メモリ型プロセッサによる動画像処理システム RVS-2 —基本ソフトウェア—

5R－5

許 昭倫　岡崎信一郎　佐藤 完†　古賀拓也
NEC 情報メディア研究所　†(株)NEC 情報システムズ

1 はじめに

低～中レベル画像処理に有効な1次元SIMD型プロセッサアレイのアーキテクチャを有するRVS-2[1]は、その高速性とコンパクト性に大きな特徴を持つ。本稿はRVS-2の基本ソフトウェア、特にそのシステム記述用高級言語1DC (One Dimensional C) について述べる。

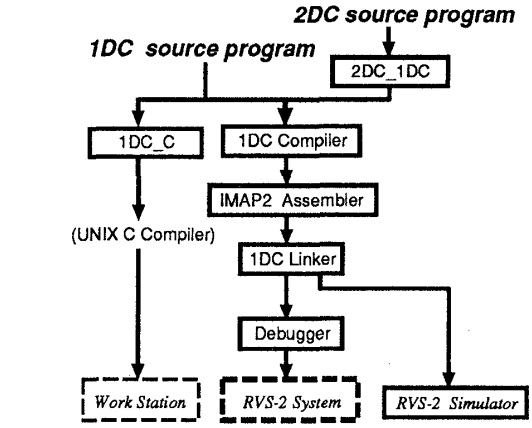
2 RVS-2基本ソフトウェアの構成

RVS-2の基本ソフトウェアは、アセンブラ、デバッガ、シミュレータ、1DCコンパイラとリンカ、そして1DC_Cと2DC_1DCの両トランスレータからなる。全ツールはワークステーション上で動作し、デバッガやシミュレータはXウインドウベースのものとなっている (図3)。高級言語はシステム記述用に1DC、より高レベルな記述用に2DC (Two Dimensional C) の2種類のCライクな言語を有し、1DCはRVS-2の1次元SIMD的な動作を明快に記述できる高級言語として、また2DCは点集計型演算[2]のもとで記述の簡潔性と簡易性とを追求した汎用の画像処理言語として設計した。なお以下では1DCについてのみ説明し、2DCは[2]を参照されたい。

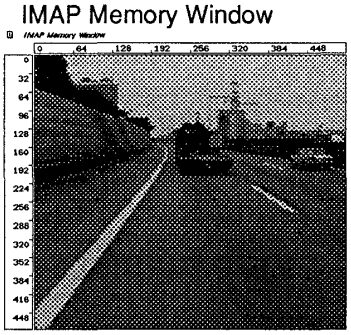
3 IMAPシステム記述用高級言語1DC

1DCではRVS-2のアーキテクチャを考慮し、データ型とは別に宣言時の修飾子の有無により変数に主にcommon、separate、singleの3属性を与え、common変数をコントローラのデータメモリに、separate変数を各PEの同一メモリアドレスに、そしてsingle変数を特定の1PEにそれぞれ割り付ける。

RVS-2: A Real-Time Vision System based on Integrated Memory Array Processors —System Software—
Sholin Kyo, Shin'ichiro Okazaki, †Kan Sato, Takuya Koga
Information Technology Research Laboratories,
NEC Corporation
†NEC Informatec Systems, Ltd.



IMAP Memory Window



Global Memory Window

	+0	+1	+2	+3	+4
00000	61516	28673	49335	1	8207
00008	0	4104	4105	49666	4107
00010	13391	8193	36910	36869	8207
00018	2060	49156	16386	61961	49152
00020	57351	8204	49214	57382	22831
00028	32776	16391	36865	4099	33033
00030	27673	16417	40975	16451	53402
00038	8192	21512	12	49152	24832
00040	36960	4107	61455	33800	49289
00048	49152	16397	28674	0	0
00050	31115	57422	57886	13	63519
00058	53252	8204	4100	58966	12320
00060	57673	53256	41481	12366	21518
00068	8450	32775	5377	33285	0
00070	28765	4098	62621	1	57603
00078	20481	6144	28708	4096	45603
00080	57355	40974	57346	10	4399
00088	32770	34820	49161	4100	40968

Program Memory Window

```
1000  nop :: nma :: cnop :: cimm 0x0000 :: iena
1001  nop :: nma :: cnop :: cimm 0x0000 :: idis
1002  nop :: nma :: cnop :: cimm 0x0000 :: idis
1003  nop :: nma :: call 0x3509 :: cimm 0x3509 :: idis
1004  nop :: nma :: cnop :: cimm 0x0000 :: idis
1005  nop :: nma :: call 0x1009 :: cimm 0x1009 :: idis
1006  inv imm, mr :: nma :: cout c0, 0x0000, F :: cimm 0x0000 :: idis
1007  nop :: nma :: ret :: cimm 0x0000 :: idis
1008  nop :: nma :: cnop :: cimm 0x0000 :: idis
1009  nop :: nma :: cnop :: cimm 0x0000 :: idis
100a  nop :: idl :: cout 0x0400, c0, F :: cimm 0x0400 :: idis
100b  nop :: idh :: cout 0x0400, c0, F :: cimm 0x0400 :: idis
100c  inv mdr, mdr :: nma :: cout c0, c0, F :: cimm 0x0000 :: idis
100d  inv imm, mdr :: nma :: cnop :: cimm 0x0000 :: idis
100e  nop :: stl :: cout 0x0400, c0, F :: cimm 0x0400 :: idis
100f  nop :: sth :: cout 0x0400, c0, F :: cimm 0x0400 :: idis
1010  nop :: nma :: cnop :: cimm 0x0000 :: idis
```

RVS Controller Chart Window

	0		2		4		6	
pc	10	13107	13108	13109	13110	13111	13112	13113
sp	0	4	4	4	4	4	4	4
psw	512	512	512	512	512	512	512	512
ilatch								
(cfunc)	63	35	35	63	35	63	35	63
(ifunc)	63	63	63	63	3	40	3	40
(mfunc)	8	0	1	8	8	8	8	8
(intr)								
c0	0	57	57	57	57	57	57	57
c1	0	1080	1080	1080	1080	1080	1080	1080
c2	0	456	456	456	456	456	456	456
c3	0	2104	2104	2104	2104	2104	2104	2104
c4	0	0	0	0	0	0	0	0
c5	0	16	16	16	16	16	16	16

図1: 基本ソフトウェア構成とデバッガの画面

separate変数とは、各PE毎に異なる値を持ち得る変数である(図2)。それ以外の変数は、実行時に必要に応じその値が全PEへ放送される。separate変数及びその配列は常に全PEによって一斉に処理され、またalign修飾子付きで宣言されたseparate(変数の)配列は、IMAP-2の間接アドレッシング機能を効率よく利用できるようにメモリのページ境界に配置される。single変数は特定の1PE上に存在し、その配列の各要素は通常当該PEのアドレス方向に順に配置されるが、配列をspread修飾子付きで宣言した場合はその各要素がPE方向に配置され(図2)、これに応じコンパイラは前者に対し1PEによる逐次処理、後者に対し要素数だけのPEによる並列処理のコードを生成する。

構造文ではmfor文やmif文のように頭にmがつく複文は、その述語部分の評価結果が真であるPE上でのみ実行され、通常のfor文やif文等の複文はコントローラ上で実行される。また一般的なC言語の演算子以外に、separate変数aをオペランドとする演算子として、:||aが全PE上のaの値の論理OR、:<aが右隣PE、:>aが左隣PE上のaの値の参照、そしてa:[x:]がx番目のPE上のaの値の参照を意味する(図2)。例として3×3の平均値フィルタ処理の1DC記述を以下に示す。但しINは外部入力画像、OUTは外部出力画像を格納するseparate配列へのポインタとする。

```
average_filter()
{
   separate unsigned char   acc;
   common   int             i,idx;
   for(idx=0; idx<512; idx++){
      for (i=acc=0;i<3;i++)  acc += IN[idx+i];
      acc += (:<acc + :>acc);
      OUT[idx] = (acc /= 9);
   }
}
```

4 リアルタイム画像処理プログラムの開発

RVS-2ではビデオボードからコントローラに、奇数/偶数フィールド開始/終了、奇数/偶数ライン入力終了、の計6つのビデオ信号が割込み信号として送られて来る。ユーザ定義のプログラムは、これらの割込み信号によって、フレームごとに繰り返し起動される。なお1フレーム時間(33ms)内にユーザプログラムが終了しなかった場合は、その起動を見合すこともできる。

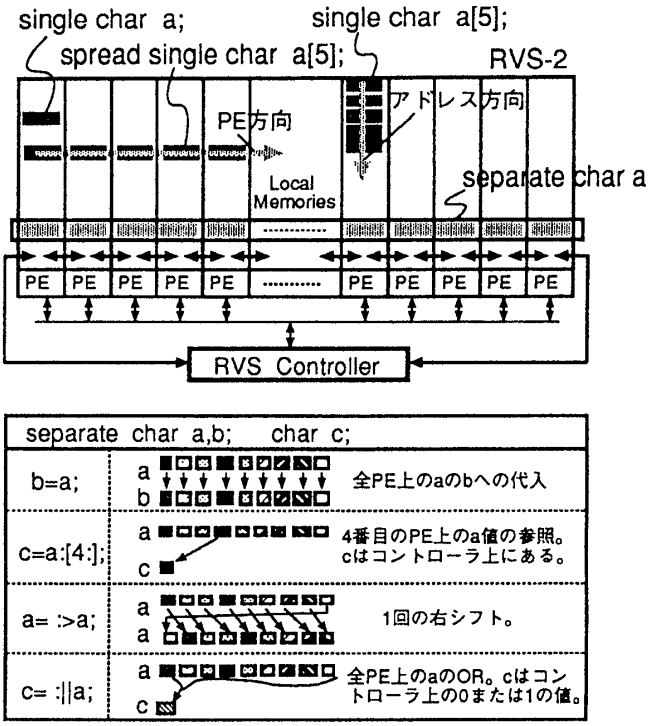


図2: データのメモリ配置とseparate変数の演算

1DCではこれらのビデオ信号に対応し、計6種類の名前の異なる実行開始関数(C言語のmain関数にあたる)を記述でき、それぞれが独立したジョブとして各ビデオ信号に同期して起動される。例えば外部との画像入出力を実現するにはライン入力信号が起動する実行開始関数内に、シフトレジスタ内の1行の入力画像データを一斉に所定のメモリ行にストアし、また別の所定のメモリ行のデータを一斉に同シフトレジスタにロードする処理を記述すればよい。コンパイラはコード生成に際し各ジョブ内に適宜の数の割込み可能領域を設けることで、ジョブ実行中に他のビデオ信号の割込みを許可する。

5 おわりに

本稿では主にIMAPのシステム記述用高級言語1DC、及びそれを用いたリアルタイム画像処理プログラムの開発方法について述べた。こうした高級言語を中心としたプログラミング環境の存在は、今後のRVS-2の適用分野の拡大に大きく貢献することと思われる。

参考文献

[1] 岡崎 他, メモリ型プロセッサによる動画像処理システムRVS-2 — システム構成と特徴 —, 第49回情処全大,5R-4 (1994).

[2] 許, 面型画像処理言語の提案 — 点集計型演算に基づく並列画像処理用言語2DC—, 信学技報, CPSY94-45, pp.33-40 (1994).