TWO PASS ASSEMBLER

INTRODUCTION

This TWO PASS ASSEMBLER has been specifically designed to run on the Tug 80 column card with the Microtan interfaced to a Silver-Reed EXP 500 daisy wheel printer for hard copy. Spaced output to the 80 column card is provided in assembly and source list modes of operation.

DASM is the most complete and effective TWO PASS ASSEMBLER available for the Microtan system outside Flex and CP/M, using in all modes of operation the 8082 card and providing disc file handling facilities.

The systems operator is advised to study this manual before real applications are applied.

SYSTEM REQUIREMENTS

Minimum:-

Tugbug Xbug Vbug 6(PTL).
Microtan 65.
Tanex fitted with 8k of ram and two VIAs.
Ram from C000 to DFFF.
Tug 8082 card.
A Diablo compatible printer connected to the second VIA.

Optional:-

Tanram with ram running contiguous from $2000.
Tandos with one or more drives.

The package is supplied in cassette form for loading into random access memory, the following should be observed:-

Filename..."DASM"

Location...$C000-$DEB1

Start $C000

TWO PASS ASSEMBLER COMMANDS

BREAK/CTRL C = Command mode

A0	Full assembly listing
A1	Errors only listing
A2	Assemble to ram with offset
A3	Assemble direct to ram
A0P	Printer listing of second pass
Lxx	List source code
LP	Printer listing of source code
Txx	List lables
TP	Printer listing of lables
TE	Top of table address
Exx	Edit source code
S,xx	Search for string
S	Continue search
ST	Restart search
Nxx	Automatic line numbering on
NO	Automatic line numbering off
NA	Increment line number by 10
R	Renumber source code
Ixx	Initialise
G	Disc read source code
W	Disc write source code
D	Tape save source code
F	Tape fetch source code
V	Tape verify source code
J	Merge two sources
H	Relocate source code
K	Start new source code
P0	Source code protection
P1	Remove protection
C1	80 characters per line
C2	40 characters per line
M	Return to monitor

ELEMENTS OF THE ASSEMBLY LANGUAGE

LABELS:

Labels are used for:

1. Identifying program points mnemonically, or by name, rather than by machine address of the location.

2. A numerical value which is not a program point.

3. Automatically updating all references to a program point when its machine address changes due to changes in the program.

A Label consists of one to six characters from the set A - Z & 0 - 9. ('SPACES' ARE COUNTED AS CHARACTERS!)

THE FIRST CHARACTER IN A LABEL MUST BE A LETTER.

There is one predefined label '*' which refers to the program counter.

CONSTANTS:

The Assembler permits the use of three data formats, Ascii - Hexidecimal - Decimal:

With no prefix - decimal constant	eg	171
With '$' prefix - hexidecimal constant	eg	$20
With ''' prefix - Ascii constant	eg	'B

The colon ':' is only required if a label is used. The label and comment fields are optional.

```
Eg.       EXAM:LDA TEST         ; Get Check
               STX VALUE + 1    ; Save IX
          CONT:SEC
```

DIRECTIVE STATEMENTS:

The fields in a directive statement are:

(Label) : (Directive) (Operand) ; (Comment)

DEFB

DEFINE BYTE:

```
Eg      DEFB VALUE,0,$20,'A,100,VALUED -7.
```

DEFW

DEFINE WORD: (In low, High order by byte)

```
Eg      DEFW VALUE,0,1000,$412,VALUE + 20.
```

DEFM

DEFINE MESSAGE:

```
Eg      DEFM 'This is a test'
```

Message enclosed in the quote marks "'" CHR Code $27

EQUALS (=) DIRECTIVE:

SETTING THE PROGRAM LOCATION ORIGIN

```
Eg     *  =  $0400
       *  =  1400
       *  =  START
       *  =  END - 40
```

RESERVING SPACE:

```
Eg     *  =  " + 20          reserves 20      places
       *  =  " + $20             "    32        "
       *  =  " + 7000            "    7000      "
```

EXPRESSIONS:

The Arithmetic operators are:

```
'+'  Add
'-'  Subtract
```

The format is:

(Label) (+ or-) (Constant)

```
Eg:     START + 1
        OPCHR - 2
        CHR   + $20
        VALUE + 'A
```

ASSEMBLER STATEMENTS:

COMMENT STATEMENTS:

- provide headings and other explanatory information.
- these do not offset the machine code generated by the Assembler.

A comment statement is created by a semicolon ';' as the leftmost non blank character followed by any commentary whatever.

```
Eg.     ; ASSEMBLER
        ;''OUTPUT SUBROUTINE''
        ;
```

INSTRUCTION STATEMENTS:

The fields in an instruction statment are:
(Label) : (Opcode) (Operand) ; (Comment)

The spaces between each field is optional

EQUATING LABELS TO VALUES:

```
Eg.     TEN  = 10
        EXAM = $400
```

COMMANDS:

ASSEMBLY COMMANDS:

A0 - FULL ASSEMBLY LISTING DOES NOT WRITE TO RAM

From left to right for each line of source code.

- Program location
- Object (Machine) Code
- Source code line number
- Source code

After the first pass the user is asked to enter the line from which A0 should start. A carriage return starts A0 from the first line. Scrolling can be stopped by keying S and restarted with any key, BREAK returns to command mode.

A1 - ERRORS ONLY LISTING DOES NOT WRITE TO RAM

A2 - ASSEMBLE TO RAM WITH OFFSET

After typing A2 (CR) the program responds with 'Offset' = (Offset in Hex) Now type in your offset value followed by (CR)

Eg If you want a M/C program starting at $2000 where the source code starts , in this case at $2000 and finishes at $2470, we can operate an A2 assembly with Offset 500. This will store the program from $2500 onwards as though it was written into $2000 onwards where it was intended to go. - Very useful for relocating programs for Eproms.

A3 - ASSEMBLE DIRECT TO RAM

If any errors occur in any of the Assembly modes, the source code line is displayed.

TOP OF TABLE ADDRESS

TE returns the address of the first ram location free above the lable table. This table starts immediately above the source code.

EDITOR

E (CR) List source code from the beginning for editing.
E xx (CR) List source code from line number xx to end.

Only eight lines are output at one time, to continue the listing, type any key. To abort type BREAK. A maximum of 32 characters per line can be entered.

EDIT & CURSOR CONTROL COMMANDS

CNTRL	U	= Cursor	UP
"	D	– "	DOWN
"	R	– "	RIGHT
"	L	– "	LEFT
"	T	– "	MOVE ALL CHARACTERS RIGHT OF THE CURSOR ONE PLACE TO THE RIGHT
"	I	– "	TAB CURSOR 21 PLACES TO THE RIGHT OF THE MARGIN
"	E	–	ERASE CHARACTER INDICATED ABOVE CURSOR

DELETE

To delete a line, type in the line number followed by (CR).

LIST

L (CR) List source code from the beginning.
L xx (CR) List source code from line number xx to end.
Scrolling can be stopped by keying S and restarted with any key, BREAK returns to command mode.

ASCII STRING SEARCH COMMANDS

S,xx (CR)

Searches through the source code until it locates a matching Ascii string to xx, at which point it prints out the line containing this string.

S (CR)

Continues the search for more Ascii strings equal to xx from the top of the source code.

ST (CR)

Continues to search for more Ascii strings equal to xx from the top of the source code.

AUTOMATIC LINE NUMBERING

Nxx (CR)

Automatic line numbering ON.
After each return a new line number is printed xx greater than the last one where xx is in hexidecimal.

Eg For a step of 10 type NA (CR) and the line number printed will be 10 greater than the last output. Spurious line numbering will occur from the start.

NO (CR)

Automatic line numbering OFF.
To leave the automatic line numbering control, backspace to the left hand margin and type NO (CR)

AUTOMATIC LINE RENUMBERING

– R

Resequences all line numbers by 10 starting from 10.

PRINTER COMMANDS

AOP LP and TP.

These commands provide for the printing of:-

AOP - printing the second pass
LP - printing the source code
TP - printing the lables

After the first pass the user is asked to enter the following for AOP and LP:-

Printer on (Y/N)?

Wait between pages?

From page (1-99)?

Title? (up to 30 characters in length)

Date? (up to 15 characters in length)

Scrolling can be stoped by keying S and restarted with any key, BREAK returns to command mode. A carriage return after From page (1-99)? starts AOP and LP from the first page.

TP provides the following:-

Printer on (Y/N)?

Wait between pages?

LIST LABLES

Tn lists lable where n = >1 and < = 5. Defaults to 2.

SOURCE CODE PROTECTION

P0 to protect source code
P1 to remove protection

P0 uses Tanram paging facilities to inhibit writing to Tanram. The source code should be located above $2000 and the object below $2000. With protection in operation Tanram can still be read.

VDU CONTROL

C1 selects 80 characters per line
C2 selects 40 characters per line

CASSETTE FILE HANDLING

The cassette file handling routines are entered from the command level, the required Cuts or Fast speed being set by the input of the capital letter appropriate to the question Fast (Y/N)? To dump data to tape simply input "D" and CR, the program will respond automatically and await from the operator the required filename, on completion input a CR and the dump to cassette will commence. On completion of the file dump the command level is reentered.

Verify and loading of a cassette file are similar to saving a file.

F to load a file.
V to verify a file.

On completion of the load operation the assembler is automaticly initialised and the command level reentered.

A file name may be up to six characters in length.

DISC FILE HANDLING

DISC READ

The read disc option is entered from command level by typing G followed by a carriage return.

The assembler responds:-

File retrieval

Enter as

"DRV:Filename"

Enter file name and drive number, failure to enter drive number will default to DRV 0.

The assembler will then request S or B.

Where:-

S = Source - File suffixed SRC.
B = Back up - File suffixed SBK.

SRC and SBK are added automatically to the file name on disc write.

Select source or back up and the chosen file will be loaded into ram, upon completion the file xxxx loaded is displayed and the command level is reentered.

DISC WRITE

The write disc option is entered from command level by typing W followed by a carriage return.

The assembler responds:-

File creation

Enter as

"DRV:Filename"

Enter file name and drive number a search is now made to see whether the filename already exists, if not present the assembler creates file xxxx.SRC.

If file is already present a search is now made for the back up file SBK of the same name.

If the back up copy is present it will now be deleted, the present source SRC made the back up file SBK and the new source file is saved on disc with the suffix SRC.

If no back up file SBK is found the present source SRC will be made SBK and the new source file saved on disc as SRC.

The command level is now reentered.

THE DISC BUFFER

With so many different memory configurations now available for the Microtan the following information on the disc buffer has been provided.

The editor is provided with a 2K buffer for use with the disc option, the package supplied positions this store at $E800-$EFFF.

To make the package more flexible here is provided the necessary modifications to move the above store to a more convenient position.

Modify the following addresses:-

```
D21A,   D222,   D228,   D230,   D243,
D248,   D24B,   D252,   D259,   D265,
D271,   D42B,   D430,   D4D7,   D4DC,
```

RELOCATE

H relocates the source code above free or backwards.

MERGE TWO SOURCES

J adds a second source code to the end of the current source code.

INITIALISE

Ixx sets up assembler parameters to the source code starting at xx.

RETURN TO MONITOR

- M

TO START THE ASSEMBLER

Activate the program GC000 (CR)

On activation the program will respond:

Initialise (Y/N)?

If you reply 'Y' the program will respond:

Source Add = Enter 2000 (CR)

Type in the Hex value of the start address of the source code, as this package resides at C000 - DEB1 the start must be below this.

From this point the program sets up the required parameters for assembly and jumps into command mode and all previous data is destroyed.

WARM START:

If at any time the user wishes to re-enter the program from the Monitor, at 'initialise (Y/N)? reply 'N', at this point the program will jump to command mode and all previous data is secure.

If by accident you answer 'Y', hit 'Reset' and re-enter by GC000 (CR).

TWO PASS ASSEMBLER COMMANDS

BREAK/CTRL C = Command mode

A0	Full assembly listing
A1	Errors only listing
A2	Assemble to ram with offset
A3	Assemble direct to ram
AOP	Printer listing of second pass
Lxx	List source code
LP	Printer listing of source code
Txx	List lables
TP	Printer listing of lables
TE	Top of table address
Exx	Edit source code
S.xx	Search for string
S	Continue search
ST	Restart search
Nxx	Automatic line numbering on
NO	Automatic line numbering off
NA	Increment line number by 10
R	Renumber source code
Ixx	Initialise
G	Disc read source code
W	Disc write source code
D	Tape save source code
F	Tape fetch source code
V	Tape verify source code
J	Merge two sources
H	Relocate source code
K	Start new source code
PO	Source code protection
P1	Remove protection
C1	80 characters per line
C2	40 characters per line
M	Return to monitor

TWO PASS ASSEMBLER PART TWO

PART TWO PROVIDES THE FOLLOWING FACILITIES

EDITOR

Lxx is now Exx.

E (CR) List source code from the beginning for editing.
E xx (CR) List source code from line number xx to end.

Only eight lines are output at one time, to continue the listing, type any key. To abort type BREAK. A maximum of 32 characters per line can be entered.

EDIT & CURSOR/CONTROL COMMANDS

CNTRL	U =	Cursor	UP
"	D -	"	DOWN
"	R -	"	RIGHT
"	L -	"	LEFT
"	T -	"	MOVE ALL CHARACTERS RIGHT OF THE CURSOR ONE PLACE TO THE RIGHT
"	I -	"	TAB CURSOR 21 PLACES TO THE RIGHT OF THE MARGIN
"	E -		ERASE CHARACTER INDICATED ABOVE CURSOR

DELETE

To delete a line, type in the line number followed by (CR).

LIST

Provides spaced output to the 80 column card.

L (CR) List source code from the beginning.
L xx (CR) List source code from line number xx to end.
Scrolling can be stopped by keying S and restarted with any key, BREAK returns to command mode.

AO - FULL ASSEMBLY LISTING DOES NOT WRITE TO RAM

Provides spaced output to the 80 column card.

From left to right for each line of source code:

- Program location
- Object (Machine) Code
- Source code line number
- Source code

After the first pass the user is asked to enter the line from which AO should start. A carriage return starts AO from the first line. Scrolling can be stopped by keying S and restarted with any key, BREAK returns to command mode.

PRINTER COMMANDS

Provides spaced output to the 80 column card and printer.

AOP LP and TP.

These commands allow for the printing of:-

AOP - printing the second pass
LP - printing the source code
TP - printing the lables

After the first pass the user is asked to enter the following for AOP and LP:-

Printer on (Y/N)?

Wait between pages?

From page (1-99)?

Title? (up to 30 characters in length)

Date? (up to 15 characters in length)

Scrolling can be stopped by keying S and restarted with any key, BREAK returns to command mode. A carriage return starts AOP an LP from the first page.

TP provides the following:-

Printer on (Y/N)?

Wait between pages?

TOP OF TABLE ADDRESS

TE returns the address of the first ram location free above the lable table. This table starts immediately above the source code.

CASSETTE FILE HANDLING

The cassette file handling routines are entered from the command level, the required Cuts or Fast speed being set by the input of the capital letter appropriate to the question Fast (Y/N)? To dump data to tape simply input "D" and CR, the program will respond automatically and await from the operator the required filename, on completion input a CR and the dump to cassette will commence. On completion of the file dump the command level is reentered.

Verify and loading of a cassette file are similar to saving a file.

F to load a file.
V to verify a file.

On completion of the load operation the assembler is automaticly initialised and the command level reentered.

A file name may be up to six characters in length.

SOURCE CODE PROTECTION

P0 to protect source code
P1 to remove protection

P0 uses Tanram paging facilities to inhibit writing to Tanram. The source code should be located above $2000 and the object below $2000. With protection in operation Tanram can still be read.

VDU CONTROL

C1 selects 80 characters per line
C2 selects 40 characters per line