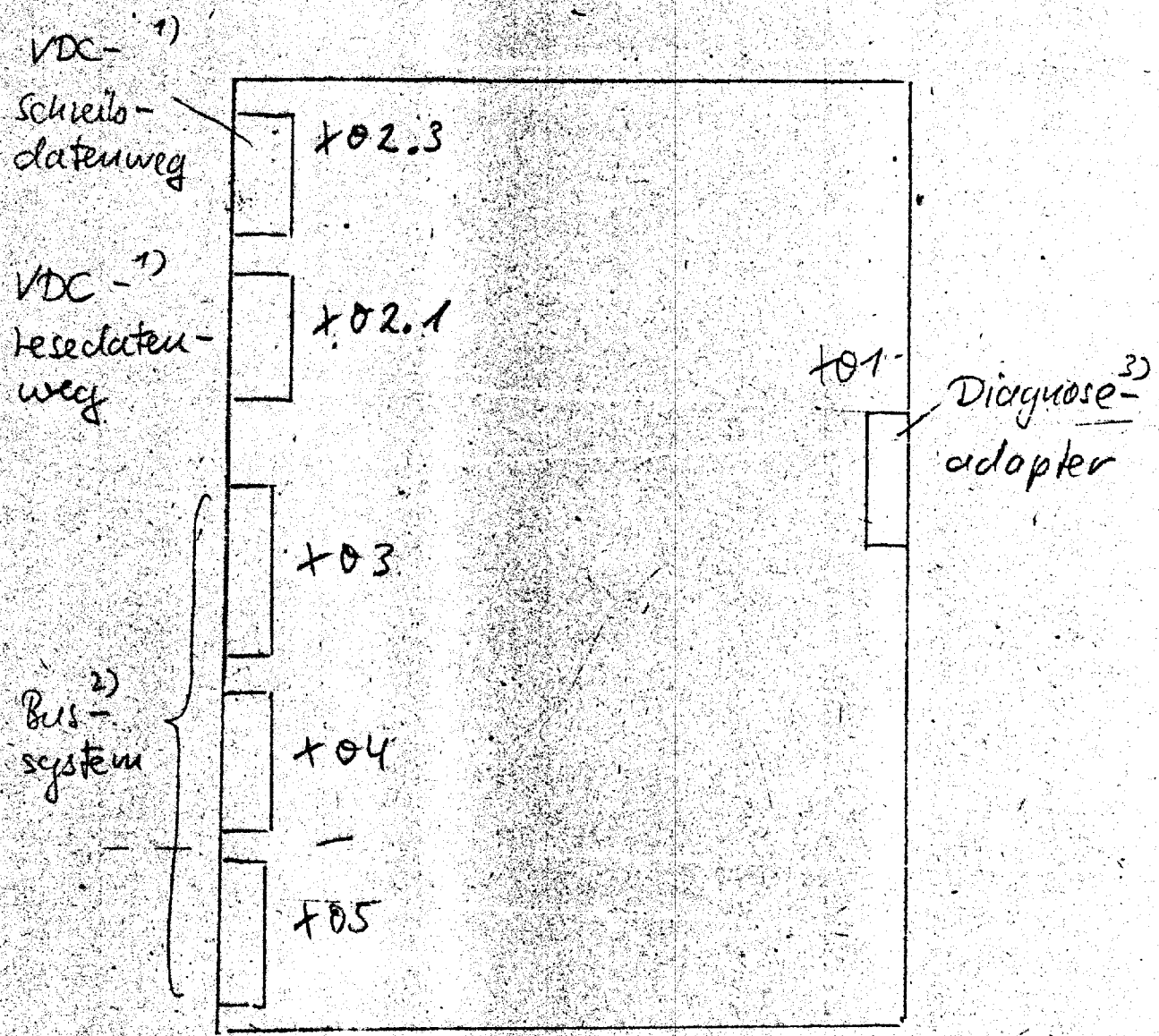


Steckverbinder



- 1) : Symbol
- 2) : Symbol
- 3) : Symbol

Übersicht VDC-Interface

Steckverbinder: X02.1, X02.3

Lfd. Nr.	Reihe A	Reihe C
1	INFO 0	VDC BUSY ¹⁾ ; VDC STROBE ²⁾
2	INFO 1	INFO 16
3	INFO 2	INFO 15
4	INFO 3	INFO 14
5	-	INFO 13
6	INFO 4	-
7	INFO 5	INFO 12
8	INFO 6	INFO 11
9	INFO 7	INFO 10
10	INFO 8	INFO 9

1) : bei X02.1

2) : bei X02.3

INFO : X02.1 : VDC READ INFO 16-0

X02.3 : VDC WRITE INFO 16-0

Übersicht über die

PROM-Schaltgleichungen

29-1

A15..12 BUFFERED CPU ADRS 15..12 15

MR MREQ
 RF RFSH
 CM CONTROL MODE
 NDZ /DIAG. ZERO FORCING

FAP FORCE EVEN PARITY

MA MA

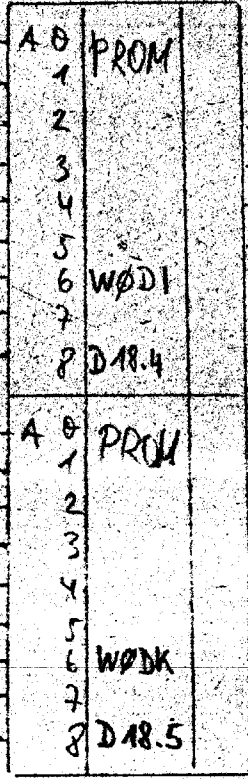
NDB /DIAG CONTROL MODE BREAK

RTB RAM TO BUFFER

14
 13
 12
 15
 14
 13

NO

MI



ROM ACCESS ENABLE
 SCRATCHPAD ACCESS ENABLE
 REGISTER ACCESS
 RAM ACCESS

ROC
 SAC
 REA
 RAC

WAIT. ENABLE
 CONTROL MODE ACCESS
 SHORT SEGMENT ACCESS
 CPU BREAK REQUEST

WTE
 CMA
 SSA
 CBR

5.D7

Schaltgleichungen PROM W00T

ROM ACCESS ENABLE

erlaubt Zugriffe zum ROM seitens der CPU (direkte sowie CONTROL MODE - Zugriffe). Im CONTROL MODE wird durch ENABLE ROM READ gesteuert, ob der Datenbus vom ROM belegt wird oder nicht.

|| $ROC = /A15 \& MR \& /RF \& ND2$

SCRATCHPAD ACCESS ENABLE

erlaubt Zugriffe zum Scratchpad seitens der CPU (direktes Lesen / Schreiben). Im CONTROL MODE gibt es keine Zugriffe.

|| $SAC = A15 \& /A14 \& /A13 \& MR \& /RF \& ND2 \& ~~ND1~~$

Hinweis: Scratchpad-Zugriffe finden auch parallel zu Registerzugriffen statt. Beim Lesen des DIAG IN-Registers (Diagnoseadapter) wird ENABLE SCRATCHPAD READ inaktiv, so daß das Register den Datenbus belegen kann.

REGISTER ACCESS

erlaubt Zugriffe zu den Registern (Lesen / Schreiben von der CPU, nicht im CONTROL MODE).

$$| \text{REA} = A_{15} \& /A_{14} \& /A_{13} \& /A_{12} \& \text{MR} \\ \& /RF \& \text{NDZ} \& \del{A_{11}}$$

RAM ACCESS

erlaubt Zugriffe zum RAM seitens der CPU (direktes Lesen / Schreiben; nicht im CONTROL MODE).

$$| \text{RAC} = (A_{15} \& /A_{14} \& A_{13} * A_{15} \& A_{14}) \& \\ \cdot \text{MR} \& /RF \& \text{NDZ} \& \del{A_{11}}$$

Schaltgleichungen PROM WODU

WAIT ENABLE

erlaubt Einschalten von WAIT (und damit Anforderung eines CPU-Zyklus) mit der Vorderflanke von MREQ.

$$| \text{WE} = ([A_{15} \& /A_{14} \& A_{13} * A_{15} \& A_{14}] \& \\ \text{NDZ} \& /CM) * (/A_{15} \& A_{14} \& \text{NDZ} \& \text{NDB}) \\ * (CM \& \text{NDB}) * (/A_{15} \& /A_{14} \& \text{NDZ} \& \\ \text{M1}) \del{\& \text{M1}}$$

Hinweis: WAIT wird eingeschaltet bei RAM-Zugriffen, bei CONTROL MODE sowie bei MA-
RAM-Zugriffen

CONTROL MODE ACCESS

erlaubt Einschalten von CONTROL MODE (bei M1) mit der Vorderflanke von MREQ.

$$\| \text{CMA} = /A_{15} \& A_{14} \& \text{NDZ}$$

SHORT SEGMENT ACCESS

steuert die Bits 2-0 des Segmentregisters zu den RAM-Adressen 13-11 durch.

Dies erfolgt bei entsprechenden gerichteten CPU-Zugriffen bzw. im CONTROL MODE bei RAM TO BUFFER-Datentransporten.

$$\| \text{SSA} = (A_{15} \& /A_{14} \& A_{13} \& /\text{CM}) * \text{RTB}$$

CPU BREAK REQUEST

fordert den sofortigen Abbruch eines CPU-Zyklus, wird wirksam bei M1-Zugriffen zum ROM.

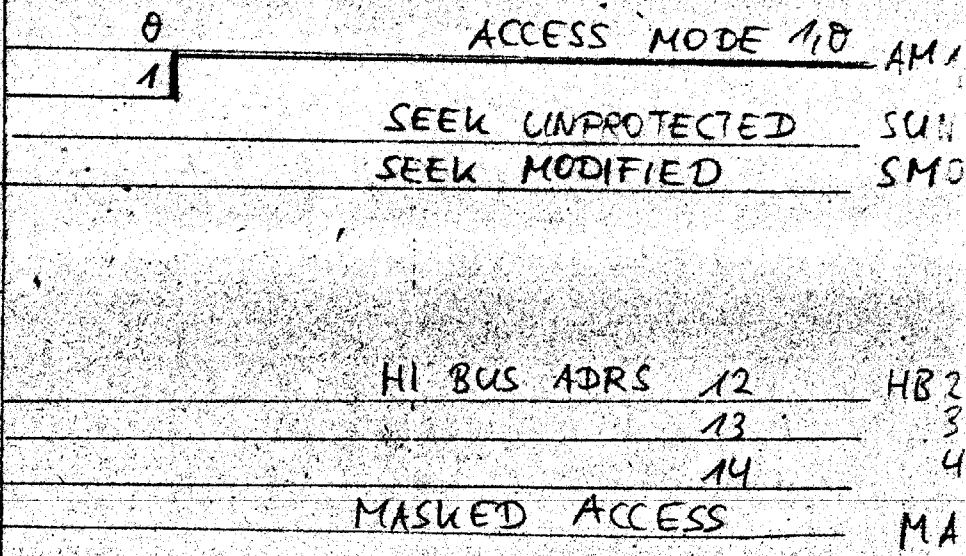
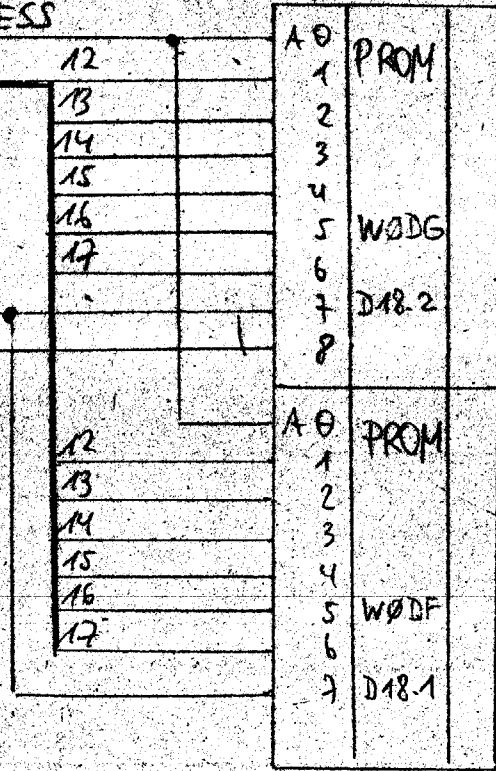
$$\| \text{CBR} = /A_{15} \& /A_{14} \& M1$$

SLA SYNCHRONIZED SLAVE ACCESS

IRV P23

BUS ADRS (RECEIVED) 17-12

INTERRUPT (RECEIVED)
PIO 2 PORT B3



S.D.3

Schaltgleichungen PROM WDOG

ACCESS MODE 1, 0

bestimmen die Auswahl eines Codefeldes im Wandlungs-PROM

<u>AM1</u>	<u>AM0</u>	
0	0	SCM bzw. MASKED ACCESS
0	1	ISO
1	0	IBM
1	1	IBM mit Wandlung LC → UC

$$AM0 = \left(\left(\neg IRV \& (B17 \& B16 \& B14) * \right. \right. \\ \left. \left. \neg B17 \& B16 \& \neg B14 \& (B13 \& B12) \right) * \right. \\ \left. IRV \& P23 \& (B17 * B16) \right) \& SLA$$

$$AM1 = \left(\neg IRV \& \neg B17 \& B16 \& (B14 * \neg B13) \right. \\ \left. * IRV \& (B17 * B16) \right) \& SLA$$

SEEK UNPROTECTED

wird wirksam für das Suchen ungeschützter Felder:

$$SU = B17 \& \neg B16 \& B14 \& \neg B13 \& SLA \& \neg IRV$$

SEEK MODIFIED

wird wirksame für das Suchen modifizierter Felder;

$$SM = B17 \& /B16 \& B14 \& B13 \& SLA \& /IRV$$

Schaltgleichungen PROM WODF

Höherwertige Bus-Adressen für RAM-Zugriffe

$$HB2 = (B12 \& / (B17 \& B16 \& /B14)) \& SLA \& /IRV$$

$$HB3 = (B13 \& / (/B17 \& B16 \& /B14 * B17 \& /B16 \& B14)) \& SLA \& /IRV$$

$$HB4 = B14 \& /B17 \& /B16 \& SLA \& /IRV$$

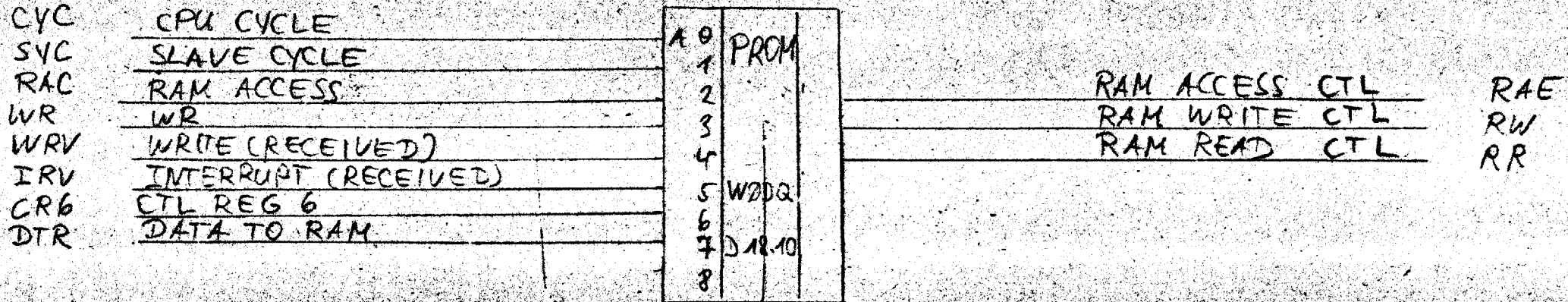
MASKED ACCESS

stellt Bus-Lesedatenpfad auf Suchzugriffe um;

$$MA = (/B17 \& B16 \& /B14 \& B13 * B17 \& /B16 \& B14) \& SLA \& /IRV$$

01

01



S.D4

69

Schaltgleichungen PROM WDDQ

RAM ACCESS ENABLE

erlaubt die Abgabe von Steuerimpulsen (RAS, CAS) zum RAM.

$$RAE = [CYC \& (RAC \times CR6)] \times (SYC \& /IRV)$$

RAM WRITE $(CR6 \& /DTR \times DTR \& WR)$ \circ hier nicht \rightarrow s. 892. $?$ $?$ $?$

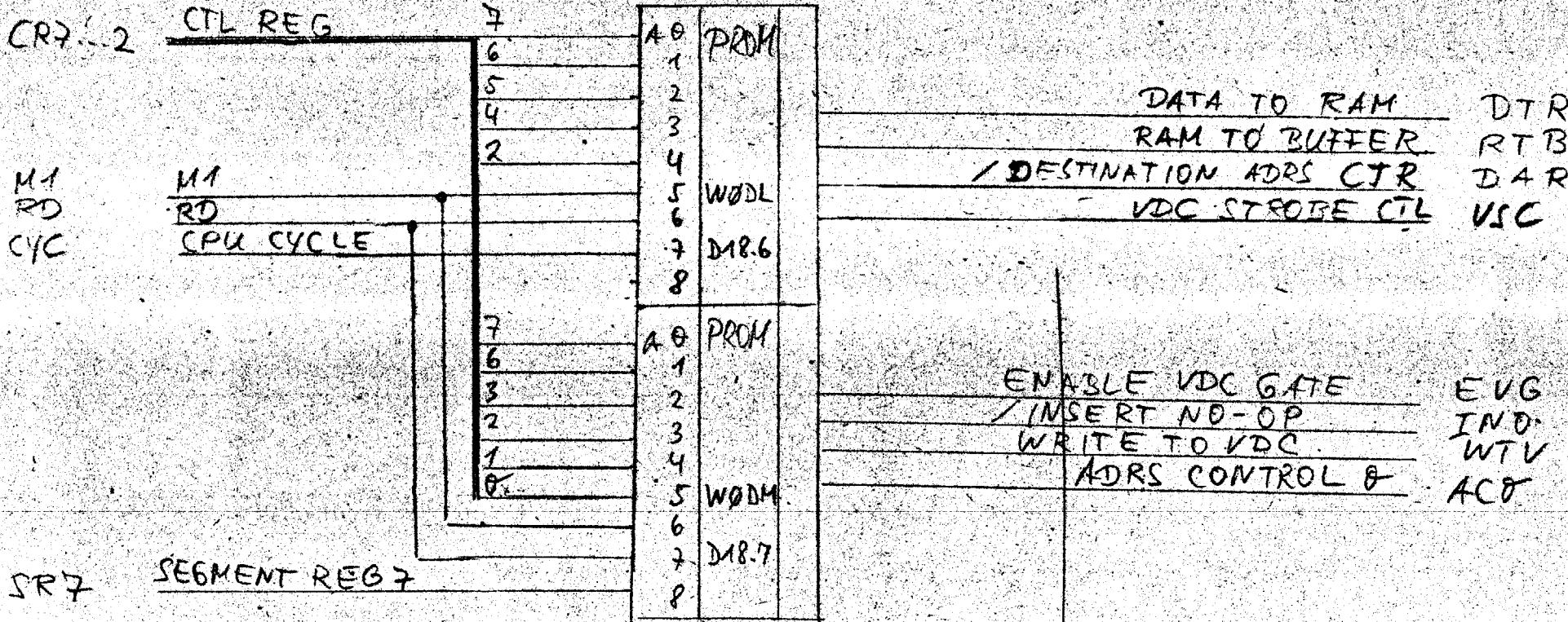
erlaubt die Abgabe des Schreibimpulses zum RAM.

$$RW = [CYC \& (RAC \& WR)] \times DTR \times (SYC \& WRV \& /IRV)$$

RAM READ

zeigt RAM-Lesen an, um die Auswertung der Paritätskontrolle zu erlauben.

$$RR = [CYC \& (RAC \& /WR)] \times [CR6 \& /DTR] \times (SYC \& /WRV \& /IRV)$$



S.D5

Schaltgleichungen PROM WDDL

DATA TO RAM

steuert Datentransport von der CPU zum RAM im CONTROL MODE

$$\| DTR = \neg CR7 \& CR6 \& \neg M1$$

RAM TO BUFFER

steuert Datentransport vom RAM zum VDC im CONTROL MODE

$$\| RTB = CR7 \& CR6$$

DESTINATION ADRS CTR

wenn aktiv (= 0) erfolgt Schreiben des ADRS CTR zum VDC. Dazu wird VDC WRITE INFO 16 = 1, und alle 16 Adressenbits der CPU werden zum VDC durchgesteuert.

$$\| DAR = \neg (CR7 \& CR6 \& \neg RD \& CR5 \& CR4)$$

VDC STROBE CTL

erlaubt Senden eines Strobe-Impulses zum VDC und das Laden der LeseRegister.

$$\| VSC = CR7 \cdot CYC \& (CR6 \& \neg M1)$$

Schaltgleichungen PROM WDM⁹³

ENABLE VDC GATE

erlaubt aufschalten der VDC-LeseRegister auf den Datenbus.

$$\| \text{EVG} = ((\neg \text{CR7} \& \neg \text{CR6}) * (\text{CR7} \& \neg \text{CR6})) \& \text{M1} \& \text{RD}$$

INSERT NO-OP

Schaltet NO-OP-Befehl (00H) auf Datenbus auf.

$$\| \text{INOP} = \neg (\text{CR7} \& \text{CR6} \& \text{CR2})$$

WRITE TO VDC

Zeigt auf dem Schreibdatenweg an, daß ein Schreibzugriff zum VDC erfolgt.

$$\| \text{WTV} = (\text{CR7} \& \text{CR6}) * (\text{CR7} \& \neg \text{CR6} \& \neg \text{RD})$$

ADRS CONTROL 0

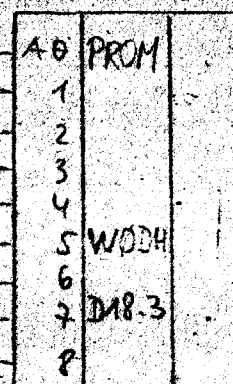
repräsentiert Bit 11 der VDC WRITE INFO.

Bei CONDITIONED SKIP darf das Bit nicht eingeschaltet werden, solange SEGMENT REG 7 (SKIP CONDITION) = 0 ist.

$$\| \text{AC0} = \text{CR0} \& (\neg [\text{7:6}] \vee \neg \text{CR3} \vee \text{CR1} \& \text{SR7})$$
$$[\text{CR7} \& \text{CR6}] * \neg \text{CR3} * \text{CR1} \& \text{SR7}.$$

WRV
 IRV
 SYC
 B17
 B16
 I1R
 I2R
 I3R

WRITE (RECEIVED)
 INTERRUPT (RECEIVED)
 SLAVE CYCLE
 BUS ADRS (RECEIVED) 17
 " " 16
 ITRP 1 RDY
 " 2 "
 " 3 "



ENABLE ITRP 1 STB
 " 2 "
 " 3 "

SLAVE CYCLE BREAK

I1S
 I2S
 I3S
 SCB

Schaltgleichungen PROM

35

W0DH

ENABLE ITRP 1 STB

Erlaubnissignal für Strobeimpuls zu
PIO 1 Port A

$$I1S = SYC \& WRV \& IRV \& /B17 \& B16 \& I1R$$

ENABLE ITRP 2 STB

Erlaubnissignal für Strobeimpuls zu
PIO 1 Port B

$$I2S = SYC \& WRV \& IRV \& B17 \& /B16 \& I2R$$

ENABLE ITRP 3 STB

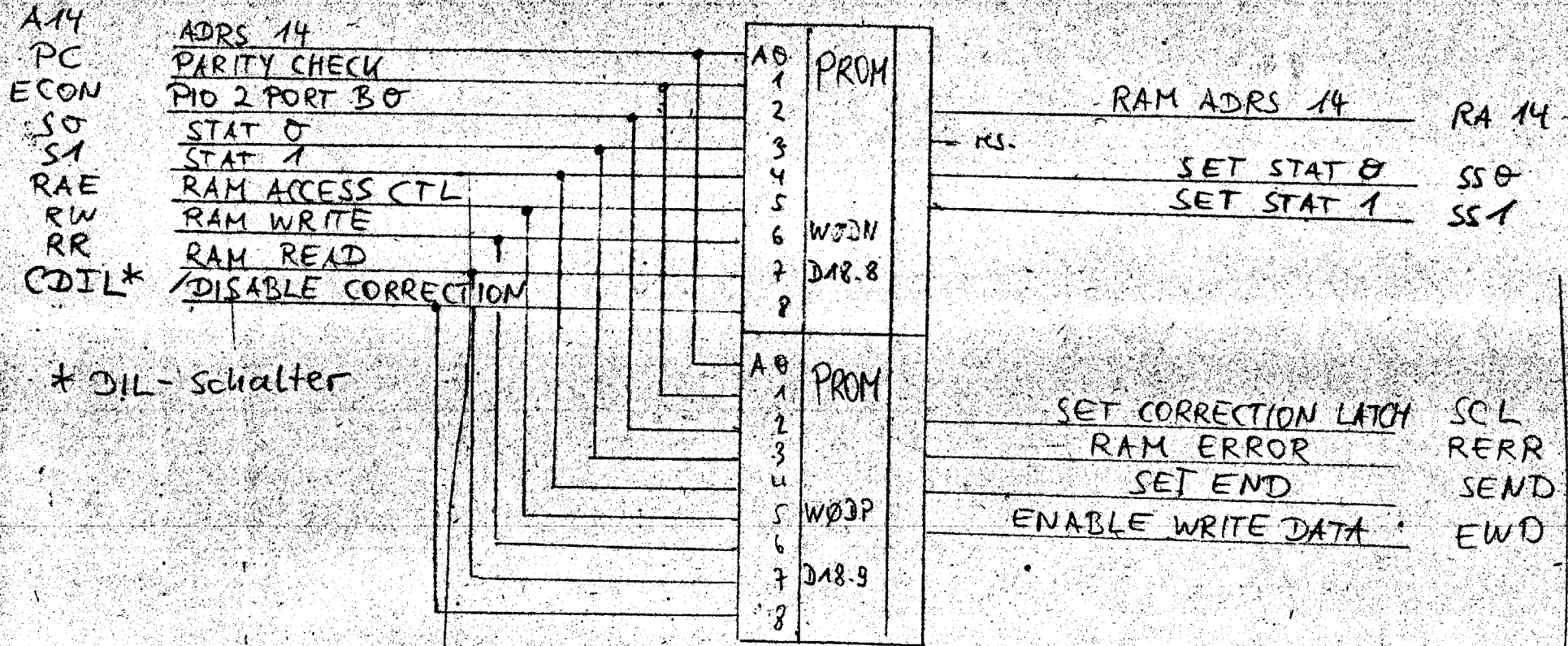
Erlaubnissignal für Strobeimpuls zu
PIO 2 Port A

$$I3S = SYC \& WRV \& IRV \& /B17 \& /B16 \& I3R$$

SLAVE CYCLE BREAK

Erlaubnissignal für Abbruch des Slave-
zyklus mit RELEASE

$$SCB = IRV \& WRV \& / (I1R \& I2R \& I3R)$$



Hilfsvariable in den Schaltgleichungen:

ECO: ENABLE CORRECTION

SOX; SECOND BYTE O.K.

S. EP

Schaltgleichungen PROM WDP

SET CORRECTION LATCH

schaltet CORRECTION LATCH ein

$$SCL = ([RAE \& A14 \& RR \& PC] \& SOK) \& \cancel{[/CDIL \& /ECON]} \& CDIL \& ECON$$

RAM ERROR

schaltet Fehlerflipflop ein

$$RERR = [[/CDIL \& /ECON] \& RAE \& RR \& PC] \& [ECO \& RRE \& S0 \& PC]$$

SET END

beendet den Zyklus.

$$SEMD = /ECO \& /CDIL \& \{ ECO \& [S1 \& [S0 \& (RW \& RR \& PC)] \& S0 \& S1 \& RRE \& PC] \}$$

ENABLE WRITE DATA

steuert Schreibdaten zum RAM durch (sonst: Lese-/Verkehrsdaten)

$$EWD = / [ECO \& S1]$$

Hilfsvariable ENABLE CORRECTION

ECO = CDIL & ECON & /A14 & RAE

Hilfsvariable SECOND BYTE O.K.

SOK = ECO & S0 & RR & /FC