

SYSTEM SOFTWARE  
BULLETIN NO. 26  
JANUARY 5, 1983

TO: Distribution

SUBJECT: BLUE WHALE DEVELOPMENT SYSTEM ON VAX

This documentation contains description of the operational commands and system commands for Blue Whale Development Station on VAX system.

Programmers: Chan Min, John Spies, Jon Yin

DISTRIBUTION

Hugh Barnes 728-12  
Gabriel Baum 729-12  
Ron Carlson 726-12  
Dave Chandler 729-12  
Don Daglow 729-12  
Richard Decker 726-12  
Bill Fisher 729-12  
Russ Haft 729-12  
Jim Haupt 726-12  
Bob Hogue 726-12  
Les Hutchinson 728-12  
George Jump 729-12

Don Mills 728-12  
Mike Minkoff 729-12  
Tom O'Brien 729-12  
Larry Pumphrey 724-12  
Keith Robinson 724-12  
Chuck Rudd 726-12  
Tim Scanlan c/o R. Timmins 729-12  
Joey Silvian 729-12  
Ron Surratt 729-12  
Rick Timmins 729-12  
Mark Urbaniec 729-12  
Mike Winas 729-12  
System Software Group

## List of commands for the Blue Whale Development System on VAX

### 1. Commands to VAX

The character "%" is the VAX prompt. The addresses are in Hex-ASCII format.

#### 1.1 Blue Whale Format Command

```
% blue file1 > file2
```

Converts the output file file1 from the linker to the Blue Whale format, and directs it to file2.

#### 1.2 Download Command

```
% whale [infile]
```

Downloads infile in Blue Whale format to the Blue Whale. If infile not specified, it directs the video terminal to the Blue Whale.

#### 1.3 Upload Command

```
% upwhale outfile addr1>addr2
```

Uploads the contents of the memory location from addr1 thru addr2 to outfile in Blue Whale format.

### 2. Commands to Blue Whale

The character ">" is the Blue Whale prompt. All addresses and data in Hex-ASCII format. The letters representing commands are in upper case. The brackets "[]" represent for optional parameters.

#### 2.1 Go and Execution Command

```
>[addr]G
```

Starts execution from location addr. If addr not specified starts at location (1041)H to start cartridge execution. This command ends with a carriage return.

```
>P
```

## 2.1 (continued)

Starts execution from present program location. The P command ends with a Carriage Return

## 2.2 Single Step Command

>[addr]T

Single step starts at the location addr. If addr not specified, starts at present program location.

## 2.3 Memory Manipulation Command

>[addr]M

Displays 8 consecutive memory location contents starting from the address addr. If addr not specified, assumes the current memory pointer.

>addr/datal data2

Alters the memory location addr's content datal for data2. This command terminates with a Line Feed or Carriage Return.

>addr1<addr2,addr3\$M

Moves the content of memory location addr2 to addr1, then increments addr1 and addr2, repeats the move until addr1> addr3. This command can be used to fill memory with a pattern of data. First, store the desired data into addr2. Then let addr1=addr2+1 and do the above command. Memory from addr1 through addr3+1 will be filled with data. In a similar fashion, multiple decles can also fill memory.

>addr1,addr2V

Displays the Data-IO checksums (low and high) and IMI checksum of the memory location addr1 to addr2.

>data,addr1,addr2S

Searches memory location with content data from addr1 to addr2, and displays all matching addresses. If data is specified with one or two digits, only the low 8 bits of the 16 or 10 bit memory word is searched for. If 3 or 4 digits are specified, all 16 bits are checked. Control-S can be used to stop the display.

>ZERO

Zeroes the memory locations 5000-5FFF, 6000-6FFF, 9000-9FFF. Ends with Carriage Return.

#### 2.4 Breakpoint Command

>addr\$B

Sets the address addr as a breakpoint. One can set up to eight breakpoints.

>addr\$R

Resets the breakpoint addr.

>\$D

Displays all current breakpoint addresses.

>\$C

Clears all current breakpoints.

MISCELLANEOUS - These commands are used automatically by system programs and are not needed by most people using the whale to write games.

>addrGO<return>

This command causes the 6502 whale code to branch to a new 6502 address. This is used when testing new versions of whale code with blowing more EPROMS.

>addr;

Displays the byte addressed using 6502 address. Is otherwise like '/'.

>addr1,addr2,countU

Uploads data records starting from addr1 and continuing through addr2. Each record has a length of 'count' 16 bit words. Each record begins with a : character. It takes four characters per 16 bit word. The low four bits of each character contains the data. Each character can range from @ABCDEFGHIJKLMNO. Then checksum results from adding up all the 16 bit words and ignoring overflow. The whale will wait for an ack or nak character.

If ack (control-F) is received, the next record is transmitted. If nak (control-U) is received, the previous record is re-transmitted. This command is used for uploading MR. COLOR files.

>:

Download data to the whale. The next character determines if the downloaded data is in 1610 format or 6502 format. The 6502 format is standard Intel format. Whale format is of the form:

:aaaadd.....;c

The first four characters is the 16 bit address. Successive character pairs form succeeding decles. The ; finishes the data record and a 5 bit checksum follows. Notice that a 5 bit checksum is not adequate for transmission over error prone lines, like telephone lines. If you get even one BAD RECORD message while downloading, something should be fixed. Data is held in the low 5 bits of each character. The whale returns control-f, carriage return, line feed, > if the record was received correctly. Otherwise, control-U is returned.

Bringing up new versions of the whale:

To test out a new version of whale, first use MICEV to download the 6502 version. Do a 20BEGO<return> to start the new 6502 code. This address should be label NXCOM. Then use ORCA to download the new 1610 code and start it with BO61G<return>. The new 6502 code module is called BCOM. The new 1610 code is called DDBBGG.

There are several things that must be changed from the debugging version of the whale firmware before burning a set of EPROMS.

1. Modify module B1610.ASM to not JSR DNLOAD.
2. Modify module BEQAT.ASM to start 1610 code at \$8800 instead of \$B000.
3. Modify module BEXEC.ASM to clear memory to \$C000 instead of only to \$A000. This step is usually skipped to allow loading new versions of whale into locations \$A000-\$AFFF new 6502 code and \$B000-\$BFFF (new 1610 whale code).
4. Change BA6502.CC1 to load program into \$E000 instead of \$2000.
5. Change DDBBGG.CCL to load program into \$8800 instead of \$B000.