Information for Cirrus Chipset Users

Harm Hanemaayer (*hhanemaa@cs.ruu.nl*), Randy Hendry (*randy@sgi.com*) 27 January 1995

Contents

1 Supported chipsets

There are two different SVGA drivers for Cirrus chipsets, one called "cirrus" and one called "cl64xx". The "cirrus" driver is used in the 256-color SVGA server (with acceleration) and the mono server (without acceleration). The SVGA server supports 16 and 32 bits-per-pixel truecolor modes on some configurations. The "cl64xx" driver is used in the 256-color SVGA, 16-color and mono servers. Note that except where stated otherwise, this document is referring to the "cirrus" driver. The following chipsets by Cirrus Logic are supported:

CL-GD5420

ISA SVGA chipset, 1Mbyte; maximum dot clock is 45 MHz (256 color server). Acceleration with extended write modes (used for scrolling and solid filling in this driver). This chipset can *not* support 1024x768 non-interlaced in 256 colors.

CL-GD5422

Enhanced version of the 5420 (32-bit internal memory interface). Maximum dot clock is 80 MHz.

CL-GD6205/6215/6225/6235

Laptop chipsets more or less compatible with the 5420. The only dot clock supported is 25 MHz (more on an external display). Some problems have been reported with these chipsets (especially on external displays). Take note of the "`noaccel`" option.

CL-GD6420/6440

These chipsets are not compatible with the 542x series, but are supported by the "cl64xx" driver. It is used in recent laptops, and bears some similarity to old Cirrus chipsets (5410/AVGA2). The driver may also work for other 64xx chips. The configuration identifiers for this driver are "`cl6420`" and `"cl6440"`. This driver is discussed in detail in section 6.

CL-GD5424

Basically VLB version of the 5422, but resembles the 5426 in some respects.

CL-GD5426

Supports both ISA bus and VLB, and up to 2Mbyte of memory. Has BitBLT engine for improved acceleration (BitBlt, image transfer, text). Dot clock limit is 85 MHz.

CL-GD5428

Enhanced version of the 5426.

CL-GD5429

Enhanced version of the 5428; officially supports higher MCLK and has memory-mapped I/O.

CL-GD5430

Similar to 5429, but with 543x core (32-bit host interface). Does not have 64-bit memory mode.

CL-GD5434

Next generation 'Alpine' family chip with 64-bit internal memory interface. Used by the Orchid Kelvin 64. The chip can only support 64-bit mode if equipped with 2 Mbytes of memory; cards with only 1 Mbyte are severely limited. Support dot clocks up to 110 MHz.

Here's a list of maximum dot clocks for each supported depth:

```
                    mono       8 bpp (256c)    16 bpp____      32 bpp
CL-GD62x5        45 MHz      45 MHz____
CL-GD5420        80 MHz      45 MHz (1)
CL-GD542x/512K   80 MHz      45 MHz____
CL-GD5422/24     80 MHz      80 MHz____      40 MHz____
CL-GD5426/28     85 MHz      85 MHz____      45 MHz (2)
CL-GD5429        85 MHz      85 MHz____      50 MHz____
CL-GD5430        85 MHz      85 MHz____      45 MHz (2)
CL-GD5434/1Mb    85 MHz      85 MHz____      45 MHz (2)
CL-GD5434/2Mb    85 MHz      85 MHz____      85 MHz____      45 MHz (2)
```

```
(1) with 512K memory.
(2) 50 MHz with high MCLK setting.
```

Rough virtual/physical screen resolution limits for different amounts of video memory:

	mono	8bpp	16 bpp	32 bpp
256K_	800x600__	640x400__	_________	________
512K_	1152x900_	800x600__	640x400__	________
1024K	1600x1200	1152x900_	800x600__	________
2048K	2304x1728	1600x1200	1152x900_	800x600_
4096K	2304x1728	2272x1704	1600x1200	1152x900

The virtual width must be a multiple of 32 if acceleration is used. The horizontal monitor timings must be below 2048.

To run `XF86_SVGA` at 16 or 32 bpp, pass options to the X server as follows:

```
startx -- -bpp 16               5-6-5 RGB ('64K color', XGA)
startx -- -bpp 16 -weight 555   5-5-5 RGB ('Hicolor')
startx -- -bpp 32               8-8-8 XRGB truecolor (5434)
```

2 Basic configuration

It is recommended that you generate an XF86Config file using the '`xf86config`' program, which should produce a working high-resolution 8bpp configuration. You may want to include mode timings in the `Monitor` section that better fit your monitor (e.g 1152x900 modes). The driver options are described in detail in next section; here the basic options are hinted at.

For all chipsets, a `Clockchip "cirrus"` line in the `Device` section can be useful. This allows the use of any dot clocks, instead of one out of the fixed set of dot clocks supported by the driver. This is required if you want a 12.6 MHz dot clock for low-resolution modes.

If graphics redrawing goes wrong, first try the `"no_bitblt"` option if your chip has the BitBLT engine, if that doesn't work, try the `"noaccel"`; option, which disables all accelerated functions. The `"no_imageblt"` option may be sufficient.

In order to be able to run at a depth of 16bpp and 32bpp, and to improve performance at 8bpp, linear addressing must be enabled. This is generally possible on 543x local bus cards, and if you have less than 16Mb of system memory, on local bus 542x cards and ISA 543x cards. You must specify the `"linear"` option and possibly a `Membase` address. See the following sections for a detailed description.

Lastly, on the 543x, it is advisable to use the `"mmio"` option in the Device section for better acceleration (this might also apply to the 5429, which has not been tested).

3 XF86Config options

Don't use the '`Clocks`' command. The clocks are fixed (i.e. not probed), and there should be no variation between cards (other than the maximum supported clock for each chipset).

The following options are of particular interest to the Cirrus driver. Each of them must be specified in the '`svga`' driver section of the `XF86Config` file, within the `Screen` subsections of the depths to which they are applicable (you can enable options for all depths by specifying them in the Device section).

Option "noaccel"

This option will disable the use of any accelerated functions. This is likely to help with some problems related to DRAM timing, high dot clocks, and bugs in accelerated functions, at the cost of performance (which will still be reasonable on VLB).

Option "fast_dram" "med_dram" "slow_dram" (5424/6/8/9, 543x)

The `"fast_dram"` option will cause the driver to set the internal memory clock (MCLK) register of the video card to a higher value. Normally, this register is not touched but it appears that the standard CL-GD542x BIOS initializes it to a value that is somewhat on the low side (limited by the chip specification), which has a negative influence on performance of high dot clock modes. This is especially true if extended RAS timing is being used (this is indicated in the server probe). The actual speed of DRAM is not a critical factor in the determining whether this option is appropriate; one card with 80ns DRAM using Extended RAS timing, which came with a DOS driver utility to set the MCLK to this value (0x22), seems to run stable at higher MCLK.

There are also (mainly brand name) cards whose customized BIOS does initialize to a higher non-standard value. In this case, the use of this option is probably not appropriate.

The `"slow_dram"` option will set the MCLK to the value used by the standard CL-GD542x BIOS (0x1c). Symptoms of a MCLK that is too high can be vertical bands of flickering pixels on the screen, erroneous pixels appearing in text, and loosing pixels in the textmode font after running X (note that very similar effects can be caused by an MCLK setting that is too low).

Upon start-up, the driver will report the value of the MCLK register (check this first), and also any changes that are made.

Typical MCLK values:

0x1c (50 MHz)

This is usually the BIOS default. It is forced by the `"slow_dram"` option.

0x1f (55 MHz)

Value used by the `"med_dram"` option. Highest value that 542x based cards seem to be able to handle with linear addressing enabled.

0x22 (60 MHz)

Value that most (Extended RAS) 542x cards seem to be able to handle, used by the `"fast_dram"` option.

The official maximum of the 542x chips is 50 MHz. The official spec. for the 5434 is also 50 MHz (0x1c) and that for the 5429 and 5430 is probably 60 MHz (0x22). Current revisions of the 5434 (E and greater) support 60 MHz MCLK in graphics modes, and the driver will program this automatically. If it causes problems, use the "slow_dram" option.

The driver takes the MCLK into account for clock limits that are determined by DRAM bandwidth.

If you are not having any problems (performance or stability at high dot clocks), it is best not to use the "`fast_dram`" option.

Option "no_bitblt"

This option, when used with a 5426/28/29/3x, will have the effect of disabling the use of the BitBLT engine (which the 5424 does not have), while retaining some acceleration. This will be useful for problems related to functions that use the BitBLT engine.

Option "no_imageblt"

This option, when used with a 5426/28/29/3x, will have the effect of disabling the use of BitBLT functions that go from system memory to video memory. This is useful for problems relating to image write, such as a little white line at the top left corner of the screen, or a skewed image after a console switch back to the server, which have been observed on some configurations, especially VLB 5426 and 5434 with a fast CPU. Note that this option results in reduced performance.

chipset "clgd54xx"

Force detection of the given chipset. Useful if you have a supported chipset that is not properly detected, or if you have an unsupported chip that might be compatible with a supported one.

videoram 1024 (or another value)

This option will override the detected amount of video memory, and pretend the given amount of memory is present on the card. This is useful on cards with 2Mbyte of memory whose DRAM configuration is not compatible with the way the driver enables the upper megabyte of memory, or if the memory detection goes wrong. It must be specified in the Device section.

Option "fifo_conservative" (5424/6/8/9/30)

This option will set the CRT FIFO threshold to a conservative value for high dot clocks (>= 65 MHz), reducing performance but hopefully alleviating problems with 'streaks' on the screen (especially when a BitBLT operation is in progress).

Option "fifo_aggressive" (5424/6/8/9/30)

This option will set the CRT FIFO threshold to an aggressive value; it will be the same as that used for lower dot clocks. It improves performance at high dot clocks.

Option "no_2mb_banksel" (542x)

This option will cause the driver not to set the 'DRAM bank select' bit to enable the upper megabyte of memory on a 2Mbyte card. This should be helpful with cards equipped with 512Kx8 DRAMs, as opposed to 256Kx4/16 DRAMs, when using a virtual screen configuration that uses more than 1Mbyte of memory.

Option "probe_clocks"

This option will force probing of dot clocks on the card. This should not be necessary, since the clocks are fixed and the same for all Cirrus chipsets. However, they do depend on the motherboard supplying a proper standard 14.31818 MHz frequency on the bus. There may be ill-designed VLB motherboards that do not supply this frequency correctly for certain bus speeds (e.g. scaled up from 33 MHz). If in doubt, use this option and run '`X -probeonly`'. If the clocks are very different from the correct clocks shown below, you can try a `Clocks` line in the `XF86Config` with your deviating clocks. In this situation, the MCLK is probably also screwed.

Correct clocks:

```
25.2  28.3  41.1  36.1  31.5  40.0  45.1  49.9
65.0  72.2  75.0  80.0  85.2
```

Clockchip "cirrus"

This enables programmable clocks. It must be specified in the Device section. With this option, the clocks the modes use will be automatically selected. Do not specify any Clocks line. This option makes a 12.5 MHz clock possible for a 320x200 Doublescan mode. Note that some frequencies may be unstable. Only tried and tested frequencies (like the default clocks) are guaranteed to be stable.

Option "linear"

This enables linear addressing, which is the mapping of the entire framebuffer to a high address beyond system memory, so that SVGA bank switching is not necessary. It enhances performance at 256 colors, and is currently required for 16bpp and 32bpp. See section 4 for details.

Membase 0x00e00000 (or a different address)

This sets the physical memory base address of the linear framebuffer. It must be specified in the Device section. It is required for most linear addressing configurations.

Option "favour_bitblt" (5426)

This option will, when the BitBLT engine is enabled, favour its use over framebuffer color expansion functions that are normally faster. This can give a performance improvement on a heavily CPU-loaded system (e.g. running gcc and other programs at the same time) in certain situations.

Option "mmio" (543x)

This enables the use of memory-mapped I/O to talk to the BitBLT engine on the 543x/5429, which is a bit faster. It also enables additional acceleration using memory-mapped I/O.

Option "sw_cursor" (542x/3x)

This disables use of the hardware cursor provided by the chip. Try this if the cursor seems to have problems. In particular, use this when using dot clocks greater than 85 MHz on the 5434 since the chip doesn't fully support the hardware cursor at those clocks.

Option "clgd6225_lcd"

Provides a work-around for problems on the LCD screen of some 62x5 laptop chipsets with maximum white colors.

4 Mode issues

The accelerated 256-color driver uses 256 bytes of scratch space in video memory, and the hardware cursor also uses 1K. Consequently, a 1024x1024 virtual resolution should not be used with a 1Mbyte card.

The use of a higher dot clock frequencies has a negative effect on the performance of graphics operations, especially BitBlt, when little DRAM bandwidth is left for drawing (the amount is displayed during start-up). With default MCLK setting (0x1c) and a 32-bit memory interface, performance with a 65 MHz dot clock can be half of that with a dot clock of 25 MHz. So if you are short on DRAM bandwidth and experience blitting slowness, try using the lowest dot clock that is acceptable; for example, on a 14" or 15" screen 800x600 with high refresh (50 MHz dot clock) is not so bad, with a large virtual screen.

It does not make much sense performance-wise to use the highest clock (85 MHz) for 1024x768 at 76 Hz on a 542x; the card will almost come to a standstill. A 75 MHz dot clock results in 70 Hz which should be acceptable. If you have a monitor that supports 1024x768 at 76 Hz with a 85 MHz dot clock, a standard 5426/5428 based card is a poor match anyway.

5434-based cards with 2Mbyte of memory do much better at high dot clocks; the DRAM bandwidth is basically double that of the 542x series. The 543x chips also make more efficient use of the available DRAM bandwidth.

5 Linear addressing and 16bpp/32bpp modes

Currently the unaccelerated 16-bit and 32-bit pixel support in the SVGA server requires linear addressing. This restriction will hopefully be removed in a future version. Option "linear" can be specified in a depth-specific screen section to enable linear addressing; a MemBase setting (in the device section) is probably also required. There are a number of different card configurations.

If you have a 542x/543x on the ISA bus, and you have 16Mb or more of system memory, linear addressing is impossible. 16bpp is out, sorry. If you have less than 14Mb of memory, you may be able to map the framebuffer at 14Mb, using '`MemBase 0x00e00000`'. That's five zeros after the 'e'. Unfortunately many ISA cards don't support linear addressing.

If you have a 5424/26/28/29 on VESA local bus, the situation is more complicated. There are two different types of cards w.r.t. linear addressing:

- Cards that can only map in the lower 16Mb, like cards on the ISA bus. This is the case with most cards. The same restrictions apply (i.e. you must have less than 16Mb of memory).
- Cards that connect address line A26 and always map at 64Mb + 14Mb. In this case specify '`MemBase 0x04e00000`'. This assumes you have a VLB motherboard implementation that implements A26. This has not been tested; it is not quite clear whether it maps at 0x04000000 or 0x04e00000.

You will probably have to rely on trial and error. If you have less than 16Mb memory, the 'wrong' membase setting will result in no graphics being displayed, but you can probably exit with ctrl-alt-backspace.

If you have >= 16Mb memory, the first type of card (and even the second type with a stupid VLB motherboard) will result in a crash (probably a spontaneous hard reboot).

It may be possible to find out the type by visual inspection. If the card has a pin at A26, it is likely to map beyond 64Mb. To do this, take the card out. At the VESA local bus pins (this is the smaller strip of connector pins at the non-slot side of the card), consider the right side (this is the side of the board where all the chips are mounted). There are 45 pins here. They are numbered 1 to 45, from the inside (i.e. the one nearest to the card end is 45). Counting from the inside, the 17th pin is probably not present, then there are pins at 18-20. The 21st is A30, the 22nd is A28 and the 23rd is A26. So, if we have no pins at at 21-23, the card doesn't map beyond 64Mb. If there's only a gap of two pins at 21 and 22 (or they are both present) and there's a pin at 23, the card does probably map beyond 64Mb. If there's a little logic near that pin on the card, it's more likely.

With a 543x on the local bus things are simpler (the Cirrus Logic windows drivers use it), but it is not quite without problems.

With a card on the PCI bus, there is a PCI configuration register that holds the framebuffer base address. However, the driver currently does not read it out yet, so you'll have to specify the address with MemBase. On one system tested, it mapped at 0xa0000000, and it appeared addresses where assigned in 0x08000000 increments over the PCI devices. A program is most likely available to read out the PCI configuration.

On the VESA local bus, most 543x cards have a default mapping address of 64Mb, with jumper options for 2048Mb and 32Mb. This is probably described in the documentation that came with the card, or look in the MS-Windows system.ini file (something with linearaddr = <offset in megabytes>). These different settings were added by Cirrus Logic after finding that many VLB motherboard implementations don't implement different address pins (however, they failed to handle this smoothly in their initial Windows drivers). The driver assumes a default of 64Mb if MemBase isn't specified. A few examples for MemBase:

```
MemBase 0x02000000      32Mb
MemBase 0x04000000      64Mb
MemBase 0x80000000      2048Mb
```

The 16bpp and 32bpp modes have limited acceleration (scroll, text) which means they can be slow, although a good local bus host interface (like the 543x has) helps a lot. Note that the standard cfb framebuffer code can be very slow for monochrome stipples and bitmaps. 32bpp mode is only supported on a 5434 with 2Mb or more of memory.

In the `XF86Config` `"Screen"` section, a `"Display"` subsection must be defined for each depth that you want to run, with separates Modes and virtual screen size. Example (2Mb of video memory):

```
Section "screen"
    SubSection "Display"
        Depth 8
        Virtual 1152 900
        ViewPort 0 0
        Modes "640x480" "800x600" "1024x768"
        Option "linear"
    EndSubSection
    SubSection "Display"
        Depth 16
        Virtual 1152 900
```

```
        ViewPort 0 0
        Modes "640x480" "800x600" "1024x768"
        Option "linear"
    EndSubSection
    SubSection "Display"
        Depth 32
        Virtual 800 600
        ViewPort 0 0
        Modes "640x480" "800x600"
        Option "linear"
    EndSubSection
EndSection
```

6 The “cl64xx” Driver

The cl64xx driver supports the cl-gd6440 found in many laptops. For example, Nan Tan Computer’s NP9200, NP3600, etc., which are OEM-ed by Sager, ProStar, etc. and Texas Instruments TI4000 series are supported.

The driver works in LCD-only, CRT-only, and the chip’s SimulScan mode which allows one to use both the LCD and external CRT displays simultaneously. The LCD and Simulscan modes’ resolution is 640x480 while, for CRT-only, the standard VESA modes of 640x480, 600x800, and 1024x768 have been tested. Interlaced 1024x768 mode has never been debugged and does not work on the machines tested.

The chip has a documented maximum operating limit for its dot clock that is related to its core voltage. Specifically, for 5.0V the maximum dot clock is 65MHz and for 3.3V the maximum dot clock is 40MHz. The driver checks the core voltage and limits the maximum dot clock to the corresponding value. This translates to a maximum resolution of about 1024x768 at a 60Hz refresh rate. The internal frequency generator can be programmed higher than these limits and is done so during server startup when the clocks are probed which momentarily exceeding the chip’s operating limit. Once a set of valid clocks is obtained, I would recommend using Clocks lines in `XF86Config`. Doing so will also decrease startup time significantly. The clocks may be obtained by running the X server -probeonly (see the XFree86 man page for more information about -probeonly).

The data book indicates that only a configuration of one megabyte of video memory is supported by the chip. This size has been directly set in the driver. If one finds a need, one should be able to override the default size in `XF86Config`. Also, with 1MB of video memory, one should be able to have a virtual screen size of e.g. 1024x1024 and this is possible in CRT-only screen mode. However, whenever the LCD is in use (LCD and SimulScan), the chip uses a portion of upper video ram for its own internal acceleration purposes. Thus, the maximum video memory available for virtual resolution in LCD modes is about 0.75MB e.g. 1024x768. If you set the virtual resolution above this, you will see what might be described as a compressed aliased band when the accelerated area is displayed.

Currently, the driver does not support switching of screen modes among LCD, CRT, and SimulScan, and, at least on the NP9200, the mode must be chosen at OS boot time (e.g. Linux’s LILO) while the BIOS is still active. It should be possible to add screen mode type selection as a ModeLine flag option in `XF86Config` to allow for dynamic screen mode selection from within the X server. Finally, the driver does not currently support any of the powerdown saving features of the chip nor does it shut off the LCD's backlight on screen blank. I hope to implement all these features in future releases.

The primary contact for the cl6440 problems with "cl64xx" driver is Randy Hendry <*randy@sgi.com*>.

7 Trouble shooting with the "cirrus" driver

First of all, make sure that the default modes selected from your `XF86Config` is supported by your monitor, i.e. make sure the horizontal sync limit is correct. It is best to start with standard 640x480x256 with a 25.175 MHz clock (by specifying a single horizontal sync of 31.5) to make sure the driver works on your configuration. The default mode used will always be the first mode listed in the modes line, with the highest dot clock listed for that resolution in the timing section.

Note that some VESA standard mode timings may give problems on some monitors (try increasing the horizontal sync pulse, i.e. the difference between the middle two horizontal timing values).

There is a video signal, but the screen doesn't sync.

You are using a mode that your monitor cannot handle. If it is a non-standard mode, maybe you need to tweak the timings a bit. If it is a standard mode and frequency that your monitor should be able to handle, try to find different timings for a similar mode and frequency combination.

Sparklies/streaks at high dot clocks.

You can try the `"fast_dram"` option, or use a lower dot clock. If it happens when moving something on the screen, try the `"fifo_conservative"` option. If that is not sufficient, the `"noaccel"` option or `"no_bitblt"` will probably help.

'Wavy' screen, horizontal jitter.

You are probably using a dot clock that is too high; it is also possible that there is interference with a close MCLK. Try a lower dot clock. You can also try to tweak the mode timings; try increasing the second horizontal value somewhat. Here's a 65 MHz dot clock 1024x768 mode (about 60 Hz) that might help:

```
 "1024x768"     65      1024 1116 1228 1328     768  783  789  818
```

If you are using programmable clocks with Clockchip `"cirrus"`, try disabling it and using the default set of clocks.

Crash or hang after start-up (probably with a black screen).

Try the `"noaccel"` option. If that works, try Option `"no_bitblt"` for somewhat better performance. Check that the BIOS settings are OK; in particular, disable caching of 0xa0000-0xaffff. Disabling hidden DRAM refresh may also help.

Crash, hang, or trash on the screen after a graphics operation.

This may be related to a bug in one of the accelerated functions, or a problem with the BitBLT engine. Try the `"noaccel"` option, or the `"no_bitblt"` option. Also check the BIOS settings.

'Blitter timeout' messages from the server.

Same as for the above entry.

Screen is 'wrapped' vertically.

This indicates a DRAM configuration problem. If your card has two megabytes of memory, try the `"no_2mb_banksel"` option, or use `videoram "1024"` if you only use 1 Mbyte for the virtual screen.

Corrupted text in terminal window.

This has been reported on non-standard video implementations. Use the `"no_bitblt"` option.

Streaks or hangs with laptop chipset

This can happen if the dot clock is high enough to leave very little bandwidth for drawing (e.g. 40 MHz on a 512K card), and (5422-style) acceleration is used.

Occasional erroneous pixels in text, pixel dust when moving window-frame

Probably related to MCLK setting that is too high (can happen with linear addressing even though banked mode runs OK).

Chipset is not detected.

Try forcing the chipset to a type that is most similar to what you have.

Incorrect little lines (mostly white) appear occasionally

This may be related to a problem with system-to-video-memory BitBLT operations. Try the `"no_imageblt"` option if it annoys you.

Textmode is not properly restored

This has been reported on some configurations. In XFree86 3.1 the SVGA server probe would corrupt a register on the 543x, requiring Chipset line. Normally you should be able to restore the textmode font using a utility that sets it (`setfont`, `runx`, `restorefont` on Linux).

Erratic system behaviour at very high dot clocks

It is possible that high dot clocks on the video card interfere with other components in the system (e.g. disk I/O), because of a bad card and/or motherboard design. It has been observed on some PCI 5428-based cards (which are very rare, since the 5428 doesn't support PCI).

For other screen drawing related problems, try the `"noaccel"` option (if `"no_bitblt"` doesn't help).

If are having driver-related problems that are not addressed by this document, or if you have found bugs in accelerated functions, you can try contacting the XFree86 team (the current driver maintainer can be reached at *hhanemaa@cs.ruu.nl*), or post in the Usenet newsgroup "*comp.windows.x.i386unix*".

8 Driver Changes

Bugs fixed since XFree86 3.1:

- Accelerated 14-pixel wide font drawing bug.
- Server crash when switched to another VC with linear addressing enabled at 8bpp.
- DAC programming lock-up at VC switch at 16bpp on a 542x.
- Driver undoes SVGA probe corruption to 543x register.
- Linear addressing on a 543x with 8bpp (may have caused grief when trying to configure 16/32bpp).
- Hardware cursor color mapping at 16/32bpp.

New features in XFree86 3.1.1:

- Some speed improvements.
- Scrolling/text/fill acceleration at 16/32bpp.
- Support for programmable clocks.
- Support for Memory-Mapped I/O on 543x.
- Support for dot clocks up to 110 MHz on the 5434.
- Support for the CL6440 in the "cl64xx" driver.

`$XConsortium: cirrus.sgml,v 1.4 95/01/27 16:14:36 kaleb Exp $`
`$XFree86: xc/programs/Xserver/hw/xfree86/doc/sgml/cirrus.sgml,v 3.9 1995/01/28 16:03:03 dawes Exp $`