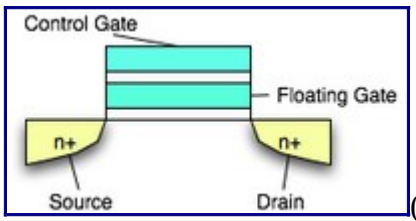
Les **SSD** (pour ***Solid State Drive***, contrôleur à semi-conducteurs) sont des dispositifs qui utilisent autre chose que la technologie magnétique des disques durs pour stocker des données => mémoire flash.

La **mémoire flash** NAND est une mémoire de stockage qui utilise des transistors comme support. Son fonctionnement est basé sur l'effet tunnel (plus précisément l'effet *Fowler-Nordheim*).

Les transistors utilisés dans la mémoire flash contiennent deux grilles, une de contrôle et une deuxième, appelée grille flottante, qui est en suspension dans un oxyde, le tout est placé sur un substrat qui contient deux électrodes.



(image Wikpedia)

Pour écrire une donnée, on doit faire passer un courant électrique (7 V) entre les deux électrodes (drain et source) et une tension plus élevée (aux environs de 12 V) dans la grille de contrôle. L'effet *Fowler-Nordheim* implique qu'une partie des électrons qui passent entre les électrodes va se déplacer vers la grille flottante, à travers l'oxyde. Une fois la grille saturée avec des électrons, elle devient isolante et est considérée comme un 0 binaire.

L'effacement d'une cellule s'effectue de la même façon, mais en faisant passer une tension négative dans la grille de contrôle. Les électrons se déplacent alors de la grille flottante vers le substrat. Une fois la grille flottante « vidée » de ses électrons, elle est considérée comme dans un 1 binaire.

Pour la lecture, c'est assez simple : il faut mesurer la résistance de la grille flottante, en faisant passer une tension faible (5 V) dans la grille de contrôle et dans une des électrodes. Si les électrons passent entre la grille de contrôle et l'électrode, la grille flottante n'est pas isolante, on a un 1 binaire. Si le courant ne passe pas, on a un 0 binaire. La lecture est donc plus rapide que l'écriture ou l'effacement, car on ne doit pas remplir ou vider la grille flottante avec des électrons.

Mémoire SLC et mémoire MLC

Notons qu'il existe deux types de NAND, la SLC (*Single Layer Cell*) qui stocke un seul bit dans la grille flottante, et la MLC (*Multi Layer Cell*), qui stocke plusieurs bits dans la même cellule. Techniquement, la MLC divise la grille flottante en deux parties, avec une différence de tension entre les deux parties. On peut donc doubler la capacité de stockage en gardant la même taille physique, ce qui est évidemment avantageux. La mémoire MLC a cependant des défauts : comme on doit travailler avec plusieurs tensions différentes, l'écriture et l'effacement des données sont plus lents qu'avec de la SLC. La lecture reste rapide, mais pas autant qu'avec de la mémoire SLC (environ 80 % du débit d'une SLC équivalente).

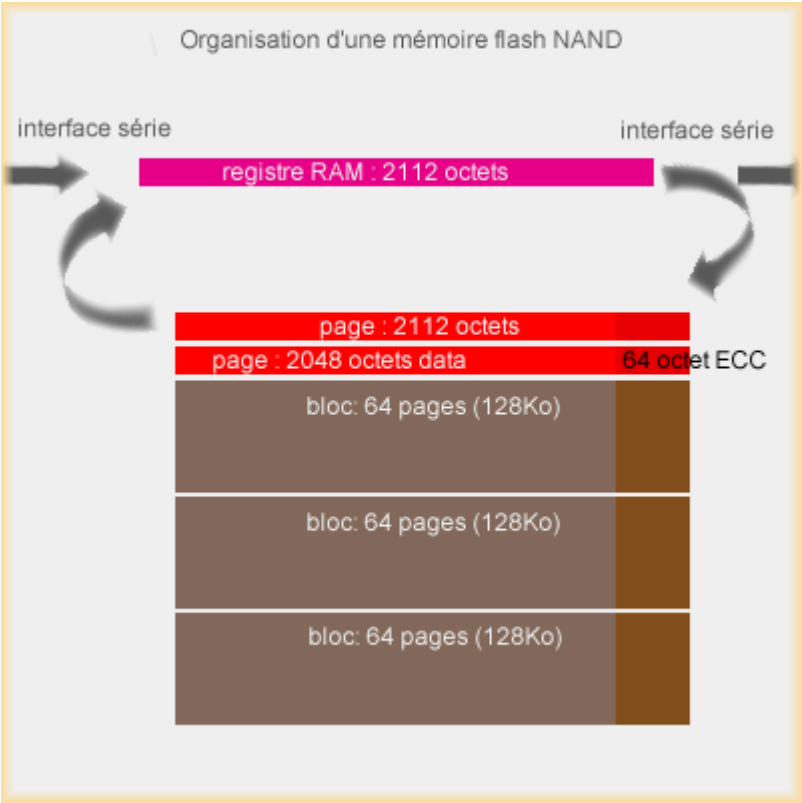
Actuellement, on utilise essentiellement de la mémoire MLC dans les SSD, pour des raisons de coût. Et pour des raisons de performances, c'est de la mémoire qui stocke deux bits par cellule qui est utilisée. Des mémoires à trois ou quatre bits existent, mais elles ont le gros défaut de diminuer la durée de vie et d'offrir des performances en écriture très faible. Les mémoires 3 bits et 4 bits actuelles se destinent essentiellement aux cartes microSD et aux clés USB d'entrée de gamme.

La flash NAND et l'organisation en blocs

La mémoire flash NAND travaille avec un bus série, en accès séquentiel. Il est impossible d'accéder directement à un bit en particulier, contrairement à la mémoire NOR. Pour accéder à une information précise, on doit charger entièrement une partie des données dans une petite mémoire RAM, et ensuite lire ce que l'on veut dans cette mémoire.

Dans un système classique (une mémoire de 16 Go par exemple) on va travailler avec des blocs de 256 ou

512 ko. Le bloc est divisé en 64 ou 128 pages de 4 ko. En réalité, une « page » fait plus que 4 ko : on a 4 096 octets accessibles, et 128 octets qui servent de contrôle (gestion ECC, etc.). Les anciens systèmes (ou ceux de faible capacité) travaillent plutôt avec des pages de 2 ko avec des 64 octets de gestion et des blocs de 128 ko.



Pour ce qui est de l'accès, on travaille en *shadowing* : le système accède en fait à une mémoire RAM qui contient les données demandées. On n'accède jamais directement aux données stockées dans la mémoire NAND. Le contrôleur s'occupe de gérer la copie de la page dans la mémoire en question et de la gestion des blocs.

Lecture sur une page, écriture sur un bloc

L'interface séquentielle oblige aussi à travailler avec la page comme unité minimale pour la lecture. Si on veut accéder à un bit précis, il faut charger entièrement une page. Cette particularité explique que la mémoire flash manque parfois d'efficacité avec les très petits fichiers : la lecture d'un fichier de la taille d'une page (généralement 4 ko) et d'un fichier plus petit prend le même temps.

Pour l'écriture, on travaille au niveau du bloc : la moindre écriture oblige à effacer entièrement le bloc de données avant d'écrire une nouvelle valeur. On a donc le même problème qu'en lecture, écrire 1 bit ou écrire 256 ko (taille typique d'un bloc) nécessite le même temps au final : on doit reprogrammer entièrement le bloc. En pratique, l'écriture de petits fichiers (sous les 256 ko) est donc assez lente avec de la mémoire flash.

Parlons un peu performances : du fait de son mode de fonctionnement (séquentiel), l'accès aux données n'est pas instantané. Il faut compter environ 25 µs pour un accès à une page (temps de copie dans la RAM interne). L'accès aux autres pages du bloc est plus rapide (environ 0,03 µs), alors que l'effacement d'un bloc prend environ 2 ms.

En comparaison, sur des mémoires de type NOR, la lecture aléatoire d'une donnée est de 12 µs (quel que soit l'emplacement de celle-ci) et l'effacement d'un bloc est très lent : 750 ms.

Dans les contrôleurs actuels, une mémoire cache permet de regrouper en partie les écritures pour éviter d'effacer des blocs en masse, même si le fonctionnement de cette mémoire cache varie en fonction des contrôleurs utilisés et que tous les contrôleurs n'utilisent pas cette technique.

La durée de vie de la mémoire flash

Il y a deux raisons à la durée de vie limitée de la mémoire flash. La première vient de l'oxyde utilisé pour séparer les grilles. Comme nous l'avons vu, les électrons doivent traverser cet oxyde pour passer dans la grille flottante ou en sortir. De temps, il peut arriver que des électrons restent captifs de cet oxyde, et soient relâchés plus tard, ce qui peut perturber les écritures et la lecture.

La deuxième vient de la structure de la grille flottante elle-même : avec le temps, les tensions élevées peuvent l'endommager, ce qui à terme va la rendre inutilisable. On considère généralement qu'une cellule de mémoire SLC peut subir environ 100 000 écritures avant destruction, et que la mémoire MLC est moins endurante : entre 3 000 et 10 000 cycles d'écriture, en fonction de la finesse de gravure. Une mémoire intermédiaire, la eMLC, atteint 30 000 cycles.

La solution

La majorité des dispositifs utilisant de la mémoire flash disposent d'un contrôleur interne, qui va gérer les accès à la mémoire. Il est utilisé pour les transferts de données, mais aussi pour vérifier que le dispositif fonctionne bien.

Ce contrôleur vérifie automatiquement les écritures en relisant la dernière donnée écrite. S'il lui est impossible de la relire, le bloc complet est déterminé comme défectueux et l'écriture est relancée sur un autre bloc. Donc sauf si la mémoire flash est entièrement utilisée ou défectueuse, les données seront écrites sur le support.

La deuxième vérification s'effectue aussi durant l'écriture, avec la correction d'erreurs ECC. Toutes les puces de mémoire flash disposent de bits de contrôle, qui servent à la vérification des données écrites. En général, le contrôle ECC permet de corriger les erreurs de 1 bit et de détecter les erreurs sur 2 bits. Dans ce deuxième cas, le bloc est indiqué comme défectueux et l'écriture relancée sur un autre bloc.

On dispose de 3 octets d'ECC par bloc de 512 octets (donc 24 bits). À chaque écriture, le code ECC est calculé et écrit dans la zone inutilisée des blocs de données. Pendant la relecture, le code ECC des données est calculé et comparé au code ECC de départ (celui calculé à l'écriture). Si les deux codes sont identiques, le travail continue. Le premier cas intervient s'il y a une erreur sur 1 bit, à ce moment-là c'est corrigé et l'écriture continue. Enfin, il se peut qu'il y ait une erreur de 2 bits, ou une erreur dans le code ECC lui-même. Ces deux cas spécifiques ne sont pas corrigibles, et l'écriture est relancée sur un autre bloc et le bloc est indiqué comme défectueux.

Bien évidemment, les constructeurs se sont penchés sur ce problème de la durée de vie, et ils ont implémenté plusieurs techniques pour limiter l'usure de la mémoire NAND, c'est ce qu'on appelle le *Wear Leveling*, ou gestion de l'usure.

Dans une clé USB ou un périphérique de stockage comme une carte mémoire le fonctionnement est simple : le contrôleur va intercepter les écritures et les distribuer aléatoirement sur des blocs situés dans l'espace libre. Comme les écritures ne seront plus concentrées sur le même bloc physique, on ne risque pas de détruire un bloc en particulier si un programme écrit en permanence sur le même fichier. Le problème de cette technique est simple : si l'espace libre est trop faible, les écritures vont se faire fréquemment sur les mêmes blocs, qui vont s'user et donc devenir inutilisables.

Le Static Wear Leveling

La technique utilisée est bien plus complexe. Le contrôleur enregistre le nombre d'écritures sur chaque bloc, et la dernière date d'utilisation de celui-ci. Il est donc capable de déterminer la fréquence d'utilisation d'un bloc et son usure. Si on doit écrire une donnée, il va d'abord chercher le bloc qui a subi le moins de cycles. S'il est libre, le contrôleur l'utilise. Par contre, si le bloc contient des données, il va vérifier la dernière fois qu'il a été écrit et déterminer si c'est une donnée statique (pas d'écriture depuis x temps) ou bien dynamique (le bloc a été écrit récemment).

Si c’est une donnée statique, il va la déplacer vers un bloc usé et mettre la nouvelle donnée à sa place. Si c’est une donnée dynamique qui se trouve sur le bloc, il va en chercher un autre. L’intérêt de la technique consiste à placer les données qui ne sont pas souvent écrites sur des blocs usés et de placer les données souvent modifiées sur des blocs qui ont subi peu d’écriture. Cette technologie permet de garder une usure constante sur le support, et de ce fait d’augmenter la durée de vie globale.

Au final, selon les constructeurs, avec les blocs de réserve et la technique du *Static Wear Leveling*, la durée de vie des SSD est supérieure à celle des disques durs. Point intéressant, la durée de vie augmente avec la capacité, étant donné qu’on dispose de plus de blocs pour distribuer les écritures.

De plus, la majorité des pannes sur les supports de stockage vient en général d’une défaillance mécanique, et les SSD sont dépourvus de pièces en mouvement, ce qui les protège évidemment de ce type de panne.

La capacité

La mémoire flash, comme la mémoire RAM, offre une capacité qui est une puissance de deux. Une puce de 8 « Go » offre en fait une capacité réelle de 8 Gio, soit 8 589 934 592 octets. Dans les SSD, une partie de cette mémoire est réservée, c'est ce que l'on appelle l'overprovisioning. Cette mémoire réservée sert à plusieurs choses : à la gestion de l'usure, en cas de problèmes quand un bloc est défectueux, etc.

Dans un SSD classique, la mémoire réservée est de 7,3 %, c'est la différence entre la taille annoncée en Go (base 10) et en Gio (base 2). Quand une puce offre 8 Go (8 000 000 000 d'octets), elle a une capacité réelle de 8 Gio (8 589 934 592 octets). Certains constructeurs jouent plus ou moins sur les mots et il n'est pas rare de trouver des modèles de 120 Go qui n'offrent pas réellement 120 000 000 000 octets.

Avec certains contrôleurs, notamment le SandForce, la mémoire réservée est plus grande.

TRIM

La commande TRIM est une solution à la perte de performances induite par le fonctionnement des SSD. Comme nous l'avons vu, la gestion de l'usure implique de devoir chercher une cellule vide ou à défaut une cellule peu usée. Le problème, c'est que le contrôleur n'est pas capable de déterminer directement si une cellule est vide, étant donné que les systèmes de fichiers n'effacent pas réellement les données. La solution est simple : implémenter un moyen de permettre au contrôleur de déterminer si des données ont été effacées.

Sitôt dit, sitôt fait, tout du moins quand toute la chaîne est compatible. Concrètement, la commande TRIM consiste à indiquer au SSD que les données sont effacées, contrairement à la technique habituelle qui consiste à marquer les données supprimées sans les effacer réellement et sans que le SSD puisse le savoir. Avec le TRIM, le SSD est capable de voir où sont les données effacées et donc évite de chercher à les trouver. L'intérêt, c'est que passer la commande est simple : elle ne nécessite pas de ressources spécifiques. En effet, c'est un simple indicateur qui va permettre au SSD de mettre à jour sa table interne et lui permettre de connaître le statut d'une cellule. Il n'y a pas d'effacement physique des données, le fonctionnement au niveau du système d'exploitation ne change pas, mais le contrôleur du SSD sait maintenant si une cellule est effacée (logiquement).

Le principal problème du TRIM, c'est que toute la « chaîne » doit supporter la commande. On doit donc avoir un système d'exploitation qui supporte le TRIM, un pilote de contrôleur SATA qui accepte la commande et un SSD qui l'interprète. Si pour le premier et le dernier point, c'est assez simple — tous les SSD récents supportent le TRIM et WIndows 7 aussi — c'est plus gênant pour le second. En effet, certains pilotes ne laissent pas passer la commande, ce qui bloque la commande. Si Intel ou tout simplement Microsoft proposent des pilotes compatibles, ce n'est pas nécessairement le cas sur d'autres contrôleurs. De plus, les contrôleurs RAID ne laissent pas passer la commande TRIM et les systèmes à base de RAID0 ne peuvent donc pas tirer parti de la commande.

Avant le TRIM, Indilinx proposait une solution intermédiaire, qui avait l'avantage d'être utilisable avec la majorité des Windows, le Wiper. Le **Wiper** est un petit programme, utilisable sur certains SSD (en fonction du firmware) qui sert en fait à synchroniser le SSD et le système d'exploitation. Cas simple : vous avez un SSD de 32 Go, que vous avez rempli avec votre système d'exploitation et un fichier provenant d'un DVD que vous avez compressé. Sans le TRIM, supprimer le fichier est une action que le SSD ne verra pas, pour lui, il est toujours là. Le Wiper sert à faire correspondre la base du système de fichier avec la base du SSD, pour lui faire comprendre que le fichier a été supprimé. Défaut de la technique (outre des bugs sur les Windows 64 bits), il faut lancer le programme périodiquement, ce n'est pas automatique. De plus, le Wiper n'est utilisable qu'en NTFS (une version Linux, en bêta, existe malgré tout). Dans la pratique, c'est plus une solution pour revenir à la normale qu'une façon pérenne de corriger le problème.

Les optimisations

Première chose à faire, passer à un système récent comme Windows 7. En effet, ce dernier est capable de détecter les SSD, tout du moins les modèles récents, et il applique quelques optimisations. La première, c'est la désactivation de la défragmentation automatique, car le fonctionnement intrinsèque des SSD fait que la fragmentation du système de fichiers n'a pas d'impact réel sur les performances. La seconde, c'est l'intégration du TRIM au niveau du système lui-même. La troisième, c'est l'alignement des partitions.

Un SSD travaille, comme nous l'avons vu, avec des pages de 2 ou 4 ko en interne, alors que les disques durs classiques utilisent généralement des secteurs de 512 octets en interne — les dernières générations passent au 4 ko aussi. Avec un système d'exploitation classique, le début d'une partition est positionné à 63 octets du « début » du périphérique de stockage, ce qui provoque un décalage entre la gestion interne du périphérique et la gestion du système de fichier. Avec un alignement à 63 octets, les données sont à cheval sur deux pages une fois écrites sur le SSD, ce qui ralentit les opérations de lecture et d'écriture. Avec Windows 7 et les dernières versions de Mac OS X, Linux et les utilitaires récents, la première partition commence à une position multiple de 1 024 octets, ce qui permet de « synchroniser » le système de fichier et la structure interne du périphérique.

Avec certains « vieux » SSD, Windows 7 n'arrive pas à déterminer si le périphérique de stockage est un SSD. En effet, la détection est simple habituellement : une commande de la norme ATA-8 envoie la vitesse de rotation, et elle est à 0 si c'est un SSD. Mais sur les anciens modèles, qui datent d'avant 2009, la commande n'est pas nécessairement implantée. Dans ce cas précis — rare —, il faut appliquer les optimisations à la main. De même, pour les personnes encore sous Windows XP, Windows Vista ou une ancienne version de Linux, il est nécessaire d'optimiser quelques points à la main. Le premier, c'est vérifier que la partition est alignée et le cas échéant le faire. Le second point, c'est éviter la défragmentation. Le dernier, c'est d'utiliser un outil de TRIM manuel, comme le Wiper d'Indilinx, de façon régulière.

Les autres optimisations que l'on voit parfois sur certains sites Internet n'ont généralement pas d'impact réel et sont parfois contre-productives. Certains conseillent par exemple de désactiver le cache du SSD, ce qui a un impact très net sur les performances : elles s'effondrent.

Température

Comme tous les semi-conducteurs, la mémoire flash chauffe quand elle fonctionne. Et comme tous les semi-conducteurs, elle fonctionne mieux une fois refroidie. Maintenant, entendons-nous bien : les SSD chauffent, c’est un fait, mais sans excès. Il est parfaitement possible de mettre sa main sur un SSD après un test, chose impossible sur un disque dur très rapide comme le Raptor. Les dernières générations de contrôleurs, plus complexes, chauffent un peu plus mais aucun ne nécessite pour le moment de radiateur : seuls quelques modèles (chez Toshiba, notamment) sont couverts d'un pad thermique pour transmettre la chaleur au boîtier du SSD, généralement en aluminium.

Consommation

Autant les premiers SSD offraient un gain en autonomie substantiel, autant les SSD récents sont très proches des disques durs. Il y a deux raisons : la quantité de mémoire flash augmente, ce qui a un impact direct sur la consommation, et les disques durs s'améliorent avec le temps.