



(12) **EUROPÄISCHE PATENTANMELDUNG**

(43) Veröffentlichungstag:
03.08.2005 Patentblatt 2005/31

(51) Int Cl.7: **G09G 3/20**

(21) Anmeldenummer: **05001684.9**

(22) Anmeldetag: **27.01.2005**

(84) Benannte Vertragsstaaten:
**AT BE BG CH CY CZ DE DK EE ES FI FR GB GR
HU IE IS IT LI LT LU MC NL PL PT RO SE SI SK TR**
Benannte Erstreckungsstaaten:
AL BA HR LV MK YU

(72) Erfinder:
• **Feuchtgruber, Norbert**
86911 Diessen a. Ammersee (DE)
• **Warmuth, Stefan Johann**
8293 Mittelstetten (DE)

(30) Priorität: **27.01.2004 DE 102004004072**
27.01.2004 DE 202004001183 U

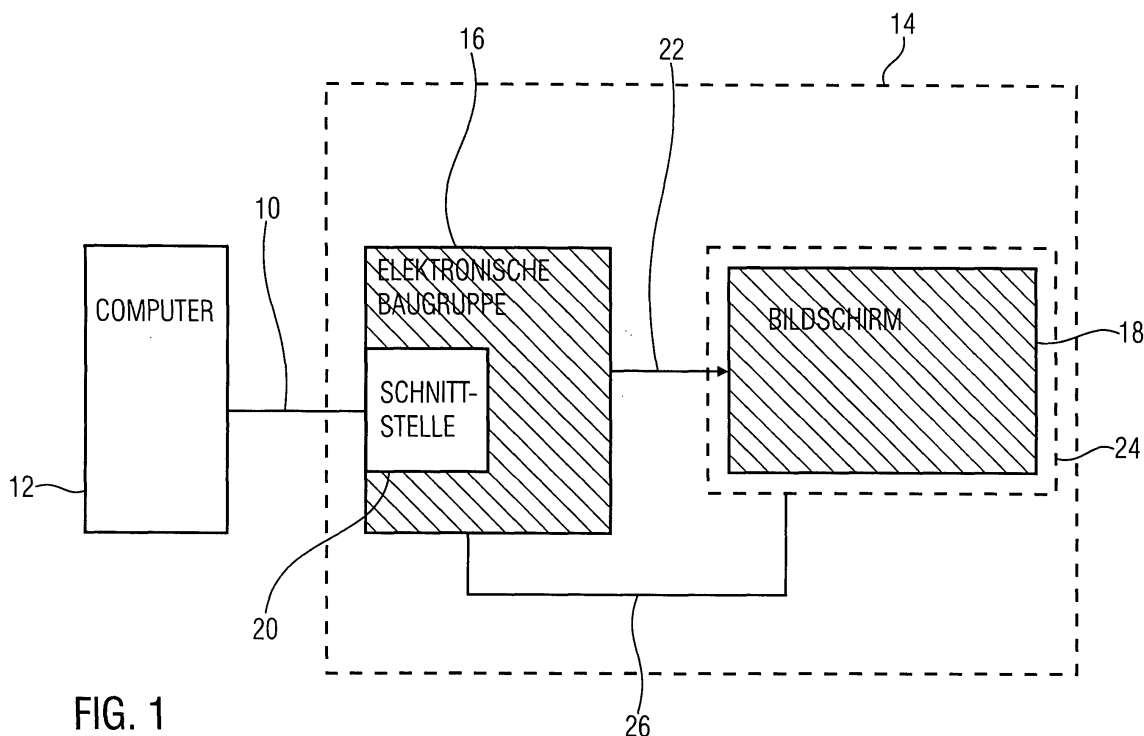
(74) Vertreter: **Zimmermann, Tankred Klaus et al**
Schoppe, Zimmermann, Stöckeler & Zinkler
Patentanwälte
Postfach 246
82043 Pullach bei München (DE)

(71) Anmelder: **Data Display AG**
82110 Germering (DE)

(54) **System zum Betreiben von Anzeigegeräten für Bilddaten über vorbestimmte Datenbuskonfigurationen**

(57) Bei einem Verfahren und einer Vorrichtung zum Betreiben eines Anzeigegeräts, dessen Anzeige durch aufeinanderfolgend empfangene Rahmen von Bilddaten, die ein oder mehrere Pixel definieren, erzeugt und kontinuierlich erneuert wird, wobei, werden die Bilddaten für einen Rahmen durch einen Host-Computer über einen seriellen Bus bereitgestellt. Die aus den über den

seriellen Bus empfangenen Daten werden an einer elektronischen Baugruppe, die zwischen den Host-Computer und das Anzeigegerät geschaltet ist, extrahiert. Die extrahierten Bilddaten werden in einer Bildspeicher der elektronischen Baugruppe gespeichert und unter Steuerung der elektronischen Baugruppe angezeigt. Diese Schritte werden für jeden der Rahmen wiederholt.



Beschreibung

[0001] Die vorliegende Erfindung bezieht sich auf ein System zum Betreiben von Anzeigegeräten für Bilddaten über vorbestimmte Datenbuskonfigurationen. Insbesondere bezieht sich die vorliegende Erfindung auf ein Verfahren und eine Vorrichtung zum Bereitstellen von Bilddaten an ein Anzeigegerät, und auf ein Verfahren und eine Vorrichtung zum Bereitstellen von Bilddaten für eine Übertragung an ein Anzeigegerät gemäß einer vorbestimmten Datenbuskonfiguration.

[0002] Übliche Signalverarbeitungssysteme, wie z.B. PCs (Personalcomputer), PDAs (Persönliche Digitale Assistenten), Computer, Fertigungssteuerungen und medizinische Geräte, bewerkstelligen die Ansteuerung von Anzeigegeräten für Bilddaten mittels eines Graphikprozessors, der wiederum über ein Betriebssystem und den dazu passenden Treiber angesteuert wird. Der Anschluss eines TFT-LCD-Bildschirms (TFT = Thin Film Transistor = Dünnschichttransistor, LCD = Liquid Crystal Display = Flüssigkristallanzeige) erfolgt über einen Graphikprozessor. Der Graphikprozessor wiederum wird über die in dem Signalverarbeitungssystem üblichen internen Busse, wie z. B. den PCI-Bus, angeschlossen. Meist befinden sich die Graphikprozessoren auf Graphikkarten. Es gibt aber auch Signalverarbeitungssysteme, in denen der Graphikprozessor anderweitig integriert ist. Gerade TFT-LCD-Bildschirme stellen bezüglich der Anschlüsse besondere Anforderungen an die Graphikkarten bzw. Graphikprozessoren. Um TFT-LCD-Bildschirme an Signalverarbeitungssystemen zu betreiben, ergibt sich für den Benutzer die Notwendigkeit, die Signalverarbeitungssysteme zu öffnen und, falls möglich, mit geeigneten Graphikprozessoren auszustatten. Es gibt zwar technische Möglichkeiten, die Busse nach außen zu führen, das technische Prinzip bleibt aber dasselbe und somit der technische Aufwand relativ hoch.

[0003] Für den Betrieb von Peripheriegeräten, wie z. B. Drucker, Modems, etc., haben Signalverarbeitungssysteme einfach zu handhabende Busverbindungen, wie z. B. USB-, RS232-, Parallelport-, Ethernet- und V.24-Busverbindungen.

[0004] Im Stand der Technik sind Ansätze bekannt, um TFT-LCD-Bildschirme über die RS232-Schnittstelle oder die Ethernet-Schnittstelle eines Computers anzusteuern. Hier müssen aber "intelligente" Steuerungsschaltungen zwischen dem Computer und dem TFT-LCD-Bildschirm geschaltet sein oder in dem TFT-LCD-Bildschirm enthalten sein, da über die genannten Schnittstellen keine Bilddaten in Form von Pixeln erhalten werden, sondern so genannte Bildbeschreibungsdaten in Form ausgewählter Steuerbefehle. Die Steuerungsschaltung muss diese Steuerbefehle decodieren und ferner geeignete Schaltungen aufweisen, um basierend auf diesen Steuerbefehlen die erforderlichen Bilddaten in Form von Pixeln zur Anzeige auf dem TFT-LCD-Bildschirm zu erzeugen. Die Steuer-

schaltung umfasst z. B. einen Zeichengenerator, der aufgrund der empfangenen Steuerbefehle ein anzuzeigendes Zeichen erzeugt und auf dem TFT-LCD-Bildschirm anzeigt. Bei diesen Systemen werden über den RS232-Bus lediglich die Steuersignale übertragen, die dann unter Verwendung einer vordefinierten Graphikfunktionalität in Bilder umgesetzt werden (Kreis, Linien, Rechtecke, etc.). Die Graphikfunktionalität ist daher stark eingeschränkt. Für die Übertragung und Darstellung echter Bilder (z.B. Fotos), die vom Computer gesendet werden, sind diese Systeme nicht gut geeignet.

[0005] Der gerade beschriebene Ansatz ist nachteilhaft, da hier von einem Host-Rechner nur Steuersignale übertragen werden, die von der Anzeigensteuerung (Display-Controller) erst in Bilddaten umgewandelt werden, so dass eine bestimmte Rechenleistung vorausgesetzt werden muss, die beispielsweise von kostengünstigen MCUs (Micro Controller Units) nur schwer bereitgestellt werden kann.

[0006] Eine einfache Möglichkeit, einen TFT-LCD-Bildschirm über die oben genannten Schnittstellen zu betreiben, so dass Bilddaten an denselben übertragen werden, existieren nicht. Sollen TFT-LCD-Bildschirme mit dem genannten Computer über eine der genannten Schnittstellen betrieben werden, so muss dieser Computer über die betreffende Schnittstelle mit einem zweiten Computer verbunden werden, wobei der TFT-LCD-Bildschirm dann aber auf herkömmliche Weise (Graphikkarte) mit diesem zweiten Computer verbunden ist. Allein der damit verbundene Aufwand spricht gegen eine solche Lösung.

[0007] Der vorliegenden Erfindung liegt daher die Aufgabe zugrunde, ein vereinfachtes Verfahren und eine vereinfachte Vorrichtung zum Betreiben von Anzeigegeräten für Bilddaten über vorbestimmte Datenbuskonfigurationen zu schaffen.

[0008] Diese Aufgabe wird durch Verfahren gemäß Anspruch 1 oder 8, bzw. eine Vorrichtung gemäß Anspruch 15 oder 21 gelöst.

[0009] Die vorliegende Erfindung schafft ein Verfahren zum Bereitstellen von Bilddaten an ein Anzeigegerät, mit folgenden Schritten:

(a) Empfangen der Bilddaten, die ein oder mehrere Pixel definieren, in einem Übertragungsformat, das einer vorbestimmten Datenbuskonfiguration entspricht; und

(b) Extrahieren der Bilddaten aus dem Übertragungsformat, und Erzeugen eines Anzeigesignals für das Anzeigegerät basierend auf den extrahierten Bilddaten.

[0010] Die vorliegende Erfindung schafft ein Verfahren zum Bereitstellen von Bilddaten für eine Übertragung an ein Anzeigegerät gemäß einer vorbestimmten Datenbuskonfiguration, mit folgenden Schritten:

Empfangen zu übertragender Bilddaten, die ein oder mehrere Pixel definieren;

Einfügen von zu übertragenden Bilddaten in ein der Datenbuskonfiguration entsprechendes Übertragungsformat; und

Ausgeben der zu übertragenden Bilddaten in dem Übertragungsformat.

[0011] Die vorliegende Erfindung schafft eine Vorrichtung zum Bereitstellen von Bilddaten an ein Anzeigegerät, mit folgenden Merkmalen:

einem Eingang zum Empfangen der Bilddaten, die ein oder mehrere Pixel definieren, in einem Übertragungsformat, das einer vorbestimmten Datenbuskonfiguration entspricht; und

einer Signalverarbeitungseinrichtung, die ausgebildet ist, um die Bilddaten aus dem Übertragungsformat zu extrahieren, und um basierend auf den extrahierten Bilddaten ein Anzeigesignal für ein Anzeigegerät zu erzeugen.

[0012] Die vorliegende Erfindung schafft eine Vorrichtung zum Bereitstellen von Bilddaten für eine Übertragung an ein Anzeigegerät gemäß einer vorbestimmten Datenbuskonfiguration, mit folgenden Merkmalen:

einem Eingang zum Empfangen zu übertragender Bilddaten, die ein oder mehrere Pixel definieren;

einer Signalverarbeitungseinrichtung, die ausgebildet ist, um die zu übertragenden Bilddaten in ein der Datenbuskonfiguration entsprechendes Übertragungsformat einzufügen; und

einem Ausgang zum Ausgeben der zu übertragenden Bilddaten in dem Übertragungsformat.

[0013] Gemäß einem Aspekt schafft die vorliegende Erfindung ferner ein Verfahren und eine Vorrichtung zum Übertragen von Bilddaten von einer Bilddatenquelle an ein Anzeigegerät, bei dem die Bilddaten von der Bilddatenquelle zu dem Anzeigegerät übertragen werden, wobei die Bilddaten und die Anzeigesignale erfindungsgemäß bereitgestellt werden.

[0014] Gemäß einem weiteren Aspekt schafft die vorliegende Erfindung ein digitales Speichermedium, insbesondere eine Diskette mit elektronisch auslesbaren Steuersignalen, die so mit einer Verarbeitungseinrichtung zusammenwirken können, dass das erfindungsgemäße Verfahren ausgeführt wird, und ein Programmprodukt mit auf einem maschinenlesbaren Träger gespeicherten Programmcode zur Durchführung des erfindungsgemäßen Verfahrens, wenn das Programmprodukt auf einer Verarbeitungseinrichtung abläuft.

[0015] Erfindungsgemäß wird somit eine einfache Möglichkeit geschaffen, Anzeigegeräte für Bilddaten, wie z. B. TFT-LCD-Bildschirme, über einen standardisierten Peripherieverbindungsbus, wie z. B. die USB-, RS232-, Parallelport-, Ethernet- oder V.24-Busverbindung, zu betreiben. Der ansonsten technisch aufwendige Einsatz gesonderter Graphikprozessor auf Graphikarten und das damit verbundene Öffnen des Systems entfällt. Die Systeme erzeugen die Bilddaten und geben diese nicht an einen Graphikprozessor oder eine Graphikkarte weiter, sondern geben diese Bilddaten direkt über die Schnittstelle aus, so dass dieselben bei Empfang an dem Anzeigegerät ohne weitere Umwandlung angezeigt werden können.

[0016] Gemäß einem bevorzugten Ausführungsbeispiel umfasst das Anzeigegerät einen Touchscreen, der über dieselbe Schnittstelle betrieben wird, über die die Bilddaten empfangen werden.

[0017] Ein Vorteil der vorliegenden Erfindung besteht darin, dass die vorgesehene Verarbeitungseinrichtung (MCU) nur wenig leisten muss, da nur der Datentransport überwacht werden muss und die Pixeldaten 1:1 an den Bildschirm weitergereicht werden. Das Bild wird in dem Host-Rechner erzeugt und als fertiges Bild (image) an die Anzeigensteuerung übergeben. Ein weiteres "Interpretieren" der empfangenen Daten ist nicht erforderlich.

[0018] Bevorzugte Weiterbildungen der vorliegenden Erfindung sind in den Unteransprüchen definiert.

[0019] Bevorzugte Ausführungsbeispiele der vorliegenden Erfindung werden nachfolgend Bezug nehmend auf die beiliegenden Zeichnungen näher erläutert. Es zeigen:

Fig. 1 ein Blockdiagramm des erfindungsgemäßen Systems zum Betreiben eines Anzeigegeräts über eine Schnittstelle einer vorbestimmten Datenbuskonfiguration gemäß einem bevorzugten Ausführungsbeispiel;

Fig. 2 ein Blockdiagramm der in Fig. 1 gezeigten elektronischen Baugruppe mit angeschlossenem Bildschirm;

Fig. 3 ein detailliertes Blockdiagramm der in Fig. 2 gezeigten elektronischen Baugruppe gemäß einem bevorzugten Ausführungsbeispiel;

Fig. 4 die Schritte in der MCU aus Fig. 3 bei der Datenübertragung;

Fig. 5 die Schritte in der PLD aus Fig. 3 zur Erzeugung eines Adresssignals aus den Zählwerten eines horizontalen Zählers und eines vertikalen Zählers;

Fig. 6 die Schritte in der PLD aus Fig. 3 zur Ansteuerung des Anzeigegeräts;

- Fig. 7 die Schritte in der PLD aus Fig. 3 zur vollständigen Übertragung eines Bildes in den Bildspeicher;
- Fig. 8 die Schritte in der PLD aus Fig. 3 zur Feststellung, ob ein Bildwechsel auf dem Anzeigegerät stattfindet;
- Fig. 9 ein Blockdiagramm des in Fig. 1 gezeigten Computers mit angeschlossenem Anzeigesystem zur Verdeutlichung des Informationsaustausches zwischen einer Anwendung, die die Bilddaten erzeugt, einer Ansteuereinrichtung für das Anzeigegerät, einer Einrichtung, die den Datentransfer ausführt, und einer Einrichtung zur Ansteuerung des Anzeigegeräts;
- Fig. 10 einen Kommunikationsprozess zwischen der Anwendung aus Fig. 9 und einem Treiber; und
- Fig. 11 einen Ablauf der Prozedur einer Datenübergabe an ein USB-System.

[0020] Anhand der Fig. 1 wird nun ein Blockdiagramm des erfindungsgemäßen Systems zum Betreiben eines Anzeigegeräts über eine Schnittstelle einer vorbestimmten Datenbuskonfiguration gemäß einem bevorzugten Ausführungsbeispiel näher beschrieben. In der nachfolgenden Beschreibung der bevorzugten Ausführungsbeispiele werden gleiche oder gleich wirkende Elemente mit den gleichen Bezugszeichen versehen.

[0021] Das in Fig. 1 gezeigte System umfasst einen Datenbus 10, der einen Computer 12, z.B. einen PC, einen PDA, eine Fertigstellungssteuerung, ein medizinisches Gerät oder ein anderes Computersystem, mit einer Anzeigeeinrichtung 14 verbindet, die eine elektronische Baugruppe 16 und ein Anzeigegerät bzw. einem Bildschirm 18, z. B. einen TFT-LCD-Bildschirm, umfasst. Die elektronische Baugruppe 16 umfasst ferner eine Schnittstelle 20 und erzeugt aus den an der Schnittstelle 20 empfangenen Signalen/Daten ein Anzeigesignal für den Bildschirm 18. Das Anzeigesignal wird über eine Verbindung 22 von der elektronischen Baugruppe 16 an den Bildschirm 18 übertragen.

[0022] Optional kann dem Bildschirm 18 ein Touchscreen 24 zugeordnet sein, der über eine Verbindung 26 mit der elektronischen Baugruppe 16 verbunden ist, um Informationen betreffend einen berührten Punkt auf dem Bildschirm 18 an die elektronische Baugruppe 16 weiterzuleiten und von dort über die Schnittstelle 20 an den Computer 12, um dort die geeigneten Schritte einzuleiten.

[0023] Die auf dem Bildschirm 18 anzuzeigenden Bilddaten werden von dem Computer 12 in Form von Pixeln, vorzugsweise in Form von Bitmaps, bereitgestellt, anschließend in ein Format entsprechend der

Schnittstelle 20, die einem vorbestimmten Datenbuskonfiguration entspricht, umsetzt und an einer in Fig. 1 nicht gezeigten Schnittstelle des Computers 12 für eine Übertragung an das Anzeigesystem 14 bereitgestellt. Diese Daten werden über den Datenbus 10, der die vorbestimmte Datenbuskonfiguration unterstützt, an die Schnittstelle 20 übertragen. Die an der Schnittstelle 20 empfangenen Daten werden in der elektronischen Baugruppe 16 zum Erzeugen des Anzeigesignals verarbeitet. Die elektronische Baugruppe 16 extrahiert die Bilddaten aus dem entsprechenden Datenbusformat und erzeugt das Anzeigesignal für den Bildschirm 18, das über die Verbindung 22 an den Bildschirm gesendet wird.

[0024] Das Datenbusformat zur Übertragung der Bilddaten und die Konfiguration der Schnittstellen des Computers 12 und des Anzeigesystems 14 entsprechen gemäß dem bevorzugten Ausführungsbeispiel dem USB-, dem RS232- oder dem Parallelanschluss-Datenbusformat. Andere bekannte Schnittstellen- bzw. Datenbuskonfigurationen, wie z. B. der Ethernetstandard, der V. 24-Standard, etc., können ebenfalls verwendet werden.

[0025] Eine typische Anwendung der vorliegenden Erfindung ist der Einsatz von Automaten, z.B. Verkaufsautomaten für Kleinwaren. Typischerweise werden moderne Verkaufsautomaten Bildschirme oder Anzeigen zur Benutzerführung aufweisen. Die vorliegende Erfindung ist insbesondere für kleine Anzeigen mit einer Auflösung von z.B. 320x240 Bildpunkten, auch QVGA genannt, geeignet. Erfindungsgemäß kann ein Bild bei der Nutzung des USB 2.0 Busses zur Übertragung etwa 50 mal in der Sekunde erneuert werden, was für die oben genannte Anwendung ausreichend ist. Bei anderen Anwendungen können auch mehrere Anzeigen bzw. Bildschirme parallel zu herkömmlichen Bildschirmen verwendet werden.

[0026] Nachfolgend wird die vorliegende Erfindung anhand des Beispiels der USB-Schnittstelle detailliert erläutert.

[0027] Über die USB-Schnittstelle 20 wird von Seiten des Host-Systems 12 (z.B. PC) eine Grafik-Bitmap übertragen. Bei der hier gezeigten Lösung wird ein Format von 320 x 240 Bildpunkten (Pixeln) mit 16 Bit Farbtiefe verwendet. Selbstverständlich ist die vorliegende Erfindung nicht hierauf beschränkt und es können auch andere Formate verwendet werden. Die Grafik-Bitmap ist somit ein binäres Datenformat von 320x240x16 Bit.

[0028] Ein bevorzugtes Ausführungsbeispiel der elektronischen Baugruppe 16 und der zugehörigen Software wird nachfolgend anhand der Fig. 2 bis 8 erläutert. Ein bevorzugtes Ausführungsbeispiel der erforderlichen Ausgestaltung des Computers 12 wird anhand der Fig. 9 bis 11 erläutert.

[0029] Fig. 2 zeigt ein Blockdiagramm der in Fig. 1 gezeigten elektronischen Baugruppe 16 mit angeschlossenem Bildschirm 18. Die elektronische Baugruppe 16 umfasst eine Steuerung 28 (MCU), die die USB-Schnittstelle 20 umfasst. Ferner umfasst die elektronische

Baugruppe 16 eine Zeitgebungseinrichtung 30 (PLD = Programmable Logic Device = programmierbares Logikeinrichtung oder -element), einen Bildspeicher 32, vorzugsweise ein RAM-Speicher (RAM = Random Access Memory = Speicher mit wahlfreiem Zugriff), und eine optionale Touchscreensteuerung 34, die vorgesehen ist, wenn dem Bildschirm 18, wie in dem gezeigten Beispiel, der Touchscreen 24 zugeordnet ist. Wie in Fig. 2 zu erkennen ist, sind die einzelnen Komponenten der elektronischen Baugruppe 16 über eine Mehrzahl von Leitungen zum Datenaustausch verbunden. Genauer gesagt tauschen die MCU 28 und die PLD 30 die Signale WR_START, WR_FULL und WR_EN über je eine Leitung aus, und über eine andere Leitung das Taktsignal CLK. Die MCU 28 ist ferner über einen Datenbus, der die Pixeldaten trägt, mit dem Bildspeicher 32 und dem Bildschirm 18 verbunden. Die PLD 30 stellt dem Bildspeicher 32 die Signale OE und addr bereit, und dem Bildschirm 18 die Signale HSYNC und VSYNC sowie das Taktsignal CLK. Die Touchscreensteuerung 34 ist über die Verbindung 26 mit dem Touchscreen 24 verbunden und tauscht über die Verbindung 36 die zum Betrieb des Touchscreen 24 erforderlichen Informationen mit der MCU 28 aus.

[0030] Wie zu erkennen ist, werden die Signale HSYNC, VSYNC, CLK und die Pixeldaten über die gemeinsame Verbindung 22 übertragen.

[0031] Die elektronische Baugruppe 16 besteht im wesentlichen aus zwei Teilen, der MCU 28 (Micro-Controller-Unit) und der Zeitgebungseinrichtung 30 (Timing-Engine), die hier vorzugsweise durch die PLD implementiert ist.

[0032] Die MCU 28 ist hauptsächlich für den Datentransport von dem Computer 12 (Host) zu der elektronischen Baugruppe 16 zuständig. Die PDL 30 generiert hauptsächlich die Synchronsignale HSYNC (horizontale Synchronisation), VSYNC (vertikale Synchronisation) für den Bildschirm 18 und die Adress-Signale addr für den Bildspeicher 32.

[0033] Die MCU 28 ist mit der USB-Schnittstelle 20 ausgestattet, über die die anzuzeigenden Grafik-Bitmap-Daten von dem Computer 12 an die elektronische Baugruppe 16 übertragen werden. Die Daten werden dann in den Bildspeicher 32 weitergeleitet.

[0034] Da der Bildschirm 18, der Bildspeicher 32 und die MCU 28 einen gemeinsamen Datenbus (Datenbus "Pixel-Daten" in Fig. 2) haben, muss der Zugriff auf diese Element gesteuert werden. Der Ausgang des Bildspeichers 32 wird hierzu im Zeitraum der fallenden Flanke des Taktsignals CLK und der Ausgang der MCU 28 im Zeitraum der steigenden Flanke des Taktsignals CLK freigeschaltet. Die technologisch bedingten Setup- und Hold-Zeiten werden dabei berücksichtigt.

[0035] Die optionale Touchscreen-Steuerung 34 ist handelsüblich und wird über den seriellen Bus 36 an der MCU 28 betrieben. An diesem Bus hängt, wie später noch näher erläutert wird, auch ein Programmspeicher für die MCU-Firmware. Die Touchscreen-Daten werden

durch einen USB-Interrupt-Transfer an Computer 12 übertragen.

[0036] Die Helligkeitsregelung des Bildschirms 18 wird durch das PWM-Verfahren (PWM = Pulsweitenmodulation) bewerkstelligt. Hierfür sind zwei, in der MCU 28 integrierte "Auto Reload Timer" (Zeitgeber) im Einsatz. Über die PLD 30 wird abwechselnd jeweils ein Zeitgeber für die "High-Phase" (hoher logischer Regel) und der andere für die "Low-Phase" (niedriger logischer Regel) des PWM-Signals aktiviert. Mit den Signalen, die die Zeitgeber beim Ablauf eines Intervalls generieren, toggelt die PLD 30 das PWM-Signal, wie dies später noch näher erläutert wird.

[0037] Anhand der Fig. 3 wird nun ein detailliertes Blockdiagramm eines bevorzugten Ausführungsbeispiels der in Fig. 2 gezeigten elektronischen Baugruppe und hier insbesondere der MCU 28, der PLD 30 und des Bildspeichers 32 näher beschrieben.

[0038] Die MCU 28 umfasst neben der USB-Schnittstelle einen Datentransferabschnitt 38, der einen FI-FO-Speicher 40 (FIFO = First-In-First-Out) und eine Mehrzweck-Schnittstelle 42 (GPIF = General Purpose Interface) aufweist. Wie zu erkennen ist, gibt die GPIF 42 die Signale WR_START und WR_EN an die PLD 30 aus, empfängt von derselben das Signal WR_FULL und tauscht mit derselben das Signal WR_RST aus. Der FI-FO-Speicher 40 ist über den 16 Bit Datenbus 44 mit dem Bildspeicher 32 und dem Bildschirm 18 zur Übertragung der Pixeldaten verbunden. Der FIFO-Speicher 40 und die GPIF 42 sind über eine Verbindung 46 verbunden. Der Datentransferabschnitt 38 der MCU 28 empfängt ferner von der PLD 30 einen Schnittstellentakt IFCLK.

[0039] Die MCU 28 umfasst einen weiteren Abschnitt 48, in dem die nicht gezeigte CPU der MCU 28 angeordnet ist. In dem weiteren Abschnitt 48 ist eine Zeitgeber/PWM-Schaltung 50 angeordnet, die an die PLD 30 Signale bereitstellt und von derselben empfängt. Ferner wird der MCU-Takt MCUCLK an die PLD 28 bereitgestellt, ebenso wie das Signal CHANGE_EN. Ferner wird das Signal BKL_ON erzeugt und ausgegeben. Das Signal BKL_ON schaltet die Hintergrundbeleuchtung des TFT-LCD-Bildschirms an oder aus, was über einen USB-VENDOR-CALL gesteuert wird. Von der Anwendungs/Applikations-Seite aus wird dieser USB-VENDOR-CALL über die treiberseitige Funktion IOCTL() ausgelöst. Die MCU 28 ist über den seriellen Bus 36 mit der Touchscreensteuerung 34 und einem Speicher 52 (vorzugsweise ein EEPROM), in dem die MCU-Firmware abgelegt ist, verbunden.

[0040] Die PLD 30 umfasst einen Taktteiler 54, der von der MCU 28 den MCU-Takt MCUCLK empfängt und an den Datentransferabschnitt 38 der MCU 28 den Schnittstellentakt IFCLK ausgibt. Ferner gibt der Taktteiler 54 das Taktsignal CLK aus.

[0041] Eine PWM-Schaltung 56 ist vorgesehen, die das Taktsignal CLK empfängt und an die Zeitgeber/PWM-Schaltung 50 der MCU 28 Signale bereitstellt und von derselben empfängt. Ferner stellt die PWM-Schal-

tung 56 ein Steuersignal auf der Leitung 58 bereit.

[0042] Ein erster Zähler 60 (wr_count) empfängt von der MCU 28 die Signale WR_START und WR_EN. Er gibt an die MCU 28 das Signal WR_FULL aus. Ferner tauscht der erste Zähler 60 das Signal WR_RST mit der MCU 28 aus. Von dem Takteiler 54 empfängt der erste Zähler 60 das Taktsignal CLK. Über einen ersten Bus 62 gibt der erste Zähler 60 seinen Zählwert an einen Adressenmultiplexer 64 (addrmux), der ferner einen Zählwert eines zweiten Zählers 66 über einen zweiten Bus 68 empfängt. Über einen 17 Bit Bus gibt der Adressenmultiplexer 64 das Adresssignal ADDR[0..16] an den Bildspeicher 32 aus. Der Adressenmultiplexer 64 empfängt ferner das Taktsignal CLK von dem Takteiler 54.

[0043] Der zweite Zähler 66 empfängt das Taktsignal CLK von dem Takteiler 54 und gibt neben dem Zählwert die Signale DE, HSYNC und VSYNC an den Bildschirm 18 aus. Das Signal VSYNC wird ferner als Signal new_frame (neuer Rahmen) an eine Adressraum-Auswahlschaltung 70 angelegt, die ferner das Signal WR_FULL empfängt und das Signal WR_RST sendet. Die Adressraum-Auswahlschaltung 70 gibt ein Auswahlssignal ADDR_17 an den Bildspeicher 32 aus und empfängt ferner das Taktsignal CLK von dem Takteiler 54.

[0044] Die PLD 30 umfasst ferner die Schaltung 72 (OE) zum Erzeugen eines Aktivierungssignals OE für den Bildspeicher 32, abhängig von dem empfangenen Taktsignal CLK.

[0045] Die elektronische Baugruppe 16 umfasst ferner die PWM-Schaltung 74, die von der PWM-Schaltung 56 der PDL 30 über die Leitung 58 angesteuert wird, um ein Helligkeitssignal zu erzeugen.

[0046] Der Bildspeicher 32 hat zwei Speicherabschnitte 32a und 32b, die abwechselnd Pixeldaten von der MCU 28 empfangen und an den Bildschirm 18 ausgeben.

[0047] Anhand der Fig. 3 und der Fig. 4 wird nachfolgend die Funktionalität der MCU 28 näher erläutert. Die MCU 28 regelt den Transfer der Grafik-Bitmap-Daten von dem Host 12 in den Bildspeicher 32. Der eigentliche Datentransfer wird mit der in der MCU 28 integrierten GPIF-Einheit 42 weitgehend autark von der integrierten CPU bewerkstelligt. Die GPIF-Einheit 42 ist eine Zustandsmaschine (Statemachine), die einen DMA Datentransfer vom dem ebenfalls in der MCU 28 integrierten FIFO-Speicher 40 in den Bildspeicher 32 steuert. Die GPIF 42 verfügt dafür über einen Zähler der mit der Zahl der zu übertragenden Daten gefüllt wird, also die Anzahl der Pixel einer kompletten Bitmap. Der Zähler wird durch einen USB-Vendor-Aufruf (Call) "WR_START" gesetzt. Zudem wird die GPIF 42 initialisiert. Der Aufruf WR_START setzt auch das Signal WR_START, um den Adresszähler 60 in der PLD 30 auf die Speicher-Start-Position/Adresse eines Bildes zu setzen.

[0048] Die von dem USB-Bus 10 ankommenden Bitmap-Daten werden in dem FIFO-Speicher 40 abgelegt.

Dies bewerkstelligt eine in der MCU 28 integrierte SIE-Einheit (nicht gezeigt; SIE = USB Serial Interface Engine) automatisch. Die GPIF 42 erkennt, dass Daten im FIFO-Speicher 40 zur Verfügung stehen und überträgt diese selbstständig an den Bildspeicher 32.

[0049] Die Adressierung des Speichers erfolgt durch die PDL 30. Legt die GPIF 42 gültige Pixel-Daten auf den Datenbus 44, so wird dies durch das Signal WR_EN angezeigt. Die Daten werden mit der steigenden Flanke des Taktsignals CLK im Bildspeicher 32 übernommen. Für jedes übertragene Pixel wird der Zähler der GPIF 42 dekrementiert. Ist ein komplettes Bild übertragen, wird in der MCU 28 eine Unterbrechung (Interrupt) durch das Signal WR_FULL aus der PLD 30 ausgelöst.

Die dadurch in der MCU 28 gestartete ISR (Interrupt Service Routine) überprüft den Zählerstand der GPIF 42. Ist dieser "NULL", zeigt dies einen erfolgreichen Transfer an. Damit ist ein komplettes neues Bild im Bildspeicher 32 abgelegt. Dies wird mit dem Signal CHANGE_EN durch die MCU 28 angezeigt. Dadurch wechselt die Adressraum-Auswahlschaltung 70 der PLD 30 den Adressraum zum nächstmöglichen Zeitpunkt, nämlich zum Zeitpunkt der Vertikalaustastlücke, und das neue Bild wird angezeigt. Wechselt die PLD 30 den Adressraum, so setzt dieselbe den ersten Zähler 60 und damit das Signal WR_FULL zurück. Dies wird mit dem Signal WR_RST angezeigt und löst wiederum eine Unterbrechung (Interrupt) in der MCU 28 aus. Die dadurch gestartete ISR setzt das Signal CHANGE_EN wieder zurück, setzt den Zähler der GPIF 42 wieder auf die Pixelzahl eines kompletten Bildes und startet die GPIF 42 für einen erneuten Datentransfer. Zudem wird dem Host 12 der erfolgreiche Datentransfer über einen USB-Interrupt-Transfer mitgeteilt.

[0050] Wird das Signal WR_FULL angezeigt und erkennt die damit ausgelöste ISR, dass der Zähler der GPIF 42 nicht "NULL" ist, liegt ein Fehlverhalten vor. Dies wird durch einen USB-Interrupt-Transfer dem Host 12 mitgeteilt. Der Host muss dann über den USB-Vendor-Call "WR_START" das System zurücksetzen und den Datentransfer neu starten.

[0051] Ein weiteres Fehlverhalten liegt vor wenn der Zähler der GPIF 42 auf "NULL" läuft, aber das Signal WR_FULL von der PLD 30 nicht gesetzt wird. Läuft die GPIF dann auf "NULL" wird ein Interrupt in der MCU 28 ausgelöst. Dieser ist niedriger priorisiert als der Interrupt durch das Signal WR_FULL. Deshalb wird im Normalfall der Interrupt "GPIF-Zähler NULL" durch die "ISR WR_FULL" ausmaskiert und löst damit nicht die ISR "GPIF-Zähler NULL" aus. Bei einem Fehlverhalten wird die ISR "GPIF-Zähler NULL" aber aktiviert, da kein Signal WR_FULL anliegt, um das Fehlverhalten anzuzeigen. Dies wird durch einen USB-Interrupt-Transfer dem Host 12 mitgeteilt. Der Host 12 muss dann über den USB-Vendor-Call "WR_START" das System zurücksetzen und den Datentransfer neu starten.

[0052] Anhand des in Fig. 4 gezeigten Flussdiagramms wird die gerade beschriebene Funktionalität

nochmals verdeutlicht. Der Ablauf startet bei 100 durch einen USB-Vendor-Aufruf (Call) und geht dann zu Block 102 (WR_START). Alternativ kehrt der Ablauf aus einer Fehlerbehandlungsroutine zu dem Block 102 zurück, wie dies bei Block 104 angedeutet ist. Im Block 102 wird das Signal WR_START gesetzt. Damit wird der Adresszähler 60 in der PLD 30 zurückgesetzt und der Zähler der GPIF 42 wird auf die Anzahl der zu übertragenden Pixel (hier 320x240 Pixel) gesetzt.

[0053] Der Ablauf geht dann weiter zum Block 106 (TRANSFER), der die Übertragung der Bilddaten bewirkt. Die GPIF 42 überträgt die Bilddaten selbständig und bei jedem übertragenen Pixel (hier 16 Bit) wird der Zähler der GPIF 42 dekrementiert.

[0054] Während der Datenübertragung werden das Signal WR_FULL und der Zählwert "count" des GPIF-Zählers überwacht. Im Block 108 wird die Interrupt Service Routine "ISR GPIF DONE" durch die GPIF 42 aufgerufen. Die GPIF überträgt die Bilddaten bis der Datenzähler "count" auf Null läuft. Dies löst den Interrupt 4 aus. Im Block 110 wird anhand des Signals WR_FULL überprüft, ob der Bildspeicher vollständig gefüllt ist, nachdem alle Pixel übertragen wurden. Ist dies nicht der Fall, so geht der Ablauf zum Fehlerblock 112, der einen Fehler bei der Übertragung anzeigt, die GPIF 42 und die PLD 30 zurücksetzt und dem Host 12 den Fehler über den Interrupt-Transfer anzeigt. Wird angezeigt, dass der Bildspeicher 32 vollständig gefüllt ist, nachdem alle Pixel übertragen wurden, so geht der Ablauf zum Block 114 (CHANGE_EN).

[0055] Läuft der Adresszähler 60 der PLD 30 über, so wird dies durch das Signal WR_FULL angezeigt und bei Block 116 wird der Interrupt 1 ausgelöst. Ferner muss der Interrupt 4 maskiert und zurückgesetzt werden, da dieser im Normalfall unmittelbar danach ausgelöst wird. Da das Signal WR_FULL einen Zustand hat, der anzeigt, dass der Bildspeicher gefüllt ist, wird anschließend beim Block 118 überprüft, ob die erforderliche Anzahl von Pixeln übertragen wurde. Ist dies nicht der Fall, geht der Ablauf zum Block 120, der einen Fehler bei der Übertragung anzeigt, die GPIF 42 und die PLD 30 zurücksetzt und dem Host 12 den Fehler über den Interrupt-Transfer anzeigt. Wird angezeigt, dass alle Pixel übertragen wurden ("count" = NULL), so geht der Ablauf zum Block 114 (CHANGE_EN).

[0056] Im Block 114 (CHANGE_EN) wird das Signal CHANGE_EN aktiviert, um der PLD 30 zu erlauben, den Speicherbereich im Bildspeicher 32 zu wechseln. Damit wird das übertragene Bild angezeigt und der Speicherbereich des vorangegangenen Bildes frei für eine erneute Übertragung.

[0057] Im Block 122 wird der Wechsel des Speicherbereichs und damit des angezeigten Bildes in der Vertikalaustastlücke vorgenommen. Der Wechsel wird mit dem Signal WR_RST angezeigt. Die PLD 32 setzt dabei den Zähler 60 selbständig zurück. Das Signal WR_RST löst einen Interrupt aus, der beim Block 124 anzeigt, dass der Transfer ordnungsgemäß war, wobei dem Host 12

beim Block 124 ferner über einen Interrupt-Transfer der erfolgreiche Datentransfer einer Bitmap angezeigt wird. Anschließend wird im Block 122 das Signal CHANGE_EN wieder zurückgesetzt.

[0058] Anschließend wird im Block 126 festgestellt, dass ein freier Speicherplatz für die Übertragung bereitsteht. Der Zähler der GPIF 42 wird wieder auf die Anzahl der Pixel eines Bildes gesetzt (hier 320x249 Pixel).

[0059] Im Block 128 wird ein Rahmenzähler inkrementiert. Der Zählerstand dieses Rahmenzählers (Frame-Counter) kann jederzeit über einen Vendor-Call seitens des Hosts 12 abgefragt werden. Damit kann der Host über die USB-Verbindung 16 feststellen, welche Bilder (Frames) übertragen und angezeigt worden sind und welche nicht.

[0060] Anschließend wird bei Block 130 eine neue Übertragung von Bilddaten mit der GPIF 42 gestartet und der Ablauf geht zurück zu Block 106 oder endet bei 132 wenn eine Ende-Transfer-Anweisung vorliegt, die durch einen USB-Vendor-Call mitgeteilt wurde.

[0061] Anhand der Fig. 3 und der Fig. 5 bis 8 wird nachfolgend die Funktionalität der PDL 30 näher erläutert. Die PLD 30 erzeugt die für den Bildschirm 18 erforderlichen Synchronsignale HSYNC und VSYNC und erzeugt die Adressen für den Bildspeicherzugriff. Das Taktsignal CLK wird aus dem Taktsignal MCUCLK der MCU 28 durch Frequenzteilung erzeugt. Außerdem steuert die PLD 30 das Signal OE (Output-Enable) für den Bildspeicher 32. Der Ausgang des Bildspeichers 32 wird so frei geschaltet, dass der Bildschirm 18 die Pixeldaten zur fallenden Flanke des Taktsignals CLK aus dem Bildspeicher 32 übernehmen kann.

[0062] Es sind die zwei Zählereinheiten 60 und 66 implementiert. Die erste Einheit 60 ist für den Datentransport von der MCU 28 zu dem Bildspeicher 32 zuständig. Die andere Einheit 66 ist für die Ansteuerung des Bildschirms 18 zuständig. Aus den Zählerwerten wird jeweils das Adress-Signal für den Bildspeicher 32 generiert. Dazu bestehen die Zählereinheiten 60 und 66 aus jeweils zwei Zählern, dem Horizontal-Zähler und dem Vertikal-Zähler. Der Horizontalzähler wird mit dem Taktsignal CLK inkrementiert. Der Wert des Zählers entspricht somit der Position eines Pixels innerhalb einer Zeile. Der Vertikalzähler wird durch den Überlauf des Horizontalzählers inkrementiert, also mit Zeilenfrequenz. Der Wert des Vertikalzählers entspricht somit der Zeilenposition. Der Horizontal-Zähler und Vertikal-Zähler zusammen ergeben zusammen eine Speicheradresse die der Pixelposition entspricht. Wie in Fig. 5 bei 140 gezeigt ist, umfasst der Vertikal-Zähler einen 8 Bit Zählwert. Der Horizontal-Zähler umfasst, wie bei 142 gezeigt ist, einen 9 Bit Zählwert. Durch geeignete Verknüpfung der Zählwerte ergibt sich das 17 Bit Adresssignal, wie es bei 144 gezeigt ist.

[0063] Durch die Aufteilung in den Vertikal-Zähler und den Horizontal-Zähler lassen sich auch die Synchronsignale generieren. Die Zählereinheit 66 zur Ansteuerung des Bildschirms generiert die Signale HSYNC und

VSYNC. Komparatoren überwachen die Zählerstände und setzen bzw. rücksetzen die Synchronsignale an den entsprechenden Positionen nach der Spezifikation des TFT-Bildschirms 18.

[0064] Fig. 6 verdeutlicht die Erzeugung der Synchronsignale. Mit jeder steigenden Flanke des Taktsignals CLK wird bei Block 150 bestimmt, ob ein Zeilenende erreicht wurde oder nicht. Wenn das Zeilenende noch nicht erreicht wurde, so wird der Horizontal-Zähler ("hcount") im Block 152 inkrementiert. Ist das Ende der Zeile erreicht, so wird der Horizontal-Zähler im Block 154 auf den Wert 1 gesetzt. Nachdem der Horizontal-Zähler im Block 154 auf den Wert 1 gesetzt wurde, wird im Block 156 überprüft, ob das Bildende erreicht worden ist oder nicht. Ist das Bildende noch nicht erreicht worden, wird der Vertikalzähler ("vcount") im Block 158 inkrementiert. Ist das Bildende bzw. Rahmen-Ende erreicht, so wird der Vertikalzähler im Block 160 auf den Wert 1 gesetzt. Anschließend wird auf der Basis der so eingestellten Zählerwerte die Synchronisierung des Bildschirms gesteuert.

[0065] Im Block 162 wird basierend auf einer Spezifizierung für die SYNCH-Signale des Bildschirms überprüft, ob der Wert des Horizontal-Zählers ("hcount") einem vorgegebenen HSYNC-Bereich entspricht. Ist dies der Fall, wird HSYNC im Block 164 auf den Wert 0 gesetzt. Ist dies nicht der Fall, wird HSYNC im Block 166 auf den Wert 1 gesetzt.

[0066] Im Block 168 wird basierend auf einer Spezifizierung für die HSYNCH/SYNCH-Phasenverschiebung des Bildschirms überprüft, ob der Wert des Horizontal-Zählers einen minimalen Phasenverschiebungswert unterschreitet oder nicht. Unterschreitet der Wert des Horizontal-Zählers den minimalen Phasenverschiebungswert, so wird keine weitere Aktion durchgeführt. Erreicht der Wert des Horizontal-Zählers den minimalen Phasenverschiebungswert oder überschreitet diesen, so wird im Block 170 basierend auf einer Spezifizierung für die SYNCH-Signale des Bildschirms überprüft, ob der Wert des Vertikal-Zählers ("vcount") einem vorgegebenen VSYNC-Bereich entspricht. Ist dies der Fall, wird VSYNC im Block 172 auf den Wert 0 gesetzt. Ist dies nicht der Fall, wird VSYNC im Block 174 auf den Wert 1 gesetzt.

[0067] Abschließend wird im Block 176 aus den Zählerwerten "vcount" und "hcount" die Speicheradresse RD_ADDR ermittelt.

[0068] Die Zählereinheit 60, die für den Transport der Bilddaten von der MCU 28 in den Bildspeicher 24 zuständig ist, zeigt durch das Signal WR_FULL die vollständige Übertragung eines Bilds in den Bildspeicher 32 an, wie dies anhand der Fig. 7 verdeutlicht wird.

[0069] Zu jeder steigenden Flanke des Taktsignals CLK wird im Block 180 basierend auf dem Signal WR_EN überprüft, ob Daten in den Speicher geschrieben werden bzw. ob die MCU 28 gültige Daten zur Verfügung stellt.

[0070] Stehen keine Daten zum Schreiben an

(WR_EN ungleich 1), so wird die Speicheradresse im Block 182 nicht verändert, da die Zählwerte "vcount" und "hcount" nicht geändert werden.

[0071] Liegen Daten zum Schreiben in den Speicher vor, wird im Block 184 überprüft, ob das Zeilenende erreicht wurde (Wert "hcount" ist größer oder gleich der Zeilenlänge) oder nicht (Wert "hcount" ist kleiner der Zeilenlänge). Wurde das Zeilenende noch nicht erreicht, wird der Wert des Horizontal-Zählers im Block 186 inkrementiert. Andernfalls wird im Block 188 überprüft, ob das Bildende erreicht wurde (der Wert des Vertikal-Zählers "vcount" ist größer oder gleich der Zeilenanzahl) oder nicht (der Wert des Vertikal-Zählers "vcount" ist kleiner der Zeilenanzahl). Ist das Bildende noch nicht erreicht, so wird der Vertikal-Zähler im Block 190 inkrementiert und der Horizontal-Zähler wird im Block 192 auf den Wert 1 gesetzt, was bedeutet, dass die Zeilennummer erhöht wird und die Pixelnummer wieder auf den Anfang der Zeile gesetzt wird.

[0072] Wurde das Bildende erreicht, so wird das Signal WR_FULL im Block 194 auf den Wert 1 gesetzt, und der Wert des Horizontal-Zählers wird im Block 196 auf eine ungenutzte Speicheradresse gesetzt. So wird angezeigt, dass das komplette Bild im Speicher enthalten ist.

[0073] Anschließend werden im Block 198 die Signale WR_START und WR_RST überprüft. Sind diese nicht aktiviert (hier: niedriger logischer Pegel), d.h. es hat keine neue Datenübertragung begonnen, so werden keine weiteren Änderungen der Zählerwerte durchgeführt und bei 182 die Speicheradresse gebildet. Andernfalls, also wenn die Signale WR_START und WR_RST anzeigen, dass eine neue Datenübertragung begonnen wird (neues Bild), wird im Block 200 der Wert des Vertikal-Zählers auf den Wert 1 gesetzt, im Block 202 wird der Wert des Horizontal-Zählers auf den Wert 1 gesetzt und im Block 204 wird das Signal WR_FULL deaktiviert, um die beiden Zähler auf den Bildanfang zu setzen und das Signal WR_FULL zurückzusetzen. Anschließend wird bei Block 182 die Speicheradresse gebildet.

[0074] Um so genannte Frame-Tears zu vermeiden wird mit dem "double Buffer"-Prinzip gearbeitet. Hierfür ist der Bildspeicher 32 in die zwei Bereiche 32a und 32b aufgeteilt, die jeweils ein komplettes Bild fassen können. Das höchstwertigste Adressbit des Bildspeichers 32 wird verwendet, um diesen in den oberen und den unteren Speicherbereich 32a, 32b aufzuteilen. Ein Bereich dient zur Versorgung des Bildschirms 18 mit den Bilddaten und der andere nimmt die neuen Bilddaten, die von der MCU 28 kommen, auf. Zwischen den Bereichen 32a, 32b wird ständig hin und her geschaltet. Zur steigenden Takt-Flanke ist der Bereich aktiv der die neuen Daten von der MCU 28 aufnimmt und zur fallenden Flanke der Bereich mit den anzuzeigenden Bilddaten. Ist ein neues Bild komplett übertragen, werden die Bereiche getauscht. Hierfür ist die Adressraum-Einheit 70 zuständig. Über das Signal WR_FULL erkennt die Adressraum-Einheit 70, dass eine neues Bild im Spei-

cher 32 liegt. Durch das Signal new_frame, das dem VSYNC-Signal entspricht, erkennt die Adressraum-Einheit 70 die Vertikalaustastlücke. Zu diesem Zeitpunkt kann ein Bildwechsel vorgenommen werden. Die MCU 28 erteilt mit dem Signal CHANGE_EN die Erlaubnis zum Bildwechsel. Werden die beiden Signale WR_FULL und new_frame angezeigt und ist das Signal CHANGE_EN gesetzt, wird der Speicherbereich getauscht und somit ein neues Bild auf dem Bildschirm ausgegeben. Der Wechsel wird durch das Signal WR_RST angezeigt, dessen Erzeugung anhand der Fig. 8 näher erläutert wird.

[0075] Mit jeder steigenden Flanke eines Taktes wird im Block 210 bestimmt, ob durch die Signale WR_FULL, new_frame und CHANGE_EN ein Wechsel der Adressräume angezeigt wird oder nicht. Soll ein Wechsel der Adressräume erfolgen, so wird im Block 212 das Rücksetzsignal WR_RST aktiviert, um den Zähler 60 zurückzusetzen und der MCU 28 den Bildwechsel anzuzeigen. Andernfalls wird das Signal WR_RST im Block 218 deaktiviert.

[0076] Entsprechend des gerade aktiven Adressbereichs wird das Adress-Signal der Zählereinheit für den Datentransport bzw. zur Ansteuerung des Bildschirms auf die Adressleitungen des Bildspeichers 32 gelegt, was durch den Adressmultiplexer 64 (addrmux) erfolgt.

[0077] Anhand der Fig. 9 bis 11 wird ein Blockdiagramm eines Ausführungsbeispiels des in Fig. 1 gezeigten Computers 12 mit angeschlossenem Anzeigesystem 14 erläutert. Bei diesem Ausführungsbeispiel ist eine Anwendung 250 vorgesehen, die auf dem Computer abläuft und die Bilddaten erzeugt. Die Anwendung gibt die Bilddaten an einen Treiber 252, der mit einem USB-Teilsystem 254 des Betriebssystems zusammenwirkt.

[0078] Auf der Host-Seite 12 wird der Transfer von Grafik-Bitmap-Daten mit dem Softwaretreiber 252 bewerkstelligt. Dieser bietet dem Anwendungs-/Applikationsprogrammierer bzw. der Anwendung/Applikation 250 die erforderlichen Schnittstellen für die Übergabe der Bitmaps und liefert Erfolgs- bzw. Fehlercodes.

[0079] Anhand der Fig. 10 werden die Schritte zum Öffnen und Schließen der Schnittstelle und der Übertragung von Daten näher erläutert. Nach dem anfänglichen Öffnen der Schnittstelle wird im Block 256 eine Übertragung von dem Host 12 mit dem "USB-Vendor-Call" gestartet, um den Transfer zu initialisieren. Der Aufruf WR_START setzt das Signal WR_START um die Adresszähler 60 (wr_count) in der PLD 30 auf die Speicher-Start-Position/Adresse eines Bildes zu setzen. Der Aufruf wird beim Öffnen der Treiber-Schnittstelle über den Aufruf OPEN() (siehe Fig. 9) ausgeführt. Falls der USB-Vendor-Call im Block 256 nicht ausgeführt werden kann, so wird im Block 258 ein Fehlercode erzeugt und zurückgegeben, der ein nicht erfolgreiches Öffnen des USB-Vendor-Calls anzeigt. Ansonsten, wenn also kein Fehler auftritt, wird durch den Block 256 über einen Rückgabewert die erfolgreiche Initialisierung an die An-

wendung 250 bestätigt.

[0080] Jetzt können über die Funktion WRITE() des Treibers Grafik-Bitmaps übergeben werden. Es können nur Bitmaps im passenden Format, hier 320x240x16Bit, übergeben werden. Andere Formate werden vom Treiber mit einem Fehlercode im Rückgabewert zurückgewiesen und verworfen. Sollen andere Formate verwendet werden, so ist der Treiber entsprechend zu modifizieren. Es kann auch vorgesehen sein, dass der Treiber 254 mehrere Formate unterstützt und zunächst das geeignete Format feststellt.

[0081] Fig. 11 verdeutlicht den einen Ablauf des Datentransfers von dem Computer 12 an das Anzeigesystem 14. Die Bitmap-Datenübertragung erfolgt über einen USB-Bulk-Transfer. Der USB-Bus übernimmt beim Bulk-Transfer die Datensicherung (siehe USB Spezifikation). Deshalb kann man sich Treiber- und Firmware-seitig auf einen fehlerfreien Datentransport verlassen. Kann der USB-Host-Controller die Daten nicht zustellen, wird ein Fehlercode zurückgegeben. Die Fehler die zwischen der MCU 28 und der PLD 30 stattfinden und von der MCU-Firmware erkannt werden, werden durch einen USB-Interrupt-Transfer übermittelt. Erkennt der Treiber 252 einen Fehler wird die Datenübertragung abgebrochen und ein Fehlercode zurückgegeben. Bei korrekter Übertragung wartet der Treiber 252 solange, bis die MCU 28 die Anzeige der neuen Bitmap mit einem USB-Interrupt-Transfer bestätigt. Wird dies bestätigt, wird die Anzahl der übertragenen Bytes der schreibenden Anwendung/Applikation 250 zurückgegeben. Die Anwendung 250 erkennt somit eine fehlerfreie Übertragung. Dann kann ein neues Bild übertragen werden. Im Fehlerfall muss die Treiberschnittstelle von der Anwendung geschlossen und erneut geöffnet werden um das System zurückzusetzen. Alternativ kann das System auch über die IOCTL() Funktion zurückgesetzt werden.

[0082] Wie in Fig. 11 zu erkennen ist, wird im Block 260 nach dem Write-Systemruf eine Grafik-Bitmap dem Treiber 252 übergeben. Anschließend wird die Bitmap im Block 262 übertragen. Nachdem die Bitmap vollständig dem USB-Host übergeben wurde, wird im Block 262 gewartet, bis über einen Interrupt-Transfer eine erfolgreiche Übermittlung angezeigt wird. Im Block 266 wird eine erfolgreiche Übertragung angezeigt, indem die Anzahl der übertragenen Bytes, also eine komplette Bitmap, zurückgegeben wird (bei dem hier beispielhaft beschriebenen 16 Bit System die Anzahl der Zeilen einer Bitmap). Anschließend wird die Schnittstelle geschlossen.

[0083] Wird beim Übergeben der Bitmap im Block 260 festgestellt, dass die übergebene Bitmap nicht die richtige Größe bzw. nicht das richtige Format hat, wird über den Block 268 ein Fehlerwert bestimmt und beim Block 266 zurückgegeben. Auch wenn beim Übertragen der Bitmap ein Fehler auftritt, wird durch einen Interrupt-Transfer ein Fehler angezeigt und die Übertragung unterbrochen, über den Block 268 ein Fehlerwert bestimmt und beim Block 266 zurückgegeben. Gleiches gilt, wenn

im Block 264 festgestellt wird, das eine zulässige Wartezeit überschritten wurde.

[0084] Abhängig von den Gegebenheiten kann das erfindungsgemäße Verfahren in Hardware oder in Software implementiert werden. Die Implementierung kann auf einem digitalen Speichermedium, insbesondere einer Diskette oder CD mit elektronisch auslesbaren Steuersignalen erfolgen, die so mit einem programmierbaren Computersystem zusammenwirken können, dass das entsprechende Verfahren ausgeführt wird. Allgemein besteht die vorliegende Erfindung somit auch in einem Computer-Programmprodukt mit auf einem maschinenlesbaren Träger gespeicherten Programmcode zur Durchführung des erfindungsgemäßen Verfahrens, wenn das Computer-Programmprodukt auf einem Rechner abläuft. Mit anderen Worten kann die Erfindung somit als ein Computerprogramm mit einem Programmcode zur Durchführung des Verfahrens realisiert werden, wenn das Programm auf einem Computer abläuft.

[0085] Obwohl oben ein bevorzugtes Ausführungsbeispiel der vorliegenden Erfindung anhand des USB-Formats beschrieben wurde, ist die vorliegende Erfindung nicht hierauf beschränkt. Auch andere Schnittstellenformate, z.B. RS232, Parallelanschluss, Ethernet, etc., können verwendet werden. Auch können die Bilddaten in Bitmaps mit anderer Auflösung oder in anderer Form übergeben werden.

[0086] Bei dem oben beschriebenen Ausführungsbeispiel werden über die Schnittstelle lediglich Pixel-Bilddaten, vorzugsweise in der Form von Bitmaps übertragen. Die vorliegende Erfindung ist jedoch nicht hierauf beschränkt. So kann es bei bestimmten Anwendungsfällen erwünscht sein, zunächst eine Adressierung des Bildspeichers 32 über die Schnittstelle zu steuern. In diesem Fall wird dann vor dem eigentlichen Bilddatenpaket eine Speicheradresse über die Schnittstelle gesendet. Anschließend wird das eigentliche Bilddatenpaket gesendet und die darin enthaltenen Daten basierend auf der vorab erhaltenen Speicheradresse in den Bildspeicher weitergeleitet.

[0087] Wie die obige Beschreibung zeigt, ist die erfindungsgemäße Lösung vorteilhaft, da hier direkt verwendbare Bilddaten (Pixel) übertragen werden, die ohne weitere Umwandlung angezeigt werden können, wodurch der software- und hardwaretechnische Aufwand reduziert wird.

[0088] Bei der Beschreibung des bevorzugten Ausführungsbeispiels wurde die Übertragung der Bilddaten ohne Komprimierung beschrieben. Ist die Datenmenge jedoch groß, so können die Bilddaten vor der Übertragung über den seriellen Bus mittels bekannter Techniken komprimiert werden. Das Extrahieren der Daten an der elektronischen Baugruppe umfasst dann auch des Dekomprimieren der Daten.

Patentansprüche

1. Verfahren zum Betreiben eines Anzeigegeräts (18), dessen Anzeige durch aufeinanderfolgend empfangene Rahmen von Bilddaten erzeugt und kontinuierlich erneuert wird, wobei die Bilddaten ein oder mehrere Pixel definieren, wobei das Verfahren folgende Schritte umfasst:

(a) Bereitstellen der Bilddaten für einen Rahmen durch einen Host-Computer (12) über einen seriellen Bus (10);

(b) Extrahieren der Bilddaten aus den über den seriellen Bus (10) empfangenen Daten an eine elektronische Baugruppe (16), die zwischen dem Host-Computer (12) und das Anzeigegerät (18) geschaltet ist;

(c) Speichern der extrahierten Bilddaten in einem Bildspeicher (32) der elektronischen Baugruppe (16) und Anzeigen der gespeicherten Bilddaten unter Steuerung der elektronischen Baugruppe (16); und

(d) Wiederholen der Schritte (a) bis (c) für jeden der Rahmen.

2. Verfahren nach Anspruch 1, bei dem der Schritt (c) das Sammeln einer vorbestimmten Menge von Bilddaten in dem Bildspeicher (32) umfasst, wobei die Bilddaten bei Erreichen der vorbestimmten Menge angezeigt werden.

3. Verfahren nach Anspruch 2, bei dem das Sammeln der Bilddaten ein Zählen der extrahierten und zwischengespeicherten Bilddaten umfasst.

4. Verfahren nach einem der Ansprüche 1 bis 3, bei dem das Anzeigegerät (18) eine erste Auflösung in horizontaler Richtung und eine zweite Auflösung in vertikaler Richtung aufweist, wobei der Bildspeicher (32) eine Mehrzahl von Zeilenabschnitten aufweist, wobei die Größe der Zeilenabschnitte der horizontalen Auflösung des Anzeigegeräts (18) entspricht, wobei die Anzahl der Zeilenabschnitte der vertikalen Auflösung des Anzeigegeräts (18) entspricht, wobei das Verfahren ferner folgende Schritte umfasst:

beginnend mit einem ersten Zeilenabschnitt, Einschreiben der Bilddaten in den Bildspeicher (32), bis der erste Zeilenabschnitt gefüllt ist;

Inkrementieren des zu beschreibenden Zeilenabschnitts und Einschreiben der Bilddaten in den weiteren Zeilenabschnitt; und

Wiederholen der Schritte des Inkrementierens und Einschreibens, bis die Anzahl der beschriebenen Zeilenabschnitte der vertikalen Auflösung des Anzeigegeräts (18) entspricht.

5. Verfahren nach Anspruch 4, bei dem der Bildspeicher (32) einen ersten Speicherabschnitt (32a) und einen zweiten Speicherabschnitt (32b) umfasst, wobei der erste/zweite Speicherabschnitt (32a) Bilddaten empfängt, während der zweite/erste Speicherabschnitt (32b) Bilddaten zur Anzeige bereitstellt.

6. Verfahren nach einem der Ansprüche 1 bis 5, bei dem dem Anzeigegerät (18) ein Touchscreen (24) zugeordnet ist, wobei das Verfahren ferner folgende Schritte umfasst:

Empfangen von Informationen über die Koordinaten eines berührten Punktes auf dem Anzeigegerät (18);

Umsetzen der empfangenen Informationen in der elektronischen Baugruppe (16) in ein Format entsprechend der Konfiguration des seriellen Bus (10); und

Bereitstellen der empfangenen Informationen an einem Ausgang.

7. Verfahren nach einem der Ansprüche 1 bis 6, bei dem Schritt (a) folgende Schritte umfasst:

Einfügen von zu übertragenden Bilddaten in ein serielles Datenbusformat; und

Bereitstellen der zu übertragenden Bilddaten an einer seriellen Schnittstelle des Host-Computers (12).

8. Verfahren gemäß Anspruch 7 mit folgendem Schritt:

Erzeugen der Bilddaten durch eine Anwendung (250).

9. Verfahren nach einem der Ansprüche 1 bis 8, bei dem der serielle Bus (10) USB-, RS232-, Ethernet- oder V.24-Standard unterstützt.

10. Verfahren nach einem der Ansprüche 1 bis 9, bei dem die Bilddaten in Bitmaps konfiguriert sind.

11. Verfahren nach einem der Ansprüche 1 bis 10, bei dem das Bereitstellen der Bilddaten durch den Host-Computer (12) ein Komprimieren der Bilddaten umfasst, und bei dem das Extrahieren der Bilddaten ein Dekomprimieren der Bilddaten umfasst.

12. Vorrichtung zum Betreiben eines Anzeigegeräts (18), dessen Anzeige durch aufeinanderfolgend empfangene Rahmen von Bilddaten erzeugt und kontinuierlich erneuert wird, wobei die Bilddaten ein oder mehrere Pixel definieren, mit folgenden Merkmalen:

einer seriellen Schnittstelle (20) zum Empfangen der Bilddaten für einen Rahmen über einen seriellen Bus (10) von dem Host-Computer (12);

einer elektronischen Baugruppe (16), die zwischen dem Host-Computer (12) und dem Anzeigegerät (18) geschaltet ist und die ausgebildet ist, um für jeden Rahmen die Bilddaten aus den über den seriellen Bus (10) empfangenen Daten zu extrahieren, die extrahierten Bilddaten in einem Bildspeicher (32) der elektronischen Baugruppe (16) zu speichern und die gespeicherten Bilddaten unter Steuerung der elektronischen Baugruppe (16) anzuzeigen.

13. Vorrichtung nach Anspruch 12, die der die elektronische Baugruppe (16) ausgebildet ist, um eine vorbestimmte Menge von Bilddaten in dem Bildspeicher (32) zu sammeln und um die Bilddaten bei Erreichen der vorbestimmten Menge anzuzeigen.

14. Vorrichtung (14) nach Anspruch 12 oder 13, bei dem das Anzeigegerät (18) eine erste Auflösung in horizontaler Richtung und eine zweite Auflösung in vertikaler Richtung aufweist, und bei dem der Bildspeicher (32) eine Mehrzahl von Zeilenabschnitten aufweist, wobei die Größe der Zeilenabschnitte der horizontalen Auflösung des Anzeigegeräts (18) entspricht, wobei die Anzahl der Zeilenabschnitte der vertikalen Auflösung des Anzeigegeräts (18) entspricht.

15. Vorrichtung nach Anspruch 14, bei dem der Bildspeicher (32) einen ersten Speicherabschnitt (32a) und einen zweiten Speicherabschnitt (32b) umfasst, wobei der erste/zweite Speicherabschnitt (32a, 32b) konfigurierbar ist, um Pixeldaten zu empfangen, während der zweite/erste Speicherabschnitt (32b, 32a) Pixeldaten zur Anzeige bereitstellt.

16. Vorrichtung nach einem der Ansprüche 12 bis 15, bei dem dem Anzeigegerät (18) ein Touchscreen (24) zugeordnet ist.

17. Vorrichtung gemäß einem der Ansprüche 12 bis 16, bei dem der serielle Bus (10) den USB-, RS232-, Ethernet- oder V.24-Standard unterstützt.

18. Vorrichtung gemäß einem der Ansprüche 12 bis 17,

bei dem die Pixeldaten in Bitmaps angeordnet sind.

19. Vorrichtung nach einem der Ansprüche 12 bis 18, bei der die Bilddaten komprimierte Bilddaten sind, und bei der die elektronische Baugruppe (16) ausgebildet ist, um die Bilddaten zu dekomprimieren. 5

10

15

20

25

30

35

40

45

50

55

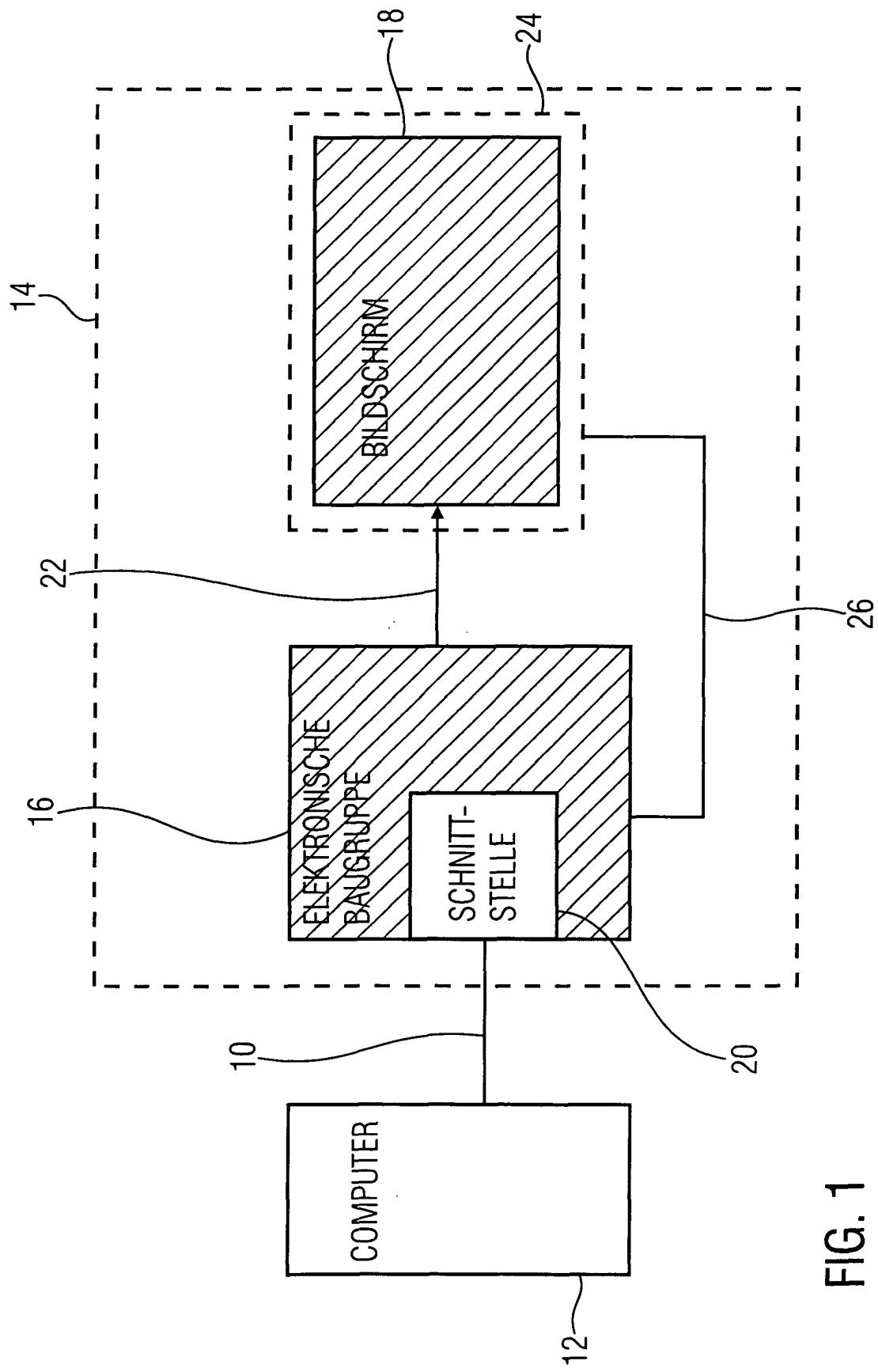


FIG. 1

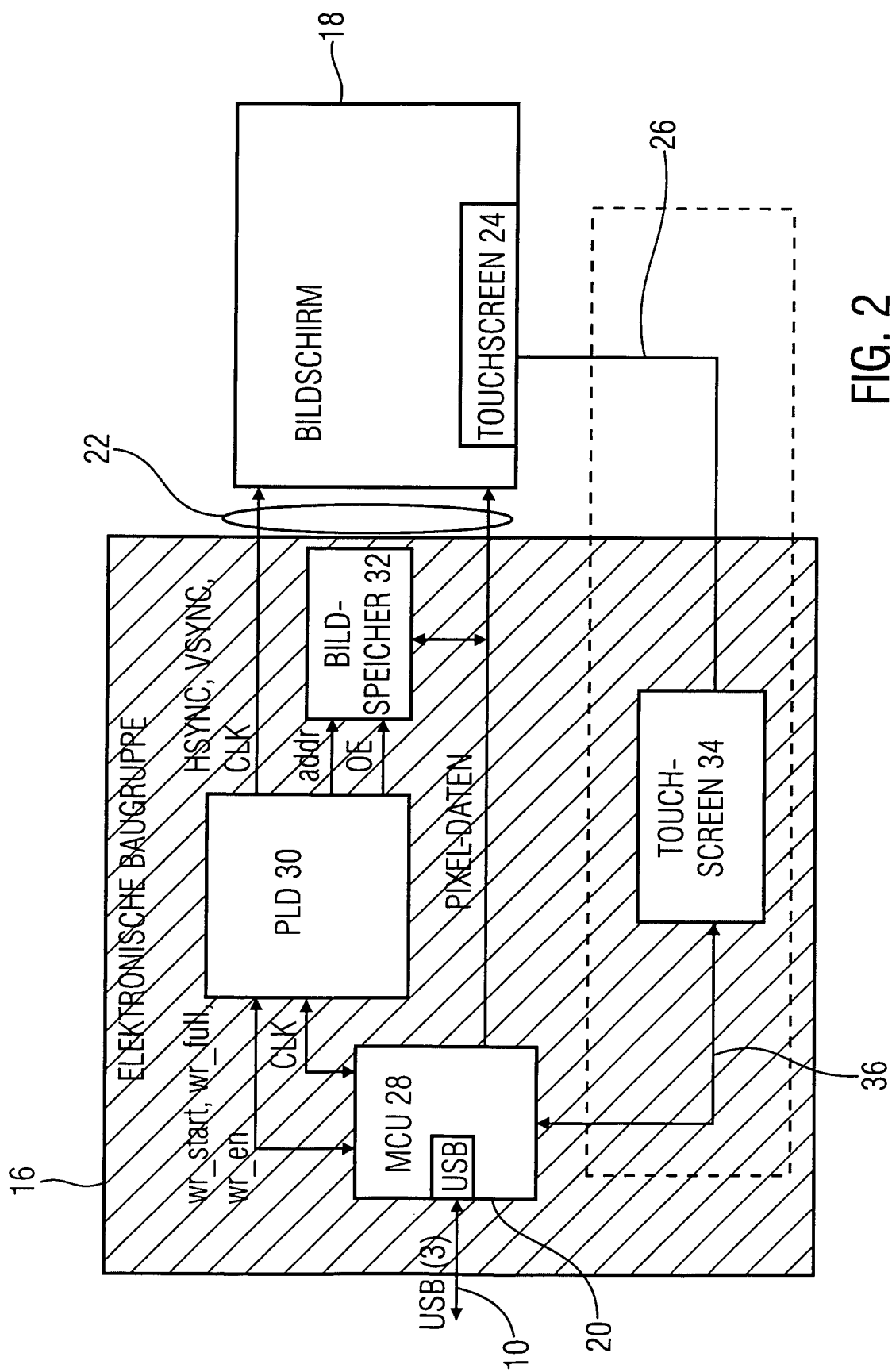


FIG. 2

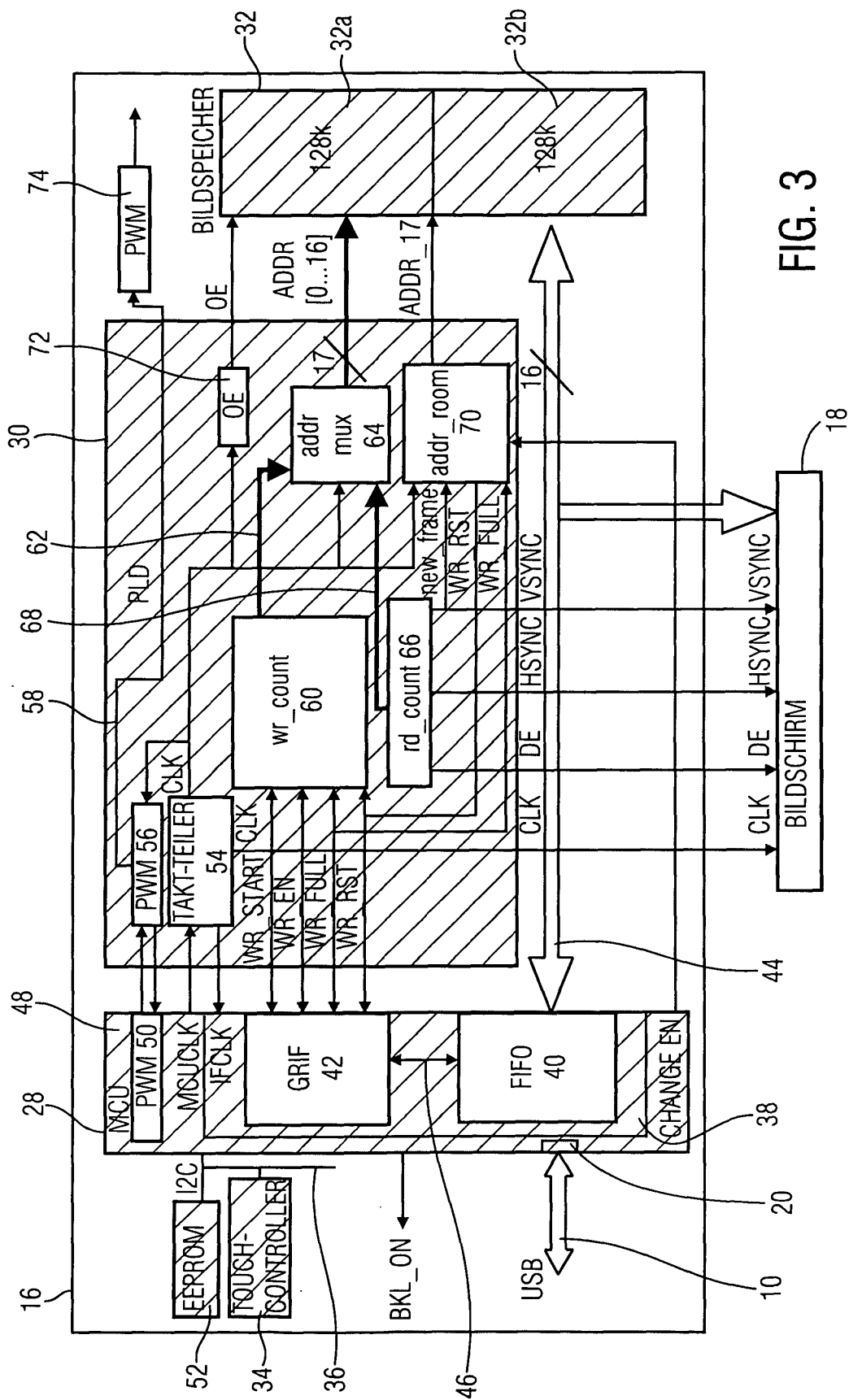


FIG. 3

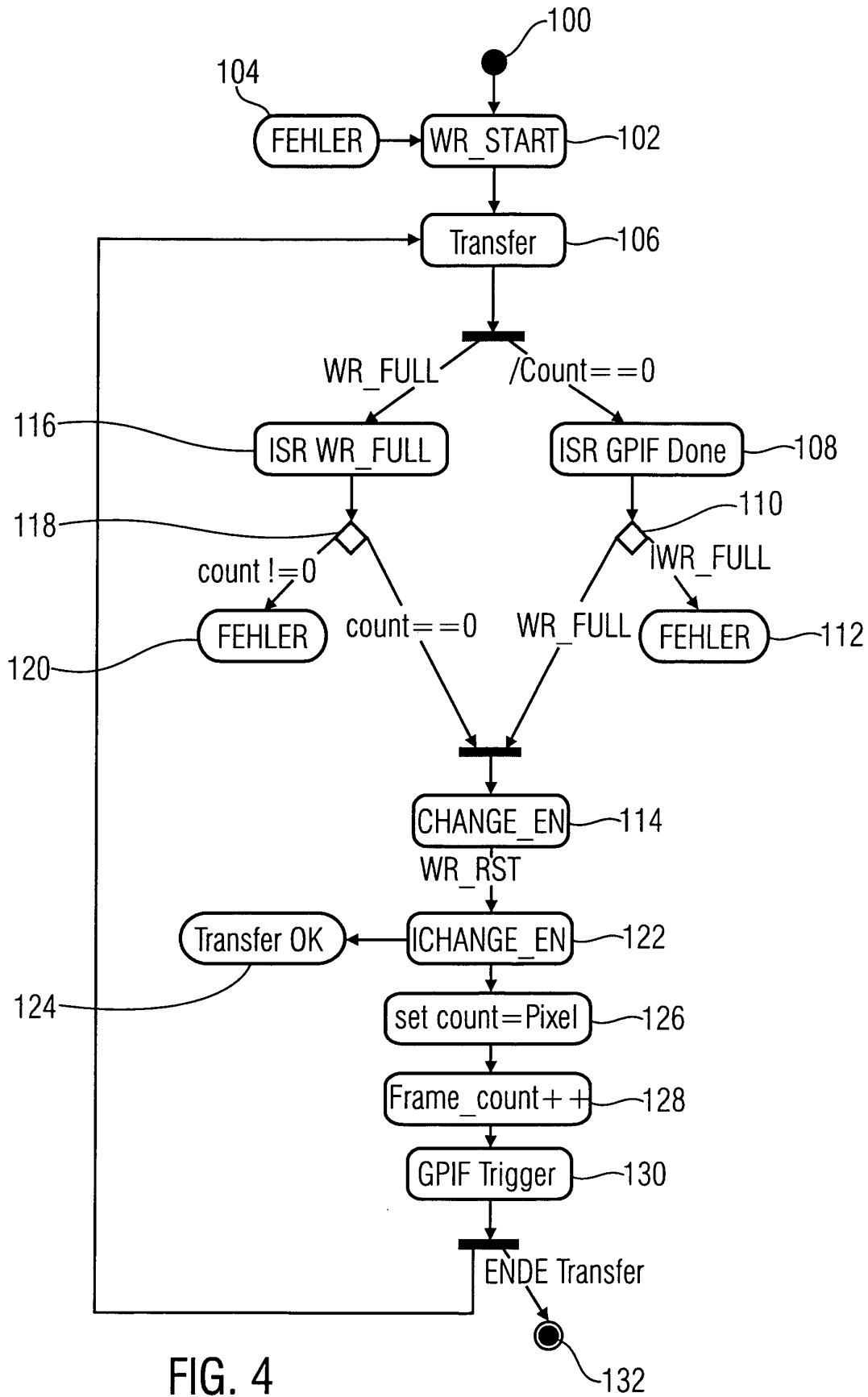


FIG. 4

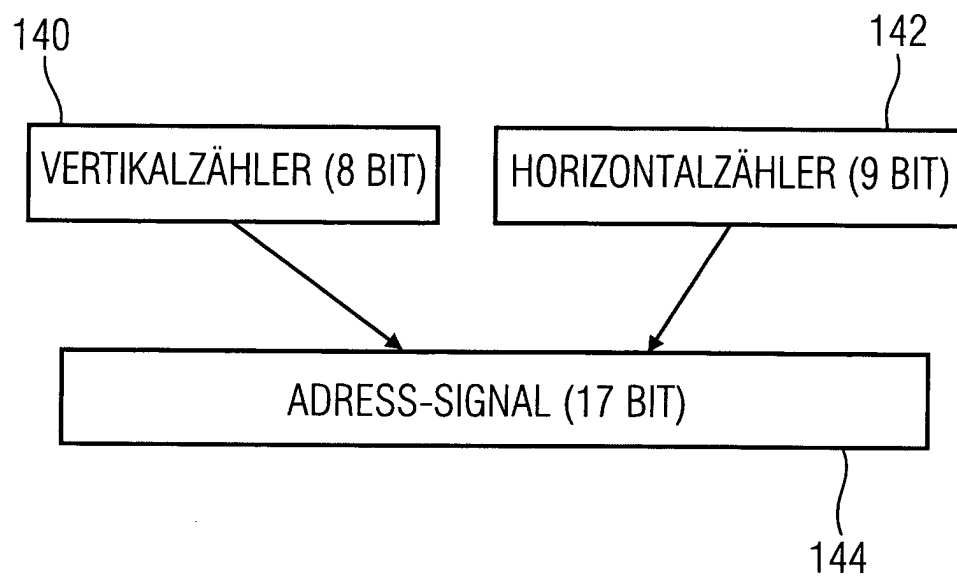


FIG. 5

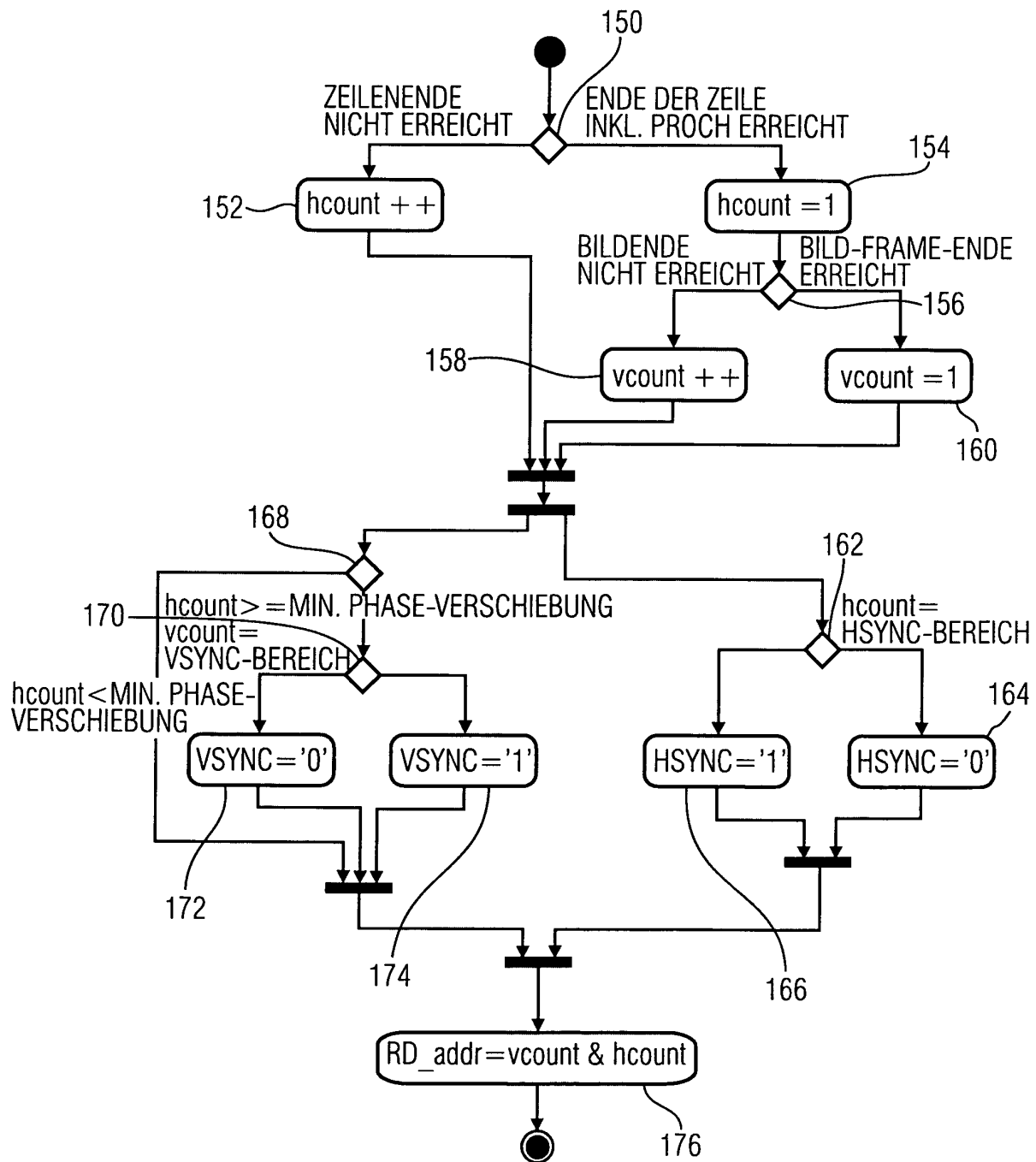


FIG. 6

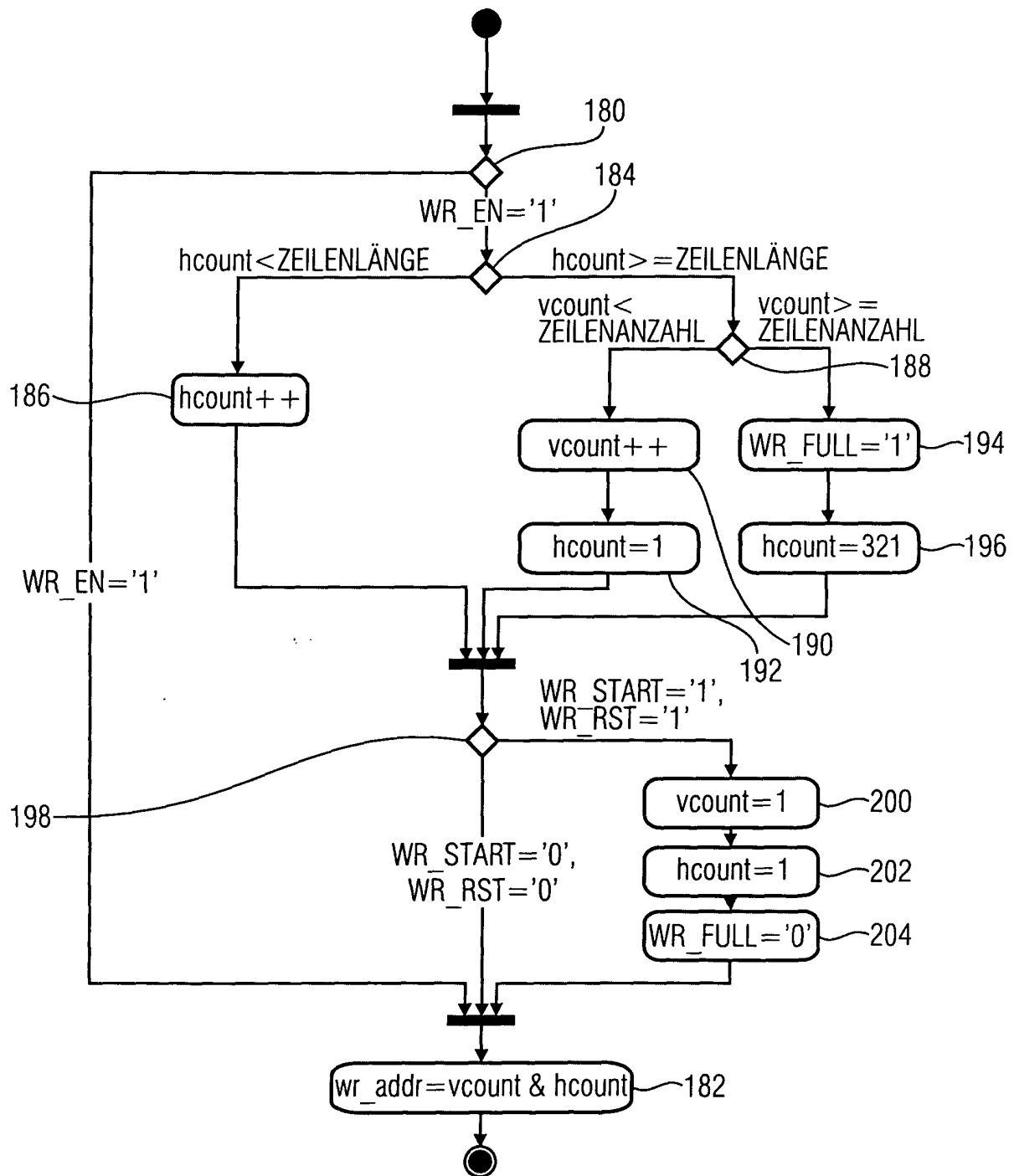


FIG. 7

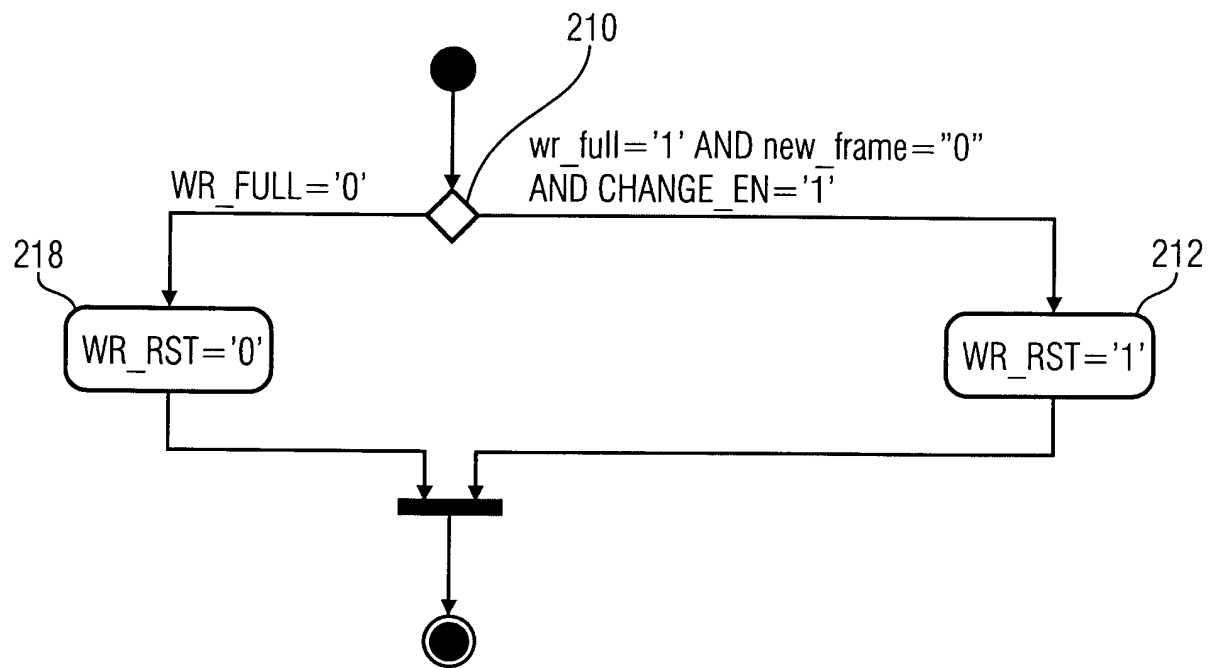


FIG. 8

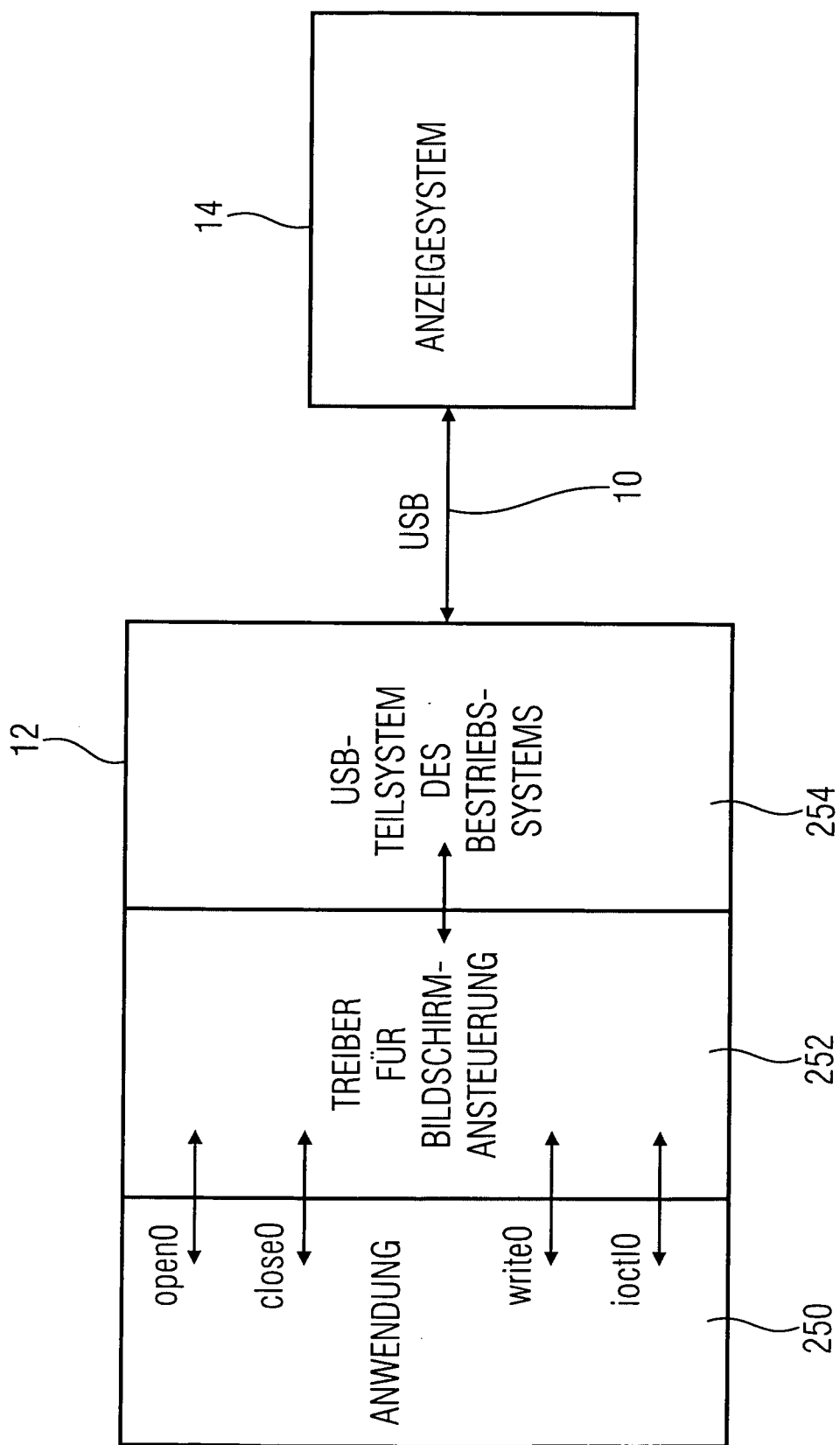


FIG. 9

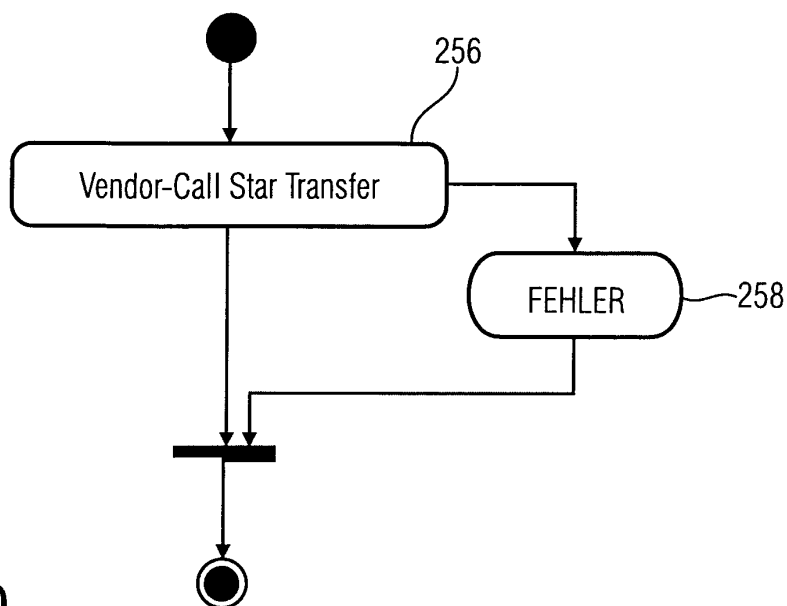


FIG. 10

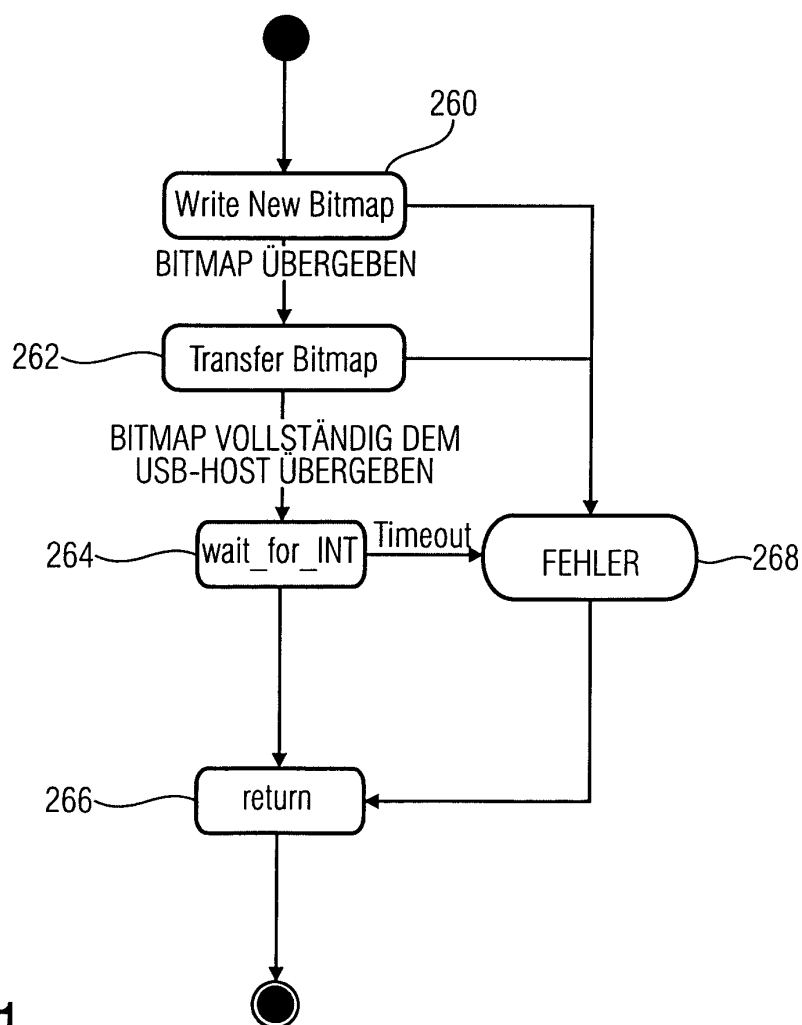


FIG. 11