




# Séance 4 : Circuits logiques

## Corrigé

### 1 Introduction

Les passages nécessitant des pré-déterminations ou des réflexions théoriques sont indiqués par un symbole  dans la marge, ceux nécessitant de manipuler du matériel par le symbole  et les passages informatif par .

#### 1.1 But de la manipulation et objectifs d'apprentissage

Cette manipulation a pour but d'illustrer :

- au niveau «application» : le fonctionnement de circuits logiques câblés
- au niveau «composant» : le fonctionnement de circuits logiques discrets.

À la fin de ce laboratoire, vous devez être capable de :

- donner la table de vérité d'un circuit logique ;
- réaliser et tester un circuit logique ;
- comprendre la notion de temps de propagation ;
- comprendre le fonctionnement d'une bascule R-S ;
- réaliser et tester une bascule R-S et une bascule D à l'aide de portes logiques.

#### 1.2 Prérequis

Chapitres n° 24 et n° 25 du livre de référence (ed 5).

En particulier :

- algèbre de Boole
- portes logiques, états logiques
- logique combinatoire
- délai de propagation
- logique séquentielle : bascules R-S et D.

### 1.3 Matériel

Composant	Valeur	Quantité
74HC00		x1
74HC32		x1
74HC04		x1
Interrupteur		x3
Résistance	330	x2
LED TLHR5400		x2

### 1.4 Prédéterminations

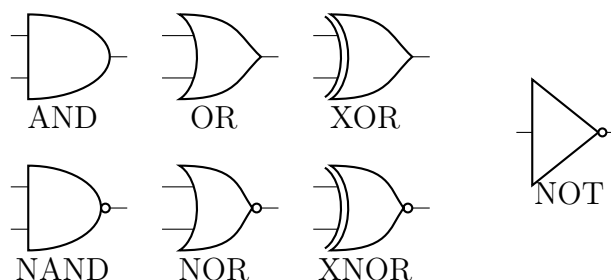
Les questions 1, 2, 3, 6, 12 à 18 et 22 doivent être faites **avant** l'arrivée au laboratoire.

## 2 Concepts

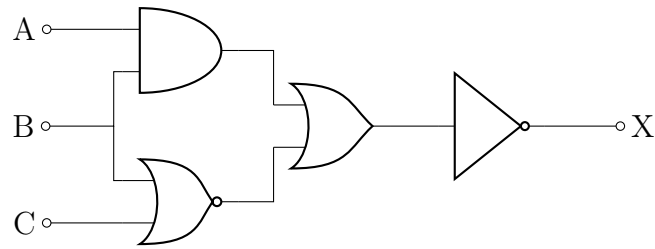
Les circuits logiques sont invisibles mais fortement présent autour de nous : ordinateurs, téléphones, tablettes numériques... Même si la complexité des circuits a fortement augmenté depuis quelques décennies, les fonctions de base sont restées essentiellement identiques. Ce labo a pour but de vous montrer comment réaliser les fonctions de base de la logique numérique avec des circuits conçus à cet effet.

## 3 Logique combinatoire

Les circuits décrits dans cette section sont **combinatoires**, c'est à dire que leur sortie ne dépend que de l'état des entrées.  
Les opérations logiques usuelles sont :



**Question 1.** Donnez la table de vérité du circuit suivant :

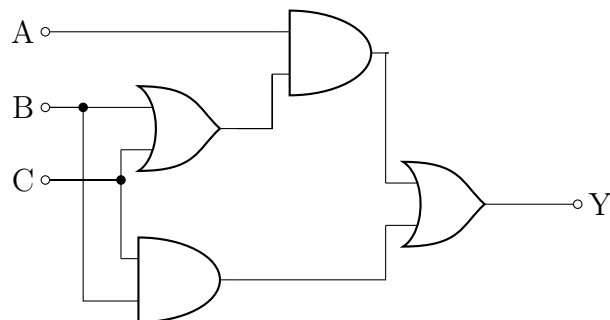


Réponse :

A	B	C	X
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	0
1	1	1	0



**Question 2.** En utilisant une table de vérité, démontrez que ce circuit est un « voteur » :



Réponse :

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



**Question 3.** Comment représenter une porte logique AND à l'aide de deux portes NAND ?

Réponse :



**Question 4.** Réalisez ce circuit sur le protoboard au moyen de circuits intégrés 74HC00 (NAND) et 74HC32 (OR) (sans utiliser les portes NOT), les brochages se trouvent en annexe.

- Branchez les entrées sur les leds pour visualiser leur état.
- Branchez les sorties intermédiaires sur les leds pour vérifier progressivement votre câblage.
- Branchez la sortie sur une led pour vérifier son état.
- **N'oubliez pas d'alimenter vos circuits logiques en 5V !**

**Astuce**

Placez bien vos portes logiques *à cheval* sur la travée du protoboard. N'oubliez pas de connecter une résistance en série de votre LED pour la protéger !

Réponse :



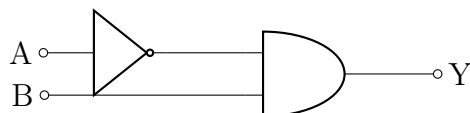
**Question 5.** Testez son bon fonctionnement à l'aide des LEDs et des interrupteurs. Vérifier si sa table de vérité correspond à celle déterminée précédemment.

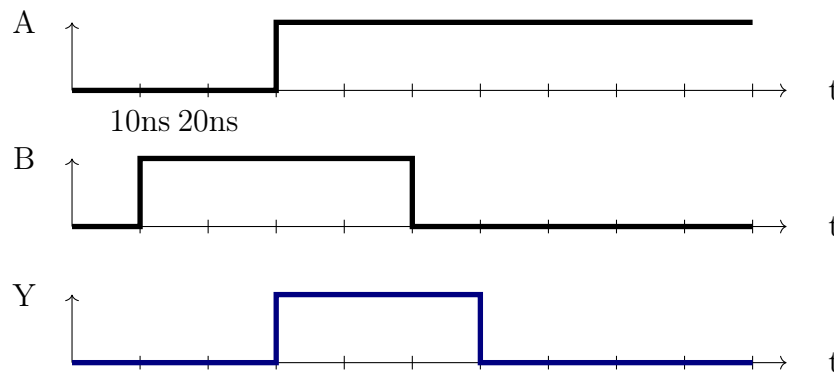
Réponse :

### 3.1 Temps de propagation des signaux



**Question 6.** Complétez le chronogramme ci-dessous en considérant que la porte NOT a un temps de propagation de 10ns et que la porte AND a un temps de propagation de 20 ns.



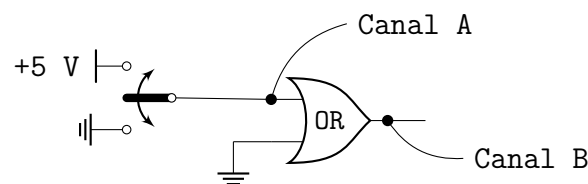


**Réponse :**



Vous allez à présent mesurer le temps de propagation de l'une de vos portes OR à l'aide de l'oscilloscope. À cette fin, vous allez brancher une sonde à la sortie de l'interrupteur et une seconde à la sortie de la porte logique. Lorsqu'elle changera de l'état bas à l'état haut, vous pourrez observer l'écart entre les deux signaux qui est dû au délai de propagation du composant.

Voici le montage que vous devez réaliser :




Configurez le logiciel du Picoscope de la façon suivante :

Paramètre	Réglage	Commentaire
Trigger	Single	On ne veut faire qu'une seule acquisition.
Trigger Channel	A	C'est au changement d'état du canal A que l'acquisition doit démarrer.
Rising Edge		On veut observer une transition de l'état bas (0 V) vers l'état haut (5 V).
Threshold	3 V	Lorsque le canal A atteint 3 V sur un flanc montant, l'acquisition doit être déclenchée.
Pre-Trigger	50 %	Le déclenchement de l'acquisition sera au milieu de la fenêtre : 50 % avant et 50 % après.
Échelle temporelle	200 ns/div	

Vous devriez voir le bandeau suivant en bas de votre fenêtre :



**Question 7.** Préparez votre prise de mesure :

- Placez l'interrupteur en position « 0 V ».
- Appuyez sur « Go » dans le logiciel : 
- Passez l'interrupteur à « 5 V ».

Mesurez le délai de propagation lorsque les signaux passent à 2.5 V.

**Astuce**

Veillez à utiliser les mêmes types de sonde pour les deux canaux afin de limiter les écarts de mesure.

**Réponse :**

**Question 8.** Quelle est la valeur *typique* renseignée dans les caractéristiques dynamiques de la fiche technique du 74HC32 ?

**Astuce**

Aidez-vous des chronogrammes schématisés se trouvant dans la fiche technique, ils expliquent les acronymes utilisés.

**Réponse :**

6 ns à 25°C et avec un CL de 15pf. On peut trouver l'input capacitance de la sonde de l'oscilloscope dans la datasheet du picoscope 2204A, elle est de 1Mohm//15pF +- 2pF.

**Question 9.** Quelle est le plus petit intervalle de temps que vous pouvez mesurer avec le picoscope 2204A ?

**Astuce**

Vous trouverez la réponse dans sa fiche technique. On y indique notamment qu'il peut capturer 100 millions d'échantillons par seconde (100MS/s), lorsqu'un seul canal est utilisé. Si les deux canaux sont utilisés simultanément, on descend à 50MS/s.

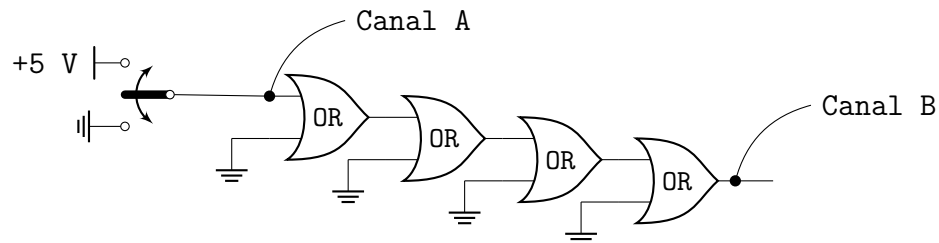
**Réponse :**

10 ns

**Question 10.** À la lumière de ces résultats, critiquez la pertinence de votre mesure à la question 7.

**Réponse :**

**Question 11.** En supposant les délais de propagation identiques entre les différentes portes logiques d'un même package 74HC32, connectez quatre portes OR en série et calculez le temps de propagation moyen à l'aide d'une nouvelle mesure.



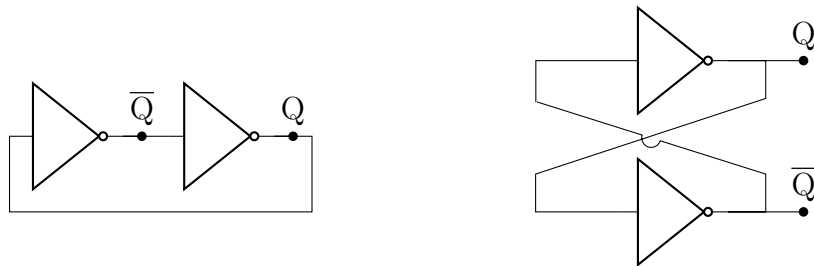
**Réponse :**

## 4 Logique séquentielle, mémorisation

Les bistables sont des circuits très employés en électronique numérique en raison de leur multiples applications. Leur première fonction est de mémoriser une information logique. Les bistables sont des circuits **séquentiels**, c'est à dire que leur sortie dépend des entrées **et** de l'état précédent du système. Contrairement aux circuits combinatoires de la section précédente, l'état actuel du système dépend de son passé.

### 4.1 Bistable élémentaire

Ci-dessous est représenté sous 2 formes différentes le bistable le plus simple.



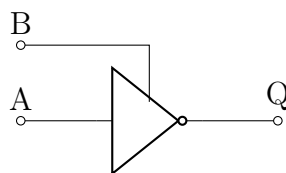
Ce bistable possède 2 états stables (d'où son nom) :

- soit  $Q$  est à l'état haut (1) et  $\overline{Q}$  est à l'état bas (0)
- soit  $Q$  est à l'état bas (0) et  $\overline{Q}$  est à l'état haut (1)

Ce circuit est donc une mémoire. Malheureusement, il est difficile de modifier son état autrement qu'en court-circuitant une des sortie à un état déterminé : il n'y a en effet pas d'entrée à ce circuit.

### 4.2 Amélioration

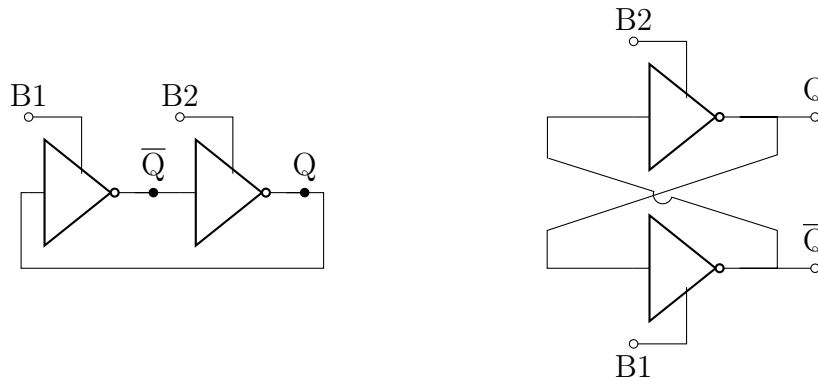
Imaginons que l'on dispose d'inverseurs dont on peut mettre la sortie dans un état déterminé. Par exemple l'inverseur suivant :



- si  $B$  est à l'état bas, alors  $Q$  est l'inverse de  $A$  (inversion normale de  $A$ )
- si  $B$  est à l'état haut, alors  $Q$  est à l'état bas (par exemple)

On peut alors réaliser le montage suivant :





On peut maintenant réaliser les opérations suivantes :

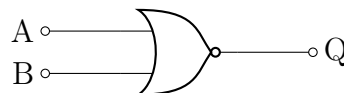
- mettre l'entrée B2 à l'état haut et l'entrée B1 à l'état bas : la sortie Q est alors mise à l'état bas.
- mettre l'entrée B2 à l'état bas et l'entrée B1 à l'état haut : la sortie Q est alors mise à l'état haut.
- mettre les 2 entrées B1 et B2 à l'état bas : le circuit reste dans son état précédent : c'est la mémorisation.

Le cas où les 2 entrées B1 et B2 sont à l'état haut n'est pas intéressant : il force les 2 sorties Q et  $\bar{Q}$  à l'état bas. C'est un cas indésirable.

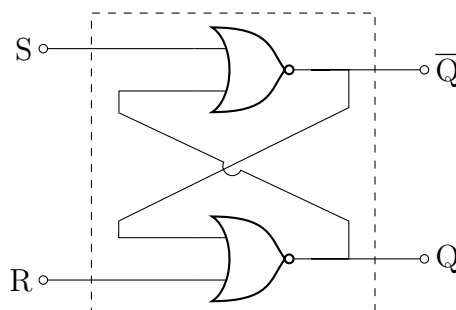
### 4.3 Bistable RS

Vous connaissez déjà des éléments logiques qui répondent à l'inverseur spécial vu ci-dessus.

Prenons par exemple la porte NOR (Non OU).



Cette porte répond parfaitement à la table de vérité de l'inverseur spécial : si son entrée B est à l'état haut, la sortie Q est forcée à l'état bas ; sinon, la sortie Q est l'inverse de l'entrée A. Construisons alors le bistable à partir de ces portes et renommons (S, R) les entrées (B1, B2).



- Pour mettre la sortie Q à l'état bas, il suffit d'appliquer un état haut à l'entrée R (Reset : mise à l'état bas) et un état bas à l'entrée S (Set : mise à l'état haut)

- Pour mettre la sortie  $Q$  à l'état haut, il suffit d'appliquer un état haut à l'entrée  $S$  (Set : mise à l'état haut) et un état bas à l'entrée  $R$  (Reset : mise à l'état bas)
- Pour être en mémorisation, il suffit d'appliquer un état bas aux 2 entrées.

Si on appelle l'état bas le niveau inactif et l'état haut le niveau actif, alors on peut dire :

- pour mettre le bistable dans un état déterminé, il suffit d'appliquer un niveau actif à l'entrée correspondante.
- pour mettre le bistable en mémorisation, il suffit de ne mettre aucun niveau actif
- l'état indésirable a lieu lorsqu'on demande au bistable en même temps une mise à l'état haut et une mise à l'état bas.

#### 4.4 Table de vérité du bistable R-S



Si on n'est pas convaincu par les développements précédents, on peut écrire la table de vérité du bistable R-S :

R	S	$Q_n$	$\overline{Q_n}$	$Q_{n+1}$	$\overline{Q_{n+1}}$
0	0	0	0	?	?
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	interdit	
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	interdit	
1	0	0	0	0	1
1	0	0	1	0	1
1	0	1	0	0	1
1	0	1	1	interdit	
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	interdit	

En entrée de la table, on met les 2 sorties stables à l'instant  $t$  et les entrées que l'on applique à ce même instant. En sortie de la table, on détermine l'état des 2 sorties à l'instant  $t+t_p$ , c'est-à-dire après le temps de propagation du bistable. Cette table de vérité peut être simplifiée, en effet, les états interdits ne sont pas à prendre en considération puisqu'ils ne seront jamais rencontrés. Il ne faut cependant pas confondre état interdit en entrée (lignes « interdits ») et état indéterminé en sortie (ligne 1). La table devient :

R	S	$Q_n$	$\overline{Q_n}$	$Q_{n+1}$	$\overline{Q_{n+1}}$
0	0	0	0	?	?
0	0	0	1	0	1
0	0	1	0	1	0
0	1	X	X	1	0
1	0	X	X	0	1
1	1	X	X	0	0

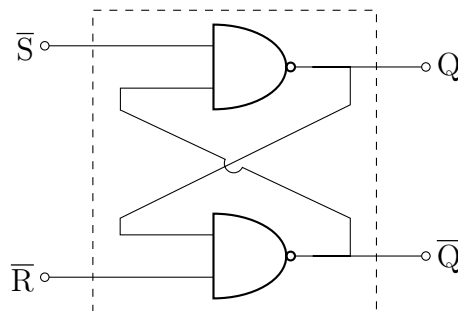
Le dernier état est évidemment indésirable puisqu'il rend égales des sorties normalement complémentaires (l'ordre donné au bistable est incohérent). Si on évite cet état, on peut supprimer la dernière ligne du tableau ainsi que la première. On peut alors simplifier cette table pour avoir :

R	S	$Q_{n+1}$	$\overline{Q_{n+1}}$
0	0	$Q_n$	$\overline{Q_n}$
0	1	1	0
1	0	0	1

De cette table, on retire heureusement les mêmes conclusions que précédemment. La méthode par table de vérité est assez lourde dans son entièreté. Il vaut donc mieux essayer d'établir directement la table simplifiée comme vu précédemment.

## 4.5 Réalisation du bistable R-S

Soit le circuit suivant :



**Question 12.** Quel est l'état des entrées qui assure la fonction de mémorisation ?

**Réponse :**

$\overline{R}, \overline{S} = (1, 1)$



**Question 13.** Que faut il appliquer au circuit pour réaliser un Set ou Reset ?

**Réponse :**

	$\overline{S}$	$\overline{R}$
Set :	0	1
Reset :	1	0



**Question 14.** Quelle est l'anomalie des sorties lorsque les deux entrées sont à l'état bas ?

**Réponse :**

$Q = \overline{Q} = 1$ , alors que les deux sorties devraient être complémentaires.



**Question 15.** Donnez à chaque valeur des entrées  $(\overline{R}, \overline{S})$  une appellation parmi les suivantes :

- mémorisation
- mise à 0
- mise à 1
- indésirable

**Réponse :**

- mémorisation :  $(\overline{R}, \overline{S}) = (1, 1)$
- mise à 0 :  $(\overline{R}, \overline{S}) = (0, 1)$
- mise à 1 :  $(\overline{R}, \overline{S}) = (1, 0)$
- indésirable :  $(\overline{R}, \overline{S}) = (0, 0)$



**Question 16.** Justifiez appellation R-S. Quel est le niveau logique actif de ces 2 entrées ?

**Réponse :**

(R : Reset S : Set) et dans la configuration proposée (avec des portes NAND), les entrées sont actives à l'état bas.



**Question 17.** Pourquoi parle-t-on d'entrées en logique inverse ?

**Réponse :**

La description habituelle d'un bistable R-S part de la description avec des portes NOR. Si le bistable est réalisé avec des portes NAND, toute la logique est **inversée**.



**Question 18.** Que pourrait-il se passer si on passe de l'état [mise à 1] à l'état [mémo-risation] en passant transitoirement par l'état [indésirable] ?

**Réponse :**

Les deux sorties seraient actives en même temps. Tant que cette situation est *transitoire*, ce n'est pas un problème majeur.



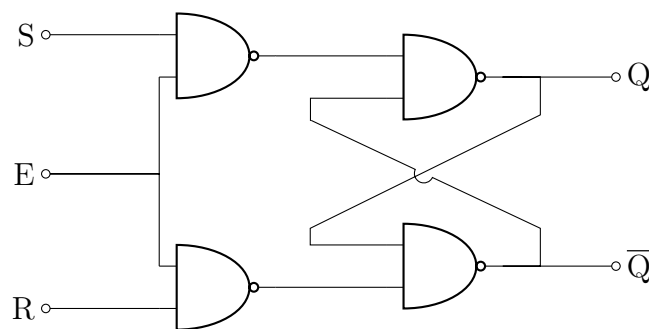
**Question 19.** Réalisez le circuit ci-dessus sur le protoboard et vérifiez vos conclusions des questions 12 à 18.

**Réponse :**

## 4.6 Bistable R-S avec entrée d'activation (Enable)



Modifiez le schéma précédent comme suit :



**Question 20.** Mettez en évidence l'avantage de l'entrée enable du point de vue des modifications des entrées R et S.

**Réponse :**

**Question 21.** Constatez que vous pouvez encore atteindre l'état « interdit »  $Q = \overline{Q}$ .

### Astuce

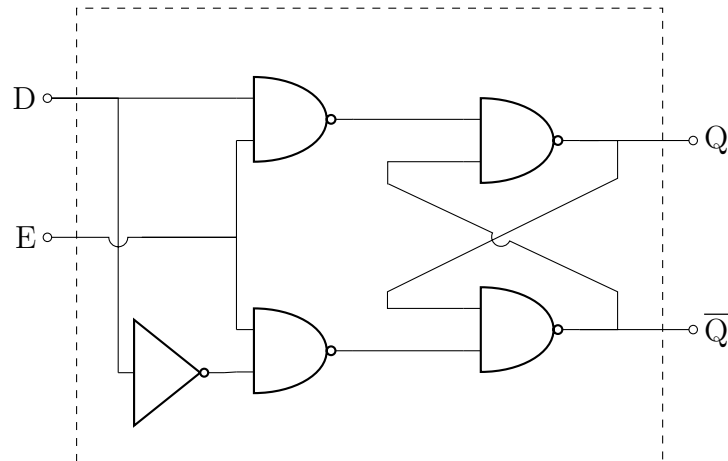
On parle d'état « interdit » parce que les sorties  $Q$  et  $\overline{Q}$  doivent être complémentaires (opposées) par convention.

**Réponse :**

## 4.7 D latch



Le D latch peut être réalisé avec le schéma suivant :



L'inverseur en tête du montage a pour effet de supprimer l'état illicite du bistable R-S de sortie.



**Question 22.** Déterminez son fonctionnement (table de vérité simplifiée).

Réponse :

D	E	$Q_{n+1}$	$\overline{Q_{n+1}}$
0	0	$Q_n$	$\overline{Q_n}$
0	1	0	1
1	0	$Q_n$	$\overline{Q_n}$
1	1	1	0



**Question 23.** Réalisez le D LATCH à l'aide de portes logiques. Le circuit 74HC04 contient 6 inverseurs (portes NOT). Vérifiez que votre circuit reproduit bien le fonctionnement attendu.

Réponse :



**Question 24.** Peut-on, à votre avis, modifier D n'importe quand sans peur de provoquer une sortie « interdite » ?

Réponse :

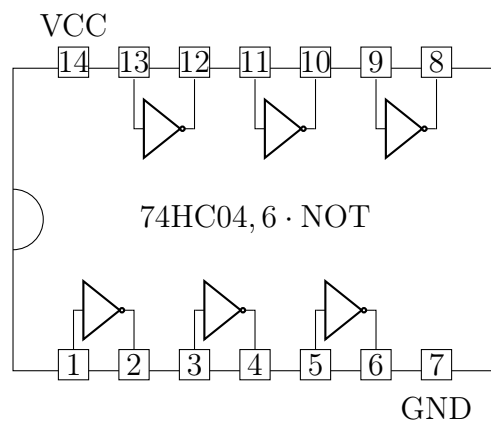
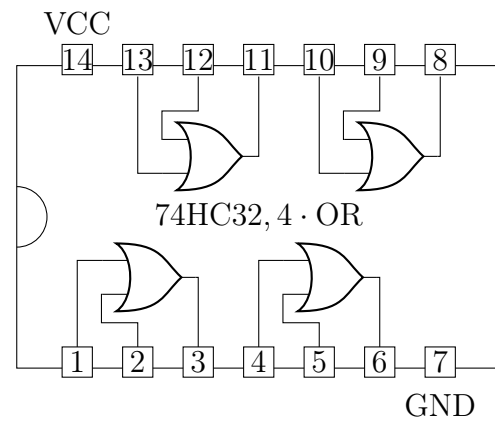
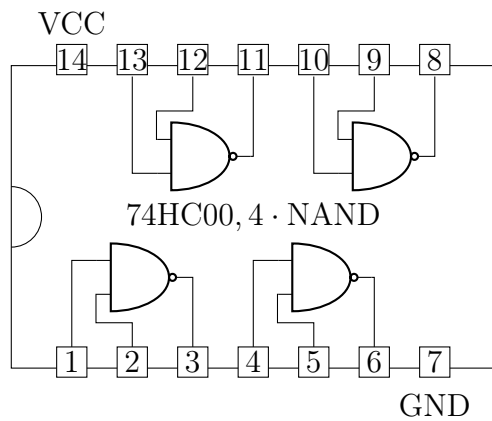
Si on considère la porte logique NOT comme non idéale (avec un temps de propagation non nul), alors il est encore possible d'atteindre l'état de sortie indésirable. En considérant :

- une transition de B de l'état bas vers l'état haut au moment  $t_D$
- un délai de la porte NOT  $\Delta t_{NOT}$

— le prochain rising edge de l'enable en  $t_{\text{en}} > t_{\text{D}}$

Il est alors nécessaire que  $t_{\text{D}} + \Delta t_{\text{NOT}} < t_{\text{en}}$  pour éviter la situation instable.

## A Brochages



Pour information, les documentations complètes :

- 74HC00 (Quad NAND) [https://assets.nexperia.com/documents/data-sheet/74HC\\_HCT00.pdf](https://assets.nexperia.com/documents/data-sheet/74HC_HCT00.pdf)
- 74HC04 (Hex NOT) [https://assets.nexperia.com/documents/data-sheet/74HC\\_HCT04.pdf](https://assets.nexperia.com/documents/data-sheet/74HC_HCT04.pdf)
- 74HC32 (Quad OR) [https://assets.nexperia.com/documents/data-sheet/74HC\\_HCT32.pdf](https://assets.nexperia.com/documents/data-sheet/74HC_HCT32.pdf)