

Groupe : numéro 11

Auteurs : Piemontesi Gwendal, Trüeb Guillaume

Date : 23 novembre 2023

QUESTION 1 :

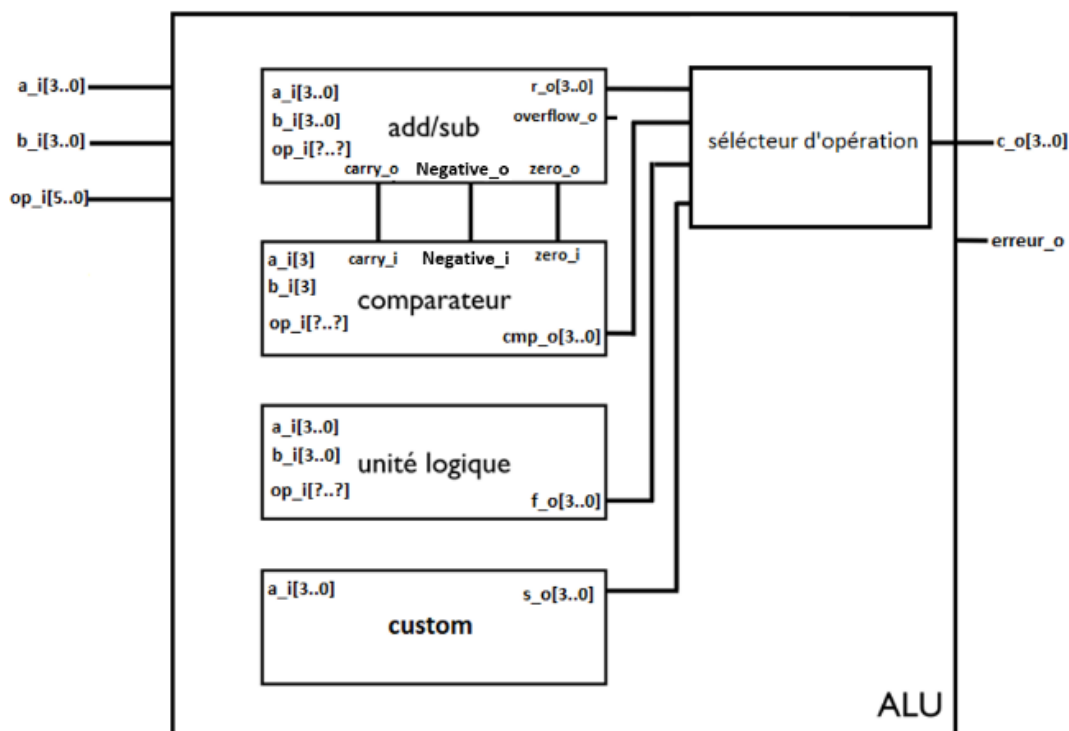
Identifiez et documentez les différents opcodes entraînant une erreur

Dans le bloc **Compareur** nous aurons des erreurs dans les cas où l'opcode vaut 011000, 011111 ou si le 4ème bit (depuis le LSB) vaut 0.

QUESTION 2

Reprenez le schéma disponible en Figure 1 et donnez, pour chaque bloc, les bits d'opcode utilisés (l'idée est de répondre aux points d'interrogation dans le schéma). Vos réponses doivent être de la forme "les bits d'opcode du bloc add/sub sont les bits [X:Y]".

Schéma bloc de l'ALU



Format d'opcode de 6 bits : (MSB→)φφφφφφ(←LSB)

- Bits d'opcode du bloc add/sub [3:2] → φφφφφφ
- Bits d'opcode du bloc compareur [2:0] → φφφφφφ
- Bits d'opcode du bloc logique [1:0] → φφφφφφ

Le bloc custom n'a pas besoin de faire un choix d'opération car il n'a qu'un choix possible.

QUESTION 3

Lorsqu'un overflow se produit, le signe n'est plus géré correctement et la sortie `negative_o` n'a donc pas la bonne valeur. Corrigez ce problème et expliquez votre raisonnement

Dans le cas ou nous avons un overflow et que l'addition ou soustraction sont signées nous devons contrôler le bit de carry car il devient le 5ème bit (depuis le LSB) du resultat qui indique le signe et nous permettra de savoir si la valeur était négative ou pas (débordement).

QUESTION 4

Mettez en place une table de vérité, puis justifiez le développement de votre circuit.

Table de veritéé du bloc custom :

3	2	1	0	S
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

La fonction custom permet de définir si la valeur en entrée est une puissance de deux. Pour que ce soit le cas, il faut qu'un seul bit soit à 1 et tous les autres à 0.

QUESTION 5

En plus des opcodes invalides, certains cas d'utilisation doivent générer une erreur. Déterminez ces différents cas et documentez les avant d'implémenter les modifications nécessaires.

Des incohérences peuvent survenir lors de l'utilisation du bloc comparaison.

En effet celui-ci utilise les résultats du bloc add/sub, il arrive dans certain cas qu'une comparaison effectuée avec des valeurs totalement correctes que nous voulons effectuer produise un overflow dans le bloc add/sub.

Conclusion, lors de l'utilisation du bloc comparaison ne pas prendre en compte l'overflow du bloc add/sub qui devient incohérent.

QUESTION 6

Avec $a_i[3:0] = 0001$, $b_i[3:0] = 1001$ et $op_i[6:0] = 011011$ quelle est la valeur observée pour l'overflow ? Et pour la sortie erreur_o? Cela correspond-t-il à ce que vous attendiez à la suite de la Question 5 ?

Il y a un overflow et une erreur.

L'overflow peut s'expliquer car la comparaison utilise les résultats du bloc add/sub. Dans ce cas le bloc add/sub effectue l'opération $A - B$ signée ce qui donne $0001 - 1001$ et ceci vaut $1\ 1000$ il a donc overflow et une erreur mais ceci est incohérent car nous utilisons le bloc comparaison qui ne doit pas prendre en compte toute erreur générée dans le bloc add/sub.

Cela correspond donc à nos attentes du point 5.