

*Nom : AIDER*  
*Prénom : Smail*  
*N°Etudiant : 3603379*  
*Parcours : SAR*  
*Résponsable : M Alain Greiner*

***Compte-Rendu TP3***  
***Architecture interne du contrôleur de cache***

## **C) Application logicielle**

### **Question C1 :**

L'adresse de la première instruction de la fonction main() est : 0x00400000

- Elle correspond à la première adresse du segment code.

L'adresse de la première instruction de la boucle loop est : 0x0040000C

- 0x00400000 + 3 mots (0x0C).

### **Question C2 :**

Adresse du tableaux A : 0X01000080

- 0x01000000 + 0x01000080 (124+4 octets)

Adresse du tableaux B : 0x00000100

- 0X01000080 + 0x00000080 (20+48 = 68 octets → 128)

Adresse du tableaux C : 0x00000180

- 0x00000100 + 0x00000080 (20+48 = 68 octets → 128)

### **Question C3 :**

L'instruction « sw » est placée après l'instruction « bne » à cause du pipeline. Comme c'est une instruction qu'on est sûr d'exécuter, on préfère la mettre après pour qu'il n'y est pas un vide.

### **Question C4 :**

On a besoin de 5 cycles pour exécuter une itération de boucle.

## **D) Fonctionnement du cache instruction**

### **Question D1:**

- Cache Size : 128 octets

- Block Size : 16 octets

- 1-Way

Donc :  $128 / 16 = 8$  Sets

→ 3 bits pour coder le N° de Sets

→ 4 bits pour l'offset dans le Bloc

→ 25 bits pour le TAG

Adresse 0x00400000

→ Set : 0x000

→ Offset : 0x0000

→ Tag : 0x110000110101 = 0x800

### Question D2:

TAG	V	Word 3	Word 2	Word 1	Word 0
0x800	1	I0(loop)	I-1	I-2	I-3(main)
0x800	1	I4	I3	I2	I1
0x800	1			I6	I5(bne)

 MISS

Les instructions qui ont provoqués un MISS :

- I0 : lw (0x00400010)
- I5 : bne (0x00400020)

### Question D3:

Le contenu du cache instruction à la fin de la 20<sup>e</sup> itération ne change pas de celui qu'on a obtenu jusqu'à la fin de la 1<sup>e</sup> itération, car on a toutes les instructions dans le cache.

Le nombre total de MISS: 3

Le nombre total d'instructions : 144

→ Taux de MISS: 2 % (144 / 3)

### Question D4:

L'état MISS\_SELECT est indispensable pour les caches Set-Associative «Nombre de Ways > 1» car il faut choisir le way(la victime) à évincer.

### Question D5:

$$A = REQ . IUNC . IMISS$$

$$B = REQ . \overline{IUNC} . IMISS$$

$$C = \overline{A} + \overline{B}$$

$$F = VALID . \overline{ERROR}$$

$$G = VALID . ERROR$$

$$H = \overline{VALID}$$

$$I = 1$$

$$J = \overline{VALID}$$

$$K = VALID . ERROR$$

$$L = VALID . \overline{ERROR}$$

$$M = 1$$

$$N = 1$$

$$O = 1$$

### Question D5:

Cet automate est forcé dans l'état IDLE lors de l'activation du signal RESETN.

Autres effets :

- Tout les bits valides sont remis à zéro.
- Tout les registres Tag sont initialisés à 1.
- La pile LRU est initialisée à l'ordre par défaut, l'ordre décroissant des secteurs.
- Tout les secteurs du cache sont dans l'état déverrouiller.

### E) Fonctionnement du cache de données

#### Question E1:

TAG | SET | BYTE

Adresse A[0] : 0x01000080 → 0x2001 | 000 | 0000

Adresse B[0] : 0x01000100 → 0x2002 | 000 | 0000

→ Les deux tableaux partagent le même N° de Set.

TAG	V	Word 3	Word 2	Word 1	Word 0
0x2002	1	$\text{D}_{A3} - \text{D}_{B3}$	$\text{D}_{A2} - \text{D}_{B2}$	$\text{D}_{A1} - \text{D}_{B1}$	$\text{D}_{A0} - \text{D}_{B0}$
0x2002	1	$\text{D}_{A7} - \text{D}_{B7}$	$\text{D}_{A6} - \text{D}_{B6}$	$\text{D}_{A5} - \text{D}_{B5}$	$\text{D}_{A4} - \text{D}_{B4}$
0x2002	1	$\text{D}_{A11} - \text{D}_{B11}$	$\text{D}_{A10} - \text{D}_{B10}$	$\text{D}_{A9} - \text{D}_{B9}$	$\text{D}_{A8} - \text{D}_{B8}$
0x2002	1	$\text{D}_{A15} - \text{D}_{B15}$	$\text{D}_{A14} - \text{D}_{B14}$	$\text{D}_{A13} - \text{D}_{B13}$	$\text{D}_{A12} - \text{D}_{B12}$
0x2002	1	$\text{D}_{A19} - \text{D}_{B19}$	$\text{D}_{A18} - \text{D}_{B18}$	$\text{D}_{A17} - \text{D}_{B17}$	$\text{D}_{A16} - \text{D}_{B16}$

 MISS

$\text{D}_x \text{X}$  Évincé

Les instructions qui entraînent un MISS sur le cache de données sont :

- lw # \$10 ← A[0]
- lw # \$11 ← B[0]

#### Question E2:

Les deux tableaux partagent les mêmes numéro de SET donc il va y avoir un Miss pour chaque donnée chargée, pour les deux tableaux.

Deux types de Miss :

- Miss Compulsif
- Miss Conflit

Taux de Miss : 100 %

### Question E3:

$$\begin{array}{ll} A = REQ . \overline{WRITE} . DUNC . DMISS & I = 1 \\ B = REQ . \overline{WRITE} . \overline{DUNC} . DMISS & J = \overline{VALID} \\ D = REQ . WRITE . \overline{DMISS} . WOK & K = VALID . ERROR \\ E = REQ . WRITE . DMISS . WOK & L = VALID . \overline{ERROR} \\ C = \overline{A} + \overline{B} + \overline{D} + \overline{E} & M = 1 \\ F = VALID . \overline{ERROR} & N = 1 \\ G = VALID . ERROR & O = 1 \\ H = \overline{VALID} & P = 1 \end{array}$$

## F) Accès au PIBUS

### Question F1:

Les écritures sont plus prioritaires pour une raison de cohérence de données.

Si on avait une demande de lecture et d'écriture en même temps (possible à cause du pipeline), il faut d'abord faire l'écriture puis la lecture.

### Question F2:

Le mécanisme utilisé pour transmettre une requête de lecture vers l'automate PIBUS\_FSM :  
Les deux caches mettent la valeur dans un tampon. Ce tampon peut être :

- Une FIFO
- Un Buffer protégé par une bascule Set/Reset

### Question F3:

L'automate PIBUS\_FSM n'a pas besoin de signaler qu'une requête d'écritures transmise par le tampon d'écritures postées s'est terminée parce que le processeur n'a pas à attendre une réponse lorsqu'il transmet une requête sur le tampon.

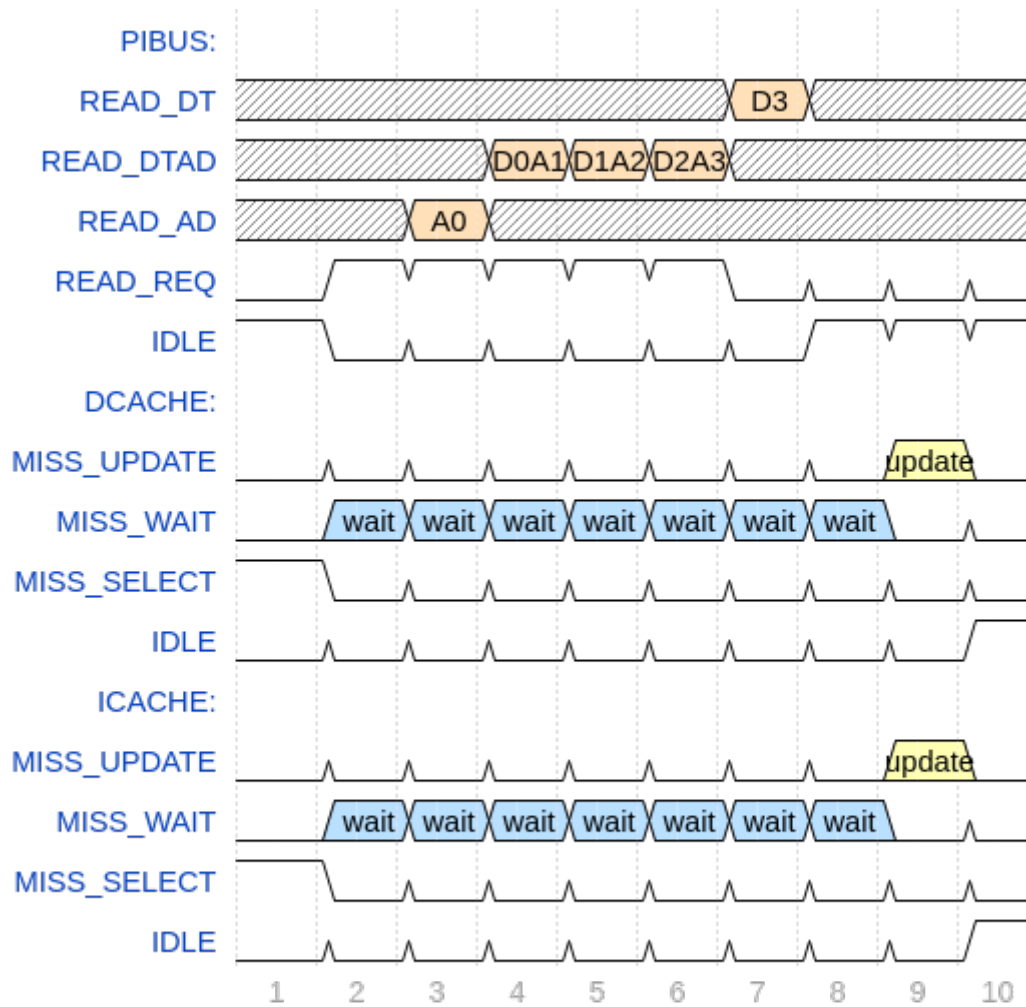
Dans le cas d'une transaction d'écriture, la réponse permet au maître de savoir si l'écriture a bien été effectuée.

### Question F4:

$$\begin{array}{ll} X = SC . ROK & E = GNT \\ B = (ACK = WAIT) & E' = \overline{GNT} \\ B' = \overline{GNT} & F = LAST \\ C = 1 & F' = \overline{LAST} \\ D = (ACK \neq WAIT) & G = LAST . (ACK \neq WAIT) \\ D' = (ACK = WAIT) & G' = \overline{LAST} . (ACK = WAIT) \\ Y = \overline{SC} . \overline{ROK} . (IUNC + IMISS + DUNC + DMISS) & H = (ACK \neq WAIT) \\ Z = \overline{Y} + \overline{X} & H' = (ACK = WAIT) \end{array}$$

### Question F5:

Le Chronogramme :



### Question F6:

Le nombre de cycles pour exécuter les 20 itérations de la boucle est de 625 cycles :

- Cycle 71 : exécution de la première instruction de la boucle « loop »
- Cycle 696 : exécution de la dernière instruction

## **G) Accès au PIBUS**

### **Question G1:**

Le processeur exécute sa première instruction au cycle 10,  
Cette instruction correspond au code de reset (boot).  
Le cout de MISS sur le cache instructions est de 10 cycles :  
- Cycle 00 : Demande de l'adresse 0xbfc00000.  
- Cycle 10 : Réception d'une réponse.

### **Question G2:**

Le processeur se branche au programme principal main() au cycle 57 :  
- Le processeur demande l'adresse du main (0x400000) au cycle 47.  
- Il reçoit une réponse au cycle 57.

### **Question G3:**

Le cout de MISS sur le cache de données est de 10 cycles :  
- Cycle 72 : Demande de la donnée A[0]  
- Cycle 82 : Réception d'une réponse

Le processeur termine l'exécution de la première itération de la boucle au cycle 108.  
La durée totale de la première itération est de  $108 - 61 = 47$  cycles (en prenant en compte le cout de MISS pour la première instruction, la première instruction s'exécute au cycle 71 et pas 61).

### **Question G4:**

La seconde itération dure :  $138 - 108 = 30$  cycles.  
La troisième itération dure :  $169 - 139 = 30$  cycles.  
Le cout de MISS pour les itérations 2 et 3 est plus élevé que pour la première itération à cause d'une requête d'écriture.

### **Question G5:**

Le taux de MISS sur le cache de données à la fin de l'exécution de la boucle est de 100 %.  
La durée totale du programme main() est de :  $(3266 - 47) - (3265 - 697) = 651$  cycles  
- Début du programme : cycle 47                      - Début d'affichage : cycle 697  
- Fin du programme : cycle 3266                      - Fin d'affichage : cycle 3265

