

# Une Approche de Calcul Stochastique pour l'Evaluation de Fiabilité Précis et Efficace

## Une Implementation Python

Jake Humphrey

Department of Electronic and Electrical Engineering  
Imperial College London  
[jbh111@ic.ac.uk](mailto:jbh111@ic.ac.uk)

November 25, 2014

# Fiabilité des Circuits

Des Portes dans un circuit logique sont sensibles aux erreurs:

- ▶ **Erreur Coincé-A-Un:** La sortie va haute.
- ▶ **Erreur Coincé-A-Zero:** La sortie va basse.
- ▶ **Erreur Von Neumann:** La sortie est inversée.

# Effets de masquage

Cependant, il est possible que l'erreur n'affecte pas la sortie en raison de l'un des *effets de masque* suivants:

- ▶ **Masquage Electrique:** Signal d'erreur trop faible pour être détecté.
- ▶ **Masquage Temporel:** Erreur manque la fenêtre de détection d'un latch.
- ▶ **Masquage Logique:** L'erreur ne modifie pas la sortie d'une porte logique.

# Analyse de Fiabilité Principes

Masquage logique est le plus commun, et on essaye d'analyser des circuits sur leur capacité à masquer logiquement des erreurs.

*Probabilité* d'un signal = Chance qu'il est logiquement vrai.

*Fiabilité* d'un signal = Chance qu'il prend la valeur correcte.

Puis:

- ▶ Construire une représentation défectueux du circuit.
- ▶ Dériver les *probabilités* des sorties à partir des entrées.
- ▶ Trouvez les *fiabilités* des sorties.

# Analyse de Fiabilité

## Modèles de Portes Probabilistes

Cependant, les algorithmes existants sont inefficaces!

Par exemple, les *Modèles de Portes Probabilistes* ( En Anglais *Probabilistic Gate Models*, PGMs) tenter de dériver les probabilités de sortie de façon déterministe et analytique.

# Analyse de Fiabilité

## Modèles de Portes Probabilistes

Le problème se produit lorsque les entrées à une porte sont statistiquement dépendante.

Les équations PGM ne tiennent pas compte des signaux statistiquement dépendantes, et la solution exige que le circuit soit divisé en deux sous-circuits, qui double le coût de l'algorithme.

# Analyse de Fiabilité Calcul Stochastique

L'utilisation de *Calcul Stochastique* peut éviter ces problèmes.

Générez des flux binaires d'entrée et les propagez à travers le circuit.

Les probabilités de sortie peuvent ensuite être calculés avec précision à partir des flux de bits de sortie.

Algorithmes existants de calcul stochastique utilisent les probabilités d'entrée pour générer des *Suites Bernoulli* de la forme:

$$[X_0, X_1 \dots X_{n-1}]; X_i \sim B(p)$$

pour chaque entrée, où  $p$  est la probabilité d'entrée.

Nécessite que  $n$  nombres aléatoires doivent être générés pour chaque entrée!



# Analyse de Fiabilité Calcul Stochastique avec les Suites

## non-Bernoulli

*Suites non-Bernoulli* réduisent le coût en raison de génération de nombres aléatoires.

Ils sont générés de manière déterministe avec l'espérance des 1s, puis permutées au hasard.

Seulement une génération d'un nombre aléatoire est requis par flux binaire d'entrée!

# Algorithme d'Analyse de la Fiabilité

En utilisant le calcul stochastique avec les suites non-Bernoulli, on arrive à un algorithme pour analyser de fiabilité, que je vais décrire dans les diapositives suivantes.

# Algorithme d'Analyse de la Fiabilité Pseudocode

**Data:** circuit logique à tester

**Result:** fiabilités pour chaque sortie

**for** *chaque porte dans le circuit* **do**

    | représenter porte dans le circuit défectueux;

**end**

**for** *chaque entrée du circuit défectueux* **do**

    | générer une suite non-Bernoulli;

**end**

**for** *chaque sortie dans le circuit* **do**

    | **for** *chaque vecteur d'entrée* **do**

        | **if** *les sorties sont les mêmes pour chaque circuit* **then**

            | ajouter  $1/n$  à la fiabilité de sortie ;

        | **end**

    | **end**

**end**

# Algorithme d'Analyse de la Fiabilité

## Analyse de la Complexité

La section la plus coûteuse est la double boucle finale!

Il faut propager un signal à travers chaque circuit une fois pour chaque sortie, et  $n$  fois.

Peut-être, au pire,  $O(ogn)$ , où:

$o$  = nombre de sorties

$n$  = longueur de suites non-Bernoulli

$g$  = nombre de portes

# Algorithme d'Analyse de la Fiabilité d'Autre Travail

Il y a encore place à l'amélioration: les entrées se propagent à travers le circuit une fois par sortie.

Il doit recalculer de nombreuses valeurs intermédiaires redondantes!

Calculer toutes les valeurs de sortie une seule fois permettrait de réduire la double boucle à un seul sur la longueur des séquences d'entrée.

Ceci réduit la complexité de l'algorithme à  $O(gn)$

# Algorithme d'Analyse de la Fiabilité

Analyse d'Exécution

| Fichiers d'entrée             | c17.v      | c432.v        |
|-------------------------------|------------|---------------|
| Entrées                       | 5          | 36            |
| Sorties                       | 2          | 7             |
| Portes                        | 6          | 160           |
| Longueur de la Suite d'Entrée | runtime /s | runtime /s    |
| 1                             | 0.00023    | 1.34          |
| 10                            | 0.00067    | 12.2          |
| 100                           | 0.00485    | 122           |
| 1 000                         | 0.0473     | 1 250         |
| 10 000                        | 0.458      | Aucune donnée |
| 100 000                       | 4.66       | Aucune donnée |

# Algorithme d'Analyse de la Fiabilité

## Analyse d'Exécution

