

TP 2 : Unité arithmétique et logique

Exercice 1 - Additionneur

- 1.1 [★] Définir un nouveau composant `add_1` réalisant un *full-adder*.
- 1.2 [★] En chaînant 8 composants `add_1`, définir un nouveau composant `add_8` réalisant l'addition de deux entiers codés en binaires sur 8 bits.

Exercice 2 - Soustracteur

- 2.1 [★] Définir un nouveau composant `alu_8` réalisant un additionneur/soustracteur de taille 8 bits. On utilisera un bit de contrôle qui, lorsqu'il est à 1, provoque le calcul de la soustraction au lieu de l'addition.
- 2.2 [★] Ajouter une sortie qui aura pour valeur 1 quand le résultat de l'opération est 0, et 0 sinon. Cette sortie correspondra donc au drapeau Z(ero).
- 2.3 Ajouter trois autres sorties correspondant aux drapeaux S(ign), C(arry) et O(verflow).

Exercice 3 - Unité arithmétique ... et logique aussi

Le but de cet exercice est d'étendre l'additionneur 8 bits réalisé précédemment. Vous aurez besoin uniquement du circuit `add_8`, que l'on utilisera toujours avec une retenue entrante `c0` valant 0 dans cet exercice.

En plus des deux entiers a et b codés chacun sur 8 bits, on utilisera en entrée les 5 bits de contrôle suivants :

- `zb` pour ne pas tenir de l'entrée b (remplacée par 0 quand `zb` = 1),
- `na` pour remplacer a par son complément à 1 quand `na` = 1,
- `nb` pour remplacer b par son complément à 1 quand `nb` = 1,
- `lg` pour choisir entre la sortie de `add_8` (`lg` = 0) et le résultat d'un `and` bit à bit entre a et b (`lg` = 1),
- `no` pour remplacer la sortie par son complément à 1 quand `no` = 1.

L'effet de `zb` s'appliquera avant l'effet de `nb`. Ainsi, le circuit réalisant l'éventuelle mise à 0 de b devra être placé avant celui qui se charge de l'éventuel complément à 1 de b

3.1 Pour chacune des opérations suivantes, déterminez les valeurs à utiliser pour les cinq bits de contrôle afin que l'ALU réalise l'opération demandée.

- | | | | | |
|-------------------|------------------|--------------|----------------|--------------|
| (i) $a + b$ | (iii) $a \vee b$ | (v) a | (vii) $-a$ | (ix) $b - a$ |
| (ii) $a \wedge b$ | (iv) $\neg a$ | (vi) $a - 1$ | (viii) $a + 1$ | (x) $a - b$ |

3.2 Réalisez le design proposé dans cet exercice dans `diglog`, et vérifiez vos réponses à la question précédente.