

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4203480号  
(P4203480)

(45) 発行日 平成21年1月7日(2009.1.7)

(24) 登録日 平成20年10月17日(2008.10.17)

(51) Int.Cl.		F I			
<b>G06T</b>	<b>1/20</b>	<b>(2006.01)</b>	G06T	1/20	B
<b>H04N</b>	<b>1/46</b>	<b>(2006.01)</b>	H04N	1/46	Z
<b>H04N</b>	<b>1/60</b>	<b>(2006.01)</b>	H04N	1/40	D

請求項の数 4 (全 13 頁)

(21) 出願番号	特願2005-52323 (P2005-52323)	(73) 特許権者	505012542
(22) 出願日	平成17年2月28日(2005.2.28)		今井 正治
(65) 公開番号	特開2006-236175 (P2006-236175A)		大阪府豊中市上野西2-18-14
(43) 公開日	平成18年9月7日(2006.9.7)	(73) 特許権者	398034168
審査請求日	平成17年9月13日(2005.9.13)		株式会社アクセル
			東京都千代田区外神田四丁目14番1号
		(74) 代理人	100101982
			弁理士 久米川 正光
		(72) 発明者	今井 正治
			大阪府豊中市上野西2-18-14
		(72) 発明者	吉村 龍洋
			大阪府吹田市山田東4-36-29 ヴィ ラエスポワール202

最終頁に続く

(54) 【発明の名称】 画像処理プロセッサ

(57) 【特許請求の範囲】

【請求項1】

画像処理プロセッサにおいて、  
複数の色成分について共通で用いられ、共通演算部と、共通レジスタとを有する共通処理ユニットと、

複数の色成分のそれぞれに対応して色成分毎に専用で用いられ、それぞれが、専用演算回路と、専用レジスタとを有し、各色成分の処理を並行的に実行する複数の専用処理ユニットと、

前記共通処理ユニットと、前記複数の専用処理ユニットとを制御するとともに、前記共通処理ユニットから前記複数の専用処理ユニットへのデータ転送と、前記複数の専用処理ユニットから前記共通処理ユニットへのデータ転送とを制御する制御部とを有し、

前記共通レジスタには、前記共通演算部から出力されたデータおよび前記専用処理ユニットから出力されたデータのいずれかが格納され、

前記専用レジスタには、当該専用レジスタに対応する前記専用演算回路から出力されたデータが格納され、

前記制御部は、前記専用レジスタに格納されたデータを、当該専用レジスタに対応する前記専用演算回路および前記共通処理ユニットのいずれかに出力させることを特徴とする画像処理プロセッサ。

【請求項2】

画像処理手順を規定する一連の命令に基づいて、画像処理を実行する画像処理プロセ

10

20

サにおいて、

複数の色成分について共通で用いられ、共通演算部と、共通レジスタとを有する共通処理ユニットと、

複数の色成分のそれぞれに対応して色成分毎に専用で用いられ、それぞれが、専用演算回路と、専用レジスタとを有し、各色成分の処理を並行的に実行する複数の専用処理ユニットと、

前記共通処理ユニットと、前記複数の専用処理ユニットとを制御するとともに、前記共通処理ユニットから前記複数の専用処理ユニットへのデータ転送と、前記複数の専用処理ユニットから前記共通処理ユニットへのデータ転送とを制御する制御部とを有し、

前記共通レジスタには、前記共通演算部から出力されたデータおよび前記専用処理ユニットから出力されたデータのいずれかが格納され、

前記専用レジスタには、当該専用レジスタに対応する前記専用演算回路から出力されたデータが格納され、

前記制御部は、各色成分の演算を実行する第1の命令に応じて、前記専用レジスタに格納されたデータを当該専用レジスタに対応する前記専用演算回路の入力とした演算を並行的に実行させるとともに、各色成分のデータを出力する第2の命令に応じて、前記専用レジスタのそれぞれに格納されたデータを前記共通処理ユニットに一括で出力させることを特徴とする画像処理プロセッサ。

【請求項3】

前記制御部は、各色成分のデータを所定値に設定する第3の命令に応じて、前記専用レジスタのそれぞれに格納されたデータを前記所定値に一括で設定することを特徴とする請求項2に記載された画像処理プロセッサ。

【請求項4】

前記共通レジスタは、ビット幅nの記憶領域を有し、

前記専用レジスタは、前記画像データを構成する画素値の各色成分のビット幅よりも大きなビット幅を有する中間データを格納するために、前記共通レジスタよりも大きなビット幅m ( $m > n$ ) の記憶領域を有し、

前記専用レジスタに格納されたデータを前記共通処理ユニットに出力する場合、当該データをビット幅mからビット幅nに整数化することを特徴とする請求項1から3のいずれかに記載された画像処理プロセッサ。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、画像データを入力とした画像処理を実行する画像処理プロセッサに関する。

【背景技術】

【0002】

近年の画像処理技術の向上により、リアルタイムで生成される画像において、様々な効果を引き出すために、画像データの加工技術の多様化が進んでいる。例えば、特許文献1には、入力された命令およびデータに基づいて所定の画像処理を行う汎用処理ユニットのみならず、少なくとも一つの専用処理ユニットをも備えた画像処理プロセッサが開示されている。専用処理ユニットは、透視変換/光源処理、セットアップ/DDA処理またはフラグメント処理といったように、予め定められた所定の画像処理のみを行う。

【特許文献1】特開2003-108989号公報

【発明の開示】

【発明が解決しようとする課題】

【0003】

本発明の目的は、画像処理の高速化を図り得る新規な画像処理プロセッサを提供することである。

【課題を解決するための手段】

【0004】

10

20

30

40

50

かかる課題を解決するために、第1の発明は、画像処理プロセッサを提供する。この画像処理プロセッサは、複数の色成分について共通で用いられる共通処理ユニットと、複数の色成分のそれぞれに対応して色成分毎に専用で用いられる複数の専用処理ユニットと、制御部とを有する。共通処理ユニットは、共通演算部と、共通レジスタとを有する。それぞれの専用処理ユニットは、専用演算回路と、専用レジスタとを有し、各色成分の処理を並行的に実行する。制御部は、共通処理ユニットと、複数の専用処理ユニットとを制御する。また、制御部は、共通処理ユニットから複数の専用処理ユニットへのデータ転送と、複数の専用処理ユニットから共通処理ユニットへのデータ転送とを制御する。共通レジスタには、共通演算部から出力されたデータおよび専用処理ユニットから出力されたデータのいずれかが格納される。専用レジスタには、この専用レジスタに対応する専用演算回路から出力されたデータが格納される。制御部は、専用レジスタに格納されたデータを、この専用レジスタに対応する専用演算回路および共通処理ユニットのいずれかに出力させる。

10

#### 【0005】

第2の発明は、画像処理手順を規定する一連の命令に基づいて、画像処理を実行する画像処理プロセッサを提供する。この画像処理プロセッサは、複数の色成分について共通で用いられる共通処理ユニットと、複数の色成分のそれぞれに対応して色成分毎に専用で用いられる複数の専用処理ユニットと、制御部とを有する。共通処理ユニットは、共通演算部と、共通レジスタとを有する。それぞれの専用処理ユニットは、専用演算回路と、専用レジスタとを有し、各色成分の処理を並行的に実行する。制御部は、共通処理ユニットと、複数の専用処理ユニットとを制御する。また、制御部は、共通処理ユニットから複数の専用処理ユニットへのデータ転送と、複数の専用処理ユニットから共通処理ユニットへのデータ転送とを制御する。共通レジスタには、共通演算部から出力されたデータおよび専用処理ユニットから出力されたデータのいずれかが格納される。専用レジスタには、この専用レジスタに対応する専用演算回路から出力されたデータが格納される。制御部は、各色成分の演算を実行する第1の命令に応じて、専用レジスタに格納されたデータをこの専用レジスタに対応する専用演算回路の入力とした演算を並行的に実行させる。また、制御部は、各色成分のデータを出力する第2の命令に応じて、専用レジスタのそれぞれに格納されたデータを共通処理ユニットに一括で出力させる。

20

#### 【0006】

ここで、第2の発明において、制御部は、各色成分のデータを所定値に設定する第3の命令に応じて、専用レジスタのそれぞれに格納されたデータを所定値に一括で設定することが好ましい。

30

#### 【0007】

第1または第2の発明において、共通レジスタは、ビット幅 $n$ の記憶領域を有するとともに、専用レジスタは、画像データを構成する画素値の各色成分のビット幅よりも大きなビット幅を有する中間データを格納するために、共通レジスタよりも大きなビット幅 $m$  ( $m > n$ )の記憶領域を有することが好ましい。その際、専用レジスタに格納されたデータを共通処理ユニットに出力する場合には、このデータをビット幅 $m$ からビット幅 $n$ に整数化することが望ましい。

40

#### 【発明の効果】

#### 【0008】

本発明によれば、共通処理ユニットに加え、色成分ごとに個別に設けられた複数の専用処理ユニットを設けることで、共通処理ユニットのみを用いた処理と比較して、画像処理の高速化を図ることができる。

#### 【0009】

また、専用処理ユニットに専用レジスタを内蔵し、自己の演算回路による処理結果を専用レジスタに格納することで、専用処理ユニット外のレジスタの使用頻度を減らすことができるので、画像処理の高速化を図ることができる。

#### 【発明を実施するための最良の形態】

50

## 【 0 0 1 0 】

図 1 は、本実施形態に係る画像処理プロセッサ 1 の概略的なブロック構成図である。画像処理プロセッサ 1 は、例えばフレームメモリ 2 に格納されている画素値（輝度を示す RGB データと透明度を示す データとを含む）を入力として、画像データに対する演算が必要な画像処理を実行する。なお、この画像処理プロセッサ 1 の特徴は、画像処理演算部 5 の構成にあり、それ以外については一般的な構成を有する。

## 【 0 0 1 1 】

画像処理プロセッサ 1 は、制御部 3 と、命令格納部 4（例えばレジスタ）と、画像処理演算部 5 とから構成されている。制御部 3 は、命令格納部 4 にアクセスし、個々に格納されている一連の命令（画像処理の手順を規定）にしたがって、画像処理演算部 5 を制御する。

10

## 【 0 0 1 2 】

画像処理演算部 5 は、共通処理ユニット 6 と、複数の専用処理ユニット 7 とから構成される。共通処理ユニット 6 は、共通演算部 6 a と共通レジスタ 6 b とを有し、複数の色成分について共通で用いられ、ある命令に応じた処理を実行する。共通演算部 6 a は、制御部 3 からの命令に応じた処理を実行する。共通レジスタ 6 b は、共通演算部 6 a での処理結果である出力データ、または、専用処理ユニット 7 での処理結果である出力データを一時的に格納する。複数の専用処理ユニット 7 は、複数の色成分（RGB）のそれぞれに対応して色成分毎に専用で用いられ、ある命令に応じた処理をそれぞれが並行的に実行する。

20

## 【 0 0 1 3 】

図 2 は、専用処理ユニット 7 のブロック構成図である。それぞれの色成分（RGB）に対応した各専用処理ユニット 7 は、専用加算器 7 a および専用乗算器 7 b の双方を含む専用演算回路と、専用セクタ 7 c と、専用レジスタ 7 d とから構成されている。専用セクタ 7 c は、専用加算器 7 a から出力されたデータ、専用乗算器 7 b から出力されたデータおよび前記共通ユニットから転送されたデータのいずれかを後段の専用レジスタ 7 d に出力し、格納する。制御部 3 は、この専用レジスタ 7 d に格納されたデータを専用演算回路への入力としてフィードバックし、または、この格納されたデータを共通処理ユニット 6 に出力する。

## 【 0 0 1 4 】

それぞれの専用処理ユニット 7 は、画素値を構成する各色成分本来のビット幅（ $n$  ビット）よりも大きなビット幅（ $m$  ビット： $m > n$ ）を有する中間データを処理対象としている。そのため、専用処理ユニット 7 に設けられた専用レジスタ 7 d は、この中間データを格納すべく、少なくとも  $m$  ビット幅の記憶領域を有する。本実施形態において、共通レジスタのビット幅は各色成分それぞれ 8 ビット（ $n = 8$ ）であり、専用レジスタ 7 d のビット幅は各色成分それぞれ 32 ビット（ $m = 32$ ）である。

30

## 【 0 0 1 5 】

つぎに、命令格納部 4 に格納される各種命令（コマンド）のうち、本実施形態に固有の命令である専用命令について詳述する。以下は、説明を簡略化すべく、フレームメモリ 2 の画像データは、共通レジスタ 6 b に既に転送・格納されているものとする。

40

## 【 0 0 1 6 】

## [ データ設定命令 ( rgbset ) ]

図 3 は、データ設定命令 ( rgbset ) の説明図である。この命令 ( rgbset ) は、各専用レジスタ 7 d の値を所定の値に一括かつ画的に設定する命令である。具体的には、命令中の引数が設定値として専用レジスタ 7 d に格納される。例えば、命令「rgbset 0」、すなわち、この命令に記述された設定値が「0」（初期化）の場合、RGB それぞれの専用レジスタ 7 d の値が全て「0」に設定される。

## 【 0 0 1 7 】

## [ 加減算命令 ( rgbadd ) ]

図 4 は、加減算命令 ( rgbadd ) の説明図である。この命令 ( rgbadd ) は、命令中の引数

50

によってアドレス指定された共通レジスタ 6 b の RGB データの値と、各専用レジスタ 7 d に格納された値との加算を同時並行的に行い、この演算結果を各専用レジスタ 7 d へ上書格納する命令である。具体的には、まず、命令中に記述された共通レジスタ 6 b の指定アドレスに格納されている RGB データの値が各専用加算器 7 a に転送される。つぎに、専用加算器 7 a は、共通レジスタ 6 b より読み出された値と、専用レジスタ 7 d よりフィードバックされた値との加算を行う。そして、専用セクタ 7 c の選択により、専用加算器 7 a の出力が専用レジスタ 7 d に転送され、元のデータに上書きされる。例えば、共通レジスタ 6 b のアドレス r0 に RGB データの値 (R0, G0, B0) が纏めて格納され、R 用の専用レジスタ 7 d に「Rind」、G 用のそれに「Gind」、B 用のそれに「Bind」がそれぞれ格納されているケースを考える。この場合、「rgbbadd r0」が実行されると、それぞれの専用レジスタ 7 d の値 Rind, Gind, Bind は以下のように更新される。

```
Rind      Rind + R0
Gind      Gind + G0
Bind      Bind + B0
```

#### 【0018】

なお、専用加算器 7 a は、加算のみならず減算も可能である。減算は、共通レジスタ 6 b に格納されている値または専用レジスタ 7 d に格納されている値の一方を、マイナスの値にすることによって実行される。本実施形態を実装する場合は、専用加算器 7 a の入力側にそれぞれの値に対するビット反転回路を設け、専用加算器 7 a に入力される値の一方に対するビット反転を、制御部 3 から制御することで実現する。なお、専用加算器 7 a は、このビット反転回路を含めて加減算器とみなすこともできる。

#### 【0019】

[乗除算命令 (rgbmult)]

図 5 は、乗除算命令 (rgbmult) の説明図である。この命令 (rgbmult) は、専用レジスタ 7 d に格納された RGB データの値と、任意の値とで乗算を同時並行的に行い、この演算結果を専用レジスタ 7 d へ上書きして格納する命令である。具体的には、まず、命令中に引数として記述された指定アドレスに格納されている値が共通レジスタ 6 b より読み出され、これが各専用乗算器 7 b に転送される。つぎに、専用乗算器 7 b は、この共通レジスタ 6 b から転送された値と、専用レジスタ 7 d に格納された値との乗算を行う。そして、専用セクタ 7 c の選択により、専用乗算器 7 b の出力が専用セクタ 7 c に転送され、元のデータに上書きされる。例えば、共通レジスタのアドレス r0 に定数 Const が格納され、R 用の専用レジスタ 7 d に「Rind」、G 用のそれに「Gind」、B 用のそれに「Bind」がそれぞれ格納されているケースを考える。この場合、「rgbmult r0」が実行されると、それぞれの専用レジスタ 7 d の値 Rind, Gind, Bind は以下のように更新される。

```
Rind      Rind x Const
Gind      Gind x Const
Bind      Bind x Const
```

#### 【0020】

なお、専用乗算器 7 b は、乗算のみならず除算も可能である。除算は、共通レジスタ 6 b の値が 1 未満の場合に実行される。本実施形態を実装する場合は、共通レジスタ 6 に格納されている除算したい値 (例えば a とする) を、あらかじめ共通処理ユニット 6 で計算しておき、この計算結果の値 (1/a) を乗算することで実現する。

#### 【0021】

[データ出力 (rgbget)]

図 6 は、データ出力命令 (rgbget) の説明図である。この命令 (rgbget) は、専用レジスタ 7 d に格納された RGB データの各値を一括して出力し、これらの値を共通レジスタ 6 b における引数としての指定アドレスに格納する命令である。具体的には、まず、R G

10

20

30

40

50

Bそれぞれの専用レジスタ7 dに格納されている値が出力され、命令中の指定アドレスにしたがって、これらの値が共通レジスタ6 bへ格納される。例えば、専用レジスタ7 dのそれぞれにR ind, G ind, B indが格納されていたとする。ここで、「rgbset r0」が実行されると、R ind, G ind, B indの各値が、アドレスr0の共通レジスタ6 bに纏めて同時並行的に格納される。なお、専用レジスタ7 dのビット幅は、色成分ごとにそれぞれ32ビット設けられているが、共通レジスタ6 bの各色成分のビット幅は、8ビットしかないので、それぞれ整数化(8ビット化)されて、共通レジスタ6 bに格納される。

【0022】

本実施形態に係る画像処理プロセッサ1は、上述したような固有の専用命令と、それ以外の汎用命令とを組み合わせることによって、様々な画像処理を実行可能である。以下、このような画像処理の一例について、プログラムリストを例示しながら詳述する。

10

【0023】

(モザイク処理・ぼかし処理)

図7は、モザイク処理のプログラムリストである。モザイク処理とは、フレームメモリ2上に配置されている画像領域を所定サイズの画素ブロックに分け、その画素ブロック内で画素の平均値を算出する処理である。一例として、画像ブロックを2×2サイズとし、利用する画素値は互いに隣接する4つのアドレス(Addr1, Addr2, Addr3, Addr4)に格納されているものとする。

【0024】

なお、同図のプログラムリストに記載されている命令「read」は、フレームメモリ2に格納されているRGBデータの値を、共通レジスタ6 bに読み出す命令である。また、「write」は、共通レジスタ6 bに格納されているRGBデータの値をフレームメモリ2に書き出す命令である。例えば、「read r0, Addr1」が読み出されると、フレームメモリ2のアドレスAddr1におけるRGBデータの値が共通レジスタ6 bのr0に格納される。また、「write r0, Addr1」が読み出されると、共通レジスタ6 bに格納されているRGBデータの値が、フレームメモリ2のAddr1に書き出される。

20

【0025】

1行目の命令「rgbset 0」によって、各色成分の専用レジスタ7 dが0に設定される。つぎに、2行目の命令「read r0, Addr1」によって、フレームメモリ2のアドレスAddr1に格納されているRGBデータの各値が、共通レジスタ6 bのアドレスr0に転送される。3行目の命令「rgbadd r0」によって、共通レジスタ6 bのアドレスr0に格納されているRGBデータの値R0, G0, B0が、色成分ごとに各専用処理ユニット7の専用加算器7 aへ同時並行的にデータ転送される。転送されたこれらのデータは、専用加算器7 aにおいて、各専用レジスタ7 dに格納されている値と加算され、専用セクタ7 cを介して、各専用レジスタ7 dへ上書きして格納される。

30

【0026】

そして、4行目～9行目の命令において、2～3行目の命令と同様に、フレームメモリ2のアドレスを順次Addr2～Addr4とシフトさせながら、各色成分の値を共通レジスタ6 bに格納する。共通レジスタ6 bに格納された各色成分の値は、各専用レジスタ7 dの値と合算される。

40

【0027】

10行目の命令「rgbmult 1/4」によって、各専用レジスタ7 dに格納されている値が1/4で乗算(4で除算)される。11行目の命令「rgbget r0」によって、各専用レジスタ7 dに格納されている値が共通レジスタ6 bのアドレスr0に対して纏めて返される。最後に、12～15行目の命令「write r0, Addrn」(n=1～4)によって、共通レジスタ6 bのアドレスr0に格納されている各RGBデータの値が、フレームメモリ2の各アドレス(Addr1, Addr2, Addr3, Addr4)に返される。以上の処理によって、モザイク処理が完了する。

【0028】

なお、上記モザイク処理の変形例として、ぼかし処理を実行することも可能である。ぼ

50

かし処理は、モザイク処理と比べて、平均値を算出するメモリフレームのアドレスが、互いに隣接するブロック状（矩形状）でない点で異なる。プログラム上におけるぼかし処理は、フレームメモリ内の各アドレスについて、そのアドレスの周囲の数ピクセルにおける、RGBデータの値の読み込みと加算（2～3行目のプログラム）とを繰り返し、最後に平均値をとることで実現する。具体的なプログラムについては、モザイク処理と大差ないので、これ以上の説明を省略する。

#### 【0029】

（YUVデータ変換処理）

図8は、YUVデータ変換処理のプログラムリストである。YUVデータ変換処理とは、RGBデータから、YUVデータ（輝度と色差情報からなるデータ）へフォーマットを変換する処理である。RGBデータにおける各色成分の値をR、G、Bとすると、YUVデータにおける各色成分の値Y、U、Vは、以下のような式の計算で求められる。

$$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B \quad \dots \text{式(1)}$$

$$U = -0.147 \times R - 0.289 \times G + 0.436 \times B \quad \dots \text{式(2)}$$

$$V = 0.615 \times R - 0.515 \times G - 0.100 \times B \quad \dots \text{式(3)}$$

#### 【0030】

上記処理を、本実施形態において実装するために、変換するアドレスのR、G、Bの各値を読み込んで、所定の定数と乗算（除算）させるプログラムを設定する。本実施形態において、YUVデータに変換する処理はAddr1に格納されているRGBデータである。

#### 【0031】

なお、同図のプログラムリストに記載されている命令「ldi」とは、任意の定数を、記述されたアドレスを持つ共通レジスタ6bに格納するという命令である。例えば、命令「ldi r0, 10」が実行されると、共通レジスタ6bのr0に、定数「10」が格納される。また、命令「ldi r1, (20, 30)」が実行されると、共通レジスタ6bのr1における上位16ビットに「20」が格納され、r1における下位16ビットに「30」が格納される。これらの命令は、本実施形態に限らず一般のプロセッサでも同様の命令を持っているが、その他に持っている基本的な命令の組み合わせでも同じ処理が実現できるものである。本発明を実装する場合、RGBデータにおける各ビット幅の合計が、専用レジスタ7dに出力されるビット幅以下であればよい。

#### 【0032】

また、命令「rgmult2」とは、前述した命令「ldi」によって格納される共通レジスタ6bの値と、専用レジスタ7dとを乗算するという処理である。例えば、各専用レジスタに格納されている値がR、G、Bであり、上記の命令「ldi」により、共通レジスタ6bのr0に1つ、r1に2つの係数がそれぞれ格納されているとする。このとき、命令「rgmult2 r0, r1」が読み出されると、専用処理ユニット7内で以下の処理が同時並行的に行われる。

$$\begin{array}{ll} R & R \times r0 \\ G & G \times (r1 \text{の上位} 16 \text{ビットに格納された値}) \\ B & B \times (r1 \text{の下位} 16 \text{ビットに格納された値}) \end{array}$$

#### 【0033】

なお、上記命令は、r0～r2のそれぞれに1つの係数が格納されていたとして、「rgmult r0, r1, r2」と記述してもよい。本発明を実装する場合、1行の命令で3つ以上の共通レジスタ6bを指定するのは、あまり一般的ではないため、1つのレジスタに2つの係数を格納した。

#### 【0034】

図8を用いてプログラムの具体的な説明をする。プログラムの全体の流れとしては、まず、1行目の命令によって、フレームメモリ2の処理対象となるアドレスのRGBデータを共通レジスタ6bへ読み取る。そして、2～16行目の命令によって、RGBスケール

を Y U V スケールに変換処理を行い、17 ~ 20 行目の命令によって、共通レジスタ 6 b への Y U V データの出力を行う。

【 0 0 3 5 】

なお、2 ~ 6 行目までの命令は Y データの値を算出する命令であり、同様に、7 行目 ~ 11 行目までの命令は U データの値、12 ~ 16 行目までのプログラムは V データの値をそれぞれ算出する。また、17 行目 ~ 19 行目のプログラムによって、Y, U, V の値が、再度各専用レジスタ 7 d に格納される。そして、20 行目のプログラムによって、専用レジスタ 7 d に格納されていた Y, U, V の値はそれぞれ 8 ビット幅に変換されて、共通レジスタ 6 b に格納される。

【 0 0 3 6 】

Y U V データ変換処理における各成分値の算出するプログラムは類似しており、特に算出に係る係数のみが異なる。そのため、U データおよび V データの算出についての説明は省略し、2 ~ 6 行目のプログラムによる、Y データの値の算出を説明する。まず、2 行目の命令によって、r0 に 0.229 という定数が格納される。つぎに、3 行目の命令によって、r1 の上位 16 ビットに 0.587 が、また下位 16 ビットに 0.114 が格納される。つぎに、4 行目の命令によって、専用レジスタ 7 d に、各色成分に対応した R G B データの値が格納される。つぎに、5 行目の命令によって、R の専用処理ユニット 7 内では、専用レジスタ 7 d に格納されている値 R と、共通レジスタ 6 b に格納されている係数 (0.299) とが乗算され、その結果が専用レジスタに再び格納される。同様に、G および B の専用処理ユニット 7 内では、専用レジスタ 7 d に格納されている値と、r1 に格納されている 2 つの係数 (0.587, 0.114) が乗算され、その結果が専用レジスタに再び格納される。ちなみに、5 行目の命令は式 (1) に相当し、Y の値を算出している。最後に、6 行目の命令によって、共通レジスタ r3 に Y データの値が出力される。

【 0 0 3 7 】

( 畳込処理 )

図 9 は、畳込処理のプログラムリストである。畳み込み (コンボリューション) 処理とは、注目する R G B データとその周囲の R G B データに、任意の重みをかけて足す処理である。なお、この処理は画像表示される物体の輪郭などを検出する画像処理にも適用可能である。

【 0 0 3 8 】

プログラム中では、Addr1 が処理対象の画素のアドレス、Addr2 ~ 9 が周囲の画素のアドレスとし、Const1 ~ Const9 は各ピクセルの重みを表す定数とする。このとき、例えば、Const1 ~ Const9 の値が全て 1/9 である場合、出力される値は、単純に注目ピクセルと周囲のピクセルを足した平均値となり、これはぼかし処理と同一の処理となる。

【 0 0 3 9 】

なお、図 9 に記載されている命令「rgbmuladd r0, r1」は、共通レジスタ 6 b のアドレス r0 に格納されている値に、共通レジスタ 6 b のアドレス r1 に格納されている R G B データの値を乗算し、さらに、各色成分の専用レジスタ 7 d に格納されている値を足す命令である。例えば、r0 に格納されている各色成分の値を Ra, Ga, Ba とし、r1 に格納されている値を Const1 とする。ここで、命令「rgbmuladd r1」が実行されると、以下の処理が同時並行的に行われる。

$$\begin{aligned} R &= R + R \times \text{Const1} \\ G &= G + G \times \text{Const1} \\ B &= B + B \times \text{Const1} \end{aligned}$$

【 0 0 4 0 】

このとき、R G B データのそれぞれに r1 を乗算する回路が、専用処理ユニット 7 の外部にある回路に用意されている場合、乗算処理は、専用乗算器 7 b の外部で行った方がよい。これは、実装上において、制御部 3 が行う他の命令と共有できるようにするためである。

10

20

30

40

50



## 【 0 0 4 1 】

図 9 を用いてプログラムの具体的な説明をする。プログラムの全体的な流れとしては、まず、1 行目の命令によって専用レジスタ 7 d のリセットが行われる。そして 2 ~ 4 行目までの 3 行の命令によって、9 画素うちの 1 つに対して重み付けを行った値が算出される。そして、5 ~ 1 3 行目までにかけて、2 ~ 4 行目と同様の処理が行われ、残りの 8 画素における重み付けを行った値が算出され、算出した結果が順次合算される。最後に 1 4 行目の命令により、各専用レジスタ 7 d に格納された値が、共通レジスタ 6 b へ出力される。

## 【 0 0 4 2 】

畳込処理における各 RGB データの値に重み付けを行うプログラムは類似しており、特に算出に係る係数のみが異なる。そのため、2 番目から 9 番目の RGB データの値における処理の説明は省略し、1 ~ 4 行目のプログラムによる、1 番目の RGB データの値における処理を説明する。まず、1 行目の命令によって、各専用レジスタ 7 d 内の値が 0 に格納される。つぎに、2 行目の命令によって、フレームメモリ 2 に格納されている画素値が共通レジスタ 6 b の r 0 に格納される。つぎに、3 行目の命令によって、共通レジスタ 6 b の r 1 に定数 Const1 が格納される。つぎに、4 行目の命令によって、共通レジスタ r 0 の各色成分に r 1 を乗算し、その乗算結果に専用レジスタ 7 d の値を加算して、再び専用レジスタ 7 d に格納する。5 行目以降の命令については、2 ~ 4 行目の命令を、読み出すフレームメモリ 2 のアドレスを変位させながら繰り返して処理を行う。

## 【 0 0 4 3 】

本実施形態によれば、専用処理ユニット 7 を画像処理装置内に設け、初期化をはじめとした各色成分ごとの演算処理に関する複数の命令を 1 命令で実行する。これにより、各色成分ごとに個別に設けられた複数の専用処理ユニット 7 が処理を並行的に実行するので、共通処理ユニット 6 のみを用いた処理と比較して、画像処理の高速化を図ることができる。また、専用処理ユニット 7 に専用レジスタ 7 d を内蔵し、自己の演算回路による処理結果を専用レジスタ 7 d に出力をすることで、専用処理ユニット 7 外のレジスタの使用頻度を減らすことができるので、より一層の画像処理の高速化を図ることができる。

## 【 0 0 4 4 】

また、専用レジスタ 7 d のビット幅は、共通レジスタ 6 b (またはフレームメモリ 2) に格納されている色成分のビット幅より大きいため、従来であれば演算処理において、3 2 ビットの専用レジスタを 8 ビットの整数型へ変換する必要があったが、本実施形態では、専用加算器 7 a ・専用乗算器 7 b も 3 2 ビットでの対応が可能であるため、さらなる処理速度の高速化を図ることができる。

## 【 図面の簡単な説明 】

## 【 0 0 4 5 】

【 図 1 】 本実施形態に係る画像処理プロセッサの概略的なブロック構成図

【 図 2 】 専用演算部のブロック構成図

【 図 3 】 データ設定命令 ( rgbset ) の説明図

【 図 4 】 加減算命令 ( rgbadd ) の説明図

【 図 5 】 乗除算命令 ( rgbmult ) の説明図

【 図 6 】 データ出力命令 ( rgbget ) の説明図

【 図 7 】 モザイク処理のプログラムリスト

【 図 8 】 YUV データ変換処理のプログラムリスト

【 図 9 】 畳込処理のプログラムリスト

## 【 符号の説明 】

## 【 0 0 4 6 】

- 1 画像処理プロセッサ
- 2 フレームメモリ
- 3 制御部
- 4 命令格納部

10

20

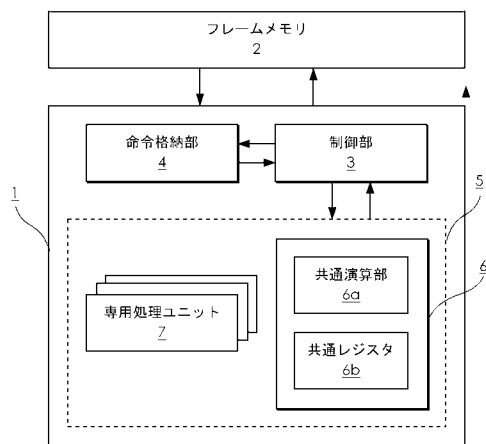
30

40

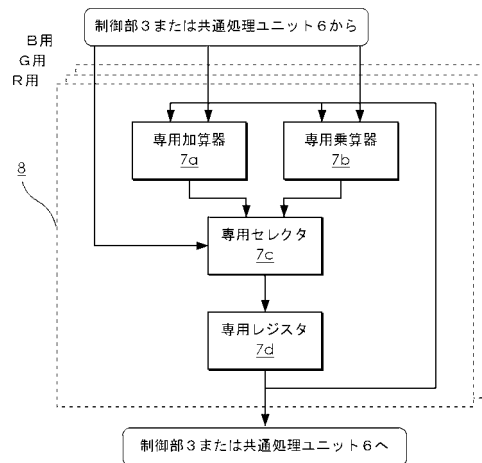
50

- 5 画像処理演算部
- 6 共通処理ユニット
- 6 a 共通演算部
- 6 b 共通レジスタ
- 7 専用処理ユニット
- 7 a 専用加算器
- 7 b 専用乗算器
- 7 c 専用セレクタ
- 7 d 専用レジスタ

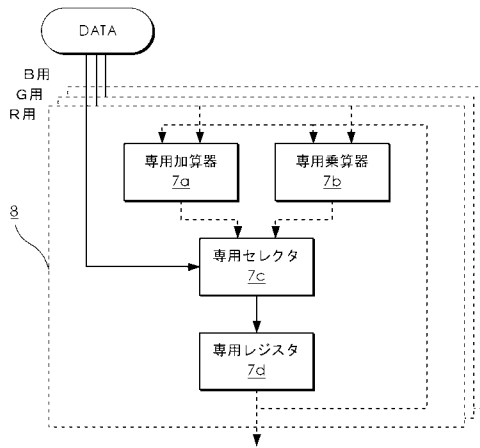
【図 1】



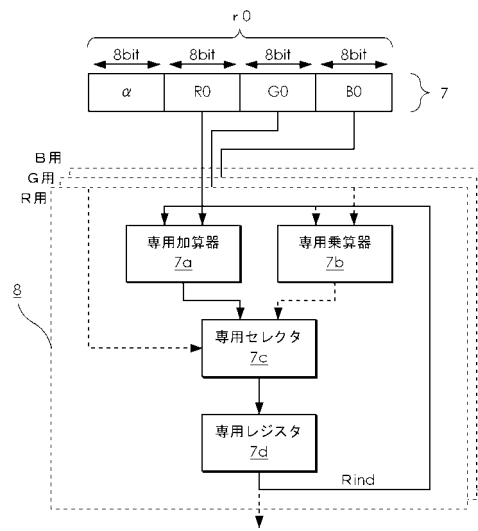
【図 2】



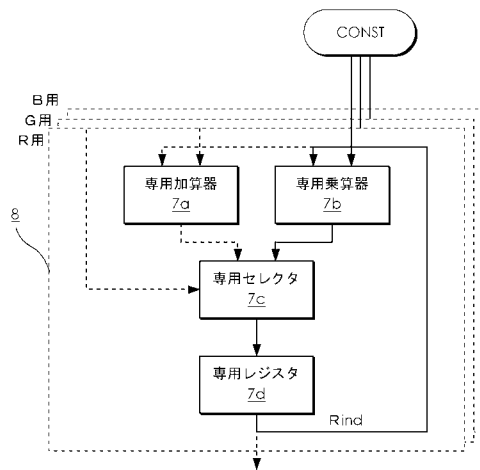
【図3】



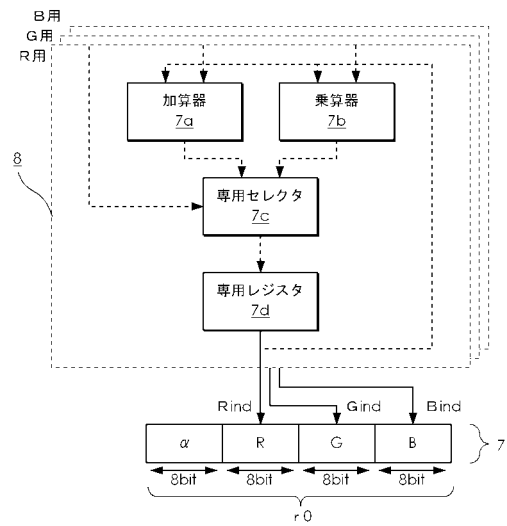
【図4】



【図5】



【図6】



## 【 図 7 】

```

1: rgbset 0
2: read r0, Addr1
3: rgbadd r0
4: read r0, Addr2
5: rgbadd r0
6: read r0, Addr3
7: rgbadd r0
8: read r0, Addr4
9: rgbadd r0
10: rgbmult 1/4
11: rgbget r0
12: write r0, Addr1
13: write r0, Addr2
14: write r0, Addr3
15: write r0, Addr4

```

## 【 図 8 】

```

1: read r2, Addr1
2: ldi r0, 0.299
3: ldi r1, (0.587, 0.114)
4: rgbset r2
5: rgbmult2 r0, r1
6: rgbget r3
7: rgbset r2
8: ldi r0, -0.147
9: ldi r1, (-0.289, 0.436)
10: rgbmult2 r0, r1
11: rgbget r4
12: rgbset r2
13: ldi r0, 0.615
14: ldi r1, (-0.515, -0.100)
15: rgbmult2 r0, r1
16: rgbget r5
17: rset r3
18: gset r4
19: bset r5
20: rgbset r2

```

## 【 図 9 】

```

1: rgbset 0
2: read r0, Addr1
3: ldi r1, Const1
4: rgbmuladd r0, r1
5: read r0, Addr2
6: ldi r1, Const2
7: rgbmuladd r0, r1
8: read r0, Addr3
9: ldi r1, Const3
10: rgbmuladd r0, r1
    .
    .
    .
11: read r0, Addr9
12: ldi r1, Const9
13: rgbmuladd r0, r1
14: rgbget r0

```

---

フロントページの続き

(72)発明者 蟹江 幸司

東京都千代田区飯田橋4 - 8 - 13 タカラビル 株式会社アクセル内

審査官 真木 健彦

(56)参考文献 特開2000 - 148996 (JP, A)

特開平10 - 042150 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G06T 1/20

G06T 1/00 510