

ALM (Hard) : Travail pratique N° 2

Conception d'une Unité Arithmétique et Logique (UAL)

1 Compte-rendu(à rendre par binome)

- Il doit contenir une étude théorique de l'UAL (en particulier sur la réalisation de la soustraction à partir de l'additionneur).
- Un dessin du circuit à l'aide de portes logiques et d'un additionneur.
- La description du circuit en langage Lustre (**fortement commentée**).
- Les résultats des simulations d'un jeu de test choisis judicieusement et **fortement commentés**. Ces tests doivent permettre de se persuader que le circuit " fonctionne " correctement. On s'appliquera en particulier à vérifier la validité des 4 flags Z, N, C, V pour chaque opération suivant le choix du codage utilisé (base 2 ou complément à 2).

2 Introduction

Vous allez concevoir une Unité Arithmétique et Logique (UAL) n bits. Cette UAL ne fait pas de calcul Logique mais on pourrait très bien étendre son principe et rajouter un certain nombre d'opérations booléennes (ET, OU, XOR bit à bit...).

Elle servira par la suite pour la conception de circuits plus complexes, il est donc important d'arriver à une réalisation correcte.

La description de ce circuit sera faite dans le langage LUSTRE (voir la documentation jointe). Cette description LUSTRE pourrait servir de point de départ à un outil de CAO qui génère les dessins des masques des portes logiques nécessaires à la réalisation véritable d'un circuit.

3 Première étape : l'additionneur 1 bit

- Retrouvez les équations booléennes de l'addition binaire.
- Décrivez dans le langage LUSTRE un additionneur 1 bit.
- Compilez et simulez.

4 L'additionneur n bits

Décrivez en LUSTRE un additionneur n bits. Servez-vous de l'additionneur 1 bit réalisé. Fixez n à 4 bits et simulez.

5 L'UAL n bits

Il s'agit de concevoir une Unité Arithmétique et Logique réalisant les 4 fonctions **Add**, **Incr**, **Sub**, **Opposé** sur deux nombres A (a_i) et B (b_i) codés soit en base 2, soit en complément à 2 sur n bits. Le résultat T (t_i) est un nombre codé en base 2 ou en complément à 2 sur n bits. Le choix du codage est laissé à l'appréciation de l'utilisateur.

L'opération effectuée dépend de deux bits de commandes *uc1* et *uc0*. La règle de choix du calcul est donnée par la table de la figure 1.

| uc1 | uc0 | résultat | OPÉRATION EFFECTUÉE |
|-----|-----|-------------|-----------------------|
| 0 | 0 | $T = -A$ | OPPOSE |
| 0 | 1 | $T = A - B$ | SOUSTRACTION (Sub) |
| 1 | 0 | $T = B + 1$ | INCREMENTATION (Incr) |
| 1 | 1 | $T = A + B$ | ADDITION (Add) |

FIG. 1 – Opérations de l'Unité Arithmétique

L'UAL délivre aussi les bits de compte-rendu :

- N (N = 1 si le résultat est strictement négatif en complément à 2),
- Z (Z=1 si le résultat est nul)
- C (égale à la dernière retenue)
- V (V=1 si le résultat est faux en interprétant les opérandes et le résultat en complément à 2).

La figure 2 représente les entrées/sorties de l'UAL.

5.1 Indications pour la réalisation de l'UAL

On peut utiliser un seul additionneur. En effet on peut réaliser les différentes opérations de l'UA en modifiant les entrées de l'additionneur suivant l'opération à réaliser.

Réfléchir pour cela à la réalisation d'un soustracteur à l'aide d'un additionneur et d'inverseurs. Pour choisir les bonnes entrées de l'additionneur (fonctions booléennes de A , B , uc1 et uc0) on peut utiliser des multiplexeurs ou réaliser directement à l'aide de portes logiques les fonctions booléennes correspondantes.

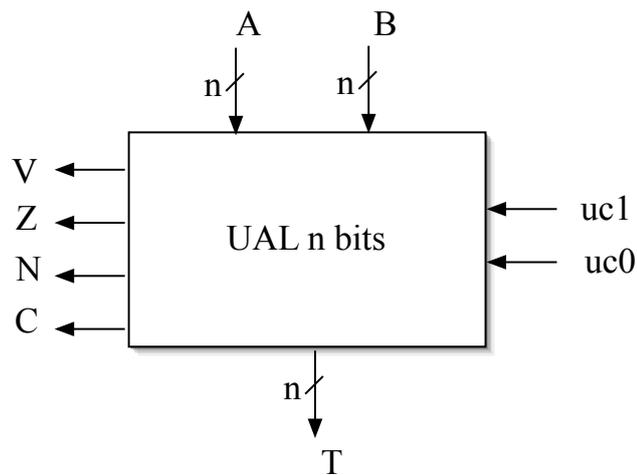


FIG. 2 – Entrées/sorties de l'UAL

Si l'on pose $X=(x_{n-1}, \dots, x_1, x_0)$ et $Y=(y_{n-1}, \dots, y_1, y_0)$ les opérandes de l'additionneur, x_i et y_i sont des fonctions booléennes de variables $a_i, b_i, uc1, uc0$. Ces circuits élémentaires sont identiques pour tous les bits.

Construire de même un circuit qui "fabrique" la retenue entrante v_0 de rang zéro de l'additionneur à partir de $uc1$ et $uc0$.

Assembler le tout de part et d'autre de l'additionneur (voir la figure 3).

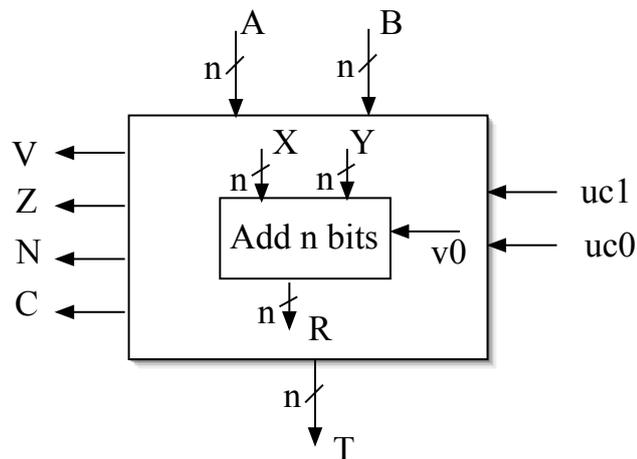


FIG. 3 – L'UAL construit à partir d'un additionneur

5.2 Tests

Réalisez un ensemble de test permettant de vérifier le fonctionnement de l'UAL pour chacune de ses 4 opérations. On vérifiera la validité des flags et on donnera l'interprétation de leurs valeurs dans les deux cas de codage : complément à 2 (signé) et base 2 (non signé).

5.3 Variante de réalisation

Construire une UAL 1 bit à partir de l'additionneur 1 bit. En déduire une UAL n bits.