DRM / KMS

DRM

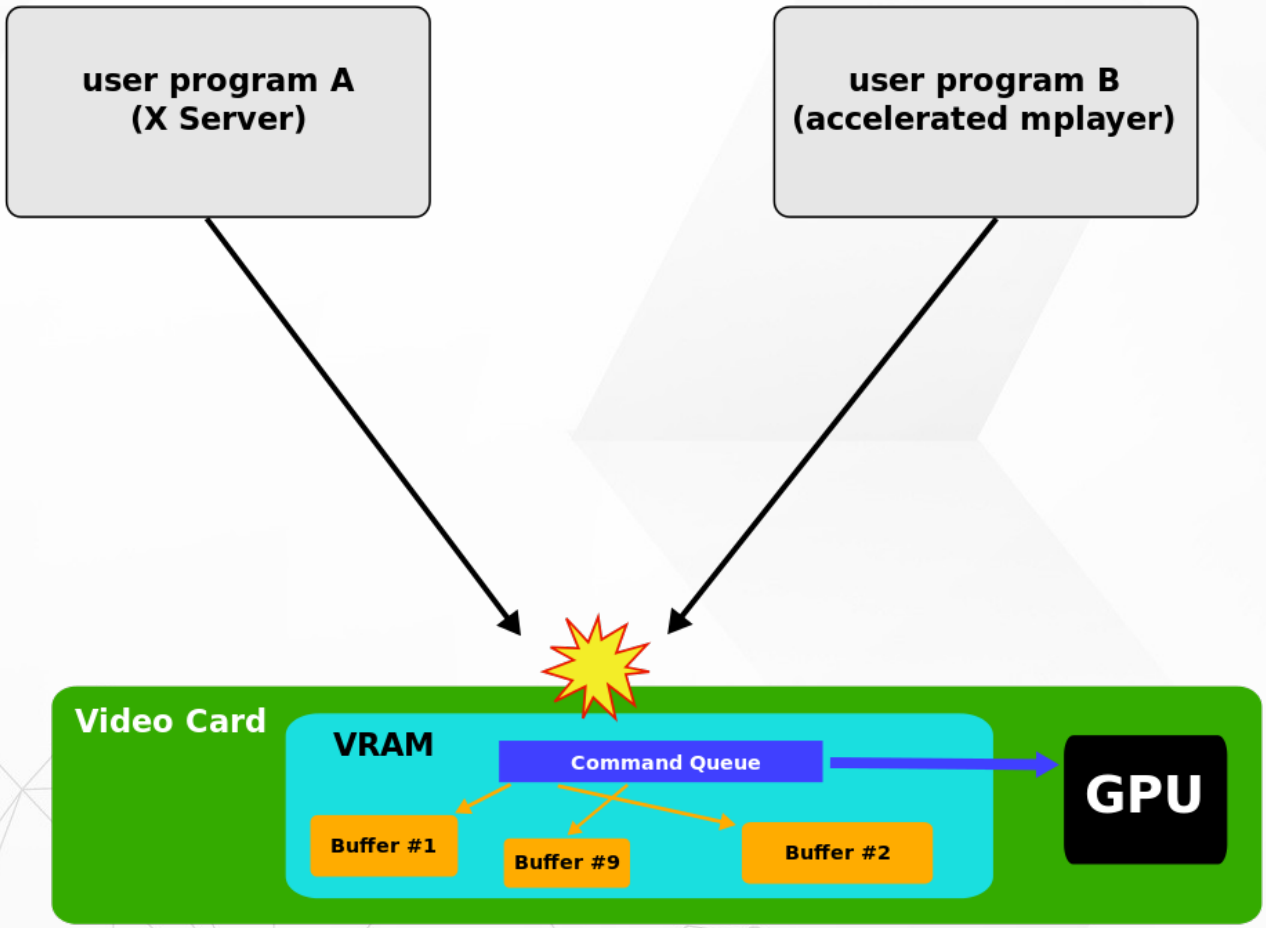
- Direct Rendering Manager
  - Management of buffers and free space within that memory.
  - Solve Frame buffer driver cannot be used GPU and multi-user process.

DRM

- DRM consists of
  - KMS : Kernel Mode Setting
    - Change resolution and depth
  - DRI : Direct Rendering Infrastructure
    - Interfaces to access hardware directly
  - GEM : Graphics Execution Manager
    - Buffer management
  - DRM Driver in kernel side
    - Access hardware

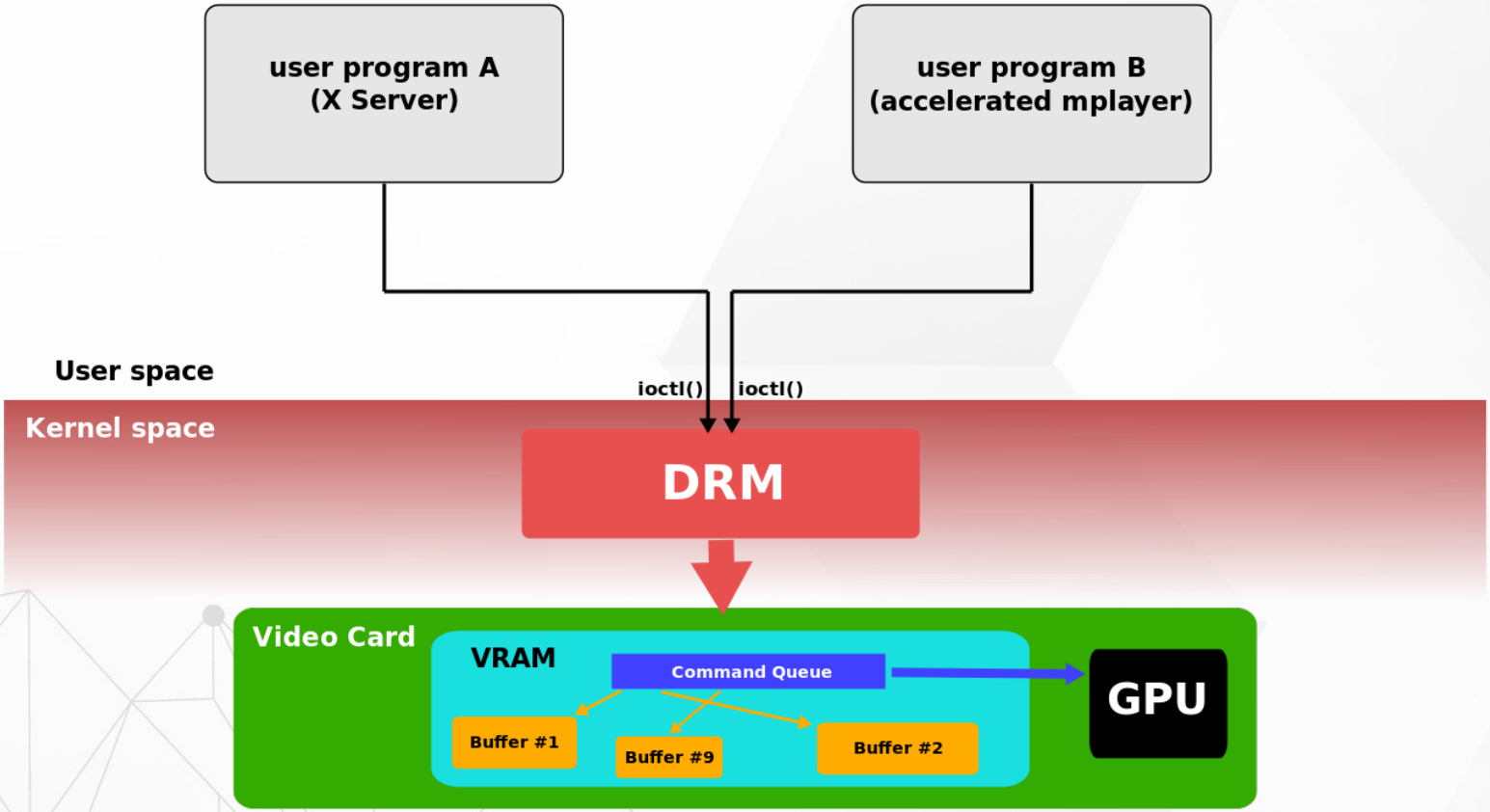
DRM

If no use DRM



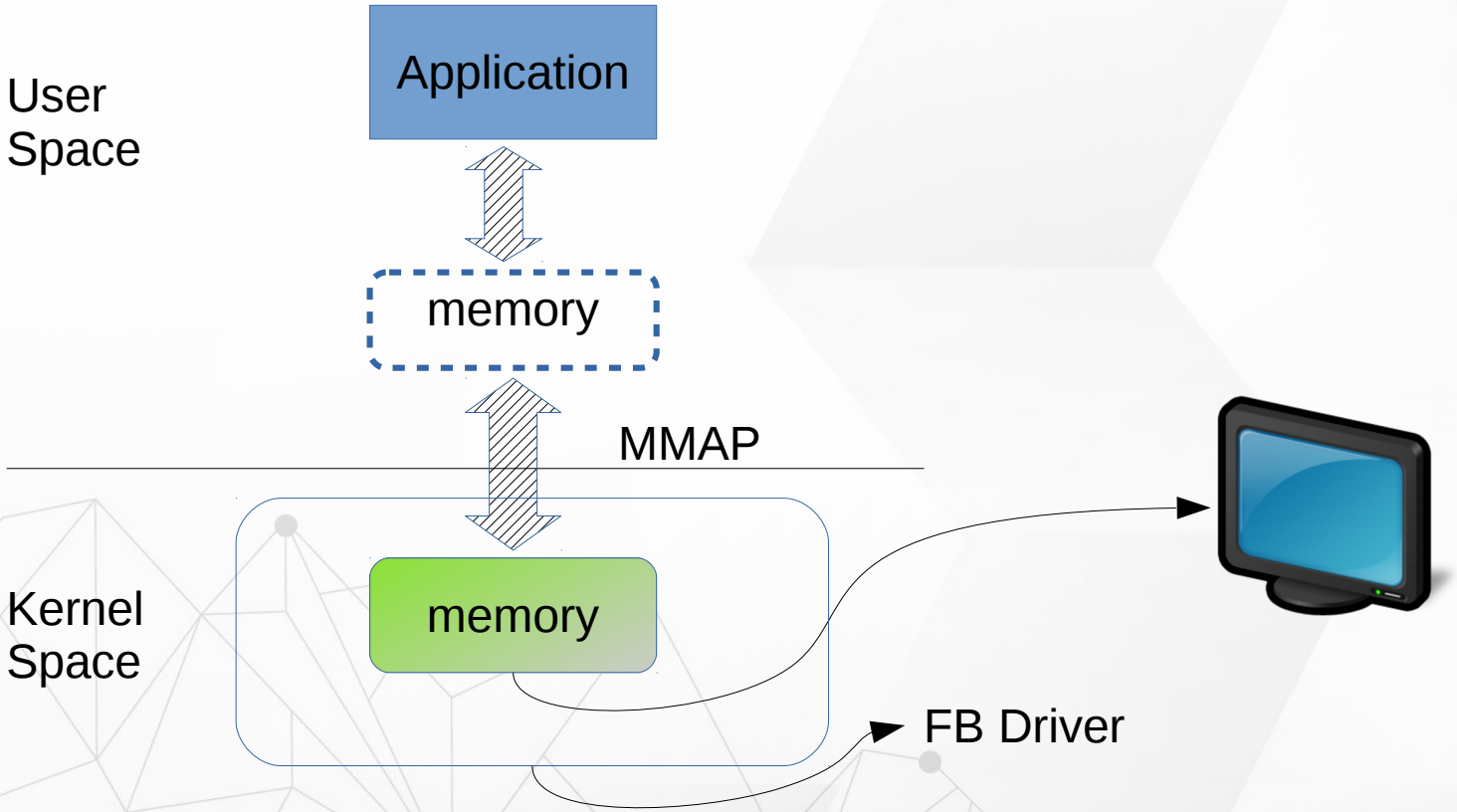
DRM

Use DRM



FBDEV

- The frame buffer device provides an abstraction for the graphics hardware.



V4L2

- Video For Linux 2
  - Multimedia, Video-In
  - Rotator and Scaler
  - Video codec

KMS

- KMS device model
  - CRTCs
  - Connectors
  - Encoders
  - Planes
- Kernel Mode Setting
  - screen resolution
  - color depth and
  - refresh rate

KMS

- linux-4.20.7/drivers/gpu/drm/
- linux-4.20.7/drivers/gpu/drm/exynos

exynos_drm_crtc.c
exynos_drm_dpi.c
exynos_drm_drv.c
exynos_drm_dsi.c
exynos_drm_fb.c
exynos_drm_fbdev.c
exynos_drm_fimc.c
exynos_drm_fimd.c

exynos_drm_g2d.c
exynos_drm_gem.c
exynos_drm_gsc.c
exynos_drm_iommu.c
exynos_drm_ipp.c
exynos_drm_mic.c
exynos_drm_plane.c
exynos_drm_rotator.c
exynos_drm_scaler.c
exynos_drm_vidi.c
exynos_hdmi.c
exynos_mixer.c

KMS

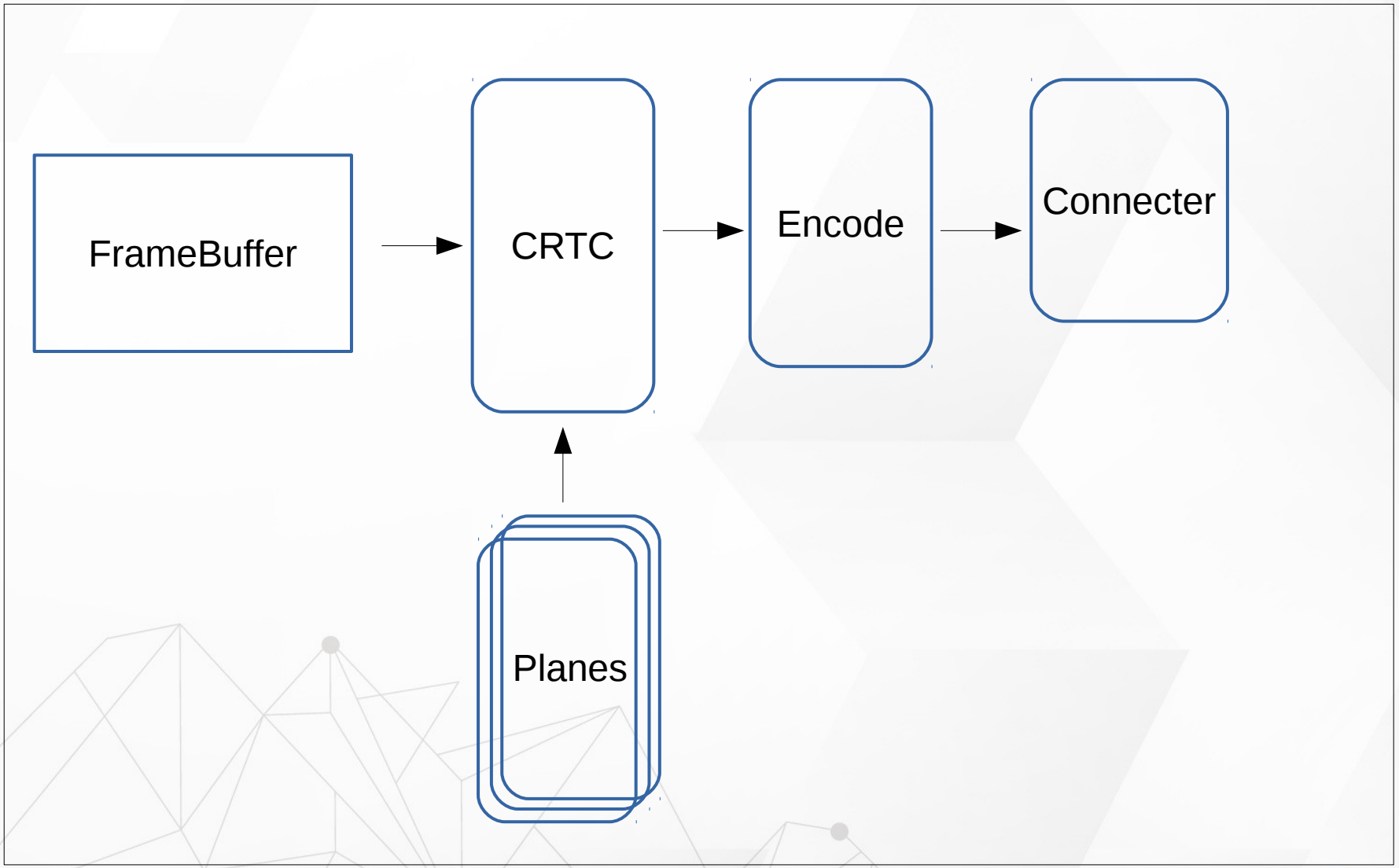
exynos4412-tiny4412.dts

```
&fimd {
        pinctrl-0 = <&lcd_clk &lcd_data24>;
        pinctrl-names = "default";

        samsung,invert-vden;
        samsung,invert-vclk;

        display-timings {
                native-mode = <&timing0>;
                timing0: timing {
                        clock-frequency = <33000000>;
                        hactive = <800>;
                        vactive = <480>;
                        hfront-porch = <90>;
                        hback-porch = <10>;
                        hsync-len = <28>;
                        vback-porch = <29>;
                        vfront-porch = <13>;
                        vsync-len = <3>;
                        vsync-active = <0>;
                        hsync-active = <0>;
                        pixelclk-active = <0>;
```

KMS



LCD Parameters

- VBPD : vertical back porch
- VFBD : vertical front porch
- VSPW : vertical sync pulse width
- HBPD : horizontal back porch
- HFPD : horizontal front porth
- HSPW : horizontal sync pulse width

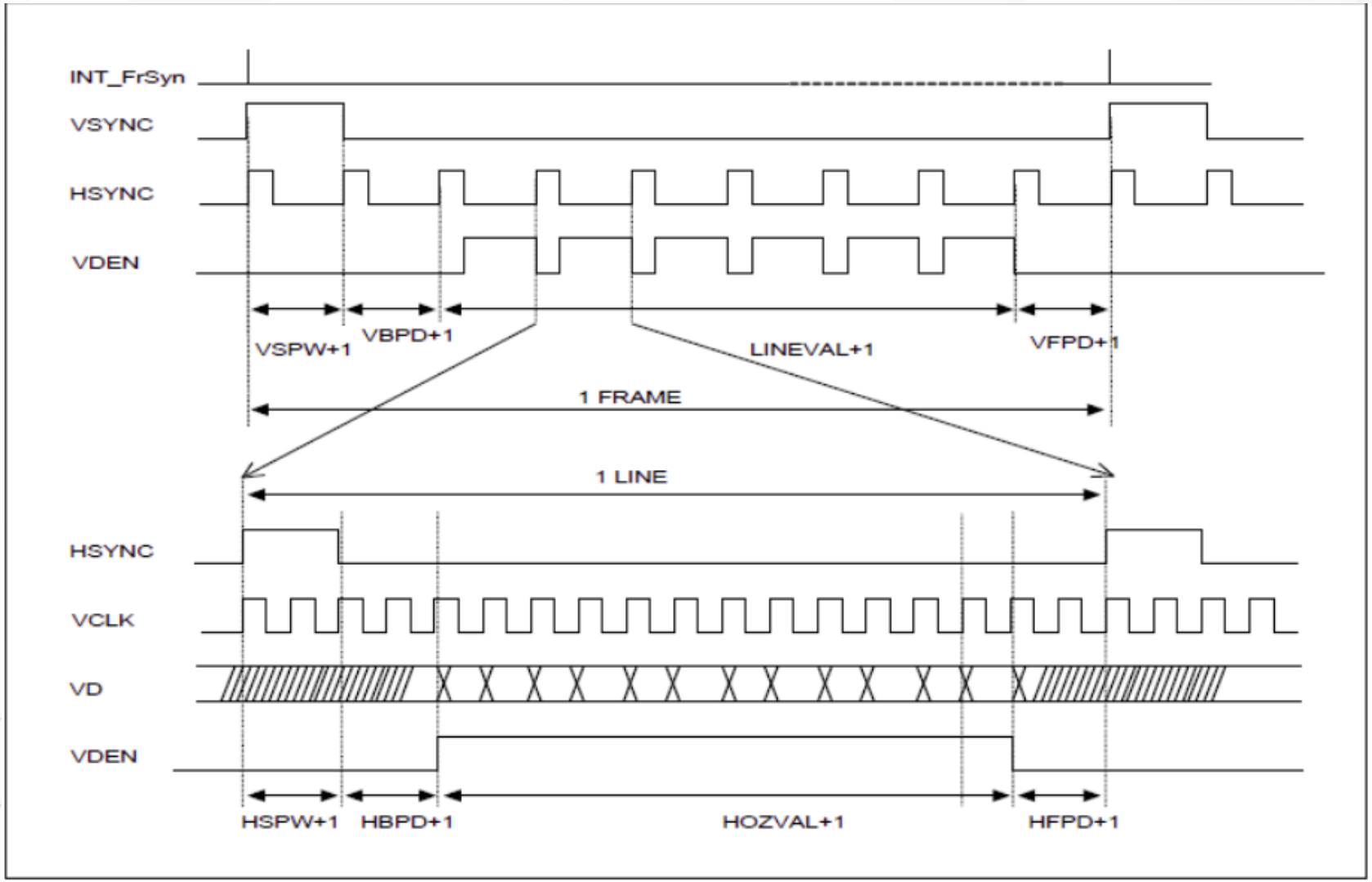
LCD Display Parameters

- Dotclock
  - The video hardware draws pixels on the display speed

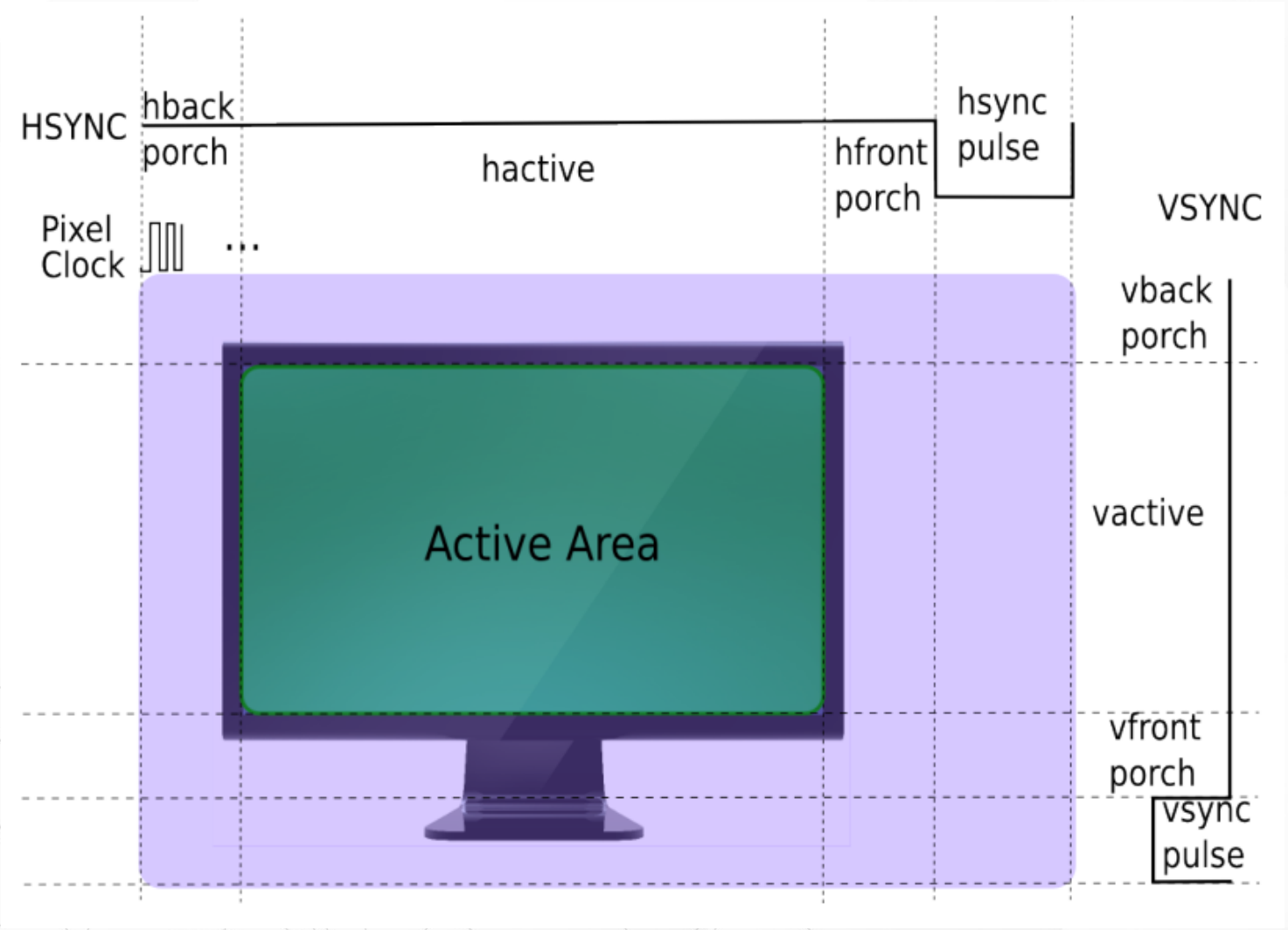
```
dotclock = (X-resolution + left margin + right
           Margin + HSYNC length) *
           (Y-resolution + upper margin + lower
            margin + VSYNC length) * refresh rate


pixclock = 1/dotclock
```

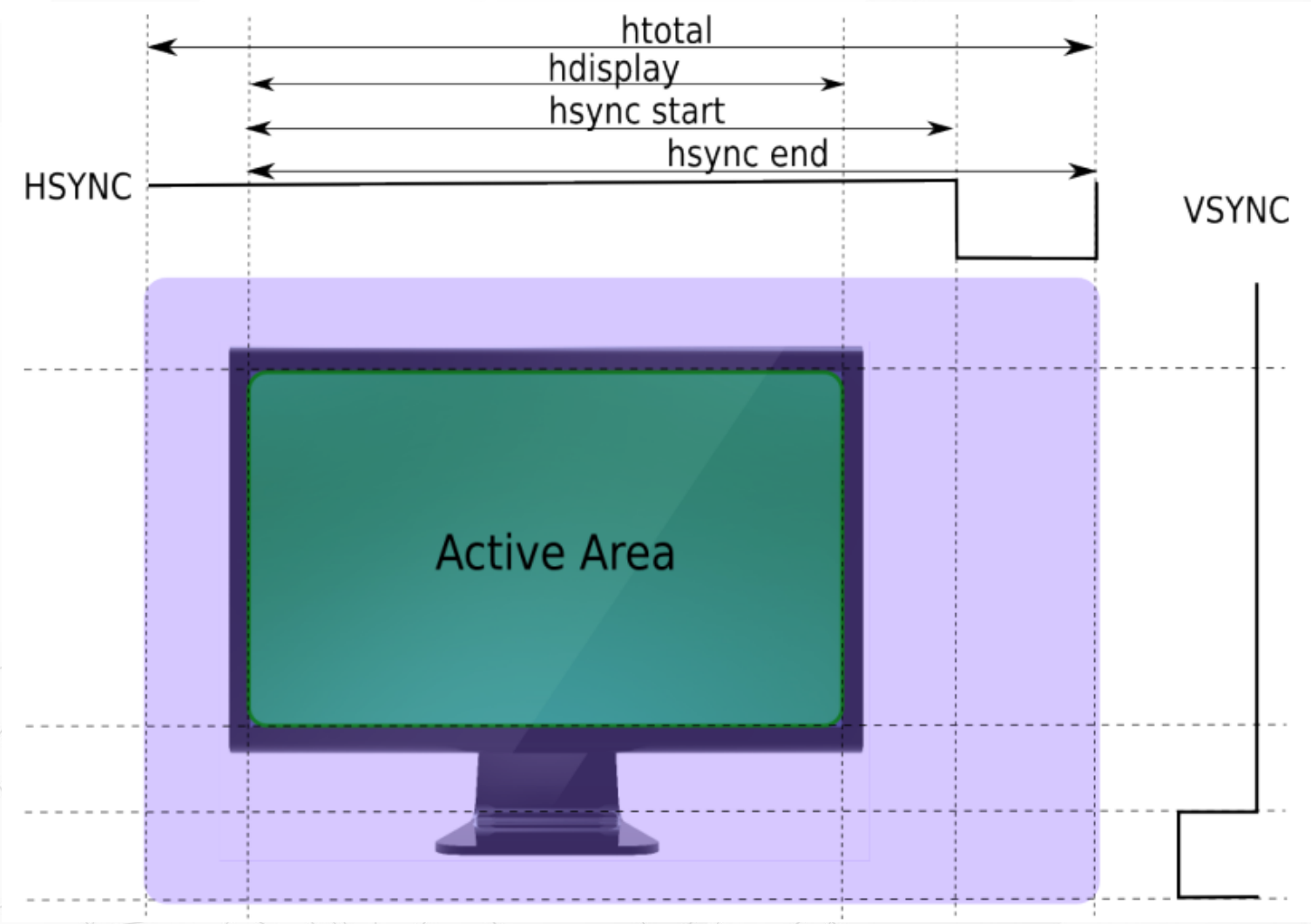
LCD Display Parameters



LCD Display Parameters



LCD Display Parameters



Tiny4412 LCD

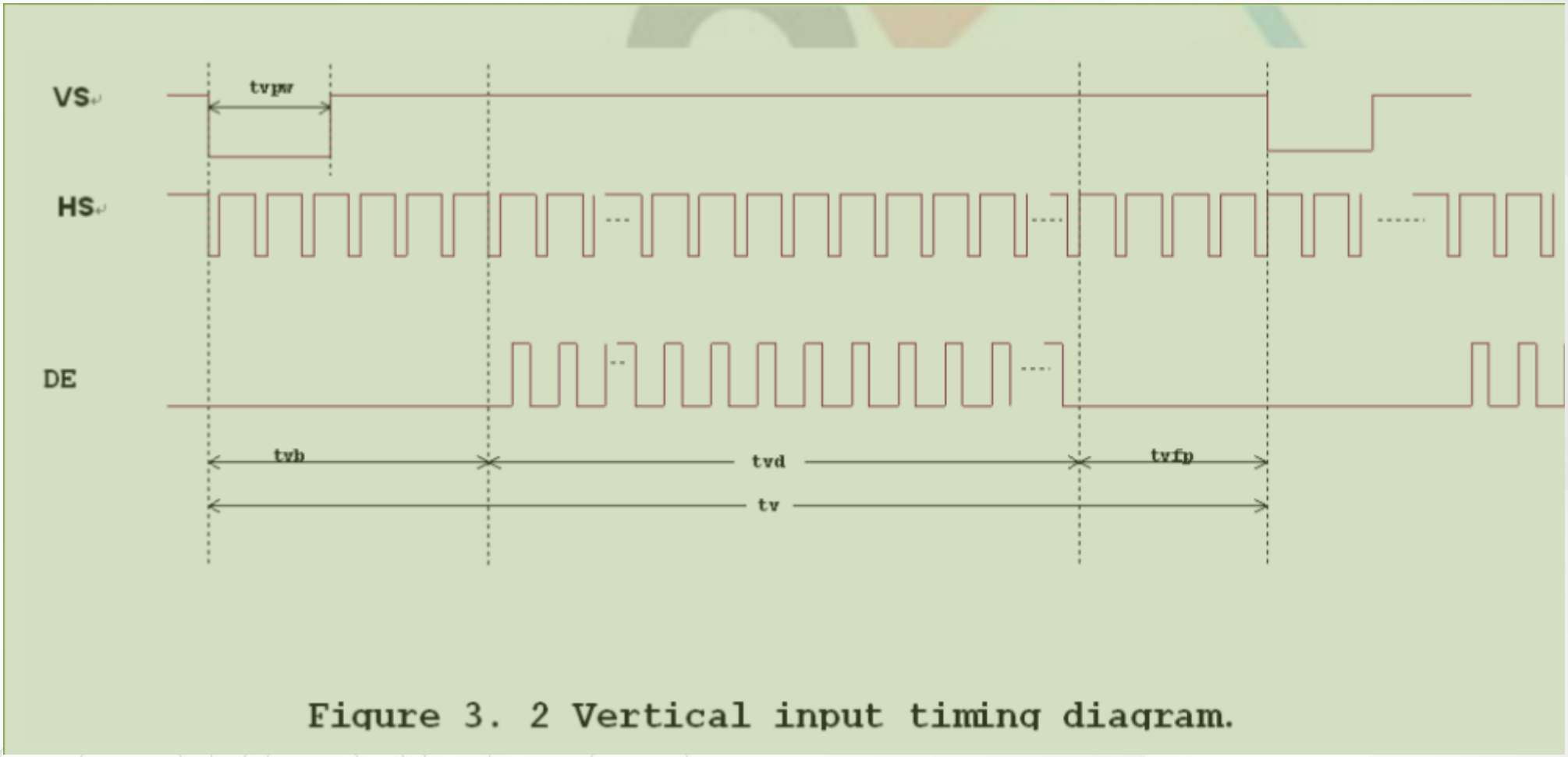


Figure 3. 2 Vertical input timing diagram.

Tiny4412 LCD

Item	Symbol	Values			Unit	Remark
		Min.	Typ.	Max.		
Horizontal Display Area	thd	-	800	-	DCLK	
DCLK Frequency	fclk	26.4	33.3	46.8	MHz	
One Horizontal Line	th	862	1056	1200	DCLK	
HS pulse width	thpw	1	-	40	DCLK	
HS Blanking	thb	46	46	46	DCLK	
HS Front Porch	thfp	16	210	354	DCLK	

Tiny4412 LCD

Item	Symbol	Values			Unit	Remark
		Min.	Typ.	Max.		
Vertical Display Area	tvd	-	480	-	TH	
VS period time	tv	510	525	650	TH	
VS pulse width	tvpw	1	-	20	TH	
VS Blanking	tvb	23	23	23	TH	
VS Front Porch	tvfp	7	22	147	TH	

Tiny4412 LCD

Item	Symbol	Values			Unit	Remark
		Min.	Typ.	Max.		
Vertical Display Area	tvd	-	480	-	TH	
VS period time	tv	510	525	650	TH	
VS pulse width	tvpw	1	-	20	TH	
VS Blanking	tvb	23	23	23	TH	
VS Front Porch	tvfp	7	22	147	TH	

```
static int exynos_dpi_parse_dt(struct exynos_dpi *ctx)
{
                [...]
        np = of_get_child_by_name(dn, "display-timings");
        if (np) {
                struct videomode *vm;
                int ret;

                of_node_put(np);

                        […]

                ret = of_get_videomode(dn, vm, 0);
                if (ret < 0) {
                        devm_kfree(dev, vm);
                        return ret;
                }

        [...]

        return 0;
}
```