情報処理学会第39回(平成元年後期)全国大会

6M-5

高速汎用画像処理システムＨＩＧＩＰＳのモニタ機能について
～ＳＵＢＣとＳＭＰＵのモニタ機能～

姚 風会　玉木 明和　加藤 清史
(九州工業大学)

1．はじめに

現在，我々の研究室では高速性と汎用性を兼備した画像処理システムHIGIPSを開発している．それのハードウェア構成と有用性は既に発表した[1]-[3] 本稿ではHIGIPS上で高速，柔軟な画像処理を行うための制御プログラム――モニタについて述べる．

2．ＨＩＧＩＰＳのハードウェア構成

HIGIPSのアーキテクチャーは全体的にパイプライン形式，また，パイプラインの各ステージはマルチプロセッサー形式を取っている．それの構成は図１に示すように画像入力装置IM (Input Module)，M個のプロセッサモジュールPM (Processor Module)，(M+1)個のメモリモジュールMM(Memory Module)，出力装置OM(Output Module)とシステムコントローラSC(System Controller，パーソナルコンピュータPC98を使用) から構成される．また，PM内部は一つのサブコントローラSUBC(Sub-controller)とN個のスレーブプロセッサSMPU(Slave MPU,現在のプロトタイプでは，M=3,N=7)及びCROM(Common ROM)などから構成される．図２にHIGIPSの１ステージのメモリマップを示す．上位512KBはグローバルメモリGM(Global Memory)で，下位512KBはローカルメモリLM (Local Memory)である．また，GMは最上位の16KBのCROM(IPL用)領域，128Bのメールボックス MB(Mail Box)と画像データ用領域に分けている．LMはモニタ領域，スケジューラ領域（SMPUの場合はスケジューラとユーザプログラム）に分けている．

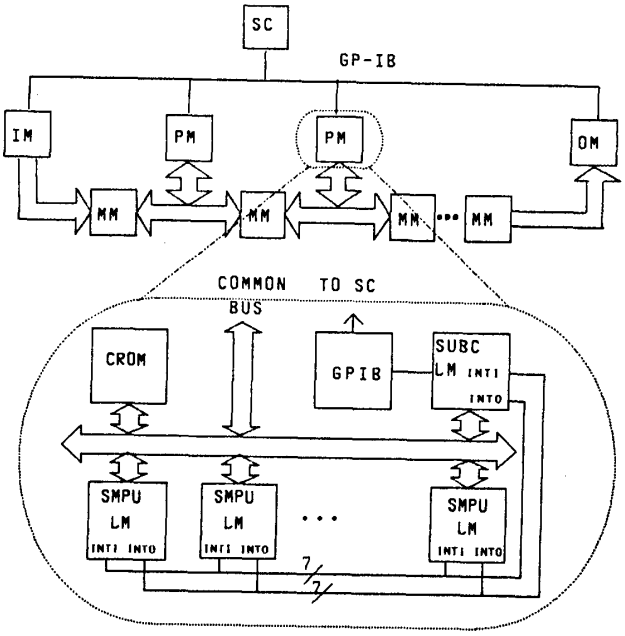
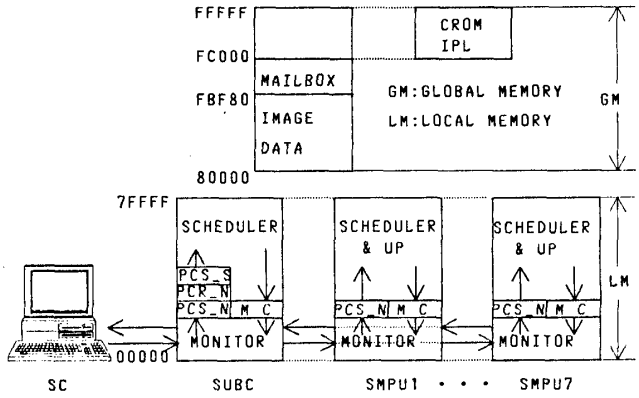


図1　ＨＩＧＩＰＳのハードウェア構成図



UP:USER PROGRAM　MC:MONITOR CALL　PCS_N:PROCESS_NO.
PCS_S:PROCESS_STATUS　PCR_N:PROCESSOR_NO.

図2　ＨＩＧＩＰＳに１ステージのメモリマップ

3．ＨＩＧＩＰＳのモニタの機能

HIGIPSは通常に動作する時，SCと各ステージのSUBC間のデータ（あるいはプログラム）転送，メッセージ交換が必要であり，また，ステージ内ではSUBCと各SMPU間のメッセージ交換，ユーザプログラムとモニタ間のやりとりなども必要である．

ここでは主にSUBCと各SMPU間に於てモニタの必要な機能について取り上げる．

3．1　画像処理の前準備

ある画像処理を行う時，図３(a)に示すような画像処理フローグラフL1が得られる．この処理グラフの各ノードP_iは一つの処理単位で，プロセスと呼ぶ．それらの処理プログラムはSC上で高級言語(例えばC言語など)によって開発される．図３(a)の処理フローグラフは図４(b)のような処理グラフL2に分割できる．分割基準を次に示す．

基準１：順序関係$(x,y)\in L1 \rightarrow (x,y)\in L2$

基準２：$T=\max\{T_0,T_1,T_2\}$を最小にする．

(x,y)は，プロセスxがプロセスyより先に実行することを表す．T_0はステージ0の処理時間，すなわち，フローグラフ図３(b)に於てq_{0S}からq_{0E}までの時間である．T_1,T_2も同様にステージ1,2の処理時間である．

処理プログラム$p_1,\ldots,p_9,p_{10},\ldots,p_{16},p_{17},\ldots,p_{23}$をそれぞれHIGIPSのステージ0，1，2のGMにダウンロードする．

Monitor of High-speed General-purpose Image Processing System
～Facilities of the Monitor between Sub-controller and Slave MPU～
Fenghui, YAO,　Akikazu TAMAKI,　Kiyoshi KATO
Kyushu Institute of Technology

各ステージのSMPUはSUBCの制御の下で，GMにダウンロードされた処理プログラムを各自のLMにコピーする．同一ステージの各SMPUは同様な処理プログラムを持っている．

各ステージのSUBCはステージ内の各SMPUを実行可能な状態にセットして置く．SCからのコマンドを待つ．

3．2　画像処理プロセスの制御

(1)　サブコントローラSUBC側

ステージ0のSUBCはステージ内の各プロセッサの初期設定が終わったら，プロセスQ_{0S}に入る．SCから実行コマンドが来たら，P_1を$SMPU_1$に，...，P_5を$SMPU_5$に割り当て，フローグラフL2に従ってスケジューリングを始める．ステージ1,2のSUBCも同様な動作をする．プロセスQ_{0S}に入ったら，SCに実行終了メッセージを送り，ステージで処理した画像データの移動（ステージ0からステージ1に，ステージ1からステージ2に）を要求する．

各ステージのSUBCは次の手順でステージ内の各SMPUを管理し，ステージに与えられたプロセスをスケジューリングする．

①　図2に示すようにSUBC側のユーザプログラム（スケジューラ）の中に変数領域PCS_S(process_status),PCR_N（processor_no.）とPCS_N（process_no.）を設けている．通常，ユーザプログラムのスケジューラはこの領域を調べつつある．

②　プロセスを終えて待状態となったSMPUは割り込みをかけて来ると，このスケジューラから抜け出して，モニタを起動する．モニタは受信メールボックスから前に処理したプロセス番号，例えば，iとその処理状態を得る．また，モニタは何番のSMPUが割り込みをかけて来たかが分かっている．

③　モニタはこれらをそれぞれ変数領域PCS_N，PCS_SとPCR_Nに書き込んで，スケジューラに戻る．

④　ユーザプログラムのスケジューラはこれらを受け取ると，プロセスjに対して，プロセスjに先行するプロセスがすべて完了しいているかどうかをチェックする．全部完了していれば，jプロセスをスレーブプロセッサkに割り当てるために，モニタコールをする．これによって，プロセスjとプロセッサ番号kをモニタに渡す．

⑤ モニタはユーザプログラムからプロセス番号jとプロセッサ番号kを送信メールボックスに書き込んで，k番目のSMPUに割り込みをかける．k番目のSMPUのモニタにメッセージがあると知らせる．モニタコールを終了し，ユーザプログラムのスケジューラに戻る．

(2)　スレーブプロセッサSMPU側

各SMPUは次の手順でSUBCからのプロセス分配を受取，または，メッセージ交換を行う．

①　図2に示すようにSMPUのユーザプログラムの中に変数領域PCS_N(process_no.)を設けている．ユーザプログラムのスケジューラは通常この領域を調べつつある．

②　SUBCから割り込みがかかって来たら，このスケジューラを中断する．モニタはSMPUに対応している送信ボックスからプロセス番号を受け取る．受け取ったプロセス番号は変数領域PCS_Nに書き込んで，スケジューラに戻る．

③　ユーザプログラムのスケジューラはこの変数領域

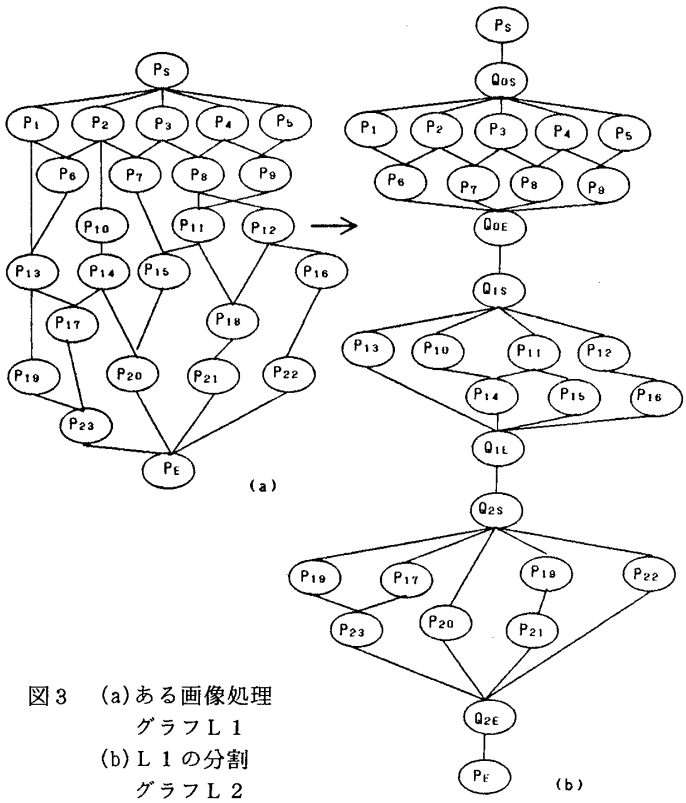


図3　(a)ある画像処理グラフL1
(b)L1の分割グラフL2

からPCS_Nを受け取ると，j番目のプロセス（関数の形）を呼び出して，画像処理単位P_iを行う．j番目のプロセスはうまく完了したら，プロセス状態PCS_Sを0にし，他の場合は1にする．

④　この処理したプロセスの状態と処理を行ったプロセッサ番号をモニタに渡すために，スケジューラはモニタコールする．

⑤　モニタは受け取ったPCS_SとPCS_NをSMPUに対応している受信メールボックスのk番目に書き込んで，SUBCに割り込みをかけて，プロセスの完了終了あるいは異常終了を知らせる．モニタコールを終了し，ユーザプログラムのスケジューラに戻る．

4．おわりに

本稿ではHIGIPSのSUBCと各SMPU間のモニタの機能について述べた．しかし，SCと各SUBC間のモニタにどういう機能を持たせるかをも検討しなければならない．また，画像処理フローグラフを与えると，本システムに適した分割をするための支援ツールをも開発しなければならない．

[参考文献]

[1]　姚風会　他:高速汎用画像処理システムHIGIPSのアーキテクチャについての検討,情報処理学会第37回（昭和63年後期）全国大会

[2]　姚風会　他:高速汎用画像処理システムHIGIPSのソフトウェア開発,昭和63年度電気関係学会九州支部連合大会講演論文集

[3]　姚風会　他：マイクロプロセッサを用いた高速汎用画像処理システムの開発,昭和62年度電気関係学会九州支部連合大会講演論文集