

TP — Circuits logiques avec Cedar Logic

CEDAR Logic est un logiciel de simulation de circuits logiques combinatoires (portes) et séquentiels (bascules). Particulièrement utilisé lors de l'apprentissage des notions fondamentales pour les circuits logiques.

Il offre une interface graphique assez intuitive et une librairie de composants bien fournie, allant des portes logiques de base à des éléments de calcul, des compteurs ou différents types de mémoire. L'édition d'un circuit se fait simplement de manière graphique en sélectionnant dans la liste des composants disponibles les éléments à placer sur le circuit. La liaison se fait en tirant les fils à la souris entre les points de contacts des différents éléments.

Si Cedar Logic n'est pas installé sur votre machine, téléchargez-le via <https://sourceforge.net/projects/cedarlogic/> et installez-le.

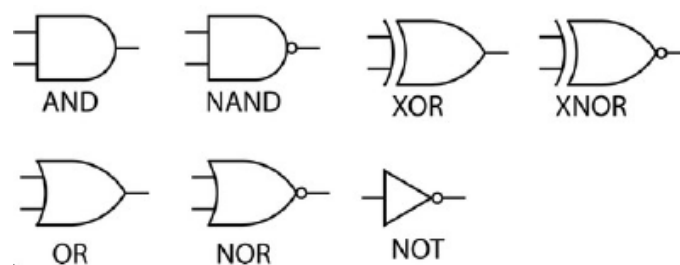
Travail à rendre

Vous devez déposer sur EPREL les fichiers Cedar Logic avec les circuits que vous avez réalisés. Ces rendus seront évalués et formeront la note pratique de la ressource.

Certains exercices nécessitent de simplifier des équations logiques, notamment les exercices 6 et 5. Les calculs et raisonnements doivent être rendus avec les fichiers Cedar Logic dans un compte rendu au format PDF (qui peut éventuellement être manuscrit et scanné). Tout autre calcul ou raisonnement à votre initiative peut être joint au compte-rendu et sera valorisé.

Prise en main de Cedar Logic

On rappelle les représentations des portes logiques usuelles, qui sont utilisées par Cedar Logic :

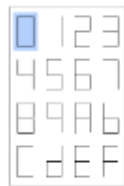


On utilisera les interrupteurs pour attribuer les valeurs aux variables logiques : si l'interrupteur est éteint, la variable vaut 0 et elle vaut 1 s'il est allumé. Pour les sorties des circuits, on utilise une LED, qui est éteinte si la sortie vaut 0 et allumée si elle vaut 1. Les interrupteurs et LED sont représentés sous Cedar Logic par les symboles suivants :



On peut utiliser un *clavier numérique 4 bits* en entrée d'un circuit, pour générer un entier en binaire entre $0000_{(2)}$ et $1111_{(2)}$ (ou entre $0_{(16)}$ et $F_{(16)}$). De même à la sortie d'un circuit, on peut utiliser un *afficheur 16 segments* pour afficher un entier entre $0_{(16)}$ et $F_{(16)}$:

Clavier numerique 4 bits



bit de poids fort

bit de poids faible

Afficheur 7 segments

bit de poids fort
bit de poids faible



Exercice 1 (Mise en route). 1) On considère les tables de vérité suivantes :

a	b	f_1
0	0	1
0	1	1
1	0	0
1	1	1

a	b	f_2
0	0	1
0	1	1
1	0	1
1	1	0

a	b	f_3
0	0	0
0	1	1
1	0	0
1	1	1

Déterminer les fonctions logiques qui réalisent f_1 , f_2 et f_3 et câbler les circuits correspondants sur Cedar Logic.

2) Même question pour les tables de vérité suivantes :

a	b	c	f_4
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

a	b	c	f_4
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

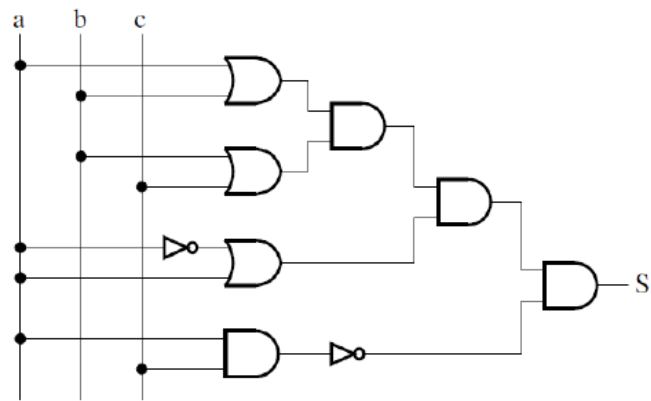
Exercice 2 (Comparateur). Un *comparateur n bits* est un circuit qui compare deux mots $a_{n-1}a_{n-2} \cdots a_1a_0$ et $b_{n-1}b_{n-2} \cdots b_1b_0$. La sortie vaut 1 si les deux mots sont identiques et 0 sinon.

- 1) Établir la table de vérité du comparateur à 1 bit. Câbler le circuit correspondant avec Cedar Logic.
- 2) En déduire un circuit réalisant un comparateur 4 bits. On utilisera deux claviers numériques 4 bits en entrée et une LED en sortie.

Circuits logiques

Dans cette partie du TP, on reprend les exercices 3 et 4 du TD n° 2.

Exercice 3. On reprend le circuit étudié à l'exercice 3 du TD :



- 1) Câbler ce circuit sur Cedar Logic.
- 2) En utilisant les réponses à l'exercice du TD, proposer un circuit qui réalise la formule simplifiée de S .

Exercice 4. On reprend les tables de vérité données à l'exercice 4 du TD :

a	b	c	S_1	S_2
0	0	0	0	1
0	0	1	0	1
0	1	0	1	0
0	0	1	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

a	b	c	d	S_3	S_4
0	0	0	0	1	1
0	0	0	1	0	0
0	0	1	0	1	1
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	1	0
0	1	1	1	1	1
1	0	0	0	0	1
1	0	0	1	0	0
1	0	1	0	0	1
1	0	1	1	1	1
1	1	0	0	0	1
1	1	0	1	0	1
1	1	1	0	0	0
1	1	1	1	1	1

Câbler sur Cedar Logic les circuits qui réalisent ces fonctions. On utilisera les formules logiques simplifiées obtenues lors du TD.

Circuits arithmétiques

On s'intéresse dans cette partie à des circuits électroniques réalisant des opérations arithmétiques usuelles : sommes, incrémentation, soustractions, etc. Dans les exercices qui suit, on se concentre uniquement sur deux opérations : la somme de deux nombres entiers, et l'opération d'incrémentation.

Exercice 5 (Additionneur 4 bits). 1) Réaliser sur Cedar Logic le circuit de l'additionneur 1 bit.

2) Concevoir un additionneur 4 bits à partir de 4 additionneurs 1 bit.

Exercice 6 (Incrémenteur 4 bits). Une (grande) partie des opérations effectuées en informatique consistent à ajouter 1 à une variable de type entier. On dit que la variable *incrémentée*. Dans cet exercice, on s'intéresse à un circuit électronique qui incrémente une variable sur 4 bits.

- 1) On note $\text{Inc}_0 : \begin{cases} \{0, 1\} \longrightarrow \{0, 1\} \times \{0, 1\} \\ a_0 \longmapsto (c_0, s_0) \end{cases}$ la fonction qui renvoie la somme $a_0 + 1$ sous la forme de couple (c_0, s_0) , où c_0 est la retenue et s_0 le chiffre (en base 2) des unités. Établir la table de vérité de Inc_0 .
- 2) Câbler sur Cedar Logic l'incrémenteur Inc_0 .
- 3) On note Inc_4 l'incrémenteur à 4 bits avec retenue. Câbler le circuit correspondant avec Cedar Logic et le tester. On utilisera un clavier numérique 4 bits en entrée et un afficheur sept segments en sortie.