ATI Adapters README file

Marc Aurele La France

2001 April 10

Abstract

This is the README for the XFree86 ATI driver included in this release.

1. Statement of intent

Generally speaking, the driver is intended for all ATI video adapters, providing maximum video function within hardware limitations. The driver is also intended to optionally provide the same level of support for generic VGA or 8514/A adapters. This driver is still being actively developed, meaning that it currently does not yet fully meet these goals.

The driver will provide

- accelerated support if an ATI accelerator is detected *and* the user has not requested that this support be disabled; otherwise
- accelerated support if a non-ATI 8514/A-capable adapter is detected *and* the user has requested such support; otherwise
- unaccelerated SuperVGA support if an ATI VGA-capable adapter is detected; otherwise
- generic VGA support if a non-ATI VGA-capable adapter is detected *and* the user has requested such support.

Thus, the level of support provided not only depends on what the driver detects in the system, but also, on what the user specifies in the XF86Config file. See the **''XF86Config specifications''** section below for details.

If none of the above conditions are met, the ATI driver will essentially disable itself to allow other drivers to examine the system.

2. A note on acceleration

The meaning of ''acceleration'', as used in this document, needs to be clarified. Two of the many components in an accelerator are the CRT controller (CRTC) and the Draw Engine. This is in addition to another CRTC that, generally, is also present in the system (often in the same chip) and typically provides EGA, VGA or SuperVGA functionality.

A CRTC is the component of a graphics controller that is responsible for reading video memory for output to the screen. A Draw Engine is an accelerator component that can be programmed to manipulate video memory contents, thus freeing the CPU for other tasks.

When the VGA CRTC is used, all drawing operations into video memory are the responsibility of the system's CPU, i.e. no Draw Engine can be used. On the other hand, if the accelerator's CRTC is chosen to drive the screen, the Draw Engine can also be used for drawing operations, although the CPU can still be used for this purpose if it can access the accelerator's video memory.

Video acceleration refers to the programming of an accelerator's Draw Engine to offload drawing operations from the CPU, and thus also implies the use of the accelerator's CRTC.

3. Current implementation for ATI adapters

The driver currently supports the SuperVGA capabilities of all ATI adapters except some early Mach8 and Mach32 adapters that do not provide the required functionality. This support works for monochrome, 16-colour and 256-colour video modes, if one of the following ATI graphics controller chips is present:

```
VGAWonder series:  18800, 18800-1, 28800-2, 28800-4, 28800-5, 28800-6
   Mach32 series:  68800-3, 68800-6, 68800AX, 68800LX
   Mach64 series:  88800GX-C, 88800GX-D, 88800GX-E, 88800GX-F, 88800CX,
                   264CT, 264ET, 264VT, 264GT (3D Rage), 264VT-B, 264VT3,
                   264VT4, 264GT-B (3D Rage II), 3D Rage IIc, 3D Rage Pro,
                   3D Rage LT, 3D Rage LT Pro, 3D Rage XL, 3D Rage XC,
                   3D Rage Mobility (including the -M and -P variants)
```

The driver also supports 32K, 64K and 16M-colour modes on the 264xT and 3D Rage series of adapters using the accelerator CRTC (but not the VGA CRTC).

The newer Rage 128 and Radeon chips are not yet supported by this driver. Rage 128's and Radeon's are, however, supported by separate drivers, and owners of such adapters should consult the documentation provided with these drivers. This driver will also invoke the appropriate driver if it finds Rage 128 and/or Radeon adapter(s) in the system.

Adapters based on the above chips have been marketed under a rather large number of names over the years. Among them are:

```
VGAWonder series:  VGAWonder V3, VGAWonder V4, VGAWonder V5, VGAWonder+,
                   VGAWonder XL, VGAWonder XL24, VGAWonder VLB, VGA Basic,
                   VGA Basic 16, VGA Edge, VGA Edge 16, VGA Integra,
                   VGA Charger, VGAStereo F/X, VGA 640, VGA 800, VGA 1024,
                   VGA 1024D, VGA 1024 XL, VGA 1024 DXL, VGA 1024 VLB
    Mach8 series:  Graphics Ultra, Graphics Vantage, VGAWonder GT
                   (None of the 8514/Ultra and 8514 Vantage series is
                    supported at this time)
   Mach32 series:  Graphics Ultra+, Graphics Ultra Pro, Graphics Wonder,
                   Graphics Ultra XLR, Graphics Ultra AXO, VLB mach32-D,
                   PCI mach32-D, ISA mach32
   Mach64 series:  Graphics Xpression, Graphics Pro Turbo, WinBoost,
                   WinTurbo, Graphics Pro Turbo 1600, Video Xpression,
                   3D Xpression, Video Xpression+, 3D Xpression+,
                   3D Charger, Video Charger, WinCharger, All-In-Wonder,
                   All-In-Wonder PRO, 3D Pro Turbo, XPERT@Play,
                   XPERT@Play 98, XPERT@Work, XPERT 98, XPERT LCD,
                   XPERT XL
```

Also, a number of mainboards, laptops and notebooks harbour a Mach32 or Mach64 controller.

VGAWonder, Mach8 and Mach32 ISA adapters are available with or without a mouse.

These adapters are available with a variety of clock generators and RAMDACs. The 264xT and 3D Rage series of chips are integrated controllers, meaning that they include a programmable clock generator and a RAMDAC.

For all but Mach64 adapters, this driver still does not provide support for accelerated drawing to the screen. This means that all drawing is done by the CPU, rather than by any accelerator present in the system. This can make opaque moves, for example, quite ''jerky''. Also, given that IBM 8514/A and ATI Mach8 do not allow CPU access to their frame buffer, the driver will currently ignore these accelerators. Most Mach32 adapters provide both accelerated function and SuperVGA functionality, but the driver currently only uses the VGA.

The driver *does* however support the accelerator CRTC present in all ATI Mach64 adapters. For 256-colour, and higher depth modes, this support will be used by default, although an XF86Config option can be specified to use the SuperVGA CRTC instead. A linear video memory aperture is also available in 256-colour and higher depth modes and enabled by default if a 264xT or 3D Rage controller is detected or, on 88800 controllers, if the accelerator CRTC is used. XF86Config options are available to disable this aperture, or (for non-PCI adapters) enable it or move it to some other address.

By default, the driver provides some acceleration for Mach64 if the accelerator CRTC is used, and modes whose colour depth greater than or equal to 8 are to be used. This support is as yet incomplete and can be disabled entirely with an XF86Config option.

On non-Intel platforms, the driver can, currently, only support PCI Mach64 adapters.

4. Current implementation of generic VGA support for non-ATI adapters

Support for generic VGA with non-ATI adapters is also implemented, but has undergone only limited testing. The driver will intentionally disallow the use of this support with ATI adapters. This support must be explicitly requested through an XF86Config ChipSet specification. This prevents the current VGA generic driver from being disabled.

This driver's generic VGA support is intended as an extension of that provided by the current generic driver. Specifically, within the architectural bounds defined by IBM's VGA standard, this driver will allow the use of any 256-colour mode, and any dot clock frequencies both of which allow for many more mode possibilities.

The driver will enforce the following limitations derived from IBM's original VGA implementation:

- There can only be a set of four (non-programmable) clocks to choose from.
- Video memory is limited to 256kB in monochrome and 16-colour modes.
- Video memory is limited to 64kB in 256-colour modes.
- Interlaced modes are not available.
- Colour depths higher than 8 are not available.

5. XF86Config specifications

The driver recognises a number of XF86Config options. In general, all such options should be specified in a ''Device'' section, and affect only that ''Device'' section.

Those options that affect how the driver associates adapters with ''Device'' sections are described first. The driver will ignore (with a message) a ''Device'' section if the section cannot be associated with exactly one adapter in the system. Similarly, the driver will ignore, or disable, (with a message) any adapter that cannot be associated with exactly one ''Device'' section. Thus, these options will be required in those uncommon cases where such unique associations cannot automatically be made by the driver.

Other options affect the driver's operation once an adapter has been assigned to the ''Device'' section which contains them.

5.1 Driver “ati”

The use of this specification is highly recommended if the “Device” section is to be recognised by the driver. In fact, it is almost (but not quite) mandatory, particularly when using the loader server as it indicates what driver is to be loaded and associated with the “Device” section.

5.2 ChipSet “name”

The default ChipSet name for this driver is “*ati*”. In this case, any ATI adapter can be associated with the “Device” section. If an ATI accelerator is detected and the driver supports it, the accelerator’s CRTC will be used to drive the screen. Otherwise, the driver will programme the adapter’s SuperVGA CRTC.

If “*ativga*” is specified instead, the driver will ignore any ATI accelerator it detects, but otherwise operate as if “*ati*” had been specified. This specification ensures the VGA CRTC is used.

A ChipSet name of “*ibmvga*” causes any VGA-capable adapter in the system to be associated with the “Device” section. It enables the driver’s generic VGA support, but only for non-ATI adapters. If an ATI adapter is associated with the “Device” section, the driver will operate as if “*ativga*” had been specified instead.

A ChipSet name of “*vgawonder*” is equivalent to “*ativga*”, except that only VGAWonder-capable adapters can be assigned to the “Device” section. This specifically excludes the newer integrated Mach64 controllers.

In some PCI or AGP systems, the driver will not, by default, probe for non-PCI Mach32’s or Mach64’s. This is because, before doing any such probe, the driver attempts to determine if the probe can cause a lockup. If the driver has enough information to determine that a lockup would occur, it will skip the probe. In some situations, this determination cannot be accurate, and the driver will err on the side of caution, skipping the probe. Specifying a ChipSet name of “*mach32*” or “*mach64*”, as appropriate, will force the driver to probe for the non-PCI adapter. These ChipSet names should, therefore, only be used when there is in fact such an adapter in the system. They are otherwise equivalent to “*ati*”.

On non-Intel platforms, only “*ati*” and “*mach64*” ChipSet values are operative.

5.3 ChipID & ChipRev specifications

These specifications will cause the driver to associate the “Device” section only with an adapter having the same attributes, or an adapter whose PCI device ID the driver does not recognise. In the second case, these options cause the driver to treat the adapter as if it was one with the specified PCI device ID or revision. ChipID can only be used with Mach32 or Mach64 adapters, and, thus, specifically excludes any other adapter from matching the “Device” section. ChipRev is meaningful only with Mach64 adapters, and then only if ChipID is also specified in the same “Device” section.

5.4 IOBase

This option limits the adapters that can be associated with the “Device” section to the one with the specified I/O base. This option only applies to Mach64 adapters and specifically excludes other adapters.

5.5 BusID

This option limits the adapters that can be associated with the “Device” section to the one with the specified PCI Bus ID. This specification excludes non-PCI adapters.

5.6 Clocks

For the purpose of specifying a clock line in your XF86Config, one of four different situations can occur, as follows.

Those configuring the driver's generic VGA support for a non-ATI adapter, can skip ahead to the **"Clocks for non-ATI adapters"** section below. Those *not* trying to configure the driver for a Mach64 adapter, can skip ahead to the **"Clocks for fixed clock generators on ATI adapters"** section below.

The very earliest Mach64 adapters use fixed (i.e. non-programmable) clock generators. Very few of these (mostly prototypes) are known to exist, but if you have one of these, you can also skip ahead to the **"Clocks for fixed clock generators on ATI adapters"** section below.

The two cases that are left deal with programmable clock generators, which are used on the great majority of Mach64 adapters.

If you are uncertain which situation applies to your adapter, you can run a clock probe with the command "`X -probeonly`".

5.6.1 Clocks for supported programmable clock generators

At bootup, video BIOS initialisation programmes an initial set of frequencies. Two of these are reserved to allow the setting of modes that do not use a frequency from this initial set. One of these reserved slots is used by the BIOS mode set routine, the other by the particular driver used (e.g. MS-Windows, AutoCAD, X, etc.). The clock numbers reserved in this way are dependent on the particular clock generator used by the adapter.

The driver currently supports all programmable clock generators known to exist on Mach64 adapters. In this case, the driver will completely ignore any XF86Config clock specification, and programme the clock generator as needed by the modes used during the X session.

5.6.2 Clocks for unsupported programmable clock generators

This case is unlikely to occur, but is documented for the sake of completeness.

In this situation, the driver will probe the adapter for clock frequencies unless XF86Config clocks are already specified. In either case, the driver will then attempt to normalise the clocks to one of the following specifications:

```
BIOS setting 1:

  Clocks   0.000 110.000 126.000 135.000  50.350  56.640  63.000  72.000
           0.000  80.000  75.000  65.000  40.000  44.900  49.500  50.000
           0.000  55.000  63.000  67.500  25.180  28.320  31.500  36.000
           0.000  40.000  37.500  32.500  20.000  22.450  24.750  25.000


BIOS setting 2:

  Clocks   0.000 110.000 126.000 135.000  25.180  28.320  31.500  36.000
           0.000  80.000  75.000  65.000  40.000  44.900  49.500  50.000
           0.000  55.000  63.000  67.500  12.590  14.160  15.750  18.000
           0.000  40.000  37.500  32.500  20.000  22.450  24.750  25.000


BIOS setting 3:

  Clocks   0.000   0.000   0.000   0.000  25.180  28.320   0.000   0.000
           0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
           0.000   0.000   0.000   0.000  12.590  14.160   0.000   0.000
           0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
```

If the driver matches the clocks to the third setting above, functionality will be *extremely* limited (assuming the driver works at all).

5.6.3 Clocks for fixed clock generators on ATI adapters

This section applies to all VGAWonder and Mach32 adapters, and to early Mach64 prototypes.

One of the following clocks specifications (or an initial subset thereof) can be used depending on what the adapter uses to generate dot clocks:

```
Crystals (VGA Wonder V3 and V4 adapters only):

  Clocks  50.000  56.644   0.000  44.900  44.900  50.000   0.000  36.000
          25.000  28.322   0.000  22.450  22.450  25.000   0.000  18.000
          16.667  18.881   0.000  14.967  14.967  16.667   0.000  12.000
          12.500  14.161   0.000  11.225  11.225  12.500   0.000   9.000


ATI 18810 clock generator:

  Clocks  30.240  32.000  37.500  39.000  42.954  48.771   0.000  36.000
          40.000   0.000  75.000  65.000  50.350  56.640   0.000  44.900
          15.120  16.000  18.750  19.500  21.477  24.386   0.000  18.000
          20.000   0.000  37.500  32.500  25.175  28.320   0.000  22.450
          10.080  10.667  12.500  13.000  14.318  16.257   0.000  12.000
          13.333   0.000  25.000  21.667  16.783  18.880   0.000  14.967
           7.560   8.000   9.375   9.750  10.739  12.193   0.000   9.000
          10.000   0.000  18.750  16.250  12.586  14.160   0.000  11.225


ATI 18811-0 and ATI 18812-0 clock generators:

  Clocks  30.240  32.000 110.000  80.000  42.954  48.771  92.400  36.000
          39.910  44.900  75.000  65.000  50.350  56.640   0.000  44.900
          15.120  16.000  55.000  40.000  21.477  24.386  46.200  18.000
          19.955  22.450  37.500  32.500  25.175  28.320   0.000  22.450
          10.080  10.667  36.667  26.667  14.318  16.257  30.800  12.000
          13.303  14.967  25.000  21.667  16.783  18.880   0.000  14.967
           7.560   8.000  27.500  20.000  10.739  12.193  23.100   9.000
           9.978  11.225  18.750  16.250  12.588  14.160   0.000  11.225


ATI 18811-1 and ATI 18811-2 clock generators:

  Clocks 135.000  32.000 110.000  80.000 100.000 126.000  92.400  36.000
          39.910  44.900  75.000  65.000  50.350  56.640   0.000  44.900
          67.500  16.000  55.000  40.000  50.000  63.000  46.200  18.000
          19.955  22.450  37.500  32.500  25.175  28.320   0.000  22.450
          45.000  10.667  36.667  26.667  33.333  42.000  30.800  12.000
          13.303  14.967  25.000  21.667  16.783  18.880   0.000  14.967
          33.750   8.000  27.500  20.000  25.000  31.500  23.100   9.000
           9.978  11.225  18.750  16.250  12.588  14.160   0.000  11.225


ICS 2494-AM clock generators (found on some Dell motherboards):

  Clocks  75.000  77.500  80.000  90.000  25.175  28.322  31.500  36.000
         100.000 110.000 126.000 135.000  40.000  44.900  50.000  65.000
          37.500  38.750  40.000  45.000  12.588  14.161  15.750  18.000
          50.000  55.000  63.000  67.500  20.000  22.450  25.000  32.500
          25.000  25.833  26.667  30.000   8.392   9.441  10.500  12.000
          33.333  36.667  42.000  45.000  13.333  14.767  16.667  21.667
          18.750  19.375  20.000  22.500   6.294   7.081   7.875   9.000
          25.000  27.500  31.500  33.750  10.000  11.225  12.500  16.250
```

VGAWonder VLB, VGA 1024 VLB, Mach32 and Mach64 owners should only specify up to the first 32 frequencies. Any more will be ignored.

Other clock generators that have been used on ATI adapters (which can all be said to be clones of one of the above) might generate non-zero frequencies for those that are zero above, or vice-versa.

The order of the clocks *is* very important, although the driver will reorder the clocks if it deems it appropriate to do so. Mach32 and Mach64 owners should note that this order is different than what they would use for previous XFree86 accelerated servers.

5.6.4 Clocks for non-ATI adapters

If no clocks are specified in the XF86Config, the driver will probe for four clocks, the second of which will be assumed to be 28.322 MHz. The first clock will typically be 25.175 MHz, but there are exceptions. You can include up to four clock frequencies in your XF86Config to specify the actual values used by the adapter. Any more will be ignored.

5.7 Option "crt_screen"

This specification is only effective when the driver detects that the adapter's BIOS has initialised both the digital flat panel and CRT interfaces. In such a situation, the driver will normally drive both the panel and the CRT. This specification causes the driver to disable the digital flat panel and display the screen image on the CRT instead.

5.8 Option "noaccel"

By default, the driver will accelerate draw operations if a Mach64 CRTC is used to drive the display. As implemented in this driver, acceleration does not require a linear video memory aperture. This option disables this acceleration.

5.9 Option "nolinear"

By default, the driver will enable a linear video memory aperture for 256-colour and higher depth modes if it is also using a Mach64 accelerator CRTC or an integrated Mach64 graphics chip. This option disables this linear aperture.

On non-Intel platforms, the driver requires a linear aperture and, so, this option is ignored.

5.10 Option "HWCursor" and Option "SWCursor"

Option *"HWCursor"*, which is the default, specifies that hardware facilities are to be used to paint the mouse pointer on the screen. Option *"SWCursor"* specifies that the mouse pointer is to by drawn by software, which is much slower. If both options are specified, option *"SWCursor"* prevails. Currently, these options are only acted upon for 256-colour or higher depth modes, if a Mach64 accelerator CRTC, or a Mach64 integrated controller is being used. In all other situations, a software cursor will be used.

5.11 Option "SilkenMouse"

This option is only acted upon when a hardware cursor is being used. It specifies that the cursor's position on the screen is to be updated as quickly as possible when the mouse is moved. This is the default behaviour. If this option is negated, the cursor may lag the mouse when the X server is very busy.

5.12 Option "shadowfb"

If this option is enabled, the driver will cause the CPU to do each drawing operation first into a shadow frame buffer in system virtual memory and then copy the result into video memory. If this option is not active, the CPU will draw directly into video memory. Enabling this option is beneficial for those systems where reading from video memory is, on average, slower than the corresponding read/modify/write operation in system virtual memory. This is normally the case for PCI or AGP adapters, and, so, this option is enabled by default. For other bus types, the default behaviour is to disable this option.

Note that, due to various limitations, this option is forcibly disabled when a linear video memory aperture is not enabled, when the frame buffer depth is less than 8, or when acceleration is used.

5.13 Option "dpms"

This option enables the driver's support for VESA's Display Power Management Specification.

5.14 Option "backingstore"

This is not specifically a driver option. It is used to enable the server's support for backing store, a mechanism by which pixel data for occluded window regions is remembered by the server thereby alleviating the need to send expose events to X clients when the data needs to be redisplayed.

5.15 MemBase address

This specification is only effective for non-PCI Mach64 adapters, and is used to override the CPU address at which the adapter will map its video memory. Normally, for non-PCI adapters, this address is set by a DOS install utility provided with the adapter. The MemBase option can also be used to enable the linear aperture in those cases where ATI's utility was not, or can not be, used.

For PCI and AGP adapters, this address is determined at system bootup according to the PCI Plug'n'Play specification which arbitrates the resource requirements of most devices in the system. This means the driver can not easily change the linear aperture address.

5.16 Option "ReferenceClock" "frequency"

This option is only applicable to non-Intel platforms, where an adapter BIOS is not available to the driver. The option specifies the reference frequency used by the adapter's clock generator. The default is 14.318 MHz, and other typical values are 28.636, or 29.5 MHz.

5.17 ClockChip "name"

This option is only applicable to non-Intel platforms, where an adapter BIOS is not available to the driver, and the driver cannot reliably determine whether the clock generator the adapter uses is a variant of an ATI 18818 (a.k.a. ICS 2595) or an unsupported clock generator. The only values that are acted upon are *"ATI 18818-0"* or *"ATI 18818-1"*. From this specification, the driver derives a reference divider of 43 or 46 (respectively) for use in clock programming calculations. The driver's default behaviour, in this case, is to assume an unsupported clock generator, which means it will treat it as a fixed-frequency clock generator, as described under the heading **"Clocks for unsupported programmable clock generators"** above.

6. Video modes

Mode timings can be derived from the information in XFree86's doc subdirectory. However, it is no longer required to specify such timings in an XF86Config's "Monitor" section(s), if only standard mode timings are to be used. The server automatically inserts VESA standard mode timings in every "Monitor" section, and these modes will be checked first for mode constraints (monitor sync tolerances, video memory size, etc.).

Furthermore, it is also no longer required to specify mode names in "Display" subsections. Should no mode names be specified (or those specified do not yield a usable mode), the server will automatically select as a default resolution the largest usable mode, whether or not the chosen mode is specified in the corresponding "Monitor" section.

For a CRT monitor, you should still specify horizontal sync and vertical refresh rates in the corresponding ‘‘Monitor’’ section. Otherwise, the server will default these to those of a VGA monitor, which will limit resolutions to not much more than 640x480 at 72 Hz. For a digital flat panel, any sync tolerances should be removed from the corresponding ‘‘Monitor’’ section. The driver will automatically calculate these from the mode that is active on server entry. The driver also inserts timings for a mode called *"Native panel mode"* that represents the panel’s native resolution.

7. Known problems and limitations

There are several known problems or limitations related to the XFree86 ATI driver. They include:

- A number of system lockups and blank screens have been reported when using PCI Mach64 adapters. The great majority of these problems have been found to be due to system aspects that are unrelated to this driver. As of this writing, these problems can be divided into two general areas:

  Improper mouse protocol specification with some recent mice. Try different protocol specifications or another mouse.

  A system conflict with APM. This problem is Linux-specific. There is a bug in kernels 2.0.31 or earlier that prevents proper APM operation. Upgrade to a more recent kernel or disable APM support in the kernel.

- When using a Mach64’s accelerator CRTC, the virtual resolution must be less than 8192 pixels wide. The VGA CRTC further limits the virtual resolution width to less than 4096 pixels, or to less than 2048 pixels for adapters based on 18800-x’s (with 256kB of memory) and on Mach64 integrated controllers. These are hardware limits that cannot be circumvented.
- Virtual resolutions requiring more than 1MB of video memory (256kB in the monochrome case) are not supported by the VGA CRTC on 88800GX and 88800CX adapters. This is a hardware limit that cannot be circumvented.
- Due to hardware limitations, doublescanned modes are not supported by the accelerator CRTC in 88800GX, 88800CX, 264CT and 264ET adapters.
- The ‘‘VScan’’ modeline parameter is only supported when using the VGA CRTC.
- Interlaced modes are not supported on 18800-x and 28800-x adapters when using a virtual resolution that is 2048 pixels or wider. When using a 18800-x with 256kB of video memory in 256-colour modes, this limit is reduced to 1024. This is yet another hardware limitation that cannot be circumvented.
- Video memory banking does not work in monochrome and 16-colour modes on 18800-x adapters. This appears to be another hardware limit, but this conclusion cannot be confirmed at this time. The driver’s default behaviour in this case is to limit video memory to 256kB.
- Video memory corruption can still occur during mode switches on 18800-x adapters. Symptoms of this problem include garbled fonts on return to text mode, and various effects (snow, dashed lines, etc) on initial entry into a graphics mode. In the first case, the workaround is to use some other means of restoring the text font. On Linux, this can be accomplished with the kbd or svgalib packages. In the second case, xrefresh(1) will usually clean up the image. No complete solution to this problem is currently known. It appears this corruption occurs due to either video memory bandwidth or RAMDAC limitations, and so the driver will limit mode clocks to 40MHz.
- There is some controversy over what the maximum allowed clock frequency should be on 264xT and 3D Rage adapters. For now, clocks will, by default, be limited to 80MHz, 135MHz, 170MHz, 200MHz or 230MHz, depending on the specific controller. This limit can only be increased (up to a driver-calculated absolute maximum) through the DACSpeed

specification in XF86Config. Be aware however that doing so is untested and might damage the adapter.

- Except as in the previous items, clocks are limited to 80MHz on most adapters, although many are capable of higher frequencies. This will be fixed in a future release.

Support for the following will be added in a future release:

- Mach32's accelerator CRTC. This support is the first step towards accelerated support for Mach32's, Mach8's, 8514/A's and other clones.
- Colour depth greater than 8 on non-integrated controllers, where permitted by the hardware.
- Mach32, Mach8 and 8514/A Draw Engines.
- Hardware cursors where implemented by hardware. This has already been done for Mach64 integrated controllers.
- TVOut, i.e. the ability to use a television screen as a monitor.
- Motion Video, i.e. displaying an asynchronous data stream (TV signal, DVD, etc.) in a window or full-screen.
- 3D operations.

8. Reporting problems

If you are experiencing problems that are not already recorded in this document, first ensure that you have the latest current release of this driver and XFree86. Check the server's log (usually found in /var/log/XFree86.0.log) and ftp://ftp.xfree86.org/pub/XFree86 if you are uncertain.

Secondly, please check XFree86's doc directory for additional information.

Thirdly, a scan through the comp.windows.x.i386unix and comp.os.linux.x newsgroups using your favourite archiving service can also prove useful in resolving problems.

If you are still experiencing problems, you can send me *non-HTMLised* e-mail at <tsi@xfree86.org>. Please be as specific as possible when describing the problem(s), and include an *unedited* copy of the server's log and the XF86Config file used.

9. Driver history

The complete history of the driver is rather cloudy. The following is more than likely to be incomplete and inaccurate.

Apparently, Per Lindqvist first got a driver working with an early ATI adapter under X386 1.1a. This original driver might have actually been based on a non-functional ATI driver written by Thomas Roell (currently of Xi Graphics).

Then Doug Evans (*dje@cygnus.com*) added support for the ATI VGA Wonder XL, trying in the process to make the driver work with all other ATI adapters available at the time.

Rik Faith (*faith@precisioninsight.com*) obtained the X11R4 driver from Doug Evans in the summer of 1992 and ported the code to the X386 part of X11R5. This subsequently became part of XFree86.

I (Marc Aurele La France) took over development and maintenance of the driver in the fall of 1993 after Rik got rid of his VGA Wonder adapter.

10. Driver versions

Due to the introduction of loadable drivers in XFree86 4.0, it has become necessary to track driver versions separately. Driver releases use the following version numbering scheme.

Version 1 of this driver is the one I inherited from Rik Faith. This is the version found in XFree86 2.0 and 2.1.

Version 2 is my first rewrite of this code which only ended up being a partially unsuccessful attempt at generalising the driver for all VGA Wonder, Mach32, and early Mach64 adapters. Various releases of this version of the driver can be found in XFree86 2.1.1, 3.1, 3.1.1 and 3.1.2.

Version 3 represents my second rewrite (although a rather lame one as rewrites go). Into version 3, I introduced clock programming for Mach64 adapters and merged in the old ati_test debugging tool. This is the version found in XFree86 3.2, 3.3 and 3.3.1.

Version 4 is a rather major restructuring of version 3, which became larger than I could comfortably handle in one source file. This is the version found in XFree86 3.3.2, 3.3.3, 3.3.3.1, 3.3.3.2, 3.3.4, 3.3.5 and 3.3.6.

Version 5 is an almost complete restructuring of version 4 to fit in the newer driver API of XFree86 4.0 and later.

The introduction of version 6 is a first swipe at porting the driver to non-Intel architectures.

CONTENTS

`$XFree86: xc/programs/Xserver/hw/xfree86/doc/sgml/ati.sgml,v 3.39 2002/01/16 16:25:54 tsi Exp $`