

robotron

Geräte der Dezentralen Datentechnik

Bausteinübersicht

Heft 2

5. Auflage
Karl-Marx-Stadt 1988

Inhaltsverzeichnis

robotron Stand	ESER	DDR - Typ				SU-Typ	RGW-Typ	NSW-Typ	Seite
		Standard	DL	DS	C-Mos				
Q282					U2164020	K565RU5W		i 2164	2
LX23			DL 123					SN 74LS123	3
PS75		D 175				K155 TM 7		SN 7475	4-5
LY53			DL 253			K555KP12		SN 74LS253	5-6
LY57			DL 257			K555KP11		SN 74LS257	6
PS89						K155RU2		SN 7489N	7
LA27			DL 8127					AmZ8127	8-10
AP70		B 3170H						NSCLM317K	11
AP71		B 3171H						NSCLM317HVK	11
AN70		B 3370H						NSCLM337K	11
AN71		B 3371H						NSCLM337HVK	11
Q365					UB8001C			Z8001	12-37
Q284						KR565RU6W			37-39
		B 315 D							40
		MB 111							40
		UB 8010C							41-49
PX81		B 461G						SAS 261G4	50
LS90		DL 090						SN 74LS90	51-53
AT56		B 556						LM 556CN	53
CD93					U 4093	K 561 TL1		HEF 4093	54
Formalzeichen und Abkürzungen									54-55

Achtung!

In den Erzeugnissen der Baureihe "Bürocomputer" und "Schalterterminals" werden Bausteine der D-Baureihe durch Bausteine der DL-Baureihe abgelöst.
Werden im Reparaturfall Bausteine der D-Baureihe durch die DL-Baureihe ersetzt, muß unbedingt auf den Lastfaktor geachtet werden.

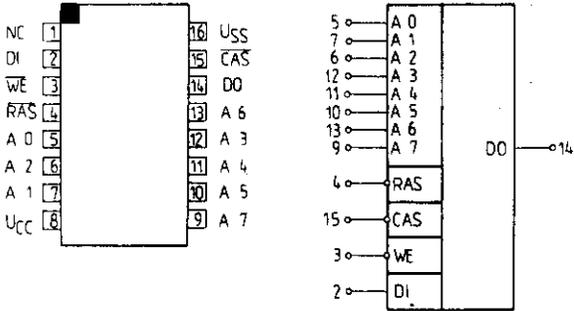
Merke: Ein DL-Gatter-Ausgang schaltet maximal fünf D-Gatter!

Buchstaben-Bezeichnung der DDR-Typen

A^x, B^{xx}, C^{xxx} = Analoge Schaltkreise
D^x, E^{xx}, F^{xxx}, DS, DL = Digitale bipolare TTL-Schaltkreise
V/U^x, W^{xx}, Z^{xxx} = Digitale unipolare Schaltkreise

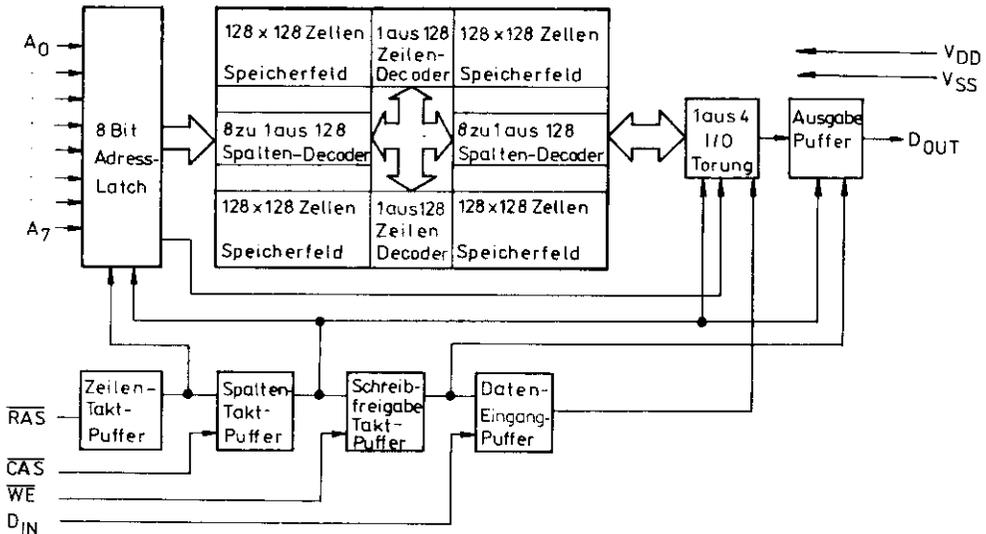
x = Temperaturbereich von 0 °C bis 70 °C, bei V/U - 25 ° bis 70 °
xx = Temperaturbereich von - 25 °C bis 85 °C
xxx = Temperaturbereich von - 55 °C bis 125 °C
DS = Schottky
DL = Low-Power-Schottky

65 536 x 1 Bit dynamischer RAM



- 1 nicht belegt
- 2 Dateneingang
- 3 Schreibsignal
- 4 Eingang des Signals zur Zeilenauswahl
- 5-7, 9-13 Adresseneingänge
- 8 Betriebespannung + 5 V
- 14 Datenausgang
- 15 Eingang des Signals zur Spaltenauswahl
- 16 Bezugspotential

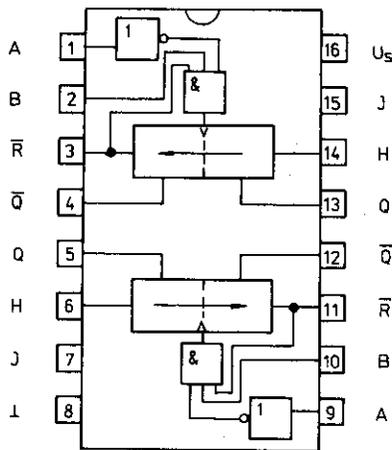
Block Diagramm



Nachtriggerbare monostabile Kippstufe mit Rückstelleingang, zweifach

Die A-Eingänge triggern die Kippstufe bei Übergang von H auf L, während die anderen Eingänge auf H liegen.

Die B-Eingänge triggern bei Übergang von L auf H, während A auf L und \bar{R} auf H liegt. Bei L am Rückstelleingang \bar{R} sind die Eingänge gesperrt und die Ausgänge Q (\bar{Q}) gehen auf L (H) zurück. Die Bausteine können in getriggertem Zustand erneut getriggert werden, so daß die Dauer des Ausgangsimpulses vom letzten Triggerimpuls bestimmt wird.



Logisches Verhalten:

Eingänge Ausgänge

A	B	Q	\bar{Q}
H*	X	L	H
X*	L	L	H
L	\lrcorner	\lrcorner	\lrcorner
\lrcorner	H	\lrcorner	\lrcorner

X = H-oder L-Signal

\lrcorner = H-Impuls

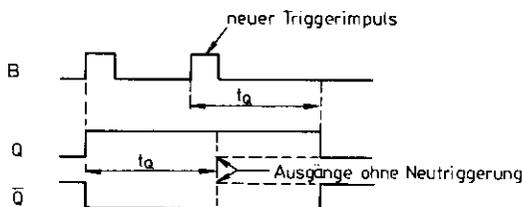
\lrcorner = L-Impuls

\lrcorner = Impuls-Wechsel von L auf H

\lrcorner = Impuls-Wechsel von H auf L

* = Eingangsbedingungen für Rückstellung

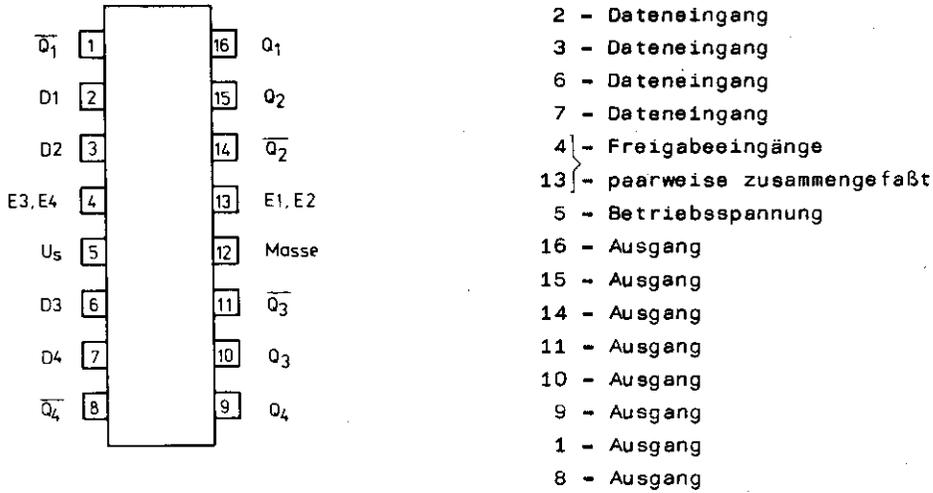
Impulsdiagramm:



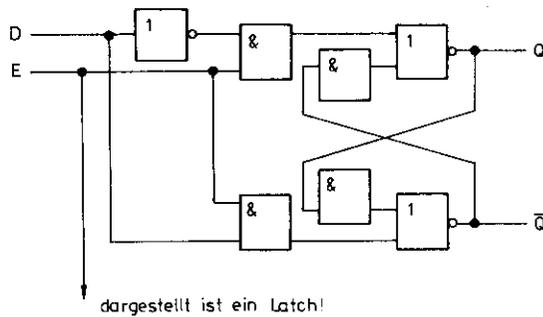
Statische Kenndaten

		Min.	typ.	Max.	Einheit
Speisespannung	U_S	4,75	5,0	5,25	V
H-Eingangsspannung	U_{IH}	2			V
L-Eingangsspannung	U_{IL}				
} bei $U_S = 4,75$ V					
H-Ausgangsspannung	U_{QH}	2,7			V
L-Ausgangsspannung	U_{QL}			0,5	V
bei $U_S = 4,75$ V, $I_Q = 8$ mA					
H-Eingangsstrom pro Eingang	I_{IH}			100	μ A
bei $U_I = 7,0$ V					
L-Eingangsstrom an A oder B	$-I_{IL}$			360	μ A
bei $U_{IL} = 0,4$ V					
L-Eingangsstrom an \bar{R}	$-I_{IL}$			100	mA
bei $U_{IL} = 0,4$ V					
Kurzschlußausgangsstrom pro Ausgang	$-I_Q$	20		100	mA
bei $U_S = 5,25$ V					
Speisestrom	I_S			20	mA
bei $U_S = 5,25$ V					

Vierfach Speicher-Flip-Flop



Blockschaltbild:



Logische Funktion

t_n			t_{n+1}		
D	E	Q	D	E	Q
H	H	H	L	H	L
L	H	L	H	H	H
H	H	H	H	L	H
H	L	H	L	L	H
L	L	H	L	H	L
L	H	L	L	L	L
L	L	L	H	L	L
H	L	L	H	H	H

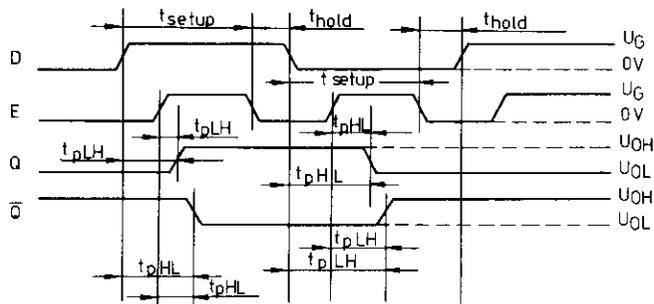
t_n = Zeit vor dem Eingangsimpuls

t_{n+1} = Zeit nach dem Eingangsimpuls

Betriebsbedingungen:

		min.	max.	
Betriebsspannung	U_G	4,75	5,25	V
H-Eingangsspannung	U_{IH}	2		V
L-Eingangsspannung	U_{IL}		0,8	V
Freigabeimpulsdauer	t_{pE}	20		ns
Voreinstellzeit	t_{setup}	20		ns
Haltezeit	t_{hold}	0		ns
Betriebstemperaturbereich	ϑ_a	0	70	$^{\circ}C$

Impulsdiagramm:

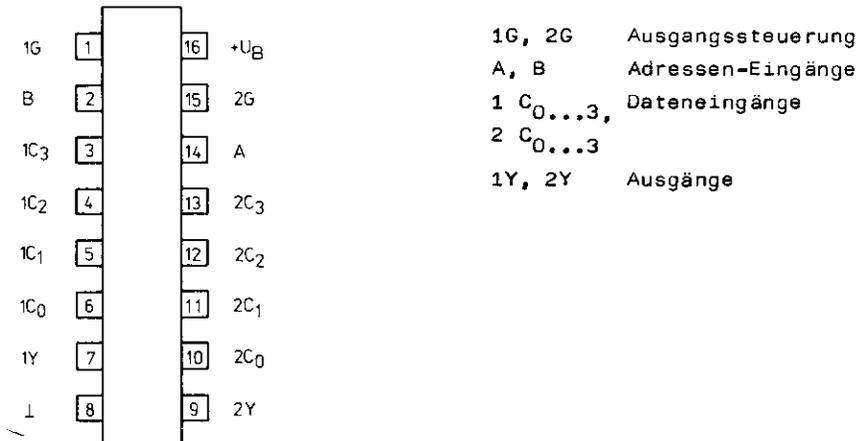


	min.	max.
$t_{pLH} (D-Q)$	16	30 ns
$t_{pHL} (D-Q)$	14	25 ns
$t_{pLH} (D-\bar{Q})$	24	40 ns
$t_{pHL} (D-\bar{Q})$	7	15 ns
$t_{pLH} (E-Q)$	16	30 ns
$t_{pHL} (E-Q)$	7	15 ns
$t_{pLH} (E-\bar{Q})$	16	30 ns
$t_{pHL} (E-\bar{Q})$	7	15 ns

DL 253

SN74LS253

4-auf-1-Multiplexer, zweifach



Funktionstabelle

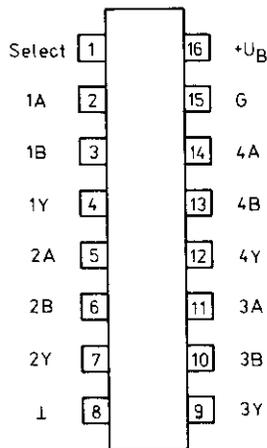
Ausgangs- Steuerung	Adressen- Eingänge		Ausgang
	B	A	
H	*	*	hochohm.
L	L	L	C ₀
L	L	H	C ₁
L	H	L	C ₂
L	H	H	C ₃

* = beliebig L oder H

DL 257

SN 74LS257

2-auf-1-Multiplexer, vierfach



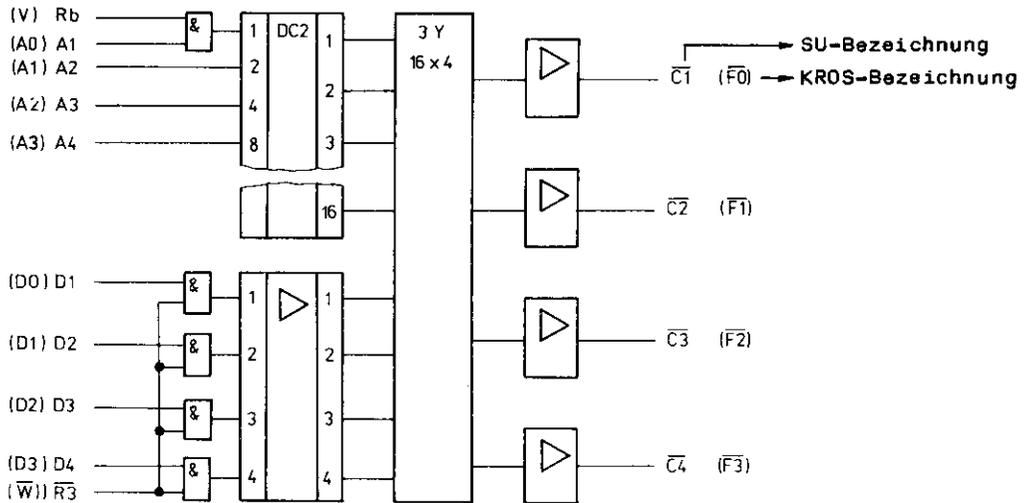
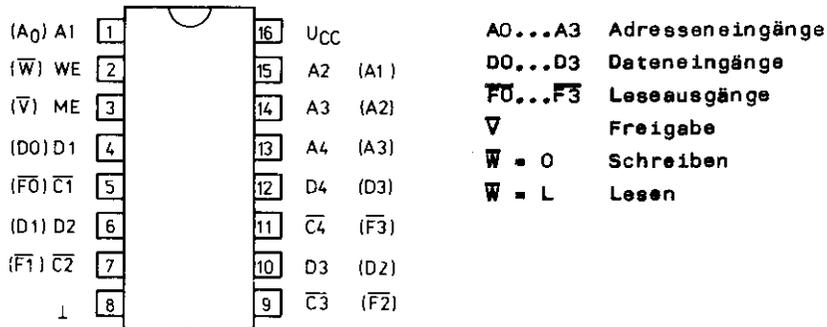
G Ausgangsteuerung
 Select Select Adresseneingang
 Y Ausgang

Funktionstabelle

Ausgangs- steuerung	Select Adressen- eingang	Ausgang
H	*	hochohm.
L	L	A
L	H	B

* = beliebig L oder H

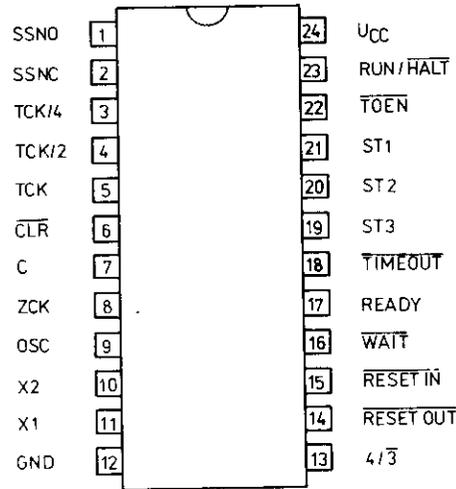
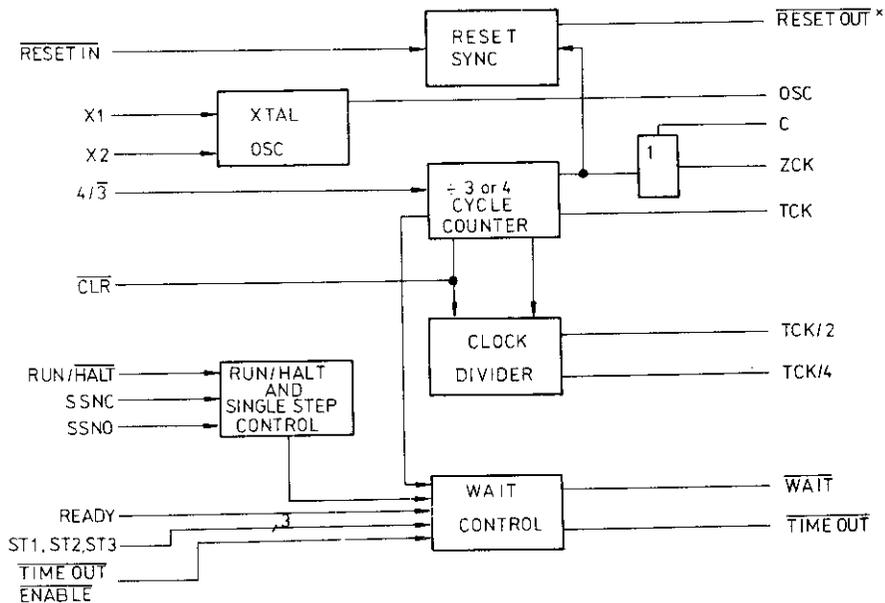
64 Bit SRAM (16X4)



Elektrische Daten

Betriebsspannung	5 + 5%	V
Ausgangstrom Lesezykluszeit	20	µA
Ausgangstrom Speicherzykluszeit	20	µA
Logischer "0" Ausgangspegel für nicht ausgesprochenen Speicherplatz mit eingespeicherter "1"	0,4	V
Logischer "0" Ausgangspegel während der Lesezeit	0,4	V
Abfrageverstärkererholzeit nach Schreibzyklus für "0" oder "1"	70	nsec
Abfrageverstärker-Einschalt/Ausschaltverzögerungszeit vom Freigabeeingang	50	nsec
Abfrageverstärker-Einschalt/Ausschaltverzögerungszeit vom Adresseneingang zum -ausgang	60	nsec

System-Taktgenerator

Blockschaltbild

* RESET OUT ist aktiv low, wenn 4/3 = high

Betriebsbedingungen

		min.	typ.	max.	
Betriebespannung	U_{CC}	4,75	5,0	5,25	V
H-Eingangsspannung	U_{IH}				
ST1, ST2, ST3, X1, \overline{CLS} , TOEN, READY		2,0			V
RUN/ \overline{HALT} , SSNO, SSNC, $4/\overline{3}$		2,4			V
$\overline{RESET\ IN}$		2,8			V
L-Eingangsspannung	U_{IL}				
ST1, ST2, ST3, X1, \overline{CLR} , TOEN, READY				0,8	V
RUN/ \overline{HALT} , SSNO, SSNC, $4/\overline{3}$				0,4	V
H-Ausgangstrom	I_{OH}				
ZCK				0,1	mA
TTL-Ausgänge				2,6	mA
L-Ausgangstrom	I_{OL}				
ZCK				0,1	mA
TTL-Ausgänge				16,0	mA

Max. Quarzfrequenz = 24 MHz

Beschreibung der Anschlüsse

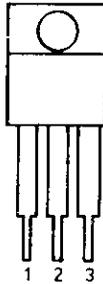
$4/\overline{3}$	Eingang zur Steuerung der Betriebsart. H ergibt ein Taktastverhältnis von 1 : 2. L ergibt ein Taktastverhältnis von 1 : 3, sowie die Negation des CPU-Taktes ZCK gegenüber dem TTL-Takt TCK.
$\overline{RESET\ IN}$	L-aktiver Eingang zur Synchronisation der RESET-Signale mit dem CPU-Takt.
$\overline{RESET\ OUT}$	Synchronisierter RESET-Ausgang. L-aktiv, wenn $4/\overline{3} = H$, sonst H-aktiv.
X1, X2	Eingänge für externen Quarzanschluß.
ZCK	Gepuffertes Taktausgang für CPU und Peripherie. Er liefert die erforderliche H-Ausgangsspannung ($U_{CC} = - 0,4 V$).
TCK	Gepuffertes TTL-Taktausgang mit der gleichen Frequenz wie ZCK, TCK ist mit ZCK synchronisiert. Wenn $4/\overline{3} = L$, ist ZCK zu TCK negiert.
TCK/2, TCK/4	Gepufferte TTL-Taktausgänge mit der halben bzw. viertel TCK-Frequenz. Synchronisiert mit der Anstiegsflanke von TCK.
OSC	Oszillatortaktausgang, TTL-gepuffert. Liefert high-speed-Takt für dynamische Speichersteuerung oder andere Anwendungen. Die ZCK- und TCK-Ausgänge sind mit der OSC-Anstiegsflanke synchronisiert.
\overline{CLR}	L-aktiver Eingang mit der OSC-Anstiegsflanke synchronisiert. Setzt die internen Taktteiler zurück, um die Synchronisation der Taktausgänge mehrerer Schaltkreise zu gewährleisten.

WAIT	Verriegelter (Latched) WAIT -Ausgang zur Verbindung mit der CPU. Wird von den READY- , ST1- , ST2- , ST3- , RUN/HALT -und SINGLE-STEP -Eingängen zur Forderung von Wartezuständen gesteuert.
READY	H-aktiver Eingang zur Verbindung mit peripheren Geräten. Einhaltung der dem WAIT -Latch entsprechenden Einstell- und Haltezeitbedingungen erforderlich.
ST1, ST2, ST3	Statusseingänge von CPU und peripheren Geräten. L an allen Eingängen kennzeichnet interne CPU-Arbeit oder REFRESH -Zyklen. Während dieser Zeit ist TIMEOUT abgeschaltet, um ein unpassendes Interrupt zu vermeiden. Die Statusseingänge sind von den Einstell- und Haltezeitbedingungen des WAIT -Latches abhängig.
RUN/HALT	Eingang, der Start/Stop- und Einzelschrittsteuerung ermöglicht. L erzwingt ein L am WAIT -Ausgang.
SSNO, SSNC	SINGLE-STEP -Steuereingänge, kurzzeitiges Abschalten von SSNC von Masse und Erden von SSNO bewirkt das Übergehen der CPU von einem Wartezustand in den anderen. RUN/HALT muß für Einzelschrittsteuerung auf L liegen.
TIMEOUT	Ausgang, hauptsächlich zur Verbindung mit einem Interrupteingang der CPU (NMI). Der TIMEOUT -Zähler zählt die ZCK/TCK -Taktzyklen, um die unbeantwortete WAIT -Forderung eines peripheren Gerätes auf 15 Taktzyklen zu begrenzen. Das heißt, nach 15 Taktzyklen wird ein H am WAIT -Ausgang erzwungen.
TOEN	L-aktiver timeout-enable-Eingang. L gibt den TIMEOUT -Zähler frei. H schaltet ihn ab und ermöglicht die WAIT -Steuerung durch die READY- , RUN/HALT -und SINGLE-STEP -Eingänge.
C	Bootstrap-Eingang, Kondensator C_R wird mit C und ZCK verbunden, um eine kürzere ZCK -Anstiegszeit zu ermöglichen.

B 3170 und B 3370
 B 3171 und B 3371

B 3170 H - B 3370 H Positiv-Spannungsregler
 B 3171 H - B 3371 H Negativ-Spannungsregler

Integrierte einstellbare Spannungsregler, einsetzbar für Ströme bis 1,5 A und Ausgangsspannungen von 1,2 V bis 37 V (B 3170 H), 1,2 V bis 57 V (B 3171 H), - 1,2 V bis 47 V (B 3371 H). Die Spannungsregler sind gegen Übertemperatur und Kurzschluß geschützt.



Anschlußbelegungen

B 3170/3171 H	B 3370/3371 H
1 Abgleich	1 Einstellenschluß
2 Ausgang	2 Eingang
3 Eingang	3 Ausgang

Grenzwerte, gültig für den Betriebstemperaturbereich

		B 3170 H	B 3370 H	
		B 3171 H	B 3371 H	
max. Eingangs-/Ausgangsspannungsdifferenz	U_D	40	60	V
	$-U_D$	40	50	V

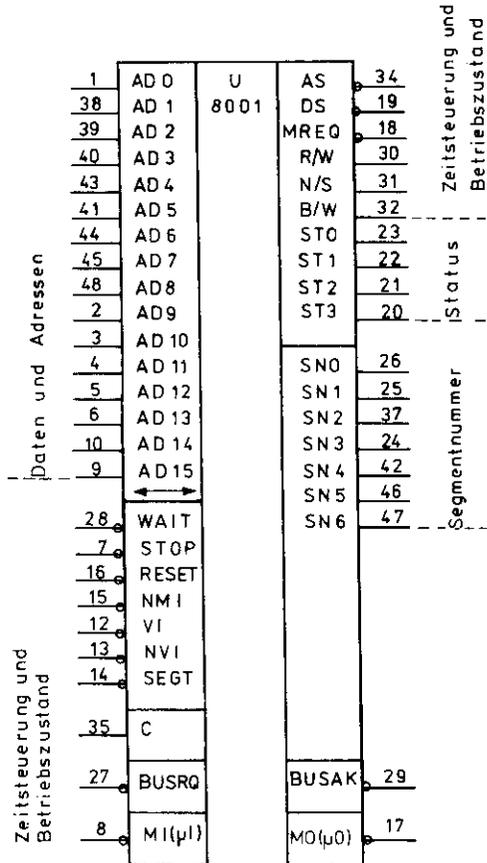
Elektrische Kenndaten

min. Ausgangsspannung	U_{Omin}	1,2 ... 1,3 V
$U_I = 4,25 \dots 41,3 \text{ V}$ (B 3170 H)		
4,25 ... 51,3 V (B 3370 H)		
min. Eingangs-/Ausgangsspannungsdifferenz	U_{Dmin}	3 V
$I_O = 10 \text{ mA} \dots 1,5 \text{ A}$		
min. Ausgangsstrom	I_{Omin}	10 mA
max. Ausgangsstrom	I_{Omax}	1,5 A
max. Eingangsspannungsausregelung	ΔU_{Oumax}	0,4 % 0,6 % ¹⁾²⁾
max. Lastregelung	ΔU_{OI}	30 mV
$U_I = 4,25 \text{ V}$		
$I_{O1} = 10 \text{ mA}$		
$I_{O2} = 1,5 \text{ A}$		
Ausgangskurzschlußstrom	I_{OK}	1,6 ... 3 A
max. Strom aus dem Einstellenschluß	$I_{Einst.-max}$	10 μA

1) B 3171 H 2) B 3371 H

U 8001 Mikroprozessor

1	AD0	AD8	48
2	AD9	SN6	47
3	AD10	SN5	46
4	AD11	AD7	45
5	AD12	AD6	44
6	AD13	AD4	43
7	STOP	SN4	42
8	MI(μI)	AD5	41
9	AD15	AD3	40
10	AD14	AD2	39
11	Vcc	AD1	38
12	VI	SN2	37
13	NVI	GND	36
14	SEGT	CLOCK	35
15	NMI	AS	34
16	RESET	DCL	33
17	M0(μ0)	B/W	32
18	MREQ	N/S	31
19	DS	R/W	30
20	ST3	BUSAK	29
21	ST2	WAIT	28
22	ST1	BUSRQ	27
23	ST0	SN0	26
24	SN3	SN1	25



Funktionen der Bausteinanschlüsse (Taktzeit Δ 250 ns!)

AD0 ... AD15 (Adressen, Daten):

Diese 16 Anschlüsse übertragen den primären Informationsaustausch zwischen der Peripherie (Speicher oder E/A-Port) und der CPU. Jede Leitung hat 3 Funktionen: Adreßbit ausgeben, Datenbit übernehmen (lesen) oder ausgeben (schreiben).

Der Zustand der Zeitsteuersignale \overline{AS} und \overline{DS} gibt eindeutig an, wann an den AD-Anschlüssen ein Adreß- oder Datenwort anliegt.

\overline{AS} Adreß-Strobe (Ausgang):

Dieser Anschluß ist immer im ersten Taktzyklus eines CPU-Maschinenzyklus etwa 120 ns aktiv. Er zeigt damit das Anliegen eines Adreßwortes am AD-Bus an. Die Adresse ist bei \overline{AS} gleich L in der Phase der Ausgabe und im weiteren Verlauf vorhanden. Sie wird jedoch bei der L-H-Flanke erst als absolut gültig definiert.

\overline{DS} Daten-Strobe (Ausgang):

Dieser Anschluß wird im zweiten Taktzyklus eines CPU-Maschinenzyklus aktiv; frühestens nach 370 ns, gemessen von der L-H-Taktflanke. \overline{DS} gleich L zeigt an, daß von diesem Zeitpunkt an Daten von der CPU akzeptiert (lesen) oder von ihr ausgesandt werden (schreiben). Für \overline{DS} aktiv entscheiden unterschiedliche Operationen: Speicher lesen (370 ns), Speicher schreiben (470 ns) und E/A-Port-Lesen/Schreiben (500 ns).

\overline{DS} bleibt länger aktiv, wenn über den \overline{WAIT} -Eingang der CPU Wartezyklen eingeschoben werden, d. h. der Datentransfer wird künstlich verlängert. Über den R/\overline{W} -Ausgang wird die Richtung definiert.

\overline{MREQ} Memory Request (Ausgang):

Dieser Ausgang wird im L-Abschnitt des ersten Taktzyklus eines CPU-Mz wirksam. Er definiert genau, um welche Art eines Datentransfers es sich bei dem momentanen Maschinenzyklus handelt. L zeigt einen Speichertransfer an. H zeigt einen Transfer mit einem E/A-Port an.

R/\overline{W} Lesen/Schreiben (Ausgang):

Der Zustand dieses Ausgangs definiert die Richtung eines Datentransfers. H zeigt an, daß das Datenbyte/-wort von der Peripherie in die CPU übertragen wird (lesen). L zeigt an, daß die CPU an die Peripherie ein Datenbyte / -wort ausgibt (schreiben). Der Ausgang wird im H-Abschnitt des ersten Taktzyklus eines CPU-Mz wirksam.

N/\overline{S} Normal/System (Ausgang):

Die CPU besitzt zwei Betriebsarten, die sich in der Ausführbarkeit der Befehle unterscheiden. Die 110 Befehlstypen der CPU enthalten einige bevorzugte Befehle, die nur dann ausgeführt werden, wenn die Betriebsart "System" gewählt ist (im FCW Bit 14 setzen). H zeigt die Normal-Betriebsart an, die die Ausführung privilegierter Befehle nicht gestattet. L zeigt die System-Betriebsart an, die die Ausführung aller Befehle gestattet.

B/\overline{W} Byte/Wort (Ausgang):

Dieser Ausgang wird im H-Abschnitt des ersten Taktzyklus eines CPU-Mz wirksam. Die CPU führt Byte- und Wortbefehle (plus Langwortbefehle) aus. Dieser Ausgang zeigt den Unterschied eines Byte-Mz (H) oder eines Wort-Mz (L) an. Speziell bei Speicherdaten-Transfer wird zur hardwaremäßigen Unterscheidung dieser Zyklen die Adreßleitung A_0 und B/\overline{W} benutzt. Ist A_0 gleich L (= gerade Adresse) wird für B/\overline{W} gleich H das höherwertige Datenbyte (D8 bis D15) beachtet.

ST0 ... ST3 Status (Ausgänge):

Die 4 Anschlüsse sind binär kodiert und ergeben 16 diskrete Zustände. 15 davon definieren einen momentanen Arbeitegang (Mz) der CPU. Sie zeigen genau an, welchen Operationstyp die CPU gerade ausführt. Die ST-Ausgänge sind im H-Abschnitt des ersten Taktzyklus eines CPU-Mz wirksam.

Tabelle der Betriebszustände

ST3 ... ST0	Nr.	Bedeutung
0 0 0 0	0	interne Operation
0 0 0 1	1	Auffrischen des dynamischen RAM-Speicherbereichs
0 0 1 0	2	E/A-Operation mit allgemeiner Anwenderschnittstelle
0 0 1 1	3	E/A-Operation mit der MMU (Befehle SINDR, SINI, SOTIR, SOUT)
0 1 0 0	4	CPU akzeptiert einen Segment-Trap
0 1 0 1	5	CPU akzeptiert einen NMI (nicht maskierter Interrupt)
0 1 1 0	6	CPU akzeptiert einen NVI (nicht vektorisierter Interrupt)
0 1 1 1	7	CPU akzeptiert einen VI (Vektorinterrupt)
1 0 0 0	8	Anforderungszyklus für den Datenspeicher
1 0 0 1	9	Anforderungszyklus für den Stackspeicher
1 0 1 0	10	EPU-Anforderung für den Datenspeicher
1 0 1 1	11	EPU-Anforderung für den Stackspeicher
1 1 0 0	12	Befehlszyklus für n-tes Wort im Code-Speicher
1 1 0 1	13	Befehlsholzyklus für erstes Wort im Code-Speicher
1 1 1 0	14	Transfer-Operation von der EPU zur CPU
1 1 1 1	15	ohne definierten Betriebszustand

WAIT CPU-Steuerung (Eingang):

Über diesen Eingang kann man die CPU dazu veranlassen, einen Datentransfer zeitlich auf die Peripherie abzustimmen. Die Peripherie müßte bei einem Transfer innerhalb von 350 ns reagieren können. Diese Zeit ist für den überwiegenden Teil der Speicherbausteine und E/A-Ports zu schnell. Liegt während einer H-L-Flanke im zweiten Taktzyklus eines CPU-Mz ein L am WAIT-Eingang, so schiebt die CPU einen Wartezyklus (Wz) ein. Ein Wz entspricht 250 ns. Der AD-Bus und alle wichtigen Status- und Zeitsignale der CPU bleiben in dieser Zeit unverändert.

Bei einem E/A-Transfer wird zu einem automatisch eingeschobenen Wz ein weiterer hinzugefügt, wenn bei der H-L-Flanke des Wz-Taktes am WAIT-Eingang L anliegt. Wz können beliebig eingeschoben werden.

STOP CPU-Steuerung (Eingang):

Der STOP-Eingang ermöglicht eine einfache Kontrolle über die Ausführung eines Programmes. Man kann dabei immer nur einen Mz von der CPU ausführen lassen. Ist der STOP-Eingang aktiviert, wird die CPU angehalten. Sie gibt nur noch Speicherauffrischadressen aus. Der STOP-Eingang wird im letzten Takt eines CPU-Mz abgefragt. Ist STOP gleich L, wird nur noch der folgende Befehl geholt.

BUSRQ Bus-Anforderung (Eingang):

Wird der CPU-Bus von einem DMA-Baustein benötigt um einen Datentransfer im direkten Speicherzugriffs-Modus vorzunehmen, muß die Anmeldung für die Bussteuerung über den BUSRQ-Eingang eingeleitet werden. Wird L an diesen Anschluß gelegt, wird in der CPU ein Flip-Flop gesetzt. Nach dem dritten Takt eines beliebigen Mz werden der AD-Bus, die Segment-adreß-Ausgänge, die Zeitsteuer- und Betriebszustands-Ausgänge in den hochohmigen Zustand versetzt. Gleichzeitig antwortet die CPU mit L am BUSAK-Ausgang. Je nach Dauer der DMA-Operation muß BUSRQ auf L gehalten werden und dieser wird anschließend vom DMA-Baustein selbst wieder inaktiviert. Nach zwei Taktzyklen übernimmt die CPU wieder die Buspriorität.

BUSAK Busabgabe-Anerkennung (Ausgang):

Ist der BUSRQ-Eingang der CPU aktiviert worden, wird die Anforderung auf die Buspriorität eines DMA-Bausteins von der CPU nach etwa drei Taktzyklen anerkannt. Die Busabgabe-Anerkennung der CPU erfolgt durch L am Anschluß BUSAK.

NMI Nicht maskierter Interrupt (Eingang):

Dieser Programmunterbrechungs-Eingang besitzt die höchste Priorität aller Interruptebenen des U8001. Er folgt in der generellen Priorität nach einem internen Trap. Er wird flankengesteuert (H-L-Flanke). Ein nur für diese Ebene vorhandenes Flip-Flop aktiviert den NMI zu jeder Zeit des Mz.

VI Vektorierter Interrupt (Eingang):

Die nächstniedrige Priorität der Programmunterbrechung hat der VI. Er folgt in der generellen Priorität nach einem Segment-Trap der MMU. Der VI muß so lange angelegt sein (L), bis eine Anerkennung der CPU durch ein entsprechendes Statussignal erfolgt. Ein VI wird niemals anerkannt, wenn das Bit 12 im FCW, genannt VIE, nicht gesetzt ist.

NVI Nicht vektorierter Interrupt (Eingang):

Die absolut niedrigste Priorität einer Unterbrechung ist der NVI. Er muß so lange angelegt sein (L), bis eine Anerkennung der CPU durch ein entsprechendes Statussignal erfolgt. Ein NVI wird niemals anerkannt, wenn das Bit 11 im FCW, genannt NVIE, nicht gesetzt ist.

µI Multi-Mikro (Eingang):

Über diesen Eingang wird eine Anforderung auf die Buspriorität in die CPU eingegeben. Dieser Eingang ist Teil der Multi-Mikro Daisy-Chain-Buskonfiguration eines Mehrprozessorsystems.

µO Multi-Mikro (Ausgang):

Wie oben, nur wird eine Anforderung auf die Buspriorität von der CPU ausgegeben.

RESET (Eingang):

Wird L an den RESET-Eingang gelegt, werden zuerst alle Funktionsleitungen der CPU in einen inaktiven Zustand gebracht. Nach drei Taktzyklen beginnen drei Halbzyklen für das Laden des FCW, Programmzähler-Segment und -offset.

CLK CPU-Takt (Eingang):

Die U8001 ermöglicht den Betrieb mit einem Einphasentakt in einem Bereich von 500 kHz bis 4 MHz (entsprechend 2 μ s ... 250 ns Zykluszeit). Der H-L-Abschnitt soll mäanderförmig sein (105 ns sollen nicht unterschritten werden). Die Flanken sollen in 20 ns oder schneller fallen oder steigen.

SNO ... SN6 Segment-Adresse (Ausgänge):

Über diese Anschlüsse wird die Segment-Adresse ausgegeben.

SECT Segment-Trab (Eingang):

Dieser Anschluß ist für den Betrieb mit einer MMU vorbereitet.

Achtung: Die überwiegende Zahl aller Steuersignal-Ein-/Ausgänge sind low-aktiv, ausgenommen der AD-Bus, vier Statusausgänge und die SegmentadreseAusgänge. Ferner sind der AD-Bus, die Segmentadreß-Ausgänge und die Zeitsteuer- und Betriebszustands-Ausgänge Tri-State-Anschlüsse.

Registerorganisation

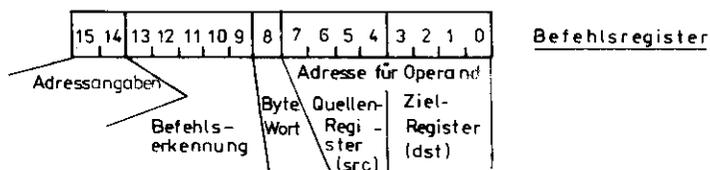
Allgemeine Register

Byte-Register		Byte-Register		Register-Kennung	Word (16 Bit)	Long-Word (32 Bit)	4-fach Word (64 Bit)
7	RH0	0H ¹⁾	0	0000	R0	RR0	RQ0
15	RH1	1H	0	0001	R1		
	RH2	2H		0010	R2		
	RH3	3H		0011	R3	RR2	
	RH4	4H		0100	R4		
	RH5	5H		0101	R5	RR4	RQ4
	RH6	6H		0110	R6		
	RH7	7H		0111	R7	RR6	
				1000	R8		
				1001	R9	RR8	RQ8
				1010	R10		
				1011	R11	RR10	
				1100	R12		
				1101	R13	RR12	RQ12
				1110	R14		
	System-Stackpointer		(Segment)	1110	R14'	RR14	
	Normal-Stackpointer		(Segment)				
	System-Stackpointer		(Offset)	1111	R15		1) Reg.-Kennung hexadezimal!
	Normal-Stackpointer		(Offset)		R15'		

Spezial-Register

Reserviert	Res.	} Programmstatusregister
SEGS/N EPU/VI/NVI C Z S P/V/D/H	FCW (Flag u. Controlword)	
Segment-Nr.	PC-S (Programm Counter-Segment Nr.)	
Segment-Offset	PC-O (Programm Counter-Offset)	
Segment-Nr.	} Programmstatusbereichszeiger	
höH-Offset		0 0 0 0 0 0 0 0
RE Rate ROW	Refresh-Zähler	
(Auffrisch-Rate)	(Zeilen-Adresse)	

Das Befehlsregister und das Flag- und Steuerwort-Register



Zeitabläufe

Der U8001 führt Befehle durch schrittweise Abarbeitung von Folgen von Basis-Maschinenzyklen wie Speicherlesen und -schreiben, E/A-Lesen und -Schreiben, Interruptbestätigung und interne Funktionen aus. Jeder dieser Grundzyklen erfordert zwischen 3 und 10 Taktzyklen zur Abarbeitung. Befehle, die mehr als 10 Taktzyklen erfordern, werden auf verschiedene Maschinenzyklen verteilt. Somit ist kein Maschinenzyklus länger als 10 Taktzyklen und eine schnelle Reaktion auf ein BUSRQ ist gewährleistet.

Der Befehlscode wird durch eine Speicherlese-Operation geholt. Ein Refresh-Zyklus kann eingefügt werden direkt nach Abschluß eines ersten Befehls-Lesezyklus, ebenso während der Abarbeitung folgender Befehle:

MULT, MULTL, DIV, DIVL, HALT, alle Verschiebeoperationen, alle Blockbewegungen und Multi-Micro-Anforderung (MREQ).

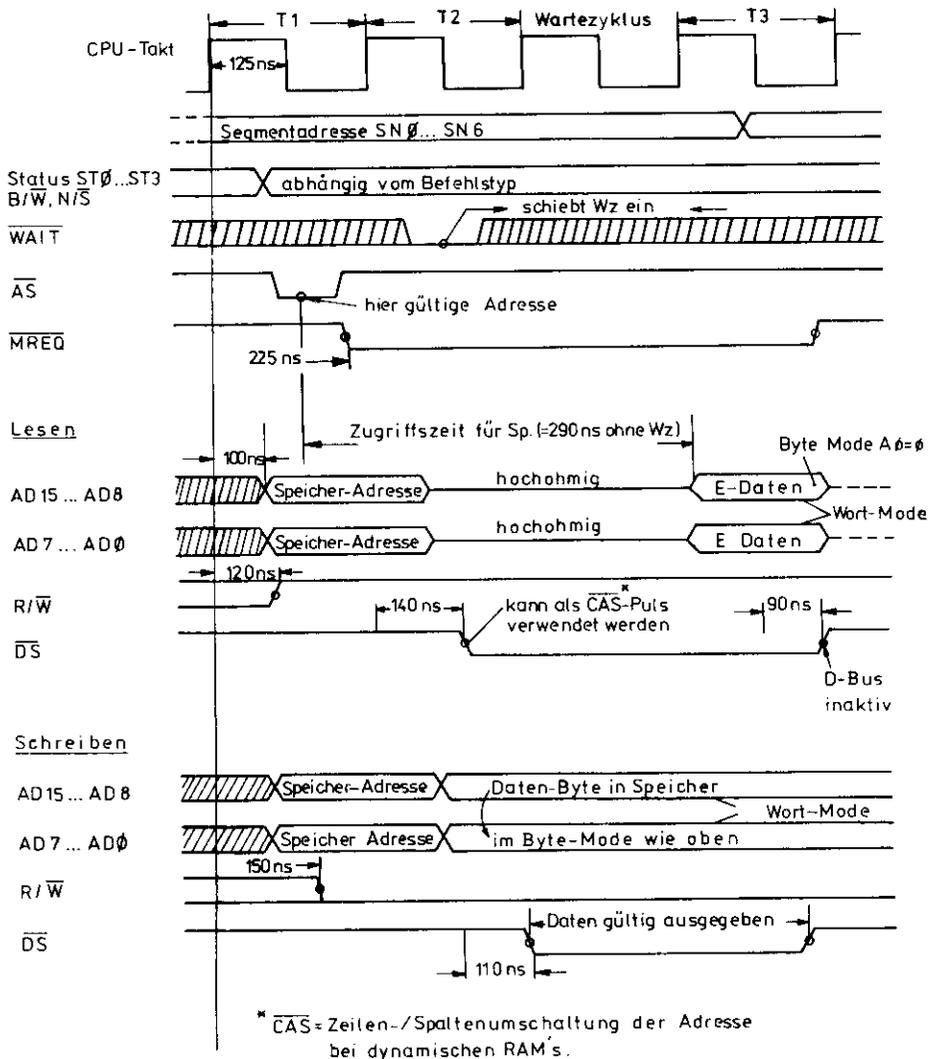
Die folgenden Zeitdiagramme zeigen die relativen Zeitverhältnisse aller CPU-Signale während jeder der Grundoperationen. Benötigt ein Maschinenzyklus zusätzliche Takte für CPU-interne Operationen, werden zwischen 1 und 5 Taktzyklen hinzugefügt. Speicher- sowie E/A-Schreiben und -Lesen können durch Aktivierung des WAIT-Einganges verlängert werden.

Speicher-Lesen und -Schreiben

Speicherlesen und Befehlslesen sind identisch ausschließlich der Statusinformationen auf den Ausgängen ST0 bis ST3. Während eines Speicherlesezyklus wird zu Beginn der ersten Taktperiode eine 16-bit-Adresse auf die Ausgänge ADO bis AD15 gelegt (beim U8001 wird die 7-bit-Segmentnummer um eine Taktperiode früher auf die Ausgänge SN₀ bis SN₆ gelegt als die 16-bit-Adresse, um die Verzögerung im Speicherverwaltungsschaltkreis zu kompensieren).

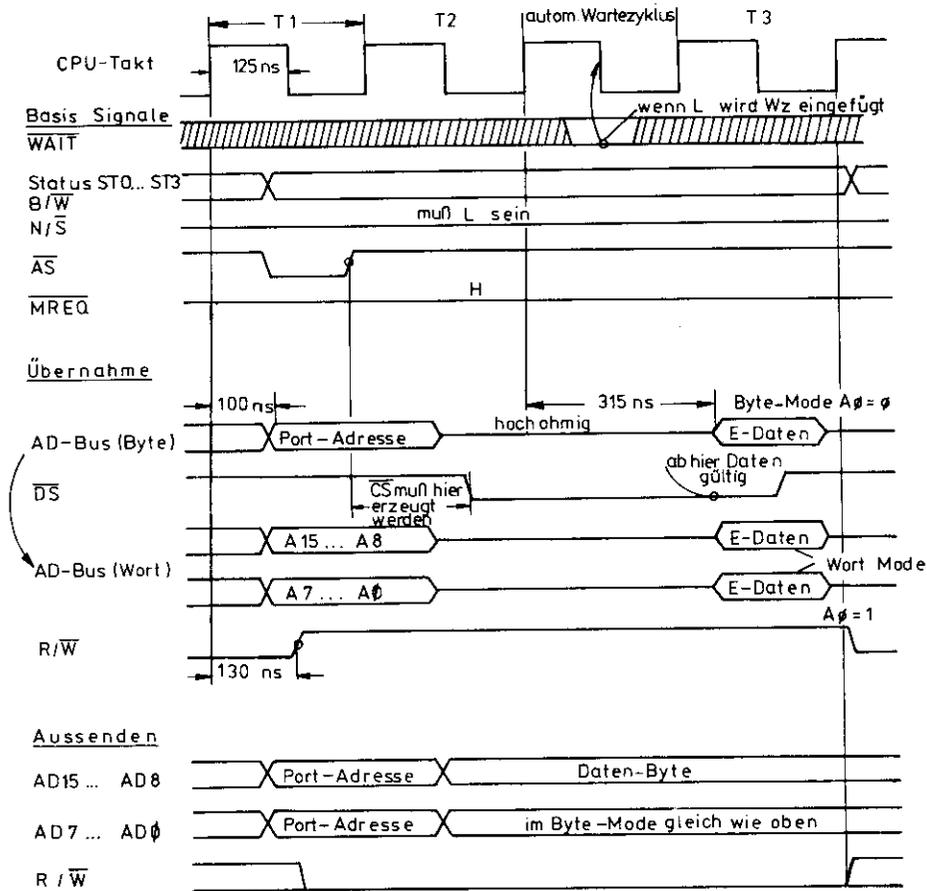
Eine gültige Adresse wird durch eine L-H-Flanke an AS angezeigt. Status- und Betriebsartinformation werden früh im Speicherzugriffszyklus gültig und bleiben die ganze Zeit stabil. Der Zustand des WAIT-Einganges wird im zweiten Taktzyklus durch die fallende Taktflanke ausgewertet. Ist WAIT = L, wird eine zusätzliche Taktperiode zwischen T2 und T3 eingefügt. In der Mitte dieses Zyklus wird WAIT erneut abgefragt, weitere WAIT-Zyklen können eingefügt werden. Das erlaubt den Anschluß langsamer Speicher. Während des WAIT-Zustandes werden keine Steuersignale verändert.

Da der U8001-Speicher wortorganisiert ist, wird der Speicher Byteweise adressiert. In einem 16-Bit-Wort wird das höherwertige Byte (D₈ - D₁₅) durch die niederwertige Adresse (A₀ = L), das niedrigwertige Byte (D₀ - D₇) durch die höherwertige (A₀ = H) adressiert.



E/A-Operation

Der Zeitablauf von E/A-Vorgängen ist ähnlich dem bei der Speicherarbeit. Abweichend davon wird aber automatisch ein WAIT-Zyklus zwischen T_2 und T_3 eingefügt. Der segmentierte UB001 verwendet 16 Bit-I/O-Adressen.

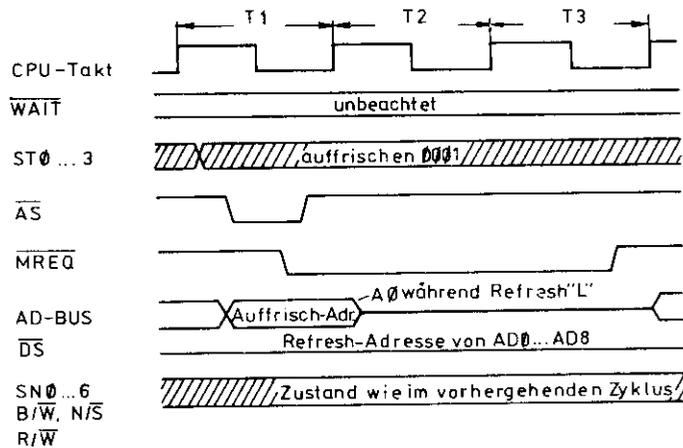
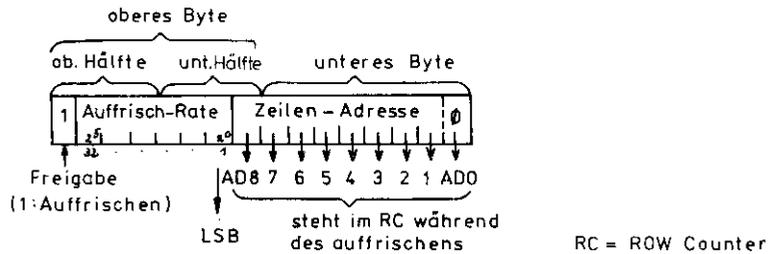


Das Refresh-Register und Refresh-Diagramm

Der U8001 enthält ein 16-Bit-Auffrisch-Register. Die Refresh-Adresse wird auf der Breite des gesamten AD-Busses ausgegeben. Das Register ist aufgeteilt in: Freigabe (1 Bit), Ratezähler (6 Bit), Zeilenzähler (9 Bit).

Der 9-Bit-Zeilenzähler (RC) kann 256 Zeilen adressieren. Er wird immer um 2 inkrementiert, wenn der 6-Bit-Ratezähler seinen Endwert erreicht hat. Der Ratezähler definiert die Zeit, nach der jeweils ein Auffrischzyklus erfolgt ist. Er wird mit dem vierten Teil der Taktfrequenz dekrementiert und liefert 64 verschiedene Auffrischraten von 1... 64 µs in Schritten einer Mikrosekunde. Die Refresh-Zyklen werden wie folgt gebildet:

Der 2^0 -Wert (AD_0) des Zeilenzählers RC ist immer Null, d. h. bei der Refresh-Adreßausgabe wird immer eine gerade Adresse erscheinen, so daß immer nur das Zeilenadreibregister in den dynamischen RAMs geladen wird. Ferner muß das Auffrisch-Freigabebit Nr. 15 gesetzt sein, sonst erfolgt kein Refresh. Das LSB des Ratezählers ist Bit 9. Steht Null im Ratezähler, erfolgt erst nach 256 Taktzyklen (entsprechend 64 µs bei einem 4 MHz-Takt) ein Refreshzyklus. Steht 1 im Ratezähler (1000001 ...), erfolgt nach 4 T-Zyklen ein Refresh. Steht 2 darin nach 8 T-Zyklen usw., d. h. hat dieser Zähler den Wert Null erreicht, wird die Zeilenadresse im 9-Bit-Zeilenzähler ausgegeben und danach der programmierte Ratewert neu geladen.

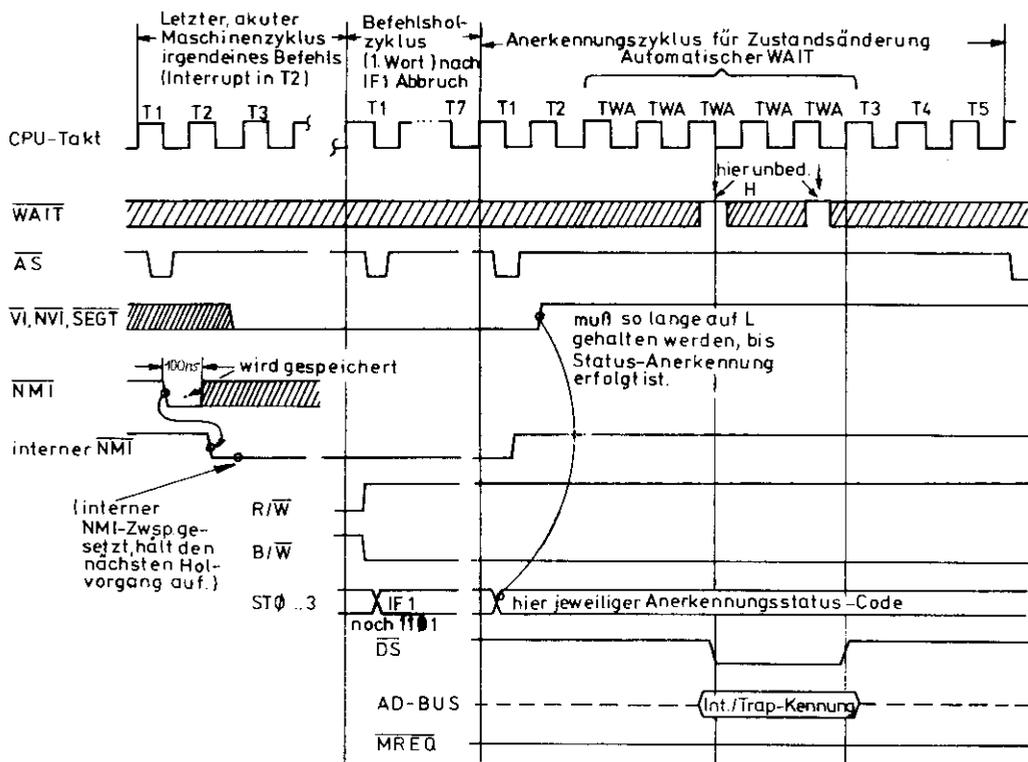


Interrupt- und Segment-Trap-Anforderung und -bearbeitung

Die U8001-CPU erkennt drei Interrupt-Eingaben (nicht maskierbar, vektoriert und nicht vektoriert) und eine Segment-Trap-Eingabe. Jeder H-L-Übergang auf dem NMI-Eingang wird asynchron als Flanke erkannt und setzt das interne NMI-Latch. Die VI, NVI und SEGT-Eingaben werden ebenso wie der Status des internen NMI-Latch zu Beginn von T3 im letzten Maschinenzyklus jedes Befehls abgefragt. In Reaktion auf ein Interrupt oder Trap wird der folgende IF-Zyklus bearbeitet, aber dann abgebrochen. Der Befehlszähler wird nicht erhöht, der Stackpointer jedoch dekrementiert.

Der nächste Maschinenzyklus ist der Interruptquittungszyklus. Dieser hat fünf automatische WAIT-Zustände, zusätzliche sind möglich.

Nach dem letzten WAIT-Zyklus liest die CPU die Information von AD₀ ... AD₁₅ und speichert sie zwischen, um sie für die weitere Abarbeitung zu sichern. Dieses Datenwort identifiziert die Quelle des Interrupt oder Trap. Bei nicht vektoriertem und nicht maskiertem Interrupt können alle 16 Bits Statusinformationen von Peripheriegeräten beinhalten. Beim vektorierten Interrupt ist das niedrigere Byte der Sprungvektor, während das höhere Byte einen zusätzlichen Anwenderstatus haben kann. Beim Segment-Trap ist das obere Byte die Identifikation für die MMU (Memory Management Unit), das untere Byte unbestimmt. Nach dem Quittungszyklus indiziert der N/S-Ausgang den automatischen Übergang in den System-Mode.



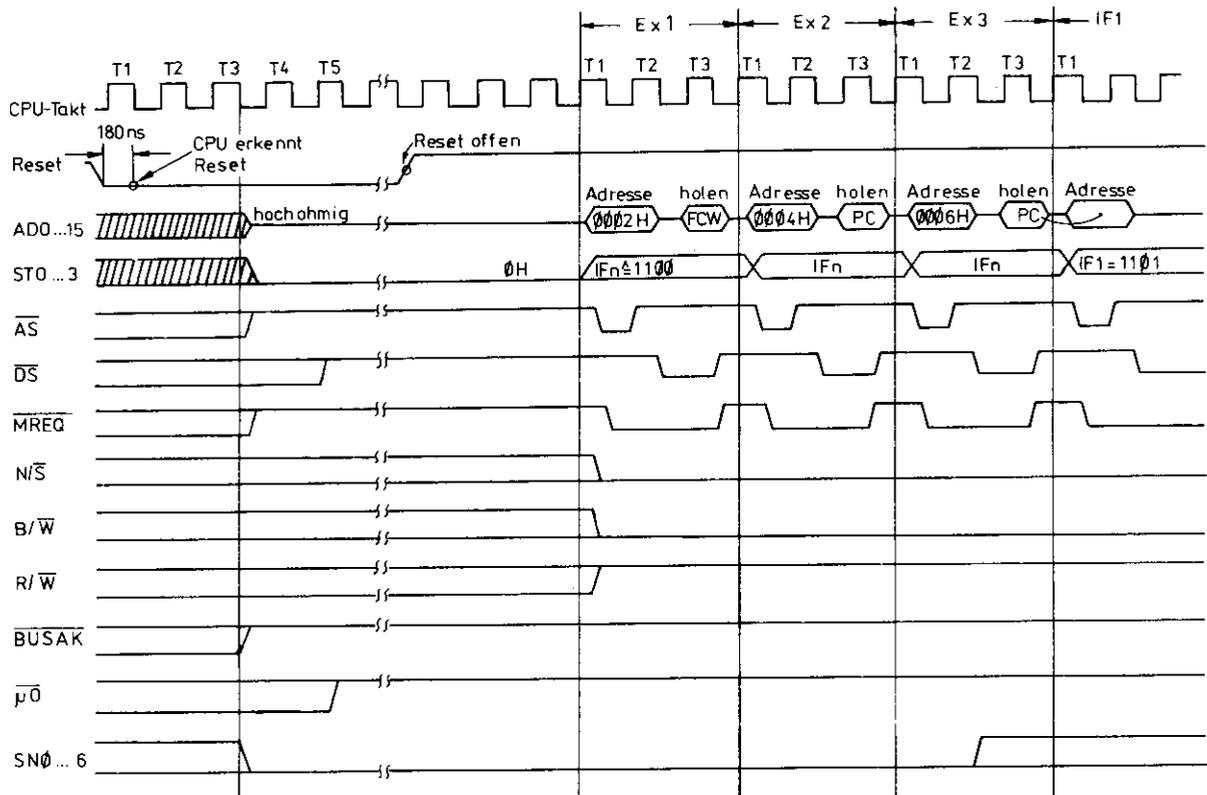
Antwort auf RESET

Ein L-Signal auf dem **RESET**-Eingang bewirkt innerhalb der nächsten max. 5 Taktzyklen die folgenden Effekte:

- $AD_0 \dots AD_{15}$ werden hochohmig
- \overline{AS} , \overline{DS} , \overline{MREQ} , \overline{BUSAK} und $\overline{M0}$ werden auf H gesetzt
- $ST_0 \dots ST_3$ und $SN_0 \dots SN_6$ werden auf L gesetzt
- Refresh ist unmöglich
- R/\overline{W} , B/\overline{W} und N/\overline{S} werden wirkungslos

Wird **RESET** für 3 Taktzyklen H, werden zwei aufeinanderfolgende Speicherlesezyklen im System-Mode durchgeführt.

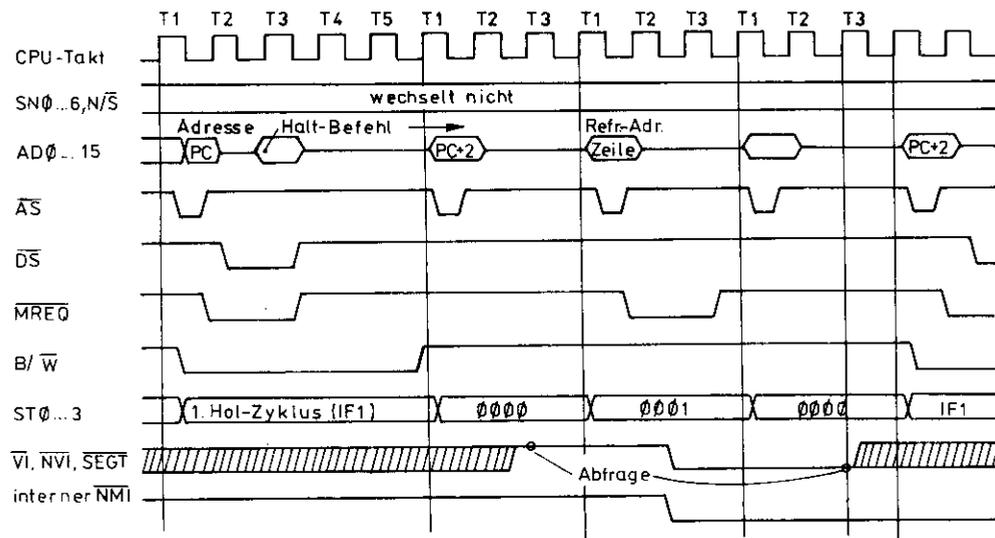
Beim U8001 wird im ersten Zyklus das Flag- und Steuerwort vom Platz **0002** gelesen, im nächsten die 7-Bit-Befehlszähler-Segmentnummer vom Platz **0004**, im nächsten der 16-Bit-Befehlszählerstand vom Platz **0006** und der folgende **IF1**-Zyklus startet das Programm.



Antwort auf HALT-Befehl

Ein HALT-Befehl führt eine unbegrenzte Anzahl von internen Operationen mit je 3 Zyklen durch, unterbrochen durch Refresh-Zyklen bei Bedarf. HALT kann nur durch Segment-Trap oder Rücksetzen verlassen werden.

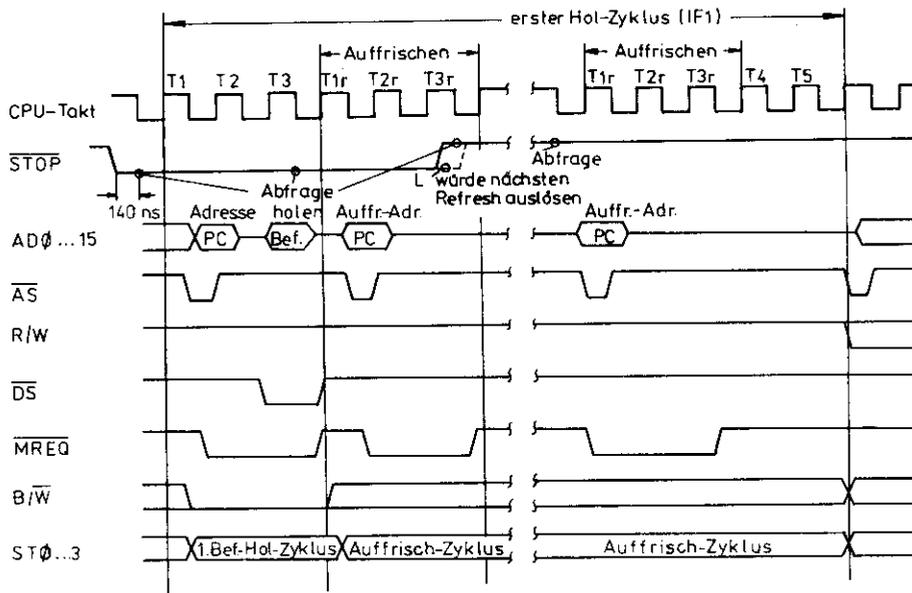
Die CPU fragt die Eingänge \overline{VI} , \overline{NVI} , \overline{NMI} und \overline{SEGT} zu Beginn jedes T3-Zyklus ab. Wird einer dieser Eingänge während zweier aufeinanderfolgender Abfragen als aktiviert erkannt, wird zwar der darauffolgende IF1-Zyklus durchgeführt, aber ignoriert und der Interrupt-Quitungszyklus wird gestartet.



Einzel-Schritt-Operation mit STOP

Hiermit kann ein Befehl fast isoliert ausgeführt werden. Der STOP-Eingang wird während der H/L-Flanke des letzten Taktes irgendeiner Befehlsausführung abgefragt. Ist dieser L, holt die CPU nur noch das erste Wort des nächsten Befehls (Status Nr. 13) und beginnt nach den drei T-Zyklen mit einem Refresh-Zyklus. Der STOP-Eingang wird wieder im letzten Takt, hier T3, abgefragt und bei L ein weiterer Refresh-Zyklus angefügt. Ist bei der nächsten Abfrage STOP weiterhin L, folgt ein weiterer Refresh-Zyklus. Dies setzt sich beliebig fort. Gegensätzlich zum programmierten Ablauf im Refresh-Register folgen ständig Refresh-Zyklen, wenn in der Mitte des 3. Taktes STOP gleich L war. Stellt die CPU im 3. Takt eines Refresh-Zyklus ein H fest, wird noch ein Refresh-Zyklus gemacht und der IF1 schließt mit den Takten T4 und T5 ab. Wenn STOP also L gesetzt wird, ist der vorhergehende Befehl ausgeführt. Nach dem Holen des ersten Wortes vom folgenden Befehl wird die CPU beliebig lange angehalten, solange STOP zu dem genannten Zeitpunkt L ist. STOP ist also taktsynchron.

Achtung: Wird STOP längere Zeit aktiv gehalten, werden, da alle 500 ns ein Refresh-Zyklus läuft, kritische Temperaturwerte der RAM-Chips erreicht, Zerstörungsgefahr!



Bezeichnungen/Abkürzungen

Control Bits:

SEG Segmentation Mode
S/N System Mode/Normal Mode
EPA Extended Prozessor Architektur (setzt Vorhandensein von EPU = Extended Prozessor Unit voraus)
VI Vectored Interrupt enable (VI erlaubt)
NVI Non Vectored Interrupt enable (NVI erlaubt)

Register

RH 0 ... 7	}	8 Bit-Register	oberes Byte
RL 0 ... 7			unteres Byte
R 0 ... 15		16 Bit-Register	(Wort-Register)
RRO _{2,4} ... 14		32 Bit-Register	(Langwort-Registerpaar)
RQO _{4,8,12}		64 Bit-Register	(4-fach Wort, Registerquadrupole)
Rn			(allgemein: Wortregister n)

Mit besonderer Bedeutung: R14 (Normal) Stack-Pointer-Seg.Nr.
R14' (System)
R15 (Normal) Stack-Pointer-Offset
R15' (System)

SP Stack Pointer (R14, R15)
PC Programm Counter (PC/PC-Seg.Nr./PC-Offset)
FCW²⁾ Flag and Control Word
PSAP¹⁾ Programm Status Area Pointer (Zeiger für Programm-Status-Bereich, in dem alle Angaben für Traps und Interrupts gespeichert sind)
PSA Programm Status Area
REFRESH¹⁾ Refresh Register (programmierbar)

1) Nur mit dem Befehl LDCTL (= Load control) einstellbar

2) wie 1), wird bei Trap und Interrupt gerettet und durch Parameter aus PSA ersetzt, Rückladen erfolgt mit dem Befehl IRET (im System Mode und Segmentation Mode).

+ bei den so gekennzeichneten Befehlen ist für den unsegmentierten Modus ein Wortregister, und für den segmentierten Modus ein Wortregisterpaar zu verwenden.

++ vom Befehl LDB (bei Direktwert) existieren zwei Befehlsformate, der Assembler verwendet das Kurzformat

* Kennzeichen für privilegierte Befehle (nur im System-Mode verwendbar)

Direktwert

@ indirekte Adresse

IM immediate (Befehl mit Direktwerten)

lsb Least significant bit (niederwertigstes bit)

msb most significant bit (höchstwertigstes bit)

tmp temporary (zeitweiliger interner Zwischenspeicher)

dst destination (Ziel)

src source (Quelle)

Assembler-Schreibweise (allgemein)

Befehl dst, src	Grundform der Befehle
Befehl dst, src, n	Befehle mit n-facher Wirkung (n = Zahl)
Befehl dst, src, r	Befehle mit mehrfacher Wirkung (r = Register, Zähler)
Befehl dst, src, r, cc	wie oben, mit Bedingungscode
Befehl dst	Befehl mit nur einem Operand/Ziel
Befehl cc, dst	wie oben, Ausführung nur bei Bedingung
Befehl r, dst	wie Befehl dst, mit mehrfacher Wirkung (r = Register, Zähler)
Befehl src	Befehl mit nur einem Operand/Quelle
Befehl src1, src2, r	Befehl mit 2 Quellen und mehrfacher Wirkung (r = Register, Zähler)
Befehl cc	Bedingungsabfrage
Befehl int	Interruptsteuerung
Befehl flag	Flagsteuerung

Bedeutung und Formen

dst = destination (Ziel)

src = source (Quelle)

R Rd, Rbd, RRd RQd
(Wortreg.) (Bytereg.) (Registerpaar) (2er Reg.-Paar)

Rs, Rbs, RRs, RQs

IM ---

data

DA address

address

IR @ Rd

@ Rs

X addr (Rd)

addr (Rs)

RA address

address

BA Rd (# displacement)

Rs (# disp)

BX Rd (Rx)

Rs (Rx)

Allgemein wird die Adressierungsart für die src-Adresse angegeben (außer für spezielle LOAD-Befehle und Befehle mit nur 1 Adresse)

Adressangaben

address/addr

Nonsegmented

address

Segmented SS |«Segment» Offset|

Segmented SL «Segment» Offset

RA displacement (Verschiebung)

CD: Condition Codes

CC	Math. Bed.	Bedeutung	Flags	Binär-codierung
-	nein	Always false	--	0000
-	ja	Always true	--	1000
LT	<	Less than	(S XOR V) = 1	0001
GE	≧	Greater than or equal	(S XOR V) = 0	1001
LE	≦	Less than or equal	(Z OR(S XOR V)) = 1	0010
GT	>	Greater than	(Z OR(S XOR V)) = 0	1010
ULE	≧	Unsigned less than or equal	(C OR Z) = 1	0011
UGT	>	Unsigned greater than	(C OR Z) = 0	1011
OV		Overflow	V = 1	0100
PE		Parity even	P = 1	0100
NOV		No overflow	V = 0	1100
PO		Parity odd	P = 0	1100
MI	-	Minus	S = 1	0101
PL	+	Plus	S = 0	1101
Z	= 0	Zero	Z = 1	0110
EQ	=	Equal	Z = 1	0110
NZ	≠ 0	Not zero	Z = 0	1110
NE	≠	Not equal	Z = 0	1110
C		Carry	C = 1	0111
ULT	<	Unsigned less than	C = 1	0111
NC		Not carry	C = 0	1111
UGE	≧	Unsigned greater than or equal	C = 0	1111

Format und Inhalt der Angaben im Befehl

Bezeichnung	Länge	Bedeutung	Wert
Rbd, Rbs	4 Bit	Byteregister	RH0...RH7, RLO...RL7
Rd, Rs	4 Bit	Wortregister	RO...R15
RRd, RRs	4 Bit	Wortregisterpaar	RR0, RR2, RR4...RR14
RQd, RQs	4 Bit	4 Wortregister	RQ0, RQ4, RQ8, RQ12
r	4 Bit	Register (als Zähler)	RO...R15
n	4 Bit	Nummer (Anzahl)	immediate value
cc	4 Bit	Conditioncode	cc-code
x	4 Bit	Indexregister	RO...R15
data	variabel	Direktoperand	immediate value
address, addr	variabel	Adreßangabe	Abhängig von Segm.
displ	variabel	Displacement	

Adressierungsarten

Art		Befehl	Register	Speicher
IM	Immediate	Instruktion Operand		
R	Register	Instruktion	Operand	
IR	Indirekt Register	Instruktion	Address	Operand
DA	Direkt Address	Instruktion Address		Operand
X	Indexed Address	Instruktion Base Address	Displacement	Operand
RA	Relativ Address	Instruktion Displacement	Progr. Counter	Operand
BA	Base Address	Instruktion Displacement	Base Address	Operand
BX	Base Index	Instruktion	Base Address Displacement	Operand

Befehlsformate

Adresse		nicht segmentiert	segmentiert	
			Short Offset	Long Offset
—		^	Befehl	^
IM	1	Byte	Befehl Data Data	^
	2		Befehl Data	
	3	Langwort	Befehl Data (H) Data (L)	
	4		Befehl Data	
R	1	Byte	Befehl	^
	2		Befehl Data	
	3	Befehl Data Data		
	4	Wort	Befehl Data	
	5		Befehl Data (H) Data (L)	
	6	Langwort	Befehl Befehl	
IR	1	Wort	Befehl	^
	2		Befehl Befehl	
	3	Befehl Data		
	4	Byte	Befehl Data Data	
	5		Befehl Data	
DA	1		Befehl Port	
	2	Befehl Adresse	Befehl Ø Segment Offset	Befehl 1 Segment 00000000 Offset
	3	Befehl Befehl Adresse	Befehl Befehl Ø Segment Offset	Befehl Befehl 1 Segment 00000000 Offset

Adresse		nicht segmentiert	segmentiert	
			Short Offset	Long Offset
DA	4	Befehl Adresse Data Data	Befehl Ø Segment Offset Data Data	Befehl 1 Segment ØØØØØØØØ Offset
	5	Befehl Adresse Data	Befehl Ø Segment Offset Data	Befehl 1 Segment ØØØØØØØØ Offset Data
	6	Befehl Data Adresse	Befehl Data Ø Segment Offset	Befehl Data 1 Segment ØØØØØØØØ Offset
X	1	Befehl Adresse	Befehl Ø Segment Offset	Befehl 1 Segment ØØØØØØØØ Offset
	2	Befehl Befehl Adresse	Befehl Befehl Ø Segment Offset	Befehl Befehl 1 Segment ØØØØØØØØ Offset
	3	Befehl Adresse Data	Befehl Ø Segment Offset Data	Befehl 1 Segment ØØØØØØØØ Offset Data
	4	Befehl Adresse Data Data	Befehl Ø Segment Offset Data Data	Befehl 1 Segment ØØØØØØØØ Offset Data Data
	5	Befehl Data Adresse	Befehl Data Ø Segment Offset	Befehl Data 1 Segment ØØØØØØØØ Offset
RA	1	} $\hat{=}$ }	Befehl Displacement	} $\hat{=}$ }
	2		Befehl Displacement	
BA		} $\hat{=}$ }	Befehl Displacement	} $\hat{=}$ }
BX		} $\hat{=}$ }	Befehl Befehl	} $\hat{=}$ }

Interrupt und Trap Behandlung

Interrupt - asynchrones Auftreten
Trap - synchrones Auftreten

} beide werden annähernd gleich behandelt

Interrupttypen: nicht maskiert
vektorierte Interrupts
nicht vektorierte Int.

} beide maskierbar

Traptypen: System call
nicht implementierter Befehl
privilegierter Befehl
Segmenttrap (Ext. Trap)

Prioritäten: interne Traps
nicht maskierte Interrupts
Segmenttraps
vektorierte Interrupts
nicht vektorierte Interrupts

Bildung des Identifikators

int. Traps: erstes Wort der Trapoperation
NVI, NMI: Daten-Bus entspricht dem Gerätestatus
VI: Low-Byte ist der Sprungvektor
High-Byte entspricht einer Statusinformation
Segm.-Trap: Low-Byte ist undefiniert
High-Byte ist MMU-Identifikator

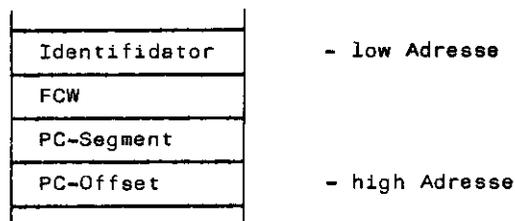
Arbeitsweise bei Vektorinterrupt

Der entsprechende PSA-Inhalt wird adressiert über:

PSAP + 60 + 2 x Vektor, d. h. Vektor 0 PSAP + 60
Vektor 2 PSAP + 64

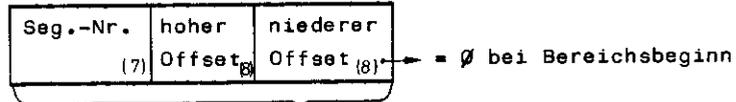
Beachte: Beim U8001 nur gerade Vektoren verwenden!

Gerettetes Programmstatusformat im Systemstack

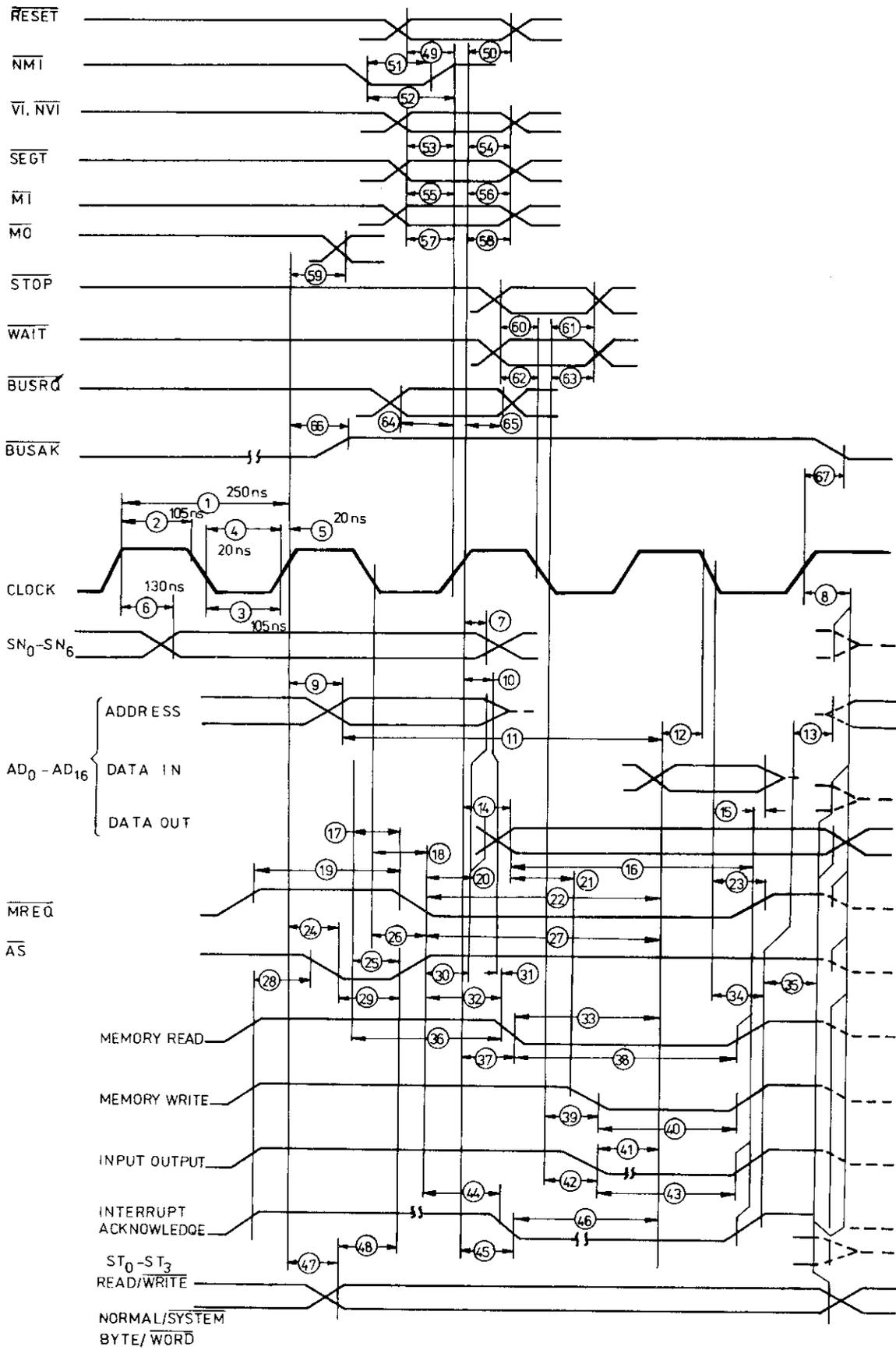


Programmstatusbereich

Programmstatusbereichszeiger



Anz. Hex	Byte Dez		
0	0		
			reserviert
8	8	Res. FCW PC-Seg. PC-Off.	Trap-erweiterte Befehle
10	16	Res. FCW PC-Seg. PC-Off.	Trap-privilegierte Befehle
18	24	Res. FCW PC-Seg. PC-Off.	Trap-Systemcall
20	32	Res. FCW PC-Seg. PC-Off.	Trap-Segment
28	40	Res. FCW PC-Seg. PC-Off.	nicht-maskierter Interrupt
30	48	Res. FCW PC-Seg. PC-Off.	nicht-vektoriertes Interrupt
38	56	Res. FCW PC-Seg. 1 PC-Off. 1	vektorierte Interrupts
40	64	PC-Seg. 2 PC-Off. 2	
.	.	.	
.	.	.	
23A	570	PC-Seg. 128 PC-Off. 128	



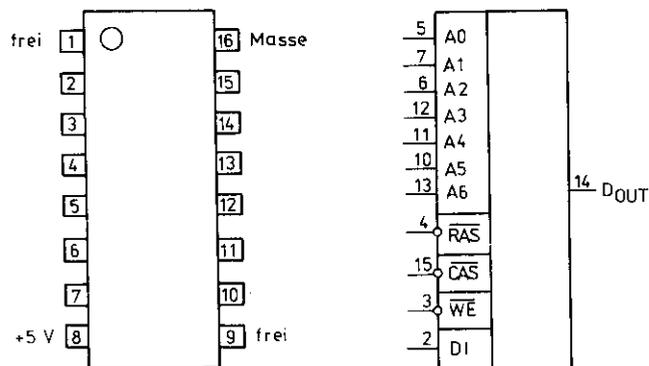
Dynamische Kenndaten (vorläufige Daten)

Number	Symbol	Parameter	Min	Max	Uni
1	TcC	Clock Cycle Time	250	2000	ns
2	TwCh	Clock Width (High)	105	2000	ns
3	TwCl	Clock Width (Low)	105	2000	ns
4	TfC	Clock Fall Time		20	ns
5	TrC	Clock Rise Time		20	ns
6	TdC (SNv)	Clock ↑ to Segment Number Valid (50 pF load)		130	ns
7	TdC (SNn)	Clock ↑ to Segment Number Not Valid	20		ns
8	TdC (Bz)	Clock ↑ to Bus Float		65	ns
9	TdC (A)	Clock ↑ to Address Valid		100	ns
10	TdC (Az)	Clock ↑ to Address Float		65	ns
11	TdA (DI)	Address Valid to Data In Required Valid	400		ns
12	TsDI (C)	Data In to Clock ↓ Setup Time	70		ns
13	TdDS (A)	\overline{DS} ↑ to Address Active	80		ns
14	TdC (DO)	Clock ↑ to Data Out Valid		100	ns
15	ThDI (DS)	Data In to \overline{DS} ↑ Hold Time	0		ns
16	TdDO (DS)	Data Out Valid to \overline{DS} ↑ Delay	230		ns
17	TdA (MR)	Address Valid to \overline{MREQ} ↓ Delay	55		ns
18	TdC (MR)	Clock ↓ to \overline{MREQ} ↓ Delay		80	ns
19	TwMRh	\overline{MREQ} Width (High)	190		ns
20	TdMR (A)	\overline{MREQ} ↓ to Address Not Active	70		ns
21	TdDO (DSW)	Data Out Valid to \overline{DS} ↓ (Write) Delay	55		ns
22	TdMR (DI)	\overline{MREQ} ↓ to Data In Required Valid	330		ns
23	TdC (MR)	Clock ↓ to \overline{MREQ} ↑ Delay		80	ns
24	TdC (ASf)	Clock ↑ to \overline{AS} ↓ Delay		80	ns
25	TdA (AS)	Address Valid to \overline{AS} ↑ Delay	55		ns
26	TdC (ASr)	Clock ↓ to \overline{AS} ↑ Delay		90	ns
27	TdAS (DI)	\overline{AS} ↑ to Data In Required Valid	290		ns
28	TdDS (AS)	\overline{DS} ↑ to \overline{AS} ↓ Delay	70		ns
29	TwAS	\overline{AS} Width (Low)	80		ns
30	TdAS (A)	\overline{AS} ↑ to Address Not Active Delay	60		ns
31	TdAZ (DSR)	Address Float to \overline{DS} (Read) ↓ Delay	0		ns
32	TdAS (DSR)	\overline{AS} ↑ to \overline{DS} (Read) ↓ Delay	70		ns
33	TdDSR (DI)	\overline{DS} (Read) ↓ to Data In Required Valid	155		ns
34	TdC (DSr)	Clock ↓ to \overline{DS} ↑ Delay		70	ns
35	TdDS (DO)	\overline{DS} ↑ to Data Out and STATUS Not Valid	80		ns
36	TdA (DSR)	Address Valid to \overline{DS} (Read) ↓ Delay	120		ns
37	TdC (DSR)	Clock ↑ to \overline{DS} (Read) ↓ Delay		120	ns
38	TwDSR	\overline{DS} (Read) Width (Low)	275		ns
39	TdC (DSW)	Clock ↓ to \overline{DS} (Write) ↑ Delay		95	ns
40	TwDSW	\overline{DS} (Write) Width (Low)	160		ns
41	TdDSI (DI)	\overline{DS} (Input) ↓ to Data In Required Valid	315		ns
42	TdC (DSI)	Clock ↑ to \overline{DS} (I/O) ↓ Delay		120	ns
43	TwDS	\overline{DS} (I/O) Width (Low)	400		ns
44	TdAS (DSA)	\overline{AS} ↑ to \overline{DS} (Acknowledge) ↓ Delay	960		ns
45	TdC (DSA)	Clock ↑ to \overline{DS} (Acknowledge) ↓ Delay		120	ns
46	TdDSA (DI)	\overline{DS} (Ack) ↓ to Data In Required Delay	420		ns
47	TdC (S)	Clock ↑ to Status Valid Delay		110	ns
48	TdS (AS)	Status Valid to \overline{AS} ↑ Delay	40		ns

Number	Symbol	Parameter	Min	Max	Uni
49	TsR (C)	$\overline{\text{RESET}}$ to Clock \uparrow Setup Time	180		ns
50	ThR (C)	$\overline{\text{RESET}}$ to Clock \uparrow Hold Time	0		ns
51	TwNMI	$\overline{\text{NMI}}$ Width (Low)	100		ns
52	TsNMI (C)	$\overline{\text{NMI}}$ to Clock \uparrow Setup Time	140		ns
53	TsVI (C)	$\overline{\text{VI}}$ $\overline{\text{NVI}}$ to Clock \uparrow Setup Time	110		ns
54	ThVI (C)	$\overline{\text{VI}}$ $\overline{\text{NVI}}$ to Clock \uparrow Hold Time	0		ns
55	TsSGT (C)	$\overline{\text{SEGT}}$ to Clock \uparrow Setup Time	70		ns
56	ThSGT (C)	$\overline{\text{SEGT}}$ to Clock \uparrow Hold Time	0		ns
57	TsMI (C)	$\overline{\text{MI}}$ to Clock \uparrow Setup Time	180		ns
58	ThMI (C)	$\overline{\text{MI}}$ to Clock \uparrow Hold Time	0		ns
59	TdC (MO)	Clock \uparrow to $\overline{\text{MO}}$ Delay		120	ns
60	TsSTP (C)	$\overline{\text{STOP}}$ to Clock \downarrow Setup Time	140		ns
61	ThSTP (C)	$\overline{\text{STOP}}$ to Clock \downarrow Hold Time	0		ns
62	TsWT (C)	$\overline{\text{WAIT}}$ to Clock \downarrow Setup Time	70		ns
63	ThWT (C)	$\overline{\text{WAIT}}$ to Clock \downarrow Hold Time	0		ns
64	TsBRO (C)	$\overline{\text{BUSRO}}$ to Clock \uparrow Setup Time	90		ns
65	ThBRO (C)	$\overline{\text{BUSRO}}$ to Clock \uparrow Hold Time	0		ns
66	TdC (BAKr)	Clock \uparrow to $\overline{\text{BUSAK}}$ \uparrow Delay		100	ns
67	TdC (BAKF)	Clock \uparrow to $\overline{\text{BUSAK}}$ \downarrow Delay		100	ns

KR565RU6

16 K dynamischer RAM (16384x1 Bit)



PIN-Belegung

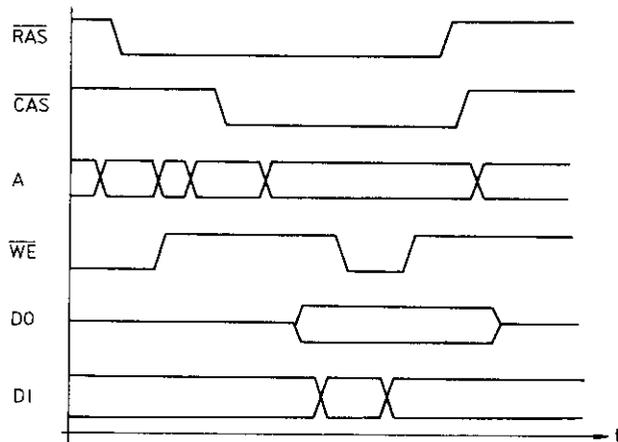
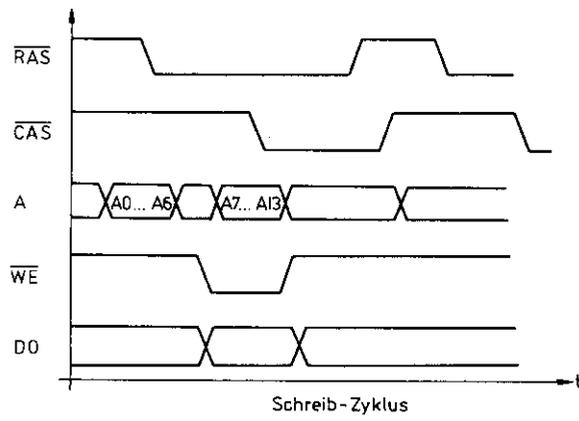
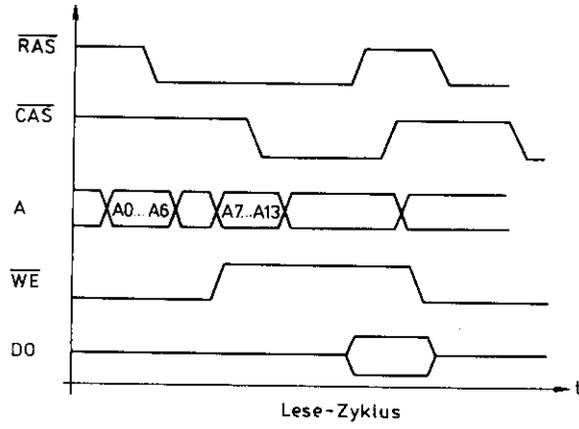
- 1 nicht belegt
- 2 Daten Eingang DI
- 3 Schreiben Freigabe $\overline{\text{WE}}$
- 4 Zeilenadress-Strobe $\overline{\text{RAS}}$
- 5 Adress-Eingang A0

6	Adreß-Eingang A2
7	Adreß-Eingang A1
8	+ 5 V (U_{CC})
9	nicht belegt
10	Adreß-Eingang A5
11	Adreß-Eingang A4
12	Adreß-Eingang A3
13	Adreß-Eingang A6
14	Datenausgang D_{OUT}
15	Spaltenadreß-Strobe-Eingang \overline{CAS}
16	Masse

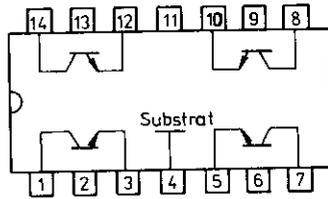
Elektrische Kennwerte

Betriebsspannung	U_{CC}	$5,0 \pm 10 \%$	V
"L"-Eingangsspannung	U_{IL}	- 1,0 bis 0,8	V
"H"-Eingangsspannung	U_{IH}	2,4 bis 6,0	V

Der KP 565 RU 6 ist als 2 x 8 Bit Speichereinheit ausgelegt. Der Zeilenadreßcode (A0 - A6) wird über \overline{RAS} (Zeilenadreß-Strobe), dem ersten Taktsignal, im Zeilenadreßregister zwischengespeichert. Ebenso wird der Kolonnenadreßcode (A7 - A13) über \overline{CAS} (Kolonnenadreß-Strobe), dem zweiten Taktsignal, im Kolonnenadreßregister zwischengespeichert. Der Zustand des Dateneingaberegisters (DI) wird durch \overline{RAS} , \overline{CAS} und \overline{WE} Signale, welche (die Zeit betreffend) jeweils zuletzt auftreten, bestimmt. Wenn sich \overline{CAS} im Zustand "H" befindet, wird der Datenausgang hochohmig. Ferner, wenn \overline{WE} bezogen auf \overline{CAS} zuerst in Schreibmodus auftritt, bleibt der Datenausgang während des ganzen Schreibzyklus hochohmig. Die obengenannte Datenausgangskonfiguration läßt die Steuerung des Datenausgangszustandes durch \overline{WE} während des Schreibzyklus (unter variierendem Signalvorgang in bezug auf \overline{CAS}) und durch \overline{CAS} während des Lesezyklus (durch Ändern seiner Impulsbreite) zu. Im Lesezyklus geht der Datenausgang über eine Zeitspanne, die der Zugriffszeit gleich ist, vom hochohmigen zum aktiven Zustand über ("L" oder "H"). Der Datenausgang bleibt aktiv bis zu dem Moment, wo \overline{CAS} in einen inaktiven "H" Zustand übergeht. In diesem Falle spielt es keine Rolle, ob \overline{RAS} eliminiert wird oder nicht. Während des gesamten Lesezyklus sollte \overline{WE} in "H" Zustand gehalten werden. Da bei der KR 565 RU 6 - Konfiguration Ausgangspuffer zur Festlegung von Ausgangsdaten fehlen, kann \overline{RAS} oder \overline{CAS} in der Speichereinheit als "Chipauswahl" Signal dienen. Bei Verwendung beider Signale für diesen Zweck, kann eine Doppelkoordinaten (X,Y) - Chipauswahlordnung realisiert werden. Auffrischung erfolgt durch Ansteuerung jeder der 128 Zeilen während einer 2 ms Dauer. Sie kann in jedem beliebigen Zyklus des Speicherbetriebes erfolgen. Es ist günstig, die Auffrischung in der durch \overline{RAS} gesteuerten Betriebsart durchzuführen, aber nur wenn sich \overline{CAS} im inaktiven "H" Zustand befindet.



Integrierte Transistorarrays mit vier Si-npn-Transistoren

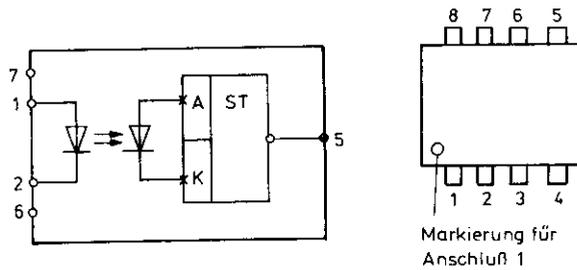


Grenzwerte, gültig für den Betriebstemperaturbereich

		max.
Kollektor-Emitter-Spannung	U_{CEO}	15
Kollektor-Basis-Spannung	U_{CBO}	20
Emitter-Basis-Spannung	U_{EBO}	5
Kollektorstrom	J_C	0,5
Impulsspitzenstrom	J_C	1,0
Gesamtverlustleistung	P_{tot}	1,3
Betriebstemperaturbereich	ϑ_a	-25... +85
Gleichstromverstärkung	min	max
$U_{CB} = 2 V, J_E = 50 mA$		
Gruppe D	112	280

MB 111

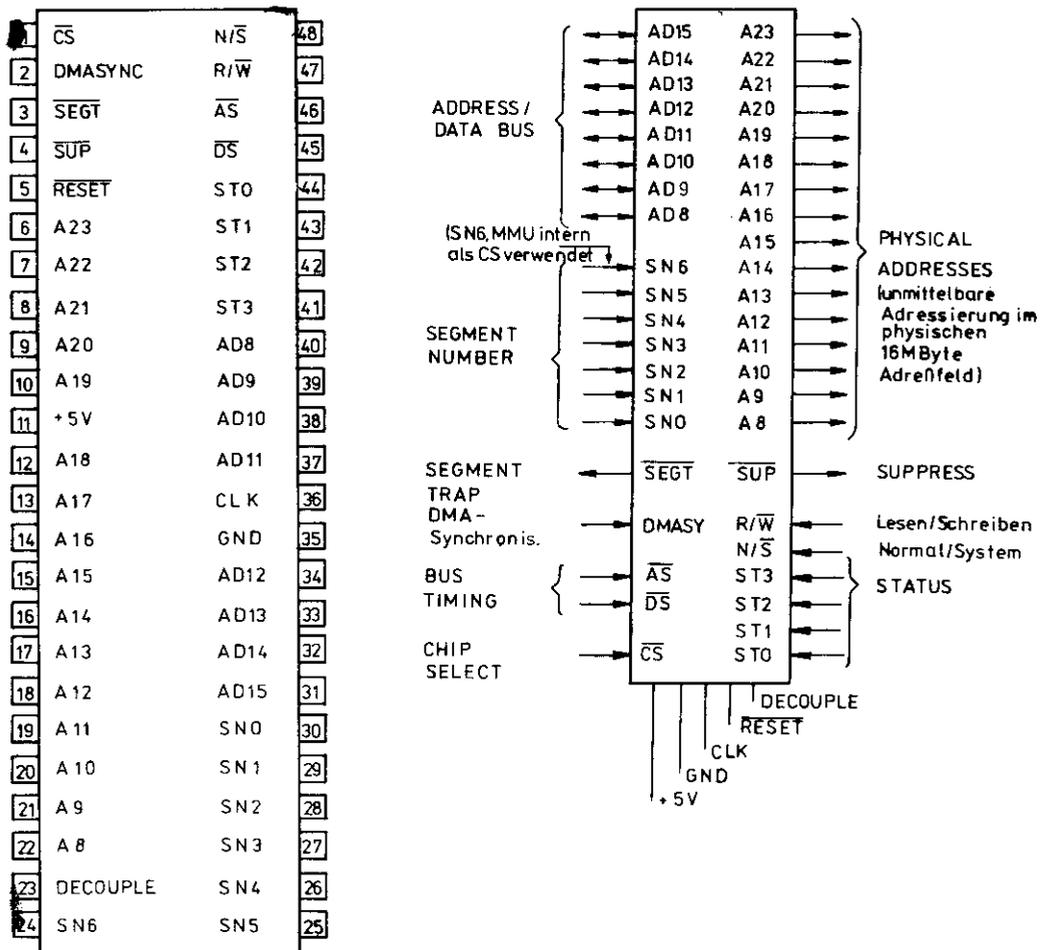
Optoelektronischer Koppler aus Infrarot-Emitter-Diode als Sender und Silizium-Fotodiode mit nachfolgender integrierter Verstärkerstufe als Empfänger zur galvanischen Trennung von Stromkreisen mit hohen Potentialdifferenzen. Der Koppler ist TTL-kompatibel. Er stellt einen hybriden Logikschaltkreis dar (logische Funktion $Y = \bar{A}$ - positive Logik). Der Kollektor des Ausgangstransistors ist offen.



Anschlußbelegung

- 1 JRED - Anode
- 2 JRED - Katode
- 3 nicht belegt
- 4 nicht belegt
- 5 JS - Ausgang
- 6 JS - Masse
- 7 JS - Betriebsspannung
- 8 JS - innere Verbindung

UB 8010C - MMU (Memory-Management-Unit)



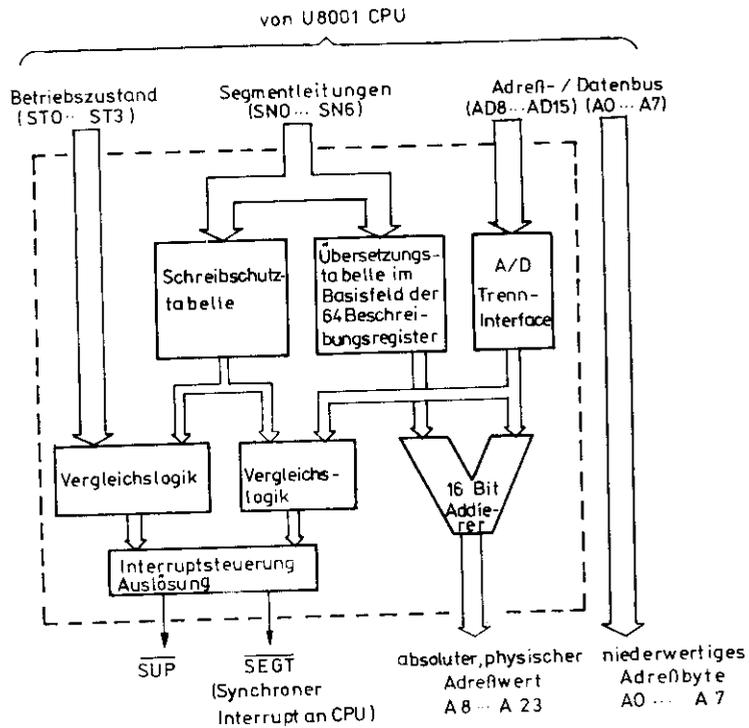
Eigenschaften der MMU

- Umsetzung der logischen 23 Bit CPU-Adresse in eine 24 Bit Adresse für das physische Adreßfeld
- Homogene Adressierung in Form einer 24 Bit Adresse
- Dynamische Verschiebung eines logischen CPU-Segments im gesamten physischen 16 MB-Feld
- Variable Größendefinition eines Anwendersegments von 256 Byte bis 64 KB in 256 Byte-Schritten ("Blöcke")
- Vielfältiger Schutz eines Anwendersegments:
 - nur lesen
 - nur im System-Modus zugriffsfähig
 - nur im Programm-Ausführungsmodus zugriffsfähig
 - Segmentenschutz vor DMA-Zugriff
 - Segmentenschutz vor CPU-Zugriff (DMA-Segment)
- 64 frei definierbare Segmente unter der Organisation einer MMU, wahlweise die ersten 64 oder die letzten 64 Segmente von den 128 durch die CPU adressierten
- Mehrfachanwendung von MMU's in einem U 8000 System
- Genereller Schutz vor willkürlichem Überschreiten eines Datensegments
- Schutz vor unberechtigtem Zugriff in das Betriebssystem
- Hardware-Synchronisation vor einem Zugriff des DMA-Steuerbausteins auf ein Segment

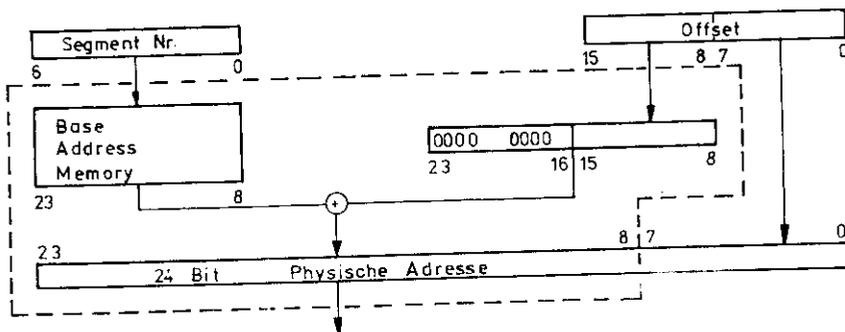
Die MMU besteht aus:

- 1 Adreß/Daten-Businterface
- 1 Segment-Beschreibungsregisterfeld (64 RAM-Register mit 32-Bit-Format)
- 1 16-Bit Adreßaddierwerk sowie eine Prüf- und Vergleichslogik
- Die interne Logik der MMU wird durch eine Reihe von Status- und Steuerregistern für die CPU transparent gehalten.
- Einige spezielle E/A-Befehle der CPU ermöglichen einen schnellen Datenaustausch bezüglich Segmentbeschreibungen.

Interner Aufbau U 8010-MMU



Adreßbildung in der MMU



- MMU-Eingänge:
- höherwertige 8 Bit Offset
 - Segmentnummer
 - Statusinformation der CPU

MMU-Kommandos (22)

- Befehle für den Zugriff auf die Segment-Beschreibungsregister

Opcode	Befehl
	(SIN) (SOUT)
08H	Lesen/Schreiben Basis-Feld
09H	" " Limit-Feld
0AH	" " Definitions-Feld
0BH	" " Segment-Beschreibung (alle 3 Felder)
0CH	" " Basis-Feld, inkrementieren von SAR (2)
0DH	" " Limit-Feld, " " "
0EH	" " Definitions-Feld, " " "
0FH	" " Segment-Beschreibung, " " "
15H	alle CPU-Inhibitflags werden im Segmentbeschreibungsregister gesetzt
16H	alle MMU-Inhibitflags werden im Segmentbeschreibungsregister gesetzt

- Befehle für den Zugriff auf die MMU-Steuerregister

Opcode	Befehl
	(SIN) (SOUT)
00H	Lesen/Schreiben Betriebsart-Register
01H	" Segmentadreibregister
20H	" DSC-Teil (Description-Selection-Counter)

- Befehle für den Zugriff auf die MMU-Statusregister

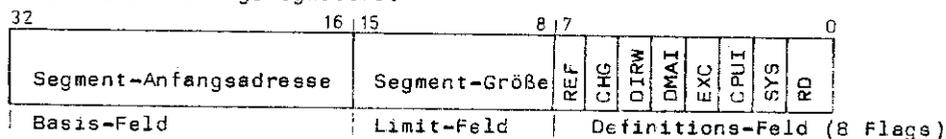
Opcode	Befehl
	(SIN)
02H	Lesen Segmentadreibdefinition
03H	" Segmentnummer
04H	" Segmentoffset (oberes Byte)
05H	" Buszyklus der CPU bei Trap
06H	" Segmentnummer
07H	" Segmentoffset (oberes Byte)
11H	Reset SAD (Segment-Adreib-Definition)
13H	" SWW-Flag in SAD
14H	" FATL-Flag in SAD

} bei Trap (03H-05H)
 } bei Befehl (06H-07H)

* Segment-Beschreibungsregister

- Die MMU hat 64 Segmentbeschreibungsregister mit je 32 Bit, es sind Lese- oder Schreibmodus möglich

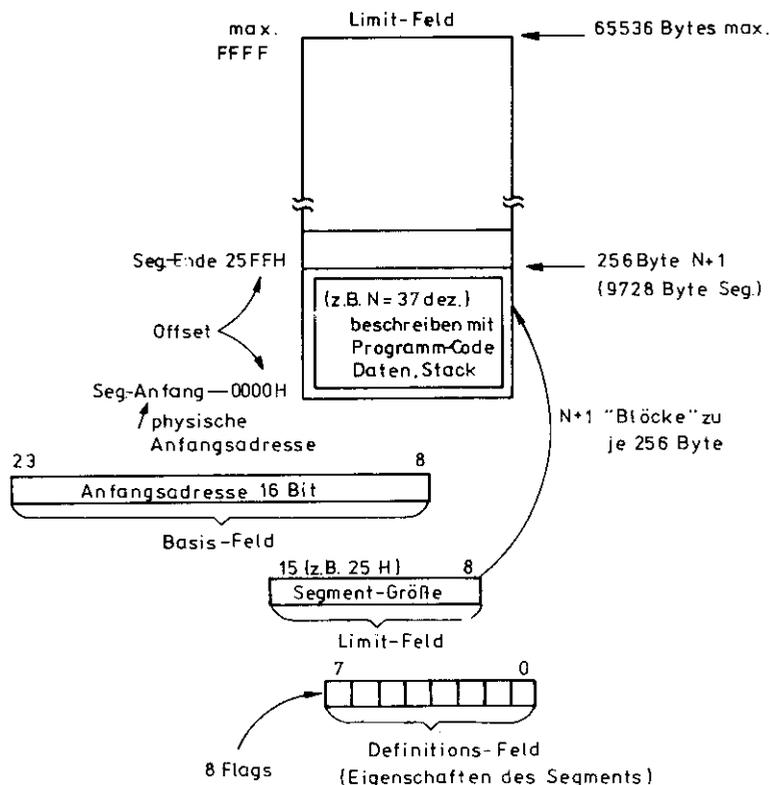
Aufbau eines Beschreibungsregisters:



- Basis-Feld: Anfangsspeicheradresse des Segmentes (0...64K, entsprechend Wert im Limit-Feld)
- Limit-Feld: Segmentgröße in "Blocks" von 256 Bytes (0=256, 1=512, FF=64 KI)
- Definitions-Feld: besteht aus 8 Flags zum Schutz des Segmentes vor unberechtigtem Schreib und Lesezugriff

- RD (Read Only): es sind nur Lesebefehle zulässig, bei Schreibbefehl wird SUP-Signal gebildet, Abfrage R/W
- SYS (System Only): es sind nur Zugriffe im Systemmode zulässig, Zugriff im Normalmodus ergibt Trap, Abfrage N/S
- CPUI (CPU Inhibit): CPU Zugriff ist verboten, nur für DMA, sonst Trap zur Programmausführung verwenden, (Status 12/13)
- EXC (Execute Only): DMA Zugriff ist verboten, Abfrage DMASYN, SUP-Signalbildung
- DIRW (Direction And Warning): Generierung eines Traps zur Anzeige einer Schreibwarnung (bei Schreibbeginn in die letzten 256 Byte eines Segmentes)
- CHG (Changed): Schreibbefehl (durch DMA oder CPU) auf das Segment, führt automatisch zum Setzen dieses Bits
- REF (Referenced): beliebiger zulässiger Zugriff auf ein Segment (DMA, CPU) ist erfolgt

Kurze Erklärung des Beschreibungsregisters anhand eines "Beispiels"!



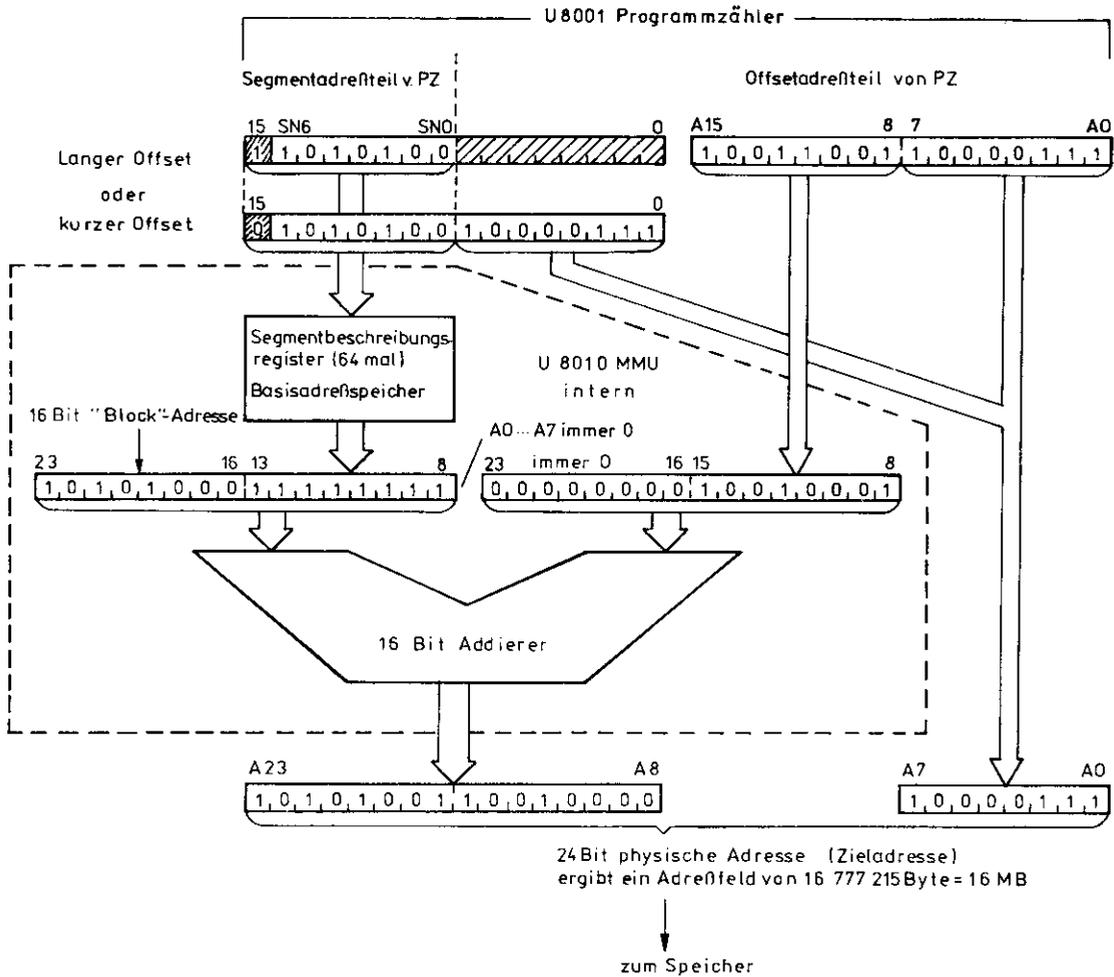
Die 32 Bit ergeben sich wie folgt:

- Im Basis-Feld (16 Bit) steht die Anfangsadresse des Segmentes im physischen Speicher-raum (vom Programmierer sinnvoll zu definieren)
- Im Limit-Feld (8 Bit) wird die Segmentgröße definiert, der Intervall der Größenänderung

ist mit 256 Byte gegeben!

Steht in diesem Registerteil der Wert 25 H, bedeutet dies 38 "Blöcke" zu je 256 Byte. Das Segment hat somit 9728 Byte. Eine Offsetadresse für dieses Segment erlaubt nur Werte zwischen 0H ... 25FFH. Ein Wert für N ergibt immer N+1 "Blöcke" zu 256 Byte.

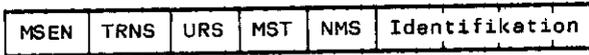
- In dem Definitions-Feld (8 Bit) wird die Definition für ein einziges Segment festgelegt. (Allein 5 dieser Flags dienen ausschließlich dem Schutz vor unberechtigtem Schreib- oder Lesezugriff.)



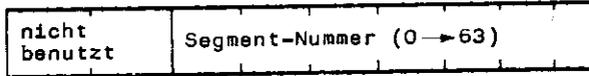
Dieses Bild zeigt die Umsetzung einer Segment- und Offsetadresse der CPU in einen physischen Adreßwert

Steuerregister

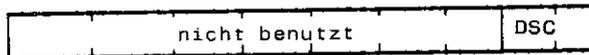
- Lese- oder Schreibmodus möglich
- Aufbau: 3 x 8 Bit Register



Betriebsregister (5 Flags)
(wird über CPU-Busabschnitt AD8 ... AD15 gesetzt)



Segmentadreßregister



Description Selection Counter

Zu Betriebsartregister

MSEN (Master Enable): MSEN=H → Freigabe der Logikfunktion der MMU
 TRNS (Translate): TRNS=L → keine Adreßumsetzung (logische in physikalische)
 URS (Upper Range Select): Abfrage SN6 und URS-Flag
 SN6=H + URS=H → MMU enthält Segment-Nr. 64-127
 SN6=L + URS=L → MMU enthält Segment-Nr. 0- 63
 MST (Multiple Select): System enthält mehrere Übersetzungstabellen
 NMS (Normal Mode Select): nur in Verbindung mit MST zu sehen
 MST=H } NMS=L Systemmodus
 } NMS=H Normalmodus
 ID (Identifikationsfeld): enthält den logischen Namen (0 ... 7) der entsprechenden MMU!

} nur bei Benutzung von mehreren MMU's!

Zu Segmentadreßregister

Es trifft die Auswahl eines Beschreibungsregisters
 Segment Descriptor Nr.: enthält die Adressen der Segmente innerhalb der MMU (beachte URS Flag)

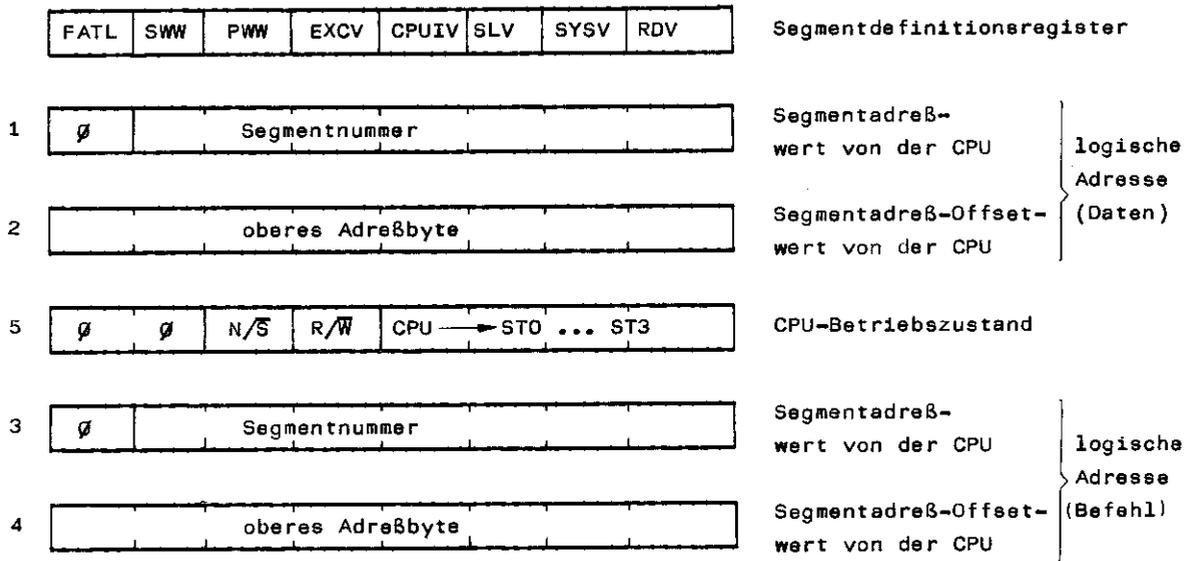
Zu Description Selection Counter

Er gibt an, welches Byte des Beschreibungsregisters angesprochen wird

- 00 H-Byte des Basis-Feldes
- 01 L-Byte des Basis-Feldes
- 10 Limit-Feld
- 11 Definitions-Feld

Statusregister

- Ein Zugriff ist nur im Lesemodus möglich
- Aufbau: 6 x 8 Bit Register



Das Segmentdefinitionsregister (es enthält die Ursache des Segmenttrap)

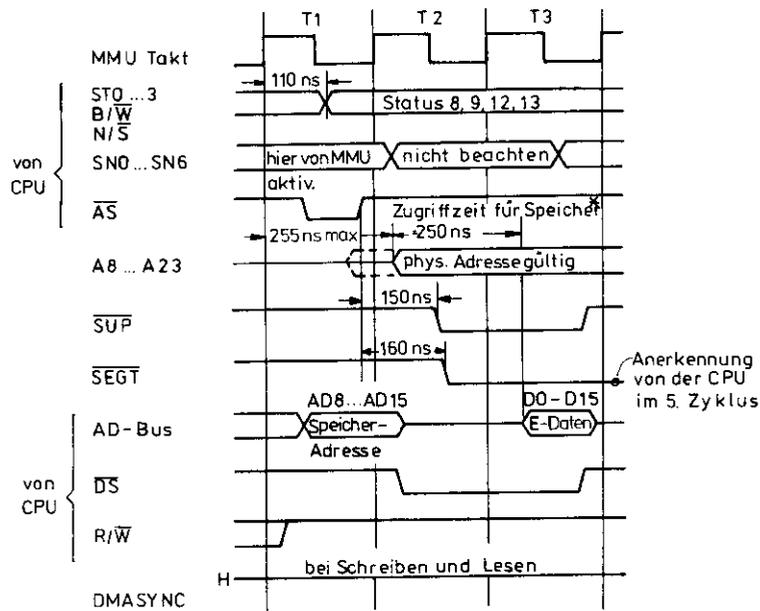
- RDV (Read Only Violation): CPU will in ein Segment schreiben, daß nur gelesen werden darf
- SYSV (System Only Violation): CPU will im Normalmodus auf ein Segment zugreifen
- CPUIV (CPU Inhibit Violation): CPU-Zugriff trotz Setzen des CPUI-Flag
- EXCV (Execute Only Violation): CPU liest nicht während eines Befehlszyklus im Ausführungssegment

Flags, die den Entstehungsort einer Störung beschreiben:

- SLV (Segment Length Violation): Offset außerhalb des legalen Segmentbereiches
- PWW (Primary Write Warning): Schreibbefehl in die letzten 256 Bytes des Segments (DIRW = H) → primäre Schreibwarnung
- SWW (Secondary Write Warning): Überlauf System-Stack möglich → Aktivierung SUP
- FATL (Fatal Condition): Speicherzugriffsfehler während einer Trapbehandlung

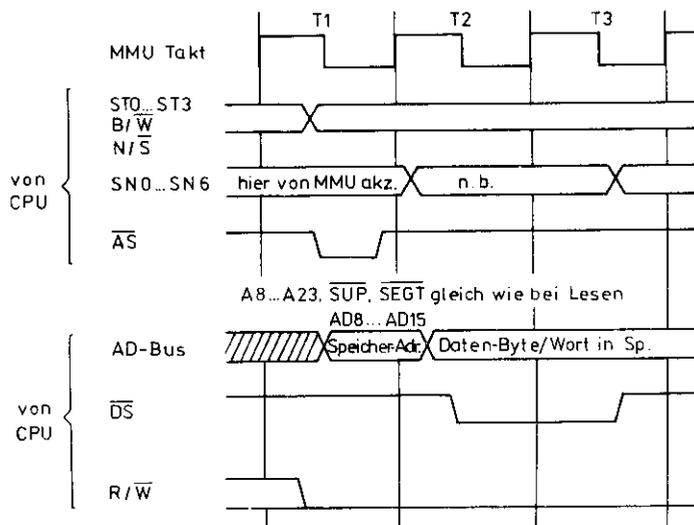
- | | | |
|------------------------|---|--|
| 1 Segmentnummer : | } | Sie beinhalten die höchsten 15 Bit der logischen Adresse, die der Trap verursacht hat |
| 2 oberes Adreßbyte: | | |
| 3 Segmentnummer : | } | Sie beinhalten die höchsten 15 Bit der logischen Adresse des letzten Befehls vor der ersten Zugriffsverletzung |
| 4 oberes Adreßbyte: | | |
| 5 CPU-Betriebszustand: | | beschreibt den Betriebszustand der CPU während eines unzulässigen Zugriffs |
| | | N/S Normalmode/Systemmode |
| | | R/W Lesen/Schreiben |

Speicher Lesen mit MMU

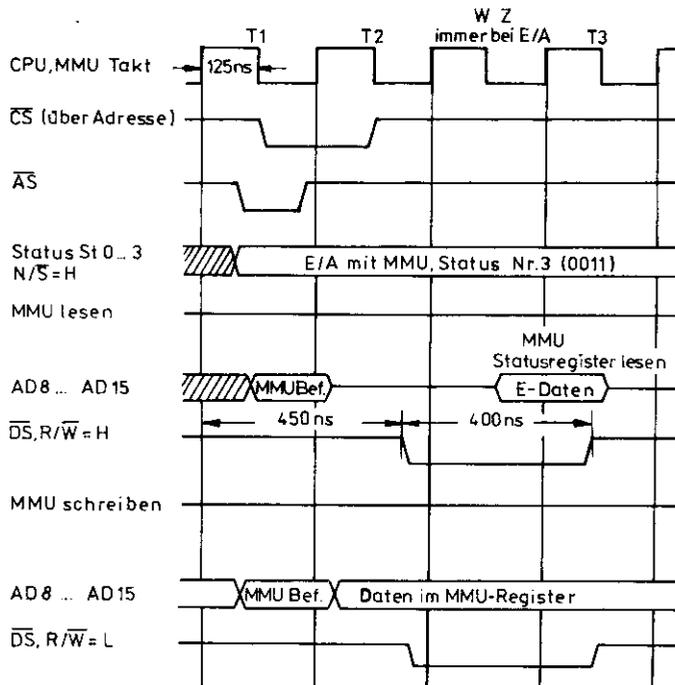


* bis 250 ns Zugriffszeit des Speichers, kein WZ für Speicher nötig (gegenüber 350 ns ohne MMU)

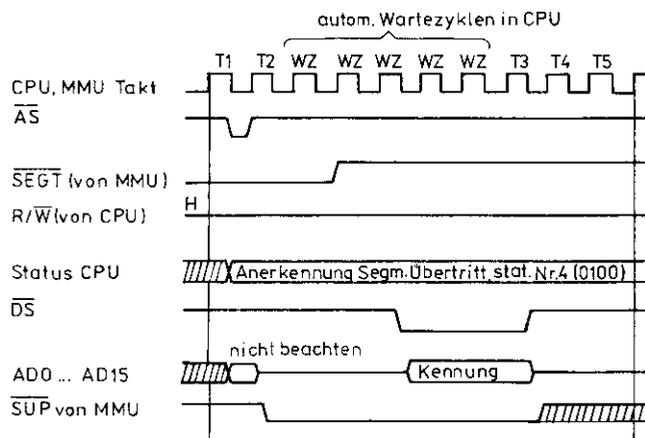
Speicher Schreiben mit MMU



CPU-Steuerfunktionen von und zur MMU

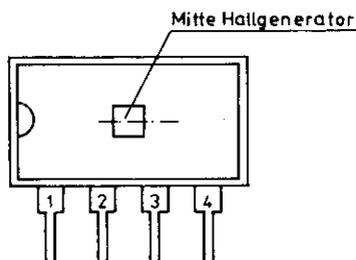


Segment-Trap-Anerkennungszyklus der CPU



Integrierter kontaktloser magnetisch betätigter Schalter (Hall-Effekt) mit Freigabeeingang und offenem Kollektorausgang

Wenn ein ausreichend großes Magnetfeld vorhanden ist ($B \geq B_E$) und ein H-Signal am Freigabeeingang anliegt, schaltet der offene Kollektorausgang von H nach L. Das Magnetfeld muß senkrecht mit dem Südpol auf die die Kennzeichnung enthaltende Fläche einwirken.



- 1 - Masse
- 2 - Ausgang
- 3 - Freigabeeingang
- 4 - Betriebsspannung

Betriebsbedingungen:

		min.	max.	
Betriebsspannung	U_S	4,75	5,25	V
Ausgangsstrom	I_{OL}		16	mA

Statische Kennwerte:

Stromaufnahme

$U_{IL3} = 0 \text{ V}, B \geq B_E$	I_{S1}	0,5	mA
$U_{IH3} = 5 \text{ V}, B \geq B_E$	I_{S2}	5	mA
$U_{IH3} = 5 \text{ V}, B \leq B_A$	I_{S3}	3	mA

Einschaltinduktion

$U_{IH3} = 2,4 \text{ V}$	$B_E^{1)}$	65	mT
---------------------------	------------	----	----

Ausschaltinduktion

$U_{IH3} = 2,4 \text{ V}$	$B_A^{2)}$	5	mT
---------------------------	------------	---	----

Eingangsströme

$U_{IL3} = 0,4 \text{ V}$	$-I_{IL3}$	5	μA
$U_{IH3} = 2,4 \text{ V}$	I_{IH3}	0,5	μA

Ausgangsstrom

$U_{OH2} = U_S, U_{IH3} = 2,4 \text{ V}, B \geq B_A$	I_{OH2}	10	μA
$U_{OH2} = U_S, U_{IL3} = 0,8 \text{ V}, B \geq B_E$	I_{OH2}	10	μA

Ausgangsspannung

$U_{IH3} = 2 \text{ V}, I_{OL2} = 16 \text{ mA}, B \geq B_E$	U_{OL2}	0,4	V
--	-----------	-----	---

1) B_E ist derjenige Wert der Induktion, bei der der Ausgangsstrom I_2 vom H-Pegel ($I_{OH2} \leq 10 \mu\text{A}$) auf den L-Pegel ($I_{OL2} = 16 \text{ mA}$) ansteigt.

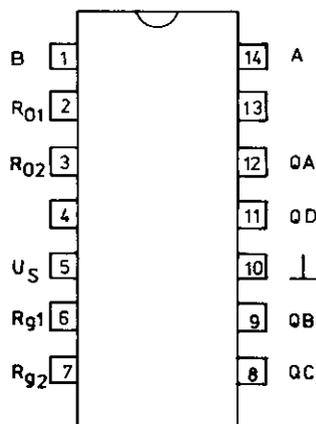
2) B_A ist derjenige Wert der Induktion, bei der der Ausgangsstrom I_2 vom L-Pegel ($I_{OL2} = 16 \text{ mA}$) auf den H-Pegel ($I_{OH2} \leq 10 \mu\text{A}$) absinkt.

Dezimalzähler

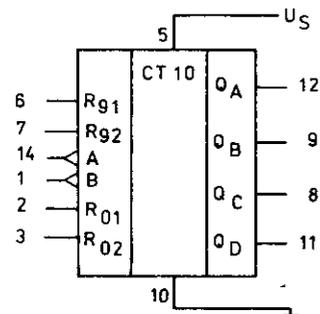
Er enthält 4 Master-Slave-FF. Das erste FF ist so geschaltet, daß am Ausgang Q_A die 2:1 geteilte Impulsfolge des Takteinganges A abgenommen werden kann.

Die weiteren FF mit dem Takteingang B und den Ausgängen Q_B , Q_C und Q_D realisieren einen 5:1 Teiler.

Die Ausgänge schalten bei der H-L-Flanke des jeweiligen Takteinganges. Um den vollen Zählumfang zu realisieren, wird der Ausgang Q_D mit Takteingang B verbunden. Wird der Ausgang Q_D mit Eingang A verbunden und Eingang B als Takteingang benutzt, steht am Ausgang Q_A eine symmetrische 10:1 geteilte Impulsfolge bereit.



Anschlußbelegung:



Logiktablelle:

Zählstand	Ausgänge			
	Q_D	Q_C	Q_B	Q_A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H

BCD-Zählung (Ausgang Q_A mit Eingang B verbunden)

Zählstand

Ausgänge

Funktionstabelle I-K-FF

	Q_D	Q_C	Q_B	Q_A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	H	L	L	L
6	H	L	L	H
7	H	L	H	L
8	H	L	H	H
9	H	H	L	L

t_n		$t_n + 1$
J	K	Q_n
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$

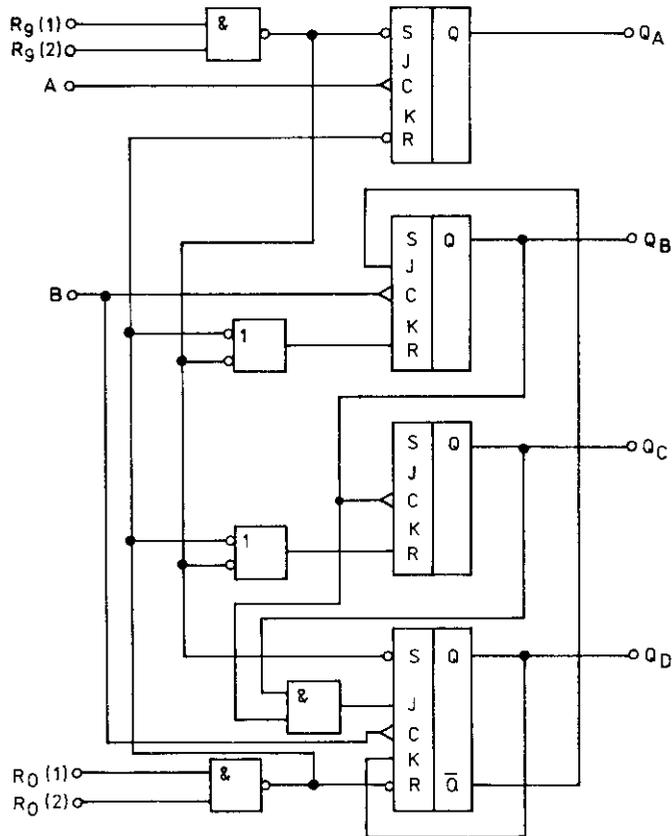
t_n = Zeit vor dem Taktimpuls

$t_n + 1$ = Zeit nach dem Taktimpuls

Symmetrische 10:1 Zählung

(Ausgang Q_D mit Eingang A verbunden)

Logische Schaltung:



Betriebsbedingungen:

		min.	max.	
Betriebsspannung	U_S	4,75	5,25	V
Umgebungstemperatur	ϑ_a	0	+ 70	°C
H-Ausgangsstrom	$- I_{OH}$		400	μA
L-Ausgangsstrom	I_{OL}		8	mA
Zählfrequenz	f_{max}			
Eingang A		0	32	MHz
Eingang B		0	16	MHz
Impulsbreite	t_w			
Eingang A		15		ns
Eingang B		30		ns
Eingang R		15		ns
Voreinstellzeit	t_{sn}			
Eingänge R		25		ns

B 556

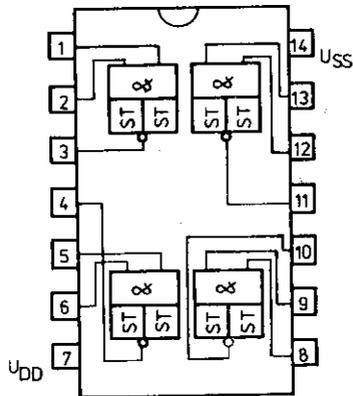
Taktgenerator zweifach

Siehe hierzu auf Seite 47 bei B 555 D1

Der B 556 enthält 2 B 555!

Anschlußbelegung





$$Q = \overline{E_1 \cdot E_2}$$

E_1	E_2	Q
1	2	3
6	5	4
8	9	10
13	12	11

Formelzeichen und Abkürzungen

B	Großsignalverstärkung des Transistors
C	Kapazität
D	1. Diode, 2. Vorbereitungseingang des D-Flipflops
N_L	Lastfaktor
N_{Le}	Eingangslastfaktor
N_{La}	Ausgangslastfaktor
f	Frequenz
Δf	Frequenzabweichung
G	Gatter
H	Potential Hoch (HIGH)
I_C	Kollektorstrom des Transistors
I_I	Eingangsstrom
I_S	Speisestromaufnahme
I_{SH}	Speisestromaufnahme bei U_{OH}
I_{SL}	Speisestromaufnahme bei U_{OL}

IS	Integrierte Schaltung
J	} Vorbereitungseingang des JK-Flipflops
K	
L	
L	Potential Tief (LOW)
P _S	Leistungsaufnahme
Q	Flipflopaustrag
\bar{Q}	Flipflopaustrag negiert
R	Rücksetzeingang (bei Flipflop)
S	1. Setzeingang (bei Flipflop) 2. Eingang für Stop-Signale bei Impulsgeneratoren 3. Summeneingang (bei Adder)
T	1. Periodendauer 2. Takteingang 3. Transistor
TK	Temperaturkoeffizient
t _D	Verzögerungszeit allgemein
t ^D HL	Verzögerungszeit der Flanke nach L (gemessen als Zeitabstand nach 10%iger Amplitudenänderung bezogen auf den Endwert)
t ^D LH	Verzögerungszeit der Flanke nach H (wie oben!)
t _H	Zeitdauer des H-Potentiale
t _L	Zeitdauer des L-Potentiale
t _I	Zeitdauer des Eingangsimpulses
t _O	Zeitdauer des Ausgangsimpulses
t _P	Signal-Verzögerungszeit
t ^{PHL}	Signal-Verzögerungszeit der Flanke nach L (gemessen bei 50% des Amplitudenwertes)
t ^{PLH}	Signal-Verzögerungszeit der Flanke nach H (gemessen bei 50% des Amplitudenwertes)
t ^{THL}	Abfallzeit des Signals
t ^{TLH}	Anstiegszeit des Signals
U	Spannung
U _{IN, L}	Eingangsspannung
U _{OH, L}	Ausgangsspannung
U _S	Speisespannung
U _T	Schwellspannung
X	Schaltungsanschluß
\bar{X}	Schaltungsanschluß negiert
Y	Schaltungsausgang
0	Obertrag
θ	Temperatur
θ _B	Betriebstemperatur
θ _S	Lagertemperatur

robotron

VEB Robotron
Buchungsmaschinenwerk
Karl-Marx-Stadt
Annaberger Straße 93
PSF 129
Karl-Marx-Stadt, DDR – 9010

Exporteur:

Robotron – Export/Import
Volkseigener
Außenhandelsbetrieb
der Deutschen
Demokratischen Republik
Allee der Kosmonauten 24
PSF 11
Berlin, DDR – 1040

Kv 1799/88 V 7 1 1639 N2