

# DFÜ-Board

## Multi-BUS-I

Produktspezifikation  
der Flachbaugruppe DUEAI  
(S 26361 - D277)

				11.10.83					
				Ullrich					
						Flachbaugruppe DUEAI			
				Spezif.277					
				Siemens AG					
						S 26361 - D 277 - X - * - 59			
2		2.12.85		U1					

**Inhalt:**

<b>1</b>	<b><u>Uebersicht</u></b>	<b>3</b>
<b>2</b>	<b><u>Kennwerte</u></b>	
2.1	Eigenschaften der seriellen Schnittstellen	4
2.2	Elektrische Eigenschaften	4
<b>3</b>	<b><u>Funktionen</u></b>	
3.1	Blockschaltbild	5
3.2	Schaltungsbeschreibung	5
3.3	Adress-Tabelle des DUE-Prozessors	10
3.3.1	Memory-Adressen	10
3.3.2	I/O-Adressen	11
3.4	Interne Register	11
<b>4</b>	<b><u>Software-Interface</u></b>	
4.1	Multibus I/O-Adressierung	
4.1.1	BADR-Register	16
4.1.2	Lower- / Upper-Reg	17
4.2	EPROM-Boot	17
4.3	Aufbau des Kommunikationsbereichs	20
4.3.1	Status-Register	21
4.3.2	Befehls-Register	22
4.3.3	Firmwareversion	22
4.3.4	Speicherausbau	23
4.3.5	A.out Format des Warmladers	23
<b>5</b>	<b><u>Hardware-Interface</u></b>	
5.1	System-Schnittstelle	24
5.2	DUE-Schnittstellen	24

**Anhang**

1.	Bauderate-Einstellungen fuer den SCC (asynchron Mode)	27
2.	Schaltereinstellungen	28
3.	Diagnose-DUART	29
4.	Firmware	30

## 1. Uebersicht

Die Flbgr. DUEAI ist eine Standard Multibus-Flachbaugruppe. An der Systemschnittstelle gelten die funktionellen und elektrischen Eigenschaften, wie sie in der "MULTIBUS SPECIFICATION" beschrieben sind.

Die Flbgr. dient der Entlastung des Hauptprozessors der 9780 Systemeinheit durch eigenstaendiges Abarbeiten von ladbaren DUE-Prozeduren. Es besteht die Moeglichkeit, wahlweise zwei X21, zwei V24 oder eine V24 und eine X21 Schnittstelle zu steuern.

Denkbare Einsatzmoeglichkeiten sind TRANSDATA-Anschluss, SNA-Anbindung (SDLC) oder Datexanschluss (HDLC).

Es ist sowohl synchrone als auch asynchrone Betriebsart moeglich.

### Literaturhinweise

- [1] Normen fuer Informationsverarbeitung, Datenkommunikation
- [2] CCITT Empfehlungen der V- und der X-Serie (Decker Verlag)
- [3] Component Data Catalog Intel Jan. 82 (Nr. 210 298-001)
- [4] Rahmenspezifikation 9780 S26361 - L6 - X - \* - 59
- [5] Spezifikation des Zilog Bausteins 8530 SCC  
(Technical Manual, April 1982  
SCC Initialization - Application Note, Sept. 1982)
- [6] Multibus-Spezifikation der Fa. Intel (Rev. 4 6.82  
Order number 9800683-04)

## 2. Kennwerte

### 2.1 Eigenschaften der seriellen Schnittstellen

- Betriebsarten: synchron oder asynchron
- max. Bitrate 1 Mbit/sek (synchron od. asynchron)
- Zeichenrahmen 5-6-7-8 Bit/Zeichen
- Zeichensicherung: Paritaetsbit even od. odd moeglich.
- Blocksicherung per CRC (CCITT oder CRC-16) moeglich.
- Taktrueckgewinnung ueber Empfangsdaten moeglich
- NRZ-, NRZI-, Manchester-, FM-Decodierung moeglich
- local loop moeglich (Diagnose- bzw. Testmoeglichkeit)
- Sender oder Empfaenger sind per Befehl abschaltbar.
- Unterdrueckung von SYN im Empfangspuffer moeglich

### 2.2 Elektrische Kennwerte

Die Stromaufnahme betraegt:

+5V ca. 2,1 A
+12V ca. 0,1 A
-12V ca. 0,1 A

Die Toleranzwerte dieser Spannungen sind festgelegt

- bei +5V durch Regeln fuer TTL Technik
- bei +/- 12V durch Regeln der Schnittstelle V28

Fuer die elektrischen Kennwerte der beiden Schnittstellentypen wird auf die Definition der beiden Schnittstellen hingewiesen.

[1] und [2]

### 3. Funktionen

#### 3.1 Blockschaltbild

s. naechste Seite

#### 3.2 Schaltungsbeschreibung

Die Flbgr. arbeitet als "Slave" am Multibus und besteht im wesentlichen aus folgenden Komponenten

##### - Taktversorgung

Von zwei Quarzoszillatoren (16 MHz, 7.3728 MHz) werden alle notwendigen Taktfrequenzen abgeleitet.

16 MHz: Prozessorclock, Speichertiming

3.68 MHz: SCC-Systemclock

Mittels einer monostabilen Kippstufe wird der Refreshclock von 62.5 KHz erzeugt.

##### - DUE-Prozessorteil

Mikroprozessor 80188, interne Taktfrequenz 8 MHz mit integriertem Interrupt-Controller, integrierter CS-Logik sowie 3 Timer und 2 DMA-Kanaele, alle notwendigen Bussignale werden vom Prozessor erzeugt,

8 kB EPROM (Bootstrap-Lader)

##### - Dual Port Ram

Diskret aufgebauter Ram-Controller,

moegliche Speicherausbauten: 64 kB, 128 kB, 192 kB oder 256 kB, der Speicher ist parityueberwacht.

Zugriffe des DUE-Prozessors erfolgen ohne wait-states.

Das gesamte Ram oder Teilbereiche davon koennen ueber Multibus adressiert werden. Drei ueber Multibus ladbare Register legen dazu die Basisadresse des gesamten Rams, sowie die obere und untere Grenze des Dual port Fensters fest. Die Arbitration zwischen beiden Schnittstellen geschieht ueber die Prozessorsignale "HOLD" und HOLD-ACKN." Bei beiden Schnittstellen kann durch spezielle Lock-Befehle eine Busuebergabe verhindert werden (beim Multibus durch das Signal "LOCK/", bzw. beim DUE-Prozessor durch das LOCK-Praefix im Programmcode).

Die automatische Generierung des Multibus-Signals INH1/ verhindert Kollisionen, falls das Dual port Ram im Adressbereich des Hauptspeichers liegt. Die Datenbreite zum DUE-Prozessor wie auch zum Multibus betraegt 8 Bit.

### - Schnittstellenkanaele

#### V24:

Die Steuerleitung S1 (DEE betriebsbereit) kann fuer jeden Kanal ueber das jeweilige I/O-Reg. gesteuert werden.

Die Leitung M3 (ankommender Ruf) beider Kanaele kann einen Sammelinterrupt (INT 3) beim DUE-Prozessor ausloesen und ueber den Befehl "Read I/O-Latch" gelesen werden (s. 3.4).

Der SCC kann neben dem synchron Mode auch im asynchron Mode arbeiten. (Einstellungen im Anhang).

#### Ebene 1:

Den Bedingungen der Norm DIN 66259 wird entsprochen.

Als Treiberbausteine finden 75150, als Empfaenger 75154 Verwendung. Als Versorgungsspannung dient diesen Bausteinen +/- 12V.

Logische Verknuepfungen zwischen den einzelnen Signalen werden nicht vorgenommen.

Im einzelnen koennen folgende Schnittstellenleitungen kontrolliert bzw. gelesen werden:

Leitung	Steckerpin Trapez- buchse	geht nach / kommt von
D1 (Sendedaten)	2	SCC: TxD
D2 Empfangsdaten	3	SCC: RxD
T1 Sendetakt	24	SCC: TRxC
T2 "	15	"
T4 Empfangstakt	17	SCC: RTxC
S1 Betriebsbereit	20	I/O Mode-latch
S2 Sendeteil ein	4	SCC: RTS
S4 hohe Uebertr.- geschw. ein	23	liegt fest auf "ein"
M1 Betriebsbereit	6	SCC: DCD
M2 Sendebereit	5	SCC: CTS
M3 ankomm. Ruf	22	I/O Buffer

X21:

Ueber die Modemleitungen des SCCs koennen die einzelnen Leitungen gesetzt und ueberwacht werden.

Die Zuordnung ist dabei: (bei normaler Betriebsart (s. 3.4))

c steuern	--->	SCC: RTS
t senden	--->	SCC: TxD
i melden	--->	SCC: DSR <i>110(alld)</i>
r empfangen	--->	SCC: RxD
s Schritttakt	--->	SCC: TRxC und RTxC

Die Baugruppe bietet folgende Erkennungs- und Hilfsschaltungen:

- a)  $r=0$  und  $i=aus$  ueber 16 Schritttakte: Zustand 19 (Ausloesung) geht an den DCD-Eingang des SCC und kann einen Interrupt ausloesen.
- b)  $i=ein$  ueber 16 Schritttakte: geht an SCC: CTS und kann einen Interrupt ausloesen. (jeder Zustand von  $r$  und  $i$  muss mind. 16 Takte anliegen, damit er als Dauerzustand bewertet und entsprechend reagiert werden kann).
- c) ueber einen Multiplexer kann eingestellt werden, welcher Sender die Leitungen  $t$  und  $c$  steuert.  
Folgende 4 Moeglichkeiten sind einstellbar:
  1. Zustand 24 ("DEE gestoert",  $t=0$  und  $c=aus$ ) damit werden nach einem Reset der Baugruppe die Schnittstellenleitungen definiert gesetzt.
  2. normale Betriebsart: die Leitungen  $t$  und  $c$  koennen ueber den SCC (TxD, RTS) gesteuert werden.
  3. Zustand 14 ("DEE kontrolliert nicht bereit",  $t=01010\dots$  und  $c=aus$ ) die Signalfolge auf der Ltg.  $t$  wird dabei synchron zu  $s$  (Schritttakt) ausgegeben.
  4. Es kann auf vorher im I/O-Latch programmierte Zustaende von  $t$  und  $c$  geschaltet werden. (Dies erlaubt eine Neuinitialisierung des SCCs nach einem Verbindungsaufbau).
- d) die Zaehleingaenge der Timer 0 und 1 sind mit den Schritttakt  $s$  der Kanale A und B verbunden und koennen damit als Zeitueberwachung fuer die Leitungen  $c$  und  $t$  eingesetzt werden. (Min. 24 Takte bei Signalaenderungen auf den Leitungen).



Diagnosestecker:

Als Testhilfsmittel waehrend der Phase der Softwareintegration kann ueber einen Diagnosestecker ein Babyboard mit einem DUART angeschlossen werden.

(Damit Anschlussmoeglichkeit fuer ein Terminal oder einen Drucker, s. Anhang).

- Systemschnittstelle

Die Flbgr. arbeitet als "Slave" am Multibus. Alle benoetigte Bussignale sind gepuffert. Der Multibus selbst wird mit max. 4 LS Eingangslasten belastet.

- Interruptlogik

a. DUE-Proz. --> Hauptprozessor

Der DUE-Prozessor kann ueber eine entsprechende I/O-Adresse einen Interrupt beim Hauptprozessor ausloesen. Ueber einen Dip Fix Schalter kann eine der 8 Multibus-Intltg. sel. werden.

b. Flbgr. interne Interruptmoeglichkeiten

INT 0: SCC (wird der SCC im "vector mode" betrieben, muss INT 0 des 80188 auf "cascade mode" programmiert werden! INT 2 dient dann als INTA fuer den SCC) s.[3]

INT 1: Interrupt vom Hauptprozessor

INT 3: Sammelinterrupt fuer V24-Schnittstellensignale  
(M3, s. 3.4)

NMI: Parity-Fehler

Timer 0: Zeitueberwachung fuer Kanal A

Timer 1: " " B

Timer 2: universell einsetzbar

DMA 0: Interrupt vom DMA Controller 0

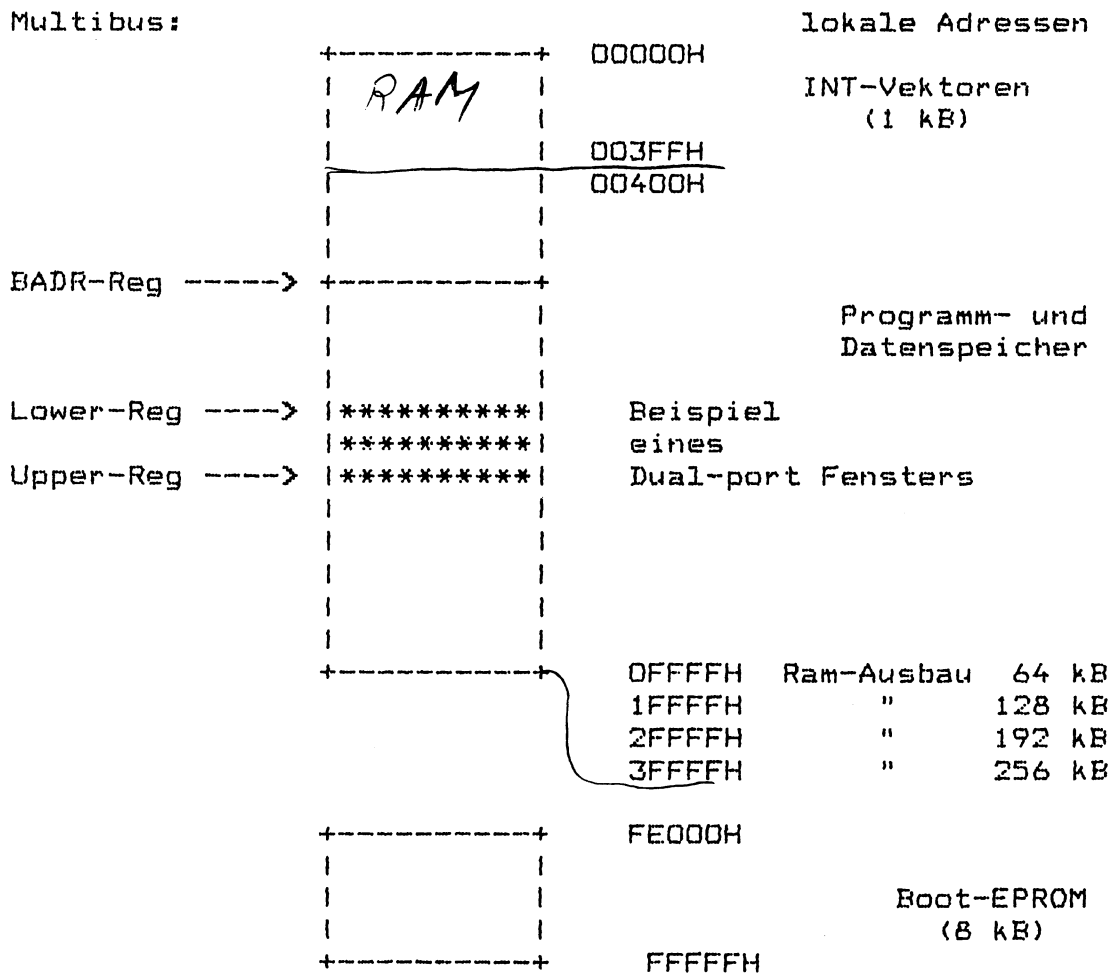
DMA 1: " 1

Der Hauptprozessor kann mittels Multibus I/O Adressen beim DUE-Prozessor einen INT 1 oder einen Reset auslösen. Reset bewirkt einen Rucksprung in den Boot-EPROM. Alle uebrigen Interrupt-Routinen muessen von der DUE-Prozedur festgelegt werden.

### 3.3 Adress-Tabelle des DUE-Prozessors

#### 3.3.1 Memory-Adressen

Die Rams werden selektiert mit LMCS (lower memory chip select) Das EPROM mit UMCS (upper memory chip select) [3]:



### 3.3.2 I/O - Adressen

Alle I/O Peripherals werden mit den PCS-Leitungen (peripheral chip select) angesprochen. Die Basisadresse wird innerhalb der Initialisierung auf 0 festgelegt (s. [3]).

PCS 0:	000H	Status/CMD Kanal B	\	
	001H	Daten-Reg "		SCC
	002H	Status/CMD Kanal A	>	8530
	003H	Daten-Reg "		
			/	
PCS 1:	080H	Reset INT 1 (Interrupt vom Multibus)		
PCS 2:	100H	Reset INT 3 (V24-Meldeltg. M3)		
PCS 3:	180H	Set Multibus-Interrupt		
PCS 4:	200H	Set Prozessor Mode-Reg.		
	210H	Set I/O Reg. Kanal A		
	220H	"	B	
	230H	Read I/O Latch		

ueber die I/O-Adressen 280H ... 28FH kann ein aufsteckbares Babyboard mit einem Diagnose-DUART angesprochen werden.

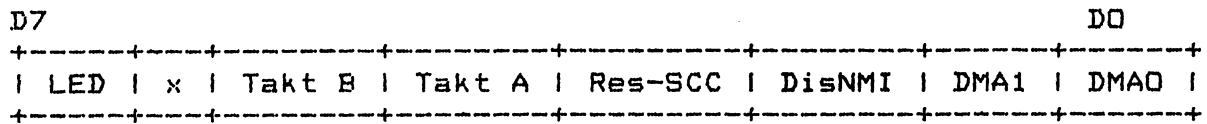
(s. Anhang)

### 3.4 Interne Register

Eine detaillierte Beschreibung der einzelnen Register des SCC (Zilog 8530) ist dem Datenbuch [5] zu entnehmen.

Nachfolgend sind deshalb nur die internen Register beschrieben.

a. Set Prozessor Mode-Register (200H)



Ueber die beiden Bits DMA 0 und DMA 1 kann selektiert werden, welcher der beiden Kanale A und B im DMA-Mode betrieben werden soll.

DMA 1	DMA 0	Betriebsart
0	0	Die SCC-Kanale A und B werden halbduplex betrieben (W/REQA ---> DREQ0 W/REQB ---> DREQ1)
0	1	SCC-Kanal A wird vollduplex betrieben (W/REQA ---> DREQ0 DTR/REQA ---> DREQ1)
1	0	SCC-Kanal B wird vollduplex betrieben (W/REQB ---> DREQ0 DTR/REQB ---> DREQ1)
1	1	DMA disabled
DisNMI: 0 ---> NMI enable		
1 ---> NMI disabled		
Res-SCC:0 ---> Reset fuer SCC		
1 ---> normaler Betrieb		

Takt A/B: ermoglicht bei den V24-Schnittstellen neben der ext. Taktzufuehrung (T2 und T4) auch eine Eigentaktung (T1).

0 ---> T2 und T4 werden dem SCC zugefuehrt

T2 ---> Pin 14 (TRxCA) fuer Kanal A

Pin 26 (TRxCB) " B

T4 ---> Pin 12 (RTxCA) " A

Pin 28 (RTxCB) " B

1 ---> die externe Taktzufuehrung fuer T2 wird disabled. Der Sendetakt kann (bei entsprechender Programmierung des SCCs) vom SCC selbst erzeugt und als T1 an die jeweilige V24 Schnittstelle geschaltet werden.

Pin 14 (TRxCA) ---> T1 fuer Kanal A

Pin 26 (TRxCB) ---> " B

(T4 wird davon nicht beruehrt)

! Achtung: Bevor die SCC-Pins 14 od. 26 als Taktausgang  
!  
!  
! programmiert werden, ist es zwingend erforderlich,  
!  
! das zuerst die Bits "Takt A/B" im Prozessor Mode-Register gesetzt werden.  
!

Die einzelnen Bits koennen ueber den Befehl "Read I/O Latch" rueckgelesen werden.

(Diese Betriebsart ist bei den Prototyp-Flbgr. (ersichtlich durch den Aufdruck W26361-D277-Z4-01-5 auf der Bauteileseite nicht moeglich!)

LED: Steuert die rote Diagnose-LED

0 ---> LED an

1 ---> LED aus

Das Prozessor Mode-Register kann nur beschrieben werden!

b. Set I/O Mode-Register Kanal A/B (210H / 220H)

D7									D0							
+	+	+	+	+	+	+	+	+	+							
	V24		ENA 1		ENA 0		t		c		MODE 1		MODE 0		S1	
+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+

Beide Register besitzen gleiches Datenformat

S1: steuert das Schnittstellensignal S1 der V24-Schnittstelle

0 ---> S1 aus

1 ---> S1 ein

MODE 0 MODE 1 steuert die X21-Schnittstelle

0 0 Zustand 24 (DEE gestoert, wird nach Reset ausgegeben)

0 1 normaler Betriebsmode (SCC steuert t und c)

1 0 latched mode: die im I/O Mode-Reg. fuer t und c angegebenen Werte werden ausgegeben.

t = 0/1 ---> X21: t = 0/1

c = 0/1 ---> X21: c = aus/ein

1 1 Zustand 14 (DEE kontrolliert nicht bereit, t = 01010..., c = aus) wird ausgegeben.

ENA 0: steuert bei V24 die Interruptmoeglichkeit von M3

(ankommender Ruf)

steuert bei X21 die Erkennungsschaltung fuer Zustand

19 (Ausloesung)

0 ---> disable und clear

1 ---> enable

ENA 1: steuert bei X21 die Erkennungsschaltung "i ein" ueber  
16 Schritttakte

0 ----> disable und clear

1 ----> enable

V24: steuert den Schnittstellen-Multiplexer V24 - X21

0 ----> X21 selektiert

1 ----> V24 "

Die I/O Mode-Register koennen nur beschrieben werden!

c. Read I/O Latch (230H)

D7				D0			
TAKT A	TAKT B	B.I	A.I	M3-B	M3-B	M3-A	M3-A
					INT-FF	INT-FF	

Die einzelnen Bits geben die Zustaende der Meldeleitungen M3  
der beiden V24-Schnittstellen wieder

M3-A: Leitung M3 fuer Kanal A

M3-B: " " B

0 ----> aus

1 ----> ein

M3-A INT-FF, M3-B INT-FF: gibt den Zustand des jeweiligen  
Interrupt Flip-flops an.

(Eine Zustandsaenderung von "aus" nach "ein" setzt das ent-  
sprechende Flip-flop)

0 ----> Grundzustand

1 ----> Leitungsaenderung erkannt

Ueber die I/O-Adressierung 100H kann dieser INT wieder rueck-  
gesetzt werden.

A.I / B.I: Geben den Zustand auf den X21-Schnittstellenleitungen

I (melden) fuer den jeweiligen Kanal A bzw. B an

0 ---> aus

1 ---> ein

TAKT A/B: Die ueber das Prozessor Mode-Reg. programmierten

Taktmodi koennen rueckgelesen werden

0 ---> extern Takt

1 ---> disable extern Takt

#### 4. Software-Interface

##### 4.1 Multibus I/O-Adressierung

Die Basisadresse kann mit einem 8 pol. Dip Fix in Schritten  
von <sup>256</sup>4 kByte eingestellt werden. (Einstellung: s. Anhang)

Nachfolgend sind die einzelnen Register mit ihren Offsetwerten  
zur eingestellten Basisadresse angegeben.

000H: Set NMI

001H: Enable Dual-port Fenster

002H: Reset Multibus Interrupt Flipflop

003H: Set INT 1 (Interrupt an DUE-Prozessor)

004H: Set Reset (bewirkt Ruecksprung in EPROM-Boot)

005H: Set Basisadresse (BADR-Reg)

006H: Set Upper-Register

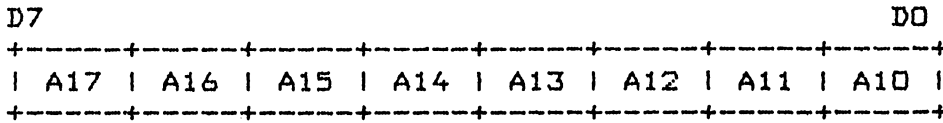
007H: Set Lower-Register

008H .... 0FFH sind reserviert und duerfen nicht angesprochen  
werden.

*Stimmt nicht mit Stromlauf MB-Inv. ADR!*



4.1.1 BADR-Register



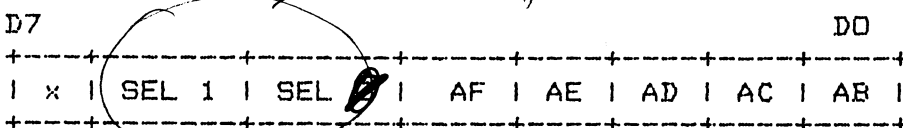
Im BADR-Reg wird die Multibus-Basisadresse des Dual port Segments angegeben. Mit A17 ... A10 koennen die Multibusadressen ADR 17/ ... ADR 10/ festgelegt werden.

Das Dual port Segment kann damit innerhalb eines Adressbereiches von 16 MB bei einer Schrittweite von 64 kB beliebig verschoben werden. Sollte der DUE-Prozessor in einer 9780-Systemeinheit ohne P2-Platter arbeiten, muessen die Bits (A14 ... A17) auf 0 gesetzt sein. Beispiele zur Programmierung des Dual port Segments:

Multibus-Adresse	Registerwert
000000H	00H
010000H	01H
020000H	02H
040000H	04H
080000H	08H
100000H	10H etc.

*aus Compendium keine Pullup*

4.1.2 Lower-Req. / Upper-Req.



*Stromlauf an Pins! In Data MB*  
*positive Logik*  
*(2) 1 sel0 -> in welchen Register? bzw Ausgerde } dann immer richtig*  
*auf beide*

Mit den Registern koennen die Grenzen des Dual port Fensters innerhalb des Dual port Segments festgelegt werden. AB ... AF entsprechen dabei den Multibus-Adressbits ADR B/ bis ADR F/.

Mit SEL 0 und SEL 1 kann festgelegt werden, auf welche Rambank des DUE-Prozessors zugegriffen werden soll.

SEL 1	SEL 0	Rambank	lokaler Adressbereich	
0	0	0	00000H ... 0FFFFH	<i>untersten 64k 1.</i>
0	1	1	10000H ... 10000H	2.
1	0	2	20000H ... 2FFFFH	3.
1	1	3	30000H ... 3FFFFH	4. <i>Bauk</i>

Ein Zugriff in das Ram wird gestattet bei Erfuellen der folgenden beiden Bedingungen:

1. [Lower-Reg] <= Multibus-Adr. <= [Upper-Reg]  
[AB ... AF] AB ... AF [AB ... AF]
2. [BADR-Reg] <= Multibus-Adr. < [BADR-Reg] + 64 kB

Die minimale Fensterbreite bzw. Schrittweite ist 2kB:

$$[\text{Lower-Reg}] = [\text{Upper-Reg}]$$

Die maximale Fensterbreite ist 64 kB:

[Lower-Reg] = 00H,	[Upper-Reg] = 1FH	(Rambank 0)
[ " ] = 20H,	[ " ] = 3FH	" 1
[ " ] = 40H,	[ " ] = 5FH	" 2
[ " ] = 60H,	[ " ] = 7FH	" 3

Das Fenster kann geschlossen werden, wenn:

$$[\text{Lower-Reg}] > [\text{Upper-Reg}]$$

ⓑ
ⓑ

Beispiele zur Programmierung der Fenstergrenzen

Offset zur Segment-Adr	Reg. Wert			
	Rambank 0	Rambank 1	Rambank 2	Rambank 3
00000H	00H	20H	40H	60H
00800H	01H	21H	41H	61H
01000H	02H	22H	42H	62H
01800H	03H	23H	43H	63H
04000H	08H	28H	48H	68H
06000H	0CH	2CH	4CH	6CH
0A000H	14H	34H	54H	74H
0F800H	1FH	3FH	5FH	7F

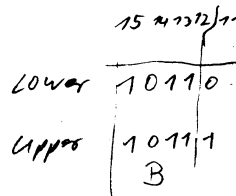
Beispiel: Das Dual port Fenster soll unter den Multibusadressen 0FB000H bis 0FBFFFH initialisiert werden.

Selektiert werden soll die lokale Rambank 0 des DUE-Prozessors:

[BADR-Reg] : 0FH

[Lower-Reg]: 16H

[Upper-Reg]: 17H



Nachdem die einzelnen Register geladen worden sind, muss mit einem Schreibzyklus auf die I/O-Adresse 01H das Dual port Fenster enabled werden (s. 4.1)

#### 4.2 EPROM-Boot

Nch einem Hardware- bzw. Software-Reset laeuft automatisch der EPROM-Boot an. Er initialisiert den Kontroll-Block des uP 80188 und fuehrt einen ROM-Test, einen Ram-Test und einen SCC-Test

durch. Das Testergebnis, der Ramausbau und die Firmwareversion werden in den jeweiligen Registern angezeigt.

Die Dauer der Initialisierung ist abhaengig vom Speicherausbau und dauert bei 256 kB Ram ca. 6s.

Danach wird auf einen Interrupt vom System gewartet, der dann durch einen Quittungsinterrupt seitens des DUE-Prozessors beantwortet wird. (Der genaue Ablauf der Selbsttests ist im Anhang angegeben)

Der Hauptprozessor oeffnet ab der lokalen Adresse B000H ein Fenster im Ram, laedt den Warmlader, setzt im Befehls-Register das Bit "AOUTSTART" und erzeugt einen Interrupt.

*also keine MB-Verruegung*

*Welche Bank? kann nur Bank 4 sein*

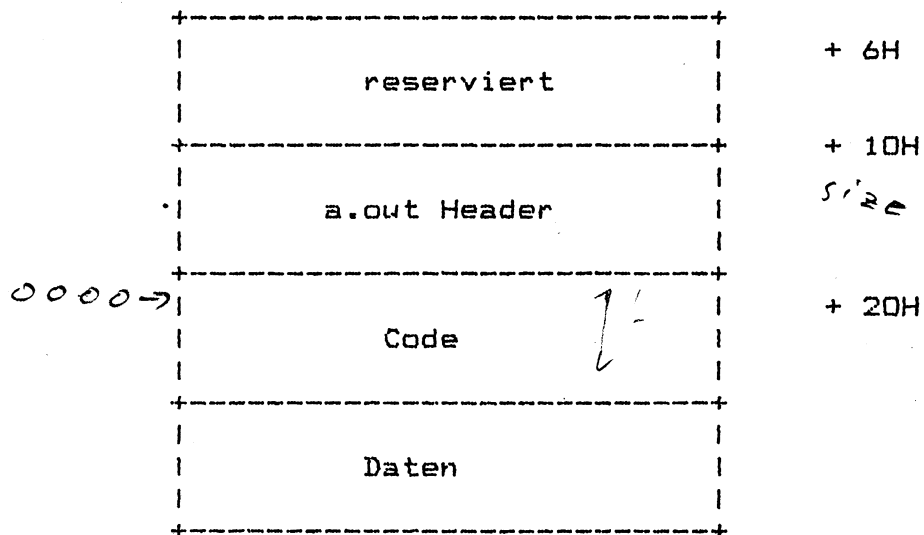
Anhand des a.out Headers ermittelt sich der DUE-Prozessor die Startadresse, setzt die Segmentregister DS, SS und CS auf den Codeanfang (Fensteradresse + 20H) bzw. ES auf das Fenster selbst und startet den Warmlader.

#### 4.3 Aufbau des Kommunikationsbereich

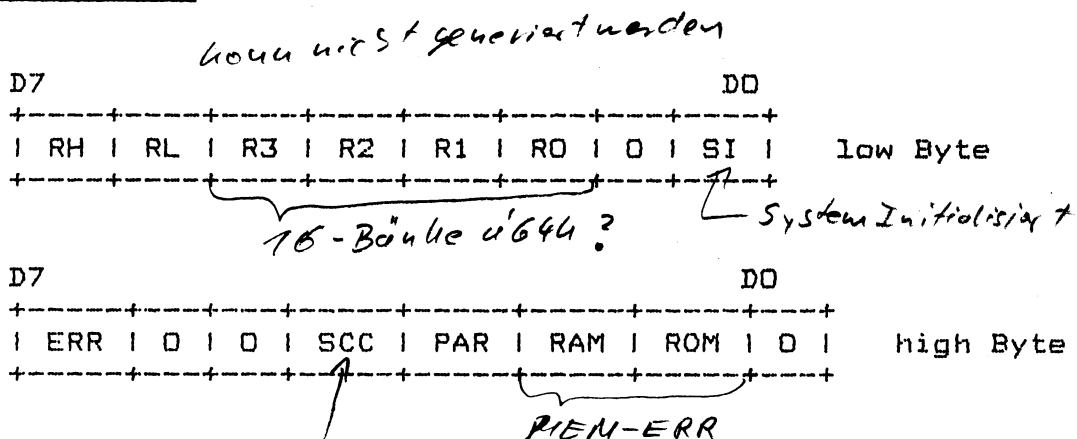
Der Koomunikationsbereich liegt innerhalb des Dual port Fensters und befindet sich vor Prozeduruebernahme ab der Adresse B000H (lokale Adresse; entspricht den Multibusadresse FB000H. Die obere Grenze des Fensters kann beliebig gewaehlt werden.

*so eingestellt*

-----+-----		
Status-Reg. (low Byte)		Fensteradr.
-----+-----		
Status-Reg. (high Byte)		+ 1H
-----+-----		
→ Befehls-Register		+ 2H
-----+-----		
reserviert		+ 3H
-----+-----		
Firmware-Version		+ 4H
-----+-----		
Speicherausbau		+ 5H
-----+-----		



#### 4.3.1 Statusregister



ROM: ROM-Fehler

*Lokal  
I/O-ERR*

kennzeichnet einen waehrend des Romtest erkannten Fehler

RO ... R3, RAM, PAR: RAM-Fehler, Parity-Fehler

kenzeichnet einen waehrend des Ramtest erkannten Fehler

Der Index gibt dabei die fehlerhafte Ram-Bank

RL: Der Ram-Fehler liegt im Low-Ram (Bit 0 ... 3)

RH: " High-Ram (Bit 4 ... 7)

SCC: kennzeichnet einen waehrend des SCC-Test erkannten Fehler

ERR: Error, wird gesetzt, wenn einer der vorher aufgefuehrten Fehler erkannt wurde

SI: System initialisiert  
kennzeichnet den Abschluss der Initialisierungs- und Test-routinen.  
Sollte waehrend der Testprogramme einer der aufgefuehrten Fehler erkannt worden sein, wird der Warmlader nicht gestartet.

### 4.3.2 Befehlsregister

(Fensteradr. + 2H)

```
D7                                D1                                D0
+---+---+---+---+---+---+---+---+---+
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | AOUTSTART |
+---+---+---+---+---+---+---+---+
```

AOUTSTART: signalisiert dem DUE-Prozessor, dass der Warmlader vom Hauptprozessor ins Ram geladen wurde und gestartet werden kann.

Dem Befehl muss ein INT 1 folgen.

### 4.3.3 Firmwareversion

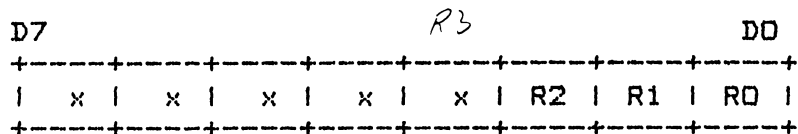
(Fensteradr. + 4H)

```
D7                                D0
+---+---+---+---+---+---+---+---+---+
| S7 | S6 | S5 | S4 | S3 | S2 | S1 | S0 |
+---+---+---+---+---+---+---+---+
```

Die Bits S0 bis S7 geben den auf der Baugruppe befindliche Firmwareversion an.

#### 4.3.4 Speicherausbau

(Fensteradr. + 5H)



Die Bits R0 bis R3 geben den auf der Baugruppe befindlichen Speicherausbau an

R2	R1	R0	
0	0	1	64 kB
0	1	0	128 kB
0	1	1	192 kB
1	0	0	256 kB
<i>1 0</i>	<i>0</i>	<i>0</i>	<i>512 kByte</i>

#### 4.3.5 A.out Format des Warmladers

*A.out - Header*

Der Header des Warmladers beginnt bei Fensteradresse + 10H und hat folgende Form:

```

struct header {
int      a_magic      /* magic number          */
unsigned a_text      /* Laenge des Code-Segments
unsigned a_data      /* Laenge der init. Daten
unsigned a_bss       /* Laenge der nicht init. Daten
unsigned a_syms      /* hier 0                */
unsigned a_entry     /* Startadresse rel. zu a_text
unsigned a_unused    /* hier 0                */
unsigned a_flag      /* hier immer 1          */
};
    
```

## 5 Hardware-Interface

### 5.1 Systemschnittstelle

Die Systemschnittstelle ist multibuskompatibel.

Die Flbgr. selbst arbeitet als "Slave" am Bus.

Alle Bussignale, sowie das Bus-Timing sind in der Multibus-Spezifikation (Intel) beschrieben.

### 5.2 DUE-Schnittstellen

Die Verbindung zwischen der Flbgr. und dem Anschlussfeld der 9780 Systemeinheit geschieht ueber spezielle Flachbandleitungen.

Die Zuordnung der verschiedenen Stecker zu den einzelnen Kanalen ist dabei:

V24 Kanal A ---> P4 Stecker  
V24 Kanal B ---> P6 Stecker  
X21 Kanal A ---> P5 Stecker  
X21 Kanal B ---> P7 Stecker

Nachfolgend ist die Schnittstellenbelegung der Flachbaugruppensteckverbinder und der zugehoerigen Trapez-Buchsen im Anschlussfeld aufgefuehrt.



Schnittstelle V24

Steck-Mechanik: 22 pol. cod. Modulstecker

Signaldefinition: gem. DIN 66020 Teil 1

Modul	AMP	Signal-Name	Bemerkungen
01			codiert
02	02	D1 TxD 103	Sendedaten
03	03	D2 RxD 104	Empfangsdaten
04	04	S2 RTS 105	Sendeteil einschalten
05	05	M2 CTS 106	Sendebereitschaft DUEE
06	06	M1 DSR 107	Betriebsbereitschaft DUEE
07	07	E2 GND 102	Signalerde
08	-	-	-
09	23	S4 111	hohe Ueb.-geschw. ein
10	24	T1 113	
11	-	-	
12	-	-	
13	-	-	
14	15	T2 114	Sendeschriftakt
15	-	-	
16	17	T4 115	Empfangsschriftakt
17	-	-	
18	-	-	
19	20	S1.2 DTR 108/2	DEE betriebsbereit
20	22	M3 125	ankommender Ruf
21	-	-	codiert
22	-	-	

Beide V24-Schnittstellen haben die gleiche Steckerbelegung

Alle nicht aufgefuehrten V24-Schnittstellensignale werden nicht verwendet bzw. nicht gesteuert.

Schnittstelle X21

Steck-Mechanik: 14 pol. cod. Modulstecker

Signaldefinition: gem. DIN 66020 Teil 2

Beide X21 - Schnittstellen haben identische Steckerbelegungen

*15 pol  
Cannon  
Belegung*  
↓

Modul	AMP	Signal-Name	Bemerkungen
01	-		codiert
08	08	G	Signalerde, verb. mit 0 V
06	02	T(A)	Senden
14	09	T(B)	"
12	03	C(A)	Steuern
05	10	C(B)	"
11	04	R(A)	Empfangen
04	11	R(B)	"
10	05	I(A)	melden
03	12	I(B)	"
09	06	S(A)	Schrittakt
02	13	S(B)	"

Anhang

Bauderate-Einstellungen fuer den SCC

(fuer asynchrone Betriebsart)

Voreinstellung SCC:

- WR-Reg 4:           x16 clock
- WR-Reg 11:        Transmit-clock = BR-Generator output  
                  Receive-clock =                   "
- WR-Reg 14:        BR-Generator enable  
                  BR-Generator source = SCC-PCLK (3.6864 MHz)
- WR-Reg 12:                            Bauderate-Einstellung
- WR-Reg 13:

Unten aufgefuehrte Tabelle gibt die Werte fuer die gaengigsten Uebertragungsgeschwindigkeiten an.

Daneben laesst sich der Wert fuer jede beliebige Bauderate nach folgender Formel berechnen (x16 clock!)

$$x = \frac{115313}{\text{Bauderate}} - 2$$

Dabei ist x der Wert fuer WR-Reg 12/13; die Bauderate wird in Bit/s angegeben.

Count value	Bauderate [Bit/s]
1	38400
4	19200
10	9600
22	4800
46	2400
94	1200
190	600
382	300

## 2.1 Schaltereinstellungen (Vorserien- und Serienflbgr.)

### a. Multibus I/O-Adresse

Einstellung mit den Schaltern S2 ... S9

S 9   S 8   S 7   S 6   S 5   S 4   S 3   S 2	
AF/   AE/   AD/   AC/   AB/   AA/   A9/   A8/	Multibusadressen
op   op   op   clo   clo   op   op   clo	= 1900H ... 19FFH

Bei Auslieferung der Baugruppe sind S2, S5 und S6 geschlossen.  
Die Baugruppe ist damit auf die I/O-Adresse 1900H eingestellt.

### b. Multibusinterrupt

Einstellung mit den Schaltern S10 ... S17

S17   S16   S15   S14   S13   S12   S11   S10	
7   6   5   4   3   2   1   0	Multibus-
op   op   clo   op   op   op   op   op	INT-Ltgen.

Bei Auslieferung der Baugruppe ist S16 geschlossen und  
erzeugt damit einen INT 5/.

Die Schalter S1 und S18 dienen ausschliesslich prueftechn.  
Zwecken und muessen im Normalbetrieb immer geschlossen sein.

## 2.2 Schaltereinstellungen (Prototypen)

(Prototypen sind erkenntlich am Leiterplattenaufdruck  
"W26361-D277-Z4-01-5" auf der Bauteileseite.)

### a. Multibus I/O-Adresse

Einstellung mit den Schaltern S9 ... S16

S16   S15   S14   S13   S12   S11   S10   S 9	
AF/   AE/   AD/   AC/   AB/   AA/   A9/   A8/	Multibusadressen
op   op   op   clo   clo   op   op   clo	= 1900H ... 19FFH

### b. Multibusinterrupt

Einstellung mit den Schaltern S 1 ... S 8

S 8   S 7   S 6   S 5   S 4   S 3   S 2   S 1	
7   6   5   4   3   2   1   0	Multibus-
op   op   clo   op   op   op   op   op	INT-Ltgen.

3. Diagnose-DUART = *A Baud-Rate*

Zum Testen von geladenen Programmen kann ueber den Diagnosestecker P3 ein Babyboard mit einem DUART 2681 (Signetics) angeschlossen werden.

Am seriellen Ausgang des Babyboards koennen Geraete mit SS 97 angeschlossen werden.

Unter den folgenden I/O-Adressen koennen die einzelnen Register des DUARTs angesprochen werden.

I/O-Adr.	Lesen	Schreiben
280H	Mode-Reg 1/2 Kanal A	Mode-Reg 1/2 Kanal A
281H	Status-Reg. A	Clock-sel. A
282H	----	Command-Reg. A
283H	Receive-Reg. A	Transmit-Reg A
284H	----	Aux. control Reg

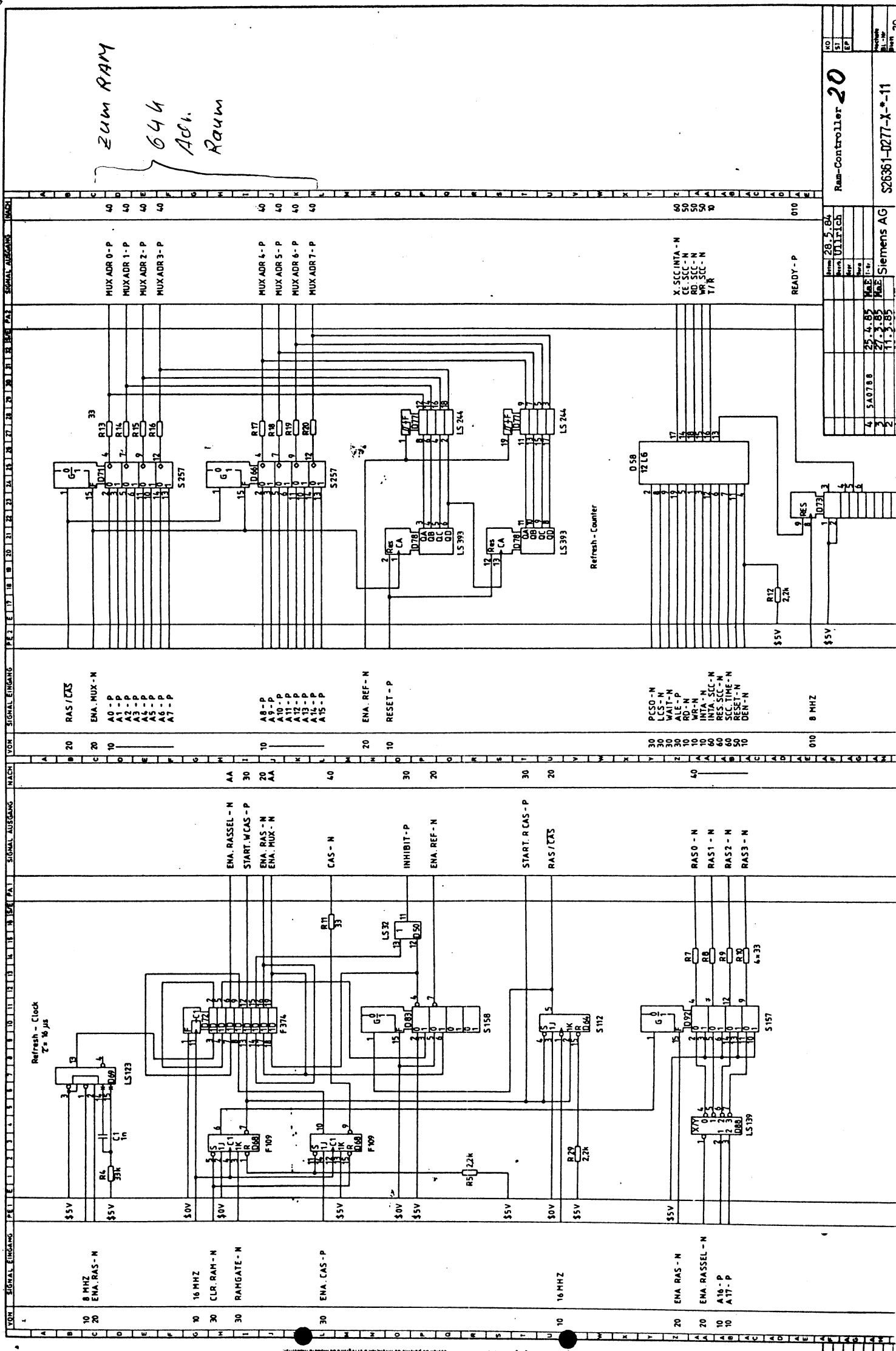
Die nicht aufgefuehrten Register werden nicht benoetigt bzw. von der Hardware nicht unterstuetzt.

Der DUART wird mit einem externen Clock von 3.6864 MHz betrieben.

Die Spannungsversorgung des Boards geschieht ueber den P3 Stecker vom DUE-Prozessor aus.



ZUM RAM  
64k  
Adv.  
Rauw



NO	REV	DATE	BY
1			
2			
3			
4			

REV	DATE	BY	REASON
1			
2			
3			
4			

REV	DATE	BY	REASON
1			
2			
3			
4			

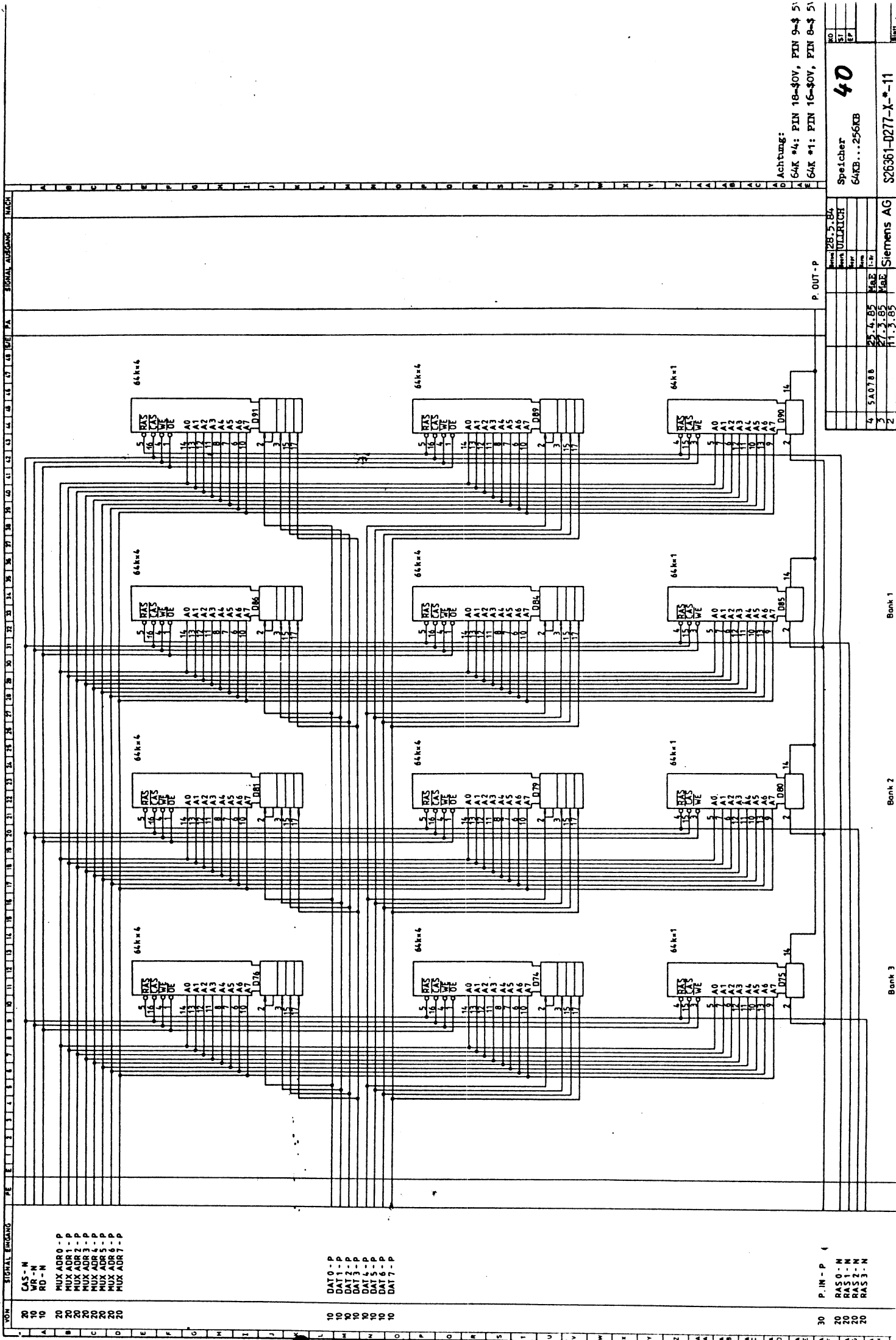
REV	DATE	BY	REASON
1			
2			
3			
4			

REV	DATE	BY	REASON
1			
2			
3			
4			

REV	DATE	BY	REASON
1			
2			
3			
4			







PIN SIGNAL EINGANG PE E 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100  
 SIGNAL AUSGANG NACH  
 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z AA AB AC AD AE AF AG AH AI AJ AK AL AM AN AO AP AQ AR AS AT AU AV AW AX AY AZ BA BB BC BD BE BF BG BH BI BJ BK BL BM BN BO BP BQ BR BS BT BU BV BW BX BY BZ CA CB CC CD CE CF CG CH CI CJ CK CL CM CN CO CP CQ CR CS CT CU CV CW CX CY CZ DA DB DC DD DE DF DG DH DI DJ DK DL DM DN DO DP DQ DR DS DT DU DV DW DX DY DZ EA EB EC ED EE EF EG EH EI EJ EK EL EM EN EO EP EQ ER ES ET EU EV EW EX EY EZ FA FB FC FD FE FF FG FH FI FJ FK FL FM FN FO FP FQ FR FS FT FU FV FW FX FY FZ GA GB GC GD GE GF GG GH GI GJ GK GL GM GN GO GP GQ GR GS GT GU GV GW GX GY GZ HA HB HC HD HE HF HG HH HI HJ HK HL HM HN HO HP HQ HR HS HT HU HV HW HX HY HZ IA IB IC ID IE IF IG IH II IJ IK IL IM IN IO IP IQ IR IS IT IU IV IW IX IY IZ JA JB JC JD JE JF JG JH JI JJ JK JL JM JN JO JP JQ JR JS JT JU JV JW JX JY JZ KA KB KC KD KE KF KG KH KI KJ KL KM KN KO KP KQ KR KS KT KU KV KW KX KY KZ LA LB LC LD LE LF LG LH LI LJ LK LM LN LO LP LQ LR LS LT LU LV LW LX LY LZ MA MB MC MD ME MF MG MH MI MJ MK ML MN MO MP MQ MR MS MT MU MV MW MX MY MZ NA NB NC ND NE NF NG NH NI NJ NK NL NO NP NQ NR NS NT NU NV NW NX NY NZ OA OB OC OD OE OF OG OH OI OJ OK OL OM ON OP OQ OR OS OT OU OV OW OX OY OZ PA PB PC PD PE PF PG PH PI PJ PK PL PM PN PO PP PQ PR PS PT PU PV PW PX PY PZ QA QB QC QD QE QF QG QH QI QJ QK QL QM QN QO QP QQ QR QS QT QU QV QW QX QY QZ RA RB RC RD RE RF RG RH RI RJ RK RL RM RN RO RP RQ RR RS RT RU RV RW RX RY RZ SA SB SC SD SE SF SG SH SI SJ SK SL SM SN SO SP SQ SR SS ST SU SV SW SX SY SZ TA TB TC TD TE TF TG TH TI TJ TK TL TM TN TO TP TQ TR TS TT TU TV TW TX TY TZ UA UB UC UD UE UF UG UH UI UJ UK UL UM UN UO UP UQ UR US UT UY UZ VA VB VC VD VE VF VG VH VI VJ VK VL VM VN VO VP VQ VR VS VT VY VZ WA WB WC WD WE WF WG WH WI WJ WK WL WM WN WO WP WQ WR WS WT WY WZ XA XB XC XD XE XF XG XH XI XJ XK XL XM XN XO XP XQ XR XS XT XU XV XW XX XY XZ YA YB YC YD YE YF YG YH YI YJ YK YL YM YN YO YP YQ YR YS YT YU YV YW YX YY YZ ZA ZB ZC ZD ZE ZF ZG ZH ZI ZJ ZK ZL ZM ZN ZO ZP ZQ ZR ZS ZT ZU ZV ZW ZX ZY ZZ

Achtung:  
 64K #4: PIN 18-30V, PIN 9-3 5!  
 64K #1: PIN 16-30V, PIN 8-3 5!

NO	
ST	
EP	
<b>40</b>	
Speicher	64KB...256KB
S26361-D277-X-11	Siemens AG

28.5.84	
ULLRICH	
4	5A0788
3	25.4.85
2	27.2.82
	H.3.85

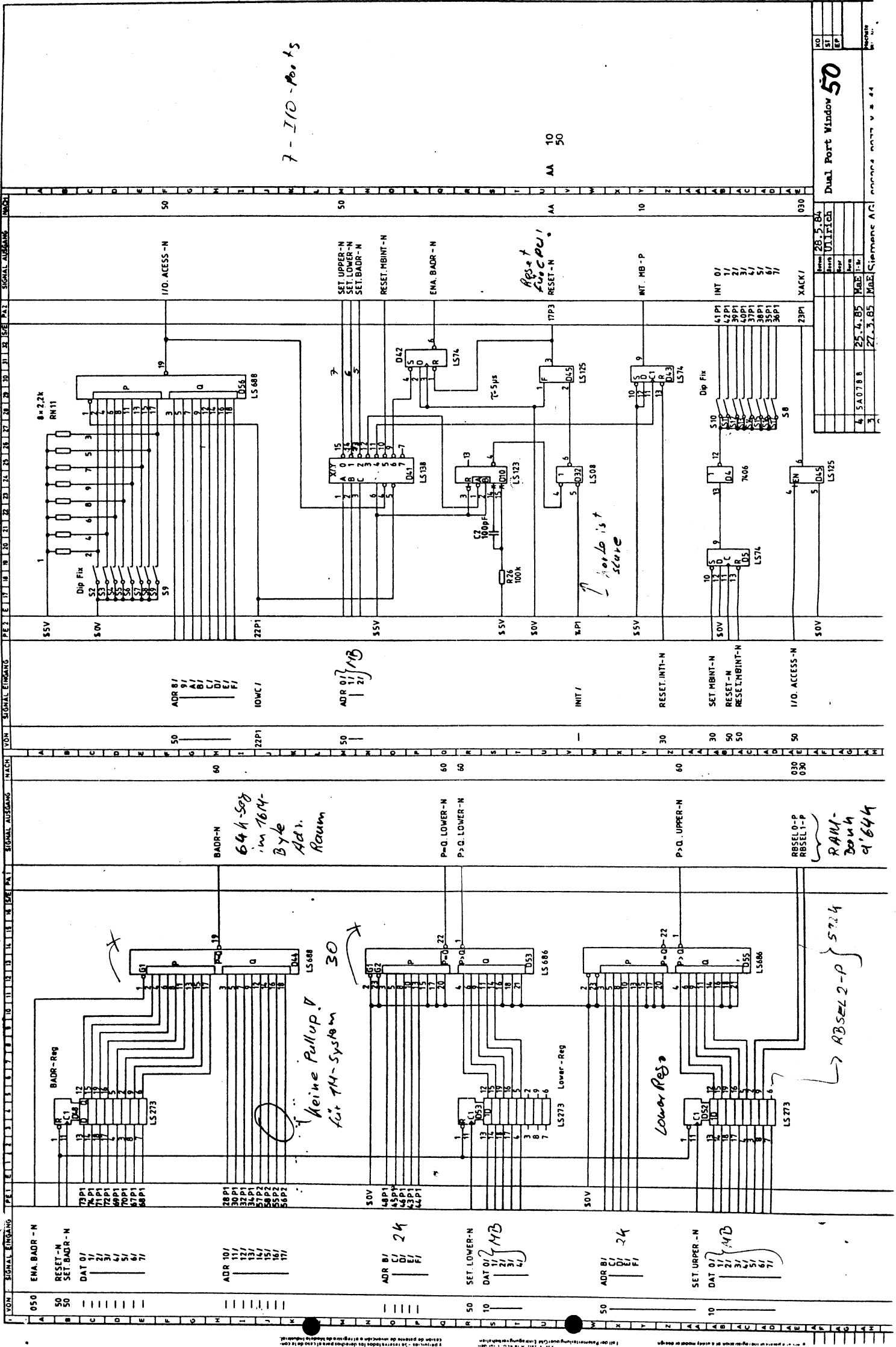
P. OUT - P	
Bank 1	
Bank 2	
Bank 3	

P. IN - P	
RAS0-N	
RAS1-N	
RAS2-N	

30	
20	
20	
20	

CAS-N 20  
 RD-N 10  
 MUX ADDR 0-P 20  
 MUX ADDR 1-P 20  
 MUX ADDR 2-P 20  
 MUX ADDR 3-P 20  
 MUX ADDR 4-P 20  
 MUX ADDR 5-P 20  
 MUX ADDR 6-P 20  
 MUX ADDR 7-P 20  
 DAT 0-P 10  
 DAT 1-P 10  
 DAT 2-P 10  
 DAT 3-P 10  
 DAT 4-P 10  
 DAT 5-P 10  
 DAT 6-P 10  
 DAT 7-P 10  
 P. IN - P 30  
 RAS0-N 20  
 RAS1-N 20  
 RAS2-N 20

Some rights reserved. All rights are reserved in the event of the original application. The name of the manufacturer is not to be used in advertising or promotional material without the express written consent of the manufacturer.



7- I/O-Ports

Reset  
für CPU!

AA 10  
AA 50

BADR-N  
64k-Soj  
im 16M-  
Byte  
Adi.  
Raum

Keine Pullup  
für TM-System

ADR B/  
C/  
D/  
E/  
F/

Lower Reg

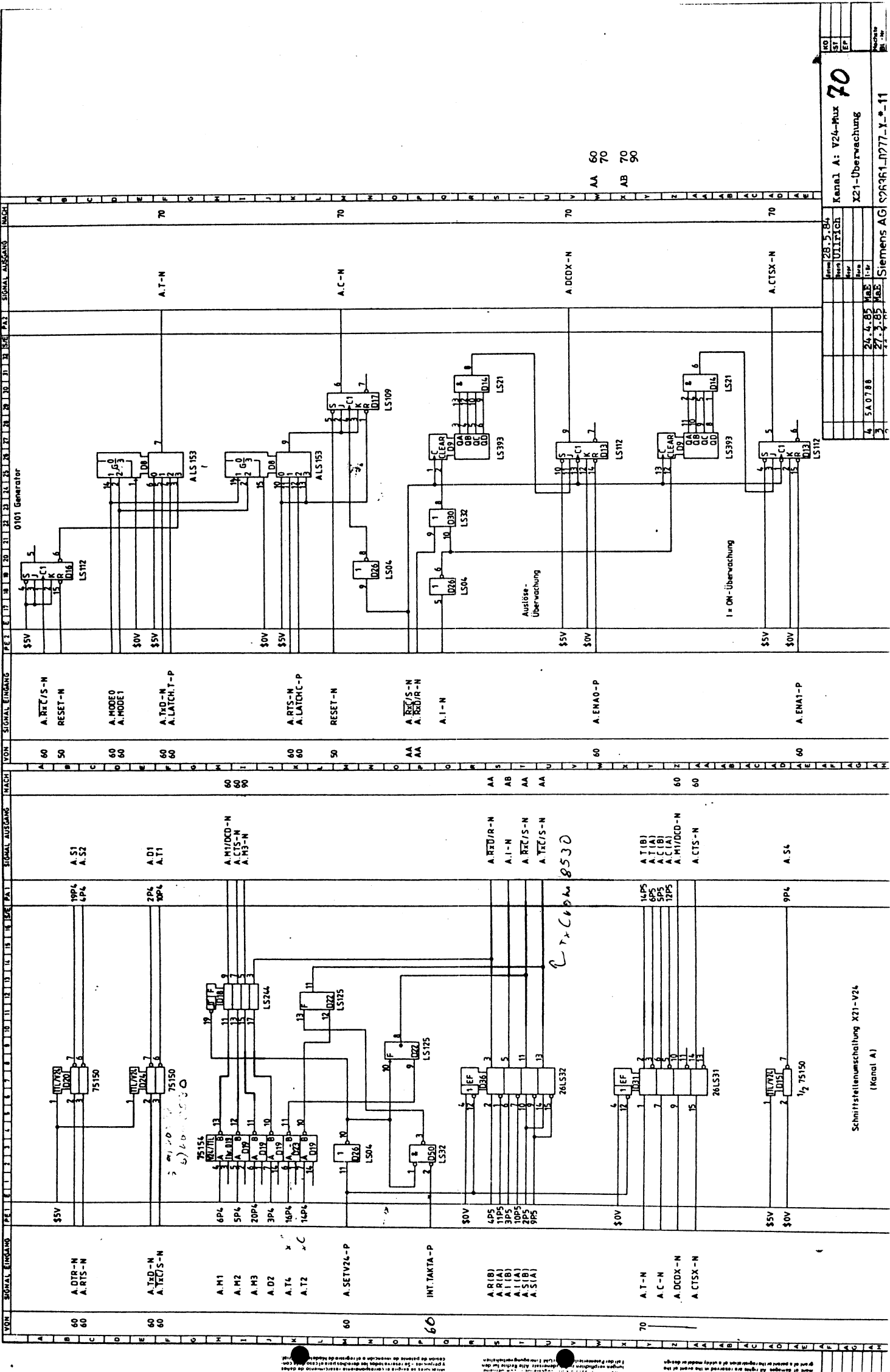
RAM-  
Bereich  
64k

Dual Port Window 50

Item	28.5.84	25.4.85	27.3.85
Beitrag	1/	1/	1/
Niedrig	2/	3/	4/
Menge	4/	5/	6/
Preis	1/	1/	1/

M&E Siemens AG





AA 60  
AB 70  
AC 70  
AD 70  
AE 70

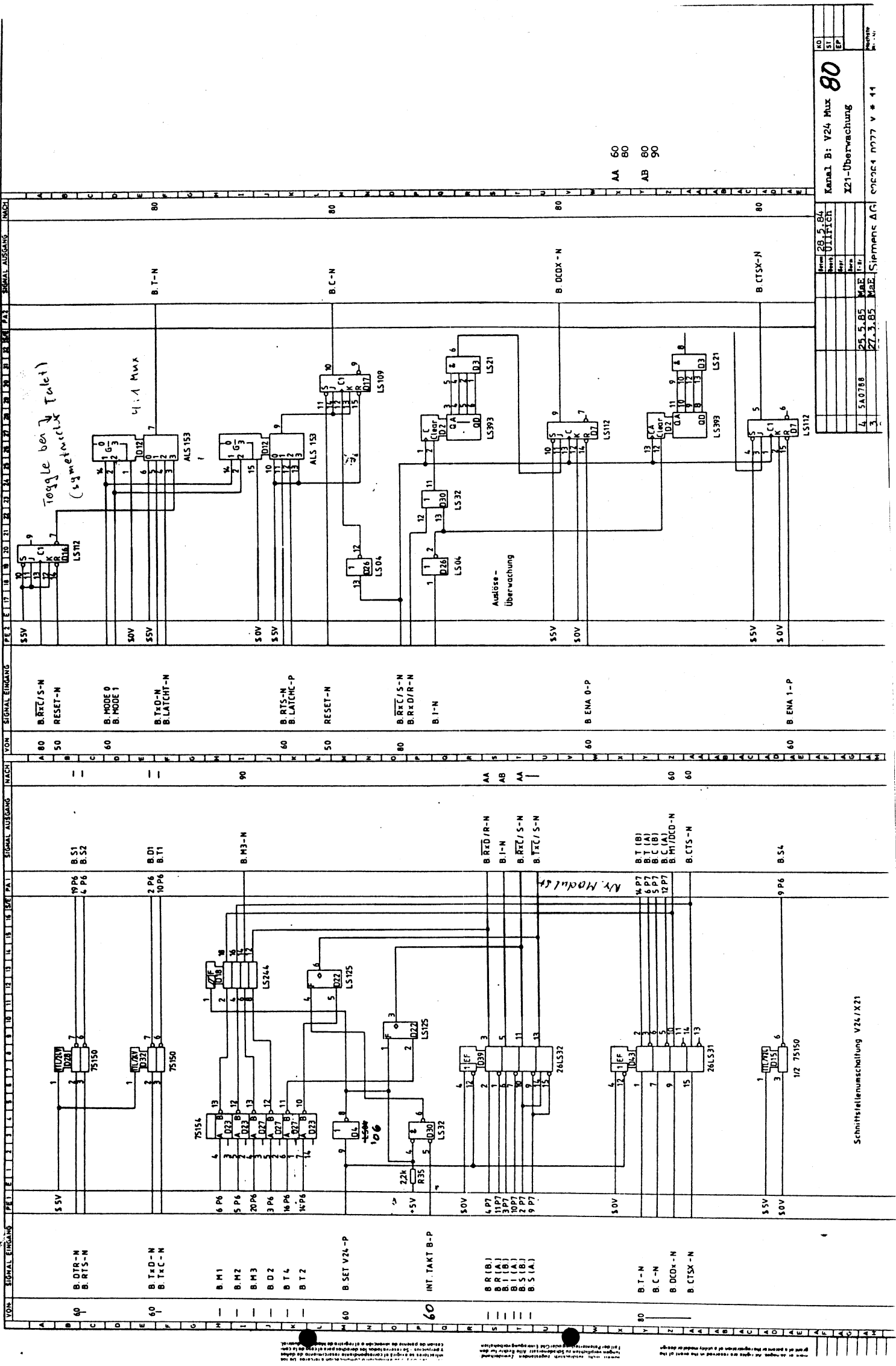
Kanal	28	5	84	U11	F101	U11	F101
4	5A.0788	24.4.95	10.8				
3		27.2.95	10.8				

Kanal A: V24-Mux  
X21-Überwachung  
Siemens AG SP6961-1077-Y-11

0101 Generator

Schnittstellenschaltung X21-V24  
(Kanal A)

Handwritten notes and signatures at the bottom of the page.



AA	60
AB	80
AC	80
AD	90

AA	60
AB	80
AC	80
AD	90

AA	60
AB	80
AC	80
AD	90

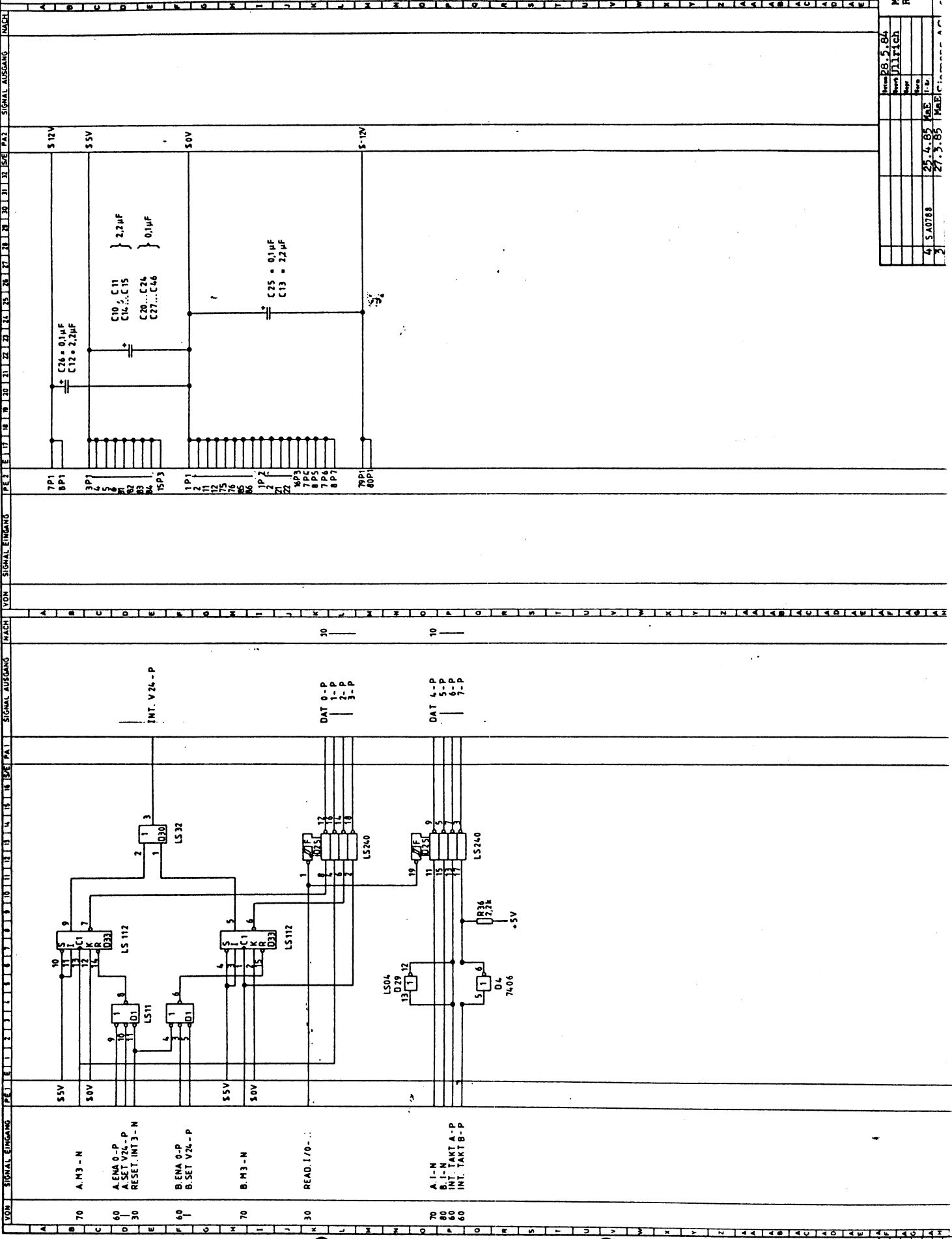
AA	60
AB	80
AC	80
AD	90

AA	60
AB	80
AC	80
AD	90

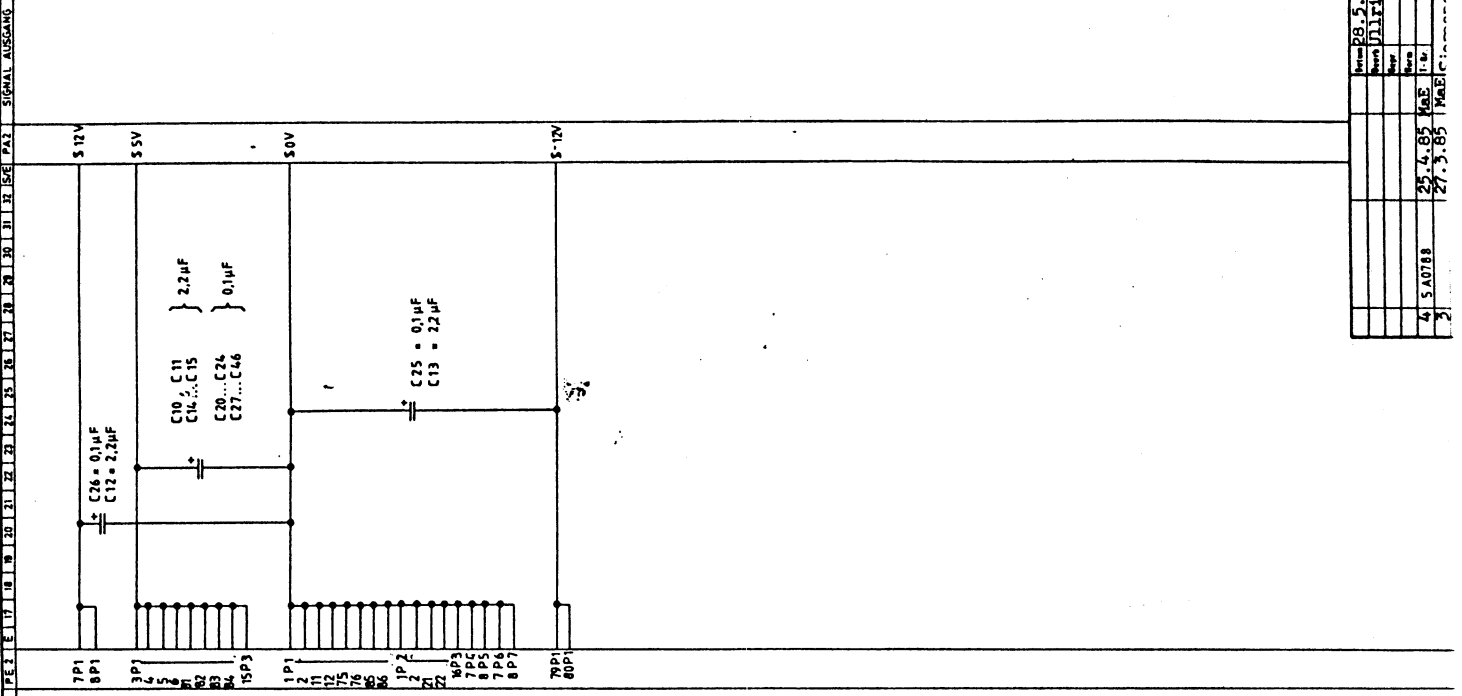
AA	60
AB	80
AC	80
AD	90

Schrittrelenumschaltung V24/X21

Kanal B: V24 Mux 80  
X21-Überwachung  
Siemens AG



VON SIGNAL EINGANG NACH SIGNAL AUSGANG VON SIGNAL EINGANG NACH SIGNAL AUSGANG



7P1 BP1 5V  
 7P2 BP2 5V  
 1P1 IP1 0V  
 1P2 IP2 0V  
 7P1 BP1 5-12V  
 6P1 BP1 5-12V

70 A.M3 - N  
 60 A.EMA 0-P  
 30 A.SET V26-P  
 30 RESET.INT 3-N  
 60 B.EMA 0-P  
 1 B.SET V26-P  
 70 B.M3 - N  
 30 READ.I/O-  
 70 A.I - N  
 80 B.I - N  
 60 INT.TAKT A - P  
 60 INT.TAKT B - P

NO	SI	EP
90		
M1-Überwachung Read I/O-Latch		
4	5A0788	25 4.85
3		27.5.85

Rev.	By	Date
1		

no included in the original design. The design is the property of the company. All rights reserved. The company is not responsible for any damage or loss of data. The company is not responsible for any damage or loss of data.

\*DEBUG TYPE  
\*PAGELENGTH(65)

Hv. Reif 089/636-46216  
Hv. Ullrich 0827/599-2103  
Hv. Haidbender 089/636-47987

NAME FRM277

\*\*\*\*\*  
\*  
\* INITIALISIERUNG UND SELBSTEST \*  
\*  
\* DER FLBGR. -D277 (DUEAI) \*  
\*  
\*\*\*\*\*

ULLRICH K WS TD 132 5. 8.1985  
TEL. 2103

VERSION: \*.3 VORSERIE  
CHECKBYTE: 1FFEH: H  
1FFFH: 3EH

*A8 2C*  
*3EH*  
*STSP 472*  
*C9*

INITIALISIERUNGSABLAUF

*Charles de Gaulles Str.*  
*Mü 83*

1. INITIALISIEREN DES KONTROLL-BLOCKS DES B0188  
LOC. NACH 1000H IM I/O-ADRESSRAUM
2. KOMTEST: BEI FEHLER GEHT LED KURZZEITIG 1X AUS (0.5 S).  
VERSUCH EINER FEHLERMELDUNG IM KOM-BEREICH
3. RAMTEST: BEI FEHLER GEHT LED KURZZEITIG 2 X AUS  
  
TESTMUSTER: 2 AUS 8 CODIERUNG  
WIEDERHOLUNG DES TEST MIT VIER STARTMUSTERN  
TESTMUSTER  
BEI FEHLER LED GEHT 2X AUS UND VERSUCH EINER FEHLERMELDUNG  
IM KOM-BEREICH
4. LOCAL LOOP DES SCC: BEI FEHLER VERSUCH EINER FEHLERMELDUNG  
IM KOM-BEREICH. LED GEHT 3X AUS
5. KOM-BEREICH FESTLEGEN  
LED BLINKT IM 1 HZ RHYTHMUS  
WARTEN AUF 1. INT VOM MULTIBUS  
QUITTIEREN DURCH INT AN HAUPTPROZ.  
WARTEN AUF 2. INT VOM MULTIBUS  
EINSPRUNG IN WARMLADER

\*EJECT

SYMBOLTABELLE

PUSHA	EQU	60H;	OPCODE FUR PUSH ALL	
POPA	EQU	61H;	" POP "	
SEG_ROM	EQU	0FE00H;	PHYS. ADRESSE	: EPROM (8K)
SEG_DATA	EQU	0H;		: DATENSEGMENT
SEG_STACK	EQU	0H;		: STACKSEGMENT
STCK_TOP1	EQU	1FH;		: STCKPTR VOR RAMTES
STCK_TOP2	EQU	0FFFH;		: " NACH "
SEG_KOM	EQU	0B00H;		: IP-FENSTER
SEG_WARML	EQU	0B02H;		: WARMLADER



```

ROMLAENGE      EQU    1FFCH;    EPROM: 8 K (-2 BYTE)
RAMANFANG1     EQU    20H;      (64K - 20H BYTE)
RAMLAENGE1     EQU    7FF0H;
RAMANFANG2     EQU    0H;
RAMLAENGE2     EQU    7FFFH;    VOLLSTAENDIGES 64K SEGMENT
    
```

; I/O-ADRESSEN

```

SCC_STAT_B     EQU    0H      PCS0
SCC_DATA_B     EQU    1H
SCC_STAT_A     EQU    2H
SCC_DATA_A     EQU    3H
RES_INT1       EQU    80H; PCS1 INT. VOM MULTIBUS
RES_INT3       EQU    100H; PCS2 V24 (M3) INTERRUPT
SET_MBINT      EQU    180H; PCS3 MULTIBUS-INTERRUPT
SET_PROZ_MODE  EQU    200H; PCS4 PROZESSOR-MODE-REGISTER
SET_IO_A       EQU    210H; V  SCHNITTSTELLEN-REG.
SET_IO_B       EQU    220H;
READ_IO        EQU    230H; V  V24-LATCH
    
```

```

;
UMCS_A         EQU    10A0H;    AB 1000H IN I/O-SPACE
LMCS_A         EQU    10A2H;    ADRESSEN DER EINZELNEN INTERNEN
PACS_A         EQU    10A4H;    REGISTER IM KONTROLLBLOCK
MMCS_A         EQU    10A6H
MPCS_A         EQU    10A8H
    
```

```

INTO_CTR_A     EQU    1038H;    INTERRUPT-CONTROLLER
INT1_CTR_A     EQU    103AH
INT3_CTR_A     EQU    103EH
INT_REQ        EQU    102EH
EOI_A          EQU    1022H
    
```

\*EJECT

; INITIALISIERUNGS-WERTE

```

;
RELOC          EQU    10H;    RELOCATION REG.: KONTROLL-BLOCK
;                                     AB 1000H IM I/O-SPACE
;
UMCS           EQU    0FE3CH;    UPPER CHIP SEL., 8KB, 0 WAIT, NO EXT.
LMCS           EQU    03FF8H;    LOWER CHIP SEL., 256KB, 0 WAIT AND
;                                     EXT. RDY
;
PACS           EQU    3EH;    PERIPHERAL BASE ADDRESS: 0H, 2 WAIT
;                                     FOR PCS 0..3, NO EXT. RDY
;
PACS           EQU    3BH;    3 WAIT AND EXTERNAL RDY
;
MMCS           EQU    41FCH;    MID RANGE BASE ADDRESS: 40000H, NO WAI
;                                     NO EXT. RDY
;
MPCS           EQU    01BDH;    MID RANGE SIZE: 8KB, 7 PCS LINES, 1 WA
;                                     FUER PCS 4...7, NO EXT. RDY
;                                     0 ist falsch!
;
INTO_MASK      EQU    28H;    SCC-INTERRUPT: CASCADE-MODE, EDGE TRIG
;                                     MASKIERT
;
INTO_ENA       EQU    20H;    "          ENABLED, HIGHEST PRIO.
;
INT1_MASK      EQU    0EH;    MB-INTERRUPT: NONCASCADE, EDGE TRIGG.,
;                                     MASKIERT
;
INT1_ENA       EQU    06H;    "          ENABLED, LOW PRIO.
;
INT3_MASK      EQU    0FH;    V24-INTERRUPT: IMMER MASKIERT.
;
    
```

*4 \* 64k = 256k*

3

```

EOI          EQU    8000H;  NON-SPECIFIC END OF INT.
WR3_0        EQU    0C003H;  RXD 8-BIT, RX DISABEL
WR3_1        EQU    0C103H;  "          ENABLE
WR4          EQU    4504H;   x16 CLOCK, 1 STOP-BIT, PARITY ODD
WR5_0        EQU    6005H;  TXD 8-BIT, TX DISABLE
WR5_1        EQU    6805H;  "          ENABLE
WR11         EQU    500BH;   BR-GEN = TXC UND RXC
WR12         EQU    10CH;   38400 B/S
WR13         EQU    0DH;
WR14         EQU    130EH;  LOCAL LOOP, BR-GEN. ENABLE
    
```

\*EJECT

; BITMUSTER UND MASKEN

```

SI_BIT       EQU    1H
ROMBIT       EQU    8201H
RBO_BIT      EQU    8405H
RB1_BIT      EQU    8409H
RB2_BIT      EQU    8411H
RB3_BIT      EQU    8421H
RL_BIT       EQU    8441H
RH_BIT       EQU    8481H
STACK_BIT    EQU    84C5H
PAR0_BIT     EQU    8805H
PAR1_BIT     EQU    8809H
PAR2_BIT     EQU    8811H
PAR3_BIT     EQU    8821H
SCC_BIT      EQU    9001H
    
```

```

ADUTSTART_BIT EQU    1H
FRMVR5        EQU    3H;  FIRMWARE-VERSION
    
```

```

TESTMUSTER   EQU    0DE21H;  TESTANFANG FUER RAMTEST
RAMSIZE_0    EQU    1H;    0 ... 64KB
RAMSIZE_1    EQU    2H;    64 ... 128KB
RAMSIZE_2    EQU    3H;    128 ... 192KB
RAMSIZE_3    EQU    4H;    192 ... 256KB
    
```

```

ENA_NMI      EQU    0H
DIS_NMI      EQU    4H;  BIT IM PROZESSOR-MODE-REG
ENA_SCC      EQU    8H
    
```

\*EJECT

; PROGRAMMBEGINN

; -----

```

STACK SEGMENT
      DB 100H DUP (?)
STACK ENDS
    
```

KOM BEREICH SEGMENT

DB 100 DUP (?)

KOM\_BEREICH ENDS

INTERN\_RAM SEGMENT

RAM_AUSBAU	DB	?	
JUMP_FLAG	DB	?	SIGN. EINSPRUNG IN WARMLADER
NMI_TESTFLG	DB	?	SIGN. FEHLENDE RAMBAENKE

ORG 8H

OFF_NMI	DW	?	INTERRUPT-TABLE
SEG_NMI	DW	?	

ORG 30H

OFF_SCC	DW	?	INT 0
SEG_SCC	DW	?	
OFF_INT1	DW	?	MB-INTERRUPT
SEG_INT1	DW	?	

ORG 0B000H ⇒ 1MByte .bss

STATUS	DW	?	7	}
CMD	DW	?	7	
FIRMWARE	DB	?	4	
RAMSIZE	DB	?	5	
RES_1	DW	?	8	
RES_2	DW	?	3	
RES_3	DW	?	A	
RES_4	DW	?	C	
RES_5	DW	?	E	
A_MAGIC	DW	?	11	}
A_TEXT	DW	?	12	
A_DATA	DW	?	14	}
A_BSS	DW	?	16	
A_SYMS	DW	?	18	
A_ENTRY	DW	?	1A	}
A_UNUSED	DW	?	1C D	

ES: ↓

↓ ↓

0.out.6

offset → ip  
→ CS

2 F. d

1E F

INTERN\_RAM ENDS

SEG-Lader → 103020

INIT SEGMENT AT OFFFOH

ASSUME CS:INIT

```
START: MOV DX,OFFFEH;      RELOCATION-REG AUF 1000 SETZEN
        MOV AX,RELOC = 0010H
        OUT DX,AX
```

```
MOV BX,SEG_DATA; } BX ← 15
MOV DS,BX
MOV BX,SEG_STACK }
MOV SS,BX
MOV SP,STCK_TOP1 - 7F
```

DATENSEGMENT UND STACKSEGMENT LADEN

```
MOV BX,SEG_KOM; } ← 0B00H
MOV ES,BX
```

ES ZEIGT AUF KOMMUNIKATIONSBEREICH

\*EJECT

5

```

MOV IX,UMCS_A; 10A0 UPPER CHIP SELECT FESTLEGEN
MOV AX,UMCS FE3C
OUT IX,AX
    
```

```

MOV IX,LMCS_A; LOWER
MOV AX,LMCS
OUT IX,AX
    
```

```

MOV IX,PACS_A; 10A4 PERIPHERAL BASE ADDRESS
MOV AX,PACS 0038
OUT IX,AX
    
```

```

MOV IX,MMCS_A; 10A6 MID RANGE CHIP SELECT
MOV AX,MMCS 47FC
OUT IX,AX
    
```

```

MOV IX,MPCS_A; 10A8 MID RANGE SIZE, PCS 4..7
MOV AX,MPCS 018DH
OUT IX,AX
    
```

```

MOV IX,RES_INT1; 0180H RESET INT1- UND INT3-FF
OUT IX,AL
MOV IX,RES_INT3 E100H
OUT IX,AL
    
```

```

MOV IX,INTO_CTR_A; 1038H INT-CONTR. INIT.
MOV AX,INTO_MASK; SCC-INT MASKIEREN
OUT IX,AX 26
    
```

```

MOV IX,INT1_CTR_A; 103AH MB-INT. MASK.
MOV AX,INT1_MASK
OUT IX,AX 0E
    
```

```

MOV IX,INT3_CTR_A; 103E V24-INT. MASK.
MOV AX,INT3_MASK
OUT IX,AX
    
```

```

0200H = PCS4
MOV IX,SET_PROZ_MODE; NMI DISABLEN
MOV AL,DIS_NMI
OUT IX,AL
JMP FAR PTR ROMTEST
    
```

INIT ENDS

\*EJECT

BOOT SEGMENT

ASSUME CS:BOOT, ES:KOM\_BEREICH, SS:STACK, DS:INTERN\_RAM

; ROMTEST

; -----

;

- ; AX: LAUFENDE SUMME
- ; IX: LAUFENDE ADRESSE
- ; CX: ENDADRESSE
- ; IX: GELESENE CHECKSUMME

```

ROMTEST:MOV AX,0H
MOV BX,OFFFEH
MOV CX,ROMLAENGE
    
```

6

```

AO:   INC BX
      INC BX
      ADD AX,CS:[BX]
      CMP BX,CX;
      JNZ AO
    
```

ROMENDE ERREICHT ?

```

      ADD BX,2
      MOV CS:[BX],AX;
      MOV IX,CS:[BX];
      CMP AX,IX
      JZ  STACKTEST_ANF
    
```

SUMME IN LETZTE ADRESSE SCHREIBEN  
 SUMME AUS LETZTER ADRESSE LESEN

ROMERROR:

```

      MOV AX,ROMBIT
      MOV STATUS,AX;
      MOV AL,FRMURS
      MOV FIRMWARE,AL
      MOV AL,RAMSIZE_3
      MOV RAMSIZE,AL
    
```

VERSUCH EINER FEHLERMELDUNG IM STATUS

```

ROM1:  MOV IX,SET_PROZ_MODE;
      MOV AL,80H
      OUT IX,AL;
      MOV CX,0FFFFH
    
```

BEI ROM\_FEHLER SCHNELLES BLINKEN

LED AUS

AUS\_1:

```

      NOP
      NOP
      NOP
      NOP
      NOP
      NOP
    
```

6

```

      LOOP AUS_1
    
```

```

      MOV AL,0H
      OUT IX,AL;
    
```

LED AN

```

AN_1:  MOV IX,INT_REQ
      IN  AX,IX
      AND AX,020H
      CMP AX,020H
      JNE AN_1
    
```

QUITTUNGS-INTERRUPT

```

      MOV IX,SET_MBINT;
      OUT IX,AX
    
```

```

AN_2:  JMP AN_2
    
```

```

*EJECT
; SPEICHER-TEST
; -----
    
```

; ZUNAECHEST MUSS DER SPEICHER-BEREICH VON 0H BIS 1FH GETESTET  
 ; WERDEN. ER DIENST IM WEITEREN GESCHEHEN DANACH ALS STACK.  
 ; (DABEI WIRD NMI DISABLED)  
 ; FALLS STACK-ERROR: LED BLINKT MITTELSCHNELL UND INITIALISIERUNG WIRD  
 ; ABGEBROCHEN.

```

STACK_ERR:MOV AX,STACK_BIT
          MOV STATUS,AX
          MOV FIRMWARE,FRMURS
          MOV RAMSIZE,RAMSIZE_3
          JMP ERROR_LOOP
    
```

```

STACKTEST_ANF:
          MOV CX,1FH
    
```

STACKTEST:

```
MOV AL,OFFH
MOV BX,CX;      ADRESSIERUNG DER ZU TESTENDEN ADRESSEN UEBER BX
MOV [BX],AL
CMP [BX],AL
JNE STACK_ERR
```

```
MOV AL,55H
MOV [BX],AL
CMP [BX],AL
JNE STACK_ERR
```

```
MOV AL,0AAH
MOV [BX],AL
CMP [BX],AL
JNE STACK_ERR
```

```
MOV AL,0H;      JMP-FLAG DAMIT 0
MOV [BX],AL
CMP [BX],AL
JNE STACK_ERR
LOOP STACKTEST
```

\*EJECT     ↓ CX=0

VEC\_INIT:

```
MOV OFF_NMI,OFFSET NMIO_ROUTINE
MOV SEG_NMI,SEG_ROM
MOV NMI_TESTFLG,0;      NMI-FLAG RUECKSETZEN

MOV IX,SET_PROZ_MODE
MOV AL,ENA_NMI;      NMI ENABLE
OUT IX,AL
```

; FALLS DER STACKBEREICH O.K. IST, KOENNEN ALLE UEBRIGEN SPEICHERTESTS  
; UEBER CALLS AUFGERUFEN WERDEN.  
; ALS 2. SCHRITT WIRD NUN BEREICH VON 20H BIS OFFFFH (1. 64K SEGMENT)  
; GETESTET.  
; FALLS PARITY- ODER RAM\_FEHLER: SPRUNG IN NMIX\_ROUTINE UND PARITY-BIT  
; BZW RAMBANK-BIT IM STATUS-REGISTER (KOMMBEREICH) SEZEN

RAMBANKO:

```
MOV RAM_AUSBAU,RAMSIZE_0
MOV AX,RBO_BIT
PUSH AX
MOV AX, RAMLAENGE1
PUSH AX;      ALLE PARAMETER WERDEN IM STACK UEBERGE
MOV AX, RAMANFANG1
PUSH AX
CALL RAMTEST
```

```
MOV SP,STCK_TOP2;      ENTGUELTIGEN STACK BEI OFFFH ERR
MOV OFF_INT1,OFFSET FIRST_INT1; INTERRUPT-TABLE VERVOLLSTAENDIGE
MOV SEG_INT1,SEG_ROM
MOV OFF_NMI,OFFSET NMI_TEST
MOV SEG_NMI,SEG_ROM
```

\*EJECT

; DANACH MUSS DER WEITERE SPEICHERAUSBAU FESTGESTELLT WERDEN.  
; IIAZU WIRD UNTER DEN ADRESSEN 0H UND 10000H VERSCHIEDEN BITMUSTER  
; GESCHRIEBEN UND ANSCHLIESSEND WIEDER GELESEN.  
; WURDE BEIM LESEN EIN NMI AUSGELOEST, IST DIE ENTSPRECHENDE RAMBANK  
; NICH BESTUECKT. WIRD KEIN NMI AUSGELOEST, WIRD NORMALER RAMTEST AUFGERU

RAMBANK1:

```

MOV AX,1000H;
MOV DS,AX
MOV BX,0H
MOV DL,51H
MOV [BX],DL;
MOV DH,[BX]
MOV AX,SEG_DATA;
MOV DS,AX;

```

UEBER DS WIRD RAMBANK 1 ADRESSIERT

TEST: ADRESSE 10000H

DATENSEGMENT WIEDER INSTALLIEREN

```

CMP DL,DH;
JE WEITER1;
JMP SCC_TEST

```

MUSS AUS KOMPATIBILITAETSGRUENDEN ZU DEN  
PROTOTYPEN GEMACHT WERDEN

```

MOV AL,NMI_TESTFLG
CMP AL,1H;
JNE WEITER1
JMP SCC_TEST

```

FALLS NMI: KEIN WEITERER SPEICHER DA

```

;
;
;
;
SPEICHERAUSBAU GROESSER WIE 64K
----> ES WIRD DAS NAECHSTE SEGMENT
GETESTET.

```

WEITER1:

```

MOV AX,RB1_BIT
PUSH AX
MOV AX,RAMLAENGE2
PUSH AX
MOV AX,RAMANFANG2
PUSH AX
MOV OFF_NMI,OFFSET NMI1_ROUTINE
MOV SEG_NMI,SEG_ROM
MOV RAM_AUSBAU,RAMSIZE_1

```

```

MOV AX,1000H;
MOV DS,AX
CALL RAMTEST
MOV AX,SEG_DATA;
MOV DS,AX

```

DATENSEGMENT AENDERN

RAMSIZE MERKEN

\*EJECT

;
RAMBANK2:;

WEITERER SPEICHER VORHANDEN ?  
(128KB - 192KB)

```

MOV OFF_NMI,OFFSET NMI_TEST
MOV SEG_NMI,SEG_ROM

```

```

MOV AX,2000H;
MOV DS,AX
MOV BX,0H
MOV DL,51H
MOV [BX],DL;
MOV DH,[BX]
MOV AX,SEG_DATA;
MOV DS,AX;

```

UEBER DS WIRD RAMBANK 2 ADRESSIERT

TEST: ADRESSE 20000H

DATENSEGMENT WIEDER INSTALLIEREN

```

CMP DL,DH;
JE WEITER2;
JMP SCC_TEST

```

MUSS AUS KOMPATIBILITAETSGRUENDEN ZU DEN  
PROTOTYPEN GEMACHT WERDEN

```

MOV AL,NMI_TESTFLG
CMP AL,1H;
JNE WEITER2
JMP SCC_TEST

```

FALLS NMI: KEIN WEITERER SPEICHER DA

WEITER2:MOV AX,RB2\_BIT  
PUSH AX

```

MOV AX,RAMLAENGE2
PUSH AX
MOV AX,RAMANFANG2
PUSH AX
MOV OFF_NMI,OFFSET NMI2_ROUTINE
MOV SEG_NMI,SEG_ROM
MOV RAM_AUSBAU,RAMSIZE_2

```

```

MOV AX,2000H
MOV DS,AX

```

```

CALL RAMTEST;          TEST DES 2. SEGMENTS
MOV AX,SEG_DATA
MOV DS,AX

```

```

;          WEITERER SPEICHER VORHANDEN ?
;          (192KB - 256KB)
RAMBANK3:

```

```

MOV OFF_NMI,OFFSET NMI_TEST
MOV SEG_NMI,SEG_ROM

```

```

MOV AX,3000H;          UEBER ES WIRD RAMBANK 3 ADRESSIERT
MOV DS,AX
MOV BX,0H
MOV DI,51H
MOV [BX],DI;          TEST: ADRESSE 30000H
MOV DI,[BX]
MOV AX,SEG_DATA;
MOV DS,AX;            DATENSEGMENT WIEDER INSTALLIEREN

```

```

CMP DI,DI;            MUSS AUS KOMPATIBILITAETSGRUENDEN ZU DEN
JE WEITER3;           PROTOTYPEN GEMACHT WERDEN
JMP SCC_TEST

```

```

MOV AL,NMI_TESTFLG
CMP AL,1H;            FALLS NMI: KEIN WEITERER SPEICHER DA
JNE WEITER3
JMP SCC_TEST

```

```

WEITER3:

```

```

MOV AX,RB3_BIT
PUSH AX
MOV AX,RAMLAENGE2
PUSH AX
MOV AX,RAMANFANG2
PUSH AX
MOV OFF_NMI,OFFSET NMI3_ROUTINE
MOV SEG_NMI,SEG_ROM
MOV RAM_AUSBAU,RAMSIZE_3

```

```

MOV AX,3000H
MOV DS,AX
CALL RAMTEST;
MOV AX,SEG_DATA
MOV DS,AX

```

```

*EJECT

```

```

SCC_TEST:

```

```

MOV AX,SEG_DATA;     SEGMENTREGISTER WIEDER ERRICHTEN
MOV DS,AX

```

```

MOV DX,SET_PROZ_MODE;  SCC ENABLE

```



```

MOV AL,ENA_SCC
OUT IX,AL
MOV DX,SCC_STAT_A;      INITIALISIERUNG KANAL A
CALL SCC_INIT

MOV BL,55H;              TESTMUSTER (55H) UEBERGEHEN
MOV IX,SCC_DATA_A
CALL LOCAL_LOOP
MOV BL,0AAH;              "      (0AAH)      "
MOV IX,SCC_DATA_A
CALL LOCAL_LOOP

MOV IX,SCC_STAT_B;      TESTWIEDERHOLUNG FUER KANAL B
CALL SCC_INIT
MOV BL,55H
MOV IX,SCC_DATA_B
CALL LOCAL_LOOP
MOV BL,0AAH
MOV IX,SCC_DATA_B
CALL LOCAL_LOOP

```

\*EJECT

```

MELDEN:;                  STATUSBEREICH DEFINIEREN
MOV AX,SI_BIT
MOV STATUS,AX
MOV FIRMWARE,FRMVR8
MOV AL,RAM_AUSBAU
MOV RAMSIZE,AL

MOV OFF_INT1,OFFSET FIRST_INT1;  INTERRUPT-VEKTOREN FESTLEGEN
MOV SEG_INT1,SEG_ROM

STI;                      PROZ. INT ENABLE
MOV DX,INT1_CTR_A;        MB-INT. ENABLE
MOV AX,INT1_ENA
OUT IX,AX

L10:  MOV DX,SET_PROZ_MODE;    LANGSAMER BLINKEN DER LED
      MOV AL,8BH
      OUT DX,AL;              LED AUS
      MOV BX,3H;              AUESSERE SCHLEIFE
      MOV CX,0A000H;          INNERE "

LED_AUS10:
      CMP JUMP_FLAG,01H;
      JE START_LAIER
      LOOP LED_AUS10

      MOV CX,0A000H
      DEC BX
      CMP BX,0H
      JNE LED_AUS10

      MOV AL,0H
      OUT DX,AL;              LED AN
      MOV BX,3H
      MOV CX,0A000H

LED_AN10:
      CMP JUMP_FLAG,01H;
      JE START_LAIER
      LOOP LED_AN10

      MOV CX,0A000H
      DEC BX

```

(17)

```
CMP BX,0H
JNE LED_AN10
JMP L10
```

START\_LADER:

```
MOV DX,SET_PROZ_MODE;
MOV AL,88H
OUT DX,AL; 0B01C LED AUS

MOV A_UNUSED,SEG_LADER 0B020
MOV AX,SEG_LADER
MOV DS,AX; DS UND SS ZEIGEN AUF WARMLADER
MOV SS,AX
MOV BX,1AH
JMP DWORD PTR ES:[BX]; → EINSPRUNG IN WARMLADE
```

\*EJECT

```
; INTERRUPT_ROUTINEN 0B00 001A → 0B07A → q.out.h → Entry
; -----
```

; DER 1. INTERRUPT VOM HAUPTPROZESSOR WIRD DURCH EINEN QUITTUNGSINTERRUPT  
; BEANTWORTET

FIRST\_INT1 LABEL NEAR

```
DB PUSHA
MOV DX,RES_INT1; RESET INT-FF
OUT DX,AL

MOV DX,SET_MBINT; INT QUITTIEREN
OUT DX,AL

MOV OFF_INT1,OFFSET SECOND_INT; INT-VEKTOR AUF 2. INT SETZEN
MOV DX,EOI_A; END OF INT
MOV AX,EOI
OUT DX,AX
DB POPA
IRET
```

; NACH DEM 2. INTERRUPT VOM SYSTEM WIRD DAS BEFEHLSREGISTER NACH  
; AOUTSTART GETESTET.  
; FALLS GESETZT, WERDEN DIE SEGMENTREGISTER DEFINIERT GESETZT  
; UND DER WARMLADER GESTARTET

SECOND\_INT LABEL NEAR

```
DB PUSHA
MOV DX,RES_INT1; INT-FF RUECKSETZEN
OUT DX,AL

CMP CMD,AOUTSTART_BIT; GUELTIGER CMD VOM MB?
JNE BACK
MOV JUMP_FLAG,01H; SPRUNG SIGNALISIEREN
BACK: MOV DX,EOI_A; INT-CONTR. "
MOV AX,EOI
OUT DX,AX
DB POPA
IRET
```

```
NMI_TEST:;
    PUSH DS;
    PUSH AX
    MOV AX,SEG_DATA
    MOV DS,AX
    MOV NMI_TESTFLG,1H
    POP AX
    POP DS
    IRET
```

WIRD AUSGELÖST, FALLS RAMBANK NICHT  
BESTUECKT

; NMIO ... NMI3 WERDEN ANGESPRUNGEN, FALLS WAEHREND DER RAMTESTS EIN  
; PARITY-FEHLER AUFTRITT.

```
NMIO_ROUTINE:
    MOV STATUS,PAR0_BIT;
    MOV AX,0
    MOV DS,AX
    JMP ERROR_LOOP
```

VERSUCH EINER FEHLERMELDUNG

```
NMI1_ROUTINE:
    MOV STATUS,PAR1_BIT
    JMP ERROR_LOOP
```

*MOV AX, SEG\_DATA*  
*MOV DS, AX*  
*MOV FIRMWARE, FIRMVRS*  
*MOV RAMSIZE, RAMSIZE-1*

```
NMI2_ROUTINE:
    MOV STATUS,PAR2_BIT
    JMP ERROR_LOOP
```

```
NMI3_ROUTINE:
    MOV STATUS,PAR3_BIT
```

ERROR\_LOOP:

```
    MOV IX,SET_PROZ_MOIE
    MOV AL,84H
    OUT IX,AL;          LED AUS, NMI DIS.
    MOV AL,55H
    MOV BX,0H
    MOV CX,0FFFFH
L1:  MOV [BX],AL;
    MOV AH,[BX];
    LOOP L1

    MOV AL,4H
    OUT IX,AL;          LED AN
    MOV CX,0FFFFH
L2:  MOV [BX],AL;
    MOV AH,[BX];
    LOOP L2

    MOV AL,84H
    OUT IX,AL
    MOV CX,0FFFFH
L3:  MOV [BX],AL;          LED AUS
    MOV AH,[BX];
    LOOP L3;

    MOV AL,4H
    OUT IX,AL;          LED AN
    MOV CX,0FFFFH
```

```

L4:  MOV [BX],AL;
      MOV AH,[BX];
      LOOP L4

L5:  MOV AL,55H
      MOV [BX],AL;          TESTMUSTER SCHREIBEN
      MOV AH,[BX];          UND LESEN
      MOV DX,INT_REQ
      IN  AX,DX
      AND AX,020H
      CMP AX,020H
      JNE L5
      MOV DX,SET_MBINT;     QUITTUNGS-INTERRUPT
      OUT IX,AX

L6:  MOV AL,55H
      MOV [BX],AL
      MOV AH,[BX]
      JMP L6
    
```

\*EJECT

RAMTEST PROC NEAR

```

; RAMTEST
; -----
    
```

```

; TESTMUSTER: 55AAH, AA55H, FF00H, 0000H
; WIRD WAEHREND DES RAMTESTS EIN FEHLER ERKANNT, WIRD DER RAMTEST SOFORT
; ABGEBROCHEN UND EINE FEHLERMELDUNG IM STATUS ANGEGEBEN
    
```

```

; AL: SCHLEIFENZAehler
; BX: LAUFENDE ADRESSE
; CX: RAMLAENGE (WORTE), MUSS IM STACK UEBERGEHEN
; DX: TESTMUSTER
; DI: INCREMENT 2 FUER NEUE ADRESSE
; SP: RAMANFANG, MUSS IM STACK UEBERGEHEN WERDEN
    
```

RAMTESTANFANG:

```

PUSH BP;          ALTEN BP RETTEN
MOV BP,SP
MOV AL,0H;        SCHLEIFENZAehler
MOV IX,55AAH;     1.TESTMUSTER
MOV DI,2H
    
```

```

WRITE: MOV CX,[BP+6]; RAMLAENGE LADEN
        MOV BX,[BP+4]; RAMANFANG
    
```

```

WR_LOOP:
        MOV [BX],DX;    TESTMUSTER SCHREIBEN
        ADD BX,DI;      NAECHSTE ADRESSE
        LOOP WR_LOOP
    
```

```

READ:  MOV BX,[BP+4];
        MOV CX,[BP+6]; RAMLAENGE AUS SS
    
```

```

RD_LOOP:
        CMP [BX],DX
        JNE RAMERROR
        ADD BX,DI
        LOOP RD_LOOP
    
```

```

INC AL
CMP AL,1H
JG M3
    
```

```

MOV IX,0AA55H;          TESTMUSTER 2
JMP WRITE

M3:  CMP AL,2
      JG  M4
      MOV IX,OFF00H;      TESTMUSTER 3
      JMP WRITE

M4:  CMP AL,3H
      JG  RUECKSPRUNG
      MOV IX,0H;          TESTMUSTER 4
      JMP WRITE

RUECKSPRUNG:
      POP BP;              ALTEN BP HERSTELLEN
      RET 2
    
```

```

RAMERROR:
      MOV AX,SEG_DATA
      MOV DS,AX
      MOV FIRMWARE,FRMVR5
      MOV AL,RAM_AUSBAU
      MOV RAMSIZE,AL

      MOV AX,[BP+8]
      OR STATUS,AX;       RAMBANK ANGEBEN

      CMP [BX],DL;        FEHLERBYTE ERMITTELN
      JNE MASKE
      INC BX
    
```

```

MASKE:  MOV CL,[BX]
         AND CL,0FH;      BITS 4...7 AUSBLENDEN
         AND DL,0FH
         JE B4_7
         OR STATUS,RL_BIT
         JMP ERROR_LOOP
    
```

```

B4_7:  OR STATUS,RH_BIT
        JMP ERROR_LOOP
    
```

```

RAMTEST  ENDP
    
```

```

SCC_INIT  PROC  NEAR
    
```

; DIE PROCEDUR INITIALISIERT DIES SCC FUER DEN LOCAL LOOP  
; DIE ADRESSE DES JEWEILIGEN KANALS (SCC\_STAT\_x) WIRD IM DX UEBERGEHEN

```

      MOV AX,WR4;          ALLG. PARAMETER
      CALL SCC

      MOV AX,WR3_0;        RECEIVER INIT
      CALL SCC

      MOV AX,WR5_0;        TRANSMITTER INIT
      CALL SCC

      MOV AX,WR11;         CLOCK SELECT
      CALL SCC

      MOV AX,WR12;         BAUDIERATE LOW-BYTE
      CALL SCC

      MOV AX,WR13;         HIGH-BYTE
      CALL SCC
    
```

75

```

MOV AX,WR14;          LOCAL LOOP, ENA BAUDR. GEN.
CALL SCC

MOV AX,WR5_1;        TRANSMITTER ENA
CALL SCC

MOV AX,WR3_1;        RECEIVER ENABLE
CALL SCC
RET
    
```

SCC\_INIT ENDP

SCC PROC NEAR

```

; DIENT ZUM INITIALISIEEN DER EINZELNEN SCC-REGISTER.
; IM AL WIRD DER REGISTER-POINTER, IM AH DER ZU LADENDE REGISTERWERT UEBE
; GEBEN. DX ENTHAELT DIE I/O-ADRESSE DES ENTSPRECHENDEN KANALS
    
```

```

OUT IX,AL;           POINTER AUF REGISTER
MOV AL,AH
OUT IX,AL;           REGISTER LADEN
RET
    
```

SCC ENDP

LOCAL\_LOOP PROC NEAR

```

; DIE ADRESSE DES JEWELIGEN DATENREGISTERS (SCC_DATA_x) WIRD IM IX
; UEBERGEHEN, DAS TESTMUSTER IN BL.
; CX ARBEITET ALS TIMEOUT-UEBERWACHUNG
    
```

```

DEC IX;              STATUSREG
IN AL,DX
AND AL,4H;           TXEMPTY
CMP AL,4H
JNE SCC_ERR

INC IX;              DATENREG
MOV AL,BL;           TESTMUSTER LADEN
OUT IX,AL
MOV CX,OFFFH;       TIMEOUT-ZAEHLER
DEC IX
    
```

```

TXRDY:  IN AL,IX;    STATUS LESEN
        AND AL,1H
        CMP AL,1H;   RXCHAR. AVAILABEL
        JE  RXRDY
        LOOP TXRDY
    
```

```

RXRDY:  INC DX;      ADRESSE DATENREGISTER
        IN AL,DX
        CMP AL,BL
        JE  SCC_END
    
```

```

SCC_ERR:
MOV AX,SCC_BIT
MOV STATUS,AX
MOV FIRMWARE,FRMVR5
MOV AL,RAM_AUSBAU
MOV RAMSIZE,AL
    
```

(16)

```
SCC_ERR1: MOV IX,SET_PROZ_MODE;      3X BLINKEN IER LED
          MOV AL,80H
          OUT IX,AL;                  1. LED AUS
          MOV CX,OFFFFH;

LED_AUS1:
          LOOP LED_AUS1
          MOV CX,OFFFFH

          MOV AL,0H
          OUT IX,AL;                  LED AN
          MOV CX,OFFFFH

LED_AN1: NOP
          NOP
          NOP
          LOOP LED_AN1

          MOV AL,80H
          OUT IX,AL;                  2. LED AUS
          MOV CX,OFFFFH;

LED_AUS2:
          LOOP LED_AUS2

          MOV AL,0H
          OUT IX,AL;                  LED AN
          MOV CX,OFFFFH

LED_AN2: NOP
          NOP
          NOP
          LOOP LED_AN2

          MOV AL,80H
          OUT IX,AL;                  3. LED AUS
          MOV CX,OFFFFH;

LED_AUS3:
          LOOP LED_AUS3

          MOV AL,0H
          OUT IX,AL;                  LED AN

SCC_ERR2:
          MOV IX,INT_REQ
          IN AX,DX
          AND AX,020H
          CMP AX,020H
          JNE SCC_ERR2
          MOV IX,SET_MBINT;          QUITTUNGS-INTERRUPT
          OUT IX,AX

SCC_ERR3:
          JMP SCC_ERR3

SCC_END:RET

LOCAL_LOOP      ENIP

          RET
```

```
BOOT ENDS
END START
CICOF6F7FBF9S
```

Produktspezifikation  
der Flachbaugruppe DUEAI  
(S 26361 - D277)

			11.10.83		
			Ullrich		
			Spezif.277		Flachbaugruppe DUEAI
			Siemens AG		
					S 26361 - D 277 - X - * - 59
2		2.12.85		U1	



**Inhalt:**

<b>1</b>	<b><u>Uebersicht</u></b>	<b>3</b>
<b>2</b>	<b><u>Kennwerte</u></b>	
2.1	Eigenschaften der seriellen Schnittstellen	4
2.2	Elektrische Eigenschaften	4
<b>3</b>	<b><u>Funktionen</u></b>	
3.1	Blockschaltbild	5
3.2	Schaltungsbeschreibung	5
3.3	Adress-Tabelle des DUE-Prozessors	10
3.3.1	Memory-Adressen	10
3.3.2	I/O-Adressen	11
3.4	Interne Register	11
<b>4</b>	<b><u>Software-Interface</u></b>	
4.1	Multibus I/O-Adressierung	
4.1.1	BADR-Register	16
4.1.2	Lower- / Upper-Reg	17
4.2	EPROM-Boot	17
4.3	Aufbau des Kommunikationsbereichs	20
4.3.1	Status-Register	21
4.3.2	Befehls-Register	22
4.3.3	Firmwareversion	22
4.3.4	Speicherausbau	23
4.3.5	A.out Format des Warmladers	23
<b>5</b>	<b><u>Hardware-Interface</u></b>	
5.1	System-Schnittstelle	24
5.2	DUE-Schnittstellen	24

**Anhang**

1.	Bauderate-Einstellungen fuer den SCC (asynchron Mode)	27
2.	Schaltereinstellungen	28
3.	Diagnose-DUART	29
4.	Firmware	30

## 1. Uebersicht

Die Flbgr. DUEAI ist eine Standard Multibus-Flachbaugruppe. An der Systemschnittstelle gelten die funktionellen und elektrischen Eigenschaften, wie sie in der "MULTIBUS SPECIFICATION" beschrieben sind.

Die Flbgr. dient der Entlastung des Hauptprozessors der 9780 Systemeinheit durch eigenstaendiges Abarbeiten von ladbaren DUE-Prozeduren. Es besteht die Moeglichkeit, wahlweise zwei X21, zwei V24 oder eine V24 und eine X21 Schnittstelle zu steuern.

Denkbare Einsatzmoeglichkeiten sind TRANSDATA-Anschluss, SNA-Anbindung (SDLC) oder Datexanschluss (HDLC).

Es ist sowohl synchrone als auch asynchrone Betriebsart moeglich.

## Literaturhinweise

- [1] Normen fuer Informationsverarbeitung, Datenkommunikation
- [2] CCITT Empfehlungen der V- und der X-Serie (Decker Verlag)
- [3] Component Data Catalog Intel Jan. 82 (Nr. 210 298-001)
- [4] Rahmenspezifikation 9780 S26361 - L6 - X - \* - 59
- [5] Spezifikation des Zilog Bausteins 8530 SCC  
(Technical Manual, April 1982  
SCC Initialization - Application Note, Sept. 1982)
- [6] Multibus-Spezifikation der Fa. Intel (Rev. 4 6.82  
Order number 9800683-04)

## 2. Kennwerte

### 2.1 Eigenschaften der seriellen Schnittstellen

- Betriebsarten: synchron oder asynchron
- max. Bitrate 1 Mbit/sek (synchron od. asynchron)
- Zeichenrahmen 5-6-7-8 Bit/Zeichen
- Zeichensicherung: Paritätsbit even od. odd moeglich.
- Blocksicherung per CRC (CCITT oder CRC-16) moeglich.
- Taktrueckgewinnung ueber Empfangsdaten moeglich
- NRZ-, NRZI-, Manchester-, FM-Decodierung moeglich
- local loop moeglich (Diagnose- bzw. Testmoeglichkeit)
- Sender oder Empfaenger sind per Befehl abschaltbar.
- Unterdrueckung von SYN im Empfangspuffer moeglich

### 2.2 Elektrische Kennwerte

Die Stromaufnahme betraegt:

+5V ca. 2,1 A
+12V ca. 0,1 A
-12V ca. 0,1 A

Die Toleranzwerte dieser Spannungen sind festgelegt

- bei +5V durch Regeln fuer TTL Technik
- bei +/- 12V durch Regeln der Schnittstelle V28

Fuer die elektrischen Kennwerte der beiden Schnittstellentypen wird auf die Definition der beiden Schnittstellen hingewiesen.

[1] und [2]

### 3. Funktionen

#### 3.1 Blockschaltbild

s. naechste Seite

#### 3.2 Schaltungsbeschreibung

Die Flbgr. arbeitet als "Slave" am Multibus und besteht im wesentlichen aus folgenden Komponenten

##### - Taktversorgung

Von zwei Quarzoszillatoren (16 MHz, 7.3728 MHz) werden alle notwendigen Taktfrequenzen abgeleitet.

16 MHz: Processorclock, Speichertiming

3.68 MHz: SCC-Systemclock

Mittels einer monostabilen Kippstufe wird der Refreshclock von 62.5 KHz erzeugt.

##### - DUE-Prozessorteil

Mikroprozessor 80188, interne Taktfrequenz 8 MHz mit integriertem Interrupt-Controller, integrierter CS-Logik sowie 3 Timer und 2 DMA-Kanaele, alle notwendigen Bussignale werden vom Prozessor erzeugt,

8 kB EPROM (Bootstrap-Lader)

##### - Dual Port Ram

Diskret aufgebauter Ram-Controller,

moegliche Speicherausbauten: 64 kB, 128 kB, 192 kB oder 256 kB, der Speicher ist parityueberwacht.

Zugriffe des DUE-Prozessors erfolgen ohne wait-states.

Das gesamte Ram oder Teilbereiche davon koennen ueber Multibus adressiert werden. Drei ueber Multibus ladbare Register legen dazu die Basisadresse des gesamten Rams, sowie die obere und untere Grenze des Dual port Fensters fest. Die Arbitration zwischen beiden Schnittstellen geschieht ueber die Prozessorsignale "HOLD" und HOLD-ACKN." Bei beiden Schnittstellen kann durch spezielle Lock-Befehle eine Busuebergabe verhindert werden (beim Multibus durch das Signal "LOCK/", bzw. beim DUE-Prozessor durch das LOCK-Praefix im Programmcode).

Die automatische Generierung des Multibus-Signals INH1/ verhindert Kollisionen, falls das Dual port Ram im Adressbereich des Hauptspeichers liegt. Die Datenbreite zum DUE-Prozessor wie auch zum Multibus betraegt 8 Bit.

### - Schnittstellenkanaele

V24:

Die Steuerleitung S1 (DEE betriebsbereit) kann fuer jeden Kanal ueber das jeweilige I/O-Reg. gesteuert werden. Die Leitung M3 (ankommender Ruf) beider Kanaele kann einen Sammelinterrupt (INT 3) beim DUE-Prozessor ausloesen und ueber den Befehl "Read I/O-Latch" gelesen werden (s. 3.4). Der SCC kann neben dem synchron Mode auch im asynchron Mode arbeiten. (Einstellungen im Anhang).

Ebene 1:

Den Bedingungen der Norm DIN 66259 wird entsprochen. Als Treiberbausteine finden 75150, als Empfaenger 75154 Verwendung. Als Versorgungsspannung dient diesen Bausteinen +/- 12V.

Logische Verknuepfungen zwischen den einzelnen Signalen werden nicht vorgenommen.

Im einzelnen koennen folgende Schnittstellenleitungen kontrolliert bzw. gelesen werden:

Leitung	Steckerpin Trapez- buchse	geht nach / kommt von
D1 (Sendedaten)	2	SCC: TxD
D2 Empfangsdaten	3	SCC: RxD
T1 Sendetakt	24	SCC: TRxC
T2 "	15	"
T4 Empfangstakt	17	SCC: RTxC
S1 Betriebsbereit	20	I/O Mode-latch
S2 Sendeteil ein	4	SCC: RTS
S4 hohe Uebertr.- geschw. ein	23	liegt fest auf "ein"
M1 Betriebsbereit	6	SCC: DCD
M2 Sendebereit	5	SCC: CTS
M3 ankomm. Ruf	22	I/O Buffer

X21:

Ueber die Modemleitungen des SCCs koennen die einzelnen Leitungen gesetzt und ueberwacht werden.

Die Zuordnung ist dabei: (bei normaler Betriebsart (s. 3.4))

c steuern	--->	SCC: RTS
t senden	--->	SCC: TxD
i melden	--->	SCC: DSR <i>110 (add)</i>
r empfangen	--->	SCC: RxD
s Schritttakt	--->	SCC: TRxC und RTxC

Die Baugruppe bietet folgende Erkennungs- und Hilfsschaltungen:

- a)  $r=0$  und  $i=aus$  ueber 16 Schritttakte: Zustand 19 (Ausloesung) geht an den DCD-Eingang des SCC und kann einen Interrupt ausloesen.
- b)  $i=ein$  ueber 16 Schritttakte: geht an SCC: CTS und kann einen Interrupt ausloesen. (jeder Zustand von  $r$  und  $i$  muss mind. 16 Takte anliegen, damit er als Dauerzustand bewertet und entsprechend reagiert werden kann).
- c) ueber einen Multiplexer kann eingestellt werden, welcher Sender die Leitungen  $t$  und  $c$  steuert.  
Folgende 4 Moeglichkeiten sind einstellbar:
  1. Zustand 24 ("DEE gestoert",  $t=0$  und  $c=aus$ ) damit werden nach einem Reset der Baugruppe die Schnittstellenleitungen definiert gesetzt.
  2. normale Betriebsart: die Leitungen  $t$  und  $c$  koennen ueber den SCC (TxD, RTS) gesteuert werden.
  3. Zustand 14 ("DEE kontrolliert nicht bereit",  $t=01010...$  und  $c=aus$ ) die Signalfolge auf der Ltg.  $t$  wird dabei synchron zu  $s$  (Schritttakt) ausgegeben.
  4. Es kann auf vorher im I/O-Latch programmierte Zustaende von  $t$  und  $c$  geschaltet werden. (Dies erlaubt eine Neuinitialisierung des SCCs nach einem Verbindungsaufbau).
- d) die Zaehleingaenge der Timer 0 und 1 sind mit den Schritttakt  $s$  der Kanale A und B verbunden und koennen damit als Zeitueberwachung fuer die Leitungen  $c$  und  $t$  eingesetzt werden. (Min. 24 Takte bei Signalaenderungen auf den Leitungen).

**Diagnosestecker:**

Als Testhilfsmittel waehrend der Phase der Softwareintegration kann ueber einen Diagnosestecker ein Babyboard mit einem DUART angeschlossen werden.

(Damit Anschlussmoeglichkeit fuer ein Terminal oder einen Drucker, s. Anhang).

**- Systemschnittstelle**

Die Flbgr. arbeitet als "Slave" am Multibus. Alle benoetigte Bussignale sind gepuffert. Der Multibus selbst wird mit max. 4 LS Eingangslasten belastet.

**- Interruptlogik**

a. DUE-Proz. --> Hauptprozessor

Der DUE-Prozessor kann ueber eine entsprechende I/O-Adresse einen Interrupt beim Hauptprozessor ausloesen. Ueber einen Dip Fix Schalter kann eine der 8 Multibus-Intltg. sel. werden.

b. Flbgr. interne Interruptmoeglichkeiten

INT 0: SCC (wird der SCC im "vector mode" betrieben, muss INT 0 des 80188 auf "cascade mode" programmiert werden! INT 2 dient dann als INTA fuer den SCC) s.[3]

INT 1: Interrupt vom Hauptprozessor

INT 3: Sammelinterrupt fuer V24-Schnittstellensignale  
(M3, s. 3.4)

NMI: Parity-Fehler

Timer 0: Zeitueberwachung fuer Kanal A

Timer 1: " " B

Timer 2: universell einsetzbar



DMA 0: Interrupt vom DMA Controller 0

DMA 1: " " 1

Der Hauptprozessor kann mittels Multibus I/O Adressen beim DUE-Prozessor einen INT 1 oder einen Reset auslösen. Reset bewirkt einen Rucksprung in den Boot-EPROM. Alle uebrigen Interrupt-Routinen muessen von der DUE-Prozedur festgelegt werden.

### 3.3 Adress-Tabelle des DUE-Prozessors

#### 3.3.1 Memory-Adressen

Die Rams werden selektiert mit LMCS (lower memory chip select) Das EPROM mit UMCS (upper memory chip select) [3]:

Multibus:			lokale Adressen
	+-----+	00000H	
			INT-Vektoren
			(1 kB)
		003FFH	
		00400H	
BADR-Reg ---->	+-----+		Programm- und
			Datenspeicher
Lower-Reg ---->	*****	Beispiel	
	*****	eines	
Upper-Reg ---->	*****	Dual-port Fensters	
	+-----+	0FFFFH	Ram-Ausbau 64 kB
		1FFFFH	" 128 kB
		2FFFFH	" 192 kB
		3FFFFH	" 256 kB
	+-----+	FE000H	
			Boot-EPROM
			(8 kB)
	+-----+	FFFFFFH	

### 3.3.2 I/O - Adressen

Alle I/O Peripherals werden mit den PCS-Leitungen (peripheral chip select) angesprochen. Die Basisadresse wird innerhalb der Initialisierung auf 0 festgelegt (s. [3]).

PCS 0:	000H	Status/CMD Kanal B	\	
	001H	Daten-Reg "		SCC
	002H	Status/CMD Kanal A	>	8530
	003H	Daten-Reg "		
			/	
PCS 1:	080H	Reset INT 1 (Interrupt vom Multibus)		
PCS 2:	100H	Reset INT 3 (V24-Meldeltg. M3)		
PCS 3:	180H	Set Multibus-Interrupt		
PCS 4:	200H	Set Prozessor Mode-Reg.		
	210H	Set I/O Reg. Kanal A		
	220H	" " B		
	230H	Read I/O Latch		

ueber die I/O-Adressen 280H ... 28FH kann ein aufsteckbares Babyboard mit einem Diagnose-DUART angesprochen werden.

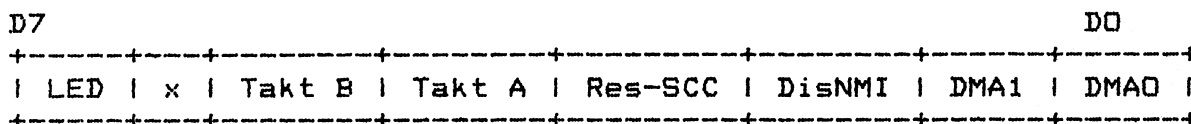
(s. Anhang)

### 3.4 Interne Register

Eine detaillierte Beschreibung der einzelnen Register des SCC (Zilog 8530) ist dem Datenbuch [5] zu entnehmen.

Nachfolgend sind deshalb nur die internen Register beschrieben.

a. Set Prozessor Mode-Register (200H)



Ueber die beiden Bits DMA 0 und DMA 1 kann selektiert werden, welcher der beiden Kanäle A und B im DMA-Mode betrieben werden soll.

DMA 1	DMA 0	Betriebsart
0	0	Die SCC-Kanäle A und B werden halbduplex betrieben (W/REQA ---> DREQ0 W/REQB ---> DREQ1)
0	1	SCC-Kanal A wird vollduplex betrieben (W/REQA ---> DREQ0 DTR/REQA ---> DREQ1)
1	0	SCC-Kanal B wird vollduplex betrieben (W/REQB ---> DREQ0 DTR/REQB ---> DREQ1)
1	1	DMA disabled
DisNMI: 0		---> NMI enable
1		---> NMI disabled
Res-SCC: 0		---> Reset fuer SCC
1		---> normaler Betrieb

Takt A/B: ermöglicht bei den V24-Schnittstellen neben der ext. Taktzufuehrung (T2 und T4) auch eine Eigentaktung (T1).

0 ---> T2 und T4 werden dem SCC zugefuehrt

T2 ---> Pin 14 (TRxCA) fuer Kanal A

Pin 26 (TRxCB) " B

T4 ---> Pin 12 (RTxCA) " A

Pin 28 (RTxCB) " B

1 ---> die externe Taktzufuehrung fuer T2 wird disabled. Der Sendetakt kann (bei entsprechender Programmierung des SCCs) vom SCC selbst erzeugt und als T1 an die jeweilige V24 Schnittstelle geschaltet werden.

Pin 14 (TRxCA) ---> T1 fuer Kanal A

Pin 26 (TRxCB) ---> " B

(T4 wird davon nicht beruehrt)

! Achtung: Bevor die SCC-Pins 14 od. 26 als Taktausgang  
! programmiert werden, ist es zwingend erforderlich,  
! das zuerst die Bits "Takt A/B" im Prozessor Mode-Register gesetzt werden.  
!

Die einzelnen Bits koennen ueber den Befehl "Read I/O Latch" rueckgelesen werden.

(Diese Betriebsart ist bei den Prototyp-Flbgr. (ersichtlich durch den Aufdruck W26361-D277-24-01-5 auf der Bauteileseite nicht moeglich!)

LED: Steuert die rote Diagnose-LED

0 ---> LED an

1 ---> LED aus

Das Prozessor Mode-Register kann nur beschrieben werden!

b. Set I/O Mode-Register Kanal A/B (210H / 220H)

D7								D0
V24	ENA 1	ENA 0	t	c	MODE 1	MODE 0	S1	

Beide Register besitzen gleiches Datenformat

S1: steuert das Schnittstellensignal S1 der V24-Schnittstelle

0 ---> S1 aus

1 ---> S1 ein

MODE 0 MODE 1 steuert die X21-Schnittstelle

0 0 Zustand 24 (DEE gestoert, wird nach Reset ausgegeben)

0 1 normaler Betriebsmode (SCC steuert t und c)

1 0 latched mode: die im I/O Mode-Reg. fuer t und c angegebenen Werte werden ausgegeben.

t = 0/1 ---> X21: t = 0/1

c = 0/1 ---> X21: c = aus/ein

1 1 Zustand 14 (DEE kontrolliert nicht bereit, t = 01010..., c = aus) wird ausgegeben.

ENA 0: steuert bei V24 die Interruptmoeglichkeit von M3  
(ankommender Ruf)

steuert bei X21 die Erkennungsschaltung fuer Zustand  
19 (Ausloesung)

0 ---> disable und clear

1 ---> enable

ENA 1: steuert bei X21 die Erkennungsschaltung "i ein" ueber  
16 Schritttakte

0 ----> disable und clear

1 ----> enable

V24: steuert den Schnittstellen-Multiplexer V24 - X21

0 ----> X21 selektiert

1 ----> V24 "

Die I/O Mode-Register koennen nur beschrieben werden!

c. Read I/O Latch (230H)

D7				D0			
TAKT A	TAKT B	B.I	A.I	M3-B	M3-B	M3-A	M3-A
					INT-FF		INT-FF

Die einzelnen Bits geben die Zustaende der Meldeleitungen M3  
der beiden V24-Schnittstellen wieder

M3-A: Leitung M3 fuer Kanal A

M3-B: " " B

0 ----> aus

1 ----> ein

M3-A INT-FF, M3-B INT-FF: gibt den Zustand des jeweiligen  
Interrupt Flip-flops an.

(Eine Zustandsaenderung von "aus" nach "ein" setzt das ent-  
sprechende Flip-flop)

0 ----> Grundzustand

1 ----> Leitungsaenderung erkannt

Ueber die I/O-Adressierung 100H kann dieser INT wieder rueck-  
gesetzt werden.

A.I / B.I: Geben den Zustand auf den X21-Schnittstellenleitungen

I (melden) fuer den jeweiligen Kanal A bzw. B an

0 ---> aus

1 ---> ein

TAKT A/B: Die ueber das Prozessor Mode-Reg. programmierten

Taktmodi koennen rueckgelesen werden

0 ---> extern Takt

1 ---> disable extern Takt

#### 4. Software-Interface

##### 4.1 Multibus I/O-Adressierung

Die Basisadresse kann mit einem 8 pol. Dip Fix in Schritten  
von 4 kByte eingestellt werden. (Einstellung: s. Anhang)

Nachfolgend sind die einzelnen Register mit ihren Offsetwerten  
zur eingestellten Basisadresse angegeben.

001H: Enable Dual-port Fenster

002H: Reset Multibus Interrupt Flipflop

003H: Set INT 1 (Interrupt an DUE-Prozessor)

004H: Set Reset (bewirkt Ruecksprung in EPROM-Boot)

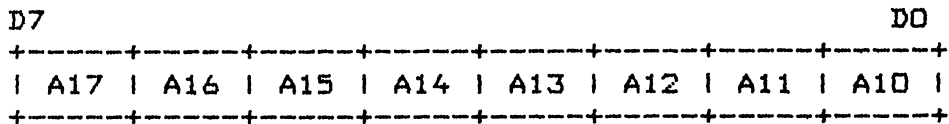
005H: Set Basisadresse (BADR-Reg)

006H: Set Upper-Register

007H: Set Lower-Register

008H .... 0FFH sind reserviert und duerfen nicht angesprochen  
werden.

4.1.1 BADR-Register



Im BADR-Reg wird die Multibus-Basisadresse des Dual port Segments angegeben. Mit A17 ... A10 koennen die Multibusadressen ADR 17/ ... ADR 10/ festgelegt werden.

Das Dual port Segment kann damit innerhalb eines Adressbereiches von 16 MB bei einer Schrittweite von 64 kB beliebig verschoben werden. Sollte der DUE-Prozessor in einer 9780-Systemeinheit ohne P2-Platter arbeiten, muessen die Bits A14 ... A17 auf 0 gesetzt sein.

Beispiele zur Programmierung des Dual port Segments:

Multibus-Adresse	Registerwert
000000H	00H
010000H	01H
020000H	02H
040000H	04H
080000H	08H
100000H	10H etc.

4.1.2 Lower-Reg. / Upper-Reg.



Mit den Registern koennen die Grenzen des Dual port Fensters innerhalb des Dual port Segments festgelegt werden.

AB ... AF entsprechen dabei den Multibus-Adressbits ADR B/ bis ADR F/.



Mit SEL 0 und SEL 1 kann festgelegt werden, auf welche Rambank des DUE-Prozessors zugegriffen werden soll.

SEL 1	SEL 0	Rambank	lokaler Adressbereich
0	0	0	00000H ... 0FFFFH
0	1	1	10000H ... 10000H
1	0	2	20000H ... 2FFFFH
1	1	3	30000H ... 3FFFFH

Ein Zugriff in das Ram wird gestattet bei Erfuellen der folgenden beiden Bedingungen:

1. [Lower-Reg] <= Multibus-Adr. <= [Upper-Reg]  
 [AB ... AF] AB ... AF [AB ... AF]
2. [BADR-Reg] <= Multibus-Adr. < [BADR-Reg] + 64 kB

Die minimale Fensterbreite bzw. Schrittweite ist 2kB:

$$[\text{Lower-Reg}] = [\text{Upper-Reg}]$$

Die maximale Fensterbreite ist 64 kB:

[Lower-Reg] = 00H,	[Upper-Reg] = 1FH	(Rambank 0)
[ " ] = 20H,	[ " ] = 3FH	" 1
[ " ] = 40H,	[ " ] = 5FH	" 2
[ " ] = 60H,	[ " ] = 7FH	" 3

Das Fenster kann geschlossen werden, wenn:

$$[\text{Lower-Reg}] > [\text{Upper-Reg}]$$

$$[\text{AB ... AF}] \quad [\text{AB ... AF}]$$

## Beispiele zur Programmierung der Fenstergrenzen

Offset zur Segment-Adr	Reg. Wert			
	Rambank 0	Rambank 1	Rambank 2	Rambank 3
00000H	00H	20H	40H	60H
00800H	01H	21H	41H	61H
01000H	02H	22H	42H	62H
01800H	03H	23H	43H	63H
04000H	08H	28H	48H	68H
06000H	0CH	2CH	4CH	6CH
0A000H	14H	34H	54H	74H
0F800H	1FH	3FH	5FH	7F

Beispiel: Das Dual port Fenster soll unter den Multibusadressen  
0FB000H bis 0FBFFFH initialisiert werden.

Selektiert werden soll die lokale Rambank 0 des DUE-  
Prozessors:

[BADR-Reg] : 0FH

[Lower-Reg]: 16H

[Upper-Reg]: 17H

Nachdem die einzelnen Register geladen worden sind, muss mit  
einem Schreibzyklus auf die I/O-Adresse 01H das Dual port  
Fenster enabled werden (s. 4.1)

#### 4.2 EPR0M-Boot

Nch einem Hardware- bzw. Software-Reset laeuft automatisch der  
EPR0M-Boot an. Er initialisiert den Kontroll-Block des uP 80188  
und fuehrt einen ROM-Test, einen Ram-Test und einen SCC-Test

durch. Das Testergebnis, der Ramausbau und die Firmwareversion werden in den jeweiligen Registern angezeigt.

Die Dauer der Initialisierung ist abhaengig vom Speicherausbau und dauert bei 256 kB Ram ca. 6s.

Danach wird auf einen Interrupt vom System gewartet, der dann durch einen Quittungsinterrupt seitens des DUE-Prozessors beantwortet wird. (Der genaue Ablauf der Selbsttests ist im Anhang angegeben)

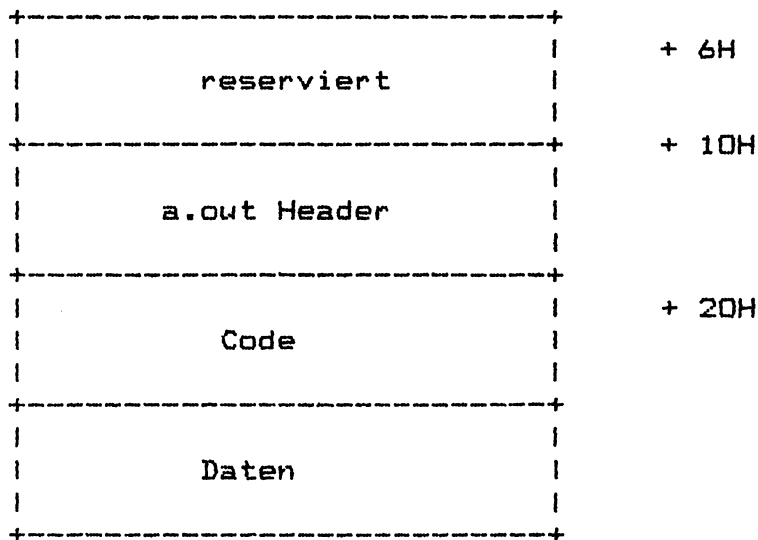
Der Hauptprozessor oeffnet ab der lokalen Adresse B000H ein Fenster im Ram, laedt den Warmlader, setzt im Befehls-Register das Bit "AOUTSTART" und erzeugt einen Interrupt.

Anhand des a.out Headers ermittelt sich der DUE-Prozessor die Startadresse, setzt die Segmentregister DS, SS und CS auf den Codeanfang (Fensteradresse + 20H) bzw. ES auf das Fenster selbst und startet den Warmlader.

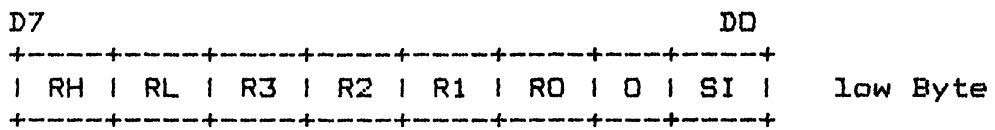
#### 4.3 Aufbau des Kommunikationsbereich

Der Koomunikationsbereich liegt innerhalb des Dual port Fensters und befindet sich vor Prozeduruebernahme ab der Adresse B000H (lokale Adresse; entspricht den Multibusadresse FB000H. Die obere Grenze des Fensters kann beliebig gewaehlt werden.

+-----+		
Status-Reg. (low Byte)		Fensteradr.
+-----+		
Status-Reg. (high Byte)		+ 1H
+-----+		
Befehls-Register		+ 2H
+-----+		
reserviert		+ 3H
+-----+		
Firmware-Version		+ 4H
+-----+		
Speicherausbau		+ 5H
+-----+		



#### 4.3.1 Statusregister



ROM: ROM-Fehler

kennzeichnet einen waehrend des Romtest erkannten Fehler

RO ... R3, RAM, PAR: RAM-Fehler, Parity-Fehler

kenzeichnet einen waehrend des Ramtest erkannten Fehler

Der Index gibt dabei die fehlerhafte Ram-Bank

RL: Der Ram-Fehler liegt im Low-Ram (Bit 0 ... 3)

RH: " High-Ram (Bit 4 ... 7)

SCC: kennzeichnet einen waehrend des SCC-Test erkannten Fehler

ERR: Error, wird gesetzt, wenn einer der vorher aufgefuehrten Fehler erkannt wurde

SI: System initialisiert  
kennzeichnet den Abschluss der Initialisierungs- und Test-  
routinen.  
Sollte waehrend der Testprogramme einer der aufgefuehrten Fehler erkannt worden sein, wird der Warmlader nicht gestartet.

#### 4.3.2 Befehlsregister

(Fensteradr. + 2H)

```
D7                                     D0
+---+---+---+---+---+---+---+---+
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | AOUTSTART |
+---+---+---+---+---+---+---+---+
```

AOUTSTART: signalisiert dem DUE-Prozessor, dass der Warmlader vom Hauptprozessor ins Ram geladen wurde und gestartet werden kann.

Dem Befehl muss ein INT 1 folgen.

#### 4.3.3 Firmwareversion

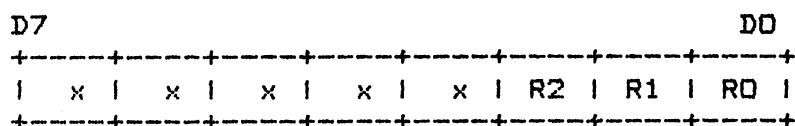
(Fensteradr. + 4H)

```
D7                                     D0
+---+---+---+---+---+---+---+---+
| S7 | S6 | S5 | S4 | S3 | S2 | S1 | S0 |
+---+---+---+---+---+---+---+---+
```

Die Bits S0 bis S7 geben den auf der Baugruppe befindliche Firmwareversion an.

#### 4.3.4 Speicherausbau

(Fensteradr. + 5H)



Die Bits R0 bis R3 geben den auf der Baugruppe befindlichen Speicherausbau an

R2	R1	R0	
0	0	1	64 kB
0	1	0	128 kB
0	1	1	192 kB
1	0	0	256 kB

#### 4.3.5 A.out Format des Warmladers

Der Header des Warmladers beginnt bei Fensteradresse + 10H und hat folgende Form:

```

struct header {
    int      a_magic      /* magic number          */
    unsigned a_text      /* Laenge des Code-Segments
    unsigned a_data      /* Laenge der init. Daten
    unsigned a_bss       /* Laenge der nicht init. Daten
    unsigned a_syms      /* hier 0
    unsigned a_entry     /* Startadresse rel. zu a_text
    unsigned a_unused    /* hier 0
    unsigned a_flag      /* hier immer 1
};
    
```

## 5 Hardware-Interface

### 5.1 Systemschnittstelle

Die Systemschnittstelle ist multibuskompatibel.

Die Flbgr. selbst arbeitet als "Slave" am Bus.

Alle Bussignale, sowie das Bus-Timing sind in der Multibus-Spezifikation (Intel) beschrieben.

### 5.2 DUE-Schnittstellen

Die Verbindung zwischen der Flbgr. und dem Anschlussfeld der 9780 Systemeinheit geschieht ueber spezielle Flachbandleitungen.

Die Zuordnung der verschiedenen Stecker zu den einzelnen Kanaelen ist dabei:

V24 Kanal A ----> P4 Stecker

V24 Kanal B ----> P6 Stecker

X21 Kanal A ----> P5 Stecker

X21 Kanal B ----> P7 Stecker

Nachfolgend ist die Schnittstellenbelegung der Flachbaugruppensteckverbinder und der zugehoerigen Trapez-Buchsen im Anschlussfeld aufgefuehrt.

Schnittstelle V24

Steck-Mechanik: 22 pol. cod. Modulstecker

Signaldefinition: gem. DIN 66020 Teil 1

Modul	AMP	Signal-Name	Bemerkungen
01			codiert
02	02	D1 TxD 103	Sendedaten
03	03	D2 RxD 104	Empfangsdaten
04	04	S2 RTS 105	Sendeteil einschalten
05	05	M2 CTS 106	Sendebereitschaft DUEE
06	06	M1 DSR 107	Betriebsbereitschaft DUEE
07	07	E2 GND 102	Signalerde
08	-	-	-
09	23	S4 111	hohe Ueb.-geschw. ein
10	24	T1 113	
11	-	-	
12	-	-	
13	-	-	
14	15	T2 114	Sendeschriftakt
15	-	-	
16	17	T4 115	Empfangsschriftakt
17	-	-	
18	-	-	
19	20	S1.2 DTR 108/2	DEE betriebsbereit
20	22	M3 125	ankommender Ruf
21	-	-	codiert
22	-	-	

Beide V24-Schnittstellen haben die gleiche Steckerbelegung

Alle nicht aufgefuehrten V24-Schnittstellensignale werden nicht verwendet bzw. nicht gesteuert.



Schnittstelle X21

Steck-Mechanik: 14 pol. cod. Modulstecker

Signaldefinition: gem. DIN 66020 Teil 2

Beide X21 - Schnittstellen haben identische Steckerbelegungen

Modul	AMP	Signal-Name	Bemerkungen
01	-		codiert
08	08	G	Signalerde, verb. mit 0 V
06	02	T(A)	Senden
14	09	T(B)	"
12	03	C(A)	Steuern
05	10	C(B)	"
11	04	R(A)	Empfangen
04	11	R(B)	"
10	05	I(A)	melden
03	12	I(B)	"
09	06	S(A)	Schrittakt
02	13	S(B)	"

Anhang

Bauderate-Einstellungen fuer den SCC

(fuer asynchrone Betriebsart)

Voreinstellung SCC:

- WR-Reg 4:           x16 clock
- WR-Reg 11:        Transmit-clock = BR-Generator output
- Receive-clock =                   "
- WR-Reg 14:        BR-Generator enable
- BR-Generator source = SCC-PCLK (3.6864 MHz)
- WR-Reg 12:                 Bauderate-Einstellung
- WR-Reg 13:

Unten aufgefuehrte Tabelle gibt die Werte fuer die gaengigsten Uebertragungsgeschwindigkeiten an.

Daneben laesst sich der Wert fuer jede beliebige Bauderate nach folgender Formel berechnen (x16 clock!)

$$x = \frac{115313}{\text{Bauderate}} - 2$$

Dabei ist x der Wert fuer WR-Reg 12/13; die Bauderate wird in Bit/s angegeben.

<u>Count value</u>	<u>Bauderate [Bit/s]</u>
1	38400
4	19200
10	9600
22	4800
46	2400
94	1200
190	600
382	300

## 2.1 Schaltereinstellungen (Vorserien- und Serienflgar.)

### a. Multibus I/O-Adresse

Einstellung mit den Schaltern S2 ... S9

S 9	S 8	S 7	S 6	S 5	S 4	S 3	S 2	
AF/	AE/	AD/	AC/	AB/	AA/	A9/	A8/	Multibusadressen
op	op	op	clo	clo	op	op	clo	= 1900H ... 19FFH

Bei Auslieferung der Baugruppe sind S2, S5 und S6 geschlossen.

Die Baugruppe ist damit auf die I/O-Adresse 1900H eingestellt.

### b. Multibusinterrupt

Einstellung mit den Schaltern S10 ... S17

S17	S16	S15	S14	S13	S12	S11	S10	
7	6	5	4	3	2	1	0	Multibus-
op	op	clo	op	op	op	op	op	INT-Ltgen.

Bei Auslieferung der Baugruppe ist S16 geschlossen und

erzeugt damit einen INT 5/.

Die Schalter S1 und S18 dienen ausschliesslich prueftechn.

Zwecken und muessen im Normalbetrieb immer geschlossen sein.

2.2 Schaltereinstellungen (Prototypen)

(Prototypen sind erkenntlich am Leiterplattenaufdruck  
"W26361-D277-Z4-01-5" auf der Bauteileseite.)

a. Multibus I/O-Adresse

Einstellung mit den Schaltern S9 ... S16

S16	S15	S14	S13	S12	S11	S10	S 9	
AF/	AE/	AD/	AC/	AB/	AA/	A9/	A8/	Multibusadressen
op	op	op	clo	clo	op	op	clo	= 1900H ... 19FFH

b. Multibusinterrupt

Einstellung mit den Schaltern S 1 ... S 8

S 8	S 7	S 6	S 5	S 4	S 3	S 2	S 1	
7	6	5	4	3	2	1	0	Multibus-
op	op	clo	op	op	op	op	op	INT-Ltgen.

### 3. Diagnose-DUART

Zum Testen von geladenen Programmen kann ueber den Diagnosestecker P3 ein Babyboard mit einem DUART 2681 (Signetics) angeschlossen werden.

Am seriellen Ausgang des Babyboards koennen Geraete mit SS 97 angeschlossen werden.

Unter den folgenden I/O-Adressen koennen die einzelnen Register des DUARTs angesprochen werden.

I/O-Adr.	Lesen	Schreiben
280H	Mode-Reg 1/2 Kanal A	Mode-Reg 1/2 Kanal A
281H	Status-Reg. A	Clock-sel. A
282H	----	Command-Reg. A
283H	Receive-Reg. A	Transmit-Reg A
284H	----	Aux. control Reg

Die nicht aufgefuehrten Register werden nicht benoetigt bzw. von der Hardware nicht unterstuetzt.

Der DUART wird mit einem externen Clock von 3.6864 MHz betrieben.

Die Spannungsversorgung des Boards geschieht ueber den P3 Stecker vom DUE-Prozessor aus.

#### 4. Firmware

Nach einem Hard- bzw. Softwarereset wird in den Eprom-Boot gesprungen. Gleichzeitig wird die Diagnose-Led aktiviert.

1. Initialisierung des internen Kontrollblocks:

Saemtliche CS-Signale definieren, Int-Contr. initialisieren, alle Interrupts sperren, alle INT-Flipflops ruecksetzen.

2. Romtest:

Addieren des Rominhalts und Vergleich mit dem in der letzten Zelle stehenden Checkword.

Falls Uebereinstimmung: Sprung zum Ramtest

Falls Fehler: Led wird kurzzeitig ausgeschaltet (ca.300 ms)

Im Statusregister wird eine Fehlermeldung hinterlegt, danach wird die Initialisierung abgebrochen (Sprung nach 5.)

3. Ramtest:

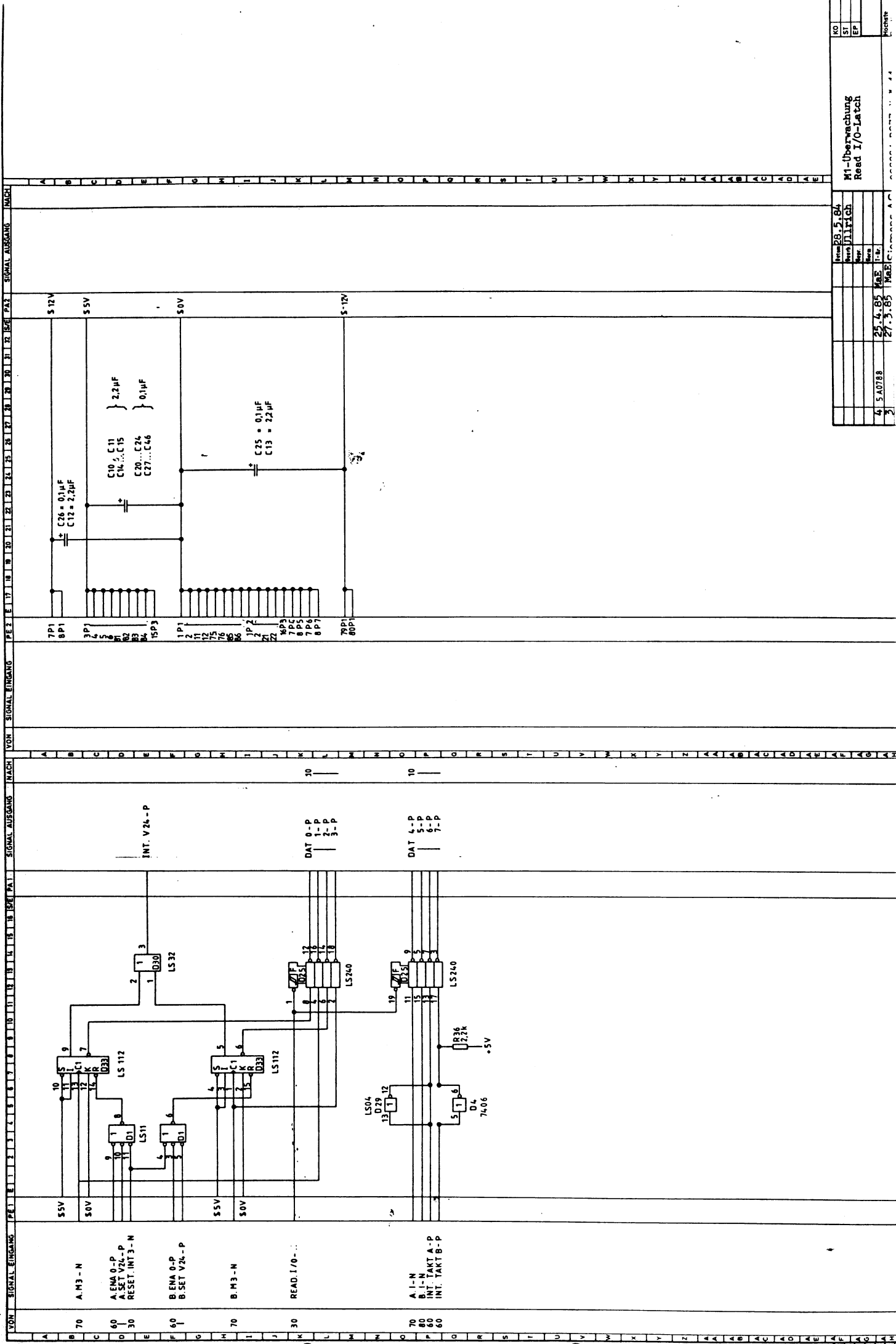
Test der untersten 32 Bytes (dienen spaeter als Stack)

Test der Rambank 0

Danach wird der Speicherausbau ermittelt und die vorhandenen Speicherbaenke getestet.

Falls Fehler: Led wird 2x kurzzeitig ausgeschaltet

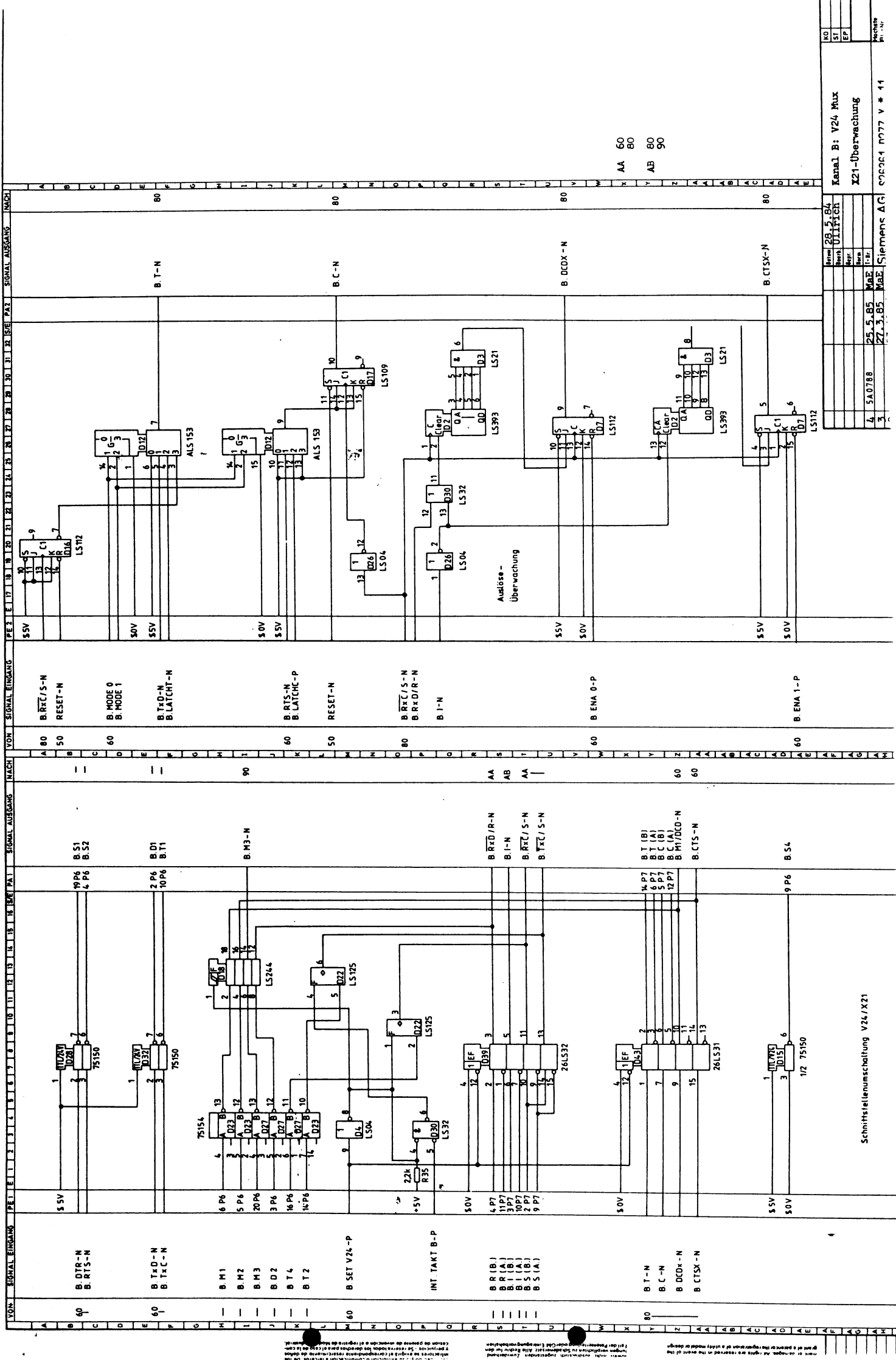
und im Status-Register eine Fehlermeldung hinterlegt. Die Initialisierung wird danach abgebrochen (Sprung nach 5.)



NO	28.5.84
ST	J. Ulrich
EP	
MAE	25.4.85
MAE	27.3.85

1	5 A 0788	25.4.85	MAE
2		27.3.85	MAE

M1-Überwachung  
Read I/O-Latch



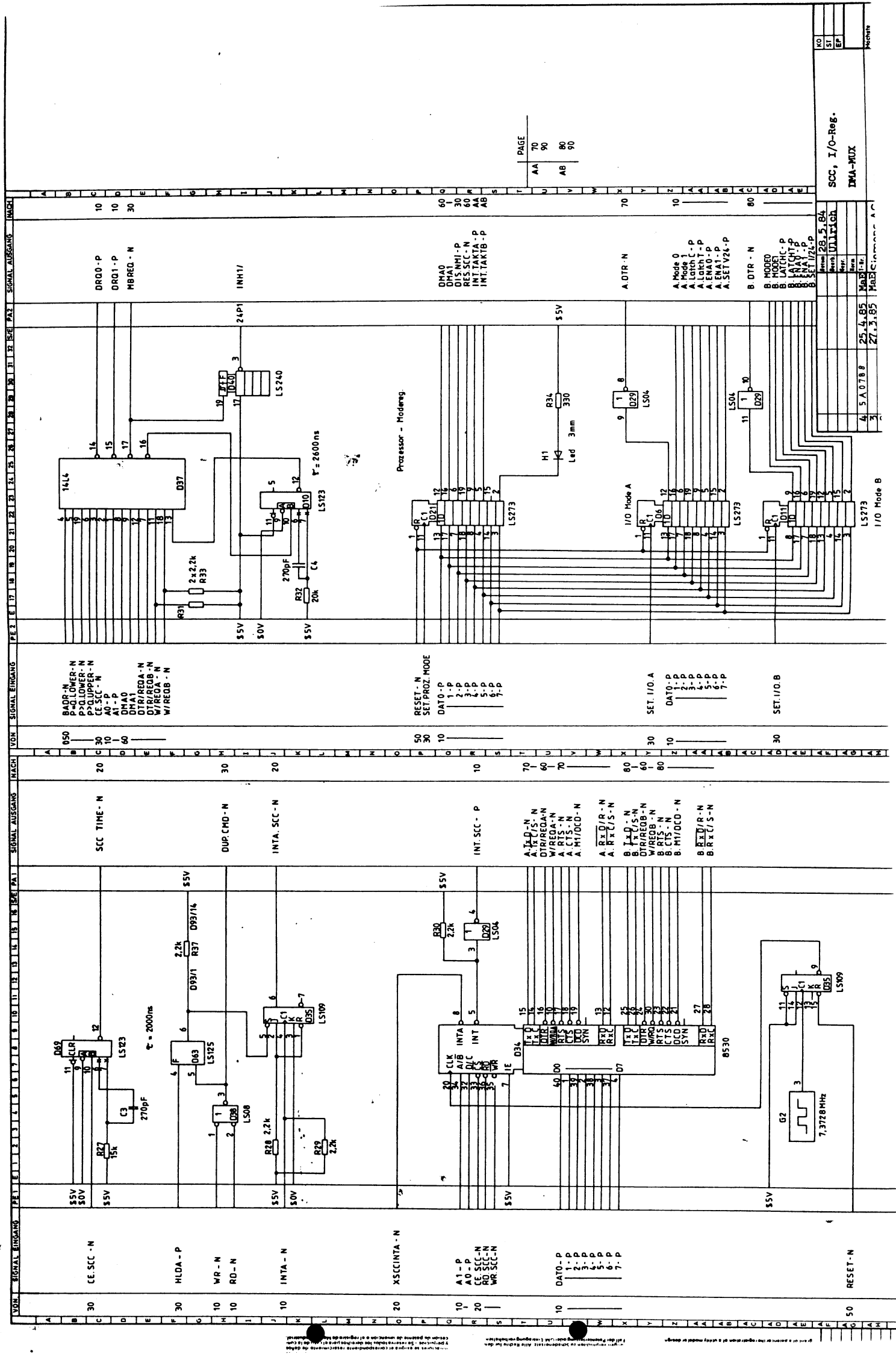
NO	REV	DATE	BY	CHKD	DESCRIPTION
1					
2					
3					
4	5A0788	25.5.85	M&E	T-F.	
5	27.3.85	M&E			

Form	28+5.84
Best.	ULLI FICH
Verf.	
Best.	
Verf.	
Best.	
Verf.	

Kanal B: V24 Mux  
 X21-Überwachung  
 Siemens AG 67694 1077 V \* 11  
 Schmittstufenschaltung V24/X21







VON: SIGNAL EINGANG PE 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50

SIGNAL AUSGANG INACH

PE 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50

CE SEC - N  
HILDA - P  
WR - N  
RD - N  
INTA - N  
XSXCINTA - N  
A1 - P  
A0 - P  
CE SEC - N  
RD SEC - N  
WR SEC - N  
DAT0 - P  
1 - P  
2 - P  
3 - P  
4 - P  
5 - P  
6 - P  
7 - P  
RESET - N

SCC TIME - N  
DUP CHD - N  
INTA. SCC - N  
INT. SCC - P  
A1/D - N  
A1/C/S - N  
DTR/REQA - N  
W/REQA - N  
A. RTS - N  
A. CTS - N  
A. M1/OCD - N  
A. RXD/R - N  
A. RxC/S - N  
B. TXD - N  
B. TXC/S - N  
B. WREQB - N  
B. RTS - N  
B. CTS - N  
B. M1/OCD - N  
B. RXD/R - N  
B. RxC/S - N

BADR - N  
P-LOWER - N  
P-UPPER - N  
CE. SEC - N  
A1 - P  
DMA0  
DTR/REDA - N  
DTR/REDB - N  
W/REDA - N  
W/REDB - N

RESET - N  
SET. PROZ. MODE  
DAT0 - P  
1 - P  
2 - P  
3 - P  
4 - P  
5 - P  
6 - P  
7 - P

SET. I/O. A  
DAT0 - P  
1 - P  
2 - P  
3 - P  
4 - P  
5 - P  
6 - P  
7 - P

SET. I/O. B

DRQ0 - P  
DRQ1 - P  
MBRED - N  
INNH17  
DMA0  
DMA1  
DIS. MHI - P  
RES. SCC - N  
INT. TAKTA - P  
INT. TAKTB - P

A. DTR - N  
A. Mode 0  
A. Mode 1  
A. Latch 1 - P  
A. EMAL0 - P  
A. EMAL1 - P  
A. SETV24 - P  
B. DTR - N  
B. MODE0  
B. MODE1  
B. LATCH - P  
B. LATCH1 - P  
B. EMAL0 - P  
B. EMAL1 - P  
B. SETV24 - P

SCC, I/O-Reg.  
DMA-MUX

AA 70  
AB 80  
AB 90

PAGE

AA 70  
AB 80  
AB 90

AA 70  
AB 80  
AB 90

AA 70  
AB 80  
AB 90

AA 70  
AB 80  
AB 90

AA 70  
AB 80  
AB 90

AA 70  
AB 80  
AB 90

AA 70  
AB 80  
AB 90

AA 70  
AB 80  
AB 90

AA 70  
AB 80  
AB 90

AA 70  
AB 80  
AB 90

AA 70  
AB 80  
AB 90

AA 70  
AB 80  
AB 90

AA 70  
AB 80  
AB 90

AA 70  
AB 80  
AB 90

AA 70  
AB 80  
AB 90

AA 70  
AB 80  
AB 90

AA 70  
AB 80  
AB 90

AA 70  
AB 80  
AB 90

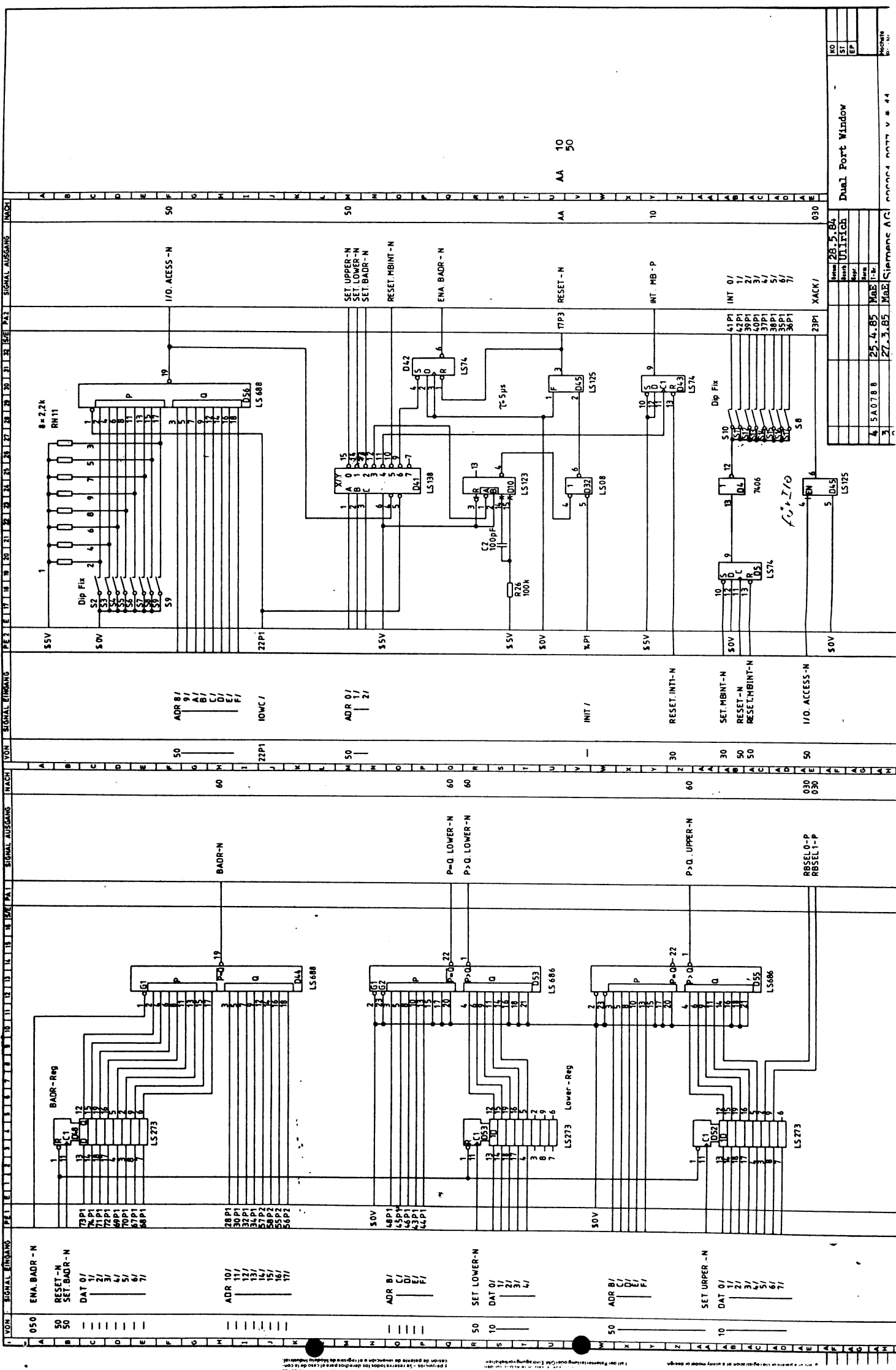
AA 70  
AB 80  
AB 90

AA 70  
AB 80  
AB 90

AA 70  
AB 80  
AB 90

AA 70  
AB 80  
AB 90

AA 70  
AB 80  
AB 90



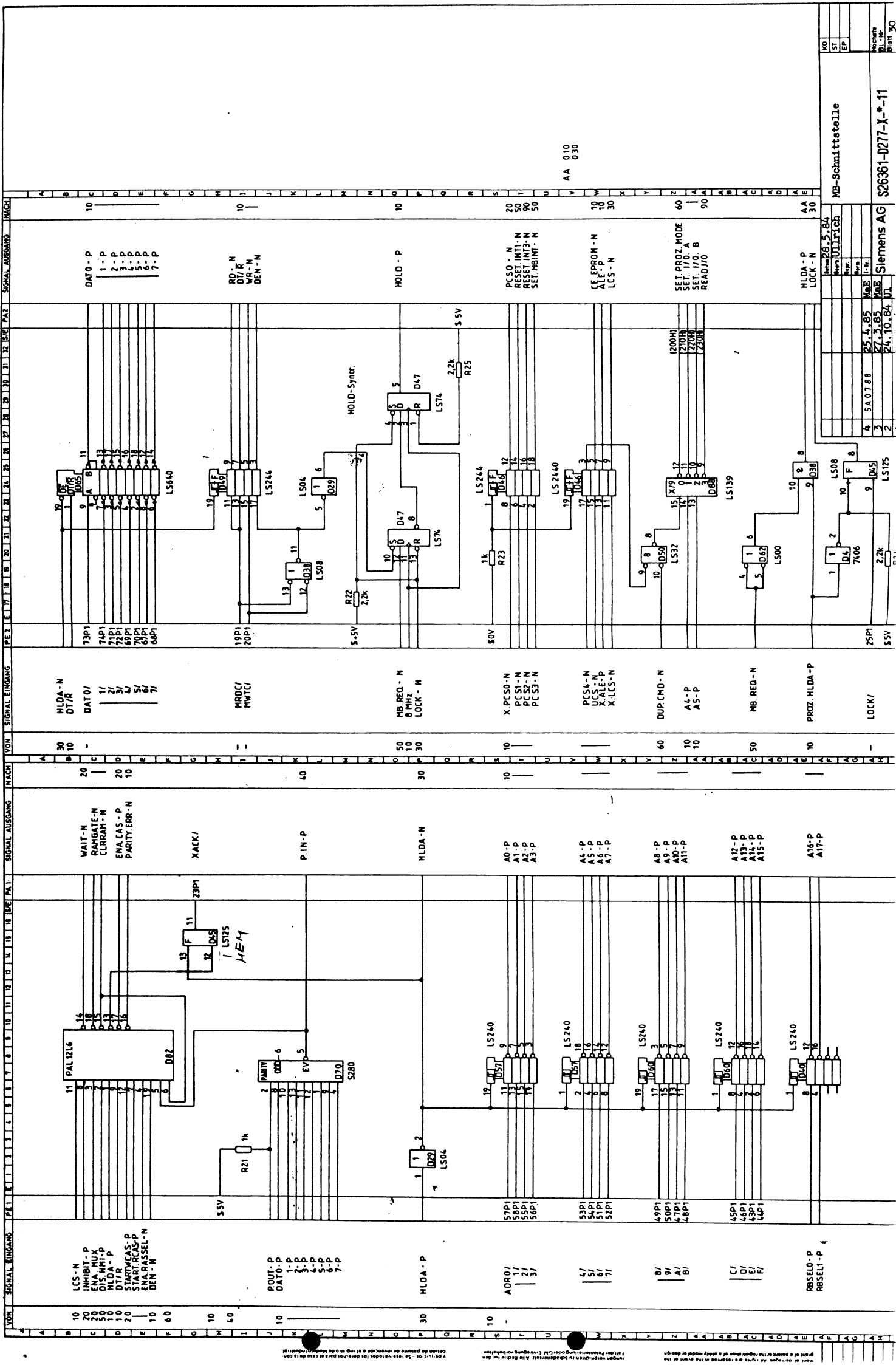
NO	ST	EP	MEMORIE
1			
2			
3			
4	SA0788	25-4.85	M&B
5		27-3.85	M&B
Siemens AG			

Item	28.5.84	UTILITÄT
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		
50		
51		
52		
53		
54		
55		
56		
57		
58		
59		
60		
61		
62		
63		
64		
65		
66		
67		
68		
69		
70		
71		
72		
73		
74		
75		
76		
77		
78		
79		
80		
81		
82		
83		
84		
85		
86		
87		
88		
89		
90		
91		
92		
93		
94		
95		
96		
97		
98		
99		
100		

Dual Port Window  
 AA 10  
 50





NO	ST	EP	Rechnung	Blatt
				30

NO	ST	EP	Rechnung	Blatt
1			Siemens AG	50
2				
3				
4				

NO	ST	EP	Rechnung	Blatt
1			Siemens AG	50
2				
3				
4				

NO	ST	EP	Rechnung	Blatt
1			Siemens AG	50
2				
3				
4				

NO	ST	EP	Rechnung	Blatt
1			Siemens AG	50
2				
3				
4				

NO	ST	EP	Rechnung	Blatt
1			Siemens AG	50
2				
3				
4				

NO	ST	EP	Rechnung	Blatt
1			Siemens AG	50
2				
3				
4				

NO	ST	EP	Rechnung	Blatt
1			Siemens AG	50
2				
3				
4				

NO	ST	EP	Rechnung	Blatt
1			Siemens AG	50
2				
3				
4				

NO	ST	EP	Rechnung	Blatt
1			Siemens AG	50
2				
3				
4				

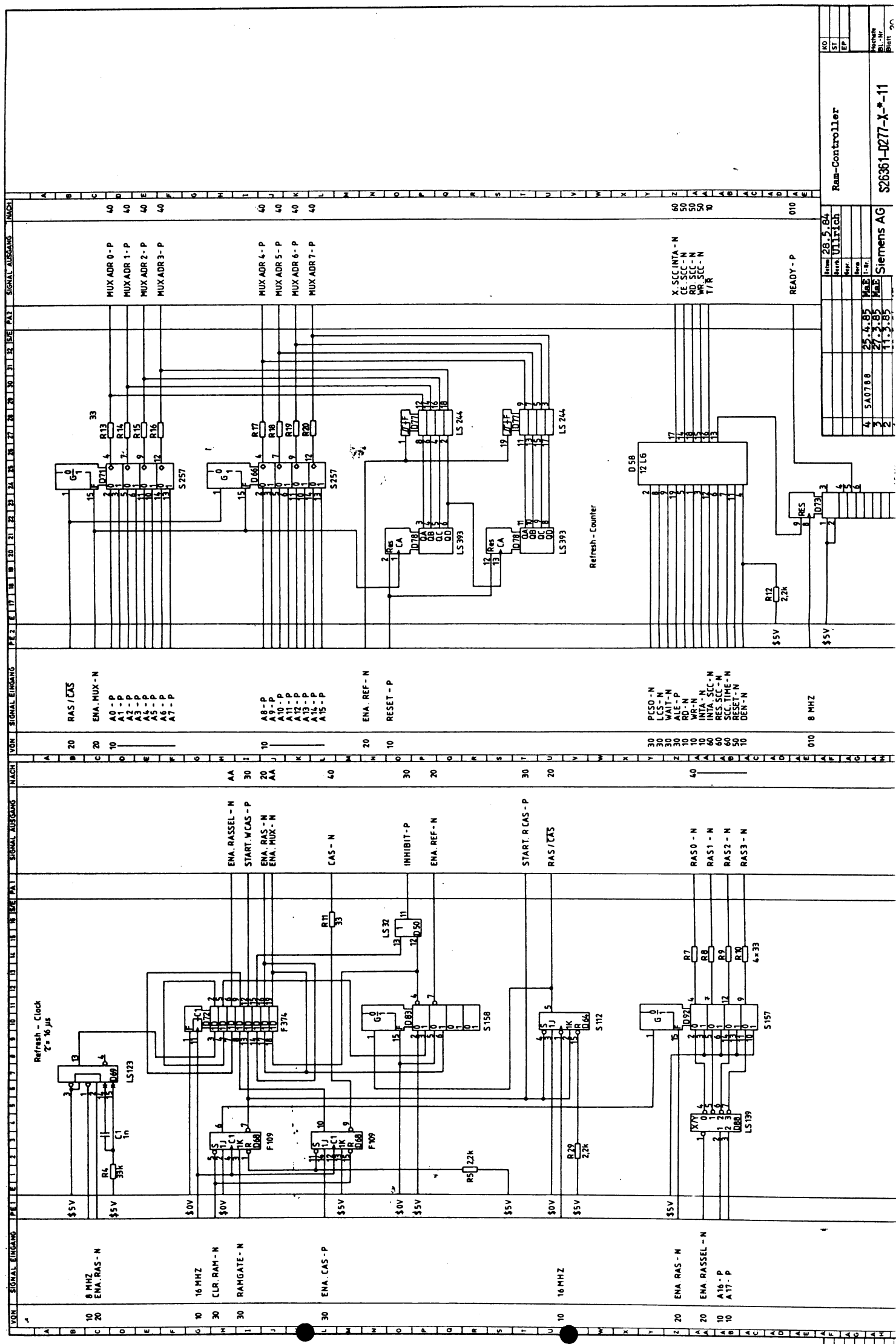
NO	ST	EP	Rechnung	Blatt
				30
1			Siemens AG	50
2				
3				
4				

NO	ST	EP	Rechnung	Blatt
				30
1			Siemens AG	50
2				
3				
4				

NO	ST	EP	Rechnung	Blatt
				30
1			Siemens AG	50
2				
3				
4				

NO	ST	EP	Rechnung	Blatt
				30
1			Siemens AG	50
2				
3				
4				

NO	ST	EP	Rechnung	Blatt
				30
1			Siemens AG	50
2				
3				
4				



RO	
ST	
EP	
Siemens AG	
S26361-0277-X*-11	
Blank	

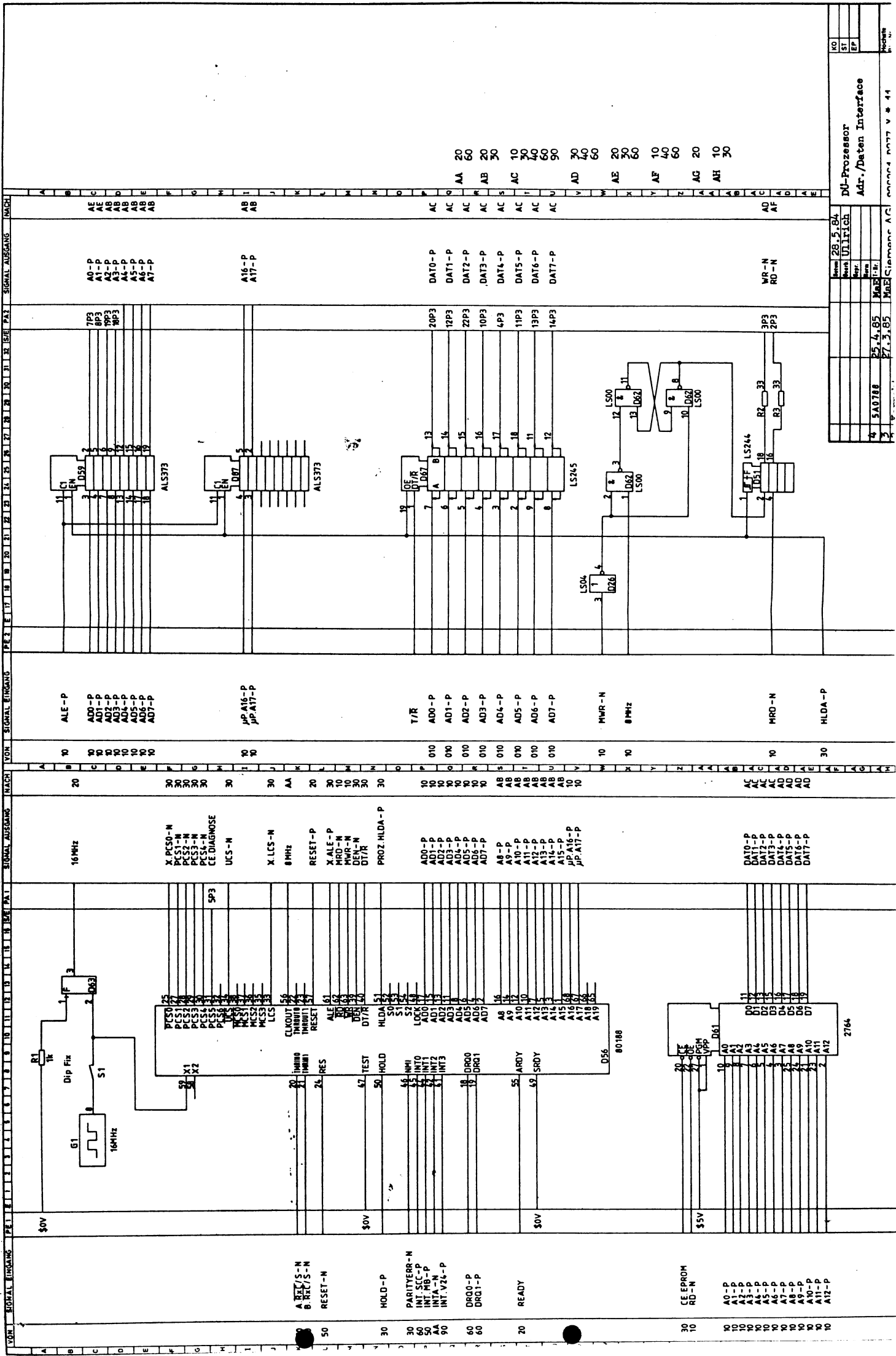
Internal	28.5.84
Serial	Utilitech
Rev.	
Drawn	
Checked	
Approved	
Siemens AG	
S26361-0277-X*-11	
Blank	

Internal	28.5.84
Serial	Utilitech
Rev.	
Drawn	
Checked	
Approved	
Siemens AG	
S26361-0277-X*-11	
Blank	

Internal	28.5.84
Serial	Utilitech
Rev.	
Drawn	
Checked	
Approved	
Siemens AG	
S26361-0277-X*-11	
Blank	

Internal	28.5.84
Serial	Utilitech
Rev.	
Drawn	
Checked	
Approved	
Siemens AG	
S26361-0277-X*-11	
Blank	

Internal	28.5.84
Serial	Utilitech
Rev.	
Drawn	
Checked	
Approved	
Siemens AG	
S26361-0277-X*-11	
Blank	



VON	SIGNAL	ERWARTUNG	PAZ	PAZ	SIGNAL	AUSGANG	MACH
50	ALE-P	20	10	10	ALE-P	20	10
30	AD0-P	10	10	10	AD0-P	10	10
30	AD1-P	10	10	10	AD1-P	10	10
30	AD2-P	10	10	10	AD2-P	10	10
30	AD3-P	10	10	10	AD3-P	10	10
30	AD4-P	10	10	10	AD4-P	10	10
30	AD5-P	10	10	10	AD5-P	10	10
30	AD6-P	10	10	10	AD6-P	10	10
30	AD7-P	10	10	10	AD7-P	10	10
30	AD8-P	10	10	10	AD8-P	10	10
30	AD9-P	10	10	10	AD9-P	10	10
30	AD10-P	10	10	10	AD10-P	10	10
30	AD11-P	10	10	10	AD11-P	10	10
30	AD12-P	10	10	10	AD12-P	10	10
30	AD13-P	10	10	10	AD13-P	10	10
30	AD14-P	10	10	10	AD14-P	10	10
30	AD15-P	10	10	10	AD15-P	10	10
30	AD16-P	10	10	10	AD16-P	10	10
30	AD17-P	10	10	10	AD17-P	10	10
30	DAT0-P	10	10	10	DAT0-P	10	10
30	DAT1-P	10	10	10	DAT1-P	10	10
30	DAT2-P	10	10	10	DAT2-P	10	10
30	DAT3-P	10	10	10	DAT3-P	10	10
30	DAT4-P	10	10	10	DAT4-P	10	10
30	DAT5-P	10	10	10	DAT5-P	10	10
30	DAT6-P	10	10	10	DAT6-P	10	10
30	DAT7-P	10	10	10	DAT7-P	10	10
30	RD-N	10	10	10	RD-N	10	10
30	WR-N	10	10	10	WR-N	10	10
30	HOLD-P	30	10	10	HOLD-P	30	10

NO	ST	EP	HOCHZ
1	28.5.84		
2	Ulrich		
3			
4	5.10.88	25.4.85	
5	27.3.85		

NO	ST	EP	HOCHZ
1	28.5.84		
2	Ulrich		
3			
4	5.10.88	25.4.85	
5	27.3.85		

alt

Produktspezifikation  
 der Flachbaugruppe DUEAI  
 (S 26361 - D277)

			11.10.83		
			Ullrich		
					Flachbaugruppe DUEAI
			Spezif.277		
			Siemens AG		
					S 26361 - D 277 - X - * - 59
09	6.03.85	U1			



---

**Inhalt:**  
-----

1	Uebersicht	3
2	Kennwerte	
2.1	Eigenschaften der seriellen Schnittstellen	4
2.2	Elektrische Eigenschaften	4
3	Funktionen	
3.1	Blockschaltbild	5
3.2	Schaltungsbeschreibung	5
3.3	Adress-Tabelle des DUE-Prozessors	10
3.3.1	Memory-Adressen	10
3.3.2	I/O-Adressen	11
3.4	Interne Register	11
4	Software-Interface	
4.1	Multibus I/O-Adressierung	
4.1.1	BADR-Register	16
4.1.2	Lower- / Upper-Reg	17
4.2	EPROM-Boot	17
4.3	Aufbau des Kommunikationsbereichs	20
4.3.1	Status-Register	21
4.3.2	Befehls-Register	22
4.3.3	Firmwareversion	22
4.3.4	Speicherausbau	23
4.3.5	A.out Format des Warmladers	23
5	Hardware-Interface	
5.1	System-Schnittstelle	24
5.2	DUE-Schnittstellen	24
<b>Anhang</b>		
1.	Bauderate-Einstellungen fuer den SCC (asynchron Mode)	27
2.	Schaltereinstellungen	28
3.	Diagnose-DUART	29
4.	Firmware	30

## 1. Uebersicht

Die Flbgr. DUEAI ist eine Standard Multibus-Flachbaugruppe. An der Systemschnittstelle gelten die funktionellen und elektrischen Eigenschaften, wie sie in der "MULTIBUS SPECIFICATION" beschrieben sind.

Die Flbgr. dient der Entlastung des Hauptprozessors der 9780 Systemeinheit durch eigenstaendiges Abarbeiten von ladbaren DUE-Prozeduren. Es besteht die Moeglichkeit, wahlweise zwei X21, zwei V24 oder eine V24 und eine X21 Schnittstelle zu steuern.

Denkbare Einsatzmoeglichkeiten sind TRANSDATA-Anschluss, SNA-Anbindung (SDLC) oder Datexanschluss (HDLC).

Es ist sowohl synchrone als auch asynchrone Betriebsart moeglich.

## Literaturhinweise

- [1] Normen fuer Informationsverarbeitung, Datenkommunikation
- [2] CCITT Empfehlungen der V- und der X-Serie (Decker Verlag)
- [3] Component Data Catalog Intel Jan. 82 (Nr. 210 298-001)
- [4] Rahmenspezifikation 9780 S26361 - L6 - X - \* - 59
- [5] Spezifikation des Zilog Bausteins 8530 SCC  
(Technical Manual, April 1982  
SCC Initialization - Application Note, Sept. 1982)
- [6] Multibus-Spezifikation der Fa. Intel (Rev. 4 6.82  
Order number 9800683-04)

## 2. Kennwerte

### 2.1 Eigenschaften der seriellen Schnittstellen

- Betriebsarten: synchron oder asynchron
- max. Bitrate 1 Mbit/sek (synchron od. asynchron)
- Zeichenrahmen 5-6-7-8 Bit/Zeichen
- Zeichensicherung: Paritaetsbit even od. odd moeglich.
- Blocksicherung per CRC (CCITT oder CRC-16) moeglich.
- Taktrueckgewinnung ueber Empfangsdaten moeglich
- NRZ-, NRZI-, Manchester-, FM-Decodierung moeglich
- local loop moeglich (Diagnose- bzw. Testmoeglichkeit)
- Sender oder Empfaenger sind per Befehl abschaltbar.
- Unterdrueckung von SYN im Empfangspuffer moeglich

### 2.2 Elektrische Kennwerte

Die Stromaufnahme betraegt:

+5V	ca. 2,1 A
+12V	ca. 0,1 A
-12V	ca. 0,1 A

Die Toleranzwerte dieser Spannungen sind festgelegt

- bei +5V durch Regeln fuer TTL Technik
- bei +/- 12V durch Regeln der Schnittstelle V28

Fuer die elektrischen Kennwerte der beiden Schnittstellentypen wird auf die Definition der beiden Schnittstellen hingewiesen.

[1] und [2]

### 3. Funktionen

---

#### 3.1 Blockschaltbild

---

s. naechste Seite

#### 3.2 Schaltungsbeschreibung

---

Die Flbgr. arbeitet als "Slave" am Multibus und besteht im wesentlichen aus folgenden Komponenten

##### - Taktversorgung

---

Von zwei Quarzoszillatoren (16 MHz, 7.3728 MHz) werden alle notwendigen Taktfrequenzen abgeleitet.

16 MHz: Prozessorclock, Speichertiming

3.68 MHz: SCC-Systemclock

Mittels einer monostabilen Kippstufe wird der Refreshclock von 62.5 KHz erzeugt.

##### - DUE-Prozessorteil

---

Mikroprozessor 80188, interne Taktfrequenz 8 MHz mit integriertem Interrupt-Controller, integrierter CS-Logik sowie 3 Timer und 2 DMA-Kanaele, alle notwendigen Bussignale werden vom Prozessor erzeugt

8 kB EPROM (Bootstrap-Lader)

##### - Dual Port Ram

---

Diskret aufgebauter Ram-Controller.

moegliche Speicherausbauten: 64 kB, 128 kB, 192 kB oder 256 kB, der Speicher ist parityueberwacht.

Zugriffe des DUE-Prozessors erfolgen ohne wait-states.

Das gesamte Ram oder Teilbereiche davon koennen ueber Multibus adressiert werden. Drei ueber Multibus ladbare Register legen dazu die Basisadresse des gesamten Rams, sowie die obere und untere Grenze des Dual port Fensters fest. Die Arbitration zwischen beiden Schnittstellen geschieht ueber die Prozessorsignale "HOLD" und "HOLD-ACKN." Bei beiden Schnittstellen kann durch spezielle Lock-Befehle eine Busuebergabe verhindert werden (beim Multibus durch das Signal "LOCK/", bzw. beim DUE-Prozessor durch das LOCK-Praefix im Programmcode).

Die automatische Generierung des Multibus-Signals INH1/ verhindert Kollisionen, falls das Dual port Ram im Adressbereich des Hauptspeichers liegt. Die Datenbreite zum DUE-Prozessor wie auch zum Multibus betraegt 8 Bit.

#### - Schnittstellenkanaele

---

##### V24:

Die Steuerleitung S1 (DEE betriebsbereit) kann fuer jeden Kanal ueber das jeweilige I/O-Reg. gesteuert werden.

Die Leitung M3 (ankommender Ruf) beider Kanaele kann einen Sammelinterrupt (INT 3) beim DUE-Prozessor ausloesen und ueber den Befehl "Read I/O-Latch" gelesen werden (s. 3.4).

Der SCC kann neben dem synchron Mode auch im asynchron Mode arbeiten. (Einstellungen im Anhang).

##### Ebene 1:

Den Bedingungen der Norm DIN 66259 wird entsprochen.

Als Treiberbausteine finden 75150, als Empfaenger 75154 Verwendung. Als Versorgungsspannung dient diesen Bausteinen +/- 12V.

Logische Verknuepfungen zwischen den einzelnen Signalen werden nicht vorgenommen.

Im einzelnen koennen folgende Schnittstellenleitungen kontrolliert bzw. gelesen werden:

Leitung	Steckerpin Trapez- buchse	geht nach / kommt von
D1 (Sendedaten)	2	SCC: TxD
D2 Empfangsdaten	3	SCC: RxD
T1 Sendetakt	24	SCC: TRxC
T2 "	15	"
T4 Empfangstakt	17	SCC: RTxC
S1 Betriebsbereit	20	I/O Mode-latch
S2 Sendeteil ein	4	SCC: RTS
S4 hohe Uebertr.- geschw. ein	23	liegt fest auf "ein"
M1 Betriebsbereit	6	SCC: DCD
M2 Sendebereit	5	SCC: CTS
M3 ankomm. Ruf	22	I/O Buffer

X21:

Ueber die Modemleitungen des SCCs koennen die einzelnen Leitungen gesetzt und ueberwacht werden.

Die Zuordnung ist dabei: (bei normaler Betriebsart (s. 3.4))

- c steuern ----> SCC: RTS
- t senden ----> SCC: TxD
- i melden ----> SCC: DSR
- r empfangen ----> SCC: RxD
- s Schritttakt ----> SCC: TRxC und RTxC

Die Baugruppe bietet folgende Erkennungs- und Hilfsschaltungen:

- a)  $r=0$  und  $i=aus$  ueber 16 Schritttakte: Zustand 19 (Ausloesung) geht an den DCD-Eingang des SCC und kann einen Interrupt ausloesen.
- b)  $i=aus$  ueber 16 Schritttakte: geht an SCC: CTS und kann einen Interrupt ausloesen. (jeder Zustand von  $r$  und  $i$  muss mind. 16 Takte anliegen, damit er als Dauerzustand bewertet und entsprechend reagiert werden kann).
- c) ueber einen Multiplexer kann eingestellt werden, welcher Sender die Leitungen  $t$  und  $c$  steuert.  
Folgende 4 Moeglichkeiten sind einstellbar:
  1. Zustand 24 ("DEE gestoert",  $t=0$  und  $c=aus$ ) damit werden nach einem Reset der Baugruppe die Schnittstellenleitungen definiert gesetzt.
  2. normale Betriebsart: die Leitungen  $t$  und  $c$  koennen ueber den SCC (TxD, RTS) gesteuert werden.
  3. Zustand 14 ("DEE kontrolliert nicht bereit",  $t=01010...$  und  $c=aus$ ) die Signalfolge auf der Ltg.  $t$  wird dabei synchron zu  $s$  (Schritttakt) ausgegeben.
  4. Es kann auf vorher im I/O-Latch programmierte Zustaende von  $t$  und  $c$  geschaltet werden. (Dies erlaubt eine Neuinitialisierung des SCCs nach einem Verbindungsaufbau).
- d) die Zaehleingaenge der Timer 0 und 1 sind mit den Schritttakt  $s$  der Kanale A und B verbunden und koennen damit als Zeitueberwachung fuer die Leitungen  $c$  und  $t$  eingesetzt werden. (Min. 24 Takte bei Signalaenderungen auf den Leitungen)

### Diagnosestecker:

Als Testhilfsmittel waehrend der Phase der Softwareintegration kann ueber einen Diagnosestecker ein Babyboard mit einem DUART angeschlossen werden.

(Damit Anschlussmoeglichkeit fuer ein Terminal oder einen Drucker, s. Anhang).

### - Systemschnittstelle

Die Flbgr. arbeitet als "Slave" am Multibus. Alle benoetigte Bussignale sind gepuffert. Der Multibus selbst wird mit max. 4 LS Eingangslasten belastet.

### - Interruptlogik

#### a. DUE-Proz. --> Hauptprozessor

Der DUE-Prozessor kann ueber eine entsprechende I/O-Adresse einen Interrupt beim Hauptprozessor ausloesen. Ueber einen Dip Fix Schalter kann eine der 8 Multibus-Intltg. sel. werden.

#### b. Flbgr. interne Interruptmoeglichkeiten

INT 0: SCC (wird der SCC im "vector mode" betrieben, muss INT 0 des 80188 auf "cascade mode" programmiert werden! INT 2 dient dann als INTA fuer den SCC) s.[3]

INT 1: Interrupt vom Hauptprozessor

INT 3: Sammelinterrupt fuer V24-Schnittstellensignale  
(M3, s. 3.4)

NMI: Parity-Fehler

Timer 0: Zeitueberwachung fuer Kanal A

Timer 1: " " B

Timer 2: universell einsetzbar



DMA 0: Interrupt vom DMA Controller 0

DMA 1: " 1

Der Hauptprozessor kann mittels Multibus I/O Adressen beim DUE-Prozessor einen INT 1 oder einen Reset auslösen. Reset bewirkt einen Rucksprung in den Boot-EPROM. Alle uebrigen Interrupt-Routinen muessen von der DUE-Prozedur festgelegt werden.

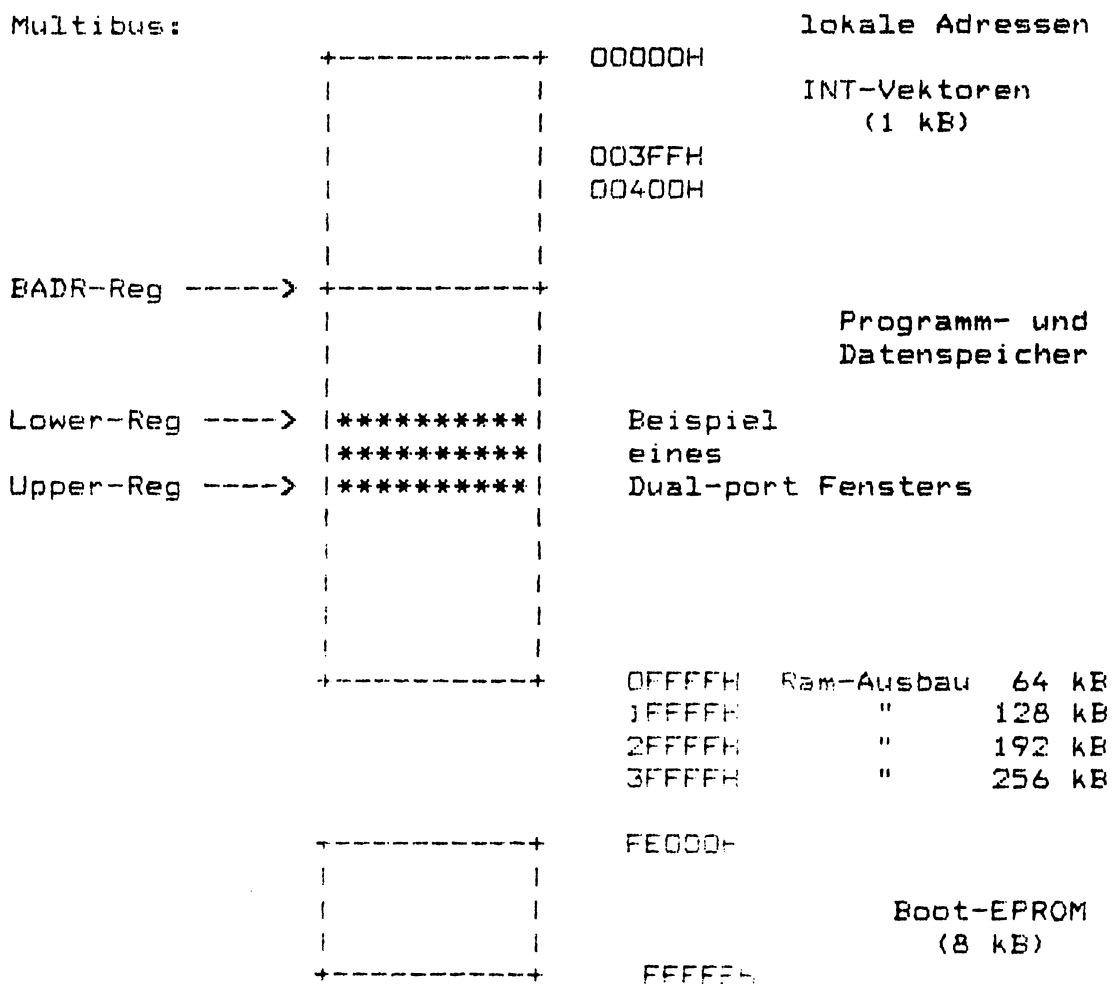
### 3.3 Adress-Tabelle des DUE-Prozessors

#### 3.3.1 Memory-Adressen

Die Rams werden selektiert mit LMCS (lower memory chip select)

Das EPROM mit UMCS (upper memory chip select):

Datenblatt 80188 [3])



### 3.3.2 I/O - Adressen

Alle I/O Peripherals werden mit den PCS-Leitungen (peripheral chip select) angesprochen. Die Basisadresse wird innerhalb der Initialisierung auf 0 festgelegt (s. [3]).

PCS 0:	000H	Status/CMD Kanal B	\	
	001H	Daten-Reg "		SCC
	002H	Status/CMD Kanal A	>	8530
	003H	Daten-Reg "		
			/	
PCS 1:	080H	Reset INT 1 (Interrupt vom Multibus)		
PCS 2:	100H	Reset INT 3 (V24-Meldeltg. M3)		
PCS 3:	180H	Set Multibus-Interrupt		
PCS 4:	200H	Set Prozessor Mode-Reg.		
	210H	Set I/O Reg. Kanal A		
	220H	" " B		
	230H	Read I/O Latch		

ueber die I/O-Adressen 280H ... 28FH kann ein aufsteckbares Babyboard mit einem Diagnose-DUART angesprochen werden.

(s. Anhang)

### 3.4 Interne Register

Eine detaillierte Beschreibung der einzelnen Register des SCC (Zilog 8530) ist dem Datenbuch [5] zu entnehmen.

Nachfolgend sind deshalb nur die internen Register beschrieben.

Set Prozessor Mode-Register (200H)

```

D7 -----+----- D0
+-----+-----+-----+-----+-----+-----+-----+
| LED | x | Takt B | Takt A | Res-SCC | DisNMI | DMA1 | DMA0 |
+-----+-----+-----+-----+-----+-----+-----+

```

Ueber die beiden Bits DMA 0 und DMA 1 kann selektiert werden, welcher der beiden Kanäle A und B im DMA-Mode betrieben werden soll.

DMA 1	DMA 0	Betriebsart
0	0	Die SCC-Kanäle A und B werden halbduplex betrieben (W/REQA ----> DREQ0 W/REQB ----> DREQ1)
0	1	SCC-Kanal A wird vollduplex betrieben (W/REQA ----> DREQ0 DTR/REQA ----> DREQ1)
1	0	SCC-Kanal B wird vollduplex betrieben (W/REQB ----> DREQ0 DTR/REQB ----> DREQ1)
1	1	DMA disabled

DisNMI: 0 ----> NMI enable  
1 ----> NMI disabled

Res-SCC: 0 ----> Reset fuer SCC  
1 ----> normaler Betrieb

Takt A/B: ermöglicht bei den V24-Schnittstellen neben der ext. Taktzufuehrung (T2 und T4) auch eine Eigentaktung (T1).

0 ----> T2 und T4 werden dem SCC zugefuehrt

T2 ----> Pin 14 (TRxCA) fuer Kanal A

Pin 26 (TRxCB) " B

T4 ----> Pin 12 (RTxCA) " A  
Pin 28 (RTxCB) " B

1 ----> die externe Taktzufuehrung fuer T2 wird disabled. Der Sendetakt kann (bei entsprechender Programmierung des SCCs) vom SCC selbst erzeugt und als T1 an die jeweilige V24 Schnittstelle geschaltet werden.

Pin 14 (TRxCA) ----> T1 fuer Kanal A

Pin 26 (TRxCB) ----> " B

(T4 wird davon nicht beruehrt)

! Achtung: Bevor die SCC-Pins 14 od. 26 als Taktausgang  
! programmiert werden, ist es zwingend erforderlich,  
! das zuerst die Bits "Takt A/B" im Prozessor Mode-Register gesetzt werden.

Die einzelnen Bits koennen ueber den Befehl "Read I/O Latch" rueckgelesen werden.

(Diese Betriebsart ist bei den Prototyp-F1bgr. (ersichtlich durch den Aufdruck W26361-D277-Z4-01-5 auf der Bauteileseite nicht moeglich!)

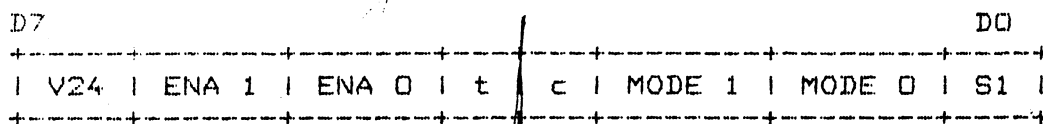
LED: Steuert die rote Diagnose-LED

0 ----> LED an

1 ----> LED aus

Das Prozessor Mode-Register kann nur beschrieben werden!

Set I/O Mode-Register Kanal A/B (210H / 220H)



Beide Register besitzen gleiches Datenformat

S1: steuert das Schnittstellensignal S1 der V24-Schnittstelle

0 ---> S1 aus

1 ---> S1 ein

MODE 0 MODE 1 steuert die X21-Schnittstelle

0 0 Zustand 24 <sup>bei X21</sup> (DEE gestoert, wird nach Reset ausgegeben)

0 1 normaler Betriebsmode (SCC steuert t und c)

1 0 latched mode: die im I/O Mode-Reg. fuer t und c angegebenen Werte werden ausgegeben.

t = 0/1 ---> X21: t = 0/1

c = 0/1 ---> X21: c = aus/ein

1 1 Zustand 14 (DEE kontrolliert nicht bereit, t = 01010..., c = aus) wird ausgegeben.

ENA 0: steuert bei V24 die Interruptmoeglichkeit von M3 (ankommender Ruf)

steuert bei X21 die Erkennungsschaltung fuer Zustand 19 (Ausloesung)

0 ---> disable und clear

1 ---> enable

ENA 1: steuert bei X21 die Erkennungsschaltung "i aus" ueber  
16 Schritttakte

0 ---> disable und clear

1 ---> enable

V24: steuert den Schnittstellen-Multiplexer V24 - X21

0 ---> X21 selektiert

1 ---> V24 "

Die I/O Mode-Register koennen nur beschrieben werden!

c. Read I/O Latch (230H)

D7							D0
TAKT A	TAKT B	B.I	A.I	M3-B	M3-B	M3-A	M3-A
				INT-FF		INT-FF	

Die einzelnen Bits geben die Zustaende der Meldeleitungen M3  
der beiden V24-Schnittstellen wieder

M3-A: Leitung M3 fuer Kanal A

M3-B: " " B

0 ---> aus

1 ---> ein



M3-A INT-FF, M3-B INT-FF: gibt den Zustand des jeweiligen  
Interrupt Flip-flops an.

(Eine Zustandsaenderung von "aus" nach "ein" setzt das ent-  
sprechende Flip-flop)

0 ---> Grundzustand

1 ---> Leitungsaenderung erkannt

Ueber die I/O-Adressierung 100H kann dieser INT wieder rueck-  
gesetzt werden.

A.I / B.I: Geben den Zustand auf den X21-Schnittstellenleitungen

1 (melden) fuer den jeweiligen Kanal A bzw. B an

0 ---> aus

1 ---> ein

TAKT A/B: Die ueber das Prozessor Mode-Reg. programmierten

Taktmodi koennen rueckgelesen werden

0 ---> extern Takt

1 ---> disable extern Takt

## Software-Interface

### 4.1 Multibus I/O-Adressierung

Die Basisadresse kann mit einem 8 pol. Dip Fix in Schritten von 4 kByte eingestellt werden. (Einstellung: s. Anhang)

Nachfolgend sind die einzelnen Register mit ihren Offsetwerten zur eingestellten Basisadresse angegeben.

*1200 H oder 2900*

+ 001H: Enable Dual-port Fenster

002H: Res. Multibus Interrupt Flipflop

003H: Set INT 1 (Interrupt an DUE-Prozessor)

004H: Set Reset (bewirkt Ruecksprung in EPROM-Boot)

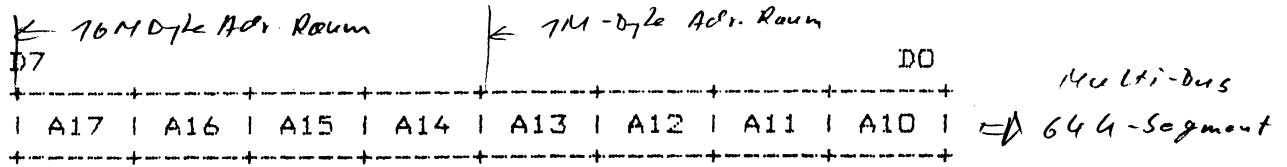
005H: Set Basisadresse (BADR-Reg)

006H: Set Upper-Register

007H: Set Lower-Register

008H ..... 0FFH sind reserviert und duerfen nicht angesprochen werden.

4.1.1 BADR-Register

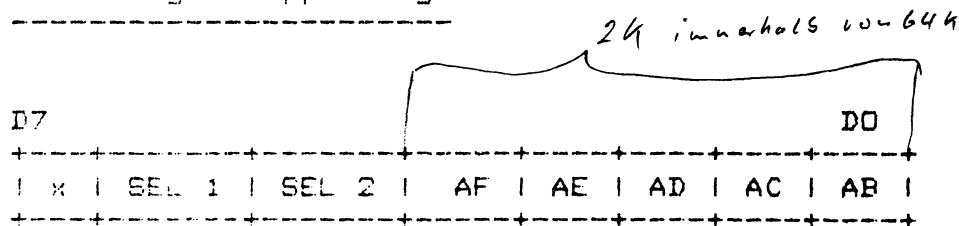


Im BADR-Reg wird die Multibus-Basisadresse des Dual port Segments angegeben. Mit A17 ... A10 koennen die Multibusadressen ADR 17/ ... ADR 10/ festgelegt werden.

Das Dual port Segment kann damit innerhalb eines Adressbereiches von 16 MB bei einer Schrittweite von 64 kB beliebig verschoben werden. Sollte der DUE-Prozessor in einer 9780-Systemeinheit ohne P2-Platter arbeiten, muessen die Bits A14 ... A17 auf 0 gesetzt sein. Beispiele zur Programmierung des Dual port Segments:

Multibus-Adresse	Registerwert
000000H	00H
010000H	01H
020000H	02H
040000H	04H
080000H	08H
100000H	10H etc.

4.1.2 Lower-Reg. / Upper-Reg.



Mit den Registern koennen die Grenzen des Dual port Fensters innerhalb des Dual port Segments festgelegt werden.

AB ... AF entsprechen dabei den Multibus-Adressbits ADR B/ bis ADR F/.



Mit SEL 0 und SEL 1 kann festgelegt werden, auf welche Rambank des DUE-Prozessors zugegriffen werden soll.

SEL 1	SEL 0	Rambank	lokaler Adressbereich
0	0	0	00000H ... 0FFFFH
0	1	1	10000H ... 10000H
1	0	2	20000H ... 2FFFFH
1	1	3	30000H ... 3FFFFH

Ein Zugriff in das Ram wird gestattet bei Erfuellen der folgenden beiden Bedingungen:

1. [Lower-Reg] <= Multibus-Adr. <= [Upper-Reg]  
[AB ... AF] AB ... AF [AB ... AF]
2. [BADR-Reg] <= Multibus-Adr. < [BADR-Reg] + 64 kB

Die minimale Fensterbreite bzw. Schrittweite ist 2kB:

$$[\text{Lower-Reg}] = [\text{Upper-Reg}]$$

Die maximale Fensterbreite ist 64 kB:

[Lower-Reg] = 00H,	[Upper-Reg] = 1FH	(Rambank 0)
[ " ] = 20H,	[ " ] = 3FH	" 1
[ " ] = 40H,	[ " ] = 5FH	" 2
[ " ] = 60H,	[ " ] = 7FH	" 3

Das Fenster kann geschlossen werden, wenn:

$$[\text{Lower-Reg}] > [\text{Upper-Reg}]$$

$$[\text{AB ... AF}] \quad [\text{AB ... AF}]$$

Beispiele zur Programmierung der Fenstergrenzen

Offset zur Segment-Adr	Reg. Wert			
	Rambank 0	Rambank 1	Rambank 2	Rambank 3
00000H	00H	20H	40H	60H
00800H	01H	21H	41H	61H
01000H	02H	22H	42H	62H
01800H	03H	23H	43H	63H
04000H	08H	28H	48H	68H
06000H	0CH	2CH	4CH	6CH
0A000H	14H	34H	54H	74H
0F800H	1FH	3FH	5FH	7F

Beispiel: Das Dual port Fenster soll unter den Multibusadressen  
0FB000H bis 0FBFFFH initialisiert werden.

Selektiert werden soll die lokale Rambank 0 des DUE-  
Prozessors:

[BADR-Reg] : 0FH

[Lower-Reg]: 16H

[Upper-Reg]: 17H

Nachdem die einzelnen Register geladen worden sind, muss mit  
einem Schreibzyklus auf die I/O-Adresse 01H das Dual port  
Fenster enabled werden (s. 4.1)

4.2 EPROM-Boot

Nach einem Hardware- bzw. Software-Reset laeuft automatisch der  
EPROM-Boot an. Er initialisiert den Kontroll-Block des uP 80188  
und fuehrt einen ROM-Test, einen Ram-Test und einen SCC-Test

durch. Das Testergebnis, der Ramausbau und die Firmwareversion werden in den jeweiligen Registern angezeigt.

Die Dauer der Initialisierung ist abhaengig vom Speicherausbau und dauert bei 256 kB Ram ca. 6s.

Danach wird auf einen Interrupt vom System gewartet, der dann durch einen Quittungsinterrupt seitens des DUE-Prozessors beantwortet wird. (Der genaue Ablauf der Selbsttests ist im Anhang angegeben)  
Der Hauptprozessor oeffnet ab der lokalen Adresse 8000H ein Fenster im Ram, laedt den Warmlader, setzt im Befehls-Register das Bit "AOUTSTART" und erzeugt einen Interrupt.

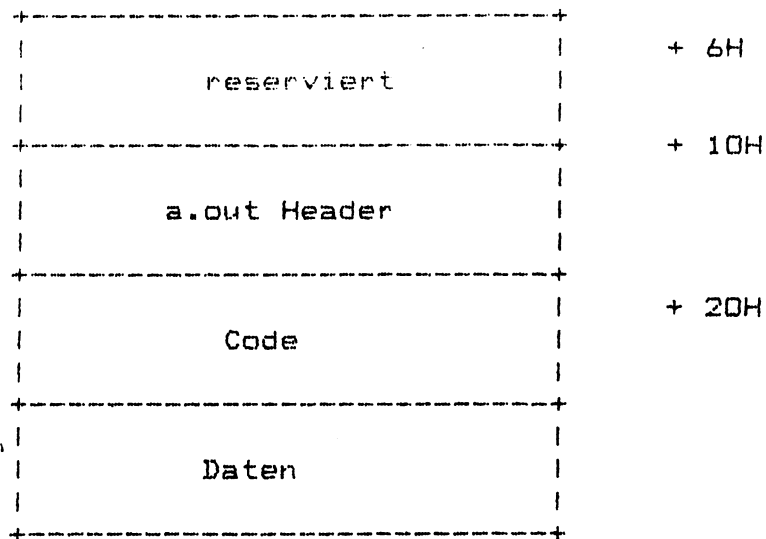
Anhand des a.out Headers ermittelt sich der DUE-Prozessor die Startadresse, setzt die Segmentregister DS, SS und CS auf den Codeanfang (Fensteradresse + 20H) bzw. ES auf das Fenster selbst und startet den Warmlader.

#### 4.3 Aufbau des Kommunikationsbereich

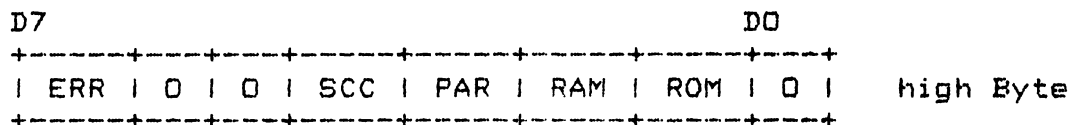
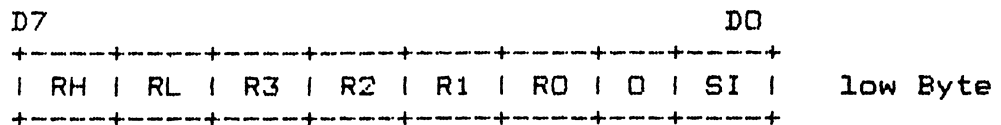
---

Der Kommunikationsbereich liegt innerhalb des Dual port Fensters und befindet sich vor Prozeduruebernahme ab der Adresse 8000H (lokale Adresse; entspricht den Multibusadresse FB000H. Die obere Grenze des Fensters kann beliebig gewaehlt werden.

+-----+		
Status-Reg. (low Byte)		Fensteradr.
+-----+		
Status-Reg. (high Byte)		+ 1H
+-----+		
Befehls-Register		+ 2H
+-----+		
reserviert		+ 3H
+-----+		
Firmware-Version		+ 4H
+-----+		
Speicherausbau		+ 5H
+-----+		



4.3.1 Statusregister



ROM: ROM-Fehler

kennzeichnet einen waehrend des Romtest erkannten Fehler

RD ... R3, RAM, PAR: RAM-Fehler, Parity-Fehler

kenzeichnet einen waehrend des Ramtest erkannten Fehler

Der Index gibt dabei die fehlerhafte Ram-Bank

RL: Der Ram-Fehler liegt im Low-Ram (Bit 0 ... 3)

RH: " High-Ram (Bit 4 ... 7)

SCC: kennzeichnet einen waehrend des SCC-Test erkannten Fehler

ERR: Error, wird gesetzt, wenn einer der vorher aufgefuehrten Fehler erkannt wurde

SI: System initialisiert  
kennzeichnet den Abschluss der Initialisierungs- und Test-routinen.  
Sollte waehrend der Testprogramme einer der aufgefuehrten Fehler erkannt worden sein, wird der Warmlader nicht gestartet.

#### 4.3.2 Befehlsregister

-----

(Fensteradr. + 2H)

D7	D0
+-----+-----+-----+-----+	
0   0   0   0   0   0   0   0   AOUTSTART	
+-----+-----+-----+-----+	

AOUTSTART: signalisiert dem DUE-Prozessor, dass der Warmlader vom Hauptprozessor ins Ram geladen wurde und gestartet werden kann.

Dem Befehl muss ein INT 1 folgen.

#### 4.3.3 Firmwareversion

-----

(Fensteradr. + 4H)

D7	D0
+-----+-----+-----+-----+	
S7   S6   S5   S4   S3   S2   S1   S0	
+-----+-----+-----+-----+	

Die Bits S0 bis S7 geben den auf der Baugruppe befindliche Firmwareversion an

#### 4.3.4 Speicherausbau

---

(Fensteradr. + 5H)

```

D7                                     DO
+-----+-----+-----+-----+-----+
| x | x | x | x | x | x | R2 | R1 | R0 |
+-----+-----+-----+-----+-----+

```

Die Bits R0 bis R3 geben den auf der Baugruppe befindlichen Speicherausbau an

R2	R1	R0	
0	0	0	64 kB
0	0	1	128 kB
0	1	0	192 kB
1	0	0	256 kB

#### 4.3.5 A.out Format des Warmladers

---

Der Header des Warmladers beginnt bei Fensteradresse + 10H und hat folgende Form:

```

struct header {
    int      a_magic      /* magic number          */
    unsigned a_text      /* Laenge des Code-Segments
    unsigned a_data      /* Laenge der init. Daten
    unsigned a_bss       /* Laenge der nicht init. Daten
    unsigned a_syms      /* hier 0                */
    unsigned a_entry     /* Startadresse rel. zu a_text
    unsigned a_unused    /* hier 0                */
    unsigned a_flag      /* hier immer 1          */
};

```

## 5 Hardware-Interface

### 5.1 Systemschnittstelle

Die Systemschnittstelle ist multibuskompatibel.

Die Flbgr. selbst arbeitet als "Slave" am Bus.

Alle Bussignale, sowie das Bus-Timing sind in der Multibus-Spezifikation (Intel) beschrieben.

### 5.2 DUE-Schnittstellen

Die Verbindung zwischen der Flbgr. und dem Anschlussfeld der 9780 Systemeinheit geschieht ueber spezielle Flachbandleitungen.

Die Zuordnung der verschiedenen Stecker zu den einzelnen Kanaelen ist dabei:

V24 Kanal A	---	P4 Stecker
V24 Kanal B	---	P5 Stecker
X21 Kanal A	---	P6 Stecker
X21 Kanal B	---	P7 Stecker

Nachfolgend ist die Schnittstellenbelegung der Flachbaugruppensteckverbinder und der zugehoerigen Trapez-Buchsen im Anschlussfeld aufgefuehrt.

Schnittstelle V24

Steck-Mechanik: 22 pol. cod. Modulstecker

Signaldefinition: gem. DIN 66020 Teil 1

Modul	AMP	Signal-Name	Bemerkungen
01			codiert
02	02	D1 TxD 103	Sendedaten
03	03	D2 RxD 104	Empfangsdaten
04	04	S2 RTS 105	Sendeteil einschalten
05	05	M2 CTS 106	Sendebereitschaft DUEE
06	06	M1 DSR 107	Betriebsbereitschaft DUEE
07	07	E2 GND 102	Signalerde
08	-	-	-
09	23	S4 111	hohe Ueb.-geschw. ein
10	24	T1 113	
11	-	-	
12	-	-	
13	-	-	
14	15	T2 114	Sendeschriftakt
15	-	-	
16	17	T4 115	Empfangsschriftakt
17	-	-	
18	-	-	
19	20	S1.2 DTR 108/2	DEE betriebsbereit
20	22	M3 125	ankommender Ruf
21	-	-	codiert
22	-	Schutzerde	Schutzerde ?

Beide V24-Schnittstellen haben die gleiche Steckerbelegung

Alle nicht aufgefuehrten V24-Schnittstellensignale werden nicht verwendet bzw. nicht gesteuert.



Schnittstelle X21

Steck-Mechanik: 14 pol. cod. Modulstecker

Signaldefinition: gem. DIN 66020 Teil 2

Beide X21 - Schnittstellen haben identische Steckerbelegungen

Modul	AMP	Signal-Name	Bemerkungen
01	-		codiert
08	08	G	Signalerde, verb. mit 0 V
06	02	T(A)	Senden
14	09	T(B)	"
12	03	C(A)	Steuern
05	10	C(B)	"
11	04	R(A)	Empfangen
04	11	R(B)	"
10	05	I(A)	melden
03	12	I(B)	"
09	06	S(A)	Schrittakt
02	13	S(B)	"

Anhang

Bauderate-Einstellungen fuer den SCC

(fuer asynchrone Betriebsart)

Voreinstellung SCC:

WR-Reg 4: x16 clock

WR-Reg 11: Transmit-clock = BR-Generator output

Receive-clock = "

WR-Reg 14: BR-Generator enable

BR-Generator source = SCC-PCLK (3.6864 MHz)

WR-Reg 12:

Bauderate-Einstellung

WR-Reg 13:

Unten aufgefuehrte Tabelle gibt die Werte fuer die gaengigsten Uebertragungsgeschwindigkeiten an.

Daneben laesst sich der Wert fuer jede beliebige Bauderate nach folgender Formel berechnen (x16 clock!)

$$x = \frac{115313}{\text{Bauderate}} - 2$$

Dabei ist x der Wert fuer WR-Reg 12/13; die Bauderate wird in Bit/s angegeben.

Count value	Bauderate [Bit/s]
1	38400
4	19200
10	9600
22	4800
46	2400
94	1200
190	600
382	300

2. Schaltereinstellungen

*Gilt erst ab Version 1,  
bis dahin siehe Aufgabe 08  
bezüglich Schalterstellungen.*

a. Multibus I/O-Adresse

Einstellung mit den Schaltern S2 ... S9

<i>A15</i>								<i>A12   A11 - A0</i>	
S 9	S 8	S 7	S 6	S 5	S 4	S 3	S 2		
A-13/	A-12/	A-11/	A-10/	A-F/	A-E/	A-D/	A-C/		Multibusadressen
op	op	clo	op	op	op	op	op		= 2000H ... 2FFFH
									<i>4K</i>

Bei Auslieferung der Baugruppe sind S12 und S13 geschlossen. ?  
Die Baugruppe ist damit auf die I/O-Adresse 1800H eingestellt.

b. Multibusinterrupt

Einstellung mit den Schaltern S1 ... S8

S17	S16	S15	S14	S13	S12	S11	S10		
7	6	5	4	3	2	1	0		Multibus-INT-Ltgen.

Bei Auslieferung der Baugruppe ist S16 geschlossen und erzeugt damit einen INT 6/.

Die Schalter S1 und S8 dienen ausschliesslich Prueftechn. Zwecken und muessen im Normalbetrieb immer geschlossen sein.

3.

Diagnose-DUART

Zum Testen von geladenen Programmen kann ueber den Diagnosestecker P3 ein Babyboard mit einem DUART 2681 (Signetics) angeschlossen werden.

Am seriellen Ausgang des Babyboards koennen Geraete mit SS 97 angeschlossen werden.

Unter den folgenden I/O-Adressen koennen die einzelnen Register des DUARTs angesprochen werden.

I/O-Adr.	Lesen	Schreiben
280H	Mode-Reg 1/2 Kanal A	Mode-Reg 1/2 Kanal A
281H	Status-Reg. A	Clock-sel. A
282H	----	Command-Reg. A
283H	Receive-Reg. A	Transmit-Reg A
284H	----	Aux. control Reg

Die nicht aufgefuehrten Register werden nicht benoetigt bzw. von der Hardware nicht unterstuetzt.

Der DUART wird mit einem externen Clock von 3.6864 MHz betrieben.

Die Spannungsversorgung des Boards geschieht ueber den P3 Stecker vom DUE-Prozessor aus.

---

## Firmware

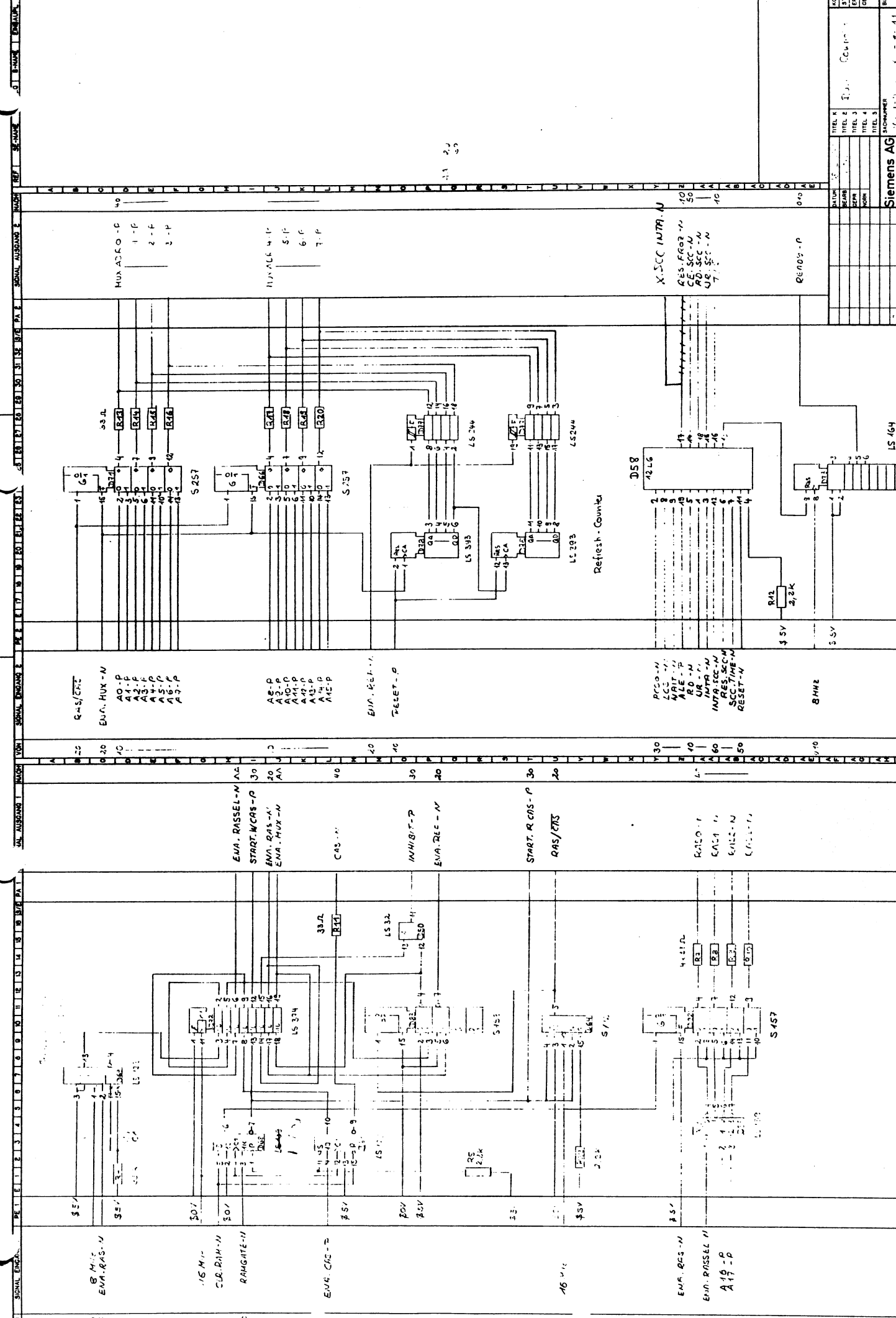
Nach einem Hard- bzw. Softwarereset wird in den Eprom-Boot gesprungen. Gleichzeitig wird die Diagnose-Led aktiviert.

1. Initialisierung des internen Kontrollblocks:  
Saemtliche CS-Signale definieren, Int-Contr. initialisieren, alle Interrupts sperren, alle INT-Flipflops ruecksetzen.
2. Romtest:  
Addieren des Rominhalts und Vergleich mit dem in der letzten Zelle stehenden Checkword.  
Falls Uebereinstimmung: Sprung zum Ramtest  
Falls Fehler: Led wird kurzzeitig ausgeschaltet (ca.300 ms)  
Im Statusregister wird eine Fehlermeldung hinterlegt, danach wird die Initialisierung abgebrochen.
3. Ramtest:  
Test der untersten 32 Bytes (dienen spaeter als Stack)  
Test der Rambank 0.  
Danach wird der Speicherausbau ermittelt und die vorhandenen Speicherbaenke getestet.  
Falls Fehler: Led wird 2x kurzzeitig ausgeschaltet und im Status-Register eine Fehlermeldung hinterlegt. Die Initialisierung wird danach abgebrochen.
4. SCC-Test:  
Der SCC wird im local loop modus betrieben.  
Falls Fehler: Led wird 3x kurzzeitig ausgeschaltet und eine Fehlermeldung im Status-Register hinterlegt. Die Initialisierung wird abgebrochen.

5. Warten auf Interrupt vom Hauptprozessor.  
Danach Generieren eines Quitungsinterrupts.  
Warten auf 2. Interrupt vom Hauptprozessor.  
Falls danach im Befehlsregister "AOUTSTART" gesetzt ist,  
wird Warmlader gestartet.

In der Zeit zwischen Beendigung der Tests und Start Warmlader blinkt die Led im 1 Hz Rhythmus und zeigt damit optisch den fehlerfreien Abschluss der Tests an.





NO	DATE	BY	CHK	REVISION
1	01.10.81	J. Müller		1
2	15.11.81	J. Müller		2
3	05.12.81	J. Müller		3
4	10.01.82	J. Müller		4
5	20.02.82	J. Müller		5
6	01.03.82	J. Müller		6
7	15.03.82	J. Müller		7
8	30.03.82	J. Müller		8
9	15.04.82	J. Müller		9
10	01.05.82	J. Müller		10
11	15.05.82	J. Müller		11
12	30.05.82	J. Müller		12
13	15.06.82	J. Müller		13
14	01.07.82	J. Müller		14
15	15.07.82	J. Müller		15
16	30.07.82	J. Müller		16
17	15.08.82	J. Müller		17
18	30.08.82	J. Müller		18
19	15.09.82	J. Müller		19
20	30.09.82	J. Müller		20
21	15.10.82	J. Müller		21
22	30.10.82	J. Müller		22
23	15.11.82	J. Müller		23
24	30.11.82	J. Müller		24
25	15.12.82	J. Müller		25
26	30.12.82	J. Müller		26
27	15.01.83	J. Müller		27
28	30.01.83	J. Müller		28
29	15.02.83	J. Müller		29
30	30.02.83	J. Müller		30
31	15.03.83	J. Müller		31
32	30.03.83	J. Müller		32
33	15.04.83	J. Müller		33
34	30.04.83	J. Müller		34
35	15.05.83	J. Müller		35
36	30.05.83	J. Müller		36
37	15.06.83	J. Müller		37
38	30.06.83	J. Müller		38
39	15.07.83	J. Müller		39
40	30.07.83	J. Müller		40
41	15.08.83	J. Müller		41
42	30.08.83	J. Müller		42
43	15.09.83	J. Müller		43
44	30.09.83	J. Müller		44
45	15.10.83	J. Müller		45
46	30.10.83	J. Müller		46
47	15.11.83	J. Müller		47
48	30.11.83	J. Müller		48
49	15.12.83	J. Müller		49
50	30.12.83	J. Müller		50

















9780 SOFTWARE-ENTWICKLUNGS-PROJEKT-BAND

Version 1

BAND: 9780 SINIX

KAPITEL 70: Produktspezifikation

Verteiler:

ABSCHNITT 80: DUE- System, DUEAI-Treiber f. MSV1

siehe Deckblatt

D ST DF 122  
46/623

Hr. Novotny

Mch P

Raum

Bitte  
lesen

Hei-

Bearbeiter: Heidbreder, D ST SP412 089/636-

Mch-P/DG, R431  
Tel. 96-47987

SIEMENS - DATENVERARBEITUNG  
MUENCHEN

25. 4.1985  
Seite 1

44232 Novotni



## INHALT

1.	Allgemeine Beschreibung des DUEAI- Anschlusses .....	2
2.	Leistungsbeschreibung des SINIX-Treibers .....	3
2.1.	Uebersicht .....	3
2.2.	SINIX- Schnittstellen zum DUEAI - Treiber .....	5
3.	Allgemeine Schnittstelle zu Kontrollerbaugruppen .....	9
3.1.	Dual- Port Ram Aufteilung .....	9
3.2.	Einteilung der Kontrollblöcke .....	12
3.3.	Ablaufdiagramme .....	14
4.	Kommunikation zw. SINIX-Rechner und DUEAI-Karte .....	16
4.1.	Aufteilung des gem. Speichers auf der DUEAI- Baugruppe ...	16
4.2.	Kommandos / Meldungen zwischen DUEAI- Treiber und DUEAI-Softw	
5.	Laden der DUEAI Baugruppe .....	23
5.1.	Kommandos zum Laden der DUEAI- Software .....	23
5.2.	Synchronisation zwischen SINIX und DUEAI Baugruppe. ..	24
5.3.	Ablaufdiagramme .....	26

9780 SOFTWARE-ENTWICKLUNGS-PROJEKT-BAND	Version 1
BAND: 9780 SINIX	
KAPITEL 70: Produktspezifikation	Verteiler:
ABSCHNITT 80: DUE- System, DUEAI-Treiber f. MSV1	siehe Deckblatt

Verantwortlich fuer diese Spezifikation  
Hr. Heidbreder  
D ST SP 412, Tel 47987

Stand: 25. 04. 1985

#### 1. Allgemeine Beschreibung des DUEAI- Anschlusses

Fuer die Integration des DUEAI-Kontrollers S26361-D277 wird in SINIX fuer den BS2000-Anschluss mit einer MSV1-Prozedur ein DUEAI- Treiber realisiert.

In dieser Spezifikation werden die dafuer realisierten Schnittstellen zwischen der Anwendung in SINIX und dem DUEAI- Treiber und zwischen dem DUEAI- Treiber und der DUEAI- Software beschrieben.

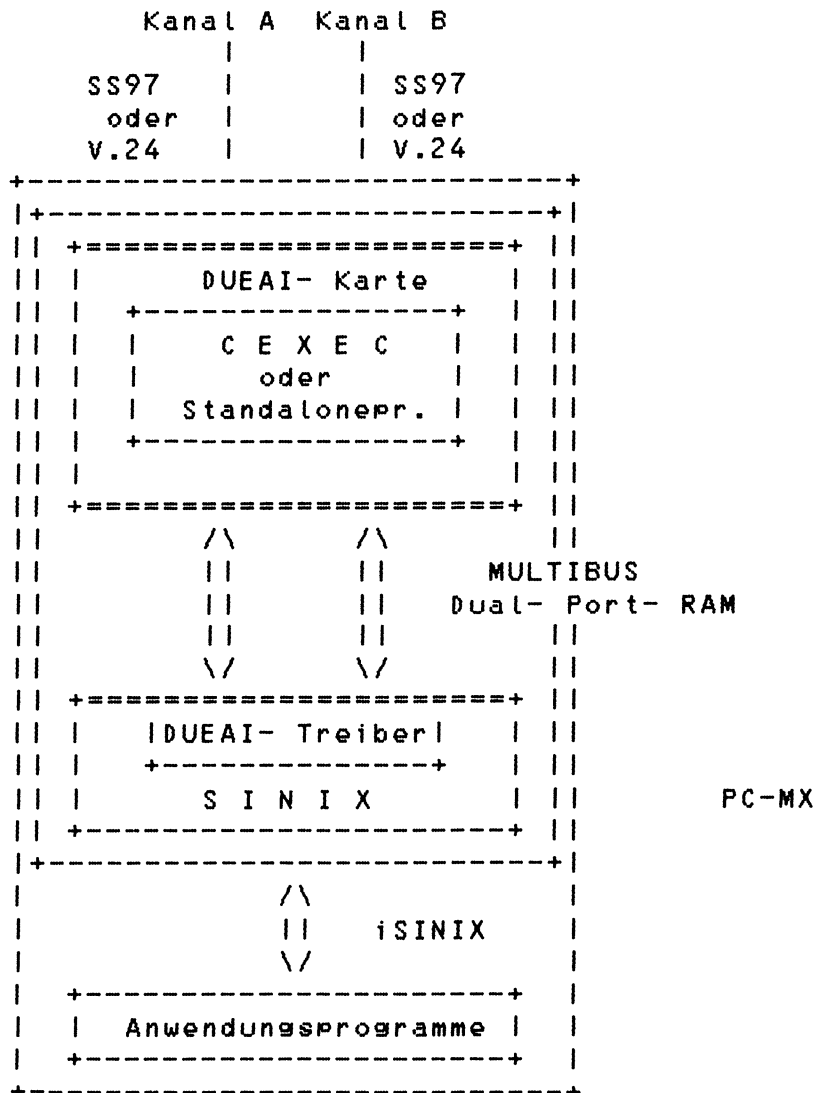
Bearbeiter: Heidbreder, D ST SP412

Mch-P/DG, R431	SIEMENS - DATENVERARBEITUNG	25. 4.1985
Tel. 96-47987	MUENCHEN	Seite 2

2. Leistungsbeschreibung des SINIX-Treibers

2.1. Uebersicht

Der SINIX-Treiber realisiert eine Standard-SINIX Ein/ Ausgabesch-nittstelle zu der DUEAI-Baugruppe, die mit ein - oder zwei Leitung-sanschlussen (Kanaelen) betrieben werden kann.



Folgende SINIX- Aufrufe werden den Anwendungsprogrammen zur Verfügung gestellt:

- open Kontaktaufnahme des Anwendungsprogramms mit dem DUEAI-Treiber
- close Beenden des Kontaktes mit dem DUEAI-Treiber
- read Daten vom DUEAI-Treiber lesen
- write Daten an den DUEAI-Treiber uebergeben
- ioctl Uebergabe von prozedur- und leitungsspezifischen Parametern oder speziellen Kommandos an den DUEAI-Treiber

Die SINIX- Systemaufrufe werden vom DUEAI-Treiber an die DUEAI-Software weitergeleitet.

In der /dev - Directory wird der DUEAI-Treiber z. Beispiel ueber folgende Geraetenamen angesprochen:

	Major#	Minor#	
/dev/d0cold	11	0	⌘ Kaltlader mit dem ein Lader ⌘ auf die Baugruppe gebracht ⌘ wird.
/dev/d0warm	11	2	⌘ Warmlader zum Laden der ⌘ Anwendersoftware.
/dev/d0wins	11	3	⌘ Inspizieren der ersten ⌘ DUEAI-Baugruppe ⌘ (wird nicht von "stmsv1" ⌘ unterstuetzt).
/dev/dfuea00 bis	14	0	⌘ Senden/Empfangen Kanal A bis
/dev/dfuea38	14	38	⌘ Senden/Empfangen Kanal A
/dev/dfadma	14	39	⌘ Administration Kanal A
/dev/dfueb40 bis	14	40	⌘ Senden/Empfangen Kanal B bis
/dev/dfueb78	14	78	⌘ Senden/Empfangen Kanal B
/dev/dfadmb	14	79	⌘ Administration Kanal B

Pro DUEAI-Baugruppe koennen maximal zwei Leitungen unterstuetzt werden.

Die Minor- Device- Nummern 00 bis 39 werden vom DUEAI-Treiber der Leitung A zugeordnet, die Nummern 40 bis 79 der Leitung B. Es besteht also eine feste Zuordnung zwischen der Geraetedatei und dem

Ausgabekanal.

Eine Parametrisierung der Leitungen kann nur ueber die Gerateadresse /dev/dfadma oder /dev/dfadmb (ueber den ioctl- Aufruf) durchgefuehrt werden.

Das Programm "stmsv1" auf der DUEAI-Baugruppe des PC- MX unterstuezt zur Zeit nur die Minordevice-Nummer 0 bis 11 und 39 fuer die Leitung A.

## 2.2. SINIX- Schnittstellen zum DUEAI - Treiber

### 2.2.1. SINIX- Schnittstellen fuer /dev/dfadma oder /dev/dfadmb

Folgende Aufrufe werden vom DUEAI- Software fuer diese Gerateadressen unterstuezt:

- open                    Eroeffnen Administrator
- ioctl                  Setzen, Aendern oder Abfragen der Parameter, moegliche weitere Kommandos
- close                  Schliessen Administrator

#### 2.2.1.1. OPEN - Aufruf

Der OPEN-Aufruf entspricht Standard-Sinix.

Das Administratordevice wird eroeffnet, d.h. dem aufrufenden Prozess wird der Filedeskriptor uebergeben.

Ein Eröffnungswunsch wird abgelehnt, wenn die DUEAI- Baugruppe nicht geladen ist.

#### 2.2.1.2. IOCTL - Aufruf

Der IOCTL-Aufruf entspricht Standard-Sinix.

Die Parameter fuer den ioctl- Aufruf werden an die DUEAI- Software weitergegeben. Nach Quittierung des Kommandos werden die Daten zurueckgelesen und dem Anwendungsprogramm zurueckgegeben.

#### 2.2.1.3. CLOSE - Aufruf

Der CLOSE-Aufruf entspricht Standard-Sinix.

Der Filedeskriptor fuer den Administratorprozess wird geschlossen.

#### 2.2.1.4. READ - Aufruf

----- wird nicht vom DUEAI-Software mit MSV1-Prozedur bei einem Administratordevice unterstuezt -----

Der Aufruf wird mit Returncode "EINVAL" (Fehler) beantwortet !!

2.2.1.5. WRITE - Aufruf

----- wird nicht vom DUEAI-Software mit MSV1-Prozedur bei  
einem Administratordevice unterstuetzt -----

Der Aufruf wird mit Returncode "EINVAL" (Fehler) beantwortet !!

### 2.2.2. SINIX- Schnittstellen fuer /dev/dfueann

Folgende Aufrufe werden vom DUEAI- Treiber fuer diese Gerateadressen unterstuetzt:

- open        Eroeffnen
- close      Schliessen
- read       Empfangsdaten lesen
- write      Auszusendende Daten uebergeben
- ioctl      Parameter setzen

#### 2.2.2.1. OPEN - Aufruf

Der OPEN-Aufruf entspricht Standard-Sinix.  
 Eine SINIX- Anwendung konnektiert sich mit dem DUEAI- Treiber.  
 Der Treiber meldet dieses mit der entsprechenden Device-Adresse an die DUEAI- Software.  
 Der Aufruf wird abgelehnt, wenn die DUEAI- Baugruppe nicht geladen ist. Wenn die Minor- Device- Nummer in einem nicht definierten Bereich liegt, wird der Aufruf mit dem Fehlercode ENXIO (no such device or adress) negativ quittiert.

#### 2.2.2.2. CLOSE - Aufruf

Der CLOSE-Aufruf entspricht Standard-Sinix.  
 Eine SINIX- Anwendung diskonnektiert sich von dem DUEAI-Treiber.  
 Der Treiber uebergibt die entsprechende Device-Adresse an die DUEAI- Software.

#### 2.2.2.3. IOCTL - Aufruf

----- wird nicht von der DUEAI-Software mit MSV1-Prozedur  
 bei einem Anwenderdevice unterstuetzt -----  
 Der Aufruf wird mit Returncode "EINVAL" (Fehler) beantwortet !!

#### 2.2.2.4. READ - Aufruf

Der READ-Aufruf entspricht Standard-Sinix.  
 Die auf der Leitung empfangenen Daten werden an die Anwendung uebergeben. Der read Aufruf liefert als Ergebnis die Anzahl der empfangenen Zeichen oder -1, falls Fehler aufgetreten sind. Dieser Fehler ist dann in errno genauer spezifiziert.

## 2.2.2.5. WRITE - Aufruf

Der WRITE-Aufruf entspricht Standard-Sinix.

Mit "write" werden von der SINIX- Anwendung Daten an die DUEAI-Software zum Aussenden uebergeben. Der write Aufruf liefert als Ergebnis die Anzahl der uebergebenen Zeichen oder -1, falls Fehler aufgetreten sind. Dieser Fehler ist dann in errno genauer spezifiziert.



### 3. Allgemeine Schnittstelle zu Controllerbaugruppen

Es wird eine allgemeine Schnittstelle zu Controllerbaugruppen definiert. Die Schnittstelle umfasst die Aufteilung eines zur Kommunikation dienenden Dual-Port RAM's. Sie wurde bewusst universell ausgelegt, so dass auch andere Baugruppen eine solche Schnittstelle verwenden koennen. Die Normierung der Schnittstelle vereinfacht die Ansteuerung der Baugruppe(n), wodurch wiederum die Treiber vereinfacht werden.

#### 3.1. Dual- Port Ram Aufteilung

Zur Kommunikation der Baugruppen mit dem SINIX- System dient ein Dual-Port RAM Fenster (im folgenden auch DPR genannt).

Jede Baugruppe teilt den DPR-Bereich in eine Anzahl Steuerregister (Softwareregister) und einen Datenbereich, welcher als Puffer dient, auf.

Dual- Port RAM- Ende

Puffer fuer Indicationkontroll-  
bloecke

Puffer fuer Requestkontroll-  
bloecke

Sende/Empf.-buffer f. Laden/Dbg.

+ 96 / 512 Byte

Indicationkontrollblock fuer  
Leitung B  
8 Worte

+ 80

Requestkontrollblock fuer  
Leitung B  
8 Worte

+ 64

Indicationkontrollblock fuer  
Leitung A  
8 Worte

+ 48

Requestkontrollblock fuer  
Leitung A  
8 Worte

+ 32

Kontrollblock fuer Laden/Debug  
8 Worte

+ 16

frei f. Erweiterungen

reservier f. Interruptbits AT11

+ 8

reservier f. Interruptbits AT01

+ 6

Interruptbits (ATTN-Bits) AT10  
(Interrupts DUEAI --> 9780)  
Kontrollbl. 0 -15

+ 4

Interruptbits (ATTN-Bits) AT00  
(Interrupts DUEAI <-- 9780)  
Kontrollbl. 0 -15

+ 2

Anzahl der Kontrollbloecke NCH

Anfang des Dual Port RAM

Aufteilung des Dual-Port-RAM Fensters

\*\*\*\*\*

3.1.1. NCH Register: Anzahl der Kontrollblöcke.

Im NCH Register steht die Anzahl der verwendeten Kontrollblöcke. Diese Anzahl bestimmt die entsprechende Aufteilung des DPR. Der erste Kontrollblock (mit der Nummer 0) ist reserviert zum Laden der Baugruppe.

3.1.2. ATT Register Attention oder Interruptbits

Fordert der Treiber für einen der Kontrollblöcke der Baugruppe einen Interrupt an, so setzt er vor dem Hardwareinterrupt das zu dem Kontrollblock gehörende Bit des ATO-Registers. Die Baugruppe erkennt an diesem Bit, für welchen Kontrollblock eine Aktion nötig ist, und löscht dann das ATO Bit.

Ein zweiter ATT-Registersatz (ATI) signalisiert Interrupts von der Baugruppe an das SINIX System. Das Bit wird von der Baugruppe gesetzt und vom SINIX System gelöscht.

Die Wortlänge von 16 Bit begrenzt die Anzahl möglicher Kontrollblöcke auf 32 (da zwei Worte verwendet werden).

Bearbeiter: Heidbreder, D ST SP412

### 3.2. Einteilung der Kontrollblöcke

Jeder Steuerbereich besteht aus 8 Registern:

Zusatzregister 3 (Auxiliary Register 3) AUX3	Wort 7
Zusatzregister 2 (Auxiliary Register 2) AUX2	Wort 6
Zusatzregister 1 (Auxiliary Register 1) AUX1	Wort 5
Zusatzregister 0 (Auxiliary Register 0) AUX0	Wort 4
Adressregister (Offset Register) OFF	Wort 3
Zählregister (Transfer Count Reg.) TCR	Wort 2
Kommandoregister (Command Register) CR	Wort 1
Statusregister (Status Register) SR	Wort 0

Für jeden Kontrollblock gibt es einen Master, der den Kontrollblock mit Kommandos füllt und den Status später überprüft. Ebenso gibt es einen Slave, der diese Kommandos aus dem Kontrollblock liest, verarbeitet und das Status Register setzt. Für die Requestkontrollblöcke ist der SINIX-Treiber der Master, und das Programm auf der Baugruppe der Slave. Für die Indicationkontrollblöcke ist das Programm auf der Baugruppe der Master und der SINIX-Treiber der Slave.

#### 3.2.1. Statusregister SR

Anhand des Statusregisters kann der Master den Zustand des Kontrollblocks bestimmen, oder der Slave den Zustand erfahren. Das Statusregister wird vom Master vor der Übergabe eines Kommandos beschrieben (gelöscht). Nach der Kommandoausführung werden die Statusbits vom Master überprüft und entsprechende Aktionen ausgeführt.

#### 3.2.2. Kommandoregister CR

In das Kommandoregister schreibt der Master das auszuführende Kommando.

#### 3.2.3. Zählregister TCR

Im Zählregister (TCR) wird bei Kommandos, welche einen Datenblock im Puffer übergeben, die Blocklänge übergeben.

Bearbeiter: Heidbreder, D ST SP412

(= Anzahl der zu uebertragenden Zeichen bzw. Bytes)

Bei "write"-Kommandos schreibt der SINIX-Treiber die Anzahl zu uebergegebener Zeichen in dieses Register.

Bei "read"-Kommandos schreibt der SINIX-Treiber die Anzahl der gewuenschten Zeichen in das Register.

#### 3.2.4. Adressregister OFF

Im Adressregister (OFF) wird bei Kommandos, welche einen Datenblock im Puffer uebergeben, die Pufferadresse als Offset relativ zum Anfang des Dual-Port RAM's uebergeben.

Zusammen mit dem Zaehregister wird hiermit der Datenblock im Puffer genau definiert.

#### 3.2.5. Auxiliary Register AUX0

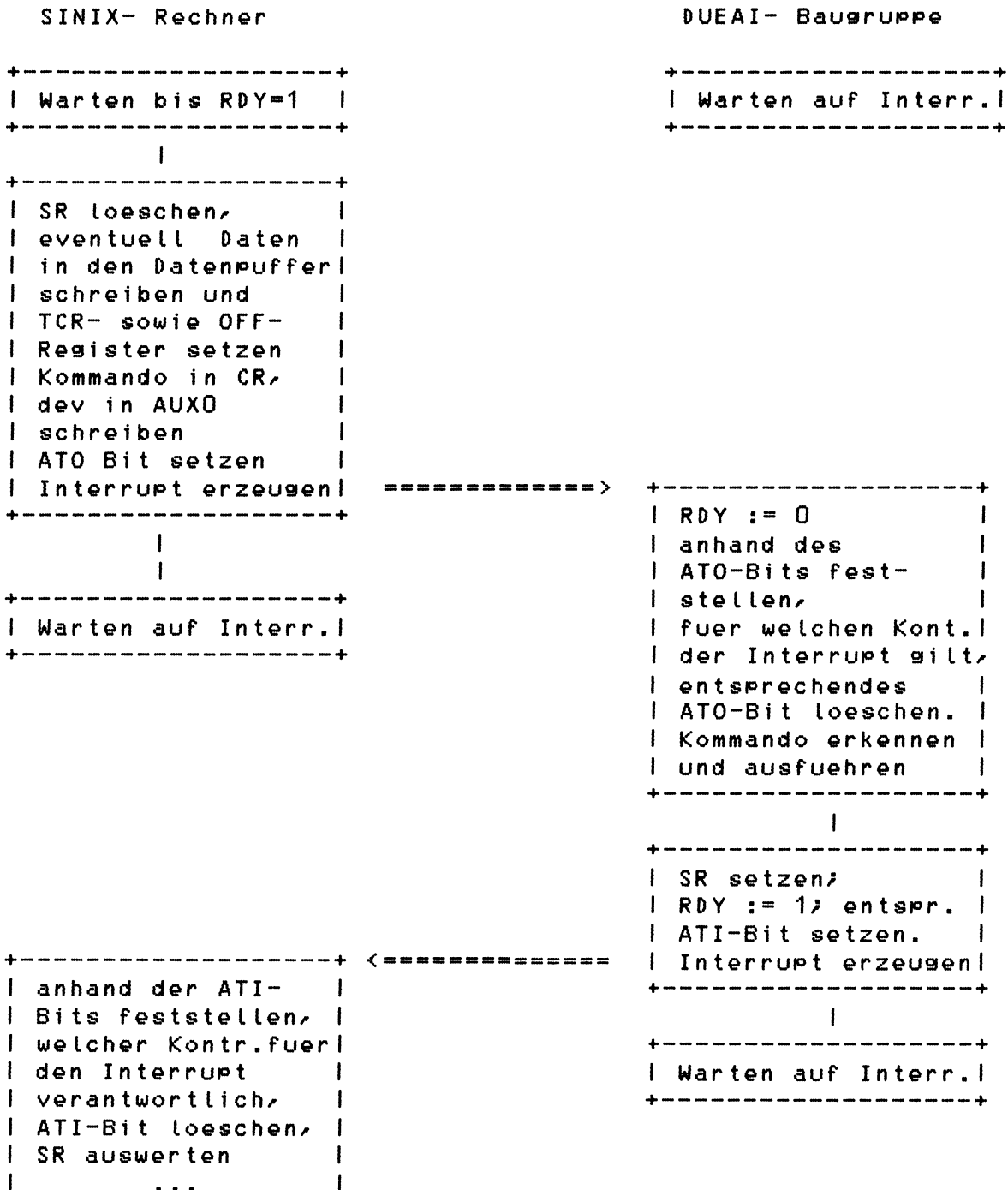
Im Auxiliary-Register 0 ist die Minor-Nummer des betreffenden Geraetes eingetragen.

#### 3.2.6. Zusatzregister AUX1 - AUX3

Die Verwendung der Zusatzregister ist baugruppenspezifisch bzw. kommandospezifisch und wird in der jeweiligen Spezifikation definiert.

### 3.3. Ablaufdiagramme

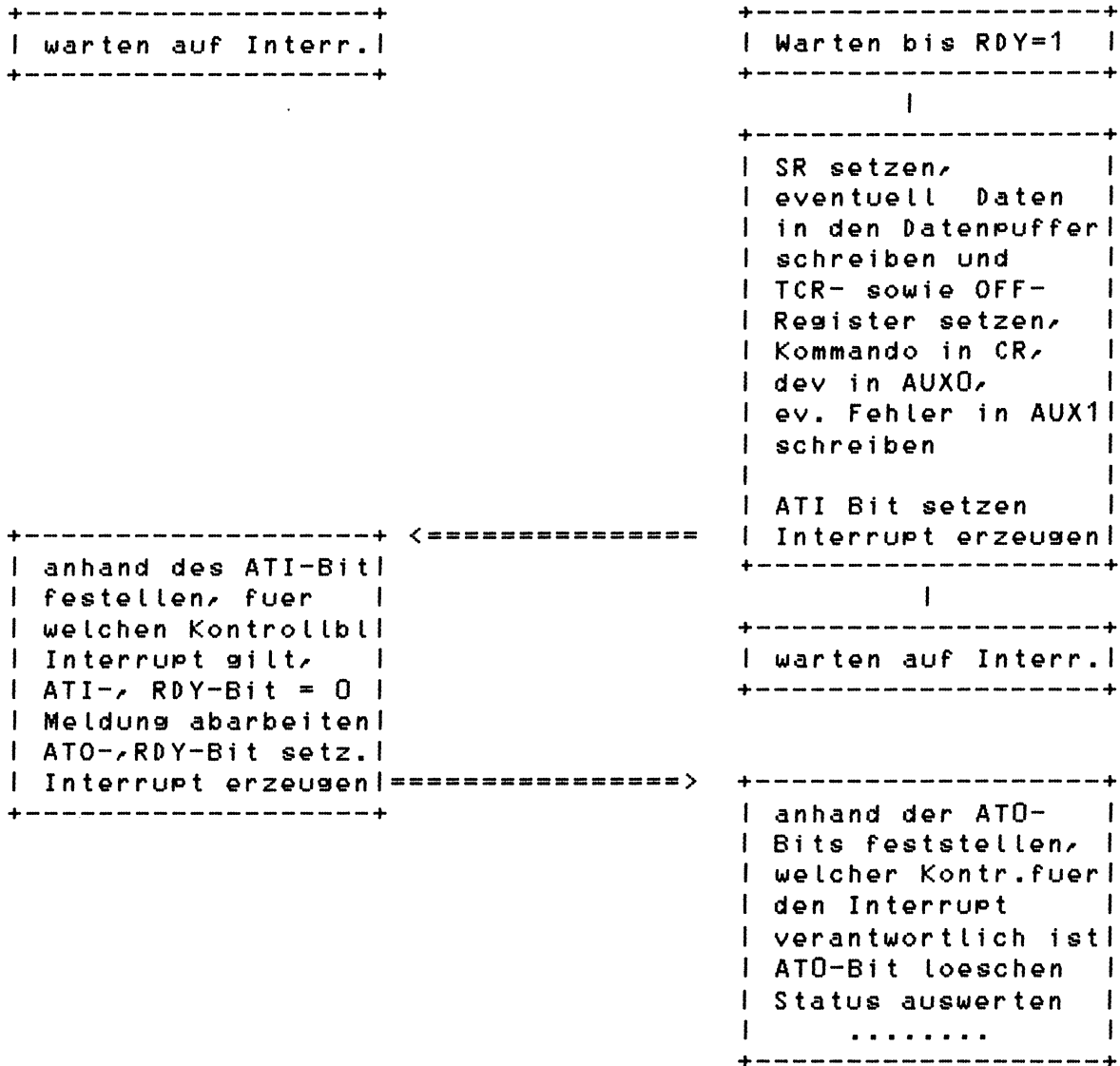
Fuer die Kommandos des Requestkontrollblocks gilt folgendes Ablaufdiagramm:



Fuer die Kommandos des Indicationkontrollblocks gilt folgendes Ablaufdiagramm:

SINIX- Rechner

DUEAI- Baugruppe



#### 4. Kommunikation zw. SINIX-Rechner und DUEAI-Karte (Aufbau und Verwaltung des Dual-Port RAM)

Die Prozesse des SINIX 9780 kommunizieren mit der DUEAI- Software ueber den DUEAI- Treiber.

Dieser Treiber wiederum tauscht Steuerinformation und Daten ueber ein Fenster im Dual- Port RAM mit der Anwendung auf der DUEAI- Baugruppe. Ziel dieser Schnittstelle ist es, die Kommunikationsbloecke moeglichst kurz nur zu belegen, um jede Aenderung an der Schnittstelle zwischen SINIX- Prozess und DUEAI- Treiber der Anwendung auf der DUEAI- Baugruppe mitteilen zu koennen.

##### 4.1. Aufteilung des gem. Speichers auf der DUEAI- Baugruppe

###### 4.1.1. Uebersicht

Die Schnittstelle zur DUEAI- Baugruppe wird im Rahmen der Beschreibung " Allgemeine Schnittstelle zu Controllerbaugruppen " (siehe 3.1) realisiert. Es werden fuer jede physikalische Leitung (hier A- und B- Kanal der DUEAI- Baugruppe) zwei Kontrollbloecke in dem Dual- Port- RAM verwaltet. DPR- Kontrollblock 0 ist fuer die Ladeoperationen reserviert. Deshalb koennen fuer die DUEAI- Baugruppe bis zu 5 Kontrollbloecke festgelegt werden, wobei dann die Pufferbereiche fuer jeden Request- Kontrollblock fest definiert sind.

Bei der MSV1-Prozedur sind zur Zeit 3 Kontrollbloecke (nur Kanal A wird betrieben) festgelegt, es koennten aber spaeter auch 5 werden. Dies wuerde dann eine entsprechend andere Aufteilung des DPR ergeben.

Die Steuerung der Kanale A und B erfolgt wie in 3.1. beschrieben, hier werden lediglich die DUEAI- spezifischen Belegungen definiert.

###### 4.1.1.1. Statusregister SR

Der Aufbau des Statusregisters ist fuer alle realisierten Kontrollbloecke fuer die Kanale (A + B) gleich. Jedes gesetzte Bit hat dabei folgende Lage und Bedeutung:

15	14	13	12	11	10	9	0
ERR		CERR	0	0	BERR		RDY

###### ERR ERROR

Bei der Kommanduebergabe oder der Ausfuehrung des letzten Kommandos trat auf diesem Kontrollblock ein Fehler auf. Dieses Bit wird vom Partner mit Erhalt des naechsten Kommandos auf diesem Kontrollblock geloescht.

Bearbeiter: Heidbreder, D ST SP412



CERR COMMAND ERROR

Das CERR-Bit wird gleichzeitig mit dem ERR-Bit bei Erhalt eines illegalen Kommandos gesetzt.

BERR BUFFER ERROR

Das BERR-Bit wird gleichzeitig mit dem ERR-Bit gesetzt, wenn momentan kein Sendebuffer zur Verfügung steht.

RDY READY

Das RDY Bit ist gesetzt, wenn der Kontrollblock frei ist für ein neues Kommando.

Während der Initialisierungsphase der DUEAI-Baugruppe oder während der Ausführung eines Kommandos ist im Kontrollblock RDY = 0.

Nach der Ausführung eines Kommandos, bevor quittiert wird, werden RDY und ATN vom Slave gesetzt und ein Interrupt erzeugt.

## 4.1.1.2. Kommandoregister CR

Folgende Kommandos werden von den verschiedenen Kontrollblöcken behandelt:

Kommando	-- Kontrollblöcke --	
	Request	Indicat.
CMD_OPEN	x	Fehler
CMD_CLOSE	x	Fehler
CMD_READ	x	Fehler
CMD_DATA	x	Fehler
CMD_WR	x	Fehler
CMD_WCMD	x	Fehler
CMD_IOCTL	x	Fehler
CMD_NOP	x	Fehler
IND_DATA	Fehler	x
IND_WRITE	Fehler	x
IND_ERROR	Fehler	x

x := "wird vom Kontrollblock behandelt"

Die Kommandos werden in den folgenden Kapiteln noch im Detail beschrieben.

## 4.1.1.3. Zaehregister TCR

Im Zaehregister (TCR) wird bei Datenuebergabe (READ/WRITE-Kommandos) die Blocklaenge uebergeben.

(Blocklaenge = Anzahl der zu uebergebenden Bytes)

## 4.1.1.4. Adressregister OFF

Im Adressregister (OFF) wird bei Datenuebergabe (READ/WRITE-Kommandos) die Adresse des Datenblocks im Puffer relativ zum Anfang des Dual-Port RAM's uebergeben.

Da die Pufferbereiche den Kontrollblöcken fest zugeordnet sind, wird hier in jedem Kontrollblock eine Konstante stehen (siehe oben).

Das Adressregister wird nur vom DUEAI- Treiber beschrieben, die DUEAI- Software greift auf dieses nur lesend zu.

#### 4.1.1.5. Auxiliary 0

Im Auxiliary 0 ist die Minornummer des Gerätes (/dev/duex) vermerkt, fuer die das Kommando relevant ist.

#### 4.1.1.6. Auxiliary 1

Im Auxiliary 1 wird von der DUEAI-Software eine moegliche Fehlermeldung eingetragen, die speziell fuer das eine Device bestimmt, das im Auxililiary 0 eingetragen ist.

#### 4.1.1.7. Zusatzregister 2 - 3 AUX2, AUX3

In den Zusatzregister koennen befehlspezifische Werte stehen (bei der MSV1- Prozedur wird nur noch AUX2 ausgenutzt).

4.2. Kommandos / Meldungen zwischen DUEAI- Treiber und DUEAI- Software

4.2.1. Kommandos in den Requestkontrollbloecken

Die Requestkontrollblocke erlauben die Kommandos:

CMD_OPEN	--	Eroeffnungswunsch einer Anwendung
CMD_READ	--	Leseauftrag an den Kanal
CMD_DATA	--	Holen von empfangenen Daten
CMD_WRMD	--	Daten an den Kanal uebergeben mit weiteren Daten
CMD_WR	--	Daten an den Kanal uebergeben, keine weiteren Daten
CMD_IOCTL	--	_IOCTL-Parameter setzen/holen
CMD_NOP	--	Abbruch eines Auftrages
CMD_CLOSE	--	Diskonnektierung einer Anwendung

4.2.1.1. Das CMD\_OPEN- Kommando

Das CMD\_OPEN- Kommando reicht den Eroeffnungswunsch einer SINIX-Anwendung mit der Minordevicenumber vom DUEAI- Treiber zur DUEAI- Software durch.

Es erfolgt keine uebergabe von Daten, d.h. kein Eintrag in TCR und OFF.

Falls die Verbindung auf der Leitung nicht zustande kommen kann, wird dieses von der DUEAI- Software im Status markiert und im AUX1 spezifiziert fuer die Bestaetigung des Kommandos.

Die OPEN- Systemroutine erhaelt dann dies Fehlermeldung aus AUX1.

4.2.1.2. Das CMD\_READ-, CMD\_DATA Kommando

Das CMD\_READ- Kommando wird an die DUEAI- Software uebergeben und von ihr bestaetigt, wenn der UNIX-Prozess den Systemaufruf READ macht. Beim CMD\_READ- Kommando erfolgt keine uebergabe von Daten, jedoch die geforderte Anzahl von Bytes werden im TCR- Register uebergeben.

Treffen fuer diesen Prozess Daten ein, so gibt die DUEAI- Software eine IND\_DATA Indikation an den DUEAI- Treiber ab.

Der DUEAI- Treiber fordert darauf die Daten mit dem CMD\_DATA Kommando an. Die Anzahl der geforderten Datenbytes richtet sich nach der Groesse des Datenpuffers und steht im TCR- Register.

Die DUEAI- Software schreibt darauf die Anzahl von Daten in das DPR, wie im TCR und OFF eingetragen. Ist der Puffer im DPR nicht gross genug, werden mehrere CMD\_DATA Kommandos abgesetzt.

Somit koennen auf der Leitung empfangene Daten dem entsprechenden Prozess (Systemaufruf READ) zur Verfuegung gestellt werden.

#### 4.2.1.3. Das CMD\_WR-, CMD\_WRMD- Kommando

Mit dem WR- und WRMD-Kommando werden Daten von der SINIX-Anwendung ueber den DUEAI-Treiber in einen internen DUEAI-Puffer uebertragen.

Das WRMD-Kommando uebertraegt einen Teil eines Datenblocks vom Datenpuffer in einen internen DUEAI-Puffer. Weitere Teildatenbloecke werden von der DUEAI-Software mit fruerehen Bloecken solange gekettet, bis der letzte Teildatenblock mit dem WR-Kommando uebergeben wurde.

Im Zaehlregister (TCR) steht die Anzahl zu uebertragender Zeichen (= Bytes). Das Adressregister (OFF) gibt die Position des Datenblocks im DPR an.

Ist zur Zeit des WR/WRMD-Kommandos der Sendeuffer schon gefuellt und koennen keine weiteren Sendedatenbloecke von der DUEAI-Software angenommen werden, so quittiert sie diese Kommandos mit den Bits ERR und BERR im Statusregister SR. Es koennen von der DUEAI-Software mit MSV1-Prozedur Datenbloecke mit maximal 4096 Bytes ausgesendet werden.

Sind die Daten gut oder fehlerhaft ausgesendet worden, wird dieses dem DUEAI-Treiber mit der IND\_WRITE Meldung gesagt.

#### 4.2.1.4. Das CMD\_IOCTL- Kommando

Das CMD\_IOCTL-Kommando dient dazu, die Parameterliste der Prozedur MSV1 in das DPR der DUEAI-Baugruppe zu uebertragen.

Der Puffer des SINIX-Prozesses, dessen Groesse der Laenge der Struktur "msv1p" entspricht, wird in das DPR eingetragen.

Im Zaehlregister steht die Laenge der Parameterliste.

Danach wird der DFUEAI-Baugruppe das CMD\_IOCTL Kommando uebergeben.

Nachdem die DUEAI-Software das Kommando bestaetigt hat, wird der gleiche DPR-Puffer mit gleicher Laenge dem SINIX-Prozess zurueckgeschrieben.

Dieses Prinzip, einen Datenblock der DUEAI-Software zu uebergeben und danach die gleiche Datengroesse zurueckzuschreiben, koennte fuer Diagnosemoeglichkeiten verwendet werden, wobei der DUEAI-Treiber neutral bleibt, bei allen moeglichen Kommandos innerhalb des IOCTL-Systemaufrufes.

Der IOCTL-Systemparameter "cmd" steht im AUX2.

Zur Zeit wird vom DUEAI-Treiber und von der DUEAI-Software nur das oben beschriebene Kommando realisiert.

#### 4.2.1.5. Das CMD\_CLOSE- Kommando

Das CMD\_CLOSE-Kommando reicht den Diskonnektierungswunsch fuer eine Anwendung vom DUEAI-Treiber zur DUE-Software durch. Im AUX0 steht die Minordevivenumber der Anwendung.

#### 4.2.2. Meldungen in den Indicationbloecken

Die Meldungen in den Empfangskontrollbloecken sind:

IND_DATA	--	Datenblock lesen
IND_WRITE	--	Datenblock gesendet
IND_ERROR	--	Globale Fehlermeldung an alle beteiligten Prozesse, die mit einer Leitung zusammenarbeiten.

##### 4.2.2.1. IND\_DATA Meldung

Die IND\_DATA Meldung darf nur gesendet werden, wenn zuvor ein CMD\_READ Request abgesetzt wurde. Eine IND\_DATA Indication, die sich nicht auf ein READ Request bezieht, wird vom DUEAI- Treiber mit Fehlerbit ERR quittiert, weil vom Benutzerprozess kein Systemaufruf READ ansteht und deshalb keine Daten weitergeleitet werden koennen. Die IND\_DATA Meldung fordert dazu auf, auf der Leitung empfangene Daten abzuholen, oder aber sie meldet fuer das spezifizierte Device einen aufgetretenen Fehler beim Empfang von Daten (ERR-Bit gesetzt), die Fehlermeldung steht im AUX1.

##### 4.2.2.2. IND\_WRITE Meldung

Die IND\_WRITE Meldung liefert einen Status fuer den ausgesendeten Datenblock fuer das im AUX0 spezifizierte Device. Ist das ERR-Bit nicht gesetzt, sind die Daten gut ausgesendet worden. Ist das ERR-Bit gesetzt, so steht im AUX1 eine Fehlermeldung.

##### 4.2.2.3. IND\_ERROR Meldung

Die IND\_ERROR Meldung wird verwendet, wenn allen Prozessen einer Leitungsgruppe eine Fehlermeldung geliefert werden soll. Die Fehlermeldung steht im AUX1, das ERR-Bit im SR ist gesetzt. Der Inhalt des AUX0- Register ist ohne Bedeutung.

## 5. Laden der DUEAI Baugruppe

Das Laden der DUEAI Baugruppe mit der Anwendung laeuft in zwei Schritten ab. Im ersten Schritt wird ein Warmlader auf die Baugruppe geladen. Dieses Programm muss kleiner als 2K, die Groesse des Kaltladefensters der Baugruppe sein. Das Kaltladefenster ist waehrend der Kaltladephase fest auf die Multibusadresse 0xfb000, dies entspricht 0xb000 lokal auf der DUEAI Baugruppe, definiert. Danach kann das DPR Fenster durch entsprechende Befehle verschoben werden. Ist der Warmlader geladen, kann die eigentliche Anwendung, die MSV1 - Prozedur, geladen werden. Auf der SINIX Seite wird zum Laden ein Treiber benutzt, der die Verbindung zur Slave- Baugruppe DUEAI herstellt. Fuer die zwei Phasen des Ladens werden Geraeteeintraege mit gleicher Majornummer aber unterschiedlichen Minornummern verwendet. Zum Kaltladen wird das Geraet /dev/d0cold, zum Laden der MSV1 - Prozedur wird /dev/d0warm verwendet.

Der Treiber hat eine Standardchnittstelle zum Benutzer mit den Systemaufrufen open, close, read, write, ioctl. Fuer die Benutzer- schnittstelle gibt es zwei Programme "coldload" und "warmload", die die zwei Ladephasen realisieren. Prinzipiell ist es aber auch moeglich UNIX Kommandos dafuer zu verwenden. Der Vorteil von coldload und warmload ist die bessere Verarbeitung der Fehlermeldungen, die von der DUEAI Baugruppe geliefert werden. Bei warmload kommt noch hinzu, dass die Anwendung an beliebige Positionen im lokalen Speicher der Baugruppe geladen werden kann.

Die MSV1 Prozedur wird im lokalen Speicher der Baugruppe auf die Adresse 0xb000 geladen. Damit ist garantiert, dass das Kommunikationsfenster nicht durch das Programm ueberlagert wird.

### 5.1. Kommandos zum Laden der DUEAI- Software

Nach dem Laden des Warmladers auf die Baugruppe wird zwischen SINIX und der Baugruppe mit PEEK, POKE und START Kommandos gearbeitet. Die Systemaufrufe read, write und close werden auf diese Kommandos abgebildet.

Mit diesen Kommandos wird das Anwendungsprogramm geladen. Zusaet- zlich zu den Geraeten /dev/d0cold und /dev/d0warm gibt es die Geraete /dev/d0cins und /dev/d0wins. Ueber "/dev/d0cins" kann der DPR- Speicher der DUEAI- Baugruppe untersucht werden. Auf /dev/d0wins kann nur zugegriffen, wenn das Anwendungsprogramm die PEEK- und POKE- Kommandos unterstuetzt ( bei der MSV1- Prozedur nicht realisiert ). Der Zugriff auf den Speicher sieht fuer den Benutzer wie der Zugriff auf "/dev/mem" aus, so dass auch UNIX Utility-Programme mit diesem arbeiten koennen:

Zum Beispiel:

xd /dev/dOcins	Dump des DPR der DUEAI- Baugruppe
adb /dev/dOwins	Debugger zum Patchen

Die im folgenden beschriebenen Kommandos werden (von der DUEAI-Software) nur auf dem Ladekanal realisiert.

## 5.2. Synchronisation zwischen SINIX und DUEAI Baugruppe.

Das DPR ist strukturiert. Es enthaelt mehrere Kontrollbloেকে, ueber die Kommandos abgesetzt werden koennen. Fuer das Laden und Inspizieren wird nur der Kommandokanal 0 verwendet, waehrend fuer die eigentliche Anwendung die uebrigen Kontrollbloেকে verwendet werden. Fuer die Synchronisation des einzelnen Kontrollblocks gibt es ein READY Flag, das von dem Programm auf der Baugruppe gesetzt wird, wenn ein Kommando quittiert wird. Wird ein Kommando abgesetzt, wird das READY Flag geloescht.

Ein Kommando laeuft folgendermassen ab:

- (1) Warten bis das READY Flag gesetzt ist.
- (2) Kommando und Parameter in die Kontrollblock im DPR eintragen.
- (3) Interrupt Flag fuer diesen Kontrollblock im DPR setzten, damit die Anwendung weiss, fuer welchen Kontrollblock eine Anforderung ansteht.
- (4) Interrupt fuer die Baugruppe generieren.
- (5) Das Programm auf der Baugruppe erhaelt den Interrupt, setzt das READY Flag zurueck, und fuehrt das Kommando aus.
- (6) Das Status Register fuer diesen Kontrollblock wird gesetzt.
- (7) Danach wird das READY Flag gesetzt, und ein Interrupt erzeugt.

### 5.2.1. PEEK Kommando

Das PEEK-Kommando liest einen Bereich des DUEAI- Speichers und kopiert diesen in den DPR-Puffer.

Die (absolute) Adresse wird in den AUX0- und AUX1-Registern uebergeben. Hierbei stehen im AUX0- Register die hoeherwertigen, im AUX1- Register die niederwertigen Bits der Adresse.

Die Laenge des uebergebenen Blocks steht im TCR-Register, die DPR-Pufferadresse im OFF-Register.

### 5.2.2. POKE Kommando

Das POKE-Kommando kopiert einen Block aus dem DPR-Puffer in den DUEAI- Speicher.

Die (absolute) Adresse wird in den AUX0- und AUX1-Registern - wie



oben - uebergeben; Laenge und Pufferadresse stehen in TCR und OFF.

### 5.2.3. START Kommando

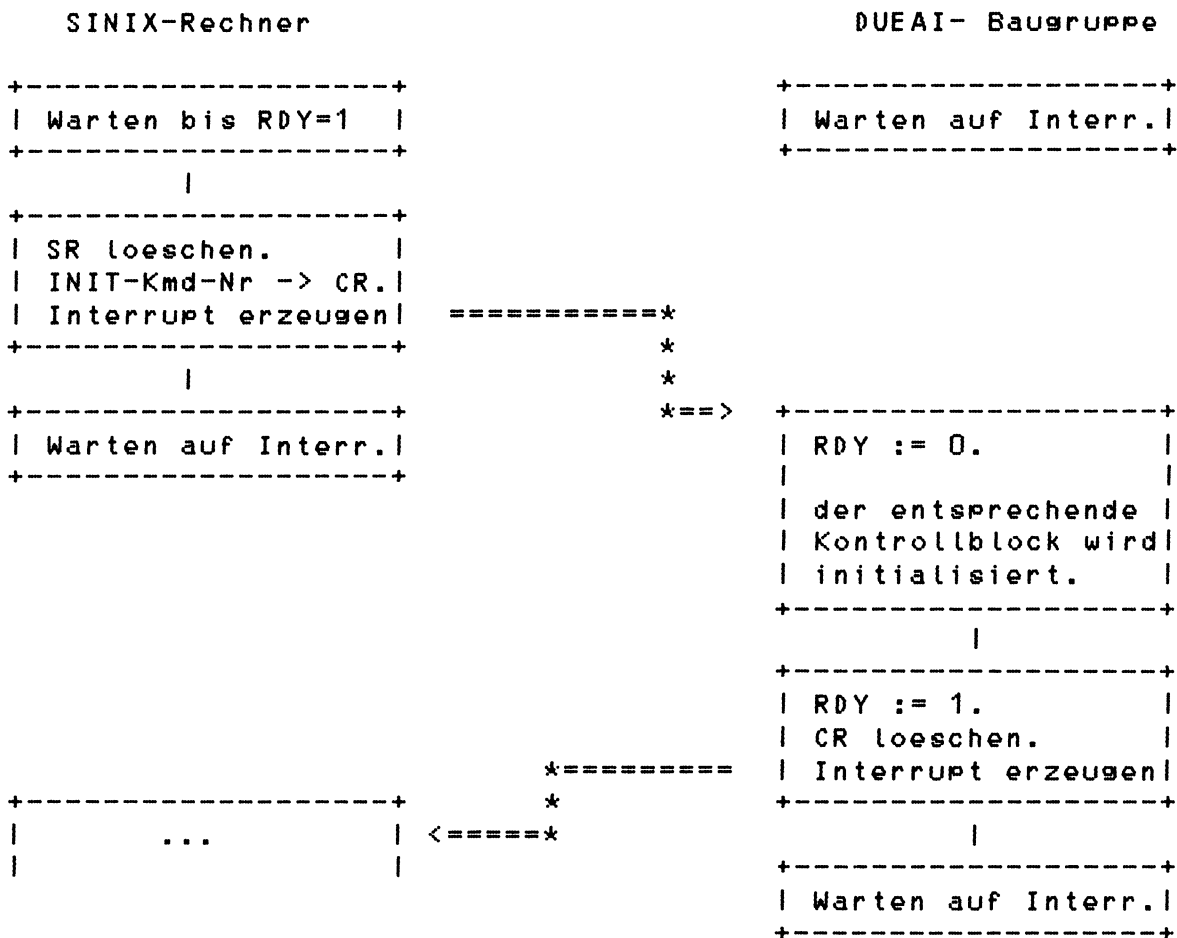
Das START-Kommando startet ein Programm ab der durch AUX0 und AUX1 angegebenen Adresse. Das gestartete Programm sollte die Bedienung des DPR's in gleicher Weise wie bisher sicherstellen, damit mit "/dev/d0wins" weiter gearbeitet werden kann.

### 5.3. Ablaufdiagramme

#### 5.3.1. NOP-Kommando

Ein NOP-Kommando wird von der DUEAI- Baugruppe sofort mit RDY und einem Interrupt quittiert.

#### 5.3.2. INIT-Kommando



5.3.3. PEEK-Kommando (nur Ladekanal)

SINIX-Rechner

DUEAI- Baugruppe

```
+-----+
| Warten bis RDY=1 |
+-----+
```

```
+-----+
| Warten auf Interr.|
+-----+
```

```
|
+-----+
| SR loeschen.      |
| Peek-Kommando     |
| -> CR?            |
| Pufferadresse->OFF|
| Anzahl -> TCR     |
| Absolutadr.->AUXD |
| und AUX1?         |
| Interrupt erzeugen|
+-----+
```

=====\*

\*

\*

\*====>

```
|
+-----+
| Warten auf Interr.|
+-----+
```

```
+-----+
| RDY := 0.          |
|                   |
| Datenblock aus dem|
| Speicher in den   |
| Pufferbereich     |
| kopieren?         |
| Adresse in AUXD,1|
| Laenge in TCR?   |
| Pufferadr. in OFF |
+-----+
```

```
|
+-----+
| ...                |
| ...                |
| Daten aus dem     |
| Pufferbereich     |
| lesen und an den  |
| Benutzer geben   |
| ...              |
+-----+
```

\*=====\*

\*

<=====\*

```
+-----+
| RDY := 1.          |
| CR loeschen.      |
| Interrupt erzeugen|
+-----+
```

```
+-----+
| Warten auf Interr.|
+-----+
```

5.3.4. POKE-Kommando (nur Ladekanal)

SINIX-Rechner

DUEAI- Baugruppe

```

+-----+
| Warten bis RDY=1 |
+-----+
|
+-----+
| SR loeschen      |
| Zu uebertragenden|
| Datenblock in den|
| Puffer schreiben.|
|
| POKE-Kommando->CR|
| Adresse -> AUX0   |
| und AUX1          |
| Laenge -> TCR    |
| Pufferadr. -> OFF|
|
| Interrupt erzeugen|
+-----+

```

```

+-----+
| Warten auf Interr.|
+-----+

```

```

+-----+
| Warten auf Interr.|
+-----+

```

```

=====*
*
*
*====>

```

```

+-----+
| RDY := 0.         |
|
| Datenblock aus dem|
| Pufferbereich in |
| den Speicher      |
| kopieren?         |
| Adresse in AUX0,1|
| Laenge in TCR?   |
| Pufferadr. in OFF|
+-----+

```

```

+-----+
| ...              |
| ...              |
+-----+

```

```

*=====  
*  
<====*

```

```

+-----+
| RDY := 1         |
| CR Loeschen      |
|
| Interrupt erzeugen|
|
+-----+
| Warten auf Interr.|
+-----+

```

Bearbeiter: Heidbreder, D ST SP412

5.3.5. START-Kommando

SINIX-Rechner

DUEAI- Baugruppe

```

+-----+
| Warten bis RDY=1 |
+-----+
|
+-----+
| SR loeschen      |
| START-Kommando   |
| in CR,           |
| Startadresse -> |
| AUX0, AUX1,      |
|
| Interrupt erzeugen|
+-----+
|
+-----+
| Warten auf Interr.|
+-----+

```

```

+-----+
| Warten auf Interr.|
+-----+
|
+-----+
| RDY := 0.         |
|
| Adresse aus AUX0,1|
| vermerken        |
+-----+
|
+-----+
| RDY := 1.         |
| CR loeschen      |
|
| Interrupt erzeugen|
+-----+
|
+-----+
| Sprung zu obiger |
| Adresse           |
+-----+

```

```

+-----+
| ...             |
| ...             |
+-----+

```

```

=====*
*
*
*====>
*
*
*====*
*
<====*

```

Die weitere Kontrolle ueber das DPR uebernimmt dann das zuvor mit dem Warmlader geladene Programm.

9780 SOFTWARE-ENTWICKLUNGS-PROJEKT-BAND	Version 1
BAND: 9780 SINIX	
KAPITEL 70: Produktspezifikation	Verteiler:
ABSCHNITT 80: DUE- System, DUEAI-Treiber f. MSV1	siehe Deckblatt

Bearbeiter: Heidbreder, D ST SP412	
------------------------------------	--

Mch-P/DG, R431	SIEMENS - DATENVERARBEITUNG	25. 4.1985
Tel. 96-47987	MUENCHEN	Seite 31

# Z8530 SCC Serial Communications Controller

# Zilog

## Product Specification

September 1983

### Features

- Two independent, 0 to 1M bit/second, full-duplex channels, each with a separate crystal oscillator, baud rate generator, and Digital Phase-Locked Loop for clock recovery.
- Multi-protocol operation under program control; programmable for NRZ, NRZI, or FM data encoding.
- Asynchronous mode with five to eight bits and one, one and one-half, or two stop bits per character; programmable clock factor; break detection and generation; parity, overrun, and framing error detection.
- Synchronous mode with internal or external character synchronization on one or two synchronous characters and CRC generation and checking with CRC-16 or CRC-CCITT preset to either 1s or 0s.
- SDLC/HDLC mode with comprehensive frame-level control, automatic zero insertion and deletion, I-field residue handling, abort generation and detection, CRC generation and checking, and SDLC Loop mode operation.
- Local Loopback and Auto Echo modes.

### General Description

The Z8530 SCC Serial Communications Controller is a dual-channel, multi-protocol data communications peripheral designed for use with conventional non-multiplexed buses. The SCC functions as a serial-to-parallel, parallel-to-serial converter/controller. The SCC can be software-configured to satisfy a

wide variety of serial communications applications. The device contains a variety of new, sophisticated internal functions including on-chip baud rate generators, Digital Phase-Locked Loops, and crystal oscillators that dramatically reduce the need for external logic.

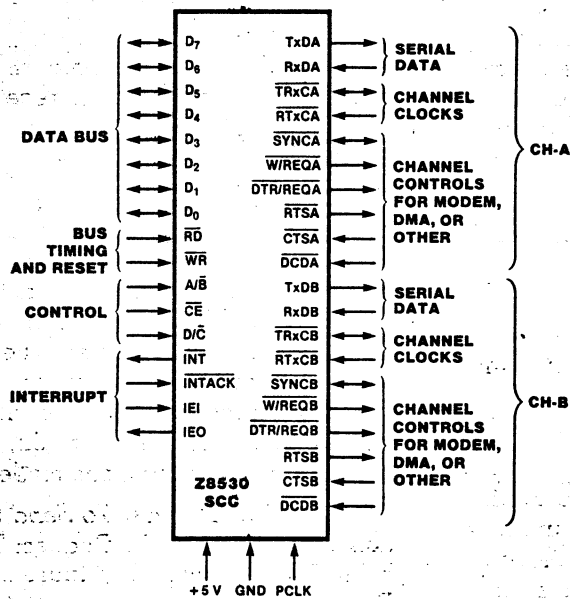


Figure 1. Pin Functions

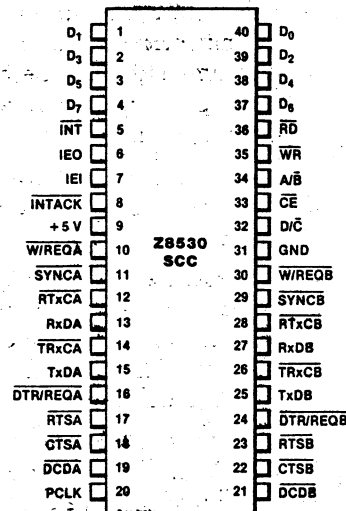


Figure 2. Pin Assignments

Z8530 SCC

**General Description**  
(Continued)

The SCC handles asynchronous formats, Synchronous byte-oriented protocols such as IBM Bisync, and Synchronous bit-oriented protocols such as HDLC and IBM SDLC. This versatile device supports virtually any serial data transfer application (cassette, diskette, tape drives, etc.).

The device can generate and check CRC codes in any Synchronous mode and can be programmed to check data integrity in various modes. The SCC also has facilities for

modem controls in both channels. In applications where these controls are not needed the modem controls can be used for general-purpose I/O.

The Z-Bus daisy-chain interrupt hierarchy also supported—as is standard for Zilog peripheral components.

The Z8530 SCC is packaged in a 40-pin ceramic DIP and uses a single +5 V power supply.

**Pin Description**

The following section describes the pin functions of the SCC. Figures 1 and 2 detail the respective pin functions and pin assignments.

**A/B.** *Channel A/Channel B Select* (input). This signal selects the channel in which the read or write operation occurs.

**CE.** *Chip Enable* (input, active Low). This signal selects the SCC for a read or write operation.

**CTSA, CTSB.** *Clear To Send* (inputs, active Low). If these pins are programmed as Auto Enables, a Low on the inputs enables the respective transmitters. If not programmed as Auto Enables, they may be used as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow rise-time inputs. The SCC detects pulses on these inputs and can interrupt the CPU on both logic level transitions.

**D/C.** *Data/Control Select* (input). This signal defines the type of information transferred to or from the SCC. A High means data is transferred; a Low indicates a command.

**DCDA, DCDB.** *Data Carrier Detect* (inputs, active Low). These pins function as receiver enables if they are programmed for Auto Enables; otherwise they may be used as general-purpose input pins. Both pins are Schmitt-trigger buffered to accommodate slow rise-time signals. The SCC detects pulses on these pins and can interrupt the CPU on both logic level transitions.

**D<sub>0</sub>-D<sub>7</sub>.** *Data Bus* (bidirectional, 3-state). These lines carry data and commands to and from the SCC.

**DTR/REQA, DTR/REQB.** *Data Terminal Ready/Request* (outputs, active Low). These outputs follow the state programmed into the DTR bit. They can also be used as general-purpose outputs or as Request lines for a DMA controller.

**IEI.** *Interrupt Enable In* (input, active High). IEI is used with IEO to form an interrupt daisy chain when there is more than one interrupt-driven device. A High IEI indicates that no other higher priority device has an interrupt under service or is requesting an interrupt.

**IEO.** *Interrupt Enable Out* (output, active High). IEO is High only if IEI is High and the CPU is not servicing an SCC interrupt or the SCC is not requesting an interrupt (Interrupt Acknowledge cycle only). IEO is connected to the next lower priority device's IEI input and thus inhibits interrupts from lower priority devices.

**INT.** *Interrupt Request* (output, open-drain, active Low). This signal is activated when the SCC requests an interrupt.

**INTACK.** *Interrupt Acknowledge* (input, active Low). This signal indicates an active Interrupt Acknowledge cycle. During this cycle, the SCC interrupt daisy chain settles. When  $\overline{RD}$  becomes active, the SCC places an interrupt vector on the data bus (if IEI is High). INTACK is latched by the rising edge of PCLK.

**PCLK.** *Clock* (input). This is the master SCC clock used to synchronize internal signals. PCLK is a TTL level signal.

**RD.** *Read* (input, active Low). This signal indicates a read operation and when the SCC is selected, enables the SCC's bus drivers. During the Interrupt Acknowledge cycle, this signal gates the interrupt vector onto the bus if the SCC is the highest priority device requesting an interrupt.

**RxDA, RxDB.** *Receive Data* (inputs, active High). These input signals receive serial data at standard TTL levels.

**RTxCA, RTxCB.** *Receive/Transmit Clocks* (inputs, active Low). These pins can be programmed in several different modes of operation. In each channel, RTxC may supply the receive clock, the transmit clock, the clock for the baud rate generator, or the clock for the Digital Phase-Locked Loop. These pins can also be programmed for use with the respective SYNC pins as a crystal oscillator. The receive clock may be 1, 16, 32, or 64 times the data rate in Asynchronous modes.

**RTSA, RTSB.** *Request To Send* (outputs, active Low). When the Request To Send (RTS) bit in Write Register 5 (Figure 11) is set, the RTS signal goes Low. When the RTS bit is reset in the Asynchronous mode and Auto



**Description**  
(continued)

Enable is on, the signal goes High after the transmitter is empty. In Synchronous mode or in Asynchronous mode with Auto Enable off, the RTS pin strictly follows the state of the RTS bit. Both pins can be used as general-purpose outputs.

**SYNCA, SYNCB.** *Synchronization* (inputs or outputs, active Low). These pins can act either as inputs, outputs, or part of the crystal oscillator circuit. In the Asynchronous Receive mode (crystal oscillator option not selected), these pins are inputs similar to **CTS** and **DCD**. In this mode, transitions on these lines affect the state of the Synchronous/Hunt status bits in Read Register 0 (Figure 10) but have no other function.

In External Synchronization mode with the crystal oscillator not selected, these lines also act as inputs. In this mode, **SYNC** must be driven Low two receive clock cycles after the last bit in the synchronous character is received. Character assembly begins on the rising edge of the receive clock immediately preceding the activation of **SYNC**.

In the Internal Synchronization mode (Monosync and Bisync) with the crystal oscillator not selected, these pins act as outputs and are active only during the part of the receive clock cycle in which synchronous characters are recognized. The synchronous

condition is not latched, so these outputs are active each time a synchronization pattern is recognized (regardless of character boundaries). In SDLC mode, these pins act as outputs and are valid on receipt of a flag.

**TxDA, TxDB.** *Transmit Data* (outputs, active High). These output signals transmit serial data at standard TTL levels.

**TRxCA, TRxCB.** *Transmit/Receive Clocks* (inputs or outputs, active Low). These pins can be programmed in several different modes of operation. **TRxC** may supply the receive clock or the transmit clock in the input mode or supply the output of the Digital Phase-Locked Loop, the crystal oscillator, the baud rate generator, or the transmit clock in the output mode.

**WR.** *Write* (input, active Low). When the SCC is selected, this signal indicates a write operation. The coincidence of **RD** and **WR** is interpreted as a reset.

**W/REQA, W/REQB.** *Wait/Request* (outputs, open-drain when programmed for a Wait function, driven High or Low when programmed for a Request function). These dual-purpose outputs may be programmed as Request lines for a DMA controller or as Wait lines to synchronize the CPU to the SCC data rate. The reset state is Wait.

Z8530 SCC

**Functional Description**

The functional capabilities of the SCC can be described from two different points of view: as a data communications device, it transmits and receives data in a wide variety of data communications protocols; as a microprocessor peripheral, the SCC offers valuable features such as vectored interrupts, polling, and simple handshake capability.

**Data Communications Capabilities.** The SCC provides two independent full-duplex channels programmable for use in any common Asynchronous or Synchronous data-communication protocol. Figure 3 and the

following description briefly detail these protocols.

**Asynchronous Modes.** Transmission and reception can be accomplished independently on each channel with five to eight bits per character, plus optional even or odd parity. The transmitters can supply one, one-and-a-half, or two stop bits per character and can provide a break output at any time. The receiver break-detection logic interrupts the CPU both at the start and at the end of a received break. Reception is protected from spikes by a transient spike-rejection

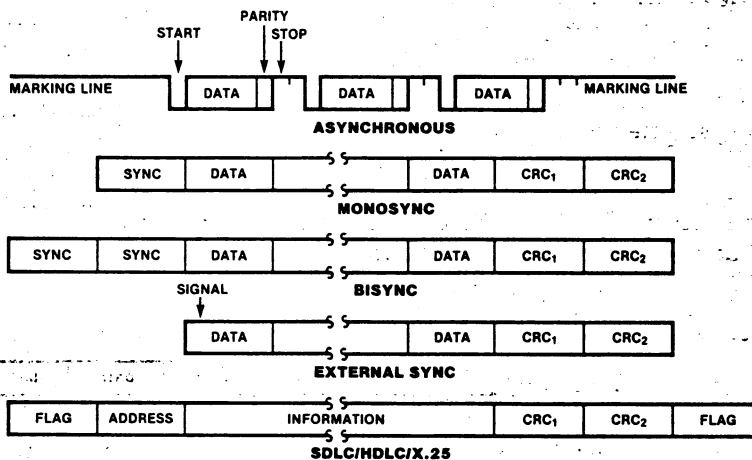


Figure 3. Some SCC Protocols

**Functional Description**  
(Continued)

mechanism that checks the signal one-half a bit time after a Low level is detected on the receive data input (RxDA or RxDB in Figure 1). If the Low does not persist (as in the case of a transient), the character assembly process does not start.

Framing errors and overrun errors are detected and buffered together with the partial character on which they occur. Vectored interrupts allow fast servicing or error conditions using dedicated routines. Furthermore, a built-in checking process avoids the interpretation of a framing error as a new start bit: a framing error results in the addition of one-half a bit time to the point at which the search for the next start bit begins.

The SCC does not require symmetric transmit and receive clock signals—a feature allowing use of the wide variety of clock sources. The transmitter and receiver can handle data at a rate of 1, 1/16, 1/32, or 1/64 of the clock rate supplied to the receive and transmit clock inputs. In Asynchronous modes, the SYNC pin may be programmed as an input used for functions such as monitoring a ring indicator.

*Synchronous Modes.* The SCC supports both byte-oriented and bit-oriented synchronous communication. Synchronous byte-oriented protocols can be handled in several modes, allowing character synchronization with a 6-bit or 8-bit synchronous character (Monosync), any 12-bit synchronization pattern (Bisync), or with an external synchronous signal. Leading sync characters can be removed without interrupting the CPU.

Five- or 7-bit synchronous characters are detected with 8- or 16-bit patterns in the SCC by overlapping the larger pattern across multiple incoming synchronous characters as shown in Figure 4.

CRC checking for Synchronous byte-oriented modes is delayed by one character time so that the CPU may disable CRC checking on specific characters. This permits the implementation of protocols such as IBM Bisync.

Both CRC-16 ( $X^{16} + X^{15} + X^2 + 1$ ) and CCITT ( $X^{16} + X^{12} + X^5 + 1$ ) error checking polynomials are supported. Either polynomial may be selected in all Synchronous modes. Users may preset the CRC generator and checker to all 1s or all 0s. The SCC also provides a feature that automatically transmits CRC data when no other data is available for

transmission. This allows for high speed transmissions under DMA control, with no need for CPU intervention at the end of a message. When there is no data or CRC to send in Synchronous modes, the transmitter inserts 6-, 8-, or 16-bit synchronous characters, regardless of the programmed character length.

The SCC supports Synchronous bit-oriented protocols, such as SDLC and HDLC, by performing automatic flag sending, zero insertion, and CRC generation. A special command can be used to abort a frame in transmission. At the end of a message, the SCC automatically transmits the CRC and trailing flag when the transmitter underruns. The transmitter may also be programmed to send an idle line consisting of continuous flag characters or a steady marking condition.

If a transmit underrun occurs in the middle of a message, an external/status interrupt warns the CPU of this status change so that an abort may be issued. The SCC may also be programmed to send an abort itself in case of an underrun, relieving the CPU of this task. One to eight bits per character can be sent, allowing reception of a message with no prior information about the character structure in the information field of a frame.

The receiver automatically acquires synchronization on the leading flag of a frame in SDLC or HDLC and provides a synchronization signal on the SYNC pin (an interrupt can also be programmed). The receiver can be programmed to search for frames addressed by a single byte (or four bits within a byte) of a user-selected address or to a global broadcast address. In this mode, frames not matching either the user-selected or broadcast address are ignored. The number of address bytes can be extended under software control. For receiving data, an interrupt on the first received character, or an interrupt on every character, or on special condition only (end-of-frame) can be selected. The receiver automatically deletes all 0s inserted by the transmitter during character assembly. CRC is also calculated and is automatically checked to validate frame transmission. At the end of transmission, the status of a received frame is available in the status registers. In SDLC mode, the SCC must be programmed to use the SDLC CRC polynomial, but the generator and checker may be preset to all 1s or all 0s.

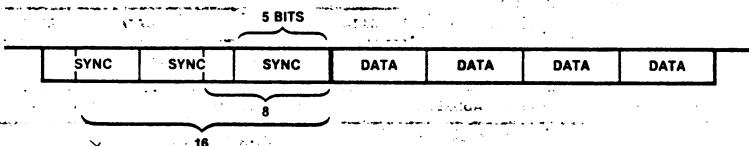


Figure 4. Detecting 5- or 7-Bit Synchronous Characters

The CRC is inverted before transmission and the receiver checks against the bit pattern 0001110100001111.

NRZ, NRZI or FM coding may be used in any 1x mode. The parity options available in Asynchronous modes are available in Synchronous modes.

The SCC can be conveniently used under DMA control to provide high speed reception or transmission. In reception, for example, the SCC can interrupt the CPU when the first character of a message is received. The CPU then enables the DMA to transfer the message to memory. The SCC then issues an end-of-frame interrupt and the CPU can check the status of the received message. Thus, the CPU is freed for other service while the message is being received. The CPU may also enable the DMA first and have the SCC interrupt only on end-of-frame. This procedure allows all data to be transferred via the DMA.

**SDLC Loop Mode.** The SCC supports SDLC Loop mode in addition to normal SDLC. In an SDLC Loop, there is a primary controller station that manages the message traffic flow on the loop and any number of secondary stations. In SDLC Loop mode, the SCC performs the functions of a secondary station while an SCC operating in regular SDLC mode can act as a controller (Figure 5).

A secondary station in an SDLC Loop is always listening to the messages being sent around the loop, and in fact must pass these messages to the rest of the loop by retransmitting them with a one-bit-time delay. The secondary station can place its own message on the loop only at specific times. The controller signals that secondary stations may transmit messages by sending a special character, called an EOP (End Of Poll), around the loop. The EOP character is the bit pattern 11111110. Because of zero insertion during messages, this bit pattern is unique and easily recognized.

When a secondary station has a message to transmit and recognizes an EOP on the line, it

changes the last binary 1 of the EOP to a 0 before transmission. This has the effect of turning the EOP into a flag sequence. The secondary station now places its message on the loop and terminates the message with an EOP. Any secondary stations further down the loop with messages to transmit can then append their messages to the message of the first secondary station by the same process. Any secondary stations without messages to send merely echo the incoming messages and are prohibited from placing messages on the loop (except upon recognizing an EOP).

SDLC Loop mode is a programmable option in the SCC. NRZ, NRZI, and FM coding may all be used in SDLC Loop mode.

**Baud Rate Generator.** Each channel in the SCC contains a programmable baud rate generator. Each generator consists of two 8-bit time constant registers that form a 16-bit time constant, a 16-bit down counter, and a flip-flop on the output producing a square wave. On startup, the flip-flop on the output is set in a High state, the value in the time constant register is loaded into the counter, and the counter starts counting down. The output of the baud rate generator toggles upon reaching 0, the value in the time constant register is loaded into the counter, and the process is repeated. The time constant may be changed at any time, but the new value does not take effect until the next load of the counter.

The output of the baud rate generator may be used as either the transmit clock, the receive clock, or both. It can also drive the Digital Phase-Locked Loop (see next section).

If the receive clock or transmit clock is not programmed to come from the TRxC pin, the output of the baud rate generator may be echoed out via the TRxC pin.

The following formula relates the time constant to the baud rate (the baud rate is in bits/second and the BR clock period is in seconds):

$$\text{baud rate} = \frac{1}{2 (\text{time constant} + 2) \times (\text{BR clock period})}$$

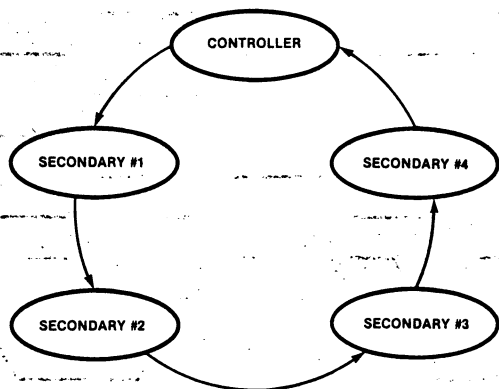


Figure 5. An SDLC Loop

**Digital Phase-Locked Loop.** The SCC contains a Digital Phase-Locked-Loop (DPLL) to recover clock information from a data stream with NRZI or FM encoding. The DPLL is driven by a clock that is nominally 32 (NRZI) or 16 (FM) times the data rate. The DPLL uses this clock, along with the data stream, to construct a clock for the data. This clock may then be used as the SCC receive clock, the transmit clock, or both.

For NRZI encoding, the DPLL counts the 32x clock to create nominal bit times. As the 32x clock is counted, the DPLL is searching the

**Functional Description**  
(Continued)

incoming data stream for edges (either 1 to 0 or 0 to 1). Whenever an edge is detected, the DPLL makes a count adjustment (during the next counting cycle), producing a terminal count closer to the center of the bit cell.

For FM encoding, the DPLL still counts from 0 to 31, but with a cycle corresponding to two bit times. When the DPLL is locked, the clock edges in the data stream should occur between counts 15 and 16 and between counts 31 and 0. The DPLL looks for edges only during a time centered on the 15 to 16 counting transition.

The 32x clock for the DPLL can be programmed to come from either the RTxC input or the output of the baud rate generator. The DPLL output may be programmed to be echoed out of the SCC via the TRxC pin (if this pin is not being used as an input).

**Data Encoding.** The SCC may be programmed to encode and decode the serial data in four different ways (Figure 6). In NRZ encoding, a 1 is represented by a High level and a 0 is represented by a Low level. In NRZI encoding, a 1 is represented by no change in level and a 0 is represented by a change in level. In FM1 (more properly, bi-phase mark), a transition occurs at the beginning of every bit cell. A 1 is represented by an additional transition at the center of the bit cell and a 0 is represented by no additional transition at the center of the bit cell. In FM0 (bi-phase space), a transition occurs at the beginning of every bit cell. A 0 is represented by an additional transition at the center of the bit cell, and a 1 is represented by no additional transition at the center of the bit cell. In addition to these four methods, the SCC can be used to decode Manchester (bi-phase level) data by using the DPLL in the FM mode and programming the receiver for NRZ data. Manchester encoding always produces a transition at the center of the bit cell. If the transition is 0 to 1, the bit is a 0. If the transition is 1 to 0, the bit is a 1.

**Auto Echo and Local Loopback.** The SCC is capable of automatically echoing everything it receives. This feature is useful mainly in Asynchronous modes, but works in Synchronous and SDLC modes as well. In Auto Echo mode, TxD is RxD. Auto Echo mode can be used with NRZI or FM encoding with no additional delay, because the data stream is not decoded before retransmission. In Auto Echo mode, the CTS input is ignored as a transmitter enable (although transitions on this input can still cause interrupts if programmed to do so). In this mode, the transmitter is actually bypassed and the programmer is responsible for disabling transmitter interrupts and WAIT/REQUEST on transmit.

The SCC is also capable of local loopback. In this mode TxD is RxD, just as in Auto Echo mode. However, in Local Loopback mode, the internal transmit data is tied to the internal receive data and RxD is ignored (except to be echoed out via TxD). The CTS and DCD inputs are also ignored as transmit and receive enables. However, transitions on these inputs can still cause interrupts. Local Loopback works in Asynchronous, Synchronous and SDLC modes with NRZ, NRZI or FM coding of the data stream.

**I/O Interface Capabilities.** The SCC offers the choice of Polling, Interrupt (vectored or nonvectored), and Block Transfer modes to transfer data, status, and control information to and from the CPU. The Block Transfer mode can be implemented under CPU or DMA control.

**Polling.** All interrupts are disabled. Three status registers in the SCC are automatically updated whenever any function is performed. For example, end-of-frame in SDLC mode sets a bit in one of these status registers. The idea behind polling is for the CPU to periodically read a status register until the register contents indicate the need for data to be transferred. Only one register needs to be

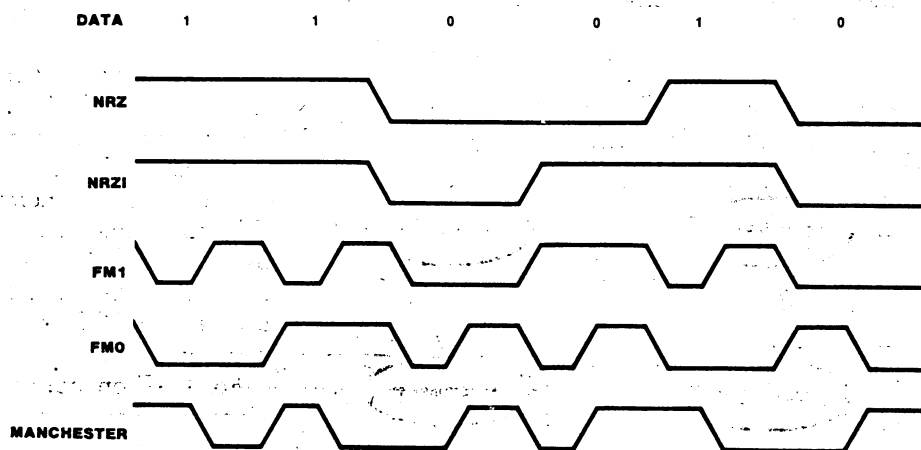


Figure 6. Data Encoding Methods

read; depending on its contents, the CPU either writes data, reads data, or continues. Two bits in the register indicate the need for data transfer. An alternative is a poll of the Interrupt Pending register to determine the source of an interrupt. The status for both channels resides in one register.

**Interrupts.** When an SCC responds to an Interrupt Acknowledge signal ( $\overline{INTACK}$ ) from the CPU, an interrupt vector may be placed on the data bus. This vector is written in WR2 and may be read in RR2A or RR2B (Figures 10 and 11).

To speed interrupt response time, the SCC can modify three bits in this vector to indicate status. If the vector is read in Channel A, status is never included; if it is read in Channel B, status is always included.

Each of the six sources of interrupts in the SCC (Transmit, Receive, and External/Status interrupts in both channels) has three bits associated with the interrupt source: Interrupt Pending (IP), Interrupt Under Service (IUS), and Interrupt Enable (IE). Operation of the IE bit is straightforward. If the IE bit is set for a given interrupt source, then that source can request interrupts. The exception is when the MIE (Master Interrupt Enable) bit in WR9 is reset and no interrupts may be requested. The IE bits are write only.

The other two bits are related to the interrupt priority chain (Figure 7). As a microprocessor peripheral, the SCC may request an interrupt only when no higher priority device is requesting one, e.g., when IEI is High. If the device in question requests an interrupt, it pulls down  $\overline{INT}$ . The CPU then responds with  $\overline{INTACK}$ , and the interrupting device places the vector on the data bus.

In the SCC, the IP bit signals a need for interrupt servicing. When an IP bit is 1 and the IEI input is High, the  $\overline{INT}$  output is pulled Low, requesting an interrupt. In the SCC, if the IE bit is not set by enabling interrupts, then the IP for that source can never be set. The IP bits are readable in RR3A.

The IUS bits signal that an interrupt request is being serviced. If an IUS is set, all interrupt sources of lower priority in the SCC and

external to the SCC are prevented from requesting interrupts. The internal interrupt sources are inhibited by the state of the internal daisy chain, while lower priority devices are inhibited by the IEO output of the SCC being pulled Low and propagated to subsequent peripherals. An IUS bit is set during an Interrupt Acknowledge cycle if there are no higher priority devices requesting interrupts.

There are three types of interrupts: Transmit, Receive, and External/Status. Each interrupt type is enabled under program control with Channel A having higher priority than Channel B, and with Receiver, Transmit, and External/Status interrupts prioritized in that order within each channel. When the Transmit interrupt is enabled, the CPU is interrupted when the transmit buffer becomes empty. (This implies that the transmitter must have had a data character written into it so that it can become empty.) When enabled, the receiver can interrupt the CPU in one of three ways:

- Interrupt on First Receive Character or Special Receive Condition.
- Interrupt on All Receive Characters or Special Receive Condition.
- Interrupt on Special Receive Condition Only.

Interrupt on First Character or Special Condition and Interrupt on Special Condition Only are typically used with the Block Transfer mode. A Special Receive Condition is one of the following: receiver overrun, framing error in Asynchronous mode, end-of-frame in SDLC mode and, optionally, a parity error. The Special Receive Condition interrupt is different from an ordinary receive character available interrupt only in the status placed in the vector during the Interrupt Acknowledge cycle. In Interrupt on First Receive Character, an interrupt can occur from Special Receive Conditions any time after the first receive character interrupt.

The main function of the External/Status interrupt is to monitor the signal transitions of the  $\overline{CTS}$ ,  $\overline{DCD}$ , and  $\overline{SYNC}$  pins; however, an

Z8530 SCC

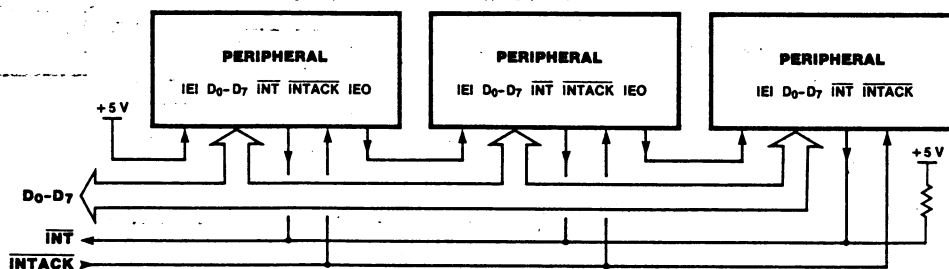


Figure 7. Interrupt Schedule

**Functional Description**  
(Continued)

External/Status interrupt is also caused by a Transmit Underrun condition, or a zero count in the baud rate generator, or by the detection of a Break (Asynchronous mode), Abort (SDLC mode) or EOP (SDLC Loop mode) sequence in the data stream. The interrupt caused by the Abort or EOP has a special feature allowing the SCC to interrupt when the Abort or EOP sequence is detected or terminated. This feature facilitates the proper termination of the current message, correct initialization of the next message, and the accurate timing of the Abort condition in external logic in SDLC mode. In SDLC Loop mode, this feature allows secondary stations to recognize the wishes of the primary station to regain control of the loop during a poll sequence.

**CPU/DMA Block Transfer.** The SCC provides a Block Transfer mode to accommodate CPU block transfer functions and DMA control. The Block Transfer mode uses the WAIT/REQUEST output in conjunction with the Wait/Request bits in WR1. The WAIT/REQUEST output can be defined under software control as a WAIT line in the CPU Block Transfer mode or as a REQUEST line in the DMA Block Transfer mode.

To a DMA controller, the SCC REQUEST output indicates that the SCC is ready to transfer data to or from memory. To the CPU the WAIT line indicates that the SCC is not ready to transfer data, thereby requesting the CPU extend the I/O cycle. The DTR/REQUEST line allows full-duplex operation under DMA control.

**Architecture**

The SCC internal structure includes two full-duplex channels, two baud rate generators, internal control and interrupt logic, and a bus interface to a nonmultiplexed bus. Associated with each channel are a number of read and write registers for mode control and status information, as well as logic necessary to interface to modems or other external devices (Figure 8).

The logic for both channels provides formats, synchronization, and validation for data transferred to and from the channel interface. The modem control inputs are monitored

by the control logic under program control. All of the modem control signals are general purpose in nature and can optionally be used for functions other than modem control.

The register set for each channel includes ten control (write) registers, two sync-character (write) registers, and four status (read) registers. In addition, each baud rate generator has two (read/write) registers for holding the time constant that determines the baud rate. Finally, associated with the interrupt logic is a write register for the interrupt vector accessible through either channel,

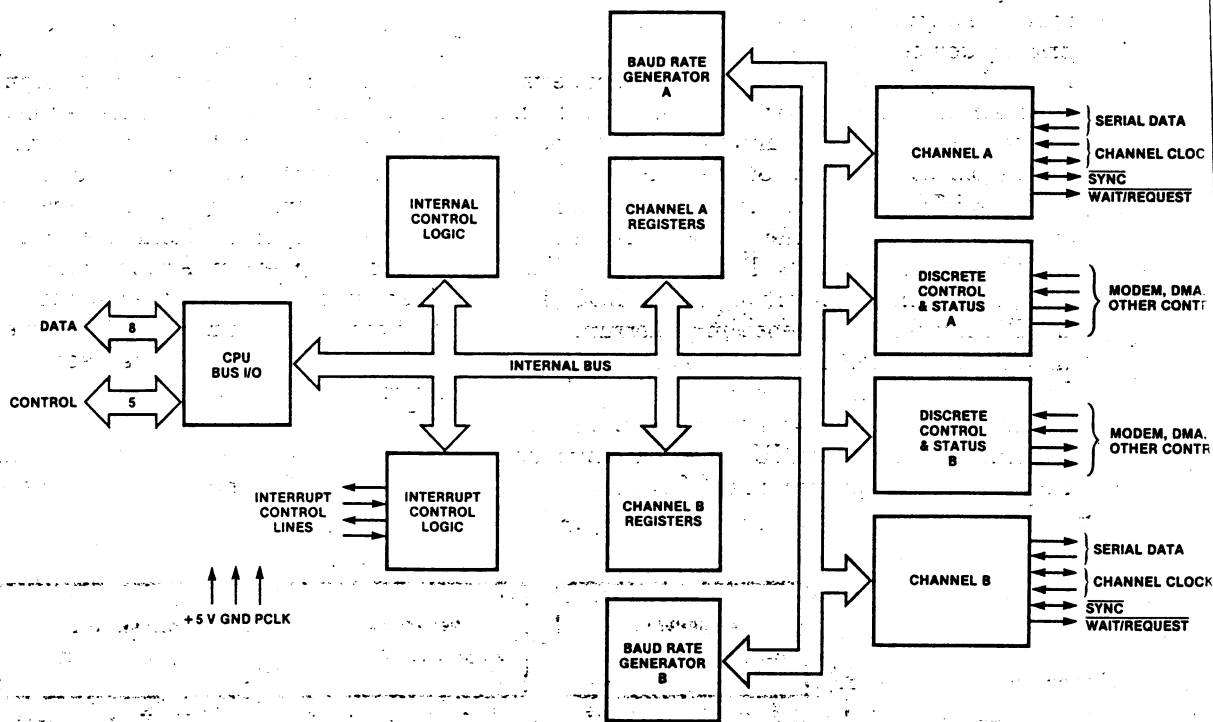


Figure 8. Block Diagram of SCC Architecture

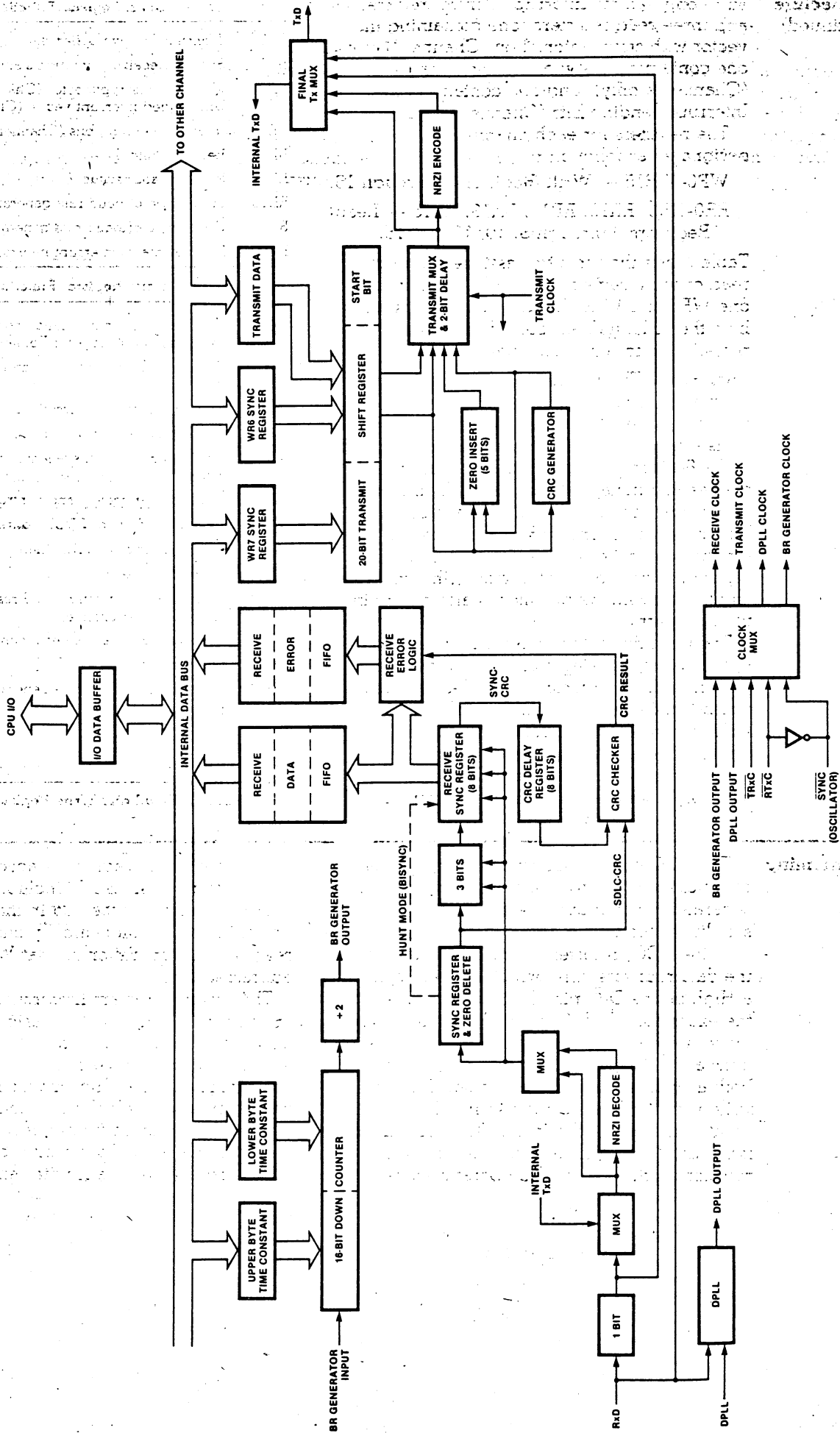


Figure 9. Data Path

28530 SCC

**Architecture**  
(Continued)

write only Master Interrupt Control register and three read registers: one containing the vector with status information (Channel B only), one containing the vector without status (Channel A only), and one containing the Interrupt Pending bits (Channel A only).

The registers for each channel are designated as follows:

WR0-WR15 — Write Registers 0 through 15.

RR0-RR3, RR10, RR12, RR13, RR15 — Read Registers 0 through 3, 10, 12, 13, 15.

Table 1 lists the functions assigned to each read or write register. The SCC contains only one WR2 and WR9, but they can be accessed by either channel. All other registers are paired (one for each channel).

**Data Path.** The transmit and receive data path illustrated in Figure 9 is identical for both channels. The receiver has three 8-bit buffer registers in an FIFO arrangement, in addition to the 8-bit receive shift register. This scheme creates additional time for the CPU to service an interrupt at the beginning of a block of high speed data. Incoming data is routed through one of several paths (data or CRC) depending on the selected mode (the character length in Asynchronous modes also determines the data path).

The transmitter has an 8-bit Transmit Data buffer register loaded from the internal data bus and a 20-bit Transmit Shift register that can be loaded either from the synchronous character registers or from the Transmit Data register. Depending on the operational mode, outgoing data is routed through one of four main paths before it is transmitted from the Transmit Data output (TxD)

**Read Register Functions**

RR0	Transmit/Receive buffer status and External status
RR1	Special Receive Condition status
RR2	Modified interrupt vector (Channel B only) Unmodified interrupt vector (Channel A only)
RR3	Interrupt Pending bits (Channel A only)
RR8	Receive buffer
RR10	Miscellaneous status
RR12	Lower byte of baud rate generator time constant
RR13	Upper byte of baud rate generator time constant
RR15	External/Status interrupt information

**Write Register Functions**

WR0	CRC initialize, initialization commands for the various modes, Register Pointers
WR1	Transmit/Receive interrupt and data transfer mode definition
WR2	Interrupt vector (accessed through either channel)
WR3	Receive parameters and control
WR4	Transmit/Receive miscellaneous parameters and modes
WR5	Transmit parameters and controls
WR6	Sync characters or SDLC address field
WR7	Sync character or SDLC flag
WR8	Transmit buffer
WR9	Master interrupt control and reset (accessed through either channel)
WR10	Miscellaneous transmitter/receiver control bits
WR11	Clock mode control
WR12	Lower byte of baud rate generator time constant
WR13	Upper byte of baud rate generator time constant
WR14	Miscellaneous control bits
WR15	External/Status interrupt control

**Table 1. Read and Write Register Functions**

**Programming**

The SCC contains 13 write registers in each channel that are programmed by the system separately to configure the functional personality of the channels.

In the SCC, register addressing is direct for the data registers only, which are selected by a High on the D/C pin. In all other cases (with the exception of WR0 and RR0), programming the write registers requires two write operations and reading the read registers requires both a write and a read operation. The first write is to WR0 and contains three bits that point to the selected register. The second write is the actual control word for the selected register, and if the second operation is read,

the selected read register is accessed. All of the registers in the SCC, including the data registers, may be accessed in this fashion. The pointer bits are automatically cleared after the read or write operation so that WR0 (or RR0) is addressed again.

The system program first issues a series of commands to initialize the basic mode of operation. This is followed by other commands to qualify conditions within the selected mode. For example, the Asynchronous mode, character length, clock rate, number of stop bits, even or odd parity might be set first. Then the interrupt mode would be set, and finally, receiver or transmitter enable.



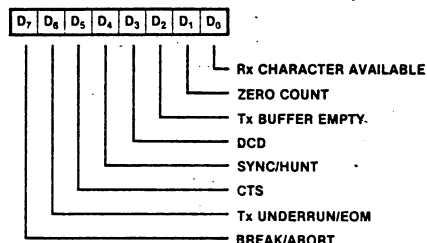
**Programming Read Registers.** (Continued)

The SCC contains eight read registers (actually nine, counting the receive buffer (RR8) in each channel). Four of these may be read to obtain status information (RR0, RR1, RR10, and RR15). Two registers (RR12 and RR13) may be read to learn the baud rate generator time constant. RR2 contains either the unmodified interrupt vector (Channel A) or the vector modified by status information

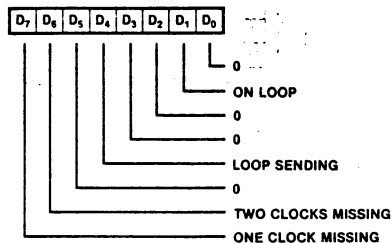
(Channel B). RR3 contains the Interrupt Pending (IP) bits (Channel A). Figure 10 shows the formats for each read register.

The status bits of RR0 and RR1 are carefully grouped to simplify status monitoring; e.g., when the interrupt vector indicates a Special Receive Condition interrupt, all the appropriate error bits can be read from a single register (RR1).

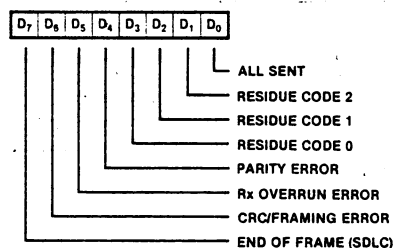
**Read Register 0**



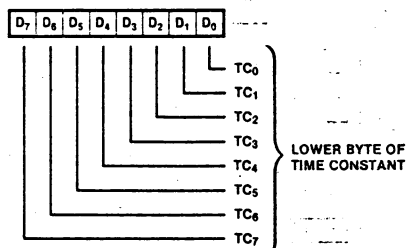
**Read Register 10**



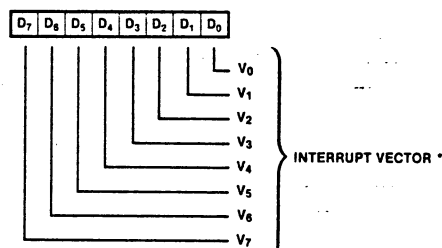
**Read Register 1**



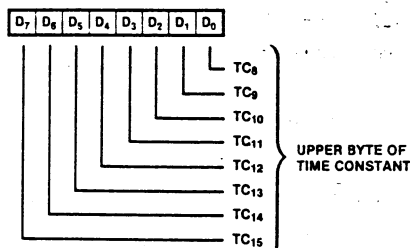
**Read Register 12**



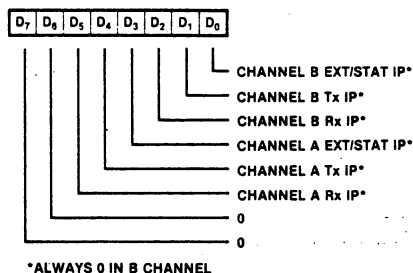
**Read Register 2**



**Read Register 13**



**Read Register 3**



**Read Register 15**

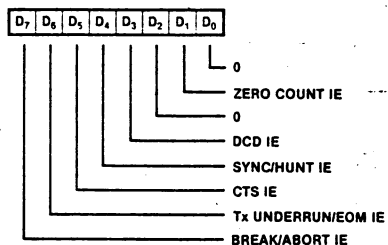


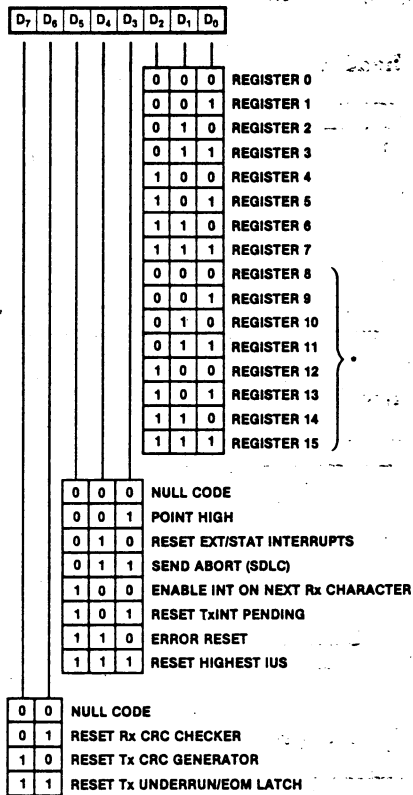
Figure 10. Read Register Bit Functions

Z8530 SCC

**Programming Write Registers.** The SCC contains 13 write registers (14 counting WR8, the transmit buffer) in each channel. These write registers are programmed separately to configure the functional "personality" of the channels. In addition, there are two registers (WR2 and

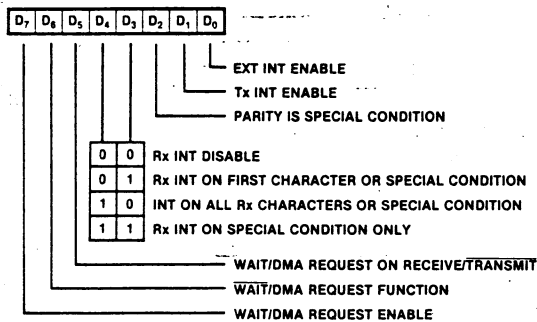
WR9) shared by the two channels that may be accessed through either of them. WR2 contains the interrupt vector for both channels, while WR9 contains the interrupt control bits. Figure 11 shows the format of each write register.

**Write Register 0**

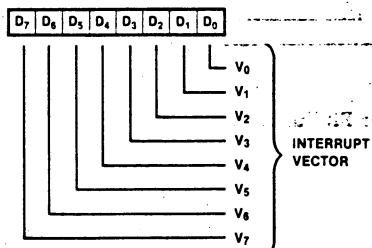


\*WITH POINT HIGH COMMAND

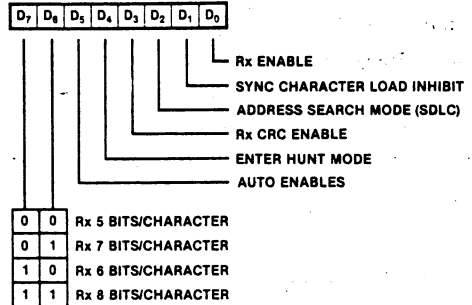
**Write Register 1**



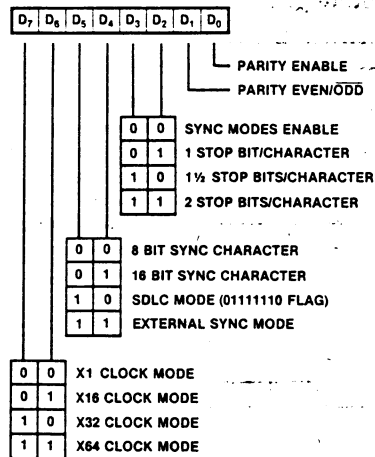
**Write Register 2**



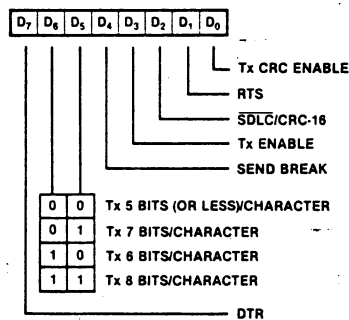
**Write Register 3**



**Write Register 4**



**Write Register 5**



**Write Register 6**

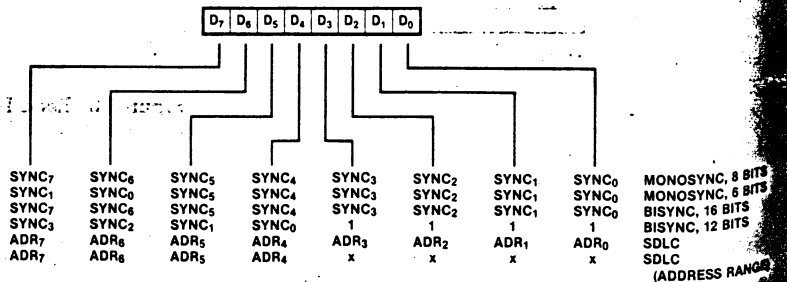
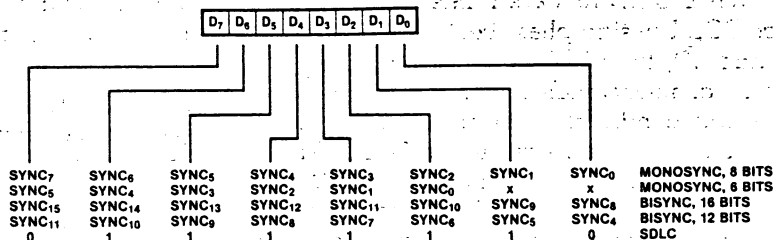
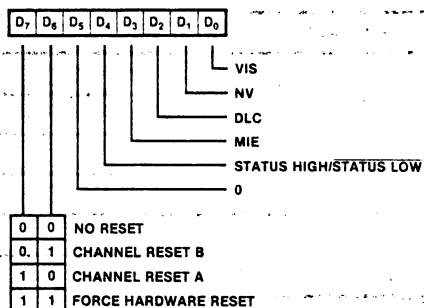


Figure 11. Write Register Bit Functions

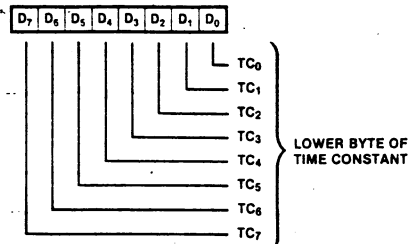
### Write Register 7



### Write Register 9

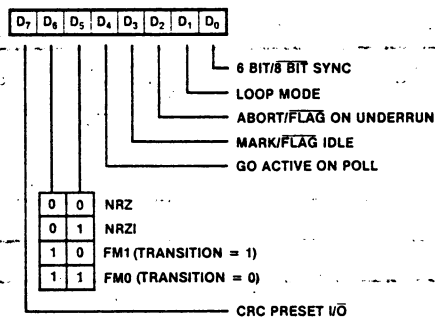


### Write Register 12

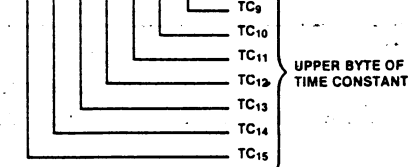


### Write Register 13

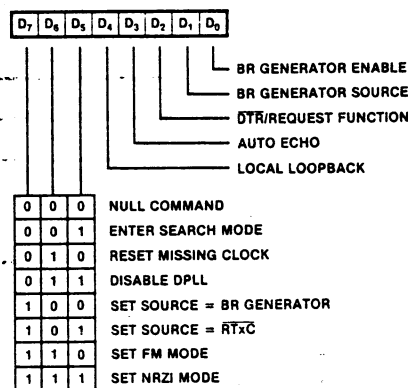
### Write Register 10



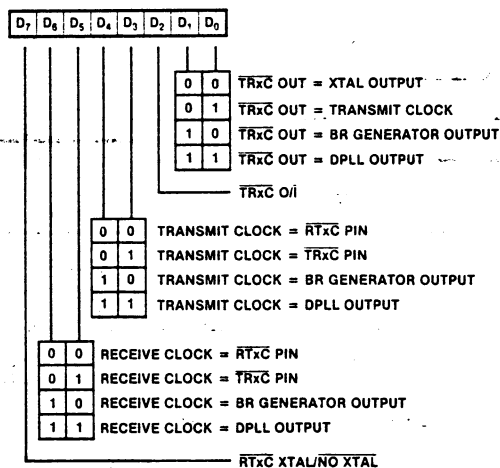
### Write Register 13



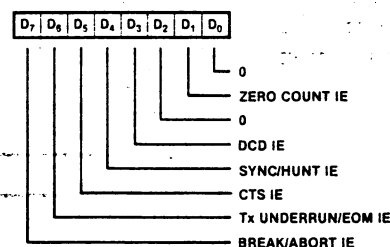
### Write Register 14



### Write Register 11



### Write Register 15



Z8530 SCC

Figure 11. Write Register Bit Functions (Continued)

## Timing

The SCC generates internal control signals from  $\overline{WR}$  and  $\overline{RD}$  that are related to PCLK. Since PCLK has no phase relationship with  $\overline{WR}$  and  $\overline{RD}$ , the circuitry generating these internal control signals must provide time for metastable conditions to disappear. This gives rise to a recovery time related to PCLK. The recovery time applies only between bus transactions involving the SCC. The recovery time required for proper operation is specified from the rising edge of  $\overline{WR}$  or  $\overline{RD}$  in the first trans-

action involving the SCC to the falling edge of  $\overline{WR}$  or  $\overline{RD}$  in the second transaction involving the SCC. This time must be at least 6 PCLK cycles plus 200 ns.

**Read Cycle Timing.** Figure 12 illustrates Read cycle timing. Addresses on  $A/\overline{B}$  and  $D/\overline{C}$  and the status on  $\overline{INTACK}$  must remain stable throughout the cycle. If  $\overline{CE}$  falls after  $\overline{RD}$  falls or if it rises before  $\overline{RD}$  rises, the effective  $\overline{RD}$  is shortened.

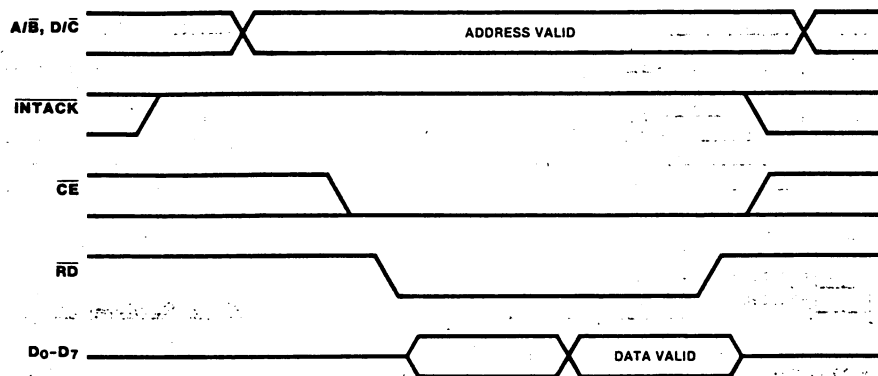


Figure 12. Read Cycle Timing

**Write Cycle Timing.** Figure 13 illustrates Write cycle timing. Addresses on  $A/\overline{B}$  and  $D/\overline{C}$  and the status on  $\overline{INTACK}$  must remain stable

throughout the cycle. If  $\overline{CE}$  falls after  $\overline{WR}$  falls or if it rises before  $\overline{WR}$  rises, the effective  $\overline{WR}$  is shortened.

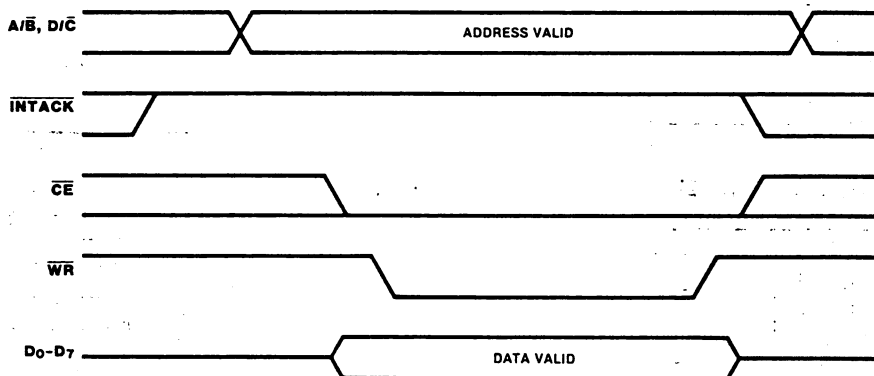


Figure 13. Write Cycle Timing

**Interrupt Acknowledge Cycle Timing.** Figure 14 illustrates Interrupt Acknowledge cycle timing. Between the time  $\overline{INTACK}$  goes Low and the falling edge of  $\overline{RD}$ , the internal and external IEL/IEO daisy chains settle. If there is an interrupt pending in the SCC and IEL is

High when  $\overline{RD}$  falls, the Acknowledge cycle is intended for the SCC. In this case, the SCC may be programmed to respond to  $\overline{RD}$  Low by placing its interrupt vector on  $D_0-D_7$  and it then sets the appropriate Interrupt-Under-Service latch internally.

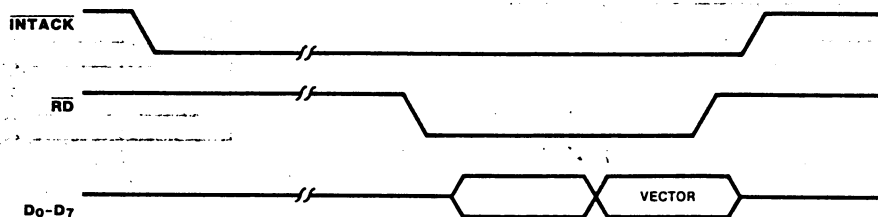


Figure 14. Interrupt Acknowledge Cycle Timing

**Absolute Maximum Ratings**

Voltages on all inputs and outputs with respect to GND ..... -0.3 V to +7.0 V  
 Operating Ambient Temperature ..... As Specified in Ordering Information  
 Storage Temperature ..... -65°C to +150°C

Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**Standard Test Conditions**

The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

- $+4.75\text{ V} \leq V_{CC} \leq +5.25\text{ V}$
  - $GND = 0\text{ V}$
  - $T_A$  as specified in Ordering Information
- All ac parameters assume a load capacitance of 50 pF max.

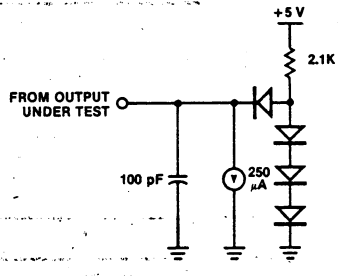


Figure 15. Standard Test Load

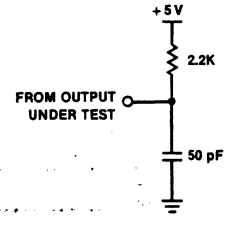


Figure 16. Open-Drain Test Load

Z8530 SCC

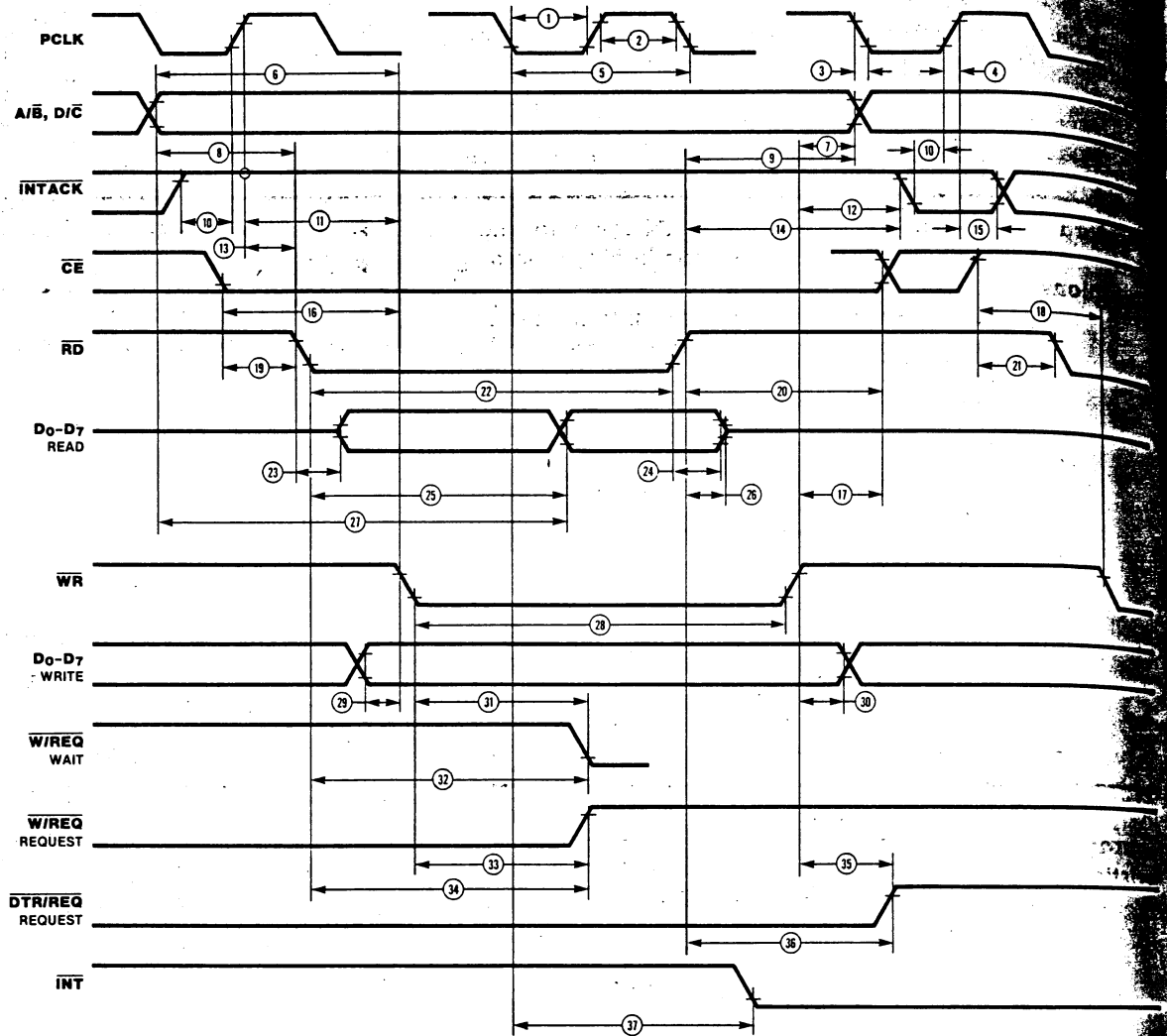
DC Characteristics	Symbol	Parameter	Min	Max	Unit	Condition
	$V_{IH}$	Input High Voltage	2.0	$V_{CC} + 0.3$	V	
	$V_{IL}$	Input Low Voltage	-0.3	0.8	V	
	$V_{OH}$	Output High Voltage	2.4		V	$I_{OH} = -250\ \mu\text{A}$
	$V_{OL}$	Output Low Voltage		0.4	V	$I_{OL} = +2.0\ \text{mA}$
	$I_{IL}$	Input Leakage		$\pm 10.0$	$\mu\text{A}$	$0.4 \leq V_{IN} \leq +2.4\text{V}$
	$I_{OL}$	Output Leakage		$\pm 10.0$	$\mu\text{A}$	$0.4 \leq V_{OUT} \leq +2.4\text{V}$
	$I_{CC}$	$V_{CC}$ Supply Current		250	mA	

$V_{CC} = 5\text{ V} \pm 5\%$  unless otherwise specified, over specified temperature range.

Capacitance	Symbol	Parameter	Min	Max	Unit	Test Condition
	$C_{IN}$	Input Capacitance		10	pF	Unmeasured Pins Returned to Ground
	$C_{OUT}$	Output Capacitance		15	pF	
	$C_{I/O}$	Bidirectional Capacitance		20	pF	

$f = 1\ \text{MHz}$ , over specified temperature range.

# Read and Write Timing



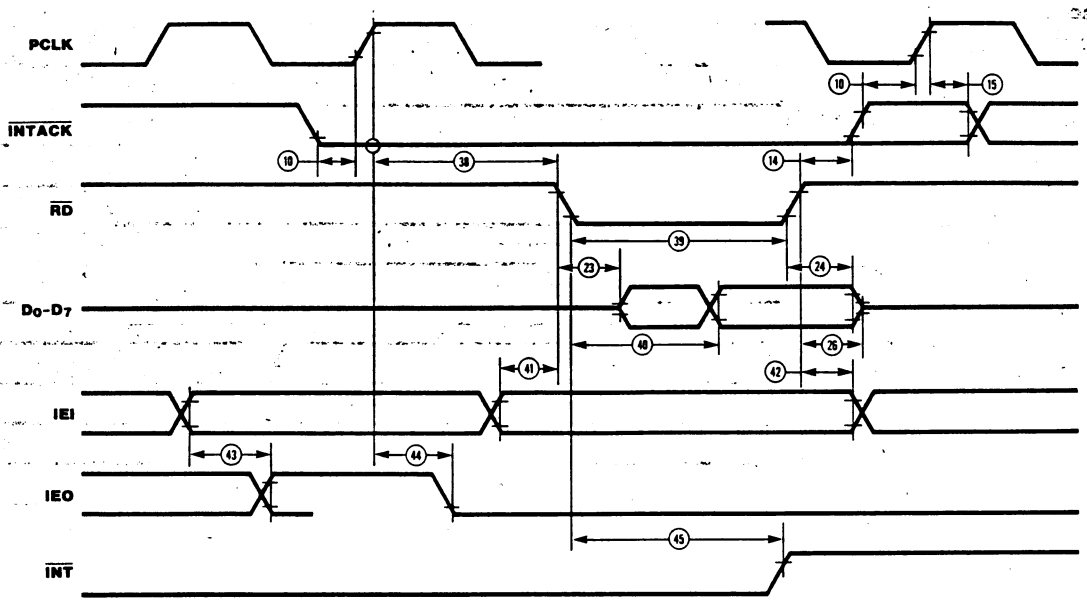
No.	Symbol	Parameter	4 MHz		6 MHz		Notes†
			Min	Max	Min	Max	
1	TwPCl	PCLK Low Width	105	2000	70	1000	
2	TwPCh	PCLK High Width	105	2000	70	1000	
3	TfPC	PCLK Fall Time		20		10	
4	TrPC	PCLK Rise Time		20		15	
5	TcPC	PCLK Cycle Time	250	4000	165	2000	
6	TsA(WR)	Address to WR ↓ Setup Time	80		80		
7	ThA(WR)	Address to WR ↑ Hold Time	0		0		
8	TsA(RD)	Address to RD ↓ Setup Time	80		80		
9	ThA(RD)	Address to RD ↑ Hold Time	0		0		
10	TsIA(PC)	INTACK to PCLK ↑ Setup Time	0		0		
11	TsIAi(WR)	INTACK to WR ↓ Setup Time	200		160		1
12	ThIA(WR)	INTACK to WR ↑ Hold Time	0		0		
13	TsIAi(RD)	INTACK to RD ↓ Setup Time	200		160		1
14	ThIA(RD)	INTACK to RD ↑ Hold Time	0		0		
15	ThIA(PC)	INTACK to PCLK ↑ Hold Time	100		100		
16	TsCEl(WR)	CE Low to WR ↓ Setup Time	0		0		
17	ThCE(WR)	CE to WR ↑ Hold Time	0		0		
18	TsCEh(WR)	CE High to WR ↓ Setup Time	100		70		
19	TsCEl(RD)	CE Low to RD ↓ Setup Time	0		0		1
20	ThCE(RD)	CE to RD ↑ Hold Time	0		0		1
21	TsCEh(RD)	CE High to RD ↓ Setup Time	100		70		1
22	TwRDl	RD Low Width	390		250		1
23	TdRD(DRA)	RD ↓ to Read Data Active Delay	0		0		
24	TdRD(DR)	RD ↑ to Read Data Not Valid Delay	0		0		
25	TdRD(DR)	RD ↓ to Read Data Valid Delay		250		180	
26	TdRD(DRz)	RD ↑ to Read Data Float Delay		70		45	2

**NOTES:**

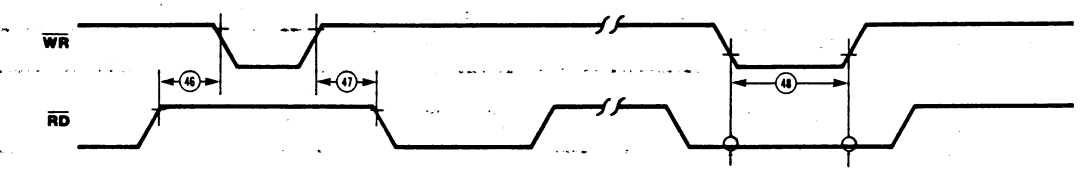
- Parameter does not apply to Interrupt Acknowledge transactions.
- Float delay is defined as the time required for a ±0.5 V change

in the output with a maximum dc load and minimum ac load.  
 \* Timings are preliminary and subject to change.  
 † Units in nanoseconds (ns).

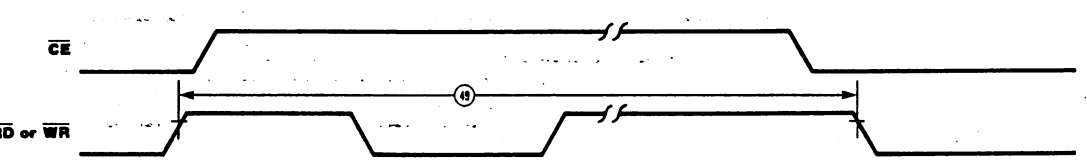
**Interrupt Acknowledge**



**Reset Timing**



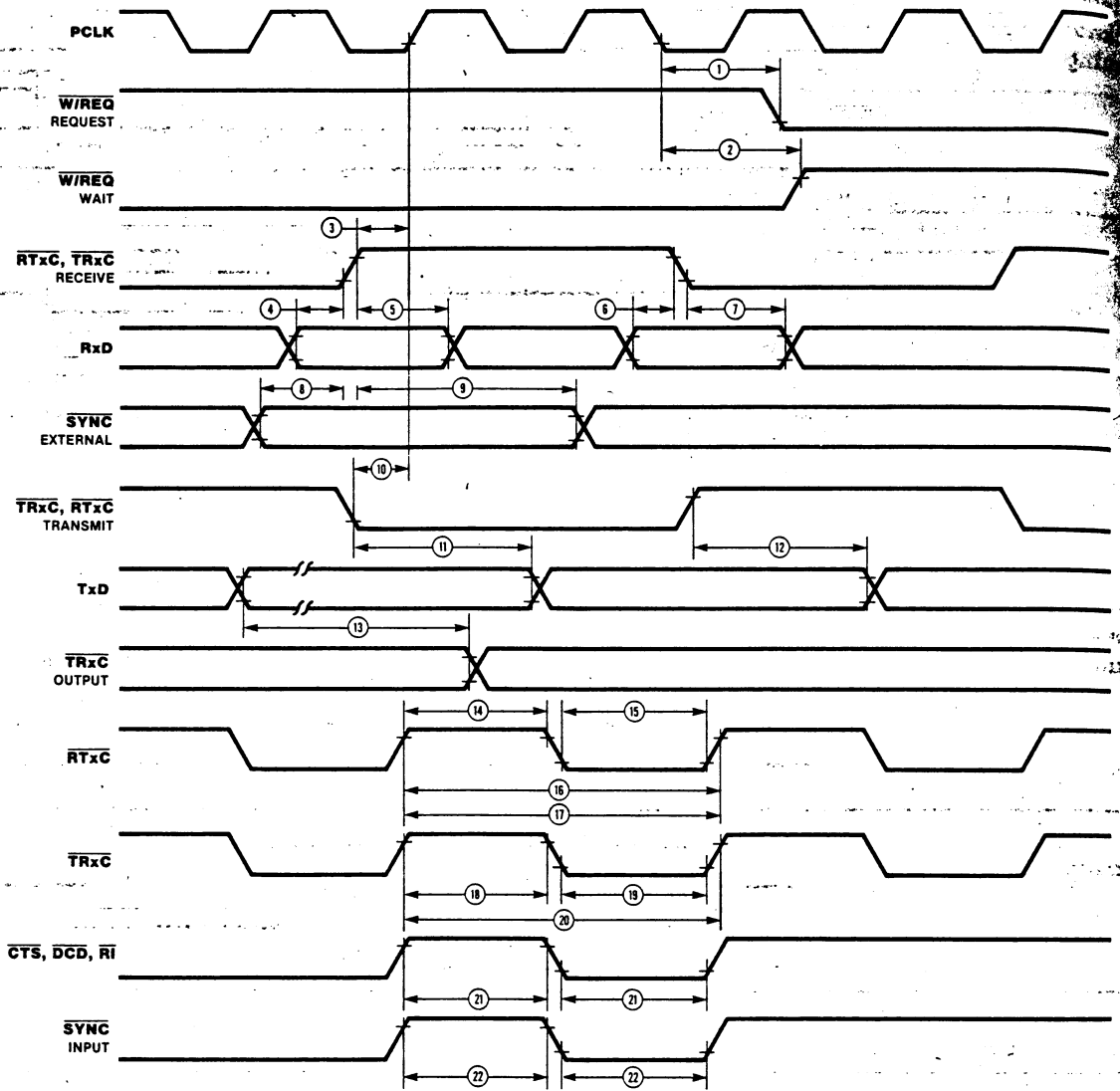
**Cycle Timing**



No.	Symbol	Parameter	4 MHz		6 MHz		Notes*†
			Min	Max	Min	Max	
27	TdA(DR)	Address Required Valid to Read Data Valid Delay		590		420	
28	TwWR1	WR Low Width	390		250		
29	TsDW(WR)	Write Data to WR ↓ Setup Time	0		0		
30	ThDW(WR)	Write Data to WR ↑ Hold Time	0		0		
31	TdWR(W)	WR ↓ to Wait Valid Delay		240		200	4
32	TdRD(W)	RD ↓ to Wait Valid Delay		240		200	4
33	TdWRf(REQ)	WR ↓ to W/REQ Not Valid Delay		240		200	
34	TdRDf(REQ)	RD ↓ to W/REQ Not Valid Delay		240		200	
35	TdWRr(REQ)	WR ↑ to DTR/REQ Not Valid Delay		5TcPC + 300		5TcPC + 250	
36	TdRDd(REQ)	RD ↑ to DTR/REQ Not Valid Delay		5TcPC + 300		5TcPC + 250	
37	TdPC(INT)	PCLK ↓ to INT Valid Delay					4
38	TdIAi(RD)	INTACK to RD ↓ (Acknowledge) Delay	250		250		5
39	TwrDA	RD (Acknowledge) Width	285		250		
40	TdRDA(DR)	RD ↓ (Acknowledge) to Read Data Valid Delay		190		180	
41	TsIEI(RDA)	IEI to RD ↓ (Acknowledge) Setup Time	120		100		
42	ThIEI(RDA)	IEI to RD ↑ (Acknowledge) Hold Time	0		0		
43	TdIEI(IEO)	IEI to IEO Delay Time		120		100	
44	TdPC(IEO)	PCLK ↑ to IEO Delay		250		250	
45	TdRDA(INT)	RD ↓ to INT Inactive Delay		500		500	4
46	TdRD(WRQ)	RD ↑ to WR ↓ Delay for No Reset	30		15		
47	TdWRQ(RD)	WR ↑ to RD ↓ Delay for No Reset	30		30		
48	TwRES	WR and RD Coincident Low for Reset	250		250		
49	Trc	Valid Access Recovery Time	6TcPC + 200		6TcPC + 130		3

NOTES:  
 3. Parameter applies only between transactions involving the SCC.  
 4. Open-drain output, measured with open-drain test load.  
 5. Parameter is system dependent. For any SCC in the daisy chain, TdIAi(RD) must be greater than the sum of TdPC(IEO) for the SCC, and TdIEI(IEO) for each device separating them in the daisy chain.  
 \* Timings are preliminary and subject to change.  
 † Units in nanoseconds (ns).

**General  
Timing**





No.	Symbol	Parameter	4 MHz		6 MHz		Notes*†
			Min	Max	Min	Max	
	TdPC(REQ)	PCLK ↓ to $\overline{W/REQ}$ Valid Delay		250		250	
	TdPC(W)	PCLK ↓ to Wait Inactive Delay		350		350	
3	TsRXC(PC)	$\overline{RxC}$ ↑ to PCLK ↑ Setup Time (PCLK + 4 case only)	80	TwPC1	70	TwPC1	1,4
4	TsRXD(RXCr)	RxD to $\overline{RxC}$ ↑ Setup Time (X1 Mode)	0		0		1
5	ThRXD(RXCr)	RxD to $\overline{RxC}$ ↑ Hold Time (X1 Mode)	150		150		1
6	TsRXD(RXCf)	RxD to $\overline{RxC}$ ↓ Setup Time (X1 Mode)	0		0		1,5
7	ThRXD(RXCf)	RxD to $\overline{RxC}$ ↓ Hold Time (X1 Mode)	150		150		1,5
8	TsSY(RXC)	$\overline{SYNC}$ to $\overline{RxC}$ ↑ Setup Time	-200		-200		1
9	ThSY(RXC)	$\overline{SYNC}$ to $\overline{RxC}$ ↑ Hold Time	3TcPC +200		3TcPC +200		1
10	TsTXC(PC)	$\overline{TxC}$ ↓ to PCLK ↓ Setup Time	0		0		2,4
11	TdTXCf(TXD)	$\overline{TxC}$ ↓ to Tx D Delay (X1 Mode)		300		230	2
12	TdTXCr(TXD)	$\overline{TxC}$ ↑ to Tx D Delay (X1 Mode)		300		230	2,5
13	TdTXD(TRX)	TxD to $\overline{TRxC}$ Delay (Send Clock Echo)		200		200	
14	TwRTXh	$\overline{RTxC}$ High Width	180		180		6
15	TwRTXl	$\overline{RTxC}$ Low Width	180		180		6
16	TcRTX	$\overline{RTxC}$ Cycle Time	400		400		6
	TcRTXX	Crystal Oscillator Period	250	1000	250	1000	3
18	TwTRXh	$\overline{TRxC}$ High Width	180		180		6
19	TwTRXl	$\overline{TRxC}$ Low Width	180		180		6
20	TcTRX	$\overline{TRxC}$ Cycle Time	400		400		6
21	TwEXT	$\overline{DCD}$ or $\overline{CTS}$ Pulse Width	200		200		
22	TwSY	$\overline{SYNC}$ Pulse Width	200		200		

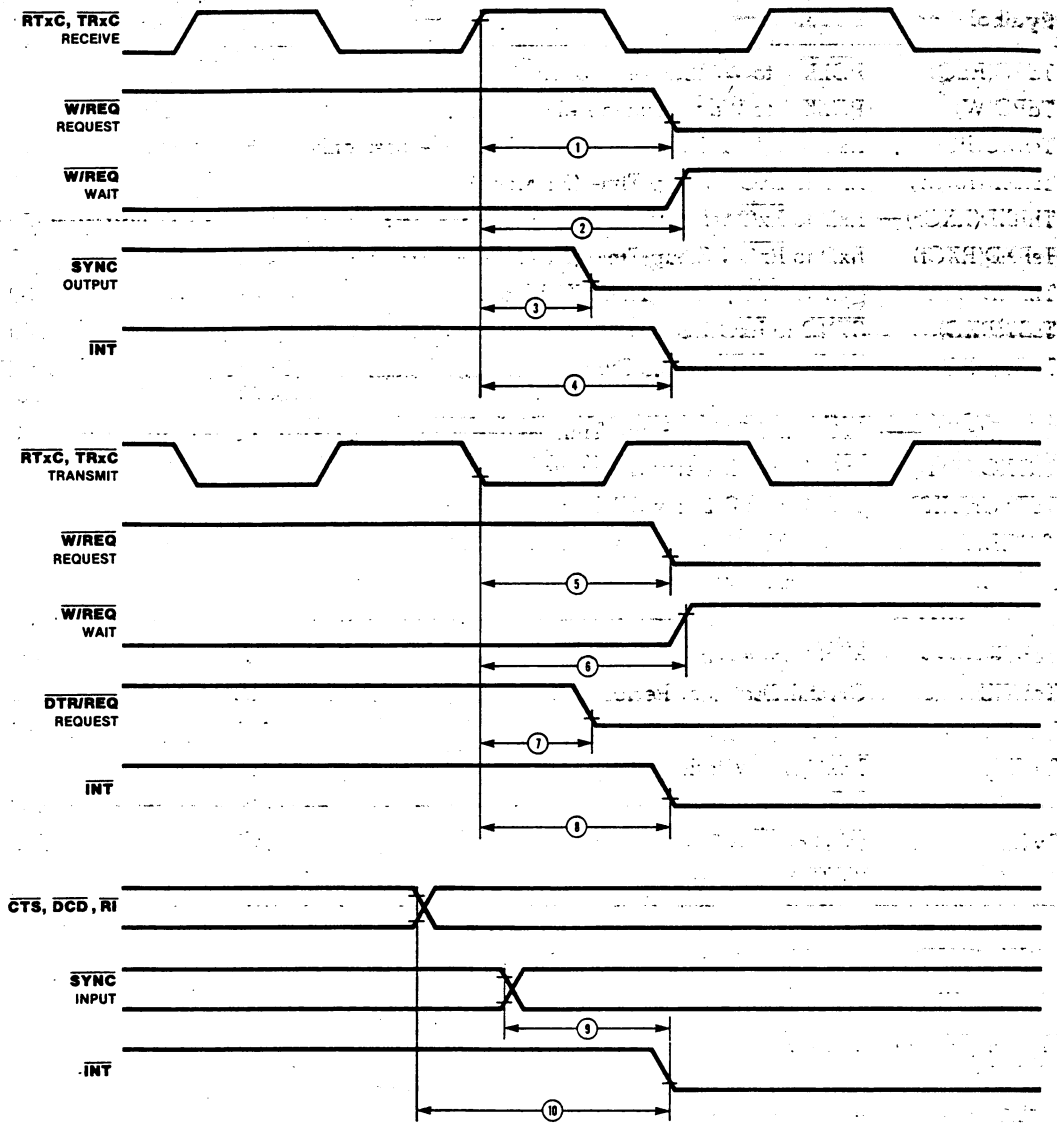
NOTES:

1.  $\overline{RxC}$  is  $\overline{RTxC}$  or  $\overline{TRxC}$ , whichever is supplying the receive clock.
2.  $\overline{TxC}$  is  $\overline{TRxC}$  or  $\overline{RTxC}$ , whichever is supplying the transmit clock.
3. Both  $\overline{RTxC}$  and  $\overline{SYNC}$  have 30 pF capacitors to ground connected to them.
4. Parameter applies only if the data rate is one-fourth the PCLK rate. In all other cases, no phase relationship between  $\overline{RxC}$  and PCLK or  $\overline{TxC}$  and PCLK is required.

5. Parameter applies only to FM encoding/decoding.
  6. Parameter applies only for transmitter and receiver; DPLL and baud rate generator timing requirements are identical to chip PCLK requirements.
- \* Timings are preliminary and subject to change.  
† Units in nanoseconds (ns).

Z8530 SCC

# System Timing



No.	Symbol	Parameter	4 MHz		6 MHz		Notes†
			Min	Max	Min	Max	
1	TdRXC(REQ)	$\overline{Rx}C \uparrow$ to $\overline{W}/REQ$ Valid Delay	8	12	8	12	2
2	TdRXC(W)	$\overline{Rx}C \uparrow$ to Wait Inactive Delay	8	12	8	12	1,2
3	TdRXC(SY)	$\overline{Rx}C \uparrow$ to $\overline{SYNC}$ Valid Delay	4	7	4	7	2
4	TdRXC(INT)	$\overline{Rx}C \uparrow$ to $\overline{INT}$ Valid Delay	10	16	10	16	1,2
5	TdTXC(REQ)	$\overline{Tx}C \downarrow$ to $\overline{W}/REQ$ Valid Delay	5	8	5	8	3
6	TdTXC(W)	$\overline{Tx}C \downarrow$ to Wait Inactive Delay	5	8	5	8	1,3
7	TdTXC(DRQ)	$\overline{Tx}C \downarrow$ to $\overline{DTR}/REQ$ Valid Delay	4	7	4	7	3
8	TdTXC(INT)	$\overline{Tx}C \downarrow$ to $\overline{INT}$ Valid Delay	6	10	6	10	1,3
9	TdSY(INT)	$\overline{SYNC}$ Transition to $\overline{INT}$ Valid Delay	2	6	2	6	1
10	TdEXT(INT)	$\overline{DCD}$ or $\overline{CTS}$ Transition to $\overline{INT}$ Valid Delay	2	6	2	6	1

**NOTES:**

1. Open-drain output, measured with open-drain test load.
2.  $RxC$  is  $RTxC$  or  $TRxC$ , whichever is supplying the receive clock.
3.  $TxC$  is  $TRxC$  or  $RTxC$ , whichever is supplying the transmit clock.

\* Timings are preliminary and subject to change.  
 † Units equal to  $TcPC$ .

**Ordering Information**

Product Number	Package/ Temp	Speed	Description	Product Number	Package/ Temp	Speed	Description
Z8530	CE	4.0 MHz	SCC (40-pin)	Z8530A	CE	6.0 MHz	SCC (40-pin)
Z8530	CM	4.0 MHz	Same as above	Z8530A	CM	6.0 MHz	Same as above
Z8530	CMB	4.0 MHz	Same as above	Z8530A	CMB	6.0 MHz	Same as above
Z8530	CS	4.0 MHz	Same as above	Z8530A	CS	6.0 MHz	Same as above
Z8530	DE	4.0 MHz	Same as above	Z8530A	DE	6.0 MHz	Same as above
Z8530	DS	4.0 MHz	Same as above	Z8530A	DS	6.0 MHz	Same as above
Z8530	PE	4.0 MHz	Same as above	Z8530A	PE	6.0 MHz	Same as above
Z8530	PS	4.0 MHz	Same as above	Z8530A	PS	6.0 MHz	Same as above

NOTES: C = Ceramic, D = Cerdip, P = Plastic; E = -40°C to +85°C, M = -55°C to 125°C, MB = -55°C to 125°C with MIL-STD-883 with Class B processing, S = 0°C to +70°C.

**Z8530 SCC**