

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4271843号
(P4271843)

(45) 発行日 平成21年6月3日(2009.6.3)

(24) 登録日 平成21年3月6日(2009.3.6)

(51) Int. Cl.		F I			
G06F 1/24	(2006.01)	G06F 1/00	350A		
G06F 12/06	(2006.01)	G06F 12/06	521J		

請求項の数 4 (全 16 頁)

(21) 出願番号	特願2000-352194 (P2000-352194)	(73) 特許権者	504199127
(22) 出願日	平成12年11月20日(2000.11.20)		フリースケール セミコンダクター イン
(65) 公開番号	特開2001-209463 (P2001-209463A)		コーポレイテッド
(43) 公開日	平成13年8月3日(2001.8.3)		アメリカ合衆国 78735 テキサス州
審査請求日	平成19年10月11日(2007.10.11)		オースティン ウィリアム キャノン
(31) 優先権主張番号	09/447253		ドライブ ウェスト 6501
(32) 優先日	平成11年11月23日(1999.11.23)	(74) 代理人	100091915
(33) 優先権主張国	米国 (US)		弁理士 本城 雅則
		(74) 代理人	100099106
			弁理士 本城 吉子
		(74) 代理人	100116322
			弁理士 桑垣 衛

最終頁に続く

(54) 【発明の名称】 リセット後にデータ処理システムを構成するための方法および装置

(57) 【特許請求の範囲】

【請求項 1】

データ処理システムを構成する方法において、

リセット信号の解除に应答して、格納装置に対する第1バス・アクセスを開始する段階であって、前記第1バス・アクセスは前記リセット信号の解除後に他のどのバス・アクセスよりも先立って実行される、段階と、

前記格納装置から構成値を検索する段階と、

後続のバス・アクセスを実行する前に、前記構成値を用いて前記データ処理システムのバスのタイミング特性を構成する段階と、によって構成され、

前記バスのタイミング特性は、前記後続のバス・アクセスの間に使用される待機状態情報を含むことを特徴とする方法。 10

【請求項 2】

データ処理システムを構成する方法において、

リセット信号の解除に应答して、格納装置に対する第1バス・アクセスを開始する段階であって、前記第1バス・アクセスは前記リセット信号の解除後に他のどのバス・アクセスよりも先立って実行される、段階と、

前記格納装置から構成値を検索する段階と、

後続のバス・アクセスを実行する前に、前記構成値を用いて前記データ処理システムのエンディアン方式を構成する段階と、によって構成され、

前記エンディアン方式によって、前記データ処理システム中のリトル・エンディアン方 20

式およびビッグ・エンディアン方式の1つが選択されることを特徴とする方法。

【請求項3】

データ処理システムを構成する方法において、

リセット信号の解除に応答して、格納装置に対する第1バス・アクセスを開始する段階であって、前記第1バス・アクセスは前記リセット信号の解除後に他のどのバス・アクセスよりも先立って実行される、段階と、

前記格納装置から構成値を検索する段階と、

後続のバス・アクセスを実行する前に、前記構成値を用いて前記データ処理システムのデバッグ機能およびエミュレーション機能の少なくとも1つの実行可能化を判断する段階と、

によって構成されることを特徴とする方法。

【請求項4】

データ処理システムを構成する方法において、

リセット信号の解除に応答して、格納装置に対する第1バス・アクセスを開始する段階であって、前記第1バス・アクセスは前記リセット信号の解除後に他のどのバス・アクセスよりも先立って実行される、段階と、

前記格納装置から複数の構成ビットを検索する段階と、

後続のバス・アクセスを実行する前に、前記複数の構成ビットの少なくとも一部を用いて前記データ処理システムの少なくとも一部を構成する段階と、によって構成され、

前記複数の構成ビットの少なくとも一部は少なくとも1つの制御ビットおよび少なくとも1つのデータ・ビットからなり、前記少なくとも1つのデータ・ビットの使用は条件付であり、かつ前記少なくとも1つの制御ビットは前記少なくとも1つのデータ・ビットが使用されるかどうかを示す、

ことを特徴とする方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】

本発明は、データ・プロセッサに関し、さらに詳しくは、リセット後にデータ処理システムを構成するための方法および装置に関する。

【0002】

【従来の技術および発明が解決しようとする課題】

データ処理システムを既知の初期状態に置くために、大半のデータ処理システムには、システムをリセットするためのピンまたはその他の機構が装備される。処理システムがリセット条件から抜けると、リセット・ベクトル（リセット除外ベクトル）のアドレスがメモリ内の周知の位置から取り出される。このアドレスを用いて、リセット除外ルーチンを取り出し、実行する。リセット除外ルーチンには、通常、システムを初期化し、その後のプログラミング命令を実行するために準備を行ういくつかの命令が含まれる。

【0003】

32ビットのアドレス空間を備える処理システムにおいては、リセット・ベクトルは4個の8ビット・バイトで構築される32ビット・ワードが1個含まれる。リセット・ベクトルは、ビット幅が可変するメモリ構造から取り出される。たとえば、32ビット幅のメモリ構造、16ビット幅のメモリ構造または8ビット幅のメモリ構造を用いて、特定のシステムのリセット・ベクトルを格納することができる。リセット・ベクトルを格納するメモリ構造の種類により、リセット・ベクトルを構築する4個の8ビット・バイトを取り出すために必要なメモリ動作が決まる。

【0004】

リセット・ベクトルを格納するメモリのメモリ構成を決定する従来技術による方法の1つに処理システム上にピンを追加する方法があり、この場合は、追加されたピン上の信号がメモリ構成を示す。追加ピンは処理システムのコストを上げるので望ましくない。

【0005】

10

20

30

40

50

メモリ構成を決定する別の従来技術による方法には、処理システムのデータ・バスをサンプリングする方法がある。データ・バスのサンプリングから決定される値は、リセット・ベクトルが格納される特定のメモリ構成を示す。この方法によって処理システム内で異なるメモリ構造に対応することができるが、使用されているメモリ構成を示すために必要な特定の値にデータ・バスを駆動するためには外部の接着論理 (glue logic) が必要とされる。外部接着論理のために処理システム全体のコストが上がり、余分な寄生効果を招くために処理システムの機能には悪影響を及ぼし、さらに消費電力も増大する。

【0006】

リセット・ベクトルを格納するメモリの構成を決定する別の方法は、処理システム内に不揮発性格納手段を備えることである。不揮発性格納手段を用いて、特定のシステムに用いられるメモリ構成を表す値を格納することができる。この場合も、この方法により処理システム内に種々のビット幅のメモリ構造を用いることができるが、不揮発性格納手段を備えることにより、システムのコストが上がる。これは、電氣的消去可能プログラミング可能な読取専用メモリなどの不揮発性格納手段は複雑で実現するには費用がかかるためである。コストが上がるだけでなく、不揮発性格納手段は、再プログラミングすることが困難であることが多く、そのために異なるメモリ構成間の変更を行うことがさらに難しくなる。

10

【0007】

他の従来技術によるシステムには専用の外部メモリ構成を想定するものがあり、これは外部メモリをその想定に合わせることが必要である。図1は、CPUとメモリ・コントローラとを備えるプロセッサを示す。図1には、3つの異なるメモリ構成も図示される。すなわち、32ビット、16ビットおよび8ビットの構成である。メモリ・コントローラが異なる、可能性のあるメモリ構造の各々にアクセスするために用いるアドレス・データおよび制御信号も図示される。

20

【0008】

図1に示されるシステムで、16ビット・データ・バスを有するメモリ構造を用いてリセット・ベクトルを格納するとすると、アドレス 0×00 (16進数フォーマット) および 0×02 に対する2回の連続する読込動作が実行されて、32ビット・リセット・ベクトルを構築する4バイト (T, U, V, W) が検索される。これら2回のアクセスの実行は16ビット・メモリに関して行われるが、同じ想定を用いてリセット・ベクトルのアドレスをロードしようとする試みは、32ビットおよび8ビットのメモリ構造に関しては失敗する。

30

【0009】

8ビット・メモリ構造に関しては、2つのアドレスを用いる2回の読込動作のみが実行される。メモリ・コントローラは各読込動作に関してD[31:0]データ線上の16ビットのデータを受信することを予想するが、8ビット・メモリはD[31:24]データ線を駆動するに過ぎない。従って、リセット・ベクトル・アドレスを構築する4バイトのうち2バイトだけが8ビット・メモリから検索される。

【0010】

32ビット・メモリに関しては、アドレス線A[1]が32ビット・メモリにより無視される。そのため、 0×00 , 0×02 に対する2回の連続した読込を行うと、32ビット・メモリ内の位置 0×00 に格納される32ビット・ワードを重複して読み込むことになる。32ビット・メモリはリセット・ベクトル・アドレス全体をデータ線D[31:0]に駆動するので、メモリ・コントローラは、16ビットのデータのみをデータ線D[31:16]上の各メモリ・アクセスについて受信することになる。そのため、メモリ・コントローラは、メモリ動作の結果を変換して、完全な4バイト・アドレス (TUVW) の代わりにリセット・ベクトル・アドレスを構築する4つのバイトのうち2つを2回のコピーしたもの (VWVW) を含むアドレスに到達することになる。

40

【0011】

従って、追加ピン、不揮発性格納位置またはデータ・バスサンプリングを必要とせずに処

50

理システムが種々のビット幅のメモリ構造を利用して、リセットから抜けたときにリセット・ベクトルのアドレスを正確に決定することができるようにする方法および装置が必要である。また、リセット脱出直後に種々の処理システム・パラメータの構成を行って、後の動作が構成されたパラメータを利用できるようにする方法および装置を有することが望ましい。

【 0 0 1 2 】

【実施例】

一般に、本発明は種々のビット幅を持つメモリを用いて、リセット状態から抜け出したときにデータ処理システムを構成することを可能にする方法および装置を提供する。ある実施例においては、リセット・ベクトルを構築する4つのバイトをリセット・ベクトルを格納するメモリから、別々のメモリ動作により個別に取り出す。これら4つのバイトは、可能なメモリ構造の各々の中に所定の方法で格納されており、4つの所定のアドレスがデータ・バスの共通部分の上で4つの異なるバイトを検索するようになっている。検索されるリセット・ベクトルの部分は、データ処理システム内で種々のパラメータを構成するために用いられ、メモリやシステムのその他の機能的特性に関わる追加パラメータが初期化される。リセット・ベクトルに含まれる構成値には、データ部および制御部が含まれ、制御部によって、構成値のデータ部がデータ処理システム内の種々のパラメータにどのように適応されるかが決まる。

【 0 0 1 3 】

別の実施例においては、構成値はメモリからリセット・ベクトルを検索する前に取り出される。取り出される構成値には、メモリ構造のビット幅（データ・フィールド・サイズ）、メモリ構成に対応するアクセス時間に関わる情報（待機状態制御情報）や、メモリ構成内で用いられるバイト順序（リトルまたはビッグ・エンディアン（little or big endian）方式）などの情報を含む。そのため、検索される構成値を利用して、その後のメモリからのリセット・ベクトルの取り出しを制御するパラメータを構成するために利用することができるので、メモリ構造におけるリセット・ベクトルの格納を、利用されるメモリの種類に合致するよう最適化することができる。他の実施例においては、検索される構成値を用いて、データ処理システムに含まれるデバッグまたはエミュレーション機能を可能化/不能化するパラメータを構成することもできる。

【 0 0 1 4 】

本発明は、従来技術によるシステムにおいては不揮発性格納手段、追加の接着論理または追加のピンを加えずには不可能であったリセット・ベクトルの格納に関して、データ処理システムがメモリの可変構成を利用することができるようにする。リセット・ベクトルに加えて取り出される、あるいはそれに含まれる構成情報を用いて、処理システム内で追加のパラメータを初期化し、リセット脱出命令の初期の実行を、このようなパラメータを構成するためには複数の命令が必要であった従来技術による方法よりも効率的な方法で実行することができる。

【 0 0 1 5 】

本発明は、図2ないし図8を参照して、より良く理解することができる。図2は、データ処理システムと共に利用することのできるメモリ・マップ210を示す。メモリ・マップ210内の位置は、 $0 \times 00 \sim 0 \times F F F F F F F F$ と示されるアドレス・ビットを用いて定義される。リセット・ハンドラ・ルーチン220がメモリ・マップ210内に含まれ、処理システムがリセット状態を脱すると実行される。リセット・ハンドラ・ルーチン220を実行するためには、リセット・ベクトルと呼ばれるリセット・ハンドラ・ルーチン220のアドレスをメモリから取り出す。リセット・ベクトルは、 $0 \times T U V W$ の値を有するものとして示される。文字T, U, V, Wの各々は32ビットのリセット・ベクトルの8ビット・バイト部分を表す。

【 0 0 1 6 】

図示されるように、リセット・ハンドラ・ルーチン220は、ブート・メモリ装置240に格納されることが好ましい。リセット・ハンドラ・ルーチン220を実行するために取

10

20

30

40

50

り出さねばならないリセット・ベクトルもブート・メモリ装置 240 に格納するか、あるいは異なるメモリ装置に格納することができる。リセット・ハンドラ・ルーチン・アドレスを構築する 4 バイトの取出を容易にするために、4 バイトは種々のメモリ構成の各々の中で隣り合わないバイト位置に格納される。このため、リセット・ベクトルの隣接部分は、メモリ内では隣り合わないアドレスに格納され、少なくとも 1 ビットは隔てられる。好ましくは、メモリ内のリセット・ベクトルの部分の間隔は、処理システムのデータ・バスの幅の関数に基づく。リセット・ベクトルの部分の続きのアドレスは、好ましくは、第 1 (ベース) アドレスを、データ・バス内のバイト数に等しい増分値だけ増分することにより生成される。たとえば、データ・バスのバス幅が 32 ビット (4 バイト) であるとする

10

【0017】

本明細書で用いる「バス」という用語は、データ、アドレス、制御または状況情報など 1 種類以上の情報を転送するために用いられる複数の信号または導体を指すために用いられる。バスの導体または値のビット位置を示すためにカギ括弧を用いる。たとえば、D [31 : 24] は、データ・バスの最上位の 8 ビットを示すために用いられる。

【0018】

メモリ・アクセスのたびに 32 ビットのデータを戻す 32 ビット・メモリ 250 においては、リセット・ハンドラ・ルーチン・アドレスの 4 バイトの各々が、別々の 32 ビット・ワードの最上位バイトに含まれる。このため、リセット・ベクトルの最上位バイトである T 252 は、位置 0 x 00 にあるワードの最上位バイトである。そのため、アドレス 0 x 00 に対する第 1 回の読込動作が実行されると、値 T がデータ・バスの D [31 : 24] データ線上に現れる。その後 3 回の読込動作を行うと、後の読込動作においてアドレス 0 x 04, 0 x 08, 0 x 0C を用いてデータ・バスの同じ部分に値 U 254, V 256, W 258 を戻すことになる。

20

【0019】

32 ビット・メモリ 250 から値 T ~ W 252 ~ 258 を検索するのと同じメモリ動作を用いて、16 ビット・メモリ 260 から値 T ~ W 262 ~ 268 を検索することができる。16 ビット・メモリは、データ・バスの 16 ビットしか駆動しないので、特定のアドレスに関して検索される最上位データ・バイトをデータ・バスの最上位部分 (D [31 : 24]) に戻すような方法でデータ・バスに結合しなければならないことに留意されたい。このため、16 ビット・メモリ 260 から位置 0 x 00 を読み込むメモリ動作により、32 ビット・メモリ 250 を用いるアドレスに対する読込動作と同じデータ・バスの部分にデータ T 262 が戻ることになる。

30

【0020】

図 2 は、32 ビット・メモリ 250 または 16 ビット・メモリ 260 のいずれかに適応されるのと同じメモリ動作でリセット・ベクトルに関する適切な値を検索することのできる方法で、リセット・ベクトルを構築する 4 バイトを格納する 8 ビット・メモリ 270 をも示す。従って、アドレス 0 x 00, 0 x 04, 0 x 08, 0 x 0C にアクセスする 4 回のメモリ読込動作を 8 ビット・メモリ 270 に適応して、値 T ~ W 272 ~ 278 をそれぞれ戻すことができる。種々のメモリ構成の各々についてデータ・バスの同じ部分でデータを駆動することができるようにするためには、8 ビット・メモリ 270 を、データ・バスの最上位の 8 ビット部分 (D [31 : 24]) を駆動するよう構成すべきである。

40

【0021】

図 3 は、処理システム内で 1 つ以上のパラメータを構成するために取り出されるリセット・ベクトル値 300 に含まれる構成値 310 の利用法を示す。図示されるように、取り出されるリセット・ベクトル値 300 は、リセット・ハンドラ・ルーチンが格納されるアドレスを構築する 4 つのデータ・バイト T, U, V, W を含む。取り出されるリセット・ベクトル値 300 を構築するビットの部分構成値 310 として利用することによって、データ処理システムの構成パラメータは、リセット条件直後に取り出されるリセット・ベク

50

トル値 300 の初期検索の間に検索することができる。そのため、構成値 310 を利用して構成されるパラメータは、リセット脱出直後に構成され、それに続く処理システム動作が変更されたパラメータ値を利用することができるようにする。

【0022】

特定の方法で処理システム内の特定のパラメータを構成するために必要とされる構成値 310 に含まれるビットの設定により、取り出されるリセット・ベクトル値 300 が示すアドレスに変更が加わる。そのため、ブート・メモリ装置 240 は、1つ以上のリセット・ハンドラ・ルーチンを含めるよう構築される。これらのルーチンの各々は、構成値 310 内の特定の設定に基づき取り出されるリセット・ベクトル値 300 に関して得られる種々の値の1つに対応するアドレスに格納される。図2は、アドレス 0xT'U'V'W' に格納される、このような代替のリセット・ハンドラ・ルーチン 230 を示す。代替のリセット・ハンドラ・ルーチン 230 は、プロセッサを、実行される実際のルーチンを格納する別の位置に再指示するポインタに過ぎないことに留意されたい。従って、複数の構成値 310 が1つのシステム内に想定される場合は、複数のリセット・ベクトル・アドレスに対応する複数のハンドラ・ルーチンがブート・メモリ装置 240 に含まれることもある。

10

【0023】

構成値 310 内の異なる設定が複数のリセット・ハンドラ・ルーチンを（ポインタに過ぎなくとも）要求するので、構成値 310 は、取り出されたりセット・ベクトル値 300 のより下位の部分に含まれることが好ましい。そのため、必要とされる複数のリセット・ハンドラ・ルーチンが拡散するブート・メモリ装置 240 内のメモリ領域は小さくなる。取り出されるリセット・ベクトル 300 内のより上位のビットが構成値 310 に関して用いられる場合は、複数のルーチンがブート・メモリ装置 240 全体に拡散する。そのため、ブート・メモリ装置 240 を他の目的のために利用できる効率性は低くなる。

20

【0024】

構成値 310 は、処理システム内の特定のパラメータを構成する単独のビットである場合も、処理システム内の1つ以上のパラメータを構成する複数のビットである場合もある。構成値 310 に複数のビットが含まれる場合は、構成値 310 は、データ部と制御部とを備える。データ部は、特定のパラメータ位置に格納される実際のデータ値に対応して、これらのパラメータを適切な設定に構成する。制御部は、構成値 310 のデータ部またはその部分の応用を制御するために用い

30

ることができる。

【0025】

図3に示される例は、複数のデータ部分 312 ~ 318 を含むデータ部を示す。図3の例における構成値 310 の制御部は、パラメータ制御部分 320 を含んで図示される。図示されるように、データ部分 312 ~ 316 は、制御レジスタ 350 のパラメータ A 352 , B 354 , C 356 に適応される。

【0026】

制御レジスタ 350 は、CPU内、インタフェース・ブロック内またはデータ処理システム内に含まれる他の追加論理内に常駐する。制御レジスタ 350 は、データ処理システムがチップ選択信号を介してアクセスされるブロックとインタフェースする方法を制御するチップ選択制御レジスタである。ある例では、制御レジスタ 350 は、メモリ構造に対応するチップ選択レジスタである。このような例では、パラメータ A 352 が、チップ選択信号が選択したメモリ構造に対するアクセスのために利用される待機状態数に対応する。パラメータ B 354 は、処理システムがデッド・サイクルをメモリ動作に含むか否かを判断するデッド・サイクル・パラメータに対応する。最後に、パラメータ C 356 は、メモリ構造のデータ・バス幅に対応する。かくして、リセット状態から出るとすぐに特定のメモリ構造に関する正しいパラメータ設定がチップ選択レジスタ内に入れられ、後のメモリ動作が適切に構築される。

40

【0027】

別の実施例においては、制御レジスタ 350 は、デバッグまたはエミュレーション設定お

50

よび動作を制御する制御レジスタに対応する。そのため、パラメータ A ~ C 3 5 2 ~ 3 5 6 が、一定のデバッグまたはエミュレーション・モードが機能するか否かを判断する。たとえば、内部信号を外部からアクセス可能な信号上に見えるようにするショウ状態が、パラメータ A ~ C 3 5 2 ~ 3 5 6 により可能化または不能化される。その他のデバッグおよびエミュレーション機能には、性能監視，プログラム追跡，リセット状態脱出後のデバッグ・モードに対する直接エントリの可能化，1つ以上のモジュールに対するクロックの可能化/不能化，高速または低速クロック速度のリセット脱出可能化などが含まれる。

【 0 0 2 8 】

構成値 3 1 0 のデータ部分 3 1 8 は、制御レジスタ 3 5 0 の外にあるシステム制御パラメータ 3 6 2 に与えられる。同様に、構成値 3 1 0 のデータ部の異なる部分が、データ処理システム全体を通じて種々のパラメータに適應される。そのため、データ部分は、データ処理システムの広範囲に機能性に対応するパラメータを構成することができる。

10

【 0 0 2 9 】

前述の如く、図 3 の図ではパラメータ制御部分 3 2 0 を含む構成値 3 1 0 の制御部は、それぞれのパラメータに対する種々のデータ部分 3 1 2 ~ 3 1 8 の応用を制御するために用いられる。ある実施例では、パラメータ制御部分 3 2 0 は、データ部分 3 1 2 ~ 3 1 8 のいずれかがそれぞれのパラメータに適應されるか否かを示す単独のビットである。他の実施例では、データ部分 3 1 2 ， 3 1 4 が、パラメータ制御部分 3 2 0 の状態に関わらず、個々のパラメータに適應される。しかし、同じ実施例で、データ部分 3 1 6 ， 3 1 8 がパラメータ制御部分 3 2 0 により制御され、パラメータ制御部分 3 2 0 の状態に基づいて個々のパラメータに適應されたりされなかつたりする。

20

【 0 0 3 0 】

さらに別の実施例においては、パラメータ制御部分 3 2 0 は、構成値 3 1 0 内の特定のデータ部分に各々が関連する複数のビットを含む。かくして、各データ部分 3 1 2 ~ 3 1 8 は、特定のデータ部分がその個別のパラメータに適應されるか否かを決定するパラメータ制御部分 3 2 0 内の対応ビットを有することがある。実施例によっては、データ部分 3 1 2 ~ 3 1 8 の一部分だけが、パラメータ制御部分 3 2 0 内で対応するビットにより制御されることがある。

【 0 0 3 1 】

構成値 3 1 0 の制御部に関する別の可能な用途は、データ部分 3 1 2 ~ 3 1 8 の各々が適應される、あるいはされない個別パラメータを決定することである。かくして、構成値 3 1 0 の制御部は、特定のデータ部分が第 1 制御レジスタに適應されるか、あるいは同じ制御レジスタまたは他の制御レジスタ内に位置する第 2 パラメータに適應されるかを決定する。当業者には明白であろうが、データおよび制御部の様々な組み合わせを構成値 3 1 0 内で利用して、特定のデータ処理システム内で必要とされるパラメータ構成を実現することができる。

30

【 0 0 3 2 】

図 4 は、図示される種々のメモリ構成に結合する柔軟性を有するデータ処理システム 4 0 0 を示す。データ処理システム 4 0 0 は、プロセッサ 4 1 0 と、32 ビット・メモリ 4 4 0 ， 16 ビット・メモリ 4 5 0 および 8 ビット・メモリ 4 6 0 のうち少なくとも 1 つのメモリとを備える。プロセッサ 4 1 0 は、インタフェース回路構成（インタフェース・ブロック 4 1 6 ）に動作可能に結合される中央処理装置（CPU 4 1 2 ）を備える。プロセッサ 4 1 0 は、単独の集積回路上に形成されており、マイクロコントローラまたは同様の装置とすることができる。メモリ構造の各々は任意の種類メモリとすることができる。

40

【 0 0 3 3 】

プロセッサ 4 1 0 は、CPU 4 1 2 およびインタフェース・ブロック 4 1 6 の一方または両方に動作可能に結合される追加論理 4 2 0 も備える。追加論理 4 2 0 は、システムにデバッグおよび/またはエミュレーション機能を与えるために用いられるデバッグ/エミュレーション回路構成とすることができる。このようなデバッグ/エミュレーション機能には、ショウ・サイクル，バックグラウンド・デバッグ・モード，プログラム追跡などの可能

50

化が含まれる。インタフェース・ブロック 416 は、データ処理システム 400 に含まれるメモリ構造（格納回路構成）に結合される。図 4 に示される実施例においては、プロセッサ 401 は 32 ビット・データ・バスと N ビット・アドレス・バスとを備える。メモリ動作を実行するために必要とされる制御信号が制御バスを介してメモリ装置に伝えられる。

【0034】

32 ビット・メモリ 440 がデータ処理システム 400 に含まれる場合は、メモリ読込動作がインタフェース・ブロック 416 により開始されるとデータ・バス全体（D [31:0]）が駆動される。リセット・ベクトル・アドレス（T~W）に対応するデータ・バイトが 32 ビット・メモリ 440 内に図示されるように格納されると、アドレス 0x00, 0x04, 0x08, 0x0C を用いる 4 回の読込動作によって、リセット・ベクトルを構築する 4 バイトがデータ線 D [31:24] を介してインタフェース・ブロック 416 に提示される。

10

【0035】

これらの値を検索するために用いられるアドレス間の間隔は、32 ビット・メモリ 440 内で可能な最小のアドレス増分にあることに注目されたい。ここで言う最小増分は、32 ビット・メモリ 440 がアドレス線 A [1:0] を受信しないか無視するという事実に基づいて決まる。

【0036】

16 ビット・メモリ 450 を含むデータ処理システムにおいては、32 ビット・メモリ 440 からリセット・ベクトルを検索することに関して上記に説明されたのと同じメモリ動作を用いて、データ・バスの同じ部分で 16 ビット・メモリ 450 からリセット・ベクトルを検索することができる。16 ビット・メモリ 450 は、アドレス線 A [1] を無視しないことに注目されたい。しかし、32 ビット・メモリ 440（図示される例のデータ処理システム内で可能な最大ビット幅を有する）内で可能なアドレス指定粒度に基づきリセット・ベクトルを構築するバイトを隔てることによって、リセット・ベクトルを取り出すために用いられるアドレスをすべて種々の潜在的メモリ構成に適應することができる。

20

【0037】

データ処理システムが 8 ビット・メモリ 460 を含む場合は、32 ビットまたは 16 ビット・メモリ・システムに関してリセット・ベクトルを検索するために用いられるのと同じ集合のアドレスを利用して、8 ビット・メモリ 460 からリセット・ベクトルを検索することができる。8 ビット・メモリ 460 は、アドレス線 A [1:0] を受信および利用することに注目されたい。しかし、リセット・ベクトルのバイト部分を格納するために選択されるアドレス位置は、32 ビット・メモリ 440 内でアドレス可能な最小粒度に基づき互いに隔たっているので、リセット・ベクトルを検索するために用いられるアドレスを全般的に適應することができる。8 ビット・メモリ 460 は、データ・バスの 8 ビット部分に結合されるに過ぎないことに留意されたい。図 4 に示す例では、これにはデータ線 [31:24] が含まれる。そのため、インタフェース・ブロック 416 は、それらの特定のデータ線上のリセット・ベクトルの種々の部分に対応するデータを受信することが予測される。当業者には明白であろうが、リセット・ベクトル・アドレスを構築するデータ・バイトを検索するために用いられる特定のデータ線を、インタフェース・ブロック 416 と可能な 8 ビットおよび 16 ビットのメモリ構造 450, 460 との間の接続に基づいて変更することができる。

30

40

【0038】

図 3 に関して説明したように、リセット・ベクトルは、データ処理システム 400 内で種々のパラメータに適應される構成値を含むことがある。プロセッサ 410 は、その種々のブロック内に複数の制御レジスタ 414, 418, 422 を備えて図示され、これらのブロックは、この構成値に基づいて構成されるパラメータを含む。たとえば、CPU 412 は制御レジスタ 414 を備えて図示され、インタフェース・ブロック 416 は制御レジスタ 418 を備えて図示され、追加論理ブロック 420 は制御レジスタ 422 を備えて図示さ

50

れる。かくして、リセット・ベクトルに含まれる構成値は、これらの制御レジスタ 4 1 4 , 4 1 8 , 4 2 2 のうち 1 つ以上のレジスタ内に含まれるパラメータに適応される値を含む。

【 0 0 3 9 】

図 5 は、データ処理システムを構成する方法に対応する流れ図である。本方法は、段階 5 0 2 で始まり、データ処理システムがリセット状態で始動する。リセット信号がネゲート(解除)されて、リセット状態を脱すると、方法は段階 5 0 4 に進む。段階 5 0 4 において、データ処理システム内のハードウェアがデータ処理システムが動作を開始できるようにする種々の所定の状態に初期化される。

【 0 0 4 0 】

段階 5 0 6 において、リセット・ベクトル・アドレスが生成される。図 4 のデータ処理システムにおいては、段階 5 0 6 で生成されるリセット・ベクトル・アドレスは、メモリからリセット・ベクトルを検索するために用いられるベース・アドレスに対応する。図 4 に示される特定の例では、これはアドレス 0 x 0 0 に対応する。リセット・ベクトルのベース・アドレスはメモリ構造内の様々な点に位置する場合があることに留意されたい。

【 0 0 4 1 】

段階 5 0 8 において、リセット除外ハンドラ・ルーチン(リセット・ベクトル)のアドレスが、ハードウェアに関するリセット初期化値を用いて取り出される。3 2 ビットのアドレス空間を有するシステムにおいては、リセット・ベクトルは、メモリからの 4 回の個別のバイト・アクセスを実行することにより取り出される。この 4 回の個別のバイト・アクセスは、まずリセット・ベクトル・アドレスに基づきメモリにアクセスし、次にそのアドレスを既知の数量だけ増分して後続のバイト・アクセスのための後続アドレスを生成することにより実行される。好ましくは、後続アドレスを生成するために用いられる増分値は、リセット・ベクトルが格納される最大ビット幅のメモリのバイトで測定される幅に等しい。たとえば、3 2 ビット・メモリが最大の場合、増分値は 4 となる。この例では、4 の任意の倍数も増分値として機能するが、増分値が大きくなるとメモリ空間利用の効率が下がることに留意されたい。最大ビット幅メモリのバイト幅を増分値として用いると、リセット・ベクトルのバイト部分を、種々のメモリ構造内でできるだけ密接して配置することができる。このようにしてリセット・ベクトルを取り出すことにより、データ処理システムは、リセット・ベクトルを構築するバイトを格納する複数の異なるメモリ構成と共に動作することができる。

【 0 0 4 2 】

段階 5 1 0 において、段階 5 0 8 で実行される取出を通して受信されるデータ・バイトが検証される。図 3 に関して説明したように、取り出されたりリセット・ベクトルには構成値が含まれる。構成値には、制御部とデータ部とが含まれ、データ部の様々な部分が制御部の部分に基づきシステム内の種々のパラメータに適応されたりされなかったりする。従って、構成値に含まれるデータ部分の一部は無条件に適応されるが、条件付きで適応されるデータ部分もある。

【 0 0 4 3 】

段階 5 1 2 において、条件付き更新が実行されるか否かが判断される。好ましくは、この動作は構成値の制御部分を検証することによって実行される。制御部分が、いずれかの条件付きデータが個別の制御パラメータに適応されることを示すと、方法は段階 5 1 4 に進む。条件付き更新が実行されない場合は、方法は段階 5 1 6 に進む。

【 0 0 4 4 】

段階 5 1 4 において、個別の制御パラメータに条件付きで適応されるデータ部分を用いて、これらの制御パラメータが更新される。かくして、段階 5 0 4 においてこれらの制御パラメータに関して決定された初期化値は、構成値に含まれるデータ部分により無効とされる。段階 5 1 4 が終了すると、方法は段階 5 1 6 に進む。

【 0 0 4 5 】

段階 5 1 6 において、個別パラメータは無条件に適応される構成値のデータ部分を用いて

10

20

30

40

50

、個別パラメータを更新する。これにより、初期化状態に基づきこれらのパラメータに格納される前者の値は、構成値の無条件に適応されるデータ部分に含まれる値により無効になる。たとえば、ハードウェアが外部の16ビット幅メモリ構造と対話するよう初期構成される場合、段階508で取り出されるリセット・ベクトルは、この特定のパラメータを再構成して32ビット・メモリが実際に利用されるようにする構成値を含む。

【0046】

段階518において、リセット除外ハンドラの取出および実行が継続され、取出と実行の継続は段階514, 516で修正された可能な新たな更新制御パラメータを利用する。従って、制御値を取り出す第1バス・アクセスに続きシステム内で実行される第2バス・アクセスは、第1バス・アクセス中に取り出された制御値から導かれた再構成されたパラメータを利用する。そのため、段階518で行われる取出と実行は、段階504においてこれらの制御パラメータに格納される標準的な初期化済みの値がそのまま残される場合に可能であるよりも、さらに効率的に実行される。たとえば、システムが16ビット外部メモリに初期化され、32ビット外部メモリが実際に存在する場合は、外部メモリのビット幅を表すパラメータを変更することにより、段階518で行われる取出を、はるかに効率的に実行することができる。これは、データ・バスの16ビット部分のみでなく、32ビットすべてが関連すると考えられるためである。

【0047】

図2ないし図5に関して説明される本発明の実施例は、取り出されたりリセット・ベクトルに構成値を含む可能性がある。代替の実施例においては、構成値は、続いてリセット・ベクトルを取り出すことを前提として、独立したデータ部分として構成値を取り出す。かくして、リセット状態から抜け出した後でリセット・ベクトルを取り出す前に、構成値を検索して、システム内の初期化状態を変更するために用い、リセット・ベクトルの取出や、それに続くその他のメモリおよび他の動作がより効率的に実行されるようにする。このような技術を採用する方法を示す流れ図を図6に示す。

【0048】

段階602において、処理システムをリセット状態のままにするリセット信号がネゲートされたか否かが判定される。リセット信号がネゲートされた場合は、方法は段階604に進む。段階604は、ハードウェアが所定の状態に初期化される点で図5の段階504と類似のものである。

【0049】

段階606において、構成値アドレスが生成される。段階608において、構成値アドレスを用いて、メモリから構成値を検索する。構成値は、リセット信号のネゲーション後に他のいずれかのバスにアクセスする前に始動される格納装置(メモリ構造の1つ)に対するバス・アクセスを用いて検索される。構成値の検索は、CPU412またはメモリ・インタフェース416により実行される。構成値には、その初期化状態から改変されるパラメータの数に応じて、単独のデータ・バイトまたは複数のデータ・バイトを含む。構成値が複数のバイトを含む場合は、図2および図4においてリセット・ベクトルを取り出すために用いられたのと同様のアドレス法が利用されると、様々な異なる潜在的メモリ構成に対応する場合に、処理システムがリセットから確実に抜け出すことができる。

【0050】

段階612において、構成値を構築する検索された1つ以上のデータ・バイトが検証される。段階614において、決定が構成値の制御部に基づく場合に実行する必要がある条件付き更新が存在するか否かが判定される。制御部により一定の条件付き更新が行われることが示されると、方法は段階616に進む。条件付き更新を行わない場合は、方法は、段階616を迂回して、直接618に進む。

【0051】

段階616において、パラメータに対して行われる条件付き更新が実行される。段階618において、無条件の、あるいは制御部における特定の状態に依存しない更新も実行される。かくして、段階616, 618は、種々のパラメータを構成し、それらの状態が初期

10

20

30

40

50

化段階 604 において設定された状態から改変される。これは、図 7 に示される制御レジスタ内で検索された構成値の少なくとも一部分を格納することによって実行される。好ましくは、パラメータ構成は、構成値の初期検索に続く追加のバス・アクセスに先立って行われる。

【0052】

種々のパラメータが改変されてシステムが構成されると、方法は段階 619 に進み、ここで、種々のパラメータに関して新たに更新された値を用いて、リセット除外ハンドラ・ルーチンのアドレスが取り出される。かくして、段階 616, 618 で更新されるパラメータにリセット除外ハンドラのアドレスが格納されるメモリをアドレス指定するために用いられるチップ選択レジスタに対応するパラメータが含まれる場合は、段階 619 で行われる取出は、当初の初期化設定に基づいて可能であったものよりも、さらに効率的な方法で行われることになる。段階 620 において、メモリ読込動作を含むバス・サイクルを介してリセット除外ハンドラ・ルーチンが検索され、段階 619 で検索されたアドレスに基づいて実行される。段階 620 で行われる取出および実行は、修正されたパラメータ値を利用して行われることもある。このパラメータ値には、データ幅、バイト順序、必要な待機状態数またはシステムに含まれるメモリに対応するその他のバス・タイミング特性が含まれる。パラメータを構成することにより、取出および実行を、これらのパラメータの当初の初期化値で可能であるよりも効率的な方法で行うことができる。

【0053】

他の実施例では、構成値に基づき構成されるパラメータは、処理システムに含まれるデバッグまたはエミュレーション機能に関連するものを含む。そのため、デバッグまたはエミュレーション機能は、リセット状態脱出直後に可能化または不能化することができ、このような機能の用途が強化される。

【0054】

図 7 は、データ処理システムに含まれる種々のパラメータを構成するために構成値を利用する場合のブロック図を示す。構成値 710 は、リセット・ベクトルとは関連しない単独の値として図示されるが、これは図 3 の構成値 310 の場合とは異なる。かくして、構成値 710 は処理システムのリセット・ベクトルを取り出す前にメモリから取り出される。

【0055】

図 3 の構成値 310 の場合と同様に、図 7 の構成値 710 には、データ部と制御部とが含まれる。図示されるように、図 7 の構成値 710 のデータ部はデータ部分 712 ~ 718 を含む。これらのデータ部分は、制御レジスタ 350 の種々のパラメータ A ~ C 352 ~ 356 を構成するために利用され、また制御レジスタ 350 の外側に位置する他のシステム制御パラメータ 762 を構成するためにも用いられる。パラメータ A ~ C は、特定のチップ選択制御レジスタまたはデバッグまたはエミュレーション動作を伴う制御レジスタに関して説明される種々のパラメータであることに留意されたい。構成値 710 のパラメータ制御部分 720 を用いて、システム内の種々のパラメータに対する一定の更新を実行するか否かを決定することができる。かくして、種々のパラメータに対する更新は、構成値 710 の制御部により制御されるように条件付きの方法で行うことができるものもある。

【0056】

図 8 は、データ処理システム 800 を示す。このデータ処理システムは、3つの異なる可能なメモリ構成のうち1つ以上の構成に結合されるプロセッサ 810 を備えて図示される。かくして、プロセッサ 810 は、32ビット・メモリ 840, 16ビット・メモリ 850 および 8ビット・メモリ 860 のうちいずれかに結合される。図 4 のプロセッサ 410 の場合と同様に、プロセッサ 810 は、制御レジスタ 814 を備える CPU 812, 制御レジスタ 818 を備えるインタフェース・ブロック 816 および制御レジスタ 822 を備える追加論理ブロック 820 を有する。図示されるように、CPU 812 は、メモリに対するアクセスを CPU に提供するインタフェース・ブロック 816 に動作状態に結合される。追加論理 820 は、CPU 812 およびインタフェース・ブロック 816 のうち1つ以上と動作可能に結合される。

10

20

30

40

50

【 0 0 5 7 】

各メモリ 8 4 0 , 8 5 0 , 8 6 0 に関して図示されるように、構成情報 8 6 2 は、種々のメモリ構成の各々の中の同一アドレスを介してアクセスされる位置に格納される。さらに詳しくは、図示される例については、アドレス 0 x 0 0 を用いて、種々のメモリ構造のいずれかから構成情報を検索することができる。この場合、データ線 3 1 : 2 4 は、選択されたメモリからの構成情報により駆動される。

【 0 0 5 8 】

8 ビット・メモリ 8 6 0 の構成情報は、好ましくは、インタフェース・ブロック 8 1 6 が、8 ビット・メモリ 8 6 0 がデータ処理システムに含まれることを認識するよう構成されるようなパラメータ設定を含む。そのため、8 ビット・メモリ 8 6 0 への後のメモリ動作を、メモリの構成に適するように調整することができる。そのため、リセット・ベクトル・データ・バイト T , U , V , W を含むリセット・ベクトルを、8 ビット・メモリ 8 6 0 により連続するメモリ・アドレスに格納することができ、これらの連続するメモリ・アドレスは、インタフェース・ブロック 8 1 6 によりアクセス可能である。

10

【 0 0 5 9 】

それに対して、16 ビット・メモリ 8 5 0 は、インタフェース・ブロック 8 1 6 に対して、16 ビット・メモリがデータ処理システム 8 0 0 に含まれることを知らせる自身の構成情報 8 5 2 を含むことが好ましい。そのため、リセット・ベクトルに対応するデータ・バイトを16 ビット・メモリ 8 5 0 内で2つの16 ビット・ハーフ・ワードに配列することができ、インタフェース・ブロック 8 1 6 は、そのフォーマットでリセット・ベクトルを検索および利用することができる。

20

【 0 0 6 0 】

データ処理システム 8 0 0 に含まれる場合は、32 ビット・メモリ 8 4 0 も、インタフェース・ブロック 8 1 6 が32 ビット・データ 8 4 0 がデータ処理システム 8 0 0 に含まれることを認識するようパラメータ設定を含むそれ独自の構成情報 8 4 2 の集合を備えることが好ましい。そのため、32 ビット・メモリ 8 4 0 に対する単独のワード・アクセスを用いて、リセット・ベクトルを検索し、リセット除外ルーチンを実行することができる。構成情報をリセット・ベクトルに先立って検索することができるようにすることで、構成情報に含まれるパラメータ設定をリセット・ベクトルの検索に利用することができるので、検索はデータ処理システムに含まれるメモリ構造に最も適した方法で実行されることに注目されたい。

30

【 0 0 6 1 】

前記の説明において、本発明は特定の実施例に関して解説された。しかし、添付の請求項に明記される本発明の範囲から逸脱することなく、種々の修正および変更が可能であることが当業者には理解頂けよう。よって、仕様および図面は、制約するためのものでなく説明的なものであると見なされるべきであり、このような変更のすべては本発明の範囲内に含まれるものとする。

【 0 0 6 2 】

利点、他の長所および問題に対する解決策が、特定の実施例に関して上記に解説された。しかし、利点、長所および問題解決策と、利点、長所および問題解決策を実行するあるいは、より顕著なものとする要素は、請求項の一部または全部の決定的、所要のあるいは基本的特徴または要素として解釈すべきではない。本明細書で用いられる「によって構成される (comprises)」、「によって構成されている (comprising)」またはその変形は、非排他的包含を意図するものであり、要素の一覧を構成するプロセス、方法、物品または装置がその要素だけを含むものではなく、明確に記載されない、あるいはこのようなプロセス、方法、物品または装置に固有の他の要素を含むこともある。

40

【 図面の簡単な説明 】

本発明は、添付の図面において例をあげ、しかし、それに制約されずに説明される。図面内では、同様の参照記号は類似の要素を示す。

【 図 1 】 従来技術による処理システムのブロック図である。

50

【図2】本発明のある実施例によるリセット・ベクトル・アドレスを格納する種々のメモリ構成のブロック図である。

【図3】本発明のある実施例による制御レジスタ構成に利用される取り出されたりセット・ベクトル値のブロック図である。

【図4】本発明のある実施例によるデータ処理システムのブロック図である。

【図5】本発明のある実施例によりデータ処理システムを構成する方法の流れ図である。

【図6】本発明のある実施例によりデータ処理システムを構成する代替法の流れ図である。

【図7】本発明のある実施例により処理システム内に制御レジスタを構成するために用いられる構成値のブロック図である。

【図8】本発明の別の実施例によるデータ処理システムのブロック図である。図面内の要素は、簡単明瞭に図示されており、必ずしも同尺に描かれないことは当業者には理解頂けよう。たとえば、図面内のある要素の寸法は、本発明の実施例の理解を高めるために他の要素に対して誇張されている場合がある。

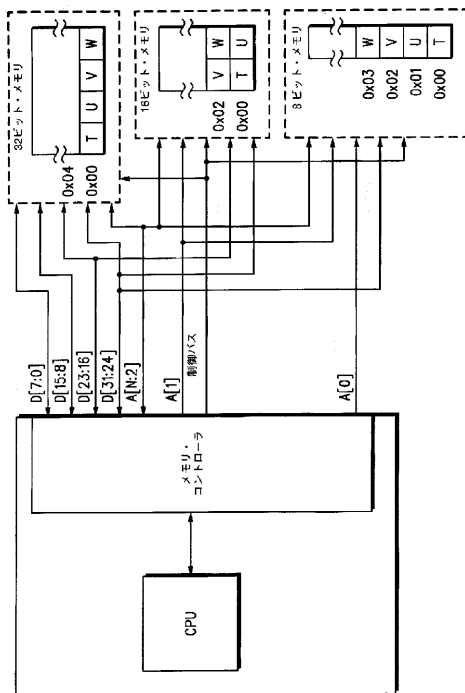
【符号の説明】

- 400 データ処理システム
- 410 プロセッサ
- 414, 418, 422 制御レジスタ
- 416 インタフェース・ブロック
- 420 追加論理
- 440 32ビット・メモリ
- 450 16ビット・メモリ
- 460 8ビット・メモリ
- B 制御バス

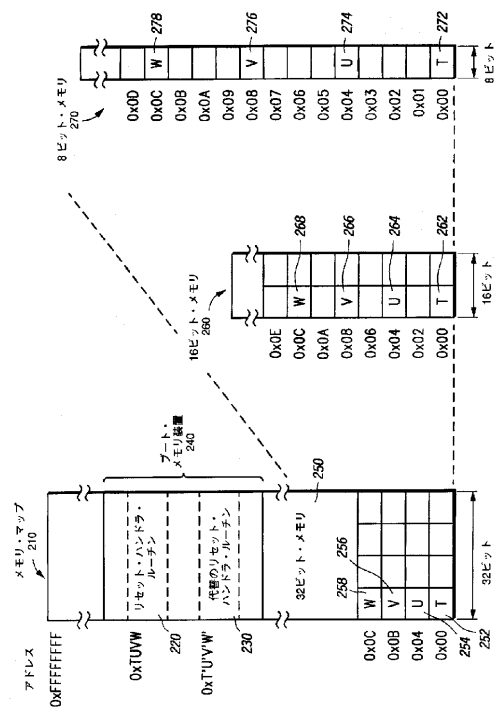
10

20

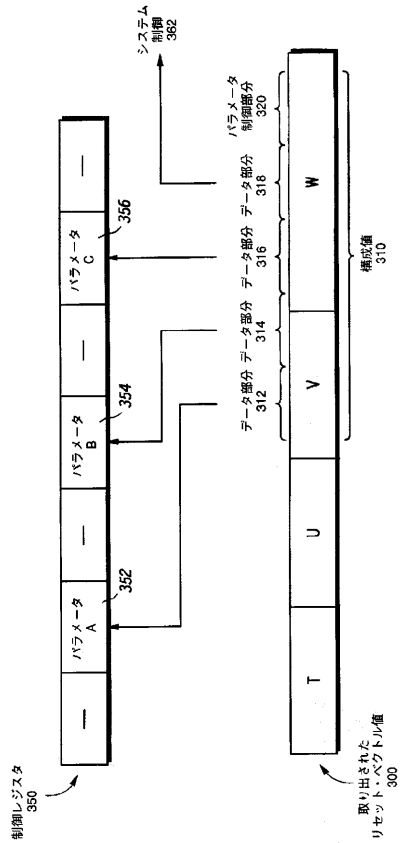
【図1】



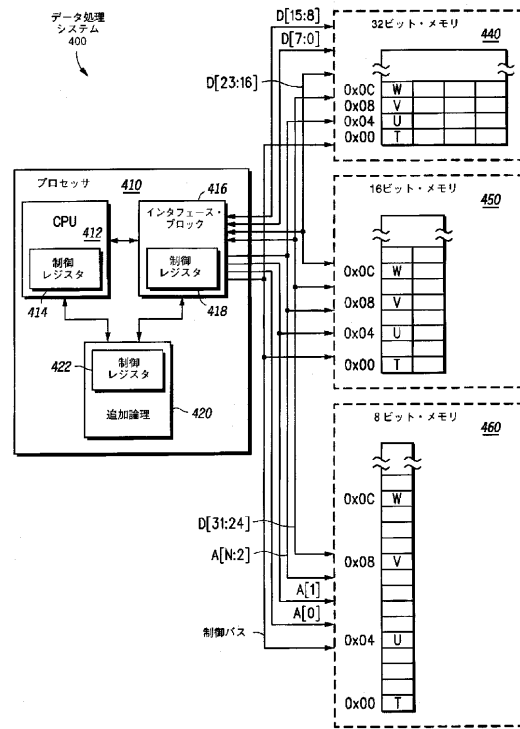
【図2】



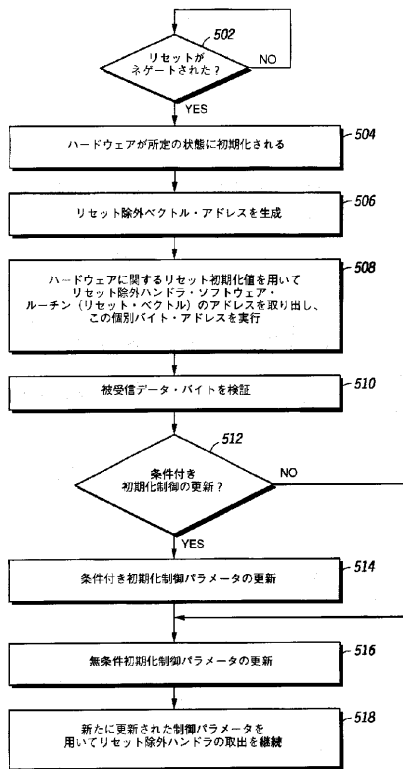
【図3】



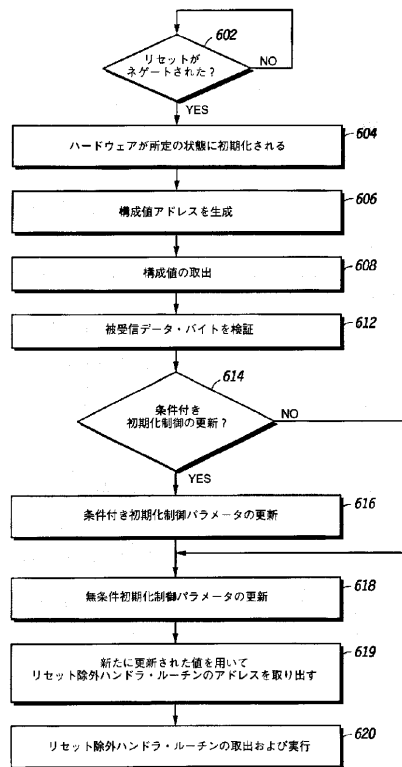
【図4】



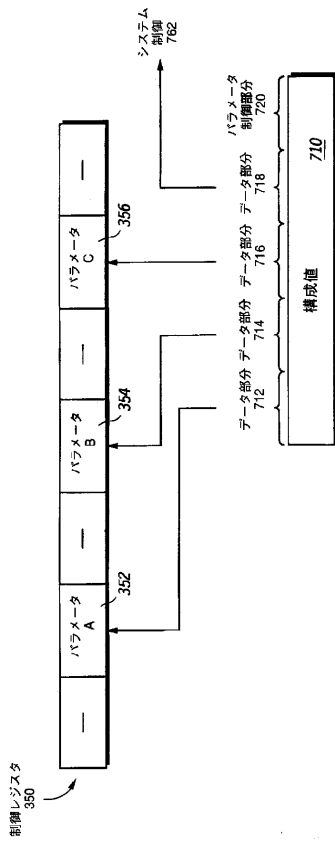
【図5】



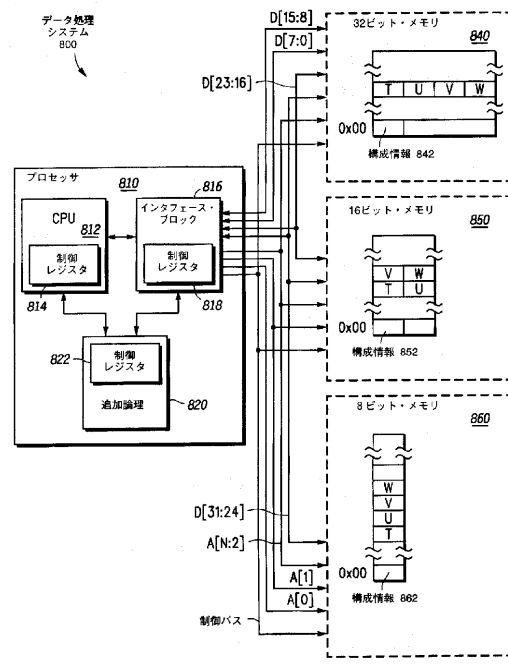
【図6】



【図7】



【図8】



フロントページの続き

- (72)発明者 ウィリアム・シー・モイヤー
アメリカ合衆国 テキサス州 78620 ドリッピング・スプリングス ピア・ブランチ・ロード1005
- (72)発明者 マイケル・ディー・フィッツシモンズ
アメリカ合衆国 テキサス州 78731 オースチン ノース・キャピタル・オブ・テキサス・ハイウェイ・ナンバー723 7700
- (72)発明者 ジェイムス・シー・ナッシュ
アメリカ合衆国 テキサス州 78746 オースチン クレスティッド・ビュート1700

審査官 杉藤 泰子

- (56)参考文献 特開平03-260858(JP,A)
特開平10-198463(JP,A)
特開平09-016292(JP,A)
特開昭61-011873(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F 1/24
G06F 12/06