Driver for EP93xx LCD controller

The EP93xx LCD controller can drive both standard desktop monitors and embedded LCD displays. If you have a standard desktop monitor then you can use the standard Linux video mode database. In your board file:

```
static struct ep93xxfb_mach_info some_board_fb_info = {
        .num_modes      = EP93XXFB_USE_MODEDB,
        .bpp            = 16,
};
```

If you have an embedded LCD display then you need to define a video mode for it as follows:

```
static struct fb_videomode some_board_video_modes[] = {
        {
                .name           = "some_lcd_name",
                /* Pixel clock, porches, etc */
        },
};
```

Note that the pixel clock value is in pico-seconds. You can use the KHZ2PICOS macro to convert the pixel clock value. Most other values are in pixel clocks. See Documentation/fb/framebuffer.rst for further details.

The ep93xxfb_mach_info structure for your board should look like the following:

```
static struct ep93xxfb_mach_info some_board_fb_info = {
        .num_modes      = ARRAY_SIZE(some_board_video_modes),
        .modes          = some_board_video_modes,
        .default_mode   = &some_board_video_modes[0],
        .bpp            = 16,
};
```

The framebuffer device can be registered by adding the following to your board initialisation function:

```
ep93xx_register_fb(&some_board_fb_info);
```

Video Attribute Flags

The ep93xxfb_mach_info structure has a flags field which can be used to configure the controller. The video attributes flags are fully documented in section 7 of the EP93xx users' guide. The following flags are available:

EP93XXFB_PCLK_FALLING	Clock data on the falling edge of the pixel clock. The default is to clock data on the rising edge.
EP93XXFB_SYNC_BLANK_HIGH	Blank signal is active high. By default the blank signal is active low.
EP93XXFB_SYNC_HORIZ_HIGH	Horizontal sync is active high. By default the horizontal sync is active low.
EP93XXFB_SYNC_VERT_HIGH	Vertical sync is active high. By default the vertical sync is active high.

The physical address of the framebuffer can be controlled using the following flags:

EP93XXFB_USE_SDCSN0	Use SDCSn[0] for the framebuffer. This is the default setting.
EP93XXFB_USE_SDCSN1	Use SDCSn[1] for the framebuffer.
EP93XXFB_USE_SDCSN2	Use SDCSn[2] for the framebuffer.
EP93XXFB_USE_SDCSN3	Use SDCSn[3] for the framebuffer.

Platform callbacks

The EP93xx framebuffer driver supports three optional platform callbacks: setup, teardown and blank. The setup and teardown functions are called when the framebuffer driver is installed and removed respectively. The blank function is called whenever the display is blanked or unblanked.

The setup and teardown devices pass the platform_device structure as an argument. The fb_info and ep93xxfb_mach_info structures can be obtained as follows:

```
static int some_board_fb_setup(struct platform_device *pdev)
{
        struct ep93xxfb_mach_info *mach_info = pdev->dev.platform_data;
        struct fb_info *fb_info = platform_get_drvdata(pdev);

        /* Board specific framebuffer setup */
}
```

Setting the video mode

The video mode is set using the following syntax:

```
video=XRESxYRES[-BPP][@REFRESH]
```

If the EP93xx video driver is built-in then the video mode is set on the Linux kernel command line, for example:

```
video=ep93xx-fb:800x600-16@60
```

If the EP93xx video driver is built as a module then the video mode is set when the module is installed:

```
modprobe ep93xx-fb video=320x240
```

Screenpage bug

At least on the EP9315 there is a silicon bug which causes bit 27 of the VIDSCRNPAGE (framebuffer physical offset) to be tied low. There is an unofficial errata for this bug at:

```
https://marc.info/?l=linux-arm-kernel&m=110061245502000&w=2
```

By default the EP93xx framebuffer driver checks if the allocated physical address has bit 27 set. If it does, then the memory is freed and an error is returned. The check can be disabled by adding the following option when loading the driver:

```
ep93xx-fb.check_screenpage_bug=0
```

In some cases it may be possible to reconfigure your SDRAM layout to avoid this bug. See section 13 of the EP93xx users' guide for details.