

(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2004-164113

(P2004-164113A)

(43) 公開日 平成16年6月10日(2004.6.10)

(51) Int. Cl.⁷

G06F 15/177

G06F 9/46

G06F 11/22

G06F 11/28

F I

G06F 15/177 678H

G06F 15/177 670A

G06F 9/46 310K

G06F 11/22 310R

G06F 11/28 L

テーマコード(参考)

5B042

5B045

5B048

5B098

審査請求 未請求 請求項の数 12 O L (全 17 頁)

(21) 出願番号 特願2002-327238(P2002-327238)

(22) 出願日 平成14年11月11日(2002.11.11)

(71) 出願人 000232036

NECマイクロシステム株式会社
神奈川県川崎市中原区小杉町1丁目403
番53

(74) 代理人 100109313

弁理士 机 昌彦

(74) 代理人 100085268

弁理士 河合 信明

(74) 代理人 100111637

弁理士 谷澤 靖久

(72) 発明者 宮本 康宏

神奈川県川崎市中原区小杉町一丁目403
番53 エヌイーシ
ーマイクロシステム株式会社内

最終頁に続く

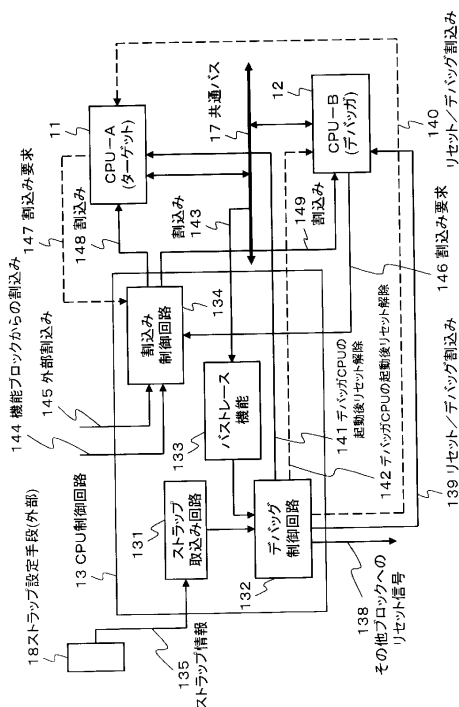
(54) 【発明の名称】 マルチCPUのリセット回路およびリセット方法

(57) 【要約】

【課題】 システムLSIまたはその搭載システムの評価の容易性向上を図る。

【解決手段】 複数のCPUを内蔵するシステムLSIのリセットおよび割込みを制御するCPU制御回路13は、外部からストラップ情報が設定されるストラップ取込み回路131と、CPUのうち一方をターゲットCPU11に、他方をデバッグCPU12に指定し、CPU12に対しリセット/デバッグ割込み信号を出力し、CPU12の起動後、CPU11のリセットを解除するとともに、機能ブロック14~16にもリセット信号を出力するデバッグ制御回路132と、機能ブロックおよび外部からの割込み信号とCPU12からの割込み要求信号とを入力し、CPU11およびCPU12に割込みをかける割込み制御回路134と、共通バスライン17からCPU11およびCPU12の情報を取り込むバストレース手段133とを備える。

【選択図】 図2



【特許請求の範囲】

【請求項 1】

複数の CPU を内蔵するシステム LSI のリセットおよび割込みを制御する CPU 制御手段を備え、前記 CPU 制御手段が、前記複数の CPU のうちの一つを、通常動作のプログラムをデバッグプログラムに置き換えたデバッガ手段に設定する機能を有することを特徴とするマルチ CPU のリセット回路。

【請求項 2】

前記 CPU 制御手段によって前記デバッガ手段に置き換えられた前記 CPU が、他方の CPU をターゲット CPU としてデバッグする機能を有する請求項 1 記載のマルチ CPU のリセット回路。

10

【請求項 3】

3 個以上の CPU を内蔵するシステム LSI のリセットおよび割込みを制御する CPU 制御手段を備え、前記 CPU 制御手段による前記 CPU の選択結果に対応して、前記 3 個以上の CPU のうちの 1 個がデバッガ手段として他方をデバッグする相互デバッグ機能を有することを特徴とするマルチ CPU のリセット回路。

【請求項 4】

前記 CPU 制御手段は、前記デバッガ手段として設定された CPU とターゲット側 CPU および各機能ブロックとを予め定めた一定の時間差でリセット解除し、前記デバッガとして設定された CPU を前記ターゲット側 CPU および前記各機能ブロックよりも先に起動させる機能を有する請求項 1 または 3 記載のマルチ CPU のリセット回路。

20

【請求項 5】

前記一定の時間差に代えて、デバッガの起動後に前記制御回路に割込み信号で通知するタイミングとする請求項 4 記載のマルチ CPU のリセット回路。

【請求項 6】

前記 CPU 制御手段は、リセット信号の活性化以前に予め外部で設定されたストラップ情報をストラップ取込み手段により取り込み、取り込んだ前記ストラップ情報に基づきデバッグ制御手段により前記第 1 および前記第 2 の CPU を一方はデバッガ CPU に他方はターゲット CPU に起動状態を切り替えるとともに、前記切り替え結果に応じて前記デバッガ CPU と前記ターゲット CPU および前記機能ブロックとを時間差をもってリセット解除し、通常動作時またはデバッグ動作時ともに外部からの割込みまたは内部の機能ブロックからの割込みを割込み制御手段により検知して前記デバッガ CPU および前記ターゲット CPU に割込みをかけ、さらにデバッグ時には前記デバッガ CPU からの割込み要求を受けて前記ターゲット CPU に対して割込み制御を行う機能を有する請求項 1 または 3 記載のマルチ CPU のリセット回路。

30

【請求項 7】

前記 CPU 制御手段は、前記 CPU それぞれがインタフェースするバスラインの情報を取り込む機能を有する請求項 1 ~ 5 または 6 記載のマルチ CPU のリセット回路。

【請求項 8】

外部メモリと接続するバスラインと、前記バスラインを介して相互に接続される、第 1 の CPU、第 2 の CPU、機能ブロック群および CPU 制御手段を有し、前記 CPU 制御手段は、外部からストラップ情報が設定されるストラップ取込み手段と、前記 CPU のうち一方をターゲット CPU に、他方をデバッガ手段としてのデバッガ CPU に指定し、前記デバッガ CPU に指定された CPU に対しリセットおよびデバッグ割込み信号を出力し、前記デバッガ CPU の起動後、前記ターゲット CPU のリセットを解除するとともに、前記機能ブロックにもリセット信号を出力するデバッグ制御手段と、前記機能ブロックからの割込み信号と外部からの割込み信号と前記デバッガ CPU からの割込み要求信号を入力し、前記ターゲット CPU および前記ターゲット CPU に割込みをかける割込み制御回路と、前記共通バスラインから前記デバッガ CPU および前記ターゲット CPU の情報を取り込むバstreasure手段と、を備えることを特徴とするマルチ CPU のリセット回路。

40

【請求項 9】

50

前記ストラップ情報の取り込み端子は、前記CPU制御手段によるリセット信号の活性化が解除された後の非活性化期間中は他の信号の入力、出力または入出力端子となる請求項6記載のマルチCPUのリセット回路。

【請求項10】

複数のCPUを内蔵するシステムLSIのリセットおよび割込みを制御するCPU制御手段を用いて、前記複数のCPUのうちの一つを、通常動作用のプログラムをデバッグプログラムに置き換えたデバッガ手段に設定して、他方の前記CPUをデバッグすることを特徴とするマルチCPUのリセット方法。

【請求項11】

3個以上のCPUを内蔵するシステムLSIのリセットおよび割込みを制御するCPU制御手段の有する相互デバッグ機能により、前記CPUの選択結果に対応して、前記3個以上のCPUのうち選択された1個のCPUがデバッガ手段として残りのCPUをデバッグすることを特徴とするマルチCPUのリセット方法。

【請求項12】

システムLSIに内蔵するCPU制御手段が、動作環境を設定するためにリセット信号の活性化以前に予め外部で設定されたストラップ情報の取り込み手段と、前記CPUをデバッガ手段とするかターゲット側とするかの起動状態を切り替える切り換え手段と、前記CPUとともに内蔵する機能ブロックのリセットを制御する内部制御手段と、通常動作時またはデバッグ動作時に外部または内部からの割込みを検知して前記CPUそれぞれに対して割込み制御を行う割込み制御手段とを有し、前記CPU制御手段により、

前記リセット信号を検知すると内部の各機能ブロックに対してリセット信号を発行する処理と、

外部で設定された前記ストラップ情報を取り込み、前記CPUの動作設定を読み取るとともに、読みとった内容が通常動作を指定している場合は内部の全てのリセットを解除し、デバッグモードを指定している場合は、前記CPUのいずれかを前記デバッガ手段のCPUに指定する処理と、

前記デバッガ手段に指定したCPUに対するリセットを他のブロックよりも先に解除し、リセットを解除された前記CPUを起動してデバッグプログラムを読み込ませる処理と、前記CPUにデバッグプログラムを実行させてデバッグを行う準備が整った後に、残りのCPUおよび内部の前記機能ブロックそれぞれのリセットを解除して、通常動作を起動させる処理と、を有することを特徴とするマルチCPUのリセット方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明はマルチCPUのリセット回路およびリセット方法に係わり、特にシステムLSIに内蔵する複数のCPUを切り替えて相互にデバッグ可能とするマルチCPUのリセット回路およびリセット方法に関する。

【0002】

【従来の技術】

近年、半導体素子の微細化技術の進展に伴い、これらの半導体素子で構成したメモリ、マイクロコンピュータ等の半導体装置(LSI)の機能も複雑、かつ大規模化しているが、これら複数のLSIを1チップに搭載したシステムLSIであるシステムオンチップ(SOC: System On Chip)などでは、ある特定の目的のために種々の回路を構成する場合が多い。

【0003】

上述したシステムLSIのうちCPUを複数個内蔵するシステムLSIを搭載したマイコン応用製品開発も仕様の複雑化とともにプログラム容量の増加による開発時間も当然のことながら増大してきているが、市場の要求に答えるためには、複雑化したプログラム開発に短時間で対処しなければならないため、デバッグを効率的に行えるように予め対策したシステムLSIが必要となる。

10

20

30

40

50

【0004】

一方、前述したシステムLSIでは、例えば、トランスポートストリーム(TS: Transport Stream)デコーダ、MPEG(Moving Pictures Expert Group)ビデオデコーダ、オーディオデコーダ、メモリ転送、表示系に分かれて、それぞれ個別にハードウェアロジックを構成していたが、近年はCPUの処理能力の向上に伴って、ハードウェアロジックに相当する処理を行う別のCPUを内蔵することも可能になっている。

【0005】

上述した従来のシステムLSIの一例の構成を示した図7を参照すると、システムLSI 31は、CPU-A: 311と、CPU-B(機能ブロック4): 312と、機能ブロック1: 313~機能ブロック3: 315とが共通バスライン316で相互に接続され、共通バスライン316は外部メモリ32にも接続されている。

10

【0006】

また、外部の機器とは外部入出力(パラレル/シリアル)信号線317を介してインタフェースする。

【0007】

外部メモリ32にはCPU-A用のプログラム33とCPU-B用のプログラム(機能ブロック4の動作プログラム)とが格納されている。

【0008】

ところで、上述した従来の構成を備えるシステムLSIのデバッグ方法は、以下のよう

20

【0009】

システムLSI 31が内蔵する機能ブロック1: 313が例えばJTAG(Joint Test Action Group)などのデバッグポートを経由して、内部メモリ、外部メモリ、各種レジスタなどの情報を設定し、または出力させる。

【0010】

上述のJTAGと呼ばれるテスト手法は、デバイスのほとんど全てのピンに、あらかじめシリコンのテストプローブが内蔵されているもので、シリコンのテストプローブは、例えばLSI内部の機能ブロックと入出力ピンとの間にセルと称するシフトレジスタを配置することにより、セルにデータを入出力することができ、セルにテスト時に用いるテストの

30

【0011】

上述した従来の構成を備え、かつJTAGを利用したシステムLSIの一例を示した図8を参照すると、JTAGからの出力情報は、信号線46を介しそれを接続したロジックアナライザや、汎用のPC等評価用機器45に出力または蓄積する。

【0012】

この例においては、次のようなデバッグ手法も行われる。すなわち、内蔵するCPU-B(機能ブロック4)411でデバッグ用のプログラムを実行しておき、シリアルないしパラレルのポートにより信号線47を介して評価用機器45に評価用のプログラムを転送して実行する。その結果は、外部のPC等へ出力され、割込み等をトリガーとして観測する。

40

上述したJTAG、外部インタフェースを有する例では例えば特開平7-182204号公報がある(特許文献1参照)。

【0013】

また、他の方法の一例として、JTAGを利用し、かつROMシミュレータも利用したシステムLSIの一例を示した図9を参照すると、ROMエミュレータ52等を接続し、CPU-A評価用のプログラム551をCPU-B: 512の起動時に信号線58からバス58を介してROMエミュレータ52に読み込ませ、評価を実行する。その動作結果は外部入出力(シリアル/パラレル)のポートから信号線57を介して外部のPC等へ出力され、割込み等の信号をトリガーとして観測する。この例では例えば特開2002-268911号公報がある(特許文献2参照)。

50

【0014】

あるいは、JTAGを利用したシステムLSIのさらに他の一例を示した図10を参照すると、システムLSI61内部に動作状況を記録する専用のトレースRAM(トレースメモリ)614とトレース制御部613とを設け、システムLSI61内部に保有する専用の回路によって内部の動作を逐次記録しておき、不正なアクセスまたは外部からの割り込み等によって記録を停止し、その後トレースRAM614からシステムLSI61の外部に読み出して動作を解析するという方法がある。この例では例えば特許第2752592号公報がある(特許文献3参照)。

【0015】

一方、複数のプロセッサや機能ブロックからなるシステムLSIにおいては、構成する各LSIブロックでのバグ時のリセットを、デバッグ対象の指定に応じて適切に制御する必要があるが、特開2002-73363号公報では、LSIデバッグ装置のチップを複数使用し、それぞれのLSIブロックのデバッグ機能を提供すると同時に、全体ではLSIデバッグ装置の本来の全LSI端子機能を実現することにより、LSIブロックのデバッグ時のリセットを、デバッグ対象の指定に応じて適切に制御する(特許文献4参照)。

【0016】

本発明が関するシステムLSIにおいては、近年次のような環境状態にある。

【0017】

LSIの動作スピードが向上していること、半導体素子の微細化の進展によりLSI集積度が向上し機能が複雑化していること、チップサイズの更なる縮小(コスト削減)への要求があること等である。

【0018】

【特許文献1】

特開平7-182204号公報(段落番号「0020」「0023」~「0024」、図1、図2)

【特許文献2】

特開2002-268911号公報(段落番号「0017」~「0021」、図3)

【特許文献3】

特許第2752592号公報(段落番号「0004」「0009」、図3)

【特許文献4】

特開2002-73363号公報(段落番号「0034」「0035」「0036」、図4)

【0019】

【発明が解決しようとする課題】

従来システムLSIは、上述した環境状態にあるという背景に照らし合わせると、以下のような問題点を有する。

【0020】

システムの動作スピードが向上して、外部端子経由での実動作の測定が困難になる。

【0021】

システムの動作が複雑化して、固定化したトレースRAM等では測定の限界に達する。

【0022】

通常動作に寄与しない回路の増大があり、コストがかかる。

【0023】

上述の問題点が生じる理由を、以下に述べる。

【0024】

まず、JTAGやROMエミュレータを介してのデバッグ作業は、転送速度等に限界があり、CPUをはじめとするシステムLSIの動作速度が向上するにしたがい、実際と同じ状態での動作を観測することが難しくなっている。また、障害のひとつにシステム立ち上げ時の動作不良という問題がしばしば発生する。その場合、従来方法ではシステムが正常に動作している状態において観測が可能である。すなわち、システムが起動すること

を前提にしているため、システムを起動するときの動作については観測することができない。

【0025】

これに関する解決策のひとつが、デバッグのためにトレース用のメモリを搭載することである。この方法では、構成しだいでシステムの起動時における種々の動作を観測・記憶することが可能である。

【0026】

しかし、システムの動作速度が向上し、かつ複雑化しているために、トレースメモリにおいて観測できる範囲および時間が限定されてしまう問題が挙げられる。また、トレース専用の回路とRAMを載せることで通常動作に寄与しない回路規模が増大し、チップサイズ縮小といったコスト削減の要求を満たすことができない。 10

【0027】

また、上記特許文献4におけるリセットの方法では、通常動作時とデバッグ時とのリセット制御端子が異なるので端子割り当てに余裕が必要である。

【0028】

本発明の目的は、上述した従来の欠点に鑑みなされたものであり、システムLSIに内蔵した複数のCPUのうちの一つを、通常動作用のプログラムをデバッグプログラムに置き換えることとあわせて、従来技術でトレースのために用いていた回路とトレースメモリをCPUの処理によって代替することを可能とし、また複数のCPUを切り替えて相互にデバッグを可能とすることによって、当該システムLSIないしそれを組み込んだシステムにおける評価の容易性向上を図ることにある。 20

【0029】

【課題を解決するための手段】

本発明のマルチCPUのリセット回路は、複数のCPUを内蔵するシステムLSIのリセットおよび割込みを制御するCPU制御手段を備え、前記CPU制御手段が、前記複数のCPUのうちの一つを、通常動作用のプログラムをデバッグプログラムに置き換えたデバッグ手段に設定する機能を有することを特徴とする。

【0030】

また、前記CPU制御手段によって前記デバッグ手段に置き換えられた前記CPUが、他方のCPUをターゲットCPUとしてデバッグする機能を有することができる。 30

【0031】

本発明のマルチCPUのリセット回路の他の特徴は、3個以上のCPUを内蔵するシステムLSIのリセットおよび割込みを制御するCPU制御手段を備え、前記CPU制御手段による前記CPUの選択結果に対応して、前記3個以上のCPUのうち1個がデバッグ手段として他方をデバッグする相互デバッグ機能を有することにある。

【0032】

また、前記CPU制御手段は、前記デバッグ手段として設定されたCPUとターゲット側CPUおよび各機能ブロックとを予め定めた一定の時間差でリセット解除し、前記デバッグ手段として設定されたCPUを前記ターゲット側CPUおよび前記各機能ブロックよりも先に起動させる機能を有することができる。 40

【0033】

また、前記一定の時間差に代えて、デバッグの起動後に前記制御回路に割込み信号で通知するタイミングとする。

【0034】

さらに、前記CPU制御手段は、リセット信号の活性化以前に予め外部で設定されたストラップ情報をストラップ取込み手段により取り込み、取り込んだ前記ストラップ情報に基づきデバッグ制御手段により前記第1および前記第2のCPUを一方はデバッグCPUに他方はターゲットCPUに起動状態を切り替えるとともに、前記切り替え結果に応じて前記デバッグCPUと前記ターゲットCPUおよび前記機能ブロックとを時間差をもってリセット解除し、通常動作時またはデバッグ動作時ともに外部からの割込みまたは内部の機 50

能ブロックからの割込みを割込み制御手段により検知して前記デバッガCPUおよび前記ターゲットCPUに割込みをかけ、さらにデバッグ時には前記デバッガCPUからの割込み要求を受けて前記ターゲットCPUに対して割込み制御を行う機能を有する。

【0035】

さらにまた、前記CPU制御手段は、前記CPUそれぞれがインタフェースするバスラインの情報を取り込む機能を有する。

【0036】

本発明のマルチCPUのリセット方法の特徴は、外部メモリと接続するバスラインと、前記バスラインを介して相互に接続される、第1のCPU、第2のCPU、機能ブロック群およびCPU制御手段を有し、前記CPU制御手段は、外部からストラップ情報が設定されるストラップ取込み手段と、前記CPUのうち一方をターゲットCPUに、他方をデバッガ手段としてのデバッガCPUに指定し、前記デバッガCPUに指定されたCPUに対しリセットおよびデバッグ割込み信号を出力し、前記デバッガCPUの起動後、前記ターゲットCPUのリセットを解除するとともに、前記機能ブロックにもリセット信号を出力するデバッグ制御手段と、前記機能ブロックからの割込み信号と外部からの割込み信号と前記デバッガCPUからの割込み要求信号を入力し、前記ターゲットCPUおよび前記ターゲットCPUに割込みをかける割込み制御回路と、前記共通バスラインから前記デバッガCPUおよび前記ターゲットCPUの情報を取り込むバストレーズ手段と、を備えることにある。

10

【0037】

また、前記ストラップ情報の取り込み端子は、前記CPU制御手段によるリセット信号の活性化が解除された後の非活性化期間中は他の信号の入力、出力または入出力端子とすることにある。

20

【0038】

本発明のマルチCPUのリセット方法のさらに他の特徴は、複数のCPUを内蔵するシステムLSIのリセットおよび割込みを制御するCPU制御手段を用いて、前記複数のCPUのうちの一つを、通常動作のプログラムをデバッグプログラムに置き換えたデバッガ手段に設定して、他方の前記CPUをデバッグすることを特徴とする。

【0039】

本発明のマルチCPUのリセット方法のさらにまた他の特徴は、3個以上のCPUを内蔵するシステムLSIのリセットおよび割込みを制御するCPU制御手段の有する相互デバッグ機能により、前記CPUの選択結果に対応して、前記3個以上のCPUのうち選択された1個のCPUがデバッガ手段として残りのCPUをデバッグすることにある。

30

【0040】

本発明のマルチCPUのリセット方法のまた他の特徴は、システムLSIに内蔵するCPU制御手段が、動作環境を設定するためにリセット信号の活性化以前に予め外部で設定されたストラップ情報の取り込み手段と、前記CPUをデバッガ手段とするかターゲット側とするかの起動状態を切り替える切り換え手段と、前記CPUとともに内蔵する機能ブロックのリセットを制御する内部制御手段と、通常動作時またはデバッグ動作時に外部または内部からの割込みを検知して前記CPUそれぞれに対して割込み制御を行う割込み制御手段とを有し、前記CPU制御手段により、

40

前記リセット信号を検知すると内部の各機能ブロックに対してリセット信号を発行する処理と、

外部で設定された前記ストラップ情報を取り込み、前記CPUの動作設定を読み取るとともに、読みとった内容が通常動作を指定している場合は内部の全てのリセットを解除し、デバッグモードを指定している場合は、前記CPUのいずれかを前記デバッガ手段のCPUに指定する処理と、

前記デバッガ手段に指定したCPUに対するリセットを他のブロックよりも先に解除し、リセットを解除された前記CPUを起動してデバッグプログラムを読み込ませる処理と、前記CPUにデバッグプログラムを実行させてデバッグを行う準備が整った後に、残りの

50

CPUおよび内部の前記機能ブロックそれぞれのリセットを解除して、通常動作を起動させる処理と、を有することにある。

【0041】

【発明の実施の形態】

まず、本発明のマルチCPUのリセット回路およびリセット方法の概要を述べると、複数のCPUを内蔵するシステムLSIに、図1に示すようなりセット等を制御する手段を付加し、この手段によって、システムLSIに内蔵した複数のCPUのうちの一つを、通常動作のプログラムをデバッグプログラムに置き換えることができるものである。

【0042】

すなわち、従来のシステムLSIのデバッグを行うためには、LSI外部端子を經由して内部・外部メモリ、そしてレジスタ等の情報をLSI外部あるいはシステムの外部から観測する方法、またはトレースのために専用の回路とトレース専用の内部メモリを用いて記録する方法などがあった。

【0043】

これに対し、本発明では、これらを内蔵CPUの処理によって代替することを可能とし、また複数のCPUを切り替えて相互にデバッグを可能とする手段を提供するものである。

【0044】

したがって、本発明によって、システムLSIないしそれを組み込んだシステムにおける評価の容易性を向上させることができる。

【0045】

次に、本発明の第1の実施の形態を図面を参照しながら説明する。

【0046】

本発明の第1の実施の形態におけるシステムLSIの主要構成を示した図1を参照すると、本発明のシステムLSI1と、システムLSI1の共通バスラインと接続し外部に配置される外部メモリ2と、外部メモリ2に格納される、システムLSI1内部のCPU-A用のプログラム3およびCPU-AをデバッグするためCPU-Bに格納されるデバッグ用のプログラム4とで構成される。

【0047】

システムLSI1は、ターゲットとなるCPU-A:11と、トレース機能およびデバッグ機能を有するデバッガ手段として指定されるCPU-B:12と、複数のCPUを持つシステムLSIのリセットおよび割込み等を制御するCPU制御回路13と、外部機器とインタフェースする機能ブロック1:14と、その他の機能ブロック2:15、機能ブロック3:16と、これらの各CPU-A:11、CPU-B:12CPU制御回路、各機能ブロックおよび外部メモリを相互に接続する共通バスライン17とから構成される。

【0048】

CPU制御回路13の構成を示した図2を参照すると、このCPU制御回路13は、外部に配置されるストラップ情報設定手段18から入力するストラップ等の情報を取り込むストラップ取込み回路131と、ストラップ取込み回路131からの信号に应答して、CPU-A:11およびCPU-B:12にリセットおよび割込み信号をそれぞれ与えるとともに、その他の機能ブロックへのリセット信号を供給するデバッグ制御回路132と、共通バスライン17を監視し、CPUの起動状態を切り替えるとともにCPU-A:11およびCPU-B:12からの情報を収集するバストレーサ機能用ブロック133と、CPU-A:11およびCPU-B:12からの割込み要求信号と機能ブロック1:14~機能ブロック3:16からの割込み信号と外部からの割込み信号を検知するとともに、CPU-A:11およびCPU-B:12に対して割込みをかける割込み制御回路134とを有する。

【0049】

なお、デバッグ制御回路132から出力するリセット/デバッグ割込み信号139および140と、割込み要求信号146および147と、デバッガCPU起動後のリセット解除信号141および142とは、デバッグ時にはそれぞれ排他的に供給され、または入力す

る。

【0050】

つまり、例えばCPU-A:11がターゲット側、CPU-B:12がデバッグ側の時は、図中実線で示した信号が入出力され、その逆の場合は点線で示した信号が入出力される。

【0051】

ここでストラップは、当該LSIの基本的な動作環境を設定するためにLSIの動作の立ち上がりより以前に外部で設定されている情報を指す。その設定は、主としてLSIの外部端子を用いて、リセット期間においてLSI内部の取込み回路によって読み込まれ、ハイレベル・ローレベル、およびそれらの組み合わせで行う。リセットを解除した後は、通常の外部端子として機能する。

10

【0052】

デバッグ制御回路132は、LSIの組み込まれたシステムにおいてリセット、ここでは電源の投入、またはリセットボタンの押下などを感知し、内部の各ブロックに対してクロックに同期化させたリセットの発行と停止を行って、各ブロックの初期化と起動を行わせるものである。

【0053】

割込み制御回路134は、システムLSI1またはそれを含むシステムにおいて障害が発生した場合、またはデバッグを行う時などにおいて、システムLSI内部または外部端子を通じてシステムから割込み要求を受けた時に、CPU-A:11またはCPU-B:12に対して動作の停止を指示する等の機能をもっている。

20

【0054】

当該システムLSI1においては、CPUは複数存在するため、割込みを要求するCPUを選択する機能をもっている。

【0055】

例えば通常の動作時においては、この制御回路13は、複数のCPU-A:11およびCPU-B:12と各機能ブロック1:14～機能ブロック3:16とに対して同時にリセットを発行、解除する機能をもっている。

【0056】

しかし、どちらか一方のCPUで、ストラップ等によってデバッグを行うよう設定された場合は、この制御回路13はリセットに関して特別な動作を行う。

30

【0057】

すなわち、デバッグを行う状態において、デバッグ用プログラムを実行する方のCPUを例えばデバッガCPU-B:12とし、他方のCPU-A:11をターゲット側CPUとして区別する。

【0058】

図1においては、CPU-A:11をターゲット側、CPU-B:12をデバッガ側としているが、各々はストラップ等の設定によってその役割を交換することが可能である。そのため、信号配線は両CPUに対して同じような信号線が予め平列に接続されている。

【0059】

デバッグ制御回路132は、デバッガCPU-B:12とターゲット側CPU-A:11および各機能ブロック1:14～機能ブロック3:16とを、予め定める一定の時間差でリセット解除するように制御する。

40

【0060】

すなわち、デバッガCPU-B:12は、ストラップ情報に応答してリセットを検出すると、他のブロックよりも先に起動する。これはCPU-A:11のブートアップから監視するために、デバッガCPU-B:12を先に起動して監視の準備を行うためである。

【0061】

加えて、制御回路13は、前述したバストレー스機能用ブロック133によって、CPU-A:11およびCPU-B:12が属する共通バスライン17の情報を取り込むことが

50

できる。

【0062】

したがって、外部メモリ2に格納するデバッグ用のプログラム4は、通常動作時のためのプログラムと同様に、あらかじめ外部メモリ2に配置しておき、デバッガCPU-B:12は起動時に共通バスライン17を介して、これを読み込んで実行する。

【0063】

ターゲット側CPU-A:11は、デバッガCPU-B:12の起動が完了した後にリセットが解除され、通常の手順で起動し、通常動作を行う。

【0064】

以下、本実施の形態の動作を図1および図2と、通常起動状態におけるCPU制御回路の動作説明用タイミングチャートを示した図3と、デバッグモードでの起動状態におけるCPU制御回路の動作説明用タイミングチャートを示した図4をそれぞれ参照しながら説明する。

10

【0065】

まず、図1,図2および図3において、CPU制御回路13は、まず、タイミングt1で、ストラップ取込み回路131により、ハードウェアリセットを検知すると、タイミングt2において、LSI内部のCPU-A:11, CPU-B:12および各機能ブロック1:14~機能ブロック3:16に対してリセットを発行する。

【0066】

次に、タイミングt3において、外部のストラップ情報を取り込み、ハードウェアの動作設定を読み取る。ここで、ストラップによる設定が通常動作を示している場合は、制御回路13はシステムLSI1が通常動作状態で動作するよう、タイミングt4において各ブロックのリセットを解除する。

20

【0067】

一方、ストラップ設定がデバッグモードを示した場合は、その動作をCPU-A:11またはCPU-B:12のいずれかに行わせるための作業を行う。

【0068】

すなわち、図1,図2および図4において、制御回路13は、2個あるCPU-A:11およびCPU-B:12のうち、デバッガとして動作させる方のCPUをタイミングt3において決定する。この決定はストラップの取り込み等による。

30

【0069】

タイミングt3においてデバッガとするCPUを、例えばCPU-B:12に決定した制御回路13は、タイミングt4においてそのCPU-B:12に対するリセットを、他のブロックに先駆けて解除する。リセットを解除されたCPU-B:12は、起動してプログラムを読み込み実行する。

【0070】

CPU-B:12がプログラムを実行してデバッグを行う準備が整った後に、タイミングt5において制御回路13は他方のCPU-A:11および内部の各機能ブロック1:14~機能ブロック3:16のリセットを解除する。解除されたブロックは通常と同じ手順にて起動する。

40

【0071】

デバッガCPU-B:12の起動と他のブロックの起動との間に設ける時間(タイミングt4~t5)は、デバッガがその準備を終えるのに十分な時間であり、これはハードウェアの設定として一定の時間であってもよいし、デバッガが起動した後に、制御回路13に対して準備ができたことを割込み等によって通知する方法を取ってもよい。

【0072】

これら一連の動作は、相互に交換可能なようになっており、CPU-A:11をデバッガとして使用する場合はCPU-A:11を先に起動させ、一方、CPU-B:12を使用する場合はCPU-B:12を先に起動させることができるものである。

【0073】

50

次に、本実施の形態におけるリセット方法をフローチャートにより説明する。

【0074】

本発明のリセット回路のリセット方法をフローチャートで示した図3と、図1および図2を併せて参照すると、まず、処理ステップS1においてCPU制御回路13がハードウェアリセットを検知する。

【0075】

ハードウェアリセットを検知すると、処理ステップS2においてデバッグ制御回路132が、内部の各機能ブロック1:14～機能ブロック3:16にリセットを発行する。

【0076】

次に処理ステップS2においてストラップ取込み回路131が、ストラップ等の外部設定情報を読み込む。 10

【0077】

処理ステップS3において、読み込んだストラップ情報に基づき、通常動作かデバッグ動作かを判断し、通常動作であれば、処理ステップS4においてデバッグ制御回路132が、読み込んだストラップ情報に基づきCPU-A:11およびCPU-B:12と各機能ブロック1:14～機能ブロック3:16のリセットを解除する。

【0078】

処理ステップS5においては、前の処理でのリセット解除を受けて各機能ブロック1:14～機能ブロック3:16はリセット後の動作を開始することになる。

【0079】

一方、この回路を用いてシステム起動時をデバッグするときの一連の動作は以下のとおりである。 20

【0080】

前述したように、処理ステップS1からS2でCPU制御回路13がハードウェアリセットを検知し、処理ステップS3でデバッグ制御回路11が内部の各機能ブロック14～17にリセットを発行すると、ストラップ取込み回路131は、ストラップ等外部設定情報を読み込む。

【0081】

処理ステップS3において、ストラップ等外部設定情報がデバッグを指定している、処理ステップS6でバグ処理を実行する。 30

【0082】

処理ステップS6では、まず、処理ステップS61では、デバッグ制御回路132によるデバッグの選定結果を判断する。判断結果がデバッガとしてCPU-B:12を選定している、処理ステップS62において、デバッグ制御回路132がストラップにより選択されたCPU-B:12のリセットを解除する。

【0083】

リセットを解除することにより、処理ステップS63でデバッガCPU-B:12が起動し、処理ステップS64で立ち上がり動作が完了すると、処理ステップS65でターゲット側CPU-A:11および各機能ブロックをリセットを解除する。

【0084】

処理ステップS61の判断結果がデバッガとしてCPU-A:11を選定している、処理ステップS66において、デバッグ制御回路132がストラップにより選択されたCPU-A:11のリセットを解除する。 40

【0085】

リセットを解除することにより、処理ステップS67でデバッガCPU-A:11が起動し、処理ステップS68で立ち上がり動作が完了すると、処理ステップS69でターゲット側CPU-B:12および各機能ブロックのリセットを解除する。

【0086】

上述した第1の実施の形態では、システムLSI内部の回路を用いて内蔵した複数のCPUを時間差をもって起動をさせることができ、システム立ち上がり時の動作を従来より詳 50

細に観測することができる。

【0087】

また、2つのCPUの機能動作を相互に入れ替えられるように制御回路を構成しているので、必要に応じて観測の範囲を変化させることが可能である。

【0088】

さらにこの回路を持ったシステムLSIにおいては、外部の測定機器を用いることなくシステムLSIのデバッグを進めることが可能になる。その理由は、内部でCPUによって必要な部分の観測を行うことができるためである。

【0089】

そしてこの制御回路は、システムLSIの回路規模縮小に寄与することができる。それは、観測用の回路を専用回路ではなく汎用CPUを用いているので、通常動作時に不要な回路をLSIに配置する必要がないからである。

【0090】

本発明の第2の実施の形態を説明する。

【0091】

本発明の第2の実施の形態におけるシステムLSIの主要構成を示した図6を参照すると、システムLSI21と、システムLSI21の共通バスラインと接続し外部に配置される外部メモリ22と、外部メモリ22に格納される、システムLSI21内部のCPU-A, C, D, Eのプログラム用メモリ23およびCPU-B用のデバッグプログラムを格納するメモリ24とで構成される。

【0092】

システムLSI1は、ターゲットとなるCPU-A:211と、トレース機能およびデバッグ機能を有するデバッガ手段として指定されるCPU-B:212と、複数のCPUを持つシステムLSI21のリセットおよび割込み等を制御するCPU制御回路216と、外部機器とインタフェースする機能ブロック1:217と、その他の機能ブロック2:218, 機能ブロック3:219と、これらの各CPU、CPU制御回路、各機能ブロックおよび外部メモリを相互に接続する共通バスライン220とから構成される。

【0093】

基本的構成は前述した第1の実施の形態と同様であるが、内蔵するCPUの個数が3個以上、ここでは例えば5個であっても回路の構成によって同等の効果が得られる。すなわち任意の個数のCPUにおいて、そのいずれか1つを先行して起動し、デバッガとして動作させることができる。

【0094】

また、当該回路の動作において、複数のCPUが互いに完全に独立した動作を行う場合については言うまでもないが、それ以外に、本発明においてデバッグ用プログラムが交換可能であることを利用して、複数のCPUが、互いに協調して動作を行う場合についても、デバッガCPUに相応の動作、つまり、複数のCPUに実際に協調動作をさせる機能あるいは疑似的に協調動作をさせる機能のプログラムを格納しておくことで、デバッグCPUにそのプログラムを実行させることもできる。

【0095】

【発明の効果】

上述したように、本発明のマルチCPUのリセット回路およびリセット方法は、複数のCPUを内蔵するシステムLSIに、リセット等を制御する手段を付加し、この手段によって、システムLSIに内蔵した複数のCPUのうちの一つの通常動作用のプログラムを、デバッグプログラムに置き換えることができる。

【0096】

また、従来は、LSIの外部端子を経由して内部・外部メモリ、レジスタ等の情報をLSI外部あるいはシステムの外部から観測したり、トレースのために専用の回路とトレース専用の内部メモリを用いて記録していたが、本発明では、これらを内蔵CPUの処理によって代替することを可能にし、また複数のCPUを切り替えて相互にデバッグが

出来るようにした。

【0097】

したがって、システムLSI内部の回路を用いて内蔵した複数のCPUを時間差をもって起動をさせることができ、システム立ち上がり時の動作を従来より詳細に観測することができる。

【0098】

また、2つのCPUの機能動作を相互に入れ替えられるように制御回路を構成しているので、必要に応じて観測の範囲を変化させることが可能である。

【0099】

さらに、内部CPUによって必要な部分の観測を行うことができるため外部の測定機器を用いることなくシステムLSIのデバッグを進めることができる。 10

【0100】

さらにまた、観測用の回路を専用回路ではなく汎用CPUを用いているので、通常動作時に不要な回路をLSIに配置する必要がなく、システムLSIの回路規模縮小に寄与することができる。

【図面の簡単な説明】

【図1】本発明の第1の実施の形態におけるシステムLSIの主要構成を示した図である。

【図2】図1におけるCPU制御回路の構成図である。

【図3】通常の起動状態における制御回路の動作説明用のタイミングチャートである。 20

【図4】ハードウェアリセット時における制御回路の動作説明用のタイミングチャートである。

【図5】CPU制御回路を含むシステムLSIのリセット方法を示したフローチャートである。

【図6】本発明の第2の実施の形態におけるシステムLSIの主要構成を示した図である。

【図7】従来のシステムLSIの一例の構成を示した図である。

【図8】従来の構成を備え、かつJTAGを利用したシステムLSIの一例を示した図である。

【図9】JTAGを利用し、かつROMシミュレータも利用したシステムLSIの一例を示した図である。 30

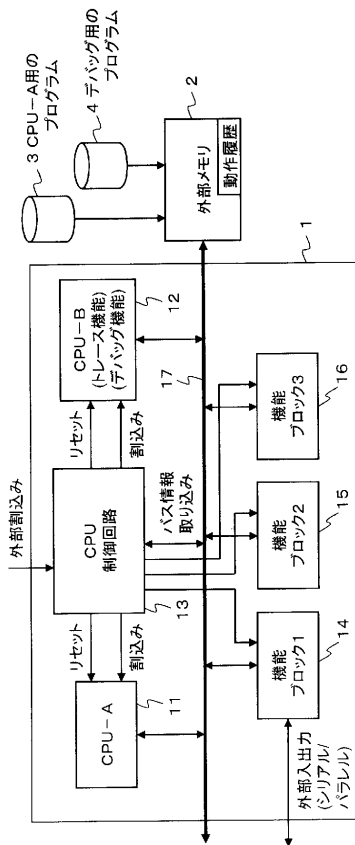
【図10】JTAGを利用したシステムLSIのさらに他の一例を示した図である。

【符号の説明】

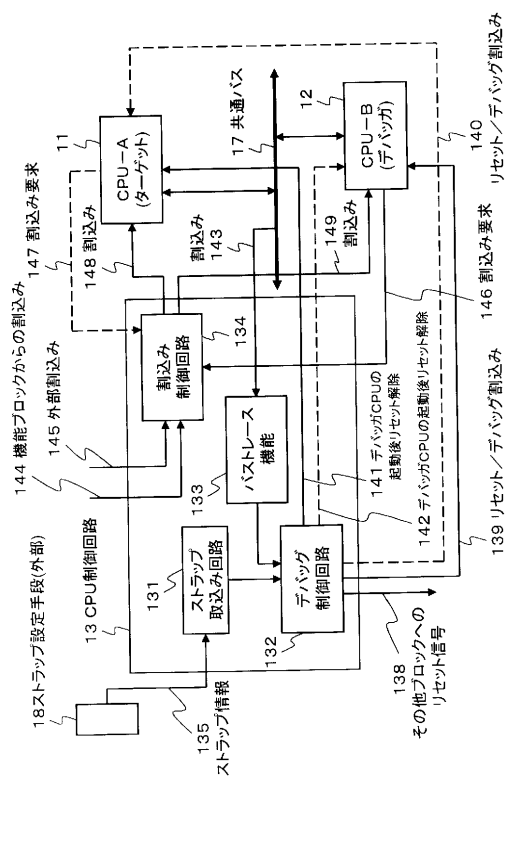
- 1 システムLSI
- 2 外部メモリ
- 3 CPU-A用のプログラム
- 4 デバッグ用のプログラム
- 11 CPU-A
- 12 CPU-B
- 13 CPU制御回路 40
- 14 機能ブロック1
- 15 機能ブロック2
- 16 機能ブロック3
- 17 共通バスライン
- 18 ストラップ情報設定手段
- 131 ストラップ取込み回路
- 132 デバッグ制御回路
- 133 バストレース機能用ブロック
- 134 割込み制御回路
- 139, 140 リセット/デバッグ割込み信号 50

141, 142 デバッガCPU起動後のリセット解除信号
 146, 147 割り込み要求信号

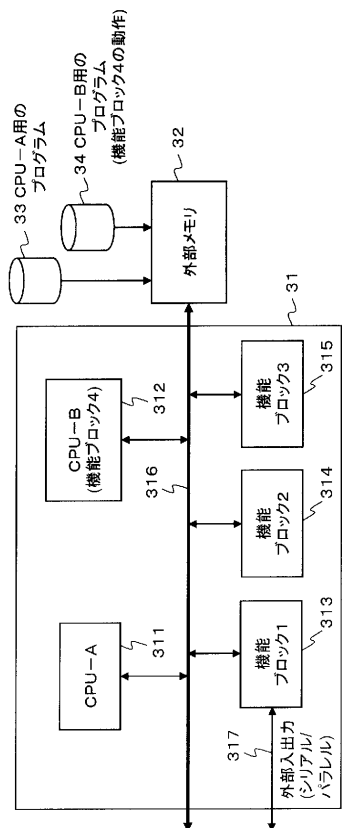
【図1】



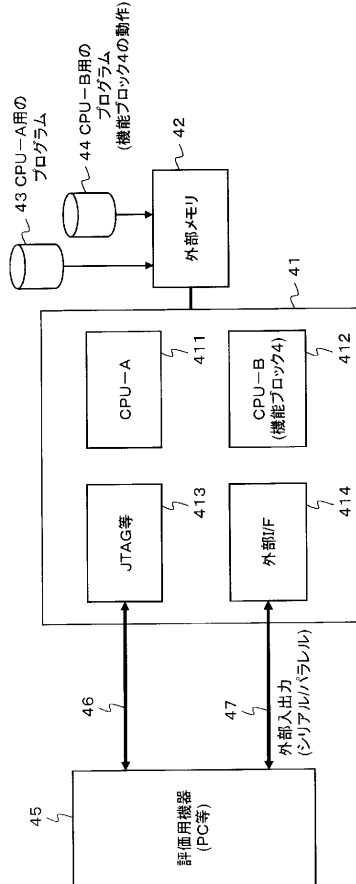
【図2】



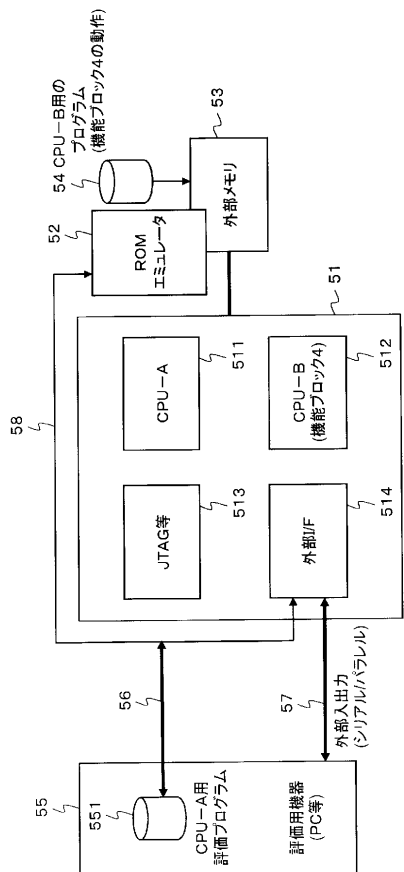
【図7】



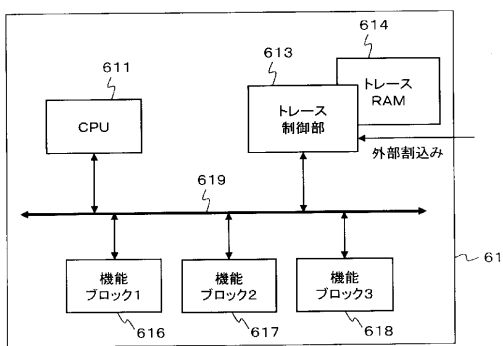
【図8】



【図9】



【図10】



フロントページの続き

(72)発明者 石川 幸司

神奈川県川崎市中原区小杉町一丁目403番53

ステム株式会社内

エヌイーシーマイクロシ

Fターム(参考) 5B042 GA11 GA13 HH01 HH25 LA18

5B045 FF03 HH04 JJ02 JJ49

5B048 AA12 AA20 DD10 FF03

5B098 AA10 BA05