

SPEEDY 1050



Speedy 1050-Super Speed

ANWENDER- HANDBUCH

S P E E D Y 1 0 5 0

Ein Anwender Handbuch
und ROM Listing

(c)1986 Compy-Shop
Version 1.0 vom 16.10.86

Autoren:

Peter Bee, Erwin Reuß und Reinhard Wilde

Copyright Notiz:

ATARI, ATARI 1050, ATARI 800XL, ATARI 130XE und sind
eingetragene Warenzeichen der Firma ATARI CORP. DEUTSCHLAND.

MAC/65, ACTION!, BASIC XL, BASIC XE und DOS XL sind
eingetragene Warenzeichen der Firma OPTIMIZED SYSTEM
SOFTWARE INC., SAN JOSE, CA, USA.

"Die Informationen im vorliegenden Handbuch werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht. Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt. Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Die Autoren und Herausgeber dieses Handbuches können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen. Für Verbesserungsvorschläge und Hinweise auf Fehler sind die Autoren dankbar.

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien sowie der Übersetzung in fremde Sprachen. Die gewerbliche Nutzung der in diesem Handbuch gezeigten Modelle, Programme und Arbeiten ist nur mit der ausdrücklichen Genehmigung der Autoren erlaubt.

SPEEDY 1050 ANWENDER HANDBUCH
(c) 1986 Compy-Shop Ohg, 4330 Mülheim Ruhr
Alle Rechte vorbehalten

Druck: DER DRUCKER, BOCHUM
Printed in Germany

INHALTSVERZEICHNIS:

Vorwort	1
Der Aufbau der SPEEDY 1050 Platine	3
Die Grundversion	3
Die Erweiterte Version	3
Die Funktionsweise der SPEEDY 1050	4
Die Datenübertragung zum Computer	6
Programmieren der SPEEDY 1050	7
Speicheraufteilung SPEEDY 1050	8
Erklärung zur Speicherbelegung	9
Die Einsprungadressen	11

LABEL	SEITE
RESET	12
RESET2	12
BEREIT	13
MOTON	13
TSTMON	14
MOTOFF	14
SDELAY	14
SDRDDP	15
XWAIT	15
X2WAIT	15
TRACKO	15
TRADJA	16
TRADJ	16
TRVR	16
CONRES	17
CONRE2	17
WREADY	17
RD128	17
RD256	17
RDBTS	18
RDSFOL	18
RDTRA	19
RDTRAV	19
TSTWR	20
TSTDAT	21
SD128B	21
SD256B	21
SDBTS	21

LABEL	SEITE
SEND41	22
SEND43	22
SEND45	22
SEND4E	22
RDSECT	23
RDSECT1	23
WRSECT	24
WRSECT1	24
TSTWRP	24
VERSEC	25
VERSEC1	25
STELL	25
QUITT	26
RDHEAD	26
RDHD1	26
CALCTS	27
SETBUF	27
SETBUF2	27
SEXBUF	28
SETRWL	28
COPSLT	28
BELL1	29
CLRDSP	29
TRAANZ	29
DEZOUT	30
HEXOUT	30
DENDSP	30
SETTIM	31
CLRTRA	31
CLRDSK	31
RAMTST	32
ROMTST	33
SPEEDT	33

DIE BEFEHLE DER SPEEDY 1050

34

BEFEHL	SEITE
\$52	35
\$50	35
\$57	35
\$53	36
\$21	37
\$22	37
\$20	38
\$3F	38
\$4E	39
\$4F	39
\$51	40
\$44	40
\$4B	41
\$4C	41
\$4D	42
\$41	42
\$68	43
\$69	43
\$60	44
\$62	44

ANHANG A - DEMONSTRATIONS-PROGRAMME

Demonstration 1 für Kommando \$52
Demonstration 2 für Kommando \$52
Demonstration 1 für Kommando \$50
Demonstration 2 für Kommando \$50
Demonstration für Kommando \$3F
Demonstration für Kommando \$44
Demonstration für Kommando \$4B
Demonstration für Kommando \$4C
Demonstration für Kommando \$4D
Demonstration 1 für Kommando \$41
Demonstration 2 für Kommando \$41
Demonstration für Kommando \$60
Demonstration für Kommando \$62

Lesen der SIO Routine vom Laufwerk

ANHANG B - ERWEITERUNG DER SPEEDY N AUF SPEEDY T
EINBAUANLEITUNG
BESTÜCKUNGSPLAN

ANHANG C - DAS ROM-LISTING DER SPEEDY

VORWORT

Wenn Sie dieses Handbuch gekauft haben, sind Sie wahrscheinlich schon Besitzer einer SPEEDY 1050, und kennen die Vorteile, die Ihnen dieser Hardwarezusatz für Ihr Laufwerk bietet, und wollen jetzt etwas mehr über die Möglichkeit der Programmierung der SPEEDY 1050 erfahren.

Genau das wollen wir Ihnen mit diesem ANWENDER HANDBUCH ermöglichen: Die Programmierung Ihrer SPEEDY 1050.

Und genau hierfür haben wir dieses Handbuch geschrieben. Sie sollen die Möglichkeit bekommen, die speziellen Fähigkeiten der SPEEDY 1050 für Ihre eigenen Programme zu nutzen. So können Sie zum Beispiel sehr leicht die SUPER SPEED Routine der SPEEDY 1050 in Ihre Programme einbauen, oder die Fähigkeit der SPEEDY, eine Diskette zu formatieren, ohne daß Sie dazu den Computer oder ein DOS benutzen müssen. Oder Sie können sich selber ein Kopierprogramm schreiben, mit dem Sie Ihre Originaldisketten kopieren können.

Sie werden dazu in diesem ANWENDER HANDBUCH die genaue Dokumentation des ROM-Listings, sowie eine ausführliche Dokumentation der Einsprungsadressen und natürlich Demoprogramme finden. Wir glauben, das wir dieses ANWENDER HANDBUCH leicht verständlich geschrieben haben.

Wir wenden uns mit diesem ANWENDER HANDBUCH an die Programmierer, die wissen, wie Sie einen Floppy-Controller vom Typ 2797 oder 2793 programmieren müssen. Es würde zu weit führen, die Programmierung dieses Floppy-Controllers in diesem Handbuch zu erklären. Wir möchten in diesem Zusammenhang auch auf die ausgezeichnete Dokumentation des Herstellers dieses Bausteins hinweisen, die Sie in jedem guten Zubehörhandel bekommen können.

Hier also noch einmal: Dieses ANWENDER HANDBUCH ist nicht gedacht für den Anfänger in der Maschinensprach Programmierung! Es ist speziell geschrieben worden für den fortgeschrittenen Programmierer.

Wichtig für einen Programmierer ist auch noch der folgende Hinweis: Sie brauchen an uns keine weitere Lizenzgebühr zu zahlen, wenn Sie Software für die SPEEDY 1050 schreiben, oder Routinen aus der SPEEDY 1050 innerhalb Ihrer Software für die SPEEDY 1050 benutzen. Sie können also Ihre Programme Kommerziell vermarkten. Wir möchten Sie nur bitten, uns eine Kopie Ihres Programmes zu kommen zu lassen.

Und nun viel Spaß bei der Programmierung Ihrer SPEEDY 1050.

Ihr COMPY-SHOP TEAM

DER AUFBAU DER SPEEDY 1050 PLATINE

Es gibt zwei verschiedene Versionen der SPEEDY 1050. Die Erste ist die Grundaufführung. Die zweite Version der SPEEDY 1050 ist eine erweiterte Ausführung mit Trackanzeige und einem akustischen Fehlermelder, dem Summer. Technisch, und auf die Laufwerksleistungen bezogen, sind beide Versionen identisch.

DER AUFBAU DER GRUNDVERSION

Die Grundversion besteht aus der Platine, einem 8K RAM-IC, einem 8K EProm mit dem Betriebssystem, dem Mikroprozessor 65C02 und diversen Kodier-IC's.

DER AUFBAU DER ERWEITERTEN VERSION

Zusätzlich zu den Bauteilen der Grundversion kommen bei der erweiterten Version noch die Bauteile für den Summer und die Trackanzeige hinzu.

Die Grundversion lässt sich leicht durch einen entsprechenden Bausatz mit Trackanzeige und Summer nachrüsten.

DIE FUNKTIONSWEISE DER SPEEDY 1050

Ein normales ATARI 1050 Laufwerk besitzt einen RAM Buffer von 256 Bytes Größe. Diesen RAM-Bereich müssen sich Datenspeicher und Mikroprozessor teilen. Da für einen Sektor ja bereits 128 Bytes gebraucht werden, hat der Mikroprozessor nicht mehr viel Platz zum arbeiten. Aus diesem Grund kann bei einem ATARI 1050 Laufwerk pro Umdrehung der Diskette jeweils nur ein Sektor eingelesen und zum Computer gesendet werden. In einem Track liegen, bei SINGLE DENSITY, 18 Sektoren. Die Diskette rotiert mit ca. 5 Umdrehungen pro Sekunde (288 Umdrehungen/Minute).

Das ergibt, für einen kompletten Track, eine Ladezeit von ca. 3,6 Sekunden. Das ist die Zeit, die das ATARI 1050 Laufwerk benötigt, um einen kompletten Track einzulesen.

Die SPEEDY 1050 besitzt einen 8K-Byte großen RAM Speicher. Dieser arbeitet als DATENSPOOLER. Pro Umdrehung der Diskette kann nun ein kompletter Track in den RAM-Buffer eingelesen werden. Das ergibt eine Ladezeit für einen Track von 0,2 Sekunden. Die Geschwindigkeit ist also um den Faktor 18 erhöht worden. Diese Geschwindigkeit ist die normale Arbeitsgeschwindigkeit eines SPEEDY 1050 Laufwerkes.

Durch die Zwischenspeicherung der Daten kann nun auch die Übertragungsgeschwindigkeit zum Computer erhöht werden. Durch diesen grossen RAM Buffer können nun auch mehr als 128 Bytes in einem Sektor geschrieben werden. Es wird also echte DOUBLE DENSITY, 256 Bytes per Sektor, möglich. Auch bei dieser Speicherdichte, von nun 176K Byte auf einer Diskettenseite, arbeitet das SPEEDY 1050 Laufwerk mit der hohen Geschwindigkeit.

Auf der SPEEDY 1050 Platine befindet sich neben dem RAM aber auch noch ein neuer Mikroprozessor. Dieser Mikroprozessor ist der 65C02. Gegenüber dem 6507 einer normalen ATARI 1050 bietet der 65C02 zwei grosse Vorteile.

Erstens kann der 65C02 bis zu 64K adressieren, der 6507 nur 8K-Bytes, und zweitens besitzt der 65C02 einen erweiterten Befehlssatz mit zusätzlichen, sehr nützlichen Befehlen. Aufgrund dieser zusätzlichen Befehle konnte das Betriebssystem der SPEEDY 1050 kurz gehalten werden, und die Geschwindigkeit der Programmausführung wird durch die geschickte Ausnutzung des erweiterten Befehlssatzes gesteigert.

DIE DATENÜBERTRAGUNG ZUM COMPUTER

Bei einem normalen ATARI 1050 Laufwerk wird, wie wir schon erwähnt haben, pro Diskettenumdrehung ein einzelner Sektor eingelesen und dann sofort an den Computer weitergegeben.

Diese Methode ist sehr Zeitraubend. Bei einem SPEEDY 1050 Laufwerk wird, bei einer Diskettenumdrehung, ein kompletter Track in den RAM Buffer eingelesen. Dadurch hat der Computer jederzeit Zugriff auf alle Sektoren, die sich in diesem Track befinden. Durch diese Zwischenspeicherung der Daten ist, im Normalmodus, eine Übertragung der Daten ohne Pause (bedingt durch die Ladezeit zwischen den Sektoren) möglich.

Das bedeutet, daß im Normalmodus die Lesegeschwindigkeit im Laufwerk mit maximaler Geschwindigkeit läuft, die Datenübertragung zum Computer aber mit normaler Geschwindigkeit geschieht. Aber aufgrund der wegfallenden Pausen verkürzt sich die Ladezeit um ca. 50%.

Beim aktivieren der speziellen SPEEDY 1050 Geschwindigkeit, der SUPER SPEED, wird die Datenübertragung vom Laufwerk zum Computer auf das Maximum gesetzt. Bei einem normalen ATARI 1050 Laufwerk geschieht die Datenübertragung zum Laufwerk mit 19.200 Baud. Beim aktivieren der SUPER SPEED erhöht sich diese Baudrate auf das 4 fache.

PROGRAMMIEREN DER SPEEDY 1050

Aufgrund der besonderen Fähigkeiten der SPEEDY 1050 haben Sie nun auch die Möglichkeit das Laufwerk individuell zu programmieren.

Die Möglichkeiten, die sich Ihnen damit eröffnen, sind fast unerschöpflich. So können Sie zum Beispiel Diskettenformate nach Ihrem eigenen Bedarf erstellen, einen eigenen Kopierschutz erzeugen oder einen fremden kopieren.

Auch die SUPER SPEED können Sie sehr leicht für Ihre eigenen Programme nutzen, wie Sie anhand eines Demoprogrammes sehen werden.

SPEICHERAUFTeilUNG SPEEDY 1050

Nachfolgend finden Sie ein Blockschaltbild mit der genauen Speicherbelegung der SPEEDY 1050.

\$FFFF	
ROM - 8 K B Y T E	
\$E000	
\$DFFF	
U N B E N U T Z T	
\$A000	
\$9FFF	
R A M - 8 K B Y T E	
\$8000	
\$7FFF	
U N B E N U T Z T	
\$0404	
\$0403	
C O N T R O L L E R 2 7 9 3 / 9 7	
\$0400	
\$03FF	
U N B E N U T Z T	
\$0300	
\$02FF	
P O R T (I / O + T I M E R)	
\$0280	
\$027F	
U N B E N U T Z T	
\$0200	
\$01FF	
S T A C K	
\$0100	
\$00FF	
Z E R O P A G E	
\$0000	

ERKLÄRUNG ZUR SPEICHERBELEGUNG

\$E000 - \$FFFF - BETRIEBSSYSTEM

Hier liegt das Betriebssystem Ihrer SPEEDY 1050. Änderungen können Sie hier nicht vornehmen. Ein genaues Listing dieses Betriebssystems finden Sie im Anhang dieses Handbuches.

\$8000 - \$9FFF - ARBEITSSPEICHER

Der 8K Ram Block ist in 5 Bereiche unterteilt:

\$9F80 - \$9FFF - Hier liegen Einsprung und Rücksprungvektoren für die Bereitschaftsroutine des Betriebssystems. Außerdem können Sie hier Erweiterungen der RESET Routine vornehmen.

\$9F00 - \$9F7F - Die normale und erweiterte Kommandotabelle und die entsprechenden Einsprünge sind hier zu finden. Über das Kommando \$41 können Sie diese Tabelle beliebig verändern. Diesen Befehl (und andere) haben wir Ihnen bereits im SPEEDY 1050 Handbuch erklärt, Sie werden Sie aber auch noch einmal, etwas später, in diesem ANWENDER HANDBUCH finden.

\$9E00 - \$9EFF - Der EXTENDED BUFFER dient zur Zwischenspeicherung von Sektordaten bei FAST WRITE oder beim SLOW MODE diverser Laufwerksfunktionen.

\$8C00 - \$9DFF - In diesem Bereich liegt der Trackbuffer. Hier werden bei FAST WRITE oder FAST READ erst alle Sektordaten eines Tracks zwischengespeichert. Schalten Sie die SPEEDY 1050 zum Beispiel mit Hilfe der Menu-Diskette (Menupunkt SLOW MODE CONTROL) für READ SECTOR und WRITE SECTOR in den SLOW MODUS, wird dieser Speicherbereich nicht mehr vom Betriebssystem benutzt. So können Sie auch hier eigene Routinen ablegen.

\$8000 - \$8BFF - Freier Speicherbereich, der dem Programmierer ständig zur Verfügung steht, also wo Sie Ihre eigenen Programme ablegen können!

\$0400 - \$0403 - Hier liegen die Register des Disk-Controllers
2793/97:

- \$0400 : Lesen=Statustregister, Schreiben=Command-
register
- \$0401 : Lesen + Schreiben=Trackregister
- \$0402 : Lesen + Schreiben=Sektorregister
- \$0403 : Lesen + Schreiben=Datenregister

\$0280 - \$02FF - Hier befinden sich die Register des Port IC's
6532 (RIOT)

Die gebräuchlichsten Register:

- \$0280 : Port A Datenregister
- \$0281 : Port A Richtungsregister
- \$0282 : Port B Datenregister
- \$0283 : Port B Richtungsregister
- \$0296 : Timer lesen/schreiben, Timer IRQ
abschalten
- \$029F : Timer mit Teilerverhältnis 1:1K lesen/
schreiben, Timer IRQ einschalten

\$0000 - \$00FF - Die Zeropage und die Page 1 überlagern sich.
Das heißt, Speicherstelle \$0000 entspricht der
Speicherstelle \$0100, Speicherstelle \$0001 ent-
spricht der Speicherstelle \$0101, usw. In der
Zeropage stehen dem Benutzer die Speicherstellen
\$0090 bis \$00CF zur freien Verfügung.

DIE EINSPRUNGADRESSEN

Nachdem wir Ihnen nun die Speicherbelegung der SPEEDY 1050 erklärt haben, erfahren Sie jetzt etwas über die Einsprungadressen. Wichtig ist dabei eines zu wissen: Wir haben am Ende des ROM-Bereiches ab \$FF00 eine Jumptable eingerichtet. Wenn Sie nun eine Funktion innerhalb der SPEEDY ausführen wollen, brauchen Sie nur eine, oder mehrere Adressen dieser Jumptable anzuspringen.

Welche Versionen es in Zukunft auch von der SPEEDY 1050 Software geben wird, diese Jumptable wird immer an der gleichen Stelle und in der gleichen Reihenfolge erhalten bleiben. Somit ist gewährleistet, das alle Software, die jetzt für die SPEEDY 1050 geschrieben wird, auch auf den zukünftigen Versionen laufen wird. (Wenn die internen Routinen über diese Tabelle angesprungen werden!)

Hier die Beschreibung der einzelnen Jump Vektoren:

RESET - \$FF00 - DRIVE KALTSTART

Port A und Port B (6532) werden initialisiert, der komplette RAM Bereich gelöscht und der Disk-Controller getestet. Sollte ein Fehler beim Testen auftreten, erfolgt ein Sprung nach "SYSERO", wo 2 mal "BELL" ausgegeben wird, und anschliessend folgt ein Sprung nach "RESET2".

RESET2 - \$FF03 - DRIVE WARMSTART

Die System-Variablen werden neu gesetzt, das Laufwerk in den "SINGLE-DENSITY-MODUS" gebracht, der SchreibLesekopf auf Track 0 justiert und jede Controller Tätigkeit gestopt. Zum zurücksetzen des Laufwerkes sollte diese Routine angesprungen werden, da bei "\$FF00 - RESET" alle im RAM befindlichen Daten und Programme gelöscht werden.

BEREIT - \$FF06 - BEREITSCHAFTSRoutine

Zuerst wird der "SLOW" Schalter abgefragt und das Laufwerk gegebenenfalls in den "SLOW" Modus geschaltet. Sollten sich noch zu schreibende Sektoren im RAM befinden, wird direkt nach "TSTCO" verzweigt, damit, falls die Laufwerksklappe geöffnet wird, der Motor nicht ausgeschaltet wird. Wurde nicht verzweigt, wird die Laufwerksklappe mit der letzten Stellung verglichen. Sollte die Klappe geöffnet worden sein, so werden der Antriebsmotor und der Steppermotor ausgeschaltet und der Controller-Status nach "CONST" kopiert. Ist die Klappe geschlossen worden, wird die DENSITY neu festgestellt, das Laufwerk entsprechend eingestellt und die Sektor-Folge neu gelesen. In der "TESTCO" Routine wird die "COMMAND" Leitung vom Computer abgefragt. Sollte diese "gesetzt" werden, erfolgt ein Sprung nach "RDINF", wo das Kommando vom Computer eingelesen wird. Ist die "COMMAND"-Leitung nicht gesetzt wird der Motor-Timer heraufgezählt. Sollte dieser den Wert \$9800 (16 Bit) erreichen, wird "TSTWR" aufgerufen um alle Sektoren zu schreiben, die sich noch im RAM befinden.

MOTON - \$FF09 - MOTOR ZWINGEND EINSCHALTEN

Der Antriebsmotor wird ohne Rücksicht auf die Stellung der Laufwerksklappe eingeschaltet.

TSTMON - \$FFOC - MOTOR EINSCHALTEN, WENN DIE LAUFWERKSKLAPPE
GESCHLOSSEN IST

Erst wird die Laufwerkklappe abgefragt. Ist diese geschlossen, wird der Antriebsmotor eingeschaltet und eine Zeitverzögerung durchgeführt, um den Motor auf Touren kommen zu lassen.

MOTOFF - \$FFOF - MOTOR AUSSCHALTEN

Der Antriebsmotor wird ausgeschaltet und das entsprechende Bit in "DRSTAT" gesetzt.

SDELAY - \$FF12 - MOTOR TIMER EINSTELLEN

Diese Routine wird nach einer Kommandoausführung abgearbeitet. Zuerst wird die Zeit gestellt, wie lange der Motor nach einer Kommandoausführung noch eingeschaltet bleiben soll, dann der Kommando-Buffer gelöscht. Danach erfolgt ein Rücksprung in die Motor-Timer-Routine innerhalb der Bereitschafts-Routine.

SDRDDP - \$FF15 - DRIVE DENSITY EINSTELLEN UND ANZEIGEN

Zuerst wird nach "DENDSP" gesprungen um die aktuelle DENSITY auf dem Display anzuzeigen. Danach wird, je nach Wert in "FORKEN" der Drive-Status, die Anzahl der Sektoren pro Track und die Anzahl der Bytes per Sektor gesetzt.

XWAIT - \$FF18 - WARTESCHLEIFE KURZ

Der Wert im X-Register gibt die Anzahl der Schleifendurchläufe an. Ein Schleifendurchlauf entspricht ca. 100 Taktzyklen (0.1 msec/100müsec).

X2WAIT - \$FF1B - LANGE WARTESCHLEIFE

Der Wert im X-Register gibt die Anzahl der Schleifendurchläufe an. Ein Schleifendurchlauf entspricht ca. 100.000 Taktzyklen (0.1 sec/100msec).

TRACKO - \$FF1E - TRACK 0 POSITIONIEREN

Der Disk-Controller wird gestoppt und der Schreib/Lesekopf so lange zurückgezogen, bis der Track-0-Sensor anspricht. Danach wird eine Zeitverzögerung zum ausschwingen der Kopfmechanik durch geführt.

TRADJA - \$FF21 - TRACK ANZEIGEN UND SCHREIB/LESEKOPF POSIT-
IONIEREN

Zuerst wird die Tracknummer angezeigt und der Controller gestoppt. Ist die Klappe geschlossen wird der Motor eingeschaltet und die Anzahl der Doppel-Steps errechnet, die durchgeführt werden müssen, um die gewünschte Kopfposition zu erreichen. Sollte der Trackwechsel über 40 Tracks gehen, wird 2*"BELL" ausgegeben und der Kopf auf Track 0 positioniert.

Nach erfolgreicher Schreib/Lesekopf Positionierung wird die Tracknummer in das Track-Register des Controllers kopiert und der Schreib/Lesekopf Mechanik Zeit zum ausschwingen gegeben.

TRADJ - \$FF24 - SCHREIB-LESE-KOPF POSITIONIEREN

Entspricht der \$FF21 Routine, mit dem Unterschied, daß die Tracknummer nur dann angezeigt wird, wenn ein Trackwechsel statt gefunden hat.

TRVR - \$FF27 - 1 STEP VORWÄRTS ODER RÜCKWÄRTS AUSFÜHREN

Im Y-Register ist die Kennung für Step vorwärts oder rückwärts (plus oder minus). Die Bitposition des Stepermotors wird entsprechend herauf- oder herabgezählt und das neue Bitmuster in Port B des Portbausteins geschrieben.

Anmerkung:

Für 1 Trackwechsel müssen immer 2 Steps erfolgen. Das heißt aber nicht, daß ohne bedenken 80 Tracks geschrieben oder gelesen werden können. Die beiden Steps sind verschieden lang!

CONRES - \$FF2A - DISK CONTROLLER STOPPEN

Dem Disk-Controller wird der Befehl gegeben, alle laufenden Aktionen zu stoppen. In "WREADY" wird gewartet, bis das der Controller den Befehl als "ausgeführt" meldet.

CONRE2 - \$FF2D - \$FF2A WIRD ZWEIMAL AUSGEFÜHRT

Beim Einsprung in diese Routine wird die Routine bei \$FF2A - DISK CONTROLLER STOPPEN zweimal ausgeführt.

WREADY - \$FF30 - AUF CONTROLLER "IN USE" FLAG=0 WARTEN

Bei dieser Routine wird darauf gewartet, daß der Controller meldet, daß der letzte Befehl abgeschlossen wurde.

RD128 - \$FF33 - 128 BYTES VOM COMPUTER NACH "EXBUF" HOLEN

Der Buffer wird auf "EXBUF" (Extended Buffer) und die IO-Länge auf 128 Bytes gesetzt.

RD256 - \$FF36 - 256 BYTES VOM COMPUTER NACH "EXBUF" HOLEN

Wie bei der vorhergehenden Routine wird der Buffer auf "EXBUF", aber die IO-Länge auf 256 Bytes gesetzt.

RDBTS - \$FF39 - ANZAHL BYTES IN DEN BUFFER HOLEN (X/Y REGISTER)

Im Accu stehen die Anzahl der Bytes, im X- und Y-Register die Low- und High-Adresse des Buffers, in die die Daten vom Computer abgelegt werden sollen. Der Timer wird gesetzt, um zu verhindern, das der Prozessor hängen bleiben kann. Es wird jeweils ein Byte über einen indirekten Jump-Vector vom Computer geholt (ausser bei HIGH SPEED) und in dem Buffer abgelegt. Anschließend wird die Checksumme heraufgezählt und geprüft, ob alle Bytes und Datenblöcke geholt wurden. Die Checksumme wird verglichen und die IO-Länge neu gesetzt.

RDSFOL - \$FF3C - NACH VERZÖGERUNG SEKTORFOLGE VOM AKTUELLEM TRACK LESEN

Die Verzögerungsschleife am Anfang der Routine verhindert, daß versucht wird, die HEADER schon zu lesen, wenn die Laufwerks-Klappe noch nicht vollständig geschlossen ist. Danach wird die Zeit gestellt, die der Controller zur Verfügung hat, um alle HEADER zu lesen. Ist ein HEADER eingelesen, wird die Track- und Sektor- Nummer auf Gültigkeit geprüft. Befindet sich die Sektornummer des gelesenen HEADERS bereits in der Sektorfolge (doppelter Sektor), so wird die Sektorfolge nicht mehr weiter gelesen und das Laufwerk in den Slow-Modus geschaltet. Stimmt die Anzahl der gelesenen Sektor-HEADER nicht mit der vorgegebenen (18 oder 26 Sektoren) überein, wird das Laufwerk ebenfalls in den Slow-Modus geschaltet.