

Communications de type *bus* dans une machine

Architecture avancée, 2018-2019

Les « bus » sont le moyen utilisé pour interconnecter entre eux un petit nombre de composants. C'est typiquement ce que l'on va trouver dans une machine composée de quelques processeurs, quelques mémoires et quelques périphériques.

Ex. 1 : Communications sur PI-Bus

Le but de cet exercice est de concevoir des automates maître et esclave pour PI-Bus pour un sous ensemble des fonctionnalités de ce bus. Nous définissons un PI-Bus allégé dont les signaux sont les suivants :

Nom	émis	reçu	cycle	commentaires
RESETN	environnement	tous	-	remise à zéro.
CLK	environnement	tous	-	horloge système, active sur front montant.
REQ_x	maître <i>x</i>	BCU	requête	Requête du maître <i>x</i> pour le bus.
GNT_x	BCU	maître <i>x</i>	requête	Donne le bus au maître <i>x</i> .
LOCK	maître	BCU	adresse	Interdit l'arbitrage au cycle suivant.
READ	maître	esclave	adresse	Lecture si 1, écriture si 0.
OPC[3 :0]	maître	esclave, BCU	adresse	Type de transfert.
A[31 :2]	maître	esclave, BCU	adresse	Adresse du transfert.
SEL_y	BCU	esclave <i>y</i>	adresse	Sélection de l'esclave <i>y</i> .
D[31 :0]	maître, esclave	esclave, maître	donnée	Donnée.
ACK[2 :0]	esclave, BCU	maître	donnée	Réponse au transfert.

BCU désigne le contrôleur de bus (bus control unit). Les signaux sont tous actifs à l'état haut, hormis le **RESETN** qui est actif à l'état bas. Nous rappelons que les adresses sont sur 30 bits, les 2 bits de poids faibles étant définis par les opcodes spécifiant le transfert ; dans notre cas, les octets significatifs sont correctement alignés pour les transferts entre mémoire et processeur. Les signaux sont positionnés dans les cycles indiqués comme le montre le chronogramme de la figure 1.

Afin de simplifier les exercices, nous faisons les hypothèses suivantes :

- il n'y a pas de maître par défaut, et les maîtres font des requêtes d'exactly un élément : mot, demi-mot, octet. Les seuls opcodes autorisés sont définis ci-dessous.

OPC	symbole	description
0010	WDU	transfert de mots.
10a0	HWa	transfert de demi-mots, adresse du demi-mot telle que adresse $\bmod 4 = a0$.
11ab	BYab	transfert d'octets, adresse de l'octet telle que adresse $\bmod 4 = ab$.

- les esclaves ayant été sélectionnés peuvent donner les réponses suivantes (codées sur ACK) à une requête de transfert :

ACK	symbole	description
011	RDY	transfert effectué.
010	ERR	transfert avorté.

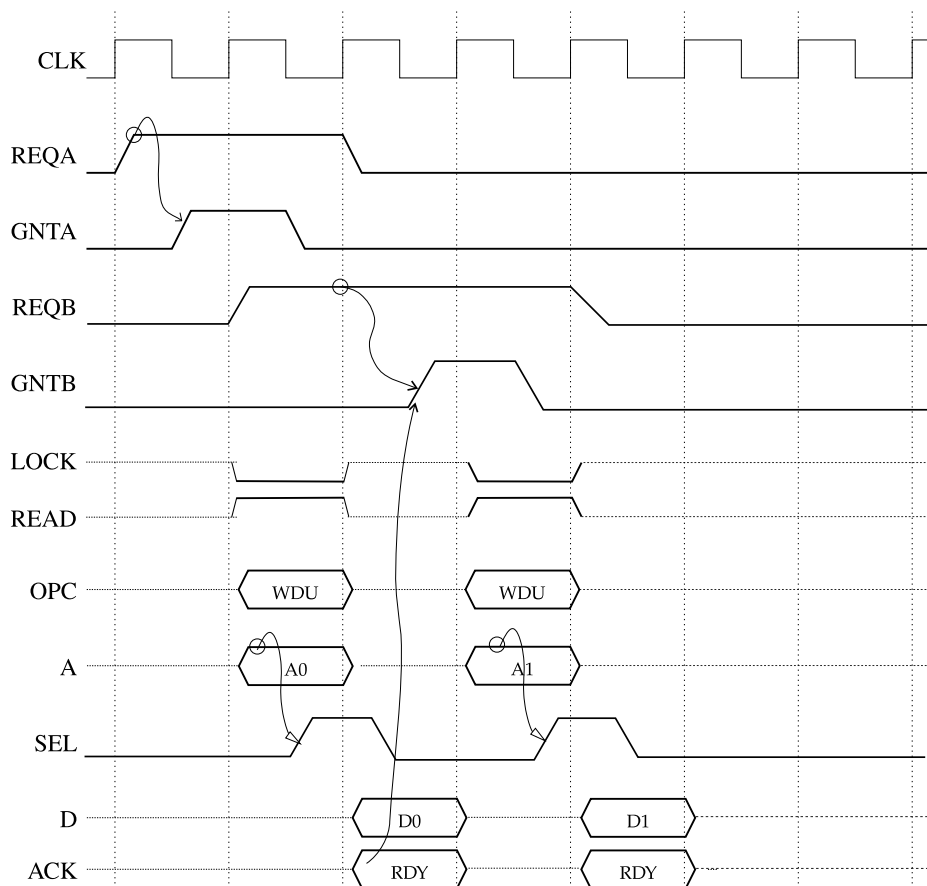


FIGURE 1 – Déroulement de 2 transactions sur le PI-Bus.

L'interface de la mémoire utilisée (mémoire asynchrone) ainsi que celle du MIPS sont rappelées sur la figure 2. Pour la mémoire, les entrées \overline{OE} (Output Enable) et \overline{WE} (Write Enable) sont actives au niveau bas.

Pour le MIPS, on suppose¹ que :

- les mots et demi-mots sont alignés à droite dans les registres.
- les bits **ADR[1 :0]** du MIPS sont à 0 pour les lectures/écritures de mots ;
- le bit **ADR[0]** du MIPS est à 0 pour les écritures de $\frac{1}{2}$ mots. Le bit **ADR[1]** spécifie si c'est le demi-mot d'adresse zéro ou le demi-mot d'adresse 2 dans le mot qui doit être écrit ;
- pour l'écriture des octets, les bits **ADR[1 :0]** spécifient l'adresse dans le mot de l'octet devant être écrit ;

On cherche à réaliser un circuit d'interface mémoire / PI bus n'insérant pas de cycles d'attente en réponse à une demande du processeur, en supposant que la boîte mémoire utilisé est de type asynchrone (réponse à une demande de lecture durant le même cycle d'horloge). Ce circuit d'interface est de type PC/PO : partie opérative (PO) et automate de contrôle (PC).

Question 1 – Quels sont les signaux d'entrée et de sortie de ce bloc de mémoire connecté sur un PI-Bus ?

Les signaux émis et reçus par un esclave PI.

1. Ces hypothèses sont valides pour le MIPS3000.

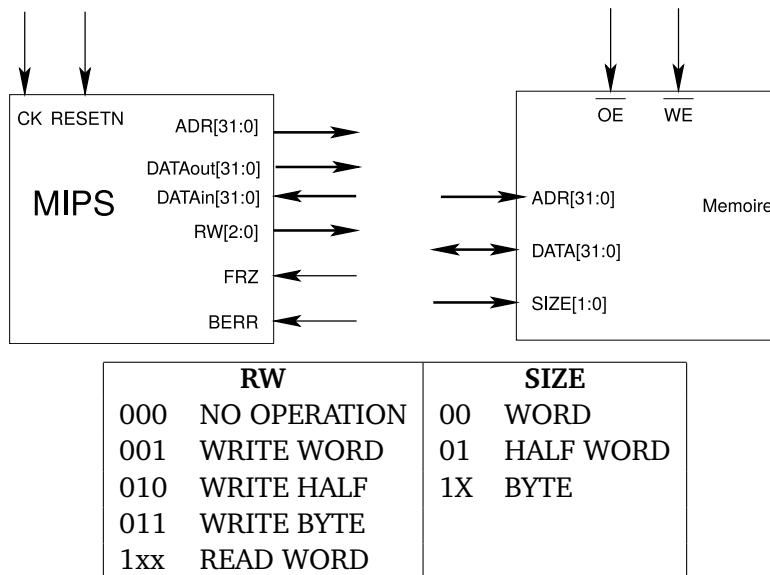
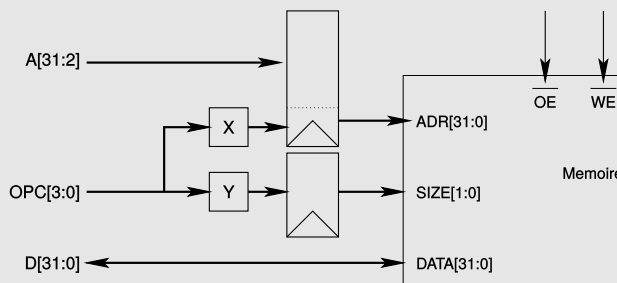


FIGURE 2 – Interfaces du MIPS et de la mémoire

Entrées Sorties
SEL **ACK**
READ **D**
D
A
OPC
RESETN
CLK

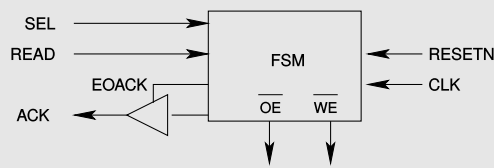
Les signaux en sortie ne sont activés que lorsque la mémoire est sélectionnée.

Question 2 – Proposez une partie opérative permettant à cette mémoire de se connecter au PI-Bus. Remarquez que sur le PI-Bus l'adresse, la commande d'écriture et l'opcode sont présentés un cycle avant la donnée à écrire ou avant l'apparition de la donnée lue.



X et Y sont des petites fonctions combinatoires de transcodage. Avec un tableau de Karnaugh, on trouve $X(1) = OPC(3) \cdot OPC(0)$, $X(0) = OPC(0)$, $Y(1) = OPC(2)$, $Y(0) = OPC(3)$

Question 3 – Quels sont les signaux d'entrée et de sortie de l'automate de contrôle ?



Le buffer 3 états peut aussi être positionné dans la PO pour faciliter la compréhension des étudiants.

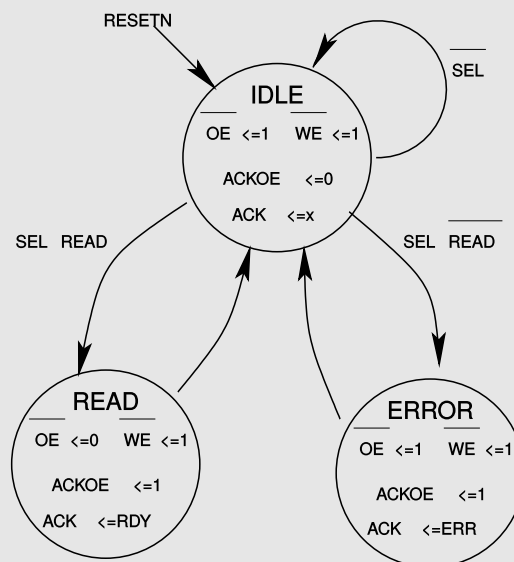
Question 4 – Cet automate est-il un automate de Moore ou un automate de Mealy? Justifiez votre réponse.

C'est un automate de Moore, car sur le chronogramme les sorties ne dépendent que de l'état courant.

On fait l'hypothèse que la mémoire est une ROM.

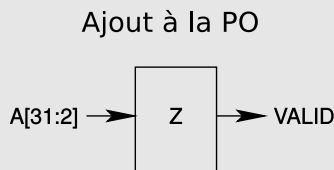
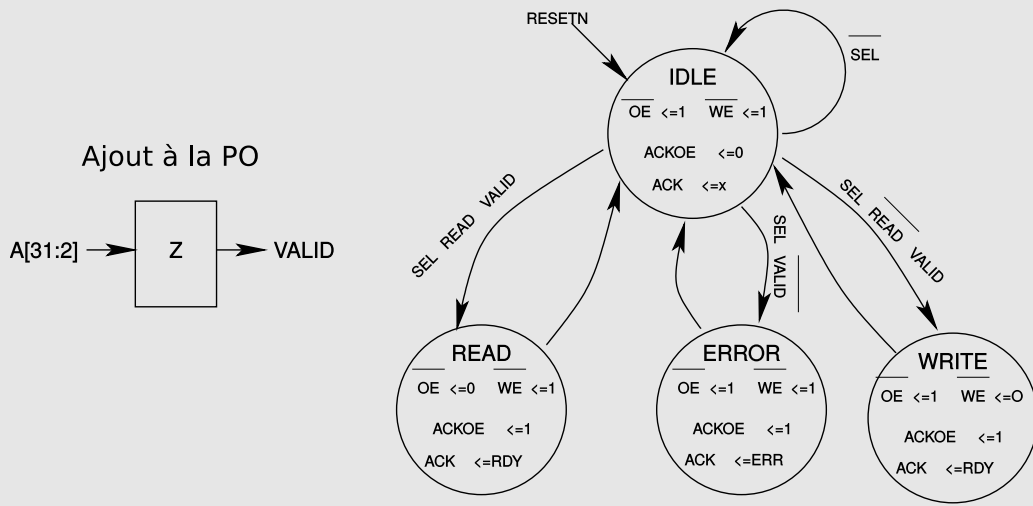
Question 5 – Proposez un graphe d'états de l'automate de contrôle qui permet les lectures et répond ERR en cas d'écriture.

Correction sans accès consécutif. Version simplifié pour être cohérent avec le chronogramme simplifié.



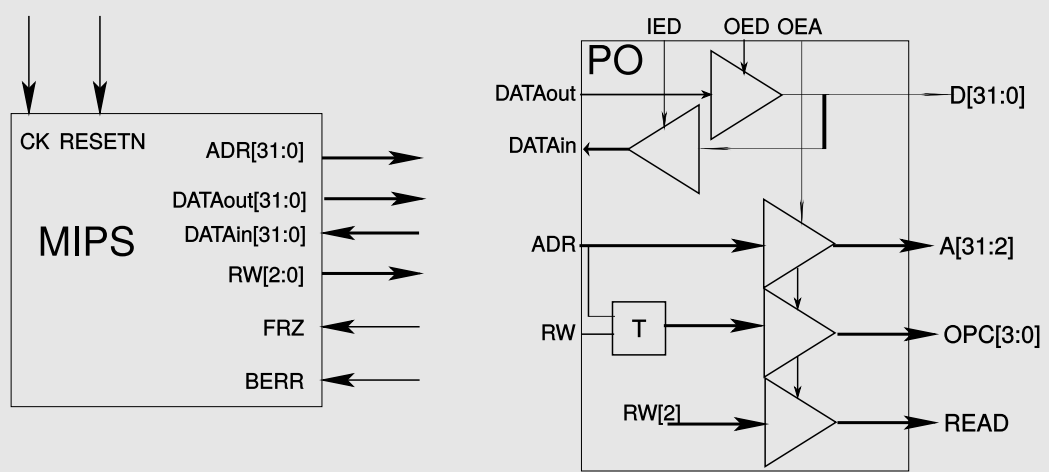
On fait l'hypothèse que la mémoire est une RAM.

Question 6 – Proposez le graphe d'états de l'automate d'état gérant les lectures, les écritures et les adresses hors mémoire (détectées par un circuit, inclus dans la partie opérative, qui génère un signal VALID si l'adresse est correcte)



Dans les questions suivantes, on fait l'hypothèse que l'esclave ne répond pas ERR. On cherche à réaliser le circuit d'interface au PI-bus pour la lecture de l'instruction par le MIPS R3000. Le processeur indique la requête sur les fils **RW** et l'adresse de l'instruction sur **ADR**. Le circuit d'interface doit en réponse positionner l'instruction sur la nappe **DATA**. Durant la récupération de l'instruction, le processeur ne doit pas continuer d'exécuter quoi que ce soit. Le signal **FRZ** permet de geler immédiatement (dans le cycle courant) le processeur durant la récupération de l'instruction.

Question 7 – Proposez une partie opérative permettant au MIPS d'envoyer ses requêtes sur le PI-Bus.



Les drivers de DATA sont déjà positionnés par le MIPS. Il faut déconnecter au besoin A, OPC et READ. La fonction combinatoire T s'obtient encore par un tableau de Karnaugh à 5 variables. Ce dernier peut

se réduire à :

$RW(2)/RW(1-0)$	00	10	11	01
0	Φ	$10ADR_10$	$11ADR_1ADR_0$	0010
1	0010	0010	0010	0010

On trouve :

$$OPC(3) = \overline{RW(2)}.RW(1)$$

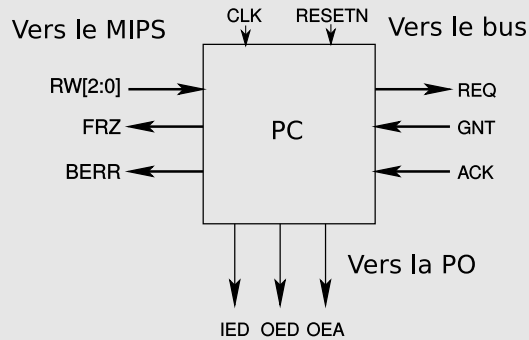
$$OPC(2) = \overline{RW(2)}.RW(1).RW(0)$$

$$OPC(1) = ADR_1 + \overline{RW(2)}.RW(1)$$

$$OPC(0) = \overline{RW(2)}.RW(1).RW(0).ADR_0$$

En utilisant les hypothèses sur les bits ADR sortant du MIPS, on sait que ADR_0 vaut 0 sur les cases 010 et 001. On peut donc simplifier $OPC(0) = \overline{RW(2)} \cdot ADR_0$. On peut aussi remarquer que le terme de $OPC(3)$ apparaît comme facteur commun à toutes les expressions.

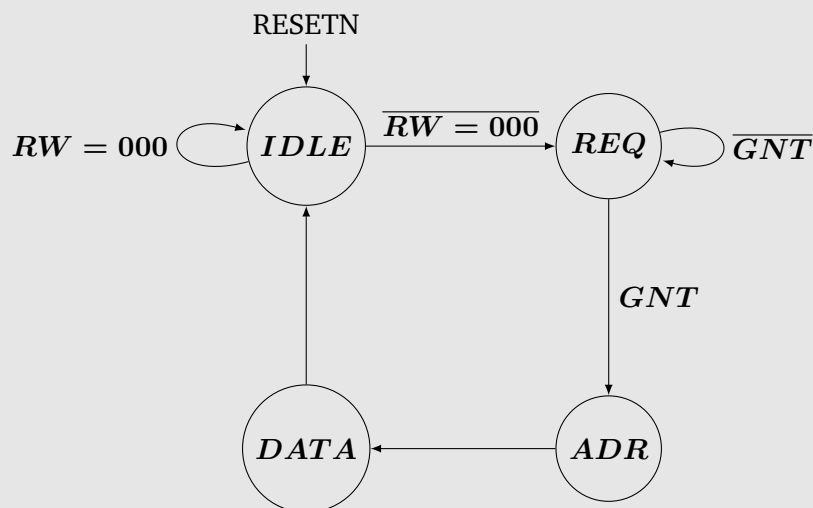
Question 8 – Quels sont les signaux d'entrée et de sortie de l'automate de contrôle ?



Question 9 – Cet automate est-il un automate de Moore ou un automate de Mealy ? Justifiez votre réponse.

C'est un automate de Mealy, car le signal **FRZ** est combinatoire. Lors du chargement d'une instruction, le processeur suppose que IR contient la nouvelle instruction au cycle suivant, or il faut plusieurs cycles pour aller récupérer la donnée de la mémoire, donc il faut immédiatement geler le processeur.

Question 10 – Proposez un graphe d'états pour cet automate.

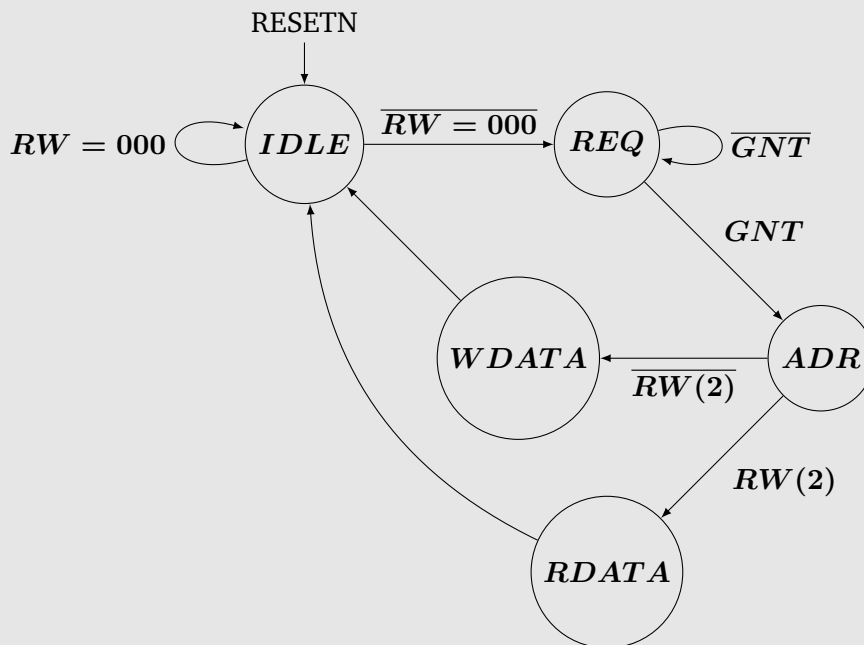


Etat	RW	GNT	OED	IED	OEA	BERR	REQ	FRZ
IDLE	000	Φ	0	0	0	0	0	0
	Φ	Φ	-	-	-	-	-	1
REQ	Φ	Φ	0	0	0	0	1	1
ADR	Φ	Φ	0	0	1	0	1	1
DATA	Φ	Φ	0	1	0	0	0	0

On cherche maintenant à réaliser l'automate de lecture/écriture du MIPS R3000. On suppose pour cela que le MIPS émet de nouveau sa requête grâce au signal **RW** pour un transfert dont l'adresse se trouve sur **ADR**. Comme précédemment, le signal **FRZ** permet de geler le processeur durant le transfert de données.

Question 11 – Proposez un graphe d'états pour l'automate de contrôle qui effectue les lectures et les écritures.

Le type d'écriture est géré combinatoirement dans le chemin de données, et on n'a pas à s'en préoccuper dans l'automate.



Etat	RW	GNT	OED	IED	OEA	BERR	REQ	FRZ
RDATA	Φ	Φ	0	1	0	0	0	0
WDATA	Φ	Φ	1	0	0	0	0	0

Plus simplement, on peut aussi garder l'automate de la question précédente et dire :

Etat	RW	GNT	OED	IED	OEA	BERR	REQ	FRZ
DATA	Φ	Φ	$\overline{RW(2)}$	$RW(2)$	0	0	0	0

Question 12 – Que faut-il faire en cas de réponse ERR de l'esclave ?

On propage l'erreur immédiatement au processeur.

Etat	RW	GNT	OED	IED	OEA	BERR	REQ	FRZ
DATA	Φ	RDY	$\overline{RW(2)}$	$RW(2)$	0	0	0	0
DATA	Φ	ERR	$\overline{RW(2)}$	$RW(2)$	0	1	0	0