レジスタ割付からみたスライドウインドウアーキテクチャの優位性について 3F－3

秡川友宏　添野元秀
山下義行　中田育男

筑波大学

1. はじめに

スライドウインドウアーキテクチャは，元来，繰り返し演算におけるメインメモリのアクセスレーテンシを隠蔽するために提案され[1]，本学と日立製作所が共同開発した超並列計算機 CP-PACS のノードプロセッサとして採用された．ここでは，その詳細は参考文献に譲り，slide +k 命令でレジスタ #0, #1, ... を #(0+k), #(1+k), ... に一斉にリネームできるアーキテクチャと単純化して考えることにする．k は正でも負でもよい．

スライドウインドウ機構を用いると，スケジューラにとっては，ソフトウェア・パイプライン化コードを生成する際にループ立ち上げ間隔（II: Initiation Interval）の算出や命令スケジューリングをごく自然な発想に基づいて素直に実現できるという利点がある．反面，レジスタ割付器にとっては，そのアルゴリズムがレジスタ番号の変化のために複雑になると考えられてきた[2]．しかし，当初の予想とは反対に，単純なアルゴリズムで少ないレジスタに割付できる[3]など，レジスタ割付の面から見ても有利であることが明らかになってきている．

今回は，スライドウインドウ機構を有効に利用することによって，必要とされるレジスタ数の上限を従来のアーキテクチャよりも小さく抑えることができるということがわかったので報告する．

2. 従来アーキテクチャにおけるレジスタ割付

2.1 Cyclic Interval Graph

Cyclic Interval Graph（CIG）は，Fat Cover Algorithm とともにループプログラムのレジスタ割付手法として提案された[4]．CIG は，横軸にループ内の命令ステップ番号，縦軸にレジスタ番号をとった枠を用いて記述する（図1）．CIG 上では，変数のライブレンジは線分で表現される．最後の点そのものは含まない．

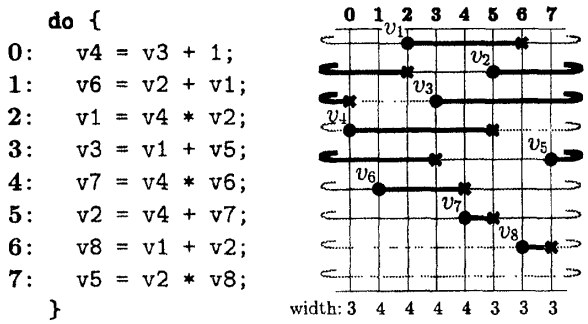

図1　ループ(左) と Cyclic Interval Graph による表現(右)

CIG に関する定義と定理をあげる．

定義1　CIG G において，ある時刻 t に生存しているライブレンジの数を，t における G の幅といい，$width(G,t)$ と書く．また，G の最大の幅，最小の幅をそれぞれ $W_{max}(G)$，$W_{min}(G)$ と書く．　□

定理1　CIG G は $W_{max}(G)$ 以上 $W_{max}(G)+W_{min}(G)$ 色以下で彩色可能である．　□

3. スライドウインドウにおけるレジスタ割付

定理1は，任意のライブレンジ集合が与えられたときの必要レジスタ数 R が $W_{max}(G) \leq R \leq W_{max}(G)+W_{min}(G)$ の範囲にあることを保証している．スライドウインドウアーキテクチャにおいては，従来のアーキテクチャと異なり，R は $W_{max}(G) \leq R \leq W_{max}(G)+1$ の範囲で済む．Spiral Graph を用いてこのことの証明を行なった．

3.1 Spiral Graph

Spiral Graph（SG）はスライドレジスタを自然に表現するためのグラフとして提案された[3]．SG は，CIG のスライドウインドウアーキテクチャへの素直な拡張である．SG は，CIG では水平であった横軸を傾けて取っているのが特徴であり（図2-左），この斜線はループの繰り返しを経るたびにレジスタが別名になることを意味している．本稿では，ループの末尾（時刻 $II-1$ と時刻 0 の間）に slide +1 命令を置き，繰り返しごとにレジスタ番号がひとつずつずれるように設定している（CP-PACS が採用しているスライドウインドウアーキテクチャは自由度が高く，slide +k 命令でレジス

An Advantage of Slide-Window Architecture in Register Allocation
Tomohiro HARAIKAWA, Motohide Soeno, Yoshiyuki YAMASHITA, Ikuo NAKATA
University of Tsukuba

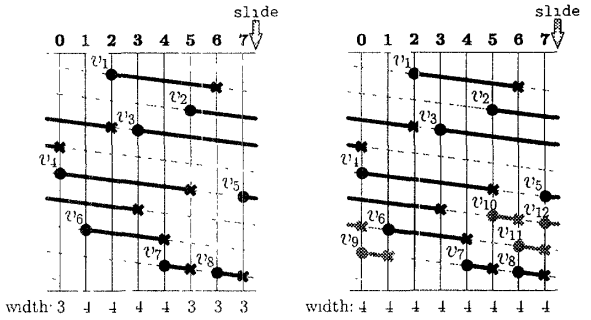

図2　Spiral Graph(左) と その閉包(右)

タ番号を k ずらすことができ，また，**slide** 命令をプログラム中に好きなだけ配置することができる．**slide** が増えればグラフの傾きは急になる）．

図2-左 は，図1 と同じライブレンジを SG 上に表したものである．ライブレンジを，SG では $v_2 = [5, 10)$ などと表現する．やはり最後の点そのものは含まない．

定義2　ライブレンジ $v = [s, e)$ において，s，e をそれぞれ v の**始点**，**終点**という．$0 \leq s \leq II - 1$ である．また，$s \bmod II$, $e \bmod II$ をそれぞれ v の**開始時刻**，**終了時刻**といい，$\mathrm{ST}(v)$，$\mathrm{ET}(v)$ と書く．□

スライドウインドウアーキテクチャにおいては，定理1 に対応する定理は次のようになる．

定理2　SG G は W_{max} 以上 $W_{max}+1$ 色で彩色可能である．□

この定理の証明のために，いくつかの定義を与え，また補題を証明した．

定義3　2 つのライブレンジ v_i，v_j において，$\mathrm{ET}(v_i) = \mathrm{ST}(v_j)$ であるとき，v_i と v_j は（この順に）**連結可能**であるという．v_i と v_j を連結したものをライブレンジの列とよび，(v_i, v_j) と書く．列 $L = (v_{i_1}, v_{i_2}, \ldots, v_{i_n})$ において，$\mathrm{ST}(L)$, $\mathrm{ET}(L)$ を $\mathrm{ST}(L) = \mathrm{ST}(v_{i_1})$, $\mathrm{ET}(L) = \mathrm{ET}(v_{i_n})$ で定める．□

定義4　SG G と，それに含まれる変数のライブレンジの集合 $V = \{v_1, v_2, \ldots, v_N\}$ を考える．G にライブレンジ $[t, t+1)$ を適当に加え，$width(\overline{G}, t) = W_{max}(G)$，$0 \leq t \leq II - 1$ となるようなグラフ $\overline{G}$ を構成する（図2-右）．$\overline{G}$ を G の**閉包**とよび，$\overline{G}$ を構成するために加えたライブレンジを slack とよぶ．□

以下，断りのない限り，slack もはじめからあった他のライブレンジと特に区別せずに扱う．

補題3　SG G の閉包 $\overline{G}$ において，あるライブレンジ v が時刻 t に終点を持つとする．このとき，同時刻に始点を持つライブレンジ v' が必ず存在する（自分自身でもよい）．また，任意の時刻 t において，終点と始点とは同じ数だけ存在する．□

定義5　ライブレンジの列 L が，$\mathrm{ST}(L) = \mathrm{ET}(L)$ をみたすとき，L を **slide cover** という．□

補題4　SG G の閉包 $\overline{G}$ に含まれるライブレンジを適当に組み合わせて，slide cover だけからなるグラフを作ることができる．□

定義6　Slide cover S_1, S_2 において，S_1 の終点と同じレジスタに S_2 の始点を配置できないとき，S_1 と S_2 は（この順に）**干渉する**という．□

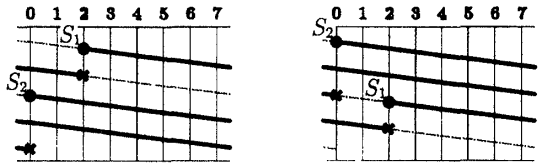

図3　干渉する slide cover (左) と 干渉しない slide cover (右)

補題5　SG G に含まれるライブレンジをいくつかの slide cover に余すところなく作りかえ得るなら，G は $W_{max}(G) + 1$ 色で彩色可能である．□

4. むすび

定理2 の証明は，補題4，5 の証明ができれば明らかであり，補題の証明も完了している．

今回は，基本ループの末尾に **slide +1** 命令を配置した場合について述べてきたが，その場合においては，ライブレンジを最小数のレジスタに割り付ける多項式時間アルゴリズムも考案されている（従来のアーキテクチャでは，この割付問題は一般には $\mathcal{NP}$ 困難な問題である）．今後は，条件分岐はもとより，**slide +k** 命令に関する考察をすすめて，一般のレジスタウインドウ方式のプロセッサに応用し得るアルゴリズムを考えたい．

参考文献

1) H. Nakamura, H. Imori, K. Nakazawa, T. Boku, I. Nakata, Y. Yamashita, H. Wada & Y. Inagami, "A Scaler Architecture for Pseudo Vector Processing based on Slide Windowed Registers", *ACM Proc. Intl. Conf. on Supercomputing* '93, 298–397, 1993.

2) 中田育男，山下義行，小柳義夫，"超並列計算機 **CP-PACS** のソフトウェア"，情報処理 37-1, 29–37, 1996.

3) 秡川友宏，添野元秀，山下義行，中田育男，"スライドウインドウを考慮したレジスタ割付"，日本ソフトウェア科学会 第13回大会論文集, 201-204, 1996.

4) L. J. Hendren, G. R. Gao, E. R. Altman and C. Mukerji, "Register Allocation Using Cyclic Interval Graphs", *Technical Report ACAPS-MEMO* **33**, McGill University, 1991.