

新「小原のアルゴリズム」に基づく二電子積分計算専用LSIについて

戸川勝巳[†] 小原繁^{††} 上原正光^{†††} 佐藤比佐夫^{†††} 波多江秀典[†] 中村健太^{††††} 村上和彰[†]

[†]九州大学 大学院システム情報科学府 情報工学専攻

^{††}北海道教育大学 教育学部

^{†††}セイコーエプソン 株式会社

^{††††}九州大学 工学部 電気情報工学科

[†]九州大学 大学院システム情報科学研究所 情報理学部門

E-mail: ehpc-lsi@star.fuji-ric.co.jp

非経験的分子軌道法計算を高速に処理することを目的として、分子軌道法専用ロジックの開発を行っている。問題サイズ N に対して演算量が N^4 である二電子積分計算を高速に計算することが重要である。「小原のアルゴリズム」より二電子積分は積和演算で計算できるが、このアルゴリズムの特徴を利用することにより高速処理が可能となる。本稿では、浮動小数点積和演算器を複数個集積し、浮動小数点除算、開平逆数、指数の専用演算回路を搭載した VLIW アーキテクチャに関して、その性能評価を行う。4 個の積和演算器を搭載した 4-way の場合; 1-way (1 個の積和演算器) と比較すると、約 2.5 倍の二電子積分計算の速度向上を達成できた。

Custom Logic for *ab initio* Molecular Orbital Calculation

Katsumi TOGAWA[†] Shigeru OBARA^{††} Masamitsu UEHARA^{†††}
Hisao SATO^{†††} Hidenori HATAE[†] Kenta NAKAMURA[†]
Kazuaki MURAKAMI[†]

[†]Department of Computer Science and Communication Engineering, Kyushu University

^{††}Hokkaido University of Education

^{†††}Seiko Epson Corporation

E-mail: ehpc-lsi@star.fuji-ric.co.jp

We are developing a custom logic for *ab initio* Molecular Orbital Calculation to reduce the calculation time. The execution time is $O(N^4)$ where N is the number of basis functions, and most of the time is spent to the calculations of two-electron integrals. Using characterization of two-electron integrals in the "Obara method", it is possible to reduce the calculation time. This paper evaluates the performance of VLIW (Very Long Instruction Word) architecture with a lot of floating-point (FP) Multiply and Adder Units and FP divide, FP the reciprocal of square-root and FP exponential units. It is observed that the speedup achieved by having four Mul/Add unit for the calculations of two-electron integrals is about 2.5 times, compared to having a Mul/Add unit.

1 はじめに

近年の計算機の飛躍的な向上により、分子シミュレーションのような大規模科学技術計算が盛んに行われるようになった。特に、非経験的分子軌道法は、新素材、高性能材料、医薬品等の開発に利用されており、これからもますます発展する分野である。

しかしながら、非経験的分子軌道法は、基底関数

の数の 4 乗に比例する演算量と補助記憶量を必要とする。そのため、現在のシステムでは、性能とコストの面で、ユーザが満足のいくシミュレーションが行われていない。物質科学のさらなる発展を考えると、非経験的分子軌道法の飛躍的な高速化と計算コストの低減化は不可欠である。その実現のためには、

非経験的分子軌道法を高速に実行する専用計算機システムの開発が必要である [1][2]. 筆者らは、非経験的分子軌道法的全計算時間の90%を占める二電子積分計算を高速に処理することを目的として、分子軌道法専用ロジックの開発を行っている。

二電子積分は、小原のアルゴリズムを適用することにより、漸化計算で表され、積和演算で計算できる。また、この積和演算には多くの命令レベル並列性が存在するため、この命令レベル並列性を利用することで高速演算が可能である。浮動小数点除算、開平逆数、指数がクリティカル・パスとなり命令レベル並列性が多く存在しない部分では、それらの演算の専用回路をロジック上に搭載刷ることで高速処理が可能となる。

そこで、本稿では、二電子積分計算における命令レベル並列性に着目し、二電子積分専用ロジック開発のためのさまざまな評価を行う。以下、第2章では、非経験的分子軌道法と「小原のアルゴリズム」[3]による二電子積分計算の概要を説明する。次に、第3章では二電子積分計算のさまざまな特徴について述べる。第4章では性能評価と考察を行う。最後に、第5章で簡単にまとめる。

2 計算アルゴリズム

本章では、非経験的分子軌道法の概要を説明し、その中で最も計算時間を費やす二電子積分について述べる。

2.1 非経験的分子軌道法

非経験的分子軌道法の解法には、ハートリーフォック法(HF法)を用いる [5]. フォック行列の二電子積分寄与部分は、基底関数のインデックス r, s, t, u を用いると以下ようになる。

$$F'_{rs} = \sum_{t=1}^N \sum_{u=1}^N P_{tu} \left\{ (rs, tu) - \frac{1}{2}(rt, su) \right\} \quad (1)$$

(1) 式の P_{tu} を密度行列、 (rs, tu) を二電子積分と呼ぶ。非経験的分子軌道法計算において、二電子積分計算は、基底関数の数 N の4乗に比例する計算量が必要であるため、最も計算時間を費やす部分である。したがって、二電子積分計算を高速に処理することが重要である。

2.2 二電子積分計算のアルゴリズム

二電子積分計算のアルゴリズムにはいくつかの方法が知られているが、「小原のアルゴリズム」に改良を加えられた新「小原のアルゴリズム」を採用する。二電子積分を決定させる4つのガウス型関数それぞれの軌道量子ベクトルを a, b, c, d とおき、二電子積分を (ab, cd) と表す。

二電子積分 (ab, cd) は、補助積分と呼ばれる物理量 $(ab, cd)^m$ (m は0以上の整数) を用いて、ガウス型関数を基盤とした以下の漸化関係式にて表される。

$$(ab, cd) = (ab, cd)^{(0)} \quad (2)$$

$$\begin{aligned} ((a + 1_\mu)b, cd)^{(m)} = & \\ & C0 \times (ab, cd)^{(m)} + C1 \times (ab, cd)^{(m+1)} \\ & + C2 \times ((a - 1_\mu)b, cd)^{(m)} \\ & + C3 \times ((a - 1_\mu)b, cd)^{(m+1)} \\ & + C4 \times ((a(b - 1_\mu), cd)^{(m)} \\ & + C5 \times (a(b - 1_\mu), cd)^{(m+1)} \\ & + C6 \times (ab, (c - 1_\mu)d)^{(m+1)} \\ & + C7 \times (ab, c(d - 1_\mu))^{(m+1)} \end{aligned} \quad (3)$$

ここで、 $C0 \sim C7$ は軌道指数や原子の中心座標によって決まる係数、 μ は3次元軌道量子ベクトルの値、 $a_\mu, b_\mu, c_\mu, d_\mu$ はそれぞれ a, b, c, d の μ 成分である。 1_μ とは、 μ 成分の時に1となり、その他の成分の時は0になるベクトルである。なお、軌道量子ベクトルの3成分の和を軌道量子数と呼び、その値が0, 1, 2の軌道をそれぞれ、s軌道、p軌道、d軌道と呼ぶ。これらの軌道を用いて、二電子積分は (pp, ss) , (dd, ds) と表記される。4つの軌道量子数により識別した積分のことを積分タイプと呼ぶ。

小原の方法の特徴は、まず入力データからいくつかの初期パラメータを計算し、初期積分 $((ss, ss)$ 型の積分) を求めた後、それらを用いて、目的の二電子積分まで漸化計算により求める点にある。したがって、二電子積分計算は、初期積分計算と漸化計算の2つに大きく分けられる。

2.3 初期積分と漸化計算の計算時間

5つのペプチド分子G, GA, GAQ, GAQM, GAQMYにおいて、「小原のアルゴリズム」による二電子積分計算に要する計算時間(second)を表1に示す。基底関数系とは基底関数のセットのことで、つまり、基底

表 1: 二電子積分の計算時間 (second(%))

分子	G	GA	GAQ	GAQM	GAQMY
基底関数系	/631/31/ for C, O, N, /31/ for H				
基底関数の数	55	110	207	316	427
初期積分	3.98(89.0%)	48.04(88.6%)	441.0(87.1%)	2123.4(86.2%)	5200.3(84.5%)
漸化計算	0.36(8.1%)	4.26(7.9%)	41.6(8.2%)	211.6(8.6%)	573.1(9.3%)
その他	0.13(2.9%)	1.94(3.6%)	23.7(4.7%)	127.0(5.2%)	378.3(6.1%)
全計算時間	4.47	54.24	506.3	2462.0	6151.7
基底関数系	/631/31/1/ for C, O, N, /31/ for H				
基底関数の数	85	170	321	484	661
初期積分	8.41(60.7%)	104.49(58.0%)	873.5(54.6%)	4337.1(53.8%)	10420.9(54.3%)
漸化計算	5.12(37.0%)	71.15(39.5%)	673.7(42.1%)	3439.7(42.7%)	8046.9(41.9%)
その他	0.32(2.3%)	4.64(2.6%)	52.6(3.3%)	284.5(3.5%)	739.8(3.9%)
全計算時間	13.85	180.28	1599.8	8061.4	19207.6
基底関数系	/631/31/11/ for C, O, N, /31/ for H				
基底関数の数	115	230	435	652	895
初期積分	15.51(40.6%)	213.92(39.2%)	1769.2(36.1%)	8986.7(36.6%)	12619.0(35.2%)
漸化計算	22.00(57.5%)	322.61(59.1%)	3032.2(61.9%)	15091.9(61.4%)	22575.3(62.9%)
その他	0.72(1.9%)	9.04(1.7%)	93.5(1.9%)	496.3(2.0%)	704.4(2.0%)
全計算時間	38.23	545.57	4895.0	24575.0	35898.7

関数に何を用いるかを示したものである。また、それぞれの分子で異なる基底関数系を用いた場合の計算時間も表に示す。/631/31/は、s, p型の基底関数を持つ。/631/31/1/は、/631/31/にd型の基底関数を1つ付加したものであり、s, p, d型の基底関数を持つ基底関数系である。/631/31/11/は、s, p型と2つのd型の基底関数を持つ基底関数系である。なお、計測はSun Enterprise 420R(CPU: UltraSPARC-II 450MHz, 主記憶:4GB)上で行い、コンパイルはSun WorkShop Compiler C SPARC(version 5.000)を用い、-xO3の最適化オプションを使用した。この結果より、以下のことが確認できる。

- s, p型だけの基底関数の場合、初期積分計算の割合は80%以上、漸化計算の割合は10%以下
- d型の基底関数を1つ含む場合、初期積分計算の割合は50%以上、漸化計算の割合は50%以下
- d型の基底関数を2つ含む場合、初期積分計算の割合は40%以下、漸化計算の割合は60%以上

つまり、d型の基底関数を含む場合、漸化計算の割

合が大きくなり、d型の基底関数の数が増加するに伴い、漸化計算の割合が増加することが分かる。

有機化合物において、計算精度を上げるためにd型の基底関数を1つ付加したものを基底関数として用いることが一般的である。したがって、d型の基底関数を含めた計算を高速にすることが重要であり、初期積分計算、漸化計算の両方を高速に処理することが必要となる。

3 二電子積分計算の特徴

二電子積分計算は、小原のアルゴリズムにより、初期積分計算部分と漸化計算部分から成る。本章では、初期積分計算と漸化計算の特徴についてそれぞれ述べる。

3.1 初期積分計算

初期積分計算は図1のように4重ループの構造となる。なお、 N_i , N_j , N_k , N_l は縮約項数と呼ばれるもので、基底関数系を6-31Gとすると N_i , N_j , N_k , N_l の値は、1, 3, 6のいずれかである。初期積分計算は大きく9個のブロックに分けることがで

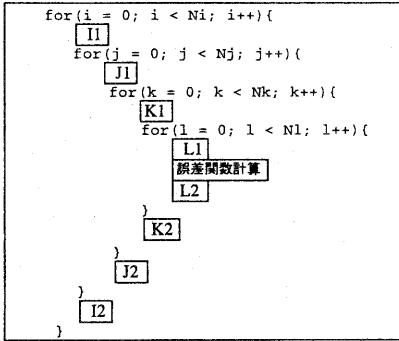


図 1: 初期積分計算におけるループ構造

表 2: I1, J1, K1, L1 ブロックにおける浮動小数点演算数

	加減算	乗算	除算	開平逆数	指数
I1	0	6	0	0	0
J1	5	8	2	0	1
K1	1	5	0	0	0
L1	4	9	3	1	1

きる。ここでは便宜上、各ブロックを I1, J1, K1, L1, 誤差関数, I2, J2, K2, L2 と名付ける (図 1 参照)。

表 2 は、I1, J1, K1, L1 ブロックにおける浮動小数点演算数を示す。これらのブロックは、演算数は少ないが、除算、開平逆数、指数の複雑な算術演算を含んでいる。さらに、除算、指数がクリティカル・パス上に存在している。また、誤差関数でも開平逆数が現れる。したがって、除算、開平逆数、指数演算の高速化が重要となる。

I2, J2, K2, L2 ブロックにおける積分タイプ毎の命令レベル並列度を示したものが図 2 である。命令レベル並列度とは、プロセッサ資源に関する制約が一切ないと仮定した場合のプログラムに内在する命令レベルの平均の並列度である。命令レベル並列度は (ss, ss) から (dd, dd) に向かうに従い、大きくなっていることが分かる。

3.2 漸化計算

4 つの軌道量子数のうち少なくとも 1 つが 1 以上である補助積分を求める場合、(3) 式の漸化関係は、以下のような積和演算を繰り返すことにより計算できる。

$$\langle \text{左辺積分} \rangle = \frac{+}{-} \langle \text{係数} \rangle \times \langle \text{右辺積分} \rangle \quad (4)$$

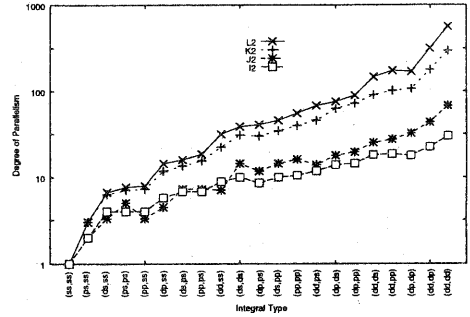


図 2: I2, J2, K2, L2 ブロックにおける命令レベル並列度

ただし、(3) 式の左辺の $((a + 1_\mu)b, cd)^{(m)}$ を左辺積分、右辺の $(ab, cd)^{(m)}$, $(ab, cd)^{(m+1)}$ 等を右辺積分とする。漸化計算に含まれる積和演算数、クリティカル・パス長、命令レベル並列度を表 3 に示す。これより漸化計算には多くの命令レベル並列性が存在することが確認できる。

4 性能評価

本章では、積和演算器を複数個集積し、かつ、浮動小数点除算、開平逆数、指数の専用回路を搭載した VLIW アーキテクチャに関して、二電子積分計算の性能評価を行う。

4.1 評価モデル

二電子積分に内在する命令レベル並列性を利用するため、分子軌道法専用ロジックには、VLIW (Very Long Instruction Word) アーキテクチャを採用。評価モデルには、表 4 の 2 つのモデルを用いる。つまり、浮動小数点除算、開平逆数、指数をソフトウェアで実装したモデルとそれらの演算を専用回路で実装したモデルである。表 4 の条件のもと、モデル 2 に関してはさらに、1way (積和演算器: 1 個), 2way (積和演算器: 2 個), 4way (積和演算器: 4 個), 8way (積和演算器: 8 個), 16way (積和演算器: 16 個) の 5 つのモデルについて評価を行った。全ての積和演算器は共有のレジスタファイルと接続しており、レジスタファイルのサイズは無限大とする。

評価方法に関しては、二電子積分計算の実行サイクル数が最小となるように、クリティカル・パス法によるリスト・スケジューリング [4] を行い、その結果による実行サイクル数を求めた。この際、二電子

表 3: 漸化計算における積分タイプと並列度の関係

積分タイプ	積和演算数	クリティカルパス長	命令レベル並列度
(ps, ss)	12	7	1.71
(ds, ss)	78	18	4.33
(ps, ps)	87	16	5.44
(pp, ss)	108	19	5.68
(ds, ps)	369	27	13.67
(dp, ss)	418	30	13.93
(pp, ps)	498	28	17.79
(ds, ds)	1365	40	34.12
(dd, ss)	1432	42	34.10
(dp, ps)	1585	39	40.64
(ds, ps)	1740	41	42.44
(pp, pp)	2400	42	57.14
(dd, ps)	4825	51	94.61
(dp, ds)	5028	52	96.69
(dp, pp)	6387	53	120.51
(dd, ds)	13668	64	213.56
(dd, pp)	17179	65	264.29
(dp, dp)	18272	66	276.85
(dd, dp)	43577	78	558.68
(dd, dd)	114283	92	1242.21

積分計算に必要なデータはすべて、レジスタに存在していると仮定している。Result Latency は、オペレーションが終了するのに要するクロックサイクル数を示し、Issue Latency とは、次の命令が発行可能となるまでに要するクロックサイクル数である。

4.2 漸化計算の速度向上比

漸化計算のコード・スケジューリングの結果より得られた所要クロックサイクル数を用いて、モデル 2 の積分タイプ毎の速度向上比を求めた (図 3)。速度向上比は、積分タイプ (ss, ss) から (dd, dd) に向かうに従い増加している。(ss, ss) から (pp, ss) までは、16way の場合でも速度向上比が 2 であるが、(dp, pp) では、16way で速度向上比が 16 に近い値となっている。これは、二電子積分に内在する命令レベル並列度が、積分タイプ (ss, ss) から (dd, dd) に向かって、増加するためである。

表 4: 2つのモデルにおける倍精度浮動小数点演算の Result Latency(RL) と Issue Latency(IL)

浮動小数点命令	model1		model2	
	RL	IL	RL	IL
加減算	2	1	2	1
乗算	4	1	4	1
積和演算	6	1	6	1
除算	20	1	10	10
開平逆数	40	1	20	20
指数演算	60	1	30	30

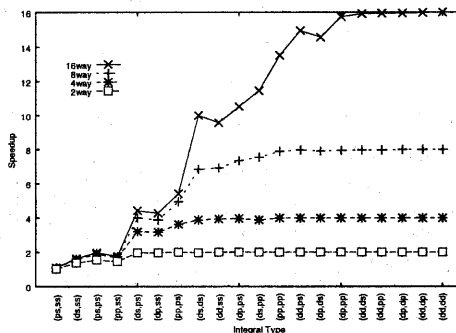


図 3: 漸化計算におけるモデル 2 の速度向上比

4.3 GAQMY 分子に対する二電子積分の計算時間

上記のコード・スケジューリング結果より、実際の二電子積分において、どれくらいの性能が得られるか調べることは重要である。そこで、コード・スケジューリング結果を用い、それぞれ、モデル 1, モデル 2(1, 2, 4, 8, 16-way) についてペプチド分子である GAQMY(基底関数の数: 661, 基底関数系: /631/31/1/ for C, O, N, /31/ for H) の二電子積分計算の所要クロックサイクル数を求めた (図 4)。二電子積分の所要クロックサイクル数は、各積分タイプのクロックサイクル数に積分タイプ毎の個数を乗じたものの和で表される。図 4において、各棒グラフは分割されているが、I1, J1, K1, L1 の 4 つブロック、誤差関数, I2, J2, K2, L2 の 4 つのブロック、漸化計算の所要クロックサイクル数をそれぞれ表している。

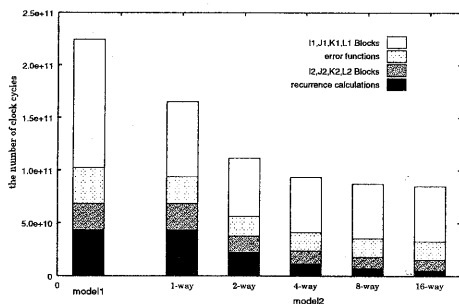


図 4: GAQMY 分子に対する各モデルの計算時間 (クロックサイクル数)

表 5: 速度向上比と I1, J1, K1, L1 ブロックの占める割合 (%)

ウェイ数	1	2	4	8	16
速度向上比	1.36	2.01	2.40	2.57	2.65
I1, J1, K1, L1 ブロックの割合	43.2	49.8	55.9	59.8	61.7

4.4 考察

図 4 より、命令レベル並列度の高い I2, J2, K2, L2 ブロック、及び、漸化計算部分は、ウェイ数に対する所要クロックサイクル数の削減が大きいことが分かる。二電子積分の所要クロックサイクル数より、モデル 1 と比較した場合のモデル 2 の 1, 2, 4, 8, 16-way の速度向上比は、表 5 のようになる。

これより、ウェイ数に対する速度向上比がそれほど高くないことが言える。これは I1, J1, K1, L1 ブロックの命令レベル並列度と関係してくる。表 5 は、これら I1, J1, K1, L1 の 4 つのブロックが総計算時間に占める割合である。これらのブロックが占める計算時間が非常に大きいことが確認できる。しかし、I1, J1, K1, L1 ブロックの速度向上比は、クリティカル・パス上にレイテンシの大きな演算 (浮動小数点除算、浮動小数点指数) が存在するため、ウェイ数に対して非常に低い。16-way においても 1.5 以下である。そのため、全体の計算時間に関する速度向上比も低くなる。

命令レベル並列性が低い I1, J1, K1, L1、及び、誤差関数部分の高速化のためには高速な除算、開平逆数、指数回路が必要であり、並列性の高い I2, J2, K2, L2、及び、漸化計算部分では積和演算器を多

く集積することで高速化が可能である。しかし、積和演算器を 16 個も搭載する必要はなく、4-way が適当であると考えられる。

5 おわりに

非経験的分子軌道法において全計算時間の 90% を占める二電子積分計算の「小原のアルゴリズム」でのさまざまな計算の特徴を調べた。また、いくつかのプロセッサモデルについて、性能評価を行った。浮動小数点除算、開平逆数、指数の専用演算回路を集積し、積和演算を複数個搭載することにより、速度向上が可能となることを確認した。今後は、より細かいアーキテクチャの検討を行う予定である。

謝辞

本研究は一部、平成 12 年度科学技術振興調整費総合研究「科学技術計算専用ロジック組込み型プラットフォーム・アーキテクチャに関する研究」に依る。

参考文献

- [1] 村上和彰, 稲垣祐一郎, 上原正光, 大谷康昭, 小原繁, 小関史朗, 佐々木徹, 棚橋隆彦, 中馬寛, 塚田捷, 長嶋雲兵, 中野達也, “科学技術計算専用ロジック組込み型プラットフォーム・アーキテクチャの開発 - プロジェクト全体像 -,” HPC82-1, 2000 年 8 月.
- [2] 戸川勝巳, 上原正光, 小原繁, 波多江秀典, 村上和彰, “科学技術計算専用ロジック組込み型プラットフォーム・アーキテクチャの開発 - 分子軌道法専用ロジックの検討 -, ” HPC82-3, 2000 年 8 月.
- [3] S.Obara and A.Saika “General recurrence formulas for molecular integrals over Cartesian Gaussian function,” J.Chem.Phys.Vol98 NO.3, Aug 1988
- [4] 村上和彰, “スーパースカラ・プロセッサの性能を最大限に引き出すコンパイラ技術,” NIKKEI ELECTRONICS, no.521, 1991 年 3 月
- [5] 大原寛, 内田希, 生田茂, “分子軌道法,” 講談社, 1994