



(19)  
Bundesrepublik Deutschland  
Deutsches Patent- und Markenamt

(10) DE 699 26 365 T2 2006.05.24

(12)

## Übersetzung der europäischen Patentschrift

(97) EP 1 080 422 B1

(21) Deutsches Aktenzeichen: 699 26 365.4

(86) PCT-Aktenzeichen: PCT/US99/10171

(96) Europäisches Aktenzeichen: 99 924 159.9

(87) PCT-Veröffentlichungs-Nr.: WO 99/060488

(86) PCT-Anmeldetag: 10.05.1999

(87) Veröffentlichungstag

der PCT-Anmeldung: 25.11.1999

(97) Erstveröffentlichung durch das EPA: 07.03.2001

(97) Veröffentlichungstag

der Patenterteilung beim EPA: 27.07.2005

(47) Veröffentlichungstag im Patentblatt: 24.05.2006

(51) Int Cl.<sup>8</sup>: G06F 17/00 (2006.01)

G06F 13/26 (2006.01)

G06F 9/46 (2006.01)

G06F 13/24 (2006.01)

(30) Unionspriorität:

82432 20.05.1998 US

(84) Benannte Vertragsstaaten:

AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LI, LU, MC, NL, PT, SE

(73) Patentinhaber:

Sony Electronics Inc., Park Ridge, N.J., US

(72) Erfinder:

COX, R., Steven, San Jose, US

(74) Vertreter:

Mitscherlich & Partner, Patent- und  
Rechtsanwälte, 80331 München

(54) Bezeichnung: SOFTWARE-KONFIGURIERBARE TECHNIK ZUM PRIORISIEREN VON UNTERBRECHUNGEN IN  
EINEM AUF EINEM MIKROPROZESSOR BASIERTEN SYSTEM

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 eingelebt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

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

## Beschreibung

**[0001]** Die Erfindung betrifft das Gebiet der mikroprozessorbasierten Systeme. Die Erfindung bezieht sich insbesondere auf das Konfigurieren und Priorisieren von Interrupts in mikroprozessorbasierten Systemen.

**[0002]** Mikroprozessorbasierte Systeme arbeiten im allgemeinen so, daß Befehle eines Softwareprogramms, die in aufeinanderfolgenden Speicherplätzen gespeichert sind, sequentiell ausgeführt werden. Ein Programmzähler enthält eine Speicheradresse des nächsten gespeicherten Befehls innerhalb der Sequenz. Dieser Programmzähler wird bei jedem ausgeführten Befehl inkrementiert. Das Softwareprogramm kann Verzweigungen enthalten, die nach Maßgabe von in dem Programm spezifizierten Bedingungen beschritten werden. Für eine bedingte Verzweigung wird in dem Programmzähler eine Zweigadresse angeordnet, falls die Bedingung erfüllt ist, und die sequentielle Ausführung wird, beginnend mit der Zweigadresse, wieder aufgenommen.

**[0003]** Es ist jedoch typisch, daß das mikroprozessorbasierte System auf das Eintreten von Ereignissen reagieren muß, die mit dem gespeicherten Softwareprogramm nicht synchronisiert sind. So müssen z.B. Daten aus einem peripheren Gerät, wie einem Modem oder einer Tastatur, nach der Erfassung der Daten alsbald verarbeitet werden. Dementsprechend haben Mikroprozessoren üblicherweise die Fähigkeit, ein oder mehrere Interrupt-Signale zu empfangen, die von einem oder mehreren peripheren Geräten aktiviert werden. Als Reaktion auf ein aktiviertes Interrupt-Signal unterbricht der Mikroprozessor die Folge von Befehlen, die gerade ausgeführt wird, und beginnt mit der Ausführung eines anderen Befehlsatzes, der im allgemeinen als eine für den aktivierten Interrupt passende Interrupt-Serviceroutine bezeichnet wird.

**[0004]** Um mit der Ausführung der passenden Interrupt-Serviceroutine zu beginnen, wird die Adresse in dem Programmzähler gesichert, und in dem Programmzähler wird eine Vektoradresse angeordnet, die auch als Interrupt-Vektor bezeichnet wird. Im allgemeinen stehen zwei Verfahren zur Festlegung der passenden Vektoradresse für die Ausführung der Interrupt-Serviceroutine zur Verfügung, nämlich ein vektorisiertes und ein autovektorisiertes Verfahren. Wenn ein Interrupt vektorisiert ist, liefert das periphere Gerät die Startadresse der passenden Interrupt-Serviceroutine. Diese Startadresse wird in dem Programmzähler des Mikroprozessors plaziert. Alternativ kann die in dem Programmzähler angeordnete Adresse die Adresse eines Platzes in dem Speicher sein, der die Startadresse der Interrupt-Serviceroutine enthält. Wenn ein Interrupt autovektorisiert ist, zeigt die Vektoradresse auf einen vorbestimmten

Platz in dem Speicher unabhängig davon, ob der Interrupt aktiv ist. Dieser vorbestimmte Platz ist üblicherweise die Startadresse einer Softwareroutine, die im allgemeinen als Interrupt-Handler bezeichnet wird. Der Interrupt-Handler initiiert dann eine für das aktive Interrupt-Signal passende Interrupt-Serviceroutine.

**[0005]** Sobald der Mikroprozessor die Interrupt-Serviceroutine ausgeführt hat, wird der Programmzähler so restauriert, daß die Ausführung der ursprünglichen Befehlssequenz wieder aufgenommen wird.

**[0006]** Wenn zwei oder mehr Interrupt-Signale gleichzeitig anhängig sind, muß der Mikroprozessor mit einer geeigneten Priorität auf die Interrupts antworten. Deshalb ist in mikroprozessorbasierten Systemen im allgemeinen die Möglichkeit vorgesehen, die Interrupt-Signale zu priorisieren. Darüber hinaus ist es unter gewissen Umständen wünschenswert, daß der Mikroprozessor ein oder mehrere Interrupt-Signale ignoriert. So z.B. wenn der Mikroprozessor eine kritische Operation ausführt, die nicht unterbrochen werden darf. Zu diesem Zweck enthalten Mikroprozessoren im allgemeinen eine Einrichtung zum Maskieren von Interrupts.

**[0007]** Ein Beispiel für einen in großem Umfang benutzten Mikroprozessor sind die Prozessoren der 68000-Serie von Motorola, Inc.. Die Interrupts für einen Motorola-Prozessor der 68000-Serie werden implementiert, indem von jedem peripheren Gerät eine festverdrahtete Interrupt-Signalleitung mit einem Interrupt-Prioritätskodierer verbunden wird. Der Interrupt-Prioritätskodierer ist eine logische Hardware-Schaltung, die jedes aktive Interrupt-Signal aufnimmt und nur das aktive Interrupt-Signal mit der höchsten Priorität an einen von mehreren Interrupt-Eingängen des Motorola-Prozessors der Serie 68000 liefert. Der Prozessor initiiert dann die passende Interrupt-Serviceroutine nach dem vektorisierten oder autovektorierten Verfahren.

**[0008]** Ein Nachteil dieses Verfahrens zum Implementieren der Interrupts in den Motorola-Prozessoren der Serie 68000 besteht darin, daß die Verbindungen der festverdrahteten Interrupt-Signalleitungen zu dem Interrupt-Prioritätskodierer neu gruppiert werden müssen, um die relativen Prioritäten der Interrupt-Signale zu ändern. Das bedeutet, daß die relativen Prioritäten unter den Interrupts praktisch nicht geändert werden können, sobald ein solches System konfiguriert ist.

**[0009]** Ein anderes Beispiel für einen allgemein verbreiteten Mikroprozessor sind die RISC-(Reduced Instruction Set Computers)-Prozessoren von MIPS Technologies, Inc., wie z.B. die Prozessoren 82000, 83000, 84000 und 86000. Der MIPS-RISC-Prozessor wird durch das Setzen passender Bits in einem spe-

zialisierten Register, das als Fallregister bezeichnet wird, über anhängige Interrupts informiert. Der Prozessor initiiert dann die passende Interrupt-Service-routine, indem er zunächst einen Interrupt-Handler ausführt. Der Interrupt-Handler legt die Priorität unter den anhängigen Interrupts fest.

**[0010]** Ein Nachteil dieses Verfahrens zur Implementierung der Interrupts in MIPS-RISC-Prozessoren besteht darin, daß zur Änderung der relativen Prioritäten der Interrupt-Signale der Interrupt-Handler geändert werden muß. Somit können die relativen Prioritäten unter den Interrupts praktisch nicht geändert werden, sobald ein solches System konfiguriert ist. Darüber hinaus wird bei dem Abarbeiten der Interrupts durch die für den Zugriff auf den und die Ausführung des Interrupt-Handlers eine Latenzzeit eingeführt. Diese Latenzzeit kann in einem solchen System unerwünschte Größen erreichen.

**[0011]** Man benötigt deshalb wird ein leichter konfigurierbares Verfahren zum Priorisieren von Interrupts in einem mikroprozessorbasierten System.

**[0012]** In US-A-5 594 905 ist ein Ausnahme-Handler beschrieben, bei dem eine Prioritätstabelle vorgesehen ist, in der Informationen über die Prioritäten zur Behandlung eines oder mehrerer konkurrierend erzeugter Interrupts gespeichert sind. Der Ausnahme-Handler enthält auch ein Fallregister, das Flag-Bits aufweist, um anzusehen, wann ein Interrupt erzeugt wurde, wobei die Bits in dem Fallregister eine indexierende Adresse zu einer Stelle in der Prioritätstabelle definieren, die die Prioritätsinformation für eine gegebene Kombination aus aktiven und nichtaktiven Interrupts enthält. In einer Implementierung speichert die Prioritätstabelle die Adresse von Interrupt-Serviceroutinen (ISRs) für die Behandlung desjenigen aktiven Interrupts, das von allen laufend aktiven Interrupts die höchste Priorität hat. In einer anderen Implementierung speichert die Prioritätstabelle Indexadressen zu einer zweiten ISR-Adressentabelle, die die Interrupts mit den Adressen der entsprechenden ISRs verknüpft, die sie bedienen.

**[0013]** Die Aspekte und Merkmale der Erfindung sind in den Ansprüchen definiert.

**[0014]** Ein Ausführungsbeispiel der Erfindung liefert ein softwarekonfigurierbares Verfahren zum Priorisieren und Maskieren von Interrupts in einem mikroprozessorbasierten System. Es sind mehrere Interrupt-Konfigurationsregister zum Abbilden eines entsprechenden Exemplars aus einer Mehrzahl von Interrupt-Signalen auf ein geeignetes Exemplar aus einer Mehrzahl von Interrupt-Vektoradressenregistern und zum Maskieren ausgewählter Exemplare der Interrupt-Signale vorgesehen. Jedes einzelne aus der Mehrzahl von Interrupt-Konfigurationsregistern ist softwarekonfigurierbar, so daß es ein Bitmuster ent-

hält, das eines aus der Mehrzahl von Vektoradres- senregistern identifiziert oder aber anzeigt, daß das entsprechende Interrupt-Signal maskiert werden soll. Die Mehrzahl von Vektoradressenregistern ist entsprechend einer vorbestimmten Priorität angeordnet, und jedes einzelne aus der Mehrzahl von Vektoradres- senregistern ist softwarekonfigurierbar, so daß es die Startadresse einer passenden Interrupt-Serviceroutine für den entsprechenden Interrupt enthält.

**[0015]** Jedes der Interrupt-Signale wird einem Ein- gang eines entsprechenden Exemplars aus einer Mehrzahl von Demultiplexern zugeführt. Jedes einzelne aus der Mehrzahl von Interrupt-Konfigurationsregistern ist mit den Auswahlleitungen eines entspre- chenden Exemplars aus der Mehrzahl von Demultiplexern verbunden. Die Ausgänge jedes einzelnen aus der Mehrzahl von Demultiplexern sind entspre- chend der logischen "ODER"-Funktion durch eine Mehrzahl von logischen "ODER"-Gliedern mit entspre- chenden Ausgängen jedes anderen Demultiplexers verbunden, um an den Ausgängen der logischen "ODER"-Glieder eine Mehrzahl von Aktivierungssig- nalen auszugeben. Die einzelnen Aktivierungssig- nale werden entsprechend einer Bedingung des korre- spondierenden Interrupt-Signals selektiv aktiviert. Auf diese Weise bilden die Demultiplexer und die lo- gischen "ODER"-Glieder die Interrupt-Signale nach Maßgabe der Inhalte der Interrupt-Konfigurationsre- gister auf die Ausgänge der logischen "ODER"-Glie- der ab.

**[0016]** Jedes Aktivierungssignal entspricht einem aus der Mehrzahl von Vektoradressenregistern. Meh- rere logische "UND"-Gliedern sind so angeordnet, daß sie die Aktivierungssignale empfangen und nur dem aktivierten Aktivierungssignal mit der höchsten Priorität erlauben, das korrespondierende Exemplar aus der Mehrzahl von Vektoradressenregistern zu aktivieren. Die in dem aktivierten Vektoradres- senregister gespeicherte Adresse wird auf einen Vektoradres- senbus für den Mikroprozessor gegeben. Der Mikroprozessor ordnet dann die Vektoradresse in seinem Programmzähler an und führt, beginnend an der Vektoradresse, die passende Interrupt-Service- routine aus.

**[0017]** Sobald der Interrupt mit der höchsten Priori- tät durch die Ausführung der zugehörigen Interrupt-Serviceroutine abgearbeitet wurde, wird der an- hängige Interrupt mit der nächsten höheren Priorität abgearbeitet, bis es keine anhängigen Interrupts mehr gibt. Sobald keine anhängigen Interrupts mehr vor- handen sind, wird der Programmzähler restau- riert, so daß der Mikroprozessor die Ausführung sei- ner ursprünglichen Befehlssequenz wieder auf- nimmt.

**[0018]** Deshalb sind gemäß vorliegender Erfindung die Interrupts nach Maßgabe der Befehle aus der

Mehrzahl von Interrupt-Konfigurationsregistern und nach Maßgabe der Inhalte der Vektoradressenregister konfigurierbar. Diese Inhalte werden relativ leicht geändert, so daß die vorliegende Erfindung im Vergleich zu den Verfahren nach dem Stand der Technik Interrupts zur Verfügung stellt, die leichter konfigurierbar sind. Darüber hinaus reduziert die vorliegende Erfindung die Latenzzeit bei der Abarbeitung der einzelnen Interrupts, indem sie das Eliminieren der Priorität und das Zerlegen (Parsing) der Vektoradresse in einem Interrupt-Handler, wie es bei früheren Verfahren notwendig war, erübrigt. Statt dessen wird gemäß vorliegender Erfindung die Priorisierung der einzelnen Interrupts und die Bereitstellung der passenden Vektoradresse für den Interrupt primär in Hardware ausgeführt mit dem Ergebnis, daß die Latenzzeit kürzer ist als bei früheren Verfahren.

[0019] [Fig. 1](#) zeigt ein funktionales Blockdiagramm eines Ausführungsbeispiels der Erfindung mit einer Mehrzahl von Interrupt-Konfigurationsregistern und einer Mehrzahl von Vektoradressenregistern,

[0020] [Fig. 2A–D](#) zeigen Beispiele für die Zuteilung von Prioritäten und das Abbilden von Vektoradressen auf Interrupts nach Ausführungsbeispielen der Erfindung,

[0021] [Fig. 3](#) zeigt ein schematisches Diagramm einer Schaltung zur Implementierung eines Ausführungsbeispiels der Erfindung.

[0022] In der Anordnung nach [Fig. 1](#) sind eine Mehrzahl von Interrupt-Konfigurationsregistern **100** und eine Mehrzahl von Vektoradressenregistern **200** für die Reaktion auf Interrupt-Signale in einem mikroprozessorbasierten System vorgesehen. Jedes Interrupt-Signal INT.A bis INT.n, das von einem peripheren Gerät erzeugt wird, entspricht einem aus der Mehrzahl von Interrupt-Konfigurationsregistern **100**. Die Zahl der Interrupt-Signale variiert in Abhängigkeit von der Systemimplementierung. Jedes der Interrupt-Konfigurationsregister **100** ist mittels Software so konfigurierbar, daß es ein Bitmuster enthält, das eines der Vektoradressenregister **200** oder ein Bitmuster identifiziert, welches anzeigt, daß der entsprechende Interrupt maskiert (deaktiviert) werden soll. Dementsprechend werden die Interrupt-Signale INT.A bis INT.n nach Maßgabe der Inhalte der Register **100** jeweils selektiv auf ein passendes Exemplar der Register **200** abgebildet.

[0023] Jedes Vektoradressenregister aus der Mehrzahl von Vektoradressenregistern **200** enthält die Startadresse einer für das entsprechende Interrupt-Signal geeigneten Interrupt-Serviceroutine. In einem System, das für die Spezifizierung von Speicheradressen 32 Bits benutzt, hält jedes der Register **200** 32 Bits, obwohl die Register **200** für andere Adressierungsschemata offensichtlich eine andere

Größe haben können. Die Vektoradressenregister **200** sind nach einer vorbestimmten Priorität angeordnet. Dem Vektoradressenregister **202** ist z.B. die niedrigste Priorität zugeordnet, mittlere Register, wie das Register **204**, haben aufsteigende Prioritätsniveaus, während dem Vektoradressenregister **206** das höchste Prioritätsniveau zugeordnet ist. Die relative Prioritätsreihenfolge kann jedoch offensichtlich geändert werden.

[0024] Die Zahl der von jedem der Interrupt-Konfigurationsregister **100** gespeicherten Bits ist groß genug, um für jedes der Vektoradressenregister **200** ein eindeutiges Bitmuster bereitzustellen und ein eindeutiges Bitmuster zur Anzeige eines maskierten Interrupts bereitzustellen. Wenn es bis zu drei Interrupt-Signale gibt, hält somit jedes der Register **100** vorzugsweise zwei Bits. Das Bitmuster 00 zeigt z.B. an, daß der entsprechende Interrupt maskiert ist, während die übrigen Bitmuster 01, 10 und 11 jeweils einem der Vektoradressenregister **200** entsprechen. Wenn es bis zu sieben Interrupt-Signale gibt, speichert jedes aus der Mehrzahl von Interrupt-Konfigurationsregistern **100** vorzugsweise drei Bits. Das Bitmuster 000 zeigt z.B. an, daß der betreffende Interrupt maskiert ist. Jedes der übrigen Bitmuster 001 bis 111 entspricht einem der Vektoradressenregister **200**. Für den Fall, daß es bis zu 15 Interrupt-Signale gibt, speichert jedes aus der Mehrzahl von Konfigurationsregistern vorzugsweise vier Bits. In einem System, das noch mehr Interrupts aufweist, werden offensichtlich mehr Bits benötigt.

[0025] Die folgenden Beispiele veranschaulichen die Arbeitsweise der vorliegenden Erfindung. Es sei angenommen, daß ein mikroprozessorbasiertes System auf fünf verschiedene Interrupt-Signale reagieren soll, die mit INT.A bis INT.E bezeichnet sind. So mit muß es fünf Interrupt-Konfigurationsregister **100** geben, deren jedes drei Bits speichert. Darüber hinaus muß es fünf Vektoradressen geben, die mit A' bis E' bezeichnet sind. In den folgenden Beispielen entsprechen die Vektoradressen A' bis E' jeweils einem Interrupt, das die gleiche Buchstabenbezeichnung hat (z.B. hat die für INT.A passende Interrupt-Serviceroutine eine Startadresse in dem Speicher mit der Adresse A').

[0026] Ein erstes Beispiel ist in [Fig. 2A](#) dargestellt. Es sei angenommen, daß den Interrupts A bis E Prioritäten in der folgenden Reihenfolge (von der höchsten Priorität zur niedrigsten Priorität) zugeteilt werden sollen: C, E, A, D, B. Deshalb sind die Interrupt-Konfigurationsregister **100** und die Vektoradressenregister **200** so konfiguriert, wie dies in [Fig. 2A](#) dargestellt ist: Das Bitmuster 101 ist in dem Interrupt-Konfigurationsregister angeordnet, das dem Interrupt INT.C entspricht, und zeigt an, daß INT.C auf die höchste Priorität der Vektoradressenregister **200** abgebildet werden soll, während die Vektoradresse C' für INT.C

in dem Vektoradressenregister mit der höchsten Priorität angeordnet wird. Das Bitmuster 100 ist in dem Interrupt-Konfigurationsregister angeordnet, das dem Interrupt INT.E entspricht, und zeigt an, daß INT.E in dem Vektoradressenregister **200** mit der zweithöchsten Priorität abgebildet werden soll, während die Vektoradresse E' für INT.E in dem Vektoradressenregister mit der zweithöchsten Priorität angeordnet wird. Das Bitmuster 011 ist in dem Interrupt-Konfigurationsregister angeordnet, das dem Interrupt INT.A entspricht, und zeigt an, daß INT.A auf das Vektoradressenregister **200** mit der dritthöchsten Priorität abgebildet werden soll, während die Vektoradresse A' für INT.A in dem Vektoradressenregister mit der dritthöchsten Priorität angeordnet wird. Das Bitmuster 010 ist in dem Interrupt-Konfigurationsregister angeordnet, das dem Interrupt INT.D entspricht, und zeigt an, daß INT.D an der vierthöchsten Priorität des Vektoradressenregisters **200** abgebildet werden soll, während die Vektoradresse D' für INT.D in dem Vektoradressenregister mit der vierthöchsten Priorität angeordnet wird. Das Bitmuster 001 ist in dem Interrupt-Konfigurationsregister angeordnet, das dem Interrupt INT.B entspricht, und zeigt an, daß INT.B an der niedrigsten Priorität des Vektoradressenregisters **200** abgebildet werden soll, während die Vektoradresse B' für INT.B in dem Vektoradressenregister mit der niedrigsten Priorität angeordnet wird.

**[0027]** Wenn bei dem in [Fig. 2A](#) dargestellten Beispiel ein Interrupt auftritt, wird die entsprechende Vektoradresse dem Mikroprozessor zugeführt, um sie in dessen Programmzähler zu laden. Wenn z.B. der Interrupt INT.A auftritt, wird dem Mikroprozessor die entsprechende Vektoradresse A' zugeführt. Wenn mehrere Interrupts, z.B. die Interrupts INT.A und INT.E, anhängig sind, wird nur die Vektoradresse desjenigen der anhängigen Interrupts dem Mikroprozessor zugeführt, der die höchste Priorität hat. Da in diesem Beispiel der Interrupt INT.E eine höhere Priorität hat als der Interrupt INT.A, wird dem Mikroprozessor zuerst die Vektoradresse E' zugeführt. Sobald die Interrupt-Serviceroutine für den Interrupt INT.E ausgeführt ist, wird dem Mikroprozessor die Vektoradresse A' zugeführt, falls der Interrupt INT.A noch anhängig ist. ,

**[0028]** Es sei angenommen, daß das Bitmuster 000 einem maskierten Interrupt entspricht. Es ist jedoch offensichtlich, daß ein beliebiges Bitmuster als Entsprechung für einen maskierten Interrupt ausgewählt werden kann. Falls einer oder mehrere der Interrupts INT.A bis INT.E maskiert werden soll, werden die Bitmuster in dem Konfigurationsregister, das einem zu maskierenden Interrupt-Signal entspricht, durch das Bitmuster 000 ersetzt. [Fig. 2B](#) zeigt ein solches Beispiel. Das in [Fig. 2B](#) dargestellte Beispiel ähnelt dem Beispiel von [Fig. 2A](#) mit der Ausnahme, daß der Interrupt INT.E maskiert ist. Da der Interrupt INT.E maskiert ist, ist es nicht notwendig, eine entsprechende

Vektoradresse E' vorzusehen. Deshalb wird dem Mikroprozessor keine entsprechende Vektoradresse zugeführt, selbst wenn der Interrupt E aktiv ist. Falls einer oder mehrere der übrigen nicht maskierten Interrupts INT.A bis INT.D aktiv ist, wird der Programmfluß jedoch unterbrochen, und die passende Vektoradresse wird dem Mikroprozessor zugeführt. Auf diese Weise können jedes einzelne oder mehrere der Interrupt-Signale A bis E maskiert werden.

**[0029]** Die vorliegende Erfindung ermöglicht eine signifikante Flexibilität bei der Konfigurierung der Interrupts und ihrer relativen Prioritäten. Die Vektoradresse, die Maskierung und die relative Priorität für jeden Interrupt können durch einfache Änderung der Inhalte der entsprechenden Exemplare der Konfigurationsregister **100** oder der Vektoradressenregister **200** oder beider geändert werden.

**[0030]** Es sei noch einmal auf [Fig. 2A](#) Bezug genommen, und als Beispiel für diese Flexibilität sei angenommen, daß bei dem Abarbeiten des Interrupts mit der höchsten Priorität die relativen Prioritäten so geändert werden sollen, daß der Interrupt mit der höchsten Priorität zu dem Interrupt mit der niedrigsten Priorität wird und daß die Priorität der übrigen Interrupts inkrementiert wird. Dies wird erfindungsgemäß dadurch erreicht, daß die Einträge in den Interrupt-Konfigurationsregistern **100** und den Vektoradressenregistern **200** so geändert werden, wie dies in [Fig. 2C](#) dargestellt ist. Man erkennt, daß in [Fig. 2A](#) der Interrupt INT.C an der Stelle mit der höchsten Priorität in den Vektoradressenregistern **200** abgebildet ist, während in [Fig. 2C](#) der Interrupt INT.C in den Vektoradressenregistern **200** an der Stelle mit der niedrigsten Priorität abgebildet ist. Zusätzlich wurde in [Fig. 2C](#) die Priorität der einzelnen Interrupts INT.A, INT.B, INT.D und INT.E gegenüber ihren Positionen in [Fig. 2A](#) inkrementiert.

**[0031]** Bei den obigen Beispielen wird angenommen, daß jedes Interrupt-Signal auf eine eindeutige Interrupt-Serviceroutine abgebildet wird, es ist jedoch offensichtlich, daß mehrere Interrupts gemäß vorliegender Erfindung in einer einzigen Interrupt-Serviceroutine (oder in einem einzigen Interrupt-Handler) abgebildet werden können. [Fig. 2B](#) zeigt zwei Interrupt-Signale, die an einer einzigen Vektoradresse abgebildet sind. Im speziellen Fall sind dies das Interrupt-Signal INT.A und das Interrupt-Signal INT.E, die beide auf der Vektoradresse A' abgebildet sind.

**[0032]** [Fig. 3](#) zeigt ein schematisches Blockdiagramm einer Schaltung **300** zum Implementieren der vorliegenden Erfindung. Zur Vereinfachung der Erläuterung ist die in [Fig. 3](#) dargestellte Schaltung **300** für ein System mit fünf Interrupts INT.A bis INT.E ausgelegt, obwohl es offensichtlich ist, daß die hier offenbarten Prinzipien auch zum Aufbau einer Schaltung gemäß der Erfindung für ein System mit mehr oder

weniger Interrupts angewendet werden können. Das Interrupt-Signal INT.A ist mit dem Eingang eines Eins-zu-Acht-Multiplexers **302** verknüpft. Ein Konfigurationsregister **102** für das Interrupt-Signal INT.A speichert drei Bits. Die drei Bits des Registers **102** sind jeweils mit einer von drei Auswahlleitungen des Demultiplexers **302** verknüpft. Das Interrupt-Signal INT.B ist mit dem Eingang eines Eins-zu-Acht-Demultiplexers **304** verknüpft. Ein Konfigurationsregister **104** für das Interrupt-Signal INT.B speichert drei Bits. Die drei Bits des Registers **104** sind jeweils mit einer von drei Auswahlleitungen des Eins-zu-Acht-Demultiplexers **304** verknüpft.

**[0033]** Das Interrupt-Signal INT.C ist mit dem Eingang eines Eins-zu-Acht-Demultiplexers **306** verknüpft. Ein Konfigurationsregister **106** für das Interrupt-Signal INT.C speichert drei Bits. Die drei Bits des Registers **106** sind mit jeweils einer der drei Auswahlleitungen des Demultiplexers **306** verknüpft. Das Interrupt-Signal INT.D ist mit dem Eingang eines Eins-zu-Acht-Demultiplexers **308** verknüpft. Ein Konfigurationsregister **108** für das Interrupt-Signal INT.D speichert drei Bits. Die drei Bits des Registers **108** sind mit jeweils einer der drei Auswahlleitungen des Demultiplexers **308** verknüpft. Das Interrupt-Signal INT.E ist mit dem Eingang eines Eins-zu-Acht-Demultiplexers **310** verknüpft. Ein Konfigurationsregister **110** für das Interrupt-Signal INT.E speichert drei Bits. Die drei Bits des Registers **110** sind mit jeweils einer der drei Auswahlleitungen des Demultiplexers **310** verknüpft.

**[0034]** Der Ausgang "1" des Demultiplexers **302** ist mit dem ersten Eingang eines logischen "ODER"-Glieds **312** verbunden. Der Ausgang "2" des Demultiplexers **302** ist mit dem ersten Eingang eines logischen "ODER"-Glieds **314** verbunden. Der Ausgang "3" des Demultiplexers **302** ist mit dem ersten Eingang eines logischen "ODER"-Glieds **316** verbunden. Der Ausgang "4" des Demultiplexers **302** ist mit dem ersten Eingang eines logischen "ODER"-Glieds **318** verbunden. Der Ausgang "5" des Demultiplexers **302** ist mit dem ersten Eingang eines logischen "ODER"-Glieds **320** verbunden. Die Ausgänge "0", "6" und "7" des Demultiplexers **302** haben keine Verbindung.

**[0035]** Der Ausgang "1" des Demultiplexers **304** ist mit dem zweiten Eingang des logischen "ODER"-Glieds **312** verbunden. Der Ausgang "2" des Demultiplexers **304** ist mit dem zweiten Eingang des logischen "ODER"-Glieds **314** verbunden. Der Ausgang "3" des Demultiplexers **304** ist mit dem zweiten Eingang des logischen "ODER"-Glieds **316** verbunden. Der Ausgang "4" des Demultiplexers **304** ist mit dem zweiten Eingang des logischen "ODER"-Glieds **318** verbunden. Der Ausgang "5" des Demultiplexers **304** ist mit dem zweiten Eingang des logischen "ODER"-Glieds **320** verbunden. Die Ausgänge "0",

"6" und "7" des Demultiplexers **304** haben keine Verbindung.

**[0036]** Der Ausgang "1" des Demultiplexers **306** ist mit dem dritten Eingang des logischen "ODER"-Glieds **312** verbunden. Der Ausgang "2" des Demultiplexers **306** ist mit dem dritten Eingang des logischen "ODER"-Glieds **314** verbunden. Der Ausgang "3" des Demultiplexers **306** ist mit dem dritten Eingang des logischen "ODER"-Glieds **316** verbunden. Der Ausgang "4" des Demultiplexers **306** ist mit dem dritten Eingang des logischen "ODER"-Glieds **318** verbunden. Der Ausgang "5" des Demultiplexers **306** ist mit dem dritten Eingang des logischen "ODER"-Glieds **320** verbunden. Die Ausgänge "0", "6" und "7" des Demultiplexers **306** haben keine Verbindung.

**[0037]** Der Ausgang "1" des Demultiplexers **308** ist mit dem vierten Eingang des logischen "ODER"-Glieds **312** verbunden. Der Ausgang "2" des Demultiplexers **308** ist mit dem vierten Eingang des logischen "ODER"-Glieds **314** verbunden. Der Ausgang "3" des Demultiplexers **308** ist mit dem vierten Eingang des logischen "ODER"-Glieds **316** verbunden. Der Ausgang "4" des Demultiplexers **308** ist mit dem vierten Eingang des logischen "ODER"-Glieds **318** verbunden. Der Ausgang "5" des Demultiplexers **308** ist mit dem vierten Eingang des logischen "ODER"-Glieds **320** verbunden. Die Ausgänge "0", "6" und "7" des Demultiplexers **308** haben keine Verbindung.

**[0038]** Der Ausgang "1" des Demultiplexers **310** ist mit dem fünften Eingang des logischen "ODER"-Glieds **312** verbunden. Der Ausgang "2" des Demultiplexers **310** ist mit dem fünften Eingang des logischen "ODER"-Glieds **314** verbunden. Der Ausgang "3" des Demultiplexers **310** ist mit dem fünften Eingang des logischen "ODER"-Glieds **316** verbunden. Der Ausgang "4" des Demultiplexers **310** ist mit dem fünften Eingang des logischen "ODER"-Glieds **318** verbunden. Der Ausgang "5" des Demultiplexers **310** ist mit dem fünften Eingang des logischen "ODER"-Glieds **320** verbunden. Die Ausgänge "0", "6" und "7" des Demultiplexers **310** haben keine Verbindung.

**[0039]** Der Ausgang des logischen "ODER"-Glieds **312** ist mit einem ersten nichtinvertierenden Eingang eines logischen "UND"-Glieds **322** verbunden. Der Ausgang des logischen "ODER"-Glieds **314** ist mit einem zweiten invertierenden Eingang des logischen "UND"-Glieds **322** und mit einem ersten nichtinvertierenden Eingang eines logischen "UND"-Glieds **324** verbunden. Der Ausgang des logischen "ODER"-Glieds **316** ist mit einem dritten invertierenden Eingang des logischen "UND"-Glieds **322**, mit einem zweiten invertierenden Eingang des logischen "UND"-Glieds **324** und mit einem ersten nichtinvertier-

renden Eingang eines logischen "UND"-Glieds **326** verbunden. Der Ausgang des logischen "ODER"-Glieds **318** ist mit einem vierten invertierenden Eingang des logischen "UND"-Glieds **322**, mit einem dritten invertierenden Eingang des logischen "UND"-Glieds **324**, mit einem zweiten invertierenden Eingang des logischen "UND"-Glieds **328** und mit dem ersten nichtinvertierenden Eingang eines logischen "UND"-Glieds **328** verbunden.

**[0040]** Der Ausgang des logischen "ODER"-Glieds **320** ist mit einem invertierenden fünften Eingang des logischen "UND"-Glieds **322**, mit einem invertierenden vierten Eingang des logischen "UND"-Glieds **324**, mit einem invertierenden dritten Eingang des logischen "UND"-Glieds **326** und mit einem invertierenden zweiten Eingang des logischen "UND"-Glieds **328** verbunden und so angeordnet, daß er ein Adressenregister **212** aktiviert. Der Ausgang des logischen "UND"-Glieds **322** ist so angeordnet, daß er ein Vektoradressenregister **202** aktiviert. Der Ausgang des logischen "UND"-Glieds **324** ist so angeordnet, daß er ein Vektoradressenregister **204** aktiviert. Der Ausgang des logischen "UND"-Glieds **326** ist so angeordnet, daß er ein Vektoradressenregister **208** aktiviert. Der Ausgang des logischen "UND"-Glieds **328** ist so angeordnet, daß er ein Vektoradressenregister **210** aktiviert.

**[0041]** Die Inhalte des Vektoradressenregisters **202** werden in Abhängigkeit von dem Pegel am Ausgang des logischen "UND"-Glied **322** selektiv einem Vektoradressenbus **330** zugeführt. Die Breite des Vektoradressenbusses **330** entspricht vorzugsweise der Bitzahl, die der Mikroprozessor für die Speicheradressierung benutzt (z.B. 32 Bits), obwohl auch andere Adressierungsschemata benutzt werden können. Die Inhalte des Vektoradressenregisters **204** werden in Abhängigkeit von dem Pegel am Ausgang des logischen "UND"-Glieds **324** selektiv dem Vektoradressenbus **330** zugeführt. Die Inhalte des Vektoradressenregisters **208** werden in Abhängigkeit von dem Pegel am Ausgang des logischen "UND"-Glieds **326** selektiv dem Vektoradressenbus **330** zugeführt.

**[0042]** Die Inhalte des Vektoradressenregisters **210** werden in Abhängigkeit von dem Pegel am Ausgang des logischen "UND"-Glieds **328** selektiv dem Vektoradressenbus **330** zugeführt. Die Inhalte des Vektoradressenregisters **212** werden in Abhängigkeit von dem Pegel am Ausgang des logischen "ODER"-Glieds **320** selektiv dem Vektoradressenbus **330** zugeführt. Der Vektoradressenbus **330** ist mit einem Mikroprozessor **332** für das System verbunden und liefert geeignete Interrupt-Vektoradressen an die Logik in dem Mikroprozessor **332**, um einen Programmzähler für den Mikroprozessor **332** zu implementieren.

**[0043]** Für den Betrieb der Schaltung **300** werden

die Konfigurationsregister **102** bis **110** und die Vektoradressenregister **202** bis **212** passend konfiguriert. Die Register **102** bis **110** und **202** bis **212** sind für den Mikroprozessor **332** zugänglich und werden beim Hochfahren des mikroprozessorbasierten Systems z.B. durch eine Start-Softwareeroutine, einer Anfangskonfiguration unterzogen. Sobald das System arbeitet, können die Inhalte der Register **102** bis **110** und **202** bis **212** durch Softwarebefehle oder durch eine Benutzer-Eingabe geändert werden. Alternativ werden die Register **102** bis **110** und **202** bis **212** von logischen Schaltungen konfiguriert, die außerhalb des Mikroprozessors **332** angeordnet sind.

**[0044]** Es sei z.B. angenommen, daß die Register **102** bis **110** und **202** bis **212** entsprechend dem in **Fig. 2B** dargestellten Beispiel konfiguriert sind. Deshalb enthält das Interrupt-Konfigurationsregister **102** das Bitmuster 001, das Interrupt-Konfigurationsregister **104** enthält das Bitmuster 001, das Interrupt-Konfigurationsregister **106** enthält das Bitmuster 101, das Interrupt-Konfigurationsregister **108** enthält das Bitmuster 010, und das Interrupt-Konfigurationsregister **110** enthält das Bitmuster 000. Außerdem enthält das Vektoradressenregister **202** die Vektoradresse B', das Vektoradressenregister **204** enthält die Vektoradresse D', und das Vektoradressenregister **208** enthält die Vektoradresse A'. Die Inhalte des Vektoradressenregisters **210** sind ohne Bedeutung, da keines der Interrupt-Signale auf dem Register **210** abgebildet ist. Das Vektoradressenregister **212** ist mit der Vektoradresse C' konfiguriert.

**[0045]** Deshalb führt der Ausgang "3" des Demultiplexers **302** jedesmal, wenn das Interrupt-Signal INT.A aktiv ist, hohe logische Spannung, so daß auch am Ausgang des logischen "ODER"-Glieds **316** hohe logische Spannung auftritt. Jedesmal, wenn das Interrupt-Signal INT.B aktiv ist, führt der Ausgang "1" des Demultiplexers **304** hohe logische Spannung, so daß auch an dem Ausgang des logischen "ODER"-Glieds **312** hohe logische Spannung auftritt. Jedesmal, wenn das Interrupt-Signal INT.C aktiv ist, führt der Ausgang "5" des Demultiplexers **306** hohe logische Spannung, so daß am Ausgang des logischen "ODER"-Glieds **320** hohe logische Spannung auftritt. Jedesmal, wenn das Interrupt-Signal INT.D aktiv ist, führt der Ausgang "2" des Demultiplexers **308** hohe logische Spannung, so daß an dem Ausgang des logischen "ODER"-Glieds **314** hohe logische Spannung auftritt. Jedesmal, wenn das Interrupt-Signal INT.E aktiv ist, führt der Ausgang "0" des Demultiplexers **310** hohe logische Spannung. Weil der Ausgang "0" keine Verbindung hat, ist das Interrupt-Signal INT.E maskiert. Deshalb wirkt die Konfiguration der Demultiplexer **302** bis **310** und der logischen "ODER"-Glieder **312** bis **320** als Mapping-Schaltung für das richtige Abbilden der Interrupt-Signale INT.A bis INT.E auf die Ausgänge der logischen "ODER"-Glieder **312** bis **320** nach Maßgabe der In-

halte der Konfigurationsregister **102** bis **110**.

**[0046]** Bei dem obigen Beispiel wird unterstellt, daß die Interrupt-Signale INT.A bis INT.E aktive Hochpegel-Signale sind. Wenn sie aktive Niedrigpegel-Signale sind, ist für jedes Interrupt-Signal ein Inverter vorgesehen, um die aktiven Niedrigpegel-Signale in aktive Hochpegel-Signale umzuwandeln. Falls die Interrupt-Signale flankensensitiv sind, ist für jedes Interrupt-Signal ein Verriegelungsglied vorgesehen, um die flankensensitiven Signale in pegelsensitive Signale umzuwandeln.

**[0047]** Die Konfiguration der logischen "UND"-Glieder **322** bis **328** fungiert als Prioritätskodiererschaltung zur Priorisierung der Interrupt-Signale. Der auf das Vektoradressenregister **212** abgebildete Interrupt hat die höchste Priorität, während der auf das Vektoradressenregister **202** abgebildete Interrupt die niedrigste Priorität hat. Jedesmal, wenn ein auf den Ausgang des logischen "ODER"-Glieds **320** abgebildetes Interrupt-Signal (in dem Beispiel das Interrupt-Signal INT.C) aktiv ist, blockiert der Ausgang des logischen "ODER"-Glied **320** über die logischen "UND"-Glieder **322** bis **328** alle anderen Interrupt-Signale und bewirkt, daß die in dem Register **212** enthaltene Vektoradresse (im Beispiel die Vektoradresse C') auf dem Vektoradressenbus **330** plaziert wird.

**[0048]** Wenn man annimmt, daß keines der Interrupt-Signale mit höherer Priorität aktiv ist, wenn ein auf einen der Ausgänge der "ODER"-Glieder **312** bis **318** abgebildetes Interrupt-Signal aktiv ist, blockiert das "ODER"-Ausgangssignal des betreffenden "ODER"-Glieds alle Interrupt-Signale mit niedrigerer Priorität über die logischen "UND"-Glieder, die den Interrupt-Signalen mit niedrigerer Priorität zugeteilt sind, und bewirkt, daß die Vektoradresse, die dem aktiven Interrupt-Signal mit der höchsten Priorität entspricht, auf dem Vektoradressenbus **330** plaziert wird.

**[0049]** Sobald der Interrupt mit der höchsten Priorität durch Ausführen der passenden Interrupt-Serviceroutine abgearbeitet ist, wird das Interrupt-Signal deaktiviert, und der anhängige Interrupt mit der nächsten höheren Priorität wird abgearbeitet, bis es keine anhängigen Interrupts mehr gibt. Sobald es keine anhängigen Interrupts mehr gibt, wird der Programmzähler zurückgesetzt, so daß der Mikroprozessor die Ausführung des Softwareprogramms wieder aufnimmt.

**[0050]** Gemäß vorliegender Erfindung sind die Interrupt-Signale nach Maßgabe der Inhalte in der Mehrzahl von Interrupt-Konfigurationsregister **102** bis **110** und nach Maßgabe der Inhalte der Vektoradressenregister **202** bis **212** konfigurierbar. Diese Inhalte werden relativ leicht geändert, so daß die vorliegende Erfindung Interrupts zur Verfügung stellt, die

im Vergleich zu früheren Verfahren leichter konfigurierbar sind.

**[0051]** Die vorliegende Erfindung wurde anhand spezifischer Ausführungsbeispiele beschrieben, die Details enthalten, um das Verständnis der Konstruktions- und Funktionsprinzipien der Erfindung zu erleichtern. Die Bezugnahme auf spezifische Ausführungsbeispiele und deren Details soll den Rahmen der anliegenden Ansprüche nicht einschränken. Es ist für den einschlägigen Fachmann offensichtlich, daß an dem für die Erläuterung ausgewählten Ausführungsbeispiel Modifikationen vorgenommen werden können, ohne daß dadurch der Rahmen der Erfindung verlassen wird, wie sie beansprucht wird. Insbesondere ist es für den einschlägigen Fachmann offensichtlich, daß die Vorrichtung gemäß vorliegender Erfindung auch in anderer Weise implementiert werden könnten und das oben offenbarte Gerät lediglich das bevorzugte Ausführungsbeispiel der Erfindung veranschaulicht und keinesfalls eine Beschränkung darstellt. Es ist z.B. offensichtlich, daß das in Fig. 3 dargestellte Gerät modifiziert oder durch Hinzufügungen ergänzt werden kann. Es können insbesondere logische Schaltungen enthalten sein, die die Interrupt-Signale INT.A, INT.B, INT.C, INT.D und INT.E verriegeln und die prüfen, ob die einzelnen Interrupt-Signale vorderflanken-, hinterflanken- oder pegelsensitiv sind. Weiterhin können auch alternative Verfahren zum Maskieren der Interrupts implementiert werden, z.B. durch selektive Steuerung eines Chip-Selekt-Anschlusses für jeden der Demultiplexer **302**, **304**, **306**, **308** und **310**.

### Patentansprüche

1. Verfahren zum Antworten auf aktive Exemplare aus einer Mehrzahl von Interrupt-Signalen in einem mikroprozessorbasierten System, wobei das Verfahren die Verfahrensschritte umfaßt:

- Empfangen der aktiven Exemplare aus der Mehrzahl von Interrupt-Signalen,
- Abilden jedes der aktiven Exemplare aus der Mehrzahl von Interrupt-Signalen entsprechend den Inhalten eines entsprechenden Exemplars aus einer ersten Mehrzahl von Registern (**100**) auf eines aus einer zweiten Mehrzahl von Registern (**200**), wobei jedes Register der zweiten Mehrzahl von Registern eine entsprechende, durch das betreffende Exemplar der ersten Mehrzahl von Registern spezifizierte relative Priorität hat und eine Vektoradresse für das entsprechende Interrupt-Signal enthält, und
- Liefern der Vektoradresse für jedes der aktiven Exemplare der Interrupt-Signale an den Mikroprozessor nach Maßgabe der relativen Priorität jedes der aktiven Exemplare der Interrupt-Signale.

2. Verfahren nach Anspruch 1 mit dem weiteren Verfahrensschritt des Änderns der relativen Priorität der Interrupt-Signale durch Ändern der Inhalte der

ersten und der zweiten Mehrzahl von Registern.

3. Verfahren nach Anspruch 1 mit dem weiteren Verfahrensschritt des Maskierens eines ausgewählten Exemplars aus der Mehrzahl von Interrupts nach Maßgabe der Inhalte der betreffenden Exemplare aus der ersten Mehrzahl von Registern.

4. Verfahren nach Anspruch 1, bei dem eine Vektoradresse mehr als einem Interrupt-Signal entspricht.

5. Gerät zum Antworten auf aktive Exemplare aus einer Mehrzahl von Interrupt-Signalen in einem mikroprozessorbasierten System, wobei das Gerät aufweist:

- a. eine erste Mehrzahl von Registern (**100**),
- b. eine mit der ersten Mehrzahl von Registern verbundene Mapping-Schaltung (**302, 304, 306, 308, 310, 312, 314, 316, 318, 320**) zum Abbilden jedes der aktiven Exemplare aus der Mehrzahl von Interrupt-Signalen auf eines aus einer Mehrzahl von Aktivierungssignalen auf Prioritätsbasis nach Maßgabe der Inhalte eines entsprechenden Exemplars aus der ersten Mehrzahl von Registern.
- c. eine mit der Mapping-Schaltung verbundene Prioritätskodierschaltung (**322, 324, 326, 328**) zum Blockieren aller aus der Mehrzahl von Aktivierungssignalen mit Ausnahme des Aktivierungssignals mit der höchsten Priorität und
- d. eine zweite Mehrzahl von Registern (**200**), die so angeordnet sind, daß sie die Mehrzahl von Aktivierungssignalen aufnehmen, wobei jedes Register aus der zweiten Mehrzahl von Registern eine Vektoradresse für das entsprechende Aktivierungssignal enthält und die zweite Mehrzahl von Registern dazu dient, die dem Aktivierungssignal mit der höchsten Priorität entsprechende Vektoradresse an den Mikroprozessor zu liefern.

6. Gerät nach Anspruch 5, bei dem die relative Priorität der Interrupt-Signale durch Ändern der Inhalte der ersten und der zweiten Mehrzahl von Registern geändert wird.

7. Gerät nach Anspruch 5, bei dem ausgewählte Exemplare aus der Mehrzahl von Interrupts durch die Mapping-Schaltung nach Maßgabe der Inhalte des betreffenden Exemplars aus der ersten Mehrzahl von Registern maskiert werden.

8. Gerät nach Anspruch 5, bei dem eine Vektoradresse mehr als einem Interrupt-Signal entspricht.

9. Gerät nach Anspruch 5, bei dem die Prioritätskodierschaltung eine Mehrzahl von logischen UND-Gliedern (**322, 324, 326, 328**) aufweist und jedes logische UND-Glied einem der Aktivierungssignale entspricht, welches nicht das Aktivierungssignal mit der höchsten Priorität ist, wobei jedes logische

UND-Glied so angeordnet ist, daß eine logische UND-Funktion an einem logischen Pegel des entsprechenden Exemplars der Aktivierungssignale und an einem invertierten logischen Pegel jedes Aktivierungssignals ausführt, das eine höhere Priorität hat als das entsprechende Exemplar der Aktivierungssignale, und wobei das Ausgangssignal jedes logischen UND-Glieds und die höchste Priorität der Aktivierungssignale jeweils so gekoppelt sind, daß sie ein entsprechendes Exemplar aus der zweiten Mehrzahl von Registern in die Lage versetzen, seine Vektoradresse an den Mikroprozessor liefern.

10. Gerät nach Anspruch 5, bei dem die erste Mehrzahl von Registern einen relativen Prioritätswert speichern und jedes Register aus der ersten Mehrzahl von Registern einem aus der Mehrzahl von Interrupt-Signalen entspricht.

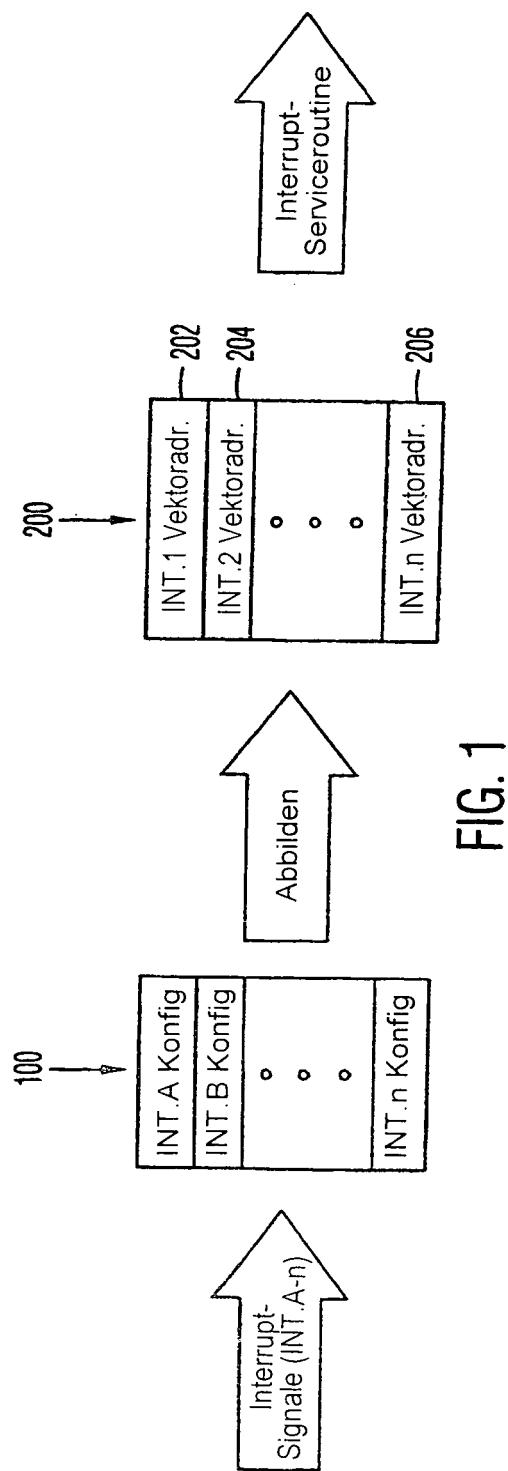
11. Gerät nach Anspruch 5, bei dem die Mapping-Schaltung ferner eine Mehrzahl von Demultiplexern (**302, 304, 306, 308, 310**) aufweist und jeder Demultiplexer eine Mehrzahl von Auswahlleitungen besitzt, die mit einem entsprechenden Exemplar aus der ersten Mehrzahl von Registern verbunden sind, um einen relativen Prioritätswert zu empfangen, ferner einen Eingang, der so angeordnet ist, daß er das entsprechende Exemplar der Interrupt-Signale aufnimmt, sowie eine Mehrzahl von Ausgängen.

12. Gerät nach Anspruch 11, bei dem die Mapping-Schaltung ferner eine Mehrzahl von logischen ODER-Gliedern (**312, 314, 316, 318, 320**) aufweist und jedes logische ODER-Glied eine Mehrzahl von Eingängen und einen Ausgang besitzt, wobei jeder Eingang jedes logischen ODER-Glieds mit einem aus der Mehrzahl von Ausgängen jedes Demultiplexers verbunden ist und wobei die Ausgänge der logischen ODER-Glieder die Mehrzahl von Aktivierungssignalen erzeugen.

13. Gerät nach Anspruch 12, bei dem ausgewählte Exemplare aus der Mehrzahl von Interrupts durch geeignetes Konfigurieren der Inhalte des betreffenden Exemplars aus der ersten Mehrzahl von Registern maskiert werden, um einen Ausgang des entsprechenden Demultiplexers auszuwählen, der nicht mit einem aus der Mehrzahl von logischen ODER-Gliedern verbunden ist.

Es folgen 7 Blatt Zeichnungen

## Anhängende Zeichnungen



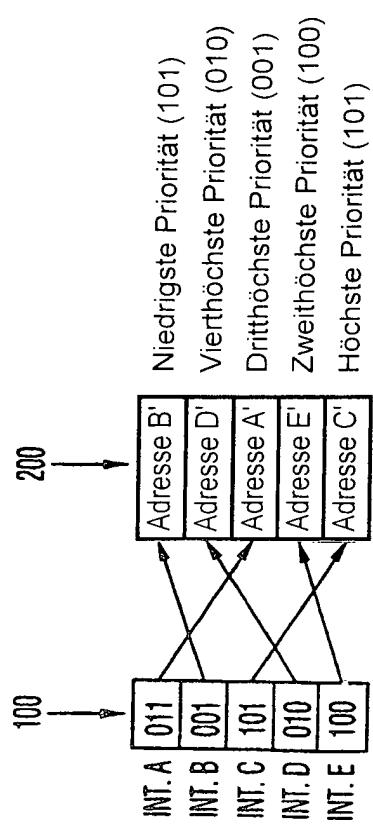


FIG. 2A

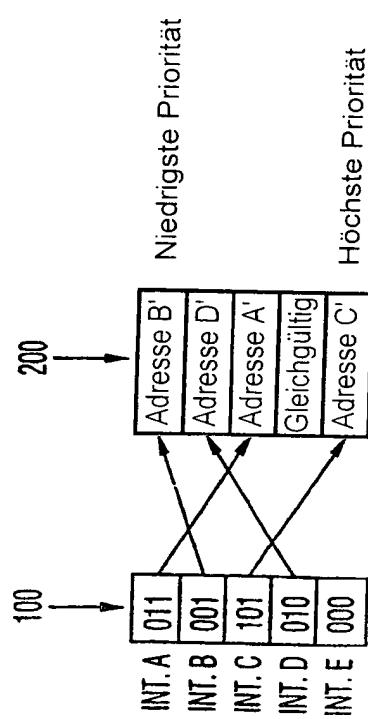


FIG. 2B

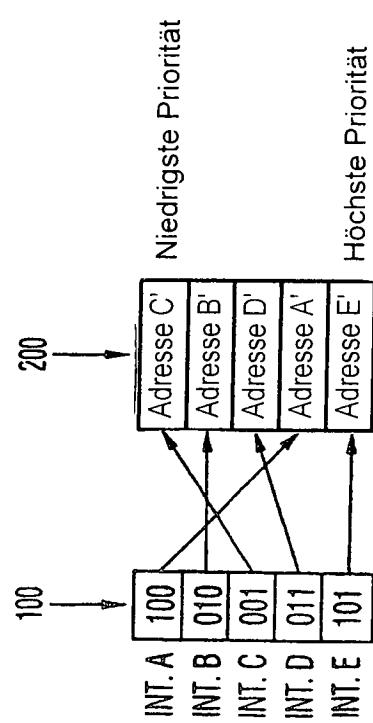


FIG. 2C

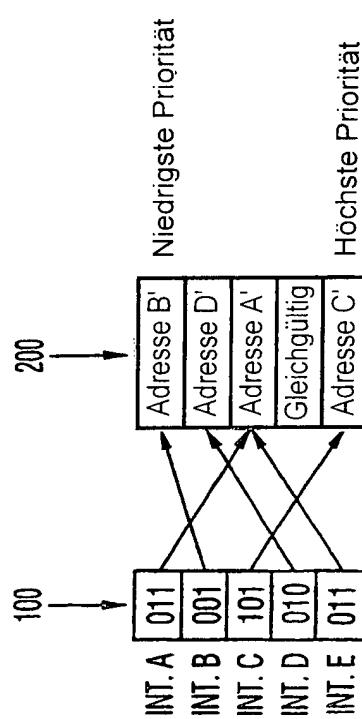


FIG. 2D

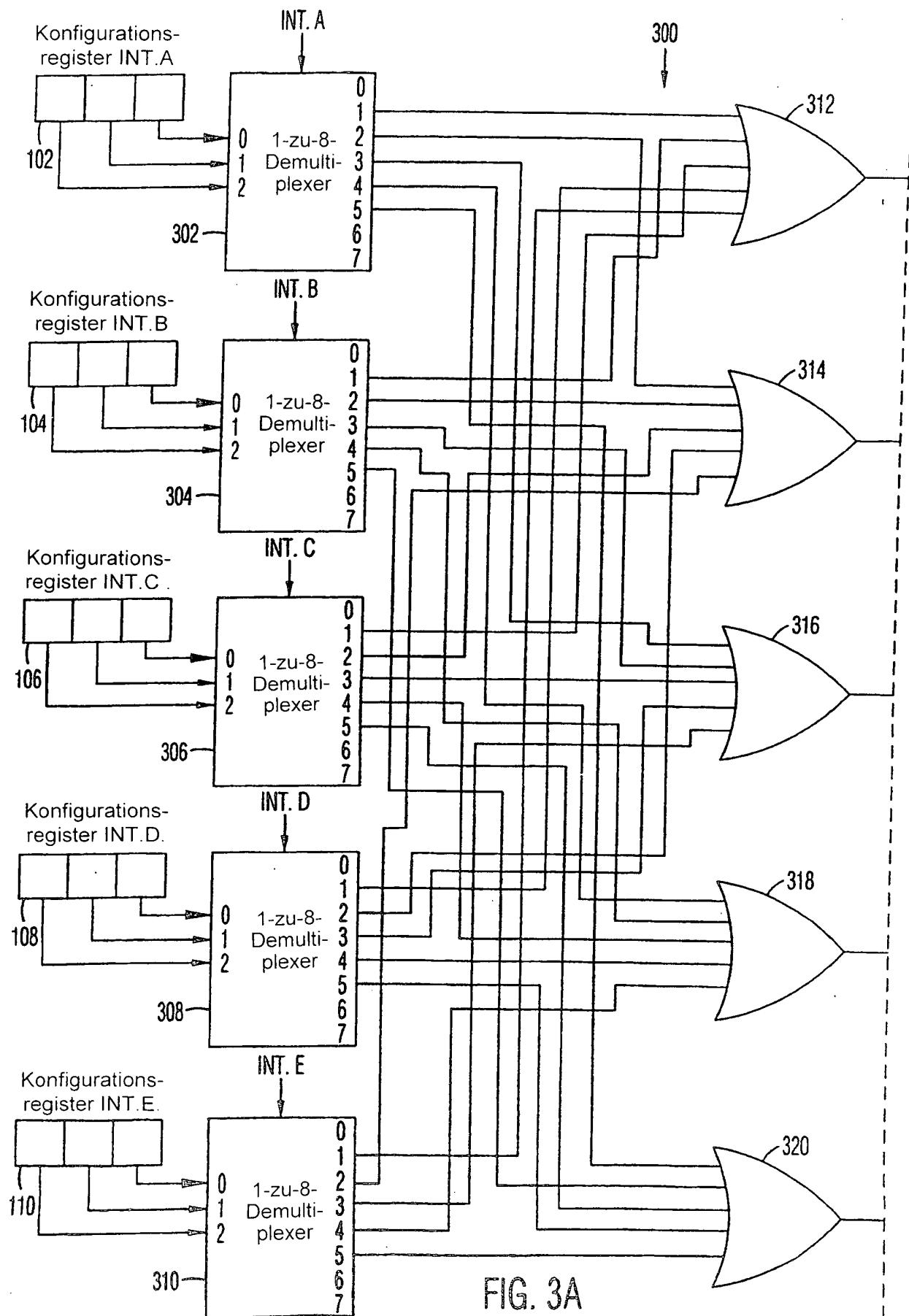


FIG. 3A

