Mémoire à code correcteur d’erreur et codes de Hamming

**Positionnement thématique :**

Informatique théorique, Informatique pratique, Technologies informatiques, Algèbre

**Mots clés :**

|  |  |
| --- | --- |
| **Mots clés** (français)**:**  Code correcteur d’erreurs  Codes linéaires  Corps finis  Mémoire à code correcteur d’erreur  Corruption de bits | **Mots clés** (anglais) :  Error-correcting codes  Linear codes  Finite fields  ECC (Error Correcting Code) memory  Bit flipping |

**Bibliographie commentée :**

Il y a certains domaines où les données informatiques ne doivent en aucun cas être corrompues, comme la finance, le calcul scientifique ou la gestion de serveurs. Dans ces domaines de précision, une erreur sur le moindre bit de donnée peut altérer les résultats [1]. Ces erreurs sont dues principalement à deux phénomènes : des réactions engendrées par des rayonnements cosmiques (interaction avec des neutrons) [2] ou le phénomène de martellement de mémoire, effet secondaire de l’accès mémoire qui peut modifier les cases mémoires adjacentes à celle consultée [3]. Pour pallier à cela, il existe un type spécial de mémoire RAM, appelée en anglais ECC Memory (pour Error Correcting Code Memory ou encore mémoire à code correcteur d’erreur). Cette mémoire, en plus du module DRAM classique, contient des bits de stockage supplémentaires permettant l’utilisation de contrôle de parité ou de code correcteur d’erreur [1, 4].

Le principe de la correction d’erreurs est d’introduire de la redondance afin de pouvoir retrouver l’information même si celle-là a été endommagée. C’est ce principe qui est illustré avec le contrôle de parité. Celui-ci consiste à ajouter à chaque octet un ‘1’ s’il y a un nombre impair de ‘1’ dans l’octet et un ‘0’ s’il y en a un nombre pair [5]. Par exemple, l’octet 01101001 deviendra 011010010. Le problème de ce système est qu’il ne détecte qu’un nombre impair d’erreurs, et qu’il ne peut que les détecter [5].

On utilise donc un code correcteur pour pouvoir détecter deux erreurs et en corriger une : un code de Hamming [4]. Plus précisément, il s’agit d’un code de Hamming avec contrôle de parité appelé SECDED pour Single Error Correction and Double Error Detection [6].

Un code de Hamming consiste à associer à un mot un certain nombre de sommes de contrôle. Le mot de code correspondant à un mot est alors la concaténation de ce mot avec les sommes de contrôle. Dans le cas du code SECDED, il s’agit d’un code de paramètre (72,64), c’est-à-dire qu’à un mot de 8 octets, on associe un mot de 9 octets, ceci incluant le bit de contrôle de parité.

Un code de Hamming est parfait, c’est-à-dire qu’à chaque mot de 71 bits, on peut associer un unique mot de code de 71 bits qui correspond au mot de code « le plus proche » (pour la distance de Hamming : il s’agit du nombre de bits en lesquels deux mot binaires de même longueur diffèrent) du mot considéré [6,7]. Ainsi si une erreur est commise, elle pourra être corrigée. Mais, si deux erreurs ont lieu, alors le mot ne sera pas corrigé en le mot original, c’est pourquoi on ajoute un contrôle de parité [6].

Lorsque l’on écrit dans la mémoire, le mot passe dans un encodeur qui génère les sommes de contrôle à l’aide de circuits XOR (OU-exclusif), et le mot ainsi que les sommes de contrôle sont stockés dans la RAM. Lors de la lecture de données, le mot passe dans un circuit décodeur utilisant des tables mathématiques et qui renvoient le nombre d’erreurs (0, 1 ou 2) présentes dans le mot ainsi que le mot corrigé, si cela est possible [6].

Un des problèmes posés par ce type de mémoire est qu’il faut plus de place pour stocker les mêmes données (9 octets au lieu de 8). Elle est également moins rapide car les données doivent subir des traitements supplémentaires à l’écriture et à la lecture. Son utilisation n’est donc pas préconisée en toutes circonstances.

**Problématique retenue :**

La mémoire ECC est une technologie de pointe et son implémentation la rend plus chère et légèrement moins performante que de RAM standard [1, 6]. Quels sont précisément les différences de performance entre ces deux types de mémoire et dans quelles situations est-il pertinent de choisir de la mémoire ECC ?

**Objectifs du TIPE :**

* Mise en évidence d’erreurs mémoire et quantification
* Etude du code de Hamming (72 ,64) et de ses algorithmes de codage et de décodage
* simulation python d’une mémoire ECC

**Références bibliographiques :**

[1] Jeff Layton, admin-magazine.com, *Error detection and correction,* <http://www.admin-magazine.com/Articles/Monitoring-Memory-Errors>, mis à jour le 29/07/2016, consulté le 20/12/2016

[2] Eugene Normand, *Single Event Upset at Ground Level,* Boeing Defense and Space Group, Seattle, WA 98124-2499

[3] Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai et Onur Mutlu, *Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors,* Carnegie Mellon University, 2014

[4] Werner Fisher, admin-magazine.com, *RAM Revealed,* <http://www.admin-magazine.com/Articles/RAM-Revealed>, mis à jour le 09/03/2016, consulté le 20/12/2046

[5] crucial.com, *What is ECC memory ?,* <http://www.crucial.com/usa/en/memory-server-ecc>*,* mis à jour le 28 août 2016, consulté le 22/12/2016

[6] Simon Tam, *Single Error Correction and Double Error Detection,* Xilinx, 2006

[7] Michel Demazure, *Cours d’algèbre,* Cassini, 2008

Dans certains domaines, la perte ou la corruption de données sont inacceptables. Néanmoins, ce phénomène, ayant de multiples causes, est inévitable. Pour le contrer, il existe un type spécial de mémoire permettant la détection et la correction de la plupart des erreurs. Cette technologie étant couteuse et entrainant une perte de performances, le choix de son utilisation doit être cependant réfléchit.  
  
J'ai choisi ce thème car il permet de mieux comprendre le fonctionnement des ordinateurs, que nous côtoyons quotidiennement. Il permet également  d'explorer la théorie des codes, branche des mathématiques à la croisée entre algèbre linéaire et théorie de Galois.

Lors du transfert ou du stockage de données, des contraintes physiques peuvent altérer ou effacer ces données. Pour parer à ce phénomène, on code l'information de sorte que l'on puisse détecter voire corriger ces erreurs. Mais ces codes introduisent de la redondance et nécessitent des calculs, leur utilisation entraine donc une perte de performance. Il faut donc selon la situation, choisir le code optimal, s'adaptant au mieux à ces contraintes.  
J'ai choisi ce sujet car il permet d'explorer la théorie des codes, branches des mathématiques développée dans le but de répondre à ces problématiques concrètes en usant d'outils mathématiques abstraits.