

AltiVec

AltiVec est un ensemble d'instructions SIMD d'opérations en virgule flottante conçu par, et propriété de, **Apple**, **IBM** et **Motorola** (l'alliance AIM), et mis en application sur des versions du PowerPC telle le **G4** de Motorola et le **G5** d'IBM. AltiVec est un nom commercial détenu uniquement par Motorola ; ainsi l'ensemble est également appelé ***Velocity Engine*** par Apple et ***VMX*** par IBM.

Sommaire

Origines
Principes de fonctionnement <div> <div>Jeu d'instructions</div> <div>Exemple d'utilisation</div> </div>
Applications
AltiVec et l'industrie
Le futur d'AltiVec
Voir aussi <div> <div>Notes et références</div> <div>Liens externes</div> </div>

Origines

À la suite des performances démontrées dans le calcul de l'architecture **SIMD**, avec le supercalculateur *Cray-1* en 1976, ce type d'architecture devient une technique importante dans le domaine du calcul vectoriel et plus généralement matriciel.

En 1993, le processeur IBM/Motorola PowerPC 603 permet déjà d'exécuter des instructions spécialisées dans le calcul vectorielle de type multiplication + addition ou soustraction sur des registres flottants, appelée *fmuladd* et *fmulsub*, le système de pipeline permet de compenser les quatre cycles d'horloge nécessaire à son exécution en autorisant un lancement d'exécution par cycle¹.

Au milieu des années 1990, différents fabricants de micro-processeurs s'intéressèrent au développement d'unités de calculs parallèles qui permettraient d'alléger la charge du processeur en traitant de front de grandes quantités de données. En effet, l'essor des activités multimédia avait mis en évidence les besoins en puissance de ces applications par rapport à des applications classiques comme la bureautique. Les premiers essais de HP et de Sun Microsystems² permirent de définir les bases des unités de traitement SIMD dans le cadre de telles applications et ouvrirent la voie à Intel, AMD et à Motorola. Ainsi, en 1997, AMD sort 3DNow! et Intel, finalise son extension MMX³ qui permettait de travailler sur des données 64 bits via une unité de calcul vectoriel. Cette extension se développa pour donner naissance aux unités SSE, SSE2, SSE3 et SSE4 qui, en reprenant la base de MMX, autorisent le travail sur des flottants simple ou double précision dans le cadre d'applications multimédia comme la compression vidéo ou la synthèse d'images 3D.

Pourtant, malgré l'avancée que représentait MMX et de ses successeurs en termes de performances, cette technologie semblait être sous-exploitée dans les communautés de développeurs. En effet, des problèmes comme la non-orthogonalité de leurs jeux d'instructions, les problèmes d'alignement mémoires et les problèmes d'adéquations entre les structures de données et les limitations de ces unités ralentissaient grandement le développement d'application de grande envergure. Parallèlement aux travaux d'Intel et en constatant les problèmes auxquels les fabricants de microprocesseurs se heurtaient, Motorola s'allia en 1996 à Apple pour concevoir la future architecture du PowerPC G4. Keith Diefendorf, alors chef du projet AltiVec, définit les grandes lignes de cette unité de calcul en n'en faisant pas qu'une simple extension multimédia, mais plutôt une unité de calcul polyvalente. Tirant parti des expériences de ses concurrents, Motorola décida donc de reprendre à zéro la conception matérielle de ce type d'unité de calcul. Finalement, en 1999, le PowerPC G4 d'Apple se vit doté de l'unité de calcul AltiVec de Motorola. En 2002, IBM intègre AltiVec dans les processeurs qui équipent les PowerPC G5.

Lorsqu'il a été présenté la première fois, à la fin des années 1990, AltiVec était le système SIMD le plus puissant, pour un processeur d'une unité centrale de bureau. Comparé à ses contemporains (MMX d'Intel, en nombre entier uniquement ; SSE en virgule flottante, et divers autres systèmes de fournisseurs de processeurs RISC), AltiVec offrait plus de registres, qui pouvaient être utilisés de façons plus variées et être opérés par un ensemble d'instructions beaucoup plus flexible. Cependant, SSE2, le système SIMD d'Intel de quatrième génération, présenté avec le Pentium 4, propose un grand nombre de fonctionnalités qui viennent combler les différences entre ces technologies.

Principes de fonctionnement

AltiVec est basé sur une implémentation intra-processeur du modèle SIMD aussi nommé SWAR pour SIMD Within A Register (terme inventé en 1996 par l'Université de Purdue, pour différencier le SIMD interne au processeur du SIMD externe au processeur, c'est-à-dire les machines parallèles). Dans ce modèle, la manipulation de structures de données spéciales permet de reproduire le comportement d'une unité SIMD au sein d'un seul processeur⁴. Ces types de données nommés **vecteurs** sont des blocs de 128 bits pouvant contenir des données de plusieurs types et sont traités par des opérateurs spécifiques qui effectuent leurs calculs simultanément sur tous les éléments du vecteur. Un jeu de 32 registres vectoriels 128 bits permettent alors de stocker 16 entiers 8 bits signés ou non signés, 8 entiers 16 bits signés ou non signés, 4 entiers 32 bits signés ou non signés ou bien 4 nombre réels 32-bit en virgule flottantes. Contrairement à SSE2, AltiVec ne propose pas de support pour les nombres réels double précision.

Au sein de l'architecture du PowerPC, AltiVec est complètement indépendante des unités de calcul classiques. Elle possède quatre unités de traitement équipées de pipelines qui lui permettent de traiter plusieurs opérations sur des flottants simple précision, des opérations de permutations, des opérations simples sur des entiers (comme l'addition) ou des opérations composites sur des entiers (comme les multiplications partielles) de manière efficace. Elle est en outre capable de démarrer simultanément l'exécution de plusieurs opérations dans plusieurs unités de calcul, on parle alors d'unité de calcul superscalaire. L'utilisateur a en outre la possibilité d'utiliser l'ensemble des registres normaux fournis par le PowerPC et les 32 registres vectoriels fournis par AltiVec de manière simultanée.

Jeu d'instructions

Les versions récentes des outils de développement, tels GNU Compiler Collection, VisualAge d'IBM et d'autres compilateurs, fournissent le support direct aux instructions d'AltiVec depuis des programmes en C et C++. En pratique, 162 primitives C sont disponibles et se répartissent en quatre grands groupes :

- Les instructions de chargement** qui permettent de charger une ou plusieurs valeurs de la mémoire vers un registre vectoriel ou d'un registre vers la mémoire.
- Les instructions de pré-chargement** ou de *prefetch* qui autorisent la manipulation directe des informations présentes dans le cache de manière à optimiser leur ordre de traitement et de ce fait augmenter les performances globales de l'application.
- Les instructions de manipulation des vecteurs**. Certainement les fonctions les plus puissantes d'AltiVec, on trouve dans cette catégorie toutes les instructions de décalages de bits, de permutations d'éléments, de fusion, de duplication de données vectorielles.
- Les instructions arithmétiques** qui permettent d'effectuer classiquement des opérations telles que l'addition, la soustraction ou des opérations sur les nombres en virgule flottante (arrondis ou troncature) sur des vecteurs. AltiVec fournit aussi un certain nombre d'instructions émulant des fonctions de type DSP, des opérations booléennes ou de comparaisons.

La principale différence entre le jeu d'instructions fourni par AltiVec et celui de SSE2 est qu'AltiVec fournit un grand nombre d'opérations dites *horizontales*, permettant de réaliser des *réductions* c'est-à-dire des opérations qui s'appliquent non pas aux éléments de deux vecteurs mais au contenu d'un seul vecteur. Par exemple, la fonction **vec_sums** effectue la somme des éléments d'un vecteur.

Exemple d'utilisation

L'utilisation d'AltiVec au sein d'un code C est relativement simple. Considérons par exemple une fonction C calculant la différence de deux images de NxN pixels. Il s'agit ici d'effectuer la soustraction des intensités des pixels et d'y appliquer un seuillage pour obtenir une image en noir et blanc.

```
-----
unsigned char I1[N*N], I2[N*N], R[N*N];
for(size_t i = 0; i < N*N; i++)
{
    signed short s = I2[i] - I1[i];
    R[i] = (s<10 || s>240) ? 0 : 255;
}
-----
```

Le code C AltiVec équivalent est le suivant :

```
-----
unsigned char I1[N*N], I2[N*N], R[N*N];
vector unsigned char vS,vR,v240,v10,v0,V255;
vector bool char vC;
v0 = vec_splat_u8(0);
v10 = vec_splat_u8(10);
v240 = (vector unsigned char)vec_splat_s8(-15);
v255 = (vector unsigned char)vec_splat_s8(-1);
for(size_t i = 0; i < (N*N)/16; i++)
{
    vS = vec_sub(vec_ld(I2,16*i), vec_ld(I1,16*i));
    vC = vec_or(vec_cmplt(vS, v10), vec_cmpgt(vS, v240));
    vR = vec_sel(v255, v0, vC);
    vec_st(vR, R, i*16);
}
-----
```

Plusieurs points sont à noter :

- Les variables vectorielles sont déclarées en utilisant la classe de stockage de type *vector*. Ainsi, pour un vecteur contenant des éléments d'un type *T*, le type correspondant est *vector T*;
- Les primitives *vec_ld* et *vec_st* remplacent les accès aux éléments du tableau en effectuant respectivement le chargement ou l'écriture de blocs;
- Chaque primitive AltiVec traite les données par blocs de 16 éléments, ce qui implique que la boucle principale ne doit plus effectuer NxN itérations, mais seulement NxN/16;
- Les constantes vectorielles sont générées hors de la boucle principale pour des raisons de performances;
- Les tests nécessaires au seuillage sont aussi vectorisés grâce à la primitive *vec_sel* qui permet d'effectuer l'équivalent de 16 sélections en trois cycles seulement.

L'avantage principal de ces fonctions est leur facilité d'utilisations due à une surcharge transparente pour l'ensemble des types vectoriels, contrairement au instructions disponibles sur les architectures Intel ou le nom de la fonction est modifiée en fonction du type du vecteur. Une liste exhaustive des fonctions disponibles est accessible sur la page *AltiVec Instructions Cross References* de Apple (https://developer.apple.com/hardware/drivers/ve/instruction_crossref.html).

Applications

Apple est le premier client d'AltiVec, et l'utilise pour accélérer les applications de multimédia telles QuickTime et iTunes, et les programmes de traitement d'images tels Photoshop d'Adobe. Motorola a fourni des processeurs AltiVec pour toutes les unités centrales de bureau depuis l'introduction du G4. AltiVec est également employé dans quelques systèmes embarqués, pour fournir de grandes performances lors du traitement numérique du signal.

AltiVec et l'industrie

Depuis le retrait d'Apple du monde PowerPC et AltiVec, les principaux constructeurs de machines à base de powerPC AltiVec sont Thales Computer et Mercury Computer.

Le futur d'AltiVec

Bien qu'ayant disparu des dernières machines d'Apple, AltiVec continue d'exister dans des domaines moins grand public comme les stations de travail et serveurs de calcul d'IBM à travers le PowerPC 970 MP (la dernière version à avoir équipé les PowerMac G5) et à travers le POWER6 d'IBM. AltiVec continue aussi d'exister dans l'un des domaines grand public à très forte croissance: les consoles de jeux. Ainsi la XBOX 360 de Microsoft est un processeur triple cœur AltiVec dérivant du 970 MP, et la PS3 et son fameux processeur Cell, un octo-processeur piloté par un processeur maître (le PPE) très proche du 970. L'octo-processeur contient lui aussi un jeu d'instruction SWAR nommé SPE (*Synergistic Processing Elements*) dont le jeu d'instruction est très proche d'AltiVec.

Voir aussi

Notes et références

- (en) PowerPC 603e and EM603e RISC Microprocessor Family User's Manual (https://www-01.ibm.com/chips/techlib/techlib.nsf/techdocs/852569B20050FF7785256996006F9795/\$file/603e_um.pdf), sur IBM.com – chapitre « 2.3.4.2.2 Floating-Point Multiply-Add Instructions », page 2-26 et chapitre « 6.4.3 Floating-Point Unit Execution Timing », page 6-17
- (en) N. Slingerman and A. J. Smith - « **Performance analysis of instructions set architecture extensions for multimedia** » (http://pdcl.wayne.edu/msp01/paper7.pdf)^{[Archive (http://web.archive.org/web/*/http://pdcl.wayne.edu/msp01/paper7.pdf) • Wikiwix (http://archive.wikiwix.com/cache?url=http://pdcl.wayne.edu/msp01/paper7.pdf) • Archive.is (http://archive.is/http://pdcl.wayne.edu/msp01/paper7.pdf) • Google (https://www.google.fr/search?q=cache:http://pdcl.wayne.edu/msp01/paper7.pdf) • Que faire ?} - 2001
- (en) A. Peleg and U. Weiser - **MMX technology extension to the Intel architecture** (http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel1/40/11116/00526924.pdf?arnumber=526924) - 1996, IEEE Micro 4(16), p. 42-50
- (en) K. Diefendorf - **AltiVec extension to Power PC accelerates media processing** - 2001

Liens externes

- (en) Page AltiVec chez Freescale (http://www.freescale.com/webapp/sp5/site/overview.jsp?nodeId=018rH3bTdGmKqW5Nf2)
- (en) Page *Velocity Engine* chez Apple (https://developer.apple.com/hardware/ve/)
- (en) Liste des fonctions C AltiVec (https://developer.apple.com/hardware/drivers/ve/instruction_crossref.html)
- (en) Mercury Computer Systems (http://www.mc.com)
- (en) Thales Computer (http://www.thalescomputers.com)

Ce document provient de « https://fr.wikipedia.org/w/index.php?title=AltiVec&oldid=150742591 ».

La dernière modification de cette page a été faite le 27 juillet 2018 à 23:00.

Droit d'auteur : les textes sont disponibles sous licence Creative Commons attribution, partage dans les mêmes conditions ; d’autres conditions peuvent s’appliquer. Voyez les conditions d’utilisation pour plus de détails, ainsi que les crédits graphiques. En cas de réutilisation des textes de cette page, voyez comment citer les auteurs et mentionner la licence.
Wikipedia® est une marque déposée de la Wikimedia Foundation, Inc., organisation de bienfaisance régie par le paragraphe 501(c)(3) du code fiscal des États-Unis.