



Modélisation et simulation de l'effet de phénomènes de nature aléatoire dans les circuits intégrés

Adam Heriban

Laboratoire de Calcul et Environnement de conception,
CEA/DRT/LIST/DACLE

Résumé : Les outils déterministes utilisés jusqu'à aujourd'hui pour les simulations de microprocesseurs sont désormais insuffisants. Il faut donc explorer de nouvelles méthodes de simulations intégrant, à présent, un aspect stochastique.



On retient tout d'abord une loi de vieillissement en particulier développée dans [1], basée sur un couplage HCI/BTI. Pour répondre aux contraintes imposées par la physique du vieillissement, on est obligé, pour les tirages aléatoires, d'utiliser une nouvelle loi statistique. On effectue ici des tirages sur l'incrément de retard d'une porte logique :

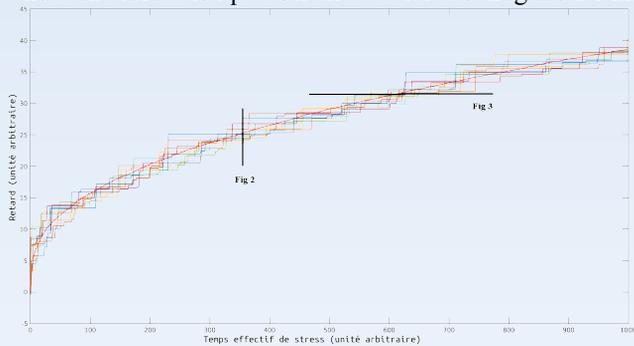


Fig 1 : En rouge : la loi déterministe. Autres couleurs : les essais aléatoires.

On a ainsi des tirages aléatoires qui suivent en moyenne la loi déterministe (première contrainte physique), qui sont strictement croissants durant le temps effectif de stress (seconde contrainte). On remarque également dans les figures 2 et 3 que les essais suivent à chaque instants une loi gaussienne (troisième contrainte). De plus, si l'on fixe un retard correspond à une défaillance, les instants de défaillance sont répartis selon une loi Gamma. Ce n'était pas une contrainte, mais le fait que cette loi soit commune pour la modélisation de phénomènes d'usure renforce la cohérence du processus de simulation.

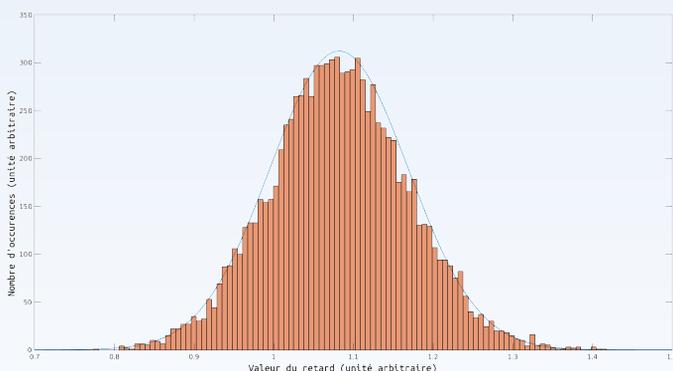


Fig 2 : En orange : histogramme des essais. En bleu : gaussienne associée.

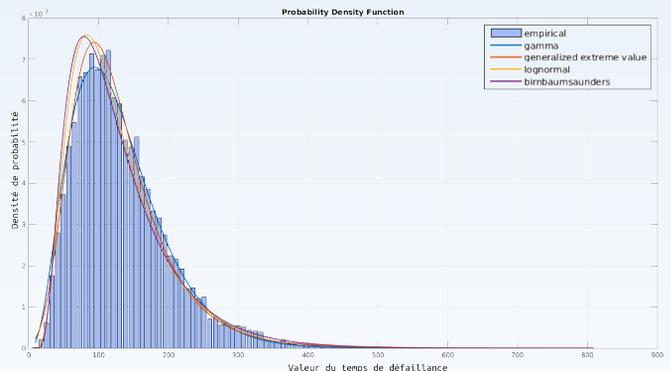


Fig 3 : En bleu : histogramme des essais. Cyan : Loi Gamma associée

Conclusion : On a bien créé un outil permettant d'inclure un aspect stochastique dans la chaîne de simulation. Ce dernier reste cependant à tester de façon approfondie.

Références :

[1] HUARD, V., PION, E., CACHO, F., et al. A predictive bottom-up hierarchical approach to digital system reliability. In : IEEE International Reliability Physics Symposium (IRPS'12). IEEE Computer Society, 2012. p. 4B. 1.1-4B. 1.10.