

Driver Monoculture:

- Older hardware is important.
- Don't expect users to buy specific hardware.
- Don't thin market out further.
- Don't tie the fate of a project to a single “vendor”.
- Recent “vendor” tactics are not that commendable.

xf86-video:

- Existing framework
- Well established
- Big install base
- Supports many different vendors & devices

xf86-video state:

- Most drivers are orphaned.
- Opaque code, different styles.
- DIX changes are like walking a tightrope.
- DIX and DDIXen have become solidified. Natural evolution has stopped.
- Many things are either broken or extremely crufty.

xf86-video cruft examples:

- ISA.
- banked memory.
- non-programmable dotclocks.
- forces old (IBM) VGA mode limitations.
- drivers' dependence on vgaHW helpers.
- rigid mode validation.
- ...

Clocks:

- Multiple clockranges are only used by tseng.
- Remove non-programmable clocks.
- Remaining clockRange members -> scrnInfoPtr

```
/* the list of clock ranges */
typedef struct x_ClockRange {
    struct x_ClockRange *next;
    int                minClock; /* (kHz) */
    int                maxClock; /* (kHz) */
    int                clockIndex; /* -1 for programmable clocks */
    Bool               interlaceAllowed;
    Bool               doubleScanAllowed;
    int                ClockMulFactor;
    int                ClockDivFactor;
    int                PrivFlags;
} ClockRange, *ClockRangePtr;
```

VGA Limitations:

- problem:
H/VBlankStart/End are altered to suit antique VGA limitations.
- solution:
move CRT register limitation checking to ValidMode.

```
/*  
 * XXX  
 *  
 * The following is taken from VGA, but applies to other cores as well.  
 */  
p->CrtcVBlankStart = min(p->CrtcVSyncStart, p->CrtcVDisplay);  
p->CrtcVBlankEnd = max(p->CrtcVSyncEnd, p->CrtcVTTotal);  
if ((p->CrtcVBlankEnd - p->CrtcVBlankStart) >= 127) {  
    /*  
     * V Blanking size must be < 127.  
     * Moving blank start forward is safer than moving blank end  
     * back, since monitors clamp just AFTER the sync pulse (or in  
     * the sync pulse), but never before.  
     */  
    p->CrtcVBlankStart = p->CrtcVBlankEnd - 127;  
    /*  
     * If VBlankStart is now > VSyncStart move VBlankStart  
     * to VSyncStart using the maximum width that fits into  
     * VTTotal.  
     */  
    if (p->CrtcVBlankStart > p->CrtcVSyncStart) {  
        p->CrtcVBlankStart = p->CrtcVSyncStart;  
        p->CrtcVBlankEnd = min(p->CrtcHBlankStart + 127, p->CrtcVTTotal);  
    }  
}  
p->CrtcHBlankStart = min(p->CrtcHSyncStart, p->CrtcHDisplay);  
p->CrtcHBlankEnd = max(p->CrtcHSyncEnd, p->CrtcHTotal);  
  
if ((p->CrtcHBlankEnd - p->CrtcHBlankStart) >= 63 * 8) {  
    /*  
     * H Blanking size must be < 63*8. Same remark as above.  
     */  
}
```

vgaHW helper dependence:

Most drivers depend on vgaHW helpers like vgaHWInit, vgaHWLock, vgaHWProtect, vgaHWSave.

- VGA specific and seriously deficient.
- Reduces driver portability and readability.
- Code reuse advantage is very limited.
- Full “WriteMode” implementations are better.
- Change on a driver-per-driver basis.

Mode Revalidation.

- Problem:
 - modes are validated at PreInit.
 - failed for whatever reason? -> pruned and lost forever.
 - new modes -> XvidMode or restart.
- Solution: twofold mode validation
 - first stage: PreInit: all modes that are completely impossible are pruned. (MODECHECK_INITIAL + MODECHECK_FINAL)
 - second stage: checked again before calling ScreenInit or SwitchMode (or WriteMode). (MODECHECK_WRITE)
 - painless transition.

VBE:

- easy.
- taken for granted.
- non free.
- full vendor dependence.
- not always benevolent (i8xx).
- not impossible to work around.

Outputs: ye olde days

external RAMDACs:

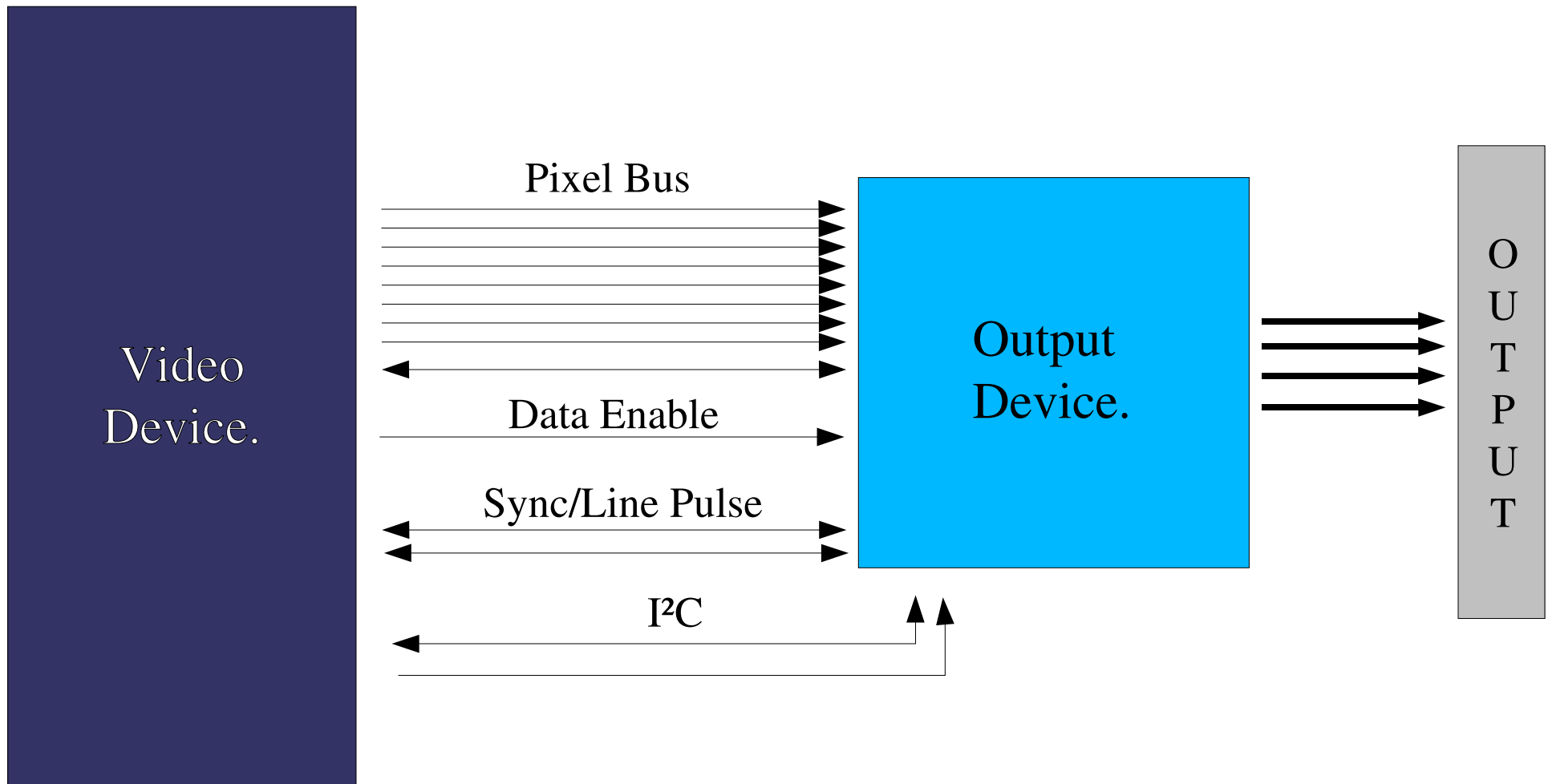
- well supported before Xfree86 4.0 but mostly broken now.
- common before ~1995
- rare on PCI devices:
 - tseng ET4000W32p (stg1703, ch8398, ...)
 - S3 until TRIO32 (stg1703, ch8398, ...)
 - old mga millenium (tvp3026)
 - rendition rv1000 (tvp3026)
 - 3dfx voodoo 1,2 (pass through)
- No longer worthwhile -> VBE

Outputs: today

- at least 1 RAMDAC integrated. (CRT).
- other output devices are common:
 - Secondary CRT
 - Panel
 - LVDS
 - TMDS (DVI/HDMI)
 - TV

either onboard or through highly modular external devices.

External output devices:



Chrontel:

- CH7xxx:
 - TV: s-video / composite / RGB
 - LVDS
 - TMDS
 - shared devices: dual inputs, LVDS + TV and TMDS + TV
- free datasheets.
- everywhere

Conexant:

- BT86x and follow-ups CX2587x/89x/90x.
 - TV encoders
 - latest devices also do TMDS.
- free datasheets (except for CX2589x and 90x)
- BT868/9 was very widespread.

Focus:

- FS45x
 - TV encoders
- datasheets upon request (NDA?).
- found on some Xbox version and on some EPIA board.

NatSemi:

- DS90C2501/C38x
 - LVDS encoders
- free datasheets.
- availability?

Philips:

- SAA710x/712x
 - shared TV encoder/decoders.
 - TV encoders.
- free datasheets.
- usually as part of framegrabber or tuner chip.

Silicon Image

- SIL1xx/1xxx
 - TMDS, for parallel pixelbus and SDVO
- no datasheets.
- ADD/ADD2 cards
- IP hog

SiS:

- SiS301/302
 - TV/CRT
 - TMDS + TV/CRT
 - LVDS + TV/CRT
- no datasheets.
- only with SiS products.

VIA:

- VT162x/VT163x
 - TV
 - TMDS
 - LVDS
- no datasheets (except VT1621)
- only with VIA/S3 products.

What does this mean?

Vendors can:

- mix and match output types.
- create wildly different products.
- provide add on boards.
- from the same ASIC run.

Vendors have it cheap and hasslefree.

But why aren't we exploiting this modularity?

Needed infrastructure:

- Detection + initial setup.
- Connection sensing.
- Mode validation.
- Mode setting.
- Power management.
- Register save/restore.

Add a pI2CDev, some bookkeeping (sync/clock direction, ownership info, power/output states) and a pointer to device specific Privates.

Current unichrome implementation: (WIP & TV only)

```
/* TV entries */
int      TVEncoder;
int      TVOutput;
Bool     TVActive;
I2CDevPtr TVI2CDev;
int      TVType;
Bool     TVDotCrawl;
int      TVDeflicker;
CARD8    TVRegs[0xFF];

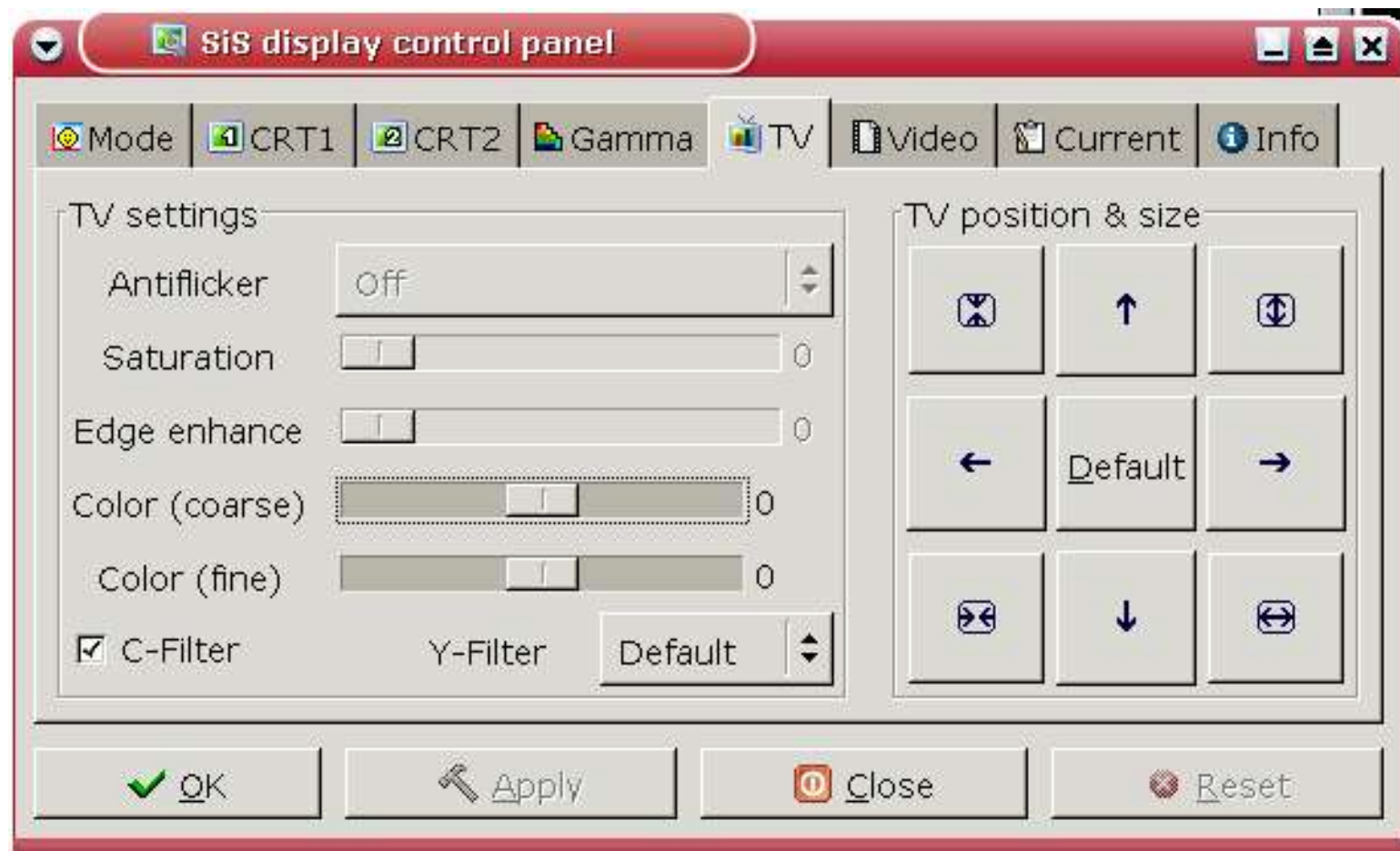
/* TV Callbacks */
void (*TVSave) (ScrnInfoPtr pScrn);
void (*TVRestore) (ScrnInfoPtr pScrn);
Bool (*TVDACSense) (ScrnInfoPtr pScrn);
ModeStatus (*TVModeValid) (ScrnInfoPtr pScrn, DisplayModePtr mode);
void (*TVModeI2C) (ScrnInfoPtr pScrn, DisplayModePtr mode);
void (*TVModeCrtc) (ScrnInfoPtr pScrn, DisplayModePtr mode);
void (*TVPower) (ScrnInfoPtr pScrn, Bool On);
DisplayModePtr TVModes;
void (*TVPrintRegs) (ScrnInfoPtr pScrn);
```

So what about VBE?:

- With output devices abstracted there is no more need to be that scared.
- Crtc is usually documented or implemented (at least primary).
- remaining work is:
 - digital scaler (for LVDS or LFP).
 - glue: i2c, bus setup.

Control of output devices:

- Often done by way of a separate Xv port.
- Thomas Winischhofer has written a special extension for sis driver: SISCTRL



Needed: server wide attribute system

- largely “all encapsulating”
- handles mode control entirely (superceding XVM/RandR)
- handles all aspects of output and input devices.
- gamma
- Xv attributes
- ...

Final rant: MIT license

- Doesn't provide:
 - everything except:
- Does provide:
 - Line in the sand: Copyright (c) <year> <copyright holders>
 - Freedom to ...
 - **Disclaimer:** THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Typical: THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL VIA, S3 GRAPHICS, AND/OR ITS SUPPLIERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Extreme: ADOBE MAKES NO REPRESENTATIONS ABOUT THE SUITABILITY OF THE SOFTWARE FOR ANY PURPOSE. IT IS PROVIDED "AS IS" WITHOUT EXPRESS OR IMPLIED WARRANTY. ADOBE DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL ADOBE BE LIABLE TO YOU OR ANY OTHER PARTY FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE, STRICT LIABILITY OR ANY OTHER ACTION ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. ADOBE WILL NOT PROVIDE ANY TRAINING OR OTHER SUPPORT FOR THE SOFTWARE.