

Examen d'Architecture Avancée

E. Mesnard
06 décembre 2005

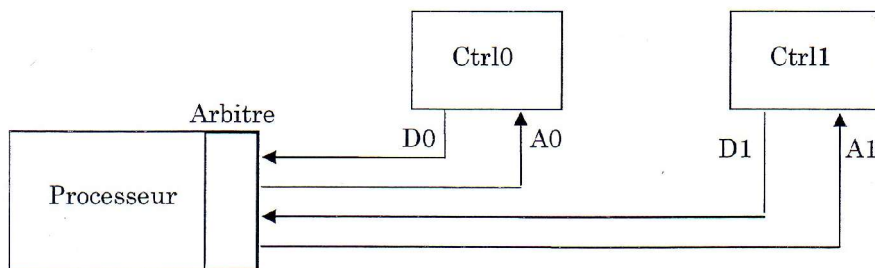
Documents et calculatrices **autorisés**.
Durée : **2 heures**

Exercice 1 (5 points) Demandes d'interruptions

L'étude porte sur les demandes d'interruptions à l'entrée d'un processeur. Lorsque les entrées de demande d'interruption sont distinctes, les circuits demandeurs (généralement des contrôleurs d'E/S qui cherchent à obtenir un accès sur un bus) peuvent éventuellement rentrer en conflit (on parle alors de « demandes simultanées »).

Une solution (pour le processeur) consiste à exploiter un « Arbitre » qui gère ces demandes en n'attribuant qu'un accord à la fois.

On suppose ici que l'architecture ne comporte que **deux** contrôleurs d'E/S (Ctrl0 et Ctrl1). Ces derniers informent l'arbitre de leur demande à l'aide des signaux D0 et D1. Di vaut 1 quand le circuit fait une demande d'interruption, et 0 soit quand il ne fait pas de demande, soit quand il libère la ressource accordée. L'arbitre donne son accord à l'aide de ses deux sorties A0 et A1. Si Ai vaut 1, alors la ressource est accordée au circuit « Ctrl*i* » tant qu'il ne la libère pas.



Concevoir ce circuit « Arbitre » (synthèse Unité de Contrôle/Unité de Traitement) en précisant **clairement** le diagramme d'états (sans le synthétiser en bascules) sachant que la politique d'attribution est la suivante :

- L'accord est donné (donc, la ressource est attribuée) au premier qui en fait la demande.
- En cas de demandes simultanées, la ressource est attribuée selon une priorité tournante : la priorité change à chaque attribution (le dernier à l'obtenir devient le moins prioritaire).

Exercice 2 (7 points) La sélection conjointe dans une RAM

Le but de cet exercice est d'étudier l'impact de l'amélioration de l'architecture d'une RAM. Pour cela, nous allons étudier une mémoire **RAM 64K × 4 bits**.

- 1) Donner le schéma externe de cette mémoire. Préciser les noms (des fils et des bus) et les tailles des bus.
- 2) On désire concevoir cette mémoire avec des cellules élémentaires de 4×4 bits. Donner le schéma (simplifié) de représentation interne pour cette mémoire, en précisant les tailles des bus.
Donner le nombre de porte ET nécessaire au décodeur d'adresse, sachant qu'un démultiplexeur possédant k signaux de contrôle (donc 2^k sorties) nécessite 2^k portes ET avec, pour chacune, k entrées.
- 3) On désire maintenant appliquer la technique de sélection conjointe : cela consiste à répartir les cellules élémentaires (dans une représentation 2D carrée) de telle façon à ce que l'on puisse accéder simultanément aux lignes, d'une part, et aux colonnes, d'autre part. Donner le schéma (simplifié) de cette mémoire

En déduire le nombre total de portes ET nécessaires à cette nouvelle architecture.

- 4) Déterminer les adresses de sélection ligne et colonne qui sont activées lorsque l'on accède à un mot de 4 bits dont l'adresse d'entrée est l'équivalent binaire de 1234 (en base 10).

Exercice 3 (8 points) Circuit de contrôle d'une DRAM

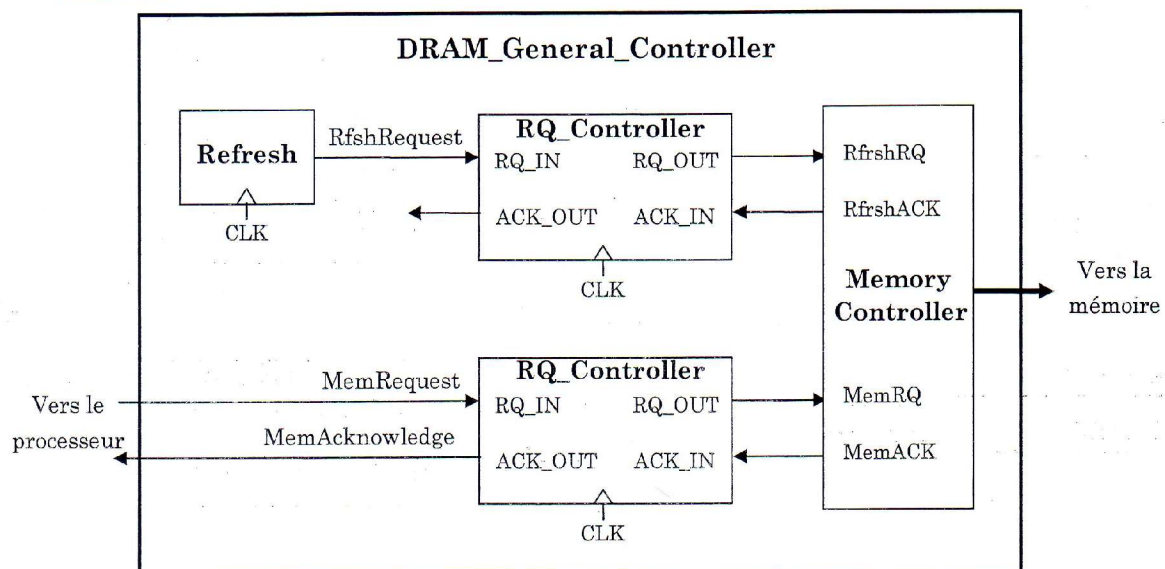
Cet exercice a pour objectif d'étudier le circuit de contrôle (« DRAM_General_Controller ») d'une mémoire dynamique (DRAM). Cette mémoire, de type IMS2801 contient 256 lignes de 1024 bits à rafraîchir au pire toutes les 4,4ms.

Les autres caractéristiques de la mémoire sont :

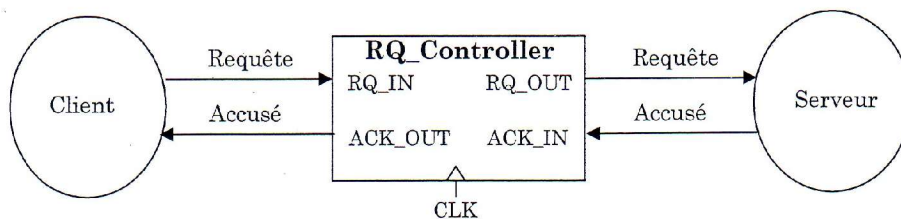
$T_{\text{Accès RAS}} = T_{\text{Write}} = 120\text{ns}$

$T_{\text{Accès CAS}} = 20\text{ns}$

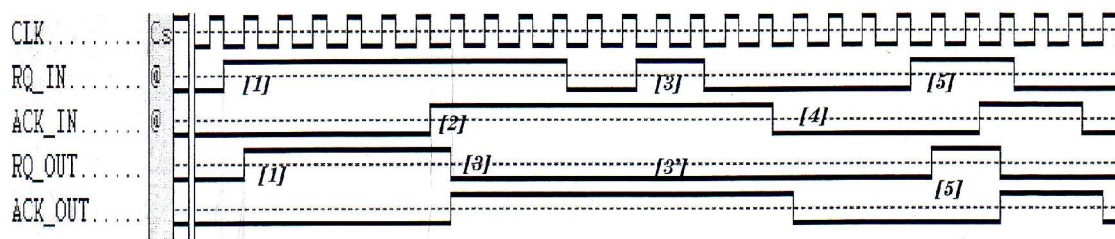
$T_{\text{Précharge RAS}} = 75\text{ns}$



- 1) Sachant que le rafraîchissement se fait ligne à ligne, donner la période maximale de rafraîchissement d'une ligne.
- 2) Concevoir le circuit « Refresh » provoquant la demande de rafraîchissement d'une ligne (signal RfshRequest), sachant que l'on dispose d'une horloge (notée CLK) de période 50ns.
- 3) Concevoir le circuit générique « RQ_Controller ».
Ce dernier implante un protocole basé sur des échanges de requêtes (RQ) et d'accusés de réception (ACK). Il garantit que chaque demande (RQ_IN) ne sera traitée qu'une fois, et une seule.
Ainsi, il possède pour cela 3 entrées (CLK, RQ_IN et ACK_IN) et 2 sorties (RQ_OUT et ACK_OUT) :



Le protocole implanté est le suivant :



- [1] un « client » fait une requête RQ_IN que le « RQ_Controller » passe au « serveur » par RQ_OUT.
- [2] le serveur réagit, dès qu'il le peut, à cette demande (RQ_OUT) par un accusé de réception.
- [3] cet accusé de réception, perçu sur ACK_IN par le « RQ_Controller », bloque toutes nouvelles émissions de requêtes (ainsi, la sortie RQ_OUT n'est pas transmise au serveur [3']).
- [4] quand le serveur a terminé le traitement, il relâche son accusé de réception, qui laisse alors passer les nouvelles requêtes (en [5]).

Durant tout le traitement, l'accusé ACK_OUT est remis au client. C'est l'image de ACK_IN fourni par le serveur, à un léger décalage temporel près.