(19) **Bundesrepublik Deutschland**
Deutsches Patent- und Markenamt

(10) **DE 199 41 196 B4** 2006.09.28

(12) **Patentschrift**

(21) Aktenzeichen: **199 41 196.4**
(22) Anmeldetag: **30.08.1999**
(43) Offenlegungstag: **23.03.2000**
(45) Veröffentlichungstag der Patenterteilung: **28.09.2006**

(51) Int Cl.[8]: ***G11C 7/00*** (2006.01)

(30) Unionspriorität:
09/156,516 17.09.1998 US

(73) Patentinhaber:
National Semiconductor Corp.(n.d.Ges.d.Staates Delaware), Santa Clara, Calif., US

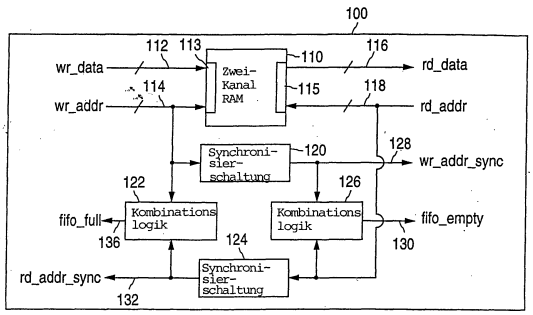
(74) Vertreter:
BOEHMERT & BOEHMERT, 80336 München

(72) Erfinder:
Wen, Sheung-Fan, Sunnyvale, Calif., US

(56) Für die Beurteilung der Patentfähigkeit in Betracht gezogene Druckschriften:
US 55 55 524 A

(54) Bezeichnung: **Zweikanal-FIFO mit synchronisierten Lese- und Schreibzeigern**

(57) Hauptanspruch: FIFO-Speicher mit folgenden Merkmalen:
ein Speicher (110), der einen FIFO-Stapel speichern kann;
ein Schreibkanal (112, 114), der gestützt auf einen Schreibadreßzeiger dem FIFO Stapel Elemente hinzufügen kann, wobei der Schreibkanal (112, 114) in einem ersten Taktbereich arbeitet;
ein Lesekanal (116, 118), der gestützt auf einen Leseadreßzeiger Elemente von dem FIFO-Stapel lesen kann, wobei der Lesekanal (116, 118) in einem zweiten Taktbereich arbeitet, der sich von dem ersten Taktbereich unterscheidet;
eine erste Synchronisierschaltung (120), die mit dem Schreibkanal (112, 114) betrieblich gekoppelt und so konfiguriert ist, daß sie den Schreibadreßzeiger empfängt und den Schreibadreßzeiger zu dem zweiten Taktbereich synchronisiert; und
eine zweite Synchronisierschaltung (124), die mit dem Lesekanal (116, 118) betrieblich gekoppelt und so konfiguriert ist, daß sie den Leseadreßzeiger empfängt und den Leseadreßzeiger zu dem ersten Taktbereich synchronisiert, wobei
die erste und die zweite Synchronisierschaltung (120, 124) jeweils ein erstes Zeitsteuer-Flip-Flop (420; 520) und...



Beschreibung

[0001] Die Erfindung betrifft im allgemeinen die Synchronisierung von Adreßzeigern über mehrere Taktbereiche hinweg und insbesondere einen Zweikanal-FIFO-Speicher (FIFO = First-In-First-Out), der einen Lese- und einen Schreibzeiger über mehrere Taktbereiche (Domänen) synchronisiert.

[0002] Ein FIFO-Speicher ist ein allgemein bekannter Speichertyp, der zahlreiche Anwendungen in elektronischen Schaltungen und Systemen findet. Ein FIFO-Speicher speichert Elemente in einem Stapel, wobei die ältesten Elemente als erste entnommen werden. In vielen Anwendungen kann ein Prozeß dem Stapel Elemente hinzufügen. Dieser Prozeß wird als Schreibprozeß bezeichnet. Ein anderer Prozeß kann Elemente von dem Stapel entnehmen. Dieser Prozeß wird als Leseprozeß bezeichnet. Der Schreibprozeß muß einen Adreßzeiger aufrechterhalten, damit er dem Stapel Elemente hinzufügen kann. Ähnlich muß der Leseprozeß einen Adreßzeiger aufrechterhalten, damit er Elemente aus dem Stapel entnehmen kann.

[0003] In vielen Anwendungen wird der FIFO-Speicher mit Hilfe eines Zweikanal-RAM (DPRAM = Dual Port RAM) realisiert. Ein Kanal oder Port wird von dem Schreibprozeß verwendet, und der andere Kanal oder Port wird von dem Leseprozeß verwendet. Der Schreibprozeß beginnt mit der Speicherung eines Elements bei der niedrigsten verfügbaren Speicherstelle. Der Schreibprozeß fügt dann bei aufeinanderfolgenden Speicherstellen Elemente hinzu, indem ein Schreibzeiger inkrementiert wird. Wenn der Schreibprozeß die höchste verfügbare Speicherzelle erreicht, wird der Schreibzeiger inkrementiert, damit er zu der niedrigsten verfügbaren Speicherstelle zurückkehrt. Der FIFO-Speicher arbeitet somit umlaufend.

[0004] Der Leseprozeß beginnt mit der Entnahme des Elementes bei der niedrigsten verfügbaren Speicherstelle. Der Leseprozeß setzt sich fort, indem er Elemente bei nachfolgenden Speicherstellen entnimmt, wobei der Lesezeiger inkrementiert wird. Wenn der Lesezeiger bei dem Schreibzeiger ankommt, ist der Speicher leer, und der Leseprozeß entnimmt keine weiteren Elemente von dem FIFO-Stapel. Wenn der Schreibzeiger bei dem Lesezeiger ankommt, ist der Speicher voll, und der Schreibprozeß beendet das Hinzufügen von Elementen.

[0005] In vielen Anwendungen arbeiten der Schreibprozeß und der Leseprozeß in verschiedenen Taktbereichen. Es muß daher eine Schaltung vorgesehen werden, um den Schreibzeiger in einem Taktbereich zu erzeugen und den Lesezeiger in dem anderen Taktbereich zu erzeugen. Zusätzlich muß die Schaltung die Schreib- und Lesezeiger über die Taktbereiche hinweg synchronisieren, damit der Schreibprozeß nur leeren Speicherstellen ein Element hinzufügt und der Leseprozeß Elemente aus gültigen Speicherstellen entnimmt. Diese Synchronisierung kann erhebliche Verzögerungen zwischen den Lese- und Schreibprozessen einführen. Zusätzlich geht die Synchronisierung häufig mit komplizierten, anwendungsspezifischen Schaltungen einher, die üblicherweise in den Standardbibliotheken der CMOS-Zellen nicht verfügbar sind. Die Entwicklung solcher anwendungsspezifischer Schaltungen führt zu zusätzlichen Kosten und zusätzlichem Zeitaufwand.

Stand der Technik

[0006] Die Druckschrift US 5,555,524 offenbart einen Zweikananl-FIFO-Speicher mit Synchronisierschaltungen und mit Füllstandsanzeige-Schaltungen. Die Speicheradressen liegen in Gray-Code vor, während die Synchronisierschaltungen mit einem einfachen D-Flip-Flop vorgesehen sind, der das Adreßsignal hält. In der Füllstandsanzeige-Schaltung verzögert ein D-Flip-Flop das Lese- oder Schreib-Freigabesignal, wobei die Füllstandsanzeige-Schaltung den Füllstand mit einer Genauigkeit von 1 angibt. Die Implementierung der Synchronisierungsschaltung bietet nur dann korrekte synchronisierte Adreßsignale, wenn das Taktsignal des Ziel-Taktbereichs rechtzeitig die Synchronisierungsschaltung triggert. Weder bietet der Zweikanal-FIFO-Speicher Adreßsignale, die jederzeit korrekt sind, noch exakte Angaben bezüglich des Füllstands. Ferner ist das Gray-Format zwingend für das Adreßsignal, wodurch gegebenenfalls Codeumsetzer verwendet werden müssen.

Aufgabenstellung

[0007] Es ist daher Aufgabe der Erfindung, einen FIFO-Speicher bzw. einen Synchronisierungsmechanismus vorzusehen, der jederzeit korrekte Adreßsignale bzw. Füllstandssignale vorsieht und dessen Adreßsignalformat nicht auf ein einziges Format beschränkt ist. Diese Aufgabe wird durch den FIFO-Speicher nach Anspruch 1, die Synchronisierschaltung nach Anspruch 8, das Verfahren nach Anspruch 11 und durch den Signalsatz nach Anspruch 15 erreicht.

[0008] Gemäß eines Aspekts der Erfindung umfaßt ein FIFO einen Speicher, einen Schreibkanal, einen Lesekanal und eine erste und eine zweite Synchronisierschaltung. Der Speicher ist so konfiguriert, daß er einen FIFO-Stapel speichert. Der Schreibkanal ist so konfiguriert, daß er gestützt auf einen Schreibadreßzeiger dem FIFO-Stapel Elemente hinzufügt. Der Schreibkanal arbeitet in einem ersten Taktbereich. Der Lesekanal ist so konfiguriert, daß er gestützt auf einen Leseadreßzeiger Elemente von dem FIFO-Stapel liest.

[0009] Der Lesekanal arbeitet in einem zweiten Taktbereich, der sich vom ersten Taktbereich unterscheidet. Die erste Synchronisierschaltung ist mit dem Schreibkanal betrieblich gekoppelt und so konfiguriert, daß sie den Schreibadreßzeiger empfängt und den Schreibadreßzeiger mit dem zweiten Taktbereich synchronisiert. Die zweite Synchronisierschaltung ist mit dem Lesekanal betrieblich gekoppelt und so konfiguriert, daß sie den Leseadreßzeiger empfängt und den Leseadreßzeiger mit dem ersten Taktbereich synchronisiert.

[0010] Gemäß eines weiteren Aspekts der Erfindung umfaßt eine Synchronisierschaltung, die sich zum Koordinieren von Adreßzeigern über Taktbereiche hinweg eignet, ein erstes und ein zweites Zeitsteuer-Flip-Flop und eine Inversionsschaltung. Das erste Zeitsteuer-Flip-Flop ist so konfiguriert, daß es erste Zeitsignale erzeugt. Das erste Zeitsteuer-Flip-Flop arbeitet in einem ersten Taktbereich. Die Inversionsschaltung ist mit dem ersten Zeitsteuer-Flip-Flop betrieblich gekoppelt und so konfiguriert, daß sie invertierte Zeitsignale gestützt auf die ersten Zeitsignale erzeugt. Das zweite Zeitsteuer-Flip-Flop ist mit dem Invertierer betrieblich gekoppelt und so konfiguriert, daß es zweite Zeitsignale gestützt auf die invertierten Zeitsignale erzeugt. Das zweite Zeitsteuer-Flip-Flop arbeitet in einem zweiten Taktbereich, der sich vom ersten Taktbereich unterscheidet.

[0011] Die Erfindung ist im folgenden anhand bevorzugter Ausführungsformen mit Bezug auf die Zeichnungen näher erläutert. In den Figuren zeigt:

[0012] Fig. 1 ein Blockdiagramm einer bevorzugten Ausführungsform eines FIFO-Speichers;

[0013] Fig. 2 ein Flußdiagramm eines bevorzugten Schreibprozesses, der in dem FIFO-Speicher der Fig. 1 abläuft;

[0014] Fig. 3 ein Flußdiagramm eines bevorzugten Leseprozesses, der in dem FIFO-Speicher der Fig. 1 abläuft;

[0015] Fig. 4 einen Schaltplan einer bevorzugten Ausführungsform einer Adreßsynchronisierschaltung;

[0016] Fig. 5 einen Schaltplan einer anderen bevorzugten Ausführungsform einer Adreßsynchronisierschaltung; und

[0017] Fig. 6 ein Zeitablaufdiagramm der Signale aus der Schaltung der Fig. 5.

Ausführungsbeispiel

[0018] Mit Bezug auf Fig. 1 wird im folgenden eine bevorzugte Ausführungsform eine FIFO-Speichers **100** beschrieben. Der FIFO-Speicher **100** umfaßt ein DPRAM **110**. Ein Kanal oder Port **113** wird zum Schreiben von Daten in das DPRAM **110** verwendet, und ein Kanal oder Port **115** wird zum Lesen von Daten aus dem DPRAM **110** verwendet. Der Kanal **113** und der Kanal **115** umfassen jeweils einen Adreßkanal und einen Datenkanal. Die Leitungen wr_addr **114** sind mit dem Kanal **113** verbunden und werden zum Auswählen einer Speicherstelle in dem DPRAM **110** verwendet. Die Leitungen wr_data **112** sind ebenfalls mit dem Kanal **113** verbunden. Diese Leitungen werden dazu verwendet, während einer Schreiboperation Datensignale an die ausgewählte Adreßstelle zu liefern. Gemeinsam werden die Leitungen wr_addr **114** und wr_data **112** dazu verwendet, einem FIFO-Stapel in dem DPRAM **110** ein Element hinzuzufügen.

[0019] Die Leitungen rd_addr **118** sind mit dem Kanal **115** verbunden und werden dazu verwendet, eine Speicherstelle in dem DPRAM **110** auszuwählen. Die Leitungen rd_data **116** sind ebenfalls mit dem Kanal **115** verbunden. Diese Leitungen werden dazu verwendet, während einer Leseoperation Datensignale von einer ausgewählten Adreßstelle zu empfangen. Gemeinsam werden die Leitungen rd_addr **118** und rd_data **116** dazu verwendet, ein Element von dem FIFO-Stapel zu entnehmen.

[0020] Bei einer bevorzugten Ausführungsform sieht das DPRAM **110** nur die Schreibfunktion für den Schreibkanal **113** und nur die Lesefunktion für den Lesekanal **115** vor. Eine Vielzahl im Handel erhältlicher DPRAMs bieten jedoch sowohl die Schreib- als auch die Lesefunktion für beide Kanäle, und solche DPRAMs eignen sich als Alternativen zum Realisieren der Erfindung.

[0021] Der FIFO-Speicher **100** ist so konfiguriert, daß über den Schreibkanal **113** dem Stapel Elemente hinzugefügt werden können und daß über den Lesekanal **115** dieselben Element von dem Stapel entnommen werden können. Elemente werden über den Schreibkanal **113** in aufeinanderfolgenden Speicherstellen hinzugefügt. Das erste Element wird in die Speicherstelle „000h" geschrieben, das nächste Element wird in die Speicherstelle „001h" geschrieben usw. Schließlich fügt der Schreibkanal **113** der höchsten Speicherstelle ein Element hinzu (z.B. „FFFh" in einem 4K Speicher). Nach diesem Schreibvorgang wird der Schreibzeiger inkrementiert, so daß er zur ersten Speicherstelle „000h" zurückkehrt. Der Lesezeiger arbeitet auf dieselbe Weise.

[0022] Um sicherzustellen, daß ein Schreibkanal **113** nicht versucht, ein Element in eine Speicherstelle hinzuzufügen, die noch nicht gelesene Daten enthält (d.h. ein Element, das noch nicht über den Lesekanal **115** entnommen wurde), wird ein FIFO-Voll-Signal erzeugt. Ähnlich wird, um sicherzustellen, daß der Le-

sekanal **115** nicht versucht, ein Element aus einer Speicherstelle zu lesen, in die keine Daten geschrieben wurden (d.h. eine Speicherstelle, in die von dem Schreibkanal **113** keine Daten geschrieben wurden), ein FIFO-Leer-Signal erzeugt.

[0023] Der Schreibkanal **113** und der Lesekanal **115** können in verschiedenen Taktbereichen (Taktdomänen) arbeiten. Deshalb werden der Lese- und der Schreibadreßzeiger über die Taktbereiche hinweg synchronisiert. Die Lese- und Schreibadreßzeiger werden dazu verwendet, FIFOvoll(fifo_full) und FIFO-leer (fifo_empty)-Signale zu erzeugen.

[0024] Im einzelnen sind die Leitungen wr_addr **114** mit einer Synchronisiereinrichtung **120** und einer Schreiblogikschaltung **122** verbunden, und die Leitungen rd_addr **118** sind mit einer Synchronisiereinrichtung **124** und einer Leselogikschaltung **126** verbunden. Die Synchronisiereinrichtung **120** empfängt den aktuellen Schreibzeiger über die Leitungen wr_addr **114**. Nach einer ausreichenden Verzögerung zum Koordinieren der Taktbereiche liefert die Synchronisiereinrichtung **120** den Schreibzeiger an die Leselogikschaltung **126** über die Leitungen **128**. Ähnlich empfängt die Synchronisiereinrichtung **124** den aktuellen Lesezeiger über die Leitungen rd_addr **118**. Nach einer ausreichenden Verzögerung zum Koordinieren der Taktbereiche liefert die Synchronisiereinrichtung **124** den Lesezeiger an die Schreiblogikschaltung **122** über die Leitungen **132**. Die Leselogikschaltung **126** und die Schreiblogikschaltung **122** erzeugen das fifo_empty-Signal **130** bzw. das fifo_full-Signal **136**.

[0025] Eine bevorzugte Art zum Erzeugen des fifo_empty-Signals **130** und des fifo_full-Signals **136** besteht darin, auf der MSB-Seite (Seite des höchstwertigen Bits) der Signale wr_addr und rd_addr ein zusätzliches Bit hinzuzufügen und dieses als ein Teil der Signale wr_addr bzw. rd_addr zu behandeln, wenn diese inkrementiert werden. Wenn z.B. angenommen wird, daß wr_addr und rd_addr 12 Bit breit sind, macht das Hinzufügen eines 13. Bits auf der MSB-Seite diese 13 Bit breit. Während des Zurücksetzens werden wr_addr [12:0] und rd_addr [12:0] alle auf null zurückgesetzt. wr_addr wird inkrementiert, bis fifo_full aktiviert (logisch wahr gesetzt) wird; rd_addr wird inkrementiert, bis fifo-empfy aktiviert wird. Man beachte, daß wr_addr [12:12] (das dreizehnte Bit des Signals wr_addr) und rd_addr [12:12] (das dreizehnte Bit des Signals rd_addr) während geradzahliger Durchgänge des FIFO-Speichers den Wert null haben und während ungeradzahliger Durchgängen den Wert eins haben. Aus der folgenden logischen Gleichung ergibt sich daher das Signal fifo_full:
(wr_addr [12:12] XOR rd_addr_sync [12:12]) UND (wr_addr [11:0] XNOR rd_addr_sync [11:0]);
fifo_empty ergibt sich aus der logischen Gleichung:
rd_addr [12:0] XNOR wr_addr_sync [12:0],
wobei XOR exklusiv-ODER und XNOR exklusiv-NICHT-ODER bezeichnet.

[0026] Die Schreiblogikschaltung **122** und die Leselogikschaltung **126** implementieren diese Logikfunktionen zum Erzeugen der Signale fifo_full bzw. fifo_empty.

[0027] Bei der Initialisierung werden der Lese- und der Schreibadreßzeiger auf „000h" gesetzt. Wenn Elemente hinzugefügt werden, wird der Schreibzeiger inkrementiert. Demzufolge wird die Leselogikschaltung **126** das Signal fifo_empty deaktivieren (logisch falsch setzen). Abhängig davon werden Elemente von dem FIFO gelesen, und der Lesezeiger wird inkrementiert. Wenn der Lesezeiger den Schreibzeiger einholt, wird das Signal fifo_empty aktiviert, bis dem FIFO-Stapel zusätzliche Elemente hinzugefügt worden sind.

[0028] Wie oben erwähnt, wird der Schreibzeiger, wenn er die höchste Speicherstelle erreicht, inkrementiert, um zur ersten Speicherstelle zurückzukehren. Wenn zusätzliche Elemente hinzugefügt werden, kann der Schreibadreßzeiger schließlich den Leseadreßzeiger einholen. Wenn zusätzliche Elemente hinzugefügt würden, würden diese Elemente andere Elemente überschreiben, die noch nicht von dem Stapel gelesen worden sind. Um dies zu vermeiden, wird das Signal fifo_full aktiviert. Es darf nicht weitergeschrieben werden, bis weitere Elemente von dem FIFO-Stapel entnommen worden sind.

[0029] Wie oben erwähnt, ist der FIFO-Stapel in dem DPRAM **110** realisiert. Diese Konfiguration eignet sich zur Verwendung mit einem DPRAM jeder üblichen Größe und Breite.

[0030] Mit Bezug auf Fig. 2 ist im folgenden der Prozeß zum Hinzufügen von Elementen zu dem FIFO-Stapel mit weiteren Einzelheiten beschrieben. Im Schritt **210** ermittelt das FIFO, ob das Signal fifo_full aktiviert ist. Wenn ja, bleibt das FIFO beim Schritt **210**. Anderenfalls geht das FIFO zum Schritt **212** weiter. Hier inkrementiert das FIFO den Schreibzeiger, so daß in der nächsten Speicherstelle ein Element hinzugefügt werden kann. Im Schritt **214** fügt das FIFO dem Stapel bei der von dem Schreibzeiger ausgewählten Adreßstelle ein Element hinzu. Im Schritt **216** wird der neue Schreibzeiger über den Taktbereich hinweg übergeben. Dies erlaubt es dem Leseprozeß, das nächste Element zu entnehmen. Das FIFO geht dann zum Schritt **210** zurück.

[0031] Mit Bezug auf Fig. 3 ist der Prozeß zum Entnehmen eines Elements aus dem FIFO mit weiteren Einzelheiten beschrieben. Im Block **310** ermittelt das FIFO, ob das Signal fifo_empty aktiviert ist. Wenn ja, bleibt das FIFO im Block **310**. Anderenfalls geht das

FIFO zum Block **312** weiter. Hier liest das FIFO Daten aus der Speicherstelle, die durch den Lesezeiger gewählt wurde. Im Block **314** inkrementiert das FIFO den Lesezeiger. Im Block **316** wird ein neuer Lesezeiger über den Taktbereich hinweg übergeben. Dies erlaubt es dem Schreibprozeß, bei der geleerten Speicherstelle ein Element hinzuzufügen. Das FIFO geht dann zum Block **310** zurück.

[0032] Mit Bezug auf Fig. 4 ist im folgenden eine bevorzugte Ausführungsform einer Adreßsynchronisierschaltung **410** (oder Adreßsynchronisiereinrichtung) beschrieben. Die Adreßsynchronisierschaltung **410** kann dazu verwendet werden, die Synchronisiereinrichtung **120** und die Synchronisiereinrichtung **124** der Fig. 1 zu realisieren. Die Adreßsynchronisierschaltung **410** ist in zwei Taktbereiche (Taktdomänen) aufgeteilt. Insbesondere arbeitet der Block **412** auf der Basis eines Takts **411** (clock_src), während der Block **414** auf der Basis eines Takts **413** (clock_dest) arbeitet. Der Takt **411** und der Takt **413** sind zueinander asynchron und können verschiedene Frequenzen haben. Demnach definiert der Block **412** einen Taktbereich gestützt auf den Takt **411**, und der Block **414** definiert einen Taktbereich gestützt auf den Takt **413**.

[0033] Die Adreßsynchronisierschaltung **410** dient dazu, den Übergang eines Adreßzeigers **423** (addr_ptr) von dem Taktbereich des Blocks **412** zu einem synchronisierten Adreßzeiger **439** (addr_ptr_sync) in dem Taktbereich des Blocks **414** zu koordinieren.

[0034] Die Schaltung beginnt ihren Betrieb bei Empfang eines Signals reset_z durch ein UND-Gatter **416**. Das Signal reset_z wird dazu verwendet, den normalen Schaltungsbetrieb zu starten. Während des Betriebs der Synchronisierschaltung **410** muß das Signal reset_z auf einem hohen Zustand bleiben. Das UND-Gatter **416** empfängt auch ein Signal dest_out von dem Zeitsteuer-Flip-Flop **434**. Wenn das Signal reset_z und das Signal dest_out beide einen hohen Zustand haben, liefert das UND-Gatter **416** ein Signal mit hohem Pegel an das Zeitsteuer-Flip-Flop **418**. Das Zeitsteuer-Flip-Flop **418** wird von dem Takt **411** getriggert. Bei Beginn eines neuen Taktzyklus gibt das Zeitsteuer-Flip-Flop **418** das hochpegelige Signal von dem UND-Gatter **416** als das Signal src_sync1 weiter. Dieses Signal wird seinerseits an das Zeitsteuer-Flip-Flop **420** und das XOR-Gatter **422** weitergegeben. Das Zeitsteuer-Flip-Flop **420** arbeitet ähnlich wie das Zeitsteuer-Flip-Flop **418**. Bei dem Beginn eines neuen Taktzyklus gibt das Zeitsteuer-Flip-Flop **420** den Wert des Signals src_sync1 als das Signal src_out weiter. Wenn also das Signal sr_sync1 umschaltet (Übergang), stimmt der Wert des Signals src_out während einer Taktperiode nicht überein. Da diese Signale beide an das XOR-Gatter **422** übergeben werden, erzeugt dieses während dieser Taktperiode ein Signal mit hohem Pegel.

[0035] Das Ausgangssignal des XOR-Gatters **422** wird an den Steuereingang eines Multiplexers **424** angelegt. Der Multiplexer **424** empfängt auch den Wert des Zeigers, der von Halte-Flip-Flops **426** gehalten wird, sowie addr_ptr **423**, der den momentanen Zeigerwert angibt. Wenn der Multiplexer **424** ein Signal mit hohem Pegel von dem XOR-Gatter **422** empfängt, liefert er den Wert von addr_ptr **423** an die Halte-Flip-Flops **426**. Anderenfalls liefert er den Wert des Zeigers vom Ausgang der Halte-Flip-Flops **426** an den Eingang der Halte-Flip-Flops **426**. Der Zeigerwert vom Ausgang der Halte-Flip-Flops **426** wird über die Taktgrenze hinweg an den Multiplexer **438** übergeben.

[0036] Wiederum empfängt das Zeitsteuer-Flip-Flop **430** das Signal src_out vom Zeitsteuer-Flip-Flop **420**. Beim Beginn eines neuen Taktzyklus innerhalb des Taktbereichs des Blocks **414** gibt das Zeitsteuer-Flip-Flop **430** den Wert des Signals src_out als das Signal dest_sync1 weiter. Das Signal dest_sync1 wird an den Invertierer **432** geliefert. Das invertierte Signal dest_sync1 wird an das Zeitsteuer-Flip-Flop **434** und das XOR-Gatter **436** übergeben. Das Zeitsteuer-Flip-Flop **434** arbeitet ähnlich wie das Zeitsteuer-Flip-Flop **430**. Beim Beginn eines neuen Taktzyklus gibt das Zeitsteuer-Flip-Flop **434** den Wert des invertierten Signals dest_sync1 als das Signal dest_out weiter. Wenn also das invertierte Signal dest_sync2 sich ändert (Übergang) stimmt der Wert des Signals dest_out während einer Taktperiode nicht überein. Da beide Signale an das XOR-Gatter **436** geliefert werden, erzeugt dies während dieser Taktperiode ein Signal mit hohem Pegel.

[0037] Das Ausgangssignal des XOR-Gatters **436** wird an den Steuereingang des Multiplexers **438** übergeben. Der Multiplexer **438** empfängt auch den Wert des Zeigers, der momentan in Abtast-Flip-Flops **440** gehalten wird, sowie das Ausgangssignal der Halte-Flip-Flops **426**. Wenn der Multiplexer **438** ein Signal mit hohem Pegel von dem XOR-Gatter **436** empfängt, liefert er den Zeigerwert von den Halte-Flip-Flops **426**. Anderenfalls liefert er den Zeigerwert vom Ausgang der Abtast-Flip-Flops **440** an den Eingang der Abtast-Flip-Flops **440**. Der Zeigerwert vom Ausgang der Abtast-Flip-Flops **440** ist das Signal addr_ptr_sync. Dieses Signal folgt somit dem Signal addr_ptr. Das Signal addr_ptr_synch ist jedoch im Taktbereich des Blocks **414** gültig.

[0038] Die Synchronisierschaltung **410** kann mit den Schaltkreiselementen einer im Handel erhältlichen Standard-CMOS-Zellenbibliothek realisiert werden. Alle Gatter, Flip-Flops und Multiplexer sind allgemein erhältlich.

[0039] Im normalen Betrieb übersetzt die Adreßsynchronisierschaltung **410** das Signal addr_ptr **423** über die Taktbereiche hinweg in addr_ptr_sync **439**. Es ist jedoch möglich, daß das Zeitsteuer-Flip-Flop **418** in einen meta-stabilen Zustand gerät. Um eine Verfälschung der Daten aufgrund des meta-stabilen Zustands zu vermeiden, wird das Flip-Flop **420** Rücken an Rücken mit dem Flip-Flop **418** angeordnet, so daß das Flip-Flop **418** genug Zeit hat, sich von einem meta-stabilen Zustand zu erholen, bevor Daten in das Flip-Flop **420** eingehen. Diese Anordnung kann auch als Doppelsynchronisierung bezeichnet werden und schafft einen stabilen Betrieb, wenn die Signalverzögerung vom Zeitsteuer-Flip-Flop **418** zum Halte-Flip-Flop **426** geringer als die Taktperiode des Signals clk_src ist und wenn die Signalverzögerung vom Zeitsteuer-Flip-Flop **430** zum Abtast-Flip-Flop **440** geringer als die Taktperiode des Signals clk_dest ist. Die Signalverzögerung vom Zeitsteuer-Flip-Flop **418** zum Halte-Flip-Flop **426** umfaßt: t (Takt des Flip-Flops **418** zu Q + t (Erholung des Flip-Flops **418** aus meta-stabilem Zustand) + t (XOR-Gatter) + t (Multiplexer **424**) + t (Einrichten des Flip-Flops **426**) + t (Leitungsverzögerungen).

[0040] Die Signalverzögerung vom Zeitsteuer-Flip-Flop **430** zum Abtast-Flip-Flop **440** umfaßt: t (Takt des Flip-Flops **430** zu Q + t (Erholung des Flip-Flops **430** aus meta-stabilem Zustand) + t (Invertierer **432**) + t (XNOR-Gate **436**) + t (Multiplexer **438**) + t (Einrichten des Flip-Flops **440**) + t (Leitungsverzögerungen).

[0041] Wenn diese Bedingungen nicht erfüllt werden können, kann sich ein meta-stabiler Zustand ergeben, der undefinierte Übergänge verursachen kann. Dieser potentielle meta-stabile Zustand kann vermieden werden, indem der Adreßsynchronisierschaltung **410** zusätzliche Zeitsteuer-Flip-Flops hinzugefügt werden.

[0042] Mit Bezug auf Fig. 5 ist eine weitere bevorzugte Ausführungsform einer Adreßsynchronisierschaltung **510** (oder Adreßsynchronisiereinrichtung) beschrieben. Die Adreßsynchronisierschaltung **510** kann dazu verwendet werden, die Synchronisiereinrichtung **120** und die Synchronisiereinrichtung **124** der Fig. 1 zu realisieren. Die Adreßsynchronisierschaltung **510** arbeitet weitgehend ähnlich wie die Adreßsynchronisierschaltung **410** der Fig. 4. Die Adreßsynchronisierschaltung **510** verwendet jedoch zwei zusätzliche Zeitsteuer-Flip-Flops. Diese Zeitsteuer-Flip-Flops reduzieren die Wahrscheinlichkeit des oben beschriebenen metastabilen Zustands und führen nur eine minimale zusätzliche Synchronisierungsverzögerung ein.

[0043] Die Adreßsynchronisierschaltung **510** wird in zwei Taktbereiche aufgeteilt. Insbesondere arbeitet der Block **512** auf der Basis eines Takts **511** (clock_src), während der Block **514** auf der Basis eines Takts **513** (clock_dest) arbeitet. Der Takt **511** und der Takt **513** sind zueinander asynchron und können verschiedene Frequenzen haben. Der Block **512** definiert somit einen Taktbereich gestützt auf den Takt **511**, und der Block **514** definiert einen Taktbereich gestützt auf den Takt **513**.

[0044] Die Adreßsynchronisierschaltung **510** dient dazu, die Überleitung eines Adreßzeigers **523** (addr_ptr) vom Taktbereich des Blocks **512** zu einem synchronisierten Adreßzeiger **539** (addr_ptr_sync) im Taktbereich des Blocks **514** zu koordinieren.

[0045] Die Schaltung beginnt ihren Betrieb bei Empfang eines Signals reset_z durch das UND-Gatter **516**. Das Signal reset_z wird dazu verwendet, den normalen Schaltkreisbetrieb zu starten. Während des Betriebs der Synchronisierschaltung **510** muß das Signal reset_z einen hohen Zustand behalten. Das UND-Gatter **516** empfängt auch ein Signal dest_out von dem Zeitsteuer-Flip-Flop **534**. Wenn sowohl das Signal reset_z als auch das Signal dest_out einen hohen Zustand haben, liefert das UND-Gatter **516** ein Signal mit hohem Pegel an das Zeitsteuer-Flip-Flop **518**. Das Zeitsteuer-Flip-Flop **518** wird durch den Takt **511** getriggert. Bei Beginn eines neuen Taktzyklus gibt das Zeitsteuer-Flip-Flop **518** das Signal mit hohem Pegel von dem UND-Gatter **516** als das Signal src_sync1 weiter. Dieses Signal wird seinerseits an das Zeitsteuer-Flip-Flop **519** übergeben. Das Zeitsteuer-Flip-Flop **519** arbeitet ähnlich wie das Zeitsteuer-Flip-Flop **518**. Bei Beginn eines neuen Taktzyklus gibt das Zeitsteuer-Flip-Flop **519** den Wert des Signals scr_sync1 als das Signal src_sync2 weiter. Dieses Signal wird seinerseits an das Zeitsteuer-Flip-Flop **520** und das XOR-Gatter **522** weitergegeben. Das Zeitsteuer-Flip-Flop **520** arbeitet ähnlich wie das Zeitsteuer-Flip-Flop **518**. Bei Beginn eines neuen Taktzyklus gibt das Zeitsteuer-Flip-Flop **520** den Wert des Signals src_sync2 als das Signal src_out weiter. Wenn daher das Signal src_sync2 umschaltet (Übergang), stimmt der Wert des Signals src_out während einer Taktperiode nicht überein. Da beide Signale an das XOR-Gatter **522** übergeben werden, erzeugt dieses während dieser Taktperiode ein Signal mit hohem Pegel.

[0046] Das Ausgangssignal des XOR-Gatters **522** wird an den Steuereingang eines Multiplexers **524** übergeben. Der Multiplexer **524** empfängt auch den Wert des Zeigers, der von Halte-Flip-Flops **526** gehalten wird (addr_held), sowie das Signal addr_ptr **523**, das den momentanen Zeigerwert angibt. Wenn der Multiplexer **524** ein Signal mit hohem Pegel von dem XOR-Gatter **522** empfängt, liefert er den Wert des Signals addr_ptr an Halte-Flip-Flops **526**. Anderenfalls liefert er den Zeigerwert vom Ausgang der Halt-Flip-Flops **526** an den Eingang der Halte-Flip-Flops **526**. Der Zeigerwert vom Ausgang der

Halte-Flip-Flops **526** wird über die Taktgrenze hinweg an den Multiplexer **538** übergeben.

[0047] Wiederum empfängt das Zeitsteuer-Flip-Flop **530** das Signal src_out vom Zeitsteuer-Flip-Flop **520**. Bei Beginn eines neuen Taktzyklus innerhalb des Taktbereichs des Blocks **514** gibt das Zeitsteuer-Flip-Flop **530** den Wert des Signals src_out als das Signal dest_sync1 weiter. Dieses Signal wird seinerseits an das Zeitsteuer-Flip-Flop **531** übergeben, das ähnlich wie das Zeitsteuer-flip-Flop **530** arbeitet. Bei Beginn eines neuen Taktzyklus gibt das Zeitsteuer-Flip-Flop **531** den Wert des Signals dest_sync1 als das Signal dest_sync2 weiter. Das Signal dest_sync2 wird an den Invertierer **532** und das XNOR-Gatter **536** übergeben. Das invertierte Signal dest_sync2 wird an das Zeitsteuer-Flip-Flop **534** übergeben. Das Zeitsteuer-Flip-Flop **534** arbeitet ähnlich wie das Zeitsteuer-Flip-Flop **530**. Bei Beginn eines neuen Taktzyklus gibt das Zeitsteuer-Flip-Flop **534** den Wert des invertierten Signals dest_sync2 als das Signal dest_out weiter. Wenn sich also das invertierte Signal dest_sync1 ändert (Übergang), wird der Wert des Signals dest_out während einer Taktperiode auf einem Pegel sein, der äquivalent zu dest-sync2 ist. Da diese Signale beide an das XNOR-Gatter **536** geliefert werden, erzeugt dies während dieser Taktperiode ein Signal mit hohem Pegel.

[0048] Das Ausgangssignal des XNOR-Gatters **536** wird an den Steuereingang des Multiplexers **538** geliefert. Der Multiplexer **538** empfängt auch den Zeigerwert, der momentan von den Abtast-Flip-Flops **540** gehalten wird, sowie das Ausgangssignal der Halt-Flip-Flops **526**. Wenn der Multiplexer **538** von dem XOR-Gatter **536** ein Signal mit hohem Pegel empfängt, liefert er den Zeigerwert von den Halte-Flip-Flops **526**. Anderenfalls liefert er den Zeigerwert vom Ausgang der Abtast-Flip-Flops **540** an den Eingang der Abtast-Flip-Flops **540**. Der Zeigerwert vom Ausgang der Abtast-Flip-Flops **540** ist das Signal addr_ptr_sync **539**. Dieses Signal folgt somit dem Signal addr_ptr **523**. Das Signal addr_ptr_sync **539** ist jedoch im Taktbereich des Blocks **514** gültig.

[0049] Die Synchronisierschaltung **510** kann mit Schaltkreiselementen einer im Handel erhältlichen Standard-CMOS-Zellenbibliothek realisiert werden. Alle Gatter, Flip-Flops und Multiplexer sind allgemein erhältlich.

[0050] Mit Bezug auf Fig. 6 wird ein Zeitablaufdiagramm der Signale in Fig. 5 beschrieben. Das Zeitablaufdiagramm umfaßt das Signal clock_src (Ursprungstakt) und das Signal clock_dest (Zieltakt). Dies sind Taktsignale, die asynchron und mit verschiedenen Frequenzen arbeiten. Die Signale src_sync1, src_sync2, src_out, sel_src und addr held arbeiten auf der Basis des Signals clock_src. Die Signale dest_sync1, dest_sync2, dest_out, sel_dest und addr_prt_sync arbeiten auf der Basis des Signals clock_dest.

[0051] Das Signal reset_z beginnt bei einem niedrigen Zustand und geht dann auf einen hohen Zustand über. Dieser Übergang wird zum Auslösen des normalen Betriebs der Synchronisierschaltung verwendet. Nachdem das Signal reset_z während einer Taktperiode des Signals clock_src in dem hohen Zustand war, geht das Signal src_sync1 auf einen hohen Zustand. Dann, nach einer Taktperiode des Signals clock_src, geht das Signal src_sync2 auf einen hohen Zustand. Da das Signal src_sync2 nicht mit dem Signal src_out übereinstimmt, geht das Signal sel_src auf einen hohen Zustand. Nach einer Taktperiode des Signals clock_src bewirkt das Signal sel_src, daß addr_held das Signal des nächsten Adreßzeigers liest. Das erste Adreßzeigersignal ist mit A1 bezeichnet. Nachfolgende Adreßzeigersignale sind mit A2 und A3 bezeichnet.

[0052] Nachdem das Signal src_sync2 während einer Taktperiode auf einem hohen Zustand war, geht das Signal src_out auf einen hohen Zustand über. Da das Signal src_out nun mit dem Signal src_sync2 übereinstimmt, wird das Signal sel_src deaktiviert. Somit wird der Adreßzeiger A1 von dem Signal addr_held gehalten.

[0053] Der Übergang des Signals src_out wird an das Signal dest_sync1 weitergegeben. Insbesondere macht bei der nächsten steigenden Flanke des Signals cllock_dest das Signal dest_sync1 denselben Übergang. Nach einer Periode des Signals clock_dest folgt das Signal dest_sync2 dem Signal dest_sync1 und geht auf einen hohen Zustand über.

[0054] Da das Signal dest_sync2 mit dem Signal dest_out übereinstimmt, geht das Signal sel_dest auf einen hohen Zustand. Nach einer Taktperiode des Signals clock_dest bewirkt das Signal sel_dest, daß addr_prt_sync das nächste Adreßzeigersignal liest. Das erste Adreßzeigersignal wird als A1_sync identifiziert. Dieses Signal stimmt mit dem Signal addr_held A1 überein, ist jedoch im Taktbereich des Signals clock_dest gültig. Nachfolgende Adreßzeigersignale im Taktbereich des Signals clock_dest werden als A2_sync und A3_sync bezeichnet. Diese Signale folgen A2 bzw. A3.

[0055] Eine Taktperiode nach dem Übergang des Signals dest_sync2 macht das Signal dest_out den entgegengesetzten Übergang. Da das Signal dest_out nicht mehr mit dem Signal dest_sync2 übereinstimmt, wird das Signal sel_dest deaktiviert, und das Signal addr_ptr_sync hält den Adreßzeiger A1 sync.

[0056] Wie durch das Zeitablaufdiagramm dargestellt, bewirken nachfolgende Übergänge der Zeit-

steuersignale, daß die Adreßzeigersignale A2 und A3 über die Taktbereiche hinweg an die synchronisierten Adreßsignale A2_sync und A3_sync übergeben werden.

[0057] Die Übergänge des Signals src_out lösen das Lesen eines neuen Adreßzeigers aus. Diese Übergänge werden von dem Taktbereich clock_src an den Taktbereich clock_dest übergeben. Gestützt auf das Signal src_out macht das Signal dest_out einen verzögerten Übergang. Dieser verzögerte Übergang wird dazu verwendet, die Übertragung eines neuen Adreßzeigers über die Taktbereiche hinweg auszulösen. Dieser verzögerte Übergang wird auch über die Taktbereiche hinweg zurückgegeben, um einen weiteren Übergang des Signals src_sync1 auszulösen. Dadurch wird wiederum das Lesen eines neuen Adreßzeigers ausgelöst.

[0058] Anders gesagt, die Übergänge des Signals src_out und des Signals dest_out arbeiten wie Token. Dieser Token wird fortlaufend von dem Ursprungstaktbereich **512** an den Zieltaktbereich **514** und wieder zurück übergeben. Der Token wird durch die Aktivierung des Signals reset_z (Setzen auf logisch wahr) ausgelöst und beginnt als ein Übergang des Signals dest_out. Der Token wird dann als Übergänge der Signale src_sync1, src_sync2, src_out, dest_sync1 und dest_sync2 weitergegeben. Der Token wird an dest_out durch einen Übergang in dem Signal dest_sync2 zurückgegeben.

[0059] Ein Token src_out (d.h. ein Übergang des Signals src_out) zeigt an, daß ein neuer Adreßzeiger in die Halte-Flip-Flops geschrieben wurde. Wenn dieser Token über Taktbereiche hinweg weitergegeben wird, löst er die Synchronisierung des neuen Adreßzeigers aus.

[0060] Ähnlich zeigt der Token dest_out an, daß der neue Adreßzeiger über die Taktbereiche hinweg synchronisiert wurde und in den Abtast-Flip-Flops gehalten wird. Wenn dieser Token über die Taktbereich hinweg zurückgegeben wird, löst er das Lesen eines neuen Adreßzeigers aus. Auf diese Art werden Adreßzeiger über Taktbereich hinweg übergeben.

[0061] Die Synchronisierschaltung **510** (Fig. 5) kann weiter optimiert werden, indem weitere logische Gatter zum Steuern des Anfangs und der Beendigung dieser Tokenübertragung eingeführt werden. Für diesen Zweck kann z.B. ein zusätzlicher Eingang eines UND-Gatters **516** verwendet werden.

[0062] Obwohl die Ausführungsformen der Erfindung hier in bezug auf einen FIFO-Speicher beschrieben wurden, ist die Erfindung auf andere Schaltungen anwendbar, die die Synchronisierung von Daten über Taktbereiche hinweg erfordern. Obwohl die Ausführungsformen hier mit Bezug auf bestimmte Schaltungen und Blockdiagramme beschrieben wurden, eignen sich auch andere Konfigurationen für die Realisierung der beschriebenen Funktionen. Der Fachmann wird aus den hier offenbarten Ausführungsformen sicher verstehen, daß viele Modifikationen möglich sind, ohne die Lehre der Erfindung zu verlassen. All diese Modifikationen sollen im Bereich der folgenden Ansprüche liegen.

Patentansprüche

1. FIFO-Speicher mit folgenden Merkmalen:
ein Speicher (**110**), der einen FIFO-Stapel speichern kann;
ein Schreibkanal (**112**, **114**), der gestützt auf einen Schreibadreßzeiger dem FIFO Stapel Elemente hinzufügen kann, wobei der Schreibkanal (**112**, **114**) in einem ersten Taktbereich arbeitet;
ein Lesekanal (**116**, **118**), der gestützt auf einen Leseadreßzeiger Elemente von dem FIFO-Stapel lesen kann, wobei der Lesekanal (**116**, **118**) in einem zweiten Taktbereich arbeitet, der sich von dem ersten Taktbereich unterscheidet;
eine erste Synchronisierschaltung (**120**), die mit dem Schreibkanal (**112**, **114**) betrieblich gekoppelt und so konfiguriert ist, daß sie den Schreibadreßzeiger empfängt und den Schreibadreßzeiger zu dem zweiten Taktbereich synchronisiert; und
eine zweite Synchronisierschaltung (**124**), die mit dem Lesekanal (**116**, **118**) betrieblich gekoppelt und so konfiguriert ist, daß sie den Leseadreßzeiger empfängt und den Leseadreßzeiger zu dem ersten Taktbereich synchronisiert, wobei
die erste und die zweite Synchronisierschaltung (**120**, **124**) jeweils ein erstes Zeitsteuer-Flip-Flop (**420**; **520**) und ein zweites Zeitsteuer-Flip-Flop (**434**; **534**) umfassen, die miteinander gekoppelt sind, die ersten Zeitsteuer-Flip-Flops (**420**; **520**) erste Zeitsteuersignale (src_out) erzeugen und die zweiten Zeitsteuer-Flip-Flops (**434**; **534**) zweite Zeitsteuersignale (dest_out) erzeugen, und der Taktbereich der ersten Zeitsteuer-Flip-Flops sich jeweils von dem Taktbereich der zweiten Zeitsteuer-Flip-Flops unterscheidet, wobei die ersten und die zweiten Zeitsteuer-Flip-Flops alternativ in dem ersten oder in dem zweiten Taktbereich arbeiten.

2. FIFO-Speicher nach Anspruch 1, bei dem der Speicher (**110**) ein DPRAM umfaßt.

3. FIFO-Speicher nach Anspruch 1 oder 2, bei dem der Schreibkanal folgende Merkmale aufweist:
einen Adreßport (**114**), der so konfiguriert ist, daß er eine Schreibadreßstelle in dem Speicher (**110**) auswählt; und
einen Datenport (**112**), der so konfiguriert ist, daß er Daten an die Schreibadreßstelle liefert.

4. FIFO-Speicher nach einem der vorangehenden Ansprüche, bei dem der Lesekanal folgende

Merkmale aufweist:
einen Adreßport (**118**), der so konfiguriert ist, daß er eine Leseadreßstelle in dem Speicher auswählt; und
einen Datenport (**116**), der so konfiguriert ist, daß er Daten von der Leseadreßstelle empfängt.

5. FIFO-Speicher nach einem der vorangehenden Ansprüche, bei dem die erste und die zweite Synchronisierschaltung (**120**, **124**) jeweils folgende Merkmale aufweisen:
eine Halteschaltung (**426**; **526**), die so konfiguriert ist, daß sie einen Adreßzeiger empfängt, wobei die Haltchaltung in dem ersten Taktbereich arbeitet; und
eine Abtastschaltung (**440**; **540**), die mit der Halteschaltung (**426**; **526**) betrieblich gekoppelt und so konfiguriert ist, daß sie den Adreßzeiger empfängt, wobei die Abtastschaltung (**440**; **540**) in dem zweiten Taktbereich arbeitet.

6. FIFO-Speicher nach Anspruch 5, bei dem die erste und die zweite Synchronisierschaltung (**120**, **124**) folgende weitere Merkmale aufweisen:
ein XOR-Gatter (**422**, **522**), das mit dem ersten Zeitsteuer-Flip-Flop (**418**, **420**; **518**, **520**) betrieblich gekoppelt und so konfiguriert ist, daß es ein Steuersignal erzeugt, wenn das Zeitsteuersignal einen Übergang macht; und
einen Multiplexer (**424**; **524**), der mit der Haltesqchaltung (**426**; **526**) und dem XOR-Gatter (**422**; **522**) betrieblich gekoppelt ist, einen Adreßzeigereingang aufweist und so konfiguriert ist, daß er den Adreßzeigereingang auswählt, wenn das XOR-Gatter (**422**; **522**) das Steuersignal erzeugt.

7. FIFO-Speicher nach einem der vorangehenden Ansprüche, mit folgenden weiteren Merkmalen:
eine Leselogikschaltung (**126**), die mit der ersten Synchronisierschaltung (**120**) und dem Lesekanal (**116**, **118**) betrieblich gekoppelt ist, wobei die Leselogikschaltung (**126**) ein FIFO-Leer-Signal erzeugen kann; und
eine Schreiblogikschaltung (**122**), die mit dem Schreibkanal (**112**, **114**) und dem Lesekanal (**116**, **118**) betrieblich gekoppelt ist, wobei die Schreiblogikschaltung (**122**) ein FIFO-Voll-Signal erzeugen kann.

8. Synchronisierschaltung zum Koordinieren von Adreßzeigern über Taktbereiche hinweg, mit folgenden Merkmalen:
ein erstes Zeitsteuer-Flip-Flop (**420**; **520**), das so konfiguriert ist, daß es erste Zeitsteuersignale erzeugt, wobei das erste Zeitsteuer-Flip-Flop (**420**; **520**) in einem ersten Taktbereich arbeitet;
eine Inversionsschaltung (**432**; **532**), die mit dem ersten Zeitsteuer-Flip-Flop (**420**; **520**) betrieblich gekoppelt und so konfiguriert ist, daß es gestützt auf die ersten Zeitsteuersignale invertierte Zeitsteuersignale erzeugt; und
ein zweites Zeitsteuer-Flip-Flop (**434**; **534**), das mit der Inversionsschaltung (**432**; **532**) betrieblich gekoppelt und so konfiguriert ist, daß es gestützt auf die invertierten Zeitsteuersignale zweite Zeitsteuersignale erzeugt, wobei das zweite Zeitsteuer-Flip-Flop (**434**; **534**) in einem zweiten Taktbereich arbeitet, der sich von dem ersten Taktbereich unterscheidet.

9. Synchronisierschaltung nach Anspruch 8, mit folgenden weiteren Merkmalen:
ein erstes Gatter (**422**; **522**), das mit dem ersten Zeitsteuer-Flip-Flop (**420**; **520**) betrieblich gekoppelt und so konfiguriert ist, daß es ein erstes Steuersignal erzeugt, wenn die ersten Zeitsteuersignale einen Übergang machen;
ein erster Multiplexer (**424**; **524**), mit dem ersten Gatter (**422**; **522**) betrieblich gekoppelt ist und mehrere Eingänge sowie einen Ausgang für Adreßzeiger aufweist, wobei der erste Multiplexer (**424**; **524**) gestützt auf das erste Steuersignal einen der mehreren Eingänge auswählt; und
ein Halte-Flip-Flop (**426**; **526**), das mit dem Ausgang des ersten Multiplexers (**424**; **524**) betrieblich gekoppelt und so konfiguriert ist, daß es die Adreßzeiger hält.

10. Synchronisierschaltung nach Anspruch 9, mit folgenden weiteren Merkmalen:
ein zweites Gatter (**436**; **536**), das mit dem zweiten Zeitsteuer-Flip-Flop (**430**; **530**) betrieblich gekoppelt und so konfiguriert ist, daß es ein zweites Steuersignal erzeugt, wenn die zweiten Zeitsteuersignale einen Übergang machen;
ein zweiter Multiplexer (**438**; **538**), der mit dem zweiten Gatter (**436**; **536**) betrieblich gekoppelt ist und mehrere Eingänge sowie einen Ausgang für die Adreßzeiger aufweist, wobei der zweite Multiplexer (**438**; **538**) gestützt auf das zweite Steuersignal einen der mehreren Eingänge auswählt; und
ein Abtast-Flip-Flop (**440**; **540**), das mit dem Ausgang des zweiten Multiplexers (**438**; **538**) betrieblich gekoppelt und so konfiguriert ist, daß es die Adreßzeiger hält.

11. Verfahren zum Synchronisieren von Eingangs-Speicheradressen (addr_ptr) und synchronisierten Speicheradressen (addr_ptr_sync) in einem Zweikanal-FIFO-Speicher, wobei die Eingangs-Speicheradressen (addr_ptr) und synchronisierten Speicheradressen (addr_ptr_sync) in verschiedenen Taktbereichen vorgesehen sind, die ein Eingangs-Taktsignal (clock_src) und ein dazu asynchrones Ziel-Taktsignal (clock_dest) aufweisen, das die folgenden Schritte umfaßt:
Senden eines Eingangs-Synchronisierungssignals (src_sync1) mit dem Beginn eines Taktzyklus des ersten Taktsignals (clock_src);
Empfangen des Eingangs-Synchronisierungssignals (src_sync1) und Senden eines ersten Synchronisierungs-Ausgangssignals (src_out) mit dem Beginn eines neuen Taktzyklus des ersten Taktsignals (clock_src);

Empfangen des ersten Synchronisierungs-Ausgangssignals (src_out) und Senden eines Ziel-Synchronisierungssignals (dest_sync1) mit dem Beginn eines neuen Taktzyklus des Ziel-Taktsignals (clock_dest);
Invertieren des Ziel-Synchronisierungssignals (dest_sync1);
Empfangen des invertierten Ziel-Synchronisierungssignals und Senden des invertierten Ziel-Synchronisierungssignals als ein zweites Synchronisierungs-Ausgangssignals (dest_out) mit dem Beginn einer neuen Taktperiode des Ziel-Taktsignals (clock_dest), wobei das Eingangs-Synchronisierungssignal (src_sync1) von dem zweiten Synchronisierungs-Ausgangssignal (dest_out) abhängt;
Auswählen der Eingangs-Speicheradressen (addr_ptr) zwischen der aktuellen ausgewählten Eingangs-Speicheradresse und einer folgenden ausgewählten Eingangs-Speicheradresse abhängig von dem Eingangs-Synchronisierungssignal (src_sync1) und dem ersten Synchronisierungs-Ausgangssignal (src_out); und
Auswählen der synchronisierten Speicheradressen (addr_ptr_sync) zwischen der aktuellen ausgewählten synchronisierten Speicheradresse und einer folgenden ausgewählten aktuellen synchronisierten Speicheradresse abhängig von dem Ziel-Synchronisierungssignal (dest_sync1) und dem zweiten Synchronisierungs-Ausgangssignal (dest_out).

12. Verfahren nach Anspruch 11, wobei
das Auswählen der Eingangs-Speicheradressen (addr_ptr) von der XOR-Verknüpfung des Eingangs-Synchronisierungssignals (src_sync1) mit dem ersten Synchronisierungs-Ausgangssignal (src_out) abhängt und
das Auswählen der synchronisierten Speicheradressen (addr_ptr_sync) von der XOR-Verknüpfung des Ziel-Synchronisierungssignals (dest_sync1) mit dem zweiten Synchronisierungs-Ausgangssignal (dest_out) abhängt.

13. Verfahren nach Anspruch 11 oder 12, wobei das Eingangs-Synchronisierungssignal (src-sync1) ferner mit dem Pegel eines Startsignals (reset_z) UND-verknüpft ist.

14. Verfahren nach Anspruch 11, 12 oder 13, wobei nach dem Auswählen der Eingangs-Speicheradresse diese in Halte-Flip-Flops gehalten wird und nach dem Auswählen der synchronisierten Speicheradresse diese in Abtast-Flip-Flops gehalten wird.

15. Signalsatz zur Steuerung eines Zweikanal-FIFO – Speichers mit den folgenden Signalen:
ein erstes Taktsignal (clock_src) mit einem ersten Frequenzbereich;
ein zweites Taktsignal (clock_dest) mit einer zweiten Frequenz, wobei das zweite Taktsignal (clock-dest) zu dem ersten Taktsignal (clock_src) asynchron ist;
einem ersten Adressignal (addr_ptr), das einen ersten Speicherort in dem Speicher kennzeichnet, wobei das erste Adressignal (addr_ptr) von dem ersten Taktsignal (clock_src) getaktet ist;
einem ersten Synchronisierungssignal (sel_src), das angibt, wann das erste Adressignal (addr_ptr) einen neuen Speicherort kennzeichnet, wobei das erste Synchronisierungssignal (sel_src) von dem ersten Taktsignal (clock_src) getaktet ist;
ein zweites Synchronisierungssignal (sel_dest), das kennzeichnet, wann ein zweites Adressignal (addr_ptr_sync) einen neuen Speicherort kennzeichnet, wobei das zweite Synchronisierungssignal (sel_dest) von dem zweiten Taktsignal (clock_dest) getaktet ist;
und das zweite Adressignal (addr_ptr_sync) einen Speicherort in dem Speicher kennzeichnet, wobei das zweite Adressignal (addr_ptr_sync) von dem zweiten Taktsignal (clock_dest) getaktet ist, und das zweite Adressignal (addr_ptr_sync) den neuen Speicherort kennzeichnet, nachdem das zweite Synchronisierungssignal (sel_dest) angibt, daß das erste Adressignal (addr_ptr) den neuen Speicherort kennzeichnet.

16. Signalsatz nach Anspruch 15, mit
einem ersten Ausgangs-Synchronisierungssignal (src_out), das von dem ersten Taktsignal (clock_src) getaktet ist, und
einem zweiten Ausgangs-Synchronisierungssignal (dest_out), das von dem zweiten Taktsignal (clock_dest) getaktet ist, wobei
das erste Synchronisierungssignal (sel_src) durch eine XOR-Verknüpfung des ersten Ausgangs-Synchronisierungssignals (src_out) mit einer verzögerten Version des zweiten Ausgangs-Synchronisierungssignals (dest_out) vorgesehen ist, wobei die verzögerte Version des zweiten Ausgangs-Synchronisierungssignals (dest_out) verzögert ist, um mit einem folgendem Taktzyklus des erste Taktsignals (clock_src) zu beginnen,
das zweite Synchronisierungssignal (sel_dest) durch eine XOR-Verknüpfung des zweiten Ausgangs-Synchronisierungssignals (dest_out) verzögerten Version des ersten Ausgangs-Synchronisierungssignals (src_out) vorgesehen ist, wobei die verzögerte Version des ersten Ausgangs-Synchronisierungssignals (src_out) verzögert ist, um mit einem folgenden Taktzyklus des zweiten Taktsignals (clock_dest) zu beginnen,
das erste Ausgangs-Synchronisierungssignal (src_out) zu der verzögerten Version des zweiten Ausgangs-Synchronisierungssignals (dest_out) verzögert ist, um mit einem folgendem Taktzyklus des erste Taktsignals (clock_src) zu beginnen und
das zweite Ausgangs-Synchronisierungssignal (dest_out) zu der verzögerten Version des ersten Ausgangs-Synchronisierungssignals (src_out) invertiert und verzögert ist, um mit einem folgenden Taktzyklus des zweiten Taktsignals (clock_src) zu begin-

nen.

Es folgen 4 Blatt Zeichnungen

Anhängende Zeichnungen

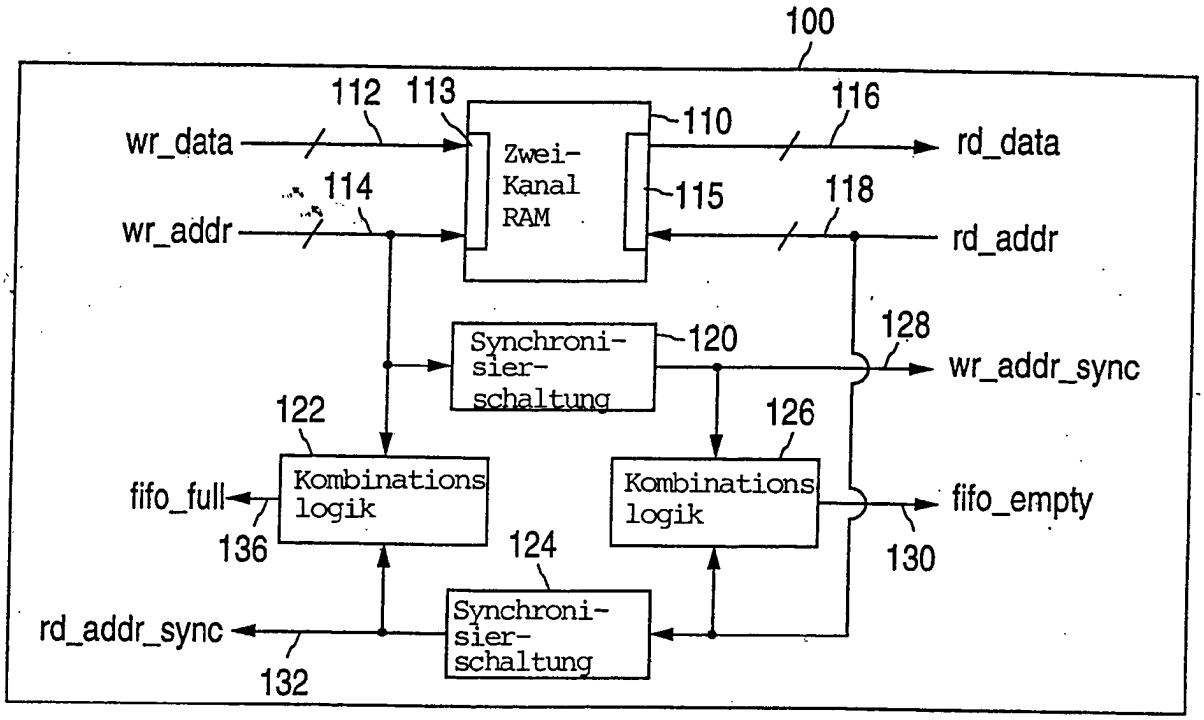


FIG. 1

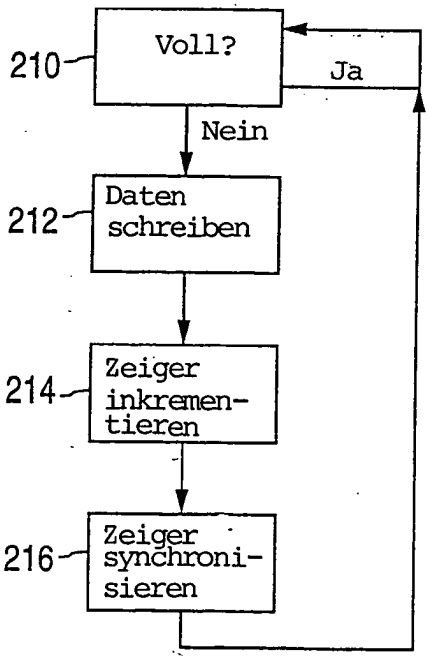


FIG. 2

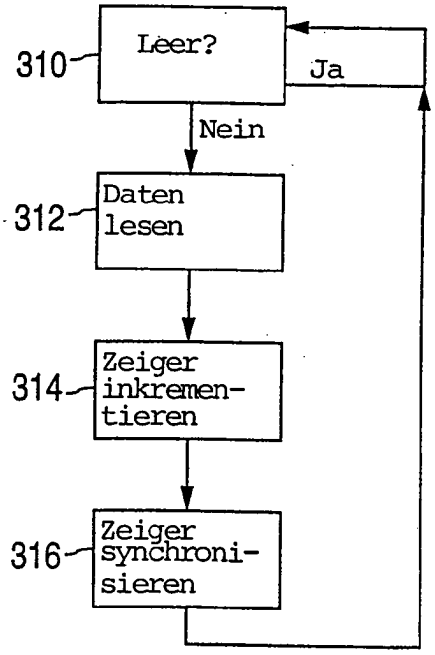


FIG. 3

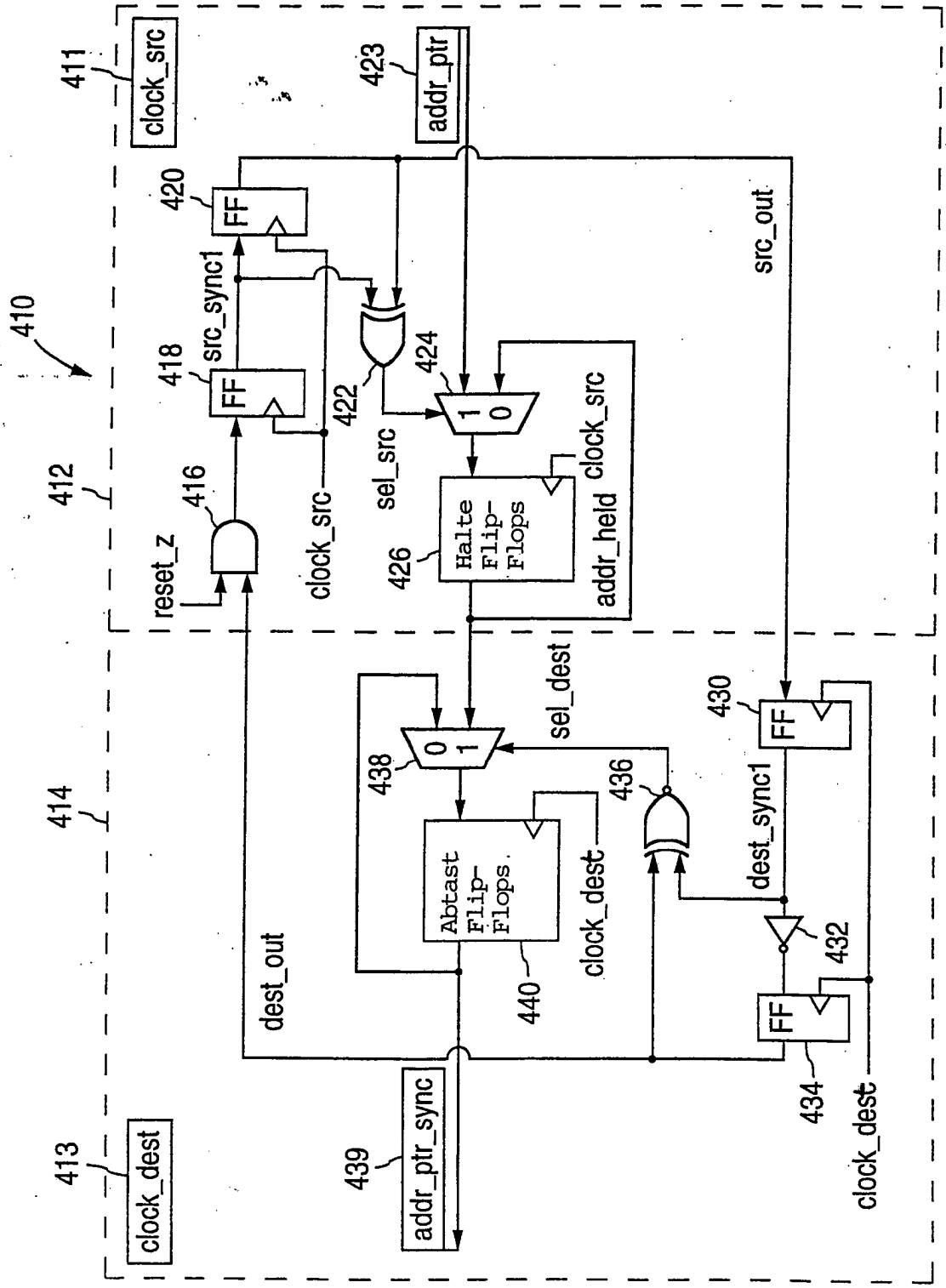

410
411
clock_src
412
413
clock_dest
414
420
FF
418
FF
src_sync1
422
424
1
0
sel_src
423
addr_ptr
src_out
416
reset_z
clock_src
426
Halte Flip-Flops
addr_held
clock_src
sel_dest
438
0
1
Abtast Flip-Flops
440
clock_dest
436
dest_out
439
addr_ptr_sync
430
FF
dest_sync1
432
FF
434
clock_dest

FIG. 4

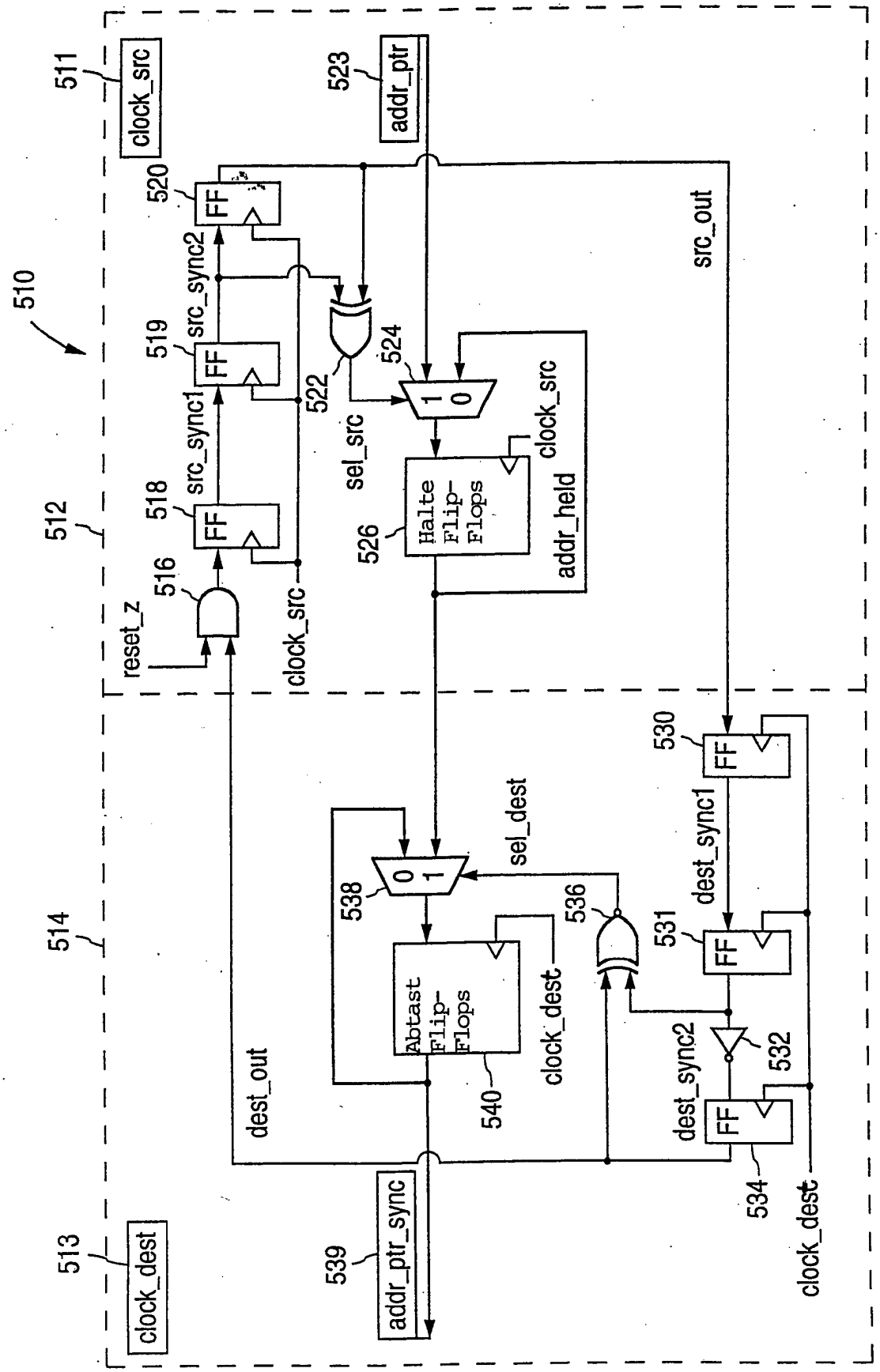

510
511
clock_src
512
513
clock_dest
514
520
FF
519
FF
518
FF
516
reset_z
clock_src
src_sync2
src_sync1
522
sel_src
523
addr_ptr
524
1
0
Halte Flip-Flops
526
addr_held
clock_src
src_out
530
FF
dest_sync1
531
FF
dest_sync2
532
FF
534
clock_dest
536
sel_dest
538
0
1
Abtast Flip-Flops
540
clock_dest
dest_out
539
addr_ptr_sync

FIG. 5

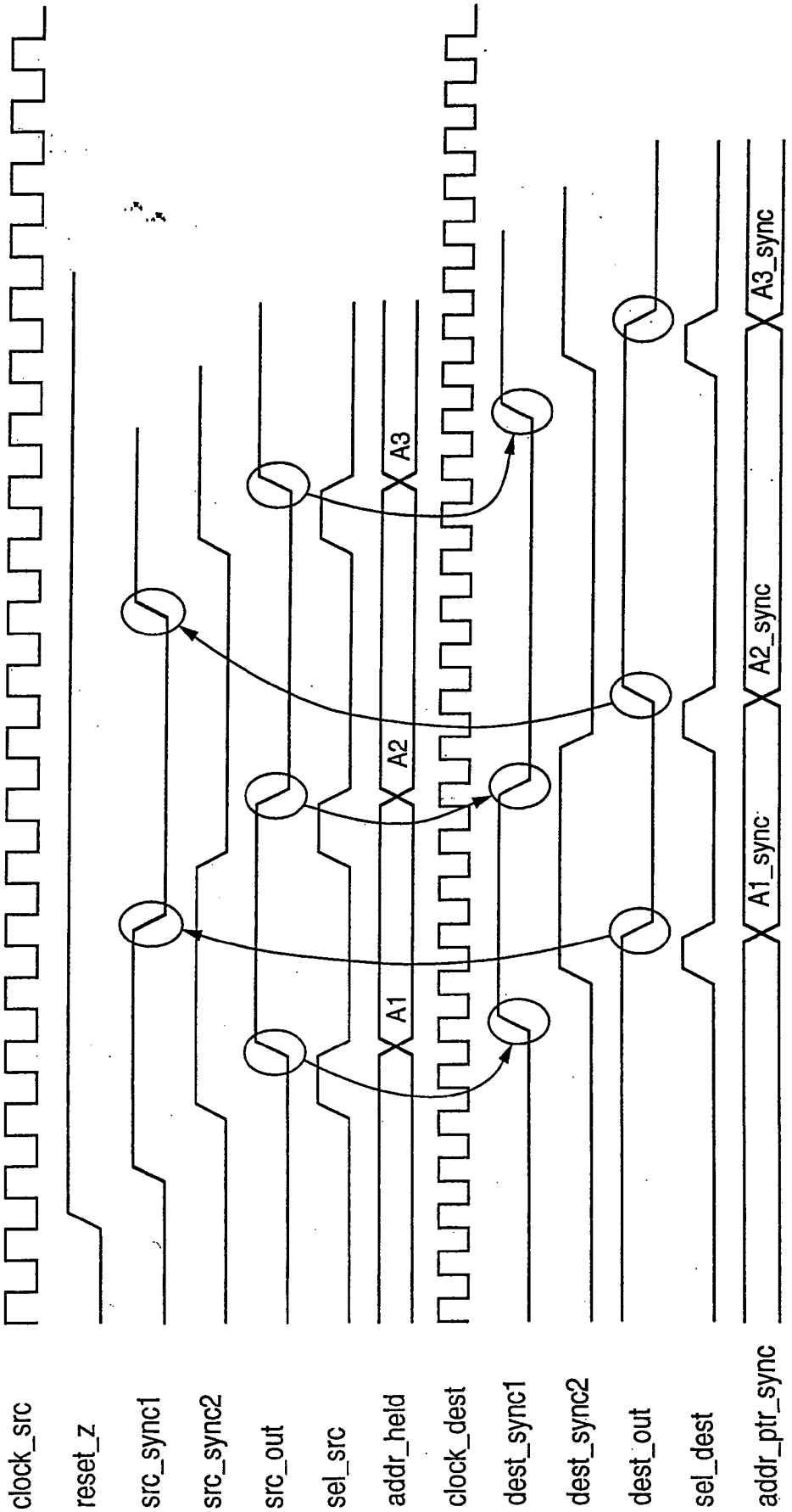

clock_src
reset_z
src_sync1
src_sync2
src_out
sel_src
addr_held
clock_dest
dest_sync1
dest_sync2
dest_out
sel_dest
addr_ptr_sync
A1
A2
A3
A1_sync
A2_sync
A3_sync

FIG. 6