

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 -

戸川勝巳[†] 上原正光^{††} 小原繁^{†††} 波多江秀典[†] 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

^{††}セイコーエプソン 株式会社

^{†††}北海道教育大学 教育学部

[†]九州大学 大学院システム情報科学研究院 情報工学部門

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。二電子積分に内在する多大な命令レベル並列性、ならびに、二電子積分計算間の並列性を利用することにより、二電子積分計算の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た VLIW アーキテクチャに関して、その性能評価を行う。16 個の積和演算器を搭載した場合、1 個の積和演算器と比較すると、約 13 倍の二電子積分計算の速度向上を達成できた。

A Platform Architecture for Embedded High-Performance

Computing

- Custom Logic for Molecular Orbital Calculation -

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

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

^{††}Seiko Epson Corporation

^{†††}Hokkaido University of Education

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

We are developing a custom logic for 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 high Instruction-Level Parallelism in two-electron integrals and parallelism of two-electron integrals, 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 Multiply and Adder Unit. It is observed that the speedup achieved by having sixteen Mul/Add unit for the calculations of two-electron integrals is about 13 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]. 筆者らは、非経験的分子軌道法の全計算時間の 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。また、二電子積分計算間の並列性を併用することにより、所要クロックサイクル数を大幅に削減できる。

そこで、本稿では、二電子積分計算における積和演算の命令レベル並列性と二電子積分計算間の並列性に着目し、二電子積分専用ロジック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 章では二電子積分に内在する命令レベル並列性について述べる。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 [4]. 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、「小原の方法」[2] 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。

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

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

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。 $\mathbf{1}_\mu$ 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。

2.2.1 初期積分計算

初期積分を $(ss, ss)^{(m)}$ とするとその表式は以下のようになる。

$$(ss, ss)^{(m)} = \text{param}(\zeta; R) \times F_m(T) \quad (4)$$

漸化計算を行うには初期積分中の m の値が、原始積分の軌道量子数の総和以下の非負正数になっているもの全てが必要になる。つまり、 (pp, pp) の場合では、 m の値が 4 となり、 $(ss, ss)^{(0)}$, $(ss, ss)^{(1)}$,

$(ss, ss)^{(2)}$, $(ss, ss)^{(3)}$, $(ss, ss)^{(4)}$ の5つの初期積分が必要となる。param, T は原子の中心座標 R や軌道指数 ζ を用いて計算される。 $F_m(T)$ は誤差関数と呼ばれ、その定義式は以下のとおりである。

$$F_m(T) = \int_0^1 dt \cdot t^{2m} \exp(-Tt^2) \quad (5)$$

表1に初期積分 $((ss, ss)^{(0)})$ 計算に要する各演算の演算数を示す。ここで、式(5)の計算にはテーラー展開を用いる。

表1: 初期積分 $((ss, ss)^{(0)})$ 計算の演算数

演算	加減算	乗算	除算	開閉	指数
演算数	34	72	3	1	2

2.2.2 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。

$$\langle \text{左辺積分} \rangle^+ \langle \text{係数} \rangle \times \langle \text{右辺積分} \rangle \quad (6)$$

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。

3 二電子積分計算

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に内在する命令レベル並列度について述べる。

3.1 命令レベル並列度

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以下の式で与えられる。

$$\text{命令レベル並列度} = \frac{\text{命令数}}{\text{クリティカル・パス長}} \quad (7)$$

上式でのクリティカル・パス長とは、すべての演算を単一サイクル命令としたときの命令列におけるクリティカル・パスのサイクル数である。

表2: 初期積分と命令レベル並列度の関係

mの値	演算数	クリティカル・パス長	命令レベル並列度
0	176	53	3.32
1	183	53	3.45
2	190	53	3.58
3	197	53	3.71
4	204	53	3.85
5	211	53	3.98
6	218	53	4.11
7	225	53	4.25
8	232	53	4.38

表3: 積分タイプと漸化計算の並列度の関係

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)	6	2	3.00
(ds,ss)	30	6	5.00
(ps,ps)	33	5	6.60
(pp,ss)	36	6	6.00
(ds,ps)	99	9	11.00
(dp,ss)	108	10	10.80
(pp,ps)	144	10	14.40
(ds,ds)	315	14	22.50
(dp,ps)	363	14	25.93
(ds,pp)	372	14	26.57
(dd,ss)	378	16	23.63
(pp,pp)	564	16	35.25
(dd,ps)	1056	20	52.80
(dp,ds)	1077	19	56.68
(dp,pp)	1308	20	65.40
(dd,ds)	2874	26	110.54
(dd,pp)	3198	26	123.00
(dp,dp)	3669	25	146.76
(dd,dp)	9156	32	286.13
(dd,dd)	26331	40	658.28

3.2 二電子積分計算に内在する命令レベル並列度

式(7)より求めた初期積分計算に内在する命令レベル並列度を表2に示す。 m の値の増加に伴い、初期積分の命令レベル並列度も増加することが分かる。

漸化計算は積和演算により計算されるが、積和演算数は求める積分タイプ(例えば、 (ps, ss) や (pp, pp) など)により異なる。表3は、積和演算に存在する命令レベル並列度を調べた結果である。なお、表3は、積和演算数の少ない順に積分タイプをソートしている。これより、命令レベル並列度は、積和演算数に比例して増加することが確認できる。つまり、表3において、 (ss, ss) から (dd, dd) に向かって、命令レベル並列度は大きくなっていることが分かる。

表2と表3の結果から、二電子積分計算には、多くの命令レベル並列性が存在することが分かる。この多大な命令レベル並列度を活用することで、二電子積分の計算時間を低減することができる。

4 性能評価

二電子積分計算に内在する命令レベル並列性を利用し、複数個の積和演算器を集積することで、二電子積分の高速演算が可能である。本章では、積和演算器を複数個集積した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る。表4の条件のもと、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、二電子積分計算の実行サイクル数が最小となるように、クリティカル法によるリスト・スケジューリング[3]を行った。この際、二電子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。除算、開閉、指数演算は、Newton法を適用することにより、積和演算器で演算を行う。

表4: 演算のResult LatencyとIssue Latency

演算	Result Latency	Issue Latency
倍精度浮動小数点加減算	2	1
倍精度浮動小数点乗算	4	1
倍精度浮動小数点積和演算	6	1
倍精度浮動小数点除算	10	10
倍精度浮動小数点開閉	10	10
倍精度浮動小数点指数演算	10	10

4.2 速度向上比

スケジューリングの結果より得られた、各積分タイプでの所要クロックサイクルを図1に示す。また、図1より、速度向上比を求めた。速度向上比は、以下の式で表される。

$$\text{速度向上比} = \frac{1\text{wayの所要クロックサイクル数}}{\text{各wayの所要サイクル数}} \quad (8)$$

図2は、上式より求めた積分タイプ毎の速度向上比である。

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、(dd,dd)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。

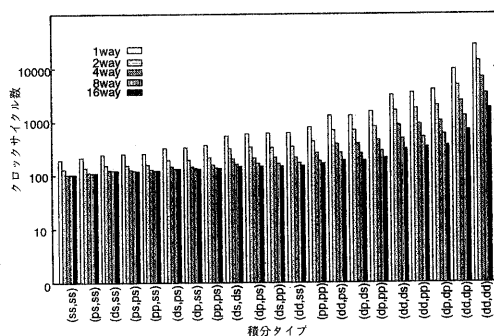


図1: 積分タイプと所要クロックサイクル数の関係

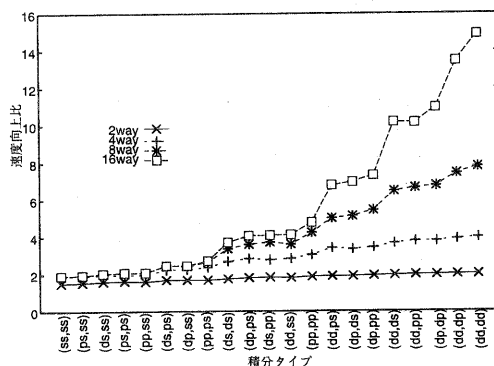


図2: 積分タイプに対する速度向上比

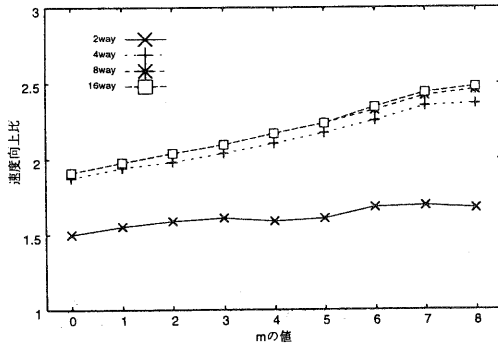


図 5: 初期積分計算の速度向上比

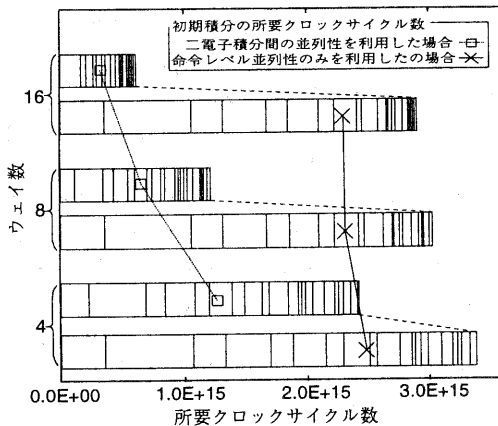


図 6: 二電子積分計算間の並列性を利用した所要クロックサイクル数

期積分の速度向上比は、全ての積分タイプで、16way においても高々2.5に過ぎない(図5)。

- 各積分タイプの速度向上比に最適な資源を提供：速度向上比は積分タイプによって異なる。

4way, 8way, 16way に関して、命令レベル並列性だけでなく、二電子積分計算間の並列性も利用した gramicidin に対する二電子積分の計算時間(クロックサイクル数)を図6に示す。比較のために、二電子積分計算間の並列性を利用していない場合の計算時間も示す。この図より、所要クロックサイクル数が削減できたことが確認できる。表6は、図6より求めた速度向上比である。表5と表6を比較すると、速度向上比が大幅に改善されたことが分かる。

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命令レベル並列度を調べ、

表 6: 積分タイプの並列性を利用した場合の速度向上比

ウェイト数	4	8	16
速度向上比	3.32	6.60	13.13

多くの命令レベル並列性が存在することを確認した。複数の積和演算器に対するコード・スケジューリングを行い、積分タイプ毎のクロックサイクル数を求めた。実際に、gramicidinの二電子積分計算を例とし、コード・スケジューリング結果を用いて、所要クロックサイクル数を求め、それに対する評価を行った。命令レベル並列性だけでなく、二電子積分計算間の並列性を利用することにより、速度向上比が飛躍的に向上することを確認した。

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] S. Obara and A. Saika “General recurrence formulas for molecular integrals over Cartesian Gaussian function,” J. Chem. Phys. Vol. 98 NO. 3, Aug 1988
- [3] 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
- [4] 大原寛, 内田希, 生田茂, “分子軌道法,” 講談社, 1994