

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2010-193443
(P2010-193443A)

(43) 公開日 平成22年9月2日(2010.9.2)

(51) Int.Cl. F I テーマコード (参考)
H04Q 3/52 (2006.01) H04Q 3/52 A 5K069

審査請求 未請求 請求項の数 32 O L 外国語出願 (全 123 頁)

(21) 出願番号 特願2010-15462 (P2010-15462)
(22) 出願日 平成22年1月27日 (2010.1.27)
(31) 優先権主張番号 12/379, 191
(32) 優先日 平成21年2月13日 (2009.2.13)
(33) 優先権主張国 米国 (US)
(31) 優先権主張番号 12/458, 511
(32) 優先日 平成21年7月14日 (2009.7.14)
(33) 優先権主張国 米国 (US)

(71) 出願人 305024466
ザ リージェンツ オブ ザ ユニバーシ
ティ オブ ミシガン
The Regents of the
University of Michi
gan
アメリカ合衆国, ミシガン州 4810
9, アン アーバー
Ann Arbor, MI 48109
(US)
(74) 代理人 100108453
弁理士 村山 靖彦
(74) 代理人 100064908
弁理士 志賀 正武

最終頁に続く

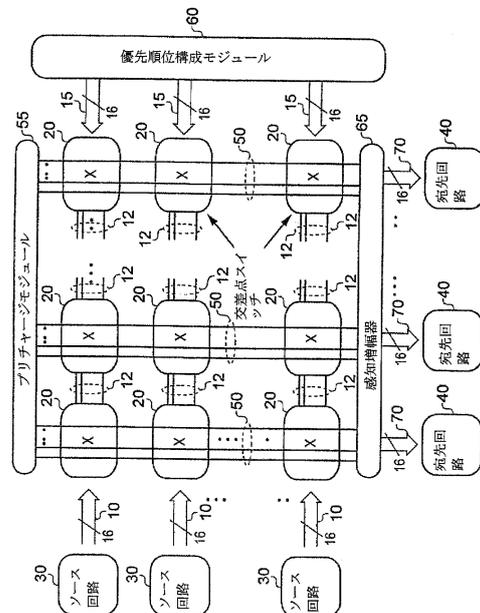
(54) 【発明の名称】 クロスバー回路およびそのようなクロスバー回路の動作方法

(57) 【要約】

【課題】クロスバー回路およびそのようなクロスバー回路の動作方法を提供する。

【解決手段】クロスバー回路はデータ入力経路およびデータ出力経路の配列を有し、データ出力経路はデータ入力経路を横断する。データ入力経路とデータ出力経路との間の各交差点には、ルーティング値を記憶するようにプログラムできる構成記憶回路と、伝送回路と、アービトレーション回路とを備えるクロスバーセルが提供される。伝送動作モードにおいて、伝送回路は、データ入力経路に沿ってデータ入力を検出して、関連付けた交差点においてそのデータの指示をデータ出力経路上に出力するように、ルーティング値に応答する。アービトレーション動作モードにおいて、アービトレーション回路は、関連付けられたソース回路からの複数のアサートされた伝送要求が存在する場合、所定の優先順位スキームに従い複数のアサートされた伝送要求間のコンフリクトを解消する。

【選択図】 図 1



【特許請求の範囲】

【請求項 1】

複数のソース回路と、複数の宛先回路とを相互接続し、よって、前記複数のソース回路のうちのいずれかからクロスバー回路へのデータ入力を、前記複数の宛先回路のうちのいずれかに出力することができる、クロスバー回路であって、

前記クロスバー回路を通過する複数のデータ入力経路であって、各データ入力経路は、前記複数のソース回路のうちの1つに接続でき、かつ複数のワード線を提供する、複数のデータ入力経路と、

前記複数のデータ入力経路を横断する前記クロスバー回路を通過する複数のデータ出力経路であって、各データ出力経路は、前記複数の宛先回路のうちの1つに接続でき、かつ複数のビット線を提供できる、複数のデータ出力経路と、

前記データ入力経路のうちの1つと、前記データ出力経路のうちの1つとの間の各交差点に関連付けられるクロスバーセルであって、各クロスバーセルは、

ルーティング値を記憶するようにプログラムできる構成記憶回路であって、前記ルーティング値は、前記データ入力経路のワード線に沿った前記関連付けた交差点へのデータ入力が、前記関連付けた交差点において、前記データ出力経路のビット線上に出力されることを、第1の値が示すようにプログラムされ、前記ルーティング値は、前記データ入力経路のワード線に沿った前記関連付けた交差点へのデータ入力が、前記関連付けた交差点において、前記データ出力経路のビット線上に出力されないことを、第2の値が示すようにプログラムされる、構成記憶回路と、

伝送動作モードで、前記データ入力経路のワード線に沿って前記データ入力を検出し、前記関連付けた交差点において、そのデータの指示を前記データ出力経路の前記ビット線上に出力するよう、前記第1の値を有する前記ルーティング値に応答する伝送回路と、

前記関連付けた交差点の前記データ入力経路に接続された前記ソース回路から、前記クロスバーセルによって受信された前記伝送要求に基づいて、アービトレーション動作モードで動作するアービトレーション回路であって、伝送要求が、前記ソース回路が前記関連付けた交差点において、前記データ入力経路から前記データ出力経路にデータをルーティングしたいことを示すようにアサートされる場合、前記アービトレーション回路は、前記データ出力経路の前記ビット線を再利用するように同じデータ出力経路と関連付けられた他のクロスバーセルの前記アービトレーション回路と組み合わせ動作し、前記同じデータ出力経路の複数のアサートされた伝送要求の存在を検出し、このような複数のアサートされた伝送要求の場合、所定の優先順位スキームを実装して、前記同じデータ出力経路と関連付けられた1つだけのクロスバーセルの前記構成記憶回路を、そのルーティング値が前記第1の値にプログラムされるように配設され、それによって、前記所定の優先順位スキームに従い前記複数のアサートされた伝送要求間のコンフリクトを解消する、アービトレーション回路と、を備える、クロスバーセルと、

【請求項 2】

各データ出力経路が n 本のビット線を提供し、

各構成記憶回路は、同じデータ出力経路と関連付けられた n 個までのクロスバーセルに対して、それらのクロスバーセルの前記構成記憶回路が、前記 n 本のビット線のうちの関連付けられた異なるビット線を有するように、それを介して前記ルーティング値が前記構成記憶回路にプログラムされる前記 n 本のビット線のうちの1つと関連付けられ、

前記アービトレーション動作モードにおいて、アサートされた伝送要求を受信する同じデータ出力経路と関連付けられた任意のクロスバーセルの前記アービトレーション回路は、前記所定の優先順位スキームに依存する前記 n 本のビット線に対する電圧を選択的に修正するように配設され、

その後、アサートされた伝送要求を受信するクロスバーセルの各構成記憶回路は、前記ルーティング値をプログラムするために、その関連付けられたビット線に対する電圧をサンプリングする、請求項 1 に記載のクロスバー回路。

10

20

30

40

50

【請求項 3】

前記ビット線のそれぞれは、第 1 の電圧レベルにプリチャージされ、次いで、前記アービトレーション動作モード中、アサートされた伝送要求を受信する同じデータ出力経路と関連付けられた任意のクロスバーセルの前記アービトレーション回路は、前記所定の優先順位スキームに依存する前記 n 本のビット線に対する前記電圧を選択的にディスチャージする、請求項 2 に記載のクロスバー回路。

【請求項 4】

各構成記憶回路は、アサートされた伝送要求が存在する場合、前記アービトレーション回路により実行される前記選択的ディスチャージ動作の後にその関連付けられたビット線に対する電圧を感知する感知増幅器有効ラッチを備える、請求項 3 に記載のクロスバー回路。

10

【請求項 5】

各データ入力経路が n 本のワード線を提供し、

前記アービトレーション動作モードにおいて、同じデータ入力経路と関連付けられた n 個までのクロスバーセルに対する前記伝送要求は、その同じデータ入力経路の前記 n 本のワード線を介して前記クロスバー回路に入力される、請求項 1 に記載のクロスバー回路。

【請求項 6】

各構成記憶回路は、同じデータ入力経路と関連付けられた n 個までのクロスバーセルに対して、それらのクロスバーセルの前記構成記憶回路が、それを介して前記伝送要求が受信される前記 n 本のワード線のうちの関連付けられた異なるワード線を有するように、前記 n 本のワード線のうちの 1 つと関連付けられる、請求項 5 に記載のクロスバー回路。

20

【請求項 7】

$m \times n \times m \times n$ マトリクスのクロスバーセルが提供され、式中 m は 2 以上の整数であり、

前記マトリクスは複数のセクションに分割され、

前記所定の優先順位スキームを実装して、前記同じデータ出力経路と関連付けられた 1 つだけのクロスバーセルの前記構成記憶回路を、そのルーティング値が前記第 1 の値にプログラムされるようにするために、一連のアービトレーション動作が使用され、それにより、前記所定の優先順位スキームに従い複数のアサートされた伝送要求間のコンフリクトを解消し、

前記一連における各アービトレーション動作に対して、前記複数のセクションにおける 1 つ以上のセクションがそのアービトレーション動作を受ける、請求項 6 に記載のクロスバー回路。

30

【請求項 8】

各アービトレーション動作は、前記一連のアービトレーション動作が行われた後に前記複数のセクションのすべてが前記アービトレーション動作を受けるように、単一のセクションに対して動作する、請求項 7 に記載のクロスバー回路。

【請求項 9】

前記一連のうちの第 1 のアービトレーション動作において、前記複数のセクションのすべては、ルーティング値が前記第 1 の値にプログラムされる前記構成記憶回路を含有する、前記複数における前記セクションのうちの 1 つを識別するために、前記第 1 のアービトレーション動作を受け、

40

前記一連のうちの第 2 のアービトレーション動作において、前記第 1 のアービトレーション動作により識別された前記セクションのうちの前記 1 つは、ルーティング値が前記第 1 の値にプログラムされるそのセクション内の前記構成記憶回路を識別するために、第 2 のアービトレーション動作を受ける、請求項 7 に記載のクロスバー回路。

【請求項 10】

前記所定の優先順位スキームが固定される、請求項 1 に記載のクロスバー回路。

【請求項 11】

各クロスバーセルは、

前記所定の優先順位スキームを実装するために前記 n 本のビット線のうちの 1 つ以上に

50

関して前記関連付けられたアービトレーション回路により実行される前記アクションを識別する優先順位データを記憶するようにプログラム可能な優先順位記憶回路をさらに備える、請求項 1 に記載のクロスバー回路。

【請求項 1 2】

アサートされたロード優先順位信号を受信する任意の優先順位記憶回路が、それに記憶された優先順位データを更新するために少なくとも 1 つのワード線上のデータをサンプリングするように、優先順位割当動作モードにおいて、前記データ入力経路のうちの 1 つ以上の前記複数のワード線を介して優先順位データを入力し、前記クロスバーセルのうちの 1 つ以上の前記優先順位記憶回路に優先順位ロード信号をアサートする、優先順位構成モジュールをさらに備える、請求項 1 1 に記載のクロスバー回路。

10

【請求項 1 3】

各データ出力経路が n 本のビット線を提供し、

各構成記憶回路は、同じデータ出力経路と関連付けられた n 個までのクロスバーセルに対して、それらのクロスバーセルの前記構成記憶回路が、前記 n 本のビット線のうちの関連付けられた異なるビット線を有するように、それを介して前記ルーティング値が前記構成記憶回路にプログラムされる前記 n 本のビット線のうちの 1 本と関連付けられ、

各クロスバーセル内において、前記アービトレーション回路および前記優先順位記憶回路は、そのクロスバーセルの前記構成記憶回路に関連付けられた 1 本のビット線を除く $n - 1$ 本のビット線と関連付けられ、前記優先順位記憶回路は、前記所定の優先順位スキームを実装するために、それらの $n - 1$ 本のビット線のうちのどれが前記アービトレーション回路により電圧を修正されるべきかを識別する、請求項 1 1 に記載のクロスバー回路。

20

【請求項 1 4】

前記複数のワード線は n 本のワード線を備え、

$n / m \times n / m$ マトリクスのクロスバーセルが提供され、式中 m は 2 以上の整数であり、

各クロスバーセルは、複数の優先順位記憶回路を備え、前記複数の、 m 以下であり、

前記優先順位構成モジュールは、前記データ入力経路の n 本のワード線を使用して、各クロスバーセルに複数の優先順位記憶回路をプログラムするように配設される、請求項 1 2 に記載のクロスバー回路。

【請求項 1 5】

30

前記複数のソース回路は、前記複数のデータ入力経路の第 1 の端部に接続可能であり、

前記優先順位構成モジュールは、前記複数のデータ入力経路の第 2 の端部に接続可能であり、

前記ソース回路は、各データ入力経路の前記複数のワード線を介して、アサートされた伝送要求を発行するように配設され、

前記優先順位構成モジュールは、各ソース回路により発行された前記アサートされた伝送要求を監視するように、また、各ソース回路からのアサートされた伝送要求の数に依存して、前記優先順位割当動作モード中、前記優先順位構成モジュールにより出力される前記優先順位データを修正するように配設される、請求項 1 2 に記載のクロスバー回路。

【請求項 1 6】

40

各クロスバーセルは、

そのクロスバーセルの構成記憶回路と関連付けられた 1 つのビット線と連結された解放回路をさらに備え、アサートされた解放要求の受信後に、チャンネル解放回路が、そのクロスバーセルの前記構成記憶回路における前記ルーティング値として前記第 2 の値を記憶させるレベルまで、その 1 つのビット線に対する電圧を修正させる、請求項 2 に記載のクロスバー回路。

【請求項 1 7】

前記ビット線のそれぞれが、第 1 の電圧レベルにプリチャージされ、

前記アサートされた解放要求の受信後に、前記解放回路は、そのクロスバーセルの前記構成記憶回路と関連付けられた前記 1 本のビット線に対する電圧をディスチャージし、

50

各構成記憶回路は、前記関連付けられた解放回路へのアサートされた解放要求が存在する場合、前記ルーティング値を前記第2の値に更新するために、前記解放回路により実行されるディスチャージ動作の後にその関連付けられたビット線に対する電圧を感知する感知増幅器有効ラッチを備え、それにより、関連付けられたデータ出力経路を解放する、請求項16に記載のクロスバー回路。

【請求項18】

各データ入力経路がn本のワード線を提供し、

同じデータ入力経路と関連付けられたn個までのクロスバーセルに対する前記解放要求は、その同じデータ入力経路の前記n本のワード線を介して前記クロスバー回路に入力され、

10

各解放回路は、同じデータ入力経路と関連付けられたn個までのクロスバーセルに対して、それらのクロスバーセルの前記解放回路が、それを介して前記解放要求が受信される前記n本のワード線のうちの関連付けられた異なるワード線を有するように、前記n本のワード線のうちの1つと関連付けられる、請求項16に記載のクロスバー回路。

【請求項19】

前記関連付けられたデータ入力経路の前記n本のワード線が解放要求または伝送要求を担持しているかどうかを識別するために、前記アービトレーション動作モードにおいて前記解放要求が発行され、前記アービトレーション動作モードにおいて前記複数のソース回路により制御信号が発行される、請求項18に記載のクロスバー回路。

【請求項20】

20

構成記憶回路が前記アービトレーション動作モード中に前記第1の値にプログラムされるそのルーティング値を有する、同じデータ出力経路と関連付けられた前記1つのクロスバーセルに対し、前記アサートされた伝送要求が承諾されたことを確認するために前記関連付けられたソース回路に承諾信号がアサートされる、請求項1に記載のクロスバー回路。

【請求項21】

前記伝送動作モードにおいてクロスバーセルに関連付けられた各データ出力経路は、データ転送の前に第1の論理レベルにプリチャージされ、各クロスバーセルの前記伝送回路は、

前記データ出力経路と第2の論理レベルとの間に直列に接続される、第1および第2のスイッチを備え、

30

前記伝送動作モードにおいて、前記第1のスイッチは、前記関連付けた構成記憶回路内に記憶されているルーティング値に依存して開かれるか、または閉じられ、前記第2のスイッチは、前記データ入力経路上で入力される前記データに依存して開かれるか、または閉じられる、請求項1に記載のクロスバー回路。

【請求項22】

各前記データ入力経路は、前記伝送動作モード中に、nビットの入力データ値を担持するための、n本のワード線を備え、各前記データ出力経路は、前記伝送動作モード中に、nビットのデータ値を担持するための、n本のビット線を備え、少なくとも前記第2のスイッチは、ビット線ごとに複製される、請求項21に記載のクロスバー回路。

40

【請求項23】

前記伝送動作モードにおいて、前記データ出力経路のビット線は、前記ルーティング値が前記第1の値であり、かつ対応するワード線上の前記入力データビットが、前記第1の論理レベルにある場合、前記第2の論理レベルにプルされる、請求項21に記載のクロスバー回路。

【請求項24】

各クロスバーセルは、前記ルーティング値に関わらず前記第1のスイッチをオフにし、一方で、前記関連付けたデータ出力経路を前記第1の論理レベルにプリチャージするための、および前記第1のスイッチを、前記ルーティング値によって制御し、その後、前記関連付けたデータ出力経路を前記第1の論理レベルにプリチャージできるようにするため

50

の、条件付きディスチャージ回路をさらに備える、請求項 2 3 に記載のクロスバー回路。

【請求項 2 5】

前記伝送動作モード中に、前記データ出力経路の前記ビット線の上の前記データ出力を検出し、それによって、前記データ出力経路上のビット線の電圧が、前記第 2 の論理レベルに到達する前に、前記第 2 の論理レベルへの遷移を検出できるようにする、感知増幅回路をさらに備える、請求項 2 3 に記載のクロスバー回路。

【請求項 2 6】

前記複数のソース回路のそれぞれと、前記複数のデータ入力経路との間の符号化回路と、

前記複数のデータ出力経路のそれぞれと、前記複数の宛先回路との間の復号回路と、をさらに備え、

前記符号化回路は、各ソース回路によって提供される入力データを符号化した形態に符号化するように、符号化動作を適用し、該動作は、元の入力データが前記クロスバー回路を通過した場合、前記データ出力経路を前記第 2 の論理レベルにプルするのに必要になり得る回数と比較した時に、前記出力経路を前記第 2 の論理レベルにプルし、その後、それらを前記第 1 の論理レベルにプリチャージするのに必要な回数を低減し、

前記復号回路は、前記データ出力経路上で出力される前記符号化したデータから、前記ソース回路によって提供される前記元の入力データを同定するように、対応する復号動作を適用する、

請求項 2 3 に記載のクロスバー回路。

【請求項 2 7】

前記複数のソース回路は、前記複数のデータ入力経路のいずれかの端部に接続できる、請求項 1 に記載のクロスバー回路。

【請求項 2 8】

前記複数の宛先回路は、前記複数のデータ出力経路のいずれかの端部に接続できる、請求項 1 に記載のクロスバー回路。

【請求項 2 9】

各データ入力経路は、ワード線ドライバにより隔てられた複数の入力経路部分からなり、

各ワード線ドライバは、前記関連付けられたデータ入力経路に沿って提供されるが、そのワード線ドライバよりも前記ソース回路から遠くにあるクロスバーセルの前記構成記憶回路に記憶された前記ルーティング値に基づき有効化され、

各ワード線ドライバは、前記関連付けられたデータ入力経路に沿って提供されるが、そのワード線ドライバよりも前記ソース回路から遠くにある少なくとも 1 つのクロスバーセルの前記構成記憶回路が、前記第 1 の値に設定されたルーティング値を有する場合にのみ有効化される、請求項 1 に記載のクロスバー回路。

【請求項 3 0】

データ処理装置であって、

データ値を記憶するための複数のメモリデバイスと、

前記複数のメモリデバイスに記憶されている複数のデータ値に対して、並列してデータ処理動作を実行するための、複数のプロセッサと、

前記複数のメモリデバイスのうちのいずれかのメモリデバイスから、前記複数のプロセッサのうちのいずれかのプロセッサに、前記データ値をルーティングするための、請求項 1 に記載のクロスバー回路と、

を備える、データ処理装置。

【請求項 3 1】

複数のソース手段と、複数の宛先手段とを相互接続し、よって、前記複数のソース手段のうちのいずれかからクロスバー回路へのデータ入力を、前記複数の宛先手段のうちのいずれかに出力することができる、クロスバー回路であって、

前記クロスバー回路を通過する複数のデータ入力経路手段であって、各データ入力経路

10

20

30

40

50

手段は、前記複数のソース手段のうちの1つに接続し、かつ複数のワード線手段を提供するための、データ入力経路手段と、

前記複数のデータ入力経路手段を横断する前記クロスバー回路を通過する複数のデータ出力経路手段であって、各データ出力経路手段は、前記複数の宛先手段のうちの1つに接続し、かつ複数のビット線手段を提供するための、データ出力経路手段と、

前記データ入力経路手段のうちの1つと、前記データ出力経路手段のうちの1つとの間の各交差点に関連付けられるクロスバーセル手段であって、各クロスバーセル手段は、

ルーティング値を記憶するためにプログラムできる構成記憶手段であって、前記ルーティング値は、前記データ入力経路手段の前記ワード線手段に沿った前記関連付けた交差点へのデータ入力が、前記関連付けた交差点において、前記データ出力経路手段の前記ビット線手段上に出力されることを、第1の値が示すようにプログラムされ、前記ルーティング値は、前記データ入力経路手段の前記ワード線手段に沿った前記関連付けた交差点へのデータ入力が、前記関連付けた交差点において、前記データ出力経路手段の前記ビット線手段上に出力されないことを、第2の値が示すようにプログラムされる、構成記憶手段と

、
前記データ入力経路手段の前記ワード線手段に沿って前記データ入力を検出し、前記関連付けた交差点において、そのデータの指示を前記データ出力経路手段の前記ビット線手段上に出力するための、伝送動作モードで、前記第1の値を有する前記ルーティング値に
10 応答する伝送手段と、

前記関連付けた交差点の前記データ入力経路手段に接続された前記ソース手段から、前記クロスバーセル手段によって受信された伝送要求に基づいて、アービトレーション動作モードで動作するためのアービトレーション手段であって、前記ソース手段が、前記関連付けた交差点において、前記データ入力経路手段から前記データ出力経路手段へデータをルーティングしたいことを示すように、前記伝送要求がアサートされる場合、前記アービトレーション手段は、同じデータ出力経路手段と関連付けられた他のクロスバーセル手段の前記アービトレーション手段と組み合わせて動作し、前記データ出力経路手段の前記ビット線手段を再利用して、前記同じデータ出力経路手段の複数のアサートされた伝送要求の存在を検出し、このような複数のアサートされた伝送要求の場合、所定の優先順位スキームを実装して、前記同じデータ出力経路手段と関連付けられた1つだけのクロスバーセル手段の前記構成記憶手段を、そのルーティング値が前記第1の値にプログラムされるようにし、それによって、前記所定の優先順位スキームに従い、前記複数のアサートされた伝送要求間のコンフリクトを解消する、アービトレーション手段と、を備える、クロスバーセル手段と、

を備える、クロスバー回路。

【請求項32】

複数のソース回路と複数の宛先回路とを相互接続し、よって、前記複数のソース回路のうちのいずれかからクロスバー回路へのデータ入力を、前記複数の宛先回路のうちのいずれかにも出力することができ、前記クロスバー回路は、前記クロスバー回路を通過する複数のデータ入力経路を有し、各データ入力経路は、前記複数のソース回路のうちの1つに接続でき、かつ複数のワード線を提供し、複数のデータ出力経路は、前記複数のデータ入力経路を横断する前記クロスバー回路を通過し、各データ出力経路は、前記複数の宛先回路のうちの1つに接続でき、かつ複数のビット線を提供する、クロスバー回路を動作させる方法であって、

前記データ入力経路のうちの1つと、前記データ出力経路のうちの1つとの間の各交差点を関連付けて、クロスバーセルを採用するステップと、

各クロスバーセルでルーティング値をプログラミングするステップであって、前記ルーティング値は、前記データ入力経路の前記ワード線に沿った前記関連付けた交差点へのデータ入力が、前記関連付けた交差点において、前記データ出力経路の前記ビット線
40 上に出力されることを、第1の値が示すようにプログラムされ、前記ルーティング値は、前記データ入力経路の前記ワード線に沿った前記関連付けた交差点へのデータ入力が、前記関連
50

付けた交差点において、前記データ出力経路のビット線に出力されないことを、第2の値が示すようにプログラムされる、ステップと、

伝送動作モードにおいて、前記データ入力経路の前記ワード線に沿って前記データ入力を検出し、前記関連付けた交差点において、そのデータの指示を、前記データ出力経路の前記ビット線に出力するよう、前記クロスバーセルを、前記第1の値を有する前記ルーティング値に応答させるステップと、

アービトレーション動作モードにおいて、前記関連付けた交差点の前記データ入力経路に接続された前記ソース回路から、前記クロスバーセルによって受信された伝送要求に基づいて、前記クロスバーセル内のアービトレーション回路を動作させるステップであって、前記ソース回路が、前記関連付けた交差点において、前記データ入力経路から前記データ出力経路へデータをルーティングしたいことを示すように、前記伝送要求がアサートされる場合、前記アービトレーション回路は、同じデータ出力経路と関連付けられた他のクロスバーセルの前記アービトレーション回路と組み合わせ動作し、データ出力経路のビット線を再利用して、前記同じデータ出力経路の複数のアサートされた伝送要求の存在を検出し、このような複数のアサートされた伝送要求の場合、所定の優先順位スキームを実装して、前記同じデータ出力経路と関連付けられた1つだけのクロスバーセルの前記構成記憶回路を、そのルーティング値が前記第1の値にプログラムされるようにし、それによって、前記所定の優先順位スキームに従い前記複数のアサートされた伝送要求間のコンフリクトを解消する、ステップと、

を含む、方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、クロスバー回路およびそのようなクロスバー回路の動作方法に関する。

【背景技術】

【0002】

クロスバー回路は、マトリクス様態で複数の入力を複数の出力に接続するためのスイッチ基盤である。したがって、クロスバー回路を使用して、複数のソース回路と、複数の宛先回路とを相互接続することができ、よって、複数のソース回路のうちのいずれかからクロスバー回路へのデータ入力を、複数の宛先回路のうちのいずれかに出力することができる。クロスバー回路は、様々な実現形態で使用することができる。例えば、データ処理システムの実現形態では、このようなクロスバー回路を使用して、データ値に対してデータ処理動作を実行するのに使用される複数のプロセッサと、それらのデータ値を記憶するのに使用される複数のメモリデバイスとを相互接続することができ、それによって、データ値を、任意のメモリデバイスから任意のプロセッサにルーティングできるようにする。

【0003】

クロスバー回路を生成するための既知の手法は、クロスバー回路を形成するのに必要な構成要素に起因して、クロスバー回路のための大きな面積と、制御信号をそれらの構成要素にルーティングするために必要な多数の制御線と、を必要とし、また、大量の電力も消費する。さらに、それらの複雑さは、サイズとともに急激に増大し、既知の手法の大部分を、多数のソース回路と多数の宛先回路とを相互接続するのに必要なクロスバー回路との使用について、非実用的なものとする。いくつかの既知の手法を以下に述べる。

【0004】

K Chang 他による論文「A 50Gb/s 32x32 CMOS Crossbar Chip using Asymmetric Serial Links」、1999 Symposium on VLSI Circuits, Digest of Technical Papers、19~22ページ、およびT Wu 他による論文「A 2Gb/s 256x256 CMOS Crossbar Switch Fabric Core Design using Pipelined MUX」、IEEE International Symposium on Circuits

10

20

30

40

50

and System、2002、568～571ページは、任意の入力ソースから任意の出力宛先へのデータのルーティングを可能にするように、マルチプレクサの階層的配設を使用したクロスバー回路を記載している。しかしながら、このようなMUXに基づくクロスバー回路は、比較的サイズが大きく、電力消費が多い。さらに、それらは、一般的に、種々のマルチプレクサを制御するために、相当な数の制御線を必要とする。このようなMUXに基づく設計は、一般的に、少なくとも部分的には、サイズが大きくなるにつれて、必要な制御信号を種々のマルチプレクサにルーティングすることがますます困難になるため、サポートする入力および出力数の増加により拡張することができない。さらに、入力データが、入力パス上でルーティングされるマルチビットデータである場合、データ経路自体のルーティングが非常に複雑になる。

10

【0005】

R Golshan他による論文「A Novel Reduced Swing CMOS Bus Interface Circuit for High Speed Low Power VLSI Systems」、IEEE International Symposium on Circuits and System、351～354ページ、1994は、入力経路が水平方向に通り、出力経路が垂直方向に通る、X-Y方式のクロスバー回路を記載している。各入力経路と出力経路との間の交差点には、フリップフロップ回路の形態で記憶素子が提供され、その出力は、入力経路と出力経路とを連結するのに使用されるトランジスタを制御する。しかしながら、このような設計は、クロスバー回路が所要のルーティングを実行するために、種々のフリップフロップをプログラムするのに多数の制御線を必要とする。さらに、入力データ経路上に提供される入力データは、関連する出力データ経路上の出力データを駆動するのに使用される。多数の入力および出力に適應するようにクロスバー回路が大きくなるにつれて、出力データ経路の容量が増加し、それに応じて、容量の増加を克服するために、より大きな駆動トランジスタを入力に提供する必要がある。加えて、データ入力経路とデータ出力経路との間の交差点で、フリップフロップによって駆動される連結トランジスタは、クロスバー回路のサイズが増加した時に、サイズを増加させる必要もある。さらに、一般的に、クロスバー回路のサイズが増加した時には、データ出力経路内に1つ以上のバッファを含めることが必要となる。これらの全ての要因は、より具体的にはより多くの入力および出力に適應するようにクロスバー回路のサイズを増加させた時に、クロスバー回路の素子および関連付けた制御線のレイアウトに重大な問題を生じさせる。したがって、この手法は、複雑になり、拡張することができない。

20

30

【0006】

P Wijetungaによる論文「High-Performance Crossbar Design for System-On-Chip」、Proceedings of the Third IEEE International Workshop on System-On-Chip for Real-Time Applications、2003は、データ入力経路とデータ出力経路との間の各交差点に位置付けられる伝送回路としてパストラジスタチェーンを採用する、クロスバー設計を記載している。入力データを出力データ経路へ連結するのに、パストラジスタチェーンが必要である時、出力データ経路上に位置付けられる電流感知回路を使用して、入力データ値を検出する。この設計の極めて不利な点は、電流感知デバイスを各出力経路上に位置付けるための要件に起因して、多量の電力を消費することである。さらに、クロスバー回路内で多数の制御信号をルーティングする必要があり、実際に、それらの制御信号は、より多くの入力および出力に適應するようにクロスバー回路のサイズを増加させるにつれて、ルーティング要件を支配するようになる。したがって、ここでも、この設計は、より大きいクロスバー回路の設計に、容易に拡張することができない。

40

【0007】

M Borgatti他による論文「A Multi-Context 6.4 Gb/s/Channel On-Chip Communication Network

50

using 0.18 μ m Flash-EEPROM Switches and Elastic Interconnects」、ISSCC 2003、Session 26、Embedded and Digital Systems、Paper 26.5は、改良したフラッシュEEPROMデバイスのマトリクスを使用して実現した、プログラム可能なクロスバーを記載している。しかしながら、図26.5.6から明らかのように、ソースデバイスから宛先デバイスへのデータ入力のルーティングに關与するメモリセルの数は、それらのデバイスがクロスバー内に接続される場所に依存し、したがって、信号がクロスバーを通過するタイミングは、決定論的ではない。さらに、種々のフラッシュEEPROMセルのプログラミングを可能にするには、多数の制御線が必要となり、それらの種々のフラッシュEEPROMセルをプログラムするには、かなり多くの時間が必要となる。したがって、クロスバーデバイスのあらゆる再構成にも、かなり時間がかかる。

10

【0008】

したがって、このようなクロスバーの設計は、複雑であり、また、必要とされる制御線の急増により、より多くの入力デバイスおよび出力デバイスをクロスバーによってサポートすることが必要になるので、複雑さが増加する。さらに、クロスバーのタイミングは、決定論的でないため、クロスバーの設計を、特定の實現形態について不適当なものとする。

【0009】

要約すると、上述の考察から、既存のクロスバー設計は、一般的に、制御信号の複雑なルーティングを伴い、その複雑さは、クロスバーのサイズが増加するにつれて、急激に増加することが理解されよう。必要な制御線の数に部分的に起因して、およびクロスバーのサイズが増加するにつれて、クロスバー内に提供される特定の構成要素のサイズを増加させる必要性に部分的に起因して、しばしば、該設計は、大量の電力を消費し、かつ拡張性に乏しい。

20

【0010】

クロスバー設計における別の問題は、コリジョン検出および解消能力を備えたクロスバーをいかに提供するかである。スイッチングファブリックにおける同じ宛先に対する複数の要求は、コリジョンと呼ばれる。ソースおよび宛先の数が増加するに伴い、コリジョンはより頻繁となる。そのような状況下では、クロスバー回路の全体的な効率においてアービトレーションが障壁となる。

30

【0011】

最も最近のスイッチングファブリックは、主として、データを伝送するクロスバー、およびクロスバーを構成するアービターの2つのモジュールからなる。そのような實現形態において、ソース回路は、アービターにチャンネルの要求を送信する。アービターはすべての要求をサンプリングし、いくつかの優先順位割当スキームを使用して要求の一部またはすべてを承諾し、それに従いクロスバーを構成する。このスキームは、拡張性に関して次の2つの大きな問題を有する。

【0012】

1) ソース回路からアービターへのすべての要求信号、およびすべての承諾信号の返信をルーティングすることは、より大きなシステムではますます困難となる。

40

2) アービターは、決定を下すことができるためには、受信するすべての要求だけでなくクロスバーの現在の状態を知っている必要がある。周期ごとにクロスバーの状態を監視するには、追加の論理および相互接続が必要である。これは追加的な遅延の一因となる。

【0013】

パケットスイッチングネットワークの拡張性およびアービトレーション遅延に対応するために、多くの試みがなされている。Chi H. 他による論文「Decomposed Arbiters for Large Crossbars with Multi-Queue Input Buffers」、IEEE International Conference on Computer Design、14-16 Oct

50

1991、233～238ページにおいて、著者は、アービトレーションプロセスが完了する前にいくつかの要求が承諾され得るようなアービターの分解について述べている。しかしながら、最悪条件のアービトレーション遅延は依然として同じままである。一般に、アービトレーション遅延はサイズとともに直線的に増大する。

【0014】

Delgado-Frias 他による論文「A VLSI Crossbar Switch with Wrapped Wave Front Arbitration」、IEEE Transactions on Circuits and Systems, Volume 50, Issue 1, Jan. 2003, 135～141ページ、および Kavaldjiev N. 他による論文「A Virtual Channel Router for On-chip Networks」、IEEE International SoC Conference, 12-15 Sept. 2004, 289～293ページでは、著者は、クロスバー内におけるアービトレーションの取扱いについて述べている。しかしながら、開示された実現形態は、拡張性がなく、4*4のサイズのクロスバーに制限されている。

10

【0015】

Shin E. 他による論文「Round-robin Arbiter Design and Generation」、International Symposium on System Synthesis, 2002, 243～248ページでは、著者は、ラウンドロビンアービターを生成するツールを提案している。この手法は階層的であり、4*4スイッチから32*32スイッチを形成しようとしている。

20

【0016】

William W. Plummer による論文「Asynchronous Arbiters」、IEEE Transactions on Computers Archive Volume 21, Issue 1 (January 1972)、37～42ページ、Charles E. Molnar 他による論文「Simple Circuits that Work for Complicated Reasons」、International Symposium on Advanced Research in Asynchronous Circuits and Systems, 2000. (ASync 2000) Proceedings、および Mark B. Josephs 他による論文「CMOS Design of the Tree Arbiter Element」、IEEE Transactions on VLSI systems, Volume 4, Issue 4, Dec 1996, 472～476ページでは、アービトレーションのためのいくつかの非同期的技法が記載されている。これらの設計は、無作為性を達成するためにメタ安定性を利用する。しかしながら、メタ安定システムにおける遅延が時折高くなる可能性があり、それにより確実なスループットを必要とするリアルタイムシステムにおけるその使用は制限されている。

30

【0017】

要約すると、文献に見られる多くの解決法は、2*2または4*4スイッチを使用した大型クロスバースイッチの作製に対する階層的手法を使用する。したがって、クロスバースイッチのサイズとともに遅延が直線的に増大する。これにより、幅広い並列システム(SIMD/MIMD型アプリケーション)において電圧拡張性の余地がなくなる。

40

【0018】

さらに、例えば上述した論文「A Virtual Channel Router for On-chip Networks」、IEEE International SoC Conference, 12-15 Sept. 2004, 289～293ページに述べられているような、アービターおよびクロスバーを互いに統合する以前の試みにおいては、それらを空間的に近接させることのみが可能となっている。しかしながら、両方の機能性のための論理および相互接続は、相容れないままであった。

50

【 0 0 1 9 】

従来の実現形態において、コリジョン検出および解消は階層的に行われる。これはアービトレーション論理の複数の段階において達成され、各段階の後に要求の数がある特定の割合低減される。

【 先行技術文献 】

【 非特許文献 】

【 0 0 2 0 】

【 非特許文献 1 】 K Chang 他, “ A 50 Gb/s 32x32 CMOS Crossbar Chip using Asymmetric Serial Links ” 1999 Symposium on VLSI Circuits, Digest of Technical Papers, Pages 19 to 22

10

【 非特許文献 2 】 T Wu 他, “ A 2 Gb/s 256x256 CMOS Crossbar Switch Fabric Core Design using Pipelined MUX ”, IEEE International Symposium on Circuits and System, 2002, Pages 568 to 571

【 非特許文献 3 】 R Golshan 他, “ A Novel Reduced Swing CMOS Bus Interface Circuit for High Speed Low Power VLSI Systems ”, IEEE International Symposium on Circuits and System, Pages 351 to 354, 1994

【 非特許文献 4 】 P Wijetunga, “ High-Performance Crossbar Design for System-On-Chip ”, Proceedings of the Third IEEE International Workshop on System-On-Chip for Real-Time Applications, 2003

【 非特許文献 5 】 M Borgatti 他, “ A Multi-Context 6.4 Gb/s/Channel On-Chip Communication Network using 0.18 μ m Flash-EEPROM Switches and Elastic Interconnects ”, ISSCC 2003, Session 26

20

【 非特許文献 6 】 Chi H. 他, “ Decomposed Arbiters for Large Crossbars with Multi-Queue Input Buffers ”, IEEE International Conference on Computer Design, 14-16 Oct 1991, pages: 233-238

【 非特許文献 7 】 Delgado-Frias 他, “ A VLSI Crossbar Switch with Wrapped Wave Front Arbitration ”, IEEE Transactions on Circuits and Systems, Volume 50, Issue 1, Jan. 2003 Pages: 135-141

【 非特許文献 8 】 Kavaljdjiev N. 他, “ A Virtual Channel Router for On-chip Networks ”, IEEE International SoC Conference, 12-15 Sept. 2004, pages: 289-293

30

【 非特許文献 9 】 Shin E. 他, “ Round-robin Arbiter Design and Generation ”, International Symposium on System Synthesis, 2002, pages: 243-248

【 非特許文献 10 】 William W. Plummer, “ Asynchronous Arbiters ”, IEEE Transactions on Computers Archive Volume 21, Issue 1 (January 1972) Pages 37-42

【 非特許文献 11 】 Charles E. Molnar 他, “ Simple Circuits that Work for Complicated Reasons ”, International Symposium on Advanced Research in Asynchronous Circuits and Systems, 2000. (ASYNC 2000) Proceedings

【 非特許文献 12 】 Mark B. Josephs 他, “ CMOS Design of the Tree Arbiter Element ”, IEEE Transactions on VLSI systems, Volume 4, Issue 4, Dec 1996 Page(s):472-476

40

【 発明の概要 】

【 発明が解決しようとする課題 】

【 0 0 2 1 】

したがって、上記の問題を緩和する、改良されたクロスバー設計を提供することが望ましい。

【 課題を解決するための手段 】

【 0 0 2 2 】

第1の態様から見ると、本発明は、複数のソース回路と、複数の宛先回路とを相互接続し、よって、該複数のソース回路のうちのいずれかからクロスバー回路へのデータ入力を、該複数の宛先回路のうちのいずれかに出力することができる、クロスバー回路を提供し

40

、該クロスバー回路は、該クロスバー回路を通過する複数のデータ入力経路であって、各データ入力経路は、該複数のソース回路のうちの1つに接続でき、かつ複数のワード線を提供する、データ入力経路と、複数のデータ入力経路を横断する該クロスバー回路を通過する複数のデータ出力経路であって、各データ出力経路は、該複数の宛先回路のうちの1つに接続でき、かつ複数のビット線を提供する、データ出力経路と、該データ入力経路のうちの1つと該データ出力経路のうちの1つとの間の各交差点に関連付けられるクロスバーセルと、を備え、各クロスバーセルは、ルーティング値を記憶するようにプログラムできる構成記憶回路であって、ルーティング値が、データ入力経路のワード線に沿った関連付けた交差点へのデータ入力が、関連付けた交差点において、データ出力経路のビット線に出力されることを、第1の値が示すようにプログラムされ、ルーティング値が、データ入力経路のワード線に沿った関連付けた交差点へのデータ入力が、関連付けた交差点において、データ出力経路のビット線に出力されないことを、第2の値が示すようにプログラムされる、構成記憶回路と、伝送動作モードにおいて、データ入力経路のワード線に沿ってデータ入力を検出し、関連付けた交差点において、そのデータの指示を、データ出力経路のビット線に出力するよう、該第1の値を有するルーティング値に応答する、伝送回路と、関連付けた交差点のデータ入力経路に接続されたソース回路からクロスバーセルによって受信された伝送要求に基づいて、アービトレーション動作モードで動作するアービトレーション回路であって、該ソース回路が、関連付けた交差点において、データ入力経路からデータ出力経路にデータをルーティングしたいことを示すように、伝送要求がアサートされる場合、アービトレーション回路は、同じデータ出力経路に関連付けられた他のクロスバーセルのアービトレーション回路と組み合わせて動作し、データ出力経路のビット線を再利用して、該同じデータ出力経路の複数のアサートされた伝送要求の存在を検出し、このような複数のアサートされた伝送要求の場合、所定の優先順位スキームを実装して、該同じデータ出力経路と関連付けられた1つだけのクロスバーセルの構成記憶回路を、そのルーティング値が該第1の値にプログラムされるように配設し、それによって、該所定の優先順位スキームに従い該複数のアサートされた伝送要求間のコンフリクトを解消する、アービトレーション回路と、を備える。

【0023】

本発明によれば、クロスバーセルは、データ入力経路とデータ出力経路との間の各交差点と関連付けられ、各クロスバーセルは、伝送動作モードにおいて、それに記憶されたルーティング値に基づきそのデータ入力経路上のデータをそのデータ出力経路に選択的に接続するように配設された構成記憶回路と、アービトレーション動作モードにおいて、複数のアサートされた伝送要求間のコンフリクトを解消するために、他のクロスバーセルのアービトレーション回路と組み合わせて動作するアービトレーション回路とを備える。具体的には、伝送要求がソース回路からクロスバーセルに対しアサートされた場合、そのクロスバーセルのアービトレーション回路は、同じデータ出力経路と関連付けられた他のクロスバーセルのアービトレーション回路と組み合わせて動作し、データ出力経路のビット線を再利用して同じデータ出力経路の複数のアサートされた伝送要求の存在を検出する。そのような複数のアサートされた伝送要求の場合、アービトレーション回路は、所定の優先順位スキームを実装して、同じデータ出力経路と関連付けられた1つだけのクロスバーセルの構成記憶回路を、そのルーティング値が第1の値(すなわち、伝送動作モードにおいて、クロスバーセルに、そのデータ入力経路をそのデータ出力経路に連結させる値)にプログラムされるようにするために、組み合わせて動作する。

【0024】

したがって、本発明によれば、アービトレーション動作モードにおいて、特定のデータ出力経路のコンフリクトを検出するために、および任意の時点で1つだけのソース回路が特定のデータ出力経路へのアクセスを承諾されるようにコンフリクトを解消するために、データ出力経路のビット線が再利用される。

【0025】

本発明の配設において、アービターおよびクロスバー機能の両方が、ともにクロスバー

回路内に統合される。アービターは極めて論理中心であるがクロスバーは極めてルーティング中心であり、クロスバー回路のクロスバーセル内でこれらの2つの機能を互いにブリッジすることにより、ルーティングトラックおよびチップにおけるシリコン空間のより効率的な利用が達成される。さらに、このように従来のクロスバーとアービターとの間に生じる通信オーバーヘッドが軽減される。クロスバー回路内の各交差点に関連付けられたクロスバーセルにクロスバー構成を記憶することにより、アービトレーションのためのデータ出力経路のビット線の再利用が可能となる。

【0026】

本発明によれば、クロスバー回路は、SRAMアレイトポロジーの形式でレイアウトすることができ、クロスバーセルがアレイとしてレイアウトされ、データ入力経路がワード線を提供し、データ出力経路がビット線を提供する。そのような配設により、サイズおよび必要とされる相互接続の数の双方の観点から、非常に効率的なレイアウトが可能となる。

10

【0027】

一実施形態において、各データ出力経路は n 本のビット線を提供し、各構成記憶回路は、同じデータ出力経路と関連付けられた n 個までのクロスバーセルに対して、それらのクロスバーセルの構成記憶回路が、該 n 本のビット線のうちの関連付けられた異なるビット線を有するように、それを介してルーティング値が構成記憶回路にプログラムされる該 n 本のビット線のうちの1つと関連付けられる。該アービトレーション動作モードにおいて、アサートされた伝送要求を受信する同じデータ出力経路と関連付けられた任意のクロスバーセルのアービトレーション回路は、該所定の優先順位スキームに依存する n 本のビット線に対する電圧を選択的に修正するように配設され、その後、アサートされた伝送要求を受信するクロスバーセルの各構成記憶回路は、該ルーティング値をプログラムするために、その関連付けられたビット線に対する電圧をサンプリングする。

20

【0028】

同じデータ出力経路における異なるクロスバーセルの構成記憶回路が異なるビット線と関連付けられ、またアサートされた伝送要求を受信するそのデータ出力経路におけるそれらのクロスバーセルのアービトレーション回路が n 本のビット線に対する電圧を選択的に修正するため、アサートされた伝送要求を受信する各クロスバーセルの構成記憶回路がアービトレーションプロセスの後にその関連付けられたビット線上の電圧をサンプリングする時点で、ビット線の1つだけが、関連付けられた構成記憶回路に該第1の値をルーティング値として記憶させる値にあることを確実にすることにより、所定の優先順位スキームを実装することが可能であり、それにより、単一のソース回路が任意の時点でデータ出力経路へのアクセスを承諾されることが可能となる。

30

【0029】

アービトレーション回路が n 本のビット線に対する電圧を選択的に修正する状態では、実現形態に依存して変化し得る。しかしながら、一実施形態において、ビット線のそれぞれは第1の電圧レベルにプリチャージされ、次いでアービトレーション動作モード中、アサートされた伝送要求を受信する同じデータ出力経路と関連付けられた任意のクロスバーセルのアービトレーション回路は、該所定の優先順位スキームに依存する n 本のビット線に対する電圧を選択的にディスチャージする。したがって、そのような実施形態において、ビット線は、所定の優先順位スキームを実装するために条件付きでディスチャージされ、それにより、アービトレーション動作モード中のいかなるコンフリクトも解消される。

40

【0030】

ソース回路からクロスバーセルに伝送要求がアサートされ得る様式は数多くある。しかしながら、一実施形態において、各データ入力経路は n 本のワード線を提供し、アービトレーション動作モードにおいて、同じデータ入力経路と関連付けられた n 個までのクロスバーセルに対する伝送要求は、その同じデータ入力経路の n 本のワード線を介してクロスバー回路に入力される。したがって、そのような実施形態において、データ入力経路は、アサートされた伝送要求の提供のためにそれ自体アービトレーション動作モードにおいて

50

再利用される。

【0031】

具体的な一実施形態において、各構成記憶回路は、同じデータ入力経路と関連付けられた n 個までのクロスパーセルに対して、それらのクロスパーセルの構成記憶回路が、それを介して伝送要求が受信される該 n 本のワード線のうちの関連付けられた異なるワード線を有するように、該 n 本のワード線のうちの 1 つと関連付けられる。

【0032】

データ入力経路の n 本のワード線およびデータ出力経路の n 本のビット線がアービトレーション動作モード中に再利用されるそのような実施形態において、コンフリクト検出および解消が単一段階で実行可能であり、それにより、コリジョン検出および解消が複数の

10

【0033】

具体的には、 $n \times n$ までのマトリクスのクロスパーセルを備えるクロスパー回路において、コンフリクト検出および解消が単一段階で実行可能であることが分かる。

【0034】

しかしながら、本発明の実施形態の技法は、より大きなマトリクスのクロスパーセルを有するクロスパー回路とともに使用することもできる。具体的には、一実施形態において、 $m \times n \times m \times n$ (m は 2 以上の整数である) マトリクスのクロスパーセルが提供され、マトリクスは複数のセクションに分割され、所定の優先順位スキームを実装して、該同じデータ出力経路と関連付けられた 1 つだけのクロスパーセルの構成記憶回路を、そのルーティング値が該第 1 の値にプログラムされるようにするために、一連のアービトレーション動作が使用され、それにより、該所定の優先順位スキームに従い複数のアサートされた伝送要求間のコンフリクトを解消する。一連における各アービトレーション動作に対して、該複数のセクションにおける 1 つ以上のセクションがそのアービトレーション動作を受ける。

20

【0035】

具体的な一実施形態において、各アービトレーション動作は、該一連のアービトレーション動作が行われた後に該複数のセクションのすべてが該アービトレーション動作を受けるように、単一のセクションに対して動作する。したがって、例えば、16 ビットデータ入力経路およびデータ出力経路を備える 64×64 クロスパー回路の場合、クロスパー回路は 4 つのセクションに分割することができ、アービトレーション動作はそれらのセクションのそれぞれに対し順次実行される。一実施形態において、周期ごとにカウンタをインクリメントして、その周期においてアービトレーション中に処理するセクションを選択することができる (上記の例では 2 ビットカウンタで十分である)。そのような実現形態はまだ最小限の配線および論理オーバーヘッドの利益を維持するが、所与の要求に対するアービトレーション待ち時間は、その要求が関連付けられているセクションに依存して変化し得る (上記の例では、待ち時間は 1 周期から 4 周期まで変化し得る)。

30

【0036】

代替の実施形態において、一連のうちの第 1 のアービトレーション動作において、複数のセクションのすべては、ルーティング値が該第 1 の値にプログラムされる構成記憶回路を含有する、該複数のうちの 1 つのセクションを識別するために、第 1 のアービトレーション動作を受ける。次いで、一連のうちの第 2 のアービトレーション動作において、第 1 のアービトレーション動作により識別されたセクションのうちの該 1 つは、ルーティング値が該第 1 の値にプログラムされるそのセクション内の構成記憶回路を識別するために、第 2 のアービトレーション動作を受ける。そのような実施形態によれば、クロスパーは再びセクションに分割されるが、その場合アービトレーションは 2 つのステップ、つまり、まずセクション間、次いで所与のセクションにおけるすべての要求間で階層的に実行される。両方のアービトレーションステップに、同じ群のビット線を使用することができる。

40

【0037】

50

一実施形態において、各クロスパーセルは2つの構成記憶回路を備え、第1は関連付けられたセクションが最も高い優先順位のアサートされた伝送要求を含有するかどうかを検出するために使用され、第2は最も高い優先順位のアサートされた伝送要求がその特定のクロスパーセルと関連付けられているかどうかを検出するための後続のアービトレーションステップにおいて使用される。

【0038】

例えば、16ビットデータ入力経路およびデータ出力経路を備える64×64クロスパーセルにおいて、クロスパーセルは再び4つのセクションに分割され得る。セクション0が最も高い優先順位を有し、そのセクションと関連付けられた少なくとも1つのソースが伝送要求をアサートする場合、アービトレーションの第1の段階では、そのセクションにおける各クロスパーセル内の第1の構成記憶回路は、ルーティング値が第1の値にプログラムされ、一方他のセクションにおけるそれらはそのルーティング値が第2の値にプログラムされる(両方の場合において、プログラムは、第1の構成記憶回路それぞれに関連付けられたビット線に対する電圧の結果生じる)。次の周期において、セクション0に関してアサートされた伝送要求のみが考慮され、ビット線はまたアービトレーションに使用され、ここで第2の構成記憶回路は関連付けられたビット線を感知する。特定のクロスパーセルで両方の構成記憶回路がそのルーティング値を第1の値に設定された場合、これは、関連付けられたアサートされた伝送要求が優先し、データ伝送動作モードにおいてそのクロスパーセルがその入力をその出力に接続することを意味する。

10

【0039】

そのような実現形態は、いくつかの追加的な論理を犠牲にするが追加的な相互接続は犠牲にせず、固定された2周期アービトレーション待ち時間を有する。データ出力経路における16本のビット線では、16個までのセクション間を調停することが可能であり、各セクションは16個までのアサートされた伝送要求を受信することができる。したがって、アービトレーション待ち時間は、256×256クロスパーセルまでのクロスパーセルに対し2周期に維持され得る。

20

【0040】

アービトレーション動作モード中にアービトレーション回路により実装される所定の優先順位スキームは、様々な形態をとり得る。一実施形態において、所定の優先順位スキームが固定され得る。しかしながら、代替の実施形態において、各クロスパーセルは、所定の優先順位スキームを実装するためにn本のビット線のうちの1つ以上に関して関連付けられたアービトレーション回路により実行されるアクションを識別する優先順位データを記憶するようにプログラム可能な優先順位記憶回路をさらに備える。したがって、これは、所定の優先順位スキームが経時的に再プログラムされることを可能とし、これによりアービトレーションスキームに様々なソース回路によるクロスパーセル回路へのアクセスの公平性を改善させることができる。

30

【0041】

一実施形態において、クロスパーセル回路は、アサートされたロード優先順位信号を受信する任意の優先順位記憶回路が、それに記憶された優先順位データを更新するために少なくとも1つのワード線上のデータをサンプリングするように、優先順位割当動作モードにおいて、データ入力経路のうちの1つ以上の複数のワード線を介して優先順位データを入力し、クロスパーセルのうちの1つ以上の優先順位記憶回路に優先順位ロード信号をアサートする、優先順位構成モジュールをさらに備える。この手法を介して、選択された優先順位記憶回路と関連付けられた優先順位を更新することができるか、または代替として、各クロスパーセルの優先順位記憶回路を並行して再プログラムすることができる。

40

【0042】

一実施形態において、各データ出力経路はn本のビット線を提供し、各構成記憶回路は、同じデータ出力経路と関連付けられたn個までのクロスパーセルに対して、それらのクロスパーセルの構成記憶回路が、該n本のビット線のうちの関連付けられた異なるビット線を有するように、それを介してルーティング値が構成記憶回路にプログラムされる該n

50

本のビット線のうちの1つと関連付けられる。各クロスパーセル内において、アービトレーション回路および優先順位記憶回路は、そのクロスパーセルの構成記憶回路に関連付けられた1本のビット線を除く $n - 1$ 本のビット線と関連付けられ、優先順位記憶回路は、該所定の優先順位スキームを実装するために、それらの $n - 1$ 本のビット線のうちのどれがアービトレーション回路により電圧を修正されるべきかを識別する。

【0043】

具体的な一実施形態において、優先順位記憶回路は、優先順位記憶回路と関連付けられた $n - 1$ 本のビット線毎に1つの、 $n - 1$ 個の優先順位記憶素子で形成され得る。優先順位記憶素子は、様々な形態をとり得るが、一実施形態においてはSRAMセルで形成される。

10

【0044】

クロスパー回路が $n \times n$ クロスパーセル未満である場合、前述したようなアービトレーションモードにおいて再利用されないワード線およびビット線が存在することが理解される。例えば、64ビットデータ入力経路およびデータ出力経路を備える 16×16 クロスパー回路を考慮すると、前述の手法を用いて、アービトレーション中にワード線およびビット線の4分の1のみが使用される。そのような場合、各クロスパーセルに複数の優先順位記憶回路を提供することにより、各クロスパーセルに複数の優先順位を記憶することができる。すると優先順位構成モジュールは、データ入力経路のワード線を使用して、各クロスパーセルに複数の優先順位記憶回路をプログラムすることができる。するとその後、予めロードされた様々な優先順位の間を、それらを再ロードする必要なくスイッチすることが可能であり、それにより、単一の周期で1つの優先順位スキームと別の優先順位スキームとの間をスイッチさせることが可能となる。例えば、上述の所与のシナリオにおいて、各クロスパーセルに4つの優先順位を記憶し、アービトレーション中にそれらのいずれかを選択的に使用することが可能である。また、優先順位割当動作モード中に4つすべての優先順位を1周期でロードするために、余分の入力線を使用することも可能である。

20

【0045】

優先順位構成モジュールは、様々な様式で動作可能である。例えば、一実施形態において、それは、ワード線を使用してクロスパーセルの優先順位記憶回路に優先順位を定期的書き込む有限状態機械(FSM)として機能することができる。具体的な一実施形態において、それはラウンドロビンスキームで優先順位を割り当てることができるが、その他の任意の好適な優先順位スキームを使用することができることが理解される。

30

【0046】

しかしながら、代替の実施形態において、クロスパー回路の全体的な効率を増加させることを目指すために、優先順位構成モジュールを使用して適応型優先順位プログラミングを実行することができる。具体的には、一実施形態において、複数のソース回路は、該複数のデータ入力経路の第1の端部に接続可能であり、優先順位構成モジュールは、該複数のデータ入力経路の第2の端部に接続可能であり、ソース回路は、各データ入力経路の複数のワード線を介してアサートされた伝送要求を発行するように配設され、優先順位構成モジュールは、各ソース回路により発行されたアサートされた伝送要求を監視するように、また、各ソース回路からのアサートされた伝送要求の数に依存して、優先順位割当動作モード中、優先順位構成モジュールにより出力される優先順位データを修正するように配設される。したがって、そのような実施形態において、優先順位構成モジュールは、特定のソース回路によりどのデータ出力経路が最も使用されるかを割り出すために、クロスパーに送信されたアサートされた伝送要求の数を監視することができる。次いでそれは、クロスパー回路の全体的な効率が増加するように、この情報を使用して優先順位をより知的に割り当てる。効率は、クロスパー回路が伝送動作モードにある動作時間(すなわち、伝送要求間のコンフリクトを検出し解消するのではなくデータを送信するためにビット線が使用される期間)の割合として定義することができる。

40

【0047】

一実施形態において、各クロスパーセルは、そのクロスパーセルの構成記憶回路と関連

50

付けられた1つのビット線と連結された解放回路をさらに備え、アサートされた解放要求の受信後に、チャンネル解放回路が、そのクロスバーセルの構成記憶回路における該ルーティング値として該第2の値を記憶させるレベルまで、その1つのビット線に対する電圧を修正させる。これは、データの伝送を終了した後にソース回路に特定のデータ出力経路を解放させ、それによってより低い優先順位のソース回路が後にそのデータ出力経路を獲得することを目指すのを可能とするための、単純で効果的な機構を提供する。

【0048】

具体的な一実施形態において、ビット線のそれぞれは第1の電圧レベルにプリチャージされ、該アサートされた解放要求の受信後に、解放回路は、そのクロスバーセルの構成記憶回路と関連付けられた1つのビット線に対する電圧をディスチャージする。さらに、各構成記憶回路は、関連付けられた解放回路へのアサートされた解放要求が存在する場合、ルーティング値を該第2の値に更新するために、解放回路により実行されるディスチャージ動作の後にその関連付けられたビット線に対する電圧を感知する感知増幅器有効ラッチを備え、それにより、関連付けられたデータ出力経路を解放する。したがって、そのような実施形態において、クロスバーセルのチャンネルを解放するために、関連付けられた感知増幅器有効ラッチにより感知されるビット線がディスチャージされ、後にサンプリングされる。

10

【0049】

一実施形態において、各データ入力経路はn本のワード線を提供し、同じデータ入力経路と関連付けられたn個までのクロスバーセルに対する開放要求は、その同じデータ入力経路のn本のワード線を介してクロスバー回路に入力される。さらに、各解放回路は、同じデータ入力経路と関連付けられたn個までのクロスバーセルに対して、それらのクロスバーセルの解放回路が、それを介して解放要求が受信される該n本のワード線のうちの関連付けられた異なるワード線を有するように、該n本のワード線のうちの1つと関連付けられる。したがって、そのような実施形態において、解放要求は、データ入力経路のワード線の再利用により効率的に発行され得る。

20

【0050】

具体的な一実施形態において、関連付けられたデータ入力経路のn本のワード線が解放要求または伝送要求を担持しているかどうかを識別するために、該アービトラクション動作モードにおいて解放要求が発行され、アービトラクション動作モードにおいて複数のソース回路により制御信号が発行される。したがって、そのような実施形態において、制御信号を使用して、ワード線を介してルーティングされた情報が解放要求または伝送要求を識別するかどうか、ひいてはアービトラクション回路または解放回路が有効化される必要があるかどうかを認めることができる。具体的な一実施形態において、制御信号は、実際に2つの別個の信号線の形態をとり、1つはチャンネル要求信号、もう1つはチャンネル解放信号であり、これらの2つの信号のうちの1つだけが任意の時点で設定され得る。一実施形態において、これらの2つの信号は行ごとに提供され、したがって任意の特定行内で、クロスバーセルは任意の時点で解放要求または伝送要求を処理している。いくつかの実現形態において、ある行のいくつかのクロスバーセルが開放要求を処理する一方でその他のクロスバーセルが伝送要求を処理することができるように、より細かい粒度の制御信号を提供することが可能である。

30

40

【0051】

一実施形態において、アービトラクション動作モード中にクロスバーセルがそのルーティング値を第1の値にプログラムされた場合、アサートされた伝送要求が承諾されたことを確認するために、関連付けられたソース回路に承諾信号がアサートされる。一実施形態において、ソース回路は、1度に1つの要求のみを送信でき、そのような実施形態においては、典型的には行ごとに1つだけの承諾信号が発行される。しかしながら、代替の実施形態において、ソース回路は、1度に複数の要求を送信可能となることができ、そのような実施形態においては、行ごとに複数の承諾信号を提供することができ、1つの承諾信号は各データ出力経路と関連付けられている。

50

【 0 0 5 2 】

伝送回路は、様々な様式で動作可能である。しかしながら、一実施形態において、伝送動作モードでクロスバーセルに関連付けられる各データ出力経路は、データ転送の前に第1の論理レベルにプリチャージされ、各クロスバーセルの伝送回路は、データ出力経路と第2の論理レベルとの間に直列に接続される、第1および第2のスイッチを備え、伝送動作モードにおいて、第1のスイッチは、関連付けた構成記憶回路内に記憶されているルーティング値に依存して開かれるか、または閉じられ、第2のスイッチは、データ入力経路上で入力されるデータに依存して開かれるか、または閉じられる。特定の一実施形態において、第1の論理レベルは、供給電圧レベル V_{dd} であり、第2の論理レベルは、接地である。したがって、このような配設によって、入力データ経路上のデータは、出力データ経路上のデータを直接的に駆動せず、代わりに、出力データ経路上のデータは、第1の論理レベルに留まるか、または第1および第2のスイッチの両方が閉じられる場合、第2の論理レベルの方へディスチャージされる。

10

【 0 0 5 3 】

このような伝送回路の配設によって、伝送回路は、クロスバー回路のサイズに関わらず、変化させる必要が無く、したがって、データ出力経路の長さ、さらには入力データ経路上で入力データを提供する駆動回路も、クロスバー回路のサイズが増加した時にサイズを変更する必要が無い。したがって、クロスバー回路のサイズが増加した場合、各クロスバーセルの回路を変更する必要が無く、代わりに、クロスバーセルの数を増加させるだけでよい。したがって、このような設計を使用することによって、クロスバー回路の遅延は、サイズとともに直線的に増大し、このようなクロスバー回路の設計を、(例えば、 128×128 または 256×256 入力/出力の)非常に大きいクロスバーに対して使用できるようにする。加えて、本設計は、非常に標準的であり、クロスバー回路を通じた全ての経路にわたる遅延は均一である。

20

【 0 0 5 4 】

一実施形態において、各該データ入力経路は、伝送動作モード中に、 n ビットの入力データ値を担持するための、 n 本のワード線を備え、各該データ出力経路は、該伝送動作モード中に、 n ビットのデータ値を担持するための、 n 本のビット線を備え、少なくとも第2のスイッチは、各ビット線ごとに複製される。したがって、伝送回路の設計は、いかなるクロスバー回路の著しい複雑さの増加も伴わずに、種々のサイズのデータ入力経路およびデータ出力経路に容易に適応することができる。

30

【 0 0 5 5 】

一実施形態では、伝送動作モードにおいて、データ出力経路のビット線は、ルーティング値が該第1の値であり、かつ対応するワード線上の入力データビットが該第1の論理レベルにある場合に、第2の論理レベルにプル (pull) される。したがって、構成記憶回路内に記憶されているルーティング値が、データ入力経路を、データ出力経路に連結すべきであり、かつデータ入力経路上のデータが論理1レベルにあることを示している場合、データ出力経路は、第2の論理レベルの方へディスチャージされる。

【 0 0 5 6 】

一実施形態において、各クロスバーセルは、ルーティング値に関わらず第1のスイッチをオフにし、一方で、関連付けたデータ出力経路を第1の論理レベルにプリチャージし、および第1のスイッチを、ルーティング値によって制御し、その後、関連付けたデータ出力経路を第1の論理レベルにプリチャージできるようにするための、条件付きディスチャージ回路をさらに備える。したがって、このような条件付きディスチャージ回路は、入力からデータ出力経路のビット線を分離し、一方で、それらは、プリチャージされ、それによって、プリチャージ動作の電力消費を低減する。また、このような配設によって、データ入力経路は、その時の入力からデータ出力経路を分離させる条件付きディスチャージ回路により、プリチャージ動作と同時に駆動することができ、それによって、動作の速度を増加させることを可能にする。

40

【 0 0 5 7 】

50

さらに、上述した伝送回路の配設を使用することによって、全てのデータ出力経路のビット線を、第1の論理レベルにプリチャージし、次いで、関連付けたデータ入力経路のワード線への入力値およびルーティング値に依存して、第1の論理レベルのままにするか、または第2の論理レベルに遷移させることに留意されたい。その結果、2本の隣接するビット線への電圧が反対方向に移動するといったいかなる状況も生じず、それによって、容量結合効果が低減され、それによって、動作の速度が向上する。

【0058】

データ出力経路が、データ転送の前にプリチャージされ、次いで、ルーティング値およびデータ入力値に依存して、選択的にディスチャージされる実施形態では、感知増幅回路の使用を通じてクロスバー回路の電力消費を低減することができる。より具体的には、一実施形態において、クロスバー回路は、伝送動作モード中に、データ出力経路のビット線上のデータ出力を検出し、それによって、データ出力経路のビット線の電圧が第2の論理レベルに到達する前に、第2の論理レベルへの遷移を検出できるようにする、感知増幅回路をさらに備える。遷移の検出は、データ出力経路のビット線が第2の論理レベルに到達する前に生じるので、データ出力経路のビット線を第1の論理レベルへプリチャージするのに必要な電力が、大幅に低減される。

10

【0059】

このようなデータ出力経路のプリチャージが生じる実施形態では、よって、クロスバー回路に提供される前に、入力データを好適に符合化することによって、さらなる省電力化を得ることができる。より具体的には、一実施形態において、クロスバー回路は、該複数のソース回路のそれぞれと、該複数のデータ入力経路との間の符号化回路と、該複数のデータ出力経路のそれぞれと、該複数の宛先回路との間の復号回路と、をさらに備え、符号化回路は、各ソース回路によって提供される入力データを符号化した形態に符号化するように、符号化動作を適用し、該動作は、元の入力データがクロスバー回路を通過した場合、データ出力経路を、第2の論理レベルにプルするのに必要になり得る回数と比較した時に、出力経路を、第1の論理レベルにプリチャージした後に第2の論理レベルにプルするのに必要な回数を低減し、復号回路は、データ出力経路上で出力される符号化したデータから、ソース回路によって提供される元の入力データを同定するように、対応する復号動作を適用する。

20

【0060】

特定の一実施形態において、符合化動作は、入力データから生成される符号化した形式が、入力データが変化する時に、論理1値だけであり、それによって、データ出力経路が、入力に変化した時にだけディスチャージされるようにする。復号回路は、次いで、データ出力経路上で出力される符号化したデータから、元の入力データを再形成する。

30

【0061】

本発明の実施形態のクロスバー回路の標準的な設計、および各クロスバーセルの伝送回路が動作する様態のため、複数のソース回路を、複数のデータ入力経路のいずれかの端部に接続でき、それによって、クロスバー回路およびソース回路が提供される装置のレイアウトを設計する時の柔軟性が大幅に向上する。同様に、複数の宛先回路は、複数のデータ出力経路のいずれかの端部に接続できる。

40

【0062】

一実施形態において、ワード線ドライバのシーケンスを使用して、データ入力経路に接続された様々なクロスバーセルに、それらのデータ入力経路に沿ってデータ入力値を伝播することができる。マルチキャストがサポートされず、したがって各ソースが1度に1つの出力経路へのアクセスしか要求しない単一要求の実施形態において、それが必要とされない状況において、データ入力経路の全体に沿ってデータを伝播することに大量の電力が消費され得る。一実施形態において、この電力消費は、部分的に有効化されたネットワーク配置の提供により軽減される。具体的には、一実施形態において、各データ入力経路は、ワード線ドライバにより隔てられた複数の入力経路部分からなり、各ワード線ドライバは、関連付けられたデータ入力経路に沿って提供されるが、そのワード線ドライバ

50

よりもソース回路から遠くにあるクロスバーセルの構成記憶回路に記憶されたルーティング値に基づき有効化される。結果として、各ワード線ドライバは、関連付けられたデータ入力経路に沿って提供されるが、そのワード線ドライバよりもソース回路から遠くにある少なくとも1つのクロスバーセルの構成記憶回路が、該第1の値に設定されたルーティング値を有する場合にのみ有効化される。そのような手法は、著しい省電力化をもたらすことができる。

【0063】

本発明の実施形態のクロスバー回路は、様々なシステムで利用されてもよい。しかしながら、本発明の第2の態様によれば、データ処理装置が提供され、データ値を記憶するための複数のメモリデバイスと、該複数のメモリデバイス内に記憶されている複数のデータ値に対して、並列してデータ処理動作を実行するための複数のプロセッサと、該複数のメモリデバイスのうちのいずれかから、該複数のプロセッサのうちのいずれかに、データ値をルーティングするための、本発明の第1の態様によるクロスバー回路と、を備える。

10

【0064】

本発明の実施形態のクロスバー回路は、メモリデバイスのうちのいずれかから、プロセッサのうちのいずれかにデータ値をルーティングするための、とりわけ単純で、拡張性があり、また電力効率の良い機構を提供する。

【0065】

第3の態様から見ると、本発明は、複数のソース手段と、複数の宛先手段とを相互接続し、よって、該複数のソース手段のうちのいずれかからクロスバー回路へのデータ入力を、該複数の宛先手段のうちのいずれかに出力することができるクロスバー回路を提供し、該クロスバー回路は、該クロスバー回路を通過する複数のデータ入力経路手段であって、各データ入力経路手段が、該複数のソース手段のうちの1つに接続し、かつ複数のワード線手段を提供するための、データ入力経路手段と、複数のデータ入力経路手段を横断する該クロスバー回路を通過する複数のデータ出力経路手段であって、各データ出力経路手段が、該複数の宛先手段のうちの1つに接続し、かつ複数のビット線手段を提供するための、データ出力経路手段と、該データ入力経路手段のうちの1つと、該データ出力経路手段のうちの1つとの間の各交差点に関連付けられるクロスバーセル手段であって、各クロスバーセル手段は、ルーティング値を記憶するためにプログラムできる構成記憶手段であって、ルーティング値は、データ入力経路手段のワード線手段に沿った関連付けた交差点へのデータ入力が、関連付けた交差点において、データ出力経路手段のビット線手段上に出力されることを、第1の値が示すようにプログラムされ、ルーティング値は、データ入力経路手段のワード線手段に沿った関連付けた交差点へのデータ入力が、関連付けた交差点において、データ出力経路手段のビット線手段上に出力されないことを、第2の値が示すようにプログラムされる、構成記憶手段と、データ入力経路手段のワード線手段に沿ってデータ入力を検出し、関連付けた交差点において、そのデータの指示をデータ出力経路手段のビット線手段上に出力するための、伝送動作モードで、該第1の値を有するルーティング値に応答する、伝送手段と、関連付けた交差点のデータ入力経路手段に接続されたソース手段から、クロスバーセル手段によって受信された伝送要求に基づいて、アービトレーション動作モードにおいて動作するためのアービトレーション手段であって、該ソース手段が関連付けた交差点において、データ入力経路手段からデータ出力経路手段へデータをルーティングしたいことを示すように、伝送要求がアサートされる場合、アービトレーション手段は、同じデータ出力経路手段に関連付けられた他のクロスバーセル手段のアービトレーション手段と組み合わせて動作し、データ出力経路手段のビット線手段を再利用して、該同じデータ出力経路手段の複数のアサートされた伝送要求の存在を検出し、このような複数のアサートされた伝送要求の場合、所定の優先順位スキームを実装して、該同じデータ出力経路手段に関連付けられた1つだけのクロスバーセル手段の構成記憶手段を、そのルーティング値が該第1の値にプログラムされるようにし、それによって、該所定の優先順位スキームに従い該複数のアサートされた伝送要求間のコンフリクトを解消する、アービトレーション手段と、を備える。

20

30

40

50

【 0 0 6 6 】

第4の態様から見ると、本発明は、複数のソース回路と、複数の宛先回路とを相互接続し、よって、該複数のソース回路のいずれかからクロスバー回路へのデータ入力を、該複数の宛先回路のうちのいずれかに出力することができ、クロスバー回路は、該クロスバー回路を通過する複数のデータ入力経路を有し、各データ入力経路は、該複数のソース回路のうちの1つに接続でき、かつ複数のワード線を提供し、複数のデータ出力経路は、複数のデータ入力経路を横断する該クロスバー回路を通過し、各データ出力経路は、該複数の宛先回路のうちの1つに接続でき、かつ複数のビット線を提供する、クロスバー回路を動作させる方法を提供し、該データ入力経路のうちの1つと、該データ出力経路のうちの1つとの間の各交差点を関連付けて、クロスバーセルを採用するステップと、各クロスバーセルでルーティング値をプログラミングするステップであって、ルーティング値は、データ入力経路のワード線に沿った関連付けた交差点へのデータ入力が、関連付けた交差点において、データ出力経路のビット線上に出力されることを、第1の値が示すようにプログラムされ、ルーティング値は、データ入力経路のワード線に沿った関連付けた交差点へのデータ入力が、関連付けた交差点において、データ出力経路のビット線上に出力されないことを、第2の値が示すようにプログラムされる、ステップと、伝送動作モードにおいて、データ入力経路のワード線に沿ってデータ入力を検出し、関連付けた交差点において、そのデータの指示をデータ出力経路のビット線上に出力するよう、クロスバーセルを、該第1の値を有するルーティング値に応答させるステップと、アービトレーション動作モードにおいて、クロスバーセル内のアービトレーション回路を、関連付けた交差点のデータ入力経路に接続されたソース回路からクロスバーセルによって受信された伝送要求に基づいて動作させるステップであって、該ソース回路が関連付けた交差点において、データ入力経路からデータ出力経路へデータをルーティングしたいことを示すように、伝送要求がアサートされる場合、アービトレーション回路は、同じデータ出力経路に関連付けられた他のクロスバーセルのアービトレーション回路と組み合わせて動作し、データ出力経路のビット線を再利用して、該同じデータ出力経路の複数のアサートされた伝送要求の存在を検出し、このような複数のアサートされた伝送要求の場合、所定の優先順位スキームを実装して、該同じデータ出力経路に関連付けられた1つだけのクロスバーセルの構成記憶回路を、そのルーティング値が該第1の値にプログラムされるようにし、それによって、該所定の優先順位スキームに従い該複数のアサートされた伝送要求間のコンフリクトを解消する、ステップと、を含む。

10

20

30

【 図面の簡単な説明 】

【 0 0 6 7 】

【 図 1 】一実施形態による、クロスバー回路のブロック図である。

【 図 2 】一実施形態に従って、どのようにコリジョンが検出および解消され得るかを示す図である。

【 図 3 】各クロスバーセルに従って提供される優先順位記憶回路が、一実施形態においてどのように用いられ得るかを示す図である。

【 図 4 】特定のデータ出力経路（本明細書においてチャンネルとも称される）を解放するためのクロスバーセルを有効にするために、一実施形態においてどのように解放回路が使用され得るかを示す図である。

40

【 図 5 A 】列中の多数のクロスバーセルに対して各クロスバーセル内に提供される回路を示す図である。

【 図 5 B 】列中の多数のクロスバーセルに対して各クロスバーセル内に提供される回路を示す図である。

【 図 6 A 】行中の多数のクロスバーセルに対して各クロスバーセル内に提供される回路を示す図である。

【 図 6 B 】行中の多数のクロスバーセルに対して各クロスバーセル内に提供される回路を示す図である。

【 図 7 】一実施形態に従って、各クロスバーセル内に提供され得る伝送回路およびアービ

50

トレーション回路をより詳細に示す図である。

【図 8】一実施形態に従って、各クロスパーセル内に提供され得る伝送回路および解放回路をより詳細に示す図である。

【図 9】一実施形態に従って、パケットスイッチングクロスパーの実現形態を概略的に示す図である。

【図 10 A】一実施形態に従って、クロスパー回路の各クロスパーセル内に提供される構成要素をより詳細に示す図である。

【図 10 B】一実施形態に従って、クロスパー回路の各クロスパーセル内に提供される構成要素をより詳細に示す図である。

【図 10 C】一実施形態に従って、クロスパー回路の各クロスパーセル内に提供される構成要素をより詳細に示す図である。

【図 10 D】一実施形態に従って、クロスパー回路の各クロスパーセル内に提供される構成要素をより詳細に示す図である。

【図 11 A】代替の実施形態に従って、クロスパー回路の各クロスパーセル内に提供される構成要素をより詳細に示す図である。

【図 11 B】代替の実施形態に従って、クロスパー回路の各クロスパーセル内に提供される構成要素をより詳細に示す図である。

【図 11 C】代替の実施形態に従って、クロスパー回路の各クロスパーセル内に提供される構成要素をより詳細に示す図である。

【図 11 D】代替の実施形態に従って、クロスパー回路の各クロスパーセル内に提供される構成要素をより詳細に示す図である。

【図 12】一実施形態に従って、各クロスパーセル内に提供される構成記憶素子の配設をより詳細に示す図である。

【図 13 A】一実施形態に従って、ディスチャージおよび感知有効 (SE) 信号を生成するためのパルス発生器の構成を示す図である。

【図 13 B】一実施形態に従って、ディスチャージおよび感知有効 (SE) 信号を生成するためのパルス発生器の構成を示す図である。

【図 13 C】一実施形態に従って、ディスチャージおよび感知有効 (SE) 信号を生成するためのパルス発生器の構成を示す図である。

【図 14】一実施形態に従って、チャンネルフリー信号がどのように生成されるかを示す図である。

【図 15 A】一実施形態に従って、図 1 の優先順位構成モジュールの動作を示す図である。

【図 15 B】代替の実施形態に従って、図 1 の優先順位構成モジュールの動作を示す図である。

【図 16 A】クロスパー回路内の電力消費を低減するために、クロスパー回路への入力の前に入力データを符号化し、クロスパー回路からの出力データを復号する、一実施形態で使用され得る、符号化および復号回路を示す図である。

【図 16 B】クロスパー回路内の電力消費を低減するために、クロスパー回路への入力の前に入力データを符号化し、クロスパー回路からの出力データを復号する、一実施形態で使用され得る、符号化および復号回路を示す図である。

【図 16 C】クロスパー回路内の電力消費を低減するために、クロスパー回路への入力の前に入力データを符号化し、クロスパー回路からの出力データを復号する、一実施形態で使用され得る、符号化および復号回路を示す図である。

【図 16 D】クロスパー回路内の電力消費を低減するために、クロスパー回路への入力の前に入力データを符号化し、クロスパー回路からの出力データを復号する、一実施形態で使用され得る、符号化および復号回路を示す図である。

【図 16 E】クロスパー回路内の電力消費を低減するために、クロスパー回路への入力の前に入力データを符号化し、クロスパー回路からの出力データを復号する、一実施形態で使用され得る、符号化および復号回路を示す図である。

10

20

30

40

50

【図17】一実施形態に従って、狭いデータ入力経路およびデータ出力経路を有するクロスバー回路が動作するためにどのように配設され得るかを示す図である。

【図18A】代替の実施形態に従って、狭いデータ入力経路およびデータ出力経路を有するクロスバー回路が動作するためにどのように配設され得るかを示す図である。

【図18B】代替の実施形態に従って、狭いデータ入力経路およびデータ出力経路を有するクロスバー回路が動作するためにどのように配設され得るかを示す図である。

【図19】一実施形態に従って、広いデータ入力経路およびデータ出力経路を有するクロスバー回路がどのように動作し得るかを示す図である。

【図20】一実施形態に従って、部分的に有効化された構成において、クロスバー回路がどのように使用され得るかを示す図である。

10

【発明を実施するための形態】

【0068】

以下、ほんの一例として、添付図面に示されているその実施形態を参照して、本発明をさらに説明する。

【0069】

図1は、一実施形態に従ったクロスバー回路と、それに接続された複数のソース回路および宛先回路とを備える、提案されるスイッチングファブリックの上位図を示す。クロスバー回路は、クロスバー回路を通じて第1の方向で動作する複数のデータ入力経路12と、第1の方向を横断する第2の方向で動作する複数のデータ出力経路50と、を含む。図1の実施形態において、これらの経路は両者とも、マルチビットバスによって形成されており、特に、各データ入力経路は、複数のワード線を備え、各データ出力経路は、複数のビット線を備える。

20

【0070】

各データ入力経路とデータ出力経路との間の交差点には、クロスバーセル20が提供され、関連付けたデータ入力経路で受信されたデータを、関連付けたデータ出力経路上に選択的にルーティングするために使用される。各クロスバーセルは、各クロスバーセルのオン/オフ状態を記憶するための、構成記憶素子を含み、よって、クロスバーセルがオン状態にある場合は、データ入力経路上で受信した入力データをデータ出力経路にルーティングし、セルがオフ状態にある場合は、入力データは、関連付けたデータ出力経路上にルーティングされずに、単にセルを通過するだけである。

30

【0071】

各データ入力経路12は、入力データ10を受信し得る関連付けたソース回路30に連結され、各データ出力経路50は、そのデータ出力経路上に提供される出力データ70を受信するように配設される、関連付けた宛先回路40に連結される。

【0072】

示される実施形態において、各データ出力経路50のビット線は、プリチャージモジュール55を用いてプリチャージされ、該ビット線は、ソース回路のうちの1つから、そのソース回路のデータをデータ出力経路に接続するクロスバーセルへのデータ入力に依存して、選択的にディスチャージされる。出力データ70を生成するために、このビット線の選択的なディスチャージは感知増幅回路65により検出される。

40

【0073】

クロスバーが伝送動作モードにある時、データは上記の様式でクロスバー回路を通してルーティングされるが、クロスバー回路は必ずしも伝送動作モードにあるとは限らない。具体的には、様々なソース回路30によって発行される伝送要求間に何らかのコンフリクトを検出および解消する間、クロスバー回路はアービトレーション動作モードを有する。具体的には、スイッチングファブリックにおける同じ宛先回路への複数の要求はコリジョンと称され、ソースおよび宛先の数が増加するにつれて、コリジョンはより頻繁になる。後により詳細に論じられるように、各クロスバーセル20は、アービトレーション動作モードにおいて、複数のアサートされた伝送要求の存在を検出するためにデータ出力経路のビット線を再利用して、他のクロスバーセルのアービトレーション回路と組み合わせて動

50

作し、またそのような複数のアサートされた伝送要求の場合には、それらのコンフリクトを解消するために所定の優先順位スキームを実装する、アービトレーション回路を含む。提案される手法は、そのようなコリジョンを検出および解消するための迅速、低電力、かつ高度に拡張可能なソリューションを提供する。

【0074】

所定の優先順位スキームは固定されていてもよいが、一実施形態において、各クロスバーセルは、優先順位割当動作モードにおいて優先順位構成モジュール60によってプログラムすることができる、優先順位記憶回路を含む。優先順位割当動作モードでは、クロスバーセル内の様々な優先順位記憶回路を再プログラムするために、データ入力経路のワード線が用いられる。したがって、ワード線を介して、所定の優先順位スキームを再プログラムするために優先順位割当動作モードを用いることができ、クロスバー回路を介して、様々なソース回路から様々な宛先回路へのより公平なアクセスの分布が可能になる。

10

【0075】

図1では、ソース回路がクロスバーの左側に示され、宛先回路がクロスバーの下方に示されているが、ソース回路はクロスバーのいずれの側に提供されてもよく、また同様に、宛先回路はデータ出力経路のいずれの端部に提供されてもよい。したがって、本発明の実施形態のクロスバー回路は、ソースを両水平端部に、また、宛先を両垂直端部に配置するといった柔軟性を提供することを認識されるであろう。これは、ルーティングの輻輳を低減することによって、設計のフロアプランニングを簡略化する。

【0076】

20

図2は、アービトレーション動作モード中のコンフリクトを検出および解消するために、どのようにデータ出力経路のビット線を再利用することができるかを示す。本例では、伝送要求0~15を発行し得る、クロスバー回路に接続された16個のソース回路がある。いずれの特定のデータ出力経路50に対しても、各ソース回路は、感知増幅およびラッチの形態で、その構成記憶回路内に組み込まれるクロスバーセル20と関連付けられる。したがって、ソース0は感知増幅およびラッチ100を含むクロスバーセルと関連付けられ、ソース1は感知増幅およびラッチ105を含むクロスバーセルと関連付けられ、ソース2は感知増幅およびラッチ110を含むクロスバーセルと関連付けられ、残りのソース回路も同様であり、ソース回路15は感知増幅およびラッチ115を含むクロスバーセルと関連付けられる。

30

【0077】

アービトレーション動作モードでは、プリチャージモジュール55が各データ出力経路50のすべてのビット線をプリチャージし、その後、それらのビット線は、アサートされた伝送要求に依存して、また所定の優先順位スキームに従って、選択的にディスチャージされる。本例では、各データ出力経路50が16本のビット線を備え、感知増幅およびラッチ回路の各々が、それらのビット線のうちの異なる1本と関連付けられていることが分かる。ビット線のプリチャージ後、様々なソース回路によってアサートされた伝送要求が評価されるが、本実施形態においてアサートされる伝送要求は論理1値を取っている。

【0078】

アサートされた伝送要求は、最初に、様々なトランジスタ107、112、114、117、119、121、123を介して、ビット線の選択的なディスチャージを引き起こす。見て分かるように、本例の実施形態ではソース15が最も高い優先順位を有するため、ソース15が伝送要求をアサートする場合、それと関連付けられた感知増幅およびラッチ回路115のビット線を除くすべてのビット線のディスチャージが引き起こされる。ソース14が次に高い優先順位を有し、ソース13がその次に高い優先順位を有する等、ソース0が最も低い優先順位を有する。具体的には、見て分かるように、ソース0からの伝送要求(すなわち要求0)がアサートされた場合、いずれのビット線にもディスチャージが引き起こされることはない。選択的なディスチャージ動作に続いて、アサートされた伝送要求を受信した任意の感知増幅およびラッチ回路は、関連付けられたクロスバーセルが、伝送動作モードにおいて、その入力をその出力に連結するかどうかを決定するルーティ

40

50

ング値を記憶するために、関連付けられたビット線のサンプリングを行う。具体的には、ルーティング値が第1の値（一実施形態において、これは論理1値である）を有する場合にのみ、関連付けられたクロスバーセルは伝送動作モード中にその入力データをその出力に接続し、また上記スキームによって、アービトレーション動作モードの終わりには、1つのクロスバーセルのみが感知増幅およびラッチ回路を論理1値に設定することが理解されるであろう。

【0079】

したがって、例として、ソース1およびソース2の両方がアサートされた伝送要求を発行した場合、アサートされた要求1およびアサートされた要求2の両方が、トランジスタ107、112を介して第1のビット線のディスチャージを引き起こすことが分かる。結果として、ソース0が要求をアサートした場合でも、関連付けられた感知増幅およびラッチ回路100がアービトレーション動作モードの終わりに論理0値をラッチするため、その要求は承諾されない。さらには、アサートされた要求2が、トランジスタ114を介して第2のビット線のディスチャージを引き起こすので、アービトレーション動作モードの終わりに、ソース1と関連付けられた感知増幅およびラッチ回路105が論理0値をラッチする。したがって、たとえソース1が伝送要求をアサートしたとしても、ソース2も既にアサートされた伝送要求を発行しているため、その伝送要求は承諾されない。その周期では他のソース回路は伝送要求をアサートしていないと仮定すると、ソース回路2と関連付けられた感知増幅およびラッチ回路110が、アービトレーション動作モードの終わりに論理1値をラッチするため、ソース2からの伝送要求が承諾されることが分かるであろう。

10

20

【0080】

上記の説明から、受信するアサートされた伝送要求が、より低い優先順位のソースからの他のアサートされた伝送要求の抑制をもたらすことが分かる。次いで、いずれかのアサートされた伝送要求に対する感知増幅およびラッチ回路は、要求が承諾されるか否かを決定するために、それらに関連付けられたビット線のサンプリングを行う。この技法によって、単一周期内でコリジョンを検出および解消することができる。さらに、アービトレーション動作モード中にコンフリクトを検出および解消するために用いられたのと同じビット線が、それに続く伝送動作モードにおいてデータの転送のために用いられることが分かる。

30

【0081】

アサートされた伝送要求は、多数の異なる様式でクロスバー回路に入力することができるが、一実施形態において、アサートされた伝送要求を入力するために、データ入力経路のワード線が用いられる。したがって、各データ入力経路が16本のワード線を有する実施形態を考えた場合、いずれかの特定のソース回路が、それらのワード線を介して、最大16個までのデータ出力経路のうち、伝送要求をアサートしたいものを特定することが可能であり、アービトレーション動作モードにおいて、個々のワード線の各々は異なるデータ出力経路と関連付けられることが分かる。

【0082】

図2には明確に示されていないが、典型的には、様々なトランジスタ107、112、114、117、119、121、123と関連付けられたいくつかの分離回路が存在し、プリチャージ動作モード中に、それらのゲートに提供される様々な要求入力の値にかかわらず、これらのトランジスタが確実にオフになるようにする。

40

【0083】

図2ではアービトレーションに用いられる優先順位が固定されているが、一実施形態において、図3に概略的に示されるように、優先順位記憶回路の提供を通じて、優先順位をクロスバーセルに記憶させることができる。具体的には、図3に示すように、各クロスバーセルは、そのクロスバーセルの感知増幅およびラッチ回路に接続されたビット線を除く各ビット線と関連付けられたトランジスタ135のシーケンスを含む。次いで、優先順位記憶素子130が、これらの優先順位記憶素子にプログラムされた値に基づいて、トラン

50

ジスタ 1 3 5 のゲートに値を提供するために用いられる。図 3 に概略的に示されるように、図 2 の固定された配設に示されるような同じ優先順位スキームが所望される場合は、濃い色で示した優先順位記憶素子 1 3 0 のすべてが論理 1 値にプログラムされ、薄い色で示した優先順位記憶素子 1 3 0 のすべてが論理 0 値にプログラムされる。したがって、論理 1 値を記憶する優先順位記憶素子に取り付けられたトランジスタはビット線をディスチャージするが、論理 0 値を記憶する優先順位記憶素子に取り付けられたトランジスタはディスチャージしない。

【 0 0 8 4 】

図 1 の優先順位構成モジュール 6 0 は、優先順位割当動作モードにおいて個々の優先順位記憶素子 1 3 0 をプログラムするために用いることができる。このモードでは、新しい優先順位スキームを割り当てるためにワード線が用いられ、具体的には、各クロスパーセルの優先順位記憶回路を形成する 1 5 個の優先順位記憶素子をプログラムするために、1 6 本のワード線のうちの 1 5 本を用いることができる。

10

【 0 0 8 5 】

図 3 に概略的に示される技法を用いて、データ出力経路あたり 1 6 の階乗 (1 6 !) 個すべての順列は、各データ出力経路 (本明細書において、各データ出力経路はチャンネルとも称される) と関連付けられた優先順位記憶回路にプログラムすることができる。

【 0 0 8 6 】

図 3 に示した優先順位記憶素子 1 3 0 の他に、プリチャージモジュール 5 5 によって実行されるプリチャージ動作中に、トランジスタ 1 3 5 が確実にオフになるように、典型的には、各トランジスタ 1 3 5 に対して関連付けられた分離回路がいくつか存在する。

20

【 0 0 8 7 】

一実施形態において、不要となった場合にチャンネルを解放するための機構を提供するために、各クロスパーセルに解放回路が提供される。具体的には、いったんソース回路 3 0 がチャンネルへのアクセスを承諾されると、そのソース回路のみが、それに続く異なるソース回路への割付のためにチャンネルを解放することができ、このチャンネルの解放は、現在そのチャンネルへのアクセスが承諾されているソース回路からのアサートされた解放要求に回答して、解放回路によって実行される。あるチャンネルが特定のソース回路に対して承諾されても、他のソース回路がそのチャンネルへのアクセスを得ることはできず、他のソース回路によってそのチャンネルに関して発行されたいずれかのアサートされた伝送要求も、チャンネルがソース回路に割り付けられている間は有効化されない。

30

【 0 0 8 8 】

図 4 は、一実施形態に従って加えられた解放回路 1 4 0 を示す。優先順位記憶素子と同様の様式において、解放回路 1 4 0 は関連付けられたトランジスタ 1 4 5 を駆動するために用いられ、この場合トランジスタは、ルーティング値を決定するために同じクロスパーセルの感知増幅およびラッチ回路によってサンプリングされるビット線に連結されている。したがって、ソース回路がチャンネルを解放したいと望む場合には解放回路 1 4 0 に解放要求を送信し、それによってトランジスタ 1 4 5 に論理 1 値が出力され、関連付けられたビット線がディスチャージされる。その後、関連付けられた感知増幅およびラッチがビット線のサンプリングを行い、それによってそこに論理 0 値を記憶し、ひいてはチャンネルを解放する。再び、プリチャージモジュール 5 5 によって実行されるプリチャージ動作中に、トランジスタ 1 4 5 が確実にオフになるように、関連付けられたいくつかの分離回路が各トランジスタ 1 4 5 に提供される。

40

【 0 0 8 9 】

一実施形態では、関連付けられたデータ入力経路のワード線が解放要求または伝送要求を担持しているかどうかを判定するために、解放要求はアービトレーション動作モードにおいて発行され、制御信号はアービトレーション動作モードにあるソース回路によって発行される。したがって、今度はアサートされた解放要求を関連付けられた解放回路 1 4 0 へと担持するために、再度データ入力経路のワード線を再利用することが可能であり、それによって、以前に特定のソース回路に対して承諾されたチャンネルを解放するために特に

50

効率的な機構が提供される。

【 0 0 9 0 】

図 5 A , B は各クロスパーセル内に提供される主な構成要素を示し、具体的には行 0、1 および 2 の列 0 に提供されるクロスパーセルを示す。したがって、クロスパーセル 2 0 0 は行 0 と列 0 の間の交差点に提供され、クロスパーセル 2 3 0 は行 1 と列 0 の間の交差点に提供され、クロスパーセル 2 6 0 は行 2 と列 0 の間の交差点に提供される。

【 0 0 9 1 】

第 1 のクロスパーセル 2 0 0 を考えると、これは構成記憶回路 2 0 5、伝送回路 2 1 0、チャンネル解放回路 2 1 5、アービトレーション回路 2 2 0 および優先順位記憶回路 2 2 5 から構成されている。アービトレーション動作モードでは、伝送要求がデータ入力経路（図 5 A , B では入力バスとも称される）を介してアサートされるため、ソース 0 が列 0 に対して伝送要求をアサートすることを望む場合は、入力バスのビット 0 を設定することでそれを行う。見て分かるように、ビット 0 の値は AND ゲート 2 0 7 に入力され、他の入力は要求チャンネル制御信号によって駆動される。アービトレーション動作モード中に入力バスを介していずれかのアサートされた伝送要求を発行する場合、ソース回路は要求チャンネル制御信号をアサートする。つまり、ソース 0 がチャンネル 0 に対してアサートされた伝送要求を発行している場合、AND ゲート 2 0 7 に対する両方の入力が論理 1 値となるため、アービトレーション回路 2 2 0 および構成記憶回路 2 0 5 の両方が有効化される（構成記憶回路は OR ゲート 2 0 8 を介して有効化される）。

【 0 0 9 2 】

見て分かるように、構成記憶回路 2 0 5 はチャンネル 0 のためのデータ出力経路（図 5 A , B では出力バスとも称される）のビット 0 に接続されており、アービトレーション動作モードの終わりに、そこに記憶されるルーティング値を決定するために出力バスのビット 0 の値をサンプリングする。しかしながら、その時点までは、アービトレーション回路 2 2 0 は、出力バスのビット線を選択的にディスチャージするために、アサートされた伝送要求を受信した列と関連付けられた他のいずれかのクロスパーセルのアービトレーション回路 2 5 0、2 8 0 等と関連して動作する。具体的には、図 5 A , B に見られるように、クロスパーセル 2 0 0 のアービトレーション回路 2 2 0 は、ビット線 0 を除く出力バスの他のすべてのビット線に接続されており、優先順位記憶回路 2 2 5 内にプログラムされた所定の優先順位スキームに依存して、それらのビット線を選択的にディスチャージする。

【 0 0 9 3 】

上記のように、アービトレーション回路 2 2 0 の選択的なディスチャージ動作後、構成記憶回路 2 0 5 はその関連する出力ビット線の値をサンプリングし、具体的には、そのビット線がまだプリチャージ電圧レベルにある場合は、ソース 0 がチャンネル 0 へのアクセスを承諾されていることを示すために、構成記憶回路 2 0 5 内の論理 1 ルーティング値を記憶する。列中のクロスパーセルのアービトレーション回路によって実装された所定の優先順位スキームの性質によって、アービトレーション動作モードの終わりに、いずれかの特定の列に対して 1 個のクロスパーセルのみが、その構成記憶回路を論理 1 値に設定するので、1 個のソース回路のみが、任意の時点における特定の宛先回路へのアクセスを承諾され得る。

【 0 0 9 4 】

アービトレーション動作モード後、スイッチングファブリックは伝送動作モードに入り、その間、特定のチャンネルへのアクセスを承諾された各ソース回路は、入力バスを介してその入力データを提供する。したがって、ソース 0 が列 0 へのアクセスを承諾されたと仮定すると、入力バスを介して伝送回路 2 1 0 へとそのデータが発行され、構成記憶回路 2 0 5 内に論理 1 値が記憶されると、チャンネル 0 のための出力バスへとその入力を接続する。

【 0 0 9 5 】

上で論じたように、アービトレーション動作モードおよび伝送動作モードの他に、優先順位割当動作モードも提供される。優先順位割当動作モードは頻繁に使用されることを意

10

20

30

40

50

図するものではないが、各クロスパーセルの優先順位記憶回路225、255、285内に符合化されたものとして優先順位スキームの修正が可能となり、経時的にアービトレーションに公平性を持たせることができる。優先順位割当動作モードでは、入力バスを介して優先順位データを提供することが可能であり、入力バス上の値をサンプリングさせるために、ロード優先順位信号が各クロスパーセルの優先順位記憶回路に発行される。図3および図4についての上記考察からも明白であるように、16ビットのデータ入力経路およびデータ出力経路を持つクロスパーセルの16×16アレイを考えた場合、各クロスパーセルの優先順位記憶回路は、実際には15個のワード線を用いてプログラムすることが可能である15個の優先順位記憶素子から構成され、図5A、Bの例では、この目的のためにワード線1～15が用いられると推定される。

10

【0096】

上記したとおり、アービトレーション動作モードでは、チャンネルが既に特定のソースに承諾されている場合、それらを解放することも可能である。したがって、例として、ソース0が以前にチャンネル0へのアクセスを承諾されており、構成記憶回路205がそこに記憶された論理1ルーティング値を有する場合は、アービトレーション動作モードにおいて、チャンネル0に対してアサートされた解放要求を特定するために、入力バスのビット0上の論理1値とともに、ソース0はANDゲート217への解放チャンネル制御信号を発行することができる。これによって論理1値がチャンネル解放回路215に入力され、構成記憶回路がORゲート208を介して有効化される。この論理1値に応答して、チャンネル解放回路は出力バスのビット0をディスチャージし、次いで構成記憶回路205が出力バスを再度サンプリングすることにより、論理0値が構成記憶回路205内に記憶される。

20

【0097】

一実施形態において、要求チャンネル制御信号および解放チャンネル制御信号の両方を2ビット信号に符号化することができ、その2ビット信号を用いて、アービトレーション動作モード中に、関連するソース回路がいずれかのアサートされた伝送要求を発行しているか、アサートされた解放チャンネル要求を発行しているか、またはいずれのアサートされた要求も発行していないか、を特定する。

【0098】

クロスパーセル230および260はクロスパーセル200とまったく同じ様式で構成されているため、クロスパーセル230の素子235、237、238、240、245、247、250および255、ならびにクロスパーセル260の素子265、267、268、270、275、277、280および285は、それぞれクロスパーセル200の素子205、207、208、210、215、217、220および225に対応している。しかしながら、図5A、Bから明らかであるように、これらの様々な素子が様々なワード線およびビット線に接続されている様式は若干異なる。クロスパーセル200、230、260の各々が列0と関連付けられ、アサートされた伝送要求およびアサートされた解放要求は常に関連付けられた入力バスのビット0に提供されるため、この目的のために、3つすべてのクロスパーセルがビット0（ワード線0）に接続されている。同様に、3つすべてのセルにおいて、優先順位割当動作モードで提供される優先順位情報は、入力バスのビット1～15によって提供され得る。しかしながら、ビット線との接続を考えると、クロスパーセル200の構成記憶回路205が出力バスのビット0に接続されている一方で、クロスパーセル230の構成記憶回路235はビット1に接続され、クロスパーセル260の構成記憶回路265はビット2に接続されていることが分かる。各クロスパーセルのチャンネル解放回路は、関連付けられた構成記憶回路と同じビット線に接続されている。一方、アービトレーション回路は、関連付けられた構成記憶回路に接続されたビット線を除くすべてのビット線に接続されている。伝送動作モードでは、関連付けられた構成記憶回路のルーティング値が論理1値に設定されている場合、伝送回路は入力バス上のデータを関連付けられた出力バスに接続するため、当然、伝送回路はすべてのワード線およびビット線に接続されている。

30

40

【0099】

50

図 6 A , B は図 5 A , B と同様の図であるが、行 0 と、列 0、列 1、および列 2 の間のそれぞれの交差点に提供されるクロスパーセル 2 0 0、3 0 0、3 3 0 を示している。クロスパーセル 2 0 0 については図 5 A , B を参照して前述したとおりであり、クロスパーセル 3 0 0、3 3 0 は同じ様式で構成されている。したがって、クロスパーセル 3 0 0 の素子 3 0 5、3 0 7、3 0 8、3 1 0、3 1 5、3 1 7、3 2 0 および 3 2 5 ならびにクロスパーセル 3 3 0 の素子 3 3 5、3 3 7、3 3 8、3 4 0、3 4 5、3 4 7、3 5 0 および 3 5 5 は、それぞれクロスパーセル 2 0 0 の素子 2 0 5、2 0 7、2 0 8、2 1 0、2 1 5、2 1 7、2 2 0 および 2 2 5 に対応する。本例では、すべてのクロスパーセルが同じ行と関連付けられているため、それらはそれぞれの出力バスのビット線とまったく同じ様式で接続されている。しかしながら、入力バスを考えた場合、チャンネル 0 へのアサートされた伝送要求または解放要求はワード線 0 上での入力である一方で、チャンネル 1 へのアサートされた伝送または解放要求はワード線 1 上の入力であり、チャンネル 2 へのアサートされた伝送または解放要求はワード線 2 上の入力である。したがって、各クロスパーセルの構成記憶回路、チャンネル解放回路およびアービトレーション回路は、それに応じて、異なる入力ワード線によって駆動される。しかしながら、示される実施形態においては、優先順位記憶回路は、優先順位割当動作モード中に、すべて同じワード線 1 ~ 1 5 からプログラムすることができる。

10

【 0 1 0 0 】

図 7 は、伝送回路およびアービトレーション回路が、各ビット線にどのように接続されているかをより詳細に示している。具体的には、素子 4 2 0 が個々のビット線と関連付けられた伝送回路の部分を示し、素子 4 4 0 が個々のビット線と関係付けられたアービトレーション回路の部分を示している。

20

【 0 1 0 1 】

1 6 ビット幅のチャンネルを考えると、アービトレーション回路は、そのクロスパーセルが接続された構成記憶回路 4 1 0 へのビット線を除くすべてのビット線と関連付けられるため、1 6 本のビット線のうちの 1 5 本が図 7 に示す様式で接続される。構成記憶回路 4 1 0 は、いったんクロスパーセルに提供され、そのクロスパーセルに対してアサートされた伝送要求またはアサートされた解放要求の存在下において設定される記憶構成信号を受信し、図 5 A , B および図 6 A , B について上で論じたように、その設定された記憶構成信号に反応して、接続された特定のビット線上の値をその構成としてサンプリングする。これはアービトレーション動作モード中に起こり、この結果として、アービトレーション動作モードの終わりに論理 1 ルーティング値が構成記憶回路 4 1 0 に記憶される場合、クロスパーセル回路が次いでデータ伝送動作モードに入る時、伝送回路 4 2 0 のトランジスタ 4 2 2 がオンになることが分かる。したがって、対応するワード線上のデータ入力も論理 1 値である場合は、これによりトランジスタ 4 2 4 がオンになり、ひいてはビット線 4 2 6 がディスチャージするが、このビット線は、以前にプリチャージ回路 4 0 0 によって論理 1 レベルまでプリチャージされている。ビット線 4 2 6 のあらゆるディスチャージが、感知増幅回路 4 3 0 によって感知される。

30

【 0 1 0 2 】

アービトレーション動作モードでは、伝送回路は用いられず、その代わりにアービトレーション回路 4 4 0 が用いられ、AND ゲート 4 5 5 からのトランジスタ 4 2 2 によって受信される入力に依存して、ビット線 4 2 6 を選択的にディスチャージする。具体的には、優先順位記憶素子 4 5 0 が論理 1 値を記憶し、アサートされた伝送要求が対応するワード線を介して受信された場合、論理 1 値がトランジスタ 4 4 2 に入力され、トランジスタ 4 2 6 のディスチャージを引き起こす。アービトレーション動作モードの終わりにかけて、アサートされた伝送要求を受信していた別のクロスパーセルのビット線 4 2 6 と接続されている任意の構成記憶回路が、そのクロスパーセルに適用可能なルーティング値を決定するために、ビット線上の値のサンプリングを行う。

40

【 0 1 0 3 】

別個の優先順位記憶素子 4 5 0 が、それぞれの関連付けられたアービトレーション素子

50

440に提供され、その値は、優先順位割当動作モードの間に優先順位構成モジュール60からアサートされるロード優先順位信号の存在下においてプログラム可能である。具体的には、アサートされたロード優先順位信号の存在下において、そこに記憶された優先順位の値を再プログラムするために、優先順位記憶素子450は、関連付けられたワード線上の値をサンプリングする。

【0104】

図8は、一実施形態に従った伝送回路および関連付けられた解放回路を示す。16ビットのチャンネルを考えると、このように1本のビット線が各クロスパーセル内に接続される。伝送回路の動作は図7について論じたものと変わらないため、ここでさらに論じることはない。しかしながら、解放回路460が、図7のアービトレーション回路440の代わりに、このビット線462に接続されている。また、クロスパーセルの構成記憶回路410も、ビット線462に接続されていることにも留意されたい。図中に示したANDゲートに対応するANDゲート470は、図5A、Bおよび図6A、Bでチャンネル解放回路に供給しており、したがって、解放チャンネル制御信号がアサートされ、かつ関連付けられた入力ワード線上のビットがアサートされた場合、このことは特定のチャンネルに対してアサートされた解放要求の存在を示唆するものであり、図8から分かるように、トランジスタ465がオンになり、ビット線462をディスチャージする結果となる。続いて、構成記憶回路410は、ビット線462上の値を再度サンプリングさせるアサートされた記憶構成信号を受信させられ、それによって論理0ルーティング値がそこに記憶され、こうしてチャンネルが解放される。

【0105】

上記の技法を用いると、図9に概略的に示されるように、パケットスイッチングクロスパーを実装することが可能となる。パケットスイッチング環境において、ソースは最初に要求を送信し、次いで承諾された信号を受信し次第データを送信し、要求とデータは同じ入力線を介して送信される。本発明の実施形態についての上記考察から、アサートされた伝送要求を、その後、伝送動作モードにおいてデータを担持するために用いられるのと同じワード線を介して入力可能とすることにより、本発明の実施形態のクロスパー回路が、このようなパケットスイッチング実施形態に迅速に役立つことが分かる。図5A、Bおよび図6A、Bを参照して上で論じた要求チャンネルおよび解放チャンネル制御信号は、2ビットの要求/解放入力線500を介して入力することができ、アサートされた伝送要求またはアサートされた解放要求は、関連付けられたデータ入力経路12を介して入力される。関連付けられたソース回路が承諾されたことを示唆するために、アービトレーション動作モード中に、クロスパーセル内の特定の構成記憶回路が論理1値を記憶する時、承諾信号が線505を介してソース回路へと返される。単一要求の実装では、いずれかの特定のソース回路が、任意の時点において、1つの出力チャンネルを要求することしかできない場合、単一の承諾信号線505のみが要求される。しかしながら、マルチキャスト（1つのソース回路が、任意の時点で複数のチャンネルを介してデータを同時通信することができる）を支持する代替の実施形態においては、ソース回路はいつの時点でも複数のアサートされた伝送要求を発行することができ、そのような実施形態においては、マスターがどのチャンネルのアクセスを承諾したかを特定するために、複数ビットの承諾信号線505が提供され得る。

【0106】

アービトレーション動作モード中は、それぞれのクロスパーセルも、線510を介してチャンネルフリー信号を受信して、関連付けられたチャンネルが要求中のソースに自由に割り付けられるかどうかを示唆する（この信号の使用は後に図10および11を参照して論じられる）。クロスパー回路によって用いられる優先順位スキームを変更することが所望される場合、その動作モード中に、新しい優先順位スキームデータを、ワード線を介した入力として各クロスパーセル内の優先順位記憶回路に記憶させるために、優先順位割当動作モード中に経路515を介してロード優先順位信号が発行されてもよい。

【0107】

10

20

30

40

50

したがって、図9および本発明の実施形態についての上記考察からも理解されるように、このパケットスイッチングクロスバーの実装において、プリチャージおよび条件付きディスチャージスキームが、伝送動作モード中にクロスバー回路を介してデータを伝送するために用いられる。前述のアービトレーション動作モード中には、コリジョンの検出および解消のためにビット線が再利用され、アサートされた伝送要求を伝送するためにワード線が用いられる。ワード線は、優先順位割当動作モードの間に、クロスバーセル20内に個々の優先順位記憶素子を記憶するための優先順位データを送信するためにも再利用される。2ビットの要求/解放信号は、特定のアービトレーション動作モード中に、伝送要求または解放要求がアサートされるかどうかを特定するために用いることができる。伝送要求がアサートされたと仮定すると、その要求が要求されたチャンネルを獲得するのに成功したかどうかをソースに示唆するために、次いで承諾信号が用いられる。

10

【0108】

既存のワード線（入力バス）およびビット線（出力バス）が、上記の機能性のすべてを実現するために用いられ、それによって、これらの機能性のすべてが最小限のルーティングオーバーヘッドで達成可能となる。

【0109】

図10A~10Dは、第1の実施形態に従って、各クロスバーセル20内に提供される回路をより詳細に示しており、ソース回路は一度に単一の伝送要求のみ発行することができる。図7および8についての上記考察から明らかなように、出力経路の各ビット線について、伝送回路を形成するために対のトランジスタ710、715、720が提供され、これらのトランジスタのうちの一つは、その入力で、対応するワード線上の値を受信し、他方のトランジスタは、その入力で、関連付けられた構成記憶素子700におけるルーティング値を受信する。プリチャージ動作段階中はディスチャージ信号が論理0値に設定されるため、条件付きディスチャージ回路705に、構成記憶回路700のコンテンツから各伝送回路のこれらのトランジスタ710、715、720を切り離させる。しかしながら、プリチャージ動作が終了し、感知動作が開始したことを示すためにディスチャージ信号が高くなる時、構成記憶素子700が、今度は各伝送回路の第2のトランジスタ710、715、720を駆動するために用いられる値を提供し、第1のトランジスタで受信される入力データに依存して、関連付けられたビット線の選択的なディスチャージを引き起こすことが分かる。条件付きディスチャージ回路705もWL_b信号を用い、図10Bは、WL_b信号がどのように生成され、図10Bの回路765がクロスバーセルごとに1回提供されるかを示す。

20

30

【0110】

図10Aの右側に示されるように、ビット線のうちの一つは、ANDゲート732および関連付けられたトランジスタ712によって形成される解放チャンネル回路に接続される一方、他のすべてのビット線は、それぞれアービトレーション回路および関連付けられた優先順位記憶素子717、742および722、752に接続されている。構成要素730、740および750の各々は、構成要素760と組み合わせて作動し、プリチャージ動作モード中に、関連付けられたトランジスタ712、717、722を分離する、条件付きディスチャージ回路を提供する。具体的には、プリチャージ動作モード中は、ディスチャージ信号が低いためにWL信号の値にかかわらずブロック760からの出力を低減させ、同様に構成要素730、740および750の出力を低減させるので、トランジスタ712、717、722がオフになる。

40

【0111】

しかしながら、ディスチャージ信号が高くなる時は、WL信号が高くなる場合、これによって論理1値が構成要素760から出力される。したがって、アービトレーション回路ならびに関連付けられた優先順位記憶素子717、742および722、752を考えると、関連付けられた記憶素子742、752も論理1値を記憶する状況においてアサートされた伝送要求の場合、このことが関連付けられたトランジスタ717、722をそれぞれディスチャージさせ、それによって上で論じた優先順位スキームが実装される。しかし

50

ながら、優先順位記憶素子のうちのいずれかが論理0値を記憶する場合、またはアサートされた伝送要求が存在しないためにWL信号がアサートされない場合は、対応するトランジスタ717、722はディスチャージしない。

【0112】

同様に、解放回路712、732を考えると、アサートされた解放要求の存在下において、解放信号および入力0のワード線の両方が高く設定される場合、これは解放トランジスタ712のディスチャージを引き起こし、後に構成記憶素子700によってビット線が再度サンプリングされ、ルーティング値が0にリセットされる。

【0113】

図10Bは、アサートされた要求チャンネルまたは解放チャンネルの信号を認めるために用いられる回路を示す。それは、要求/解放チャンネルの信号が認められた場合にのみ高く設定されるWL信号を生成する。このWL信号は、次いで構成記憶素子700を有効化するために用いられる。同じWL（およびWL_b）信号は、条件付き解放回路705およびアービトレーション回路760を適切に動作させるためにも用いられる。

10

【0114】

図10Cは、記憶された値を更新させるため、またその更新された値に依存して承諾信号を生成するために、構成記憶素子700の感知増幅およびラッチ回路775に関連して提供される回路を示す。具体的には、感知有効(SE)信号が設定され、かつWL信号が設定される場合、これにより、ANDゲート770が設定されたQSE（認められたSE）信号を出力させる。QSE信号が設定される場合、これにより、対応するビット線上の電流値が感知増幅およびラッチ回路775よりサンプリングされ、本例では、感知増幅およびラッチ回路はビット線0に接続されている。

20

【0115】

承諾信号は、感知増幅およびラッチ回路775の現在のコンテンツから生成される。具体的には、現在のコンテンツおよびWL信号に基づいて、NANDゲート780が、分離回路785にルーティングされる信号を出力する。分離回路785は、特定の行において1個のクロスバーセルのみが承諾信号を駆動することを確実にする。具体的には、ソース回路が現在要求しようとしている列と関連付けられたクロスバーセルのみが、承諾信号を生じることができる。したがって、本例では、伝送要求が高く設定され、かつワード線0上の入力が高く設定される場合、論理1値は分離回路785内のより低いトランジスタに提供され、論理0値はより高いトランジスタに提供されるため、インバータがオンになり、承諾信号が伝播される（感知増幅およびラッチ回路775が論理1値を記憶される場合、承諾信号がアサートされる）。しかしながら、そうでない場合は、論理0値はより低いトランジスタに提供され、論理1値はより上位にあるトランジスタに提供されるため、インバータがオフになり、承諾信号は伝播されない。

30

【0116】

図10Dは、一実施形態に従った各優先順位記憶素子742、752の構造を示す。この実施形態において、各優先順位記憶素子はSRAMセル790によって提供されることが分かり、図10の特定の例においては、SRAMセルは6T SRAMセルである。優先順位割当動作モードにおいてロード優先順位信号が高く設定されると、対応する入力ワード線に提供されたデータがSRAMセルによってサンプリングされ、そこに記憶された優先順位データが更新される。

40

【0117】

図11A～11Dは、本発明の代替の実施形態に従って各クロスバーセル内に提供される回路を示しており、各ソースは同時に複数の要求を発行することができる。これにより、クロスバー内でマルチキャストを実行するための、より効率的な実現形態が可能になる。具体的には、図10A～10Dの実施形態において、ソース回路は一度に1つのチャンネルのみ要求できるため、マルチキャストが要求される場合は、複数のチャンネルを獲得するために複数のアービトレーション周期を要求する。しかしながら、図11A～11Dの実施形態では、ソース回路は同時に複数のチャンネルに対する要求を発行するこ

50

とができ、潜在的に1つのアービトレーション周期で複数のチャンネルを獲得することができるため、マルチキャストがより効率的に達成可能となる。

【0118】

図11A、11Bおよび11Dは図10A、10Bおよび10Dとまったく同じであり、したがって、ここでは詳細に論じることはしない。しかしながら、図11Cに示すように、分離回路の必要がまったくないので、構成記憶回路の構造は大きく簡素化されている。代わりに、各列と関連して別個の承諾線が提供されているため、感知増幅およびラッチ回路775の現在のコンテンツは、設定されたWL信号の存在下において、承諾信号を直接生成するために用いることができる。具体的には、WL信号が設定された場合、ソース回路はアサートされた伝送要求を発行し、感知増幅およびラッチ回路775の現在のコンテンツが論理1値に設定されて伝送要求が成功であったことが示され、次いで論理1の承諾信号が発行されてソース回路に返される。

10

【0119】

図12は、一実施形態に従って、各クロスバーセル内に提供される構成記憶回路(すなわち、感知増幅およびラッチ回路)の配設を示す。図10Cおよび11C中のANDゲート770によって生成されるQSE信号は、インバータ850を通過してQSE_b信号を生成する。したがって、QSE信号が高くなると、PMOSトランジスタ800が開かれ、感知増幅(感知増幅はトランジスタ805、810、815、820から形成される)の感知動作が開始される。感知動作の始めに、トランジスタ815および820がオフになる。トランジスタ805にルーティングされた入力線がディスチャージし始める場合、これによりトランジスタ805がオンになり、中間モードIをVddに引き上げる。これにより代わりにトランジスタ810がオンになり、短時間で感知増幅器内に正のフィードバックループが作製され、その時点では、感知増幅器は、その出力を生成するために、接地へと遷移する入力信号には依存しない。

20

【0120】

一連のトランジスタ825、830、835は、感知増幅と構成要素840および845から構成されるラッチとの間の転送機構としての役割を果たす。具体的には、動作の感知段階中にトランジスタ830がオンになり、ラッチのコンテンツが、感知増幅の中間ノードIの値によって決定されるようにする。具体的には、感知増幅への入力が論理0値へと遷移する場合は、これにより中間ノードIがVddへと遷移し、トランジスタ835がオンになり、ラッチ内に論理0値が記憶されることが分かる。反対に、入力線がディスチャージしない場合は、中間ノードの電圧は論理0レベルに留まり、トランジスタ825をオンにしてラッチ内に論理1値が記憶される。

30

【0121】

QSE信号が低くなるとトランジスタ830がオフになり、感知増幅出力からラッチが分離されるため、ラッチは、QSE信号が高い期間のみ、その入力を更新することが理解されよう。

【0122】

図13A~13Cは、SEおよびディスチャージ信号の両方を生成するために一実施形態において用いられるパルス生成回路を示す。図13Aは、構成要素860によっていくつかの遅延が導入された後に、SE信号からどのようにディスチャージ信号が生成されるかを示している。実際には、図13Aの回路はANDゲートとしての役割を果たし、SE信号と遅延されたSE信号とのANDを行う。

40

【0123】

図13Bは、SE信号を作製するために用いられるリング発振器の設計を示す。発振入力信号が高い時は、リング発振器を有効化してクロック信号を生成し、SE信号はいくつかの遅延回路870を介したクロック信号に由来する。

【0124】

図13Cに示すように、SE信号が低くなる間の期間、つまりd1期間は構成要素870によって決定され、ディスチャージが高くなる間の期間、つまりd2期間は図13Aの

50

遅延回路 860 によって決定される。

【0125】

電圧 V_{b_SE} および $V_{b_}$ ディスチャージは、遅延期間 d_1 および d_2 を制御するために用いられ、具体的には、これらの電圧を減少させると、それに応じて遅延期間が長くなる。

【0126】

クロックのポジティブエッジでは、データがワード線上に立ち上げられる。この時点では「ディスチャージ」信号が低いため、ビット線はワード線から分離されている。「 d_1 」期間の間は、データはワード線上に留まり、ビット線が完全にプリチャージされる。「 d_2 」は、ビット線が条件付きでディスチャージされる期間である。同時に、SE 信号が高くなり、感知増幅がビット線のサンプリングを開始する。ちょうど感知増幅が検出するのに十分なだけ、ビット線がディスチャージされる。これは節電のために行われる。したがって、「 d_2 」時間の後、「ディスチャージ」信号が低くなるが、一方で、SE 信号は高いままで留まる。

10

【0127】

図 14 は、図 10B および 11B に参照されるチャンネル__フリー信号が一実施形態に従ってどのように生成されるかを概略的に示している。具体的には、示されるように、特定の出力チャンネルに接続されたクロスパーセル内の様々な構成記憶回路 900、905、910 からの出力が、OR ゲート 902、907、912 によってともに論理的に OR が行われ、インバータ 915 によってその結果が逆にされる。したがって、構成記憶回路のうちのいずれかが論理 1 ルーティング値を記憶する場合は、チャンネル__フリー信号が論理 0 値となり、それらすべてが論理 0 ルーティング値を記憶する場合にのみ、チャンネル__フリー信号が 1 に設定されることが分かる。したがって、アービトレーション動作モードの開始時点で、構成記憶回路のうちの 1 つが既に論理 1 値に設定されており、ソース回路が既にそのチャンネルのオーナーシップを有することを示唆している場合、その時点においてアサートされた伝送要求は 1 つも承諾されず、そのチャンネルがいったん解放された時点でのみ、別のソースが要求を行い、そのチャンネルに対して承諾されることが可能になる。

20

【0128】

図 1 に示される優先順位構成モジュールは、様々な様式で配設することができる。図 15A は、優先順位構成モジュール 930 の一実現形態を示す。この実施形態では、優先順位構成モジュール 930 は、各データ入力経路 940、945、950 のワード線を用いて、クロスパーネットワーク内に周期的に優先順位を書き込む有限状態機械 (FSM) としての役割を果たす。本例では、データ入力線 940 がソース回路 0 に連結され、データ入力経路 945 がソース回路 1 に連結され、データ入力経路 950 がソース回路 15 に連結されている。優先順位構成モジュール 930 が各ソースのために優先順位ベクトル 935 を維持し、また本例では、値 0 が最も高い優先順位を示し、値 15 が最も低い優先順位を示す。したがって、現在選択されている優先順位ベクトル 935 に従って、ソース 0 が最も優先順位の高いソースであり、ソース 1 が 2 番目に優先順位の高いソースであり、ソース 15 が最も優先順位の低いソースである。したがって、論理 1 値は、データ入力経路 940 を介してソース回路 0 と関連付けられたクロスパーセルの優先順位記憶素子内にプログラムされるため、ソース回路 0 がいつ要求をアサートしようとも、ソース回路 0 に連結された構成記憶回路によって監視されるビット線を除く、他のすべてのビット線を確実にディスチャージすることが分かる。

30

40

【0129】

同様に、データ入力バス 945 を介して、ソース回路 0 と関連付けられたビット線を除くすべてのビット線に接続された優先順位記憶素子が論理 1 値に書き込まれ、ソース 1 からのアサートされた伝送要求の存在下において、これらのビット線が条件付きでディスチャージされるようになる。しかしながら、そのようなアサートされた伝送要求の場合でも、ソース回路 0 と関連付けられたビット線はディスチャージされない。

【0130】

50

同様に、ソース回路 15 を考えると、論理 0 値は対応するデータ入力バス 950 を介して出力されるため、ソース回路 15 と関連付けられたクロスバー内のすべての優先順位記憶素子が論理 0 値を記憶し、そのためソース回路 15 からのアサートされた伝送要求の存在下において、いかなるビット線もディスチャージされなくなる。そのような状況において、ソース回路 15 は、その時点で他のソース回路が伝送要求をアサートしていない場合にのみ、そのアサートされた伝送要求を承諾する。

【0131】

しかしながら、クロスバーの使用における公平性を確保とするため、本実施形態の優先順位構成モジュール 930 は、図 15 A に概略的に示される他の優先順位ベクトルのうちの 1 つに切り替えることで、周期的に優先順位を更新する。具体的には、示される実施形態では、優先順位構成モジュールがラウンドロビンスキームにおける優先順位を割り当てるので、経時的に、各ソース回路が最も高い優先順位のソースとなる期間を有する。

10

【0132】

しかしながら、図 15 B に示すような代替の実施形態において、優先順位構成モジュールは、適応性優先順位構成モジュール 960 の形態を取る。適応性優先順位構成モジュールは 1 つの優先順位ベクトル 965 を維持するのみであるが、この優先順位ベクトルは、アクティビティ分析回路 970 によって実行されるアクティビティ分析に依存して、優先順位ベクトル修正回路 975 によって修正される。具体的には、アクティビティ分析回路 970 は各ソースからのアサートされた伝送要求を監視し、また任意選択的には、経時的にどのソースがどのチャンネルにアクセスを承諾されるかについての理解を構築するために、それらのアサートされた伝送要求に回答して生成される承諾信号も監視する。適応性優先順位構成モジュール 960 は、クロスバーネットワークの全体的な効率を高めることを目的として、単に図 15 A のラウンドロビン手法を用いるよりも、よりインテリジェントに優先順位を割り当てるために、この情報を用いる。効率は、クロスバーネットワークが伝送動作モードにある動作時間の割合として定義することができる（すなわち、ビット線がデータを送信するためには使用されるが、コンフリクトを検出および解消するためには使用されない、時間の長さ）。

20

【0133】

アクティビティ分析回路 970 によって受信されるアクティビティ情報に依存して、どのように優先順位ベクトルを更新するかを決定することができる方法が多数存在することが理解されよう。例えば、優先順位ベクトルを更新するための可能なスキームには以下を含む。

30

【0134】

1) 優先順位が周期的に回転されるラウンドロビンスキーム
 2) 乱数発生器を用いてランダムな優先順位を生成するランダム割り当て
 3) LRG (承諾後最長時間経過: least recently granted) スキーム。このスキームでは、チャンネルがチャンネルの獲得に成功すると、すぐさまそのチャンネルにとって最も低い優先順位が割り当てられる。同時に、最も高い優先順位がそのチャンネルを獲得してから最長時間が経過しているソースに割り当てられる。

40

【0135】

上述の実施形態では、出力データ線が高くプリチャージされるので、静的な高入力、クロックサイクルごとにそれらのデータ線をディスチャージさせる。図 16 A ~ 16 E は、これを軽減するのに使用することができ、それによって電力消費を低減する、符号化スキームを示している。より具体的には、図 16 A は、ソースとクロスバーの入力データ経路との間に配置されてもよい、符号化器回路を示し、図 16 B は、感知増幅器 65 の出力と宛先回路との間に配置されてもよい、関連付けた復号器回路を示す。図 16 A の符号化器回路を先ず考えると、新しいデータの各アイテムは、フリップフロップ 600 内にラッチされ、古いデータは NAND ゲート 605 を経てフリップフロップ 610 に伝搬される。次いで、比較器 615 が新しいデータと古いデータとを比較し、差異があればいつでも、論理 0 値をインバータ 620 に出力させ、その結果、論理 1 値が符号化器回路によって

50

出力される。したがって、提供された入力データから、符号化器は、入力データが論理 0 値から論理 1 値に、または論理 1 値から論理 0 値に変化するたびに、論理 1 値によって分離される論理 0 値を生成する。クロスパーセルの上述の考察から、データ出力線は、このような論理 1 値が現れた時にだけディスチャージされ、それによって、プリチャージ動作に関連付けられる電力消費を低減することを認識されるであろう。

【0136】

第 1 のデータアイテムが入力された時には、それに対していかなる以前のデータも比較されず、それに応じて、`sync` 信号を使用して、フリップフロップ 610 内の初期状態を設定する。さらに、クロスパーを新しい構成に切り替えるたびに、同期化パルスを使用して、符号化器ハードウェアを（クロックサイクルを費やさずに）再設定することができる。

10

【0137】

対応する復号器回路を図 16B に示す。構成要素 630 および 635 の構造は、構成要素 660 によって図 16C に概略的に示す。示されているように、このような構成要素は、実際に、直列に接続される一連のトランジスタ 665、670、675、680 から成る。

【0138】

`sync_d` 信号は、`sync` 信号と同じであるが、1 クロックサイクルだけ遅延される。`sync_d_b` 信号は、逆 `sync_d` 信号である。これらの 3 つの信号の関係を図 16E に示す。`sync` 信号は、アクティブロー信号であり、したがって通常動作下において（回路が新しい構成に切り替えられていない時）、`sync` および `sync_d` は、高い論理 1 レベルになり、一方で、`sync_d_b` 信号は、低い論理 0 レベルになることに留意されたい。

20

【0139】

16B から分かるように、復号器は、感知増幅器、クロック信号、および `sync_d_b` 信号から出力を受信し、一連の NAND ゲート 645、650、655 を経て、内部クロック信号をフリップフロップ 640 に提供する。フリップフロップ 640 からの出力は、出力を宛先回路に駆動し、感知増幅器を経て受信される符合化した出力から、元の入力データを再作成する。

【0140】

通常動作において、`sync` 信号は、論理 1 レベルであり、`sync_d_b` 信号は、論理 0 レベルであるので、構成要素 635 が起動され、一方で、構成要素 630 が停止される。3 個の NAND ゲート 645、650、655 は、感知増幅器が遷移を検出して、高い信号を送信する時に、パルスをフリップフロップ 640 に提供する。新しい構成への切り替えが生じた時には、`sync` および `sync_d` 信号が低くなり、一方で、`sync_d_b` 信号が高くなる。したがって、その時に、構成要素 630 が起動され、一方で、構成要素 635 が停止される。その結果、インバータ 635 を介して以前のデータをトグルする代わりに、フリップフロップ 640 は、感知増幅器からデータを（構成要素 630 がそれを逆にした後）に取り込む。

30

【0141】

図 16D は、符号化器回路への入力データ、結果として生じた符合化器回路からの符号化したデータ出力、符号化したデータが高くなるたびにディスチャージされる、対応するビット線の指示を示し、また、感知増幅器から受信した情報に基づいて、復号器回路によって生成される出力データを示している。出力データは、符号化器に渡された入力データを忠実に再生していることが分かる。

40

【0142】

上で論じた実施形態では、クロスパー回路のサイズは入力バスおよび出力バスのサイズに対応する。したがって、16 本のワード線を提供する入力バスおよび 16 本のビット線を提供する出力バスを考慮して、16 × 16 のクロスパー回路が論じられてきた。しかしながら、その技法は、入力バスおよび出力バスの幅よりも大きいクロスパーセルのレイ

50

を有するクロスバー回路を用いて動作させるためにも適応することができ（本明細書において幅狭チャンネル設計と称される）、または実際にクロスバーセルのレイがバス幅よりも小さい状況に適応させることもできる（本明細書において幅広チャンネル設計と称される）。

【0143】

最初に幅狭チャンネル設計を考えると、図17はそのような幅狭チャンネル設計を実装するための一実施形態を示す。この特定の例では、クロスバーセル20の64×64のレイがクロスバー内に提供されており、ここでもまたプリチャージモジュール55が、クロスバーを通るビット線をプリチャージするために用いられることが推定される。しかしながら、データ入力経路およびデータ出力経路はわずか16ビットの幅しかないため、単一周
10
期において上述のアービトレーションプロセスを実行することは不可能である。代わりに、そのような実施形態では、クロスバーをいくつかのセクションに分割することが可能であり、本例ではクロスバーが1010、1020、1030、の4つのセクションに分割され、カウンタ1040は現在どのセクションが有効であるかを特定するために用いられる。したがって、そのアービトレーション周期におけるアービトレーション中に、どのセクションが働くべきかを選択するために、アービトレーション周期ごとにカウンタをインクリメントすることができる。各セクションは、それでも各行に64個のクロスバーセルを有するため、データ入力経路1000単独では、アサートされた伝送要求によって特定される個々のデータ出力経路を有効化しないことは明らかである。したがって、一実施形
20
態において、関連付けられたソース回路によってどのデータ出力経路が要求されるかを特定するために、伝送要求をアサートする時に2ビットの追加信号1002が提供される。

【0144】

したがって、カウンタが00の値を有する時、ソース0～15からのアサートされた伝送要求のみが検討され、各データ出力経路には、そのデータ出力経路へのアクセスを要求しているソース0～15の中で最も高い優先順位のソースが、次の伝送周期においてデータを転送するためのアクセスを承諾される。そのようなアービトレーションおよび伝送シ
30
ーケンスの後、カウンタが01にインクリメントされ、次のアービトレーション周期では、ソース16～31からのアサートされた伝送要求のみが検討される。このプロセスは各セクションで順に繰り返される。結果として、図17の例の寸法については、任意の所与のアサートされた伝送要求に対するアービトレーション待ち時間は、1周期から4周期ま
30
までと異なることが理解されるであろう。例えば、カウントが00のアービトレーション周期において、ソース0が伝送要求をアサートする場合、その要求はその周期内ですぐに調停され、セクション1における最も優先順位の高い要求である場合、その要求は承諾される。しかしながら、カウントが01の時にソース0が伝送要求をアサートする場合、カウンタが00に戻る（それによって、セクション1である1010が調停される）次の4番目のアービトレーション周期まで、その要求はアービトレーションの候補にさえならない。

【0145】

そのような手法の利点は、配線および論理のオーバーヘッドが最小限であること、またそのため、該ソリューションは、アービトレーション待ち時間における変動が許容され得
40
る状況下で、良好なソリューションを提供することである。しかしながら、セクションの数が増加するにつれて、最悪の場合のアービトレーション待ち時間も増加することに留意すべきである。

【0146】

図18Aおよび18Bは、クロスバーはここでもまたセクションに分割されているが、アービトレーションは階層的に2段階で行われる、代替の実施形態を示す。第1の段階で、セクション間でアービトレーションが実行され、第2の段階で、アービトレーションは所与のセクションにおけるすべてのアサートされた伝送要求間で実行される。同じビット線群が両方のアービトレーション段階のために用いられる。本例では、クロスバーセルは20
50
という数で表され、それぞれのクロスバーセルは、基本的には上記で論じたように

構成されているが、ここでは、各クロスパーセルが2つの感知増幅有効ラッチ (S A E L) を含み、第1の S A E L は第1のアービトレーション周期で用いられ、第2の S A E L は第2のアービトレーション周期で用いられる。

【 0 1 4 7 】

したがって、アービトレーション周期1において、セクション1050、1060、1070、1080の各々が検討されるが、所与のセクション内の特定のデータ出力経路に対する個々のクロスパーセル20は、ボックス1052、1062、1072、1082によって概略的に示されるように、事実上組み合わせて検討される。具体的には、16ビットのチャンネルを有する図18Aの64×64クロスパーの例を考えると、4つの別個のセクション1050、1060、1070、1080の各々が相対的な優先順位を有する。セクション1が最も高い優先順位を有する場合、ソース0～15からのいずれかのアサートされた伝送要求が、セクション2、3および4（すなわちセクション1060、1070、1080）へのビット線をディスチャージさせる。結果として、セクション1では第1の組の S A E L が論理1値をラッチし、他のセクションでは第1の組の S A E L が論理0値をラッチする。

10

【 0 1 4 8 】

次のアービトレーション周期では、図18Bに示されるように、セクション0におけるリクエストのみ（上で論じたイベントのシーケンスと仮定して）が、アービトレーションにビット線を用いる。このセクション1050内で実行されるプロセスは上で論じたとおりであり、すなわち、ビット線がプリチャージされ、次いで、各クロスパーセル内に記憶された優先順位情報に従って、選択的にディスチャージされる。図17において上で論じられたように、どの出力経路が、ワード線1000を介して16ビット値の入力によって要求されているかを認めるために、追加の2ビット信号1002が用いられる。いずれかの特定のデータ出力線に対する複数の要求については、ソース0～15のうちで最も優先順位の高い要求がアクセスを承諾され、第2の組の S A E L が標準的な様式でビット線を検知する。特定のソースに承諾が与えられたかどうかを決定するために、次いでクロスパーセル20の両方の S A E L が検討される必要があり、具体的には、それらの両方が論理1値を記憶する場合、承諾信号が生成されてソースへと返される。

20

【 0 1 4 9 】

そのような手法は、追加の論理を犠牲にして固定された2周期のアービトレーション待ち時間を有するが、追加の相互接続構造は有さない。チャンネルに16ビット線を持つことで、潜在的に16セクション間で調停することが可能になり、各セクションは16個の要求を含む。したがって、アービトレーション待ち時間は、2周期で256×256のクロスパーまで保有することができる。

30

【 0 1 5 0 】

図18A / 18Bの手法の一実施形態において、各クロスパーセルに2組の優先順位記憶素子を提供することができ、1つの組は第1のアービトレーション周期と関連して用いられ（第1の組の S A E L が用いられる）、1つの組は第2のアービトレーション周期と関連して用いられる（第2の組の S A E L が用いられる）。

【 0 1 5 1 】

図19は、入力および出力チャンネルの幅がクロスパーアレイのサイズを上回る、幅広チャンネルの実現形態を示している。この特定の例では、入力および出力経路は32ビット幅であり、クロスパーは16×16クロスパーであると仮定されるため、クロスパーセル20の16×16アレイを形成している。このような配設では、アービトレーション中に用いられる必要があるのはビット線の半分のみである。そのような条件下において、複数組の優先順位記憶回路を提供することにより、各クロスパーセルに複数の優先順位が記憶できるようにし、続いて、優先順位構成モジュール60からいずれの優先順位データもリロードすることを必要とせずに、優先順位構成の切り替えができるようにすることで、この事実の利点を利用することができる。具体的には、優先順位構成モジュールは、各クロスパーセルに2組の優先順位ビットをプログラムすることができ、その後、いずれの時間ベ

40

50

ナルティもなく、これらの2組の優先順位ビット間で選択的に切り替えることが可能であることが理解されるであろう。具体的には、図19に示される実施形態において、優先順位カウンタ1130は、プリチャージモジュール55と関連して保有されることができ、各クロスパーセル20内にどの優先順位ビットの組が用いられるべきかを特定するために、その優先順位カウンタの値が線1120を介して出力される。どの優先順位ビットの組が選択されるかに依存して、このことは、データ出力経路1110の様々なビット線が、ソース0~15のうちいずれかからのアサートされた伝送要求に応じてどのようにディスチャージされるかを変更する。優先順位割当動作モード中に、単一周期において両方の組の優先順位をチャンネルにロードする能力が与えられる場合、優先順位割当動作モードを頻繁に用いる必要がないため、このことは、そのような状況に著しいパフォーマンスの向上を提供する。さらに、その後の2組の優先順位ビット間の切り替えは、いずれの時間ペナルティもなく生じ得る。

10

【0152】

図19に示される実施形態において、優先順位カウンタ1130は、シングルビットカウンタでさえあれば良いことが理解されるであろう。しかしながら、他の構成においてマルチビットカウンタを用いることができる。例えば、64ビットチャンネルを有する16×16クロスパー回路では、各クロスパーセル内に4組の優先順位を記憶することが可能であり、したがって、2ビットの優先順位カウンタ1130をそのような実施形態で用いることができる。

20

【0153】

図20は、データ入力経路を介して、データ入力値をそれらのデータ入力経路に接続された様々なクロスパーセル20まで伝播するために、ワード線ドライバ1200、1205、1210のシーケンスが用いられる、部分的に有効化されたネットワークの実施形態を示す。このような部分的に有効化されたネットワークなしでは、それが必要とされない状況においてデータ入力経路全体を介してデータを伝播する際に、著しい電力が消費され得る。

【0154】

しかしながら、データ入力経路を介して残りのクロスパーセルの各々に記憶されるルーティング値に由来する信号に基づいて、ワード線ドライバの各々が条件付きで有効化される、図20に提示される部分的に有効化されたネットワークの手法を用いることにより、節電を達成することができる。具体的には、示されるように、特定のクロスパーセルに記憶されるルーティング値は、他のすべてのクロスパーセルに保有されるルーティング値によって、特定の行においてその右側に論理的にORされ、ワード線ドライバがそのチェーン内で到達する点のその信号の値に依存して、各ワード線ドライバ1200、1205、1210が有効化される。

30

【0155】

したがって、上に挙げた例では、行0の最初の15個のクロスパーセルのうち少なくとも1つは、そのルーティング値が1に設定されており、対応するソースは出力経路0~15のうちの一つにアクセスを承諾されているが、その行のさらに右側にある他のクロスパーセルは、いずれもルーティング値が1に設定されておらず、両方のワード線ドライバ1205および1210が論理0有効信号を受信する一方で、ワード線ドライバ1200は論理1有効信号を受信することを示している。図20の下半分の回路に示されるように、それぞれのワード線ドライバ(例示目的のためにワード線ドライバ1200が検討されているが、ワード線ドライバのすべてが同様に構成されている)には、ワード線ドライバを選択的に有効化するために有効信号が用いられる。具体的には、ワード線ドライバ内の主インバータ1260は有効信号が高い場合にのみ有効化され、これによって、トランジスタ1250および1255の両方がオンになる。有効信号が低いときは、これらのトランジスタはオンにならず、インバータ1260は有効化されない。有効化されると、インバータ1260および1265は組み合わせて作動し、さらにデータ入力経路を介して伝播するために入力データを増幅する。

40

50

【0156】

したがって、図20を参照して考察した実施形態において、その後に残りのクロスバースェルスイッチがオフにされるチャンネルを超えて、データがワード線上で駆動されることはない。これによって、大きなクロスバースェルに有用な節電スキームを提供することができ、めったにマルチキャストイングを行わない実現形態において特に良好な節電をもたらすことができる。

【0157】

上記の実施形態の説明から、このような実施形態は、マルチコアアプリケーションに必要とされ得るような、非常に大きいクロスバースェルを設計するのに使用することができる、新規なクロスバースェル実装スキームを提供することが分かるであろう。本発明の実施形態のクロスバースェル回路は、コンフリクトを検出および解消し、それに応じて各クロスバースェル内の構成記憶素子をプログラムするために、アービトレーション動作モード中にデータ出力線を再利用する。提案した本発明の実施形態の実装は、クロスバースェルの設計が可能な最小の面積を使用する。交差点の数および各交差点に組み込まれる論理の数の両者が増大する、典型的な先行技術の実現形態とは対照的に、本発明の実施形態によれば、交差点の数だけが増大しない。

10

【0158】

クロスバースェルのアレイのサイズが入力および出力チャンネルの幅を上回らない実施形態においては、単一周期においてアービトレーションを実行することが可能であり、それによって、アービトレーションを実行する際の遅延オーバーヘッドが最小限に留められる。さらに、実施形態はプログラム可能な優先順位を提供し、ランタイム中のアプリケーションを最適化するために、さらなる余裕を持たせることができる。例えば、マルチメディアプロセス中、より重要な仕事を行うソース回路は、より高いレベルのキャッシュ等、メモリデバイスにアクセスするためのより高い優先順位を与えられることが可能である。

20

【0159】

さらに、アービターをクロスバースェルと統合することにより、その2つの間の通信オーバーヘッドが排除される。

【0160】

加えて、上記の実施形態のクロスバースェル構造を採用した時に、各入力データ経路上に入力データを提供するのに使用される、伝送回路のサイズも、駆動回路のサイズも、クロスバースェル回路のサイズが増加した時に、サイズを増加させる必要が無い。さらに、単一の入力データ値が複数の出力経路に同報通信される、マルチキャストイングをサポートするのに、いかなる追加のハードウェアまたはサイジングも不要である。

30

【0161】

さらに、上記の実施形態には標準的な構造が採用されるため、ルーティングは、典型的な先行技術の手法よりもかなり直接的になる。

【0162】

本発明の一実施形態では、入力は、例えば、マルチプレクシング手法を採用している典型的な先行技術のクロスバースェルで必要とされ得るような、異なるバスからの同じビットを備えるパケットとしてではなく、バスとしてクロスバースェル内に供給される。これは、ワイドバスの相互接続部として、本発明の実施形態のクロスバースェル回路を利用し易くする。

40

【0163】

本発明の実施形態のクロスバースェル回路の設計により、クロスバースェルを通じた遅延は、サイズとともに直線的に増大する。これは、このような実施形態のクロスバースェル回路を、 128×128 または 256×256 入力/出力のクロスバースェル等の、非常に大きいクロスバースェルに容易に使用できるようにする。

【0164】

上述のように、本設計は、非常に標準的であり、全てのチャンネルにわたる遅延は均一である。さらに、入力は、いずれの水平端部からも供給することができ、出力は、いずれの垂直端部でも利用できる。これは、ルーティングの輻輳を低減することによって、設計フ

50

ロアプランニングを容易にする。

【0165】

本発明の実施形態のクロスバー回路の標準ビットセル様アーキテクチャに起因して、クロスバー回路の設計およびレイアウトは、既存のCAD（コンピュータ支援設計）フロープロセスに容易に統合することができる。

【0166】

先行技術の手法と比較すると、本発明の実施形態のクロスバー回路は、相互接続のサイズがより小さく、高速感知技術が用いられているため、より迅速に動作しながらも、消費する電力はより少ない。

【0167】

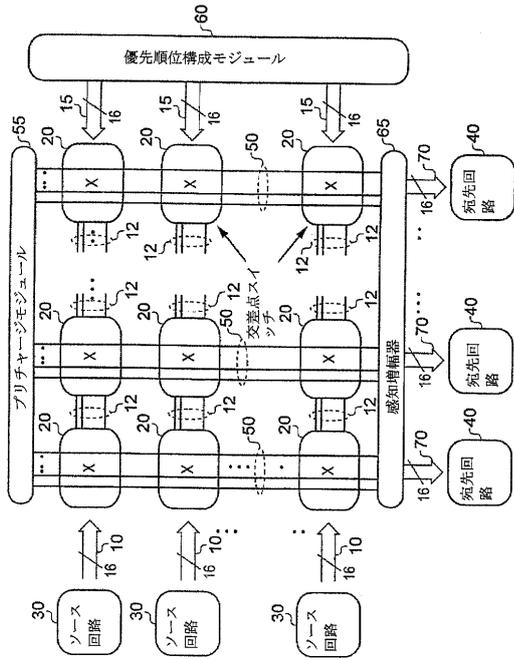
本明細書では本発明の特定の一実施形態を説明したが、本発明は、それに限定されるものではなく、本発明の範囲内で、多数の変更および追加が行われ得ることが明白であろう。例えば、本発明の範囲を逸脱しない範囲で、以下の従属請求項の特徴の、独立請求項の特徴との種々の組み合わせを行うことができる。

【符号の説明】

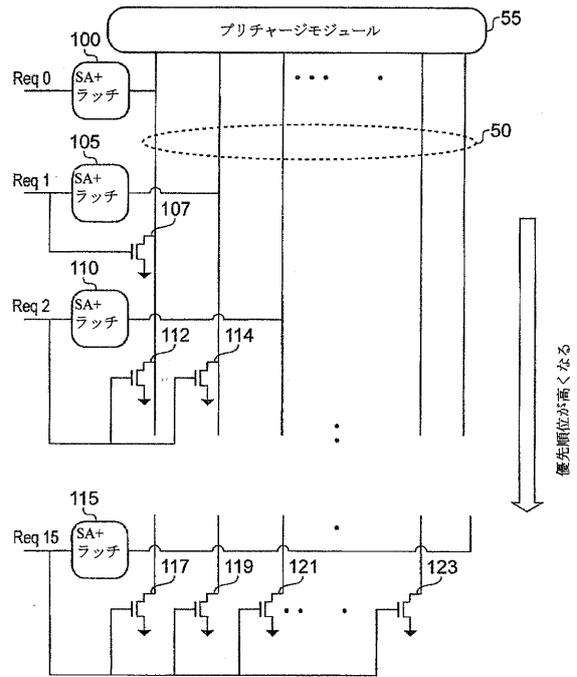
【0168】

10	入力データ		
12	データ入力経路		
15	ソース回路		
20	クロスバーセル		20
30	ソース回路		
40	宛先回路		
50	データ出力経路		
55	プリチャージモジュール		
60	優先順位構成モジュール		
65	感知増幅回路		
70	出力データ		
200、230、260、300、330	クロスバーセル		
205、235、265、305、335	構成記憶回路		
210、240、270、310、340	伝送回路		30
215、245、275、315、345	チャンネル解放回路		
220、250、280、320、350	アービトレーション回路		
225、255、285、325、355	優先順位記憶回路		
207、217、237、247、267、277	ANDゲート		
208、238、268	ORゲート		
307、317、337、347	ANDゲート		
308、338	ORゲート		

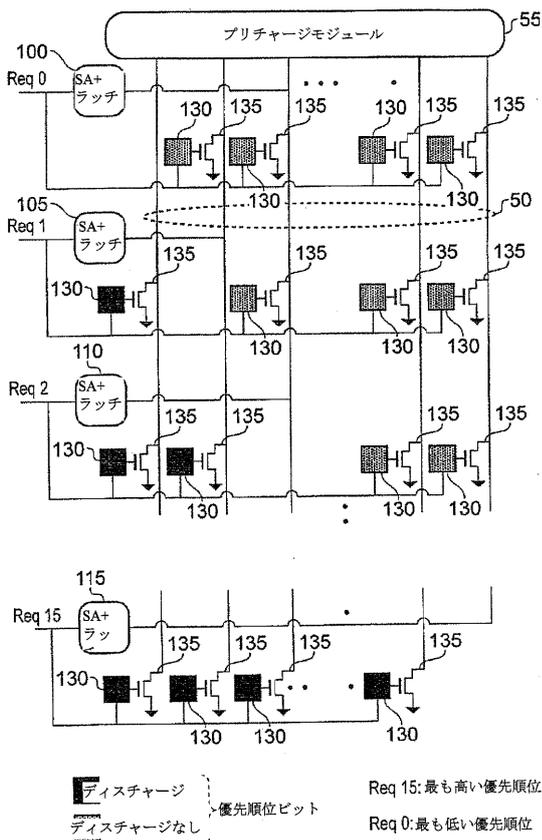
【図1】



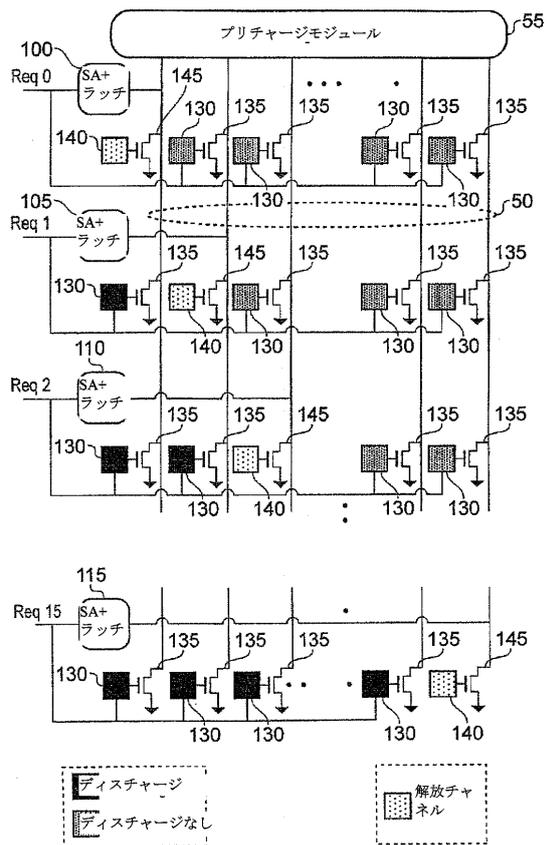
【図2】



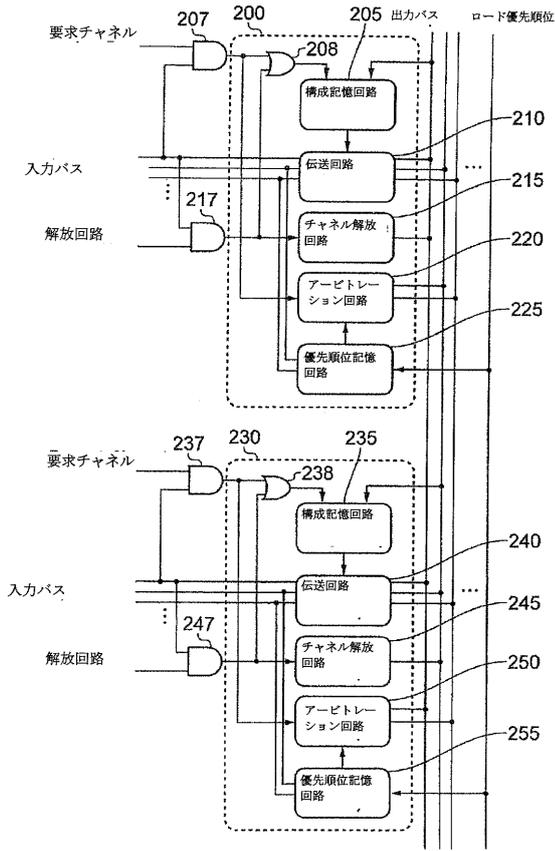
【図3】



【図4】

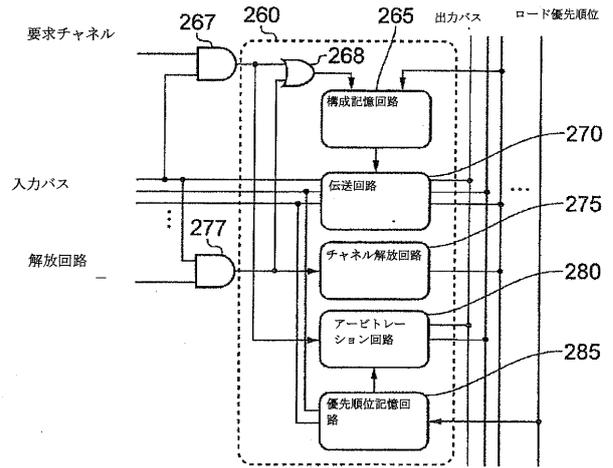


【図 5 A】

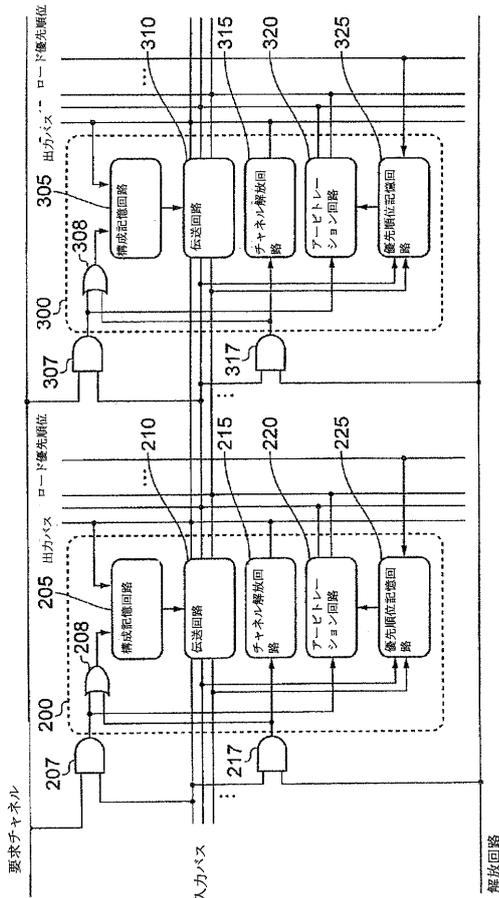


【図 5 B】

(図 5 A の続き)

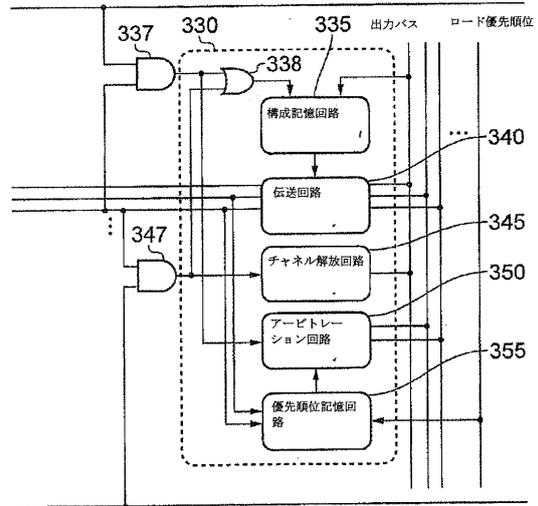


【図 6 A】

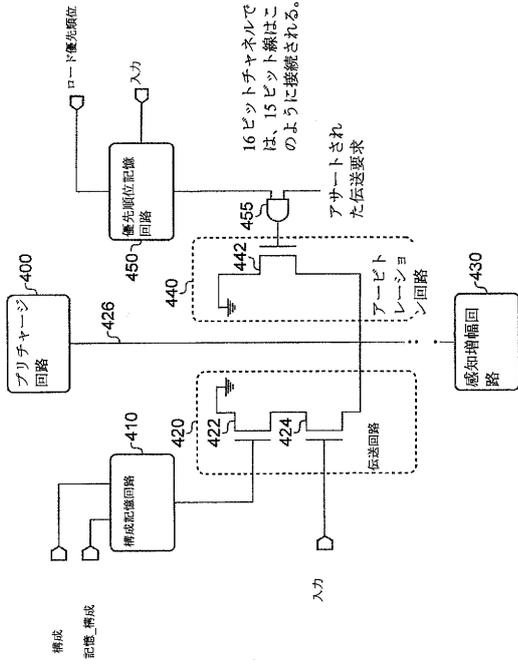


【図 6 B】

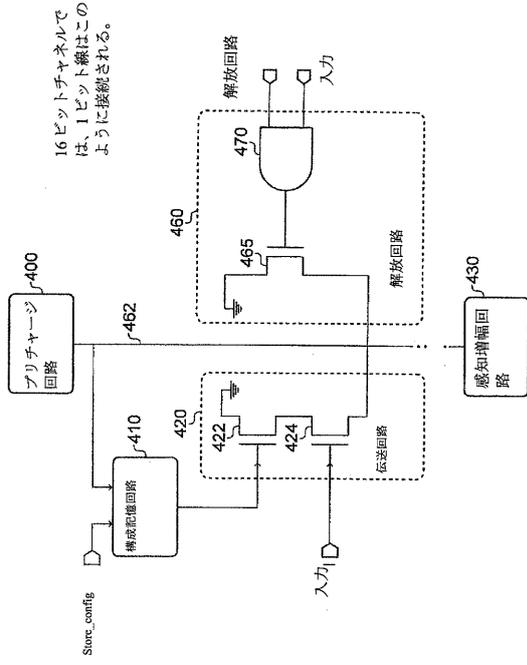
(図 6 A の続き)



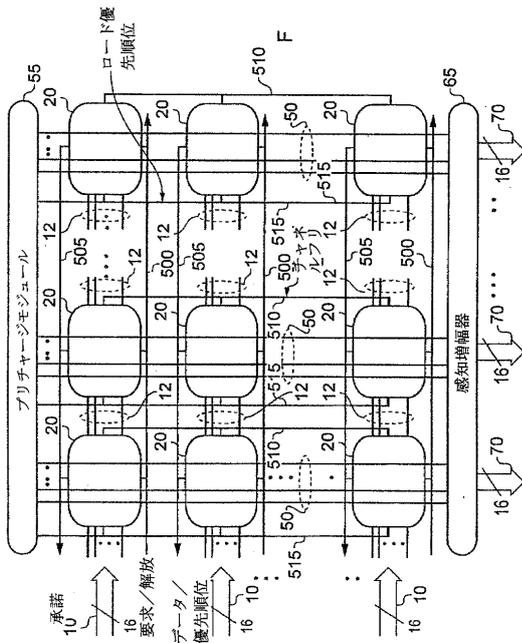
【図 7】



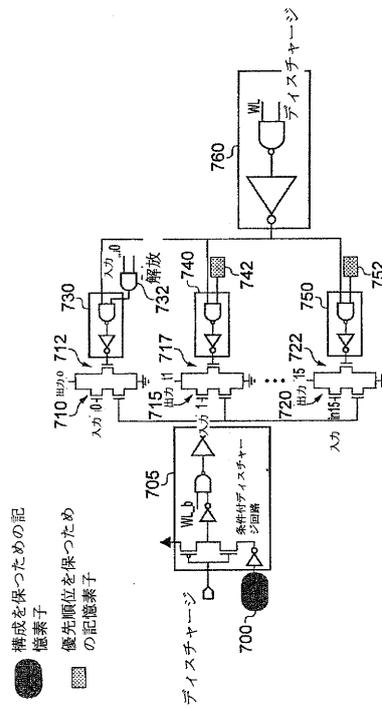
【図 8】



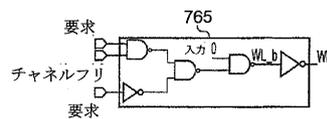
【図 9】



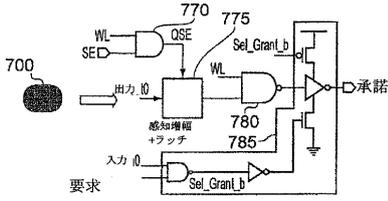
【図 10 A】



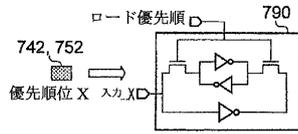
【図 10 B】



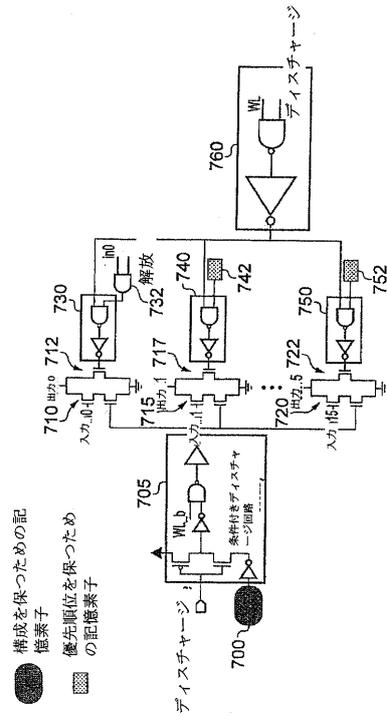
【図10C】



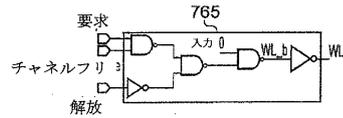
【図10D】



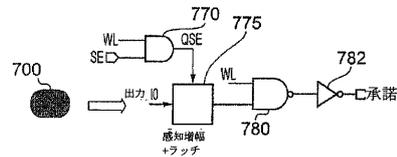
【図11A】



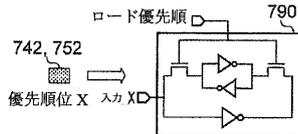
【図11B】



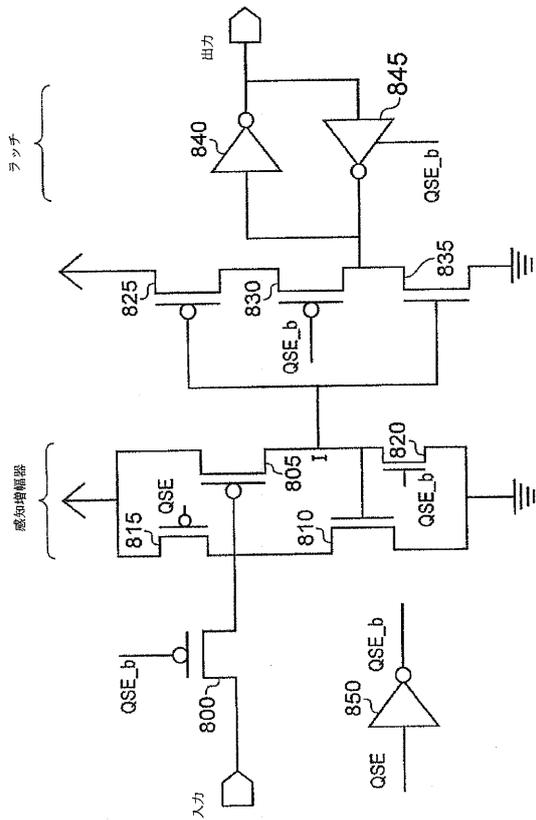
【図11C】



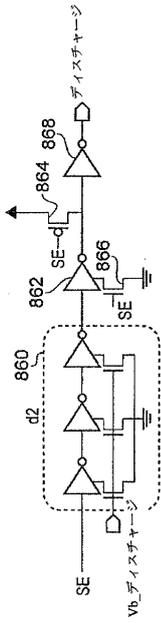
【図11D】



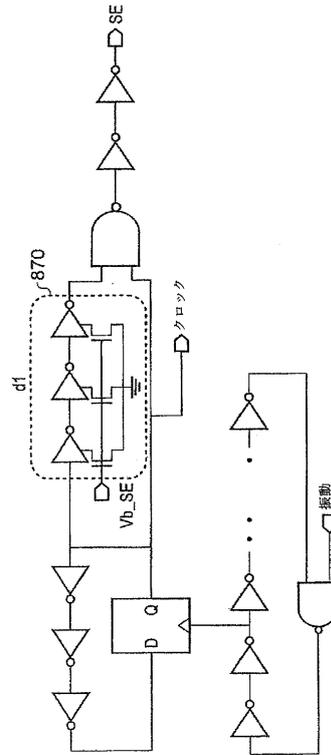
【図12】



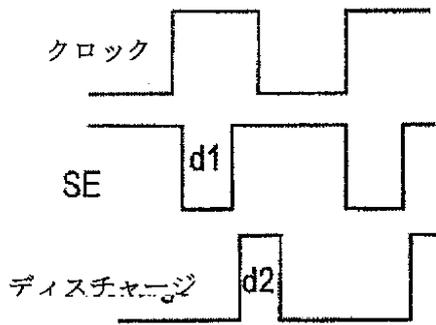
【図 13 A】



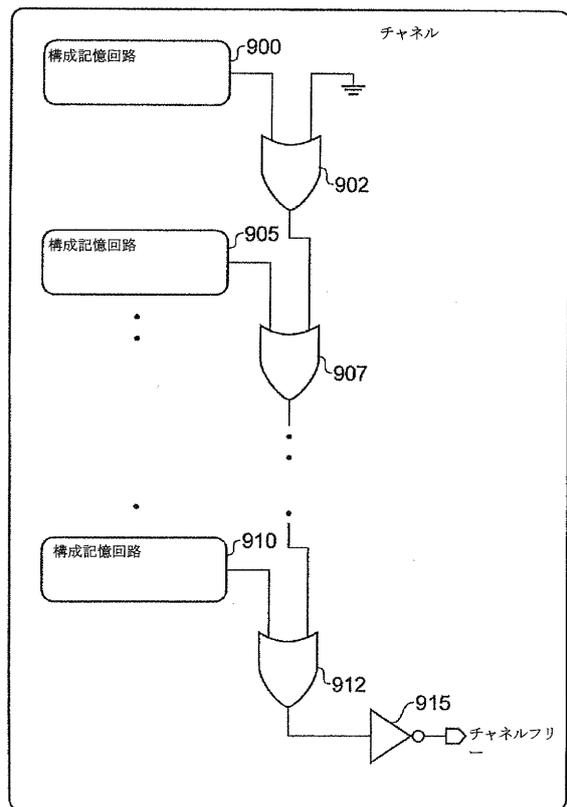
【図 13 B】



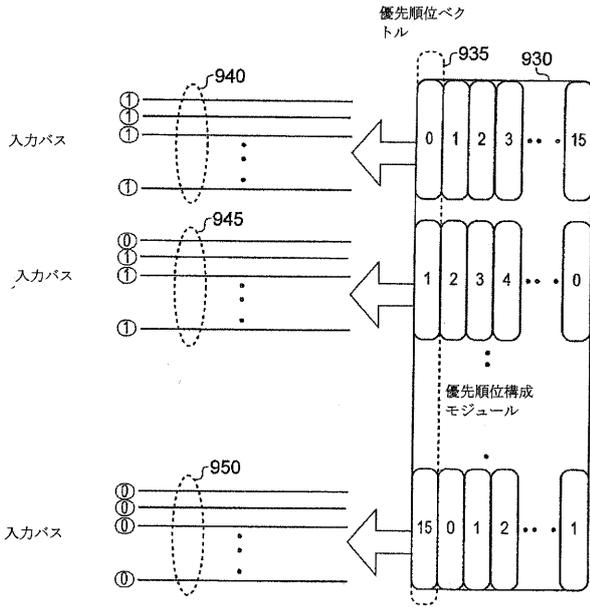
【図 13 C】



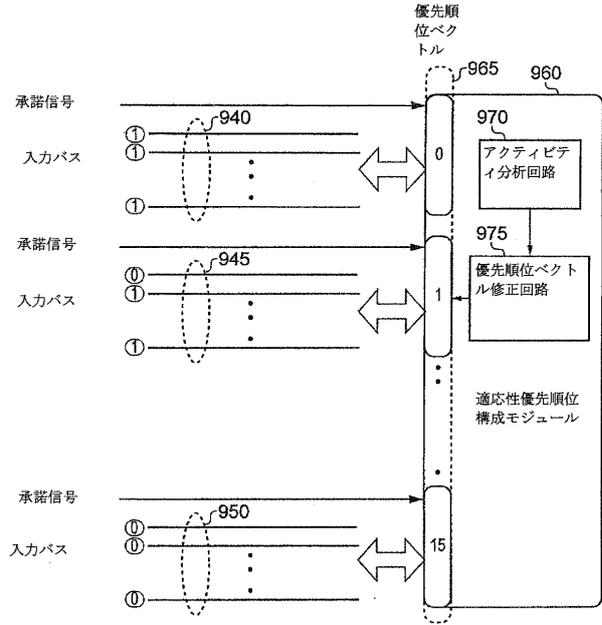
【図 14】



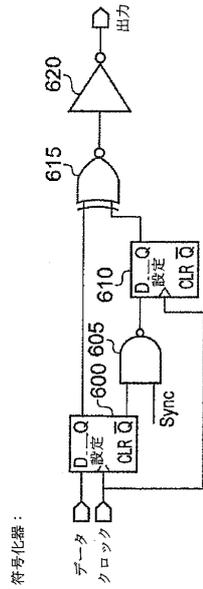
【図15A】



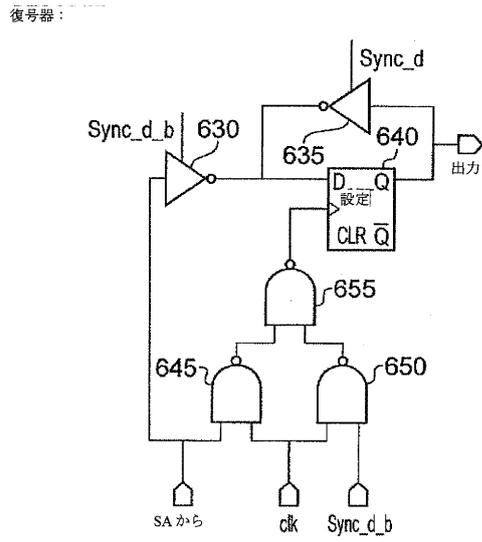
【図15B】



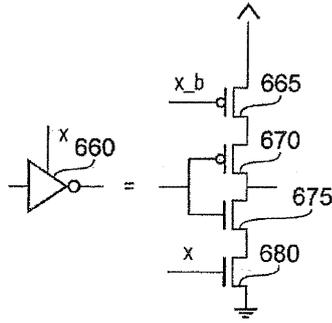
【図16A】



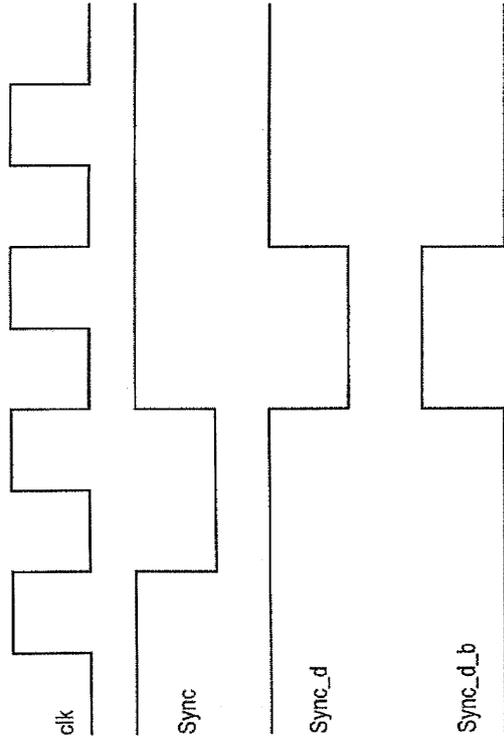
【図16B】



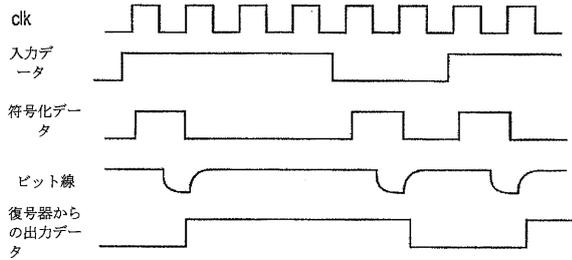
【図16C】



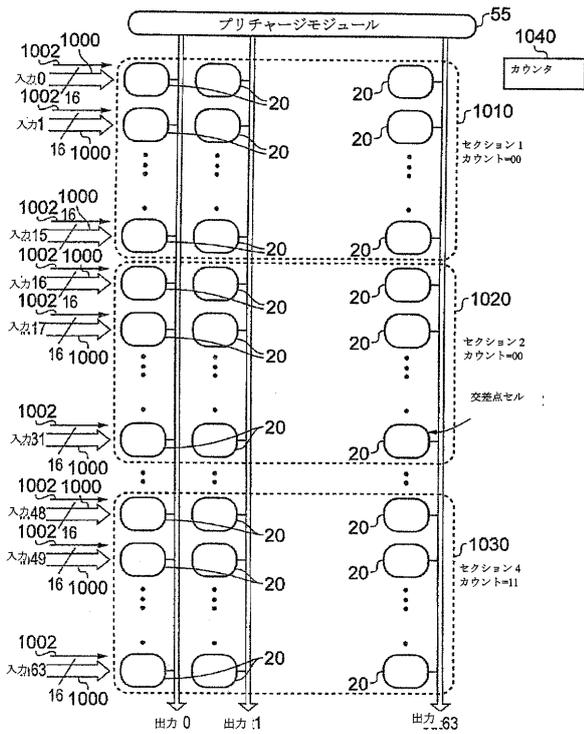
【図16E】



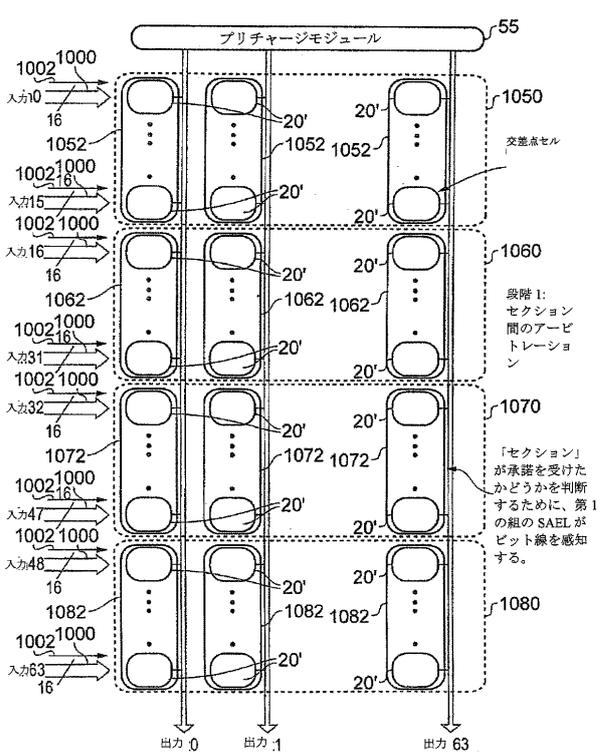
【図16D】



【図17】



【図18A】



フロントページの続き

- (74)代理人 100089037
弁理士 渡邊 隆
- (74)代理人 100110364
弁理士 実広 信哉
- (72)発明者 スティール・クマール・サトパティ
アメリカ合衆国・ミシガン・48105・アン・アーバー・シャーリー・レーン・1835・アパートメント・エー8
- (72)発明者 デヴィッド・セオドア・ブラウ
アメリカ合衆国・ミシガン・48014・アン・アーバー・グレンウッド・ロード・1811
- (72)発明者 トレヴァー・ナイジェル・マッジ
アメリカ合衆国・ミシガン・48105・アン・アーバー・ウィンストン・ドライブ・3801
- (72)発明者 デニス・マイケル・シルヴェスター
アメリカ合衆国・ミシガン・48103・アン・アーバー・ハーレー・ドライブ・1912
- Fターム(参考) 5K069 AA08 AA17 DB02 DB07 DB09 FC02

【 外国語明細書 】

CROSSBAR CIRCUITRY AND METHOD OF OPERATION OF SUCH CROSSBAR CIRCUITRY

Field of the Invention

The present invention relates to crossbar circuitry and a method of operation of such crossbar circuitry.

Description of the Prior Art

A crossbar circuit is a switch infrastructure for connecting multiple inputs to multiple outputs in a matrix manner. Accordingly, crossbar circuitry can be used to interconnect a plurality of source circuits and a plurality of destination circuits such that data input to the crossbar circuitry from any of the plurality of source circuits can be output to any of the plurality of destination circuits. Crossbar circuits can be used in a variety of implementations. For example, in a data processing system implementation, such crossbar circuitry can be used to interconnect a plurality of processors used to perform data processing operations on data values with a plurality of memory devices used to store those data values, thereby allowing the data values from any memory device to be routed to any processor.

Known techniques for producing crossbar circuits require a large area for the crossbar circuitry due to the components required to form the crossbar circuitry and the significant number of control lines required for routing control signals to those components, and also consume significant power. Further, their complexity tends to grow rapidly with size, making many of the known techniques impractical for use with crossbar circuits required to interconnect a large number of source circuits with a large number of destination circuits. Some known techniques are discussed below.

The article "A 50 Gb/s 32x32 CMOS Crossbar Chip using Asymmetric Serial Links" by K Chang et al, 1999 Symposium on VLSI Circuits, Digest of Technical Papers, Pages 19 to 22, and the article "A 2 Gb/s 256x256 CMOS Crossbar Switch Fabric Core Design using Pipelined MUX" by T Wu et al, IEEE International Symposium on Circuits and System, 2002, Pages 568 to 571, describe crossbar circuits using a hierarchical arrangement of multiplexers to enable the routing of data from any input source to any output destination. However, such MUX-based crossbar circuits are relatively large in size, and have a large power consumption. Further, they typically require a considerable

number of control lines in order to control the various multiplexers. Such MUX-based designs are not typically scalable with an increase in the number of inputs and outputs to be supported, at least in part because it becomes increasingly difficult as the size increases to route the necessary control signals to the various multiplexers. Further, if the input data is multi-bit data routed over an input bus, the routing of the data paths themselves becomes very complex.

The article “A Novel Reduced Swing CMOS Bus Interface Circuit for High Speed Low Power VLSI Systems” by R Golshan et al, IEEE International Symposium on Circuits and System, Pages 351 to 354, 1994, describes an X-Y style crossbar circuit where the input paths pass in a horizontal direction and the output paths pass in a vertical direction. At the intersection between each input path and output path, a storage element in the form of a flip-flop circuit is provided whose output controls a transistor used to couple the input path to the output path. However, such a design requires a large number of control lines to program the various flip-flops in order to configure the crossbar circuit to perform the required routing. Further, the input data provided on an input data path is used to drive the output data on the relevant output data path. As the crossbar circuit becomes larger to accommodate larger numbers of inputs and outputs, the capacitance of the output data paths increases, and accordingly it is necessary to provide larger drive transistors on the inputs in order to overcome the increased capacitance. In addition, the coupling transistor driven by the flip-flop at the interconnection between a data input path and a data output path also needs to increase in size as the crossbar circuit increases in size. Further, it will typically be necessary to include one or more buffers in the data output paths as the crossbar circuitry increases in size. All of these factors give rise to significant problems in layout of the elements of the crossbar circuit and the associated control lines, particularly as the crossbar circuit increases in size to accommodate more inputs and outputs. Accordingly, this approach becomes complex and is not scalable.

The article “High-Performance Crossbar Design for System-On-Chip” by P Wijetunga, Proceedings of the Third IEEE International Workshop on System-On-Chip for Real-Time Applications, 2003, describes a crossbar design employing a pass-transistor chain as transmission circuitry to be located at each intersection between the data input paths and the data output paths. When the pass-transistor chain is required to couple the input data to the output data path, a current sensing circuit located on the

output data path is used to detect the input data value. A significant disadvantage of this design is that it consumes large power due to the requirement for current sensing devices to be located on each output path. Further, a large number of control signals need routing within the crossbar circuitry, and indeed those control signals will dominate the routing requirements as the crossbar circuit increases in size to accommodate more inputs and outputs. Hence, again, this design is not readily scalable to larger designs of crossbar circuitry.

The article “A Multi-Context 6.4 Gb/s/Channel On-Chip Communication Network using 0.18 μm Flash-EEPROM Switches and Elastic Interconnects” by M Borgatti et al, ISSCC 2003, Session 26, Embedded and Digital Systems, Paper 26.5, describes a programmable crossbar implemented using a matrix of modified flash-EEPROM devices. However, as is clear from Figure 26.5.6, the number of memory cells involved in routing a data input from a source device to a destination device depends on where those devices are connected into the crossbar, and accordingly the timing for signals passing through the crossbar is not deterministic. Further, a large number of control lines will be required to enable programming of the various flash-EEPROM cells, and a significant amount of time will be required to program those various flash-EEPROM cells. Any reconfiguration of the crossbar device will hence also take a significant time.

Accordingly, such a design of crossbar is complex, and will increase in complexity as more input devices and output devices need to be supported by the crossbar, due to the proliferation in control lines required. Further, since the timing of the crossbar is not deterministic, this will make the crossbar design inappropriate for certain implementations.

In summary, it will be appreciated from the above discussions that existing crossbar designs typically involve complex routing of control signals, with that complexity rapidly increasing as the size of the crossbar increases. Often the designs consume significant power consumption and lack scalability, due partly to the number of control lines required, and partly due to the need to increase the size of certain components provided within the crossbar as the size of the crossbar increases.

Another issue in crossbar design is how to provide the crossbar with collision detection and resolution abilities. Multiple requests for the same destination in a

switching fabric is termed a collision. With increasing number of sources and destinations, collisions get more frequent. Under such circumstances, arbitration becomes a bottle-neck in the overall efficiency of the crossbar circuitry.

Most contemporary switching fabrics consist primarily of two modules: a crossbar to transmit data, and an arbiter to configure the crossbar. In such an implementation, the source circuits send requests to the arbiter for channels. The arbiter samples all requests and using some priority assignment scheme, grants some or all of the requests and configures the crossbar accordingly. This scheme poses two major problems for scalability:

- 1) Routing all request signals from the source circuits to the arbiter, and all the grant signals back, becomes increasingly difficult for larger systems; and
- 2) The arbiter needs to have knowledge of all incoming requests as well the current state of the crossbar, before it can make a decision. Monitoring the state of the crossbar every cycle requires additional logic as well as interconnects. This contributes to additional delay.

A number of attempts have been made to address the scalability and arbitration delay of packet switching networks. In the article by Chi H. et. al. entitled "Decomposed Arbiters for Large Crossbars with Multi-Queue Input Buffers," IEEE International Conference on Computer Design, 14-16 Oct 1991, pages: 233 - 238, the authors discuss decomposing arbiters, so that some requests can be granted before the arbitration process completes. However, the worst case arbitration delay still remains the same. In general, the arbitration delay grows linearly with size.

In the article by Delgado-Frias et. al. entitled "A VLSI Crossbar Switch with Wrapped Wave Front Arbitration," IEEE Transactions on Circuits and Systems, Volume 50, Issue 1, Jan. 2003 Pages: 135 – 141, and the article by Kavaldjiev N. et. al. entitled "A Virtual Channel Router for On-chip Networks," IEEE International SoC Conference, 12-15 Sept. 2004, pages: 289- 293, the authors discuss the handling of arbitration within a crossbar. However, the disclosed implementations are not scalable and are restricted to a crossbar of size 4*4.

In the article by Shin E. et. al. entitled "Round-robin Arbiter Design and Generation," International Symposium on System Synthesis, 2002, pages: 243-248,

the authors propose a tool to generate a round robin arbiter. The approach is hierarchical, attempting to make a 32*32 switch out of 4*4 switches.

In the article by William W. Plummer entitled "Asynchronous Arbiters", IEEE Transactions on Computers Archive Volume 21 , Issue 1 (January 1972) Pages 37-42, the article by Charles E. Molnar et. al. entitled "Simple Circuits that Work for Complicated Reasons," International Symposium on Advanced Research in Asynchronous Circuits and Systems, 2000. (ASYNC 2000) Proceedings, and the article by Mark B. Josephs et. al. entitled "CMOS Design of the Tree Arbiter Element," IEEE Transactions on VLSI systems, Volume 4, Issue 4, Dec 1996 Page(s):472 – 476, some asynchronous techniques for arbitration are described. These designs exploit meta-stability to achieve randomness. However, delay in meta-stable systems can become high at times, thereby restricting their use in real time systems that require guaranteed throughput.

In summary, many solutions available in literature take a hierarchical approach towards making a large crossbar switch using 2*2 or 4*4 switches. Hence, the delay grows linearly with the size of the crossbar switch. This leaves no room for voltage scaling in wide and parallel systems (SIMD/MIMD type applications).

Further, in previous attempts to integrate the arbiter and the crossbar together, such as for example discussed in the earlier mentioned article "A Virtual Channel Router for On-chip Networks," IEEE International SoC Conference, 12-15 Sept. 2004, pages: 289-293, it has only been possible to bring them spatially close. However, the logic and interconnect for both functionalities have remained exclusive.

In conventional implementations, collision detection and resolution is done hierarchically. This is achieved in multiple stages of arbitration logic, wherein after each stage the number of requests is reduced by a certain fraction.

Accordingly, it would be desirable to provide an improved crossbar design that alleviates the above described problems.

SUMMARY OF THE INVENTION

Viewed from a first aspect, the present invention provides crossbar circuitry for interconnecting a plurality of source circuits and a plurality of destination circuits such that data input to the crossbar circuitry from any of said plurality of source circuits can be output to any of said plurality of destination circuits, the crossbar circuitry

comprising: a plurality of data input paths passing through said crossbar circuitry, each data input path being connectable to one of said plurality of source circuits and providing a plurality of word lines; a plurality of data output paths passing through said crossbar circuitry transverse to the plurality of data input paths, each data output path being connectable to one of said plurality of destination circuits and providing a plurality of bit lines; a crossbar cell associated with each intersection between one of said data input paths and one of said data output paths, each crossbar cell comprising: configuration storage circuitry programmable to store a routing value, the routing value being programmed to a first value to indicate that data input along the word lines of the data input path to the associated intersection is to be output on the bit lines of the data output path at the associated intersection, and the routing value being programmed to a second value to indicate that data input along the word lines of the data input path to the associated intersection is not to be output on the bit lines of the data output path at the associated intersection; transmission circuitry which in a transmission mode of operation is responsive to the routing value having said first value to detect the data input along the word lines of the data input path and to output an indication of that data on the bit lines of the data output path at the associated intersection; and arbitration circuitry that operates in an arbitration mode of operation in dependence on a transmission request received by the crossbar cell from the source circuit connected to the data input path of the associated intersection, if the transmission request is asserted to indicate that said source circuit wishes to route data from the data input path to the data output path at the associated intersection, the arbitration circuitry being arranged to operate in combination with the arbitration circuitry of other crossbar cells associated with the same data output path to re-use the bit lines of the data output path to detect the presence of multiple asserted transmission requests for said same data output path, and in the event of such multiple asserted transmission requests to implement a predetermined priority scheme to cause the configuration storage circuitry of only one crossbar cell associated with said same data output path to have its routing value programmed to said first value, thereby resolving conflict between said multiple asserted transmission requests according to said predetermined priority scheme.

In accordance with the present invention, crossbar cells are associated with each intersection between a data input path and a data output path, with each crossbar cell comprising configuration storage circuitry arranged in a transmission mode of operation to selectively connect data on its data input path to its data output path in dependence on a routing value stored therein, and arbitration circuitry that operates in an arbitration mode of operation in combination with the arbitration circuitry of other crossbar cells in order to resolve conflicts between multiple asserted transmission requests. In particular, if a transmission request is asserted from a source circuit to a crossbar cell, then the arbitration circuitry of that crossbar cell operates in combination with the arbitration circuitry of other crossbar cells associated with the same data output path to re-use the bit lines of the data output path to detect the presence of multiple asserted transmission requests for that same data output path. In the event of such multiple asserted transmission requests, the arbitration circuits operate in combination to implement a predetermined priority scheme to cause the configuration storage circuitry of only one crossbar cell associated with the same data output path to have its routing value programmed to a first value (i.e. the value that causes the crossbar cell to couple its data input path to its data output path in the transmission mode of operation).

Hence, it can be seen that in accordance with the present invention, in the arbitration mode of operation, the bit lines of the data output paths are re-used to detect conflicts for particular data output paths, and to resolve those conflicts such that at any point in time only one source circuit is granted access to a particular data output path.

In the arrangement of the present invention, both the arbiter and the crossbar functions are integrated together within the crossbar circuitry. The arbiter is highly logic dominated while the crossbar is highly routing dominated, and by bringing these two functions together within the crossbar cells of the crossbar circuitry, a more efficient utilisation of routing tracks and silicon space in the chip is achieved. In addition, the communication overhead that traditionally is incurred between the crossbar and the arbiter is now alleviated. By storing the crossbar configuration at the crossbar cells associated with each intersection within the crossbar circuitry, this allows re-utilisation of the bit lines of the data output paths for arbitration.

In accordance with the present invention, the crossbar circuitry can be laid out in the style of an SRAM array topology, with the crossbar cells being laid out in an array, the data input paths providing word lines, and the data output paths providing bit lines. Such an arrangement allows for a very efficient layout both in terms of size and in terms of the number of interconnects required.

In one embodiment, each data output path provides n bit lines, and each configuration storage circuitry is associated with one of said n bit lines via which the routing value is programmed into the configuration storage circuitry, such that for up to n crossbar cells associated with the same data output path, the configuration storage circuitry of those crossbar cells have a different associated bit line of said n bit lines. In said arbitration mode of operation, the arbitration circuitry of any crossbar cells associated with the same data output path that receive an asserted transmission request are arranged to selectively modify the voltage on the n bit lines dependent on said predetermined priority scheme, whereafter each configuration storage circuitry of a crossbar cell receiving an asserted transmission request samples the voltage on its associated bit line in order to program said routing value.

Since the configuration storage circuits of different crossbar cells in the same data output path are associated with different bit lines, and the arbitration circuits of those crossbar cells in that data output path receiving an asserted transmission request selectively modify the voltage on the n bit lines, it is possible to implement the predetermined priority scheme by ensuring that at the point the configuration storage circuit of each crossbar cell receiving an asserted transmission request samples the voltage on its associated bit line following the arbitration process, only one of the bit lines will be at a value that would cause the associated configuration storage circuit to store said first value as the routing value, thereby allowing only a single source circuit to be granted access to a data output path at any point in time.

The manner in which the arbitration circuits selectively modify the voltage on the n bit lines can be varied dependent on implementation. However, in one embodiment, each of the bit lines are precharged to a first voltage level and then during the arbitration mode of operation the arbitration circuitry of any crossbar cells associated with the same data output path that receive an asserted transmission request selectively discharge the voltage on the n bit lines dependent on said predetermined

priority scheme. Hence, in such embodiments, the bit lines are conditional discharged in order to implement the predetermined priority scheme, thereby resolving any conflicts during the arbitration mode of operation.

There are a number of ways in which the transmission request can be asserted from the source circuits to the crossbar cells. However, in one embodiment, each data input path provides n word lines, and in the arbitration mode of operation the transmission requests for up to n crossbar cells associated with the same data input path are input to the crossbar circuitry via the n word lines of that same data input path. Hence, in such embodiments, the data input paths are themselves re-used in the arbitration mode of operation for the provision of asserted transmission requests.

In one particular embodiment, each configuration storage circuitry is associated with one of said n word lines, such that for up to n crossbar cells associated with the same data input path, the configuration storage circuitry of those crossbar cells have a different associated word line of said n word lines via which the transmission request is received.

In such embodiments where the n word lines of the data input paths and the n bit lines of the data output paths are re-used during the arbitration mode of operation, it can be seen that conflict detection and resolution can be performed in a single stage, thereby yielding significant performance benefits when compared with the conventional prior art approach where collision detection and resolution is done hierarchically in multiple stages.

In particular, it will be seen that for crossbar circuitry containing a matrix of up to $n \times n$ crossbar cells, conflict detection and resolution can be performed in a single stage.

However, the techniques of embodiments of the present invention can also be used with crossbar circuits having a larger matrix of crossbar cells. In particular, in one embodiment, a matrix of $mn \times mn$ crossbar cells are provided, wherein m is an integer of 2 or more, the matrix is divided into a plurality of sections, and a series of arbitration operations are used to implement the predetermined priority scheme to cause the configuration storage circuitry of only one crossbar cell associated with said same data output path to have its routing value programmed to said first value, thereby resolving conflict between multiple asserted transmission requests according to said

predetermined priority scheme. For each arbitration operation in the series, one or more of the sections in said plurality of sections are subjected to that arbitration operation.

In one particular embodiment, each arbitration operation operates on a single section, such that after said series of arbitration operations have been performed all of said plurality of sections have been subjected to said arbitration operation. Hence, by way of example, for a 64 x 64 crossbar circuit with 16-bit data input paths and data output paths, the crossbar circuitry can be divided into four sections, with the arbitration operation being performed in sequence on each of those sections. In one embodiment a counter can be incremented every cycle to select which section to serve during arbitration in that cycle (in the above example a two bit counter would suffice). Such an implementation still maintains the benefits of minimal wiring and logic overhead, but the arbitration latency for a given request can vary dependent on the section with which that request is associated (in the above example the latency can vary from one to four cycles).

In an alternative embodiment, in a first arbitration operation of the series, all of the plurality of sections are subjected to the first arbitration operation in order to identify one of the sections in said plurality that contains the configuration storage circuitry whose routing value is to be programmed to said first value. Then, in a second arbitration operation of the series, said one of the sections identified by the first arbitration operation is subjected to the second arbitration operation to identify the configuration storage circuitry within that section whose routing value is to be programmed to said first value. In accordance with such embodiments, the crossbar is again divided into sections, but arbitration is then performed hierarchically in two steps, first among the sections, and then among all requests in a given section. The same group of bit lines can be used for both arbitration steps.

In one embodiment, each crossbar cell is provided with two configuration storage circuits, the first being used to detect if the associated section contains the highest priority asserted transmission request, and the second being used in the subsequent arbitration step to detect if the highest priority asserted transmission request is associated with that particular crossbar cell.

By way of example, in a 64 x 64 crossbar with 16-bit data input paths and data output paths, the crossbar could again be divided into four sections. If section 0 has the highest priority and at least one source associated with that section asserts a transmission request, then in the first stage of arbitration the first configuration storage circuits in each crossbar cell in that section will have a routing value programmed to the first value, whilst those in the other sections will have their routing values programmed to the second value (in both cases the programming occurring as a result of the voltages on the bit lines associated with each first configuration storage circuit). In the next cycle, only asserted transmission requests in respect of section 0 are considered, with the bit lines again being used for arbitration, and with the second configuration storage circuits now sensing the associated bit lines. In the event that both configuration storage circuits at a particular crossbar cell have their routing value set to the first value, this indicates that the associated asserted transmission request has prevailed, and in the data transmission mode of operation that crossbar cell will connect its input to its output.

Such an implementation will have a fixed two cycle arbitration latency at the expense of some additional logic, but no additional interconnect. With 16 bit lines in a data output path, it is possible to arbitrate between up to 16 sections, with each section being able to receive up to 16 asserted transmission requests. Hence, the arbitration latency can be retained at two cycles for a crossbar up to a 256 x 256 crossbar.

The predetermined priority scheme implemented by the arbitration circuits during the arbitration mode of operation can take a variety of forms. In one embodiment, the predetermined priority scheme may be fixed. However, in an alternative embodiment, each crossbar cell further comprises priority storage circuitry programmable to store priority data identifying the actions to be performed by the associated arbitration circuitry in respect of one or more of the n bit lines in order to implement the predetermined priority scheme. This hence allows the predetermined priority scheme to be reprogrammed over time, thereby allowing the arbitration scheme to improve fairness of access to the crossbar circuitry by the various source circuits.

In one embodiment the crossbar circuitry further comprises a priority configuration module, which in a priority assignment mode of operation, inputs

priority data over the plurality of word lines of one or more of the data input paths, and asserts a priority load signal to the priority storage circuitry of one or more of the crossbar cells, such that any priority storage circuitry receiving an asserted load priority signal samples the data on at least one word line in order to update the priority data stored therein. Via this approach, the priorities associated with selected priority storage circuits can be updated, or alternatively the priority storage circuits of each crossbar cell can be reprogrammed in parallel.

In one embodiment, each data output path provides n bit lines, and each configuration storage circuitry is associated with one of said n bit lines via which the routing value is programmed into the configuration storage circuitry, such that for up to n crossbar cells associated with the same data output path, the configuration storage circuitry of those crossbar cells have a different associated bit line of said n bit lines. Within each crossbar cell the arbitration circuitry and the priority storage circuitry is associated with the $n-1$ bit lines other than the one bit line associated with the configuration storage circuitry of that crossbar cell, with the priority storage circuitry identifying which of those $n-1$ bit lines should have their voltages modified by the arbitration circuitry in order to implement said predetermined priority scheme.

In one particular embodiment, the priority storage circuitry may be formed of $n-1$ priority storage elements, one for each of the $n-1$ bit lines associated with the priority storage circuitry. The priority storage elements can take a variety of forms, but in one embodiment are formed of SRAM cells.

If the crossbar circuitry has less than $n \times n$ crossbar cells then it will be appreciated that there will be word lines and bit lines that do not get re-utilised in the arbitration mode as discussed earlier. For example, considering a 16×16 crossbar circuit, with 64-bit data input paths and data output paths, only a quarter of the word lines and bit lines will be used during arbitration using the earlier described approach. In such instances, it is possible to store multiple priorities at each crossbar cell, by providing each crossbar cell with a plurality of priority storage circuits. The priority configuration module can then program the multiple priority storage circuits in each crossbar cell using the word lines of the data input path. Subsequently, it is then possible to switch between the various preloaded priorities without needing to reload them, thereby allowing a single cycle switching between one priority scheme and

another priority scheme. For example, in the above given scenario, it is possible to store four priorities at each crossbar cell and selectively use any of them during arbitration. It is also possible to use the extra input lines in order to load all four priorities in one cycle during a priority assignment mode of operation.

The priority configuration module can operate in a variety of ways. For example, in one embodiment, it may act as a finite state machine (FSM) that periodically writes priorities into the priority storage circuits of the crossbar cells using the word lines. In one particular embodiment it can assign priorities in a round-robin scheme, but it will be appreciated that any other suitable priority scheme could be used.

However, in an alternative embodiment, the priority configuration module can be used to perform adaptive priority programming in order to seek to increase the overall efficiency of the crossbar circuitry. In particular, in one embodiment the plurality of source circuits are connectable to a first end of said plurality of data input paths, the priority configuration module is connectable to a second end of said plurality of data input paths, the source circuits are arranged to issue asserted transmission requests via the plurality of word lines of each data input path, and the priority configuration module is arranged to monitor the asserted transmission requests issued by each source circuit, and to modify the priority data to be output by the priority configuration module during the priority assignment mode of operation dependent on the number of asserted transmission requests from each source circuit. Accordingly, in such embodiments, the priority configuration module can monitor the number of asserted transmission requests sent to the crossbar in order to find out which data output paths are used most by specific source circuits. It can then use this information to assign priorities more intelligently, so that the overall efficiency of the crossbar circuitry increases. The efficiency can be defined as the proportion of the operation time that the crossbar circuitry is in the transmission mode of operation (i.e. the duration for which the bit lines are used to send data rather than to detect and resolve conflicts between transmission requests).

In one embodiment, each crossbar cell further comprises release circuitry coupled to the one bit line associated with the configuration storage circuitry of that crossbar cell, on receipt of an asserted release request the channel release circuitry causing the voltage on that one bit line to be modified to a level that causes said

second value to be stored as said routing value in the configuration storage circuitry of that crossbar cell. This provides a simple and effective mechanism for allowing a source circuit to release a particular data output path when it has finished transmitting data, thereby allowing a lower priority source circuit to subsequently seek to acquire that data output path.

In one particular embodiment, each of the bit lines are precharged to a first voltage level, and on receipt of said asserted release request, the release circuitry discharges the voltage on the one bit line associated with the configuration storage circuitry of that crossbar cell. Further, each configuration storage circuitry comprises a sense-amp enabled latch which, in the presence of an asserted release request to the associated release circuitry, senses the voltage on its associated bit line following the discharge operation performed by the release circuitry in order to update the routing value to said second value, thereby releasing the associated data output path. Hence, in such embodiments, to release a channel at a crossbar cell, the bit line that is sensed by the associated sense-amp enabled latch is discharged and subsequently sampled.

In one embodiment, each data input path provides n word lines, and the release requests for up to n crossbar cells associated with the same data input path are input to the crossbar circuitry via the n word lines of that same data input path. Further, each release circuitry is associated with one of said n word lines, such that for up to n crossbar cells associated with the same data input path, the release circuitry of those crossbar cells have a different associated word line of said n word lines via which the release request is received. Hence, in such embodiments, the release requests can be issued in an efficient manner through re-use of the word lines of the data input paths.

In one particular embodiment, the release requests are issued in said arbitration mode of operation, and a control signal is issued by the plurality of source circuits in the arbitration mode of operation to identify if the n word lines of the associated data input path are carrying release requests or transmission requests. Hence, in such embodiments the control signal can be used to qualify whether the information routed over the word lines identifies release requests or transmission requests, and hence whether the arbitration circuitry or the release circuitry needs to be activated. In one particular embodiment, the control signal actually takes the form of two separate signal lines, one being a request_channel signal and the other being a release_channel signal,

only one of these two signals being allowed to be set at any point in time. In one embodiment these two signals are provided on a per row basis, and hence within any particular row the crossbar cells will either be processing release requests or transmission requests at any point in time. In some implementations it would be possible to provide a finer granularity of control signals, so that some crossbar cells in a row could be processing release requests whilst others are processing transmission requests.

In one embodiment, when a crossbar cell has its routing value programmed to the first value during the arbitration mode of operation, a grant signal is asserted to the associated source circuit to confirm that the transmission request that it asserted has been granted. In one embodiment, a source circuit can only send a single request at a time, and in such an embodiment there will typically be only a single grant signal issued per row. However, in an alternative embodiment, a source circuit may be able to send multiple requests at a time, and in such embodiments multiple grant signals per row can be provided, with one grant signal being associated with each data output path.

The transmission circuitry can operate in a variety of ways. However, in one embodiment, each data output path associated with a crossbar cell in the transmission mode of operation is precharged to a first logic level prior to data transfer, the transmission circuitry of each crossbar cell comprising: first and second switches connected in series between the data output path and a second logic level; in the transmission mode of operation the first switch being open or closed dependent on the routing value stored in the associated configuration storage circuitry, and the second switch being open or closed dependent on the data input over the data input path. In one particular embodiment, the first logic level is the supply voltage level V_{dd} and the second logic level is ground. Hence, by such an arrangement, the data on the input data path does not directly drive the data on the output data path, and instead the data on the output data path either remains at the first logic level, or is discharged towards the second logic level in the event that both the first and second switches are closed.

By such an arrangement of transmission circuitry, the transmission circuitry does not need to be altered irrespective of the size of the crossbar circuitry and hence the length of the data output path, and further the drive circuitry providing the input data over

the input data path also does not need resizing as the crossbar circuitry is increased in size. Hence, if the crossbar circuitry is increased in size, the circuitry of each crossbar cell does not need to be changed and instead only the number of crossbar cells needs to increase. Thus, by using such a design, the delay of the crossbar circuitry grows linearly with size, making such a design of crossbar circuitry useable with very large crossbars (for example with 128 x 128 or 256 x 256 inputs/outputs). In addition, the design is very regular and the delay across all routes through the crossbar circuitry is uniform.

In one embodiment, each said data input path comprises n word lines for carrying an n -bit input data value during the transmission mode of operation, and each said data output path comprises n bit lines for carrying an n -bit data value during said transmission mode of operation, at least the second switch being replicated for each bit line. Accordingly, the design of the transmission circuitry can readily accommodate various sizes of data input paths and data output paths, without any significant increase in the complexity of the crossbar circuitry.

In one embodiment, in the transmission mode of operation, a bit line of the data output path is pulled to the second logic level if the routing value is said first value and the input data bit on the corresponding word line is at said first logic level. Hence, if the routing value stored in the configuration storage circuitry indicates that the data input path should be coupled to the data output path, and the data on the data input path is at a logic one level, then the data output path will be discharged towards the second logic level.

In one embodiment each crossbar cell further comprises conditional discharge circuitry for turning off the first switch irrespective of the routing value whilst the associated data output path is being precharged to the first logic level, and for allowing the first switch to be controlled by the routing value following the precharging of the associated data output path to the first logic level. Accordingly, such conditional discharge circuitry isolates the bit lines of the data output path from the inputs whilst they are being precharged, thereby reducing the power consumption of the precharge operation. Also, by such an arrangement, the data input paths can be driven simultaneously with the precharge operation, due to the conditional discharge circuitry isolating the data output paths from the inputs at that time, thereby enabling an increased speed of operation.

Furthermore, by using the arrangement of transmission circuitry described above, it should be noted that all bit lines of the data output path are precharged to the first logic level, and then either stay at the first logic level or transition to the second logic level dependent on the input value on the associated word line of the data input path and the routing value. As a result, no situation arises where the voltages on two adjacent bit lines are moving in opposite directions, thereby reducing capacitive coupling effects and thereby improving speed of operation.

In embodiments where the data output path is precharged prior to data transfer, and then selectively discharged dependent on the routing value and the data input value, power consumption of the crossbar circuitry can be reduced through the use of sense amplifier circuitry. In particular, in one embodiment, the crossbar circuitry further comprises sense amplifier circuitry to detect, during the transmission mode of operation, the data output on the bit lines of the data output path, thereby allowing detection of a transition towards the second logic level before the voltage on a bit line of the data output path has reached the second logic level. Since the detection of the transition occurs before a bit line of the data output path has reached the second logic level, the power required to precharge the bit lines of the data output path back to the first logic level is significantly reduced.

In embodiments where such precharging of the data output paths occurs, then further power savings can be obtained by suitable encoding of the input data prior to provision to the crossbar circuitry. In particular, in one embodiment, the crossbar circuitry further comprises: encode circuitry between each of said plurality of source circuits and said plurality of data input paths; decode circuitry between each of said plurality of data output paths and said plurality of destination circuits; the encode circuitry applying an encoding operation to encode input data provided by each source circuit into an encoded format that reduces the number of times the data output paths need to be pulled to the second logic level following their precharge to the first logic level, when compared to the number of times the data output paths would need to be pulled to the second logic level if the original input data were passed through the crossbar circuitry; the decode circuitry applying a corresponding decoding operation to identify the original input data provided by the source circuit from the encoded data output over the data output path.

In one particular embodiment, the encoding operation ensures that the encoded format produced from the input data is only a logic one value when the input data changes, and accordingly the data output path is only discharged when the input changes. The decode circuitry then recreates the original input data from the encoded data output over the data output path.

Due to the regular design of the crossbar circuitry of embodiments of the present invention, and the manner in which the transmission circuits of each crossbar cell operate, the plurality of source circuits are connectable to either end of the plurality of data input paths, thereby significantly improving flexibility when designing the layout of the apparatus in which the crossbar circuitry and source circuits are provided. Similarly, the plurality of destination circuits are connectable to either end of the plurality of data output paths.

In one embodiment, a sequence of word line drivers may be used to propagate data input values along the data input paths to the various crossbar cells connected to those data input paths. In a single request embodiment, where multi-casting is not supported, and accordingly each source will only request access to a single output path at a time, significant power may be consumed in propagating the data along the entirety of the data input path in situations where it is not needed. In one embodiment this power consumption is alleviated through the provision of a partially activated network arrangement. In particular, in one embodiment, each data input path consists of multiple input path portions separated by word line drivers, and each word line driver is activated in dependence on the routing values stored in the configuration storage circuitry of crossbar cells provided along the associated data input path but further from the source circuit than that word line driver. As a result, each word line driver is only activated if the configuration storage circuitry of at least one crossbar cell provided along the associated data input path but further from the source circuit than that word line driver has a routing value set to said first value. Such an approach can yield significant power savings.

The crossbar circuitry of embodiments of the present invention may be utilised in a variety of systems. However, according to a second aspect of the present invention, a data processing apparatus is provided comprising: a plurality of memory devices for storing data values; a plurality of processors for performing data processing operations in

parallel on multiple of the data values stored in said plurality of memory devices; and crossbar circuitry in accordance with the first aspect of the present invention for routing the data value from any of said plurality of memory devices to any of said plurality of processors.

The crossbar circuitry of embodiments of the present invention provides a particularly simple, scalable and power efficient mechanism for routing data values from any of the memory devices to any of the processors.

Viewed from a third aspect, the present invention provides crossbar circuitry for interconnecting a plurality of source means and a plurality of destination means such that data input to the crossbar circuitry from any of said plurality of source means can be output to any of said plurality of destination means, the crossbar circuitry comprising: a plurality of data input path means passing through said crossbar circuitry, each data input path means for connecting to one of said plurality of source means and providing a plurality of word line means; a plurality of data output path means passing through said crossbar circuitry transverse to the plurality of data input path means, each data output path means for connecting to one of said plurality of destination means and providing a plurality of bit line means; a crossbar cell means associated with each intersection between one of said data input path means and one of said data output path means, each crossbar cell means comprising: configuration storage means programmable for storing a routing value, the routing value being programmed to a first value to indicate that data input along the word line means of the data input path means to the associated intersection is to be output on the bit line means of the data output path means at the associated intersection, and the routing value being programmed to a second value to indicate that data input along the word line means of the data input path means to the associated intersection is not to be output on the bit line means of the data output path means at the associated intersection; transmission means, responsive to the routing value having said first value in a transmission mode of operation, for detecting the data input along the word line means of the data input path means and to output an indication of that data on the bit line means of the data output path means at the associated intersection; and arbitration means for operating in an arbitration mode of operation in dependence on a transmission request received by the crossbar cell means from the source means connected to the data input path means

of the associated intersection, if the transmission request is asserted to indicate that said source means wishes to route data from the data input path means to the data output path means at the associated intersection, the arbitration means for operating in combination with the arbitration means of other crossbar cell means associated with the same data output path means to re-use the bit line means of the data output path means to detect the presence of multiple asserted transmission requests for said same data output path means, and in the event of such multiple asserted transmission requests, for implementing a predetermined priority scheme to cause the configuration storage means of only one crossbar cell means associated with said same data output path means to have its routing value programmed to said first value, thereby resolving conflict between said multiple asserted transmission requests according to said predetermined priority scheme.

Viewed from a fourth aspect, the present invention provides a method of operating crossbar circuitry to interconnect a plurality of source circuits and a plurality of destination circuits such that data input to the crossbar circuitry from any of said plurality of source circuits can be output to any of said plurality of destination circuits, the crossbar circuitry having a plurality of data input paths passing through said crossbar circuitry, each data input path being connectable to one of said plurality of source circuits and providing a plurality of word lines, and a plurality of data output paths passing through said crossbar circuitry transverse to the plurality of data input paths, each data output path being connectable to one of said plurality of destination circuits and providing a plurality of bit lines, the method comprising the steps of: employing a crossbar cell in association with each intersection between one of said data input paths and one of said data output paths; programming a routing value in each crossbar cell, the routing value being programmed to a first value to indicate that data input along the word lines of the data input path to the associated intersection is to be output on the bit lines of the data output path at the associated intersection, and the routing value being programmed to a second value to indicate that data input along the word lines of the data input path to the associated intersection is not to be output on the bit lines of the data output path at the associated intersection; in a transmission mode of operation, causing the crossbar cell to be responsive to the routing value having said first value to detect the data input along the word lines of the data input path and to

output an indication of that data on the bit lines of the data output path at the associated intersection; and in an arbitration mode of operation, causing arbitration circuitry in the crossbar cell to operate in dependence on a transmission request received by the crossbar cell from the source circuit connected to the data input path of the associated intersection, if the transmission request is asserted to indicate that said source circuit wishes to route data from the data input path to the data output path at the associated intersection, the arbitration circuitry operating in combination with the arbitration circuitry of other crossbar cells associated with the same data output path to re-use the bit lines of the data output path to detect the presence of multiple asserted transmission requests for said same data output path, and in the event of such multiple asserted transmission requests to implement a predetermined priority scheme to cause the configuration storage circuitry of only one crossbar cell associated with said same data output path to have its routing value programmed to said first value, thereby resolving conflict between said multiple asserted transmission requests according to said predetermined priority scheme.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be described further, by way of example only, with reference to embodiments thereof as illustrated in the accompanying drawings, in which:

Figure 1 is a block diagram of crossbar circuitry in accordance with one embodiment;

Figure 2 illustrates how collisions may be detected and resolved in accordance with one embodiment;

Figure 3 illustrates how priority storage circuitry provided in association with each crossbar cell may be used in one embodiment;

Figure 4 illustrates how release circuitry may be used in one embodiment to enable a crossbar cell to release a particular data output path (also referred to herein as a channel);

Figure 5 illustrates the circuitry provided within each crossbar cell for a number of crossbar cells in a column;

Figure 6 illustrates the circuitry provided within each crossbar cell for a number of crossbar cells in a row;

Figure 7 illustrates in more detail the transmission circuitry and arbitration circuitry that may be provided within each crossbar cell in accordance with one embodiment;

Figure 8 illustrates in more detail the transmission circuitry and release circuitry that may be provided within each crossbar cell in accordance with one embodiment;

Figure 9 schematically illustrates a packet switching crossbar implementation in accordance with one embodiment;

Figures 10A to 10D illustrate in more detail components provided within each crossbar cell of the crossbar circuit in accordance with one embodiment;

Figures 11A to 11D illustrate in more detail the components provided within each crossbar cell of the crossbar circuit in accordance with an alternative embodiment;

Figure 12 illustrates in more detail the arrangement of the configuration storage element provided within each crossbar cell in accordance with one embodiment;

Figures 13A to 13C illustrate the configuration of a pulse generator for generating discharge and sense enable (SE) signals in accordance with one embodiment;

Figure 14 illustrates how a `channel_free` signal is generated in accordance with one embodiment;

Figure 15A illustrates the operation of the priority configuration module of Figure 1 in accordance with one embodiment;

Figure 15B illustrates the operation of the priority configuration module of Figure 1 in accordance with an alternative embodiment;

Figures 16A to 16E illustrate encoding and decoding circuitry that may be used in one embodiment to encode the input data prior to input into the crossbar circuitry, and to decode the output data from the crossbar circuitry, in order to reduce power consumption within the crossbar circuitry;

Figure 17 illustrates how crossbar circuitry with narrow data input paths and data output paths may be arranged to operate in accordance with one embodiment;

Figures 18A and 18B illustrate how crossbar circuitry with narrow data input paths and data output paths may be arranged to operate in accordance with an alternative embodiment;

Figure 19 illustrates how crossbar circuitry having wide data input paths and data output paths may operate in accordance with one embodiment; and

Figure 20 illustrates how crossbar circuitry may be used in a partially activated configuration in accordance with one embodiment.

DESCRIPTION OF EMBODIMENTS

Figure 1 illustrates a top-level diagram of a proposed switching fabric, comprising a crossbar circuitry in accordance with one embodiment and a plurality of source circuits and destination circuits connected thereto. The crossbar circuitry includes a plurality of data input paths 12 running in a first direction through the crossbar circuitry, and a plurality of data output paths 50 running in a second direction transverse to the first direction. In the embodiment of Figure 1, both of these paths are formed by multi-bit buses, and in particular each data input path comprises a plurality of word lines and each data output path comprises a plurality of bit lines.

At the intersection between each data input path and data output path, a crossbar cell 20 is provided, which is used to selectively route data received on the associated data input path out on the associated data output path. Each crossbar cell includes a configuration storage element for storing the on/off status of each crossbar cell, such that if the crossbar cell is in the on state, then it will route input data received on the data input path to the data output path, and if the cell is in the off state, the input data will merely be passed through the cell without being routed onto the associated data output path.

Each data input path 12 is coupled to an associated source circuit 30 from which input data 10 may be received, and each data output path 50 is coupled to an associated destination circuit 40 arranged to receive the output data 70 provided on that data output path.

In the embodiment shown, the bit lines of each data output path 50 are precharged using a pre-charge module 55, and then the bit lines are selectively discharged dependent on the data input from one of the source circuits to a crossbar cell connecting that source circuit's data to the data output path. This selective discharge of the bit lines is detected by the sense amplifier circuitry 65, in order to produce the output data 70.

Whilst data is routed through the crossbar circuitry in the above described manner when the crossbar is in a transmission mode of operation, the crossbar circuitry will not always be in the transmission mode of operation. In particular, the crossbar circuitry has an arbitration mode of operation during which it detects and resolves any conflicts between transmission requests issued by the various source circuits 30. In particular,

multiple requests for the same destination circuit in a switching fabric is termed a collision, and with increasing numbers of sources and destinations, collisions become more frequent. As will be discussed in more detail later, each crossbar cell 20 includes arbitration circuitry which, in the arbitration mode of operation, operates in combination with arbitration circuitry of other crossbar cells, re-using the bit lines of the data output paths to detect the presence of multiple asserted transmission requests and, in the event of such multiple asserted transmission requests, to implement a predetermined priority scheme to resolve those conflicts. The proposed approach provides a fast, low-power and highly scalable solution to detect and resolve such collisions.

Whilst the predetermined priority scheme may be fixed, in one embodiment each crossbar cell includes priority storage circuitry which can be programmed by a priority configuration module 60 in a priority assignment mode of operation. In the priority assignment mode of operation, the word lines of the data input paths are used to re-program the various priority storage circuits within the crossbar cells. Hence, via the word lines, the priority assignment mode of operation can be used to reprogram the predetermined priority scheme, to allow for a more fair distribution of access to the various destination circuits from the various source circuits via the crossbar circuitry.

Whilst in Figure 1 the source circuits are shown to the left hand side of the crossbar and the destination circuits are shown at the bottom of the crossbar, the source circuits can be provided at either side of the crossbar circuitry, and similarly the destination circuits can be provided at either end of the data output paths. Hence, it will be appreciated that the crossbar circuitry of embodiments of the present invention offers the flexibility of placing sources at either horizontal ends and destinations at either vertical ends. This simplifies design floor planning by reducing routing congestion.

Figure 2 illustrates how the bit lines of the data output paths may be re-used to detect and resolve conflicts during the arbitration mode of operation. In this example, there are 16 source circuits connected to the crossbar circuitry, that may issue transmission requests 0 to 15. For any particular data output path 50, each source circuit will be associated with a crossbar cell 20 which incorporates within it configuration storage circuitry in the form of a sense amp and latch. Hence, source 0 is associated with the crossbar cell containing the sense amp and latch 100, source 1 is associated with the crossbar cell containing the sense amp and latch 105, source 2 is associated with the

crossbar cell containing the sense amp and latch 110, and so for the remainder of the source circuits, such that source circuit 15 is associated with the crossbar cell containing the sense amp and latch 115.

In the arbitration mode of operation, the precharge module 55 precharges all of the bit lines of each data output path 50, whereafter those bit lines are selectively discharged dependent on the asserted transmission requests and in accordance with a predetermined priority scheme. In this example, each data output path 50 comprises 16 bit lines, and it can be seen that each of the sense amp and latch circuits are associated with a different one of those bit lines. Following the precharge of the bit lines, the transmission requests asserted by the various source circuits are evaluated, an asserted transmission request in this embodiment taking a logic one value.

The asserted transmission requests first cause the selective discharge of bit lines via the various transistors 107, 112, 114, 117, 119, 121, 123. As can be seen, in this example embodiment source 15 has the highest priority, such that if it asserts a transmission request, this causes the discharge of all of the bit lines other than the bit line of its associated sense amp and latch circuit 115. Source 14 has the next highest priority, source 13 the next highest priority and so on, with source 0 having the lowest priority. In particular, as can be seen, if the transmission request from source 0 (i.e. req 0) is asserted, this does not cause any bit lines to be discharged. Following the selective discharge operation, then any sense amp plus latch circuit that receives an asserted transmission request samples its associated bit line in order to store a routing value that will determine whether the associated crossbar cell will then, in the transmission mode of operation, couple its input to its output. In particular, only if the routing value has a first value (in one embodiment this being a logic one value) will the associated crossbar cell connect its input data to its output during the transmission mode of operation, and by virtue of the above described scheme it will be appreciated that only one crossbar cell will have its sense amp and latch circuitry set to a logic one value at the end of the arbitration mode of operation.

Hence, by way of example, if both source 1 and source 2 issue asserted transmission requests, it will be seen that both asserted request 1 and asserted request 2 cause the first bit line to be discharged via the transistors 107, 112. As a result, even if source 0 asserts a request, its request will not be granted, since the associated sense amp

and latch circuitry 100 will latch a logic zero value at the end of the arbitration mode of operation. Further, the asserted request 2 causes the second bit line to be discharged via the transistor 114, such that at the end of the arbitration mode of operation, the sense amp and latch circuitry 105 associated with source 1 will latch a logic zero value. Hence, even though source 1 asserted a transmission request, that transmission request is not granted, given that source 2 has also issued an asserted transmission request. Assuming no other source circuits assert a transmission request in that cycle, then it will be seen that the sense amp and latch circuitry 110 associated with source circuit 2 will latch a logic one value at the end of the arbitration mode of operation, and accordingly the transmission request from source 2 will be granted.

From the above description, it will be seen that incoming asserted transmission requests cause the suppression of other asserted transmission requests from lower priority sources. The sense amplifier and latch circuits for any asserted transmission request then sample their associated bit lines to determine if the request is granted or not. By virtue of this technique, collisions can be detected and resolved in a single cycle. Further, it can be seen that the same bit lines that are used during the arbitration mode of operation to detect and resolve conflicts are then used for the transmission of data in the following transmission mode of operation.

Whilst the asserted transmission requests can be input into the crossbar circuitry in a number of different ways, in one embodiment the word lines of the data input paths are used to input the asserted transmission requests. Hence, considering embodiments where each data input path has 16 word lines, it will be seen that any particular source circuit can specify via those word lines which of up to 16 data output paths it wishes to assert a transmission request for, in the arbitration mode of operation each of the individual word lines being associated with a different data output path.

Whilst not explicitly shown in Figure 2, there will typically be some isolation circuitry associated with the various transistors 107, 112, 114, 117, 119, 121, 123 to ensure that during the precharge mode of operation those transistors are turned off irrespective of the values of the various request inputs provided to their gates.

Whilst in Figure 2 a fixed order to priority is used for arbitration, in one embodiment the priorities can be stored at the crossbar cells through the provision of priority storage circuitry, as illustrated schematically in Figure 3. In particular, as shown

in Figure 3, each crossbar cell includes a sequence of transistors 135 associated with each of the bit lines other than the bit line connected to the sense amp and latch circuitry of that crossbar cell. Priority storage elements 130 are then used to provide a value to the gate of those transistors 135, in dependence on a value programmed into those priority storage elements. As shown schematically in Figure 3, if the same priority scheme is desired as shown by the fixed arrangement of Figure 2, then all of the priority storage elements 130 shaded dark will be programmed to a logic one value and all of the priority storage elements 130 shaded light will be programmed to a logic zero value. Hence, transistors attached to priority storage elements storing a logic one value discharge the bit lines while those attached to priority storage elements storing a logic zero value do not.

The priority configuration module 60 of Figure 1 can be used to program the individual priority storage elements 130 in the priority assignment mode of operation. In this mode, the word lines are used to assign a new priority scheme, and in particular fifteen of the sixteen word lines can be used to program the fifteen priority storage elements forming the priority storage circuitry of each crossbar cell.

Using the technique illustrated schematically in Figure 3, all sixteen factorial (16!) permutations per data output path can be programmed into the priority storage circuitry associated with each data output path (each data output path also being referred to herein as a channel).

In addition to the priority storage elements 130 shown in Figure 3, there will also typically be some associated isolation circuitry for each transistor 135, to ensure that the transistors 135 are turned off during the precharge operation performed by the precharge module 55.

In one embodiment, each crossbar cell is provided with release circuitry to provide a mechanism for releasing a channel when it is no longer required. In particular, once a source circuit 30 has been granted access to a channel, only that source circuit can release the channel for subsequent allocation to a different source circuit, and this releasing of the channel is performed by the release circuitry in response to an asserted release request from the source circuit currently granted access to that channel. While a channel is granted to a particular source circuit, no other source circuit can gain access to that channel, and any asserted transmission requests in respect of that channel issued by other source circuits will not be actioned whilst the channel is allocated to a source circuit.

Figure 4 illustrates the release circuitry 140 that is added in accordance with one embodiment. In a similar way to the priority storage elements, the release circuitry 140 is used to drive an associated transistor 145, and in this case the transistor is coupled to the bit line that is sampled by the sense amp and latch circuitry of the same crossbar cell to determine the routing value. Hence, when a source circuit wishes to release a channel, it sends a release request to the release circuitry 140, which causes a logic one value to be output to the transistor 145, causing the associated bit line to discharge. Thereafter, the associated sense amp and latch samples the bit line, to thereby store a logic zero value therein, thus releasing the channel. Again, some associated isolation circuitry is provided for each transistor 145, to ensure that the transistor 145 is turned off during the precharge operation performed by the precharge module 55.

In one embodiment, the release requests are issued in the arbitration mode of operation, and a control signal is issued by the source circuits in the arbitration mode of operation to identify if the word lines of the associated data input path are carrying release requests or transmission requests. Hence, again the word lines of the data input paths can be re-used, this time to carry asserted release requests to the associated release circuits 140, thereby providing a particularly efficient mechanism for releasing channels that have previously been granted to a particular source circuit.

Figure 5 illustrates the main components provided within each crossbar cell, and in particular shows the crossbar cells provided for column 0 in respect of rows 0, 1 and 2. Hence, the crossbar cell 200 is provided at the intersection between row 0 and column 0, the crossbar cell 230 is provided at the intersection between row 1 and column 0, and the crossbar cell 260 is provided at the intersection between row 2 and column 0.

Considering first the crossbar cell 200, this consists of a configuration storage circuit 205, transmission circuitry 210, channel release circuitry 215, arbitration circuitry 220 and priority storage circuitry 225. In the arbitration mode of operation, transmission requests are asserted over the data input path (also referred to in Figure 5 as the input bus), and accordingly if source 0 wishes to assert a transmission request for column 0, it does so by setting bit 0 of the input bus. As can be seen, the value of bit 0 is input to AND gate 207, whose other input is driven by the request channel control signal. A source circuit will assert the request channel control signal if it is issuing any asserted transmission requests over the input bus during the arbitration mode of operation.

Accordingly, if source 0 has issued an asserted transmission request for channel 0, both inputs to the AND gate 207 will be at a logic one value, and accordingly both the arbitration circuitry 220 and the configuration storage circuitry 205 will be activated (the configuration storage circuitry being activated via OR gate 208).

As can be seen, the configuration storage circuitry 205 is connected to bit 0 of the data output path for channel 0 (also referred to in Figure 5 as the output bus), and at the end of the arbitration mode of operation will sample the value on bit 0 of the output bus in order to determine the routing value to be stored therein. However, prior to that point, the arbitration circuitry 220 will operate in association with the arbitration circuitry 250, 280, etc. of any other crossbar cells associated with that column that have received asserted transmission requests, in order to selectively discharge bit lines of the output bus. In particular, as can be seen in Figure 5, the arbitration circuitry 220 of the crossbar cell 200 is connected to all of the other bit lines of the output bus other than bit line 0, and will selectively discharge those bit lines dependent on the predetermined priority scheme programmed into the priority storage circuitry 225.

As mentioned above, following the selective discharge operation of the arbitration circuitry 220, the configuration storage circuitry 205 will then sample the value on its associated output bit line, and in particular if that bit line is still at the precharge voltage level, it will store a logic one routing value within the configuration storage circuitry 205 to indicate that source 0 has been granted access to channel 0. Due to the nature of the predetermined priority scheme implemented by the arbitration circuits of the crossbar cells in the column, only one crossbar cell for any particular column will have its configuration storage circuit set to a logic one value at the end of the arbitration mode of operation, and accordingly only one source circuit can be granted access to a particular destination circuit at any point in time.

Following the arbitration mode of operation, the switching fabric enters the transmission mode of operation, during which time each source circuit that has been granted access to a particular channel provides its input data over the input bus. Accordingly, assuming source 0 was granted access to column 0, it will issue its data over the input bus to the transmission circuitry 210, which given the logic one value stored in the configuration storage circuitry 205, will then connect its input to the output bus for channel 0.

As discussed previously, in addition to the arbitration mode of operation and the transmission mode of operation, a priority assignment mode of operation is also provided. The priority assignment mode of operation is intended to be used less frequently, but allows the priority scheme as encoded in the priority storage circuitry 225, 255, 285 of each crossbar cell to be modified, to allow fairness in the arbitration scheme over time. When in the priority assignment mode of operation, priority data can be provided over the input bus, and a load priority signal is issued to the priority storage circuitry of each crossbar cell to cause it to sample the values on the input bus. As will be apparent from the earlier discussion of Figures 3 and 4, considering a 16 x 16 array of crossbar cells with 16-bit data input and data output paths, the priority storage circuitry of each crossbar cell will actually consist of fifteen priority storage elements, which can be programmed up using fifteen word lines, in the example of Figure 5 it being assumed that word lines 1 to 15 are used for this purpose.

As also mentioned previously, in the arbitration mode of operation it is also possible to release channels when they have previously been granted to a particular source. Hence, by way of example, if source 0 has previously been granted access to channel 0, and accordingly the configuration storage circuitry 205 has a logic one routing value stored therein, then in the arbitration mode of operation, the source 0 can issue a release channel control signal to the AND gate 217, along with a logic one value on bit 0 of the input bus to specify an asserted release request for channel 0. This will cause a logic one value to be input to the channel release circuitry 215, and the configuration storage circuitry to be activated via OR gate 208. In response to this logic one value, the channel release circuitry discharges bit 0 of the output bus, and then the configuration storage circuitry 205 re-samples the output bus, thereby causing a logic zero value to be stored within the configuration storage circuitry 205.

In one embodiment, both the request channel control signal and the release channel control signal can be encoded in a two bit signal, with that two bit signal specifying during the arbitration mode of operation whether the associated source circuit is issuing any asserted transmission requests, is issuing any asserted release channel requests, or is not issuing any asserted requests.

The crossbar cells 230 and 260 are constructed in an identical way to the crossbar cell 200 and hence the elements 235, 237, 238, 240, 245, 247, 250 and 255 of the

crossbar cell 230, and the elements 265, 267, 268, 270, 275, 277, 280 and 285 of the crossbar cell 260 correspond with the elements 205, 207, 208, 210, 215, 217, 220 and 225 of the crossbar cell 200, respectively. However, as is clear from Figure 5, the way in which those various elements are connected to the various word lines and bit lines is slightly different. Since each of the crossbar cells 200, 230, 260 is associated with column 0, then the asserted transmission requests and asserted release requests are always provided on bit 0 of the associated input bus, and accordingly all three crossbar cells are connected to bit 0 (word line 0) for this purpose. Similarly, in all three cells the priority information provided in the priority assignment mode of operation can be provided by bits 1 to 15 of the input bus. However, considering the connections to the bit lines, it will be seen that whilst the configuration storage circuitry 205 of crossbar cell 200 is connected to bit 0 of the output bus, configuration storage circuitry 235 of crossbar cell 230 is connected to bit 1 and configuration storage circuitry 265 of crossbar cell 260 is connected to bit 2. The channel release circuitry of each crossbar cell is connected to the same bit line as the associated configuration storage circuitry. Meanwhile the arbitration circuitry is connected to all of the bit lines other than the bit line connected to the associated configuration storage circuitry. The transmission circuitry is of course connected to all of the word lines and bit lines, since in the transmission mode of operation, if the routing value in the associated configuration storage circuitry is set to a logic 1 value, then the transmission circuitry will connect the data on the input bus to the associated output bus.

Figure 6 is a similar diagram to Figure 5, but shows the crossbar cells 200, 300, 330 provided at the intersection between row 0 and column 0, column 1 and column 2, respectively. The crossbar cell 200 is as discussed earlier with reference to Figure 5, and the crossbar cells 300, 330 are constructed in the same way. Accordingly, the elements 305, 307, 308, 310, 315, 317, 320 and 325 of crossbar cell 300 and the elements 335, 337, 338, 340, 345, 347, 350 and 355 of crossbar cell 330 correspond with the elements 205, 207, 208, 210, 215, 217, 220 and 225 of crossbar cell 200, respectively. In this example, since all of the crossbar cells are associated with the same row, they are connected to the bit lines of their respective output bus in an identical manner. However, when considering the input bus, the asserted transmission requests or release requests for channel 0 are input on word line 0 whilst asserted transmission or release requests for

channel 1 are input on word line 1 and asserted transmission or release requests for channel 2 are input on word line 2. Accordingly, the configuration storage circuit, channel release circuit and arbitration circuit of each crossbar cell are driven by different input word lines accordingly. However, in the embodiment shown, the priority storage circuits can all be programmed from the same word lines 1 to 15 during the priority assignment mode of operation.

Figure 7 illustrates in more detail how transmission circuitry and arbitration circuitry are connected to each bit line. In particular, element 420 shows that part of the transmission circuitry associated with an individual bit line and element 440 shows that part of the arbitration circuitry associated with an individual bit line.

Considering a 16-bit wide channel, fifteen of the sixteen bit lines will be connected in the manner shown in Figure 7 since arbitration circuitry will be associated with all of the bit lines other than the bit line to which the configuration storage circuitry 410 of that crossbar cell is connected. The configuration storage circuitry 410, which is provided once for the crossbar cell, receives a store configuration signal which is set in the presence of an asserted transmission request or an asserted release request to that crossbar cell, and responsive to that set store configuration signal, samples as its configuration the value on a particular bit line to which it is connected, as discussed earlier with reference to Figures 5 and 6. This takes place during the arbitration mode of operation, and if at the end of the arbitration mode of operation this results in a logic one routing value being stored in the configuration storage circuitry 410, then when the crossbar circuitry subsequently enters the data transmission mode of operation, it will be seen that the transistor 422 of the transmission circuitry 420 is turned on. Accordingly, if the data input on the corresponding word line is also at a logic one value, this will cause the transistor 424 to turn on and will accordingly cause the bit line 426 to discharge, this bit line having previously been precharged to a logic one level by the precharge circuitry 400. Any discharge of the bit line 426 will be sensed by the sense amp circuitry 430.

In the arbitration mode of operation, the transmission circuitry is not used, and instead the arbitration circuitry 440 is used to selectively discharge the bit line 426 dependent on an input received by the transistor 442 from AND gate 455. In particular, if the priority storage element 450 stores a logic one value, and an asserted transmission request is received over the corresponding word line, then a logic one value will be input

to the transistor 442, causing the transistor 426 to discharge. Towards the end of the arbitration mode of operation, any configuration storage circuitry connected to that bit line 426 in another crossbar cell that has received an asserted transmission request will sample the value on that bit line in order to determine the value of the routing value applicable to that crossbar cell.

A separate priority storage element 450 will be provided for each associated arbitration element 440, with its value being programmable in the presence of a load priority signal asserted from the priority configuration module 60 during the priority assignment mode of operation. In particular, in the presence of an asserted load priority signal, the priority storage element 450 samples the value on an associated word line in order to reprogram the priority value stored therein.

Figure 8 illustrates the transmission circuitry and associated release circuitry in accordance with one embodiment. Considering a 16-bit channel, one bit line will be connected in this manner within each crossbar cell. The operation of the transmission circuitry is unchanged from that discussed with reference to Figure 7, and accordingly will not be discussed further herein. However, the release circuitry 460 is connected to this bit line 462 instead of the arbitration circuitry 440 of Figure 7. It will also be noted that the configuration storage circuitry 410 of the crossbar cell is also connected to this bit line 462. The AND gate 470 corresponds with the AND gates shown feeding the channel release circuits in Figures 5 and 6, and hence if the release channel control signal is asserted and the bit on the associated input word line is asserted, this indicates the presence of an asserted release request for the particular channel, and as can be seen from Figure 8 results in the transistor 465 being turned on to discharge the bit line 462. Subsequently the configuration storage circuitry 410 will be caused to receive an asserted store configuration signal which will cause it to resample the value on the bit line 462, thereby causing a logic zero routing value to be stored therein, thus releasing the channel.

Using the above described techniques, it is possible to implement a packet switching crossbar as shown schematically in Figure 9. In a packet switching environment, a source first sends a request, and then on receipt of a grant signal sends data, with the request and data being sent over the same input lines. From the above discussion of embodiments of the present invention, it can be seen that the crossbar circuitry of embodiments of the present invention lends itself readily to such a packet

switching embodiment, by allowing asserted transmission requests to be input over the same word lines that subsequently, in the transmission mode of operation, will be used for carrying data. The request channel and release channel control signals discussed earlier with reference to Figures 5 and 6 can be input via a 2-bit request/release input line 500, with the asserted transmission requests or asserted release requests being input over the associated data input paths 12. When a particular configuration storage circuit in a crossbar cell stores a logic one value during the arbitration mode of operation to indicate that the associated source circuit has been granted, this causes a grant signal to be returned to the source circuit over line 505. In a single request implementation, where any particular source circuit is only able to request one output channel at any point in time, only a single grant signal line 505 is required. However, in an alternative embodiment supporting multi-casting (where one source circuit can broadcast data over multiple channels at any point in time), then the source circuit will be able to issue multiple asserted transmission requests at any point in time, and in such embodiments a multi-bit grant signal line 505 can be provided to identify which channels the master has been granted access to.

During the arbitration mode of operation, each of the crossbar cells also receives a channel free signal over lines 510, indicating whether the associated channel is free to be allocated to a requesting source, the use of this signal being discussed later with reference to Figures 10 and 11. When it is desired to change the priority scheme used by the crossbar circuits, a load priority signal can be issued over paths 515 during the priority assignment mode of operation, to cause the priority storage circuitry within each crossbar cell to store new priority scheme data as input over the word lines during that mode of operation.

It will hence be appreciated from Figure 9, and the earlier discussion of embodiments of the present invention, that in this packet switching crossbar implementation, a precharge and conditional discharge scheme is used to transmit data through the crossbar circuitry during a transmission mode of operation. During a preceding arbitration mode of operation, the bit lines are re-used for collision detection and resolution, with the word lines being used to transmit asserted transmission requests. During a priority assignment mode of operation, the word lines are also re-used to send priority data for storing in individual priority storage elements within the crossbar cells 20.

A 2-bit request/release signal can be used to specify whether transmission requests or release requests are being asserted during a particular arbitration mode of operation. Assuming transmission requests are being asserted, then the grant signal is used to indicate to the source whether its request succeeded in acquiring the requested channel.

The existing word lines (input buses) and bit lines (output buses) are used to realise all of the above mentioned functionalities, thereby allowing all of these functionalities to be achieved with minimum routing overhead.

Figures 10A to 10D illustrate in more detail the circuitry provided within each crossbar cell 20 in accordance with a first embodiment, where a source circuit may only issue a single transmission request at a time. As is apparent from the earlier discussion of Figures 7 and 8, for each bit line of the output path, a pair of transistors 710, 715, 720 are provided to form the transmission circuitry, with one of those transistors receiving at its input the value on a corresponding word line, and the other transistor receiving at its input the routing value in the associated configuration storage element 700. During a precharge stage of the operation, a discharge signal will be set to a logic zero value, causing the conditional discharge circuitry 705 to isolate those transistors of each transmission circuit 710, 715, 720 from the contents of the configuration storage circuit 700. However, when the discharge signal goes high to indicate that the precharge operation has finished, and the sensing operation has begun, it can be seen that the configuration storage element 700 now provides a value which is used to drive the second transistor of each transmission circuit 710, 715, 720, causing the selective discharge of the associated bit line dependent on the input data received at the first transistor. The conditional discharge circuitry 705 also uses a WL_b signal, and Figure 10B illustrates how that WL_b signal is generated, the circuitry 765 of Figure 10B being provided once per crossbar cell.

As shown on the right hand side of Figure 10A, one of the bit lines will be connected to release channel circuitry formed by the AND gate 732 and the associated transistor 712, whilst all of the other bit lines will be connected to arbitration circuitry and associated priority storage elements 717, 742 and 722, 752, respectively. Each of the components 730, 740 and 750 act in combination with the component 760 to provide conditional discharge circuitry which isolates the associated transistors 712, 717, 722 during the precharge mode of operation. In particular, during the precharge mode of

operation, the discharge signal will be low causing the output from block 760 to be low irrespective of the value of the WL signal, which in turn causes the output of the components 730, 740 and 750 to be low, thereby turning off the transistors 712, 717, 722.

However, when the discharge signal goes high, then if the WL signal goes high, this will cause a logic one value to be output from the component 760. Hence, considering the arbitration circuitry and associated priority storage elements 717, 742 and 722, 752, in the event of an asserted transmission request in a situation where the associated storage element 742, 752 also stores a logic one value, this will cause the associated transistor 717, 722, respectively, to discharge, thereby implementing the priority scheme discussed earlier. However, if either of the priority storage elements stores a logic zero value, or the WL signal is not asserted because an asserted transmission request is not present, then the corresponding transistor 717, 722 will not discharge.

Similarly, considering the release circuitry 712, 732, in the presence of an asserted release request, where both the release signal and the input 0 word line is set high, this will cause the release transistor 712 to discharge, whereafter the bit line will be resampled by the configuration storage element 700 to reset the routing value to zero.

Figure 10B illustrates circuitry which is used to qualify an asserted request channel or release channel signal. It produces the WL signal, which is set high only if a request/release channel signal is qualified. This WL signal is then used to activate the configuration storage element 700. The same WL (and WL_b) signal is also used to appropriately operate the conditional discharge circuitry 705 and the arbitration circuitry 760.

Figure 10C illustrates the circuitry provided in association with the sense amp and latch circuitry 775 of the configuration storage element 700 in order to cause it to update its stored value, and in order to generate a grant signal dependent on the updated value. In particular, if a sense enable (SE) signal is set and the WL signal is set, this causes the AND gate 770 to output a set QSE (qualified SE) signal. If the QSE signal is set, this causes the sense amp and latch circuitry 775 to sample the current value on the corresponding bit line, in this example the sense amp and latch circuitry being connected to bit line 0.

A grant signal is generated from the current contents of the sense amp and latch circuitry 775. In particular, based on the current contents and the WL signal, the NAND gate 780 outputs a signal which is routed to the isolation circuitry 785. The isolation circuitry 785 ensures that only one crossbar cell in a particular row will drive a grant signal. In particular only the crossbar cell associated with the column that the source circuit is currently seeking to request can give rise to the grant signal. Hence, in this example, if the transmission request is set high and the input on word line 0 is set high, this will cause a logic one value to be provided to the lower transistor in the isolation circuitry 785 and a logic zero value to be provided to the higher transistor, thereby turning on the inverter and causing a grant signal to be propagated (the grant signal being asserted if the sense amp and latch circuit 775 stores a logic one value). However, otherwise a logic zero value is provided to the lower transistor and a logic one value to the upper transistor, thereby causing the inverter to be turned off and no grant signal to be propagated.

Figure 10D illustrates the structure of each priority storage element 742, 752 in accordance with one embodiment. In this embodiment, it can be seen that each priority storage element is provided by an SRAM cell 790, in the particular example of Figure 10 the SRAM cell being a 6T SRAM cell. When the load priority signal is set high in the priority assignment mode of operation, then the SRAM cell is caused to sample data provided on a corresponding input word line to thereby update the priority data stored therein.

Figures 11A to 11D illustrate the circuitry provided within each crossbar cell in accordance with an alternative embodiment of the present invention, where each source may issue multiple requests simultaneously. This allows a more efficient implementation for performing multi-casting within the crossbar. In particular, in the embodiment of figures 10A to 10D, a source circuit can only request one channel at a time, and so will require multiple arbitration cycles in order to acquire multiple channels if multi-casting is required. However, with the embodiment of figures 11A to 11D, a source circuit can issue requests for multiple channels at the same time and can potentially acquire multiple channels in one arbitration cycle, thereby enabling multi-casting to be achieved more efficiently.

Figures 11A, 11B and 11D are identical to Figures 10A, 10B and 10D, and accordingly will not be discussed further herein. However, the structure of the configuration storage circuitry is significantly simplified, as shown in Figure 11C, since there is no need for any isolation circuitry. Instead, a separate grant line is provided in association with each column and hence the current contents of the sense amp and latch circuitry 775 can be used to directly produce the grant signal in the presence of a set WL signal. In particular, if the WL signal is set, the source circuit has issued an asserted transmission request, and the current contents of the sense amp and latch circuitry 775 are set to a logic one value indicating the transmission request has been successful, then a logic one grant signal is issued back to the source circuit.

Figure 12 illustrates the arrangement of the configuration storage circuitry (i.e. the sense amp and latch circuitry) provided within each crossbar cell in accordance with one embodiment. The QSE signal generated by the AND gate 770 in Figures 10C and 11C is passed via the inverter 850 to produce the QSE_b signal. Accordingly, when the QSE signal goes high, the PMOS transistor 800 is opened to start the sensing operation of the sense amp, the sense amp being formed by the transistors 805, 810, 815, 820. At the start of the sensing operation, the transistors 815 and 820 are turned off. If the input line routed to the transistor 805 starts to discharge, this will cause the transistor 805 to turn on, pulling the intermediate node I towards Vdd. This in turn will cause the transistor 810 to turn on creating a positive feedback loop within the sense amp after a short period of time, at which point the sense amp is no longer reliant on the input signal transitioning towards ground in order to generate its output.

The series of transistors 825, 830, 835 acts as a transfer mechanism between the sense amp and the latch constructed of the components 840 and 845. In particular, during the sensing stage of the operation, the transistor 830 turns on, allowing the contents of the latch to be dictated by the value at the intermediate node I of the sense amp. In particular, it will be seen that if the input to the sense amp is transitioned towards the logic zero value, this will cause the intermediate node I to transition towards Vdd, thereby causing the transistor 835 to turn on in order to store a logic zero value within the latch. Conversely, if the input line does not discharge, the voltage at the intermediate node remains at a logic zero level, thereby turning on the transistor 825 and causing a logic one value to be stored within the latch.

It will be appreciated that the latch only updates its input during a period when the QSE signal is high, since as soon as the QSE signal goes low, the transistor 830 is turned off, thereby isolating the latch from the sense amp output.

Figures 13A to 13C illustrate the pulse generator circuitry used in one embodiment to generate both the SE and discharge signals. Figure 13A shows how the discharge signal is produced from the SE signal after some delay introduced by the component 860. In effect the circuitry of Figure 13A acts as an AND gate, ANDing together the SE signal with a delayed version of the SE signal.

Figure 13B illustrates a ring oscillator design used to create the SE signal. When the oscillate input signal is high, this activates the ring oscillator to then produce a clock signal, with the SE signal being derived from the clock signal via some delay circuitry 870.

As shown in Figure 13C, the duration for which the SE signal goes low, namely duration d1, is determined by the component 870, and the period of time for which the discharge signal goes high, namely duration d2, is determined by the delay circuitry 860 of Figure 13A.

The voltages Vb_SE and Vb_Discharge are used to control the delay periods d1 and d2, and in particular if these voltages are reduced, the delay periods will grow accordingly.

At the positive edge of the clock, data is launched onto the word lines. Since the "Discharge" signal is low at this time, the bit lines are isolated from the word lines. During period "d1", data settles on the word lines and the bit lines get fully pre-charged. "d2" is the period for which the bit lines are conditionally discharged. Simultaneously the SE signal goes high and the sense amplifier starts sampling the bit lines. The bit line is discharged just sufficiently enough so that the sense amplifier can detect it. This is done to save power. Hence, after time "d2", the "Discharge" signal goes low, while the SE signal remains high.

Figure 14 schematically illustrates how the channel_free signal referred to in Figures 10B and 11B is generated in accordance with one embodiment. In particular, as shown, the outputs from the various configuration storage circuits 900, 905, 910 within crossbar cells connected to a particular output channel are logically OR-ed together by the OR gates 902, 907, 912, with the result then being inverted by the inverter 915.

Accordingly, it can be seen that the `channel_free` signal will be at a logic zero value if any of the configuration storage circuits stores a logic one routing value, and only if all of them store a logic zero routing value will the `channel_free` signal be set to one. Hence, if at the start of the arbitration mode of operation, one of the configuration storage circuits is already set to a logic one value, indicating that a source circuit already has ownership of that channel, then none of the asserted transmission requests at that time will be granted, and only once that channel has been released will it be possible for another source to request and be granted that channel.

The priority configuration module shown in Figure 1 may be arranged in a variety of ways. Figure 15A shows one implementation of the priority configuration module 930. In this embodiment, the priority configuration module 930 acts as a finite state machine (FSM) that periodically writes priorities into the crossbar network using the word lines of each data input path 940, 945, 950. In this example, the data input line 940 is coupled to source circuit 0, the data input path 945 is coupled to source circuit 1, and the data input path 950 is coupled to source circuit 15. The priority configuration module 930 maintains a priority vector 935 for each source, and in this example the value 0 implies the highest priority, whilst the value 15 implies the lowest priority. Hence, in accordance with the currently selected priority vector 935, source 0 is the highest priority source, source 1 is the second highest priority source, and source 15 is the lowest priority source. Hence it can be seen that logic one values are programmed via the data input path 940 into the priority storage elements of the crossbar cell associated with source circuit 0, thereby ensuring that whenever source circuit 0 asserts a request, it will discharge all of the other bit lines other than the one being monitored by the configuration storage circuitry coupled to source circuit 0.

Similarly, via the data input bus 945, the priority storage elements connected to all of the bit lines other than the one associated with source circuit 0 are written to a logic one value to cause those bit lines to be conditionally discharged in the presence of an asserted transmission request from source 1. However, even in the event of such an asserted transmission request, the bit line associated with source circuit 0 will not be discharged.

Similarly, considering source circuit 15, logic zero values are output over the corresponding data input bus 950, so that all of the priority storage elements in the

crossbar cells associated with source circuit 15 store logic zero values, thereby causing no bit lines to be discharged in the presence of an asserted transmission request from source circuit 15. In such situations, source circuit 15 will only be granted its asserted transmission request if no other source circuits are asserting a transmission request at the time.

However, to ensure fairness in use of the crossbar, the priority configuration module 930 of this embodiment periodically updates the priorities by switching to one of the other priority vectors illustrated schematically in Figure 15A. In particular, in the embodiment shown, the priority configuration module assigns priorities in a round-robin scheme, so that over time each source circuit has a period of time in which it is the highest priority source.

However, in an alternative embodiment as shown in Figure 15B, the priority configuration module takes the form of an adaptive priority configuration module 960. The adaptive priority configuration module only maintains one priority vector 965, but this priority vector is modified by the priority vector modification circuitry 975 dependent on an activity analysis performed by the activity analysis circuitry 970. In particular, the activity analysis circuitry 970 monitors the asserted transmission requests from each source, and optionally also monitors the grant signals generated in response to those asserted transmission requests, in order to build up an understanding of which sources are being granted access to which channels over time. The adaptive priority configuration module 960 then uses this information to assign priorities more intelligently than merely using the round-robin approach of Figure 15A, with the aim of increasing the overall efficiency of the crossbar network. The efficiency can be defined as the proportion of the operation time that the crossbar network is in the transmission mode of operation (i.e. the duration of time for which the bit lines are used to send data and not to detect and resolve conflicts).

It will be appreciated that there are a number of ways in which it may be decided how to update the priority vectors dependent on the activity information received by the activity analysis circuitry 970. For example, possible schemes to update the priority vectors include:

- 1) a round robin scheme in which priorities are rotated periodically;

- 2) a random assignment in which a random number generator is used to generate random priorities; or
- 3) a LRG (least recently granted) scheme. In this scheme, as soon as a channel succeeds in acquiring a channel, it is assigned the lowest priority for that channel. Simultaneously the highest priority is assigned to the source that has least recently acquired that channel.

Since the output data lines are precharged high in the above described embodiment, a static high input would result in those data lines being discharged every clock cycle. Figures 16A to 16E illustrate an encoding scheme that can be used to mitigate this, and thereby reduce power consumption. In particular, Figure 16A shows encoder circuitry that may be placed between the source and the input data path of the crossbar, and Figure 16B illustrates associated decoder circuitry that may be placed between the output of the sense amplifiers 65 and the destination circuit. Considering first the encoder circuitry of Figure 16A, each item of new data is latched in the flip-flop 600 and the old data is propagated via the NAND gate 605 to the flip-flop 610. The comparator 615 then compares the new data with the old data and whenever there is a difference this causes a logic zero value to be output to the inverter 620 resulting in a logic one value being output by the encoder circuitry. Hence, from the provided input data, the encoder produces logic zero values separated by logic one values every time the input data changes from a logic zero to a logic one value or from a logic one value to a logic zero value. From the earlier discussion of the crossbar cells, it will be appreciated that the data output lines will only be discharged when such logic one values appear, thereby significantly reducing the power consumption associated with the precharge operation.

When the first data item is input, there will be no previous data to compare against, and accordingly the sync signal is used to set an initial state in the flip-flop 610. Further, the synchronisation pulse can be used to reset the encoder hardware (without the expense of a clock cycle) every time the crossbar switches to a new configuration.

The corresponding decoder circuitry is illustrated in Figure 16B. The structure of the components 630 and 635 is illustrated schematically in Figure 16C by the component 660. As shown, such a component actually consists of a series of transistors 665, 670, 675, 680 connected in series.

The sync_d signal is the same as the sync signal, but delayed by one clock cycle. The sync_d_b signal is the inverted sync_d signal. The relationship between these three signals is shown in Figure 16E. It should be noted that the sync signal is an active low signal and hence under normal operation (when the circuitry is not switching to a new configuration), sync and sync_d will be at a high, logic one, level, whilst the sync_d_b signal will be a low, logic zero, level.

As can be seen from Figure 16B, the decoder receives the output from the sense amplifier, the clock signal and the sync_d_b signal and, via a series of NAND gates 645, 650, 655, provides an internal clock signal to the flip-flop 640. The output from the flip-flop 640 drives the output to the destination circuit, and recreates the original input data from the encoded output received via the sense amplifier.

In normal operation, since the sync signal will be at a logic 1 level and the sync_d_b signal will be a logic 0 level, component 635 will be activated while component 630 is deactivated. The three NAND gates 645, 650, 655 provide a pulse to Flip-Flop 640 when the sense amplifier detects a transition and sends a high signal. When a switch to a new configuration takes place, the sync and sync_d signals go low while the sync_d_b signal goes high. Hence, at that time, component 630 is activated while component 635 is deactivated. As a result, instead of toggling the previous data via the inverter 635, the flip-flop 640 takes the data (after component 630 inverts it) from the sense-amplifier.

Figure 16D illustrates the input data to the encoder circuitry, the resultant encoded data output from the encoder circuitry, an indication of the corresponding bit line being discharged each time the encoded data goes high, and shows the output data produced by the decoder circuitry based on the information received from the sense amp. It can be seen that the output data faithfully reproduces the input data that was passed to the encoder.

In the embodiments discussed above, the size of the crossbar circuitry corresponds with the size of the input buses and output buses. Hence, considering input buses providing 16 word lines and output buses providing 16 bit lines, a 16 x 16 crossbar circuit has been discussed. However, the technique can also be adapted to operate with crossbar circuits having arrays of crossbar cells larger than the width of the input buses and output buses (what will be referred to herein as a narrow channel

design), or indeed can be adapted for situations where the array of crossbar cells is smaller than the bus widths (what will be referred to herein as a wide channel design).

Considering first a narrow channel design, Figure 17 illustrates one embodiment for implementing such a narrow channel design. In this particular example, it is assumed that a 64 x 64 array of crossbar cells 20 is provided within the crossbar, with the precharge module 55 again being used to precharge the bit lines running through the crossbar. However, the data input paths and data output paths are only 16 bits wide, and accordingly it is not possible to perform the earlier described arbitration process in a single cycle. Instead, in such embodiments, the crossbar can be divided into sections, in this example the crossbar being divided into four sections 1010, 1020, 1030, with a counter 1040 being used to identify which section is currently active. The counter can hence be incremented every arbitration cycle to select which section to serve during arbitration in that arbitration cycle. Since each section still has 64 crossbar cells in each row, it is clear that the data input paths 1000 do not by themselves enable individual data output paths to be identified by asserted transmission requests. Accordingly, in one embodiment a two bit additional signal 1002 is provided when asserting transmission requests to identify which data output path is being requested by the associated source circuit.

Accordingly, when the counter has a value of 00, only asserted transmission requests from sources 0 to 15 are considered, and for each data output path the highest priority source amongst sources 0 to 15 requesting access to that data output path is granted access for transferring data in the subsequent transmission cycle. Following such an arbitration and transmission sequence, the counter is then incremented to 01, so that in the next arbitration cycle only asserted transmission requests from sources 16 to 31 are considered. This process is repeated for each section in turn. As a result, it will be appreciated that the arbitration latency for any given asserted transmission request will vary from one to four cycles for the dimensions of the example in Figure 17. For example, if source 0 asserts a transmission request in an arbitration cycle for which the count is 00, then its request can be arbitrated straightaway in that cycle, and if it is the highest priority request in section 1, it will be granted. However, if source 0 asserts a transmission request when the count is 01, then its request will not even be a

candidate for arbitration until the fourth following arbitration cycle when the counter has returned to 00 (and accordingly section 1 1010 is being arbitrated).

A benefit of such an approach is that it has a minimal wiring and logic overhead, and hence the solution will provide a good solution in situations where the variation in arbitration latency can be tolerated. However, it is worth noting that as the number of sections increases, the worst case arbitration latency also increases.

Figures 18A and 18B illustrate an alternative embodiment where the crossbar is again divided into sections, but arbitration is then done hierarchically in two steps. In a first step, arbitration is performed amongst the sections, and then in the second step arbitration is performed amongst all the asserted transmission requests in a given section. The same group of bit lines is used for both arbitration steps. In this example, the crossbar cells are denoted by the numeral 20', since each of the crossbar cells is constructed basically as discussed earlier, but now each crossbar cell includes two sense amp enabled latches (SAELs), with the first SAEL being used in the first arbitration cycle and the second SAEL being used in the second arbitration cycle.

Hence, in arbitration cycle 1, each of the sections 1050, 1060, 1070, 1080 are considered but the individual crossbar cells 20' for a particular data output path within the given section are in effect considered in combination, as indicated schematically by the boxes 1052, 1062, 1072, 1082. In particular, considering the 64 x 64 crossbar example of Figure 18A with 16-bit channels, then each of the four separate sections 1050, 1060, 1070, 1080 will have relative priorities. If section 1 has the highest priority, then any asserted transmission request from sources 0 to 15 will cause the bit lines for sections 2, 3 and 4 (i.e. sections 1060, 1070, 1080) to be discharged. As a result, the first set of SAELs in section 1 will latch a logic one value, whilst the first set of SAELs in the other sections will latch a logic zero value.

In the next arbitration cycle, as illustrated by Figure 18B, only requests in section 0 (assuming the sequence of events discussed above) use the bit lines for arbitration. Now the process performed within this section 1050 is as discussed earlier, i.e. the bit lines are precharged, and then selectively discharged in accordance with the priority information stored within each crossbar cell. As was discussed earlier in Figure 17, an additional 2-bit signal 1002 is used to qualify which output path is being requested by the 16-bit value input over the word lines 1000. For multiple requests for

any particular data output line, the highest priority request from amongst the sources 0 to 15 will then be granted access, with the second set of SAELs sensing the bit lines in the standard manner. To determine whether a grant has been given to a particular source, then both of the SAELs of a crossbar cell 20' need to be considered, and in particular if both of them store a logic one value, then a grant signal is generated back to the source.

Such an approach has a fixed two cycle arbitration latency at the expense of some additional logic, but no additional interconnect structure. With 16 bit lines in a channel it is possible to potentially arbitrate between 16 sections, with each section containing 16 requests. Hence the arbitration latency can be retained at two cycles for a crossbar up to a 256 x 256 crossbar.

In one embodiment of the Figures 18A/18B approach, two sets of priority storage elements can be provided in each crossbar cell, one set used in association with the first arbitration cycle (where the first set of SAELs are used), and one set used in association with the second arbitration cycle (where the second set of SAELs are used).

Figure 19 illustrates a wide channel implementation, where the width of the input and output channels exceeds the size of the crossbar array. In this particular example, it is assumed that the input and output paths are 32-bits wide, and that the crossbar is a 16 x 16 crossbar, thereby forming a 16 x 16 array of crossbar cells 20. In such an arrangement, only half of the bit lines need to be used during arbitration. Under such conditions, it is possible to take advantage of this fact by allowing multiple priorities to be stored at each crossbar cell through the provision of multiple sets of priority storage circuitry, subsequently allowing the priority configuration to be switched without needing to reload any priority data from the priority configuration module 60. In particular, it will be appreciated that the priority configuration module can program two sets of priority bits in each crossbar cell, whereafter it is possible to selectively switch between these two sets of priority bits without any timing penalty. In particular, in the embodiment illustrated in Figure 19, a priority counter 1130 can be retained in association with the precharge module 55, with the value of that priority counter then being output over the lines 1120 to identify which set of priority bits should be employed within each crossbar cell 20. Dependent on which set of priority bits are chosen, this will alter how the various bit lines of the data output paths 1110

are discharged in response to asserted transmission requests from any of the sources 0 to 15. This provides significant performance improvement in such situations, since the priority assignment mode of operation needs to be used less frequently, given the ability to load both sets of priorities for a channel in a single cycle during the priority assignment mode of operation. Further, subsequent switching between those two sets of priority bits can occur without any timing penalty.

It will be appreciated that in the embodiment illustrated in Figure 19, the priority counter 1130 merely needs to be a single bit counter. However, in other configurations a multi-bit counter can be used. For example, in a 16 x 16 crossbar circuit with 64-bit channels, it is possible to store four sets of priorities within each crossbar cell, and accordingly a 2-bit priority counter 1130 could be used in such embodiments.

Figure 20 illustrates a partially activated network embodiment where a sequence of word line drivers 1200, 1205, 1210 are used to propagate data input values along the data input paths to the various crossbar cells 20 connected to those data input paths. Without such a partially activated network, significant power may be consumed in propagating the data along the entirety of the data input path in situations where it is not needed.

However, power savings can be achieved by using the partially activated network approach set out in Figure 20, where each of the word line drivers is conditionally activated based on a signal derived from the routing values stored in each of the remaining crossbar cells along the data input path. In particular, as shown, a routing value stored in a particular crossbar cell is logically OR-ed with the routing value held in all of the other crossbar cells to its right in a particular row, with each word line driver 1200, 1205, 1210 being activated in dependence on the value of that signal at the point that word line driver is reached in the chain.

Hence, in the example given above, where at least one of the first fifteen crossbar cells in row 0 has its routing value set to one, indicating that the corresponding source has been granted access to one of the output paths 0 to 15, but none of the other crossbar cells further to the right in that row have their routing values set to one, then both the word line drivers 1205 and 1210 will receive a logic zero activate signal, whilst the word line driver 1200 will receive a logic one activate signal.

As shown by the circuitry in the lower half of Figure 20, for each of the word line drivers (for the purposes of illustration word line driver 1200 is considered but all of the word line drivers are constructed similarly), the activate signal is used to selectively activate the word line driver. In particular, the main inverter 1260 within the word line driver is only activated if the activate signal is high, this causing both the transistors 1250 and 1255 to turn on. When the activate signal is low, these transistors do not turn on and the inverter 1260 is not activated. When activated, the inverters 1260 and 1265 act in combination to amplify the input data for propagation further along the data input path.

Accordingly, in the embodiment discussed with reference to Figure 20, data is not driven on the word lines beyond the channel after which the remaining crossbar cell switches are turned off. This can provide a useful power saving scheme for large crossbars, and can yield particularly good power savings in implementations which rarely do multi-casting.

From the above description of embodiments, it will be seen that such embodiments provide a novel crossbar implementation scheme that can be used to design very large crossbars, such as may be required for multi-core applications. The crossbar circuitry of embodiments of the present invention re-uses the data output lines during the arbitration mode of operation to detect and resolve conflicts, and to program the configuration storage elements within each crossbar cell accordingly. The proposed implementation of embodiments of the present invention uses the minimum possible area to design the crossbar. As opposed to typical prior art implementations, where both the number of cross-points as well as the logic embedded at each cross-point grows, in accordance with embodiments of the present invention only the number of cross-points grows.

For embodiments where the size of the crossbar cell array does not exceed the width of the input and output channels, arbitration can be performed in a single cycle, thereby minimising delay overhead in performing arbitration. Further, embodiments provide programmable priority allowing more scope to optimise applications during run time. For example, during multi-media processing, source circuits doing more critical jobs can be given higher priorities to access memory devices such as a higher level cache.

Further, by integrating the arbiter with the crossbar, communication overhead between the two has been eliminated.

In addition, when employing the crossbar cell structure of the above described embodiments, neither the size of the transmission circuitry, nor the driver circuitry used to provide the input data on each input data path, need to increase in size as the crossbar circuitry increases in size. Further, no additional hardware or sizing is required to support multi-casting, where a single input data value is broadcast to multiple output paths.

Further, due to the regular structure employed in the above described embodiments, routing becomes significantly more straightforward than in typical prior art approaches.

In embodiments of the present invention, inputs are fed into the crossbar as buses and not as packets comprising the same bits from different buses, as for example would be required in a typical prior art crossbar employing multiplexing techniques. This makes the crossbar circuitry of embodiments of the present invention easy to utilise as a wide bus interconnect.

Due to the design of the crossbar circuitry of embodiments of the present invention, the delay through the crossbar grows linearly with size. This makes the crossbar circuitry of such embodiments readily usable for very large crossbars, such as 128 x 128 or 256 x 256 input/output crossbars.

As mentioned earlier, the design is very regular and the delay across all channels is uniform. Further, inputs can be fed from either horizontal end and outputs are available at either vertical end. This makes design floor planning easy by reducing routing congestion.

Due to the regular bit-cell like architecture of the crossbar circuitry of embodiments of the present invention, the design and layout of the crossbar circuitry can readily be integrated into existing CAD (computer aided design) flow processes.

When compared with prior art techniques, the crossbar circuitry of embodiments of the present invention runs faster, yet consumes less power, due to the smaller size of the interconnect and the fast sensing techniques employed.

Although a particular embodiment of the invention has been described herein, it will be apparent that the invention is not limited thereto, and that many modifications and additions may be made within the scope of the invention. For

example, various combinations of the features of the following dependent claims could be made with the features of the independent claims without departing from the scope of the present invention.

1. Crossbar circuitry for interconnecting a plurality of source circuits and a plurality of destination circuits such that data input to the crossbar circuitry from any of said plurality of source circuits can be output to any of said plurality of destination circuits, the crossbar circuitry comprising:

a plurality of data input paths passing through said crossbar circuitry, each data input path being connectable to one of said plurality of source circuits and providing a plurality of word lines;

a plurality of data output paths passing through said crossbar circuitry transverse to the plurality of data input paths, each data output path being connectable to one of said plurality of destination circuits and providing a plurality of bit lines;

a crossbar cell associated with each intersection between one of said data input paths and one of said data output paths, each crossbar cell comprising:

configuration storage circuitry programmable to store a routing value, the routing value being programmed to a first value to indicate that data input along the word lines of the data input path to the associated intersection is to be output on the bit lines of the data output path at the associated intersection, and the routing value being programmed to a second value to indicate that data input along the word lines of the data input path to the associated intersection is not to be output on the bit lines of the data output path at the associated intersection;

transmission circuitry which in a transmission mode of operation is responsive to the routing value having said first value to detect the data input along the word lines of the data input path and to output an indication of that data on the bit lines of the data output path at the associated intersection; and

arbitration circuitry that operates in an arbitration mode of operation in dependence on a transmission request received by the crossbar cell from the source circuit connected to the data input path of the associated intersection, if the transmission request is asserted to indicate that said source circuit wishes to route data from the data input path to the data output path at the associated intersection, the arbitration circuitry being arranged to operate in combination with the arbitration circuitry of other crossbar cells associated with the same data output path to re-use the

bit lines of the data output path to detect the presence of multiple asserted transmission requests for said same data output path, and in the event of such multiple asserted transmission requests to implement a predetermined priority scheme to cause the configuration storage circuitry of only one crossbar cell associated with said same data output path to have its routing value programmed to said first value, thereby resolving conflict between said multiple asserted transmission requests according to said predetermined priority scheme.

2. Crossbar circuitry as claimed in Claim 1, wherein:

each data output path provides n bit lines;

each configuration storage circuitry is associated with one of said n bit lines via which the routing value is programmed into the configuration storage circuitry, such that for up to n crossbar cells associated with the same data output path, the configuration storage circuitry of those crossbar cells have a different associated bit line of said n bit lines;

in said arbitration mode of operation, the arbitration circuitry of any crossbar cells associated with the same data output path that receive an asserted transmission request being arranged to selectively modify the voltage on the n bit lines dependent on said predetermined priority scheme;

whereafter each configuration storage circuitry of a crossbar cell receiving an asserted transmission request samples the voltage on its associated bit line in order to program said routing value.

3. Crossbar circuitry as claimed in Claim 2, wherein:

each of the bit lines are precharged to a first voltage level and then during the arbitration mode of operation the arbitration circuitry of any crossbar cells associated with the same data output path that receive an asserted transmission request selectively discharge the voltage on the n bit lines dependent on said predetermined priority scheme.

4. Crossbar circuitry as claimed in Claim 3, wherein each configuration storage circuitry comprises a sense-amp enabled latch which, in the presence of an asserted

transmission request, senses the voltage on its associated bit line following the selective discharge operation performed by the arbitration circuitry.

5. Crossbar circuitry as claimed in Claim 1, wherein:

each data input path provides n word lines; and

in the arbitration mode of operation the transmission requests for up to n crossbar cells associated with the same data input path are input to the crossbar circuitry via the n word lines of that same data input path.

6. Crossbar circuitry as claimed in Claim 5, wherein each configuration storage circuitry is associated with one of said n word lines, such that for up to n crossbar cells associated with the same data input path, the configuration storage circuitry of those crossbar cells have a different associated word line of said n word lines via which the transmission request is received.

7. Crossbar circuitry as claimed in Claim 6, wherein:

a matrix of $m \times n$ crossbar cells are provided, wherein m is an integer of 2 or more;

the matrix is divided into a plurality of sections;

a series of arbitration operations are used to implement the predetermined priority scheme to cause the configuration storage circuitry of only one crossbar cell associated with said same data output path to have its routing value programmed to said first value, thereby resolving conflict between multiple asserted transmission requests according to said predetermined priority scheme;

for each arbitration operation in the series, one or more of the sections in said plurality of sections being subjected to that arbitration operation.

8. Crossbar circuitry as claimed in Claim 7, wherein each arbitration operation operates on a single section, such that after said series of arbitration operations have been performed all of said plurality of sections have been subjected to said arbitration operation.

9. Crossbar circuitry as claimed in Claim 7, wherein:

in a first arbitration operation of the series, all of the plurality of sections are subjected to the first arbitration operation in order to identify one of the sections in said plurality that contains the configuration storage circuitry whose routing value is to be programmed to said first value; and

in a second arbitration operation of the series, said one of the sections identified by the first arbitration operation is subjected to the second arbitration operation to identify the configuration storage circuitry within that section whose routing value is to be programmed to said first value.

10. Crossbar circuitry as claimed in Claim 1, wherein said predetermined priority scheme is fixed.

11. Crossbar circuitry as claimed in Claim 1, wherein each crossbar cell further comprises:

priority storage circuitry programmable to store priority data identifying the actions to be performed by the associated arbitration circuitry in respect of one or more of the n bit lines in order to implement the predetermined priority scheme.

12. Crossbar circuitry as claimed in Claim 11, further comprising:

a priority configuration module, which in a priority assignment mode of operation, inputs priority data over the plurality of word lines of one or more of the data input paths, and asserts a priority load signal to the priority storage circuitry of one or more of the crossbar cells, such that any priority storage circuitry receiving an asserted load priority signal samples the data on at least one word line in order to update the priority data stored therein.

13. Crossbar circuitry as claimed in Claim 11, wherein:

each data output path provides n bit lines;

each configuration storage circuitry is associated with one of said n bit lines via which the routing value is programmed into the configuration storage circuitry, such that for up to n crossbar cells associated with the same data output path, the

configuration storage circuitry of those crossbar cells have a different associated bit line of said n bit lines;

within each crossbar cell the arbitration circuitry and the priority storage circuitry is associated with the $n-1$ bit lines other than the one bit line associated with the configuration storage circuitry of that crossbar cell, with the priority storage circuitry identifying which of those $n-1$ bit lines should have their voltages modified by the arbitration circuitry in order to implement said predetermined priority scheme.

14. Crossbar circuitry as claimed in Claim 12, wherein:

said plurality of word lines comprise n word lines;

a matrix of $n/m \times n/m$ crossbar cells are provided, wherein m is an integer of 2 or more;

each crossbar cell comprises a plurality of priority storage circuitry wherein said plurality is less than or equal to m ;

whereby the priority configuration module is arranged to program multiple priority storage circuitry in each crossbar cell using the n word lines of the data input path.

15. Crossbar circuitry as claimed in Claim 12, wherein:

the plurality of source circuits are connectable to a first end of said plurality of data input paths;

the priority configuration module is connectable to a second end of said plurality of data input paths;

the source circuits being arranged to issue asserted transmission requests via the plurality of word lines of each data input path; and

the priority configuration module being arranged to monitor the asserted transmission requests issued by each source circuit, and to modify the priority data to be output by the priority configuration module during the priority assignment mode of operation dependent on the number of asserted transmission requests from each source circuit.

16. Crossbar circuitry as claimed in Claim 2, wherein each crossbar cell further comprises:

release circuitry coupled to the one bit line associated with the configuration storage circuitry of that crossbar cell, on receipt of an asserted release request the channel release circuitry causing the voltage on that one bit line to be modified to a level that causes said second value to be stored as said routing value in the configuration storage circuitry of that crossbar cell.

17. Crossbar circuitry as claimed in Claim 16, wherein:

each of the bit lines are precharged to a first voltage level;

on receipt of said asserted release request, the release circuitry discharges the voltage on the one bit line associated with the configuration storage circuitry of that crossbar cell; and

each configuration storage circuitry comprises a sense-amp enabled latch which, in the presence of an asserted release request to the associated release circuitry, senses the voltage on its associated bit line following the discharge operation performed by the release circuitry in order to update the routing value to said second value, thereby releasing the associated data output path.

18. Crossbar circuitry as claimed in Claim 16, wherein:

each data input path provides n word lines; and

the release requests for up to n crossbar cells associated with the same data input path are input to the crossbar circuitry via the n word lines of that same data input path;

each release circuitry is associated with one of said n word lines, such that for up to n crossbar cells associated with the same data input path, the release circuitry of those crossbar cells have a different associated word line of said n word lines via which the release request is received.

19. Crossbar circuitry as claimed in Claim 18, wherein:

the release requests are issued in said arbitration mode of operation, and a control signal is issued by the plurality of source circuits in the arbitration mode of

operation to identify if the n word lines of the associated data input path are carrying release requests or transmission requests.

20. Crossbar circuitry as claimed in Claim 1, wherein for the one crossbar cell associated with the same data output path whose configuration storage circuitry has its routing value programmed to said first value during the arbitration mode of operation, a grant signal is asserted to the associated source circuit to confirm that the asserted transmission request has been granted.

21. Crossbar circuitry as claimed in Claim 1, wherein each data output path associated with a crossbar cell in the transmission mode of operation is precharged to a first logic level prior to data transfer, the transmission circuitry of each crossbar cell comprising:

first and second switches connected in series between the data output path and a second logic level;

in the transmission mode of operation the first switch being open or closed dependent on the routing value stored in the associated configuration storage circuitry, and the second switch being open or closed dependent on the data input over the data input path.

22. Crossbar circuitry as claimed in Claim 21, wherein each said data input path comprises n word lines for carrying an n -bit input data value during the transmission mode of operation, and each said data output path comprises n bit lines for carrying an n -bit data value during said transmission mode of operation, at least the second switch being replicated for each bit line.

23. Crossbar circuitry as claimed in Claim 21, wherein in the transmission mode of operation a bit line of the data output path is pulled to the second logic level if the routing value is said first value and the input data bit on the corresponding word line is at said first logic level.

24. Crossbar circuitry as claimed in Claim 23, wherein each crossbar cell further comprises conditional discharge circuitry for turning off the first switch irrespective of the routing value whilst the associated data output path is being precharged to the first logic level, and for allowing the first switch to be controlled by the routing value following the precharging of the associated data output path to the first logic level.

25. Crossbar circuitry as claimed in Claim 23, further comprising:

sense amplifier circuitry to detect, during the transmission mode of operation, the data output on the bit lines of the data output path, thereby allowing detection of a transition towards the second logic level before the voltage on a bit line of the data output path has reached the second logic level.

26. Crossbar circuitry as claimed in Claim 23, further comprising:

encode circuitry between each of said plurality of source circuits and said plurality of data input paths;

decode circuitry between each of said plurality of data output paths and said plurality of destination circuits;

the encode circuitry applying an encoding operation to encode input data provided by each source circuit into an encoded format that reduces the number of times the data output paths need to be pulled to the second logic level following their precharge to the first logic level, when compared to the number of times the data output paths would need to be pulled to the second logic level if the original input data were passed through the crossbar circuitry;

the decode circuitry applying a corresponding decoding operation to identify the original input data provided by the source circuit from the encoded data output over the data output path.

27. Crossbar circuitry as claimed in Claim 1, wherein said plurality of source circuits are connectable to either end of said plurality of data input paths.

28. Crossbar circuitry as claimed in Claim 1, wherein said plurality of destination circuits are connectable to either end of said plurality of data output paths.

29. Crossbar circuitry as claimed in Claim 1, wherein:

each data input path consists of multiple input path portions separated by word line drivers;

each word line driver is activated in dependence on the routing values stored in the configuration storage circuitry of crossbar cells provided along the associated data input path but further from the source circuit than that word line driver;

whereby each word line driver is only activated if the configuration storage circuitry of at least one crossbar cell provided along the associated data input path but further from the source circuit than that word line driver has a routing value set to said first value.

30. A data processing apparatus comprising:

a plurality of memory devices for storing data values;

a plurality of processors for performing data processing operations in parallel on multiple of the data values stored in said plurality of memory devices; and

crossbar circuitry as claimed in Claim 1 for routing the data value from any of said plurality of memory devices to any of said plurality of processors.

31. Crossbar circuitry for interconnecting a plurality of source means and a plurality of destination means such that data input to the crossbar circuitry from any of said plurality of source means can be output to any of said plurality of destination means, the crossbar circuitry comprising:

a plurality of data input path means passing through said crossbar circuitry, each data input path means for connecting to one of said plurality of source means and providing a plurality of word line means;

a plurality of data output path means passing through said crossbar circuitry transverse to the plurality of data input path means, each data output path means for connecting to one of said plurality of destination means and providing a plurality of bit line means;

a crossbar cell means associated with each intersection between one of said data input path means and one of said data output path means, each crossbar cell means comprising:

configuration storage means programmable for storing a routing value, the routing value being programmed to a first value to indicate that data input along the word line means of the data input path means to the associated intersection is to be output on the bit line means of the data output path means at the associated intersection, and the routing value being programmed to a second value to indicate that data input along the word line means of the data input path means to the associated intersection is not to be output on the bit line means of the data output path means at the associated intersection;

transmission means, responsive to the routing value having said first value in a transmission mode of operation, for detecting the data input along the word line means of the data input path means and to output an indication of that data on the bit line means of the data output path means at the associated intersection; and

arbitration means for operating in an arbitration mode of operation in dependence on a transmission request received by the crossbar cell means from the source means connected to the data input path means of the associated intersection, if the transmission request is asserted to indicate that said source means wishes to route data from the data input path means to the data output path means at the associated intersection, the arbitration means for operating in combination with the arbitration means of other crossbar cell means associated with the same data output path means to re-use the bit line means of the data output path means to detect the presence of multiple asserted transmission requests for said same data output path means, and in the event of such multiple asserted transmission requests, for implementing a predetermined priority scheme to cause the configuration storage means of only one crossbar cell means associated with said same data output path means to have its routing value programmed to said first value, thereby resolving conflict between said multiple asserted transmission requests according to said predetermined priority scheme.

32. A method of operating crossbar circuitry to interconnect a plurality of source circuits and a plurality of destination circuits such that data input to the crossbar circuitry from any of said plurality of source circuits can be output to any of said plurality of destination circuits, the crossbar circuitry having a plurality of data input paths passing through said crossbar circuitry, each data input path being connectable to one of said plurality of source circuits and providing a plurality of word lines, and a plurality of data output paths passing through said crossbar circuitry transverse to the plurality of data input paths, each data output path being connectable to one of said plurality of destination circuits and providing a plurality of bit lines, the method comprising the steps of:

employing a crossbar cell in association with each intersection between one of said data input paths and one of said data output paths;

programming a routing value in each crossbar cell, the routing value being programmed to a first value to indicate that data input along the word lines of the data input path to the associated intersection is to be output on the bit lines of the data output path at the associated intersection, and the routing value being programmed to a second value to indicate that data input along the word lines of the data input path to the associated intersection is not to be output on the bit lines of the data output path at the associated intersection;

in a transmission mode of operation, causing the crossbar cell to be responsive to the routing value having said first value to detect the data input along the word lines of the data input path and to output an indication of that data on the bit lines of the data output path at the associated intersection; and

in an arbitration mode of operation, causing arbitration circuitry in the crossbar cell to operate in dependence on a transmission request received by the crossbar cell from the source circuit connected to the data input path of the associated intersection, if the transmission request is asserted to indicate that said source circuit wishes to route data from the data input path to the data output path at the associated intersection, the arbitration circuitry operating in combination with the arbitration circuitry of other crossbar cells associated with the same data output path to re-use the bit lines of the data output path to detect the presence of multiple asserted transmission requests for said same data output path, and in the event of such multiple asserted transmission

requests to implement a predetermined priority scheme to cause the configuration storage circuitry of only one crossbar cell associated with said same data output path to have its routing value programmed to said first value, thereby resolving conflict between said multiple asserted transmission requests according to said predetermined priority scheme.

1 . Abstract

Crossbar circuitry, and a method of operation of such crossbar circuitry, are provided. The crossbar circuitry has an array of data input paths and data output paths where the data output paths are transverse to the data input paths. At each intersection between a data input path and a data output path, a crossbar cell is provided that comprises a configuration storage circuit programmable to store a routing value, a transmission circuit, and an arbitration circuit. In a transmission mode of operation, the transmission circuit is responsive to the routing value being a first value, indicating that the data input path should be coupled to the data output path, to detect the data input along the data input path, and to output an indication of that data on the data output path at the associated intersection. In an arbitration mode of operation, the arbitration circuitry is operable in the presence of an asserted transmission request from the associated source circuit to operate in combination with the arbitration circuits of other crossbar cells associated with the same data output path to re-use the bit lines of the data output path to detect the presence of multiple asserted transmission requests for the same data output path. In the event of such multiple asserted transmission requests, the arbitration circuitry operates in combination with the other arbitration circuits to implement a predetermined priority scheme to cause the configuration storage circuit of only one crossbar cell associated with the same data output path to have its routing value programmed to the first value, thereby resolving conflict between the multiple asserted transmission requests according to the predetermined priority scheme. Such a construction of crossbar circuitry enables a very efficient resolution of conflicts to be performed, whilst providing a very regular design, with uniform delay across all paths, and which requires significantly less control lines than typical prior art crossbar designs. Such crossbar circuitry is readily scalable to form large crossbars.

2 . Representative Drawing

Figure 1

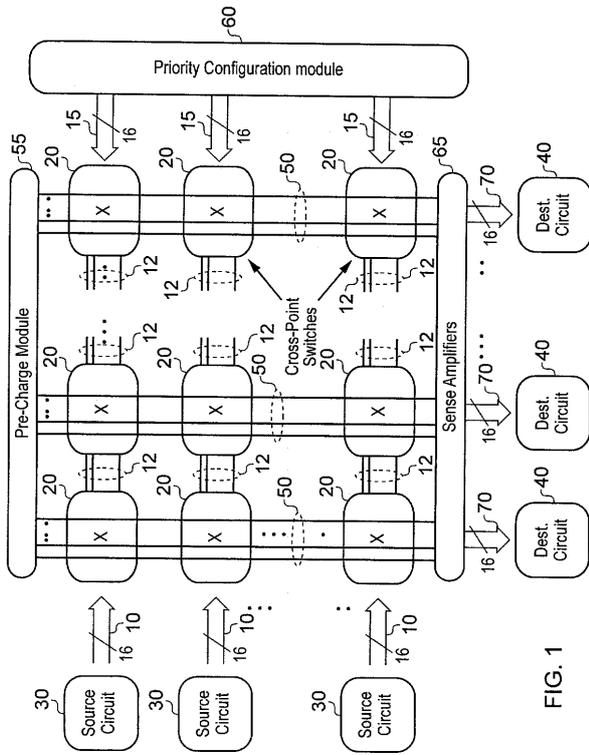


FIG. 1

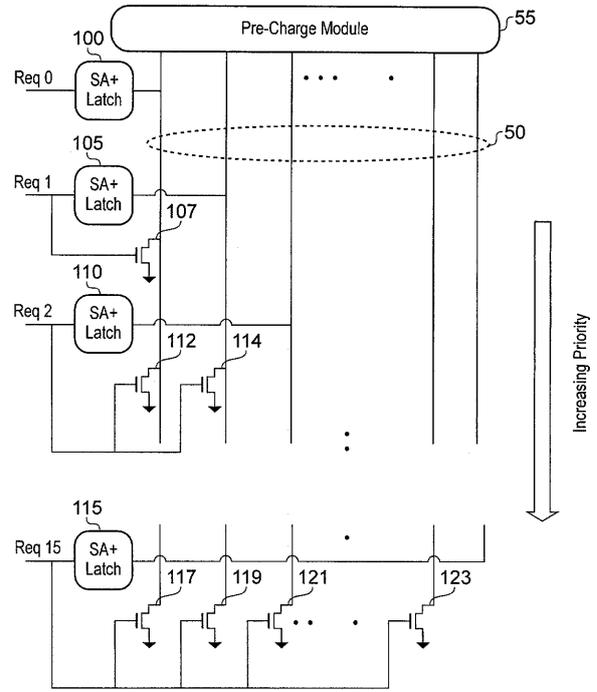


FIG. 2

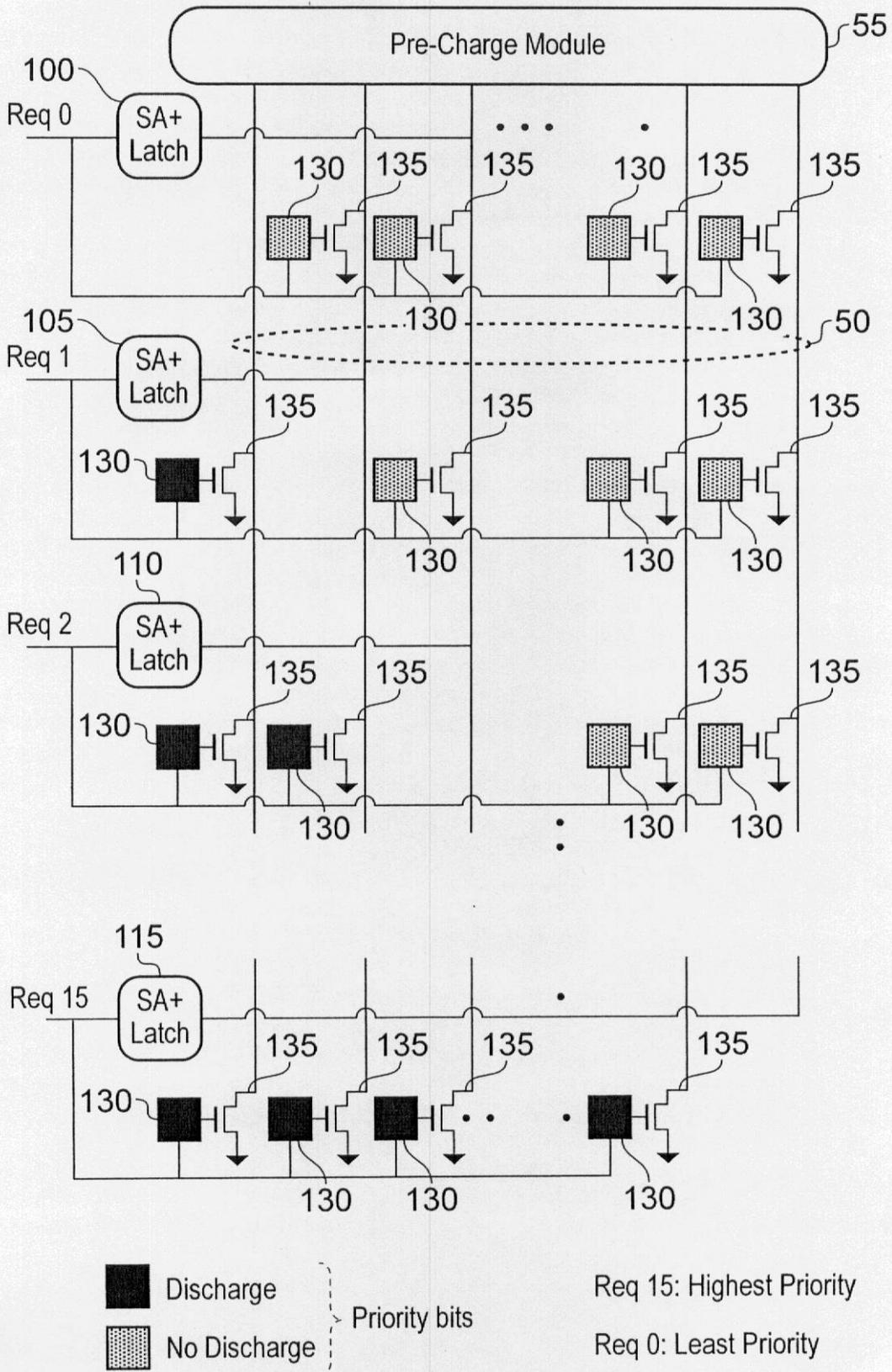


FIG. 3

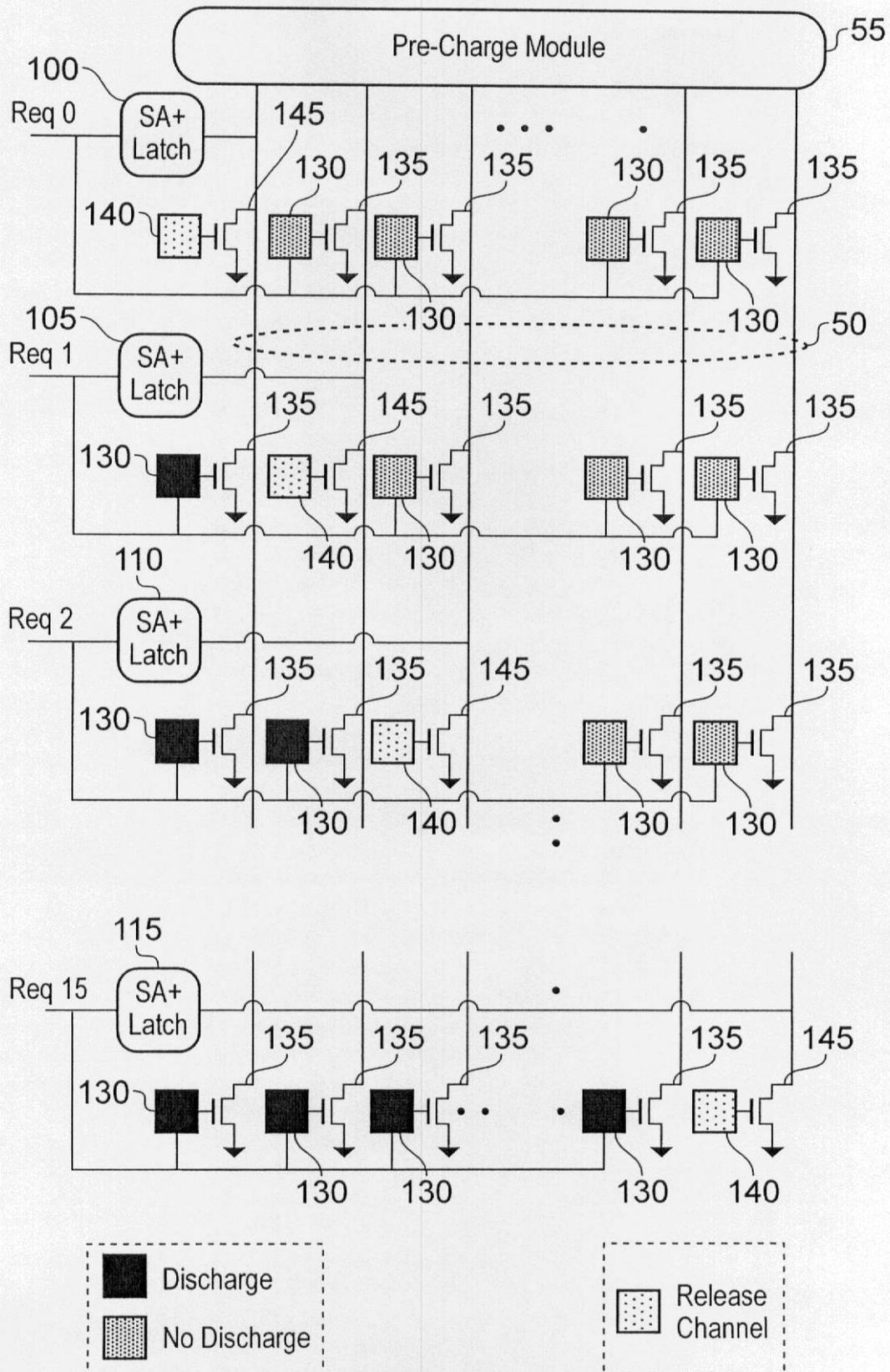
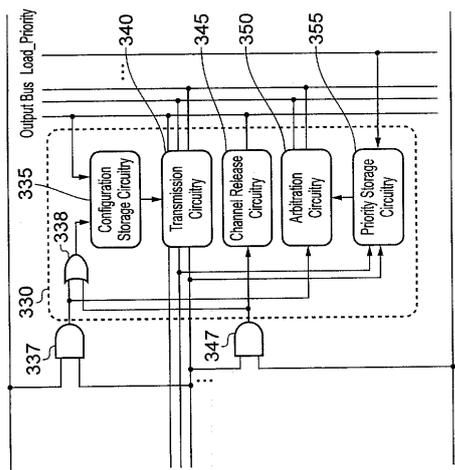
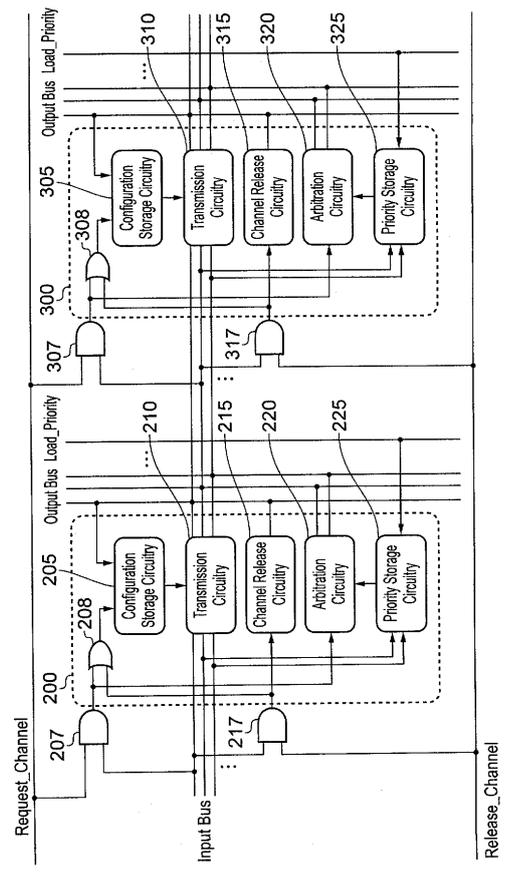
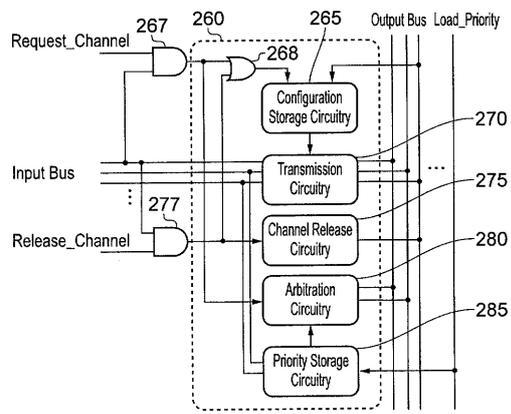
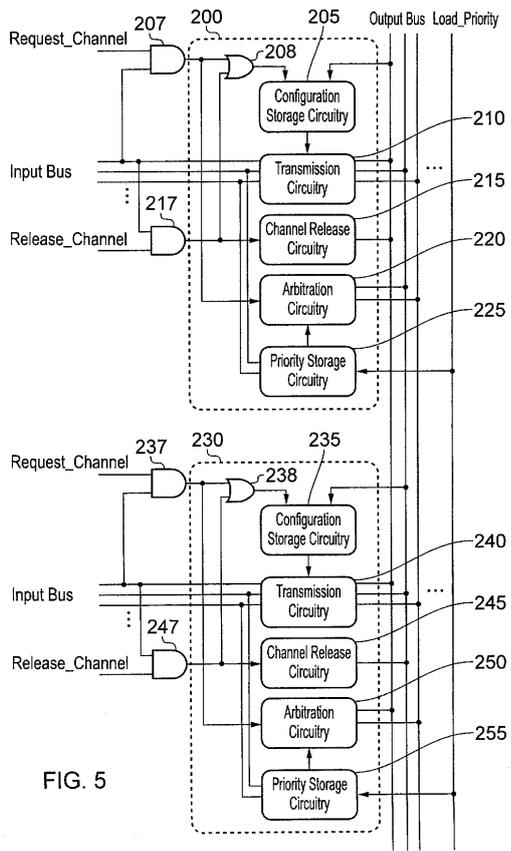


FIG. 4



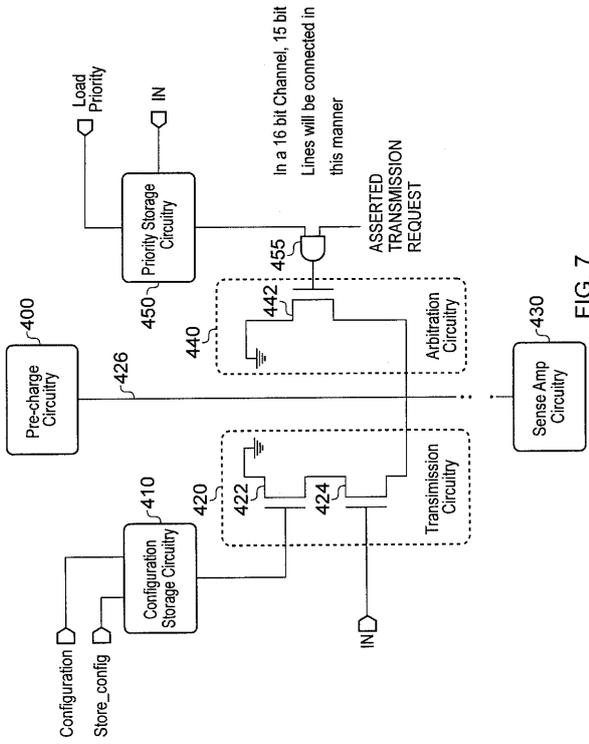


FIG. 7

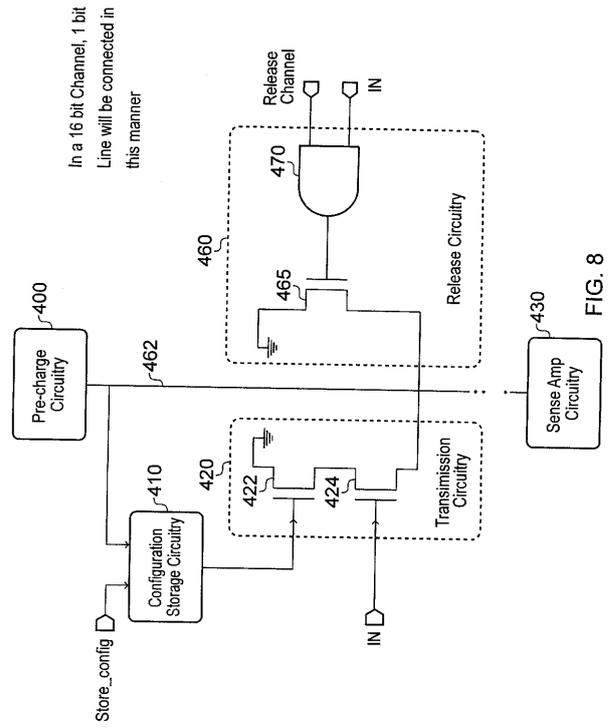


FIG. 8

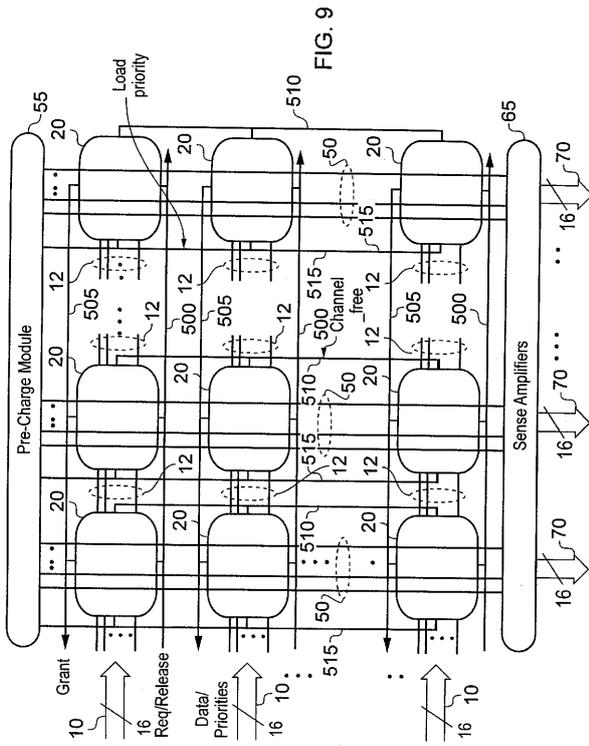


FIG. 9

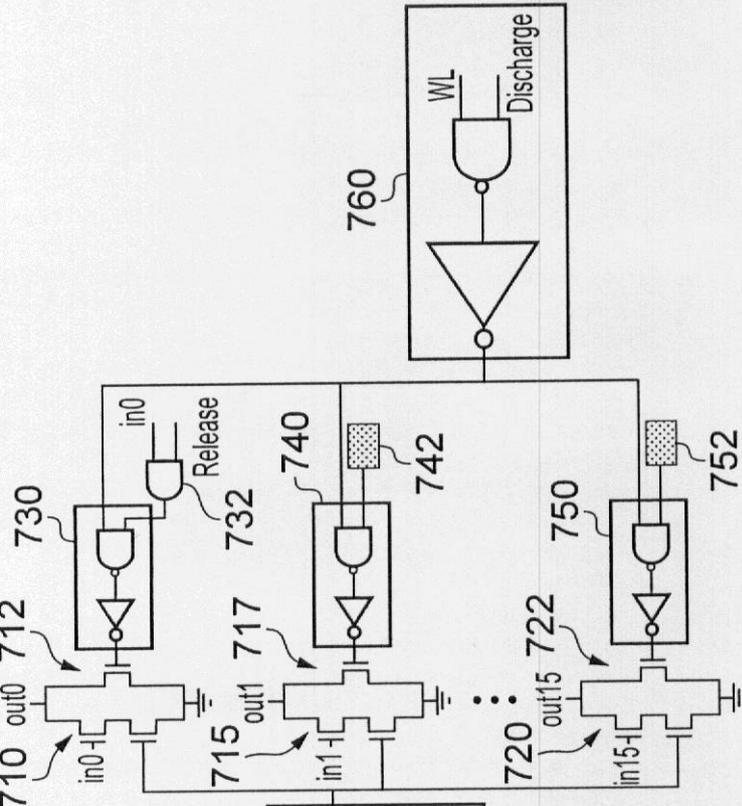


FIG. 10A

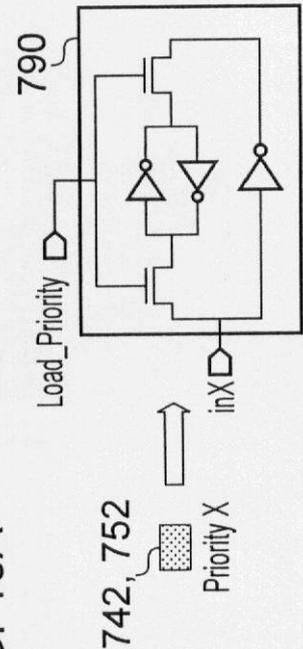


FIG. 10B

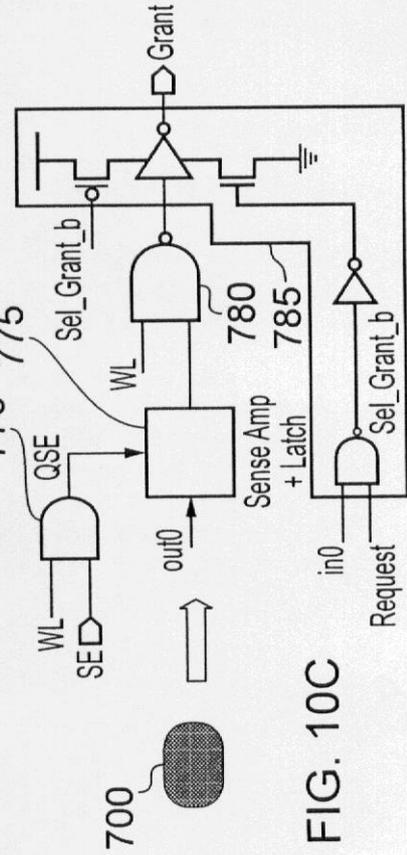


FIG. 10C

-  Storage element to hold Configuration
-  Storage element to hold Priority

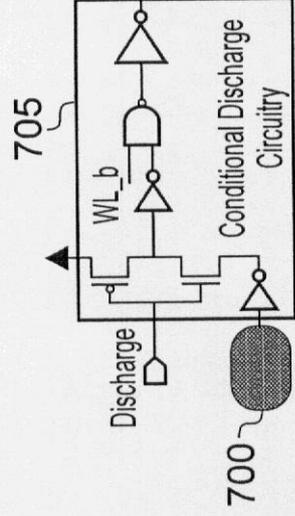


FIG. 10D

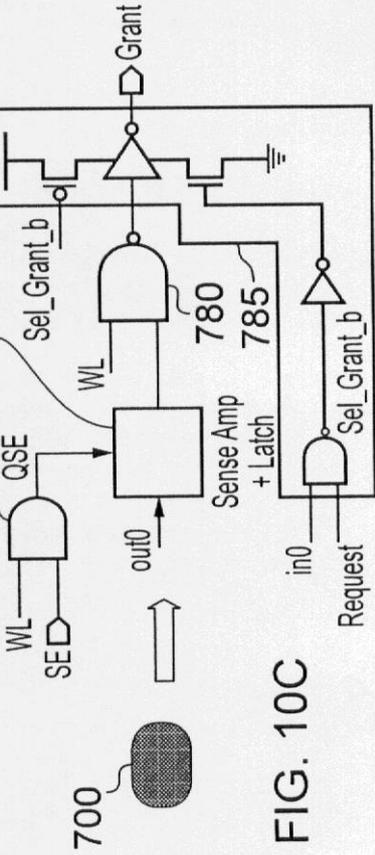


FIG. 10E

-  Storage element to hold Configuration
-  Storage element to hold Priority

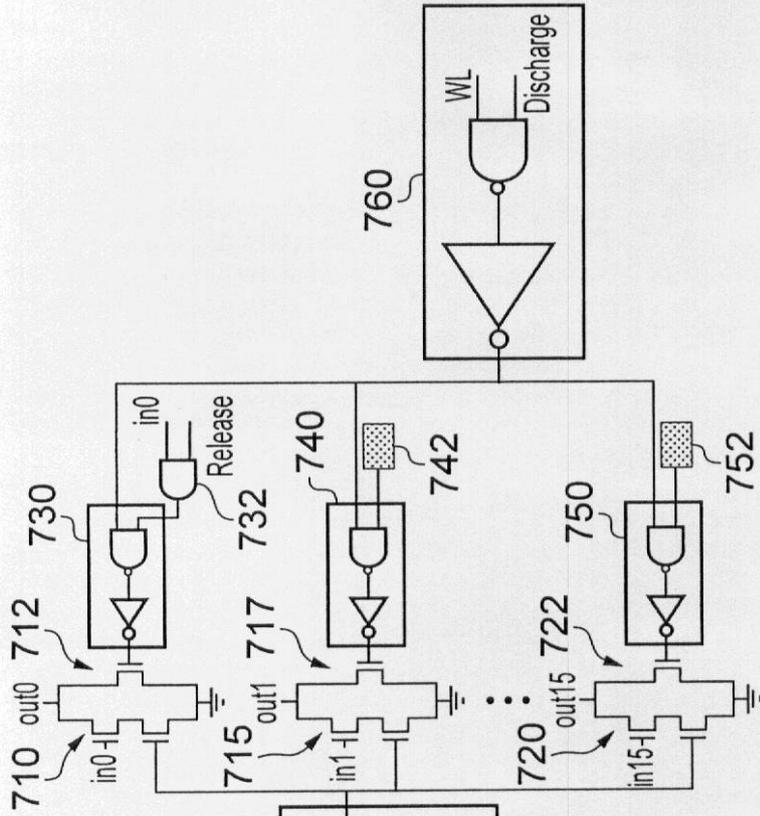


FIG. 11A

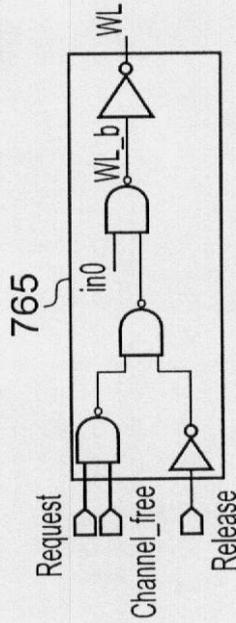
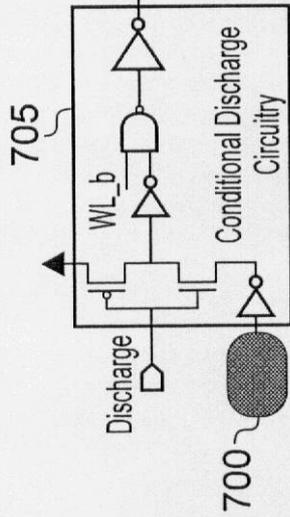


FIG. 11B

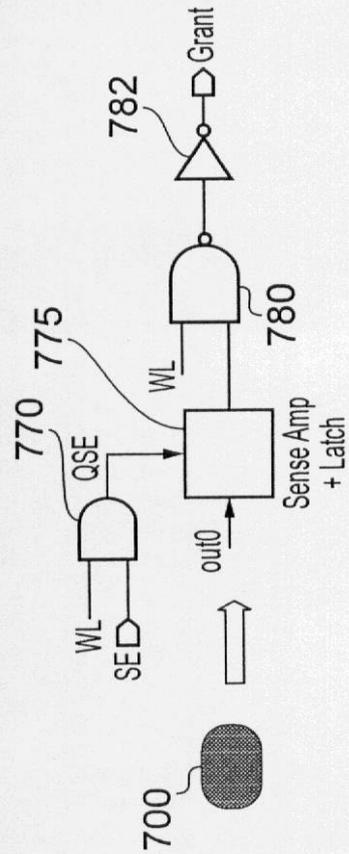


FIG. 11C

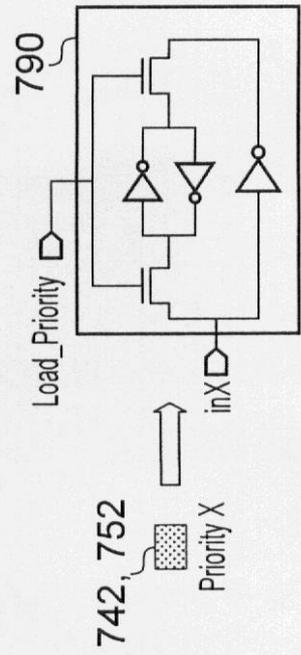


FIG. 11D

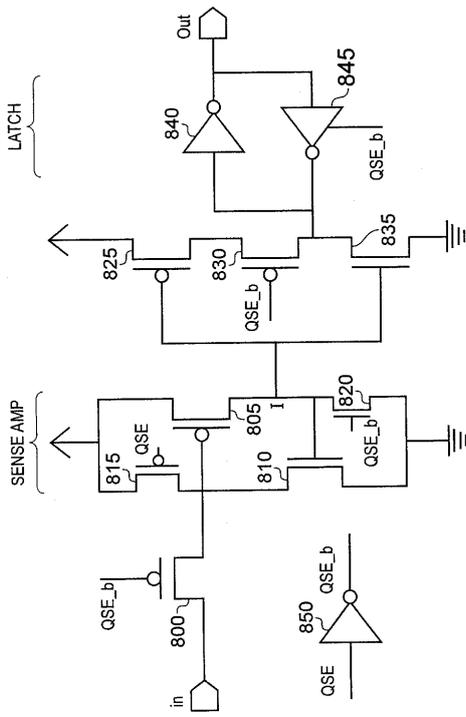


FIG. 12

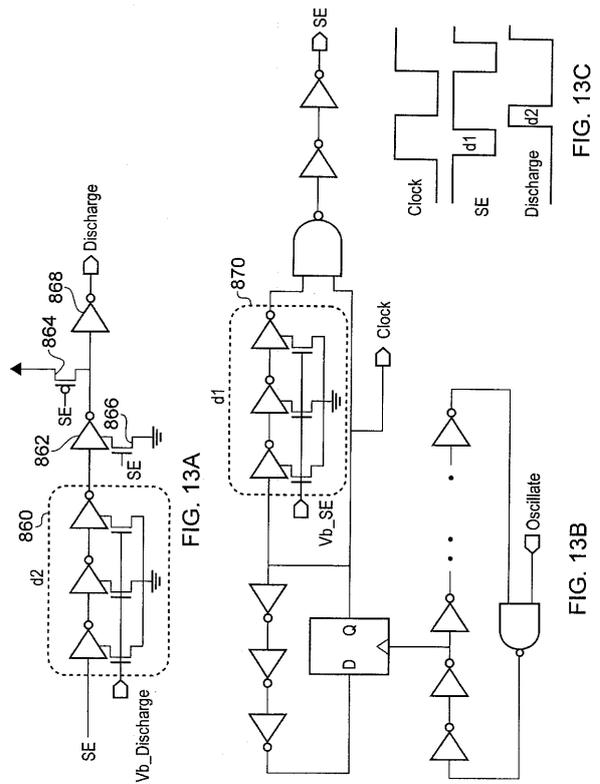


FIG. 13C

FIG. 13B

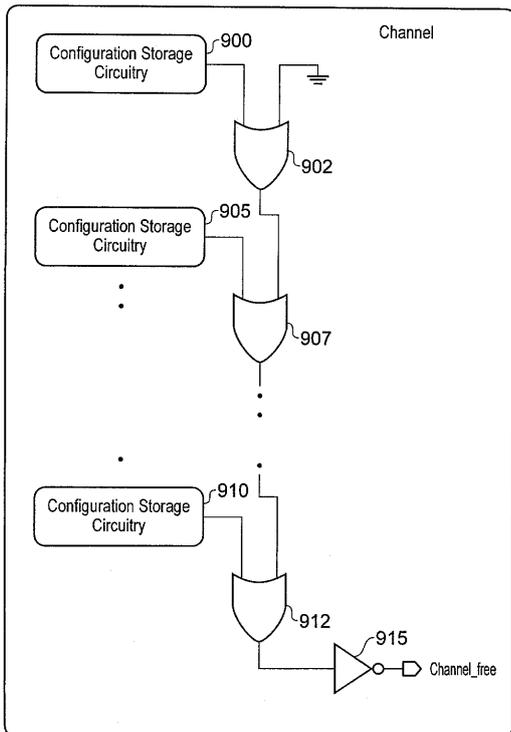


FIG. 14

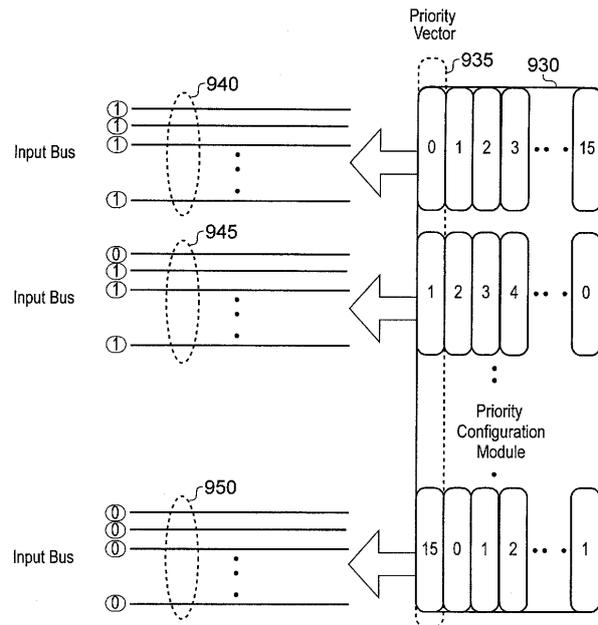


FIG. 15A

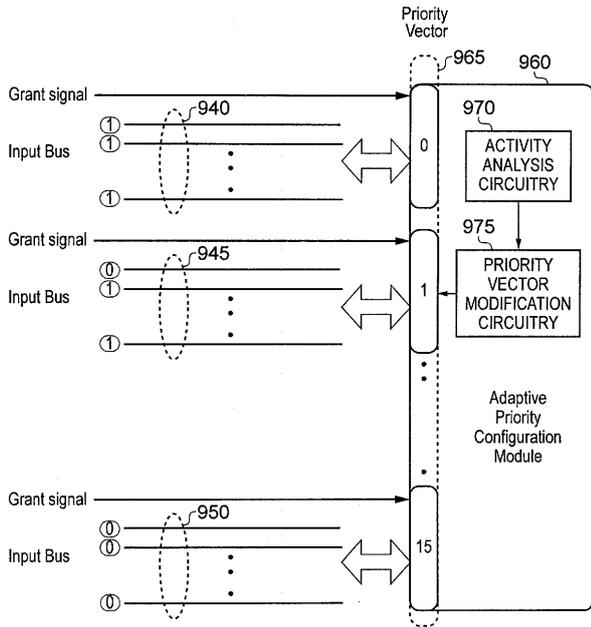


FIG. 15B

ENCODER:

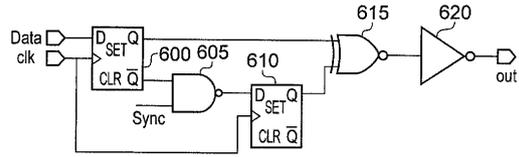


FIG. 16A

DECODER:

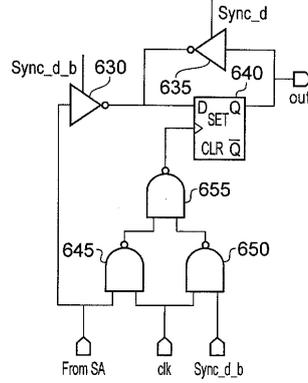


FIG. 16B

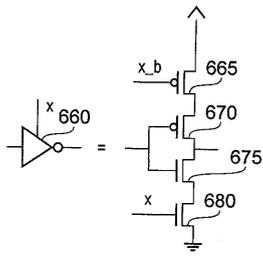


FIG. 16C

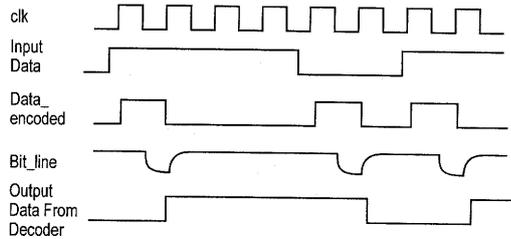


FIG. 16D

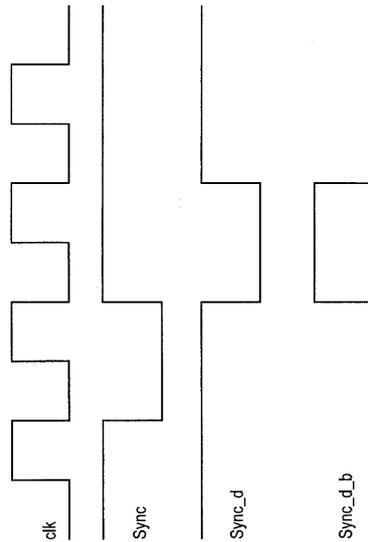


FIG. 16E

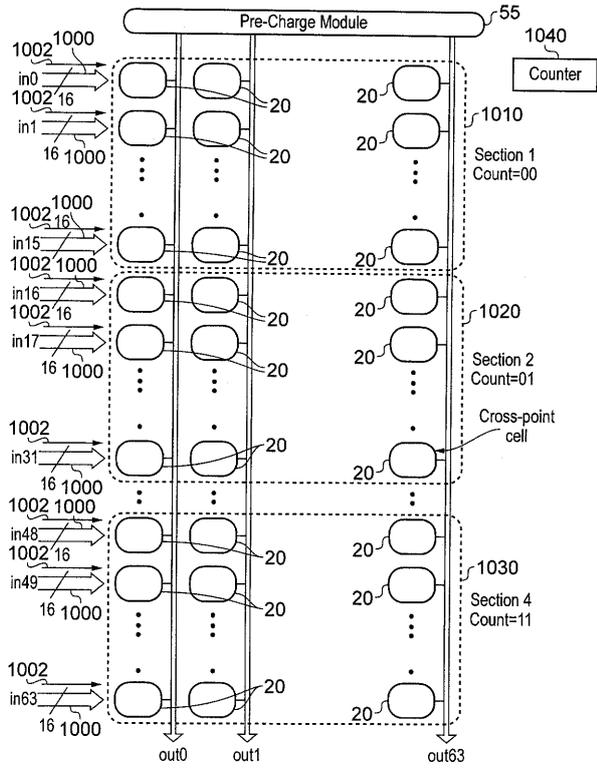


FIG. 17

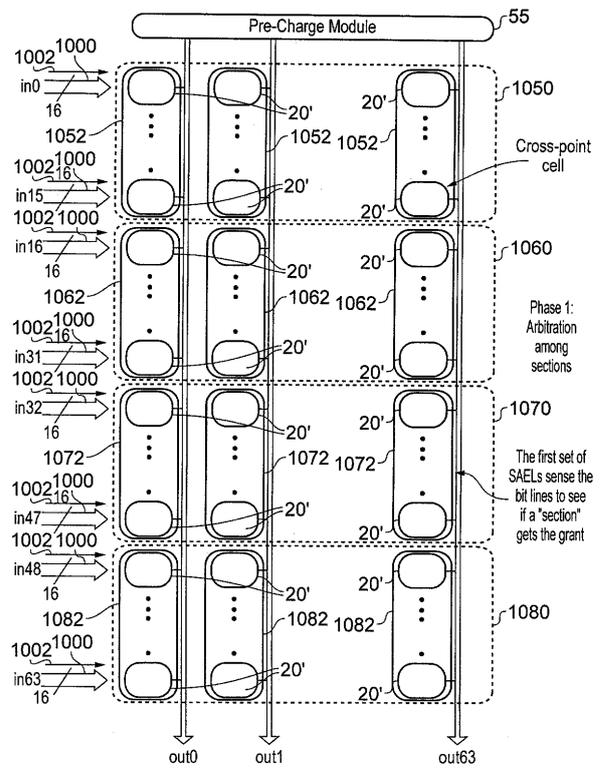


FIG. 18A

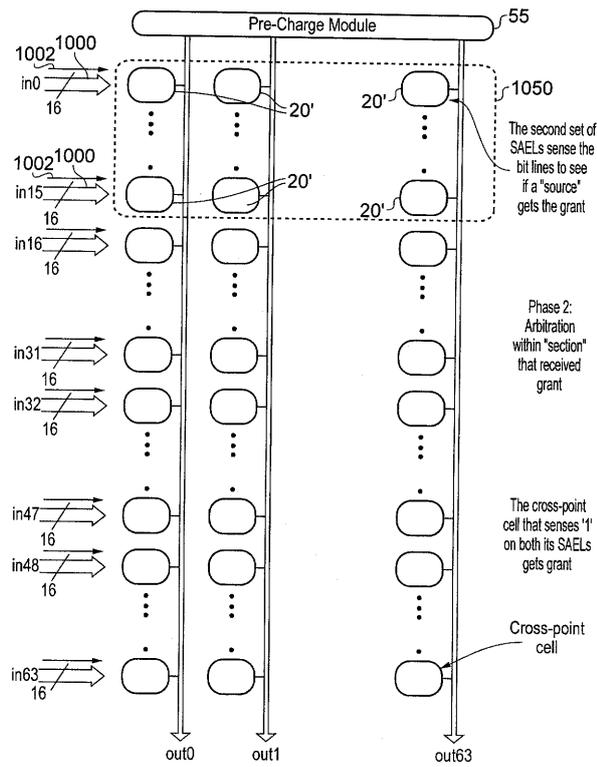


FIG. 18B

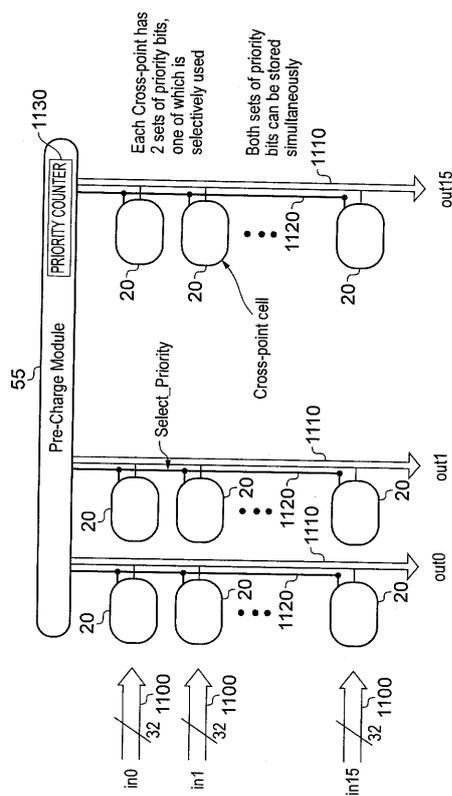


FIG. 19

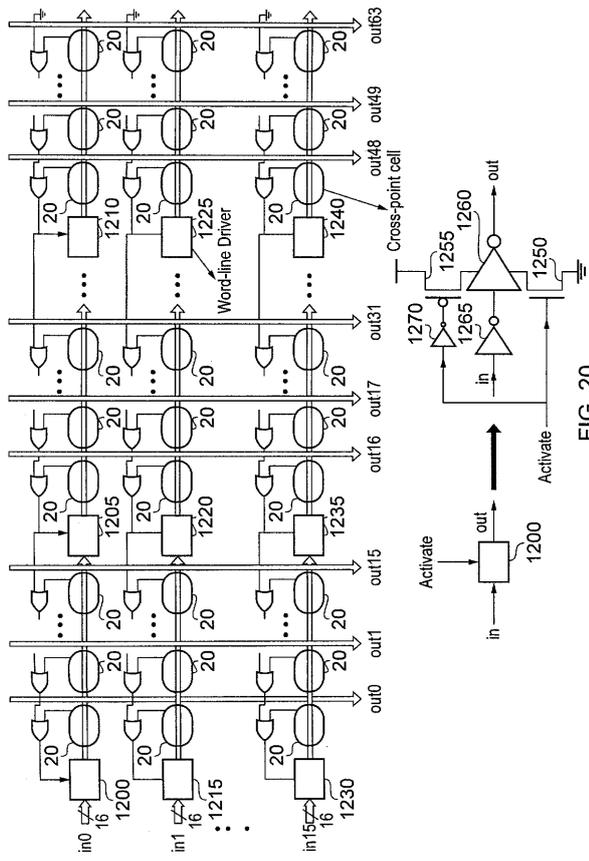


FIG. 20