

## SYSTEMES LOGIQUES

### TRAVAIL PRATIQUE TP01 - SOLUTIONS

#### 1. TP01: CIRCUITS COMBINATOIRES ET ALGÈBRE BOOLÉENNE

Beaucoup de circuits électroniques sont disponibles comme éléments discrets. Il est alors possible d'utiliser un circuit intégré discret, qui comprend un jusqu'à typiquement huit circuits identiques en interne. Les schémas de deux de ces circuits commerciaux sont étudiés dans ce TP.

##### 1.1 LOGICIEL LOGISIM-EVOLUTION

Pour la suite du cours, il est pratique de disposer des logiciels installés sur un PC ou laptop personnel.

Toutes les questions relatives à l'utilisation de logisim-evolution sont répondues dans le tutoriel qui est disponible depuis le menu help: Help→Tutorial.

##### 1.2 CIRCUIT COMBINATOIRE, PARTIE 1

Considérons le schéma d'un circuit intégré commercial représenté en Figure 1.1 (cf. donnée).

Les tâches suivantes sont proposées:

- Etablir les expressions algébriques non-simplifiées de C1 et S1; puis,

$$S1 = \overline{C0 \cdot \overline{C1} + A1 \cdot \overline{C1} + B1 \cdot \overline{C1} + C0 \cdot A1 \cdot B1}$$

$$= \overline{C0 \cdot \overline{C1} + A1 \cdot \overline{C1} + B1 \cdot \overline{C1} + C0 \cdot A1 \cdot B1}$$

$$\overline{C1} = \overline{C0 \cdot A1 + C0 \cdot B1 + A1 \cdot B1}$$

$$C1 = C0 \cdot A1 + C0 \cdot B1 + A1 \cdot B1$$

- dresser les tables de vérité de C1 et S1, exprimées en fonction de A1, B1 et C0;

A1	B1	C0	C1	C1b	A1	B1	C0	S1
0	0	0	0	1	0	0	0	0
0	0	1	0	1	0	0	1	1
0	1	0	0	1	0	1	0	1
0	1	1	1	0	0	1	1	0
1	0	0	0	1	1	0	0	1
1	0	1	1	0	1	0	1	0
1	1	0	1	0	1	1	0	0
1	1	1	1	0	1	1	1	1

- procéder de même pour  $\overline{C2}$  et  $\overline{S2}$ , exprimés en fonction de A2, B2 et C1.

$$\overline{S2} = \overline{\overline{C1} \cdot C2 + \overline{A2} \cdot C2 + \overline{B2} \cdot C2 + \overline{C1} \cdot \overline{A2} \cdot \overline{B2}}$$

$$= \overline{C1 \cdot C2 + A2 \cdot C2 + B2 \cdot C2 + \overline{C1} \cdot \overline{A2} \cdot \overline{B2}}$$

$$\overline{C2} = \overline{\overline{C1} \cdot \overline{A2} + \overline{C1} \cdot \overline{B2} + \overline{A2} \cdot \overline{B2}}$$

$$= \overline{C1 \cdot \overline{A2} + C1 \cdot \overline{B2} + \overline{A2} \cdot \overline{B2}}$$

A2	B2	C1	C2b	C2	A2	B2	C1	S2b	S2
0	0	0	1	0	0	0	0	1	0
0	0	1	1	0	0	0	1	0	1
0	1	0	1	0	0	1	0	0	1
0	1	1	0	1	0	1	1	1	0
1	0	0	1	0	1	0	0	0	1
1	0	1	0	1	1	0	1	1	0
1	1	0	0	1	1	1	0	1	0
1	1	1	0	1	1	1	1	0	1

- Transformer le schéma logique donné en un schéma logique équivalent ne comportant que des portes NAND; dans ce cas, le raisonnement est basé sur de modifications du schéma logique.

Les quatre étapes suivantes sont appliquées au schéma logique, Figure 1.1:

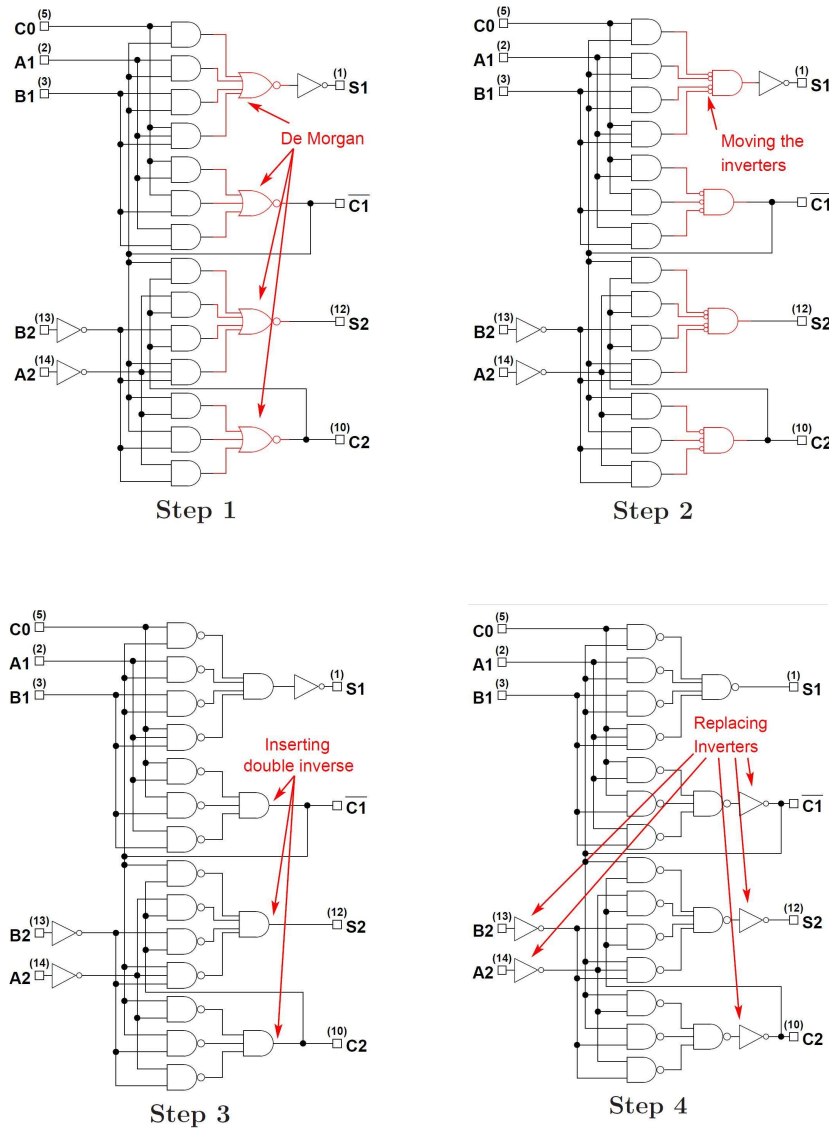


Figure 1.1: Schéma du circuit étudié et étapes de modification visant à un développement en portes NAND.

le résultat final est présenté en Figure 1.2:

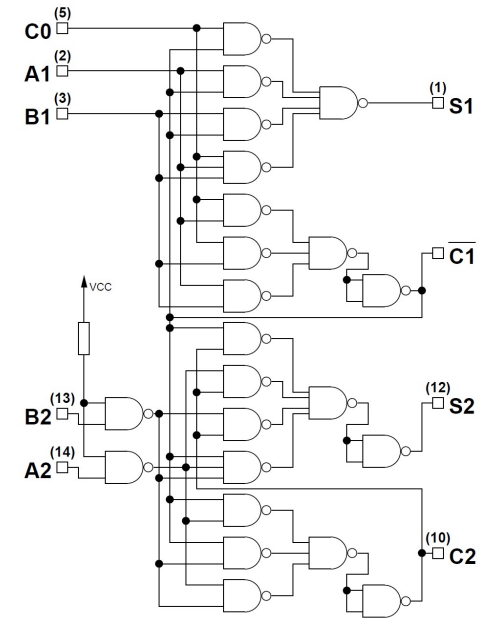


Figure 1.2: Circuit résultant équivalent en portes NAND.

- Réaliser le circuit dans logisim-evolution, et vérifier que pour tous les états possibles des variables d'entrée A1, B1 et C0, respectivement A2, B2 et C1, les variables de sortie C1 et S1, respectivement C2 et S2 satisfont aux tables de vérité précédemment établies; puis,
- déterminer la fonctionnalité du circuit.

Il s'agit d'un additionneur 2-bit; A et B sont les entrées (2-bit), S (sum) est la somme (2-bit) et C (carry) le bit de retenue (C1, 1-bit est utile en interne, et C2, 1-bit est présenté au monde extérieur).

### 1.3 CIRCUIT COMBINATOIRE, PARTIE 2

Considérons le schéma d'un circuit intégré commercial représenté en Figure 1.2 (cf. donnée).

Les tâches suivantes sont proposée:

- Etablir les expressions algébriques non-simplifiées de E1 et E0 exprimées en fonction de A1, B1, et respectivement A0 B0; puis,

$$E0 = \overline{\overline{A0} \cdot B0 + A0 \cdot \overline{B0}} = \overline{A0} \cdot \overline{B0} + A0 \cdot B0 = \overline{A0 \oplus B0}$$

$$E1 = \overline{\overline{A1} \cdot B1 + A1 \cdot \overline{B1}} = \overline{A1} \cdot \overline{B1} + A1 \cdot B1 = \overline{A1 \oplus B1}$$

- procéder de même pour ZG, ZP et ZE, exprimés en fonction de G, P, E, E1, E0, A1, A0, B1, B0;

$$ZE = E \cdot E0 \cdot E1$$

$$ZP = P \cdot E0 \cdot E1 + \overline{A0} \cdot E1 \cdot B0 + \overline{A1} \cdot B1$$

$$ZG = G \cdot E0 \cdot E1 + A0 \cdot E1 \cdot \overline{B0} + A1 \cdot \overline{B1}$$

- dresser les tables de vérité de ZG, ZP et ZE=f(A1, A0, B1, B0) pour le cas particulier où (G, P, E)=001.

Considérant que G=0 et P=0 et E=1, les dernières expressions se réduisent à la formulation suivante:

$$ZE = E0 \cdot E1$$

$$ZP = \overline{A0} \cdot E1 \cdot B0 + \overline{A1} \cdot B1$$

$$ZG = A0 \cdot E1 \cdot \overline{B0} + A1 \cdot \overline{B1}$$

Les tables de vérité aux nœuds E0 et E1 sont

A0	B0	E0	A1	B1	E1
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	1	1	1

Ainsi, les tables de vérité complètes peuvent être obtenues.

A1	B1	A0	B0	E1	E0	ZE	ZP	ZG
0	0	0	0	1	1	1	0	0
0	0	0	1	1	0	0	1	0
0	0	1	0	1	0	0	0	1
0	0	1	1	1	1	1	0	0
0	1	0	0	0	1	0	1	0
0	1	0	1	0	0	0	1	0
0	1	1	0	0	0	0	1	0
0	1	1	1	0	1	0	1	0
1	0	0	0	0	1	0	0	1
1	0	0	1	0	0	0	0	1

1	0	1	0	0	0	0	0	1
1	0	1	1	0	1	0	0	1
1	1	0	0	1	1	1	0	0
1	1	0	1	1	0	0	1	0
1	1	1	0	1	0	0	0	1
1	1	1	1	1	1	1	0	0

- Transformer le schéma logique donné en un schéma logique équivalent ne comportant que des portes NAND; dans ce cas, le raisonnement est basé sur de modifications du schéma logique.

La transformation du schéma logique partiel en Figure 1.3 peut être appliquée au sous-circuit se terminant au nœud E0 ou E1

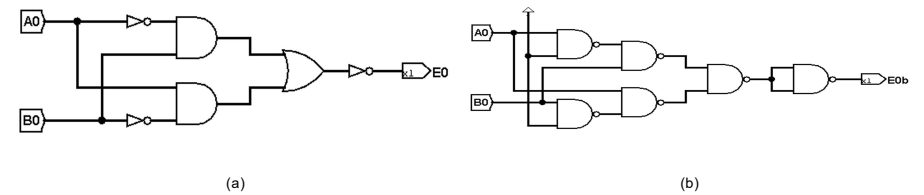


Figure 1.3: Equivalences de circuits en (a) et (b)

Le résultat peut aussi être obtenu par transformation algébrique

$$E0 = \overline{\overline{A0} \cdot B0 + A0 \cdot \overline{B0}} = \overline{A0} \cdot \overline{B0} + A0 \cdot B0 = \overline{A0 \oplus B0}$$

Le chemin couvrant la variable E1 est obtenu suivant une méthode identique.

Les manipulations algébriques suivantes permettent de déterminer les parties de sortie du circuit.

$$ZE = E \cdot E0 \cdot E1 = \overline{\overline{E \cdot E0 \cdot E1}}$$

$$\begin{aligned} ZP &= P \cdot E0 \cdot E1 + \overline{A0} \cdot E1 \cdot B0 + \overline{A1} \cdot B1 \\ &= \overline{\overline{P \cdot E0 \cdot E1 + \overline{A0} \cdot E1 \cdot B0 + \overline{A1} \cdot B1}} \\ &= \overline{P \cdot E0 \cdot E1 \cdot \overline{A0} \cdot E1 \cdot B0 \cdot \overline{A1} \cdot B1} \end{aligned}$$

$$\begin{aligned}
ZG &= \overline{G \cdot E0 \cdot E1 + A0 \cdot E1 \cdot \overline{B0} + A1 \cdot \overline{B1}} \\
&= \overline{G \cdot E0 \cdot E1 + A0 \cdot E1 \cdot \overline{B0} + A1 \cdot \overline{B1}} \\
&= \overline{G \cdot E0 \cdot E1 \cdot A0 \cdot E1 \cdot B0 \cdot A1 \cdot B1}
\end{aligned}$$

Finalement, le circuit réalisé au moyen de porte NAND est présenté en Figure 1.4:

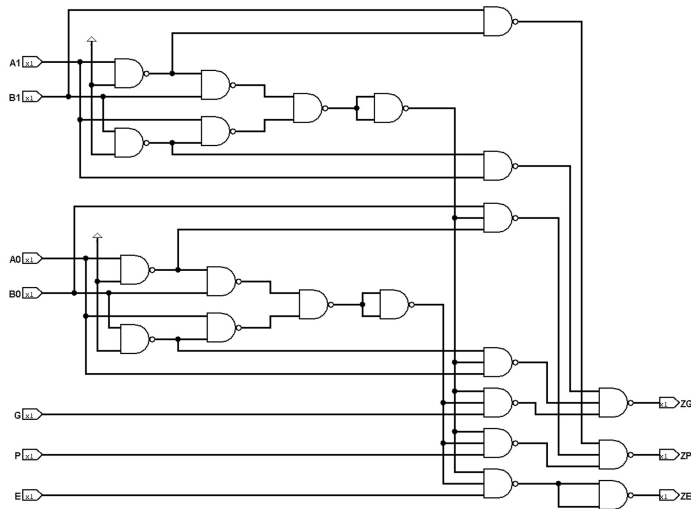


Figure 1.4: Circuit résultant équivalent en portes NAND.

- Réaliser le circuit dans logisim-evolution, et vérifier que pour tous les états possibles des variables d'entrée A1, A0, B1, B0, les variables de sortie ZG, ZP et ZE satisfont aux tables de vérité précédemment établies; puis,
- déterminer la fonctionnalité du circuit.

Il s'agit d'un comparateur d'amplitude 2-bit (magnitude comparator).

- ZE = (A==B); ZE est égal à logic-1 si la valeur A est égale à la valeur B;
- ZP = (B>A); ZP est égal à logic-1 si la valeur de A est inférieure à la valeur de B;
- ZG = (A>B); ZG est égal à logic-1 si la valeur de B est inférieure à la valeur de A.

## 1.4 CARTE DE10-LITE

Cette partie est un entraînement à la conception hiérarchique dans logisim-evolution, et à la validation sur le système-cible. Le fichier TP01S.circ disponible sur le site Moodle est un exemple.