

Rolf Keller

Der Floppy-Disk-Controller-Chip WD2797 von Western Digital übertrumpft andere Bausteine ähnlicher Art durch seine integrierten Schaltungsteile für Datenseparation und Schreib-Vorkompensation. Damit werden einige Bauelemente bei der äußeren Beschaltung eingespart, und man kommt mit der 'unberechenbaren' Analogtechnik kaum noch in Berührung. Der 2797 wird zum Beispiel im c't 86 und im c't 64000 eingesetzt.

Floppy-Disk-Controller WD2797

Der 2797 ist ein Mitglied der Bausteinfamilie 279x, die ihrerseits eine (softwarekompatible) Weiterentwicklung der verbreiteten Familie 179x ist. Behandelt wird hier nur der Typ 2797, die anderen Typen unterscheiden sich davon geringfügig. Zur Klärung dieser Unterschiede sollte man das Datenblatt zu Rate ziehen, wenn es um die letzten Feinheiten geht.

Alle Typen der Familie 279x können 8"- und 5,25"-Laufwerke (und dazu kompatible Mikrofloppy-Laufwerke) in Single und Double Density ansteuern. Dabei können alle verbreiteten IBM-ähnlichen Datenformate benutzt werden.

Beschaltung

Bild 1 zeigt die Pinbelegung des 2797, geordnet nach Funktionsgruppen. Die auf dem Chip integrierten Analogschaltungen für Datenseparation und Schreib-Vorkompensation benötigen einige Bauteile für den Abgleich (Bild 1 links unten). Bei der Aktivierung des Eingangs TEST kann der Abgleich ohne Umstände in der betriebsbereiten Schaltung erfolgen.

Die Laufwerksignale können in der Regel direkt mit den Pins der ICs verbunden werden, die die gleiche Signalbezeichnung tragen. Besondere Beachtung verdienen die Signale HLT (Head Load Timing = Kopf-

aufsetzzeit) und READY. HLT teilt dem 2797 mit, daß die Beruhigungszeit nach dem Aufsetzen des Kopfes abgelaufen ist. Die meisten Laufwerke stellen dieses Signal nicht zur Verfügung; man muß es dann mit einem Zeitglied erzeugen. Das Signal READY soll die Lese-/Schreib-Bereitschaft des Laufwerks anzeigen, das heißt Diskette eingelegt, Tür geschlossen und Spindrehzahl erreicht. Meistens liefern 5,25"-Laufwerke dieses Signal nicht. Vielfach sieht man Schaltungen, die die Signale HLT und READY nicht unter allen Bedingungen (zum Beispiel beim Umschalten zwischen zwei Laufwerken) korrekt erzeugen. Dieser Fehler bleibt oft unerkannt, da der 2797 einen Lese-/Schreib-Vorgang erst nach fünf Plattenumdrehungen abbricht. Diese Zeit ist gewöhnlich länger als die Hochlaufzeit des Motors und die Beruhigungszeit beim Kopfaufsetzen, so daß am Ende meist alles gutgeht. Tatsächlich funktioniert es daher sogar, wenn man HLT und READY fest auf High-Pegel legt, aber empfehlenswert ist dieses Verfahren nicht.

Der Eingang DDEN des ICs dient zum Umschalten auf Double-Density-Betrieb, der Eingang ENP zum Einschalten der Schreib-Vorkompensation. Mit dem Eingang 3/8 kann man zwischen 5,25"- und 8"-Laufwerken wählen, muß aber

gleichzeitig den Takt ϕ umschalten (1 MHz bei 5,25", 2 MHz bei 8"). MR ist der Reset-Eingang des Bausteins.

Die fünf adressierbaren Register des Bausteins werden wie folgt angesprochen:

CS	A1	A0	RE	WE
0	0	0	Status	Befehl
0	0	1	Spur	Spur
0	1	0	Sektor	Sektor
0	1	1	Daten	Daten

Der Ausgang DRQ (Data Request = Datenanforderung) kann bei reinem Polling-Betrieb unbenutzt bleiben, ansonsten wird er mit einem DMA-Baustein oder über eine Zusatzlogik mit dem WAIT-Eingang der CPU verbunden. Voll DMA-fähig ist der 2797 allerdings nicht: Ihm fehlt dazu ein DMA-Acknowledge-Eingang. Alternativ kann man den Ausgang DRQ auch als Interrupt-Ausgang (Daten-Interrupts) betrachten.

Der Ausgang INTRQ (Interrupt Request = Interrupt-Anforderung) kann beim Polling ebenso unbenutzt bleiben. Sein Hauptzweck ist es, beim DMA-Betrieb das Befehlsende zu melden. Im Fehlerfall würde der DMA-Baustein sonst 'ewig' auf Daten warten.

Leider 'kennt' der 2797 nur ein einziges Laufwerk, das heißt, er hat nur für dieses eine 'Drive'

Anschlußpins und interne Register. Um mehrere Laufwerke anschließen zu können, ist ein zusätzlicher Ausgabeport nötig, mit dem diese selektiert werden können. Mit den übrigen Portbits kann man dann gleich die Signale DDEN, ENP, 3/8 und am besten auch MR per Software steuern.

Programmierung

Die möglichen Befehle (Tabelle 1) sind in vier Typen eingeteilt. Der Befehl 'Interrupt' nimmt eine Sonderstellung ein. Befehle der Typen 1 bis 3 erteilt man einfach durch Schreiben des entsprechenden Codes in das Befehlsregister. Dies darf aber nur dann geschehen, wenn der 2797 gerade nichts zu tun hat (BUSY-Bit im Statusregister = 0). Die Spur- und Sektorregister müssen vor der Ausgabe des Befehls geladen werden, sofern die Werte für die Ausführung des Befehls benötigt werden. Falls man mit mehreren Laufwerken arbeitet, muß man beim Wechsel auf ein anderes Laufwerk Spur- und Sektorregister entsprechend umladen, da der 2797 diese Werte nur für ein Laufwerk speichern kann.

Die Befehlsbearbeitung beginnt mit dem Setzen des Statusregisters. Der Baustein setzt das BUSY-Bit, löscht die Fehlerbits und aktualisiert die übrigen Bits gemäß Tabelle 2 beziehungsweise 3. Dabei ist wichtig, daß die Statusbits 0 und 7

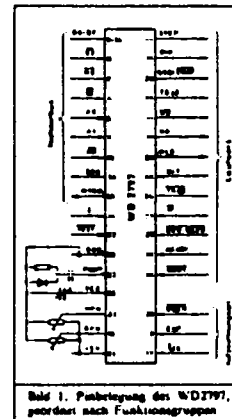
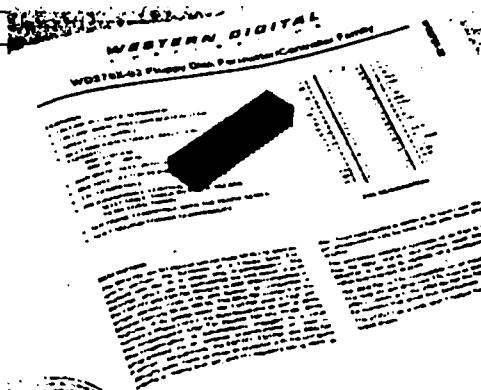


Bild 1. Pinbelegung des WD2797, geordnet nach Funktionsgruppen

Bit	Typ	Befehl	Aktion
7 6 5 4 3 2 1 0			
0 0 0 0 0 K P R1 R0	1	Auflösung Suchen	Positionieren auf Spur 0 Positionieren auf Spur 1: Datenregister = x, Spurregister = aktuelle Spur
0 0 0 1 K P R1 R0	1	Spur weiter	Positionieren auf Nachbarnach- spur: Richtung wie beim letzten Positionierbefehl
0 1 0 A K P R1 R0	1	Spur vor	Positionieren auf nächst- höhere Spur
0 1 1 A K P R1 R0	1	Spur zurück	Positionieren auf nächst- niedrigere Spur
1 0 0 M L W S 0	2	Sektor lesen	Sektor x der aktuellen Spur lesen; Sektorregister = x Kopf muß gemäß Spur- register positioniert sein
1 0 1 M L W S Ad	2	Sektor schreiben	Gesamte aktuelle Spur schreiben; Kopf muß gemäß Spur- register positioniert sein
1 1 0 0 0 W S 0	3	Adreßfeld lesen	Nächstes verarbeitendes Sektordrehschritt der aktuellen Spur lesen
1 1 1 0 0 W S 0	3	Spur lesen	Gesamte aktuelle Spur transparenz lesen
1 1 1 1 0 W S 0	3	Spur formatieren	Gesamte aktuelle Spur formatieren
1 1 0 1 12 12 11 10	4	Interrupt	siehe Text

K (Kopf) 1: Zu Beginn des Befehls den Kopf aufrufen 0: Zu Beginn des Befehls den Kopf abheben
P (Prüfung) 1: Nach Erreichen der Zeitdauer wird automatisch durch Lesen der Adreßfelder auf der Platte geprüft, ob es sich wirklich um die gewünschte Spur handelt, deren Nummer im Spurregister steht 0: Keine Prüfung
A (Abkürzungen) 1: Das Spurregister wird bei Positioniervorgängen automatisch aktualisiert (weitergezählt). 0: Spurregister bleibt unverändert
M (Mehrfachsektoren) 1: Beginnend mit dem vorgegebenen Sektor werden automatisch Sektoren mit den nächstfolgenden Nummern so lange bearbeitet, bis das Spurrenne erreicht ist. 0: Nur der dem vorgegebenen Sektor wird bearbeitet
W (Wartezeit) 1: Nach Anlieferung des Ausgangs HLD (Kopf aufzuheben) steht eine Wartezeit (15 ms bei 8", 30 ms bei 5,25") ab, bevor wiederum der Eingang HLT (Kopf ab aufsetzen?) angefragt wird. 0: Keine Wartezeit
S (Seitenwahl) 1: Seite 1 auswählen 0: Seite 0 auswählen
Ad (Adreßmarkierung) 1: Adreßmarkierung schreiben (selected data address mark) 0: Adreßmarkierung löschen (normal data address mark), dies ist bei den meisten Betriebssystemen der Normalfall
L (Länge eines Sektors) Kodierung im Adreßfeld: 00 01 10 11 L=0: 256 512 1024 128 L=1: 128 256 512 1024 L=1 bei IBM Format
R (Schrittweite für Spurwechsel) 0: 0 1: 12 ms 2: 25 ms 3: 30 ms Die angegebenen Zeiten gelten für 5,25", bei 8" halbiert sich die Zeiten
I (Unterbrechung) 10: Wechsel nicht READY → READY 11: Wechsel READY → nicht READY 12: Interrupt-impuls 13: schwebender Interrupt

Tabelle 1. Übersicht über die Befehle des FDC 2797

zu jedem Zeitpunkt dieselbe Bedeutung haben; bei den Bits 1 bis 6 ist das nicht der Fall.

Nach der Ausgabe eines Befehls muß man mit dem Auslesen des Statuswortes etwas warten (max. 56 µs): Der Chip 'denkt' nicht so schnell. Ansonsten kann aber das Statusregister zu jedem Zeitpunkt gelesen werden.

Ist die Ausführung eines Befehls beendet, so wird das BUSY-Bit wieder gelöscht. Trat während der Ausführung

des Befehls ein Fehler auf, ist das jeweilige Fehlerbit im Statusregister gesetzt.

Datentransfer

Weil der 2797 nur ein oder zwei Datenbytes intern speichern kann, muß er jedes Byte von beziehungsweise zur Diskette synchron zu deren Drehzahl abgenommen beziehungsweise geliefert bekommen. Dies geschieht durch Lesen/Schreiben des Datenregisters. Der 2797 fordert dazu pro Byte einmal

auf, indem er das DRQ-Bit (Data Request = Datenanforderung) im Statusregister (Tabelle 3) setzt. Der Ausgangspin DRQ zeigt diese Anforderung ebenfalls an. Die DRQ-Anforderung wird beim Zugriff auf das Datenregister automatisch wieder gelöscht. Die maximal zulässige Reaktionszeit beträgt beim Schreiben 11,3 µs (8" DD), 23,5 µs (8" SD/5,25" DD) oder 47 µs (5,25" SD). Um hier mithalten zu können, ist ein schnelles Programm (Polling unter Interrupt-Sperre) oder Zusatzhardware (zum Beispiel DMA mit Hilfe des DRQ-Pins) erforderlich. Wird die Zeit 'verpaßt', so ist Datenverlust die Folge. Der 2797 setzt dann das Fehlerbit 2 im Statusregister (Tabelle 3).

Interrupts

Nach der Ausführung jedes Befehls (mit Ausnahme eines 'Interrupt'-Befehls mit I3 = 0) aktiviert der 2797 seinen INTRQ-Ausgang. Dabei ist es gleichgültig, ob der Befehl erfolgreich ausgeführt wurde oder nicht. Eine Interrupt-Sperre ist nicht möglich.

Zusätzlich kann man noch mit Hilfe eines 'Interrupt'-Befehls weitere Interrupt-Bedingungen vorgeben (Tabelle 1). Der Befehl 'SD1' bewirkt zum Beispiel, daß dann ein Interrupt erzeugt wird, wenn der Pegel am READY-Eingang von high auf low wechselt. 'I3 = 1' löst sofort einen Interrupt aus. Damit kann man den Durchlauf der Interrupt-Routine erzwingen, wenn das programmtechnisch erforderlich ist.

Alle Interrupts werden quittiert durch Lesen des Statusregisters

oder durch Schreiben in das Befehlsregister. Ein durch 'I3 = 1' ausgelöster Interrupt kann jedoch nur durch den 'Interrupt'-Befehl 'SD0' quittiert werden.

Eine zweite Wirkung des 'Interrupt'-Befehls ist es, daß er die Ausführung jedes anderen Befehls unmittelbar abbricht. Dies ist möglich, weil der 'Interrupt'-Befehl der einzige Befehl ist, der ausgegeben werden darf, während ein anderer in Bearbeitung ist (BUSY = 1).

Um es ganz kompliziert zu machen, beeinflußt der 'Interrupt'-Befehl auch noch das Statusregister. War gerade ein anderer Befehl in Bearbeitung, so wird nur das BUSY-Bit auf null gesetzt, die übrigen Bits bleiben erhalten. War aber der 2797 gerade nicht 'beschäftigt', so wechselt das Statusregister in den Zustand, den es nach der Ausführung eines Befehls vom Typ 1 hat (Tabelle 2).

Verwendung der Befehle

Vor jedem Datenzugriff muß der Kopf des Laufwerks auf die gewünschte Spur gestellt werden. Dazu sind die Befehle des Typs 1 (Positionierbefehle) bestimmt. Für einen 'normalen' Disk-Treiber benötigt man davon nur die Befehle 'Nullstellung' und 'Suchen'.

Vor der Ausgabe des Befehls 'Suchen' muß das Spurregister die Nummer der Spur enthalten, auf der der Kopf gerade steht. Nach dem Einschalten und nach Fehlern ist das nicht unbedingt der Fall. Man gibt dann zunächst den Befehl 'Nullstellung'. Dabei werden so lange Schritte in Richtung Spur

0	Laufwerk nicht READY
1	Das Bit zeigt den invertierten Zustand des READY-Eingangs.
2	Diskette Schreibgeschützt
3	Das Bit zeigt den invertierten Zustand des WPRT-Eingangs.
4	Kopf aufgesetzt
5	Das Bit zeigt an, daß die Signale HLD und HLT auf high-Pegel liegen.
6	Positionierfehler
7	Das Bit zeigt an, daß nach der Ausführung eines Befehls mit gesetztem P-Bit der Kopf nicht auf der richtigen Spur steht.
8	Prüfsummenfehler
9	Das Bit zeigt an, daß bei der Prüfung nach einem Befehl mit gesetztem P-Bit ein Sektor Adreßfeld mit verlässlicher Information gefunden wurde.
10	Spur 00
11	Das Bit zeigt den invertierten Zustand des TR00-Eingangs.
12	Index Impuls
13	Das Bit zeigt den invertierten Zustand des IF-Eingangs.
14	Baustein BUSY
15	Das Bit zeigt an, daß der Baustein gerade mit der Ausführung eines Befehls beschäftigt ist.

Tabelle 2. Statusregister nach Befehlen vom Typ 1

7	Laufwerk nicht READY Das Bit zeigt den invertierten Zustand des READY-Eingangs.
6	Diskette schreibgeschützt Nur bei den Befehlen 'Sektor schreiben' und 'Spur formatieren', sonst 0.
5	Art der Adreßmarke Nur beim Befehl 'Sektor lesen', sonst 0. Das Bit zeigt an, welche Daten Adreßmarke gelesen wurde. 0: Adreßmarke SFB (normal data address mark); 1: Adreßmarke SFB (deleted data address mark).
4	Sektor nicht gefunden Nur bei den Befehlen 'Sektor lesen', 'Sektor schreiben' und 'Adreßfeld lesen', sonst 0. Das Bit zeigt an, daß der Baustein kein Sektor-Adreßfeld mit passender Sektor-, Spur- und Seitennummer einschließlich korrekter Prüfsumme gefunden hat.
3	Prüfsummen-Fehler Nur bei den Befehlen 'Sektor lesen', 'Sektor schreiben' und 'Adreßfeld lesen', sonst 0. Das Bit zeigt an, daß der Baustein anhand der Prüfsumme einen Fehler auf der Platte erkannt hat. Ist Bit 4 ebenfalls gesetzt, so lag der Fehler in einem Sektor-Adreßfeld, andernfalls in einem Datenfeld.
2	Datenverlust Das Bit zeigt an, daß der Baustein nach einer Anforderung über das Statusbit 1 nicht rechtzeitig (schrittweise) mit der Umdrehung der Diskette durch Lesen bzw. Schreiben des Datenregisters beendigt wurde.
1	Datenanforderung Das Bit fordert zum Lesen des nächsten Datenbytes (nach Lesebefehlen) aus dem Datenregister bzw. zum Schreiben des nächsten Bytes (nach Schreibbefehlen) in das Datenregister auf.
0	Baustein BUSY Das Bit zeigt an, daß der Baustein gerade mit der Ausführung eines Befehls beschäftigt ist.

Tabelle 1. Statusregister nach Befehlen von Typ 2 und 3

00 ausgeliefert, bis das Signal TR00 (vom Endschiefer des Laufwerks) das Erreichen von Spur 00 meldet. Damit ist die aktuelle Position des Kopfes bekannt; der 2797 selbst setzt sein Spurregister auf 00.

Vor der Ausgabe des Befehls 'Suchen' läßt man die Nummer der zu suchenden Spur in das Datenregister. Der Baustein vergleicht sie mit der Nummer der momentanen Spur im Spurregister und führt so viele Schritte in der richtigen Richtung aus, bis die gewünschte Spur erreicht ist. Damit steht im Spurregister dann auch die Nummer der neuen, nunmehr aktuellen Spur.

Jedes Laufwerk verfügt nur eine bestimmte Schrittweite und kommt 'aus dem Tritt', wenn man sie überschreitet. Deshalb müssen die Bits R1 und R0 bei jedem Positionierbefehl korrekt gesetzt werden.

Nach einem Positionierbefehl folgt sinnvollerweise ein Schreib- oder Lesebefehl, für den der Kopf aufgesetzt sein muß. Um die Beruhigungszeit beim Aussetzen zu sparen, kann man den Kopf mit Hilfe des 'K-90' schon zu Beginn des Suchens aufsetzen. Das Setzen des 'P-Bits' ist meist nicht sinnvoll, da die Ausführung der Prüfung Zeit kostet. Bei den Befehlen des Typs 2 wird außerdem die Spurnummer automatisch kontrolliert.

Datenzugriff

Das Schreiben und Lesen eines Sektors wird durch die Befehle des Typs 2 ermöglicht. Zuvor muß der Kopf auf die gewünschte Spur gestellt worden sein. Vor der Befehlsausgabe läßt man das Sektorregister mit der gewünschten Sektornummer. Das Spurregister steht meist wegen des vorangegangenen Positionierbefehls schon richtig. Zu Beginn wählt der 2797 über den Ausgang SSO (Side Select Output = Seitenwahl) die durch das 'S-Bit' vorgegebene Plattenseite an und setzt den Kopf auf. Dann liest er die Spur (auch bei 'Sektor schreiben') so lange, bis ein Sektoradreßfeld erscheint, das die richtige Spur-, Sektor- und Seitennummer einschließlich korrekter Prüfsumme (CRC) enthält. Falls der Controller innerhalb von fünf Plattenumdrehungen kein passendes Sektoradreßfeld findet, bricht er ab und setzt das Fehlerbit 4 (zusätzlich auch Bit 3) im Statusregister. Dieser Fehler kann durch schlechte Positionierung verursacht worden sein. Ein 'defensives' Programm wird deshalb 'Nullstellung' und 'Suchen' befehlen und dann einen neuen Versuch starten.

Im Normalfall wird ein passendes Sektoradreßfeld gefunden, und der 2797 wartet auf den Beginn des folgenden zugehörigen

Datenfeldes, um direkt mit dem Schreiben beziehungsweise Lesen zu beginnen (DRQ-Anforderung). Die Länge des Sektors (Anzahl der Datenbytes) ist beim Formatieren der Diskette festgelegt worden; der Baustein findet diese Angabe (Tabelle 1, L-Bit) im Sektoradreßfeld.

Adreßfeld lesen

Der Befehl 'Adreßfeld lesen' liest das nächste in der momentanen Spur am Kopf vorbeilaufende Sektoradreßfeld und liefert seinen Inhalt (6 Byte: Spur-, Seiten-, Sektornummer, Sektortlänge, Prüfsumme davon) als Daten ab. Im Normalfall kann man damit nicht allzuviel anfangen. Eine Anwendung wäre zum Beispiel ein schnelles 1:1-Kopierprogramm, das jeweils eine komplette Spur in einer einzigen Plattenumdrehung liest beziehungsweise schreibt. Der Befehl macht es dann möglich, unmittelbar an der augenblicklichen Stelle mit dem Lesen zu beginnen. Bei den folgenden Befehlen ('Sektor lesen/schreiben') fängt man dann nicht mit Sektor 1 an, sondern mit demjenigen, der gerade vorbeiläuft.

Spur lesen

Der Befehl 'Spur lesen' ermöglicht es, eine komplette Spur in einem Stück transparent (d. h. einschließlich aller Adreßmarken, Datenbytes, Prüfsummen, Synchronisationsbytes und Füllbytes) zu lesen. Die Auswertung des Gelesenen ist aber recht kompliziert, insbesondere dadurch, daß einige Bytes 'angeschnitten' sein können. Der Befehl wird im normalen Betrieb nicht benutzt. Er kann Anwendung finden bei Hardware-Tests, beim Überprüfen von Formatierprogrammen oder zur Untersuchung fremder Aufzeichnungsarten.

Spur formatieren

Der Befehl 'Spur formatieren' formatiert die gesamte Spur, auf der der Kopf gerade steht, indem er sie vollkommen neu mit leeren Sektoren beschreibt. Ein vorheriges Kontroll-Lesen, wie etwa bei 'Sektor schreiben', findet nicht statt. Da man normalerweise die gesamte Platte formatiert und damit löscht, ist das nicht gefährlich. Es ist sogar erwünscht, denn nur so

kann man eine fabrikneue oder fehlerhaft beschriebene Diskette 'auf Vordermann bringen'. Man beginnt dann mit Spur 00 und geht schrittweise mit Hilfe des 'Spur vor'-Befehls weiter, so daß Positionierfehler nicht zu befürchten sind.

Im Gegensatz zu anderen Bausteinen (zum Beispiel μ PD 765) ist das Formatieren beim 2797 ziemlich programmieraufwendig. Man muß dem Chip als Daten jeweils den kompletten Spurinhalte übergeben. Um diese Daten zusammensteilen zu können, bedarf es einer genauen Kenntnis der Datenstruktur auf einer Diskette. Es handelt sich dabei um mehrere tausend Bytes (siehe auch unter 'Spur lesen'), nicht etwa nur um Anzahl und Länge der Sektoren. Anders gesagt: Der Baustein 'weiß' gar nicht, wie man eine Platte formatiert. Es wird ihm per Programm mitgeteilt. Der Vorteil dieser Methode ist es, daß der Programmierer freie Hand hat und auch 'ausgefallene' Formate erzeugen kann.

Probleme . . .

Beim Arbeiten mit dem 2797 fielen zwei Probleme auf. Wie schon erwähnt, versucht der Baustein im Fehlerfall fünf Plattenumdrehungen lang, den gewünschten Sektor aufzufinden. Leider mißt er aber diese fünf Umdrehungen nicht zeitlich ab, sondern zählt einfach fünf Indexpulse (Eingang IP). Falls nun keine Diskette eingelegt ist, das Signal READY aber trotzdem 'high' ist (Tür kann geschlossen sein, Motor kann laufen), so wartet der 2797 'ewig'. Ein ihm anzulastender Fehler ist das strenggenommen nicht, es stört nur bei einigen 'Sparschaltungen'.

Ein echter Fehler zeigt sich dagegen bei den Befehlen 'Sektor lesen', 'Sektor schreiben' und 'Spur lesen'. Dabei darf man nach der Befehlsausgabe zunächst nur die Bits 0 (BUSY) und 1 (Datenanforderung) abfragen. Die übrigen Bits zeigen zu Beginn Fehler an, die nicht aufgetreten sind. Erst nach der Datenanforderung für das allererste Datenbyte (Statusbit 1) oder nach Abbruch des Befehls (Statusbit 0) sollte man die übrigen Statusbits abfragen. Dieser Fehler wirkt sich allerdings nur aus, wenn man den Datentransfer durch Polling abwickelt. □