

**ATLAS 9600/COLT 300
USER DEFINABLE
DISASSEMBLER
Programmer's Guide**

**Publ. No. 3 9DIS-03/A
January 1986**

INHALTSVERZEICHNIS

- 1. Einfuehrung
- 2. Bedienungselemente
 - 2.1.1 Bus Cycles
 - 2.1.2 Timestamp
 - 2.1.3 Cursorfunktionen
 - 2.2 Set-Marker
 - 2.3 Memory Select
 - 2.4 Start und Stop
 - 2.5 Scratch
- 3. Erstellung eines Disassemblers
 - 3.1 Aufbau eines Disassemblers
 - 3.1.1 Display Layout
 - 3.2 Der Controller
 - 3.3 uP spezifischer Teil
 - 3.3.1 Messages und Headlines
 - 3.3.2 Initialisierung
 - 3.3.3 Opcode Fetch Decoder
 - 3.3.4 Opcode Interpreter
- 4. Generierung eines lauffaehigen Disassemblers
 - 4.1 Assemblierung
 - 4.1.1 Assemblierung mit RMAC
 - 4.1.2 Assemblierung mit MB0
 - 4.2 Linken zu einem lauffaehigen Programm
 - 4.2.1 Linken mit Controller als .REL File
 - 4.2.2 Linken mit Controller als .LIB File
 - 4.3 Aufruf des generierten Disassemblers
- 5. Appendix
 - A Bibliotheksfunktionen
 - B Status Informationen
 - C Beispielprogramm
 - D Sourcelisting von PBXXXX.MAC
 - E Benoeetigte Disketten
 - F Zusaetzliche Dokumentation

Wordmaster - Trademark von MicroPro

MACRO-80 - Trademark von Microsoft

MP/M II - Trademark von Digital Research

1. Einfuehrung

Bei der Hardware-Software Integration oder waehrend der Entwicklung an einem Mikroprozessorsystem koennen Probleme auftreten, die sich mit einem Emulator nicht mehr analysieren lassen. Hier bietet sich der Logik Analysator an mit seinen grossen Speicherkapazitaeten und komplexen Triggermoeglichkeiten. Waehrend sich die Statussignale noch gut im Timingdiagramm verfolgen lassen, bereitet es Muehe den aufgezeichneten Programmablauf in der Datenliste zu interpretieren. Ein geeignetes Hilfsmittel ist hier der Disassembler, der die Daten in prozessorspezifische Mnemonics umwandelt und darstellt.

Die vorliegende Dokumentation bietet dem Anwender die Moeglichkeit seinen eigenen Disassembler zu schreiben. Dazu benoetigt er lediglich Kenntnisse in 8080 / Z80 Assembler. Der generierte Disassembler ist lauffaehig auf einem ATLAS / COLT System. Die zu interpretierenden Daten koennen dabei mit einem General Purpose Analyser, Universal Trace Modul oder auch durch einen kundenspezifischen Einschub aufgezeichnet werden. Die Programmentwicklung kann man ebenfalls auf einem ATLAS / COLT Systemdurchfuehren. Dazu benoetigt man das Macro Assembler Package MACRO - 80 von Microsoft, um den Befehlssatz des Z80 auszunutzen. Beschraenkt man sich auf den Befehlssatz des 8080, so kommt man auch mit dem Relocating Macro Assembler RMAC aus der im Lieferumfang des ATLAS / COLT Systems enthalten ist.

2. Bedienungselemente

2.1.1 Bus Cycles

Dieses Bedienungselement wird generell angeboten, dahinter verbirgt sich aber nur dann eine Funktion wenn sie durch den uP spezifischen Teil unterstuetzt wird.
Aufruf der Routinen >IFETCH< und >IFCYC<.

2.1.2 Timestamp

Die Timestampanzeige ist nur mit dem Einschub 9602 - 20 MHz moeglich, bzw. mit uP Trace Units die mit den 20 MHz SELA's aufgebaut wurden.
Aufruf der Routine >IFCYC<.

2.1.3 Cursorfunktionen

Die Cursorfunktionen 'up / down' inkrementieren, bzw. dekrementieren den Cursor zum naechsten gueltigen Instruction Fetch, bzw. Bus Cycle. Mit 'specify cursor' kann man den gewuenschten Cursor direkt eingeben. Sollte auf dieser Location kein gueltiger Bus Cycle gefunden werden, wird der Cursor auf den naechsten gueltigen gesetzt.
Aufruf der Routine >IFCYC<.

2.2 Setmarker

Hiermit wird der Setmarker mit dem gerade aktuellen Cursor gleichgesetzt.
Aufruf der Routine >IFCYC<.

2.3 Memory Select

Memory Select gibt die Moeglichkeit zum Umschalten zwischen Source und Reference Daten. Der aktuelle Memory Select Status wird in >FSSTAT< geflagged (siehe auch Appendix B).
Aufruf der Routinen >IFETCH< und >IFCYC<.

2.4 Start und Stop

Der Funktion von Start und Stop duerfte ziemlich klar sein. Nach Beendigung der Datenaufzeichnung werden die Routinen >DINITL<, >IFETCH< und >IFCYC< aufgerufen.

User definable disassembler

2.5 Scratch

Das Scratch Menue ermoeeglicht STORE / RECALL von Setups und Datensaeetzen. Abgespeicherte Datensaeetze koennen spaeter wieder in einen LA (oder auch Slot 0F) geladen und wieder disassembliert werden.

3. Erstellung eines Disassemblers

Um einen eigenen Disassembler zu erstellen muss natuerlich einige Editierarbeit geleistet werden. Diese Arbeiten kann man untergliedern in

- a) Programmierung der erforderlichen Routinen
- b) Aenderung von Tabellen und Texten, die das Interface zum Controller bilden

Dies soll in den folgenden Abschnitten erlaeutert werden. Zuvor wird aber der Editor gestartet um das Testprogramm zu editieren.

```
QA>WM PBZ80.MAC      Aufruf des WORDMASTER (Editor)
                     um einen Disassembler >PBZ80.MAC<
                     fuer einen Z80 Prozessor zu
                     editieren. Der Name kann dabei
                     beliebig gewaehlt werden.
```

Der Editor meldet sich mit seinem Prompt '*' und mit dem ersten Befehl laedt man ein bereits vorhandenes Programm, das den Rahmen fuer den zu erstellenden Disassembler enthaelt.

```
*Y PBXXX.MAC<CR>
```

Am Anfang des Programmes befindet sich ein Formular, das man immer ausfuellen sollte um den nachfolgenden Disassembler zu dokumentieren.

```
NAME      ('PBXXX')
TITLE     Disassembler fuer ATLAS / COLT
SUBTTL    XXXX uP Support
.Z80

;
;program      : Disassembler fuer XXXX uP
;-----
;system       : ATLAS/COLT unter MP/M
;-----
;configuration : 9602-10 / 20MHz Plug-In
;-----
;
;author       : V.Krichbaum   date: 16-OCT-1984
;-----
;
;modified by      author      date      rev.level
;-----
; Muster          I.Zilog      11-11-84   0.0
;
;
;assembler/compiler version : Microsoft MACRO-80
;
;list of library modules    : ATLTEST.REL
;
;description: Disassembler fuer XXXX uP mit
;             Darstellung von Bus Cycles
;
```

User definable disassembler

3.1 Aufbau eines Disassemblers

Der Disassembler im ATLAS / COLT System besteht aus zwei Komponenten:

- a) dem Controller und
- b) dem uP spezifischen Teil

die im Nachfolgenden beschrieben werden. Fuer den Disassembler steht eine RAM Bank mit insgesamt 8k Bytes (im Adressbereich 0E000H - 0FFFFH) zur Verfuegung, davon beansprucht der Controller nicht ganz 2,5k. Die verbleibenden 5,5k kann der Anwender als Programm- und Datenspeicher nutzen. Die Podbelegung mit Adressen, Daten und Statussignalen kann frei gewaehlt werden. Allerdings ist zu beruecksichtigen, dass Area Trace eine Qualifizierung auf den Pods A&B bewirkt. Deswegen ist es empfehlenswert diese Kanalgruppe fuer Adressen zu reservieren.

3.1.1 Display Layout

Die ersten drei Headlines werden vom Controller versorgt. Die Aufteilung fuer 'ADDRESS DATA MNEMONIC' ist im uP spezifischen Teil einstellbar. Fuer die eigentliche Disassemblierung stehen 17 Zeilen mit 71 Zeichen (minus 10 fuer Cursor / Timestamp Darstellung) zur Verfuegung.

```
SLOT 01 40x4000BIT REFERENCE ----- DATA FILE:Z80 .DAT
SEARCH FOR SRC=REF RESULT:EVENT:0000 TOTAL:0000 SETUP :.....SET
CURS=0181 CURS/SET=+0181 TRIGGER=1999 BUS CYCLE [ON ]
```

CURSOR	ADDRESS DATA		MNEMONIC -Z80-
0174	0023	D301	OUT (01H),A
0176	0081	08	i/o-write
0177	0025	D800	IN A,(00H)
0179	0000	08	i/o-read
0180	0027	08	int-acknowl
0181	608F	00	mem-write
0182	608E	27	mem-write
0183	0038	F3	DI
0184	0039	3E55	LD A,55H
0186	0038	325555	LD (5555H),A
0189	5555	55	mem-write
0190	003E	2F	CPL
0191	003F	32AAAA	LD (AAAAH),A
0194	AAAA	AA	mem-write
0195	0042	AF	XOR A
0196	0043	320000	LD (0000H),A
0199	0000	00	mem-write

3.2 Der Controller

Dieser Teil des Disassemblers wird von der Firma Dolch zur Verfuegung gestellt und bildet zum einen das Interface zum uebrigen System und steuert zum anderen den uP spezifischen Teil ueber die Bedienungselemente. Der Controller ist als Bibliotheksfunktion generiert und wird beim Linken (siehe auch Abschnitt 4.2) mit eingebunden. Einige Programme des Controllers sind als Subroutine aufrufbar und im Appendix A beschrieben. Waehrend der hier beschriebene Controller fest vorgegeben ist, sind die in Abschnitt 3.3 vorgestellten uP spezifischen Teile zu modifizieren (sieh auch >PBXXX.MAC) und groesstenteils selbst zu entwickeln. In >PBXXX.MAC< definierte Labels duerfen nicht veraendert werden, da sie sonst beim Linken den Fehler 'undefined symbol' erzeugen. Im Falle das der uP spezifische Teil keine interpretierbare Cycles (anhand der Statussignale) identifiziert, meldet sich der Controller mit

'NO FETCH CYCLES ENCOUNTERED'

waehrend des Aufbaues der Opcode Fetch Table (Abschnitt 3.3.3) mit

'DISASSEMBLER IS BUSY'

und bei Fehlbedienung mit

'INVALID ENTRY'

3.3 uP spezifischer Teil

Der uP spezifische Teil wird durch das existierende File >PBXXX.MAC< vorgegeben. Dabei sind die Messages und Headlines entsprechend der eigenen Anwendung abzuaendern und die erforderlichen Routinen zu entwickeln.

3.3.1 Messages und Headlines

Im uP spezifischen Teil sind zwei Headlines und eine Message zu versorgen. Die Message >BMSG1< enthaelt den Filenamen und die Versionsnummer. Beide Informationen koennen im Monitor ('select diagnostic monitor' und 'show slot directory') abgerufen werden. Die Headlines >HDRMSG< und >TIMHDR< sind bis auf die Cursor / Zeitdarstellung identisch (siehe Abschnitt 2.1.2). Ueber den Softkey 'ON / OFF TIMESTAMP' wird dieser Mode umgeschaltet und die entsprechende Headline dargestellt. Beide Headlines sollten in ihrer Laenge nicht veraendert werden.

User definable disassembler

3.3.2 Initialisierung

Eine Initialisierung wird nach jeder Datenaufzeichnung, bzw. Aufruf des Disassemblers nach einer Aufzeichnung, ueber die Routine >DINITL< durchgefuehrt. Generell initialisiert sie die Register >DFLAG<, >XYFLAG< und >OFFSTA< auf 00. Diese drei Register stehen ebenfalls frei zur Verfuegung. Darueberhinaus koennen natuerlich auch selbstdefinierte Register und Tabellen dort initialisiert werden. Sollte dies nicht notwendig sein genuegt ein einfaches 'RET'.

Beispiel aus >PBZ80.MAC<

```
DINITL: XOR      A
         LD       (DFLAG),A      ;hier nur als 8Bit Register
         LD       (XYFLAG),A    ;genutzt
         RET
```

Beispiel ohne notwendige Initialisierung

```
DINITL: RET
```

3.3.3 Opcode Fetch Decoder

Der Disassembler im ATLAS / COLT System ist selbstsynchronisierend, d.h. er kann von jedem aktuellen Cursorwert aus den naechsten Opcode Fetch identifizieren. Um dies zu leisten benoetigt er die Routine >IFETCH< die, aehnlich wie >DINITL<, nach jeder Aufzeichnung aufgerufen wird. Diese Routine wird fuer jeden Cursorwert aufgerufen, prueft die Statussignale und markiert anschliessend jeden gueltigen Opcode Fetch Cycle. Darueberhinaus liegt es beim Anwender ob er zusaetzlich den Bus Cycle Mode auswertet und auch 'memory read', 'memory write', 'i/o request' und andere Cyclen als gueltig markiert. Der Controller legt mit diesen Markern eine Tabelle an, die einen weiteren Vorteil bietet. Durch die Tabelle ist der Controller selbst in der Lage bei Betaetigung von Cursorfunktionen in beiden Richtungen den naechsten gueltigen Cycle zu identifizieren, ansonsten muesste dieser wieder anhand der Statussignale interpretiert werden.

```
Funktion:      read (status from pod)
-----
               if (bus cycle == opcode fetch)
                   accu = 00;
               else accu = 0FFH;
```

Einen kompletten Datensatz (Pod A - F) liest man wie folgt, dabei hat man auch die Auswahl zwischen Source und Reference Daten, abhaengig vom Memory Select (Abschnitt 2.3).

User definable disassembler

Vereinfachtes Beispiel aus >PBZ80.MAC<

```
IFETCH: LD      A,RDATA
        LD      (COMREQ),A
        CALL    COMSUB          ;lese kompletten Datensatz
        LD      IY,DATRG       ;Pointer zur Datentabelle
        LD      A,(FSSTAT)
        AND     09
        JR      NZ,IF1#        ;springe wenn Source selektiert
        LD      EC,006
        ADD     IY,EC           ;addiere Offset zu Reference
                                ;Daten
IF1#:   LD      A,(IY+02)       ;lade Datenbyte von Pod C,
                                ;hier Pod mit Statussignalen
        AND     00010111B      ;maskiere Bits fuer Opcode
                                ;Fetch
        CP      05             ;gueltig wenn M1=0/IOREQ=1/
                                ;MREAD=1
        LD      A,0FFH         ;markiere Cycle ungueltig
        JR      NZ,IF2#        ;springe wenn Cycle ungueltig
        XOR     A              ;markiere Cycle gueltig
IF2#:   RET
```

Das Originalbeispiel aus >PBZ80.MAC< ist etwas aufwendiger, da auch der Bus Cycle 'ON' Mod ausgewertet wird.

3.3.4 Opcode Interpreter

Diese Routine, >IFCYC<, bildet den eigentlichen interpretativen Teil Disassemblers. Sie liest die Adresse und das zugehoerige Datenbyte von der durch den Controller in >PCNTR< uebergebenen Location und stellt sie (nach Umwandlung in ASCII) im Line Buffer dar. Zur Darstellung der Mnemonic muss das Datenbyte natuerlich interpretiert und eventuell weitere Datenbytes als Operanden gelesen und dargestellt werden. Der Pointer zum Line Buffer wird im Register >CRTP2< uebergeben. Zuvor werden noch, entsprechend der Headline, fuer Adresse, Daten und Mnemonic die Feldpointer >APNT<, >DPNT< und >MPNT< besetzt. Um bei mehrbyte Befehlen weitere Datenbytes (Operanden) lesen zu koennen, wird der Pointer >PCNTR< auf die naechste Location inkrementiert. Damit spaeter der Controller weiter synchron laeuft, muss der Pointer auf die Tabelle mit den Opcode Fetch Markern >OMTP< parallel inkrementiert werden. Zur Umwandlung von Adress- und Datenbytes in ASCII stehen die Routinen >HBPNT< und >HEXAT< zur Verfuegung. Der Unterschied zwischen beiden Routinen liegt in der Verarbeitung von 1 Byte bzw. 1 Nibble, damit auch 4 Bit, z.B. bei 20 Bit Adressierung, dargestellt werden koennen (siehe auch Appendix A). Es ist empfehlenswert >IFCYC< in einen administrativen und einen interpretativen Teil >OPCODE< aufzutrennen, da der interpretative Teil dann 'stand alone' getestet werden kann.

User definable disassembler

```

IFCYC:  LD      HL,(CRTP2)      ;pointer to line buffer
        LD      (APNT),HL      ;SET FIELD POINTERS
        LD      BC,08
        ADD     HL,BC
        LD      (DPNT),HL
        LD      BC,21
        ADD     HL,BC
        LD      (MPNT),HL
        LD      A,RDATA
        LD      (COMREQ),A
        CALL    COMSUB          ;RDATA, read complete data set
        LD      IY,DATRG       ;pointer to data set
        LD      A,(FSSTAT)     ;get memory select status
        AND     9
        JR      NZ,10$        ;if source memory is selected
        LD      BC,6           ;offset to reference data set
        ADD     IY,BC
10$:    LD      HL,(PCNTR)
        LD      IX,(OMTP)
        INC     HL              ;inc actual cursor and pointer
        INC     IX              ;to opcode fetch table for
        LD      (PCNTR),HL      ;further read operations
        LD      (OMTP),IX
        LD      HL,(APNT)
        LD      A,(IY+00)       ;POD AC
        CALL    HBPNT          ;PRINT A12-A19
        LD      A,(IY+01)       ;POD B
        CALL    HBPNT          ;PRINT A4-A11
        LD      A,(IY+03)       ;POD D
        LD      B,A             ;save data byte for >OPCODE<
        LD      HL,(DPNT)
        CALL    HBPNT          ;PRINT DATA BYTE
        LD      (DPNT),HL
        CALL    OPCODE          ;DECODE INSTRUCTION passed in reg.B
        RET

```

User definable disassembler

4. Generierung eines lauffaehigen Disassemblers

Zur Generierung eines Systems benoetigt man die MP/M Utilities, bei Ausnutzung des Z80 Befehlssatzes zusaetzlich das Microsoft Macro Assembler Package MACRO-80.

Festlegungen: Bei RMAC haben Sourcefiles die Extension .ASM, bei M80 .MAC.

Objektfiles haben die Extension .REL, Library-files .LIB.

4.1 Assemblierung

4.1.1 Assemblierung mit RMAC

QA>RMAC PBZ80	Aufruf des Assemblers
CP/M RMAC ASSEM 1.1	Assemblierung von PBZ80.ASM und
0004	Erstellung von .REL (Objekt),
000H USE FACTOR	.PRN (Liste) und .SYM (Symbole)
END OF ASSEMBLY	
QA>	

4.1.2 Assemblierung mit M80

QA>M80	Aufruf des Assemblers
*PBZ80,PBZ80=PBZ80	Assemblierung von PBZ80.MAC und
	Erstellung von .REL (Objekt)
No Fatal error(s)	und .PRN (Liste)

User definable disassembler

4.2 Linken zu einem lauffähigen Programm

4.2.1 Linken mit Controller als .REL File

0A>LINK MMZ80=MM04,PBZ80[LE000] Aufruf des Linkers

LINK 1.3 Erstellung von MMZ80.COM

BMSG1	E94A	BMSG1C	0025	DFLAG	E943	DINITL	E9EF
HBPNT	E856	HDRMSG	E96F	HEXAT	E85F	IDTXT1	E94C
IDTXT2	0020	IFCYC	EAE6	IFETCH	E9F7	MNEMON	E000
OFFSTA	E947	RDBYT1	E839	RDBYTE	E81A	TIMHDR	E9AF
XYFLAG	E946						

ABSOLUTE 0000

CODE SIZE 1432 (E000-F431)

DATA SIZE 0000

COMMON SIZE 0000

USE FACTOR 46

0A>

User definable disassembler

4.2.2 Linken mit Controller als .LIB File

0A>LINK MMZ80=PBZ80,MM04.LIB[S,LE000,GMNEMON] Aufruf des Linkers

LINK 1.3

Erstellung von MMZ80.COM

BMSG1	E005	BMSG1C	0025	DFLAG	F430	DINITL	E0AA
HBPNT	F343	HDRMSG	E02A	HEXAT	F34C	IDTXT1	F007
IDTXT2	0020	IFCYC	E1A1	IFETCH	E0B2	OFFSTA	F434
RDBYT1	F326	RDBYTE	F307	TIMHDR	E06A	XYFLAG	F433
MNEMON	EAED						

ABSOLUTE 0000

CODE SIZE 1432 (E000-F434)

DATA SIZE 0000

COMMON SIZE 0000

USE FACTOR 46

0A>

User definable disassembler

4.3 Aufruf des generierten Disassemblers

Bei einem gebooteten LA-Einschub erscheint bei Betaetigung des MNEMONIC Flipkeys das LOADER MENUE, das es ermoeeglicht einen Disassembler zu laden. Mit nochmaliger Betaetigung des Flipkeys aktiviert man den Disassembler. Da man in der Regel taeglich mit dem Disassembler arbeitet kann man ihn auch direkt booten lassen. Dies erreicht man dadurch, dass man ihn mit einer MP/M Utility auf >MM02< 'renamed'.

QA>REN MM02.COM=MMZ80.COM

5. Appendix

A Bibliotheksfunktionen

Read data set at actual >PCNTR<

Call : COMSUB

COMREQ : RDATA=89H

Entry Parameter : cursor location to read from in >PCNTR<

Exit Condition : data set in >DATRG<, see appendix B

Read data byte from pod D (mostly used for D0-D7) at actual >PCNTR< and print it into data field of line buffer

Call : RDBYTE

Entry Parameter : cursor location to read from in >PCNTR<, data field pointer in >DPNT<

Exit Condition : data byte of pod D in accu A and reg. B >PCNTR< and >DMTP< incremented

Read data byte from pod D (mostly used for D0-D7) at actual >PCNTR<

Call : RDBYT1

Entry Parameter: cursor location to read from in >PCNTR<

Exit Condition : data byte of pod D in accu

Print hex byte in ASCII

Call : HBFNT

Entry Parameter : hex byte in accu, field pointer in reg.HL

Exit Condition : field pointer in reg.HL incremented

User definable disassembler

Print hex nibble in ASCII

Call : HEXAT

Entry Parameter : hex nibble in lsb of accu, field pointer
in reg.HL

Exit Condition : field pointer in reg.HL incremented

User definable disassembler

B Status Informationen

Register : COMREQ
Description : register to pass the common request in case of calling a routine from main program
Value : in this application 89H to call >RDATA<, see appendix A also

Register : DATRG
Description : table for 6 source and 6 reference data, each pod A - F, offset 00 - 05H for pod A - F source, offset 06 - 0BH for pod A - F reference, recommended to be accessed by index register
Value : --

Register : PCNTR
Description : actual cursor position, normally supported by controller, must be incremented by uP specific part to read additional data bytes, must be decremented to look for previous address, data or control bytes
Value : 000 - cursor max.

Register : FSSTAT
Description : memory select status register
Value : source if 2⁰ or 2³ high, otherwise reference

Recommended sequence: LD A,(FSSTAT)
AND 09
JR NZ,SOURCE ; ITS SOURCE
.. ... ; ITS REFERENCE

User definable disassembler

Register : OMTP
Description : pointer to opcode fetch marker table
Value : if bit >FOPTM< (equated) high it marks a valid cycle at actual cursor >PCNTR<, please note that >PCNTR< and >OMTP< have to be manipulated synchronously

Register : APNT, DPNT, MPNT
Description : pointer to address, data and mnemonic fields within line buffer
Value : --

Register : DFLAG (16 bit), XYFLAG (8bit), OFFSTA (8 bit)
Description : registers for free use
Value : --

Register : BUTFLG
Description : status flag for bus cycle mode
Value : bit 2^0 - 0/1 means off/on

User definable disassembler

C Beispielprogramm

DISASSEMBLER FOR ATLAS / COLT MACRO-80 3.44 09-Dec-81 PAGE 1

```

NAME ('PBZ80')
TITLE DISASSEMBLER FOR ATLAS / COLT
SUBTTL PERSONALITY BOARD Z80
.Z80

;
;
;program      :      Z80 interpreter
;-----
;system       :      ATLAS 9600
;-----
;configuration :      SLOT 9604 with PBZ80
;-----
;
;author        :                      date:
;-----
;
;modified by    author      date      rev.level
;-----
;compatible with WILHELM    13/07/83      2.0
;COLAOS version 2.0
;BUS-cycles included
;-----
;DEBUGGED VERSION  WI          13/09/83      2.10
;WITH MEM-READ FLAG
;-----
;bus cycle status display VK          31/01/84      3.00
;timestamp on/off toggle
;-----
;DINITL - initialize  L.Hechler    24/09/84      3.01
;data RAMs
;-----
;
;assembler/compiler version      :      Microsoft MACRO-80
;
;list of library modules      :      MM04 Controller Module
;
;
;external input data:
;-----
;external output data:
;-----
;
;internal data:
;-----
;
;description:      uP specific part as example for user
;-----      definable disassembler
;
PAGE 55

```

DISASSEMBLER FOR ATLAS / COLT MACRO-80 3.44 09-Dec-81 PAGE 1-1
 PERSONALITY BOARD 780

```

;
;
;=====
;      SYSTEM EQUATES DECLARATION
;=====
;
0109      CONSUB EQU    109H      ;COMMON SUBROUTINE CALLING SEQUENCE
;
01BF      OMTF  EQU    01BFH
01C3      BUTFLG EQU    01C3H
01C4      APNT  EQU    BUTFLG + 1
01C6      DPNT  EQU    APNT + 2
01C8      MPNT  EQU    DPNT + 2
0DA4      COMREQ EQU    0DA4H
0F20      DATRG EQU    0F20H
0196      FSSTAT EQU    0196H
01BA      PCNTR EQU    01BAH
0DB7      CRTP2 EQU    0DB7H
;
007F      TERM  EQU    7FH
0006      TERMA EQU    06H
0000      CR    EQU    00H
000A      LF    EQU    0AH
0008      XYLISA EQU    08H
0002      LISA  EQU    02H
;
0004      FOPTM EQU    04H
;
;=====
;      DISASSEMBLER COMMON ROUTINE EQUATES
;=====
;
0009      RDATA EQU    09H
;
;=====
;      DISASSEMBLER LOCAL EQUATES
;=====
;
0000'     ADDR: DEFS  2
;
;=====
;      INTERNAL DECLARATION
;=====
;
      PUBLIC  BMSG1,BMSG1C
      PUBLIC  DINITL
      PUBLIC  HDRMSG
      PUBLIC  IFCYC,IFETCH,IDTXT1,IDTXT2
      PUBLIC  TIMHDR

```

```
;  
;=====;  
;        EXTERNAL DECLARATION  
;=====;  
;  
;        EXTRN    APNT  
;        EXTRN    BUTFLG  
;        EXTRN    COMREQ,CRTF2  
;        EXTRN    DATRG,DPNT  
;        EXTRN    DFLAG  
;        EXTRN    FSSTAT  
;        EXTRN    HBPNT,HEXAT  
;        EXTRN    MPNT  
;        EXTRN    OMTP  
;        EXTRN    OFFSTA  
;        EXTRN    PCNTR  
;        EXTRN    RDBYTE,RDBYT1  
;        EXTRN    XYFLAG  
;  
;        PAGE
```

DISASSEMBLER FOR ATLAS / COLT MACRO-80 3.44 09-Dec-81 PAGE 1-3
PERSONALITY BOARD Z80

```

;
;*****
;*
;* Z80 DISASSEMBLER *
;*
;*****
;
; POD A: A8 - A15
;      B: A0 - A7
;      C: control lines 2^0 - 2^7 -> wr, rd, iorq, areq, m1, nmi, int, halt
;      D: D0 - D7
;      E: control lines 2^0 - 2^7 -> rfs, wait, reset, busrq, busak, nc, nc, nc
;          or timestamp
;      F: control lines 2^0 - 2^7 -> nc, nc, nc, nc, nc, nc, nc, area
;          or timestamp
;
; PAGE

```

User definable disassembler

DISASSEMBLER FOR ATLAS / COLT MACRO-80 3.44 09-Dec-81 PAGE 1-4
PERSONALITY BOARD 280

```

;-----
;Disassembler Message Part
;-----
;      boot message
;
BMSG1: DEFB      TERM,TERMA
IDTXT1: DEFM      'MMZ80 .COM version 3.01 loaded'

0002' 7F 06
0004' 4D 4D 5A 38
0008' 30 20 20 20
000C' 2E 43 4F 4D
0010' 20 76 65 72
0014' 73 69 6F 6E
0018' 20 33 2E 30
001C' 31 20 6C 6F
0020' 61 64 65 64
0020
IDTXT2 EQU      $ - IDTXT1
0024' 00 0A      DEFB      CR,LF
0026' 24      DEFM      '$'
0025
BMSG1C EQU      $ - BMSG1
;
;=====
; H D R M S G - HEADLINE MESSAGE WITH CURSOR-INFORMATION
;=====
;
HDRMSG: DEFB      XYLISA,0,5,LISA
        DEFM      'CURSOR ADDRESS DATA'
Mnemonic -Z80-

0027' 00 00 05 02
0028' 43 55 52 53
002F' 4F 52 20 20
0033' 20 20 41 44
0037' 44 52 45 53
003B' 53 20 44 41
003F' 54 41 20 20
0043' 20 20 20 20
0047' 20 20 20 20
004B' 20 20 20 20
004F' 20 20 20 4D
0053' 4E 45 4D 4F
0057' 4E 49 43 20
005B' 2D 5A 38 30
005F' 2D 20 20 20
0063' 20
0064' 0D 0A 7F      DEFB      CR,LF,TERM
;
;
;=====
; T I M H D R - HEADLINE MESSAGE WITH TIMESTAMP-INFORMATION
;=====
;
TIMHDR: DEFB      XYLISA,0,5,LISA
        DEFM      'TIME ADDRESS DATA'
Mnemonic -Z80-

0067' 00 00 05 02
006B' 54 49 4D 45
006F' 20 20 20 20

```

DISASSEMBLER FOR ATLAS / COLT MACRO-80 3.44 09-Dec-81 PAGE 1-5
PERSONALITY BOARD 780

0073' 20 20 41 44
0077' 44 52 45 53
007B' 53 20 44 41
007F' 54 41 20 20
0083' 20 20 20 20
0087' 20 20 20 20
008B' 20 20 20 20
008F' 20 20 20 4D
0093' 4E 45 4D 4F
0097' 4E 49 43 20
009B' 2D 5A 38 30
009F' 2D 20 20 20
00A3' 20
00A4' 0D 0A 7F

DEFB CR,LF,TERM

;
;
;=====
; D I N I T L - INITIALIZE DATA RAMs
;=====

00A7' AF DINITL: XOR A
00AB' 32 0000* LD (DFLAG),A
00AB' 32 0000* LD (XYFLAG),A
00AE' C9 RET

;
;=====
; I F E T C H - INSTRUCTION FETCH DECODER
;=====

00AF' 3E 89 IFETCH: LD A,RDATA
00B1' 32 0DA4 LD (COMREQ),A
00B4' CD 0109 CALL COMSUB
00B7' FD 21 0F20 LD IY,DATRG
00BB' 3A 8196 LD A,(FSSTAT)
00BE' E6 09 AND 9
00C0' 20 05 JR NZ,IF1\$
00C2' 01 0006 LD BC,6
00C5' FD 09 ADD IY,BC
00C7' FD 66 00 IF1\$: LD H,(IY+00)
00CA' FD 6E 01 LD L,(IY+01)
00CD' FD 7E 02 LD A,(IY+02)
00D0' E6 17 AND 00010111B ;MASK OUT M1/BUSCYCLE
00D2' FE 05 CP 05 ;VALID IF M1=0/IORQ=1/RD=1
00D4' 47 LD B,A
00D5' E5 PUSH HL ;SAVE ACTUAL ADDRESS
00D6' C2 010A' JP NZ,BUSTST
;
00D9' 3E FF LD A,0FFH ;SET FOR NO IFETCH
00DB' 21 0000* LD HL,DFLAG ;TEST FOR PREVIOUS DOUBLE BYTE OPCODE
00DE' CB 7E BIT 7,(HL)
00E0' CB 8E RES 7,(HL)

DISASSEMBLER FOR ATLAS / COLT MACRO-80 3.44 09-Dec-81 PAGE 1-6
PERSONALITY BOARD 280

```

00E2' 20 1C          JR      NZ,IFTST
;
00E4' CB 46          BIT      0,(HL)
00E6' CB 86          RES      0,(HL)
00E8' 20 16          JR      NZ,IFTST
00EA' FD 7E 03       LD      A,(IY+03)      ;POD D
00ED' FE CB          CP      0CBH          ;SKIP DOUBLE BYTE OPCODES
00EF' 20 0C          JR      Z,SKIPM1
00F1' FE DD          CP      0DDH
00F3' 20 08          JR      Z,SKIPM1
00F5' FE ED          CP      0EDH
00F7' 20 04          JR      Z,SKIPM1
00F9' FE FD          CP      0FDH
00FB' 20 02          JR      NZ,NOSKIP
00FD' CB C6          SKIPM1: SET 0,(HL)
00FF' AF            NOSKIP: XOR  A          ;SET FOR IFETCH
0100' B7            IFTST: OR   A          ;TEST IFETCH CONDITION
0101' E1            POP      HL
0102' 23            INC      HL          ;PREVIOUS ADDRESS
0103' 22 0000'       LD      (ADDRS),HL
0106' CD 0100'       CALL    NMI
0109' C9            RET

;
;=====
;B U S T S T - FLAG BUSCYCLE
;=====
;
BUSTST: LD      A,(BUTFLG)      ;TEST DISASSEM.-MODE
        BIT      0,A
        LD      A,0FFH
        JR      Z,IFTST          ;RETURN IF DISAS. WITHOUT BUSCYCLES
        LD      A,B          ;CONTROLL DATA IN ACCU
        CP      03H          ;INT-ACKNOWLEDGE
        JR      Z,NOSKIP
        CP      016H          ;MEM-WRITE
        JR      Z,NOSKIP
        CP      011H          ;I/O-READ
        JR      Z,NOSKIP
        CP      012H          ;I/O-WRITE
        JR      Z,NOSKIP
        CP      015H          ;MEM-READ
        LD      A,0FFH          ;SET FOR NO BUSCYCLE
        JR      NZ,IFTST
        LD      BC,(ADDRS)      ;GET EXPECTED ADDRESS
        OR      A
        SBC      HL,BC          ;TEST IF ACTUELL ADDRESS = EXPECTED ADDRESS
        JR      NZ,NOSKIP      ;VALID READ-CYCLE
        LD      HL,(PCNTR)
        PUSH    HL
        DEC      HL          ;TEST IF MACHINE-CYCLE BEFORE,ARE A VALID
        LD      (PCNTR),HL      ;READ-CYCLE

```

DISASSEMBLER FOR ATLAS / COLT MACRO-80 3.44 89-Dec-81 PAGE 1-7
PERSONALITY BOARD 190

```

013B' 3E 89          LD      A,RDATA
013D' 32 8DA4        LD      (COMREQ),A
0140' CD 0109        CALL    COMSUB
0143' FD 7E 02       LD      A,(IY+02)      ;GET CONTROLL DATA OF LOCATION BEFORE
0146' E6 17          AND     00010111B
0148' FE 15          CP      015H
014A' E1            POP     HL
014B' 22 81BA        LD      (PCNTR),HL
014E' 20 0C          JR      NZ,BT1$      ;RETURN IF NO READ-CYCLE
0150' DD 2A 81BF     LD      IX,(ONTP)    ;IS THE READ-CYCLE FLAGED?
0154' DD 2B          DEC     IX
0156' DD CB 00 66    BIT     FOPTM,(IX+00)
015A' 20 A3          JR      NZ,NOSKIP    ;YES FLAG READ-CYCLE OF ACTUELL LOCATION
015C' 3E FF          BT1$: LD      A,0FFH
015E' 1B A0          JR      IFTST

;
;=====
; N M I
;=====
;
NMI:
0160' F5            PUSH    AF
;
0161' CD 0182'      CALL    READC      ;GET BYTE WITH CONTROLLSIGNALS
;
0164' CB 6F          BIT     5,A
0166' 20 18          JR      NZ,NMI2$
0168' 2A 81BA        LD      HL,(PCNTR)
016B' E5            PUSH    HL
016C' 2B            DEC     HL
016D' 22 81BA        LD      (PCNTR),HL
0170' CD 0182'      CALL    READC      ;GET PREVIOUS CONTROLLSIGNALS
0173' E1            POP     HL
0174' 22 81BA        LD      (PCNTR),HL  ;CORRECT PCNTR
0177' CB 6F          BIT     5,A
0179' 2B 05          JR      Z,NMI2$    ;STILL NMI
;
017B' 21 0000*       LD      HL,DFLAG
017E' CB FE          SET     7,(HL)
0180' F1            NMI2$: POP     AF
0181' C9            RET

;
READC:
0182' 3E 89          LD      A,RDATA
0184' 32 8DA4        LD      (COMREQ),A
0187' CD 0109        CALL    COMSUB
018A' FD 21 8F20     LD      IY,DATRG
018E' 3A 8196        LD      A,(FSSTAT)
0191' E6 09          AND     9
0193' 2B 05          JR      NZ,NMI1$
0195' 01 0006        LD      BC,6
0198' FD 09          ADD     IY,BC

```

DISASSEMBLER FOR ATLAS / COLT MACRO-80 3.44 09-Dec-81 PAGE 1-8
 PERSONALITY BOARD Z80

```

019A'  FD 7E 02      NMI1$: LD      A,(IY+02)
019D'  C9              RET

;
;=====
; I F C Y C - BUILD A DISASSEMBLED LINE IN CRTBUF
;=====
;
IFCYC: LD      HL,(CRTP2)
        LD      (APNT),HL      ;SET FIELD POINTERS
        LD      BC,08
        ADD     HL,BC
        LD      (DPNT),HL
        LD      BC,21
        ADD     HL,BC
        LD      (MPNT),HL
        LD      A,RDATA
        LD      (COMREQ),A
        CALL    COMSUB      ;RDATA
        LD      IY,DATRG
        LD      A,(FSSTAT)
        AND     9
        JR      NZ,IC10$
        LD      BC,6
        ADD     IY,BC
IC10$: LD      HL,(PCNTR)
        LD      IX,(OMTP)
        INC     HL      ;INC COUNTER/POINTER
        INC     IX
        LD      (PCNTR),HL
        LD      (OMTP),IX
        LD      HL,(APNT)
        LD      A,(IY+00)      ;POD A
        CALL    HBPNT      ;PRINT A12-A19
        LD      A,(IY+01)      ;POD B
        CALL    HBPNT      ;PRINT A4-A11
        LD      A,(IY+03)      ;POD D
        LD      B,A
        LD      HL,(DPNT)
        CALL    HBPNT      ;PRINT DATA BYTE
        LD      (DPNT),HL
        CALL    HBCODE      ;DECODE INSTRUCTION
        RET
PAGE

```

DISASSEMBLER FOR ATLAS / COLT MACRO-88 3.44 09-Dec-81 PAGE 1-9
PERSONALITY BOARD I80

```

;*****
; STAND ALONE DISASSEMBLER
;*****
;
;=====
;M B C O D E- DECODE BUSCYCLE
;=====
;
01FB' FD 7E 02      MBCODE: LD    A,(IY+02)      ;POD C (CONTROLL DATA)
01FE' E6 17          AND    00010111B
0200' FE 05          CP     05              ;TEST FOR M1-CYCLE
0202' CA 0215'       JP     Z,OPCODE
0205' 47            LD     B,A
0206' 0E 05          LD     C,05           ;NUMBER OF BUSCYCLES
0208' 11 0639'       LD     DE,MMT0
020B' 21 0634'       LD     HL,MCT0
020E' CD 07E9'       CALL   DECD0
0211' CD 0A09'       CALL   PMNEM          ;WRITE MNEMONIC OF BUSCYCLES
0214' C9            RET

;
;=====
; O P C O D E - TEST FOR INDEXED INSTRUCTION
;=====
;
0215' 21 0000*      OPCODE: LD    HL,XYFLAG
0218' 36 00          LD     (HL),0        ;NO INDEX MARK
021A' 78            LD     A,B
021B' E6 DF          AND    11011111B
021D' FE DD          CP     11011101B
021F' 20 0C          JR     NZ,NOINDX     ;NOT INDEXED
0221' CB FE          SET    7,(HL)        ;SET INDEX MARK
0223' CB 68          BIT    5,B           ;TEST IX/IY
0225' CA 0557'       JP     Z,INDEXE
0228' CB C6          SET    0,(HL)        ;SET IY FLAG
022A' C3 0557'       JP     INDEXE
022D'               NOINDX:
;
;
;TEST FOR CB CODE
;
022D' 78            LD     A,B
022E' FE CB          CP     0CBH
0230' CA 045E'       JP     Z,TSTCB
;
;TEST FOR ED CODE
;
0233' FE ED          CP     0EDH
0235' CA 04AD'       JP     Z,TSTED
;

```

```
                                ;TEST FOR 1 BYTE OPCODE  
                                ;  
0238'  21 0684'                LD      HL,CT0  
0238'  11 0697'                LD      DE,MT0  
023E'  0E 13                   LD      C,19  
0240'  CD 09E9'                CALL    DECOD  
0243'  38 04                   JR      C,TST1  
0245'  CD 0A09'                CALL    PMNEM  
0248'  C9                      RET  
                                ;
```

User definable disassembler

DISASSEMBLER FOR ATLAS / COLT MACRO-80 3.44 09-Dec-81 PAGE 5
PERSONALITY BOARD Z80

Macros:

Symbols:

0000' ADDR5	01C4 APNT	00021' BMSG1
00251' BMSG1C	015C' BT1\$	010A' BUSTST
01C3 BUTFLG	049C' CBSHFT	07F5' CCLST
00A4 COMREQ	0109 COMSUB	0000 CR
00B7 CRTP2	0684' CT0	0727' CT1
0923' CT10	0765' CT2A	0767' CT2B
0709' CT3	0711' CT4A	0713' CT4B
078B' CT5	0799' CT6	079B' CT6ED
081B' CT7	084C' CT8	08F5' CT9
0F20 DATRG	09E9' DECOD	09EF' DECOD1
09F5' DECOD2	09EA' DECOD3	017C* DFLAG
00A71' DINITL	01C6 DPNT	07C8' DRLST1
07D4' DRLST2	07E0' DRLST3	07EA' DRLST4
0A03' ERRMSG	0004 FOPTH	0196 FSSTAT
09B8* HBPNT	00271' HDRMSG	09B7* HEXAT
01CA' IC10\$	0969' ICT0	00041' IDXTT1
00201 IDXTT2	00C7' IF1\$	019E1' IFCYC
00AF1' IFETCH	0100' IFTST	0AC2' ILLCOD
0925' INT0	0557' INDEXE	0565' IND00
05FC' IND01	0614' IND02	0572' IND0A
0506' IND0B	059D' IND0C	05B1' IND0D
05C4' IND0E	05D7' IND0F	05EB' IND0G
09D5' INXREG	09E5' INVREG	032D' JRCND
000A LF	0002 LISA	0963' MADD
01FB' MBCODE	0634' MCT0	07AF' MDJNZ
07A9' NJR	079D' MLD	0639' MMT0
01C8 MPNT	07A3' MRST	0697' MT0
072F' MT1	0911' MT10	076A' MT2
0711' MT3	0715' MT4	0721' MT4A
077F' MT5	078D' MT6	0809' MT7
081E' MT8	084C' MT9	0160' NMI
019A' NM11\$	0180' NM12\$	022D' NOINDX
00FF' NOSKIP	0ABE* OFFSTA	01BF OMTP
0215' OPCODE	0A86' PADR	0A93' PADR1
01BA PCNTR	0A48' PCOMMA	0A40' PDR1
0A09' PMNEM	0A4D' PNTACH	0A54' PNTCMA
0A5C' PNTCMX	0A69' PNTCR	098D' PNTDHX
09B7' PNTD11	09A5' PNTDIS	09C4' PNTH
0AB0' PNTHD	0970' PNTHX	0AA6' PNTHL
0973' PNTHX1	097F' PNTHX2	0A23' PNTIDX
0A64' PNTLD	0A7D' PNTPA1	0A78' PNTPAR
0A13' PNTRGB	0A1E' PNTRG1	09B4' POSDIS
0A99' PPORT	09CA' PRINT1	09C6' PRINTM
0089 RDATA	0000* RDBYT1	0ABA* RDBYTE
0AB8' RDOFF	0182' READC	07B5' RLST
00FD' SKIPM1	007F TERM	0006 TERMA
00671' TIMHDR	0249' TST1	0355' TST10

User definable disassembler

DISASSEMBLER FOR ATLAS / COLT MACRO-80 3.44 09-Dec-81 PAGE S-1
PERSONALITY BOARD Z80

0375'	TST11	0385'	TST11A	0389'	TST11B
03AD'	TST12	0389'	TST12A	03CF'	TST13
03ED'	TST13A	03FA'	TST13B	0412'	TST13C
0419'	TST13D	0431'	TST13E	0438'	TST14
0457'	TST14A	0269'	TST1A	0270'	TST2
0284'	TST3	0299'	TST4	0286'	TST5
02D0'	TST6	02ED'	TST7	0307'	TST8
031E'	TST9	0348'	TST9A	0326'	TST9X
045E'	TSTCB	046F'	TSTCB0	048B'	TSTCB1
0495'	TSTCB2	04AD'	TSTED	04C1'	TSTED1
04F0'	TSTED2	04FD'	TSTED3	051D'	TSTED4
0547'	TSTED5	0A04'	XTAC	09FD'	XTRACT
0A14*	XYFLAG	0008	XYLISA		

No Fatal error(s)

User definable disassembler

D Sourcelisting von PBXXXX.MAC

NAME ('PBXXXX')
TITLE DISASSEMBLER FOR ATLAS / COLT
SUBTTL PERSONALITY BOARD XXXX
.Z00

```
;
;
;program      :      XXXX interpreter
;-----
;system       :      ATLAS 9600
;-----
;configuration :      SLOT 9604 with PBXXXX
;-----
;
;author       :              date:
;-----              ----
;
;modified by   author       date       rev.level
;-----       -----       ----       -----
;
;-----
;
;assembler/compiler version :      Microsoft MACRO-00
;
;list of library modules    :      MM04 Controller Module
;
;
;external input data:
;-----
;external output data:
;-----
;
;internal data:
;-----
;
;description:      uP specific part as example for user
;-----          definable disassembler
;
```

User definable disassembler

```

;
;
;=====
;      SYSTEM EQUATES DECLARATION
;=====
;
;CONSUB EQU    107H      ;COMMON SUBROUTINE CALLING SEQUENCE
;
;OMTP EQU     81BFH
;BUTFLG EQU    81C3H
;APNT EQU     BUTFLG + 1
;DPNT EQU     APNT + 2
;MPNT EQU     DPNT + 2
;COMREQ EQU    8DA4H
;DATRG EQU    8F20H
;FSSTAT EQU    8196H
;PCNTR EQU    81BAH
;CRTP2 EQU    8DB7H
;
;TERM EQU     7FH
;TERMA EQU    06H
;CR EQU       0DH
;LF EQU       0AH
;XYLISA EQU    08H
;LISA EQU     02H
;
;FOPTM EQU    04H
;
;=====
;      DISASSEMBLER COMMON ROUTINE EQUATES
;=====
;
;RDATA EQU    89H
;
;=====
;      DISASSEMBLER LOCAL EQUATES
;=====
;
;ADDRS: DEFS   2
;
;=====
;      INTERNAL DECLARATION
;=====
;
;      PUBLIC BMSG1,BMSG1C
;      PUBLIC DINITL
;      PUBLIC HDRMSG
;      PUBLIC IFCYC,IFETCH,IDTXT1,IDTXT2
;      PUBLIC TIMHDR

```

User definable disassembler

```
;
;=====
;  EXTERNAL DECLARATION
;=====
;
;  EXTRN  APNT
;  EXTRN  BUTFLG
;  EXTRN  COMREQ,C RTP2
;  EXTRN  DATRG,DPNT
;  EXTRN  DFLAG
;  EXTRN  FSSTAT
;  EXTRN  HBPNT,HEXAT
;  EXTRN  MPNT
;  EXTRN  QMTP
;  EXTRN  OFFSTA
;  EXTRN  PCNTR
;  EXTRN  RDBYTE,RDBYT1
;  EXTRN  XYFLAG
;
```

User definable disassembler

```

;
;*****
;*
;* XXXX DISASSEMBLER *
;*
;*****
;
; POD A:
; B:
; C:
; D:
; E:
; F:
;
; PAGE
;-----
;Disassembler Message Part
;-----
; boot message
;
BMSG1: DEFB TERM,TERMA
IDTXT1: DEFM 'MMXXX .COM version x.xx loaded'
IDTXT2 EQU $ - IDTXT1
        DEFB CR,LF
        DEFM '$'
BMSG1C EQU $ - BMSG1
;
;=====
; H D R M S G - HEADLINE MESSAGE WITH CURSOR-INFORMATION
;=====
;
HDRMSG: DEFB XYLISA,0,5,LISA
        DEFM 'CURSOR ADDRESS DATA MNEMONIC -XXXX-'
        DEFB CR,LF,TERM
;
;
;=====
; T I M H D R - HEADLINE MESSAGE WITH TIMESTAMP-INFORMATION
;=====
;
TIMHDR: DEFB XYLISA,0,5,LISA
        DEFM 'TIME ADDRESS DATA MNEMONIC -XXXX-'
        DEFB CR,LF,TERM

```

User definable disassembler

```
;
;
;=====
; D I N I T L - INITIALIZE DATA RAMs
;=====
;
DINITL: XOR    A
        LD     (DFLAG),A
        LD     (XYFLAG),A
        RET

;
;=====
; I F E T C H - INSTRUCTION FETCH DECODER
;=====
;
IFETCH:
        RET

;
;=====
; I F C Y C - BUILD A DISASSEMBLED LINE IN CRTBUF
;=====
;
IFCYC:
        CALL   OP CODE      ;DECODE INSTRUCTION
        RET
```

User definable disassembler

```
;*****  
; STAND ALONE DISASSEMBLER  
;*****  
;  
;  
;=====  
; O P C O D E -  
;=====  
;  
OPCODE:  
    RET  
;  
    END
```

User definable disassembler

E Benoetigte Disketten

2 Disketten mit MP/M Operating System & Utilities

Disketten mit den Utilities: WM, MB0

1 Diskette mit den Files : MM04.REL, MM04.LIB, PBXXX.MAC,
PBZB0.MAC

F Zusaetzliche Dokumentation

Microsoft Macro-B0 Utility Manual

Digital Research LINK-B0 Operator's Guide und

MAC Macro Assembler: Language Manual and Applications Guide

Wordmaster Reference Manual

Beispielprogramm >PBZB0.MAC<

Operator's Manual Microprocessor Trace Unit Series 9604