

ARC1 - TP 4

Léo Noël-Baron & Thierry Sampaio

23/10/2015

Résumé

Ce TP propose l'élaboration d'une Unité Arithmétique et Logique (UAL) simpliste, permettant quatre opérations logiques ainsi que l'addition et la soustraction et traitant des données sur 4 bits. Cette UAL comportera trois opérandes (dont une commande sur 3 bits) et trois sorties (le résultat sur 4 bits et les retenues de l'addition et de la soustraction).

Opérations logiques

ET/ID On souhaite implémenter les opérations logiques suivantes sur deux opérandes X et Y : $X \wedge Y$, Y , $X \wedge \bar{Y}$ et \bar{Y} . Pour cela, on réalise tout d'abord un circuit ET/ID qui renvoie $A \wedge B$ ou B selon une commande \bar{et} . L'expression formelle correspondant à ce traitement est $S_i = B_i \wedge (A_i \vee \bar{et})$ (pour $i \in [0..3]$), ce qui donne le circuit en Figure 1.

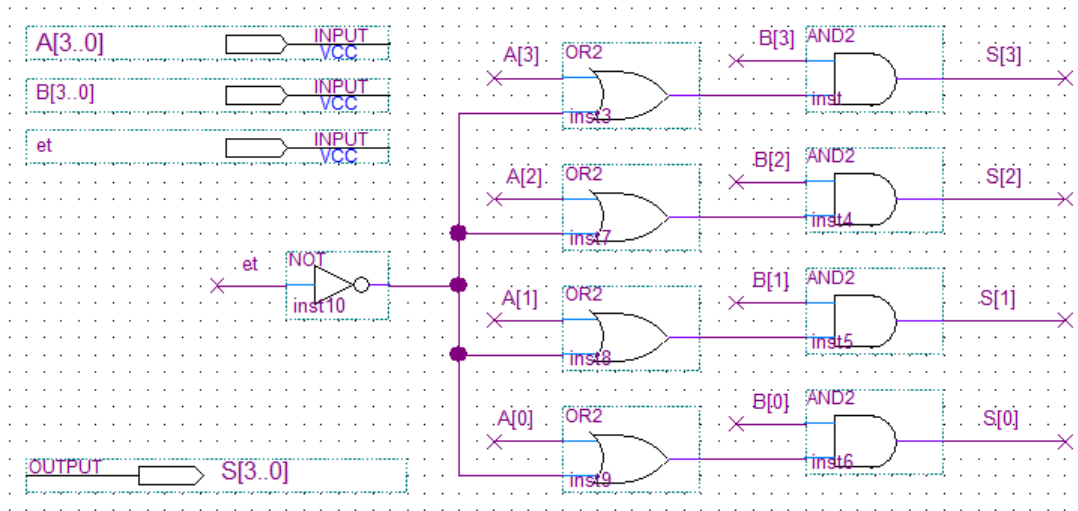


FIGURE 1 – Schéma du composant ET/ID

INVCOND On conçoit ensuite un inverseur conditionnel au moyen de portes XOR; l'expression formelle $B_i = A_i \oplus inv$ (inv étant la commande d'inversion) s'implémente comme en Figure 2.

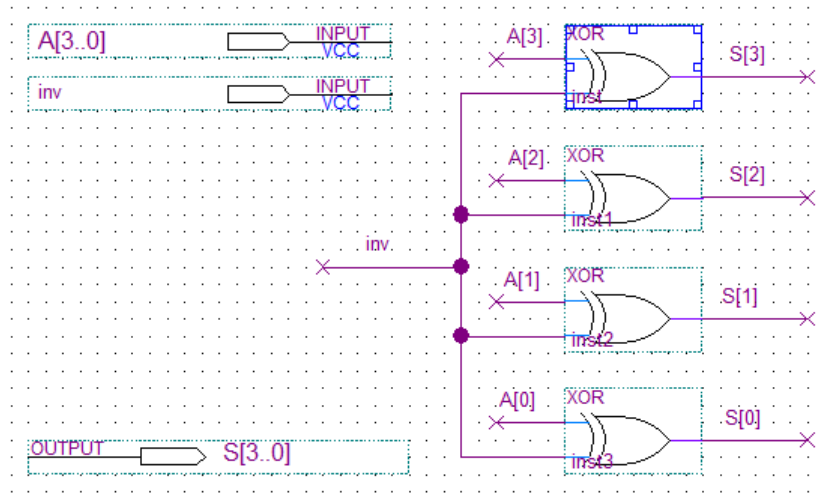


FIGURE 2 – Schéma du composant INVCOND

OPLOG Les deux circuits précédents permettent de construire le circuit d'opérations logiques de l'UAL. En effet, la table des commandes impose pour les deux bits de poids faible ce

| $L_1 L_0$ | 00 | 10 | 01 | 11 |
|------------|--------------|-----|-------------------------|----------------|
| Opérations | $X \wedge Y$ | Y | $X \wedge \overline{Y}$ | \overline{Y} |

qui permet directement de commander les circuits ET/ID et INVCOND avec respectivement les signaux L_1 et L_0 (Figure 3).

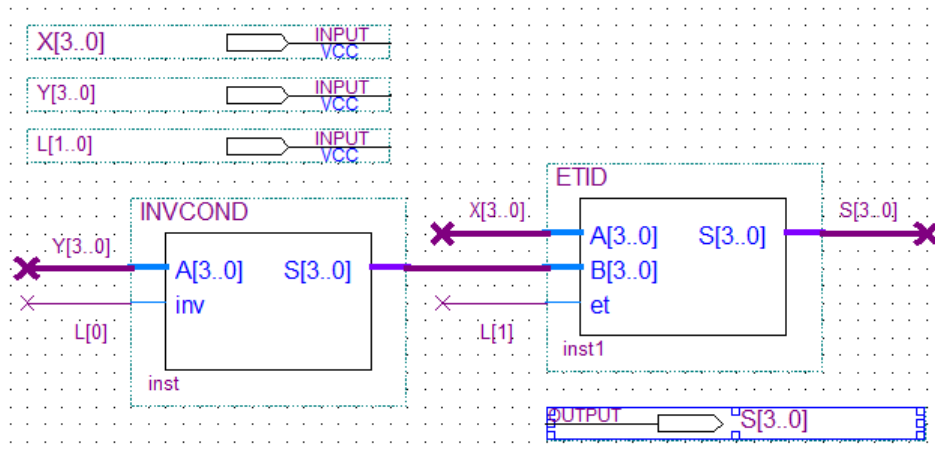


FIGURE 3 – Schéma du composant OPLOG

Opérations arithmétiques

Conception Le circuit arithmétique de l'UAL prend deux opérandes sur 4 bits X et Y , ainsi qu'une commande AS pour choisir l'opération. Il peut être réalisé au moyen des circuits

ADD4GROUP et INVCOND ; en effet, la notation en complément à deux se fait en inversant bit à bit une opérande binaire et en lui ajoutant 1. On inverse donc au besoin Y grâce au circuit INVCOND commandé par AS , et on redirige le signal dans ADD4GROUP où on réutilise AS en retenue entrante pour obtenir la notation en complément de Y .

La retenue sortante et le débordement nécessitent un traitement supplémentaire, différencié selon l'opération. L'addition ne peut donner qu'une retenue sortante, qui est celle de l'additionneur ADD4GROUP. La soustraction donne en retenue sortante l'inverse de celle d'ADD4GROUP, mais peut aussi provoquer un débordement qui est détectable si le résultat est du signe inverse des opérandes. On construit donc le circuit présenté en Figure 4.

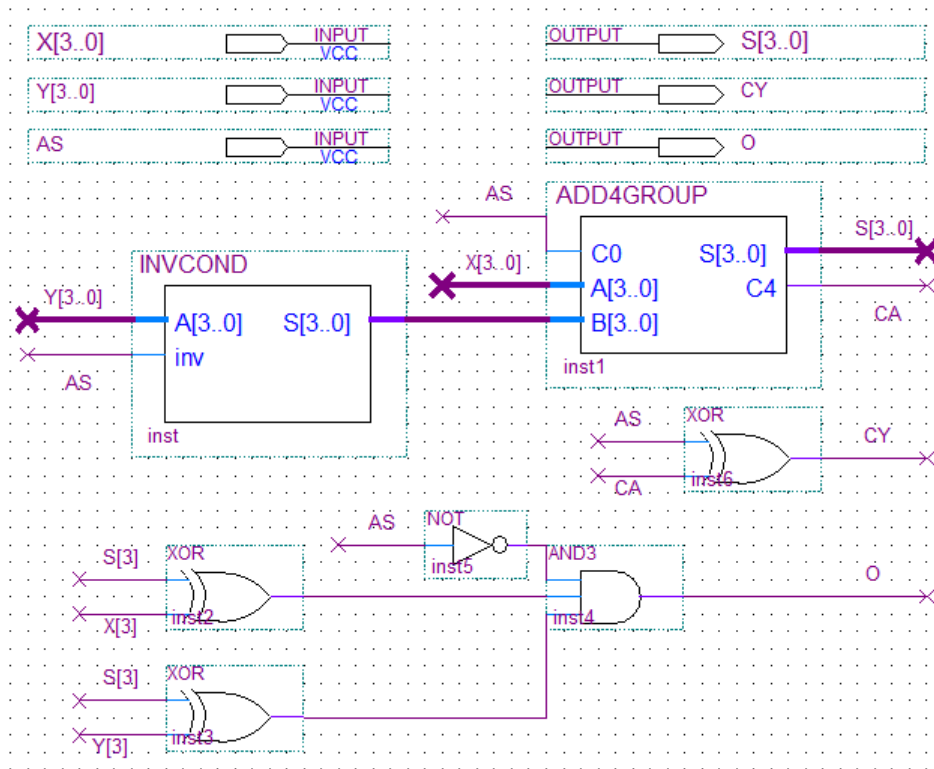


FIGURE 4 – Schéma du composant ADD/SOUS

Jeu de test Pour tester l'addition, on peut effectuer les opérations :

- 1001 + 0011 (9 + 3, pas de retenue sortante) et
- 0101 + 1101 (5 + 13, retenue sortante).
- 1110 - 0011 (14 - 3, pas de retenue mais débordement) ;
- 1010 - 0100 (10 - 4, pas de retenue ni de débordement) ;
- 0010 - 0101 (2 - 5, retenue mais pas de débordement) et
- 0011 - 1101 (3 - 13, débordement).

Assemblage de l'UAL

L'UAL se construit à partir des composants précédemment définis, en dirigeant les opérandes et les signaux de commandes adéquats aux entrées du circuit arithmétique et du circuit logique, puis en sélectionnant le résultat voulu avec un SEL4 ; ceci donne le circuit final présenté en Figure 5.

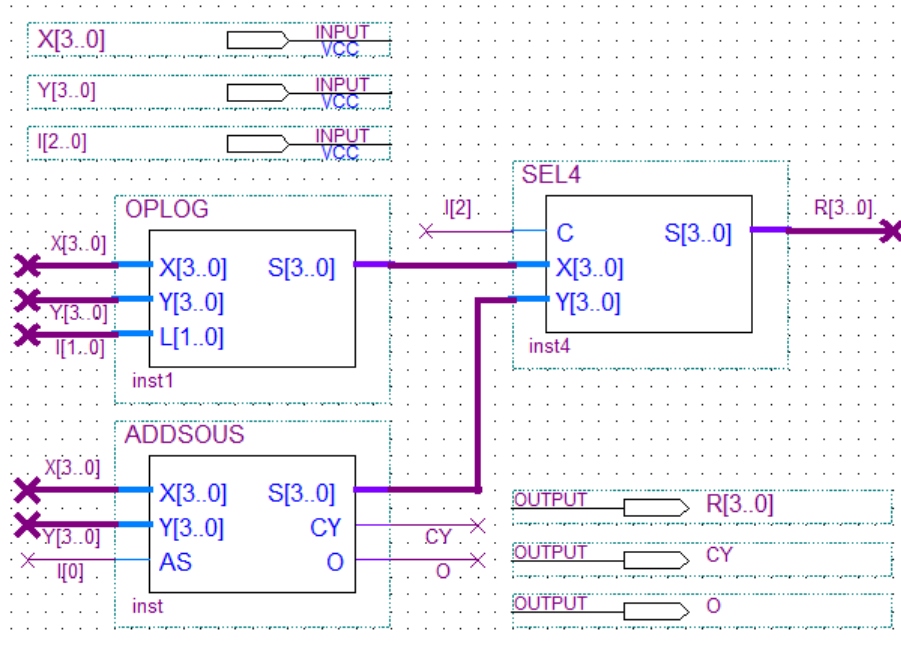


FIGURE 5 – Schéma de l'UAL