

TP 1

+ Réalisation de fonctions logiques simples à l'aide des opérateurs logiques NAND, NOR et XOR.



‡ Rôle des LED (Diodes électroluminescentes) et des interrupteurs.

- Les opérateurs utilisés fonctionnent en logique positive.

Valeur logique	Niveau logique	Potentiel Électrique
1	HAUT	+ 5V
0	BAS	Masse

- Toute entrée d'un opérateur logique non connecté est considérée comme un niveau indéterminé.
- Les interrupteurs permettent de fixer les niveaux logiques des variables d'entrée.
- Les LED permettent de visualiser les niveaux logiques de sorties.



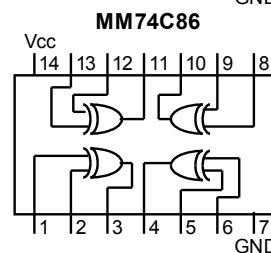
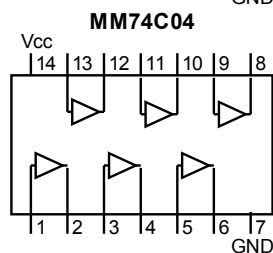
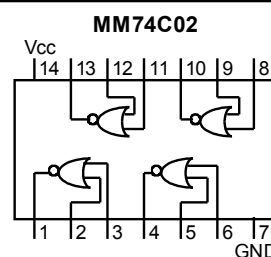
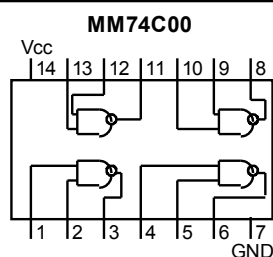
‡ Description des circuits intégrés utilisés.

MM74C00 quad two-input NAND gate

MM74C02 quad two-input NOR gate

MM74C04 hex inverter

MM74C86 quad two-input EXCLUSIVE-OR gate





À partir de maintenant, on utilisera le simulateur **ktechlab** pour réaliser les circuits mais auparavant il faut exécuter la commande suivante dans une interface de commande pour installer les circuits utilisés :

```
> ~casse/L2-ARCHI/install-tp1.sh
```



En utilisant les tables de vérité correspondantes, vérifier matériellement, sur la station mise à votre disposition, le bon fonctionnement des circuits suivants :

- 74C04
- 74C00
- 74C02
- 74C86

On teste en circuit en vérifiant qu'il se comporte comme prévu selon sa table de vérité :

- on place des interrupteurs sur chaque entrée pour faire varier leur valeur ;
- on place des lampes des sorties pour visualiser leur valeur ;
- on teste ensuite toutes les lignes de la table de vérité en vérifiant que les sorties sont valides.



En utilisant exclusivement les opérandes NAND, transformer les fonctions suivantes :

- $F = \overline{A}$
- $F = A \cdot B$
- $F = A + B$
- $F = AB + CD$
- $F = A \oplus B$



Réaliser et vérifier la fonction $F = AB + CD$ en utilisant exclusivement des opérateurs NAND.



Réaliser et vérifier la fonction $F = (A + B) \cdot (C + D)$ en utilisant exclusivement des opérateurs NOR.
En utilisant exclusivement les opérateurs "OU EXCLUSIF" ou XOR, réaliser et vérifier la fonction $F = A \oplus B \oplus C$.



L'information est vulnérable (SEU, transfert, etc). Pour vérifier qu'un mot de 4-bit n'a pas été altéré, on ajoute un bit supplémentaire P qui est à 1 pour que le nombre de 1 soit pair (codage en parité paire). Ainsi, si un des bits est altéré (passage de 0 à 1 ou 1 à 0), le bit de contrôle de parité ne correspondra pas (nombre de 1 impair) et l'erreur sera détectée.



Tout d'abord, nous allons réaliser un circuit qui permet de calculer le bit P. Il aura pour entrée le mot de 4-bits B_3 à B_0 et renvoie P en sortie de telle manière que le mot composé de P et de B_3 à B_0 ait un nombre pair de 1. Pour cela, (1) réaliser la table de vérité, (2) obtenir les équations simplifiées par Karnaugh, (3) simplifier en utilisant des XOR et (4) réaliser et tester le circuit.



Nous allons réaliser un circuit qui permet de vérifier qu'il n'y a pas eu d'erreur de transmission. Il prend 5 entrées B_3 à B_0 et P et renvoie en sortie ERR à 1 si une erreur est détectée (nombre de 1 impair). Peut-on s'inspirer du circuit précédent pour réaliser ce circuit ?



En utilisant exclusivement des opérandes NAND à deux entrées, transformer la fonction $F = \overline{A} + \overline{B} + \overline{C}$

En utilisant exclusivement des opérandes NAND à trois entrées, puis à deux entrées, transformer la fonction $F = A C \overline{D} + A C + B$.