VG5000 BIOS SECTION

Within the BASIC ROM of the VG5000 there are supplied a number of jump vectors at fixed addresses which give access to a number of useful routines which are described as follows:-

address (hex) name description

0000 reset

Restart BASIC from the power up condition. Initializes all variables and memory and I/O.
entry none
exit none
modifies all

0008 chkchr

Used only as an RST instruction. Is used to verify that the character at the text pointer position (registers HL) is the character that immediately follows the RST 8 instruction otherwise a BASIC syntax error is printed.
entry HL=text position
exit drops into chget
modifies AF,HL

0010 chget

gets next character into register A from next text pointer position (HL+1) Condition codes are set to reflect the kind of character found.
entry HL=text position
exit A=character code
C flag set if numeric
Z flag set if end-of-line
modifies all

0018 outdo

Character output routine. Character in register A is printed to the output device whose code is given by the contents of PRTFLG (see memory map).
entry A=character code
exit none
~~modifies AF~~

001B setext

Program the shape of a programmable character cell in video memory. The character code of the character shape to be programmed is in the least significant 7 bits of register A and the most significant bit signals extension text characters if reset and extension graphic characters if set. HL points to the start of a 10 byte buffer which is the character shape data, the first byte in the buffer being the top 8 piccells of the character shape, etc.
entry A=character code. Bit 7=0 is text.
HL=start of character shape data buffer

```
                          exit       none
                          modifies   all

0020      dcompr         Compares the 2 register pairs HL and DE
                         without modifying either.
                          entry      HL and DE=registers to be compared
                          exit       Z flag set if equal.
                                     C flag set if DE greater than HL
                          modifies   AF

0030      usrrst         RST location available to be reprogrammed by
                         user.  Jumps to a hook in RAM (see memory
                         listing).
                          entry      none
                          exit       none
                          modifies   none

0033      warmgo         A power up of BASIC without destroying the
                         contents of memory.  Same point to which the
                         keyboard reset key will go.
                          entry      none
                          exit       none
                          modifies   all

0080      retfun         Position to which a user function will jump on
                         completion in order to make the user function
                         assignment equal to the contents of register
                         A.
                          entry      A=data to return as a result of function
                                       call
                                     HL=BASIC text pointer
                          exit       none
                          modifies   all

0083      deint          Pick up parameter within a user function from
                         parameter field and leave in registers DE.
                          entry      HL=BASIC text pointer
                          exit       DE=evaluation of user parameter
                          modifies   All

0086      getbyt         Pick up an 8 bit parameter from the text
                         position at HL and leave in register A
                          entry      HL=BASIC text pointer
                          exit       A=evaluation of user parameter
                          modifies   all

0089      frmnum         Pick up a 16 bit signed parameter from the
                         text position at HL and leave in the floating
                         point accumulator.  If the result is required
                         in registers DE, call "deint" next.
                          entry      HL=BASIC text pointer
                          exit       FAC=evaluation of user parameter
                          modifies   all

008C      beep           Make a sound via the sound channel of duration
                         given by registers DE and mark/space time given
```

by the contents of SONSAV (see memory listing)

entry DE=duration of sound
(SONSAV)=mark time of note
(SONSAV+1)=space time of note
exit none
modifies all

008F play

Play a sequence of ASCII coded notes according to the specification of the BASIC PLAY command. Registers BC point~~s~~ to the start of a sequence of ASCII coded bytes which must follow the exact format of the PLAY command. For example, if the PLAY subcommand "T100" is to be actioned, the play buffer will be &"54",&"31",&"30",&"30" and not &"54",&"64". Register E is the length of the buffer.

entry BC=start of ASCII buffer coded in PLAY format
E=number of bytes in the buffer
exit none
modifies all

0092 putahl

Write a 16 bit character code pair directly to the video memory at cursor position given by the registers HL. This is an alternative method of communication to that which BASIC normally uses. Instead of characters going via a video buffer in Z80 memory a character is displayed immediately at the specified cursor position.

entry H=cursor Y position. 0=top line
8-31=lines 1-24
L=cursor X position. 0-39=column 0-39
D=character code of displayed character
E=attribute code of displayed character
exit none
modifies BC and AF

0095 putici

Write a 16 bit character code pair directly to the video memory at the current cursor position.

entry D=character code of displayed character
E=attribute code of displayed character
exit none
modifies BC and AF

0098 getahl

Get a 16 bit character code pair directly from the video memory at cursor address given by register HL, into registers DE.

entry H=cursor Y position. 0=top line
8-31=lines 1-24
L=cursor X position. 0-39=column 0-39
exit D=character code of displayed character
E=attribute code of displayed character
modifies DE, BC and AF

009B getici

Get a 16 bit character code pair directly from the video memory at the current cursor

position into registers DE.
entry   none
exit   D=character code of displayed character
           E=attribute code of displayed character
modifies   DE, BC and AF

009E   cls   Clear internal screen map so that each line is the same as that found in the description of the clear line routine below.
entry   (ATTBAK) = background colour
           (ATTCAR) = character colour
exit   none
modifies   all

if ATTCAR = text and 00 hex
if ATTCAR = graphics

00A1   cll   Clear a line of the internal screen map so that at the start of the line there is a delimiter character whose attribute is found in the contents of (ATTBAK). ~~and~~ The character code programmed is 20 hex with an attribute found in the contents of (ATTCAR).
entry   (ATTBAK) = background colour
           (ATTCAR) = character colour
           A = line no. to be cleared
exit   none
modifies   all

00A4   wait   Halt program continuation until video chip is available for data transfer.
entry   none
exit   none
modifies   af

00A7   fndadr   Find physical address in video screen map corresponding to the X,Y cursor position in register pair HL.
entry   H=Y cursor position
           L=X cursor position
exit   HL = address in video screen map area
modifies   all

00AA   kbscan   Scan keyboard matrix and produce a code for a depressed key in the A register.
entry   none
exit   A=ASCII code produced by depressed key, else 0
modifies   all

00AD   regset   Loads the registers of the video processor. The table is pointed to by the register pair HL. The first byte in the table is the count of the number of data elements in the table. The data elements in the table are grouped in pairs and the first byte is the register select address and the second byte, the data to that register.
entry   HL=table pointer
exit   ~~none~~ B= 0

```
                   modifies  HL,BC,AF

0OB0    stkv      Scan  the joysticks/cursor keys to examine  if
                  the  vertical  axis  is  active.  The  choice
                  between  joysticks and cursor keys is made  by
                  entering  with a parameter in the  accumulator
                  and  on  exit the  accumulator  indicates  the
                  vertical status.
                   entry      A=0 for left joystick
                              A=1 for right joystick
                              A=2 for cursor keys
                   exit       A=0 for no action
                              A=1 for down action
                              A=255 for up action
                   modifies  AF,B

0OB3    stkh      Scan  the joysticks/cursor keys to examine  if
                  the horizontal axis  is active.
                  entry      A=0 for left joystick
                              A=1 for right joystick
                              A=2 for cursor keys
                   exit       A=0 for no action
                              A=1 for right action
                              A=255 for left action
                   modifies  AF,B

0OB6    stka      Scan  the  joystick action keys/space  bar  to
                  examine  if  the  action  button  is  active.
                   entry      A=0 for left joystick
                              A=1 for right joystick
                              A=2 for cursor keys
                   exit       A=0 for no action
                              A=1 for action button
                   modifies  AF,B

0OB9    break     Scan keyboard for break sequence (shift/stop).
                   entry      none
                   exit       A=1 if stop not pressed
                              A=4 if stop pressed but shift not pressed
                              A=0 if both pressed, C flag is set
```