

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4494899号
(P4494899)

(45) 発行日 平成22年6月30日(2010.6.30)

(24) 登録日 平成22年4月16日(2010.4.16)

(51) Int.Cl.		F I			
G06F 11/22	(2006.01)	G06F 11/22	340C		
G06F 15/78	(2006.01)	G06F 11/22	360P		
		G06F 15/78	510K		

請求項の数 4 (全 17 頁)

(21) 出願番号	特願2004-222398 (P2004-222398)	(73) 特許権者	000005223
(22) 出願日	平成16年7月29日(2004.7.29)		富士通株式会社
(65) 公開番号	特開2006-40172 (P2006-40172A)		神奈川県川崎市中原区上小田中4丁目1番1号
(43) 公開日	平成18年2月9日(2006.2.9)	(74) 代理人	100089118
審査請求日	平成18年10月27日(2006.10.27)		弁理士 酒井 宏明
前置審査		(72) 発明者	山下 英男
			神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
		(72) 発明者	菅 竜二
			神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
		審査官	久保 正典

最終頁に続く

(54) 【発明の名称】 プロセッサデバッグ装置およびプロセッサデバッグ方法

(57) 【特許請求の範囲】

【請求項1】

プロセッサ内のシフトレジスタをスキャンして読み出すプロセッサデバッグ装置であって、

プロセッサ内に設けられ、該プロセッサ内の所定の信号の値を複数のクロック分記憶するシフトレジスタと、

前記シフトレジスタにより記憶された信号値をスキャンして読み出す信号読出手段と、を備えたことを特徴とするプロセッサデバッグ装置。

【請求項2】

前記シフトレジスタに信号値を記憶するタイミングを指定するタイミング指定手段と、前記タイミング指定手段により指定されたタイミングを識別するタイミング値を前記シフトレジスタにより記憶される複数の信号のそれぞれに対応して記憶するタイミング記憶手段と、

をさらに備えたことを特徴とする請求項1に記載のプロセッサデバッグ装置。

【請求項3】

プロセッサ内の複数の信号から一つの信号を選択する信号選択手段をさらに備え、前記シフトレジスタは、前記信号選択手段により選択された信号の値を複数のクロック分記憶することを特徴とする請求項1または2に記載のプロセッサデバッグ装置。

【請求項4】

プロセッサ内のシフトレジスタをスキャンして読み出すプロセッサデバッグ方法であっ

10

20

て、

プロセッサ内に設けられたシフトレジスタに、該プロセッサ内の所定の信号の値が複数のクロック分保持されるように記録する信号記録工程と、

前記信号記録工程によりシフトレジスタに記録された信号値をスキャンして読み出す信号読出工程と、

を含んだことを特徴とするプロセッサデバッグ方法。

【発明の詳細な説明】

【技術分野】

【0001】

この発明は、プロセッサ内のラッチをスキャンして読み出すプロセッサデバッグ装置およびプロセッサデバッグ方法に関し、特に、プロセッサ内信号の時間的遷移の把握を可能とし、もってプロセッサのデバッグ効率を向上させることができるプロセッサデバッグ装置およびプロセッサデバッグ方法に関するものである。

10

【背景技術】

【0002】

近年、プロセッサの集積度は増加し、それに伴ってプロセッサ内の論理は複雑化してきている。論理が複雑化するにつれプロセッサのデバッグ作業は困難なものになり、その期間は増加する一方であるため、デバッグの効率化のためさまざまな工夫がなされている。

【0003】

そのひとつとして、JTAG (Join Test Action Group) 対応のプロセッサ内のラッチにスキャン回路を付加しそのラッチの情報を読み出す手法が広く用いられている (JTAG については、非特許文献1参照)。このスキャンの値を調査することでプロセッサの状態を外部から知ることが可能となり、デバッグの効率を上げている。しかし、このスキャンの調査では、ある1クロックタイミングの状態しか分からないため、プロセッサ内の信号の時間的遷移を知ることができず、信号の値の前後関係で発生するタイミング障害などについては解析が困難となっている。

20

【0004】

これを解決するため、プロセッサの動作を時間的に知ることができるよう、予め設計者が調査に重要な信号を各タイミングごとに専用のRAMに記録させるような回路をプロセッサ内に付加し、そのRAMに記録されている過去複数サイクル分の値を調査することも行われている。ここで、記録する情報量やクロック数は使用RAMの容量によるが、数100サイクル分の情報を記録させタイミング障害のデバッグに効果を発揮している。

30

【0005】

しかし、RAMは1Kビット、2Kビットなど記録する容量が決まっているため、設計者が実際に必要である信号数以上にデバッグ機構を実装することとなり、また、RAM自身の動作を試験するために診断回路を付加しなくてはならず、回路の増大、設計工数の増大、ひいてはコストの増大を招く。

【0006】

また、RAMは専用の微細なトランジスタで構成されているためロジックと比べ製造上問題を発生させやすく、デバッグ機能のためだけにRAMを実装することはLSIの歩留まりを低下させる要因となる。

40

【0007】

従って、このデバッグ用のRAMは、命令制御部やメモリ制御部など100サイクルにわたる情報がデバッグに必要な箇所においてのみ使用されている。そのため、デバッグに数100サイクルもの情報が必要でない演算実行部などではこのRAMは実装せず、ラッチのスキャン情報のみでデバッグを行っている。

【0008】

【非特許文献1】IEEE1349.1、「Test Access Port and Boundary Scan Architecture」、IEEE規格

【発明の開示】

50

【発明が解決しようとする課題】

【0009】

しかしながら、スキャン情報のみではその1タイミングの状況のみで演算実行部の動作を推測するしかないため、演算実行部内回路の時間的遷移状況は知ることができず、演算実行部のデバッグ作業が困難であるという問題があった。

【0010】

この発明は、上述した従来技術による問題点を解消するためになされたものであり、RAMを用いることなくプロセッサ内信号の時間的遷移の把握を可能とし、もってプロセッサのデバッグ効率を向上させることができるプロセッサデバッグ装置およびプロセッサデバッグ方法を提供することを目的とする。

10

【課題を解決するための手段】

【0011】

上述した課題を解決し、目的を達成するため、本発明は、プロセッサ内のシフトレジスタをスキャンして読み出すプロセッサデバッグ装置であって、プロセッサ内に設けられ、該プロセッサ内の所定の信号の値を複数のクロック分記憶するシフトレジスタと、前記シフトレジスタにより記憶された信号値をスキャンして読み出す信号読出手段と、を備えたことを特徴とする。

【0012】

また、本発明は、プロセッサ内のシフトレジスタをスキャンして読み出すプロセッサデバッグ方法であって、プロセッサ内に設けられたシフトレジスタに、該プロセッサ内の所定の信号の値が複数のクロック分保持されるように記録する信号記録工程と、前記信号記録工程によりシフトレジスタに記録された信号値をスキャンして読み出す信号読出工程と、を含んだことを特徴とする。

20

【0013】

かかる発明によれば、プロセッサ内の所定の信号の値を複数のクロック分プロセッサ内のシフトレジスタに保持されるように記録し、シフトレジスタにより記録した信号値をスキャンして読み出すよう構成したので、設計者は、プロセッサ内の信号の時間的遷移を把握することができる。

【発明の効果】

【0014】

本発明によれば、設計者は、プロセッサ内の信号の時間的遷移を把握することができるので、プロセッサのデバッグを効率良く行うことができるという効果を奏する。

30

【発明を実施するための最良の形態】

【0015】

以下に添付図面を参照して、この発明に係るプロセッサデバッグ装置およびプロセッサデバッグ方法の好適な実施例を詳細に説明する。

【実施例1】

【0016】

まず、本実施例1に係るプロセッサデバッグ装置による信号記録方式について図1-1~図1-3を用いて説明する。図1-1は、信号Aの過去6サイクル分の情報を記録するシフトレジスタ("B0"~"B5")を示す図である。本実施例1に係るプロセッサデバッグ装置では、このようなシフトレジスタを用いて、プロセッサ内の信号Aを記録する。

40

【0017】

今、この信号Aの値をスキャンで読み出したタイミングを"TIM 0"とすると、レジスタB0には"TIM 0"での信号Aの値が、レジスタB1には"TIM 0"の1クロック前のタイミングである"TIM-1"での信号Aの値が、レジスタB2には"TIM 0"の2クロック前のタイミングである"TIM-2"での信号Aの値が、レジスタB3、"B4"、"B5"にはそれぞれ3、4、5クロック前での信号Aの値が記録されている。

【0018】

例として、ある回路において信号Aが図1-2に示すように"TIM-9"から"TIM 0"までの

50

10クロックの間変化していたとする。このとき、シフトレジスタが記録している値は図1-3に示す通りである。

【0019】

従って、本実施例1に係るプロセッサデバッグ装置では、設計者は、シフトレジスタの値を調査することで図1-2に示した"TIM-5"から"TIM 0"までの6サイクル間の信号Aの状態を知ることができ、デバッグを効率良く行うことができる。

【0020】

次に、本実施例1に係るプロセッサの構成について説明する。図2は、本実施例1に係るプロセッサの構成を示す機能ブロック図である。同図に示すように、このプロセッサ100は、命令制御ユニット110と、データ制御ユニット120と、演算ユニット130とを有する。

10

【0021】

命令制御ユニット110は、命令の実行を制御する制御部であり、演算ユニット130に対して3ビットの信号OPCODEによって処理する演算を指示する。図3は、3ビットのOPCODE割り当ての一例を示す図である。同図に示すように、ここでは、"nop"から"SHIFT_RIGHT"までの8種類の演算を3ビットのOPCODEに割り当てている。

【0022】

データ制御ユニット120は、演算ユニット130で演算されるデータに関する制御を行う制御部であり、演算ユニット130に演算データを出力する。

【0023】

演算ユニット130は、データ制御ユニット120から受け取った演算データに対して命令制御ユニット110からOPCODEで指示された演算を実行する処理部であり、演算制御部131と、演算実行部132と、デバッグ機構133とを有する。

20

【0024】

演算制御部131は、命令制御ユニット110からの制御信号により演算実行部132を制御する制御部である。演算実行部132は、データ制御ユニット120からの演算データに対して演算制御部131からの制御信号に従って演算を行う処理部である。

【0025】

デバッグ機構133は、プロセッサ100のデバッグを支援する機能を提供する機構である。具体的には、このデバッグ機構133は、演算制御部131からOPCODEを入力してシフトレジスタ134にその値を記録する。図4は、デバッグ機構133の構成を示す機能ブロック図である。同図に示すように、このデバッグ機構133は、シフトレジスタ134とスキャン部135とを有する。

30

【0026】

シフトレジスタ134は、OPCODEの過去6サイクル分の値を記憶するシフトレジスタである。すなわち、レジスタ"C0"は最新のOPCODEを記憶し、レジスタ"C1"は1クロック前のOPCODEを記憶し、レジスタ"C2"は2クロック前のOPCODEを記憶する。以下同様に、シフトレジスタ134は、5クロック前までのOPCODEの値を記憶する。

【0027】

今、図5に示すタイミングで命令制御ユニット110から演算ユニット130へOPCODEが送られていたとする。このときのデバッグ機構133内のシフトレジスタ134に保持されているOPCODEの値とそのタイミングは図6に示す通りである。図6に示すように、シフトレジスタ134には、"TIM 0" ~ "TIM-5"までの過去6サイクル分のOPCODEが記録される。

40

【0028】

このように、シフトレジスタ134に過去6サイクル分のOPCODEの値を記録することによって、設計者は、過去6サイクルの間、SUB -> DIV -> MULT -> NOP -> DIV -> ADDの順序で命令制御ユニット110から演算制御信号が送られていたことを知ることができ、デバッグを効率良く行うことができる。

【0029】

50

スキャン部 135 は、シフトレジスタ 134 に記憶された値をスキャンして読み出す処理部である。このスキャン部 135 がシフトレジスタ 134 に記憶された値をスキャンして読み出すことによって、設計者は、過去 6 サイクル分の OP CODE を知ることができる。

【0030】

上述してきたように、本実施例 1 では、デバッグ機構 133 がシフトレジスタ 134 に過去 6 サイクル分の OP CODE を記憶し、スキャン部 135 がシフトレジスタ 134 に記憶された OP CODE をスキャンして読み出すこととしたので、設計者は、OP CODE の時間的遷移を把握することができ、プロセッサのデバッグを効率良く行うことができる。

【0031】

また、このような構成をとることにより、わずかな回路を追加するだけで、製造上の歩留まりや信頼性を低下させることなく、プロセッサ内回路の信号の時間的遷移を外部から知ることを可能とし、プロセッサのデバッグ作業の効率を向上させることができる。

【0032】

なお、本実施例 1 では、シフトレジスタが 6 段構成である場合について説明したが、他の段数のシフトレジスタを使用することもできる。また、OP CODE 以外の他の信号をシフトレジスタに記録することもできる。

【実施例 2】

【0033】

ところで、上記実施例 1 では、過去 6 サイクル分の OP CODE を全てシフトレジスタ 134 に記録するプロセッサデバッグ装置について説明した。しかしながら、全てのサイクルの OP CODE ではなく、特定のサイクルの OP CODE だけを知りたい場合もある。

【0034】

例えば、図 7 は、信号 A とその有効性を示す信号 V を示す図である。同図において、信号 V の値が "1" である時のみ信号 A の値は有効であるとする。この場合、タイミング "TIM-2"、"TIM-4"、"TIM-6" および "TIM-7" での信号 A は無効であるため、デバッグ機構のシフトレジスタにこのタイミングの信号 A の値を記録しても意味を持たない。

【0035】

従って、限られたデバッグ機構の資源を有効に使用するため、信号 V の値が "1" であるときにのみ信号 A の値を記録することが考えられる。そこで、本実施例 2 では、ある信号の値が "1" であるときにのみ特定の信号の値をシフトレジスタ記録するプロセッサデバッグ装置について説明する。

【0036】

まず、本実施例 2 に係るプロセッサデバッグ装置による信号記録方式について図 8 - 1 ~ 図 8 - 3 を用いて説明する。図 8 - 1 は、信号 A の過去 6 サイクル分の情報を記録するシフトレジスタ ("B10" ~ "B15") を示す図である。

【0037】

ただし、このシフトレジスタ ("B10" ~ "B15") は、シフトレジスタ ("B0" ~ "B5") と異なり、信号 V が "1" である時のみ信号 A を記録するように制御される。また、本実施例 2 に係るプロセッサデバッグ装置では、毎サイクルサイクリックにカウントする 4 ビットのカウンタを新たに設け、このカウンタ値も信号 A が記録される同じタイミングでシフトレジスタ ("B20" ~ "B25") に記録する。

【0038】

このように、信号 A が記録される同じタイミングでカウンタ値をシフトレジスタ ("B20" ~ "B25") に記録することによって、カウンタ値を用いて信号 A が記録されたタイミングを相対的に特定することができる。

【0039】

図 7 に示した信号 A、V の状態の変化にカウンタの値を追記した例を図 8 - 2 に示す。このカウンタの値は信号 A、V のタイミングを相対的に示すものであり、今、"TIM 0" のタイミングで値が H'C' (16 進数の 'C') であったとする。このとき、デバッグ機構の各レジスタが記録している信号 A とカウンタの値、記録されているタイミングは図 8 - 3 に

10

20

30

40

50

示す通りである。

【0040】

設計者は、シフトレジスタ("B10"~"B15")に記録されている信号Aの値と、同じタイミングで記録されているシフトレジスタ("B20"~"B25")のカウント値を調査することで、図8-2で示す信号Aの時間的遷移について知ることができる。また、設計者は、"TIM-9"から"TIM 0"までの10サイクル間の状態を知ることができるため、より効率良くデバッグを行うことができる。

【0041】

次に、本実施例2に係るプロセッサの構成について説明する。図9は、本実施例2に係るプロセッサの構成を示す機能ブロック図である。なお、ここでは説明の便宜上、図2に示した各部と同様の役割を果たす機能部については同一符号を付すこととしてその詳細な説明を省略する。

10

【0042】

図9に示すように、このプロセッサ200は、命令制御ユニット210と、データ制御ユニット120と、演算ユニット230とを有する。命令制御ユニット210は、命令制御ユニット110と同様、命令の実行を制御する制御部であるが、演算ユニット230に対してOPCODEに加えてREQUEST_VALIDを出力する。ここで、REQUEST_VALIDは、OPCODEの有効性を示す信号であり、このREQUEST_VALIDが"1"のときのみOPCODEは有効となる。

【0043】

演算ユニット230は、演算ユニット130と同様、演算を実行する処理部であり、演算制御部231と、演算実行部232と、デバッグ機構233とを有する。演算制御部231は、命令制御ユニット210からOPCODEに加えてREQUEST_VALIDを制御信号として受け取り、演算実行部232を制御する制御部である。演算実行部232は、データ制御ユニット120からの演算データに対して演算制御部231からの制御信号に従って演算を行う処理部である。

20

【0044】

デバッグ機構233は、プロセッサ200のデバッグを支援する機能を提供する機構であり、演算制御部231からOPCODEに加えてREQUEST_VALIDを入力し、REQUEST_VALIDが"1"のときのOPCODEをシフトレジスタ234に記録する。

【0045】

図10は、デバッグ機構233の構成を示す機能ブロック図である。同図に示すように、このデバッグ機構233は、シフトレジスタ234および235と、カウンタ236と、スキャン部237とを有する。

30

【0046】

シフトレジスタ234は、OPCODEとREQUEST_VALIDを入力し、REQUEST_VALIDが"1"のときのみOPCODEの値を記憶する。カウンタ236は、毎サイクルサイクリックに'H'0'~'H'F'をカウントする4ビットのカウンタであり、OPCODEをシフトレジスタ234に記録したタイミングを特定するために使用される。

【0047】

シフトレジスタ235は、REQUEST_VALIDが"1"のとき、すなわちOPCODEがシフトレジスタ234に記録されたタイミングでのカウンタ236の値を記憶する。スキャン部237は、シフトレジスタ234および235に記憶された値をスキャンして読み出す処理部である。

40

【0048】

今、図11に示すタイミングで命令制御ユニット210から演算ユニット233へOPCODEおよびREQUEST_VALIDが送られていたとする。このときのデバッグ機構233内のシフトレジスタ234および235に保持されているOPCODEおよびカウンタの値とそのタイミングは図12に示す通りである。

【0049】

図12に示すように、シフトレジスタ234には、REQUEST_VALIDが"1"である"TIM 0"

50

、"TIM-1"、"TIM-3"、"TIM-5"、"TIM-8"および"TIM-9"のOPCODEが記録され、シフトレジスタ235には、"TIM 0"、"TIM-1"、"TIM-3"、"TIM-5"、"TIM-8"および"TIM-9"のカウンタ値が記録される。

【0050】

従って、設計者は、デバッグ機構233のシフトレジスタ234とシフトレジスタ235に記録された値を調査することで、過去10サイクル中の6サイクルで、ADD -> SQRT -> SUB -> MULT -> DIV -> ADDの順序で命令制御ユニット210から有効な演算制御信号が送られ、演算実行部232で演算が行われていたことを知ることができ、さらに効率良くデバッグを行うことができる。

【0051】

上述してきたように、本実施例2では、デバッグ機構233がOPCODEに加えてREQUEST_VALIDを入力し、REQUEST_VALIDが"1"であるときのみOPCODEの値をシフトレジスタ234に記憶することとしたので、シフトレジスタ234には有効な情報だけが記憶され、シフトレジスタ234を効率良く使用することができる。

【実施例3】

【0052】

実施例1および2では、一つのプロセッサ内に一つの演算ユニットがある場合について説明したが、スーパースカラ方式のプロセッサなどでは、一つのプロセッサ内に複数の演算ユニットがある。そこで、本実施例3では、一つのプロセッサ内に二つの演算ユニットがある場合について説明する。なお、ここでは、各演算ユニットごとに信号を記録するシフトレジスタを備えることとする。

【0053】

まず、本実施例3に係るプロセッサデバッグ装置による信号記録方式について図13-1～図13-3を用いて説明する。今、図13-1に示すように、シフトレジスタXおよびYにおいて、それぞれ信号XA、YAをそれらが有効であるXV、YVが"1"であるときのみシフトレジスタ("B30"～"B35")および("B50"～"B55")へ記録させることとする。また、両シフトレジスタで共通のカウンタを設けその値もXA、YAが記録された同じタイミングでシフトレジスタ("B40"～"B45")および("B60"～"B65")へ記録するように制御する。

【0054】

例として、図13-2に示すように、信号X、Y、XVおよびYVが"TIM-9"から"TIM 0"での10クロックの間変化していたとし、"TIM 0"のタイミングでのカウンタ値が'H'C'であったとする。この時の各シフトレジスタが記録している信号XA、YAとカウンタの値は図13-3の通りである。

【0055】

このように、シフトレジスタ("B30"～"B35")、("B40"～"B45")、("B50"～"B55")および("B60"～"B65")に記録された値を調査することで、設計者は、図13-2での信号XA、YAの時間的遷移について、信号XAについては"TIM-9"から"TIM 0"までの10サイクル間の状態を知ることができ、信号YAについては"TIM-8"から"TIM 0"までの9サイクル間の状態を知ることができるとともに、両演算実行部での演算の同時実行状態も知ることができる。

【0056】

次に、本実施例3に係るプロセッサの構成について説明する。図14は、本実施例3に係るプロセッサの構成を示す機能ブロック図である。なお、ここでは説明の便宜上、図2に示した各部と同様の役割を果たす機能部については同一符号を付すこととしてその詳細な説明を省略する。

【0057】

図14に示すように、このプロセッサ300は、命令制御ユニット310と、データ制御ユニット320と、演算ユニットX330と、演算ユニットY340と、デバッグ機構350とを有する。

10

20

30

40

50

【 0 0 5 8 】

命令制御ユニット 3 1 0 は、命令の実行を制御する制御部であり、演算ユニット X 3 3 0 に対して OPCODE_X および REQUEST_VALID_X を出力し、演算ユニット Y 3 4 0 に対して OPCODE_Y および REQUEST_VALID_Y を出力する。

【 0 0 5 9 】

ここで、OPCODE_X および OPCODE_Y は、演算を指示する 3 ビットの信号であり、その割り当ては、図 3 に示した 3 ビットの OPCODE の割り当てと同じである。また、REQUEST_VALID_X は、OPCODE_X の有効性を示す信号であり、この REQUEST_VALID_X が "1" のときのみ OPCODE_X は有効となり、REQUEST_VALID_Y は、OPCODE_Y の有効性を示す信号であり、この REQUEST_VALID_Y が "1" のときのみ OPCODE_Y は有効となる。

10

【 0 0 6 0 】

データ制御ユニット 3 2 0 は、演算ユニット X 3 3 0 および演算ユニット Y 3 4 0 で演算されるデータに関する制御を行う制御部であり、演算ユニット X 3 3 0 および演算ユニット Y 3 4 0 に演算データを出力する。

【 0 0 6 1 】

演算ユニット X 3 3 0 は、データ制御ユニット 3 2 0 から受け取った演算データに対して命令制御ユニット 3 1 0 から指示された演算を実行する処理部であり、演算制御部 3 3 1 と、演算実行部 3 3 2 とを有する。

【 0 0 6 2 】

演算制御部 3 3 1 は、命令制御ユニット 3 1 0 から OPCODE_X および REQUEST_VALID_X を受け取り、演算実行部 3 3 2 を制御する制御部である。演算実行部 3 3 2 は、データ制御ユニット 3 2 0 からの演算データに対して演算制御部 3 3 1 からの制御信号に従って演算を行う処理部である。なお、演算ユニット Y 3 4 0 も、演算ユニット X 3 3 0 と同様の機能構成を有する。

20

【 0 0 6 3 】

デバッグ機構 3 5 0 は、プロセッサ 3 0 0 のデバッグを支援する機能を提供する機構であり、演算制御部 3 3 1 から OPCODE_X および REQUEST_VALID_X を入力し、REQUEST_VALID_X が "1" であるときのみ OPCODE_X の値をシフトレジスタ 3 5 1 に記録し、演算制御部 3 4 1 から OPCODE_Y および REQUEST_VALID_Y を入力し、REQUEST_VALID_Y が "1" であるときのみ OPCODE_Y の値をシフトレジスタ 3 5 3 に記録する。

30

【 0 0 6 4 】

図 1 5 は、デバッグ機構 3 5 0 の構成を示す機能ブロック図である。同図に示すように、このデバッグ機構 3 5 0 は、シフトレジスタ 3 5 1 ~ 3 5 4 と、カウンタ 3 5 5 と、スキャン部 3 5 6 とを有する。

【 0 0 6 5 】

シフトレジスタ 3 5 1 は、OPCODE_X と REQUEST_VALID_X を入力し、REQUEST_VALID_X が "1" のときのみ OPCODE_X の値を記憶する。シフトレジスタ 3 5 3 は、OPCODE_Y と REQUEST_VALID_Y を入力し、REQUEST_VALID_Y が "1" のときのみ OPCODE_Y の値を記憶する。

【 0 0 6 6 】

カウンタ 3 5 5 は、毎サイクルサイクリックにカウント H'0' ~ H'F' をカウントする 4 ビットのカウンタであり、OPCODE_X または OPCODE_Y を記録したタイミングを特定するために使用される。

40

【 0 0 6 7 】

シフトレジスタ 3 5 2 は、REQUEST_VALID_X が "1" のとき、すなわち OPCODE_X がシフトレジスタ 3 5 1 に記録されたタイミングでのカウンタ 3 5 5 の値を記憶する。シフトレジスタ 3 5 4 は、REQUEST_VALID_Y が "1" のとき、すなわち OPCODE_Y がシフトレジスタ 3 5 3 に記録されたタイミングでのカウンタ 3 5 5 の値を記憶する。スキャン部 3 5 6 は、シフトレジスタ 3 5 1 ~ 3 5 4 に記憶された値をスキャンして読み出す処理部である。

【 0 0 6 8 】

今、図 1 6 に示すタイミングで命令制御ユニット 3 1 0 から、演算ユニット X 3 3 0 へ

50

OPCODE_XおよびREQUEST_VALID_Xが送られ、演算ユニットY 3 4 0へOPCODE_YおよびREQUEST_VALID_Yが送られていたとする。このときのデバッグ機構3 5 0内のシフトレジスタ3 5 1および3 5 2に保持されているOPCODE_Xおよびカウンタの値と、シフトレジスタ3 5 3および3 5 4に保持されているOPCODE_Yおよびカウンタの値と、それらのタイミングは図1 7に示す通りである。

【0 0 6 9】

図1 7に示すように、シフトレジスタ3 5 1には、REQUEST_VALID_Xが"1"である"TIM 0"、"TIM-1"、"TIM-3"、"TIM-5"、"TIM-8"および"TIM-9"のOPCODE_Xが記録され、シフトレジスタ3 5 2には、"TIM 0"、"TIM-1"、"TIM-3"、"TIM-5"、"TIM-8"および"TIM-9"のカウンタ値が記録される。また、シフトレジスタ3 5 3には、REQUEST_VALID_Yが"1"である"TIM 0"、"TIM-2"、"TIM-4"、"TIM-5"、"TIM-6"および"TIM-8"のOPCODE_Yが記録され、シフトレジスタ3 5 4には、"TIM 0"、"TIM-2"、"TIM-4"、"TIM-5"、"TIM-6"および"TIM-8"のカウンタ値が記録される。

10

【0 0 7 0】

従って、設計者は、デバッグ機構3 5 0のシフトレジスタ3 5 1および3 5 2の値を調査することで、演算ユニットX 3 3 0において過去1 0サイクルの間、ADD -> SQRT -> SUB -> MULT -> DIV -> ADDの順序で命令制御ユニット3 1 0から有効なOPCODE_Xが送られ、演算実行部3 3 2で演算が行われていたことを知ることができ、シフトレジスタ3 5 3および3 5 4の値を調査することで、演算ユニットY 3 4 0において過去9サイクルの間、ADD -> MULT -> SHIFT_LEFT -> ADD -> MLT -> SHIFT_LEFTの順序で命令制御ユニット3 1 0から有効なOPCODE_Yが送られ、演算実行部3 4 2で演算が行われていたことを知ることができる。

20

【0 0 7 1】

また、シフトレジスタ3 5 2および3 5 4で記録しているカウンタ値は演算ユニットX 3 3 0および演算ユニットY 3 4 0で共通なものであるため、設計者は、これらに記録されている値より両演算ユニットで処理した演算のタイミングを知ることができ、デバッグの効率をさらに向上することができる。例えば、設計者は、"TIM 0"、"TIM-5"および"TIM-8"においては演算ユニットX 3 3 0と演算ユニットY 3 4 0が同時に演算を行っていたことを知ることができる。

【0 0 7 2】

上述してきたように、本実施例3では、デバッグ機構3 5 0が二つの演算ユニットで行われていた演算を記録し、両演算ユニットで処理された演算のタイミングを設計者が知ることを可能としたので、デバッグの効率をさらに向上することができる。

30

【0 0 7 3】

なお、本実施例3では、プロセッサが二つの演算ユニットを有する場合について説明したが、デバッグ機構内のシフトレジスタの個数を増やすことによって、より多くの演算ユニットを有するプロセッサを対象とすることもできる。

【実施例4】

【0 0 7 4】

実施例1～3では、プロセッサ内の決められた信号の時間的遷移を調査する場合について説明したが、プロセッサ内の複数の信号のうちいずれかの信号を選択してその時間的遷移を調査したい場合もある。そこで、本実施例4では、プロセッサ内の二つの信号のうちいずれかの信号を選択してシフトレジスタに記録する場合について説明する。

40

【0 0 7 5】

図1 8は、本実施例4に係るプロセッサの構成を示す機能ブロック図である。なお、ここでは説明の便宜上、図2に示した各部と同様の役割を果たす機能部については同一符号を付すこととしてその詳細な説明を省略する。

【0 0 7 6】

同図に示すように、このプロセッサ4 0 0の演算ユニット4 3 0は、演算制御部1 3 1と、演算実行部4 3 2と、デバッグ機構4 3 3とを有する。

50

【 0 0 7 7 】

演算実行部 4 3 2 は、演算制御部 1 3 1 からの制御信号に従って演算を行う処理部であり、デバッグ機構 4 3 3 に対して RUPT_CODE を出力する。ここで、RUPT_CODE は、演算結果の状態を示す 3 ビットの信号である。

【 0 0 7 8 】

デバッグ機構 4 3 3 は、プロセッサ 4 0 0 のデバッグを支援する機能を提供する機構であり、選択信号に基づいて OPCODE または RUPT_CODE のいずれかをシフトレジスタに記録する。ここで、選択信号は、スキャン可能なラッチの出力であり、プロセッサ 4 0 0 の外部から設定可能である。

【 0 0 7 9 】

図 1 9 は、デバッグ機構 4 3 3 の構成を示す機能ブロック図である。同図に示すように、このデバッグ機構 4 3 3 は、シフトレジスタ 4 3 4 と、セクタ 4 4 0 と、スキャン部 1 3 5 とを有する。

【 0 0 8 0 】

シフトレジスタ 4 3 4 は、OPCODE または RUPT_CODE の過去 6 サイクル分の値を記憶するシフトレジスタである。セクタ 4 4 0 は、OPCODE、RUPT_CODE および選択信号を入力し、選択信号に基づいて OPCODE または RUPT_CODE を選択し、シフトレジスタ 4 3 4 に出力する処理部である。すなわち、シフトレジスタ 4 3 4 は、セクタ 4 4 0 によって選択された OPCODE または RUPT_CODE の過去 6 サイクル分の値を記憶する。

【 0 0 8 1 】

上述してきたように、本実施例 4 では、セクタ 4 4 0 が OPCODE または RUPT_CODE を選択し、シフトレジスタ 4 3 4 がセクタ 4 4 0 によって選択された OPCODE または RUPT_CODE の過去 6 サイクル分の値を記憶することとしたので、記録可能な信号の種類をわずかなハードウェアの増加で増やすことができる。

【 0 0 8 2 】

なお、本実施例 4 では、REQUEST_VALID を使用しない場合について説明したが、REQUEST_VALID を使用し、REQUEST_VALID の値が "1" であるときのみ OPCODE または RUPT_CODE をシフトレジスタに記録することもできる。

【 0 0 8 3 】

また、本実施例 1 ~ 4 に係るデバッグ機構に記録させる信号数とクロック数、カウンタのビット幅は設計者がプロセッサ内に搭載できる回路量とその必要性から任意に決定できるため、このデバッグ機構の回路規模は、RAM で構成されたものが 1 K、2 K ビットなどある単位でしか実装できないのに比べ、設計者が必要な最低限で実現することができる。

【 0 0 8 4 】

(付記 1) プロセッサ内のラッチをスキャンして読み出すプロセッサデバッグ装置であって、

プロセッサ内の所定の信号の値を複数のクロック分記憶するレジスタと、
前記レジスタにより記憶された信号値をスキャンして読み出す信号読出手段と、
を備えたことを特徴とするプロセッサデバッグ装置。

【 0 0 8 5 】

(付記 2) 前記レジスタに信号値を記憶するタイミングを指定するタイミング指定手段と、

前記タイミング指定手段により指定されたタイミングを識別するタイミング値を前記レジスタにより記憶される複数の信号のそれぞれに対応して記憶するタイミング記憶手段と、

をさらに備えたことを特徴とする付記 1 に記載のプロセッサデバッグ装置。

【 0 0 8 6 】

(付記 3) プロセッサ内の複数の信号から一つの信号を選択する信号選択手段をさらに備え、

10

20

30

40

50

前記レジスタは、前記信号選択手段により選択された信号の値を複数のクロック分記憶することを特徴とする付記 1 または 2 に記載のプロセッサデバッグ装置。

【0087】

(付記 4) 前記レジスタは、シフトレジスタであることを特徴とする付記 1 または 2 に記載のプロセッサデバッグ装置。

【0088】

(付記 5) 前記タイミング記憶手段により記憶されるタイミング値は、カウンタによりカウントされるカウント値であることを特徴とする付記 2 に記載のプロセッサデバッグ装置。

【0089】

(付記 6) プロセッサ内のラッチをスキャンして読み出すプロセッサデバッグ方法であって、

プロセッサ内の所定の信号の値が複数のクロック分レジスタに保持されるように記録する信号記録工程と、

前記信号記録工程によりレジスタに記録された信号値をスキャンして読み出す信号読出工程と、

を含んだことを特徴とするプロセッサデバッグ方法。

【0090】

(付記 7) 前記信号記録工程は、指定されたタイミングでレジスタに信号値を記録し、前記タイミングを識別するタイミング値を前記レジスタにより記録される複数の信号のそれぞれに対応して記録するタイミング記録工程、

をさらに含んだことを特徴とする付記 6 に記載のプロセッサデバッグ方法。

【0091】

(付記 8) プロセッサ内の複数の信号から一つの信号を選択する信号選択工程をさらに含み、

前記信号記録工程は、前記信号選択工程により選択された信号の値を複数のクロック分レジスタに記録することを特徴とする付記 6 または 7 に記載のプロセッサデバッグ方法。

【0092】

(付記 9) 前記レジスタは、シフトレジスタであることを特徴とする付記 6 または 7 に記載のプロセッサデバッグ方法。

【0093】

(付記 10) 前記タイミング記録工程により記録されるタイミング値は、カウンタによりカウントされるカウント値であることを特徴とする付記 7 に記載のプロセッサデバッグ方法。

【産業上の利用可能性】

【0094】

以上のように、本発明に係るプロセッサデバッグ装置およびプロセッサデバッグ方法は、プロセッサの試験に有用であり、特に、信号の値の前後関係で発生するタイミング障害の解析などに適している。

【図面の簡単な説明】

【0095】

【図 1 - 1】本実施例 1 に係るプロセッサデバッグ装置による信号記録方式を説明するための説明図 (信号を記録するシフトレジスタ) である。

【図 1 - 2】本実施例 1 に係るプロセッサデバッグ装置による信号記録方式を説明するための説明図 (記録される信号) である。

【図 1 - 3】本実施例 1 に係るプロセッサデバッグ装置による信号記録方式を説明するための説明図 (シフトレジスタに記録される信号) である。

【図 2】本実施例 1 に係るプロセッサの構成を示す機能ブロック図である。

【図 3】3 ビットの OPCODE 割り当ての一例を示す図である。

【図 4】デバッグ機構の構成を示す機能ブロック図である。

10

20

30

40

50

【図 5】OPCODEの各タイミングでの値の例を示す図である。

【図 6】シフトレジスタに記録されたOPCODEを示す図である。

【図 7】信号 A とその有効性を示す信号 V を示す図である。

【図 8 - 1】本実施例 2 に係るプロセッサデバッグ装置による信号記録方式を説明するための説明図（信号を記録するシフトレジスタとカウンタ値を記録するシフトレジスタ）である。

【図 8 - 2】本実施例 2 に係るプロセッサデバッグ装置による信号記録方式を説明するための説明図（記録される信号とその有効性を示す信号）である。

【図 8 - 3】本実施例 2 に係るプロセッサデバッグ装置による信号記録方式を説明するための説明図（二つのシフトレジスタにそれぞれ記録される信号およびカウンタ値）である。

10

【図 9】本実施例 2 に係るプロセッサの構成を示す機能ブロック図である。

【図 10】デバッグ機構の構成を示す機能ブロック図である。

【図 11】OPCODEおよびREQUEST_VALIDの各タイミングでの値の例を示す図である。

【図 12】二つのシフトレジスタにそれぞれ記録されたOPCODEおよびカウンタ値を示す図である。

【図 13 - 1】本実施例 3 に係るプロセッサデバッグ装置による信号記録方式を説明するための説明図（二つの信号およびカウンタ値のそれぞれを記録する四つのシフトレジスタ）である。

【図 13 - 2】本実施例 3 に係るプロセッサデバッグ装置による信号記録方式を説明するための説明図（記録される二つの信号とそれらの有効性をそれぞれ示す二つの信号）である。

20

【図 13 - 3】本実施例 3 に係るプロセッサデバッグ装置による信号記録方式を説明するための説明図（四つのシフトレジスタにそれぞれ記録される信号）である。

【図 14】本実施例 3 に係るプロセッサの構成を示す機能ブロック図である。

【図 15】デバッグ機構の構成を示す機能ブロック図である。

【図 16】OPCODE_X、OPCODE_Y、REQUEST_VALID_XおよびREQUEST_VALID_Yの各タイミングでの値の例を示す図である。

【図 17】四つのシフトレジスタにそれぞれ記録されたOPCODE_X、カウンタ値、OPCODE_Yおよびカウンタ値を示す図である。

30

【図 18】本実施例 4 に係るプロセッサの構成を示す機能ブロック図である。

【図 19】デバッグ機構の構成を示す機能ブロック図である。

【符号の説明】

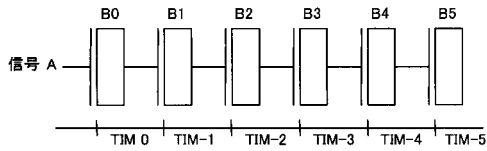
【0096】

100, 200, 300, 400 プロセッサ
 110, 210, 310 命令制御ユニット
 120, 320 データ制御ユニット
 130, 230, 430 演算ユニット
 131, 231, 331, 341 演算制御部
 132, 232, 332, 342, 432 演算実行部
 133, 233, 350, 433 デバッグ機構
 134, 234, 351, 353, 434 シフトレジスタ
 135, 237, 356 スキャン部
 235, 352, 354 シフトレジスタ
 236, 355 カウンタ
 330 演算ユニット X
 340 演算ユニット Y

40

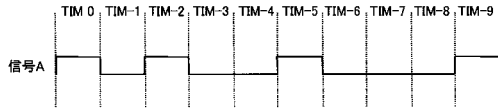
【図 1 - 1】

本実施例1に係るプロセッサデバッグ装置による信号記録方式を説明するための説明図(信号を記録するシフトレジスタ)



【図 1 - 2】

本実施例1に係るプロセッサデバッグ装置による信号記録方式を説明するための説明図(記録される信号)



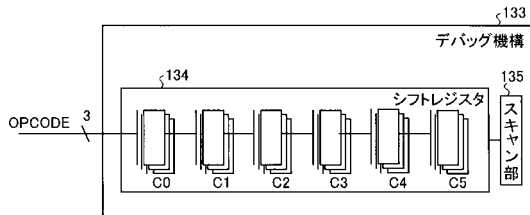
【図 1 - 3】

本実施例1に係るプロセッサデバッグ装置による信号記録方式を説明するための説明図(シフトレジスタに記録される信号)

	信号A	記録タイミング
レジスタ(B0)	1	TIM 0
レジスタ(B1)	0	TIM-1
レジスタ(B2)	1	TIM-2
レジスタ(B3)	0	TIM-3
レジスタ(B4)	0	TIM-4
レジスタ(B5)	1	TIM-5

【図 4】

デバッグ機構の構成を示す機能ブロック図



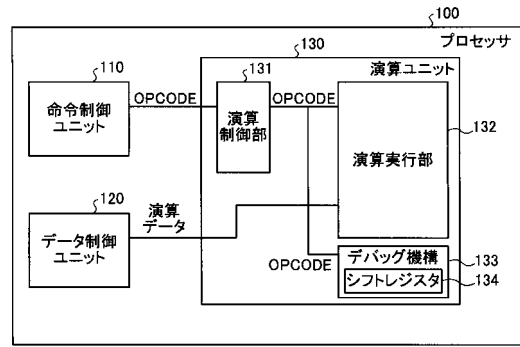
【図 5】

OP CODEの各タイミングでの値の例を示す図

	TIM 0	TIM-1	TIM-2	TIM-3	TIM-4	TIM-5	TIM-6	TIM-7	TIM-8	TIM-9
OP CODE	001	100	000	011	100	010	001	011	101	001

【図 2】

本実施例1に係るプロセッサの構成を示す機能ブロック図



【図 3】

3ビットのOP CODE割り当ての一例を示す図

OP CODE<3:0>	
000	nop
001	ADD
010	SUB
011	MULT
100	DIV
101	SQRT
110	SHIFT_LEFT
111	SHIFT_RIGHT

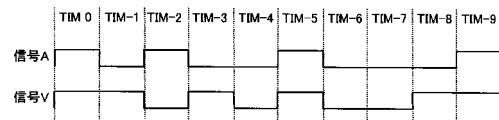
【図 6】

シフトレジスタに記録されたOP CODEを示す図

	OP CODE	記録タイミング
レジスタ(C0)	001 (ADD)	TIM 0
レジスタ(C1)	100 (DIV)	TIM-1
レジスタ(C2)	000 (NOP)	TIM-2
レジスタ(C3)	011 (MULT)	TIM-3
レジスタ(C4)	100 (DIV)	TIM-4
レジスタ(C5)	010 (SUB)	TIM-5

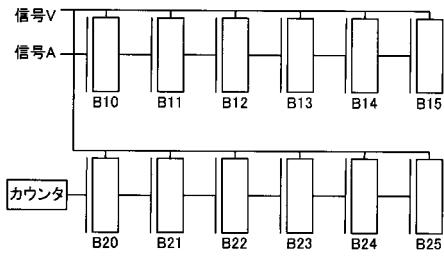
【図 7】

信号Aとその有効性を示す信号Vを示す図



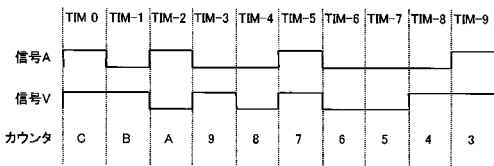
【図 8 - 1】

本実施例2に係るプロセッサデバッグ装置による信号記録方式を説明するための説明図(信号を記録するシフトレジスタとカウンタ値を記録するシフトレジスタ)



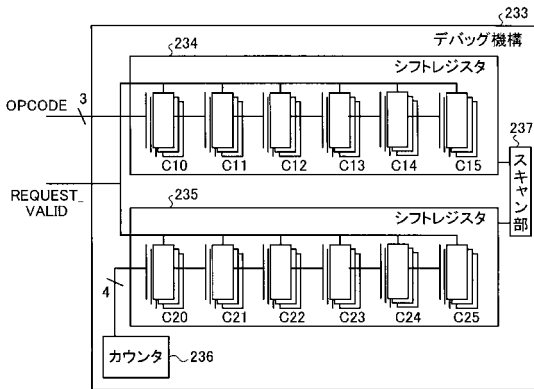
【図 8 - 2】

本実施例2に係るプロセッサデバッグ装置による信号記録方式を説明するための説明図(記録される信号とその有効性を示す信号)



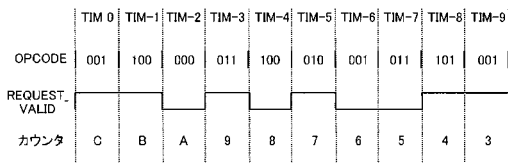
【図 10】

デバッグ機構の構成を示す機能ブロック図



【図 11】

OPCODEおよびREQUEST_VALIDの各タイミングでの値の例を示す図



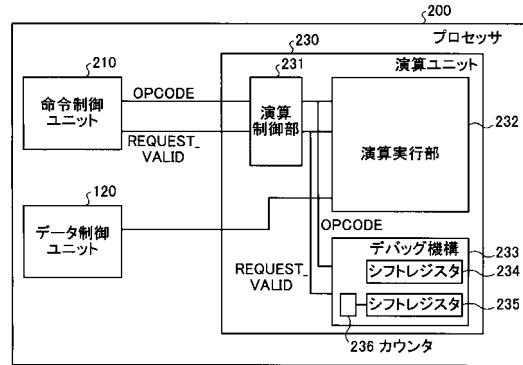
【図 8 - 3】

本実施例2に係るプロセッサデバッグ装置による信号記録方式を説明するための説明図(二つのシフトレジスタにそれぞれ記録される信号およびカウンタ値)

	信号A		カウンタ値	記録タイミング
レジスタ(B10)	1	レジスタ(B20)	C	TIM-0
レジスタ(B11)	0	レジスタ(B21)	B	TIM-1
レジスタ(B12)	0	レジスタ(B22)	9	TIM-3
レジスタ(B13)	1	レジスタ(B23)	7	TIM-5
レジスタ(B14)	0	レジスタ(B24)	4	TIM-8
レジスタ(B15)	1	レジスタ(B25)	3	TIM-9

【図 9】

本実施例2に係るプロセッサの構成を示す機能ブロック図



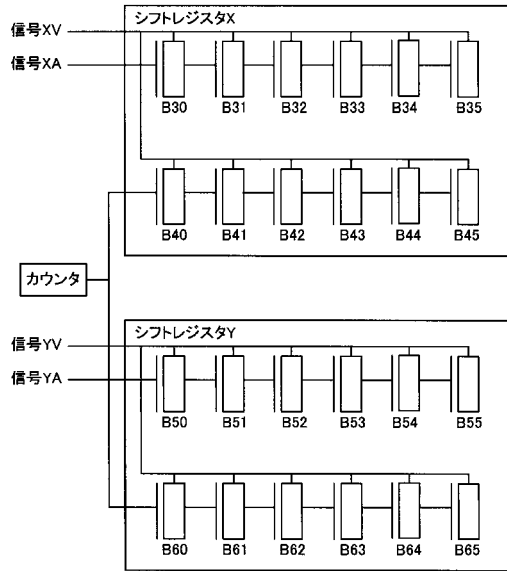
【図 12】

二つのシフトレジスタにそれぞれ記録されたOPCODEおよびカウンタ値を示す図

	OPCODE		カウンタ値	記録タイミング
レジスタ(C10)	001 (ADD)	レジスタ(C20)	C	TIM-0
レジスタ(C11)	100 (DIV)	レジスタ(C21)	B	TIM-1
レジスタ(C12)	011 (MULT)	レジスタ(C22)	9	TIM-3
レジスタ(C13)	010 (SUB)	レジスタ(C23)	7	TIM-5
レジスタ(C14)	101 (SQRT)	レジスタ(C24)	4	TIM-8
レジスタ(C15)	001 (ADD)	レジスタ(C25)	3	TIM-9

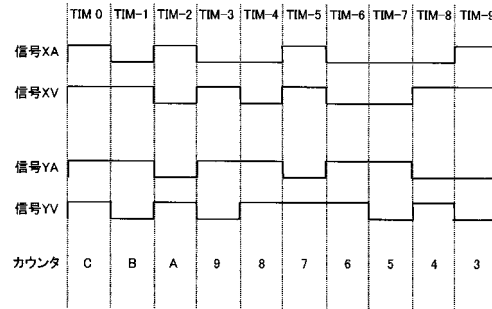
【図13-1】

本実施例3に係るプロセッサデバッグ装置による信号記録方式を説明するための説明図(二つの信号およびカウンタ値のそれぞれを記録する四つのシフトレジスタ)



【図13-2】

本実施例3に係るプロセッサデバッグ装置による信号記録方式を説明するための説明図(記録される二つの信号とそれらの有効性をそれぞれ示す二つの信号)



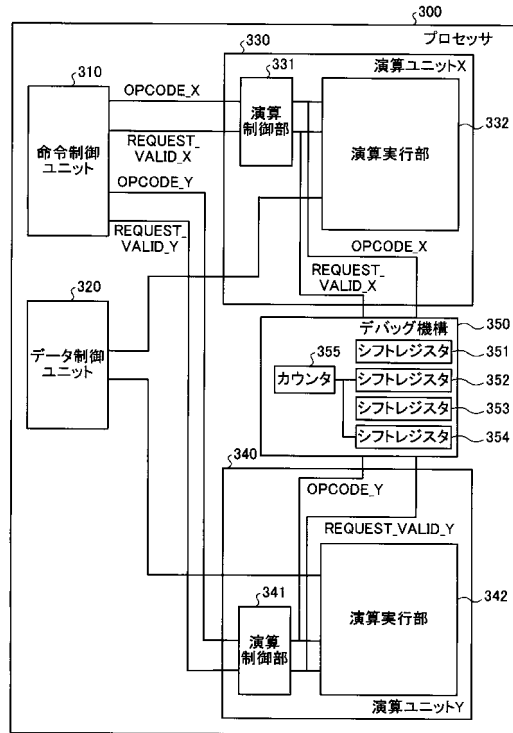
【図13-3】

本実施例3に係るプロセッサデバッグ装置による信号記録方式を説明するための説明図(四つのシフトレジスタにそれぞれ記録される信号)

信号XA		カウンタ値		記録タイミング
レジスタ(B30)	1	レジスタ(B40)	C	TIM 0
レジスタ(B31)	0	レジスタ(B41)	B	TIM-1
レジスタ(B32)	0	レジスタ(B42)	9	TIM-3
レジスタ(B33)	1	レジスタ(B43)	7	TIM-5
レジスタ(B34)	0	レジスタ(B44)	4	TIM-8
レジスタ(B35)	1	レジスタ(B45)	3	TIM-9
信号YA		カウンタ値		記録タイミング
レジスタ(B50)	1	レジスタ(B60)	C	TIM 0
レジスタ(B51)	0	レジスタ(B61)	A	TIM-2
レジスタ(B52)	1	レジスタ(B62)	8	TIM-4
レジスタ(B53)	0	レジスタ(B63)	7	TIM-5
レジスタ(B54)	1	レジスタ(B64)	6	TIM-6
レジスタ(B55)	0	レジスタ(B65)	4	TIM-8

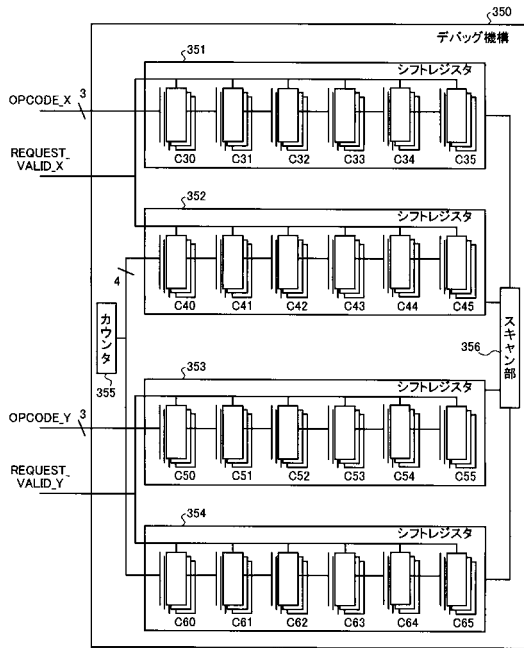
【図14】

本実施例3に係るプロセッサの構成を示す機能ブロック図



【図15】

デバッグ機構の構成を示す機能ブロック図



【図16】

OPCODE_X、OPCODE_Y、REQUEST_VALID_XおよびREQUEST_VALID_Yの各タイミングでの値の例を示す図

	TIM 0	TIM-1	TIM-2	TIM-3	TIM-4	TIM-5	TIM-6	TIM-7	TIM-8	TIM-9
OPCODE_X	001	100	000	011	100	010	001	011	101	001
REQUEST_VALID_X										
OPCODE_Y	110	010	011	100	001	110	011	010	001	100
REQUEST_VALID_Y										
カウンタ	C	B	A	9	8	7	6	5	4	3

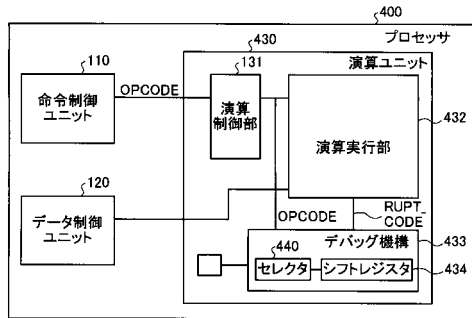
【図17】

四つのシフトレジスタにそれぞれ記録されたOPCODE_X、カウンタ値、OPCODE_Yおよびカウンタ値を示す図

	OPCODE_X	カウンタ値	記録タイミング
レジスタ(C30)	001 (ADD)	C	TIM 0
レジスタ(C31)	100 (DIV)	B	TIM-1
レジスタ(C32)	011 (MULT)	9	TIM-3
レジスタ(C33)	010 (SUB)	7	TIM-5
レジスタ(C34)	101 (SQRT)	4	TIM-8
レジスタ(C35)	001 (ADD)	3	TIM-9
	OPCODE_Y	カウンタ値	記録タイミング
レジスタ(C50)	110 (SHIFT_LEFT)	C	TIM 0
レジスタ(C51)	011 (MULT)	A	TIM-2
レジスタ(C52)	001 (ADD)	8	TIM-4
レジスタ(C53)	110 (SHIFT_LEFT)	7	TIM-5
レジスタ(C54)	011 (MULT)	6	TIM-6
レジスタ(C55)	001 (ADD)	4	TIM-8

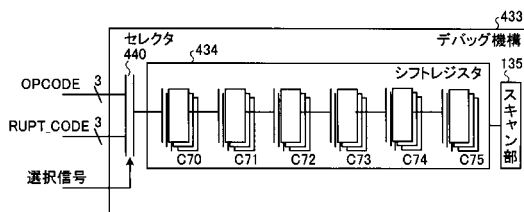
【図18】

本実施例4に係るプロセッサの構成を示す機能ブロック図



【図19】

デバッグ機構の構成を示す機能ブロック図



フロントページの続き

(56)参考文献 特開平08-171505(JP,A)
特開昭60-159951(JP,A)
特開2002-149442(JP,A)

(58)調査した分野(Int.Cl., DB名)
G06F11/22-11/36
G01R31/28-31/3193
G06F15/78