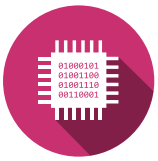


Unité arithmétique et logique

Laboratoire Digital Design

Contenu

1 Objectifs 1
2 Unité logique LU 2
2.1 Réalisation et simulation 2
3 Unité arithmétique AU 3
3.1 Réalisation et simulation 3
4 Unité arithmétique et logique 4
4.1 Sélection des résultats 4
4.2 Réalisation et simulation 5

1 Objectifs

Ce laboratoire exerce la conception de circuits logiques à l'aide de multiplexeurs. Ceux-ci seront utilisés dans un premier temps pour réaliser les fonctionnalités décrites dans une table de vérité puis dans un deuxième temps comme moyen de sélection.

Ce laboratoire présente une méthode de réalisation d'une unité arithmétique et logique de microprocesseur. Les unités logique et arithmétique seront réalisées lors d'une première séance de laboratoire alors que l'Arithmetic and Logical Unit (ALU) complète sera finalisée lors d'une deuxième séance.

2 Unité logique LU

La Fig. 1 présente le circuit d'une unité logique (Logical Unit (LU)) d'un microprocesseur. Les opérations logiques se font bit à bit.

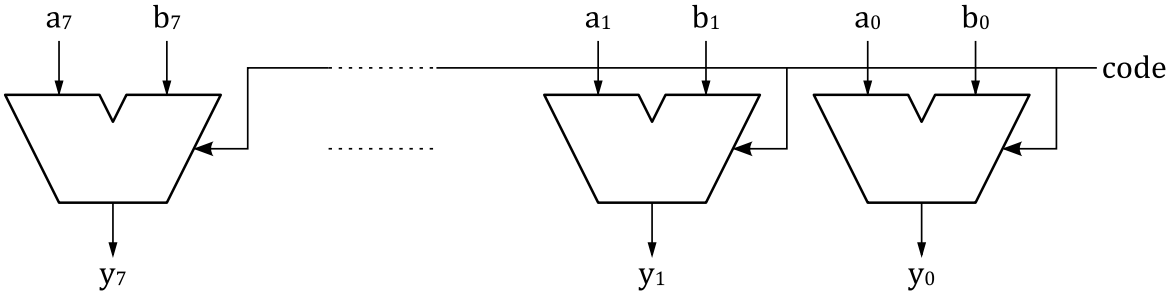


Fig. 1. – Unité logique

Les blocs itératifs de l'unité logique sont réalisés par un multiplexeur qui réalise une table de vérité. Les signaux de contrôle du multiplexeur sont : $\text{sel}_{0i} = a_i$, $\text{sel}_{1i} = b_i$ et $(\text{sel}_3, \text{sel}_2) = \text{code}[1:0]$ servant à sélectionner la fonction à effectuer.

Écrire la table de vérité de la fonction logique qui réalise les opérations suivantes.

- $y_i = b_i$ pour **`code = "00"`** ⇨ chargement de b
- $y_i = a_i * b_i$ pour **`code = "01"`** ⇨ fonction ET entre a et b
- $y_i = a_i + b_i$ pour **`code = "10"`** ⇨ fonction OU entre a et b
- $y_i = a_i \oplus b_i$ pour **`code = "11"`** ⇨ fonction XOR entre a et b

2.1 Réalisation et simulation

Compléter le circuit du bloc itératif de l'unité logique qui réalise les 4 opérations précitées.

Compléter les stimulis de test et vérifier le fonctionnement de l'unité logique complète.

3 Unité arithmétique AU

La Fig. 2 présente le circuit itératif d'une unité arithmétique (Arithmetic Unit (AU)).

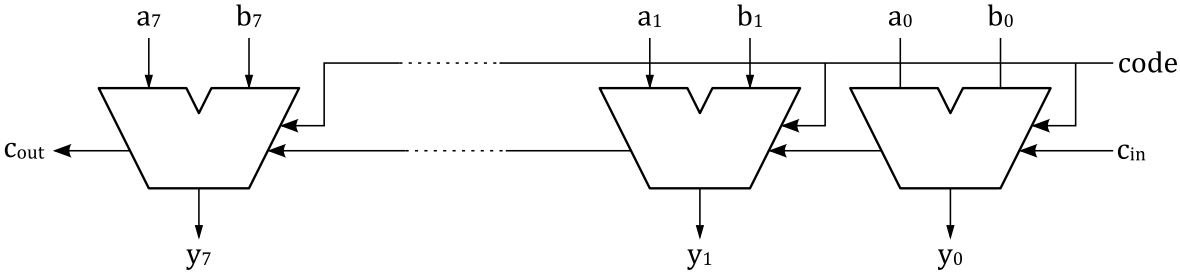


Fig. 2. – Unité arithmétique itérative

Écrire la table de vérité de la fonction logique d'un bloc itératif de la Figure Fig. 2 pour chacune des opérations suivantes sur des nombres entiers:

- $y = a + b$ pour **`code[0] = '0'`** ⇨ addition
- $y = a - b$ pour **`code[0] = '1'`** ⇨ soustraction

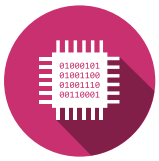
La fonction sera réalisée avec un multiplexeur dont les signaux de contrôle sont $\text{sel}_0 = a_i$, $\text{sel}_1 = b_i$,$\text{sel}_2 = c_{\text{in}}$ et $\text{sel}_3 = \text{code}[0]$.

En considérant que le décalage à gauche correspond à une multiplication par deux, proposer un complément au circuit itératif pour réaliser la fonction de décalage à gauche : $y = a \ll 1 = 2a = a + a$, pour **`code[1] = '1'`**.

3.1 Réalisation et simulation

Compléter le circuit du bloc itératif de l'unité arithmétique qui réalise les 3 opérations précitées.

Compléter les stimulis de test et vérifier le fonctionnement de l'unité arithmétique complète.



4 Unité arithmétique et logique

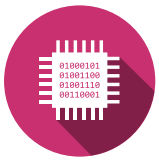
4.1 Sélection des résultats

L'unité arithmétique et logique (Arithmetic and Logical Unit (ALU)) est réalisée ici par la combinaison de l'unité logique et de l'unité arithmétique développées jusqu'ici. Elle comprend en outre une opération de décalage vers la droite.

Compléter le circuit de l'ALU pour générer les signaux de contrôle des multiplexeurs servant à sélectionner les opérations à effectuer. Ces opérations sont données dans la table suivante:

Instruction $I_{17} : I_{13}$	Mnemonic	ALU code Operation	Operation
00000	LOAD	LOAD B	$y = b$
00001	*ununsed*	-	-
00010	INPUT	LOAD B	$y = b$
00011	FETCH	LOAD B	$y = b$
00100	*ununsed*	-	-
00101	AND	AND	$y = a$ AND b
00110	OR	OR	$y = a$ OR b
00111	XOR	XOR	$y = a$ XOR b
01000	*ununsed*	-	-
01001	TEST	AND	$y = a$ AND b
01010	COMPARE	SUB	$y = a - b$
01011	*ununsed*	-	-
01100	ADD	ADD	$y = a + b$
01101	ADDCY	ADDCY	$y = a + b + c_{\text{in}}$
01110	SUB	SUB	$y = a - b$
01111	SUBCY	SUBCY	$y = a - b - c_{\text{in}}$
10000	SH / ROT	SHR	$a \gg 1$
10001	SH / ROT	SHL	$a \ll 1$
10010	*ununsed*	-	-
10011	*ununsed*	-	-
10100	*non-ALU*	-	-
...	...	...	...
11111	*non -LU*	-	-

Tableau 1. – Instructions effectuées par l'ALU



4.2 Réalisation et simulation

A partir de la Tableau 1, compléter la table de vérité Tableau 2 qui donne les signaux de commande des multiplexeurs de l'ALU ainsi que les commandes de l'unité logique et de l'unité arithmétique en fonction du code de l'ALU (code[4 : 0]).

code[4 : 0]	$LU_{code}[1:0]$	$AU_{code}[1:0]$	$select_{AU}$	$select_{SR}$	c_{in_AU}
00000					
00001					
00010					
00011					
00100					
00101					
00110					
00111					
01000					
01001					
01010					
01011					
01100					
01101					
01110					
01111					
10000					
10001					
10010					
10011					
10100					
...					
11111					

Tableau 2. – Signaux de commande de l'ALU

Établir les équations de ces signaux de contrôle.

Compléter le circuit de l'ALU et les stimulis de test puis vérifier le bon fonctionnement de l'ALU.