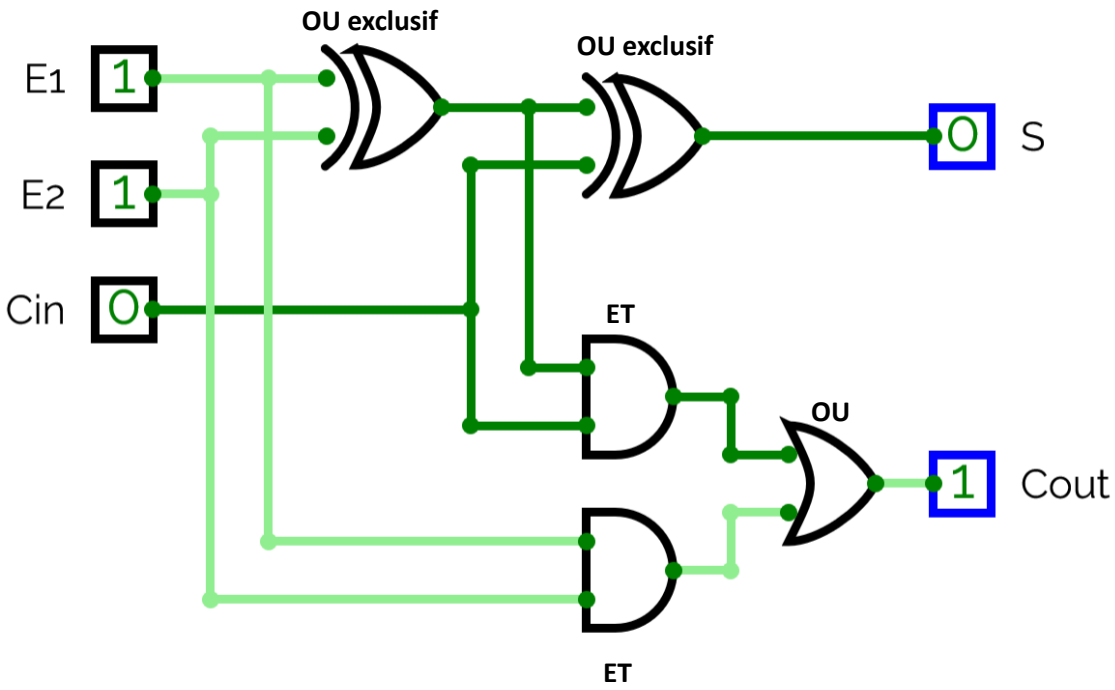
Exercices - Architecture ordinateur- OS

⇨ Répondre sur ce document papier qui est à rendre en fin d'activité.

EXERCICE 1. : ADDITIONNEUR SUR 1 BIT

Le circuit logique présenté ci-dessous est un additionneur. Il permet d'additionner 2 bits (E1 et E2) en tenant compte de la retenue entrante (« Cin » , *carry in* en anglais).

En sortie on obtient le résultat de l'addition (S) et la retenue sortante (« Cout », *carry out*).



⇨ Retrouver ce schéma sur la page : https://circuitverse.org/simulator/embed/21530

⇨ Réaliser les opérations binaires suivantes et vérifier le résultat en utilisant le simulateur :

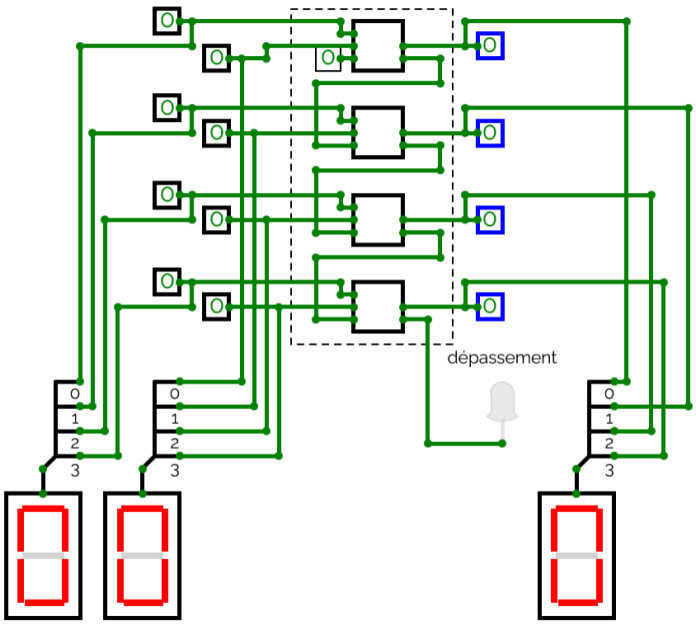
- $0 + 0 + 0 =$
- $0 + 1 + 0 =$
- $1 + 1 + 0 =$
- $0 + 0 + 1 =$
- $1 + 1 + 1 =$

EXERCICE 2. : ADDITIONNEUR SUR 4 BITS

Pour réaliser une addition sur des nombres de plusieurs bits, on combine plusieurs circuits logiques d'addition.

Remarques :

- *Le premier bit de retenue entrante vaut 0*
- *Le dernier bit de retenue sortante est « perdu » lorsqu'il y a dépassement de la capacité (longueur du mot)*



⇨ Retrouver ce schéma sur la page : https://circuitverse.org/simulator/embed/21553

⇨ quelle est la valeur décimale du nombre binaire 1111 ?

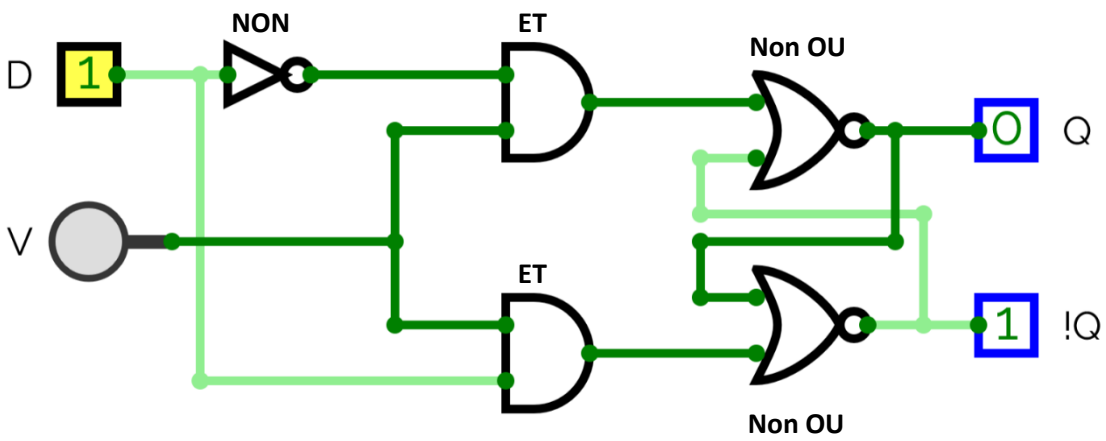
⇨ Réaliser les opérations binaires suivantes et vérifier le résultat en utilisant le simulateur :

- $0010 + 0010 =$
- $0100 + 0100 =$
- $1000 + 1000 =$
- $11111 + 1111 =$

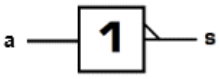
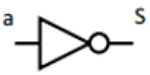
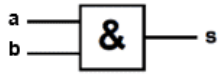
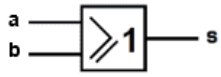
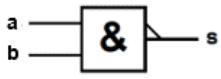
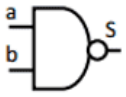
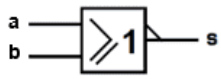
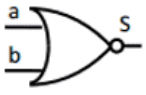
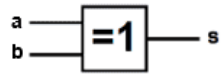
EXERCICE 3. : MEMOIRE VIVE SUR 1 BIT

Il existe plusieurs circuit logiques, appelés **bascules**, permettant de mémoriser des *bits*. On donne l'exemple ci-dessous d'une bascule D à verrouillage :

L'entrée D correspond au *bit* à mémoriser. Pour effectuer la mémorisation, il faut activer le bit de validation V. La sortie Q prend alors la valeur de D à l'instant de la validation, lorsque V passe de 0 à 1 (on parle de **front montant** de V). Tant que V reste à 0, Q garde la valeur mémorisée, quelle que soit la valeur de D.

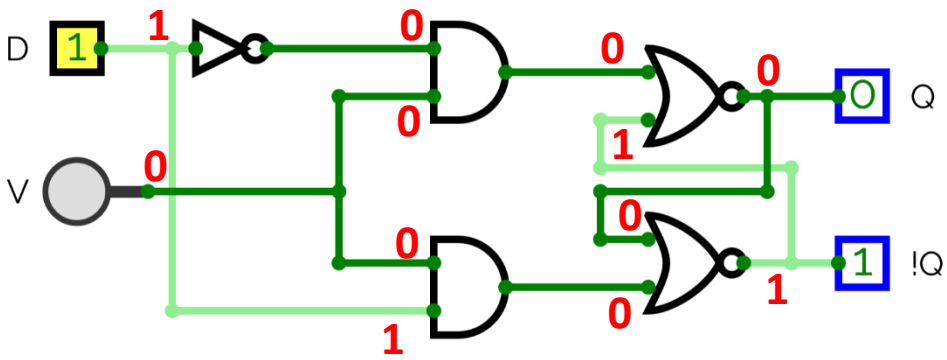


On rappelle ci-dessous les tables de vérité des principales fonctions logiques utilisées :

Nom - équation	Norme IEEE	Norme ANSI	Table de vérité
NON (NO) $S = \overline{a}$			a | S 0 | 1 1 | 0
ET (AND) $S = a.b$			a | b | S 0 | 0 | 0 0 | 1 | 0 1 | 0 | 0 1 | 1 | 1
OU (OR) $S = a + b$			a | b | S 0 | 0 | 0 0 | 1 | 1 1 | 0 | 1 1 | 1 | 1
NON ET (NAND) $S = \overline{a.b}$			a | b | S 0 | 0 | 1 0 | 1 | 1 1 | 0 | 1 1 | 1 | 0
NON OU (NOR) $S = \overline{a+b}$			a | b | S 0 | 0 | 1 0 | 1 | 0 1 | 0 | 0 1 | 1 | 0
OU EXCLUSIF (XOR) $S = a \oplus b$ $= \overline{a}.b + a.\overline{b}$			a | b | S 0 | 0 | 0 0 | 1 | 1 1 | 0 | 1 1 | 1 | 0

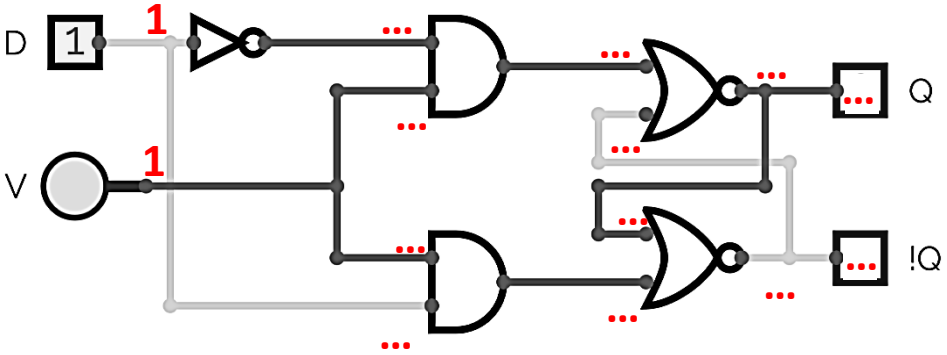
⇨ Retrouver ce schéma sur la page : https://circuitverse.org/simulator/embed/21556

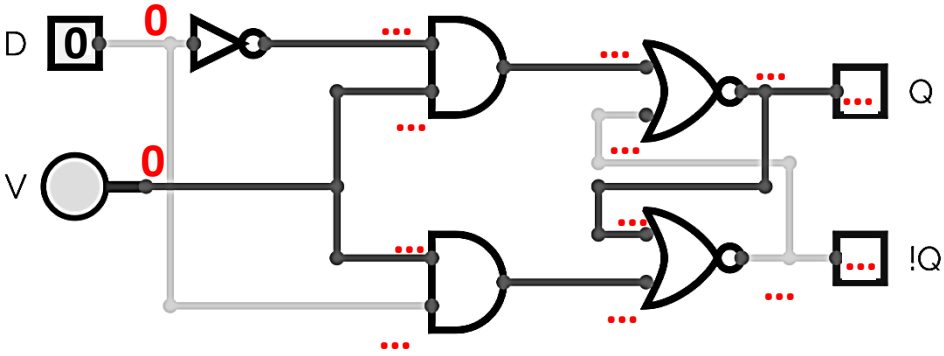
Pour chaque fil de ce schéma logique, on indique 1 si le fil est alimenté électriquement et 0 sinon pour une situation ou initialement $D = 1$, $V = 0$ et $Q = 0$.



⇨ Compléter les états logiques des fils lorsqu'on modifie l'état logique de V : $V = 1$:

En basculant l'état de V à 1, on a pu modifier la valeur de D dans Q.



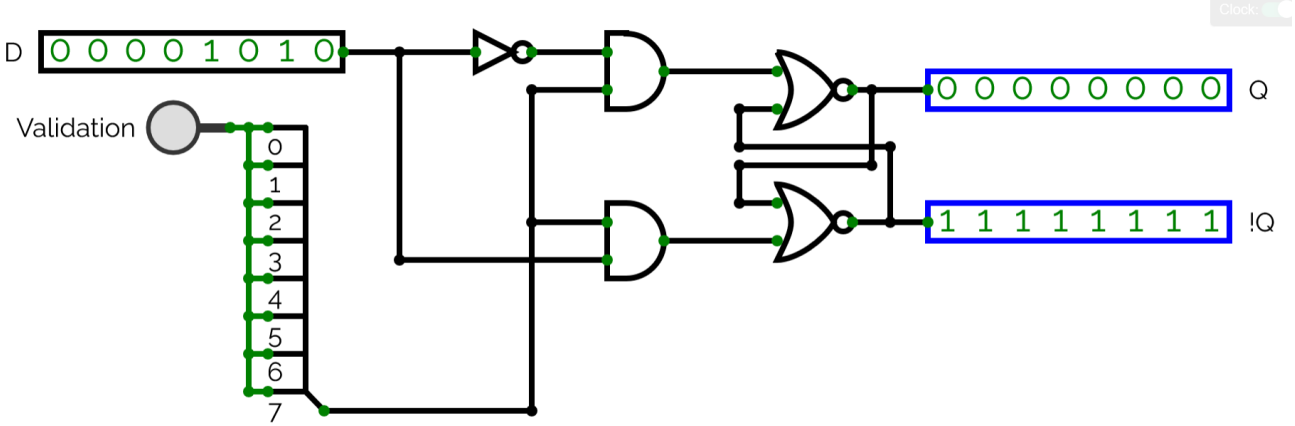


⇨ Compléter à présent les états logiques des fils lorsqu'on modifie l'état logique de D : $D = 0$ avec V qui reste à 0 : $V = 0$ et Q qui conserve initialement la valeur précédente : $Q = 1$

La valeur de Q reste bien à a valeur, tant que l'impulsion V n'a pas été réalisée.

EXERCICE 4. : MEMOIRE VIVE SUR 8 BITS

Les circuits intégrés de mémoire vive comportent de très nombreuses bascules, accessibles par une **adresse** donnant accès à 8 *bits* (soit un octet) à la fois, en lecture ou en écriture. On donne en exemple ci-dessous une bascule D à verrouillage pour 1 octet :



⇨ Retrouver ce schéma sur la page : https://circuitverse.org/simulator/embed/21557

⇨ Tester ce circuit logique. Est-il capable de mémoriser 1 octet ?