

Fiche 4 : Séquentiel/RAM



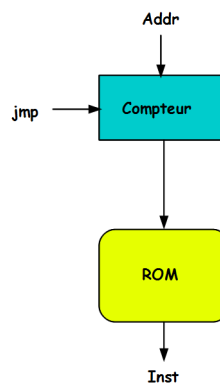
Romain Kamdem

🔗 EXERCICE 1 (PGCounter) 📝

On veut construire le circuit de la figure 1. Le fonctionnement du circuit est le suivant : le **compteur** est initialisé par l'entrée *addr* si *jmp* est égal à un dans le cas contraire l'entrée *addr* est ignorée et le **compteur** est incrémenté à chaque coup d'horloge.

La sortie du circuit est le mot mémoire qui est à l'adresse donnée par la sortie du **compteur**.

Figure 1: Chemin de données



1. Définir en VHDL l'entité de ce circuit
2. Construire en VHDL l'architecture de ce circuit
3. Construire un testbench pour valider votre circuit

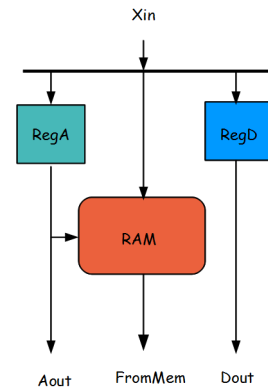
🔗 EXERCICE 2 (MemREGs) 📝

On se propose dans cet exercice de construire le circuit dont le chemin de données est décrit par la figure 2. C'est une mémoire avec deux registres avec le fonctionnement décrit dans le tableau 1.

Table 1: Spécifications

entrées	comportement
a	Ecrire Xin dans le registre RegA
d	Ecrire Xin dans le registre RegD
sa	Ecrire Xin dans la RAM à l'adresse RegA

Figure 2: Chemin de données



Les entrées **a**, **d** et **sa** peuvent être combinés c'est à dire on peut écrire simultanément dans les registres et la mémoire. Quand les entrées sont toutes à zéro, l'entrée **Xin** est tout simplement ignorée.

1. Définir en VHDL l'entité de ce circuit
2. Construire en VHDL l'architecture structurelle de ce circuit
3. Construire un testbench pour valider votre circuit (faite des séquences come suit : écrire une adresse dans le registre **RegA**, écrire une valeur dans le registre **RegD**, puis sauvegarder la valeur de **RegD** en mémoire)

🔗 EXERCICE 3 (InstDecodeur) 📝

On veut construire ici un circuit qui prend une instruction sur 16 bits et le décompose de la manière suivante :

Table 2: Décodage

bit 15	Type instruction
0	Donnée On recopie l'entrée sur W
1	Commande On décompose l'entrée comme indiqué dans le tableau 3

Table 3: Spécifications

bits	Sorties
15	ci
14	ignoré
13	ignoré
12	sm
6-11	opc
3-5	dst
0-2	cnd

Si le bit 15 est égal à zéro alors on recopie l'entrée sur la sortie **W** et on met le bit de poids fort de **dst** à un, dans le cas contraire on utilise le tableau 3

1. Définir en VHDL l'entité de ce circuit
2. Construire en VHDL l'architecture de ce circuit
3. Construire un testbench pour valider votre circuit

🔗 EXERCICE 4 (Condition) 📝

On se propose ici de construire un circuit qui prend en entrée Xin sur N bits qui a une sortie jmp sur un bit qui respecte les spécifications du tableau 4.

Table 4: Spécifications

lt	eq	gt	jmp vaut 1 quand
0	0	0	jamais
0	0	1	$Xin > 0$
0	1	0	$Xin = 0$
0	1	1	$Xin \geq 0$
1	0	0	$Xin < 0$
1	0	1	$Xin \neq 0$
1	1	0	$Xin \leq 0$
1	1	1	toujours

La sortie de ce circuit est donnée par l'équation suivante :

$$jmp = ((\neg(iszero(Xin) \oplus isneg(Xin))) \wedge gt) \vee ((eq \wedge iszero(Xin)) \vee (lt \wedge isneg(Xin))) \quad (1)$$

NB(voir bas de page) ¹

1. Définir en VHDL l'entité de ce circuit
2. Construire en VHDL l'architecture de ce circuit
3. Construire un testbench pour valider votre circuit

¹ \neg : not; \oplus : xor; \wedge : and; \vee : or;