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

(10) **DE 696 35 368 T2** 2006.08.03

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 0 794 654 B1**
(21) Deutsches Aktenzeichen: **696 35 368.7**
(96) Europäisches Aktenzeichen: **96 113 636.3**
(96) Europäischer Anmeldetag: **26.08.1996**
(97) Erstveröffentlichung durch das EPA: **10.09.1997**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **02.11.2005**
(47) Veröffentlichungstag im Patentblatt: **03.08.2006**

(51) Int Cl.[8]: ***H04N 1/405*** (2006.01)

(30) Unionspriorität:
610683 **04.03.1996** **US**

(84) Benannte Vertragsstaaten:
DE, FR, GB

(73) Patentinhaber:
Hewlett-Packard Development Co., L.P., Houston, Tex., US

(72) Erfinder:
Fujii, David B., Boise, Idaho 83706, US; Rust, Robert A., Boise, Idaho 83706, US

(74) Vertreter:
Schoppe, Zimmermann, Stöckeler & Zinkler, 82049 Pullach

(54) Bezeichnung: **Hochgeschwindigkeitssystem zur Ausrichtung und Verlegung von Schwellenmatrizen während der Erzeugung von binären Halbtonbildern**

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

GEBIET DER ERFINDUNG

[0001] Diese Erfindung bezieht sich auf eine Umwandlung von Graustufenbildern in binäre Halbtonbilder und insbesondere auf ein System, das beim Durchführen einer solchen Bildumwandlung eine Graustufenschwellenmatrix verwendet.

HINTERGRUND DER ERFINDUNG

[0002] Sowohl Farb- als auch Graustufenbilder bestehen aus Bildelementen (Pixeln), wobei jedes Pixel durch mehrere Bits dargestellt wird, die entweder eine Farbe oder eine Graustufe definieren. Um derartige Bilder auf einem Zweistufendrucker (Schwarz-Weiß-Drucker) darzustellen, werden die Daten, wenn sie nicht bereits eine Graustufe umfassen, in eine Graustufen-Mehrbitkonfiguration (z. B. 8 Bits/Pixel) umgewandelt, und anschließend werden die einzelnen Graustufenpixel unter Verwendung eines Skalierungs- und Dither-Prozesses in Binärpegelpixel umgewandelt.

[0003] Die Pixel in dem ursprünglichen Mehrbit-pro-Pixel-Bild weisen üblicherweise nicht die Auflösung der Zweistufendruckerauflösung auf und müssen somit skaliert werden, um mit der Druckerauflösung übereinzustimmen. Beispielsweise weisen viele moderne Drucker eine Auflösung von 600 Punkten pro Zoll (dpi – dot per inch) auf, wohingegen Computer Daten bei einer Auflösung von 1/6 der Druckerauflösung oder bei einer noch niedrigeren Auflösung ausgeben. Demgemäß ist es erforderlich, dass der Zweistufendrucker die empfangenen Graustufenpixel räumlich zu der Druckerauflösung skaliert. Dann werden die sich ergebenden skalierten Graustufenpixel mit einer Schwellenmatrix verglichen, die die Umwandlung der Graustufen in entsprechende Muster von Zweitstufendaten steuert. Dieser Prozess wird als „Dithern" bezeichnet.

[0004] Wenn 8 Bits verwendet werden, um die Graustufe darzustellen, können 256 Graustufen dargestellt werden. Die Schwellenmatrix umfasst eine Mehrzahl von zeilenartig angeordneten Graustufenwerten, die die Umwandlung der Graustufenpixelwerte in binäre Pixelwerte steuern. Im Wesentlichen ist jeder Eintrag in die Schwellenmatrix ein Schwellengraustufenwert, der, wenn er durch den Bildgraustufenpixelwert überschritten wird, bewirkt, dass ein Graustufenbildpixel in ein schwarzes Pixel umgewandelt wird. Wenn der Bildgraustufenpixelwert dagegen geringer als der oder gleich dem entsprechenden Schwellenmatrixgraustufenwert ist, wird er in ein „weißes" Pixel umgewandelt.

[0005] Der zuvor erwähnte Prozess ist in Fig. 1 veranschaulicht, bei der ein Quellenbild **10** eine 4×4-Matrix von Graustufenpixelwerten umfasst. Eine Schwellenmatrix **12** umfasst für dieses Beispiel ferner eine 4×4-Matrix von Graustufenschwellwerten. Angenommen, dass das Quellenbild **10** um einen Faktor 2 nach oben skaliert werden soll, repliziert ein Skalierungsvorgang die ursprünglichen Graustufenpixelwerte in dem Quellenbild **10** in die Form, die in dem skalierten Bild **14** gezeigt ist. Ein Skalieren kann durch eine Pixelreplikationsprozedur bewerkstelligt werden, die bewirkt, dass Pixel, die repliziert werden sollen, die Leerräume, die durch das Skalieren bewirkt werden, füllen. Andere Verfahren, z. B. eine bilineare Skalierung, werden ebenfalls verwendet, um die Graustufenwerte zwischen ursprünglichen Graustufenpixelwerten des Quellenbildes zu bestimmen. Ein bilineares Skalieren zieht benachbarte Graustufenpixelwerte in Betracht, wenn Räume, die durch das Skalieren verursacht wurden, gefüllt werden.

[0006] Als Nächstes wird die Schwellenmatrix **12** „fliesenartig verlegt" bzw. „gefliest", sodass sie über dem gesamten skalierten Bild **14** liegt. Die geflieste Schwellenmatrix **12** ist in Fig. 1 bei **16** gezeigt. Das skalierte Bild **14** wird als Nächstes durch einen Vergleichsvorgang, der in einem Vergleichsmodul **20** stattfindet, in ein Zweipegel-Zieldatenbild **18** umgewandelt. Jedes Graustufenpixel des skalierten Bildes **14** wird mit einem entsprechend positionierten Graustufenwert der gefliesten Schwellenmatrix **16** verglichen, und je nachdem, ob der skalierte Bildgraustufenpixelwert die geflieste Schwellenmatrix überschreitet, gleich derselben oder geringer als dieselbe ist, gibt das Vergleichsmodul **20** entweder ein 1- oder ein 0-Bit zur Speicherung an der entsprechenden Pixelstelle in dem Zweipegel-Zieldatenbild **18** aus.

[0007] Bei dem in Fig. 1 gezeigten Beispiel wird die Schwellenmatrix **12** einmal nach links und einmal nach unten gefliest, wobei bei ihrer oberen linken Ecke begonnen wird. Mit welchem Teil der Schwellenmatrix begonnen wird, wird durch einen Schwellenmatrix-„Anker" zu der Seite relativ zu dem Quellenbild **10** und ihren Anker auf der Seite bestimmt. Im Einzelnen definiert ein Anker einen Punkt auf einem Bild, wo ein spezifischer Punkt eines zweiten Bildes (z. B. eine Schwellenmatrix) überlagert wird, wodurch ermöglicht wird, dass das zweite Bild mit einer bekannten Positionierung über das erste Bild gelegt wird.

[0008] Bei Fig. 2a ist ein Quellengraustufenbild **22** gezeigt. Bei Fig. 2b ist ein Beispiel einer nach oben skalierten Version **24** des Graustufenbildes **22** gezeigt, das um den Faktor 1,5 nach oben skaliert wurde. Bei Fig. 2c ist ein Graustufenbild **26** gezeigt, dessen Auflösung um einen Wert von 0,75 nach unten skaliert wurde. In jedem Fall erreichte der Stand der Technik ein derartiges Skalieren durch Verwendung eines Horizontaleingabeindexarrays (HIIA) und eines Vertikaleingabeindexarrays (VIIA). Dies wird durch

Bezugnahme auf Fig. 3 mehr verständlich, bei der ein Quellenbild **30** eine 4×4-Matrix von 16 Pixeln umfasst. Das Quellenbild **30** soll in ein skaliertes Bild **32** umgewandelt werden, das eine 2,5-mal höhere Auflösung aufweist als das Quellenbild **30**. Das skalierte Bild **32** umfasst somit 10 Spalten von Pixeln.

[0009] Um den Skalierungsvorgang zu bewerkstelligen, berechnet ein Prozessor ein HIIA **34**, das einen Mehrbitwert für jede Spalte des skalierten Bildes **32** umfasst. Jeder HIIA-Wert stellt einen Versatz in das Quellenbild **30** dar und identifiziert dasjenige Pixel aus dem Quellenbild **30**, das an einer Pixelposition, die dem HIIA-Wert entspricht, in das skalierte Bild **32** eingefügt werden soll. Jeder HIIA-Wert gibt an, wie viele Pixel man in das Quellenbild **30** hineingehen soll, um das richtige Pixel zur Einfügung in eine entsprechende Position in dem skalierten Bild **32** zu erhalten.

[0010] Dasselbe Konzept wird bei der vertikalen Richtung verwendet. Nachdem eine Zeile eines skalierten Bildes abgeschlossen ist, bestimmt ein VIIA-Indexbit, ob die Zeile gemäß dem Skalierungsfaktor repliziert werden sollte. Im Einzelnen gibt VIIA **36** an, dass die ersten zwei Zeilen zu der ersten Zeile des Quellenbilds **30** indexiert werden sollten. Die zweiten zwei Zeilen des skalierten Bildes **32** sind von der zweiten Zeile des Quellenbildes **30** abzuleiten. Der Skalierungsfaktor von 2,5 wird erhalten, in dem manche Pixel um 3 skaliert werden, während andere Pixel um 2 skaliert werden.

[0011] Fig. 4 veranschaulicht die Anwendung von HIIAs und VIIAs auf die Skalierung eines Bildes unter Verwendung eines exemplarischen bilinearen Skalierungsalgorithmus. Der bilineare Skalierungsalgorithmus betrachtet benachbarte Pixel, wenn die Pixel zwischen den Werten, die von einem Quellenbild **40** kopiert wurden, gefüllt werden. Bei dem in Fig. 4 gezeigten Beispiel ist der Pixelwert (15) zu der oberen Zeile, zweite Spalte, das Ergebnis einer Mittelwertsbildung des ersten und des zweiten Pixels in dem Quellenbild **40**. Wenn man HIIA **42** untersucht, kann man erkennen, dass das erste Pixel des Quellenbildes **40** zu dem ersten Pixel in dem skalierten Bild **44** kopiert wird. Das zweite Pixel des Quellenbilds **40** geht zu dem dritten Pixel in dem skalierten Bild **44**. Das zweite Pixel in dem skalierten Bild **44** ist ein interpolierter Wert zwischen dem ersten und dem dritten Pixel. Bei Fig. 5 ist ein Herabskalierungsbeispiel veranschaulicht, bei dem ein Quellenbild **46** durch die Verwendung eines HIIA **50** und eines VIIA **52** zu einem skalierten Bild **48** herabskaliert wird.

[0012] Bei jeder der obigen bekannten Skalierungsprozeduren sind beträchtliche Speichermengen erforderlich, um das HIIA und das VIIA zu speichern. Beispielsweise liegen für ein 8,5 Zoll breites Blatt Papier auf einen 600-dpi-Drucker etwa 4800 Punktelemente pro Linie vor. Um jede Spalte des skalierten Bildes ordnungsgemäß zu adressieren, wurde bisher eine Quantität von vollen 16 Bit für jedes Element des HIIA verwendet. Somit sind 9,6 Kilobytes erforderlich, um HIIA-Daten unterzubringen. Ferner ist ein VIIA erforderlich, um das Skalieren von Zeilen in der vertikalen Richtung zu ermöglichen. Für ein 11 Zoll langes Blatt Papier auf einen 600-dpi-Drucker liegen etwa 6000 Punkte in der vertikalen Abmessung vor. 16-Bit-Quantitäten werden ebenfalls verwendet, um jedes VIIA-Zeilenelement zu adressieren (was 12 Kilobytes erfordert). Somit müssen für eine ganze Seite insgesamt 21,6 Kilobytes Speicher für die Eingabeindexarrays reserviert werden.

[0013] Unter Bezugnahme auf die Ausrichtung und Verfliesung einer Schwellenarraymatrix über ein skaliertes Quellenbild nimmt die bekannte Prozedur beträchtliche Zeit für das Durchführen von Berechnungen in Anspruch, die für jedes Pixel in dem sich ergebenden gefliesten Bild benötigt werden. Sie muss Indizes für jedes Element in der Schwellenmatrix berechnen und das spezifische indexierte Speicherelement wiedergewinnen. Folglich ist die Prozedur rechen- und zeitaufwendig. Somit kann es sein, dass bekannte Bildumwandlungsprozeduren Minuten brauchen, um Bilder in einer softwaregesteuerten Umgebung aufzubereiten. Derartige Umwandlungszeiten erfüllen nicht die Anforderungen des Marktes.

[0014] Die US-A-4486788 beschreibt ein bekanntes Verfahren zum Erzeugen einer Halbtonplatte aus einem Kontinuierlich-Ton-Originalbild.

[0015] Demgemäß besteht eine Aufgabe dieser Erfindung darin, ein Hochgeschwindigkeitssystem zum Umwandeln eines Graustufenquellenbildes in ein binäres Halbtonzielbild zu liefern.

[0016] Ein weiteres Ziel dieser Erfindung besteht darin, ein verbessertes System zum Umwandeln eines Graustufenquellenbildes in ein binäres Halbtonzielbild zu liefern, bei dem Speicherressourcen eingespart werden.

[0017] Ein weiteres Ziel dieser Erfindung besteht darin, ein verbessertes System zum Umwandeln von Graustufenquellenbildern mit einer Auflösung in binäre Zielbilder mit einer zweiten Auflösung zu liefern.

[0018] Ein weiteres Ziel dieser Erfindung besteht darin, ein verbessertes System zum Umwandeln von Graustufenquellenbildern in binäre Halbtonzielbilder zu liefern, bei dem eine Schwellenmatrix synchron und mit hoher Geschwindigkeit über ein Graustufenzielbild ausgerichtet und gefliest ist.

ZUSAMMENFASSUNG DER ERFINDUNG

[0019] Die Erfindung ist in den Patentansprüchen

dargelegt.

[0020] Ein System wandelt ein Quellenbild, das in einer ersten Auflösung angeordnete Quellenpixel umfasst, in Zielbildpixel um, die binär sind und in einer zweiten Auflösung angeordnet sind. Das System umfasst einen Speicher, der zumindest einen Teil einer Zeile von Quellengraustufenpixeln und eine entsprechende Zeile einer Schwellenmatrix mehrerer Zeilen von Graustufenpixelwerten, wobei jede Zeile zumindest W Pixelwerte umfasst, speichert. Ein Skalenregister speichert N Quellengraustufenpixelwerte, die skaliert wurden, um mit N Graustufenzielbildpixelpositionen übereinzustimmen. Ein Schwellenregister gibt parallel bis zu N der W Schwellenpixelwerte aus. Eine Ausrichtungslogik gibt die Schwellenpixelwerte auf steuerbare Weise in das Schwellenregister ein. Eine Steuerung betreibt die Ausrichtungslogik dahingehend, die W Schwellenpixelwerte in N zusammenhängende Speicherungspositionen in dem Schwellenregister einzugeben, und ermöglicht, wenn W größer als N ist, eine Eingabe von Überschüssigen der W Schwellenpixelwerte in zusammenhängende Speicherungspositionen in dem Schwellenregister, nachdem die (W – N) zuvor eingegebenen Schwellenpixelwerte ausgegeben wurden. Ein Komparator vergleicht die N Quellenpixelwerte mit Entsprechenden der N Schwellenpixelwerte und weist dem Zielbild gemäß dem Vergleichsvorgang N binäre Werte zu.

BESCHREIBUNG DER ZEICHNUNGEN

[0021] Fig. 1 ist ein schematisches Diagramm, das den Prozess des Skalierens eines Quellenbilds; des Ausrichtens und Fliesens einer Schwellenmatrix, um mit dem skalierten Quellenbild übereinzustimmen; und des Vergleichens des skalierten Quellenbildes und der gefliesten Schwellenmatrix, um bei einem Zweipegel-Zielbild anzukommen, veranschaulicht.

[0022] Fig. 2a veranschaulicht ein Quellenbild, das mehrere Graustufenpixelwerte aufweist.

[0023] Fig. 2b veranschaulicht das Quellenbild der Fig. 2a nach einem Hochskalieren um einen Faktor von 1,5×.

[0024] Fig. 2c veranschaulicht das Quellenbild der Fig. 2a nach einem Herabskalieren um einen Faktor von 0,75×.

[0025] Fig. 3 veranschaulicht ein bekanntes Verfahren zum Hochskalieren eines Quellenbildes unter Verwendung von Mehrbit-Eingabeindexarrays und einer Pixelreplikationsprozedur.

[0026] Fig. 4 veranschaulicht ein bekanntes Verfahren zum Hochskalieren eines Quellenbildes unter Verwendung von Mehrbit-Eingabeindexarrays, bei dem ein bilinearer Skalierungsalgorithmus verwendet wird.

[0027] Fig. 5 veranschaulicht ein bekanntes Verfahren zum Herabskalieren eines Quellenbildes unter Verwendung von Mehrbit-Eingabeindexarrays und eines Unterabtastalgorithmus.

[0028] Fig. 6 ist ein grobes Blockdiagramm eines Systems, das die vorliegende Erfindung beinhaltet.

[0029] Fig. 7 ist ein logisches Flussdiagramm der Erfindung, das eine Interaktion zwischen verschiedenen Datenquellen während einer Umwandlung eines Graustufenquellenbildes in ein Zweipegel-Zielbild veranschaulicht.

[0030] Fig. 7a ist ein Flussdiagramm, das angibt, wie SRAM-Zugriffe gesteuert werden.

[0031] Fig. 8 ist ein Diagramm einer Rasterbildseitendarstellung, das vertikale Bänder veranschaulicht, die durch die Erfindung verarbeitet werden.

[0032] Fig. 9a veranschaulicht schematisch eine Unterteilung eines Speichers, wenn ein niedriger Skalenfaktor auf ein Quellenbild angewandt werden soll.

[0033] Fig. 9b veranschaulicht schematisch ein Unterteilen eines Speichers, wenn ein großer Skalenfaktor auf ein Quellenbild angewandt werden soll.

[0034] Fig. 10 ist ein Diagramm, das ein „relatives" Eingabeindexarray-Format veranschaulicht.

[0035] Fig. 11 ist ein Diagramm, das die Beziehung zwischen Quellendaten und Zieldaten veranschaulicht, wie sie durch den Wert von Indexbitpositionen eines relativen Eingabeindexarrays gesteuert wird.

[0036] Fig. 12 ist ein ausführliches Blockdiagramm, das eine Schaltung veranschaulicht, die gemäß der Erfindung arbeitet.

[0037] Fig. 13a–Fig. 13d veranschaulichen vier modengesteuerte Verbindungen zwischen einem Schwellenmatrixregister und einem Schwellenausgaberegister.

[0038] Fig. 14 veranschaulicht die Funktionsweise einer Schwellenmatrixausrichtungsprozedur.

AUSFÜHRLICHE BESCHREIBUNG DER ERFINDUNG

[0039] Unter Bezugnahme auf Fig. 6 ist ein System **60** angepasst, um ein Quellenbild, das mehrere Graustufenpixelwerte mit einer ersten Auflösung aufweist, in ein binäres Pixelzielbild mit einer zweiten

Auflösung umzuwandeln. Das System **60** kann in einem Personal-Computer, einen Drucker, eine Kombination eines Personal-Computers und Druckers oder in eine beliebige andere Kombination von Vorrichtungen integriert sein, bei der Graustufenpixeldaten in binäre Pixeldaten einer anderen Auflösung umgewandelt werden.

[0040] Das System **60** umfasst einen Prozessor **62** mit einem zugeordneten dynamischen Direktzugriffsspeicher (DRAM) **64**. Der Prozessor **62** kann ein Graustufenquellenbild **66** von einer externen Quelle empfangen haben oder kann ein Graustufenquellenbild **66** infolge der Verarbeitung einer internen Anwendung erzeugt haben. Um eine Umwandlung eines Quellenbildes **66** in ein binäres Zielbild zu erzielen, sind Steuerparameter erforderlich. Ein Skalenfaktor **68** definiert das Verhältnis von Quellenbildpixeln zu Zielbildpixeln. Bei den meisten Druckeranwendungen weist das Zielbild ein beträchtlich höheres Auflösungsniveau auf als das Quellenbild **66**. Somit gibt der Skalenfaktor **68** allgemein einen Wert zwischen 2 und 10 (z. B. 6,5) an, was bedeutet, dass für jedes Quellenbildpixel mehrere Zielpixel vorliegen.

[0041] Die Steuerparameter umfassen ferner eine Schwellenmatrix **70**, die, wie oben angegeben wurde, mehrere Zeilen von Schwellengraustufenpixelwerten umfasst, die jeweils vorgeben, ob ein in Bezug auf die Position entsprechendes skaliertes Quellenbildpixel in einen weißen oder schwarzen Punkt in dem Zielbild umgewandelt wird. Der Aufbau der Schwellenmatrix **70** ist derart, dass gewährleistet wird, dass die binären Daten in dem Zielbild eine so gute Graustufendarstellung liefern, wie sie bei der Auflösung des Zielbildes erhältlich ist. Da in der Technik eine Vielzahl von Schwellenmatrizes **70** bekannt sind, ist eine weitere Beschreibung derselben nicht erforderlich.

[0042] Der DRAM **64** umfasst ferner einen Schwellenmatrixanker, der definiert, wo in dem skalierten Quellenbild die Schwellenmatrix anfänglich positioniert werden soll, bevor sie über das skalierte Quellenbild „gefliest" wird. Der Schwellenmatrixanker **72** definiert allgemein einen Eckpixelwert des skalierten Quellenbildes. Ein Bildanker **74** definiert, wo in einem Quellenbild ein bestimmter Abschnitt vorliegt, der skaliert und in ein Binärpixelzielbit umgewandelt werden soll.

[0043] Der DRAM **64** speichert ferner ein RIIA (horizontal) **76**, deren Struktur nachfolgend ausführlich betrachtet wird. Das RIIA **76** ist ein Array von Binärwertindexbits, mit einem Bit für jede Spalte des Zielbildes. Das RIIA **76** wird durch einen Prozessor **62** gemäß einem Skalenfaktor **68** berechnet. Kurz gesagt wird das RIIA **76** verwendet, um Quellenbildpixel zu der Auflösung des Zielbildes zu skalieren. Wenn der Skalenfaktor **68** also sechs beträgt, liegen sechs Zielpixel für jedes Quellenpixel vor. Jedes RIIA-Indexbit definiert, ob ein entsprechendes Zielpixel eine Funktion eines aktuellen Quellenpixels oder eines nächsten Pixels ist. Auf diese Weise ermöglicht ein einzelnes Bit pro Spalte des Zielbildes ein vollständiges Skalieren des Quellenbildes zu der Zielbildauflösung unter Verwendung eines Minimums an Indexbits.

[0044] Eine ähnliche Funktion erfüllt das RIIA (vertikal) **78**, jedoch wird das RIIA **78** verwendet, um die Ähnlichkeit oder Verschiedenheit aufeinander folgender Zeilen von Zielbilddaten zu bestimmen. Sowohl das RIIA (horizontal) **76** als auch das RIIA (vertikal) **78** sind „relative" Eingabeindexarrays, da sie ermöglichen, dass Quellenpixelwerte auf der Basis von Beziehungen, die aus den Indexbitwerten bestimmt werden können, bestimmt werden.

[0045] Wie aus der nachfolgenden Beschreibung hervorgeht, wird die Umwandlung eines Quellenbildes **64** in ein Zielbild durch eine Schaltungsanordnung, die in einer anwendungsspezifischen integrierten Schaltung (ASIC – application specific integrated circuit) **80** enthalten ist, auf synchrone und rasche Weise bewerkstelligt. Die ASIC **80** kommuniziert über einen Datenbus **82** und eine Busschnittstelle **84** sowohl mit dem Prozessor **62** als auch dem DRAM **64**. Ein statischer Direktzugriffsspeicher (SRAM) **86** liegt auf der ASIC **80** vor und umfasst unterteilte Bereiche **88**, **90**, **92** und **94** zum Speichern zumindest eines Teils einer Zeile von Quellenbildpixeln, der Schwellenmatrix, eines RIIA (horizontal) für das spezifische Quellenbild sowie von binären Zielbilddaten, die sich aus dem Bildumwandlungsvorgang ergeben.

[0046] Ein Skalen- und Dithermodul **96** umfasst eine Zustandsmaschine **98**, die auf synchrone Weise eine Skalierung der Quellenbildpixel zu der Zielbildpixelauflösung und einen anschließenden Vergleich von skalierten Quellenbildpixeln mit entsprechend positionierten Schwellenmatrixpixeln durchführt, um eine Umwandlung in binäre Daten zu ermöglichen. Die Funktionen des SRAM **86** und des Skalen- und Dithermoduls **96** unterliegen der Steuerung der Steuerung **100**. Taktsignale werden sowohl an die Steuerung **100**, das Skalen- und Dithermodul **96** als auch an die Zustandsmaschine **98** angelegt, um einen synchronen Betrieb derselben zu ermöglichen, wenn entsprechende Quellenbilddaten, Schwellenmatrixdaten und RIIA-Indexbits in Register geladen wurden, die sich in der Zustandsmaschine **98** befinden. Die Funktionsweise der Steuerung **100** in Kombination mit dem SRAM **86**, der Zustandsmaschine **98** und der Busschnittstelle **84** wird aus der nachfolgenden ausführlichen Beschreibung verständlich.

[0047] Fig. 7 ist ein grobes logisches Flussdiagramm, das eine Interaktion zwischen den verschie-

denen Datenquellen, die durch die Erfindung verwendet werden, veranschaulicht. Ferner veranschaulicht Fig. 7 eine Mehrzahl von Zeigern, die durch die Zustandsmaschine **98** verwendet werden, um die Funktionsweise der Erfindung zu steuern. Der SRAM **86** (Fig. 6) speichert Quellendaten **88**, eine Schwellenmatrix **90**, Horizontalskalierungsinformationen **92** (RIIA) und Zweipegel-Zieldaten **94**. Ein Quellenzeiger wird von Quellenbildpixel zu Quellenbildpixel inkrementiert, da jedes Quellenbildpixel einer Skalierung in dem Skalierungslogikmodul **102** unterworfen wird. Bei einem spezifischen Ausführungsbeispiel der Erfindung werden 4 Bytes (32 Bits) von Graustufenquellenpixeldaten der Skalierungslogik **102** zugeführt, und 4 Indexbits werden gleichzeitig von den Horizontalskalierungsinformationen **92** zugeführt. Ein Skalierungsstartzeiger gibt den nächsten Teilsatz von Skalierungsinformationen an, der der Skalierungslogik **102** zugeführt werden soll.

[0048] Bei Fig. 7a ist ein grobes Flussdiagramm gezeigt, das die Prozedur für SRAM-Zugriffe veranschaulicht. Jeder Schritt, der mit „Zyklus" bezeichnet ist, gibt an, dass ein Maschinenzyklus verwendet werden muss, um die Aufgabe zu bewerkstelligen. Ansonsten wird eine Aufgabe bewerkstelligt, ohne die Leistungsfähigkeit der Prozedur zu beeinflussen. Wenn ein Zyklus ausgeführt wird, wird der Zeiger zu einer Position in SRAM inkrementiert.

[0049] Anfänglich wird ein Zugriff auf Schwellendaten durchgeführt (Kästchen **150**). Unter gewissen Umständen (die später zu beschreiben sind) werden ausgegebene geflieste Schwellwerte ungültig und werden ignoriert. Dieser Vorgang ist durch das Entscheidungskästchen **152** angegeben, das ermöglicht, dass die Prozedur fortfährt, falls die Schwellendaten gültig sind. Falls dann zusätzliche Quellendaten benötigt werden (Entscheidungskästchen **154**), wird ein Zugriff auf Quellendaten aus dem SRAM zur Verwendung durch die Skalierungslogik **102** durchgeführt (Kästchen **156**). Ein Vergleichsvorgang zwischen den Schwellendaten und den skalierten Quellendaten wird anschließend durchgeführt, und resultierende binäre Zielpixelwerte werden zwischengespeichert (Kästchen **158**). Ein Halbbyte-Zählwert von Zielpixeln wird aufrechterhalten und wird jedes Mal inkrementiert, wenn vier Zielpixel zwischengespeichert werden (Kästchen **160**). Falls der Halbbyte-Zählwert nicht gleich acht ist, was angibt, dass 32 Zielpixelwerte noch nicht zwischengespeichert wurden, durchläuft die Prozedur einen erneuten Zyklus (Entscheidungskästchen **162**). Andernfalls werden die zwischengespeicherten Zweipegel-Zielpixeldaten in den SRAM gespeichert (Kästchen **164**), zusätzliche RIIA-Daten werden aus dem SRAM wiedergewonnen (Kästchen **166**), der Halbbyte-Zählwert wird gelöscht (Kästchen **168**) und die Prozedur durchläuft einen erneuten Zyklus.

[0050] Unter erneuter Bezugnahme auf Fig. 7 steuern die empfangenen Indexbits in der Skalierungslogik **102** das Umschalten einzelner Quellenbildpixel auf einen von vier Byteausgaben aus der Skalierungslogik **102**. Diese Ausgaben speisen ein Vergleichs- und Zwischenspeicherungs-Zweipegeldaten-Modul **104**. Da die Skalierungsoperation allgemein eine beträchtliche Erhöhung der Auflösung von den Quellenbilddaten zu den Zielbilddaten (z. B. in der Größenordnung von 6 oder mehr) beinhaltet, ist es oft der Fall, dass jedes Quellenpixel (unter der Annahme, dass eine Pixelreplikation verwendet wird) unter der Steuerung der Indexbits 6-mal oder häufiger repliziert wird. Wie aus der nachfolgenden Beschreibung der Fig. 8 hervorgeht, ermöglicht die Redundanz von Quellenbildpixeldaten auf einer Zeile-um-Zeile-Basis außerdem, dass viele Zugriffe auf den SRAM **86** eliminiert werden.

[0051] Gleichzeitig werden bei dem oben beschriebenen Skalierungsvorgang Schwellenmatrixdaten **90** unter der Steuerung eines Zeilenstartzeigers, eines Aktuelle-Position-Zeigers und eines Zeilenende-Zeigers einem Schwellenfliesungslogikmodul **106** zugeführt. Darin wird die Schwellenmatrix mit einem Zielbildauflösungsankerpunkt ausgerichtet und anschließend repliziert, um eine geflieste Schwellenmatrix zu erzeugen, bei der jede Pixelposition des abschließenden Zielbildes einen entsprechend positionierten Schwellenpixelwert aufweist. Desgleichen manifestiert die Ausgabe der Skalierungslogik **102** eine Zielbildauflösung, bei der für jede Pixelposition in dem Zielbild ein Quellenbildpixel vorliegt.

[0052] Jedes skalierte Quellenbildpixel und ein entsprechend positionierter, gefliester Schwellenmatrixbildpixelwert wird anschließend in dem Vergleichs- und Zwischenspeicherungs-Zweipegeldaten-Modul **104** verglichen. Je nachdem, ob der Quellenpixelwert den entsprechenden Schwellenmatrixpixelwert überschreitet, gleich demselben oder geringer als derselbe ist, wird ein binäres Ausgangsbit bereitgestellt, um den Quellenpixelwert zu ersetzen. Diese binären Bits werden dem Zweipegel-Zieldatenspeicher **94** zugeführt. Der Zweipegel-Zeiger ermöglicht, dass das SRAM-Steuermodul **100** die binären Zielbits in die korrekten Zielbildpixelpositionen in dem SRAM **86** positioniert.

[0053] Fig. 7 veranschaulicht, dass die Erfindung sowohl eine Skalierung als auch eine Schwellenfliesung auf einer im Wesentlichen parallelen und unabhängigen Basis durchführt. Beide derartigen Aktionen werden unter der Steuerung angelegter Taktsignale synchron durchgeführt, nachdem die entsprechenden Daten in zugeordnete Eingaberegister geladen wurden. Nachdem jede Zeile des Zielbildes verarbeitet wurde, wird ferner ein RIIA (vertikal) verwendet, um anzugeben, ob, angesichts des Skalierungsfaktors des Quellenbilds, entsprechende Quellenpi-

xel in einer nächsten Zeile identisch mit den entsprechenden Pixeln in der vorhergehenden Zeile sind. Falls angenommen wird, dass ein Skalenfaktor von 6 oder mehr verwendet wird, so sind im Durchschnitt zumindest 5 aufeinander folgende Zeilen identisch, wenn die Verwendung eines Pixelreplikationsprotokolls gegeben ist.

[0054] Um zu ermöglichen, dass eine Verringerung der erforderlichen Zyklen den Bildumwandlungsvorgang bewerkstelligt (siehe Fig. 8), wird das Bild zum Verarbeiten in mehrere vertikale Bänder segmentiert. Dabei wird die Tatsache ausgenutzt, dass der Skalenfaktor in der vertikalen Richtung derart ist, dass auf die Quellendaten nicht für jede folgende Zeile zugegriffen werden muss und dass dieselben für aufeinander folgende Zeilen verwendet werden können – wodurch die zugeordneten Speicherzugriffszyklen vermieden werden.

[0055] Unter erneuter Bezugnahme auf Fig. 6 ist der SRAM **86** (aus Kostengründen) vorzugsweise ein RAM mit einem Zugriffskanal. Die Erfindung verschachtelt Zugriffe auf den SRAM **86** seitens der verschiedenen Logikmodule (siehe Fig. 7) und erlaubt, dass eine einzelne Adresse zu dem SRAM **86** auf eine korrekte Position bei jedem Taktzyklus zeigt. Der SRAM **86** ist bei jedem Taktzyklus aktiv, während die Erfindung Daten verarbeitet.

[0056] Der SRAM **86** kann zu dem Zeitpunkt, zu dem ein Bild rasterisiert wird, unterteilt werden. Es gibt vier Objekte einer variablen Größe in dem SRAM **86**, d. h. das Schwellenarray (das von Bild zu Bild variieren kann), das RIIA, die Quellenpixeldaten und die Zweipegel-Zieldaten. Durch Verwendung eines einzelnen SRAM **86** kann die Steuerung **100** eine Unterteilung desselben erzielen, wenn unterschiedliche Skalenfaktoren und eine konstante Schwellenmatrix gegeben sind, und kann eine äußerst effiziente Nutzung seiner Speicherkapazität erzielen. Wie in Fig. 9a gezeigt ist, weist die Steuerung **100**, wenn ein kleiner Skalenfaktor vorliegt, somit beträchtlich mehr Speicherraum den Quellenbilddaten zu als den Zweipegel-Zieldaten (da die Quellenbilddaten 8 Bits pro Pixel aufweisen, wohingegen die Zweipegel-Daten ein einzelnes Bit pro Pixel aufweisen). Im Gegensatz dazu wird, wenn ein großer Skalenfaktor verwendet wird (siehe Fig. 9b), ein beträchtlich größerer Bereich des SRAM **86** zur Speicherung der RIIA-Bits und der Zweipegel-Daten zugewiesen.

[0057] Bevor ein spezifisches Beispiel der Erfindung beschrieben wird, wird das relative Eingabeindexarray (RIIA) beschrieben. Das RIIA ist ein äußerst effizientes Datenarray, das ein Skalieren von Quellenbildgraustufendaten zu der Auflösung der Zielbilddaten ermöglicht. Das RIIA-Format ist in Fig. 10 gezeigt und weist ein 4-Byte-Anfangsblockfeld **120** und ein Eingabeindexfeld **122** einer variablen Länge auf, das definiert, welches Quellenbildpixel bei jedem Skalierungsvorgang verwendet werden soll, um ein Quellenbild zu erzielen, das auf die Auflösung des Zielbildes skaliert ist. Für einen 600-dpi-Drucker beträgt der Eingabeindexabschnitt **122** des horizontalen RIIA 600 dpi × 8 Zoll = 4800 Bits oder 600 Bytes.

[0058] Der Anfangsblock **120** umfasst die folgenden Felder: Ein 8-Bit-DS-Indexfeld **124**, das eine minimale Anzahl von Quellenpixeln spezifiziert, die bei jeder Suche nach einem nächsten Quellenpixel, das bei einem Herabskalierungsvorgang verwendet werden soll, vorgerückt werden sollen. Für einen Einheits- oder Hochskalierungsvorgang ist dieses Feld „0". Ein src-Überspringungsfeld **126** ist eine 16-Bit-Menge, die eine Anzahl von Pixeln angibt, die übersprungen werden sollen, um ein erstes Pixel eines Quellenbildes zu finden, das skaliert werden soll.

[0059] Der Eingabeindex **122** weist ein einzelnes Bit pro Spalte de Zielbildes auf. Er ist einem Quellenpixelzeiger zugeordnet, der beim Skalieren verwendet wird und der auf ein aktuelles Quellenpixel zeigt, das in einer Zielbittabelle verwendet wird. Der Quellenpixelzeiger wird auf das Pixel initialisiert, das durch src Überspringen spezifiziert wird, d. h. auf das erste Pixel, das während eines Skalierungsvorgangs durch die Vorrichtung berücksichtigt werden soll.

[0060] Es gibt zwei allgemeine Formen des Eingabeindex.

Fall der Einheit oder des Hochskalierens:

[0061] In einem solchen Fall enthält das DS-Indexfeld **124** einen Wert von 0. Eine „0" in einer Bitposition des Eingabeindex **122** gibt an, dass das durch einen Quellenpixelzeiger angegebene Pixel in dem Zielbild verwendet werden soll. Eine „1" in einer Bitposition des Eingabeindex **122** gibt an, dass das durch (Quellenpixelzeiger +1) angegebene Pixel in dem Zielbild verwendet werden sollte und dass der Quellenpixelzeiger anschließend inkrementiert werden soll.

Fall des Herabskalierens:

[0062] Dieser Fall wird spezifiziert, wenn das DS-Indexfeld **124** einen Wert zwischen 1 und 255 enthält. Das erste Quellenpixel wird als das erste Zielpixel verwendet, ungeachtet des Werts des ersten Bits des Eingabeindex **122**. Für nachfolgende Einträge in dem Array gibt ein „0"-Wert an, dass das entsprechende Zielpixel das Quellenpixel sein wird, auf das durch (Quellenpixelzeiger + DS-Index) gezeigt wird. Eine „1" gibt an, dass das entsprechende Zielpixel das Quellenpixel sein wird, auf das durch (Quellenpixelzeiger + DS-Index +1) gezeigt wird. In jedem Fall wird der Quellenpixelzeiger auf das verwendete neue Quellenpixel aktualisiert.

[0063] Das RIIA (vertikal) funktioniert auf dieselbe Weise, jedoch werden statt Spaltenpixeln Zeilen skaliert.

[0064] Fachleute werden erkennen, dass die obige Beschreibung spezifische Bit- und Byte-Längen zu Beispielszwecken verwendet und dass auch andere Datenlängen verwendet werden können. Jedoch sollte der Eingabeindex **122** nie mehr als 1 Bitposition pro Spalte von Zielbildzeilendaten aufweisen.

[0065] Unter Bezugnahme auf Fig. 11 wird ein Beispiel der Verwendung eines RIIA (horizontal) beschrieben. Angenommen, dass die Quellenbilddaten hochskaliert werden, beträgt der Wert des DS-Index **124** 0; die Anzahl von Quellenpixeln, die beim Finden eines nächsten Quellenpixels, das bei dem Ziel verwendet werden soll, vorgerückt werden sollen, beträgt 0; und src Überspringen = 0. Das erste Quellenpixel, das beim Aufbauen des Zielbildes verwendet werden soll, ist dann das erste Quellenpixel, und es sollen keine übersprungen werden.

[0066] Angenommen, dass die ersten vier Quellenpixel jeweils 8-Bit-Werte sind, die Graustufen von **235**, **142**, **228** und **250** entsprechen, ist ihre Positionierung in dem Quellenbild so, wie sie in Fig. 11 bei **130** gezeigt ist. In dem RIIA **132** liegt für jedes Byte in dem Zielbild **134** ein Eingabeindexbit vor. Unter Einhaltung der oben beschriebenen Regeln wird, wenn ein RIIA-Indexbit „0" ist, das „aktuelle" Quellenpixel in das Zielbild positioniert, und dieses Quellenpixel bleibt das aktuelle Quellenpixel. Falls das RIIA-Indexbit „1" ist, wird das „nächste" Quellenpixel in das Zielbild positioniert, und dieses nächste Pixel wird zum „aktuellen" Pixel. Bei dem in Fig. 11 gezeigten Beispiel gibt die erste 0 in dem RIIA **132** vor, dass der Quellenpixelwert von 235 in die erste Zielpixeldatenposition positioniert werden soll. Die zweite „0" in dem RIIA **132** platziert den Wert 235 gemäß den zuvor erwähnten Regeln in die Zieldaten **134**, da die Position des aktuellen Quellenpixels nicht inkrementiert wurde. Der nächste „1"-Wert in dem RIIA **132** gibt an, dass der Zeiger auf das Quellenbildpixel um Eins inkrementiert werden muss und dass der Pixelwert, auf den gezeigt wird, in die nächste verfügbare Zielbildpixelposition platziert werden muss, usw., usw.

[0067] Der Herabskalierungsvorgang wird für Fachleute aufgrund der zuvor beschriebenen Hochskalierungsprozedur offensichtlich.

[0068] Eine weitere ausführliche Implementierung der Erfindung ist in Fig. 12 gezeigt, und die folgende Beschreibung betrachtet einen Fall von 8-Bit-pro-Pixel-Daten und 4 Pixeln, die gleichzeitig gehandhabt werden. Die veranschaulichte Topologie kann skaliert werden, um jegliche Datenpixellänge bei jeglicher Anzahl von parallelen Pixeln zu handhaben.

[0069] Bei Fig. 12 bewirkt die Steuerung **100** (Fig. 6), dass vier 8-Bit-Quellenpixel in ein Bildquellenregister **200** eingefügt werden. Desgleichen bewirkt die Steuerung **100**, dass mehrere RIIA-Indexbits (horizontales RIIA) in ein RII-Register **202** eingefügt werden. Ferner wird eine Zeile aus der Schwellenmatrix in dem SRAM **86** in ein Schwelleneinspeicherungsregister **204** eingefügt. Die Steuerung **100** bewerkstelligt das Schreiben von Daten in Register **200**, **202** und aus dem SRAM zu Zeitpunkten, die während einer anschließenden Verarbeitung in dem System gewährleisten, dass alle Ausgangsdaten auf einen nachfolgenden Taktzyklus hin gültig sein werden. Dieser Vorgang gewährleistet, dass das gesamte System gemäß sequenziell angelegten Taktpulsen synchron betrieben werden kann. Demgemäß erfolgt das Skalieren von Quellenpixeln zu einer Zielauflösung und die Ausrichtung und das Verfliesen von Schwellenmatrixpixelwerten gleichzeitig und in einem Verriegelungsschritt mit angelegten Taktsignalen. Wenn ferner die Schwellenmatrixpixelwerte an Komparatoren **206** angelegt werden, ist gewährleistet, dass die skalierten Quellenbildpixel von dem skalierten Bildquellenregister **208** vorliegen und gültig sind.

[0070] Obwohl die Skalierungs- und Schwellenausrichtungsfunktionen gleichzeitig ablaufen, wird die Funktionsweise der Fig. 12 zunächst unter Bezugnahme auf das Skalieren der Quellenbildpixel und anschließend unter Bezugnahme auf die Ausrichtung und das Verfliesen der Schwellenmatrixpixelwerte beschrieben.

[0071] Nachdem die Steuerung **100** den Bildumwandlungsvorgang begonnen hat, bewirkt jedes an das RII-Register **202** angelegte Taktsignal, dass Indexbitwerte (in Vierergruppen) an einen Multiplexer **210** ausgegeben werden. Eine „Auswahl"-Eingabe in den Multiplexer **210** bewirkt eine Verbindung der vier Indexbits mit entsprechenden Spaltenpositionen in einer Skalenschaltlogik **212**. Gleichzeitig werden vier Quellenpixel von dem Bildquellenregister **200** zu der Skalenschaltlogik **212** getaktet. Gemäß den oben beschriebenen Regeln bewirken die angelegten Indexbits, dass Jeweilige der Quellenpixel auf mehreren Ausgabeleitungen **214** erscheinen und in das skalierte Bildquellenregister **208** eingegeben werden.

[0072] Falls in dem Bildquellenregister **200** nicht ausreichend Quellenbytes vorliegen, ist ein SRAM-Zugriff erforderlich. Jegliche Quellenpixelrestwerte, die in dem Bildquellenregister **200** verbleiben, werden gemäß den RIIA-Indexbits in das skalierte Bildquellenregister **208** geladen. Bei dem nächsten Zyklus wird auf die zusätzlichen Quellenpixel zugegriffen, und dieselben werden in das Bildquellenregister **200** geladen.

[0073] Wenn angenommen wird, dass ein Skalen-

faktor von 6 oder besser verwendet wird und dass die Skalierungsmethode eine Pixelreplikation ist, so wird ein 8-Bit-Quellenpixel auf jeder der Ausgangsleitungen **214** repliziert und auf identische Weise in jede der vier Registerpositionen des skalierten Bildquellenregisters **208** eingegeben. Das nächste Taktsignal bewirkt, dass jeder dieser Quellenpixelwerte an Komparatoren **206** ausgegeben wird.

[0074] Gleichzeitig mit dem obigen Skalierungsvorgang folgt eine Schwellenmatrixausrichtung und ein Verfliesen. Falls die Anzahl von Schwellenmatrixpixelwerten in jeder Zeile der Schwellenmatrix gleich der Anzahl von skalierten Bildquellenpixeln in dem Register **208** ist, ist die Ausrichtungsfunktion unkompliziert. Die Ausrichtung wird bewerkstelligt, indem ein „Mode"-Signal an ein Paar von Schwellenausrichtungsschaltern **216** und **218** angelegt wird. Wenn jedoch die Anzahl von Schwellenmatrixpixelwerten in einer Zeile der Schwellenmatrix gleich der Anzahl von skalierten Quellenbildpixeln ist, werden beide Schwellenausrichtungsschalter **216** und **218** betrieben, um zu bewirken, dass die Schwellenmatrixpixelwerte durch ein Schwellenpipelineregister **220** gespült und in entsprechenden Pixelpositionen in ein Schwellenregister **222** platziert werden.

[0075] In dem Fall, in dem die Anzahl von Schwellenmatrixpixelwerten (d. h. = B) in einer Zeile der Schwellenmatrix nicht gleich der Anzahl (d. h. = N) von skalierten Bildquellenpixeln aus dem skalierten Bildquellenregister **208** ist, ist das Problem der Ausrichtung komplexer. Unter derartigen Bedingungen müssen Elemente aus zwei nachfolgenden Schwellenmatrixpixelzeileneingaben verwendet werden, um eine ordnungsgemäße Ausrichtung der Schwellenmatrixzeilenwerte mit den skalierten Quellenbildpixelwerten zu bewerkstelligen. Aus diesem Grund wird das Schwellenpipelineregister **220** zusätzlich zu dem Schwellenregister **222** aufgenommen. Jeder Schwellenausrichtungsschalter **216** und **218** arbeitet auf multiplexe Weise, um Schwellenmatrixpixelwerte anzuleiten, die Schwellenmatrixpixelwerte mit den skalierten Bildquellenwerten auszurichten. Man beachte, dass die Ausgaben aus dem Schwellenausrichtungsschalter **216** nicht nur dem Schwellenpipelineregister **220** zugeführt werden, sondern auch direkt dem Schwellenausrichtungsschalter **218** zugeführt werden und das Schwellenpipelineregister **220** umgehen.

[0076] Jeder der Schwellenausrichtungsschalter **216** und **218** wird, wie zuvor erwähnt, durch Modeneingaben von der Steuerung **100** gesteuert. Wie in den Fig. 13a–Fig. 13d gezeigt ist, legt die Steuerung 100N gesonderte Modensignale gemäß der Breite B der Schwellenmatrixzeile und ihrer Beziehung zu den skalierten Bildquellendaten aus dem Register **208** an. Wie in Fig. 13a gezeigt ist, wird also der Mode 0 verwendet, wenn B gleich N ist (wobei N die Anzahl von Pixeln der skalierten Bildquellendaten in dem Register **208** ist). Jeder der in den Fig. 13b, Fig. 13c und Fig. 13d veranschaulichten anderen Moden ermöglicht das Schalten der aus dem SRAM ausgelesenen Schwellenpixelwerte, auf Pipeline-Art, in das Schwellenregister **222** in einer modifizierten Ausrichtung, die den Unterschied zwischen B und N berücksichtigt.

[0077] Das Schalten, das in jedem der Schwellenausrichtungsschalter **216** und **218** ansprechend auf verschiedene Modeneingaben erfolgt, wird unter Bezugnahme auf das in Fig. 14 gezeigte Beispiel verständlich. Angenommen, die Schwellenmatrix weist eine Breite B = 5 auf, die, wie sich herausstellt, ein am wenigsten effizienter und etwas einem „schlimmsten Fall" ähnelnder Umstand ist, in dem das System überhaupt arbeiten kann. Die jeweiligen Schwellenpixelwerte, die eine Schwellenbreite B aufweisen, werden hiernach als Bytes A, B, C, D, und E bezeichnet. Fig. 14 veranschaulicht die aufeinander folgenden Bytes von Schwellenpixelwerten nach einer Ausrichtung und Einfügung in das Schwellenregister **222**. Danach werden die jeweiligen Schwellenpixelwerte in Komparatoren **206** mit den skalierten Quellenbildpixeln aus dem Register **208** verglichen.

[0078] Fig. 14 liefert eine Takt-um-Takt-Darstellung der Inhalte des Schwelleneinspeicherungsregisters **204**, des Schwellenpipelineregisters **220** und des Schwellenregisters **222**. Während der Taktzyklen 1 und 2 legt die Steuerung **100** Mode = 0 an die Schwellenausrichtungsschalter **216** und **218** an. Folglich erfolgt eine durchgehende Pipelineverarbeitung von Schwellenmatrixpixelwerten aus dem SRAM in das Schwellenregister **220**.

[0079] Erst am Ende einer Schwellenzeile wird eine Modenänderung ausgelöst, und in diesem Fall geht die Steuerung **100** von Mode = 0 zu Mode = 3 über. Unter derartigen Umständen (in dem Taktzyklus 3) empfängt jede Registerposition in dem Schwellenpipelineregister **210** den Wert des Schwellenpixelbytes D, und die A-, B- und C-Schwellenpixelbytes werden direkt in die drei Bytepositionen der niedrigeren Ordnung in dem Schwellenregister **222** eingegeben. Aus dem vorherigen Taktzyklus befindet sich das Schwellenpixelbyte E in der Byteposition der höchsten Ordnung. Wenn also ein nachfolgender Taktzyklus erfolgt, werden die Schwellenpixelwerte E, A, B, und C in die Komparatoren **206** geschrieben. Während des Taktzyklus 4 wird das Byte D von dem Pipelineschwellenregister **220** in das Pipelineregister **222** eingegeben, und die Bytes E, A und B werden direkt aus dem SRAM **204** geladen. Der Vorgang setzt sich während der Taktzyklen 5-8 durch die Moden 2 und 1 fort, wie gezeigt, und wiederholt sich anschließend.

[0080] Man beachte, dass, wenn die Schwellenma-

trix gefliest ist, um mit der Zielbildauflösung zusammenzufallen, ein Versatz bei den Endbytes vorliegen kann (falls jede Schwellenzeile nicht ein ganzzahliger dezimaler Teil des Zielbildes ist). In einem solchen Fall werden die Schwellenmatrixbytewerte am Ende einer Zeile in einem SRAM „ausgefüllt", um ein gleichzeitiges Enden sowohl der Schwellenmatrixzeile als auch des Zielbildes zu gewährleisten. Beispielsweise werden während des Taktzyklus 4 die Bytes A und B verwendet, um das Wort auszufüllen, das in das Pipelineregister **222** eingegeben wird.

[0081] Aufgrund des Fehlens von eindeutigen Schwellenpixelwerten, die den 4 Bytes zugeordnet sind, die das letzte Byte (5.) der Schwellenpixelzeile enthalten, treten im Ergebnisstrom (d. h. Inhalte des Schwellenregisters **222**) Lücken auf. Beispielsweise befinden sich am Ende des Taktzyklus 4 C-Bytes in allen Positionen in dem Schwellenpipelineregister **220**. Damit ein gültiges Wort (4 Bytes) während des Taktzyklus 5 aus dem Pipelineregister **222** ausgegeben wird, sollten die zwei Bytes der höheren Ordnung C und D sein, in dem Schwellenpipelineregister **220** liegt jedoch kein D-Wert vor, der in das Pipelineregister **222** geladen werden könnte. Folglich ist das in das Pipelineregister **222** geladene Wort ungültig. Wenn das ungültige Wort an die Komparatoren **206** gesendet wird, ergeben sich ungültige Zweipegel-Daten. Jedoch hemmt die Steuerung **100** das Zwischenspeichern der Zweipegel-Ausgangsdaten in ein Zweipegel-Register **230** und hemmt ferner die Einfügung neuer skalierter Quellenpixel in das skalierte Bildquellenregister **208**. Wenn ein nächstes gültiges Wort aus dem Schwellenregister **222** ausgegeben wird, beginnt die Vergleichsprozedur von neuem.

[0082] Wiederum ist eine Schwellenmatrix mit einer B von 5 ein am wenigsten effizienter Fall, und das Modensignal ändert sich in jedem zweiten Taktzyklus. Drei der acht Taktzyklen ergeben ungültige Daten. Wenn sich das Modensignal für andere Werte von B nicht ändert, treten bei jedem Taktzyklus aus dem Schwellenregister **222** gültige Schwellenmatrix-4-Byte-Gruppen hervor, und je länger die Schwellenzeile ist, desto größer ist die Effizienz, da sich das Modensignal lediglich an einer Zeilengrenze ändern kann. Die üblichste Schwellenmatrix beträgt 16×16, und eine Verwendung dieser Matrix und einer beliebigen anderen Matrix, deren B ein Vielfaches von 4 ist, führt zu einen konstanten Modensignalbetrieb.

[0083] Unter erneuter Bezugnahme auf Fig. 12 sprechen die Komparatoren **206** auf Ausgaben aus dem Schwellenregister **222** und dem skalierten Bildquellenregister **208** an, indem sie die jeweiligen Graustufenwerte entsprechender Bytes vergleichen: Falls der skalierte Bildquellenpixelwert größer ist als der Schwellenregisterpixelwert, wird ein 1-Bit aus dem jeweiligen Komparator ausgegeben. Unter anderen Umständen wird ein 0-Bit aus dem jeweiligen Komparator ausgegeben und in ein 32-Bit-Zweipegel-Register **230** eingegeben. Wenn das 32-Bit-Zweipegel-Register **230** voll ist, wird sein Inhalt zur Speicherung in den SRAM **86** geschrieben.

[0084] Das System handhabt die ungültigen Wörter, die erzeugt werden, wenn B und N nicht identisch oder exakte Vielfache voneinander sind, indem es 2 Variablen verwendet. Die erste Variable ist „Mode", was oben erörtert wurde und bestimmt, wie Schwellenpixeldaten in das Schwellenpipelineregister **220** und Schwellenregister **222** geschaltet werden. Die zweite Variable wird als „Modendelta" bezeichnet und gibt an, wie stark die Variable „Mode" zu ändern ist, wenn ein Ende einer Schwellenzeile erfahren wird und eine nächste Zeile von Schwellenpixeldaten begonnen wird. Modendelta ist gleich der Anzahl von Bytes in einem Wort gesetzt, die „übrig bleiben", nachdem ein letztes Byte in einer Schwellenzeile verarbeitet wurde. Die Modendelta-Variable wird jedes Mal, wenn eine Schwellenzeile „neu begonnen" wird, angelegt.

[0085] Die zuvor erwähnten Variablen ermöglichen, dass, wenn B nicht gleich N ist, eine Schwellenmatrix verwendet wird und ferner bestimmt, wann N ungültige Bytes in dem Schwellenregister **222** vorliegen. Unter Verwendung lediglich der Mode-Variable und der Modendelta-Variable ist es möglich zu wissen, wenn ein ungültiges Wort aus der in Fig. 12 gezeigten Logik erzeugt wird.

[0086] Wie oben angegeben wurde, ist Modendelta gleich einer Anzahl von Schwellenpixeln, die in einer Schwellenzeile verbleiben. Der maximale Wert von Mode beträgt N – 1. Wenn Modendelta am Ende einer Schwellenzeile zu Mode hinzugefügt wird und das Ergebnis N – 1 überschreitet, so ist das erste Wort (z. B. 4 Bytes), das sich in dem Schwellenregister **222** befindet, zu Beginn einer neuen Schwellenmatrixzeile, ungültig. Bei dem oben angegebenen Beispiel liegen 4 Werte von Mode vor (d. h. 0 – 3), und der maximale Wert von Mode beträgt 3 (d. h. 4 – 1 = 3). Die Summe von Mode und Modendelta (Modulo N) bestimmt, was ein nächster Modenwert sein wird. Wenn Modendelta z. B. 2 und Mode 3 ist, ist die Summe derselben 1 (Modulo 4), und Mode 1 ist der nächste angelegte Steuerzustand. Wenn ein Modendeltawert (d. h. = 3) zu einem aktuellen Modewert hinzugefügt wird und die Summe den Wert von 3 überschreitet, ist das Wort (4 Bytes), das aus dem Schwellenregister **222** austritt, ungültig.

[0087] Anhand eines Beispiels in einem Fall eines Neustarts einer Schwellenzeile wird der maximale Wert von 3 nicht überschritten, wenn der Wert von Mode 0 ist und ein Modendelta von 3 zu 0 hinzugefügt wird. Somit ist das erste Wort, das für einen Fall von Mode = 3 erzeugt wird, gültig. Wenn jedoch wiederum der Modendeltawert zu demselben hinzuad-

diert wird, ist das Ergebnis 6, was den maximalen Wert überschreitet, sodass das erste Wort ungültig ist. Der Modenwert ist das Ergebnis einer Modulo-4-Funktion – also wird er 2. Jedes Mal, wenn ein Sprung zurück zu dem Start einer Schwellenzeile erfolgt, wiederholt sich dieser Vorgang.

[0088] Man sollte verstehen, dass die vorstehende Beschreibung die Erfindung lediglich veranschaulicht. Fachleute können verschiedene Alternativen und Modifikationen ersinnen, ohne von der Erfindung abzuweichen. Demgemäß soll die vorliegende Erfindung alle derartigen Alternativen, Modifikationen und Variationen, die in den Schutzumfang der beigefügten Patentansprüche fallen, umfassen.

Patentansprüche

1. Ein System (**60**) zum Umwandeln eines Quellenbildes (**10**), das Quellenpixel umfasst, die in einer ersten Auflösung von Zeilen und Spalten angeordnet sind, wobei jedes Quellenpixel einen Mehrbitgraustufenwert manifestiert, in ein Zielbild (**18**) von binären Pixeln, die in einer zweiten Auflösung von Zeilen und Spalten angeordnet sind, wobei das System (**60**) folgende Merkmale aufweist:
eine Speichereinrichtung (**86**) zum Speichern zumindest eines Teils einer Zeile der Quellenpixel, wobei eine Matrix (**12**) mehrere Zeilen von Matrixpixelwerten aufweist, wobei jede Zeile zumindest W Matrixpixelwerte umfasst, wobei W eine Ganzzahl und > 1 ist und wobei ein Ankerwert eine anfängliche Pixelposition einer Zeile des Zielbildes (**18**) identifiziert:
eine Skalenregistereinrichtung (**200**, **212**, **208**) zum Speichern von N Quellenmehrbitgraustufenpixelwerten, wobei N eine Ganzzahl > 1 ist und sich von W unterscheiden kann;
eine Schwellenregistereinrichtung (**220**, **222**) zum Speichern einer Zeile von W Matrixpixelwerten und zum parallelen Ausgeben von bis zu N der Matrixpixelwerte;
eine zwischen die Speichereinrichtung (**86**) und die Schwellenregistereinrichtung (**220**, **222**) gekoppelte Ausrichtungslogikeinrichtung (**216**, **218**) zum steuerbaren Eingeben der Zeile von W Matrixpixelwerten aus der Speichereinrichtung (**86**) in die Schwellenregistereinrichtung (**220**, **222**); und
eine Steuereinrichtung (**100**), die mit der Ausrichtungslogikeinrichtung (**216**, **218**) gekoppelt ist und auf die Werte von W und N anspricht, zum Steuern der Ausrichtungslogikeinrichtung (**216**, **218**) dahingehend, die W Matrixpixelwerte in N zusammenhängende Speicherungspositionen in der Schwellenregistereinrichtung (**220**, **222**) einzugeben, und falls W > N, eine Eingabe von Überschüssigen der W Matrixpixelwerte in zusammenhängende Speicherungspositionen in der Schwellenregistereinrichtung (**220**, **222**), nachdem (W – N) zuvor eingegebene Matrixpixelwerte ausgegeben wurden, zu ermöglichen.

2. Das System (**60**) gemäß Anspruch 1, bei dem die Skalenregistereinrichtung (**200**, **212**, **208**) N Quellenmehrbitgraustufenpixelwerte, die skaliert wurden, um mit N Mehrbitgraustufenzielbildpixelpositionen übereinzustimmen, speichert, wobei das System ferner folgendes Merkmal aufweist:
eine mit der Schwellenregistereinrichtung (**220**, **222**) und der Skalenregistereinrichtung (**200**, **212**, **208**) gekoppelte Komparatoreinrichtung (**206**) zum Vergleichen der N Quellenmehrbitgraustufenpixelwerte mit Entsprechenden der aus der Schwellenregistereinrichtung (**220**, **222**) ausgegebenen N Matrixpixelwerte und zum Zuweisen von N binären Werten zu dem Zielbild (**18**) gemäß dem Vergleichen.

3. Das System (**60**) gemäß Anspruch 2, bei dem die Speichereinrichtung (**86**) ferner einen Ankerwert speichert, der eine anfängliche Pixelposition einer Zeile des Zielbildes (**18**) speichert, und bei dem die Steuereinrichtung (**100**) den Ankerwert verwendet, um die Speichereinrichtung (**86**) zu veranlassen, ein Matrixpixel, das ein erster Graustufenschwellwert ist, einer ersten Zeile der Matrix (**12**) an die Schwellenregistereinrichtung (**220**, **222**) auszugeben, um mit einem ersten Quellenpixelwert in der Skalenregistereinrichtung (**200**, **212**, **208**) ausgerichtet zu werden.

4. Das System (**60**) gemäß Anspruch 2, bei dem die Schwellenregistereinrichtung (**220**, **222**) ein Schwellenpipelineregister (**220**) und ein Schwellenregister (**222**) umfasst und die Ausrichtungslogikeinrichtung (**216**, **218**) einen zwischen die Speichereinrichtung (**86**) und das Schwellenpipelineregister (**220**) gekoppelten ersten Logikschalter (**216**) und einen zwischen das Schwellenpipelineregister (**220**) und das Schwellenregister (**222**) gekoppelten zweiten Logikschalter (**218**) aufweist, wobei sowohl das Schwellenpipelineregister (**220**) als auch das Schwellenregister (**222**) N Speicherungspositionen umfassen;
wobei die Steuereinrichtung (**100**) während aufeinander folgender Taktzyklen ein Modensteuersignal anlegt, um zu bewirken, dass erste N Matrixpixelwerte von der Speichereinrichtung (**86**) in das Schwellenpipelineregister (**220**) und anschließend in das Schwellenregister (**222**) eingegeben werden; und
wobei die Steuereinrichtung (**100**) während eines weiteren Taktzyklus ein Modensteuersignal anlegt, um ferner zu bewirken, dass die Matrixpixelwerte von der Speichereinrichtung (**86**) in das Schwellenpipelineregister (**220**) eingegeben werden, wobei, falls W > N, bewirkt wird, dass Überschüssige der ersten Matrixpixelwerte von der Speichereinrichtung (**86**) in zusammenhängende Positionen des Schwellenregisters (**222**) eingegeben werden.

5. Das System (**60**) gemäß Anspruch 4, bei dem ein erstes Moduszustandssignal durch die Steuereinrichtung (**100**) angelegt wird, wenn W = N, und den ersten und den zweiten Logikschalter (**216**, **218**) ver-

anlasst, die N Schwellenpixelwerte sowohl in das Schwellenpipelineregister (**220**) als auch in das Schwellenregister (**222**) einzugeben.

6. Das System (**60**) gemäß Anspruch 4, bei dem ein zweites Moduszustandssignal durch die Steuereinrichtung (**100**) angelegt wird, wenn W = N + 1, und den ersten und den zweiten Logikschalter (**216**, **218**) veranlasst, einen N + 1. Matrixpixelwert von dem Schwellenpipelineregister (**220**) in eine höherrangige Speicherungsposition in dem Schwellenregister (**222**) und Zusätzliche eines nächsten Satzes von N Matrixpixelwerten in niedrigerrangige Positionen in dem Schwellenregister (**222**) einzugeben.

7. Das System (**60**) gemäß Anspruch 4, bei dem Ausgaben aus dem Schwellenregister (**222**) ungültige Daten manifestieren können, wobei die Steuereinrichtung (**100**) eine Ausgabe der ungültigen Daten aus dem System (**60**) hemmt.

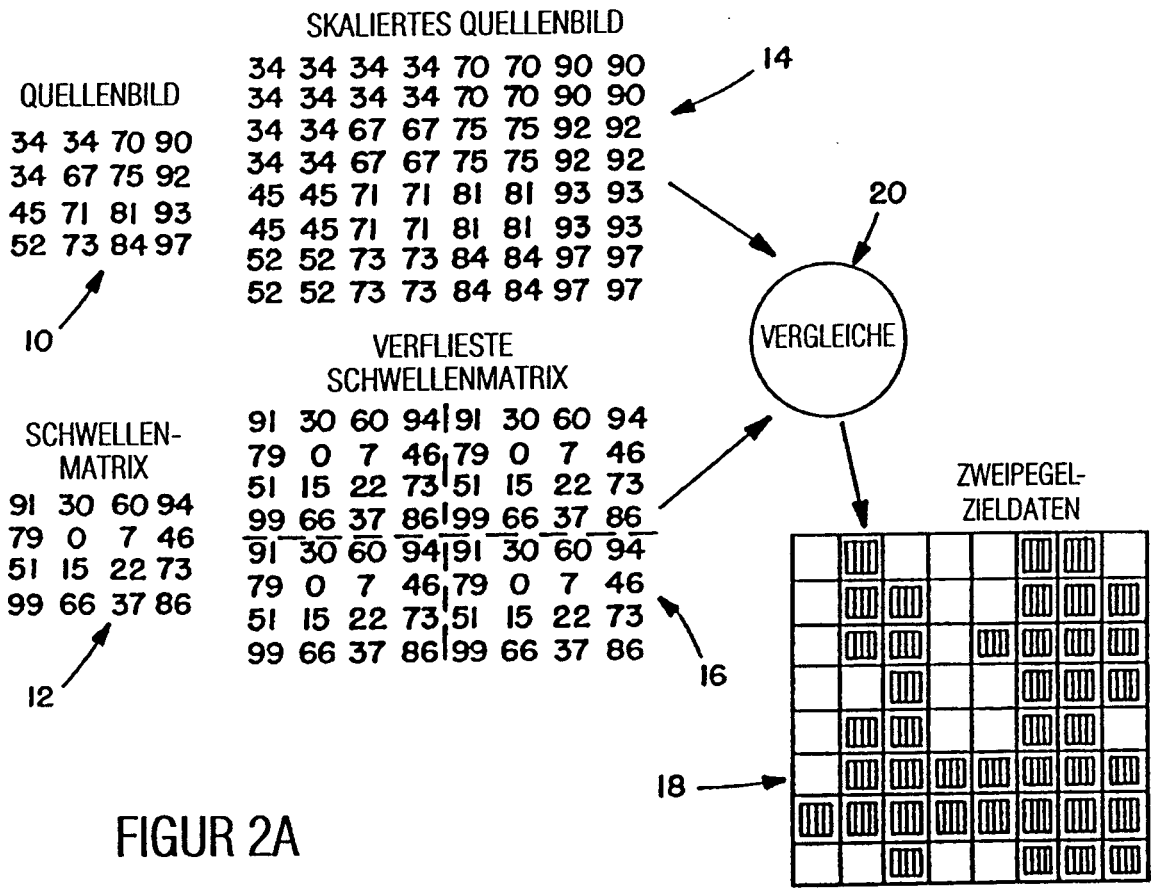
8. Das System (**60**) gemäß Anspruch 7, bei dem die Steuereinrichtung (**100**) zwei Variablen verwendet, um zu bestimmen, wann die Ausgabe zu hemmen ist, wobei eine erste Variable gleich einer Zahl < N von Matrixpixel ist, die an einem Ende einer Zeile der Matrix (**13**) verbleiben, und wobei eine zweite Variable gleich Einem einer Anzahl von Schaltzuständen ist, die durch die Steuereinrichtung (**100**) an die Ausrichtungslogikeinrichtung (**216**, **218**) angelegt werden, wobei die Steuereinrichtung (**100**) die Ausgabe hemmt, wenn eine Summe der ersten Variable und der zweiten Variable die Anzahl der Schaltzustände übersteigt.

9. Das System (**60**) gemäß Anspruch 8, bei dem die Steuereinrichtung (**100**) die Summe verwendet, um einen nächsten Schaltzustand einzuleiten.

Es folgen 9 Blatt Zeichnungen

Anhängende Zeichnungen

FIGUR 1



FIGUR 2A

235	142	228	250
1	2	45	65
69	29	11	189
254	201	120	240

22

QUELLENDATEN

FIGUR 2B

235	235	142	228	228	250
235	235	142	228	228	250
1	1	2	45	45	65
69	69	29	11	11	189
69	69	29	11	11	189
254	254	201	120	120	240

24

UM 1,5X HOCHSKALIEREN

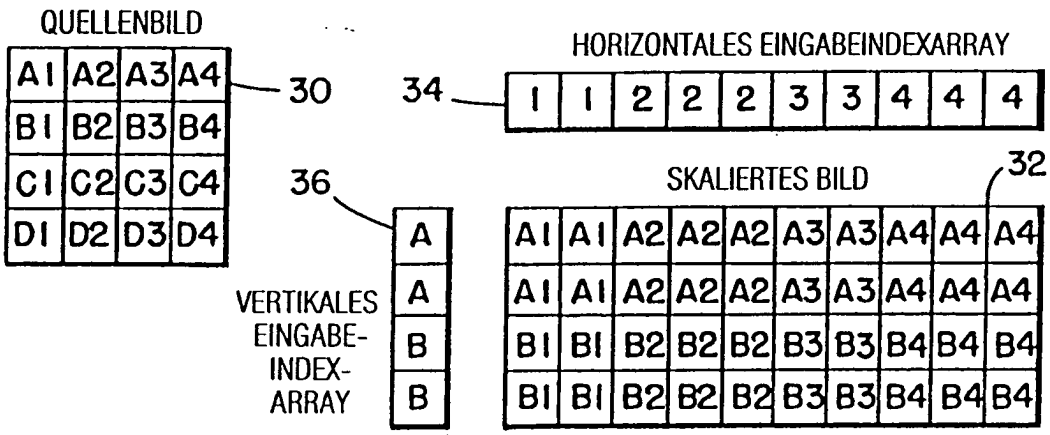
FIGUR 2C

235	142	250
1	2	65
254	201	240

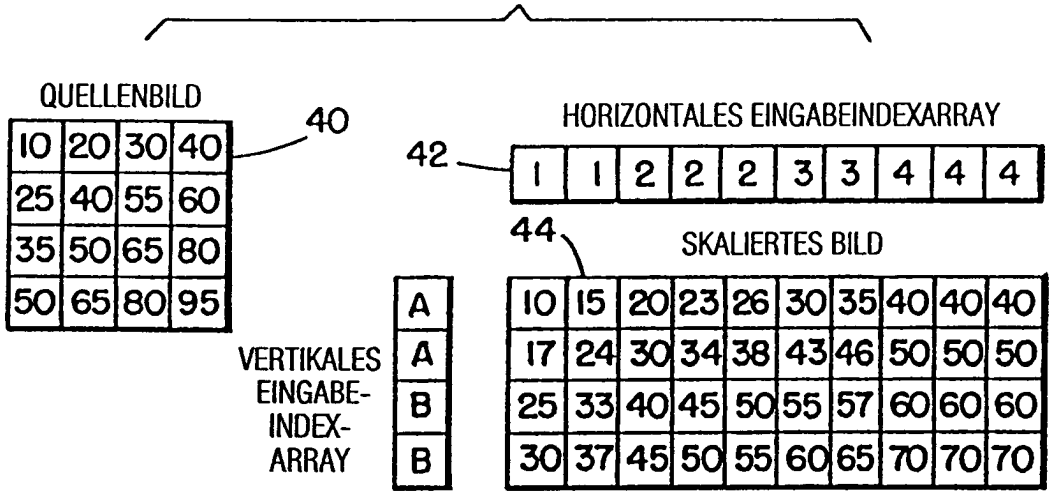
26

UM 0,75X HERUNTERSKALIEREN

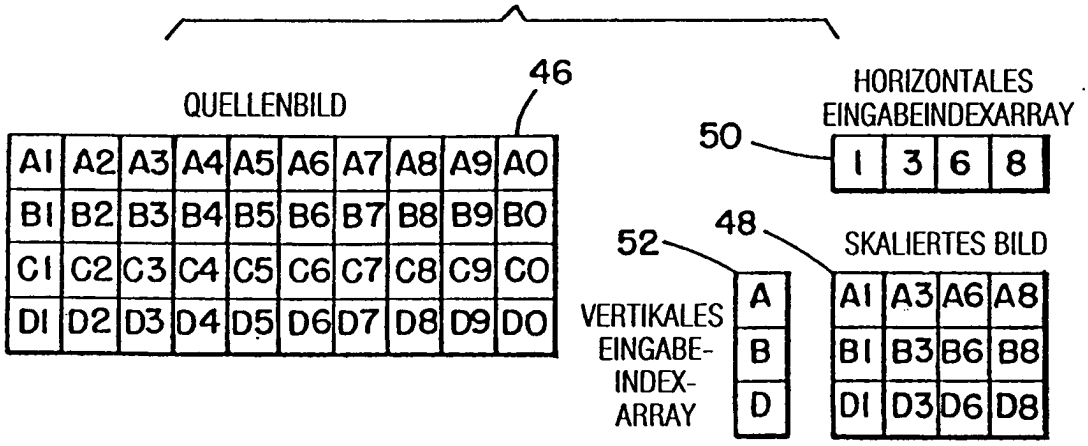
FIGUR 3



FIGUR 4



FIGUR 5



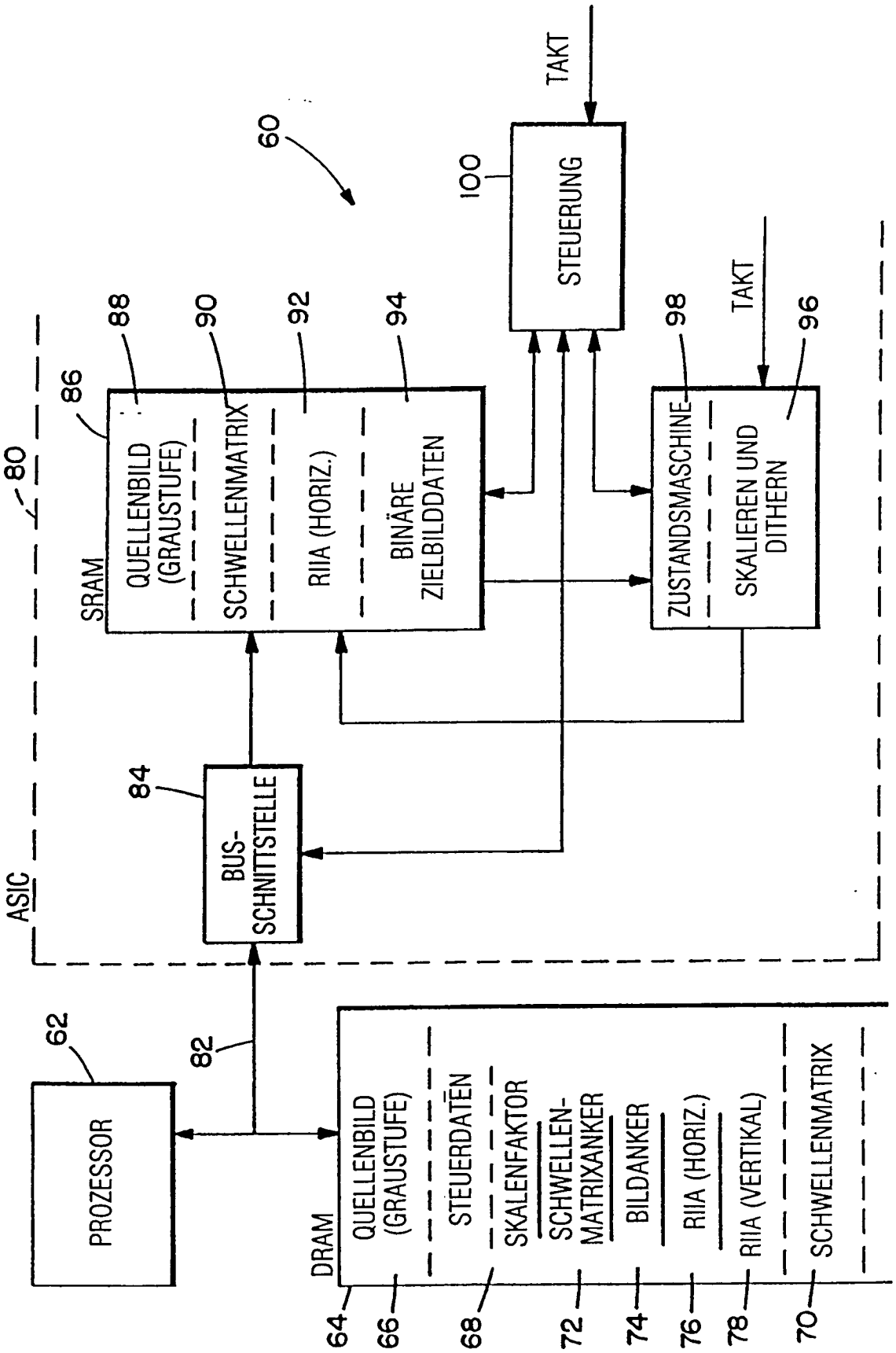
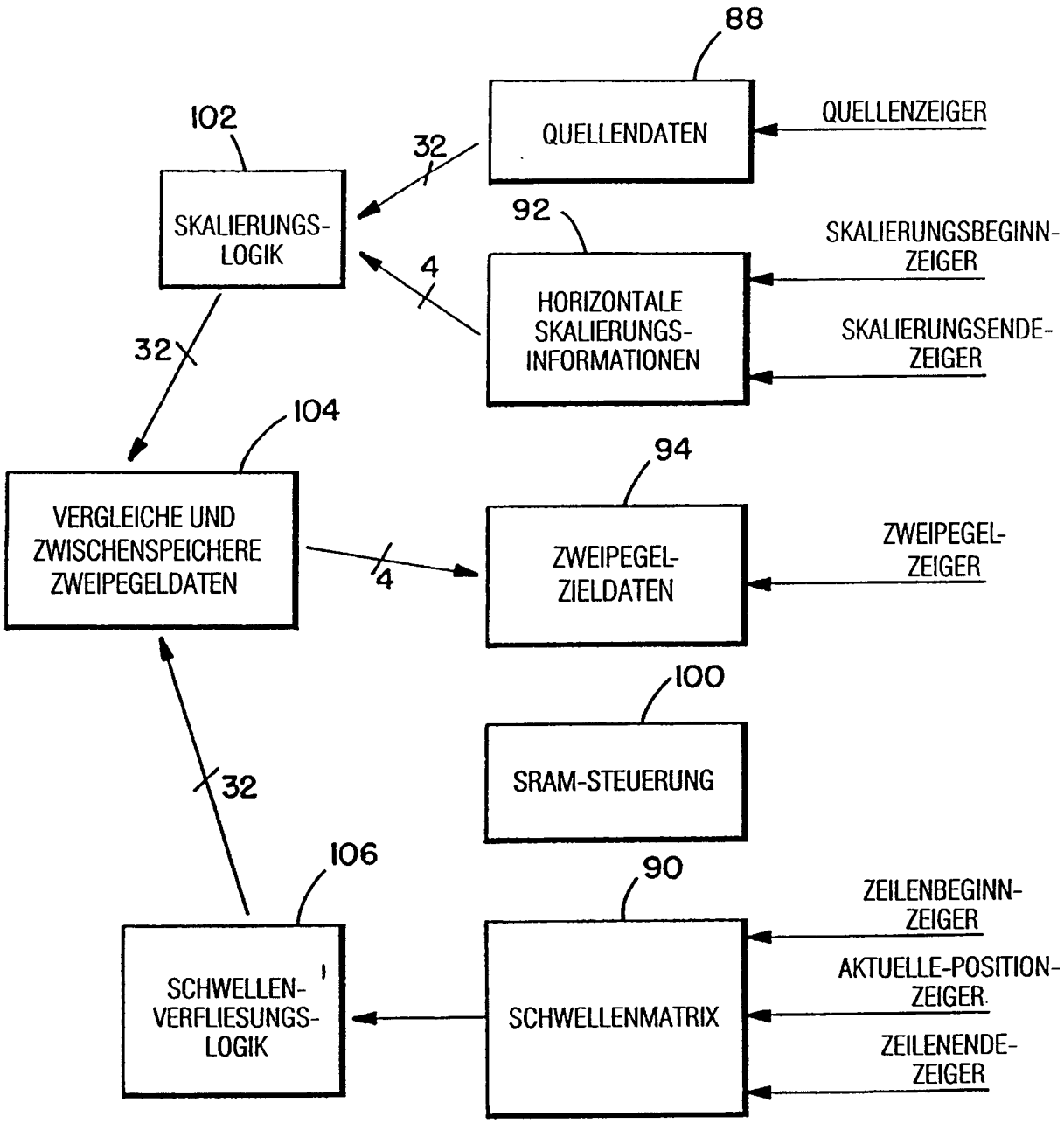
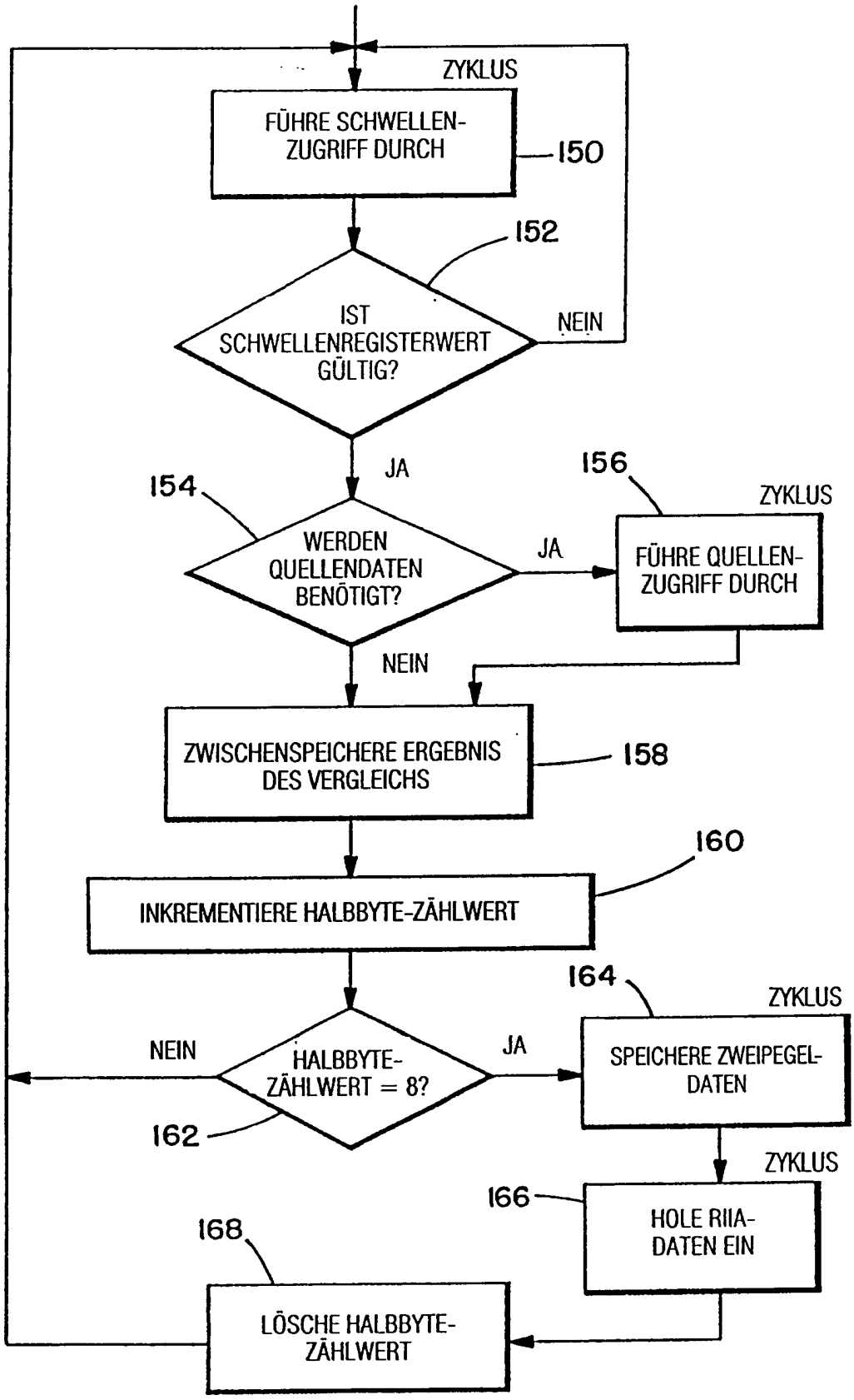

60
TAKT
100
STEUERUNG
80
ASIC
86
88
90
92
94
SRAM
QUELLENBILD (GRAUSTUFE)
SCHWELLENMATRIX
RIIA (HORIZ.)
BINÄRE ZIELBILDDATEN
98
TAKT
96
ZUSTANDSMASCHINE
SKALIEREN UND DITHERN
84
BUS-SCHNITTSTELLE
62
PROZESSOR
82
DRAM
QUELLENBILD (GRAUSTUFE)
STEUERDATEN
SKALENFAKTOR
SCHWELLEN-MATRIXANKER
BILDANKER
RIIA (HORIZ.)
RIIA (VERTIKAL)
SCHWELLENMATRIX
64
66
68
72
74
76
78
70

FIG. 6

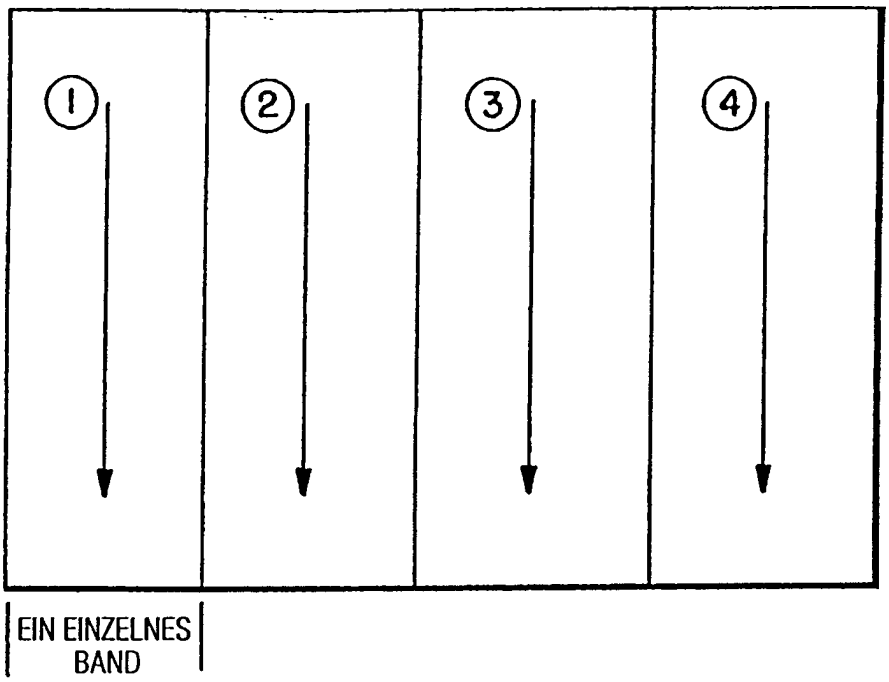
FIGUR 7


88
QUELLENDATEN
QUELLENZEIGER
102
32
SKALIERUNGS-
LOGIK
92
SKALIERUNGSBEGINN-
ZEIGER
4
HORIZONTALE
SKALIERUNGS-
INFORMATIONEN
SKALIERUNGSENDE-
ZEIGER
32
104
VERGLEICHE UND
ZWISCHENSPEICHERE
ZWEIPEGELDATEN
94
4
ZWEIPEGEL-
ZIELDATEN
ZWEIPEGEL-
ZEIGER
100
SRAM-STEUERUNG
32
106
90
ZEILENBEGINN-
ZEIGER
SCHWELLEN-
VERFLIESUNGS-
LOGIK
SCHWELLENMATRIX
AKTUELLE-POSITION-
ZEIGER
ZEILENENDE-
ZEIGER

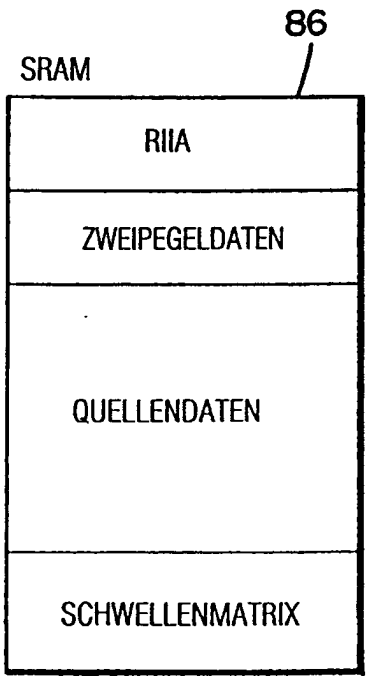
FIGUR 7A


ZYKLUS
FÜHRE SCHWELLEN-ZUGRIFF DURCH
150
152
IST SCHWELLENREGISTERWERT GÜLTIG?
NEIN
JA
154
WERDEN QUELLENDATEN BENÖTIGT?
156
ZYKLUS
JA
FÜHRE QUELLEN-ZUGRIFF DURCH
NEIN
ZWISCHENSPEICHERE ERGEBNIS DES VERGLEICHS
158
160
INKREMENTIERE HALBBYTE-ZÄHLWERT
164
ZYKLUS
NEIN
HALBBYTE-ZÄHLWERT = 8?
JA
SPEICHERE ZWEIPEGEL-DATEN
162
ZYKLUS
166
HOLE RIIA-DATEN EIN
168
LÖSCHE HALBBYTE-ZÄHLWERT

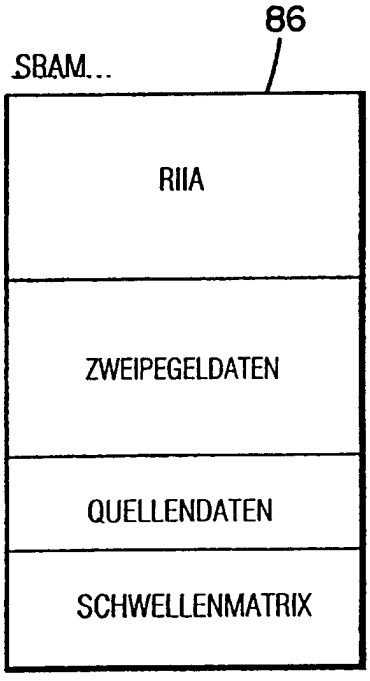
FIGUR 8



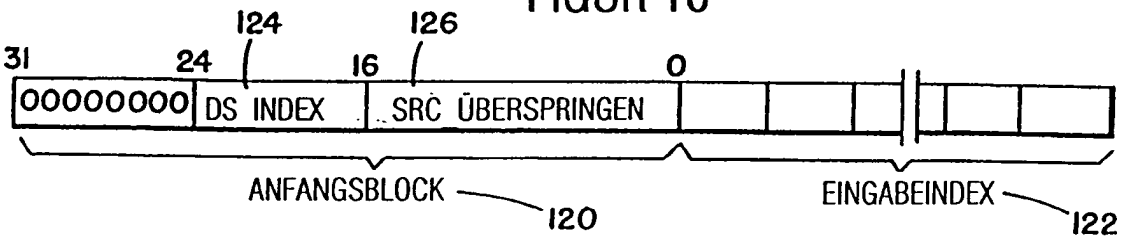
FIGUR 9A



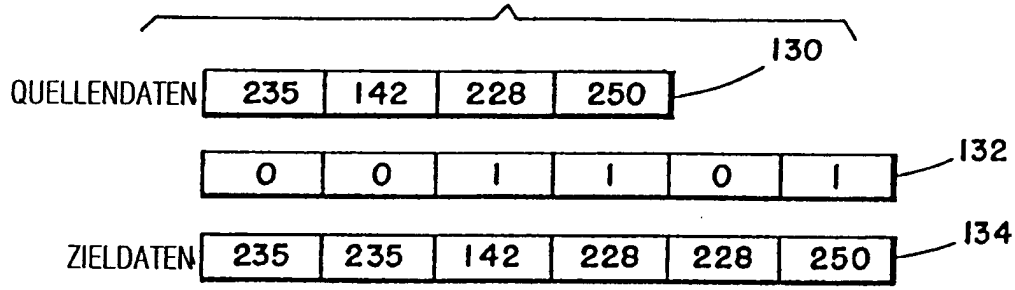
FIGUR 9B



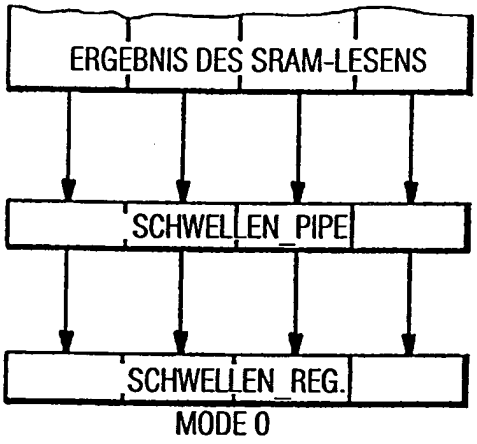
FIGUR 10



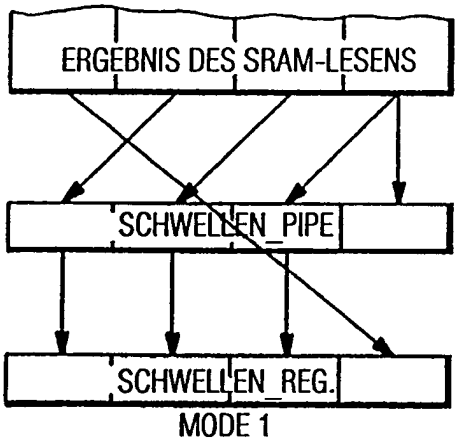
FIGUR 11



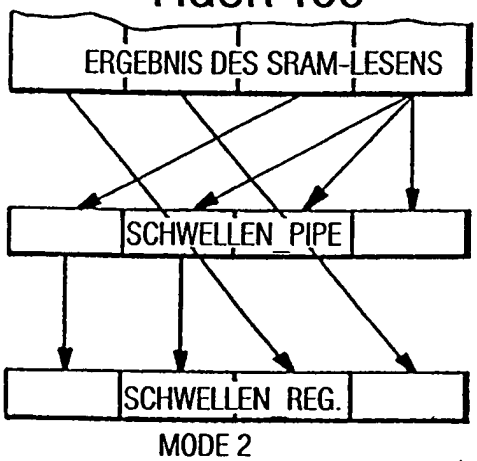
FIGUR 13A



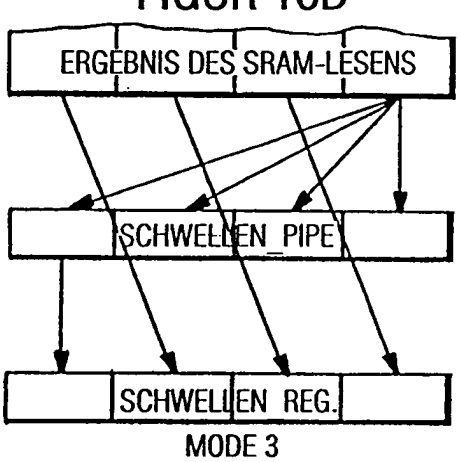
FIGUR 13B



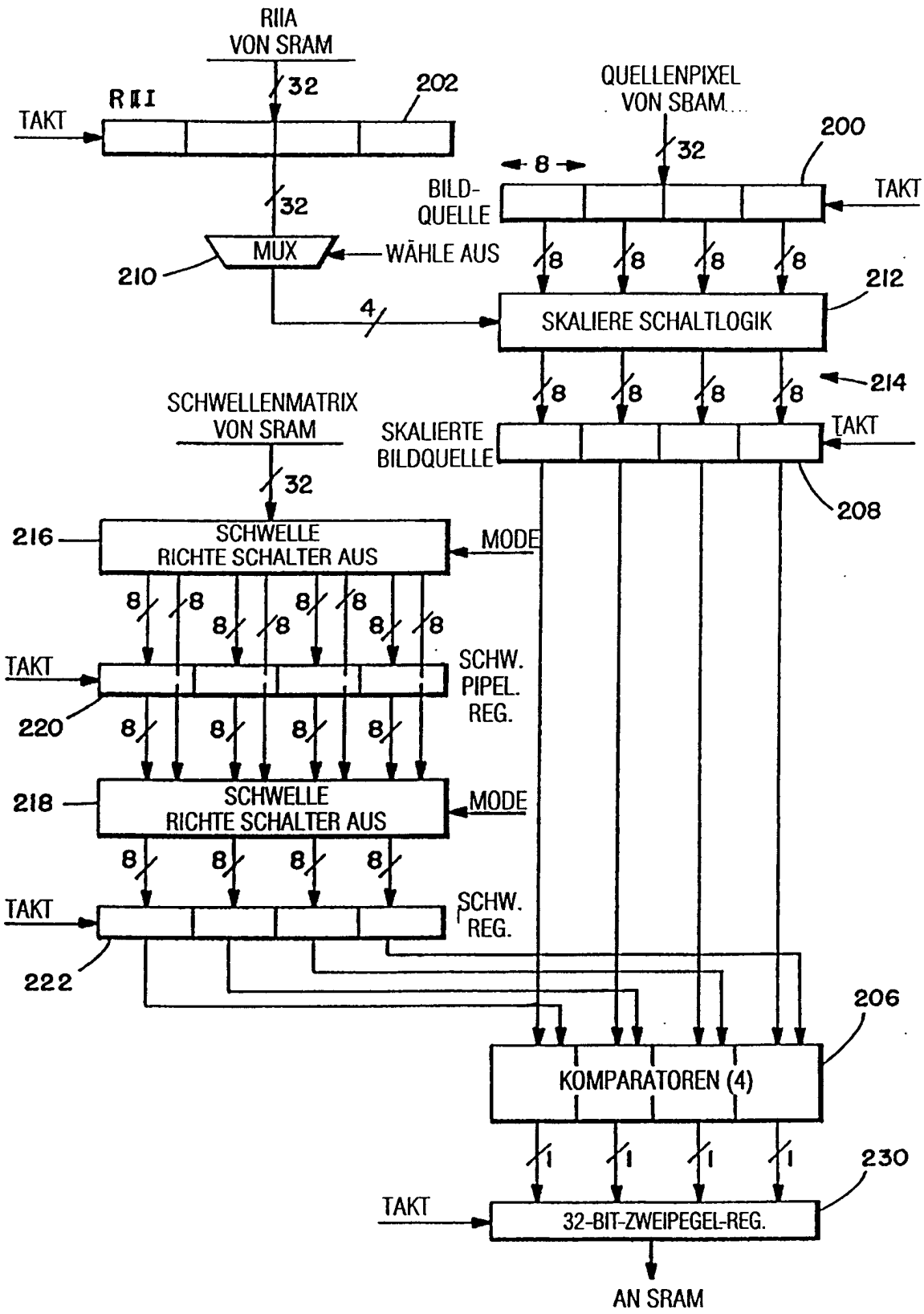
FIGUR 13C



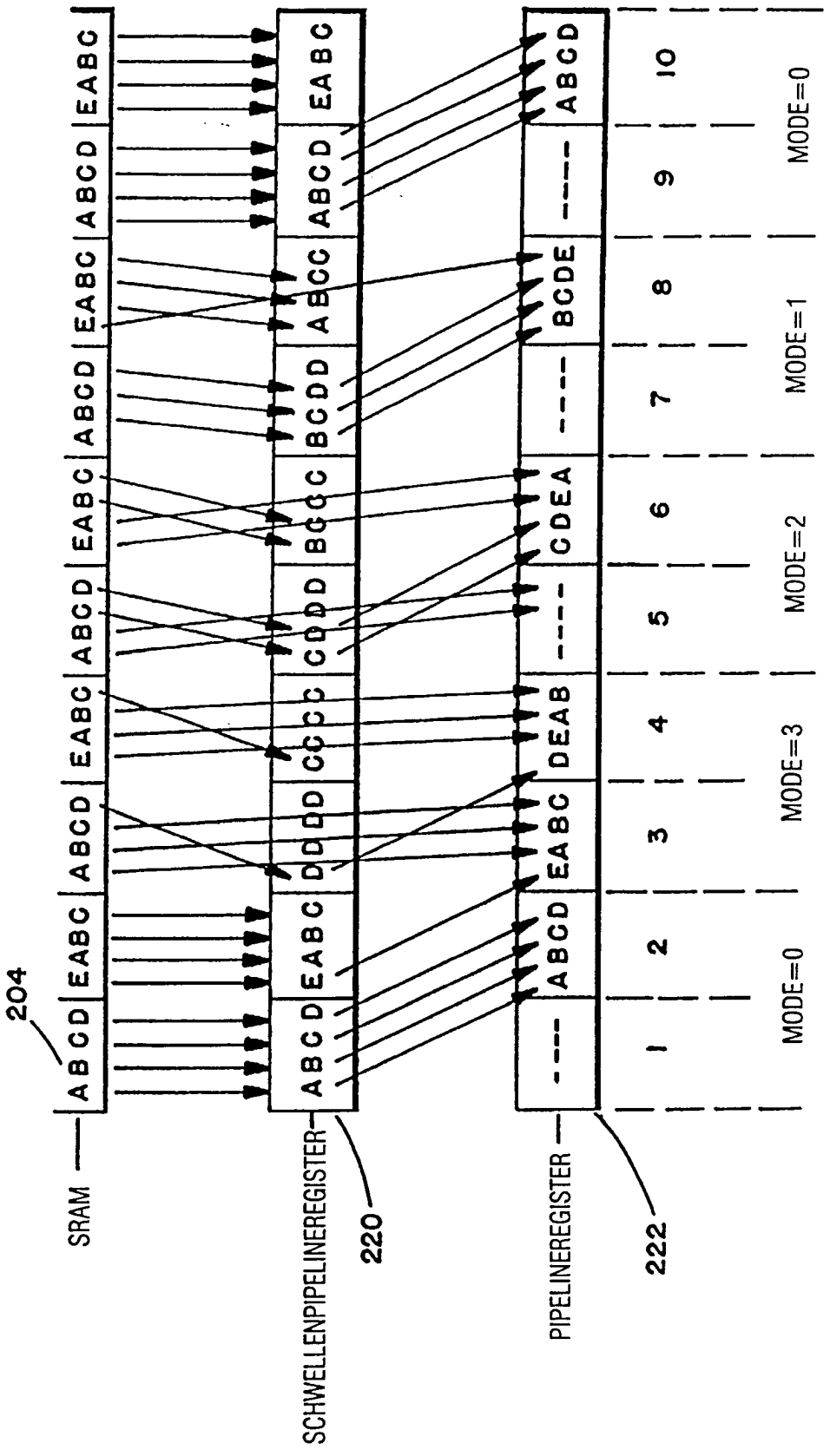
FIGUR 13D



FIGUR 12


RIIA
VON SRAM
32
202
RIII
TAKT
32
MUX
WÄHLE AUS
210
QUELLENPIXEL
VON SBAM...
200
8
32
BILD-
QUELLE
TAKT
8
8
8
8
212
4
SKALIERE SCHALTLOGIK
214
8
8
8
8
SCHWELLENMATRIX
VON SRAM
SKALIERTE
BILDQUELLE
TAKT
208
32
216
SCHWELLE
RICHTE SCHALTER AUS
MODE
8
8
8
8
8
8
8
8
TAKT
SCHW.
PIPEL.
REG.
220
8
8
8
8
218
SCHWELLE
RICHTE SCHALTER AUS
MODE
8
8
8
8
TAKT
SCHW.
REG.
222
206
KOMPARATOREN (4)
1
1
1
1
230
TAKT
32-BIT-ZWEIPEGEL-REG.
AN SRAM

FIGUR 14


SRAM
204
SCHWELLENPIPELINEREGISTER
220
PIPELINEREGISTER
222
MODE=0
MODE=3
MODE=2
MODE=1
MODE=0
1
2
3
4
5
6
7
8
9
10