

TD organisation mémoire

1 Le processeur et la mémoire

Soit un processeur à connecter à 4 boîtiers mémoire. Chaque boîtier est organisé en 2^{20} octets (1 Méga octet). Voir figure 1.

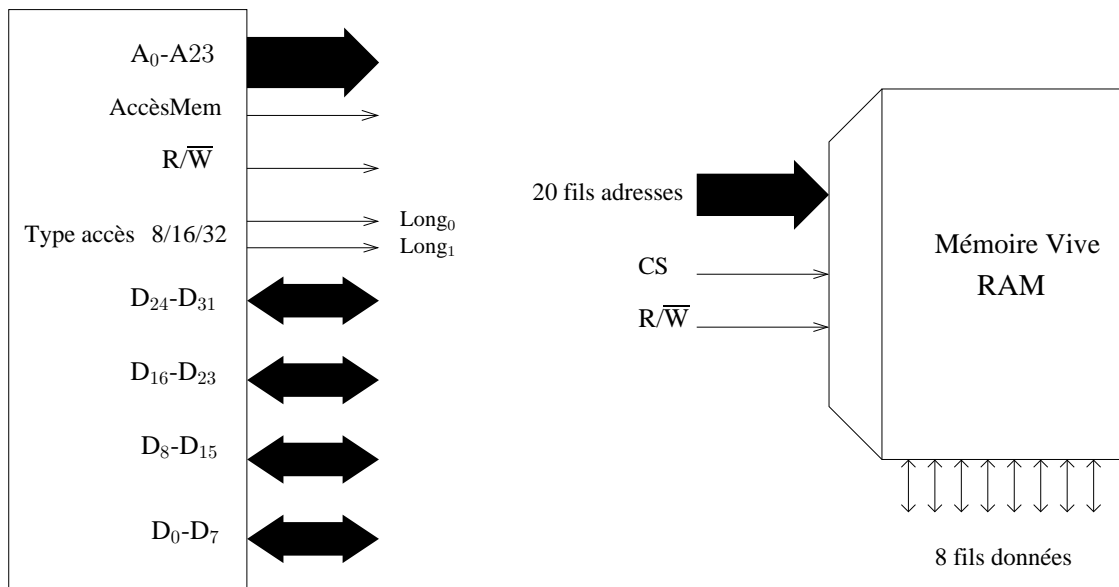


FIG. 1 – Description du processeur et des boîtiers RAM

Le processeur a un bus de données de 32 bits et un bus d'adresses de 24 bits. Le processeur peut vouloir écrire/lire différents types de données (un octet, deux octets, quatre octets) en un cycle d'horloge. Nous supposons ici que la convention de ce processeur est «big endian».

2 Accès à des données des mots de 4 octets

Nous supposons dans un premier temps que le processeur ne peut accéder qu'à des mots de 4 octets (instructions assembleur ldr et str du ARM).

1. Comment connecter la RAM pour accéder à quatre octets en un cycle ?
2. Dessiner le processeur et les 4 boîtiers de RAM ainsi que leurs connexions.

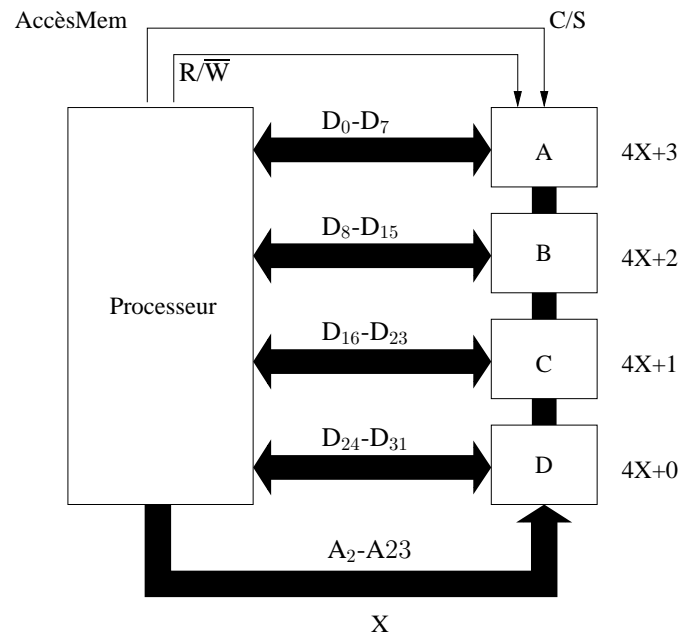


FIG. 2 – Accès à quatre octets en un cycle

Comme illustré dans la figure 2, en envoyant X sur les bits de poids fort du bus d'adresses, on peut accéder aux octets :

- de poids fort d'adresse $4X$ stocké dans D via $D_{24}-D_{31}$.
- de poids moyen d'adresses $4X+1$ et $4X+2$ stockés dans B et C via D_8-D_{15} et $D_{16}-D_{23}$.
- de poids faible d'adresse $4X+3$ stocké dans D via D_0-D_7 .

3. Dessiner les octets d'adresses 4, 5, 6 et 7 (données par le processeur) dans les boîtiers de RAM. L'emplacement de ces octets dépend-il de la convention little/big endian ?
4. Le processeur émet l'adresse 256 (en décimal) pour accéder à un mot long. Donner les adresses correspondantes arrivant sur les boîtiers de RAM.
5. Si le processeur accède à l'adresse "8 Méga" que se passe-t-il ? Comment vérifier que de la mémoire est présente pour une adresse donnée ?

3 Accès à des données de taille différente

Problème des tailles multiples : Le processeur peut maintenant accéder des mots de un, deux ou quatre octets (instruction assembleur ldr, str, strb, strh, ldrb,

ldrh). Pour cela il donne le code de la taille sur les broches Long₁ et Long₀ comme suit :

Long ₁ Long ₀	:	Taille
00	:	1 octet
01	:	2 octets
10	:	4 octets
11	:	Erreur

1. Quelles sont les combinaisons de parité d'adresses et de tailles d'objet permises pour un accès en un seul cycle ? Décrire le décodeur d'adresses. La fonction de décodage doit calculer les signaux d'accès à chacun des boîtiers utilisés et le signal d'erreur d'accès à une adresse invalide.
2. Le processeur émet l'adresse 258 (en décimale) pour accéder à un mot de deux octets. Donner les adresses correspondants côté mémoire.
 - 1 octet : toutes adresses autorisées : on prend dans le bon boîtier, le processeur remet l'octet dans le poids faible du registre. A titre d'exemple, l'adresse 258 est de la forme $4X+2$, l'octet est lu dans B. En interne, le processeur recale cet octet sur les bits 7-0 d'un registre.
 - 2 octets : pour les adresses $4X+3$, les octets concernés, $4X+3$ et $4(X+1)$ se trouvent dans A et D respectivement. cette combinaison est impossible car il faudrait mettre à la fois X et X+1 dans le bus d'adresses entre A₂ et A₂₃. Les autres combinaisons sont possibles mais en pratique on interdit toutes les adresses impaires.
 - 4 octets : seulement les adresses multiples de 4 sont autorisées.

En résumé, les cas d'erreur sont les suivants :

- 2 octets : adresse impaire \implies Erreur
- 4 octets : adresse non multiple de quatre \implies Erreur

Connexion de la RAM : on rajoute un boîtier de sélection des boîtiers de la mémoire. Ce boîtier est donné dans la figure 3.

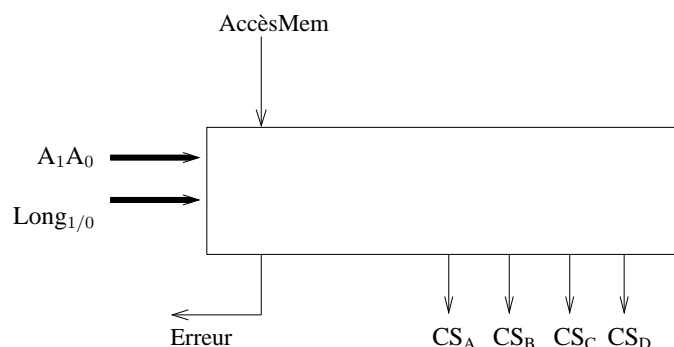


FIG. 3 – Décodeur d’adresse dans le cas d’accès à des tailles multiples

La table de vérité est donnée dans la figure 1.

A ₁	A ₀	Long ₁	Long ₀	CS _A	CS _B	CS _C	CS _D	Erreur
0	0	0	0	0	0	0	1	0
0	1	0	0	0	0	1	0	0
1	0	0	0	0	1	0	0	0
1	1	0	0	1	0	0	0	0
0	0	0	1	0	0	1	1	0
0	1	0	1	0	0	0	0	1
1	0	0	1	1	1	0	0	0
1	1	0	1	0	0	0	0	1
0	0	1	0	1	1	1	1	0
0	1	1	0	0	0	0	0	1
1	0	1	0	0	0	0	0	1
1	1	1	0	0	0	0	0	1
φ	φ	1	1	0	0	0	0	1

TAB. 1 – Fonction de décodage d’adresses

4 Augmentation de la mémoire disponible

On ajoute 12 boîtiers pour remplir tout l’espace d’adressage (2^{24} octets). On dispose au total de 16 boîtiers de 1 Mega octet chacun et possédant comme précédemment 20 fils d’adresses et 8 fils de données.

1. Décrire l'organisation de la mémoire et le décodage d'adresses sachant que l'on veut pouvoir accéder à des octets, à des mots de 16 bits ou à des mots de 32 bits ?

On assemble comme précédemment 4 blocs de 4 boîtiers de 1 Megaoctets de RAM chacun. Ces 4 blocs peuvent être vus comme des boîtiers possédant 22 fils de bus adresses et 32 fils de bus données. On rajoute un fils de sélection CS à ces blocs qui doit être à 1 pour sélectionner le bloc. Les fils A_0 à A_{21} sont connectés à chacun de ces blocs et les bits de poids fort A_{22} et A_{23} du bus d'adresses permettent de choisir le bon bloc comme suit :

A_{23}	A_{22}	
0	0	Bloc 0
0	1	Bloc 1
1	0	Bloc 2
1	1	Bloc 3

La figure 4 donne les connexions des 4 blocs dénommés A0, A1, A2 et A3.

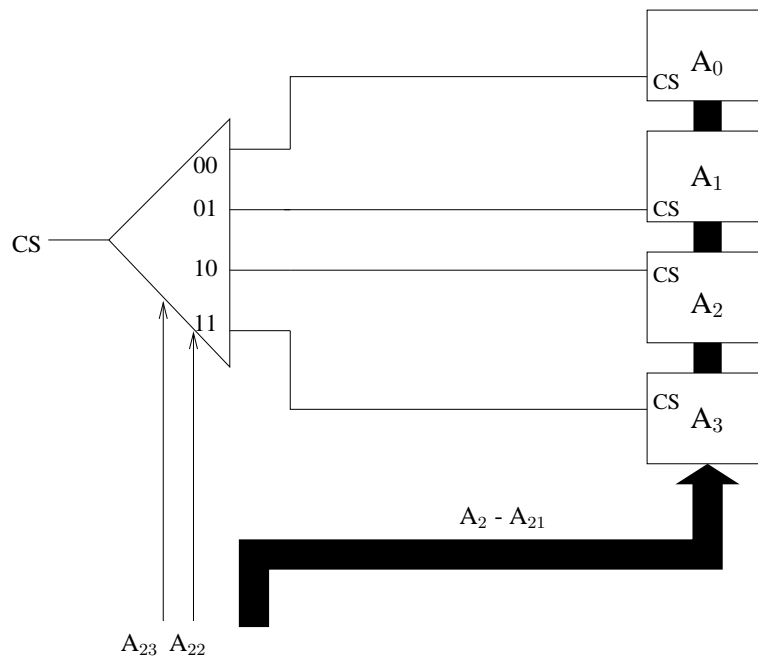


FIG. 4 – Sélection du boîtier

5 Boîtiers de tailles différentes

Il existe un mode superviseur et un mode utilisateur sur ce processeur. Le signal envoyé par le processeur vers l'extérieur S/\bar{U} indique s'il est en mode superviseur ou utilisateur. Une erreur doit être émise dans le cas où un accès à une zone superviseur est demandée alors que le processeur n'est pas en mode superviseur.

On considère une mémoire dans laquelle l'accès à des octets, des mots de 16 bits ou 32 bits est possible.

La mémoire est organisée comme suit :

- Quatre boîtiers de mémoire morte (accès en lecture seulement) de 256 Ko chacun. La ROM occupe les adresses [0 à 1M [en mode superviseur seulement. C'est une partie du noyau du système qui s'exécute au moment du "boot" de la machine.
- Quatre boîtiers de mémoire vive de 256 Ko chacun, occupent les adresses [3 à 4M [en mode superviseur seulement. C'est la pile système.
- Huit boîtiers de mémoire vive de 1 Mo chacun, occupent [8 à 16M [accessibles dans les deux modes.
Ils contiendront les programmes et données utilisateurs.
- Un boîtier d'E/S huit bits accessible en mode superviseur seulement. Il est considéré comme trois octets.
Deux broches (a1, a0) permettent d'adresser les 3 cases mémoires du coupleur :

a1, a0	case mémoire
0 0	registre données en L/E
1 0	registre état en lecture
1 1	registre commande en écriture

On veut qu'il soit "vu" par le processeur aux adresses suivantes :

1M + 3	:	registre données	L/E
1M + 11	:	registre état	Lecture
1M + 15	:	registre commande	Ecriture

- Un boîtier d'E/S seize bits accessible en mode superviseur uniquement.
Deux broches (a1,a0) permettent d'adresser les 3 cases mémoires comme pour le premier coupleur.

On veut qu'il soit "vu" par le processeur aux adresses suivantes :

2M + 2	:	registre données	L/E
2M + 10	:	registre état	Lecture
2M + 14	:	registre commande	Ecriture

1. Sur quelle partie du bus données les coupleurs d'entrée sortie doivent ils être branchés pour pouvoir utiliser les adresses données ci-dessus ?
Quelles sont les fils du bus adresse du processeur qu'il faut brancher sur les deux fils (a1, a0) des coupleurs ?
2. Donner la fonction de décodage d'adresses pour cette mémoire. La fonction de décodage doit calculer les signaux d'accès à chacun des boîtiers utilisés et le signal d'erreur d'accès à la mémoire pour tous les accès interdits (accès utilisateur, écriture en ROM, accès impossible suivant l'adresse) .

A ₂₃	s/ \bar{u}	A ₂₂	A ₂₁	A ₂₀	r/ \bar{w}	A ₁₉	...	A ₃	A ₂	A ₁	A ₀	L ₀	L ₁	E	ROM				E/S		RAM				RAM																																																																							
															0	1	2	3	a	b	a	b	c	d	0	1	2	3	4	5	6	7																																																																
0	1	0	0	1	ϕ	0	...	0	0	1	1	0	0	0	0				1	0																																																																												
												0	1		0	0	0	0		0	0																																																																											
												1	ϕ									0	0	0	0	0	0																																																																					
												0	ϕ															0	0	0	0	0	0																																																															
												1	0																					0	0	0	0	0	0																																																									
												0	ϕ																											0	0	0	0	0	0																																																			
					0			1	0	0	0	0	0	0																																																																																		
					1			ϕ							0	0	0	0	0	0																																																																												
					ϕ			ϕ													0	0	0	0	0	0																																																																						
					Autre			ϕ																			0	0	0	0	0	0																																																																
					1			0																									0	1	ϕ	0	...	0	ϕ							1	0	0	0	0	0	0				0	0																																							
																																								0	1	0	0	0	0				0	0																																														
		1	ϕ	0		0	0		0	0	0																																																																																					
		0	ϕ									0	0	0	0	0	0																																																																															
		1	0															0	0	0	0	0	0																																																																									
		0	ϕ																					0	0	0	0	0	0																																																																			
		0	1																											0	0	0								0	0											0																																												
		1	ϕ																																																		0	0	0		0	0	0																																					
		ϕ	ϕ																																																									0	0	0	0	0	0																															
		Autre	ϕ																																																															0	0	0	0	0	0																									
		1	ϕ																																																																					ϕ	ϕ	ϕ	Autre	ϕ	...	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	0				0	0						
																																																																																		0	ϕ		0	0	0	0		0						
					ϕ			ϕ																									0	0	0	0	0																																																											
					Autre			ϕ																														0	0			0	0	0																																																				
1	ϕ			ϕ	ϕ	ϕ	Autre	ϕ	...	ϕ	ϕ																																		ϕ	ϕ	ϕ	ϕ	0	0						0																										0														
												0	ϕ	0	0	0	0																														0																																																	
ϕ	ϕ	0	0	0	0	0																																																																																										
Autre	ϕ						0	0	0	0	0																																																																																					

TAB. 3 – Fonction de décodage d'adresses pour les deux boîtiers servant aux entrées/sorties

