BAMAS Mathias

BENLAKHLEF Mehdi

BIHI Ayoub

CHOPARD William

GROUPE 2

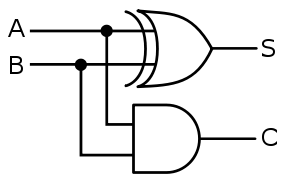
**SAE : Calculatrice, Conception d’un additionneur/soustracteur de 4 chiffres.**

**1ERE ETAPE : LE DEMI ADDITIONEUR :**

Conception d’un demi additionneur puis d’un additionneur :

On a déduit la conception du demi-additionneur et de l’additionneur grâce aux opérations booléennes de base, en commençant sur 1 bit. Par exemple 1 + 1 on doit renvoyer 0 et mettre une retenue, 1 + 0 on revoit 1 sans retenue etc…

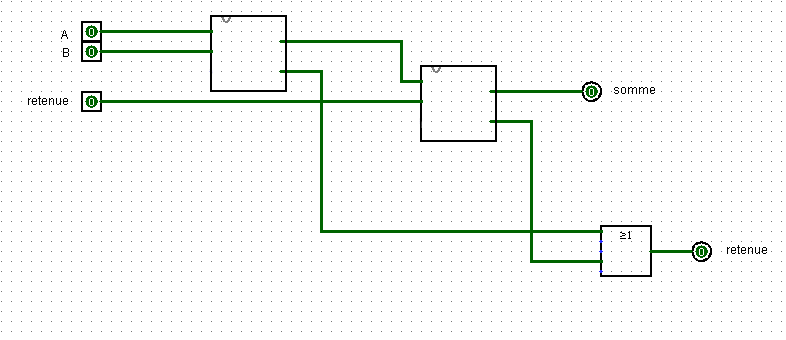
Demi-additionneur :



On en a ainsi déduit la table de vérité du demi-additionneur

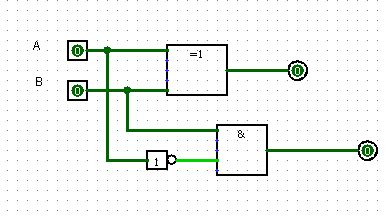
|  |  |  |  |
| --- | --- | --- | --- |
| A | B | R | S |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

**Additionneur sur Logisim :**



Voici notre additionneur qui prend en compte une retenue donc si par exemple le chiffre 1 est égal à 1 et que la retenue aussi, cela renverra 0 avec une retenue, etc…

A partir de là, on s’est dit qu’on allait faire la même chose avec la soustraction, on a donc fait un demi-soustracteur a part sur 1 bit. Cela a donné :



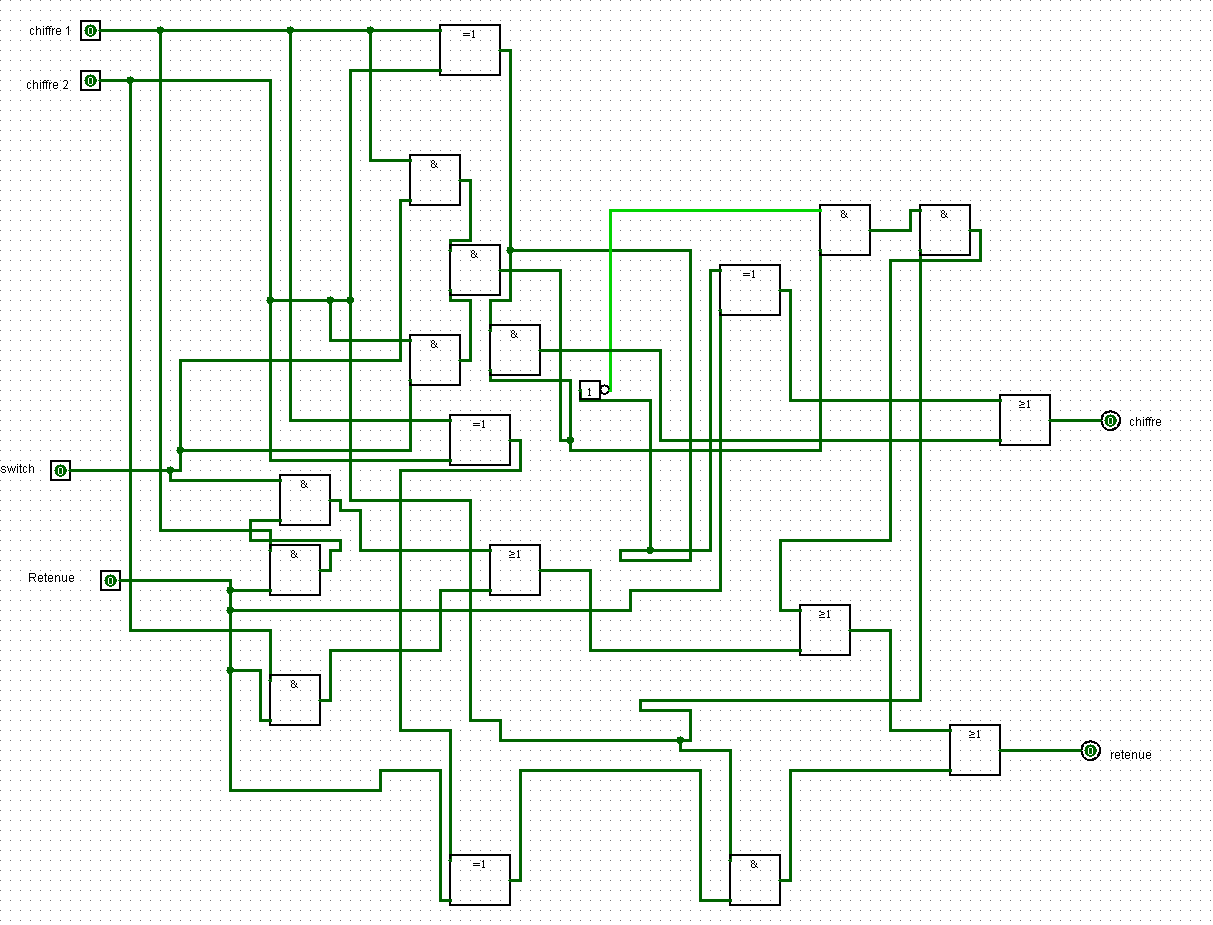
Notre demi-soustracteur respectait donc la table de vérité

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | D | C |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

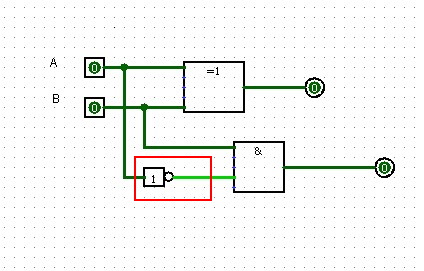
Pour obtenir un additionneur et un soustracteur séparés on a donc relié deux demi-additionneur/soustracteur ensembles avec un bloc OR pour obtenir une somme et une retenue.

**2E ETAPE : CHANGEMENT ADDITION/SOUSTRACTION**

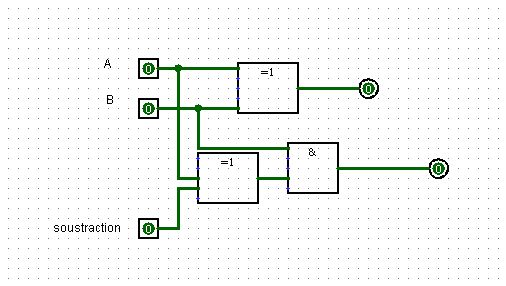
Après avoir créé un additionneur on a voulu combiner l’additionneur et le soustracteur ensemble avec un bouton qui permet de switch entre l’addition et la soustraction. On donc rajouté un bouton switch en entrée sur le schéma de base de notre soustraction et quand on appuie dessus on a fait en sorte que le résultat change et affiche celui de l’addition. Notre premier schéma était :



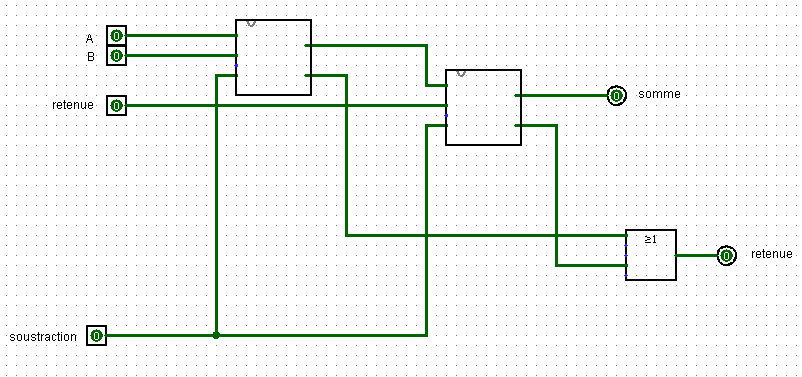
On a rencontré un problème sur ce schéma donc on a comparé les différences entre l’additionneur et le soustracteur avant de les combinés. On a ainsi remarqué que la seule différence entre un demi soustracteur et un demi-additionneur était que le premier bit était inversé dans cette partie :



Nous sommes partis d’un additionneur et dans le demi-additionneur/soustracteur nous avons ajouté un bloc XOR afin d’inverser le courant quand une entrée « soustracteur » est activée. On en est donc arrivé à ces schémas pour le demi-additionneur/soustracteur et l’additionneur/soustracteur.



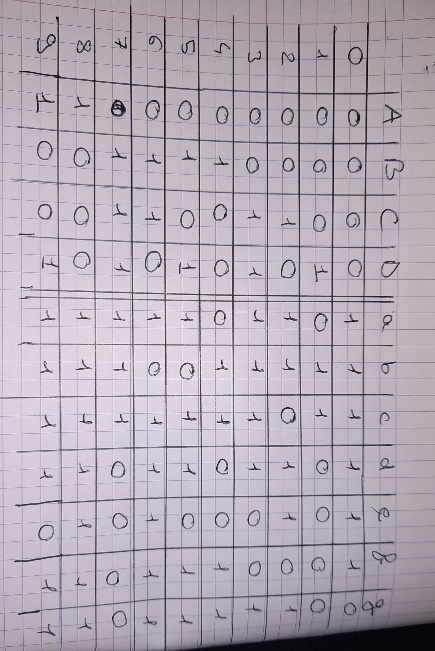
**DEMI-ADDITIONEUR/SOUSTRATEUR**



**ADDITIONEUR/SOUSTRATEUR**

**3E ETAPE : L’AFFICHAGE**

Pour savoir quel segment allumé pour quelle valeur on a dressé dans un premier temps un tableau pour les segments indiquant leur affichage ou nom selon les valeurs des 4 bits en entrée.

****

Avec ce tableau on a ensuite dressé une table de Karnaugh pour chaque segment afin d’obtenir les conditions d’affichage de chaque segment en fonction des 4 bits en entrée

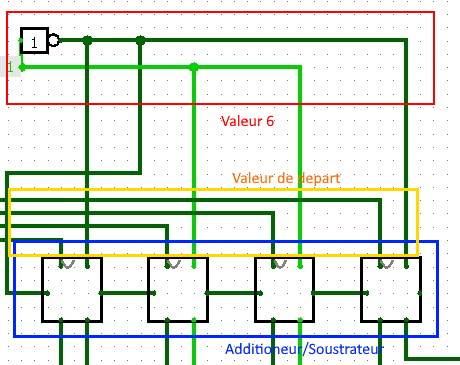
Pour *c* par exemple on obtient la condition suivante : « A + B̅ + C »

En combinant les conditions de chaque segment on obtient le circuit pour l’affichage des nombres.

**4E PARTIE : LA RETENUE DECIMALE**

Pour afficher la bonne valeur lorsque le résultat est supérieur à 9 on doit prendre en compte en reporté une retenue vers le poids supérieur.

On vérifie d’abord si le résultat est supérieur à 9 et si c’est le cas on lui ajoute 6 en cas d’addition ou -6 en cas de soustraction. On réalise cela avec 4 additionneur/soustracteur. Cela nous permet de renvoyer les retenues et d’obtenir le bon chiffre a afficher.



**5E ETAPE L’ASSEMBLAGE FINAL**

Maintenant que nous avons toutes les parties du circuit principal il nous suffit de les assemblée On relie chaque chiffres qui composent les nombres a des *input* qui permettent de choisir une valeur en binaire (tout en affichant « E » si le chiffre choisi est supérieur a 9.

On relie le 1er bit du chiffre 1 au 1er bit du chiffre 2 dans l’additionneur/soustracteur et ainsi de suite tout en reportant la retenue de chaque additionneur/soustracteur au suivant.

On relie la retenue du dernier additionneur/soustracteur au premier additionneur/soustracteur de poids supérieur.

Le résultat de chaque chiffre passe par le circuit de retenue décimale pour enfin être affiché dans le résultat. Cependant la calculatrice ne supporte malheureusement pas les résultats négatifs.

