

Early Splash Screen on DRA7x Devices

Venkateswara Rao Mandela

ABSTRACT

This application report provides information on how to display a splash screen as soon as possible on the DRA7xx devices and perform a glitch-free transition from the splash screen to the full system UI. The steps in this document are demonstrated using U-Boot and Linux kernel. The same steps can be applied to other bootloaders and operating systems.

Contents

1	Introduction	2
2	Organization of the Document	2
3	Hardware Used	2
4	High-Level Approach	4
5	Testing the Splash Screen	5
6	Measurements	7
7	Customizing the Implementation.....	7
8	Summary.....	7
9	References	8

List of Figures

1	Hardware Setup for Splash Screen	3
2	Display Subsystem Overview	4
3	In Bootloader	4
4	During Kernel Boot and UI Initialization.....	5
5	After UI Initialization	5
6	Splash Screen Disabled	5

List of Tables

1	List of U-Boot Patches	5
2	List of Kernel Patches.....	6
3	Boot Time Measurement for Splash Screen	7

1 Introduction

Displaying a splash screen on boot can significantly improve the user perception of the system boot time. This application report describes how a splash screen can be displayed on an LCD in less than 500 ms on a DRA7xx device. The methods described in this document can be used on AM57x devices as well.

2 Organization of the Document

In this document, the hardware required to replicate the results and the high level implementation details are discussed first. Then, a description is provided on how the results can be replicated independently and the steps to customize the software as required.

3 Hardware Used

For this document, the Rev G DRA75x EVM with 10.1" 1280x800 LCD is used. The LCD is connected to VOUT1 of the DRA75x device. VOUT1 is controlled by the Display Subsystem (DSS) of the DRA75x device (see [Figure 1](#)).



Figure 1. Hardware Setup for Splash Screen

DSS supports multiple display outputs and multiple display planes. Two or more DSS display planes can be overlaid in a specified order and sent to one DSS display output. Figure 2 shows an overview of the Display Subsystem.

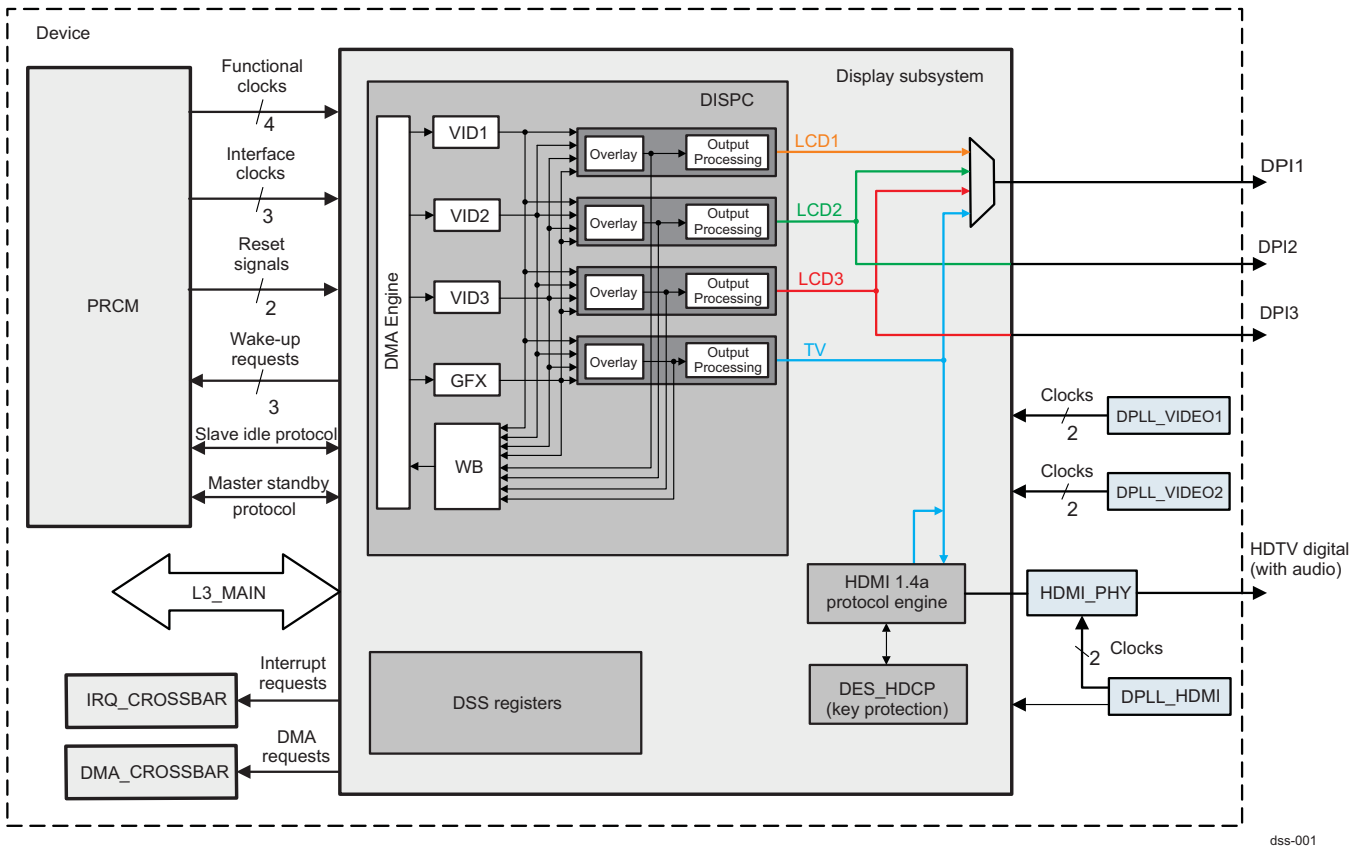


Figure 2. Display Subsystem Overview

4 High-Level Approach

On boot, one display plane is used in the DSS to display the splash screen. This display plane is given the highest priority so that it always stays on top. This plane is set in self-refresh mode so that software interaction is not required to maintain the splash screen. Figure 3 shows how the planes are ordered in DSS and the image rendered on the LCD. The left hand side shows two display planes in DSS with the highest priority plane on top. The right hand side shows the image rendered on the screen. In this case, the display plane with the splash screen has the highest priority and is rendered on the screen.

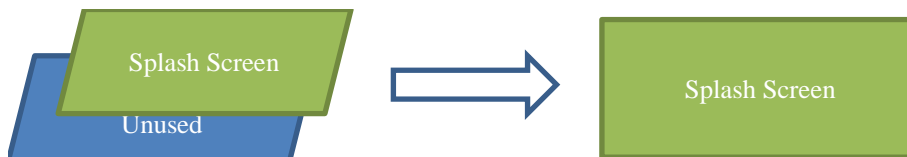


Figure 3. In Bootloader

Booting the Linux kernel is continued in the background. Normally, Linux kernel resets and reinitializes each peripheral it controls. The Linux kernel was modified to skip reinitializing the DSS and avoid using the display plane that is used for the splash screen. This functionality to skip initializing the DSS is controlled through the device tree.

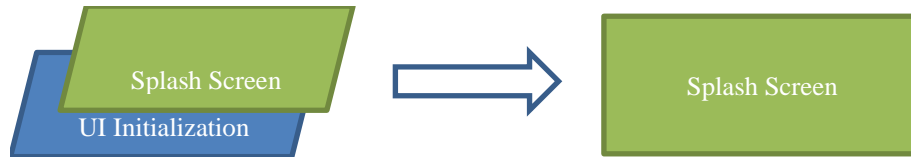


Figure 4. During Kernel Boot and UI Initialization

The kernel boots and launches userspace. Userspace initializes the UI using the remaining display planes. All this happens while the splash screen remains on top and completely hides any delay in initializing the UI.

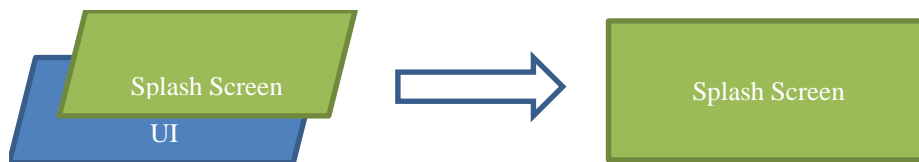


Figure 5. After UI Initialization

Once the UI is fully initialized, the UI is switched to the front by simply disabling the display plane showing the splash screen (see Figure 6). This results in a glitch-free transition from the splash screen to the UI.

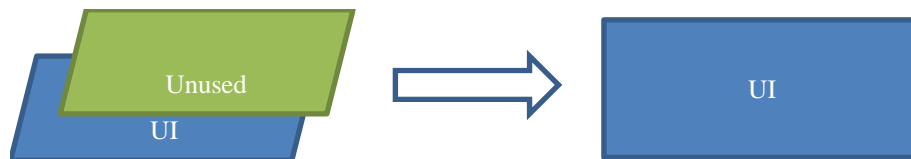


Figure 6. Splash Screen Disabled

Once the UI is displayed, the displayed plane used for the splash screen can be reclaimed and used for other purposes.

5 Testing the Splash Screen

The splash screen functionality is implemented on top of the GLSDK 7.04 release. The same implementation can be used with the 6AL1.1 release of the Android SDK. For the below steps, it is assumed that a working GLSDK 7.04 installation is present and you are able to build U-Boot and Linux Kernel. For more information, see .

1. Download the u-boot patches shown Table 1 and apply them on top of the GLSDK 7.04 u-boot repository. The base commit id on which these patches were tested is 5ac4f5ccd2ca66e5b787d34b9aa06227040c3c03.

Table 1. List of U-Boot Patches

S. No	Headline	URL
1	spl: Enable display of splash screen	http://review.omapzoom.org/37778
2	spl: dra7xx: enable cache for ocmc 1 and 2	http://review.omapzoom.org/37779
3	Enable compression	http://review.omapzoom.org/37780
4	Loading logo from QSPI	http://review.omapzoom.org/37781
5	spl: dra7xx: enable loading logo before kernel	http://review.omapzoom.org/37782

- Download the kernel patches shown in [Table 2](#) and apply them on top of the GLSDK 7.04 kernel repository. The base commit id on which these patches were tested is 8615cf6f067a6a6f5b89255c1d57b72748c86212.

Table 2. List of Kernel Patches

S. No	Headline	URL
1	OMAPDSS: provide a way to set number of overlays	http://review.omapzoom.org/36935
2	OMAPDSS: Add methods for skipping display initialization	http://review.omapzoom.org/36936
3	OMAPDSS: Adapt DSS driver for display sharing use-case	http://review.omapzoom.org/36937
4	OMAPDSS: tlc59108: adapt panel driver for display sharing use case	http://review.omapzoom.org/36938
5	OMAPDSS: Adapt DSS driver for skipping initialization during bootup	http://review.omapzoom.org/36939
6	arm: dra7: hwmod: remove opt clks enable flag from hwmod	http://review.omapzoom.org/36946
7	ARM: dts: omap: Add enable optional clks during reset binding for OMAP	http://review.omapzoom.org/36944
8	ARM: dts: DRA7: enabling optional clks in DT	http://review.omapzoom.org/36947
9	arm: OMAP2+: hwmod: Extract enable-opt-clks-on-reset info from DT	http://review.omapzoom.org/36940
10	ARM: DTSi and DRA7 EVM DTS: changes to define few nodes as variable nodes	http://review.omapzoom.org/37773
11	DRA7: HWMOD: WORKAROUND to Exclude unused clock gating and parent hwmod controls for DSS	http://review.omapzoom.org/37774
12	DSS: Changes to share Display subsystem with remotecores on DRA7	http://review.omapzoom.org/37775
13	DSS: Do not disable PLL if DSS is initialized	http://review.omapzoom.org/37776
14	arch: arm: dra7: add dts for splash screen functionality	http://review.omapzoom.org/37777

- Rebuild the kernel and u-boot.
- Copy MLO and u-boot.img into the FAT partition of the SD card.
- Copy zImage into the “/boot” folder of the ext4 partition on the SD card.
- Copy “dra7-evm-lcd10-splash.dtb” into the “/boot” folder of the ext4 partition on the SD card.
- Modify “uenv.txt” on the SD card to used “dra7-evm-lcd10-splash.dtb” as the device tree file.
- Clone the Git repository containing the logo images and copy “drive_innovation_1280x800.yuv.lzo” into the FAT partition of the SD card as “logo.lzo”.

```
host $ git clone git://git.ti.com/glsdk/logos.git
```
- Insert the SD card into the EVM; ensure that the EVM is configured for SD boot and reboot. A splash screen should come up as soon as the board is powered on and stay on screen while the full system boots in the back ground.
- Start Weston by running “pvrsrvinit” and “weston -tty=1 -idle-time=0 &”. The splash screen should still remain on top.

```
target # pvrsrvinit
target # weston --tty=1 --idle-time=0 &
```
- Launch a Weston application. The splash screen continues to stay on the top.

```
target # weston-simple-egl &
```
- Use omapconf to disable the display plane (VID3) showing the splash screen and bring Weston to the foreground.

```
target # omapconf write 0x58001370 0x0
```

6 Measurements

The measurements for the splash screen shown below are obtained using the 32 K timer present in the DRA7xx SOC's. The procedure for boot time measurements is described in [4]. Two measurements for the splash screen display are shown below.

- Time taken from SOC PORZ to displaying the splash screen on the LCD
- Time spent in the functions related to splash screen display. The measurements are performed for:
 - SD boot - as this is easier to try out
 - QSPI boot - as it provides the fastest possible time to splash screen

Table 3. Boot Time Measurement for Splash Screen

Boot Media	Time to Splash Screen From PORZ (ms)	Time Spent in Showing the Splash Screen (ms)
SD card	486	40
QSPI4	141	32

The measurements above were taken in SD production boot mode and QSPI production boot mode, respectively. The SYSBOOT SW2[5:0] settings used are 11 0000 and 11 0111, respectively.

7 Customizing the Implementation

1. To obtain the splash screen even faster, boot from QSPI. Flash the logo file into QSPI at offset "0x9e0000". For information on flashing to QSPI, see the .
2. The logo format used is NV12, for example, Y+UV (two planes, 4:2:0 format). To use another format, appropriately change the format field in the VID3 pipeline attributes. It might also be necessary to update the source address of the frames. The required source code can be found in the *setup_display()* function in "board\ti\dra7xx\display.c" in u-boot.
3. This code currently supports splash screen on:
 - (a) 7" LCD 800x480 supplied with the earlier versions of J6 EVM
 - (b) 10" LCD 1280x800 supplied with Rev G J6 EVM
 To support other LCD's, modify the source files "board\ti\dra7xx\display.c" and "board\ti\dra7xx\display.h". PLL settings for Video PLL are performed in "board\ti\dra7xx\video_pll.c".
4. The logo file is loaded into OCMC1/2 and decompressed into OCMC1/2. This was done to remove any dependency on DDR and speed up the decompression time. This will not function on variants of DRA7xx that do not have OCMC1/2 memory. To customize the locations where the logo is stored and decompressed, modify the below macros in "include\configs\dra7xx_evm.h".
 - (a) DRA7_LOGO_MAX_COMP_SIZE – Maximum size of the compressed logo
 - (b) DRA7_LOGO_START – Indicates where the logo is stored after decompression
 - (c) DRA7_LOGO_COMP_START – Indicates the location where the compressed logo is loaded from QSPI/SD/eMMC.
 - (d) DRA7_LOGO_MAX_SIZE – Maximum size of the uncompressed logo.
 When using DDR for the logos, ensure that the corresponding region is marked as reserved in the devicetree using "memreserve".
5. The compression format used for the logo is "lzo". "lzo" format was selected due to the fast decompression time on the target.

8 Summary

This application report illustrates how a splash screen can be displayed on the DRA7xx platform and how to perform a glitch-free transition from the splash screen to UI. It has also been demonstrated through measurements how quickly this can be done. Information was also provided on how the splash screen can be customized further.

9 References

1. *DRA75x, DRA74x SoC for Automotive Infotainment Silicon Revision 2.0, 1.x Technical Reference Manual* (SPRUH12)
2. [DRA7xx GLSDK Software Developers Guide](#), wiki
3. [DRA7xx GLSDK Software Developers Guide Using QSPI Boot](#), wiki
4. [DRA7xx GLSDK Software Developers Guide, Build Instructions for Boot Time Measurement](#) wiki

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ("TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products (<http://www.ti.com/sc/docs/stdterms.htm>), [evaluation modules](#), and [samples](http://www.ti.com/sc/docs/sampterm.htm) (<http://www.ti.com/sc/docs/sampterm.htm>).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2017, Texas Instruments Incorporated