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

(10) **DE 698 27 544 T2** 2005.11.10

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 0 867 809 B1**
(21) Deutsches Aktenzeichen: **698 27 544.6**
(96) Europäisches Aktenzeichen: **98 302 259.1**
(96) Europäischer Anmeldetag: **25.03.1998**
(97) Erstveröffentlichung durch das EPA: **30.09.1998**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **17.11.2004**
(47) Veröffentlichungstag im Patentblatt: **10.11.2005**

(51) Int Cl.[7]: **G06F 9/46**
G06T 1/20

(30) Unionspriorität:
7493197 **27.03.1997** **JP**

(73) Patentinhaber:
Sony Computer Entertainment Inc., Tokio/Tokyo, JP

(74) Vertreter:
Mitscherlich & Partner, Patent- und Rechtsanwälte, 80331 München

(84) Benannte Vertragsstaaten:
BE, CH, DE, DK, ES, FI, FR, GB, IT, LI, NL, SE

(72) Erfinder:
Suzuoki, Masakazu, Shinagawa-ku, Tokyo 141, JP

(54) Bezeichnung: **Datenverarbeitungsgerät und -Verfahren**

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

Beschreibung

[0001] Die vorliegende Erfindung bezieht sich auf eine Informationsverarbeitungsvorrichtung sowie auf Informationsverarbeitungsverfahren. Die vorliegende Erfindung betrifft insbesondere eine solche Informationsverarbeitung, bei der Befehle erzeugt und zu einer Ausführungseinrichtung, wie zu einer Zeicheneinheit, entsprechend zugewiesenen Prioritäten übertragen werden (siehe EP-A-0 598 231).

[0002] Mit stärkerem Integrationsgrad von LSI-Schaltungen (Großintegrationsschaltungen), der in den vergangenen Jahren erreicht wurde, kann eine Vielzahl von identischen Schaltungen parallel in einer LSI-Schaltung angeordnet werden, was die dadurch ausgeführte Verarbeitungsgeschwindigkeit zu steigern gestattet. So umfasst beispielsweise ein in den vergangenen Jahren hergestelltes Videospielsystem eine zentrale Verarbeitungseinheit bzw. Zentraleinheit (CPU) und eine Zeicheneinheit. Auf den Empfang von Daten, die ein anzuzeigendes Polygon darstellen, von einer Steuerungseinrichtung her führt die CPU eine zweidimensionale oder dreidimensionale Koordinatentransformation bezüglich der Daten durch und erzeugt eine Liste von Zeichnungsbefehlen, die zum Zeichnen des Polygons in einem zweidimensionalen Bildpuffer herangezogen werden. Die CPU überträgt sodann die Liste, auf die nachstehend Bezug genommen wird, als Anzeigeliste zu der Zeicheneinheit zum Zeichnen des Polygons entsprechend der betreffenden Anzeigeliste.

[0003] Die Auswirkungen einer Verbesserung des Durchsatzes eines Verarbeitungssystems durch Anwendung eines Parallelverarbeitungs verfahrens werden effektiver bei einem System erlangt, bei dem eine Anzahl von einfachen Operationen wiederholt ausgeführt wird, wie im Falle der Zeicheneinheit. Infolgedessen ist die durch die CPU ausgeführte Verarbeitungsgeschwindigkeit relativ niedrig im Vergleich zu einer parallel arbeitenden bzw. verarbeitenden Zeicheneinheit, wie sie in den vergangenen Jahren hergestellt wurde. Um die Menge an dreidimensionalen Modelldaten zu verringern, ist es überdies erforderlich, eine komplexe Graphik, wie eine gekrümmte Fläche, aus einigen wenigen Teilen von Eckpunkt- bzw. Scheitelpunktdaten (Steuerpunkte) jedes Mal dann zu erzeugen, wenn eine Zeichenoperation ausgeführt wird. Es ist die CPU, welche die Last der Verarbeitung für die Erzeugung einer derartigen komplexen Graphik zu tragen hat. Infolgedessen entsteht ein Problem dahingehend, dass die Zeichenverarbeitung nicht mit einem hohen Grad an Effizienz ausgeführt werden kann.

[0004] Angesichts des Ansprechens der oben beschriebenen Probleme ist es ein Ziel zumindest einer bevorzugten Ausführungsform der vorliegenden Erfindung, eine Informationsverarbeitungsvorrichtung und ein Informationsverarbeitungsverfahren bereitzustellen, die bzw. das Daten mit einem hohen Grad an Effizienz dadurch zu verarbeiten erlaubt, dass einer Vielzahl von Zentraleinheiten bzw. CPUs Prioritäten zugewiesen werden, wobei man die CPUs gleichzeitig arbeiten und Anzeigelisten verarbeiten lässt, die durch die CPUs auf einer Prioritätsbasis erzeugt werden, und zwar beginnend mit einer Anzeigeliste, die durch eine eine höchste Priorität besitzende CPU erzeugt wird.

[0005] Gemäß einem Aspekt der Erfindung wird eine Informationsverarbeitungsvorrichtung bereitgestellt, umfassend
eine erste Erzeugungseinrichtung zur Ausführung einer Verarbeitung für die Erzeugung von ersten Befehlen,
eine zweite Erzeugungseinrichtung zur Ausführung einer Verarbeitung für die Erzeugung von zweiten Befehlen, wobei die betreffenden zweiten Befehle eine niedrigere Priorität besitzen als die genannten ersten Befehle und temporär gespeichert werden,
eine Ausführungseinrichtung zur Ausführung der genannten ersten und zweiten Befehle durch Umschalten von einem der genannten ersten Befehle auf einen der genannten zweiten Befehle und umgekehrt mit einer bestimmten Zeitsteuerung,
eine erste Übertragungseinrichtung zur Übertragung der genannten ersten Befehle zu der genannten Ausführungseinrichtung und eine zweite Übertragungseinrichtung zur Übertragung der genannten zweiten Befehle zu der genannten Ausführungseinrichtung,
wobei dann, wenn sich die genannte erste Erzeugungseinrichtung im Zuge der Verarbeitung zur Erzeugung eines der genannten ersten Befehle befindet und die genannte Ausführungseinrichtung sich in einem Wartezustand befindet, die genannte zweite Übertragungseinrichtung einen der genannten gespeicherten zweiten Befehle, falls überhaupt, an die genannte Ausführungseinrichtung abgibt und wobei auf den Empfang des genannten einen Befehls der betreffenden zweiten Befehle hin die genannte Ausführungseinrichtung den genannten einen der betreffenden zweiten Befehle ausführt.

[0006] Gemäß einem weiteren Aspekt der Erfindung wird ein Informationsverarbeitungsverfahren bereitgestellt, welches in einer Informationsverarbeitungsvorrichtung anzuwenden ist, umfassend die Schritte:
Erzeugen eines ersten Befehls unter Heranziehung einer ersten Erzeugungseinrichtung,
Erzeugen eines zweiten Befehls unter Heranziehung einer zweiten Erzeugungseinrichtung,
wobei die betreffenden zweiten Befehle eine niedrigere Priorität besitzen als die genannten ersten Befehle und temporär gespeichert werden,
Ausführen der genannten ersten und zweiten Befehle mittels einer Ausführungseinrichtung durch Umschalten von einem der genannten ersten Befehle auf ei-

nen der genannten zweiten Befehle und umgekehrt mit einer bestimmten Zeitsteuerung,
Übertragender genannten ersten Befehle zu der genannten Ausführungseinrichtung unter Heranziehung einer ersten Übertragungseinrichtung,
und Übertragen der genannten zweiten Befehle zu der genannten Ausführungseinrichtung unter Heranziehung einer zweiten Übertragungseinrichtung,
wobei dann, wenn die genannte erste Erzeugungseinrichtung sich im Zuge der Verarbeitung zur Erzeugung eines der genannten ersten Befehle befindet und die genannte Ausführungseinrichtung sich in einem Wartezustand befindet, die genannte zweite Übertragungseinrichtung einen der genannten gespeicherten zweiten Befehle, falls überhaupt, an die genannte Ausführungseinrichtung abgibt, und wobei auf den Empfang des genannten einen Befehls der genannten zweiten Befehle hin die genannte Ausführungseinrichtung den genannten einen Befehl der genannten zweiten Befehle ausführt.

[0007] Die Erfindung wird nunmehr unter Bezugnahme auf die beigefügten Zeichnungen beispielhaft beschrieben, in denen entsprechende Einzelteile durch entsprechende Bezugszeichen bezeichnet sind. In den Zeichnungen zeigen:

[0008] Fig. 1 eine Draufsicht, die ein typisches Heimunterhaltungssystem veranschaulicht, bei dem eine die vorliegende Erfindung verkörpernde Informationsverarbeitungsvorrichtung angewandt werden kann,

[0009] Fig. 2 ein Diagramm, welches eine Vorderseite des in Fig. 1 dargestellten Heimunterhaltungssystems veranschaulicht,

[0010] Fig. 3 ein Diagramm, welches eine Seitenansicht des in Fig. 1 dargestellten Heimunterhaltungssystems veranschaulicht,

[0011] Fig. 4 eine Draufsicht, die eine typische CD-ROM zeigt, von der Informationen in dem in Fig. 1 dargestellten Heimunterhaltungssystem wiedergegeben werden,

[0012] Fig. 5 ein Blockdiagramm, welches einen typischen inneren elektrischen Aufbau des in Fig. 1 dargestellten Heimunterhaltungssystems veranschaulicht,

[0013] Fig. 6 ein Blockdiagramm, welches einen detaillierten Aufbau von in Fig. 5 dargestellten Einrichtungen zeigt, nämlich einer Haupt-DMAC, einer Haupt-CPU, des Hauptspeichers, einer zweiten Vektorverarbeitungsmaschine (VPE1) und einer GPU,

[0014] Fig. 7A und Fig. 7B in Diagrammen eine Prozedur zur Verarbeitung von Anzeigelisten, die durch eine Vielzahl von Prozessoren erzeugt werden bzw. sind,

[0015] Fig. 8 in einem Blockdiagramm einen weiteren typischen Aufbau des Heimunterhaltungssystems, in welchem drei Prozessoren die GPU-Einheit steuern,

[0016] Fig. 9 in einem Diagramm ein typisches Format eines Meta-Befehls,

[0017] Fig. 10 ein Diagramm, welches zur Erläuterung einer Prozedur zur Übertragung von Daten entsprechend Meta-Befehlen herangezogen wird,

[0018] Fig. 11 ein Diagramm, welches zur Erläuterung einer weiteren Prozedur für die Übertragung von Daten entsprechend Meta-Befehlen herangezogen wird,

[0019] Fig. 12 ein Diagramm, welches zur Erläuterung einer noch weiteren Prozedur für die Übertragung von Daten entsprechend Meta-Befehlen herangezogen wird,

[0020] Fig. 13 ein Diagramm, welches zur Erläuterung einer Prozedur für die Übertragung von Daten entsprechend einer Anzeigeliste herangezogen wird, und

[0021] Fig. 14 ein Diagramm, welches zur Erläuterung einer Boxsteuerung herangezogen wird.

[0022] Fig. 1 bis Fig. 3 veranschaulichen jeweils in einem Diagramm ein typisches Heimunterhaltungssystem, bei dem eine die vorliegende Erfindung verkörpernde Informationsverarbeitungsvorrichtung angewandt werden kann. Wie dargestellt, umfasst das Heimunterhaltungssystem eine Unterhaltungssystem-Haupteinheit **2** zusätzlich zu einer Bedienungseinheit **17** und einer Aufzeichnungseinheit **38**, die mit der Unterhaltungssystem-Haupteinheit **2** verbunden werden kann.

[0023] Wie in Fig. 1 bis Fig. 3 veranschaulicht, weist die Unterhaltungssystem-Haupteinheit **2** eine nahezu rechteckige bzw. quadratische Form auf. Die Unterhaltungssystem-Haupteinheit **2** umfasst eine Diskaufnahme-Subeinheit **3** zur Aufnahme bzw. Unterbringung einer CD-ROM **40** (das ist ein Compaktdisk-Festwertspeicher), die in ihrer Mitte untergebracht ist bzw. wird, einen Rücksetzschalter **4**, der sich an einer geeigneten Stelle in bzw. an der Unterhaltungssystem-Haupteinheit **2** zur Nutzung durch den Anwender befindet, um eine laufende Anwendung beliebig zurückzusetzen, einen Spannungsversorgungs- bzw. Netzschalter **5** zur Nutzung durch den Anwender, und zwar zum Einschalten und Ausschalten einer Spannungsversorgung, einen Disk-Bedienungsschalter **6** für die Nutzung durch den Anwender, um eine Disk in der Diskaufnah-

me-Subeinheit **3** unterzubringen, sowie Anschlüsse bzw. Anschlussbuchsen **7A** und **7B** auf der linken bzw. rechten Seite für die Nutzung durch den Anwender, um die Unterhaltungssystem-Haupteinheit **2** mit der Bedienungseinheit **17** zu verbinden, die zur Ausführung von Operationen genutzt wird, während eine Anwendung läuft, und mit der Aufzeichnungseinheit **38** zur Aufzeichnung der Einstellung verschiedener Informationsarten der laufenden Anwendung und dergleichen zu verbinden. Es sei darauf hingewiesen, dass die CD-ROM eine Art einer optischen Platte ist, deren eine in Fig. 4 gezeigt ist. Die CD-ROM ist eine Disk, die als Aufzeichnungsträger einer laufenden Anwendung genutzt wird.

[0024] Wie in Fig. 2 und Fig. 3 gezeigt, sind die Anschlüsse bzw. Verbinder **7A** und **7B** jeweils in zwei Ebenen angeordnet. In einer oberen Ebene der Verbinder **7A** und **7B** ist jeweils ein Aufzeichnungs-Einführungsbereich **8** zur Verbindung der Unterhaltungssystem-Haupteinheit **2** mit der Aufzeichnungseinheit **38** vorgesehen. In der unteren Ebene der Verbinder **7A** und **7B** ist jeweils auf der anderen Seite ein Verbinderstift-Einführungsbereich **12** zur Verbindung der Unterhaltungssystem-Haupteinheit **2** mit der Bedienungseinheit **17** vorgesehen.

[0025] Der Aufzeichnungs-Einführungsbereich **8** umfasst ein horizontales langes rechteckförmiges Einführungsloch und einen Speicheranschluss, in den die Aufzeichnungseinheit **38** eingeführt wird. Der Speicheranschluss ist innerhalb des Loches untergebracht und in der Zeichnungsfigur nicht dargestellt. Ist die Aufzeichnungseinheit **38** nicht mit der Unterhaltungssystem-Haupteinheit **2** verbunden, ist der Aufzeichnungs-Einführungsbereich **8** von einem Verschluss **9** abgedeckt, um den Speicheranschluss vor Staub und dergleichen zu schützen, wie dies in Fig. 2 gezeigt ist. Es sei darauf hingewiesen, dass die Aufzeichnungseinheit **38** einen elektrisch programmierbaren ROM-Speicher enthält, in welchem die Haupt-CPU **44** Daten einer Anwendungs-Software aufzeichnet.

[0026] Wenn die Aufzeichnungseinheit **38** an der Unterhaltungssystem-Haupteinheit **2** angebracht wird, drückt der Anwender den Verschluss **9** zur Innenseite des Aufzeichnungs-Einführungsbereiches **8** mit dem Ende der Aufzeichnungseinheit **38** und unter weiterer Einführung der Aufzeichnungseinheit **38** in das Einführungsloch auf, bis die Aufzeichnungseinheit **38** mit dem Speicheranschluss in Verbindung gelangt.

[0027] Wie in Fig. 2 veranschaulicht, umfasst der Verbinderstift-Einführungsbereich **12** ein horizontales langes rechteckförmiges Einführungsloch und einen Verbinderanschluss **12A** zur Verbindung des Verbinder-Einführungsbereichs **12** mit dem Verbinder-Anschlussbereich **26** der Bedienungseinheit **17**.

[0028] Wie in Fig. 1 gezeigt, weist die Bedienungseinheit **17** einen Aufbau auf, der von und zwischen den beiden Händen des Anwenders gehalten und durch unbehinderte Bewegungen der fünf Finger der jeweiligen Hand bedient werden kann. Die Bedienungseinheit **17** weist Bedienungs-Untereinheiten **18** und **19**, die symmetrisch auf den rechten und linken Seiten vorgesehen sind, einen Auswahlschalter **22** und einen Startschalter **23**, die zwischen den Bedienungs-Untereinheiten **18** und **19** vorgesehen sind, Bedienungs-Untereinheiten **24** und **25**, die vor den Bedienungs-Untereinheiten **18** bzw. **19** liegen, und einen Verbinder **26** sowie ein Kabel **27** zur Verbindung der Bedienungseinheit **17** mit der Unterhaltungssystem-Haupteinheit **2** auf.

[0029] Fig. 5 veranschaulicht in einem Diagramm einen typischen inneren elektrischen Aufbau der Unterhaltungssystem-Haupteinheit **2**. Wie in der betreffenden Zeichnungsfigur veranschaulicht, weist die Unterhaltungssystem-Haupteinheit **2** einen Hauptbus **41** und einen Sub- bzw. Unterbus **42** auf, wobei die beiden Busse bzw. Busleitungsanordnungen durch eine Subbus-Schnittstelle (SBUSIF) **43** miteinander verbunden sind.

[0030] Mit dem Hauptbus **41** sind eine Haupt-CPU **44** (Zentraleinheit bzw. zentrale Verarbeitungseinheit), (eine Listenerzeugungseinrichtung), die durch Komponenten, wie einen Mikroprozessor und eine VPEO-Maschine **71** (eine erste Vektor-Verarbeitungsmaschine) implementiert bzw. realisiert ist, ein durch einen RAM-Speicher mit wahlfreiem Zugriff implementierter Hauptspeicher **45**, eine Haupt-DMAC-Steuereinrichtung **46** (Haupt-Direktspeicherzugriffs-Steuereinrichtung), (eine Datenübertragungseinrichtung), ein MDEC-Decoder **47** (MPEG-(Bewegtbild-Expertengruppe-)-Decoder) und eine VPE1-Maschine **48** (eine zweite Vektor-Verarbeitungsmaschine) verbunden. Außerdem ist eine GPU-Einheit **49** (eine graphische Verarbeitungseinheit) ebenfalls mit dem Hauptbus **41** verbunden, und zwar über eine GPUIF-Schnittstelle **72** (graphische Verarbeitungseinheit-Schnittstelle). An bzw. in der GPU-Einheit **49** ist eine CRTC-Steuereinrichtung **84** (Kathodenstrahlröhren-Steuereinrichtung) vorgesehen. Überdies ist mit der GPU-Einheit **49** ein Bild- bzw. Vollbildspeicher **58** verbunden.

[0031] Andererseits sind an dem Sub- bzw. Unterbus **42** eine Unter- bzw. Sub-CPU **50**, die durch Komponenten, wie einen Mikroprozessor implementiert bzw. realisiert ist, ein Sub-Speicher **51**, der durch einen RAM-Speicher implementiert ist, eine Sub-DMAC-Steuereinrichtung **52**, ein ROM-Speicher **53** zur Speicherung von Programmen, wie eines Betriebssystems, eine SPU-Einheit **54** (Tonverarbeitungseinheit), eine Kommunikationssteuereinheit (ATM) **55**, ein CD-ROM-Laufwerk **56**, das auch als die früher zitierte Disk-Aufnahme-Subeinheit **3** dient,

und eine Eingabeeinheit **57** angeschlossen. Der Verbinderanschluss **12A** der Eingabeeinheit **57** ist mit dem Verbinder-Anschlussbereich **26** der Bedienungseinheit **17** verbunden, wie dies früher beschrieben worden ist.

[0032] Die mit dem Hauptbus **41** und dem Subbus **42** verbundene Schnittstelleneinrichtung SBUSIF **43** leitet von dem Hauptbus **41** kommende Daten zum Subbus **42** weiter, und sie leitet demgegenüber Daten, die vom Subbus **42** herkommen, zum Hauptbus **41** weiter.

[0033] Wenn die Unterhaltungssystem-Haupteinheit **2** aktiviert ist, holt die Haupt-CPU **44** Befehle eines Aktivierungsprogramms aus dem mit dem Subbus **42** verbundenen ROM-Speicher **53** mittels der Schnittstelleneinrichtung SBUSIF **43** ab und führt die Befehle des Aktivierungsprogramms aus, um das Bedienungssystem zu aktivieren.

[0034] Darüber hinaus gibt die Haupt-CPU **44** eine Anforderung zum Lesen von Daten an das CD-ROM-Laufwerk **56** ab, um Daten und ein Anwendungsprogramm von einer CD-ROM **40** zu erlangen, die im CD-ROM-Laufwerk **56** untergebracht ist, und um das Anwendungsprogramm in den Hauptspeicher **45** zu laden.

[0035] Ferner erzeugt die Haupt-CPU **44** in Verbindung mit der ersten Vektor-Verarbeitungsmaschine (VPE0) **71** Daten für eine nichttypische Verarbeitung, d.h. für eine Polygon-Festlegungsinformation aus Daten eines dreidimensionalen Objekts, welches eine Vielzahl von Grundfiguren, wie Polygone, umfasst, die aus der CD-ROM **40** ausgelesen werden. Ein Beispiel der Daten eines dreidimensionalen Objekts sind Koordinatenwerte von Eckpunkten oder repräsentativen Punkten eines Polygons. Die Vektor-Verarbeitungsmaschine VPE0 (die erste Vektor-Verarbeitungsmaschine) **71** weist eine Vielzahl von Verarbeitungselementen zur Verarbeitung von reellen Gleitkommazahlen auf und ist somit imstande, Teile einer Gleitkommaverarbeitung parallel auszuführen.

[0036] Um ins Detail zu gehen, führen die Haupt-CPU **44** und die erste Vektor-Verarbeitungsmaschine **71** (VPE0) eine Geometrieverarbeitung aus, mit der detaillierte Operationen in Polygoneinheiten verbunden sind. Ein Beispiel einer derartigen Verarbeitung ist die Erzeugung von Daten eines Polygons, welches einen schwingenden Zustand eines Blattes eines vom Wind angeblasenen Baumes darstellt, oder einen Zustand von Regentropfen, die auf die Frontscheibe eines Autos auftreffen. Sodann werden die Eckpunktinformationen, die aus der Verarbeitung ermittelt sind, und die Polygon-Festlegungsinformationen, wie eine Übergangsbetrieb-Information, dem Hauptspeicher **45** als Pakete über den Hauptbus **41** zugeführt.

[0037] Die Polygon-Festlegungsinformation umfasst eine Zeichenbereich-Festlegungsinformation und eine Polygon-Information. Die Zeichenbereich-Festlegungsinformation enthält Versatz-Koordinaten im Bildspeicher **58** eines Zeichenbereiches, d.h. eine Bildspeicheradresse des Zeichenbereichs, und Koordinaten eines Zeichen-Abschneidebereichs zum Löschen einer Operation, um einen Zeichenbereich zu zeichnen, der durch ein Polygon bestimmt ist, dessen Koordinaten außerhalb des Zeichenbereichs liegen. Andererseits enthält die Polygon-Information eine Polygon-Attributinformation und eine Eckpunkt- bzw. Eckeninformation. Hier stellt die Polygon-Attributinformation eine Information dar, die zur Spezifizierung eines Übergangsbetriebs, eines ALPHA-Mischbetriebs und eines Textur-Abbildungsbetriebs herangezogen wird. Andererseits stellt die Eckpunkt- bzw. Eckeninformation eine Information hinsichtlich der Koordinaten in einem Eckpunkt-Zeichenbereich, Koordinaten in einem Eckpunkt-Texturbereich und die Farbe eines Eckpunkts dar, um einige wenige zu nennen.

[0038] Wie die erste Verarbeitungsmaschine (VPE0) **71** so weist auch die zweite Vektor-Verarbeitungsmaschine (VPE1) **48** eine Vielzahl von Verarbeitungselementen zur Verarbeitung von reellen Gleitkommazahlen auf, und sie ist somit imstande, Teile einer Gleitkommaverarbeitung parallel auszuführen. Die VPE1-Maschine **48** ist imstande, ein Bild entsprechend Operationen, die durch Heranziehen der Bedienungseinheit **17** ausgeführt werden, und entsprechend Matrixoperationen zu erzeugen. Dies bedeutet, dass die zweite Vektor-Verarbeitungsmaschine (VPE1) **48** Daten (eine Polygon-Festlegungsinformation) für die Typ-Verarbeitung erzeugt, d.h. für die Verarbeitung, die relativ einfach genug ist, um durch Ausführen eines Programms auf der VPE1-Maschine **48** ausgeführt zu werden. Beispiele einer derartigen Verarbeitung, die durch die zweite Vektor-Verarbeitungsmaschine (VPE1) **48** ausgeführt werden, sind eine Radius-Kopierumsetzung für ein Objekt mit einer einfachen Form, wie ein Gebäude oder ein Auto, eine parallele Lichtquellenberechnung und die Erzeugung einer zweidimensional gekrümmten Fläche. Sodann wird die durch die VPE1-Maschine **48** erzeugte Polygon-Festlegungsinformation an die Schnittstelleneinrichtung GPUIF **72** abgegeben.

[0039] Die Schnittstelleneinrichtung GPUIF **72** empfängt, durch die Haupt-CPU **44** gesteuert, außerdem eine Polygon-Festlegungsinformation von dem Hauptspeicher **45** über den Hauptbus **41**. Aus diesem Grunde stellt die Schnittstelleneinrichtung GPUIF **72** die Verarbeitungszeit so ein, dass verhindert ist, dass die Polygon-Festlegungsinformation, die von der Haupt-CPU **44** herstammt, mit der Poly-

gon-Festlegungsinformation kollidiert, welche ihr von der zweiten Vektor-Verarbeitungsmaschine **48** zugeführt wird, wobei die betreffenden Informationen zu der GPU-Einheit **49** hin geleitet werden.

[0040] Die GPU-Einheit **49** zeichnet ein Bild, welches ein dreidimensionales Objekt ausdrückt, unter Verwendung eines Polygons auf der Grundlage der Polygon-Festlegungsinformation, die ihr über die Schnittstelleneinrichtung GPUIF **72** von dem Bildspeicher **58** zugeführt wird. Ein Bild, welches unter Heranziehung eines Polygons zur Darstellung eines dreidimensionalen Objekts gezeichnet ist, wird nachstehend als Polygonbild bezeichnet. Da die GPU-Einheit **49** im Stande ist, den Bildspeicher **58** auch als Textur-Speicher zu nutzen, kann die GPU-Einheit **49** eine Textur-Abbildungsverarbeitung ausführen, um ein Pixelbild in dem Bildspeicher **58** an ein Polygon als Textur anzuhängen.

[0041] Die Haupt-DMAC-Steuereinrichtung **46** wird zur Steuerung herangezogen, und zwar unter anderem für Operationen bezüglich DMA-Übertragungen zu und von einer Vielzahl von mit dem Hauptbus **41** verbundenen Schaltungen. Darüber hinaus ist die Haupt-DMAC-Steuereinrichtung **46** in Abhängigkeit vom Zustand der Schnittstelleneinrichtung SBUSIF **43** ferner imstande, unter anderem Operationen bezüglich DMA-Übertragungen zu und von einer Vielzahl von mit dem Subbus **42** verbundenen Schaltungen zu steuern. Die MDEC-Steuereinrichtung **47** arbeitet gleichzeitig mit der Haupt-CPU **44** und dekomprimiert Daten, die entsprechend einem MPEG-(Bewegtbild-Expertengruppe-)-System oder entsprechend einem JPEG-System (vereinte Photographische Expertengruppe) komprimiert worden sind.

[0042] Die Sub-CPU **50** führt verschiedene Arten der Verarbeitung durch Ausführen von im ROM-Speicher **53** gespeicherten Programmen aus. Die Sub-DMAC-Steuereinrichtung **52** steuert unter anderem Operationen bezüglich DMA-Übertragungen zu und von einer Vielzahl von mit dem Subbus **42** verbundenen Schaltungen lediglich dann, wenn die Schnittstelleneinrichtung SBUSIF **43** vom Hauptbus **41** und vom Subbus **42** getrennt ist.

[0043] Die SPU-Einheit **54** liest Ton- bzw. Schalldaten aus einem Ton- bzw. Schallspeicher **59** aus und gibt die betreffenden Ton-Daten als Audiosignal entsprechend einem Ton-Befehl ab, der von der Sub-CPU **50** oder der Sub-DMAC-Steuereinrichtung **52** empfangen wird. Das ausgangsseitige Audiosignal wird dann über eine Verstärkerschaltung **201** an einen Lautsprecher **202** abgegeben, um von dem Lautsprecher **202** schließlich als Schall abgegeben zu werden.

[0044] Die Kommunikationssteuereinheit (ATM) **55** ist mit einer öffentlichen Kommunikations- bzw. Übertragungsleitung oder dgl. verbunden, und sie wird zur Übertragung und zum Empfang von Daten über die betreffende Leitung genutzt.

[0045] Die Eingabeeinheit **57** umfasst den Verbinderanschluss **12A** für den Anschluss der Bedienungseinheit **17** an der Unterhaltungssystem-Haupteinheit **2**, eine Video-Eingabeschaltung **82** zur Zuführung von Videodaten, die von anderen, in der Zeichnungsfigur nicht dargestellten Vorrichtungen bzw. Geräten herkommen, zu der Unterhaltungssystem-Haupteinheit **2** und eine Audio-Eingabeschaltung **83** zur Zuführung von Audiodaten, die von anderen Vorrichtungen bzw. Geräten herkommen, zu der Unterhaltungssystem-Haupteinheit **2**.

[0046] Fig. 6 veranschaulicht in einem Blockdiagramm einen detaillierten Aufbau von in Fig. 5 dargestellten Einrichtungen, und zwar der Haupt-DMAC-Steuereinrichtung **46**, der Haupt-CPU **44**, des Hauptspeichers **45**, der zweiten Vektor-Verarbeitungsmaschine (VPE1) **48** und der GPU-Einheit **49**.

[0047] Wie in Fig. 6 gezeigt, umfasst die Haupt-CPU **44** einen CPU-Kern (KERN) **94**, einen Befehls-Cache-Speicher (I$) **95**, einen Notizblock-RAM-Speicher (SPR) **96**, einen Daten-Cache-Speicher (D$) **97** und die erste Vektor-Verarbeitungsmaschine (VPE0) **71**. Der CPU-Kern **94** führt bestimmte Befehle aus. Der Befehls-Cache-Speicher **95** wird zur vorübergehenden Speicherung von Befehlen genutzt, die dem CPU-Kern **94** zuzuführen sind. Der Notizblock-RAM-Speicher **96** wird zur Speicherung von Ergebnissen der Verarbeitung genutzt, die durch den CPU-Kern **94** ausgeführt werden. Schließlich wird der Daten-Cache-Speicher **97** zur temporären Speicherung von Daten genutzt, die bei der Ausführung der Verarbeitung durch den CPU-Kern **94** verwendet werden.

[0048] Die erste Vektor-Verarbeitungsmaschine (VPE0) **71** umfasst einen Mikrospeicher (microMEM) **98**, eine FMAC-(Gleitkomma-Vielfach-Addierer-Berechnungs-)-Einheit **99**, eine Dividier- bzw. Teiler-(DIV)-Einheit **100**, eine funktionale Einheit **101**, die als VU-MEM bezeichnet ist, und eine Paket-Dehnungseinrichtung (PKE) **102**. Die Einheit VU-MEM **101** enthält eine Gleitkomma-Vektor-Prozessoreinheit (VU) und einen eingebetteten Speicher (MEM). Die Gleitkomma-Vektor-Prozessoreinheit führt 64-Bit-Mikrobefehle eines in dem Mikrospeicher **98** gespeicherten Mikroprogramms aus, das später zu beschreiben ist, um in internen Registern der Prozessoreinheit VU und im eingebetteten Speicher gespeicherte Daten zu verarbeiten.

[0049] Die Paket-Dehnungseinheit PKE **102** dehnt einen ihr zugeführten Mikrocode entsprechend der durch eine Haupt-DMAC-Steuereinrichtung **109** aus-

geführten Steuerung, die weiter unten zu beschreiben ist, zu Mikrobefehlen, die als Mikroprogramm im Mikrospeicher **98** zu speichern sind und die durch die Prozessoreinheit VU auszuführen sind, und die betreffende Einrichtung dehnt ein Paket von Paketdaten, das ihr ebenfalls zugeführt ist, entsprechend der durch die Haupt-DMAC-Steuereinrichtung **109** ausgeführten Steuerung und speichert das gedehnte Paket in dem eingebetteten Speicher (MEM), der in der VU-MEM-Einrichtung **101** verwendet ist. Die FMAC-(Gleitkomma-Mehrfach-Addierer-Berechnungs-)-Einheit **99** führt eine Gleitkomma-Verarbeitung aus, während die Dividiereinrichtung (DIV) **100** eine Division ausführt. Wie oben beschrieben, ist die erste Vektor-Verarbeitungsmaschine (VPE0) **71** in der Haupt-CPU **44** eingebettet, die eine nicht-typische Verarbeitung in Verbindung mit der Verarbeitungsmaschine VPE0 **71** ausführt.

[0050] Wie die erste Vektor-Verarbeitungsmaschine (VPE0) **71** so umfasst auch die zweite Vektor-Verarbeitungsmaschine (VPE1) **48** einen Mikrospeicher (microMEM) **103**, eine FMAC-(Gleitkomma-Mehrfach-Addierer-Berechnungs-)-Einheit **104**, eine Divisionseinrichtung (DIV) **106**, eine als VU-MEM-Einrichtung bezeichnete funktionale Einheit **107** und eine Paket-Dehnungseinrichtung (PKE) **108**. Die Einheit VU-MEM **107** enthält eine Gleitkomma-Vektor-Prozessoreinheit (VU) und einen eingebetteten Speicher (MEM). Die Gleitkomma-Vektor-Prozessoreinheit führt 64-Bit-Mikrobefehle eines in dem Mikrospeicher (microMEM) **103** gespeicherten Mikroprogramms aus, was später zu beschreiben ist, um in internen Registern der Einheit VU und im eingebetteten Speicher gespeicherte Daten zu verarbeiten.

[0051] Die Paket-Dehnungseinrichtung PKE **108** dehnt den ihr zugeführten Mikrocode entsprechend der Steuerung, welche durch die Haupt-DMAC-Steuereinrichtung **46** ausgeführt wird, zu Mikrobefehlen, die als Mikroprogramm im Mikrospeicher **103** zu speichern und durch die Verarbeitungseinheit VU auszuführen sind, und sie dehnt ein Paket von paketierten Daten, das ihr ebenfalls zugeführt ist, entsprechend der Steuerung, welche durch die Haupt-DMAC-Steuereinrichtung **46** ausgeführt wird, und sie speichert das gedehnte Paket in dem in der Einrichtung VU-MEM **107** verwendeten eingebetteten Speicher (MEM). Die FMAC-(Gleitkomma-Mehrfach-Addierer-Berechnungs-)-Einheit **104** führt eine Gleitkommaverarbeitung durch, während die Divisions- bzw. Dividiereinrichtung (DIV) **106** eine Division ausführt. Die zweite Vektor-Verarbeitungsmaschine **48** führt eine Typen-Verarbeitung bezüglich der ihr vom Hauptspeicher **45** zugeführten Daten aus und liefert die Ergebnisse der Verarbeitung über die GPU-Einheit **49** an die Schnittstelleneinrichtung GPUIF **72**.

[0052] Der Hauptspeicher **45** wird zur Speicherung von Daten eines dreidimensionalen Objekts verwendet, und er gibt, wenn dies notwendig ist, die Daten an die erste Vektor-Verarbeitungsmaschine **71** und an die zweite Vektor-Verarbeitungsmaschine **48** ab. Eine gemeinsam durch die Haupt-CPU **44** und die erste Vektor-Verarbeitungsmaschine (VPE0) **71** erzeugte Anzeigeliste wird temporär in einem Speicher FIFO (MFIFO), der im Hauptspeicher **45** eingebettet ist, gespeichert, bevor sie über den Hauptbus **41** an die Schnittstelleneinrichtung GPUIF **72** abgegeben wird. Der Grund dafür, warum die Anzeigeliste im Speicher FIFO temporär gespeichert wird, liegt darin, dass die Haupt-CPU **44** und die erste Vektor-Verarbeitungsmaschine **71** jeweils eine niedrigere Verarbeitungspriorität aufweisen als die zweite Vektor-Verarbeitungsmaschine **48**, wodurch es notwendig wird, die Anzeigeliste im Speicher FIFO so lange zu behalten, bis die zweite Vektor-Verarbeitungsmaschine **48** in einen Ruhezustand eintritt.

[0053] Darüber hinaus erzeugen die Haupt-CPU **44** und die erste Vektor-Verarbeitungsmaschine (VPE0) **71** gemeinsam eine durch die zweite Vektor-Verarbeitungsmaschine **48** zu verarbeitende Matrix und sie speichern die Matrix im Hauptspeicher **45**. Sodann bildet die zweite Vektor-Verarbeitungsmaschine **48** eine Anzeigeliste unter Heranziehung der Matrix.

[0054] Um eine von der ersten Vektor-Verarbeitungsmaschine **71** über die Schnittstelleneinrichtung GPUIF **72** zugeführte Anzeigeliste für eine nicht-typische Verarbeitung und eine Anzeigeliste für eine typischer Verarbeitung, die von der zweiten Vektor-Verarbeitungsmaschine **48** zugeführt ist, zu verarbeiten, enthält die GPU-Einheit **49** einen graphischen Kontext (das ist ein Zeicheneinstellzustand) von unter anderem einem Zeichen-Versatz und einem Abschneidebereich, worauf zu einer Zeichenzeit für jede der Anzeigelisten Bezug zu nehmen ist. Eine in der folgenden Beschreibung benutzte Angabe CG0 bezeichnet einen graphischen Kontext für eine nicht-typische Verarbeitung, während eine Angabe CG1 einen graphischen Kontext für eine typischer Verarbeitung angibt, wie dies weiter unten beschrieben wird.

[0055] Wie oben beschrieben, dehnt die Paket-Dehnungseinrichtung PKE **102** beispielsweise einen an die erste Vektor-Verarbeitungsmaschine **71** von dem Hauptspeicher **45** über den Hauptbus **41** abgegebenen Mikrocode entsprechend einer Steuerung, die durch die DMAC-Steuereinrichtung **109** ausgeführt wird, zu Mikrobefehlen, welche als Mikroprogramm im Mikrospeicher **98** zu speichern und durch die VU-Einheit auszuführen sind, und die betreffende Einrichtung dehnt ein Paket aus paketierten Daten, wie Daten eines dreidimensionalen Objekts, welches ihr ebenfalls von dem Hauptspeicher **45** über den Hauptbus **41** zugeführt ist, entsprechend der Steuerung, die durch die DMAC-Steuereinrichtung **109** ausgeführt wird, und sie speichert das gedehnte Paket in dem eingebetteten Speicher (MEM), der in der

Einrichtung VU-MEM **101** verwendet ist. Sodann führen die FMAC-Einheit **99** und die Dividiereinrichtung DIV **100** Teile der Verarbeitung, wie eine Matrix-Verarbeitung, eine Transformation von Koordinaten und eine Radius-Kopierumsetzung bezüglich der Daten des dreidimensionalen Objekts durch. Zu diesem Zeitpunkt wird außerdem eine komplexe Verarbeitung in Verbindung mit dem CPU-Kern **94** ausgeführt. Die Verarbeitung führt in typischer Weise zu einer Anzeigeliste zum Zeichnen eines schwingenden Zustands eines Blattes eines vom Wind angeblasenen Baumes oder eines Zustands eines Regentropfens, der auf die Frontscheibe eines Autos auftrifft.

[0056] Eine Anzeigeliste (komplexer Strom) zum Zeichnen eines auf diese Weise erzeugten zweidimensionalen Objekts auf einem Anzeigeschirm wird temporär im Speicher MFIFO des Hauptspeichers **45** über den Hauptbus **41** gespeichert, bevor sie schließlich an die Schnittstelleneinrichtung GPUIF **72** abgegeben wird.

[0057] Andererseits dehnt, wie oben beschrieben, die Dehnungseinrichtung PKE **108** den an die zweite Vektor-Verarbeitungsmaschine **48** von dem Hauptspeicher **45** über den Hauptbus **41** abgegebenen Mikrocode entsprechend der in der Haupt-DMAC-Steuereinrichtung **46** ausgeführten Steuerung zu Mikrobefehlen, die als Mikroprogramm im Mikrospeicher **103** zu speichern und durch die Einrichtung VU auszuführen sind, und die betreffende Dehnungseinrichtung dehnt ein Paket aus paketierten Daten, wie Daten eines dreidimensionalen Objekts, welche ihr ebenfalls von dem Hauptspeicher **45** über den Hauptbus **41** zugeführt sind, entsprechend der durch die Haupt-DMAC-Steuereinrichtung **46** ausgeführten Steuerung und speichert das gedehnte Paket in dem eingebetteten Speicher (MEM), der in der Einrichtung VU-MEM **107** verwendet ist. Sodann führen die Einrichtungen FMAC **104** und DIV **106** Teile der Verarbeitung, wie eine Matrix-Verarbeitung, eine Transformation von Koordinaten und eine Radius-Kopierumsetzung bezüglich der Daten des dreidimensionalen Objekts aus. Auf der Grundlage einer Matrix und eines graphischen Kontextes, die gemeinsam durch die Haupt-CPU **44** und die erste Vektor-Verarbeitungsmaschine **71** erzeugt und an die zweite Vektor-Verarbeitungsmaschine **48** aus dem Hauptspeicher **45** über den Hauptbus **41** abgegeben sind, ist die Verarbeitung eine relativ einfache Verarbeitung.

[0058] Eine Anzeigeliste (Einfachstrom) zum Zeichnen eines auf diese Weise erzeugten zweidimensionalen Objekts auf einem Anzeigeschirm wird schließlich über den Hauptbus **41** an die Schnittstelleneinrichtung GPUIF **72** abgegeben. Die beiden Ströme, das sind die komplexen und einfachen Ströme, werden dann durch ein Zuteilungsverfahren auf einer Zeitmultiplexbasis zur GPU-Einheit **49** übertragen.

[0059] Die GPU-Einheit **49** führt die Zeichenverarbeitung auf der Grundlage der Anzeigelisten aus, die ihr von der Schnittstelleneinrichtung GPUIF **72** zugeführt werden, und sie zeichnet Polygone im Bildspeicher **58**. Falls die Anzeigeliste eine Anzeigeliste ist, die gemeinsam von der Haupt-CPU **44** und der ersten Vektor-Verarbeitungsmaschine **71** in der Hauptspeichereinheit **45** erzeugt und dann an die GPU-Einheit **49** über den Hauptbus **41** abgegeben ist, führt die GPU-Einheit **49** die Zeichenverarbeitung unter Heranziehung der früher zitierten graphischen Kontext-GC0-Einrichtung aus. Falls die Anzeigeliste eine Anzeigeliste ist, welche durch die zweite Vektor-Verarbeitungsmaschine **48** erzeugt ist, führt andererseits die GPU-Einheit **49** die Zeichenverarbeitung unter Heranziehung des zuvor erwähnten graphischen Kontextes GC1 aus.

[0060] Ein im Bildspeicher **58** gezeichnetes Polygon wird in ein Ausgabe-Videosignal für das Polygon entsprechend der durch die CRTC-Steuereinrichtung **84** ausgeführten Steuerung umgesetzt.

[0061] **Fig.** 7 veranschaulicht in einem Diagramm die zeitliche Steuerung, mit der zwei Anzeigelisten verarbeitet werden. Ein in **Fig.** 7 dargestelltes Geometrie-Subsystem 0 entspricht der in Fig. 6 dargestellten zweiten Vektor-Verarbeitungsmaschine **48**, während das Geometrie-Subsystem 1 der Haupt-CPU **44** und der ersten Vektor-Verarbeitungsmaschine **71** entspricht. Ein Rendering- bzw. Wiedergabe-Subsystem entspricht der GPU-Einheit **49**. Es sei darauf hingewiesen, dass ein in der Zeichnungsfigur dargestellter schraffierter Bereich angibt, dass sich ein durch einen Prozess- bzw. Task-Namen bezeichneter Prozess im Ruhezustand befindet.

[0062] Fig. 7A veranschaulicht in einem Diagramm eine Verarbeitungsprozedur für einen Fall, bei dem lediglich ein Prozessor, das ist das Geometrie-Subsystem 0, existiert. In diesem Falle bildet das Geometrie-Subsystem 0 eine Anzeigeliste (Liste #0-1) und gibt die Liste an das Wiedergabesystem ab. Sodann setzt das Geometrie-Subsystem 0 die Bildung der Anzeigeliste, welche der Liste #0-1 folgt, fort, d.h. mit der Liste #0-2, und anschließend zeigt es die Listen an. Das Wiedergabesystem führt eine Zeichenverarbeitung entsprechend der Anzeigeliste (Liste #0-1) aus, die ihm vom Geometrie-Subsystem 0 zugeführt ist. Falls das Geometrie-Subsystem 0 noch die nächste Anzeigeliste (Liste #0-2) zu dem Zeitpunkt bildet, zu dem das Wiedergabe-Subsystem die Zeichenverarbeitung entsprechend der Liste #0-1 abschließt, tritt das Wiedergabe-Subsystem in einen Ruhezustand ein und wartet darauf, dass das Geometrie-Subsystem 0 die Erzeugung der nächsten Anzeigeliste (Liste #0-2) abschließt und die Liste an das Wiedergabe-Subsystem liefert.

[0063] Falls danach in entsprechender Weise, wie

dies oben beschrieben worden ist, das Geometrie-Subsystem 0 die Verarbeitung zur Bildung der nächsten Anzeigeliste zu einem Zeitpunkt noch nicht abgeschlossen hat, zu dem das Wiedergabe-Subsystem die Zeichen-Verarbeitung entsprechend der gegenwärtigen Anzeigeliste abschließt, tritt das Wiedergabe-Subsystem in einen Ruhezustand ein und wartet darauf, dass das Geometrie-Subsystem 0 die nächste Liste an das Wiedergabe-Subsystem liefert.

[0064] Fig. 7B veranschaulicht in einem Diagramm eine Verarbeitungsprozedur für einen Fall, in welchem zwei Prozessoren existieren, das sind das Geometrie-Subsystem 0 und das Geometrie-Subsystem 1. Während in diesem Fall das Geometrie-Subsystem 0 eine Anzeigeliste (Liste #0-1) bildet, würde das Wiedergabe-Subsystem in einen Ruhezustand versetzt werden. Aus diesem Grunde werden die Daten, welche einer Anzeigeliste (Liste #1-1) zugehörig sind, die bereits vom Geometrie-Subsystem 1 erzeugt und im Hauptspeicher **45** gespeichert ist, dem Wiedergabe-Subsystem zugeführt. Auf den Empfang der ersten Anzeigeliste (Liste #1-1), die durch das Geometrie-Subsystem 1 erzeugt ist, führt das Wiedergabe-Subsystem eine Zeichenverarbeitung auf der Grundlage eines graphischen Kontextes für das Geometrie-Subsystem 1 aus, welcher der dem Wiedergabe-System von dem Geometrie-Subsystem 1 gelieferten ersten Anzeigeliste (Liste #1-1) angehängt ist.

[0065] Wenn das Geometrie-Subsystem 0 die Verarbeitung abschließt, um die erste Anzeigeliste (Liste #0-1) zu erzeugen, liefert das Geometrie-Subsystem 1 eine nächste Anzeigeliste (Liste #1-2) an das Wiedergabe-System. Zu diesem Zeitpunkt wird das Geometrie-Subsystem 1 veranlasst, die 0peraration anzuhalten, um die nächste Anzeigeliste (Liste #1-2) an das Wiedergabe-Subsystem abzugeben. Somit kann das Geometrie-Subsystem 0 nunmehr die vervollständigte erste Anzeigeliste (Liste #0-1) an das Wiedergabe-Subsystem abgeben und die Bildung der nächsten Anzeigeliste (Liste #0-2) beginnen. Auf den Empfang der ersten Anzeigeliste (Liste #0-1) von dem Geometrie-Subsystem 0 her führt das Wiedergabe-Subsystem eine Zeichenverarbeitung auf der Grundlage der ersten Anzeigeliste (Liste #0-1) aus.

[0066] Wenn das Wiedergabe-Subsystem die Zeichen- bzw. Zeichnungsverarbeitung auf der Grundlage der ersten Anzeigeliste (Liste #0-1) abschließt, bildet das Geometrie-Subsystem 0 noch die nächste Anzeigeliste (Liste #0-2). Aus diesem Grunde nimmt das Geometrie-Subsystem 1 die ausgesetzte Operation wieder auf, um die nächste Anzeigeliste (Liste #1-2) an das Wiedergabe-Subsystem abzugeben. Ansonsten würde das Wiedergabe-Subsystem in einen Ruhezustand eintreten. Auf den Empfang der nächsten durch das Geometrie-Subsystem 1 erzeugten Anzeigeliste (Liste #1-2) beginnt das Wiedergabe-Subsystem die Ausführung der Zeichen- bzw. Zeichnungsverarbeitung auf der Grundlage der nächsten Anzeigeliste (Liste #1-2).

[0067] Danach liefert das Geometrie-Subsystem 1 im Wesentlichen in der gleichen Weise, wie oben beschrieben, eine Anzeigeliste, die dadurch erzeugt ist, dass lediglich das Geometrie-Subsystem 0 noch eine Anzeigeliste bildet, wobei das Wiedergabe-Subsystem in einen Ruhezustand versetzt ist. Infolgedessen können durch eine Vielzahl von Prozessoren erstellte Anzeigelisten durch das Wiedergabe-Subsystem effizient verarbeitet werden.

[0068] Um beispielsweise die Koordinaten-Transformationsverarbeitung schneller auszuführen, kann ein Sub-Prozessor oder ein Koordinatentransformations-Coprozessor separat zur CPU (die früher erwähnte VU-Einheit) in jeder Maschine einer Vielzahl von Vektor-Verarbeitungsmaschinen vorgesehen sein, die eine gemeinsame Zeicheneinheit gemeinsam nutzen, das ist die GPU-Einheit **49**, und die jeweils Anzeigelisten an die GPU-Einheit **49** abgeben. Durch Anwendung eines derartigen Sub-Prozessors oder Coprozessors ist die CPU (Prozessor) in der jeweiligen Vektor-Verarbeitungsmaschine nunmehr im Stande, Anzeigelisten häufiger an die gemeinsam genutzte GPU-Einheit **49** zu liefern. Somit ist die GPU-Einheit **49** von einem Prozessor zu einem anderen in kurzen Zeitintervallen umzuschalten. Andernfalls wird ein Überlauf in einem für den jeweiligen Prozessor vorgesehenen lokalen Speicher auftreten. Aus diesem Grunde wird jedem Prozessor eine Prioritätsebene zugeteilt, d.h. dem Geometrie-Subsystem 0 und dem Geometrie-Subsystem 1, wie in Fig. 7B veranschaulicht. Wenn keine Anzeigeliste mehr zu der GPU-Einheit **49** von einem Haupt-Prozessor zu übertragen ist, das ist eine CPU mit der höchsten Priorität oder die CPU, die in der zweiten Vektor-Verarbeitungsmaschine **48** im Falle der in Fig. 6 dargestellten Informationsverarbeitungsvorrichtung verwendet wird, oder das in **Fig.** 7 dargestellte Geometrie-Subsystem 0, geht das Recht zur Vornahme eines Zugriffs auf die GPU-Einheit **49** auf einen Slave-Prozessor über, das ist eine CPU, die eine zweite Priorität bezüglich der Haupt-CPU **44** oder der in der ersten Vektor-Verarbeitungsmaschine **71** verwendeten CPU (VU-Einheit) im Falle der in Fig. 6 dargestellten Informationsverarbeitungsvorrichtung oder des in **Fig.** 7 dargestellten Geometrie-Subsystems 1 besitzt.

[0069] Sobald der Haupt- bzw. Master-Prozessor die Verarbeitung zur Herstellung einer Anzeigeliste abschließt und für die Übertragung der Anzeigeliste zur GPU-Einheit **49** vorbereitet ist, wird der Slave-Prozessor veranlasst, das Recht zum Zugriff auf die GPU-Einheit **49** an den Master-Prozessor sogar dann zurückzugeben, wenn eine Anzeigeliste noch durch den Slave-Prozessor an die GPU-Einheit **49** zu

vervollständigen und zu übertragen bleibt.

[0070] Im Allgemeinen ist der Master-Prozessor im Stande, eine Verarbeitung bei einer hohen Geschwindigkeit auszuführen; er weist jedoch einen lokalen Speicher mit einer relativen kleinen Speicherkapazität auf. Andererseits führt ein Slave-Prozessor eine Verarbeitung mit einer relativ niedrigen Geschwindigkeit aus, weist jedoch einen lokalen Speicher mit einer relativ großen Speicherkapazität auf.

[0071] Es gibt außerdem eine Informationsverarbeitungsvorrichtung, in der ferner ein Prozessor 2, der als Slave für einen Slave-Prozessor 1 dient, angeschlossen ist, wie dies in Fig. 8 veranschaulicht ist. In einer solchen Informationsverarbeitungsvorrichtung benötigt ein Prozessor mit einer insbesondere niedrigen Priorität einen lokalen Speicher mit einer sogar größeren Speicherkapazität zur Speicherung von mehr Anzeigelisten. Aus diesem Grunde wird normalerweise eine niedrige Priorität einem Hauptprozessor zugeteilt, der mit einem Hauptspeicher versehen ist. Auf diese Weise dient der Master- bzw. Haupt-Prozessor ebenso auch als Slave-Prozessor.

[0072] Bei einer durch die GPU-Einheit **49** ausgeführten Zeichen- bzw. Zeichnungsverarbeitung sind zusätzlich zu der in einer Anzeigeliste beschriebenen Eckpunktinformation Umgebungsparameter oder Zeichnungseinstellzustände bzw. -bedingungen, die als graphischer Kontext bezeichnet werden, wie ein Zeichnungsversatz und ein Abschneidebereich zur Zeichnungszeit ebenfalls erforderlich, wie oben beschrieben. Das Wiedergabe-Subsystem (d.h. die GPU-Einheit **49**) führt eine Zeichenverarbeitung auf der Grundlage von Anzeigelisten, welche von dem jeweiligen Geometrie-Subsystem (das ist die CPU) geliefert werden, in Übereinstimmung mit einem graphischen Kontext für das Geometrie-Subsystem aus. Wenn der Anzeigelisten-Lieferant von einem Geometrie-Subsystem zu einem anderen Subsystem umgeschaltet wird, muss jedoch eine Menge Arbeit zur neuen Festlegung eines graphischen Kontextes getan werden. Um dieses Problem zu lösen, enthält das Wiedergabe-Subsystem so viele graphische Kontexte wie Geometrie-Subsysteme vorliegen.

[0073] Ein graphischer Kontext wird einer Anzeigeliste hinzugefügt, wie dies in **Fig.** 7 in typischer Weise für jedes der GPU-Einheit **49** zugeführte Objekt veranschaulicht ist, um dadurch gezeichnet zu werden. Infolgedessen ist die GPU-Einheit **49** im Stande, eine Zeichnungsverarbeitung für jedes Objekt auf der Grundlage eines dem Objekt zugehörigen graphischen Kontextes auszuführen.

[0074] Geometrie-Subsysteme und das Wiedergabe-Subsystem nutzen den Hauptbus **41** gemeinsam, der einen Datenbus und einen Adressbus umfasst. Das Geometrie-Subsystem, welches auf das Wiedergabe-Subsystem zugreift, überträgt die ID des Geometrie-Subsystems und eine durch das Geometrie-Subsystem erzeugte Anzeigeliste über den Adressbus bzw. den Datenbus zu dem Wiedergabe-Subsystem. Auf den Empfang der ID und der Anzeigeliste hin wählt das Wiedergabe-Subsystem einen graphischen Kontext entsprechend der ID aus und interpretiert die Anzeigeliste auf der Grundlage des graphischen Kontextes. Das Wiedergabe-Subsystem zeichnet sodann ein Bild im Bildpuffer.

[0075] Dadurch, dass man eine Vielzahl von Prozessoren (Vektor-Verarbeitungsmaschinen oder Geometrie-Subsysteme) eine GPU-Einheit **49** (Wiedergabe-Subsystem) auf einer Prioritätsbasis steuern lässt, wie dies oben beschrieben worden ist, kann die Speicherkapazität eines in jedem der Prozessoren vorgesehenen lokalen Speichers zur temporären Speicherung einer Anzeigeliste, die im Prozessor erstellt ist, auf ein Minimum reduziert werden. Infolgedessen ist es möglich, die Erstellung von Anzeigelisten parallel in den Prozessoren auszuführen, ohne die Kosten lokaler Speicher zu erhöhen. Dadurch, dass ein graphischer Kontext für jeden der Prozessoren in der GPU-Einheit **49** (dem Wiedergabe-Subsystem) festgehalten wird, kann überdies die Anzahl von doppelten Datenübertragungen, d.h. die Menge der Overhead-Arbeit, die während einer Kontext-Umschaltung auszuführen ist, reduziert werden.

[0076] Genau gesagt nutzen die Prozessoren den Datenbus gemeinsam und folglich den Hauptspeicher auf einer Zeitmultiplexbasis. Im Folgenden wird eine Beschreibung eines Verfahrens zur Steuerung von Daten entsprechend einem in den Daten selbst eingebetteten Meta-Befehl während einer Übertragung der Daten zu der GPU-Einheit **49** gegeben.

[0077] Fig. 9 veranschaulicht in einem Diagramm ein typisches Format eines Meta-Befehls. Ein Meta-Befehl ist ein Befehl, der vor zu übertragenden Daten hinzugefügt ist. Ein Meta-Befehl schreibt die Länge der übertragenen Daten, ein Ziel der Datenübertragung und den Operationscode des Meta-Befehls vor. Ein Meta-Befehl umfasst 128 Bits, von denen lediglich 64 in der Zeichnungsfigur dargestellte Bits gültig sind. Die Größe der zu übertragenden Daten wird in dem ersten 16-Bit-Feld QWC spezifiziert. Der Operationscode dieses Meta-Befehls belegt ein Feld vom 24sten Bit bis zum 31sten Bit. Ein Feld vom 32sten Bit bis zum 63sten Bit wird zur Spezifizierung einer Adresse verwendet, unter der diese zu übertragenden Daten gespeichert sind oder ein Meta-Befehl als nächster Befehl auszulesen ist.

[0078] Die Übertragung von Daten wird entsprechend dem Operationscode eines in den Daten eingebetteten Meta-Befehls wie folgt gesteuert.

[0079] Falls der Operationscode gegeben ist mit

„cnt" wird, nachdem so viele Datenwörter, wie durch das QWC-Feld spezifiziert sind, welches diesem Meta-Befehl folgt, übertragen worden sind, ein Meta-Befehl, der unter einer diesem Paket (das sind der Meta-Befehl und die Daten) folgenden Adresse gespeichert ist, durch den Prozessor ausgeführt. Falls der Operationscode gegeben ist mit „cnts", wird, nachdem so viele Datenwörter, wie durch das diesem Meta-Befehl folgende QWC-Feld durch Ausführen einer Boxsteuerung übertragen worden sind, ein Meta-Befehl, der unter einer diesem Paket folgenden Adresse gespeichert ist, durch den Prozessor ausgeführt. Falls der Operationscode gegeben ist mit „next", wird, nachdem so viele Datenwörter wie durch das diesem Meta-Befehl folgenden QWC-Feld spezifiziert sind, übertragen worden sind, ein Meta-Befehl, der unter einer in dem Adressenfeld spezifizierten Adresse gespeichert ist, durch den Prozessor ausgeführt.

[0080] Die Boxsteuerung ist eine Zeitsteuerung, die durch einen Prozessor selbst ausgeführt wird, um einen durch den Prozessor auf den Hauptspeicher **45** vorgenommenen Zugriff solange in einen Wartezustand zu bringen, bis ein durch einen anderen Prozessor vorgenommener Zugriff auf den Hauptspeicher **45** abgeschlossen ist.

[0081] Falls der Operationscode gegeben ist mit „REF", wird, nachdem so viele Datenwörter, wie durch das QWC-Feld spezifiziert sind, welches unter einer in dem Adressenfeld spezifizierten Adresse ADDR gespeichert ist, übertragen worden sind, ein Meta-Befehl, der unter einer diesem Meta-Befehl folgenden Adresse gespeichert ist, durch den Prozessor ausgeführt. Falls der Operationscode gegeben ist mit „refs", wird, nachdem so viele Datenwörter, wie durch das QwC-Feld spezifiziert sind, welches unter einer in dem Adressenfeld spezifizierten Adresse ADDR gespeichert ist, durch Ausführen der Boxsteuerung übertragen worden sind, ein Meta-Befehl, der unter einer diesem Meta-Befehl folgenden Adresse gespeichert ist, durch den Prozessor ausgeführt. Wie oben beschrieben, wird ein „REF"-Meta-Befehl zur Übertragung von Daten mit einer in dem QWC-Feld spezifizierten Länge von einer in dem Adressenfeld spezifizierten Adresse verwendet.

[0082] Falls der Operationscode gegeben ist mit „call", wird, nachdem so viele Datenwörter, wie durch das QWC-Feld spezifiziert sind, welches diesem Meta-Befehl folgt, übertragen worden sind, eine diesem Paket folgende Adresse in ein Register als Rückkehradresse geschoben (oder geladen), und ein Meta-Befehl, der unter einer in dem Adressenfeld des Meta-Befehls spezifizierten Adresse gespeichert ist, wird durch den Prozessor ausgeführt. Falls der Operationsode gegeben ist mit „ret", wird, nachdem so viele Datenwörter, wie durch das QWC-Feld spezifiziert sind, das diesem Meta-Befehl folgt, übertragen worden sind, ein Meta-Befehl, der unter einer Adresse gespeichert ist, die aus dem Register herausgeholt (oder ausgelesen) wird, durch den Prozessor ausgeführt. Es sei darauf hingewiesen, dass die Adresse in das Register als Rückkehradresse während der Ausführung eines Meta-Befehls mit dem „CALL"-Operationscode geschoben worden ist, der diesem „RET"-Meta-Befehl zugehörig ist. Falls der Operationscode gegeben ist mit „end", wird, nachdem so viele Datenwörter wie durch das QWC-Feld spezifiziert sind, das diesem Meta-Befehl folgt, übertragen worden sind, die Verarbeitung beendet.

[0083] Fig. 10 zeigt ein Diagramm, welches zur Erläuterung von Operationen herangezogen wird, die durch den Prozessor ausgeführt werden, wenn der Operationscode der Meta-Befehle gegeben ist mit „next", was bedeutet, dass die nächsten diesem Meta-Befehl folgenden Daten zu übertragen sind. Zunächst liest die Haupt-DMAC-Steuereinrichtung **46** ein 1-Wort als Meta-Befehlswort aus einer Adresse ADDR0 aus, die in einem Identifizierungs- bzw. Tag-Adressregister Dn_TADR gespeichert ist. Unter der Annahme, dass der Meta-Befehl gegeben ist mit „NEXT, ADDR = ADDR2, LEN = 8", was bedeutet, dass der Operationscode gegeben ist mit „next", das das QWC-Feld spezifiziert, dass die Länge der zu übertragenden Daten gegeben ist mit 8-q-Wörtern (Quadlet-Wörter), wobei ein q-Wort 128 Bits umfasst, und dass ADDR2 eine Adresse ist, die im Adressenfeld spezifiziert ist. Somit werden bei der Ausführung des Meta-Befehls „NEXT, ADDR = ADDR2, LEN = 8" 8-qWortdaten übertragen. Sodann wird ein Meta-Befehl „NEXT, ADDR = ADDR1, LEN = 2", der unter der Adresse ADDR2 gespeichert ist, ausgeführt.

[0084] Durch dasselbe Zeichen bzw. denselben Token werden bei der Ausführung des Meta-Befehls „NEXT, ADDR = ADDR1, LEN = 2" 2-qWortdaten unter der Steuerung übertragen, welche durch die Haupt-DMAC-Steuereinrichtung **46** ausgeführt wird. Sodann wird ein Meta-Befehl „END, ADDR = –, LEN = 8" ausgeführt, der unter einer Adresse ADDR1 gespeichert ist. Bei der Ausführung des Meta-Befehls „END, ADDR = –, LEN = 8" werden 8-qWortdaten übertragen. Sodann wird die Verarbeitung beendet.

[0085] Fig. 11 zeigt ein Diagramm, welches zur Erläuterung von Operationen herangezogen wird, die durch den Prozessor ausgeführt werden, wenn der Operationscode der Meta-Befehle gegeben ist mit „REF". Zunächst liest die Haupt-DMAC-Steuereinrichtung **46** ein 1-Wort als Meta-Befehlswort aus einer Adresse ADDR0 aus, die im Tag-Adressregister Dn_TADR gespeichert ist. Es sei angenommen, dass der Meta-Befehl gegeben ist mit „REF, ADDR = ADDR2, LEN = 2". Bei der Ausführung des Meta-Befehls „REF, ADDR = ADDR2, LEN = 2" werden 2-qWortdaten übertragen, die unter einer Adresse ADDR2 gespeichert sind. Sodann wird ein Meta-Be-

fehl „REF, ADDR = ADDR1, LEN = 8" ausgeführt, der diesem Meta-Befehl folgt.

[0086] Bei der Ausführung des Meta-Befehls „REF, ADDR = ADDR1, LEN = 8" werden 8-qWortdaten übertragen, die unter einer Adresse ADDR1 gespeichert sind. Sodann wird der diesem Meta-Befehl folgende Meta-Befehl „END, ADDR = –, LEN = 8" ausgeführt. Bei der Ausführung des Meta-Befehls „END, ADDR = –, LEN = 8" werden 8-qWortdaten übertragen. Die Verarbeitung wird sodann beendet.

[0087] Fig. 12 zeigt ein Diagramm, welches zur Erläuterung von Operationen herangezogen wird, die durch den Prozessor ausgeführt werden, wenn die Operationscodes der Meta-Befehle gegeben sind mit „CALL" und „RET". Zunächst liest die Haupt-DMAC-Steuereinrichtung **46** ein 1-Wort als Meta-Befehlswort aus einer Adresse ADDR0, die im Tag-Adressregister Dn_TADR gespeichert ist. Es sei angenommen, dass der Meta-Befehl gegeben ist mit „CALL, ADDR = ADDR1, LEN = 0". (Bei der Ausführung des Meta-Befehls „CALL, ADDR = ADDR1, LEN = 0" werden keine diesem Meta-Befehl folgende Daten übertragen, da LEN = 0 ist. Die Adresse eines Meta-Befehls „CALL, ADDR = ADDR2, LEN = 8", die diesem Meta-Befehl folgt, wird in ein erstes Register als Rückkehradresse geschoben.). Nach der Ausführung des Meta-Befehls „CALL, ADDR = ADDR1, LEN = 0" wird ein Meta-Befehl „CALL, ADDR = ADDR2, LEN = 8" ausgeführt, der auf der rechten Seite von Fig. 12 dargestellt ist und der unter einer Adresse ADDR1 gespeichert ist. Bei der Ausführung des Meta-Befehls „CALL, ADDR = ADDR2, LEN = 8" werden diesem Meta-Befehl folgende 8-qWortdaten übertragen (die Adresse eines Meta-Befehls „RET, ADDR = –, LEN = 0", die diesem Paket folgt, wird in ein zweites Register als Rückkehradresse geschoben.). sodann wird ein Meta-Befehl „RET, ADDR = –, LEN = 8" ausgeführt, der unter einer Adresse ADDR2 gespeichert ist.

[0088] Bei der Ausführung des Meta-Befehls „RET, ADDR = –, LEN = 8" werden 8-qWortdaten auf diesen Meta-Befehl folgend übertragen. Sodann wird der Meta-Befehl „RET, ADDR = –, LEN = 0" ausgeführt, dessen Adresse in das zweite Register bei der Ausführung des Meta-Befehls „CALL, ADDR = ADDR2, LEN = 8" geschoben wurde, wie er auf der rechten Seite dargestellt ist. Bei der Ausführung des Meta-Befehls „RET, ADDR = –, LEN = 0" werden auf diesen Meta-Befehl folgend keine Daten übertragen, da LEN = 0 ist.

[0089] Sodann wird der Meta-Befehl „CALL, ADDR = ADDR2, LEN = 8" auf der linken Seite, dessen Adresse in das erste Register bei der Ausführung des Meta-Befehls „CALL, ADDR = ADDR1, LEN = 0" geschoben wurde, ausgeführt. Bei der Ausführung des Meta-Befehls „CALL, ADDR = ADDR2, LEN = 8" auf der linken Seite werden auf diesen Meta-Befehl folgend 8-qWortdaten übertragen (die Adresse eines Meta-Befehls „END, ADDR = –, LEN = 0" auf dieses Paket folgend wird in das erste Register als Rückkehradresse geschoben.). Sodann wird der Meta-Befehl „RET, ADDR = –, LEN = 8", der unter ADDR2 gespeichert ist, ausgeführt.

[0090] Bei der Ausführung des Meta-Befehls „RET, ADDR = –, LEN = 8" werden 8-qWortdaten auf diesen Meta-Befehl folgend übertragen. Sodann wird der Meta-Befehl „END, ADDR = –, LEN = 0" ausgeführt, dessen Adresse in das erste Register bei der Ausführung des Meta-Befehls „CALL, ADDR = ADDR2, LEN = 8" auf der linken Seite geschoben wurde. Bei der Ausführung des Meta-Befehls „END, ADDR = –, LEN = 0" werden keine Daten übertragen, da LEN = 0 ist. Sodann wird die Verarbeitung beendet.

[0091] Wie oben beschrieben, wird eine Übertragung von Daten in Übereinstimmung mit einem in den Daten eingebetteten Meta-Befehl gesteuert.

[0092] Fig. 13 veranschaulicht in einem Diagramm einen Zustand, in welchem eine Übertragung von Daten entsprechend einem in den Daten eingebetteten Meta-Befehl gesteuert wird. Während die Haupt-CPU **44** eine Anzeigeliste (Anzeigeliste #0) erstellt, werden Daten, die einer Anzeigeliste (Anzeigeliste #1) zugehörig sind, welche der Anzeigeliste #0 um ein Bild vorangeht, zu der zweiten Vektor-Verarbeitungsmaschine (VPE1) **48** übertragen.

[0093] Zunächst erstellt die Haupt-CPU **44** in Verbindung mit der ersten Vektor-Verarbeitungsmaschine **71** eine Anzeigeliste (Anzeigeliste #0), die einen Meta-Befehl mit dem Operationscode „NEXT", einen Meta-Befehl mit dem Operationscode „REF", einen Meta-Befehl mit dem Operationscode „REF", eine Matrix, einen Meta-Befehl mit dem Operationscode „REF", eine Matrix, einen Meta-Befehl mit dem Operationscode „REF", eine Matrix, einen Meta-Befehl mit dem Operationscode „REF", einen Meta-Befehl mit dem Operationscode „REF", einen Meta-Befehl mit dem Operationscode „REF", eine Matrix und einen Meta-Befehl mit dem Operationscode „RET" umfasst, wie dies in Fig. 13 gezeigt ist.

[0094] Während die Haupt-CPU **44** eine Anzeigeliste (Anzeigeliste #0) in Verbindung mit der ersten Vektor-Verarbeitungsmaschine **71** erstellt, wie dies oben beschrieben worden ist, werden Daten, die einer Anzeigeliste (Anzeigeliste #1) zugehörig sind, welche der Anzeigeliste #0 um ein Bild vorangeht, zu der zweiten Vektor-Verarbeitungsmaschine (VPE1) **48** wie folgt übertragen. Zunächst wird ein Meta-Befehl mit dem Operationscode „NEXT" am Kopf der Anzeigeliste #1 ausgeführt, um den nachfolgenden Kontext zu der zweiten Vektor-Verarbeitungsmaschine **48** zu übertragen. Sodann wird ein erster Meta-Be-

fehl mit dem Operationscode „REF" auf den übertragenen Kontext folgend ausgeführt, um ein in einer Objektdatenbank innerhalb des Hauptspeichers **45** gespeichertes Programm 0 auszulesen (und dann das Programm zu der zweiten Vektor-Verarbeitungsmaschine **48** zu übertragen). Ein Eckpunkt-Datensatz, das ist der Inhalt eines in Fig. 13 dargestellten Objekts, kann beispielsweise irgendwo gespeichert sein, und lediglich Matrizen einer Anzeigeliste werden aktualisiert. Es ist somit möglich, ein Bild auf der Grundlage der Beobachteraugen zu erzeugen. Auf diese Weise variieren Daten mit ihrem Inhalt nicht von Bild zu Bild, wenn ein Programm aus einer Anzeigeliste durch Verwendung eines Meta-Befehls ausgelesen wird (um die Daten zu der zweiten Vektor-Verarbeitungsmaschine **48** ohne die Forderung zu übertragen, die Daten in die Anzeigeliste einzuschließen). Feste Daten, wie Zeichen und konstante Daten zwischen ihnen können von Meta-Befehlen in verschiedenen Anzeigelisten gemeinsam genutzt werden. Infolgedessen kann eine Anzeigeliste dadurch erstellt werden, dass lediglich Positionsdaten (das sind Matrizen), die in der Anzeigeliste enthalten sind, deren Inhalt sich von Bild zu Bild ändert, von einer zuvor erstellten Anzeigeliste mit Leichtigkeit aktualisiert werden.

[0095] Es sei darauf hingewiesen, dass die Objektdatenbank dreidimensionale Daten zur Beschreibung eines dreidimensionalen Objekts (nachstehend auch als Eckpunkt eines Objekts bezeichnet) und ein Programm zum Interpretieren der Objektdaten enthält. Falls eine Texturabbildung an einer Verzierung eines Objekts ausgeführt wird, werden die als Texturdaten (auch als Texturbild bezeichnet) verwendeten Bilddaten überdies auch in der Objektdatenbank gespeichert.

[0096] Sodann wird ein zweiter Meta-Befehl mit dem Operationscode „REF" auf den obigen ersten Meta-Befehl „REF" folgend ausgeführt, um einen dreidimensionalen Koordinaten-Dateneckpunkt des Objekts 0 auszulesen, d.h. Eckpunktkoordinaten des Objekts 0 (und um sodann die Eckpunktkoordinaten des Objekts 0 zu der zweiten Vektor-Verarbeitungsmaschine **48** zu übertragen). Sodann wird eine erste Matrix zu der zweiten Vektor-Verarbeitungsmaschine **48** (durch Ausführen eines Meta-Befehls mit dem Operationscode NEXT auf den zweiten Meta-Befehl „REF" folgend am Kopf der ersten Matrix) übertragen. Anschließend wird ein dritter Meta-Befehl mit dem Operationscode „REF" auf die übertragene erste Matrix hin ausgeführt, um dreidimensionale Koordinaten-Daten-Eckpunkte des Objekts 1 auszulesen, d.h. Eckpunktkoordinaten des Objekts 1 (und um sodann die Eckpunktkoordinaten des Objekts 1 zur zweiten Vektor-Verarbeitungsmaschine **48** hin zu übertragen).

[0097] Sodann wird eine zweite Matrix auf den obigen dritten Meta-Befehl „REF" hin zur zweiten Vektor-Verarbeitungsmaschine **48** durch Ausführen eines Meta-Befehls mit dem Operationscode NEXT auf den dritten Meta-Befehl „REF" am Kopf der zweiten Matrix übertragen. Anschließend wird ein vierter Meta-Befehl mit dem Operationscode „REF" auf die übertragene zweite Matrix hin ausgeführt, um erneut die dreidimensionalen Koordinaten-Daten-Eckpunkte des Objekts 1 auszulesen, d.h. die Eckpunktkoordinaten des Objekts 1 (und um sodann die Eckpunktkoordinaten des Objekts 1 zur zweiten Vektor-Verarbeitungsmaschine **48** zu übertragen). Sodann wird eine dritte Matrix zur zweiten Vektor-Verarbeitungsmaschine **48** durch Ausführen eines Meta-Befehls mit dem Operationscode NEXT auf den vierten Meta-Befehl „REF" hin im Kopf der dritten Matrix übertragen. Anschließend wird ein fünfter Meta-Befehl mit dem Operationscode „REF" auf die übertragene dritte Matrix folgend ausgeführt, um ein Programm 3 auszulesen (und um sodann das Programm zur zweiten Vektor-Verarbeitungsmaschine **48** zu übertragen). Sodann wird ein sechster Meta-Befehl mit dem Operationscode „REF" auf den obigen fünften Befehl „REF" folgend ausgeführt, um Textur-Bilddaten aus dem Bildspeicher **58** auszulesen und um die Daten zur zweiten Vektor-Verarbeitungsmaschine **48** zu übertragen.

[0098] Falls die Textur-Bilddaten noch nicht im Bildspeicher **58** gespeichert sind, werden die Textur-Bilddaten zum Bildspeicher **58** übertragen, bevor Objektdaten-Eckpunkte des Objekts 4 durch den folgenden siebten Meta-Befehl mit dem Operationscode „REF" übertragen werden. Falls die Textur-Bilddaten „auftauende" Daten sind, die von der MDEC-Steuereinrichtung **47** her kommen, oder wenn es sich um übertragene Daten handelt, die vom Subbus **42** her kommen, ändern sich die Textur-Bilddaten von Bild zu Bild. In diesem Fall wird eine Boxfunktion angewandt, um eine Synchronisation der Datenübertragung herbeizuführen, wie dies später beschrieben wird.

[0099] Während Bilddaten zur zweiten Vektor-Verarbeitungsmaschine **48** übertragen werden, wird die durch die zweite Vektor-Verarbeitungsmaschine **48** ausgeführte Verarbeitung temporär eingestellt. Es ist somit notwendig, die Übertragungszeitspanne zur Übertragung von Bilddaten dadurch zu minimieren, dass durch andere DMA-Kanäle während dieser Zeitspanne ausgeführte Aktivitäten angehalten werden. Das Anhalten von Aktivitäten, welche durch andere DMA-Kanäle ausgeführt werden, kann durch ein bestimmtes Steuerungsbit in einem Meta-Befehl spezifiziert werden, der zur Übertragung von Bilddaten verwendet wird. So werden beispielsweise die 24sten und 25sten Bits eines Meta-Befehls, wie in Fig. 9 dargestellt, als derartige Steuerungsbits verwendet.

[0100] Sodann wird der letzte (siebte) Meta-Befehl

mit dem Operationscode „REF" ausgeführt, um dreidimensionale Koordinaten-Daten-Eckpunkte des Objekts 4 auszulesen (und um sodann die Daten zur zweiten Vektor-Verarbeitungsmaschine **48** zu übertragen). Sodann wird die letzte (vierte) Matrix zur zweiten Vektor-Verarbeitungsmaschine **48** übertragen (durch Ausführen eines Meta-Befehls mit dem Operationscode NEXT auf den siebten Meta-Befehl „REF" am Kopf der (vierten) Matrix hin. Schließlich wird ein Meta-Befehl mit dem Operationscode „RET" zur Beendigung der Übertragungsverarbeitung ausgeführt.

[0101] Fig. 14 zeigt ein Diagramm, welches zur Erläuterung der oben zitierten Boxsteuerung herangezogen wird. Es sei angenommen, dass Daten von der Vorrichtung bzw. dem Gerät 0 zu einem Hauptspeicher und sodann von dem Hauptspeicher zu dem Gerät bzw. der Vorrichtung 1 in einer steigenden Reihenfolge von Datenspeicheradressen im Hauptspeicher übertragen werden. In diesem Fall kann eine Adresse im Hauptspeicher, aus welchem Daten zur Vorrichtung bzw. zum Gerät 1 zu übertragen sind, zu einem Zeitpunkt eine Adresse im Hauptspeicher, zu der Daten gerade kürzlich vom Gerät 0 übertragen worden sind, aus einigen Gründen überschreiten. Während einer solchen Zeitspanne wird die Übertragung der Daten aus dem Hauptspeicher zu der Vorrichtung bzw. dem Gerät 1 in einen eingestellten Zustand gebracht.

[0102] Bei dem in Fig. 13 dargestellten Beispiel von Datenübertragungen werden Textur-Bilddaten von einem Speicher in der MDEC-Steuereinrichtung **47** zum Hauptspeicher **45** und sodann vom Hauptspeicher zu der zweiten Vektor-Verarbeitungsmaschine **48** mit einer steigenden Reihenfolge von Datenspeicheradressen im Hauptspeicher **45** übertragen. In diesem Falle kann eine Adresse im Hauptspeicher **45**, aus dem Daten zur zweiten Vektor-Verarbeitungsmaschine **48** zu übertragen sind, zu einem Zeitpunkt eine Adresse im Hauptspeicher **45**, dem Daten von dem Speicher in der MDEC-Steuereinrichtung **47** gerade kürzlich übertragen worden sind, aus einigen Gründen übersteigen. Während einer derartigen Zeitspanne wird die Übertragung der Textur-Bilddaten von dem Hauptspeicher **45** zur zweiten Vektor-Verarbeitungsmaschine **48** in einen eingestellten Zustand gebracht, um eine Übertragungs-Synchronisation herbeizuführen.

[0103] Wie oben beschrieben, holt die Haupt-DMAC-Steuereinrichtung **46** einen Meta-Befehl aus einer Anzeigeliste ab und führt den Meta-Befehl aus, um Daten an Prozessoren zu verteilen. Infolgedessen können durch Programmieren der Reihenfolge und der Form oder der Priorität der Übertragung von Daten in den Daten zum Zeitpunkt, zu dem eine Anzeigeliste durch den Prozessor vorab erzeugt wird, die Daten in Abhängigkeit von den Charakteristiken der Daten in einer optimalen weise übertragen werden. Dadurch, dass der Prozessor die Reihenfolge der Übertragung von Daten in Form einer Liste, wie einer Anzeigeliste, vorab vorschreibt, ist es überdies für den Prozessor nicht notwendig, unnütz kopierte Daten für die Übertragungsarbeit in einem Speicher festzuhalten. Infolgedessen sind die Anzahl von unnützen Zugriffen auf den Speicher und die Größe der Anzeigeliste reduziert.

[0104] Überdies ist es notwendig, lediglich die jeweiligen zu übertragenden Datenteile, die sich von Bild zu Bild ändern, separat an zwei Stellen für individuelle Anzeigelisten zu speichern. Jeglicher Teil der Anzeigelisten, der sich von Bild zu Bild nicht ändert, kann in einem für sämtliche Anzeigelisten gemeinsamen Speicherbereich gespeichert werden. Damit kann die Größe eines zur Speicherung von Anzeigelisten benötigten Speichers reduziert werden. Dies heißt, dass eine Anzahl von Anzeigelisten in einem Speicher mit einer kleinen Speicherkapazität gespeichert werden kann.

[0105] Da Daten entsprechend einem in den Daten eingebetteten Meta-Befehl übertragen werden, kann an oberster Stelle stehend eine Synchronisation von Operationen zum Auslesen und Schreiben von Daten zwischen einer Vielzahl von Prozessoren mit Leichtigkeit hergestellt werden. Infolgedessen ist es einer Vielzahl von Prozessoren ermöglicht, einen Speicher ohne die Forderung nach Bereitstellung eines doppelten Puffers im Speicher gemeinsam zu nutzen.

[0106] Im Falle der oben beschriebenen Ausführungsform werden Daten auf einer CD-ROM gespeichert. Es sei jedoch darauf hingewiesen, dass auch ebenso gut andere Aufzeichnungsmedien verwendet werden können.

[0107] Wenn bei der zuvor beschriebenen Informationsverarbeitungsvorrichtung und dem zuvor beschriebenen Informationsverarbeitungsverfahren eine erste Erzeugungseinrichtung sich im Zuge der Verarbeitung befindet und eine Ausführungseinrichtung sich in einem Wartezustand befindet, gibt eine zweite Übertragungseinrichtung einen zweiten Befehl, falls überhaupt, an die Ausführungseinrichtung ab, und die Ausführungseinrichtung führt auf den Empfang des zweiten Befehls hin den zweiten Befehl aus.

Patentansprüche

1. Informationsverarbeitungsvorrichtung, umfassend
eine erste Erzeugungseinrichtung (**48**) zur Ausführung einer Verarbeitung für die Erzeugung von ersten Befehlen,
eine zweite Erzeugungseinrichtung (**71**) zur Ausführung einer Verarbeitung für die Erzeugung von zwei-

ten Befehlen,
wobei die betreffenden zweiten Befehle eine niedrigere Priorität besitzen als die genannten ersten Befehle und temporär gespeichert werden,
eine Ausführungseinrichtung (**49**) zur Ausführung der genannten ersten und zweiten Befehle durch Umschalten von einem der genannten ersten Befehle auf einen der genannten zweiten Befehle und umgekehrt mit einer bestimmten Zeitsteuerung,
eine erste Übertragungseinrichtung zur Übertragung der genannten ersten Befehle zu der genannten Ausführungseinrichtung und eine zweite Übertragungseinrichtung zur Übertragung der genannten zweiten Befehle zu der genannten Übertragungseinrichtung,
wobei dann, wenn sich die genannte erste Erzeugungseinrichtung im Zuge der Verarbeitung zur Erzeugung eines der genannten ersten Befehle befindet und die genannte Ausführungseinrichtung sich in einem Wartezustand befindet, die genannte zweite Übertragungseinrichtung einen der genannten gespeicherten zweiten Befehle, falls überhaupt, an die genannte Ausführungseinrichtung abgibt und wobei auf den Empfang des genannten einen Befehls der betreffenden zweiten Befehle hin die genannte Ausführungseinrichtung den genannten einen der betreffenden zweiten Befehle ausführt.

2. Informationsverarbeitungsvorrichtung nach Anspruch 1, wobei die genannte Ausführungseinrichtung so viele Parameter für die Ausführung der genannten ersten und zweiten Befehle wie die genannten ersten und zweiten Erzeugungseinrichtungen festhält und wobei dann, wenn der genannte erste Befehl von der genannten ersten Erzeugungseinrichtung empfangen wird, die genannte Ausführungseinrichtung den genannten ersten Befehl durch Heranziehung eines Parameters ausführt, der der betreffenden ersten Erzeugungseinrichtung zugeordnet ist, welche den genannten ersten Befehl abgibt, und wobei dann, wenn der genannte zweite Befehl von der genannten zweiten Erzeugungseinrichtung empfangen wird, die genannte Ausführungseinrichtung den genannten zweiten Befehl durch Heranziehung eines Parameters ausführt, der der betreffenden zweiten Erzeugungseinrichtung zugeordnet ist, welche den genannten zweiten Befehl abgibt.

3. Informationsverarbeitungsvorrichtung nach Anspruch 1, wobei sie ferner mit einer Speichereinrichtung zur Speicherung des durch die genannte zweite Erzeugungseinrichtung erzeugten zweiten Befehls versehen ist und wobei die genannte zweite Übertragungseinrichtung den in der genannten Speichereinrichtung gespeicherten zweiten Befehl zu der genannten Ausführungseinrichtung überträgt.

4. Informationsverarbeitungsvorrichtung nach Anspruch 1, wobei der genannte erste Befehl für eine Typ-Verarbeitung erzeugt wird, während der genannte zweite Befehl für eine Nicht-Typ-Verarbeitung erzeugt wird.

5. Informationsverarbeitungsverfahren, welches in einer Informationsverarbeitungsvorrichtung anzuwenden ist, umfassend die Schritte:
Erzeugen eines ersten Befehls unter Heranziehung einer ersten Erzeugungseinrichtung (**48**),
Erzeugen eines zweiten Befehls unter Heranziehung einer zweiten Erzeugungseinrichtung (**71**),
wobei die betreffenden zweiten Befehle eine niedrigere Priorität besitzen als die genannten ersten Befehle und temporär gespeichert werden,
Ausführen der genannten ersten und zweiten Befehle mittels einer Ausführungseinrichtung (**49**) durch Umschalten von einem der genannten ersten Befehle auf einen der genannten zweiten Befehle und umgekehrt mit einer bestimmten Zeitsteuerung,
Übertragen der genannten ersten Befehle zu der genannten Ausführungseinrichtung unter Heranziehung einer ersten Übertragungseinrichtung
und Übertragen der genannten zweiten Befehle zu der genannten Ausführungseinheit unter Heranziehung einer zweiten Übertragungseinrichtung,
wobei dann, wenn die genannte erste Erzeugungseinrichtung sich im Zuge der Verarbeitung zur Erzeugung eines der genannten ersten Befehle befindet und die genannte Ausführungseinrichtung sich in einem Wartezustand befindet, die genannte zweite Übertragungseinrichtung einen der genannten gespeicherten zweiten Befehle, falls überhaupt, an die genannte Ausführungseinrichtung abgibt, und wobei auf den Empfang des genannten einen Befehls der genannten zweiten Befehle hin die genannte Ausführungseinrichtung den genannten einen Befehl der genannten zweiten Befehle ausführt.

Es folgen 13 Blatt Zeichnungen

Anhängende Zeichnungen

FIG. 1

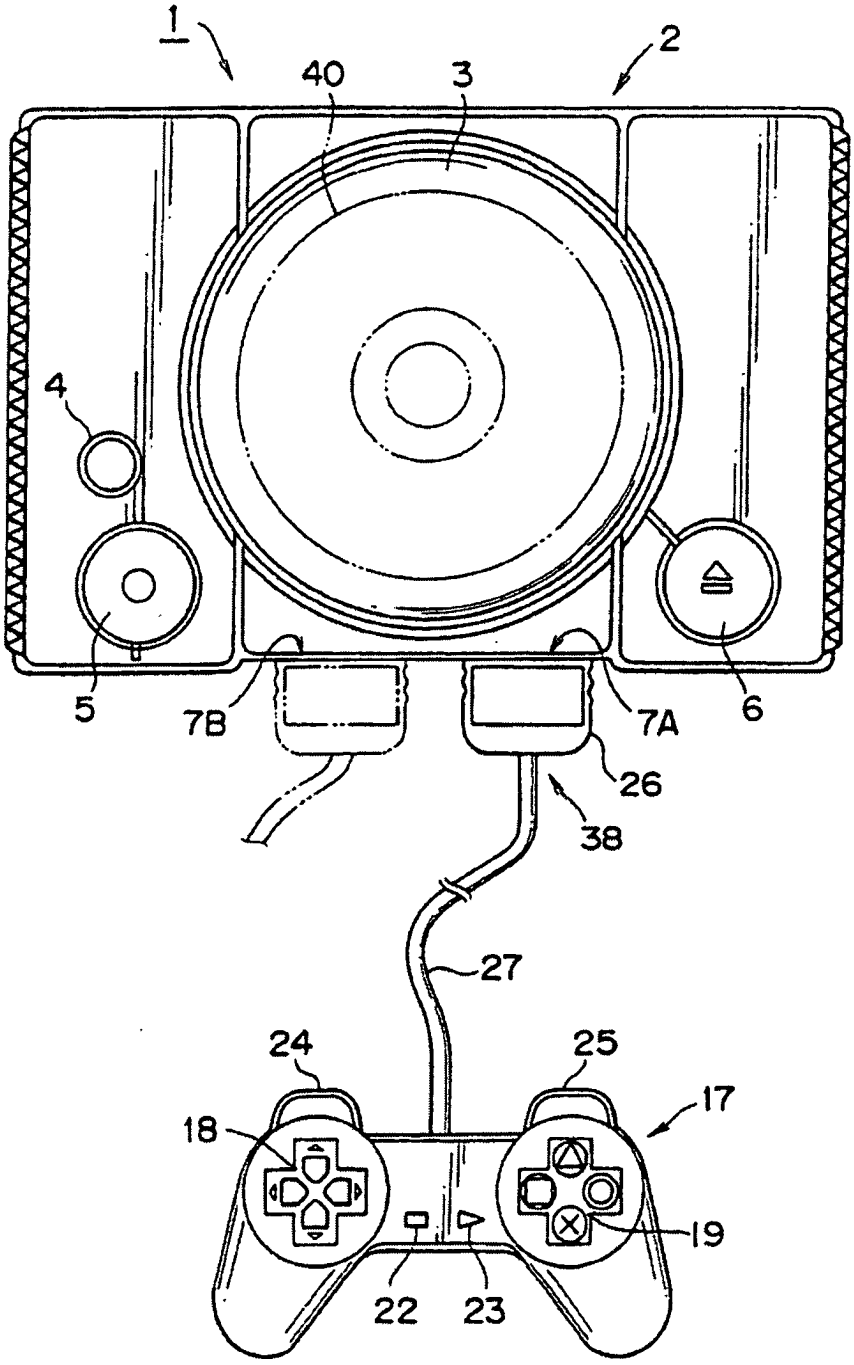


FIG. 2

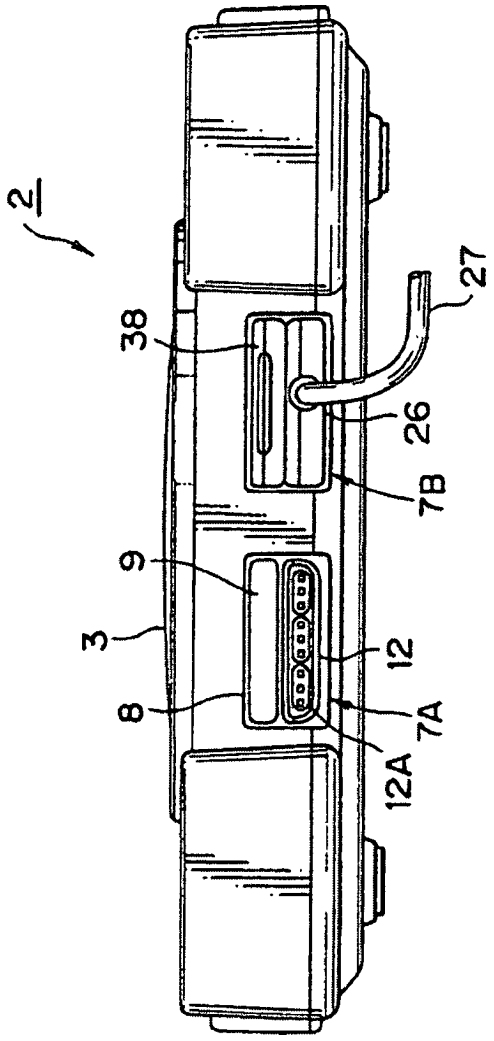


FIG. 3

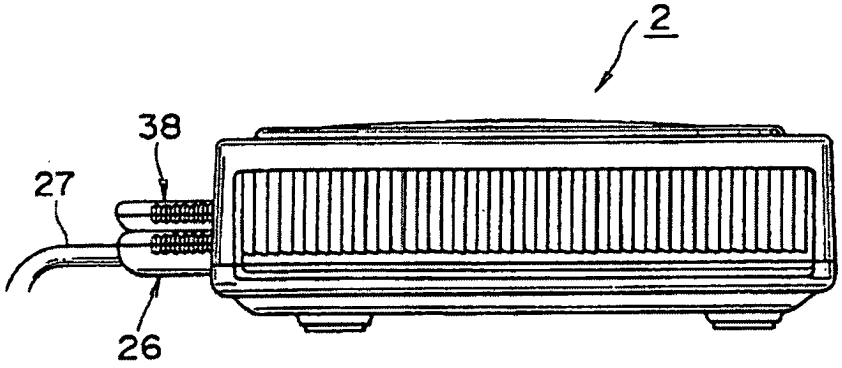


FIG. 4

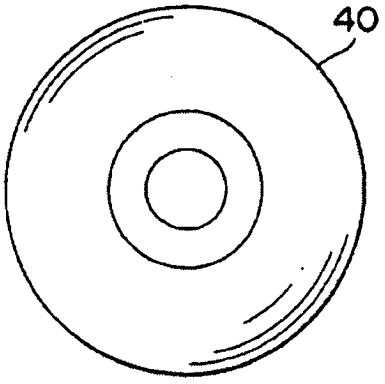


FIG. 5

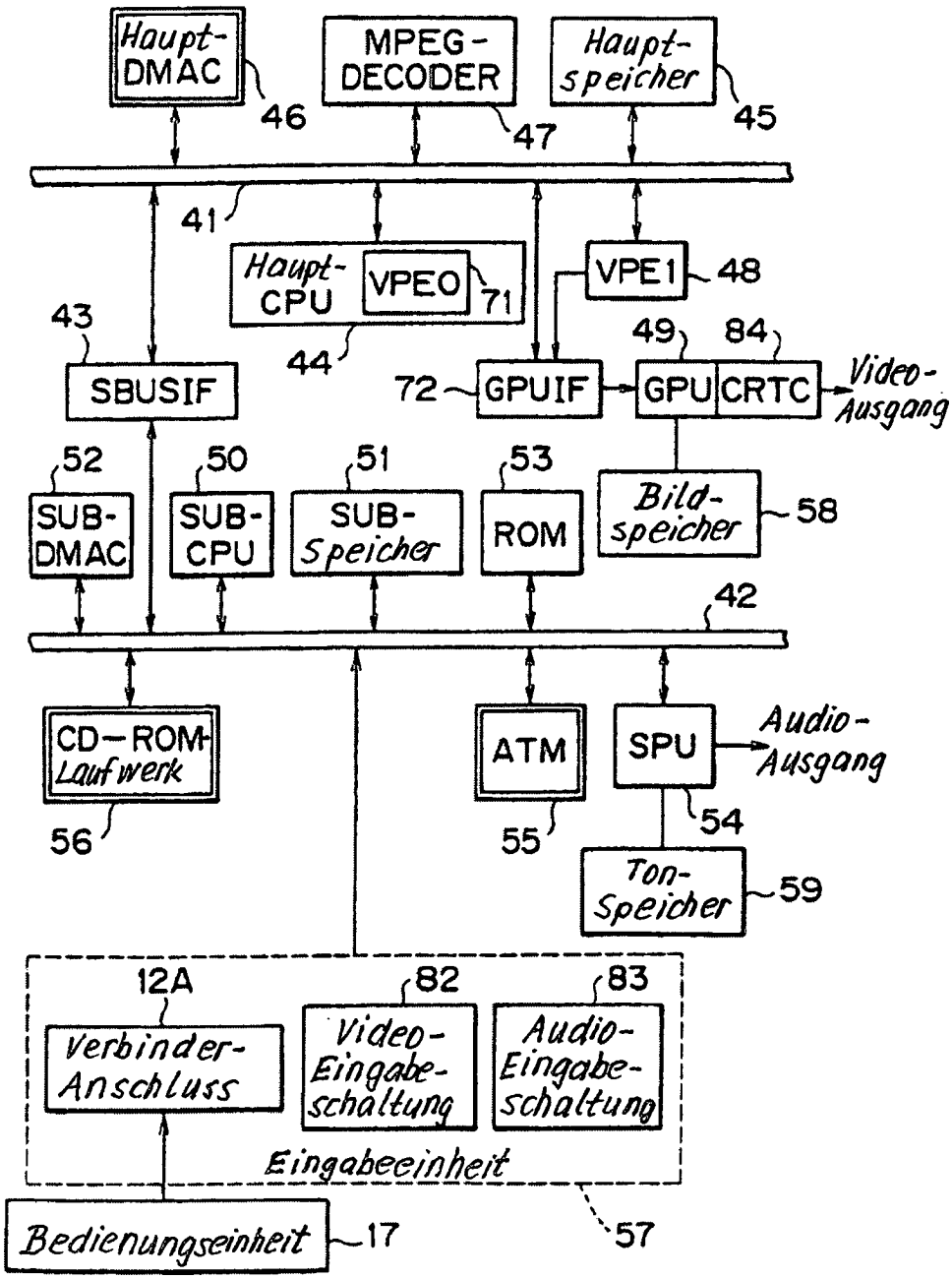

Haupt-DMAC
46
MPEG-DECODER
47
Haupt-speicher
45
41
Haupt-CPU
VPEO
71
VPE1
48
43
44
49
84
SBUSIF
72
GPUIF
GPU
CRTC
Video-Ausgang
52
50
51
53
SUB-DMAC
SUB-CPU
SUB-Speicher
ROM
Bild-speicher
58
42
CD-ROM-Laufwerk
56
ATM
55
SPU
Audio-Ausgang
54
Ton-Speicher
59
12A
82
83
Verbinder-Anschluss
Video-Eingabe-schaltung
Audio-Eingabe-schaltung
Eingabeeinheit
57
Bedienungseinheit
17

FIG. 6

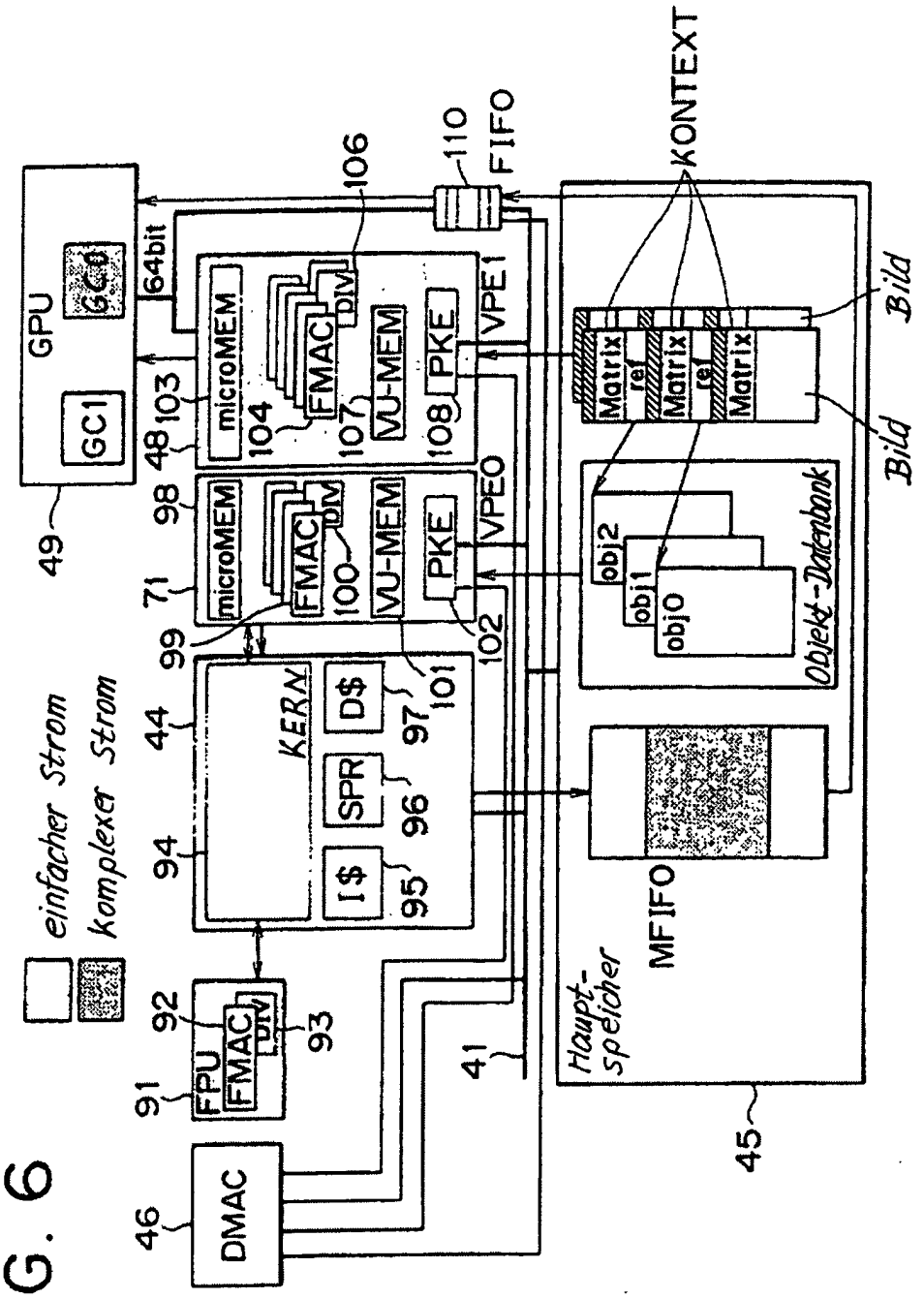

einfacher Strom
komplexer Strom
GPU
GC0
GC1
64bit
49
48
103
microMEM
104
FMAC
DIV
106
107
VU-MEM
108
PKE
VPE1
110
FIFO
KONTEXT
98
71
microMEM
FMAC
DIV
100
VU-MEM
PKE
VPE0
99
101
102
44
KERN
D$
97
SPR
96
I$
95
94
92
FPU
FMAC
DIV
93
91
41
DMAC
46
Matrix
ref
Matrix
ref
Matrix
Bild
Bild
obj2
obj1
obj0
Objekt-Datenbank
MFIFO
Haupt-speicher
45

FIG. 7A

FIG. 7B

Ablauf einer Anzeigeliste

FIG. 8

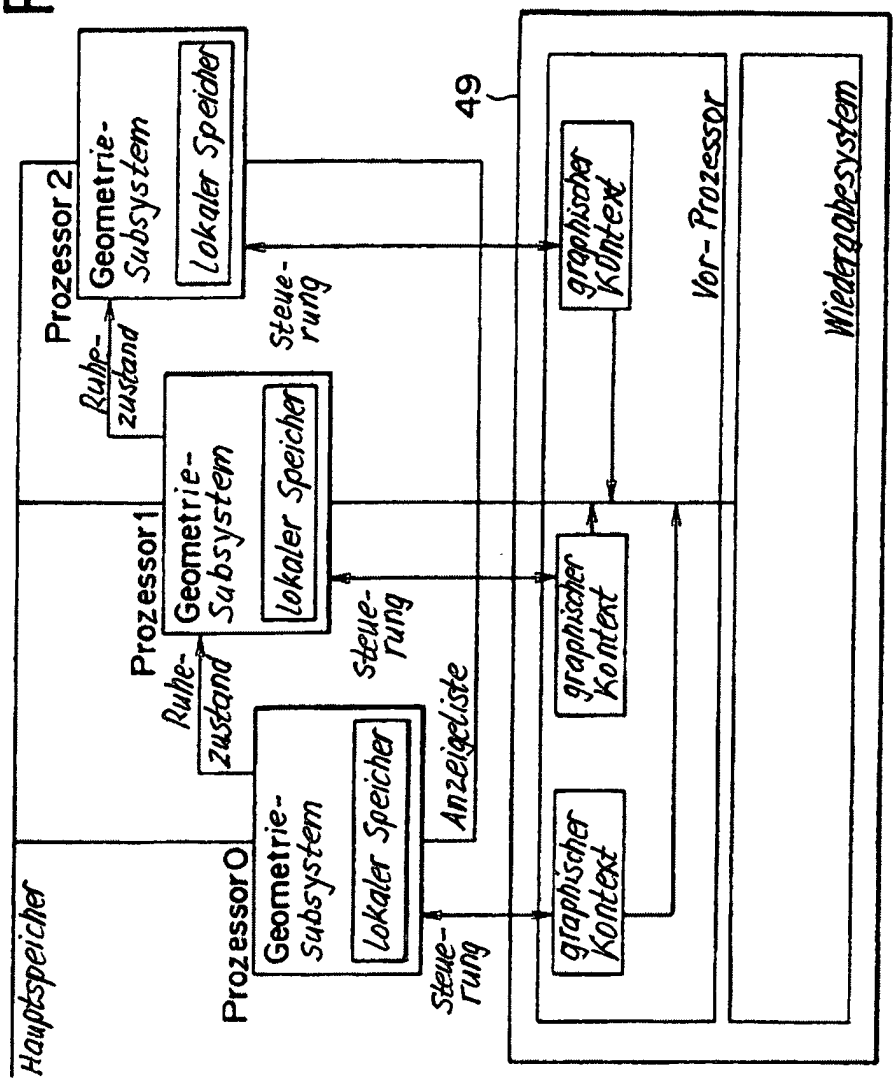

Hauptspeicher
Prozessor 0
Geometrie-Subsystem
Lokaler Speicher
Steue-rung
Anzeigeliste
Prozessor 1
Ruhe-zustand
Geometrie-Subsystem
Lokaler Speicher
Steue-rung
Prozessor 2
Ruhe-zustand
Geometrie-Subsystem
Lokaler Speicher
Steue-rung
49
graphischer Kontext
graphischer Kontext
graphischer Kontext
Vor-Prozessor
Wiedergabesystem

FIG. 9

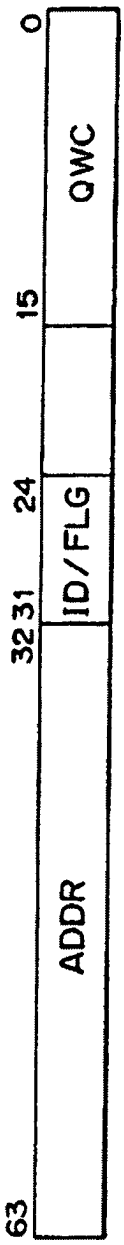


FIG. 10

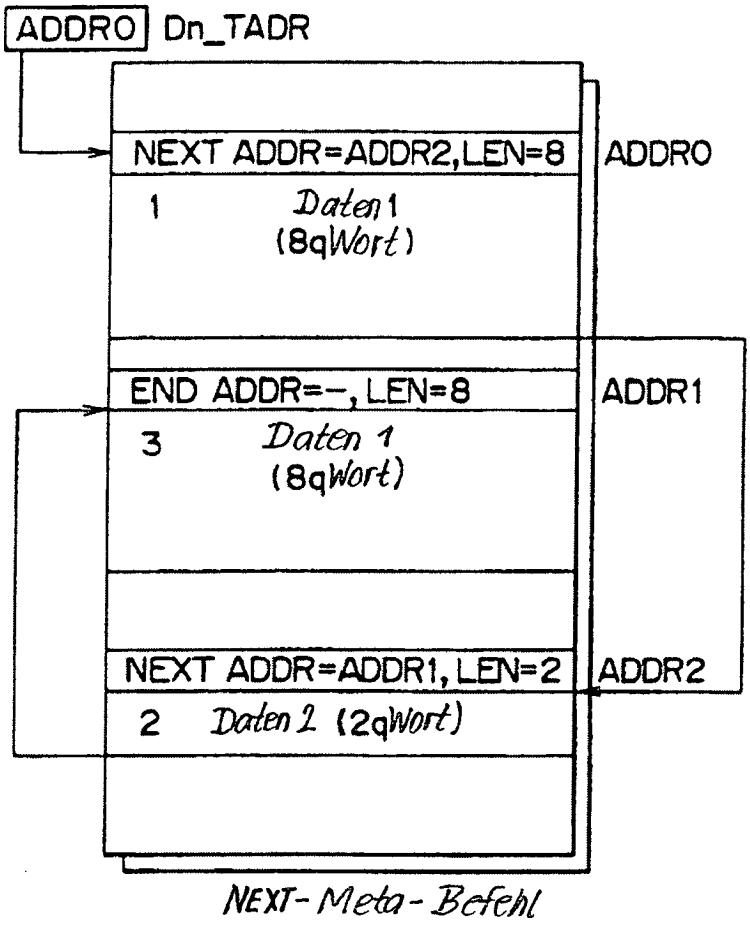

ADDR0
Dn_TADR
NEXT ADDR=ADDR2,LEN=8
ADDR0
1
Daten 1
(8qWort)
END ADDR=–, LEN=8
ADDR1
3
Daten 1
(8qWort)
NEXT ADDR=ADDR1, LEN=2
ADDR2
2
Daten 2 (2qWort)
NEXT-Meta-Befehl

FIG. 11

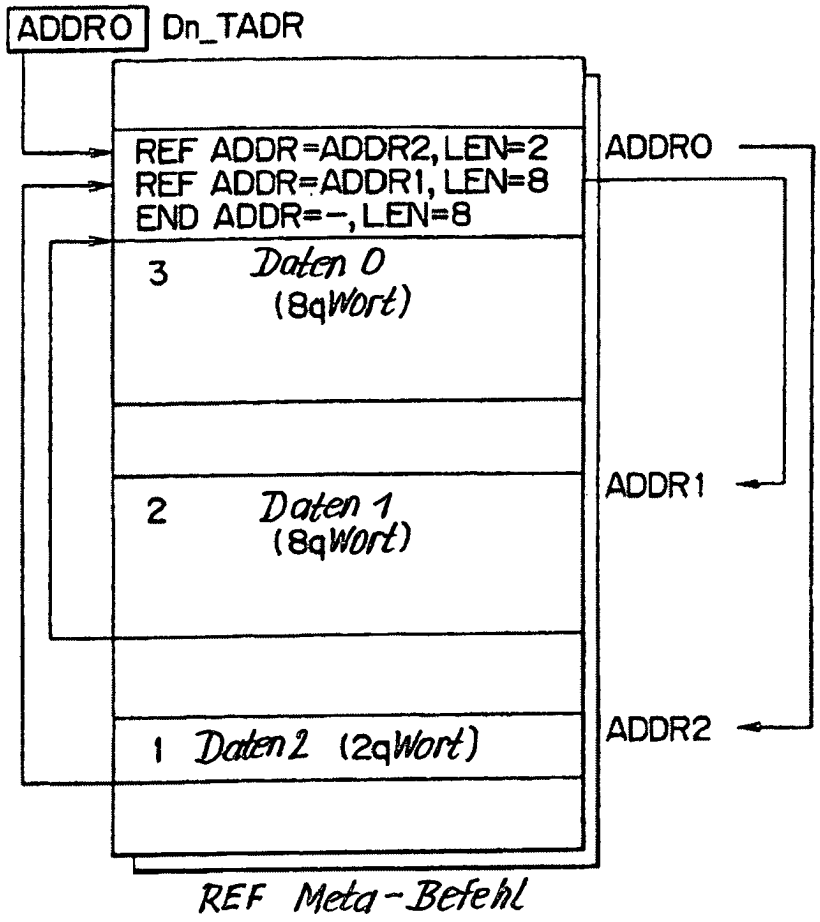

ADDR0
Dn_TADR
REF ADDR=ADDR2, LEN=2
REF ADDR=ADDR1, LEN=8
END ADDR=–, LEN=8
ADDR0
3
Daten 0
(8qWort)
ADDR1
2
Daten 1
(8qWort)
ADDR2
1
Daten 2 (2qWort)
REF Meta-Befehl

FIG. 12

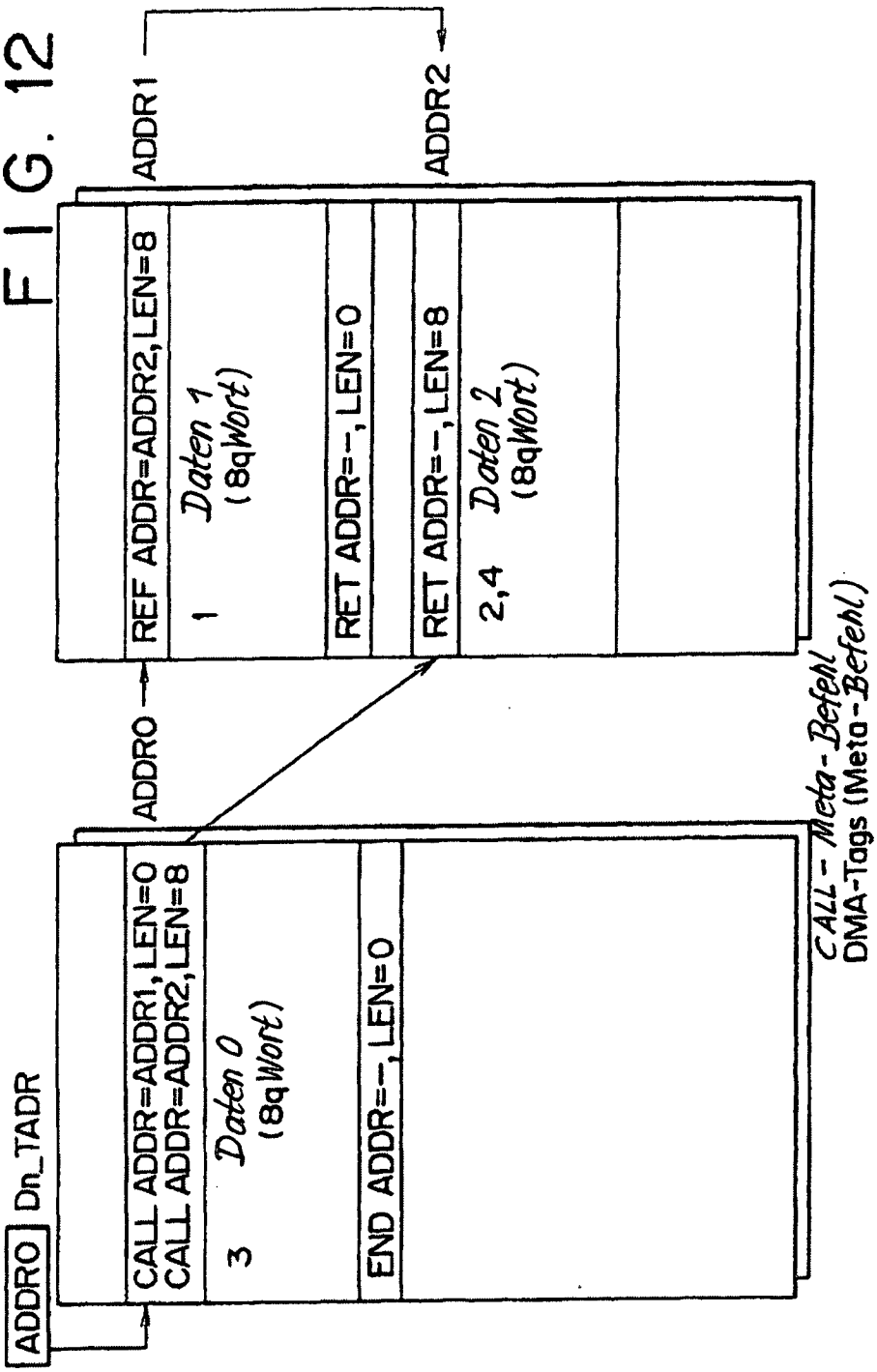

ADDR1
ADDR2
REF ADDR=ADDR2, LEN=8
1 Daten 1 (8q Wort)
RET ADDR=–, LEN=0
RET ADDR=–, LEN=8
2,4 Daten 2 (8q Wort)
ADDR0
ADDR0 Dn_TADR
CALL ADDR=ADDR1, LEN=0
CALL ADDR=ADDR2, LEN=8
3 Daten 0 (8q Wort)
END ADDR=–, LEN=0
CALL – Meta-Befehl
DMA-Tags (Meta-Befehl)

FIG. 13

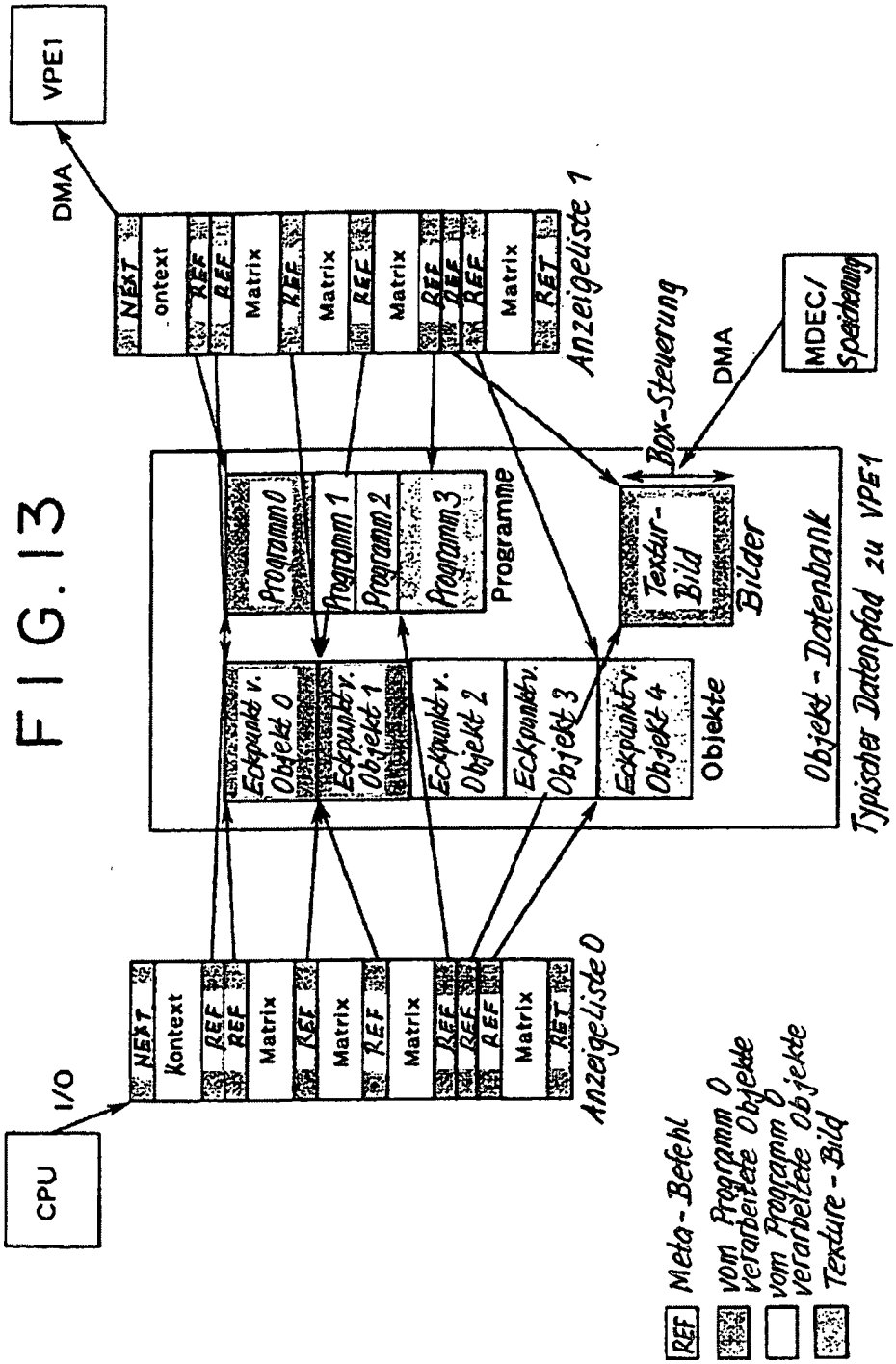
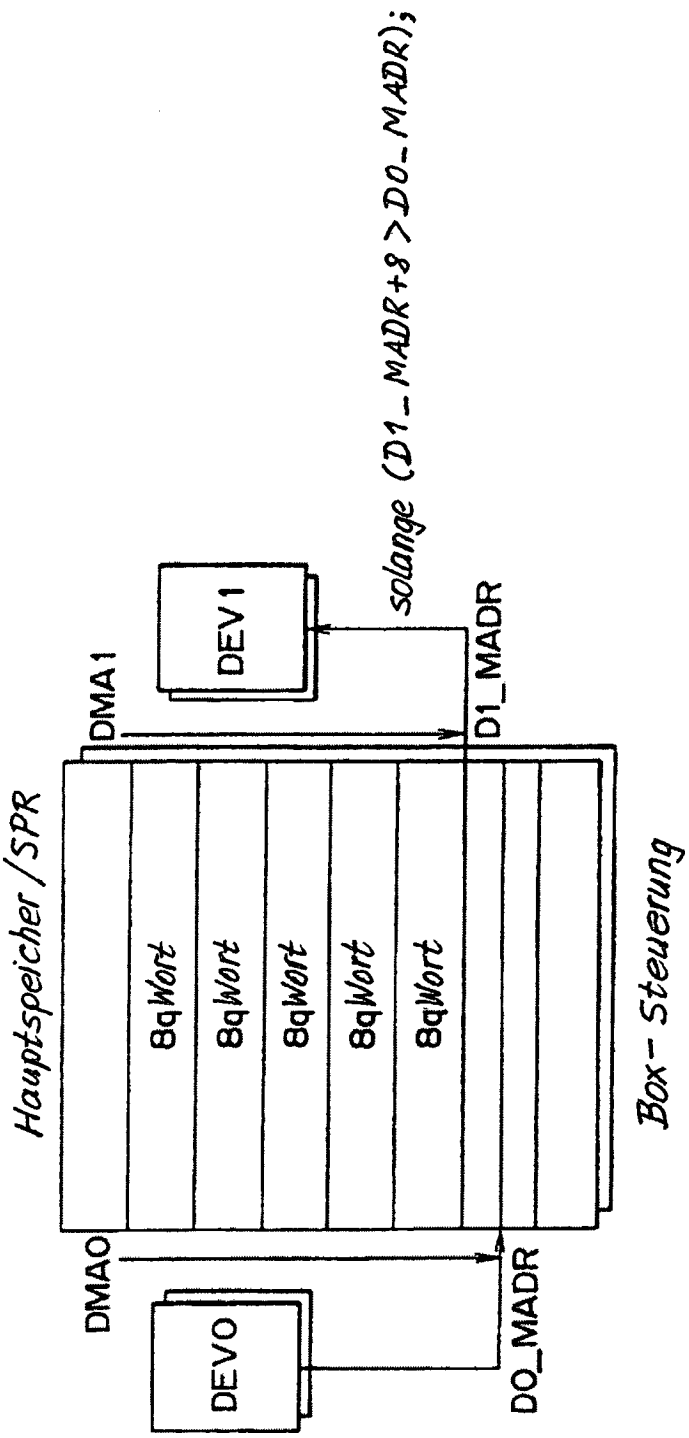

CPU
I/O
VPE1
DMA
NEXT
Kontext
REF
Matrix
Anzeigeliste 0
Anzeigeliste 1
Programm 0
Programm 1
Programm 2
Programm 3
Programme
Eckpunktv. Objekt 0
Eckpunktv. Objekt 1
Eckpunktv. Objekt 2
Eckpunktv. Objekt 3
Eckpunktv. Objekt 4
Objekte
Textur-Bild
Bilder
Box-Steuerung
MDEC/ Speicherung
Objekt-Datenbank
Typischer Datenpfad zu VPE1
Meta-Befehl
vom Programm 0 verarbeitete Objekte
vom Programm 0 verarbeitete Objekte
Texture-Bild

FIG. 14


solange (D1_MADR+8 > D0_MADR);
DEV1
DMA1
D1_MADR
Hauptspeicher /SPR
8qWort
8qWort
8qWort
8qWort
8qWort
Box-Steuerung
DMA0
DEV0
D0_MADR