Digisolve
Micro electronic design consultancy.
Aire & Calder Works, Cinder Lane, Castleford,
West Yorkshire WF10 1LU
Telephone: (0977) 513141/4 or 513382
Telex: 557661 AGRAM

VGP][
Dual standard
VECTOR GRAPHICS PROCESSOR
for
APPLE][COMPUTERS

USERS MANUAL.

Manufactured and designed by:

DIGISOLVE LTD.
AIRE AND CALDER WORKS,
CINDER LANE,
CASTLEFORD,
WEST YORKSHIRE,
ENGLAND. WF10 1LU

Telephone: (0977) 513141
(0977) 513382
(0977) 510511

Telex: 557661 AGRAM G

Contents.
---

1) Introduction.

Thank you for purchasing a Digisolve Graphic Processor for your computer. You are now entering the world of professional high resolution computer graphics. Not long ago high resolution computer graphics was the dream of many computer owners being reserved for educational establishments with special government grants or high powered industry. Now thanks to new advances in semi conductor technology Digisolve is able to bring high resolution graphics to your computer at a fraction of the cost of other units. Advances are still being made in the field of computer graphics and Digisolve is committed to keep you up with these advances. Please fill in and return the newsletter application form enclosed with this manual to receive regular updates on new graphics technology from Digisolve.

Your VGP][card is the result of 2 years of development work carried out in the United Kingdom by Digisolve. The VGP][is the first product in the second generation of graphics peripherals from Digisolve and was designed after lengthy consultations between Digisolve, software developers and end users of Digisolve's first generation of graphics products. Many new features have been incorporated into the VGP][card to meet the requirements of as wide a range of graphics users as possible.

VGP][Main Features.

512 x 512 resolution graphics display, 512 x 416 resolution in USA mode.

64k of onboard memory.

Vector drawing at up to 1400000 pixels per second.

Hardware character generator, up to 85 characters by 64 lines.

8 write modes.

Hardware PAN and SCROLL.

Programmable border colours.

2 screen images stored at the same time.

Applesoft, Pascal, Assembler and TASC software support.

Printer dump and Disk save utilities.

Colour expansion capability.

Video synch lock option for video recording.

Video soft switch.

Fully Apple][, Apple][e and Apple][+ compatible.

2) Unpacking and installation.

Carefully remove all the items supplied to you from their packing. You should have the following:

One Digisolve VGP][card.

One Video redirection lead.

One Applesoft support disk.

One TASC support disk.

One Pascal support disk.

This manual.

If any of these items are missing then contact your dealer for a replacement.

Now inspect the VGP][card and associated parts for any shipping damage. If the card is damaged then return the complete VGP][package to your dealer for replacement. Do not attempt to use the card if you are in any doubt about its condition.

During the following installation procedure for the VGP][card the Apple computer must be switched OFF. Any attempt to install the VGP][when the Apple is switched ON will possibly damage the VGP][card and/or the Apple computer. Any attempt to install the VGP][with the Apple powered up will INVALIDATE THE WARRANTY on the VGP][card.

The first stage of the installation procedure is to attach the video lead to the VGP][card itself. The video lead has a brown four way connector at one of its ends and two phono connectors at the other end. The end with the brown four way connector plugs into the VGP][card at its narrow end. Examine the four way connector on the video lead to find the side which has four thin rectangular holes. This side must face out from the VGP][card when the cable has been plugged in. Plug the video cable into the card and make sure that the orientation is correct.

Now remove the cover from your computer and find a free slot for the VGP][card. It can be plugged into slots 2,3,4 or 5. Digisolve strongly recommend that slot 4 be used for the card if at all possible as many software packages which run with the VGP][card require that this slot be used. Carefully route the video cable alongside the VGP][card and out of the rear of the computer.

There should now be 2 separate cables which form the video cable assembly running out of the back of the computer. One of these cables has a phono plug on its end and the other a phono socket. The cable with the phono plug on the end plugs into the video out connector of the Apple and the cable with the phono socket on its end has the cable which connects to your video monitor (the one which was plugged into the Apple) plugged into it.

The VGP][card, like many 80 column cards, has a video switch circuit built into it. This allows one monitor to show either the 40 column Apple video or the graphics video from the VGP][card. Whenever the computer is 'RESET', the Apple's video output is selected.

If you wish to use two separate monitors then simply connect the VGP monitor to the lead which has the phono SOCKET on its end. You will usually need a phono plug to phono plug lead to do this.

Now replace the lid to the computer taking care not to trap any wires which may connect your Apple to any peripheral devices when doing so. Switch the computer on and boot DOS from the usual systems disk. If you do not get a display on the monitor then the most likely cause of problems is that the video cable has been installed incorrectly. Check its installation and try the system again.

Suitable Video monitors.

For best results Digisolve recommend the use of a P39 long persistance video monitor with at least 12MHz video bandwidth. We have found that the Apple /// monitor gives acceptable results although many independant manufacturers of video monitors can also offer products which will work with the VGP][card.

If you use an ordinary green or white phosphor short persistance monitor the the card will still work perfectly, however single horizontal lines may seem to flicker. This effect is due to the use of interlaced video to display horizontal resolutions of greater than 256 pixels. If you find this flicker is unacceptable then we suggest that you either purchase a long persistance video monitor or operate the VGP][card in one of the non-interlaced modes which do not show flicker with any display phosphor. As a general rule flickering can be kept to a minimum by keeping the brightness and contrast levels of the monitor to the lowest possible levels, and not operating the VGP][card in a room which has been lit with fluorescent lighting.

Video Synch and other options.

The Video synch option allows the VGP][card to be locked to an external synch source to allow video overlays. This option consists of a separate card which can plug into any free slot in your Apple computer, and which plugs into an external 1v p-p composite synch source and the VGP][card itself. The main use of the video synch add on is in the television studio where the VGP][card can be used to generate video overlays or titles for images being generated by a television camera.

The second add on product in the VGP][family is the VGP][colour expander card. This card plugs into the slot to the right of the VGP][card and upgrades the VGP]['s operation to display each pixel in one of 8 colours. This card can drive any colour monitor with separate R,G,B and SYNCH inputs or drive a monochrome monitor in eight levels of intensity.

Many software packages are being prepared or are already available for the VGP][card. Applications already available include Architecture, Design, Graphics Art, Thermal modelling and PCB design.

3) Unpacking and installation - colour expansion.

Digisolve's VGP][colour expander allows the operation of the VGP][graphics processor peripheral for the Apple][computer to be expanded for operation in 8 colours. The VGP][colour expander package consists of a VGP][colour expander card with 128k bytes of screen memory, a monochrome video lead and a colour video lead. The colour expander card occupies the slot to the right of the VGP][card in the Apple computer and plugs into the VGP][card. It can drive any colour monitor with separate R G B and synch inputs at either TTL or analogue 1v p-p levels, or a monochrome monitor in 8 grey scale levels.

As with the installation of the VGP][card it is most important that the power to your computer is switched OFF during this installation procedure. Switch your Apple][off and remove the top cover. Locate the slot in which your VGP][card has been installed and make sure that the slot to its right is also free for use. If this slot is not available then you must re-arrange your cards so that you have two adjacent free slots 2-3,3-4, or 4-5. Remove the VGP][card from the Apple and disconnect the video lead from it.

Check the VGP][expander card for any shipping damage. If you are in any doubt as to the condition of the VGP][expander, return it to your dealer, do not try to use it. The VGP][card has a 26 way connector close to its upper edge. The rear of the VGP][expander also has a 26 way connector. Line up the connectors and gently push the two cards together.

For grey scale monochrome operation, take the monochrome video lead, which has a 2 pin connector on one end and a phono plug on the other end, and plug it into the two pin connector on the VGP][expander.

For colour operation, plug the colour video cable into one of the two eight way connectors at the narrow end of the expander card. The upper connector generates analogue 1v p-p video signals and the lower connector generates TTL level signals.

Replace the VGP][card and the VGP][expander into the Apple system and route the video cable out of the back of the computer. Replace the top cover of the computer. Connect up the Apple's monitor to the Apple video output and the other monitor to the output from the expander card.

The pin out of the 9 way cannon socket that is on the other end of the colour video cable from the VGP][card is given below:

```
  RED - 1 o
                o 6 - ground
GREEN - 2 o
                o 7 - ground
 BLUE - 3 o
                o 8 - n/c
SYNCH - 4 o
                o 9 - n/c
  n/c - 5 o
```

4.) General Information and Terminology.

Your VGP card acts very much like a sophisticated pen plotter with drawings appearing on a video monitor instead of a sheet of paper. You can draw lines move to points and plot characters.

Applesoft support provides simple commands that make all of these operations easy from Applesoft. The source code of this support is also supplied so that you can add new commands of your own or use these routines from machine code.

Unlike a real pen plotter your VGP][can store two screens full of information and you can choose which one that you would like to see on the monitor. This enables sophisticated animation effects to be created since images can be created on the screen which is not being displayed and then screens switched so that complete new images appear with no signs of the drawing process.

In a real pen plotter it is not possible to erase a line once it has been drawn. Your VGP allows lines to write and erase screen information as well as to invert data on the screen (useful for drawing non destructive cursors.)

The VGP card has a resolution of 512 x 512 points, any one of which may be on or off. Point 0,0 is in the bottom left hand corner of the screen, point 511,511 is in the top right hand corner. Your VGP card can not only draw lines and points but also characters of varying size and orientation. The maximum number of characters that can be displayed on the screen is 85 characters by 57 rows (with descenders) or 85 characters by 64 rows if lower case is not required.

Above all the Vector Processor is very fast. It can draw characters and lines at up to 1400000 pixels per second, and does this in parallel with the Apple's processor. This means that whilst the VGP is drawing a line the Apple can be calculating the co-ordinates of the next line to be drawn.

In this manual X and Y co-ordinates of points and lines are often referred to. The VGP's X axis is from left to right and the Y axis from bottom to top. Thus a line drawn from X=0,Y=0 to X=511, Y=511 would be a 45 degree diagonal from the bottom left hand corner to the top right hand corner.

The VGP]['s image space can be viewed as 1024 pixels across by 512 pixels deep. When the first page is displayed the left hand side 512 pixels X = 0 to X = 511 are shown on the screen. When the second page is displayed the right hand side of the image area X = 512 to X = 1023 is displayed on the screen.

The VGP][can also PAN and SCROLL the graphics image on the screen. This feature has been provided to allow scrolling of text and fast movement of graphics images. A PAN movement is the sideways shift of the screen image, a SCROLLING movement is the vertical movement of the screen image.

The VGP][colour expander system expands the operation of the VGP][to 8 colours. In this mode each pixel may be individually set to any one of the eight possible colours which are available with the expander. The support software has been written to operate with the VGP][card on its own in monochrome, and with the VGP][card plus Colour expander card in colour.

5.) Applesoft support.

With your vector processor comes an Applesoft support disk. This contains a program in machine code that adds extra commands to standard Applesoft which allow you to use the VGPJI in a very easy way.

Should you not wish to use Applesoft for your programs then you may use the supplied source code of the Applesoft support as the basis of a machine code program to drive the card, or use the TASC or PASCAL support which is provided with the VGPJI card.

To get Applesoft support to work you must boot from the normal DOS diskette. Place the Applesoft support disk into drive 1 then type 'RUN HELLO'. The program will run and ask for the slot number in which the VGPJI card is installed. Type the slot number followed by <RETURN>. the program will load in the Applesoft support.

If the loading operation is successful then you will see the usual Applesoft flashing cursor on the screen. If you do not get to this stage then consult your dealer or Digisolve direct. Digisolve have encountered some hardware configurations (such as hard disks) that are not directly compatible with the VGPJI support. If this is the case then special versions of the support software may be available for your particular hardware configuration. Digisolve will make every effort to give you a version of the support software which will work with your system.

When the support is loaded it initialises the VGPJI card, monochrome operation is assumed, both screens are cleared, and display of screen one, pen, pen down and normal write mode are selected. The video switch is set to display Apple video, the display format is set to USA 60Hz 525 lines with a display resolution of 512 x 416 pixels and the border colour is set to black. This allows drawing to start right away.

You are now in a position to try the demonstration program on the support disk. Type 'RUN DEMO'. The program asks if the colour expander is fitted. Type Y or N. The correct operating mode is selected and then a series of programs are run. These all use Applesoft support and are not protected in any way so that you may look at them and see examples of the support commands in operation.

6.) Applesoft Support - commands.

Applesoft support provides you with new commands that operate only with Digisolve's VGP][card. All of these commands start with a & (the ampersand character), followed by the function name and usually one or more parameters. If you use another system on the Apple, which also uses the & character to implement extensions, it will not operate with the VGP][support. In this case, contact Digisolve direct for advice on how to use the graphics system.

&FOR - select VGP][format.

This command is usually the first command issued to the support. It specifies the video format (either USA 60Hz 525 line or UK 50Hz 625 line) and if a colour expander card has been fitted.

e.g. &FOR F,C

If F is 0 then USA video format is selected, if F is 1 then UK video format is selected. If the VGP][is being operated without the expander, then C should be 0, otherwise C should be 1.

&INT - select interlace mode.

The VGP][uses interlaced video to generate the full resolution displays. The VGP][can be set to display non-interlaced video, which is usually used to eliminate flicker on short-persistence monitors.

e.g. &INT I

If I is 0 then non-interlaced video is generated, if I is 1 then interlaced video is generated.

&S - set the subpage for non-interlaced mode.

If the VGP][has been set to display non-interlaced video then it only uses half of its onboard memory to hold the two screen images. This command lets you specify which of the two halves of the display memory should be used, thus allowing four screen pages to be stored.

e.g. &S 0 select screen page set 1
&S 1 select screen page set 2

&NEW - re-initialise the VGP][.

This command resets all the VGP][settings except for format, colour and interlace (see above commands). Both screens are cleared, screen one is selected together with pen, pen down and normal write (white) mode.

e.g. &NEW

&PLOT - plot a line.

This command is used to draw a line from any point to any other point, or to draw a line from the last point drawn to a new point.

e.g. &PLOT X1,Y1 TO X2,Y2 would draw a line between 2 points

The above statement may be continued as follows to chain between points.

e.g. &PLOT X1,Y1 TO X2,Y2 TO X3,Y3 TO X4,Y4.....

Lines may also be drawn from the current screen position to a new point as follows.

e.g. &PLOT TO X6,Y6

The following examples are also valid uses of the &PLOT command.

```
&PLOT A*3,B TO C*4,D-1 TO F,H TO Y,Z
&PLOT TO 121,161 TO A,C TO B,1.08
&PLOT 1,1 TO 100,108
```

&M - move to a point.

This command is used to move to a new screen position without drawing onto the screen.

e.g. &M 0,0 would move to the bottom left hand corner of the screen without drawing a line.

&U - lift pen/eraser.

This command takes no parameters and is used to tell the VGP to lift its pen. Commands that plot which follow this command will seem to do nothing since the pen has been lifted. The screen position will be updated as usual however.

e.g. &U would lift the pen or eraser (suspend its action).

&D - lower pen/eraser.

This command is used to lower the VGP's pen or eraser so that drawing commands which follow it will have an effect on the screen.

e.g. &D would lower the pen or eraser.

&P - select pen.

This command is used to tell the VGP to use its pen. Lines drawn onto the screen with the pen down will appear on the screen.

```
e.g.                &P        select pen.
```

&E - select eraser.

This command tells the VGP to use its eraser instead of the pen. Lines drawn with the eraser down will erase data from the screen.

```
e.g.                &E        select eraser.
```

&CLEAR - clear the screen.

This command clears the screen but does not change the current screen position. If you want to clear screen one then it is imperative that the current point is in the first screen area, and that if you wish to clear the second screen that the current point is in the second screen.

```
e.g.                &CLEAR      clear the screen.
                    &M 0,0 : &CLEAR     clear screen 1
                    &M 512,0 : &CLEAR   clear screen 2
```

&HOME - set current position to 0,0.

This moves the current screen position to 0,0 the bottom left hand corner.

```
e.g.                &HOME
```

&FLASH - fill the screen.

This command is used to set every pixel on the screen on. It does not effect the current screen position.

```
e.g.                &FLASH    fill the screen every pixel will be on.
```

&DRAW - select line type.

This command is used to tell the vector processor which type of line it should draw. The vector processor knows about four types of lines; solid, dotted, dashed and dotted - dashed.

```
e.g.                &DRAW 0   select solid lines
                    &DRAW 1   select dotted lines
                    &DRAW 2   select dashed lines
                    &DRAW 3   select dotted dashed lines
```

&HGR - select screen page 1.

This command selects the first screen page for display. (X = 0 to 511).

e.g. &HGR select page 1

&HGR2 - select screen page 2.

This command selects the second screen page for display. (X = 512 to 1023).

e.g. &HGR2 select page 2

&TEXT - select character type.

This command tells the VGP what type and size of characters will be used when the &PRINT command is used. &TEXT must be followed by four parameters. The first two describe the orientation of the characters and the second two the X and Y scaling factor for the characters.

e.g. &TEXT A,B,C,D

```
if A=0 then the text will be normal
if A=1 then the text will be italic
if B=0 then the text will be on the X axis
if B=1 then the text will be on the Y axis

C is the X scaling factor
D is the Y scaling factor
```

The scaling factors for the characters should be in the range 1-16 where 1 gives the smallest characters and 16 the largest. If zero is used for either of these parameters then this will also result in the largest size of character.

The following examples are also valid uses of the &TEXT command.

```
&TEXT 0,0,1,1   the smallest characters are selected
&TEXT 0,0,1,2   the characters which could be used to give an 80*24 display
                are selected.
&TEXT X%,Y%,Q,2
```

If characters are drawn vertically then they appear to the LEFT of the start point co-ordinates. Thus &M 0,0:&TEXT 0,1,2,2:&PRINT "CAN'T SEE ME" would seem to do nothing since the characters are being drawn to the left of the current position , that is, off the left of the screen.

&PRINT or &? - print text.

This command is used to print characters onto the screen in the size and style that have been specified in the &TEXT command.The VGP knows the full 96 character upper and lower case character set.

e.g. &PRINT "text to be printed" would print 'text to be printed' at the current screen position.

&READ - read the current screen position.

This command is used to find out the current screen position. After a &PLOT or any other command the VGP knows the last point that was drawn or moved to. This command reads the co-ordinate of that point. The command is followed by two variables which will contain the X and Y positions of the current position after the command. These two variables must be integers.

e.g. &READ X%,Y% reads the current position and sets X% and Y% accordingly.

&AT - read screen colour.

This command is used to return the colour of a specific screen pixel. It is followed by three parameters the first two of which are the co-ordinates of the pixel to be tested, the last being an integer which will be set to the colour of the pixel.

e.g. &AT X,Y,P%

This will read the pixel at X,Y and set P% to the colour of that pixel. If the colour expander is not fitted, the value returned will be 0 if the pixel is on (white) or 1 if the pixel is off (black). A colour system returns a value from 0 to 7.

&C - select cyclic mode.

This command tells the VGP that lines drawn off the right hand side of the screen or the top of the screen should not be clipped and be displayed by wrapping them round to the other side of the screen.

e.g. &C set cyclic mode

&D - set clipping mode.

This command is used to tell the VGP that it should clip lines that are drawn off the top or right hand side of the screen.

e.g. &D select cliping mode.

&INVERSE - select inversion mode.

This command tells the VGP that when it writes information to the screen it should be the inverse of the existing screen data. This facility allows for non destructive cursors to be drawn onto the screen.

e.g. &INVERSE select inversion

&NORMAL - select normal writing mode.

This command tells the VGP to write information to the screen with out taking into account existing screen data.

e.g. &NORMAL select normal mode.

&POS - draw a cursor.

This command causes a non destructive inverted cross hair cursor at the current screen position to be drawn. This cursor will be of the size specified by the &STORE command.

e.g. &POS draw a cursor.

In order to remove the cursor and return the screen image to its previous state the &POS command should be used again.

&STORE - set cursor size.

This command is used to specify the size of the cursor drawn by the &POS command. Valid cursor sizes are from 0-255.

e.g. &STORE 20 select cursor size to 20

&I - invert the screen.

This command is used to invert the state of every pixel on the screen. The current screen position is not changed by this command. Every black pixel will be changed to white and every white pixel will be changed to black.

e.g. &I invert the screen

Drawing blocks.

It is possible to draw square or rectangular blocks using the vector processor. By executing &PRINT CHR$(10) a 5 by 8 pixel block scaled by the character size in the &TEXT command will be drawn at the current screen position. By executing &PRINT CHR$(11) a 4 by 4 solid block will be drawn scaled as above.

Write modes and colour commands.

The VGPJI card can be selected to draw in one of eight write modes:

Write mode	Description.	Pen draw effect.	Eraser draw effect.
0	Normal	draw	erase
1	Write protect	no effect	no effect
2	Erase regardless	erase	erase
3	Write regardless	draw	draw
4	Invert 1	invert	erase
5	Invert 2	invert	draw
6	Invert 3	invert	no effect
7	Or	draw	no effect

The VGPJI card by itself has only one memory plane, that is, each screen pixel is represented by one bit. If the colour expander is fitted, then the system has three memory planes (to give the eight different states for each pixel). The above write modes can be set independantly for each memory plane in the system. The following command is provided to allow full control over the write mode(s).

&ASC - select A Special Colour.

This command allows the specification of a particular write mode for each colour plane. It is followed by one parameter if in monochrome mode, or three parameters if in colour mode. The value of each parameter is from 0 to 7.

```
e.g.      &ASC M          in monochrome mode, set write mode M
          &ASC R,G,B      in colour mode, R is the write mode for RED,
                          G is the write mode for GREEN, B for BLUE.
```

It is obvious what this command achieves in monochrome mode, however the effect in colour mode needs explanation. To display the colour red, the red memory plane must set on, and both the green and blue memory planes must be cleared. Therefore to draw in red, the red plane write mode should be 0, and the green and blue plane write modes should be 2. This draws in an absolute colour which is not affected by the previous screen contents. If the blue and green write modes were set to 1 (write protect), the red plane would still be set on, but the green and blue planes would not be affected. This is additive colour, that is if the green or blue planes were off, red would be displayed, but if either or both planes were on, some other colour would be displayed.

The next three commands are normally used to specify colour selection. The single parameter supplied refers to the colour, rather than a write mode. Note that these commands also work in monochrome mode, only the least significant bit of the parameter is used, 0 is white and 1 is black.

&N - set absolute colour.

This command selects which colour planes are to be erased and which are to be set when drawing with the pen. If the eraser is selected after this command, all planes will be erased by drawing.

e.g.	&N 7	Black.	All planes are erased when drawn to.
	&N 6	Red.	The blue and green planes are erased and the red plane is set on if drawn.
	&N 5	Green.	
	&N 4	Blue.	
	&N 3	Yellow.	
	&N 2	Magenta.	
	&N 1	Cyan.	
	&N 0	White.	All planes are set on.

&W - set additive colour (write-protect).

This command selects which colour planes are to be write-protected and which are to be set when drawing with the pen. If the eraser is selected after this command, planes that have not been write-protected will be erased.

e.g.	&W 7	will write-protect all planes from drawing or erasure.
	&W 6	will write-protect the blue and green planes and set the red plane when drawn. A black pixel becomes red if drawn.

&R - set inverting colour.

This command selects which colour planes are to be inverted and which are to be set when drawing with the pen. If the eraser is selected after this command then all planes will be erased by drawing.

e.g.	&R 7	will select all planes to be inverted when drawn.
	&R 6	will invert the green and blue planes and set the red plane when drawn. A white pixel becomes red when drawn.

&OR - select border colour.

The VGP][can fill in the area between the edge of the display region and the edge of the screen with a solid colour. The command takes one parameter which specifies the border colour. The colour parameter is as the &N command above.

e.g.	&R 0	select border to be white

&A - select Apple video.

This command is used to set the VGP][‘s video switch to select the Apple‘s video. If you are using two monitors one for the VGP][and one to display the Apple‘s output then this command will switch the video to the VGP‘s monitor off.

e.g. &A select the Apple‘s video

&V - select VGP][video.

This command is used to set the VGP][‘s video switch to select the video from the VGP to be displayed on the monitor. If you are using two monitors then this command will switch the video to the VGP‘s monitor on.

e.g. &V select the VGP‘s video

&X - set X screen offset - PAN image.

This command is used to set an X position offset for the stored graphics screen image. The effect of this command is that the image on the screen moves to the left or right relative to the bottom left hand corner. The point 0,0 however, remains in the bottom left hand corner of the screen for new data which is to be written onto the screen. This command takes one parameter which is the number of pixels to set the offset to in multiples of 8 pixels (the smallest amount which can be moved).

e.g. &X 0 set the X offset to 0
&X 1 set the X offset to 8 pixels (image moves left 8 pixels)

&Y - set Y screen offset - SCROLL image.

Like the &X command this command is used to select an offset for the screen image. The effect of this command is to offset the image on the screen vertically by a multiple of 8 pixels relative to the bottom left hand corner. The point 0,0 however, remains in the bottom left hand corner of the screen. This command takes one parameter which is the number of 8 bits to scroll the display.

e.g. &Y 0 set the Y offset to 0
&Y 1 set the Y offset to 8 pixels (image moves down 8 pixels)

&> , &< , &+ and &- - Pan or scroll image by eight pixels.

These four commands move the image on the screen by panning or scrolling. &> moves the image to the right, &< moves it to the left, &+ moves it up, and &- moves the image down.

&. - plot single pixel.

This command plots a single pixel at the present screen position. The screen position is unaltered.

&S - plot circle outline.

This command is used to plot a circle centred at the present screen position with a given radius.

e.g. &S 30 would draw a circle of radius 30 pixels.

The initial screen position is restored on completion.

&F - plot filled circle.

This command is used to plot a filled circle (disk) centred at the present screen position with a given radius.

e.g. &F 30 would draw a disk with a radius of 30 pixels.

The initial screen position is restored on completion.

Bitmaps.

The following commands use the concept of bitmaps. A bitmap is a memory block containing screen image information. Two types of bitmap are supported.

A monochrome bitmap maps the memory contents to the display such that a single bit defines if a pixel is to be plotted or not. If the bit is a one then the pixel is plotted, otherwise, the pixel is skipped. The memory block is composed of a number of lines, each line is a whole number of bytes. The least significant bit of the first byte of each line is mapped to the initial horizontal screen position, more significant bits are mapped to the right. The first line is mapped to the initial vertical screen position, with subsequent lines mapped above the first.

A colour bitmap maps the memory contents to the display such that a single byte defines how a pixel will be operated upon. The least significant 3 bits contain a colour (the same value as would be used with the colour commands), the most significant two bits define the operation, if both bits are 0 then the pixel is skipped, otherwise a colour command is issued and the pixel is plotted. If both bits are 1 the &N command (absolute colour) is used, if only the most significant bit is 1 the &R command (invert colour) is used, otherwise the &W command (additive colour) is used. The memory block is composed of a number of lines. the first byte of each line is mapped to the present horizontal screen position with subsequent bytes mapped to the right. The first line is mapped to the initial vertical screen position, with subsequent lines mapped above the first.

&DEF - define bit map.

In order to use bitmaps, the memory blocks must be set up and then the bitmap parameters defined. One or more bitmaps are POKEd or BLOADed into a free area of memory. The memory address, size in the horizontal direction, size in the vertical direction and bitmap type are then defined by using this command.

e.g. &DEF A,X,Y,T defines a bitmap.

Where A is the starting address, X is the number of pixels horizontally, Y is the number of pixels vertically, and T is the type, 0 for monochrome, 1 for colour.

&B - plot a bitmap.

This command is used to plot a pre-defined bitmap at the present screen position. The initial screen position is restored on completion.

&L - complex area fill.

This command is used to fill an area of the screen with solid colour or a pre-defined bitmap. The boundary of the area to be filled is defined by the two parameters supplied with the command or the screen boundary. Any complex area with up to 255 involutions can be filled and if the limit is exceeded the fill operation will terminate.

e.g. &L M,C

The first parameter M should be in the range 0 - 7. A value of greater than 3 will cause the area to be filled with the pre-defined bitmap. If the value is 0 or 4 then the C parameter is not given, the fill boundary being screen pixels containing a different colour to the colour of the initial position.

If M is 1 or 5 the fill boundary is where a screen pixel contains one or more bits of the supplied colour (C). For example, &L 1,6 will fill until screen pixels which have any red colour in them. If M is 2 or 6 the fill boundary is screen pixels that are of a different colour than the supplied colour. If M is 3 or 7 the fill boundary is screen pixels that are the same colour as the supplied colour.

7). Applesoft support - utilities.

Two utility programs have now been included on the Applesoft support disk. They support printer dump and disk saving of screen images. Each utility consists of an Applesoft basic program and a relocatable machine language program. The basic program is used to load the machine language program and to configure the settings to be used.

Printer dump.

The printer dump utility will dump the VGPJI display onto an MX80, MX100, RX80, RX100, MX82 or FX80 printer, via Grappler, Grappler+ or Epson parallel interface cards.

The disk contains the Applesoft basic program SETUP DUMP. When this program is run, it BLOADs the relocatable machine language program DUMPT.OBJ0 into the highest free memory and then displays the configuration settings on the monitor and prompts: "D(UMP),E(DIT) OR Q(UIT) ?".

If the settings are not correct they may be edited by typing E then <RETURN>. Once the settings are correct, leave edit by typing 0 then <RETURN>, which stores the new settings on the file DUMPT.OBJ0, which can be BRUN later to give the dump operation.

Type D then <RETURN> to dump the screen. "PRINTED HEADING Y/N ?" is displayed. Type Y then <RETURN> to enter a one line heading to be printed at the top of the picture. The dump will then be printed and finished off with a form-feed.

Note that only the MX82 and FX80 printers allow a dot pitch of 1/72" in both directions. The other printers are set to a horizontal dot pitch of 1/60". In the case of the MX80 and RX80 printers, only the left 480 of the 512 dots are printed.

Disk save.

The disk utility allows screen images to be stored onto and recalled from the floppy disk. Run-length encoding is used to compress the data.

The disk contains the Applesoft basic program SETUP DISK. When this program is run, the relocatable machine language program DISKT.OBJ0 is BLOADed into the highest free memory. The settings for the slot number of the VGPJI, and the filename to be used, and the prompt "E(DIT),S(AVE),D(ISPLAY,Q(UIT) ?" are displayed.

If the settings are not as required they may be changed by typing E then <RETURN>. Once the settings are correct, the screen image may be saved by typing S then <RETURN>, or the file contents may be displayed by typing D then <RETURN>.

The minimum file size for 512*512 points is 11 blocks. The minimum time for disk save is 30 seconds, and for display is 3 seconds.

B). TASC support.

A TASC support disk is supplied with your VGPJI. The disk contains a machine language program which when BLOADed with a compiled TASC program performs all the operations as provided by Applesoft support.

TASC will not compile the & extensions used in Applesoft support so that an alternative method of support must be used. The interface from TASC basic is a series of CALL addresses and a set of common variables through which values are passed to the machine language subroutines. These are listed in the next section.

The disk contains the file VGP2X TASC BASE which is a TASC program that declares the common variables and then CALLs the initialise VGP subroutine. This file should be used as the start of any TASC program which uses the support.

The disk also contains the file VGP2X TEST which demonstrates how the support is used. It is a test program which exercises the support subroutines and the VGPJI. The next page shows how this program was compiled and run.

The source files for the machine language support are TS2A and TS2B. By changing the single statement SLOT EQU n, five binary files have been assembled, one for each slot. They are called TS2Xn where n is 1 to 5. When loading your program remember to load the appropriate binary file for the slot in which your VGPJI is installed.

```
]RUN TASC

MICROSOFT TASC
V 2.0, 1/10/82
COPYRIGHT (C)
1981 MICROSOFT

SOURCE FILE? VGP2X TEST,D2          the source file containing
                                    VGP2X TASC BASE.
OBJECT CODE FILE:
(DEFAULT VGP2X TEST.OBJ,D2,S6)?     type <RETURN> to accept.

MEMORY USAGE:
DEFAULT CONFIGURATION? N            do not use default.

ALTERNATE CONFIGURATION:

ADDRESS FOR LIBRARY:
(DEFAULT 2051)?                     type <RETURN> to accept.

ADDRESS FOR PROGRAM:
(NUMBER, 'HGR1', 'HGR2',
OR DEFAULT END OF LIBRARY)? 11320   program starts at address 11320
                                    after TASC support machine code.
ADDRESS FOR VARIABLES:
(DEFAULT END OF PROGRAM)?           type <RETURN> to accept.

LIBRARY OCCUPIES 2051 - 6063
PROGRAM BEGINS AT 11320
VARIABLES BEGIN AT END OF PROGRAM

ARE THESE ADDRESSES CORRECT? Y      type Y to accept.

OPTIONS:
DEFAULT CONFIGURATION?              type <RETURN> to accept.

*****BEGINNING PASS 1
the program listing is displayed here
*****BEGINNING PASS 2
CODE GENERATION COMPLETE
*****COMPILATION COMPLETE

]BLOAD RUNTIME                      load in TASC's runtime support.
]BLOAD TS2X3,D2                     load in TASC support for VGP slot 3.
]BRUN VGP2X TEST.OBJ                run the compiled program.
```

9). TASC support - subroutines.

The TASC support is a series of CALL addresses. These subroutines mimic the Applesoft support commands such that the order and value of all parameters are the same. The following table then lists the CALL addresses, parameters and function against the Applesoft command names. Refer to the Applesoft commands section for explanation of individual functions.

Applesoft command	CALL address	Parameters	Description of function
+	6070		Scroll up
-	6073		Scroll down
.	6076		Draw single pixel
<	6079		Pan left
>	6082		Pan right
A	6085		Apple video display
ASC	6088	V1 (,V2,V3)	Special colour: V2,V3 if colour
AT	6091	V1,V2,V3	V3 = screen colour at V1,V2
B	6094		Plot pre-defined bitmap
C	6097		Set cyclic mode
CLEAR	6100		Clear screen
D	6103		Pen/eraser down
DEF	6106	V1,V2,V3,V4	Define bitmap
DRAW	6109	V1	Select line style
E	6112		Select eraser
F	6115	V1	Filled circle of radius V1
FLASH	6118		Fill screen
FOR	6121	VFRMT,VCOLR	Select VGPJI format + colour
HGR	6124		Display page 1
HGR2	6127		Display page 2
HOME	6130		Reset position to 0,0
I	6133		Invert screen
INT	6136	V1	Select interlaced mode
INVERSE	6139		Set inverse write mode
L	6142	V1,V2	Complex area fill
M	6145	V1,V2	Move to position V1,V2
N	6148	V1	Set absolute colour
NEW	6151		Initialise VGPJI
NORMAL	6154		Normal write mode (white)
O	6157		Select clip mode
ON	6160	V1	Select non-interlaced subpage
OR	6163	V1	Set border colour
P	6166		Select pen
PLOT	6169	V1,V2,V3,V4	Draw line from V1,V2 to V3,V4
PLOTTO	6172	V1,V2	Draw line to V1,V2
POS	6175		Draw cursor at present X,Y
PRINT	6178	V$	Print text string V$
R	6181	V1	Set inverting colour
READ	6184	V1,V2	V1,V2 = present X,Y position
S	6187	V1	Draw circle radius V1
STORE	6190	V1	Set cursor size
TEXT	6193	V1,V2,V3,V4	Set text slant,orient and size
U	6196		Pen/eraser up
V	6199		Select VGPJI video
W	6202	V1	Set additive colour (protect)
X	6205	V1	Set pan register
Y	6208	V1	Set scroll register

Monochrome VGP][TASC support.

This note covers the upgrading of TASC application programs written for the VGP card for use with the VGP][card. New users of the VGP][card will normally use the new TASC support as described in the VGP][manual.

The VGP][TASC support contains the same subroutines at the same line numbers and uses the same variables as the VGP TASC support. Machine language routines are used for line-plotting as before, and also for initialisation, status checking, pixel testing and moving. The new machine language code still fits into the same memory space as the VGP version.

The TASC support disk supplied with the VGP][contains the file TS2, the DOS toolkit source of the support code, and the five binary files TS21 to TS25, one for each Apple expansion slot. The disk also contains VGP2 TASC SUBS which contains the source of the Basic subroutines which interface to the machine language routines. This file should be used as the basis of a TASC application program.

Also included on the disk is the file VGP2 TEST which has been compiled to obtain VGP2 TEST.OBJ. This is a test program which exercises all the routines in the support. This may be used to verify the operation of the system and also may be used as a reference to see how the support routines are used.

The following table lists the subroutines which comprise the support.

Subroutine	Line no.	Note	Description
INIT	60000	5	Initialises the VGPJI
DRAW	60100	2	Draw line from X1,Y1 to X2,Y2
LINE	60150	2	Draw line from present position to X2,Y2
CSIZE	60200	1	Select character size HS,VS
LINETYPE	60300	1	Select line type LT
PEN	60400	1	Select pen
ERASER	60500	1	Select eraser
PRINT	60600	2	Print text string CH$ at X1,Y1
CSTYLE	60700	1	Select text slant and direction DR
CLEAR	60800	2	Clear selected screen
BLOCK5	60900	1	Draw 5 x 8 block scaled by character size
BLOCK4	61000	1	Draw 4 x 4 block scaled by character size
TEST	61100	1	Read status (colour) of present point
PAGE1	61200	3	Display page 1
PAGE2	61300	3	Display page 2
NORMAL	61400	1	Select normal write mode
INVERSE	61500	1	Select inverse write mode
DOWN	61600	1	Select pen/eraser down
UP	61700	1	Select pen/eraser up
VGPVIDEO	61800	4	Select VGP video for display
APPLEVIDEO	61900	4	Select APPLE video for display

Notes:
======

1. These routines are exactly compatible with old support.

2. X values between 0 and 511 affect page 1, values between 512 and 1023 affect page 2. This applies to drawing and clearing.

3. Page selection is for display only. The X value selects which page is written to.

4. These are extra routines included for single monitor operation.

5. Initialisation is performed by a machine language routine. Both pages are cleared, page 1 is displayed, cyclic write, pen down, solid lines, minimum character size and normal write mode are selected. The X,Y position is set to 0,0.

   The VGPJI has programmable display formats. Both UK (50Hz 625 line) and USA (60Hz 525 line) may be used in either interlaced or non-interlaced modes. The variable FRMT (line 60045) should be set to 1 for UK interlaced operation. If you are using a seperate monitor for the graphics display then FRMT should be set to 5.

   Remember to set the variable SLOT (line 60040) to the slot number in which the VGPJI is installed.

10) Pascal support.

Introduction.

A Pascal support disk is supplied with the VGP][containing the code files and the text files which where used to create them.

The Pascal support provides the means of using the VGP][colour graphics system from Apple Pascal. 51 procedures have been provided to interface with the VGP][. No in depth knowledge of the operation of the vector processor hardware is now required to operate the vector processor from Pascal.

Detailed explanations of how to use the colour registers and how bitmaps are used are given in the Applesoft support commands section. Most of the procedures described in the next section have direct analogs in the Applesoft support and the reader is recommended to refer to that section when more detail is required.

How Pascal support works.

Apple Pascal provides a means of executing machine code procedures as if they were Pascal procedures themselves. Digisolve's Pascal support makes use of this feature to provide a number of procedures which interface between Pascal and the vector processor hardware for you. By using machine code for these routines a high performance system has been produced which because of Pascal's own speed can provide fast graphics.

How to use Pascal support.

The disk contains the files: VGP2SUP@.CODE where @ is a number from 2 to 5. Each code file has been assembled for use with the VGP][installed in a particular expansion slot. E.g. VGP2SUP4.CODE works with a VGP][installed into slot 4.

The disk contains the source for the above code files: VGP2.@.TEXT where @ is a letter from A to D. A single equate 'SLOT .EQU n' defines which slot is to be driven. This is the only difference which produced the four code files.

The file VGP2.X.TEXT contains an external declaration for all of the support procedures. This file will normally be used as the basis for Pascal programs written to operate the VGP][. If you do not use some of the procedures, their external declarations should be removed from the start of the program as this will save code space.

The file VGP2.U.TEXT is the source for the utility program described in the following sections. It was compiled and then linked with VGP2SUP4.CODE to produce the file VGP2U.CODE, which will only work with a VGP][in slot 4. If your VGP][is not in slot 4, compile VGP2.U.TEXT, then link the .CODE file with the appropriate support file to produce an executable utility file. (see over).

Creation of an executable code file.

Once a Pascal program has been written which requires the use of the Pascal support procedures the following method can be used to compile and link the program to produce an executable code file.

First C)ompile the Pascal source program in the usual way. This should result in a .CODE file which is ready to be linked to the system routines and also to the VGPI[support routines.

Next L)ink the code file produced above with the support code file which is associated with the slot in which the Vector Processor has been installed. Enter the linker and to the prompt 'HOST FILE' enter the name of the code file generated above. The linker will now ask for the name of a library file which should be 'VGP2SUP@.CODE' where @ is the slot of the vector processor. The linker will open this file and then ask for any further library names. If you have other library procedures then the name of their library should now be entered. Once all the library modules have been defined type <RETURN> to the library file prompt. The linker will now ask for a map file and you should respond with <RETURN> unless this option is required. The linker finally asks for an output file name which can be any convenient name and which will contain the executable code after the linking has taken place.

The code file produced by the above method can now be X)ecuted in the usual way.

11). Pascal support - procedures.

FORMAT(FMT:INTEGER);

This procedure tells the vector processor which of its 4 display formats to use, and if the colour expander card is fitted. When format is set to 0 then the VGP][will be in USA 512 x 416 mode, when format is 1 the display will be UK 512 x 512, when format is 2 the display will be 512 x 208 USA standard and when format is set to 3 the display will be 512 x 256 UK standard. The above values of format are used if the VGP][is operated on its own. If the colour expander is fitted, add 4 to the above values.

This procedure is normally the first call in an application program. It sets the VGP][as required, and then calls the following procedure INIT.

INIT;

This procedure re-initialises the vector processor. All registers are initialised, the character size is set to the minimum, lines will be drawn in solid form, page 1 is selected and cleared, the pen is selected and put down. The X and Y offsets are set to zero and the video switch is set to send 40 column video to the monitor.

This procedure doesn't change the video format, colour or interlace settings, as defined by the FORMAT procedure above.

CLEAR;

This procedure will clear the screen. To clear a specific screen the current position must be in the actual screen to be cleared.

PEN;

This procedure tells the vector processor that drawing operations will be carried out with the pen.

ERASER;

This procedure tells the vector processor that drawing operations will be carried out with the eraser.

UP;

This procedure will cause the vector processor to raise its pen . Drawing operations that follow this command will have no effect on the screen.

DOWN;
=====

This procedure tells the vector processor to lower its pen, drawing commands that follow this command will be seen on the screen.

HOME;
=====

This procedure resets the current screen position to 0,0 the bottom left hand corner.

BLOCK4;
=======

This procedure draws a square 4 by 4 pixel block at the current screen position. This block will be scaled by the character scale.

BLOCK5;
=======

This procedure draws a 5 by 8 pixel block at the current screen position. This block will be scaled by the current character scaling factor.

FILL;
=====

This procedure will set every screen pixel on.

CYCLIC;
=======

This procedure sets the cyclic mode of operation. Lines that go off the edge of the screen will wrap round and appear on the other side of the screen.

CLIP;
=====

This procedure sets the clipped mode of operation. Lines that go off the edge of the screen will be clipped and not appear on the other side of the screen.

HIGH;
=====

This procedure selects the high speed mode of operation. In this mode display is no longer performed and the screen will be blanked.

LOW;
====

This procedure selects the low or standard speed of operation.

NORMAL;
=======

This procedure selects the normal write mode of operation. Data being written to the screen will take no account of existing screen information.

INVERSE;
========

This procedure selects the inversion mode of operation. Data being written to the screen will be the inverse of the existing screen information.

PAGE1;
======

This procedure tells the vector processor to display the first page.

PAGE2;
======

This procedure tells the vector processor to display the second page.

LINETYPE(LIN:INTEGER);
======================

This procedure sets the line type with which the vector processor is operating. This can be 0,1,2 or 3. 0 is solid lines, 1 dotted lines, 2 dashed lines and 3 dotted-dashed lines.

DOT(X,Y:INTEGER);
=================

This procedure draws a single pixel at X,Y on the screen.

MOVE(X,Y:INTEGER);
==================

This procedure sets the current screen point to be X,Y.

LINE(X,Y:INTEGER);
==================

This procedure draws a line from the current screen point to X,Y.

DRAW(X1,Y1,X2,Y2);
==================

This procedure moves to the first point (X1,Y1), and then draws a line to the second point (X2,Y2).

TEXT(SLANT,ORIENT,XS,YS);

This procedure tells the vector processor the text style and size that should be used for writing. When SLANT=0 then normal text is used, when SLANT=1 slanted text is used. When ORIENT=0 then horizontal text is drawn, when ORIENT=1 vertical text is drawn. XS and YS select the character size.

VPRINT(VAR TEX:STRING);

This procedure prints the variable text string TEX at the current screen position, in the style specified by the 'TEXT' procedure.

PRINT(TEX:STRING);

This procedure is implemented in Pascal in the file VGP2.X.TEXT and is used to print text strings which are not in variables e.g. PRINT('Text string');

TEST(VAR COL:INTEGER);

This procedure is used to return the colour of a screen pixel at the present position. The value is returned in COL. In monochrome mode, the pixel may be on or off, which return values of 0 or 1 respectively. If the colour expander is fitted, a value from 0 to 7 is returned, 0 for white and 7 for black.

CURSOR(SIZE:INTEGER);

This procedure inverts a non destructive cursor at the current screen position of the size SIZE.

WHERE(VAR X,Y:INTEGER);

This procedure returns the screen position in the two variables X and Y.

MODE(MD,PLANE:INTEGER);

This procedure is used to specify the write mode for a particular colour plane. In a monochrome system there is only one plane so PLANE should be 0. If the colour expander is fitted the three planes, Red, Green and Blue, are set by PLANE values of 0,1 and 2 respectively. Valid write modes are from 0 to 7 and are described in the Applesoft support - commands section.

COLOUR(COL:INTEGER);

This procedure is used to select the absolute colour for drawing. Colours are 0:White, 1:Cyan, 2:Magenta, 3:Yellow, 4:Blue, 5:Green, 6:Red, 7:Black.

PROTECT(COL:INTEGER);
==========

This procedure is used to select the additive colour for drawing.

INVERT(COL:INTEGER);
==========

This procedure is used to select the inverting colour for drawing.

XOFFSET(OFST:INTEGER);
==========

This procedure is used to set the VGPII's X offset to PAN the display image.

YOFFSET(OFST:INTEGER);
==========

This procedure is used to set the VGPII's Y offset to SCROLL the display image.

VIDEO(VID:INTEGER);
==========

This procedure is used to control the VGP's soft video switcher. If the parameter passed is a zero then the Apple's 40 column video will be passed to the video monitor, if the parameter is a one then the VGPII's graphics image will be displayed on the screen.

INTERLACE(INT:INTEGER);
==========

This procedure is used to select interlaced or non-interlaced mode. Non-interlaced mode is usually used to eliminate flicker on short persistence monitors. If INT is 0 then non-interlaced is selected, if INT is 1 then interlaced mode is selected.

SUBPG(PAG:INTEGER);
==========

This procedure is used to select which of the two sets of pairs of pages is to be used for display and writing in non-interlaced modes.

BORDER(BRD:INTEGER);
==========

This procedure is used to select the border colour of the area between the active graphics area of the screen and the edge of the monitor tube. If the parameter passed is a one then the border will be off, if the parameter is zero then the border will be on. If the colour expander is fitted then BRD selects one of eight colours.

PAGDIS(PAG:INTEGER);

This procedure is used to select which of the two stored pages is used to be displayed on the monitor. If the parameter is zero then the first page is displayed, if the parameter is one then the second page will be displayed.

INVERT_SCREEN;

This procedure is used to invert the screen.

CIRCLE (RADIUS :INTEGER);

This procedure is used to plot a circle outline with the given radius with the present position being used as the centre point.

FILLED_CIRCLE (RADIUS :INTEGER);

This procedure is used to plot a filled circle (disk) with the given radius with the present position used as the centre point.

DEFINE_MAP (VAR MEM; XSIZE,YSIZE,TYPE :INTEGER);

This procedure is used to define the start address (MEM), size and type of a bitmap to be used in the following two procedures.

DRAW_MAP;

This procedure is used to draw a pre-defined bitmap at the present screen position.

CAFILL (MODE,COL :INTEGER);

This procedure is used to fill an area of the screen with solid colour or a pre-defined bitmap. MODE values of 0 to 3 fill with solid colour and MODE values of 4 to 7 fill with a bitmap. The 2 least significant bits of mode define the boundary of the area to be filled:
If they are 0, a boundary exists if the screen colour does not equal the initial point colour (COL is ignored).
A value of 1 gives a boundary if the screen colour contains any bits of COL.
Values of 2 and 3 give a boundary if the screen colour doesn't equal or equals COL respectively.

Utility procedures.

DUMP (PTYPE,ITYPE,ISLOT :INTEGER);

This procedure dumps the entire screen to a specified printer via a Grappler, Grappler+ or Epson parallel interface card. Each screen pixel is scanned and if its colour is not black , the corresponding dot on the paper is printed.

PTYPE	printer	ITYPE	interface
0	MX80	0	Grappler
1	MX100	1	Grappler+
2	RX80	2	Epson
3	RX100		
4	MX82		
5	FX80 / FX100		

The ISLOT value is set to the slot in which the interface card is installed.

Only the MX82 and the FX80 printers allow a dot pitch of 1/72" in both directions. The other printers all have a horizontal dot pitch of 1/60". In the case of the MX80 and RX80 printers, only the left 480 of the 512 dots are printed.

CODE_INIT;

This procedure is called once before a series of encode or decode calls. It saves the present VGPJI screen position and initialises the encoding and decoding variables.

function ENCODE (VAR BUFFER) :INTEGER;

This function loads BUFFER (a packed array [0..511] of char) with up to 512 bytes of run-length encoded screen data. The returned function value is the number of bytes stored, which will be 512 until the whole screen is encoded, at which point some value less than 512 is returned. Subsequent calls will return a value of 0.

To store a run-length encoded screen image, first call CODE_INIT, then, while ENCODE returns a non-zero value, store the buffer contents (on a disk file).

DECODE (VAR BUFFER);

This procedure takes the run-length encoded data in buffer and decodes and displays the information on the screen.

The normal sequence of restoring a screen image is to first call CODE_INIT, then, until end-of-file is reached on the disk file, load buffer from the file and call DECODE.

12). Pascal support - Utilities.

Introduction

The Pascal support utility program is executed from the Pascal command level. Screen images may be stored onto disk, recalled from disk or dumped to a printer.

The disk contains the file VGP2U.CODE. This file will only work with a VGP][installed in slot 4. See the Pascal support section if your VGP][is not in slot 4.

Using the utility program.

X(ecute the file VGP2U.CODE (or the name of the file containing the Linker output if you have relinked the utility program for a different slot). The program will run and starts with the format section in which a series of prompts are issued to enter the VGP][settings. Once the settings have been entered, they are sent to the VGP][and the effect may be seen. When you accept the settings the program will display the prompt:
"VGP: G(et S(ave D(ump F(rmt Q(uit [1.1]".

Type G to copy a screen image file to the VGP][. The prompt "Get ?" is displayed. Type the filename which contains the required image. If the filename extension is not .PIC then it is automatically added. If the file cant be found, the message "*** File not found ***" is displayed. If the file is found, the image is copied to the VGP][screen.

Type S to save a screen image onto a disk file. The prompt "Save as ?" is displayed. Type the name of a file in which the image is to be stored. If the extension is not .PIC then it is automatically added. If the file already exists, the prompt "Remove old filename ?" is displayed. Type N to keep the old file and terminate the operation. Type Y to continue. The screen image is encoded and stored onto the file.

Type D to dump the VGP][screen image to a printer. The prompt: "Dump using an RX80 printer, via a Grappler+ card in slot 1 OK?" is displayed. If the details are correct, type Y to commence the dump.
Otherwise type N and the prompt:
"Printer (0:MX80 1:MX100 2:RX80 3:RX100 4:MX82 5:FX80) Q(uit ?" is displayed. Type Q to terminate, or select the correct printer by typing its number. The next prompt: "Card (0:Grappler 1:Grappler+ 2:Epson) ?" is displayed. Select the interface card to be used. The next prompt "Card slot no. (1 - 5) ?" is displayed. Type the slot number in which the printer interface card is installed. The new settings are then displayed by the initial dump prompt.

Type F to re-enter the format settings as was done at the beginning of the program.

13) Technical reference guide.

Introduction.

This technical manual gives a detailed description of the operation of Digisolve's vector graphic processor system when used with the Apple JI computer. This peripheral system gives the Apple computer good quality, high resolution monochrome and colour graphics with onboard line drawing.

This peripheral does not rely on interrupts or DMA for its operation and thus should function in any Apple system without problems. This section is intended to provide enough information to software developers who require to use the card from an unsupported environment or directly from any language.

In order to drive the VGPJI, the user's software must interface to memory mapped registers which are used to communicate parameters and commands with the card. In this way there is no intermediate interface between the user and the graphics (such as RS232 or IEEE interfaces) to slow down the communication with the graphics hardware. The following section gives the programming model for the card(s) when used on an Apple JI computer system.

Vector processor programming model for Apple][computers.

The Apple computer system communicates parameters and commands to the card via fifteen memory mapped registers that are mapped to the peripheral and program space associated with the Apple expansion slots. The main registers of the vector processor are mapped to the I/O space of the slot whilst the mode control registers are mapped to the slot program space. All that the user's program must do is to write to the memory locations associated with the slot in which the card resides in order to communicate with the vector processor. The following two equations give the addresses of the memory area associated with individual slots.

```
BASE1 = 49152 + 128 + (slot*16)      -- address of main VGP registers
BASE2 = 49152 + (256*slot)           -- address of colour control registers
```

The following table gives the programming model for the vector processor.

Register Address	Bits used in register (MSB ... LSB)	Read	Write
BASE2	7 6 5 4 3 2 1 0	NV	FORMAT
BASE2+1	7 6 5 4 3 2 1 0	NV	MODE
BASE2+2	/ / 5 4 3 2 1 0	NV	YOFFST
BASE2+3	/ / 5 4 3 2 1 0	NV	XOFFST
BASE2+4	7 6 5 4 3 2 1 0	SCREAD	NV
BASE1	7 6 5 4 3 2 1 0	STATUS	CMD
BASE1+1	/ / / / / 3 2 1 0	CTRL1	CTRL1
BASE1+2	/ / / / / 3 2 1 0	CTRL2	CTRL2
BASE1+3	7 6 5 4 3 2 1 0	CSIZE	CSIZE
BASE1+5	7 6 5 4 3 2 1 0	DELTAX	DELTAX
BASE1+7	7 6 5 4 3 2 1 0	DELTAY	DELTAY
BASE1+8	/ / / / / 3 2 1 0	XMSB	XMSB
BASE1+9	7 6 5 4 3 2 1 0	XLSB	XLSB
BASE1+10	/ / / / / 3 2 1 0	YMSB	YMSB
BASE1+11	7 6 5 4 3 2 1 0	YLSB	YLSB

A number in the above table defines a used bit, a / defines an unused bit which should always be ignored when read and written as a zero. 'NV' in the read write table indicates an invalid operation.

Digisolve Vector Processor - Technical Specification.

RESOLUTION:	512 x 512 pixels, interlaced video 85 characters by 57 rows of alphanumeric text.
FORMAT:	Selectable UK/USA interlaced/noninterlaced.
MEMORY:	64K bytes, 2 screen buffer pages are stored.
SPEED:	A maximum vector draw speed of 1.2M dot/sec with an average of 0.9M dot/sec.
CHARACTERS:	96 ASCII charaters with descenders, with variable size and orientation.
LINES:	4 line types: solid, dotted, dashed and dotted dashed.
BLOCKS:	Variable size for fast area fill.
DATA INVERT:	For cursor drawing, intersecting lines can be inverted for clarity.
SCREEN READ:	To dump a drawing to a line printer, each pixel may be read individually.
SCREEN BUFFERS:	2 buffers may be swapped under software control.
VIDEO:	1V p-p composite & TTL.
SUPPLY:	Single +5v supply.
CONSTRUCTION:	Industrial components hand soldered on a high quality PCB. Soak tested for high reliability and MTBF.
COLOUR:	VGPJI expander card expands all functions to 8 colour operation.
WRITE MODES:	8 different write modes are available.
BORDER COLOUR:	Video border colour selectable.
VIDEO SWITCH:	To allow VGPJI and Apple JI to use same monitor
PAN and SCROLL:	Hardware pan and scroll for ultimate display flexability.

VGPJI REGISTER DESCRIPTION.

a) Status/Command register.

This register has two different functions; when writing data as a means of sending commands to the vector processor, and when read as a means of reading the internal status of the vector processor.

i) Status/Command writing - sending commands to the vector processor.

The following table shows the commands that are executed by the vector processor when the command bytes shown below are issued.

b3 b2 b1 b0		b7 b6 b5 b4 = 0 0 0 0	b7 b6 b5 b4 = 0 0 0 1	b7 b6 b5 b4 = 0 0 1 0	b7 b6 b5 b4 = 0 0 1 1	b7 b6 b5 b4 = 0 1 0 0	b7 b6 b5 b4 = 0 1 0 1	b7 b6 b5 b4 = 0 1 1 0	b7 b6 b5 b4 = 0 1 1 1	b7 b6 b5 b4 = 1 0 0 0	b7 b6 b5 b4 = 1 0 0 1	b7 b6 b5 b4 = 1 0 1 0	b7 b6 b5 b4 = 1 0 1 1	b7 b6 b5 b4 = 1 1 0 0	b7 b6 b5 b4 = 1 1 0 1	b7 b6 b5 b4 = 1 1 1 0	b7 b6 b5 b4 = 1 1 1 1
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0 0 0 0	0	Set bit 1 of CTRL1 : Pen selection	Vector operation (for b2, b1, b0 see small vector definition)	SPACE	0	@	P	`	p								
0 0 0 1	1	Clear bit 1 of CTRL 1 : Eraser selection		!	1	A	Q	a	q								
0 0 1 0	2	Set bit 0 of CTRL1 : Pen Eraser down selection		"	2	B	R	b	r								
0 0 1 1	3	Clear bit 0 of CTRL 1 : Pen Eraser up selection		=	3	C	S	c	s								
0 1 0 0	4	Clear screen		$	4	D	T	d	t								
0 1 0 1	5	X and Y registers reset to 0		%	5	E	U	e	u								
0 1 1 0	6	X and Y reset to 0 and clear screen		&	6	F	V	f	v								
0 1 1 1	7	Clear screen, set CSIZE to code "minsize" All other registers reset to 0 (except XLP, YLP)		'	7	G	W	g	w								
1 0 0 0	8		Special direction vectors (for b2, b1, b0 see small vector definition)	(	8	H	X	h	x								
1 0 0 1	9			)	9	I	Y	i	y								
1 0 1 0	A	5 x 8 block drawing (size according to CSIZE)		*	:	J	Z	j	z								
1 0 1 1	B	4 x 4 block drawing (size according to CSIZE)		+	;	K	[	k	{								
1 1 0 0	C	Screen scanning : Pen or Eraser as defined by CTRL1		,	<	L	\	l	¦								
1 1 0 1	D	X register reset to 0		-	=	M	]	m	}								
1 1 1 0	E	Y register reset to 0		.	>	N	↑	n	←								
1 1 1 1	F	Direct image memory access request for the next free cycle.		/	?	O	←	o	▒								

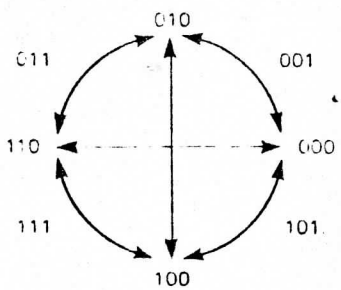
SMALL VECTOR DEFINITION :

b7	b6 b5	b4 b3	b2 b1 b0
1	ΔX	ΔY	Direction

Dimension

ΔX or ΔY		Vector length
0	0	0 step
0	1	1 step
1	0	2 steps
1	1	3 steps

Direction



A command may be issued by simply writing the value byte of the desired command to the command register after having checked that the vector processor is not busy obeying a previous command.

Commands can be broken up into four sections as follows:-

a. House keeping commands HEX 00-0F.
b. Vector generation commands HEX 10-1F.
c. Character commands HEX 20-7F.
d. Small vector generation commands HEX 80-FF.

a.) House keeping commands HEX 00-0F.

Two commands are not self explanatory from the previous command table and will now be explained.

1) Screen Scanning. (hex 0C)

This command can be used to either fill or clear the screen depending on the state of the pen/eraser bit in CTRL1. When pen has been selected this command will cause the screen to be filled, with eraser selected the screen will be cleared.

2) Direct Image-Memory access request. (hex 0F)

This command is used to enable the screen memory to be read on a pixel by pixel basis. The X and Y position registers are set to point to the pixel that is to be read and then this command is issued. When the vector processor is no longer busy the screen read bit may then be read (from SCREAD on the VGPJI, and if colour, SCREADG and SCREADB on the expander) to determine the state of a pixel.

b) Vector Generation Commands.

One of the most powerful features of the vector processor is its ability to generate vectors at a very high speed. A vector is specified to the vector processor as its start point (X-Y) and the displacement along the X and Y axes between its start and end points. Once this has been done the vector may be generated by determining the quadrant in which the vector is to appear and using this information to form the actual command that is written to the command register which causes the vector to be generated. On completion of the vector drawing process the X and Y registers point to the last pixel on the vector that was just created.

BASIC VECTOR DRAWING COMMANDS.

```
                 MSB                              LSB
                 ---------------------------------
CMD REGISTER     ! 0 ! 0 ! 0 ! 1 ! 0 ! X ! X ! 1 !
                 ---------------------------------
                                       !   !------ DELTA X SIGN = 0 if pos
                                       !---------- DELTA Y SIGN = 1 if neg
```

0 = bit is a zero, 1= bit is a one, X = bit defined by required option.

```
DELTA X NEG   ! DELTA X POS
DELTA Y POS   ! DELTA Y POS
              !
  hex 13      !   hex 11
--------------------------
DELTA X NEG   ! DELTA X POS
DELTA Y NEG   ! DELTA Y NEG
              !
  hex 17      !   hex 15
```

These four commands will draw vectors into the four quadrants as shown, in any direction in that quadrant as defined by the X and Y projections.

Supplementary Commands.

These commands ignore DELTA X or DELTA Y values by considering them to be of zero value. This allows lines to be drawn on the horizontal or vertical axes simply by specifying DELTA X or DELTA Y.

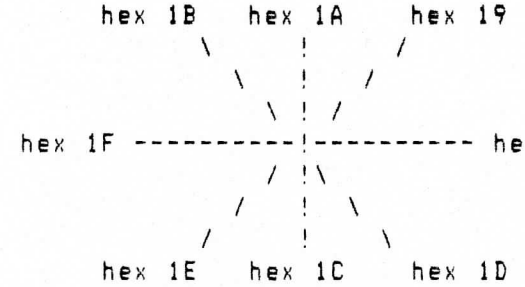
```
                 MSB                              LSB
                 ---------------------------------
CMD REGISTER     ! 0 ! 0 ! 0 ! 1 ! 0 ! X ! X ! 0 !
                 ---------------------------------
                  !-------------------------------!---!
---
0 0  DELTA Y ignored, DELTA X > 0
0 1  DELTA X ignored, DELTA Y > 0
1 0  DELTA X ignored, DELTA Y < 0
1 1  DELTA Y ignored, DELTA X < 0
```

```
                 hex 12
                    !
                    !
                    !
hex 16 ------------------- hex 10
                    !
                    !
                    !
                 hex 14
```

These commands allow vectors to be drawn along a major axis with only the DELTA X or DELTA Y register specified.

Eight further vector commands allow vectors to be drawn parallel to the axes or diagonals by considering the smaller DELTA value to be equal to the larger one.

```
                 ---------------------------------
CMD REGISTER ! 0 ! 0 ! 0 ! 1 ! 1 ! X ! X ! X !
                 ---------------------------------
                                       !---!---!----- Defined by direction.
```



Vectors may be drawn along the eight major axis by these commands.

c) Character Commands HEX 20-7F.

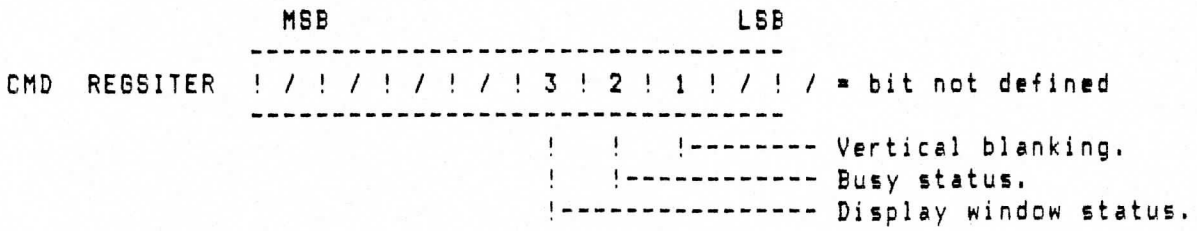
The vector processor contains an internal character generator to provide a 96 character upper/ lower case set with descenders. (see Appendix 1) To write a character on the screen its starting co-ordinate must be be specified (X and Y registers), its size (CSIZE register) and tilt type (CTRL2 register) must be specified. Once these registers have been set up, the ASCII code for the required charater (20-7F hex) is simply written to the command register. After drawing the character the vector processor updates the X and Y position registers to point to the position for the next character.

The vector processor only draws the locations that form the character. In the case of space, no drawing is performed. To delete a previously drawn character, set the eraser, and draw the 5 x 8 block (code hex 0A) or draw the same character, on top of the character to be deleted.

d) Small vector generation commands (hex 80-FF)

These commands allow short vectors to be generated without the need to set the DELTA X or DELTA Y registers. Vectors with projections of between 0 and 3 pixels in either direction may be drawn in any direction. After executing these commands the X and Y registers point to the pixel at the end of the short vector just drawn.

ii) STATUS/CMD reading - determining the vector processor's status.



Vertical blanking.

This bit indicates that the video signal generated by the card will be blanked (ie undergoing flyback). A high level indicates that the video signal is blanked. Certain commands produce a better display if executed during a blanking sequence (for example a bank switch between the two display pages looks better if executed during blanking.)

Busy Status.

After the vector processor executes a command the BUSY STATUS bit will go low to indicate that the vector processor is busy. When the vector processor can accept a new command this bit will go high to indicate this.

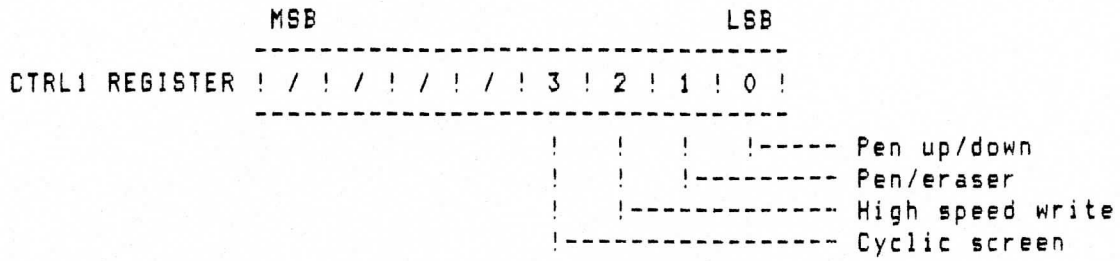
If a new command or any of the other registers in the vector processor are changed whilst the previous command is in process this may result in unpredictable results. Programs driving the vector processor from a language of high efficiency (assembler, a good compiler etc) should always check the status bit before changing registers. If a slow language is being used such as a BASIC interpreter then this process may not be needed as the vector processor is fast enough to have completed its previous command before receiving new ones from the user program in BASIC.

Display window status.

If this bit is high then it indicates that the last command to the vector processor resulted in the points on the line or character just drawn being outside the 512 x 512 display range.

b) CTRL1 register.

Unlike the CMD/STATUS register this register has the same meaning when written to or read from.



Pen up/down.

When this bit is set high,lines are drawn onto the screen. When this bit is low,lines drawn will not apear on the screen but the X and Y registers will be updated as usual. This bit can also be set or cleared by the pen up and pen down commands issued to the command register (hex 02 or hex 03).

Pen/eraser.

This bit controls the selection of pen for writing lines when high, or the eraser for rubbing out lines when low. Like the pen up bit this bit can also be controlled by commands to the control register (hex 00 or hex 01).

High Speed write.

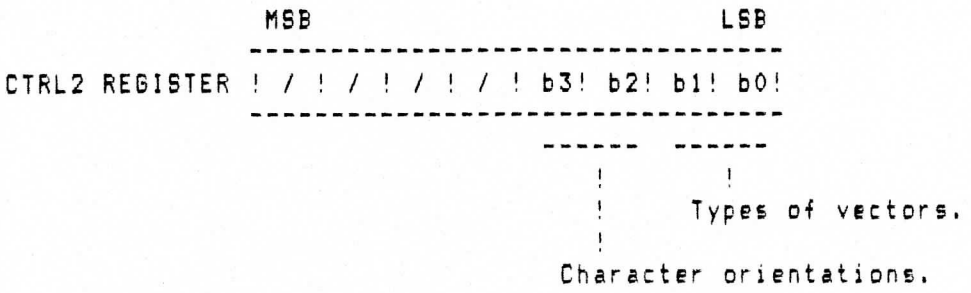
This bit controls the operation mode of the vector processor. When this bit is high the video display is switched off and the vector processor is freed to spend all of its time in writing data to the video memory. When this bit is set back low again video display will resume. This is a very useful feature for an application where a lot of data must be written to the screen in one go. The high speed write bit can be set to get the maximum speed from the vector processor, the drawing completed and then this bit cleared to resume normal viewing. This bit also provides a means for preparing a picture and then flashing it onto the screen at once so as to show no drawing operations.

Cyclic Screen.

The vector processor has the capability to handle X and Y addresses that are in the range of 0-4096. When the cyclic bit is clear lines that are drawn outside the display area will not be displayed on the screen, the vector processor automatically clipping to the screen edges. When the cyclic bit is high the screen repeats over the full 4096 by 4096 range allowed such that every line or point drawn will always appear on the display screen.

c) CTRL2 register.

This register is used to define the type of line being drawn and the orientation of characters.



The vector types are defined as follows:-

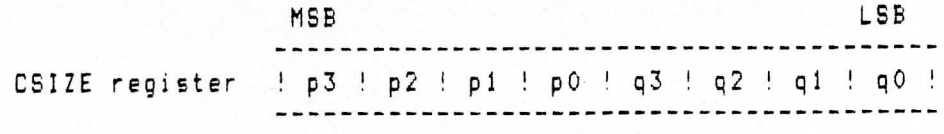
b1	b0	
0	0	continuous line
0	1	dotted line
1	0	dashed line
1	1	dotted dashed line

Character orientations are defined as follows:-

b3	b2	
0	0	normal characters
0	1	slanted characters
1	0	characters on the Y axis
1	1	slanted characters on the Y axis

d) CSIZE register.

This register is used to control the X and Y scaling of characters and blocks drawn by the vector processor.



p3,p2,p1,p0 is the scaling on the X axis
q3,q2,q1,q0 is the scaling on the Y axis

The two independant halves of this register may take values from hex 0 to hex F, where 1 represents the smallest characters through to F and 0 the largest size characters.

e) DELTA X register.

This eight bit register is used to specify the absolute X axis projection of a vector. This is simply the absolute difference value between the start and end points of the line which is required to be drawn. If the difference value is greater than 255 (the maximum possible in this register) then the projections should be divided by two and the vector required drawn as two halves.

f) DELTA Y register.

This eight bit register is used in the identical fashion as the DELTA X register but defines the projection on the Y axis.

g) XMSB, XLSB registers.

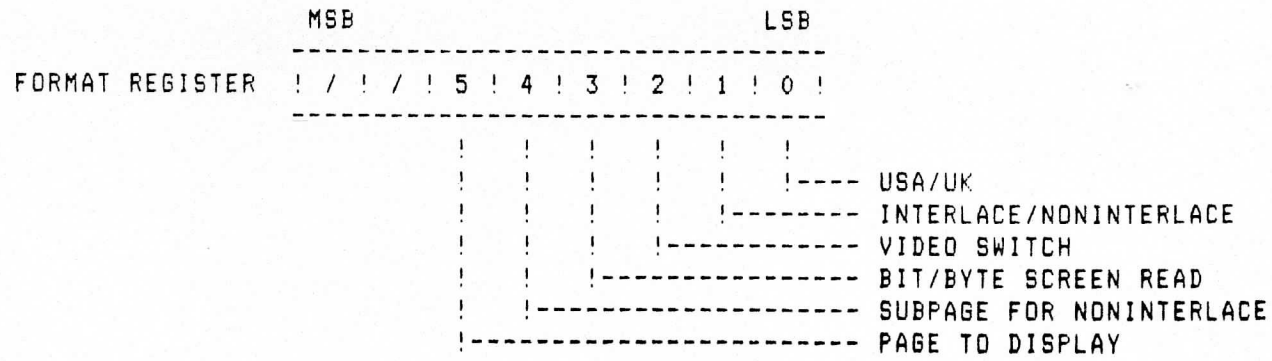
These two registers form the 12 bit X position value used by the vector processor to specify the start of lines or characters. After the vector processor has completed a command these registers will have been updated to point to the last pixel of the line or character just drawn. The XMSB register in non cyclic screen mode only consists of the least significant bit, or of the lower four bits when cyclic screen has been enabled.

h) YMSB, YLSB registers.

These two registers are used to form the 12 bit Y position value used by the vector processor.

i) FORMAT register.

This register is a write only 6 bit register which is used to select the display format and other mode control options.



1) USA/UK

When this bit is low then the VGPJI will generate 525 line 60Hz video, and when it is high 625 line 50Hz video will be generated.

2) INTERLACE/NONINTERLACE

When this bit is low interlaced video will be generated to give the higher resolution displays of 512 x 416 in USA mode, and 512 x 512 in UK mode.

3) VIDEO SWITCH

When this bit is low the video switch circuit on the VGPJI will send the video from the phono plug input to the phono socket. When this bit is high the video generated by the VGPJI's circuitry will be sent to the phono socket.

4) BIT/BYTE

When this bit is low the VGPJI circuitry will deposit the contents of the whole byte in which the requested pixel exists into the screen read data register. When the bit is high the data deposited into this register will be the state of the specified pixel only. When data is read from the screen read data register in byte mode it will be the inverse of the screen data, a high bit representing an off pixel. When bit mode is selected the data returned will be all bits on for an off pixel or any bit on for an on pixel.

5) SUBPAGE

When the VGPJI is in non interlaced modes two sets of pairs of pages can be stored in the VGPJI's memory. This bit is used to specify which one of these two pairs is used for display and writing.

6) PAGE TO DISPLAY

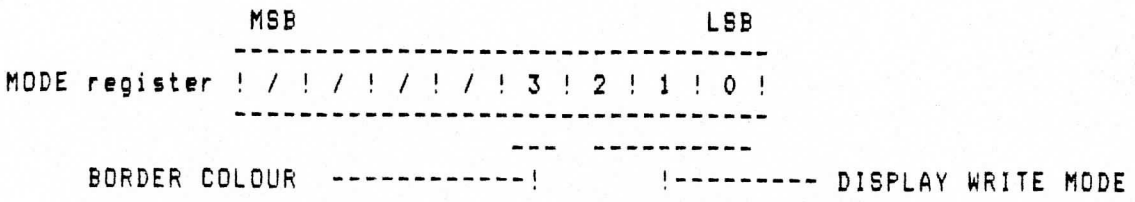
This register is used to specify which of the two display pages will be shown on the monitor in all modes. When this bit is low the left hand side of the display image X=0 to X=511 will be displayed, when the bit is high the right hand side of the image X=512 to X=1023 will be displayed.

The effects of reset.
==========

When the Apple computer is reset at power on or by operating the reset switch all of the bits in this register will be set to zero. It is also important to note that commands such as clear screen and reset registers ($07 to CMD register) will not effect any of the registers at BASE2 or on the VGPJI expander card.

j) MODE register.

This register is used to control the display writing mode and the state of the display border.



1) DISPLAY WRITE MODE

These three bits control the eight write modes for the VGPII card. The 8 modes are described in the Applesoft support section of this manual.

2) BORDER COLOUR

When this bit is high the display border will be off and when the bit is low the display border will be on.

k) YOFFST register.

This 6 bit register is used to set the Y offset (scroll displacement) of the display in multiples of 8 pixels.

j) XOFFST register.

This 6 bit register is used to set the X offset (pan displacement) of the display in multiples of 8 pixels.

k) SCREAD register.

This register is used to read the state of a screen bit or byte after a direct image access request has been issued to the command register. The meaning of its contents depends on the state of the BIT/BYTE bit in the FORMAT register.

14) Technical reference guide - colour expander.

The VGPIC expander card expands the operation of the VGPIC card to 8 colours. The expander card occupies the slot to the right of the VGPIC card and contains a further 128k bytes of screen memory. The expander has three 8 bit registers which are used to control operation in colour. These registers are mapped to the register address spaces associated with the slot of the VGPIC expander.

BASE3 = 49280 + (slot*16) where slot is the slot of the VGPIC expander

	MSB 7	6	5	4	3	2	1	LSB 0	READ	WRITE
BASE3									SCREDB	NV
BASE3+1									SCREDG	NV
BASE3+2									NV	COLMOD

NV in the read/write table indicates an invalid operation.

When the VGPIC system is being operated in colour the memory on the actual VGPIC card is used as the red display page, the memory on the expander card provides the green and blue display information.

a) SCREDB

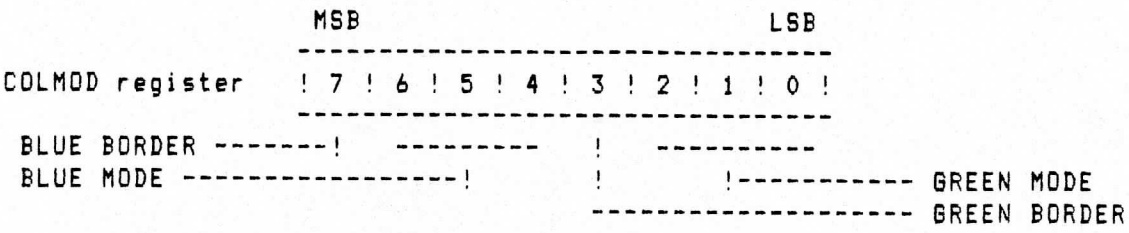
This eight bit register contains the screen read data from the blue plane after a direct image access request. The interpretation of this data depends on the setting of the BIT/BYTE control in the FORMAT register.

b) SCREDG

This eight bit register performs the same function as the above register except that the data contained in the register is data from the green plane.

c) COLMOD.

This register controls the write mode for the green and blue planes and the component of the border colour from the green and blue planes.



1) GREEN MODE

These three bits specify the write mode to be applied for the green display page. Valid write modes are described in the Applesoft support section of the manual.

2) GREEN BORDER

This bit specifies the green component of the border colour. If this bit is high no green video will be generated for the border, if the bit is low there will be a green component in the border colour.

3) BLUE MODE

These three bits specify the write mode to be applied for the blue display page.

4) BLUE BORDER

This bit specifies the blue component of the border colour. If the bit is high then there will be no blue component in the border colour, if the bit is low, blue video will be generated in the border.

Newsletter application form.

Digisolve will send you free for 1 year a newsletter of VGP developments both from DIGISOLVE and other independant software producers. To receive the VGP newsletter please fill in the following form and return it to:

Digisolve Ltd.,
Aire and Calder works,
Cinder Lane,
Castleford,
West Yorkshire,
England. WF10 1LU

Name: ----------

Company: ----------

Address: ----------

Where you obtained your VGPII: ----------

Your Computer configuration: ----------

Your Graphics application: ----------

If you are writing, or have written a program for a VGP product, please let us know. There are many other VGP users who might have a need for your program and would pay for it !