# Additionneur

Christophe Viroulaud

Première - NSI

**Archi 06**

À partir des *briques élémentaires* il est possible de construire des circuits plus complexes et ainsi permettre d'effectuer différentes opérations.

Comment construire un circuit permettant d'effectuer des additions ?

# Sommaire

# Notations booléennes

- $\neg$ pour NOT
- $\wedge$ pour AND
- $\vee$ pour OR
- $\oplus$ pour XOR

| x | $\neg x$ |
|---|---|
| 1 | 0 |
| 0 | 1 |

Tableau 1 – Table de vérité de $\neg x$

| x | y | $x \vee y$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Tableau 2 – Table de vérité de $x \vee y$

**Activité 1 :** Écrire les tables de vérités de $x \wedge y$ et $x \oplus y$.

| x | y | $x \wedge y$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

| x | y | $x \oplus y$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**Activité 2 :**

1. On définit 3 paramètres : $x, y, z$. Combien de combinaisons peut-on réaliser ?
2. Écrire la table de vérité de $x \wedge y \wedge z$.

| x | y | z | $x \wedge y \wedge z$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

**Activité 3 :** Écrire la table de vérité de $x \wedge (y \vee z)$.

| x | y | z | $y \vee z$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

| x | y | z | $y \vee z$ | $x \wedge (y \vee z)$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |

**Activité 4 :** Écrire la table de vérité de
$(x \wedge y) \oplus (\neg y \vee z)$

| x | y | z | $(x \wedge y)$ | $\neg y$ | $(\neg y \vee z)$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 |

| x | y | z | $(x \wedge y)$ | $\neg y$ | $(\neg y \vee z)$ | $(x \wedge y) \oplus (\neg y \vee z)$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |

# Sommaire

# Demi-additionneur

Un demi-additionneur prend deux bits en entrée $e_0$ et $e_1$ et renvoie la somme $e_0 + e_1$ en sortie $s$. Il faut prendre en compte une éventuelle retenue $c$.

| $e_0$ | $e_1$ | s | c |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

Tableau 3 – Table de vérité du demi-additionneur

| $e_0$ | $e_1$ | s | c |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

**Activité 5 :**

1. Quelles fonctions logiques reconnaît-on en $s$ et $c$ ?
2. En déduire le schéma du demi-additionneur.

| $e_0$ | $e_1$ | s | c |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

$$s = e_0 \oplus e_1$$

$$c = e_0 \wedge e_1$$

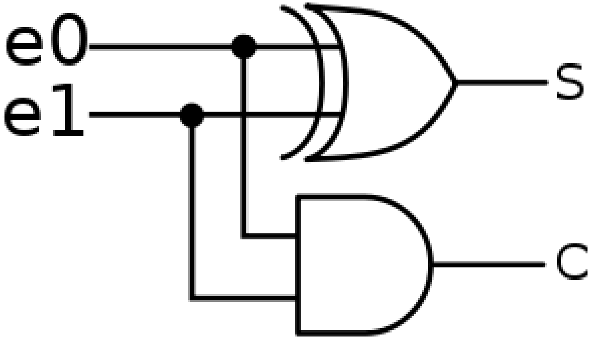


FIGURE 1 – Demi-additionneur

# Sommaire

# Additionneur



Dans une addition bit à bit il faut prendre en compte l'éventuelle retenue de l'addition précédente. Ainsi un additionneur prend trois entrées $e_0$, $e_1$ et la retenue précédente $c_0$. Il renvoie une sortie $s = e_0 + e_1 + c_0$ et une retenue éventuelle $c$.

**Activité 6 :** Compléter la table de vérité de l'additionneur.

| $e_0$ | $e_1$ | $c_0$ | s | c |
|---|---|---|---|---|
| 0 | 0 | 0 | | |
| 0 | 0 | 1 | | |
| 0 | 1 | 0 | | |
| 0 | 1 | 1 | | |
| 1 | 0 | 0 | | |
| 1 | 0 | 1 | | |
| 1 | 1 | 0 | | |
| 1 | 1 | 1 | | |

| $e_0$ | $e_1$ | $c_0$ | s | c |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

On peut remarquer :

$$s = e_0 \oplus e_1 \oplus c_0$$

$$(e_0 \land e_1) \lor (e_0 \land c_0) \lor (e_1 \land c_0)$$

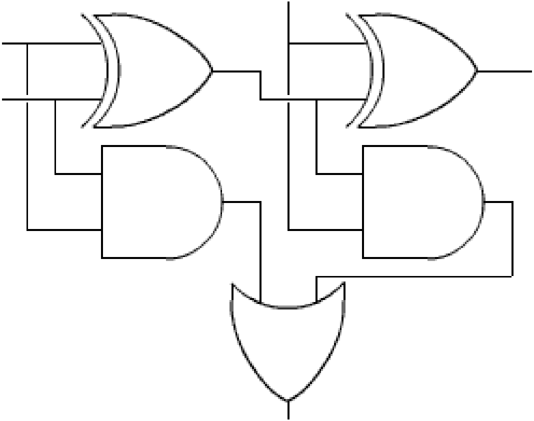

Figure 2 – Additionneur

**Activité 7 :** Placer les entrées $e_0, e_1 c_0$ et les sorties $s, c$ sur le schéma.