

LSIパイプラインソータの

2K-3

論理シミュレーション

林 隆史[†] 喜連川 優^{††} 伏見 信也[†] 田中 英彦[†] 元岡 達[†]

([†] 東京大学 工学部 ^{††} 東京大学 生産技術研究所)

1. はじめに

現在までに、多くのハードウェアソータが提案され、或いは試作されているが、本LSIパイプラインソータはパイプラインマージソータを行ない、Nケのレコードをlog N台のソータエレメントによりO(N)時間でソータすることが可能である。

前回までに、本ソータのLSI化の手法等について報告してきたが、今回はLSI化における問題点と論理シミュレーションについて報告する。

2. ソータの構成 [1]

本パイプラインマージソータは、Nケのレコードのソータを行なうものとする、一次元状に結合されたlog N台のプロセッサと、第i番目のプロセッサに対応する2ⁱ⁻¹レコード分のメモリにより構成される(図1)。

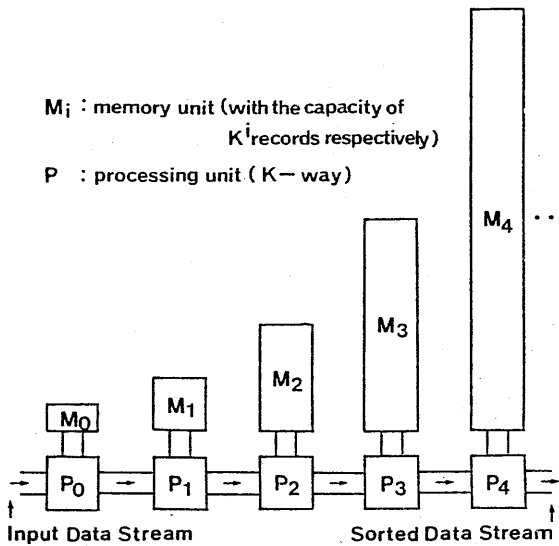


図1. パイプラインマージソータの構成

3. LSIソータのソータ能力評価

3.1. 並列バブルソータ [2, 3] との比較

ソータをLSI化する場合、LSI1チップのソータ能力が重要な問題となる。ここで大きさの単位としてトランジスタ数を用いて、チップの大きさをS、本ソータに於けるプロセッサの大きさをP、メモリ1bitの大きさをM、レコード長をR (bit)、

ソータ能力をNmレコードとすると、

$$S = \log N_m \cdot P + (N_m - 1) M \cdot R$$

からNmが求められる。

これに対し並列バブルソータでは、セルの大きさをC、ソータ能力をNbレコードとすると、

$$S = N_b \cdot R \cdot C / 2$$

となる。

ここで、S=5×10⁵、R=640(80バイト)、P=4×10⁴、M=6、C=30、として計算すると、Nb=52、log Nm=6より、Nm=64となる。更に、S=5×10⁶とすると、Nb=520、Nm=1024となり、VLSI技術の集積度が向上するほど本ソータが有利となる。

3.2. 本ソータに於ける改良

本ソータに於けるレコードを格納するメモリとして一般のRAMを用いることができるため、プロセッサ部のみをLSIとすることができる。この場合先程の計算では、log Nm=12となり、Nm=4096となる。

また、本ソータではメモリ管理をポインタを用いて行っており、ほぼ100%のメモリ効率を実現している。このためメモリ管理用ハードウェア及びマイクロ命令内に占めるメモリ管理部の比率が高くなっている。そこで、メモリ管理方式をポインタ方式からDouble Memory方式にし、より制御を容易にすることが考えられる。このDouble Memory方式は2本の入力ストリングに別々のメモリ空間を用意する方式で、メモリ効率は50%となる(図2)。

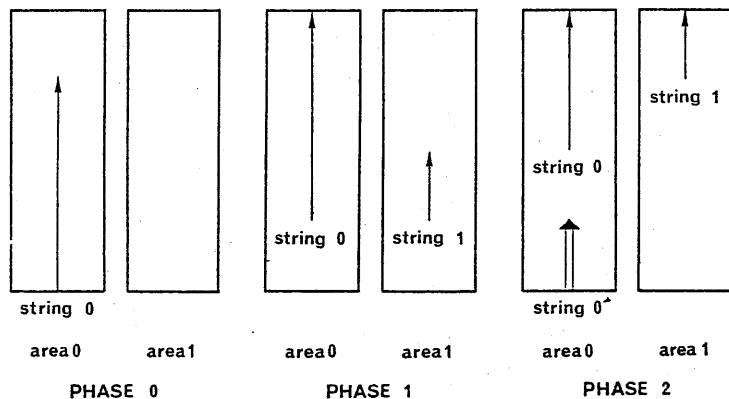


図2. Double Memory Method

この方式を用いたソータをレジスタトランスファレベルで記述し、Pascal を用いてシミュレーションを行ない検証した。その結果、マイクロ命令のステップ数が約60%に、マイクロの幅も約60%に縮小されるためマイクロ命令ROMは約40%に縮小される。他のハードウェアリソースも約10%減少し、プロセッサ部全体の大きさは60~70%に縮小される。ここで、 $P=2.8 \times 10^4$ として計算すると、 $\log Nm = 17$ より、 $Nm = 1.3 \times 10^5$ となる。

また、本ソータでは内部RAMに比較レコードの1本を常に格納しているために、2clock/cycleで処理を行なうことが可能となっている。ここで、3clock/cycleで処理を行なえば内部RAMは省略可能となる。このとき、状態遷移のstate数が大幅に減少するため、マイクロ命令のステップ数は約70%に、プロセッサ全体でも約70%に縮小される。ここで、Double Memory方式を併用して計算を行なうと、 $P=2 \times 10^4$ 程度となり、 $\log Nm = 25$ となる。しかし、実際には、ピン数の制限によりプロセッサ台数は15程度が限界である。

4. ソータの論理シミュレーション

現在ソータのLSI化はハードウェア記述を終了し、論理シミュレータを用いてデバッグ及び検証を行なっている。初期化時の動作状態に於けるTEGAS 5Fシミュレータの出力を図3に示す。これは比較器への入力レジスタに対する結果である。ここで、aはEnable信号入力、bがクロック入力であり、cがa及びbにより生成されレジスタ内のフリップフロップの入力クロックとなる。また、dは1bit分のフリップフロップの出力、eがdの反転出力を示している。このような論理シミュレーションでは分割されたモジュール毎に独立してシミュレーションを行なえることが非常に重要である。このためLSIの階層設計段階に於いて各モジュールが互いに行なうことができるだけ独立となるように設計する必要がある。本ソータもLSI化のための階層設計段階に於いて、5ケの最上位モジュールに分割されている[4]。このため本ソータは次のような方法でシミュレーションを行なうことができる。まず最初に、CLKGEN及びMICROのシミュレーションによるデバッグ及び検証を行なう。次に、他のCOMPARE等のモジュールのシミュレーションをCLKGEN、MICROと結合させた形で行なう。最後に、全体でシミュレーションを行ない、検証する。

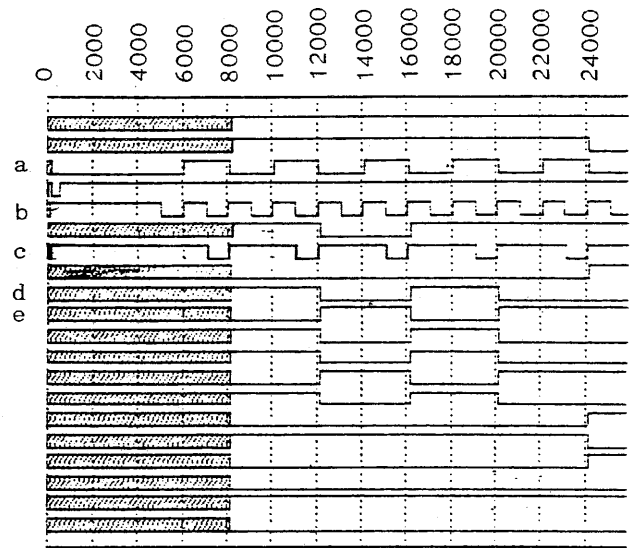


図3. シミュレータ出力

5. おわりに

現在、より詳細な論理シミュレーションを行なっており、今後、論理シミュレーションを終了したのちテストパターンの作成等を行なっていく予定である。

《 参考文献 》

- [1] 喜連川、伏見、桑原、田中、元岡
「パイプラインマージソータの構成」
信学技報 EC 82-32 1982
- [2] M. Carey, et. al.,
RESST: A VLSI Implementation of
Record-Sorting Stack,
UCB/CSD 82/102, 1982
- [3] G. Miranker, et. al.,
A "Zero-Time" VLSI Sorter,
IBM J. Res. Develop.
Vol.27, No.2, 1983
- [4] 林、喜連川、伏見、田中、元岡
「パイプラインマージソータの構成
= LSI化に関する一検討 =」
情報処理学会第25回全国大会 4P -4, 1982