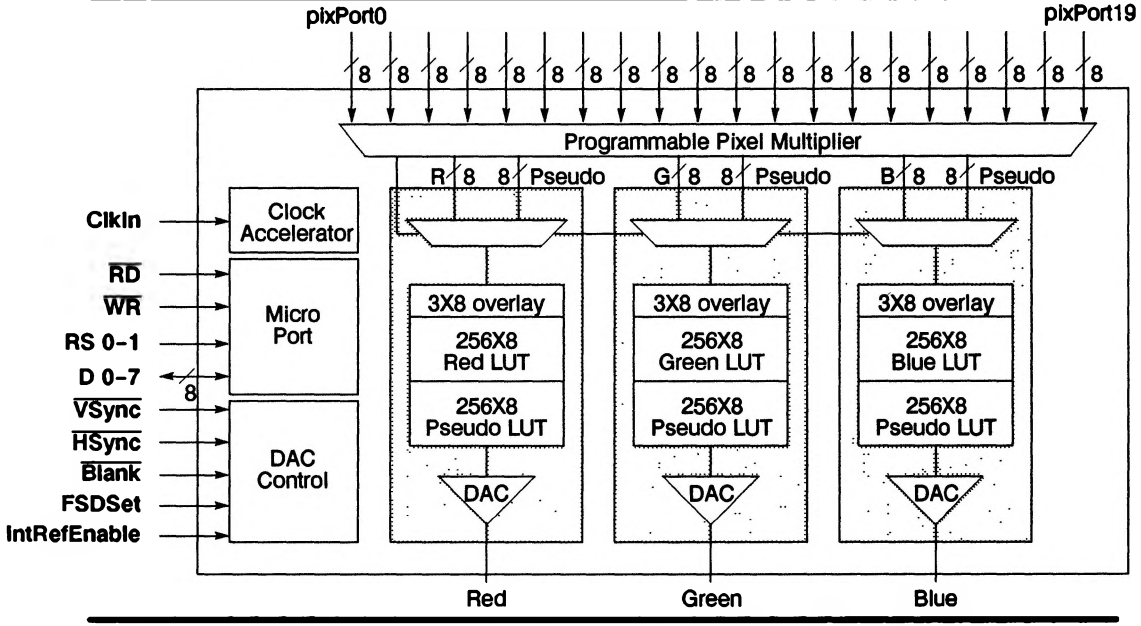
IMS G180
IMS G181
Combined true-colour and pseudo-colour look-up table

inmos®

Advance information



FEATURES

- 8 bit, 16 bit or 32 bit pixels at up to 150MHz
- Pixel multiplex at 2,3,4 or 5:1 (G180)
- Pixel multiplex at 2:1 (G181)
- Matching clock acceleration using on-chip PLL
- Three 256×8 LUTs for pseudo colour image
- Three 256×8 LUTs for gamma correction of 24 bit RGB image
- Three 3×8 bit overlay tables for cursor display
- Three EIA343-A 8 bit video DACs
- Optional on-chip DAC reference circuitry
- Pixel dot-rate checksum test facility
- Hardware pixel panning facility
- Selectable zero, 1/4, 1/2 and full DAC intensity
- All input signals and clocks at TTL rates and levels
- Composite sync and blank pedestal control
- 224 pin package (IMS G180)
- 132 pin package (IMS G181)

DESCRIPTION

The IMS G180 and G181 both combine a versatile pixel multiplexer together with three independent colour channels, each comprising two 256×8 bit colour tables, a 3×8 bit overlay table and a high performance video DAC.

By programming the pixel multiplexer the parts can be configured to work with a range of pixel sizes : 8 bit pseudo colour, 16 bit RGB or 32 bit (24 bit RGB + 8 bit pseudo). Using the latter format, it is easy to mix 24 bit true-colour and 8 bit pseudo-colour images on the same screen (picture in picture).

The G181 comes in a 132 pin package and multiplexes pixels 2:1, whilst the G180 comes in a 224 pin package and can multiplex pixels by 2,3,4 or 5:1 selectable under software control. Neither the G180 or the G181 require video rate pixel data or clock signals; the only video rate signals are the DAC outputs.

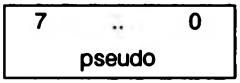
9.1 Device description

The IMS G180 and G181 are designed primarily for use in the output stages of high performance true-colour raster-scan video systems. They both contain three high-speed pipelined video channels, each containing two 256×8 bit colour look-up tables, a 3×8 bit overlay table and a video DAC (figure 9.1). All three channels are supplied with pixel data from a single programmable pixel multiplexer and are controlled from a single asynchronous 8 bit wide microprocessor interface.

9.1.1 Pixel modes

The G180 and G181 can work with pixels which are 8, 16 and 32 bits wide. In each case these pixels are multiplexed, 2:1 with the G181, and by 2, 3, 4 or 5:1 with the G180.

8 bit pixels are treated as pseudo-colour pixels and are mapped using the three 8 bit pseudo-colour tables before being displayed through the video DACs.

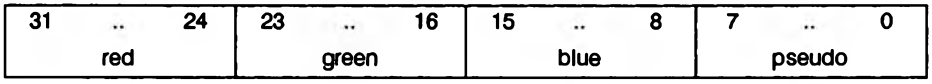


16 bit pixels are treated as true-colour pixels. 6 bits for red, 6 bits for green and 4 bits for blue. The least significant bits of each 8 bit component (red, green or blue) are set to zero and the resulting pixels are mapped using the three RGB colour tables.

The bits within a 16 bit pixel are interpreted as follows :-

32 bit pixels contain both a pseudo-colour field and a true-colour field. The pseudo-colour field contains 8 bits and is mapped with the pseudo-colour tables. The true-colour field contains 24 bits (8 bits each for red, green and blue) and is mapped using the RGB colour tables to provide gamma correction or a similar function. On each pixel cycle the part selects either the pseudo or the RGB field and translates with the appropriate colour table, before displaying the pixel using the video DACs.

A 32 bit pixel has two fields (pseudo and RGB) and is interpreted as follows :-



When operating in 32 bit pixel mode the G180 and G181 can be programmed so that the pixel select pins decide which field (RGB or pseudo) gets displayed by the video DACs. Alternatively, the decision can be based on the value of the pseudo field. In this mode if the pseudo field is 0 after masking, then the RGB field is displayed, if it is non-zero the masked pseudo field is displayed. The choice as to which method is used to switch between the two fields is made by writing to the compositing register. Both methods permit switching between RGB and pseudo streams on a pixel by pixel basis at any point in the screen, any number of times.

9.1.2 Clock acceleration

To simplify system design, both the G180 and the G181 incorporate a clock acceleration circuit. This circuit is used to multiply the **ClkIn** frequency by the same factor as the pixel data multiplexing.

This means that if for example 4:1 pixel multiplexing is selected, both the **ClkIn** and the pixel data are supplied at 1/4 of the video dot rate. This permits video operation at pixel rates of 150MHz, whilst avoiding

the need for any high speed ECL circuits outside the G180 or G181. System design is easier, cheaper and radiated emissions are kept to a minimum.

9.1.3 Overlays

The G180 and G181 both have overlay inputs. These can be used to overlay text or cursor information onto the final image. Overlays can be selectively enabled or disabled through the micro-port by writing to the overlay mask register. Overlay inputs are multiplexed by the same factor as the pixel inputs.

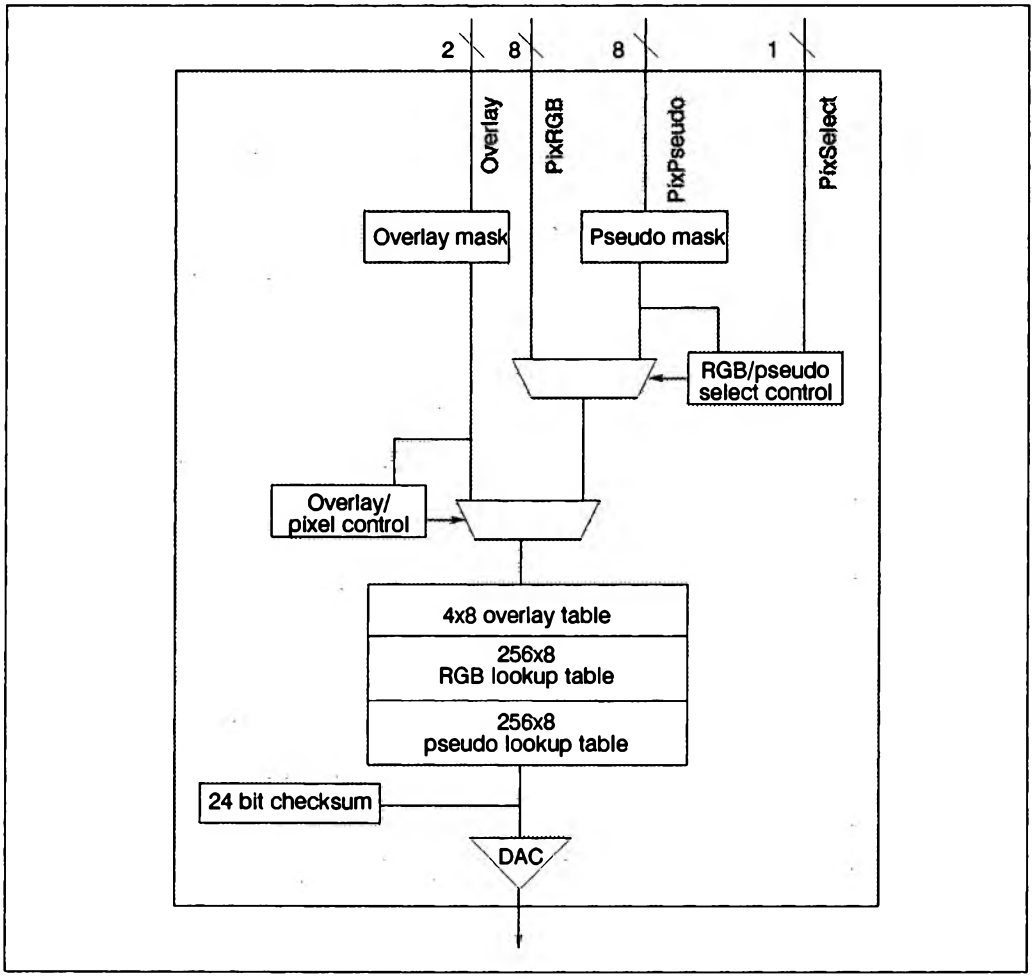


Figure 9.1 Colour Channel Architecture

9.1.4 Colour Channel Architecture

Figure 9.1 shows the architecture of each of the colour channels. The G180 and G181 each contain three such channels (i.e. one each for red, green and blue). The main pixel multiplexer takes pixel data from the pixel pins and passes this data to the inputs of the red, green and blue colour channels, under control of the pixel mode register.

Each colour channel has an RGB input, a pseudo-colour input, an overlay input and a pixSelect input. Each channel contains two 256 x 8 bit look-up tables, one for its RGB input and one for its pseudo-colour input and an overlay table containing 3 locations for the overlay input.

Pseudo pixel data is first masked with an 8 bit pseudo-colour mask. The mask register contents are bitwise ANDed with each pixel and the result passed on. By changing the contents of the pseudo-colour mask register rapid animation and flashing objects can be achieved on the screen.

The result of masking the pseudo pixel data may be compared with zero and the result of this compare is used to switch the pseudo/RGB multiplexer. If the result is zero the RGB field is selected and the RGB pixel will be mapped using the RGB colour table, if the result is non-zero then the pseudo field is selected and the pseudo pixel will be used to select a pixel value from the pseudo look-up table.

Alternatively, the RGB/pseudo multiplexer can be controlled directly by the **pixSelect** pins. The choice as to which method is used to switch between RGB and pseudo data is made by writing to the compositing register.

The final stage before the colour palette is the overlay logic. The overlay data from the pixel multiplexer is masked with the contents of the overlay mask register. The result of the masking operation is compared with zero. If the result is non-zero then the overlay multiplexer selects the overlay stream, and the overlay data will be mapped using the contents of the overlay table; if it is zero the RGB/pseudo stream is selected, and the resulting pixel will be displayed using either the pseudo or RGB colour tables. (Writing zero to the overlay mask register turns off the overlay function.)

9.1.5 Testability

All registers within the G180 and G181 can be read back through the microport. In addition, to assist in self-test and fault-finding on boards, a dot rate checksum facility is provided for each channel. This checksum resets on the rising edge of **$\overline{\text{VSync}}$** before the start of a frame and thereafter accumulates each 8 bit data value presented to the DAC inputs (except those for which **Blank** is active) into a single 24 bit checksum. This 24 bit checksum can then be read back during frame flyback before the checksum is reset and the next checksum value is computed. Thus all parts of the chip which affect the visible picture (with the exception of the DACs) are tested by this mechanism. There is a separate 24 bit checksum for each colour channel.

9.1.6 The DACS and reference circuitry

The 8 bit video DACs source current into external load resistors. They are designed to directly drive a doubly-terminated 75Ω transmission line. The DACs will drive a singly-terminated line but the edge rates, in particular the falling edge, will be slower. (Data sheet DAC parametrics are only guaranteed for double termination.)

The full-scale current of the DACs may be set by either an internal or an external reference. When using the internal reference all that is required is a single resistor, using an external reference requires a current source. The choice is determined by the **IntRefEnable** pin.

A DAC control register is provided for the selection of a variety of programmable features. These include optional **Sync** and **Blank** pedestals. In addition, the DACs can be restricted to operate in 6 bit mode, as well as being programmable to half, quarter or zero intensity.

9.1.7 Microprocessor interface

The G180 and G181 are programmed through a simple 8 bit microprocessor programming interface. All registers are selected using two register select lines $RS_0 - RS_1$. The contents of the look-up tables and all the registers are defined by writing to the microprocessor interface. All registers are readable.

9.2 Pin function reference guide

9.2.1 Pixel interface

Pin name	I/O	Signal name	Comments
ClkIn	I	Input clock	The rising edge of this clock controls the sampling of the pixel, overlay, **HSync**, **VSync** and **Blank**. The video dot rate is 2,3,4 or 5 times the frequency of **ClkIn**, depending on the pixel mode selected.
Pix0$_{0-7}$ **Pix1$_{0-7}$** **Pix2$_{0-7}$** **Pix3$_{0-7}$** **Pix4$_{0-7}$** **Pix5$_{0-7}$** **Pix6$_{0-7}$** **Pix7$_{0-7}$** **Pix8$_{0-7}$** **Pix9$_{0-7}$** **Pix10$_{0-7}$** **Pix11$_{0-7}$** **Pix12$_{0-7}$** **Pix13$_{0-7}$** **Pix14$_{0-7}$** **Pix15$_{0-7}$** **Pix16$_{0-7}$** **Pix17$_{0-7}$** **Pix18$_{0-7}$** **Pix19$_{0-7}$**	I	Pixel data	These pins are the pixel input ports (0–19 on the G180 and 0–7 on the G181), each port is eight bits wide. New pixel data is loaded on every rising edge of **ClkIn.**
PixSelectA **PixSelectB** **PixSelectC** **PixSelectD** **PixSelectE**	I	Pixel select	When programmed to operate with 32 bit pixels, these pins determine whether the 24 bit full-colour pixel is displayed by the DACs or whether the pseudo-colour pixel is displayed. (Only **PixSelectA** and **PixSelectB** are present on the G181 which can only multiplex pixels by 2:1)
OverlayA$_{0-1}$ **OverlayB$_{0-1}$** **OverlayC$_{0-1}$** **OverlayD$_{0-1}$** **OverlayE$_{0-1}$**	I	Overlay data	The overlay data sampled on these ports may be programmed to optionally override the pixel data sampled on the pixel port and substitute an overlay colour for a pixel colour. Overlay operation is defined by the contents of the overlay mask register. (Only **OverlayA–B** on the G181.)
$\overline{\text{Blank}}$	I	Blank	A low logic level on this input will cause a colour value of zero to be applied to the inputs of the DACs and will cause an offset corresponding to 5% of the DAC full-scale output to be removed if **Blank** pedestal is enabled, regardless of the colour value of the current pixel or overlay.
$\overline{\text{HSync}}$	I	Horizontal sync	A low logic level on this input indicates the beginning of a horizontal line flyback period and will cause an offset corresponding to 35% of the DAC full-scale output to be removed if composite **Sync** is enabled in the DAC control register.
$\overline{\text{VSync}}$	I	Vertical sync	A low value on this input indicates the beginning of a vertical frame flyback period and will cause an offset corresponding to 35% of the DAC full-scale output to be removed if composite **Sync** is enabled in the DAC control register. This signal is also used to reset the three 24 bit checksum registers

9.2.2 Analogue Interface

Pin name	I/O	Signal name	Comments
Red **Green** **Blue**	O O O		These signals are the outputs of the 8 bit DACs. They deliver current into a doubly-terminated 75Ω transmission line.
IntRefEnable	I	Internal Reference Enable	This pin selects whether the internal reference is to be used or not. If it is held high then the internal reference is selected and a resistor should be wired between **FSDSet** and **VDD**. If it is held low then an external reference current of 2.094mA must be connected between **FSDSet** and **VDD**.
FSDSet	I	Full-scale deflection set	When using the internal reference the value of the resistor placed between this pin and **VDD**, in combination with an on-chip reference circuit, determines the full-scale current output of the video DACs. When using an external reference a current source must be connected between this pin and **VDD**. **Warning:** If external reference is selected but resistor is connected, the chip will be damaged beyond repair.

9.2.3 Microprocessor Interface

Pin name	I/O	Signal name	Comments
$\overline{\mathbf{WR}}$	I	Write enable	The write enable signal controls the writing of data to the part. The state of the **RS** pins is sampled on the falling edge of $\overline{\mathbf{WR}}$ and the write data on $\mathbf{D_{0-7}}$ is sampled on the rising edge of $\overline{\mathbf{WR}}$
$\overline{\mathbf{RD}}$	I	Read enable	The read enable signal controls read operations on the part. The state of the **RS** pins is sampled on the falling edge of $\overline{\mathbf{RD}}$ and read data ceases to be valid on $\mathbf{D_{0-7}}$ after the rising edge of $\overline{\mathbf{RD}}$.
$\mathbf{RS_0}$-$\mathbf{RS_1}$	I	Register select	The values on these inputs are sampled on the falling edge of read or write enable. They specify which of the internal registers is to be accessed.
$\mathbf{D_0}$-$\mathbf{D_7}$	I/O	Program data	Data is transferred between the 8 bit wide program data bus and the internal registers under control of the read and write enable signals.

9.2.4 Power Supply

Pin name	Signal name	Comments
VDD	Power supply	Digital power is supplied from the **VDD** pins. All **VDD** pins must be connected to the **VDD** power plane.
AVDD	Analogue VDD	The DACs and internal reference are supplied from the **AVDD** pin.
GND	Digital ground	All **GND** pins should be connected to the **GND** plane.
AGND	Analogue GND	The analogue **GND** sinks the current from the DACs and reference circuitry.

9.3 Internal registers

All the registers within the G180 and the G181 are addressed by means of two address registers:- a set register and an index register. The set register specifies which data set is to be accessed and the index register specifies which location within the data set is to be accessed. Together the set register and the index register form a single 16 bit address which is used to access all registers.

The index register is auto-incrementing so that after the first read or write to a location within any given register set, the index register need not be re-written; successive locations in the register set may be written to or read from by repeatedly accessing the data register.

Where a register set only contains a single location the contents of the index register are ignored.

When the register set being addressed is one of the colour or overlay tables, the least-significant bit of the set register is used as a flag to indicate whether a read or a write to the register set is to be performed. This is necessary because accesses to the colour palette RAM and overlay palette RAM are internally pipelined. The read or write command must be passed down the pipeline.

Registers which are 24 bits wide are accessed one byte at a time in the order red, green and then blue. A internal counter controls this cycle and is reset to red each time a write is made to index register or the set register.

Unused bits in registers which are less than 8 bits wide are reserved and should be written with a data value of 0. When read back these reserved bits will return the data value 0 regardless of the data value written in.

The set register, the index register and all the data registers together are mapped into just three locations decoded from the state of the register select lines RS_0-RS_1 as follows :-

RS_1	RS_0	Register	Size
0	0	Set register	8
0	1	Index register	8
1	0	Data register	8
1	1	Reserved	-

Table 9.1 The G180 register mapping

Reading and writing to registers within the G180 is a three step operation, accessing these three registers in turn :-

1 Write to set register (also specifies read or write if accessing one of the pipelined LUTs)

2 Write to the index register

3 Read or write data from data register

For register sets with only one location step 2 may be omitted.

Set reg	Index register	Register set name	Read/Write	Size (of)
0	0..255	RGB LUT for write	W	24
1	0..255	RGB LUT for read	R	24
2	0..255	Pseudo LUT for write	W	24
3	0..255	Pseudo LUT for read	R	24
4–31	–	Reserved	–	–
32	0..3	Overlay LUT for write	W	24
33	0..3	Overlay LUT for read	R	24
34–63	–	Reserved	–	–
64	X	Pseudo mask register	R/W	8
65–66	X	Reserved	–	–
67	X	Overlay mask register	R/W	2(8)
68–69	X	Reserved	–	–
70	X	DAC control register	R/W	5(8)
71–72	X	Reserved	–	–
73	X	Pixel mode register	R/W	7(8)
74	X	Panning register	R/W	4(8)
75	X	Compositing control	R/W	2(8)
76–127	–	Reserved	–	–
128	X	Test checksum (lower byte)	R	24
129	X	Test checksum (middle byte)	R	24
130	X	Test checksum (upper byte)	R	24
131–255	–	Reserved	–	–

X = index register ignored

Table 9.2 The register set

9.3.1 Accessing the LUT and overlay tables

To write a set of LUT or overlay colour values the write address of the appropriate table is first written into the set register (0 for the RGB LUT, 2 for the pseudo-colour LUT or 32 for the overlay LUT). This must be followed by writing the first location to be accessed to the index register.

Red, green and blue values are then written in that order to the data register. After each group of three writes the index register will auto increment so that if a further three bytes are written to the data register the next location in each of the three colour tables will be updated. This makes updating a block of locations in the colour tables simple, since only the start address for a entire block of locations in the three colour tables need be written at the start of a sequence. (There is no facility for independently changing individual R,G or B colour components.)

Reading works in a similar way to writing. The read address of the look-up table to be accessed is written into the set register, followed by writing the address of the first location to be read to the index register. Successive reads of the data register will then return red, green and blue data in that order. Again the index register will auto-increment after three byte reads, so that if a further three bytes of data are read from the data register, the colour value data from the next location in each of the look-up tables will be returned.

9.3.2 The pseudo-colour mask and overlay mask registers

In each colour channel there is a pseudo-colour mask register which masks each bit of the pseudo-colour pixel address feeding each of the three pseudo-colour look-up tables. The contents of the mask is bit-wise ANDed with the 8 bit pixel address feeding each of the three pseudo-colour LUTs so that changing the contents of the mask register changes the address passed onto to each colour table. By changing the contents of the pseudo-colour mask register, rapid colour changes on the screen may be achieved. Also, by masking with zero the pseudo-stream may be disabled.

The overlay mask register enables or disables overlay operation. A one in a bit position in this register enables the respective overlay plane on each overlay port. If enabled, overlays operate on all pixels whether they are 8,16 or 32 bits wide.

Overlay mask register	Overlay inputs	Pixel data used
00	XX	Pixel data
01 01	X0 X1	Pixel data Overlay colour 1
10 10	0X 1X	Pixel data Overlay colour 2
11 11 11 11	00 01 10 11	Pixel data Overlay colour 1 Overlay colour 2 Overlay colour 3

Table 9.3 Overlay Operation

9.3.3 The pixel mode register

The pixel mode register controls how the part interprets the data presented at the pixel port. Pixel sizes of 8, 16 and 32 bits are supported on both the G180 and the G181.

The register is sub-divided into 3 fields (see table 9.4).

The first field, (bits 0-1) selects the pixel size, 8, 16 or 32 bits and is a binary coding of the number of bytes per pixel.

The second field (bits 2-4) selects the number of pixels which are loaded on every clock edge, ie the pixel acceleration factor. In addition to setting the pixel acceleration factor this field also sets the clock acceleration factor. For example if 4:1 acceleration is selected on the G180, 4 pixels are latched on every **ClkIn** edge and the **ClkIn** is multiplied internally by a factor of 4 to generate the video rate clock.

The third field (bits 5-6) determines the source of the pseudo-colour data fed to the colour-channels. The pixel multiplexor provides a 32 bit output to the three colour channels. These register bits select which of the 4 constituent bytes are driven to the channels as the pseudo-pixel data.

Operation			Comment
Bit 6, 5 Page mode	**Bit 4, 3, 2 Accel**	**Bit 1, 0 Pixel size**	
		00 01 10 11	32 bit pixel 8 bit pixel 16 bit pixel Reserved
	nnn		binary coded accel factor
00 01 10 11			pseudo→pseudo blue→pseudo green→pseudo red→pseudo

Table 9.4 The pixel mode register

8 bit pixels are treated as pseudo-colour pixels and are translated into RGB data using the contents of the pseudo-colour tables.

16 bit pixels are treated as true-colour pixels (6 bits red, 6 bits green and 4 bits blue). The lower bits of each colour component are set to 0, before being translated into RGB data by the RGB colour tables.

32 bit pixels have both a pseudo-colour and a true-colour field and the part can switch, on a pixel by pixel basis, between the two fields. This can be done either by controlling the **pixSelect** pins or by using the value of the pseudo field. If selected the pseudo-colour field will be displayed using the pseudo-colour table, otherwise the true-colour field will be displayed using the RGB colour tables.

The pixel port on the G180 (G181) can be viewed as 20 (8) 8 bit ports. Tables 9.5 and 9.6 show how the G180 (G181) can interpret pixel data presented at these ports. The letters in parentheses indicate ports that are used only if the multiplexing rate is high enough and which are therefore not available on the G181.

bits/pixel	Port number (each port is 8 bits wide)																			
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
32 bit	A				B				(C)				(D)				(E)			
16 bit	A			B	(C)			(D)	(E)			unused								
8 bit	A	B	(C)	(D)	(E)	unused														

Table 9.5 Scanning of pixel inputs for different multiplexing rates

bits/pixel	Bit position in word																			
	0			3	4			7	8	9	10			15	16			23	24	31
32 bpp:	P[0]....P[7]				B[0]....B[7]				G[0]....G[7]				R[0]....R[7]							
16 bpp:	B[4]..B[7]				G[2]..G[7]				R[2]..R[7]			unused								
8 bpp:	P[0]....P[7]				unused															

Note: P = Pseudo, R = Red, G = Green, B = Blue pixel data

Table 9.6 Assignment of bits in different pixel modes

9.3.4 The DAC control register

The DAC control register controls the operating features of the video DACs.

The features supported are:

- an optional **Blank** pedestal, an optional **Sync** pedestal
- a reduction in the DAC resolution from 8 to 6 bits
- reduced contrast DAC display modes, whereby the DACs may be programmed to output full intensity, half intensity or quarter intensity images as well as forcing blanked output.

Bits 4, 3	Bit 2	Bit 1	Bit 0	Operation
DAC contrast	8/6	Black lev	Sync	
			0	**Sync** pedestal on
			1	**Sync** pedestal off
		0		**Blank** pedestal on
		1		**Blank** pedestal off
	0			8 bit DAC operation
	1			6 bit DAC operation
00				Full intensity operation
01				Half intensity operation (stand-by)
10				Quarter intensity operation (stand-by)
11				DACS blanked

Table 9.7 The DAC control register

9.3.5 Panning register

This register allows the pixel pipeline of the **Sync**, **Blank** and **Overlay** inputs to be delayed by up to 7 pixels relative to the pixel data read in on the pixel ports. The user has the choice as to whether to delay the overlays or not.

By changing the contents of this register a smooth hardware horizontal pan can be easily achieved, provided the colour monitor is taking its **Sync** pulses from the DAC outputs and is not using a separate **Sync** line.

A value of 0 in the delay field sets the internal pipeline delay of the composite **Sync, Blank** and, optionally, the **Overlay** signals to be equal to the pipeline delay of the pixel information through the look-up tables. A value of one in this field delays these signals by one pixel relative to the pixel data, a value of two delays them by 2 pixels and so on.

Bit 3 selects whether the overlays pan or not. If it is set high then the overlays will pan with the picture, if it is set low the overlays will not pan with the picture i.e. are delayed with the **Syncs** and **Blanks**.

Bit 3	Bits 2, 1, 0	Blank delay relative to pixel data in units of pixels	Overlays
0			Pan with picture
1			Do not pan with picture
	000	0	
	001	1	
	..	..	
	110	6	
	111	7	

Table 9.8 The panning register

9.4 Compositing control register

The G180 provides several means for combining the pseudo and true-colour data streams. There are four options (numbered 0..3) controlled by bits 1 and 0 in the compositing control register.

Bits 1, 0	Operation
00	if the **pixSelect** pin is low, the true-colour image is selected irrespectively. If the **pixSelect** pin is high then the pseudo-colour image is selected, unless the pseudo-colour pixel is zero, in which case the true-colour image is selected.
01	If the pseudo data is zero the true-colour image is selected, otherwise the pseudo-colour image is selected.
10	If the **pixSelect** pin is high the pseudo image is selected, otherwise the true-colour image is displayed.
11	The pseudo-colour image is always selected.

9.5 Test checksum register

The G180 and G181 both have a pixel rate checksum test facility.

Each channel (red, green and blue) has a 24 bit checksum register which is reset on the rising edge of **VSync** and accumulates each data value thereafter into a 24 bit checksum register. (Pixel data supplied whilst **Blank** is asserted has no effect on the contents of the checksum register i.e. only visible pixels affect the value of the checksum.) The checksum works on one frame's worth of pixel data at a time.

At the end of a frame, during the flyback period, the microprocessor is able to read the three final checksum values for each channel. Like the 24 bit colour values the checksum values are read in the order red, green and blue. The data value must be read before the rising edge of the next **VSync** after which point the checksum will be reset ready for the next frame of data.

The checksum register is a linear feedback shift register. Data bytes are accumulated in pairs, before being exclusive OR'ed into the 24 bit checksum register.

The algorithm for the checksum can be expressed in C as follows :-

```
#define bitsincrc 24      /* number of bits in checksum */
#define tapmask 0x610000L /* generator polynomial for 24 bit checksum */
#define bitsindata 16     /* number of bits in data word */

long checksum, clipcrc, topbit;
int  dataword, clipdata, pixelnumber;

  topbit    = 1 << (bitsincrc-1);
  clipcrc   = (1<<bitsincrc) - 1;        /* for clipping crc to length */
  clipdata  = (1<<bitsindata)-1;         /* for clipping data to length */
  checksum  = clipcrc;                   /* initialise checksum */

                                         /* maxPixel = number of */
                                           visible pixels       */
  for (pixelnumber=0; pixelnumber < maxpixel;
                      pixelnumber = pixelnumber+2 )
    {
      dataword = getpixel(pixelnumber) | (getpixel(pixelnumber + 1) <<8);
      dataword = dataword & clipdata;
      checksum = checksum ^ dataword;

      if ((checksum & topbit) == 0)      /* if top bit clear */
        {
          checksum = checksum ^ tapmask; /* feedback to taps */
          checksum = checksum << 1 ;     /* shift left */
          checksum = checksum | 1;       /* set least significant bit */
        }
      else
        checksum = checksum << 1;       /* shift left and clear LSB */

      checksum = checksum & clipcrc;    /* clip to length */
    }
```