

(12)

# PATENTCHRIFT

(21) Anmeldenummer: 2453/85

(51) Int.Cl.<sup>5</sup> : G06F 9/30

(22) Anmeldetag: 22. 8.1985

(42) Beginn der Patentdauer: 15. 8.1989

(45) Ausgabetag: 26. 3.1990

(56) Entgegenhaltungen:

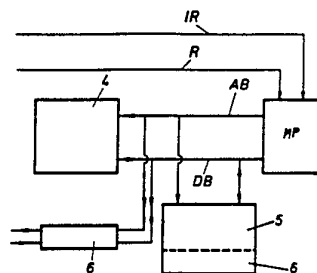
DE-OS2810168 DE-OS3321910 EP-A2 0137995 EP-A2 0154252  
EP-A2 0152900

(73) Patentinhaber:

NOVOMATIC AUTOMATENINDUSTRIE- UND  
-HANDELSGESELLSCHAFT M.B.H. & CO.KG.  
A-2352 GUMPOLDSKIRCHEN, NIEDERÖSTERREICH (AT).

(54) ANORDNUNG ZUM ENTSCHLÜSSELN VON BEFEHLEN

(57) Die Erfindung betrifft eine Anordnung zum Entschlüsseln von in zumindest einem Befehls- und/oder Datenspeicher abgelegten Befehlen und/oder Daten, welche einem Mikroprozessor über einen Umkodierer zur Verarbeitung zugeleitet sind und wobei im Umkodierer Befehle und/oder Daten durch im Umkodierer vorgegebene oder erstellte Befehle und/oder Daten ersetzt bzw. in anders lautende Befehle und/oder Daten umgewandelt werden, die sodann dem Mikroprozessor zugeführt werden. Erfindungsgemäß ist vorgesehen, daß an einen Eingang des Umkodierers (2) der Ausgang einer Steuerschaltung (7) angeschlossen ist, daß der Eingang der Steuerschaltung (7) an die Datenleitung zwischen dem Umkodierer (2) und dem Mikroprozessor (MP) angeschlossen ist und daß die Art der Umkodierung des Umkodierers (2) von der Steuerschaltung (7) in Abhängigkeit von zumindest einem der vorangehend über die Datenleitung dem Mikroprozessor zur Ausführung bzw. Abarbeitung im Zuge des Programmverlaufes zugeführten Befehls- und/oder Datenbyte bestimmt ist.



Die Erfindung betrifft eine Anordnung zum Entschlüsseln von in zumindest einem Befehls- und/oder Datenspeicher abgelegten Befehlen und/oder Daten, wobei die Befehle und/oder Daten auf Abfrage aus den genannten Speichern einem Mikroprozessor über einen gegebenenfalls mit diesem hybridisierten bzw. vergossenen Umkodierer zur Verarbeitung zugeleitet sind und wobei im Umkodierer zumindest ein Teil der dem(n) Speicher(n) entnommenen Befehle und/oder Daten durch im Umkodierer vorgegebene oder erstellte Befehle und/oder Daten ersetzt bzw. in anders lautende Befehle und/oder Daten umgewandelt wird, die sodann dem Mikroprozessor als gültige Befehle und/oder Daten über eine vom Ausgang des Umkodierers zum Befehls- bzw. Dateneingang des Mikroprozessors führende Datenleitung zugeführt sind.

Im folgenden werden bekannte Anordnungen zum Schutz von Software erläutert, zu den unter anderem auch die aus der DE-OS 2 810 168, DE-OS 3321910, EP-A2 137 995, EP-A2 154 252 und EP-A2 152 900 bekannten Anordnungen zählen.

Die Funktion von Mikroprozessorsystemen nach Fig. 1 ist im Prinzip folgende:

Durch ein Signal auf der Reset-Leitung (R) wird ein Mikroprozessor (MP) in einen definierten Anfangszustand gesetzt. Er beginnt daraufhin mit der Abarbeitung des in einem Speicher (4), (5) abgelegten Programmes an einer ganz bestimmten, vom Hersteller festgelegten Adresse. Ein handelsüblicher MP Z80 beispielsweise beginnt immer bei der Adresse 0000. Das Programm steht nicht im Klartext in den Speichern (4), (5), sondern in Form von Zahlen bestimmter Bedeutung, d. h. es ist jedem für den Mikroprozessor definierten Befehl eine Zahl bzw. Zahlenfolge zugeordnet. Welche Befehle definiert sind und wie die zugehörige Zahl lautet, ist für alle Mikroprozessoren gleichen Typs gleich und dem vom Hersteller herausgegebenen Datenblatt zu entnehmen.

Programme können dabei in Lesespeichern (4) (ROM; PROM, EPROM,...) und/oder in Schreib-/Lesespeichern (5) (RAM) abgelegt sein, wogegen der Mikroprozessor nur in Schreib-/Lesespeichern (5) Daten ablegen kann. Die Eingabe und Ausgabe der verarbeiteten Daten erfolgt über den Eingabe-Ausgabe Block (6).

Der Mikroprozessor legt also die Adresse 0000 auf den Adreßbus (AB). Mit dieser Adresse wird ein bestimmtes Byte in einem bestimmten Speicherbaustein angesprochen, das über den Datenbus (DB) zum Mikroprozessor übertragen wird. Im Programmbeispiel hat dieses Byte auf Adresse 0000 den Wert 3E (Alle Adressen und Daten bzw. Befehle werden hier nicht im Dezimal- sondern im Hexadezimalsystem dargestellt; z. B. entspricht Hexadezimal 3E der Dezimalzahl 62). Die Zahl 3E bedeutet für den Z80-MP "LD A,..." (Lade das Register (A) mit dem unter der unmittelbar folgenden Adresse abgelegten Byte). Der Mikroprozessor muß also zur vollständigen Ausführung des Befehls noch die Adresse 0001 auf den Adreßbus (AB) legen und das über den Datenbus übertragene Byte in das Register (A) laden. In unserem Beispiel hat dieses nachfolgende Byte den Wert 10, d. h., der vollständige Befehl lautet "LD A, 10" bzw. "3E 10".

Nun ist die Ausführung des ersten Befehls abgeschlossen und es folgt der zweite Befehl.

Dieser wird von der nächstfolgenden Adresse geladen, da der Mikroprozessor das Programm Schritt für Schritt durchführt, wobei die nacheinander durchzuführenden Befehle in aufeinanderfolgenden Speicherplätzen abgelegt sind.

Der Mikroprozessor legt also die Adresse 00002 auf den Adreßbus. Über den Datenbus (DB) erhält er das Byte 47, welches für den Z80 "LD B, A" (Lade den Inhalt des Registers (A) in das Register (B)) bedeutet. Nach Durchführung dieses Befehls wird die nächste Adresse, 0003, auf den Adreßbus gelegt und der Befehl, in unserem Beispiel 47 ("LD B,A"), durchgeführt.

Die Programmdurchführung muß nicht immer aus aufeinanderfolgenden Speicherzellen erfolgen. Es gibt da Ausnahmen, die sich im wesentlichen auf zwei reduzieren lassen:

Die erste Ausnahme sind sogenannte Sprünge (bedingt, unbedingt, Unterprogrammaufruf, Unterprogrammrückprung). So kann im Programm der Code C3 auftreten, der für den Z80 "JMP..." (Fahre mit der Programmabarbeitung an der durch die beiden folgenden Bytes adressierten Stelle fort; jump to adress...) bedeutet. Steht also z. B. der Code C3 12 34 im Speicher, so heißt das "JMP 3412" (es wird erst die niederwertige und dann die höherwertige Adreßhälfte übertragen), also springe mit der Programmausführung zur Adresse 3412.

Die zweite Ausnahme sind sogenannte Interrupts. So gibt es bei fast jedem Mikroprozessor die Möglichkeit, wenigstens ein Programm über einen Interrupt ausführen zu lassen. Ein Interrupt ist ein Hardware-Signal, welches den Mikroprozessor dazu veranlaßt, die Ausführung des momentanen Programms sofort zu unterbrechen und ein vom Mikroprozessor-Anwender geschriebenes, an einer vom Mikroprozessor-Hersteller festgelegten Adresse im Programmspeicher abgelegtes Programm ("Interruptroutine") durchzuführen, wobei es wesentlich ist, daß sich der Mikroprozessor die Stelle der Unterbrechung merkt und nach Beendigung des Interruptprogrammes mit der Ausführung eben dort fortfährt. Der Sinn einer solchen Möglichkeit ist die schnelle Reaktion auf externen Ereignisse oder Synchronisation mit externen Ereignissen.

Vergleichbar ist der Mikroprozessor mit Interruptmöglichkeit mit einem Menschen der arbeitet (Programm) und sobald das Telefon läutet, seine Arbeit unterbricht, den Hörer abhebt, spricht, ... schließlich den Hörer wieder auflegt (Ende des Interruptprogramms), um die Arbeit an der Unterbrechungsstelle wieder aufzunehmen. Ein Mensch ohne Interruptmöglichkeit müßte immer wieder nachschauen, ob das Telefon läutet, was zwei Folgen haben kann: Entweder der Anrufer muß lange warten, bis der Mensch wieder einmal nachschaut, oder der Mensch muß so oft nachschauen, daß seine Arbeit wesentlich langsamer vonstatten geht. Beim vorliegenden

Mikroprozessor (Z80) löst eine Art von Interrupt einen Sprung zur Speicherzelle 0066 aus, die Unterbrechungsadresse (Rücksprung-) wird im Schreib-/Lesespeicher (RAM) für den späteren Rücksprung ins Hauptprogramm abgelegt.

Es sind zu der Schaltung in Fig. 1 am Ende der Beschreibung zwei Programmbeispiele angegeben, die keine sinnvollen Programme darstellen, sondern nur den Daten-, Befehls- und Adressfluß darstellen sollen. Das zweite Programmbeispiel könnte statt dem ersten im Speicher stehen und unterscheidet sich nur durch einen Befehl (6F) (= ein Byte) von ersten Beispiel.

Die Schaltung nach Fig. 1 ist nun verständlich. Dabei sind der Mikroprozessor und die Speicherbauelemente Standardbausteine, die jeder kaufen kann. Die Befehlskodes und deren Bedeutung sind bekannt und im Datenblatt des jeweiligen Mikroprozessors nachzulesen. Aus diesem Grund ist die Software von jedem kopierbar, insbesondere wenn bestehende Geräte mit neuer Software ausgerüstet werden sollen: man beschafft sich einen Speicher (EPROM) mit neuer Software und kopiert sie mit einem handelsüblichen Programmiergerät in weitere EPROMS, die dann nur noch in die anderen Geräte gesteckt zu werden brauchen.

Der Schaden für den Software-Erzeuger ist beträchtlich.

Es ist aus diesem Grund wünschenswert, die Software so zu verschlüsseln, daß sie nicht mehr auf einem Standard-Mikroprozessor lauffähig ist.

Zu diesem Zweck werden die Befehlskodes vor der Programmierung des EPROMS verschlüsselt, d. h. es wird jedem Befehlscode ein bestimmter anderer Kode zugeordnet. Während des Programmablaufes wird dann jeder Kode wieder entschlüsselt, wobei der dazu notwendige Umkodierer (2) vorzugsweise mit dem Mikroprozessor hybridisiert ist, da er sonst leicht analysierbar ist und nachgebaut werden kann (Fig. 2).

Der Software-Erzeuger verkauft nun nicht nur den Speicher mit dem Programm sondern zusätzlich den Mikroprozessor mit Umkodierer (2).

Der Programmablauf ist wie folgt, wobei das selbe Programm durchgeführt werden soll wie bei der Schaltung nach Fig. 1.

Nach dem Reset über (R) legt der Mikroprozessor die Adresse 0000 auf den Adreßbus, über den Datenbus kommt ein verschlüsseltes Byte, in vorliegendem Fall 57, zum Umkodierer (2). Dieser wandelt - nach einem bestimmten Algorithmus - den Wert 57 in 3E um. Das Byte 3E wiederum gelangt jetzt zum Mikroprozessor, der es wie vorher als "LD A,..." interpretiert und ausführt. Zu diesem Zweck legt er die Adresse 001 auf den Adreßbus, das Byte 10 wird übertragen und kommt unverändert in den Mikroprozessor (Daten werden im vorliegenden Fall nicht verschlüsselt, da dies die Wirksamkeit der Verschlüsselung erhöht). Als nächstes wird die Adresse 0002 auf den Adreßbus gelegt. Das Byte 33 kommt zum Umkodierer (2), wird dort in 47 umgewandelt und gelangt so zum Mikroprozessor. Anschließend wird von Adresse 0003 der nächste Befehl in gleicher Weise gelesen, entschlüsselt und ausgeführt, usw.

Daten, die in den Schreib-/Lesespeicher oder zur Ausgabe gelangen sollen, werden um den Umkodierer (2) herumgeführt (Datenbus (1)) und unverschlüsselt abgelegt.

Der Interrupt (IR) und das Reset-Signal (R) funktionieren wie in der Schaltung gemäß Fig. 1.

Das zweite Programmbeispiel läuft wie das erste ab (es könnte an seiner Stelle im Speicher stehen), nur ein Befehl ist anders, nämlich auf Adresse 0004 steht das Byte 72, welches vom Umkodierer (2) in 6F umgewandelt wird, was wiederum der Mikroprozessor als "LD L, A" (Lade Inhalt von Register (A) in Register (L)) versteht.

Der Schutz durch Methoden dieser Art ist um so besser je schwieriger es ist, die Art der Kodierung zu analysieren und den Umkodierer nachzubilden.

Die oben genannte Methode ist mit einfachen Mitteln relativ leicht zu entschlüsseln, da sich jeder Mikroprozessor durch seine Hardware-Reaktion selbst verrät. So ist es durchaus möglich, einen verschlüsselten Z80-Mikroprozessor innerhalb von 30 Stunden zu analysieren und nachzubilden.

Eine Verbesserung des Schutzes bietet die Methode nach Fig. 3. In dieser Schaltung enthält der Umkodierer (2) mehrere (im vorliegenden Beispiel drei) Umkodierer gemäß Fig. 2, wobei diese Umkodierer bestimmten Adressen zugeordnet sind. Es erfolgt also die Umkodierung nicht nur in Abhängigkeit vom Byte, das an einer bestimmten Stelle gefunden wird sondern auch von der Adresse, auf die zugegriffen wird, selbst. Im Programmbeispiel sind drei Umkodierer-"Seiten" bzw. "Listen" vorhanden, die den verschiedenen Speicherplätzen bzw. Adressen wie folgt zugeordnet sind:

Seite 1	0000	0004	0008	000C	0010	0014	0018	....
Seite 2	0001	0005	0009	000D	0011	0015	0019	....
Seite 3	0002	0006	000A	000E	0012	0016	001A	....
Seite 2	0003	0007	000B	000F	0013	0017	001B	....

Wenn man also die Adressen der Reihe nach durchgeht, sind die zugehörigen Seiten 1,2,3,2, 1,2,3,2, 1,2,3,2, ....

Reset-Leitung (R) und Interrupt (IR) funktionieren wie zuvor beschrieben.

Nach dem Reset legt der Mikroprozessor die Adresse 0000 auf den Adreßbus; aus dem Programmspeicher wird über den Datenbus (DB) der Wert 57 übertragen, welcher vom Umkodierer (2) (ADR. 0000 >>> Seite 1) in 3E umgewandelt wird, welche über einen zusätzlichen Bus (9) mit entsprechenden Informationen versorgt

wird, was wiederum vom Z80 richtig als "LD A, .." ausgeführt wird. Zu diesem Zweck wird wieder die Adresse 0001 auf den Adreßbus gelegt und das Byte 10 gelangt unverändert zum Mikroprozessor. Nun wird die Adresse 0002 (>>> Seite 3) auf den Adreßbus gelegt, das Byte 33 wird gelesen und in 47 umkodiert, was der Mikroprozessor als "LD B, A" erkennt. Als nächstes folgt die Adresse 0003 (>>> Seite 2), von der das Byte 12 gelesen wird, dieses wird umkodiert in 47; der Mikroprozessor erhält also wieder den Befehl "LD B,A". Es wird also auf Adr. 0002 das Byte 33 in 47 umgedreht, während auf Adr. 0003 das Byte 12 in 47 umgewandelt wird, aber wobei ein Byte 12, das auf Adr. 0002 aufscheinen würde, eine ganz andere Umwandlung erfahren hätte.

Damit ist erreicht, daß man aus der Kenntnis der Verschlüsselung von 47 ("LD B,A") auf Adresse 0002 noch nicht auf die Verschlüsselung von 47 auf Adresse 0003 schließen kann, was es einem Uneingeweihten erschwert, Programme "auszuhorchen", abzuändern oder selbst zu schreiben.

Der Nachteil ist, daß bei stark steigendem Aufwand des Umkodierers der Analyse-Aufwand relativ wenig steigt.

Auch in dieser Schaltung werden die Daten zu RAM (4) und Ausgabe (6) mit dem Datenbus (1) um den Umkodierer herumgeführt.

Die beiden zuvor anhand Fig. 2 und 3 beschriebenen Schutzmethoden sind wohl bekannt.

Einige neue erfindungsgemäße Vorgangsweisen bzw. Anordnungen, die wesentlich schwieriger zu analysieren sind, sollen nun beschrieben werden. Diesen Anordnungen ist allen gemeinsam, daß die Funktion des Umkodierers nicht bzw. nicht nur von der Adresse und/oder dem übertragenen Byte abhängt, sondern auch von einem oder mehreren vorher ausgeführten bzw. dem Mikroprozessor zugeführten gültigen und von ihm verarbeitbaren Befehlen. Das bedeutet, daß es nicht genügt, ein Byte isoliert an einer bestimmten Stelle zu betrachten, sondern man muß - um ein Programm oder einen Mikroprozessor analysieren zu können - wissen, wie der Programmablauf davor erfolgt ist; der Umkodierer braucht also eine Art Gedächtnis für seine richtige Funktion.

Erfindungsgemäß ist eine Anordnung der eingangs genannten Art dadurch gekennzeichnet, daß an einen Eingang des Umkodierers der Ausgang einer Steuerschaltung, z. B. eines Zwischenspeichers in Form zumindest eines Flip-Flops, angeschlossen ist, daß der Eingang der Steuerschaltung an die Datenleitung zwischen dem Umkodierer und dem Mikroprozessor angeschlossen ist und daß die Art der Umkodierung des Umkodierers von der Steuerschaltung in Abhängigkeit von zumindest einem der vorangehend über die Datenleitung dem Mikroprozessor zur Ausführung bzw. Abarbeitung im Zuge des Programmlaufes zugeführten Befehls- und/oder Datenbyte bestimmt ist. Durch diese Anordnung wird die Möglichkeit geboten, daß gleiche Befehle verschiedene Verschlüsselung aufweisen können bzw. gleichlautende vom Speicher eintreffende Befehle (Daten) vom Umkodierer verschieden entschlüsselt bzw. umgewandelt werden, weil die Art der Umkodierung eben von vorangehenden (meist unterschiedlichen) Befehlen bestimmt wird.

Bei einer bevorzugten Ausführungsform der Erfindung ist vorgesehen, daß ein Eingang der Steuerschaltung mit dem Ausgang einer Erkennungsschaltung, z. B. einem Vergleichler, für die dem Mikroprozessor zugeführten Befehle und/oder Daten verbunden ist und daß der Steuerschaltung von der Erkennungsschaltung bei Feststellung einer Übereinstimmung zwischen zumindest einem auf der Datenleitung eingelangten Befehls- und/oder Datenbytes mit zumindest einem in der Erkennungsschaltung enthaltenen vorgegebenen Befehls- und/oder Datenbyte ein Signal zur Umschaltung des Umkodierers bzw. zur Abänderung der Umkodierung einer bestimmten Anzahl nachfolgend einlangender Befehle und/oder Daten zuführbar ist. Eine Erkennungsschaltung führt den Vergleich der eintreffenden Befehle durch und die Steuerschaltung bestimmt entsprechend die Umkodierung des nächsten Befehls. Dieser Aufbau ist mit einfachen Schaltungselementen realisierbar.

Vorteilhaft ist es, wenn der Umkodierer einen die Art der Abänderung bzw. der Umkodierung enthaltenden Speicher umfaßt und daß der Zugriff zu diesem z. B. verschiedene Seiten bzw. Kodierungslisten enthaltenden Speicher von der Steuerschaltung steuerbar ist; durch Umkodierung bzw. Austausch des Speichers kann die Art der Umkodierung an verschiedene Verschlüsselungen angepaßt werden.

Eine erhöhte Verschlüsselungswirkung wird erzielt, wenn zur Abänderung der Umkodierung des zumindest nächsten einlangenden Befehls- und/oder Datenbytes durch den Umkodierer zumindest ein Eingang der Steuerschaltung auch an zumindest eine Adreßleitung angeschlossen ist und das von der Steuerschaltung gebildete Steuersignal für den Umkodierer in Abhängigkeit von Adressen und/oder Daten gebildet ist bzw. wenn ein Eingang des Umkodierers in an sich bekannter Weise an den Adreßbus angeschlossen ist und daß zumindest eine der auf diesem übertragenen Adressen zusätzlich zu den Umkodierbefehlen der Steuerschaltung zur Umkodierung von Befehlen und/oder Daten herangezogen ist. Die Umkodierung erfolgt dabei unter Zuhilfenahme weiterer Parameter, sodaß für Uneingeweihte eine Programmdekodierung nahezu hoffnungslos wird.

Da die Umkodierung erfindungsgemäß in Abhängigkeit von vorangehenden Befehlen (Daten) erfolgt, am Ende eines Interrupts aber abweichend vom logischen Aufbau des Hauptprogrammes keine definierten Befehle (Daten) einlangen, ist erfindungsgemäß vorgesehen, daß dem(n) Interrupteingang(gängen) des Mikroprozessors eine Sperrschaltung vorgeschaltet ist, deren Eingang an einen Ausgang der Steuerschaltung bzw. des Zwischenspeichers und/oder der Erkennungsschaltung angeschlossen ist und deren Sperrwirkung von dieser(m) bei Eintreffen eines eine Abänderung der Befehls- und/oder Daten-Umkodierung im Umkodierer (2) bewirkenden Befehls- und/oder Datenbytes für die Zeitdauer der vorgesehenen Abänderung auslösbar ist. Diese Möglichkeit ist somit für den ungestörten Programmablauf von außerordentlicher Wichtigkeit.

Eine weitere Ausführungsform der Erfindung sieht vor, daß der Umkodierer eine Verknüpfungseinrichtung, z. B. einen bit- oder byteweisen Addierer, einen Subtrahierer, ein Exklusiv-Oder-Gatter od. dgl., umfaßt, der von der Steuerschaltung, vorzugsweise einem Zwischenspeicher, der für jedes zu speichernde Bit ein Flip-Flop aufweist, zumindest ein eingelangtes bzw. dem Mikroprozessor zugeleitetes Befehls- und/oder Datenbyte zuführbar ist, das von der Verknüpfungseinrichtung mit zumindest einem einlangenden Befehls- und/oder Datenbyte verknüpft wird (z. B. addiert, subtrahiert od. dgl.) und daß die verknüpften Befehle und/oder Daten, z. B. die Bit-Summe, als gültige Befehle und/oder Daten dem Mikroprozessor zugeführt sind. Ein schaltungsmäßig einfacher Aufbau ergibt sich, wenn ein mit einem Eingang an die Datenleitung und mit einem Ausgang an den Löscho- bzw. Reseteingang der von einem Zwischenspeicher gebildeten Steuerschaltung angeschlossener Vergleichler für die gültigen Befehls- und/oder Datenbytes mit abgeänderter Kodierung vorgesehen ist und daß vom Vergleichler bei Feststellung einer Programmverzweigung und/oder anderen bestimmten Befehlen und/oder Daten durch ein an seinem Ausgang abgegebenes Signal der Zwischenspeicher löscher bzw. auf einen bestimmten Speicherstand einstellbar ist. Um dabei einen von Interruptroutinen ungestörter Programmablauf zu garantieren, ist erfindungsgemäß vorgesehen, daß dem(n) Interrupteingang(en) des Mikroprozessors eine im Normalfall die Durchschaltung der Interruptleitung(en) verhindernde Sperrschaltung vorgeschaltet ist, deren Eingang an den Ausgang des Vergleichlers angeschlossen ist und deren Sperrwirkung bei Feststellung von Programmverzweigungsbefehlen oder anderen bestimmten Befehlen durch ein vom Vergleichler zugeführtes Signal aufhebbar ist.

Um eine vom Umkodierer ungestörten Befehls(Daten)-Übermittlung zu gewährleisten, ist vorgesehen, daß zur Abgabe von Befehlen und/oder Daten des Mikroprozessors an die Speicher oder an eine Ausgabeeinheit eine zum Umkodierer parallel geschaltete einen Buffer enthaltende Befehls- bzw. Datenleitung vorgesehen ist.

Im folgenden wird die Erfindung anhand der Fig. 4 bis 7 näher erläutert.

Es zeigen Fig. 4, 5 eine erste Ausführungsform einer erfindungsgemäßen Anordnung zur Abänderung der Kodierung, Fig. 6 eine zweite Ausführungsform einer erfindungsgemäßen Anordnung zur Abänderung der Kodierung und Fig. 7 schematisch eine erfindungsgemäße Anordnung zur Verknüpfung eintreffender Signale zur Abänderung ihrer Kodierung.

Im einfachsten Fall einer solchen Umkodierung mit Gedächtnis schaltet ein bestimmtes Byte, das den Mikroprozessor erreicht, den Umkodierer (2) für den nächstfolgenden Befehl um. Dies soll anhand von Fig. 4 und den schon bekannten Programmbeispielen erläutert werden.

#### Beispiel 1

Adresse	Daten 1	Daten 2	UM	Befehl
0000	57	3E		LD A,
0001	10	10	0	10
0002	33	47	0	LD B,A
0003	33	47	0	LD B,A
0004	33	47	0	LD B,A
0005	33	47	0	LD B,A
0006	33	47	0	LD B,A
0007	33	47	0	LD B,A
"	"	"	"	"
"	"	"	"	"
"	"	"	"	"

Beispiel 2

	Adresse	Daten 1	Daten 2	UM	Befehl	
5	0000	57	3E		LD A,	
	0001	10	10	0		10
	0002	33	47	0	LD B,A	
10	0003	33	47	0	LD B,A	
	0004	72	6F	0	LD L,A	*1*
15	0005	57	47	1	LD B,A	*2*
	0006	33	47	0	LD B,A	
	0007	33	47	0	LD B,A	
20	"	"	"	"	"	
	"	"	"	"	"	
	"	"	"	"	"	

25 Das Beispiel 1 der Fig. 4 läuft wie Beispiel 1 der Fig. 2 ab, mit der zusätzlichen Erzeugung eines Signals auf einer Signalleitung ("UM"), einer Steuerschaltung (7) für den Umkodierer (2). Zwischen dem Umkodierer (2) und dem Mikroprozessor (2) ist der Eingang einer Erkennungsschaltung (7') geschaltet, die der Steuerschaltung (7) vorgeschaltet ist.

30 Auch das Beispiel 2 ist anfangs gleich, jedoch tritt in Programmzeile 0004 das Byte 6F auf, das von der Erkennungsschaltung (7') erkannt wird und daraufhin ein, die Steuereinrichtung (7) bzw. ein der Erkennungsschaltung (7') als Steuereinrichtung (7) zugeordnetes bzw. darin enthaltenes Flip-Flop od. dgl. umschaltet, womit auch die Signalleitung ("UM") auf 1 wechselt. Der Umkodierer (2) umfaßt zwei Umkodierer entsprechend Fig. 2, d. h. er hat zwei Seiten (Listen), wobei der Seite 1 UM = 0 und der Seite 2 UM = 1 zugeordnet ist. Diese Seiten können z. B. von verschiedenen Kodelisten gebildet sein.

35 Während also alle Bytes in Beispiel 1 und die meisten in Beispiel 2 laut Seite 1 (UM = 0) umkodiert werden, schaltet das Byte 6F in der durch \*1\* gekennzeichneten Zeile für den nächsten Befehl - gekennzeichnet durch \*2\* - den Umkodierer (2) auf Seite 2 um. Das Byte 57 von Adresse 0005 wird also - entsprechend der anderen Kodierungen enthaltenden Seite 2 - auf 47 umkodiert, während es sonst auf 3E bzw. das Byte für 33 auf 47 umkodiert würde.

40 Der Umkodierer (2) wandelt somit die Daten (1) in die Daten (2) um, in Abhängigkeit von der Vorgeschichte der Daten (z. B. vorangehender Befehl, vorangehende Adresse, od. dgl.), welche Umkodierung unter Steuerung der Steuerschaltung (7) erfolgt.

45 Diese Abhängigkeit der Umkodierung von der Vorgeschichte bringt jedoch ein Problem mit sich: Es darf in dem Augenblick, in dem im Umkodierer auf die Seite 2 umgeschaltet ist, kein Interrupt angefordert werden, da das erste Befehlsbyte des Interrupt-Programms die Seite 1 voraussetzt und mit UM = 1 falsch übersetzt und daher auch falsch ausgeführt würde. Auch der letzte Befehl eines Interrupt-Programmes könnte zu einer unrichtigen Umkodierung des z. B. ersten Befehls des Hauptprogrammes führen. Um dies zu verhindern, muß der Interrupt nach jedem Befehl, der die nachfolgende Entschlüsselung verändert, durch Hardware gesperrt werden. Diese Aufgabe übernimmt eine Sperrschaltung (3), die in der Interruptleitung (IR) angeordnet ist, und wenn UM = 1 ist, kein Interruptsignal zum Mikroprozessor durchläßt.

50 Die Schaltung in Fig. 5 ist eine Detailschaltung des vergossenen (mit den strichlierten Linien angedeuteten) Moduls gemäß Fig. 4.

Über die Reset-Leitung (R) wird der Mikroprozessor in einen definierten Anfangszustand versetzt und gleichzeitig das D-Flip-Flop (7) und damit auch das Signal (UM) auf 0 gesetzt, womit auch die Umkodierer-Seite 1 gewählt wird. Über den Adreßbus (AB) gelangt die Adresse 0000 hinaus, am Datenbus (DB) kommt das Byte 57, gelangt als Adresse (vergleichbar mit einer Zeilennummer in einem Nachschlagewerk) zum PROM, an dessen Ausgang das übersetzte Byte 3E erscheint. Es wird von dort einerseits zum Mikroprozessor übertragen und andererseits zum Eingang (B) des als Erkennungsschaltung dienenden Vergleichers (7'), der die dort anliegenden Werte mit dem Eingang (A) vergleicht, an dem die vorgegebenen, z. B. aufgelisteten Umschaltbefehle anliegen, und bei Gleichheit seinen Ausgang ("A = B") auf 1 setzt (Anfangswert 0).

Als nächstes wird die Adresse 0001 auf den Adreßbus gelegt, am Datenbus erscheint 10 und wird vom PROM unverändert "durchgelassen", was dadurch möglich wird, daß der Mikroprozessor mit dem Signal auf der Leitung

(M1) alle Befehle und Daten unterscheidbar macht (Daten: M1 = 1 Befehle: M1 = 0).

Nun kommt die Adresse 0002, das Byte 33 wird übertragen und vom PROM in 47 (LD B,A) übersetzt.

In der als D-FF ausgebildeten Erkennungsschaltung (7) wird bei jedem Befehl - gesteuert durch (M1) ODER-verknüpft mit (RD) (Read) Signal des Mikroprozessors - der Zustand von A = B gespeichert, der dem Signal (UM) entspricht.

Gelangt das Byte 6F zum Vergleich (7'), wird A = B 1 (aufgrund der Übereinstimmung zwischen einlangenden und am Eingang (A) anliegenden Bytes) in die Erkennungsschaltung (7) übernommen und es wird UM=1. Das Signal (UM) schaltet das PROM bzw. den Umkodierer (2) auf eine andere "Seite" bzw. "Liste" um und sperrt mit Hilfe der beiden ODER-Gatter (10), (11) die beiden (Low-Aktiven) Interrupts (INT) und (NMI).

Daten, die aus dem Mikroprozessor hinaus in den Speicher (4) oder zur Peripherie kommen sollen, werden durch den (Tri-State-)Buffer (1), der der Leitung (1) (in Fig. 2 bis 4) entspricht, um das PROM herumgeleitet. Die Richtung des Datenflusses wird vom Mikroprozessor durch die Leitungen (RD) (Read) und (WR) (Write), die entweder den Umkodierer (2) oder den Buffer (1) freigeben, gesteuert.

Die Schaltung nach Fig. 6 funktioniert ähnlich wie die nach Fig. 5. Das PAL, im vorliegenden Fall ein PAL20R4, eine programmierbare Logikschaltung, enthält bestimmte D-Flip-Flops, UND-Gatter und ODER-Gatter, deren Verschaltung einmal gewählt und programmiert werden kann und dann hardwaremäßig festgelegt ist. In diesem Beispiel wird es so programmiert, daß es die Aufgabe der ODER-Gatter (10), (11) der Erkennungsschaltung (7') (Vergleicher) und der Steuereinrichtung (7) (D-FF) übernimmt. Zu diesem Zweck werden die Leitungen (M1), (RD), (INT), (NMI) und der Datenbus zum PAL geführt. Weggeführt vom PAL sind die Leitungen (INT'), (NMI') und eine Steuerleitung (A7) für den Umkodierer (2) (PROM), mit der PROM-"Seiten" zur Abänderung der Kodierung in Abhängigkeit von einer bestimmten Vorgeschichte vertauscht werden können.

Der Buffer (1) hat die gleiche Funktion wie in Fig. 5.

Auf dem Weg zum Mikroprozessor werden vier Datenleitungen parallel zum Umkodierer (2) (PROM) geführt, die anderen vier ((D0) bis (D3)) werden zum Umkodierer (2) geführt. Es erfolgt eine Umkodierung in Abhängigkeit von den Signalen auf den Datenleitungen (D5) und (D4), von den Signalen der Adreßleitung (A0), (ähnlich wie in Fig. 3) und von Signalen auf der Steuerleitung (A7) von PAL, mit der auch dafür gesorgt wird, daß Daten nicht verändert werden, und natürlich in Abhängigkeit von den Signalen auf den Datenleitungen (D0) bis (D3) selbst.

Man kann also sagen, daß dieser Umkodierer 16 "Seiten" hat, nämlich  $2^4$  Seiten, da die Datenleitungen (D5) und (D4), die Adreßleitung (A0) (Adresse eines Bytes gerade oder ungerade) und die Steuerleitung (A7) jeweils zwei Zustände annehmen bzw. ausdrücken können. Bei Erkennen eines Umschaltbefehles werden für den nächsten Befehl - gesteuert durch das PAL bzw. durch in einem in einem Speicher vorgesehene Befehle - die "Seiten" im Umkodierer vertauscht, was zu anderen Übersetzungen führt, wie es z. B. 2 (PROM) in den Programmbeispielen die zu Fig. 6 gehörige geänderte Kodierung des, dem Befehl LD L,A (6F) folgenden Befehles LD B,A von Adresse 0005 zeigt.

Eine weitere Möglichkeit, die "Vorgeschichte" für die Umkodierung wesentlich zu machen, zeigt die Schaltung in Fig. 7. Es sind hier nur noch die wesentlichen Details gezeichnet, die restliche Schaltung funktioniert analog den vorherigen Fig.

Die Befehlsbytes kommen zu einem als Addierschaltung ausgebildeten Umkodierer (2), in dem sie zu dem Wert des vorher durchgeführten Befehls, der in einer als Zwischenspeicher ausgebildeten Steuerschaltung (7) abgelegt wurde, addiert werden. Die Summe gelangt zum Mikroprozessor, wo der entsprechende Befehl ausgeführt wird, und außerdem in die Steuerschaltung (7). Wenn das nächste Byte über den Datenbus kommt, wird im Umkodierer (Addierer) der Inhalt der Steuerschaltung (7) (Zwischenspeicher) dazugezählt, und das Ergebnis ist das richtige Befehlsbyte.

Anstelle einer Addition können Subtraktionen, Multiplikationen, Exklusiv-Oder-Verknüpfungen, usw. stattfinden. Der Zwischenspeicher besitzt für jedes Bit ein Flip-Flop.

Nach Programmverzweigungen, z. B. Sprungbefehlen, muß der Zwischenspeicher bzw. die Steuerschaltung (13) auf 0 gesetzt werden, da die Sprungziele von verschiedenen Stellen mit verschiedenen Befehlen erreicht werden können und dort trotzdem eine definierte Umkodierung stattfinden muß. Zu diesem Zweck dient der Vergleich (14), der die Programmverzweigungsbefehle bzw. Sprungbefehle erkennt, danach den Zwischenspeicher auf 0 setzt und auch die Interrupt-Sperre freigibt.

Es können beliebige Beispiele erstellt werden; das Prinzip einiger Beispiele soll aber noch kurz erwähnt werden:

\*\*\* Invertieren des nächsten Daten- und/oder Befehlsbytes nach bestimmten, bereits eingelangten Bytes

\*\*\* Exklusiv-Oder-Verknüpfung des zuletzt durchgeführten Befehls mit dem neuen Byte (wie Fig. 7 mit EOR-Verknüpfung statt Addition).

\*\*\* Vertauschen bestimmter Datenbus-Bites in Abhängigkeit von der Summe der durchgeführten Befehle.

Zusätzlich zu den bis jetzt aufgezählten Umkodierer-Verfahren kann man auch die aufgezählten Methoden gemeinsam, beliebig kombiniert, verwenden.

Allen gemeinsam ist jedoch das Vorhandensein einer vorzugsweise von mindestens einem Flip-Flop

gebildeten Steuerschaltung (7) deren Zustand von einem oder mehreren, vorher durchgeführten Befehl(en) abhängt und die auf den Umkodierer (2) Einfluß hat, und so die Art der Umkodierung von der Vorgeschichte abhängig macht.

#### Programmbispiele

	Adresse	Befehl	Kode für Schaltung laut					
			Fig. 1	Fig. 2	Fig. 3	Fig. 4+5	Fig. 6	Fig. 7
5								
10	0000	LD A,10	3E 10	57 10	57 10	57 10	37 10	3E 10
	0002	LD B,A	47	33	33	33	41	09
	0003	LD B,A	47	33	12	33	43	00
	0004	LD B,A	47	33	98	33	4F	00
	0005	LD B,A	47	33	12	33	43	00
	0006	LD B,A	47	33	33	33	41	00
15	0007	LD B,A	47	33	12	33	43	00
	0008	LD B,A	47	33	98	33	4F	00
	0009	"	"	"	"	"	"	"
	000A	"	"	"	"	"	"	"
	000B	"	"	"	"	"	"	"
20								
	0000	LD A 10	3E 10	57 10	57 10	57 10	37 10	3E 10
	0002	LD B,A	47	33	33	33	41	09
	0003	LD B,A	47	33	12	33	43	00
	0004	LD L,A	6F 72	86	72	72	62	00
	0005	LD B,A	47	33	12	57	48	00
	0006	LD B,A	47	33	33	33	41	00
25	0007	LD B,A	47	33	12	33	43	00
	0008	LD B,A	47	33	98	33	4F	00
	0009	"	"	"	"	"	"	"
	000A	"	"	"	"	"	"	"
	000B	"	"	"	"	"	"	"
30								
35								

\*1\*

\*2\*

#### PATENTANSPRÜCHE

- 40 1. Anordnung zum Entschlüsseln von in zumindest einem Befehls- und/oder Datenspeicher abgelegten Befehlen und/oder Daten, wobei die Befehle und/oder Daten auf Abfrage aus den genannten Speichern einem Mikroprozessor über einen gegebenenfalls mit diesem hybridisierten bzw. vergossenen Umkodierer zur Verarbeitung zugeleitet sind und wobei im Umkodierer zumindest ein Teil der dem(n) Speicher(n) entnommenen
- 45 Befehle und/oder Daten durch im Umkodierer vorgegebene oder erstellte Befehle und/oder Daten ersetzt bzw. in anders lautende Befehle und/oder Daten umgewandelt wird, die sodann dem Mikroprozessor als gültige Befehle und/oder Daten über eine vom Ausgang des Umkodierers zum Befehls- bzw. Dateneingang des Mikroprozessors führende Datenleitung zugeführt sind, **dadurch gekennzeichnet**, daß an einen Eingang des Umkodierers (2) der Ausgang einer Steuerschaltung (7), z. B. eines Zwischenspeichers in Form zumindest eines Flip-Flops, angeschlossen ist, daß der Eingang der Steuerschaltung (7) an die Datenleitung zwischen dem Umkodierer (2) und dem Mikroprozessor (MP) angeschlossen ist und daß die Art der Umkodierung des Umkodierers (2) von der Steuerschaltung (7) in Abhängigkeit von zumindest einem der vorangehend über die Datenleitung dem Mikroprozessor zur Ausführung bzw. Abarbeitung im Zuge des Programmlaufes zugeführten Befehls- und/oder Datenbyte bestimmt ist.
- 50
- 55 2. Anordnung nach Anspruch 1, **dadurch gekennzeichnet**, daß ein Eingang der Steuerschaltung (7) mit dem Ausgang einer Erkennungsschaltung (7'), z. B. einem Vergleichs-, für die dem Mikroprozessor zugeführten Befehle und/oder Daten verbunden ist und daß der Steuerschaltung (7) von der Erkennungsschaltung (7') bei Feststellung einer Übereinstimmung zwischen zumindest einem auf der Datenleitung eingelangten Befehls- und/oder Datenbytes mit zumindest einem in der Erkennungsschaltung enthaltenen vorgegebenen Befehls- und/oder Datenbyte ein Signal zur Umschaltung des Umkodierers (2) bzw. zur Abänderung der Umkodierung einer bestimmten Anzahl nachfolgend einlangender Befehle und/oder Daten zuführbar ist.
- 60



3. Anordnung nach Anspruch 1 oder 2, **dadurch gekennzeichnet**, daß der Umkodierer (2) einen die Art der Abänderung bzw. der Umkodierung enthaltenden Speicher umfaßt und daß der Zugriff zu diesem z. B. verschiedene Seiten bzw. Kodierungslisten enthaltenden Speicher von der Steuerschaltung (7) steuerbar ist.

4. Anordnung nach einem der Ansprüche 1 bis 3, **dadurch gekennzeichnet**, daß zur Abänderung der Umkodierung des zumindest nächsten einlangenden Befehls- und/oder Datenbytes durch den Umkodierer (2) zumindest ein Eingang der Steuerschaltung (7) auch an zumindest eine Adreßleitung angeschlossen ist und das von der Steuerschaltung (7) gebildete Steuersignal für den Umkodierer (2) in Abhängigkeit von Adressen und/oder Daten gebildet ist. (Fig. 6).

5. Anordnung nach einem der Ansprüche 1 bis 4, **dadurch gekennzeichnet**, daß ein Eingang des Umkodierers (2) in an sich bekannter Weise an den Adreßbus angeschlossen ist und daß zumindest eine der auf diesem übertragenen Adressen zusätzlich zu den Umkodierbefehlen der Steuerschaltung (7) zur Umkodierung von Befehlen und/oder Daten herangezogen ist. (Fig. 6).

6. Anordnung nach einem der Ansprüche 1 bis 5, **dadurch gekennzeichnet**, daß dem(n) Interrupteingang(gängen) des Mikroprozessors (MP) eine Sperrschaltung (3, 10, 11) vorgeschaltet ist, deren Eingang an einen Ausgang der Steuerschaltung (7) bzw. des Zwischenspeichers und/oder der Erkennungsschaltung angeschlossen ist und deren Sperrwirkung von dieser(m) bei Eintreffen eines eine Abänderung der Befehls- und/oder Daten-Umkodierung im Umkodierer (2) bewirkenden Befehls- und/oder Datenbytes für die Zeitdauer der vorgesehenen Abänderung auslösbar ist.

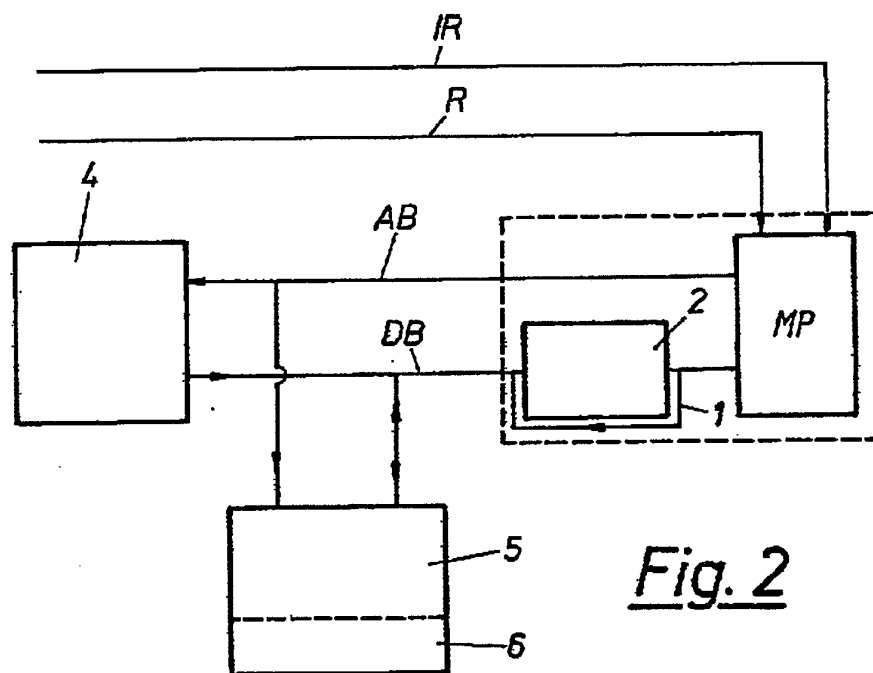
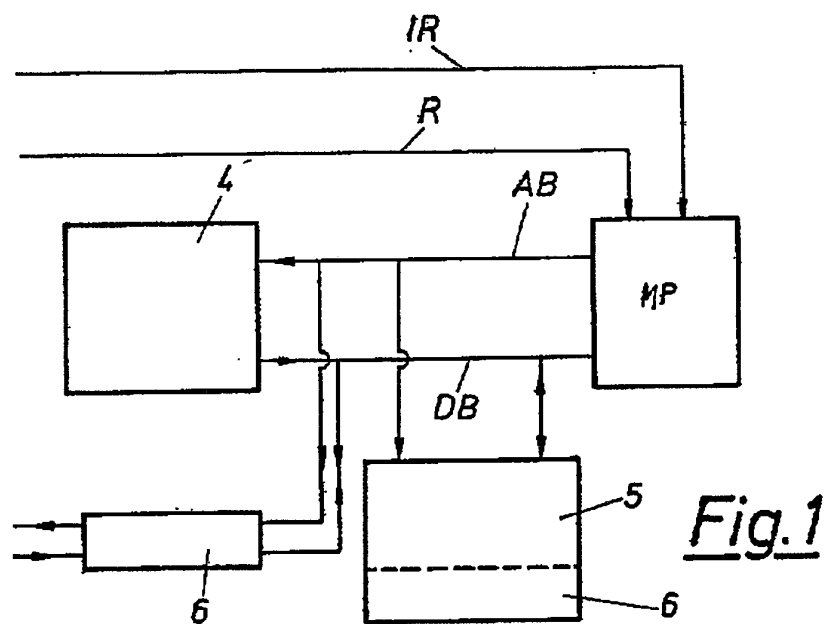
7. Anordnung nach Anspruch 1, **dadurch gekennzeichnet**, daß der Umkodierer (2) eine Verknüpfungseinrichtung, z. B. einen bit- oder byteweisen Addierer, einen Subtrahierer, ein Exklusiv-Oder-Gatter od. dgl., umfaßt, der von der Steuerschaltung (7), vorzugsweise einem Zwischenspeicher, der für jedes zu speichernde Bit ein Flip-Flop aufweist, zumindest ein eingelangtes bzw. dem Mikroprozessor (MP) zugeleitetes Befehls- und/oder Datenbyte zuführbar ist, das von der Verknüpfungseinrichtung mit zumindest einem einlangenden Befehls- und/oder Datenbyte verknüpft wird (z. B. addiert, subtrahiert od. dgl.) und daß die verknüpften Befehle und/oder Daten, z. B. die Bit-Summe, als gültige Befehle und/oder Daten dem Mikroprozessor (MP) zugeführt sind.

8. Anordnung nach Anspruch 7, **dadurch gekennzeichnet**, daß ein mit einem Eingang an die Datenleitung und mit einem Ausgang an den Löscho- bzw. Reseteingang der von einem Zwischenspeicher gebildeten Steuerschaltung (7) angeschlossener Vergleichler (14) für die gültigen Befehls- und/oder Datenbytes mit abgeänderter Kodierung vorgesehen ist und daß vom Vergleichler (14) bei Feststellung einer Programmverzweigung und/oder anderen bestimmten Befehlen und/oder Daten durch ein an seinem Ausgang abgegebenes Signal der Zwischenspeicher löscho bzw. auf einen bestimmten Speicherstand einstellbar ist.

9. Anordnung nach Anspruch 8, **dadurch gekennzeichnet**, daß dem(n) Interrupteingang(gängen) des Mikroprozessors (MP) eine im Normalfall die Durchschaltung der Interruptleitung(en) (INT) verhindernde Sperrschaltung (3) vorgeschaltet ist, deren Eingang an den Ausgang des Vergleichlers (14) angeschlossen ist und deren Sperrwirkung bei Feststellung von Programmverzweigungsbefehlen oder anderen bestimmten Befehlen durch ein vom Vergleichler (14) zugeführtes Signal aufhebbar ist.

10. Anordnung nach einem der Ansprüche 1 bis 9, **dadurch gekennzeichnet**, daß zur Abgabe von Befehlen und/oder Daten des Mikroprozessors (MP) an die Speicher (4), (5) oder an eine Ausgabereinheit (6) eine zum Umkodierer (2) parallel geschaltete, einen Buffer enthaltende Befehls- bzw. Datenleitung (1) vorgesehen ist.

Hiezu 5 Blatt Zeichnungen



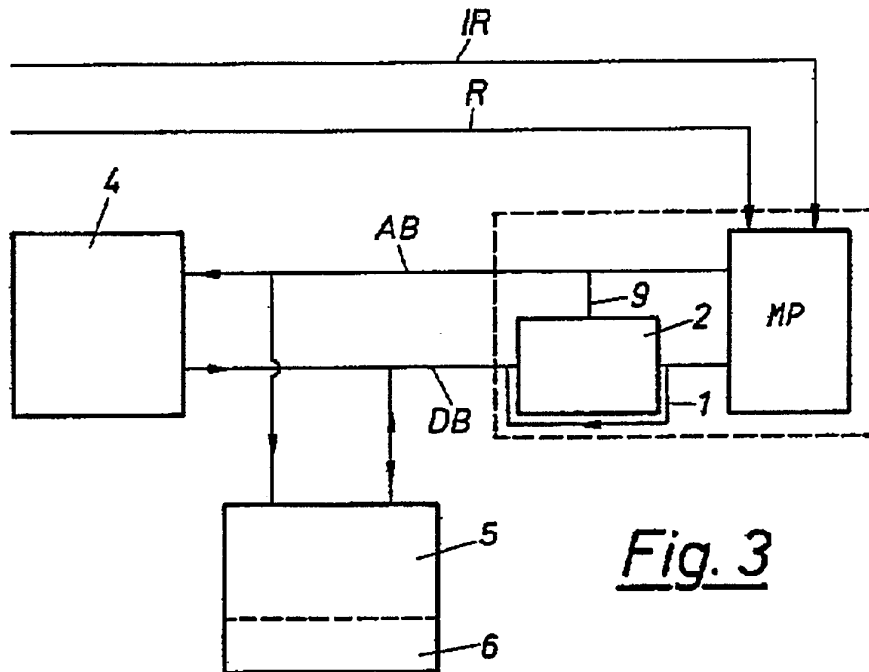


Fig. 3

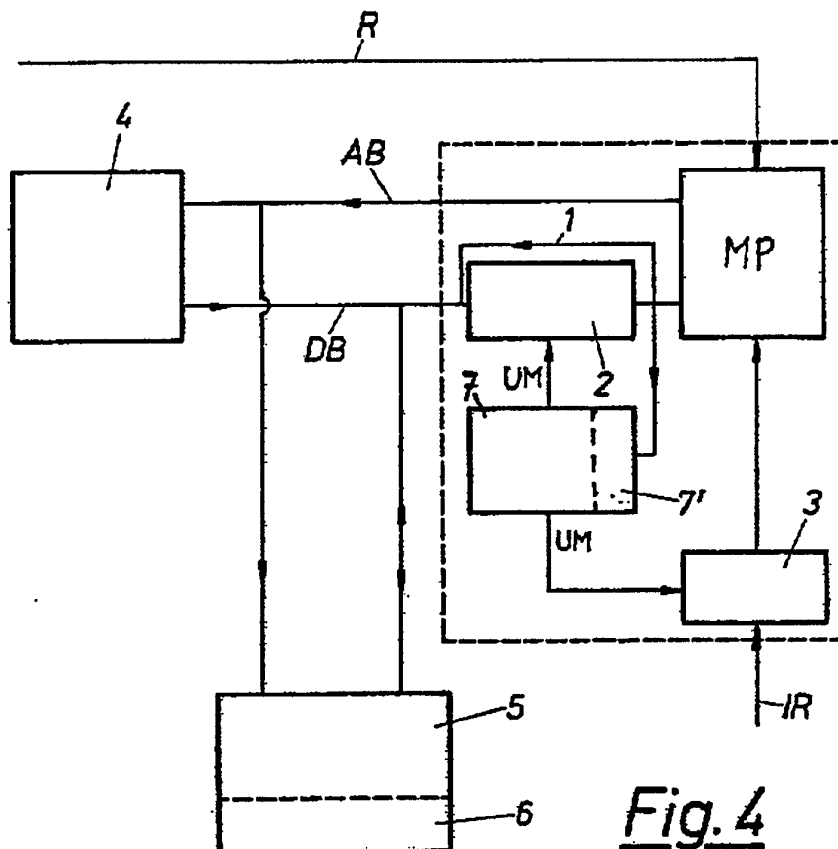


Fig. 4

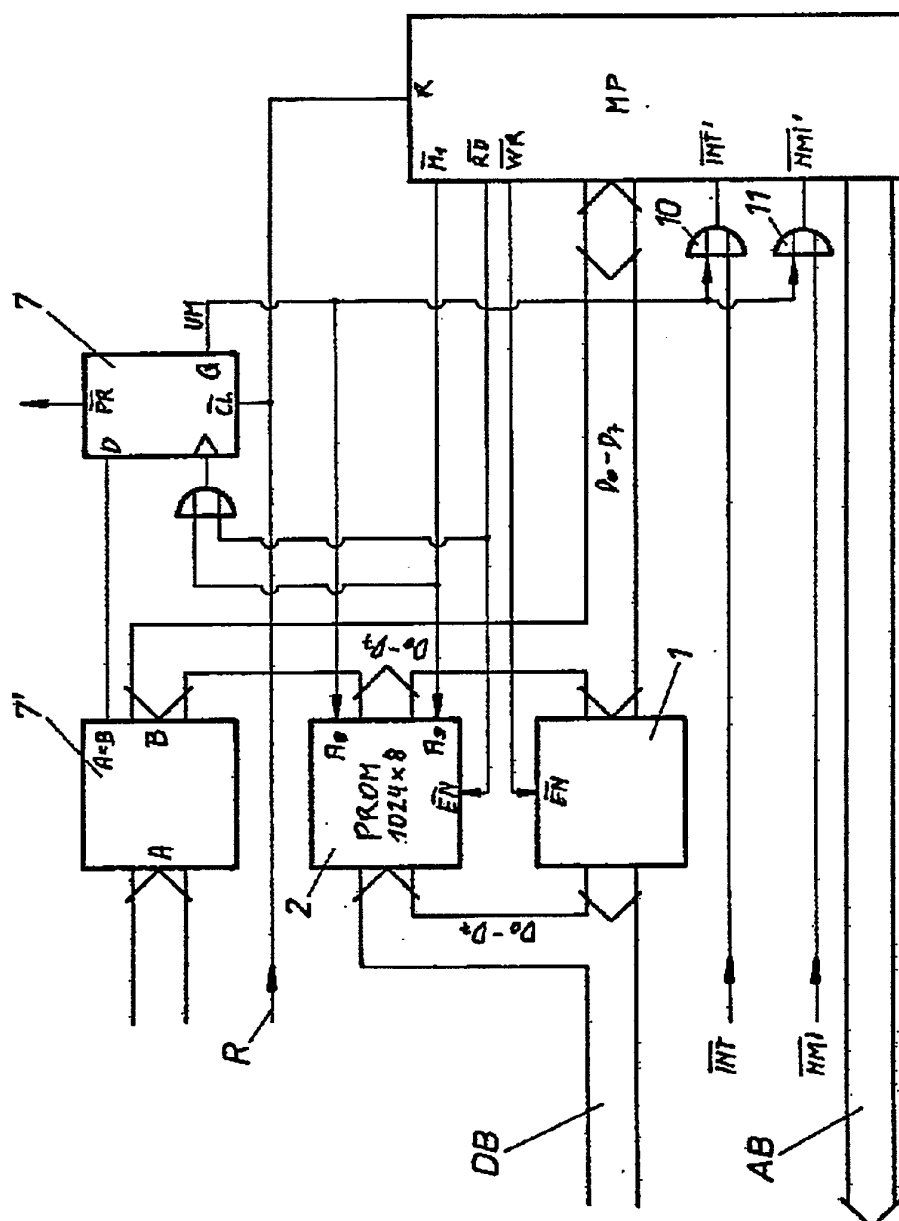


Fig. 5

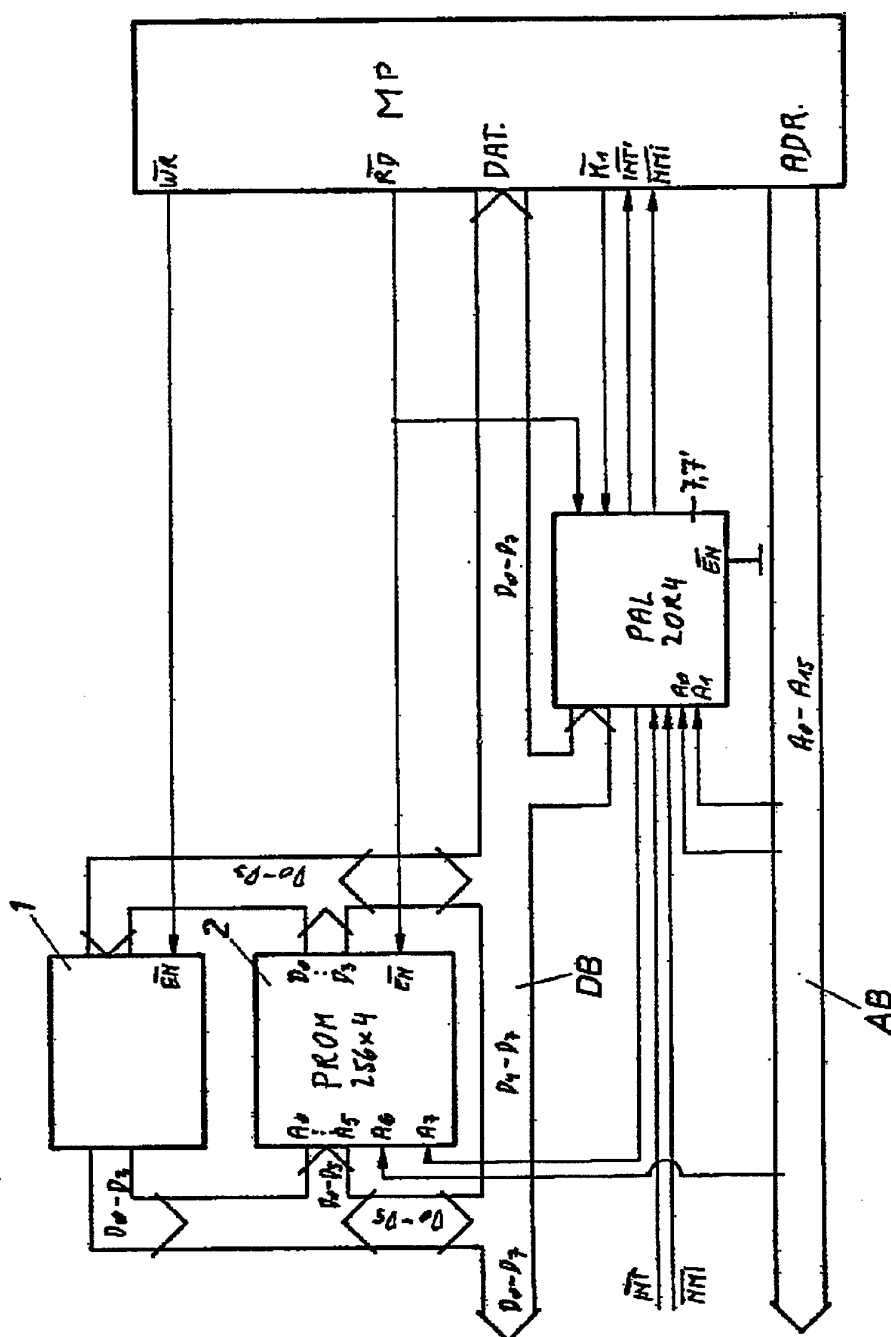


Fig. 6

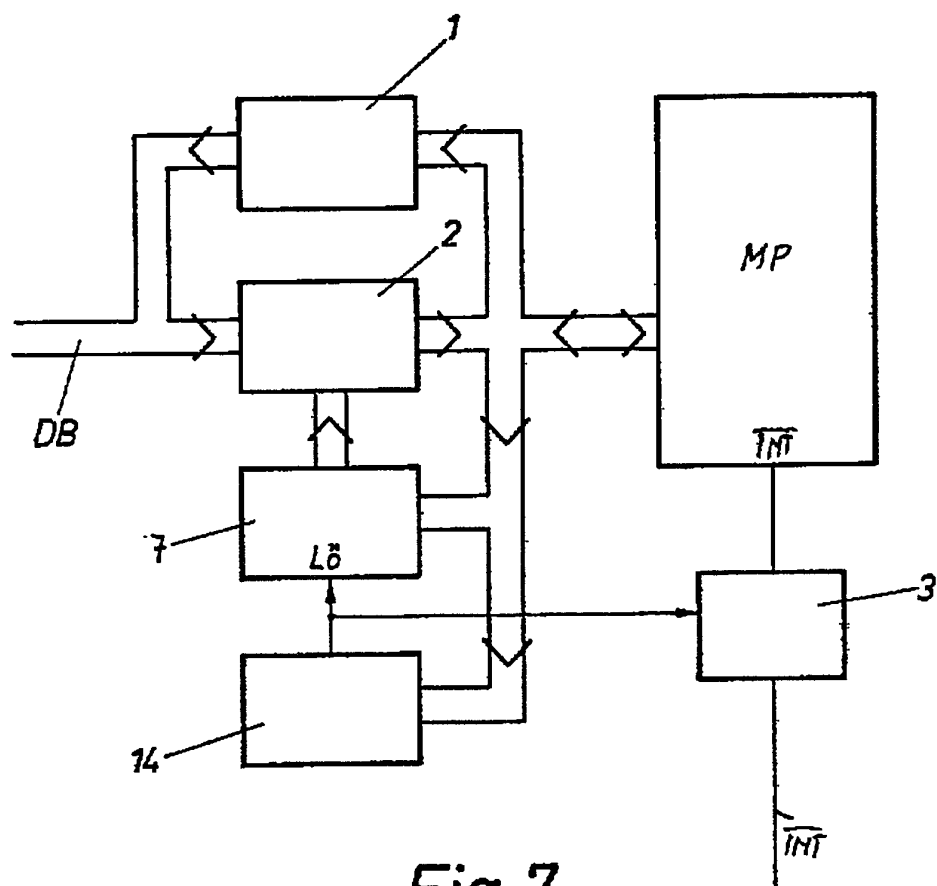


Fig. 7