



(19)
 Bundesrepublik Deutschland
 Deutsches Patent- und Markenamt

(10) **DE 10 2005 044 728 A1** 2007.03.29

(12)

Offenlegungsschrift

(21) Aktenzeichen: **10 2005 044 728.7**

(22) Anmeldetag: **19.09.2005**

(43) Offenlegungstag: **29.03.2007**

(51) Int Cl.⁸: **G06F 9/44** (2006.01)

(71) Anmelder:

Silicon Software GmbH, 68199 Mannheim, DE

(74) Vertreter:

Köllner & Partner, Patentanwälte, 60596 Frankfurt

(72) Erfinder:

Lay, Ralf, Dr., 68165 Mannheim, DE; Noffz, Klaus-Henning, Dr., 68259 Mannheim, DE

(56) Für die Beurteilung der Patentfähigkeit in Betracht gezogene Druckschriften:

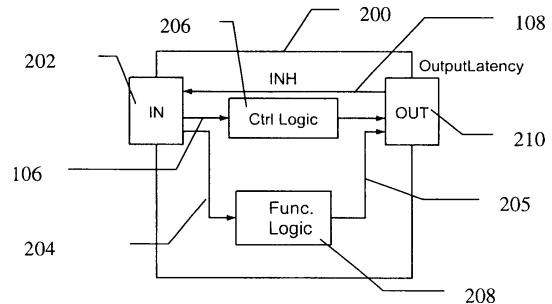
PeaCE - User's Manual, The Cap laboratory of Seoul National University, November 2004;

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

Prüfungsantrag gemäß § 44 PatG ist gestellt.

(54) Bezeichnung: **Programmierung und Layoutdesign von Hardware**

(57) Zusammenfassung: Es wird ein Verfahren zum Programmieren von Hardware für die Verarbeitung von Nutzdaten sowie ein dazu passender graphischer Editor vorgeschlagen. Dazu wird eine Mehrzahl von Modulen (200) zur Verfügung gestellt, wobei jedes Modul mindestens eine Funktion zum Verarbeiten der Nutzdaten ausführen kann. Schnittstellen (202, 210) zum Verschalten der Module werden definiert. Ferner wird vom Anwender eine Verschaltung von Modulen (Topologie), die einer für die Verarbeitung der Nutzdaten geeignete Abfolge von Funktionen entspricht, erstellt. Die Module werden nach vorgegebenen Eigenschaften in eine Mehrzahl von Modultypen klassifiziert. Abhängig von diesen Modultypen sind Verschaltungsregeln definiert, die die zulässigen Verschaltungen der verschiedenen Modultypen angeben. Aus der Topologie wird die Hardware-Programmierung erzeugt.



Beschreibung

Gebiet der Erfindung

[0001] Die Erfindung betrifft ein Verfahren zum Programmieren von Hardware, beispielsweise FPGA's (Field Programmable Gate Array) oder zur Generierung von Layouts für ASIC's (Application Specific Integrated Circuit), für die Verarbeitung von Daten, beispielsweise auf dem Gebiet der industriellen digitalen Bildverarbeitung.

Stand der Technik

[0002] Als Werkzeuge für die Programmierung von Hardware, beispielsweise von FPGA's, dienen Hardwarebeschreibungssprachen, wie u.a. VHDL (Very High Speed Integrated Circuit Hardware Description Language), die es ermöglichen, komplizierte digitale Systeme zu beschreiben.

[0003] Ein Problem bei den meisten Hardwarebeschreibungssprachen ist jedoch, dass Hardwaremodule formuliert werden, die in ihrer Schnittstelle, an die das Modul angeschlossen werden soll, feste, implizite Annahmen an den Daten- und Kontrollfluss (Protokoll und Daten) machen. Das bedeutet, dass zunächst keine Hardwaremodule direkt miteinander verbunden werden können, ohne die genaue Implementierung der Module zu kennen. Dies widerspricht modernen Ansätzen der Softwareentwicklung und behindert die Wiederverwertbarkeit von Code.

[0004] Es ist bekannt, dass Hardwaremodule nur dann miteinander verbunden werden können, wenn ihre genaue Implementierung bekannt ist. Das Fehlen von kompatiblen Schnittstellen zwischen Modulen bedeutet gleichzeitig eine fehlende Portierbarkeit von Code von einer Hardware auf eine andere. Das Auftreten eines eventuellen Datenstaus oder von Datenverlusten ist nicht auszuschließen bzw. nur mit erheblichem Aufwand zu verhindern.

[0005] Diese Situation erfordert gegenwärtig noch eine aufwendige Programmierung der Hardwaremodule, beispielsweise von FPGA's bzw. ASIC's.

Aufgabenstellung

[0006] Aufgabe der Erfindung ist es, ein Verfahren zum Programmieren von Hardware für die Verarbeitung von Nutzdaten anzugeben.

Lösung

[0007] Diese Aufgabe wird durch die Erfindungen mit den Merkmalen der unabhängigen Ansprüche gelöst. Vorteilhafte Weiterbildungen der Erfindungen sind in den Unteransprüchen gekennzeichnet. Der Wortlaut sämtlicher Ansprüche wird hiermit durch Be-

zugnahme zum Inhalt dieser Beschreibung gemacht. Die Erfindung umfasst auch alle sinnvollen und insbesondere alle erwähnten Kombinationen von unabhängigen und/oder abhängigen Ansprüchen.

[0008] Die Aufgabe wird u.a. durch ein Verfahren gelöst. Im Folgenden werden einzelne Verfahrensschritte näher beschrieben. Die Schritte müssen nicht notwendigerweise in der angegebenen Reihenfolge durchgeführt werden, und das zu schildernde Verfahren kann auch weitere, nicht genannte Schritte aufweisen.

[0009] Zunächst wird für das Verfahren zum Programmieren von Hardware für die Verarbeitung von Nutzdaten eine Mehrzahl von Modulen zur Verfügung gestellt, wobei jedes Modul mindestens eine Funktion zum Verarbeiten der Nutzdaten ausführen kann. Die Funktion wird durch den Anwender bestimmt und hängt davon ab, welche Prozesse durch die zu programmierende Hardware realisiert werden sollen.

[0010] Als Modul wird eine abgegrenzte Einheit bezeichnet, die in Hardware realisiert oder in einer höheren Repräsentationssprache dargestellt ist, welche eine bestimmte Funktion ausführen kann und über Schnittstellen mit anderen Modulen kommunizieren kann. Module können dabei fest vorgegeben sein oder flexibel parametrisierbar sein.

[0011] Als nächster Schritt werden Schnittstellen zum Verschalten der Module vorgegeben. Dabei gilt das Prinzip, dass eine Daten ausgebende Schnittstelle am Ausgang eines ersten Moduls mit einer Daten empfangenden Schnittstelle am Eingang des nächsten Moduls übereinstimmen muss, d. h. die diesbezüglichen Parameter der Module müssen übereinstimmen.

[0012] Die Module werden daher so parametrisiert, wie es für die benötigte Funktion erforderlich ist und dass Aus- bzw. Eingänge aufeinander so abgestimmt sind, dass ein Zusammenwirken der Modulen gewährleistet ist.

[0013] Im Detail gibt es für jedes Modul vier Implementierungen:

- die parametrisierbare Beschreibung des Moduls, das später synthetisiert, d. h. auf die Hardware übersetzt wird;
- eine Simulationsbeschreibung des Moduls;
- eine Beschreibung der Schnittstelle (Links, Parameter, Format, Bereichsgrenzen etc.);
- eine Beschreibung der Zugriffsschnittstelle in Hardware, die angibt, welche Parameter bzw. Register ein Anwender während der Laufzeit im Modul verändern kann und wie dies möglich ist.

[0014] Ein weiterer Verfahrensschritt besteht darin, dass eine Verschaltung von Modulen (eine Topolo-

gie), die einer für die Verarbeitung der Nutzdaten geeignete Abfolge von Funktionen entspricht, vorgegeben wird.

[0015] Danach wird aus der Topologie die Hardware-Programmierung erzeugt.

[0016] Die Modelle und Regelwerke, die dem Verfahren zugrunde liegen, enthalten Mechanismen, die es ermöglichen, die Hardwaremodule so zu formulieren, dass sie aneinander anschließbar sind. Dabei kann eine Programmierung von Hardware (beispielsweise FPGA's oder ASIC's) durch den Anwender ohne genaue Detailkenntnisse des Hardwareentwurfs realisiert werden.

[0017] Die Module werden dazu nach vorgegebenen Eigenschaften in eine Mehrzahl von Modultypen klassifiziert. Die Eigenschaften für die Klassifizierung in unterschiedliche Modultypen werden dahingehend ausgewählt, dass sie konkrete Aussagen darüber beinhalten, wie Module zur Realisierung der Topologie miteinander verschaltet werden können bzw. welche Verschaltungen zulässig sind.

[0018] In Abhängigkeit von den Modultypen werden konkrete Verschaltungsregeln definiert.

[0019] An der Schnittstelle eines Moduls existiert eine Datenstruktur (Link), die aus verschiedenen Signalen besteht. Es sind das die Nutzdatensignale (DATA), und/oder Kontrollsignale (CTRL), das Taktsignal (CLK) sowie evtl. ein Inhibit-Signal (INH).

[0020] Über die Links wird der Datenaustausch zwischen den Modulen durchgeführt. Im Prinzip stellen sie eine gerichtete Datenverbindung her, die zusätzlich über Kontrollmechanismen zur Datenflusskontrolle verfügen. Der Typ des Datenformats wird strukturiert festgelegt und muss zwischen Sender und Empfänger übereinstimmen.

[0021] Dabei transportieren die Nutzdatensignale die Nutzdaten. Sie können beliebig aufgebaut und strukturiert sein. Insbesondere sind dabei auch verschachtelte Strukturen zulässig.

[0022] Die Kontrollsignale übermitteln Zusatzinformationen zeitgleich zu den Nutzdatensignalen. Aus den Kontrollsignalen muss sich die Gültigkeit der Nutzdaten erkennen lassen. Dazu kann ein bestimmtes einzelnes Signal oder auch eine Kombination mehrerer Signale genutzt werden. Ein Beispiel für ein CTRL-Signal ist u. a. ein Valid für eine Datumsvariable.

[0023] Die Steuerung des Datenflusses erfolgt über Inhibit-Signale (INH). Dabei kann durch ein aktives Inhibit-Signal der Nutzdatenfluss gestoppt werden.

[0024] Für den Empfang von Daten sind Input-Links und für die Ausgabe von Daten sind Output-Links definiert.

[0025] Dabei ermöglichen die Schnittstellen die Verbindung eines Ausgangs eines ersten Moduls mit einem Eingang eines zweiten Moduls. Für jeden Ausgang eines Moduls wird eine Datenstruktur vom Typ Output-Link vorgesehen, während für jeden Eingang eines Moduls eine Datenstruktur vom Typ Input-Link vorgesehen wird.

[0026] Bei einem Input-Link sind alle Daten- und Kontrollsignale sowie das Taktsignal Eingangssignale, das Inhibit-Signal jedoch ein Ausgangssignal. Bei einem Output-Link sind alle Daten- und Kontrollsignale sowie das Taktsignal Ausgänge, das Inhibit-Signal ist jedoch Eingang. Alle Änderungen der Signale erfolgen synchron zum Takt. Input- und Output-Links können nur miteinander verbunden werden, wenn sie denselben Takt verwenden.

[0027] Die korrekte Bearbeitung eingehender Daten wird in einem Modul durch mindestens eine Funktions-Logik realisiert. Die Funktions-Logik ist eine in Hardware realisierte Funktion zur Verarbeitung der Daten.

[0028] Ausgangsseitig wird ein Modul durch eine Output-Latency charakterisiert. Die Output-Latency ist eine Moduleigenschaft und wird vom Programmierer des Moduls ermittelt und zur Verfügung gestellt. Es gibt Module mit konstanten Output-Latencys aber auch Module, bei der die Output-Latencys von den Einstellungen der Parameter abhängt. Die Output-Latency eines Moduls gibt an, welche maximale Anzahl von gültigen Datenworten der Output-Link noch senden kann, wenn keine neuen Daten mehr eingehen, d. h. wie viele Daten ein Modul noch senden kann, wenn man es anhält. Diese Anzahl muss von der nachfolgenden Verarbeitungskette immer aufgenommen werden können, wenn kein Datenverlust eintreten soll. Ist dies durch die verwendeten Module nicht gewährleistet, so werden automatisch die entsprechenden Steuerelemente in Form von Verzögerungs- oder Puffer- oder Fifo-Elementen eingefügt, um diesen Zustand zu gewährleisten.

[0029] Die Datenströme zwischen den Modulen werden durch Quellmodule erzeugt, durch Verarbeitungsmodule bearbeitet und verändert und durch Senkenmodule angenommen. Die Verbindung dieser Module über die Links erzeugt ein sich selbst regelndes Netzwerk von Datenflüssen. Die Prinzipien dieses Regelkreislaufes und die möglichen Topologien des Netzwerkes sind durch das erfindungsgemäße Regelwerk festgelegt. Das Regelwerk ist so ausgelegt, dass nicht nur die lokalen Verbindungen untereinander in Datenfluss und Datenformat angepasst werden, sondern dass auch garantiert wird, dass der

Gesamtdatenfluss durch das Design (Topologie) verlustfrei transportiert wird.

[0030] Für das erfindungsgemäße Verfahren werden, wie oben beschrieben, verschiedene Modultypen mit jeweils konkreten Eigenschaften definiert. Zur Unterscheidung werden dabei die Module als Module der Typen "O" (O-Modul), "P" (P-Modul) und "M" (M-Modul) bezeichnet. Die Typ-Bezeichnung kennzeichnet jeweils sowohl die Eigenschaften eines Moduls als auch dessen Verschaltungsregeln. Die Wahl der Namen ist willkürlich.

O-Modul

[0031] O-Module sind die einfachsten Module. Ein O-Modul kann die Werte von Nutzdaten ändern, aber keine Nutzdaten löschen.

[0032] Der Kontrolldatenfluss wird durch das O-Modul nicht beeinflusst.

[0033] Das in einem Link enthaltene Inhibit-Signal hat keinen Einfluss auf das O-Modul.

[0034] Die Daten erfahren beim Durchlaufen des O-Moduls eine Verzögerung (Latency) mit einer genau definierten Anzahl von Takten, die in der Regel der Anzahl der Registerstufen entspricht, die die Nutzdaten auf dem Datenpfad vom Input-Link zum Output-Link durchlaufen. Ein O-Modul ist daher ein deterministisches Modul mit genau N Takten.

[0035] Besitzt ein O-Modul mehreren Input-Links, so müssen die Nutzdaten an diesen gleichphasig anliegen. Das Inhibit-Signal wird bei einem O-Modul mit mehreren Input-Links identisch auf alle Input-Links geleitet.

[0036] Auf Grund seiner Eigenschaften kann ein O-Modul keinen Datenstau und auch keinen Datenverlust verursachen. Die Nutzdatenmenge ändert sich bei einem O-Modul nicht, die Datenwerte können sich jedoch durchaus ändern, z. B. durch Addition. Es werden keine Daten gelöscht oder erzeugt. Ebenfalls ändert sich nicht die Anzahl der Validis. Es müssen bei einem O-Modul nicht alle Registerstufen tatsächlich gültige Daten enthalten.

P-Modul

[0037] Ein weiterer für das beschriebene Verfahren verwendeter Modultyp ist das P-Modul. Es hat einen komplexeren Charakter, als ein O-Modul.

[0038] Ein P-Modul kann die Werte von Nutzdaten ändern und Nutzdaten löschen. Ein P-Modul hat insbesondere die Möglichkeit, weniger Datenworte zu liefern, als es empfängt.

[0039] Ebenso kann das P-Modul den Kontrolldatenfluss ändern und löschen. Auf Grund seiner Eigenschaften kann ein P-Modul den Kontrolldatenfluss beeinflussen, aber nur derart, dass die absolute Datenrate reduziert wird. Daraus resultiert, dass die Anzahl und die zeitliche Lage der Kontrollsignale einschließlich Data-Valid variabel sind. Mit anderen Worten: ein P-Modul kann den Valid Strom verändern, indem es die Anzahl der Validis reduzieren kann.

[0040] Ein Inhibit-Signal hat keinen Einfluss auf das P-Modul.

[0041] Die Daten erfahren beim Durchlaufen des P-Moduls eine genau definierte maximale Verzögerung (maximale Latency). Dabei kann die Verzögerung, die einzelne Nutzdaten erfahren, auch kleiner als die maximale Verzögerung sein. Ein P-Modul ist daher ebenfalls ein deterministisches Modul mit max. N Takten.

[0042] Bei einem P-Modul mit mehreren Input-Links müssen die Daten an diesen gleichphasig anliegen. Die Funktions-Logik wird dabei mit Kontrolldaten aus mindestens einem der Input-Links gespeist. Das Inhibit-Signal wird bei einem P-Modul mit mehreren Input-Links identisch auf alle Input-Links geleitet.

[0043] Nach einem Stoppen des Eingangsdatenflusses kann ein P-Modul maximal die Anzahl von Datenworten liefern, die durch die Output-Latency definiert ist.

[0044] Bei einem aktiven Inhibit-Signal können durch das Modul noch beliebig viele Datenworte aufgenommen werden. Ein Inhibit-Signal passiert ein P-Modul ohne Verzögerung. D. h., dass ein P-Modul ein Inhibit einfach nur weiter leitet. Es verändert den Inhibitstrom nicht. Diss ist auch nicht erforderlich, da ein P-Modul den Datenstrom immer nur reduzieren, aber nicht erhöhen kann. Eine aktive Steuerung des Inhibitsignals ist jeweils nur dann notwendig, wenn mehr Daten erzeugt werden können, als empfangen wurden, oder wenn eine unbestimmte Verzögerung eintreten kann, sodass ein Anhalten der Kette eingangsseitig notwendig sein kann, weil die Daten ausgangsseitig nicht mehr abgenommen werden.

[0045] Ein Beispiel für ein P-Modul ist u. a. die Umwandlung von 24 Bit RGB-Bilddaten in 32 Bit verschachtelte Bilddaten. Bei einem Vorgang dieser Art wird die Anzahl der Datenworte und damit der Kontrollsignale reduziert.

M-Modul

[0046] Ein dritter für das beschriebene Verfahren verwendeter Modultyp ist das M-Modul. Dieses ist das komplexeste der vorgeschlagenen Module.

[0047] Das M-Modul kann die Werte von Nutzdaten ändern, Nutzdaten löschen und neue Nutzdaten erzeugen.

[0048] Den Kontrolldatenfluss kann das M-Modul ändern, löschen und erzeugen.

[0049] Weiterhin kann das M-Modul ein Inhibit-Signal erzeugen.

[0050] Die Daten erfahren beim Durchlaufen des M-Moduls eine unbestimmte Verzögerung. Ein M-Modul ist daher ein nicht deterministisches Modul.

[0051] Auf ihrer jeweiligen Eingangsseite besitzen M-Module einen Eingangspuffer mit einer bestimmten Puffergröße (Buffer-Size) zur Speicherung von Daten. Diese Buffer-Size des Input-Links eines Moduls bestimmt, welche maximale Anzahl von gültigen Datenworten der Link noch aufnehmen kann, nachdem das Inhibit-Signal aktiviert wurde.

[0052] Bei einem M-Modul mit mehreren Input-Links können eingehende Nutzdaten sowie Kontrolldaten aus unabhängigen Quellen (Input-Links) stammen.

[0053] Ein M-Modul kann auf Grund seiner Eigenschaften den Datenfluss explizit beeinflussen, wobei der Datenfluss auch vollständig angehalten werden kann. Der Datenfluss wird über Inhibit-Signale gesteuert. Bei einem aktivierten Inhibit-Signal kann maximal noch die durch die Output-Latency des M-Moduls bestimmte Anzahl Datenworte ausgegeben werden. Die Anzahl und die zeitliche Lage der Kontrollsignale einschließlich Data-Valid ist variabel. Ein M-Modul kann weniger oder mehr Datenworte liefern als es empfängt.

[0054] Wie oben beschrieben, erfolgen alle Änderungen von Signalen synchron zum Takt. Das bedeutet jedoch keine Einschränkung auf Systeme, die nur einen einzigen Takt haben. Während O- und P-Module das gleiche Taktsignal an den Input- und Output-Links benötigen, können M-Module auf der Input- und Output-Seite unterschiedliche Takt-Signale (CLK-Signale) verarbeiten. Die Synchronisierung der Daten bezüglich der unterschiedlichen Input- und Output-Clocks wird im M-Modul vorgenommen. Somit können auch Hardwarebeschreibungen mit mehreren unterschiedlichen synchronen Taktsignalen erzeugt werden.

Verschaltung

[0055] Die Verschaltung von Modulen der beschriebenen unterschiedlichen Modultypen erfolgt nach konkreten, für jeden Modultyp spezifischen Regeln.

[0056] Die Verschaltung des Ausgangs eines O- oder P-Moduls mit dem Eingang eines nachfolgen-

den O- oder P-Moduls, d. h. eine Serienschaltung dieser Module, ist uneingeschränkt möglich. Die maximale Output-Latency der Gesamtanordnung ergibt sich aus der Summe der maximalen Output-Latencies der einzelnen Module. Die Gesamtanordnung kann nach aktivem Inhibit-Signal noch beliebig viele Daten aufnehmen.

[0057] Bei der Verschaltung des Ausgangs eines O- oder P-Moduls mit dem Eingang eines M-Moduls (Serienschaltung) muss die Buffer-Size des M-Moduls größer sein als die Output-Latency des O- oder P-Moduls. Andernfalls muss ein Puffer eingefügt werden, der die überzähligen Datenworte zwischenspeichern kann. Dies wird automatisch durchgeführt. Die Gesamtanordnung kann nach aktivem Inhibit-Signal nur noch so viele Daten aufnehmen, wie durch Buffer-Size und einen eventuell eingefügten Puffer spezifiziert ist. Die Output-Latency der Gesamtanordnung entspricht der Output-Latency des M-Moduls.

[0058] Die Verschaltung des Ausgangs eines M-Moduls mit dem Eingang eines O- oder P-Moduls (Serienschaltung) ist uneingeschränkt möglich. Die Gesamtanordnung kann nach aktivem Inhibit-Signal nur noch so viele Daten aufnehmen wie durch die Buffer-Size des M-Moduls spezifiziert ist. Bei der Bestimmung der maximale Output-Latency der Gesamtanordnung ist die Reihenfolge wichtig:

a) O/P-M: Das M-Modul entkoppelt die Logik des O-Moduls und reagiert auf ein Inhibit in der Topologie rechts von diesem zusammengesetzten Modul. Damit bestimmt in dieser Anordnung das M-Modul allein die OutputLatency der Verschaltung.

b) M/O/P: Hier addiert sich die OutputLatency des M-Moduls mit der maximalen OutputLatency des O- oder P-Moduls.

[0059] Wird der Ausgang eines ersten M-Moduls mit dem Eingang eines zweiten M-Moduls (Serienschaltung) verschaltet, so muss dabei die Buffer-Size des zweiten M-Moduls mindestens so groß sein wie die Output-Latency des ersten M-Moduls. Andernfalls muss ein Puffer eingefügt werden, der die überzähligen Datenworte zwischenspeichern kann, was automatisch erfolgt. Die Gesamtanordnung kann nach aktivem Inhibit-Signal nur noch so viele Daten aufnehmen wie durch die Buffer-Size des ersten M-Moduls spezifiziert ist. Die Output-Latency der Gesamtanordnung entspricht der Output-Latency des zweiten M-Moduls.

[0060] Bei Parallelschaltungen verschiedener Modultypen ist entscheidend, an die Eingänge welchen Modultyps die Ausgänge der parallel geschalteten Module wieder zusammengeführt werden.

[0061] Bei einem O- oder P-Modul müssen die CTRL-Signale aller Input-Links synchron sein. Daher

gelten hier die engsten Anforderungen. So muss bei der Parallelschaltung eines ersten und mindestens eines zweiten O-Moduls, deren Ausgänge mit den Eingängen eines dritten O- oder P-Moduls verschaltet werden, hinter oder vor dem ersten oder zweiten O-Modul, je nachdem, welches die kleinere Output-Latency hat, ein Verzögerungsregister mit so vielen Stufen eingefügt werden, wie die Differenz der Output-Latencies des ersten und zweiten O-Moduls beträgt, damit an den Eingängen des dritten Moduls eine Synchronisation der Daten gewährleistet bleibt. Die Output-Latency der Gesamtanordnung entspricht dabei der Summe aus der maximalen Output-Latency der parallel geschalteten O-Module und der maximalen Output-Latency des in Serie geschalteten O- oder P-Moduls.

[0062] Eine Parallelschaltung eines ersten P- oder M-Moduls mit mindestens einem zweiten P- oder M-Modul, deren Ausgänge mit den Eingängen eines dritten O- oder P-Moduls verschaltet werden, ist nicht zulässig. Das resultiert daraus, dass P- und M-Module über keine fest definierte Latency verfügen, so dass keine Gleichphasigkeit der Daten erreicht werden kann.

[0063] Zulässig ist jedoch eine Parallelschaltung eines ersten O-, P- oder M-Moduls mit mindestens einem zweiten O-, P- oder M-Modul, wenn deren Ausgänge mit den Eingängen eines dritten M-Moduls verschaltet werden.

[0064] Bei M-Modulen können die CTRL-Signale aller Input-Links unabhängig sein. Hier sind daher alle Kombinationen zulässig. Es wird kein Latency-Ausgleich der einzelnen Pfade vorgenommen. Die Datenflusskontrolle erfolgt ausschließlich anhand der CTRL- und INH-Signale.

[0065] Das vorgeschlagene Verfahren ist unter anderem auch dadurch gekennzeichnet, dass überprüft wird, ob die Verschaltungsregeln durch die Topologie eingehalten sind. Werden Fehler erkannt, so werden diese entweder gemeldet oder die Topologie wird derart ergänzt- sofern möglich -, dass die Verschaltungsregeln erfüllt sind. Dadurch werden Design-Fehler minimiert.

[0066] Auch kann in dem vorgeschlagenen Verfahren eine Verschachtelung von Modulanordnungen erfolgen. Dabei ist die Funktions-Logik eines Moduls aus einer Anordnung mehrerer Module aufgebaut. Bei der Verschachtelung sind beliebig tiefe Strukturen möglich. Um solche Anordnungen realisieren zu können, müssen Verbindungen zwischen Input-Links und Verbindungen zwischen Output-Links zulässig sein.

[0067] Bei einer solchen Verschachtelung werden ebenfalls ggf. nötige Verzögerungselemente und Puf-

fer automatisch eingefügt. Dadurch kann es niemals zu Datenverlusten, Datenstau oder Deadlock-Situationen kommen. Insgesamt hat das Design eine geringe Fehlerwahrscheinlichkeit.

[0068] Das erfindungsgemäße Verfahren ist so angelegt, dass die Möglichkeit besteht, eine Mehrzahl von Verschachtelungen und Verschaltungen von Modulen (Baustein) zum Realisieren jeweils einer vorgegebenen Funktion zu erstellen, woraus eine Bibliothek aufgebaut werden kann.

[0069] Zur Darstellung von Modulen für die Verarbeitung von Nutzdaten können graphische Symbole verwendet werden. Damit wird die Möglichkeit geschaffen, auf graphisch orientierter Basis eine Topologie nach dem Baukastenprinzip aus Modulen und Bausteinen aus einer Bibliothek zu erstellen.

Graphischer Editor

[0070] Die Aufgabe wird auch gelöst durch einen graphischen Editor für die Programmierung von Hardware mit graphischen Symbolen zur Darstellung von Modulen, wobei jedes Modul eine Funktion zum Verarbeiten von Daten ausführen kann, wenn das Modul in Hardware realisiert wird.

[0071] Der graphische Editor verfügt mindestens über ein graphisches Symbol zur Darstellung von Schnittstellen zum Verschalten der Module.

[0072] Weiterhin ist der graphische Editor mindestens mit einem graphischen Symbol zur Darstellung der Verschaltung von Modulen ausgestattet, wodurch eine Verschaltung einer Mehrzahl von Modulen (Topologie) darstellbar ist.

[0073] Der graphische Editor besitzt auch die Mittel zum Übersetzen der graphischen Wiedergabe der Topologie in eine Programmierung der Hardware.

[0074] Außerdem besitzt der graphische Editor die Mittel zum Überprüfen, ob eine zulässige Verschaltung der Module in der Topologie vorliegt.

[0075] Ferner wird die Aufgabe gelöst durch ein Computerprogramm, das bei Ablauf auf einer Recheneinheit, einem Mikrocontroller, DSP, FPGA oder Computer oder auf einer Mehrzahl davon in einem Netzwerk das erfindungsgemäße Verfahren oder den graphischen Editor in einer ihrer Ausgestaltungen ausführt.

[0076] Weiterhin wird die Aufgabe gelöst durch ein Computerprogramm mit Programmcode-Mitteln, um das erfindungsgemäße Verfahren in einer seiner Ausgestaltungen durchzuführen oder den graphischen Editor zu realisieren, wenn das Programm auf einer Recheneinheit, einem Mikrocontroller, DSP,

FPGA oder Computer oder auf einer Mehrzahl davon in einem Netzwerk ausgeführt wird. Insbesondere können die Programmcode-Mittel auf einem computerlesbaren Datenträger gespeicherte Instruktionen sein.

[0077] Außerdem wird die Aufgabe gelöst durch einen Datenträger, auf dem eine Datenstruktur gespeichert ist, die nach einem Laden in einen Arbeits- und/oder Hauptspeicher einer Recheneinheit, eines Mikrocontrollers, DSPs, FPGAs oder Computers oder einer Mehrzahl davon in einem Netzwerk das erfindungsgemäße Verfahren oder den graphischen Editor in einer ihrer Ausgestaltungen ausführen kann.

[0078] Auch wird die Aufgabe gelöst durch ein Computerprogramm-Produkt mit auf einem maschinenlesbaren Träger gespeicherten Programmcode-Mitteln, um das erfindungsgemäße Verfahren in einer seiner Ausgestaltungen durchzuführen oder den graphischen Editor zu realisieren, wenn das Programm auf einer Recheneinheit, einem Mikrocontroller, DSP, FPGA oder Computer oder auf einer Mehrzahl davon in einem Netzwerk ausgeführt wird.

[0079] Dabei wird unter einem Computer-Programmprodukt das Programm als handelbares Produkt verstanden. Es kann grundsätzlich in beliebiger Form vorliegen, so zum Beispiel auf Papier oder einem computerlesbaren Datenträger und kann insbesondere über ein Datenübertragungsnetz verteilt werden.

[0080] Schließlich wird die Aufgabe gelöst durch ein modulierte Datensignal, welches von einer Recheneinheit, einem Mikrocontroller, DSP, FPGA oder Computer oder von einer Mehrzahl davon in einem Netzwerk ausführbare Instruktionen zum Ausführen des erfindungsgemäßen Verfahrens oder des graphischen Editors in einer ihrer Ausgestaltungen enthält.

[0081] Als Computersystem zum Ausführen des Verfahrens oder zum Realisieren des graphischen Editors kommen sowohl ein Stand-alone Computer oder Mikrocontroller, DSPs oder FPGAs in Betracht, als auch ein Netzwerk von Mikrocontrollern, DSPs, FPGAs oder Rechnern, beispielsweise ein hausinternes, geschlossenes Netz, oder auch Rechner, die über das Internet miteinander verbunden sind. Ferner kann das Computersystem durch eine Client-Server-Konstellation realisiert sein, wobei Teile der Erfindung auf dem Server, andere auf einem Client ablaufen.

[0082] Weitere Einzelheiten und Merkmale ergeben sich aus der nachfolgenden Beschreibung von bevorzugten Ausführungsbeispielen in Verbindung mit den Unteransprüchen. Hierbei können die jeweiligen Merkmale für sich alleine oder zu mehreren in Kombination miteinander verwirklicht sein. Die Möglich-

keiten, die Aufgabe zu lösen, sind nicht auf die Ausführungsbeispiele beschränkt.

Ausführungsbeispiel

[0083] Die Ausführungsbeispiele sind in den Figuren schematisch dargestellt. Gleiche Bezugsziffern in den einzelnen Figuren bezeichnen dabei gleiche oder funktionsgleiche bzw. hinsichtlich ihrer Funktionen einander entsprechende Elemente. Im Einzelnen zeigt:

[0084] [Fig. 1](#) eine schematische Darstellung der Datenstruktur eines Input-Links und eines Output-Links;

[0085] [Fig. 2](#) eine schematische Darstellung eines O-Moduls mit einem Input-Link;

[0086] [Fig. 3](#) eine schematische Darstellung eines O-Moduls mit zwei Input-Links

[0087] [Fig. 4](#) eine schematische Darstellung eines P-Moduls mit einem Input-Link;

[0088] [Fig. 5](#) eine schematische Darstellung eines P-Moduls mit zwei Input-Links

[0089] [Fig. 6](#) eine schematische Darstellung eines M-Moduls mit einem Input-Link;

[0090] [Fig. 7](#) eine schematische Darstellung eines M-Moduls mit zwei Input-Links;

[0091] [Fig. 8](#) eine schematische Darstellung der Serienschaltung des Ausgangs eines O-Moduls an den Eingang eines O-Moduls;

[0092] [Fig. 9](#) eine schematische Darstellung der Serienschaltung des Ausgangs eines O-Moduls an den Eingang eines M-Moduls;

[0093] [Fig. 10](#) eine schematische Darstellung der Serienschaltung des Ausgangs eines M-Moduls an den Eingang eines O-Moduls;

[0094] [Fig. 11](#) eine schematische Darstellung der Serienschaltung des Ausgangs eines M-Moduls an den Eingang eines M-Moduls;

[0095] [Fig. 12](#) eine schematische Darstellung der Parallelschaltung von zwei O-Modulen bei Verschaltung der Ausgänge der parallelen Module jeweils mit einem Eingang eines O- oder P-Moduls;

[0096] [Fig. 13](#) eine schematische Darstellung der nicht zulässigen Parallelschaltung von zwei P- oder M-Modulen bei Verschaltung der Ausgänge der parallelen Module jeweils mit einem Eingang eines O- oder P-Moduls;

[0097] [Fig. 14](#) eine schematische Darstellung der Parallelschaltung von zwei O-, P- oder M-Modulen bei Verschaltung der Ausgänge der parallelen Module jeweils mit einem Eingang eines M-Moduls;

[0098] [Fig. 15](#) eine schematische Darstellung einer Verschachtelung von Hardwaremodulen;

[0099] [Fig. 16](#) eine schematische Darstellung eines Beispiels für eine Verschaltung von Hardwaremodulen (Topologie);

[0100] [Fig. 17](#) eine schematische Darstellung eines Moduls mit einem oder mehreren Input-Links;

[0101] [Fig. 18](#) eine schematische Darstellung eines Moduls mit einem oder mehreren Output-Links;

[0102] [Fig. 19](#) eine schematische Darstellung eines Moduls mit einem Input-Link und einem Output-Link;

[0103] [Fig. 20](#) eine schematische Darstellung eines Moduls mit mehreren Input-Links und einem Output-Link;

[0104] [Fig. 21](#) eine schematische Darstellung eines Moduls mit einem Input-Link und mehreren Output-Links;

[0105] [Fig. 22](#) eine schematische Darstellung eines Moduls mit mehreren Input-Links und mehreren Output-Links;

[0106] [Fig. 23](#) eine schematische Darstellung der Schichten der Implementierung eines Moduls;

[0107] [Fig. 24](#) eine graphische Darstellung (Screen shot) einer beispielhaften Ausführung einer Topologie, und

[0108] [Fig. 25](#) eine schematische Darstellung des Analysegraphen zur Topologie aus 24.

Links

[0109] Die [Fig. 1](#) zeigt schematisch die Datenstruktur eines Input-Links **100** bzw. eines Output-Links **102**. Ein Link **100**, **102** ist eine Datenstruktur, die aus Datensignalen (DATA) **104**, Kontrollsignalen (CTRL) **106**, einem Inhibit-Signal (INH) **108** sowie einem Taktsignal (CLK) **110** besteht.

[0110] Die Richtung des Signalfusses **112** ist bei einem Input-Link **100** dadurch charakterisiert, dass alle Daten- und Kontrollsignale **104**, **106** sowie das Taktsignal **110** Eingangssignale sind, während das Inhibit-Signal **108** ein Ausgangssignal ist. Demgegenüber ist die Richtung des Signalfusses **114** bei einem Output-Link **102** dadurch charakterisiert, dass alle Daten- und Kontrollsignale **104**, **106** sowie das Takt-

signal **110** Ausgangssignale sind, während das Inhibit-Signal **108** ein Eingangssignal ist.

[0111] Die Datensignale **104** transportieren die Nutzdaten. Sie können beliebig aufgebaut und strukturiert sein. Insbesondere sind dabei auch verschachtelte Strukturen zulässig. Die Kontrollsignale **106** übermitteln Zusatzinformationen zeitgleich zu den Datensignalen **104**. Aus den Kontrollsignalen **106** muss sich die Gültigkeit der Nutzdaten erkennen lassen. Dazu kann ein bestimmtes einzelnes Signal oder auch eine Kombination mehrerer Signale genutzt werden.

O-Modul

[0112] In [Fig. 2](#) wird die Verarbeitung der in der Datenstruktur eines Input-Links **100** enthaltenen verschiedenen Signale durch ein O-Modul **200** mit einem Input-Link **202** gezeigt.

[0113] Der Datenfluss der am Input-Link **202** des O-Moduls **200** anliegenden Kontrollsignale **106** wird durch das O-Modul **200** nicht beeinflusst. Der Nutzdatenfluss wird jedoch verändert. Dabei erfahren die Nutzdaten **104** beim Durchlaufen des Moduls **200** eine genau definierte Verzögerung (Latency). Diese entspricht in der Regel der Anzahl der Registerstufen, die die Nutzdaten auf den Datenwegen **204**, **205** über die Funktions-Logik **208** vom Input-Link **202** zum Output-Link **210** durchlaufen. Das O-Modul **200** kann nach Stoppen des Eingangsdatenflusses **104** noch maximal so viele Daten liefern, wie durch die Output-Latency bestimmt wird.

[0114] [Fig. 3](#) zeigt ergänzend die Besonderheiten eines O-Moduls mit mehreren Input-Links **302** und **304**. Bei dieser Ausführung des O-Moduls **300** müssen die Daten gleichphasig anliegen. Es wird davon ausgegangen, dass die CTRL-Signale **106** einschließlich Data-Valid aus derselben Quelle stammen. Daraus resultiert, dass die automatisch erzeugte CTRL-Logik **206** nur aus einem beliebig gewählten der Input-Links **302** oder **303**, beispielsweise **302**, gespeist werden muss. Das Inhibit-Signal wird über die Datenwege **310**, **312** identisch auf die Input-Links **302**, **304** geleitet.

P-Modul

[0115] [Fig. 4](#) zeigt die Verarbeitung der in der Datenstruktur eines Input-Links **100** enthaltenen verschiedenen Signale durch ein P-Modul **400** mit einem Input-Link **402** gezeigt.

[0116] Der Datenfluss der am Input-Link **402** des P-Moduls **400** anliegenden Kontrollsignale **106** kann durch das P-Modul **400** beeinflusst werden, jedoch nur indem die absolute Datenrate reduziert wird. Die Datenwege **404**, **406** sowohl der Kontrolldaten **106**

als auch der Nutzdaten **104** verlaufen daher beim P-Modul über die Funktions-Logik **408**.

Verschaltung

[0117] Die Anzahl und die zeitliche Lage der Kontrollsignale **106** einschließlich Data-Valid kann sich damit verändern. Das hat zur Folge, dass das P-Modul **400** insbesondere weniger Datenworte liefern kann als es empfängt. Das Inhibit-Signal **108** wird in der Funktions-Logik **408** verarbeitet. Der Datenweg **410** des Inhibit-Signals verläuft direkt vom Output-Link **412** zum Input-Link **402** des P-Moduls, woraus resultiert, dass das Inhibit-Signal **108** in der Funktions-Logik **408** des P-Moduls **400** nicht genutzt wird.

[0118] Bei einem in [Fig. 5](#) dargestellten P-Modul **500** mit mehreren Eingängen wird angenommen, dass die an den Input-Links **502** und **504** anliegenden Kontrollsignale aus einer Quelle stammen. Die Funktions-Logik **505** wird daher nur aus einem beliebig gewählten der Input-Links **502** oder **504**, beispielsweise **502**, über den Datenweg **506**, gespeist. Das Inhibit-Signal **108** wird identisch über die Datenwege **510**, **512** auf die Input-Links **502**, **504** geleitet. Bei dieser Ausführung des P-Moduls **500** müssen die Daten gleichphasig anliegen. Die Nutzdaten werden über die Datenwege **514**, **516** zur Funktions-Logik und weiter über den Datenweg **518** zum Output-Link **520** geführt, während die Kontrollsignale **106** über den Datenweg **522** zum Output-Link geführt werden.

M-Modul

[0119] Ein in [Fig. 6](#) dargestelltes M-Modul **600** kann den Kontrollfluss massiv beeinflussen, ja sogar vollständig anhalten. Vom Input-Link **602** des Moduls **600** verlaufen die Datenwege **604**, **606** der Nutz- und Kontrolldaten **104** bzw. **106** über die Funktions-Logik **608** zum Output-Link **610** des M-Moduls **600**. Der Datenweg **611** des Inhibit-Signals verläuft vom Output-Link **610** über die Funktions-Logik **608** zum Input-Link **602**. Das M-Modul **600** kann nach dem Stoppen des Datenflusses maximal eine solche Anzahl Datenworte aufnehmen, wie sie durch die Größe des am Input-Link des Moduls vorhanden Puffers (Buffer-Size) bestimmt wird.

[0120] Bei einem M-Modul **700** entsprechend [Fig. 7](#) mit mehreren Input-Links **702**, **704** wird davon ausgegangen, dass die Kontrollsignale aus unabhängigen Quellen stammen können. Daher werden die Kontrollsignale **106** von allen Input-Links **702**, **704** über die Datenwege **706** zur Funktions-Logik **708** und weiter zum Output-Link **710** des M-Moduls **700** geführt. Ebenso werden die Inhibit-Signale **108** über die Datenwege **712** vom Output-Link **710** zu den Input-Links **702**, **704** geführt. Die Funktions-Logik **708** realisiert die korrekte Bearbeitung und evtl. Zwischenspeicherung versetzt eingehender Daten.

[0121] [Fig. 8](#) zeigt eine Serienschaltung von zwei O-Modulen **200**. Der Ausgang **206** des ersten O-Moduls **200** ist über den Datenweg **800** mit dem Eingang **202** des zweiten O-Moduls **200** verschaltet. Die Output-Latency der gesamten Schaltungsanordnung ergibt sich aus der Summe der Output-Latencies der einzelnen Module. Die Gesamtanordnung kann nach aktivem Inhibit-Signal noch beliebig viele Daten aufnehmen.

[0122] [Fig. 9](#) zeigt eine Serienschaltung eines O-Moduls **200** mit einem M-Modul **600**. Dabei ist der Ausgang **210** des O-Moduls **200** über den Datenweg **900** mit dem Eingang **602** des M-Moduls verschaltet. Die Puffergröße am Input-Link **602** des M-Moduls **600** muss bei dieser Verschaltung mindestens so groß sein wie die Output-Latency des O-Moduls. Andernfalls muss ein Puffer eingefügt werden, der die überzähligen Datenworte zwischenspeichern kann. Dieses wird automatisch durchgeführt. Die Gesamtanordnung kann nach aktivem Inhibit-Signal nur noch so viele Daten aufnehmen, wie durch Buffer-Size am Input-Link **602** und einen eventuell eingefügten Puffer spezifiziert ist. Die Output-Latency dieser Gesamtanordnung entspricht der Output-Latency des M-Moduls **600**.

[0123] [Fig. 10](#) zeigt die Serienschaltung eines M-Moduls **600** mit einem O-Modul **200**. Dabei ist der Ausgang **610** des M-Moduls **600** über den Datenweg **1000** mit dem Eingang **202** des O-Moduls verschaltet.

[0124] Die Gesamtanordnung kann nach aktivem Inhibit-Signal nur noch so viele Daten aufnehmen, wie durch die Buffer-Size am Eingang **602** der Gesamtanordnung spezifiziert ist. Die Output-Latency der Gesamtanordnung ergibt sich aus der Summe der Output-Latencies der einzelnen Module.

[0125] In [Fig. 11](#) wird die Serienschaltung eines ersten M-Moduls **600** mit einem zweiten M-Modul **600** gezeigt. Dabei ist der Ausgang **610** des ersten M-Moduls **600** über den Datenweg **1100** mit dem Eingang **602** des zweiten M-Moduls verschaltet. Die Buffer-Size am Input-Link **602** des zweiten M-Moduls **600** muss dabei mindestens so groß sein wie die Output-Latency am Output-Link **610** des ersten M-Moduls. Die Gesamtanordnung kann nach aktivem Inhibit-Signal nur noch so viele Daten aufnehmen, wie durch die Buffer-Size am Input-Link **602** des ersten M-Moduls spezifiziert ist. Die Output-Latency dieser Gesamtanordnung entspricht der Output-Latency des zweiten M-Moduls **600**.

[0126] [Fig. 12](#) zeigt die Parallelschaltung von zwei O-Modulen. Dabei wird ein erster O-Modul **1200** mit seinem Ausgang **1202** über den Datenweg **1204** mit

einem ersten Eingang **1206** eines zweiten O- oder P-Moduls **1208** verschalten. Ein zweiter O-Modul **1210** wird parallel dazu mit seinem Ausgang **1212** über den Datenweg **1214** mit einem zweiten Eingang **1216** des Moduls **1208** verschalten. Die Eingänge **1218** und **1220** der beiden parallelen O-Module **1200** und **1210** werden über die Datenwege **1222**, **1224**, **1226** gespeist. Bei einem O- oder P-Modul müssen die CTRL-Signale aller Input-Links synchron sein. Daher gelten bei dieser Anordnung die engsten Anforderungen. Die Output-Latency am Ausgang **1228** der Gesamtanordnung ergibt sich aus der Summe der maximalen Output-Latencies der Parallelschaltung und der maximalen Output-Latency des in Serie geschalteten O- oder P-Moduls. Die Gesamtanordnung kann nach aktivem Inhibit-Signal noch beliebig viele Daten aufnehmen. Hinter dem Modul mit der kleineren Output-Latency muß zum Ausgleich ein Register **1230** mit so vielen Stufen eingefügt werden, wie die Differenz zur maximalen Output-Latency beträgt. Im dargestellten Beispiel sind dies drei Registerstufen.

[0127] Die in [Fig. 13](#) dargestellte Anordnung zeigt die Verschaltung der Ausgänge **1300**, **1302** von zwei parallel angeordneten P- und/oder M-Modulen **1304**, **1306** mit jeweils einem Eingang **1308**, **1310** eines O- oder P-Moduls **1312** über die Datenwege **1314**, **1316**. Die parallel geschalteten Module werden aus derselben Datenquelle über den Datenweg **1317** gespeist. Eine solche Anordnung ist nicht zulässig, da die P- bzw. M-Module **1304**, **1306** über keine fest definierte Verzögerung verfügen, so dass keine Gleichphasigkeit der Daten an den Ausgängen **1300**, **1302** der beiden parallel geschalteten P- oder M-Module **1304**, **1306** bzw. an den Eingängen **1308** und **1310** des nachgelagerten O/P-Moduls erreicht werden kann. Diese Schaltung ist deshalb nicht zulässig, weil die beiden parallel geschalteten P- oder M-Module **1304**, **1306** aus der gleichen Datenquelle gespeist wird. In diesem Falle kann es zu einem Deadlock kommen.

[0128] Sind die Datenquellen der parallel geschalteten Module unabhängig voneinander, so ist diese Zusammenführung in einem Sonderfall erlaubt. Dazu wird ein Synchronisierungsmodul benötigt, das beide Datenströme gleichphasig macht.

[0129] Bei der in [Fig. 14](#) dargestellten Anordnung können die Kontrollsignale aller Input-Links an den Eingängen **1400**, **1402** des M-Moduls **1404** unabhängig voneinander sein. Hier sind die Parallelschaltungen aller Kombinationen von O- und/oder P- und/oder M-Modulen **1406**, **1408** möglich, deren Ausgänge **1410**, **1412** über die Datenwege **1414**, **1416** an den Eingängen **1400**, **1402** des M-Moduls **1404** zusammengeführt werden. Wie oben beschrieben wird bei dieser Variante der parallelen Verschaltung von Modulen kein Latency-Ausgleich der einzel-

nen Pfade vorgenommen. Die Datenflusskontrolle erfolgt hier ausschließlich anhand der CTRL- und INH-Signale.

[0130] [Fig. 15](#) zeigt schematisch ein Beispiel für die Verschachtelung von Modulanordnungen. Das Modul **1500** besitzt einen Eingang **1502** und zwei Ausgänge **1504**, **1506**. Das Modul **1500** enthält eine verschachtelte Anordnung von zwei parallel verschalteten Modulen **1508**, **1510**, deren jeweiliger Ausgang **1512**, **1514** mit den Eingängen **1516**, **1518** eines dritten Moduls **1520** verschaltet sind. Bei dieser Anordnung enthält das Modul **1508** wiederum eine Parallelschaltung von zwei Modulen **1522**, **1524**, deren Ausgänge **1526**, **1528** mit den Eingängen **1530**, **1532** eines dritten Moduls **1534** zusammengeschaltet sind.

[0131] Ein Beispiel einer Topologie von Modulen wird schematisch in [Fig. 16](#) dargestellt.

[0132] Hierbei wird eine Topologie mit einer Mehrzahl von O-, P- und M-Modulen gezeigt. Der Datenfluss verläuft vom M-Modul **1604** über die Datenwege **1610**, **1612**, **1614**, **1616**, **1618**, **1620** und **1622** zum M-Modul **1605**. Dabei sind in der Reihenfolge M-Modul **1604**, P-Modul **1602**, O-Modul **1600**, P-Modul **1601**, O-Modul **1607**, O-Modul **1609** und M-Modul **1605** über diese Datenwege seriell miteinander verschaltet. Zwischen P-Modul **1601** und O-Modul **1607** verzweigt sich der Datenweg und verläuft über Datenweg **1618** zu einem zweiten Eingang des O-Moduls **1609**. Die CTRL-Signale verlaufen vom M-Modul **1604** über den Datenweg **1624** zum P-Modul **1602**, vom P-Modul **1602** über den Datenweg **1626** zum P-Modul **1601**, von diesem über den Datenweg **1606** zum M-Modul **1605**. Ein Inhibit-Signal verläuft über den Datenweg **1608** vom M-Modul **1605** zum M-Modul **1604**. Vom M-Modul **1605** verläuft der Datenfluss über die Datenwege **1628**, **1630**, **1632**, **1634** und **1636** zum M-Modul **1603**. Dabei sind in der Reihenfolge M-Modul **1605**, O-Modul **1638**, O-Modul **1640**, O-Modul **1642**, P-Modul **1644** und M-Modul **1603** über diese Datenwege seriell miteinander verschaltet. Zwischen dem M-Modul **1605** und dem O-Modul **1638** erfolgt über den Datenweg **1648** eine Verzweigung des Datenflusses zum O-Modul **1646**. Eine weitere Verzweigung erfolgt zwischen den O-Modulen **1646** und **1647** über den Datenweg **1652** zum O-Modul **1640**. Vom O-Modul **1646** verläuft der Datenfluss über den Datenweg **1650** zum O-Modul **1647** und weiter über den Datenweg **1652** zum P-Modul **1644**. Vom P-Modul verläuft der Datenfluss über den Datenweg **1636** zum M-Modul **1603**. Die CTRL-Signale verlaufen vom M-Modul **1605** zum P-Modul **1644** sowie vom P-Modul **1644** zum M-Modul **1603**. Das Inhibit-Signal wird über den Datenweg **1658** vom M-Modul **1603** zum M-Modul **1605** übertragen.

[0133] Für die Relation von Links zu Modulen wer-

den die verschiedenen möglichen Konfigurationen in den [Fig. 17](#) bis [Fig. 22](#) schematisch gezeigt.

[0134] [Fig. 17](#) zeigt ein Modul **1700**, welches nur einen oder mehrere Input-Links **1702**, **1704** hat. Hierbei handelt es sich um Module, die nur Daten aufnehmen können. Das kann beispielsweise ein DMA-Kanal (direct memory access) sein. Das Modul stellt eine Datensenke dar.

[0135] Ein Modul **1800** mit nur einem oder mit mehreren Output-Links **1802**, **1804** wird in [Fig. 18](#) gezeigt. Beispielsweise kann das eine Kamera sein, also eine Datenquelle.

[0136] In [Fig. 19](#) wird ein Modul **1900** mit einem Input-Link **1902** und einem Output-Link **1904** gezeigt. Diese Module empfangen in der Regel Daten von einer Quelle, bearbeiten diese und geben sie an den Ausgang weiter, wobei sie mit unterschiedlich großer Verzögerung **1906** an den Ausgang **1904** weiter gegeben werden. Es sind aber auch Module möglich, bei denen der Input- und der Output-Link unabhängig voneinander sind. Das können beispielsweise Module sein, die einen Write- und einen Read-DMA-Kanal realisieren.

[0137] Eine andere Konfiguration ist ein Modul **2000** mit mehreren Input-Links **2002**, **2004** und nur einem Output-Link **2006**. Diese Konfiguration wird schematisch in [Fig. 20](#) gezeigt. Diese Module **2000** empfangen in der Regel Daten von mehreren Pfaden, bearbeiten diese und geben sie mit einer möglicherweise erforderlichen Verzögerung **2008** an den Ausgang weiter. Die Daten können dabei aus derselben Quelle stammen oder völlig unabhängig voneinander sein. Es sind beispielsweise auch Module möglich, bei denen nicht alle Input-Links eine Verbindung zum Output-Link besitzen, sondern unabhängige Datensenken darstellen.

[0138] [Fig. 21](#) zeigt als eine weitere mögliche Konfiguration ein Modul **2100** mit einem Input-Link **2102** und mehreren Output-Links **2104**, **2106**. Bei diesen Modulen **2100** werden in der Regel Daten von einem Eingang **2102** empfangen und über verschiedene interne Datenwege **2108**, **2110** verarbeitet. Die Ergebnisse aller Datenwege werden jeweils über einen eigenen Ausgang ausgegeben. Die Verzögerung **2112**, **2114** auf den Datenwegen **2108**, **2110** und damit die Output-Latency kann für die einzelnen Output-Links **2104**, **2106** unterschiedlich sein. Es sind aber auch Module möglich, bei denen einzelne Output-Links nicht aus Input-Links gespeist werden, sondern unabhängige Datenquellen darstellen.

[0139] [Fig. 22](#) zeigt als letzte Konfigurationsmöglichkeit ein Modul **2200** mit mehreren Input-Links **2202**, **2204** und mehreren Output-Links **2206**, **2208**. Daten werden von mehreren Eingängen empfangen

und über verschiedene interne Datenwege **2210**, **2212** verarbeitet. Die Ergebnisse aller Datenwege werden jeweils über einen eigenen Ausgang ausgegeben. Die internen Datenwege können dabei unabhängig voneinander arbeiten oder auch über Querverbindungen **2214**, **2216** Daten austauschen. Die Verzögerung **2218** auf den Datenwegen und damit die Output-Latency kann für die einzelnen Output-Links **2206**, **2208** unterschiedlich sein. Es sind auch einzelne unabhängige Input- und Output-Links möglich.

[0140] Wie in [Fig. 23](#) gezeigt, erfolgt die Übersetzung in das Layout der Hardware vorzugsweise in einer Reihe von Schritten, ausgehend von der Topologie.

[0141] Im graphischen Editor **2300** wird zunächst die Topologie **2302** erstellt (siehe [Fig. 24](#)). Das erfolgt auf graphischer Basis oder textuell.

[0142] Anschließend wird innerhalb der Kodierungsschicht **2304** die Topologie überprüft, und auf der Grundlage der im Programm hinterlegten Verschaltungsregeln werden evtl. erforderliche Puffer, Verzögerungen und Ähnlichem ergänzt. Das Ergebnis wird in einem sog. Analysegraph **2306** (siehe [Fig. 25](#)) wiedergegeben.

[0143] Anschließend erfolgt in der VHDL-Schicht **2307** eine Übersetzung der Topologie und des Analysegraphen einschließlich aller Funktions-Logiken, Module, Puffer, Verzögerungen, etc. in eine Hardware-Beschreibungssprache, beispielsweise VHDL. Es ist auch möglich, die Hardwarebeschreibung in anderen Hardwarebeschreibungssprachen zu erzeugen.

[0144] Daraufhin wird in der Syntheseschicht **2308** mit Hilfe eines geeigneten Compilers die VHDL-Beschreibung in eine Edif-Netzliste **2310** (Edif = Electronic Design Interchange Format) übersetzt.

[0145] Ist eine Edif-Netzliste **2310** erzeugt, dann werden in der Schicht der Hardwareimplementierung **2312** bei FPGAs **2314** die Place-Route Werkzeuge der FPGA-Hersteller verwendet, um daraus eine Konfiguration zu erzeugen, die den gewünschten Algorithmus ausführt. Diese Werkzeuge erstellen das fertige FPGA-Programm, welches in den FPGA geladen werden kann (je nach FPGA-Gattung auch beliebig oft), um die Schaltung auszuführen.

[0146] Im Falle von ASICs sind es die Place, Route und Mapwerkzeuge der ASIC-Fabs. Diese werden zum Erstellen von Masken für den Herstellungsprozess verwendet. Der ASIC **2316** wird entsprechend der in der Synthese berechneten Masken gefertigt und führt den spezifizierten Algorithmus aus.

[0147] [Fig. 24](#) zeigt eine beispielhafte Darstellung eines Screenshots für eine Topologie aus Anwendersicht. Ein Kamera-Datenfluss vom Kamera-Modul **2400** wird über den Datenweg **2402** zu Modul **2404** übertragen und dort gepuffert. Über den Datenweg **2406** wird der Datenfluss weiter zu Modul **2408** übertragen und dort geteilt. Über die Datenwege **2410**, **2414**, **2416**, **2419** wird der eine Datenfluss der Reihe nach über die Filter-Module **2411** und **2415** zu einem Shift-Operator **2418** geführt, hier verkleinert und dem DMA-Kanal **2420** eines PCs zugeführt. Der zweite Datenfluss wird unverändert über den Datenweg **2412** direkt an einen zweiten DMA-Kanal **2422** weitergegeben. Die Darstellungen der Module auf der Grafik zeigen auch die graphischen Symbole der Inputlinks z. B. **2424** sowie der Output-Links z. B. **2426**, **2428**, von welchen aus die verschiedenen Datenwege verlaufen.

[0148] [Fig. 25](#) zeigt den Analysegraphen zu der in [Fig. 24](#) dargestellten Topologie.

[0149] Von einem M-Modul **2500** für die Kamera mit einer Output-Latency **2502** des Wertes -1 werden Daten zu einem nächsten M-Modul **2504** mit einer Buffer-Size **2506** von -1 . Das bedeutet, dass der Datenstrom zwischen den beiden Modulen nicht angehalten wird. Das M-Modul **2504** ist der Puffer **2404**.

[0150] Von dem M-Modul **2504** mit einer Output-Latency **2508** des Wertes 2 werden die Daten zu einem nächsten O-Modul **2510**, dem Splitter **2408**, mit einer Buffer-Size **2512** des Wertes 0 übertragen. In diesem O-Modul **2510** wird der Datenfluss geteilt und über die Output-Links **2514**, **2516** zu den M-Modulen **2522** (DMA-Kanal **2422** des PCs) bzw. **2526** (Filter-Modul **2411**) übertragen. Auf der Basis der Verschaltungsregeln wurde in diesem Falle eine erforderliche zusätzliche Puffertiefe von 2 errechnet. Im Ergebnis dessen wurden die Puffer **2518** bzw. **2520** automatisch zwischen dem O-Modul **2510** und den M-Modulen **2528** bzw. **2524** eingefügt. Die Module **2518** bzw. **2520** stellen FIFO-Elemente dar mit einer jeweiligen Puffertiefe des Wertes 2 .

[0151] Nach der Verzweigung verläuft der eine Datenfluss weiter vom M-Modul **2526** (Filter **2411**) mit einer Output-Latency **2530** des Wertes 2 zum O-Modul **2532** (Filter **2415**) mit einer Buffer-Size **2534** von 0 und einer Output-Latency **2536** von 0 . Weiter verläuft der Datenfluss zum O-Modul **2538** (Shift-Operator **2418**) mit einer Buffer-Size **2540** von 0 und einer Output-Latency **2542** von 0 und von dort zum M-Modul **2546** (DMA-Kanal **2420** des PCs) mit einer Buffer-Size **2548** von 0 .

[0152] Auf der Basis der Verschaltungsregeln wurde in diesem Falle ein zusätzlicher Puffer **2544** mit einer Puffertiefe von 2 zwischen den Modulen **2538** und **2546** eingefügt. Der zweite Datenfluss erfolgt

vom O-Modul **2510** (Splitter **2408**) über den automatisch eingefügten Puffer **2518** direkt zum M-Modul **2522** (DMA-Kanal **2422** des PCs) mit der Buffer-Size **2524** von 0 .

Bezugszeichenliste

100	Input-Link
102	Output-Link
104	Datensignal (DATA)
106	Kontrollsignal (CTRL)
108	Inhibit-Signal (INH)
110	Taktsignal (CLK)
200	O-Modul
202	Input-Link
204	Datenweg
205	Datenweg
206	CTRL-Logik
208	Funktions-Logik
210	Output-Link
300	O-Modul mit mehreren Input-Links
302	Input-Link
304	Input-Link
306	Kontrollsignal (CTRL)
310	Datenweg
312	Datenweg
400	P-Modul
402	Input-Link
404	Datenweg
406	Datenweg
408	Funktions-Logik
410	Datenweg
412	Output-Link
500	P-Modul mit mehreren Input-Links
502	Input-Link
504	Input-Link
505	Funktions-Logik
506	Datenweg
510	Datenweg
512	Datenweg
514	Datenweg
516	Datenweg
518	Datenweg
520	Output-Link
522	Datenweg
600	M-Modul
602	Input-Link
604	Datenweg
606	Datenweg
608	Funktions-Logik
610	Output-Link
700	M-Modul mit mehreren Input-Links
702	Input-Link
704	Input-Link
706	Datenweg
708	Funktions-Logik
710	Output-Link
712	Datenweg
800	Datenweg
900	Datenweg
1000	Datenweg

1100	Datenweg	1608	Signalfluss INH
1200	O-Modul	1610	Signalfluss DATA
1202	Ausgang eines Moduls	1612	Signalfluss DATA
1204	Datenweg	1614	Signalfluss DATA
1206	Eingang eines Moduls	1616	Signalfluss DATA
1208	Eingang eines Moduls	1618	Signalfluss DATA
1210	O-Modul	1620	Signalfluss DATA
1212	Ausgang eines Moduls	1622	Signalfluss DATA
1214	Datenweg	1624	Signalfluss CTRL
1216	Eingang eines Moduls	1626	Signalfluss CTRL
1218	Eingang eines Moduls	1628	Signalfluss DATA
1220	Eingang eines Moduls	1630	Signalfluss DATA
1222	Datenweg	1632	Signalfluss DATA
1224	Datenweg	1634	Signalfluss DATA
1226	Datenweg	1636	Signalfluss DATA
1228	Ausgang eines Moduls	1638	O-Modul
1300	Ausgang eines Moduls	1640	O-Modul
1302	Ausgang eines Moduls	1642	O-Modul
1304	M-Modul	1644	P-Modul
1306	M-Modul	1646	O-Modul
1308	Eingang eines Moduls	1648	O-Modul
1310	Eingang eines Moduls	1650	Signalfluss DATA
1312	P-Modul	1652	Signalfluss DATA
1314	Datenweg	1654	Signalfluss CTRL
1316	Datenweg	1656	Signalfluss CTRL
1317	Datenweg	1658	Signalfluss INH
1400	Eingang eines Moduls	1700	Modul
1402	Eingang eines Moduls	1702	Input-Link
1404	M-Modul	1704	Input-Link
1406	O- oder P-Modul	1800	Modul
1408	O- oder P-Modul	1802	Output-Link
1410	Ausgang eines Moduls	1804	Output-Link
1412	Ausgang eines Moduls	1900	Modul
1414	Datenweg	1902	Input-Link
1416	Datenweg	1904	Output-Link
1500	Modul	1906	Verzögerung (Latency)
1502	Eingang eines Moduls	2000	Modul
1504	Ausgang eines Moduls	2002	Input-Link
1506	Ausgang eines Moduls	2004	Input-Link
1508	Modul	2006	Output-Link
1510	Modul	2008	Verzögerung (Latency)
1512	Ausgang eines Moduls	2100	Modul
1514	Ausgang eines Moduls	2102	Input-Link
1516	Eingang eines Moduls	2104	Output-Link
1518	Eingang eines Moduls	2106	Output-Link
1520	Modul	2108	Datenweg
1522	Modul	2110	Datenweg
1524	Modul	2112	Verzögerung (Latency)
1526	Ausgang eines Moduls	2114	Verzögerung (Latency)
1528	Ausgang eines Moduls	2200	Modul
1530	Eingang eines Moduls	2202	Input-Link
1532	Eingang eines Moduls	2204	Input-Link
1534	Modul	2206	Output-Link
1600	O-Modul	2208	Output-Link
1602	P-Modul	2210	Datenweg
1603	M-Modul	2212	Datenweg
1604	M-Modul	2214	Datenweg (Querverbindung)
1605	M-Modul	2216	Datenweg (Querverbindung)
1606	Signalfluss CTRL	2218	Verzögerung (Latency)
1607	O-Modul	2300	Beschreibungsschicht

2302	Topologie
2304	Kodierungsschicht
2306	Analysegraph
2307	VHDL-Schicht
2308	Syntheseschicht
2310	Edif-Netzliste
2312	Hardware-Implementierungsschicht
2314	FPGA
2316	ASIC
2400	Modul
2402	Datenweg
2404	Modul
2406	Datenweg
2408	Modul
2410	Datenweg
2411	Modul
2412	Datenweg
2414	Datenweg
2415	Modul
2416	Datenweg
2418	Shift-Operator
2419	Datenweg
2420	DMA-Kanal eines PC
2422	DMA-Kanal
2424	Input-Link
2426	Output-Link
2428	Output-Link
2500	Modul
2502	Output-Latency
2504	Modul
2506	Buffer-Size
2508	Output-Latency
2510	Modul
2512	Buffer-Size
2514	Output-Link
2516	Output-Link
2518	Modul
2520	Modul
2522	Modul
2524	Buffer-Size
2526	Modul
2528	Modul
2530	Output-Latency
2532	Modul
2534	Buffer-Size
2536	Output-Latency
2538	Modul
2540	Buffer-Size
2542	Output-Latency
2544	Modul
2546	Modul
2548	Buffer-Size

Patentansprüche

1. Verfahren zum Programmieren von Hardware für die Verarbeitung von Nutzdaten mit folgenden Schritten:

a) eine Mehrzahl von Modulen wird zur Verfügung gestellt, wobei jedes Modul mindestens eine Funktion

zum Verarbeiten der Nutzdaten ausführen kann;
b) Schnittstellen zum Verschalten der Module werden vorgegeben;
c) eine Verschaltung von Modulen (Topologie), die einer für die Verarbeitung der Nutzdaten geeignete Abfolge von Funktionen entspricht, wird vorgegeben; und
d) aus der Topologie wird die Hardware-Programmierung erzeugt.

2. Verfahren nach dem vorhergehenden Anspruch, dadurch gekennzeichnet, dass die Module nach vorgegebenen Eigenschaften in eine Mehrzahl von Modultypen klassifiziert werden; und dass Verschaltungsregeln abhängig von den Modultypen definiert werden.

3. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass zur Darstellung von Modulen für die Verarbeitung von Nutzdaten graphische Symbole verwendet werden; und dass die Topologie auf graphisch orientierter Basis nach dem Baukastenprinzip aus Modulen erstellt wird.

4. Verfahren nach dem vorhergehenden Anspruch, dadurch gekennzeichnet, dass an einer Schnittstelle eines Moduls eine Datenstruktur (Link) existiert, die aus Nutzdatensignalen (DATA) und einem Taktsignal (CLK) und/oder Kontrollsignalen (CTRL) und/oder einem Inhibit-Signal (INH) besteht,
– wobei Input-Links für den Empfang von Daten und Output-Links für die Ausgabe von Daten definiert sind, und
– wobei die Schnittstellen die Verbindung eines Ausgangs eines ersten Moduls mit einem Eingang eines zweiten Moduls ermöglichen;
dass für jeden Ausgang eines Moduls eine Datenstruktur vom Typ Output-Link vorgesehen wird; und dass für jeden Eingang eines Moduls eine Datenstruktur vom Typ Input-Link vorgesehen wird.

5. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass die Steuerung des Datenflusses über die Inhibit-Signale erfolgt,
– wobei durch ein aktives Inhibit-Signal der Nutzdatenfluss gestoppt werden kann.

6. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass in einem Modul mindestens eine Funktions-Logik die Verarbeitung der Nutzdaten realisiert.

7. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass ausgangsseitig ein Modul durch eine Output-Latency charakterisiert wird, die bestimmt, welche maximale Anzahl von gültigen Datenworten ein Link noch senden kann, nachdem der Eingangsdatenfluss gestoppt wurde.

8. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass ein Modultyp mit den folgenden Eigenschaften definiert wird (O-Modul)

- a) das O-Modul kann die Werte von Nutzdaten ändern, aber keine Nutzdaten löschen;
- b) das O-Modul beeinflusst den Kontrolldatenfluss nicht;
- c) ein Inhibit-Signal hat keinen Einfluss auf das O-Modul;
- d) die Daten erfahren beim Durchlaufen des O-Moduls eine genau definierte Verzögerung (Latency);
- e) bei einem O-Modul mit mehreren Input-Links müssen die Nutzdaten gleichphasig anliegen; und
- f) bei einem O-Modul mit mehreren Input-Links wird das Inhibit-Signal identisch auf alle Input-Links geleitet.

9. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass ein Modultyp mit den folgenden Eigenschaften definiert wird (P-Modul):

- a) das P-Modul kann die Werte von Nutzdaten ändern und Nutzdaten löschen;
- b) das P-Modul kann den Kontrolldatenfluss ändern und löschen;
- c) ein Inhibit-Signal hat keinen Einfluss auf das P-Modul;
- d) die Daten erfahren beim Durchlaufen des P-Moduls eine genau definierte maximale Verzögerung (maximale Latency), wobei die Verzögerung, die einzelne Nutzdaten erfahren, auch kleiner als die maximale Verzögerung sein kann;
- e) bei einem P-Modul mit mehreren Input-Links müssen die Daten gleichphasig anliegen,
 - wobei die Funktions-Logik mit Kontrolldaten aus mindestens einem der Input-Links gespeist wird, und
 - wobei das Inhibit-Signal identisch auf alle Input-Links geleitet wird.

10. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass ein Modultyp mit den folgenden Eigenschaften definiert wird (M-Modul):

- a) das M-Modul kann die Werte von Nutzdaten ändern und Nutzdaten löschen und neue Nutzdaten erzeugen;
- b) das M-Modul kann Kontrolldatenfluss ändern, löschen und erzeugen;

- c) das M-Modul kann ein Inhibit-Signal erzeugen;
- d) die Daten erfahren beim Durchlaufen des M-Moduls eine unbestimmte Verzögerung;
- e) dass M-Module auf ihrer jeweiligen Eingangsseite einen Eingangspuffer mit einer Puffergröße (Buffer-Size) zur Speicherung von Daten besitzen; und
- f) bei einem Modul des Typs "M" mit mehreren Input-Links, können eingehende Nutzdaten sowie Kontrolldaten aus unabhängigen Quellen (Input-Links) stammen.

11. Verfahren nach einem der vorhergehenden Ansprüche,

dadurch gekennzeichnet,

dass die Verschaltung von Modulen unterschiedlicher Modultypen nach den folgenden Regeln erfolgt:

- a) die Serienschaltung eines O- oder P-Moduls mit einem O- oder P-Modul ist uneingeschränkt möglich; und/oder
- b) bei Verschaltung des Ausgangs eines O- oder P-Moduls mit dem Eingang eines M-Moduls (Serienschaltung) muss die Buffer-Size des M-Moduls größer oder gleich sein als die Output-Latency des O- oder P-Moduls; und/oder
- c) die Verschaltung des Ausgangs eines M-Moduls mit dem Eingang eines O- oder P-Moduls (Serienschaltung) ist uneingeschränkt möglich; und/oder
- d) bei Verschaltung des Ausgangs eines ersten M-Moduls mit dem Eingang eines zweiten M-Moduls (Serienschaltung) muss die Buffer-Size des zweiten M-Moduls größer sein als die Output-Latency des ersten M-Moduls; und/oder
- e) bei Parallelschaltung eines ersten und mindestens eines zweiten O-Moduls, deren Ausgänge mit den Eingängen eines dritten O- oder P-Moduls verschaltet werden, muss hinter oder vor dem ersten oder zweiten O-Modul, je nachdem, welches die kleinere Output-Latency hat, ein Verzögerungsregister mit so vielen Stufen eingefügt werden, wie die Differenz der Output-Latencies des ersten und zweiten O-Moduls beträgt, damit an den Eingängen des dritten Moduls eine Synchronisation der Daten gewährleistet bleibt; und/oder
- f) eine Parallelschaltung eines ersten P- oder M-Moduls mit mindestens einem zweiten P- oder M-Modul, deren Ausgänge mit den Eingängen eines dritten O- oder P-Moduls verschaltet werden, ist unzulässig; und/oder
- g) eine Parallelschaltung eines ersten O-, P- oder M-Moduls mit mindestens einem zweiten O-, P- oder M-Modul, deren Ausgänge mit den Eingängen eines dritten M-Moduls verschaltet werden, ist zulässig.

12. Verfahren nach dem vorhergehenden Anspruch,

dadurch gekennzeichnet,

dass überprüft wird, ob die Verschaltungsregeln durch die Topologie eingehalten sind; und dass – sofern möglich – die Topologie derart ergänzt wird,

dass die Verschaltungsregeln erfüllt sind.

13. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass eine Verschachtelung von Modulanordnungen erfolgt,
– wobei ein Modul aus einer Anordnung mehrerer Module aufgebaut ist, und
– wobei beliebig tiefe Strukturen möglich sind.

14. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass eine Mehrzahl von Verschachtelungen und Verschaltungen von Modulen (Bausteine) zum Realisieren jeweils einer vorgegebenen Funktion erstellt werden (Bibliothek).

15. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass eine Topologie auf graphisch orientierter Basis nach dem Baukastenprinzip aus Modulen und/oder Bausteinen aus einer Bibliothek erstellt wird.

16. Graphischer Editor für die Programmierung von Hardware:

- a) mit graphischen Symbolen zur Darstellung von Modulen, wobei jedes Modul eine Funktion zum Verarbeiten von Daten ausführen kann, wenn das Modul in Hardware realisiert wird;
- b) mit mindestens einem graphischen Symbol zur Darstellung von Schnittstellen zum Verschalten der Module;
- c) mit mindesten einem graphischen Symbol zur Darstellung der Verschaltung von Modulen, wodurch eine Verschaltung einer Mehrzahl von Modulen (Topologie) darstellbar ist; und
- d) mit Mitteln zum Übersetzen der graphischen Wiedergabe der Topologie in eine Programmierung der Hardware.

17. Graphischer Editor nach dem vorhergehenden Anspruch, gekennzeichnet durch Mittel zum Überprüfen, ob eine zulässige Verschaltung der Module in der Topologie vorliegt.

18. Computerprogramm, dadurch gekennzeichnet, dass es bei Ablauf auf einer Recheneinheit, einem Mikrocontroller, DSP, FPGA oder Computer oder auf einer Mehrzahl davon in einem Netzwerk das Verfahren nach einem der vorhergehenden Verfahrensansprüche oder den graphischen Editor ausführt.

19. Computerprogramm mit Programmcode-Mitteln, um ein Verfahren gemäß einem der vorhergehenden Verfahrensansprüche durchzuführen oder den graphischen Editor zu realisieren, wenn das Computerprogramm auf einer Recheneinheit, einem Mikrocontroller, DSP, FPGA oder Computer oder auf

einer Mehrzahl davon in einem Netzwerk ausgeführt wird.

20. Computerprogramm mit Programmcode-Mitteln gemäß dem vorhergehenden Anspruch, die auf einem computerlesbaren Datenträger gespeichert sind.

21. Datenträger, auf dem eine Datenstruktur gespeichert ist, die nach einem Laden in einen Arbeits- und/oder Hauptspeicher einer Recheneinheit, eines Mikrocontrollers, DSPs, FPGAs oder Computers oder einer Mehrzahl davon in einem Netzwerk das Verfahren nach einem der vorhergehenden Verfahrensansprüche oder den graphischen Editor ausführt.

22. Computerprogramm-Produkt mit auf einem maschinenlesbaren Träger gespeicherten Programmcode-Mitteln, um alle Schritte gemäß einem der vorhergehenden Verfahrensansprüche durchzuführen oder den graphischen Editor zu realisieren, wenn das Programm auf einer Recheneinheit, einem Mikrocontroller, DSP, FPGA oder Computer oder auf einer Mehrzahl davon in einem Netzwerk ausgeführt wird.

23. Moduliertes Datensignal, welches von einer Recheneinheit, einem Mikrocontroller, DSP, FPGA oder Computer oder von einer Mehrzahl davon in einem Netzwerk ausführbare Instruktionen zum Ausführen eines Verfahrens nach einem der vorhergehenden Verfahrensansprüche oder zur Realisierung des graphischen Editors enthält.

Es folgen 16 Blatt Zeichnungen

Anhängende Zeichnungen

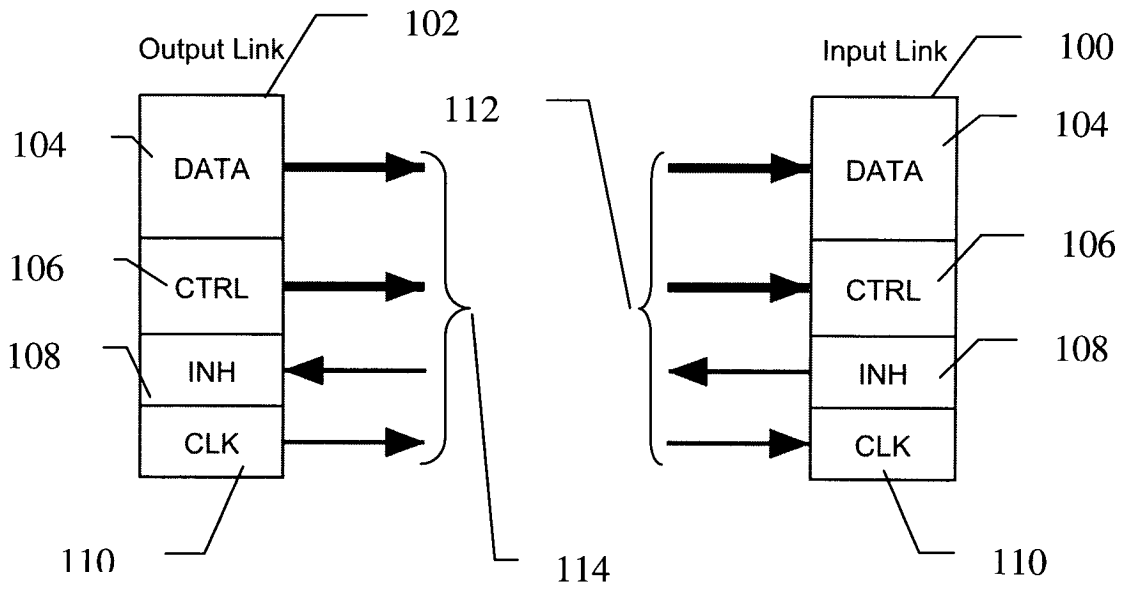


Fig. 1

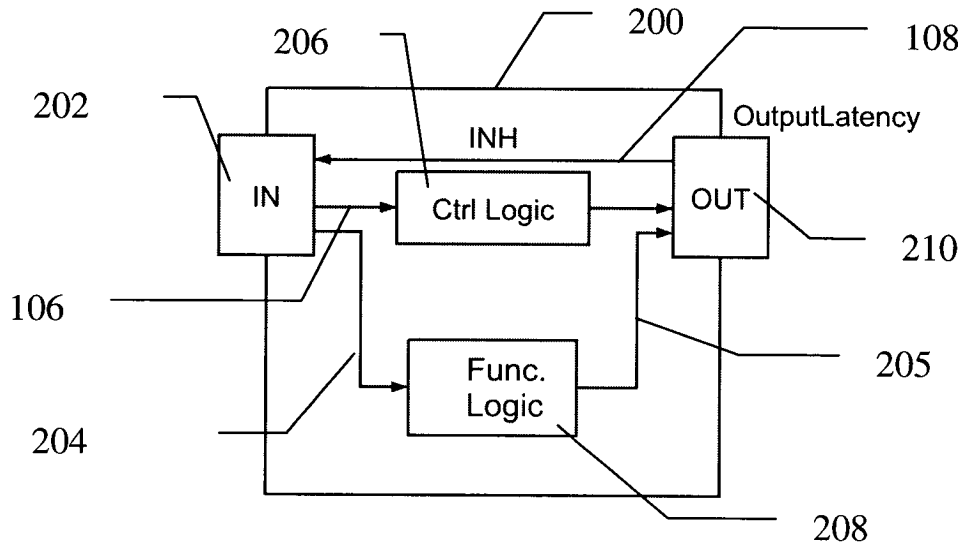


Fig. 2

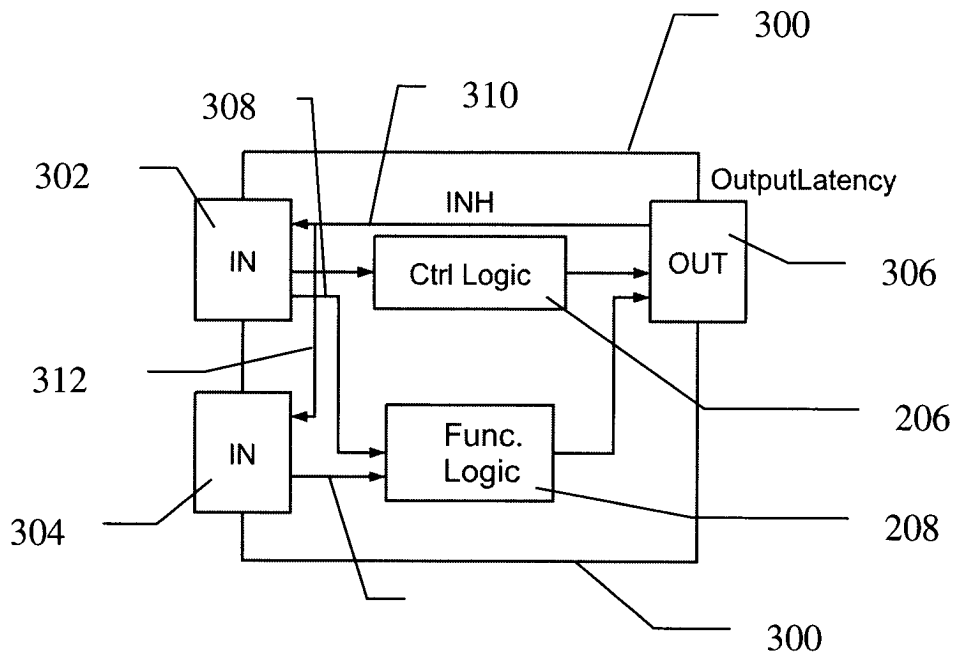


Fig. 3

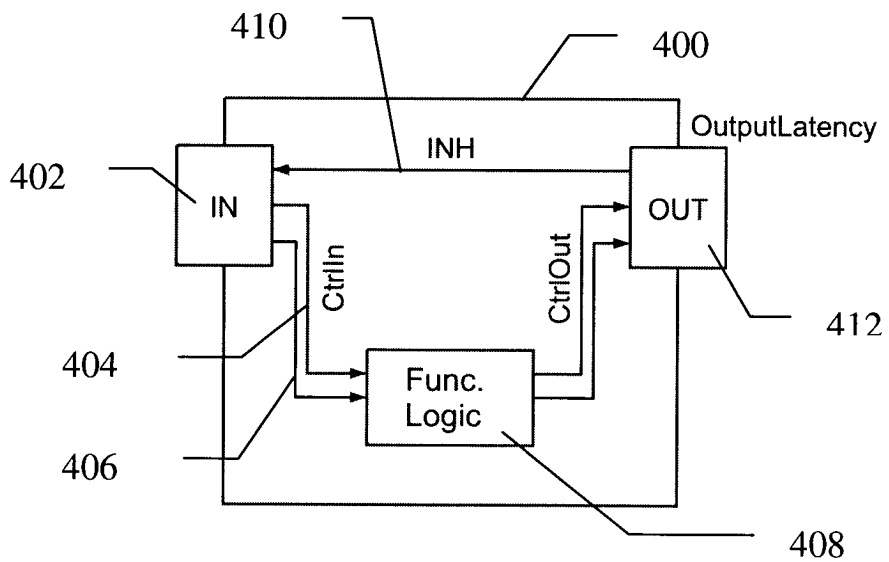


Fig. 4

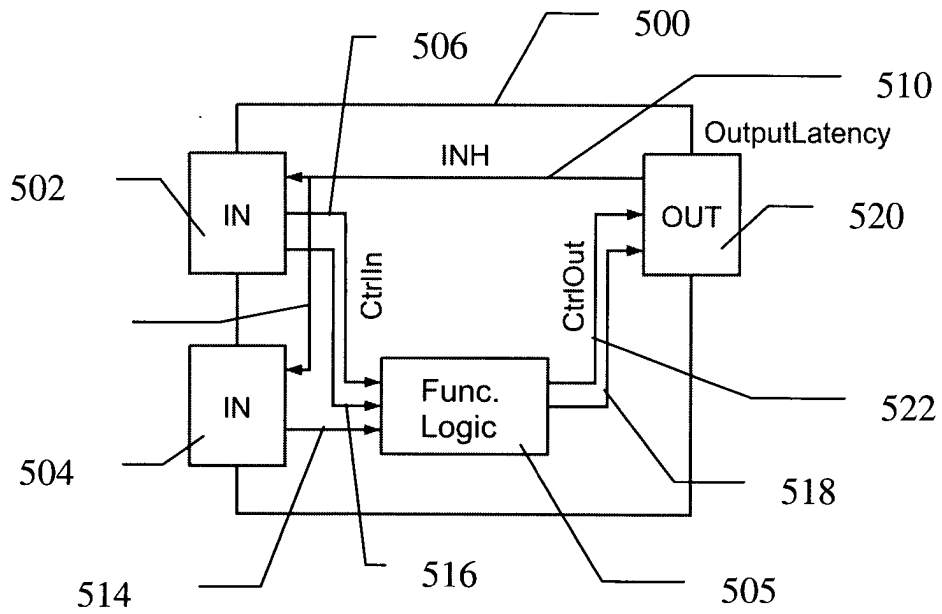


Fig. 5

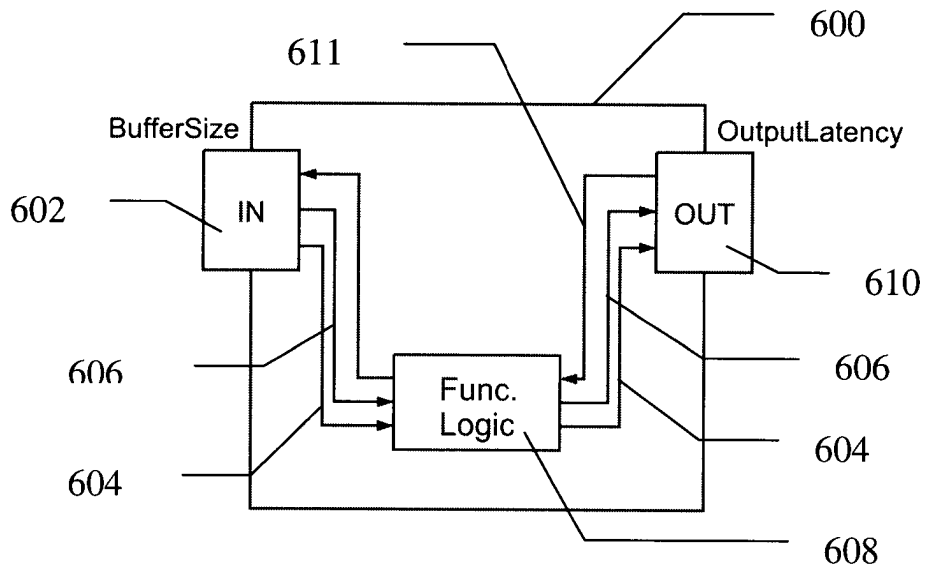


Fig. 6

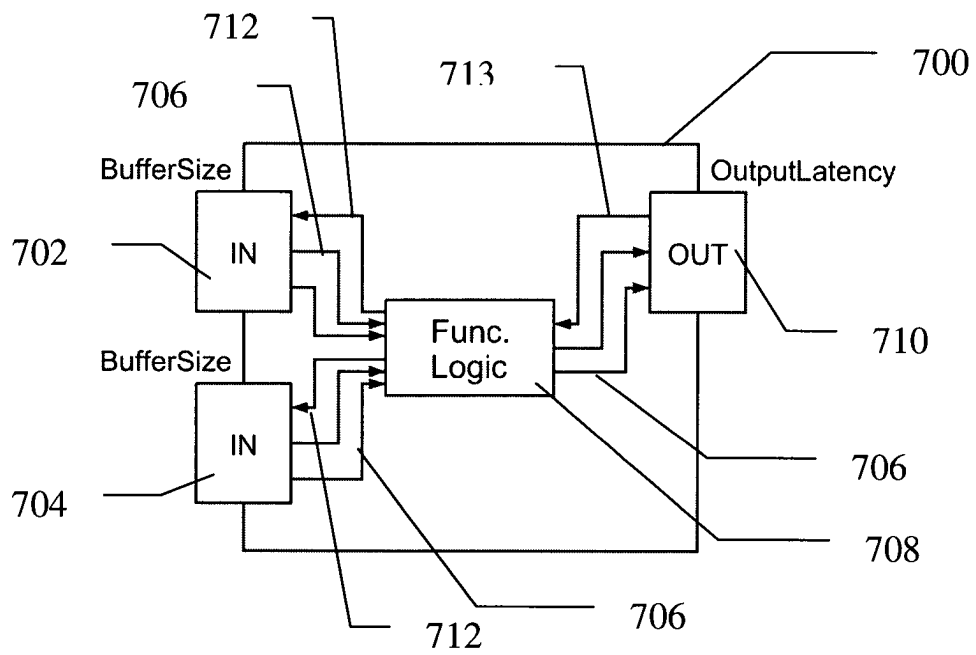


Fig. 7

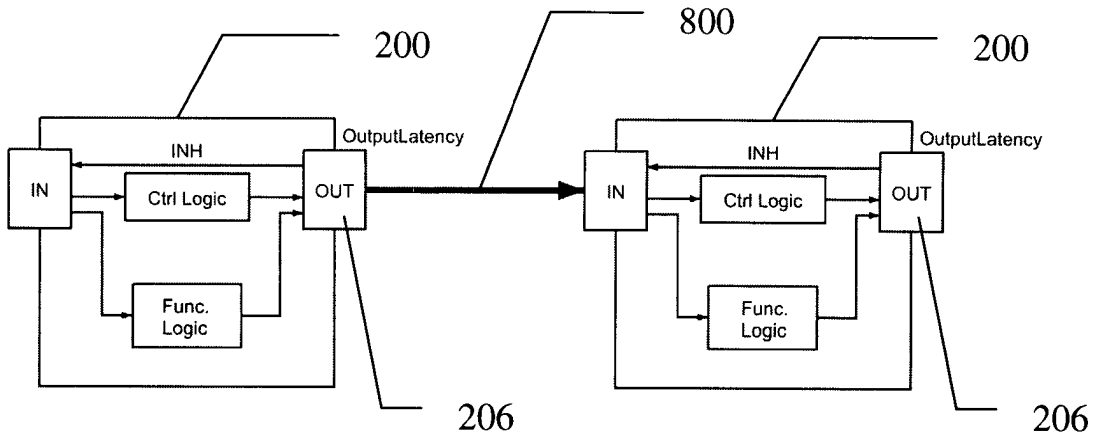


Fig. 8

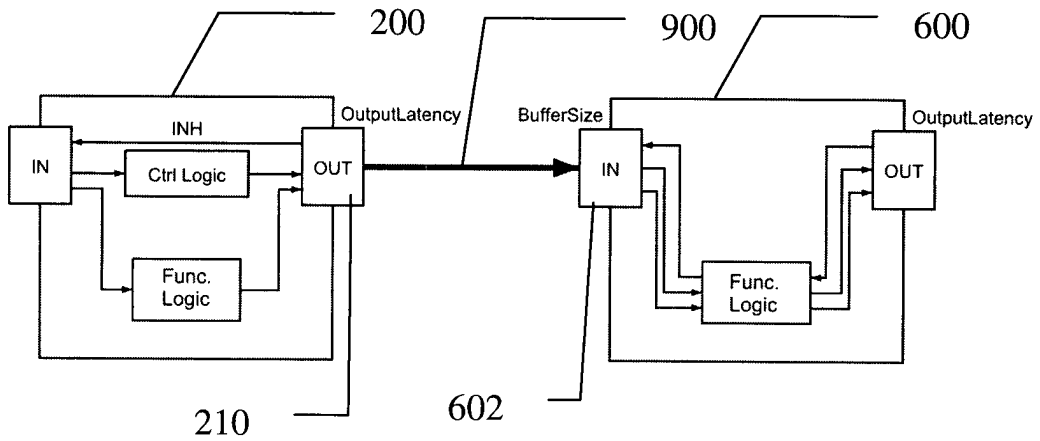


Fig. 9

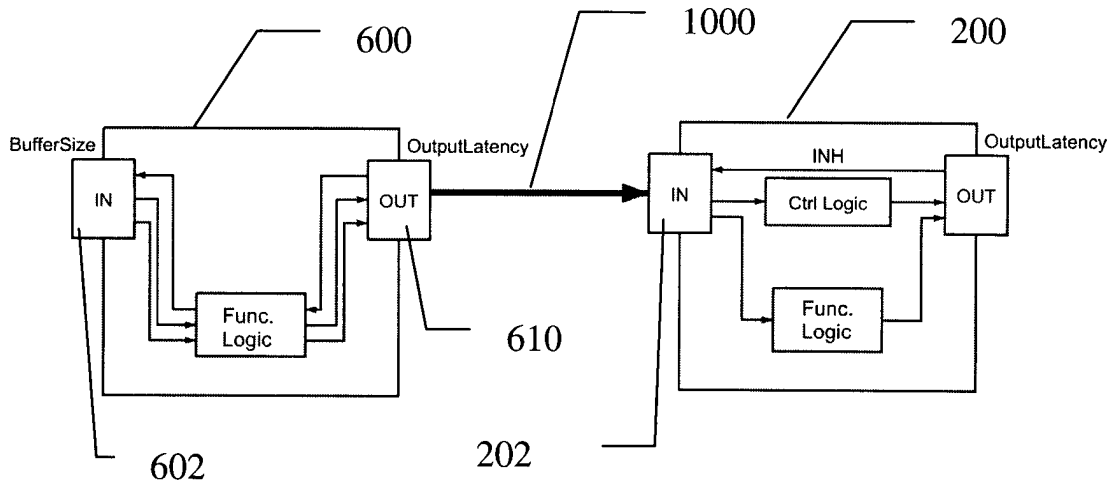


Fig. 10

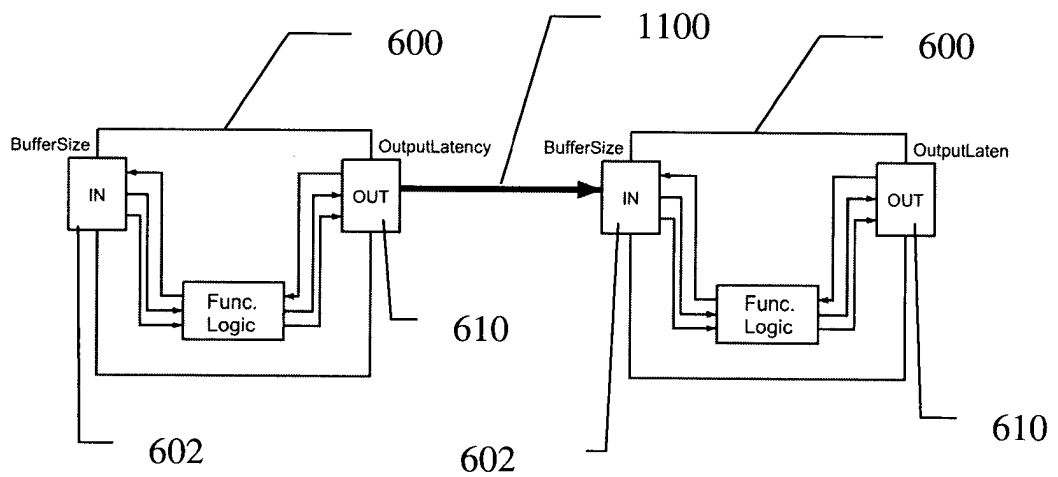


Fig. 11

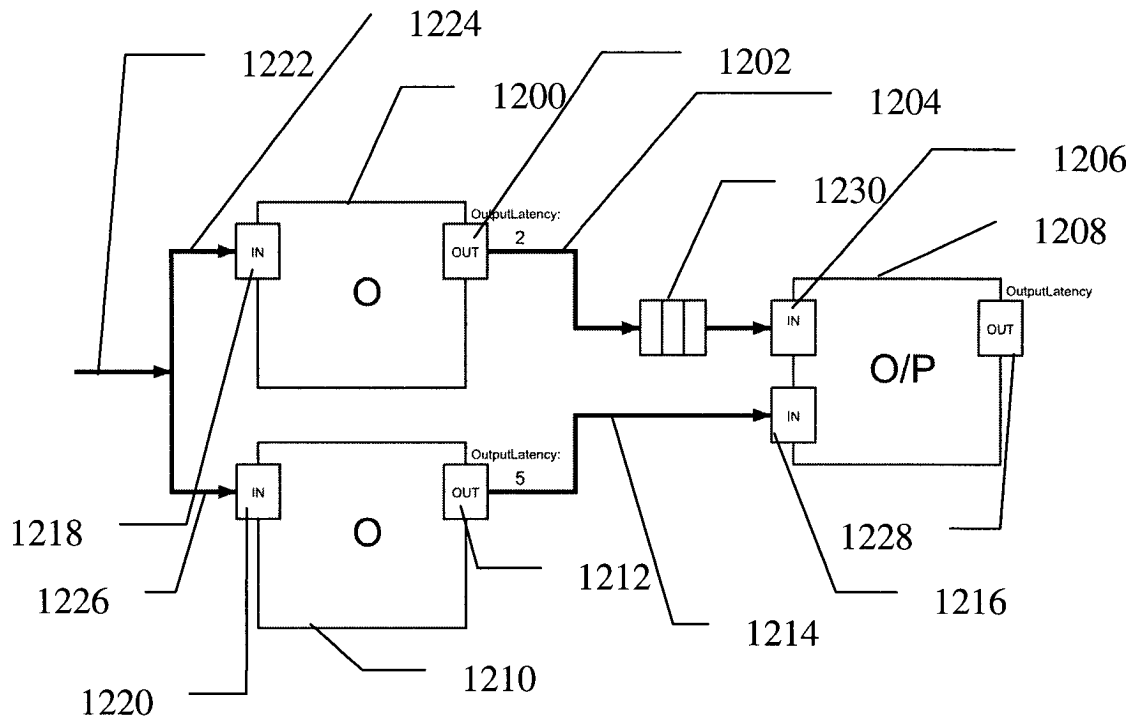


Fig. 12

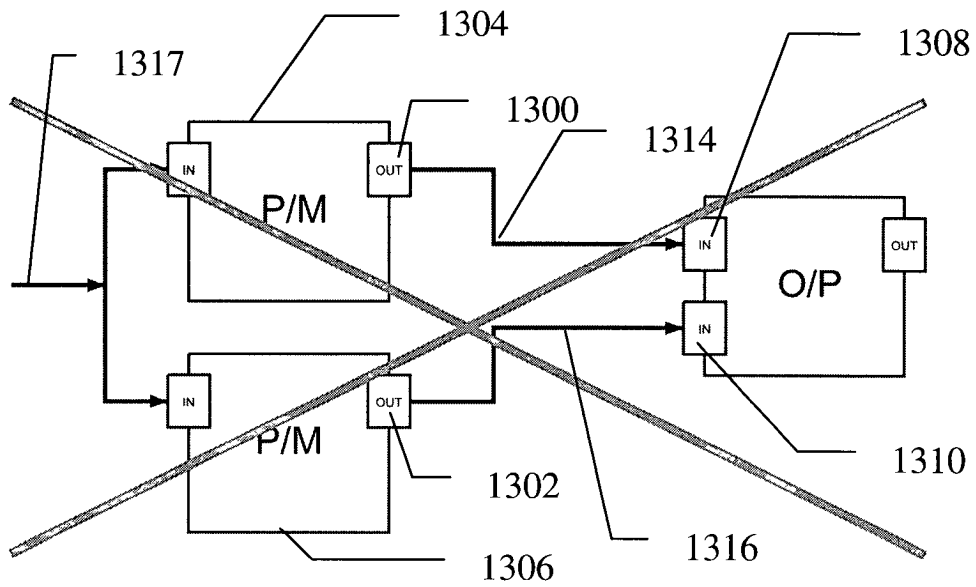


Fig. 13

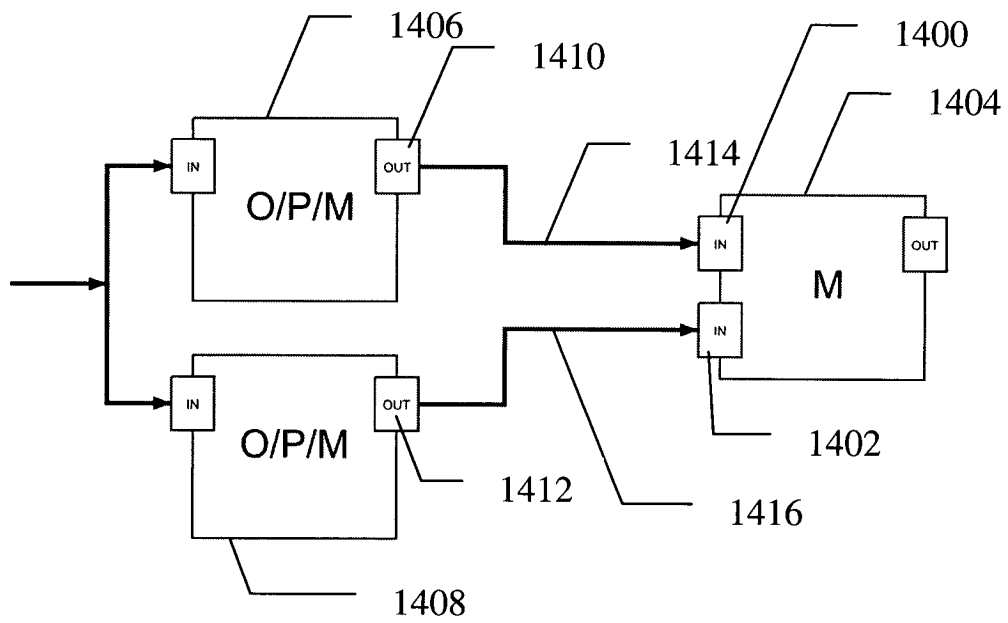


Fig. 14

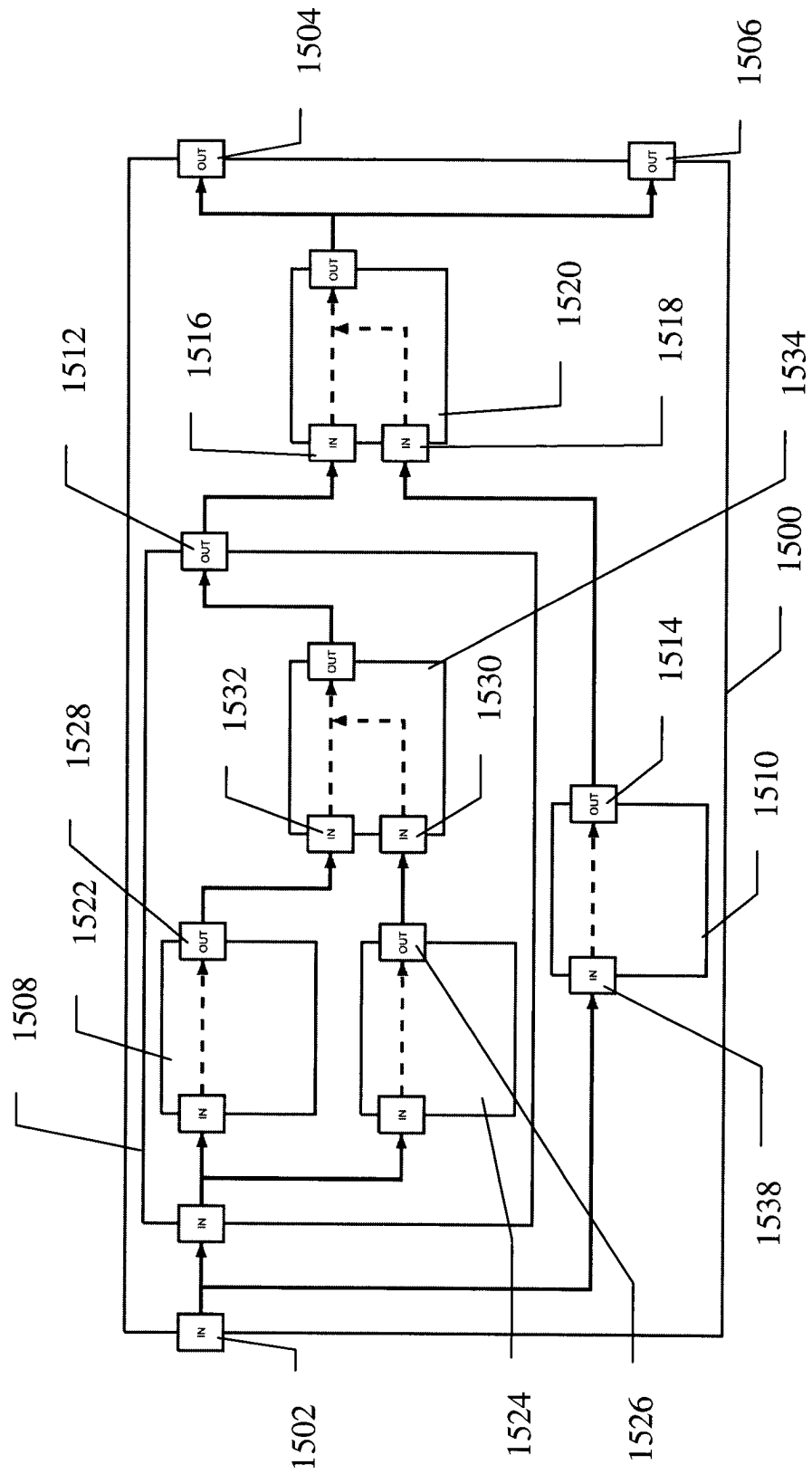


Fig 15

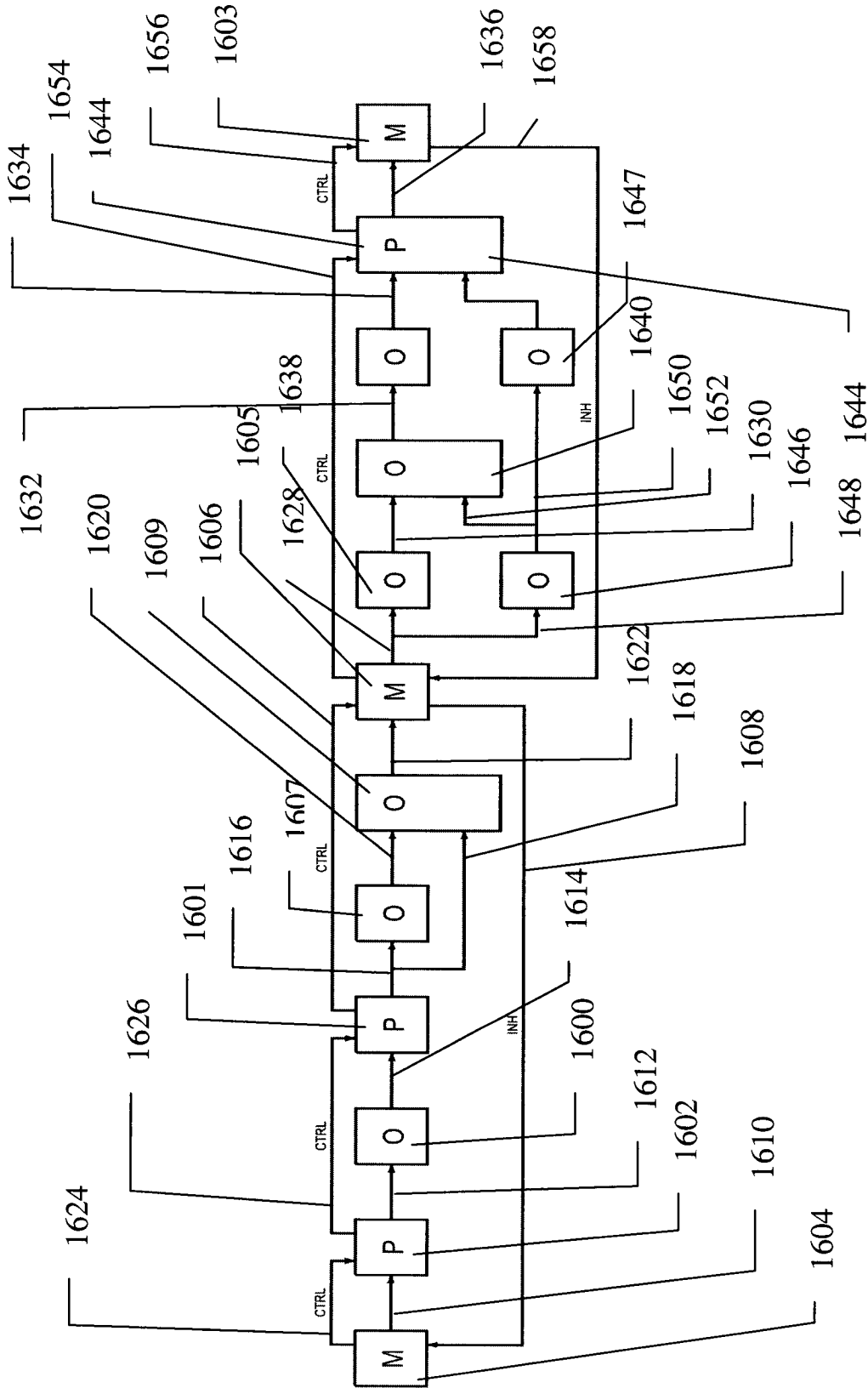


Fig. 16

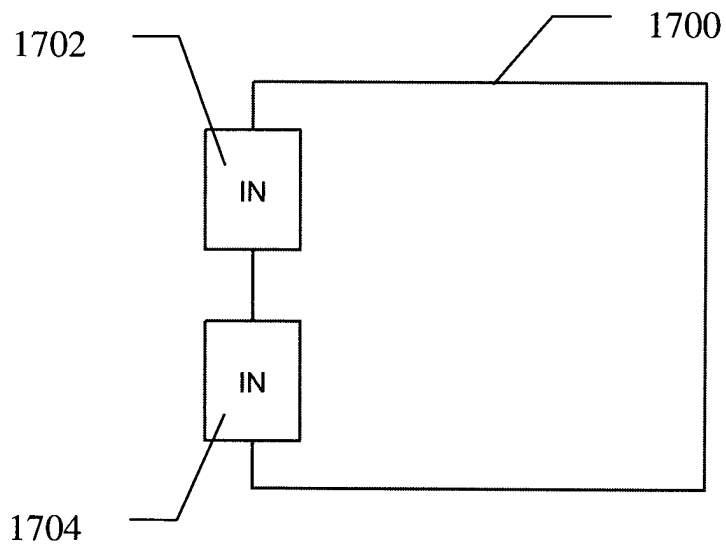


Fig. 17

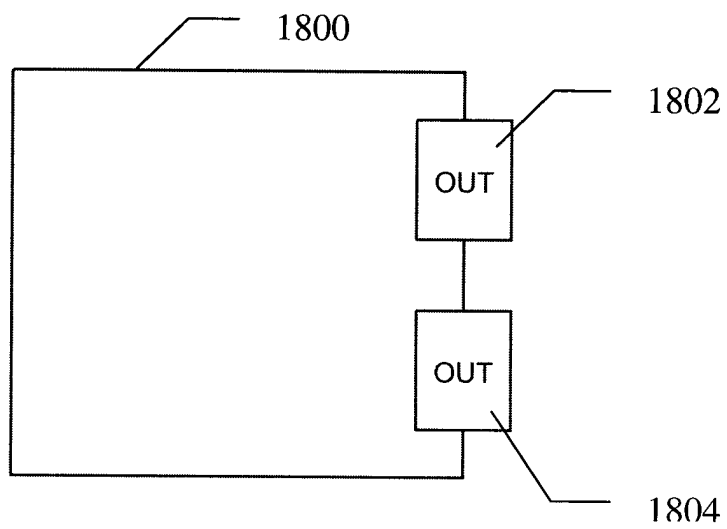


Fig. 18

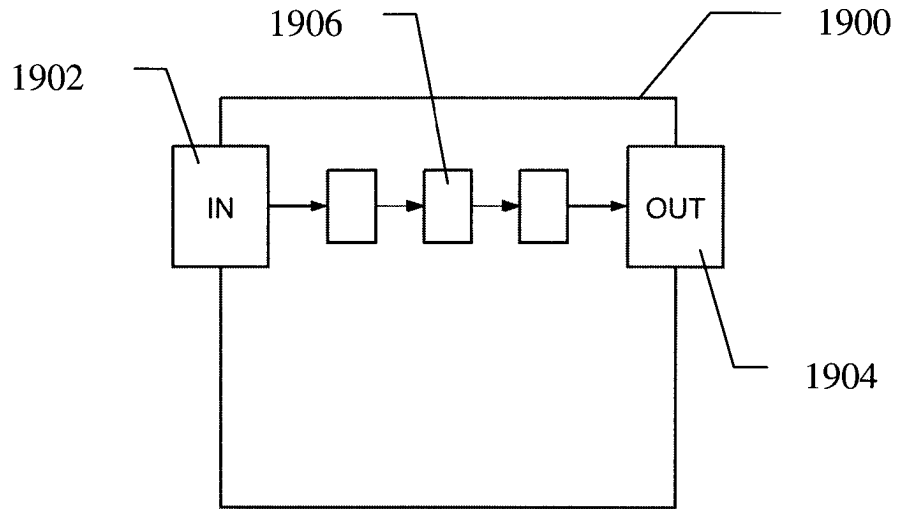


Fig. 19

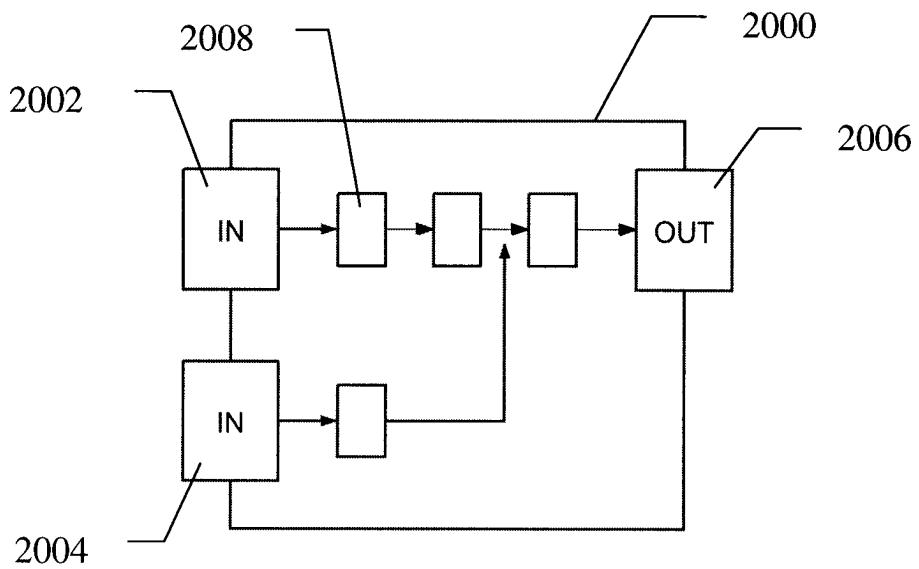


Fig. 20

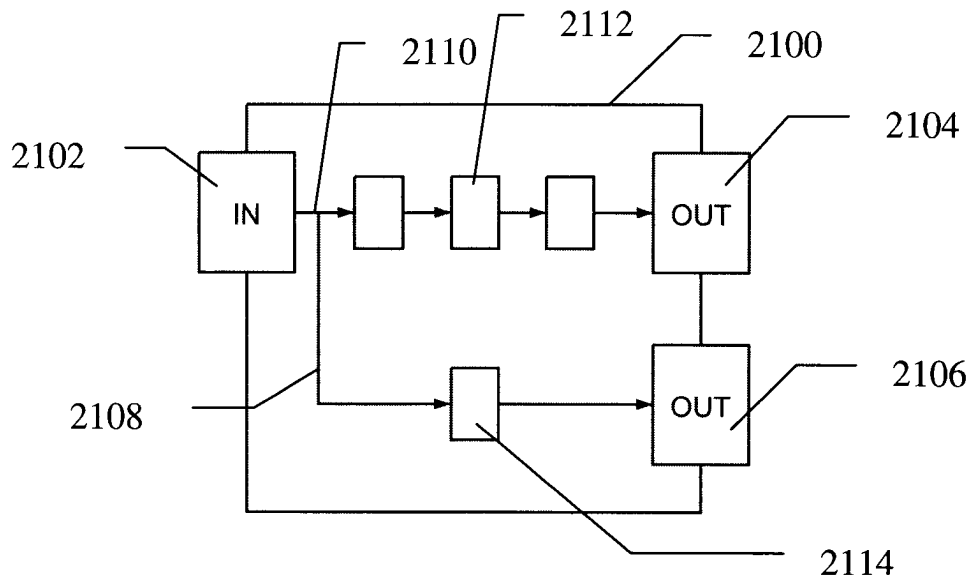


Fig. 21

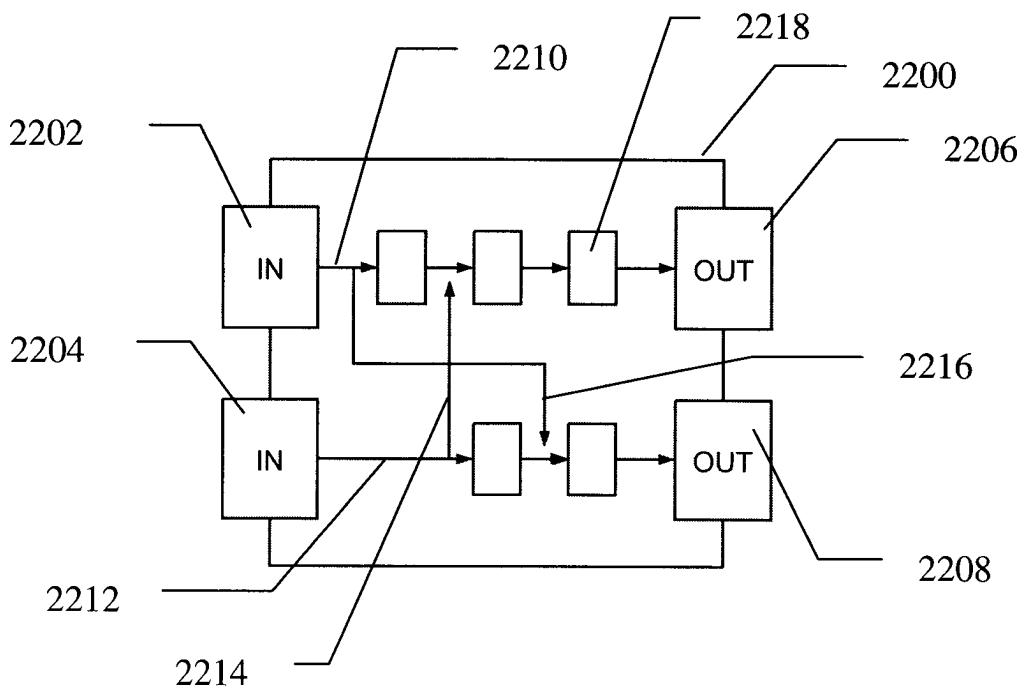


Fig. 22

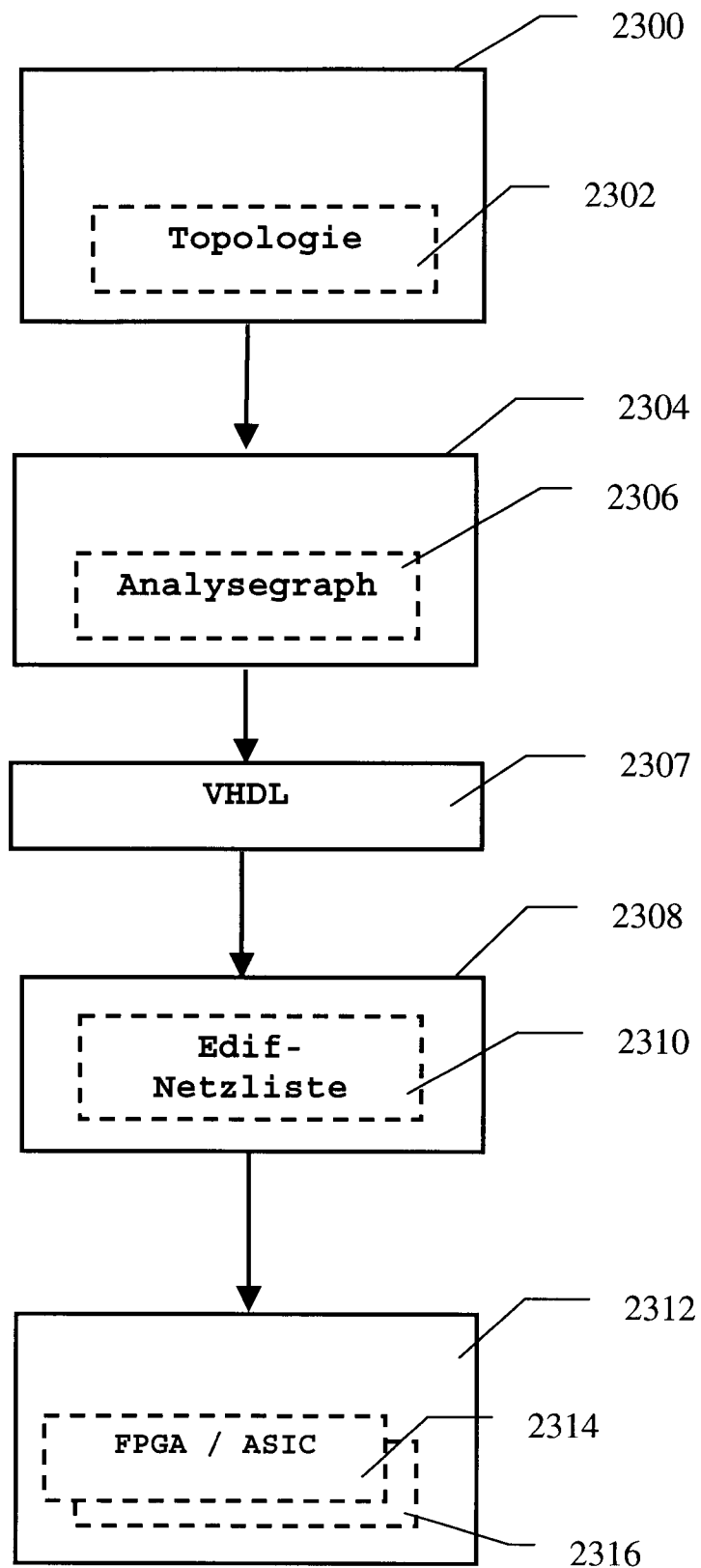


Fig. 23

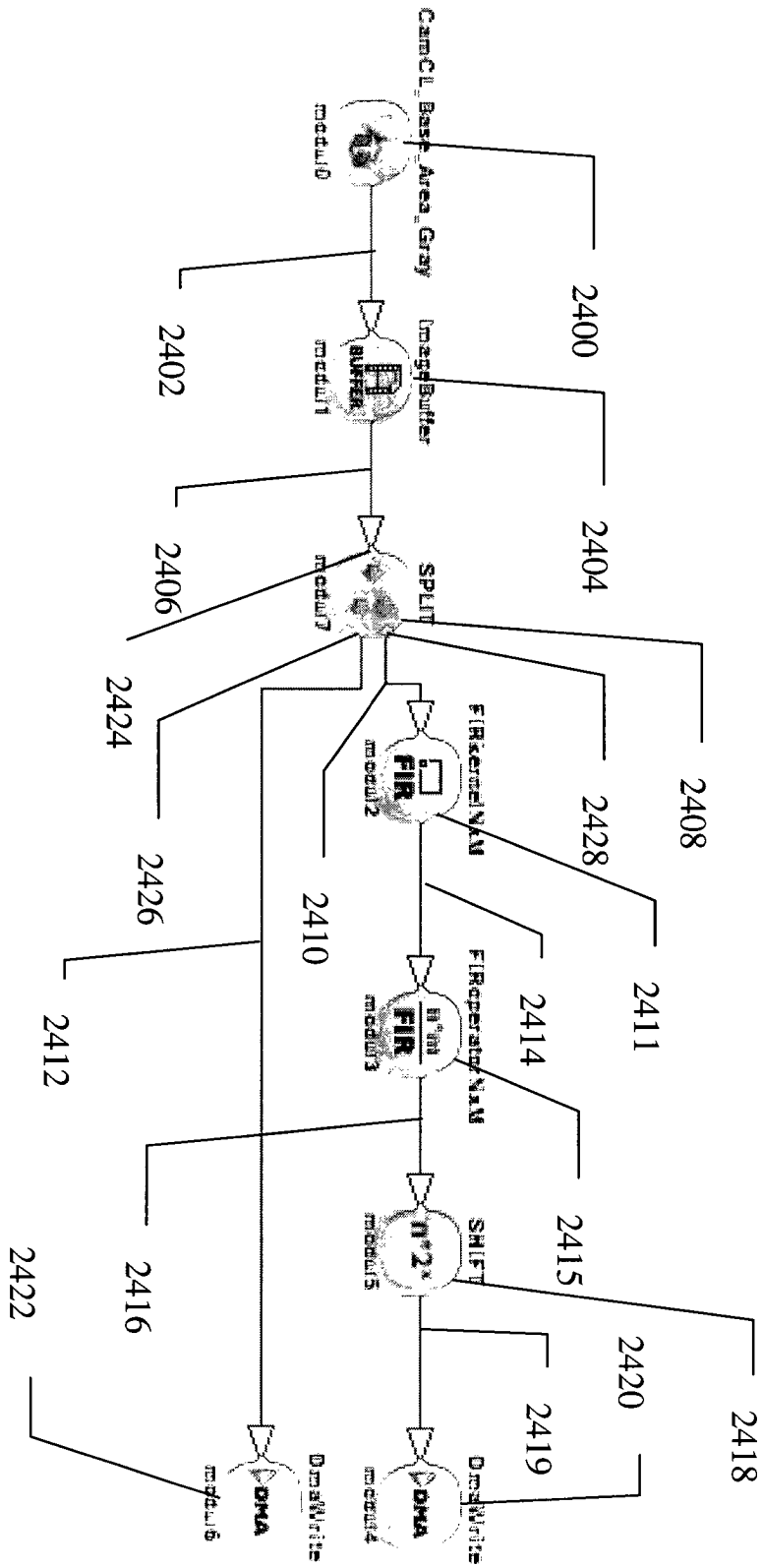


Fig. 24

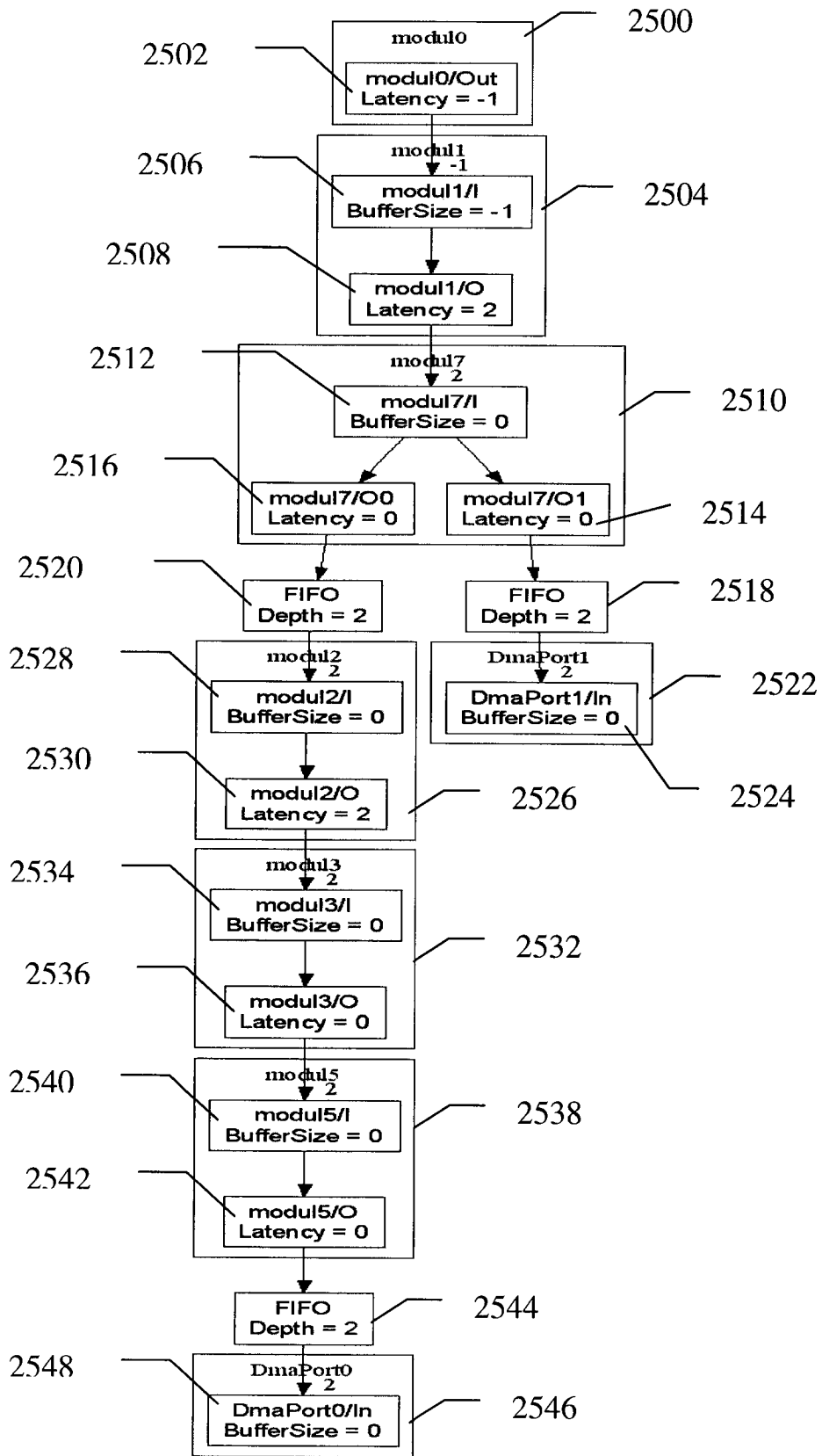


Fig. 25