

TP 1 : Initiation à l'édition de circuits avec TkGate

But de ce TP est la familiarisation avec l'éditeur et simulateur de circuits *TkGate*. Cet éditeur peut être téléchargé sur <http://www.tkgate.org> et en tant que package dans certaines distributions Linux). Il dépend cependant de Linux. Pensez à la virtualisation (virtualbox.org) pour pouvoir l'installer sur d'autres systèmes.

Si tkgate n'est pas installé sur une machine Linux : en ce cas, vous pouvez exécuter le script indiqué sur la page web ci-dessus pour l'installer temporairement dans le dossier /tmp.

Nous utiliserons dans ce cours la version 2.0b10 qui n'est pas compatible avec la version 1.87.

Exercice 1 (*Didacticiel TkGate*)

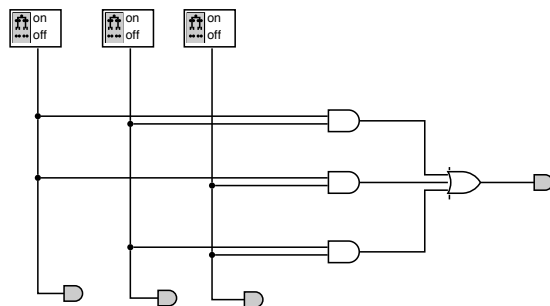
Pour démarrer le simulateur, tapez `tkgate`. Pour suivre le didacticiel anglophone, il peut être préférable de configurer les préférences linguistiques pour préférer l'anglais américain. En ligne de commande :

```
> export LANG=C
> tkgate
```

Lisez l'introduction et réalisez les exercices des différents didacticiels, chapitres 1-7.

Exercice 2 (*Réalisation de la fonction Majorité*)

Ouvrir un nouveau circuit (*File/New*) intitulé `majorite.v`. Créer le circuit majorité ci-dessous.



a	b	c	Majorité(a, b, c)
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

Enregistrer votre réalisation sur votre compte.

Exercice 3 (*Utilisation de diodes pour le test*)

Nous poursuivons l'exercice 2. Lancer *TkGate* et récupérer la réalisation de la fonction Majorité. Passer en mode simulation et démarrer la simulation, puis cliquer sur les différents interrupteurs. Vérifier le bon fonctionnement du circuit.

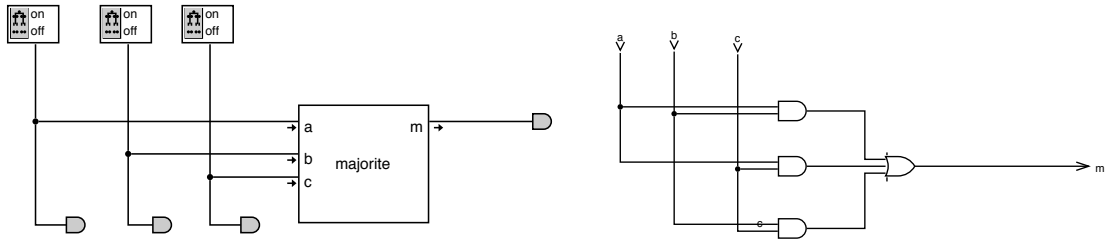
Exercice 4 (*Séparation en module*)

Lancer *TkGate* et récupérer la réalisation de la fonction Majorité. L'enregistrer sous un nouveau

nom `majorite2.v`. Créer un nouveau module et l'appeler "majorite". Sélectionner les portes, couper, ouvrir le module (icone à gauche) et coller. Ajouter des entrées et sorties (menu Component/Module) et les lier au circuit.

Passer sur l'onglet "Interface" et générer le symbole automatiquement.

Retourner au module "main" et tirer le nouveau module majorité vers le trou laissé après coupure. Lier le module au circuit main.



Passer en mode simulation et vérifier que le module fonctionne comme attendu.

Exercice 5 (Réalisation d'un Multiplexeur)

Le *Multiplexeur 2-1* est un circuit combinatoire avec trois entrées x_0 , x_1 et c et une sortie s telle que $s = x_0$ si $c = 0$ et $s = x_1$ si $c = 1$. Réaliser un multiplexeur 2-1 avec *TkGate* en tant que module.

Réaliser un circuit d'interrupteurs et de diodes pour simuler le multiplexeur. Avec le simulateur établir une table de vérité et vérifier la fonction du multiplexeur. En utilisant le simulateur (et une diode en sortie) tester la réalisation obtenue en complétant le tableau suivant :

c	x_0	x_1	Sortie
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Enregistrer cette réalisation sur votre compte.

Exercice 6 (Réalisation modulaire d'un multiplexeur 4-1)

Le *multiplexeur 4-1* est un circuit combinatoire d'usage fréquent. Il consiste en 6 entrées c_0 , c_1 , x_0 , x_1 , x_2 et x_3 et une sortie s décrite par la formule

$$s = c_1 \cdot c_0 \cdot x_3 + c_1 \cdot \overline{c_0} \cdot x_2 + \overline{c_1} \cdot c_0 \cdot x_1 + \overline{c_1} \cdot \overline{c_0} \cdot x_0.$$

Dans cet exercice, nous construisons de manière descendante un module multiplexeur 4-1 à l'aide du module **selectionneur 2-1** de l'exercice précédente.

Tester le circuit obtenu sur quelques-uns des 2^6 entrées possibles.

Exercice 7 (Réalisation modulaire d'un additionneur 8 bits)

Question 1. Réaliser un module additionneur complet et testez le.

Question 2. Réaliser un module additionneur 8 bits à l'aide du module additionneur complet.

Exercice 8

Faire une copie du circuit `majorite2.v` (avec le module `majorite`), effacer le module du circuit main

et ajouter un nouveau module `majoriteVerilog` et choisir le type HDL pour ce module. Ce type de module n'est pas dessiné mais programmé en Verilog (éditeur de texte).

Commencez avec la coquille suivante :

```
module majoriteVerilog(m,a,b,c);
  input a,b,c;
  output m;
  wire w1; // ajoutez vos cables supplémentaires

  and(w1,a,b); // w1 = a * b
  // ajoutez vos déclarations de portes supplémentaires
  // ..

endmodule
```

Passer en mode interface et générez le symbole. Intégrez le module dans le circuit main à la place de l'ancien module `majorite` et testez.