

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2008-9702

(P2008-9702A)

(43) 公開日 平成20年1月17日(2008.1.17)

(51) Int. Cl.	F I	テーマコード (参考)
G 0 6 F 12/06 (2006.01)	G 0 6 F 12/06 5 2 2 B	5 B 0 6 0
G 0 6 F 12/00 (2006.01)	G 0 6 F 12/00 5 9 7 U	
	G 0 6 F 12/06 5 2 0 F	

審査請求 未請求 請求項の数 8 O L (全 16 頁)

(21) 出願番号	特願2006-179352 (P2006-179352)	(71) 出願人	000005821
(22) 出願日	平成18年6月29日 (2006.6.29)		松下電器産業株式会社
			大阪府門真市大字門真1006番地
		(74) 代理人	100077931
			弁理士 前田 弘
		(74) 代理人	100110939
			弁理士 竹内 宏
		(74) 代理人	100110940
			弁理士 嶋田 高久
		(74) 代理人	100113262
			弁理士 竹内 祐二
		(74) 代理人	100115059
			弁理士 今江 克実
		(74) 代理人	100115691
			弁理士 藤田 篤史

最終頁に続く

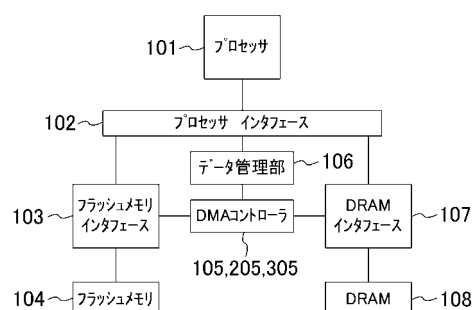
(54) 【発明の名称】 演算処理システム

(57) 【要約】

【課題】プロセッサの負荷を増加させることなく、アクセスに所定のルーチンを必要とするフラッシュメモリ等のメモリからの、読み出しにおける待ち時間を短縮する演算処理システムを提供することを目的とする。

【解決手段】演算処理システムであって、プロセッサと、前記プロセッサによるアクセスの対象であるデータを格納する第1のメモリと、転送されたデータを格納するための領域を有する第2のメモリと、前記プロセッサによるアクセスに従って、データを前記第1のメモリから前記第2のメモリの前記領域に転送させ、前記領域に格納されているデータを示す領域情報を記憶するデータ管理部とを備え、前記データ管理部は、前記プロセッサによるアクセスの対象が前記領域情報に示されているデータである場合は、前記プロセッサを前記第2のメモリにアクセスさせるものである。

【選択図】 図1



【特許請求の範囲】

【請求項 1】

プロセッサと、
前記プロセッサによるアクセスの対象であるデータを格納する第 1 のメモリと、
転送されたデータを格納するための領域を有する第 2 のメモリと、
第 1 のメモリに対するアクセスを制御する第 1 のアクセスコントローラと、
第 2 のメモリに対するアクセスを制御する第 2 のアクセスコントローラと、
前記第 1 のアクセスコントローラ及び前記第 2 のアクセスコントローラを介して、第 1
のメモリから第 2 のメモリにデータを転送するデータ転送部と、
前記プロセッサによるアクセスに従って、前記データ転送部に、データを前記第 1 のメ
モリから前記第 2 のメモリの前記領域に転送させ、前記領域に格納されているデータを示
す領域情報を記憶するデータ管理部とを備え、
前記データ管理部は、
前記プロセッサによるアクセスの対象が前記領域情報に示されているデータである場合
は、前記プロセッサを前記第 2 のメモリにアクセスさせるものである
ことを特徴とする演算処理システム。

10

【請求項 2】

請求項 1 記載の演算処理システムにおいて、
前記データ管理部は、
前記領域への書き込みを、データの優先度に応じて行うものである
ことを特徴とする演算処理システム。

20

【請求項 3】

請求項 1 記載の演算処理システムにおいて、
前記第 1 のアクセスコントローラは、
前記第 1 のメモリが、アクセスに所定のルーチンを必要とする種類のメモリ又はアドレ
スバスから受けたアドレスによってアクセスされる種類のメモリのいずれであっても、前
記プロセッサによるアクセスができるように、前記第 1 のメモリに対するアクセスを制御
するものである
ことを特徴とする演算処理システム。

30

【請求項 4】

請求項 3 記載の演算処理システムにおいて、
前記第 1 のアクセスコントローラは、
前記第 1 のメモリが、アクセスに所定のルーチンを必要とする種類のメモリ又はアドレ
スバスから受けたアドレスによってアクセスされる種類のメモリのいずれであっても、前
記プロセッサが同一のアドレス空間を用いてアクセスすることができるようにするもの
である
ことを特徴とする演算処理システム。

40

【請求項 5】

請求項 1 記載の演算処理システムにおいて、
前記データ転送部は、
前記演算処理システムの起動時に、前記第 1 のメモリが格納するデータのうち所定の領
域のデータを、自動的に前記第 2 のメモリに転送するものである
ことを特徴とする演算処理システム。

【請求項 6】

請求項 1 記載の演算処理システムにおいて、
前記データ管理部は、
前記プロセッサからの要求に応じて、前記領域の大きさを変化させるものである
ことを特徴とする演算処理システム。

【請求項 7】

請求項 1 記載の演算処理システムにおいて、

50

前記データ転送部は、

前記第 1 のメモリから前記第 2 のメモリに転送中のデータを格納するバッファメモリを備えるものであり、

前記データ管理部は、

前記プロセッサによるアクセスの対象が、前記バッファメモリが格納するデータと一致する場合には、前記プロセッサにそのデータを前記バッファメモリから読み出させるものである

ことを特徴とする演算処理システム。

【請求項 8】

請求項 1 記載の演算処理システムにおいて、

10

前記データ管理部は、

前記プロセッサによってアクセスされたデータを格納する領域の次の領域のデータを、前記第 1 のメモリから前記第 2 のメモリにさらに転送させるものである

ことを特徴とする演算処理システム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、プロセッサを搭載した演算処理システムに関する。

【背景技術】

【0002】

20

デジタルスチルカメラ等のシステム L S I (large-scale integrated circuit) において、大容量かつ比較的成本の安い N A N D 型フラッシュメモリが、広く用いられるようになった。プロセッサは、プログラム等を N A N D 型フラッシュメモリから直接読み出すことができず、プログラム等は、N A N D 型フラッシュメモリから D R A M (dynamic random access memory) 等の別のメモリに転送された上で読み出されるため、プロセッサには待ち時間が生じていた。

【0003】

そこで、プロセッサの待ち時間を短縮するために、カウンタを用いてデータ転送を監視し、既に転送されているデータについては、転送先のメモリから読み出すことを可能にしたマイクロコントローラが、例えば、下記特許文献 1 に開示されている。

30

【特許文献 1】特開 2 0 0 2 - 2 9 7 4 4 5 号公報

【発明の開示】

【発明が解決しようとする課題】

【0004】

しかしながら、従来のマイクロコントローラでは、転送先のメモリに十分な容量がない場合、全てのデータを転送することができないため、プロセッサが必要に応じてデータを転送することになる。また、プログラムやデータの構成により、必ずしも転送元のアドレス順に転送先に転送されるとは限らないため、カウンタが保持するアドレスでは、転送先に格納されているかどうかの判断ができなかった。このため、プロセッサが転送された領域を管理する必要があり、プロセッサの負荷が増すという問題があった。

40

【0005】

また、転送が完了し、プロセッサによる利用も完了したデータを、転送先のメモリから効率的に除去する必要がある。

【0006】

本発明は、プロセッサの負荷を増加させることなく、アクセスに所定のルーチンを必要とするフラッシュメモリ等のメモリからの読み出しにおける待ち時間を短縮する演算処理システムを提供することを目的とする。

【課題を解決するための手段】

【0007】

前記課題を解決するため、請求項 1 の発明が講じた手段は、演算処理システムであって

50

、プロセッサと、前記プロセッサによるアクセスの対象であるデータを格納する第１のメモリと、転送されたデータを格納するための領域を有する第２のメモリと、第１のメモリに対するアクセスを制御する第１のアクセスコントローラと、第２のメモリに対するアクセスを制御する第２のアクセスコントローラと、前記第１のアクセスコントローラ及び前記第２のアクセスコントローラを介して、第１のメモリから第２のメモリにデータを転送するデータ転送部と、前記プロセッサによるアクセスに従って、前記データ転送部に、データを前記第１のメモリから前記第２のメモリの前記領域に転送させ、前記領域に格納されているデータを示す領域情報を記憶するデータ管理部とを備え、前記データ管理部は、前記プロセッサによるアクセスの対象が前記領域情報に示されているデータである場合は、前記プロセッサを前記第２のメモリにアクセスさせるものである。

10

【０００８】

請求項１の発明によると、データ管理部は、どの領域のデータが第１のメモリから第２のメモリに転送されたかを管理し、プロセッサによるアクセスの対象のデータが第２のメモリに存在する場合は、プロセッサを第２のメモリにアクセスさせる。このため、プロセッサがデータ転送を管理する必要がない。

【０００９】

請求項２の発明は、請求項１記載の演算処理システムにおいて、前記データ管理部は、前記領域への書き込みを、データの優先度に応じて行うものである。

【００１０】

請求項２の発明によると、第２のメモリの利用効率を高くすることができる。

20

【００１１】

請求項３の発明は、請求項１記載の演算処理システムにおいて、前記第１のアクセスコントローラは、前記第１のメモリが、アクセスに所定のルーチンを必要とする種類のメモリ又はアドレスバスから受けたアドレスによってアクセスされる種類のメモリのいずれであっても、前記プロセッサによるアクセスができるように、前記第１のメモリに対するアクセスを制御するものである。

【００１２】

請求項３の発明によると、プロセッサは、第１のメモリの種類がいずれであっても、第１のメモリにアクセスすることができる。

【００１３】

30

請求項４の発明は、請求項３記載の演算処理システムにおいて、前記第１のアクセスコントローラは、前記第１のメモリが、アクセスに所定のルーチンを必要とする種類のメモリ又はアドレスバスから受けたアドレスによってアクセスされる種類のメモリのいずれであっても、前記プロセッサが同一のアドレス空間を用いてアクセスすることができるようにするものである。

【００１４】

請求項４の発明によると、プロセッサは、第１のメモリの種類を問わず、同様に第１のメモリにアクセスすることができる。このため、第１のメモリの種類の違いによって処理を分ける必要がない。

【００１５】

40

請求項５の発明は、請求項１記載の演算処理システムにおいて、前記データ転送部は、前記演算処理システムの起動時に、前記第１のメモリが格納するデータのうち所定の領域のデータを、自動的に前記第２のメモリに転送するものである。

【００１６】

請求項５の発明によると、演算処理システムの起動時に、データ転送部は、第１のメモリ内の所定の領域のデータを、自動的に第２のメモリに転送する。起動時に使用するデータ等を第２のメモリに転送しておくことにより、データ転送による待ち時間が発生することなく、演算処理システムを起動できる。

【００１７】

請求項６の発明は、請求項１記載の演算処理システムにおいて、前記データ管理部は、

50

前記プロセッサからの要求に応じて、前記領域の大きさを変化させるものである。

【0018】

請求項6の発明によると、データ管理部は、転送されたデータを格納するための領域の大きさを変化させる。このため、第2のメモリを効率的に使用できる。

【0019】

請求項7の発明は、請求項1記載の演算処理システムにおいて、前記データ転送部は、前記第1のメモリから前記第2のメモリに転送中のデータを格納するバッファメモリを備えるものであり、前記データ管理部は、前記プロセッサによるアクセスの対象が、前記バッファメモリが格納するデータと一致する場合には、前記プロセッサにそのデータを前記バッファメモリから読み出させるものである。

10

【0020】

請求項7の発明によると、データ転送部は、転送中のデータをバッファメモリに一時的に格納し、格納したデータにプロセッサがアクセスできる。このため、転送中のデータについては、転送が終わるのを待つことなく、データを読み出すことができる。

【0021】

請求項8の発明は、請求項1記載の演算処理システムにおいて、前記データ管理部は、前記プロセッサによってアクセスされたデータを格納する領域の次の領域のデータを、前記第1のメモリから前記第2のメモリにさらに転送させるものである。

【0022】

請求項8の発明によると、データ管理部は、アクセスされたデータを含む領域の次の領域のデータも第2のメモリに転送する。このため、プロセッサがある領域とその次の領域とにまたがるアクセスを行った場合には、転送のための待ち時間が発生しない。

20

【発明の効果】

【0023】

本発明によれば、第1のメモリから第2のメモリへのデータ転送を管理するデータ管理部を備えるので、プロセッサには負荷が掛からず、また、転送されて第2のメモリが格納するデータについては直接第2のメモリにアクセスすることができ、新たな転送による待ち時間が発生しない。

【発明を実施するための最良の形態】

【0024】

30

(第1の実施形態)

以下、本発明の第1の実施の形態について、図面を参照しながら説明する。

【0025】

図1は、本発明の第1の実施形態に係る演算処理システムの構成を示すブロック図である。図1の演算処理システムは、プロセッサ101と、プロセッサインタフェース102と、第1のアクセスコントローラとしてのフラッシュメモリインタフェース103と、第1のメモリとしてのフラッシュメモリ104と、データ転送部としてのDMA(direct memory access)コントローラ105と、データ管理部106と、第2のアクセスコントローラとしてのDRAMインタフェース107と、第2のメモリとしてのDRAM108とを備えている。

40

【0026】

フラッシュメモリ104は、例えば、NAND型フラッシュメモリであって、フラッシュメモリ104が格納するデータへのアクセスには所定のルーチンを必要とする。フラッシュメモリ104が格納するデータは、DRAM108に転送されてから、読み出される。フラッシュメモリインタフェース103は、所定の信号波形を用いてフラッシュメモリ104にアクセスする。

【0027】

また、フラッシュメモリ104は、アドレスバスを持たず、読み出しの際には、最初に論理的なアドレスが、フラッシュメモリ104内部に備えるアドレスレジスタに書き込まれる。その後、書き込まれた論理的なアドレスが示すデータからシーケンシャルに、デ

50

ータ読み出しが行われる構造となっている。フラッシュメモリインタフェース１０３は、プロセッサインタフェース１０２から出力されるアドレスを、論理的なアドレスに変換する。

【００２８】

プロセッサ１０１は、アドレスや、リード要求及びライト要求信号、ならびに実際にデータをやり取りするデータバスを備えたプロセッサバスを備えている。プロセッサ１０１は、このプロセッサバスを通じてプロセッサインタフェース１０２と接続されている。プロセッサ１０１は、フラッシュメモリ１０４に対するアクセスをプロセッサインタフェース１０２に出力する。

【００２９】

プロセッサインタフェース１０２は、プロセッサバスにより、プロセッサ１０１がアクセスしようとする対象を判断し、フラッシュメモリインタフェース１０３、データ管理部１０６、ＤＲＡＭインタフェース１０７の中から適切な対象を選択してアクセスする。プロセッサインタフェース１０２は、プロセッサ１０１からフラッシュメモリ１０４に対するアクセスがあったことを通知する信号及びアクセス対象のアドレスを、データ管理部１０６に出力する。

【００３０】

データ管理部１０６は、フラッシュメモリ１０４のどの領域のデータがＤＲＡＭ１０８に転送されているかを示す情報（領域情報）を保持するタグメモリ１１０を備える。データ管理部１０６は、プロセッサインタフェース１０２から受け取ったアドレスと、タグメモリ１１０が保持する内容とに基づいて、プロセッサ１０１によるアクセスの対象のデータを含む領域のデータを、フラッシュメモリ１０４からＤＲＡＭ１０８に転送するよう制御する信号を、ＤＭＡコントローラ１０５に出力する。データ管理部１０６は、データの転送が完了したことを、プロセッサインタフェース１０２に通知する。データ管理部１０６は、転送が完了する毎にタグメモリ１１０の内容を更新する。

【００３１】

ＤＲＡＭ１０８の中にフラッシュメモリ１０４のどの領域のデータが格納されているかを判別するために、タグメモリ１１０には、ＤＲＡＭ１０８に格納されているフラッシュメモリ１０４の領域の先頭アドレスが、格納される。データ管理部１０６は、プロセッサインタフェース１０２から通知されるアドレスと、タグメモリ１１０が保持するフラッシュメモリ１０４の領域の先頭アドレスとを比較することにより、プロセッサ１０１がアクセス要求を行っているフラッシュメモリ１０４の領域のデータが、ＤＲＡＭ１０８に既に格納されているか否かを、判別する。

【００３２】

タグメモリ１１０はまた、領域を示すアドレスとともに、その領域のデータの優先度を格納する。優先度は、例えば、プロセッサにおけるキャッシュメモリに採用されるアルゴリズムと同等のアルゴリズムを適用して求めても良いし、独自のアルゴリズムを適用しても良い。

【００３３】

データ管理部１０６は、プロセッサ１０１からの新たなアクセス要求によりデータを転送する場合に、ＤＲＡＭ１０８に格納されたデータの領域のうち、優先度の最も低いデータの領域に上書きし、優先度の高いものはそのまま保存するように制御する。

【００３４】

なお、タグメモリ１１０を拡張して何らかの追加情報を書き加え、最もアクセスされていない領域を上書きされるべき領域であると判断するようにしても良いし、上書きされるべき領域を単にランダムに選ぶようにしても良い。

【００３５】

なお、転送されたデータが格納されるＤＲＡＭ１０８の領域については、元のデータを退避させるために、フラッシュメモリ１０４の対応する領域に書き戻すようにしても良いし、フラッシュメモリ１０４の読み出しのみを本発明の適用対象として、単純に上書きす

10

20

30

40

50

るだけでも良い。

【0036】

DMAコントローラ105は、データ管理部106から受けた信号に従って、フラッシュメモリ104のデータをフラッシュメモリインタフェース103から受け取り、DRAMインタフェース107に転送する。

【0037】

フラッシュメモリインタフェース103は、プロセッサインタフェース102及びDMAコントローラ105から、フラッシュメモリ104に対するアクセス要求を受け付けて、フラッシュメモリ104にアクセスする。

【0038】

DRAMインタフェース107は、所定の信号波形を用いてDRAM108にアクセスする。DRAMインタフェース107は、プロセッサインタフェース102及びDMAコントローラ105から、DRAM108に対するアクセス要求を受け付けて、DRAM108にアクセスする。

【0039】

図2は、フラッシュメモリ104が格納するデータを、DRAM108に転送する処理を示す説明図である。図2(a)は、フラッシュメモリ104の論理的なメモリマップを示す説明図である。図2(b)は、タグメモリ110の論理的なメモリマップを示す説明図である。図2(c)は、DRAM108の論理的なメモリマップを示す説明図である。

【0040】

図2(a)では、フラッシュメモリ104の全領域を、例として、領域A～Pの16領域に分けている。これらの領域は、論理的なアドレス空間に基づいて決定されており、それぞれのサイズは等しい。

【0041】

図2(c)では、DRAM108の全領域のうち、フラッシュメモリ104から転送されたデータを格納する領域として、例として、3つの領域が割り当てられている。その各領域のサイズは、フラッシュメモリ104の1つの領域と同一である。なお、DRAM108のその他の領域は、他の用途に用いることが可能である。

【0042】

タグメモリ110は、フラッシュメモリ104の領域のうち、データが転送されてDRAM108に格納されている領域を示す情報を保持する。領域A～Pのいずれか、又は、何も格納されていない状態を示す空白が保持されていることになる。図2(b)では、例えば、領域A, N, Gの3つの領域のデータがフラッシュメモリ104から転送されて、DRAM108に格納されていることが、表されている。

【0043】

図3は、プロセッサ101がアクセスを要求した場合のデータ管理部106の動作を示すフローチャートである。処理が開始されると、ステップS102では、データ管理部106は、プロセッサ101によるフラッシュメモリ104に対するアクセスを通知する信号を、プロセッサインタフェース102から受け取ると、アクセスがフラッシュメモリ104のどの領域を対象とするものかを判別する。

【0044】

次に、ステップS104では、データ管理部106は、タグメモリ110にアクセスの対象の領域が示されているか否かを判定する。プロセッサ101によるアクセスが対象とするフラッシュメモリ104の領域がタグメモリ110に示されている場合は、処理はステップS106に進み、示されていない場合は、処理はステップS108に進む。

【0045】

次に、ステップS106では、プロセッサ101は、プロセッサインタフェース102及びDRAMインタフェース107を介して、DRAM108にアクセスし、データを読み出す。

【0046】

10

20

30

40

50

ステップS 1 0 8では、データ管理部 1 0 6は、タグメモリ 1 1 0を参照して、最も優先度の低いデータを判断して、D R A M 1 0 8において上書きする領域を決定する。

【 0 0 4 7 】

ステップS 1 1 0では、データ管理部 1 0 6は、フラッシュメモリ 1 0 4からアクセス対象のデータを転送して、ステップS 1 0 8で決定したD R A M 1 0 8の優先度の低いデータの領域に格納するように、D M Aコントローラ 1 0 5を制御する。

【 0 0 4 8 】

ステップS 1 1 2では、データ管理部 1 0 6は、プロセッサインタフェース 1 0 2に、D R A M 1 0 8に転送が完了したことを通知する。

【 0 0 4 9 】

ステップS 1 1 4では、プロセッサ 1 0 1は、プロセッサインタフェース 1 0 2及びD R A Mインタフェース 1 0 7を介して、D R A M 1 0 8にアクセスし、データを読み出す。

【 0 0 5 0 】

ステップS 1 1 6では、データ管理部 1 0 6は、タグメモリ 1 1 0の内容を更新する。

【 0 0 5 1 】

次に、本実施形態に係る演算処理システムの起動時に、プロセッサ 1 0 1が、フラッシュメモリ 1 0 4から起動プログラムの読み出しを行う場合について説明する。

【 0 0 5 2 】

この場合、プロセッサ 1 0 1は、プロセッサインタフェース 1 0 2を介して、フラッシュメモリ 1 0 4が格納する起動アドレスに対してリードアクセスを行う。プロセッサインタフェース 1 0 2は、データ管理部 1 0 6に対して読み出し要求があったことを通知する。

【 0 0 5 3 】

演算処理システムの起動時であるため、タグメモリ 1 1 0には何も保持されていない。そのため、データ管理部 1 0 6は、フラッシュメモリ 1 0 4が格納する、プロセッサ 1 0 1によるアクセスの対象のデータが、D R A M 1 0 8に存在しないと判断する。よって、データ管理部 1 0 6は、プロセッサ 1 0 1によるアクセスの対象のデータ、すなわち、起動アドレスのデータを含む領域のデータを転送するように、D M Aコントローラ 1 0 5を制御する。

【 0 0 5 4 】

D M Aコントローラ 1 0 5は、プロセッサ 1 0 1によるアクセスの対象のデータを含む領域のデータを、フラッシュメモリ 1 0 4からD R A M 1 0 8に転送する。転送が完了すると、データ管理部 1 0 6は、内部に持つタグメモリ 1 1 0の内容を更新し、プロセッサインタフェース 1 0 2に、D R A M 1 0 8に該当データが存在することを通知する。

【 0 0 5 5 】

プロセッサ 1 0 1は、プロセッサインタフェース 1 0 2及びD R A Mインタフェース 1 0 7を介して、フラッシュメモリ 1 0 4から転送されたデータを、D R A M 1 0 8から読み出す。

【 0 0 5 6 】

次に、プロセッサ 1 0 1が起動アドレスに続く次のアドレスへのアクセスを行う場合について、説明する。この場合、起動アドレスのデータを含む領域のデータの転送が既に行われているため、タグメモリ 1 1 0に起動アドレスが保持されている。よって、データ管理部 1 0 6は、プロセッサ 1 0 1がアクセス要求するアドレスのデータが、既にD R A M 1 0 8に格納されていると判断して、D R A M 1 0 8に該当データが存在することを、プロセッサインタフェース 1 0 2に通知する。プロセッサ 1 0 1は、プロセッサインタフェース 1 0 2及びD R A Mインタフェース 1 0 7を介して、アクセス対象のデータを、D R A M 1 0 8から読み出す。

【 0 0 5 7 】

以上のように、データ管理部 1 0 6が、プロセッサ 1 0 1からのアクセスに従ってデー

10

20

30

40

50

タを転送し、転送したデータを管理することにより、プロセッサ101はアドレスや領域管理をする必要がなく、負荷が軽減される。フラッシュメモリインタフェース103は、システムの起動を行うためにSRAMなどのメモリを持つ必要がない。

【0058】

また、タグメモリ110が格納するデータに優先度を付して、その優先度に基づいてDRAM108の過去に利用した領域のいずれかを選択し、選択された領域に上書きして、転送したデータを格納することにより、DRAM108の転送に用いる領域を使い切ることではない。

【0059】

また、演算処理システムの起動時には、プロセッサ101のアクセスにより、起動アドレスのデータを含む領域のデータが転送され、起動アドレスに続く次のアドレスのデータについても転送されているため、プロセッサ101は、転送指示プログラム等を実行する必要がない。

【0060】

(第2の実施形態)

本発明の第2の実施形態に係る演算処理システムでは、フラッシュメモリインタフェース103は、フラッシュメモリ104の種類に合わせて、プロセッサ101からのアドレス指示を変換する。すなわち、フラッシュメモリ104が、データへのアクセスに所定のルーチンが必要なものだけでなく、アドレスバスを介した指示によりデータへのアクセスが可能なフラッシュメモリであっても、フラッシュメモリインタフェース103は、プロセッサ101が、いずれも同一のアドレス空間を用いてアクセスすることができるようにする。

【0061】

フラッシュメモリ104がアドレスバスを介した指示によりデータへのアクセスが可能な種類である場合は、フラッシュメモリインタフェース103は、プロセッサインタフェース102から受け取ったアドレスを、そのままフラッシュメモリ104に出力する。

【0062】

また、本実施形態に係る演算処理システムは、図1の第1の実施形態に係る演算処理システムにおいて、DMAコントローラ105に代えてDMAコントローラ205を備える。DMAコントローラ205は、DRAM領域設定レジスタを有する。プロセッサ101からの制御でDRAM領域設定レジスタを用いることにより、DRAM108の領域のうち、フラッシュメモリ104からの転送のために使用する領域の大きさを、設定することができる。転送のために使用する領域の大きさは、フラッシュメモリ104の領域に基づいた単位で、増減させることが可能である。

【0063】

図4は、第2の実施形態において、フラッシュメモリ104が格納するデータを、DRAM108に転送する動作を示す説明図である。図4(a)は、フラッシュメモリ104の論理的なメモリマップを示す説明図である。図4(b)は、タグメモリ110の論理的なメモリマップを示す説明図である。図4(c)は、DRAM108の論理的なメモリマップを示す説明図である。

【0064】

図4では、例えば、DRAM108の領域のうち最大で7つの領域を、フラッシュメモリ104からの転送のために使用する領域として、利用可能である。

【0065】

図4(a)では、フラッシュメモリ104の全領域を、例として、領域A~Pの16領域に分けている。この区分は、論理的なアドレス空間を元に決定されており、それぞれのデータサイズは一定である。

【0066】

図4(c)では、DRAM108の全領域のうち、フラッシュメモリ104から転送されたデータを格納する領域として、例として、5つの領域が設定されている。その一つ

10

20

30

40

50

つの領域のデータサイズは、フラッシュメモリ 104 の領域と同一である。

【0067】

タグメモリ 110 は、フラッシュメモリ 104 の領域のうち、データが転送されて DRAM 108 に格納されている領域を示す情報を保持する。領域 A ~ P のいずれか、又は、何も格納されていない状態を示す空白が保持されていることになる。

【0068】

図 4 (b) では、例として、タグメモリ 110 は、最大で 7 つの領域の情報を保持できるが、DRAM 108 の領域は転送のために 5 つだけ使用されており、タグメモリ 110 が保持する情報によると、2 つの領域が未使用となっている。図 4 (b) では、例えば、領域 A, N, G, H, I の 5 つの領域のデータがフラッシュメモリ 104 から転送されて、DRAM 108 に格納されていることが、表されている。

10

【0069】

また、本実施形態においては、データ管理部 106 は、本実施形態に係る演算処理システムの起動時に、プロセッサ 101 からのアクセスを受けることなく、所定の領域のデータをフラッシュメモリ 104 から DRAM 108 に、自動的に転送する。

【0070】

図 5 は、第 1 の実施形態に係る演算処理システムの起動時における処理の流れを示すフローチャートである。図 5 は、例えば、メモリマップが図 2 (a) と同様の状態であるフラッシュメモリ 104 から、演算処理システムの起動時においてデータを読み出す場合を示す。

20

【0071】

処理が開始されると、ステップ S122 では、プロセッサ 101 は、フラッシュメモリ 104 が格納する図 2 (a) の領域 A のデータに対してアクセスを要求する。次に、ステップ S124 では、データ管理部 106 は、領域 A のデータをフラッシュメモリ 104 から DRAM 108 に転送するように、DMA コントローラ 105 を制御する。転送が完了すると、ステップ S126 では、プロセッサ 101 は、プロセッサインタフェース 102 及び DRAM インタフェース 107 を介して、DRAM 108 にアクセスし、領域 A のデータを読み出す。

【0072】

続いて、ステップ S128 では、プロセッサ 101 は、フラッシュメモリ 104 が格納する図 2 (a) の領域 B のデータに対してアクセスを要求する。次に、ステップ S130 では、データ管理部 106 は、領域 B のデータをフラッシュメモリ 104 から DRAM 108 に転送するように、DMA コントローラ 105 を制御する。転送が完了すると、ステップ S132 では、プロセッサ 101 は、プロセッサインタフェース 102 及び DRAM インタフェース 107 を介して、DRAM 108 にアクセスし、領域 B のデータを読み出す。

30

【0073】

よって、演算処理システムの起動時において、プロセッサ 101 が、フラッシュメモリ 104 の領域をまたぐアクセスを行い、かつ、データ管理部 106 が、図 5 のようにプロセッサ 101 からのアクセスを受けてからデータの転送を行う場合は、図 2 (a) の領域 B のデータを、フラッシュメモリ 104 から DRAM 108 に転送する間の待ち時間が発生する。

40

【0074】

図 6 は、第 2 の実施形態に係る演算処理システムの起動時における処理の流れを示すフローチャートである。図 6 は、例えば、メモリマップが図 4 (a) と同様の状態であるフラッシュメモリ 104 から、演算処理システムの起動時においてデータを読み出す場合を示す。

【0075】

処理が開始されると、ステップ S142 では、プロセッサ 101 は、フラッシュメモリ 104 が格納する図 4 (a) の領域 A に対してアクセスを要求する。次に、ステップ S1

50

44では、データ管理部106は、領域Aのデータをフラッシュメモリ104からDRAM108に転送するように、DMAコントローラ205を制御する。転送が完了すると、ステップS146では、プロセッサ101は、プロセッサインタフェース102及びDRAMインタフェース107を介して、DRAM108にアクセスし、領域Aのデータを読み出す。

【0076】

また、ステップS148からステップS150までは、ステップS146と並列に処理される。ステップS148では、データ管理部106は、図4(a)の領域Bのデータをフラッシュメモリ104からDRAM108に転送するように、DMAコントローラ205を制御する。ステップS150では、データ管理部106は、図4(a)の領域Cのデータをフラッシュメモリ104からDRAM108に転送するように、DMAコントローラ205を制御する。

【0077】

ステップS152では、プロセッサ101は、領域Bに対してアクセスを要求する。ステップS154では、プロセッサ101は、プロセッサインタフェース102及びDRAMインタフェース107を介して、DRAM108にアクセスし、領域Bのデータを読み出す。

【0078】

図6のように処理されることにより、演算処理システムの起動時において、領域Bのデータは、プロセッサ101が領域AのデータをDRAM108から読み出す間に、自動的にフラッシュメモリ104からDRAM108に転送されるので、領域Bのデータを転送する間の待ち時間は、発生しない。

【0079】

以上のように、フラッシュメモリインタフェース103が、プロセッサ101からのアドレス指示をフラッシュメモリ104の種類に合わせて変換することにより、プロセッサ101は、フラッシュメモリ104の種類が異なるものであっても、同一のアドレス空間を用いてアクセスできる。よって、異なる種類のフラッシュメモリ104へのアクセスのために異なるソフトウェアを用いたり、フラッシュメモリの違いによるソフトウェア修正を実施する必要がない。

【0080】

また、DMAコントローラ205がDRAM領域設定レジスタを有して、DRAM108の領域のうち、フラッシュメモリ104からの転送のために使用する領域の大きさを増減することにより、システムの状態に応じて、DRAM108を活用できる。

【0081】

さらに、演算処理システムの起動時に、プロセッサ101からのアクセスを受けることなく、所定の領域のデータをフラッシュメモリ104からDRAM108に自動的に転送することにより、プロセッサ101がフラッシュメモリ104に対して領域をまたぐアクセスを実施しても、待ち時間は発生せず、システム起動時間が短縮される。

【0082】

(第3の実施形態)

本発明の第3の実施形態に係る演算処理システムは、図1の第1の実施形態に係る演算処理システムにおいて、DMAコントローラ105に代えてDMAコントローラ305を備える。

【0083】

図7は、DMAコントローラ305の構成を示すブロック図である。DMAコントローラ305は、DMAコントローラ105において、バッファメモリ702と、アドレスポインタ704とをさらに備える。バッファメモリ702には、図1のDRAM108よりも少ない消費電力で、より高速にアクセスできる記憶装置が用いられる。

【0084】

バッファメモリ702は、図1のフラッシュメモリ104からDRAM108に転送さ

れるデータを、一時的に格納する。アドレスポインタ704は、バッファメモリ702が格納するデータのフラッシュメモリ104におけるアドレスを格納し、フラッシュメモリ104からデータが転送される毎に値が更新される。

【0085】

図1のデータ管理部106は、バッファメモリ702及びアドレスポインタ704にアクセスし、アドレスポインタ704が格納する値を参照することにより、バッファメモリ702が格納するデータの、フラッシュメモリ104におけるアドレスを判別する。データ管理部106は、図1のプロセッサ101によるアクセスの対象のデータと、バッファメモリ702が格納するデータとが一致すれば、データがDRAM108に格納される前であっても、転送中にバッファ702からデータを読み出すことができる。

10

【0086】

また、本実施形態においては、データ管理部106は、プロセッサ101のアクセスによってデータがフラッシュメモリ104からDRAM108に転送される場合は、プロセッサ101が要求するデータとともに、次の領域のデータを転送する。

【0087】

図8は、第3の実施形態において、プロセッサ101がアクセスを要求した場合のデータ管理部106の動作を示すフローチャートである。

【0088】

処理が開始されると、ステップS162では、データ管理部106は、プロセッサ101によるフラッシュメモリ104に対するアクセスを通知する信号をプロセッサインタフェース102から受け取ると、アクセスがフラッシュメモリ104のどの領域を対象とするものかを判別する。

20

【0089】

次に、ステップS164では、データ管理部106は、アクセスの対象のデータがバッファメモリ702に格納されているか否かを判定する。プロセッサ101によるアクセスの対象のデータがバッファメモリ702に格納されている場合は、処理はステップS166に進み、格納されていない場合は、処理はステップS168に進む。

【0090】

ステップS166では、プロセッサ101は、プロセッサインタフェース102及びデータ管理部106を介して、バッファメモリ702からデータを読み出す。

30

【0091】

なお、ステップS168、S170、S172、S174、S176、S178及びS180では、図3のステップS104、S106、S108、S110、S112、S114及びS116と、それぞれ同様の処理が行われる。

【0092】

次に、ステップS182では、データ管理部106は、タグメモリ110を参照して、最も優先度の低い領域を判断して、DRAM108において上書きする領域を決定する。

【0093】

ステップS184では、プロセッサ101によるアクセスが対象とする領域の、次の領域のデータを、フラッシュメモリ104からDRAM108に転送する。ステップS186では、データ管理部106は、タグメモリ110の内容を更新する。

40

【0094】

以上のように、DMAコントローラ305がバッファメモリ702を備え、プロセッサ101によるアクセスの対象のデータと、バッファメモリ702が格納するデータとが一致した場合には、プロセッサ101がバッファメモリ702からデータを読み出すようにすることにより、DRAM108へのアクセスを減らすことができる。よって、データへのアクセスに要する時間を短縮し、かつ、消費電力を抑えることができる。

【0095】

また、データ管理部106が、アクセス対象のデータを転送する際に、そのデータを格納する領域の次の領域のデータを転送しておいても良い。すると、プロセッサ101が領

50

域をまたぐアクセスを実施した場合に、次の領域のデータを転送するための待ち時間が発生しない。

【産業上の利用可能性】

【0096】

以上説明したように、本発明は、プロセッサの負荷を増加させることなく、フラッシュメモリからの読み出しにおける待ち時間を短縮することができるので、NAND型フラッシュメモリを使用するシステム全般について有用である。

【図面の簡単な説明】

【0097】

【図1】本発明の第1の実施形態に係る演算処理システムの構成を示すブロック図である 10

【図2】フラッシュメモリ104が格納するデータを、DRAM108に転送する処理を示す説明図である。図2(a)は、フラッシュメモリ104の論理的なメモリマップを示す説明図である。図2(b)は、タグメモリ110の論理的なメモリマップを示す説明図である。図2(c)は、DRAM108の論理的なメモリマップを示す説明図である。

【図3】プロセッサ101がアクセスを要求した場合のデータ管理部106の動作を示すフローチャートである。

【図4】第2の実施形態において、フラッシュメモリ104が格納するデータを、DRAM108に転送する動作を示す説明図である。図4(a)は、フラッシュメモリ104の論理的なメモリマップを示す説明図である。図4(b)は、タグメモリ110の論理的なメモリマップを示す説明図である。図4(c)は、DRAM108の論理的なメモリマップを示す説明図である。 20

【図5】第1の実施形態に係る演算処理システムの起動時における処理の流れを示すフローチャートである。

【図6】第2の実施形態に係る演算処理システムの起動時における処理の流れを示すフローチャートである。

【図7】DMAコントローラ305の構成を示すブロック図である。

【図8】第3の実施形態において、プロセッサ101がアクセスを要求した場合のデータ管理部106の動作を示すフローチャートである。

【符号の説明】 30

【0098】

101 プロセッサ

102 プロセッサインタフェース

103 フラッシュメモリインタフェース(第1のアクセスコントローラ)

104 フラッシュメモリ(第1のメモリ)

105, 205, 305 DMAコントローラ(データ転送部)

106 データ管理部

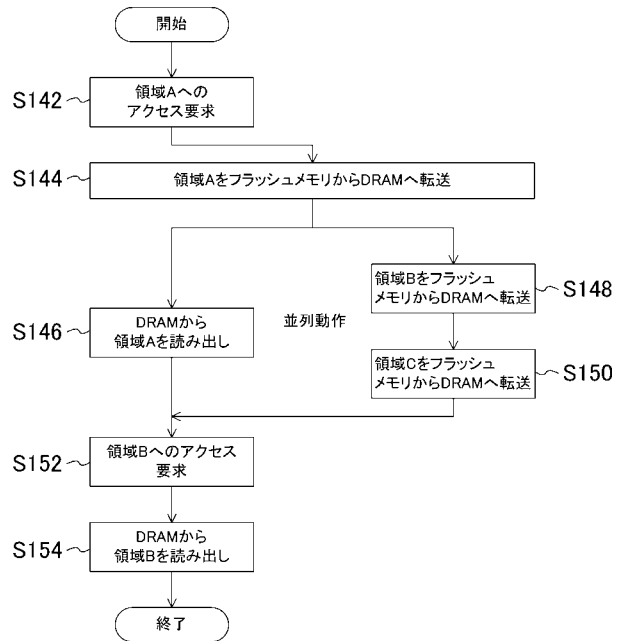
107 DRAMインタフェース(第2のアクセスコントローラ)

108 DRAM(第2のメモリ)

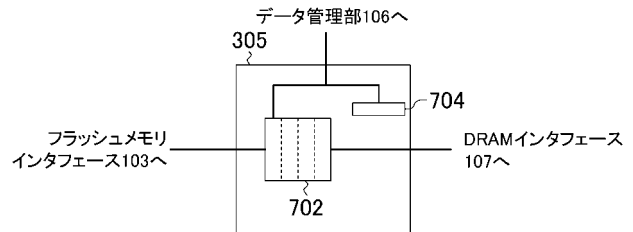
110 タグメモリ 40

702 バッファメモリ

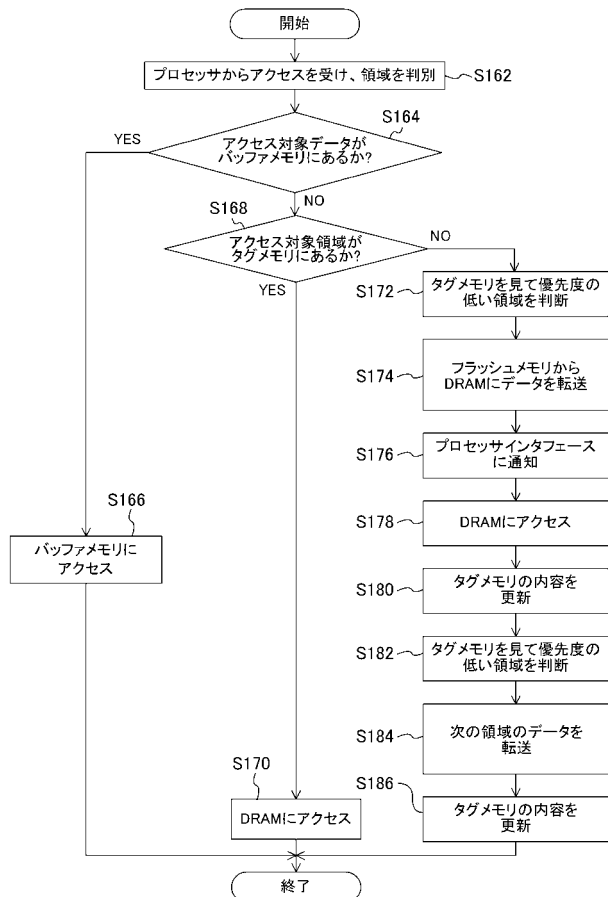
【図 6】



【図 7】



【図 8】



フロントページの続き

(74)代理人 100117581

弁理士 二宮 克也

(74)代理人 100117710

弁理士 原田 智雄

(74)代理人 100121728

弁理士 井関 勝守

(74)代理人 100124671

弁理士 関 啓

(74)代理人 100131060

弁理士 杉浦 靖也

(72)発明者 村山 謙太郎

大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

Fターム(参考) 5B060 CA17 CB01