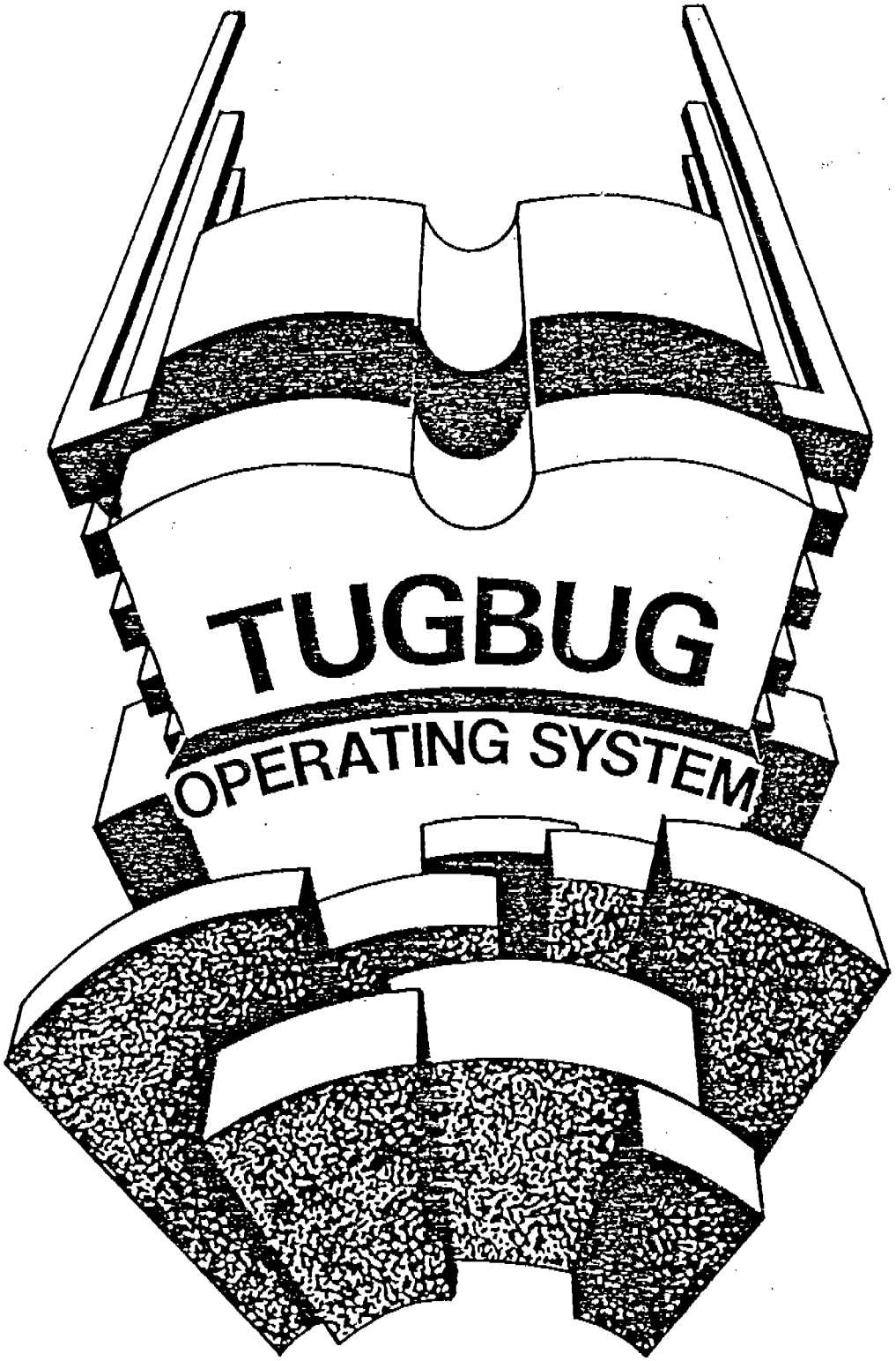

V1·0

CARD ADDRESSES

CARD 0 = $BD00 ..... CARD 4 = $BC40

CARD 1 = $BD08 ..... CARD 5 = $BC48

CARD 2 = $BD80 ..... CARD 6 = $BCC0

CARD 3 = $BD88 ..... CARD 7 = $BCC8

To use a call facility under program control the user must set up the following:-

MODADL $1C and $1D

This must contain the Eprom Start Addr.

LOPL $1E and $1F

This must contain the Eprom End Addr.

HXPKL $13 and $14

This must contain the Ram Start Addr.

In addition the 'A' register must contain:-

BITS 1 - 3 the card number (0 - 7)

BITS 5 - 6 to select the Upper or Lower banks on a 128K E.S.C. BIT 5 for Lower - BIT 6 for Upper banks.

BIT 8 if reset means READ from the card. If set means WRITE to the card.

BITS 4 - 7 are reserved for expansion if required.

BITS 5 - 6 are only set when a 128K E.S.C. is in use.

BIT 8 is only used when a Combo Module with resident 6116 Ram is to be written to.

The Read/Write routine corrupts all registers. It also uses locations $2 (OCHAR), $A & $B (ICURS) as workspace. ICURS is saved before it is used and restored afterwards, but the content of OCHAR will be the value of the 'A' register that was passed on entry. The entry addr is $FB32.

PREFACE

Tugbug V1.0 is a system designed operating monitor for the Microtan 65 computer capable of complex functions to govern the activiies of system operation. The Monitor program is contained in a 2K byte eprom which is located within the system memory map at $F800 - $FFFF.
The Tugbug monitor allows a high percentage of compatability with software or firmware packages already existing for the Microtan 65 system and those generated by the Tanbug series of monitors. As a result duplication of the Microtan/Tanbug manuals are unnecessary in this instance as Tugbug is compatable with all functions of the Tanbg V3.1 series with the exception of:-

1. The WFOR (Warm Forth) command which was felt to be unnecessary due to the integral warm start already existing within Forth 'G404.

2. Likewise the AUX command which is simply replaced with a 'GE800 call.

3. The Keypad routine has also been removed as it was felt that this routine was little used with the exception of game playing or single board users.

The removal of these little used routines has allowed us to bring you a comprehensive Monitor for the Microtan system based upon the experience gathered from extended periods of operational use.

This manual contains the necessary information on the Tugbug monitor with the exception of that already found within the Microtan 65 manuals.

Eprom Storage Card Routines

Tugbug supports up to eight Storage modules including the 128K version, Combo Modules or any combination of either.
The syntax for data transfer is as follows:-

READ MODE.

E.S.C./Combo

<R><EPROM START>,<EPROM END>,<CARD NUMBER>,<RAM START><CR>

R - Read Mode.

EPROM START - Start address on the E.S.C. of the data to be transfered.

EPROM END - End address on the E.S.C. of the data to be transfered.

CARD NUMBER - See below.

RAM START - Address in system ram to which the data is to be transfered to, i.e.

R0000,1FFF,1,400<CR>

Will transfer the first 8K of data from the E.S.C. NO 1. starting at the address $0000 ending at address $1FFF into system ram starting at address $0400.

The same syntax being used for the Combo Module 6116 Read Only.

128K E.S.C.

When using the 128K E.S.C. the syntax remains the same with the exception of the selection for the Upper and Lower banks of eproms. This is achieved by prefixing the 'Card number' by a 1 or a 2, being 1 = Lower bank, 2 = Upper bank. Therefore to load data from the top bank of the 128K E.S.C. the command would be:-

R0000,1FFF,21,400<CR>

For the lower bank the syntax would be:-

R0000,1FFF,11,400<CR>

WRITE MODE.

Combo Module Only

<W><6116 START>,<6116 END>,<CARD NUMBER>,<RAM START><CR>

W0,100,3,400<CR>

This will write 100 bytes from location $0400 into the first 100 bytes of the Combo module.

Note:- If data is written to an E.S.C. module in error the data will be lost. Eproms cannot read!.

Video Module Driver Routine

Output to the Video 80/82 Module can be via $FBEB or directly through a jump vector at location $F82F. The $F82F method disregards the flag in byte $C and passes a byte using the following code:-

```
SENCHR: BIT VDUSTA ; CARD STATUS
....... BPL SENCHR ; JUMP IF NOT READY
....... STA VDUCTL ; PASS BYTE TO CARD
....... RTS ...... ; RETURN TO CALLER
```

It can be seen from the above code that the byte to be output to the video card is passed to this routine in the "A" register.

Cursor control is upto the user.

Keyboard Toggle.

As the other output devices can be turned on and off from the keyboard using <^S>, <^V> and <^P> the video card has been given the same facility. This is <^^> and is used as a toggle just like the others and uses BIT 3 in BYTE $C. (If you haven't realised already a "^" is the short way to describe a key that is shifted with the CONTROL key on the keyboard).

Control Characters.

Any control character with the exception of <^P>, <^S>,<^V> (if control characters are allowed) and <^L> and <CR> will put Tugbug into a mode where it is "talking" only to the video card. This enables the user to set up the video card as required directly from Tugbug before starting a program. The exit from this mode is effected by a <CR>. This will return to Tugbug command mode.

Graphics.

Whenever the command mode is entered in Tugbug eg: whenever Tugbug's cursor shows it is waiting for input, the screen will be turned on and graphics mode will be reset. It seems pointless for a program or user to return to Tugbug with either of the above conditions in force especially if a breakpoint is reached when in graphics mode.

ICHAR

The routine which waits for a character from the keyboard (JPLKB) will now return the character in the "A" register as well as the interupt routine placing it in ICHAR ($1). This will save the user loading location $1 after a call to the input routine.

Breakpoints

The breakpoint routine has been altered to give the identity of the register immediately before the display of it's contents:-

XXXX P=00 S=FF X=00 A=00

Where XXXX is the address and th rest is self explanitory.

Disk System

If a program is written that requires to know if the TANDOS disk system is present, there is a routine which will provide that information. It uses the "x" register and the user will have to save it if it's contents are required. The address is $F835 and the conditional jump the user requires to execute is BNE. This will jump if there are no disks. "X" is non zero on retern if there are no disks in the system. This means that a user without a disk system can still use Tugbug which automatically assumes the correct mode of working.

List Command

The delay that was present in the "L" (list) command has been taken out, and a screen of information will now appear on the Microtan screen and will appear on the video card subject to it's handling of the output.

Lower Case Control (NOT APPLICABLE IN LATER VERSIONS)

Tugbug allows lower case characters typed in from the keyboard, whilst in Tugbug command mode, to be interpreted as Upper Case characters thus avoiding the need for the 'Caps Lock' facility of the keyboard to be engaged. This facility does not inhibit lower case character control during user programs or the Basic intepreter.