



19 **BUNDESREPUBLIK
DEUTSCHLAND**



**DEUTSCHES
PATENT- UND
MARKENAMT**

12 **Patentschrift**
10 **DE 100 49 629 C 1**

51 Int. Cl. 7:
H 03 K 19/173

21 Aktenzeichen: 100 49 629.6-31
22 Anmeldetag: 5. 10. 2000
43 Offenlegungstag: -
45 Veröffentlichungstag
der Patenterteilung: 1. 8. 2002

DE 100 49 629 C 1

Innerhalb von 3 Monaten nach Veröffentlichung der Erteilung kann Einspruch erhoben werden

73 **Patentinhaber:**
Sankowski, Hagen, 98693 Manebach, DE

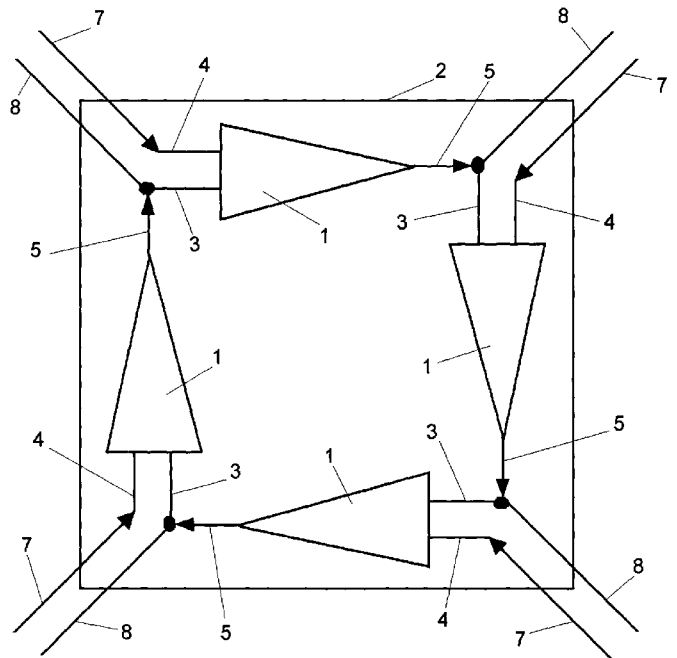
74 **Vertreter:**
Engel und Kollegen, 98527 Suhl

72 **Erfinder:**
gleich Patentinhaber

56 **Für die Beurteilung der Patentfähigkeit in Betracht
gezogene Druckschriften:**
DE 42 24 804 C1
DE 36 30 835 A1
DE 37 86 669 T4

54 **Programmierbare Schaltungsanordnung**

57 Die Erfindung betrifft eine programmierbare Schaltungsanordnung mit mehreren Logikelementen (1), die kreisförmig zusammenschaltet sind und gemeinsam einen Logikblock (2) bilden. Die Logikelemente (1) besitzen wenigstens einen ersten Dateneingang (3) und einen zweiten Dateneingang (4). Außerdem besitzen die Logikelemente (1) mindestens einen Datenausgang (5). Die Logikelemente (1) sind mit einheitlicher Signalflußrichtung vom Dateneingang zum Datenausgang kreisförmig zusammenschaltet, so daß jeder Datenausgang (5) eines ersten Logikelements (1) jeweils an einen Dateneingang (3) des nachfolgenden Logikelements (1) gekoppelt ist. Der jeweils andere Dateneingang jedes Logikelements (1) steht als Dateneingang (7) des Logikblocks (2) zur Verfügung. Der Datenausgang (5) jedes Logikelements (1) ist gleichzeitig nach außen geführt und bildet einen Datenausgang (8) des Logikblocks (2).
Mit solch einer Schaltungsanordnung kann ein FPGA effektiv aufgebaut werden, wobei ein großer Teil der herkömmlichen Verbindungskapazitäten eingespart werden kann.



DE 100 49 629 C 1

[0001] Die vorliegende Erfindung betrifft eine programmierbare Schaltungsanordnung mit mehreren Logikelementen, die wenigstens zwei Dateneingänge und einen Datenausgang haben.

[0002] Solche programmierbaren Schaltungsanordnungen sind geeignet, verschiedene komplexe digitale, kombinatorische Funktionen zu realisieren, wobei die gewünschte digitale Funktion durch eine veränderbare Programmierung der Schaltungsanordnung vom Anwender vorgegeben wird. Programmierbare Schaltungsanordnungen bilden die Grundlage für anwenderprogrammierbare Logikmatrizen (Field Programmable Gate Array = FPGA). Bei neueren Schaltkreisentwürfen werden programmierbare Schaltungsanordnungen auch als Bestandteil von speziellen anwendungsbezogenen integrierten Schaltkreisen (Application Specific Integrated Circuits = ASIC) genutzt. Generell bieten solche programmierbaren Schaltungsanordnungen den Vorteil, daß mit ein und demselben Bauelement (üblicherweise als integrierter Schaltkreis aufgebaut) je nach Bedarf verschiedene Logikfunktionen realisiert werden können. Auf diese Weise können beispielsweise Schaltungen, die auf gedruckten Leiterplatten angeordnet sind, an veränderte Einsatzbedingungen angepaßt werden.

[0003] Die Grundlagen von FPGA-Schaltungen sowie die bevorzugten Realisierungen und Anwendungsfälle sind zusammenfassend in dem Buch "Das FPGA-Kochbuch", M. Wannemacher, International Thompson Publishing, 1998, beschrieben. Dort sind auch die üblichen Schaltungsanordnungen erläutert, die bisher zur Realisierung von FPGA verwendet werden. Die derzeit üblichen Schaltungsanordnungen in FPGA sind aus zwei Hauptkomponenten zusammengesetzt, einerseits den Logikblöcken, die eine vielseitig konfigurierbare Kombinatorik umfassen, und andererseits den sogenannten Verbindungskapazitäten, die der Realisierung der jeweils notwendigen Verbindungen zwischen den Eingängen und den Ausgängen der Logikblöcke dienen. Die Verbindungskapazitäten werden üblicherweise durch eine Vielzahl von Verbindungsleitungen sowie durch Multiplexer und Demultiplexer gebildet, die letztlich die verschiedenen Verschaltungsvarianten der Verbindungsleitungen bereitstellen, in Abhängigkeit von der angewendeten Programmierung.

[0004] Um so mehr Logikblöcke in einem integrierten Schaltkreis enthalten sind und um so vielseitiger diese Logikblöcke konfigurierbar sein sollen, desto aufwendiger wird die Realisierung der Verbindungskapazitäten. Die Verbindungsleitungen, die Multiplexer und die Demultiplexer müssen es ermöglichen, alle gewünschten kombinatorischen Schaltfunktionen mit den zur Verfügung stehenden Logikblöcken zu realisieren. Die Verbindungskapazitäten sind daher eine wesentliche Größe hinsichtlich der Gesamtkomplexität des integrierten Schaltkreises, des Fertigungsaufwandes, der Fehleranfälligkeit und damit der Herstellungskosten. Obwohl die Verbindungskapazitäten selbst nicht zur Realisierung der eigentlichen Logikfunktion genutzt werden, beanspruchen sie einen erheblichen Teil der zur Verfügung stehenden Ressourcen eines FPGA. Je vielseitiger ein FPGA ausgelegt ist, um so höher ist auch der Anteil der jeweiligen Verbindungskapazitäten, der bei einer realisierten Logikfunktion nicht verwendet wird, also funktionslos oder redundant in dem jeweiligen Schaltkreis verbleibt.

[0005] Eine Aufgabe der vorliegenden Erfindung besteht somit darin, eine programmierbare Schaltungsanordnung bereitzustellen, bei welcher der Bedarf an Verbindungskapazitäten drastisch gesenkt wird, um dadurch die zur Verfügung stehenden Ressourcen in einem mikroelektronischen

Chip besser zu nutzen. Insbesondere ist es die Aufgabe der Erfindung, die Anzahl der Verbindungsleitungen zwischen den einzelnen Logikblöcken zu reduzieren und auf Multiplexer und Demultiplexer weitgehend zu verzichten. Dadurch soll es ermöglicht werden, mit vorhandenen Fertigungstechnologien höhere Packungsdichten an konfigurierbaren Schaltungen zu erreichen.

[0006] Diese und weitere Aufgaben werden durch eine programmierbare Schaltungsanordnung nach Anspruch 1 gelöst.

[0007] Auf diese Weise ist es möglich, Daten bzw. Signale in dem Logikblock in Abhängigkeit von der Programmierung der unmittelbar nachfolgenden Logikelemente, d. h. der diesen Logikelementen zugeordneten logischen Funktion, an in weiteren Logikebenen nachfolgende Logikelemente weiterzuleiten und/oder als Datenausgang des Logikblocks bereitzustellen. Dazu sind keine zusätzlichen Verbindungskapazitäten erforderlich. Vom Datenausgang eines Logikelements werden sowohl ein Dateneingang des unmittelbar nachfolgenden Logikelements als auch ein Ausgang des Logikblocks stets beliefert. Entsprechend der Programmierung der nachfolgenden Logikelemente ist bestimmt, wie die anliegenden Daten weiter verarbeitet, ggf. unverändert durchgeleitet oder ganz ignoriert werden. Damit bestimmt die Programmierung der Logikelemente auch gleichzeitig das sogenannte Routing für die Datensignale.

[0008] Eine besondere Ausführungsform dieser Schaltungsanordnung zeichnet sich dadurch aus, daß jeder Logikblock aus vier Logikelementen besteht, von denen jedes zwei Dateneingänge und einen Datenausgang besitzt, wobei alle Datenausgänge der Logikelemente auch als Datenausgänge des Logikblocks bereitgestellt sind. Diese Kombination der Logikelemente hat sich als besonders geeignet erwiesen, um möglichst vielseitig programmierbare Schaltungsanordnungen aufzubauen, ohne den schaltungs- und herstellungstechnischen Aufwand unnötig zu erhöhen.

[0009] Bei abgewandelten Ausführungsformen können auch Logikelemente mit abweichender Anzahl an Dateneingängen bzw. Datenausgängen eingesetzt werden. Es ist auch denkbar, in den Logikblöcken einzelne Logikelemente zu verwenden, deren Eingänge und/oder Ausgänge nicht unmittelbar als Dateneingänge bzw. Datenausgänge des Logikblocks zur Verfügung stehen, sondern nur über andere Logikelemente des gleichen Logikblocks erreichbar sind.

[0010] Vorzugsweise können die Logikelemente als logische Verweistabellen (look-up-table), herkömmliche Logikgatter oder Flip-Flop ausgestaltet sein. Dem Fachmann ist jedoch auch bekannt, daß andere schaltungstechnische Einheiten verwendet werden können, um derartige Logikelemente zu realisieren. Jedoch hat sich bisher gezeigt, daß insbesondere die logischen Verweistabellen einfach herstellbar sind und trotz leichter Programmierung viele verschiedene Logikfunktionen realisieren können.

[0011] Eine besonders zu bevorzugende Ausführungsform der Schaltungsanordnung zeichnet sich dadurch aus, daß mehrere solcher Logikblöcke zu einem Logiknetz zusammengeschaltet sind. Vorzugsweise wird dabei die gleiche Anordnung der Logikblöcke gewählt, wie sie auch für die Anordnung der Logikelemente innerhalb der einzelnen Logikblöcke zur Anwendung kommt.

[0012] Eine besonders zweckmäßige Ausführungsform dieses Logiknetzes ist aus einzelnen Netzmaschen zusammengesetzt, die jeweils vier kreisförmig zusammengeschaltete Logikblöcke enthalten, wobei an den Verknüpfungspunkten der Logikblöcke jeweils ein Dateneingang und ein Datenausgang der benachbarten Logikblöcke komplementär miteinander verbunden sind. Die Bestimmung der Signalwege in der gesamten Schaltungsanordnung erfolgt bei einer

derartigen Gestaltung ohne Zuhilfenahme von Multiplexern und Demultiplexern, allein durch die geeignete Konfiguration und Auswahl der Logikelemente in dem Logiknetz. Dabei werden die Signalwege (Routing) dadurch bestimmt, welches unmittelbar mit dem Datenausgang verbundene Logikelement mit welcher speziellen Konfiguration eingestellt wird. Wenn auf einen Datenausgang eines Logikelements ein ganz bestimmtes Logikelement folgen muß (mit einer bestimmten kombinatorischen Funktion), kann dieses nachfolgende Logikelement innerhalb des Logikblockes oder innerhalb einer Netzmasche realisiert werden.

[0013] In Abweichung zu den aus dem Stand der Technik bekannten Lösungen, wo die Signalwege durch Eingangsmultiplexer und Ausgangsdemultiplexer bestimmt werden, kann bei Verwendung der erfindungsgemäßen Schaltungsanordnung der Signalweg durch eine geeignete Verteilung der benötigten kombinatorischen Funktionen über die zur Verfügung stehenden Logikelemente erreicht werden.

[0014] Bei einer nochmals weitergebildeten Ausführungsform der Schaltungsanordnung ist die Struktur des Logiknetzes zumindest teilweise räumlich ausgebildet, indem ausgehend von einer flächigen Netzstruktur zumindest teilweise die Dateneingänge an einem Netzrand mit den Datenausgängen am gegenüberliegenden Netzrand verbunden sind. Durch diese Verschaltung der einzelnen Logikblöcke bzw. Netzmaschen wird eine weitgehend räumlich geschlossene Netzstruktur erzielt. Beispielsweise können zylinderförmige, kissenförmige oder kugelförmige Netzstrukturen aufgebaut werden, bei denen besonders hohe Variationsmöglichkeiten der Signalführung bestehen.

[0015] Weitere Vorteile, Einzelheiten und Weiterbildungen ergeben sich aus der nachfolgenden Beschreibung bevorzugter Ausführungsformen der Erfindung, unter Bezugnahme auf die Zeichnung. Es zeigen:

[0016] Fig. 1 ein Blockschaltbild einer ersten Ausführungsform einer programmierbaren Schaltungsanordnung mit vier Logikelementen, gemäß der Erfindung;

[0017] Fig. 2 ein Blockschaltbild einer zweiten Ausführungsform der Schaltungsanordnung mit drei Logikelementen;

[0018] Fig. 3 ein Funktionsschaltbild einer möglichen Realisierungsform eines Logikelements, welches in der erfindungsgemäßen Schaltungsanordnung verwendbar ist;

[0019] Fig. 4 eine weitere Ausführungsform der Schaltungsanordnung, wobei vier Logikblöcke zu einer Netzmasche verbunden sind;

[0020] Fig. 5 ein Blockschaltbild eines Abschnittes eines Logiknetzes mit einer Vielzahl von Logikblöcken;

[0021] Fig. 6 eine Detaildarstellung eines Randbereichs eines Logiknetzes.

[0022] Fig. 1 zeigt ein Blockschaltbild einer ersten Ausführungsform einer erfindungsgemäßen programmierbaren Schaltungsanordnung. Wesentlich für diese Schaltungsanordnung ist, daß mehrere Logikelemente **1** kreisförmig zusammengeschaltet sind und gemeinsam einen Logikblock **2** bilden. Bei der einfachsten Ausführungsform besitzen die Logikelemente **1** jeweils einen ersten Dateneingang **3** und einen zweiten Dateneingang **4**. Außerdem besitzen die Logikelemente **1** mindestens einen Datenausgang **5**. Die Logikelemente **1** sind mit einheitlicher Signalflußrichtung (betrachtet vom Dateneingang zum Datenausgang) kreisförmig zusammengeschaltet, d. h. jeder Datenausgang **5** eines ersten Logikelements **1** ist jeweils an einen Dateneingang (hier **3**) des nachfolgenden Logikelements **1** gekoppelt. Der jeweils andere Dateneingang jedes Logikelements **1** steht als Dateneingang **7** des Logikblocks **2** zur Verfügung. Der Datenausgang **5** jedes Logikelements **1** ist gleichzeitig nach außen geführt und bildet einen Datenausgang **8** des Logik-

blocks **2**.

[0023] Bei abgewandelten Ausführungsformen können die Logikelemente **1** mehrere Dateneingänge und/oder Datenausgänge besitzen.

[0024] Ebenfalls ist es möglich, andere Logikelemente in dem Logikblock vorzusehen, die beispielsweise keinen direkten Datenausgang des Logikblocks bereitstellen oder nur über andere Logikelemente innerhalb desselben Logikblocks Daten empfangen können. Außerdem kann die Anzahl der Logikelemente, die zu einem Logikblock zusammengefaßt sind, variieren.

[0025] Durch die ringförmige Anordnung der Logikelemente sind verschiedene Signalführungen möglich, die abhängig von der jeweiligen Konfiguration der Logikelemente sind. Dabei ist zu bedenken, daß gemäß dem Grundprinzip der programmierbaren Schaltungsanordnung die Logikelemente mit verschiedenen Funktionen programmiert werden können. Die ringförmige Anordnung eignet sich auch für ggf. benötigte Rückkopplungen innerhalb eines Datenflusses, da die einzelnen Logikelemente sequentiell angeordnet sind. Die programmierte Funktion jedes Logikelements bestimmt somit die weitere Verarbeitung und Verteilung des Dateneingangs. Durch die Definition der Signalwege (Routing) über die Konfiguration der verwendeten Logikelemente ist es möglich, die bei FPGA nach dem Stand der Technik üblichen Routingelemente (z. B. Multiplexer, Demultiplexer, Schaltmatrizen) einzusparen. Dies führt insbesondere zu einer effizienteren Nutzung der zur Verfügung stehenden Chipfläche, so daß letztlich größere Schaltungsanordnungen auf der Chipfläche realisiert werden können. Da auch die sonst übliche Vielzahl von verschiedenen Verbindungsleitungen drastisch reduziert wird, kann auf zusätzliche Metallisierungsebenen bei der Chipherstellung verzichtet werden, wodurch die Herstellungskosten und die Fehleranfälligkeit bei der Realisierung der Schaltungen sinken.

[0026] Beispielsweise könnte bei einer zu programmierenden Logikfunktion der Schaltungsanordnung der Bedarf bestehen, an einer bestimmten Stelle in der sequentiellen Datenverarbeitung einen Inverter vorzusehen, der den ihm eingespeisten Dateneingang invertiert. In Abhängigkeit von dem weiteren gewünschten Signalweg kann der Inverter innerhalb des im Logikblock ausgebildeten Ringes angeordnet sein oder außerhalb dieses Logikblocks, so daß das entsprechende Signal über einen Datenausgang **8** den Logikblock **2** verläßt und einem benachbarten Logikblock zugeführt wird (siehe unten).

[0027] Die dargestellte Schaltungsanordnung besitzt weiterhin den Vorteil, daß die Signalwege nicht über Multiplexer/Demultiplexer in Abhängigkeit von der Programmierung der Schaltungsanordnung an verschiedene nachfolgende Logikelemente geschaltet sind, sondern eine feste Verschaltung mit Logikelementen aufweisen, so daß konkret definierte und konstante Lastverhältnisse an den Ausgängen der Logikelemente vorhanden sind. Dadurch werden für die Schaltung zerstörerische Fehlkonfigurationen, die bei einer freien Programmierung auftreten könnten, ausgeschlossen. Es ist generell keine programmierte Konfiguration möglich, die zu einer Überlastung einzelner Logikelemente führen könnte.

[0028] Fig. 2 zeigt ein Blockschaltbild einer zweiten Ausführungsform der Schaltungsanordnung. Der Logikblock **2** enthält bei dieser Ausführungsform nur drei Logikelemente **1**. Die Verschaltung der Logikelemente **1** erfolgt in der oben beschriebenen Weise wiederum kreisförmig, so daß jeder Datenausgang **5** des Logikelements an den ersten Dateneingang **3** des nachfolgenden Logikelements gekoppelt ist. Der zweite Dateneingang jedes Logikelements steht wiederum

als Dateneingang **7** des Logikblocks **2** zur Verfügung. Bei den beiden Ausführungsformen, die in Bezug auf die **Fig. 1** und **2** beschrieben wurden, entspricht die Anzahl der Dateneingänge bzw. der Datenausgänge des Logikblocks jeweils der Anzahl der im Logikblock enthaltenen Logikelemente.

[0029] **Fig. 3** zeigt eine beispielhafte schaltungstechnische Realisierung des Logikelements **1**. Das Logikelement **1** ist hier als logische Verweistabelle (look-up-table) gestaltet. In bekannter Weise besitzt eine solche Verweistabelle neben den Dateneingängen **3, 4** und dem Datenausgang **5** mehrere Programmierereingänge **10**, über deren Voreinstellung die logische Funktion der Verweistabelle programmiert wird. Im dargestellten Beispiel ist die Verweistabelle als XOR-Gatter programmiert, wie dies auch durch die Funktion des Datenausgangs angegeben ist, $Y = a1 \text{ XOR } a2$. Die Logikelemente **1** können jedoch auch durch andere Schaltungselemente realisiert werden. Die verschiedenen Möglichkeiten für die Gestaltung der Logikelemente sind dem Fachmann aus der allgemeinen Schaltungstechnik bekannt, so daß an dieser Stelle auf weitere Beispiele verzichtet werden kann.

[0030] **Fig. 4** zeigt ein Blockschaltbild einer Netzmasche **12**, die eine Weiterbildung der Schaltungsanordnung darstellt. Die Netzmasche **12** ist aus vier Logikblöcken **2** zusammengesetzt, die wiederum kreisförmig zusammengeschaltet sind, so daß sich eine geschlossene Netzmasche ergibt. Bei anderen Ausführungsformen kann die Anzahl der für eine Netzmasche verwendeten Logikblöcke variieren. Die einzelnen Logikblöcke **2** sind derart miteinander gekoppelt, daß jeweils ein Datenausgang **8** eines Logikblocks **2** auf einen Dateneingang **7** des benachbarten Logikblocks geschaltet ist, wobei gleichzeitig eine Kopplung in entgegengesetzter Richtung besteht. Bei der Verwendung gleichmäßig orientierter Logikblöcke **2** zum Aufbau der Netzmasche **12** ergibt sich aus dieser Kopplung, daß die Signalflußrichtung innerhalb eines Logikblocks **2** in entgegengesetztem Richtungssinn gegenüber der kreisförmigen Signalflußrichtung innerhalb der Netzmasche **12** verläuft.

[0031] **Fig. 5** zeigt ein Blockschaltbild eines Abschnittes eines Logiknetzes **15**. Dieses Logiknetz **15** ist aus einer Vielzahl von Netzmaschen zusammengesetzt, so daß sich die Größe des Logiknetzes beliebig an den jeweiligen Anwendungsfall anpassen läßt. Durch einfaches Verbinden mehrerer Netzmaschen an den jeweiligen Randpunkten (siehe **Fig. 4**) kann das Logiknetz in jeder Richtung beliebig fortgesetzt werden. Die Besonderheit der in **Fig. 5** gezeigten Ausführungsform besteht darin, daß das Logiknetz **15** eine räumliche Struktur aufweist. In der gewählten zweidimensionalen Abbildung ist ein erstes Teilnetz **16** und ein zweites Teilnetz **17** erkennbar, die durch unterschiedliche Schattierungen kenntlich gemacht sind. Die beiden Teilnetze **16, 17** erscheinen in dieser Art der Darstellung mit einer unterschiedlichen Orientierung der Signalflußrichtung. Tatsächlich werden jedoch gleichartige Netzmaschen verwendet, wobei das zweite Teilnetz lediglich von der "Rückseite" gesehen wird, da es sich im Vergleich zu dem ersten Teilnetz **16** auf der anderen Seite des räumlichen Logiknetzes **15** befindet. Diese räumliche Struktur des Logiknetzes wird am einfachsten erzeugt, indem bestimmte Randbereiche eines zweidimensionalen Logiknetzes mit anderen Randbereichen dieses Netzes in der bereits beschriebenen Weise verbunden werden.

[0032] **Fig. 6** zeigt eine Detailansicht eines Randbereiches des Logiknetzes **15**. Zwei an sich gleichartige Logikblöcke **2**, die jeweils in Netzmaschen integriert sind, werden an den Randbereichen, die nicht unmittelbar an eine benachbarte Netzmasche angrenzen, mit gegenüberliegenden Netzmaschen verbunden. Aus dieser räumlichen Anordnung resultiert die in der Figur dargestellte Sichtweise. Die Verbind-

ung der Randbereiche erfolgt in der bereits beschriebenen Weise, also durch Kopplung der Dateneingänge und Datenausgänge des jeweiligen Logikblocks.

[0033] Die konkrete räumliche Ausbildung des Logiknetzes kann an die speziellen Anwendungsfälle angepaßt werden. Beispielsweise sind zylindrische Formen des Netzes denkbar. Bei der praktischen Realisierung könnten die Teilnetze des Logiknetzes zwar in verschiedenen Ebenen eines Mikrochips ausgebildet sein, jedoch ist dies aufgrund der einfachen Schaltungsstruktur nicht erforderlich. Vielmehr können die Teilnetze entkoppelt in einer Ebene des Chips liegen, wobei die Logikblöcke des ersten Teilnetzes **16** in den räumlichen Lücken des zweiten Teilnetzes **17** liegen und umgekehrt. Aufgrund der beschriebenen zyklischen Struktur der Logikelemente, Logikblöcke und Netzmaschen kann die gesamte Schaltung weitgehend kreuzungsfrei in einer Ebene eines Mikrochips aufgebaut werden. Da somit eine Großteil der üblichen Kreuzungspunkte entfallen, können die notwendigen Metallisierungsebenen reduziert werden. Für die einfachen Kreuzungen, die sich beim Ineinandersetzen der Teilnetze ergeben, genügen zwei Metallisierungsebenen, die durch herkömmliche Technologien ohne weiteres herstellbar sind.

[0034] Aus der Darstellung zeigt sich, daß unter Anwendung der erläuterten Schaltungsanordnung Logikblöcke und beliebig große Logiknetze aufgebaut werden können, in denen verschiedenste Logikfunktionen realisiert werden können. In Abhängigkeit von der Programmierung der Logikelemente wird der benötigte Signalweg in einem solchen Logiknetz verteilt, so daß keine zusätzlichen Verbindungskapazitäten erforderlich sind. Da unabhängig von der Programmierung jeder Datenausgang eines Logikelements (bei den dargestellten Ausführungsformen) lediglich an zwei Dateneingänge nachfolgender Logikelemente gekoppelt ist, sind jederzeit konstante und definierte Lastverhältnisse vorhanden. Damit sind aber auch konstante Signalverzögerungszeiten vorhanden, die nicht von unterschiedlichen Leitungslängen (bei veränderter Programmierung) abhängig sind, sondern allein durch die Anzahl der verwendeten Logikelemente bestimmt sind. Die Anzahl der in einer programmierbaren Schaltungsanordnung (z. B. FPGA) vorzuhaltenden Verbindungskapazitäten wird durch die Nutzung dieser Schaltungsanordnung drastisch verringert.

Patentansprüche

1. Programmierbare Schaltungsanordnung mit: mehreren Logikelementen (**1**), die wenigstens zwei Dateneingänge (**3, 4**) und einen Datenausgang (**5**) haben, mehreren Logikblöcken (**2**), die jeweils mindestens drei kreisförmig zusammengeschaltete Logikelemente (**1**) besitzen, wobei der Datenausgang (**5**) jedes Logikelements an einen ersten Dateneingang (**3**) des nachfolgenden Logikelements gekoppelt ist und zumindest einige der Datenausgänge der Logikelemente gleichzeitig als Datenausgänge (**8**) des Logikblocks bereitgestellt sind, und wobei ein zweiter Dateneingang (**4**) jedes Logikelements als ein Dateneingang (**7**) des Logikblocks bereitgestellt ist, einer Vielzahl von Netzmaschen (**12**), die jeweils mindestens drei kreisförmig zusammengeschaltete Logikblöcke aufweisen, wobei an den Verknüpfungspunkten der Logikblöcke jeweils ein Dateneingang (**7**) und ein Datenausgang (**8**) der benachbarten Logikblöcke miteinander verbunden sind, und einem Logiknetz (**15**), wobei Teilnetze (**16, 17**) ausgebildet sind, die jeweils eine

- Vielzahl von Netzmaschen enthalten,
 die Teilnetze in einer gemeinsamen Ebene eines Mikrochips alternierend ausgebildet sind, so dass die Logikblöcke eines Teilnetzes in den räumlichen Lücken eines anderen Teilnetzes liegen, und
 die Teilnetze an den Netzrändern miteinander verbunden sind. 5
2. Schaltungsanordnung nach Anspruch 1, dadurch gekennzeichnet, daß jeder Logikblock (2) aus vier Logikelementen (1) besteht, von denen jedes zwei Dateneingänge (3, 4) und einen Datenausgang (5) besitzt, wobei alle Datenausgänge (5) der Logikelemente auch als Datenausgänge (8) des Logikblocks bereitgestellt sind. 10
3. Schaltungsanordnung nach Anspruch 1 oder 2, dadurch gekennzeichnet, daß weiterhin Logikelemente bereitgestellt sind, die Dateneingänge besitzen, die nicht als Dateneingang des Logikblocks dienen. 15
4. Schaltungsanordnung nach Anspruch 1 oder 2, dadurch gekennzeichnet, daß wenigstens einige der Logikelemente mehr als zwei Dateneingänge und/oder mehr als einen Datenausgang besitzen, die in gleicher Weise an die nachfolgenden Logikelemente gekoppelt sind. 20
5. Schaltungsanordnung nach einem der Ansprüche 1 bis 4, dadurch gekennzeichnet, daß die Logikelemente (1) als logische Verweistabellen oder andere programmierbare Schaltungseinheiten ausgebildet sind. 25
6. Schaltungsanordnung nach Anspruch 1, dadurch gekennzeichnet, daß die beiden Verbindungen zwischen zwei benachbarten Logikblöcken (2) jeweils an die Dateneingänge desselben Logikelements gekoppelt sind. 30
7. Schaltungsanordnung nach einem der Ansprüche 1 bis 6, dadurch gekennzeichnet, daß jeweils eine Anzahl von gleichartigen Logikblöcken (2), die der Anzahl der in jedem Logikblock enthaltenen Logikelemente (1) entspricht, zu einer geschlossenen Netzmasche (12) zusammengeschaltet sind, wobei die Signalflußrichtung in der Netzmasche entgegengesetzt zur Signalflußrichtung in den beteiligten Logikblöcken ist. 35 40
8. Schaltungsanordnung nach einem der Ansprüche 1 bis 7, dadurch gekennzeichnet, daß die Struktur des Logiknetzes (15) zumindest teilweise räumlich ausgebildet ist. 45
9. Schaltungsanordnung nach einem der Ansprüche 1 bis 8, dadurch gekennzeichnet, daß das Logiknetz (15) eine Vielzahl von Netzmaschen (12) enthält, die jeweils vier kreisförmig gekoppelte Logikblöcke (2) besitzen, in denen jeweils vier Logikelemente (1) kreisförmig zusammengeschaltet sind. 50
10. Schaltungsanordnung nach einem der Ansprüche 1 bis 9, dadurch gekennzeichnet, daß sie als eine anwenderprogrammierbare Logikmatrix (FPGA) in einen integrierten Schaltkreis implementiert ist. 55
11. Schaltungsanordnung nach einem der Ansprüche 1 bis 9, dadurch gekennzeichnet, daß sie in einem anwenderspezifischen integrierten Schaltkreis (ASIC) implementiert ist. 60

Hierzu 6 Seite(n) Zeichnungen

- Leerseite -

Fig. 1

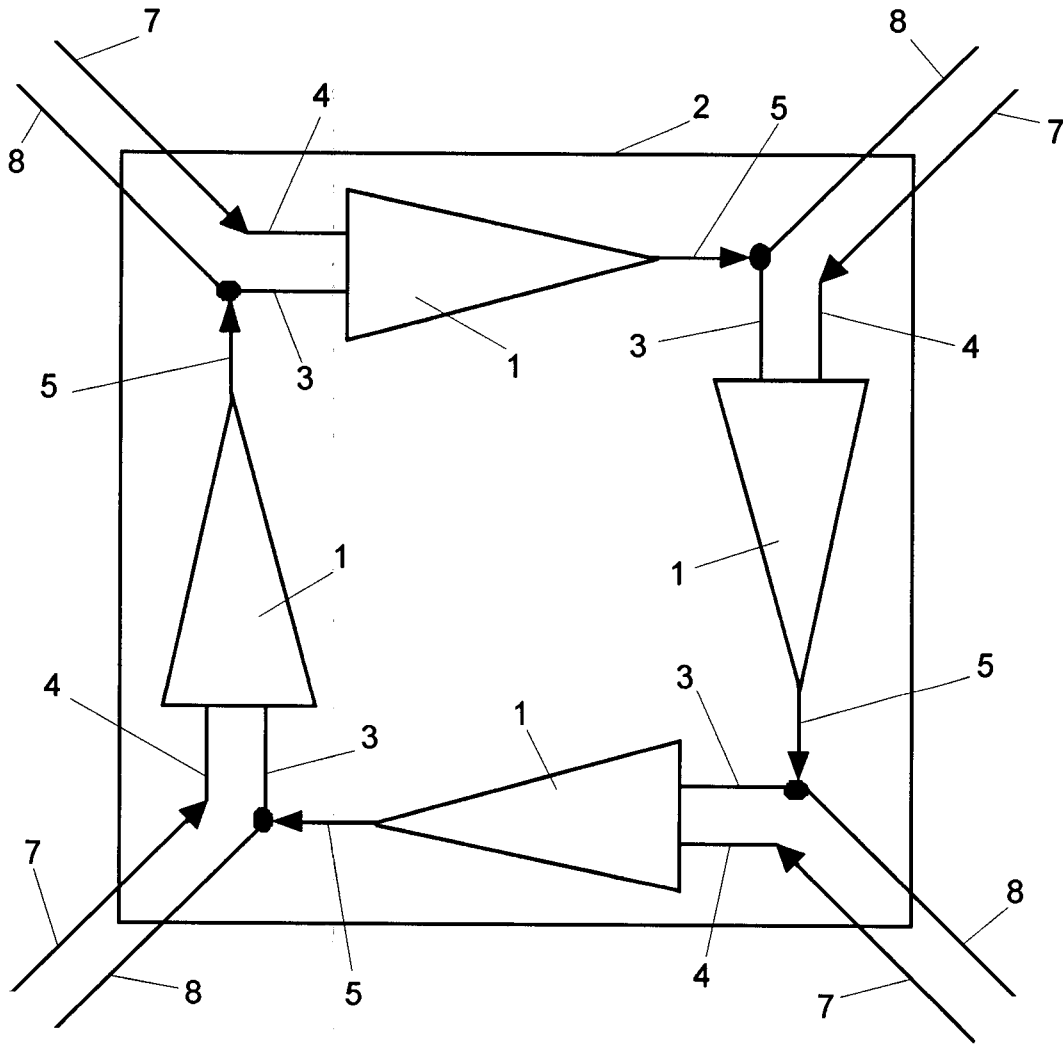


Fig. 2

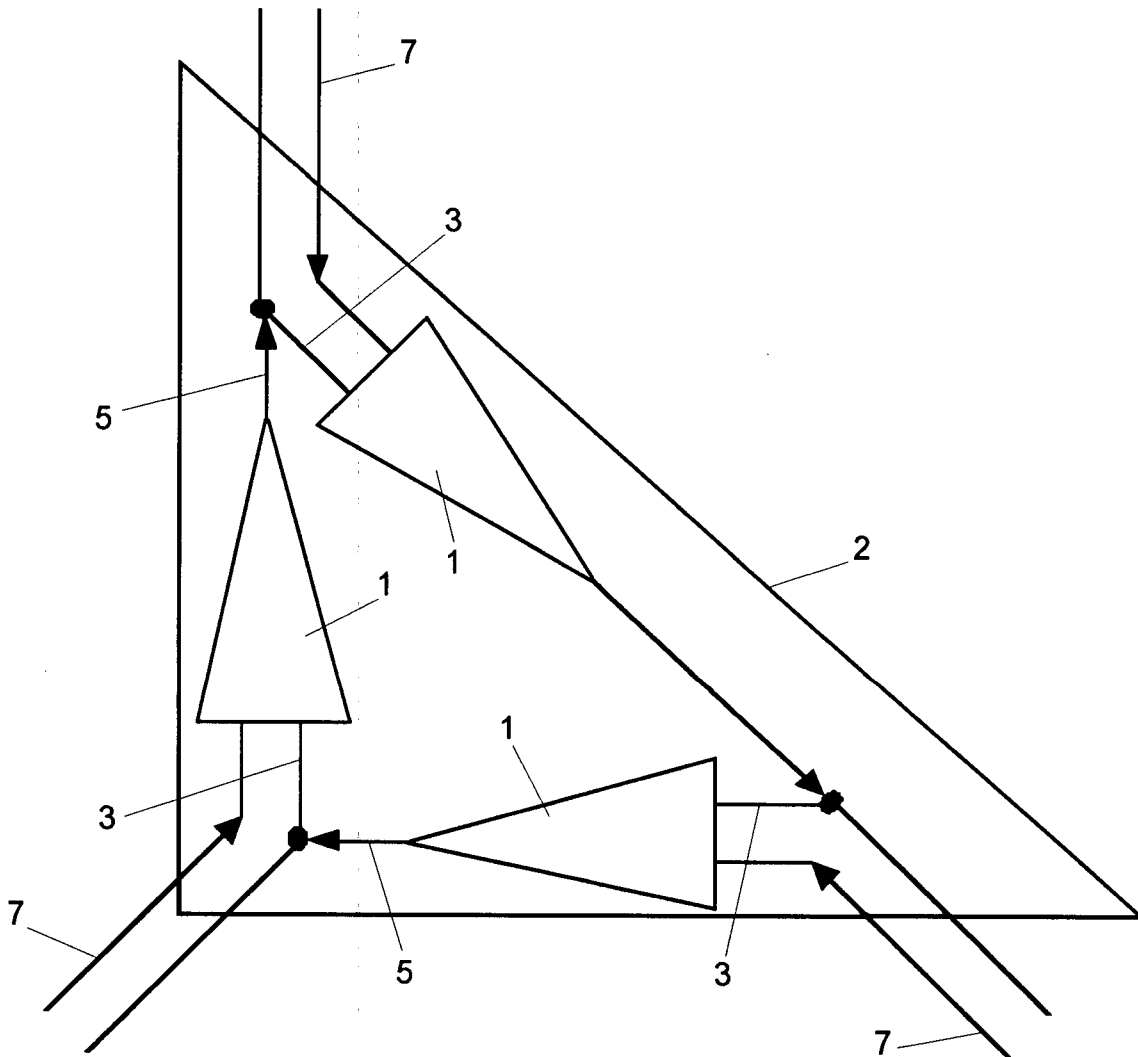


Fig. 3

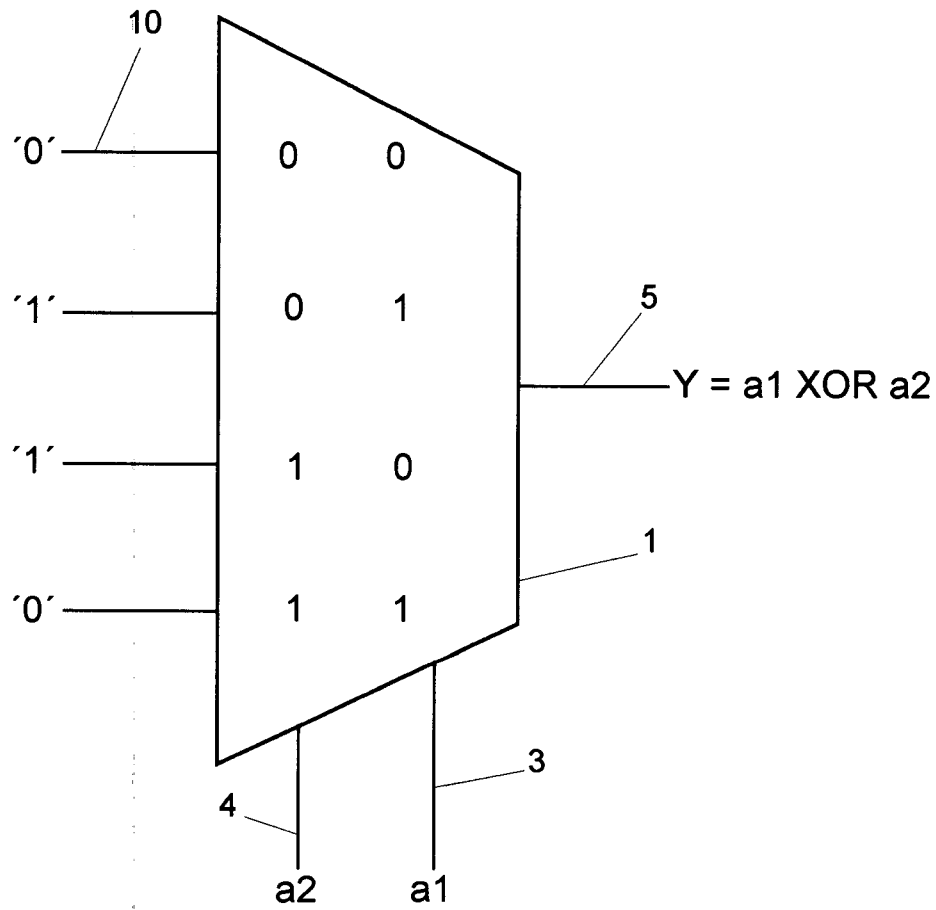


Fig. 4

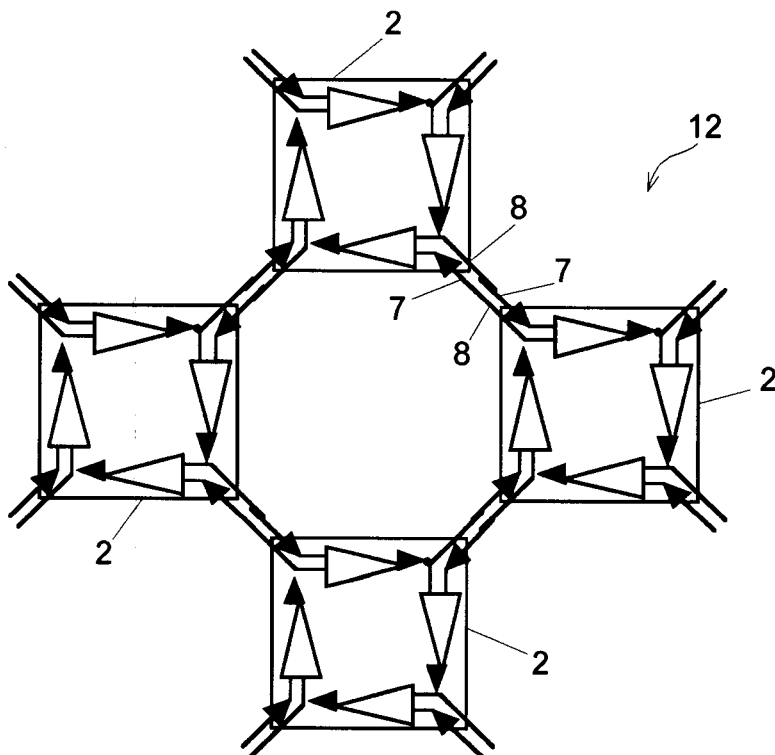


Fig. 5

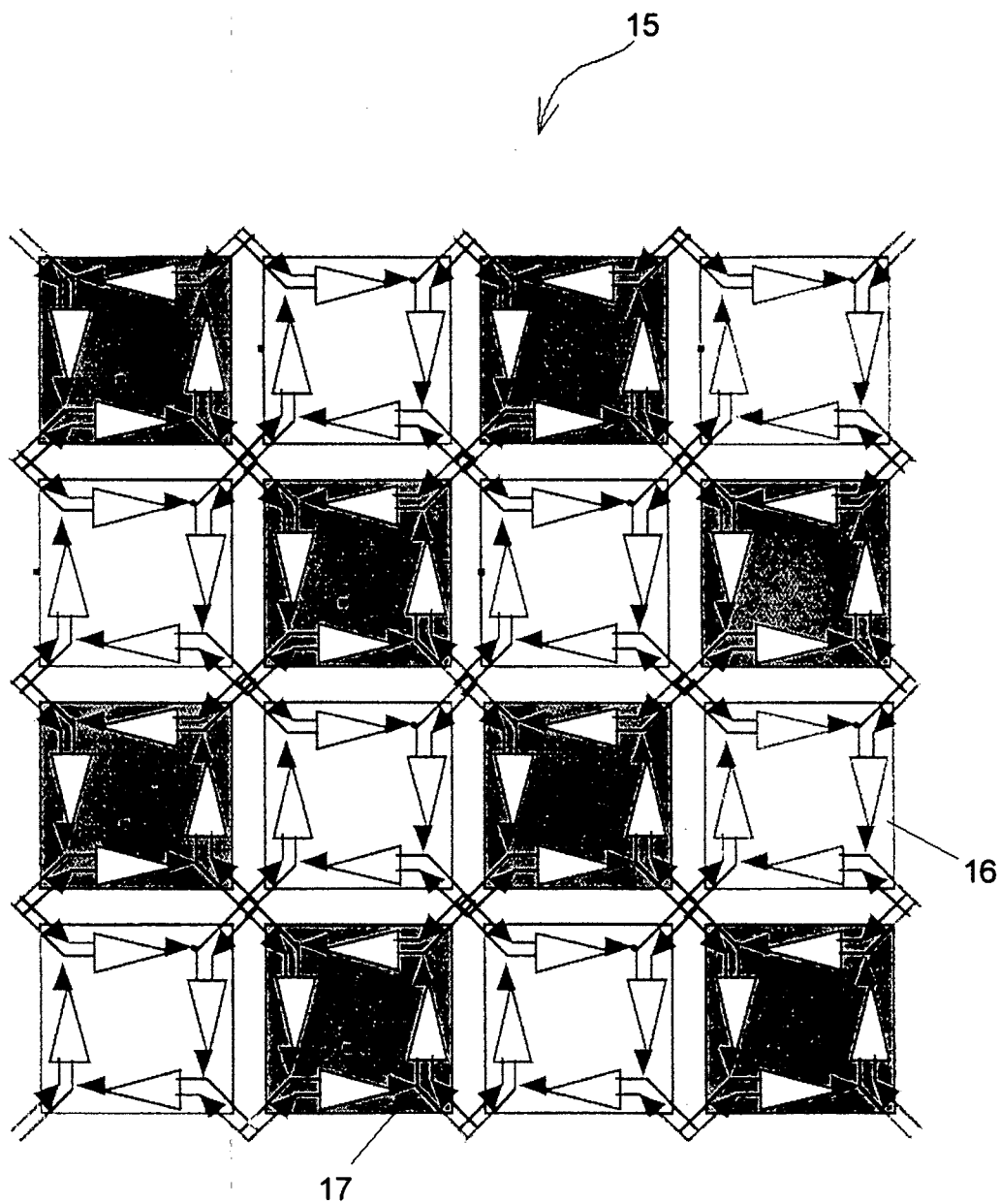


Fig. 6

