Codec Memory Management(CMM) User's Manual (Linux)

S3C6400/6410

August 26, 2008

REV 1.11

Important Notice

The information in this publication has been carefully checked and is believed to be entirely accurate at the time of publication. Samsung assumes no responsibility, however, for possible errors or omissions, or for any consequences resulting from the use of the information contained herein.

Samsung reserves the right to make changes in its products or product specifications with the intent to improve function or design at any time and without notice and is not required to update this documentation to reflect such changes.

This publication does not convey to a purchaser of semiconductor devices described herein any license under the patent rights of Samsung or others.

Samsung makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Samsung assume any liability arising out of the application or use of any product or circuit and specifically disclaims any and all liability, including without limitation any consequential or incidental damages.

"Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by the customer's technical experts.

Samsung products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, for other applications intended to support or sustain life, or for any other application in which the failure of the Samsung product could create a situation where personal injury or death may occur.

Should the Buyer purchase or use a Samsung product for any such unintended or unauthorized application, the Buyer shall indemnify and hold Samsung and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, expenses, and reasonable attorney fees arising out of, either directly or indirectly, any claim of personal injury or death that may be associated with such unintended or unauthorized use, even if such claim alleges that Samsung was negligent regarding the design or manufacture of said product

S3C6400/6410 RISC Microprocessor
CMM User's manual

Copyright © 2008 Samsung Electronics Co.,Ltd.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electric or mechanical, by photocopying, recording, or otherwise, without the prior written consent of Samsung Electronics Co.,Ltd.

Samsung Electronics Co., Ltd.
San #24 Nongseo-Dong, Giheung-Gu
Yongin-City Gyeonggi-Do, Korea
446-711

Home Page: http://www.samsungsemi.com/

E-Mail: mobilesol.cs@samsung.com

Printed in the Republic of Korea

Revision History

Revision No	Description of Change	Refer to	Author(s)	Date
1.00	Initial Version	-	Jiun Yu	2008-07-05
1.10	Node name is changed		Jiun Yu	2008-07-19
1.11	New ioctl's command is added		Jiun Yu	2008-08-26

Contents

1 Introduction

1.1 Purpose

The purpose of the document is to describe the CMM API for easy portability into different platforms by developers.

1.2 Scope

The scope of this document is to describe

- Software architecture of CMM
- Usage of CMM API
- How to use CMM driver

1.3 Intended Audience

Intended Audience	Tick whenever Applicable
Project Manager	Yes
Project Leader	Yes
Project Team Member	Yes
Test Engineer	Yes

1.4 Definitions, Acronyms, and Abbreviations

Abbreviations	Description
CMM	Codec Memory Management
API	Application Program Interface

1.5 References

Number	Reference	Description
1	SMDK6400_WinCE6.0_FMD_PortingGuide.doc	OS porting guide
2	SMDK6400_WinCE6.0_VideoDriver_UserManual.doc	Video Driver specification
3	S3C6400_6410_Linux2.6.21_CMM_UserManual_REV1.00_20080705.doc	CMM v1.00 user's manual

2 Software Architecture

2.1 Overview

When multimedia player use s/w decoder, Performance problem is often issued. CMM(Codec Memory Management) driver helps to improve rendering performance.

In common multimedia player, Decoded YUV data is transferred to video memory using memcpy(). It decreases much performance when the resolution of movie is large.

CMM Driver provides the interface to transfer decoded YUV data to video memory directly. At first, It allocates virtual address to the player. The virtual address is surely cacheable area. So, s/w decoder can utilize cache. After decoding, the player request for CMM to flush cached area. And then, the player request physical address of YUV buffer to CMM. With the physical address, the player calls video driver API for rendering. YUV data is transferred to h/w post processor by DMA.

It does not only reduce memcpy() time, but also make player to decode and render at the same time. Because rendering is done by only h/w, decoding performance is not decreased. You should make decoding and rendering as multi-threaded.

There are 2 methods to render YUV data. The one is using local path between h/w post processor and LCD. It doesn't posses data BUS. But local path only supports RGB888. The other is using DMA between post processor and LCD.

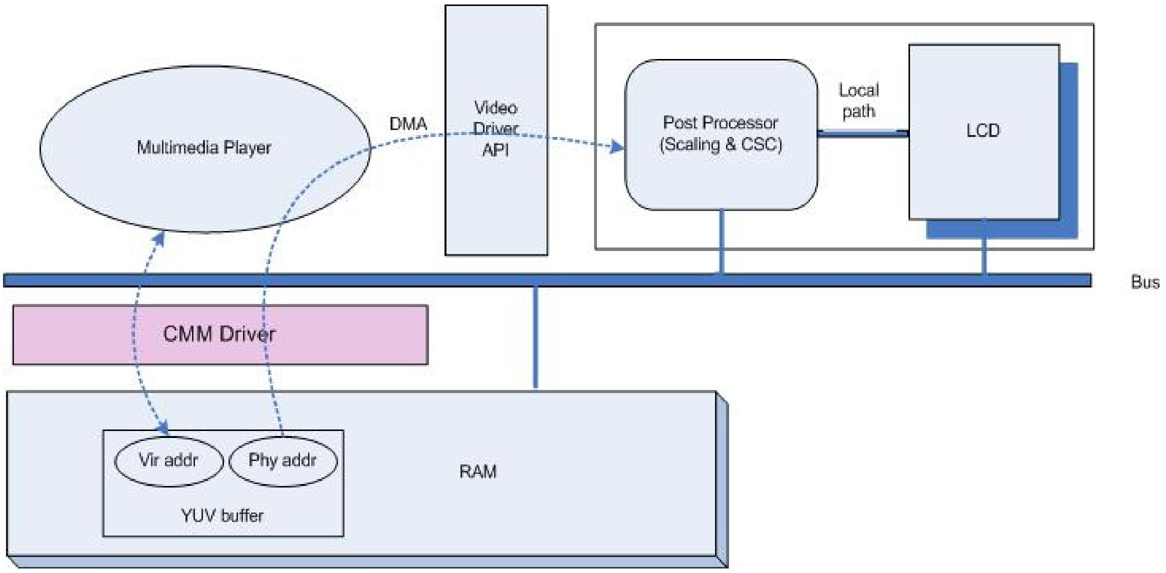


Figure . Architecture of CMM API

3 Package Guidelines

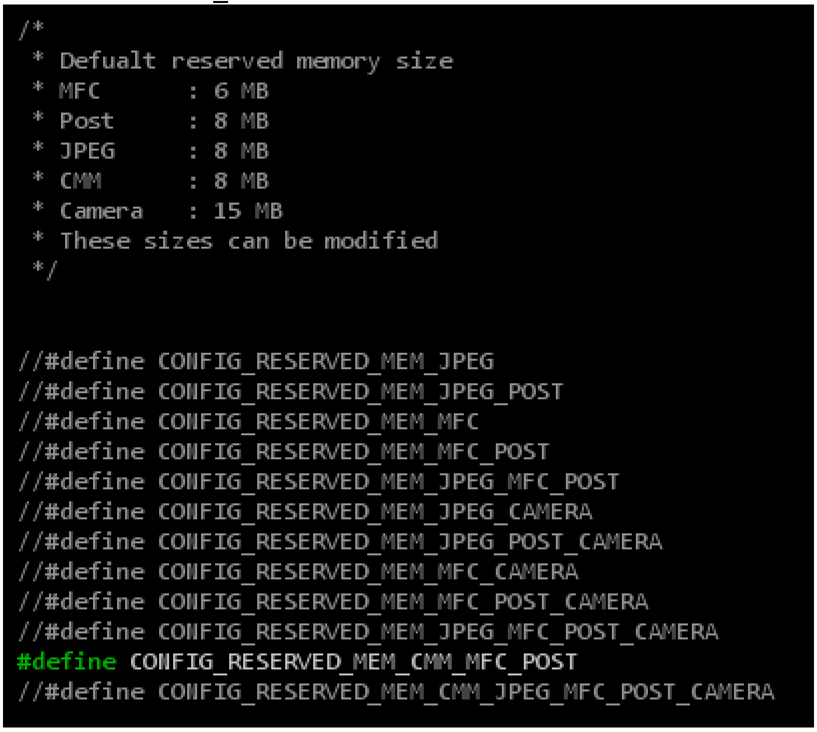
3.1 Directory Structure

Directory	Files	Description
/cmm_app/	*.c, *.h	CMM test file
/cmm_drv/	*.c, *.h	CMM Device Driver file
/doc	*.doc, *.pdf	CMM documents

4 How to Test CMM

4.1 Kernel Build

Before kernel compilation, you must setup memory layout below figure in “include/asm-arm/arch-s3c2410/reserved_mem.h” file



```
/*
 * Defualt reserved memory size
 * MFC       : 6 MB
 * Post      : 8 MB
 * JPEG      : 8 MB
 * CMM       : 8 MB
 * Camera    : 15 MB
 * These sizes can be modified
 */


//#define CONFIG_RESERVED_MEM_JPEG
//#define CONFIG_RESERVED_MEM_JPEG_POST
//#define CONFIG_RESERVED_MEM_MFC
//#define CONFIG_RESERVED_MEM_MFC_POST
//#define CONFIG_RESERVED_MEM_JPEG_MFC_POST
//#define CONFIG_RESERVED_MEM_JPEG_CAMERA
//#define CONFIG_RESERVED_MEM_JPEG_POST_CAMERA
//#define CONFIG_RESERVED_MEM_MFC_CAMERA
//#define CONFIG_RESERVED_MEM_MFC_POST_CAMERA
//#define CONFIG_RESERVED_MEM_JPEG_MFC_POST_CAMERA
#define CONFIG_RESERVED_MEM_CMM_MFC_POST
//#define CONFIG_RESERVED_MEM_CMM_JPEG_MFC_POST_CAMERA
```

4.2 CMM driver module compilation

Node name : /dev/misc/s3c-cmm

Major number : 10

Minor number : 250

1. how to make device node

```
[root@localhost CMM]# mknod /dev/misc/s3c-cmm c 10 250
```

2. module compilation

```
[root@localhost cmm_drv]# make
```

4.3 Test application compilation

```
[root@localhost cmm_app] make
```

4.4 Insert module and execute binary in Target side

Below commands are executed in target side

```
[root@Samsung cmm_drv] insmod s3c_cmm.ko
[root@Samsung cmm_drv] cd ../cmm_app/
[root@Samsung cmm_app] ./cmm_test
```