
Nur zum internen Gebrauch bestimmt

Ministerium
für Elektrotechnik
und Elektronik

Herausgeber: Ingenieurschule für Ma-
schinenbau und Elektro-
technik Berlin

Blatt-Nr.: 1 ...54

Arbeitsblätter
für die Ingenieurausbildung

Befehlssatz und Programmiertabellen für die
Bausteine des Mikroprozessor-Systems U 880

Veröffentlicht:

Ingenieurschule für Maschinenbau und Elektrotechnik
Berlin

Die Seiten 4 ... 23 sind ein überarbeiteter Nachdruck des bisher in der Reihe Mikrorechner als Heft 6 (Bestill-Nr. T.7.27.0009) von der Zentralstelle für Aus- und Weiterbildung im Industriebereich Elektrotechnik/Elektronik herausgegebenen Lehrmaterials.

Diese Seiten wurden von
Hr.-Ing. M a t s c h k e
im Auftrage der Ingenieurschule für Maschinenbau und Elektrotechnik Berlin leicht überarbeitet.

Der Nachdruck erfolgt mit Zustimmung der Zentralstelle für Aus- und Weiterbildung im Institut für Rationalisierung der Elektrotechnik/Elektronik.

Die Seiten 24 ... 54 sind vom Studenten Gerth zusammengestellt worden.

Als Manuskript gedruckt Alle Rechte vorbehalten
1. Auflage

Redaktionsachluß: 8. 10. 1985
Bestell.-Nr. : 4-6-12
Druckgen.-Nr. : BG 106/167/86
Druck : Druckerei Braul

Inhaltsverzeichnis

	Seite
Erläuterungen zur Befehlsliste des U 880	4
Befehlsliste	7
Ladebefehle (8 Bit)	7
Ladebefehle (16 Bit)	8
Austauschbefehle	10
Blockbefehle	10
Arithmetik- und Logikbefehle (8 Bit)	11
Arithmetikbefehle (16 Bit)	13
Allgemeine Arithmetik- und Steuerbefehle	15
Rotations- und Verschiebepbefehle	15
Bitmanipulationsbefehle	18
Sprungbefehle	19
Ein- und Ausgabebefehle	21
Assembler-Anweisungen für SYPS K 1520	22
ISO-7-Bit-Code nach TGL 23207	23
Codierungstabelle für den Prozessor U 880 (vollständig)	24
Programmierung PIO, CTC (Übersicht), (Codierungstabellen)	26
Programmierung SIO (Übersicht)	27
Abarbeitungsfolge bei der Programmierung der einzelnen Betriebsarten	27
Codierungstabellen für die SIO U 856	29
Programmierung DMA (Übersicht)	32
Codierungstabellen für die DMA U 858	33
Decodierungstabelle für den hexadezimalen Code des Prozessors U 880	36
- 1. Byte	36
- Fortsetzung 2. Byte, 1. Byte = CB	37
- Fortsetzung 2. Byte, 1. Byte = DD/FD	38
- Fortsetzung 2. Byte, 1. Byte = ED	39
- Fortsetzung 3. (4.) Byte, 1. Byte = DD/FD, 2. Byte CB	40
Decodierungstabelle der PIO U 855	41
Decodierungstabelle der CTC U 857	42
Decodierungstabellen der SIO U 856	43
Decodierungstabellen der DMA U 858	50

Erläuterungen zur Befehlsliste des U 880

Die auf S. 8 beginnende Befehlsliste enthält alle 260 Befehle des K 1520, aus denen sich durch Einsetzen der Register, der Bitpositionen oder der Restart-Adressen insgesamt 699 Varianten ergeben. Die Befehle sind nach Gruppen geordnet. Der Maschinencode der Befehle ist unterschiedlich lang; er besteht aus 1, 2, 3 oder 4 Byte.

In der ersten Spalte der Befehlsliste, mit Mnem (Mnemonic; Merkwort) bezeichnet, ist das Kennwort des Befehls angeführt. Die maximal 4 Buchstaben, nur bei 3 Befehlen kommt eine Ziffer vor, sind Abkürzungen aus der amerikanischen Schreibweise, z. B. JMP von jump (Sprung). Insgesamt gibt es 99 verschiedene Mnemoniks.

Die zweite Spalte enthält den oder die Operanden in symbolischer Darstellung, die weiter unten erläutert wird. Ist dieses Feld leer, so bedeutet es, daß dieser Befehl keinen Operanden besitzt, z. B. RET.

In der 3. Spalte ist unter T (Taktzyklen) angegeben, wieviele Takte für die Ausführung benötigt werden. Die Zeit für einen Takt (400 ns) ist mit der angegebenen Zahl zu multiplizieren, um die Ausführungszeit für einen Befehl zu erhalten. Enthält die Spalte 2 Zahlen (bei bedingten Befehlen), so gilt die 1. Zahl für "erfüllte Bedingung" und die 2. für "nicht erfüllte Bedingung".

In der Spalte "Maschinencode" ist das dem Befehl entsprechende Bitmuster in zweierlei Form beschrieben: als 8stellige Dualzahl und als 2stellige Hexadezimal- (Sedezimal-)Zahl. Jedes Byte steht in einer Zeile, die Anzahl der Byte für den Befehl ist sofort erkennbar. Leere Zeilen (z. B. für Werte oder Adressen) und unvollständige Zeilen (z. B. für einzuführende Register) sind durch besondere Symbole gekennzeichnet, die anschließend erläutert werden; die Spalte Hex ist dort freigelassen. Die tetradische Schreibweise für die 8stelligen

Dualzahl wurde gewählt, um die Umcodierung in eine Sedezimalzahl zu erleichtern.

In der Spalte "Flags" ist mit Symbolen beschrieben, welchen Zustand die Flags nach ausgeführtem Befehl anzeigen.

Die letzte Spalte erläutert in abgekürzter Form die Wirkung des Befehls. Weitergehende Erklärungen in Pkt. 3.4.

Verwendete Symbole, Codes und Abkürzungen

r und r'	eines der Universalregister A,B,C,D,E,H,L		
zzz	Code für das Zielregister		
qqq	Code für das Quellregister		
	zu ersetzen durch	111	für A
		000	für B
		001	für C
		010	für D
		011	für E
		100	für H
		101	für L
n	in einem Byte codierbarer Wert (0...255 bzw. 0H...FFH)		
xxxx xxxx	Code für den 8-Bit-Wert		
nn	in zwei Byte codierbarer Wert (0...65535 bzw. 0H...FFFFH)		
	Code für den 16-Bit-Wert allgemein		
nxn nxn	niederwertiger Teil		
xhxh xhxh	höherwertiger Teil		
	Code für den 16-Bit-Wert als Adresse		
anan anan	niederwertiger Teil		
ahah ahah	höherwertiger Teil		
b	Bitposition		
bbb	Code für Bitposition		
	Wert der Bitposition		
	<u>7,6,5,4,3,2,1,0</u>		
	codiert als 3stellige Dualzahl		

d Displacement (Versetzung, Verlegung)
(-128...+127)

dddd dddd Code für Displacement
negative Werte im Zweier-Komplement

e Distanz (relative Sprungentfernung)

eeee eeee Code für Distanz
zählt ab aktuellem Stand des Befehlszählers, der
bei Ausführung dieses Befehls bereits die Adresse
des nächsten Befehls enthält, also um 2 weiterge-
stellt ist; relative Sprungentfernung bei "Rück-
wärtssprüngen" im Zweier-Komplement

p eine der 8 festgelegten RESTART-Adressen

ppp Code für RESTART-Adressen

000	=	0H	=	0
001	=	8H	=	8
010	=	10H	=	16
011	=	18H	=	24
100	=	20H	=	32
101	=	28H	=	40
110	=	30H	=	48
111	=	38H	=	56

() Inhalt der Klammer wird als Adresse eines
Speicherplatzes behandelt

Beisp.: (HL) Inhalt eines Speicherplatzes, dessen
Adresse im Registerpaar HL steht
~~steht (HL) darf auch M geschrieben werden~~

Beisp.: LD A, (HL) ist gleichbedeutend mit
LD A, M

Symbole für Flagzustände

. unverändert, vom Befehl nicht beeinflusst

x verändert, aber undefiniert

! aktuell gesetzt

0 auf 0 gesetzt

1 auf 1 gesetzt

V P/V-Flag zeigt Überlauf (Overflow) an

P P/V-Flag zeigt Parität an

IF P/V-Flag zeigt Inhalt des Interrupt-FF 2

Abkürzungen

SP Stackpointer (Kellerzeiger) 16 Bit

IX Indexregister X 16 Bit

IY Indedregister Y 16 Bit

PC Programm Counter (Befehlszähler) 16 Bit

I Interruptregister 8 Bit

R Refresh-Register zur zyklischen Auffrischung von RAM

F Flag-Register

Mnem	Operand	T	Maschinencode Dual Hex	Flags CY Z S PV	Bedeutung/Hinweise
Ladebefehle (8 Bit)					
LD	r, r'	4	01zz zqqq		r := r'
LD	r, n	7	00zz z110 xxxx xxxx		r := n
LD	r, (HL)	7	01zz z110		r := (HL)
LD	r, (IX+d)	19	1101 1101 01zz z110 dddd dddd	DD	r := (IX+d)
LD	r, (IY+d)	19	1111 1101 01zz z110 dddd dddd	FD	r := (IY+d)
LD	(HL), r	7	0111 0qqq	7.	(HL) := r
LD	(IX+d), r	19	1101 1101 0111 0qqq dddd dddd	DD 7.	(IX+d) := r
LD	(IY+d), r	19	1111 1101 0111 0qqq dddd dddd	FD 7.	(IY+d) := r
LD	(HL), n	10	0011 0110 xxxx xxxx	36	(HL) := n
LD	(IX+d), n	19	1101 1101 0011 0110 dddd dddd xxxx xxxx	DD 36	(IX+d) := n
LD	(IY+d), n	19	1111 1101 0011 0110 dddd dddd xxxx xxxx	FD 36	(IY+d) := n
LD	A, (BC)	7	0000 1010	0A	A := (BC)
LD	A, (DE)	7	0001 1010	1A	A := (DE)
LD	A, (nn)	13	0011 1010 anan anan ahah ahah	3A	A := (nn)
LD	(BC), A	7	0000 0010	02	(BC) := A
LD	(DE), A	7	0001 0010	12	(DE) := A
LD	(nn), A	13	0011 0010 anan anan ahah ahah	32	(nn) := A
LD	A, I	9	1110 1101 0101 0111	ED 57	A := I
LD	A, R	9	1110 1101 0101 1111	ED 5F	A := R
LD	I, A	9	1110 1101 0100 0111	ED 47	I := A
LD	R, A	9	1110 1101 0100 1111	ED 4F	R := A

Mnem	Operand	T	Maschinencode Dual	Hex	Flage CY Z S P V	Bedeutung/Hinweise
Ladebefehle (16 Bit)						
LD	BC,nn	10	0000 0001 xxxx xxxx xhxx xhxx	01		BC := nn (B := xhxx.. C := xxxx..)
LD	DE,nn	10	0001 0001 xxxx xxxx xhxx xhxx	11		DE := nn (D := xhxx.. E := xxxx..)
LD	HL,nn	10	0010 0001 xxxx xxxx xhxx xhxx	21		HL := nn (H := xhxx.. L := xxxx..)
LD	SP,nn	10	0011 0001 xxxx xxxx xhxx xhxx	31		SP := nn
LD	IX,nn	14	1101 1101 0010 0001 xxxx xxxx xhxx xhxx	DD 21		IX := nn
LD	IY,nn	14	1111 1101 0010 0001 xxxx xxxx xhxx xhxx	FD 21		IY := nn
LD	HL,(nn)	16	0010 1010 anan anan ahah ahah	2A		H := (nn+1) L := (nn)
LD	BC,(nn)	20	1110 1101 0100 1011 anan anan ahah ahah	ED 4B		B := (nn+1) C := (nn)
LD	DE,(nn)	20	1110 1101 0101 1011 anan anan ahah ahah	ED 5B		D := (nn+1) E := (nn)
LD	HL,(nn)	20	1110 1101 0110 1011 anan anan ahah ahah	ED 6B		H := (nn+1) L := (nn) redundanter Befehl s. 3-Byte-Version oben
LD	SP,(nn)	20	1110 1101 0111 1011 anan anan ahah ahah	ED 7B		SP := (nn+1) und (nn)
LD	IX,(nn)	20	1101 1101 0010 1010 anan anan ahah ahah	DD 2A		IX := (nn+1) und (nn)
LD	IY,(nn)	20	1111 1101 0010 1010 anan anan ahah ahah	FD 2A		IY := (nn+1) und (nn)

Mnem	Operand	T	Maschinencode		Flags CY Z S PV	Bedeutung/Hinweise
			Dual	Hex		
LD	(nn),HL	16	0010 0010 anan anan ahah ahah	22		(nn+1) := H (nn) := L
LD	(nn),BC	20	1110 1101 0100 0011 anan anan ahah ahah	ED 43		(nn+1) := B (nn) := C
LD	(nn),DE	20	1110 1101 0101 0011 anan anan ahah ahah	ED 53		(nn+1) := D (nn) := E
LD	(nn),HL	20	1110 1101 0110 0011 anan anan ahah ahah	ED 63		(nn+1) := H (nn) := L redundanter Befehl s. 3-Byte-Version oben
LD	(nn),SP	20	1110 1101 0111 0011 anan anan ahah ahah	ED 73		(nn+1) und (nn) := SP
LD	(nn),IX	20	1101 1101 0010 0010 anan anan ahah ahah	DD 22		(nn+1) und (nn) := IX
LD	(nn),IY	20	1111 1101 0010 0010 anan anan ahah ahah	FD 22		(nn+1) und (nn) := IY
LD	SP,HL	6	1111 1001	F9		SP := HL
LD	SP,IX	10	1101 1101 1111 1001	DD F9		SP := IX
LD	SP,IY	10	1111 1101 1111 1001	FD F9		SP := IY
PUSH	BC	11	1100 0101	C5		einkellern BC
PUSH	DE	11	1101 0101	D5		einkellern DE
PUSH	HL	11	1110 0101	E5		einkellern HL
PUSH	AF	11	1111 0101	F5		einkellern AF
PUSH	IX	15	1101 1101 1110 0101	DD E5		einkellern IX
PUSH	IY	15	1111 1101 1110 0101	FD E5		einkellern IY
POP	BC	10	1100 0001	C1		auskellern BC
POP	DE	10	1101 0001	D1		auskellern DE
POP	HL	10	1110 0001	E1		auskellern HL
POP	AF	10	1111 0001	F1		auskellern AF

Mnem	Operand	T	Maschinencode		Flags CY Z S PV	Bedeutung/Hinweise
			Dual	Hex		
POP	IX	14	1101 1101 1110 0001	DD E1		auskellern IX
POP	IY	14	1111 1101 1110 0001	FD E1		auskellern IY
Austauschbefehle						
EX	DE, HL	4	1110 1011	EB		Austausch DE mit HL
EX	AF	4	0000 1000	08		Austausch AF mit Zweitr.
EXX		4	1101 1001	D9		Austausch BCDEHL mit Zweitregistersatz
EX	(SP), HL	19	1110 0011	E3		Austausch(SP) mit HL
EX	(SP), IX	23	1101 1101 1110 0011	DD E3		Austausch(SP) mit IX
EX	(SP), IY	23	1111 1101 1110 0011	FD E3		Austausch(SP) mit IY
Blockbefehle						
LDI		16	1110 1101 1010 0000	ED A0	. I X I	(DE):=(HL), INC von DE und HL, DEC von BC +)
LDIR		21 /16	1110 1101 1011 0000	ED B0	. I X 0	(DE):=(HL), INC von DE und HL, DEC von BC bis BC = 0
LDD		16	1110 1101 1010 1000	ED A8	. I X I	(DE):=(HL), DEC von DE, HL und BC +)
DDR		21 /16	1110 1101 1011 1000	ED B8	. I X 0	(DE):=(HL); DEC von DE, HL und BC bis BC = 0
CPI		16	1110 1101 1010 0001	ED A1	. I I I	Vergleich A mit (HL) Z=1 wenn A=(HL) Z=0 wenn A≠(HL) INC von HL, DEC von BC+)
CPIR		21 /16	1110 1101 1011 0001	ED B1	. I I I	Vergleich A mit (HL) Z=1 wenn A=(HL) Z=0 wenn A≠(HL) INC von HL, DEC von BC bis BC = 0 oder A=(HL)
CPD		16	1110 1101 1010 1001	ED A9	. I I I	Vergleich A mit (HL) Z=1 wenn A=(HL) Z=0 wenn A≠(HL) DEC von HL und BC +)
CPDR		21 /16	1110 1101 1011 1001	ED B9	. I I I	Vergleich A mit (HL) Z=1 wenn A=(HL) Z=0 wenn A≠(HL) DEC von HL und BC bis BC = 0 oder A=(HL)

+) P/V: = 1, falls BC - 1 ≠ 0
sonst 0

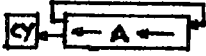
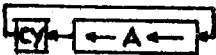
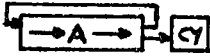
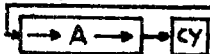
Mnem	Operand	T	Maschinencode Dual Hex	Flags CY Z S PV	Bedeutung/Hinweise
Arithmetik- und Logikbefehle (8 Bit)					
ADD	r	4	1000 0qqq	8.	! ! ! V A := A + r
ADD	n	7	1100 0110 xxxx xxxx	C6	! ! ! V A := A + n
ADD	(HL)	7	1000 0110	86	! ! ! V A := A + (HL)
ADD	(IX+d)	19	1101 1101 1000 0110 dddd dddd	DD 86	! ! ! V A := A + (IX+d)
ADD	(IY+d)	19	1111 1101 1000 0110 dddd dddd	FD 86	! ! ! V A := A + (IY+d)
ADC	r	4	1000 1qqq	8.	! ! ! V A := A + r + CY
ADC	n	7	1100 1110 xxxx xxxx	CE	! ! ! V A := A + n + CY
ADC	(HL)	7	1000 1110	8E	! ! ! V A := A + (HL) + CY
ADC	(IX+d)	19	1101 1101 1000 1110 dddd dddd	DD 8E	! ! ! V A := A + (IX+d) + CY
ADC	(IY+d)	19	1111 1101 1000 1110 dddd dddd	FD 8E	! ! ! V A := A + (IY+d) + CY
SUB	r	4	1001 0qqq	9.	! ! ! V A := A - r
SUB	n	7	1101 0110 xxxx xxxx	D6	! ! ! V A := A - n
SUB	(HL)	7	1001 0110	96	! ! ! V A := A - (HL)
SUB	(IX+d)	19	1101 1101 1001 0110 dddd dddd	DD 96	! ! ! V A := A - (IX+d)
SUB	(IY+d)	19	1111 1101 1001 0110 dddd dddd	FD 96	! ! ! V A := A - (IY+d)
SBC	r	4	1001 1qqq	9.	! ! ! V A := A - r - CY
SBC	n	7	1101 1110 xxxx xxxx	DE	! ! ! V A := A - n - CY
SBC	(HL)	7	1001 1110	9E	! ! ! V A := A - (HL) - CY
SBC	(IX+d)	19	1101 1101 1001 1110 dddd dddd	DD 9E	! ! ! V A := A - (IX+d) - CY
SBC	(IY+d)	19	1111 1101 1001 1110 dddd dddd	FD 9E	! ! ! V A := A - (IY+d) - CY

Mnem	Operand	T	Maschinencode		Flags					Bedeutung/Hinweise
			Dual	Hex	CY	Z	S	PV		
AND	r	4	1010 0qqq	A.	0	!	!	P		$A := A \wedge r$
AND	n	7	1110 0110 xxxx xxxx	E6	0	!	!	P		$A := A \wedge n$
AND	(HL)	7	1010 0110	A6	0	!	!	P		$A := A \wedge (HL)$
AND	(IX+d)	19	1101 1101 1010 0110 dddd dddd	DD A6	0	!	!	P		$A := A \wedge (IX+d)$
AND	(IY+d)	19	1111 1101 1010 0110 dddd dddd	FD A6	0	!	!	P		$A := A \wedge (IY+d)$
OR	r	4	1011 0qqq	B.	0	!	!	P		$A := A \vee r$
OR	n	7	1111 0110 xxxx xxxx	F6	0	!	!	P		$A := A \vee n$
OR	(HL)	7	1011 0110	B6	0	!	!	P		$A := A \vee (HL)$
OR	(IX+d)	19	1101 1101 1011 0110 dddd dddd	DD B6	0	!	!	P		$A := A \vee (IX+d)$
OR	(IY+d)	19	1111 1101 1011 0110 dddd dddd	FD B6	0	!	!	P		$A := A \vee (IY+d)$
XOR	r	4	1010 1qqq	A.	0	!	!	P		$A := A \nabla r$
XOR	n	7	1110 1110 xxxx xxxx	EE	0	!	!	P		$A := A \nabla n$
XOR	(HL)	7	1010 1110	AE	0	!	!	P		$A := A \nabla (HL)$
XOR	(IX+d)	19	1101 1101 1010 1110 dddd dddd	DD AE	0	!	!	P		$A := A \nabla (IX+d)$
XOR	(IY+d)	19	1111 1101 1010 1110 dddd dddd	FD AE	0	!	!	P		$A := A \nabla (IY+d)$
CMP	r	4	1011 1qqq	B.	!	!	!	V		Vergleich A mit r +)
CMP	n	7	1111 1110 xxxx xxxx	FE	!	!	!	V		Vergleich A mit n +)
CMP	(HL)	7	1011 1110	BE	!	!	!	V		Vergleich A mit (HL) +)
CMP	(IX+d)	19	1101 1101 1011 1110 dddd dddd	DD BE	!	!	!	V		Vergleich A mit (IX+d) +)
CMP	(IY+d)	19	1111 1101 1011 1110 dddd dddd	FD BE	!	!	!	V		Vergleich A mit (IY+d) +)

+) Vom Inhalt des Reg A wird der Operand abgezogen, die 12. Flagge gestellt, aber der Inhalt des Reg A bleibt unverändert (A - Operand)

Mnem	Operand	T	Maschinencode Dual Hex		Flags CY Z S PV	Bedeutung/Hinweise
INC	r	4	00zz z100		. V	r := r + 1
INC	(HL)	11	0011 0100	34	. V	(HL) := (HL) + 1
INC	(IX+d)	23	1101 1101 0011 0100 dddd dddd	DD 34	. V	(IX+d) := (IX+d) + 1
INC	(IY+d)	23	1111 1101 0011 0100 dddd dddd	FD 34	. V	(IY+d) := (IY+d) + 1
DEC	r	4	00zz z101		. V	r := r - 1
DEC	(HL)	11	0011 0101	35	. V	(HL) := (HL) - 1
DEC	(IX+d)	23	1101 1101 0011 0101 dddd dddd	DD 35	. V	(IX+d) := (IX+d) - 1
DEC	(IY+d)	23	1111 1101 0011 0101 dddd dddd	FD 35	. V	(IY+d) := (IY+d) - 1
Arithmetikbefehle (16 Bit)						
ADD	HL,BC	11	0000 1001	09	. . .	HL := HL + BC
ADD	HL,DE	11	0001 1001	19	. . .	HL := HL + DE
ADD	HL,HL	11	0010 1001	29	. . .	HL := HL + HL
ADD	HL,SP	11	0011 1001	39	. . .	HL := HL + SP
ADC	HL,BC	15	1110 1101 0100 1010	ED 4A	V	HL := HL + BC + CY
ADC	HL,DE	15	1110 1101 0101 1010	ED 5A	V	HL := HL + DE + CY
ADC	HL,HL	15	1110 1101 0110 1010	ED 6A	V	HL := HL + HL + CY
ADC	HL,SP	15	1110 1101 0111 1010	ED 7A	V	HL := HL + SP + CY
SBC	HL,BC	15	1110 1101 0100 0010	ED 42	V	HL := HL - BC - CY
SBC	HL,DE	15	1110 1101 0101 0010	ED 52	V	HL := HL - DE - CY
SBC	HL,HL	15	1110 1101 0110 0010	ED 62	V	HL := HL - HL - CY
SBC	HL,SP	15	1110 1101 0111 0010	ED 72	V	HL := HL - SP - CY

Mnem	Operand	T	Maschinencode Dual	Hex	Flags CY Z S PV	Bedeutung/Hinweise
ADD	IX,BC	15	1101 1101 0000 1001	DD 09	! . . .	IX := IX + BC
ADD	IX,DE	15	1101 1101 0001 1001	DD 19	! . . .	IX := IX + DE
ADD	IX,IX	15	1101 1101 0010 1001	DD 29	! . . .	IX := IX + IX
ADD	IX,SP	15	1101 1101 0011 1001	DD 39	! . . .	IX := IX + SP
ADD	IY,BC	15	1111 1101 0000 1001	FD 09	! . . .	IY := IY + BC
ADD	IY,DE	15	1111 1101 0001 1001	FD 19	! . . .	IY := IY + DE
ADD	IY,IY	15	1111 1101 0010 1001	FD 29	! . . .	IY := IY + IY
ADD	IY,SP	15	1111 1101 0011 1001	FD 39	! . . .	IY := IY + SP
INC	BC	6	0000 0011	03		BC := BC + 1
INC	DE	6	0001 0011	13		DE := DE + 1
INC	HL	6	0010 0011	23		HL := HL + 1
INC	SP	6	0011 0011	33		SP := SP + 1
INC	IX	10	1101 1101 0010 0011	DD 23		IX := IX + 1
INC	IY	10	1111 1101 0010 0011	FD 23		IY := IY + 1
DEC	BC	6	0000 1011	0B		BC := BC - 1
DEC	DE	6	0001 1011	1B		DE := DE - 1
DEC	HL	6	0010 1011	2B		HL := HL - 1
DEC	SP	6	0011 1011	3B		SP := SP - 1
DEC	IX	10	1101 1101 0010 1011	DD 2B		IX := IX - 1
DEC	IY	10	1111 1101 0010 1011	FD 2B		IY := IY - 1

Mnem	Operand	T	Maschinencode Dual	Hex	Flags CY Z S PV	Bedeutung/Hinweise
Allgemeine Arithmetik- und Steuerbefehle						
DAA		4	0010 0111	27	I I I P	korrigiert nach Addit. oder Subtr. gepackter Dezimalzahlen den Inhalt von A in gepackte Form (BCD-Code)
CPL		4	0010 1111	2F		bildet Einer-Komplement von A; bitweise Negation
NEG		8	1110 1101 0100 0100	ED 44	I I I V	bildet Zweier-Komplement von A; positive Zahl wird negativ
CCF		4	0011 1111	3F	I . . .	negiert CY-Flag
SCF		4	0011 0111	37	1 . . .	setzt CY-Flag auf 1
NOP		4	0000 0000	00		keine Operation; Leerbefehl
HALT		4	0111 0110	76		ZVE in Stopzustand
DI		4	1111 0011	F3		Interrupt-Annahme verboten; IFF := 0
EI		4	1111 1011	FB		Interrupt-Annahme erlaubt; IFF := 1
IM0		8	1110 1101 0100 0110	ED 46		Interrupt-Modus 0 (0 = Null)
IM1		8	1110 1101 0101 0110	ED 56		Interrupt-Modus 1
IM2		8	1110 1101 0101 1110	ED 5E		Interrupt-Modus 2
Rotations- und Verschiebebefehle						
RLCA		4	0000 0111	07	I . . .	
RLA		4	0001 0111	17	I . . .	
RRCA		4	0000 1111	0F	I . . .	
RRA		4	0001 1111	1F	I . . .	

Mnem	Operand	T	Maschinencode Dual	Hex	Flags CY Z S PV	Bedeutung/Hinweise
RLC	r	8	1100 1011 0000 0qqq	CB 0.	! ! ! P	
RLC	(HL)	15	1100 1011 0000 0110	CB 06	! ! ! P	
RLC	(IX+d)	23	1101 1101 1100 1011 dddd dddd 0000 0110	DD CB 06	! ! ! P	
RLC	(IY+d)	23	1111 1101 1100 1011 dddd dddd 0000 0110	FD CB 06	! ! ! P	
RL	r	8	1100 1011 0001 0qqq	CB 1.	! ! ! P	
RL	(HL)	15	1100 1011 0001 0110	CB 16	! ! ! P	
RL	(IX+d)	23	1101 1101 1100 1011 dddd dddd 0001 0110	DD CB 16	! ! ! P	
RL	(IY+d)	23	1111 1101 1100 1011 dddd dddd 0001 0110	FD CB 16	! ! ! P	
RRC	r	8	1100 1011 0000 1qqq	CB 0.	! ! ! P	
RRC	(HL)	15	1100 1011 0000 1110	CB 0E	! ! ! P	
RRC	(IX+d)	23	1101 1101 1100 1011 dddd dddd 0000 1110	DD CB 0E	! ! ! P	
RRC	(IY+d)	23	1111 1101 1100 1011 dddd dddd 0000 1110	FD CB 0E	! ! ! P	
RR	r	8	1100 1011 0001 1qqq	CB 1.	! ! ! P	
RR	(HL)	15	1100 1011 0001 1110	CB 1E	! ! ! P	
RR	(IX+d)	23	1101 1101 1100 1011 dddd dddd 0001 1110	DD CB 1E	! ! ! P	
RR	(IY+d)	23	1111 1101 1100 1011 dddd dddd 0001 1110	FD CB 1E	! ! ! P	

Mnem	Operand	T	Maschinencode		Flags	Bedeutung/Hinweise
			Dual	Hex	CY Z S PV	
SLA	r	8	1100 1011 0010 0qqq	CB 2.	! ! ! P	
SLA	(HL)	15	1100 1011 0010 0110	CB 26	! ! ! P	
SLA	(IX+d)	23	1101 1101 1100 1011 dddd dddd 0010 0110	DD CB 26	! ! ! P	
SLA	(IY+d)	23	1111 1101 1100 1011 dddd dddd 0010 0110	FD CB 26	! ! ! P	
SRA	r	8	1100 1011 0010 1qqq	CB 2.	! ! ! P	
SRA	(HL)	15	1100 1011 0010 1110	CB 2E	! ! ! P	
SRA	(IX+d)	23	1101 1101 1100 1011 dddd dddd 0010 1110	DD CB 2E	! ! ! P	
SRA	(IY+d)	23	1111 1101 1100 1011 dddd dddd 0010 1110	FD CB 2E	! ! ! P	
SRL	r	8	1100 1011 0011 1qqq	CB 3.	! ! ! P	
SRL	(HL)	15	1100 1011 0011 1110	CB 3E	! ! ! P	
SRL	(IX+d)	23	1101 1101 1100 1011 dddd dddd 0011 1110	DD CB 3E	! ! ! P	
SRL	(IY+d)	23	1111 1101 1100 1011 dddd dddd 0011 1110	FD CB 3E	! ! ! P	
RLD		18	1110 1101 0110 1111	ED 6F	. ! ! P	
RRD		18	1110 1101 0110 0111	ED 67	. ! ! P	

Mnem	Operand	T	Maschinencode Dual Hex	Flags CY Z S PV	Bedeutung/Hinweise
Bitmanipulationsbefehle					
BIT	b,r	8	1100 1011 01bb bqqq	CB	. ! x x Z-Flag = Bit negiert
BIT	b,(HL)	12	1100 1011 01bb b110	CB	. ! x x Z-Flag = Bit negiert
BIT	b,(IX+d)	20	1101 1101 1100 1011 dddd dddd 01bb b110	DD CB	. ! x x Z-Flag = Bit negiert
BIT	b,(IY+d)	20	1111 1101 1100 1011 dddd dddd 01bb b110	FD CB	. ! x x Z-Flag = Bit negiert
SET	b,r	8	1100 1011 11bb bqqq	CB	 Bit := 1
SET	b,(HL)	12	1100 1011 11bb b110	CB	 Bit := 1
SET	b,(IX+d)	20	1101 1101 1100 1011 dddd dddd 11bb b110	DD CB	 Bit := 1
SET	b,(IY+d)	20	1111 1101 1100 1011 dddd dddd 11bb b110	FD CB	 Bit := 1
RES	b,r	8	1100 1011 10bb bqqq	CB	 Bit := 0
RES	b, (HL)	12	1100 1011 10bb b110	CB	 Bit := 0
RES	b,(IX+d)	20	1101 1101 1100 1011 dddd dddd 10bb b110	DD CB	 Bit := 0
RES	b,(IY+d)	20	1111 1101 1100 1011 dddd dddd 10bb b110	FD CB	 Bit := 0

Mnem	Operand	T	Maschinencode Dual Hex	Flags CY Z S PV	Bedeutung/Hinweise
Sprungbefehle					
JMP	nn	10	1100 0011 anan anan ahah ahah	C3	 unbedingter Sprung nach nn
JMP	nn	10	1101 1010 anan anan ahah ahah	DA	 Sprung nach nn wenn CY-Flag = 1
JMP	nn	10	1101 0010 anan anan ahah ahah	D2	 Sprung nach nn wenn CY-Flag = 0
JMP	nn	10	1100 1010 anan anan ahah ahah	CA	 Sprung nach nn wenn Z-Flag = 1
JMP	nn	10	1100 0010 anan anan ahah ahah	C2	 Sprung nach nn wenn Z-Flag = 0
JMP	nn	10	1111 1010 anan anan ahah ahah	FA	 Sprung nach nn wenn S-Flag = 1
JMP	nn	10	1111 0010 anan anan ahah ahah	F2	 Sprung nach nn wenn S-Flag = 0
JMP	nn	10	1110 1010 anan anan ahah ahah	EA	 Sprung nach nn wenn P/V-Flag = 1
JMP	nn	10	1110 0010 anan anan ahah ahah	E2	 Sprung nach nn wenn P/V-Flag = 0
JR	e	12	0001 1000 eeee eeee	18	 unbedingter Sprung zur Relativ-Adr. PC+e
JRC	e	12 /7	0011 1000 eeee eeee	38	 Sprung zu PC+e wenn CY-Flag = 1
JRNC	e	12 /7	0011 0000 eeee eeee	30	 Sprung zu PC+e wenn CY-Flag = 0
JRZ	e	12 /7	0010 1000 eeee eeee	28	 Sprung zu PC+e wenn Z-Flag = 1
JRNZ	e	12 /7	0010 0000 eeee eeee	20	 Sprung zu PC+e wenn Z-Flag = 0
JMP	(HL)	4	1110 1001	E9	 unbedingter Sprung nach HL
JMP	(IX)	8	1101 1101 1110 1001	DD E9	 unbedingter Sprung nach IX
JMP	(IY)	8	1111 1101 1110 1001	FD E9	 unbedingter Sprung nach IY

Mnen	Operand	T	Maschinencode		Flage	Bedeutung/Hinweise
			Dual	Hex	CY Z S PV	
DJNZ	e	13 /8	0001 0000 eeee eeee	10		Sprung, solange B ≠ 0 Zyklusbefehl
CALL	nn	17	1100 1101 anan anan ahah ahah	CD		unbedingter Aufruf des Unterprogramms
CALL	C,nn	17 /10	1101 1100 anan anan ahah ahah	DC		Aufruf des Unterprogr. wenn CY-Flag = 1
CALL	NC,nn	17 /10	1101 0100 anan anan ahah ahah	D4		Aufruf des Unterprogr. wenn CY-Flag = 0
CALL	Z,nn	17 /10	1100 1100 anan anan ahah ahah	CC		Aufruf des Unterprogr. wenn Z-Flag = 1
CALL	NZ,nn	17 /10	1100 0100 anan anan ahah ahah	C4		Aufruf des Unterprogr. wenn Z-Flag = 0
CALL	M,nn	17 /10	1111 1100 anan anan ahah ahah	FC		Aufruf des Unterprogr. wenn S-Flag = 1
CALL	P,nn	17 /10	1111 0100 anan anan ahah ahah	F4		Aufruf des Unterprogr. wenn S-Flag = 0
CALL	PE,nn	17 /10	1110 1100 anan anan ahah ahah	EC		Aufruf des Unterprogr. wenn P/V-Flag = 1
CALL	PO,nn	17 /10	1110 0100 anan anan ahah ahah	E4		Aufruf des Unterprogr. wenn P/V-Flag = 0
RET		10	1100 1001	C9		unbedingte Rückkehr zur Aufrufstelle
RET	C	11 /5	1101 1000	D8		Rückkehr zur Aufruf- stelle wenn CY-Flag = 1
RET	NC	11 /5	1101 0000	D0		Rückkehr zur Aufruf- stelle wenn CY-Flag = 0
RET	Z	11 /5	1100 1000	C8		Rückkehr zur Aufruf- stelle wenn Z-Flag = 1
RET	NZ	11 /5	1100 0000	C0		Rückkehr zur Aufruf- stelle wenn Z-Flag = 0
RET	M	11 /5	1111 1000	F8		Rückkehr zur Aufruf- stelle wenn S-Flag = 1
RET	P	11 /5	1111 0000	F0		Rückkehr zur Aufruf- stelle wenn S-Flag = 0
RET	PE	11 /5	1110 1000	E8		Rückkehr zur Aufruf- stelle wenn P/V-Flag = 1
RET	PO	11 /5	1110 0000	E0		Rückkehr zur Aufruf- stelle wenn P/V-Flag = 0

Mnem	Operand	T	Maschinencode Dual	Hex	Flags CY Z S PV	Bedeutung/Hinweise
RETI		14	1110 1101 0100 1101	ED 4D		Rückkehr vom Interrupt
RETN		14	1110 1101 0100 0101	ED 45		Rückkehr vom nicht- maskierten Interrupt
RST	p	11	11pp p111			Aufruf festgelegter Adressen; Restart-Befehl
Ein- und Ausgabebefehle						
IN	(A) _n	10	1101 1011 xxxx xxxx	DB		Eingabe über Kanal n in Akkumulator
IN	r _i (n)	11	1110 1101 01zz z000	ED	. ! ! P	Eingabe über Kanal (C) in Register r +)
INF		11	1110 1101 0111 0000	ED 70	. ! ! P	setzt Flags entsprechend an Kanal (C) anliegendem Bitmuster; keine Eingabe
INI		15	1110 1101 1010 0010	ED A2	. ! x x	Eingabe in (HL) über (C) INC von HL, DEC von B +)
INIR		20 /15	1110 1101 1011 0010	ED B2	. 1 x x	Eingabe in (HL) über (C) INC von HL, DEC von B bis B = 0 +)
IND		15	1110 1101 1010 1010	ED AA	. ! x x	Eingabe in (HL) über (C) DEC von HL, DEC von B +)
INDR		20 /15	1110 1101 1011 1010	ED BA	. 1 x x	Eingabe in (HL) über (C) DEC von HL, DEC von B bis B = 0 +)
OUT	(n), A	11	1101 0011 xxxx xxxx	D3		Ausgabe des Akkumulators über Kanal n
OUT	(C), r	12	1110 1101 01zz z001	ED		Ausgabe von Register r über (C) +)
OUTI		15	1110 1101 1010 0011	ED A3	. ! x x	Ausgabe von (HL) über (C) INC von HL, DEC von B +)
OTIR		20 /15	1110 1101 1011 0011	ED B3	. 1 x x	Ausgabe von (HL) über (C) INC von HL, DEC von B bis B = 0 +)
OUTD		15	1110 1101 1010 1011	ED AB	. ! x x	Ausgabe von (HL) über (C) DEC von HL, DEC von B +)
OTDR		20 /15	1110 1101 1011 1011	ED BB	. 1 x x	Ausgabe von (HL) über (C) DEC von HL, DEC von B bis B = 0 +)

+) Während des E/A-Zyklus enthält der Adreßbus folgende Angaben:
 HWAT = Inhalt des Registers B
 NWAT = Inhalt des Registers C

Assembler-Anweisungen für SYPS K 1520

Markenfeld	Ass.-Anweieg.	Operand.-feld	Erklärung
NAME: <i>oder</i>	DS DS	nn <i>nn.n</i>	reserviert nn entsprechende Anzahl von Speicherplätzen (Byte), NAME ist Adresse des 1. Speicherplatzes, Inhalt der Byte ist undefiniert
NAME:	DB	n oder Zeichenkette	belegt einen Speicherplatz mit n=8 Bit oder belegt mehrere Byte mit Code der Zeichenkette 'ABC...Z'; NAME ist Adresse des (1.) Speicherplatzes
NAME:	DA DW	nn	speichert nn auf 2 Speicherplätzen ab, niederwert. Teil auf 1. Byte, dessen Adresse ist NAME, höherwert. Teil auf 2. Byte (nn = 16 Bit)
leer!	ORG	nn	steht am Anfang des Programms, mit nn wird die Anfangsadresse festgelegt
leer!	END	leer!	letzte Anweisung des Programms, kennzeichnet das Ende des Programms für den Assembler
<u>NAME</u>	EQU	nn	weist einer Variablen "NAME" den Wert nn zu; darf im Programm nur einmal benutzt werden
<u>NAME</u> :	DEF	nn	weist einer Variablen "NAME" den Wert nn zu; dieser Wert gilt bis zur nächsten DEF-Anweisung, die den Wert von "NAME" verändert
leer!	BN NAME	'NAME'	weist das Betriebssystem an, das Programm unter "NAME" (nur die ersten beiden Zeichen werden gewertet) zu verwalten
leer!	TITLE	<u>NAME</u>	veranlaßt den Druck von "NAME" in der Kopfzeile des Übersetzungs-Protokolls (listing)
leer!	EJECT	leer!	veranlaßt Formlervorschub beim Druck des Übersetzungs-Protokolls
<u>NAME</u>	MACRO	Liste	erste Zeile der Makrodefinition
<u>NAME</u>	ENDM	leer!	letzte Zeile der Makrodefinition
leer!	IF	<u>NAME</u>	eröffnet die bedingte Assemblierung
leer!	ENDIF	leer!	schließt die bedingte Assemblierung

: NAME = unbedingt erforderlich NAME = kann weggelassen werden

ISO-7-Bit-Code nach TGL 23207 (Sif-1000-Code)

Zeichen	hex	okt	dual	Zeichen	hex	okt	dual	Zeichen	hex	okt	dual
0	30	60	00110000	a	61	141	01100001	!	21	41	00100001
1	31	61	00110001	b	62	142	01100010	"	22	42	00100010
2	32	62	00110010	c	63	143	01100011	%	25	45	001000101
3	33	63	00110011	d	64	144	01100100	&	26	46	001000110
4	34	64	00110100	e	65	145	01100101	(	28	50	001010000
5	35	65	00110101	f	66	146	01100110	)	29	51	001010001
6	36	66	00110110	g	67	147	01100111	*	2A	52	001010100
7	37	67	00110111	h	68	150	01101000	+	2B	53	001010101
8	38	70	00111000	i	69	151	01101001	,	2C	54	001010100
9	39	71	00111001	j	6A	152	01101010	-	2D	55	001010101
				k	6B	153	01101011	.	2E	56	001010110
A	41	101	01000001	l	6C	154	01101100	/	2F	57	001010111
B	42	102	01000010	m	6D	155	01101101	:	3A	72	001110100
C	43	103	01000011	n	6E	156	01101110	;	3B	73	001110101
D	44	104	01000100	o	6F	157	01101111	<	3C	74	001110100
E	45	105	01000101	p	70	160	01110000	=	3D	75	001110101
F	46	106	01000110	q	71	161	01110001	>	3E	76	001110110
G	47	107	01000111	r	72	162	01110010	?	3F	77	001110111
H	48	110	01001000	s	73	163	01110011	\$	40	100	01000000
I	49	111	01001001	t	74	164	01110100	[	5B	133	01011011
J	4A	112	01001010	u	75	165	01110101	\	5C	134	01011100
K	4B	113	01001011	v	76	166	01110110	]	5D	135	01011101
L	4C	114	01001100	w	77	167	01110111	↑	5E	136	01011110
M	4D	115	01001101	x	78	170	01111000				
N	4E	116	01001110	y	79	171	01111001				
O	4F	117	01001111	z	7A	172	01111010				
P	50	120	01010000								
Q	51	121	01010001		20	40	00100000				Leerzeichen
R	52	122	01010010		'	27	47	00100111			Apostroph
S	53	123	01010011		#	23	43	00100011			Doppelkreuz
T	54	124	01010100		⌘	24	44	00100100			Irrungs-Zeichen
U	55	125	01010101		HT	09	11	00001001			Tabulator
V	56	126	01010110		LF	0A	12	00001010			Zeilenvorschub
W	57	127	01010111		CR	0D	15	00001101			Wagenvorlauf
X	58	130	01011000		NL	1E	36	00011110			Neue Zeile
Y	59	131	01011001		NUL	00	000	00000000			Lochstr.-Vorlauf
Z	5A	132	01011010		DEL	7F	177	01111111			Korrekt.-Lochung

Codierungstabelle für den Prozessor

U 880 (vollständig)

8-Bit-Arithm., Lade-, Logikbefehle													1. B. DD	
B	C	D	E	H	L	M	A	n	IXHIXL(4x)				IXHIXL(4x)	
LD B	40	41	42	43	44	45	46	47	06	44	45	46	44 45 46	
LD C	48	49	4A	4B	4C	4D	4E	4F	0E	4C	4D	4E	4C 4D 4E	
LD D	50	51	52	53	54	55	56	57	16	54	55	56	54 55 56	
LD E	58	59	5A	5B	5C	5D	5E	5F	1E	5C	5D	5E	5C 5D 5E	
LD H	60	61	62	63	64	65	66	67	26					66
LD L	68	69	6A	6B	6C	6D	6E	6F	2E					6E
LD M	70	71	72	73	74	75	76	77	36					
LD A	78	79	7A	7B	7C	7D	7E	7F	3E	7C	7D	7E	7C 7D 7E	
ADD	80	81	82	83	84	85	86	87	C6	84	85	86	84 85 86	
ADC	88	89	8A	8B	8C	8D	8E	8F	CE	8C	8D	8E	8C 8D 8E	
SUB	90	91	92	93	94	95	96	97	D6	94	95	96	94 95 96	
SBC	98	99	9A	9B	9C	9D	9E	9F	DE	9C	9D	9E	9C 9D 9E	
AND	A0	A1	A2	A3	A4	A5	A6	A7	E6	A4	A5	A6	A4 A5 A6	
XOR	A8	A9	AA	AB	AC	AD	AE	AF	EE	AC	AD	AE	AC AD AE	
OR	B0	B1	B2	B3	B4	B5	B6	B7	F6	B4	B5	B6	B4 B5 B6	
CMP	B8	B9	BA	BB	BC	BD	BE	BF	FE	BC	BD	BE	BC BD BE	
TNC	0A	0C	14	1C	24	2C	34	3C	24 2C					
DEC	05	0D	15	1D	25	2D	35	3D	25 2D					
LD														
IXH	60	61	62	63					67	26	64	65	67 26 64 65	
IXL	68	69	6A	6B					6F	2E	6C	6D	6F 2E 6C 6D	
(4x4)	70	71	72	73	74	75	76	77	36					

Austauschb.													Austauschb.		
LD	A, x	0A	1A	3A	ED57	ED5F							EX	DE, HL	
LD	x, A	02	12	32	ED47	ED4F							EXAF	08	
													EX	D9	
													EX (SP)	HL E3	
Hex:	00	08	10	18	20	28	30	38						DDE3	
RST	C7	CF	D7	DF	E7	EF	F7	FF							

ZVE-Steuerung													ZVE-Steuerung	
NOP	00	DI	F3	IM0	ED46	IM2	ED5E							
HALT	76	EI	FB	IM1	ED56									

Bit- und Rotationsbef.													Bit- und Rotationsbef.	
BIT	0	40	41	42	43	44	45	46	47					DDCB
SET	0	C0	C1	C2	C3	C4	C5	C6	C7					(K-2)
RES	0	80	81	82	83	84	85	86	87					
1	88	89	8A	8B	8C	8D	8E	8F	8E					
2	D0	D1	D2	D3	D4	D5	D6	D7	D6					
3	58	59	5A	5B	5C	5D	5E	5F	5E					
4	E0	E1	E2	E3	E4	E5	E6	E7	E6					
5	E8	E9	EA	EB	EC	ED	EE	EF	EE					
6	F0	F1	F2	F3	F4	F5	F6	F7	F6					
7	F8	F9	FA	FB	FC	FD	FE	FF	FE					

16-Bit-Arithmetik, Ladebefehle									
	AF	BC	DE	HL	IX	SP			
PUSH	x	F5	C5	D5	E5	DDE5			
POP	x	F1	C1	D1	E1	DDE1			
LD x, (nn)	01	11	21	DD21	31				
LD x, (nn)	ED48	ED58	2A	DD2A	ED7B				
LD (nn), x	ED43	ED53	22	DD22	ED73				
LD SP, x					DDF9				
ADD HL, x	09	19	29		39				
ADC HL, x	ED4A	ED5A	ED6A		ED7A				
SBC HL, x	ED42	ED52	ED62		ED72				
ADD IX, x	DD09	DD19			DD39				
INC x	03	13	23		DD23	33			
DEC x	0B	1B	2B		DD2B	3B			

Ein-Ausgabebeefehle 1. Byte									
	B	C	D	E	H	L	F	A	n
IN	40	48	50	58	60	68	70	78	DB
OUT	41	49	51	59	61	69	79	D3	

Sprung, UP-Aufruf, Rückkehrbef.

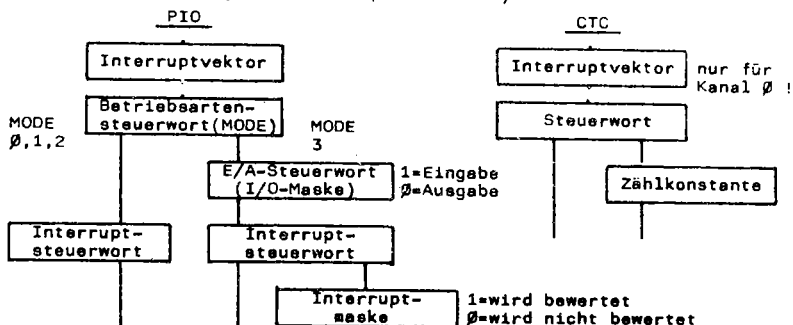
	C	NC	Z	NZ	PE	PO	M	P
JP	C3	DA	D2	CA	C2	EA	E2	FA
JR	18	38	30	28	20			F2
CALL	CD	DC	D4	CC	C4	EC	E4	FC
RET	C9	D8	D0	C8	C0	EB	E0	F8

Blockbef. 1. B. ED									
	A0	INI	A2	spez. Rot. befefhle					
LDI	A0			RLCA	07				
LDIR	B0	INIR	B2	RRCA	0F				
LDD	A8	IND	AA	SLA	17				
LDDR	B8	INDR	BA	RRA	1F				
CP1	A1	OUT1	A3	RRD	ED67				
CP1R	B1	OTIR	B3	RLD	ED6F				
CPD	A9	OUTD	AB						
CPDR	B9	OTDR	BB						

spez. Rot. befefhle									
RETI	ED4D								
RETN	ED45								
JMP (HL)	E9								
JMP (IX)	DD09								
DJNZ	10								
OUT 00 H									
ED71									

Fortsetzung, 1. Byte									
	B	C	D	E	H	L	M	A	DDCB
RLC	x	00	01	02	03	04	05	06	07
RRC	x	08	09	0A	0B	0C	0D	0E	0F
RL	x	10	11	12	13	14	15	16	17
RR	x	18	19	1A	1B	1C	1D	1E	1F
SLA	x	20	21	22	23	24	25	26	27
SLE	x	28	29	2A	2B	2C	2D	2E	2F
SRA	x	30	31	32	33	34	35	36	37
SRL	x	38	39	3A	3B	3C	3D	3E	3F
Vorbyte									
	B	C	D	E	H	L	M	A	DDCB
RLC (nn)	x	00	01	02	03	04	05	06	07
RRC (nn)	x	08	09	0A	0B	0C	0D	0E	0F
RL (nn)	x	10	11	12	13	14	15	16	17
RR (nn)	x	18	19	1A	1B	1C	1D	1E	1F
SLA (nn)	x	20	21	22	23	24	25	26	27
SLE (nn)	x	28	29	2A	2B	2C	2D	2E	2F
SRA (nn)	x	30	31	32	33	34	35	36	37
SRL (nn)	x	38	39	3A	3B	3C	3D	3E	3F
SET									
0 (nn)	x	C0	C1	C2	C3	C4	C5	C6	C7
1 (nn)	x	C8	C9	CA	CB	CC	CD	CE	CF
2 (nn)	x	D0	D1	D2	D3	D4	D5	D6	D7
3 (nn)	x	D8	D9	DA	DB	DC	DD	DE	DF
4 (nn)	x	E0	E1	E2	E3	E4	E5	E6	E7
5 (nn)	x	E8	E9	EA	EB	EC	ED	EE	EF
6 (nn)	x	F0	F1	F2	F3	F4	F5	F6	F7
7 (nn)	x	F8	F9	FA	FB	FC	FD	FE	FF
RES									
0 (nn)	x	80	81	82	83	84	85	86	87
1 (nn)	x	88	89	8A	8B	8C	8D	8E	8F
2 (nn)	x	90	91	92	93	94	95	96	97
3 (nn)	x	98	99	9A	9B	9C	9D	9E	9F
4 (nn)	x	A0	A1	A2	A3	A4	A5	A6	A7
5 (nn)	x	A8	A9	AA	AB	AC	AD	AE	AF
6 (nn)	x	B0	B1	B2	B3	B4	B5	B6	B7
7 (nn)	x	B8	B9	BA	BB	BC	BD	BE	BF

Programmierung PIO, CTC (Übersicht)



Codierungstabellen für die PIO U 855

Betriebsartensteuerwort		Interruptsteuerwort	
MODE		MODE 0,1,2	Interrupt gesperrt (INTg)
0	Byteausgabe		03 13 23 33 43 53 63 73
	0F 1F 2F 3F		Interrupt erlaubt (INTe)
1	Byteeingabe		83 93 A3 B3 C3 D3 E3 F3
	4F 5F 6F 7F		
2	bidirektionaler Betrieb (Kanal A)	MODE 3	Interrupt gesperrt
	(Kanal B nur Bitbetrieb)		Interrupt erlaubt
	0F 9F AF BF		k.neue Maske folgt
3	Bitbetrieb		k.neue Maske folgt
	CF DF EF FF		
		AND- H-aktiv	67 77 E7 F7
		Verkn. L-aktiv	47 57 C7 D7
		OR- H-aktiv	27 37 A7 B7
		Verkn. L-aktiv	07 17 87 97

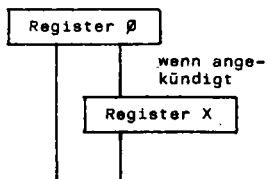
PIO-Interruptbewertung ist flankenaktiv! D.h.: H-aktiv = 0-1-Flanke usw.

Codierungstabelle für die CTC U 857

		Interrupt gesperrt				Interrupt erlaubt			
Zählvorgang		läuft		stoppt		läuft		stoppt	
Zählkonstante folgt		nein	ja	nein	ja	nein	ja	nein	ja
Zeitgeber (Timer)	Vor-teiler 1/16	Programm-start	01 11 05 15 03 13 07 17	81 91 85 95 83 93 87 97					
	Trigger- 0/1	start 1/0	19 1D 18 1F 09 0D 08 0F	99 9D 98 9F 89 8D 88 8F					
	Vor-teiler 1/256	Programm-start	21 31 25 35 23 33 27 37	A1 B1 A5 B5 A3 B3 A7 B7					
	Trigger- 0/1	start 1/0	39 3D 38 3F 29 2D 2B 2F	B9 BD BB BF A9 AD AB AF					
Zähler (Counter)	0/1		51 71 55 75 53 73 57 77	D1 F1 D5 F5 D3 F3 D7 F7					
	1/0		59 79 5D 7D 5B 7B 5F 7F	D9 F9 D5 F5 D3 F3 D7 F7					
			41 61 45 65 43 63 47 67	C1 E1 C5 E5 C3 E3 C7 E7					
			49 69 4D 6D 4B 6B 4F 6F	C9 E9 C5 E5 C3 E3 C7 E7					

Länge einer Periode in Zeitgebermode: $t_p = t_c \cdot VT \cdot TC$
 t_c = Periode des Systemtakts
 400 ns (2,5 MHz); VT = Vorteiler (16, 256); TC = Zeitkonstante

Programmierung SIO (Übersicht)



Registerübersicht

- Steuerregister (Schreibregister, Write-Register)
- WR 0 Reset-Grundbefehle, Zeiger Registerzeiger, CRC-Initialisierung, Initialisierung der Modi
- WR 1 Interrupt- und Wait/Ready-Modi Sende/Empfangs-Interrupt, Daten-Übertragungsbetriebsart
- WR 2 Interruptvektor (nur im Kanal B)
- WR 3 Empfängersteuerung Empfangsparameter, Steuerbits
- WR 4 Betriebsarten Parameter für Senden und Empfang
- WR 5 Sendersteuerung, CRC-Parameter Sendeparameter, Steuerbits
- WR 6 Synchronisation 8-Bit-Sende-Synchro-Zeichen oder erste acht Bit des 16-Bit-Synchro-Zeichens oder Such-Adressenfeld
- WR 7 Synchronisation 8-Bit-Empfangs-Synchro-Zeichen oder zweite acht Bit des 16-Bit-Synchro-Zeichens oder SDLC-Flag-Zeichen

- Statusregister (Leseregister, Read-Register)

- RR 0 allg. Status Pufferstatus Senden/Empfangen, Interruptstatus externer Status
- RR 1 Empfangsstatus Sender leer, Empfangsfehler, Status spez. Empfangsbedingungen
- RR 2 Interruptvektor nach Interruptart modifizierter Vektor - wenn zugelassen

Abarbeitungsfolge bei der Programmierung der einzelnen Betriebsarten

- Asynchrone Betriebsart

- WR 0 Kanal-RESET (WR 2 angekündigt)
- WR 2 Interruptvektor (nur Kanal B)
- WR 0 RESET für Extern/Status-Interrupt (WR 4 angekündigt)
- WR 4 Asynchrone Betriebsart, Parität, Stoppbit, Taktrate
- WR 0 -- (WR 3 ankündigen)
- WR 3 Empfängerfreigabe, Auto Enables, Empfangszeichenlänge
- WR 0 -- (WR 5 ankündigen)
- WR 5 Sendeanforderung (RTS), Senderfreigabe, Sendezeichenlänge, Betriebsbereitschaft (DTR)
- WR 0 RESET für Extern/Status-Interrupt (Wiederholung, WR 1 angekündigt)
- WR 1 Senderinterruptfreigabe, statusabhängiger Vektor (wenn erforderlich), Interrupt bei jedem Empfangszeichen, Sperrung der Wait/Ready-Funktion, Extern-Interrupt-Freigabe

Übertragung des ersten Datenbytes an die SIO, dadurch erst Interrupt möglich.

- Bisynd-Sende-Betriebsart

- WR 0 Kanal-RESET, Sender-CRC-Generator-RESET
- WR 0 -- (WR 2 ankündigen)
- WR 2 Interruptvektor (nur Kanal B)
- WR 0 -- (WR 3 ankündigen)
- WR 3 Auto Enables
- WR 0 -- (WR 4 ankündigen)
- WR 4 Sync-Mode, Parität, Taktrate 1:1
- WR 0 -- (WR 6 ankündigen)
- WR 6 Sync-Zeichen 1
- WR 0 RESET für Extern/Status-Interrupt, (WR 7 angekündigt)
- WR 7 Sync-Zeichen 2
- WR 0 RESET für Extern/Status-Interrupt (Wiederholung, WR 1 angekündigt)
- WR 1 statusabhängiger Vektor (wenn erforderlich), Extern-Interruptfreigabe, Sender-Interrupt-Freigabe oder Wait/Ready-Mode
- WR 0 -- (WR 5 ankündigen)
- WR 5 Sendeanforderung (RTS), Senderfreigabe, Bisynd-CRC-16, Sendezeichenlänge

Erstes Sync-Byte zur SIO. Möglichst mehrere vor der Datenübertragung senden.

- Biync-Empfange-Betriebsart

WR 0 Kanal-RESET, Empfänger-CRC-Prüfer-RESET
 WR 0 -- (WR 2 ankündigen)
 WR 2 Interruptvektor (nur Kanal B)
 WR 0 -- (WR 4 ankündigen)
 WR 4 Sync-Betriebsart, Parität, Takt rate 1:1
 WR 0 RESET für Extern/Status-Interrupt, (WR 5 angekündigt)
 WR 5 Biync-CRC-16, Betriebsbereitschaft (DTR)
 WR 0 -- (WR 3 ankündigen)
 WR 3 Sync-Zeichen-Ladesperre, Empfänger-CRC-Freigabe, Empfangszeich enlänge, Start für Suchbetrieb (Hunt Mode)
 WR 0 -- (WR 6 ankündigen)
 WR 6 Sync-Zeichen 1
 WR 0 -- (WR 7 ankündigen)
 WR 7 Sync-Zeichen 2
 WR 0 RESET für Extern/Status-Interrupt (Wiederholung, WR 1 angekündigt)
 WR 1 statusabhängiger Vektor (wenn erforderlich), Empfangsinterrupt nur beim ersten Zeichen, Extern-Interrupt-Freigabe
 WR 0 Interruptfreigabe für das nächste Empfangszeichen, (WR 3 angekündigt)
 WR 3 Empfängerfreigabe, Sync-Zeichen-Ladesperre (Wiederholung), Empfangswortlänge (Wiederholung), Auto Enables, Start für Suchbetrieb (Hunt Mode) (Wiederholung)

In den Sync-Modi können Sender- und Empfängerinitialisierung kombiniert werden. Dabei sind folgende allgemeine Bedingungen einzuhalten:

- RESET-Befehle, Interruptvektor, WR 4 (Mode, Parameter) müssen in dieser Reihenfolge vor WR 1, WR 3, WR 5, WR 6 und WR 7 ausgegeben werden
- Sender- oder Empfängerfreigabe erst wenn alle dazugehörigen Parameter initialisiert worden sind
- das Sync-Zeichen darf nicht mit der Folge "01.." oder "001.." beginnen
- es ist bei den Modi auf die entsprechenden Wiederholungen zu achten
- bei der mehrmaligen Ausgabe von Steuerregistern ist die ungewollte Änderung bereits initialisierter Parameter zu verhindern.

- SDLC-Sende-Betriebsart

WR 0 Kanal-RESET
 WR 0 -- (WR 2 ankündigen)
 WR 2 Interruptvektor (nur Kanal B)
 WR 0 -- (WR 3 ankündigen)
 WR 3 Auto Enables
 WR 0 RESET für Extern/Status-Interrupt, (WR 4 angekündigt)
 WR 4 SDLC-Betriebsart, Parität, Takt rate 1:1
 WR 0 RESET für Extern/Status-Interrupt (Wiederholung, WR 1 angekündigt)
 WR 1 Extern-Interrupt-Freigabe, statusabhängiger Vektor (wenn erforderlich, Sender-Interrupt-Freigabe oder Wait/Ready-Mode)
 WR 0 -- (WR 5 ankündigen)
 WR 5 Senderfreigabe, Sender-CRC-Freigabe, Sendewortlänge, SDLC-CRC, Betriebsbereitschaft (DTR), Sendeanforderung (RTS)
 WR 0 RESET für Sender-CRC-Generator

- SDLC-Empfange-Betriebsart

WR 0 Kanal-RESET
 WR 0 -- (WR 2 ankündigen)
 WR 2 Interruptvektor (nur Kanal B)
 WR 0 -- (WR 4 ankündigen)
 WR 4 Sync-Betriebsart, SDLC-Mode,
 WR 0 RESET für Extern/Status-Interrupt, (WR 5 angekündigt)
 WR 5 SDLC-CRC, Betriebsbereitschaft (DTR)
 WR 0 -- (WR 3 ankündigen)
 WR 3 Empfänger-CRC-Freigabe, Auto Enables, Start für Suchbetrieb (Hunt Mode), Empfangszeich enlänge, Adreßsuchbetriebsart
 WR 0 -- (WR 6 ankündigen)
 WR 6 Sekundäres Adressenfeld
 WR 0 -- (WR 7 ankündigen)
 WR 7 SDLC-Flag "0111110" (7E H)
 WR 7 RESET für Extern/Status-Interrupt, (Wiederholung, WR 1 angekündigt)
 WR 1 statusabhängiger Vektor (wenn erforderlich), Extern-Interruptfreigabe, Empfangsinterrupt nur beim ersten Zeichen
 WR 0 Freigabe des Interrupts beim nächsten Empfangszeichen, (WR 3 angekündigt)
 WR 3 Empfängerfreigabe, Empfänger-CRC-Freigabe, Start für Suchbetrieb (Hunt Mode), Auto Enables, Empfangszeich enlänge, Adreßsuchbetriebsart

Codierungstabellen für die SIO U 856

Steuerregister WR 0		es folgt Register							
		0	1	2	3	4	5	6	7
--	--	RESET-Empf.-CRC-Pr.	00 01 02 03	04 05 06 07	40 41 42 43	44 45 46 47	80 81 82 83	84 85 86 87	C0 C1 C2 C3
		RESET-Send.-CRC-Gen.	00 01 02 03	04 05 06 07	40 41 42 43	44 45 46 47	80 81 82 83	84 85 86 87	C0 C1 C2 C3
		Underrun/EOM-Latch	00 01 02 03	04 05 06 07	40 41 42 43	44 45 46 47	80 81 82 83	84 85 86 87	C0 C1 C2 C3
Sende Abortfolge (SDLC)	--	RESET-Empf.-CRC-Pr.	08 09 0A 0B	0C 0D 0E 0F	48 49 4A 4B	4C 4D 4E 4F	88 89 8A 8B	8C 8D 8E 8F	C8 C9 CA CB
		RESET-Send.-CRC-Gen.	08 09 0A 0B	0C 0D 0E 0F	48 49 4A 4B	4C 4D 4E 4F	88 89 8A 8B	8C 8D 8E 8F	C8 C9 CA CB
		Underrun/EOM-Latch	08 09 0A 0B	0C 0D 0E 0F	48 49 4A 4B	4C 4D 4E 4F	88 89 8A 8B	8C 8D 8E 8F	C8 C9 CA CB
RESET Ext/Status-Interrupt	--	RESET-Empf.-CRC-Pr.	10 11 12 13	14 15 16 17	50 51 52 53	54 55 56 57	90 91 92 93	94 95 96 97	D0 D1 D2 D3
		RESET-Send.-CRC-Gen.	10 11 12 13	14 15 16 17	50 51 52 53	54 55 56 57	90 91 92 93	94 95 96 97	D0 D1 D2 D3
		Underrun/EOM-Latch	10 11 12 13	14 15 16 17	50 51 52 53	54 55 56 57	90 91 92 93	94 95 96 97	D0 D1 D2 D3
Kanal-RESET	--	RESET-Empf.-CRC-Pr.	18 19 1A 1B	1C 1D 1E 1F	58 59 5A 5B	5C 5D 5E 5F	98 99 9A 9B	9C 9D 9E 9F	D8 D9 DA DB
		RESET-Send.-CRC-Gen.	18 19 1A 1B	1C 1D 1E 1F	58 59 5A 5B	5C 5D 5E 5F	98 99 9A 9B	9C 9D 9E 9F	D8 D9 DA DB
		Underrun/EOM-Latch	18 19 1A 1B	1C 1D 1E 1F	58 59 5A 5B	5C 5D 5E 5F	98 99 9A 9B	9C 9D 9E 9F	D8 D9 DA DB
Interruptfreigabe für nächstes Empfangszeichen	--	RESET-Empf.-CRC-Pr.	20 21 22 23	24 25 26 27	60 61 62 63	64 65 66 67	A0 A1 A2 A3	A4 A5 A6 A7	E0 E1 E2 E3
		RESET-Send.-CRC-Gen.	20 21 22 23	24 25 26 27	60 61 62 63	64 65 66 67	A0 A1 A2 A3	A4 A5 A6 A7	E0 E1 E2 E3
		Underrun/EOM-Latch	20 21 22 23	24 25 26 27	60 61 62 63	64 65 66 67	A0 A1 A2 A3	A4 A5 A6 A7	E0 E1 E2 E3
RESET Sender-INT-Pending	--	RESET-Empf.-CRC-Pr.	28 29 2A 2B	2C 2D 2E 2F	68 69 6A 6B	6C 6D 6E 6F	A8 A9 AA AB	AC AD AE AF	E8 E9 EA EB
		RESET-Send.-CRC-Gen.	28 29 2A 2B	2C 2D 2E 2F	68 69 6A 6B	6C 6D 6E 6F	A8 A9 AA AB	AC AD AE AF	E8 E9 EA EB
		Underrun/EOM-Latch	28 29 2A 2B	2C 2D 2E 2F	68 69 6A 6B	6C 6D 6E 6F	A8 A9 AA AB	AC AD AE AF	E8 E9 EA EB
Error-RESET	--	RESET-Empf.-CRC-Pr.	30 31 32 33	34 35 36 37	70 71 72 73	74 75 76 77	B0 B1 B2 B3	B4 B5 B6 B7	F0 F1 F2 F3
		RESET-Send.-CRC-Gen.	30 31 32 33	34 35 36 37	70 71 72 73	74 75 76 77	B0 B1 B2 B3	B4 B5 B6 B7	F0 F1 F2 F3
		Underrun/EOM-Latch	30 31 32 33	34 35 36 37	70 71 72 73	74 75 76 77	B0 B1 B2 B3	B4 B5 B6 B7	F0 F1 F2 F3
RETI (nur für Kanal A senden)	--	RESET-Empf.-CRC-Pr.	38 39 3A 3B	3C 3D 3E 3F	78 79 7A 7B	7C 7D 7E 7F	B8 B9 BA BB	BC BD BE BF	F8 F9 FA FB
		RESET-Send.-CRC-Gen.	38 39 3A 3B	3C 3D 3E 3F	78 79 7A 7B	7C 7D 7E 7F	B8 B9 BA BB	BC BD BE BF	F8 F9 FA FB
		Underrun/EOM-Latch	38 39 3A 3B	3C 3D 3E 3F	78 79 7A 7B	7C 7D 7E 7F	B8 B9 BA BB	BC BD BE BF	F8 F9 FA FB

Steuerregister WR 4		zusätzliche Paritätsinformation											
Betriebsart	Takt- rate	keine				gerade				ungerade			
1 Stoppbit	1:1	04	06	14	16	24	26	34	36	07	17	27	37
	1:16	44	46	54	56	64	66	74	76	47	57	67	77
	1:32	84	86	94	96	A4	A6	B4	B6	87	97	A7	B7
	1:64	C4	C6	D4	D6	E4	E6	F4	F6	C7	D7	E7	F7
asynch. 1,5 Stoppbit	1:1	--	--	--	--	--	--	--	--	--	--	--	--
	1:16	48	4A	58	5A	68	6A	78	7A	4B	5B	6B	7B
	1:32	88	8A	98	9A	A8	AA	B8	BA	8B	9B	AB	BB
	1:64	C8	CA	DB	DA	EB	EA	FB	FA	CB	DB	EB	FB
2 Stoppbit	1:1	0C	0E	1C	1E	2C	2E	3C	3E	0F	1F	2F	3F
	1:16	4C	4E	5C	5E	6C	6E	7C	7E	4F	5F	6F	7F
	1:32	8C	8E	9C	9E	AC	AE	BC	BE	8F	9F	AF	BF
	1:64	CC	CE	DC	DE	EC	EE	FC	FE	CF	DF	EF	FF
monosynchron	1:1	00 02				03				01			
bisynchron	1:1	10 12				13				11			
SDLC (F=01111110)	1:1	20 22				23				21			
Extern-Sync-Mode	1:1	30 32				33				31			

Steuerregister WR 1				Wait/Ready-Funktion				Wait/Ready gesperrrt			
				durch Wait	Empf. Ready	durch Send. Wait	Ready	WAIT ist "floatend"	READY ist "high"		
statusabh. Vektor				1	n	1	n	1	n	1	n
Ext./Status-Interrupt				j	n	j	n	j	n	j	n
Sender-INT bei	E-INT gesp.	j	E7 E3 A7 A3	C7 C3 87 83	07 27 03 23	47 67 43 63					
	E-INT beim	j	E6 E2 A6 A2	C6 C2 86 82	06 26 02 22	46 66 42 62					
	1. Zeichen	j	EF E8 AF AB	CF CB 8F 8B	0F 2F 0B 2B	4F 6F 4B 6B					
	leeres Reg.	j	EE EA AE AA	CE CA 8E 8A	0E 2E 0A 2A	4E 6E 4A 6A					
E-INT bei jed. Zei.(P)	E-INT bei	j	F7 F3 B7 B3	D7 D3 97 93	17 37 13 33	57 77 53 73					
	1. Zeichen	j	F6 F2 B6 B2	D6 D2 96 92	16 36 12 32	56 76 52 72					
	E-INT bei	j	FF FB BF BB	DF DB 9F 9B	1F 3F 1B 3B	5F 7F 5B 7B					
	jed. Zei.(P)	j	FE FA BE BA	DE DA 9E 9A	1E 3E 1A 3A	5E 7E 5A 7A					
Sender-INT gesp.	E-INT gesp.	j	E5 E1 A5 A1	C5 C1 85 81	05 25 01 21	45 65 41 61					
	E-INT beim	j	E4 E0 A4 A0	C4 C0 84 80	04 24 00 20	44 64 40 60					
	1. Zeichen	j	ED E9 AD A9	CD C9 8D 89	0D 2D 09 29	4D 6D 49 69					
	E-INT bei	j	EC E8 AC A8	CC C8 8C 88	0C 2C 08 28	4C 6C 48 68					
E-INT bei jed. Zei.(P)	E-INT bei	j	F5 F1 B5 B1	D5 D1 95 91	15 35 11 31	55 75 51 71					
	1. Zeichen	j	F4 F0 B4 B0	D4 D0 94 90	14 34 10 30	54 74 50 70					
	E-INT bei	j	FD F9 BD B9	DD D9 9D 99	1D 3D 19 39	5D 7D 59 79					
	jed. Zei.(P)	j	FC F8 BC B8	DC D8 9C 98	1C 3C 18 38	5C 7C 58 78					

Steuerregister WR 3				Auto Enable Sync.-Zei.-Ladesperre				kein Auto E Sync.-Zei.-Ladesperre			
				ja	nein	ja	nein	ja	nein	ja	nein
Start für Empfänger CRC-Berechnung				1	n	1	n	1	n	1	n
Anzahl der Empfangsbit											
Empfänger-freigabe	Adresse	Start	5	3F	37	3D	35	1F	17	1D	15
		für	6	8F	87	8D	85	9F	97	9D	95
		Synchr.-	7	7F	77	7D	75	5F	57	5D	55
		suche	8	FF	F7	F0	F5	0F	07	0D	05
	such-betrieb	keine	5	2F	27	2D	25	0F	07	0D	05
		Synchr.-	6	AF	A7	AD	A5	8F	87	8D	85
		suche	7	6F	67	6D	65	4F	47	4D	45
		suche	8	EF	E7	ED	E5	CF	C7	CD	C5
	kein Adresse	Start	5	3B	33	39	31	1B	13	19	11
		für	6	8B	83	89	81	9B	93	99	91
		Synchr.-	7	7B	73	79	71	5B	53	59	51
		suche	8	FB	F3	F9	F1	DB	D3	D9	D1
	such-betrieb	keine	5	2B	23	29	21	0B	03	09	01
		Synchr.-	6	AB	A3	A9	A1	8B	83	89	81
		suche	7	6B	63	69	61	4B	43	49	41
		suche	8	EB	E3	E9	E1	CB	C3	C9	C1
Empfänger gesperrt	Adresse	Start	5	3E	36	3C	34	1E	16	1C	14
		für	6	8E	86	8C	84	9E	96	9C	94
		Synchr.-	7	7E	76	7C	74	5E	56	5C	54
		suche	8	FE	F6	FC	F4	DE	D6	DC	D4
	such-betrieb	keine	5	2E	26	2C	24	0E	06	0C	04
		Synchr.-	6	AE	A6	AC	A4	8E	86	8C	84
		suche	7	6E	66	6C	64	4E	46	4C	44
		suche	8	EE	E6	EC	E4	CE	C6	CC	C4
	kein Adresse	Start	5	3A	32	38	30	1A	12	18	10
		für	6	8A	82	88	80	9A	92	98	90
		Synchr.-	7	7A	72	78	70	5A	52	58	50
		suche	8	FA	F2	F8	F0	DA	D2	D8	D0
	such-betrieb	keine	5	2A	22	28	20	0A	02	08	00
		Synchr.-	6	AA	A2	A8	A0	8A	82	88	80
		suche	7	6A	62	68	60	4A	42	48	40
		suche	8	EA	E2	E8	E0	CA	C2	C8	C0

Steuerregister WR 5			Sendeabbr. CRC- Polynom CRC-16 SDLC		Sendebehr. CRC- Polynom CRC-16 SDLC	
Start für Sender CRC-Berechnung			1	n	1	n
Anzahl der Sendebits			1	n	1	n
Sender frei gegeben	Bereitschaft gesetzt (DTR)	Sende- anforderung (RTS)	5	9F 9E 9B 9A	8F 8E 8B 8A	
			6	DF DE DB DA	CF CE CB CA	
			7	BF BE BB BA	AF AE AB AA	
	keine Sende- anforderung (RTS)		8	FF FE FB FA	EF EE EB EA	
			5	9D 9C 99 98	8D 8C 89 88	
			6	DD DC D9 D8	CD CC C9 C8	
			7	BD BC B9 B8	AD AC A9 A8	
	Bereitschaft nicht gesetzt (DTR)	Sende- anforderung (RTS)	8	FD FC F9 F8	ED EC E9 E8	
			5	1F 1E 1B 1A	0F 0E 0B 0A	
			6	5F 5E 5B 5A	4F 4E 4B 4A	
			7	3F 3E 3B 3A	2F 2E 2B 2A	
			8	7F 7E 7B 7A	6F 6E 6B 6A	
Sender gesperrt	Bereitschaft gesetzt (DTR)	Sende- anforderung (RTS)	5	10 1C 19 18	00 0C 09 08	
			6	5D 5C 59 58	4D 4C 49 48	
			7	3D 3C 39 38	2D 2C 29 28	
	keine Sende- anforderung (RTS)		8	7D 7C 79 78	6D 6C 69 68	
			5	97 96 93 92	87 86 83 82	
			6	07 06 03 02	C7 C6 C3 C2	
	Bereitschaft nicht gesetzt (DTR)	Sende- anforderung (RTS)	7	87 86 83 82	A7 A6 A3 A2	
			8	F7 F6 F3 F2	E7 E6 E3 E2	
			5	95 94 91 90	85 84 81 80	
	keine Sende- anforderung (RTS)		6	05 04 01 00	C5 C4 C1 C0	
			7	85 84 81 80	A5 A4 A1 A0	
			8	F5 F4 F1 F0	E5 E4 E1 E0	
Sender gesperrt	Bereitschaft gesetzt (DTR)	Sende- anforderung (RTS)	5	17 16 13 12	07 06 03 02	
			6	57 56 53 52	47 46 43 42	
			7	37 36 33 32	27 26 23 22	
	Bereitschaft nicht gesetzt (DTR)	Sende- anforderung (RTS)	8	77 76 73 72	67 66 63 62	
			5	15 14 11 10	05 04 01 00	
			6	55 54 51 50	45 44 41 40	
	keine Sende- anforderung (RTS)		7	35 34 31 30	25 24 21 20	
			8	75 74 71 70	65 64 61 60	

statusabhängiger Vektor

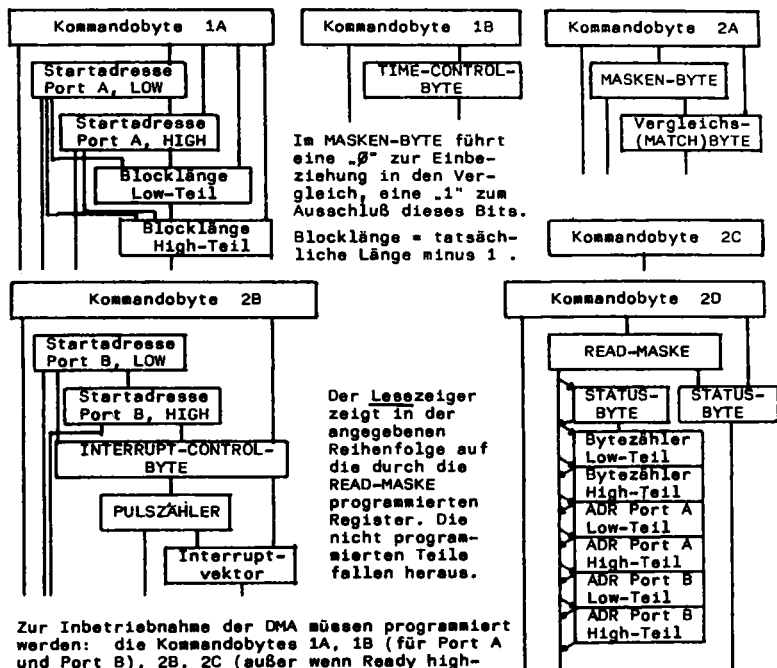
Bei zugelassener Interruptvektorbbeeinflussung werden die Bits V3, V2 und V1 des Statusregisters RR 2 (Kanal B) durch die Interruptart definiert gesetzt:

V3	V2	V1	Hex.	Bedeutung
0	0	0	x0 x1	Kanal B Senderegister leer
0	0	1	x2 x3	Kanal B Extern/Status-Veränderung
0	1	0	x4 x5	Kanal B Empfangszeichen vorhanden
0	1	1	x6 x7	Kanal B spezielle Empfangsbedingung*
1	0	0	x8 x9	Kanal A Senderegister leer
1	0	1	xA xB	Kanal A Extern/Status-Veränderung
1	1	0	xC xD	Kanal A Empfangszeichen vorhanden
1	1	1	xE xF	Kanal A spezielle Empfangsbedingung*
Ruhezustand (kein Interrupt)				
0	1	1	x6 x7	

*spezielle Empfangsbedingung: Paritätsfehler, Empfangsbitüberschreitung, Rahmenfehler, Rahmenende (SDLC).

Ob die Parität eine spez. Empfangsbedingung ist, bestimmt Bit D4 und D3 in WR 1. (P) = Paritätsfehler beeinflussen Vektor, (P) = Die Parität ist keine Empfangsbedingung (beeinflusst Vektor nicht).

Programmierung DMA (Übersicht)



Zur Inbetriebnahme der DMA müssen programmiert werden: die Kommandobytes 1A, 1B (für Port A und Port B), 2B, 2C (außer wenn Ready hochaktiv ist), sowie das INTERRUPT-CONTROL-BYTE. Die Startadressen sowie die Blocklänge müssen nur bei der Erstinitialisierung programmiert werden. Diese Informationen werden nur bei einem Hard- oder Software-RESET zerstört. Alle anderen Informationen brauchen nur bei Bedarf gegeben werden. Der Start der DMA erfolgt über CHIP ENABLE in Kommandobyte 2A (wenn letztes Programmierungsbyte) oder 87 H in Kommandobyte 2D. READY muß aktiv sein (ggf. durch erzwungenes Ready mit 0B3 H in Kommandobyte 2D). Wird „Interrupt vor Busanforderung“ programmiert erfolgt keine weitere Busanforderung, bis in Kommandobyte 2D „Freigabe nach RETI“ (0B7 H) und „Freigabe DMA“ (87 H) programmiert wurde und die Interruptserviceroutine abgearbeitet ist (Empfang des RETI-Befehls).

Der Interruptvektor wird bei programmierter Statusabhängigkeit wie folgt modifiziert:

V2	V1	
0	0	Interrupt bei READY
0	1	Interrupt, Vergleichsbyte gefunden
1	0	Interrupt am Blockende *
1	1	Interrupt am Blockende, Vergleichsbyte gefunden

* (wenn Vergleich programmiert, dann Vergleichsbyte nicht gefunden)

Codierungstabellen für die DMA U 252

Kommandobyte 1A			Daten- transfer					
			transfer		suche		+ suche	
			A/B	B/A	A/B	B/A	A/B	B/A
es folgt	Port lesen/schreiben	dann folgt						
-	-	-	05	01	06	02	07	03
	Blocklänge	Low-Teil	25	21	26	22	27	23
	Blocklänge	High-Teil	45	41	46	42	47	43
	Blocklänge	komplett	65	61	66	62	67	63
Startadresse	-	-	00	09	0E	0A	0F	0B
Port A	Blocklänge	Low-Teil	20	29	2E	2A	2F	2B
Low-Teil	Blocklänge	High-Teil	40	49	4E	4A	4F	4B
	Blocklänge	komplett	60	69	6E	6A	6F	6B
Startadresse	-	-	15	11	16	12	17	13
Port A	Blocklänge	Low-Teil	35	31	36	32	37	33
High-Teil	Blocklänge	High-Teil	55	51	56	52	57	53
	Blocklänge	komplett	75	71	76	72	77	73
Startadresse	-	-	10	19	1E	1A	1F	1B
Port A	Blocklänge	Low-Teil	30	39	3E	3A	3F	3B
komplett	Blocklänge	High-Teil	50	59	5E	5A	5F	5B
	Blocklänge	komplett	70	79	7E	7A	7F	7B

Kommandobyte 1B			A d r e s s e					
			wird inc.	wird dec.	ist fest			
			1	n	1	n	1	n
TIME-CONTROL-BYTE folgt			1	n	1	n	1	n
es wird	Zugriff des	programmiert Ports auf						
Port A	IORQ	MREQ	5C	1C	4C	0C	7C	6C
			54	14	44	04	74	64
Port B	IORQ	MREQ	58	18	48	08	78	68
			50	10	40	00	70	60

Kommandobyte 2A			es folgt					
			-	-	Vergleiche-byte			
			-	Such-	-	Such-	-	Such-
			-	maske	-	maske	-	maske
Stopp wenn Suchbyte gefunden			1	n	1	n	1	n
Chipfreigabe	Interrupt freigegeben		E4	E0	EC	E8	F4	F0
	gesperrt		C4	C0	CC	C8	D4	D0
Chip gesperrt	Interrupt freigegeben		A4	A0	AC	A8	B4	B0
	gesperrt		84	80	8C	88	94	90

Kommandobyte 2B			BYTE- CONTINUOUS BURST					
			MODE		MODE		MODE	
			1	n	1	n	1	n
INTERRUPT-CONTROL-BYTE folgt								
-	-	-	91	81	B1	A1	D1	C1
es folgt	Startadresse	Port B Low-Teil	95	85	B5	A5	D5	C5
		High-Teil	99	89	B9	A9	D9	C9
		komplett	9D	8D	BD	AD	DD	CD

Kommandobyte 2C			READY ist low-aktiv				READY ist high-aktiv			
			CE-Anschluß							
			dient CE							
Stopp an Blockende	CE/WAIT (Multiplex)		82	86	C2	C6	8A	8E	CA	CE
			92	96	D2	D6	9A	9E	DA	DE
			CE							
AUTO-REPEAT	CE/WAIT (Multiplex)		A2	A6	E2	E6	AA	AE	EA	EE
			B2	B6	F2	F6	BA	BE	FA	FE

Kommandobyte 2D		
Reset DMA, Stopp des DMA-Ver- kehrs, Rücksetzen der INT- Bedingungen	C3	Enable DMA, Chipfreigabe 87 Disable DMA, Chip gesperrt 83 Read Mask, Lesemaske folgt 8B
Reset Zeitverhalten, Rück- setzen auf Standardzyklus (T = 3)	Port A C7 Port B CB	Reset Read, Kommando Read Mask wird zurückgesetzt, Les- zeiger zeigt auf STATUS-BYTE A7
Laden Startadresse, Löschen des Bytezählers, Befehl wirkt nur auf Quellenport!	CF	Force Ready, erzwungenes Ready, Software-READY, gilt in allen Betriebsarten 83
Restart, Initialisier. bleibt!	D3	Enable RETI, DMA sendet erst dann wieder eine Busanfor- derung (BUSRQ), wenn eine Interruptservice routine abge- arbeitet wurde (RETI-Befehl) 87
Reset Interrupt, muß nach jedem INT erneut gegeben werden	A3	Clear, Löschung der Status- modifikationen im Interrupt- vektor (Match/Blockende) 8B
Enable Interrupt, muß nach jedem Rücksetzen des INT erneut gegeben werden	AB	
Disable Interrupt, INT gesp.	AF	

INTERRUPT-CONTROL-BYTE				Pulsenergieung einschalten/auschalten			
Interrupt...		Pulszähler folgt Interruptvektor folgt		j	n	j	n
Bevor Bus- anforderung erfolgt (BUSRQ)	wenn Suchbyte gefunden	am Blockende	statusabh.	j	7F FF 77 F7	7B FB 73 F3	
			INT-Vektor	n	6F EF 67 E7	6B EB 63 E3	
			fester Vektor	j	5F DF 57 D7	5B DB 53 D3	
		-	statusabh.	j	4F CF 47 C7	4B CB 43 C3	
			INT-Vektor	n	7D FD 75 F5	79 F9 71 F1	
			fester Vektor	j	6D ED 65 E5	69 E9 61 E1	
	-	am Blockende	statusabh.	j	5D DD 55 D5	59 D9 51 D1	
			INT-Vektor	n	4D CD 45 C5	49 C9 41 C1	
			fester Vektor	j	7E FE 76 F6	7A FA 72 F2	
		-	statusabh.	j	6E EE 66 E6	6A EA 62 E2	
			INT-Vektor	n	5E DE 56 D6	5A DA 52 D2	
			fester Vektor	j	4E CE 46 C6	4A CA 42 C2	
-	wenn Suchbyte gefunden	am Blockende	statusabh.	j	7C FC 74 F4	78 F8 70 F0	
			INT-Vektor	n	6C EC 64 E4	68 E8 60 E0	
			fester Vektor	j	5C DC 54 D4	58 D8 50 D0	
		-	statusabh.	j	4C CC 44 C4	48 C8 40 C0	
			INT-Vektor	n	3F BF 37 B7	3B BB 33 B3	
			fester Vektor	j	2F AF 27 A7	2B AB 23 A3	
	-	am Blockende	statusabh.	j	1F 9F 17 97	1B 9B 13 93	
			INT-Vektor	n	0F 8F 07 87	0B 8B 03 83	
			fester Vektor	j	3D BD 35 B5	39 B9 31 B1	
		-	statusabh.	j	2D AD 25 A5	29 A9 21 A1	
			INT-Vektor	n	1D 9D 15 95	19 99 11 91	
			fester Vektor	j	0D 8D 05 85	09 89 01 81	
	wenn Suchbyte gefunden	am Blockende	statusabh.	j	3E BE 36 B6	3A BA 32 B2	
			INT-Vektor	n	2E AE 26 A6	2A AA 22 A2	
			fester Vektor	j	1E 9E 16 96	1A 9A 12 92	
		-	statusabh.	j	0E 8E 06 86	0A 8A 02 82	
			INT-Vektor	n	3C BC 34 B4	38 BB 30 B0	
			fester Vektor	j	2C AC 24 A4	28 AB 20 A0	
	-	am Blockende	statusabh.	j	1C 9C 14 94	18 98 10 90	
			INT-Vektor	n	0C 8C 04 84	08 88 00 80	
			fester Vektor	j			

TIME-CONTROL-BYTE															
Die angegebenen Signale enden dann eine halbe Taktzeit vor dem Zyklusende. Die Übertragungsrate vergrößert sich dabei nicht.															
Zykluslänge als Anzahl der Systemtakte				-				IORQ				MREQ			
												IORQ MREQ			
-	4			CC	DC	EC	FC	C8	08	E8	F8	C4	D4	E4	F4
-	3			CD	DD	ED	FD	C9	09	E9	F9	C5	D5	E5	F5
-	2			CE	DE	EE	FE	CA	0A	EA	FA	C6	D6	E6	F6
-	1			CF	DF	EF	FF	C8	0B	EB	FB	C7	D7	E7	F7
RD	4			8C	9C	AC	BC	88	98	A8	B8	84	94	A4	B4
RD	3			8D	9D	AD	BD	89	99	A9	B9	85	95	A5	B5
RD	2			8E	9E	AE	BE	8A	9A	AA	BA	86	96	A6	B6
RD	1			8F	9F	AF	BF	8B	9B	AB	BB	87	97	A7	B7
WR	4			4C	5C	6C	7C	48	58	68	78	44	54	64	74
WR	3			4D	5D	6D	7D	49	59	69	79	45	55	65	75
WR	2			4E	5E	6E	7E	4A	5A	6A	7A	46	56	66	76
WR	1			4F	5F	6F	7F	4B	5B	6B	7B	47	57	67	77
RD	WR	4		0C	1C	2C	3C	08	18	28	38	04	14	24	34
RD	WR	3		0D	1D	2D	3D	09	19	29	39	05	15	25	35
RD	WR	2		0E	1E	2E	3E	0A	1A	2A	3A	06	16	26	36
RD	WR	1		0F	1F	2F	3F	0B	1B	2B	3B	07	17	27	37

READMASKE		Der Lesezeiger zeigt auf die programmierten Register :															
A d r e s s e Port A Port B		STATUS-BYTE								---							
		Bytezähler				Bytezähler											
		komp- plett	Low- Teil	High- Teil	---	komp- plett	Low- Teil	High- Teil	---								
komplett	komplett	7F	FF	78	F8	7D	F0	79	F9	7E	FE	7A	FA	7C	FC	78	F8
	Low-Teil	3F	BF	38	B8	3D	BD	39	B9	3E	BE	3A	BA	3C	BC	38	B8
	High-Teil	5F	DF	58	D8	5D	DD	59	D9	5E	DE	5A	DA	5C	DC	58	D8
	---	1F	9F	18	98	1D	9D	19	99	1E	9E	1A	9A	1C	9C	18	98
Low-Teil	komplett	6F	EF	68	E8	6D	ED	69	E9	6E	EE	6A	EA	6C	EC	68	E8
	Low-Teil	2F	AF	28	A8	2D	AD	29	A9	2E	AE	2A	AA	2C	AC	28	A8
	High-Teil	4F	CF	48	C8	4D	CD	49	C9	4E	CE	4A	CA	4C	CC	48	C8
	---	0F	8F	08	88	0D	8D	09	89	0E	8E	0A	8A	0C	8C	08	88
High-Teil	komplett	77	F7	73	F3	75	F5	71	F1	76	F6	72	F2	74	F4	70	F0
	Low-Teil	37	B7	33	B3	35	B5	31	B1	36	B6	32	B2	34	B4	30	B0
	High-Teil	57	D7	53	D3	55	D5	51	D1	56	D6	52	D2	54	D4	50	D0
	---	17	97	13	93	15	95	11	91	16	96	12	92	14	94	10	90
---	komplett	67	E7	63	E3	65	E5	61	E1	66	E6	62	E2	64	E4	60	E0
	Low-Teil	27	A7	23	A3	25	A5	21	A1	26	A6	22	A2	24	A4	20	A0
	High-Teil	47	C7	43	C3	45	C5	41	C1	46	C6	42	C2	44	C4	40	C0
	---	07	87	03	83	05	85	01	81	06	86	02	82	04	84	00	80

Decodierungstabelle für den hexadezimalen Code des Prozessors U 880 1.Byte F

NWT Ø	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
NOP	LD BC, n	LD BC, A	INC BC	INC BC	DEC B	LD B, n	RLCA	EXAF	ADD HL, BCA	LD BC	DEC BC	INC C	DEC C	LD C, n	RRCA
1	DJNZ DE, n	LD DE, A	INC DE	INC DE	DEC D	LD D, n	RLA	JR	ADD HL, DE	LD DE	DEC DE	INC E	DEC E	LD E, n	RRA
2	JRNZ HL, n	LD HL, A	INC HL	INC HL	DEC H	LD H, n	DAA	JRZ	ADD HL, HL	LD HL	DEC HL	INC L	DEC L	LD L, n	CPL
3	JRNC SP, n	LD SP, A	INC SP	INC SP	DEC M	LD M, n	SCF	JRC	LD HL, SP	LD SP	DEC A	INC A	DEC A	LD A, n	CCF
4	LD B, B	LD B, C	LD B, D	LD B, E	LD B, L	LD B, M	LD B, A	LD C, B	LD C, C	LD C, D	LD C, E	LD C, H	LD C, L	LD C, M	LD C, A
5	LD D, B	LD D, C	LD D, D	LD D, E	LD D, L	LD D, M	LD D, A	LD E, B	LD E, C	LD E, D	LD E, E	LD E, H	LD E, L	LD E, M	LD E, A
6	LD H, B	LD H, C	LD H, D	LD H, E	LD H, L	LD H, M	LD H, A	LD L, B	LD L, C	LD L, D	LD L, E	LD L, H	LD L, L	LD L, M	LD L, A
7	LD M, B	LD M, C	LD M, D	LD M, E	LD M, L	LD M, M	LD M, A	LD A, B	LD A, C	LD A, D	LD A, E	LD A, H	LD A, L	LD A, M	LD A, A
8	ADD B, C	ADD C, D	ADD D, E	ADD E, H	ADD H, L	ADD L, M	ADD M, A	ADD A, B	ADD B, C	ADD C, D	ADD D, E	ADD E, H	ADD H, L	ADD L, M	ADD M, A
9	SUB B, C	SUB C, D	SUB D, E	SUB E, H	SUB H, L	SUB L, M	SUB M, A	SUB A, B	SUB B, C	SUB C, D	SUB D, E	SUB E, H	SUB H, L	SUB L, M	SUB M, A
A	AND B, C	AND C, D	AND D, E	AND E, H	AND H, L	AND L, M	AND M, A	AND A, B	AND B, C	AND C, D	AND D, E	AND E, H	AND H, L	AND L, M	AND M, A
B	OR B, C	OR C, D	OR D, E	OR E, H	OR H, L	OR L, M	OR M, A	OR A, B	OR B, C	OR C, D	OR D, E	OR E, H	OR H, L	OR L, M	OR M, A
C	RNZ	POP BC	JPNZ	JMP	CANZ	PUSH nn	PUSH nn	RZ	RET	JPZ	BYT1	CAZ	CALL nn	ADC nn	RST 8
D	RNC	POP DE	JPMC	OUT	CANC	PUSH nn	PUSH nn	RC	EXX	JPC	IN	CAC	BYT1	SBC nn	RST 18
E	RPO	POP HL	JPPD	EX	CAPO	PUSH nn	PUSH nn	RPE	JMP	JPPE	EX	CAPE	BYT1	XOR nn	RST 28
F	RP	POP AF	JPP	DI	CAP	PUSH nn	PUSH nn	RM	RST	JPM	EI	CM	BYT1	CMR	RST 38

M = Speicher-
zelle ADR =
HL
HL = Spei-
cherzelle ADR

		2. Byte				1. Byte = CB															
Fortsetzung		1	2	3	4	5	6	7	8	9	A	B	C	D	E	F					
NWT	Ø	RLC	RLC	RLC	RLC	RLC	RLC	RLC	RLC	RLC	RLC	RLC	RLC	RLC	RLC	RLC					
Ø	Ø	B	C	O	E	H	L	M	A	B	C	D	E	H	L	M					
1	1	RL	RL	RL	RL	RL	RL	RL	RL	RR	RR	RR	RR	RR	RR	RR					
	B	C	O	E	H <td>L<td>M<td>A<td>B</td><td>C</td><td>D</td><td>E</td><td>H</td><td>L</td><td>M</td><td>A</td><td colspan="4"></td></td></td></td>	L <td>M<td>A<td>B</td><td>C</td><td>D</td><td>E</td><td>H</td><td>L</td><td>M</td><td>A</td><td colspan="4"></td></td></td>	M <td>A<td>B</td><td>C</td><td>D</td><td>E</td><td>H</td><td>L</td><td>M</td><td>A</td><td colspan="4"></td></td>	A <td>B</td> <td>C</td> <td>D</td> <td>E</td> <td>H</td> <td>L</td> <td>M</td> <td>A</td> <td colspan="4"></td>	B	C	D	E	H	L	M	A					
2	2	SLA	SLA	SLA	SLA	SLA	SLA	SLA	SLA	SRA	SRA	SRA	SRA	SRA	SRA	SRA					
	B	C	O	E	H <td>L<td>M<td>A<td>B</td><td>C</td><td>D</td><td>E</td><td>H</td><td>L</td><td>M</td><td>A</td><td colspan="4"></td></td></td></td>	L <td>M<td>A<td>B</td><td>C</td><td>D</td><td>E</td><td>H</td><td>L</td><td>M</td><td>A</td><td colspan="4"></td></td></td>	M <td>A<td>B</td><td>C</td><td>D</td><td>E</td><td>H</td><td>L</td><td>M</td><td>A</td><td colspan="4"></td></td>	A <td>B</td> <td>C</td> <td>D</td> <td>E</td> <td>H</td> <td>L</td> <td>M</td> <td>A</td> <td colspan="4"></td>	B	C	D	E	H	L	M	A					
3	3	SLE	SLE	SLE	SLE	SLE	SLE	SLE	SLE	SRL	SRL	SRL	SRL	SRL	SRL	SRL					
	B	C	D	E	H <td>L<td>M<td>A<td>B</td><td>C</td><td>D</td><td>E</td><td>H</td><td>L</td><td>M</td><td>A</td><td colspan="4"></td></td></td></td>	L <td>M<td>A<td>B</td><td>C</td><td>D</td><td>E</td><td>H</td><td>L</td><td>M</td><td>A</td><td colspan="4"></td></td></td>	M <td>A<td>B</td><td>C</td><td>D</td><td>E</td><td>H</td><td>L</td><td>M</td><td>A</td><td colspan="4"></td></td>	A <td>B</td> <td>C</td> <td>D</td> <td>E</td> <td>H</td> <td>L</td> <td>M</td> <td>A</td> <td colspan="4"></td>	B	C	D	E	H	L	M	A					
4	4	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT					
	Ø	B	Ø	C	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø					
5	5	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT					
	2	B	2	C	2	D	2	E	2	H	2	M	2	A	3	B	3				
6	6	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT					
	4	B	4	C	4	D	4	E	4	H	4	M	4	A	5	B	5				
7	7	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT					
	6	B	6	C	6	D	6	E	6	H	6	M	6	A	7	B	7				
8	8	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES					
	Ø	B	Ø	C	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø					
9	9	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES					
	2	B	2	C	2	D	2	E	2	H	2	M	2	A	3	B	3				
A	A	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES					
	4	B	4	C	4	D	4	E	4	H	4	M	4	A	5	B	5				
B	B	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES					
	6	B	6	C	6	D	6	E	6	H	6	M	6	A	7	B	7				
C	C	SET	SET	SET	SET	SET	SET	SET	SET	SET	SET	SET	SET	SET	SET	SET					
	Ø	B	Ø	C	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø					
D	D	SET	SET	SET	SET	SET	SET	SET	SET	SET	SET	SET	SET	SET	SET	SET					
	2	B	2	C	2	D	2	E	2	H	2	M	2	A	3	B	3				
E	E	SET	SET	SET	SET	SET	SET	SET	SET	SET	SET	SET	SET	SET	SET	SET					
	4	B	4	C	4	D	4	E	4	H	4	M	4	A	5	B	5				
F	F	SET	SET	SET	SET	SET	SET	SET	SET	SET	SET	SET	SET	SET	SET	SET					
	6	B	6	C	6	D	6	E	6	H	6	M	6	A	7	B	7				

Die gekennzeichneten Befehle werden vom Hersteller nicht geprüft. SLE wie SLA, nur statt Ø wird 1 eingesetzt. M = Speicherzelle ADDR = {HL}

Die gekennzeichneten Befehle werden vom Hersteller nicht geprüft. SLE wie SLA, nur statt Ø, wird .1. ein- geschoben. M = Speicher- zelle ADR = (HL)

Fortsetzung 2.Byte 4 5 6 7 8 9 A B C D E F

1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
NOP	LD	LD	INC	DEC	LD	RLCA	EXAF	ADD	LD	DEC	INC	DEC	LD	RRCA
	BC, n	BC, A	BC	B	B, n			1, BC	A, BC	BC	C	C	C, n	
DJNZ	LD	LD	INC	DEC	LD	RLA	JR	ADD	LD	DEC	INC	DEC	LD	RRA
	DE, n	DE, A	DE	D	D, n			1, DE	A, DE	E	E	E, n		
JRNZ	LD	LD	INC	DEC	LD	DAA	JRZ	ADD	LD	DEC	INC	DEC	LD	CPL
	1, n	1, n	1	1	1, n			1, 1	1, n	1	1	1	1	1, n
JRC	LD	LD	INC	DEC	LD	SCF	JRC	ADD	LD	DEC	INC	DEC	LD	CCF
	SP, n	SP, A	SP	1	1			1, SP	A, n	SP	A	A	A, n	
LD	B, B	C, B	D, B	E, B	1, B	LD	LD	LD	LD	LD	LD	LD	LD	C, C
	LD	LD	LD	LD	LD									C, A
LD	D, D	C, D	D, D	E, D	1, D	LD	LD	LD	LD	LD	LD	LD	LD	E, E
	LD	LD	LD	LD	LD									E, A
LD	1, B	1, C	1, D	1, E	1, 1, B	LD	LD	LD	LD	LD	LD	LD	LD	1, 1, C
	LD	LD	LD	LD	LD									1, 1, D
LD	LD	LD	LD	LD	LD	HALT	LD	LD	LD	LD	LD	LD	LD	LD
	LD	LD	LD	LD	LD									LD
ADD	ADD	ADD	ADD	ADD	ADD	ADD	ADD	ADD	ADD	ADD	ADD	ADD	ADD	A, A
	B, C	D, E	1, 1	1, 1	1, 1									A, A
SUB	SUB	SUB	SUB	SUB	SUB	SUB	SUB	SBC	SBC	SBC	SBC	SBC	SBC	SBC
	B, C	D, E	1, 1	1, 1	1, 1									(1)
AND	AND	AND	AND	AND	AND	AND	AND	XOR	XOR	XOR	XOR	XOR	XOR	XOR
	B, C	D, E	1, 1	1, 1	1, 1									(1)
OR	OR	OR	OR	OR	OR	OR	OR	OR	OR	OR	OR	OR	OR	OR
	B, C	D, E	1, 1	1, 1	1, 1									(1)
RNZ	POP	JPNZ	JMP	CANZ	PUSH	ADD	RZ	RET	JNZ	CALL	CALL	CALL	CALL	RST
	BC	BC	BC	BC	BC	n	n	n	n	n	n	n	n	n
RNC	POP	JPNC	OUT	CANC	PUSH	SUB	RC	EXX	JPC	IN	CAC	BYTEX	SBC	RST
	DE	DE	n	n	DE	n	n	n	n	n	n	n	n	n
RPO	POP	JPEX	CAPO	PUSH	AND	RST	RPE	JMPE	EX	CAPE	BYTEX	XOR	XOR	RST
	1	n	n	n	1	n	n	(1)	n	DE, HL	n	ED	n	28 H
RP	POP	JPF	DI	CAP	PUSH	OR	RM	JPM	EI	CAM	BYTEX	CHP	RST	38 H
	AF	n	n	AF	n	n	n	LD	SP	LD	FD	n	FD	n

Die gekennzeichneten Befehle werden vom Hersteller geprüft.
(nn) = Speicherzelle ADR
i = IX, wenn i = IY, wenn
1 = IY, wenn
11 = IXL/YL
niedrigerer Teil des Index-
registers
ih = IXL/IYH = höherwertiger Teil des Index-
registers
(id) = (ix+d) / (iy+d) = (Index-Register) + Speicherzelle ADR

NWHT
WHT
Ø

[illegible]

Fortsetzung

g	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	p
RLC	RLC	RLC	RLC	RLC	RLC	RLC	RLC	RLC	RLC	RLC	RLC	RLC	RLC	RLC	RLC	RLC
RL	RL	RL	RL	RL	RL	RL	RL	RL	RL	RL	RL	RL	RL	RL	RL	RL
SLA	SLA	SLA	SLA	SLA	SLA	SLA	SLA	SLA	SLA	SLA	SLA	SLA	SLA	SLA	SLA	SLA
SLE	SLE	SLE	SLE	SLE	SLE	SLE	SLE	SLE	SLE	SLE	SLE	SLE	SLE	SLE	SLE	SLE
BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT
RES0	RES0	RES0	RES0	RES0	RES0	RES0	RES0	RES0	RES0	RES0	RES0	RES0	RES0	RES0	RES0	RES0
RES2	RES2	RES2	RES2	RES2	RES2	RES2	RES2	RES2	RES2	RES2	RES2	RES2	RES2	RES2	RES2	RES2
RES4	RES4	RES4	RES4	RES4	RES4	RES4	RES4	RES4	RES4	RES4	RES4	RES4	RES4	RES4	RES4	RES4
RES6	RES6	RES6	RES6	RES6	RES6	RES6	RES6	RES6	RES6	RES6	RES6	RES6	RES6	RES6	RES6	RES6
SET0	SET0	SET0	SET0	SET0	SET0	SET0	SET0	SET0	SET0	SET0	SET0	SET0	SET0	SET0	SET0	SET0
SET2	SET2	SET2	SET2	SET2	SET2	SET2	SET2	SET2	SET2	SET2	SET2	SET2	SET2	SET2	SET2	SET2
SET4	SET4	SET4	SET4	SET4	SET4	SET4	SET4	SET4	SET4	SET4	SET4	SET4	SET4	SET4	SET4	SET4
SET6	SET6	SET6	SET6	SET6	SET6	SET6	SET6	SET6	SET6	SET6	SET6	SET6	SET6	SET6	SET6	SET6
SET8	SET8	SET8	SET8	SET8	SET8	SET8	SET8	SET8	SET8	SET8	SET8	SET8	SET8	SET8	SET8	SET8

Die gekennzeichneten Befehle werden vom Hersteller geprüft.
 (IX+d) = (IX+d) / (Indexregister) + Speicherzelle Adresse
 Die freien Bit-Befehlsfelder hier wird der Adreßteil von der CPU ignoriert.
 Alle Befehle dieser Seite sind Doppelbfehle. z.B.:
 LDC (IX+d) H = LD H, (IX+d) + Befehlsauflage:
 DD (FD) - CB-d-Opereationscode

Decodierungstabelle des Programmiercodes der CTC U 857															
NWT	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
HWT	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E
0	INTV T1iG	INTV T1iG	INTV T1iG	INTV T1iG	INTV T1iG	INTV T1iG	INTV T1iG	INTV T1iG	INTV T1iG	INTV T1iG	INTV T1iG	INTV T1iG	INTV T1iG	INTV T1iG	0
1	90H0	92H1 S	94H2	96H3 S,z	98H4	9AH5	9CH6	9EH7	9FH8	9AH1	9BH2	9CH3	9EH4	9FH5	1
2	INTV T1iG	INTV T1iG	INTV T1iG	INTV T1iG	INTV T1iG	INTV T1iG	INTV T1iG	INTV T1iG	INTV T1iG	INTV T1iG	INTV T1iG	INTV T1iG	INTV T1iG	INTV T1iG	2
3	INTV T2iG	INTV T2iG	INTV T2iG	INTV T2iG	INTV T2iG	INTV T2iG	INTV T2iG	INTV T2iG	INTV T2iG	INTV T2iG	INTV T2iG	INTV T2iG	INTV T2iG	INTV T2iG	3
4	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	4
5	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	5
6	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	6
7	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	INTV C1G	7
8	INTV T1iE	INTV T1iE	INTV T1iE	INTV T1iE	INTV T1iE	INTV T1iE	INTV T1iE	INTV T1iE	INTV T1iE	INTV T1iE	INTV T1iE	INTV T1iE	INTV T1iE	INTV T1iE	8
9	INTV T1iE	INTV T1iE	INTV T1iE	INTV T1iE	INTV T1iE	INTV T1iE	INTV T1iE	INTV T1iE	INTV T1iE	INTV T1iE	INTV T1iE	INTV T1iE	INTV T1iE	INTV T1iE	9
A	INTV T2iE	INTV T2iE	INTV T2iE	INTV T2iE	INTV T2iE	INTV T2iE	INTV T2iE	INTV T2iE	INTV T2iE	INTV T2iE	INTV T2iE	INTV T2iE	INTV T2iE	INTV T2iE	A
B	INTV T2iE	INTV T2iE	INTV T2iE	INTV T2iE	INTV T2iE	INTV T2iE	INTV T2iE	INTV T2iE	INTV T2iE	INTV T2iE	INTV T2iE	INTV T2iE	INTV T2iE	INTV T2iE	B
C	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	C
D	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	D
E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	E
F	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	INTV C1E	F

INTV = Interruptvektor
T = Timer
T1 = Timer mit Vorteiler 1/16
T2 = Timer mit Vorteiler 1/256
Ig = Interrupt gesperrt
Ie = Interrupt erlaubt
t = Start durch triggern
C = Counter (Zähler)
N = 1- β -flankennaktiv
P = 0-1-flankennaktiv
S = Stopp des laufendenden Zählvorganges
z = Zähl-
(zeit-)konstante folgt
Hx = Kanalangebe

Decodierungstabellen des Programmierungscodes der SIO U 856 Steuerregister WR Ø

NWT HWT:	Ø	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Ø
Ø	--,Ø	--,1	--,2	--,3	--,4	--,5	--,6	--,7	Ab,Ø	Ab,1	Ab,2	Ab,3	Ab,4	Ab,5	Ab,6	Ab,7	1
1	ES,Ø	ES,1	ES,2	ES,3	ES,4	ES,5	ES,6	ES,7	KR,Ø	KR,1	KR,2	KR,3	KR,4	KR,5	KR,6	KR,7	2
2	Ie,Ø	Ie,1	Ie,2	Ie,3	Ie,4	Ie,5	Ie,6	Ie,7	SP,Ø	SP,1	SP,2	SP,3	SP,4	SP,5	SP,6	SP,7	3
3	ER,Ø	ER,1	ER,2	ER,3	ER,4	ER,5	ER,6	ER,7	RI,Ø	RI,1	RI,2	RI,3	RI,4	RI,5	RI,6	RI,7	4
4	ECRC	ECRC	ECRC	ECRC	ECRC	ECRC	ECRC	ECRC	ECRC	ECRC	ECRC	ECRC	ECRC	ECRC	ECRC	ECRC	5
5	ES,Ø	ES,1	ES,2	ES,3	ES,4	ES,5	ES,6	ES,7	KR,Ø	KR,1	KR,2	KR,3	KR,4	KR,5	KR,6	KR,7	6
6	ECRC	ECRC	ECRC	ECRC	ECRC	ECRC	ECRC	ECRC	ECRC	ECRC	ECRC	ECRC	ECRC	ECRC	ECRC	ECRC	7
7	Ie,Ø	Ie,1	Ie,2	Ie,3	Ie,4	Ie,5	Ie,6	Ie,7	SP,Ø	SP,1	SP,2	SP,3	SP,4	SP,5	SP,6	SP,7	8
8	ER,Ø	ER,1	ER,2	ER,3	ER,4	ER,5	ER,6	ER,7	RI,Ø	RI,1	RI,2	RI,3	RI,4	RI,5	RI,6	RI,7	9
9	SCRC	SCRC	SCRC	SCRC	SCRC	SCRC	SCRC	SCRC	SCRC	SCRC	SCRC	SCRC	SCRC	SCRC	SCRC	SCRC	A
A	ES,Ø	ES,1	ES,2	ES,3	ES,4	ES,5	ES,6	ES,7	KR,Ø	KR,1	KR,2	KR,3	KR,4	KR,5	KR,6	KR,7	B
B	SCRC	SCRC	SCRC	SCRC	SCRC	SCRC	SCRC	SCRC	SCRC	SCRC	SCRC	SCRC	SCRC	SCRC	SCRC	SCRC	C
C	Ie,Ø	Ie,1	Ie,2	Ie,3	Ie,4	Ie,5	Ie,6	Ie,7	SP,Ø	SP,1	SP,2	SP,3	SP,4	SP,5	SP,6	SP,7	D
D	U/E	U/E	U/E	U/E	U/E	U/E	U/E	U/E	U/E	U/E	U/E	U/E	U/E	U/E	U/E	U/E	E
E	ES,Ø	ES,1	ES,2	ES,3	ES,4	ES,5	ES,6	ES,7	KR,Ø	KR,1	KR,2	KR,3	KR,4	KR,5	KR,6	KR,7	F
F	U/E	U/E	U/E	U/E	U/E	U/E	U/E	U/E	U/E	U/E	U/E	U/E	U/E	U/E	U/E	U/E	Ø
Ø	ER,Ø	ER,1	ER,2	ER,3	ER,4	ER,5	ER,6	ER,7	RI,Ø	RI,1	RI,2	RI,3	RI,4	RI,5	RI,6	RI,7	

ECRC = RESET Empfänger-CRC-Prüfer
 SCRC = RESET Sender-CRC-Generator
 U/E = Underrun/ECM-Latch
 Ab = Sende Ab-ortfolge i
 (SDLC)
 ES = RESET Ex-tern/Status-Interrupt
 KR = Kanal-RESET
 Ie = Interrupt-freigabe für nächstes Empfangszeichen
 SP = RESET Sender-Inter-rupt-Pending
 ER = Error-RESET
 RI = RETI
 x = Register Nr. x folgt

Fortsetzung		Steuerregister WR 1														
Ø	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
NWT		SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	
HWT		Wf, E	Wf, E	Wf, E	Wf, V	Wf, V	Wf, V	Wf, V	Wf, V	Wf, E	Wf, E	Wf, E	Wf, V	Wf, V	Wf, V	
Ø	1	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	
		Wf, E	Wf, E	Wf, E	Wf, V	Wf, V	Wf, V	Wf, V	Wf, V	Wf, E	Wf, E	Wf, E	Wf, V	Wf, V	Wf, V	
2		SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	
		Wf, E	Wf, E	Wf, E	Wf, V	Wf, V	Wf, V	Wf, V	Wf, V	Wf, E	Wf, E	Wf, E	Wf, V	Wf, V	Wf, V	
3		SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	
		Wf, E	Wf, E	Wf, E	Wf, V	Wf, V	Wf, V	Wf, V	Wf, V	Wf, E	Wf, E	Wf, E	Wf, V	Wf, V	Wf, V	
4		SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	
		Rh, E	Rh, E	Rh, E	Rh, V	Rh, V	Rh, V	Rh, V	Rh, E	Rh, E	Rh, E	Rh, E	Rh, V	Rh, V	Rh, V	
5		SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	
		Rh, E	Rh, E	Rh, E	Rh, V	Rh, V	Rh, V	Rh, E	Rh, E	Rh, E	Rh, E	Rh, E	Rh, V	Rh, V	Rh, V	
6		SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	
		Rh, E	Rh, E	Rh, E	Rh, V	Rh, V	Rh, V	Rh, E	Rh, E	Rh, E	Rh, E	Rh, E	Rh, V	Rh, V	Rh, V	
7		SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	
		Rh, E	Rh, E	Rh, E	Rh, V	Rh, V	Rh, V	Rh, E	Rh, E	Rh, E	Rh, E	Rh, E	Rh, V	Rh, V	Rh, V	
8		SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	
		RS, E	RS, E	RS, E	RS, V	RS, V	RS, V	RS, E	RS, E	RS, E	RS, E	RS, E	RS, V	RS, V	RS, V	
9		SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	
		RS, E	RS, E	RS, E	RS, V	RS, V	RS, V	RS, E	RS, E	RS, E	RS, E	RS, E	RS, V	RS, V	RS, V	
A		SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	
		RE, E	RE, E	RE, E	RE, V	RE, V	RE, V	RE, E	RE, E	RE, E	RE, E	RE, E	RE, V	RE, V	RE, V	
B		SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	
		RE, E	RE, E	RE, E	RE, V	RE, V	RE, V	RE, E	RE, E	RE, E	RE, E	RE, E	RE, V	RE, V	RE, V	
C		SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	
		WS, E	WS, E	WS, E	WS, V	WS, V	WS, V	WS, E	WS, E	WS, E	WS, E	WS, E	WS, V	WS, V	WS, V	
D		SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	
		WS, E	WS, E	WS, E	WS, V	WS, V	WS, V	WS, E	WS, E	WS, E	WS, E	WS, E	WS, V	WS, V	WS, V	
E		SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	SgEG	
		WE, E	WE, E	WE, E	WE, V	WE, V	WE, V	WE, E	WE, E	WE, E	WE, E	WE, E	WE, V	WE, V	WE, V	
F		SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	SgEP	
		WE, E	WE, E	WE, E	WE, V	WE, V	WE, V	WE, E	WE, E	WE, E	WE, E	WE, E	WE, V	WE, V	WE, V	

Wf = WAIT ist
floatend
Rh = READY ist
high
Se = Sender-INT
bei Leerem
Register
Sg = Sender-INT
gasperit
Ep = Empfänger-
Interrupt
gasperit
Es = Empfänger-
Interrupt beim
Ersten
Empfangszeichen
Ep = Empfänger-
Interrupt bei
Jedem Empfangs-
zeichen, Peri-
tät ist spez.
Empfangsbe-
dingung
Ep = wie Ep
jedoch ist
Parität keine
spez. Empfangs-
bedingung
V = Statusab-
hängiger Inter-
ruptvektor
E = Extern/
Status-Inter-
rupt erlaubt
WS = Wait-Fkt.
durch Sender-
register
RE = Ready-Fkt.
durch Empfangs-
register

Fortsetzung		Steuerregister WR 3														
Ø	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
NW	Eg	Ef	Eg	Ef	EgB	EfB	Eg	Ef	Eg	Ef	EgB	EfB	EgB	EfB	Ø	
HWT	5	L 5	L 5	L 5	5	L 5	L 5	C 5	C 5	CL 5	CL 5	C 5	C 5	CL 5	CL 5	
1	EgS	EfS	EgS	EfS	EgBS	EfBS	EgS	EfS	EgS	EfS	EgBS	EfBS	EgBS	EfBS	1	
	5	L 5	L 5	L 5	5	L 5	L 5	C 5	C 5	CL 5	CL 5	C 5	C 5	CL 5	CL 5	
2	Eg	Ef	Eg	Ef	EgB	EfB	Eg	Ef	Eg	Ef	EgB	EfB	EgB	EfB	2	
	A 5	A 5	AL 5	AL 5	A 5	A 5	AL 5	AL 5	CA 5	CA 5	CA 5	CA 5	CA 5	CA 5	CAL 5	
3	EgS	EfS	EgS	EfS	EgBS	EfBS	EgS	EfS	EgS	EfS	EgBS	EfBS	EgBS	EfBS	3	
	A 5	A 5	AL 5	AL 5	A 5	A 5	AL 5	AL 5	CA 5	CA 5	CA 5	CA 5	CA 5	CA 5	CAL 5	
4	Eg	Ef	Eg	Ef	EgB	EfB	Eg	Ef	Eg	Ef	EgB	EfB	EgB	EfB	4	
	7	L 7	L 7	L 7	7	L 7	L 7	C 7	C 7	CL 7	CL 7	C 7	C 7	CL 7	CL 7	
5	EgS	EfS	EgS	EfS	EgBS	EfBS	EgS	EfS	EgS	EfS	EgBS	EfBS	EgBS	EfBS	5	
	7	L 7	L 7	L 7	7	L 7	L 7	C 7	C 7	CL 7	CL 7	C 7	C 7	CL 7	CL 7	
6	Eg	Ef	Eg	Ef	EgB	EfB	Eg	Ef	Eg	Ef	EgB	EfB	EgB	EfB	6	
	A 7	A 7	AL 7	AL 7	A 7	A 7	AL 7	AL 7	CA 7	CA 7	CAL 7	CA 7	CA 7	CAL 7	CAL 7	
7	EgS	EfS	EgS	EfS	EgBS	EfBS	EgS	EfS	EgS	EfS	EgBS	EfBS	EgBS	EfBS	7	
	A 7	A 7	AL 7	AL 7	A 7	A 7	AL 7	AL 7	CA 7	CA 7	CAL 7	CA 7	CA 7	CAL 7	CAL 7	
8	Eg	Ef	Eg	Ef	EgB	EfB	Eg	Ef	Eg	Ef	EgB	EfB	EgB	EfB	8	
	6	L 6	L 6	L 6	6	L 6	L 6	C 6	C 6	CL 6	CL 6	C 6	C 6	CL 6	CL 6	
9	EgS	EfS	EgS	EfS	EgBS	EfBS	EgS	EfS	EgS	EfS	EgBS	EfBS	EgBS	EfBS	9	
	6	L 6	L 6	L 6	6	L 6	L 6	C 6	C 6	CL 6	CL 6	C 6	C 6	CL 6	CL 6	
A	Eg	Ef	Eg	Ef	EgB	EfB	Eg	Ef	Eg	Ef	EgB	EfB	EgB	EfB	A	
	A 6	AL 6	AL 6	A 6	A 6	AL 6	AL 6	CA 6	CA 6	CAL 6	CAL 6	CA 6	CA 6	CAL 6	CAL 6	
B	EgS	EfS	EgS	EfS	EgBS	EfBS	EgS	EfS	EgS	EfS	EgBS	EfBS	EgBS	EfBS	B	
	A 6	AL 6	AL 6	A 6	A 6	AL 6	AL 6	CA 6	CA 6	CAL 6	CAL 6	CA 6	CA 6	CAL 6	CAL 6	
C	Eg	Ef	Eg	Ef	EgB	EfB	Eg	Ef	Eg	Ef	EgB	EfB	EgB	EfB	C	
	8	L 8	L 8	L 8	8	L 8	L 8	C 8	C 8	CL 8	CL 8	C 8	C 8	CL 8	CL 8	
D	EgS	EfS	EgS	EfS	EgBS	EfBS	EgS	EfS	EgS	EfS	EgBS	EfBS	EgBS	EfBS	D	
	8	L 8	L 8	L 8	8	L 8	L 8	C 8	C 8	CL 8	CL 8	C 8	C 8	CL 8	CL 8	
E	Eg	Ef	Eg	Ef	EgB	EfB	Eg	Ef	Eg	Ef	EgB	EfB	EgB	EfB	E	
	A 8	AL 8	AL 8	A 8	A 8	AL 8	AL 8	CA 8	CA 8	CAL 8	CAL 8	CA 8	CA 8	CAL 8	CAL 8	
F	EgS	EfS	EgS	EfS	EgBS	EfBS	EgS	EfS	EgS	EfS	EgBS	EfBS	EgBS	EfBS	F	
	A 8	AL 8	AL 8	A 8	A 8	AL 8	AL 8	CA 8	CA 8	CAL 8	CAL 8	CA 8	CA 8	CAL 8	CAL 8	

Eg = Empfänger
 geperrt
 Ef = Empfänger
 freigegeben
 B = Adresssuch-
 betrieb
 S = Start für
 Synchronisa-
 tionsuche
 C = Start für
 Empfänger-CRC-
 Berechnung
 A = Auto-
 Enables
 L = Synchroni-
 sationszeichen-
 Ladesperre
 Zahlen = Anzahl
 der Empfangsbit

HWT #	Fortsetzung #	Steuerregister WR 4												F	
		1	2	3	4	5	6	7	8	9	A	B	C	D	E
1	Me	Me	Me	Me	A1	A1	A1	A1	-	-	-	-	A2	A2	A2
1	1Pu	1	1Pu	1	1Pu	1	1Pu	1	1Pg	-	-	-	1	1Pu	1
1	Bis	Bis	Bis	Bis	A1	A1	A1	A1	-	-	-	-	A2	A2	A2
1	1Pu	1	1Pu	1	1Pu	1	1Pu	1	1Pg	-	-	-	1	1Pu	1
2	SOLC	SOLC	SOLC	SOLC	A1	A1	A1	A1	-	-	-	-	A2	A2	A2
1	1Pu	1	1Pu	1	1Pu	1	1Pu	1	1Pg	-	-	-	1	1Pu	1
3	Ex	Ex	Ex	Ex	A1	A1	A1	A1	-	-	-	-	A2	A2	A2
1	1Pu	1	1Pu	1	1Pu	1	1Pu	1	1Pg	-	-	-	1	1Pu	1
4	-	-	-	-	A1	A1	A1	A1	A1,5	A1,5	A1,5	A1,5	A2	A2	A2
-	-	-	-	-	16	16Pu	16	16Pg	16	16Pu	16	16Pg	16	16Pu	16
5	-	-	-	-	A1	A1	A1	A1	A1,5	A1,5	A1,5	A1,5	A2	A2	A2
-	-	-	-	-	16	16Pu	16	16Pg	16	16Pu	16	16Pg	16	16Pu	16
6	-	-	-	-	A1	A1	A1	A1	A1,5	A1,5	A1,5	A1,5	A2	A2	A2
-	-	-	-	-	16	16Pu	16	16Pg	16	16Pu	16	16Pg	16	16Pu	16
7	-	-	-	-	A1	A1	A1	A1	A1,5	A1,5	A1,5	A1,5	A2	A2	A2
-	-	-	-	-	16	16Pu	16	16Pg	16	16Pu	16	16Pg	16	16Pu	16
8	-	-	-	-	A1	A1	A1	A1	A1,5	A1,5	A1,5	A1,5	A2	A2	A2
-	-	-	-	-	32	32Pu	32	32Pg	32	32Pu	32	32Pg	32	32Pu	32
9	-	-	-	-	A1	A1	A1	A1	A1,5	A1,5	A1,5	A1,5	A2	A2	A2
-	-	-	-	-	32	32Pu	32	32Pg	32	32Pu	32	32Pg	32	32Pu	32
A	-	-	-	-	A1	A1	A1	A1	A1,5	A1,5	A1,5	A1,5	A2	A2	A2
-	-	-	-	-	32	32Pu	32	32Pg	32	32Pu	32	32Pg	32	32Pu	32
B	-	-	-	-	A1	A1	A1	A1	A1,5	A1,5	A1,5	A1,5	A2	A2	A2
-	-	-	-	-	32	32Pu	32	32Pg	32	32Pu	32	32Pg	32	32Pu	32
C	-	-	-	-	A1	A1	A1	A1	A1,5	A1,5	A1,5	A1,5	A2	A2	A2
-	-	-	-	-	64	64Pu	64	64Pg	64	64Pu	64	64Pg	64	64Pu	64
D	-	-	-	-	A1	A1	A1	A1	A1,5	A1,5	A1,5	A1,5	A2	A2	A2
-	-	-	-	-	64	64Pu	64	64Pg	64	64Pu	64	64Pg	64	64Pu	64
E	-	-	-	-	A1	A1	A1	A1	A1,5	A1,5	A1,5	A1,5	A2	A2	A2
-	-	-	-	-	64	64Pu	64	64Pg	64	64Pu	64	64Pg	64	64Pu	64
F	-	-	-	-	A1	A1	A1	A1	A1,5	A1,5	A1,5	A1,5	A2	A2	A2
-	-	-	-	-	64	64Pu	64	64Pg	64	64Pu	64	64Pg	64	64Pu	64

A1 = asynchrone Mode,
 1 Stoppbit
 A1,5 = asynchrone Mode
 1,5 Stoppbit
 A2 = asynchrone Mode,
 2 Stoppbit
 Me = mono-
 synchrone Mode
 Bis = bi-
 synchrone Mode
 SOLC = SDLC-
 Mode, (Flag =
 01111111)
 Ex = Extern-
 synchroni-
 sation-Mode
 Zahlen = Takt-
 rate 1/x
 Pu = Parität
 ungerade
 Pg = Parität
 gerade

HWZ	p	Fortsetzung		Steuerregister WR 5												p
		1	2	3	4	5	6	7	8	9	A	B	C	D	E	
HWZ	p	Sg S5	Sg S5	SgA S5	Sg S5	Sg R5	SgA R5	Sg R5	Sf R5	Sf S5	SfA S5	SfA R5	Sf R5	SfA R5	SfA R5	1
1	1	Sg S5	Sg S5	SgA S5	Sg S5	Sg R5	SgA R5	Sg R5	Sf R5	Sf S5	SfA S5	SfA R5	Sf R5	SfA R5	SfA R5	1
2	2	Sg S7	Sg S7	SgA S7	Sg S7	Sg R7	SgA R7	Sg R7	Sf R7	Sf S7	SfA S7	SfA R7	Sf R7	SfA R7	SfA R7	2
3	3	Sg S7	Sg S7	SgA S7	Sg S7	Sg R7	SgA R7	Sg R7	Sf R7	Sf S7	SfA S7	SfA R7	Sf R7	SfA R7	SfA R7	3
4	4	Sg S6	Sg S6	SgA S6	Sg S6	Sg R6	SgA R6	Sg R6	Sf R6	Sf S6	SfA S6	SfA R6	Sf R6	SfA R6	SfA R6	4
5	5	Sg S6	Sg S6	SgA S6	Sg S6	Sg R6	SgA R6	Sg R6	Sf R6	Sf S6	SfA S6	SfA R6	Sf R6	SfA R6	SfA R6	5
6	6	Sg S8	Sg S8	SgA S8	Sg S8	Sg R8	SgA R8	Sg R8	Sf R8	Sf S8	SfA S8	SfA R8	Sf R8	SfA R8	SfA R8	6
7	7	Sg S8	Sg S8	SgA S8	Sg S8	Sg R8	SgA R8	Sg R8	Sf R8	Sf S8	SfA S8	SfA R8	Sf R8	SfA R8	SfA R8	7
8	8	SgB S5	SgB S5	SgBA S5	SgB S5	SgB R5	SgBA R5	SgB R5	SfB R5	SfB S5	SfBA S5	SfBA R5	SfB R5	SfBA R5	SfBA R5	8
9	9	SgB S5	SgB S5	SgBA S5	SgB S5	SgB R5	SgBA R5	SgB R5	SfB R5	SfB S5	SfBA S5	SfBA R5	SfB R5	SfBA R5	SfBA R5	9
A	A	SgB S7	SgB S7	SgBA S7	SgB S7	SgB R7	SgBA R7	SgB R7	SfB R7	SfB S7	SfBA S7	SfBA R7	SfB R7	SfBA R7	SfBA R7	A
B	B	SgB S7	SgB S7	SgBA S7	SgB S7	SgB R7	SgBA R7	SgB R7	SfB R7	SfB S7	SfBA S7	SfBA R7	SfB R7	SfBA R7	SfBA R7	B
C	C	SgB S6	SgB S6	SgBA S6	SgB S6	SgB R6	SgBA R6	SgB R6	SfB R6	SfB S6	SfBA S6	SfBA R6	SfB R6	SfBA R6	SfBA R6	C
D	D	SgB S6	SgB S6	SgBA S6	SgB S6	SgB R6	SgBA R6	SgB R6	SfB R6	SfB S6	SfBA S6	SfBA R6	SfB R6	SfBA R6	SfBA R6	D
E	E	SgB S8	SgB S8	SgBA S8	SgB S8	SgB R8	SgBA R8	SgB R8	SfB R8	SfB S8	SfBA S8	SfBA R8	SfB R8	SfBA R8	SfBA R8	E
F	F	SgB S8	SgB S8	SgBA S8	SgB S8	SgB R8	SgBA R8	SgB R8	SfB R8	SfB S8	SfBA S8	SfBA R8	SfB R8	SfBA R8	SfBA R8	F

Sg = Sender
 gesperrt
 Sf = Sender
 freigegeben
 B = Betriebs-
 bereiteinheit
 gesetz (DTR)
 A = Sender-
 anforderung
 gesetz (RTS)
 S = SDLC-CRC-
 Polynom
 R = CRC-16-
 Polynom
 X = Sender-
 abbruch
 C = Start für
 Sender-CRC-
 Berechnung
 Zahlen = Anzahl
 der Sendebits

Decodierungstabellen der Zustandsregister der SIO U 856																Statusregister RR			
																E			
NWT	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F				
β	ced	E	V	VE	L	LE	LV	LEV	ced	E	V	VE	L	LE	LV	LEV	β		
1	ced	E	V	EV	L	LE	LV	LEV	ced	E	V	VE	L	LE	LV	LEV	1		
2	ced	E	V	EV	L	LE	LV	LEV	ced	E	V	VE	L	LE	LV	LEV	2		
3	ced	E	V	EV	L	LE	LV	LEV	ced	E	V	VE	L	LE	LV	LEV	3		
4	N	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	4		
5	N	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	5		
6	N	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	6		
7	N	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	7		
8	B	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	8		
9	B	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	9		
A	B	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	A		
B	B	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	B		
C	BN	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	C		
D	BN	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	D		
E	BN	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	E		
F	BN	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	ced	F		

B = Unterbrechung (Break) bzw. Abbruch
 C = CTS ist
 high
 c = CTS ist low
 d = DCD ist
 high
 d = DCD ist low
 E = im Empfänger-
 puffer sind
 Daten vorhanden
 L = Senderpuffer
 ist leer
 N = Senderunter-
 lauf bzw. Nach-
 richtende
 S = SYNC ist
 high
 s = SYNC ist
 low
 v = Interrupt
 ist abhängig
 (nur über Kanal
 A lesbar)

Fortsetzung		Statusregister															RR 1		F	
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E						
0	8,2	8,2	6,0	S	-	S	-	S	-	S	-	S	-	S	-	S	-	S		
1	P	SP	P	SP	P	SP	P	SP	P	SP	P	SP	P	SP	P	SP	P	SP		
2	0	S0	0	S0	0	S0	0	S0	0	S0	0	S0	0	S0	0	S0	0	S0		
3	PO	SPO	PO	SPO	PO	SPO	PO	SPO	PO	SPO	PO	SPO	PO	SPO	PO	SPO	PO	SPO		
4	F	SF	F	SF	F	SF	F	SF	F	SF	F	SF	F	SF	F	SF	F	SF		
5	FP	SFP	FP	SFP	FP	SFP	FP	SFP	FP	SFP	FP	SFP	FP	SFP	FP	SFP	FP	SFP		
6	FO	SFO	FO	SFO	FO	SFO	FO	SFO	FO	SFO	FO	SFO	FO	SFO	FO	SFO	FO	SFO		
7	FPO	SFPO	FPO	SFPO	FPO	SFPO	FPO	SFPO	FPO	SFPO	FPO	SFPO	FPO	SFPO	FPO	SFPO	FPO	SFPO		
8	e	Se	e	Se	e	Se	e	Se	e	Se	e	Se	e	Se	e	Se	e	Se		
9	Pe	SPe	Pe	SPe	Pe	SPe	Pe	SPe	Pe	SPe	Pe	SPe	Pe	SPe	Pe	SPe	Pe	SPe		
A	Oe	SOe	Oe	SOe	Oe	SOe	Oe	SOe	Oe	SOe	Oe	SOe	Oe	SOe	Oe	SOe	Oe	SOe		
B	POe	SPOe	POe	SPOe	POe	SPOe	POe	SPOe	POe	SPOe	POe	SPOe	POe	SPOe	POe	SPOe	POe	SPOe		
C	Fe	SFe	Fe	SFe	Fe	SFe	Fe	SFe	Fe	SFe	Fe	SFe	Fe	SFe	Fe	SFe	Fe	SFe		
D	FPe	SFPe	FPe	SFPe	FPe	SFPe	FPe	SFPe	FPe	SFPe	FPe	SFPe	FPe	SFPe	FPe	SFPe	FPe	SFPe		
E	FOe	SFOe	FOe	SFOe	FOe	SFOe	FOe	SFOe	FOe	SFOe	FOe	SFOe	FOe	SFOe	FOe	SFOe	FOe	SFOe		
F	FPO	SFPO	FPO	SFPO	FPO	SFPO	FPO	SFPO	FPO	SFPO	FPO	SFPO	FPO	SFPO	FPO	SFPO	FPO	SFPO		

S = alle Bits
geendet (asynch.
Moden). sonst
ist "S" ständig
gesetzt
P = Paritäts-
fehler
O = Empfänger-
überlauf
F = CRC- bzw.
Formatfehler
e = Formatende
bei SCLC-Mode
Zahlen x,y =
Reestcode, x =
Anzahl der
Datenbit in vor-
letzten Byte,
y = Anzahl der
Datenbit im
letzten Byte

S = alle Bits
 gesendet (asynch.
 Moden), sonst
 ist „S“ ständig
 gesetzt
 P = Paritäts-
 fehler
 O = Empfänger-
 überlauf
 F = CRC- bzw.
 Formatfehler.
 e = Formatende
 bei SDC-Mode
 Zahlen x,y =
 Restcode, x =
 Anzahl der
 Datenbit im vor-
 letzten Byte,
 y = Anzahl der
 Datenbit im
 letzten Byte

Decodierungstabellen des Programmiercodes der DMA U 257 Kommandobyte 1A - 20																
NWT	g	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	MD	TOb	SOb	TSOb	A	TOb	SOb	TSOb	A	TOb	SOb	TSOb	A	TOb	SOb	TSOb
1	B	MD	TSOb	A	TOb	SOb	TSOb	A	TOb	SOb	TSOb	A	TOb	SOb	TSOb	A
2	M1	MD	TSOb	A	TOb	SOb	TSOb	A	TOb	SOb	TSOb	A	TOb	SOb	TSOb	A
3	Mx	L1	L1	L1	Mx	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1
4	Mx	L1	L1	L1	Mx	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1
5	M1	L1	L1	L1	M1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1
6	Mx	L1	L1	L1	Mx	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1
7	B	TOb	SOb	TSOb	A	TOb	SOb	TSOb	A	TOb	SOb	TSOb	A	TOb	SOb	TSOb
8	CgIg	By	RICE	Cg	CgIg	By	RICE	Cf	CgIg	By	RICE	RES	CgIg	By	RICE	RES
9	CgIg	By	RIMU	--	--	--	--	--	--	--	--	--	--	--	--	--
A	CgIg	Co	RICE	RES	CgIg	Co	RICE	RES	CgIg	Co	RICE	RES	CgIg	Co	RICE	RES
B	CgIg	Co	RIMU	Force	CgIg	Co	RIMU	Enab	CgIg	Co	RIMU	Read	CgIg	Co	RIMU	Read
C	CgIg	By	RICE	RES	CgIg	By	RICE	RES	CgIg	By	RICE	RES	CgIg	By	RICE	RES
D	CgIg	By	RIMU	Re-	CgIg	By	RIMU	Re-	CgIg	By	RIMU	Re-	CgIg	By	RIMU	Re-
E	CgIg	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
F	CgIg	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

av = Stopp bei Vergleich; T = Datentransfer; TS = Transfer+Suche; t = Time-Kontrollbyte folgt; v = Vergleichsbyte folgt; x = ADR bleibt fest

ev = Stopp bei Vergleich; T = Datentransfer; TS = Transfer+Suche; t = Time-Kontrollbyte folgt; v = Vergleichsbyte folgt; x = ADR bleibt fest

A = Port A
 Ah = es folgt
 Start-ADR für
 P. A, High-Teil
 Al = wie Ah
 aber Low-Teil
 Av = wie Ah,
 aber vollständig
 ar = Auto Repeat
 B = Port B
 Bh = es folgt
 Start-ADR für
 P. B, High-Teil
 Bl = wie Bh,
 aber Low-Teil
 Bu = BURST-Mode
 Bv = wie Bh
 aber vollständig
 By = BYTE-Mode
 Cf = Chipfrei-
 gebe
 Cg = Chip
 gesperrt
 Co = CONTINUOUS-
 Mode
 d = DEC ADR
 I = IORQ-Zugriff
 If = INT freigebe
 Ig = INT Resp.
 Ii = INC ADR
 Ik = INT-Kon-
 trollbyte folgt
 M = MREQ-Zugriff
 m = Maske folgt
 Qa = Quelle ist
 Port A
 Qb = -ist P. B
 Rh = READY ist
 high-aktiv
 Rl = -ist low-
 S = Datenauche
 se = Stopp am
 Blockende

Fortsetzung

Fortsetzung		INTERRUPT-CONTROL-BYTE																
HWZ	HWZ	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
--	V	E	VE	G	GV	GE	GVE	--	V	E	VE	G	GV	GE	GVE	1	B = Interrupt bevor der DMA-Is den Bus über BUSRQ anfordert	
--	V	E	VE	G	GV	GE	GVE	--	V	E	VE	G	GV	GE	GVE	2	E = Interrupt am Blockende	
S	SV	SE	SVE	SG	SGV	SCE	SGVE	S	SV	SE	SVE	SG	SGV	SCE	SGVE	3	G = Pulsgene- rator ein- schalten	
B	BV	BE	BVE	BG	BGV	BCE	BGVE	B	BV	BE	BVE	BG	BGV	BCE	BGVE	4	S = statusab- hängiger Inter- ruptvektor	
B	BV	BE	BVE	BG	BGV	BCE	BGVE	B	BV	BE	BVE	BG	BGV	BCE	BGVE	5	V = Interrupt wenn Vergleichs- byte (Suchbyte) gefunden	
SB	SBV	SBE	SBVE	SBG	SBGV	SBCE	SBGVE	SB	SBV	SBE	SBVE	SBG	SBGV	SBCE	SBGVE	6	.P = Pulse- zähler folgt	
SB	SBV	SBE	SBVE	SBG	SBGV	SBCE	SBGVE	SB	SBV	SBE	SBVE	SBG	SBGV	SBCE	SBGVE	7	.V = Interrupt- vektor folgt	
--	V	E	VE	G	GV	GE	GVE	--	V	E	VE	G	GV	GE	GVE	8		
--	V	E	VE	G	GV	GE	GVE	--	V	E	VE	G	GV	GE	GVE	9		
S	SV	SE	SVE	SG	SGV	SCE	SGVE	S	SV	SE	SVE	SG	SGV	SCE	SGVE	A		
S	SV	SE	SVE	SG	SGV	SCE	SGVE	S	SV	SE	SVE	SG	SGV	SCE	SGVE	B		
B	BV	BE	BVE	BG	BGV	BCE	BGVE	B	BV	BE	BVE	BG	BGV	BCE	BGVE	C		
B	BV	BE	BVE	BG	BGV	BCE	BGVE	B	BV	BE	BVE	BG	BGV	BCE	BGVE	D		
SB	SBV	SBE	SBVE	SBG	SBGV	SBCE	SBGVE	SB	SBV	SBE	SBVE	SBG	SBGV	SBCE	SBGVE	E		
SB	SBV	SBE	SBVE	SBG	SBGV	SBCE	SBGVE	SB	SBV	SBE	SBVE	SBG	SBGV	SBCE	SBGVE	F		

Fortsetzung		TIME-CONTROL-BYTE															
NWT	β	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	β
NWT	β	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	β
1	1	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	1
2	2	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	2
3	3	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	3
4	4	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	4
5	5	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	5
6	6	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	6
7	7	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	7
8	8	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	8
9	9	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	9
A	A	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	A
B	B	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	WRHI	B
C	C	MI	MI	MI	MI	MI	MI	MI	MI	MI	MI	MI	MI	MI	MI	MI	C
D	D	MI	MI	MI	MI	MI	MI	MI	MI	MI	MI	MI	MI	MI	MI	MI	D
E	E	MI	MI	MI	MI	MI	MI	MI	MI	MI	MI	MI	MI	MI	MI	MI	E
F	F	MI	MI	MI	MI	MI	MI	MI	MI	MI	MI	MI	MI	MI	MI	MI	F

I = IORD...
M = MREQ...
W = WR endet
eine halbe Tak-
zeit vor dem
Zyklusende
Zahlen = Zyklus-
länge alle Anzeile
der Systemtakte
(Standardzyklus
ist 5 Takte)

I = IORQ....
 M = MREQ....
 R = RD....
 L = WR endet
 eine halbe Takt-
 zeit vor dem
 Zyklusende
 Zahlen = Zyklus-
 länge als Anzahl
 der Systatakte
 (Standardzyklus
 ist 3 Takte)

NWT HWT	Ø	Fortsetzung		READMASKE												
		1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F

Es folgen in
angenehmer
reinerform
und konnte
lesen werden:
Port A
High-Teil
Port A
Low-Teil
Av = Adresse
Port A
vollständig
Bh = Adresse
Port B
High-Teil
B1 = Adresse
Port B
Low-Teil
Bv = Adresse
Port B
vollständig
S = STATUS-BYTE
Zb = Zähler-
stand
High-Teil
Z1 = Zähler-
stand
Low-Teil
Zv = Zähler-
stand
vollständig
Die "Reihen-
folge" be-
zieht sich auf
die Tabelle.

