Addition Soustraction BCD

Abdeslam MOKRANI
LIN Groupe 2

L'objectif de ce projet est la réalisation d'un circuit permettant d'effectuer les deux opérations, addition et soustraction, sur des nombres codés en BCD. Dans le circuit final, on pourra entrer les digits des deux opérandes, ainsi que l'opération codée sur quatre bits, séquentiellement en utilisant un bus de quatre bits. L'opération est ensuite lancée, le résultat et éventuellement une erreur sont affichés

On se limitera sur des nombres positifs ayant le même nombre de digits, un calcul sur deux digits sera présenté. La soustraction de deux nombres générera une erreur si le résultat n'est pas positif.

Le travail réparti en deux étapes. Dans la première étape on réalise un composant combinatoire permettant d'effectuer l'opération, les deux opérandes et l'opération sont alors entrés en parallèle. La deuxième étape consiste à combiner le composant précédant avec un circuit séquentiel, qui permet de saisir les données nécessaires en série, et de les diriger convenablement vers les entrées parallèles du premier composant.

Tous les circuits élémentaires ainsi que le circuit final sont réalisés à l'aide du logiciel DigLog, et sauvegardés dans un seul fichier (projet.lgf). Des circuits de tests seront réaliser pour vérifier au fur et à mesure le bon fonctionnement des composants (fichier tests.lgf).

1. Première étape

L'addition ou la soustraction de deux nombres décimaux s'effectue en additionnant ou soustrayant respectivement deux à deux leurs digits de même poids en partant du poids le plus faible. Si une retenue est générée alors elle sera prise en compte dans le calcul des deux digits suivants s'ils existent, affichée sinon. Aucune retenue dans le calcul initial (sur les digits de poids le plus faible).

Etant donné que le système électronique utilisé est binaire, chaque digit est alors représenté par un code binaire, le code BCD. Le calcul sur deux digits est alors basé sur des opérations binaires sur leurs codes. Or le code des dix digits possible (0,..,9) corresponds exactement à leurs valeurs en binaire, donc additionner ou soustraire deux digits où le résultat est représenter par un seul digit (sans retenu générée) revient à additionner ou soustraire leurs deux codes. Il ne restera alors que de résoudre le cas ou une retenue est générée.

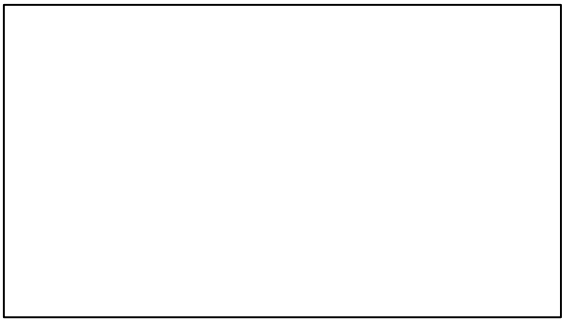
On va d'abord aborder l'addition de deux digits pour terminer par réalisez un composant qui calcul le code BCD du digit résultat et éventuellement une retenue. Le même travail sera réalisé pour la soustraction.

1.1. Addition BCD

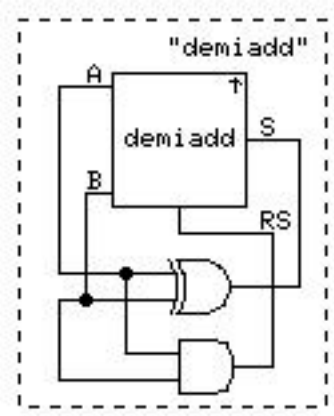
1.1.1. Additionneur binaire à un bit

Demi-Additionneur

L'addition de deux bits A et B sans retenue entrante (demi-additionneur) est résumée par la table de Karnaugh suivante, où S est le bit résultat et RS la retenue sortante.



Voici le composant correspondant au demi-additionneur réalisé avec DigLog.

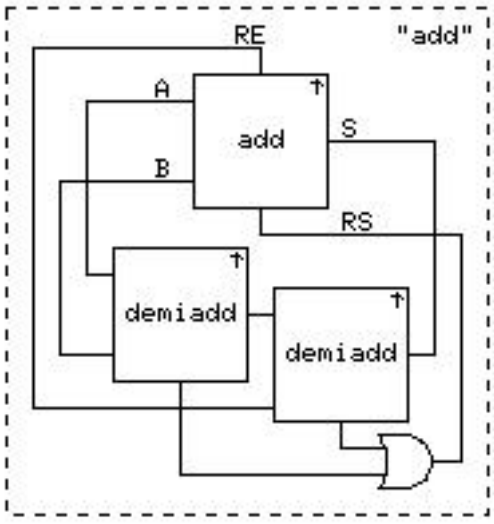


Demi-additionneur à deux bits

Additionneur complet (avec retenue entrante)

Si une retenue entrante RE existe, alors on doit l'additionner au résultat S de l'addition des deux bits A et B en utilisant le demi-additionneur précédent. La retenue sortante est alors la somme de la nouvelle retenue RS' (addition de RE et S) et l'ancienne RS (addition de A et B). Or si l'ancienne retenue RS est à 1, alors la somme S = A + B est à 0 (d'après la table Karnaugh précédente), soit la nouvelle retenue RS' (de RE + S = RE + 0) à 0. Autrement dit, les deux retenues RS et RS' ne peuvent être à 1 en même temps, donc la retenue de l'additionneur complet est simplement égale à RS+RS' (ou logique).

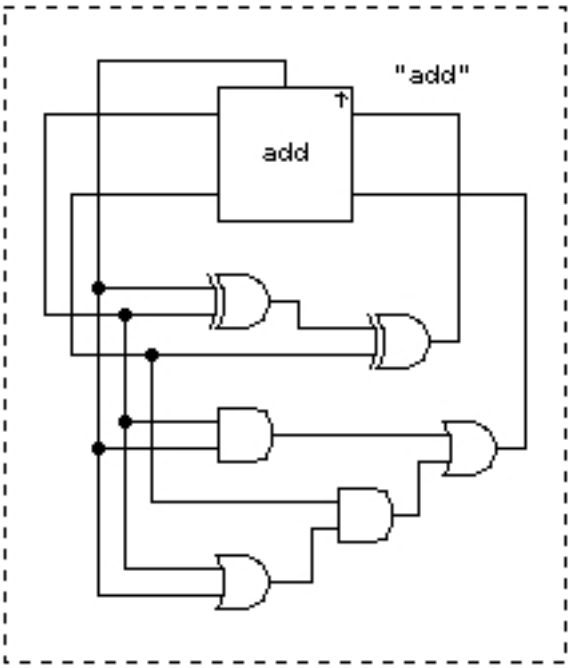
Voici le schéma du circuit de ce composant.



Additionneur à un bit

Autre conception.

En partant directement du cas où une retenue entrante existe, la table de Karnaugh mène au circuit suivant.

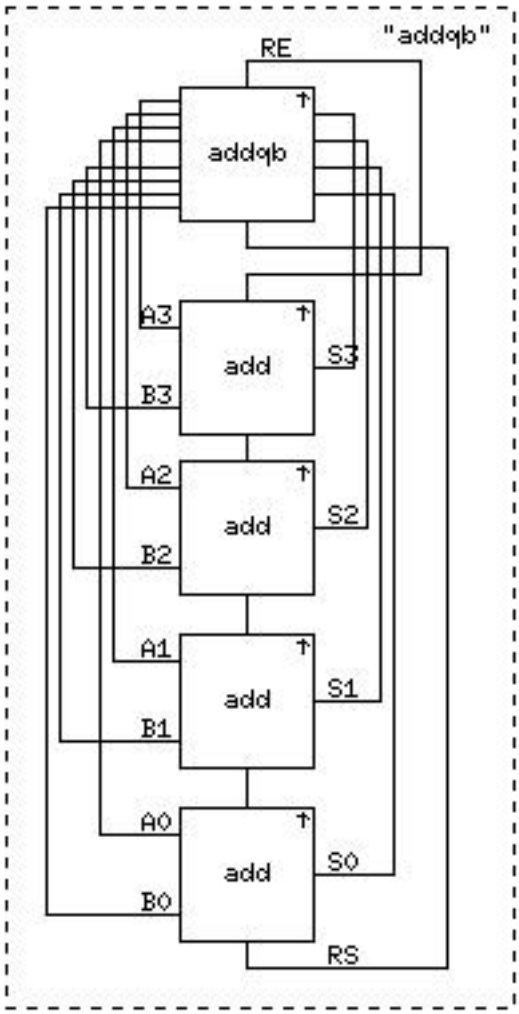


On remarque que cette nouvelle version utilise plus de portes logique que l'ancienne, on choisit donc l'ancienne version.

1.1.2. Additionneur quatre bits

Un digit est codé sur 4 bits, on doit donc réaliser un module permettant d'additionner deux nombres binaires codés sur 4 bits. Pour cela, il suffit d'utiliser 4 additionneurs à un bits, pour additionner les quatre pairs de bits de même poids en s'assurant du passage de la retenue. Une éventuelle retenue entrante est prévue ainsi qu'une retenue de sotie.

Voici le schéma d'un tel composant réalisé avec DigLog.



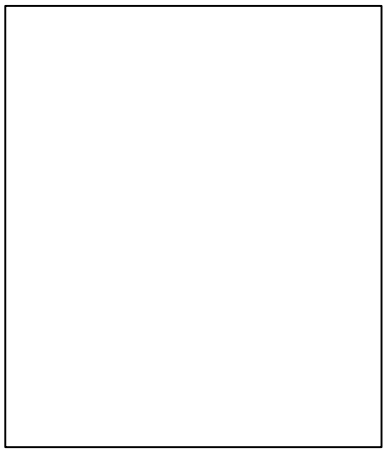
Additionneur à quatre bits

1.1.3. Additionneur BCD à un digit

soit deux digits D1 et D2, si en additionnant D1 et D2 on obtient un digit DS sans retenue (DS<10), le code BCD de DS est le résultat d'addition binaire à quatre bits des codes BCD de D1 et D2. Dans ce cas on utilise l'additionneur réalisé précédemment (Ceci correspond à 50% des cas). Sinon, l'addition binaire renvoie un résultat S ne correspondant pas à un digit ($10 \leq S < 16$) ou bien un résultat S avec une retenue si $S \geq 16$. La valeur de S ne correspond donc pas au code BCD du digit résultat, on dit qu'il y a eu un dépassement. Il faut donc réaliser une conversion vers le bon code.

Détection du dépassement

Voici la table de Karnaugh pour détecter le cas où $10 \leq S < 16$ avec A, B, C et D respectivement les bits du poids le plus fort au plus faible de S.



Si RS est la retenue sortante de l'addition, alors il y a dépassement si RE ou Dep1, soit RE+AB+AD.

Correction

Si un dépassement a eu lieu, alors on doit décaler le résultat de six pour omettre les six valeurs ne correspondant pas à des codes BCD (1010_b .. 1111_b). Il suffit donc d'ajouter la valeur 6 (0110_b).

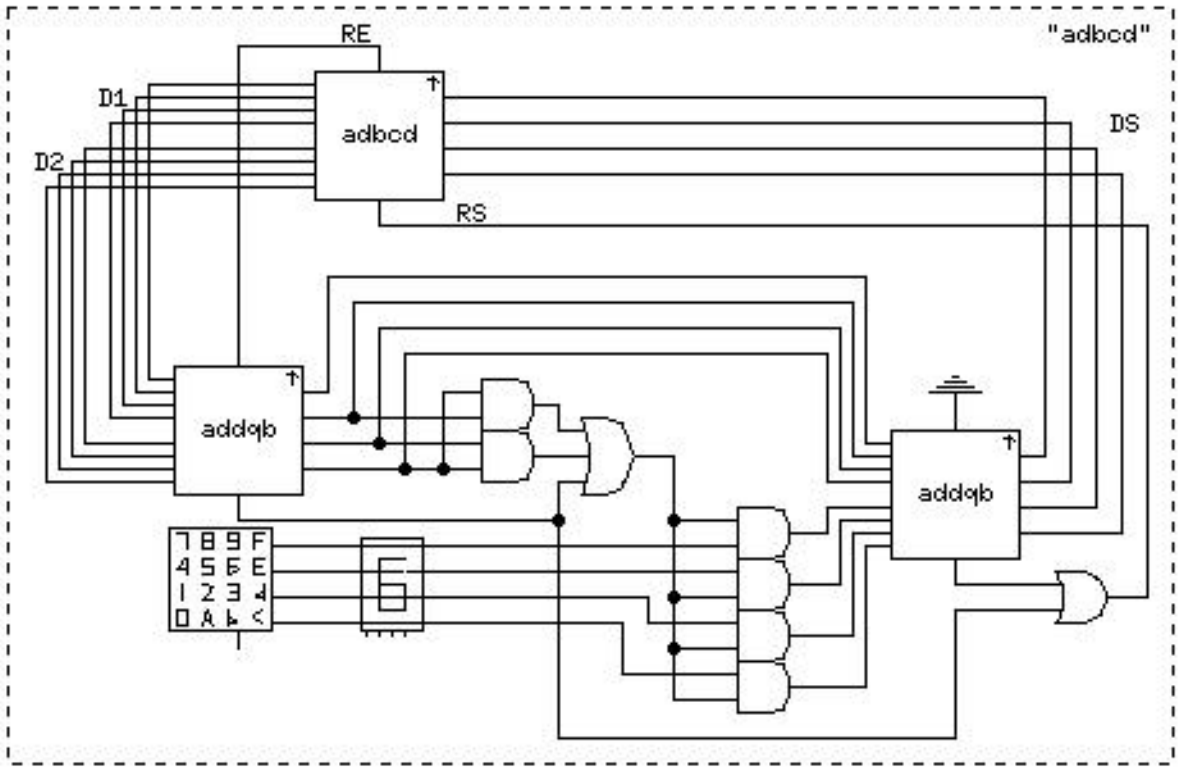
Exemples :

$5 + 4 = 0101_b + 0100_b = 1001_b = 9$ (pas de dépassement).

$5 + 7 = 0101_b + 0111_b = 1110_b$ (dépassement car 1110_b n'est pas un code BCD)
$1110_b + 0110_b = 0010_b$ avec une retenue soit 3 avec une retenue, ce qui correspond au résultat désiré.

Pour ajouter la valeur 6, on doit utiliser un second additionneur à quatre bits où le premièr opérande est le résultat de l'addition, et l'autre la valeur 6 s'il y a dépassement, 0 sinon.

Le tout est résumé par le circuit suivant qui définit le module d'addition BCD.



Additionneur bcd

1.2. Soustraction BCD

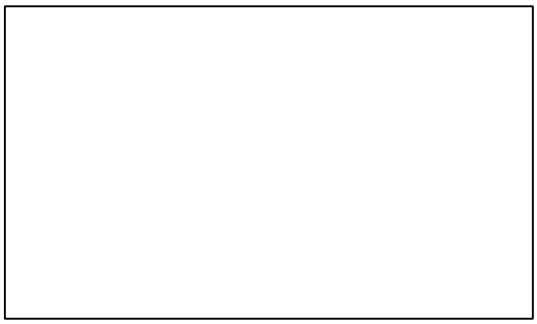
Un premier choix de conception revient à convertir le code du digit à soustraire à son complément à deux, puis l'additionner avec le code de l'autre digit (additionneur quatre bits) . On teste si le résultat est positif (absence de retenue). Si oui, alors le calcul est finit. Sinon, le calcul réalisé est la négation de la différence des deux digits codé en complément à deux, on doit donc faire une correction vers la valeur qui code de digit résultant de la soustraction et générer une retenue.

Toutefois, on préfère une conception avec des soustracteurs binaires, car ceci utilisera moins de portes logiques (pas de complément à deux). On va donc reprendre exactement les mêmes étapes que pour l'addition.

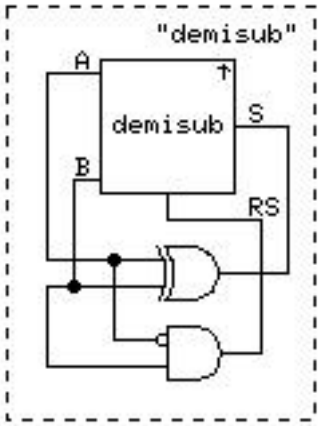
1.2.1. Soustracteur à un bit

Demi-soustracteur à un bit

Table de Karnaugh :



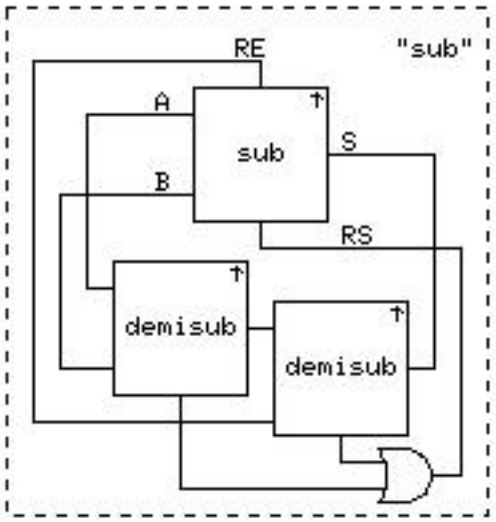
Circuit :



Demi additionneur a deux bits

Soustracteur complet

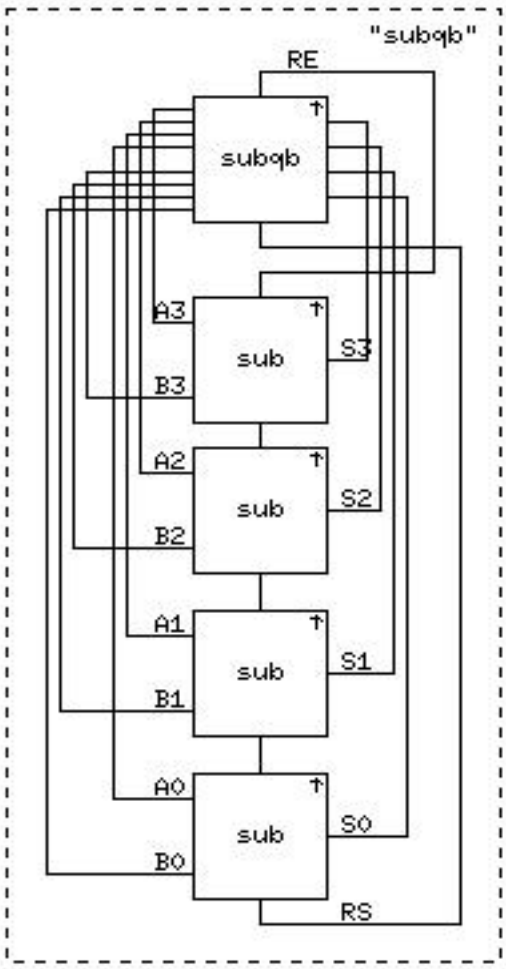
Comme pour l'addition, on déduit le circuit suivant pour la soustraction avec retenue entrante.



Soustracteur à un bit

1.2.2. Soustracteur quatre bits

Le circuit suivant définit le module de soustraction binaire sur quatre bits.

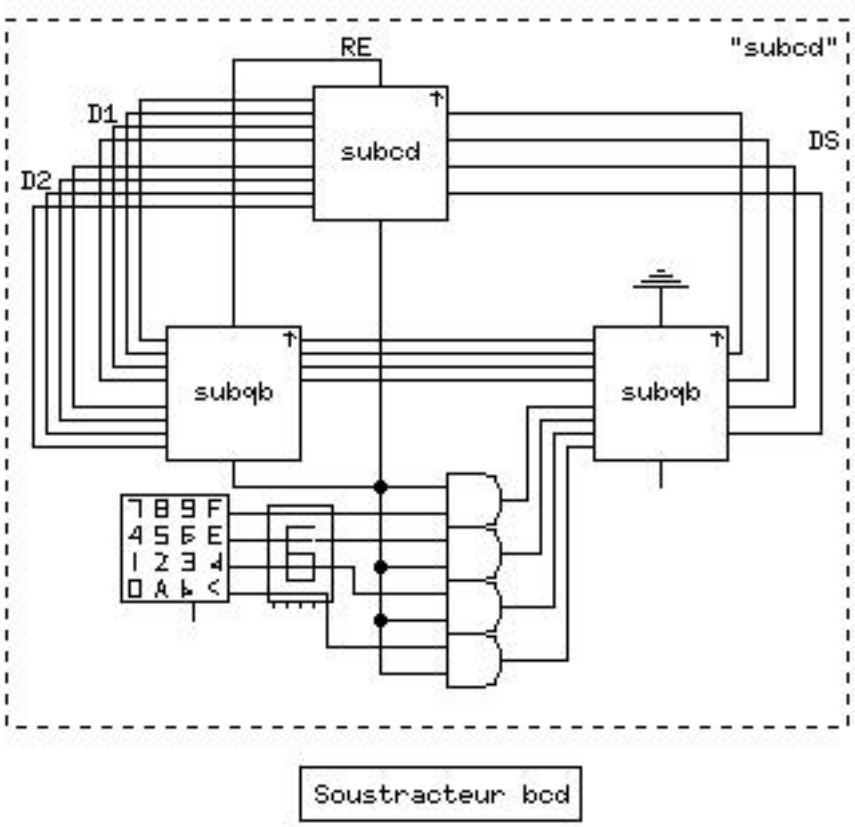


Soustracteur à quatre bits

1.2.3. Soustracteur BCD

Par un raisonnement similaire à celui de l'addition BCD, on déduit qu'une soustraction de deux digits avec le soustracteur à quatre bits est valide si aucune retenue n'est générée (pas de dépassement de 10 car il s'agit d'une soustraction). Dans le cas contraire, on doit soustraire du résultat obtenu la valeur 6 et générer une retenue (cela revient exactement à additionner le premier opérande avec dix puis effectuer la soustraction, mais cela est beaucoup plus coûteux en porte logiques).

Voici le circuit d'un tel composant.



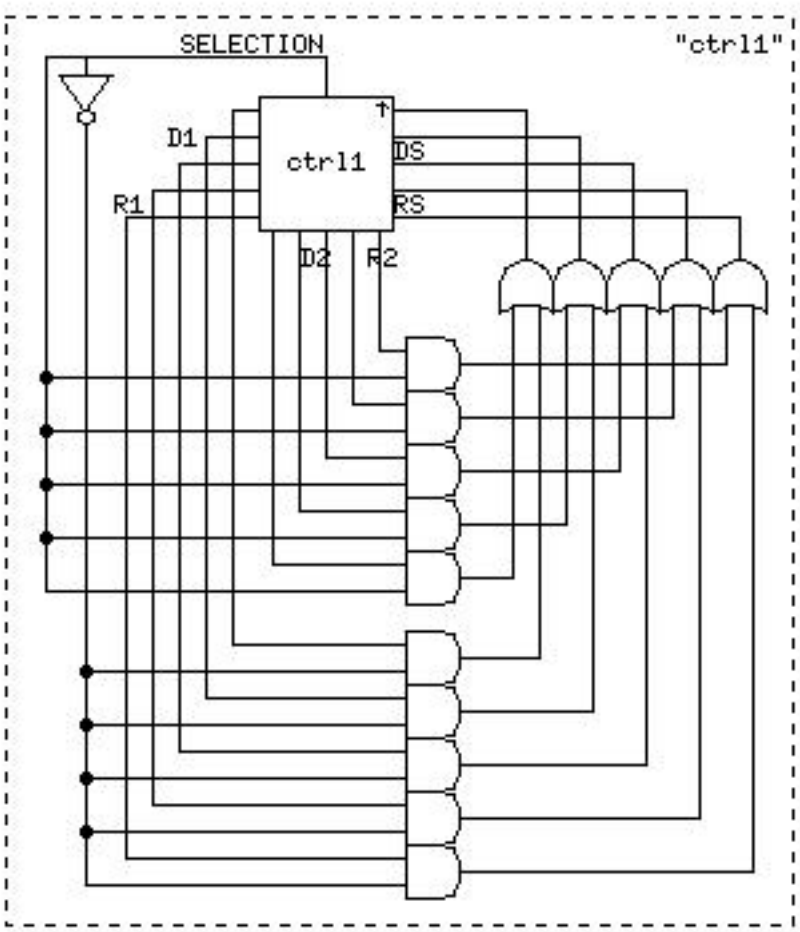
Soustracteur bcd

On remarque que ce circuit est plus simple que celui de l'addition sur quatre bits, cela s'explique par le fait qu'en soustraction on ne traite pas les résultats négatifs.

1.3. Le circuit de control

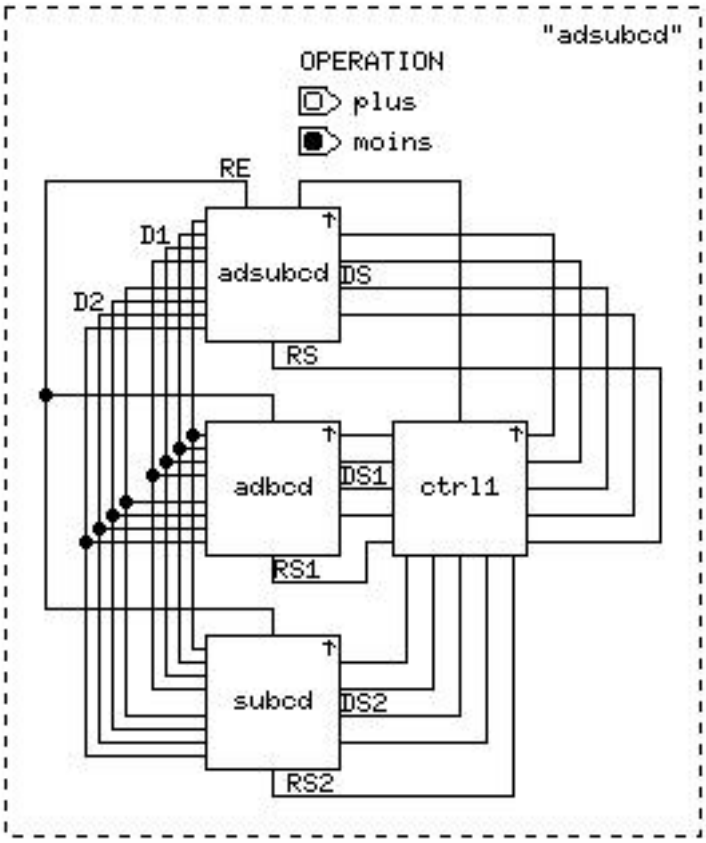
Le circuit de control permet de choisir entre deux résultats calculés (addition ou soustraction), un résultat prend 5 bits, la valeur du digit résultant sur 4 bits et la retenue. Ce circuit correspond tout simplement à un multiplexeur avec une entrée de commande à un bit, qui code les deux chois possibles.

Voici le circuit correspondant.



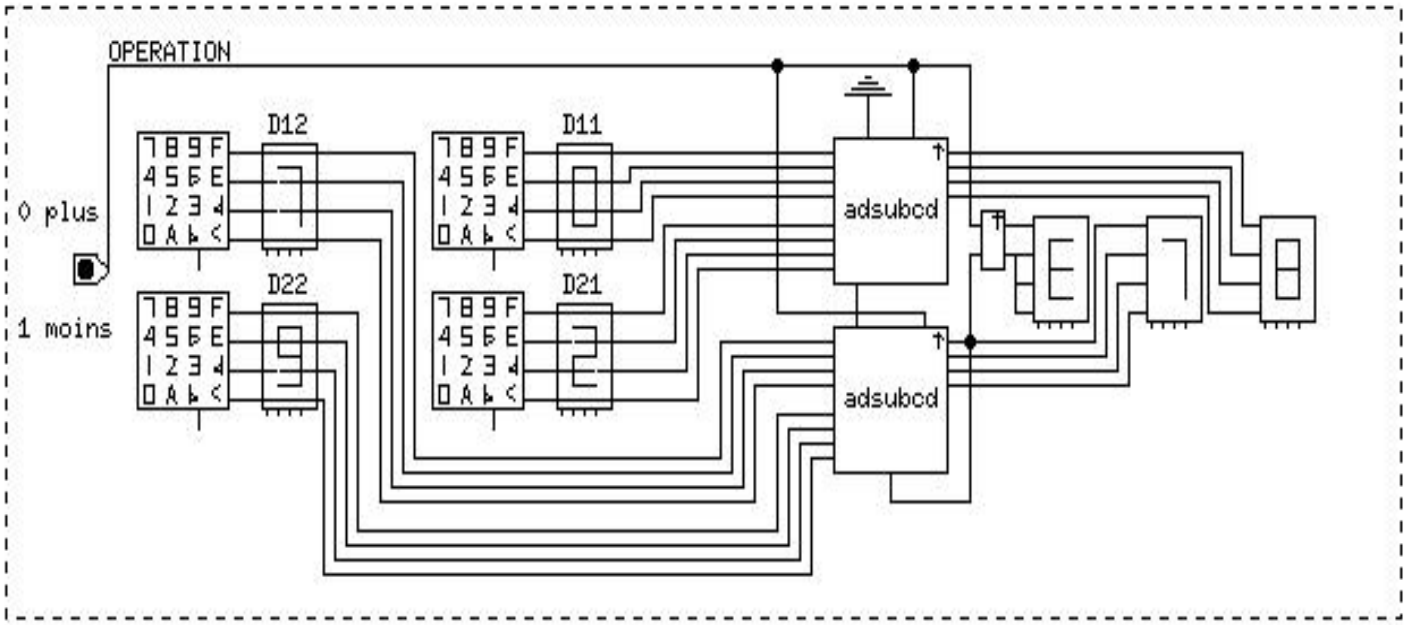
Circuite de control d'opération a éffectuer

On combine le circuit de control précédent avec un additionneur BCD et un soustracteur BCD pour obtenir un seul composant permettant de réaliser les deux opérations au choix.



Additionneur et soustracteur bcd

Voici circuit qui utilise deux composants d'addition et soustraction BCD. Il permet de réaliser une opération sur deux digits. La retenue du calcul sur les digits du poids faible est bien transmise au calcul sur les digits suivants. Si après une soustraction une retenue est générée, alors le résultat est négatif, on affiche alors une erreur. On peut avoir un résultat sur 3 digits dans le cas d'une addition, le digit des centaines est alors la retenue sortante.

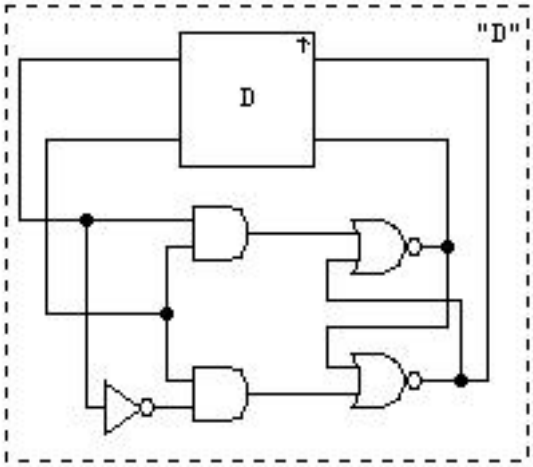


Exemple d'utilisation du module "adsubcd" avec deux digits

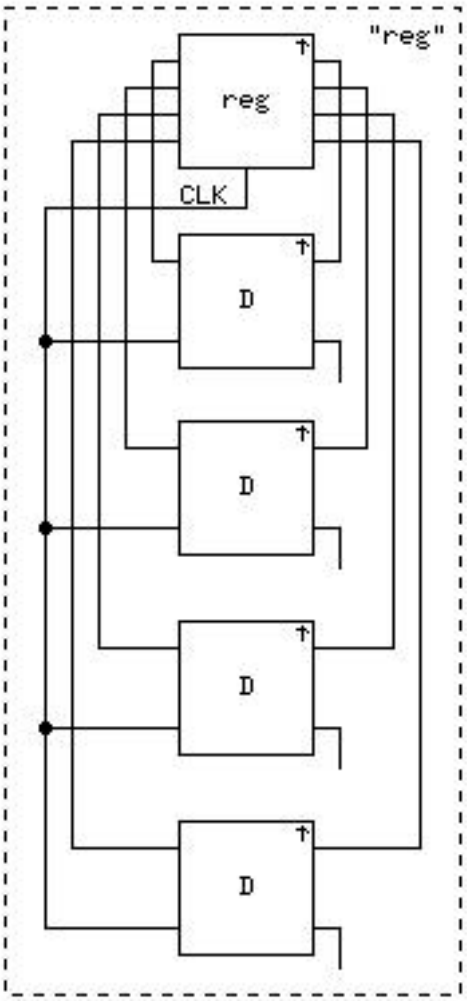
1. Second étape

Pour pouvoir sauvegarder les opérandes et opération entrées en série, on a besoin de registre de sauvegarde réalisé à l'aide de bascule D.

Voici les circuits correspondants.



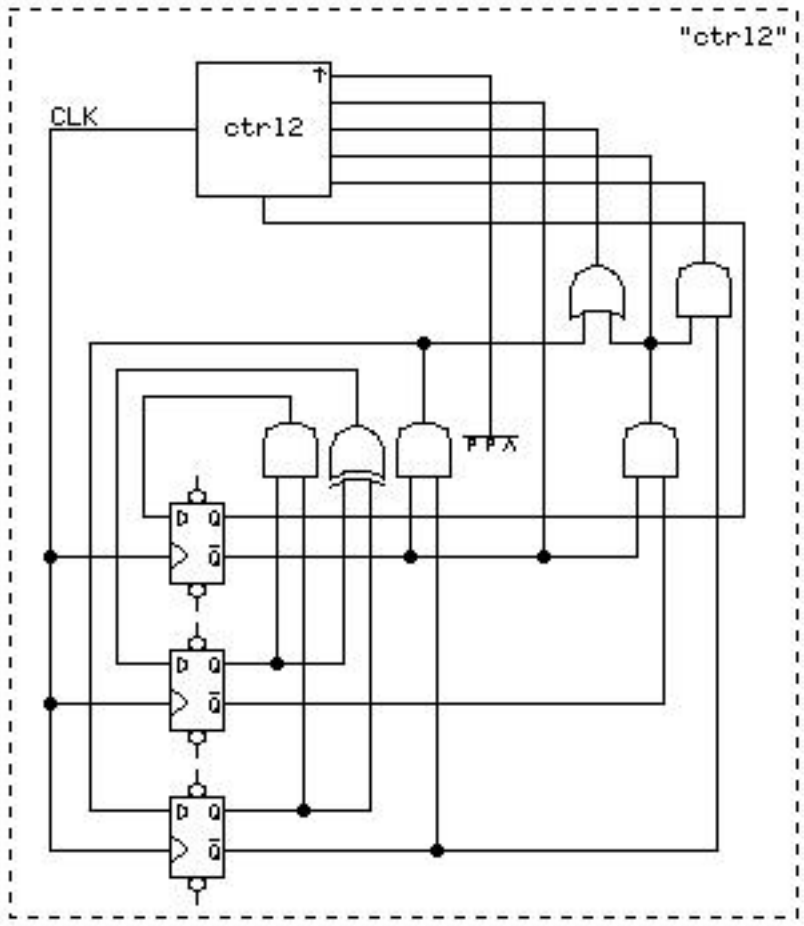
Bascule D (latch)



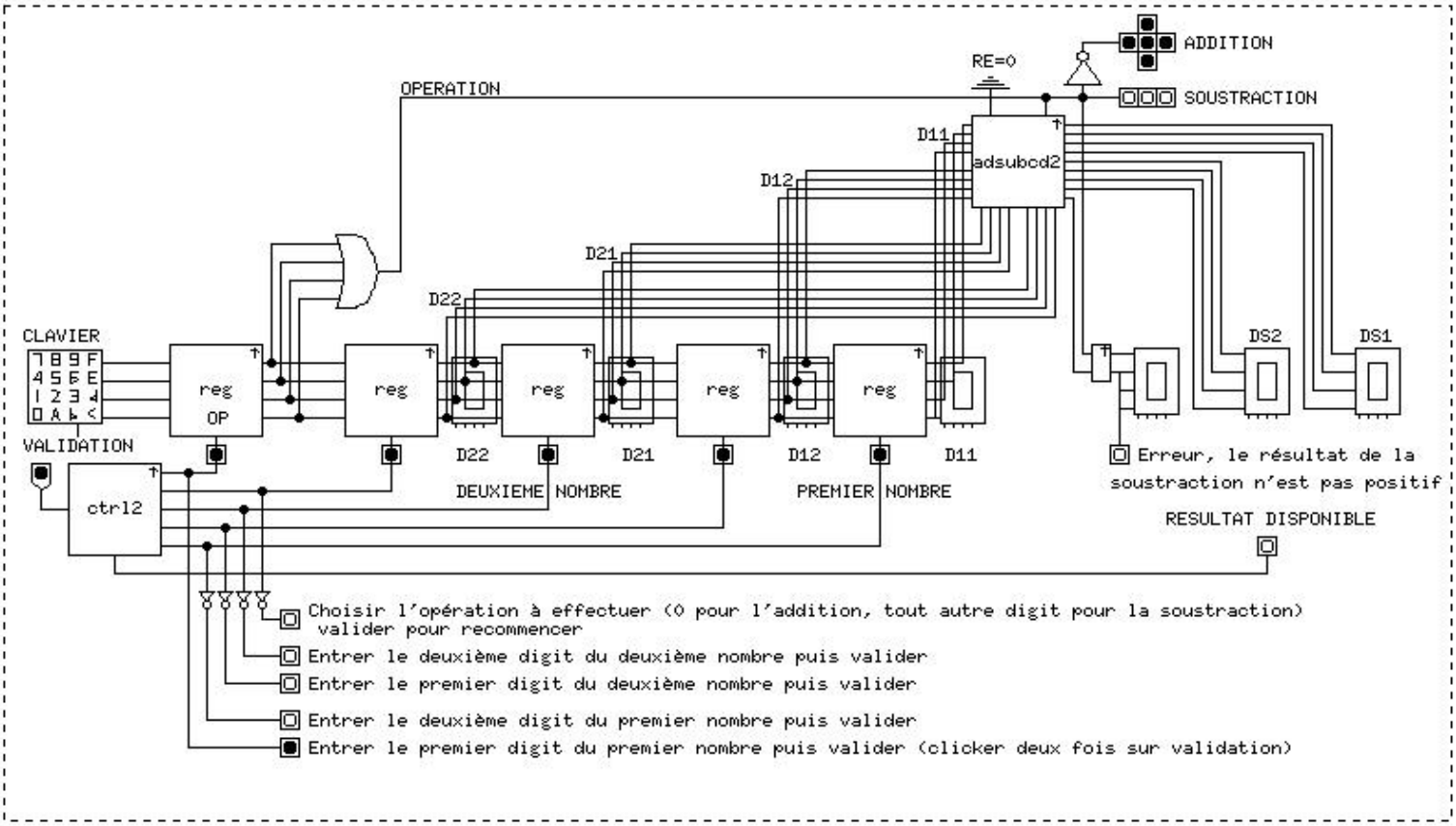
Registre quatre bits

Pour sauvegarder les données entrées en série, on doit réaliser un circuit qui commande les entrées des registres pour autoriser l'écriture.

Voici le circuit de commande suivi du circuit final d'adition et soustraction BCD.



Circuit de control sequentiel



Exemple d'utilisation du module de control séquentiel et des registres, combinés avec le module d'addition et soustraction sur deux digits