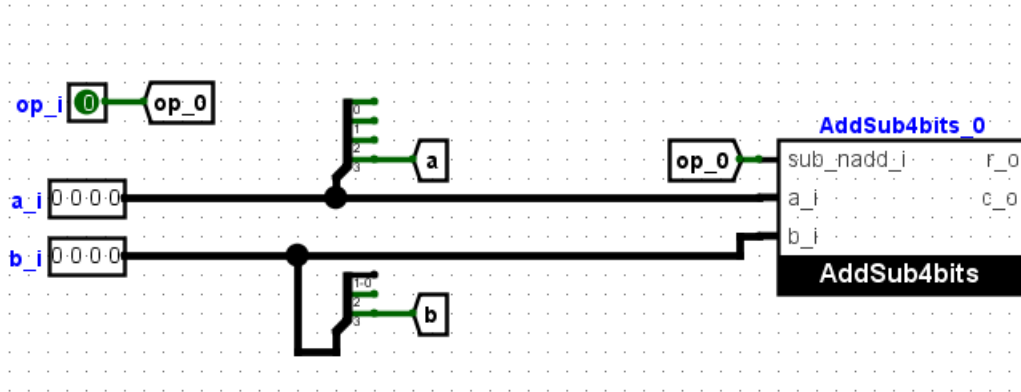


## Question 1

- Les bits d'opcode du bloc Add/Sub sont les bits [0:0]

### Justification :

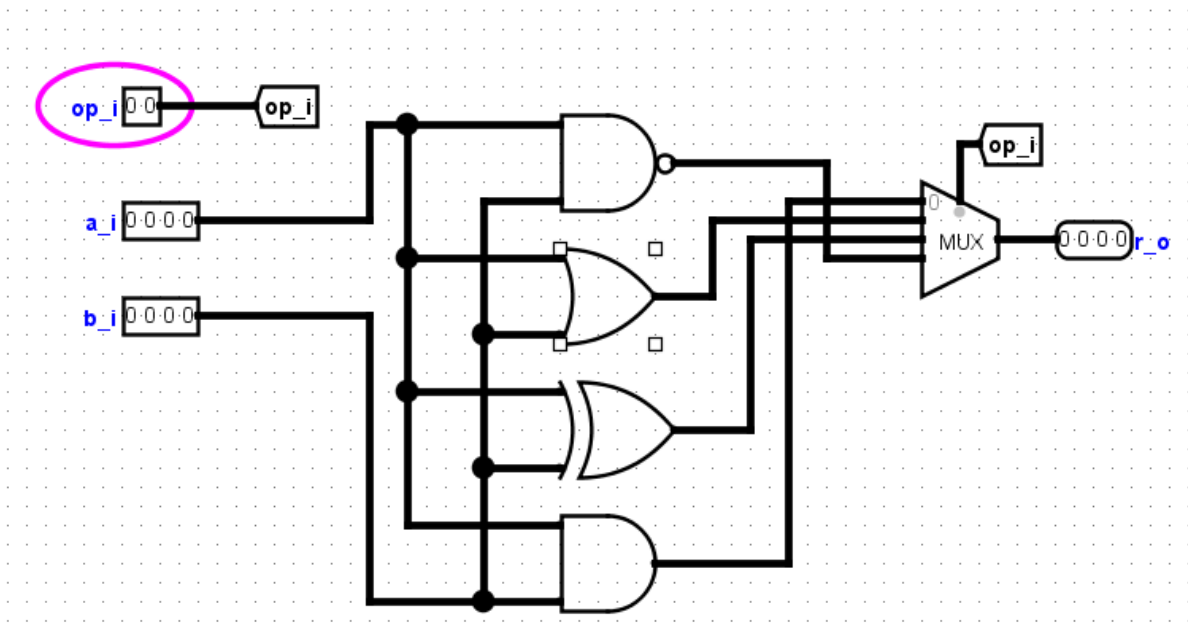
- [0:0] indique que on utilise un seul bit (le bit 0) pour sélectionner l'opération d'addition (0) ou de soustraction (1). Avec un bit on peut représenter 2 opérations différentes.



- Les bits d'opcode du bloc Logique sont les bits [1:0]

### Justification :

Avec 2 bits, on peut représenter jusqu'à 4 opérations différentes.



## Question 2

### Détection de l'overflow

- L'overflow se produit lorsque le résultat d'une addition ou d'une soustraction dépasse la capacité de représentation du type de données (dans notre cas pour une ALU 4 bits, cela signifie que le résultat est supérieur à 7 ou inférieur à -8 pour des nombres signés). Pour détecter l'overflow dans une addition, on vérifie si les signes des deux opérandes sont les mêmes et si le signe du résultat est différent. Pour une soustraction, vous pouvez utiliser une méthode similaire.

### Ajustement de la sortie negative\_o

- Si un overflow est détecté, la sortie negative\_o doit être ajustée pour refléter correctement le signe du résultat.

## Question 3

Opération	Fonction logique
$A \geq B$ (signé)	/n
$A < B$ (signé)	n
$A \neq B$	/z
$A = B$	z
$A \geq B$ (non-signé)	/c
$A < B$ (non-signé)	c

## Question 4

### Opérations valides avec 4 bits d'opcode

- $A \geq B$  (signé) : 0001
- $A < B$  (signé) : 1001
- $A \neq B$  : 0001
- $A = B$  : 0011
- $A \geq B$  (non signé) : 1011
- $A < B$  (non signé) : 0011

## Table de vérité

op_i	err_o
0000	1
0001	0
0010	1
0011	0
0100	1
0101	1
0110	1
0111	1
1000	1
1001	0
1010	1
1011	0
1100	1
1101	1
1110	1
1111	1

Pour simplifier l'équation, nous pouvons observer que err\_o est activé (1) pour toutes les valeurs de op\_i sauf pour les valeurs valides. Les valeurs valides sont :

- 0001 ( $A \geq B$  signé)
- 0011 ( $A = B$ )
- 1001 ( $A < B$  signé)
- 1011 ( $A \geq B$  non signé)

### Équation simplifiée

$$\text{err\_o} = \neg(0001) \cdot \neg(0011) \cdot \neg(1001) \cdot \neg(1011)$$

## Question 5

0011

### Question 6

