

# TP N°1

## Simulation de Fautes

### Objectifs du TP :

- Liste de fautes équivalentes
- Simulation de fautes
- Minimisation de la séquence de test

## Exercice 1 :

### 1. Partie Théorique

Considérons comme premier exemple de circuit à tester le schéma au niveau porte présenté sur la Figure 1.

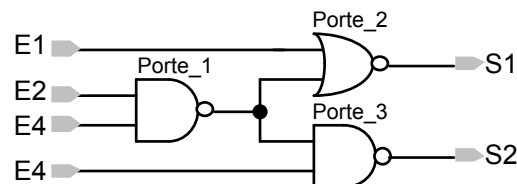


Figure 1 : Exercice 1

A partir du schéma ci-dessus voici la description *Verilog* correspondante :

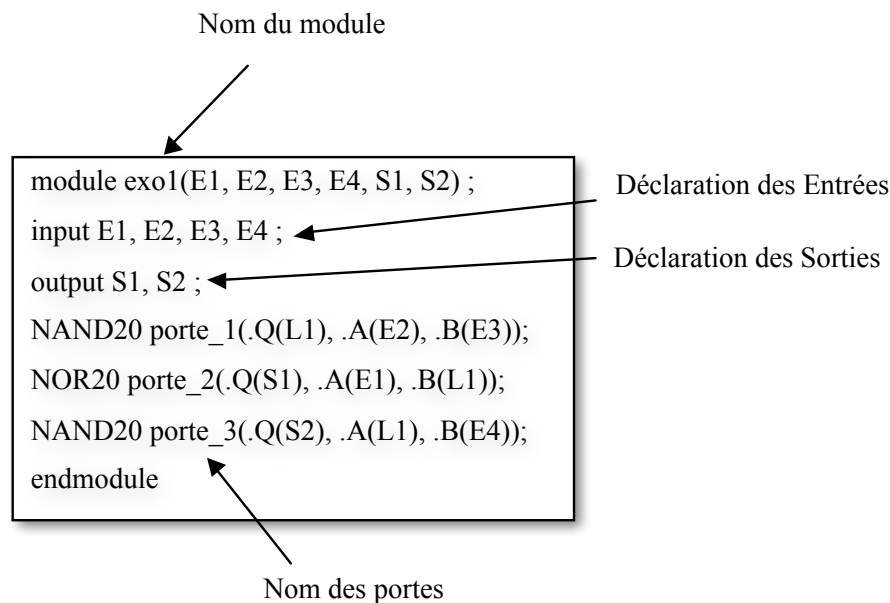


Figure 2 : Description Verilog

Le terme *module* détermine le début de la description. Il est suivi par le nom du module, dans notre exemple *exo1*, ainsi que l'ensemble des entrées et sorties. Les deux lignes suivantes indiquent séparément les entrées (*input*) et les sorties (*output*). A partir de ce point, la description structurelle du circuit commence. Les termes *NAND20*, *NOR20* ... sont liés à la bibliothèque de portes utilisée. Dans les parenthèses, on indique en premier la sortie suivie des entrées. La description se termine par le terme *endmodule*.

## QUESTIONS :

D'après l'exemple précédent voir Figure 1:

1. Quel est le nombre total de faute de collage sans minimisation ?
2. Minimisez ce nombre en utilisant les équivalences et implications.
3. Trouvez les vecteurs de test à appliquer aux entrées du circuit afin de tester les fautes de l'ensemble minimal.

## 2. Partie Pratique

Après avoir copié le dossier TP1 sur votre compte, vous devez analyser son contenu :

- Fichier « C35.V » : fichier VERILOG contenant l'ensemble des éléments (portes, FF ...) de la bibliothèque C35.
- Fichiers « exo1.v », « exo2.v » et « exo3.v » : fichiers VERILOG contenant les descriptions des exercices.
- Fichiers « example\_exo1.still », « example\_exo2.still » et « example\_exo3.still » : fichiers au format STILL des séquences de test. Analysez ce fichier et complétez le avec la séquence que vous avez générée dans la partie théorique.
- Fichier « tp1\_command\_example » : Exemple de script contenant les commandes essentielles pour lancer une simulation de faute.

```
read_verilog C35.V -library    // commande permettant de lire la techno C35
read_verilog exo1.v           // commande permettant de lire description du circuit exo1.v
run_build_model               // commande de compilation
run_drc                       // commande de lancement de l'étape de DRC (sera plus complète au TP2)
set_faults -model stuck       // sélection du modèle de faute
add_faults -all                // génération de la liste de faute totale
set_patterns -external example_exo1.stil // charger la liste de vecteurs externes
run_simulation                 // Simulation logique permettant de calculer les sorties
run_simulation -override_differences // Simulation logique stockant les valeurs sorties
report_patterns -external -all // Visualisation des vecteurs
run_simulation                 // Simulation logique permettant de calculer les sorties
run_fault_sim                  // Simulation de fautes
```

Configuration de TetraMax :

```
sh
source /soft/2018/Config/TMAX.sh
tmax -shell
```

En utilisant TetraMax:

1. Vérifiez si la séquence de test générée précédemment teste l'ensemble des fautes de collage.
2. Vérifiez les équivalences de fautes

## Exercice 2 :

Considérons comme deuxième exemple de circuit à tester le schéma au niveau porte présenté sur la Figure 3.

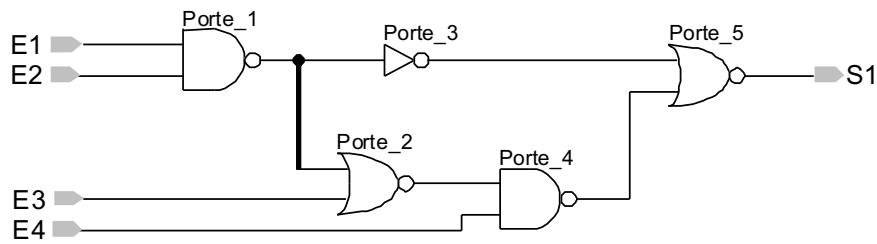


Figure 3 : Exercice 2

### QUESTIONS :

- D'après l'exemple précédent voir Figure 3:
  1. Quel est le nombre total de fautes de collage sans minimisation ?
  2. Minimisez ce nombre en utilisant les équivalences et implications.
- En utilisant TetraMax:
  1. Appliquez la séquence exhaustive en utilisant la liste de fautes trouvée précédemment.
  2. Analysez les résultats.
  3. Que peut-on faire pour que l'ensemble des fautes soient testées ?

## Exercice 3 :

Refaire les mêmes questions que celles de l'exercice 2 en utilisant le circuit de la Figure 4.

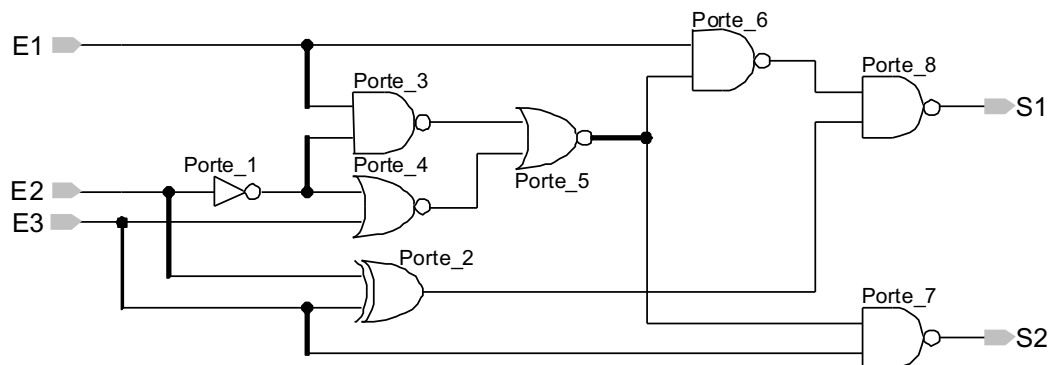


Figure 4 : Exercice 3