

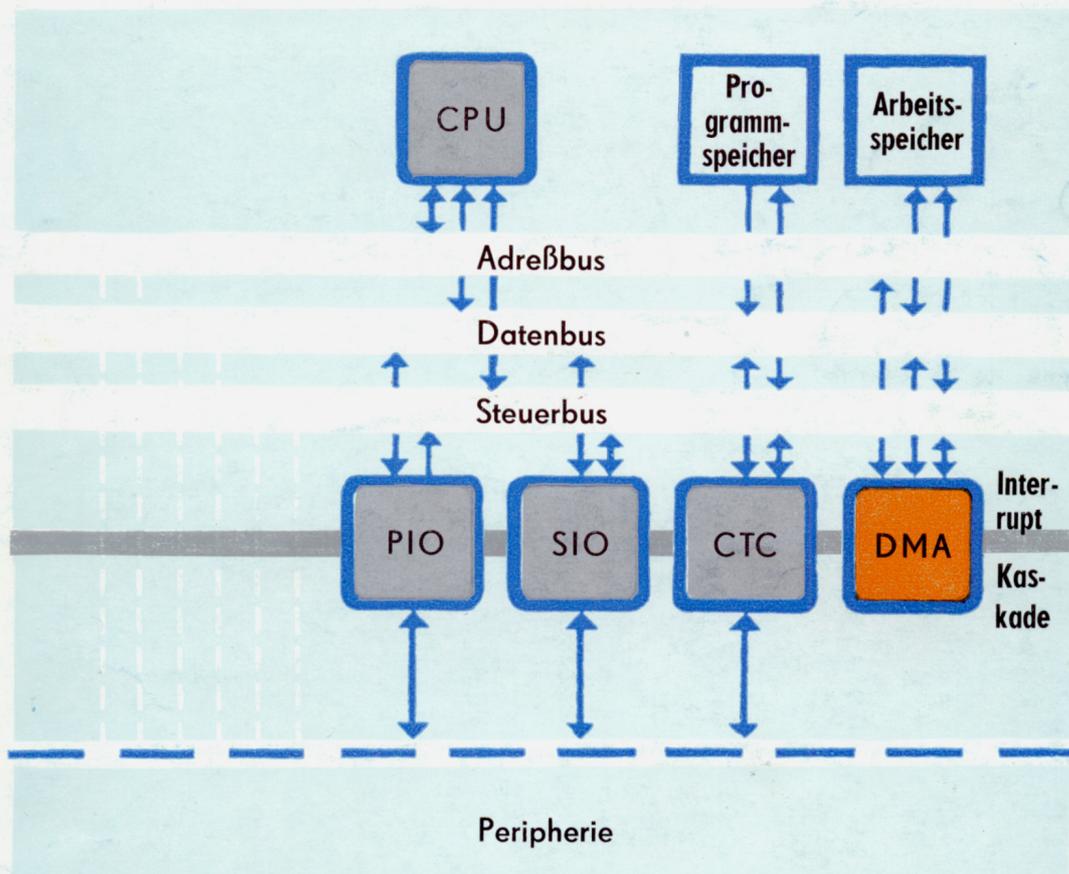
Mühlbauer



# Mikroprozessorsystem

der II. Leistungsklasse

## – Technische Beschreibung –



Schaltkreis für direkten Speicherzugriff

DMA

U 858 D

# Schaltkreis für direkten Speicherzugriff DMA U858 D

**veb mikroelektronik › karl marx › erfurt  
stammbetrieb**

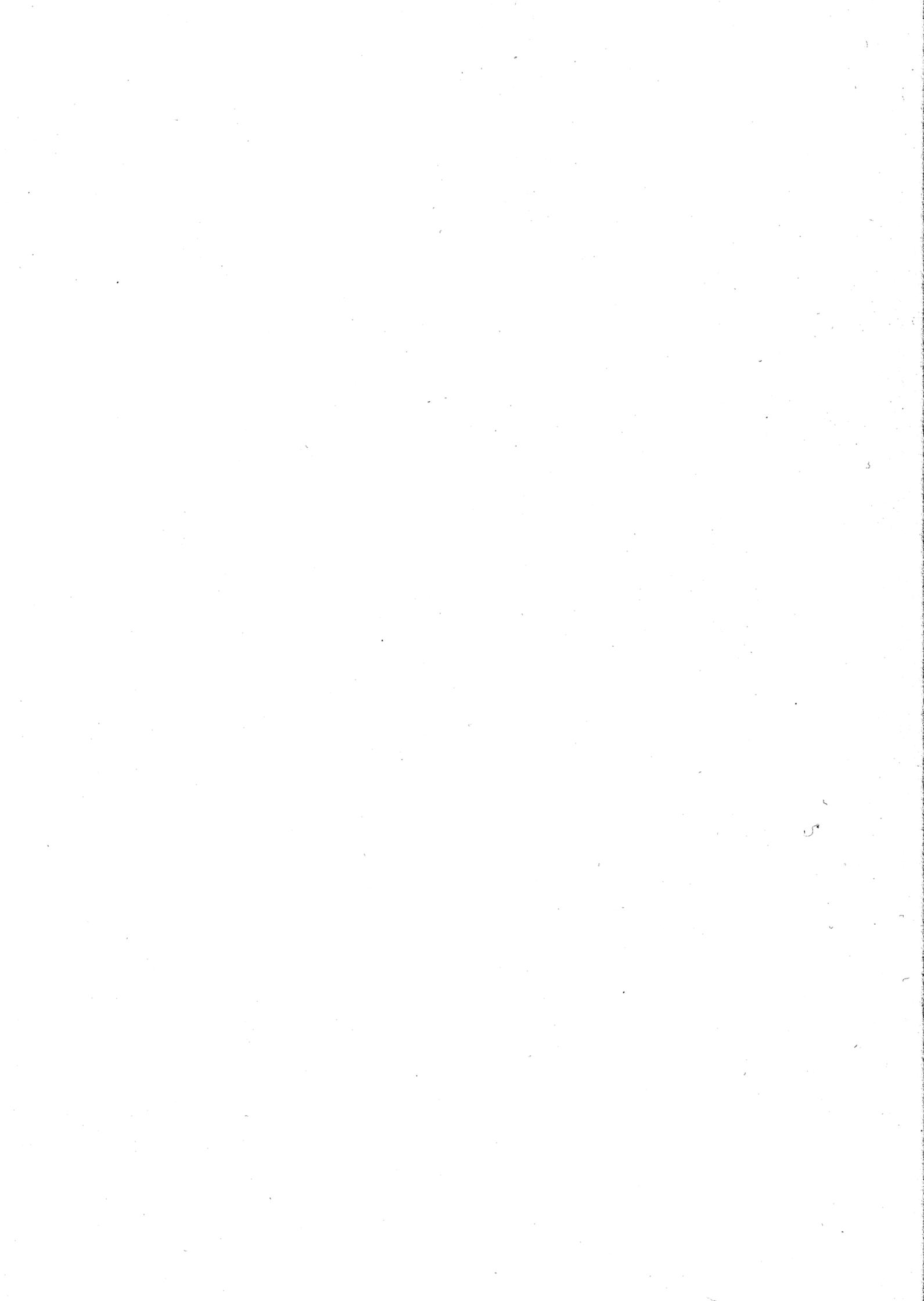


DDR – 5010 Erfurt, Rudolfstraße 47 Telefon 580 Telex 061 306

Die reproduzierten Angaben dieses Handbuchs tragen keinen Informationscharakter.  
Veränderte technische Daten- und Reaktionsgeschwindigkeit sind ausschließlich  
die Verantwortung der vorliegenden Organisation für keine Haftung über sich  
übernehmen und deshalb keine Verbindlichkeiten der Produktion.

## Inhaltsübersicht

	Seite
1. Eigenschaften des DMA	5
2. Aufbau und Signalstruktur	6
2.1. Blockschaltung	6
2.2. Symbol, Pinbelegung U 858 D	7
2.3. Pinbeschreibung	7
3. Funktionsweise	10
3.1. Funktionstypen	10
3.2. Betriebsarten	11
3.2.1. Byte-Mode (Byte at a time - Einbyteübertragung)	11
3.2.2. Burst-Mode (peripheriegesteuerte Arbeitsweise)	12
3.2.3. Continuous-Mode (programmgesteuerte Arbeitsweise)	13
4. Adreß- und Bytezähler	14
5. Variabler Zyklus	15
6. Autorestart	15
7. Impulserzeugung	15
8. Interrupts	16
8.1. Interruptvektoren	17
8.2. Interruptlatches	17
8.3. Interrupt bei Ready (RDY)	18
8.4. Interruptserviceroutinen	18
8.5. Rückkehr vom Interrupt	18
8.6. Interruptpriorisierung mehrerer DMA	18
9. Interne Struktur (Registeraufbau)	19
10. Programmierung des DMA	21
10.1. Schreibregister	21
10.2. Leseregister (RR 0 - RR 6)	30
11. Zeitverhalten des DMA	32
11.1. Inaktiver Zustand	32
11.2. Aktiver Zustand	33
11.2.1. Variabler Zyklus und Flankentaktierung	34
11.2.2. Busanforderung und Quittung	35
11.2.3. Busfreigabe	37
12. Technische Daten	39
12.1. Betriebsbedingungen UA 858 D	39
12.1.1. Statische Kennwerte	39
12.1.2. Dynamische Kennwerte	39
12.2. Betriebsbedingungen UB 858 D	42
12.2.1. Statische Kennwerte	42
12.2.2. Dynamische Kennwerte	42
13. Gehäuse	



## 2. Aufbau und Signalstruktur

### 2.1. Blockschaltung

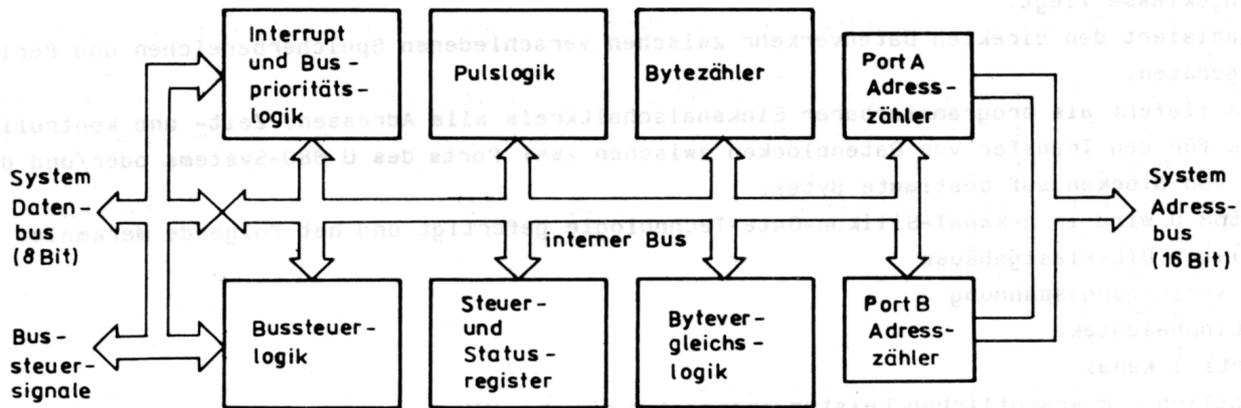


Bild 1 Blockschaltbild

Wie dem Blockschaltbild im Bild 1 zu entnehmen ist, gliedert sich der DMA in folgende Funktionseinheiten:

- |                                     |   |
|-------------------------------------|---|
| Businterfaces                       | - Die Businterfacelogik erlaubt den Anschluß an den Systembus ohne zusätzliche externe Logik.   |
| Steuerlogik und zugehörige Register | - Die Steuerlogik mit ihren dazugehörigen Registern dient im wesentlichen zur Dekodierung und Zwischenspeicherung der Steuerparameter, wie z. B. Funktionstyp und Betriebsart.  |
| Adreß-Byte-Zähler und Pulsschaltung | - Erzeugung der Portadressen für Lese- und Schreiboperation, die, abhängig von der Programmierung, inkrementiert bzw. dekrementiert oder als fixe Adressen nicht verändert werden. Weiterhin wird durch den Bytezähler bei Blockende ein Flag im Statusregister gesetzt, und durch die Pulslogik wird in 256-Byte-Intervallen immer dann ein Signal generiert, wenn die niederwertigen 8 Bit des Bytezählers mit dem Inhalt des Pulsregisters übereinstimmen. |
| Steuerung des DMA-Zeitverhaltens    | - Das Zeitverhalten bei Lese- und Schreibzugriffen für beide Ports ist variabel und wird hier zwischengespeichert bzw. ausgewertet.   |
| Bytevergleichslogik                 | - Bei den Betriebsarten Datensuche (Search) bzw. Übertragung und Datensuche (Transfer + Search) werden durch Vorgabe eines Maskenbytes bestimmte Bits der momentanen Bytes mit dem gesuchten Byte (Match-Byte) verglichen. Wird das gesuchte Byte während einer Suche bzw. einer Übertragung und Suche gefunden, so erfolgt eine Status- und (bei Freigabe) Interruptmeldung.   |
| Interrupt- und Busprioritätslogik   | - beinhaltet das Interruptsteuerregister, in dem die Bedingungen für die Auslösung eines Interrupts durch den DMA gespeichert sind. Ferner befindet sich hier auch das Interruptvektorregister, in dem der Interruptvektor gespeichert ist.   |
| Statusregister                      | - beinhaltet die Statusinformationen des DMA-Kanals   |

## 2.2. Symbol, Pinbelegung U 858 D

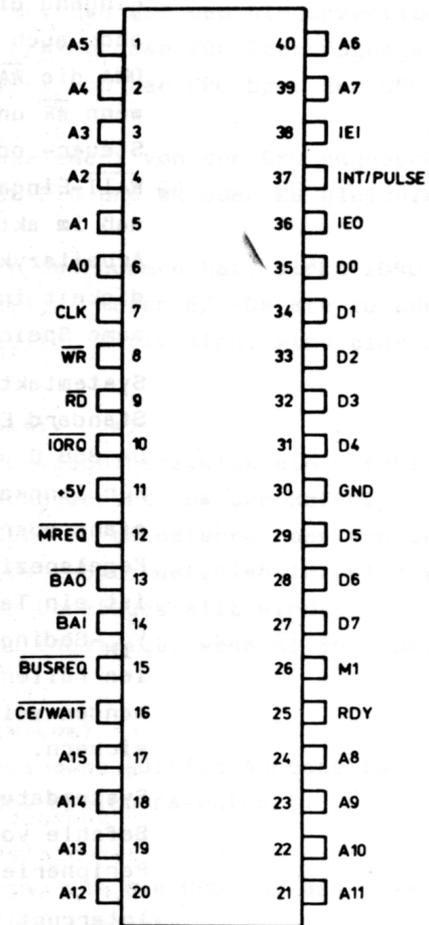
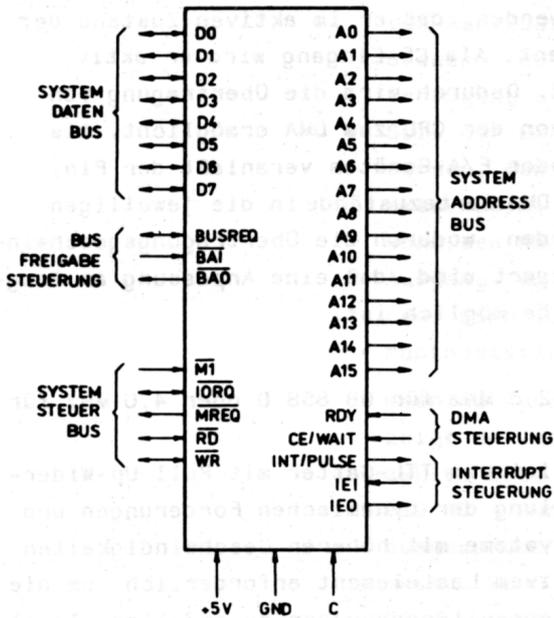


Bild 2 Symbol

Bild 3 Pinbelegung

## 2.3. Pinbeschreibung

A0 - A15

Systemadressbus (Tristateausgang)

An diesen Pins werden vom DMA zwei Adressen (Quell- und Senkenport) generiert, sowohl Speicher- als auch Peripherieadressen.

$\overline{\text{BAI}}$

Busacknowledge IN (Eingang - aktiv Low)

signalisiert, daß der Systembus für die DMA-Steuerung freigegeben ist. In Mehrfach-DMA-Konfigurationen wird der  $\overline{\text{BAI}}$  Pin des DMA mit höchster Priorität an den  $\overline{\text{BUSACK}}$  der CPU angeschlossen.  $\overline{\text{BAI}}$  vom DMA mit niedriger Priorität wird an  $\overline{\text{BAO}}$  des DMA mit höherer Priorität angeschlossen (Daisy Chain).

$\overline{\text{BAO}}$

Busacknowledge OUT (Ausgang - aktiv Low)

In Systemen mit Mehrfach-DMA-Konfigurationen signalisiert dieser Pin, daß kein DMA höherer Priorität den Systembus angefordert hat.  $\overline{\text{BAI}}$  und  $\overline{\text{BAO}}$  bilden eine Daisy Chain für Mehrfach-DMA-Prioritätskaskadierung zur Bussteuerung.

$\overline{\text{BUSRQ}}$

Bus Request (bidirektional, aktiv Low, Open Drain)

sendet als Ausgang Anforderungen zur Steuerung des Systembusses an die CPU. Wenn mehrere DMA's in einer Daisy Chain über  $\overline{\text{BAI}}$  und  $\overline{\text{BAO}}$  verbunden sind, wird über  $\overline{\text{BUSRQ}}$  als Eingang festgestellt, ob ein anderer DMA den Bus angefordert hat und veranlaßt, daß die Busanforderung bis zur Abarbeitung anderer DMA (höherpriorisierter DMA) unterlassen wird. An diesem Pin wird ein Pull-Up-Widerstand angeschlossen (Open Drain).

**CE/WAIT**

Chip Enable/WAIT (Eingang, aktiv Low)

Eingang dient zur Chipfreigabe, z. B. bei Initialisierung, kann aber auch so programmiert werden, daß er im aktiven Zustand der DMA die **WAIT**-Funktion bedient. Als CE-Eingang wird er aktiv, wenn **WR** und **IORQ** aktiv sind. Dadurch wird die Übertragung von Steuer- oder Befehlsbytes von der CPU zum DMA ermöglicht. Als **WAIT**-Eingang vom Speicher oder E/A-Geräten veranlaßt der Pin, daß im aktiven Zustand des DMA Wartezustände in die jeweiligen Arbeitszyklen eingefügt werden, wodurch die Übertragungsgeschwindigkeit in der Weise verringert wird, daß eine Anpassung an langsame Speicher oder E/A-Geräte möglich ist.

**C**

Systemtakt (Eingang)

Standard Einphasentakt von 2,5 MHz für ÜB 858 D oder 4,0 MHz für UA 858 D

Für langsamere Systemtakte ist ein TTL-Gatter mit Pull-Up-Widerstand ausreichend zur Erfüllung der dynamischen Forderungen und Pegelspezifikationen. Für Systeme mit höheren Geschwindigkeiten ist ein Takttreiber mit aktivem Lastelement erforderlich, um die  $V_{IH}$ -Bedingung und die Anstiegszeitforderungen zu erfüllen. In allen Fällen muß ein Pull-Up-Widerstand zu  $U_{CC}$  von 10 k $\Omega$  (max) vorhanden sein, um eine richtige Funktion beim Rücksetzen des DMA zu sichern.

**DO - D7**

Systemdatenbus (bidirektional, Tristate)

Befehle von der CPU, DMA-Status und Daten vom Speicher oder E/A-Peripheriegeräten werden auf diesen Leitungen übertragen.

**IEO**

Interrupt Enable Out (Ausgang, aktiv High)

IEO ist nur High, wenn IEI High ist und die CPU keinen Interrupt von diesem DMA bedient. Somit verhindert dieses Signal, daß Peripheriebausteine niederer Priorität einen Interrupt auslösen, während ein Peripheriebaustein höherer Priorität durch sein CPU-Interrupt-Serviceprogramm bedient wird.

**IEI**

Interrupt Enable In (Eingang, aktiv High)

Dieser Eingang wird mit IEO zur Bildung einer Daisy Chain verwendet, wenn sich mehrere interruptauslösende Bauelemente im System befinden. Ein High auf dieser Leitung zeigt an, daß kein anderer IS mit höherer Priorität von einem CPU-Interrupt-Service-Programm bedient wird.

**INT/PULSE**

Interrupt Request (Ausgang, aktiv Low, Open Drain)

Dieser Ausgang fordert einen CPU-Interrupt an. Die CPU bestätigt den Interrupt, indem sie während eines M1-Zyklus ihren **IORQ**-Ausgang auf Low zieht. Er ist im typischen Fall an den **INT**-Pin der CPU mit einem Pull-Up-Widerstand angeschlossen und mit allen anderen **INT**-Pin's im System verbunden. Weiterhin kann dieser Ausgang auch zur Generierung periodischer Impulse genutzt werden, wenn der DMA aktiv ist (d. h. **BUSRQ** und **BUSACK** der CPU sind beide Low und die CPU kann keine Interrupts annehmen).

**IORQ** Input/Output Request (bidirektional, aktiv Low, Tristate)  
Als Eingang zeigt dieser Pin an, daß auf dem niederwertigen Teil des Adreßbusses eine gültige E/A-Adresse für Übertragungen von Steuer-, Daten- oder Statusbytes von der CPU bzw. zur CPU zur Verfügung steht.

Soll der DMA als Peripheriebauelement von der CPU angesprochen werden, muß zusätzlich sein  $\overline{CE}$ -Pin und  $\overline{WR}$  oder  $\overline{RD}$  gleichzeitig aktiv sein.

Wenn der DMA die BUS-Kontrolle übernommen hat, wirkt  $\overline{IORQ}$  als Ausgang, um analog zur CPU, Speicher oder E/A-Geräte zu adressieren. Wenn  $\overline{IORQ}$  und  $\overline{M1}$  gleichzeitig aktiv sind, wird eine Interruptbestätigung angezeigt.

**M1** Mashine Cycle One (Eingang, aktiv Low)

zeigt an, daß der laufende CPU-Maschinenzyklus ein Befehlsholezyklus ist. Er wird vom DMA ausgewertet, um den Befehl RETI (ED 4D - Rückkehr vom Interrupt) zu dekodieren, der von der CPU ausgesendet wird. Während eines 2-Byte-Befehlsholezyklus wird  $\overline{M1}$  aktiv, wenn jedes OP-Code Byte bereitgestellt wird.

Eine Interruptbestätigung wird angezeigt, wenn  $\overline{M1}$  und  $\overline{IORQ}$  aktiv sind.

**MREQ** Memory Request (Ausgang, aktiv Low)

$\overline{MREQ}$  zeigt an, daß der Adreßbus eine gültige Adresse für eine Lese- oder Schreiboperation des Speichers enthält.

**$\overline{RD}$**  Read (bidirektional, aktiv Low)

Als Eingang zeigt dieser Pin an, daß die CPU aus den Leseregistern des DMA Statusbytes auslesen möchte.

Wenn der DMA aktiv ist, signalisiert  $\overline{RD}$  als Ausgang ein DMA-gesteuertes Lesen aus dem Speicher oder aus E/A-Geräten.

**$\overline{RDY}$**  Ready (Eingang programmierbar, aktiv High oder Low)

Dieser Eingang wird vom DMA überwacht, um festzustellen, wann ein mit einem DMA-Port verbundenes Peripheriegerät für eine Lese- oder Schreiboperation bereit ist.

Je nach der Betriebsart des DMA (Byte, Burst oder Continuous) steuert die  $\overline{RDY}$ -Leitung indirekt die DMA-Aktivität, indem sie bewirkt, daß die  $\overline{BUSREQ}$ -Leitung High oder Low wird.

**$\overline{WR}$**  Write (bidirektional, aktiv Low, Tristate)

Als Eingang wirkt er im inaktiven Zustand und zeigt an, daß die CPU Steuer- oder Befehlsbytes in die DMA-Schreibregister einspeichern möchte. Wenn der DMA die Steuerung übernommen hat, wirkt  $\overline{WR}$  als Ausgang und zeigt eine DMA-gesteuerte Schreiboperation auf eine Speicher- oder E/A-Adresse an.

### 3. Funktionsweise

#### 3.1. Funktionstypen

Der DMA ermöglicht drei grundlegende Funktionstypen.

- Übertragungen von Daten zwischen zwei Ports (Transfer)  
(Speicher oder E/A-Peripheriegeräten)
- Suchen nach einem speziellen 8-Bit maskierbaren Byte (Search) an einem Port  
(Speicher oder E/A-Peripheriegerät)
- Kombination von Datenübertragung bei gleichzeitiger Suche nach einem 8-Bit maskierbaren Byte zwischen zwei Ports (Transfer + Search)  
(Speicher oder E/A-Peripheriegeräten)

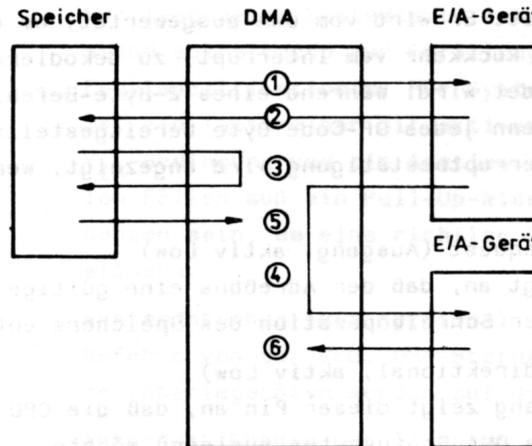


Bild 4

#### Funktionstypen des DMA

1. Transfer Speicher → E/A-Gerät (Transfer + Search)
2. Transfer E/A-Gerät → Speicher (Transfer + Search)
3. Transfer Speicher → Speicher (Transfer + Search)
4. Transfer E/A-Gerät → E/A-Gerät (Transfer + Search)
5. Search Speicher
6. Search E/A-Gerät

#### Transfer (Übertragungsoperationen)

Während einer Übertragung übernimmt der DMA die Steuerung des Systembusses.

Die Daten werden Byte für Byte von einem adressierbaren Port gelesen und auf das andere adressierbare Port geschrieben. Die Ports können so programmiert werden, daß sie entweder Hauptspeicher des Systems oder periphere E/A-Geräte sind. So besteht die Möglichkeit, einen Datenblock von einem peripheren Gerät auf ein anderes, von einem Speicherbereich in einen anderen oder von einem peripheren Gerät auf einen Speicherbereich oder umgekehrt zu übertragen.

#### Search (Suchoperationen)

Bei Suchoperationen werden die Daten vom Quellport gelesen und Byte für Byte mit einem internen DMA-Register verglichen, welches ein programmierbares Vergleichsbyte (Matchbyte) enthält. Dieses Matchbyte kann wahlweise maskiert werden, so daß nur bestimmte Bits innerhalb des Matchbytes zum Vergleich herangezogen werden. In Abhängigkeit von der Taktfrequenz und Betriebsart können Suchraten beim UB 858 D bis 1,25 MByte/s und beim UA 858 D bis 2,0 MByte/s programmiert werden.

## Transfer + Search (kombiniertes Übertragen und Suchen)

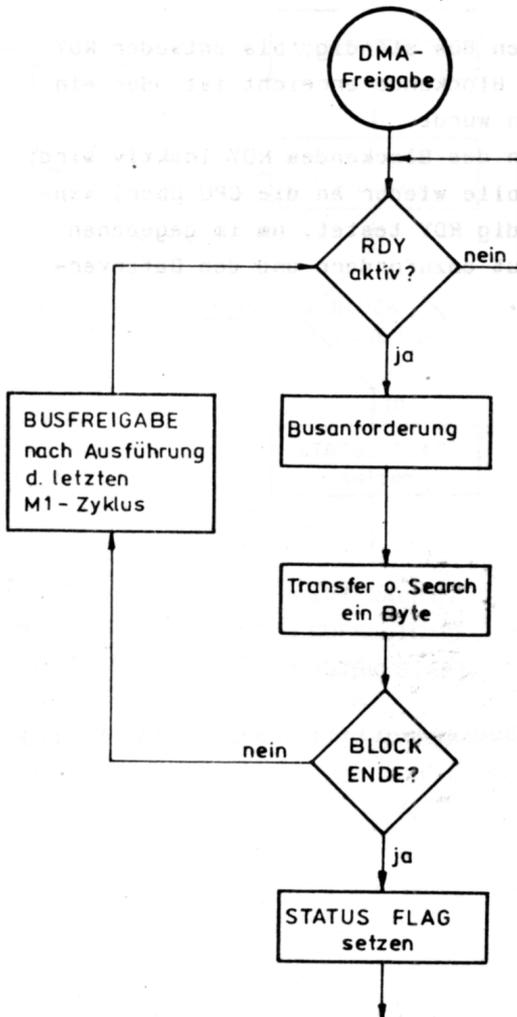
Bei kombiniertem Datenübertragen u. Suchen werden die Daten zwischen zwei Ports übertragen, während gleichzeitig nach einem bitmaskierbaren Matchbyte gesucht wird.

Datenübertragungen oder Suchen (Transfer oder Transfer + Search) können so programmiert werden, daß sie unter verschiedenen Bedingungen beendet werden oder eine Interruptanforderung auslösen (Stop bei Match und/oder Blockende, Interrupt bei Match und/oder Blockende).

### 3.2. Betriebsarten

Zu den unter Punkt 3.1. erwähnten Funktionstypen lassen sich jeweils folgende Betriebsarten programmieren.

#### 3.2.1. Byte-Mode (Byte at a time - Einbyteübertragung)



- INTERRUPT
- BUSFREIGABE
- AUTORESTART

Die Datenoperation wird jeweils nur für ein Byte ausgeführt. Zwischen den Byteoperationen wird der Systembus wieder für die CPU freigegeben. Für jede weitere Operation wird der Bus erneut angefordert (s. Bild 5). Die Operation im Byte-Mode beginnt mit dem "Enable Befehl" von der CPU und der Abfrage der RDY-Leitung deren Pegel durch E/A-Geräte bereitgestellt wird.

Nachdem RDY = aktiv erkannt wurde, fordert der DMA den Systembus über  $\overline{\text{BUSRQ}} = \text{Low}$  an, worauf die CPU die Anforderung mittels  $\overline{\text{BUSACK}} = \text{Low}$  bestätigt und anschließend ihre Tristate-Ausgänge floated.

Der Transfer oder Transfer + Search von einem Byte wird in Bild 5 veranschaulicht. Nachdem ein Byte übertragen wurde, wird der Bytezähler mit der programmierten Blocklänge verglichen.

Wurde das Blockende durch die letzte Operation nicht erreicht, gibt der DMA den Bus wieder an die CPU.

Im anderen Fall, d. h. wenn das Blockende erreicht wurde, werden entsprechende Statusbits gesetzt und die in der Initialisierungsphase festgelegten Aktivitäten werden abgearbeitet.

Die Freigabe des Systembusses zwischen jeder Byteoperation des DMA erlaubt der CPU, mindestens einen M1-Zyklus abzuarbeiten, bevor der DMA mit der nächsten Byteoperation beginnt.

Bild 5 Algorithmus Byte-Mode

### 3.2.2. Burst-Mode (peripheriegesteuerte Operation)

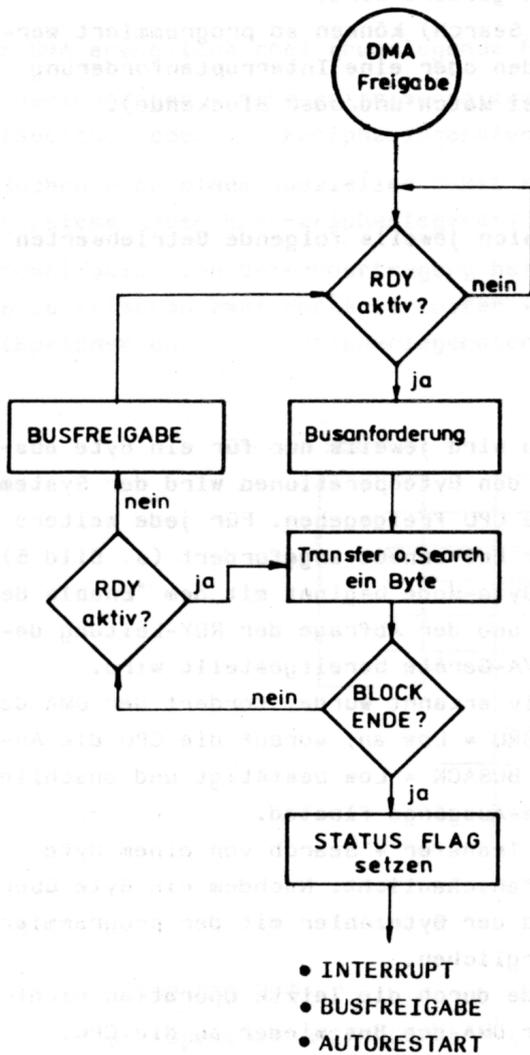


Bild 6 Algorithmus Burst-Mode

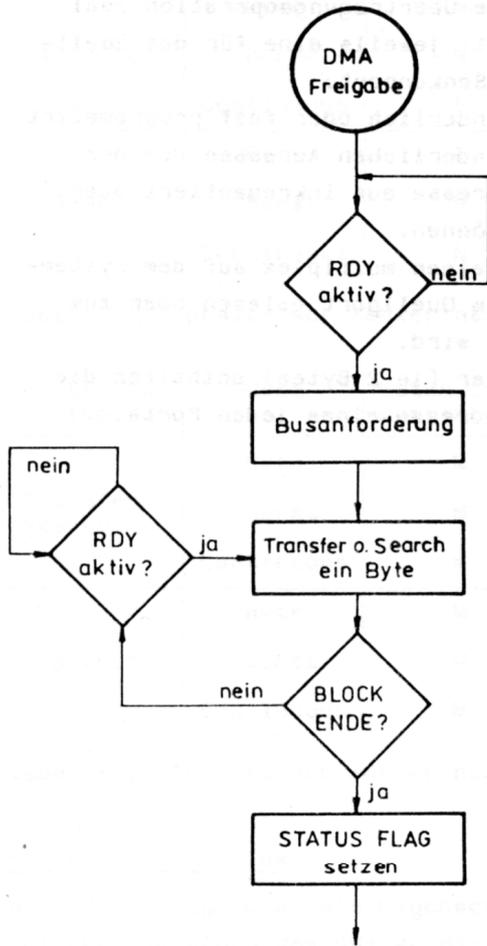
Der DMA übernimmt im jeweiligen Funktionstyp den Systembus und führt solange Datenoperationen aus, bis die RDY-Leitung eines Ports (E/A-Gerät) inaktiv wird, oder bei RDY = aktiv bis zum programmierten Blockende. Nachdem der DMA mit einem Freigabebefehl aktiv wurde, wird entsprechend Bild 6 RDY getestet und als Folge von RDY = aktiv, wird der Bus angefordert.

Der DMA steuert den Bus ständig, bis entweder RDY inaktiv wird, das Blockende erreicht ist oder ein Matchbyte gefunden wurde.

Wenn vor Erreichen des Blockendes RDY inaktiv wird, geht die Buskontrolle wieder an die CPU über, während der DMA ständig RDY testet, um im gegebenen Fall den Bus erneut anzufordern und den Datenverkehr fortzusetzen.



### 3.2.3. Continuous-Mode (programmgesteuerte Betriebsart)



- INTERRUPT
- BUSFREIGABE
- AUTORESTART

Die Aktivitäten zur Busanforderung im Continuous Mode unterscheiden sich in keiner Weise von denen in anderen Betriebsarten (Burst/Byte). Nach der Busübernahme wird ein Byte übertragen und anschließend auf Blockende bzw. auf ein aktives RDY getestet. Im Unterschied zum Burst Mode wird der Bus bei inaktivem RDY vor Blockende bzw. vor Match nicht an die CPU zurückgegeben. In diesem Fall werden noch anstehende Byteoperationen bis zum erneuten RDY = aktiv storniert.

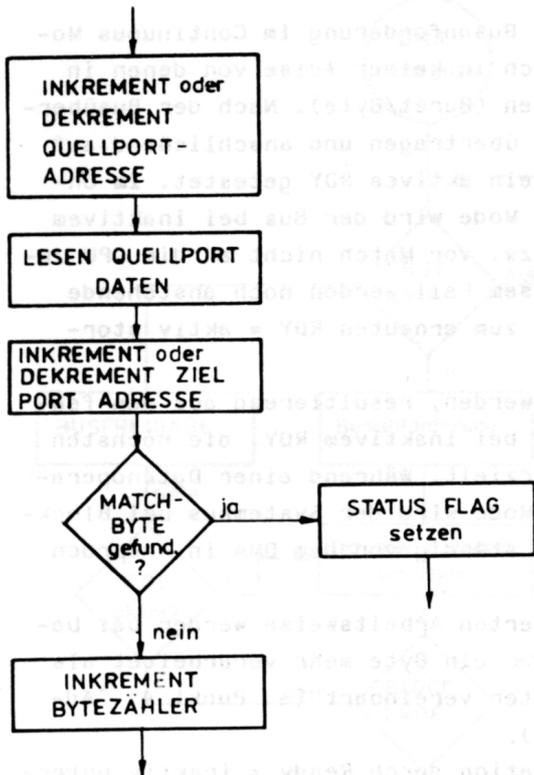
Im Continuous Mode werden, resultierend aus den fehlenden Busfreigaben bei inaktivem RDY, die höchsten Übertragungsraten erzielt. Während einer Datenoperation im Continuous Mode wird der Systembus bei Blockende bzw. bei Match ständig von dem DMA in Anspruch genommen.

Aufgrund der gepufferten Arbeitsweise werden bei Datenoperationen stets ein Byte mehr verarbeitet als im Blocklängenregister vereinbart (s. Punkt 4. "Adreß- und Bytezähler).

Wird eine Datenoperation durch Ready = inaktiv unterbrochen, so wird die momentane Byteoperation gültig abgeschlossen, bevor der DMA seine Aktivitäten unterbricht und abhängig von der gewählten Betriebsart den Systembus an die CPU übergibt oder im Continuous Mode auf ein aktives RDY wartet.

Bild 7 Algorithmus Continuous-Mode

#### 4. Adreß- und Bytezähler



Vom DMA werden für jede Übertragungsoperation zwei 16 Bit-Adressen erzeugt; jeweils eine für das Quellport und eine für das Senkenport. Jede Adresse kann veränderlich oder fest programmiert werden, wobei die veränderlichen Adressen von der programmierten Startadresse aus inkrementiert oder dekrementiert werden können. Die Portadressen erscheinen multiplex auf dem Systembus, je nachdem ob vom Quellport gelesen oder zum Senkenport geschrieben wird. Zwei lesbare Adreßzähler (je 2 Bytes) enthalten die laufende (momentane) Adresse eines jeden Ports.

Bild 8 Algorithmus zur Übertragung oder Suche eines Bytes

Der Bytezähler wird nach der Übertragung des ersten Bytes (mit dem ersten Bytezählimpuls) initialisiert bzw. auf Null gesetzt und nimmt dann folgende Zustände ein:

	Bytezählerstand
1. Byte	unbestimmt
2. Byte	∅
3. Byte	1

Als zwangsläufige Konsequenz dieser Arbeitsweise ergibt sich, daß bei Übertragungen stets ein Byte mehr übertragen wird, als im Blocklängenregister vereinbart wurde (s. Tabellen 1 und 2).

Funktions- typ	Mode	programmierte Blocklänge	Übertragene Bytes bei Stop	Byte- zähler	Quellport Adreßzähler	Zielport Adreßzähler
Transfer	Byte	N	N+1	N	$A_s \pm (N+1)$	$A_s \pm (N)$
	Burst	N	N+1	N	$A_s \pm (N+1)$	$A_s \pm (N)$
	Continuous	N	N+1	N	$A_s \pm (N+1)$	$A_s \pm (N)$
Search	Byte	N	N+1	N	$A_s \pm (N+1)$	...
	Burst	N	N+1	N	$A_s \pm (N+1)$	...
	Continuous	N	N+1	N	$A_s \pm (N+1)$	...

Tabelle 1 Inhalte der Zähler nach DMA-Stop durch Blockende

Funktions- typ	Mode	Match im mo- mentanen Byte	Übertragene Bytes bei Stop	Byte- zähler	Quellport Adreßzähler	Zielport Adreßzähler
Transfer/ Search	Byte	M	M	M-1	$A_s \pm (M)$	$A_s \pm (M-1)$
	Burst	M	M	M-1	$A_s \pm (M)$	$A_s \pm (M-1)$
	Continuous	M	M	M-1	$A_s \pm (M)$	$A_s \pm (M-1)$
Search	Byte	M	M	M-1	$A_s \pm (M)$	...
	Burst	M	M+1	M	$A_s \pm (M+1)$	...
	Continuous	M	M+1	M	$A_s \pm (M+1)$	...

Tabelle 2 Inhalte der Zähler nach DMA-Stop durch Byte Match

### 5. Variabler Zyklus

Der DMA verfügt über die Eigenschaft die Operationszykluslänge variabel zu gestalten, was bei der Anpassung des DMA an die spezifischen Übertragungsbedingungen anderer Systemkomponenten äußerst vorteilhaft ist.

Hinsichtlich der zeitlichen Anpassung der Ports an den DMA ist es möglich, einerseits die erforderliche Zykluslänge zwischen zwei und vier Takten unabhängig bei beiden Ports zu wählen und andererseits kann bei den Steuersignalen  $\overline{MREQ}$ ,  $\overline{RD}$ ,  $\overline{WR}$  und  $\overline{IORQ}$  die aktive Rückflanke per Programm einen halben Taktzyklus eher erfolgen.

Dadurch kann die jeweilige Übertragungsgeschwindigkeit der angeschlossenen Ports in weiten Grenzen sowohl schneller (2 Takte) als auch langsamer (4 Takte oder mehr, wenn Waitzyklus benutzt wird) gewählt werden (s. Bild 30).

### 6. Autorestart

Blockübertragungen können so programmiert werden, daß der DMA seine Operation automatisch wiederholt, d. h., daß die Startadressen eines jeden Ports am Ende einer Blockübertragung wieder neu geladen werden können.

Die Möglichkeit des Autorestarts befreit die CPU vom Softwareaufwand, den DMA bei wiederkehrenden Operationen jedesmal neu zu programmieren (z. B. CRT-Auffrischung).

Wenn die CPU während Byte- oder Burst-Transfers Zugriff zum Systembus hat, können darüber hinaus unterschiedliche Startadressen in die Pufferregister eingespeichert werden, so daß der Autorestart an einer neuen Stelle beginnt.

### 7. Impulserzeugung

Im Rahmen der Programmierung des DMA kann festgelegt werden, daß auf dem Pin  $\overline{INT}$  immer dann ein Impuls generiert wird, wenn der DMA aktiv ist und 256 Byte übertragen wurden. Durch ein Puls-Steuerbyte besteht die Möglichkeit, den Impuls um 1 bis 255 Bytes zu versetzen (Offset). Eine Fehlinterpretation des Impulses als Interruptanforderung an die CPU ist ausgeschlossen, da sich die CPU während dieser Zeit im inaktiven Zustand befindet.

## 8. Interrupts

Wie bereits im Punkt 1 beschrieben, ordnet sich der DMA hinsichtlich seiner Interruptstruktur bzw. seines Interruptverhaltens in das Schema der U 880-Familie ein. Der DMA nutzt im Interruptsystem der CPU den maskierbaren Interrupt (MODE 2), indem durch programmierbare Interruptvektoren eine leistungsstarke Peripherie aufgebaut werden kann. Hinsichtlich des Interruptverhaltens kann der DMA so programmiert werden, daß er die CPU jeweils beim Eintreffen einer der drei folgenden Bedingungen unterbricht:

- Interrupt bei RDY = aktiv (ehe Bus angefordert wird)
- Interrupt bei Blockende (bei Blocklängenregister = Bytezähler)
- Interrupt bei Match (Inhalt vom Match-Byte-Register = momentanem Datenbyte im Search- oder Search/Transfer-Mode)

Jeder dieser Interrupts bewirkt, daß das Interrupt-Pending-Statusbit gesetzt wird, und jeder von ihnen kann wahlweise den Interruptvektor des DMA modifizieren. Während der DMA den Systembus steuert, kann die CPU keine Interruptanforderungen bearbeiten. Der DMA nutzt die  $\overline{\text{INT}}$ -Leitung, um während der Datenübertragung periodische Impulse für entsprechende E/A-Geräte zu generieren. Nach Blockende bzw. nach einem Match-Byte gibt der DMA den Bus zur Steuerung wieder an die CPU zurück und meldet danach eine Interruptanforderung an (Bild 9).

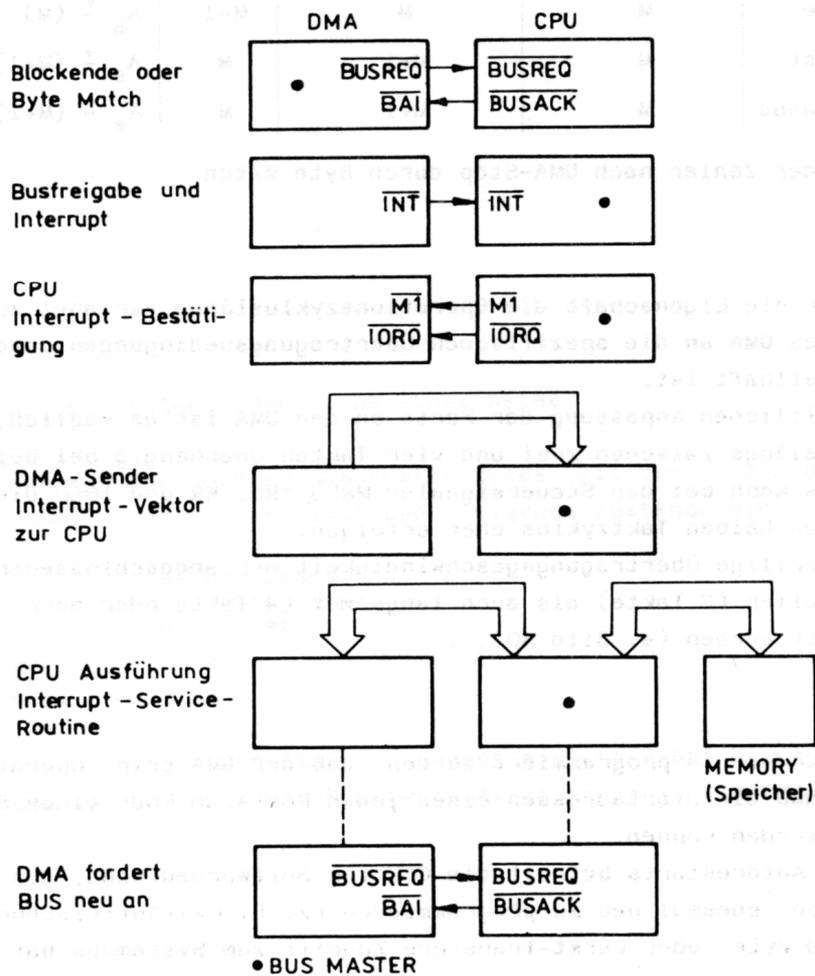


Bild 9 Interruptsequenz

Die CPU bestätigt diese Interruptanmeldung, und anschließend gibt der DMA - vorausgesetzt, daß kein höherpriorisiertes Bauelement einen Interrupt angemeldet hat - seinen Interruptvektor auf den Bus. Der Prozessor U 880 bildet mit seinem I-Register und dem Interruptvektor des DMA eine 16 Bit-Speicheradresse, die in der Regel auf eine Sprungtabelle hinweist. In der Interruptserviceroutine wird im Normalfall der DMA für erneute Buszugriffe freigegeben.

## 8.1. Interruptvektoren

Im Interruptannahmezyklus der CPU U 880 D wird der DMA veranlaßt, seinen Interruptvektor (8 Bit) auf den Datenbus zu geben. Dieser Vektor wird in der CPU in einem temporären Register abgespeichert und bildet die unteren 8 Bit einer 16 Bit-Adresse. Die oberen 8 Bit werden durch das I-Register der CPU bereitgestellt.

Eine auf diese Weise generierte Adresse zeigt auf eine Sprungtabelle, welche sich im RAM befindet. In dieser Sprungtabelle sind wiederum Adressen enthalten, die jeweils auf den ersten Befehl der gültigen Interruptserviceroutine hinweisen.

## 8.2. Interruptlatches

In der Interruptstruktur des DMA befinden sich zwei Latches, die den jeweiligen Interruptstatus anzeigen bzw. speichern.

### Interrupt Pending Latch (IP) (s. Bild 10)

Dieses Latch wird immer dann gesetzt, wenn der DMA einen Interrupt anfordert, welcher aber derzeit noch nicht bestätigt wurde. Durch dieses Latch wird die  $\overline{INT}$ -Leitung aktiviert.

### Interrupt Under Service Latch (IUS) (s. Bild 11)

Dieses Latch wird gesetzt, wenn die CPU den DMA-Interrupt bestätigt. Durch das IUS werden folgende Aktivitäten ausgelöst:

- Sperrung weiterer Interrupts dieses DMA
- Sperrung von Interrupts niederpriorisierter E/A-Geräte in der Interrupt-Daisy-Chain
- Sperrung weiterer Busanforderungen dieses DMA

Wenn beim DMA Interrupt durch RDY (Interruptauslösung bevor der Systembus angefordert wird) programmiert wurde, wird das Interrupt Pending Latch (IP) gesetzt, sobald RDY aktiv ist. Das IP-Latch wird zurückgesetzt, sobald das IUS-Latch gesetzt wird. Das IUS-Latch wird durch den U 880-Befehl RETI (Return from Interrupt) oder durch ein Steuerbyte im Schreibregister WR6 zurückgesetzt.

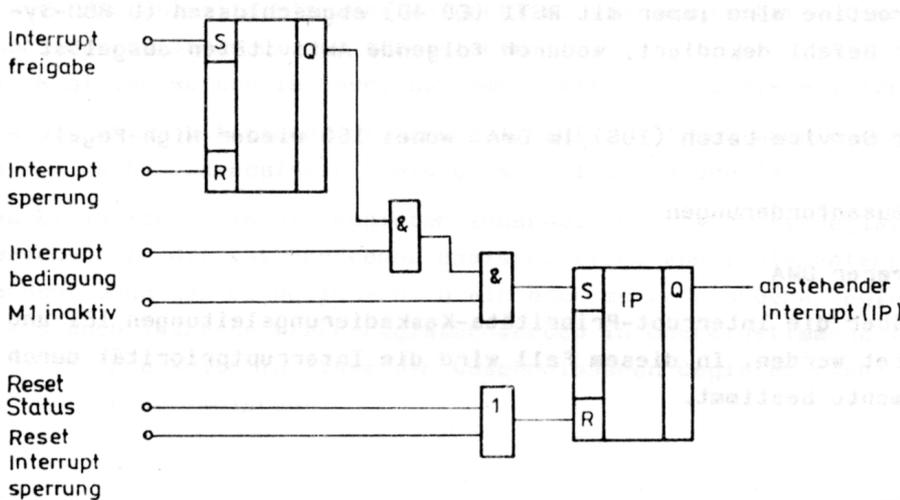


Bild 10 Interrupt Pending Latch

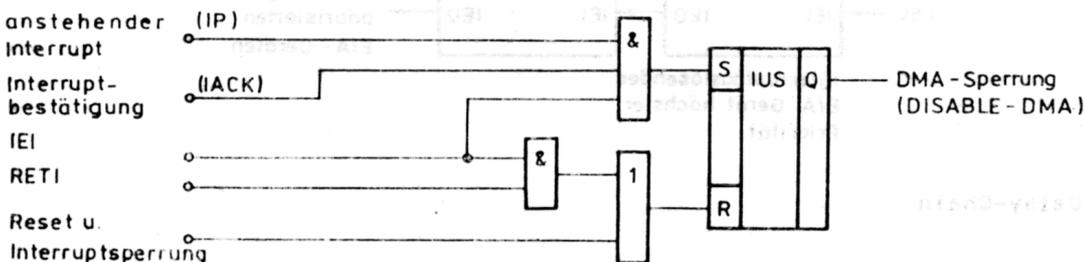


Bild 11 Interrupt Under Service Latch

### 8.3. Interrupt bei Ready (RDY)

Im Normalfall, wenn der DMA zur Busübernahme von der CPU freigegeben wurde, während RDY inaktiv war, wird durch einen Pegelwechsel zu RDY = aktiv Busrequest auf Low gezogen. Sobald aber in der DMA-Initialisierung Interrupt durch RDY vereinbart wurde, wird mit aktivem RDY eine Interruptanforderung an die CPU ausgelöst. Im Rahmen der daraufhin abzuarbeitenden Interruptserviceroutine müssen entsprechende Steuerbytes (WR6) den DMA nach Abschluß der ISR dazu befähigen, den Bus anzufordern und die programmierten Datenoperationen auszuführen. In der ISR, die durch "Interrupt bei Ready" aufgerufen wurde, müssen u. a. folgende Aktivitäten berücksichtigt werden:

1. Freigabe nach Rückkehr vom Interruptbefehl (RETI)
2. Freigabe DMA
3. RETI-Befehl des DMA setzt IUS zurück

### 8.4. Interruptserviceroutinen

In den möglichen Interruptserviceroutinen, die durch den DMA ausgelöst werden können, sollten einige wichtige Steuerbytes zur Steuerung des DMA-Verkehrs enthalten sein.

- Status lesen
- Reset DMA oder Reset Status
- Disable INT oder Reset INT
- Einschreiben der neuen Übertragungsparameter (Blocklänge, Adressen usw.)
- Laden (LOAD) evtl. FORCE READY
- Freigabe des DMA
- RETI

### 8.5. Rückkehr vom Interrupt

Das Ende einer Interruptserviceroutine wird immer mit RETI (ED 4D) abgeschlossen (U 880-System). Durch den DMA wird dieser Befehl dekodiert, wodurch folgende Aktivitäten ausgelöst werden:

- Rücksetzen des Interrupt Under Service Latch (IUS) im DMA, wobei IEO wieder High-Pegel annimmt
- Freigabe des DMA für weitere Busanforderungen

### 8.6. Interruptpriorisierung mehrerer DMA

Mehrfach-DMA-Strukturen können über die Interrupt-Prioritäts-Kaskadierungsleitungen IEI und IEO entsprechend Bild 12 verkettet werden. In diesem Fall wird die Interruptpriorität durch die lokale Anordnung der Bauelemente bestimmt.

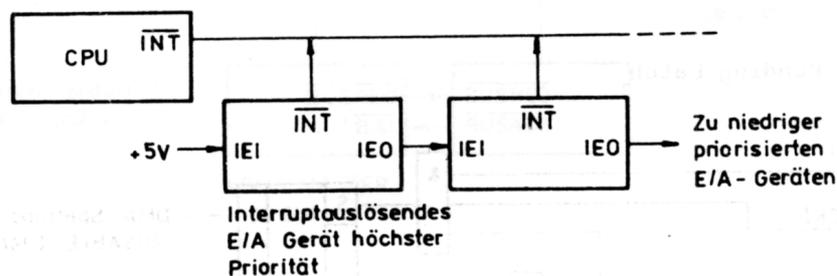


Bild 12 Interrupt-Daisy-Chain

Eine höhere Priorität eines DMA gegenüber einem anderen DMA in einer Daisy-Chain liegt immer dann vor, wenn der betreffende DMA gegenüber dem anderen DMA mit IEI näher an 5 V liegt. Das heißt, das erste Bauelement einer Daisy-Chain hat immer die höchste Priorität der Kette. In einer Daisy-Chain-Prioritätskette wird demzufolge immer das derzeitig höchstpriorisierte Interruptanmeldende Bauelement durch die CPU bedient. Ein DMA, der z. Z. der Interruptanmeldung die höchste Priorität aufweist, ist an IEI und IEO in folgender Weise gekennzeichnet:

IEI = High

IEO = Low

Immer dann, wenn eine Interruptanforderung an der CPU anliegt, ist die Interruptstruktur der CPU für weitere Interrupts gesperrt. Um in einem solchen Fall weiterhin Interrupts zuzulassen, muß die CPU im Rahmen der Interrupt-Service-Routine erneut freigegeben werden.

### 9. Interne Struktur (Registeraufbau)

Die interne Struktur des DMA umfaßt Treiber- und Empfängerschaltungen für den Anschluß an einen 8-Bit-Datenbus, 16-Bit-Adreßbus und die Systemsteuerleitungen.

Bei Verwendung der CPU U 880 D kann der DMA analog zur CPU ohne Pufferung an den Systembus angeschlossen werden, wobei nur die  $\overline{CE}/\overline{WAIT}$ -Leitung eine Ausnahme bildet.

Ein Satz von 21 beschreibbaren Steuerregistern und 7 lesbaren Registern liefert das Hilfsmittel, mit dem die CPU die Aktivitäten des DMA steuert und überwacht. Alle Register sind 8 Bit breit, wobei Zweibyte-Informationen in benachbarten Registern gespeichert sind. Die beiden Adreßzähler (je 2 Byte) für Port A und B werden durch die beiden Startadressen gepuffert. Die 21 beschreibbaren Steuerregister sind in sieben Basisregistergruppen aufgliedert, wobei einige von ihnen zugeordnete Register höherer Ordnung haben, die sequentiell beschrieben werden. Die Basisregister in jeder beschriebenen Gruppe enthalten sowohl Steuer-/Befehlsbits als auch Zeigerbits, die so gesetzt werden können, daß sie andere Register innerhalb der Gruppe adressieren.

Die Register werden in Übereinstimmung mit ihren Basisregistergruppen wie folgt bezeichnet:

WR 0 - WR 6 Schreibregistergruppen 0 bis 6 (7 Basisregistergruppen + 14 zugehörige Register)

RR 0 - RR 6 Leseregister 0 bis 6 (s. Bilder 13 und 14)

Zum Einspeichern in ein Register innerhalb einer Schreibregistergruppe gehört zuerst das Einspeichern in das entsprechende Basisregister, wobei die entsprechenden Zeigerbits gesetzt werden, dann das Einspeichern in ein oder mehrere andere Register innerhalb der Gruppe.

Die sieben lesbaren Statusregister werden in Übereinstimmung mit einer programmierbaren Maske, die ihrerseits in einem der beschreibbaren Register steht, sequentiell angesprochen.

(s. Pkt. Programmierung)

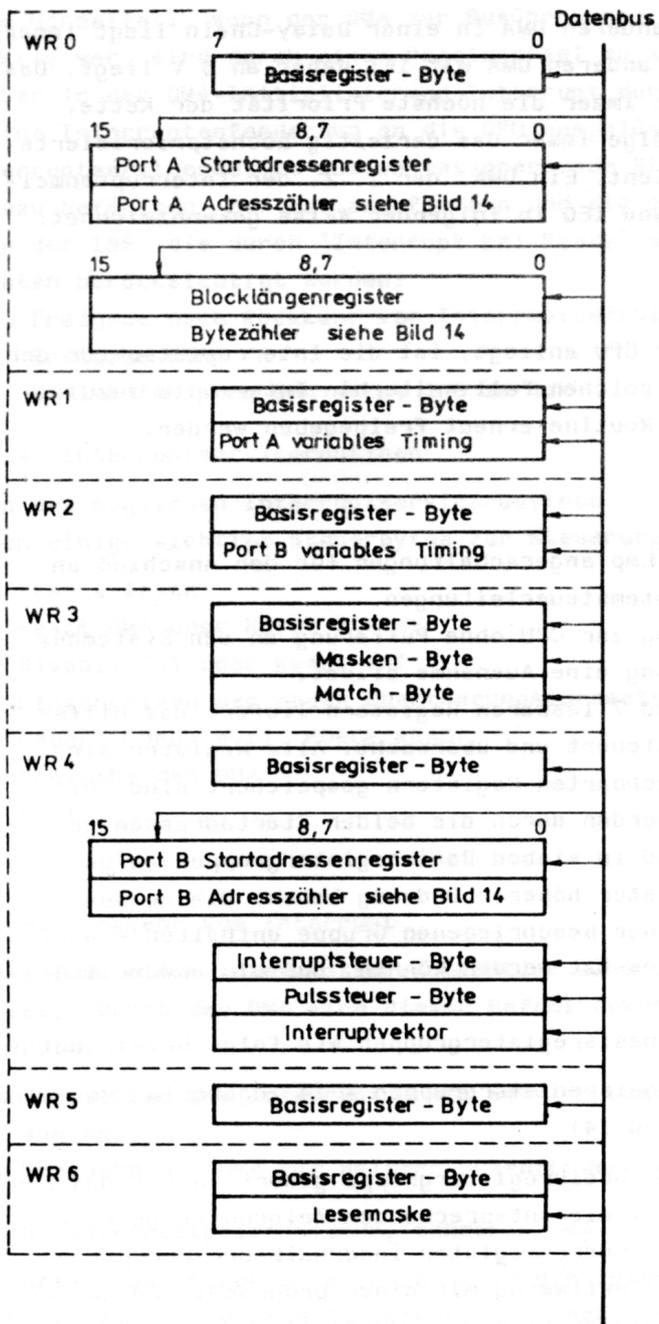


Bild 13 Schreibregisterorganisation

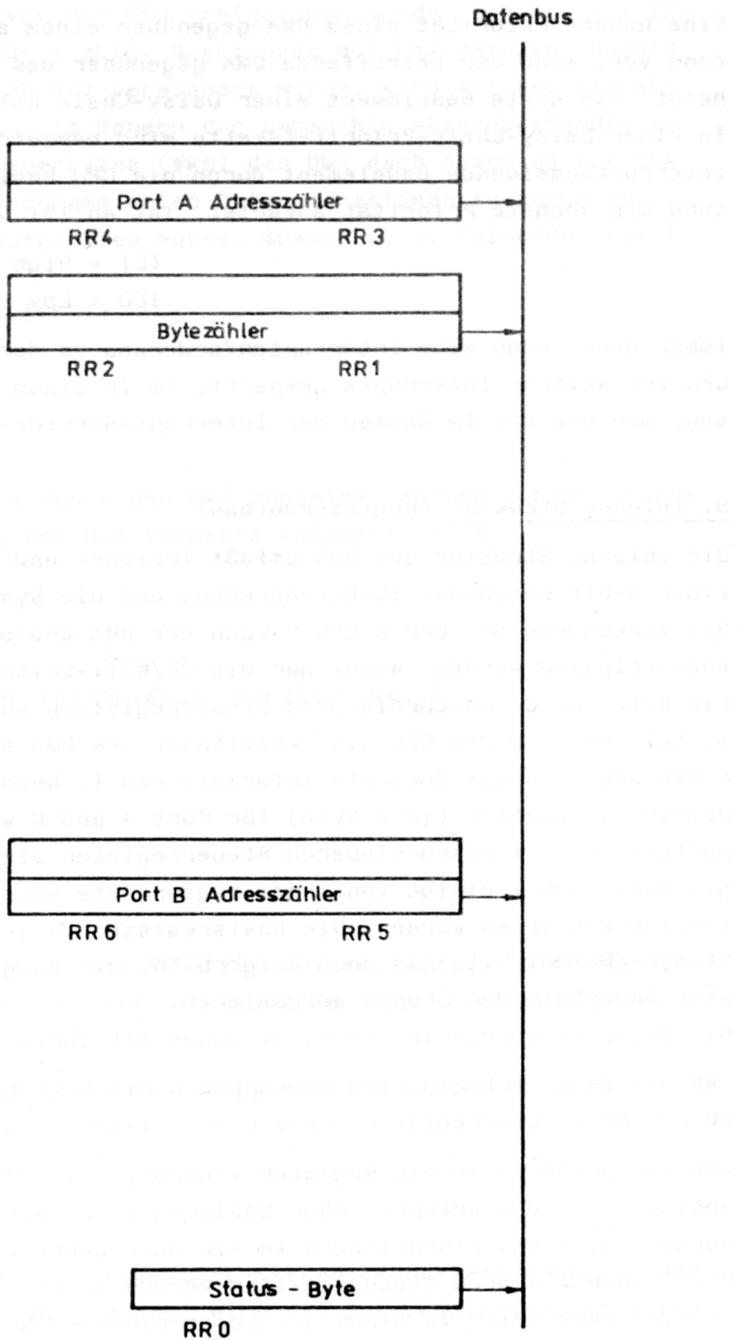


Bild 14 Leseregisterorganisation

## 10. Programmierung des DMA

Um den DMA in sinnvoller Weise zu nutzen, muß er zuvor entsprechend programmiert werden. Der DMA hat zwei programmierbare Grundzustände:

- Einen Enable-Zustand, in dem er die Steuerung des Systembusses übernehmen und die Datenoperationen zwischen den Ports steuern kann und
- einen Disable-Zustand, in dem er weder Busanforderungen noch Datenoperationen auslösen kann.

Nach dem Einschalten der Stromversorgung oder nach dem Rücksetzen befindet sich der DMA immer im Disable-Zustand. Eine Programmierung des DMA durch die CPU ist sowohl im Disable-Zustand als auch, unter bestimmten Umständen, im Enable-Zustand möglich (z. B. im Byte-Mode), führt aber dazu, daß er dabei automatisch den Disable-Zustand erreicht und dort verbleibt, bis von der CPU ein erneuter Freigabebefehl gegeben wird. (D. h., Steuerbytes können immer dann in den DMA eingeschrieben werden, wenn die CPU die Buskontrolle übernommen hat.)

Die CPU muß den DMA vor jeder Datenoperation programmieren, indem sie ihn als ein E/A-Port adressiert und eine Folge von Steuerbytes sendet.

### 10.1. Schreibregister

Steuer- oder Befehlsbytes werden in ein oder mehrere Schreibregistergruppen (WR 0 - WR 6) eingespeichert, indem zuerst auf das Basisregisterbyte in der Gruppe zugegriffen wird. Alle Gruppen besitzen Basisregister, und die meisten Gruppen haben zusätzlich zugehörige Register höherer Ordnung. Die zugehörigen Register einer Gruppe werden sequentiell angesprochen, indem zuerst ein Byte auf das Basisregister gespeichert wird, welches Registergruppenerkennungs- und Zeigerbits (1 = aktiv) für ein oder mehrere zugehörige Register des Basisregisters enthält (s. Bild 15).

In dieser Abbildung wird die Folge, in der auf die zugehörigen Register innerhalb einer Gruppe zugegriffen werden kann, durch die vertikale Position der angeschlossenen Register angegeben. Wenn z. B. ein Byte, das auf den DMA gespeichert wurde, die Bits enthält, die WR 0 (Bits D0, D1, D7) spezifizieren und, wenn ebenfalls eine logische '1' in den Bitpositionen steht, die auf die folgende Startadresse von Port A (niederwertiges und höherwertiges Adreßbyte) weisen, dann werden die nächsten beiden Bytes, die auf dem DMA gespeichert werden, in diese Register geschrieben.

#### Schreibregister 0 WR 0

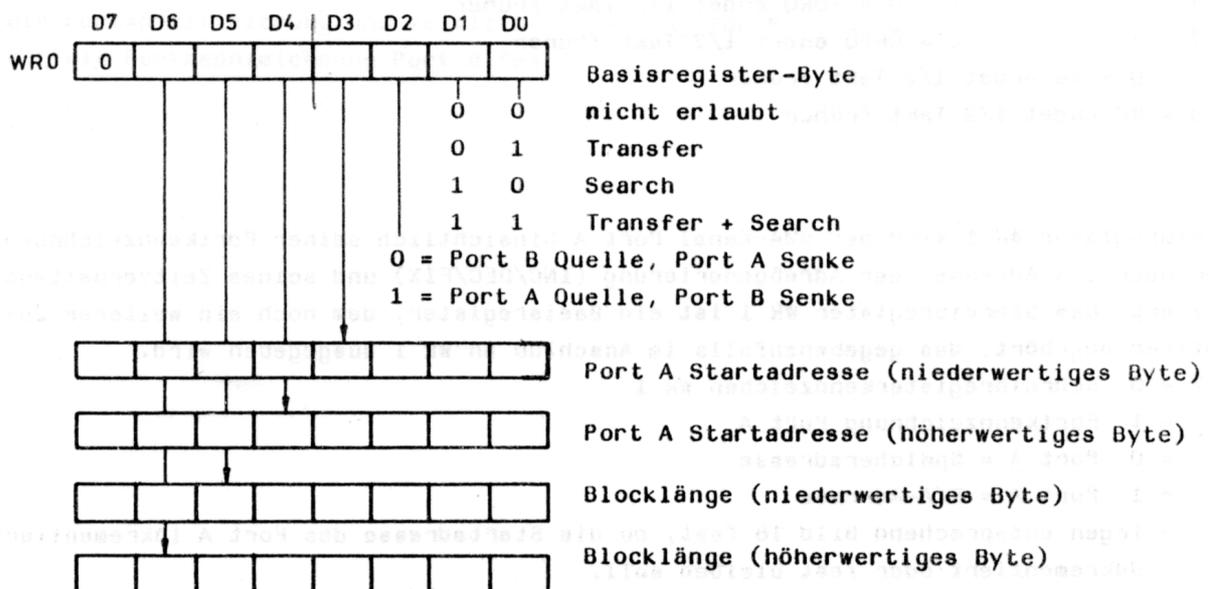


Bild 15

Das Schreibregister WR 0 ist ein Basisregister, zu dem eine Gruppe von vier Registern gehört, in denen die Portadresse A und die Blocklänge abgelegt werden. Gekennzeichnet ist WR 0 durch D7 = 0.

- D0, D1 - spezifizieren den DMA-Funktionstyp entsprechend Bild 15  
(Datentransfer, Datensuche, Datentransfer und -suche)
- D2 - legt die Übertragungsrichtung fest (entsprechend Bild 15)
- D3 = 1 - Startadresse Port A, niederwertiges Byte folgt
- D4 = 1 - Startadresse Port A, höherwertiges Byte folgt
- D5 = 1 - Blocklänge, niederwertiges Byte folgt
- D6 = 1 - Blocklänge, höherwertiges Byte folgt

Die Startadresse für das Port B folgt nach dem WR 4.

Es ist erforderlich, daß die Blocklänge stets um 1 niedriger ist als die Anzahl der zu transferierenden Bytes.

Bei Programmierung der Betriebsart Search (Suche) oder Transfer + Search muß beachtet werden, daß hierbei nur das Quellport auf ein bestimmtes Byte durchgemustert wird.

Alle Datenoperationen müssen eine vereinbarte Blocklänge aufweisen.

Bei Blocklänge = 0 erfolgt ein Datentransfer von  $2^{16} + 1$  Byte.

Die kürzeste programmierte Blocklänge beträgt 1 Byte, wobei jedoch 2 Byte übertragen werden.

### Schreibregister WR 1

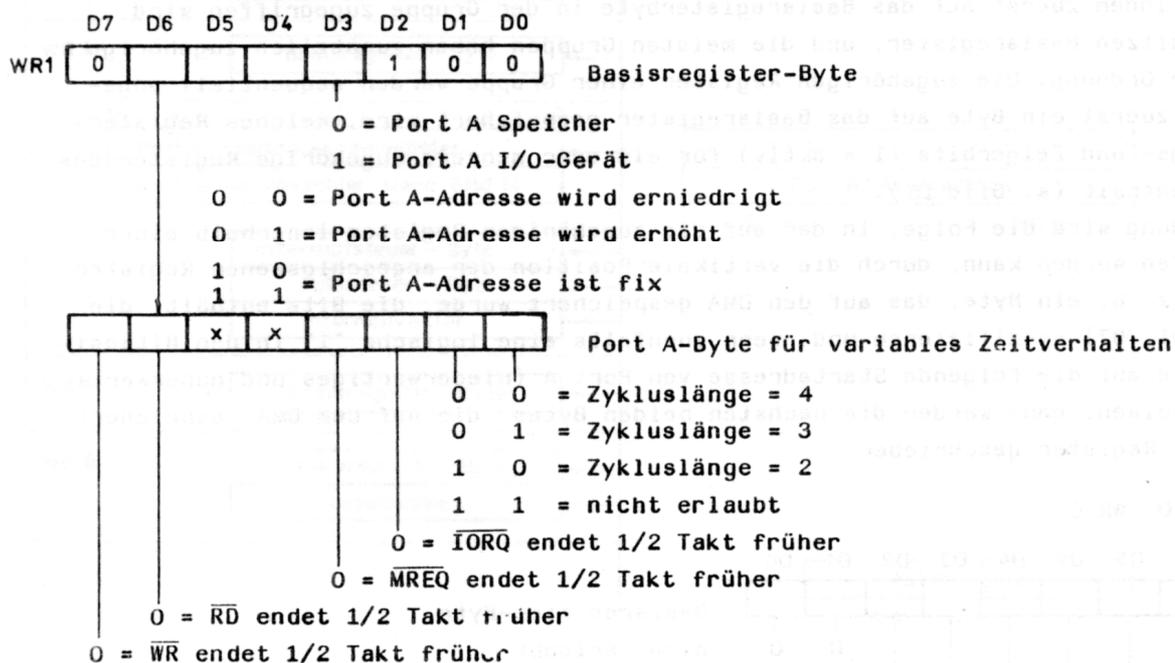


Bild 16

Im Schreibregister WR 1 wird der DMA-Kanal Port A hinsichtlich seiner Portkennzeichnung, Speicher oder E/A-Adresse, der Adreßgenerierung (INC/DEC/FIX) und seines Zeitverhaltens spezifiziert. Das Schreibregister WR 1 ist ein Basisregister, dem noch ein weiteres Zusatzregister angehört, das gegebenenfalls im Anschluß an WR 1 ausgegeben wird.

- D0/D1/D7 = 0 Schreibregisterkennzeichen WR 1
- D2 = 1 Portkennzeichnung Port A
- D3 = 0 Port A = Speicheradresse
- D3 = 1 Port A = E/A-Adresse
- D4/D5 - legen entsprechend Bild 16 fest, ob die Startadresse des Port A inkrementiert, dekrementiert oder fest bleiben soll.
- D6 = 1 legt fest, daß das nachfolgende Byte als Zeitsteuerbyte (Time-Control-Byte) im zugehörigen Register von WR 1 abzulegen ist.
- D6 = 0 kein variabler Zyklus - Standard U 880 Zeitverhalten



## Schreibregister WR 3

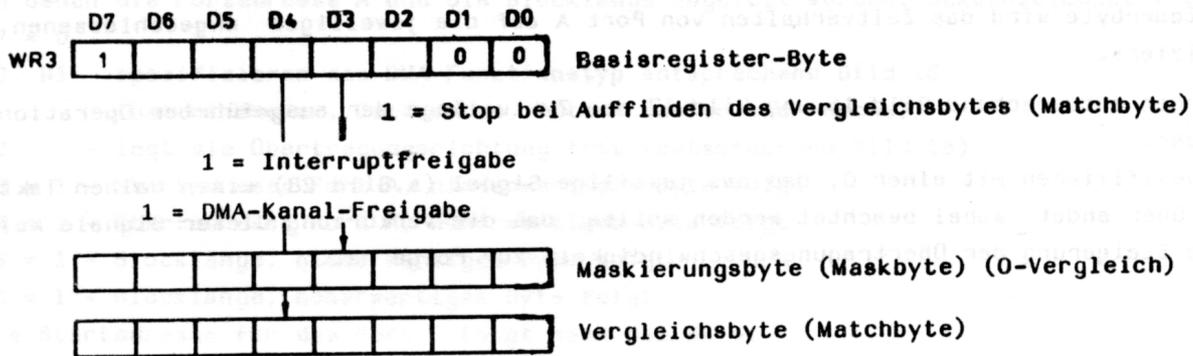


Bild 18

Das Schreibregister WR 3 ist ein Basisregister, welches zwei zugehörige, sequentiell adressierte Register (Maskierungsbyte, Mask-Byte/Vergleichsbyte, Match-Byte) besitzt. WR 3 dient im wesentlichen zur Freigabe des Datenverkehrs.

D0/D1 = 0 } Kennzeichnung WR 3  
D7 = 1 }

- D2 = 1 bewirkt ein Stoppen des DMA-Verkehrs bei Search oder Transfer + Search, wenn das Matchbyte gefunden wurde.
- D2 = 0 DMA-Verkehr wird erst am Blockende unterbrochen, wobei ein im Block gefundenes bzw. nicht gefundenes Matchbyte im Statusbyte und/oder im modifizierten Interruptvektor angezeigt wird;
- D3 = 1 spezifiziert als erstes nachfolgendes Byte das Mask-Byte.
- D4 = 1 spezifiziert als zweites nachfolgendes Byte das Match-Byte.
- D5 = 1 Interruptfreigabe  $\hat{=}$  dem Befehl AB im WR6 (enable Interrupts);
- D6 = 1 DMA-Freigabe  $\hat{=}$  dem Befehl 87 im WR6 (enable DMA);

### Maskierungsbyte (Mask-Byte)

In den Betriebsarten Search oder Transfer+Search kann ein bestimmter Speicherbereich auf ein spezielles Byte hin durchsucht werden, wobei mit dem Mask-Byte die Möglichkeit besteht, im Match-Byte nur bestimmte Bitpositionen zu testen. Durch eine 0 im Mask-Byte werden die Bitpositionen gekennzeichnet, die zum Vergleich mit dem Match-Byte herangezogen werden sollen. Bei vollständiger Maskierung (Mask-Byte = FF) wird jedes eingelesene Byte als das gesuchte Match-Byte gewertet.

### Match-Byte

Das Match-Byte, welches das zweite Byte ist, das unmittelbar auf WR 3 folgt, enthält das Byte, nach dem ein bestimmter Speicherbereich durchgemustert wird.

## Schreibregister WR 4

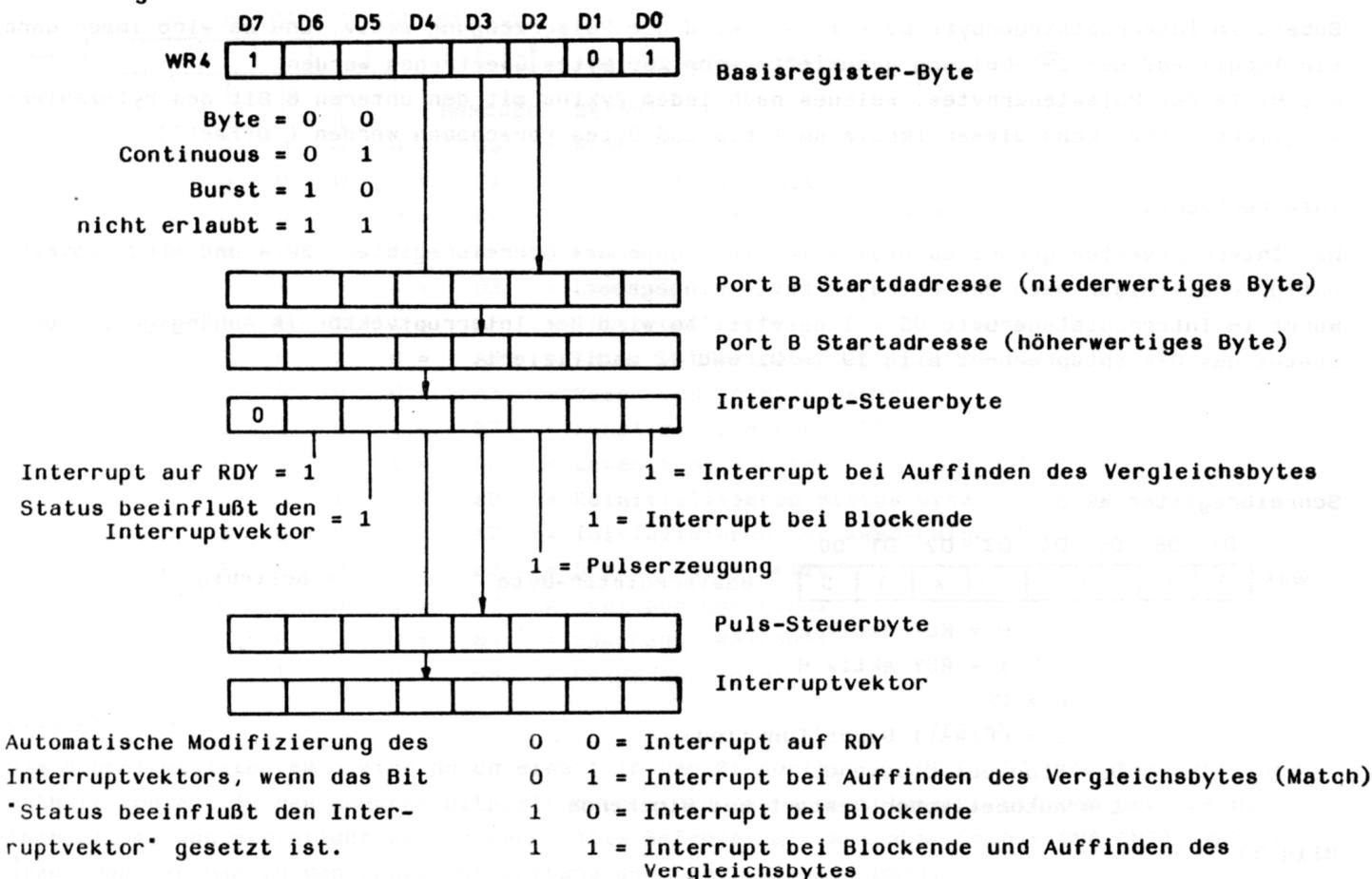


Bild 19

Das Schreibregister WR 4 ist ein Basisregister, welches fünf weitere zugehörige Register sequentiell adressieren kann. Durch  $D0 = 1 / D1 = 0$  und  $D7 = 1$  wird dieses Basisregister gekennzeichnet. In ihm wird im wesentlichen die Betriebsart (Byte, Burst, Continuous) und das Interruptverhalten spezifiziert. Weiterhin wird in zwei der zugehörigen Register die Startadresse von Port B eingetragen.

- |          |   |
|----------|---|
| $D0 = 1$ | } Kennzeichnung WR 4  |
| $D1 = 0$ |   |
| $D7 = 1$ |   |
| $D2 = 1$ | Startadresse Port B (niederwertiger Teil) folgt   |
| $D3 = 1$ | Startadresse Port B (höherwertiger Teil) folgt  |
| $D4 = 1$ | Interruptsteuerbyte (Interrupt-Control-Byte) folgt  |
| $D5/D6$  | spezifizieren entsprechend Bild 19 die Betriebsart, wobei $D5/D6$ nicht gleichzeitig eine 1 enthalten dürfen. |

### Interruptsteuerbyte (Interrupt-Control-Byte)

Im Interruptsteuerbyte wird das Interruptverhalten des Bauelementes festgelegt.

- |          |  |
|----------|--|
| $D0 = 1$ | Interruptanforderung, sobald das Match-Byte gefunden wurde                               |
| $D1 = 1$ | Interruptanforderung am Ende eines Datenblockes  |
| $D2 = 1$ | Pulsenergie einschalten  |
| $D3 = 1$ | Puls-Steuerbyte folgt  |
| $D4 = 1$ | Interruptvektor folgt  |
| $D5 = 1$ | Status beeinflusst Vektor  |
| $D6 = 1$ | Interruptanforderung, bevor der DMA-Schaltkreis den Bus anfordert, wenn Ready aktiv wird |

**Pulssteuerbyte**

Sobald im Interruptsteuerbyte D2 = 1 ist, wird die Pulserzeugung aktiv, und es wird immer dann ein Impuls auf der INT-Leitung generiert, wenn 256 Bytes übertragen werden. Mit Hilfe des Pulssteuerbytes, welches nach jedem Zyklus mit den unteren 8 Bit des Bytezählers verglichen wird, kann dieser Impuls um 1 bis 255 Bytes verschoben werden ("Offset").

**Interruptvektor**

Der Interruptvektor gehört ebenfalls in die Gruppe des Schreibregisters WR 4 und wird, soweit nötig, unmittelbar nach dem Pulssteuerbyte eingegeben. Wurde im Interruptsteuerbyte D5 = 1 gesetzt, so wird der Interruptvektor in Abhängigkeit vom Status des DMA entsprechend Bild 19 in D1 und D2 modifiziert.

**Schreibregister WR 5**

	D7	D6	D5	D4	D3	D2	D1	D0	
WR5	1	*				*	1	0	Basisregister-Byte * beliebig

0 = RDY aktiv L

1 = RDY aktiv H

0 =  $\overline{CE}$

1 =  $\overline{CE}/\overline{WAIT}$  Doppelfunktion

0 = Stop bei Blockende

1 = automatischer Restart bei Blockende

Bild 20

Das Schreibregister WR 5 ist ebenfalls ein Basisregister, hat jedoch im Gegensatz zu den anderen Basisregistern keine weiteren zugehörigen Register, die sequentiell adressiert werden. Es dient im wesentlichen zur Festlegung des aktiven Pegels von Ready, des Verhaltens am Blockende sowie zur Spezifizierung des  $\overline{CE}$ -Pins. Gekennzeichnet ist dieses Basisregister durch

D0 = 0

D1 = 1

D2 = beliebig

D6 = beliebig

D7 = 1

Schreibregister WR 6

D7	D6	D5	D4	D3	D2	D1	D0	
1						1	1	Basisregister-Byte
								Hexcode
	1	0	0	0	0	0	0	= C3 = Reset
	1	0	0	0	1			= C7 = Reset Zeitverhalten Port A
	1	0	0	1	0			= CB = Reset Zeitverhalten Port B
	1	0	0	1	1			= CF = Laden
	1	0	1	0	0			= D3 = Continue
	0	1	0	1	1			= AF = Interruptsperre
	0	1	0	1	0			= AB = Interruptfreigabe
	0	1	0	0	0			= A3 = Reset und Interruptsperre
	0	1	1	0	1			= B7 = Kanalfreigabe nach RETI
	0	1	1	1	1			= BF = Lesen Status Byte
	0	0	0	1	0			= 8B = Reinitialisieren Status Byte
	0	1	0	0	1			= A7 = Initialisieren der Read-Folge
	0	1	1	0	0			= B3 = Setzen RDY = aktiv
	0	0	0	0	1			= 87 = Freigabe DMA-Kanal
	0	0	0	0	0			= 83 = Sperrung DMA-Kanal
	0	1	1	1	0			= BB = Lesemaske folgt

Bild 21

Das Schreibregister WR 6 wird durch eine 1 in den Bitpositionen D0/D1/D7 gekennzeichnet und enthält kodiert in den anderen Bitpositionen einen Zustandsschlüssel, mit dem der DMA nach Initialisierung beeinflusst werden kann. Alle Befehle, mit Ausnahme Enable DMA (87), führen dazu, daß der DMA in den inaktiven Zustand übergeht (Disable DMA).

C3 Reset

- Durch diesen Befehl werden im DMA folgende Aktivitäten ausgelöst:
  - Rücksetzen der Interruptlogik (disable)
  - Rücksetzen der Busrequestlogik (disable)
  - Rücksetzen der Autorepeatfunktion (WR 5)
  - Rücksetzen der WAIT-Funktion (WR 5)
  - Initialisierung des Zeitverhaltens von Port A und Port B auf ein Standard-U 880-Zeitverhalten (WR 1/WR 2)
  - Rücksetzen von Force-Ready
- Um ein fehlerfreies Arbeiten des DMA zu gewährleisten, sollte am Anfang der Initialisierungsroutine einmal Reset auf den DMA gegeben werden. Bei Abbruch einer Operation sollten, um den DMA wirksam zurückzusetzen, sechs Reset-Befehle auf den Baustein ausgegeben werden. Die sechs Reset-Befehle nach einem Abbruch sind notwendig, da das Schreibregister WR4 fünf zugehörige Register besitzt, bei denen die Möglichkeit besteht, daß der Reset-Befehl falsch interpretiert wird.
- Da durch den Befehl C3 in WR 6 des DMA nicht dazu veranlaßt wird, die Read-Folge (Readsequence) zurückzusetzen, ist es im gegebenen Fall erforderlich, die Read-Folge durch den Befehl A7 = (Initialisierung der Read-Folge) zurückzusetzen.

Reset

Timing Port A  
C7

setzt die Zykluszeit T, die im Schreibregister WR 1 und im Zeitsteuerbyte (Time-Control-Byte) für Port A vereinbart wurde, auf T = 3 (entspricht dem Standard-U 880-Zeitverhalten) (Für I/O wird T = 4 gesetzt).

Reset

Timing Port B  
CB

siehe Reset, Timing Port A

Laden CF	Durch den Befehl CF werden die Startadressen beider Ports (A und B) in die entsprechenden Arbeitsregister geladen, der Bytezähler auf Null gesetzt und eine mögliche Force-Ready-Bedingung zurückgesetzt. Bevor der Befehl CF ausgegeben wird, müssen die Startadressen von Port A und Port B im WR 0 bzw. WR 4 eingetragen werden.
Conitinue D3	Dieser Befehl setzt den Bytezähler auf Null und startet die Operation der DMA beim augenblicklichen Adreßstand neu.
Disable Interrupts AF	AF sperrt anstehende Interrupts. Das Interrupt-Under-Service-Latch wird nicht zurückgesetzt.
Enable Interrupts AB	AB gibt das Interruptsystem des DMA frei, d. h., in WR 3 wird Bit 5 auf 1 gesetzt.
Reset and Disable Interrupts A3	Durch diesen Befehl werden folgende Aktivitäten im DMA ausgelöst: <ul style="list-style-type: none"> <li>- Zurücksetzen des Interrupt-Under-Service-Latches (IUS)</li> <li>- Zurücksetzen des Interrupt-Pending-Latches (IP)</li> <li>- Zurücksetzen einer internen Force-Ready-Bedingung</li> <li>- Sperren weiterer Interrupts durch den DMA (s. Disable Interrupts)</li> </ul> Der Befehl "Reset and Disable Interrupts" gefolgt vom Befehl "Enable Interrupts" ersetzt den U 880-RETI-Befehl und wird vorrangig in DMA-Konfigurationen ohne U 880-CPU verwendet. In U 880-Systemen ist er nicht erforderlich.
Enable nach RETI B7	Dieser Befehl wird nur benutzt, wenn der DMA in WR 4 im Interruptmode "Interrupt bei Ready" programmiert wurde. Durch den DMA wird keine Busanforderung ausgelöst, solange das RETI-Signal nicht empfangen wurde.
Read STATUS Byte BF	Das nächste Byte, das vom DMA durch die CPU gelesen wird, ist das Status-Byte.
Reset STATUS	Im Status-Byte werden Bit 4 (Match-Byte) und Bit 5 (Blockende) zurückgesetzt. Zurücksetzen von Bit 3 (Interrupt angemeldet/Interruptpending) wird durch eine Interruptbestätigung, Abarbeitung der Interruptserviceroutine und durch den Befehl "Reset and Disable Interrupts" erreicht. Bit 0 (DMA-Operation hat stattgefunden) wird durch den Befehl LOAD (CF) zurückgesetzt. Der Befehl 8B, gefolgt von "87" (enable DMA) ist nötig für ein "Weiterlaufen" des DMA nach gefundenem Match.
Lesemaske folgt BB	BB spezifiziert, daß das nächste Byte, welches in den DMA eingeschrieben wird, eine Lesemaske ist, die eine Folge mehrerer CPU-lesbarer Register (Read-Register) bestimmt. Die Read-Register RRO bis RR6 werden in einer festen Folge, beginnend mit RRO und endend mit RR6, gelesen. Welches der sechs Register gelesen werden kann, wird durch eine 1 in den entsprechenden Bitpositionen der Lesemaske festgelegt (s. Bild 22).
Initialisierung der Readfolge A7	Das erste lesbare, durch die Lesemaske spezifizizierte Register wird beim nächsten Lesezyklus gelesen.

FORCE READY  
(erzwungenes Ready)  
B3

Dieser Befehl wirkt in der Weise, daß eine interne Ready-Bedingung, unabhängig vom logischen Zustand der Ready-Leitung, als aktiv betrachtet wird und eine DMA-Operation auch mit inaktivem Ready vollzogen werden kann. Er wird vorwiegend in den Betriebsarten Burst oder Continuous bei Speicher-Speicher-Übertragungen oder bei Speicher-Such-Operationen eingesetzt.

Force-Ready wird durch folgende Befehle und Zustände zurückgesetzt:

- C3 Reset
- CF LOAD
- A3 Reset and Disable Interrupt
- bei Blockende
- bei einem gefundenen Match-Byte
- bei Busfreigabe durch den DMA

Wird der Befehl B3 im Byte-Mode gegeben, so erfolgt lediglich die Übertragung eines Bytes; die anschließende Busfreigabe setzt die interne RDY-Bedingung zurück, sodaß vor jedem zu übertragenden Byte der Befehl B3 erneut gegeben werden müßte.

Enable DMA  
87

Durch diesen Befehl werden, vorausgesetzt Ready ist aktiv, alle Operationen des DMA, mit Ausnahme des Interrupts, freigegeben.

Im Zusammenhang mit Bit 6 vom WR 3 (DMA Enable) ist dieser Befehl der einzige, der den DMA nicht dazu veranlaßt, den inaktiven Zustand einzunehmen.

Enable DMA gibt die Busanforderungslogik des Bausteins frei, setzt aber nichts zurück.

Disable DMA  
83

Disable DMA sperrt die Busanforderungslogik.

Alle Operationen, außer Interrupts, werden gesperrt.

## 3.2. Leseregister (RRO - RR6)

Die Leseregister RRO bis RR6 werden durch Adressieren der DMA als E/A-Einheit gelesen. Die lesbaren Bytes enthalten den DMA-Status, Bytezählerstand und die Portadressen seit dem letzten DMA-Reset.

Die Register werden immer in einer feststehenden Folge - beginnend mit RRO und endend mit RR6 - gelesen. Die in dieser Folge gelesenen Register werden durch Programmierung der Lesemaske in WR6 (BB) spezifiziert. Die Lesefolge wird durch den Befehl "Initiate Read Sequence" (A7) initialisiert. Das mit (BB) ins Lesemaskenregister (WR6) eingeschriebene Byte wird in das Zeigerregister übernommen, das den aktuellen Lesestand enthält. Nach einem DMA-Reset muß die Folge mit dem Befehl "Initiate Read Sequence" neu initialisiert werden.

### Read-Mask-Register (Lesemaske)

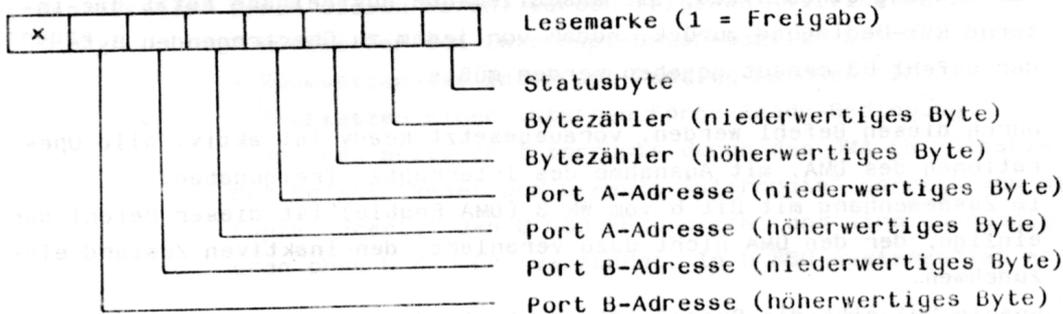


Bild 22

Durch den Befehl BB in WR6 (Lesemaske folgt) wird der DMA darauf vorbereitet, daß das nächste einzulesende Byte den Inhalt der Lesemaske darstellt.

Durch eine 1 in den entsprechenden Bitpositionen (s. Bild 22) wird ein Lesegesuch für die jeweiligen Register ausgegeben, welche danach gelesen werden können.

Die Lesefolge ist danach automatisch initialisiert.

Bit 7 im Maskenregister hat keinen Einfluß und kann demzufolge beliebig programmiert werden.

### Statusregister (RRO)

Inhalt des Status-Bytes

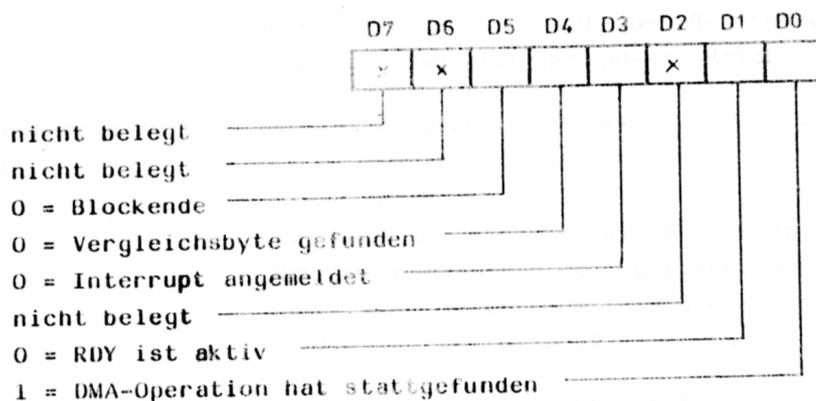


Bild 23

Das Leseregister RRO entspricht dem Statusregister des DMA, welches in der feststehenden Lesefolge am Anfang steht. Das Statusregister kann unabhängig von den anderen Leseregistern gelesen werden.

- Bit 0 zeigt, ob der DMA nach dem letzten LOAD (CF) den Bus angefordert hatte.  
D0 = 1 ja  
D0 = 0 nein
- Bit 1 zeigt den logischen Zustand der Ready-Leitung, deren aktiver Pegel in WR5/D3 festgelegt wurde, an.  
D1 = 1 Ready inaktiv  
D2 = 0 Ready aktiv
- Bit 2 nicht definiert (beliebig)
- Bit 3 zeigt den Status des Interrupt-Pending-Latches an.  
D3 = 0 Interrupt angemeldet, aber noch nicht bearbeitet  
D3 = 1 kein Interrupt angemeldet
- Bit 4 zeigt an, ob seit dem letzten DMA-Reset ein Match-Byte gefunden wurde.  
D4 = 1 kein Match-Byte gefunden  
D4 = 0 Match-Byte gefunden
- Bit 5 zeigt an, ob nach dem letzten DMA-Reset oder LOAD (CF) oder CONTINUE oder Reset-STATUS ein Blockende erreicht wurde.  
D5 = 0 Blockende erreicht  
D5 = 1 Blockende nicht erreicht
- Bit 6 nicht definiert
- Bit 7 nicht definiert

## 11. Zeitverhalten des DMA

### 11.1. Inaktiver Zustand

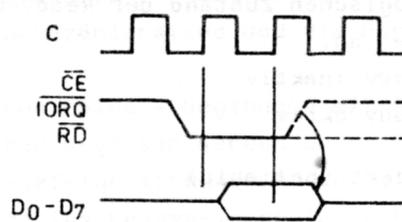
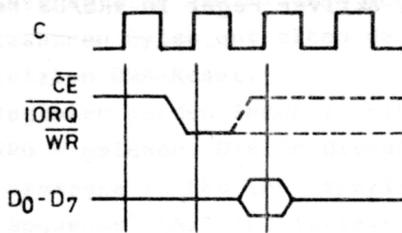


Bild 24

Bild 25

Im inaktiven Zustand können jederzeit Steuerbytes auf den DMA geschrieben bzw. Status-Bytes und entsprechende Leseregister RRO - RR6 gelesen werden.

In diesem Fall wird der DMA durch die CPU als E/A-Garät adressiert.

#### Schreiben von Steuerbytes auf den DMA

Um Steuerbytes auf den DMA zu schreiben, müssen folgende Bedingungen erfüllt werden (s. Bild 24):

- Buskontrolle durch die CPU
- $\overline{CE}$ -Leitung = aktiv (Low) Diese Bedingung wird im Regelfall durch einen Adreßdeko­der für das untere Adreßbyte des Adreßbusses erreicht.
- $\overline{IORQ}$  und  $\overline{WR}$  müssen entsprechend  $\overline{CE}$  zur selben Zeit aktiv sein.
- Das auf den DMA zu schreibende Steuerbyte muß zeitlich so auf den Bus gegeben werden, daß nach dem Einschwingen der Signale  $\overline{CE}/\overline{IORQ}$  und  $\overline{WR}$  mit der nächsten steigenden Tak­tf­lanke die Daten gültig sind (s. Bild 24). Bei Einsatz eines U 880 D als CPU werden diese Bedingungen automatisch gewährleistet.

#### Lesen von Status-Bytes (RRO - RR6)

Sollen vom DMA die lesbaren Register (RRO - RR6) gelesen werden, so muß entsprechend Bild 25 die Erfüllung folgender Bedingungen gewährleistet sein:

- Buskontrolle durch die CPU
- $\overline{CE}$  = aktiv (Low) (s. Schreiben von Steuerbytes)
- $\overline{IORQ}$  und  $\overline{RD}$  müssen entsprechend Bild 25 aktiv sein.

Diese Operationen erfordern weniger als drei T-Zyklen. Die  $\overline{CE}$ -,  $\overline{IORQ}$ - und  $\overline{RD}$ -Leitungen werden über zwei Anstiegsflanken von C aktiviert, und die Daten erscheinen einen T-Zyklus nach der Aktivierung o. g. Signale auf dem Bus.

## 11.2. Aktiver Zustand

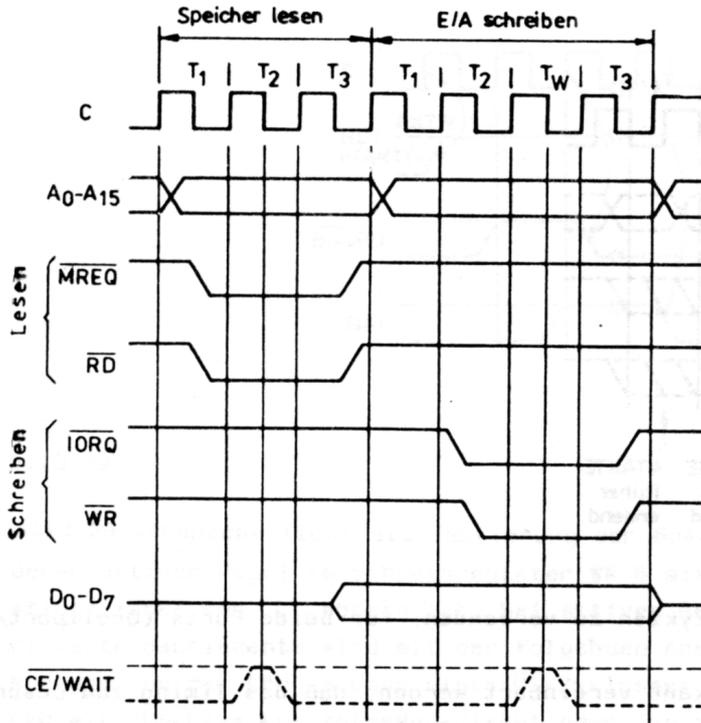


Bild 26

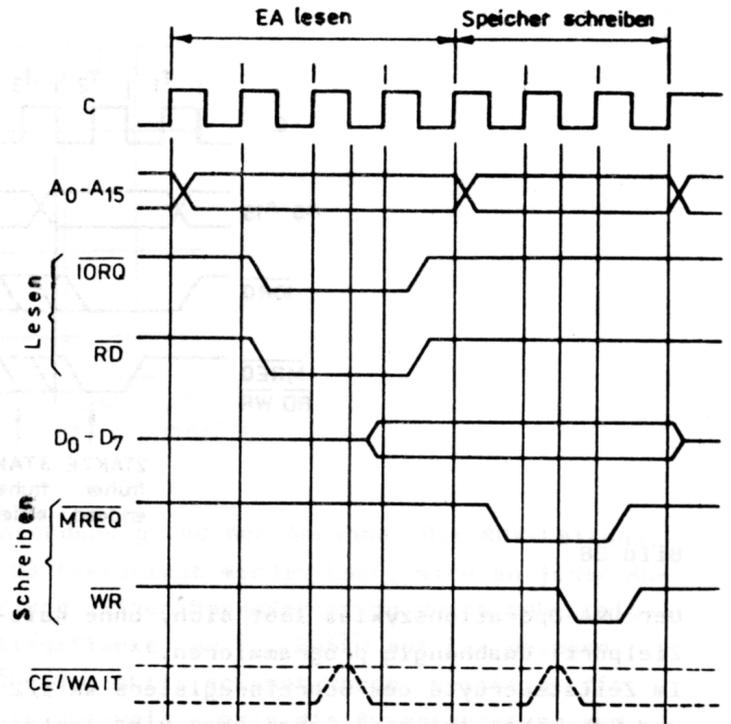


Bild 27

### Standard Lesezyklus/Schreibzyklus

Im Regelfall, d. h. in Systemkonfigurationen mit der CPU UA 880 D / UB 880 D bzw. nach dem Rücksetzen der DMA wird das Zeitverhalten, welches im Zeitsteuerbyte des Schreibregisters WR 1/2 spezifiziert wurde auf Standardwerte gesetzt und stimmt somit für Lese- und Schreibzyklen bei Speicher- bzw. bei I/O-Operationen mit dem Zeitverhalten des U 880 D überein.

Dabei gibt es eine Ausnahme; die Daten werden mit der fallenden Flanke von T3 übernommen und auf den Datenbus, über die Grenze zwischen Lese- und Schreibzyklen hinaus, bis zum Ende des folgenden Schreibzyklusses gehalten.

Bild 26 stellt die Taktierung für Übertragungen vom Speicher auf E/A-Geräte dar und Bild 27 veranschaulicht die Übertragung von E/A-Geräten auf Speicher.

Übertragungen Speicher - Speicher oder E/A-Gerät - E/A-Gerät lassen sich einfach aus den Diagrammen in Bild 26 und Bild 27 ableiten.

Die Standard-Taktierung verwendet drei T-Zyklen für Speicheroperationen. Für E/A-Operationen werden vier T-Zyklen benötigt, da zwischen T2 und T3 automatisch ein WAIT-Zyklus eingeschoben wird. Wenn die  $\overline{\text{CE/WAIT}}$ -Leitung so programmiert wurde, daß sie während des aktiven Zustandes der DMA als  $\overline{\text{WAIT}}$ -Leitung wirkt, wird sie für Speicheroperationen auf die fallende Flanke von T2 und für E/A-Operationen auf die fallende Flanke von TW abgetastet. Wenn  $\overline{\text{CE/WAIT}}$  während dieser Zeit Low ist, wird ein weiterer TW-Zyklus eingefügt. Somit kann eine Bedienung beliebig langsamer Speicher bzw. E/A-Geräte gewährleistet werden.

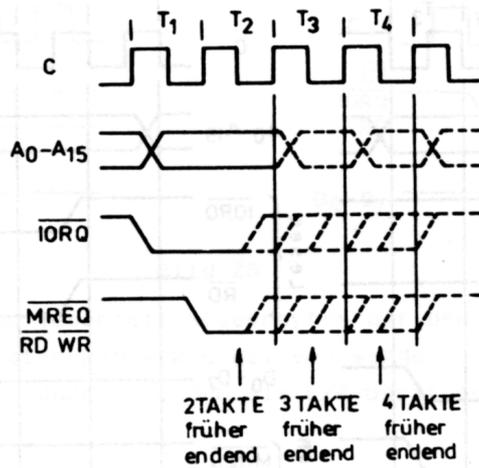


Bild 28

Der DMA-Operationszyklus läßt sich, ohne Wait-Zyklen zu verwenden, für beide Ports (Quellport/Zielpport) unabhängig programmieren.

Im Zeitsteuerbyte des Schreibregisters WR 1/2 kann vereinbart werden, daß das Timing zum Lesen und Schreiben aus zwei, drei oder vier Takten (mehr, wenn Wait-Zyklen eingefügt werden) bestehen kann. Weiterhin kann, ebenfalls im Zeitsteuerbyte des Schreibregisters WR 1/2 festgelegt, vereinbart werden, daß die Rückflanken der Steuersignale  $\overline{\text{IORQ}}$ ,  $\overline{\text{MREQ}}$ ,  $\overline{\text{RD}}$ ,  $\overline{\text{WR}}$  unabhängig voneinander, entsprechend Bild 28, einen halben Takt früher beendet werden.

Bei variablem Zyklus wird  $\overline{\text{IORQ}}$ , im Gegensatz zum Standard-Timing der DMA einen halben Taktzyklus vor  $\overline{\text{MREQ}}$ ,  $\overline{\text{RD}}$  und  $\overline{\text{WR}}$  aktiv.

$\overline{\text{CE}}/\overline{\text{WAIT}}$  kann nur dann zur Verlängerung entsprechender Speicher- oder E/A-Zyklen verwendet werden, wenn im variablen Zyklus bei Speicheroperationen 3 oder 4 T-Zyklen und bei E/A-Operationen 4 T-Zyklen programmiert wurden.  $\overline{\text{CE}}/\overline{\text{WAIT}}$  wird bei Speicheroperationen mit 3 oder 4 T-Zyklen auf die fallende Flanke von T2 abgetastet, während bei E/A-Operationen mit 4 Taktzyklen auf die fallende Flanke von TW abgetastet wird.

Bei Übertragungen werden die Daten auf die Taktflanke "gelatcht", die die Ausstiegsflanke von  $\overline{\text{RD}}$  verursacht und bis zum Ende des Schreibzyklusses gehalten (Bild 27).

## 11.2.2. Busanforderungen und Quittung

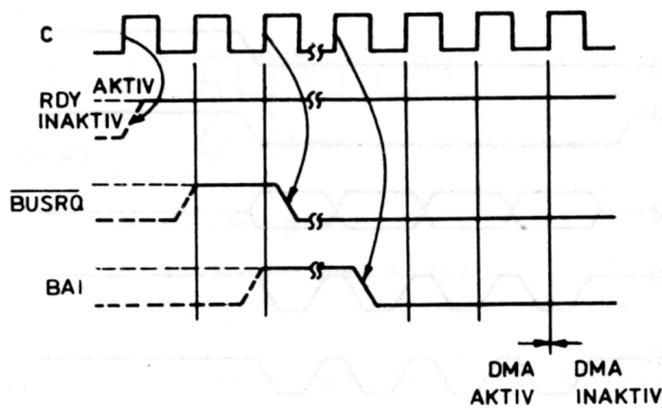


Bild 29

Bild 29 veranschaulicht die Taktierung der Busanforderung und der Annahme. Die RDY-Leitung, deren aktiver Pegel im Schreibregister WR 5 mit D3 festgelegt werden kann, wird an jeder Anstiegsflanke von C abgetastet. Bei aktivem RDY und keiner Busanforderung durch höherpriorisierte Bauelemente wird mit der folgenden Anstiegsflanke von C  $\overline{\text{BUSRQ}}$  auf Low gezogen. Nachdem von der CPU am Ende eines  $\overline{\text{M1}}$ -Zyklus  $\overline{\text{BUSRQ}}$  = aktiv erkannt wurde, bestätigt die CPU mit  $\overline{\text{BUSACK}}$  = Low entweder direkt oder über die DMA-Daisy-Chain die Busanforderung und floatet gleichzeitig ihre Tristate-Ausgänge.

Im Burst- oder Byte-Mode muß bei der Busübernahme der DMA die RDY-Leitung aktiv sein. RDY = aktiv wird nicht als Flanke, sondern als Pegel erkannt. Der einzige Fall, in dem ein Impuls auf der RDY-Leitung ausreicht den DMA zur Busübernahme zu zwingen, ist der Continuous-Mode. Das heißt, wenn der Continuous-Mode RDY bei der Busübernahme des DMA inaktiv ist, übernimmt der DMA die Buskontrolle, führt aber keine Datenoperation durch.

Wenn an  $\overline{\text{BAI}}$  für zwei aufeinanderfolgende Anstiegsflanken von C ein Low festgestellt wird, beginnt der DMA - vorausgesetzt RDY ist aktiv - mit Datenoperationen.

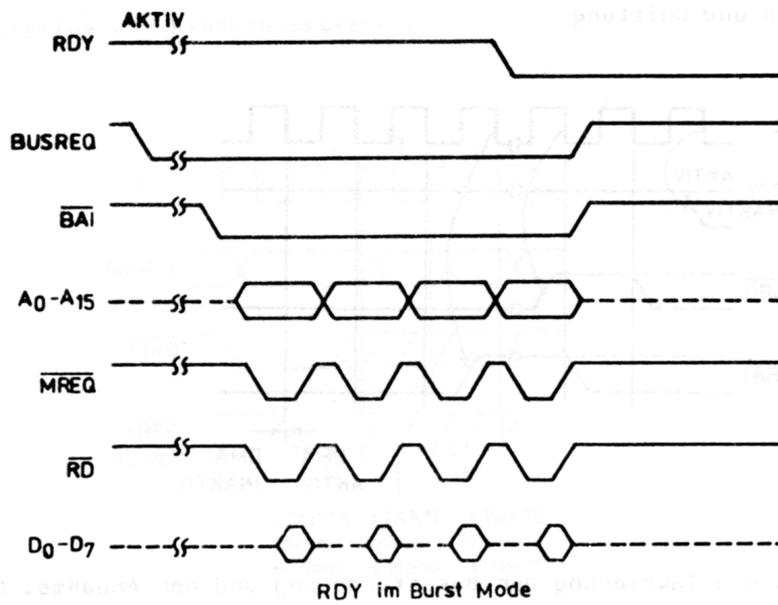


Bild 30

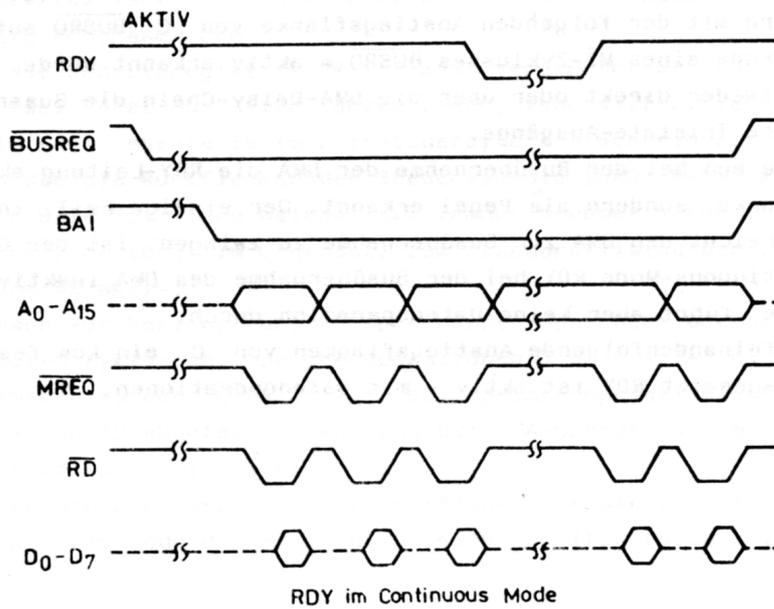


Bild 31

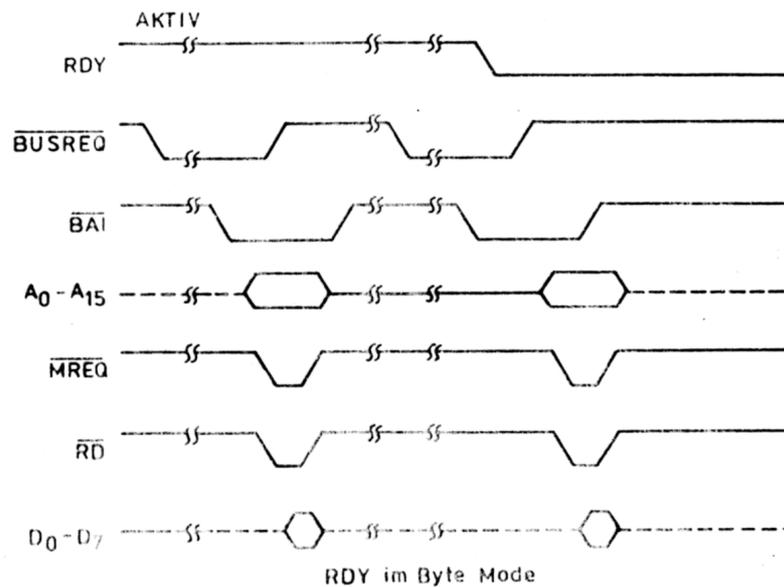


Bild 32

### 11.2.3. Busfreigabe

#### Busfreigabe im Byte-Mode (Byte at a time)

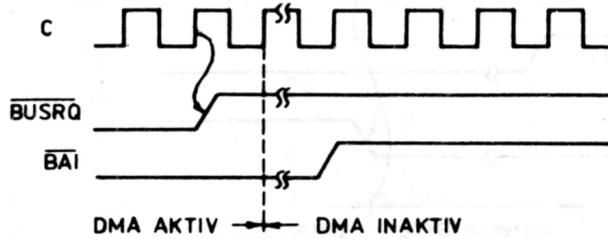


Bild 33

In der Betriebsart Byte-Mode (Search) wird  $\overline{\text{BUSRQ}}$  auf die Anstiegsflanke von C vor dem Ende eines jeden Lesezyklusses inaktiv, während bei den Funktionstypen Transfer und Transfer + Search  $\overline{\text{BUSRQ}}$  auf die Anstiegsflanke von C vor dem Ende eines jeden Schreibzyklusses inaktiv wird (s. Bild 33).

Die Aktivitäten bei der Busfreigabe im Byte-Mode geschehen unabhängig vom Zustand der RDY-Leitung. Einen Taktzyklus nachdem die CPU  $\overline{\text{BUSRQ}} = \text{High}$  erkannt hat, übernimmt sie den Bus bis zur nächsten Busanforderung durch den DMA, der nur aktiv wird, wenn sowohl  $\overline{\text{BUSRQ}}$  als auch BAI ( $\overline{\text{BUSACK}}$ ) zuvor auf High waren.

#### Busfreigabe bei Blockende

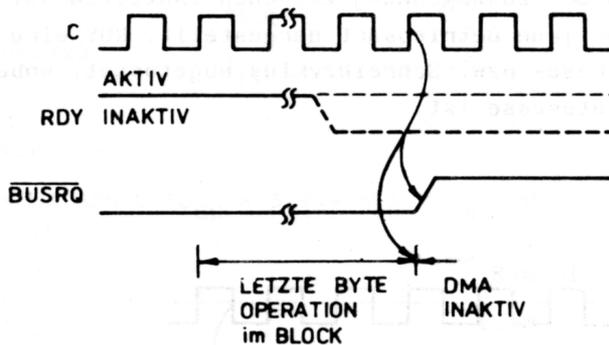


Bild 34

In den Betriebsarten Burst und Continuous wird durch Erreichen des Blockendes veranlaßt, daß  $\overline{\text{BUSRQ}}$  mit der selben Anstiegsflanke von C inaktiv wird, mit der die Übertragung des Datenblockes abgeschlossen wurde (s. Bild 34).

Das letzte Byte des Datenblockes wird auch dann noch übertragen, wenn RDY vor Beendigung der Übertragung des letzten Bytes inaktiv wird.

## Busfreigabe bei Not Ready

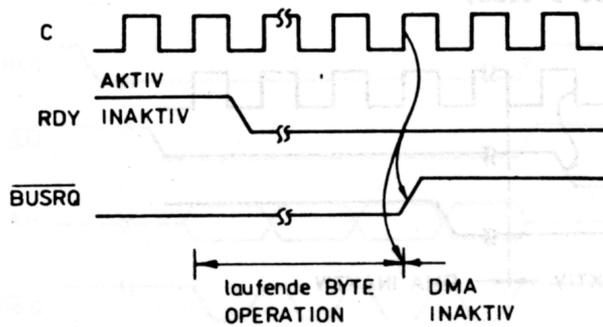


Bild 35

Im Burst-Mode wird durch ein inaktives RDY die laufende Byteoperation vollständig abgeschlossen und  $\overline{\text{BUSRQ}}$  wird mit der nächsten steigenden Flanke, entsprechend Bild 35, inaktiv. Das heißt, im Search-Mode wird  $\overline{\text{BUSRQ}}$  mit der nächsten steigenden Flanke nach dem letzten Lesezyklus im Transfer-Mode bzw. im Transfer+Search-Mode mit der nächsten steigenden Flanke nach dem letzten Schreibzyklus der DMA inaktiv. Die steigende Flanke von  $\overline{\text{BUSRQ}}$  wird durch den notwendigen, vollständigen Abschluß der momentanen Byteoperation gegenüber RDY = inaktiv etwas verzögert.

Im Continuous-Mode wird  $\overline{\text{BUSRQ}}$ , im Gegensatz zum Burst-Mode, durch ein inaktives RDY nicht freigegeben. Stattdessen bricht der DMA nach Beendigung der letzten Byteoperation seine Aktivitäten ab und befindet sich im Wartezustand, wobei er wieder auf ein aktives RDY wartet, um seine Operation fortzusetzen.

In den Bildern 30, 31 und 32 wird der Zusammenhang zwischen inaktivem RDY und den betreffenden anderen Steuerleitungen für jede Betriebsart dargestellt. RDY wird mit der steigenden Flanke von C nach jedem Lese- bzw. Schreibzyklus abgetastet, wobei der Pegel von RDY und nicht seine Flanken von Interesse ist.

## Busfreigabe bei Match

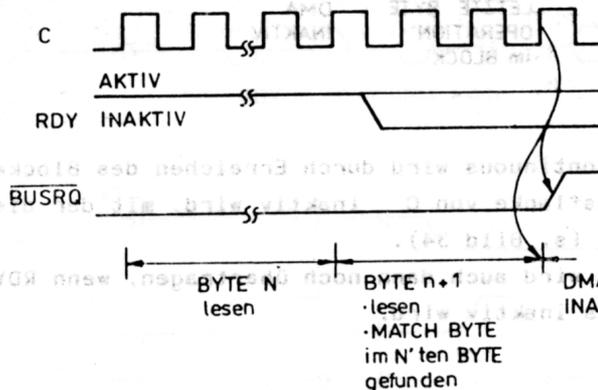


Bild 36

Bei Match im Burst-Mode oder im Continuous-Mode bricht der DMA seine Aktivitäten auf dem Bus ab, indem  $\overline{\text{BUSRQ}}$  bei der nächsten DMA-Operation entsprechend Bild 36 inaktiv wird. Das heißt, am Ende der nächsten Leseoperation im Search-Mode bzw. am Ende der folgenden Schreiboperation beim Transfer wird  $\overline{\text{BUSRQ}}$  mit der nächsten ansteigenden Flanke von C High. Infolge der gepufferten Arbeitsweise werden die Matches erkannt, während schon die nächsten Lese- und Schreiboperationen vom DMA ausgeführt werden.

RDY kann inaktiv werden, nachdem die Vergleichsoperation beginnt, ohne dieses Busfreigabe-Timing zu beeinflussen.

## 12. Technische Daten

### 12.1. Betriebsbedingungen UA 858 D

Parameter	Symbol	min	max	Einheit
Betriebstemperatur	$\vartheta_a$	0	70	°C
Betriebsspannung	$U_{CC}$	4,75	5,25	V
Eingangsspannung	$U_{IL}$	-0,5	0,8	V
Eingangsspannung	$U_{IH}$	2,0	$U_{CC}$	V
Takteingangsspannung	$U_{ILC}$	-0,5	0,45	V
Takteingangsspannung	$U_{IHC}$	$U_{CC}-0,6$	$U_{CC}$	V

#### 12.1.1. Statische Kennwerte ( $U_{CC} = 5 \text{ V} \pm 5 \%$ ) ( $\vartheta_a = 0 \dots 70 \text{ °C}$ )

Parameter	Symbol	min	max	Einheit	Meßbedingung
Ausgangsspannung, Low-Pegel	$U_{OL}$		0,4	V	$I_{OL} = 1,8 \text{ mA}$
Ausgangsspannung, High-Pegel	$U_{OH}$	2,4		V	$I_{OH} = -250 \text{ } \mu\text{A}$
Stromaufnahme	$I_{CC}$		200	mA	$\vartheta_a = 25 \text{ °C}$
Eingangsreststrom	$I_{LI}$		$\pm 10$	$\mu\text{A}$	$U_I = 0 \text{ V} \dots U_{CCMAX}$
Reststrom der Dreizustandsausgänge im hochohmigen Zustand	$I_{LOH}$		20	$\mu\text{A}$	$U_O = 5,25 \text{ V}$
	$I_{LOL}$		-20	$\mu\text{A}$	$U_O = 0 \text{ V}$
Reststrom des Datenbusses bei Eingabe	$I_{LD}$		$\pm 20$	$\mu\text{A}$	$0 \text{ V} \leq U_I \leq U_{CCMAX}$

#### 12.1.2. Dynamische Kennwerte

Passiver Zustand des UA 858 D ( $U_{CC} = 5 \text{ V} \pm 5 \%$   $\vartheta_a = 0 \dots 70 \text{ °C}$ ;  $C_L = 100 \text{ pF}$ )

Signal	Parameter	Symbol	min	max	Einheit	Nr.
C	Taktperiode	$t_c$	250	4000	ns	1
	High-Breite des Taktes	$t_w(\text{CH})$	105	2000	ns	2
	Low-Breite des Taktes	$t_w(\text{CL})$	105	2000	ns	3
	Takt-Anstiegszeit	$t_r$	-	30 <sup>(2)</sup>	ns	4
	Takt-Abfallzeit	$t_f$	-	30 <sup>(2)</sup>	ns	5
	Haltezeit für spezifizierte Setzzeiten	$t_H$	0	-	ns	6/19
$\overline{\text{CE}}$ $\overline{\text{WR}}$ $\overline{\text{IORQ}}$	Setzzeit des Steuersignals bzgl. L/H des Taktes	$t_{sc}(\text{CE})$	145	-	ns	7
D0 - D7	Datenausgangsverzögerung bzgl. H/L von $\overline{\text{RD}}$	$t_{DR}(\text{D})$	-	380	ns	8
	Datensetzzeit bzgl. L/H des Taktes während Schreib- oder $\overline{\text{M1}}$ -Zyklus	$t_{SC}(\text{D})$	50	-	ns	9
	Datenausgangsverzögerung bzgl. H/L von $\overline{\text{IORQ}}$ während INTA-Zyklus	$t_{DI}(\text{D})$	-	160	ns	10
	Floatverzögerung bzgl. L/H von $\overline{\text{RD}}$	$t_F(\text{D})$	-	110	ns	11
IEI	IEI-Setzzeit bzgl. H/L von $\overline{\text{IORQ}}$ während INTA-Zyklus	$t_S(\text{IEI})$	140	-	ns	12

Signal	Parameter	Symbol	min	max	Einheit	Nr.
IEO	IEO-Verzögerung bzgl. L/H von IEI	$t_{DH(IEO)}$	-	160	ns	13
	IEO-Verzögerung bzgl. H/L von IEI	$t_{DL(IEO)}$	-	130	ns	14
	IEO-Verzögerung bzgl. H/L von $\overline{M1}$	$t_{DM(IEO)}$	-	190	ns	15
$\overline{M1}$	$\overline{M1}$ -Setzzeit bzgl. L/H des Taktes	$t_{SC(M1)}$	90	-	ns	16
	$\overline{M1}$ -Rücksetzzeit bzgl. H/L d. Taktes	$t_{RC(M1)}$	-10 <sup>(1)</sup>	-	ns	17
$\overline{RD}$	$\overline{RD}$ -Setzzeit bzgl. L/H des Taktes während $\overline{M1}$ -Zyklus	$t_{SC(RD)}$	115	-	ns	18
$\overline{INT}$	$\overline{INT}$ -Verzögerungszeit bzgl. der Interruptursache ( $\overline{INT}$ wird nur erzeugt, wenn DMA inaktiv ist)	$t_{D(IT)}$	-	500	ns	19
$\overline{BAO}$	$\overline{BAO}$ -Verzögerung bzgl. L/H von $\overline{BAI}$	$t_{DH(BO)}$	-	150	ns	20
	$\overline{BAO}$ -Verzögerung bzgl. H/L von $\overline{BAI}$	$t_{DL(BO)}$	-	150	ns	21
RDY	RDY-Setzzeit bzgl. L/H des Taktes	$t_{SC(RDY)}$	100	-	ns	22

(1) Negativer minimaler Setzzeit-Wert heißt, daß das 1. erwähnte Ereignis nach dem 2. Ereignis kommen kann.

Aktiver Zustand des UA 858 D ( $U_{CC} = 5\text{ V} \pm 5\%$ ;  $\vartheta_a = 0 \dots 70\text{ °C}$ ;  $C_L = 100\text{ pF}$ )

Signal	Parameter	Symbol	min	max	Einheit	Nr.
C	Taktperiode	$t_C$	250	4000	ns	1
	High-Breite des Taktes	$t_{w(CH)}$	105	2000	ns	2
	Low-Breite des Taktes	$t_{w(CL)}$	105	2000	ns	3
	Takt-Anstiegszeit	$t_r$	-	30 <sup>(2)</sup>	ns	4
	Takt-Abfallzeit	$t_f$	-	30 <sup>(2)</sup>	ns	5
A 0 bis A 15	Adressenausgangsverzögerung	$t_{D(AD)}$	-	110	ns	6
	Float-Verzögerung	$t_{F(AD)}$	-	90	ns	7
	stabile Adresse vor $\overline{MREQ}$ (Speicherzyklus)	$t_{acm}$	(4)	-	ns	8
	stabile Adresse vor $\overline{IORQ}$ , $\overline{RD}$ oder $\overline{WR}$ (I/O-Zyklus)	$t_{aci}$	(5)	-	ns	9
	stabile Adresse nach $\overline{RD}$ oder $\overline{WR}$	$t_{ca}$	(6)	-	ns	10
	stabile Adresse nach $\overline{RD}$ oder $\overline{WR}$ bei Übergang der Adreßbustreiber in den hochohmigen Zustand ("FLOAT")	$t_{caf}$	(7)	-	ns	11
D 0 bis D 7	Datenausgangsverzögerungszeit	$t_{D(D)}$	-	150	ns	12
	Verzögerungszeit bis zum hochohmigen Zustand (Float) während Schreibzyklus	$t_{F(D)}$	-	90	ns	13
	Datensetzzeit bzgl. L/H des Taktes während d. Lesens, wenn die L/H-Flanke $\overline{RD}$ beendet	$t_{SC(D)}$	35	-	ns	14
	Datensetzzeit bzgl. H/L des Taktes, wenn die H/L-Flanke $\overline{RD}$ beendet	$t_{SC(D)}$	50	-	ns	15
	stabile Daten vor $\overline{WR}$ (Speicher-Zyklus)	$t_{dcm}$	(8)	-	ns	16
	stabile Daten vor $\overline{WR}$ (I/O-Zyklus)	$t_{dci}$	100	-	ns	17
	stabile Daten nach $\overline{WR}$	$t_{cdf}$	(9)	-	ns	18
	Haltezeiten für spezifizierte Setzzeiten	$t_H$	0	-	ns	19

Signal	Parameter	Symbol	min	max	Einheit	Nr.
$\overline{\text{MREQ}}$	$\overline{\text{MREQ}}$ -Verzögerungszeit nach H/L d. Taktes $\overline{\text{MREQ}}$ Low	$t_{\text{DL}\overline{\text{C}}}(\text{MR})$		85	ns	21
	$\overline{\text{MREQ}}$ -Verzögerungszeit nach L/H d. Taktes $\overline{\text{MREQ}}$ High	$t_{\text{DHC}}(\text{MR})$		85	ns	22
	$\overline{\text{MREQ}}$ -Verzögerungszeit nach H/L d. Taktes $\overline{\text{MREQ}}$ High	$t_{\text{DH}\overline{\text{C}}}(\text{MR})$		85	ns	23
	Impulsbreite, $\overline{\text{MREQ}}$ Low	$t_{\text{w}}(\text{MR L})$	(10)	-	ns	24
	Impulsbreite, $\overline{\text{MREQ}}$ High	$t_{\text{w}}(\text{MR H})$	(11)	-	ns	25
$\overline{\text{IORQ}}$	$\overline{\text{IORQ}}$ -Verzögerungszeit nach L/H d. Taktes $\overline{\text{IORQ}}$ Low	$t_{\text{DLC}}(\text{IR})$	-	75	ns	27
	$\overline{\text{IORQ}}$ -Verzögerungszeit nach L/H d. Taktes $\overline{\text{IORQ}}$ High	$t_{\text{DHC}}(\text{IR})$	-	85	ns	28
	$\overline{\text{IORQ}}$ -Verzögerungszeit nach H/L d. Taktes $\overline{\text{IORQ}}$ High	$t_{\text{DH}\overline{\text{C}}}(\text{IR})$	-	85	ns	29
$\overline{\text{RD}}$	$\overline{\text{RD}}$ -Verzögerungszeit nach L/H d. Taktes	$t_{\text{DLC}}(\text{RD})$	-	85	ns	30
	$\overline{\text{RD}}$ -Verzögerungszeit nach H/L d. Taktes $\overline{\text{RD}}$ Low	$t_{\text{DL}\overline{\text{C}}}(\text{RD})$	-	95	ns	31
	$\overline{\text{RD}}$ -Verzögerungszeit nach L/H d. Taktes $\overline{\text{RD}}$ High	$t_{\text{DHC}}(\text{RD})$		85	ns	32
	$\overline{\text{RD}}$ -Verzögerungszeit nach H/L d. Taktes $\overline{\text{RD}}$ High	$t_{\text{DH}\overline{\text{C}}}(\text{RD})$		85	ns	33
$\overline{\text{WR}}$	$\overline{\text{WR}}$ -Verzögerungszeit nach L/H d. Taktes $\overline{\text{WR}}$ Low	$t_{\text{DLC}}(\text{WR})$		65	ns	34
	$\overline{\text{WR}}$ -Verzögerungszeit nach H/L d. Taktes $\overline{\text{WR}}$ Low	$t_{\text{DL}\overline{\text{C}}}(\text{WR})$		80	ns	35
	$\overline{\text{WR}}$ -Verzögerungszeit nach L/H d. Taktes $\overline{\text{WR}}$ High	$t_{\text{DHC}}(\text{WR})$		80	ns	36
	$\overline{\text{WR}}$ -Verzögerungszeit nach H/L d. Taktes $\overline{\text{WR}}$ High	$t_{\text{DH}\overline{\text{C}}}(\text{WR})$		80	ns	37
	Impulsbreite $\overline{\text{WR}}$ Low	$t_{\text{w}}(\text{WRL})$	(12)		ns	38
$\overline{\text{WAIT}}$	$\overline{\text{WAIT}}$ -Setzzeit zu H/L d. Taktes	$t_{\text{s}}(\text{WT})$	70		ns	39
$\overline{\text{BUSRQ}}$	$\overline{\text{BUSRQ}}$ -Verzögerungszeit nach L/H d. Taktes	$t_{\text{D}}(\text{BQ})$	-	100	ns	40
$\overline{\text{IORQ}}$ , $\overline{\text{RD}}$ , $\overline{\text{WR}}$ , $\overline{\text{MREQ}}$	Verzögerungszeit bis zum hochohmigen Zustand (Float) der Steuerausgänge ( $\overline{\text{MREQ}}$ , $\overline{\text{IORQ}}$ , $\overline{\text{RD}}$ , $\overline{\text{WR}}$ ) bezogen auf L/H d. Taktes	$t_{\text{F}}(\text{C})$	-	80	ns	41

$$(2) \quad t_{\text{c}} = t_{\text{w}}(\text{CH}) + t_{\text{w}}(\text{CL}) + t_{\text{r}} \quad t_{\text{f}}$$

$$(4) \quad t_{\text{acm}} = t_{\text{w}}(\text{CH}) + t_{\text{f}} - 65$$

$$(5) \quad t_{\text{aci}} = t_{\text{c}} - 70$$

$$(6) \quad t_{\text{ca}} = t_{\text{w}}(\text{CL}) + t_{\text{r}} - 50$$

$$(7) \quad t_{\text{caf}} = t_{\text{w}}(\text{CL}) + t_{\text{r}} - 45$$

$$(8) \quad t_{\text{dcm}} = t_{\text{c}} - 170$$

$$(9) \quad t_{\text{cdf}} = t_{\text{w}}(\text{CL}) + t_{\text{r}} - 70$$

$$(10) \quad t_{\text{w}}(\text{MRL}) = t_{\text{c}} - 30$$

$$(11) \quad t_{\text{w}}(\text{MRH}) = t_{\text{w}}(\text{CH}) + t_{\text{f}} - 20$$

$$(12) \quad t_{\text{w}}(\text{WRL}) = t_{\text{c}} - 30$$

Alle Gleichungen beziehen sich auf das DMA-(Standard)-Zeitverhalten.

- Bemerkungen:
- Daten können auf den DMA-Bus gegeben werden, wenn  $\overline{\text{RD}}$  aktiv ist.
  - Alle Steuersignale werden intern synchronisiert, sodaß sie bezüglich des Taktes völlig asynchron sein können.
  - Parameter mit unterstrichenen Nummern sind im Impulsdiagramm nicht dargestellt.

Kapazitäten ( $\vartheta_a = 25 \text{ }^\circ\text{C}$ ;  $f = 1 \text{ MHz}$ )

Parameter	Symbol	max	Einheit
Taktkapazität	$C_c$	35	pF
Eingangskapazität	$C_i$	7	pF
Ausgangskapazität	$C_o$	10	pF *

\* Ein-/Ausgänge wie Ausgänge bewertet

## 12.2. Betriebsbedingungen UB 858 D

Parameter	Symbol	min	max	Einheit
Betriebstemperatur	$\vartheta_a$	0	70	$^\circ\text{C}$
Betriebsspannung	$U_{CC}$	4,75	5,25	V
Eingangsspannung	$U_{IL}$	-0,5	0,8	V
Eingangsspannung	$U_{IH}$	2,0	$U_{CC}$	V
Takteingangsspannung	$U_{ILC}$	-0,5	0,45	V
Takteingangsspannung	$U_{IHC}$	$U_{CC}$	$U_{CC}$	V

### 12.2.1. Statische Kennwerte ( $U_{CC} = 5 \text{ V} \pm 5 \%$ ) ( $\vartheta_a = 0 \dots 70 \text{ }^\circ\text{C}$ )

Parameter	Symbol	min	max	Einheit	Meßbedingung
Ausgangsspannung, Low-Pegel	$U_{OL}$		0,4	V	$I_{OL} = 1,8 \text{ mA}$
Ausgangsspannung, High-Pegel	$U_{OH}$	2,4		V	$I_{OH} = -250 \text{ } \mu\text{A}$
Stromaufnahme	$I_{CC}$		200	mA	$\vartheta_a = 25 \text{ }^\circ\text{C}$
Eingangsreststrom	$I_{LI}$		$\pm 10$	$\mu\text{A}$	$U_i = 0 \text{ V} \dots U_{CCMAX}$
Reststrom der Dreizustandsausgänge im hochohmigen Zustand	$I_{LOH}$		20	$\mu\text{A}$	$U_0 = 5,25 \text{ V}$
Reststrom des Datenbusses bei Eingabe	$I_{LD}$		-20	$\mu\text{A}$	$U_0 = 0 \text{ V}$
			$\pm 20$	$\mu\text{A}$	$0 \text{ V} \leq U_i \leq U_{CCMAX}$

### 12.2.2. Dynamische Kennwerte

Passiver Zustand des UB 858 D ( $U_{CC} = 5 \text{ V} \pm 5 \%$ ) ( $\vartheta_a = 0 \dots 70 \text{ }^\circ\text{C}$ ;  $C_L = 100 \text{ pF}$ )

Signal	Parameter	Symbol	min	max	Einheit	Nr.
C	Taktperiode	$t_c$	400	4000 <sup>(2)</sup>	ns	1
	High-Breite d. Taktes	$t_w(\text{CH})$	170	2000	ns	2
	Low-Breite d. Taktes	$t_w(\text{CL})$	170	2000	ns	3
	Taktanstiegszeit	$t_r$	-	30	ns	4
	Taktabfallzeit	$t_f$	-	30	ns	5
	Haltezeit für spezifizierte Setzzeiten	$t_H$	0	-	ns	6/19
$\overline{\text{CE}}, \overline{\text{WR}}, \overline{\text{IORQ}}$	Setzzeit des Steuersignals bzgl. L/H des Taktes	$t_{SC}(\text{CE})$	280	-	ns	7
D 0 bis D 7	Datenausgangsverzögerung bzgl. H/L von RD	$t_{DR}(\text{D})$	-	500	ns	8
	Datensetzzeit bzgl. L/H des Taktes während Schreib- od. M 1-Zyklus	$t_{SC}(\text{D})$	50	-	ns	9
	Datenausgangsverzögerung bzgl. H/L von IORQ während INTA-Zyklus	$t_{DI}(\text{D})$	-	340	ns	10
	Floatverzögerung bzgl. L/H von RD	$t_{F}(\text{D})$	160	-	ns	11

Signal	Parameter	Symbol	min	max	Einheit	Nr.
IEI	IEI-Setzzeit bzgl. H/L von IORQ während INTA-Zyklus	$t_{S(IEI)}$	140	-	ns	12
IEO	IEO-Verzögerung bzgl. L/H von IEI	$t_{DH(IEO)}$	-	210	ns	13
	IEO-Verzögerung bzgl. H/L von IEI	$t_{DL(IEO)}$	-	190	ns	14
	IEO-Verzögerung bzgl. H/L von M 1	$t_{DM(IEO)}$	-	300	ns	15
M 1	M 1-Setzzeit bzgl. L/H d. Taktes	$t_{SC(M1)}$	210	-	ns	16
	M 1-Rücksetzzeit bzgl. H/L d. Taktes	$t_{RC(M1)}$	20	-	ns	17
$\overline{RD}$	$\overline{RD}$ -Setzzeit bzgl. L/H d. Taktes während M 1-Zyklus	$t_{SC(RD)}$	240	-	ns	18
$\overline{INT}$	$\overline{INT}$ -Verzögerungszeit bzgl. der Interruptursache ( $\overline{INT}$ wird nur erzeugt, wenn DMA inaktiv ist)	$t_{D(IT)}$	-	500	ns	19
BAO	BAO-Verzögerung bzgl. L/H von BAI	$t_{DH(BO)}$	-	200	ns	20
	BAO-Verzögerung bzgl. H/L von BAI	$t_{DL(BO)}$	-	200	ns	21
RDY	RDY-Setzzeit bzgl. L/H d. Taktes	$t_{SC(RDY)}$	150	-	ns	22

Aktiver Zustand des UA 858 D ( $U_{CC} = 5 V + 5 \%$ ;  $v_a = 0 \dots 70 \text{ }^\circ\text{C}$ ;  $C_L = 100 \text{ pF}$ )

Signal	Parameter	Symbol	min	max	Einheit	Nr.
C	Taktperiode	$t_C$	400	4000	ns	1
	High-Breite d. Taktes	$t_w(CH)$	180	2000	ns	2
	Low-Breite d. Taktes	$t_w(CL)$	180	2000	ns	3
	Taktanstiegszeit	$t_r$		30 <sup>(2)</sup>	ns	4
	Taktabfallzeit	$t_f$		30 <sup>(2)</sup>	ns	5
A 0 bis A15	Adressenausgangsverzögerung	$t_{D(AD)}$	-	145	ns	6
	Floatverzögerung	$t_{F(AD)}$	-	110	ns	7
	stabile Adresse vor MREQ (Speicherzyklus)	$t_{acm}$	(4)	-	ns	8
	stabile Adresse vor IORQ, RD oder WR (I/O-Zyklus)	$t_{aci}$	(5)	-	ns	9
	stabile Adresse nach RD oder WR	$t_{ca}$	(6)	-	ns	10
	stabile Adresse nach RD oder WR bei Übergang der Adreßbustreiber in den hochohmigen Zustand ("FLOAT")	$t_{caf}$	(7)	-	ns	11
D 0 bis D 7	Datenausgangsverzögerungszeit					
	Verzögerungszeit bis zum hochohmigen Zustand (Float) während Schreibzyklus	$t_{D(D)}$	-	230	ns	12
		$t_{F(D)}$	-	90	ns	13
	Datensetzzeit bzgl. L/H d. Taktes während Lesens, wenn die L/H-Flanke RD beendet	$t_{SC(D)}$	50	-	ns	14
	Datensetzzeit bzgl. H/L d. Taktes, wenn die H/L-Flanke RD beendet	$t_{SC\bar{D}(D)}$	60	-	ns	15
	stabile Daten vor $\overline{WR}$ (Speicherzyklus)	$t_{dcm}$	(8)		ns	16
	stabile Daten vor $\overline{WR}$ (I/O-Zyklus)	$t_{dci}$	100		ns	17
	stabile Daten nach $\overline{WR}$	$t_{cdf}$	(9)		ns	18
Haltezeiten für spezifizierte Setzzeiten	$t_H$	0		ns	19	

Signal	Parameter	Symbol	min	max	Einheit	Nr.
$\overline{\text{MREQ}}$	$\overline{\text{MREQ}}$ -Verzögerungszeit nach H/L d. Taktes $\overline{\text{MREQ}}$ Low	$t_{\text{DLC}}(\overline{\text{MR}})$		100	ns	21
	$\overline{\text{MREQ}}$ -Verzögerungszeit nach L/H d. Taktes $\overline{\text{MREQ}}$ High	$t_{\text{DHC}}(\overline{\text{MR}})$		100	ns	22
	$\overline{\text{MREQ}}$ -Verzögerungszeit nach H/L d. Taktes $\overline{\text{MREQ}}$ High	$t_{\text{DHC}}(\overline{\text{MR}})$		100	ns	23
	Impulsbreite, $\overline{\text{MREQ}}$ Low	$t_{\text{w}}(\overline{\text{MR L}})$	(10)	-	ns	24
	Impulsbreite, $\overline{\text{MREQ}}$ High	$t_{\text{w}}(\overline{\text{MR H}})$	(11)	-	ns	25
$\overline{\text{IORQ}}$	$\overline{\text{IORQ}}$ -Verzögerungszeit nach L/H d. Taktes $\overline{\text{IORQ}}$ Low	$t_{\text{DLC}}(\overline{\text{IR}})$	-	90	ns	27
	$\overline{\text{IORQ}}$ -Verzögerungszeit nach L/H d. Taktes $\overline{\text{IORQ}}$ High	$t_{\text{DHC}}(\overline{\text{IR}})$	-	100	ns	28
	$\overline{\text{IORQ}}$ -Verzögerungszeit nach H/L d. Taktes $\overline{\text{IORQ}}$ High	$t_{\text{DHC}}(\overline{\text{IR}})$	-	110	ns	29
$\overline{\text{RD}}$	$\overline{\text{RD}}$ -Verzögerungszeit nach L/H d. Taktes	$t_{\text{DLC}}(\overline{\text{RD}})$	-	100	ns	30
	$\overline{\text{RD}}$ -Verzögerungszeit nach H/L d. Taktes $\overline{\text{RD}}$ Low	$t_{\text{DLC}}(\overline{\text{RD}})$	-	130	ns	31
	$\overline{\text{RD}}$ -Verzögerungszeit nach L/H d. Taktes $\overline{\text{RD}}$ High	$t_{\text{DHC}}(\overline{\text{RD}})$		100	ns	32
	$\overline{\text{RD}}$ -Verzögerungszeit nach H/L d. Taktes $\overline{\text{RD}}$ High	$t_{\text{DHC}}(\overline{\text{RD}})$		110	ns	33
$\overline{\text{WR}}$	$\overline{\text{WR}}$ -Verzögerungszeit nach L/H d. Taktes $\overline{\text{WR}}$ Low	$t_{\text{DLC}}(\overline{\text{WR}})$		80	ns	34
	$\overline{\text{WR}}$ -Verzögerungszeit nach H/L d. Taktes $\overline{\text{WR}}$ Low	$t_{\text{DLC}}(\overline{\text{WR}})$		90	ns	35
	$\overline{\text{WR}}$ -Verzögerungszeit nach L/H d. Taktes $\overline{\text{WR}}$ High	$t_{\text{DHC}}(\overline{\text{WR}})$		100	ns	36
	$\overline{\text{WR}}$ -Verzögerungszeit nach H/L d. Taktes $\overline{\text{WR}}$ High	$t_{\text{DHC}}(\overline{\text{WR}})$		100	ns	37
	Impulsbreite $\overline{\text{WR}}$ Low	$t_{\text{w}}(\overline{\text{WRL}})$	(12)		ns	38
$\overline{\text{WAIT}}$	$\overline{\text{WAIT}}$ -Setzzeit zu H/L d. Taktes	$t_{\text{s}}(\overline{\text{WT}})$	70		ns	39
$\overline{\text{BUSRQ}}$	$\overline{\text{BUSRQ}}$ -Verzögerungszeit nach L/H d. Taktes	$t_{\text{D}}(\overline{\text{BQ}})$	-	150	ns	40
$\overline{\text{IORQ}}$ , $\overline{\text{RD}}$ , $\overline{\text{WR}}$ , $\overline{\text{MREQ}}$	Verzögerungszeit bis zum hochohmigen Zustand (Float) der Steuerausgänge ( $\overline{\text{MREQ}}$ , $\overline{\text{IORQ}}$ , $\overline{\text{RD}}$ , $\overline{\text{WR}}$ ) bezogen auf L/H d. Taktes	$t_{\text{F}}(\text{C})$	-	100	ns	41

$$(2) \quad t_{\text{c}} = t_{\text{w}}(\text{CH}) + t_{\text{w}}(\text{CL}) + t_{\text{r}} + t_{\text{f}}$$

$$(4) \quad t_{\text{acm}} = t_{\text{w}}(\text{CH}) + t_{\text{f}} - 65$$

$$(5) \quad t_{\text{aci}} = t_{\text{c}} - 80$$

$$(6) \quad t_{\text{ca}} = t_{\text{w}}(\text{CL}) + t_{\text{r}} - 40$$

$$(7) \quad t_{\text{caf}} = t_{\text{w}}(\text{CL}) + t_{\text{r}} - 60$$

$$(8) \quad t_{\text{dcm}} = t_{\text{c}} - 210$$

$$(9) \quad t_{\text{cdf}} = t_{\text{w}}(\text{CL}) + t_{\text{r}} - 80$$

$$(10) \quad t_{\text{w}}(\text{MRL}) = t_{\text{c}} - 40$$

$$(11) \quad t_{\text{w}}(\text{MRH}) = t_{\text{w}}(\text{CH}) + t_{\text{f}} - 30$$

$$(12) \quad t_{\text{w}}(\text{WRL}) = t_{\text{c}} - 40$$

Alle Gleichungen beziehen sich auf das DMA-(Standard)-Zeitverhalten

- Bemerkungen:
- Daten können auf den DMA-Bus gegeben werden, wenn RD aktiv ist.
  - Alle Steuersignale werden intern synchronisiert, so daß sie bezüglich des Taktes völlig asynchron sein können.
  - Parameter mit unterstrichenen Nummern sind im Impulsdiagramm nicht dargestellt.

Kapazitäten ( $v_a = 25 \text{ }^\circ\text{C}; f = 1 \text{ MHz}$ )

Parameter	Symbol	max	Einheit
Taktkapazität	$C_c$	35	pF
Eingangskapazität	$C_i$	7	pF
Ausgangskapazität	$C_o$	10	pF *

\* Ein-/Ausgänge wie Ausgänge bewertet

"1" "0"  
 Clock 4,2 V 0,8 V  
 Output 2,0 V 0,8 V  
 Input 2,0 V 0,8 V

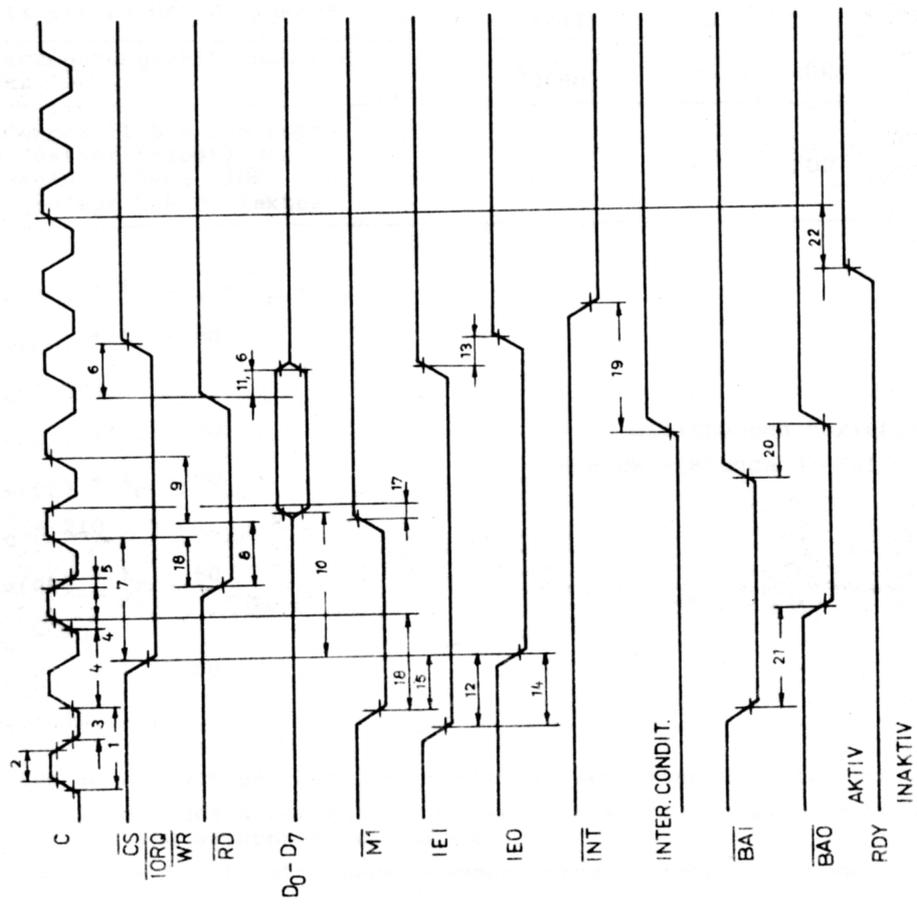


Bild 39 Impulsdiagramm (passiver Zustand)

"1" "0"  
 Clock 4,2 V 0,8 V  
 Output 2,0 V 0,8 V  
 Input 2,0 V 0,8 V  
 FLOAT  
 U = +0,5 V

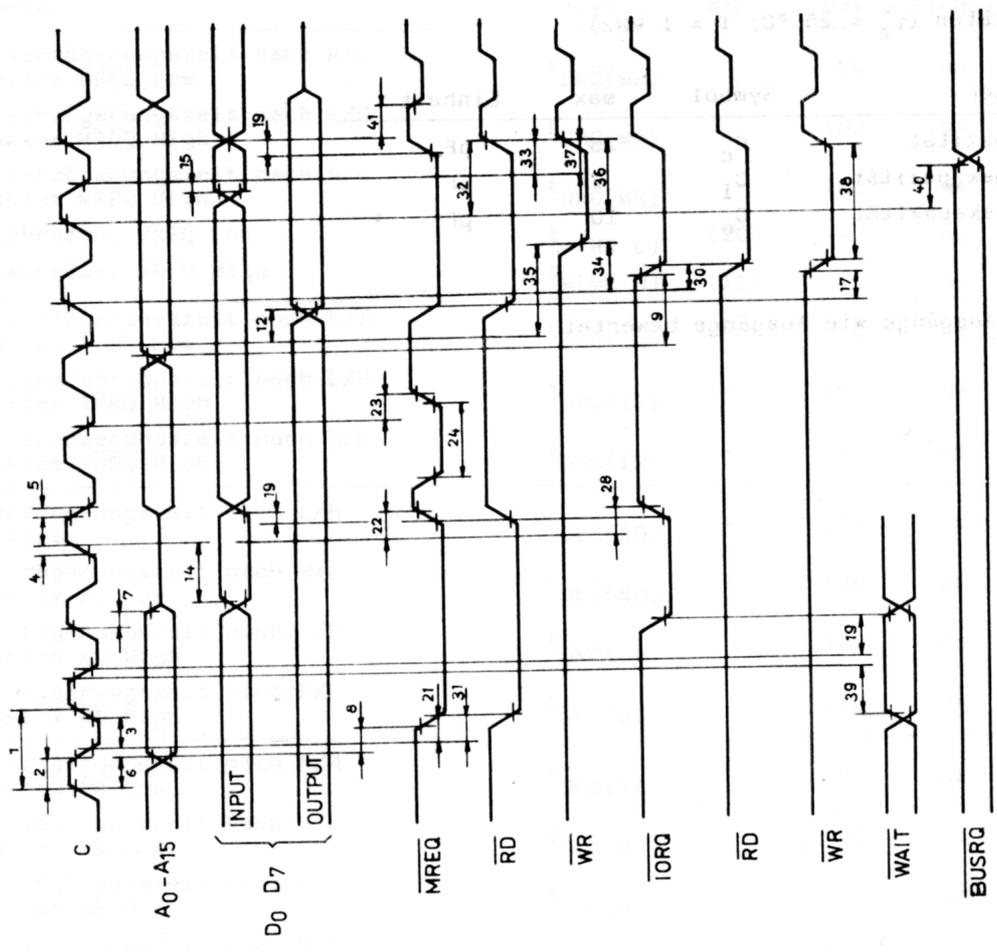


Bild 40 Impulsdiagramm (aktiver Zustand)

### 13. Gehäuse

40polig, DIL-Plast

Bauform 21.2.3.2.40 nach TGL 26 713

Masse ca. 5,4 g

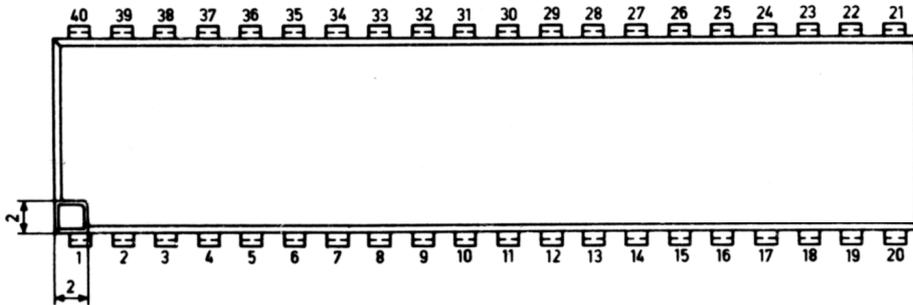
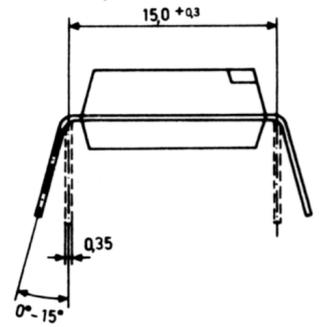
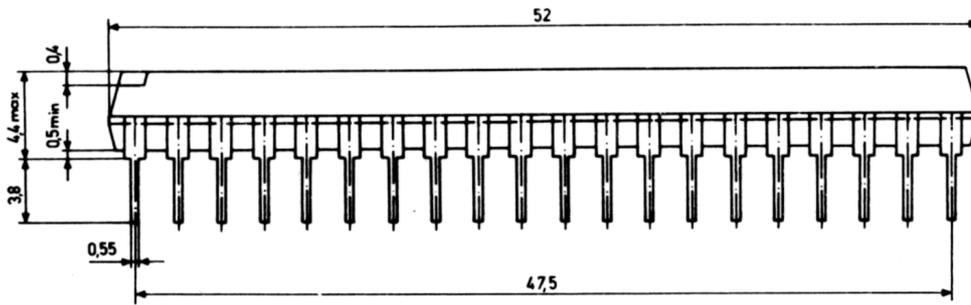


Bild 41



**elektronik**  
**export·import**

VOLKSEIGENER AUSSENHANDELSBETRIEB DER  
DEUTSCHEN DEMOKRATISCHEN REPUBLIK  
DDR 1026 BERLIN ALEXANDERPLATZ  
HAUS DER ELEKTROINDUSTRIE

**v eb mikroelektronik · karl marx · erfurt**  
**stammbetrieb**

DDR-5010 Erfurt, Rudolfstraße 47 Telefon: 5 80, Telex: 061 306