(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第6438353号
(P6438353)

(45)発行日　平成30年12月12日(2018.12.12)　(24)登録日　平成30年11月22日(2018.11.22)

(51)Int.Cl.		F I		
G06F 11/22	(2006.01)	G06F	11/22	675
G06F 11/267	(2006.01)	G06F	11/267	
G01R 31/28	(2006.01)	G01R	31/28	V

請求項の数 18　(全 53 頁)

(21)出願番号　特願2015-107472(P2015-107472)
(22)出願日　平成27年5月27日(2015.5.27)
(65)公開番号　特開2016-224531(P2016-224531A)
(43)公開日　平成28年12月28日(2016.12.28)
審査請求日　平成29年11月27日(2017.11.27)

(73)特許権者　302062931
ルネサスエレクトロニクス株式会社
東京都江東区豊洲三丁目２番２４号
(74)代理人　100103894
弁理士　家入　健
(72)発明者　芝原　真一
東京都小平市上水本町五丁目２０番１号
ルネサスシステムデザイン株式会社内
(72)発明者　川上　大輔
神奈川県川崎市中原区下沼部１７５３番地
ルネサスエレクトロニクス株式会社内
(72)発明者　井学　豊
神奈川県川崎市中原区下沼部１７５３番地
ルネサスエレクトロニクス株式会社内

最終頁に続く

(54)【発明の名称】半導体装置及び診断テスト方法

(57)【特許請求の範囲】
【請求項１】
記憶回路と、
前記記憶回路に格納されたデータを利用して処理を実行するとともに、処理の実行に応じて前記記憶回路にデータを書き込む処理回路と、
前記処理回路が処理を実行していないときに、前記処理回路に対するスキャンテストを実行するスキャンテスト回路と、
前記処理回路に対するスキャンテストを実行しているときに、前記処理回路から前記記憶回路に対するデータの書き込みを抑止する抑止回路と、を備え、
前記処理回路は、
キャッシュメモリを有し、前記処理を実行する複数の演算回路と、
前記複数の演算回路間で前記キャッシュメモリのキャッシュコヒーレンシを保証する制御を行う第１のコヒーレンシ制御回路を有する共通回路と、を含み、
前記記憶回路は、前記第１のコヒーレンシ制御回路が前記キャッシュコヒーレンシの保証に利用する、前記複数の演算回路のそれぞれのキャッシュメモリに関する管理情報が格納される管理情報記憶回路を含み、
前記抑止回路は、前記共通回路に対するスキャンテストを実行しているときに、前記第１のコヒーレンシ制御回路に対し、前記管理情報記憶回路に対するデータの書き込みを抑止する、
半導体装置。

【請求項２】
前記半導体装置は、前記処理回路に対するスキャンテストの実行後に、前記処理回路をリセットするリセットコントローラをさらに備え、
前記処理回路は、当該処理回路をリセットしたときに、前記記憶回路のデータを初期化する処理を実行するものであり、
前記抑止回路は、前記スキャンテストの実行後に前記処理回路をリセットしたとき、前記処理回路に対し前記記憶回路のデータ初期化処理の実行を抑止する、
請求項１に記載の半導体装置。
【請求項３】
前記半導体装置は、前記処理回路に対するスキャンテストの実行後に、前記処理回路をリセットするリセットコントローラをさらに備え、
前記処理回路は、当該処理回路をリセットしたときに、前記記憶回路のデータを初期化する処理を実行するものであり、
前記抑止回路は、前記処理回路から前記記憶回路に出力されるデータの書き込みのための信号を遮断することで、前記処理回路から前記記憶回路に対するデータの書き込みを抑止し、さらに、前記スキャンテストの実行後に前記処理回路をリセットしたときに、前記記憶回路のデータを初期化するための前記処理回路から前記記憶回路に対するデータの書き込みを抑止する、
請求項１に記載の半導体装置。
【請求項４】
前記記憶回路は、キャッシュメモリを含み、
前記処理回路は、
前記処理を実行する演算回路と、
前記演算回路が利用するデータの前記キャッシュメモリからの読み出しと、前記処理回路の処理に応じてデータの前記キャッシュメモリへの書き込みを制御するキャッシュコントローラを有する共通回路と、を含み、
前記抑止回路は、前記共通回路に対するスキャンテストを実行しているときに、前記キャッシュコントローラに対し、前記キャッシュメモリに対するデータの書き込みを抑止する、
請求項１に記載の半導体装置。
【請求項５】
前記処理回路は、前記処理を実行する演算回路を含み、
前記演算回路は、割り込みに応じて起床するスリープ状態に移行することで、前記処理を中断し、
前記半導体装置は、さらに、
前記演算回路に割り込みを通知する割り込みコントローラと、
前記割り込みコントローラから前記演算回路に対する割り込みを遮断する割り込みマスク回路と、を備え、
前記演算回路は、前記スリープ状態に移行する前に、前記割り込みの遮断を行うように前記割り込みマスク回路に通知する、
請求項１に記載の半導体装置。
【請求項６】
記憶回路と、
前記記憶回路に格納されたデータを利用して処理を実行するとともに、処理の実行に応じて前記記憶回路にデータを書き込む処理回路と、
前記処理回路が処理を実行していないときに、前記処理回路に対するスキャンテストを実行するスキャンテスト回路と、
前記処理回路に対するスキャンテストを実行しているときに、前記処理回路から前記記憶回路に対するデータの書き込みを抑止する抑止回路と、を備えた半導体装置であって、
前記処理回路は、

前記処理を実行する複数の演算回路と、
前記複数の演算回路によって共有される共通回路と、を含み、
前記複数の演算回路のそれぞれは、スリープ状態に移行することで、前記処理を中断し、
前記半導体装置は、前記共通回路に対するスキャンテストの実行後に、回路をリセットするリセット信号を、前記複数の演算回路及び前記共通回路のそれぞれに出力するリセットコントローラと、
前記共通回路に対するスキャンテストの実行後に前記複数の演算回路及び前記共通回路のそれぞれに出力されるリセット信号のうち、前記複数の演算回路に出力されるリセット信号を遮断するリセットマスク回路と、をさらに備えた、
半導体装置。
【請求項７】
記憶回路と、
前記記憶回路に格納されたデータを利用して処理を実行するとともに、処理の実行に応じて前記記憶回路にデータを書き込む処理回路と、
前記処理回路が処理を実行していないときに、前記処理回路に対するスキャンテストを実行するスキャンテスト回路と、
前記処理回路に対するスキャンテストを実行しているときに、前記処理回路から前記記憶回路に対するデータの書き込みを抑止する抑止回路と、を備え、
前記処理回路は、
前記処理を実行する複数の演算回路と、
前記複数の演算回路によって共有される共通回路と、
前記複数の演算回路のそれぞれに対応し、前記演算回路に対するスキャンテストを実行しているときに、前記演算回路に入力される信号及び前記演算回路から出力される信号を遮断する複数の第１のラッパ回路と、
前記共通回路に対するスキャンテストを実行しているときに、前記共通回路に入力される信号及び前記共通回路から出力される信号を遮断する第２のラッパ回路と、を含む、
半導体装置。
【請求項８】
前記半導体装置は、前記スキャンテストの実行時間が所定時間を超えた場合にタイムアウトを通知するタイマをさらに備え、
前記処理回路は、前記処理回路に対するスキャンテストを開始するときに前記タイマを設定し、前記処理回路に対するスキャンテストが終了したときに前記タイマを解除する、
請求項１に記載の半導体装置。
【請求項９】
前記半導体装置は、
複数の前記処理回路と、
前記複数の処理回路間で前記キャッシュメモリのキャッシュコヒーレンシを保証する制御を行う第２のコヒーレンシ制御回路と、を備え、
前記複数の処理回路のそれぞれは、前記共通回路に対するスキャンテストを実行しているときに、前記キャッシュコヒーレンシを保証するために、前記第２のコヒーレンシ制御回路から前記第１のコヒーレンシ制御回路に出力される信号を遮断するマスク回路を含む、
請求項１に記載の半導体装置。
【請求項１０】
記憶回路と、
前記記憶回路に格納されたデータを利用して処理を実行するとともに、処理の実行に応じて前記記憶回路にデータを書き込む処理回路と、
前記処理回路が処理を実行していないときに、前記処理回路に対するスキャンテストを実行するスキャンテスト回路と、

前記処理回路に対するスキャンテストを実行しているときに、前記処理回路から前記記憶回路に対するデータの書き込みを抑止する抑止回路と、を備えた半導体装置であって、
前記処理回路は、前記処理を実行する複数の演算回路を含み、
前記スキャンテスト回路は、前記複数の演算回路の一からの指示に応じて、当該演算回路のスキャンテストを実行するものであり、
前記複数の演算回路の一は、他の演算回路からの割り込みに応じて前記スキャンテスト回路にスキャンテストの実行を指示し、当該指示に応じたスキャンテストの実行後に他の演算回路に割り込みを通知し、
前記スキャンテスト回路は、前記演算回路全体のスキャンテストに使用する複数のテストパタンを分割した複数の単位のそれぞれについてのスキャンテストを分割スキャンテストとして実行する毎に、前記演算回路に対するスキャンテストの実行を終了するものであり、
前記半導体装置は、前記演算回路の分割スキャンテストを終了してからの時間を計測し、当該演算回路の次の分割スキャンテストを開始するタイミングを前記演算回路に通知するタイマをさらに備え、
前記演算回路は、前記タイマからの通知に応じて、前記スキャンテスト回路に前記分割スキャンテストの開始を指示する、
半導体装置。
【請求項１１】
記憶回路と、
前記記憶回路に格納されたデータを利用して処理を実行するとともに、処理の実行に応じて前記記憶回路にデータを書き込む処理回路と、
前記処理回路が処理を実行していないときに、前記処理回路に対するスキャンテストを実行するスキャンテスト回路と、
前記処理回路に対するスキャンテストを実行しているときに、前記処理回路から前記記憶回路に対するデータの書き込みを抑止する抑止回路と、を備えた半導体装置であって、
前記処理回路は、
前記処理を実行する複数の演算回路と、
前記複数の演算回路によって共有される共通回路と、を含み、
前記スキャンテスト回路は、前記演算回路からのスキャンテストの指示に応じて、前記共通回路のスキャンテストを実行するものであり、
前記複数の演算回路の一は、他の全ての演算回路のそれぞれに割り込みを通知するとともに、前記スキャンテスト回路にスキャンテストの実行を指示してから、スリープ状態に移行することで、前記処理を中断し、
前記他の全ての演算回路のそれぞれは、前記複数の演算回路の一からの割り込みに応じて、スリープ状態に移行することで、前記処理を中断し、
前記スキャンテスト回路は、前記共通回路を分割した複数の領域のそれぞれについて前記スキャンテストを実行する毎に、前記共通回路に対するスキャンテストの実行を中断するものであり、
前記半導体装置は、前記スキャンテストを中断してから再開するまでの時間を計測し、前記スキャンテストを再開するタイミングを前記演算回路に通知するタイマをさらに備え、
前記演算回路は、前記タイマからの通知に応じて、前記スキャンテスト回路に前記スキャンテストの再開を指示する、
半導体装置。
【請求項１２】
前記処理回路は、前記処理を実行する演算回路を含み、
前記演算回路は、スリープ状態に移行することで、前記処理を中断し、
前記半導体装置は、
前記演算回路をリセットするリセットコントローラと、

前記演算回路がスリープ状態に移行したときに、前記演算回路の電源をＯＦＦにし、割り込みが通知されたときに、前記演算回路の電源をＯＮにして前記リセットコントローラによって前記演算回路をリセットする電源制御回路と、をさらに備え、
前記演算回路は、前記演算回路に対するスキャンテストを実行する場合には、前記演算回路の電源ＯＦＦを抑止するように前記電源制御回路を設定してから、スリープ状態に移行することで、前記処理を中断し、
前記スキャンテスト回路は、前記演算回路に対するスキャンテストの実行後に、前記電源制御回路に割り込みを通知する、
請求項１に記載の半導体装置。
【請求項１３】
前記スキャンテスト回路は、前記半導体装置の外部に備えられた外部記憶回路に格納されたテストパタンを取得し、取得したテストパタンを前記処理回路にスキャンインすることで前記スキャンテストを実行する、
請求項１に記載の半導体装置。
【請求項１４】
前記処理回路は、前記処理を実行するＣＰＵを含み、
前記ＣＰＵは、
キャッシュメモリと、
前記ＣＰＵが利用するデータの前記キャッシュメモリからの読み出しと、前記処理回路の処理に応じてデータの前記キャッシュメモリへの書き込みを制御するキャッシュコントローラと、を有し、
前記抑止回路は、前記ＣＰＵに対するスキャンテストを実行しているときに、前記キャッシュコントローラから前記キャッシュメモリに対するデータの書き込みを抑止する、
請求項１に記載の半導体装置。
【請求項１５】
前記処理回路は、前記処理を実行するＣＰＵを含み、
前記半導体装置は、
前記ＣＰＵの処理を補助的に実行するハードウェアアクセラレータと、
時間の経過を計測し、前記ハードウェアアクセラレータに対するスキャンテストを実行するタイミングを前記ＣＰＵに通知するタイマをさらに備え、
前記ＣＰＵは、前記タイマからの通知に応じて、前記ハードウェアアクセラレータに対するスキャンテストの実行を指示し、
前記スキャンテスト回路は、さらに、前記ＣＰＵからの指示に応じて、前記ハードウェアアクセラレータに対するスキャンテストを実行する、
請求項１に記載の半導体装置。
【請求項１６】
前記処理回路は、前記処理を実行する複数の演算回路を含み、
前記複数の演算回路のうち、予め定められた少なくとも１つの演算回路は、電源ＯＮ時にＰＯＳＴを実行せず、他の演算回路は、電源ＯＮ時にＰＯＳＴを実行し、
前記スキャンテスト回路は、前記ＰＯＳＴを実行しない演算回路がブートしたときに、当該演算回路のスキャンテストを実行する、
請求項１に記載の半導体装置。
【請求項１７】
記憶回路に格納されるデータを利用して処理を実行するとともに、処理の実行に応じて前記記憶回路にデータを書き込む処理回路が処理を実行していないときに、前記処理回路に対するスキャンテストを実行し、
前記処理回路に対するスキャンテストを実行しているときに、前記処理回路から前記記憶回路に対するデータの書き込みを抑止する、診断テスト方法であって、
前記処理回路は、
キャッシュメモリを有し、前記処理を実行する複数の演算回路と、

前記複数の演算回路間で前記キャッシュメモリのキャッシュコヒーレンシを保証する制御を行う第１のコヒーレンシ制御回路を有する共通回路と、を含み、
前記記憶回路は、前記第１のコヒーレンシ制御回路が前記キャッシュコヒーレンシの保証に利用する、前記複数の演算回路のそれぞれのキャッシュメモリに関する管理情報が格納される管理情報記憶回路を含み、
前記共通回路に対するスキャンテストを実行しているときに、前記第１のコヒーレンシ制御回路に対し、前記管理情報記憶回路に対するデータの書き込みを抑止する、
診断テスト方法。
【請求項１８】
記憶回路と、
前記記憶回路に格納されたデータを利用して処理を実行するとともに、処理の実行に応じて前記記憶回路にデータを書き込む処理回路と、
前記処理回路が処理を実行していないときに、前記処理回路に対するスキャンテストを実行するスキャンテスト回路と、
前記処理回路に対するスキャンテストを実行しているときに、前記処理回路から前記記憶回路に対して出力されるデータの書き込みのための信号を遮断することによって、前記処理回路から前記記憶回路に対するデータの書き込みを抑止する抑止回路と、
前記処理回路に対するスキャンテストの実行後に、前記処理回路をリセットするリセットコントローラと、を備え、
前記処理回路は、
キャッシュメモリを有し、前記処理を実行する複数の演算回路と、
前記複数の演算回路間で前記キャッシュメモリのキャッシュコヒーレンシを保証する制御を行う第１のコヒーレンシ制御回路を有する共通回路と、を含み、
前記記憶回路は、前記第１のコヒーレンシ制御回路が前記キャッシュコヒーレンシの保証に利用する、前記複数の演算回路のそれぞれのキャッシュメモリに関する管理情報が格納される管理情報記憶回路を含み、
前記抑止回路は、前記共通回路に対するスキャンテストを実行しているときに、前記第１のコヒーレンシ制御回路に対し、前記管理情報記憶回路に対するデータの書き込みを抑止し、
前記処理回路は、当該処理回路がリセットされたときに、前記記憶回路のデータを初期化する処理を実行し、
前記抑止回路は、前記スキャンテストの実行後にリセットされたときに、前記記憶回路のデータを初期化するために前記処理回路から前記記憶回路に対して出力されるデータの書き込みのための信号を遮断するか、前記記憶回路のデータを初期化するために前記処理回路から前記記憶回路に対するデータの書き込みを抑止するための指示信号を出力し、前記スキャンテストの終了後に前記記憶回路の内容を保持し、
前記処理回路は、前記スキャンテストの終了後に、前記記憶回路の内容を利用して処理を継続する、
半導体装置。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、半導体装置及び診断テスト方法に関し、例えば記憶回路にアクセスする処理回路の診断テストを実行する技術に関する。
【背景技術】
【０００２】
機能安全をサポートする半導体装置では、回路が正しく動作しているか否かを定期的に診断する機構が必要とされている。診断対象の回路が論理規模の小さいＣＰＵ（Central Processing Unit）である場合には、ロックステップを実装することによって、ＣＰＵが正しく動作しているか否かを診断することができる。ロックステップは、複数の同一のＣ

ＰＵを半導体装置に搭載して、全てのＣＰＵが同じ動作をしているか否かを判定することで、回路が正しく動作しているか否かを診断する機構である。よって、診断対象の回路が論理規模の大きいＣＰＵである場合に、ＣＰＵの実装数が増大してしまうため、回路面積の制約上、ロックステップを実装することができないという問題がある。

【０００３】

ロックステップ以外に回路が正しく動作しているか否かを診断する機構として、ソフトウェアによるセルフテストも考えられる。しかしながら、ソフトウェアによるセルフテストは、一般的にソフトウェアの開発コストが大きいという問題がある。そのため、特許文献１に開示されるように、回路が正しく動作しているか否かを診断する機構として、ＢＩＳＴ（Built In Self Test）が採用されている。

【０００４】

しかし、ＢＩＳＴに代表されるスキャンテストを実行した場合には、テスト対象の処理回路がアクセスする記憶回路に格納されるデータが書き換えられてしまうという問題がある。例えば、スキャンテストの実行後には、テスト対象の処理回路を再び動作可能とするために処理回路をリセットする必要があるが、そのリセットに伴って処理回路が記憶回路に格納されるデータを初期化してしまう場合がある。また、例えば、スキャンテストの実行に伴ってテスト対象の処理回路から記憶回路に対して意図せぬデータの書き込みが行われてしまう場合がある。特に、記憶回路がキャッシュメモリである場合には、そのデータが初期化されてしまうと、スキャンテストの実行後に処理回路が処理を再開した際にキャッシュミスが多発し、性能劣化を招いてしまうといった問題も発生する。

【先行技術文献】

【特許文献】

【０００５】

【特許文献１】特開２０１０－１４０２１９号公報

【発明の概要】

【発明が解決しようとする課題】

【０００６】

上述したように、スキャンテストの実行に伴ってテスト対象の処理回路がアクセスする記憶回路に格納されるデータが書き換えられてしまうという問題がある。

【０００７】

その他の課題と新規な特徴は、本明細書の記述及び添付図面から明らかになるであろう。

【課題を解決するための手段】

【０００８】

一実施の形態によれば、半導体装置は、記憶回路に格納されたデータを利用して処理を実行するとともに、処理の実行に応じて記憶回路にデータを書き込む処理回路に対するスキャンテストを実行しているときに、処理回路から記憶回路に対するデータの書き込みを抑止するものである。

【発明の効果】

【０００９】

前記一実施の形態によれば、スキャンテストを実行したとしても、テスト対象の処理回路がアクセスする記憶回路に格納されるデータの書き換えを抑止することができる。

【図面の簡単な説明】

【００１０】

【図１】実施の形態１に係る半導体装置の構成を示すブロック図である。

【図２】ＦＴＴＩ及びＤＴＩについて説明するための図である。

【図３】実施の形態１に係るＣＰＵクラスタの構成を示すブロック図である。

【図４】実施の形態１に係るＣＰＵクラスタの構成を示すブロック図である。

【図５】実施の形態１に係るＣＰＵに対するラインタイムテストの実行手順を示すタイミングチャートである。

【図６】実施の形態１に係る共通回路に対するラインタイムテストの実行手順を示すタイミングチャートである。
【図７】実施の形態１に係るＣＰＵクラスタのランタイムテスト実行時の動作を示すタイミングチャートである。
【図８】実施の形態１に係るＣＰＵのランタイムテスト実行時の動作を示すタイミングチャートである。
【図９】実施の形態１に係るＣＰＵのランタイムテスト実行時の動作を示すタイミングチャートである。
【図１０】実施の形態１に係るＣＰＵのランタイムテスト実行時の動作を示すタイミングチャートである。
【図１１】実施の形態１に係る共通回路のランタイムテスト実行時の動作を示すタイミングチャートである。
【図１２】実施の形態２に係る半導体装置の構成を示すブロック図である。
【図１３】実施の形態３に係る半導体装置の構成を示すブロック図である。
【図１４】実施の形態４に係る半導体装置の構成を示すブロック図である。
【図１５】実施の形態４に係るランタイムテストの分割例を示す図である。
【図１６】実施の形態４に係るＣＰＵクラスタのランタイムテスト実行時の動作を示すタイミングチャートである。
【図１７】実施の形態４に係るＣＰＵの１回目の分割テスト実行時の動作を示すタイミングチャートである。
【図１８】実施の形態４に係るＣＰＵの２回目の分割テスト実行時の動作を示すタイミングチャートである。
【図１９】実施の形態４に係る共通回路の１回目の分割テスト実行時の動作を示すタイミングチャートである。
【図２０】実施の形態４に係る共通回路の２回目の分割テスト実行時の動作を示すタイミングチャートである。
【図２１】実施の形態５に係る半導体装置の構成を示すブロック図である。
【図２２】実施の形態６に係る半導体装置の構成を示すブロック図である。
【図２３】実施の形態７に係る半導体装置の構成を示すブロック図である。
【図２４】実施の形態８に係る半導体装置の構成を示すブロック図である。
【図２５】実施の形態８に係るＧＰＵのランタイムテスト実行時の動作を示すタイミングチャートである。
【図２６】実施の形態９に係る半導体装置の構成を示すブロック図である。
【図２７】他の実施の形態に係る半導体装置の構成を示すブロック図である。
【図２８】実施の形態に係る半導体装置の概略構成を示すブロック図である。
【発明を実施するための形態】
【００１１】
以下、図面を参照しながら、好適な実施の形態について説明する。以下の実施の形態に示す具体的な数値などは、実施の形態の理解を容易とするための例示にすぎず、特に断る場合を除き、それに限定されるものではない。また、以下の記載及び図面では、説明の明確化のため、当業者にとって自明な事項などについては、適宜、省略及び簡略化がなされている。
【００１２】
＜実施の形態１＞
図１を参照して、実施の形態１に係る半導体装置１の構成について説明する。図１に示すように、半導体装置１は、ＣＰＵクラスタ１０と、ＢＩＳＴコントローラ１１と、クロックコントローラ１２と、リセットコントローラ１３と、割り込みコントローラ１４と、タイマ１５と、ＤＤＲ（Double-Data-Rate）コントローラ１６と、外部バスコントローラ１７と、リセットマスク回路１８と、割り込みマスク回路１９と、内蔵メモリ５０とを有する。

【００１３】
ＣＰＵクラスタ１０と、割り込みコントローラ１４と、ＤＤＲコントローラ１６と、外部バスコントローラ１７と、内蔵メモリ５０は、システムバスを介して相互に接続されている。ＢＩＳＴコントローラ１１と、クロックコントローラ１２と、リセットコントローラ１３と、タイマ１５とは、ローカルバスを介して相互に接続されている。また、システムバスと、ローカルバスは、接続されている。
【００１４】
ＣＰＵクラスタ１０は、他の回路１１～１７と協働して半導体装置１としての処理を実行する回路である。ＣＰＵクラスタ１０は、複数のＣＰＵ１００～１０３と、複数のＢＩＳＴコントローラ１１０～１１３と、共通回路１２０とを有する。
【００１５】
ＣＰＵ１００～１０３のそれぞれは、ＣＰＵクラスタ１０が実行する処理を分担して実行する回路である。ＣＰＵ１００～１０３のそれぞれは、ＤＤＲメモリ２０、メモリ２１、及び、内蔵メモリ５０に格納されたデータを使用して、その処理を実行する。このデータには、ＣＰＵ１００～１０３のそれぞれに、その処理を実行させる複数の命令を含むプログラムが含まれる。また、ＣＰＵ１００～１０３のそれぞれは、その処理の実行に応じて、ＤＤＲメモリ２０、メモリ２１、及び、内蔵メモリ５０に格納されたデータを更新する。
【００１６】
ＢＩＳＴコントローラ１１０～１１３のそれぞれは、ＢＩＳＴコントローラ１１からの制御に応じて、ＣＰＵ１００～１０３のそれぞれのランタイムテストを実行する回路である。ＢＩＳＴコントローラ１１０は、ＣＰＵ１００のランタイムテストを実行し、ＢＩＳＴコントローラ１１１は、ＣＰＵ１０１のランタイムテストを実行し、ＢＩＳＴコントローラ１１２は、ＣＰＵ１０２のランタイムテストを実行し、ＢＩＳＴコントローラ１１３は、ＣＰＵ１０３のランタイムテストを実行する。
【００１７】
共通回路１２０は、ＣＰＵクラスタ１０による処理の実行に際して、ＣＰＵ１００～１０３で共通利用される回路である。ＣＰＵ１００～１０３のそれぞれは、ＣＰＵクラスタ１０とシステムバスを介して接続される割り込みコントローラ１４、ＤＤＲコントローラ１６、外部バスコントローラ１７、及び内蔵メモリ５０に対して、共通回路１２０を介してアクセスする。また、ＣＰＵ１００～１０３のそれぞれは、ＣＰＵクラスタ１０とシステムバス及びローカルバスを介して接続されるＢＩＳＴコントローラ１１、クロックコントローラ１２、リセットコントローラ１３、及び、タイマ１５に対して、共通回路１２０を介してアクセスする。
【００１８】
ＢＩＳＴコントローラ１１は、ＣＰＵクラスタ１０のランタイムテストの実行を制御する回路である。より具体的には、ＢＩＳＴコントローラ１１は、ＢＩＳＴコントローラ１１０～１１３のそれぞれによるＣＰＵ１００～１０３のそれぞれのランタイムテストの実行を制御する。よって、ＢＩＳＴコントローラ１１は、マスタとして機能し、ＢＩＳＴコントローラ１１０～１１３のそれぞれは、スレーブとして機能する。
【００１９】
クロックコントローラ１２は、クロック信号を生成して、半導体装置１に含まれる各回路１０、１１、１３～１９、５０に供給する回路である。
【００２０】
リセットコントローラ１３は、ＣＰＵクラスタ１０をリセットする回路である。より具体的には、リセットコントローラ１３は、ＣＰＵ１００～１０３及び共通回路１２０のそれぞれをリセットする。リセットコントローラ１３は、ＣＰＵ１００～１０３及び共通回路１２０のうち、いずれかの回路をリセットする場合、その回路にリセット信号を出力する。ＣＰＵ１００～１０３及び共通回路１２０のそれぞれは、リセットコントローラ１３からのリセット信号の入力に応じてリセットされる。

【００２１】
割り込みコントローラ１４は、半導体装置１内の回路が各ＣＰＵ１００～１０３に対して発生させた割り込み及び外部からの割り込みを、各ＣＰＵ１００～１０３に通知する回路である。より具体的には、割り込みコントローラ１４は、半導体装置１内の回路及び外部からの割り込み信号の入力に応じて、その割り込み信号の通知先として設定されたＣＰＵに対して割り込み信号を出力する。
【００２２】
タイマ１５は、時間の経過を計測し、ＣＰＵクラスタ１０に対してランタイムテストの実行タイミングを割り込みによって通知する回路である。より具体的には、タイマ１５は、ランタイムテストの実行タイミングとなる毎に、割り込み信号を割り込みコントローラ１４に出力する。割り込みコントローラ１４は、この割り込み信号に応じて、割り込みの通知先として予め設定されたＣＰＵに対して割り込み信号を出力する。
【００２３】
ＤＤＲコントローラ１６は、半導体装置１の外部のＤＤＲメモリ２０と接続される。ＤＤＲコントローラ１６は、半導体装置１内の回路からＤＤＲメモリ２０に対するアクセスを制御する。例えば、ＤＤＲコントローラ１６は、ＣＰＵクラスタ１０からのデータのリード要求に応じて、ＤＤＲメモリ２０からデータを読み出し、ＣＰＵクラスタ１０に出力する。また、例えば、ＤＤＲコントローラ１６は、ＣＰＵクラスタ１０からのデータのライト要求に応じて、ＣＰＵクラスタ１０から出力されたデータをＤＤＲメモリ２０に書き込む。
【００２４】
外部バスコントローラ１７は、半導体装置１の外部のメモリ２１と接続される。外部バスコントローラ１７は、半導体装置１内の回路からメモリ２１に対するアクセスを制御する。例えば、外部バスコントローラ１７は、ＣＰＵクラスタ１０からのデータのリード要求に応じて、メモリ２１からデータを読み出し、ＣＰＵクラスタ１０に出力する。また、例えば、外部バスコントローラ１７は、ＣＰＵクラスタ１０からのデータのライト要求に応じて、ＣＰＵクラスタ１０から出力されたデータをメモリ２１に書き込む。
【００２５】
リセットマスク回路１８は、リセットコントローラ１３からＣＰＵ１００～１０３及び共通回路１２０のそれぞれに対するリセットを抑止する回路である。ＣＰＵ１００～１０３及び共通回路１２０のそれぞれについてリセットを抑止するか否かは、リセットマスク回路１８に対して任意に設定することができる。この設定は、リセットコントローラ１３が有する制御レジスタ（図示せず）に対して、リセットを抑止するか否かを示す値を設定することで行われる。また、その設定も、ＣＰＵ１００～１０３のそれぞれから、任意のタイミングで変更することが可能である。リセットマスク回路１８は、制御レジスタの値に基づいて、ＣＰＵ１００～１０３及び共通回路１２０のうち、リセットを抑止すると設定されている回路に対するリセットコントローラ１３からのリセット信号をマスク（遮断）することで、その回路のリセットを抑止する。
【００２６】
割り込みマスク回路１９は、割り込みコントローラ１４からＣＰＵ１００～１０３のそれぞれに対する割り込みを抑止する回路である。ＣＰＵ１００～１０３のそれぞれについて割り込みを抑止するか否かは、割り込みマスク回路１９に対して任意に設定することができる。この設定は、クロックコントローラ１２が有する制御レジスタ（図示せず）に対して、割り込みを抑止するか否かを示す値を設定することで行われる。また、その設定も、ＣＰＵ１００～１０３のそれぞれから、任意のタイミングで変更することが可能である。割り込みマスク回路１９は、制御レジスタの値に基づいて、ＣＰＵ１００～１０３のうち、割り込みを抑止すると設定されているＣＰＵに対する割り込みコントローラ１４からの割り込み信号をマスク（遮断）することで、その回路に対する割り込みを抑止する。
【００２７】
ＤＤＲメモリ２０と、メモリ２１と、内蔵メモリ５０は、各種データが格納される。Ｄ

ＤＲメモリ２０と、メモリ２１と、内蔵メモリ５０は、例えば、上述したように、ＣＰＵ１００～１０３のそれぞれによって使用及び更新されるデータが格納される。

【００２８】

続いて、図２を参照して、実施の形態１に係るランタイムテストの実行タイミングについて説明する。ＩＳＯ２６２６２の規定では、異常発生からそのリカバリに必要な時間であるフォールトトレラント時間間隔（Fault Tolerant Time Interval：FTTI）を確保するために、診断テスト間隔（Diagnostic Test Interval：DTI）で診断を行う必要があると定められている。

【００２９】

そのため、ＣＰＵクラスタ１０は、半導体装置１の起動後に、ランタイムテストの実行タイミングをＤＴＩ毎に通知するようにタイマ１５を設定する。すなわち、ＣＰＵクラスタ１０は、ＤＴＩ毎に割り込み信号を出力するようにタイマ１５を設定する。より具体的には、この設定は、ＣＰＵ１００～１０３のうち、いずれかのＣＰＵによって行われる。タイマ１５を設定するＣＰＵは、例えば、予め定めるようにしてもよい。なお、ＦＴＴＩ及びＤＴＩは、システム開発者によって予め任意の値を定めることができる。

【００３０】

続いて、図３及び図４を参照して、実施の形態１に係るＣＰＵクラスタ１０のより詳細な構成について説明する。図３に示すように、ＣＰＵ１００は、Ｌ１キャッシュコントローラ１１４と、Ｌ１キャッシュメモリ１１５と、分岐履歴メモリ（ブランチヒストリーテーブル）１１６とを有する。なお、図３では、ＣＰＵ１００～１０３のうち、ＣＰＵ１００の構成のみを代表的に図示している。すなわち、ＣＰＵ１０１～１０３の構成も、ＣＰＵ１００と同様であるため、図示及びその説明を省略する。

【００３１】

Ｌ１キャッシュコントローラ１１４は、ＣＰＵ１００に対しメモリ２１へのデータのリード処理及びライト処理を制御するとともに、Ｌ１キャッシュメモリ１１５を管理する回路である。Ｌ１キャッシュコントローラ１１４は、ＣＰＵ１００がその処理に利用するためにメモリ２１からリードしようとしたデータがＬ１キャッシュメモリ１１５に格納されている場合、Ｌ１キャッシュメモリ１１５からそのデータを取得する。

【００３２】

一方、Ｌ１キャッシュコントローラ１１４は、ＣＰＵ１００がメモリ２１からリードしようとしたデータがＬ１キャッシュメモリ１１５に格納されていない場合、そのデータのリードを共通回路１２０に対して要求することで、そのデータを共通回路１２０のＬ２キャッシュメモリ１２４、他のＣＰＵ１０１～１０３のＬ１キャッシュメモリ１１５、又はメモリ２１から取得する。より具体的には、Ｌ１キャッシュコントローラ１１４は、メモリ２１からのデータのリードを要求するリード要求信号を共通回路１２０に出力する。共通回路１２０は、このリード要求信号に応じて、リードが要求されたデータを、Ｌ２キャッシュメモリ１２４、他のＣＰＵ１０１～１０３のＬ１キャッシュメモリ１１５、又はメモリ２１から取得し、そのデータを含むリード応答信号をＣＰＵ１００に出力する。Ｌ１キャッシュコントローラ１１４は、共通回路１２０からのリード応答信号に含まれるデータをＬ１キャッシュメモリ１１５に格納する。また、ＣＰＵ１００は、このデータを利用して処理を実行する。

【００３３】

Ｌ１キャッシュコントローラ１１４は、ＣＰＵ１００がメモリ２１に対してデータをライトしようとした場合、そのデータをＬ１キャッシュメモリ１１５に格納する。このデータは、任意のタイミングでＬ１キャッシュメモリ１１５からメモリ２１に対してフラッシュされる。より具体的には、Ｌ１キャッシュコントローラ１１４は、メモリ２１に対するデータのライトを要求するライト要求信号を共通回路１２０に出力する。このライト要求信号には、メモリ２１に対してライトを要求するデータが含まれる。共通回路１２０は、このライト要求信号に応じて、ライトが要求されたデータをＬ２キャッシュメモリ１２４及びメモリ２１に格納するための制御を行う。

【００３４】
Ｌ１キャッシュメモリ１１５は、上述したように、メモリ２１に格納されるデータが一時的にキャッシュされる記憶回路である。
【００３５】
分岐履歴メモリ１１６は、ＣＰＵ１００が実行するプログラムにおいて過去に実行した分岐命令に従って分岐処理が行われたか否かの履歴を示す情報が格納される。すなわち、ＣＰＵ１００は、分岐命令を実行したときに、その分岐命令の履歴を分岐履歴メモリ１１６に格納する。この履歴は、例えば、分岐命令のアドレスと、その分岐命令に従って分岐したか否かを示す。ＣＰＵ１００は、ある分岐命令の分岐先を予測する場合、分岐履歴メモリ１１６に格納された履歴のうち、その分岐命令と同じアドレスを示す分岐命令の履歴に基づいて分岐先を予測する。
【００３６】
また、図３に示すように、共通回路１２０は、ＢＩＳＴコントローラ１２１と、初期化マスク回路１２２と、Ｌ２キャッシュコントローラ１２３と、Ｌ２キャッシュメモリ１２４と、スヌープ制御ユニット（ＳＣＵ）１２５と、ＳＣＵタグＲＡＭ（Random Access Memory）１２６と、アクセス履歴メモリ１２７とを有する。
【００３７】
ＢＩＳＴコントローラ１２１は、ＢＩＳＴコントローラ１１からの制御に応じて、共通回路１２０のランタイムテストを実行する回路である。ＢＩＳＴコントローラ１２１は、ＢＩＳＴコントローラ１１０～１１３と同様に、スレーブとして機能する。
【００３８】
初期化マスク回路１２２は、共通回路１２０のランタイムテストを実行するときにおける、Ｌ２キャッシュメモリ１２４及びＳＣＵタグＲＡＭ１２６に対するデータの書き込みを抑止する回路である。
【００３９】
Ｌ２キャッシュコントローラ１２３は、ＣＰＵ１００からのメモリ２１へのデータのリード処理及びライト処理を制御するとともに、Ｌ２キャッシュメモリ１２４を管理する回路である。Ｌ２キャッシュコントローラ１２３は、ＣＰＵ１００からメモリ２１に対してデータのリードが要求された場合、後述するようにスヌープ制御ユニット１２５によって、他のＣＰＵ１０１～１０３のＬ１キャッシュメモリ１１５から、そのデータの取得を試みる。Ｌ２キャッシュコントローラ１２３は、スヌープ制御ユニット１２５によってデータが取得できなかった場合、Ｌ２キャッシュメモリ１２４からそのデータを取得する。
【００４０】
一方、Ｌ２キャッシュコントローラ１２３は、ＣＰＵ１００からリードが要求されたデータが、Ｌ２キャッシュメモリ１２４、スヌープ制御ユニット１２５のいずれからも取得できなかった場合、そのデータのリードを外部バスコントローラ１７に対して要求することで、そのデータをメモリ２１から取得する。より具体的には、Ｌ２キャッシュコントローラ１２３は、メモリ２１からのデータのリードを要求するリード要求信号を、システムバスを介して外部バスコントローラ１７に出力する。外部バスコントローラ１７は、このリード要求信号に応じて、メモリ２１から取得したデータを含むリード応答信号を、システムバスを介して共通回路１２０に出力する。Ｌ２キャッシュコントローラ１２３は、このリード応答信号に含まれるデータを取得する。
【００４１】
Ｌ２キャッシュコントローラ１２３は、Ｌ２キャッシュメモリ１２４、メモリ２１、もしくは、スヌープ制御ユニット１２５から取得したデータを、要求元のＣＰＵ１００に出力する。より具体的には、Ｌ２キャッシュコントローラ１２３は、取得したデータを含むリード応答信号を、要求元のＣＰＵ１００に出力する。
【００４２】
Ｌ２キャッシュコントローラ１２３は、ＣＰＵ１００からメモリ２１に対してデータのライトが要求された場合、そのデータをＬ２キャッシュメモリ１２４に格納する。このデ

ータは、任意のタイミングでＬ２キャッシュメモリ１２４からメモリ２１に対してフラッシュされる。より具体的には、Ｌ２キャッシュコントローラ１２３は、メモリ２１に対するそのデータのライトを要求するライト要求信号を、システムバスを介して外部バスコントローラ１７に出力する。外部バスコントローラ１７は、このライト要求信号に応じて、そのライト要求信号に含まれるデータをメモリ２１に格納する。

【００４３】

Ｌ２キャッシュメモリ１２４は、上述したように、メモリ２１に格納されるデータが一時的にキャッシュされる記憶回路である。Ｌ２キャッシュメモリ１２４は、Ｌ１キャッシュメモリ１１５よりも下位レベルのメモリとなる。

【００４４】

スヌープ制御ユニット１２５は、複数のＣＰＵ１００～１０３間におけるＬ１キャッシュメモリ１１５のキャッシュコヒーレンシを、スヌープ方式で保証する制御を行う。

【００４５】

スヌープ制御ユニット１２５は、ＣＰＵ１００からリードが要求されたデータが、そのリードを要求したＣＰＵ１００以外のＣＰＵ１０１～１０３のいずれかのＬ１キャッシュメモリ１１５に格納されているか否かを判定する。スヌープ制御ユニット１２５は、リードを要求したＣＰＵ１００以外のＣＰＵ１０１～１０３のＬ１キャッシュメモリ１１５にデータが格納されていると判定した場合、該ＣＰＵに対してそのデータを要求する。より具体的には、スヌープ制御ユニット１２５は、そのＬ１キャッシュメモリ１１５を有するＣＰＵに対して、そのデータを要求するスヌープ要求信号を出力する。該ＣＰＵのＬ１キャッシュコントローラ１１４は、そのスヌープ要求信号に応じて、そのスヌープ要求信号で要求されたデータをＬ１キャッシュメモリ１１５から取得し、そのデータを含むスヌープ応答信号を共通回路１２０に出力する。スヌープ制御ユニット１２５は、そのスヌープ応答信号に含まれるデータを取得する。

【００４６】

また、スヌープ制御ユニット１２５は、ＣＰＵ１００からライトが要求されたデータが、そのライトを要求したＣＰＵ１００以外のＣＰＵ１０１～１０３のいずれかのＬ１キャッシュメモリ１１５に格納されているか否かを判定する。スヌープ制御ユニット１２５は、ライトを要求したＣＰＵ１００以外のＣＰＵ１０１～１０３のＬ１キャッシュメモリ１１５にデータが格納されていると判定した場合、該ＣＰＵに対して、そのデータの無効化を要求する。より具体的には、スヌープ制御ユニット１２５は、そのＬ１キャッシュメモリ１１５を有するＣＰＵに対して、そのデータの無効化を要求するスヌープ要求信号を出力する。該ＣＰＵのＬ１キャッシュコントローラ１１４は、そのスヌープ要求信号に応じて、Ｌ１キャッシュメモリ１１５において、スヌープ要求信号で無効化が要求されたデータを無効化する。すなわち、このデータは、Ｌ１キャッシュメモリ１１５から削除され、Ｌ１キャッシュメモリ１１５には格納されていない扱いとなる。

【００４７】

ＳＣＵタグＲＡＭ１２６は、ＣＰＵ１００～１０３のＬ１キャッシュメモリ１１５に格納されているデータのそれぞれが、メモリ２１におけるどのアドレスのデータであるかを示す情報が格納される記憶回路である。

【００４８】

スヌープ制御ユニット１２５は、ＳＣＵタグＲＡＭ１２６に格納された情報に基づいて、上述のリード又はライトが要求されたデータが、そのリード又はライトを要求したＣＰＵ１００以外のＣＰＵ１０１～１０３のいずれかのＬ１キャッシュメモリ１１５に格納されているかを判定する。より具体的には、リード要求信号には、リードするデータのメモリ２１におけるアドレスが含まれており、ライト要求信号には、メモリ２１においてデータをライトするアドレスが含まれる。スヌープ制御ユニット１２５は、リード又はライトを要求したＣＰＵ１００以外のＣＰＵ１０１～１０３のいずれかのＬ１キャッシュメモリ１１５に格納されているデータのいずれかのアドレスが、リード要求信号又はライト要求信号に含まれるアドレスと一致するか否かを判定する。アドレスが一致する場合、そのリ

ード又はライトを要求したＣＰＵ１００以外のＣＰＵ１０１～１０３のいずれかのＬ１キャッシュメモリ１１５に格納されているデータについて、上述のデータの要求又は無効化の要求が行われる。

【００４９】

よって、ＣＰＵ１００～１０３のそれぞれのＬ１キャッシュコントローラ１１４は、Ｌ１キャッシュメモリ１１５にデータを格納した場合、そのデータのメモリ２１におけるアドレスを共通回路１２０に通知する。共通回路１２０のスヌープ制御ユニット１２５は、ＣＰＵ１００～１０３のそれぞれのＬ１キャッシュコントローラ１１４からのアドレスの通知に応じて、通知元のＣＰＵのＬ１キャッシュメモリ１１５に通知されたアドレスのデータが格納されていることを示すようにＳＣＵタグＲＡＭ１２６を更新する。また、共通回路１２０のＬ２キャッシュコントローラ１２３は、通知されたアドレスのデータがＬ２キャッシュメモリ１２４に格納されている場合、そのデータは最新のデータではなくなるため、そのデータを無効化する。

【００５０】

アクセス履歴メモリ１２７は、Ｌ２キャッシュメモリ１２４におけるキャッシュラインに対するアクセス履歴を示す情報である。なお、このアクセス履歴は、Ｌ２キャッシュメモリ１２４にデータを書き込む場合に、リフィル（上書き）対象となるキャッシュラインの決定に利用される内容を示す。例えば、キャッシュアルゴリズムとしてＬＲＵ（Least Recently Used）を採用した場合、アクセス履歴は、同一のインデックスアドレスで特定される複数のｗａｙのそれぞれに対応する複数のキャッシュラインのうち、最古にアクセスされたキャッシュラインを示す。また、例えば、キャッシュアルゴリズムとしてＬＦＵ（Least Frequently Used）を採用した場合、アクセス履歴は、同一のインデックスアドレスで特定される複数のｗａｙのそれぞれに対応する複数のキャッシュラインのそれぞれのデータのアクセス頻度を示す。Ｌ２キャッシュコントローラ１２３は、Ｌ２キャッシュメモリ１２４へのアクセスに応じて、アクセス履歴メモリ１２７に格納される情報を更新する。また、Ｌ２キャッシュコントローラ１２３は、Ｌ２キャッシュメモリ１２４にデータを書き込む場合、アクセス履歴メモリ１２７に格納される情報に基づいて、リフィル（上書き）対象となるキャッシュラインを決定する。

【００５１】

また、図３に示すように、ＣＰＵクラスタ１０は、ラッパ回路１３０を有する。ラッパ回路１３０は、複数のマスク回路１３１～１３４を有する。なお、図３では、ＣＰＵ１００～１０３のうち、ＣＰＵ１００に対するラッパ回路１３０のみを代表的に図示している。すなわち、ＣＰＵ１０１～１０３に対しても、ＣＰＵ１００と同様にラッパ回路１３０が設けられるため、図示及びその説明を省略する。

【００５２】

マスク回路１３１は、割り込みコントローラ１４からＣＰＵ１００に対して入力される割り込み信号をマスク（遮断）する回路である。

【００５３】

マスク回路１３２は、ＣＰＵ１００からクロックコントローラ１２及びリセットコントローラ１３に対して出力される信号をマスク（遮断）する回路である。この信号には、例えば、クロックコントローラ１２に対して、低電力状態への遷移に伴い、クロック信号の供給の停止を要求する信号がある。

【００５４】

マスク回路１３３は、ＣＰＵ１００から共通回路１２０に出力される信号をマスク（遮断）する回路である。この信号には、例えば、リード要求信号、ライト要求信号、及び、スヌープ応答信号が含まれる。

【００５５】

マスク回路１３４は、共通回路１２０からＣＰＵ１００に入力される信号をマスク（遮断）する回路である。この信号には、例えば、リード応答信号、ライト応答信号、及び、スヌープ要求信号が含まれる。

【００５６】
ここで、マスク回路１３１、１３４は、ＣＰＵ１００に対して入力される値を固定して、ＣＰＵ１００のスキャンテストにおける期待値を確定させることで、テストパタンの作成を容易化することを目的としている。マスク回路１３２、１３３は、スキャンテストの実行によって、ＣＰＵ１００から他の回路に対する意図しない信号の出力を抑止することで、システムの正常動作の阻害を防止することを目的としている。
【００５７】
マスク回路１３１～１３４が信号をマスク（遮断）するか否かは、マスク回路１３１～１３４に対して任意に設定することができる。この設定は、マスク回路１３１～１３４のそれぞれもしくはクロックコントローラ１２が有する制御レジスタ（図示せず）に対して、信号をマスクするか否かを示す値を設定することで行われる。また、その設定も、ＢＩＳＴコントローラ１１０から、任意のタイミングで変更することが可能である。よって、ＣＰＵ１０１のマスク回路１３１～１３４の設定はＢＩＳＴコントローラ１１１が変更し、ＣＰＵ１０２のマスク回路１３１～１３４の設定はＢＩＳＴコントローラ１１２が変更し、ＣＰＵ１０３のマスク回路１３１～１３４の設定はＢＩＳＴコントローラ１１３が変更することになる。ＣＰＵ１００～１０３のそれぞれのマスク回路１３１～１３４は、ＣＰＵ１００～１０３のそれぞれのランタイムテスト実行時に信号をマスクするように、その動作が有効化される。
【００５８】
また、図４に示すように、初期化マスク回路１２２は、初期化マスク回路１２２ａ～１２２ｃを含む。
【００５９】
初期化マスク回路１２２ａは、Ｌ２キャッシュコントローラ１２３がＬ２キャッシュメモリ１２４に対して初期値を書き込んでＬ２キャッシュメモリ１２４を初期化する処理と、スヌープ制御ユニット１２５がＳＣＵタグＲＡＭ１２６に対して初期値を書き込んでＳＣＵタグＲＡＭ１２６を初期化する処理とを抑止する回路である。より具体的には、初期化マスク回路１２２ａは、初期化処理の抑止を指示する指示信号をＬ２キャッシュコントローラ１２３及びスヌープ制御ユニット１２５のそれぞれに出力する。Ｌ２キャッシュコントローラ１２３及びスヌープ制御ユニット１２５のそれぞれは、共通回路１２０のリセット時に、その指示信号が入力されている場合、共通回路１２０のリセット解除で実行するＬ２キャッシュメモリ１２４及びＳＣＵタグＲＡＭ１２６のそれぞれに対するデータの初期化のための書き込みを行わないようにする。初期化マスク回路１２２ａは、共通回路１２０のランタイムテスト実行後のリセット時における、Ｌ２キャッシュメモリ１２４及びＳＣＵタグＲＡＭ１２６の初期化を抑止するよう、その動作を有効化する。
【００６０】
なお、この指示信号に応じて初期化のための書き込みを実行するか否かの動作を変更する機能は、一般的に、Ｌ２キャッシュコントローラ１２３及びスヌープ制御ユニット１２５に、（１）デバッグ目的、もしくは、（２）ＣＰＵクラスタ１０が省電力状態（一部電源ＯＦＦ状態）から復帰する際のデータ保持目的で利用される。より具体的には、（１）では、ＣＰＵクラスタ１０が異常停止した後にＣＰＵクラスタ１０をリセットして再起動する際にデバッグのためにデータを残す目的で利用される。また、（２）では、ＣＰＵクラスタ１０内の全てのＣＰＵ１００～１０３がスリープして、ＣＰＵ１００～１０３、スヌープ制御ユニット１２５、及びＬ２キャッシュコントローラ１２３を電源切断した省電力状態から、それらをリセットして復帰する際に処理を継続可能とするためにデータを引き継ぐ目的で利用される。本実施の形態１では、この機能をランタイムテストの実行に流用することで、ランタイムテスト実行時におけるＬ２キャッシュメモリ１２４及びＳＣＵタグＲＡＭ１２６のデータの引き継ぎに関して、論理設計コストの低減を実現している。
【００６１】
初期化マスク回路１２２ｂは、Ｌ２キャッシュコントローラ１２３によるＬ２キャッシュメモリ１２４に対する書き込みを抑止する回路である。より具体的には、初期化マスク

回路１２２ｂは、Ｌ２キャッシュコントローラ１２３からＬ２キャッシュメモリ１２４に対してデータの書き込みのために出力される信号をマスク（遮断）する。初期化マスク回路１２２ｂは、共通回路１２０のランタイムテスト実行時におけるＬ２キャッシュメモリ１２４へのデータの書き込みを抑止するよう、その動作を有効化する。

【００６２】

初期化マスク回路１２２ｃは、スヌープ制御ユニット１２５によるＳＣＵタグＲＡＭ１２６に対する書き込みを抑止する回路である。より具体的には、初期化マスク回路１２２ｃは、スヌープ制御ユニット１２５からＳＣＵタグＲＡＭ１２６に対してデータの書き込みのために出力される信号をマスク（遮断）する。初期化マスク回路１２２ｃは、共通回路１２０のランタイムテスト実行時におけるＳＣＵタグＲＡＭ１２６へのデータの書き込みを抑止するよう、その動作を有効化する。

【００６３】

初期化マスク回路１２２ａ～１２２ｃが初期化又は書き込みを抑止するか否かは、初期化マスク回路１２２ａ～１２２ｃに対して任意に設定することができる。この設定は、初期化マスク回路１２２ａ～１２２ｃのそれぞれもしくはクロックコントローラ１２が有する制御レジスタ（図示せず）に対して、初期化又は書き込みを抑止するか否かを示す値を設定することで行われる。また、その設定も、ＣＰＵ１００～１０３のうち、共通回路１２０のランタイムテストの実行を制御するＣＰＵから、任意のタイミングで変更することが可能である。

【００６４】

また、図４に示すように、共通回路１２０は、ラッパ回路１４０を有する。ラッパ回路１４０は、複数のマスク回路１４１～１５０を有する。

【００６５】

マスク回路１４１、１４３、１４５、１４７のそれぞれは、ＣＰＵ１００～１０３のそれぞれから共通回路１２０に入力される信号をマスク（遮断）する回路である。この信号には、例えば、リード要求信号、ライト要求信号、及び、スヌープ応答信号が含まれる。

【００６６】

マスク回路１４２、１４４、１４６、１４８のそれぞれは、共通回路１２０からＣＰＵ１００～１０３のそれぞれに出力される信号をマスク（遮断）する回路である。この信号には、例えば、リード応答信号、ライト応答信号、及び、スヌープ要求信号が含まれる。

【００６７】

マスク回路１４９は、共通回路１２０からシステムバスに出力される信号をマスク（遮断）する回路である。この信号には、例えば、リード要求信号及びライト要求信号が含まれる。

【００６８】

マスク回路１５０は、システムバスから共通回路１２０に入力される信号をマスク（遮断）する回路である。この信号には、例えば、リード応答信号及びライト応答信号が含まれる。

【００６９】

ここで、マスク回路１４１、１４３、１４５、１４７、１５０は、共通回路１２０に対して入力される値を固定して、共通回路１２０のスキャンテストにおける期待値を確定させることで、テストパタンの作成を容易化することを目的としている。マスク回路１４２、１４４、１４６、１４８、１４９は、スキャンテストの実行によって、共通回路１２０から他の回路に対する意図しない信号の出力を抑止することで、システムの正常動作の阻害を防止することを目的としている。

【００７０】

マスク回路１４１～１５０が信号をマスク（遮断）するか否かは、マスク回路１４１～１５０に対して任意に設定することができる。この設定は、マスク回路１４１～１５０のそれぞれもしくはクロックコントローラ１２が有する制御レジスタ（図示せず）に対して、信号をマスクするか否かを示す値を設定することで行われる。また、その設定も、ＢＩ

ＳＴコントローラ１２１から、任意のタイミングで変更することが可能である。マスク回路１４１～１５０は、共通回路１２０のランタイムテスト実行時に信号をマスクするよう、その動作を有効化する。

【００７１】

なお、以上の説明では、メモリ２１のデータがＬ１キャッシュメモリ１１５及びＬ２キャッシュメモリ１２４にキャッシュされる例について説明したが、ＤＤＲメモリ２０及び内蔵メモリ５０のデータも同様の制御によりキャッシュされるようにしてもよい。以降の説明においても同様である。

【００７２】

続いて、図５を参照して、実施の形態１に係るＣＰＵ１００～１０３に対するランタイムテストの実行手順について説明する。以下、ＣＰＵ１００における手順について説明するが、ＣＰＵ１０１～１０３における手順についても同様である。

【００７３】

ＣＰＵ１００は、ラインタイムテスト開始のトリガとなる割り込みの通知に応じて、自身のランタイムテストを実行する制御を開始する。まず、ＣＰＵ１００は、システムバスを介して、テスト条件をＢＩＳＴコントローラ１１の制御レジスタ（図示せず）に対して設定する（Ｓ１）。このテスト条件の設定には、テスト対象の設定が含まれる。より具体的には、ＣＰＵ１００は、ＢＩＳＴコントローラ１１に対して自身をテスト対象として設定する。

【００７４】

ＣＰＵ１００は、システムバスを介して、ランタイムテストの起動をＢＩＳＴコントローラ１１に指示する（Ｓ２）。ＢＩＳＴコントローラ１１は、この指示に応じて、テスト対象として設定されたＣＰＵ１００がスリープ状態に移行することを待ち合わせる。

【００７５】

ＣＰＵ１００は、自身が有するレジスタに格納されたデータをＤＤＲメモリ２０、メモリ２１、及び、内蔵メモリ５０に退避（ライト）する（Ｓ３）。また、ＣＰＵ１００は、Ｌ１キャッシュメモリ１１５のデータをメモリ２１に対してフラッシュする。すなわち、ＣＰＵ１００のＬ１キャッシュメモリ１１５に格納された最新のデータを下位のＬ２キャッシュメモリ１２４及びメモリ２１に反映する。これにより、ＣＰＵ１００のランタイムテストを実行することで、ＣＰＵ１００のＬ１キャッシュメモリ１１５のデータが削除されてしまっても、ランタイムテスト終了後にＣＰＵ１００～１０３で実行する処理に不整合が発生しないようにすることができる。ＣＰＵ１００は、割り込みコントローラ１４から自身に対する割り込み信号をマスクするように、割り込みマスク回路１９を設定する（Ｓ４）。そして、ＣＰＵ１００は、自身をＷＦＩ（Wait For Interrupt）命令（スリープ命令）によりスリープ状態に移行する（Ｓ５）。このＷＦＩ命令により移行するスリープ状態は、割り込みの通知に応じて起床するスリープ状態である。ここで、ＷＦＩによるスリープ状態への移行中に割り込みを受けてしまうと、ＣＰＵ１００は、スリープ状態への移行を中止してしまうという問題がある。そのため、これを防止するために、ＣＰＵ１００への割り込み信号をマスクするようにしている。ＣＰＵ１００は、スリープ状態に移行すると、自身がスリープ状態であることを通知するステータス信号をＢＩＳＴコントローラ１１に出力する。

【００７６】

ＢＩＳＴコントローラ１１は、テスト対象として設定されたＣＰＵ１００から、スリープ状態であることを通知するステータス信号が出力されたことに応じて、ＣＰＵ１００のランタイムテストの開始をＢＩＳＴコントローラ１１０に指示する。ＢＩＳＴコントローラ１１０は、ＢＩＳＴコントローラ１１からの指示に応じて、ＣＰＵ１００のランタイムテストを開始する。まず、ＢＩＳＴコントローラ１１０は、ＣＰＵ１００のラッパ回路１３０のマスク回路１３１～１３４が信号をマスクするように設定する（Ｓ６）。これにより、テスト対象のＣＰＵ１００と接続される他の回路に対して意図しない不要な信号が伝搬することと、他の回路からテスト対象のＣＰＵ１００に対して意図しない不要な信号が

伝搬することを防止する。

【００７７】

ＢＩＳＴコントローラ１１０は、ＣＰＵ１００のランタイムテストを実行する（Ｓ７）。より具体的には、ＢＩＳＴコントローラ１１０は、ＣＰＵ１００の論理回路に対してスキャンテスト（ＬＢＩＳＴ、ロジックＢＩＳＴ）を実行し、ＣＰＵ１００の記憶回路（例えば分岐履歴メモリ１１６）に対してＭＢＩＳＴ（メモリＢＩＳＴ）を実行する。

【００７８】

すなわち、ＢＩＳＴコントローラ１１０は、順次、テストパタンを生成し、ＣＰＵ１００の論理回路のスキャンチェーンにスキャンインする。また、ＢＩＳＴコントローラ１１０は、順次、ＣＰＵ１００の論理回路のスキャンチェーンから実行結果データをスキャンアウトして取得する。ＢＩＳＴコントローラ１１０は、取得した実行結果データを圧縮し、ＢＩＳＴ実行結果を生成する。ＢＩＳＴコントローラ１１０は、生成したＢＩＳＴ実行結果と、テストパタンに応じて生成した期待値とを比較する。そして、ＢＩＳＴコントローラ１１０は、比較した値が一致する場合には、ＣＰＵ１００の論理回路が正常であると判定し、比較した結果が一致しない場合には、ＣＰＵ１００の論理回路が故障していると判定する。

【００７９】

また、ＢＩＳＴコントローラ１１０は、順次、テストパタンを生成し、ＣＰＵ１００の記憶回路に書き込む。ＢＩＳＴコントローラ１１０は、ＣＰＵ１００の記憶回路から書き込んだテストパタンを読み出す。ＢＩＳＴコントローラ１１０は、書き込み前の（書き込みに使用した）テストパタンと、読み出したテストパタンとを比較する。そして、ＢＩＳＴコントローラ１１０は、比較した値が一致する場合には、ＣＰＵ１００の記憶回路が正常であると判定し、比較した結果が一致しない場合には、ＣＰＵ１００の記憶回路が故障していると判定する。ここで、自身でエラーを検出する機構（例えばＥＣＣ（Error Checking and Correction））を有している記憶回路（例えばＬ１キャッシュメモリ１１５）は、ＭＢＩＳＴの対象外となる。

【００８０】

ＢＩＳＴコントローラ１１０は、ＣＰＵ１００の論理回路及び記憶回路が全て正常であると判定した場合、ＣＰＵ１００が正常であることを示すランタイムテスト結果値を、ＢＩＳＴコントローラ１１の結果レジスタに格納する。一方、ＢＩＳＴコントローラ１１０は、ＣＰＵ１００の論理回路及び記憶回路のいずれかが故障していると判定した場合、ＣＰＵ１００が故障していることを示すランタイムテスト結果値を、ＢＩＳＴコントローラ１１の結果レジスタに格納する。

【００８１】

ＢＩＳＴコントローラ１１０は、ＣＰＵ１００のランタイムテストの実行後、ＣＰＵ１００のラッパ回路１３０のマスク回路１３１～１３４が信号のマスクを解除するように設定する（Ｓ８）。ＢＩＳＴコントローラ１１０は、ＣＰＵ１００のランタイムテストの実行終了をＢＩＳＴコントローラ１１に通知する。ＢＩＳＴコントローラ１１は、ＢＩＳＴコントローラ１１０からの通知に応じて、リセットコントローラ１３に対して、テスト対象のＣＰＵ１００のリセットを要求する要求信号を出力する（Ｓ９）。リセットコントローラ１３は、ＢＩＳＴコントローラ１１からの要求信号に応じて、その要求信号で要求されたＣＰＵ１００に対してリセット信号を出力する。これより、ＣＰＵ１００がリセットされる。

【００８２】

ＣＰＵ１００は、リセットによるブート時に、自身のリセット要因を確認する（Ｓ１０）。より具体的には、ＣＰＵ１００は、システムバスを介して、ＢＩＳＴコントローラ１１が有する制御レジスタを参照することで、自身がランタイムテスト実行後のリセットによりブートしたか否かを確認する。すなわち、ＢＩＳＴコントローラ１１は、ＣＰＵ１００のランタイムテストを実行した場合には、制御レジスタにＣＰＵ１００のランタイムテストを実行したことを示すデータを格納している。このＣＰＵ１００のランタイムテスト

を実行したことを示すデータは、ＣＰＵ１００による確認後、ＣＰＵ１００によって制御レジスタから削除される。

【００８３】

ＣＰＵ１００は、自身がランタイムテスト実行後のリセットによりブートしたことを確認した場合、そのブート時の処理において、システムバスを介して、ＢＩＳＴコントローラ１１が有する結果レジスタに格納されたランタイムテスト結果値を取得し（Ｓ１１）、ランタイムテストを終了する（Ｓ１２）。

【００８４】

ＣＰＵ１００は、ランタイムテスト結果値が、ＣＰＵ１００が正常であることを示している場合、上記ステップＳ３で退避したデータをＣＰＵ１００のレジスタに復帰させる（Ｓ１３）。ＣＰＵ１００は、割り込みコントローラ１４から自身に対する割り込み信号のマスクを解除するように、割り込みマスク回路１９を設定する（Ｓ１４）。そして、ＣＰＵ１００は、ランタイムテスト実行前に実行していたプログラムの実行を再開する。

【００８５】

一方、ＣＰＵ１００は、ランタイムテスト結果値が、ＣＰＵ１００が故障していることを示している場合、それに応じた故障対策ルーチンを実行する。故障対策ルーチンとして、例えば、半導体装置１におけるシステム全体をシャットダウンする処理を実行してもよく、故障しているＣＰＵ１００を縮退し、他のＣＰＵ１０１～１０３で半導体装置１におけるシステムの動作を継続するようにしてもよい。

【００８６】

なお、ＣＰＵ１００は、自身がランタイムテスト実行後のリセットによりブートしていない場合（通常のリセット後のブートである場合）、通常のリセット後のブート時の処理を実行し、これらのランタイムテスト結果値を取得する処理、及びその値に応じた処理を実行しない。

【００８７】

続いて、図６を参照して、実施の形態１に係る共通回路１２０に対するランタイムテストの実行手順について説明する。以下、本実施の形態１では、ＣＰＵ１００が共通回路１２０のランタイムテストの実行を制御する手順について説明するが、ＣＰＵ１００に代えてＣＰＵ１０１～１０３のいずれかが共通回路１２０のランタイムテストの実行を制御する形態としてもよい。

【００８８】

ＣＰＵ１００は、自身のランタイムテストの実行後に、共通回路１２０のランタイムテストを実行する制御を開始する。まず、ＣＰＵ１００は、自身と共通回路１２０を共有する他のＣＰＵ１０１～１０３のそれぞれに対してＣＰＵ間割り込みを通知する（Ｓ２１）。ＣＰＵ１０１～１０３のそれぞれは、このＣＰＵ間割り込みに応じて、自身をＷＦＩ命令によりスリープ状態に移行させる。

【００８９】

ＣＰＵ１００は、システムバスを介して、テスト条件をＢＩＳＴコントローラ１１の制御レジスタに対して設定する（Ｓ２２）。このテスト条件の設定には、上述したようにテスト対象の設定が含まれる。より具体的には、ＣＰＵ１００は、ＢＩＳＴコントローラ１１に対して共通回路１２０をテスト対象として設定する。

【００９０】

ＣＰＵ１００は、システムバスを介して、ランタイムテストの起動をＢＩＳＴコントローラ１１に指示する（Ｓ２３）。ＢＩＳＴコントローラ１１は、この指示に応じて、ＣＰＵ１００～１０３の全てがスリープ状態に移行することを待ち合わせる。

【００９１】

ＣＰＵ１００は、共通回路１２０が有するレジスタに格納されたデータをＤＤＲメモリ２０、メモリ２１、及び、内蔵メモリ５０に退避（ライト）する（Ｓ２４）。ＣＰＵ１００は、割り込みコントローラ１４から他のＣＰＵ１０１～１０３に対する割り込み信号をマスクするように割り込みマスク回路１９を設定する。ＣＰＵ１００は、ＢＩＳＴコント

ローラ１１からの割り込みのみについて、ＣＰＵ１００に対して割り込み信号を出力するように、システムバスを介して割り込みコントローラ１４を設定する。すなわち、ＣＰＵ１００は、ＢＩＳＴコントローラ１１以外からの割り込みに対して割り込み信号を出力しないように割り込みコントローラ１４を設定する。また、ＣＰＵ１００は、リセットコントローラ１３から共通回路１２０を利用する全てのＣＰＵ１００～１０３に対するリセット信号をマスクするようにリセットマスク回路１８を設定する（Ｓ２５）。そして、ＣＰＵ１００は、自身をＷＦＩ命令によりスリープ状態に移行する（Ｓ２６）。ＣＰＵ１００～１０３のそれぞれは、スリープ状態に移行すると、自身がスリープ状態であることを通知するステータス信号をＢＩＳＴコントローラ１１に出力する。

【００９２】

ＢＩＳＴコントローラ１１は、共通回路１２０を利用するＣＰＵ１００～１０３の全てから、スリープ状態であることを通知するステータス信号が出力されたことに応じて、共通回路１２０のランタイムテストの開始をＢＩＳＴコントローラ１２１に指示する。ＢＩＳＴコントローラ１２１は、ＢＩＳＴコントローラ１１からの指示に応じて、共通回路１２０のランタイムテストを開始する。まず、ＢＩＳＴコントローラ１２１は、共通回路１２０のラッパ回路１４０のマスク回路１４１～１５０が信号をマスクするように設定する（Ｓ２７）。これにより、テスト対象の共通回路１２０と接続される他の回路に対して意図しない不要な信号が伝搬することと、他の回路からテスト対象の共通回路１２０に対して意図しない不要な信号が伝搬することを防止する。

【００９３】

ＢＩＳＴコントローラ１２１は、共通回路１２０のランタイムテストを実行する（Ｓ２８）。より具体的には、ＢＩＳＴコントローラ１２１は、共通回路１２０の論理回路に対してスキャンテスト（ＬＢＩＳＴ、ロジックＢＩＳＴ）を実行し、共通回路１２０の記憶回路（例えばアクセス履歴メモリ１２７）に対してＭＢＩＳＴ（メモリＢＩＳＴ）を実行する。

【００９４】

すなわち、ＢＩＳＴコントローラ１２１は、順次、テストパタンを生成し、共通回路１２０の論理回路のスキャンチェーンにスキャンインする。また、ＢＩＳＴコントローラ１２１は、順次、共通回路１２０の論理回路のスキャンチェーンから実行結果データをスキャンアウトして取得する。ＢＩＳＴコントローラ１２１は、取得した実行結果データを圧縮し、ＢＩＳＴ実行結果を生成する。ＢＩＳＴコントローラ１２１は、生成したＢＩＳＴ実行結果と、テストパタンに応じて生成した期待値とを比較する。そして、ＢＩＳＴコントローラ１２１は、比較した値が一致する場合には、共通回路１２０の論理回路が正常であると判定し、比較した結果が一致しない場合には、共通回路１２０の論理回路が故障していると判定する。

【００９５】

また、ＢＩＳＴコントローラ１２１は、順次、テストパタンを生成し、共通回路１２０の記憶回路に書き込む。ＢＩＳＴコントローラ１２１は、共通回路１２０の記憶回路から書き込んだテストパタンを読み出す。ＢＩＳＴコントローラ１２１は、書き込み前の（書き込みに使用した）テストパタンと、読み出したテストパタンとを比較する。そして、ＢＩＳＴコントローラ１２１は、比較した値が一致する場合には、共通回路１２０の記憶回路が正常であると判定し、比較した結果が一致しない場合には、共通回路１２０の記憶回路が故障していると判定する。ここで、自身でエラーを検出する機構（例えばＥＣＣ）を有している記憶回路（例えばＬ２キャッシュメモリ１２４及びＳＣＵタグＲＡＭ１２６）は、ＭＢＩＳＴの対象外となる。

【００９６】

ＢＩＳＴコントローラ１２１は、共通回路１２０の論理回路及び記憶回路が全て正常であると判定した場合、共通回路１２０が正常であることを示すランタイムテスト結果値を、ＢＩＳＴコントローラ１１の結果レジスタに格納する。一方、ＢＩＳＴコントローラ１２１は、共通回路１２０の論理回路及び記憶回路のいずれかが故障していると判定した場

合、共通回路１２０が故障していることを示すランタイムテスト結果値を、ＢＩＳＴコントローラ１１の結果レジスタに格納する。

【００９７】

ＢＩＳＴコントローラ１２１は、共通回路１２０のランタイムテストの実行後、共通回路１２０のラッパ回路１４０のマスク回路１４１～１５０が信号のマスクを解除するように設定する（Ｓ２９）。ＢＩＳＴコントローラ１２１は、共通回路１２０のランタイムテストの実行終了をＢＩＳＴコントローラ１１に通知する。ＢＩＳＴコントローラ１１は、ＢＩＳＴコントローラ１２１からの通知に応じて、リセットコントローラ１３に対して、テスト対象の共通回路１２０のリセットを要求する要求信号を出力する（Ｓ３０）。リセットコントローラ１３は、ＢＩＳＴコントローラ１１からの要求信号に応じて、その要求信号で要求された共通回路１２０に対してリセット信号を出力する。これより、共通回路１２０がリセットされる。

【００９８】

ここで、リセットコントローラ１３は、一般的に、共通回路１２０をリセットする場合には、その共通回路１２０を共有するＣＰＵ１００～１０３もリセットするように設計される。これに対して、上述したように、ステップＳ２５では、ＣＰＵ１００～１０３に対するリセット信号をマスクするようにリセットマスク回路１８を設定している。そのため、ＣＰＵ１００～１０３のリセットを抑止し、共通回路１２０のランタイムテストの終了後に、ＣＰＵ１００～１０３がその動作を継続することが可能である。

【００９９】

また、ＢＩＳＴコントローラ１１は、ＢＩＳＴコントローラ１２１からの通知に応じて、割り込みコントローラ１４に対して割り込み信号を出力する。割り込みコントローラ１４は、ＢＩＳＴコントローラ１１からの割り込み信号に応じて、ＣＰＵ１００に対して割り込み信号を出力する。ＣＰＵ１００は、割り込みコントローラ１４からの割り込み信号に応じて、スリープ状態から起床し（Ｓ３１）、システムバスを介して、ＢＩＳＴコントローラ１１が有する結果レジスタに格納されたランタイムテスト結果値を取得し（Ｓ３２）、ランタイムテストを終了する（Ｓ３３）。

【０１００】

ＣＰＵ１００は、ランタイムテスト結果値が、共通回路１２０が正常であることを示している場合、上記ステップＳ２４で退避したデータを共通回路１２０のレジスタに復帰する（Ｓ３４）。ＣＰＵ１００は、割り込みコントローラ１４から他のＣＰＵ１０１～１０３に対する割り込み信号のマスクを解除するように割り込みマスク回路１９を設定する。また、ＣＰＵ１００は、ＢＩＳＴコントローラ１１以外からの割り込みについても、ＣＰＵ１００に対する割り込み信号の出力を再開するように、システムバスを介して割り込みコントローラ１４を設定する。また、ＣＰＵ１００は、リセットコントローラ１３から共通回路１２０を利用する全てのＣＰＵ１００～１０３に対するリセット信号のマスクを解除するようにリセットマスク回路１８を設定する（Ｓ３５）。

【０１０１】

ＣＰＵ１００は、自身と共通回路１２０を共有する他のＣＰＵ１０１～１０３のそれぞれに対してＣＰＵ間割り込みを通知する（Ｓ３６）。そして、ＣＰＵ１００は、ランタイムテスト実行前に実行していたプログラムの実行を再開する。また、ＣＰＵ１０１～１０３のそれぞれも、ＣＰＵ１００からのＣＰＵ間割り込みに応じて、スリープ状態から起床し、ランタイムテスト実行前に実行していたプログラムの実行を再開する。

【０１０２】

一方、ＣＰＵ１００は、ランタイムテスト結果値が、ＣＰＵ１００が故障していることを示している場合、それに応じた故障対策ルーチンを実行する。故障対策ルーチンとして、例えば、半導体装置１におけるシステム全体をシャットダウンする処理を実行してもよい。

【０１０３】

ここで、ＷＦＩ命令によるＣＰＵ１００～１０３のスリープ状態への移行は、一般的に

、半導体装置１にエミュレータを接続してデバッグを行うために使用される。本実施の形態１では、以上に説明したように、このＷＦＩ命令によるスリープ状態への移行をランタイムテストの実行に流用することで、設計コストを低減して、ランタイムテスト実行時におけるＣＰＵ１００～１０３の動作防止と、その動作の再開を実現している。

【０１０４】

続いて、図７を参照して、実施の形態１に係るＣＰＵクラスタ１０のランタイムテスト実行時の動作について説明する。

【０１０５】

図７に示すように、ＣＰＵクラスタ１０内の全ての構成要素（ＣＰＵ１００～１０３及び共通回路１２０）に対するランタイムテストは、システム全体の性能劣化を極力低減するために、ＤＴＩ内に時分割で実施する。ＤＴＩ間隔の通知は、上述したようにタイマ１５からの割り込みにより実現する。

【０１０６】

ＣＰＵ１０３は、タイマ１５からの割り込みに応じて、図５に示した手順に従って、自身の処理を中断し、自身のランタイムテストを実行する。このとき、他のＣＰＵ１００～１０２のそれぞれは、その処理の実行を継続する。ＣＰＵ１０３は、ランタイムテストの実行終了後、自身の処理を再開し、ＣＰＵ１０２に対してＣＰＵ間割り込みを通知する。

【０１０７】

ＣＰＵ１０２は、ＣＰＵ１０３からのＣＰＵ間割り込みに応じて、図５に示した手順に従って、自身の処理を中断し、自身のランタイムテストを実行する。このとき、他のＣＰＵ１００～１０１、１０３のそれぞれは、その処理の実行を継続する。ＣＰＵ１０２は、ランタイムテストの実行終了後、自身の処理を再開し、ＣＰＵ１０１に対してＣＰＵ間割り込みを通知する。

【０１０８】

ＣＰＵ１０１は、ＣＰＵ１０２からのＣＰＵ間割り込みに応じて、図５に示した手順に従って、自身の処理を中断し、自身のランタイムテストを実行する。このとき、他のＣＰＵ１００、１０２～１０３のそれぞれは、その処理の実行を継続する。ＣＰＵ１０１は、ランタイムテストの実行終了後、自身の処理を再開し、ＣＰＵ１００に対してＣＰＵ間割り込みを通知する。

【０１０９】

ＣＰＵ１００は、ＣＰＵ１０１からのＣＰＵ間割り込みに応じて、図５に示した手順に従って、自身の処理を中断し、自身のランタイムテストを実行する。このとき、他のＣＰＵ１０１～１０３のそれぞれは、自身の処理の実行を継続する。

【０１１０】

ＣＰＵ１００は、ランタイムテストの実行終了後、図６に示した手順に従って、共通回路１２０のランタイムテストを実行する。このとき、共通回路１２０を共有する全てのＣＰＵ１００～１０３のそれぞれは、上述したようにスリープして自身の処理を中断する。ＣＰＵ１００～１０３のそれぞれは、共通回路１２０のランタイムテストの実行終了後、自身の処理を再開する。

【０１１１】

そして、タイマ１５は、ＣＰＵ１０３に対して割り込みを発生させた時点から、ＤＴＩが経過する毎の時点で再びＣＰＵ１０３に割り込みを発生させ、ＣＰＵ１００～１０３及び共通回路１２０のランタイムテストが実行される。すなわち、ＤＴＩが経過する毎に、ＣＰＵ１００～１０３及び共通回路１２０のランタイムテストが実行される。

【０１１２】

続いて、図８を参照して、実施の形態１に係るＣＰＵ１０３のランタイムテスト実行時の動作について説明する。

【０１１３】

タイマ１５は、前回に割り込み信号を出力した時点から、ＤＴＩが経過した時点で、割り込みコントローラ１４を介してＣＰＵ１０３に対して割り込みを通知する。すなわち、

この例では、割り込みコントローラ１４には、タイマ１５からの割り込みの通知先として、ＣＰＵ１０３が予め設定されている。

【０１１４】

ＣＰＵ１０３は、その処理の実行中に（Ｓ１００）、タイマ１５に起因する割り込みが通知されると、タイマ１５の割り込みに応じた割り込みハンドラを実行する。ＣＰＵ１０３は、割り込みハンドラにおいて、ランタイムテストを実行するための設定（Ｓ１０１）と、ＣＰＵ１０３の情報の退避（Ｓ１０２）とを行い、スリープする（Ｓ１０３）。すなわち、ステップＳ１０１は、図５のステップＳ１、Ｓ２に対応し、ステップＳ１０２は、図５のステップＳ３、Ｓ４に対応し、ステップＳ１０３は、図５のステップＳ５に対応する。

【０１１５】

ＣＰＵ１０３のスリープ後、ＢＩＳＴコントローラ１１は、ＣＰＵ１０３のランタイムテストを実行する（Ｓ１０４）。ＢＩＳＴコントローラ１１は、ＣＰＵ１０３のランタイムテストの実行終了後、リセットコントローラ１３を介してＣＰＵ１０３をリセットする（Ｓ１０５）。すなわち、ステップＳ１０４は、図５のステップＳ６～Ｓ８に対応し、ステップＳ１０５は、図５のステップＳ９に対応する。

【０１１６】

ＣＰＵ１０３は、そのリセット後のブート時に（Ｓ１０６）、リセットハンドラを実行する。ＣＰＵ１０３は、リセットハンドラにおいて、ランタイムテスト結果の確認（Ｓ１０７）、ＣＰＵ１０３の情報の復帰（Ｓ１０８）を行う。すなわち、ステップＳ１０６は、図５のステップＳ１０に対応し、ステップＳ１０７は、図５のステップＳ１１、Ｓ１２に対応し、ステップＳ１０８は、図５のステップＳ１３、Ｓ１４に対応する。また、ＣＰＵ１０３は、リセットハンドラにおいて、次のテスト対象となるＣＰＵ１０２に対してＣＰＵ間割り込みを通知する（Ｓ１０９）。そして、ＣＰＵ１０３は、リセットハンドラを終了し（Ｓ１１０）、ランタイムテストの実行前に実行していた処理を再開する（Ｓ１１１）。

【０１１７】

続いて、図９を参照して、実施の形態１に係るＣＰＵ１０２、１０１のランタイムテスト実行時の動作について説明する。図９では、図８における処理と同様の処理については同一の符号を付し、適宜その説明を省略する。

【０１１８】

ＣＰＵ１０２、１０１のそれぞれは、その処理の実行中に（Ｓ１００）、直前にランタイムテストを実行したＣＰＵからＣＰＵ間割り込みが通知されると、そのＣＰＵ間割り込みに応じた割り込みハンドラを実行する。ＣＰＵ１０２は、ＣＰＵ１０３からＣＰＵ間割り込みが通知され、ＣＰＵ１０１は、ＣＰＵ１０２からＣＰＵ間割り込みが通知される。すなわち、ＣＰＵ１０２、１０１のそれぞれは、ＣＰＵ１０３、１０２のそれぞれからのＣＰＵ割り込みに応じた割り込みハンドラにおいて、ステップＳ１０１～Ｓ１０３の処理を実行する点が、ＣＰＵ１０３とは異なる。

【０１１９】

また、リセットハンドラにおけるステップＳ１０９では、ＣＰＵ１０２は、ＣＰＵ間割り込みをＣＰＵ１０１に通知し、ＣＰＵ１０１は、ＣＰＵ間割り込みをＣＰＵ１００に通知する。

【０１２０】

続いて、図１０を参照して、実施の形態１に係るＣＰＵ１００のランタイムテスト実行時の動作について説明する。図１０では、図８及び図９における処理と同様の処理については同一の符号を付し、適宜その説明を省略する。

【０１２１】

ＣＰＵ１００は、その処理の実行中に（Ｓ１００）、直前にランタイムテストを実行したＣＰＵ１０１からＣＰＵ間割り込みが通知されると、そのＣＰＵ間割り込みに応じた割り込みハンドラを実行する。ＣＰＵ１００は、ＣＰＵ１０２、１０１と同様に、ＣＰＵ１

０１からのＣＰＵ間割り込みに応じた割り込みハンドラにおいてステップＳ１０１～Ｓ１０３の処理を実行する。

【０１２２】

一方、ＣＰＵ１００は、リセットハンドラにおけるステップＳ１０６～Ｓ１０８の実行後、他の全てのＣＰＵ１０１～１０３のそれぞれに対してＣＰＵ割り込みを通知し（Ｓ１１２）、次に図１１を参照して説明するように共通回路１２０のランタイムテストを実行する点が、ＣＰＵ１０１～１０３とは異なる。

【０１２３】

続いて、図１１を参照して、実施の形態１に係る共通回路１２０のランタイムテスト実行時の動作について説明する。

【０１２４】

図１０でも説明したように、ＣＰＵ１００は、自身のランタイムテストの実行後、リセットハンドラにおいて、他の全てのＣＰＵ１０１～１０３のそれぞれに対してＣＰＵ割り込みを通知する（Ｓ１１２）。すなわち、ステップＳ１１２は、図６のステップＳ２１に対応する。ＣＰＵ１０１～１０３のそれぞれは、その処理の実行中に（Ｓ１３０）、ＣＰＵ１００からＣＰＵ間割り込みが通知されると、ＣＰＵ１００からのＣＰＵ間割り込みに応じた割り込みハンドラを実行する。ＣＰＵ１０１～１０３のそれぞれは、ＣＰＵ１００からのＣＰＵ間割り込みに応じた割り込みハンドラにおいて、自身をＷＦＩ命令によりスリープ状態に移行させる（Ｓ１３１）。

【０１２５】

また、ＣＰＵ１００は、リセットハンドラにおいて、ランタイムテストを実行するための設定（Ｓ１１３）と、共通回路１２０の情報の退避（Ｓ１１４）とを行い、スリープする（Ｓ１１５）。すなわち、ステップＳ１１３は、図６のステップＳ２２、Ｓ２３に対応し、ステップＳ１１４は、図６のステップＳ２４、Ｓ２５に対応し、ステップＳ１１５は、図６のステップＳ２６に対応する。

【０１２６】

ＣＰＵ１００～１０３全てのスリープ後、ＢＩＳＴコントローラ１１は、共通回路１２０のランタイムテストを実行する（Ｓ１１６）。このとき、上述したように、初期化マスク回路１２２ｂがＬ２キャッシュコントローラ１２３によるＬ２キャッシュメモリ１２４の更新を抑止し、初期化マスク回路１２２ｃがスヌープ制御ユニット１２５によるＳＣＵタグＲＡＭ１２６の更新を抑止する。

【０１２７】

ＢＩＳＴコントローラ１１は、共通回路１２０のランタイムテストの実行終了後、リセットコントローラ１３を介して共通回路１２０をリセットする（Ｓ１１７）。すなわち、ステップＳ１１６は、図６のステップＳ２７～Ｓ２９に対応し、ステップＳ１１７は、図６のステップＳ３０に対応する。このときには、上述したように、初期化マスク回路１２２ａがＬ２キャッシュコントローラ１２３及びスヌープ制御ユニット１２５によるＬ２キャッシュメモリ１２４及びＳＣＵタグＲＡＭ１２６の初期化を抑止する。

【０１２８】

これらの初期化マスク回路１２２ａ～１２２ｃによる動作によれば、ＳＣＵタグＲＡＭ１２６の内容が保持されるため、ＣＰＵ１００～１０３及び共通回路１２０が処理を再開した際におけるＬ１キャッシュメモリ１１５のキャッシュコヒーレンシを保証することができる。また、Ｌ２キャッシュメモリ１２４の内容が保持されるため、ＣＰＵ１００～１０３及び共通回路１２０が処理を再開した際におけるＬ２キャッシュメモリ１２４のミスヒットを防止して、性能劣化を抑止することができる。言い換えると、共通回路１２０のスキャンテストを実行した場合であっても、ＣＰＵ１００～１０３及び共通回路１２０が、Ｌ２キャッシュメモリ１２４及びＳＣＵタグＲＡＭ１２６に格納されたデータを利用して処理を継続することができる。

【０１２９】

ＢＩＳＴコントローラ１１は、共通回路１２０のランタイムテスト実行後に、割り込み

コントローラ１４を介してＣＰＵ１００に対して割り込みを通知する。ＣＰＵ１００は、ＢＩＳＴコントローラ１１に起因する割り込みが通知されると、スリープ状態から起床し、ＢＩＳＴコントローラ１１からの割り込みに応じた割り込みハンドラを実行する。ＣＰＵ１００は、割り込みハンドラにおいて、ランタイムテスト結果の確認（Ｓ１１８）、共通回路１２０の情報の復帰（Ｓ１１９）を行う。すなわち、ステップＳ１１８は、図６のステップＳ３１～Ｓ３３に対応し、ステップＳ１１９は、図６のステップＳ３４、Ｓ３５に対応する。また、ＣＰＵ１００は、割り込みハンドラにおいて、他の全てのＣＰＵ１０１～１０３のそれぞれに対してＣＰＵ割り込みを通知する（Ｓ１２０）。すなわち、ステップＳ１２０は、図６のステップＳ３６に対応する。そして、ＣＰＵ１００は、割り込みハンドラを終了し（Ｓ１２１）、ランタイムテストの実行前に実行していた処理を再開する（Ｓ１２２）。ＣＰＵ１０１～１０３のそれぞれは、ＣＰＵ１００からＣＰＵ間割り込みが通知されると、スリープ状態から起床して（Ｓ１３２）、その処理を再開する（Ｓ１３３）。

【０１３０】

以上に説明したように、本実施の形態１では、図２８に示すように、半導体装置９０は、記憶回路９１に格納されたデータを利用して処理を実行するとともに、処理の実行に応じて記憶回路にデータを書き込む処理回路９２に対するスキャンテストをスキャンテスト回路９４が実行しているときに、抑止回路９３が処理回路９２から記憶回路９１に対するデータの書き込みを抑止するようにしている。そのため、スキャンテストを実行したとしても、テスト対象の処理回路９２がアクセスする記憶回路９１に格納されるデータの変更を抑止することができる。なお、半導体装置９０は、半導体装置１に対応する。記憶回路９１は、Ｌ２キャッシュメモリ１２４及びＳＣＵタグＲＡＭ１２６に対応する。処理回路９２は、ＣＰＵ１００～１０３、Ｌ２キャッシュコントローラ１２３及びスヌープ制御ユニット１２５に対応する。スキャンテスト回路９４は、ＢＩＳＴコントローラ１１、１１０～１１３、１２１に対応する。抑止回路９３は、初期化マスク回路１２２に対応する。

【０１３１】

より具体的には、本実施の形態１では、抑止回路９３（初期化マスク回路１２２ｂに対応）は、Ｌ２キャッシュコントローラ１２３を有する共通回路１２０に対するスキャンテストを実行しているときに、Ｌ２キャッシュコントローラ１２３に対し、Ｌ２キャッシュメモリ１２４に対するデータの書き込みを抑止するようにしている。これによれば、Ｌ２キャッシュメモリ１２４の内容を保持することができるため、スキャンテストの終了後に、Ｌ２キャッシュメモリ１２４に格納されているデータについてはメモリ２１からデータを再取得する必要が無くなり、性能劣化を低減することができる。

【０１３２】

また、本実施の形態１では、キャッシュコヒーレンシを保証するコヒーレンシ制御回路（スヌープ制御回路１２５に対応）を有する共通回路１２０に対するスキャンテストを実行しているときに、コヒーレンシ制御回路に対し、キャッシュコヒーレンシの保証に利用する管理情報が格納された管理情報記憶回路（ＳＣＵタグＲＡＭ１２６に対応）に対するデータの書き込みを抑止するようにしている。これによれば、管理情報記憶回路の内容を保持することができるため、スキャンテストの終了後に、キャッシュコヒーレンシを保証することができる。

【０１３３】

また、本実施の形態１では、複数の演算回路（ＣＰＵ１００～１０３に対応）のそれぞれは、自身が有するＬ１キャッシュメモリ１１５のデータをその下位のメモリ（Ｌ２キャッシュメモリ１２４、ＤＤＲメモリ２０、メモリ２１、内蔵メモリ５０に対応）にフラッシュするようにしている。そのため、複数の演算回路間におけるキャッシュコヒーレンシを保証することができる。

【０１３４】

また、本実施の形態１では、複数の演算回路の一は、他の演算回路からの割り込みに応じてスキャンテスト回路にスキャンテストの実行を指示し、当該指示に応じたスキャンテ

ストの実行後に他の演算回路に割り込みを通知するようにしている。これによれば、複数の演算回路間で、演算回路のスキャンテストを時間的に重複することなく実行することができる。言い換えると、スキャンテストを実施していない演算回路で、システムの処理を継続することができる。よって、システム全体の性能劣化を抑制することができる。

【０１３５】

＜実施の形態２＞

続いて、実施の形態２について説明する。以下の実施の形態２の説明では、上述した実施の形態１と同様の内容については、同一の符号を付す等して、適宜、その説明を省略する。図１２に示すように、本実施の形態２に係る半導体装置２は、実施の形態１に係る半導体装置１と比較して、さらに、タイマ２２を有する。

【０１３６】

タイマ２２は、ランタイムテスト開始時点からの時間の経過を計測し、ランタイムテストが終了すべきとして予め定めた時間を超えたタイミングを、タイムアウトとして割り込みによってＣＰＵクラスタ１０に通知する回路である。

【０１３７】

すなわち、ランタイムテストは、毎回正常終了するとは限らない。ＢＩＳＴコントローラ１１、１１０～１１３、１２１の故障によってランタイムテストが正常終了しない可能性も考えられる。よって、ランタイムテストが終了すべき時間が経過してもランタイムテストが終了しなかった場合に、それをタイマ２２によって通知することを可能とする。

【０１３８】

ここで、タイマ２２でタイムアウトの計測対象とする期間は、予め任意の期間を定めるようにしてよい。例えば、図７を参照して説明すると、（１）ＣＰＵ１００～１０３及び共通回路１２０のそれぞれのランタイムテスト開始時からランタイムテスト終了時までの期間としてもよく、（２）ＣＰＵ１０３のランタイムテスト開始時から共通回路１２０のランタイムテスト終了時までの期間としてもよい。

【０１３９】

（１）の期間の場合

ＣＰＵ１００～１０３のそれぞれは、自身のランタイムテストを実行する場合、例えば、図８～１０のステップＳ１０１又はＳ１０２のタイミングで、システムバスを介してタイマ２２の制御レジスタ（図示せず）に対して時間の計測を開始（タイマをセット）するように設定する。タイマ２２は、この設定に応じて、時間の計測を開始する。また、ＣＰＵ１００～１０３のそれぞれは、例えば、図８～１０のステップＳ１０７～Ｓ１０９のいずれかのタイミングでシステムバスを介してタイマ２２の制御レジスタに対して時間の計測を停止（タイマを解除）するように設定する。タイマ２２は、この設定に応じて、時間の計測を停止する。

【０１４０】

また、ＣＰＵ１００～１０３のそれぞれは、例えば、タイマ２２からの割り込みに応じた割り込み信号の通知先を、自身以外のＣＰＵのいずれか又は全てのＣＰＵ１００～１０３とするように、システムバスを介して割り込みコントローラ１４を設定する。これにより、ＣＰＵ１００～１０３のそれぞれがランタイムテストを実行中であっても、他のＣＰＵによってタイムアウトに応じた処理を即座に実行可能とする。

【０１４１】

ＣＰＵ１００は、共通回路１２０のランタイムテストを実行する場合、例えば、図１１のステップＳ１１２～Ｓ１１４のいずれかのタイミングで、システムバスを介してタイマ２２の制御レジスタに対して時間の計測を開始（タイマをセット）するように設定する。タイマ２２は、この設定に応じて、時間の計測を開始する。また、ＣＰＵ１００は、例えば、図１１のステップＳ１１８～Ｓ１２０のいずれかのタイミングでシステムバスを介してタイマ２２の制御レジスタに対して時間の計測を停止（タイマを解除）するように設定する。タイマ２２は、この設定に応じて、時間の計測を停止する。

【０１４２】

また、ＣＰＵ１００は、例えば、タイマ２２からの割り込みに応じた割り込み信号の通知先を、ＣＰＵ１００～１０３のいずれか又はＣＰＵ１００～１０３の全てとなるように、システムバスを介して割り込みコントローラ１４を設定する。ただし、ＣＰＵ１００は、図６のステップＳ２５において、通知先のＣＰＵに対しては、タイマ２２からの割り込みに対して割り込み信号を出力するように割り込みコントローラ１４を設定する。また、ＣＰＵ１００は、図６のステップＳ２５において、通知先のＣＰＵに対しては、割り込み信号をマスクしないように割り込みマスク回路１９を設定する。

【０１４３】

（２）の期間の場合

ＣＰＵ１０３は、例えば、図８のステップＳ１０１又はＳ１０２のタイミングで、システムバスを介してタイマ２２の制御レジスタに対して時間の計測を開始（タイマをセット）するように設定する。タイマ２２は、この設定に応じて、時間の計測を開始する。また、ＣＰＵ１００は、例えば、図１１のステップＳ１１８～Ｓ１２０のいずれかのタイミングでシステムバスを介してタイマ２２の制御レジスタに対して時間の計測を停止（タイマを解除）するように設定する。タイマ２２は、この設定に応じて、時間の計測を停止する。

【０１４４】

また、ＣＰＵ１０３は、例えば、タイマ２２からの割り込みに応じた割り込み信号の通知先を、ＣＰＵ１００～１０３のいずれか又はＣＰＵ１００～１０３の全てとなるように、システムバスを介して割り込みコントローラ１４を設定する。ただし、ＣＰＵ１００は、図６のステップＳ２５において、通知先のＣＰＵに対しては、タイマ２２からの割り込みに対して割り込み信号を出力するように割り込みコントローラ１４を設定する。また、ＣＰＵ１００は、図６のステップＳ２５において、通知先のＣＰＵに対しては、割り込み信号をマスクしないように割り込みマスク回路１９を設定する。また、通知先のＣＰＵをＣＰＵ１００～１０３のいずれか１つとしてしまうと、その通知先のＣＰＵがランタイムテストを実行中である場合は、通知先のＣＰＵは割り込み信号を取得することができない。そのため、（２）の期間とする場合には、好ましくは、少なくとも２つ以上のＣＰＵを通知先として設定するとよい。

【０１４５】

ＣＰＵ１００～１０３のそれぞれは、タイマ２２のタイムアウトに応じた割り込み信号が入力された場合、故障対策ルーチンを実行する。故障対策ルーチンとして、例えば、半導体装置１におけるシステム全体をシャットダウンする処理を実行してもよい。

【０１４６】

以上に説明したように、本実施の形態２では、半導体装置２は、ランタイムテストの実行時間が所定時間を超えた場合にタイムアウトを通知するタイマ２２を備えている。そして、処理回路９２（ＣＰＵ１００～１０３、Ｌ２キャッシュコントローラ１２３及びスヌープ制御ユニット１２５に対応）は、処理回路９２に対するスキャンテストを開始するときにタイマ２２を設定し、処理回路９２に対するスキャンテストが終了したときにタイマ２２を解除する。これによれば、スキャンテストを実行するスキャンテスト回路（ＢＩＳＴコントローラ１１、１１０～１１３、１２１に対応）が故障していることにより、スキャンテストが停止している場合に、その故障を検出することができる。

【０１４７】

＜実施の形態３＞

続いて、実施の形態３について説明する。以下の実施の形態３の説明では、上述した実施の形態１、２と同様の内容については、同一の符号を付す等して、適宜、その説明を省略する。図１３に示すように、本実施の形態３に係る半導体装置３は、実施の形態２に係る半導体装置２と比較して、さらに、スヌープ制御システムユニット２３を有する。また、本実施の形態３に係る半導体装置３は、複数のＣＰＵクラスタ１０、７０を有する。また、本実施の形態３に係るＣＰＵクラスタ１０は、実施の形態２に係るＣＰＵクラスタ１０と比較して、さらに、スヌープマスク回路１５１を有する。なお、ＣＰＵクラスタ７０

の構成は、ＣＰＵクラスタ１０と同様であるため、その説明は省略する。また、図１３では、複数のＣＰＵクラスタ１０、７０の数が２つである例について説明しているが、ＣＰＵクラスタの数は、この例に限られない。

【０１４８】

スヌープ制御システムユニット２３は、システムバスに接続されている。スヌープ制御システムユニット２３は、複数のＣＰＵクラスタ１０、７０間におけるＬ１キャッシュメモリ１１５及びＬ２キャッシュメモリ１２４のキャッシュコヒーレンシを、スヌープ方式で保証する制御を行う。

【０１４９】

本実施の形態３では、実施の形態１、２と比較して、共通回路１２０のＬ２キャッシュコントローラ１２３は、ＣＰＵ１００～１０３からリードが要求されたデータが、Ｌ２キャッシュメモリ１２４、スヌープ制御ユニット１２５のいずれから取得できなかった場合、そのデータのリードを要求するリード要求信号をスヌープ制御システムユニット２３に出力する点が異なる。

【０１５０】

スヌープ制御システムユニット２３は、複数のＣＰＵクラスタ１０、７０のいずれかからリード要求信号が入力された場合（ここではＣＰＵクラスタ１０とする）、そのリード要求信号で要求されたデータが、要求元のＣＰＵクラスタ１０以外のＣＰＵクラスタ７０に格納されているか否かを判定する。スヌープ制御システムユニット２３は、要求元のＣＰＵクラスタ１０以外のＣＰＵクラスタ７０にデータが格納されていると判定した場合、そのデータを有するＣＰＵクラスタ７０に対してそのデータを要求する。より具体的には、スヌープ制御システムユニット２３は、ＣＰＵクラスタ７０に対して、そのデータを要求するスヌープ要求信号を出力する。ＣＰＵクラスタ７０の共通回路１２０のＬ２キャッシュコントローラ１２３は、そのスヌープ要求信号に応じて、そのスヌープ要求信号で要求されたデータをＬ２キャッシュメモリ１２４から取得する。

【０１５１】

一方、Ｌ２キャッシュメモリ１２４にデータが存在しない場合、そのＣＰＵクラスタ７０の共通回路１２０のスヌープ制御ユニット１２５は、そのＣＰＵクラスタ７０内において、そのデータを有するＣＰＵに対して、そのデータを要求するスヌープ要求信号を出力する。該ＣＰＵのＬ１キャッシュコントローラ１１４は、そのスヌープ要求信号に応じて、そのスヌープ要求信号で要求されたデータをＬ１キャッシュメモリ１１５から取得し、そのデータを含むスヌープ応答信号を共通回路１２０に出力する。スヌープ制御ユニット１２５は、そのスヌープ応答信号に含まれるデータを取得する。そして、Ｌ２キャッシュコントローラ１２３は、Ｌ２キャッシュメモリ１２４から、又は、スヌープ制御ユニット１２５の制御によって取得したデータを含むスヌープ応答信号をスヌープ制御システムユニット２３に出力する。

【０１５２】

スヌープ制御システムユニット２３は、ＣＰＵクラスタ７０から出力されたスヌープ応答信号に含まれるデータを取得する。スヌープ制御システムユニット２３は、取得したデータを含むリード応答信号を、リード要求信号を出力したＣＰＵクラスタ１０に対して出力する。これにより、リード要求信号を出力したＣＰＵクラスタ１０が、そのリード要求信号によってリードを要求したデータを取得することができる。

【０１５３】

一方、スヌープ制御システムユニット２３は、ＣＰＵクラスタ７０からデータを取得できなかった場合、そのリード要求信号を、システムバスを介して外部バスコントローラ１７に出力する。外部バスコントローラ１７は、このリード要求信号に応じて、メモリ２１から取得したデータを含むリード応答信号を、システムバスを介してスヌープ制御システムユニット２３に出力する。スヌープ制御システムユニット２３は、このリード応答信号に含まれるデータを取得する。

【０１５４】

スヌープ制御システムユニット２３は、他のＣＰＵクラスタ７０、もしくは、メモリ２１から取得したデータを、要求元の共通回路１２０に出力する。より具体的には、スヌープ制御システムユニット２３は、取得したデータを含むリード応答信号を、要求元のＣＰＵ１００に出力する。

【０１５５】

また、スヌープ制御システムユニット２３は、複数のＣＰＵクラスタ１０、７０のいずれかからライト要求信号が入力された場合（ここではＣＰＵクラスタ１０とする）、そのライト要求信号で要求されたデータが、要求元のＣＰＵクラスタ１０以外のＣＰＵクラスタ７０に格納されているか否かを判定する。スヌープ制御システムユニット２３は、要求元のＣＰＵクラスタ１０以外のＣＰＵクラスタ７０にデータが格納されていると判定した場合、そのデータを有するＣＰＵクラスタ７０に対してそのデータの無効化を要求する。より具体的には、スヌープ制御システムユニット２３は、ＣＰＵクラスタ７０に対して、そのデータの無効化を要求するスヌープ要求信号を出力する。ＣＰＵクラスタ７０の共通回路１２０のＬ２キャッシュコントローラ１２３は、そのスヌープ要求信号に応じて、Ｌ２キャッシュメモリ１２４において、そのスヌープ要求信号で要求されたデータを無効化する。すなわち、このデータは、Ｌ２キャッシュメモリ１２４から削除され、Ｌ２キャッシュメモリ１２４には格納されていない扱いとなる。

【０１５６】

また、そのＣＰＵクラスタ７０の共通回路１２０のスヌープ制御ユニット１２５は、そのＣＰＵクラスタ７０内において、そのデータを有するＣＰＵに対して、そのデータの無効化を要求するスヌープ要求信号を出力する。該ＣＰＵのＬ１キャッシュコントローラ１１４は、そのスヌープ要求信号に応じて、Ｌ１キャッシュメモリ１１５において、スヌープ要求信号で無効化が要求されたデータを無効化する。

【０１５７】

また、スヌープ制御システムユニット２３は、そのライト要求信号を外部バスコントローラ１７に出力し、メモリ２１に対するデータの格納も実行する。

【０１５８】

ここで、要求元のＣＰＵクラスタ１０以外のＣＰＵクラスタ７０に格納されているか否かは、複数のＣＰＵクラスタ１０、７０に格納されているデータのそれぞれが、メモリ２１におけるどのアドレスのデータであるかを示す情報が格納されるＳＣＵタグＲＡＭ（図示せず）をスヌープ制御システムユニット２３が有するようにし、スヌープ制御ユニット１２５及びＳＣＵタグＲＡＭ１２６と同様に制御することで判定するようにしてよい。また、スヌープ制御システムユニット２３がＳＣＵタグＲＡＭを有さない場合には、要求元のＣＰＵクラスタ１０以外の他の全てのＣＰＵクラスタ７０にスヌープ要求信号を出力するようにすればよい。

【０１５９】

なお、以上の説明では、ＣＰＵクラスタ１０からのリード要求及びライト要求に対して制御を行う例について説明したが、当然に、ＣＰＵクラスタ７０からのリード要求及びライト要求に対しても同様に制御を行うことができる。また、以上の説明では、メモリ２１のデータについてスヌープ制御システムユニット２３が制御を行う例について説明したが、当然に、ＤＤＲメモリ２０及び内蔵メモリ５０のデータも同様に制御されるようにしてもよい。以降の説明においても同様である。

【０１６０】

スヌープマスク回路１５１は、スヌープ制御システムユニット２３から共通回路１２０に対して入力されるスヌープ要求信号をマスク（遮断）する回路である。スヌープマスク回路１５１がスヌープ要求信号をマスク（遮断）するか否かは、スヌープマスク回路１５１に対して任意に設定することができる。この設定は、スヌープマスク回路１５１もしくはクロックコントローラ１２が有する制御レジスタ（図示せず）に対して、スヌープ要求信号をマスクするか否かを示す値を設定することで行われる。また、その設定も、ＢＩＳＴコントローラ１２１からの制御により、任意のタイミングで変更することが可能である

。
【０１６１】
図６を参照して説明すると、ステップＳ２７のタイミングで、ＢＩＳＴコントローラ１２１は、スヌープ要求信号をマスクするようにスヌープマスク回路１５１を設定する。また、ステップＳ２９のタイミングで、ＢＩＳＴコントローラ１２１は、スヌープ要求信号のマスクを解除するようにスヌープマスク回路１５１を設定する。なお、このマスクの解除により、マスクされていたスヌープ要求信号の処理が共通回路１２０によって開始される。
【０１６２】
これによれば、ＣＰＵクラスタ１０の共通回路１２０のランタイムテストを実行中に、他のＣＰＵクラスタ７０からスヌープ要求信号によってランタイムテスト実行中の共通回路１２０が意図せぬ動作を行ってしまうことを防止することができる。なお、スヌープ要求信号は、スヌープマスク回路１５１によるマスクが解除されるまで待たされた後に正常に処理が行われる。
【０１６３】
なお、以上の説明では、実施の形態３として、実施の形態２に対して、さらに、スヌープ制御システムユニット２３及びスヌープマスク回路１５１を有するようにし、複数のＣＰＵクラスタ１０、７０を有するようにした形態について説明したが、これに限られない。実施の形態１に対して、これらの構成を適用するようにしてもよい。
【０１６４】
以上に説明したように、本実施の形態３では、半導体装置３は、複数の処理回路９２（ＣＰＵ１００～１０３、Ｌ２キャッシュコントローラ１２３及びスヌープ制御ユニット１２５に対応）を備えるようにしている。そして、複数の処理回路９２のそれぞれは、第１のコヒーレンシ制御回路（スヌープ制御ユニット１２５に対応）を有する共通回路１２０に対するスキャンテストを実行しているときに、キャッシュコヒーレンシを保証するために、第２のコヒーレンシ制御回路（スヌープ制御システムユニット２３に対応）から第１のコヒーレンシ制御回路を有する共通回路１２０に出力される信号を遮断するマスク回路（スヌープマスク回路１５１に対応）を含んでいる。これによれば、ランタイムテスト実行中は、処理回路９２（ＣＰＵクラスタ１０、７０）に関連するトランザクションを中断でき、ランタイムテストの終了後、トランザクションを再開することで、システムに不都合を生じさせないようにすることができる。
【０１６５】
＜実施の形態４＞
続いて、実施の形態４について説明する。以下の実施の形態４の説明では、上述した実施の形態１～３と同様の内容については、同一の符号を付す等して、適宜、その説明を省略する。
【０１６６】
本実施の形態４では、ＣＰＵ１００～１０３及び共通回路１２０のそれぞれのランタイムテストを一括で実施せず、分割して実施する点が、実施の形態１～３と異なる。実施の形態４に係る半導体装置４は、実施の形態３に係る半導体装置３と比較して、さらに、タイマ２５を有する。以下、ＤＴＩにおけるランタイムテストを分割した単位のそれぞれを「分割テスト」と呼ぶ。
【０１６７】
タイマ２５は、分割テストの終了時点からの時間の経過を計測し、次の分割テストを開始すべきとして予め定めた時間を経過した時点で、自身又は共通回路１２０のランタイムテストを分割して実行中のＣＰＵに対して、分割テストの開始タイミングとなったことを割り込みによって通知する回路である。
【０１６８】
続いて、図１５を参照して、本実施の形態４に係るランタイムテストの分割例について説明する。ランタイムテストは、予め任意に定めた数に分割して実施してよい。図１５の

例では、ランタイムテストを３つの分割テストに分割して実施する例について示している。

【０１６９】

本実施の形態４では、ＤＴＩ中のランタイムテストに使用する複数のテストパタンを分割した複数の単位のそれぞれについてのランタイムテストを、分割テストとして実施する。例えば、図１５に示すように、ランタイムテスト対象の回路に２つの論理回路と３つの記憶回路とが回路に含まれる場合、そのランタイムテストに使用される複数のテストパタンを、１つの論理回路のスキャンテスト（図１５の「ＳＣＡＮ－１」）に使用される単位と、もう１つの論理回路のスキャンテスト（図１５の「ＳＣＡＮ－２」）に使用される単位と、３つの記憶回路のＭＢＩＳＴ（図１５の「ＭＢＩＳＴ」）に使用される単位とに分割するようにしてもよい。そして、図１５に示すように、それぞれの単位について、分割テストを実行する。

【０１７０】

なお、１回の分割テストでスキャンテストが実行される論理回路の数と、１回の分割テストでＭＢＩＳＴが実行される記憶回路の数は、図１５の例に限られない。また、ある分割テストでスキャンテストが実行される論理回路の数と、他の分割テストでスキャンテストが実行される論理回路の数は異なっていてもよい。同様に、ある分割テストでＭＢＩＳＴが実行される記憶回路の数と、他の分割テストでＭＢＩＳＴが実行される記憶回路の数も異なっていてもよい。

【０１７１】

あるいは、ランタイムテスト対象の回路に含まれる論理回路全体を一通りスキャンテストするために使用されるテストパタン群を複数の単位に分割し、分割した単位のそれぞれを１回の分割テストでのスキャンテストに使用してもよい。また、ランタイムテスト対象の回路に含まれる記憶回路に対し、一通りＭＢＩＳＴを行うために使用されるテストパタン群を複数の単位に分割し、分割した単位のそれぞれを１回の分割テストでのＭＢＩＳＴに使用してもよい。

【０１７２】

ここで、分割テストのそれぞれにおいて、どれだけの数のテストパタンを使用するかは、ＣＰＵ１００～１０３のそれぞれが、システムバスを介して、ＢＩＳＴコントローラ１１が有する制御レジスタに設定することでＢＩＳＴコントローラ１１が認識可能とすればよい。

【０１７３】

例えば、図１５に示すように、ＣＰＵ１００について、２つの論理回路と１つの記憶回路のそれぞれに対して３つの分割テストが実施されるものとする。この場合、ＣＰＵ１００は、１回目の分割テスト開始前には、図５のステップＳ１において１つ目の論理回路のランタイムテストで使用するテストパタン数を、システムバスを介してＢＩＳＴコントローラ１１の制御レジスタに設定する。また、ＣＰＵ１００は、２回目の分割テスト開始前には、図５のステップＳ１において２つ目の論理回路のランタイムテストで使用するテストパタン数を、システムバスを介してＢＩＳＴコントローラ１１の制御レジスタに設定する。そして、３回目の分割テスト開始前には、図５のステップＳ１において記憶回路のランタイムテストで使用するテストパタン数を、システムバスを介してＢＩＳＴコントローラ１１の制御レジスタに設定する。そして、ＢＩＳＴコントローラ１１０は、ＢＩＳＴコントローラ１１の制御レジスタに設定されたテストパタン数だけテストパタンを生成し、ランタイムテストを実行する。

【０１７４】

続いて、図１６を参照して、本実施の形態４に係るＣＰＵクラスタ１０のランタイムテスト実行時の動作について説明する。図１６の例では、ランタイムテストを２つの分割テストに分割して実施する例について示している。

【０１７５】

リアルタイム応答性が求められるシステムでは、テスト時間が割り込み応答制約時間を

上回ることがあってはならない。割り込み応答制約時間とは、割り込みが発生してから少なくともその時間を超える前に、ＣＰＵ１００～１０３が割り込みを検出して、割り込みに応じた処理を開始しなければならない時間である。そこで、本実施の形態４では、図１６に示すように、ＣＰＵ１００～１０３及び共通回路１２０のそれぞれのランタイムテストを、割り込み応答制約時間以下の時間で実施できる分割テストに分割して実行する。

【０１７６】

ＣＰＵ１０３は、タイマ１５からの割り込みに応じて、図５に示した手順に従って、自身の処理を中断し、自身の１回目の分割テストを実行する。このとき、他のＣＰＵ１００～１０２のそれぞれは、その処理の実行を継続する。ＣＰＵ１０３は、１回目の分割テストの実行終了後、自身の処理を再開し、システムバスを介してタイマ２５に対して時間の計測を開始（タイマをセット）するように設定する。タイマ２５は、この設定に応じて、時間の計測を開始する。

【０１７７】

ＣＰＵ１０３は、タイマ２５からの割り込みに応じて、図５に示した手順に従って、自身の処理を中断し、自身の２回目の分割テストを実行する。このとき、他のＣＰＵ１００～１０２のそれぞれは、その処理の実行を継続する。ＣＰＵ１０３は、２回目の分割テストの実行終了後、自身の処理を再開し、ＣＰＵ１０２に対してＣＰＵ間割り込みを通知する。

【０１７８】

ＣＰＵ１０２は、ＣＰＵ１０３からのＣＰＵ間割り込みに応じて、図５に示した手順に従って、自身の処理を中断し、自身の１回目の分割テストを実行する。このとき、他のＣＰＵ１００～１０１、１０３のそれぞれは、その処理の実行を継続する。ＣＰＵ１０２は、１回目の分割テストの実行終了後、自身の処理を再開し、システムバスを介してタイマ２５に対して時間の計測を開始（タイマをセット）するように設定する。タイマ２５は、この設定に応じて、時間の計測を開始する。

【０１７９】

ＣＰＵ１０２は、タイマ２５からの割り込みに応じて、図５に示した手順に従って、自身の処理を中断し、自身の２回目の分割テストを実行する。このとき、他のＣＰＵ１００～１０１、１０３のそれぞれは、その処理の実行を継続する。ＣＰＵ１０２は、２回目の分割テストの実行終了後、自身の処理を再開し、ＣＰＵ１０１に対してＣＰＵ間割り込みを通知する。

【０１８０】

ＣＰＵ１０１は、ＣＰＵ１０２からのＣＰＵ間割り込みに応じて、図５に示した手順に従って、自身の処理を中断し、自身の１回目の分割テストを実行する。このとき、他のＣＰＵ１００、１０２～１０３のそれぞれは、その処理の実行を継続する。ＣＰＵ１０１は、１回目の分割テストの実行終了後、自身の処理を再開し、システムバスを介してタイマ２５に対して時間の計測を開始（タイマをセット）するように設定する。タイマ２５は、この設定に応じて、時間の計測を開始する。

【０１８１】

ＣＰＵ１０１は、タイマ２５からの割り込みに応じて、図５に示した手順に従って、自身の処理を中断し、自身の２回目の分割テストを実行する。このとき、他のＣＰＵ１００、１０２～１０３のそれぞれは、その処理の実行を継続する。ＣＰＵ１０１は、２回目の分割テストの実行終了後、自身の処理を再開し、ＣＰＵ１００に対してＣＰＵ間割り込みを通知する。

【０１８２】

ＣＰＵ１００は、ＣＰＵ１０１からのＣＰＵ間割り込みに応じて、図５に示した手順に従って、自身の処理を中断し、自身の１回目の分割テストを実行する。このとき、他のＣＰＵ１０１～１０３のそれぞれは、その処理の実行を継続する。ＣＰＵ１００は、１回目の分割テストの実行終了後、自身の処理を再開し、システムバスを介してタイマ２５に対して時間の計測を開始（タイマをセット）するように設定する。タイマ２５は、この設定

に応じて、時間の計測を開始する。
【０１８３】
ＣＰＵ１００は、タイマ２５からの割り込みに応じて、図５に示した手順に従って、自身の処理を中断し、自身の２回目の分割テストを実行する。このとき、他のＣＰＵ１０１～１０３のそれぞれは、その処理の実行を継続する。
【０１８４】
ＣＰＵ１００は、２回目の分割テストの実行終了後、図６に示した手順に従って、共通回路１２０の１回目の分割テストを実行する。このとき、共通回路１２０を共有する全てのＣＰＵ１００～１０３のそれぞれは、上述したようにスリープしてその処理を中断する。ＣＰＵ１００は、１回目の分割テストの実行終了後、自身の処理を再開し、システムバスを介してタイマ２５に対して時間の計測を開始（タイマをセット）するように設定する。タイマ２５は、この設定に応じて、時間の計測を開始する。
【０１８５】
ＣＰＵ１００は、タイマ２５からの割り込みに応じて、図６に示した手順に従って、共通回路１２０の２回目の分割テストを実行する。このとき、共通回路１２０を共有する全てのＣＰＵ１００～１０３のそれぞれは、上述したようにスリープしてその処理を中断する。ＣＰＵ１００は、２回目の分割テストの実行終了後、自身の処理を再開する。
【０１８６】
以上に説明したように、ランタイムテストがＮ個の分割テストに分割される場合（Ｎは２以上の正整数）、ＣＰＵ１０３は、タイマ１５からの割り込みに応じて１回目の分割テストを実行し、タイマ２５からの割り込みに応じて２～Ｎ回目の分割テストを実行する。また、ＣＰＵ１０３は、１～（Ｎ－１）回目の分割テストの実行後にはタイマ２５を設定し、Ｎ回目の分割テストの実行後には次のテスト対象のＣＰＵ１０２に対してＣＰＵ間割り込みを通知する。
【０１８７】
ＣＰＵ１０２、１０１のそれぞれは、直前にランタイムテストが実行されたＣＰＵからのＣＰＵ間割り込みに応じて１回目の分割テストを実行し、タイマ２５からの割り込みに応じて２～Ｎ回目の分割テストを実行する。また、ＣＰＵ１０２、１０１のそれぞれは、１～（Ｎ－１）回目の分割テストの実行後にはタイマ２５を設定し、Ｎ回目の分割テストの実行後には次のランタイムテスト対象のＣＰＵ１０１又は１００に対してＣＰＵ間割り込みを通知する。
【０１８８】
ＣＰＵ１００は、直前にランタイムテストが実行されたＣＰＵ１０１からのＣＰＵ間割り込みに応じて１回目の分割テストを実行し、タイマ２５からの割り込みに応じて２～Ｎ回目の分割テストを実行する。また、ＣＰＵ１００は、１～（Ｎ－１）回目の分割テストの実行後にはタイマ２５を設定し、Ｎ回目の分割テストの実行後には、次のランタイムテスト対象の共通回路１２０のランタイムテストを実行する制御を行う。
【０１８９】
ＣＰＵ１００は、自身のランタイムテスト後に共通回路１２０の１回目の分割テストを実行し、タイマ２５からの割り込みに応じて共通回路１２０の２～Ｎ回目の分割テストを実行する。また、ＣＰＵ１００は、１～（Ｎ－１）回目の分割テストの実行後にはタイマ２５を設定する。
【０１９０】
なお、ＣＰＵ１００～１０３のそれぞれは、分割テストの実行数をカウントすることで、１～（Ｎ－１）回目の分割テストの実行後か、Ｎ回目の分割テストの実行後かを判定するようにすればよい。分割テストの実行数は、例えば、ＤＤＲメモリ２０、メモリ２１、又は、内蔵メモリ５０に格納することで保持するようにすればよい。
【０１９１】
続いて、図１７を参照して、実施の形態４に係るＣＰＵ１００～１０３の１回目の分割テスト実行時の動作について説明する。図１７では、図８～図１０における処理と同様の

処理については同一の符号を付し、適宜その説明を省略する。

【０１９２】

ＣＰＵ１００～１０３のそれぞれは、その処理の実行中に（Ｓ１００）、割り込みが通知されると、その割り込みに応じた割り込みハンドラを実行する。ＣＰＵ１００～１０３のそれぞれは、図８～図１０と同様に、割り込みに応じた割り込みハンドラにおいてステップＳ１０１～Ｓ１０３の処理を実行する。この割り込みは、ＣＰＵ１０３であればタイマ１５からの割り込みとなり、ＣＰＵ１０２～１００であれば直前にランタイムテストを実行したＣＰＵ１０３～１０１からのＣＰＵ間割り込みとなる。

【０１９３】

一方、ＣＰＵ１００～１０３のそれぞれは、リセットハンドラにおけるステップＳ１０６～Ｓ１０８の実行後、タイマ２５を設定してから（Ｓ１４０）、ステップＳ１１０、Ｓ１１１を実行する点が、図８～図１０とは異なる。

【０１９４】

続いて、図１８を参照して、実施の形態４に係るＣＰＵ１００～１０３の２～N回目の分割テスト実行時の動作について説明する。図１８では、図８～図１０、１７における処理と同様の処理については同一の符号を付し、適宜その説明を省略する。

【０１９５】

ＣＰＵ１００～１０３のそれぞれは、その処理の実行中に（Ｓ１００）、タイマ２５から割り込みが通知されると、その割り込みに応じた割り込みハンドラを実行する。ＣＰＵ１００～１０３のそれぞれは、図８～図１０、１７と同様に、割り込みに応じた割り込みハンドラにおいてステップＳ１０１～Ｓ１０３の処理を実行する。

【０１９６】

ＣＰＵ１００～１０３のそれぞれは、リセットハンドラにおけるステップＳ１０６～Ｓ１０８の実行後、他のＣＰＵに対して割り込みを通知する（Ｓ１４１）。この割り込みは、ＣＰＵ１０１～１０３であれば、図８、９と同様に次のランタイムテスト実行対象のＣＰＵに対するＣＰＵ間割り込み（Ｓ１０９）となり、ＣＰＵ１００であれば、図１０と同様に他の全てのＣＰＵ１０１～１０３に対するＣＰＵ間割り込み（Ｓ１１２）となる。また、ＣＰＵ１０１～１０３であれば、ステップＳ１４１の後に、図８、９と同様にステップＳ１１０、Ｓ１１１を実行するが、ＣＰＵ１００であれば、ステップＳ１４１の後に、図１０と同様にステップＳ１１０、Ｓ１１１を実行せず、共通回路１２０のランタイムテストを実行する制御を行う。

【０１９７】

なお、ランタイムテストを３つ以上の分割テストに分割して実行する場合には、２～（Ｎ－１）回目の分割テスト実行時の動作は、図１８においてステップＳ１４１に代えてステップＳ１４０を実行する動作となることは自明であるため、詳細な説明は省略する。

【０１９８】

続いて、図１９を参照して、実施の形態４に係る共通回路１２０の１回目の分割テスト実行時の動作について説明する。図１９では、図１１における処理と同様の処理については同一の符号を付し、適宜その説明を省略する。

【０１９９】

ＣＰＵ１００は、図１１と同様に、リセットハンドラにおいて、ステップＳ１１２～Ｓ１１５を実行する。一方、ＣＰＵ１００は、割り込みハンドラにおけるステップＳ１１８～Ｓ１２０の実行後、タイマ２５を設定してから（Ｓ１４２）、ステップＳ１２１、Ｓ１２２を実行する点が、図１１と異なる。

【０２００】

続いて、図２０を参照して、実施の形態４に係る共通回路１２０の２回目の分割テスト実行時の動作について説明する。図２０では、図１１、図１９における処理と同様の処理については同一の符号を付し、適宜その説明を省略する。

【０２０１】

ＣＰＵ１００は、その処理の実行中に（Ｓ１４３）、タイマ２５から割り込みが通知さ

れると、その割り込みに応じた割り込みハンドラを実行する。ＣＰＵ１００は、図１１、１９と同様に、割り込みハンドラにおいて、ステップＳ１１２～Ｓ１１５の処理を実行する。なお、リセットハンドラではなく、割り込みハンドラにおいてこれらの処理を実行する点は、図１１、１９とは異なる。以降、図１１と同様に、ステップＳ１１６～Ｓ１２２が実行される。

【０２０２】

なお、ランタイムテストを３つ以上の分割テストに分割して実行する場合には、２～（Ｎ－１）回目の分割テスト実行時の動作は、図２０においてステップＳ１２０の後にステップＳ１４２を実行する動作となることは自明であるため、詳細な説明は省略する。

【０２０３】

なお、以上の説明では、実施の形態４として、実施の形態３に対して、さらに、タイマ２５を有するようにし、分割テストを実施する形態について説明したが、これに限られない。実施の形態１又は２に対して、これらの構成を適用するようにしてもよい。

【０２０４】

以上に説明したように、本実施の形態４は、演算回路（ＣＰＵ１００～１０３のそれぞれに対応）全体のスキャンテストに使用する複数のテストパタンを分割した複数の単位のそれぞれについてのスキャンテストを分割テストとして実行する毎に、演算回路に対するスキャンテストの実行を終了するものである。また、本実施の形態４は、共通回路１２０全体のスキャンテストに使用する複数のテストパタンを分割した複数の単位のそれぞれについてのスキャンテストを分割テストとして実行する毎に、共通回路１２０に対するスキャンテストの実行を終了するものである。半導体装置４は、演算回路及び共通回路１２０のそれぞれの分割テストを終了してからの時間を計測し、その演算回路及び共通回路１２０のそれぞれの次の分割テストを開始するタイミングを演算回路に通知するタイマ２５を備えている。そして、演算回路は、タイマ２５からの通知に応じて、スキャンテスト回路（ＢＩＳＴコントローラ１１、１１０～１１３、１２１に対応）に分割テストの開始を指示するものである。

【０２０５】

これによれば、スキャンテストを複数の単位に分割した分割テストの実行を終了したタイミングで演算回路が割り込みに応じた処理を実行することができるため、演算回路による割り込み応答性能を向上することができる。よって、例えば、ＣＰＵ１００～１０３において割り込み応答性能が求められるアプリケーションプログラム（例えば、音声認識処理プログラムなど）を、その要求を満たして実行することができる。すなわち、リアルタイム性を要するシステムにおいても、ランタイムテストを実行することが可能となる。

【０２０６】

＜実施の形態５＞

続いて、実施の形態５について説明する。以下の実施の形態５の説明では、上述した実施の形態１～４と同様の内容については、同一の符号を付す等して、適宜、その説明を省略する。図２１に示すように、本実施の形態５に係る半導体装置５は、実施の形態３に係る半導体装置３と比較して、さらに、電源制御回路２６と、電源スイッチ１７０～１７３、１８０とを有する。

【０２０７】

電源制御回路２６は、ＣＰＵ１００～１０３及び共通回路１２０のそれぞれの電源状態を制御する回路である。電源制御回路２６は、ＣＰＵ１００～１０３から出力されるステータス信号を監視し、ＣＰＵ１００～１０３のそれぞれがスリープ状態に移行した場合には、ＣＰＵ１００～１０３のそれぞれの電源をＯＦＦにする。また、電源制御回路２６は、ＣＰＵ１００～１０３の全てがスリープ状態に移行した場合には、ＣＰＵ１００～１０３及び共通回路１２０の電源をＯＦＦにする。これにより、省電力を実現する。なお、電源制御回路２６は、共通回路１２０を電源ＯＦＦにする場合には、共通回路１２０のＬ２キャッシュメモリ１２４及びＳＣＵタグＲＡＭ１２６の電源はＯＮにしたままとし、そのデータの保持を維持する。

【０２０８】
電源スイッチ１７０は、ＣＰＵ１００と接続され、電源スイッチ１７１は、ＣＰＵ１０１と接続され、電源スイッチ１７２は、ＣＰＵ１０２と接続され、電源スイッチ１７３は、ＣＰＵ１０３と接続され、電源スイッチ１８０は、共通回路１２０と接続される。電源制御回路２６は、電源スイッチ１７０～１７３、１８０のそれぞれを操作することで、ＣＰＵ１００～１０３及び共通回路１２０のそれぞれの電源ＯＮ・ＯＦＦを行う。
【０２０９】
また、電源制御回路２６は、割り込みコントローラ１４からの割り込み信号の入力に応じて、ＣＰＵ１００～１０３及び共通回路１２０のうち、電源をＯＦＦにした回路の電源をＯＮにし、リセットコントローラ１３に対して電源をＯＮにした回路のリセットを要求する。
【０２１０】
この電源制御回路２６によるＣＰＵ１００～１０３及び共通回路１２０に対する、省電力状態への移行と、省電力状態からの復帰の動作は、上述のＣＰＵ１００～１０３及び共通回路１２０のランタイムテストを実行する状態への移行と、ランタイムテスト実行後の状態からの復帰の動作と相似する。そこで、本実施の形態５では、この電源制御回路２６による機能を流用することで、ＣＰＵ１００～１０３及び共通回路１２０のランタイムテストを実行する状態への移行とランタイムテスト実行後の状態からの復帰の動作を実現する。
【０２１１】
この動作を実現するために、本実施の形態５に係る電源制御回路２６は、ＣＰＵ１００～１０３のスリープ状態を検出した際に、ＣＰＵ１００～１０３及び共通回路１２０の電源をＯＦＦにする動作を抑止する設定を、ＣＰＵ１００～１０３のそれぞれから実施可能に改良されたものである。
【０２１２】
ＣＰＵ１００～１０３のそれぞれのランタイムテストを実行する場合には、ＣＰＵ１００～１０３のそれぞれは、図８～１０のステップＳ１０１又はＳ１０２のタイミングで、電源制御回路２６に対して電源をＯＦＦにする動作を抑止する設定を行い、図８～１０のステップＳ１０６～Ｓ１０８のいずれかのタイミングで電源制御回路２６に対して電源をＯＦＦにする動作の抑止を解除する設定を行う。
【０２１３】
また、共通回路１２０のランタイムテストを実行する場合には、ＣＰＵ１００は、図１１のステップＳ１１２～Ｓ１１４のいずれかのタイミングで、電源制御回路２６に対して電源をＯＦＦにする処理を抑止する設定を行い、図１１のステップＳ１１８～Ｓ１２０のいずれかのタイミングで電源制御回路２６に対して電源をＯＦＦにする処理の抑止を解除する設定を行う。
【０２１４】
これによれば、ＣＰＵ１００～１０３がスリープ状態に移行しても、ＣＰＵ１００～１０３及び共通回路１２０の電源がＯＦＦにされることはないため、ランタイムテストを実行することが可能となる。
【０２１５】
ＢＩＳＴコントローラ１１は、ＣＰＵ１００～１０３及び共通回路１２０のそれぞれのランタイムテストの実行後、リセットコントローラ１３にリセットを要求するのではなく、割り込みコントローラ１４に割り込み信号を出力する。割り込みコントローラ１４は、この割り込み信号に応じて、割り込み信号を電源制御回路２６に出力する。
【０２１６】
電源制御回路２６は、電源をＯＦＦする動作のみが抑止されているため、実際の電源状態はＯＮであるが、ＣＰＵ１００～１０３のいずれかをスリープ状態に移行させてスキャンテストを実行した場合には、そのＣＰＵの電源状態がＯＦＦであると認識している。そのため、この場合には、電源制御回路２６は、割り込み信号に応じて、スキャンテストを

実施したＣＰＵに対するリセットをリセットコントローラ１３に要求する。
【０２１７】
また、電源制御回路２６は、電源をＯＦＦする動作のみが抑止されているため、実際の電源状態はＯＮであるが、ＣＰＵ１００～１０３の全てがスリープ状態に移行した場合には、ＣＰＵ１００～１０３及び共通回路１２０の全ての電源状態がＯＦＦであると認識している。そのため、この場合には、電源制御回路２６は、割り込み信号に応じて、ＣＰＵ１００～１０３及び共通回路１２０の全てに対するリセットをリセットコントローラ１３に要求する。なお、ＣＰＵ１００～１０３に対するリセットは、上述したようにリセットマスク回路１８によって抑止される。
【０２１８】
すなわち、本実施の形態５によれば、実施の形態１～４では、ＢＩＳＴコントローラ１１がハンドリングしていたテスト対象の回路のリセットを電源制御回路２６の機能を流用することで容易に実現することができる。
【０２１９】
なお、以上の説明では、実施の形態５として、実施の形態３に対して、さらに、電源制御回路２６を有するようにした形態について説明したが、これに限られない。実施の形態１、２、４のいずれかに対して、この構成を適用するようにしてもよい。
【０２２０】
以上に説明したように、本実施の形態５では、半導体装置５は、演算回路（ＣＰＵ１００～１０３）がスリープ状態に移行したときに、演算回路の電源をＯＦＦにし、割り込みが通知されたときに、演算回路の電源をＯＮにしてリセットコントローラ１３によって演算回路をリセットする電源制御回路２６を備えている。演算回路は、演算回路に対するスキャンテストを実行する場合には、演算回路の電源ＯＦＦを抑止するように電源制御回路２６を設定してから、スリープ状態に移行する。そして、演算回路に対するスキャンテストの実行後に、電源制御回路２６に割り込みを通知するようにしている。これによれば、電源制御回路の機能を流用して、演算回路のスキャンテストを実行する状態への移行と、その状態からの復帰を実現することができる。すなわち、ランタイムテストを実現するための論理設計コストを低減することができる。
【０２２１】
＜実施の形態６＞
続いて、実施の形態６について説明する。以下の実施の形態６の説明では、上述した実施の形態１～５と同様の内容については、同一の符号を付す等して、適宜、その説明を省略する。図２２に示すように、本実施の形態６に係る半導体装置６は、実施の形態３に係る半導体装置３と比較して、ＢＩＳＴコントローラ１１、１１０～１１３のそれぞれに換えて、テストコントローラ６０、１６０～１６３のそれぞれを有する点が異なる。
【０２２２】
テストコントローラ６０は、ＢＩＳＴコントローラ１１と比較して、テストパタン及び期待値をＤＤＲメモリ２０から取得して、テストコントローラ１６０～１６３のそれぞれに供給する点が異なる。また、テストコントローラ１６０～１６３のそれぞれは、ＢＩＳＴコントローラ１１０～１１３のそれぞれと比較して、テストパタン及び期待値を生成せず、テストコントローラ６０から供給されたテストパタン及び期待値を使用する点が異なる。
【０２２３】
すなわち、実施の形態６では、ＤＤＲメモリ２０は、テストパタンと、期待値が予め格納されている。テストコントローラ６０は、ローカルバス及びシステムバスを介して、ＤＤＲメモリ２０からテストパタン及び期待値を読み出す。テストコントローラ１６０～１６３のそれぞれは、テストコントローラ６０が読み出したテストパタン及び期待値を使用してスキャンテストを実施する。なお、テストコントローラ６０及びＣＰＵ１００～１０３のいずれかが、半導体装置６の起動時に、外部バスコントローラ１７に接続されているメモリ２１からＤＤＲメモリ２０又は内蔵メモリ５０にテストパタン及び期待値を事前に

転送しておき、テストコントローラ６０は、ランタイムテストの実行時にＤＤＲメモリ２０又は内蔵メモリ５０からテストパタン及び期待値を取得するようにしてもよい。ここで、メモリ２１に代えてフラッシュメモリ等の不揮発性メモリを用いても良い。

【０２２４】

ここで、ＣＰＵ１００～１０３のそれぞれのランタイムテストを実行する場合、ＣＰＵ１００～１０３のそれぞれは、テストコントローラ６０が有する制御レジスタに対して、ＤＤＲメモリ２０においてＣＰＵ１００～１０３用のテストパタンが格納されているアドレスを設定する。テストコントローラ６０は、ＤＤＲメモリ２０において、制御レジスタに設定されたアドレスから、ＣＰＵ１００～１０３用のテストパタンを取得する。

【０２２５】

また、共通回路１２０のランタイムテストを実行する場合、ＣＰＵ１００は、テストコントローラ６０が有する制御レジスタに対して、ＤＤＲメモリ２０において共通回路１２０用のテストパタンが格納されているアドレスを設定する。テストコントローラ６０は、ＤＤＲメモリ２０において、制御レジスタに設定されたアドレスから、共通回路１２０用のテストパタンを取得する。

【０２２６】

なお、以上の説明では、実施の形態６として、実施の形態３に対して、ＢＩＳＴコントローラ１１、１１０～１１３に代えて、テストコントローラ６０、１６０～１６３を有するようにし、ＤＤＲメモリ２０にテストパタン及び期待値を格納した形態について説明したが、これに限られない。実施の形態１、２、４、５のいずれかに対して、これらの構成を適用するようにしてもよい。

【０２２７】

ここで、実施の形態４に対して、これらの構成を適用した場合には、ＣＰＵ１００～１０３のそれぞれは、テストコントローラ６０が有する制御レジスタに対して、テストパタン数に代えて、その分割テストでスキャンインするテストパタン量（例えば、ＤＤＲメモリ２０における開始アドレスと終了アドレス）を設定するようにしてもよい。すなわち、ＤＤＲメモリ２０に格納されたＣＰＵ１００～１０３のそれぞれに対するテストパタン群を分割した複数の単位のそれぞれについてのランタイムテストを、ＣＰＵ１００～１０３のそれぞれに対する分割テストとして実施する。また、ＤＤＲメモリ２０に格納された共通回路１２０に対するテストパタン群を分割した複数の単位のそれぞれについてのランタイムテストを、共通回路１２０に対する分割テストとして実施する。

【０２２８】

以上に説明したように、本実施の形態６では、スキャンテスト回路（テストコントローラ６０、１６０～１６３に対応）は、半導体装置６の外部に備えられた外部記憶回路（ＤＤＲメモリ２０に対応）に格納されたテストパタンを取得し、取得したテストパタンを処理回路にスキャンインすることでスキャンテストを実行するようにしている。

【０２２９】

ＢＩＳＴはランダムに生成したテストパタンを利用するため、故障検出率が上昇し難いという問題がある。それに対して、本実施の形態６では、故障検出率の高いテストパタンを事前に作成し、テスト対象の処理回路に流し込むことができるようになり、テスト時間を短縮することができる。

【０２３０】

＜実施の形態７＞

続いて、実施の形態７について説明する。以下の実施の形態７の説明では、上述した実施の形態１～６と同様の内容については、同一の符号を付す等して、適宜、その説明を省略する。

【０２３１】

図２３に示すように、本実施の形態７に係る半導体装置７は、実施の形態３に係る半導体装置３と比較して、ＣＰＵクラスタ１０がＣＰＵ１００を１つだけ有する点が異なる。また、本実施の形態７に係るＣＰＵ１００は、実施の形態３に係るＣＰＵ１００と比較し

て、さらに、初期化マスク回路１１７ａ、１１７ｂを有する点が異なる。また、本実施の形態７に係る共通回路１２０は、実施の形態３に係る共通回路１２０と比較して、スヌープ制御ユニット１２５及びＳＣＵタグＲＡＭ１２６を有さない点が異なる。これは、ＣＰＵクラスタ１０は、複数のＣＰＵを有さないため、ＣＰＵ間でＬ１キャッシュメモリ１１５のキャッシュコヒーレンシを保証する制御が不要であるからである。

【０２３２】

初期化マスク回路１１７ａは、Ｌ１キャッシュコントローラ１１４がＬ１キャッシュメモリ１１５に対して初期値を書き込んでＬ１キャッシュメモリ１１５を初期化する処理を抑止する回路である。より具体的には、初期化マスク回路１１７ａは、初期化処理の抑止を指示する指示信号をＬ１キャッシュコントローラ１１４に出力する。Ｌ１キャッシュコントローラ１１４は、ＣＰＵ１００のリセット時に、その指示信号が入力されている場合、ＣＰＵ１００のリセット解除で実行するＬ１キャッシュメモリ１１５に対するデータの初期化のための書き込みを行わないように、その動作を変更する。初期化マスク回路１１７ａは、ＣＰＵ１００のランタイムテスト実行後のリセット時における、Ｌ１キャッシュメモリ１１５の初期化を抑止するよう、その動作を有効化する。

【０２３３】

初期化マスク回路１１７ｂは、Ｌ１キャッシュコントローラ１１４によるＬ１キャッシュメモリ１１５に対する書き込みを抑止する回路である。より具体的には、初期化マスク回路１１７ｂは、Ｌ１キャッシュコントローラ１１４からＬ１キャッシュメモリ１１５に対してデータの書き込みのために出力される信号をマスク（遮断）する。初期化マスク回路１１７ｂは、ＣＰＵ１００のランタイムテスト実行時におけるＬ１キャッシュメモリ１１５へのデータの書き込みを抑止するよう、その動作を有効化する。

【０２３４】

図５を参照して説明すると、ステップＳ４のタイミングで、ＣＰＵ１００は、Ｌ１キャッシュメモリ１１５に対する初期化及び書き込みのそれぞれを抑止するように、初期化マスク回路１１７ａ、１１７ｂのそれぞれを設定する。また、ステップＳ１４のタイミングで、ＣＰＵ１００は、Ｌ１キャッシュメモリ１１５に対する初期化及び書き込みのそれぞれの抑止を解除するように、初期化マスク回路１１７ａ、１１７ｂのそれぞれもしくはクロックコントローラ１２を設定する。この設定は、初期化マスク回路１１７ａ、１１７ｂのそれぞれが有する制御レジスタ（図示せず）に対して、初期化又は書き込みを抑止するか否かを示す値を設定することで行われる。

【０２３５】

これによれば、Ｌ１キャッシュメモリ１１５の内容が保持されるため、ＣＰＵ１００が処理を再開した際におけるＬ１キャッシュメモリ１１５のミスヒットを防止して、性能劣化を抑止することができる。このように、マルチコアとは異なり、シングルコアの場合には、Ｌ１キャッシュメモリ１１５の内容をフラッシュせずに保持したままであっても、他のＣＰＵがスヌープ制御ユニット１２５の制御によってそのデータにアクセスすることができなくなってしまうといった問題も生じることはない。

【０２３６】

なお、以上の説明では、実施の形態７として、実施の形態３に対して、ＣＰＵ１００を１つだけ有するようにし、ＣＰＵ１００がさらに初期化マスク回路１１７ａ、１１７ｂを有するようにし、共通回路１２０がスヌープ制御ユニット１２５及びＳＣＵタグＲＡＭ１２６を有さないようにした形態について説明したが、これに限られない。実施の形態１、２、４～６のいずれかに対して、これらの構成を適用するようにしてもよい。

【０２３７】

以上に説明したように、本実施の形態７では、ＣＰＵ１００がＬ１キャッシュメモリ１１５と、Ｌ１キャッシュコントローラ１１４を有する。そして、ＣＰＵ１００に対するスキャンテストを実行しているときに、Ｌ１キャッシュコントローラ１１４からＬ１キャッシュメモリ１１５に対するデータの書き込みを抑止するようにしている。これによれば、ＣＰＵ１００が処理を再開した際におけるＬ１キャッシュメモリ１１５のミスヒットを防

止して、性能劣化を抑止することができる。
【０２３８】
＜実施の形態８＞
続いて、実施の形態８について説明する。以下の実施の形態８の説明では、上述した実施の形態１～７と同様の内容については、同一の符号を付す等して、適宜、その説明を省略する。図２４に示すように、本実施の形態８に係る半導体装置８は、実施の形態２に係る半導体装置２と比較して、さらに、タイマ２７と、ＧＰＵ（Graphics Processing Unit）クラスタ４０を有する。ＧＰＵクラスタ４０は、ＧＰＵ４００と、ＢＩＳＴコントローラ４１０とを有する。
【０２３９】
タイマ２７は、時間の経過を計測し、ＧＰＵクラスタ４０のランタイムテストの実行タイミングを割り込みによってＣＰＵクラスタ１０に対して通知する回路である。より具体的には、タイマ２７は、ランタイムテストの実行タイミングとなる毎に、割り込み信号を割り込みコントローラ１４に出力する。割り込みコントローラ１４は、この割り込み信号に応じて、タイマ２７からの割り込みの通知先として予め設定されたＣＰＵに対して割り込み信号を出力する。
【０２４０】
なお、ＣＰＵクラスタ１０は、半導体装置８の起動後に、ランタイムテストの実行タイミングをＤＴＩ毎に通知するようにタイマ２７を設定する。タイマ２７を設定するＣＰＵは、例えば、予め任意に定めるようにしてもよい。
【０２４１】
ＧＰＵ４００は、ＣＰＵクラスタ１０のＣＰＵ１００～１０３が実行する処理の一部を補助的に実行する回路である。すなわち、ＧＰＵ４００は、ＣＰＵ１００～１０３のいずれかによって管理され、ＣＰＵ１００～１０３のいずれかからの制御がないと動作しない回路である。
【０２４２】
ＢＩＳＴコントローラ４１０は、ＢＩＳＴコントローラ１１からの制御に応じて、ＧＰＵ４００のランタイムテストを実行する回路である。ＢＩＳＴコントローラ４１０は、ＢＩＳＴコントローラ１１０～１１３と同様に、スレーブとして機能する。
【０２４３】
続いて、図２５を参照して、実施の形態８に係るＧＰＵ４００のランタイムテスト実行時の動作について説明する。以下、本実施の形態８では、ＣＰＵ１００がＧＰＵ４００のランタイムテストの実行を制御する手順について説明するが、ＣＰＵ１００に代えてＣＰＵ１０１～１０３のいずれかがＧＰＵ４００のランタイムテストの実行を制御する形態としてもよい。
【０２４４】
タイマ２７は、前回に割り込み信号を出力した時点から、ＤＴＩが経過した時点で、割り込みコントローラ１４を介してＣＰＵ１００に対して割り込みを通知する。すなわち、この例では、割り込みコントローラ１４には、タイマ２７からの割り込みの通知先として、ＣＰＵ１００が予め設定されている。
【０２４５】
ＣＰＵ１００は、その処理の実行中に（Ｓ１５０）、タイマ２７に起因する割り込みが通知されると、タイマ２７の割り込みに応じた割り込みハンドラを実行する。ＣＰＵ１００は、割り込みハンドラにおいて、ランタイムテストを実行するための設定を行う（Ｓ１５１）。より具体的には、ＣＰＵ１００は、ＢＩＳＴコントローラ１１の制御レジスタに対してＧＰＵ４００をテスト対象として設定する。また、ＣＰＵ１００は、割り込みハンドラにおいて、ＧＰＵ４００の情報を退避する（Ｓ１５２）。より具体的には、ＣＰＵ１００は、ＧＰＵ４００の動作を停止させて、ＧＰＵ４００のレジスタに格納されたデータをＤＤＲメモリ２０、メモリ２１、及び、内蔵メモリ５０に退避する。そして、ＣＰＵ１００は、ＢＩＳＴコントローラ１１に対してランタイムテストの開始を指示し（Ｓ１５３

）、割り込みハンドラを終了する（Ｓ１５４）。ＧＰＵ４００は、スリープ状態に移行することができないため、このようにＣＰＵ１００から明示的にランタイムテストの開始を指示する。

【０２４６】

ＢＩＳＴコントローラ１１は、ＣＰＵ１００からの指示に応じて、ＧＰＵ４００のランタイムテストの開始をＢＩＳＴコントローラ４１０に指示する。ＢＩＳＴコントローラ４１０は、ＢＩＳＴコントローラ１１からの指示に応じて、ＧＰＵ４００のランタイムテストを実行する（Ｓ１５５）。

【０２４７】

ＢＩＳＴコントローラ４１０は、ＧＰＵ４００のランタイムテストの実行終了をＢＩＳＴコントローラ１１に通知する。ＢＩＳＴコントローラ１１は、ＢＩＳＴコントローラ４１０からの通知に応じて、リセットコントローラ１３を介してＧＰＵ４００をリセットする（Ｓ１５６）。また、ＢＩＳＴコントローラ１１は、割り込みコントローラ１４を介してＣＰＵ１００に対して割り込みを通知する。ＣＰＵ１００は、その処理の実行中に（Ｓ１５７）、ＢＩＳＴコントローラ１１に起因する割り込みが通知されると、ＢＩＳＴコントローラ１１からの割り込みに応じた割り込みハンドラを実行する。ＣＰＵ１００は、割り込みハンドラにおいて、ランタイムテスト結果の確認（Ｓ１５８）と、ＧＰＵ４００の情報の復帰（Ｓ１５９）を行う。より具体的には、ＣＰＵ１００は、ＤＤＲメモリ２０、メモリ２１、及び、内蔵メモリ５０に退避したデータをＧＰＵ４００のレジスタに復帰する。なお、ランタイムテスト結果の確認は、ＣＰＵ１００～１０３及び共通回路１２０のランタイムテスト実行時と同様であるため、説明を省略する。そして、ＣＰＵ１００は、割り込みハンドラを終了し（Ｓ１６０）、プログラムの実行を再開する（Ｓ１６１）。

【０２４８】

なお、以上の説明では、実施の形態８として、実施の形態２に対して、さらに、タイマ２７と、ＧＰＵクラスタ４０（ＧＰＵ４００及びＢＩＳＴコントローラ４１０）を有するようにした形態について説明したが、これに限られない。実施の形態１、３～７のいずれかに対して、これらの構成を適用するようにしてもよい。ここで、ＧＰＵクラスタ４０は、暗号処理回路や画像認識回路等のハードウェアアクセラレータであってもよい。

【０２４９】

以上に説明したように、本実施の形態８では、半導体装置８は、ＧＰＵ４００に対するスキャンテストを実行するタイミングをＣＰＵ１００に通知するタイマ２７を備えている。ＣＰＵ１００は、タイマ２７からの通知に応じて、ＧＰＵ４００に対するスキャンテストの実行を指示する。そして、スキャンテスト回路（ＢＩＳＴコントローラ１１、４１０に対応）は、ＣＰＵ１００からの指示に応じて、ＧＰＵ４００に対するスキャンテストを実行する。ＧＰＵ４００のような大規模なモジュールは、面積制限の都合上、二重化してロックステップによって動作を確認することは困難である。それに対して、本実施の形態８によれば、ＣＰＵ１００からの制御によるランタイムテストによって、大規模なＧＰＵ４００であっても定期的な正常動作を確認することが可能である。

【０２５０】

＜実施の形態９＞

続いて、実施の形態９について説明する。以下の実施の形態９の説明では、上述した実施の形態１～８と同様の内容については、同一の符号を付す等して、適宜、その説明を省略する。図２６に示すように、本実施の形態９に係る半導体装置９は、実施の形態３に係る半導体装置３と比較して、さらに、タイマ２８～３１を有する。

【０２５１】

上述したタイマ１５は、時間の経過を計測し、ランタイムテストの実行タイミングを割り込みによってＣＰＵ１０３に対して通知する回路であった。それに対して、タイマ２８～３１のそれぞれは、タイマ１５と同様に、時間の経過を計測し、ランタイムテストの実行タイミングを割り込みによってＣＰＵ１０２～１００及び共通回路１２０のそれぞれに対して通知する回路である。なお、ＣＰＵクラスタ１０は、半導体装置９の起動後に、タ

イマ１５と同様に、ランタイムテストの実行タイミングをＤＴＩ毎に通知するようにタイマ２８～３１を設定する。タイマ１５、２８～３１を設定するＣＰＵは、例えば、予め任意に定めるようにしてもよい。

【０２５２】

すなわち、割り込みコントローラ１４は、タイマ２８からの割り込みに応じた割り込み信号の通知先としてＣＰＵ１０２が予め設定されており、タイマ２９からの割り込みに応じた割り込み信号の通知先としてＣＰＵ１０１が予め設定されており、タイマ３０からの割り込みに応じた割り込み信号の通知先としてＣＰＵ１００が予め設定されており、タイマ３１からの割り込みに応じた割り込み信号の通知先としてＣＰＵ１００が予め設定されている。

【０２５３】

ＣＰＵ１０３は、これまでの実施の形態１～８と同様に、タイマ１５からの割り込みに応じて、ＣＰＵ１０３のランタイムテストを実行する制御を行う。ＣＰＵ１０２は、ＣＰＵ１０３からのＣＰＵ間割り込みでなく、タイマ２８からの割り込みに応じて、ＣＰＵ１０２のランタイムテストを実行する制御を行う点が実施の形態１～８と異なる。ＣＰＵ１０１は、ＣＰＵ１０２からのＣＰＵ間割り込みでなく、タイマ２９からの割り込みに応じて、ＣＰＵ１０１のランタイムテストを実行する制御を行う点が実施の形態１～８と異なる。ＣＰＵ１００は、ＣＰＵ１０１からのＣＰＵ間割り込みでなく、タイマ３０からの割り込みに応じて、ＣＰＵ１００のランタイムテストを実行する制御を行う点が実施の形態１～８と異なる。また、ＣＰＵ１００は、ＣＰＵ１００のランタイムテストの実行後でなく、タイマ３１からの割り込みに応じて、共通回路１２０のランタイムテストを実行する制御を行う点が実施の形態１～８と異なる。すなわち、本実施の形態９では、ＣＰＵ１０３～１０１のそれぞれは、ＣＰＵ間割り込みをＣＰＵ１０２～１００のそれぞれに通知しない。

【０２５４】

このように、ＣＰＵ１００～１０３及び共通回路１２０毎に、専用のタイマ１５、２８～３１を用意してランタイムテストを実行するようにしてもよい。

【０２５５】

＜他の実施の形態＞

（パワーオンセルフテストをランタイムテストで代用）

ＣＰＵ１００～１０３では、一般的に、電源ＯＮ直後にＬａｔｅｎｔ　Ｆａｕｌｔを検出するためにパワーオンセルフテスト（Power On Self Test：ＰＯＳＴ）が実施されているが、このＰＯＳＴに代えて、上述のランタイムテストを実施するようにしてもよい。

【０２５６】

ここで、パワーオン直後のセルフテスト時間も、起動時間の制約により、制限時間が課されることが一般的である。よって、大規模な回路をＢＩＳＴで診断する場合には、制限時間内に終わらない可能性がある。しかしながら、パワーオン直後から使用する必要のない回路は、パワーオン直後のＰＯＳＴを上述のランタイムテストによって代用することが可能である。

【０２５７】

よって、ＣＰＵ１００～１０３のうち、起動時間の制約は不要であるとして予め定められた少なくとも１つのＣＰＵについては、ＰＯＳＴを実行せず、上述のランタイムテストを実行するようにしてもよい。

【０２５８】

例えば、ＢＩＳＴコントローラ１１の制御レジスタに、ＣＰＵ１００～１０３のそれぞれについて、ＰＯＳＴを実施するＣＰＵであるか否かを示す設定値を予め格納するようにする。そして、ＢＩＳＴコントローラ１１は、制御レジスタの設定値に基づいて、ＰＯＳＴを実施しないＣＰＵに対しては、そのＣＰＵがブートした後にランタイムテストを実施するようにしてもよい。

【０２５９】

例えば、ＢＩＳＴコントローラ１１がＰＯＳＴを実施しないＣＰＵに対して、割り込みコントローラ１４を介して割り込みを通知し、ＣＰＵが割り込みに応じて、自身のランタイムテストを実行する制御を開始するようにしてもよい。なお、ＣＰＵがブートしたか否かは、例えば、ＣＰＵがブートしたときにブートしたことを示すステータス信号をＢＩＳＴコントローラ１１に送信することで、ＢＩＳＴコントローラ１１が認識可能とすればよい。以上、ＣＰＵ１００～１０３を例に説明したが、同様な考え方をＧＰＵ４００に適用できることは言うまでもない。

【０２６０】

（タイマ以外でのランタイムテストの開始・再開タイミングの通知）

また、上述した実施の形態では、タイマ１５、２７～３１でランタイムテストの開始タイミングを通知し、タイマ２５でランタイムテストの再開タイミングを通知するようにしているが、これに限られない。例えば、ＢＩＳＴコントローラ１１からランタイムテストの開始・再開タイミングを通知するようにしてもよい。例えば、ＢＩＳＴコントローラ１１にランタイムテストの開始又は再開タイミングを設定するレジスタを設ける。ＢＩＳＴコントローラ１１は、レジスタに設定された開始又は再開タイミングで、ＣＰＵ１００～１０３、共通回路１２０、及びＧＰＵ４００のそれぞれにランタイムテストの開始又は再開タイミングを通知する。

【０２６１】

なお、ＢＩＳＴコントローラ１１が開始又は再開タイミングを認識する方法は、タイマのようにクロック信号に基づいてカウントダウンを行うことで認識する方式を採用してもよく、半導体装置の外部に設けられたタイマ（Wall Clock）の出力値（時刻を示す値）を参照することで認識する方式を採用してもよい。

【０２６２】

（システムバス及びバスコントローラの二重化）

ランタイムテストのテスト対象は、上記の例のみに限られず、例えば、Ｌ３キャッシュメモリとそのキャッシュコントローラを有する回路、又は、システムバス、スヌープ制御システムユニット２３、及びバスコントローラ（ＤＤＲコントローラ１６及び外部バスコントローラ１７）を対象とすることもできる。ただし、システムバス、スヌープ制御システムユニット２３、及びバスコントローラをテスト対象とした場合には、システムバスにおけるトランザクションを阻害し、システム性能の劣化が発生してしまう。そのため、この場合には、図２７に示すように、システムバス、スヌープ制御システムユニット２３、及びバスコントローラを二重化し、ランタイムテストの実行中には、ランタイムテストを実行していないシステムバス、スヌープ制御システムユニット２３、及びバスコントローラを使用することで、システム性能の劣化を低減するようにしてもよい。

【０２６３】

なお、以上の説明では、システムバス、スヌープ制御システムユニット２３、及びバスコントローラの全てを二重化した例について説明したが、これに限られない。システムバス、スヌープ制御システムユニット２３、及びバスコントローラのうち、いずれか１つ又は２つを二重化するようにしてもよい。また、バスコントローラについても、ＤＤＲコントローラ１６及び外部バスコントローラ１７のうち、いずれか１つを二重化するようにしてもよい。また、いずれかの回路を二重化する場合に、二重化した回路を両方とも常時動作させ、それぞれの対応する出力信号の一致を常時確認する比較回路を別に設けるのであれば、その回路をランタイムテストのテスト対象にする必要は必ずしもない。

【０２６４】

以上、本発明者によってなされた発明を実施の形態に基づき具体的に説明したが、本発明は既に述べた実施の形態に限定されるものではなく、その要旨を逸脱しない範囲において種々の変更が可能であることは言うまでもない。

【０２６５】

上述した実施の形態では、ランタイムテストが実行される演算要素単位（演算回路）がＣＰＵである例について説明したが、これに限られない。例えば、マルチスレッドプロセ

ッサ又はメニーコアプロセッサにおけるコアを、ランタイムテストが実行される演算要素単位（演算回路）としてもよい。

【０２６６】

リセットマスク回路１８及び割り込みマスク回路１９の実装場所も、上述の実施の形態の例に限られない。例えば、リセットマスク回路１８は、リセットコントローラ１３内にあってもよく、ＣＰＵクラスタ１０内にあってもよい。また、例えば、割り込みマスク回路１９は、クロックコントローラ１２内にあってもよく、割り込みコントローラ１４内、又はＣＰＵクラスタ１０内にあってもよい。

【０２６７】

上述した実施の形態では、ランタイムテスト前にテスト対象回路のレジスタのデータを退避し、ランタイムテスト後に復帰するようにしているが、これに限られない。例えば、レジスタを二重化して、それらの値を比較することでレジスタの異常を検出可能とするとともに、レジスタをスキャンテスト対象外とすることで、ランタイムテストの前後でレジスタのデータを保持可能としてもよい。また、レジスタを二重化し、片方のレジスタをスキャンテスト対象外の退避用のレジスタとしてもよい。これは、メモリ２１にレジスタのデータを退避すると時間を要してしまう場合に有効である。

【０２６８】

上述した実施の形態では、ＣＰＵ１００～１０３のそれぞれは、ＢＩＳＴコントローラ１１の制御レジスタを参照することで、リセット要因がランタイムテスト実行後のリセットによりブートしたか否かを確認し、リセットハンドラにおける動作を変更していたが、これに限られない。例えば、ＣＰＵ１００～１０３のそれぞれは、ランタイムテストの実行前に、リセットベクタを、ランタイムテスト専用のリセットベクタに書き換えることによって、上述のランタイムテスト実行結果と期待値との比較などの処理を実行するランタイムテスト用のリセットハンドラが実行されるようにしてもよい。そして、このランタイムテスト用のリセットハンドラの最後の処理でリセットベクタを元に戻す処理を実行することで、通常のリセット後に通常のリセットハンドラが実行されるようにすればよい。

【０２６９】

上述した実施の形態では、共通回路１２０のリセット時におけるＬ２キャッシュメモリ１２４及びＳＣＵタグＲＡＭ１２６の初期化を、初期化マスク回路１２２ａによって実施するようにしているが、これに限られない。例えば、共通回路１２０に初期化マスク回路１２２ａを有さないようにし、共通回路１２０のリセット時も初期化マスク回路１２２ｂ、１２２ｃによるマスクを有効にすることで、Ｌ２キャッシュメモリ１２４及びＳＣＵタグＲＡＭ１２６の初期化を抑止するようにしてもよい。

【０２７０】

上述した実施の形態では、ＣＰＵ１００～１０３のそれぞれのランタイムテストを実行する前に、ＣＰＵ１００～１０３のそれぞれがスリープ状態に移行するようにしているが、ＣＰＵ１００～１０３のそれぞれが通常動作を停止した状態に移行するのであれば、これに限られない。例えば、ＣＰＵ１００～１０３のそれぞれは、無限ループ処理を実行した状態に移行するようにしてもよい。

【０２７１】

ＣＰＵ１００～１０３のいずれかが、上述のランタイムテストを実行するためのスリープ状態でなく、通常の処理に応じてスリープ状態となっている場合には、そのＣＰＵに対するランタイムテストをスキップするようにしてもよい。例えば、ＣＰＵ１０３～１０１のそれぞれは、次のランタイムテスト対象のＣＰＵがスリープ状態である場合には、そのＣＰＵをスキップして、さらにその次のランタイムテスト対象のＣＰＵにＣＰＵ間割り込みを通知するようにしてもよい。また、ＣＰＵ１０３がスリープ状態に移行する場合には、タイマ１５からの割り込みの通知先を、その次のランタイムテスト対象のＣＰＵ１０２に変更するように割り込みコントローラ１４を設定するようにしてもよい。

【０２７２】

また、上述した実施の形態では、ランタイムテスト対象の回路のレジスタのデータを、

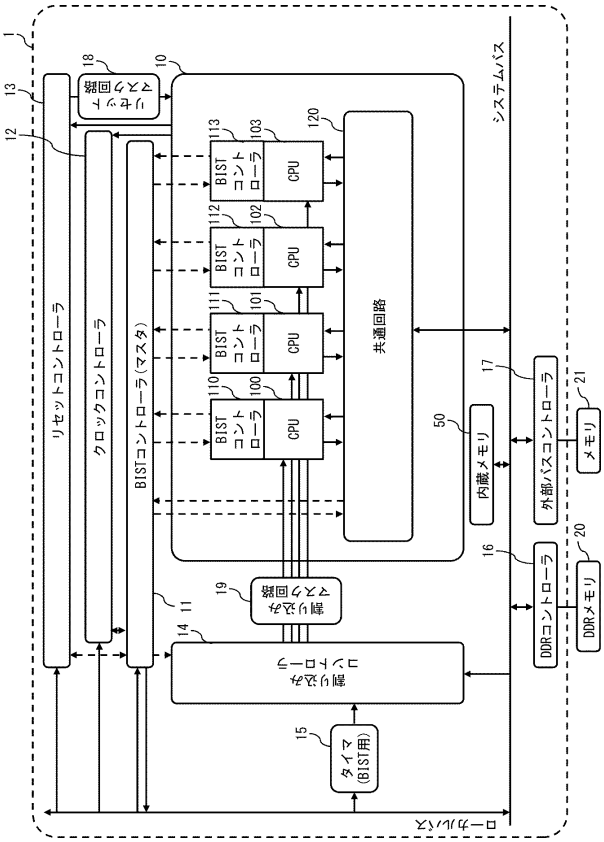
ＤＤＲメモリ２０、メモリ２１、及び、内蔵メモリ５０に退避するものとして説明したが、これらの全てを使用しなくてもよい。例えば、ＤＤＲメモリ２０、メモリ２１、及び、内蔵メモリ５０のうち、いずれか１つ又は２つにレジスタのデータを退避するようにしてもよい。

【符号の説明】

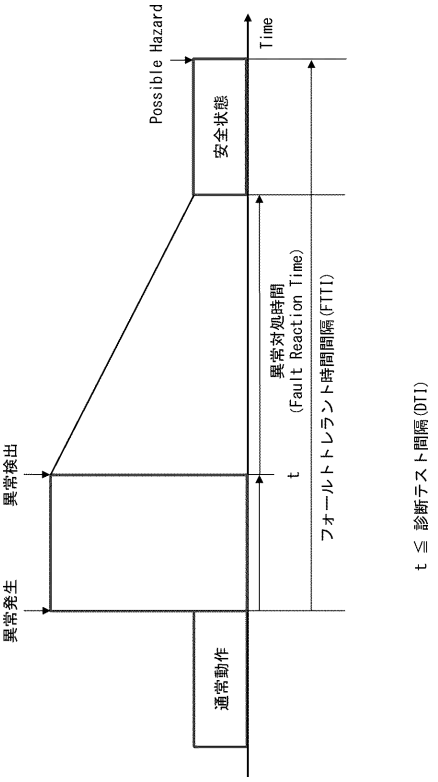
【０２７３】

１、２、３、４、５、６、７、８、９、９０　半導体装置
１０、７０　ＣＰＵクラスタ
１１　ＢＩＳＴコントローラ（マスタ）
１２　クロックコントローラ
１３　リセットコントローラ
１４　割り込みコントローラ
１５、２８、２９、３０　タイマ（ＣＰＵ　ＢＩＳＴ用）
１６　ＤＤＲコントローラ
１７　外部バスコントローラ
１８　リセットマスク回路
１９　割り込みマスク回路
２０　ＤＤＲメモリ
２１　メモリ
２２　タイマ（タイムアウト用）
２３　スヌープ制御システムユニット
２５　タイマ（インターバル用）
２６　電源制御回路
２７　タイマ（ＧＰＵ　ＢＩＳＴ用）
４０　ＧＰＵクラスタ
５０　内蔵メモリ
６０、１６０、１６１、１６２、１６３　テストコントローラ
９１　記憶回路
９２　処理回路
９３　抑止回路
９４　スキャンテスト回路
１００、１０１、１０２、１０３　ＣＰＵ
１１０、１１１、１１２、１１３、１２１　ＢＩＳＴコントローラ（スレーブ）
１１４　Ｌ１キャッシュコントローラ
１１５　Ｌ１キャッシュメモリ
１１６　分岐履歴メモリ
１２０　共通回路
１２２　初期化マスク回路
１２３　Ｌ２キャッシュコントローラ
１２４　Ｌ２キャッシュメモリ
１２５　スヌープ制御ユニット
１２６　ＳＣＵタグＲＡＭ
１２７　アクセス履歴メモリ
１３１、１３２、１３３、１３４、１４１、１４２、１４３、１４４、１４５、１４６、１４７、１４８、１４９、１５０　マスク回路
１７０、１７１、１７２、１７３、１８０　電源スイッチ
４００　ＧＰＵ
４１０　ＢＩＳＴコントローラ

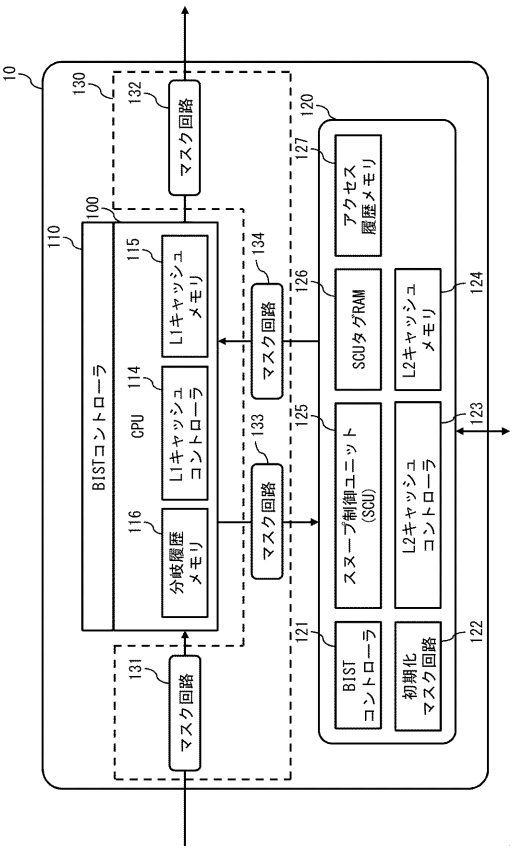
【図１】


リセットコントローラ
クロックコントローラ
BISTコントローラ(マスタ)
共通回路
システムバス
ローカルバス
BISTコントローラ
CPU
内蔵メモリ
外部バスコントローラ
メモリ
DDRコントローラ
DDRメモリ
異常検出
異常発生
タイマ(BIST用)

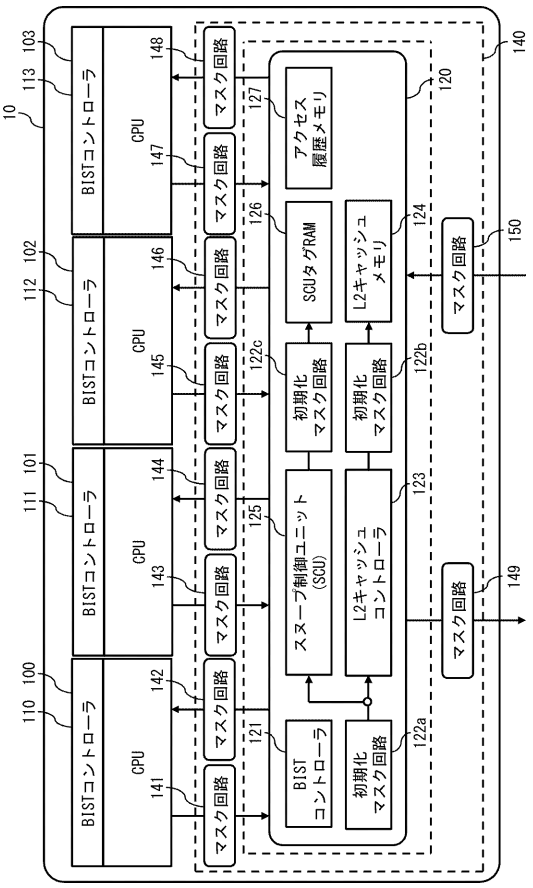
【図２】


Possible Hazard
Time
安全状態
通常動作
異常対処時間
(Fault Reaction Time)
t
フォールトトレラント時間間隔(FTTI)
t ≦ 診断テスト間隔(DTI)

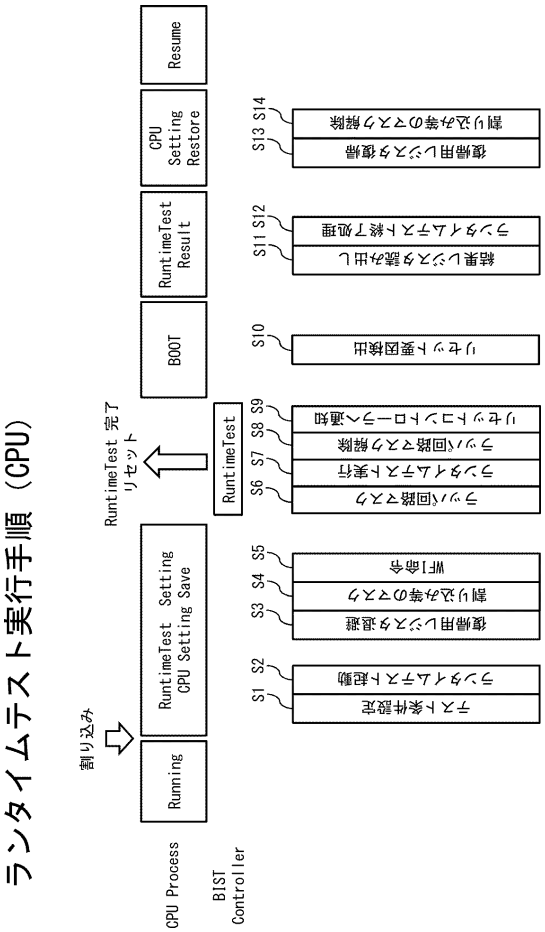
【図３】


BISTコントローラ
CPU
L1キャッシュコントローラ
L1キャッシュメモリ
分岐履歴メモリ
マスク回路
スヌープ制御ユニット(SCU)
SCUタグRAM
アクセス履歴メモリ
L2キャッシュコントローラ
L2キャッシュメモリ
BISTコントローラ
初期化マスク回路

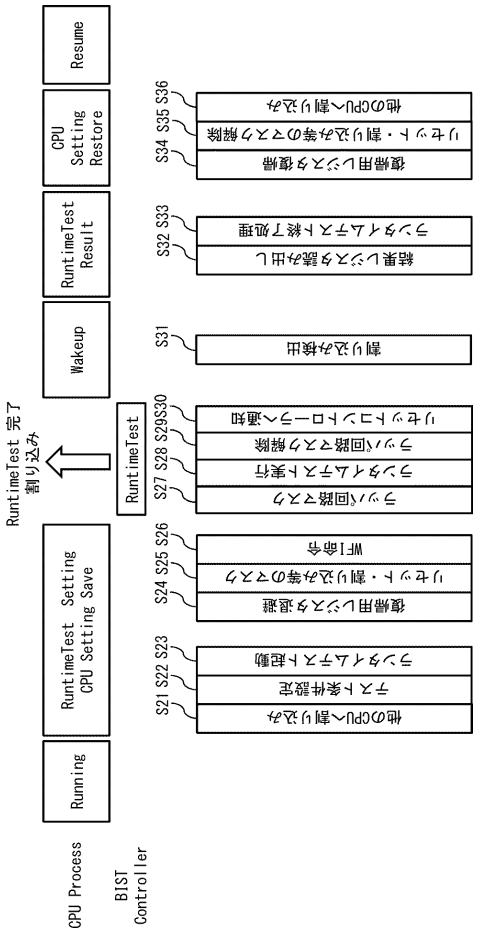
【図４】


BISTコントローラ
CPU
マスク回路
スヌープ制御ユニット(SCU)
SCUタグRAM
アクセス履歴メモリ
L2キャッシュコントローラ
L2キャッシュメモリ
BISTコントローラ
初期化マスク回路

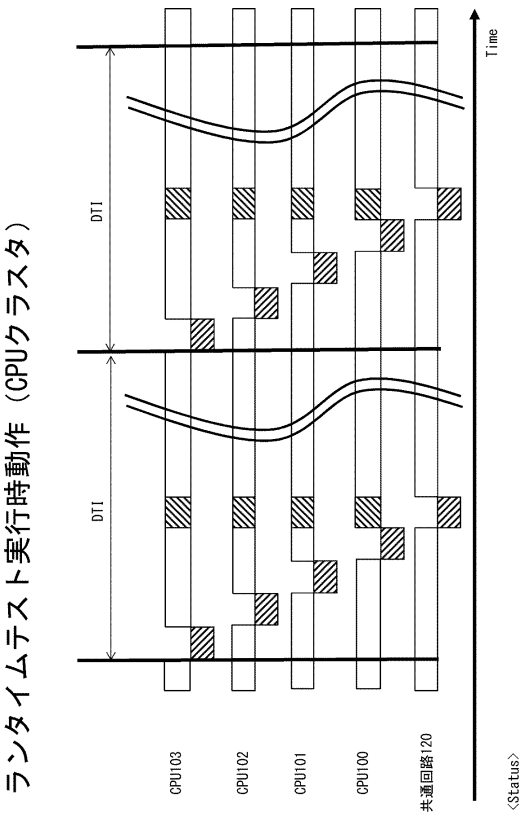
【図５】


ランタイムテスト実行手順（CPU）
CPU Process
BIST Controller
Running
割り込み
RuntimeTest Setting CPU Setting Save
RuntimeTest 完了 リセット
RuntimeTest
BOOT
RuntimeTest Result
CPU Setting Restore
Resume
S1 S2 テスト条件設定
テスト起動
S3 S4 S5 復帰用レジスタ退避
割り込み等のマスク
WFI命令
S6 S7 S8 S9 ウェイク回路マスク
ランタイムテスト実行
ウェイク回路マスク解除
リセットコントローラへ通知
S10 リセットベクタ要因検出
S11 S12 結果レジスタ読み出し
ランタイムテスト終了処理
S13 S14 復帰用レジスタ復帰
割り込み等のマスク解除

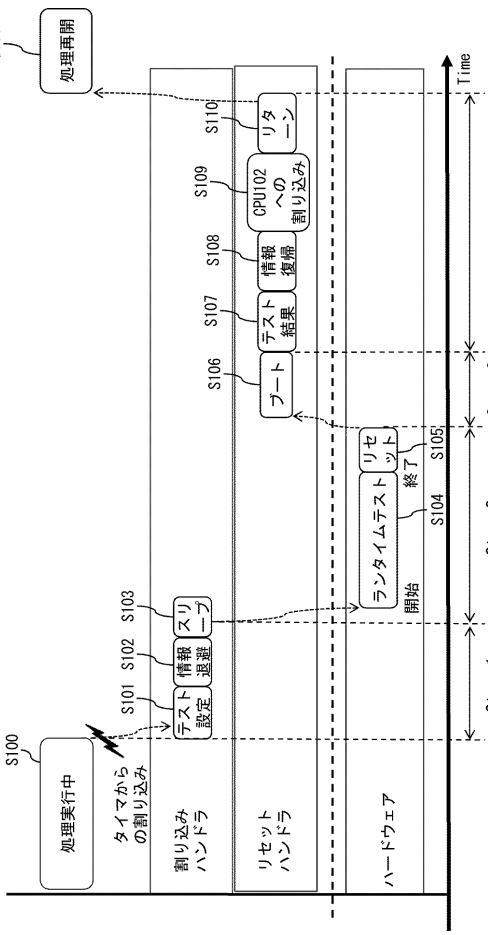
【図６】


ランタイムテスト実行手順（共通回路）
CPU Process
BIST Controller
Running
RuntimeTest Setting CPU Setting Save
RuntimeTest 完了 割り込み
RuntimeTest
Wakeup
RuntimeTest Result
CPU Setting Restore
Resume
S21 S22 S23 他のCPUへ割り込み
テスト条件設定
ランタイムテスト起動
S24 S25 S26 復帰用レジスタ退避
リセット・割り込み等のマスク
WFI命令
S27 S28 S29 S30 ウェイク回路マスク
ランタイムテスト実行
ウェイク回路マスク解除
リセットコントローラへ通知
S31 割り込み検出
S32 S33 結果レジスタ読み出し
ランタイムテスト終了処理
S34 S35 S36 復帰用レジスタ復帰
リセット・割り込み等のマスク解除
他のCPUへ割り込み

【図７】

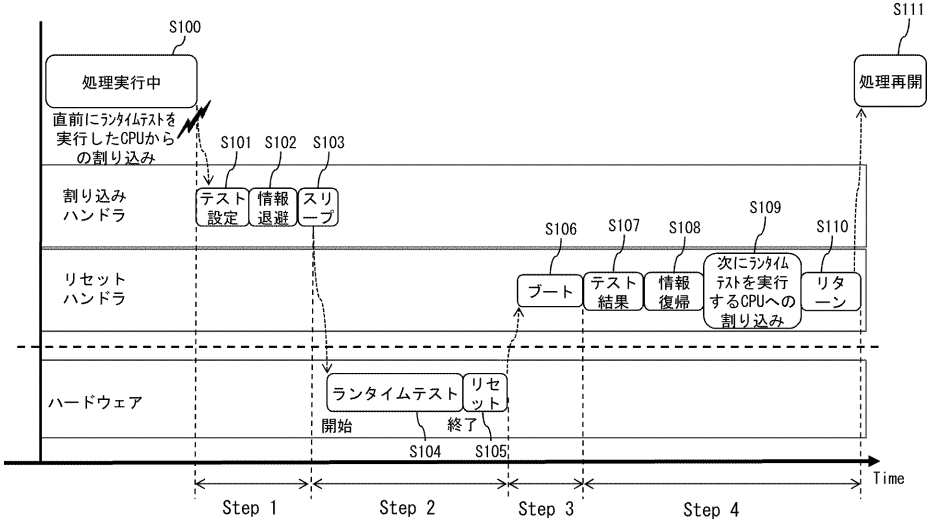

ランタイムテスト実行時動作（CPUクラスタ）
CPU103
CPU102
CPU101
CPU100
共通回路120
DTI
DTI
Time
<Status>
動作中
ランタイムテスト
CPUスリープ

【図８】


ランタイムテスト実行時動作（CPU103）
処理実行中
タイマからの割り込み
割り込みハンドラ
リセットハンドラ
ハードウェア
S100
S101 テスト設定
S102 情報退避
S103 スリープ
S104 ランタイムテスト開始
S105 終了
リセット
S106 ブート
S107 テスト結果
S108 情報復帰
S109 CPU102への割り込み
S110 リターン
S111 処理再開
Step 1
Step 2
Step 3
Step 4
Time

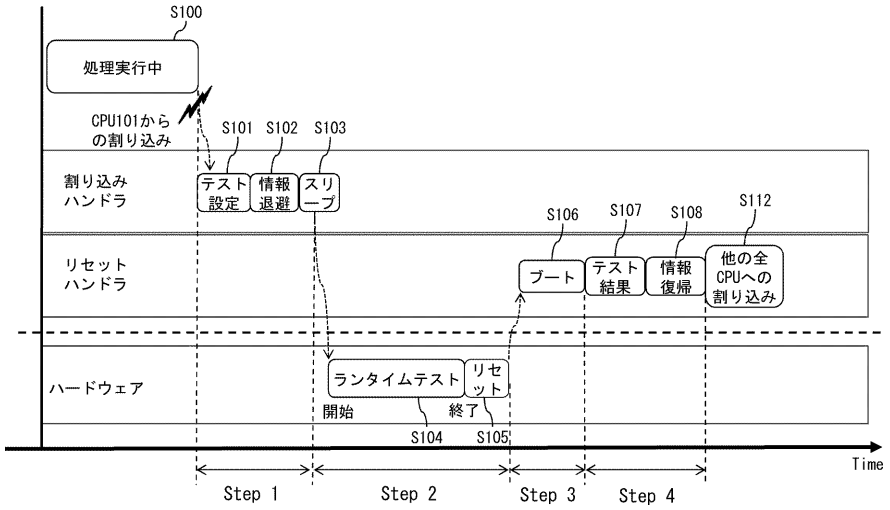
【図９】

ランタイムテスト実行時動作（CPU102、CPU101）



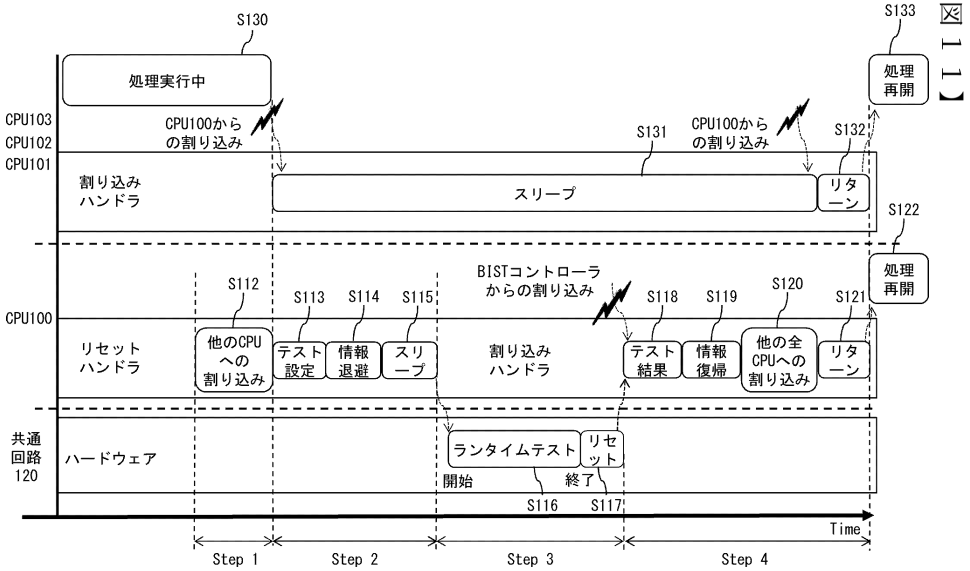
【図１０】

ランタイムテスト実行時動作（CPU100）

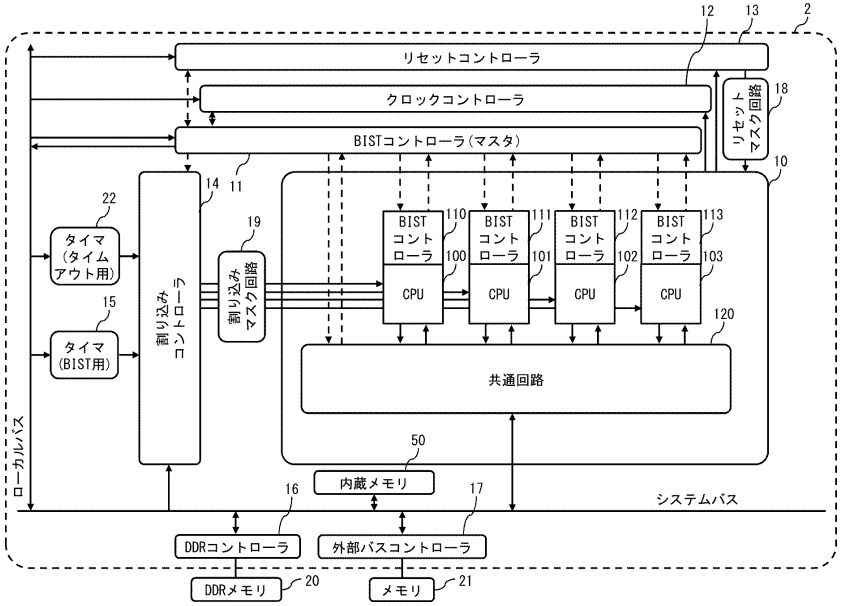


【図１１】

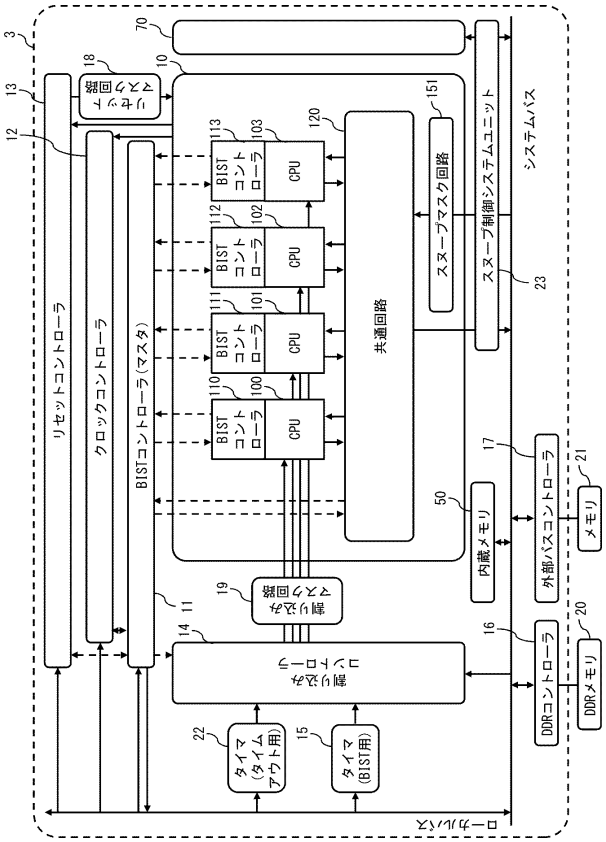
ランタイムテスト実行時動作（共通回路）



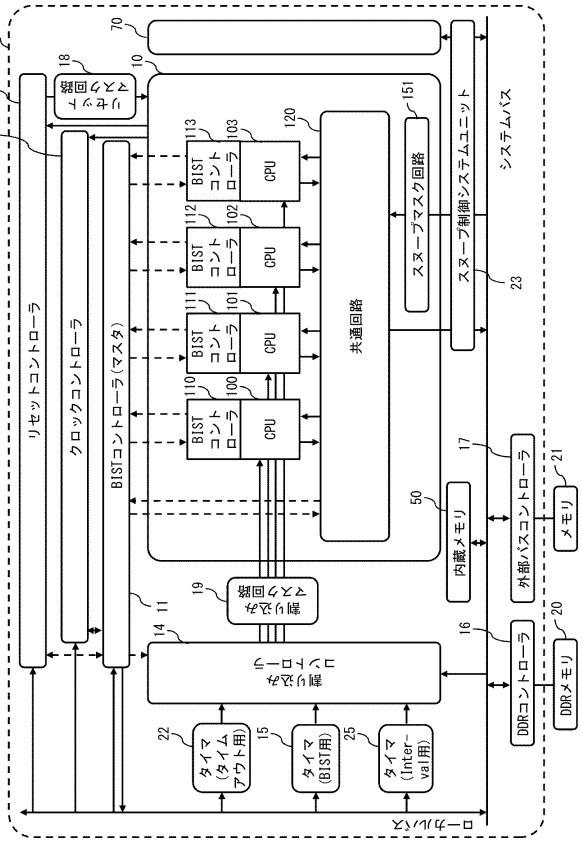
【図１２】



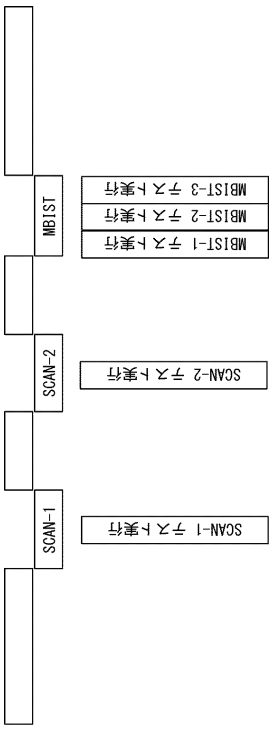
【図１３】



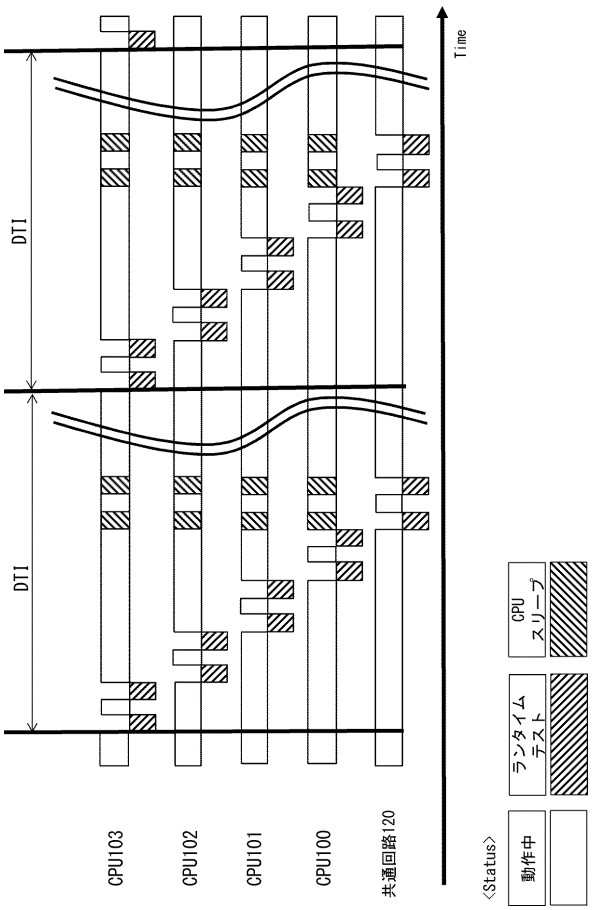
【図１４】



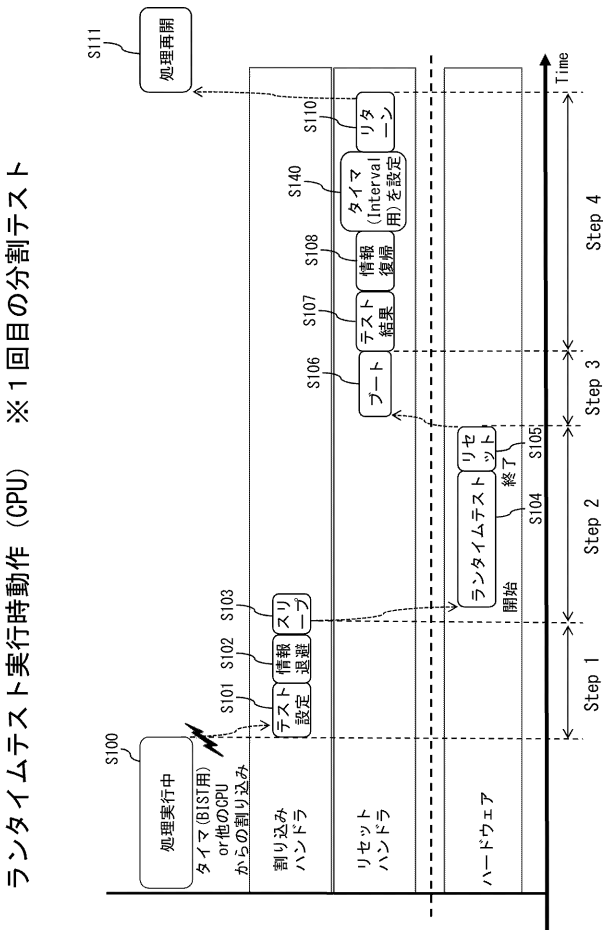
【図１５】



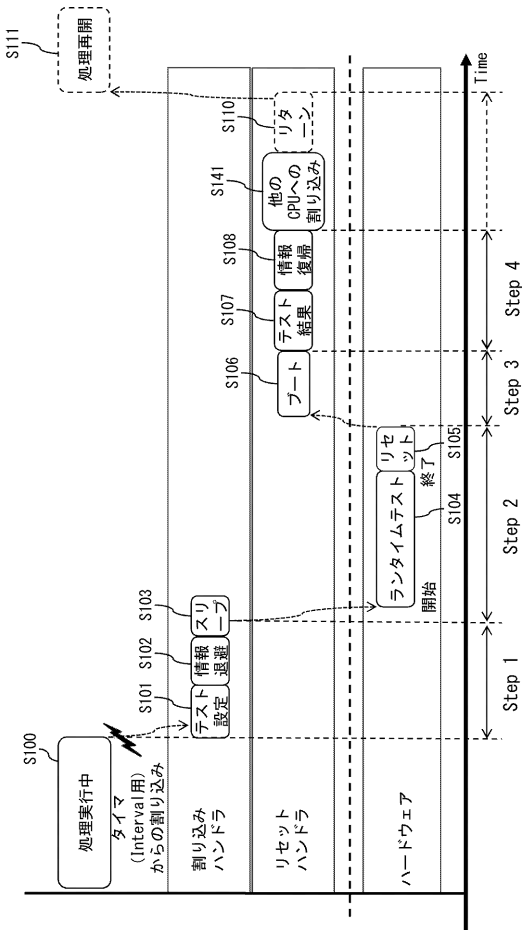
【図１６】



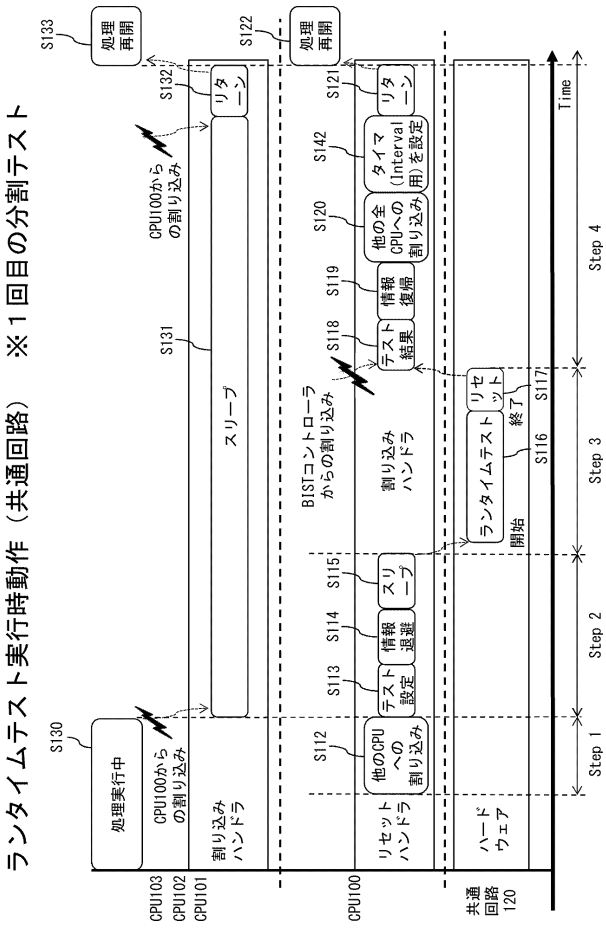
【図１７】



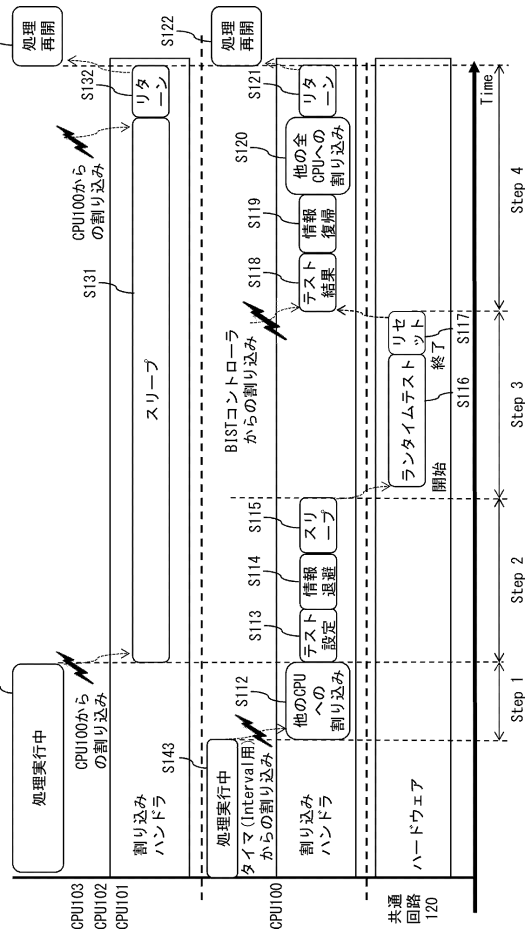
【図１８】



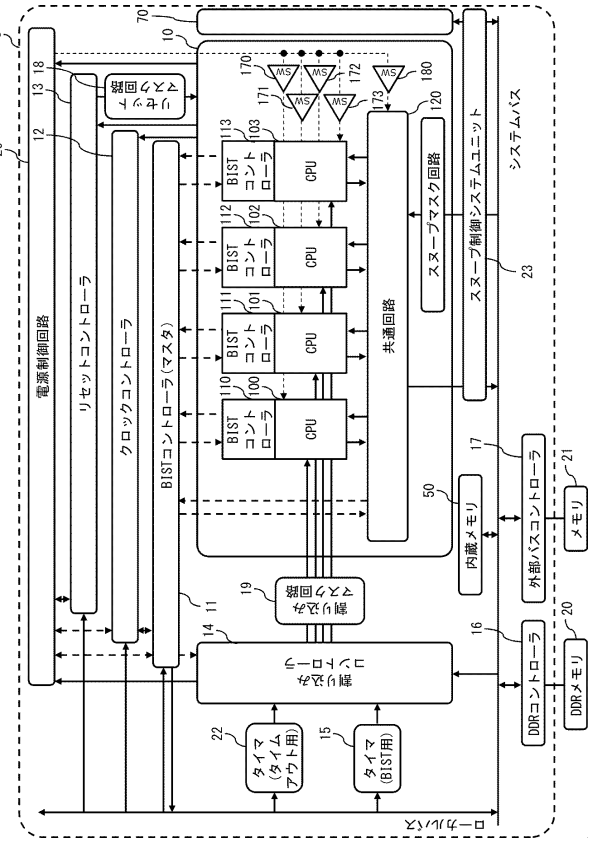
【図１９】



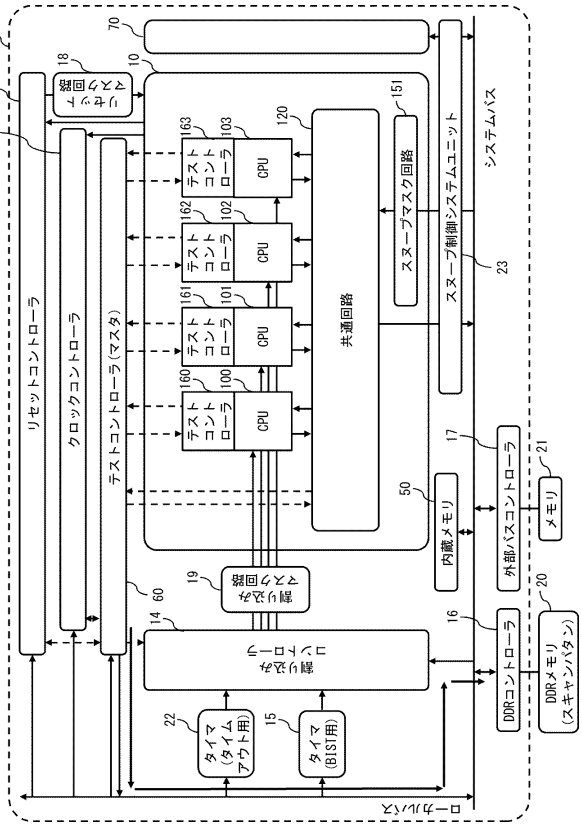
【図２０】



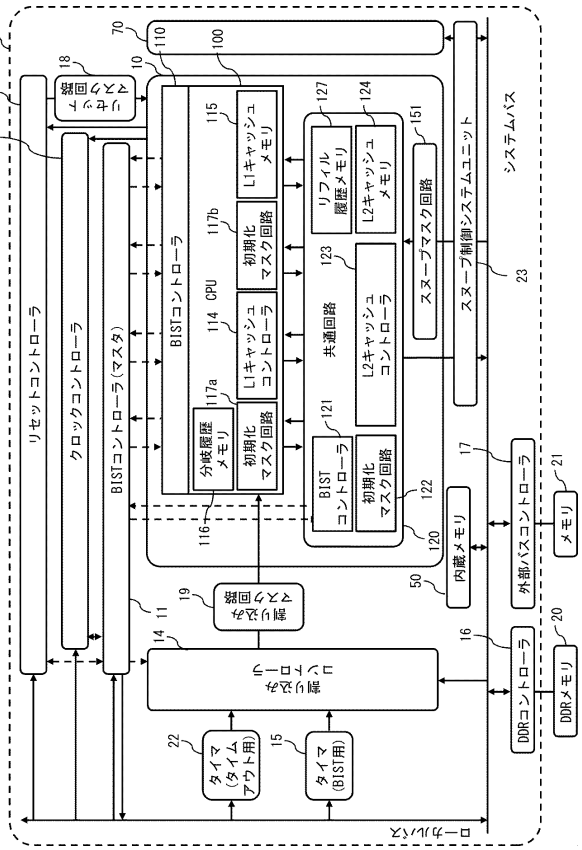
【図２１】



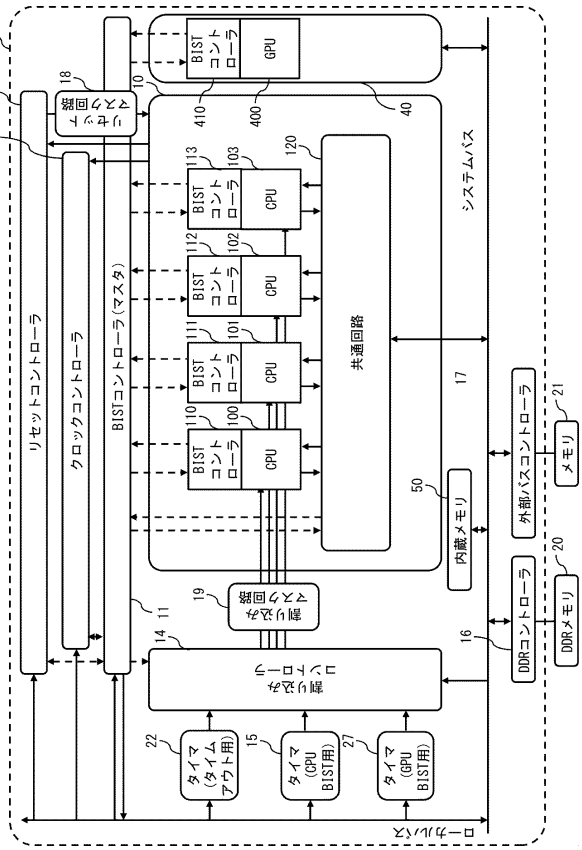
【図２２】



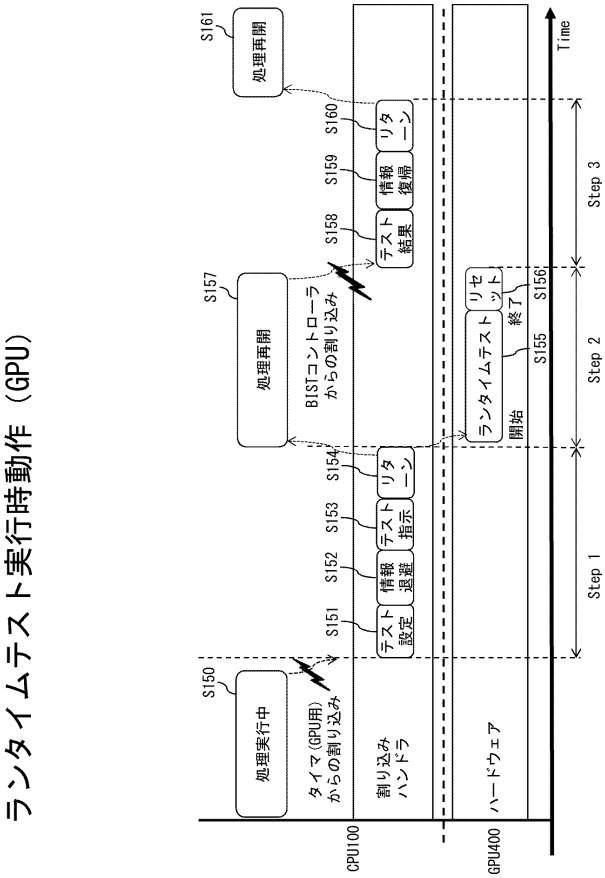
【図２３】



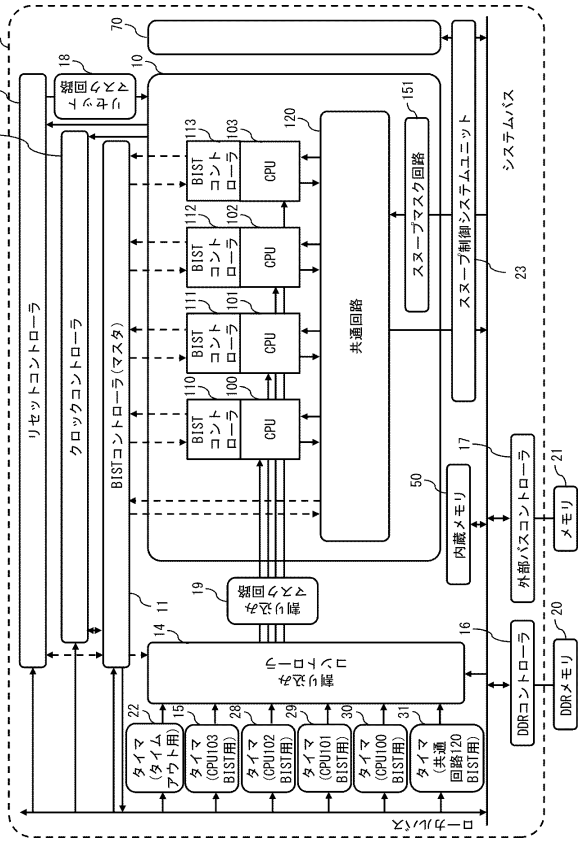
【図２４】



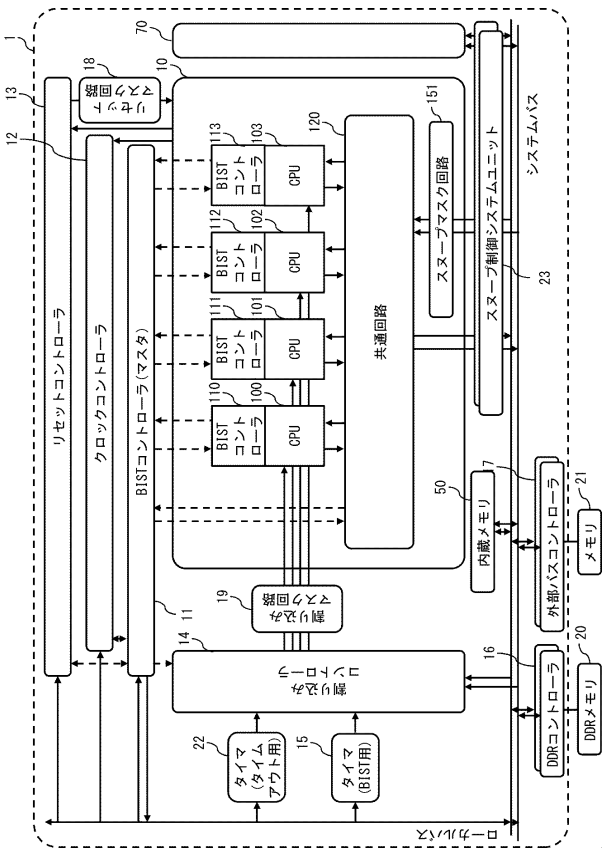
【図２５】



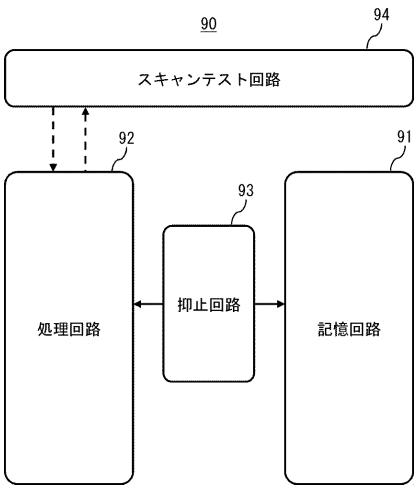
【図２６】



【図２７】



【図２８】



フロントページの続き

審査官　大桃　由紀雄

(56)参考文献　米国特許出願公開第２０１２／０２２６９４２（ＵＳ，Ａ１）
特開２０１０－１２８６２７（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１１／２２
Ｇ０１Ｒ　３１／２８
Ｇ０６Ｆ　１１／２６７