Nom : MOHAMMEDI

Prénom : Amira

M1 : HPC

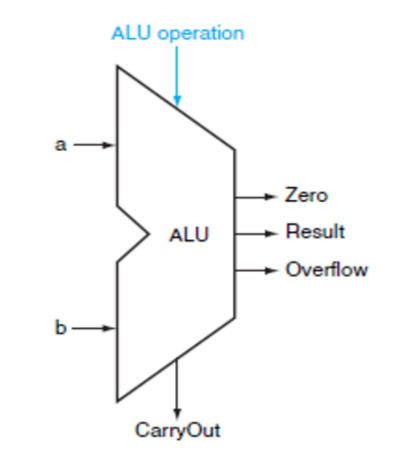
PROJET Architectures Avancées : ALU +REGISTRE FILE

M . BABA-ALI

2022

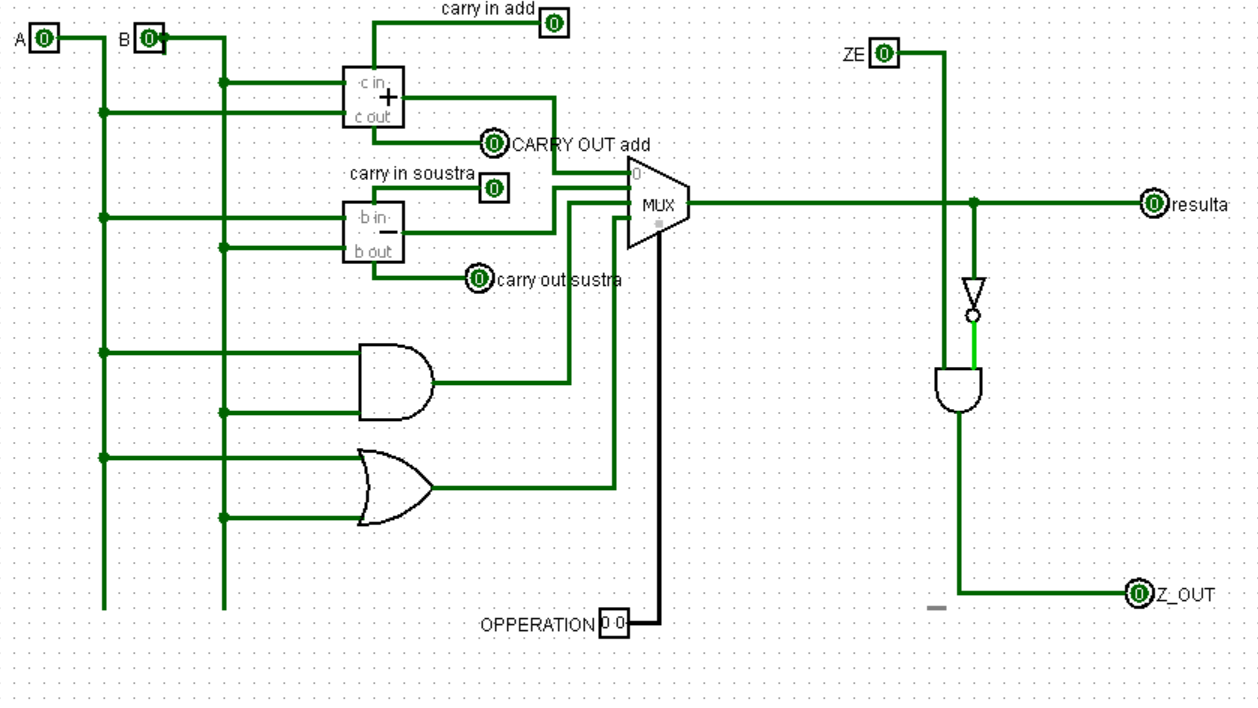
**La conception  et la réalisation d'une UAL(16 bits) connectée a un Registre File (8 registres de 16 bits) :**

* **UAL(16 bits) :**

****

* Pour notre UAL, nous allons concevoir une UAL qui sera capable de faire Arithmetic Op – (Addition, Subtraction), Logical Op – (and, or).
* Comme notre UAL fonctionnera avec des données de 16 bits, nous devrions logiquement construire un UAL de 1 bit qui effectuera nos opérations nécessaires.

**Conception UAL 1 bits :**

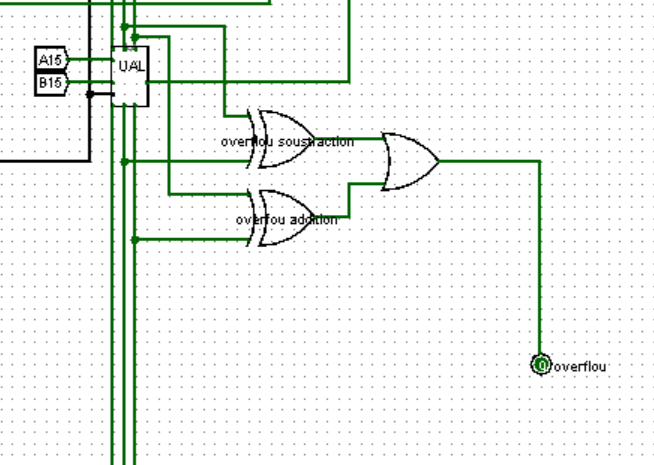


**Bits de contrôle ALU :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Opération | Sélectionnez le bit | CARRY IN ADD | CARRY IN SOUSTRACTEUR | ZE |
| ADDITION | 00 | 0 | 0 | 0 |
| SOUSTRACTION | 01 | 0 | 0 | 0 |
| ET | 10 | 0 | 0 | 0 |
| OU | 11 | 0 | 0 | 0 |

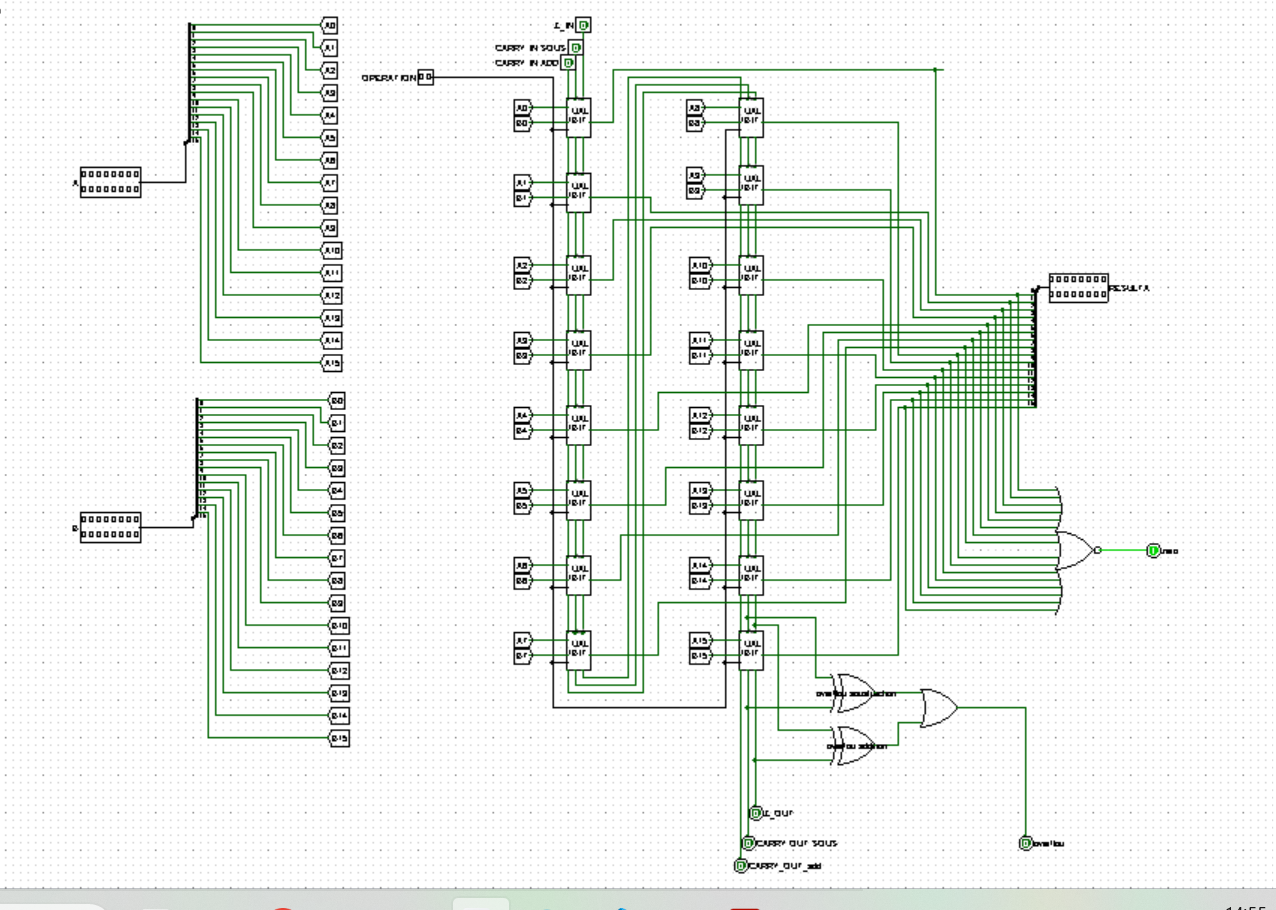
Overflow :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Entrée |  | Sortie |  |  |  |
| A | B | C\_in | C\_out | addition | soustraction | Overflow |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 |

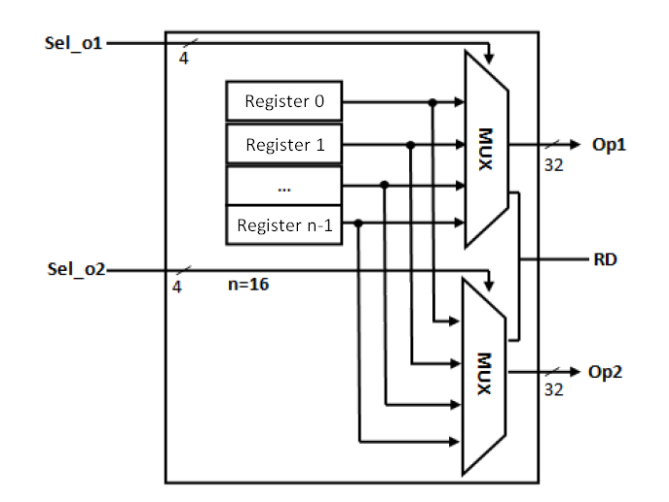
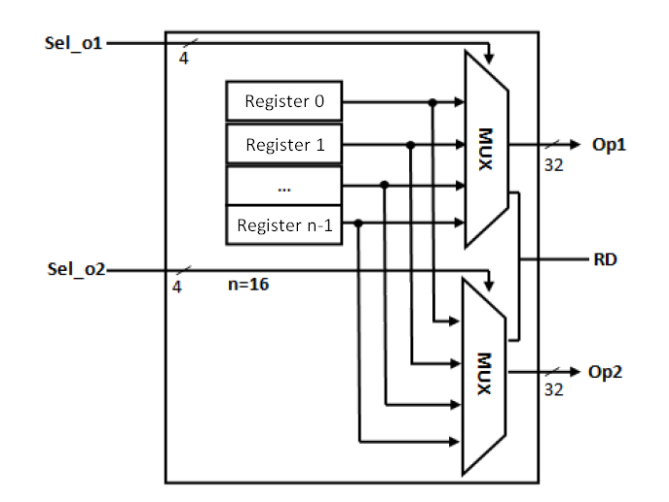


Maintenant, nous avons une ALU 1 bits entièrement fonctionnelle. Nous devons combiner 16 unités de cette ALU 1 bits pour construire une ALU 16 bits selon nos besoins. Le circuit ci-dessous est une ALU 16 bits construite avec 16 sous-circuits ALU 1 bits

**ALU 16 bits :**

****

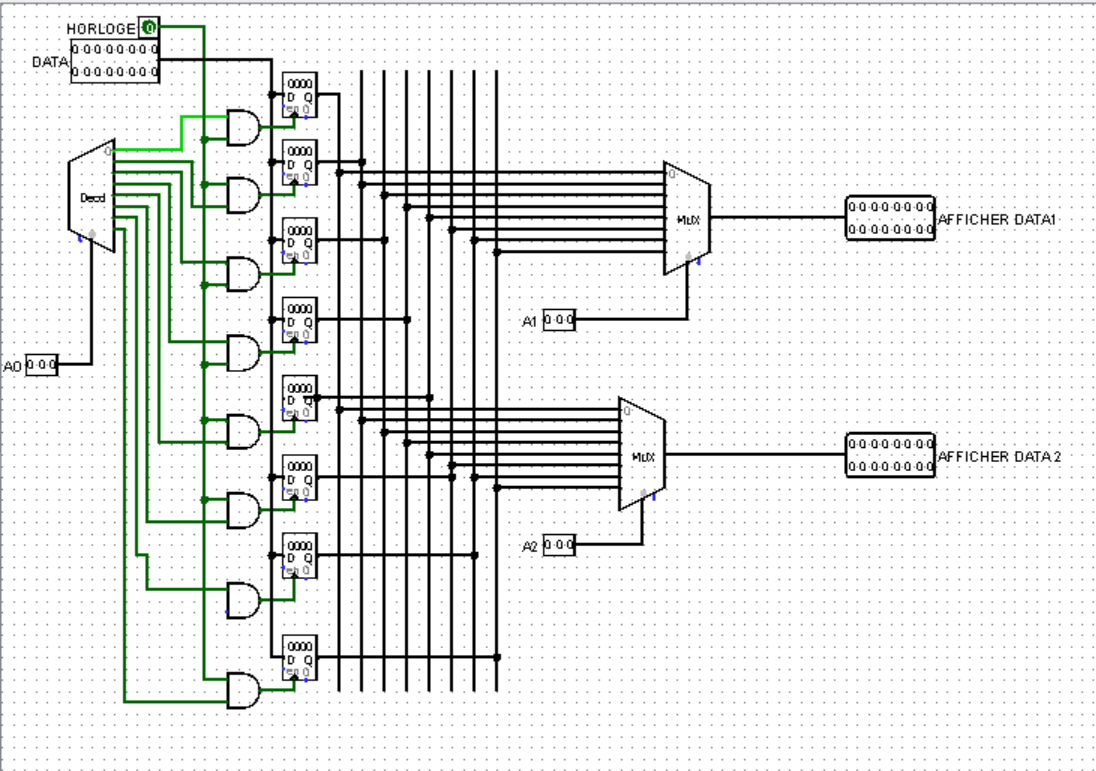
* **Registre File (8 registres de 16 bits) :**

****

**DEFINITION :** Register File est un espace mémoire présent dans le CPU . Il est utilisé par le CPU pour récupérer et conserver les données des périphériques de mémoire secondaires . Il est plus rapide par rapport aux autres dispositifs de mémoire car il est présent dans le processeur

* Pour notre registre file , On a utilisé 8 registre de front montant **,** nous pouvons dire que Q recopie D lorsque H devient égale à 1 ,Et 2 MUX de 3\*8
* Data est les données entrantes. Il est visible pour tous les registres . A0 détermine quel registre peut être écrit. Il est d'abord décodé par le décodeur. La sortie du décodeur sera ET avec l'horloge, puis connectée au port clk de chaque registre. En procédant ainsi, seul le registre actif (registre de destination) peut être écrit. Les autres registres sont désactivés (parce que la sortie du décodeur pour ce registre est faible).
* A1 et A2 contrôlent deux multiplexeurs pour sélectionner le registre à lire. Il y a deux multiplexeurs car nous voulons lire deux valeurs à la fois en tant qu'opérandes afin d'effectuer des opérations d'infixe.

Remarque : chaque registre peut stocker des données de 16 Bits

****

* **Conception :**  le fichier de registre et l'UAL sont tout ce qui est requis. L'UAL accepte son entrée des ports A et B du fichier de registre , et le fichier de registre est écrit par la sortie résulta de l'UAL, en combinaison avec le signal HORLOGE