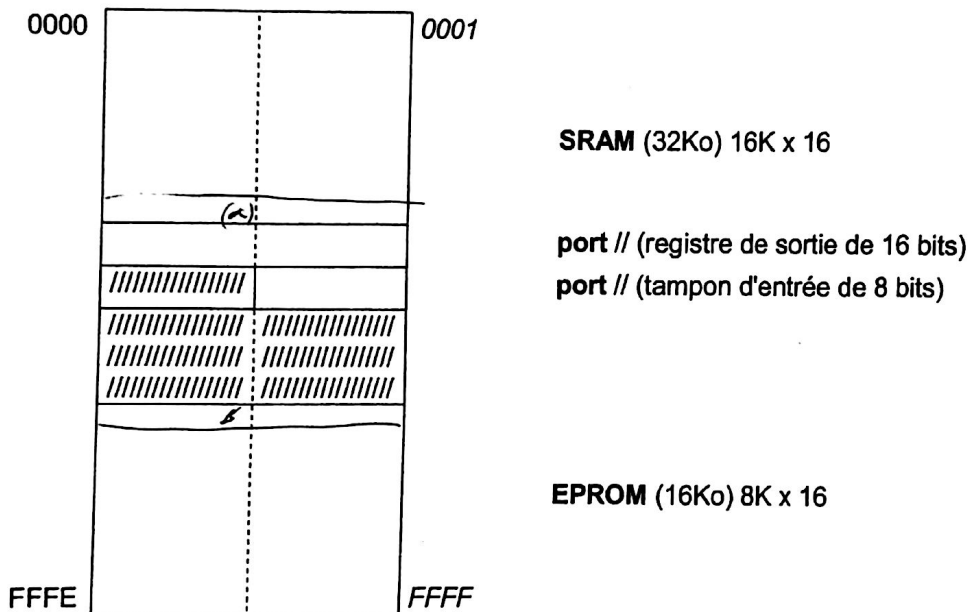


Tableau 1: Carte de l'espace d'adressage ("address space map")

2. Interfaçage d'un port parallèle sur un bus de microprocesseur

a/ Le port parallèle de sortie est un registre REG : il est chargé avec la donnée que le CPU envoie sur le bus de données lorsque le CPU écrit à l'adresse du port. Cette donnée sera donc ensuite indéfiniment visible sur la sortie du registre et disponible à l'extérieur.

La logique qui commande le port à partir du bus se nomme "décodeur d'adresse" ("address decoder").

- Dessiner le schéma-bloc du port parallèle de sortie et de sa connexion au bus en supposant qu'il soit placé ("mapped") sur l'espace d'adressage à l'adresse obtenue au §1.
- Déterminer les équations logiques de la commande de chargement L ("Load") du registre à partir des signaux concernés du bus.

b/ Le port parallèle d'entrée est un tampon ("buffer") BUF. Lorsque le CPU lit à son adresse, ce tampon met la donnée extérieure présente à son entrée sur le bus : le CPU va donc la lire.

- Dessiner le schéma-bloc du port parallèle d'entrée et de sa connexion au bus en supposant qu'il soit placé ("mapped") sur l'espace d'adressage à l'adresse obtenue au §1.
- Déterminer les équations logiques de la commande de validation E ("Enable") du tampon à partir des signaux concernés du bus.

On supposera ici que les signaux du décodeur d'adresse et du tampon se sont stabilisés entre l'instant où ADS monte et le coup d'horloge suivant.

On suppose que le bus (très proche de celui du 486 et du Pentium) est constitué des signaux:

NOM PENTIUM	SÉMANTIQUE	EXPLICATION
ADS	ADdress Strobe	1 les signaux envoyés par le CPU sont valides (pas seulement l'adresse)
RDY	ReaDY	1 les signaux renvoyés au CPU sont valides
W	Write	1 CPU écrit 0 CPU lit
CLK	Clock	↑ synchronisation
BE0	Byte Enable 0	1 Accès à l'octet de rang 0 dans le mot (donc de poids faible, à droite, rang 1) 0 cet octet n'est pas accédé
BE1	Byte Enable 1	1 Accès à l'octet de rang 1 dans le mot (donc de poids fort, à gauche, rang 1) 0 cet octet n'est pas accédé
DATA	DATA bus	16 bits de données D ₁₅ ..D ₀
ADDRESS	ADDRESS bus	15 bits de poids fort de l'adresse A ₁₅ ..A ₁

Tableau 2: Principaux signaux du bus de contrôle du Pentium

La ligne IO n'est pas prise en compte dans cet exercice.

BE0 et BE1 sont conformes au Pentium.

c/ pourrait-on multiplexer temporellement (envoyer l'une après l'autre) adresse et donnée sur les mêmes lignes AD sans perte de vitesse comme dans le bus PCI (Peripheral Card Interconnect) ?