(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2019-75101
(P2019-75101A)
(43)公開日　令和1年5月16日(2019.5.16)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 12/00	(2006.01)	G06F	12/00	560F	5B033
G06F 9/305	(2006.01)	G06F	12/00	580	5B160
G06F 9/302	(2006.01)	G06F	9/305	A	
		G06F	9/302	A	

審査請求 未請求　請求項の数 25　ＯＬ　（全 17 頁）

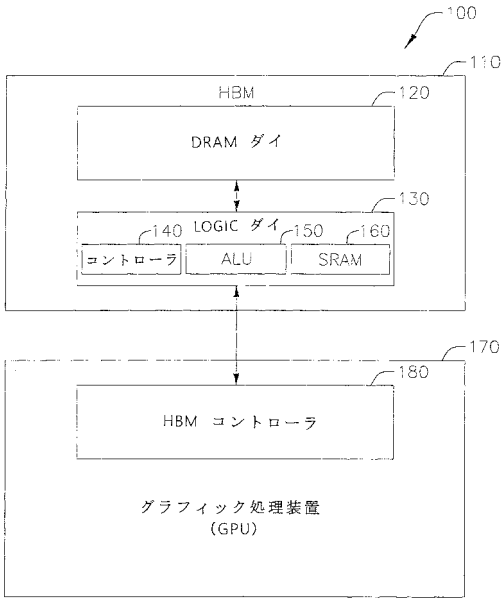
(21)出願番号　特願2018-173507 (P2018-173507)
(22)出願日　平成30年9月18日 (2018.9.18)
(31)優先権主張番号　62/573,390
(32)優先日　平成29年10月17日 (2017.10.17)
(33)優先権主張国　米国 (US)
(31)優先権主張番号　15/854,557
(32)優先日　平成29年12月26日 (2017.12.26)
(33)優先権主張国　米国 (US)

(71)出願人　390019839
三星電子株式会社
Ｓａｍｓｕｎｇ　Ｅｌｅｃｔｒｏｎｉｃｓ　Ｃｏ．，Ｌｔｄ．
大韓民国京畿道水原市霊通区三星路１２９　１２９，Ｓａｍｓｕｎｇ－ｒｏ，Ｙｅｏｎｇｔｏｎｇ－ｇｕ，Ｓｕｗｏｎ－ｓｉ，Ｇｙｅｏｎｇｇｉ－ｄｏ，Ｒｅｐｕｂｌｉｃ　ｏｆ　Ｋｏｒｅａ
(74)代理人　110000051
特許業務法人共生国際特許事務所
(72)発明者　張　牧　天
アメリカ合衆国，９５０５１　カリフォルニア州，サンタクララ，ビア　トリノ　プレイス　２９２０

最終頁に続く

(54)【発明の名称】インメモリのコマンド処理方法、これを適用する高帯域幅メモリ（ＨＢＭ）、及びＨＢＭシステム

(57)【要約】　（修正有）
【課題】高帯域幅メモリシステムでインメモリのコマンドを調整するためのシステム、及び方法を提供する。
【解決手段】高帯域幅メモリ（ＨＢＭ）システム１００において、グラフィック処理装置のＨＢＭコントローラ１８０によってファンクション・イン・メモリ命令をＨＢＭ１１０に送信する。ＨＢＭのロジック部分１３０は、ファンクション・イン・メモリ命令を受信し、ロジック部分が有するコントローラ１４０、ＡＬＵ１５０、及びＳＲＡＭ１６０を用いて命令の実行を調整する。
【選択図】図１



【特許請求の範囲】
【請求項１】
ＨＢＭ（Ｈｉｇｈ－Ｂａｎｄｗｉｄｔｈ　Ｍｅｍｏｒｙ）システムにおいてインメモリのコマンドを処理する方法であって、
グラフィック処理装置のＨＢＭコントローラによって、ファンクション・イン・メモリ命令をＨＢＭに送信する段階と、
コントローラ、ＡＬＵ（Ａｒｉｔｈｍｅｔｉｃ　Ｌｏｇｉｃ　Ｕｎｉｔ）、及びＳＲＡＭを含む前記ＨＢＭのロジック部分によって、前記ファンクション・イン・メモリ命令を受信する段階と、
前記コントローラ、前記ＡＬＵ、又は前記ＳＲＡＭのうちの少なくとも１つを用いる前記ロジック部分によって、前記ファンクション・イン・メモリ命令に基づいて、前記ファンクション・イン・メモリ命令の実行を調整する段階と、を有することを特徴とする方法。
【請求項２】
前記ファンクション・イン・メモリ命令の実行を調整する段階は、
前記コントローラによって、前記ファンクション・イン・メモリ命令を、演算及び少なくとも１つのデータ位置を含む計算ファンクション・イン・メモリ命令として識別する段階と、
前記コントローラによって、前記少なくとも１つのデータ位置に応じて、前記ＨＢＭのＤＲＡＭから少なくとも１つのデータを検索する段階と、
前記コントローラによって、前記少なくとも１つのデータ及び前記演算を前記ＡＬＵに提供する段階と、
前記ＡＬＵによって、前記少なくとも１つのデータに対して前記演算を実行する段階と、
前記演算の実行結果を前記ＤＲＡＭに格納する段階と、を有することを特徴とする請求項１に記載の方法。
【請求項３】
前記演算は、アトミック演算（ａｔｏｍｉｃ　ｏｐｅｒａｔｉｏｎ）及びデータ型を含み、
前記少なくとも１つのデータ位置は、宛先レジスタ、メモリアドレス、ソースレジスタ、定数、又は参照レジスタのうちの少なくとも１つを含むことを特徴とする請求項２に記載の方法。
【請求項４】
前記アトミック演算は、ＡＤＤ、ＳＵＢＴＲＡＣＴ、ＥＸＣＨＡＮＧＥ、ＭＡＸ、ＭＩＮ、ＩＮＣＲＥＭＥＮＴ、ＤＥＣＲＥＭＥＮＴ、ＣＯＭＰＡＲＥ－ＡＮＤ－ＳＷＡＰ、ＡＮＤ、ＯＲ、ＸＯＲ、又はＮＯＴの関数のうちの少なくとも１つを含むことを特徴とする請求項３に記載の方法。
【請求項５】
前記演算は、ＡＬＵ演算及びＤＲＡＭアクセス演算を含み、
前記少なくとも１つのデータ位置は、宛先レジスタ及び少なくとも１つのソースレジスタを含むことを特徴とする請求項２に記載の方法。
【請求項６】
前記ＤＲＡＭアクセス演算は、前記ＡＬＵ演算と対をなすロード命令又は格納命令を含むことを特徴とする請求項５に記載の方法。
【請求項７】
前記ファンクション・イン・メモリ命令の実行を調整する段階は、
前記コントローラによって、前記ファンクション・イン・メモリ命令をソースレジスタ及び宛先レジスタを含むファンクション・イン・メモリ移動命令として識別する段階と、
前記コントローラによって、前記ソースレジスタに応じて、前記ＨＢＭのＤＲＡＭから少なくとも１つのデータを検索する段階と、

前記コントローラによって、前記少なくとも１つのデータを前記宛先レジスタの前記ＤＲＡＭに格納する段階と、を有することを特徴とする請求項１に記載の方法。
【請求項８】
前記ファンクション・イン・メモリ命令の実行を調整する段階は、
前記コントローラによって、前記ファンクション・イン・メモリ命令をソースレジスタ及び宛先レジスタのうちの少なくとも１つを含むファンクション・イン・メモリスクラッチパッド命令として識別する段階と、
前記コントローラのタイミングパラメータを、ＤＲＡＭのタイミングパラメータからＳＲＡＭのタイミングパラメータに調整する段階と、
前記コントローラによって、前記ＳＲＡＭで前記ＳＲＡＭのタイミングパラメータに応じて、前記ファンクション・イン・メモリスクラッチパッド命令を実行する段階と、を有することを特徴とする請求項１に記載の方法。
【請求項９】
前記少なくとも１つのデータ位置がグラフィック処理装置のキャッシュを含む場合、前記ＨＢＭコントローラによって、前記グラフィック処理装置における前記ファンクション・イン・メモリ命令の実行を調整する段階を有することを特徴とする請求項１に記載の方法。
【請求項１０】
ＨＢＭ（Ｈｉｇｈ－Ｂａｎｄｗｉｄｔｈ　Ｍｅｍｏｒｙ）であって、
ＤＲＡＭと、
コントローラ、ＡＬＵ（Ａｒｉｔｈｍｅｔｉｃ　Ｌｏｇｉｃ　Ｕｎｉｔ）、及びＳＲＡＭを含んで命令を実行するロジック部分と、を備え、
前記命令は、前記ロジック部分によって実行されるとき、前記ロジック部分がファンクション・イン・メモリ命令に基づいて、前記ＤＲＡＭ、前記コントローラ、前記ＡＬＵ、又は前記ＳＲＡＭのうちの少なくとも１つを用いることによって、前記ファンクション・イン・メモリ命令の実行を調整することを特徴とするＨＢＭ。
【請求項１１】
前記ファンクション・イン・メモリ命令の実行調整は、
前記コントローラによって、前記ファンクション・イン・メモリ命令を演算及び少なくとも１つのデータ位置を含む計算ファンクション・イン・メモリ命令として識別し、
前記コントローラによって、前記少なくとも１つのデータ位置に応じて前記ＤＲＡＭから少なくとも１つのデータを検索し、
前記コントローラによって、前記少なくとも１つのデータ及び前記演算を前記ＡＬＵに提供し、
前記ＡＬＵによって、前記少なくとも１つのデータに対して前記演算を実行し、
前記演算の実行結果を前記ＤＲＡＭに格納することを特徴とする請求項１０に記載のＨＢＭ。
【請求項１２】
前記演算は、アトミック演算（ａｔｏｍｉｃ　ｏｐｅｒａｔｉｏｎ）及びデータ型を含み、
前記少なくとも１つのデータ位置は、宛先レジスタ、メモリアドレス、ソースレジスタ、定数、又は参照レジスタのうちの少なくとも１つを含むことを特徴とする請求項１１に記載のＨＢＭ。
【請求項１３】
前記アトミック演算は、ＡＤＤ、ＳＵＢＴＲＡＣＴ、ＥＸＣＨＡＮＧＥ、ＭＡＸ、ＭＩＮ、ＩＮＣＲＥＭＥＮＴ、ＤＥＣＲＥＭＥＮＴ、ＣＯＭＰＡＲＥ－ＡＮＤ－ＳＷＡＰ、ＡＮＤ、ＯＲ、ＸＯＲ、又はＮＯＴの関数のうちの少なくとも１つを含むことを特徴とする請求項１２に記載のＨＢＭ。
【請求項１４】
前記演算は、ＡＬＵ演算及びＤＲＡＭアクセス演算を含み、

前記少なくとも１つのデータ位置は、宛先レジスタ及び少なくとも１つのソースレジスを含むことを特徴とする請求項１１に記載のＨＢＭ。
【請求項１５】
前記ＤＲＡＭアクセス演算は、前記ＡＬＵ演算と対をなすロード命令又は格納命令を含むことを特徴とする請求項１４に記載のＨＢＭ。
【請求項１６】
前記ファンクション・イン・メモリ命令の実行調整は、
前記コントローラによって、前記ファンクション・イン・メモリ命令をソースレジスタ及び宛先レジスタを含むファンクション・イン・メモリ移動命令として識別し、
前記コントローラによって、前記ソースレジスタに応じて、前記ＤＲＡＭから少なくとも１つのデータを検索し、
前記コントローラによって、前記少なくとも１つのデータを前記宛先レジスタの前記ＤＲＡＭに格納することを特徴とする請求項１０に記載のＨＢＭ。
【請求項１７】
前記ファンクション・イン・メモリ命令の実行調整は、
前記コントローラによって、前記ファンクション・イン・メモリ命令をソースレジスタ及び宛先レジスタのうちの少なくとも１つを含むファンクション・イン・メモリスクラッチパッドの命令として識別し、
前記コントローラのタイミングパラメータをＤＲＡＭのタイミングパラメータからＳＲＡＭのタイミングパラメータに調整し、
前記コントローラによって、前記ＳＲＡＭで前記ＳＲＡＭのタイミングパラメータに応じて前記ファンクション・イン・メモリスクラッチパッド命令を実行することを特徴とする請求項１０に記載のＨＢＭ。
【請求項１８】
ＨＢＭ（Ｈｉｇｈ－Ｂａｎｄｗｉｄｔｈ　Ｍｅｍｏｒｙ）システムであって、
ＤＲＡＭを含むＤＲＡＭダイと、
コントローラ、ＡＬＵ（Ａｒｉｔｈｍｅｔｉｃ　Ｌｏｇｉｃ　Ｕｎｉｔ）、及びＳＲＡＭを含むロジックダイと、
を有するＨＢＭと、
前記ＨＢＭから分離して前記コントローラに命令を送信するグラフィック処理装置メモリコントローラと、を備え、
前記コントローラは、
前記命令を受信して、前記命令が一般命令である場合には、前記ＤＲＡＭダイの前記ＤＲＡＭに前記命令を伝達し、前記命令がファンクション・イン・メモリ命令である場合には、前記命令の実行を調整し、
前記命令の実行は、
前記コントローラ、前記ＡＬＵ、及び前記ＳＲＡＭのうちの少なくとも１つを用いて実行することを特徴とするＨＢＭシステム。
【請求項１９】
前記ファンクション・イン・メモリ命令の実行調整は、
前記コントローラによって、前記ファンクション・イン・メモリ命令を演算及び少なくとも１つのデータ位置を含む計算ファンクション・イン・メモリ命令として識別し、
前記コントローラによって、前記少なくとも１つのデータ位置に応じて前記ＤＲＡＭから少なくとも１つのデータを検索し、
前記コントローラによって、前記少なくとも１つのデータ及び前記演算を前記ＡＬＵに提供し、
前記ＡＬＵによって、前記少なくとも１つのデータに対して前記演算を実行し、
前記演算の実行結果を前記ＤＲＡＭに格納することを特徴とする請求項１８に記載のＨＢＭシステム。
【請求項２０】

前記演算は、アトミック演算（ａｔｏｍｉｃ　ｏｐｅｒａｔｉｏｎ）及びデータ型を含み、
前記少なくとも１つのデータ位置は、宛先レジスタ、メモリアドレス、ソースレジスタ、定数、又は参照レジスタのうちの少なくとも１つを含むことを特徴とする請求項１９に記載のＨＢＭシステム。
【請求項２１】
前記アトミック演算は、ＡＤＤ、ＳＵＢＴＲＡＣＴ、ＥＸＣＨＡＮＧＥ、ＭＡＸ、ＭＩＮ、ＩＮＣＲＥＭＥＮＴ、ＤＥＣＲＥＭＥＮＴ、ＣＯＭＰＡＲＥ－ＡＮＤ－ＳＷＡＰ、ＡＮＤ、ＯＲ、ＸＯＲ、又はＮＯＴの関数のうちの少なくとも１つを含むことを特徴とする請求項２０に記載のＨＢＭシステム。
【請求項２２】
前記演算は、ＡＬＵ演算及びＤＲＡＭアクセス演算を含み、
前記少なくとも１つのデータ位置は、宛先レジスタ及び少なくとも１つのソースレジスタを含むことを特徴とする請求項１９に記載のＨＢＭシステム。
【請求項２３】
前記ＤＲＡＭアクセス演算は、前記ＡＬＵ演算と対をなすロード命令又は格納命令を含むことを特徴とする請求項２２に記載のＨＢＭシステム。
【請求項２４】
前記ファンクション・イン・メモリ命令の実行調整は、
前記コントローラによって、前記ファンクション・イン・メモリ命令をソースレジスタ及び宛先レジスタを含むファンクション・イン・メモリ移動命令として識別し、
前記コントローラによって、前記ソースレジスタに応じて、前記ＤＲＡＭから少なくとも１つのデータを検索し、
前記コントローラによって、前記少なくとも１つのデータを前記宛先レジスタの前記ＤＲＡＭに格納することを特徴とする請求項１８に記載のＨＢＭシステム。
【請求項２５】
前記ファンクション・イン・メモリ命令の実行調整は、
前記コントローラによって、前記ファンクション・イン・メモリ命令をソースレジスタ及び宛先レジスタのうちの少なくとも１つを含むファンクション・イン・メモリスクラッチパッド命令として識別し、
前記コントローラのタイミングパラメータをＤＲＡＭのタイミングパラメータからＳＲＡＭのタイミングパラメータに調整し、
前記コントローラによって、前記ＳＲＡＭで前記ＳＲＡＭのタイミングパラメータに応じて前記ファンクション・イン・メモリスクラッチパッド命令を実行することを特徴とする請求項１８に記載のＨＢＭシステム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、メモリ帯域幅の管理に関し、より詳しくは、プロセッサ及び高帯域幅メモリ（ＨＢＭ：Ｈｉｇｈ－Ｂａｎｄｗｉｄｔｈ　Ｍｅｍｏｒｙ）のロジックダイ（Ｌｏｇｉｃ　Ｄｉｅ）上のメモリコントローラを有するＨＢＭシステム、及びＨＢＭの処理方法に関する。
【背景技術】
【０００２】
ＨＢＭは、主にグラフィック処理装置（ＧＰＵ）用高性能メモリとして用られる。ＨＢＭは、通常のＤＲＡＭに比べて非常に広いバスを有するという利点がある。現在のＨＢＭシステムの構造（ａｒｃｈｉｔｅｃｔｕｒｅ）は、ＨＢＭでバッファとして機能するロジックダイと、多重スタックＤＲＡＭダイ（例：ダイス（ｄｉｃｅ））と、グラフィック処理装置のＨＢＭコントローラと、を備える。メモリシステムにメモリ内の処理（例：インメモリプロセシング）機能を追加することによって、さらに性能が向上するが、ＨＢＭの

変更による既存のグラフィック処理装置の環境変化は最小限にすべきである。
【０００３】
なお、上述の内容は、本発明の実施形態における背景技術の理解を助けるためのものであって、先行技術を構成しない技術内容を含む。
【先行技術文献】
【特許文献】
【０００４】
【特許文献１】特開２００３－０１５８２４号公報
【発明の概要】
【発明が解決しようとする課題】
【０００５】
本発明は、上記従来技術に鑑みてなされたものであって、本発明の目的は、メモリシステムにファンクション・イン・メモリを導入して性能を向上しつつも、既存のグラフィック処理装置の環境変化を最小限にすることができるインメモリのコマンド処理方法と、このためのＨＢＭ、及びＨＢＭシステムを提供することにある。
【課題を解決するための手段】
【０００６】
本発明は、高帯域幅メモリシステムでインメモリのコマンドを調整するためのシステム、及び方法を提供する。
【０００７】
上記目的を達成するためになされた本発明の一態様による高帯域幅メモリ（ＨＢＭ）システムにおいてインメモリ（Ｉｎ－ｍｅｍｏｒｙ）のコマンドを処理するシステム、及び方法は、グラフィック処理装置のＨＢＭコントローラによって、ファンクション・イン・ＨＢＭ（Ｆｕｎｃｔｉｏｎ－ｉｎ－ＨＢＭ、以下「ファンクション・イン・メモリ」という）命令をＨＢＭに送信し、前記ＨＢＭのロジック部分（Ｌｏｇｉｃ　ｃｏｍｐｏｎｅｎｔ）で前記ファンクション・イン・メモリ命令を受信することを含む。前記ロジック部分は、コントローラ、ＡＬＵ（Ａｒｉｔｈｍｅｔｉｃ　Ｌｏｇｉｃ　Ｕｎｉｔ）、及びＳＲＡＭを含む。前記ロジック部分は、前記ファンクション・イン・メモリ命令に基づいて、前記コントローラ、前記ＡＬＵ、又は前記ＳＲＡＭのうちの少なくとも１つを用いることによって、前記ファンクション・イン・メモリ命令の実行を調整する。
【０００８】
前記ファンクション・イン・メモリ命令の実行調整は、前記ファンクション・イン・メモリ命令を、計算（ｃｏｍｐｕｔａｔｉｏｎａｌ）ファンクション・イン・メモリ命令として識別することを含む。前記計算ファンクション・イン・メモリ命令は、演算（ｏｐｅｒａｔｉｏｎ）及び少なくとも１つのデータ位置を含む。前記計算ファンクション・イン・メモリ命令の実行調整は、前記少なくとも１つのデータ位置に応じて、前記ＨＢＭのＤＲＡＭから少なくとも１つのデータを検索し、前記少なくとも１つのデータ及び前記演算を前記ＡＬＵに提供し、前記ＡＬＵによって前記少なくとも１つのデータに対して前記演算を実行し、前記演算の実行結果をＤＲＡＭに格納することを有する。
【０００９】
前記演算は、アトミック演算（ａｔｏｍｉｃ　ｏｐｅｒａｔｉｏｎ）及びデータ型であり、前記少なくとも１つのデータ位置は、宛先レジスタ、メモリアドレス、ソースレジスタ、定数、又は参照レジスタのうちの少なくとも１つを含む。
【００１０】
前記アトミック演算は、ＡＤＤ、ＳＵＢＴＲＡＣＴ、ＥＸＣＨＡＮＧＥ、ＭＡＸ、ＭＩＮ、ＩＮＣＲＥＭＥＮＴ、ＤＥＣＲＥＭＥＮＴ、ＣＯＭＰＡＲＥ－ＡＮＤ－ＳＷＡＰ、ＡＮＤ、ＯＲ、ＸＯＲ、又はＮＯＴの関数のうちの少なくとも１つを含む。
【００１１】
前記演算は、ＡＬＵ演算及びＤＲＡＭアクセス演算であり、前記少なくとも１つのデータ位置は、宛先レジスタ及び少なくとも１つのソースレジスタを含む。

【００１２】
前記ＤＲＡＭアクセス演算は、前記ＡＬＵ演算と対をなすロード命令又は格納命令を含む。
【００１３】
前記ファンクション・イン・メモリ命令の実行調整は、前記コントローラによって前記ファンクション・イン・メモリ命令をファンクション・イン・メモリ移動命令として識別することを含む。前記ファンクション・イン・メモリ移動命令は、ソースレジスタと、宛先レジスタと、を含む。前記コントローラは、前記ソースレジスタに応じて、前記ＨＢＭのＤＲＡＭから前記少なくとも１つのデータを検索し、前記少なくとも１つのデータを前記宛先レジスタのＤＲＡＭに格納する。
【００１４】
前記ファンクション・イン・メモリ命令の実行調整は、前記ファンクション・イン・メモリ命令をファンクション・イン・メモリスクラッチパッド命令として識別することを含む。前記ファンクション・イン・メモリスクラッチパッド命令は、ソースレジスタ又は宛先レジスタのうちの少なくとも１つを含む。前記ファンクション・イン・メモリスクラッチパッド命令の実行調整は、前記コントローラのタイミングパラメータをＤＲＡＭのタイミングパラメータからＳＲＡＭのタイミングパラメータに調整し、前記ＳＲＡＭで前記ＳＲＡＭのタイミングパラメータに応じて、前記ファンクション・イン・メモリスクラッチパッド命令を実行することを含む。
【００１５】
前記ＨＢＭシステムにおいて、前記インメモリのコマンドを処理するシステム、及び方法は、前記少なくとも１つのデータ位置がグラフィック処理装置のキャッシュを含む場合、前記ＨＢＭコントローラによって前記グラフィック処理装置における前記ファンクション・イン・メモリ命令の実行を調整することを含む。
【発明の効果】
【００１６】
本発明によれば、既存のグラフィック処理装置の環境変化を最小限にすると共に、メモリシステムの性能を向上させるためにインメモリのコマンドを処理する。
【図面の簡単な説明】
【００１７】
【図１】本発明の一実施形態による高帯域幅メモリシステム構造のブロック図である。
【図２】本発明の一実施形態によるファンクション・イン・メモリの命令セットアーキテクチャからの命令を処理する演算を示すフローチャートである。
【発明を実施するための形態】
【００１８】
以下、本発明を実施するための形態の具体例を、図面を参照しながら詳細に説明する。
【００１９】
本発明の技術的思想の特徴、及びこれを達成するための方法は、実施形態に関する以下の具体的な説明及び図面を参照して容易に理解される。以下、図面を参照しながら、本発明の実施形態について詳細に説明する。図面全般に亘って同一の参照符号は、同一の構成要素を示す。なお、本発明は、多様な実施形態を有し、本明細書で説明する実施形態のみに限定されない。本明細書で説明する実施形態は、本発明を全て完全に開示する例として提供され、これを通じて本発明が属する技術分野の通常の知識を有する者に、本発明の態様及び特徴を十分に伝えられる。従って、本発明が属する技術分野の通常の技術者が本発明の態様及び特徴を完全に理解するために、不要なプロセス、構成、及び技法については説明を省略する。特に言及しない限り、図面及び詳細な説明全般に亘って同一の参照符号は、同一の構成要素を示し、同一の構成要素に関する重複説明は省略する。図面に示す構成要素、階層、及び領域は、明確性のために誇張することがある。
【００２０】
以下の説明では、様々な実施形態が完全に理解されるように、数多くの特定の細部説明

を記載する。しかし、このような特定の細部説明がなくても、様々な実施形態を実施し得ることは明らかである。別の例では、公知の構造、及び装置は、様々な実施形態が不要且つあいまいにならないようにブロック図の形態で示す。

【００２１】

何れかの要素、階層、領域、又は構成が、他の要素、階層、領域、又は構成に対して、「上に」、「連結された」、又は「結合された」と記載される場合、これは、直接、他の要素や階層上に、他の要素や階層に連結された、又は他の要素や階層に結合されたことを意味する、或いは１つ以上の媒介要素、媒介階層、媒介領域、又は媒介構成が存在することを意味する。但し、「直接連結された／直接結合された」との記載は、中間の構成なしで他の構成と直接連結されるか、又は結合される１つの構成を示す。一方、「間に」、「間にすぐ」、又は「隣接した」及び「すぐ隣」のように、構成間の関係を説明する他の表現も同様に解釈する。また、何れのかの要素又は階層が、二つの要素又は階層の間にあると記載する場合、これは、単に二つの要素又は階層の間にあることを意味するか、又は１つ以上の媒介要素又は階層が存在することを意味する。

【００２２】

本明細書で用いる用語は、単に特定の実施形態を説明するためのものであって、本発明を限定するものではない。本明細書で用いる単数形態の用語は、文脈上特に明らかな指示がない限り、複数形態の用語も含む。また、「含む」、「有する」、「備える」等の用語を本明細書で用いる場合、これは、記載する特徴、数字、段階、動作、要素、及び／又は構成の存在を明示するが、１つ以上の他の特徴、数字、段階、動作、要素、構成、及び／又はこれらの集合の存在や付加を排除するものではない。本明細書で用いる「及び／又は」という用語は、挙げられる１つ以上の関連項目の任意の組み合わせ及び全ての組み合わせを含む。

【００２３】

本明細書で用いる「実質的に」、「約」、「略」、及びこれに類似する用語は、程度（ｄｅｇｒｅｅ）を示す用語ではなく、近似（ａｐｐｒｏｘｉｍａｔｉｏｎ）を示す用語として用いるものであり、本発明が属する技術分野の通常の技術者に認識される測定値又は計算値に内在する偏差を説明するためのものである。本明細書で用いる「約」又は「略」は、当該測定及び特定の量の測定に関する誤差（即ち、測定システムの限界）を考慮し、本発明の技術分野に属する通常の技術者が決定した特定の値に対する許容可能な偏差範囲内の平均及び記載した値を含む。例えば、「約」は、１つ以上の標準偏差内、又は記載した値の±３０％、２０％、１０％、５％内を意味する。また、本発明の実施形態を説明する際に「し得る及び／又はできる」という表現を用いる場合、これは、「本発明の１つ以上の実施形態」を示す。本明細書で用いられる「用いる」及び「用いられる」という用語は、それぞれ「利用する」及び「利用される」という用語と同じ意味である。更に、「例示的な」という用語は、例示又は一例を指称する。

【００２４】

特定の実施形態が異なって実施される場合、詳細なプロセス順序は、説明している順序とは異なって実行され得る。例えば、連続して説明した二つのプロセスは、実質的に同時に、又は説明した順序とは逆に実行される。

【００２５】

実施形態及び／又は中間構造の概略図である図面を参照して、様々な実施形態を本明細書で説明する。図示した形状は、例えば、製造技術及び／又は許容誤差の結果により変わる。また、本明細書に開示している特定の構造的又は機能的説明は、単に本発明の技術的思想による実施形態を説明するために例示する。従って、本明細書に開示する実施形態は、説明している領域の特定形態に制限されるものではなく、例えば、製造過程で生じる偏差形状を含む。例えば、矩形で示した注入領域（ｉｍｐｌａｎｔｅｄ　ｒｅｇｉｏｎ）は、通常円形や湾曲した形態を有し、及び／又は注入領域から非注入領域（ｎｏｎ－ｉｍｐｌａｎｔｅｄ　ｒｅｇｉｏｎ）への二元的変化というよりもその境界部分で注入濃度の傾斜を有する。同様に、注入によって形成された埋込領域は、注入が起きる面と埋込領域と

の間の領域にいくらかの注入をもたらす。従って、図に示す領域は、事実上概略的なものであり、その形態は装置の領域の実際の形態を示したものでなく、示した形状に制限しようとするものでもない。

【００２６】

本明細書で説明する本発明の実施形態によると、電子／電気装置、及び／又は任意の他の関連装置や構成は、任意の適切なハードウェア、ファームウェア（例えば、特定用途向け集積回路）、ソフトウェア、又はソフトウェア、ファームウェア、及びハードウェアの適切な組み合わせを利用して具現される。例えば、このような装置の様々な構成要素は、１つの集積回路（ＩＣ）チップ又は個別のＩＣチップ上に形成される。また、このような装置の様々な構成要素は、フレキシブル印刷回路フィルム（ｆｌｅｘｉｂｌｅ ｐｒｉｎｔｅｄ ｃｉｒｃｕｉｔ ｆｉｌｍ）、テープキャリアパッケージ（ＴＣＰ：Ｔａｐｅ Ｃａｒｒｉｅｒ Ｐａｃｋａｇｅ）、印刷回路基板（ＰＣＢ：Ｐｒｉｎｔｅｄ Ｃｉｒｃｕｉｔ Ｂｏａｒｄ）上に具現されるか又は１つの基板（ｓｕｂｓｔｒａｔｅ）上に形成される。更に、このような装置の多様な構成要素は、１つ以上のコンピューティング装置の１つ以上のプロセッサで実行され、コンピュータプログラムの命令を実行して、本明細書に記載する多様な機能を行うための他のシステム構成要素と相互作用するプロセス又はスレッド（ｔｈｒｅａｄ）である。コンピュータプログラムの命令はメモリに格納され、メモリは、例えばランダムアクセスメモリ（ＲＡＭ：Ｒａｎｄｏｍ Ａｃｃｅｓｓ Ｍｅｍｏｒｙ）又はフラッシュメモリ（例：ＮＡＮＤフラッシュメモリ）装置などの標準メモリ装置を用いるコンピューティング装置で具現される。コンピュータプログラムの命令は、例えばＣＤ－ＲＯＭ、フラッシュドライブ等のような他の非一時的コンピュータ読み取り可能な媒体に格納される。また、本発明が属する技術分野の通常の技術者には、本発明の技術範囲を逸脱することなく、多様なコンピューティング装置の機能が、単一のコンピューティング装置に結合若しくは統合されるか、又は特定のコンピューティング装置の機能が１つ以上の他のコンピューティング装置に分散されることが認識される。

【００２７】

本明細書で用いる技術用語及び科学用語を含む全ての用語は、特に定義しない限り、本発明が属する技術分野の通常の知識を有する者が一般的に理解するものと同様の意味を有する。また、通常用いられる辞典に定義されているような用語は、関連技術及び／又は本明細書の文脈上の意味と一致すると解釈され、本明細書で明らかに定義しない限り、理想的又は過度に形式的な意味として解釈されない。

【００２８】

図１は、本発明の一実施形態による高帯域幅メモリシステム構造のブロック図である。

【００２９】

図１に示す本発明の実施形態は、ファンクション・イン・メモリ（Ｆｕｎｃｔｉｏｎ－ｉｎ－ＨＢＭ）のＨＢＭシステム１００、及びＨＢＭ用命令セットアーキテクチャ（ＩＳＡ：Ｉｎｓｔｒｕｃｔｉｏｎ Ｓｅｔ Ａｒｃｈｉｔｅｃｔｕｒｅ）の拡張のためのシステムを提供する。

【００３０】

ＨＢＭシステム１００は、ＨＢＭ１１０に統合される追加の計算リソースを支援する。例えば、本実施形態において、ＨＢＭシステム１００は、一部のデータの演算及び移動をインメモリ（ｉｎ－ｍｅｍｏｒｙ）で実行させ、大容量のスクラッチパッドを提供する。

【００３１】

ＨＢＭシステム１００は、グラフィック処理装置（ＧＰＵ）１７０に連結された少なくとも１つのＨＢＭ１１０を含む。本実施形態において、ＨＢＭ１１０は、ＤＲＡＭ１２０（例：１つ以上のＤＲＡＭダイ（ｄｉｅ））と、ロジック部分（ｌｏｇｉｃ ｃｏｍｐｏｎｅｎｔ）１３０（例：ロジックダイ（ｄｉｅ））と、を含む。ロジック部分１３０は、コントローラ１４０、ＡＬＵ１５０、及びＳＲＡＭ１６０を有し、グラフィック処理装置１７０は、ＨＢＭ１１０とインターフェースするためのＨＢＭコントローラ１８０を含む。

【００３２】
本発明の一実施形態によるコントローラ１４０は、グラフィック処理装置１７０からの命令の実行を調整する。命令は、一般命令とファンクション・イン・メモリ命令との両方を含む。例えば、一般命令（ファンクション・イン・メモリ命令ではなく従来のロードファンクション及び格納ファンクション）は、ＨＢＭコントローラ１８０によって送信され、コントローラ１４０で受信されて、既存の方法で実行される。
【００３３】
また、コントローラ１４０は、インメモリ・ファンクション（例：ファンクション・イン・メモリ命令）の実行を調整する。例えば、コントローラ１４０は、データ移動演算（例：ロード／格納の対命令）を実行する。一例として、コントローラ１４０は、本来複数の一般命令であったファンクション・イン・メモリ命令を実行する。例えば、コントローラ１４０は、ＡＬＵ１５０を利用する計算ファンクション・イン・メモリ命令（例：アトミック命令（Ａｔｏｍｉｃ　Ｉｎｓｔｒｕｃｔｉｏｎｓ）及びＡＬＵ命令）の実行を調整する。この場合、コントローラ１４０は、ＤＲＡＭ１２０からデータを検索して、処理のために該当データ（及びＡＬＵ演算）をＡＬＵ１５０に提供することによって、命令の実行を調整する。その結果は、ＤＲＡＭ１２０に格納されるか、又はグラフィック処理装置１７０に戻される。一例として、ファンクション・イン・メモリ命令は、ロード命令又は格納命令と対をなす１つ以上のＡＬＵ命令を含む。
【００３４】
他の実施形態として、コントローラ１４０はスクラッチパッドの読み取り命令及び書き込み命令の実行を調整する。以下では、このようなファンクション・イン・メモリの各類型について詳細に説明する。
【００３５】
本発明の一実施形態によるＡＬＵ１５０は、様々な計算の動作（例：単純な計算のコマンド）を実行する。一例として、ＡＬＵ１５０は、算術演算、ビット演算、シフト演算等を実行する３２ビットＡＬＵである。例えば、ＡＬＵ１５０は、ＡＤＤ、ＳＵＢＴＲＡＣＴ、ＥＸＣＨＡＮＧＥ、ＭＡＸ、ＭＩＮ、ＩＮＣＲＥＭＥＮＴ、ＤＥＣＲＥＭＥＮＴ、ＣＯＭＰＡＲＥ－ＡＮＤ－ＳＷＡＰ、ＡＮＤ、ＯＲ、及びＸＯＲの演算を実行する。ＡＬＵ１５０は、アトミック演算及び非アトミック演算に利用される。
【００３６】
一実施形態として、コントローラ１４０は演算を提供し、ＡＬＵ１５０へのデータ入力及びＡＬＵ１５０からＤＲＡＭ１２０へのデータ出力を管理する。また、他の実施形態として、ＡＬＵ１５０が直接ＤＲＡＭ１２０からデータを検索して、ＤＲＡＭ１２０にデータを格納する。さらに他の実施形態では、コントローラ１４０がＤＲＡＭ１２０からデータを検索して、ＤＲＡＭ１２０にデータを格納する役割を担う。
【００３７】
本発明の一実施形態によるＳＲＡＭ１６０は、低レイテンシのスクラッチパッドとして構成される。一実施形態として、ＳＲＡＭ１６０は、同一のコマンド／アドレス（ＣＡ：Ｃｏｍｍａｎｄ／Ａｄｄｒｅｓｓ）及びデータ（ＤＱ）インターフェースをＤＲＡＭ１２０と共有し、他の実施形態として、ＳＲＡＭ１６０は、固有のＣＡ及びＤＱインターフェースを有する。ＳＲＡＭ１６０は、ＤＲＡＭ１２０のアドレス範囲から区別される固有のアドレス範囲を含む。
【００３８】
コントローラ１４０は、入力される読み取り／書き込み命令のアドレスを用いて、要請がスクラッチパッド演算であるか否かを判定する。他の実施形態として、グラフィック処理装置１７０は、具体的に指定されたスクラッチパッド命令をコントローラ１４０に送信する。
【００３９】
ＳＲＡＭ１６０を利用するコマンドの場合、コントローラ１４０は、ＳＲＡＭ１６０のタイミングパラメータ（例：ＤＲＡＭ１２０のタイミングパラメータより速いか又は低い

レイテンシ）に応じて動作するように自身のタイミングパラメータを変える。ＳＲＡＭ１６０の利用は、ユーザ（例：プログラマ）によって指定され、ＳＲＡＭ１６０の空間は、実行時間中に割り当てられる。動作中のスクラッチパッドは、グラフィック処理装置のＬ１スクラッチパッドと同様に動作する（例：低レイテンシメモリを提供）。グラフィック処理装置のＬ１スクラッチパッドは通常小さい（コア当たり１５ｋＢ）ため、拡張されたＨＢＭスクラッチパッド（例：ＳＲＡＭ１６０）はＤＲＡＭ１２０を利用する場合よりも性能が向上する。

【００４０】

本発明の一実施形態によるファンクション・イン・メモリＩＳＡは、ＨＢＭ１１０で利用可能な追加のリソースを利用するために提供される。例えば、ファンクション・イン・メモリＩＳＡは、計算ファンクション・イン・メモリ命令（例：ファンクション・イン・メモリアトミック命令及びファンクション・イン・メモリＡＬＵ命令）と、データ移動ファンクション・イン・メモリ命令と、ファンクション・イン・メモリスクラッチパッド命令と、を含む演算を許容するために、既存の命令セットを拡張する。各々のファンクション・イン・メモリ命令は、命令をファンクション・イン・メモリ命令として識別するファンクション・イン・メモリの指定子と、ＨＢＭにより実行される１つ以上の演算と、データ位置（例：レジスタ、メモリ、提供された定数等）と、を含む。一実施形態として、ファンクション・イン・メモリ命令は、＜ｄｅｓｉｇｎａｔｏｒ＞.＜ｏｐｅｒａｔｉｏｎ＞.＜ｄａｔａ ｌｏｃａｔｉｏｎ １＞＜ｄａｔａ ｌｏｃａｔｉｏｎ ２＞の形式に配列される。

【００４１】

本実施形態において、命令は、ユーザ（例：プログラマ）、コンパイラ、又はグラフィック処理装置によって、ファンクション・イン・メモリ命令として指定される。例えば、一部のプログラミング言語において、ユーザは、演算を実行する位置（例：インメモリ、グラフィック処理装置、又は中央処理装置）、又は利用するメモリを指定する。他の例として、コンパイラは、ＨＢＭ１１０で実行するコマンドを識別し、グラフィック処理装置１７０よりもＨＢＭ１１０のファンクション・イン・メモリのコマンドを優先して実行する。さらに他の例として、グラフィック処理装置１７０は、ソース及び宛先メモリアドレスを分析して、ファンクション・イン・メモリ命令が有効か否かを判定する。例えば、グラフィック処理装置１７０は、メモリアドレスを分析して、メモリアドレスの少なくとも１つがグラフィック処理装置のキャッシュに位置するか、又はＨＢＭではない他のメモリに位置するかを判定し、これらの場合、一般命令を実行する（例：非インメモリ命令）。

【００４２】

図２は、ファンクション・イン・メモリの命令セットアーキテクチャ（ＦＩＭ ＩＳＡ）からの命令を処理する方法を示すフローチャートである。

【００４３】

本発明の一実施形態によるグラフィック処理装置１７０は、ファンクション・イン・メモリ（ＦＩＭ）命令をＨＢＭ１１０に送信する（Ｓ２００段階）。グラフィック処理装置１７０は、要請を処理してＨＢＭ１１０に送信し、任意の返還情報を処理するＨＢＭコントローラ１８０を含む。本実施形態において、ＨＢＭコントローラ１８０は、ファンクション・イン・メモリ命令に含まれるメモリアドレスの位置を推定することによって、ファンクション・イン・メモリ命令が適切か否かを検証する。例えば、アドレスが他のＨＢＭを示すか、又はアドレスの何れかがグラフィック処理装置１７０のキャッシュ、即ちローカルキャッシュ用である場合、ファンクション・イン・メモリ命令は不適切なものとなる。本実施形態において、ファンクション・イン・メモリ命令の検証は、ＨＢＭ１１０に命令が送信される前に、初期のグラフィック処理装置パイプライン（ＧＰＵ Ｐｉｐｅｌｉｎｅ）段階で行われる。例えば、本実施形態において、グラフィック処理装置のローカルキャッシュコントローラは、ファンクション・イン・メモリ命令が適切か否かを検証する。

【００４４】

ＨＢＭ１１０は、ロジック部分１３０でファンクション・イン・メモリ命令を受信する（Ｓ２１０段階）。コントローラ１４０は、命令を処理して実行を調整する。コントローラ１４０は、命令がファンクション・イン・メモリ命令なのか否かを検証し、その命令による演算を判定する（Ｓ２２０段階）。例えば、コントローラ１４０は、命令がＡＬＵ１５０を利用する演算命令なのか、移動命令なのか、それともスクラッチパッド命令なのかを判定する。次に、コントローラ１４０は、命令自体（例：移動命令）を完了することによって、又は必須ロジックハードウェア（例：ＡＬＵ１５０又はＳＲＡＭ１６０）を用いることによって、命令の実行を調整する（Ｓ２３０段階）。

【００４５】

本発明の一実施形態によるＩＳＡは計算命令を含む。本実施形態において、コントローラ１４０は計算命令を受信して、ＡＬＵ１５０での計算命令の実行を調整する。計算命令は、ファンクション・イン・メモリアトミック命令と、ファンクション・イン・メモリＡＬＵ命令と、を含む。

【００４６】

本実施形態において、ＨＢＭ１１０は、ファンクション・イン・メモリＩＳＡを用いてファンクション・イン・メモリアトミック命令を処理する。アトミック命令は、一般に３つの段階、即ち、メモリ位置からデータを読み取る段階と、データに対して関数（例：ＡＤＤ、ＳＵＢＴＲＡＣＴ等）を実行する段階と、結果データを再度所定のメモリ位置に書き込む段階と、に分類される。

【００４７】

ＨＢＭ１１０は、ファンクション・イン・メモリアトミック命令を受信すると、内部で上記３つの段階を全て実行する。従来のアトミック命令と比較すると、ファンクション・イン・メモリアトミック命令は、命令の実行を完了するために、グラフィック処理装置１７０によってＨＢＭ１１０に送信された追加情報を含む。例えば、グラフィック処理装置１７０は、アトミック演算、データ型、宛先レジスタ、ソースレジスタ、参照レジスタ（例：ＣＯＭＰＡＲＥ－ＡＮＤ－ＳＷＡＰ関数のための）、メモリアドレス、及び関数型を含むファンクション・イン・メモリアトミック命令を送信する。

【００４８】

本実施形態において、コントローラ１４０は、グラフィック処理装置のＨＢＭコントローラ１８０からファンクション・イン・メモリアトミック命令を受信する。コントローラ１４０は、命令を読み取り、ファンクション・イン・メモリの指定子を用いて読み取られた命令が、ファンクション・イン・メモリ命令か否かを判定する。コントローラ１４０は、該当命令がファンクション・イン・メモリアトミック命令か否かを判定するために演算を用いる。この演算は、該当関数がファンクション・イン・メモリアトミック命令という信号を送信する以外にも、ＡＬＵ１５０により実行される関数型（例：アトミックＡＤＤ、アトミックＣＯＭＰＡＲＥ－ＡＮＤ－ＳＷＡＰ、アトミックＯＲ等）と、演算が実行されるデータ型（例：符号のある３２ビット定数、符号のない３２ビット定数等）と、を示す。

【００４９】

次に、コントローラ１４０は、提供されたデータ位置からデータを読み取り、関数と共にＡＬＵ１５０にデータを提供する。ＡＬＵ１５０は、データに対して関数を実行し、その結果は元のデータ位置に格納される。

【００５０】

一例として、ファンクション・イン・メモリアトミック命令の一般フォーマットは、ｆｉｍ．ａｔｏｍ．＜ｆｕｎｃｔｉｏｎ＞．＜ｄａｔａ ｔｙｐｅ＞ ＜ｄｅｓｔｉｎａｔｉｏｎ ｒｅｇｉｓｔｅｒ＞ ＜ｍｅｍｏｒｙ ａｄｄｒｅｓｓ＞ ＜ｓｏｕｒｃｅ ｒｅｇｉｓｔｅｒ ｏｒ ｃｏｎｓｔａｎｔ＞ ＜ｒｅｆｅｒｅｎｃｅ ｒｅｇｉｓｔｅｒ＞ である。表１は、例示値を有する関数の例（又はファンクション・イン・メモリアトミック命令の例）の一部を示す。

【００５１】

【表１】

関数	ファンクション・イン・メモリ命令の例
General Format	fim.atom.<function><data location 1><data location 2>
ADD	fim.atom.add.u32 %r6, [%r13], −10
Exchange	fim.atom.exch.b32 %r7, [%r14], %r4
Find Max	fim.atom.max.s32 %r8, [%r15], %r4
Find Min	fim.atom.min.s32 %r9, [%r16], %r4
Increment	fim.atom.inc.u32 %r10, [%r17], 17
Decrement	fim.atom.dec.u32 %r11, [%r18], 137
Compare−and−Swap	fim.atom.cas.b32 %r13, [%r19], %r12, %r4
AND	fim.atom.and.b32 %r17, [%r110], %r16
OR	fim.atom.or.b32 %r19, [%r111], %r18
XOR	fim.atom.xor.b32 %r20, [%r112], %r4

【００５２】

本実施形態において、ファンクション・イン・メモリ命令ＩＳＡは、ファンクション・イン・メモリＡＬＵ命令のための命令を含む。グラフィック処理装置によって実行される通常の演算は、メモリから必要なデータを引き出し、結果データを格納することに伴うロード命令及び格納命令を必要とする。本実施形態において、ファンクション・イン・メモリＡＬＵ命令は、既存の関数とこれに伴うロード命令及び格納命令を単一のファンクション・イン・メモリＡＬＵ命令に圧縮する。一例として、ファンクション・イン・メモリＡＬＵ命令の一般フォーマットは、ファンクション・イン・メモリの指定子と、ＡＬＵ演算及びこれと対をなすロード演算／格納演算を含む演算と、少なくとも１つのデータ位置と、を含む。例えば、ファンクション・イン・メモリＡＬＵ命令は、ｆｉｍ．＜ｆｕｎｃｔｉｏｎ＞．＜ｌｏａｄ／ｓｔｏｒｅ＞ ＜ｄｅｓｔｉｎａｔｉｏｎ ｒｅｇｉｓｔｅｒ＞ ＜ｓｏｕｒｃｅ ｒｅｇｉｓｔｅｒ＞ ＜ｌｏａｄ ｒｅｇｉｓｔｅｒ＞の形式に配列される。例えば、ＡＬＵ命令は、表２（ファンクション・イン・メモリＡＬＵ ロード命令／格納命令の例）に示すように、ロード命令及び／又は格納命令と対をなす。

【００５３】

【表２】

非ファンクション・イン・メモリ命令	ファンクション・イン・メモリ命令の例
ld $r1, [$r2] add $r3, $r0, $r1	fim.add.ld $r3, $r0, [$r2]
add $r3, $r0, $r1 st [$r2], $r3	fim.add.st [$r2], $r0, $r1

【００５４】

一例として、ファンクション・イン・メモリＡＬＵ命令の演算は、ファンクション・イ

ン・メモリアトミック命令の演算と同様である。例えば、本実施形態において、コントローラ１４０は、グラフィック処理装置のＨＢＭコントローラ１８０からファンクション・イン・メモリＡＬＵ命令を受信する。コントローラ１４０は、命令を読み取り、ファンクション・イン・メモリの指定子を用いて読み取った命令がファンクション・イン・メモリ命令か否かを判定する。コントローラ１４０は、命令がファンクション・イン・メモリＡＬＵ命令か否かを判定するため、及びＡＬＵ１５０により実行される関数型（例：ＡＤＤ、ＥＸＣＨＡＮＧＥ、ＭＩＮ、ＭＡＸ、ＯＲ等）を判定するために演算を用いる。

【００５５】

次に、コントローラ１４０は、提供されたデータ位置からデータを読み取り、関数と共にＡＬＵ１５０にデータを提供する。ＡＬＵ１５０は、データに対して関数を実行し、その結果は元のデータ位置又は指示されたデータ位置に格納される。

【００５６】

本実施形態において、ＨＢＭ１１０及びファンクション・イン・メモリＩＳＡは、ファンクション・イン・メモリ移動命令のために構成される。移動命令は、対応する格納命令と対をなすロード命令として識別される。ロード及び格納のアドレスが同一のＨＢＭに位置すると、関数はインメモリで実行される。本実施形態において、対をなすロード及び格納の関数は、単一のファンクション・イン・メモリ移動命令に併合される。表３は、ロード命令／格納命令の例（又はファンクション・イン・メモリの移動命令の例）を示す。

【００５７】

【表３】

非ファンクション・イン・メモリ命令	ファンクション・イン・メモリ命令の例
ｌｄ ＄ｒ１，［＄ｒ２］ ｓｔ［＄ｒ３］，＄ｒ１	ｆｉｍ．ｍｏｖ［＄ｒ３］，［＄ｒ２］

【００５８】

コントローラ１４０は、ファンクション・イン・メモリ移動命令を受信すると、ファンクション・イン・メモリの指定子により、命令がファンクション・イン・メモリ命令であることを認識して、演算が移動演算であることを認識する。次に、コントローラ１４０は、ソースアドレスからデータを読み取り、宛先アドレスにデータを格納する。本実施形態において、コンパイラは、表３に示すように、単一のファンクション・イン・メモリ移動命令に併合されるロード命令及び格納命令の対を識別する。本実施形態において、グラフィック処理装置１７０は、移動命令をＨＢＭ１１０に送信する前に移動命令を分析し、ファンクション・イン・メモリ移動命令が適切か否かを判定する。例えば、グラフィック処理装置１７０（例：ＨＢＭコントローラ１８０）は、ソースメモリアドレス及び宛先メモリアドレスを分析し、アドレスの何れかがグラフィック処理装置のキャッシュに存在するか否かを判定する。この例において、グラフィック処理装置１７０は、命令を一般のロード命令と格納命令とに分離する。

【００５９】

本実施形態において、ＨＢＭ１１０及びファンクション・イン・メモリＩＳＡは、低レイテンシスクラッチパッドとしてＳＲＡＭ１６０を利用する。コントローラ１４０は、ファンクション・イン・メモリスクラッチパッド命令を識別する。上述のように、ＳＲＡＭ１６０は、ＤＲＡＭ１２０のアドレス範囲から区別された特定のアドレス範囲を含む。コントローラ１４０は、要請がＤＲＡＭ１２０メモリアドレス又はＳＲＡＭ１６０メモリアドレスに対応するか否かを識別する。本実施形態において、ＩＳＡは、特定のスクラッチパッドのコマンドを含む。例えば、ＩＳＡは、ファンクション・イン・メモリ読み取りスクラッチパッドのコマンド（例：ＦＩＭ．ＲＤ＿ＳＰ）と、ファンクション・イン・メモリ書き込みスクラッチパッドのコマンド（例：ＦＩＭ．ＷＲ＿ＳＰ）と、を含む。この例では、コントローラ１４０は入力コマンドのメモリアドレスを推定しない。

【００６０】

ＳＲＡＭ１６０は、ＤＲＡＭ１２０より低いレイテンシ（即ち、より速く）で動作する。従って、スクラッチパッドは、データのロード時間及び格納時間に対して非決定的にＨＢＭ１１０をレンダリングする。コントローラ１４０は、ＤＲＡＭメモリが関連関数を実行する場合、ＤＲＡＭ１２０のタイミングパラメータに応じて動作し、ＳＲＡＭメモリの関連関数（例：ファンクション・イン・メモリスクラッチパッド命令）を実行する間は、ＳＲＡＭ１６０のタイミングパラメータに応じて動作することによって、命令の実行を調整する。従って、ファンクション・イン・メモリスクラッチパッド命令を受信すると、コントローラ１４０はＳＲＡＭ１６０のタイミングパラメータに対応するように自身のタイミングパラメータを調整し、ファンクション・イン・メモリ読み取り／書き込みスクラッチパッドのコマンドを実行する。

【００６１】

本実施形態において、ユーザは低レイテンシスクラッチパッドとしてＳＲＡＭ１６０を利用するデータ構造を設定する。ユーザがスクラッチパッドを用いることによってデータ構造を設定すると、コンパイラは指定子（例：アセンブリのｌｌｓｐ）を含むように要請を変換し、その結果、グラフィック処理装置１７０はＨＢＭ１１０に空間を割り当てる。

【００６２】

以上、上述した本発明の実施形態は、高帯域幅メモリシステム及び命令セットアーキテクチャを提供する。

【００６３】

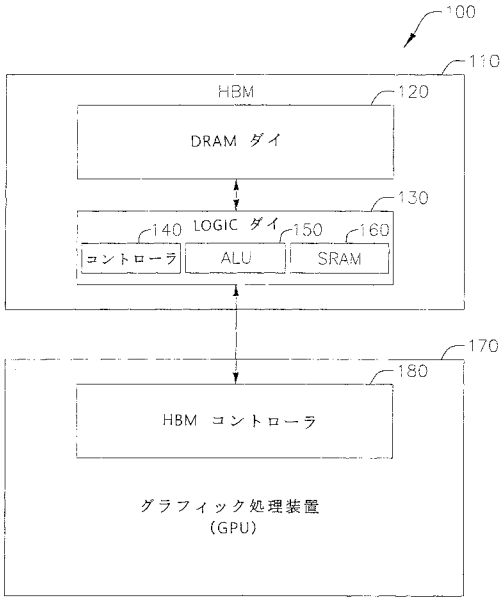
上述の説明内容は、本発明の一実施形態を説明するためのものであって、本発明を限定するものではない。本発明の技術分野に属する通常の技術者は、本発明の技術範囲から逸脱することなく、多様に変形実施することが可能である。

【符号の説明】

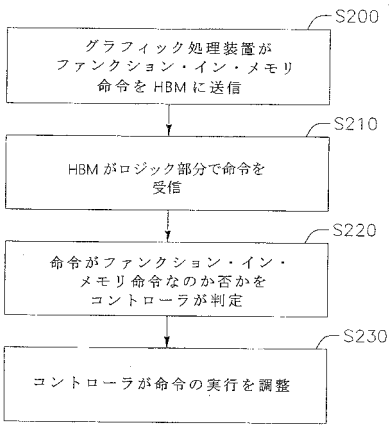
【００６４】

１００　ＨＢＭシステム
１１０　ＨＢＭ
１２０　ＤＲＡＭ
１３０　ロジック部分
１４０　コントローラ
１５０　ＡＬＵ
１６０　ＳＲＡＭ
１７０　グラフィック処理装置
１８０　ＨＢＭコントローラ

【図１】


100
110
HBM
120
DRAM ダイ
130
LOGIC ダイ
140
150
160
コントローラ
ALU
SRAM
170
180
HBM コントローラ
グラフィック処理装置
(GPU)

【図２】


S200
グラフィック処理装置が
ファンクション・イン・メモリ
命令を HBM に送信
S210
HBM がロジック部分で命令を
受信
S220
命令がファンクション・イン・
メモリ命令なのか否かを
コントローラが判定
S230
コントローラが命令の実行を調整

フロントページの続き

(72)発明者 マラディ, クリシャン テジャ
アメリカ合衆国, ９５１３５ カリフォルニア州, サンノゼ, ラウトレク ドライブ ４１９６
(72)発明者 牛 迪 民
アメリカ合衆国, ９４０８７ カリフォルニア州, サニーベール, ホルトハウス テラス ５２７
(72)発明者 チェン 宏 忠
アメリカ合衆国, ９５０３２ カリフォルニア州, ロスガトス, ユニット６ カールトン アベニュー １２０

Ｆターム(参考) 5B033 BD00
5B160 CB03 GA00