

AMT FÜR ERFINDUNGS- UND PATENTWESEN

In der vom Anmelder eingereichten Fassung veröffentlicht

(21) WP G 06 F / 321 215 8

(22) 31.10.88

(44) 28.02.90

(31) VEB Robotron-Elektronik Dresden, Postfach 240, Karl-Marx-Stadt, 9010, DD

(72) Meißelbach, Alexander, Dipl.-Ing.; Linse, Volker, DD

(54) Schaltungsanordnung zur Kopplung eines Einchipmikrorechners mit einem Mikrorechner

(55) Einchiprechner, Prozessorkopplung, Rechnerkopplung, Rechnerzusammenarbeit, direkte Datenbusverbindung, Ausnutzung, Steuereingänge

(57) Die Erfindung betrifft eine Schaltungsanordnung zur Kopplung eines Einchipmikrorechners (EMR) mit einem Mikrorechner (U880) auf der Basis des U880-Systems, bei der eine bidirektionale parallele Datenübertragung gesteuert über Interrupt möglich ist. Dabei ist der Datenbus beider Prozessoren direkt verbunden. Über vier Steuerleitungen wird gewährleistet, daß nach Auslösung einer Interruptaufforderung der U880 durch Steuerung des EMR für den Moment der Datenausgabe (U880 → EMR) bzw. der Dateneingabe (EMR → U880) in den Schrittbetrieb gezwungen wird. Fig. 1

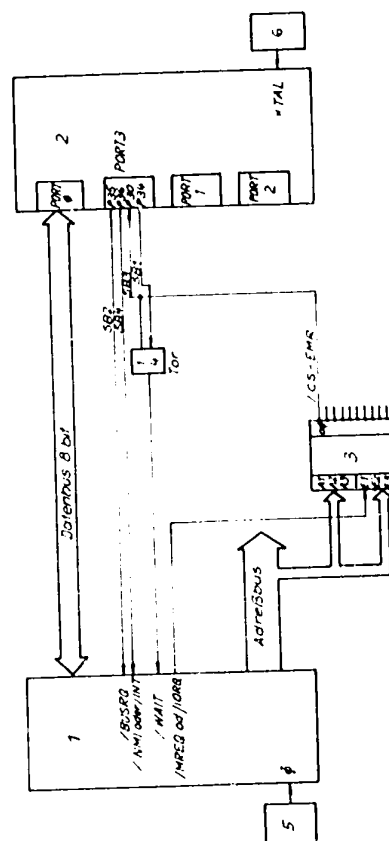


Fig. 1

Patentanspruch:

Schaltungsanordnung zur Kopplung eines Einchipmikrorechners mit einem Mikrorechner, die über ihren Datenbus zur Datenübertragung direkt verbunden sind, und einem Dekoder zur Selektierung des Speicher- bzw. E/A-Bereiches, dadurch gekennzeichnet, daß zwei Ausgabebits (SB 2, SB 4) des EMR (2) auf die Steuereingänge (/NMI und /BUSRQ) des Mikrorechners (1) geführt sind, daß ein weiteres Ausgabebit (SB 1) auf ein Tor (4) geschaltet ist, dessen Ausgang auf den /WAIT-Eingang des Mikrorechners (1) geführt ist, wobei dessen Freigabesignal (/MREQ oder /IORQ) zusammen mit Adreßsignalen über den Dekoder (3) ein Freigabesignal (/CS-EMR) bildet, welches auf das Tor (4) und einen interruptfähigen Eingang (SB3) des EMR (2) geführt ist.

Hierzu 3 Seiten Zeichnungen

Anwendungsgebiet der Erfindung

Die Erfindung betrifft eine Schaltungsanordnung zur Kopplung eines Einchipmikrorechners mit einem Mikrorechner auf der Basis des U880-Systems.

Charakteristik des bekannten Standes der Technik

Es ist bekannt, daß für die Kopplung zweier Prozessoren mehrere Möglichkeiten existieren. Alle bekannten technischen Lösungen ist die Verwendung eines Zwischenspeichermittels, welches über eine entsprechende Logik angesteuert wird, zur Kopplung gemeinsam. So erfolgt die Kopplung meistens über entsprechende E/A-Tore im parallelen oder auch seriellen Datenaustausch oder über einen gemeinsamen Speicherbereich, auf den beide Prozessoren gesteuert über eine entsprechende Auswahllogik zugreifen können.

So erfolgt im DD-WP 214011 eine Mehrrechnerkopplung über E/A-Schaltkreise der jeweiligen Rechnerfamilien. Die zur Kopplung notwendigen E/A-Schaltkreise sowie die erforderliche Logik zur Ansteuerung (Auswahlsignale usw.) ergeben einen zusätzlichen Hardwareaufwand.

Im DD-WP 234513 ist eine Kopplung zwischen Einchipmikrorechner der U881-Familie und einem Mikroprozessor auf Basis des U880 dargestellt. Die Kopplung der beiden Prozessoren erfolgt hier nicht direkt, sondern über einen Speicher, auf den beide (oder auch mehrere) Prozessoren zugreifen können. Die Steuerung des Speicherzugriffs erfolgt über eine Adreßbusschaltung, die aus mehreren Registern (z. B. Adreßauffangregister, Adreßstroberegister) sowie einer Anzahl von Gattern und Flipflops besteht. Auch in dieser Lösung ist ein relativ hoher Hardwareaufwand erforderlich.

Eine Kopplung von zwei Mikroprozessoren über ein DMA-Steuerwerk ist im DD-WP 237725 dargestellt. Zweck dieser Erfindung ist die Erzielung einer minimalen Rechenzeitbelastung der Prozessoren für den Datenaustausch untereinander, was allerdings ebenfalls einen erheblichen Hardwareaufwand erfordert.

In der Zeitschrift rf 5/1987 ist ein Koppelmodul zur Kopplung zweier Mikrorechner auf Basis des U880 vorgestellt. Der Datenaustausch erfolgt hierbei über einen Speicher, auf den beide Prozessoren zugreifen können.

Für die Steuerung des Speicherzugriffs ist ein Koppelmodul erforderlich, das neben dem eigentlichen Koppelspeicher die Ansteuerlogik enthält, die aus mehreren Registern für Daten und Adressen sowie einer Logik zur Speicheransteuerung besteht. Allen vorgestellten Lösungen ist ein erheblicher zusätzlicher Hardwareaufwand zur Kopplung von Mikroprozessoren gemeinsam.

Ziel der Erfindung

Es ist Ziel der Erfindung, eine Kopplung zwischen einem Einchipmikrorechner und einem U880 zu schaffen, die ohne zusätzlichen Hardwareaufwand auskommt.

Darlegung des Wesens der Erfindung

Der Erfindung liegt die Aufgabe zugrunde, zwei Mikrorechner so zu koppeln, daß eine bidirektionale parallele Datenübertragung gesteuert über Interrupt möglich ist. Dabei ist der Datenbus beider Prozessoren direkt verbunden.

Erfindungsgemäß wird die Aufgabe durch die im Kennzeichen des Anspruchs dargelegten Merkmale gelöst.

Das Wesen der Erfindung ist darin zu sehen, daß die Datenübertragung in der Art erfolgt, daß nach Auslösung einer Interruptanforderung der U880 durch Steuerung des Einchipmikrorechners für den Moment der Datenausgabe (bei Übertragung U880 - Einchipmikrorechner) bzw. der Dateneingabe (bei Übertragung Einchipmikrorechner - U880) in den Schrittbetrieb gezwungen wird. Somit ist eine direkte Kopplung beider Prozessoren ohne zwischengeschaltete Auffangregister möglich.

Der Vorteil der vorgeschlagenen Kopplung ist darin zu sehen, daß sie über vier Steuerleitungen ohne zusätzlichen Hardwareaufwand erfolgt, wobei 8bit parallel in beide Richtungen übertragen werden können.

Ausführungsbeispiel

Die Erfindung soll nachstehend an einem Beispiel erläutert werden. In den zugehörigen Zeichnungen zeigen:

Fig. 1: Zusammenschaltung U880 mit Einchipmikrorechner

Fig. 2: Zeitablauf einer Übertragung U880 → Einchipmikrorechner

Fig. 3: Zeitablauf einer Übertragung Einchipmikrorechner → U880.

In Fig. 1 ist die Kopplung zwischen einem Mikrorechner (U880) 1 und einem Einchipmikrorechner (EMR) 2 zur bidirektionalen Übertragung von 8 Bit dargestellt. Dabei wird der Mikrorechner 1 mit einer Takterzeugung 5 und der EMR 2 mit einer Takterzeugung 6 beschaltet. Beide Taktsysteme müssen nicht synchron laufen. Ein 8-bit-Port des EMR 2 ist mit dem Datenbus des U880 1 direkt verbunden. Weiterhin sind zwei Ausgabebits des EMR 2 mit den Steuerleitungen /NMI und /BUSRQ des U880 1 verbunden. Über einen Dekoder 3 wird durch Beschaltung mit einer gewünschten Adreßkombination und dem Freigabesignal /MREQ oder /IORQ das Signal /CS-EMR gebildet, welches auf ein Tor 4 und einen interruptfähigen Eingang des EMR 2 geführt ist. Der zweite Eingang des Tores 4 wird mit einem Ausgang des EMR 2 belegt. Der Ausgang des Tores 4 führt auf die Steuerleitung /WAIT des U880 1.

Der Dekoder 3 bedeutet keinen zusätzlichen Hardwareaufwand, da in Mikrorechnersystemen solche Dekoder zur Selektierung des Speicher- bzw. E/A-Bereiches verwendet werden, worin die Generierung des Signales /CS-EMR eingeschlossen werden kann. Eine Ansteuerung der Leitungen /WAIT und /BUSRQ von anderen Quellen ist möglich, wenn für das Tor 4 ein Gatter mit Open-Collector-Ausgang Anwendung findet sowie der entsprechende Ausgang des EMR 2 als Open-Drain konfiguriert wird, was allerdings nur mit einem Ausgang des Port 2 des EMR 2 möglich ist. Das Signal /NMI darf allerdings keine weitere Quelle haben, da eine Unterscheidung in der INT-Behandlung im U880 1 bei nicht maskiertem INT nicht möglich ist. Prinzipiell ist auch die Nutzung des /INT-Eingangs möglich, allerdings muß der U880 1 dann im INT-Mode 1 betrieben werden. Der Datenport am EMR 2 ist bidirektional, Steuerbit 3 ist ein INT-fähiger Eingang (bei 1/0-Flanke) und die Steuerbits (SB) 1, 2 und 4 sind Ausgänge.

Der Datenport am EMR 2 (Port 0) ist auf Eingabe programmiert. Der Grundzustand an den EMR-Ausgängen ist SB 1 = 0, SB 2 = SB 4 = 1. SB 3 ergibt sich entsprechend dem inaktiven Zustand des Signales /CS-EMR zu 1.

Im folgenden wird eine bidirektionale Datenübertragung beschrieben.

In Fig. 2 ist der zeitliche Ablauf einer Übertragung von U880 1 zum EMR 2 dargestellt. Der U880 1 führt einen Schreibbefehl auf die EMR-Adresse aus. Damit wird über den Dekoder 3 das Signal /CS-EMR aktiviert, was den U880 über Tor 4 in den WAIT-Zustand versetzt und gleichzeitig über Steuerbit 3 am EMR 2 ein Interrupt auslöst.

Der EMR 2 verzweigt in die entsprechende INT-Routine und liest den Datenport ein. Damit sind die anliegenden Daten übernommen. SB 2 wird 0 gesetzt und anschließend SB 1 = 1. Damit wird der U880 1 in BUSRQ versetzt und /WAIT weggenommen. Der U880 1 beendet somit den laufenden Maschinenzyklus und bleibt vor dem Takt T1 des nächsten Maschinenzyklus stehen. SB 1 wird wieder 0 gesetzt und damit für den nächsten Datentransfer aktiviert. Mit Setzen von SB 4 = 1 ist die Übertragung beendet und /BUSRQ wird vom U880 1 weggenommen.

Durch die Steuerung der Übertragung vom EMR 2 ist die Nutzung des OTI/OTIR-Befehls vom U880 1 möglich.

Die Zeit des Schrittbetriebes des U880 1 ist bestimmt von der Zeit bis zur INT-Anerkennung des EMR 2 plus Abarbeitungszeit der INT-Routine bis zum Setzen von SB 2 = 1. Beim Anschluß dynamischer RAMs am U880-System ist die Zeit des ausgesetzten Refreshes unter Umständen zu beachten. Der INT-Eingang am EMR 2, d. h. Steuerbit 3, sollte deshalb als höchstpriorisierter programmiert werden.

In Fig. 3 ist der zeitliche Ablauf beider Prozessoren für eine Übertragung vom EMR 2 zum U880 1 dargestellt.

Über einen L-Impuls an SB 4 wird eine Datensendeanforderung vom EMR 2 an den U880 1 gemeldet. Im EMR 2 wird ein Bit gesetzt, das die Ausgabeforderung registriert. Falls im gleichen Moment der INT-Auslösung zum U880 1 dieser selbst ein Byte zum EMR 2 ausgibt und sich im WAIT befindet, ist so im EMR 2 eine Unterscheidung in der INT-Routine möglich, ob eine Ausgabe von Daten erfolgen kann oder nicht. Der U880 1 verzweigt in die entsprechende INT-Routine (INT oder NMI) und arbeitet diese bis zum Lesebefehl der EMR-Adresse ab und versetzt sich über das Tor 4 in den WAIT-Zustand. Gleichzeitig wird über SB 3 dem EMR 2 über INT die Lesebereitschaft des U880 1 signalisiert. Falls das Ausgabebit im EMR 2 nicht gesetzt ist, wird der INT als Datenübertragung von U880 1 zum EMR 2 gewertet und entsprechend abgearbeitet. Sonst wird der Datenport am EMR 2 auf Ausgabe programmiert und die Daten ausgegeben.

Mit Setzen von SB 2 = 0 und SB 1 = 1 beendet der U880 1 wie bereits oben beschrieben den Lesebefehl. Der EMR-Datenport wird wieder auf Eingabe programmiert. Anschließend wird SB 1 = 0 gesetzt. Mit Setzen von SB 4 = 1 wird BUSRQ weggenommen, und beide Prozessoren arbeiten normal weiter. Im EMR 2 wird das Ausgabeanforderungsbit rückgesetzt. Die Zeit des Schrittbetriebes des U880 1 ist ebenfalls bestimmt von der Zeit bis zur INT-Annahme des EMR 2 plus Abarbeitungszeit der INT-Routine bis zum Setzen von SB 2 = 1. Die Ausgabe von U880 1 zum EMR 2 ist höher priorisiert als die umgekehrte Richtung.

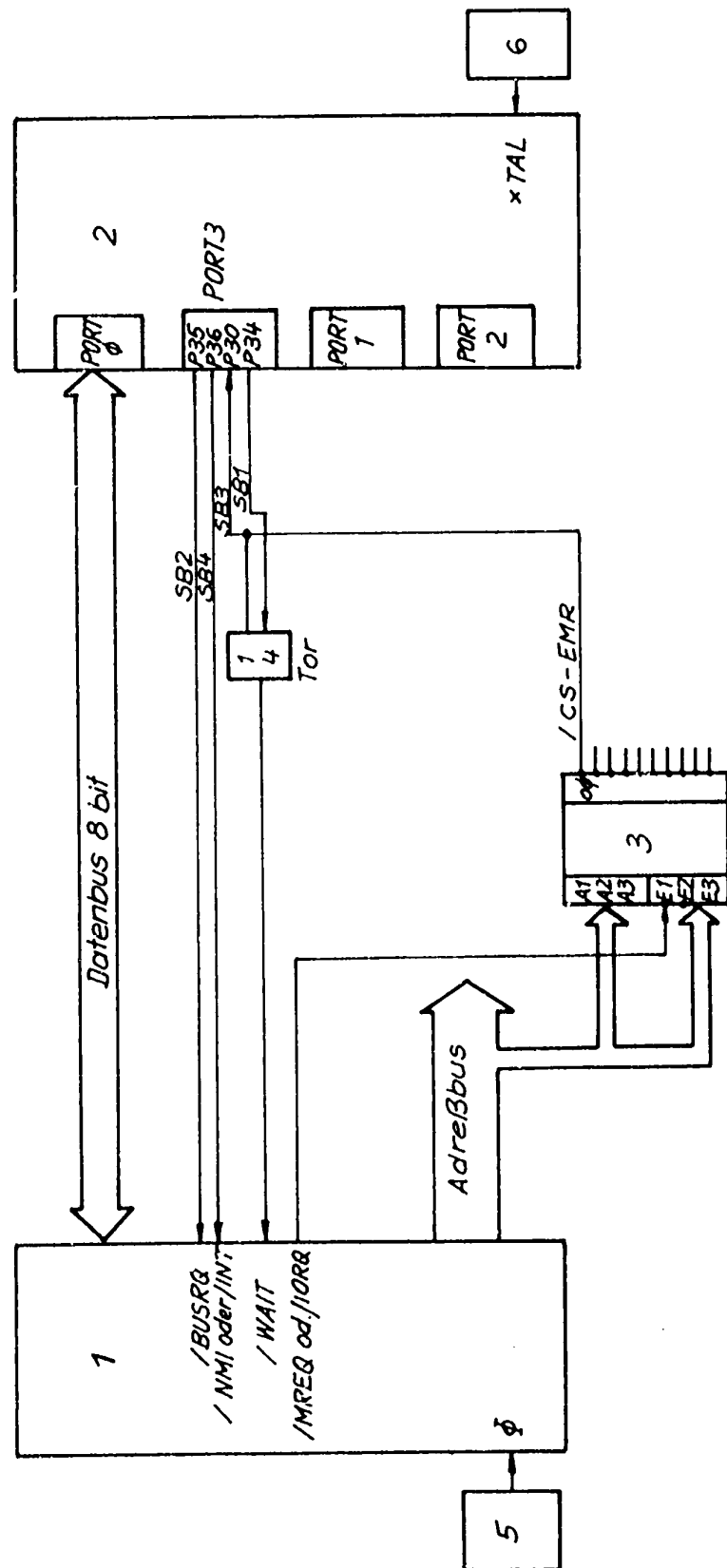
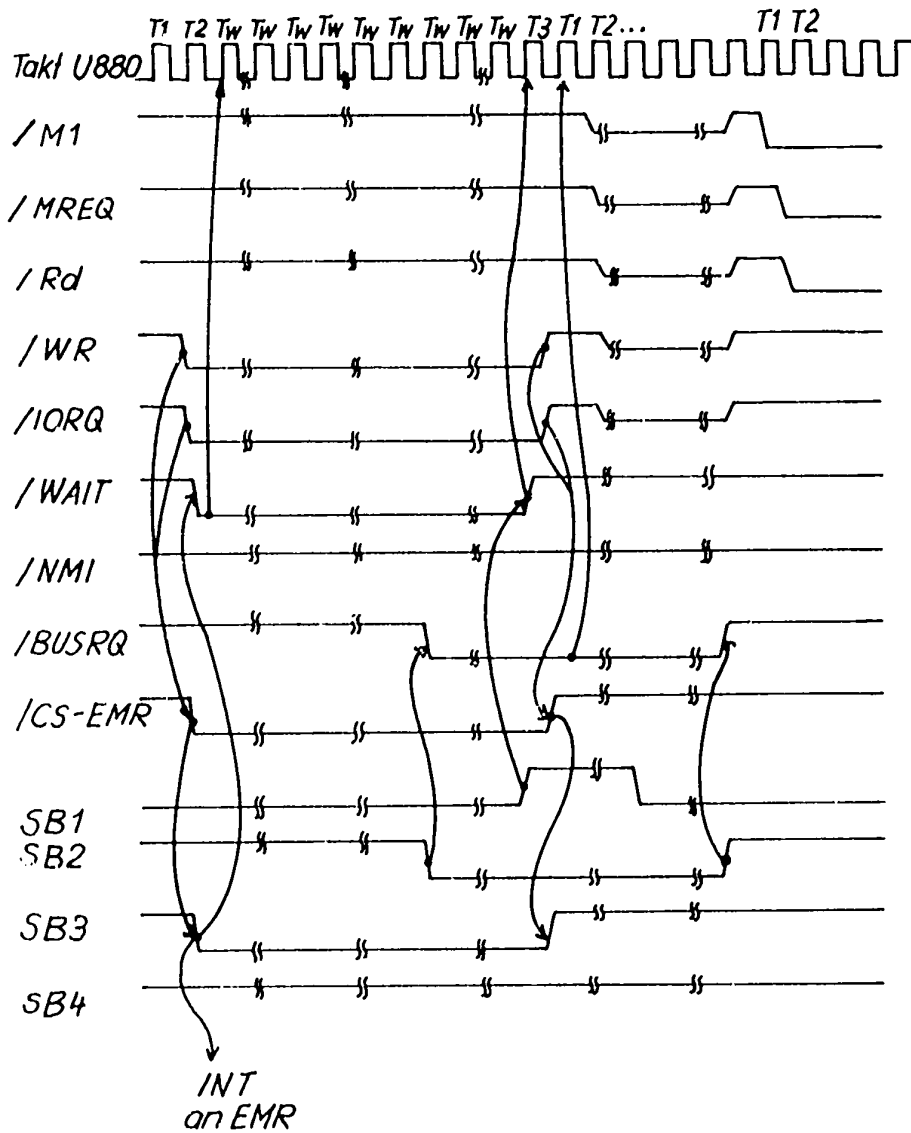


Fig. 1



U880: Ausgabe Datenbyte — Tristate — nächster M1

EMR: $\downarrow$ INT

Einlesen Datenbyte

Setzen P35 = 0

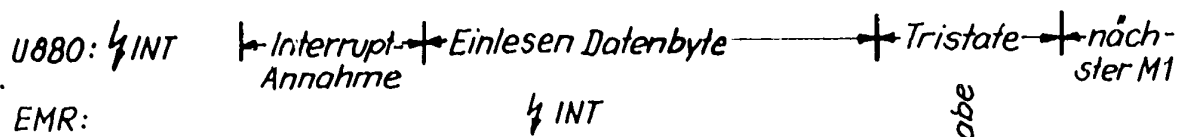
Setzen P34 = 1

Setzen P34 = 0

Setzen P35 = 1

I RET

Fig. 2



Setzen $p_{36} = q$

Setzen $P_{36} = 1$

Datenausgabe

Setzen $P_{35} = \emptyset$

Setzen P34-1

Setzen $P_{34} = \emptyset$
u. Datenpart auf Eingabe

Setzen $P_{35} = 1$

1381

Fig. 3