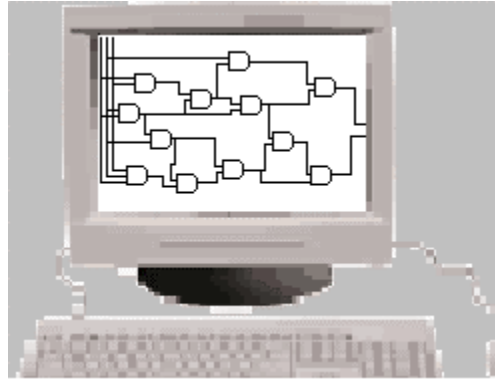


## Calculons avec Albert !



### Par :

Guy Lefebvre, 1257 rue Principale, St-Prime, G8J 1V2, 418-251-2170

Guillaume Rainville, 610 8<sup>e</sup> rue, St-Prime, G8J 1P6, 418-251-8290

**Résumé** : Lefebvre G. et Rainville G., 2001, Calculons avec Albert !, Expo-Journal, rapport interne, programme des sciences de la nature, Cégep de Saint-Félicien, Saint-Félicien, 7 pages

Notre but au départ était de concevoir une calculatrice électronique pouvant effectuer les quatre opérations, additionner, soustraire, multiplier et diviser. Pour ce, nous avons utilisé le logiciel de simulation électronique LogicWorks. Nous avons constaté qu'il était plus difficile que prévu d'effectuer les quatre opérations, surtout la division. De plus, notre logiciel avait ses limites et le temps nous a manqué. Bien que nous n'ayons pas complètement atteint notre objectif, nous avons réussi quelque chose d'intéressant et nous avons trouvé tous les éléments qui nous ont fait défaut.

**Mots-clés** : Calculatrice, circuit logique, porte logique, opération mathématique, binaire

## CALCULONS AVEC ALBERT

Notre travail, classé dans la catégorie mathématique, porte en fait sur l'électronique. Notre travail est un montage électronique sur logiciel, il n'est pas implanté dans un "Proto-Board" ou sur un circuit imprimé. Étant donné que ce n'est pas quelque chose de nouveau, nous n'avons pas formulé d'hypothèse sur le sujet. Nous nous sommes plutôt fixé un but qui est le suivant : concevoir et implanter dans un logiciel de simulation électronique les circuits logiques d'une calculatrice pouvant effectuer l'addition, la soustraction, la multiplication et la division. Pour cela, nous avons consulté des volumes sur les matériaux semi-conducteurs qui sont à la base de l'électronique (*Introduction aux semi-conducteurs* de Louis Trussart) et sur les circuits logiques, l'algèbre de Boole et la logique combinatoire (*Introduction aux circuits logiques* de Jean Letocha). Le but réel de notre projet n'était pas de modifier ou d'améliorer la calculatrice actuelle mais bien d'en concevoir une pour mieux la comprendre et, surtout, pour la faire comprendre à tous. Il est à noter que les volumes que nous avons utilisés, bien qu'ils soient très bien conçus, sont des volumes d'introduction à l'électronique. Pourtant nous avons réussi à concevoir un circuit relativement complexe. Une calculatrice demeure la pointe de l'iceberg qu'est l'électronique mais elle permet de comprendre le fonctionnement du plus puissant des ordinateurs.

## MATÉRIELS ET MÉTHODES

### Matériels

- Ordinateur
- Logiciel "LogicWorks<sup>tm</sup> for Windows" version 3.0.3. de Capilano Computing Systems Ltd

Le logiciel "LogicWorks" est un logiciel de simulation électronique qui permet de réaliser des circuits logiques et de s'assurer de leur bon fonctionnement. On l'utilise habituellement avant d'implanter réellement un circuit électronique dans un "Proto-Board" ou sur un circuit imprimé pour éviter de perdre du temps et de l'argent.

### Méthodes

La méthode que nous avons utilisée repose davantage sur la réflexion que sur la recherche et l'expérimentation.

Nous n'avons pas conçu les tables de vérités et les tables de Karnaugh comme cela comme ici

E <sub>1</sub>	E <sub>2</sub>	S
0	0	0
0	1	0
1	0	0
1	1	1

BA	00	01	11	10
0	0	1	0	1
1	1	0	1	0

qui sont à la base de l'addition et de la soustraction puisqu'elles étaient déjà présentes dans le volume *Introduction aux circuits logiques*, les schémas de l'addition et de la soustraction étaient eux aussi présents dans le volume. C'est quelque chose de très simple et nous n'aurions pas eu besoin du volume pour le faire mais puisqu'ils étaient déjà faits, nous les avons utilisés. Le travail consistait plutôt à créer des entrées, agencer les additionneurs partiels, les comparateurs et transformer les résultats binaires en décimaux. Pour cette dernière opération, nous avons utilisé la méthode des tables de vérités et des tables de Karnaugh pour formuler et simplifier nos équations. Ces méthodes sont expliquées en détail dans le volume *Introduction aux circuits logiques*. Pour ce qui est de la multiplication, nous avons procédé par simple réflexion. Sachant qu'elles étaient les entrées et les sorties, nous avons conçu un schéma de multiplicateur complet en trouvant par raisonnement logique la combinaison des portes logiques. Pour ce qui est de la division, nous avons trouvé dans un volume comment effectuer des divisions en binaire de la même manière que nous effectuons des divisions en décimal sur papier. Toutefois, cela demandait un raisonnement humain puisqu'il faut savoir combien de fois peut-on enlever un nombre d'un autre nombre. Devant la complexité de la tâche et le manque de temps à notre disposition, nous avons décidé de laisser tomber la division pour nous consacrer aux trois premières opérations. Le logiciel LogicWorks ne comportait pas de clavier numérique, seulement des interrupteurs en binaires. Nous avons donc utilisé quatre interrupteurs binaire pour entrer chacun des chiffres que nous voulions traiter.

## RÉSULTATS

### Addition

L'addition a fonctionné assez bien. Assez bien, parce qu'il y avait encore un petit défaut. Nous l'avons pas observé avant l'exposition. C'est lorsqu'un visiteur nous a demandé d'additionner un 8 avec un 9 que nous avons découvert la faille. L'addition d'un 8 et d'un 9 nous donnait 1 ou lieu de 17. Nous avons d'abord cru que quelqu'un était venu modifier notre programme mais après inspection nous avons compris ce qui se passait. L'addition de 1000 et 1001 en binaire donne 10001. Or, la réponse ne contient que 4 bits et le premier 1 est enlevé, ce qui donne 0001. Pourtant l'opération était effectuée par une puce d'addition provenant du logiciel. Nous n'avons toujours pas réglé ce problème. Le reste de l'addition fonctionne bien. Nous avons pu additionner deux nombres à trois chiffres. Ce qui donne une réponse à quatre chiffres. La réponse s'affiche en décimal sur les afficheurs numériques. Toutefois, il y a présence de 0 avant les chiffres significatifs. Par exemple, 34 s'affichait 0034. C'est un problème tellement bénin que nous n'y avons même pas pensé. De toute façon, ç'eut été très difficile à régler pour le peu d'avantage que ça apporte.

### Soustraction

Le soustracteur est fait uniquement de portes logiques. Nous n'avons pas utilisé de puce de soustraction. Les entrées et les sorties sont restées les mêmes. Nous n'avons fait que changer la partie du milieu qui fait le traitement. Nous n'avons pas fonctionné par addition de complément. Cela ressemblait trop à un additionneur. Nous avons fait de nouvelles équations avec des tables de vérité et de Karnaugh. La soustraction fonctionnait sauf qu'on avait un problème au niveau de

la retenue. Si le chiffre auquel on en soustrait un autre est plus petit, on lui ajoutait 10 comme lorsqu'on emprunte sur le voisin. Cependant le voisin n'était pas réduit en conséquence. Nous n'avons pas réussi à corriger le problème à cause d'un manque de temps. La soustraction fonctionnait correctement uniquement lorsque tous les chiffres du premier nombre étaient plus grand que ceux du deuxième nombre.

## **Multiplication**

La multiplication fonctionne en théorie. Mais comme la grandeur de la page est limitée dans le logiciel, nous n'avons pas pu l'implanter dans le logiciel. L'additionneur occupe presque tout l'espace de la page et il y en aurait fallu plusieurs pour effectuer l'addition. Nous en avons tout de même dessiné un plan sur papier. Ce plan est représenté à l'annexe A.

## **DISCUSSION**

Au départ, nous avons comme but d'implanter dans le logiciel LogicWorks les circuits logiques permettant d'additionner, de soustraire, de multiplier et de diviser. Comme nous l'avons vu dans la section Résultats, nous n'avons pas atteint notre objectif tel que formuler au début mais, en dépit des limites que nous imposait le logiciel et du manque de temps à notre disposition, nous avons presque atteint notre objectif. Examinons en détail la manière dont nous sommes parvenu à nos résultats ainsi que les améliorations possibles pour atteindre notre objectif de départ.

### **Addition**

L'addition a bien fonctionné mais elle n'était pas parfaite. D'abord, le problème dans l'addition de deux chiffres supérieurs à 7. Dans la section Résultats, nous avons cité l'exemple du  $8 + 9$ , mais, en fait, c'est à partir du moment où le résultat est égale ou supérieur à 16 que le problème se présente. En effet, 16 devient 10000 en binaire, 17 devient 10001, et 18 devient 10010. Comme la puce enlève le premier 1, cela donne respectivement 0000, 0001 et 0010. Nous aurions aussi pu faire un clavier décimal et éviter l'affichage de 0 non-significatifs. Nous verrons dans la section Suggestions comment nous aurions pu régler le problème de l'additions de deux chiffres supérieurs à 7, le problème du clavier et des 0 non-significatifs. Notre additionneur fonctionne en décimal codé binaire. C'est comme si nous effectuions une addition en décimal mais tous les chiffres sont remplacés par leur équivalent binaire. Chaque pair de chiffre à additionner (par exemple le 3 et le 5 dans  $643 + 865$ ) est regroupé dans le montage. Leurs interrupteurs sont placés un à côté de l'autre. Il y a trois paires d'interrupteurs dans le montage. Les deux chiffres sont additionnés dans la puce d'addition. S'il y a retenu, elle est dirigée vers les deux chiffres suivants où elle sera additionnée au résultat de l'addition de ces derniers. Chaque chiffre binaire faisant partie de la réponse est converti de manière en sept signaux correspondants aux sept segments de l'afficheur.

## Soustraction

La soustraction fonctionnait dans certaines conditions, mais un petit détail la rendait inopérante dans la majorité des cas. En effet, lorsque l'on soustrayait un chiffre à un autre chiffre plus petit, un problème survient. Habituellement, lorsqu'on soustrait à la main, on emprunte au voisin lorsqu'on ne peut pas soustraire un chiffre d'un autre parce que celui du bas est plus grand que celui du haut. Une dizaine devient dix unités lorsqu'on l'emprunte. Nous avons ajouté 10 au plus petit chiffre lorsque nécessaire. Il aurait fallu soustraire un à la colonne d'à côté. Dans le cas où le voisin vaudrait 0, il faut aller emprunter chez le deuxième voisin, laisser 9 au premier voisin et ajouter 10 au chiffre trop petit. Cela semble simple pour un être humain, mais, ç'aurait été très compliqué pour un circuit logique. Il aurait été préférable de procéder par addition d'un complément à 1 ou à 2. C'est ce procédé qu'on utilise habituellement pour faire des soustractions en électronique. Cela évite d'avoir recours aux emprunts. La soustraction par addition du complément à 1 est très simple. Au lieu de soustraire 0011, on additionne 1100 (on remplace les 1 par des 0 et les 0 par des 1). Pour trouver la réponse, on coupe le premier chiffre (celui dont la puissance est la plus grosse) et on ajoute 1 au dernier chiffre. Par exemple,  $1010 - 0011 = 111$ . Par addition du complément,  $1010 + 1100 = 10110$ . On enlève le premier chiffre et on additionne 1 au dernier pour obtenir 0111. Mais pour éviter le problème de l'emprunt, il aurait fallu procéder en binaire simple plutôt qu'en décimal codé binaire, puis reconverter en décimal codé binaire pour afficher le résultat. Finalement, la soustraction est plus compliquée qu'elle en a l'air

## Multiplication

Pour ce qui est de la multiplication, si elle n'a pas pu être implantée c'est uniquement parce que le logiciel ne nous offrait pas assez d'espace pour implanter toutes ses composantes. Contrairement à ce que nous croyions au départ, la multiplication n'a pas été très difficile à concevoir. La multiplication de deux chiffres binaires n'est rien d'autre qu'une porte Et. En effet,  $0 * 0 = 0$ ,  $0 * 1 = 0$ ,  $1 * 0 = 0$  et  $1 * 1 = 1$ . Nous avons, encore une fois, opéré avec le décimal codé binaire. Le procédé est le même que lorsque nous effectuons des multiplications à la main. Deux chiffres décimaux codés en binaires sont multipliés ensemble, cela donne comme résultat un nombre binaire de 7 chiffres. Par exemple,  $1000 * 1001$  donne 1001000 (8 \* 9 donne 72). Il faut cependant transformer 1001000 en 0111, 0010 pour rester en décimal codé binaire. Nous avons transformé 72 en lui enlevant 10 autant de fois que possible et en comptant ce nombre de fois. Après avoir enlevé 10 sept fois, il reste 2, d'où le 0111 et le 0010. La soustraction du 10 se fait par l'addition du complément à 2. Lorsque la soustraction devient négative (par exemple, lorsqu'on enlève 10 à 2), on arrête les soustractions de 10, on arrête le décompte des soustractions de 10 et on prend le dernier chiffre issu de la série de soustraction. Tout comme lorsque l'on fait à la main, notre multiplication comporte plusieurs additions et c'est pourquoi nous n'avons pas réussi à la faire entrer dans une page de LogicWorks.

## Division

Bien que nous n'avons pas conçu la division, nous croyons qu'il est bon d'en parler dans cette discussion. Théoriquement, il y a trois façons d'effectuer une division. La première est celle que nous utilisons pour effectuer une division à la main. Cependant, pour cela, il faut être capable de dire combien de fois un nombre entre dans un autre nombre. Cela demande un raisonnement

complexe que nous n'avons pas réussi à reproduire avec un circuit logique. Nous avons donc rejeté cette voie. La deuxième consiste à multiplier par l'inverse. Pour cela, chaque nombre aurait du avoir un exposant. Pour diviser deux nombres il aurait suffi de les multiplier en inversant le signe de l'exposant du dénominateur. Pour afficher le résultat, il aurait fallu convertir le nombre et son exposant et un nombre à la puissance 1. Cela aurait demandé plusieurs convertisseurs qui ne seraient probablement pas entrés dans la page. De plus, nous aurions eu besoin d'une virgule déplaçable sur les quatre afficheurs. Cela aurait nécessité plusieurs portes supplémentaires, ajoutant du poids à un circuit déjà trop gros. La dernière méthode consiste à soustraire le dénominateur du numérateur et de compter le nombre de fois que l'on peut le faire, un peu comme on faisait pour convertir le résultat de la multiplication. Encore là, le nombre de soustraction requiert aurait rendu le montage trop gros pour la page. Peut-être qu'avec des bascules ou des registres, nous aurions pu contourner ce problème.

## **CONCLUSION**

Notre expérimentation nous a permis de découvrir que le fonctionnement d'une calculatrice est quelque chose d'assez complexe. Complexe, mais pas incompréhensible. Nous croyons qu'avec un peu plus de temps et de moyens, nous parviendrions sans doute à atteindre notre objectif de départ qui était de concevoir les quatre opérations d'une calculatrice. Nous avons pu constater le nombre élevé de composants électroniques nécessaire pour effectuer les opérations mathématiques, surtout la division. Notre travail nous a permis de mieux constater l'état actuel de la miniaturisation. Sachant le nombre de pièces qui composent une calculatrice et la grosseur de cette dernière, on peut imaginer la petitesse de ces pièces.

## **IMPACT SUR L'ENVIRONNEMENT ET LA SOCIÉTÉ**

Bien entendu les impacts sur l'environnement de notre projet sont nuls et il serait ridicule de vouloir en inventer. Toutefois, notre projet peut avoir des répercussions intéressantes sur la société. Notre but premier était de vulgariser l'électronique, de faire comprendre la chimie à la base des portes logiques et la manière dont on arrange ces dernières pour former des circuits logiques complets. Bien sûr, ce n'est pas facile d'enseigner l'électronique en quelques minutes alors que nous avons suivi un cours d'une session. Mais l'important, c'est d'intéresser le public à l'électronique et lui montrer les possibilités qu'elle offre.

## **SUGGESTIONS**

Voici comment nous aurions pu régler les problèmes ayant rapport à l'addition. Tout d'abord pour ce qui est de l'addition de deux chiffres ayant comme résultat un nombre supérieur à 15, nous n'aurions pas dû utiliser la puce d'addition, du logiciel. Nous aurions dû en faire une avec une sortie à 5 bits. Nous aurions ensuite converti le nombre à 5 bits en deux nombres à 4 bits en soustrayant 10 au nombre à 5 bits. Ainsi, 10001 serait devenu 0001 et 0111.

Nous aurions pu faire un clavier décimal un peu rudimentaire toutefois. Pour remplacer chacune des entrées à quatre bits, nous aurions pu mettre les dix chiffres du système décimal associé à leur équivalent binaire. Les boutons auraient été des interrupteurs binaires. Pour additionner un 8 par exemple il aurait fallu laisser passer le courant dans l'interrupteur du 9

seulement. Avec des portes Et, nous aurions pu laisser entrer 1001 dans l'additionneur seulement lorsqu'un courant passe dans l'interrupteur du 9. Cela n'aurait pas pris moins de temps pour effectuer l'opération mais ç'aurait permis à une personne qui ne connaît pas le code binaire d'utiliser notre calculatrice.

Pour éviter le problème lié à la soustraction, il aurait été préférable de travailler en binaire et de faire la soustraction par addition du complément à 1 ou à 2.

La multiplication et la division pourraient toutes deux être implantées mais pas dans le logiciel LogicWorks. Un logiciel plus performant aurait été préférable. Implanter le tout dans un Proto-Board, ou plutôt dans des centaines de Proto-Board, est impensable dans le cadre du cours parce que ce serait beaucoup trop long.

## **REMERCIEMENTS**

Nous voudrions spécialement remercier Bruno Martel, professeur de physique au Cégep de St-Félicien, d'abord pour nous avoir enseigné l'électronique et pour nous avoir aidé à utiliser le logiciel LogicWorks. Nous voudrions aussi remercier René Aubry, professeur d'informatique au Cégep de St-Félicien, pour nous avoir renseigné à propos de la division en électronique.

## **BIBLIOGRAPHIE**

Letocha, Jean, 1985, Introduction aux circuits logiques, Chenelière/McGraw-Hill, 298 pages

Trussard, Louis, 1993, Introduction aux semi-conducteurs, Éditronique éditeur, 472 pages