4U-4 パイプラインマージソータに於ける String Length Tuning用フラグ自動生成機構

楊 維康　　喜連川 優　　高木 幹雄

東京大学生産技術研究所

1．はじめに

我々はパイプラインマージソータのアルゴリズム及びその実装法についての研究を行い，既に18段のソートプロセッサから構成される大容量ハードウェアソータを試作し[1]，その試作システムでソータの動作確認と性能評価を行った．又，そのソートプロセッサのLSI 化を行い，1チップのソートプロセッサを実現した[2]．そのLSI ソートチップを用いて，大容量（最大64MB），高速（最大8MB／Sec.）のハードウェアソータをコンパクトに実現することができる．本ソートチップの最大の特長は我々が開発したSLT 機構を始め，幾つかの拡張機能を有しており，それらの機能により，本チップで構成されたソータは，ソートするファイルのレコード長，レコード数等のパラメータに対して，非常に高い柔軟性を有している．それらの拡張機能を利用する為には，ソータに入力されるデータストリームの各レコードの先頭に制御フラグを付加し，必要な制御情報をソートプロセッサに知らせる必要がある．又，本ソータの動作はデータストリーム指向で，ディスク等からのデータを上位プロセッサの主記憶等に転送する時間を利用して，On-the-flyにソート処理を行うことが可能である．そのような処理ができるようにする為には，フラグの生成をOn-the-flyに行う自動生成機構が不可欠である．

本稿では，そのような制御フラグをハードウェアで自動的に生成するアルゴリズムを紹介し，その実装法について述べる．

2．SLT 及び制御フラグの種類

SLT はソートするファイルのレコード長がソータの設計レコード長と異なる時，ソータの最初の数段のソートプロセッサに於いてダイナミックに出力するストリング（ソートされた部分列）の長さを調整して，ある中間プロセッサのメモリを最大限に利用できるようなストリングを生成し，それによって，ソータ全体のメモリ利用効率をほぼデータのレコード長に影響されないように最適化するアルゴリズムである．そのアルゴリズムによって，レコード長に対する柔軟性を実現した．ストリング長の調整は図1のマージ木で示すアルゴリズムに従って行い，その実現は，初段のソートプロセッサに於いてレコードを適宜マージせずにバイパスすることによって実現されている（図1の終端で兄弟のないノード）．図の中では，dはチューニングの次数と言い，N_d はP_d が出力するストリングのレコード数である．

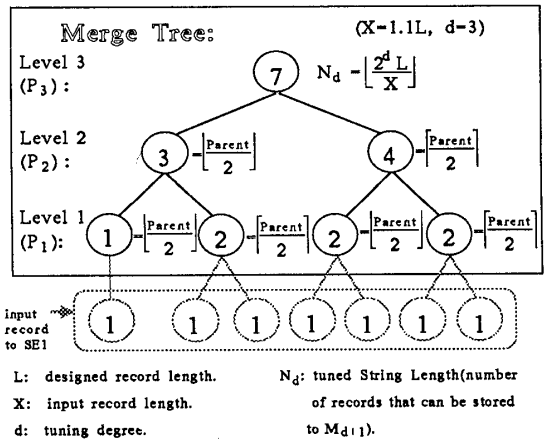


図1．マージ木の例

本ソータに於いて，その他の拡張機能の実現に合わせて，次のようなフラグを導入している．

BOS ——各ストリングの先頭レコードを示す．

LS ——データストリーム中，最終ストリングの先頭レコードを示す．

EOS ——データストリームの終了を示す．

NMP ——SLT 動作時，初段のプロセッサに於いてマージペアのないレコードを示す．

SBP ——このフラグ以降LSフラグまでのデータをバイパスすることを指示する．

DR ——SLT の動作中に内部的に生じたダミーレコードを示す．

SLT を行わない場合，ソータに入力するデータの各レコードにBOS フラグを付加し，ソートプロセッサはBOS フラグのレコードをマージする．SLT 動作時，初段でバイパスするレコードにはNMP フラグを付加する．BOS とNMP のシーケンスによって，SLTが実現されている．

3．SLT 用フラグの自動生成アルゴリズム

次数dのSLT を行う場合，N_d レコードからなるストリングを生成するマージ木は，深さd+1，N_d 個の葉ノー

Flag Generater for String Length Tuning in Pipeline Merge Sorter

W. Ynag, M. Kitsuregawa, M. Takagi
Institute of Industial Sience,
University of Tokyo

ドを持つバランスした２進木である．一般的に，$2^{d-1} < N_d < 2^d$ であり，言い換えれば，そのマージ木は深さd+1 の完全２進木の2^d 個の葉ノードの内，$2^d - N_d$ 個をバランスして切り取ったものである．但し，そのマージ木の左部分木の葉ノード数は右部分木のそれと等しいか，又は１個少ないことがある．この条件はマージ木に対して再帰的に適用できる．その切り取られた葉ノードは次のように決定することができる．

深さｄ＋１の完全２進木の2^d 個の葉ノードを左から$(2^{d-1}-1),(2^{d-1}-2),\ldots,0$と番号をつける．番号ｉを２進数で表し，そのｉの２進表現のMSB をLSB として得られた値をｉ’とする．ｉ’$\geqq N_d$ の葉ノードの右兄弟を切り取る．図２はその例を示す．

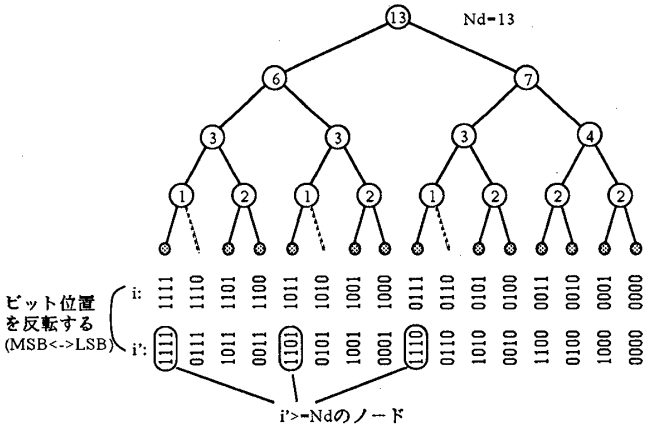


図２．SLT 用フラグ生成アルゴリズムの例

ｉ’$\geqq N_d$ の葉ノードは初段ではマージペアがなく，バイパスされるレコードに相当する．故に，そのノードにはBOS フラグの代わりにNMP フラグが付加される．

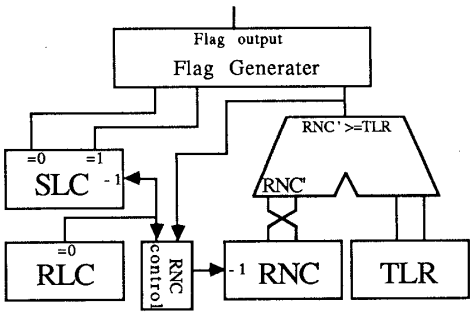
以上をまとめると，BOS とNMP フラグのシーケンスは次のようなアルゴリズムで簡単に生成することができる．

BOS,NMP フラグを生成するアルゴリズム:

```
RNC:入力レコード数をカウントするカウンター
RNC': RNC のMSB とLSB を反転した値
BEGIN
  DO
    RNC を'11..1' にリセットする;
LAB1: IF (RNC' < Nd ) THEN
        BOS フラグを出力する;
      ELSE      /*  RNC' ≧ Nd の場合  */
        NMP フラグを出力し，RNC を１回カウント
        ダウンする;
      RNC を１回カウントダウンする.
      IF (RNC ≧ 0)  GOTO LAB1;
  UNTIL(入力終了)
END
```

４．フラグ自動生成機構のハードウェア構成

図３にフラグ自動生成機構のハードウェア構成を示す．



RLC: Record Length Counter
RNC: Record Number Counter
SLC: Stream Length Counter
TLR: Tuning Length Register

The Function of Flag Generater:

SLC=0	SLC=1	RNC'>=TLR	Flag
0	0	1	NMP
0	0	0	BOS
0	1	X	LS
1	0	X	EOS

図３．フラグ自動生成機構のハードウェア構成

この図では，RNCはフラグ生成用のレコード数カウンタで，SLCはストリームの残りレコード数，RLCは処理中のレコードの残りバイト数を示すカウンタである．これらは皆ダウンカウンタである．RLCの初期値はレコード長で，１バイト入力する度にカウントダウンする．RLC=0の時，データストリームにフラグを付加する処理を行い，RLCをレコード長にリセットする．TLRはN_d の値を保持するレジスタで，RNCのMSB とLSB を反転した値RNC’と比較される．その比較結果によってSLT 用のフラグを生成する．

図３でわかるように以上のアルゴリズムを実現するハードウェアは非常に簡単で，データストリーム入力と同時に，フラグを生成することができる．簡単にする為に，各レジスタをセット，リセットする部分や，SBP フラグを生成する部分はこの図に示していない．

５．むすび

本稿では，SLT を始め，幾つかの拡張機能を有するパイプラインマージソータ用の制御フラグを自動的に生成するハードウェアのアルゴリズム及びその構成について述べた．本ハードウェアを用いて，ソータ用のフラグをOn-the-flyに生成でき，ディスク等からのデータ流に沿ってソート処理を行うことを可能にした．このような回路をソータ駆動系に組み込み，実用可能なシステムを実現するのが今後の課題である．

＜参考文献＞

[1] 楊，鈴木，喜連川『高速(4MB/Sec.)大容量(8MB) ハードウェアソータの実装』信学技報 CPSY 86-26,1986
[2] 楊，喜連川『LSI ソートチップの試作』 第37回情報処理学会全国大会 7Q-4