

Nouveau Status Update

Stéphane Marchesin
<marchesin@icps.u-strasbg.fr>

Who are you guys?

- The same guys as usual
- But this year, one full-time guy
 - At RedHat

Hardware acceleration features

- What is available
- What do we use already
- What we could add support for

		video	vertex	fragment	
nv04 TNT	Over lay 1	Blitter	None	None	Tex Comb.
nv10 GF 1/2 mx	Over lay 2		None	Fixed	Reg Comb.
nv17 GF 4 mx			H.262		
nv20 GF 3/4 Ti			H.262	Prog 1	Prog 1
nv30 GF 5			H.262		
nv40 GF 6/7			H.262	Prog 2	Prog 2
nv50 GF 8/9			H.264	Prog 3	

Driver features

- What we do

	video		modeset	2D	3D
nv04 TNT	Ovl Blitter	None	Single Head	Basic	None
nv10 GF 1/2 mx	Ovl Blitter	None		Composite	
nv17 GF 4 mx		None			
nv20 GF 3/4 Ti		None	Gears		
nv30 GF 5	Ovl tex Blitter	None	Dual Head	Some games	
nv40 GF 6/7	Tex	H.262		Some games	
nv50 GF 8/9	Tex	None		Gears	

Did you know (hardware) ?

- Nvidia uses some kind of high level approach to hardware design
 - Hardware features are wrapped in objects
 - 2D, 3D, overlay, DMA...
 - Memory protection uses objects, too
 - Table holding all the objects
 - Very weird the first time you look at it

The state of 2D

- Performance has been increasing over the last year
 - Thanks to core EXA improvements, mostly

2D plans

- Still needs a little work on nv50
 - nv50's buffer tiling
 - Need to change core EXA to support this
- Otherwise, it works, so don't touch it

Did you know (2D accel) ?

- Nouveau targets 10 years worth of graphics hardware
 - Don't throw away your old stuff just yet!
 - Modern desktop acceleration is possible on old cards

The state of 3D

- All cards have some form of a gallium driver
- Moved into mainline gallium now
- Will move into mainline mesa soon
- Not ready for global consumption

3D plans

- Finish the drivers
 - nv30/nv40 mostly there
 - nv04/nv20/nv50 on their way
- LLVM + Gallium3D integration
 - Tomorrow

Did you know (3D) ?

- Nvidia hardware uses OpenGL values and ordering at the hardware level for most things
 - Blending modes
 - Primitives
 - ...
- This helps for reverse engineering!

State of video decoding

- Summer of code project
 - Goal: achieve hardware independent video decoding at the Gallium 3d level
- A new gallium state tracker
- Does all video decoding using shaders
- For now H.262 (MPEG2)
- Should work with all gallium-supported cards

Video decoding plans

- Using a shader only model
 - Move to H.264
- Using fixed pipe decoding stages
 - Add support for fixed pipe stages
 - Since nv17 some form of fixed pipe video decoding is included in nvidia cards
 - RE the video decoding bits
 - Varies much from card to card
 - Is it worth the trouble ?

Did you know (video decoding) ?

- The nvidia hardware overlay uses a filtering table for upscaling
 - We never quite figured it out

Kernel plans

- Memory management
 - TTM with GEM interface (like radeon)
 - Will help the 3D driver become usable
 - Out of memory situations

Kernel plans

- Support kernel modesetting
- Comes right after memory management
- Last bit before the DRM interface is stable
 - Then we can release

Kernel plans

- Power management/clock frequency management
 - Proper clock frequencies should be used
 - Not set at bootup
 - Needed for best performance, and best battery life
 - RedHat interested in power management

Kernel plans

- Release ?
 - Requires stable kernel interfaces
 - Could be this year if things go well

Conclusions

- Try to replace «nv» this year
- Funny side ideas
 - Emulate an nvidia GPU
 - For example in Qemu
 - Add GPGPU support
 - Port to xbox
 - We get requests for windows CE ports...

Questions ?