ABA Logique Nspire 2

Activité de découverte du logiciel

Pour découvrir les diverses fonctionnalités du programme, nous allons résoudre quatre exercices de logique type à l'aide de celui-ci. Le dernier est un problème classique !

EXERCICE 1 :

1.1. Simplifier par Karnaugh

$$F10 = a.b + \overline{c}.\overline{d} + \overline{a}.\overline{b}.\overline{c}.\overline{d} + \overline{a}.b.c.\overline{d}$$

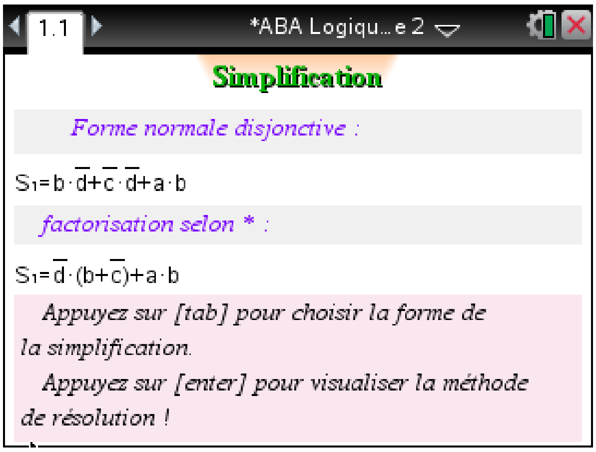
$$F8 = \overline{a}.\overline{b}.c.\overline{d} + \overline{a}.b.c.d + a.b.c.d + a.\overline{b}.\overline{c}.\overline{d} + a.b.\overline{c}.d$$

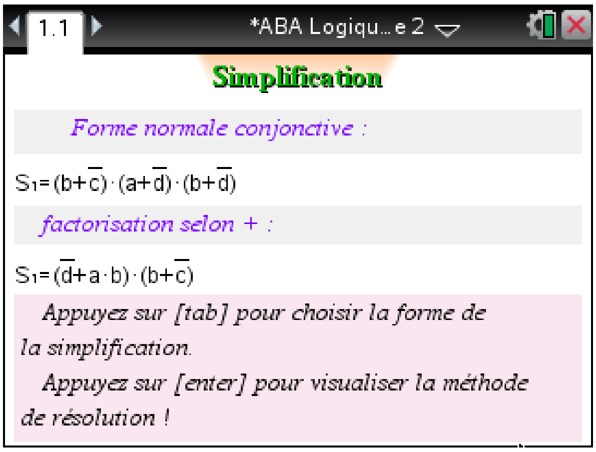
Résolution :

ABA Logique dispose d'un moteur de simplification par tableau de Karnaugh qui va nous permettre de simplifier ces expressions au maximum. Nous n'avons même pas besoin de remplir le tableau de Karnaugh correspondant, on peut saisir directement l'équation algébrique.

- Saisissons donc la première expression en cliquant sur le bouton « Nouvelle équation » dans le menu principal. (ou en le sélectionnant à l'aide des flèches puis en appuyant sur [enter], ou en tapant la touche [n])
- Effacez la ligne de saisie avec [ctrl]+[del](clear)
- Pour saisir l'équation efficacement, vous pouvez profiter de l'auto-complétion des « * » implicites ! En effet, vous n'avez qu'à saisir :
  ab+/c/d+/a/b/c/d+/abc/d
  Le logiciel reconnaîtra automatiquement les « ET ».
- Appuyez ensuite sur [enter] ou cliquez sur l'équation ou sur le bouton « Nouvelle équation » pour terminer la saisie.

- L’équation entrée est affichée en <u>écriture naturelle</u> et est ajoutée à l’<u>historique</u>.
- Nous allons à présent demander la <u>simplification</u> de l’équation courante :
  Pour cela, à l’aide des flèches ou de la souris ou bien de la combinaison [2] + [b], sélectionnez l’item « B : Simplification » dans l’onglet « 2) Opérations > ».
- L’équation simplifiée sous forme normale disjonctive ainsi que la factorisation sont alors affichées et ajoutées à l’<u>historique</u>.
  Appuyez sur [tab] pour visualiser la forme normale conjonctive et sa factorisation.
  On obtient donc :

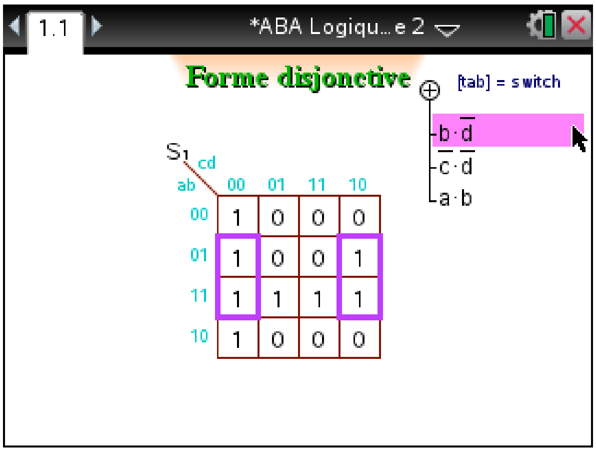


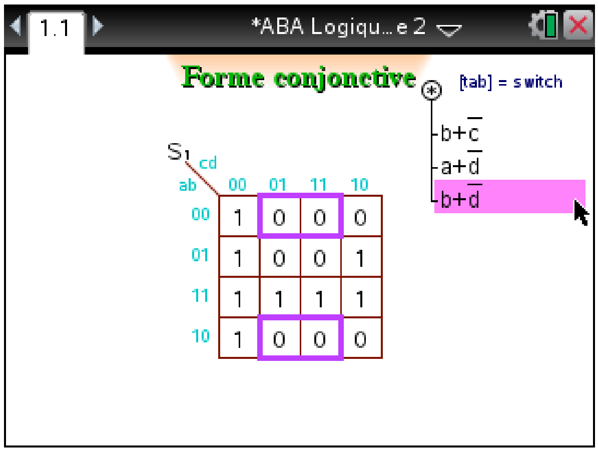


A vous de choisir la forme la plus appropriée ! On prendra pour réponse la forme normale disjonctive factorisée, et on a donc :

F10 = /d*(b+/c)+a*b

De plus, pour justifier votre simplification, vous pouvez visualiser les blocs de simplification qui ont permis de trouver la forme normale disjonctive (en regroupant les « 1 ») ou conjonctive (en regroupant les « 0 ») à l’étape suivante : appuyez sur [enter] pour afficher le tableau de Karnaugh de F10 puis naviguez dans les différents blocs de simplification avec les flèches ou avec le pointeur de la souris. Appuyez sur [tab] pour switcher entre forme normale disjonctive ou conjonctive (regroupement des « 1 » ou « 0 »).





Il n'y a vraiment qu'à recopier ! ;)

Vous pouvez suivre cette même procédure pour l'équation F8 de l'exercice. ☺
Vous devriez obtenir, pour la forme disjonctive factorisée : (10 termes)

F8 = /b*/d*(/a*c+a*/c)+b*d*(c+a)

Et pour la forme conjonctive factorisée : (8 termes)

F8 = (b+(/a+/c)*/d)*(a+c)*(/b+d)

Pour aller plus loin :

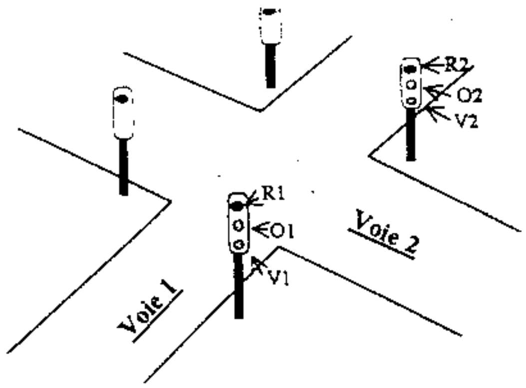
Nous allons vérifier si la simplification est correcte.

- Pour cela, utilisons 2) > « A : Vérifier équivalence »
- $S^{(1)}$ a déjà pour valeur l'équation que l'on a entrée précédemment. (c'est l'équation en cours)
- Appuyez donc sur [tab] pour saisir $S^{(2)}$.
- Cherchez dans l'historique la forme simplifiée voulue, ici celle avec l'en-tête simpFact* (factorisation de la forme disjonctive). Utilisez soit les flèches puis [enter], soit la souris, ou bien la combinaison [h] (ouvrir l'historique) + [*lettre_correspondante*] (rappeler l'équation repérée par *lettre_correspondante*)
- Appuyez sur [tab] pour continuer et constatez que le logiciel ne s'est pas trompé !
  Il devrait s'afficher $S^{(1)}$ et $S^{(2)}$ ainsi que le message « Les deux équations sont bien équivalentes ! »
- Appuyez ensuite sur [enter] ou cliquez, et vous verrez le tableau de Karnaugh de chaque équation.

EXERCICE 2 :

Présentation:

Nous nous proposons de réaliser, à l'aide de portes NAND à 2, 3 ou 4 entrées, le décodeur d'un montage électronique permettant le fonctionnement des feux tricolores d'un carrefour routier comportant 2 voies (voie 1 et 2. voir le dessin du carrefour ci-contre).



Dessin du carrefour

Le principe du montage électronique complet est présenté dans le schéma synoptique ci-dessous :

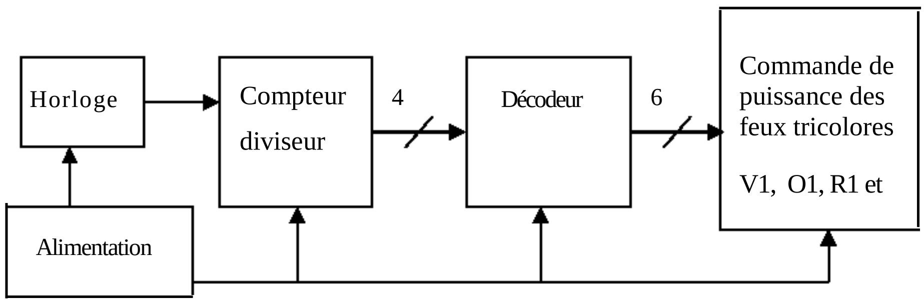


Schéma synoptique

Explication du principe:

- L'horloge délivre une impulsion toutes les 2 secondes.
- Cette impulsion est appliquée à l'entrée d'horloge d'un compteur diviseur par 16.
- Les 4 sorties (a, b, c, d) du compteur délivrent des signaux logiques conformes aux chronogrammes qui suivent, et sont appliqués aux entrées du décodeur (voir chronogrammes).

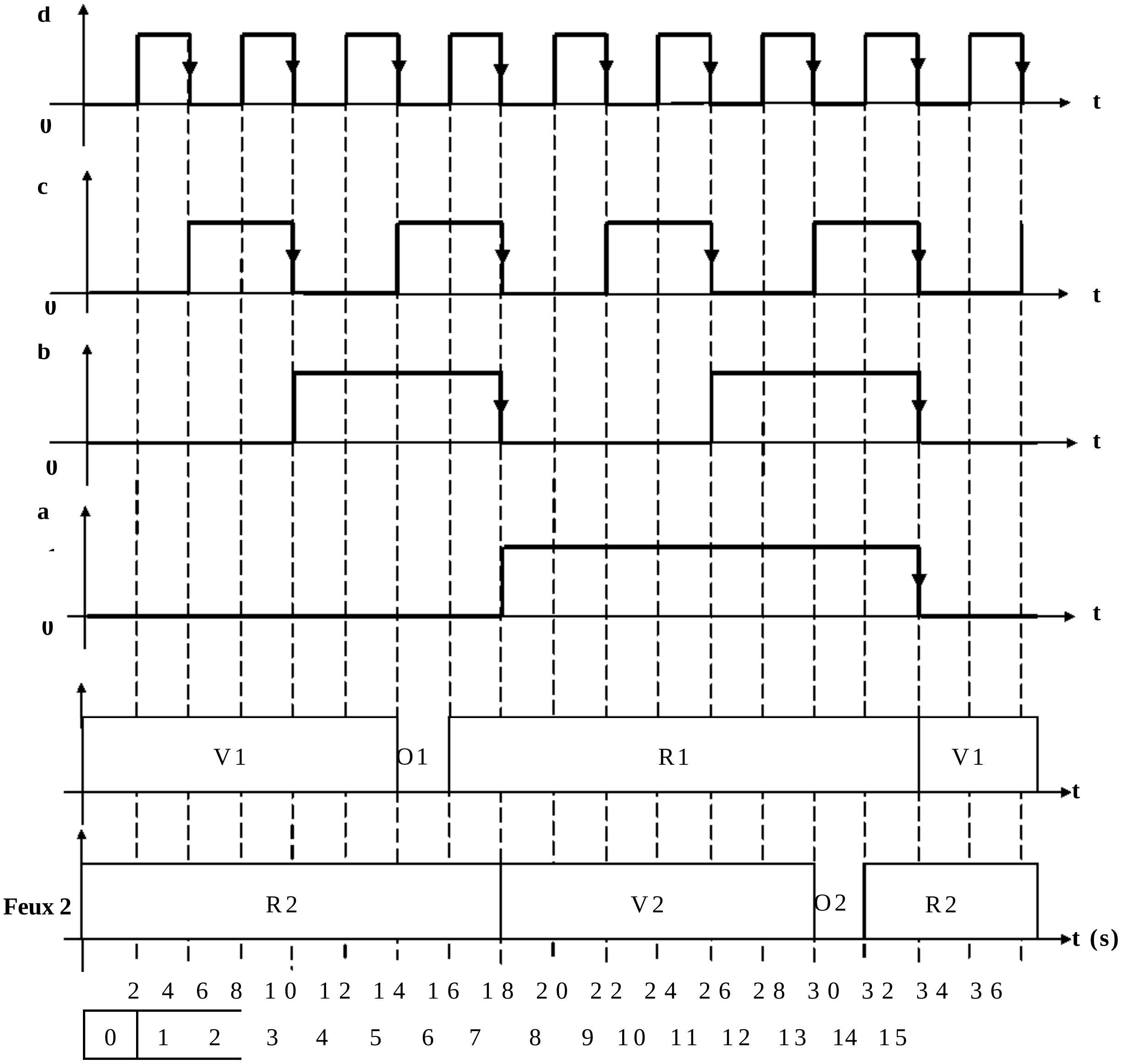
Travail demandé:

1.1. A partir des chronogrammes, remplir les tableaux de KARNAUGH de chaque sortie du décodeur en fonction des sorties du compteur.

1.2. En déduire les équations de chaque sortie.

1.3. Transformez les équations pour n'utiliser que les portes demandées dans la présentation. (Remarque : on pourra utiliser le fait qu'entre V1, 01 et R1 il n'y a toujours qu'une seule lampe d'allumée. Idem pour V2, 02 et R2).

Chronogrammes:



Résolution :

1 .1) & 1.2)

Sur le chronogramme, les quatre sorties a,b,c,d comptent en binaire. Ainsi, il est plus facile de retranscrire l'état de chaque feu dans une table de vérité en binaire naturel, car elle suit l'ordre du chronogramme.

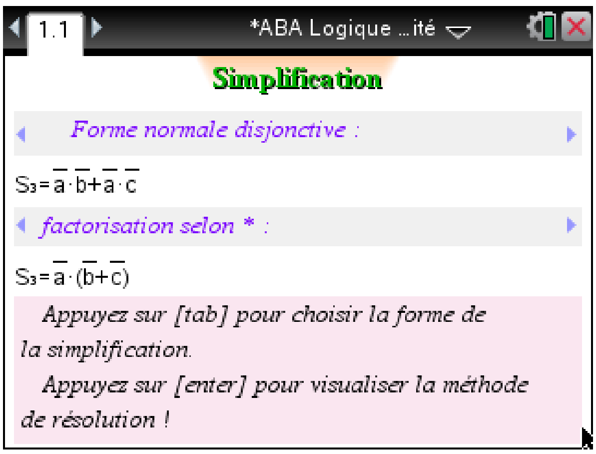
On arrive donc facilement à obtenir cette table de vérité :

a	**b**	**c**	**d**	V1	O1	R1	V2	O2	R2
0	0	0	0	1	0	0	0	0	1
0	0	0	1	1	0	0	0	0	1
0	0	1	0	1	0	0	0	0	1
0	0	1	1	1	0	0	0	0	1
0	1	0	0	1	0	0	0	0	1
0	1	0	1	1	0	0	0	0	1
0	1	1	0	0	1	0	0	0	1
0	1	1	1	0	0	1	0	0	1
1	0	0	0	0	0	1	1	0	0
1	0	0	1	0	0	1	1	0	0
1	0	1	0	0	0	1	1	0	0
1	0	1	1	0	0	1	1	0	0
1	1	0	0	0	0	1	1	0	0
1	1	0	1	0	0	1	1	0	0
1	1	1	0	0	0	1	0	1	0
1	1	1	1	0	0	1	0	0	1

Intéressons-nous donc d'abord à la sortie V1.

On va utiliser la fonction qui permet d'obtenir l'équation d'une table donnée, ici une table de vérité en binaire naturel.

- Allez dans « 3) Table=>Équation > » puis « B : depuis table bin naturel »
- Tapez sur la touche [4] car on veut une table de quatre variables (16 lignes)



- Entrez l'état associé de V1 (pour commencer) à chaque cas.
  Pour cela, vous pouvez naviguer dans la table avec les flèches et appuyer sur [1] ou [0] pour la remplir. Vous pouvez aussi utiliser le pointeur et le clic. (Vous pouvez faire défiler la table en cliquant aux extrémités de la barre de défilement.)
- Une fois le remplissage terminé, appuyez sur [enter] : le tableau de Karnaugh correspondant sera simplifié par la méthode de Karnaugh.
- Vous obtenez ainsi l'équation simplifiée de la table sous forme normale disjonctive et sa factorisation. (De même, vous pouvez switcher avec la forme normale conjonctive.) Les résultats sont ajoutés à l'historique. Vous obtiendrez :

  V1 = /a*(/b+/c)
- L'exercice demandait les tableaux de Karnaugh. Appuyez sur [enter] pour afficher le tableau de Karnaugh correspondant. Utilisez les flèches ou le pointeur pour naviguer parmi les différents « paquets » de simplification. Vous pourrez les recopier pour justifier la simplification.

1.3)

À présent, nous allons transformer l'équation trouvée pour pouvoir l'utiliser qu'avec des portes logiques NAND (NON ET).

- Pour cela, rappelez l'équation simplifiée dans l'historique (avec l'en-tête « tbnFact* »).
- Ensuite choisissez la fonction « D : Transfo pour NAND » dans l'onglet « 2) Opérations > »
- L'équation s'affiche alors en forme NAND tout en précisant le nombre de portes nécessaire ! (ici 4) Elle est aussi ajoutée à l'historique.
  V1 = //(/a*/(b*c))
  (C'est là qu'on apprécie particulièrement l'affichage pretty print ! ☺)

Maintenant, pour terminer l'exercice, vous devez refaire ces manipulations pour chacune des sorties restantes ! (O1, R1, V2, O2, R2)

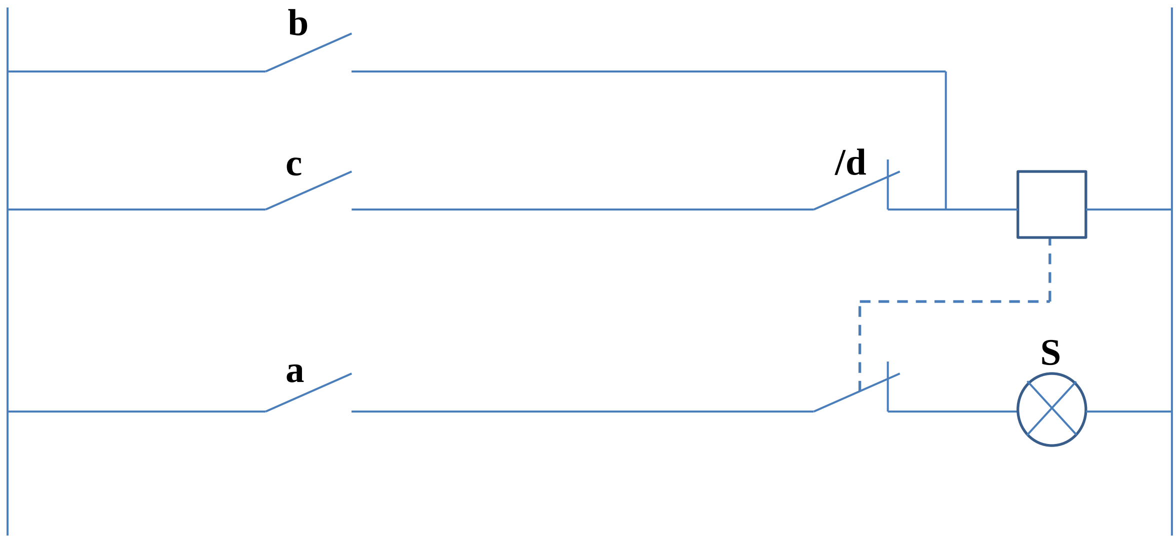
⇨ ***Voici les réponses, si vous voulez vérifier vos manipulations. (Il manque les tableaux de Karnaugh.)***

Equation simplifiée	Forme NAND
V1=/a*(/b+/c)	V1= //(/a*/(b*c))
O1=/a*b*c*/d	O1=//(/a*b*c*/d)
R1=a+b*c*d	R1=/(/a*/(b*c*d))
V2=a*(/b+/c)	V2=//(a*/(b*c))
O2=a*b*c*/d	O2=//(a*b*c*/d)
R2=/a+b*c*d	R2=/(a*/(b*c*d))

Exercices tirés de http://ressource.electron.free.fr/cours/recueil_karnaugh.pdf
Lycée Jules Ferry – Versailles

EXERCICE 3 :

Proposer un câblage équivalent (schéma ladder) sans utiliser de relais, pour réduire les coûts. Vous indiquerez les équations logiques associées à chaque câblage.



Résolution :

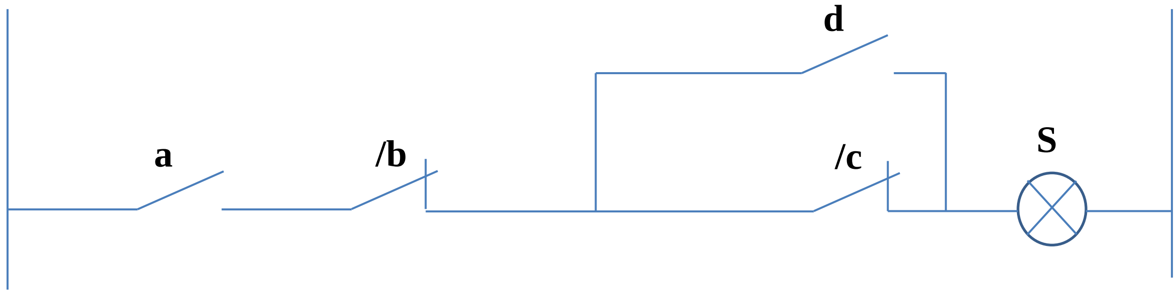
Il faut tout d'abord écrire l'équation correspondant au schéma. Si on appelle **R** la variable représentant l'état du relais, on a : **S** = **a***/**R** puisque **a** et le contact normalement fermé actionné par **R** sont en série. On cherche ensuite l'équation de **R** : **R** = **b**+**c***/**d**. L'équation logique du schéma est donc : S=a*/(b+c*/d)

On remarque que la présence d'une « grande barre » au-dessus d'une expression à plusieurs variables implique l'utilisation d'un relais commandant un contact normalement fermé.

Or on nous demande de ne pas utiliser de relais. On va donc transformer ce groupe de variables complémenté en variables uniques complémentées grâce au théorème de De Morgan.
Enfin... nous allons laisser notre chère TI le faire pour nous ! ☺

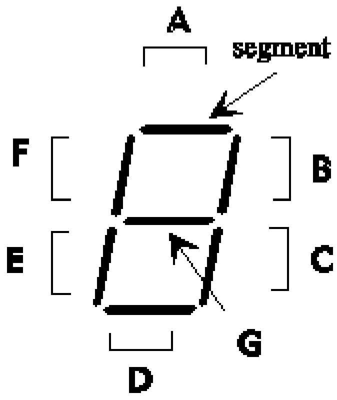
- Vous devez donc tout d'abord entrer l'équation trouvée. Là aussi, profitez de l'auto-complétion des « * » ! Vous n'avez qu'à taper : S=a/(b+c/d)   puis [enter] ou clic à la souris.
- A présent, demandez la fonction « C : Transfo pour ladder » dans « 2) Opérations > »
- Le résultat s'affiche alors et est ajouté à l'historique !
  S=a*/b*(/c+d)
- NB : vous auriez ici le même résultat en demandant la simplification, forme factorisée. Les simplifications ne font toujours intervenir que des variables uniques complémentées.

On peut donc tracer le schéma correspondant :



⇨ Pari gagné : aucun relais pour un résultat équivalent ! ;)

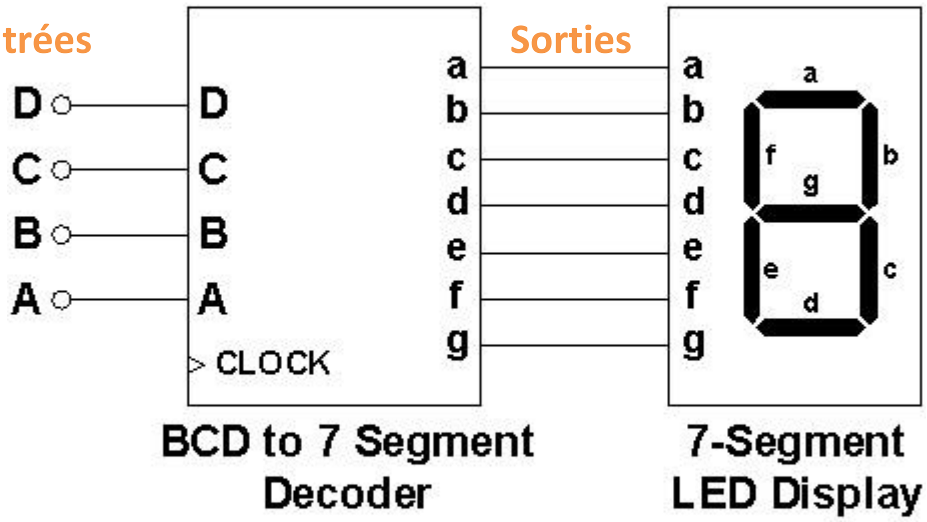
EXERCICE 4 :



On vous propose de réaliser les équations de fonctionnement d'un afficheur 7 segments, permettant de représenter le BCD (Binary Coded Decimal). Cet afficheur peut afficher les chiffres de 0 à 9.

Les variables suivront l'évolution du binaire pur.

1) Combien d'entrées binaires seront nécessaires et suffisantes pour décrire le fonctionnement de tous les chiffres ? Expliquez pourquoi.



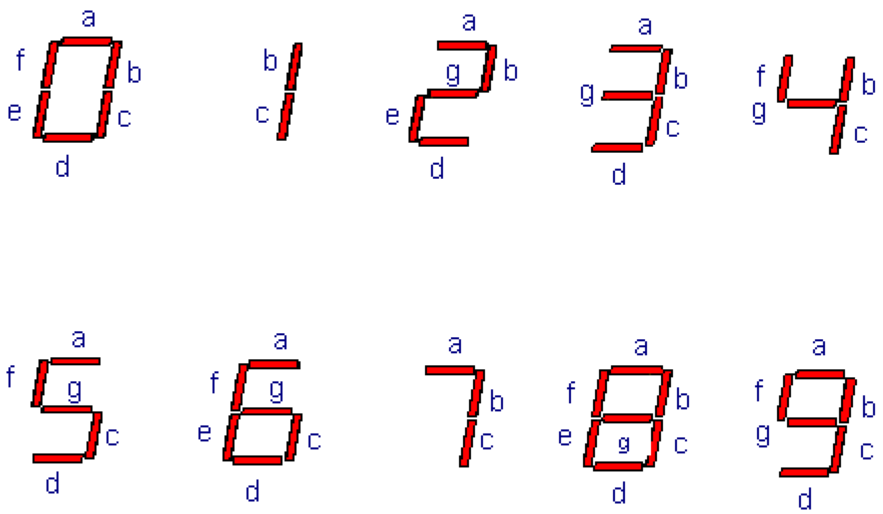
NB : Majuscules et minuscules sont inversées sur ce schéma.

2) Représentez les 10 affichages voulus selon le chiffre à afficher.
3) En déduire le tableau de Karnaugh de chaque sortie (A à G, donc 7 tableaux) sachant que « A=1 ⇔ segment A allumé ». On associera un « état indéterminé » (ϕ) aux entrées qui ne peuvent pas être représentées.
4) Simplifier au maximum ces équations par la méthode de Karnaugh, sachant que les « états indéterminés » peuvent prendre indifféremment les valeurs de 0 ou 1.

Résolution :

1) Il faut suffisamment d'entrées pour pouvoir compter jusqu'à au moins 9. Avec 3 entrées, nous pouvons compter de 0 à 2^3-1=7 < 9 : c'est insuffisant.
Avec 4 entrées, nous pouvons compter de 0 à 2^4-1=15 > 9 : c'est suffisant !
La réponse est donc **4**. Les nombres 10 à 15 seront inutilisés.
Dans le cas général pour compter de 0 à n en binaire, obtient la réponse par : $[log_2(n+1)]$

2) Ce n'est que du dessin ! ☺



NB : ici aussi, considérez qu'il s'agit de majuscules… (De A à G)

3 & 4) Grâce à la question précédente, il est facile de compléter la <u>table de vérité en binaire naturel</u> de chaque variable A à G ! (Les états associés aux nombres de 10 à 15 en binaire dans la table seront notés indéterminés : « ϕ ».)

L'utilisation des tableaux de Karnaugh permettra justement de choisir judicieusement des valeurs pour ces états indéterminés de manière à obtenir des expressions les plus simples possibles. C'est le but de la question 3 (remplir les tableaux de Karnaugh) qui permet de résoudre la question 4 (simplifier au maximum les sorties).

ABA Logique va répondre à ces deux questions en une fraction de seconde, à partir de la table de vérité que vous aurez renseignée très facilement !

En effet, commençons par la sortie A :

- Utilisez l'option « 3) Table=>Équation > » et « B : depuis table bin naturel »
- Tapez « 4 » puisqu'il y a 4 entrées.
- Tapez [B] pour pré-remplir la table de ϕ.
- Entrez dans l'ordre du dessin l'état du segment A pour chaque chiffre ! (on,off,on,on,off,on,on,on,on,on) Arrêtez-vous à la case du 9 : les suivantes sont déjà ϕ.
  Vous devriez obtenir cette table de vérité :

Table de verite ★ Binaire naturel

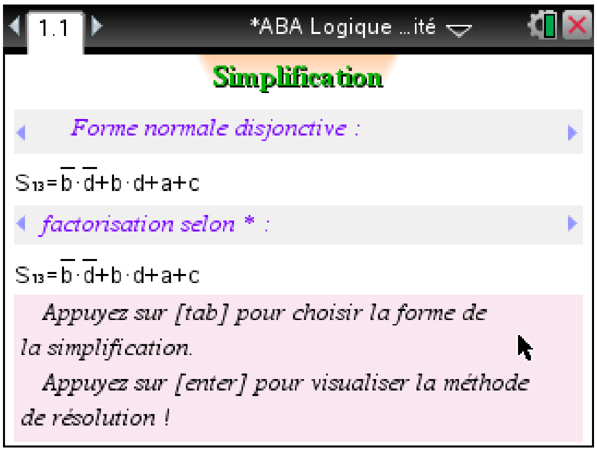
a	b	c	d	S_{12}
0	0	0	0	**1**
0	0	0	1	**0**
0	0	1	0	**1**
0	0	1	1	**1**
0	1	0	0	**0**
0	1	0	1	**1**
0	1	1	0	**1**
0	1	1	1	**1**
1	0	0	0	**1**
1	0	0	1	**1**
1	0	1	0	**Φ**
1	0	1	1	**Φ**
1	1	0	0	**Φ**
1	1	0	1	**Φ**
1	1	1	0	**Φ**
1	1	1	1	**Φ**

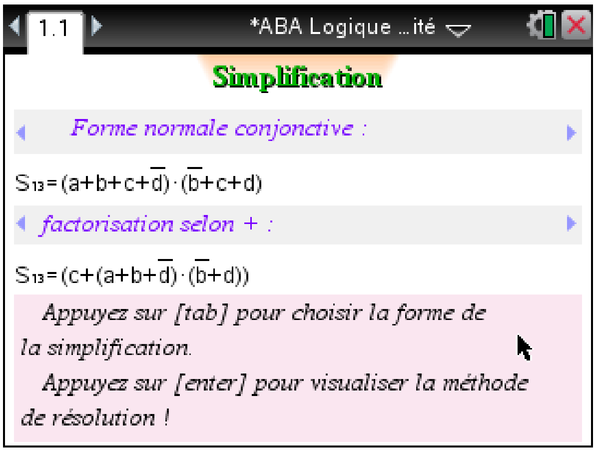
[2–6] dimension
[A] inverser tout
[B] remplir Φ
[C] Charger S_{11}

(Capture en mode « ordinateur » sur le Nspire Software.)

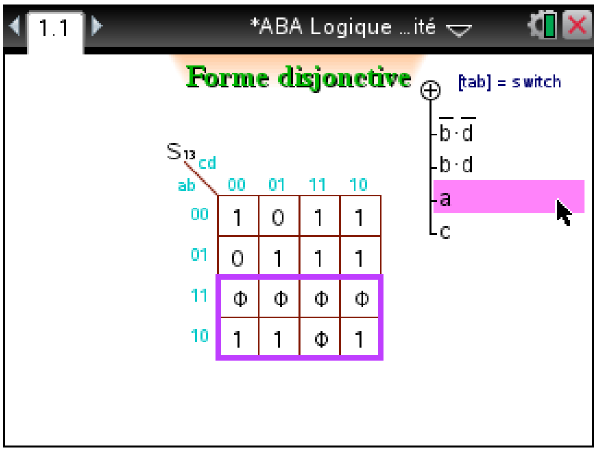
- Appuyez alors sur [enter] (ou cliquez sur le titre) et vous aurez immédiatement la simplification par tableau de Karnaugh correspondante !
  Par défaut, cette simplification est donnée en forme normale disjonctive (obtenue en regroupant les 1), mais il vous suffit d'appuyer sur [tab] pour visualiser la forme normale conjonctive (obtenue en regroupant les 0).

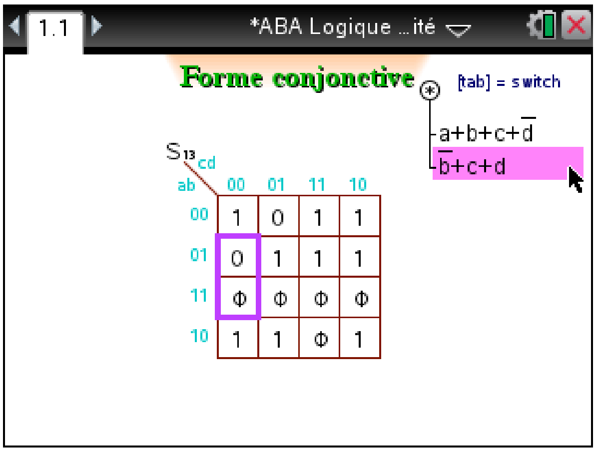
- Pour les deux méthodes, vous obtenez :





- Appuyez sur [enter] pour ensuite visualiser le tableau de Karnaugh demandé en q3 ainsi que le tracé des groupements de simplification. De même, appuyez sur [tab] pour passer d'un mode de simplification à l'autre. Vous obtenez alors respectivement :





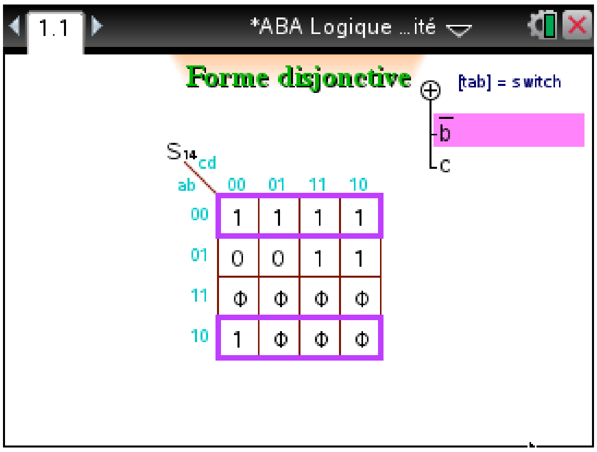
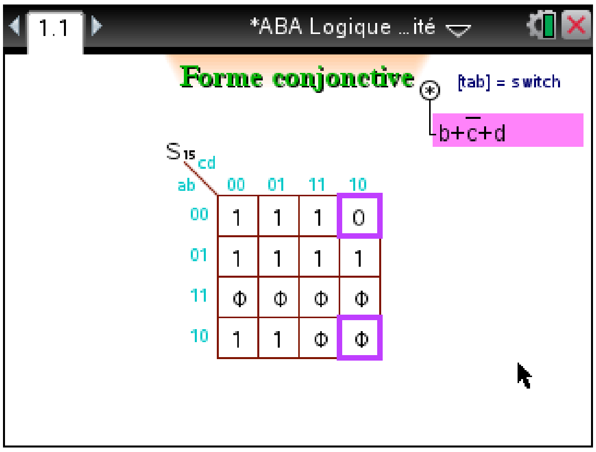
<u>**ATTENTION :**</u> lorsqu'il y a des ϕ (non, pas des « filles » ! hum…), il est possible que les deux formes ne soient pas équivalentes ! En effet, comme l'illustre cet exemple, le ϕ à la case « abcd=1100 » vaut **1** dans le cas de la forme disjonctive, alors qu'il vaut **0** dans la forme conjonctive.

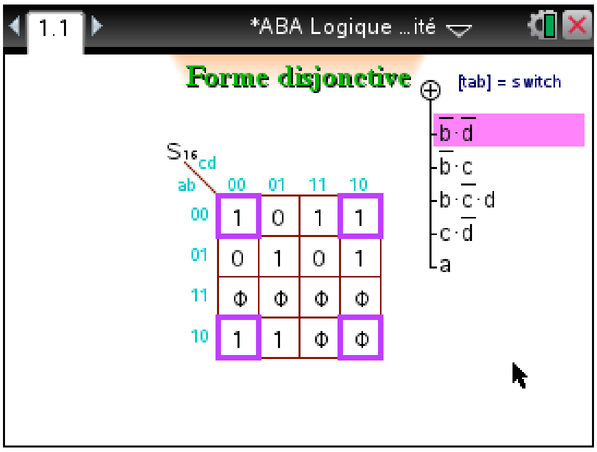
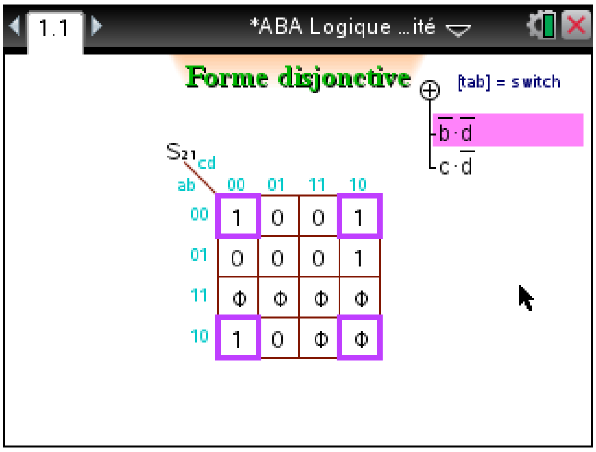
- Comment choisir la forme à donner en réponse s'il n'y a pas de contrainte sur sa forme ?
  En général dans l'enseignement, on apprend uniquement le « rassemblement des 1 » et donc c'est la forme disjonctive qui est la plus utilisée. (De plus, elle est moins lourde en parenthèses)
  Mais techniquement, il est toujours préférable d'avoir le moins de termes possible. Il faut alors choisir la forme factorisée qui présente le

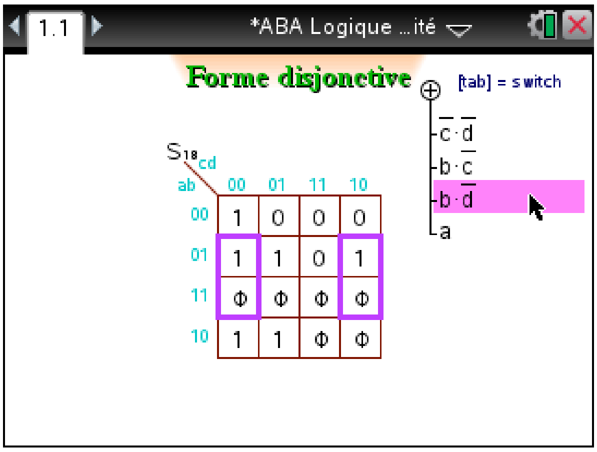
moins de terme. Ici, les deux formes factorisées ont 6 termes, donc par défaut, (et comme il y a moins de parenthèses), on prendra la forme disjonctive factorisée.

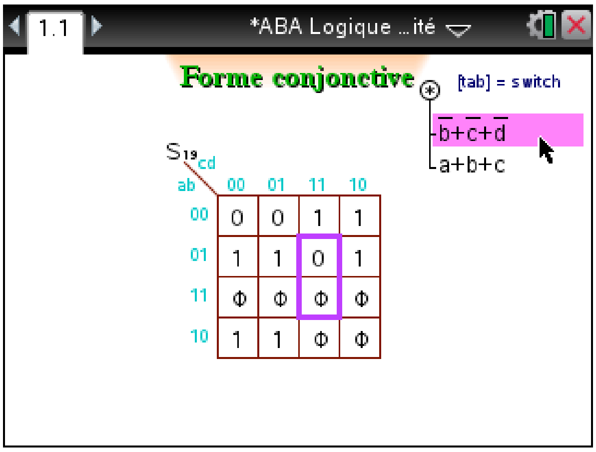
Réponse : $\boxed{\text{A = /b*/d+b*d+a+c}}$

Maintenant il ne reste plus qu'à répéter cela pour les variables B à G. Sans plus de discours, voici les résultats que l'on obtient !

	
B = /b+c	**C = b+/c+d**

	
D = /b*(/d+c)+b*/c*d+c*/d+a	**E = /d*(/b+/c)**





F = /c*(/d+b)+b*/d+a	**G = (/b+/c+/d)*(a+b+c)**

C’est dire le temps que l’on gagne sur cet exercice grâce à ABA Logique ! Cet exercice de l’afficheur 7 segment et son convertisseur est un classique, alors n’hésitez pas à le refaire et le comprendre ! ;)

Il serait intéressant ensuite de tester ces fonctions en les câblant à l’aide d’un logiciel de simulation sur un afficheur 7 segment pour vérifier les résultats.

BONUS : quels motifs obtiendrons nous pour les nombres de 10 à 15 ?

Voilà pour cette activité, je vous laisse tester et découvrir les quelques autres fonctions par vous-même ! ☺ J’espère qu’ABA Logique vous convient et vous sera utile !

Pour une explication détaillée des différentes fonctions, regardez le Lisez-moi, qui fait office de manuel en partie 4). Si vous avez des problèmes, des questions ou quelque idée d’amélioration, faites m’en part :

aphone@hotmail.fr

Ou rendez-vous sur le forum de TI Planet ! ☺

ABA Logique Nspire 2
Louis DURAND

France 54
Participation au concours Lua TI PLanet/Inspired Lua 2011