MVF DCU Driver

Release notes

The DCU driver supports the following features:

- Display configuration
- Configuring DCU layer features
  - Position
  - Offset
  - Alpha Blending
  - Updating buffer address
  - Enable/Disable
  - Pixel formats: RGB565, RGB888, ARGB8888
- Interrupt support
  - Enable/Disable
  - Register/Free interrupt handlers for DCU interrupts
- DCU configuration
  - Enable/Disable with inbuilt blanking/unblank functionality
  - Enable/Disable colorbar mode via sys class interface
  - Set background colors
- DCU driver registration as char device and provides sys class interface

<Add details on how to enable DCU driver support in kernel config>

Limitations

The DCU driver has the following limitations::

- Currently only supports single DCU
- Backlighting of display is limited to enable/disable which is directly controlled by DCU via GPIO instead of PWM timer
- DCU driver suspend/resume and power management functions are currently not supported

- DCU features not supported: color keying, CLUT, gamma correction, error handling, tile and cursor modes
- Pixel formats limited to the ones mentioned in the release notes
- Does not have any interface with TCON driver. Currently DCU directly sets the TCON bypass bit of TCON module
- DCU char driver does not permit any file operations

Prerequisites

The DCU driver has been tested on:
Hardware:
-TWR-VF600
-TWR-ELEV
  Connect A41 to A43, and A42 to A44 on the primary (white connectors) elevator.
-TWR-SER2
  10/100 Dual RMII (open J8 and J9, SW1 11000000, SW2 10100000)
  Serial-to-USB (short 1 and 2 of J7, short 3 and 4 of J7)
  RS232 DB9 (short 2 and 3 od J1, short 2 and 3 of J2, short 1 and 2 of J13)
-TWR-RGB
Software:
Tag: 1.0
Commit: 57bf21dda4582d0ca5b4ae0c0593df2f43bfeab7
Branch: dcu_development

DCU Driver Unit test

Introduction

Release 1 of the DCU driver has unit testing option integrated into the driver and compiled in by default. To disable unit test code from getting compiled into the driver comment out the define “DCU_UNIT_TEST” in /drivers/mvf/dcu4/dcu4_driver.c

Running the test

To run the DCU driver unit test, load the kernel on the TWR-VF600 board and on the busy box prompt type the below command:
cd sys/class/mvf_dcu/mvf_dcu

Unit test

echo "1" > dcu_unit_test
The following messages will appear on the console and the corresponding color patterns are displayed on the screen (Note: There is a 5 second pause between each update) if the test passes successfully:
Colorbar mode enabled
Colorbar mode disabled
Background color set to red
Background color set to green
Layer 0 enabled, 16x16 Black square @top left corner

Layer 1 enabled, 16x16 Black square @offset 16x16 from top left corner
Layer 1 position now @offset 32x32 from top left corner
Layer 1 image updated to white color
Layer 1 alpha set to 0x1F
DCU disabled
DCU enabled
Layer 1 disabled and uninit
Layer 0 disabled and uninit
DCU disabled

Colorbar mode

To enable colorbar mode, type below command in the busybox prompt:
echo "1" > dcu_colorbar_mode
Verify the LCD displays a pattern of 8 bars of different colors

To disable colorbar mode, type below command in the busybox prompt:
echo "1" > dcu_colorbar_mode
Verify the LCD goes blank with the backlight disabled

To get the state of colorbar mode, type below command in the busybox prompt:
cat dcu_colorbar_mode
Depending if colorbar mode is enabled or not the below message is displayed on the console:

Colorbar mode enabled
Colorbar mode not enabled

Test coverage

The unit test covers the following API's:

```
void dcu_unit_test_config_display(struct dcu_soc *dcu)
void dcu_enable_colorbar_mode(struct dcu_soc *dcu)
void dcu_disable_colorbar_mode(struct dcu_soc *dcu)
void dcu_set_bgnd_color(struct dcu_soc *dcu, dcu_color_t color)
void dcu_enable(struct dcu_soc *dcu)
void dcu_set_bgnd_color(struct dcu_soc *dcu, dcu_color_t color)
int32_t dcu_init_layer(struct dcu_soc *dcu, uint8_t layer,
                                                    uint32_t pixel_fmt,
                                                    uint16_t width, uint16_t height,
                                                    dma_addr_t phyaddr_0,
                                                    int16_t x_pos, int16_t y_pos)
int32_t dcu_enable_layer(struct dcu_soc *dcu, uint8_t layer)
int32_t dcu_set_layer_position(struct dcu_soc *dcu, uint8_t layer, int16_t x_pos,
                int16_t y_pos)
int32_t dcu_update_layer_buffer(struct dcu_soc *dcu, uint8_t layer,
                                    dma_addr_t phyaddr)
int32_t dcu_config_layer_alpha(struct dcu_soc *dcu, uint8_t layer,
                uint8_t alpha_value, alpha_aa_config aa)
void dcu_disable(struct dcu_soc *dcu)
int32_t dcu_disable_layer(struct dcu_soc *dcu, uint8_t layer, bool wait_for_stop)
void dcu_uninit_layer(struct dcu_soc *dcu, uint8_t layer)
void dcu_disable(struct dcu_soc *dcu)
```