

1P-2

画像処理用データフロープロセッサ
ImPPにおける流量制御方法の検討

内田 薫 天満 勉 岩下正雄
日本電気株式会社 C & C 情報研究所

1. はじめに 可変長パイプライン方式を実現するためにデータ駆動アーキテクチャを採用した画像処理プロセッサ ImPP (Image Pipelined Processor: μ PD7281) においては、データを多数生成することにより並列性の高い処理を行うことが可能であるが、同時に処理可能なデータ量にはハードウェアの容量からくる制限があるため、容量以上のトークン生成を行うプログラムでは実行時にプロセッサ内のキューオーバーフローが発生し、処理の続行が不可能となる。

特に ImPP 用の高級言語ではオーバーフローが発生しないように流量がソフトウェア的に制御されたオブジェクトを生成するコンパイラの実現が不可欠であり、そのための流量制御法を検討したので報告する。

2. ImPP とその流量制御 図1に ImPP のブロック図を示す。データの単位である「トークン」はデータ値と処理内容を規定する識別子とからなり、これがプロセッサ内部のリングを必要な回数だけ循環し、PUにおいて処理を受ける。プロセッサ内のトークン数はプロセッサ外とのトークンの入出力、トークン生成命令(GE命令)、消滅命令の実行により増減する。2項演算の場合など2つのトークンの同期を取るた

めにはDM上にとられたキュー(DMQ)でトークンの待ち合せを行うことができる。またPUの前にはGQ(GE命令トークン用: サイズ16)及びDQ(サイズ32)と呼ぶ流れの緩衝用のキューがあり、トークンはここからPUに送られて処理を受ける。DQ内に滞留するトークン数が8個以上の時はDQからのトークンが優先的にPUに送られる。

以上述べたようなDMQ、GQ、DQのキューはその容量を越えた数のトークンの流入によりオーバーフローを起こすので、高級言語化のためにはこのようなオーバーフローが発生するプログラムを安全なものに変形する作業をコンパイラ上で自動的に行うことが必要である。これを実現するための流量のソフトウェア制御としては、処理のプログラムに対応するフローグラフが与えられたときにここから実行時のオーバーフロー発生危険度の指標を静的に抽出し、これが制御目標値内になるようにデータの生成、流れ方の制御用のメカニズムをプログラムに付加するという手法を用いる。

3. 危険度の指標と制御目標値 処理を記述するフローから得られる危険度の指標とその制御目標値は次の通りである。

a) Maxflow (Mf): フローグラフ上の各アークについてある瞬間にそこに存在し得るトークン数の上限を表す。全グラフ上においてMfを(キューサイズの)16以下にすればDMQオーバーフローが防げ、その後の制御のために任意の場所に制御用キュー命令を付加することが可能になる。一般にMfは上のノードの入力アークのMfとノードのファンクションとから求まる。

b) GQトークン並列度(GQ-p): フローグ

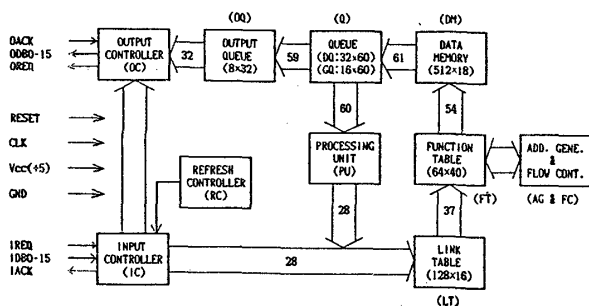


図1 ImPPのブロック図

ラフ上にアークを要素とするカットセットをとったときのGE命令ノードに接続する要素アークのMfの和の最大値。これはある瞬間にGQに滞留するトークン数の上限を表すのでGEオーバーフローを防止するためGQのサイズである16以下にする。GQ-pはフローグラフ上のカットセットを有限回動かすことにより評価できる。

c) DQトークン並列度(DQ-p): GQ-pと同様にDQ滞留トークン数の上限を示し、DQのサイズの32以下にする。

図2にフローグラフ及び上記指標の評価の例を示す。(点線は指標評価のためのカットセットを、アークにつけられた数字はMfを表す。)

4. 制御メカニズムの付加 制御は(1)処理結果に影響を与えない(2)指標を目安に完全にオーバーフローを防止する(3)なるべく実行時の効率を良くする、という原則のもとでプロセスの起動タイミングの調整により処理の並列度を減少させることを基本とする。

実際の制御手法としては、(a)後続する処理プロセスの起動トークンと数プロセス前の終了トークン(フィードバックトークン)との待ち合わせを行うことによって後のプロセスの起動を遅らせ、並列に処理される

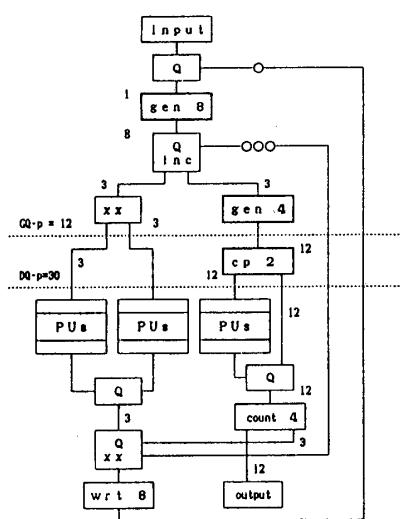


図2 フローグラフと指標評価の例

図3 (→) DQ-pの制御の例

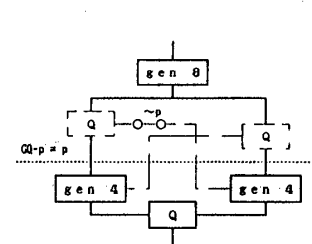
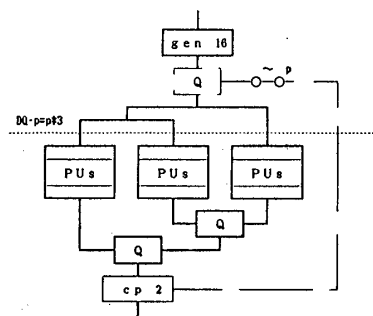


図4 GQ-pの制御の例

プロセス数を制限することによりプロセス内の流量を一定化させる (b)DQ内のトークン数が多い場合にGE命令の実行がハードウェア的に制限されることを利用し、制御用のGE命令を挿入する (c)GQ-pの値を小さくするためにGE命令を縦列化するなどの手法を用いる。

(a)の手法によってDQ-pを制御をした例を図3に、(c)の手法によってGQ-pを制御した例を図4に示す。

5. 実験結果 一般に制御を付加するとそれにより処理量が増大するが、このオーバーヘッドは加えられたアーク上のトークン流量から正確に算出することができる。実験を行った範囲ではこれは全体の処理量に対して5%程度で性能的に大きな問題はなく、この制御法が有効であることが確認できた。例えば図2の例においては全体の処理量753に対して加えた制御オーバーヘッドは17となる。

6. おわりに ImPPのオーバーフローを防止するためのソフトウェアによる流量制御法を定式化した。今後はさらに効率的な制御方法を考察すると共に高級言語仕様及びそのコンパイラへの本成果の適用を行いたい。

謝辞 日頃御指導いただくパタン認識研究部浅井部長に感謝致します。

参考文献

- 1) Temma, Iwashita et al: Data Flow Processor Chip for Image Processing, IEEE Trans. on ED., ED-32, No.9, pp. 1784-1791, Sept. 1985
- 2) Iwashita, Temma: Data Flow Chip ImPP and its System for Image Processing, IEEE ICASSP '86, 15-3, pp. 785-788
- 3) 内田、天満: データ駆動型画像処理プロセッサImPPにおけるデータ流量制御方法の検討, '85信学全大, 175, ('85.11)