Recherche de suites à faible taux de corrélation.

Lerman Benjamin

12 juin 1999

1 Abstract

This document describes an application of the FPGA for the brute force research of low-autocorrelations sequences [1]. The aim is to find a sequences A which will minimize $E(A)$ where A and $E(A)$ are defined as :

$$A = a_0, \dots, a_{n-1} \text{ where } a_i \in \{-1, 1\}$$

$$s_k(A) = \sum_{0 \leq i \leq n-1-k} a_i a_{i+k}$$

$$E(A) = \sum_{1 \leq i \leq n-1} s_i(A)^2$$

2 Considérations mathématiques

Pour énumérer l'ensemble des valeurs de A , on va utiliser un compteur de Gray. Dans ces conditions, entre deux valeurs successives de A , il n'y a qu'un unique a_i qui change de valeur. Donc chacun des s_k voit sa valeur changer de $-4, -2, 0, 2$, ou 4 . On remarque donc que la parité de s_k ne dépend que de n . On va donc stocker au fur et à mesure la partie entière de $\frac{s_k}{2}$ à qui on va additionner $-2, -1, 0, 1, 2$ en fonction du bit de A qui change. De plus quand on élève s_k au carré, on remarque que l'on obtient un terme du type $4n^2$ ou $4(n^2 + n) + 1$. Or l'on s'intéresse à la suite qui minimise $E(A)$, et non pas à $E(A)$ lui même. On peut donc calculer $E^*(A)$ obtenu en remplaçant $s_i(A)^2$ par m'^2 si $s_i(A) = 2m'$ ou par $m'^2 + m'$ si $s_i(A) = 2m' + 1$. On vérifie alors aisément qu'une suite A minimisant $E(A)$ minimise également $E^*(A)$.

3 Réalisation pour $n = 16$ bits

On commence par utiliser un compteur 16 bits afin d'obtenir un compteur de Gray. De plus, une simple comparaison entre la valeur du compteur au temps t et au temps $t + 1$ nous permet d'obtenir le bit qui a changé. On utilise donc 16 registres pour garder en mémoire la valeur du compteur au temps t .

On stocke ensuite pour chaque s_k la valeur du compteur de Gray dans 2 RAMs double port pour pouvoir à chaque coup d'horloge récupérer la valeur au coup d'horloge précédent du bit qui vient de changer. On récupère également les valeurs des bits décalés de k par rapport à ce dernier. Ces trois bits nous permettent de déterminer si la valeur de $\frac{s_k}{2}$ doit être modifiée de $-2, -1, 0, 1, 2$. On change en conséquence la RAM qui conserve la valeur de s_k .

On a ensuite deux types de ROM codant les opérations a^2 et $a^2 + a$. Les sorties de ces ROMs sont additionnées en arbre logarithmique et on compare la valeur trouvée avec la valeur au coup d'horloge précédent. Si on a trouvé une valeur inférieure, on sauve l'état du compteur ainsi que le minimum relatif.

Quand le compteur a énuméré les 2^{16} possibilités pour A , on a alors avec certitude une suite réalisant le minimum de $E(A)$. On peut alors calculer cette dernière valeur sans difficulté.

4 Passer à des suites plus longues

Le passage à des suites plus longues peut se faire de deux manières différentes (mais éventuellement complémentaires).

La première méthode est la plus élémentaire. On peut simplement augmenter la taille du compteur, augmenter la largeur des différentes RAMs et ROMs et ainsi obtenir un calcul direct pour des suites plus longues. La taille de la réalisation étant à peu près proportionnelle à n , on peut obtenir des résultats significatifs pour 32 bits, mais cette contrainte peut devenir très limitative au delà. Il peut donc être plus intéressant de procéder différemment.

Les symétries du problème nous permettent de diviser par un peu moins de 4 l'espace de recherche. Pour ce faire, on va partager l'espace de recherche en deux parties. Une des deux parties sera traitée sur le FPGA en utilisant une technique similaire à celle qui a été mise en place pour 16 bits, l'autre partie sera traitée par un ordinateur qui compilera le champ de bits pour le FPGA. Il faut alors trouver le nombre de bits qui sera traité en une fois par le FPGA de telle façon que la recompilation prenne un temps négligeable devant le temps nécessaire à faire ce calcul. On peut alors espérer un gain d'un facteur 4 en vitesse par rapport à un calcul n'utilisant que le FPGA et qui ne prend pas en compte les symétries. De plus, la partie traitée par l'ordinateur nous permet de diminuer la place nécessaire sur le FPGA et donc d'espérer calculer A pour des valeurs de n plus importantes. On peut alors espérer calculer A pour n dépassant la limite de 48 obtenue par S. Mertens [1].

Références

- [1] S. Mertens. Exhaustive search for low-autocorrelation binary sequences. *Letter to the Editor*, 1996.