PROG
TD#5: Transposition de matrice binaire

2019-11-06

Le but de cet exercice est d'implémenter la transposition d'une matrice 8×8 à coefficients dans $\mathbb{F}_2$, quand celle-ci est représentée sous la forme d'un entier non signé de 64 bits. Autrement dit, soit $\boldsymbol{M} = \left[\boldsymbol{M}_{i,j}\right]_{0 \leq i,j < 8}$ représentée *par ligne*[1] comme l'entier en base 2 $\mathtt{m} = \boldsymbol{M}_{7,7}\boldsymbol{M}_{7,6} \cdots \boldsymbol{M}_{1,1}\boldsymbol{M}_{1,0} \cdots \boldsymbol{M}_{0,2}\boldsymbol{M}_{0,1}\boldsymbol{M}_{0,0}$, on souhaite calculer l'entier représentant $\boldsymbol{M}^t$ (ou de façon équivalente, l'entier représentant $\boldsymbol{M}$ *par colonnes*[2]), c-à-d $\boldsymbol{M}_{7,7}\boldsymbol{M}_{6,7} \cdots \boldsymbol{M}_{1,1}\boldsymbol{M}_{0,1} \cdots \boldsymbol{M}_{2,0}\boldsymbol{M}_{1,0}\boldsymbol{M}_{0,0}$. On va pour cela utiliser récursivement le fait que pour une matrice $\boldsymbol{M}$ de dimension $n \times l$ où n et l sont tous deux pairs, on peut écrire celle-ci comme la matrice bloc $\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}$ où les blocs ont dimension $n/2 \times l/2$, et on a alors $\boldsymbol{M}^t = \begin{bmatrix} \boldsymbol{A}^t & \boldsymbol{C}^t \\ \boldsymbol{B}^t & \boldsymbol{D}^t \end{bmatrix}$.

Q.1 : Écrivez la transposition de $\boldsymbol{M} \in \mathbb{F}_2^{8\times 8}$ comme la composition de trois transpositions (éventuellement par blocs).

Q.2 : Écrivez une fonction C `uint64_t tr8x8_r(uint64_t x)` implémentant l'algorithme ci-dessus. Celle-ci ne doit pas utiliser de boucle, ni dépasser une douzaine de ligne.

Q.3 : L'extension d'instructions *BMI2* des processeurs Intel contient une instruction `pext`, typiquement accessible via l'intrinsèque `uint64_t _pext_u64 (uint64_t a, uint64_t mask)`, qui extrait les bits de `a` indiqués par `mask` et les écrit de façon contiguë dans les bits de poids faible du résultat.

Écrivez une fonction C `uint64_t tr8x8_bm(uint64_t x)` qui implémente la transposition de $\boldsymbol{M}$ en utilisant `_pext_u64`.

Q.4 : Comparez vos deux fonctions.

Remarque. Bien que les implémentations étudiées dans cet exercice soient plus efficaces qu'une solution naïve, la transposition (comme généralement les permutations) est une opération coûteuse. En plus grande dimension, une approche récursive reste avantageuse, mais une complexité supplémentaire peut être ajoutée par le besoin de limiter les défauts de cache.

1. *Row-major.*
2. *Column-major.*