



(12) **Offenlegungsschrift**

(21) Aktenzeichen: **10 2011 119 004.3**

(22) Anmeldetag: **19.11.2011**

(43) Offenlegungstag: **23.05.2013**

(51) Int Cl.: **G06T 1/20 (2012.01)**

**G06T 15/00 (2012.01)**

**G09G 5/377 (2012.01)**

(71) Anmelder:

**Diehl Aerospace GmbH, 88662, Überlingen, DE**

(72) Erfinder:

**Rettig, Sven, 63329, Egelsbach, DE; Hosemann,  
Thomas, 07429, Sitzendorf, DE**

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

**DE 10 2006 002 907 A1**  
**US 6 980 216 B2**  
**US 2010 / 0 289 963 A1**  
**US 5 276 798 A**

**Gerald R. MORRIS [et.al.]: High-Performance  
FPGA-Based General Reduction Methods, 2005,  
In: Field-Programmable Custom Computing  
Machines (FCCM'05), 323-324, DOI: 10.1109/  
FCCM.2005.42**

**Jan P. SPRINGER [et.al.]: Multi-Frame Rate  
Rendering and Display, In: Virtual Reality Conf.  
2007, 195- 202, DOI: 10.1109/VR.2007.352481**

**K. DANNE, M. PLATZNER: Periodic real-time  
scheduling for FPGA computers, In: Intelligent  
Solutions in Embedded Systems 2005, 117-127,  
DOI: 10.1109/WISES.2005.1438720**

**M. DUTTON, D. KEEZER: The challenges of  
graphics processing in the avionics industry, In:  
Digital Avionics Systems Conference (DASC),  
2010 IEEE/AIAA 29th, Seiten 5.A.1-1 - 5.A.1-9, DOI:  
10.1109/DASC.2010.5655325**

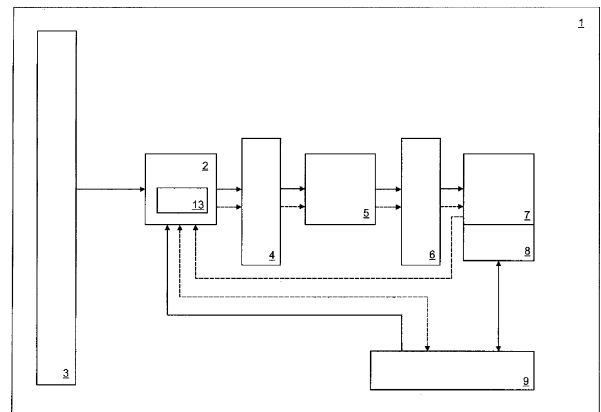
**Mark SNYDER: Designing a safety-certifiable  
OpenGL software GPU; ALT Software —  
December 3, 2008, [online] URL: [http://vita-  
technologies.com/articles/designing-safety-  
certifiable-opengl-software-gpu/](http://vita-technologies.com/articles/designing-safety-certifiable-opengl-software-gpu/) [abgerufen:  
02.07.2012]**

Prüfungsantrag gemäß § 44 PatG ist gestellt.

**Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen**

(54) Bezeichnung: **Grafikverarbeitungsvorrichtung, Anzeigevorrichtung für ein Flugzeugcockpit sowie Verfahren zur Anzeige von Grafikdaten**

(57) Zusammenfassung: Es wird eine Grafikverarbeitungsvorrichtung 1 umfassend eine Kontrolleinrichtung 2 zur Übernahme von Grafikdaten zu ersten Graphikobjekten aus einer ersten Anwendung App\_1 und zur Übernahme von Grafikdaten zu zweiten Graphikobjekten aus einer zweiten Anwendung App\_2; ein Geometriemodul 5 zur Erzeugung der Geometriedaten der Graphikobjekte aus den Grafikdaten; ein Rastermodul 7 zur Rasterung der Graphikobjekte auf Basis der Geometriedaten und zur Erzeugung von Pixeln der Graphikobjekte; wobei die Kontrolleinrichtung 2, das Geometriemodul 5 und das Rastermodul 7 eine Grafikpipeline bilden, welche ausgebildet ist, in einem seriellen Durchgang eines der Graphikobjekte abzuarbeiten, indem aus den Grafikdaten zu dem Graphikobjekt Pixel des Graphikobjekts erzeugt werden, vorgeschlagen, wobei die Grafikpipeline ausgebildet ist, die Bearbeitung eines der ersten Graphikobjekte der ersten Anwendung App\_1 in einem ersten seriellen Durchlauf zu unterbrechen, so dass ein teilbearbeitetes Graphikobjekt vorliegt, mindestens eines der zweiten Graphikobjekte der zweiten Anwendung App\_2 in einem zweiten seriellen Durchlauf zu bearbeiten und nachfolgend die Bearbeitung des teilbearbeiteten Graphikobjekts der ersten Anwendung App\_1 weiterzuführen.



## Beschreibung

**[0001]** Die Erfindung betrifft eine Grafikverarbeitungs- vorrichtung mit den Merkmalen des Oberbegriffs des Anspruchs 1. Die Erfindung betrifft auch eine Anzeigevorrichtung für ein Flugzeugcockpit mit dieser Grafikverarbeitungs- vorrichtung sowie ein Ver- fahren mit dieser Grafikverarbeitungs- vorrichtung.

**[0002]** Grafikkarten steuern üblicherweise die Bild- anzeige von Datenverarbeitungsanlagen. Dabei er- zeugt eine Anwendung auf der Datenverarbeitungs- einrichtung Daten zu einer grafischen Ausgabe und leitet diese an die Grafikkarte weiter. Die Grafikkarte wandelt die Daten so um, dass eine Anzeigeeinrich- tung die gewünschte Ausgabe als Bild wiedergeben kann. Derartige Grafikkarten sind in jedem handels- üblichen PC entweder als Erweiterungskarte oder als Onboard-Chipsatz enthalten.

**[0003]** Es ist ebenfalls von handelsüblichen PCs be- kannt, dass eine Grafikkarte zwei unterschiedliche Anzeigeeinrichtungen, zum Beispiel zwei Bildschir- me, ansteuern kann. In diesen Ausführungen ist es möglich, dass auf dem ersten Bildschirm die grafi- schen Ausgabe einer ersten Anwendung und auf dem zweiten Bildschirm die grafische Ausgabe einer zwei- ten Anwendung dargestellt werden.

**[0004]** Der Erfindung liegt die Aufgabe zugrunde, eine Grafikverarbeitungs- vorrichtung vorzuschlagen, die erhöhten Sicherheitsanforderungen entspricht. Es ist eine weitere Aufgabe der Erfindung, eine An- zeigevorrichtung für ein Flugzeugcockpit sowie ein Verfahren jeweils mit dieser Grafikverarbeitungs- vorrichtung vorzuschlagen.

**[0005]** Diese Aufgaben werden durch eine Grafik- verarbeitungs- vorrichtung mit den Merkmalen des An- spruchs 1, durch eine Anzeigevorrichtung für ein Flugzeugcockpit mit den Merkmalen des Anspruchs 12 sowie durch ein Verfahren mit den Merkmalen des Anspruchs 14 gelöst. Bevorzugte oder vorteil- hafte Ausführungsformen der Erfindungen ergeben sich aus den Unteransprüchen, der nachfolgenden Beschreibung sowie den beigefügten Figuren.

**[0006]** Der Gegenstand der Erfindung ist eine Grafik- verarbeitungs- vorrichtung, insbesondere ausgebildet als Grafikprozessor, welche geeignet und/oder aus- gebildet ist, Grafikdaten von Grafikobjekten in Pixel der Grafikobjekte umzuwandeln.

**[0007]** Die Grafikobjekte können im Allgemeinen als ein Vollbild oder als ein Teilbild ausgebildet sein. Bei bevorzugten Ausführungsformen sind die Grafikob- jekte als eine oder mehrere Grafikprimitiven realisiert. Beispielsweise werden bei der OpenGL-Spezifikation komplexe grafische Objekte durch eine oder mehrere

Grafikprimitiven dargestellt. Eine Grafikprimitive kann z. B. ein Dreieck oder eine Linie sein.

**[0008]** Die Grafikverarbeitungs- vorrichtung ist beson- ders bevorzugt als ein integrierter Schaltkreis und insbesondere als ein FPGA (Feld programmierba- re Logik-Gatter-Anordnung oder Field Programmable Gate Array) realisiert.

**[0009]** Die Grafikverarbeitungs- vorrichtung umfasst eine Kontrolleinrichtung, welche zur Übernahme von Grafikdaten zu ersten Grafikobjekten aus einer ers- ten Anwendung und zur Übernahme von Grafikdaten zu zweiten Grafikobjekten aus einer zweiten Anwen- dung ausgebildet ist. Die Grafikdaten sind insbeson- dere als ein Befehlssatz zur Erzeugung von mindes- tens einer Grafikprimitiven ausgebildet.

**[0010]** Die Kontrolleinrichtung kann beispielsweise mit einem Bus, insbesondere einem PCI-Bus, zur Übernahme der Grafikdaten signaltechnisch verbun- den und/oder eine Schnittstelle zur signaltechnischen Verbindung mit diesem Bus aufweisen.

**[0011]** Die Grafikverarbeitungs- vorrichtung umfasst ferner ein Geometriemodul zur Erzeugung der Geo- metriedaten der Grafikobjekte aus den Grafikdaten. Das Geometriemodul kann wiederum mehrere Unter- module aufweisen, die eine, einige oder alle der nachfolgenden Aufgaben umsetzen, die nur als Hin- weise ausgestaltet sind, da diese Untermodule be- kannt sind.

Untermodul zur Modell- und Kameratransformation:

**[0012]** In diesem Untermodul erfolgt eine Transfor- mation der Grafikobjekte in ein Referenzkoordinaten- system, insbesondere Weltkoordinatensystem sowie optional ergänzend eine Transformation der Kamera in den Nullpunkt des Referenzkoordinatensystems.

Untermodul zur Beleuchtung:

**[0013]** In diesem Untermodul wird für jedes Gra- fikobjekt, insbesondere für jede Grafikprimitive eine Farbe und/oder Helligkeit berechnet.

Untermodul zur Projektion:

**[0014]** In diesem Untermodul erfolgt die Transfor- mation des Sichtvolumens in einen Würfel.

Clipping:

**[0015]** In diesem Untermodul werden Grafikobjekte abgeschnitten, die sich außerhalb des Sichtvolumens befinden.

## Window-Viewport-Transformation:

**[0016]** In diesem Untermodul erfolgt eine Verschiebung sowie Skalierung der Grafikobjekte.

**[0017]** Als Verarbeitungsergebnis gibt das Geometriemodul Geometriedaten der Grafikobjekte aus. Diese Geometriedaten werden von einem Rastermodul, welches eine weitere Komponente der Grafikverarbeitungsvorrichtung bildet, aufgenommen und zur Rasterung der Grafikobjekte ausgewertet. Insbesondere werden in dem Rastermodul alle Grafikobjekte, insbesondere alle Primitiven, gerastert, also die zu ihnen gehörenden Pixel eingefärbt. Das Rastermodul erzeugt somit Pixel der Grafikobjekte.

**[0018]** Die Kontrolleinrichtung, das Geometriemodul und das Rastermodul bilden zusammen eine Grafikpipeline, welche ausgebildet ist, in einem seriellen Durchgang eines der Grafikobjekte abzuarbeiten, indem aus den Grafikdaten von der Anwendung zu dem Grafikobjekt Pixel des Grafikobjekts erzeugt und ausgegeben werden. Das Geometriemodul und das Rastermodul werden nachfolgend zusammenfassend auch als die Module bezeichnet.

**[0019]** Insbesondere ist die Grafikverarbeitungsvorrichtung als eine GPU (Graphics Processing Unit), im speziellen als eine FPGA-GPU ausgebildet, welche eine Mehrzahl, insbesondere mehr als 50, vorzugsweise mehr als 100 derartiger Grafikpipelines aufweist.

**[0020]** Im Rahmen der Erfindung wird vorgeschlagen, dass die Grafikpipeline ausgebildet ist, die Bearbeitung eines der ersten Grafikobjekte in einem ersten seriellen Durchlauf zu unterbrechen, sodass ein teilbearbeitetes Grafikobjekt vorliegt, mindestens eines der zweiten Grafikobjekte in einem zweiten seriellen Durchlauf zu bearbeiten und nachfolgend die Bearbeitung des teilbearbeiteten Grafikobjekts in der gleichen Grafikpipeline weiterzuführen.

**[0021]** Insbesondere ist die Grafikpipeline so ausgebildet, dass ein zeitliches Multiplexen in Bezug auf die Grafikobjekte der ersten und der zweiten Anwendung erfolgt. Dabei werden nacheinander Grafikobjekte der ersten und der zweiten Anwendung bearbeitet, wobei jedoch die Bearbeitung des Grafikobjekts der ersten Anwendung nicht vollständig erfolgen muss, vielmehr kann die Bearbeitung des Grafikobjekts der ersten Anwendung unterbrochen werden und die Bearbeitung eines Grafikobjekts der anderen Anwendung zwischengeschoben werden, bevor die Bearbeitung des Grafikobjekts der ersten Anwendung beendet wird.

**[0022]** Bevorzugt erfolgt die Unterbrechung für jedes Grafikobjekt der ersten Anwendung.

**[0023]** Durch die Ausbildung der Erfindung wird erreicht, dass innerhalb der Grafikverarbeitungsvorrichtung, insbesondere der Grafikpipeline, eine Trennung, insbesondere eine zeitliche Trennung, zwischen verschiedenen Anwendungen hergestellt wird. Im Englischen wird diese Trennung auch als temporal segregation bezeichnet.

**[0024]** Eine derartige Trennung ist für die Zulassung von Grafikverarbeitungsvorrichtungen nach hohem Sicherheitslevel, zum Beispiel nach der DO-178 B oder der ED-12 B notwendig. Beispielsweise können die zwei Anwendungen unterschiedlichen Sicherheitsstufen, zum Beispiel DAL A und DAL B zugeordnet sein und trotzdem auf die gleiche Datenverarbeitungsvorrichtung, insbesondere Grafikpipeline zugreifen. Der Vorteil der Erfindung ist somit darin zu sehen, dass durch die zeitliche Trennung der beiden Anwendungen in der Grafikverarbeitungsvorrichtung eine höhere Sicherheit erreicht wird. Dagegen enthalten kommerzielle Grafikprozessoren keine derartigen Sicherheitsmechanismen, insbesondere, da diese im normalen Umfeld (Einbau in PC, Notebook usw.) nicht notwendig sind.

**[0025]** Bei einer bevorzugten Ausführungsform der Erfindung gibt die Grafikpipeline eine Serie von aufeinanderfolgenden, jeweils gleichlangen Zeitscheiben vor. Beispielsweise können die aufeinanderfolgenden Zeitscheiben so gewählt sein, dass 30 Zeitscheiben pro Sekunde vorgesehen sind, die Grafikpipeline somit einen Takt von 30 Hz aufweist.

**[0026]** Es ist bevorzugt vorgesehen, dass innerhalb einer gemeinsamen Zeitscheibe die Teilbearbeitung eines der ersten Grafikobjekte und die Bearbeitung mindestens eines der zweiten Grafikobjekte erfolgt. Diese Aufteilung der Grafikobjekte auf verschiedene, jedoch gemeinsame Zeitscheiben hat den Vorteil, dass zum Beispiel von einer ersten Anwendung mit einer höheren Aktualisierungsrate wie zum Beispiel 30 Hz in jeder Zeitscheibe ein Grafikobjekt bearbeitet wird, sodass die Grafikobjekte dieser Anwendung verzögerungsfrei bearbeitet werden. Bei einer anderen Anwendung ist die Aktualisierungsrate beispielsweise geringer, zum Beispiel 15 Hz, wobei pro Zeitscheibe der Grafikpipeline ein Grafikobjekt zur Hälfte oder zu einem vorteilhaften Anteil bearbeitet wird und der verbleibende Rest des Grafikobjekts in einer nachfolgenden Zeitscheibe als teilbearbeitetes Grafikobjekt weiterbearbeitet wird.

**[0027]** Obwohl in der Beschreibung nur Bezug auf zwei Anwendungen genommen wird, ist es möglich, dass auch mehr Anwendungen auf die Grafikverarbeitungsvorrichtung insbesondere die Grafikpipeline, zugreifen und sich die Ressource teilen. Es ist prinzipiell auch denkbar, dass zwei Anwendungen mit einer Aktualisierungsrate von 15 Hz zugreifen, wobei in jeder Zeitscheibe der Grafikpipeline ein Grafikobjekt

jeder Anwendung teilweise bearbeitet wird. Weiterhin ist es denkbar, dass eine Anwendung eine noch geringere Aktualisierungsrate aufweist, zum Beispiel 10 Hz, sodass die Bearbeitung des Grafikobjekts auf mehr als zwei Zeitscheiben, hier drei Zeitscheiben verteilt wird.

**[0028]** Allgemein beschrieben ermöglicht es die Grafikverarbeitungsrichtung, Grafikobjekte verschiedener Anwendungen derart auf die Zeitscheiben zu verteilen, sodass die Bearbeitung mindestens eines Grafikobjekts über mindestens zwei oder mehrere Zeitscheiben erfolgt.

**[0029]** Bei einer bevorzugten Ausführungsform der Erfindung definieren die aufeinanderfolgenden Zeitscheiben einen Synchronkontakt für die Bearbeitung der Grafikobjekte. Die Bearbeitung der Grafikobjekte der ersten Anwendung erfolgt dagegen asynchron zu diesem Synchronkontakt. Dabei kann vorgesehen sein, dass die Unterbrechung der Bearbeitung der ersten Grafikobjekte nach einem starren Muster erfolgt, so dass z. B. die Bearbeitung eines der ersten Grafikobjekte stets auf zwei Zeitscheiben verteilt wird. Alternativ kann die Unterbrechung der Bearbeitung der ersten Grafikobjekte auch dynamisch, insbesondere in Abhängigkeit der verfügbaren Arbeitszeit innerhalb einer Zeitscheibe erfolgen. Besonders bevorzugt erfolgt die Unterbrechung stets zu dem gleichen Zeitpunkt innerhalb eines Zeitfensters.

**[0030]** Bei einer bevorzugten Realisierung der Erfindung weist die Grafikverarbeitungsrichtung eine Zeitplanungseinrichtung auf, die ausgebildet ist, ein Speichersignal zur Unterbrechung der Bearbeitung des ersten Grafikobjekts an die Grafikpipeline auszugeben.

**[0031]** Bevorzugt ist, dass die Module, insbesondere das Geometriemodul und das Rastermodul, bei Anlegen des Speichersignals ihre internen Kalkulationen beendet, also eine Primitivenrenderung oder dergleichen, aber keinen weiteren Befehl zu dem aktuellen Grafikobjekt mehr annimmt. Somit wird sichergestellt, dass die Unterbrechung der Bearbeitung des ersten Grafikobjekts nicht unkontrolliert, sondern zu einem Zeitpunkt erfolgt, an den zu einem weiteren Zeitpunkt, insbesondere während einer weiteren Zeitscheibe angeknüpft werden kann.

**[0032]** Als weitere Reaktion auf das Speichersignal ist die Grafikpipeline ausgebildet, Kontextdaten von dem teilbearbeiteten Grafikobjekt auszugeben. Die Kontextdaten umfassen insbesondere die Registerdaten der Module sowie die Daten aus den benachbarten Eingangs- und Ausgangsspeichern, insbesondere Fifo-Speichern. Diese Kontextdaten werden optional zwischengespeichert und für den späteren seriellen Durchlauf durch die Grafikpipeline zur Verfügung gestellt, um die Bearbeitung weiterzuführen.

Zu dem Zeitpunkt, an dem das Speichersignal anliegt, werden somit alle für das aktuelle, teilbearbeitete Grafikobjekt zur Weiterbearbeitung notwendigen Daten als Kontextdaten zusammengefasst und optional zwischengespeichert.

**[0033]** Nachdem die Kontextdaten des aktuellen teilbearbeiteten Grafikobjekts aus der Grafikpipeline ausgelesen sind, kann die Grafikpipeline für die Bearbeitung eines der zweiten Grafikobjekte bereitgestellt und verwendet werden.

**[0034]** Bei einer bevorzugten Weiterbildung der Erfindung werden die Kontextdaten an dem Ende der Grafikpipeline, insbesondere über das Rastermodul, an die Zeitplanungseinrichtung ausgegeben. Die Kontextdaten werden somit seriell für die unterschiedlichen Module und gegebenenfalls zwischengeschalteten Speichereinrichtungen, insbesondere Fifo-Einrichtungen, ausgelesen. Diese Realisierung vereinfacht zum einen den signaltechnischen Aufbau der Grafikverarbeitungsrichtung, da nicht jedes Modul und jede Speichereinrichtung einen eigenen Datenausgang zur Ausgabe der Kontextdaten aufweisen muss. Zum zweiten können die Kontextdaten geordnet, das heißt, in einer determinierten Reihenfolge ausgelesen werden und – wie später noch erläutert wird – zur Weiterbearbeitung wieder geordnet in die Grafikpipeline eingegeben werden.

**[0035]** Bei einer bevorzugten Ausbildung der Erfindung ist die Zeitplanungseinrichtung ausgebildet, die Kontextdaten in einem Speicher abzulegen. Dadurch, dass die Kontextdaten in einem Speicher abgelegt werden, kann die Zeitplanungseinrichtung nun Grafikdaten eines anderen Grafikobjekts in die Grafikpipeline einleiten, ohne von den Kontextdaten behindert zu sein.

**[0036]** Bei einer bevorzugten Weiterbildung der Erfindung ist die Zeitplanungseinrichtung ausgebildet, ein Wiederherstellungssignal zur Weiterführung der Bearbeitung des teilbearbeiteten Grafikobjekts an die Grafikpipeline auszugeben. Zu diesem Zeitpunkt liest die Zeitplanungseinrichtung die Kontextdaten gegebenenfalls aus dem Speicher aus und sendet diese wieder in die Grafikpipeline zur weiteren Bearbeitung.

**[0037]** Bei einer bevorzugten Realisierung der Erfindung bildet die Zeitplanungseinrichtung mit der Grafikpipeline eine Ringstruktur, sodass bei einem Speichersignal die Kontextdaten an dem Ende der Grafikpipeline ausgelesen und dass bei einem Wiederherstellungssignal die Kontextdaten in die Grafikpipeline eingespielt werden. Optional ergänzend ist an der Ringstruktur der Speicher zur temporären Speicherung der Kontextdaten angeschlossen und/oder ein Eingang zur Übernahme von Grafikdaten zu den Grafikobjekten von den Anwendungen vorgesehen.

**[0038]** Besonders bevorzugt sind zwischen den Modulen oder den Untermodulen, wie sie zuvor beschrieben wurden, Speichereinrichtungen zwischengeschaltet, welche insbesondere als Fifo-Speicher (first in first out) ausgebildet sind. Die Kontextdaten weisen insbesondere Registerdaten der Module und Fifo-Daten von den Speichereinrichtungen auf.

**[0039]** Vorzugsweise haben die Registerdaten einen Signalfag, insbesondere ein Signalbit, der bei einem normalen Betrieb der Grafikpipeline auf einen ersten Wert, nachfolgend beispielhaft 0, gesetzt ist und bei einem Speichervorgang auf einen anderen Wert, nachfolgend beispielhaft 1, gesetzt ist. Mit dem Signalfag werden somit die als Kontextdaten zu speichernden Registerdaten gekennzeichnet. Der Signalfag hat den Vorteil, dass Registerdaten, die einen Signalfag mit dem Wert 1 aufweisen, seriell durch die Grafikpipeline durchgeschoben und nicht bearbeitet werden.

**[0040]** Optional weisen auch die Fifo-Daten den Signalfag auf, wobei dieser jedoch stets auf den ersten Wert, also 0, gesetzt ist. Dadurch ist es möglich, nach einem Speichersignal Fifo-Daten von Registerdaten zu unterscheiden.

**[0041]** Nachdem – wie bereits erläutert – die Registerdaten und die Fifo-Daten seriell ausgelesen werden und zudem der Signalfag bei einem Auslesevorgang bei den Fifo-Daten auf den Wert 0 und bei den Registerdaten auf den Wert 1 gesetzt ist, kann der Signalfag zur Auswertung von Positionen von Daten herangezogen werden.

**[0042]** Betrachtet man einen Datenfluss der Kontextdaten, so kommt z. B. zunächst ein erster Registerdatenblock, dann ein erster Fifo-Datenblock, dann ein zweiter Registerdatenblock, dann ein zweiter Fifo-Datenblock etc. Diese Reihenfolge lässt sich aufgrund der Reihenfolge und des Signalfags eindeutig zu den Modulen und den Speichereinrichtungen zuordnen, sodass das spätere Wiedereinspielen der Kontextdaten in die Grafikpipeline verwechslungsfrei erfolgen kann.

**[0043]** Bei einer bevorzugten Weiterbildung der Erfindung weist die Grafikverarbeitungsvorrichtung einen Pixelspeicher auf, der zur Speicherung der Pixel aus dem Rastermodul ausgebildet ist. Es ist besonders bevorzugt, dass die Zeitplanungseinrichtung ausgebildet ist, den Pixeln von unterschiedlichen Anwendungen in dem Pixelspeicher unterschiedliche Speicherbereiche zuzuweisen. Vorzugsweise kann die Zuweisung der Speicherbereiche vom User oder von der Anwendung nicht beeinflusst werden, sodass hierdurch eine sichere örtliche Trennung (spatial Segregation) der Anwendungen gegeben ist. Diese Weiterbildung erhöht weiter den Sicherheitsstandard der Grafikverarbeitungsvorrichtung.

**[0044]** Ein weiterer Gegenstand der Erfindung betrifft eine Anzeigevorrichtung für ein Flugzeugcockpit mit der Grafikverarbeitungsvorrichtung, wie sie zuvor beschrieben wurde, beziehungsweise nach einem der vorhergehenden Ansprüche, wobei die Anzeigevorrichtung mindestens einen ersten und einen zweiten Anzeigebereich aufweist, wobei auf dem ersten Anzeigebereich Grafikobjekte der ersten Anwendung und auf dem zweiten Anzeigebereich Grafikobjekte der zweiten Anwendung darstellbar sind. Bei den Anzeigebereichen kann es sich um zwei Anzeigebereiche eines gemeinsamen Bildschirms handeln oder um Anzeigebereiche von zwei unterschiedlichen Bildschirmen.

**[0045]** In einer konkreten Ausgestaltung der Erfindung dient der erste Anzeigebereich zur Darstellung eines Primär-Flugdisplays (PFD) und der zweite Anzeigebereich zur Darstellung eines Navigationsdisplays (ND). Diese zwei Anwendungen sind in dem Flugzeugcockpit beide sicherheitsrelevant, sodass die Softwareumsetzung einen hohen Sicherheitsstandard erfordert, wie dieser durch die Grafikverarbeitungsvorrichtung, wie sie zuvor beschrieben wurde, erreicht wird.

**[0046]** Besonders bevorzugt weisen die erste und die zweite Anwendung unterschiedliche Aktualisierungsraten auf.

**[0047]** Ein weiterer Gegenstand der Erfindung betrifft ein Verfahren mit den Merkmalen des Anspruchs 14.

**[0048]** Weitere Merkmale, Vorteile und Wirkungen der Erfindung ergeben sich aus der nachfolgenden Beschreibung bevorzugter Ausführungsbeispiele der Erfindung. Dabei zeigen:

**[0049]** [Fig. 1](#) ein schematisches Blockdiagramm einer Grafikverarbeitungsvorrichtung als ein Ausführungsbeispiel der Erfindung;

**[0050]** [Fig. 2](#) eine Zeitlinie, in der schematisch die Bearbeitung von Grafikobjekten in der Grafikverarbeitungsvorrichtung der [Fig. 1](#) dargestellt ist;

**[0051]** [Fig. 3](#) eine Zeitlinie, in der schematisch die Struktur der Kontextdaten aus der Grafikverarbeitungsvorrichtung in der [Fig. 1](#) dargestellt ist;

**[0052]** [Fig. 4a](#), b, c, d die Grafikverarbeitungsvorrichtung der vorhergehenden Figuren in verschiedenen Betriebszuständen zur Erläuterung ihrer Funktionsweise;

**[0053]** [Fig. 5](#) eine schematische Darstellung einer Anzeigevorrichtung für ein Flugzeugcockpit mit der Grafikverarbeitungsvorrichtung der vorhergehenden Figuren;

**[0054]** [Fig. 6a](#), b eine schematische Darstellung der zeitlichen Abfolge bei einer Unterbrechung und einer Wiederaufnahme der Bearbeitung eines Grafikobjekts.

**[0055]** Die [Fig. 1](#) zeigt in einer schematischen Blockdarstellung eine Grafikverarbeitungsvorrichtung **1** als ein Ausführungsbeispiel der Erfindung, welche es ermöglicht, Grafikbefehle aus Anwendungen in Pixel umzusetzen, die auf Anzeigeeinrichtungen als graphische Ausgabe der Anwendungen dargestellt werden können. Die Grafikverarbeitungsvorrichtung **1** ist als Hardware realisiert und als ein FPGA ausgebildet.

**[0056]** Die Grafikverarbeitungsvorrichtung **1** umfasst eine Kontrolleinrichtung **2**, die über eine Schnittstelle **3** mit einem Datenbus, wie z. B. einem PCI-Bus, verbunden ist. Über die Schnittstelle **3** werden von Anwendungen Grafikdaten, insbesondere Grafikbefehle an die Kontrolleinrichtung **2** zur Generierung der Pixel für die Anwendung übergeben.

**[0057]** Ausgehend von der Kontrolleinrichtung **2** werden die Grafikdaten an eine erste Speichereinrichtung **4** mit einer Fifo-Architektur übergeben, die als Eingangspuffer die Grafikdaten für eine Bearbeitung in dem nachfolgenden Geometriemodul **5** bereitstellt. In dem Geometriemodul **5** werden die Grafikdaten, insbesondere die Grafikbefehle, teilweise umgesetzt und Grafikprimitive, also z. B. Dreiecke oder Linien, gegebenenfalls mit Farbdefinitionen, gebildet. Die bearbeiteten Grafikdaten bzw. die Grafikprimitive werden in eine weitere Speichereinrichtung **6**, die ebenfalls als Fifo-Speicher ausgebildet ist, übergeben und dann in ein Rastermodul **7** eingelesen. In dem Rastermodul **7** werden aus den Geometriep primitiven zusammen mit den Farbdefinitionen Pixel erzeugt, die nachfolgend in einem Pixelspeicher **8** und von diesem über eine Datenschnittstelle **9** ausgegeben werden.

**[0058]** Die Kontrolleinrichtung **2**, das Geometriemodul **5** und das Rastermodul **7** zusammen mit den Speichereinrichtungen **4** und **6** bilden eine so genannte Grafikpipeline, da die Grafikdaten innerhalb der Grafikpipeline seriell abgearbeitet werden. Während in dem gezeigten Blockschaltbild in der [Fig. 1](#) das Geometriemodul **5** und das Rastermodul **7** nur als ein einziger Block dargestellt sind, kann bei einer hardwaretechnischen Realisierung das Geometriemodul **5** auch in mehrere, insbesondere fünf Untermodule aufgeteilt sein, wobei zwischen den Untermodulen jeweils Speichereinrichtungen in Fifo-Architektur angeordnet sind. In gleicher Weise kann auch das Rastermodul **7** in mehrere Untermodule unterteilt sein, die unter Zwischenschaltung von Speichereinrichtungen, ausgebildet als Fifo-Speicher, signaltechnisch miteinander verbunden sind.

**[0059]** In der Grafikverarbeitungsvorrichtung **1** können mehrere derartige Grafikpipelines parallel zueinander angeordnet sein, um gemeinsam die Pixel für die graphische Anzeige der Anwendung zu erzeugen. Die Grafikverarbeitungsvorrichtung **1** bzw. die Grafikpipeline arbeitet in einem Synchrontakt, z. B. von 30 Hertz, so dass bei Ausnutzung des Synchrontakts die Anwendung ein anzeigbares Bild auf der Anzeigeeinrichtung mit einer Aktualisierungsrate gleich dem Synchrontakt ausgeben kann.

**[0060]** Die Grafikerzeugung für eine einzige Anwendung ist durch die bereits beschriebenen Komponenten der Grafikverarbeitungsvorrichtung **1** bereits möglich. Die Datenverarbeitungsvorrichtung **1** ist jedoch ergänzend in der Lage, mehrere Anwendungen, insbesondere mindestens zwei Anwendungen, quasi-parallel zu bearbeiten, wobei datentechnisch betrachtet eine zeitliche als auch örtliche Trennung zwischen den Anwendungen hergestellt wird (temporal and spatial segregation). Eine derartige Trennung ist für die Zulassung von Soft- und Hardware nach hohem Sicherheitslevel, z. B. DAL A/B, notwendig.

**[0061]** Das zugrunde liegende Prinzip beruht darauf, dass zu einem wählbaren Zeitpunkt Daten und Parameter innerhalb der Grafikpipeline als Kontextdaten gespeichert werden können und später wieder verwendet werden können, so dass eine Anwendung in der Grafikpipeline kurzzeitig ausgesetzt werden kann.

**[0062]** Der Vorteil der Grafikverarbeitungsvorrichtung **1** liegt in einer höheren Performance und in der Möglichkeit, verschiedene Applikationen auf der gleichen Datenverarbeitungsvorrichtung **1** zu mischen. So wird es z. B. möglich, eine DAL C-Funktion, die lediglich einzeimal pro Sekunde aufgerufen wird, mit einer DAL A-Funktion, die 30 mal pro Sekunde aufgerufen wird, zu mischen. Dabei kann das Bild der DAL C-Funktion in einem Zyklus des Synchrontakts teilweise generiert werden und die Generierung später fortgesetzt werden.

**[0063]** Diese prinzipielle Funktionsweise ist in der [Fig. 2](#) schematisiert dargestellt, die eine Zeitlinie für die Datenverarbeitungsvorrichtung **1** zeigt.

**[0064]** Der waagrecht verlaufende, untere Pfeil **10** zeigt den Synchrontakt mit der Frequenz  $f$ , wobei die einzelnen Taktabschnitte durch Pfeile **11** unterbrochen sind.

**[0065]** In dem ersten Synchrontakt **1** wird von der Datenverarbeitungsvorrichtung **1** ein Bild 0 (Frame 0) einer zweiten Anwendung App\_2 vollständig bearbeitet. Nachfolgend wird das Bild 0 (Frame 0) einer ersten Anwendung App\_1 beispielsweise zur Hälfte bearbeitet, so dass die Bearbeitung dieses Bildes 0 un-



terbrochen wird. Der verbleibende Bereich (spare) in dem Synchronkontakt 1 ist ein Sicherheitszeitpuffer.

**[0066]** In dem Synchronkontakt II wird von der zweiten Applikation App\_2 ein Bild 1 (Frame 1) vollständig bearbeitet. Von der ersten Anwendung App\_1 wird dagegen das Bild 0 (Frame 0) fertig bearbeitet, so dass dies nach dem zweiten Synchronkontakt II zur Verfügung steht.

**[0067]** In dem dritten Synchronkontakt III wird ein Bild 2 (Frame 3) der zweiten Anwendung App\_2 bearbeitet und ein Bild 1 (Frame 1) der ersten Anwendung App\_1 zumindest wieder teilweise bearbeitet.

**[0068]** Mit dieser zeitlichen Aufteilung ist die Grafikerzeugung innerhalb der Grafikverarbeitungsvorrichtung 1 bzw. innerhalb der Grafikpipeline der zwei Anwendungen App\_1 und App\_2 zeitlich voneinander getrennt.

**[0069]** Für die Anwendung App\_2 wird in jedem Synchronkontakt I, II, III ein vollständiges Bild erzeugt, so dass die zweite Anwendung App\_2 eine Aktualisierungsrate entsprechend dem Synchronkontakt, also von z. B. 30 Herz aufweist. Dagegen werden von der ersten Anwendung App\_1 in jedem Synchronkontakt I, II, III nur ein halbes Bild oder ein Teilbild bearbeitet, die Bearbeitung unterbrochen und in dem nächsten Synchronkontakt weitergeführt. Damit erhält die erste Anwendung eine Aktualisierungsrate von f-halbe, z. B. 15 Hertz.

**[0070]** Für die erste Anwendung App\_1 ist es notwendig, nach der Unterbrechung der Bearbeitung des Bildes die Daten, die zur Fertigstellung des Bildes in dem nächsten Synchronkontakt benötigt werden – nachfolgend als Kontextdaten bezeichnet –, zwischen zu speichern, um sich zeitlich und örtlich von der zweiten Anwendung App\_2 zu trennen und deren Abarbeitung nicht zu behindern.

**[0071]** Prinzipiell ist es möglich, die Bearbeitung in jedem Status des Umsetzungsprozesses zu unterbrechen, z. B. während eine Rotlinie auf der Anzeigeeinrichtung gezeichnet wird. Als Kontextdaten nach der Unterbrechung werden alle notwendigen Daten gespeichert, die zur Weiterbearbeitung benötigt werden. Für das Beispiel der roten Linie würden das Daten über die aktuelle Farbe und die Position umfassen, wo das Rendern beendet wurde.

**[0072]** Während die Grafikverarbeitungsvorrichtung 1 sich um die Umsetzung der Bilder der Anwendungen kümmert, bearbeitet die mindestens eine Grafikpipeline in der Grafikverarbeitungsvorrichtung 1 Grafikobjekte, welche Bestandteile des Bildes der Anwendung sind. Somit wird in der Grafikpipeline bei der Unterbrechung der Bearbeitung eines Bildes der ersten Anwendung in der Grafikpipeline die Bearbei-

tung von Grafikobjekten, z. B. Grafikprimitiven, in einem Synchronkontakt unterbrochen und in einem anderen weitergeführt.

**[0073]** Zur Steuerung der Grafikverarbeitungsvorrichtung 1 bzw. der Grafikpipeline weist diese eine Zeitplanungseinrichtung 13 auf, welche zum einen gemäß Pfeil 12 einen Speicherbefehl context\_store ausgibt, sobald die Bearbeitung des Bildes der ersten Anwendung App\_1 unterbrochen werden soll, sowie ein Wiederherstellungssignal context\_load, sobald die Bearbeitung des Bildes in dem nächsten Synchronkontakt weitergeführt werden soll.

**[0074]** Weitere Aufgaben der Zeitplanungseinrichtung 13 sind das Speichern und Abrufen der genannten Kontextdaten sowie die Ausgabe von Kontrolldaten zur Kontrolle des Speicherbereichs in dem Pixelspeicher 8, wobei sichergestellt wird, dass die beiden Anwendungen App\_1 und App\_2 auf unterschiedliche, nicht überlappende Speicherbereiche in dem Pixelspeicher 8 zugreifen und somit in diesem Bereich örtlich voneinander getrennt sind. Die Kontrolldaten bilden einen Teil der Daten in dem Rastermodul 7 und werden durch ein spezifisches grafisches Kommando gesetzt werden, das von den Anwendungen App\_1 und App\_2 nicht genutzt werden kann. Auf diese Weise sind die Daten in dem Pixelspeicher 8 räumlich sicher getrennt, so dass sich eine spatial segregation ergibt.

**[0075]** Nach dem Speicherbefehl context\_store werden die Module, insbesondere das Geometriemodul 5 und das Rastermodul 7 bzw. deren Untermodule, angewiesen, eine vorliegende Kalkulation von Grafikobjekten zu einem Zwischenabschluss zu bringen, der so gewählt wird, dass an diesen Zwischenabschluss in dem nächsten Synchronkontakt zur Weiterführung der Grafikgenerierung angeknüpft werden kann. Nachfolgend werden die Kontextdaten gemäß der gestrichelten Pfeile in der [Fig. 1](#) ausgelesen.

**[0076]** Die Kontextdaten, beginnend mit Fifo-Daten aus der Speichereinrichtung 4, Registerdaten aus dem Geometriemodul 5, Fifo-Daten aus der Speichereinrichtung 6 und Registerdaten aus dem Rastermodul 7 werden seriell weitergeschoben und über einen zusätzlichen Datenpfad an die Zeitplanungseinrichtung 13 übergeben. Damit wird eine Ringstruktur zwischen der Grafikpipeline und der Zeitplanungseinrichtung 13 erzeugt. Die Zeitplanungseinrichtung 13 legt die Kontextdaten über die Datenschnittstelle 9 in einem Speicher ab.

**[0077]** Nachdem die Kontextdaten ausgelesen sind, ist die Speicherverarbeitungsvorrichtung 1 bzw. die Grafikpipeline bereit, in einem nächsten Synchronkontakt ein Bild der zweiten Anwendung App\_2 zu bearbeiten.

**[0078]** Die Struktur der auszulesenden Daten ist schematisch in der [Fig. 3](#) dargestellt. In der [Fig. 3](#) ist zu erkennen, dass in einem Datenstream zunächst Fifo-Daten FIFO-4 der Speichereinrichtung **4**, nachfolgend Registerdaten REG-5 des Geometriemoduls **5**, nachfolgend Fifo-Daten FIFO-6 der Speichereinrichtung **6** und nachfolgend Registerdaten REG-7 des Rastermoduls **7** angeordnet sind. Die Daten in dem Datenstream entsprechen den Daten aus den Modulen **5**, **7** bzw. aus den Speichereinrichtungen **4**, **6** vor dem Auslesen oder Weiterschieben der Daten nach dem Speicherbefehl context-store. Falls die Grafikkpipeline mehr Untermodule und mehr zwischen den Untermodulen angeordnete Speichereinrichtungen aufweisen sollte, ist der Datenstream entsprechend zu verlängern. Die Kontextdaten werden durch einen Signalfag **14** erweitert, welcher in normalem Betriebsmodus stets auf 0 gesetzt ist, so dass die Module daraus ablesen können, dass die entsprechenden Registerdaten bearbeitet werden sollen. Liegt jedoch der Speicherbefehl context\_store an, so setzt jedes der Module den Signalfag seiner Registerdaten auf 1 und zeigt damit, dass die nachfolgenden Module die Registerdaten nicht bearbeiten sondern über einen Bypass in Richtung der Zeitplanungseinrichtung **13** weiterschieben sollen. Die Daten der Speichereinrichtungen **4**, **6** behalten den Signalfag stets auf "0", wobei der Signalfag von den nachfolgenden Modulen als "don't care" interpretiert wird, so dass auch diese Daten über einen Bypass weitergeschoben werden.

**[0079]** Für den Fall, dass in den Speichereinrichtungen **4**, **6** oder in den Modulen **5**, **7** keine Daten vorhanden sind, wird ein Dummy-Wort eingeschrieben, um die Struktur des Datenstreams zu erhalten.

**[0080]** Bei der Weiterführung der Bearbeitung in dem nächsten Synchronkontakt legt die Zeitplanungseinrichtung **13** das Wiederherstellungssignal context\_load an, liest die Kontextdaten des teilbearbeiteten Bildes über die Datenschnittstelle **9** aus und schiebt diese Kontextdaten wieder in die Grafikkpipeline. Die Zuordnung der Daten in den Kontextdaten kann über die Reihenfolge und den alternierenden Signalfag **14** erschlossen werden. So ist der Speichereinrichtung **4** bekannt, dass sie den letzten Datenblock für Fifo-Daten übernehmen muss. Das Geometriemodul **5** zählt die Anzahl der Signalfags mit und entnimmt den Datenblock, bei dem die Anzahl des Signalfags **14** gleich 2 ist. Die Speichereinrichtung **6** zählt ebenfalls die Signalfags **14** und weiß, dass sie die Fifo-Daten nach dem ersten Signalfag **14** entnehmen muss. Das Rastermodul **7** übernimmt dagegen Daten aus dem ersten Signalfag **14**. Diese Vorgehensweise ist auf eine beliebige Anzahl von Untermodulen zu erweitern, wobei jedes Untermodul die Registerdaten aus dem Datenstream entnimmt, die es anhand der Anzahl der Übergänge des Signalfags **14** verglichen mit

seiner eigenen Position innerhalb der Grafikkpipeline als eigene Daten erkennt.

**[0081]** Nachdem das teilbearbeitete Bild fertig bearbeitet wurde, können nachfolgend über die Kontrolleinrichtung **2** neue Grafikdaten in die Grafikkpipeline eingelesen werden, wobei der Signalfag auf 0 gesetzt ist.

**[0082]** In den [Fig. 4a](#), [b](#), [c](#) und [d](#) sind die Einzelschritte nochmals dargestellt. In der [Fig. 4a](#) befindet sich die Grafikkpipeline im normalen Betriebszustand, wobei Grafikdaten von der Kontrolleinrichtung **2** über die Speichereinrichtung **4** in das Geometriemodul **5**, nachfolgend in die Speichereinrichtung **6** und das Rastermodul **7** durch geleitet und bearbeitet werden.

**[0083]** Sobald das Speichersignal context\_store anliegt, werden keine weiteren Grafikdaten von der Kontrolleinrichtung **2** in die Grafikkpipeline eingespeist. Das Geometriemodul **5** und das Rastermodul **7** beenden ihre aktuellen Kalkulationen und setzen den Signalfag **14** ihrer aktuellen Registerdaten auf den Wert 1, wie das in der [Fig. 4b](#) dargestellt ist. Die Daten in den Speichereinrichtungen **4**, **6** behalten als Fifo-Daten den Signalfag auf "0".

**[0084]** Gemäß der [Fig. 4c](#) werden die Kontextdaten ausgelesen und an die Zeitplanungseinrichtung **13** übergeben, wobei das Rastermodul **7** Daten aus den Speichereinrichtungen **6** mit einem auf "0" als "don't care" gesetzten Signalfag **14** sowie Daten aus dem Geometriemodul **5** mit einem auf "1" gesetzten Signalfag über einen Bypass unbearbeitet durchreicht, so dass der Datenstream mit den Kontextdaten mit der Struktur gemäß der [Fig. 3](#) zu der Zeitplanungseinrichtung **13** fließt. Diese Kontextdaten werden wie beschrieben über die Datenschnittstelle **9** gespeichert.

**[0085]** In der [Fig. 4d](#) ist das Verhalten der Datenverarbeitungsvorrichtung **1** bei Anlegen des Wiederherstellungssignals context\_load dargestellt, wobei von der Zeitplanungseinrichtung **13** die Kontextdaten über die Datenschnittstelle **9** ausgelesen werden und in die Grafikkpipeline eingeleitet werden. Dabei zählt jede Speichereinrichtung **4**, **6** und jedes Modul **5**, **7** die Anzahl der Signalfags **14** ab, um die eigenen Daten zu erkennen. Sobald die Daten verteilt sind, wird die Weiterbearbeitung des teilbearbeiteten Grafikobjekts aufgenommen.

**[0086]** Die [Fig. 5](#) zeigt schließlich eine Anzeigeeinrichtung **15** mit einer ersten Anzeige **16a** und einer zweiten Anzeige **16b**, welche über die Grafikverarbeitungsvorrichtung **1**, wie sie zuvor beschrieben wurde, angesteuert werden. Auf der Anzeige **16a** wird beispielsweise ein Primär-Flugdisplay (PFD) und auf der Anzeige **16b** ein Navigationsdisplay (ND) angezeigt,



welche durch zwei unterschiedliche Anwendungen, wie z. B. App\_1 und App\_2 in der Grafikverarbeitungsvorrichtung erzeugt werden.

**[0087]** In den [Fig. 6a](#), b ist die zeitliche Abfolge einer Unterbrechung und der Wiederaufnahme der Bearbeitung eines Grafikobjekts in der Grafikpipeline auf einer Zeitachse t veranschaulicht:

In dem Abschnitt A wird die Bearbeitung des Grafikobjekts regulär durchgeführt. Bei Anliegen des context\_store Befehls werden die aktuellen Berechnungen in dem Abschnitt A beendet und damit die Bearbeitung des aktuellen Grafikobjekts unterbrochen. Nach der Beendigung werden in dem Abschnitt B die Kontextdaten ausgelesen. In einer nächsten Zeitscheibe werden die Kontextdaten in dem Abschnitt V wieder eingelesen und in dem Abschnitt D weiterbearbeitet.

**[0088]** Die [Fig. 6a](#) zeigt eine reguläre Unterbrechung der Bearbeitung des Grafikobjekts, die in jeder betroffenen Zeitscheibe zu dem gleichen Zeitpunkt erfolgt. In der [Fig. 6b](#) ist dagegen die Situation bei einem time-out in einer Zeitscheibe dargestellt, wobei in Abgrenzung zu der [Fig. 6a](#) das Auslesen der Kontextdaten sofort nach Anliegen des context\_store Befehls erfolgt:

#### Bezugszeichenliste

<b>1</b>	Grafikverarbeitungsvorrichtung
<b>2</b>	Kontrolleinrichtung
<b>3</b>	Schnittstelle
<b>4</b>	Speichereinrichtung
<b>5</b>	Geometriemodul
<b>6</b>	Speichereinrichtung
<b>7</b>	Rastermodul
<b>8</b>	Pixel Speicher
<b>9</b>	Datenschnittstelle
<b>10</b>	unterer Pfeil
<b>11</b>	Pfeile Taktabschnitt
<b>12</b>	Pfeil Speicherbefehl
<b>13</b>	Zeitplanungseinrichtung
<b>14</b>	Signalflag
<b>15</b>	Anzeigeeinrichtung
<b>16a</b>	erste Anzeige
<b>16b</b>	zweite Anzeige

#### Patentansprüche

1. Grafikverarbeitungsvorrichtung (1) umfassend eine Kontrolleinrichtung (2) zur Übernahme von Grafikdaten zu ersten Graphikobjekten aus einer ersten Anwendung (App\_1) und zur Übernahme von Grafikdaten zu zweiten Graphikobjekten aus einer zweiten Anwendung (App\_2); ein Geometriemodul (5) zur Erzeugung der Geometriedaten der Graphikobjekte aus den Grafikdaten; ein Rastermodul (7) zur Rasterung der Graphikobjek-

te auf Basis der Geometriedaten und zur Erzeugung von Pixeln der Graphikobjekte;

wobei die Kontrolleinrichtung (2), das Geometriemodul (5) und das Rastermodul (7) eine Grafikpipeline bilden, welche ausgebildet ist, in einem seriellen Durchgang eines der Graphikobjekte abzuarbeiten, indem aus den Grafikdaten zu dem Graphikobjekt Pixel des Grafikobjekts erzeugt werden,

**dadurch gekennzeichnet**, dass

die Grafikpipeline ausgebildet ist, die Bearbeitung eines der ersten Graphikobjekte der ersten Anwendung (App\_1) in einem ersten seriellen Durchlauf zu unterbrechen, so dass ein teilbearbeitetes Graphikobjekt vorliegt, mindestens eines der zweiten Graphikobjekte der zweiten Anwendung (App\_2) in einem zweiten seriellen Durchlauf zu bearbeiten und nachfolgend die Bearbeitung des teilbearbeiteten Graphikobjekts der ersten Anwendung (App\_1) weiterzuführen.

2. Grafikverarbeitungsvorrichtung (1) nach Anspruch 1, dadurch gekennzeichnet, dass die Grafikpipeline eine Serie von aufeinanderfolgenden Zeitscheiben vorgibt, wobei innerhalb einer gemeinsamen Zeitscheibe die Teilbearbeitung eines der ersten Graphikobjekte und die Bearbeitung eines der zweiten Graphikobjekte erfolgt.

3. Grafikverarbeitungsvorrichtung (1) nach Anspruch 2, dadurch gekennzeichnet, dass die aufeinanderfolgenden Zeitscheiben einen Synchrontakt (f) für die Bearbeitung der Graphikobjekte vorgeben.

4. Grafikverarbeitungsvorrichtung (1) nach einem der vorhergehenden Ansprüche, gekennzeichnet durch eine Zeitplanungseinrichtung (13), wobei die Zeitplanungseinrichtung (13) ausgebildet ist, ein Speichersignal (context\_store) zur Unterbrechung der Bearbeitung des ersten Graphikobjekts an die Grafikpipeline auszugeben und die Grafikpipeline ausgebildet ist, als Reaktion auf das Speichersignal (context\_store) Kontextdaten von dem teilbearbeiteten Grafikobjekt auszugeben, welche optional zwischengespeichert und für den späteren seriellen Durchlauf durch die Grafikpipeline zur Verfügung gestellt werden, um die Bearbeitung des teilbearbeiteten Grafikobjekts weiterzuführen.

5. Grafikverarbeitungsvorrichtung (1) nach Anspruch 4, dadurch gekennzeichnet, dass die Kontextdaten an dem Ende der Grafikpipeline, insbesondere von dem Rastermodul (7), an die Zeitplanungseinrichtung (13) ausgegeben werden.

6. Grafikverarbeitungsvorrichtung (1) nach Anspruch 5, dadurch gekennzeichnet, dass die Zeitplanungseinrichtung ausgebildet ist, die Kontextdaten in einem Speicher (9) abzulegen.

7. Grafikverarbeitungsvorrichtung (1) nach einem der vorhergehenden Ansprüche 4 bis 6, dadurch ge-

kennzeichnet, dass die Zeitplanungseinrichtung (**13**) ausgebildet ist, ein Wiederherstellungssignal (context\_load) zur Weiterführung der Bearbeitung des teilbearbeiteten Graphikobjekts an die Graphikpipeline auszugeben und die Grafikpipeline ausgebildet ist, als Reaktion auf das Wiederherstellungssignal die Kontextdaten von dem teilbearbeiteten Graphikobjekt einzulesen, um die Bearbeitung weiterzuführen.

8. Grafikverarbeitungsvorrichtung (**1**) nach einer der vorhergehenden Ansprüche 4 bis 7, dadurch gekennzeichnet, dass die Zeitplanungseinrichtung (**13**) mit der Graphikpipeline eine Ringstruktur bildet, so dass bei einem Speichersignal (context\_store) die Kontextdaten an dem Ende der Grafikpipeline ausgelesen und dass bei einer Wiederherstellungsbefehl (context\_load) die Kontextdaten in die Graphikpipeline eingespielt werden.

9. Grafikverarbeitungsvorrichtung (**1**) nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass die Kontextdaten Registerdaten (REG-5, REG-7) der Module (**5**, **7**) und Fifo-Daten (FIFO-4, FIFO-6) von Speichereinrichtungen (**4**, **6**) zwischen und/oder vor den Modulen (**5**, **7**) umfassen, wobei die Registerdaten (REG-5, REG-7) einen Signalflag (**14**) aufweisen, der bei einem normalen Betrieb auf einen ersten Wert (0) und bei einem Speichervorgang auf einen anderen Wert (1) gesetzt ist.

10. Grafikverarbeitungsvorrichtung (**1**) nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass die Registerdaten (REG-5, REG-7) und die Fifo-Daten (FIFO-4, FIFO-6) in den Kontextdaten abwechselnd angeordnet sind, wobei durch die Position der Registerdaten (REG-5, REG-7) und der Fifo-Daten (FIFO-4, FIFO-6) in den Kontextdaten die Zuordnung der Daten zu den Modulen (**5**, **7**) und zu den Speichereinrichtungen (**4**, **6**) festgelegt ist.

11. Grafikverarbeitungsvorrichtung (**1**) nach einem der vorhergehenden Ansprüche, gekennzeichnet durch einen Pixelspeicher (**8**), wobei die Zeitplanungseinrichtung (**13**) ausgebildet ist, den Pixeln der Graphikobjekte der ersten und der zweiten Anwendung (App\_1, App\_2) unterschiedliche Speicherbereiche zuzuweisen, so dass die Speicherbereiche der ersten und der zweiten Anwendung (App\_1, App\_2) in dem Pixelspeicher (**8**) voneinander getrennt sind.

12. Anzeigevorrichtung (**15**) für ein Flugzeugcockpit mit einer Grafikverarbeitungsvorrichtung (**1**) nach einem der vorhergehenden Ansprüche, mit mindestens einem ersten und einem zweiten Anzeigebereich (**16a**, **b**), wobei auf dem ersten Anzeigebereich (**16a**) Graphikobjekte der ersten Anwendung (App\_1) und auf dem zweiten Anzeigebereich (**16b**) Graphikobjekte der zweiten Anwendung (App\_2) darstellbar sind.

13. Anzeigevorrichtung (**15**) nach Anspruch 12, dadurch gekennzeichnet, dass die erste Anwendung (App\_1) und die zweite Anwendung (App\_2) unterschiedliche Aktualisierungsraten aufweisen.

14. Verfahren zur Darstellung von Graphikobjekten von mindestens einer ersten und einer zweiten Anwendung (App\_1, App\_2), wobei in der Grafikverarbeitungsvorrichtung (**1**) nach einem der Ansprüche 1 bis 11 oder in der Anzeigevorrichtung (**15**) nach einem der Ansprüche 12 oder 13 die Bearbeitung eines der ersten Graphikobjekte der ersten Anwendung (App\_1) in einem ersten seriellen Durchlauf unterbrochen wird, so dass ein teilbearbeitetes Graphikobjekt vorliegt, mindestens eines der zweiten Graphikobjekte der zweiten Anwendung (App\_2) in einem zweiten seriellen Durchlauf bearbeitet wird und nachfolgend die Bearbeitung des teilbearbeiteten Graphikobjekts der ersten Anwendung weitergeführt wird.

Es folgen 9 Blatt Zeichnungen

Anhängende Zeichnungen

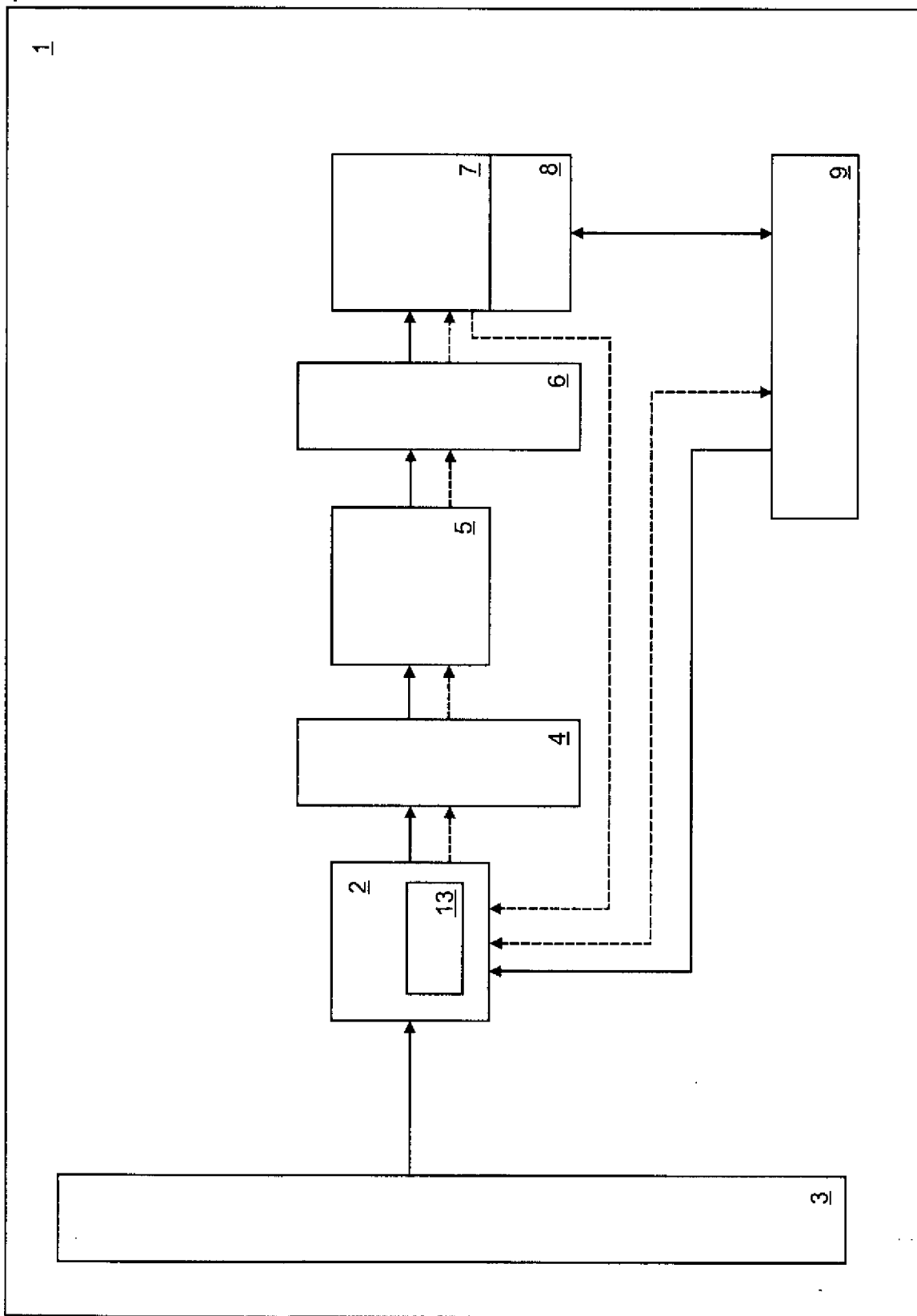


Fig. 1

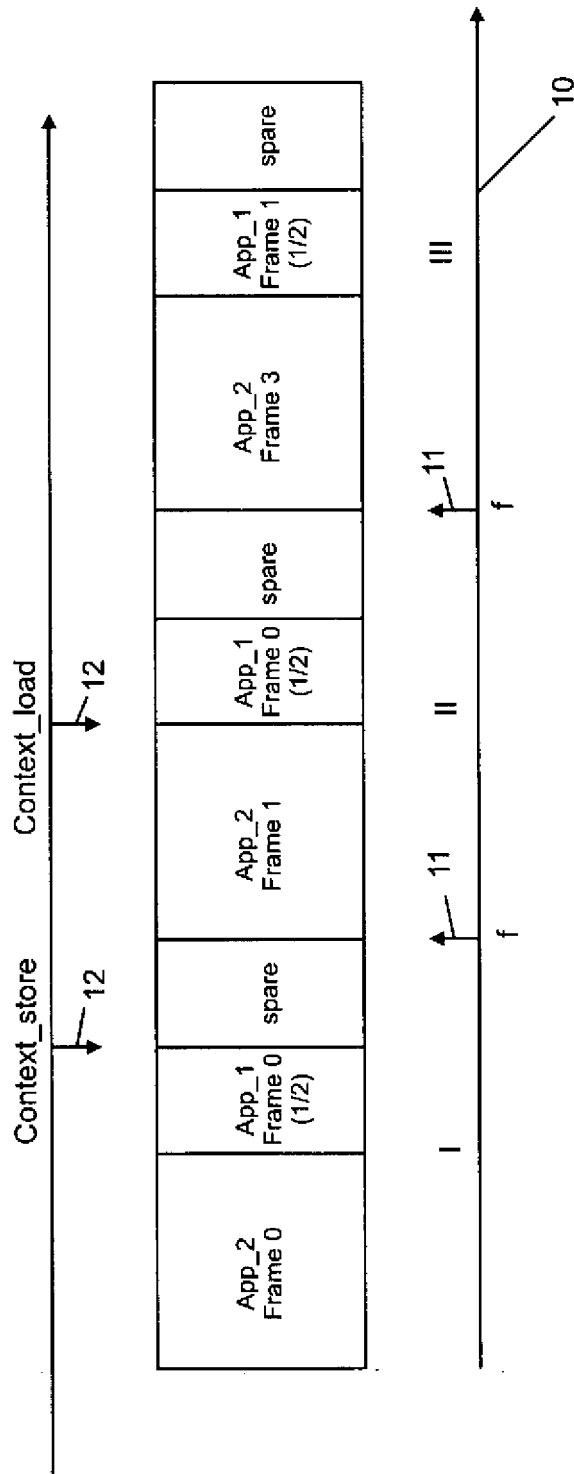


Fig. 2

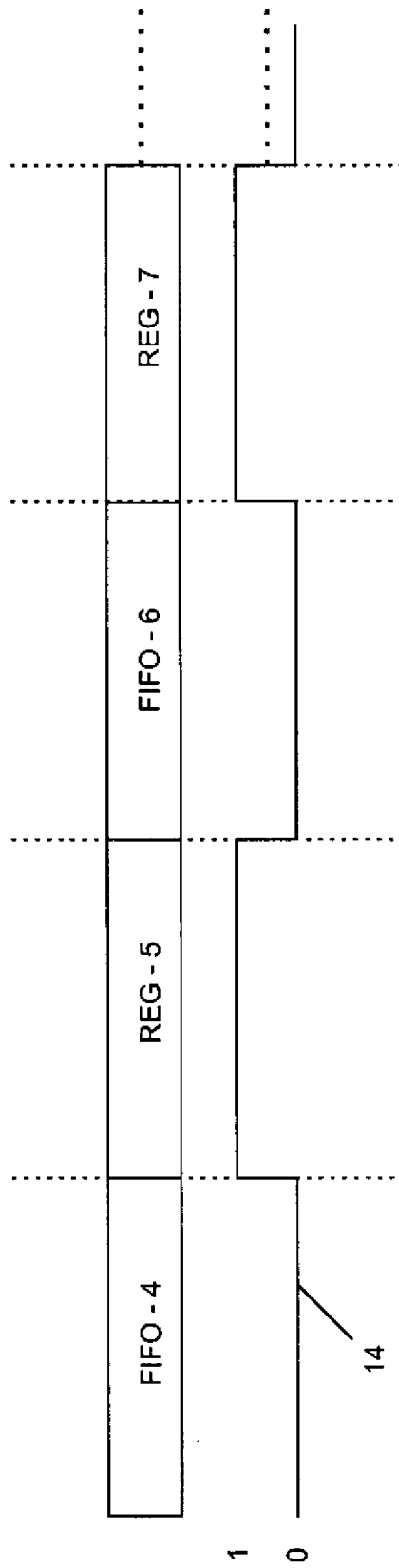


Fig. 3



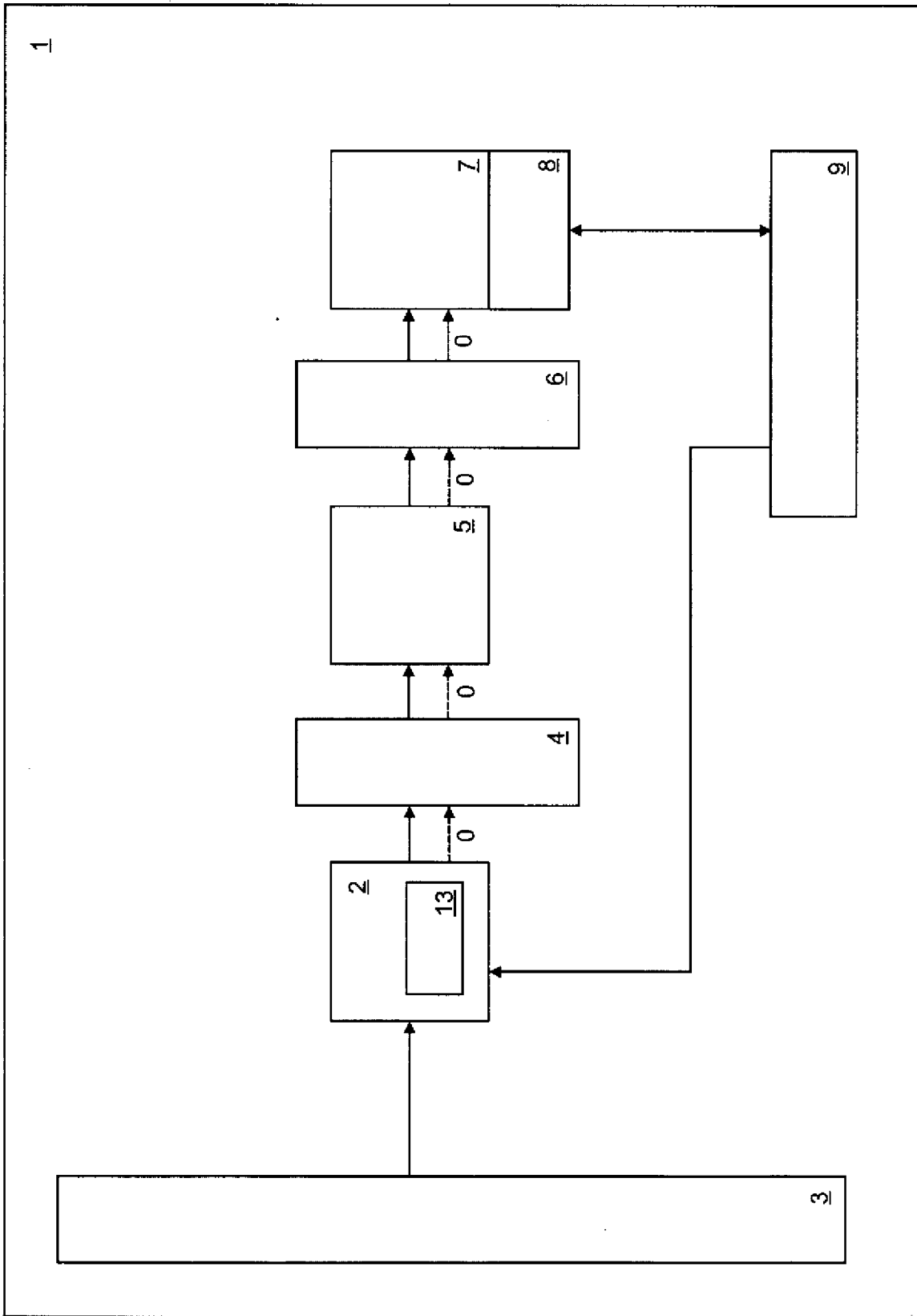


Fig. 4a

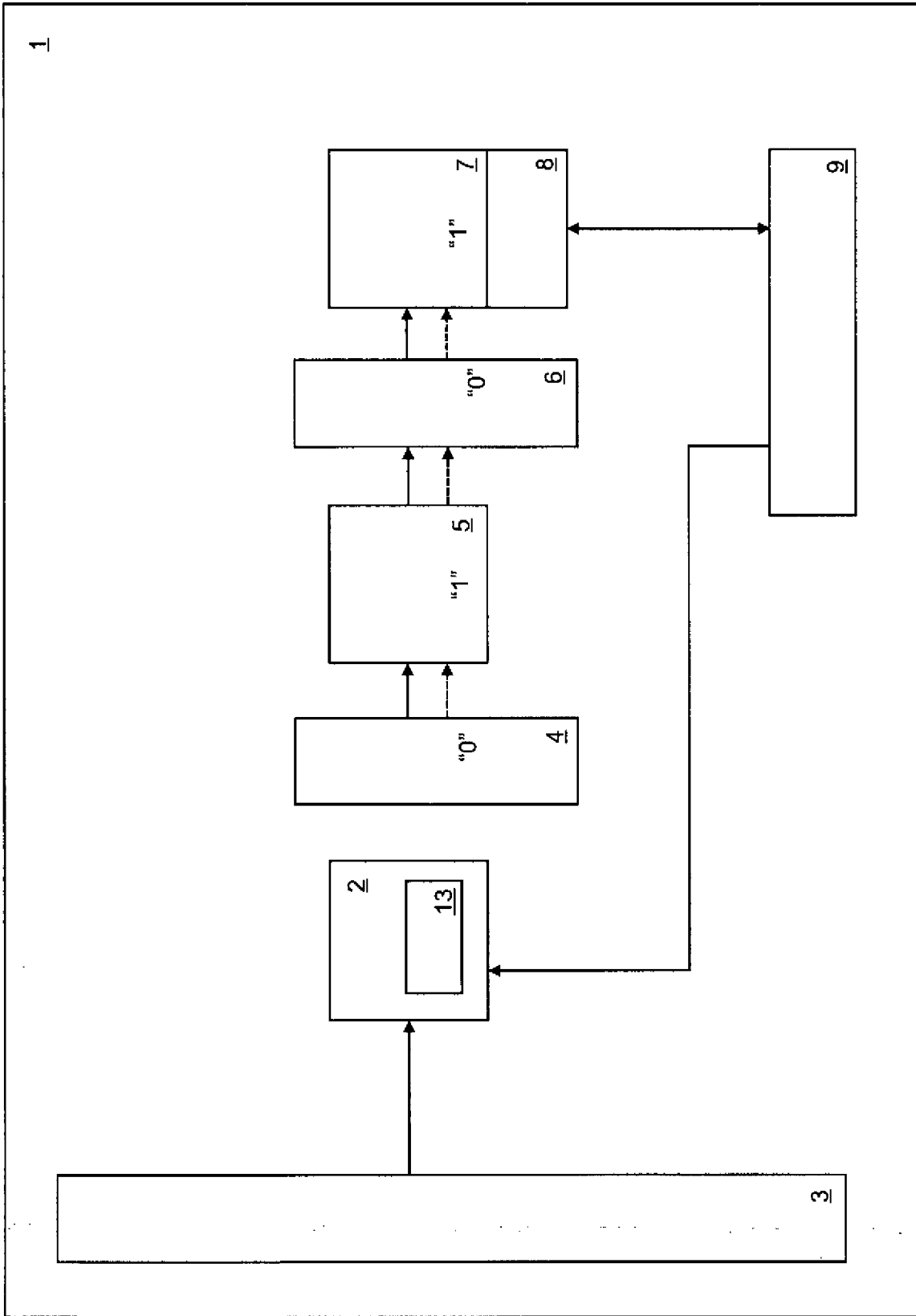


Fig. 4b

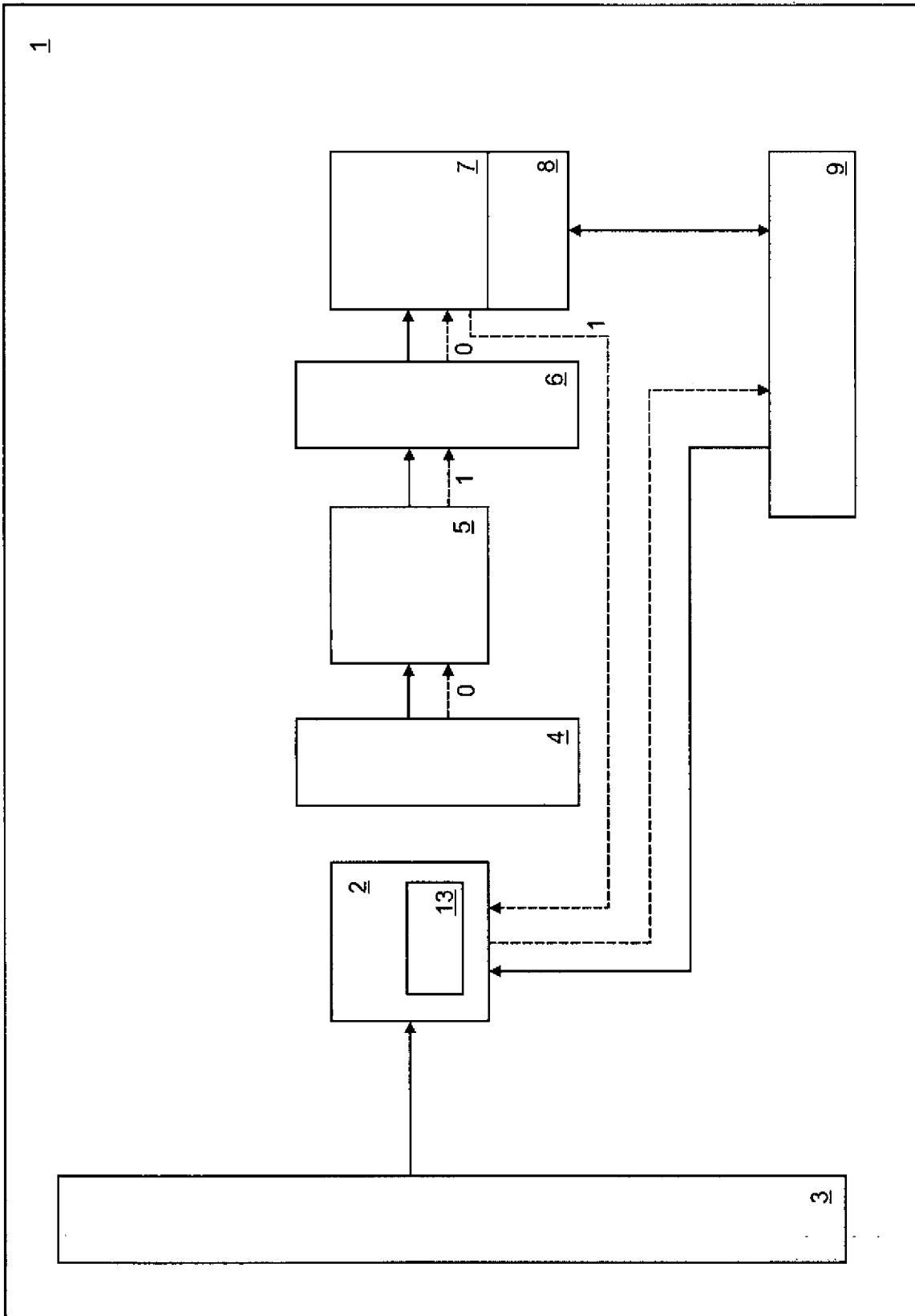


Fig. 4c

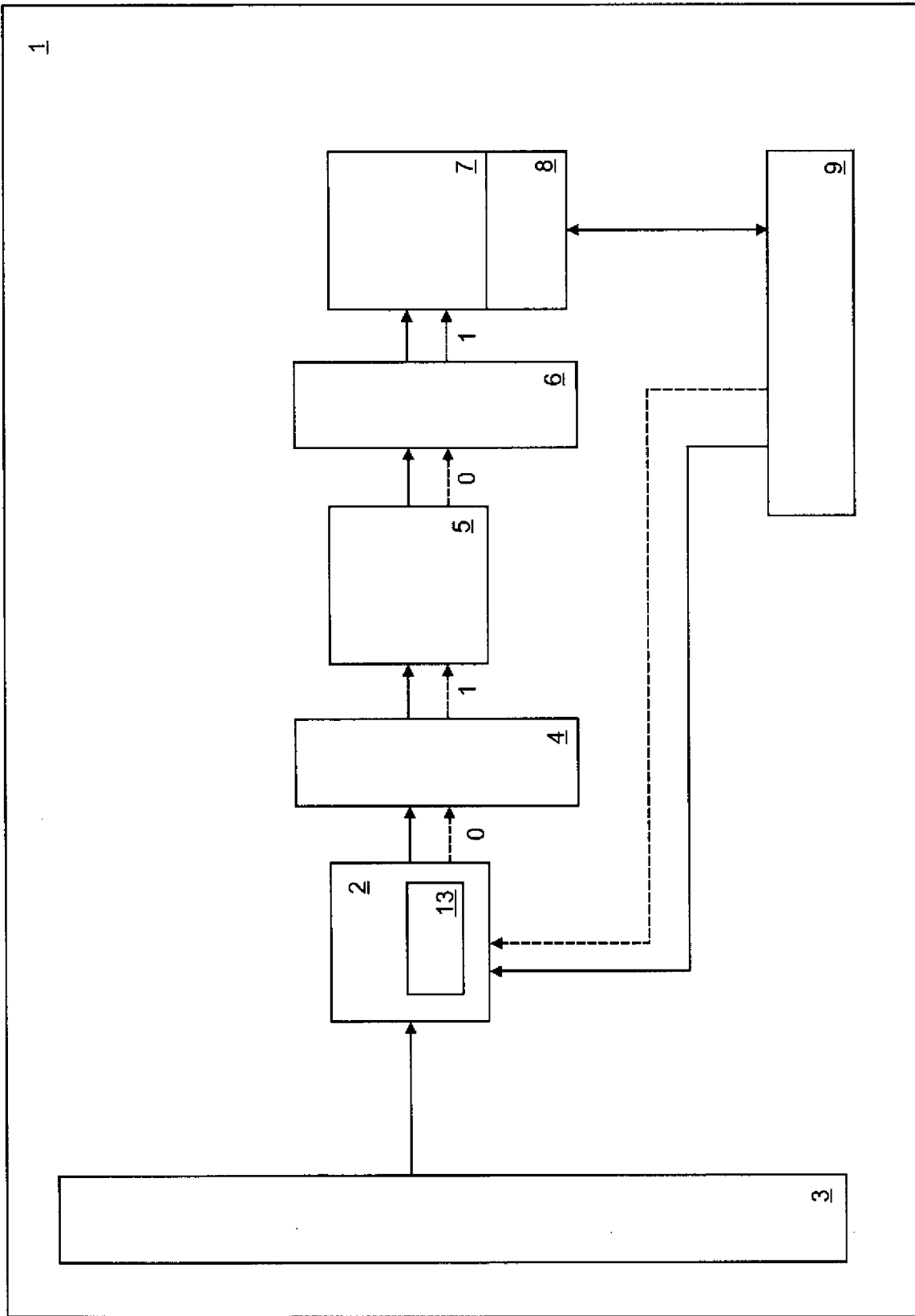


Fig. 4d

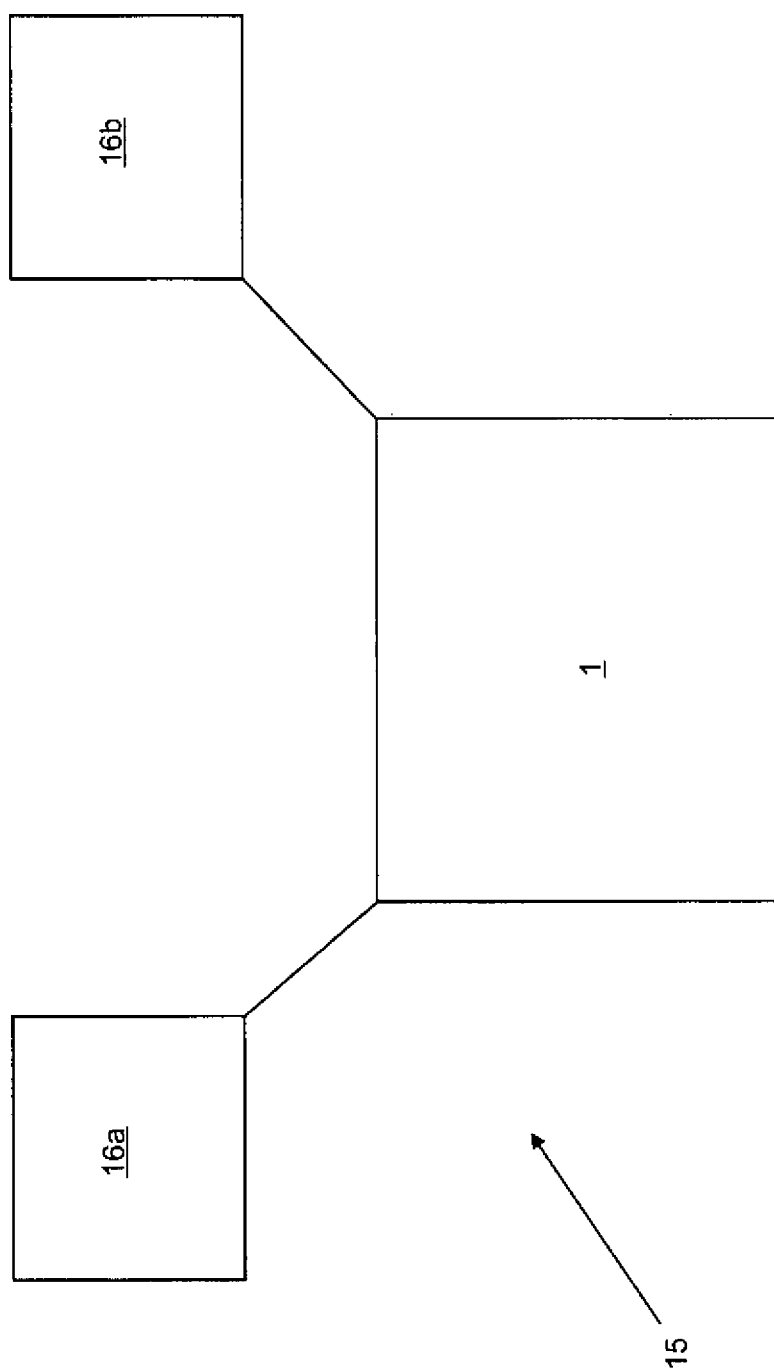


Fig. 5



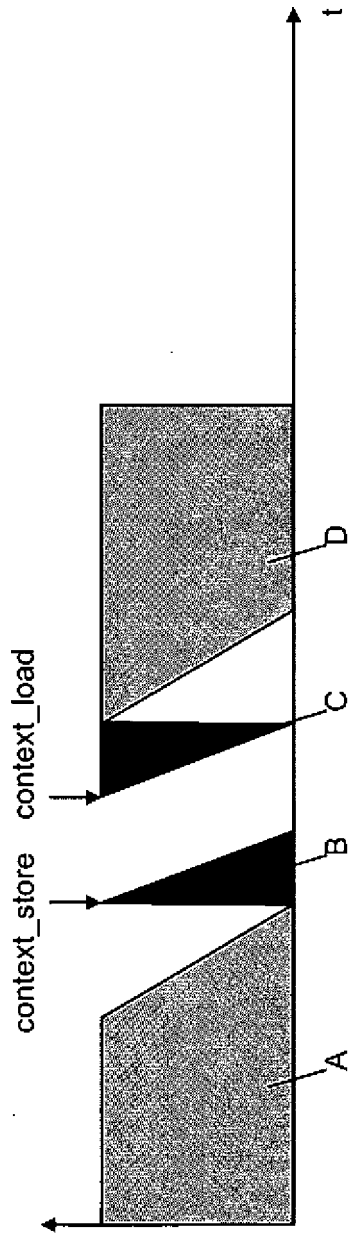


Fig. 6a

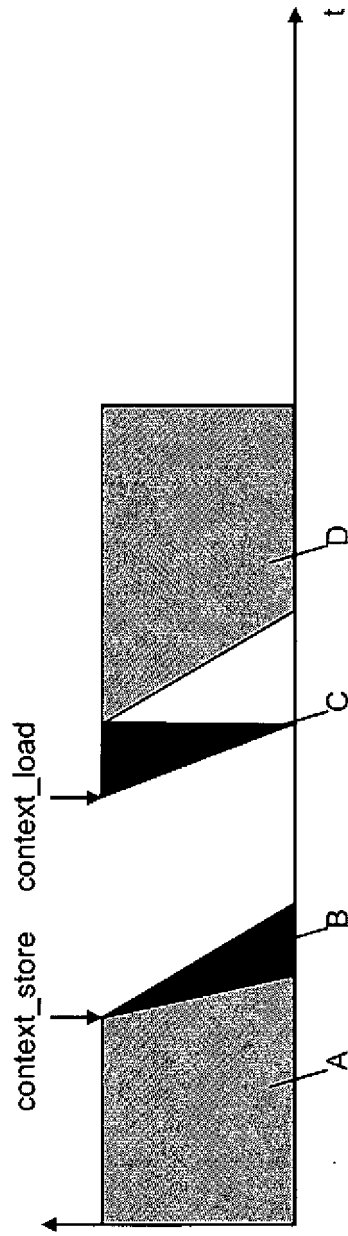


Fig. 6b