

Literatur

zum Vorwort:

- /1/ Schmidt, V.: Digitalelektronisches Praktikum,
2.Aufl. Stuttgart 1977 (Teubner Studienskripten Band 19)

zu Kapitel 1:

- /2/ National Semiconductor Application Notes:
AN-43: Tri-State Logic in Modular Systems, 1971
AN-45: Characteristics and applications of Tri-State IC's,
1971
- Hahn, W.; Bauer, F.L.: Physikalische und elektrotechnische Grundlagen für Informatiker, Berlin-Heidelberg-New York 1975 (Heidelberger Taschenbücher Band 147)
 - Advanced Micro Devices Application Note: Operation of the 9130/40 4K Static Rams
 - Texas Instruments Bulletin MOSA1: Memory System Design Utilizing TMS 4050/4051 4K Dynamic RAMs, 1976
 - Texas Instruments Bulletin MOSA3: Introduction to Refreshing TI 4K Dynamic RAMs, 1976

zu Kapitel 2:

- National Semiconductor Application Note AN-89: How to Design with Programmable Logic Arrays
- Klar, R.: Digitale Rechenautomaten, 2. Auflage Berlin 1976
- Advanced Micro Devices Application Note: TTL MSI Arithmetic Logic Units

zu Kapitel 3:

- /3/ Lewin, D.: Theory and Design of Digital Computers, London 1972
- Klar, R.: Digitale Rechenautomaten, 2. Auflage Berlin 1976

zu Kapitel 5:

- /4/ Huse, H.: I²L: Die Technologie und ihre Anwendung in einem schnellen 4-Bit-Mikroprozessorelement. Elektronik 1976, H.2, S. 79-82
- Motorola Semiconductor: M 6800 Microprocessor Applications Manual
- Motorola Semiconductor: M 6800 Microprocessor Programming Manual

zu Kapitel 7:

- /5/ Motorola Semiconductor: Motorola EXORciser Users Guide
- /6/ Farnbach, W.A.: Bring up your uP 'bit-by-bit'. Electronic Design 15, Juli 1976 - auch als Hewlett Packard Application Note Nr. 167-19
- Hewlett Packard Application Notes - Data Domain Series - Nr. 167-9, -11,-13,-14,-15,-16,-17,-18,-19

zu Kapitel 8:

- /7/ Whiting, J.S.: An Efficient Software Method for Implementing Polynomial Error Detection Codes; Computer Design, März 1975, S.73-77

Anhang: M 6800-Befehlsvorrat

Bezeichnung	Adressierung Accu Indekator Direkt Extended Implied Relative	Arithmetische/Boolesche Operation, Beschreibung M = Speicherzelle mit beliebiger Adresse A = Akkumulator A, B = Akkumulator B
TRANSFERBEBEHLE		
LDA	xxxx	A := M bzw. B := M
PSH	x	zuerst Akku-Inhalt in oberste Stack-Zelle, dann Stack Pointer dekrementieren
PUL	x	zuerst Stack-Pointer inkrementieren, dann Inhalt der obersten Stackzelle in den Akku
STA	xxx	M := A bzw. M := B
TAB	x	B := A
TBA	x	A := B
TAP	x	CCR := A
TPA	x	A := CCR
* LDX	xxxx	X(H-Byte) := M, X(L-Byte) := M+1
* LDS	xxxx	SP(H-Byte) := M, SP(L-Byte) := M+1
* STX	xxx	M := X(H-Byte), M+1 := X(L-Byte)
* STS	xxx	M := SP(H-Byte), M+1 := SP(L-Byte)
* TXS	x	SP := X - 1
* TSX	x	X := SP + 1

* 2-Byte-Operation

Anhang: M 6800-Befehlsvorrat

Bezeichnung	Adres- sierung	Arithmetische/Boolesche Operation, Beschreibung
ARITHMETISCHE UND LOGISCHE OPERATIONEN		
ADD	xxxx	A := A + M bzw. B := B + M
ABA	x	A := A + B
ADC	xxxx	A := A + M + C bzw. B := B + M + C
AND	xxxx	A := A UND M bzw. B := B UND M (Bit-weise)
SUB	xxxx	A := A - M bzw. B := B - M
SBC	xxxx	A := A - M - C bzw. B := B - M - C
SBA	x	A := A - B
Subtract Accumulators	x xx	M := 0 bzw. A := 0 bzw. B := 0
CLR	x xx	M := -M bzw. A := -A bzw. B := -B
COM	x xx	M := 0 - M bzw. A := 0 - A bzw. B := 0 - B
NEG	x xx	Wandelt das Ergebnis einer 1-Byte-Dual- Addition in Akku A in 2 BCD-Ziffern
DAA	x	M := M - 1 bzw. A := A - 1 bzw. B := B - 1
DEC	x xx	X := X - 1
* DEX	x	SP := SP - 1
* DES	x	A := A EXOR M bzw. B := B EXOR M
EOR	xxxx	M := M + 1 bzw. A := A + 1 bzw. B := B + 1
INC	x xx	X := X + 1
* INX	x	SP := SP + 1
* INS	x	

* 2-Byte-Operation

Anhang: M 6800-Befehlsvorrat

Bezeichnung	Adressierung	Arithmetische/Boolesche Operation, Beschreibung
ORA	xxxx	A := A v M bzw. B := B v M (Bit-weise)
ROL	x xx	 bzueglich M, A oder B
ROR	x xx	 bezueglich M, A oder B
ASL	x xx	 bezueglich M, A oder B
ASR	x xx	 bezueglich M, A oder B
LSR	x xx	 bezueglich M, A oder B
TEST- UND VERGLEICHOPERATIONEN		
BIT Test	xxxx	Teste (A UND M) bzw. (B UND M); setze CCR entsprechend
CMP Compare	xxxx	Teste (A - M) bzw (B - M); setze CCR entsprechend
CBA Compare Accumulators	x	Teste (A - B); setze CCR entsprechend
* CPX Compare Index Reg.	x	Teste X(H-Byte)-M und X(L-Byte); setze CCR entsprechend
TST Test, Zero or Minus	x xx	Teste (M-0) bzw. (A-0) bzw. (B-0); setze CCR entsprechend
VERZWEIGUNGS- UND SPRUNGBEFEHLE		
JMP JUMP	xx	Unbedingter Sprung; beliebige Zieladresse
JSR Jump to Subroutine	xx	Unterprogramm-Sprung mit Retten des PC auf den Stack; SP := SP - 2
RTS Return from Subroutine	x	Ruecksprung aus einem Unterprogramm; Restaurierung des PC; SP := SP + 2

* 2-Byte-Operation

Anhang: M 6800-Befehlsvorrat

	Bezeichnung	Adressierung	Arithmetische/Boolesche Operation, Beschreibung
BRA	BR <u>A</u> nch Always	x	unbedingt
BCC	Branch if <u>C</u> arry Clear	x	Carry = 0
B <u>C</u> S	Branch if <u>C</u> arry Set	x	Carry = 1
BEQ	Branch if <u>E</u> qual Zero	x	das letzte Ergebnis null war
BGE	Branch if <u>G</u> reater or <u>E</u> qual Zero	x	das letzte Ergebnis null oder positiv war
BGT	Branch if <u>G</u> reater Zero	x	das letzte Ergebnis positiv war
BHI	Branch if <u>H</u> igher	x	der Minuend bei der letzten Vergleichs- oder Subtraktions-Operation grösser als der Subtrahend war
BLE	Branch if <u>L</u> ess than or <u>E</u> qual to Zero	x	das letzte Ergebnis kleiner oder gleich null war
BL <u>S</u>	Branch if <u>L</u> ower or <u>S</u> ame	x	der Minuend bei der letzten Vergleichs- oder Subtraktions-Operation kleiner oder gleich dem Subtrahenden war
BL <u>T</u>	Branch if <u>L</u> ess Than Zero	x	das Ergebnis der letzten Vergleichs- oder Subtraktions-Operation negativ war
B <u>M</u> I	Branch if <u>M</u> inus	x	das Bit b7 des letzten Ergebnisses = 1 war
B <u>N</u> E	Branch if <u>N</u> ot Equal zero	x	das letzte Ergebnis ungleich null war
B <u>V</u> C	Branch if <u>O</u> verflow Clear	x	das Overflow-Bit des CCR null ist
B <u>V</u> S	Branch if <u>O</u> verflow Set	x	das Overflow-Bit des CCR eins ist
B <u>P</u> L	Branch if <u>P</u> lus	x	das Bit b7 des letzten Ergebnisses 0 = war
B <u>S</u> R	Branch to Subroutine		unbedingte Verzweigung mit Retten des PC auf den Stack; SP := SP - 2

Verzweigung mit Relativweite
-125 bis +129, falls:

Anhang: M 6800-Befehlsvorrat

		Adres- sierung	Arithmetische/Boolesche Operation, Beschreibung
		Accu Immediate Direct Indexed Indirect Relative Relative	
ORGANISATORISCHE BEFEHLE			
Bezeichnung			
CLC	<u>C</u> lear <u>C</u> arry	x	Carry := 0
SEC	<u>S</u> ET <u>C</u> arry	x	Carry := 1
CLI	<u>C</u> lear <u>I</u> nterrupt Mask	x	Interrupt Mask Bit := 0
SEI	<u>S</u> ET <u>I</u> nterrupt Mask	x	Interrupt Mask Bit := 1
CLV	<u>C</u> lear <u>O</u> verflow	x	Overflow := 0
SEV	<u>S</u> ET <u>O</u> verflow	x	Overflow := 1
WAI	<u>W</u> AIT for Interrupt	x	Rettet den Prozessor-Status auf den Stack; erniedrigt den SP um 7; interne Warteschleife des Prozessors
RTI	<u>R</u> eturn from <u>I</u> nterrupt	x	Ruecksprung aus einer Interrupt-Service-Routine; restau-riert den Prozessor-Status; erhoeht den SP um 7
SWI	<u>S</u> oftWare <u>I</u> nterrupt	x	Spezialbefehl - vgl. Prozessor-Handbuch
NOP	<u>N</u> o <u>O</u> peration	-	Leer-Befehl, inkrementiert nur den PC

Liste der verwendeten Abkürzungen

Zu jeder Abkürzung ist jeweils die ausgeschriebene Form angegeben und gegebenenfalls die Seitenzahl der Seite, auf der eine Erklärung zu finden ist.

A	Akkumulator A	95	FPLA	Field Programmable Logic Array	30
ACIA	Asynchronous Communication Interface Adapter	109	G	Carry Generate	37
ACK	Acknowledge	194	H	(1) High (Signalpegel) (2) Half-Carry-Bit des Status-Registers	- 97
AEQUIV	logische Aequivalenz	-	I	Interrupt-Mask-Bit des Status-Registers	97
ALU	Arithmetic Logic Unit	36	IC	Integrated Circuit	-
ASCII	American Standard Code for Information Interchange	108	ICE	In-Circuit Emulation	149
B	Akkumulator B	96	IDA	Input Data Available	115
BA	Bus Available	100	IFL	Input-FIFO Loaded	115
BC	Block Carry	39	I ² L	vergleiche /4/	-
BCD	Binary Coded Decimal	-	IM	Interrupt Mask Bit des Status-Registers	75
BG	Block Carry Generate	39	IR	Input Ready	24
BP	Block Carry Propagate	39	IRQ	Interrupt Request	57
C	Carry	-	L	Low (Signalpegel)	-
CAS	Column Address Strobe	17	LAR	Lade Adressregister	46
CCR	Condition Code Register	97	LBR	Lade Befehlsregister	46
CG	Carry Generate	37	LI	Left In	-
CI	Carry-In	-	LIFO	Last-In-First-Out	136
CLA	Carry Look Ahead	39	LO	Left Out	-
CP	(1) Clock Pulse (2) Carry Propagate	- 37	LSB	Least Significant Bit/Byte	-
CPU	Central Processing Unit	46	LSI	Large Scale Integration	-
CRC	Cyclic Redundancy Check	191	M	Mode Control	43
CS	Chip Select	-	MC	Memory Clock	17
CTS	Clear-to-Send	110	MPL	Hoehere Programmiersprache fuer Mikroprozessor M 6800	130
DCD	Data Carrier Detect	110	MPU	Microprocessing Unit	95
DDRA	Data Direction Register A	105	MS	(1) Memory Select (2) Mittelwert Schmalere Strich	- 187
DDRB	Data Direction Register B	105	MSB	Most Significant Bit/Byte	-
DIE	Data-In Enable	17	MUX	Multiplexer	-
DMA	Direct Memory Access	58	N	Negativ-Bit des Status-Registers	65
DOE	Data-Out Enable	17	NMI	Non-Maskable Interrupt	100
E/A	Ein/Ausgabe	-	NMOS	n-Kanal-MOS-Technik	-
ECA	Enable Column Address	18	OFE	Output-FIFO Empty	115
ECL	Emitter Coupled Logic	-	OR	Output Ready	25
EPROM	Erasable Programmable Read-Only Memory	30	ORA	Output Register A	105
EXOR	Exklusiv-ODER	-	ORB	Output Register B	105
ERA	Enable Row Address	18			
FE	Framing Error	112			
FF	Flipflop	-			
FIFO	First-In-First-Out	20			
FILO	First-In-Last-Out	136			

OV RN	Overrun-Bit des Status-Registers	112	V.24/V.28	Serielle Schnittstelle	113
P	Carry Propagate	37	VA	Valid Address	83
PC	Program Counter	48	VMA	Valid Memory Address	99
PE	Parity Error	113	WE	Write Enable	12
PJA	Peripheral Interface Adapter	103	X	Index Register	95
PLA	Programmable Logic Array	30	Z	Zero-Bit des Status-Registers	65
PRA	Peripheral Register A	146			
PRB	Peripheral Register B	146			
PROM	Programmable Read-Only Memory	29			
RA	Refresh Active	19			
RALU	Register and Arithmetic Logic Unit	61			
RAM	Random Access Memory	9			
RAS	Row Address Strobe	17			
RDRF	Receive Data Register Full	112			
RE	RALU Enable	75			
RePROM	Reprogrammable ROM	30			
RG	Refresh Grant	18			
RI	Right In	-			
RO	Right Out	-			
ROM	Read-Only Memory	28			
RR	Refresh Request	18			
RS	Register Select	-			
RS 232	Serielle Schnittstelle	113			
RTS	Ready-to-Send	110			
R/W	Read/Write	-			
RxClock	Receive Clock	109			
RxDData	Receive Data	109			
S	(1) Select	43			
	(2) Strichbreite	187			
SCHW	Schwelle zwischen schmal und breit	187			
SI	Shift In	25			
SO	Shift Out	25			
SP	Stack Pointer	136			
TDRE	Transmit Data Register Empty	112			
Transceiver	Transmitter/Receiver	-			
TS	(1) 3-State, vgl./2/	-			
	(2) Test Status	88			
TSC	3-State Control	99			
TTL	Transistor-Transistor Logic	-			
TTY	Teletype	113			
TxClock	Transmit Clock	109			
TxDData	Transmit Data	109			
USE	User's System Evaluator	149			
V	Overflow-Bit des Status-Registers	65			

Stichwortverzeichnis

3-State 11,53
Ablaufdiagramm 119
Ablaufsteuerung 33,48
ACIA 109ff
Adressierung, direkte 81,121
Adressierung, indizierte 80,
121
Adressierung, relative 79,122
Adressierung, symbolische
123ff
Adressierung, unmittelbare
80,120
Adressierungsarten 79,119
Adressregister 48,81
Akkumulator 95
ALGOL 130
ALU 36ff,42ff,64ff
ALU-Funktionen 68ff
Anwenderprogramm 54
Arithmetic Logic Unit 36
Arithmetik 133ff
Arithmetisch-logische Einheit
36
ASCII 108,192,193
Assembler 55
Assemblersprache 55,123ff
Assemblierprogramm 55,123ff
Asynchronous Communications
Interface Adapter 109ff
BASIC 55,130,140
Befehlscode 81
Befehlsregister 48,81ff
Befehlsstruktur 78
Befehlsvorrat 119
Befehlszaehler 48,50
Bit-Slice-Prozessor 7,61ff,
73ff
Breakpoint 142
Burst Mode Refresh 16
Bus 46,51,94
Bus-Extender 101
Carry Generate 37,44,64
Carry Propagate 37,44,64
Carry-Bit 75
Carry-In 44,64
Carry-Look-Ahead 36,39,40
Carry-Look-Ahead-Generator
61,75
Carry-Out 44,64
Central Processing Unit 46
Checkwort 174,190ff
Compiler 55,119,128
CPU 46ff,77ff
Cross-Software 55
Cycle Stealing Refresh 16
Datenblock 116
Datenrichtungsregister 105
Debug 140
Direct Memory Access 59
Displacement 80,121,122
DMA 58ff
Don't-Care-Eingaenge 34
E/A-Bauelement 102ff
E/A-Baustein, paralleler
103ff
E/A-Baustein, serieller 108ff
E/A-Befehl 57
Ein/Ausgabe 57
Entwicklungssystem 139ff
EPROM 30
Festwertspeicher 8
Fetch-Phase 50,81,88
Field Programmable Logic
Array 30
FIFO 22ff,114ff
First-In-First-Out - Speicher
22
Flussdiagramm 119
FORTRAN 55,130,140
FPLA 30ff
Framing Error 112
Gast-Rechner 55
Handshake 18
Hexadezimal-Darstellung 122
Host Computer 55
In-Circuit-Emulation 149
Indexregister 80,95
Initialisierungsprogramm 99,
106,143
Interpreter 55
Interrupt 84,135ff
Interrupt Mask Bit 75,82
Interrupt Request 57,100
Interrupt Service Routine 57,
143
IRQ 57
Kernspeicher 57
Large Scale Integration 7
Leitwerk 46,48,61,83ff
Lesestift 166
Linienstrom-Nahtstelle 113
Logic Analyzer 157ff
Logic State Analyzer 157ff
Makrobefehl 79,87
Makroprogramm 51ff
Map-Darstellung 159
Maschinencode 55,122
Masken-ROM 30
Microprocessing Unit 95
Mikrobefehl 83ff
Mikroprogramm 51ff
Mikroprogramm-Adress-
Register 85

Mikroprogramm-ROM 83ff
Mikroprogramme, lineare 83
Mikroprogrammierung 61
Mikroprogrammspeicher 28
Mikroprogrammzaehler 83ff
Mikrorechner 47,51
Mnemotechnischer Code 128
MPL 130
Multiplikation 134
Negative-Bit 65
Object-Code 54,128
Offset 81,122
One-Chip-Mikroprozessor 7,47,
94ff
Operand 78
Operationscode 78ff,120ff
Overflow 65,97
Paritaet 113
Paritaetsbit 108,111
PC 48,50
Peripheral Interface Adapter
103
PIA 103ff,116
PL/1 55,130,140
PLA 30ff
Polling 137,143
Produkt-Term 35
Program Counter 48
Programm 53
Programm-Unterbrechungen
135ff
Programmable Logic Array 30
Programmerstellung 118ff
Programmiersprache 53ff
Programmiersprache, hoehere
55,128ff
Programmierungstechnik 132ff
Programmzaehler 48,77,95
PROM 29,32,56
PROM-Programmiergeraet 29
Prozessorelement 47
Pruefbyte 174,190ff
Quell-Sprache 55
RALU 61,73
RAM 9ff,57
RAM, dynamisches 13ff
RAM, statisches 10ff
Random Access Memory 9
Re-Assemblierung 159,160
RePROM 30,56
Read-Only Memory 28
Rechenwerk 46,48,73ff
Refresh 10,13ff
Refresh Grant 18
Refresh Request 18
Refresh-Zyklus 16,18
Registerfeld 78ff
Reset 83ff,99
Ripple Carry Adder 36
ROM 28ff,32,56
RS 232 - Nahtstelle 113
Sammelschiene 46
Schaltnetz 28
Schaltwerk 33
Schreib/Lese-Speicher 8ff
Simulationsprogramm 150ff
Slice-Mikroprozessor 47
Slice-Prozessorelement 61ff
Software, residente 55
Speicher, dynamische 9ff
Speicher, statische 9ff
Speicherzugriff, direkter
58ff
Sprache, maschinenorientierte
54
Sprache,
maschinenunabhaengige 54
Sprache, problemorientierte
54
Sprungbefehl 65,84,87
Stack Pointer 95,136
Stapelbereich 136,137
Stapelspeicher 95,100
Start-Adress-ROM 83ff
State Transition Graph 33,34
Status 65
Status-Register 48,75,81ff,
95ff,112
Steuerbus 100
Strich-Code 166ff
Takt 98
Testlauf 146ff
Three-State 11,53
Trace 142
TTY-Nahtstelle 113
Uebertrag, durchlaufender 36
Uebertragsverarbeitung,
schnelle 36,39,41
Unterbrechung 57
Zentraleinheit 46,48ff,77ff
Zero-Bit 65,97
Zyluszeit 98

Teubner Studienbücher

Informatik

Berstel: **Transductions and Context-Free Languages**

278 Seiten. DM 38,- (LAMM)

Dal Cin: **Fehlertolerante Systeme**

206 Seiten. DM 23,80 (LAMM)

Ehrig et al.: **Universal Theory of Automata**

A Categorical Approach. 240 Seiten. DM 24,80

Giloi: **Principles of Continuous System Simulation**

Analog, Digital and Hybrid Simulation in a Computer Science Perspective

172 Seiten. DM 25,80 (LAMM)

Hotz: **Informatik: Rechenanlagen**

Struktur und Entwurf. 136 Seiten. DM 17,80 (LAMM)

Kandzia/Langmaack: **Informatik: Programmierung**

234 Seiten. DM 24,80 (LAMM)

Kupka/Wilsing: **Dialogsprachen**

168 Seiten. DM 19,80 (LAMM)

Maurer: **Datenstrukturen und Programmierverfahren**

222 Seiten. DM 26,80 (LAMM)

Mehlhorn: **Effiziente Algorithmen**

240 Seiten. DM 24,80 (LAMM)

Oberschelp/Wille: **Mathematischer Einführungskurs für Informatiker**

Diskrete Strukturen. 236 Seiten. DM 22,80 (LAMM)

Paul: **Komplexitätstheorie**

247 Seiten. DM 25,80 (LAMM)

Richter: **Betriebssysteme**

Eine Einführung. 152 Seiten. DM 22,80 (LAMM)

Richter: **Logikkalküle**

232 Seiten. DM 24,80 (LAMM)

Schlageter/Stucky: **Datenbanksysteme: Konzepte und Modelle**

261 Seiten. DM 24,80 (LAMM)

Schnorr: **Rekursive Funktionen und ihre Komplexität**

191 Seiten. DM 25,80 (LAMM)

Spaniol: **Arithmetik in Rechenanlagen**

Logik und Entwurf. 208 Seiten. DM 24,80 (LAMM)

Vollmar: **Algorithmen in Zellularautomaten**

Eine Einführung. 192 Seiten. DM 21,80 (LAMM)

Wirth: **Algorithmen und Datenstrukturen**

2. Aufl. 376 Seiten. DM 28,80 (LAMM)

Wirth: **Compilerbau**

Eine Einführung. 2. Aufl. 94 Seiten. DM 16,80 (LAMM)

Wirth: **Systematisches Programmieren**

Eine Einführung. 3. Aufl. 160 Seiten. DM 22,80 (LAMM)

Preisänderungen vorbehalten