

WDOS-Erweiterungsmodul - A4-Ergebnisbericht

I. Software

WEGA-DOS (WDOS)

Abarbeiten von DCP unter WEGA

Bearbeiter: P. Hoge

M. Klatter (sa.diags)

## Inhaltsverzeichnis

1.	Einleitung .....	3
2.	Komponenten von WDOS .....	4
2.1.	WDOS-Treiber cpu2 im WEGA-Kern .....	4
2.2.	WDOS-Koppelprogramm wdos .....	5
2.3.	WDOS-Menueprogramm wm .....	6
2.4.	"ROM"-BIOS .....	6
3.	DOS-Programme .....	7
4.	Ergebnisse der WDOS-Implementation .....	8
4.1.	Hardware-Voraussetzungen .....	8
4.2.	Lauffaehigkeit von Anwenderprogrammen .....	8
4.3.	Verfuegbarer RAM .....	8
4.4.	Tastatureingaben .....	9
4.5.	Bildschirmausgaben .....	9
4.6.	Diskzugriffe .....	9
4.7.	Systemuhr .....	11
4.8.	Systemdiskette .....	11
4.9.	DCP-Kommandos .....	12
5.	WDOS-Test-Version .....	13
6.	Hardware-Testprogramm sa.diags .....	13

## Anlagen zur Anlage 4

1.	WDOS-Dokumentation
2.	Quellprogramm cpu2.c
3.	Quellprogramm wdos.c
4.	Quellprogramm wm.c
5.	Quellprogramme fuer das BIOS
6.	Quellprogramme fuer die DOS-Programme

## 1. Einleitung

WDOS gestattet die Benutzung von PCDOS bzw. MSDOS und kompatiblen Betriebssystemen, wie DCP, unter WEGA. Fuer diese Betriebssysteme wird im folgenden die Bezeichnung DCP verwendet. Mit WDOS wird auf dem P8000 eine Teilkompatibilitaet zu DCP realisiert. Fuer die Benutzung von WDOS sind im P8000 die U80601-Erweiterungskarten (CPU-Karte / 1 MByte RAM-Karte) erforderlich. Auf dem P8000 muss WEGA Version 3.1 oder hoeher laufen.

Fuer die Abarbeitung der meisten DCP-Anwenderprogramme ist nur das P8000/WDOS-Terminal mit DCP-kompatibler Betriebsart geeignet.

Das Betriebssystem DCP meldet sich auch mit dem alten P8000-Terminal in der ADM31-kompatiblen Betriebsart. Einige Anwenderprogramme sind ebenfalls lauffaehig. Da aber fuer diesen Terminaltyp z.B. keine Funktionstasten, kein erweiterter Zeichensatz und kein simulierter Bildwiederholpeicher zur Verfuegung stehen, ist nur eine stark eingeschraenkte Arbeit moeglich.

Die wesentlichen Bestandteile von WDOS sind das WDOS-Menuprogramm wm, das Koppelprogramm wdos, der WDOS-Treiber cpu2 im WEGA-Kern und das ROM-BIOS. Koppelprogramm, Treiber und ROM-BIOS bilden in Verbindung mit den Erweiterungskarten die WDOS-Umgebung, in der das Betriebssystem DCP sowie die Anwenderprogramme ablaufen koennen.

WDOS arbeitet mit Programmen und Dateien, die im DCP-Format abgespeichert sind. Es ermoeglicht dem Nutzer, Dateien zu lesen oder zu schreiben, die z.B. auf dem A7150 oder EC1834 unter DCP erstellt wurden und erlaubt, viele unter DCP ausfuehrbare Programme ohne Aenderung abzuarbeiten. Programme, die Grafikausgaben enthalten, und Programme, die direkt auf die Hardware zugreifen, sind jedoch nicht lauffaehig.

WDOS simuliert mit einfachen Hardware-Mitteln eine Personal-Computer-Umgebung. Eine vollstaendig kompatible Umgebung steht nicht zur Verfuegung. Somit kann auch keine Garantie fuer die Lauffaehigkeit von speziellen Programmen uebernommen werden.

Unter WDOS laufen die DCP-Programme in Echtzeit ab. Auf den Erweiterungskarten wird dazu das unveraenderte Betriebssystem DCP gestartet. Es ergibt sich jedoch die Einschraenkung, dass nur ein WEGA-Nutzer zur gleichen Zeit DCP abarbeiten kann.

Die 1 MByte-RAM-Karte kann von WEGA auch als RAM-Disk benutzt werden. Die Abarbeitung von Programmen unter DCP ist dann jedoch nicht moeglich.

## 2. Komponenten von WDOS

### 2.1. WDOS-Treiber cpu2 im WEGA-Kern

Im Betriebssystem WEGA ist in Verbindung mit der CPU U8001 ein Schutzmechanismus realisiert, der verhindert, dass Anwenderprogramme direkt auf Systemressourcen zugreifen koennen. Nur der WEGA-Kern laeuft im Systemmode des U8001 und verfuegt damit ueber den vollen Befehlssatz der CPU. Alle anderen Programme laufen im User-Mode. Im User-Mode koennen die privilegierten Befehle des U8001 (z.B. E/A-Befehle) nicht benutzt werden. Dadurch koennen z.B. (fehlerhafte) Anwenderprogramme nicht den WEGA-Kern zerstoeren.

Fuer die Bedienung der Erweiterungskarten wurde der Treiber cpu2 fuer die Einbindung in den WEGA-Kern entwickelt. Zusaetzlich wurden in einigen weiteren Kernmodulen (event.s mch.s trap.c prf.c) geringe Erweiterungen noetig.

Der Treiber cpu2 simuliert in Verbindung mit dem "ROM"-BIOS fuer die CPU2 die peripheren Geraete Tastatur, Bildschirm, Harddisk, Floppydisk, Drucker und Echtzeit-Uhr.

Die Kommunikation mit der CPU2 wird ueber DUAL-PORT-RAM und ueber Interrupts realisiert. Der Treiber realisiert weiterhin die RESET-Steuerung fuer die CPU2.

Der Treiber realisiert folgende Kommunikationsaufgaben:

1. Tastatureingaben von WEGA zur CPU2
2. Bildschirmausgaben von der CPU2 zu WEGA
3. Druckerausgaben von der CPU2 zu WEGA
4. Diskzugriffe fuer die CPU2
5. Zugriff fuer WEGA auf die 1 MByte RAM der Erweiterungskarten
6. Realisierung einer Echtzeit-Uhr fuer CPU2

Fuer die Benutzung des Treibers sind im Directory /dev von WEGA folgende Geraete-Eintraege erforderlich:

Name	Art	Major	Minor	Zweck
/dev/cpu2tty	c	4	1	U80601-Terminal
/dev/cpu2lp	c	4	2	U80601-Drucker
/dev/cpu2disk	c	4	3	U80601-Disk-Request
/dev/cpu2rram	c	4	0	U80601-RAM-raw-Interface (Diskdaten)
/dev/cpu2ram	b	4	0	U80601-RAM-Blockinterface (1 MByte RAM-Disk fuer WEGA)

Der Treiber cpu2 darf nur vom WEGA-Koppelprogramm wdos benutzt werden. Eine Ausnahme bildet /dev/cpu2ram. Ueber diesen Gerate-Eintrag koennen die 1 MByte RAM der Erweiterungskarten von WEGA als RAM-Disk benutzt werden.

Um die Groesse des Treibers und die Belastung fuer die Haupt-CPU klein zu halten, realisiert der Treiber nur die Vermittlung von Daten zwischen WEGA und der CPU2 in Bloecken von 1 bis 800hex Bytes. Alle erforderlichen Konvertierungen muessen durch die CPU2 erfolgen.

## 2.2. WDOS-Koppelprogramm wdos

Durch Start des Koppelprogramms wega stellt der Anwender die Verbindung zur CPU2 her. Das Programm fuehrt folgende Initialisierungen aus:

1. Laden des "ROM"-BIOS bios bzw. bios.p8, in Abhaengigkeit der Umgebungsvariable TERM
2. Setzen des Terminaltreibers tty in den Raw-Mode
3. P8000/WDOS-Terminal in den WDOS-Mode schalten  
(Tastatur liefert SCAN-Code, Bildschirm realisiert abgeruesteten 8-Bit-Zeichensatz)
4. Initialisierung der Signalbehandlung
5. Aufheben von Reset fuer die CPU2

Anschliessend teilt sich das Programm in 4 Prozesse, die jeweils die verschiedenen Kommunikationsaufgaben realisieren:

1. Tastatureingaben von WEGA zur CPU2  
(Read von Treiber tty, Write zu cpu2tty)

2. Bildschirmausgaben von der CPU2 zu WEGA  
(Read von Treiber cpu2tty, Write zu tty. Um eine schnelle Ausgabe der Zeichen zu erzielen, werden alle Ausgabezeichen durch die CPU2 aufbereitet. Die Haupt-CPU kopiert nur Bloecke von 1 bis max. 800 hex Bytes.)
3. Druckerausgaben von der CPU2 zu WEGA  
(Read von Treiber cpu2lp, Write zum Druckertreiber, Druckerspooler oder in eine Datei)
4. Diskzugriffe fuer die CPU2 (Read/Write von/zu cpu2disk und cpu2rram, Zugriff erfolgt auf reale Harddisk-Partition, Floppy-Disk, Pseudoharddisks und Pseudo-Floppy-Disks)

### 2.3. WDOS-Menueprogramm wm

Zur einfacheren Handhabung des Programmsystems WDOS, wurde auf Basis der SCREEN/CURSES-Bibliothek ein Menueprogramm realisiert. Dieses Programm gestattet die Verwaltung der verschiedenen Diskarten, die Konvertierung von Dateien von/nach WEGA, die Steuerung der Druckerausgaben und den Start von DCP.

### 2.4. "ROM"-BIOS

Das "ROM"-BIOS ist die Schnittstelle der Software zur Hardware des Rechners. Alle Ein-/Ausgabefunktionen, die Systemuhr, verschiedene andere Dienste und der Start des Boot-Vorganges sind hier enthalten. Das BIOS befindet sich am Ende des 1 MByte Speicherbereiches der CPU2.

Bei WDOS ist das BIOS jedoch nicht in PROM's enthalten, sondern wird von der Haupt-CPU (U8001) in den RAM der Erweiterungskarten geladen. Das BIOS realisiert die Kommunikation mit WEGA.

Das BIOS wurde mit Hilfe des WEGA-Cross-Assemblers fuer den K1810WM86 erstellt. Es setzt sich aus folgenden Modulen zusammen:

1. bios.asm - Boot, Uhr u.a. Funktionen
2. dseg.asm - Datensegment, Interrupttabelle
3. term.asm - Terminalinterface fuer P8000/WDOS-Terminal
4. termp8.asm - Terminalinterface fuer (altes) P8000-Terminal

5. disk.asm - Diskinteface
6. orgs.asm - Systemeinspruenge, Kommunikationsbereich zu WEGA

### 3. DOS-Programme

Auf Basis von Quellprogrammen der TU Karl-Marx-Stadt wurden auf dem P8000 Programme implementiert, die von WEGA aus, den Zugriff auf DCP-Disketten, DCP-Harddisks, DCP-Pseudodisketten und -harddisks ermoglichen. Fuer diese Programme sind die U80601-Erweiterungskarten nicht erforderlich. Diese Programme werden unabhaengig von WDOS mit der WEGA Version 3.1 vertrieben. Diese Programme koennen auch vom WDOS-Menuprogramm aus aufgerufen werden. Folgende Programme stehen zur Verfuegung:

1. dosdir - DCP-Directory im DCP-Format anzeigen
2. dosls - DCP-Directory im WEGA-Format anzeigen
3. doscp - Kopieren von Dateien zwischen DCP und WEGA
4. doscat - DCP-Dateien ausgeben
5. dosrm - DCP-Dateien loeschen
6. dosmkdir - DCP-Directory erstellen
7. dosrmdir - DCP-Directory loeschen
8. dosformat - DCP-Diskette formatieren

#### 4. Ergebnisse der WDOS-Implementation

In diesem Kapitel sind wichtige Unterschiede und Besonderheiten bei der Abarbeitung von DCP unter WEGA aufgefuehrt.

##### 4.1. Hardware-Voraussetzungen

Fuer die Abarbeitung von DCP unter WEGA sind zwei U80601-Erweiterungskarten (CPU-Karte und 1MByte-RAM-Karte) im P8000 erforderlich, die auf 2 Slots der 16-Bit-Rechnerkarte im P8000 gesteckt werden. WDOS realisiert mit diesen Karten eine PC-Umgebung, die jedoch nicht vollstaendig PC-kompatibel ist. Als Terminal muss ein spezielles P8000/WDOS-Terminal mit 25 Bildschirmzeilen und Flach tastatur eingesetzt werden. Dieses Terminal besitzt eine PC-kompatible Betriebsart. In dieser Betriebsart liefert die Tastatur anstelle der ASCII-Zeichen einen SCAN-Kode und die Bildschirmsteuerung realisiert die meisten Zeichen des erweiterten ASCII-Zeichensatzes.

##### 4.2. Lauffaehigkeit von DCP-Programmen

Auf den U80601-Erweiterungskarten steht keinerlei Hardware zur Verfuegung, die direkt E/A-Geraete oder andere PC-kompatible E/A-Ports ansprechen kann. Anwenderprogramme duerfen daher keine In- und Out-Befehle enthalten. Alle Zugriffe auf periphere Geraete duerfen nur ueber das Betriebssystem oder das BIOS erfolgen. Eine Ausnahme bilden Bildschirmausgaben. Anwenderprogramme duerfen direkt in einen simulierten Bildwiederholpeicher schreiben. Es ist aber nur ein alphanumerischer Mode (entsprechend der MDA-Karte) realisiert. Alle Programme, die Grafikausgaben enthalten und auf Bildschirmkarten wie CGA, EGA ... angewiesen sind, koennen unter der WDOS-Umgebung nicht benutzt werden.

In der WDOS-Umgebung stehen 1 Systemkonsole (Tastatur/Bildschirm), 1 Drucker, max. 4 (Pseudo-)Floppy-Disk-Laufwerke und max. 2 (Pseudo-)Harddisk-Laufwerke zur Verfuegung. Serielle Schnittstellen sind nicht vorhanden.

##### 4.3. Verfuegbarer RAM

Dem Betriebssystem DCP und den Anwenderprogrammen stehen unter WDOS von 0 bis 9ffff 640 KByte RAM zur Verfuegung, sowie zusaetzlich von A0000 bis Affff weitere 64 KByte (in diesem Bereich liegt bei einem normalen PC die Grafikkarte). Im Bereich von B0000 bis B1fff wird von WDOS eine Monochrom-Bildschirmkarte simuliert. Der Bereich von F0000 bis Fffff ist fuer das "ROM"-BIOS

reserviert. Der Bereich von B2000 bis Efff ist zur Zeit frei und kann unter DCP z.B. fuer einen vom Anwender geschriebenen RAM-Disk-Treiber benutzt werden.

#### 4.4. Tastatureingaben

Die Tastatur ist PC-kompatibel und liefert einen SCAN-Kode. Unter DCP darf kein zusaetzlicher Tastatortreiber installiert werden, da dieser direkt auf die Hardware zugreift.

#### 4.5. Bildschirmausgaben

Im 1 MByte RAM wird eine Monochrom-Karte simuliert. Diese besitzt nur einen alphanumerischen Mode mit 25 Zeilen zu 80 Zeichen und einer Seite. Anwenderprogramme koennen direkt diesen "Bildwiederholtspeicher" beschreiben. Durch das "ROM"-BIOS wird der Bildwiederholtspeicher mit einem zweiten Abbild verglichen und die Differenzen zum Terminal ausgegeben. Das Terminal besitzt bezueglich der erweiterten ASCII-Zeichen und der VIDEO-Attribute Einschränkungen. Fast alle VIDEO-BIOS-Requests schreiben die Zeichen zuerst in den "Bildwiederholtspeicher". Eine Ausnahme bildet der WRTIE-TTY-Request (0eh). Hier erfolgt die Ausgabe direkt zum Terminal. Somit koennen auch ESC-Folgen (ADM31-kompatibel) ausgegeben werden. Das Betriebssystem DCP benutzt in der Regel diesen BIOS-Request.

Unter WDOS kann in DCP auch der Bildschirmtreiber "ANSI.SYS" installiert werden. Mit diesem Treiber werden ANSI-kompatible ESC-Folgen wirksam und die Ausgaben werden zuerst in den "Bildwiederholtspeicher" geschrieben. Der Vergleich der "Bildwiederholtspeicher" kostet aber Zeit und die Ausgabe der Zeichen ist daher etwas langsamer als die direkte Ausgabe zum Terminal.

Bei Verwendung des CP/M-Emulators EM80 fuer DCP (gehört nicht zum Lieferumfang von WDOS) sollte ANSI.SYS nicht installiert werden. OS/M-Programme, wie TP, sind fuer das P8000-Terminal installiert und dann mit EM80 in der WDOS-Umgebung sofort lauffaehig.

#### 4.6. Diskzugriffe

Das Betriebssystem DCP erkennt durch Informationen im Urlader jeder Diskette das entsprechende Format der Diskette und stellt sich logisch darauf ein. Es kann aber bei realen Disketten nicht das physische Format des Floppy-Disk-Treibers (im WEGA-Kern) beeinflussen. Achten Sie bitte daher beim "Mounten" von realen

Laufwerken im WDOS-Menue darauf, dass das Format mit dem Diskettenformat uebereinstimmt.

DCP ermittelt beim Systemstart das physische Format seiner Floppy-Disk-Laufwerke. Das Laufwerksformat ist aber nur beim Formatieren von Disketten von Bedeutung. Es hat keinen Einfluss beim Erkennen von Diskettenformaten. DCP erkennt z.B. eine 720 KByte Diskette in einem 360 KByte Laufwerk. Fuer die "physischen" Laufwerke E: und F: wird das Laufwerksformat mit der Aktivierung des Treibers DRIVER.SYS festgelegt (Standardeinstellung: 720 KByte Laufwerke). Fuer die Laufwerke A: und B: wird das physische Format aus dem "ROM"-BIOS ermittelt. Ist im Laufwerk A: eine 1,2 MByte-Pseudo-Systemdiskette eingelegt, liefert das "ROM"-BIOS die Werte fuer 1,2 MByte Laufwerke ohne Disk-Change-Signal, sonst fuer 720 KByte Laufwerke ohne Disk-Change-Signal. Durch Fehlen dieses Signals setzt nur PCDOS das richtige Laufwerksformat. MSDOS und DCP stellen stets 360 KByte Laufwerke ein. Durch folgende Eintraege in die Datei CONFIG.SYS kann das Laufwerksformat auf 720 KByte eingestellt werden:

```
drivparm=/d:0
Drivparm=/d:1
```

Mit drivparm lassen sich aber auch anderer Formate einstellen.

Das Disk-Change-Signal dient zum automatischen Erkennen von Diskettenwechseln. Es wird von den (Pseudo-)Laufwerken nicht bereit gestellt. Das Signal kann aber vom "ROM"-BIOS simuliert werden. Der Parameter /C von DRIVER.SYS und DRIVPARM aktiviert das Signal. DCP fuehrt dann weniger Diskettenzugriffe aus. Beachten Sie jedoch, dass dann ein Diskettenwechsel in einem Laufwerk nur noch moeglich ist, wenn zuvor ein Zugriff auf ein anderes Laufwerk erfolgte, z.B. mit dem Kommando DIR. Ein Wechsel des aktuellen Laufwerks mit dem Kommando CD reicht nicht aus.

Zum Zugriff auf CP/M-Disketten und speziellen DOS-Disketten koennen die Treiber CDRIVE.SYS und MDRIVE.SYS (nicht im Lieferumfang von WDOS) installiert werden. Es koennen aber nur die physischen Formate angesprochen werden, die in der WDOS-Umgebung zur Verfuegung stehen. Die Treiber muessen so installiert werden, dass der Floppy-Disk-Treiber des "ROM"-BIOS benutzt wird, z.B.:

```
device=cdrive.sys -rah -ul -g
device=mdrive.sys -rah -ul
```

In der WDOS-Umgebung wird von den DISK-BIOS-Requests FORMAT und VERIFY keine Funktion ausgefuehrt. Es wird die erfolgreiche Ausfuehrung der Funktion angezeigt. Somit koennen Kommandos wie FORMAT zum logischen Formatieren benutzt werden. Physisch koennen

reale Disketten direkt unter WEGA oder vom WDOS-Menue aus formatiert werden.

#### 4.7. Systemuhr

Beim Start von DCP werden die Anfangswerte fuer Datum und Zeit von WEGA uebernommen. Im "ROM"-BIOS ist eine Echtzeituhr realisiert. Diese erhaelt vom WEGA-Kern Clock-Impulse mit einer Frequenz von 10Hz. Beim verlassen von DCP wird die Uhr gestoppt. Wird anschliessend die Arbeit in DCP ohne Neustart fortgesetzt (Kommando r im WDOS-Menue bzw. wdos -r in WEGA) geht die Uhr nach.

#### 4.8. Systemdiskette

Im Lieferumfang von WDOS ist die Pseudosystemdiskette wdos\_sys enthalten. Als Systemdiskette kann aber auch eine reale Systemdiskette eines PC-kompatiblen Rechners benutzt werden. Folgende Systemdisketten wurden in der WDOS-Umgebung getestet:

- Systemdiskette DCP 3.20 fuer A7150
- Systemdiskette PCDOS 3.20 fuer IBM-PC
- Systemdiskette PCDOS 3.30 fuer IBM-PC
- Systemdiskette MSDOS 3.20 fuer Schneider-PC

Vor Verwendung einer fremden Systemdiskette, ist die Datei CONFIG.SYS dieser Diskette zu ueberpruefen. Diese Datei darf keine Treiber aktivieren, die das "ROM"-BIOS umgehen. Deshalb sollten zuerst alle Zeilen mit

```
device=...
```

entfernt werden.

Folgende Eintraege in der Datei CONFIG.SYS sind fuer die Arbeit in der WDOS-Umgebung guenstig aber nicht unbedingt erforderlich:

```
drivparm=/d:0 ...      (Format fuer Laufwerk A: setzen)
drivparm=/d:1 ...      (Format fuer Laufwerk B: setzen)
device=driver.sys /d:4 ... (Laufwerk E: aktivieren)
device=driver.sys /d:5 ... (Laufwerk F: aktivieren)
```

Auf jeder Systemdiskette befindet sich ein "Disketten"-BIOS. Probleme mit fremden Systemdisketten koennen entstehen, wenn dieses BIOS direkt die Hardware anspricht. Das "Disketten"-BIOS koennte z.B. in Abhaengigkeit der Version und des Datums des "ROM"-BIOS bestimmte Treiber des "ROM"-BIOS ersetzen.

Bei der Installation einer DCP-Harddisk sind keine Besonderheiten zu beachten. Sie erfolgt mit den Programmen FDISK, FORMAT und

COPY.

#### 4.9. DCP-Kommandos

Folgende Kommandos funktionieren in der WDOS-Umgebung nicht:

CTTY	Definition Primaerkonsole
MODE	Einstellen Geraetemodus
SELECT	Einstellen nationale Besonderheiten
GRAFTABL	Laden zweiter grafischer Zeichensatz
GRAPHICS	Vorbereiten grafische Print-Screen-Funktion
KEYBxx	Ersetzten Tastaturprogramm

Bei folgenden Kommandos sind Einschraenkungen zu beachten:

DISKCOPY	Physisches Kopieren von Disketten Unter WDOS ist das physisches Formatieren der Zieldiskette nicht moeglich.
FORMAT	Formatieren von Disketten WDOS gestattet nur das logische Formatieren von Disketten. 720 KByte "Laufwerke" gestatten nur das logische Formatieren dieses Formats. "High-Density- Laufwerke" gestattet das logische Formatieren von 1,2 MByte Pseudodisketten und von 40-spurigen Disketten. Physisch koennen Disketten im WDOS-Menue oder direkt unter WEGA formatiert werden.

Das folgende intene Kommando dient zum Aendern der Parameter der internen Laufwerke. In vielen Dokumentationen ist es nicht beschrieben:

DRIVPARM=/D:ddd [Schalter]

Schalter:

/D:ddd	logische Geraetennummer (0...255)
/F:f	Geraetetyp (0=160/180,320/360k, 1=1,2M, 2=720k) (Standard 720k)
/T:ttt	Spuren pro Seite (Standard 80)
/S:ss	Sektoren pro Spur (Standard 9)
/H:hh	Koepfe (Standard 2)
/C	Test auf Diskettenwechsel
/N	nichtwechselbare Magnetplatte

## 5. WDOS-Testversion

Zur Inbetriebnahme der U80601-Erweiterungskarten wurde eine WDOS-Testversion erstellt. Diese Version beinhaltet ein modifiziertes Koppelprogramm und ein modifiziertes "ROM"-BIOS. Die Test-Version dient vor allem zur Ueberpruefung der Funktionsfaehigkeit der 1 MByte RAM der Erweiterungskarten (DUAL-PORT-RAM) bei gleichzeitigem staendigen Zugriff durch Haupt-CPU und CPU2.

## 6. Hardware-Testprogramm sa.diags

Das Hardware-Diagnoseprogramm sa.diags wurde um die Testschritte zur Funktionspruefung der WDOS-Erweiterungskarten erweitert. Die Testschritte entsprechen im wesentlichen der Testsoftware des WDOS-Funktionspruefplatzes. Folgende Testschritte wurden realisiert:

1. Test des MP600-RAM (1MByte) vom U8000 aus
  - 1.1. Alle Segmente werden mit einem Bytemuster vollstaendig beschrieben. Regeln fuer die Mustererzeugung:
    - laenge l des Musters mit  $l < 2^x$ ,  $x \in \mathbb{N} = (0,1,2,3,\dots)$  (es wurde die Laenge  $l=17$  gewaehlt)
    - Die Musterbytes muessen voneinander verschieden sein.
  - 1.2. Vergleich der beschriebenen Segmente mit dem Soll-Bytemuster.  
Fehler: Fehlernummer, Adresse, Soll-Daten, Ist-Daten
2. Paritaetstest des MP600-RAM durch den U8000
  - 2.1. Die Segmente %70 bis %7f (MP600-RAM) werden mit einem beliebigen Muster gefuellt.
  - 2.2. Die Segmente werden Byte fuer Byte gelesen und dabei die Merzkelle fuer Paritaets-NMI getestet.  
Fehler: Fehlernummer, Adresse, Ist-Daten
3. Anlauf des MP600
  - 3.1. Der U8000 laedt ein kurzes Anlaufprogramm fuer den MP600 in den MP600-RAM und loescht einen Kommunikationsbereich. Es wird eine Transferkontrolle durchgefuehrt.  
Fehler: Fehlernummer, Adresse, Soll-Daten, Ist-Daten

- 3.2. Der U8000 setzt das RESET fuer den MP600 zurueck. Der MP600 beginnt mit der Befehlsabarbeitung und fuehrt folgende Befehle aus:

```

MOV     AX,CS
MOV     DS,AX           ;DSEG=CSEG=F000H
MOV     AX,returncode
MOV     RETCODE,AX     ;Returncode im Kommunikations-
                       ;bereich ablegen
HLT

```

(siehe dazu auch asm86-Listing 'MP600 TEST3')

- 3.3. Der U8000 wartet ca. 1 ms und ueberprueft den uebergebenen Returncode.  
Fehler: Fehlernummer, Soll-Code, Ist-Code

#### 4. INT-Verkehr U8000 <--> MP600

- 4.1. Vom U8000 werden ein INT-Initialisierungsprogramm, zwei Interrupt-Service-Routinen (dienen dem Test des INT-Vektors) und eine INT-Vektor-Tabelle fuer dem MP600 in seinen RAM geladen. Es erfolgt wiederum ein Kontrolllesen der uebertragenen Programmteile.

Fehler: Fehlernummer, Adresse, Soll-Daten, Ist-Daten

- 4.2. Der U8000 setzt das RESET-Signal des MP600 zurueck.

\* Der MP600 initialisiert sich fuer VINT

\* Meldung der erfolgreichen Initialisierung an den U8000, durch laden des Quittungsbytes 2 in den Kommunikationsbereich.

\* MP600 geht in den HALT-Zustand

- 4.3. Der U8000 wartet ca. 1 ms und ueberprueft dann das Quittungsbyte 2 fuer eine erfolgreiche INT-Initialisierung.

Fehler: Fehlernummer, Soll-Code, Ist-Code

- 4.4. U8000 sendet 256 Interrupts mit verschiedenen Vektoren (1,2,3,...,255,0) an den MP600. In der INT-Vektor-Tabelle wird vor jedem INT, der Zeiger auf die ISR1, auf den Tabellenplatz des naechsten Vektors geladen. Alle anderen Zeiger in der INT-Vektor-Tabelle zeigen auf die ISR2. Diese wird dann nur bei einem fehlerhaft gelesenen Vektor angesprungen. Nach Ausloesung eines INT durch den U8000 springt der MP600 in die ISR1. Dort wird der Stackpointer zurueckgesetzt, ein Quittungsbyte 3 fuer den korrekten Anspruch der ISR uebergeben und ein NVI (nichtvektorisierter INT) an den U8000 gesendet (durch OUT-Befehl). Danach geht

der MP600 in den HALT-Zustand und wartet auf den naechsten Interrupt.

4.5. Nachdem der U8000 einen INT an den MP600 sendete, wartet er ca. 1 ms und ueberprueft dann das Quittungsbyte 3 und seinen eigenen NVI-Status durch kontrollieren einer Merzkelle.  
Fehler: Fehlernummer fuer falsches Quittungsbyte oder kein NVI.

5. RAM-Test vom MP600 aus

5.1. Der U8000 laedt ein RAM-Testprogramm fuer den MP600 mit Transferkontrolle.  
Fehler: Fehlernummer, Adresse, Soll-Daten, Ist-Daten

5.2. Der U8000 setzt das RESET-Signal fuer den MP600 zurueck und der MP600 fuellt den 1MByte RAM mit einem vorgegebenen Datenmuster und der jeweiligen Segmentnummer.

5.3. Der U8000 wartet ca. 1 s und testet danach den RAM des MP600 auf das richtig Eingetragene Datenmuster.  
Fehler: Fehlernummer, Adresse, Soll-Daten, Ist-Daten

6. gemeinsamer Speicherzugriff

6.1. Der U8000 laedt ein Programm fuer den MP600 mit Transferkontrolle.  
Fehler: Fehlernummer, Adresse, Soll-Daten, Ist-Daten

6.2. Der U8000 setzt das RESET-Signal fuer den MP600 zurueck und der MP600 arbeitet sein Programm im Block %0f ab und greift auf Bloecke %00 bis %0e zu (schreiben lesen).

6.3. Der U8000 greift gleichzeitig auf die Bloecke %00 bis %0E zu (schreiben lesen).

6.4. Danach testet der U8000 den RAM auf korrekten Dateninhalt.  
Fehler: Fehlernummer, Adresse, Soll-Daten, Ist-Daten