(19) BUNDESREPUBLIK DEUTSCHLAND

DEUTSCHES PATENTAMT

(12) Übersetzung der europäischen Patentschrift

(87) EP 0 316 325 B1

(10) DE 37 51 037 T 2

(51) Int. Cl.[6]: G 09 G 1/00
G 09 G 3/00

DE 37 51 037 T 2

(21) Deutsches Aktenzeichen: 37 51 037.1
(86) PCT-Aktenzeichen: PCT/US87/01625
(86) Europäisches Aktenzeichen: 87 904 781.9
(87) PCT-Veröffentlichungs-Nr.: WO 88/00750
(86) PCT-Anmeldetag: 14. 7. 87
(87) Veröffentlichungstag der PCT-Anmeldung: 28. 1. 88
(87) Erstveröffentlichung durch das EPA: 24. 5. 89
(87) Veröffentlichungstag der Patenterteilung beim EPA: 1. 2. 95
(47) Veröffentlichungstag im Patentblatt: 21. 9. 95

(30) Unionspriorität: (32) (33) (31)
18.07.86 US 887053

(73) Patentinhaber:
Commodore-Amiga, Inc., Los Gatos, Calif., US

(74) Vertreter:
Grünecker, Kinkeldey, Stockmair & Schwanhäusser, Anwaltssozietät, 80538 München

(84) Benannte Vertragstaaten:
DE, FR, GB, NL

(72) Erfinder:
MICAL, Robert, J., San Mateo, CA 94404, US

(54) CURSORGESTEUERTES BENUTZERPERIPHERIESYSTEM.

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 Patentamt inhaltlich nicht geprüft.

DE 37 51 037 T 2

EP 87 904 781.9-2202
COMMODORE-AMIGA, INC.

Die vorliegende Erfindung betrifft ein Verfahren zur Verwendung in einem Datenverarbeitungssystem und zum Anzeigen und Auswählen von für eine Bedienperson auswählbarer Menüteile (menu items).

In US-A-4 464 652 ist ein cursorgesteuertes Benutzer-Interface-System beschrieben. In einem derartigen System wird eine Eingabeeinrichtung, die als Maus bezeichnet wird, von einem Benutzer verwendet, um einen Cursor (Zeiger-Symbol) auf einem Bildschirm zu bewegen, um dadurch Steuerbefehle zu erzeugen. Die Maus weist eine Taste auf, die es dem Benutzer gestattet, Steuerbefehle zu erzeugen, um damit eine gewünschte Auswahl zu treffen und um andere Steuerfunktionen auszuführen. Auswahlmöglichkeiten werden dadurch festgelegt, indem der Cursor auf verschiedene Menüs gesetzt wird, welche aufgerufen werden und auf dem Bildschirm dargestellt werden. Individuelle Menüpunkte sind auswählbar, indem die Maustaste betätigt wird, wenn sich der Cursor auf dem ausgewählten Punkt befindet.

Herkömmliche Systeme dieser Art sind aufgrund mehrerer Faktoren in ihren Fähigkeiten stark eingeschränkt. Zunächst ist ständig ein Menübalken im oberen Bereich des Anzeigenschirms vorhanden, um es dem Benutzer zu ermöglichen, die Menüs aufzurufen. Das ständige Vorhandensein des Menübalkens verkleinert die verwendbare Bildschirmfläche. Weiterhin gestattet eine einzelne Maustaste nur eine begrenzte Auswahl und Steuerung.

Aus Byte, vol. 6, no. 8, August 1981, Seiten 90-147, L. Tesler, "The Smalltalk Environment", ist ein Window-Konzept bekannt,

bei dem mehrere Fenster und in den Fenstern enthaltene Scheiben (panes) auf dem Bildschirm zu sehen sind. Der Benutzer kann gleichzeitig nur in eine Scheibe eintreten, die dann als aktive Scheibe (active pane) bezeichnet wird und das dazugehörige Fenster wird dann als aktives Fenster bezeichnet. Um eine andere Scheibe (pane) des gleichen Fensters zu aktivieren, muß der Cursor auf die neue Scheibe bewegt werden und um ein anderes Fenster zu aktivieren, muß der Cursor in ein anderes Fenster bewegt werden und es muß eine Auswahltaste auf der Zeigeeinrichtung, die die Cursorbewegung steuert, gedrückt werden. Jede Scheibe kann einen Rollbalken (scroll bar) aufweisen und jedes Fenster kann ein Menü aufweisen, wobei beide vorzugsweise nur dann zu sehen sind, wenn das entsprechende Fenster und die entsprechende Scheibe aktiviert sind. Weiterhin ist beschrieben, daß bei Maschinen, die eine Zeigereinrichtung mit drei Tasten aufweisen, es möglich ist, daß das aktive Menü verborgen ist, bis eine von zwei Menütasten gedrückt wird, wobei dann das zugeordnete Menü erscheint und so lange erhalten bleibt, bis die Taste wieder losgelassen wird.

Ausgehend von diesem Stand der Technik ist es die Aufgabe der vorliegenden Erfindung, ein verbessertes Verfahren zum Anzeigen und Auswählen von durch den Benutzer auswählbarer Menüteile anzugeben.

Diese Aufgabe wird durch die Gegenstände der Ansprüche 1 und 2 gelöst.

Im folgenden werden bevorzugte Ausführungsformen der vorliegenden Erfindung unter Bezugnahme auf die beiliegenden Zeichnungen näher erläutert, welche zeigen:

Figur 1 zeigt ein Systemblockschaltbild zur Verdeutlichung grundlegender Komponenten eines beispielhaften PC-Systems, das für die vorliegende Erfindung geeignet ist.

Figur 2 zeigt ein Funktionsschaltbild, zur Verdeutlichung der grundlegenden Betriebszustände, wie sie bei dem Menü-Steuersystem der vorliegenden Erfindung werden.

Figur 3 ist ein schematisches Diagramm zur Verdeutlichung der Zusammensetzung unterschiedlicher Menü-Zustandsdatenwörter (variablen), wie sie gemäß einer bevorzugten Ausführungsform der Erfindung verwendet werden.

Figur 4 ist ein schematisches Diagramm des Aussehens des Kopfblockes (header block), des Menüblocks, und des Submenüblocks, wie sie im Zusammenhang mit dem System der Erfindung verwendet werden können.

Figuren 5a bis 5c sind funktionelle Flußdiagramme zur Ver- Verdeutlichung der Steuersoftware, die beim Betrieb des Systems beim Menüzustand der Figur 2 ausgeführt werden.

Figur 6 ist ein funktionelles Flußdiagramm zum Verdeutlichen der Software, die durch das System der Erfindung ausgeführt wird, wenn der Menü-Ausgeführt-Zustand der Figur 2 vorliegt.

In Figur 1 umfaßt das System eine Maus-Eingabe-Einrichtung 10 mit einem Ball 30 und einem Paar von Steuertasten 26 und 28.

Die Tasten betätigen momentane Kontaktschalter, um Auswahlsignale und andere Befehle zu erzeugen. Die Taste 26 wird als Menü-Taste und die Taste 28 wird als Auswahltaste bezeichnet. Wie für den Fachmann klar ist, bewegt der Benutzer die Maus 10 entlang einer planaren Oberfläche, wie beispielsweise einer Tischfläche, um dadurch Cursor-Positions-Eingabebefehle zu erzeugen, welche der CPU 12 zugeführt werden. Der Ball 30 wirkt mit einem Mechanismus zusammen, welcher die Bewegung der Maus in x-y-Signale umsetzt, welche von der CPU wiederum verwendet werden, um die Position des Cursorsymbols auf dem Anzeigenschirm 17 der Anzeigeneinheit 16 zu steuern. Eine Anordnung zum Umwandeln der Bewegung des Balles 30 in x-y-Befehle ist im obengenannten Patent 4 464 652 beschrieben.

Die CPU 12 enthält eine Eingabe/Ausgabe-Steuereinheit 18, ein RAM 20, ein Anzeigenspeicher 22, ein Videoschnittstellenschaltkreis (VIC) 24 und ein Mikroprozessor 25. Diese Einheiten sind dem Fachmann bekannt und werden unter Steuerung der Systemsoftware betätigt, um verschiedene Eingaben zu verarbeiten und um Ausgaben zu erzeugen, die notwendig sind, um gewünschten Text und Graphikinformation auf dem Bildschirm 17 darzustellen. Ein Plattenantrieb 14 ist vorgesehen, um Steuer- und Anwendungssoftware zu gegebenen Zeiten auszutauschen und um eine Quelle zur Massenspeicherung für das System zur Verfügung zu stellen.

Der Anzeigenspeicher (Bildschirmspeicher) 22 stellt einen besonderen Abschnitt des RAMs dar, welcher verwendet wird, um Bit-Muster (Pixeldaten) zu speichern, welche von dem VIC 24 in geeigneter Synchronisation mit dem Bildschirmstrahl der Einheit 16 zugewiesen werden, um die gewünschte Graphik oder den gewünschten Text zur Verfügung zu stellen.

Die Figur 2 zeigt ein Zustandsdiagramm, das die grundlegenden funktionalen Zustände angibt, in die die menügesteuerte Software organisiert ist. Wenn die Menütaste 26 durch den Benutzer gedrückt wird, geht die Software vom Idle-Zustand 100 zum Start-Menü-Zustand 200 über. In dem zuletztgenannten Zustand werden Verifizierungsoperationen ausgeführt, um zu bestimmen, ob das System mit den Menü-Operationen weiterfahren soll, oder ob spezielle Umstände vorliegen, welche die Unterbrechung oder das Löschen von Menüoperationen notwendig machen. Falls ein Eingriff notwendig ist, geht das System in den Lösch-Menü-Zustand 300 über und kehrt dann zu Idle zurück. Die Verifikations- und Lösch-Funktionen stellen keinen Teil der vorliegenden Erfindung dar und werden deshalb nicht mehr weiter beschrieben.

Falls das System feststellt, daß Menüoperationen stattfinden können, geht die Software vom Zustand 200 in den Menüzustand 400 über. In diesem Menüzustand wird die Software ausgeführt, um den notwendigen Text und die notwendige Graphik zu erzeugen, um den Kopfblock (header block), die Menüblöcke und die Submenüblöcke, die während der Menüoperation verwendet werden, anzuzeigen. Der Zustand 400 steuert auch die Akkumulation von verschiedenen Menü- und Submenü-Punktauswahlen, wie sie durch den Benutzer durch Betätigung der Maustasten vorgenommen werden.

Wenn der Benutzer die Menütaste 26 losläßt, um so eine Menü-Session zu beenden, geht das System vom Menüzustand 400 in den Menü-Erledigt-Zustand 500 über. In dem Menü-Erledigt-Zustand, werden dem Operator seine letzten Auswahlen zu früheren Auswahlen addiert, die Menügraphiken gelöscht, geeignete Steuerwörter rückgesetzt, und eine Auswahlnachricht erzeugt und

zum Anwendungssystem übertragen, um zu ermöglichen, daß die Menü-Auswahlen des Benutzers ausgeführt werden.

Figur 3 zeigt in schematischer Weise drei Formate von Steuerwörtern (variablen), wie sie in dem Menüsteuersystem der vorliegenden Erfindung eingesetzt werden können. Die Menü-Drawn-Variable 32 enthält ein erstes Feld, in dem codierte Daten gespeichert werden, die einen bestimmten Menüblock darstellen, der momentan angezeigt wird. Das zweite Feld der Menü-Drawn-Variable zeigt einen bestimmten Submenü-Block an, soweit vorhanden, welcher momentan angezeigt wird. Die Menü-Auswahl-Variable 34 weist ein erstes Feld auf, welches den momentan aktiven Menüteil (Menütitem) anzeigt und ein zweites Feld, welches den momentanen aktiven Submenü-Teil anzeigt, falls vorhanden. Ein Menü- oder Submenü-Teil ist dann aktiv, wenn der maus-gesteuerte Cursor auf dem auf dem Bildschirm angezeigten Block positioniert ist.

Die OptionList-Variablen 36-1 bis 36-N stellen, in einem ersten Feld, Menüteile dar, die tatsächlich während der Menü-Session durch den Benutzer ausgewählt wurden. Das zweite Feld jedes OptionList-Wortes zeichnet die Submenü-Teile auf, die tatsächlich während der Menü-Session ausgewählt werden. Der ausgewählte Menüteil wird durch eine Codierung angezeigt, die den speziellen Menüblock identifiziert und den speziellen Menüteil innerhalb des Blockes der ausgewählt wurde, identifiziert. Eine ähnliche Codierung wird im zweiten OptionList-Feld verwendet, um einen ausgewählten Submenü-Teil zu identifizieren.

Figur 4 zeigt schematisch ein Beispiel einer Bildschirmgraphik, die geeignet ist, verwendet zu werden, um bestimmte Menüoptionen für den Benutzer darzustellen. Ein Kopfblock 40

erstreckt sich im oberen Teil des Bildschirms 17 und enthält vier Unterbereiche A, B, C und D. Der Benutzer bedient die Maus 10, um den Cursor 50 in einen ausgewählten Unterbereich zu positionieren. In der Praxis würden die Anzeigen A, B usw. durch File-Namen, funktionale Operationen, Format-Optionen u.dgl. ersetzt sein. In dem Falle der Figur 4 wird, wenn der Benutzer den Cursor 50 auf den Unterbereich B positioniert, durch die Steuersoftwar eine Anzeige auf den Menüblock 42 erzeugt. Der Menüblock 42 enthält vier Unterbereiche B0, B1, B2 und B3, die vier Menüteile repräsentieren, die durch den Benutzer ausgewählt werden können. Wenn der Operator den Cursor 50 in die Box B2 bewegt, erzeugt die Steuersoftware automatisch eine Anzeige des Untermenü-Blockes 44. Der letztere enthält zwei Unterbereiche B20 und B21, durch die zwei mögliche Submenü-Optionen dargestellt werden. Die Menüteile können oder können auch nicht mit Untermenüs assoziiert sein, jeweils abhängig von der Wahl des Programmierers.

Die Randbedingungen der Anzeige, wie sie in Figur 4 gezeigt sind, werden innerhalb des Systems durch die MenüDrawn-Variablen 32 (Figur 3) aufgezeichnet, welche in ihrem ersten Feld eine codierte Identifikation des Untermenü-Blockes 44 enthält. Die Menüauswahl-Variable 34 weist in ihrem ersten Feld eine Codierung auf, um anzuzeigen, daß der aktive Menüteil B2 ist. Das zweite Feld des Menü-Auswahl-Wortes ist codiert um anzuzeigen, daß das aktive Untermenü B21 ist. Wie im folgenden im Detail beschrieben wird, wird, wenn der Benutzer die Auswahltaste 28 drückt, wenn sich die Anzeige in einem Zustand befindet, wie sie in Figur 4 gezeigt ist, ein OptionList-Wort 36 erzeugt. Das erste Feld des OptionList-Wortes ist so codiert, um anzuzeigen, daß der ausgewählte Menüteil der Teil B2 des Menüblocks 42 ist und das zweite Feld des erzeugen Option-List-Wortes ist so codiert, daß angezeigt wird, daß der

ausgewählte Untermenü-Teil der Teil B21 des Untermenü-Blockes 44 ist. Da der Operator weiterhin die Menü-Taste 26 im gedrückten Zustand hält, wird die Menü-Session nicht beendet und der Benutzer kann den Curser 50 weiterhin beeinflussen und mit der Taste 28 eine weitere Auswahl treffen, um weitere OptionList-Worte zu erzeugen.

Wenn der Operator die Menütaste 26 losläßt, wenn ein Menü wie in Figur 4 gezeigt ist, wird die Menüsession beendet und das gleiche Option-List-Wort wie oben beschrieben wird, erzeugt.

Die Figuren 5a-5c zeigen in Flußdiagrammform die Steuersoftware, wie sie für die vorliegende Erfindung eingesetzt wird, um die Funktionen des Menüzustandes 400 auszuführen. Wie oben angesprochen, wird der Menüzustand betreten, wenn der Benutzer die Menütaste 25 drückt und eine spezielle Unterbrechung nicht erforderlich ist. Das Steuerprogramm fährt vom Startmenü-Zustand 401 (Figur 5a) zum Schritt 403 fort, wodurch bewirkt wird, daß Daten, die den Kopf-Block darstellen, z.B. den Block 40 der Figur 4, aus dem RAM-Speicher 20 geholt werden. Das RAM 20 speichert alle Daten, die Kopf-Block oder die Blöcke, die Menüblöcke und Untermenüblöcke, wie sie erforderlich sind, um die Menü-Operationen für ein spezielles Anwanderprogramm zu implementieren, definieren. Diese Menüdaten, sowie die geeignete Steuersoftware, ist anfänglich auf einer Platte gespeichert, die in dem Plattenlaufwerk 14 eingesetzt ist (Figur 1). Wenn die Plattendaten anfänglich in die CPU geladen werden, oder auf sie in anderer Weise während des Systembetriebs zugegriffen wird, werden die Datenstrukturen, wie sie in den Menüdaten enthalten sind, zum RAM 20 transferiert. Jede grundlegende Datenstruktur definiert einen speziellen Kopfblock, Menüblock oder Submenüblock entsprechend

dem Text und/oder der Graphik, wie sie gebraucht werden, um den Block anzuzeigen. Die Datenstruktur enthält auch bestimmte Formatdaten, wie sie notwendig sind, um den Ort des Blockes auf dem Bildschirm zu definieren, um weiterhin die Größe und die Form des Blockes, seine Farbe etc. zu definieren. Auf die Datenstrukturen kann innerhalb es RAMS auf beliebige Weise zugegriffen werden, beispielsweise dadurch, daß Menü-Blockstrukturen mit ihren entsprechenden Kopf-Block-Strukturen verknüpft werden und indem die Submenü-Blockstrukturen mit ihrer entsprechenden Menü-Block-Struktur verknüpft werden.

In Schritt 403 der Figur 5a wird die besondere Kopf-Block-Datenstruktur, die von dem Anwendungsprogramm aufgerufen wird, aus dem RAM gelesen und dem Schritt 405 in ein Bit-Muster konvertiert, welches eine visuelle Darstellung des Kopfblockes darstellt. Das Bit-Muster wird in eine besondere Sektion des Bildschirmspeichers 22 (Schritt 405) geladen und wird danach über die VIC 24 angezeigt, so daß es als Block 40 im oberen Bereich des Bildschirms wie in Figur 4 gezeigt, erscheint. Die VIC 24 arbeitet in einer Weise, daß die Bilddaten die auf dem Bildschirm 17 durch die Kopf-Block-Bilddaten verdrängt werden, an geeigneten Pixel-Stellen innerhalb der "off-screen"-Sektion des Bildschirmspeichers gepuffert werden. Wenn der Kopf-Block später gelöscht wird, werden die gepufferten Bilddaten wieder in das Videobild eingesetzt, um das Bild in voller Größe darstellen zu können.

Nachdem der Kopf-Block angezeigt wurde, wird der Schritt 407 ausgeführt, um zu bestimmen, ob der Cursor sich innerhalb eines Bereiches des Kopf-Block-Bildes befindet. Dies wird erreicht, indem die momentane Cursorposition, wie sie von den Eingangssignalen von der Faust 10 bestimmt werden, mit den Kopf-Block-Ortsdaten, wie sie in der Kopf-Block-Datenstruktur

gespeichert sind, verglichen werden. Falls sich der Cursor nicht in dem Kopfblock befindet, wird im Schritt 409 überprüft, ob eine Menütaste losgelassen wurde und falls dies nicht der Fall ist, wird zum Schritt 407 über eine Abtastschleife die später beschrieben wird, zurückgekehrt. Diese Schleife wird wiederholt, bis der Benutzer die Maus 10 betätigt hat, um dem Cursor auf das Gebiet des Kopf-Blockes zu bringen. Wenn dies der Fall ist, verzweigt die Software vom Schritt 407 zum Schritt 415, wo das MenüDrawn-Steuerwort 32 gegenüber der momentanen Cursorposition verglichen wird. Da der Menüzustand gerade eingetreten ist, befindet sich die MenüDrawn-Variale im Reset-Zustand, so daß im Schritt 415 eine Ungleichheit erhalten wird und das Programm zum Schritt 417 verzweigt. Da momentan kein Menüblock angezeigt wird, ist die Löschfunktion des Schritts 417 nicht effektiv und das Programm fährt mit Schritt 419 fort.

Im Schritt 419 werden die geeigneten Menüdaten aus dem RAM entsprechend der momentanen Position des Cursors geholt. Der Menüblock-Datenstruktur-Zugriffscode basiert auf die Bildschirmstellencoordinaten des Kopfblockunterbereichs B (bezugnehmend auf Figur 4). Da sich die Cursorposition innerhalb des Unterbereichs des Kopfblockes befindet, werden die Daten für den Menüblock 42 (Figur 4) aus dem RAM gelesen. Im Schritt 421 werden eine Reihe von Operationen ähnlich denen aus Schritt 415 ausgeführt, um die Menüblock-RAM-Daten in ein Bitmuster zu konvertieren, welches dann in Form eines Blockes 42 wie in Figur 4 gezeigt ist, angezeigt zu werden. Es sei angemerkt, daß das Menüblockbild direkt unterhalb und leicht überlappend zur Kopfblock-Untersektion B angezeigt wird. Das Programm fährt dann mit Schritt 423 fort, wobei das MenüDrawn-Steuerwort 32 erneuert wird, um dem Menüblock anzuzeigen, welcher dann angezeigt wird. Da zu dieser Zeit kein Submenü-

Block angezeigt wird, verbleibt das zweite Feld der MenüDrawn-Variable 32 im Reset-Zustand. Das Programm geht dann vom Schritt 432 zum Schritt 427 über, so wie dies in Figur 5b gezeigt ist.

Im Schritt 427 überprüft die Software, ob der Cursor sich innerhalb des angezeigten Menüblocks befindet. Dies wird erreicht, indem die momentane Cursorpositions-Daten, wie sie von der Maus 10 erhalten werden, mit den Menüblockorts-Koordinaten, wie sie in der Menüblock-Datenstruktur gespeichert sind, verglichen werden. Falls der Benutzer den Cursor noch nicht auf den Menüblock bewegt hat, um einen Menüteil auszuwählen (der Benutzer kann den Cursor bisher über den Kopf-Block geführt haben, aber nie direkt in den angezeigten Menüblock bewegt haben) kehrt das Programm zum Schritt 407 zurück. Es sei angenommen, daß der Cursor sich immer noch im Kopfblock befindet, so wird der Schritt 415 wiederholt. Falls sich der Cursor in eine unterschiedliche Untersektion des Kopfblockes bewegt hat, wird im Schritt 415 ein ungleiches Ergebnis erhalten und die Schritte 417, 19, 421 und 423 werden dann ausgeführt, um den ursprünglichen Menüblock zu löschen und um einen neuen Menüblock der mit dem Kopfblockunterabschnitt' assoziiert ist, der momentan angezeigt wird.

Falls der Cursor jedoch von dem Kopfblock-Unterabschnitt in den angezeigten Menüblock bewegt wird, der Menüblock bisher jedoch nicht betreten wurde, wird im Schritt 415 ein gleiches Ergebnis erzeugt und die Software geht vom Schritt 425 zum Schritt 427 über (Figur 5b) und wird die Schleife 427-407-415-425 ausführen, bis der Cursor den angezeigten Speicherblock betritt. Wenn dies geschieht, wird vom Schritt 427 zum Schritt 429 übergegangen.

Im Schritt 429 wird die Menü-Auswahl-Variable 34 gegenüber der Cursorposition überprüft und es ergibt sich ein ungleiches Ergebnis. Dieses Ergebnis tritt auf, da sich das Menü-Auswahl-Steuerwort noch immer im Reset-Zustand befindet und die Cursorposition den Ortscode für den Menüteil B0 entspricht. Dies führt zur Ausführung der Programmschritte 439, 441, 443. Im Schritt 439 wird der Menüteil, in dem sich der Cursor momentan befindet (B0) hell unterlegt. Dies kann auf verschiedene Arten erfolgen, z.B. durch Anzeigen einer dicken Begrenzungslinie, um die Box, durch das Verändern der Farbe der Box, durch Invertieren des Box-Bildes usw..

Diese Funktion macht es für den Benutzer leichter, zu erkennen, welcher Menüteil momentan als Kandidat für mögliche Auswahlen zur Verfügung steht.

Im Schritt 441 wird jeder vorher hell unterlegte Menüteil und jegliche vorher angezeigte Submenü-Blockbilder gelöscht und die gepufferten Bilddaten die das gelöschte Material repräsentieren, werden zurück in das Video gebracht. Schritt 443 bewirkt, daß die Menü-Auswahl-Variable 34 erneuert wird, um im Feld 1 den Code des momentan aktiven Menüteils zu reflektieren, d.h. den Teil B0 wie er in Figur 4 gezeigt ist.

Wenn der Schritt 445 betreten wird, wird die angezeigte Menü-Block-Datenstruktur überprüft, um zu bestimmen, ob der aktive Menüteil erforderlich macht, daß ein Submenü-Block angezeigt wird. Falls dem so ist, werden die Programmschritte 447, 449 und 451 ausgeführt. Im Schritt 447 wird auf das RAM 20 zugegriffen, um die Datenstruktur, die den erforderlichen Submenü-Block definiert, zu lesen. Im Schritt 449 werden eine Reihe von Operationen ähnlich denen in den Schritten 405 und 421 ausgeführt, um die Submenü-Daten in ein Bitmuster zu

konvertieren, welches dann in einem "off-screen"-Abschnitt des Anzeigespeichers gespeichert wird, und danach auf dem Bildschirm angezeigt wird. Wenn die Anzeige initiiert wird, werden die ersetzten Bilddaten in den gleichen "off-screen"-Speicherabschnitt übertragen, um dort gepuffert zu werden. Im Schritt 451 wird die MenüDrawn-Variable 32 erneuert, um in ihrem zweiten Feld den momentan aktiven (angezeigten) Submenü-Block anzuzeigen, z.B. Block 44, wie er in Figur 4 gezeigt ist.

War es für die momentan aktive Menüstelle nicht notwendig, einen Submenü-Block anzuzeigen, so bewirkt Schritt 445, daß zum Schritt 431 verzweigt wird. Der letztere Schritt bestimmt, ob die Menütaste 26 noch immer gedrückt ist und falls dies der Fall ist, wird vom Schritt 433 zum Schritt 435 übergegangen, um zu bestimmen, ob der Benutzer die Auswahltaste 28 gedrückt hat. Wenn die Auswahltaste nicht gedrückt wurde, kehrt das Programm zum Schritt 427 zurück (Figur 5b). Die Programmschritte zwischen 427 und 435 werden wiederholt durchlaufen, während der Cursor auf einem Menüblock positioniert ist, um das Fortschreiten des Cursors von Stelle zu Stelle zu verfolgen, um die notwendigen Unterlegungen zu erzeugen und um die Menü-Auswahlvariable stets zu erneuern, um die momentan aktive Menüstelle zu reflektieren.

Falls der Benutzer entscheidet, eine der Menüstellen innerhalb des Menüblocks auszuwählen, und weiterhin wünscht, keine weiteren Auswahlen zu tätigen, läßt der Benutzer die Menütaste 26 los, wenn der Cursor sich in der gewünschten Menübox befindet. Wenn dies auftritt, geht das Programm vom Schritt 431 zum Schritt 413 (Figur 5a), so daß der Menü-Zustand beendet wird und das Programm in den Menü-Erledigt-Zustand 500 übergeht (Figur 2).

Wenn der Benutzer jedoch wünscht, eine Menüstelle auszuwählen, ohne dabei die Menü-Session zu beenden, drückt der Benutzer die Auswahltaste 28, während er weiterhin die Menütaste gedrückt hält. In diesem Fall verzweigt das Programm vom Schritt 435 (Figur 5b) zum Schritt 437, bevor es zum Schritt 427 zurückkehrt, um den Cursor weiterhin in seinem Lauf durch den Menüblock zu verfolgen. Im Schritt 437 wird eine neue OptionList-Variable geschaffen und zur Liste addiert (Figur 3). Das neue OptionList-Wort wird codiert, um die Daten der momentanen MenüDrawn-Variable 32 und der Menüauswahl-Variable 34 zu reflektieren, welches die Menüstelle anzeigt, die von dem Cursor zum Zeitpunkt ders Tastendrucks angezeigt wird.

Im Falle, daß die Anzeige des aktiven Menüteils (des Menüteils, auf dem sich der Cursor momentan befindet) die Anzeige eines Submenü-Blocks erforderlich macht, verzweigt das Programm vom Schritt 445 zu den Schritten 447, 449, 451 und 453 (Figur 5c) und kehrt zum Schritt 427 zurück. Danach geht das Programm, bis sich der Cursor entweder auf eine neue Menüstelle bewegt oder einen Submenü-Block betritt, vom Schritt 427 zum Schritt 429, 431, 433, 453 und 427 in einer sich wiederholenden Schleife weiter, um das Fortschreiten des Cursors zu beobachten. Es sei angemerkt, daß das Programm den Benutzer nicht erlaubt, die Auswahltaste zu verwenden, um ein Menü auszuwählen, wenn dieser Menüteil mit einem Submenü assoziiert ist. Wie nachfolgend erklärt, wird in einem derartigen Fall eine gültige Auswahl nur dann erkannt, wenn sich der Cursor auf einem Submenü befindet.

Wenn der Cursor den Submenü-Block betritt, verzweigt das Programm beim Schritt 453 zum Schritt 455, wo das Programm sich weiter auf die Schritte 463, 465 und 467 verzweigt, da das Submenü-Feld der Menü-Auswahl-Variablen noch immer rückgesetzt

ist. Die Schritte sind ähnlich den Schritten 439, 441 und 443 und bewirken, daß der aktive Submenü-Teil aufscheint und daß die Menü-Auswahlvariable erneuert wird, um den aktiven Submenü-Teil zu reflektieren. Wenn der Benutzer einen Submenü-Teil auswählt, indem er die Menütaste groß läßt oder indem er die Auswahltaste drückt, so verzweigt das Programm (Figur 5c) im ersten Fall vom Schritt 457 zum Schritt 413 (5a), beendet die Menüsession und fährt mit dem Menü-Erledigt-Zustand fort.

Wenn ein Submenü durch Drücken der Auswahltaste 28 ausgewählt wird, verzweigt das Programm vom Schritt 459 (Figur 5c) zum Schritt 461, bevor es zum Schritt 453 zurückkehrt, um wieder das Weiterschreiten des Cursors durch den Submenüblock nachzuverfolgen. Im Schritt 461 wird ein neues OptionList-Wort erzeugt, um die akkumulierte Liste von Menü und Submenü-Auswahlen zu erweitern. Die neue OptionList-Variable wird codiert, um die Identität der Menü- und Submenü-Stelle zu reflektieren, die zu diesem Zeitpunkt des Tastendruckes aktiv sind.

Wenn der Benutzer den Cursor aus dem Submenü-Block herausbewegt und zurück zum Menüblock bewegt, ohne dabei eine Submenü-Stellenauswahl zu tätigen, verzweigt das Programm vom Schritt 453 zurück zum Schritt 427 (Figur 5b), wo dann die nächste Menüstelle unterlegt wird und als möglicher Auswahlkandidat präsentiert wird.

Spezielle Codes werden in der Menü-Auswahl-Variable 34 verwendet, um solche Menüstellen zu identifizieren, die mit Submenüs assoziiert sind. Dies gestattet dem System eine Fehlerbedingung in dem Fall zu erkennen, bei dem der Benutzer den Menüteil auswählt durch Loslassen der Menütaste, dabei jedoch keinen assoziierten Submenüteil auswählt.

Wenn der Benutzer den Cursor aus dem Menüblock herausbewegt und zurück in den Kopfblock bewegt, verzweigt das Programm vom Schritt 427 (Figur 5b) zurück zum Schritt 407 (Figur 5a), um das Fortschreiten des Cursors durch den Kopfblock nachzuverfolgen und um neue Menü- und Submenü-Blöcke als Kandidaten für die Auswahl zu präsentieren. Wenn der Benutzer den Cursor sowohl aus dem Kopfblock, dem Menüblock und dem Submenüblock herausbewegt, läuft das Programm in einer Schleife, die die Schritte 407, 409, 453 und 424 umfaßt, bis der Benutzer entweder die Menütaste losläßt oder den Cursor zurück in den Menübereich bewegt. Wenn die Taste 26 losgelassen wird, verzweigt das Programm über den Schritt 409 und 411 und fährt mit dem Menü-Erledigt-Zustand fort.

Figur 6 verdeutlicht die Software, die während des Menü-Erledigt-Zustandes 500 ausgeführt wird. Vom Startschritt 501 geht das Programm mit Schritt 503 weiter, bei dem die OptionList-Variable addiert wird und die MenüDrawn- und die MenüAuswahl-Variablen rückgesetzt werden. Das Zufügen der letzten OptionList-Variable schließt einen Schritt mit ein, der ähnlich den vorher beschriebenen Schritten 437 und 461 ist, wobei ein neues OptionList-Wort erzeugt wird, um den Status der MenüDrawn- und Menüaus-Auswahl-Variablen 32 und 34 zu refletieren, wobei dies zu dem Zeitpunkt erfolgt, zu dem die Menütaste losgelassen wird.

Somit wird in dem Fall, daß der Benutzer ein einzelnes Menü oder Submenü während der Menü-Session durch Positionierung des Cursors auf die ausgewählte Stelle und durch Loslassen der Menütaste auswählt, eine einzelne OptionList-Variable 36-1 erzeugt sobald der Menü-Erledigt-Zustand betreten wird. Im Falle daß der Benutzer eine oder mehrere Menü- oder Submenü-

Stellen während der Menüsession durch Betätigung der Auswahltaste ausgewählt hat und dann eine letzte Stelle auswählt durch geeignetes Positionieren des Cursors und Loslassen der Menütaste, tritt das System in den Menü-Erledigt-Zustand mit Ausnahme der letzten Variablen mit allen OptionList-Variablen, die in der Auswahlliste vorhanden sind, ein. Die letztlich ausgewählte Stelle wird den OptionList-Variablen während der Ausführung des Schrittes 503 zugfügt.

Danach löscht die Software im Schritt 505 alle angezeigten Blöcke und ersetzt sie durch die gepufferten Bilddaten, um das volle Bild wieder zu erhalten. Im Schritt 507 wird eine Selektionsnachricht kompiliert, welche alle existierenden OptionList-Variablen enthält. Im Schritt 509 wird die Auswahlnachricht an das Anwenderprogramm übertragen, um die Menüstellenauswahlen des Benutzers an das System für nachfolende Anwendungen weiterzugeben. Schließlich kehrt die Software zum Idle-Zustand zurück, um auf die nächste Menüauswahl zu warten.

Zusammenfassend ist die Arbeitsweise des Systems der vorliegenden Erfindung wie folgt.

Wenn der Benuzter die Menütaste 26 drückt, wird der Menükopfblock 40 (Figur 4) der vorher abwesend war, auf dem Bildschirm zur Anzeige gebracht. Durch Betätigung der Maus bewegt der Benutzer den Cursor 50 in den Kopfblock 40, und sobald der Cursor den Kopfblock betritt, wird ein Menüblock 42 angezeigt, jeweils abhängig davon, ob die Kopf-Unterabschnitte A bis D den Eintrittspunkt des Cursors repräsentieren. Der Benutzer kann den Cursor horizontal entlang des Kopfblockes bewegen und sobald ein neuer Unterabschnitt betritt wird, wird

ein neuer Menüblock angezeigt und der vorherige angezeigte Block gelöscht.

Wenn der Benutzer den Cursor in einen Menüblock bewegt, wird die Menübox, in der sich der Cursor befindet, unterlegt. Wenn ein Submenü mit der Menüstelle assoziiert ist, wird sie automatisch neben der Menüstelle als Submenü-Block 44 (Figur 4) angezeigt. Der Benutzer fährt fort den Cursor durch den Menüblock zu ziehen und die Unterlegung folgt der Position des Cursors. Wenn der Cursor aus einer Menübox mit der ein Submenü assoziiert ist, herausbewegt wird, wird das Submenü gelöscht. Wenn der Benutzer den Cursor in eine Submenü-Box bewegt, wird die Submenüstelle in der sich der Cursor befindet, unterlegt werden, wie dies der Fall bei der Menüstelle war.

Jedesmal dann wenn der Benutzer die Auswahltaste 28 drückt, wenn der Cursor sich in einem aktiven Menü oder Submenü befindet, wird ein OptionList-Wort erzeugt und zur Sammlung von Option-List-Variablen addiert, vorausgesetzt daß die selekte Stelle nicht bereits in der existierenden Auswahlliste vorhanden ist. Eine Ausnahme zu dieser OptionList-Variablen Erzeugung stellt der Fall dar, bei dem die Auswahltaste in einer aktiven Menüstelle gedrückt wird, mit der ein Submenü assoziiert ist. In diesem Fall wird keine OptionList-Variable erzeugt.

Der Benutzer kann mit der Bewegung des Cursors zu einer neuen Menü- und Submenü-Stelle fortfahren und derartige Stellen durch Drücken der Auswahltaste 28 auswählen (solang die Menütaste 26 sich in der gedrückten Position befindet) und die Liste der ausgewählten Stellen wird entsprechend erweitert. Wenn der Benutzer die letzte Stelle in einer Menüsession auswählt, läßt er die Menütaste los, worauf dann, wie vorher erklärt, in den

Menü-Erledigt-Zustand übergegangen wird und die letzte ausgewählte Stelle als letztes Option-List-Wort verwendet wird. Danach wird die Menü-Erledigt-Softwar (Figur 6) ausgeführt, um eine Auswahl nach sich zu kompilieren, wobei alle ausgewählten Stellen eingeschlossen werden und wobei die Nachricht an das Anwenderprogramm für die nachfolgende Ausführung weitergegeben wird.

Das System der vorliegenden Erfindung enthält das Drag-Auswahl-Merkmal, durch das der Benutzer mehrere Menü- und Submenü-Stellen einfach durch Halten sowohl der Menü- als auch der Auswahl-Taste während des Bewegens des Cursors durch eine Anzahl von Menü- und/oder -Submenü-Stellen auswählen kann. Jedes Mal, wenn der Cursor in eine neue Box bewegt wird, wird die Stelle automatisch ausgewählt und eine geeignete OptionList-Variable erzeugt und zur Liste addiert (falls nicht die Menüstelle mit einem Submenü assoziiert ist). Die Weise, mit der dies geschieht, ist beispielsweise in Figur 5b gezeigt. Wenn der Cursor eine neue Menübox betritt, verzweigt die Software über den Schritt 429 zu den Schritten 439, 441, 443, 445, 431, 433 und 435. Da die Auswahltaste in der gedrückten Position gehalten wird, wird mit Schritt 437 fortgefahren, um eine neue OptionList-Variable zu erzeugen. Solang der Cursor in dieser bestimmten Box verbleibt, geht das Programm in eine Schleife durch die Schritte 437, 429, 431, 433, 435 und 437 über. Eine andere OptionList-Variable wird jedoch nicht erzeugt, da das System keine redundanten Option-List-Variablen während der gleichen Menü-Session erzeugt.

Die gleiche "drag"-Auswahl Funktion wird im Falle der Submenü-Auswahl verwendet, wie dies in Figur 5c gezeigt ist. In diesem Fall wird sowohl die Menü- als auch Auswahl-Taste in der unteren Position gehalten und das Programm verzweigt zu den

Schritten 453 und 455 und fährt mit den Schritten 463, 455 und 467 fort, wenn der Cursor eine neue Submenübox betritt. Danach, wenn der Schritt 455 wieder betreten wird, verzweigt das Programm zu den Schritten 457 und 459 und betritt dann automatisch Schritt 461, da die Auswahltaste bereits gedrückt ist. Die Ausführung des letzten Schrittes erzeugt automatisch eine neue OptionList-Variable, um die automatische Auswahl der neu betretenen Submenü-Stelle zu reflektieren.

Wenn der Benuzter den Cursor außerhalb entweder des Menüblocks oder eines Busmenü-Blocks bewegt, so daß der Cursor sich nicht im Kopf, im Menü- oder Submenü-Block befindet, wird die Software vom Schritt 453 (Fig. 5c) zum Schritt 427 (Fig. 5b) und zum Schritt 407 (Fig. 5a) verzweigen. Danach wird durch den Schritt 409 zurück zum Schritt 453 (Fig. 5c) fortgefahren, wo die gleiche Schleife wieder betreten wird. Diese Schleife wird so lange wiederholt, bis der Benutzer entweder die Menütaste 26 ausläßt oder den Cursor zurück in einen der angezeigten Blöcke bewegt. Sobald einer der Blöcke wieder betreten wird, verzweigt die Software über die geeigneten Schritte 453, 427 oder 407, um die Anzeige zu erneuern und um zu ermöglichen, daß neue Kandidaten für die Auswahl präsentiert werden.

EP 87 904 781.9-2202
COMMODORE-AMIGA, INC.

Patentansprüche

1. Verfahren zur Verwendung in einem Datenverarbeitungssystem zum Anzeigen von benutzerwählbaren Menuepunkten und einem Curserbild, wobei das System eine Speichereinrichtung (20, 22) und eine benutzersteuerbare Curserpositioniereinrichtung (10) mit einer Auswahlsignalerzeugungseinrichtung, die einen Menuemodebetätiger enthält, aufweist und wobei das Verfahren die folgenden Schritte aufweist:

Detektiern der Aktivierung der Auswahlsignalerzeugungseinrichtung aufgrund eines Drückens des Menuemodebetätigers (26, 28);

Anzeigen eines Menueblocks (40, 42, 44), der Bereiche enthält, die eine Vielzahl von auswählbaren Menuepunkten bestimmen, in Antwort auf die Aktivierung der Auswahlsignalerzeugungseinrichtung;

Speichern eines Steuerwortes (32, 34, 36) in einem ersten Abschnitt der Speichereinrichtung, welches den Menueblockbereich repräsentiert, in dem das Curserbild positioniert ist, wobei das gespeicherte Steuerwort geändert wird, wenn das Curserbild bewegt wird, um den Menueblockbereich anzugeben, in dem das Curserbild momentan positioniert ist; und

Kompilieren einer Auswahlnachricht in Antwort auf die Erzeugung eines Auswahlsignals, wie es von der Auswahlsignalerzeugungseinrichtung erzeugt worden ist, wobei die Nachricht das Steuerwort enthält, das sich zum Zeitpunkt der Erzeugung

des Auswahlsignals in dem ersten Speicherbereich befindet (506, 509).

2. Verfahren zur Verwendung in einem Datenverarbeitungssystem zum Anzeigen benutzerwählbarer Menuepunkte und einem Curserbild, wobei das System eine Speichereinrichtung (20, 22) und eine benutzersteuerbare Curserpositioniereinrichtung (10) mit einer Auswahlsignalerzeugungseinrichtung, die einen Menuemodebetätiger umfaßt, aufweist und die in der Lage ist, erste und zweite Auswahlsignale zu erzeugen und wobei das Verfahren die folgenden Schritte aufweist:

Detektieren der Aktivierung der Auswahlsignalerzeugungseinrichtung in Antwort auf ein Drücken des Menuemodebetätigers (26, 28);

Anzeigen eines Menueblocks (40, 42, 44), welcher Bereiche enthält, die eine Vielzahl auswählbarer Menuepunkte bestimmen, in Antwort auf die Aktivierung der Auswahlsignalerzeugungseinrichtung;

Speichern eines Steuerwortes (32, 34, 36) in einem ersten Abschnitt der Speichereinrichtung, welches den Menueblockbereich repräsentiert, in dem sich das Curserbild befindet, wobei das gespeicherte Steuerwort geändert wird, wenn das Curserbild bewegt wird, um den Menueblockbereich anzugeben, in dem das Curserbild momentan positioniert ist;

Transferieren einer Vielzahl von unterschiedlichen Steuerwörtern (36) von dem ersten Abschnitt in einen zweiten Abschnitt der Speichereinrichtung, wobei jedes der mehreren Steuerwörter in Antwort auf die Erzeugung eines ersten Auswahlsignals transferiert wird und Kompilieren einer Auswahlnachricht in Antwort auf die Erzeugung des zweiten Auswahlsignals, wobei die Nachricht die Vielzahl von Steuerwörtern enthält, die zum

Zeitpunkt der Erzeugung des zweiten Auswahlsignals in dem zweiten Speicherabschnitt enthalten sind.

3. Verfahren nach Anspruch 1 oder 2, bei dem der Menueblock einen Menuekopfblock (40) enthält.

4. Verfahren nach einem der Ansprüche 1, 2 oder 3, weiterhin enthaltend den Schritt des Speicherns eines Steuerwortes mit einem Nullwert, wenn das Curserbild außerhalb der Grenzen des Menueblocks positioniert ist in dem ersten Speicherabschnitt in Antwort auf die Aktivierung der Auswahlsignalerzeugungseinrichtung (10), woraufhin die Kompilierung der Nachricht mit dem Nullwert-Steuerwort das Fehlen einer Menuepunktauswahl anzeigt.

5. Verfahren nach einem der Ansprüche 2, 3 oder 4, bei dem die Auswahlsignalerzeugungseinrichtung erste und zweite benutzersteuerbare Auswahlknöpfe (26, 28) aufweist und bei dem die Erzeugung des ersten Auswahlsignals die Detektion, daß der zweite Knopf (28) gedrückt worden ist, während der erste Knopf (26) in einer gedrückten Position gehalten worden ist, erfordert und bei dem die Erzeugung des zweiten Auswahlsignals die Detektion erfordert, daß der erste Knopf losgelassen worden ist.

6. Verfahren nach Anspruch 5, weiterhin enthaltend den Schritt der Erzeugung eines ersten Auswahlsignals in Antwort auf die Änderung des Steurwortes, wie es in dem ersten Speicherabschnitt gespeichert ist, zu einem Zeitpunkt, zu dem beide Auswahlknöpfe gedrückt sind.

FIG. 6

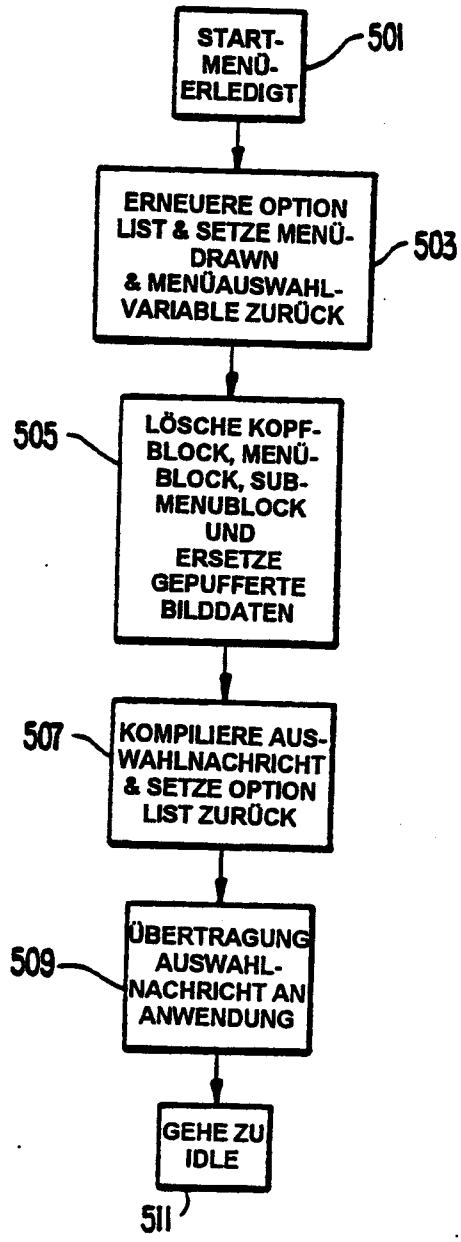

START-MENÜ-ERLEDIGT
501
ERNEUERE OPTION LIST & SETZE MENÜ-DRAWN & MENÜAUSWAHL-VARIABLE ZURÜCK
503
505
LÖSCHE KOPF-BLOCK, MENÜ-BLOCK, SUB-MENUBLOCK UND ERSETZE GEPUFFERTE BILDDATEN
507
KOMPILIERE AUS-WAHLNACHRICHT & SETZE OPTION LIST ZURÜCK
509
ÜBERTRAGUNG AUSWAHL-NACHRICHT AN ANWENDUNG
GEHE ZU IDLE
511

FIG. 1.

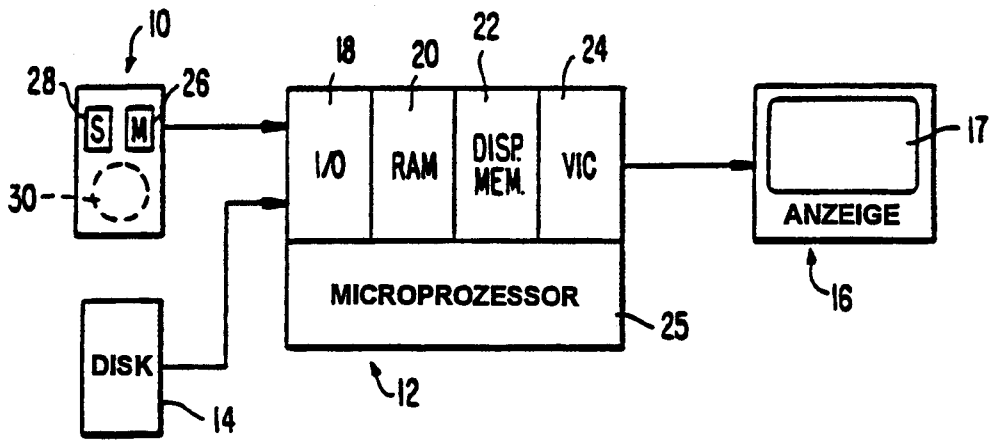

10
28
26
S
M
30
18
20
22
24
I/O
RAM
DISP. MEM.
VIC
MICROPROZESSOR
25
12
DISK
14
ANZEIGE
17
16

FIG. 2.

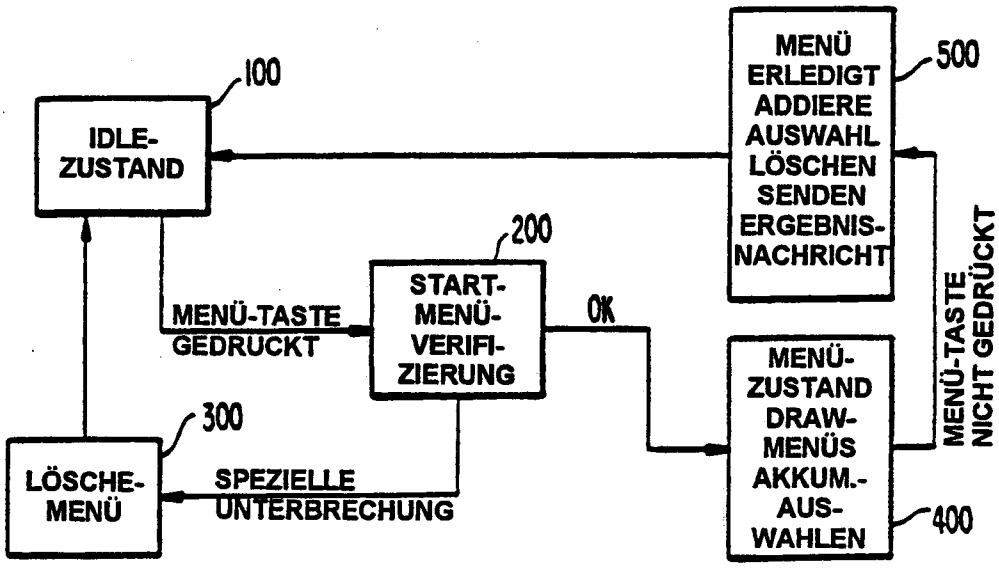

100
IDLE-ZUSTAND
MENÜ-TASTE GEDRÜCKT
200
START-MENÜ-VERIFI-ZIERUNG
OK
300
LÖSCHE-MENÜ
SPEZIELLE UNTERBRECHUNG
MENÜ ERLEDIGT ADDIERE AUSWAHL LÖSCHEN SENDEN ERGEBNIS-NACHRICHT
500
MENÜ-ZUSTAND DRAW-MENÜS AKKUM.-AUS-WAHLEN
400
MENÜ-TASTE NICHT GEDRÜCKT

FIG. 3

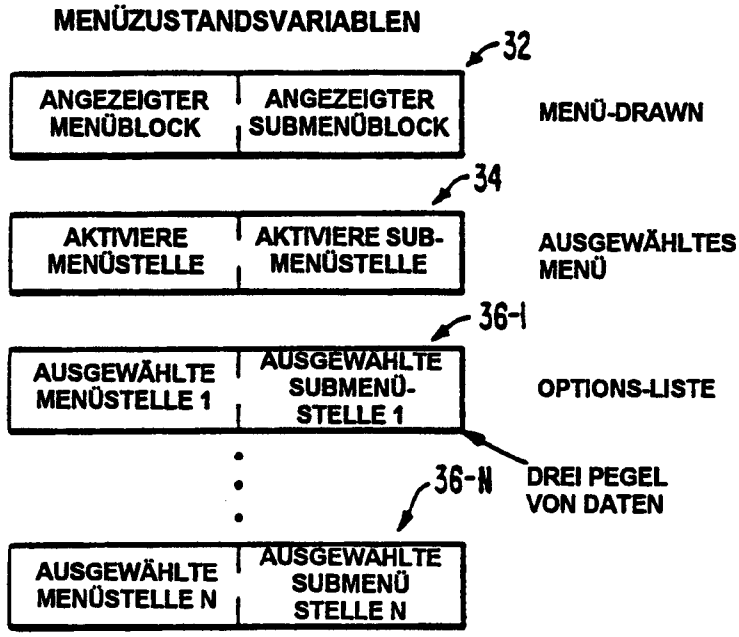

MENÜZUSTANDSVARIABLEN
32
ANGEZEIGTER MENÜBLOCK
ANGEZEIGTER SUBMENÜBLOCK
MENÜ-DRAWN
34
AKTIVIERE MENÜSTELLE
AKTIVIERE SUB-MENÜSTELLE
AUSGEWÄHLTES MENÜ
36-1
AUSGEWÄHLTE MENÜSTELLE 1
AUSGEWÄHLTE SUBMENÜ-STELLE 1
OPTIONS-LISTE
36-N
DREI PEGEL VON DATEN
AUSGEWÄHLTE MENÜSTELLE N
AUSGEWÄHLTE SUBMENÜ STELLE N

FIG. 4

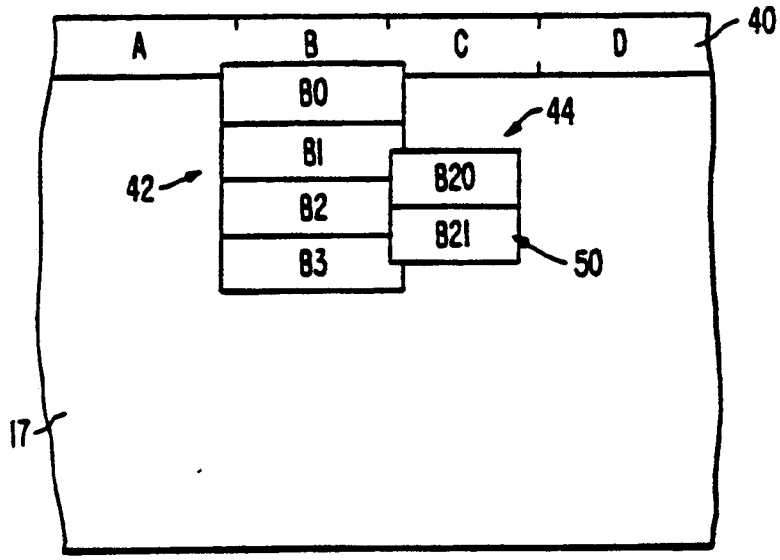

40
A
B
C
D
B0
B1
B2
B3
42
44
B20
B21
50
17

FIG. 5a

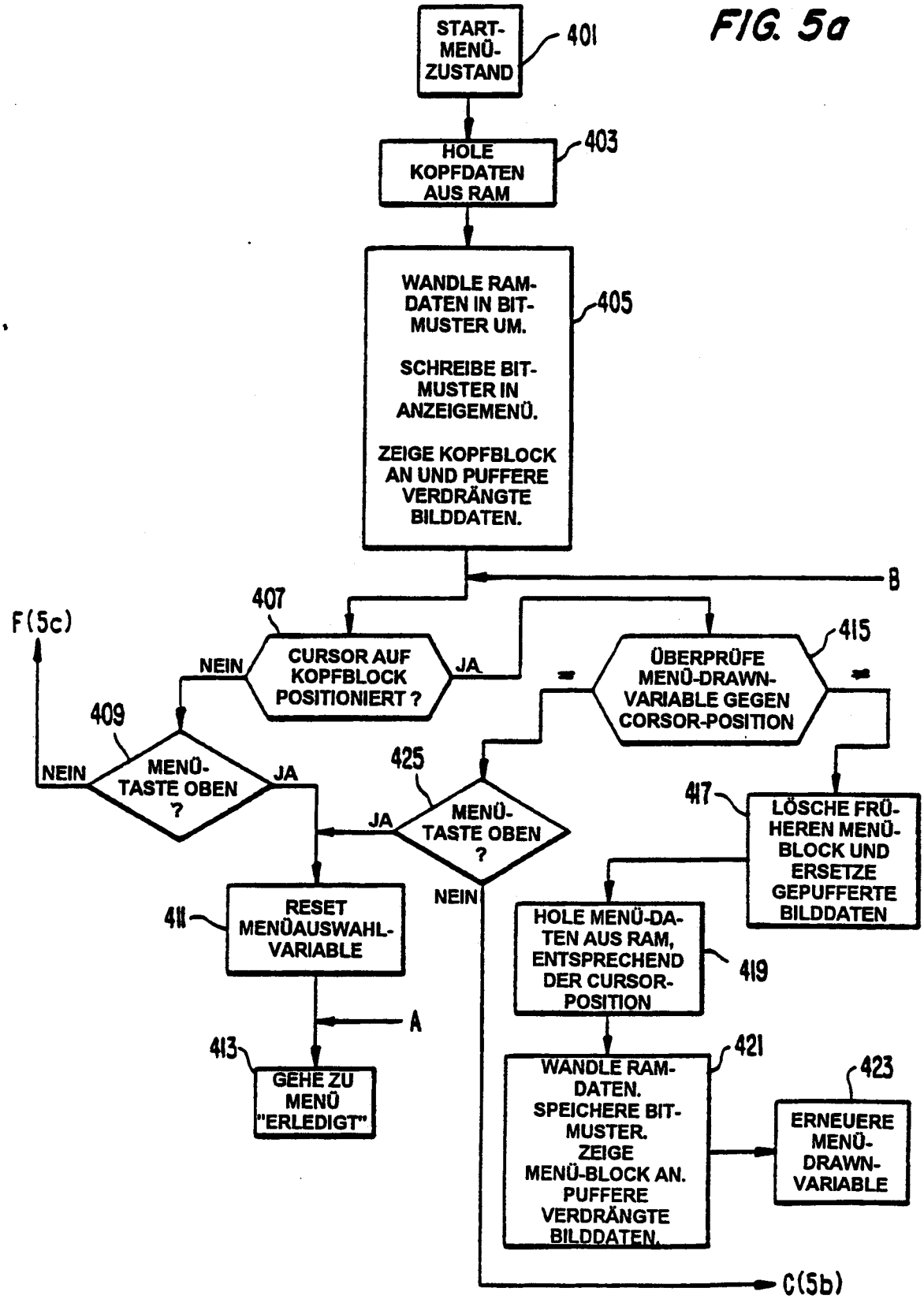

START-MENÜ-ZUSTAND
401
HOLE KOPFDATEN AUS RAM
403
WANDLE RAM-DATEN IN BIT-MUSTER UM.
SCHREIBE BIT-MUSTER IN ANZEIGEMENÜ.
ZEIGE KOPFBLOCK AN UND PUFFERE VERDRÄNGTE BILDDATEN.
405
B
407
CURSOR AUF KOPFBLOCK POSITIONIERT ?
NEIN
JA
F(5c)
409
MENÜ-TASTE OBEN ?
NEIN
JA
ÜBERPRÜFE MENÜ-DRAWN-VARIABLE GEGEN CORSOR-POSITION
415
425
MENÜ-TASTE OBEN ?
JA
NEIN
417
LÖSCHE FRÜHEREN MENÜ-BLOCK UND ERSETZE GEPUFFERTE BILDDATEN
411
RESET MENÜAUSWAHL-VARIABLE
A
413
GEHE ZU MENÜ "ERLEDIGT"
HOLE MENÜ-DATEN AUS RAM, ENTSPRECHEND DER CURSOR-POSITION
419
WANDLE RAM-DATEN. SPEICHERE BIT-MUSTER. ZEIGE MENÜ-BLOCK AN. PUFFERE VERDRÄNGTE BILDDATEN.
421
ERNEUERE MENÜ-DRAWN-VARIABLE
423
C(5b)

FIG. 5b

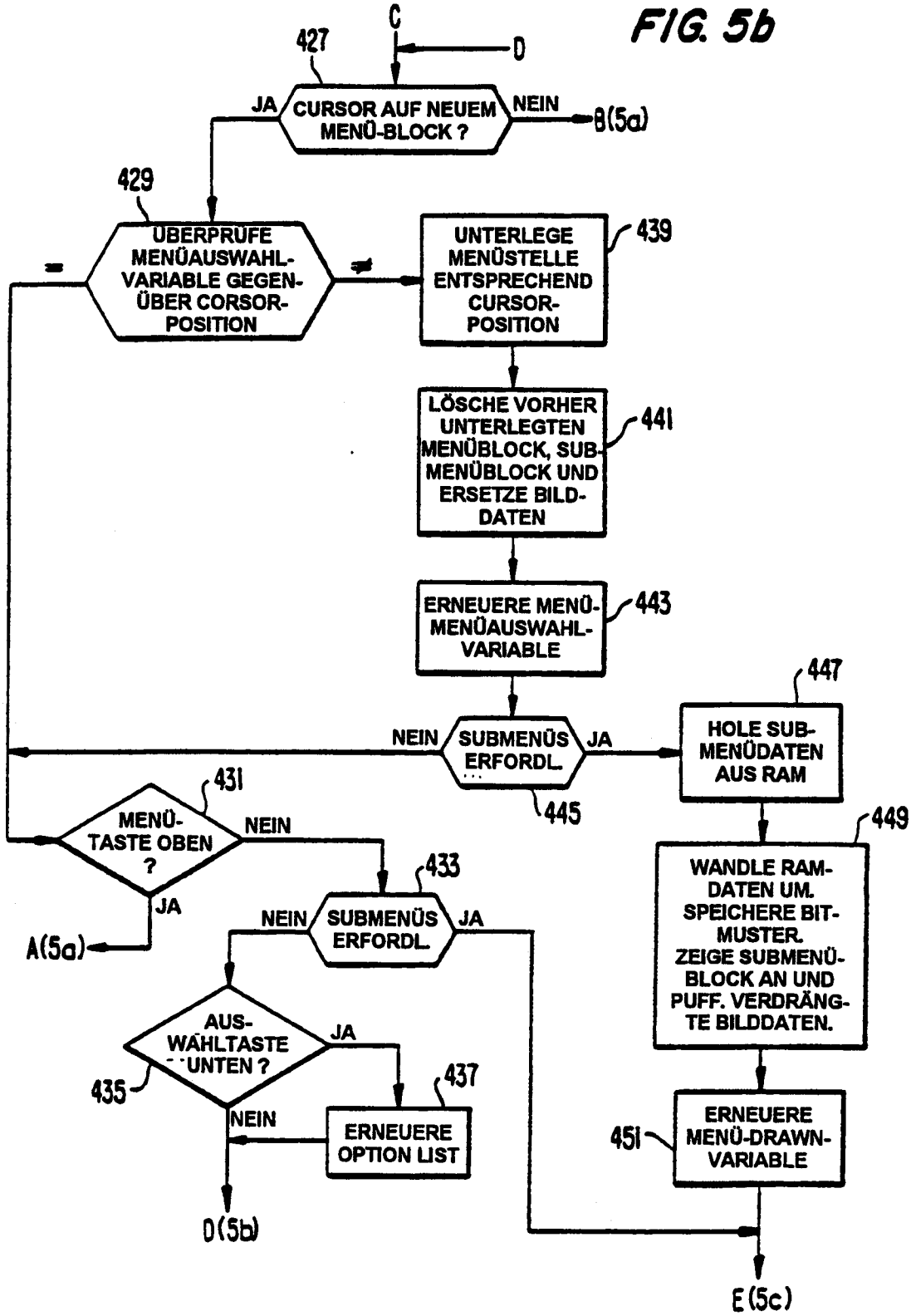
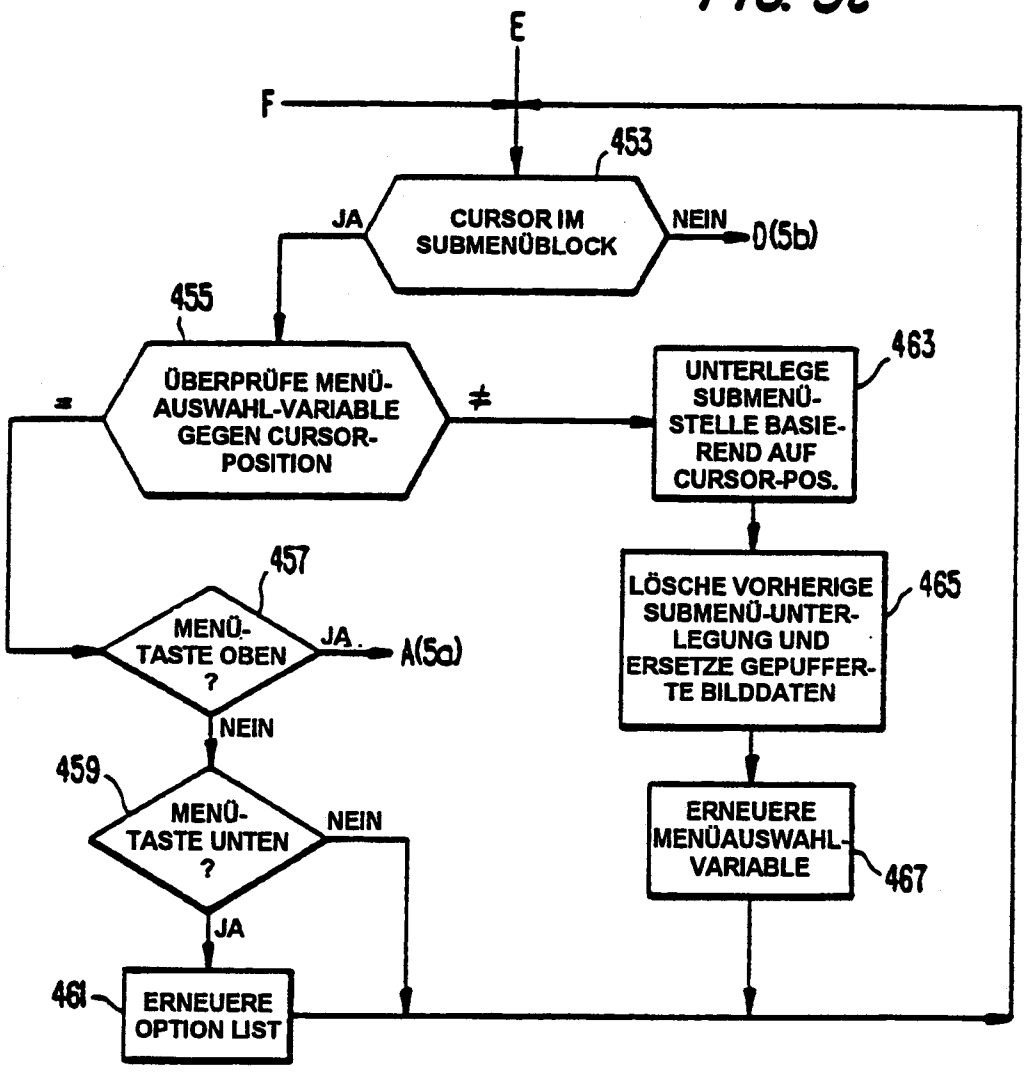

C
D
427
CURSOR AUF NEUEM MENÜ-BLOCK ?
JA
NEIN
B(5a)
429
ÜBERPRÜFE MENÜAUSWAHL-VARIABLE GEGEN-ÜBER CORSOR-POSITION
=
≠
UNTERLEGE MENÜSTELLE ENTSPRECHEND CURSOR-POSITION
439
LÖSCHE VORHER UNTERLEGTEN MENÜBLOCK, SUB-MENÜBLOCK UND ERSETZE BILD-DATEN
441
ERNEUERE MENÜ-MENÜAUSWAHL-VARIABLE
443
447
NEIN
SUBMENÜS ERFORDL.
JA
HOLE SUB-MENÜDATEN AUS RAM
445
431
MENÜ-TASTE OBEN ?
NEIN
JA
A(5a)
449
WANDLE RAM-DATEN UM. SPEICHERE BIT-MUSTER. ZEIGE SUBMENÜ-BLOCK AN UND PUFF. VERDRÄNG-TE BILDDATEN.
433
NEIN
SUBMENÜS ERFORDL.
JA
AUS-WÄHLTASTE UNTEN ?
JA
435
437
NEIN
ERNEUERE OPTION LIST
451
ERNEUERE MENÜ-DRAWN-VARIABLE
D(5b)
E(5c)

FIG. 5c


E
F
453
CURSOR IM SUBMENÜBLOCK
JA
NEIN
D(5b)
455
ÜBERPRÜFE MENÜ-AUSWAHL-VARIABLE GEGEN CURSOR-POSITION
=
≠
463
UNTERLEGE SUBMENÜ-STELLE BASIE-REND AUF CURSOR-POS.
457
MENÜ-TASTE OBEN ?
JA
A(5a)
NEIN
465
LÖSCHE VORHERIGE SUBMENÜ-UNTER-LEGUNG UND ERSETZE GEPUFFER-TE BILDDATEN
459
MENÜ-TASTE UNTEN ?
NEIN
JA
ERNEUERE MENÜAUSWAHL-VARIABLE
467
461
ERNEUERE OPTION LIST