LSP 2.10 DaVinci Linux Drivers

Data Manual

PRODUCT PREVIEW information concerns products in the formative or design phase of development. Characteristic data and other specifications are design goals. Texas Instruments reserves the right to change or discontinue these products without notice.

Literature Number: SPRS566A
July 2009

Contents

List of Figures

List of Tables

1 LSP 2.10 DaVinci™ Linux Drivers

1.1 Features

- **Supported Devices:**
  - **TMS320DM644x**
  - **TMS320DM355**
  - **TMS320DM6467**
  - **TMS320DM365**
- **Developed and Tested on DaVinci™ DM644x EVM, DM355 EVM, DM6467, and DM365 EVM Platforms**
- **Built with Linux Kernel version 2.6.18**
- **Compiled with MontaVista Professional v5.0.0 arm_v5t_le (glibc) and arm_v5t_le_uclibc (uclibc) Toolchains**
- **Supports ARM9 GNU/Linux Application Binary Interface (ABI)**
- **Supports U-boot**
- **Supports these bootmodes:**
  - **NAND boot for DM644x**
  - **NAND boot for DM355**
  - **NAND boot for DM365**
- **Video Processing Back End (VPBE)**
  - **Supported on DM644x, DM355 and DM365**
  - **Supports NTSC, PAL on S-video, Composite, Component interfaces using FBDev and V4L2-based Video Display Drivers**
  - **Supports 480p, 576p (DM644x and DM365 only) resolutions on Component interface using FBDev and V4L2-based Video Display Drivers**
  - **Supports VGA Output Using LogicPD LCD Daughter Card**
  - **Supports HD (720p / 1080i) Video Display using THS8200 Daughter Card (DM355 and DM644x only)**
  - **Supports Graphic On-screen Display (OSD) With Alphablending using FBDev Driver**
  - **Through the Linux FBDev Interface:**
    - **Supports Display Panning**
    - **Supports Programmable display modes RGB565, RGB888, and RGBA on OSD**
    - **Supports Color Lookup Table for 1, 2, 4 and 8 bits per pixel modes**
  - **The following features are supported through Linux V4L2 Interface:**
    - **Supports both the video pipelines**
  - **Supports YUYV4:2:2, UYVY4:2:2, RGB888 (DM644x only) pixel formats on each of the video windows**
    - **Supports Cropping**
- **Video Processing Subsystem Front End (VPFE)**
  - **Supported on DM644x, DM355 and DM365**
  - **Supports Composite, S-Video, and Component interfaces using V4L2-based video capture drivers**
  - **Supports NTSC/PAL on S-Video and Composite interface through TVP 5146**
  - **Supports HD(720p/1080i) on Component interface through TVP7002 (DM365 only)**
  - **Supports YUYV 4:2:2, UYVY 4:2:2 formats, and YUV 420 Semi planar (DM365 only) formats**
  - **Supports 8/10-bit raw capture from Micron Image sensor MT9T031 (DM355 and DM365 only)**
  - **Supports 8/12-bit raw capture from Micron Image sensor MT9P031 (DM365 only)**
  - **Supports Bayer pattern output using CCDC**
  - **Supports Conversion of Bayer Input Into YUYV/UYVY using Previewer driver (DM644x only)**
  - **Supports Input Image Upscaling from 1x to 4x and Downscaling up to 0.25x using Resizer Driver (DM644x only)**
  - **Supports Auto Exposure, Auto White balance (AEW), and Auto Focus (AF) on CCDC Input Data**
  - **Supports Conversion of Bayer Input YUYV/UYVY format (Previewer) as well as upscaling up to 4x and downscaling up to 0.0625x (Resizer) using IPIPE driver (DM355 only)**
  - **Supports Conversion of Bayer input to UYVY or YUV 420 Semi planar format (Previewer) as well as upscaling up to 4x and downscaling up to 0.0625x (Resizer) using previewer-resizer driver (DM365 only)**
  - **Supports chaining of previewer and resizer on the ccdc data output to do On the fly preview and resize (on DM355 and DM365 only) and capturing the processed output to SDRAM.**
  - **Supports face detection**

PRODUCT PREVIEW

- Video Port Interface (VPIF) Display
  - Supported on DM6467
  - Supports NTSC(480i),PAL (576i) , resolutions on S-Video, composite and ED(480p/576p) ,HD(720p/1080i) on component interface using V4L2-based Video Display drivers
  - Supports TV output through ADV-7343 encoder
  - Supports two display software channels. Both the software channels will support Standard Definition (SD) display but only one (video2) channel will support High Definition (HD) display
  - Supports the following VBI services
    - CGMS and Closed Captioning on NTSC display
    - WSS on PAL display
- Video Port Interface(VPIF) Capture
  - Supported on DM6467
  - Supports NTSC(480i),PAL (576i) , resolutions on S-Video, composite and ED(480p/576p) ,HD(720p/1080i) on component interface using V4L2-based Video capture drivers
  - Supports SD capture through TVP5147 encoder HD capture through TVP7002
  - Supports two capture software channels. Both the software channels will support Standard Definition (SD) capture but only one (video0) channel will support High Definition (HD) capture
  - Supports the following VBI services
    - CGMS and Closed Captioning on NTSC display
    - WSS on PAL display
- Video Data Conversion Engine(VDCE)
  - Supported on DM6467
  - Supports the following operations in Precodec-mode
    - Resizing (optional)
    - Chroma conversion from YUV 422 to YUV 420
  - Supports the following operations in Postcodec-mode
    - Resizing (optional)
    - Chroma conversion from YUV 420 and YUV 422
    - Blending (optional)
    - Range-mapping-mode (optional)
  - Supports the following operations in Transcodec-mode
    - Resizing (optional)
    - Chroma conversion (optional)
    - Blending (optional)
    - Range-mapping-mode (optional)
  - Supports Edge-Padding-mode
  - Supports both interlaced and progressive formats
  - Supports processing of 16 channels simultaneously
- Audio
  - Supports Advanced Linux Sound Architecture (ALSA) framework
  - Supports Audio Record and Playback From AIC33 for DM644x, DM355 and DM365; and AIC32 for DM6467
  - Supports McBSP in slave mode and AIC33 Codec in master mode for Davinci DM644x, DM355 and DM365
  - Supports I2S McBSP data format for DM644x, DM355 and DM365
  - Supports McASP in slave mode and AIC32 Codec in master mode for Davinci DM6467
  - Supports I2S McASP data format for DM6467
- Communication (Ethernet, USB)
  - Supports CPMAC (internal EMAC) driver for transmitting and receiving network data (DM644x and DM365 only)
  - Supports DM9000A (external Ethernet controller) driver for transmitting and receiving network data (DM355 only)
  - Supports internal Gigabit Ethernet driver for transmitting and receiving network data (DM6467 only)
  - Supports 10/100 Mbps Ethernet speeds, Half and Full duplex on DM644x, DM355 and DM365
  - Supports 1000 Mbps Ethernet speeds, Half and Full duplex for DM6467
  - Supports Auto-Negotiation on the Ethernet driver
  - Supports USB Host, Gadget and OTG (partial) modes based on Inventra USB driver stack
  - Supports USB 2.10 Full and High speed devices
  - Supports USB Mass storage class: NAND, HDD (DM644x, DM365, and DM6467), MMC/SD (DM644x, DM355 and DM365)
  - Supports USB Hub, HID and ISO classes
  - Supports USB CDC and RNDIS classes under gadget mode of operation
- Storage (ATA, MMC/SD, NAND, NOR)
  - Supports ATA Interface to IDE Hard Disk (DM644x, DM365, and DM6467 only)
  - Supports NAND flash memory with small block access for Davinci DM644x, and big

  - **block access for DM6467, DM355, and DM365**
  - **Supports NOR Flash Memory for Storage with small block access (DM644x only)**
  - **Supports High-speed MMC/SD (4-bit/1-bit) interface to external flash cards: MMC and SD on DM644x, DM355 and DM365**
- **Serial (SPI, UART, I^2C)**
  - **Supports communication through SPI interface on DM6467 and DM355**
  - **Supports I^2C Interface to MSP430 Controller (DM644x, DM355 and DM365 only)**
  - **Supports I^2C Interface to Audio/Video Encoders/Decoders**
  - **Supports Serial Communication Using UART (UART0 interface for console)**
- **Transport Stream Interface (TSIF)**
  - **Supported on DM6467**
  - **Supports Transmission and Reception at either 13.5 MHz or 16.875 MHz in Parallel mode for DM6467**
  - **Supports Transmission and Reception at either 27 MHz, 54 MHz or 81 MHz in Serial mode for DM6467**
- **PCI Slave**
  - **Supports interfacing with the host machine over the PCI interface for DM6467**
- **Miscellaneous**
  - **EDMA**
  - **PWM**
  - **GPIO**
  - **WDT**
  - **Timers**
  - **CIR for DM6467**
  - **VLYNQ**

PRODUCT PREVIEW

1.2 Description

TI DaVinci™ Linux Drivers are incorporated in the MontaVista Linux Support Package (LSP) provided with MontaVista Linux Pro v5.0.0. The LSP consists of optimized multimedia peripheral device drivers for the DM644x, DM355, DM365, and DM6467 Digital Media System-on-Chips (DMSoCs), integrated with the MontaVista Linux 2.6.18 kernel to run on the ARM926 core. The drivers enable rapid software development on the DaVinci platforms and are provided in source form to facilitate portability for production hardware platforms. This document provides an overview and performance data for each of the drivers. Table 1-1 summarizes Linux Driver support in the LSP 2.10 Release 02.100.00.020 for DM644x/DM355/DM365/DM6467 based on MontaVista Linux Pro 5.0.0 from TI.

Table 1-1. Device Drivers Summary

Peripheral	Description	Linux Driver Type	Device Support	See
Video Processing Back End (VPBE)	Enables Video Display for SD standard resolutions (NTSC, PAL) and ED resolutions	FBDev, V4L2	DM644x, DM365	
Video Processing Back End (VPBE)	Enables Video Display for SD standard resolutions (NTSC, PAL)	FBDev, V4L2	DM355	
Video Processing Front End (VPFE)	Enables Video Capture for SD standard resolutions (NTSC, PAL)	V4L2	DM644x, DM365	
Video Processing Front End (VPFE)	Enables Video Capture for SD standard resolutions (NTSC, PAL)	V4L2	DM355	
Video Port Interface (VPIF)	Enables Video Capture and Display for SD (NTSC, PAL) ED (480p, 576p) and HD (1080i, 720p) standard resolutions	V4LP	DM6467	
Preview Engine	Convert Bayer input to YCbCr format	Previewer	DM644x	
Resizer	Input image up-scaling up to 4x and downscaling up to 1/4x	Resizer	DM644x	
IPIPE	Convert Bayer input to YCbCr format Input image up-scaling up to 4x and downscaling up to1/16x	Previewer, Resizer	DM355, DM365	
Auto Exposure/Auto White Balance (AEW)	Auto Exposure/Auto White Balance for statistical collection of video data	AEW	DM644x, DM355, DM365	
Auto Focus (AF)	Auto Focus for statistical collection of video data	AF	DM644x, DM355, DM365	

Table 1-1. Device Drivers Summary (continued)

Peripheral	Description	Linux Driver Type	Device Support	See
Audio (McBSP)	Audio Record and Playback	ALSA	DM644x, DM355, DM365	
Video Data Conversion Engine (VDCE)	Video Processing and Conversions support down scaling, edge padding, range mapping, chrominance conversion, and blending		DM6467	
Audio (McASP)	Supports audio record and playback		DM644x, DM355, DM365	Section 2.5
Audio (McASP)	Supports audio record and playback and S/PDIF output		DM6467	Section 2.5
Ethernet	Transmit/receive network data. Supports Auto negotiation with 10/100 Mbps link speed.	Network Driver	DM644x, DM355, DM365	
Ethernet	Transmit/receive network data. Supports Auto negotiation with 10/100/1000 Mbps link speed.	Network Driver	DM6467	
USB 2.0 MSC Host	USB Mass Storage Class Host Driver		DM644x, DM355, DM6467, DM365	
USB 2.0 MSC Slave	USB Mass Storage Class Slave Driver	USB Gadget	DM644x, DM355, DM6467, DM365	
USB 2.0 RNDIS Slave	USB Remote Network Driver Interface specification	USB Gadget	DM644x, DM355, DM6467, DM365	
USB 2.0 CDC Slave	USB Communication Device Class Slave Driver	USB Gadget	DM644x, DM355, DM6467, DM365	
USB 2.0 HID Host	USB Human Interface Device Host Driver		DM644x, DM355, DM6467, DM365	
USB 2.0 ISO	USB Isochronous Device Driver for Video and Audio Devices		DM644x, DM355, DM6467, DM365	
ATA/ATAPI	Interface to ATA hard disk/CDROM	Block	DM644x, DM6467, DM365	
NAND	Flash Storage system	MTD Character and Block	DM644x, DM355, DM6467, DM365	Section 2.10
NOR	Flash Storage system	MTD Character and Block	DM644x	Section 2.11
MMC/SD	Interface to Multi Media /Secure digital cards	Block	DM644x, DM355, DM365	Section 2.9
UART	Serial Communication Interface	Character	DM644x, DM355, DM6467, DM365	Section 2.12
CIR	Consumer Infra Red Interface	Character	DM6467	
I^2C	Inter-IC Communication	Character	DM644x, DM355, DM6467, DM365	Section 2.13
SPI	Serial Peripheral Interface	Character	DM355, DM6467, DM365	Section 2.14
TSIF	Transport Stream Interface		DM6467	Section 2.15
VLYNQ	Serial Communication Interface that enables the extension of the internal bus		DM644x, DM6467, DM365	Section 2.16
PWM	Pulse Width Modulation typically used for motor control	Character	DM644x, DM355, DM6467, DM365	Section 2.17
Timers	General Purpose Timers and OS ticks		DM644x, DM355, DM6467, DM365	
Watchdog Timer (64-bit)	Monitor system condition		DM644x, DM355, DM6467, DM365	
GPIO	General Purpose IO		DM644x, DM355, DM6467, DM365	
PINMUX	Device Pin Configuration		DM644x, DM355, DM6467, DM365	
EDMA3	Enhanced DMA Engine	Linux DMA	DM644x, DM355, DM6467, DM365	

Table 1-1. Device Drivers Summary (continued)

Peripheral	Description	Linux Driver Type	Device Support	See
PCI Slave	Interconnect mechanism between peripheral add-in cards and processor/memory systems	Character	DM6467	

1.3 Limitations Summary

Table 1-2 through Table 1-4 summarize the known feature limitations in the LSP 2.10 Release 02.10.00.14 for DM644x/DM355/DM365/DM6467 based on MontaVista Linux Pro 5.0.0 applied.

Table 1-2. DaVinci Limitations Summary

PERIPHERAL	STATUS	USE CASE
HPI	No driver; no HPI peripheral on EVM	Host processor interface (16-bit address/data)
USB Printer Class	Class not validated	USB connected printer
USB Wireless Class	Class not validated	USB wireless device such as WLAN (thin or thick MAC) or Bluetooth

Table 1-3. DM644x Driver Limitations Summary

PERIPHERAL	STATUS	USE CASE
VLYNQ	Link between master and slave is not detected.	VLYNQ

Table 1-4. DM6467 Driver Limitations Summary

PERIPHERAL	STATUS	USE CASE
Ethernet	Gigabit Ethernet facility is not working.	Gigabit Ethernet transmit receive

Table 1-5. DM365 Driver Limitations Summary

PERIPHERAL	STATUS	USE CASE
VLYNQ	Link between master and slave is not detected.	VLYNQ

1.4 Toolchain and Version Information

The kernel and MontaVista Pro Linux (MVL) user space applications are compiled and linked with the MontaVista Linux Pro 5.0.0 arm_v5t_le toolchain. The Application Binary Interface (ABI) is GNU/Linux ARM9. The Linux kernel version in MontaVista Linux Pro 5.0.0 is 2.6.18.

The toolchain components consist of:

- GNU Compiler Collection (GCC) 4.2.0
- GNU libc (glibc) 2.5.90
- GNU Debugger (GDB) 6.6.50
- GNU Binary Utilities (binutils) 2.17.50

1.5 Tested Modes

The drivers have been tested in the following kernel preemption modes:

- Preemptible kernel (low latency desktop)
- Complete preemption (real-time)

1.6 Timers

DM644x EVM

The DaVinci DM644x EVM has three software-programmable general purpose timers. Two of the 64-bit timers are configured as four independent 32-bit timers as follows:

1. Timer 0
   - Timer 0 Low (1:2) - Free-running counter, used for cycle counter
   - Timer 0 High (3:4) - High-resolution programmable timer
2. Timer 1
   a. Timer 1 Low (1:2) - Reserved for DSP
   b. Timer 1 High (3:4) - Linux system tick
3. Timer 2 - User configurable only as a 64-bit watchdog timer.

DM355 EVM

The DM355 has two software-programmable general purpose timers. The 64-bit timers are configured as four independent 32-bit timers as follows:

1. Timer 0
   - Timer 0 Low (1:2) - Free-running counter, used for cycle counter
   - Timer 0 High (3:4) - High-resolution programmable timer
2. Timer 1
   - Timer 1 Low (1:2) - Reserved
   - Timer 1 High (3:4) - Linux system tick

DM6467 EVM

The DaVinci DM6467 EVM has three software-programmable general purpose timers. Two of the 64-bit timers are configured as four independent 32-bit timers as follows:

1. Timer 0
   - Timer 0 Low (1:2) - AV Sync
   - Timer 0 High (3:4) - Free-running timer
2. Timer 1
   a. Timer 1 Low (1:2) - Reserved for DSP
   b. Timer 1 High (3:4) - Linux system tick
3. Timer 2 - User configurable only as a 64-bit watchdog timer.

DM365 EVM

The DaVinci DM365 EVM has five software-programmable general purpose timers configured as follows:

1. Timer 0, Timer 1, Timer 3 and Timer 4 (general-purpose timers) can be programmed in 64-bit mode, dual 32-bit unchained mode, or dual 32-bit chained mode.
2. Timer 3 supports additional features over the other timers
   - External clock/event input
   - Period reload
   - Output event tied to Real Time Out (RTO) module
   - External event capture
   - Timer counter register read reset
3. Timer 2 - User configurable only as a watchdog timer.

1.7 LSP/PSP Version Information

This document applies to the driver set from the Production Release LSP 2.10.

1.8 Documentation Support

The following documents are available for download:

SPRUG85 *LSP 2.10 DaVinci Linux Consumer-IR Driver User's Guide*
describes how to install and use the Consumer-IR driver.

SPRUG86 *LSP 2.10 DaVinci Linux Resizer Device Driver User's Guide*
describes how to install and use the resizer driver.

SPRUG87 *LSP 2.10 DaVinci Linux Previewer Driver User's Guide*
describes how to install and use the previewer driver.

SPRUG88 *LSP 2.10 DaVinci Linux NOR Flash Driver User's Guide*
describes how to install and use the NOR Flash driver.

SPRUG91 *LSP 2.10 DaVinci Linux AF Driver User's Guide*
describes how to install and use the AF driver.

SPRUG92 LSP 2.10 DaVinci Linux VPBE Frame Buffer Driver User's Guide
describes how to install and use the VPBE Frame Buffer driver.

SPRUG93 *LSP 2.10 DaVinci Linux IPIPE Driver User's Guide*
describes how to install and use the IPIPE Driver.

SPRUG94 *LSP 2.10 DaVinci Linux EVM Installation User's Guide*
describes how to install the Linux EVM.

SPRUG95 *LSP 2.10 DaVinci Linux Video Sysfs User's Guide*
describes how to install and use the Video Sysfs driver.

SPRUG96 *LSP 2.10 DaVinci Linux V4L2 Display User's Guide*
describes how to install and use the V4L2 display driver.

SPRUG97 *LSP 2.10 DaVinci Linux PCI Slave Driver User's Guide*
describes how to install and use the PCI slave driver.

SPRUG98 *LSP 2.10 DaVinci Linux Audio Driver User's Guide*
describes how to install the Linux EVM.

SPRUGA0 LSP 2.10 DaVinci Linux VPIF Display Driver User's Guide
describes how to install and use the VPIF display driver.

SPRUGA1 LSP 2.10 DaVinci Linux PCI Boot Driver User's Guide
describes how to install and use the PCI boot driver.

SPRUGA2 LSP 2.10 DaVinci Linux TSIF Driver User's Guide
describes how to install and use the TSIF driver.

SPRUGA3 LSP 2.10 DaVinci Linux VDCE Driver User's Guide
describes how to install and use the VDCE driver.

SPRUGP3 LSP 2.10 DM365 Linux Face Detection Driver User's Guide
describes how to install and use the DM365 face detection driver.

SPRUGP6 LSP 2.10 DaVinci Linux VPFE Capture Driver User's Guide
describes how to install and use the VPFE capture driver.

1.9 DM644x

Audio, CCDC, EDMA, FBDEV, GPIO, H3A, NOR, Previewer, Resizer, V4L2 and VLYNQ driver applications are included in the PSP_02_10_00_14/examples/dm644x/ directory.

1.10 DM355

H3A and IPIPE driver applications are included in the PSP_02_10_00_14/examples/dm355/ directory.

1.11 DM6467

Audio, CIR, PCI, TSIF, VDCE, VLYNQ and VPIF driver applications are included in the PSP_02_10_00_14/examples/dm646x/ directory.

1.12 DM365

Audio, CCDC, EDMA, FBDEV, GPIO, H3A, NOR, Previewer, Resizer, V4L2 and VLYNQ driver applications are included in the PSP_02_10_00_14/examples/dm644x/ directory.

2 Linux Kernel Device Drivers

This section covers details on each of the device drivers from the LSP package. The scope of each device driver is to provide a high level description, features supported, features not supported, constraints, supported system calls and IOCTLs and performance and benchmarks for different MV kernel preemption modes. The performance measurements are done for a driver built with its default configuration.

2.1 DM644x Video Drivers

The DaVinci video driver is a set of video display, capture, statistics and processing components. Figure 2-1 shows the DM644x architecture.Figure 2-3 shows the DM646x architecture.

The video display driver for the DaVinci video processing back end (VPBE) peripheral is a *char* driver, compliant with the frame buffer driver (FBDev) and V4L2 framework. The device nodes created for the FBDev VPBE driver are:

- /dev/fb/0 - OSD0 Window
- /dev/fb/1 or /dev/video1 - VID0 Window
- /dev/fb/2 - OSD1 Window
- /dev/fb/3 or /dev/video3- VID1 Window

The device nodes created for V4L2 Display driver are:

- /dev/video2 - VID0 Window
- /dev/video3 - VID1 Window

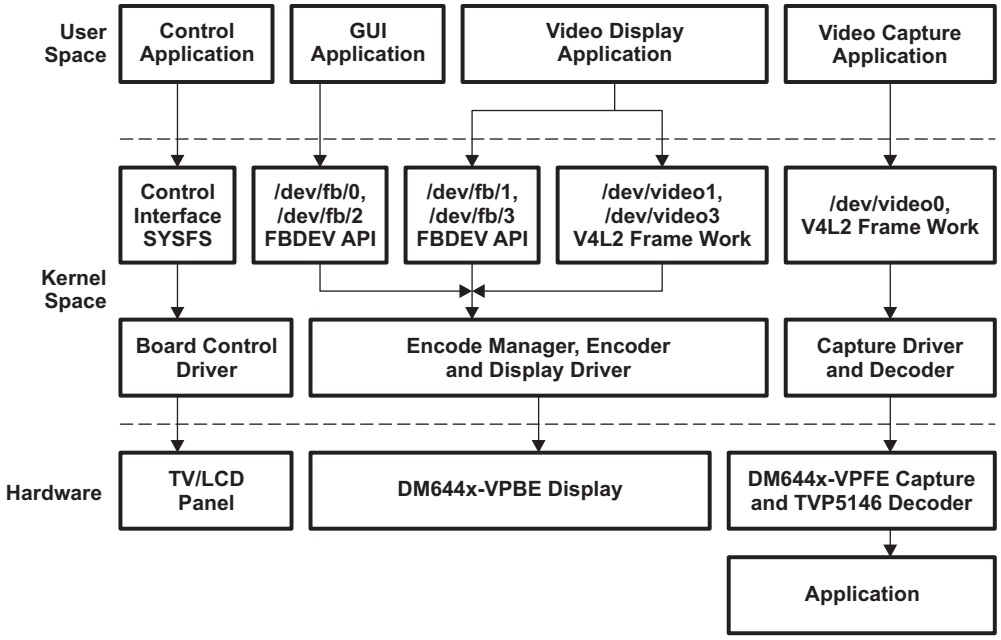

Figure 2-1. DM644x Video Driver Architecture

The video capture driver for the video processing front end (VPFE) peripheral is a *char* driver, compliant with the V4L2 specification. The device node created for the VPFE driver is /dev/video0.

The previewer, resizer, H3A (auto focus (AF), and auto expose/auto white balance (AEW)) drivers are front-end components that provide post-capture processing.

2.1.1 DM644x Video Display Driver

The VPBE display driver supports the following features:

- Two video windows (VID0 and VID1) in YUV422 format (UYVY). Both windows are triple buffered.
- OSD or bitmap window using OSD0 in RGB 565 format with double buffering
- Buffer exchange mechanism supports both blocking and non-blocking modes. These supports mmap'ed exchange mechanisms.
- Can be loaded dynamically or built statically into the Linux kernel.
- Attribute window acts as a per-pixel alpha plane using the hardware plane OSD1. This plane is double-buffered.
- Analog outputs using the four internal DACs
  - Composite video using one DAC
  - S-Video using two DACs
  - Component video using three DACs
- NTSC and PAL interlaced outputs using the internal video encoder
- Flexibility to change window sizes and starting coordinates
- Zoom feature to zoom the display to 1x, 2x and 4x
- 480p and 576p formats
- Digital output on RGB666 digital interface
- RGB888 input mode for the video window one
- Palletized input (1/2/4/8 bit) mode for OSD window
- Run time enable/disable of windows
- Color keying
- Programmed buffer flipping
- OSD0 and OSD1 windows are controlled only by FBDEV interface
- VID0 and VID1 windows can be controlled by both V4L2 and FBDEV interface
- Output and mode are set using sysfs interface
- ED display supports the following resolutions only:
  - 480p at 60 fps
  - 576p at 50 fps

2.1.1.1 Support and Constraints

Features Not Supported

- Analog RBG output
- The VID0 window cannot be used for interlaced video output.
- User pointer buffer exchange mechanism
- Currently V4L2 display is not supported.
- OSD windows are not supported by the V4L2 display driver

Constraints

- Dynamic removal of the module is not allowed when the FB console is in use.
- The vid0 and vid1 windows are mutually exclusive. This is a silicon issue.
- Noise at the top and right side of VID windows are observed on some low quality LCD TVs.
- Only VID0 will be usable in HD video modes (720p and 1080i).

Supported System Calls

open(), close(), ioctl(), mmap() and munmap()

2.1.1.2 Supported IOCTLs

The video display driver supports IOCTLs shown in Table 2-1.

Table 2-1. DM644x VPBE FBDEV IOCTLs Supported

Constant	Description
FBIO_GET_VIDEO_CONFIG_PARAMS	Gets the current video window configuration
FBIO_SET_VIDEO_CONFIG_PARAMS	Sets the configuration of the video window
FBIO_GET_BITMAP_CONFIG_PARAMS	Gets the existing configuration of the bitmap (OSD0 / OSD1) window
FBIO_SET_BITMAP_CONFIG_PARAMS	Sets the configuration of the BITMAP (OSD0 / OSD1) window
FBIO_SET_BACKG_COLOR	Sets the background color. The window should be disabled before using this command.
FBIOBLANK	Blank the display
FBIOGET_VSCREENINFO	Gets the variable screen information of the framebuffer. This command can be used for each framebuffer window.
FBIOPUT_VSCREENINFO	Sets variable screen parameters for framebuffer, including the window input format (resolution and bits per pixel)
FBIOGET_FSCREENINFO	Gets the fixed screen information of the framebuffer. This command can be used for each framebuffer window.
FBIOPUT_FSCREENINFO	Sets fixed screen parameters for framebuffer, including the window input format (resolution and bits per pixel)
FBIOPUTCMAP	Sets the pseudo palette
FBIOPAN_DISPLAY	Sets the display buffer for the window using the var_screeninfo offset of the buffer (out of number of buffers for the window: 3 for video and 2 for OSD) passed to the IOCTL. It calculates actual buffer location and sets it into the window register.
FBIO_SET_CURSOR	Configures cursor parameters
FBIO_WAITFORVSYNC	Synchronizes the buffered display by waiting for vsync before the call returns
FBIO_SETPOS	Changes the starting X and Y coordinate of the desired video or OSD plane
FBIO_SETPOSX	Changes the starting X coordinate of the desired video or OSD plane
FBIO_SETPOSY	Changes the starting Y coordinate of the desired video or OSD plane
FBIO_SETZOOM	Specifies the ZOOM parameters (identity, 2x or 4x) for the corresponding display plane

For more details refer http://www.linux-fbdev.org.

2.1.1.3 Performance and Benchmarks

The performance numbers were captured using the following:

- Sony Bravia KLV-S26A10 TV was used for Display testing
- SD Display benchmarks were captured using composite input/output video loop back
- ED Display was tested using composite input/component output video loop back

2.1.1.3.1 VPBE-FBDEV Low Latency Desktop Preemption

Table 2-2. DM644x VPBE-FBDEV Performance Values – LLD

Video Resolution	Frame Rate (Frames Per Second)	CPU Occupancy (%)
NTSC -60 Hz	30.07 fps	0.42
PAL -50 Hz	25.06 fps	0.15
480p-60 Hz	60.85 fps	0.48
576p-50 Hz	50.15 fps	0.00

2.1.1.3.2 VPBE-FBDEV Real Time Preemption

Table 2-3. DM644x VPBE-FBDEV Performance Values – RT

Video Resolution	Frame Rate (Frames Per Second)	CPU Occupancy (%)
NTSC -60 Hz	30.09 fps	0.78
PAL -50 Hz	25.06 fps	0.00
480p-60 Hz	60.12 fps	0.72
576p-50 Hz	50.17 fps	0.80

2.1.2 DM644x Video Capture Driver

The video capture driver supports the following features:

- Decoded video input in YUV422 format (UYVY or YUYV) with the help of an external decoder
- Can be loaded dynamically or built statically into the Linux kernel
- Buffer exchange mechanism supports blocking mode. This supports mmap'ed exchange mechanism.
- NTSC and PAL video input through an external decoder, including auto sensing capability
- Multi-buffered input with a minimum of three buffers, and support for more at run time
- Cropped input image with programmable start coordinates. This feature can be used to input an image at smaller sizes (e.g. QCIF). For example, the coordinates can be programmed to pick the image from the center of the capture scene
- Bayer pattern input, fault pixel correction, progressive input, LPF and time stamping
- The VPFE driver supports several IOCTLs. These IOCTLs are explained in the V4L2 specification.

2.1.2.1 Support and Constraints

Features Not Supported The driver does not support HD input.

Constraints None.

Supported System Calls open(), close(), ioctl(), mmap() and munmap()

Supported IOCTLs The video capture driver supports IOCTLs shown in Table 2-4.

Table 2-4. DM644x VPFE-V4L2 IOCTLs

Constant	Description
VIDIOC_CROPCAP	Information about the video cropping and scaling abilities
VIDIOC_ENUMINPUT	Enumerates video inputs
VIDIOC_ENUMOUTPUT	Enumerates video outputs
VIDIOC_ENUMSTD	Enumerates supported video standards
VIDIOC_G_CROP, VIDIOC_S_CROP	Gets or sets the current cropping rectangle
VIDIOC_G_CTRL, VIDIOC_S_CTRL	Gets or sets the value of a control
VIDIOC_G_FMT, VIDIOC_S_FMT, VIDIOC_TRY_FMT	Gets or sets the data format, try a format
VIDIOC_G_INPUT, VIDIOC_S_INPUT	Queries or selects the current video input
VIDIOC_G_PARM, VIDIOC_S_PARM	Gets or sets streaming parameters
VIDIOC_G_STD, VIDIOC_S_STD	Queries or selects the video standard of the current input
VIDIOC_QBUF, VIDIOC_DQBUF	Exchanges a buffer with the driver
VIDIOC_QUERYBUF	Queries the status of a buffer
VIDIOC_QUERYCAP	Queries device capabilities
VIDIOC_QUERYCTRL, VIDIOC_QUERYMENU	Enumerates controls and menu control items
VIDIOC_QUERYSTD	Senses the video standard received by the current input
VIDIOC_REQBUFS	Initiates memory mapping or user pointer I/O
VIDIOC_STREAMON, VIDIOC_STREAMOFF	Starts or stops streaming I/O
VPFE_CMD_S_MT9T001_PARAMS	Configures mt9t001 parameters

Table 2-4. DM644x VPFE-V4L2 IOCTLs (continued)

Constant	Description
VPFE_CMD_G_MT9T001_PARAMS	Gets mt9t001 parameters
VPFE_CMD_CONFIG_CCDC_RAW	Configures CCDC for raw mode
VPFE_CMD_CONFIG_CCDC_YCBCR	Configures CCDC for YUV mode

For more details refer to http://v4l2spec.bytesex.org/v4l2spec/.

2.1.2.2 Performance and Benchmarks

The performance numbers were captured using the following:

- Sony DVPNS-61P DVD player was used for capture testing
- SD Display benchmarks were captured using composite input/output video loop back

2.1.2.2.1 VPFE-V4L2 Low Latency Desktop Preemption

Table 2-5. DM644x VPFE-V4L2 Performance Values – LLD

Video Resolution	Frame Rate (Frames Per Second)	CPU Occupancy (%)
NTSC -60 Hz	30.03 fps	0.84
PAL -50 Hz	25.05 fps	0.10

2.1.2.2.2 VPFE-V4L2 Real Time Preemption

Table 2-6. DM644x VPFE-V4L2 Performance Values – RT

Video Resolution	Frame Rate (Frames Per Second)	CPU Occupancy (%)
NTSC -60 Hz	30.03 fps	0.84
PAL -50 Hz	25.05 fps	0.20

2.1.3 DM644x Preview Engine Driver

The Preview Engine Driver supports the following features:

- Input image in Bayer pattern
- Various hardware configurations such as CFA interpolation, DFC subtract, and so on.
- Accepts input from SDRAM or DDRAM
- Converts Bayer pattern input image to YCbCr 4:2:2 format
- Can be loaded dynamically or built statically into the Linux kernel
- Buffer exchange mechanism supports blocking mode. These supports mmap'ed exchange mechanism.

2.1.3.1 Support and Constraints

Features Not Supported User pointer buffer exchange mechanism

Constraints

- The Preview Engine will not work with the video port enabled in the CCDC driver.
- The Preview Engine and H3A drivers cannot be used together.

Supported System Calls open(), close(), ioctl(), mmap() and munmap()

Supported IOCTLs The Preview Engine Driver supports the IOCTLs shown in Table 2-7

Table 2-7. DM644x Preview Engine IOCTLs

Constant	Description
PREV_REQBUF	Requests frame buffer to be allocated by PREV module
PREV_QUERYBUF	Requests physical address of buffers allocated by the PREV_REQBUF ioctl
PREV_SET_PARAM	Sets the Preview Engine hardware parameters

Table 2-7. DM644x Preview Engine IOCTLs (continued)

Constant	Description
PREV_GET_PARAM	Gets the Preview Engine hardware parameters
PREV_PREVIEW	Submits a previewing task to the hardware
PREV_GET_STATUS	Gets the current status of Preview Engine hardware parameters
PREV_GET_CROPSIZE	Returns the size reduction in the output image compared to input image in terms of number of pixels per line and number of lines depending on features enabled
PREV_SET_EXP	Sets allowable delay between consecutive read requests from Preview Engine module

2.1.4 DM644x Resizer Driver

The Resizer Driver supports the following features:

- YUV422 color interleaved and 8-bit color separate data input formats
- Input from CCDC, SDRAM or DDRAM
- Standalone kernel module that can be used by multiple applications
- Upscaling/downscaling multiplier from one-fourth to 4x
- Can be loaded dynamically or built statically into the Linux kernel.
- Buffer exchange mechanism supports blocking mode. This supports mmap'ed exchange mechanism.

2.1.4.1 Support and Constraints

Features Not Supported	User pointer buffer exchange mechanism
Constraints	None
Supported System Calls	open(), close(), ioctl(), mmap() and mumap()
Supported IOCTLs	The resizer driver supports the IOCTLs shown in Table 2-8

Table 2-8. DM644x Resizer Driver Supported IOCTLs

Constant	Description
RSZ_REQBUF	Requests frame buffers to be allocated by the RSZ module
RSZ_QUERYBUF	Requests physical address of buffers allocated by the RSZ_REQBUF
RSZ_S_PARAM	Sets the Resizer hardware parameters associated with this logic channel
RSZ_G_PARAM	Gets the Resizer hardware parameters associated with this logic channel
RSZ_RESIZE	Submits a resizing task to the logic channel associated with specified channel descriptor
RSZ_G_STATUS	Gets the current status of the hardware
RSZ_S_PRIORITY	Sets the current priority setting of the logic channel identified by the channel descriptor
RSZ_G_PRIORITY	Gets the Resizer hardware parameters associated with the logic channel specified by channel descriptor
RSZ_GET_CROPSIZE	Returns the size reduction in the output image compared to input image in terms of number of pixels per line and number of lines depending on features enabled
RSZ_S_EXP	Sets allowable delay between consecutive read requests from Resizer module

2.1.5 DM644x Auto Expose/Auto White Balance (AEW) Driver

The Auto Expose/Auto White Balance (AEW) Driver supports the following features:

- Image in Bayer pattern
- Input from CCD controller
- Can be loaded dynamically or built statically into the Linux kernel.

2.1.5.1 Support and Constraints

Features Not Supported	None
Constraints	None
Supported System Calls	open(), close(), ioctl(), mmap() and mumap()
Supported IOCTLs	The AEW driver supports the IOCTLs shown in Table 2-9

Table 2-9. DM644x AEW IOCTLs

Constant	Description
AEW_S_PARAM	Sets the following parameters/modules of the AEW hardware: • Window parameters • Black window parameters • Saturation check module • A-Law compression module
AEW_G_PARAM	Gets the AEW hardware settings
AEW_ENABLE	Enables AEW Engine with parameters set by AEW_S_PARAM
AEW_DISABLE	Disables AEW Engine

2.1.6 DM644x AF Driver

The AF Driver supports the following features:

- Image in Bayer pattern
- Input from CCD controller
- Can be loaded dynamically or built statically into the Linux kernel.

2.1.6.1 Support and Constraints

Features Not Supported	None
Constraints	None
Supported System Calls	open(), close(), ioctl(), mmap() and mumap()
Supported IOCTLs	The AF driver supports the IOCTLs shown in Table 2-10

Table 2-10. DM644x AF IOCTLs

Constant	Description
AF_S_PARAM	Sets the following parameters/modules of the AF hardware: • Paxel parameters • IIR Filter parameters • HMF Filter Parameter • A-Law compression module
AF_G_PARAM	Gets the AF hardware settings
AF_ENABLE	Enables AF Engine with parameters set by AF_S_PARAM
AF_DISABLE	Disables AF Engine

2.2 DM355 Video Drivers

The DaVinci video driver is a set of video display, capture, statistics, and processing components.

The video display driver for the DaVinci Video Processing Back End (VPBE) peripheral is a char driver, compliant with the Frame Buffer Driver (FBDEV) framework and V4l2 framework. Following are the device nodes created for the VPBE driver:

- /dev/fb/0 – OSD0 Window
- /dev/fb/1 or /dev/video1 – Vid0 Window
- /dev/fb/2 – OSD1 Window
- /dev/fb/3 or /dev/video3 – Vid1 Window

The video capture driver for the Video Processing Front End (VPFE) peripheral is a char driver, compliant with the V4L2 framework. The device node created for the VPFE driver is /dev/video0.

The IPIPE (preview engine, resizer) and H3A (Auto Focus AF, Auto Expose/Auto White Balance AEW) drivers are front-end components that provide post-capture processing.

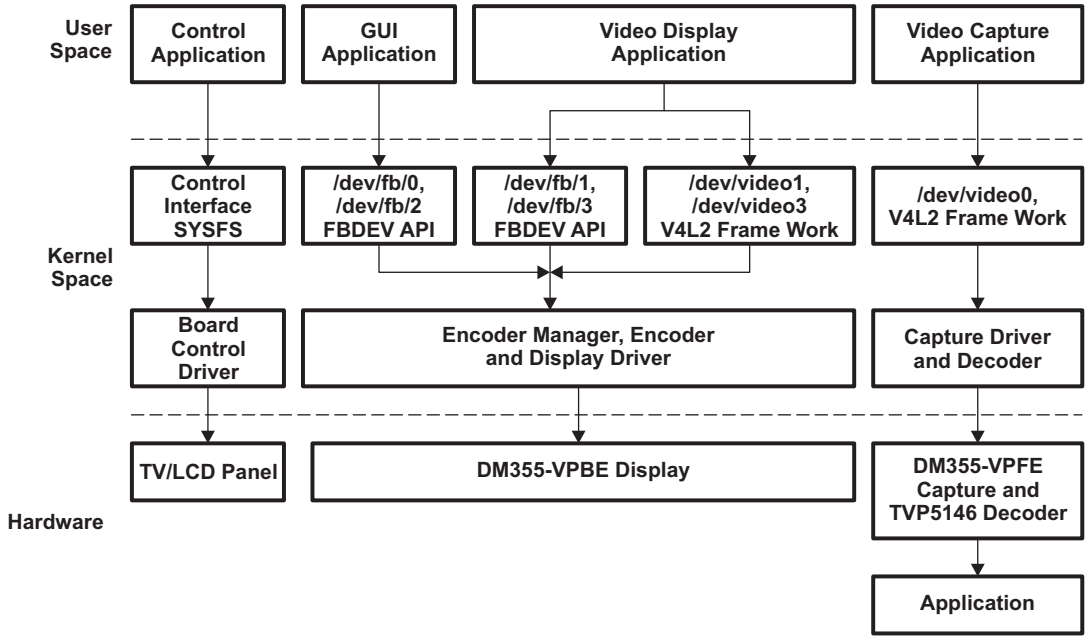

Figure 2-2. DM355 Video Driver Architecture

2.2.1 DM355 Video Display Driver

The Video Display Driver supports the following features:

- Two video windows (VID0 and VID1) supports YUV422 format. Both windows are triple-buffered.
- OSD or bitmap window using OSD0 in RGB 565 format with double buffering
- Buffer exchange mechanism supports both blocking and non-blocking modes. These supports mmap'ed exchange mechanisms.
- Can be loaded dynamically or built statically into the Linux kernel.
- Attribute window acts as a per-pixel alpha plane using the hardware plane OSD1. This plane is double-buffered.
- Analog outputs using the four internal DACs:
  - Composite video using one DAC
  - S-Video using two DACs
- NTSC and PAL interlaced outputs using the internal video encoder

- Flexibility to change window sizes and starting coordinates
- Zoom feature to zoom the display to 1x, 2x and 4x
- Digital output on RGB666 digital interface
- RGB888 input mode for the video window one
- Palletized input (1/2/4/8 bit) mode for OSD window
- Run-time enable/disable of windows
- Color-keying
- Programmed buffer flipping
- OSD0 and OSD1 windows are controlled only by FBDEV interface
- VID0 and VID1 windows can be controlled by both V4L2 and FBDEV interface
- Output and mode is set using sysfs interface

2.2.1.1 Support and Constraints

Features Not Supported

- Analog RBG output
- The VID0 window cannot be used for interlaced video output.
- User pointer buffer exchange mechanism
- OSD windows are not supported by the V4L2 driver.
- Currently V4L2 display is not supported.

Constraints

- Dynamic removal of the module is not allowed when the FB console is in use.
- Vid0 and Vid1 are mutually exclusive. This is a silicon issue.

Supported System Calls

open(), close(), ioctl(), mmap() and munmap().

Supported IOCTLs

The Video Display driver supports the IOCTLs shown in Table 2-11.

Table 2-11. DM355 VPBE-FBDEV IOCTLs

Constant	Description
FBIO_GET_VIDEO_CONFIG_PARAMS	Gets the current video window configuration
FBIO_SET_VIDEO_CONFIG_PARAMS	Sets the configuration of the video window
FBIO_GET_BITMAP_CONFIG_PARAMS	Gets the existing configuration of the bitmap (OSD0 / OSD1) window
FBIO_SET_BITMAP_CONFIG_PARAMS	Sets the configuration of the BITMAP (OSD0 / OSD1) window
FBIO_SET_BACKG_COLOR	Sets the background color. The window should be disabled before using this command.
FBIOBLANK	Blank the display
FBIOGET_VSCREENINFO	Gets the variable screen information of the framebuffer. This command can be used for each framebuffer window.
FBIOPUT_VSCREENINFO	Sets variable screen parameters for framebuffer, including the window input format (resolution and bits per pixel)
FBIOGET_FSCREENINFO	Gets the fixed screen information of the framebuffer. This command can be used for each framebuffer window.
FBIOPUT_FSCREENINFO	Sets fixed screen parameters for framebuffer, including the window input format (resolution and bits per pixel)
FBIOPUTCMAP	Sets the pseudo palette
FBIOPAN_DISPLAY	Sets the display buffer for the window using the var_screeninfo offset of the buffer (out of number of buffers for the window: 3 for video and 2 for OSD) passed to the IOCTL. It calculates actual buffer location and sets it into the window register.
FBIO_SET_CURSOR	Configures cursor parameters
FBIO_WAITFORVSYNC	Synchronizes the buffered display by waiting for vsync before the call returns
FBIO_SETPOS	Changes the starting X and Y coordinate of the desired video or OSD plane
FBIO_SETPOSX	Changes the starting X coordinate of the desired video or OSD plane

Table 2-11. DM355 VPBE-FBDEV IOCTLs (continued)

Constant	Description
FBIO_SETPOSY	Changes the starting Y coordinate of the desired video or OSD plane
FBIO_SETZOOM	Specifies the ZOOM parameters (identity, 2x or 4x) for the corresponding display plane

For more details refer http://www.linux-fbdev.org.

2.2.1.2 Performance and Benchmarks

The performance numbers were captured using the following:

- Sony Bravia KLV-S26A10 TV was used for Display testing.
- SD Display benchmarks were captured using composite input/output video loop back.

2.2.1.2.1 VPBE-FBDEV Low Latency Desktop Preemption

Table 2-12. DM355 VPBE-FBDEV Performance – LLD

Video Resolution	Frame Rate (Frames Per Second)	CPU occupancy (%)
NTSC -60 Hz	30.12 fps	0.84
PAL -50 Hz	25.05 fps	0.95

2.2.1.2.2 VPBE-FBDEV Real Time Preemption

Table 2-13. DM355 VPBE-FBDEV Performance – RT

Video Resolution	Frame Rate (Frames Per Second)	CPU occupancy (%)
NTSC -60 Hz	30.31 fps	1.61
PAL -50 Hz	25.12 fps	0.40

2.2.2 DM355 Video Capture Driver

The Video Capture Driver supports the following features:

- Decoded video input in YUV422 format (UYVY or YUYV) with the help of an external decoder.
- Can be loaded dynamically or built statically into the Linux kernel.
- Buffer exchange mechanism supports blocking mode. These supports mmap'ed exchange mechanism.
- NTSC and PAL video input through an external decoder, including auto-sensing capability.
- Multi-buffered input with minimum three buffers and supports more buffers at run time.
- Cropped input image with programmable start coordinates. This feature can be used to input an image of smaller size (example, QCIF). For example, the coordinates can be programmed to pick the image from the center of the capture scene.
- Bayer pattern input, fault pixel correction, progressive input, LPF, and time stamping.
- The VPFE driver supports several IOCTLS. These IOCTLs are explained in the V4L2 specification.

2.2.2.1 Support and Constraints

Features Not Supported	HD input
Constraints	Video artifacts like brightness/contrast issues, jitters, shadowing effect are observed in the images.
Supported System Calls	open(), close(), ioctl(), mmap() and munmap().
Supported IOCTLs	The Video Capture driver supports the IOCTLs shown in Table 2-14.

Table 2-14. DM355 VPFE-V4L2 IOCTLs

Constant	Description
VIDIOC_CROPCAP	Information about the video cropping and scaling abilities
VIDIOC_ENUMINPUT	Enumerates video inputs
VIDIOC_ENUMOUTPUT	Enumerates video outputs
VIDIOC_ENUMSTD	Enumerates supported video standards
VIDIOC_G_CROP, VIDIOC_S_CROP	Gets or sets the current cropping rectangle
VIDIOC_G_CTRL, VIDIOC_S_CTRL	Gets or sets the value of a control
VIDIOC_G_FMT, VIDIOC_S_FMT, VIDIOC_TRY_FMT	Gets or sets the data format, try a format
VIDIOC_G_INPUT, VIDIOC_S_INPUT	Queries or selects the current video input
VIDIOC_G_PARM, VIDIOC_S_PARM	Gets or sets streaming parameters
VIDIOC_G_STD, VIDIOC_S_STD	Queries or selects the video standard of the current input
VIDIOC_QBUF, VIDIOC_DQBUF	Exchanges a buffer with the driver
VIDIOC_QUERYBUF	Queries the status of a buffer
VIDIOC_QUERYCAP	Queries device capabilities
VIDIOC_QUERYCTRL, VIDIOC_QUERYMENU	Enumerates controls and menu control items
VIDIOC_QUERYSTD	Senses the video standard received by the current input
VIDIOC_REQBUFS	Initiates memory mapping or user pointer I/O
VIDIOC_STREAMON, VIDIOC_STREAMOFF	Starts or stops streaming I/O
VPFE_CMD_S_MT9T001_PARAMS	Configures mt9t001 parameters
VPFE_CMD_G_MT9T001_PARAMS	Gets mt9t001 parameters
VPFE_CMD_CONFIG_CCDC_RAW	Configures CCDC for raw mode
VPFE_CMD_CONFIG_CCDC_YCBCR	Configures CCDC for YUV mode

For more details refer http://v4l2spec.bytesex.org/v4l2spec/.

2.2.2.2 Performance and Benchmarks

The performance numbers were captured using the following:

- Sony DVPNS-61P DVD player was used for capture testing.
- SD Display benchmarks were captured using composite input/output video loop back.

2.2.2.2.1 VPFE-V4L2 Low Latency Desktop Preemption

Table 2-15. DM355 VPFE-V4L2 Performance – LLD

Video Resolution	Frame Rate (Frames Per Second)	CPU occupancy (%)
NTSC -60 Hz	30.03 fps	0.90
PAL -50 Hz	25.05 fps	0.15

2.2.2.2.2 VPFE-V4L2 Real Time Preemption

Table 2-16. DM355 VPFE-V4L2 Performance – RT

Video Resolution	Frame Rate (Frames Per Second)	CPU occupancy (%)
NTSC -60 Hz	30.03 fps	1.61
PAL -50 Hz	25.05 fps	0.50

2.2.3 DM355 Previewer-Resizer Driver

The Previewer-Resizer Driver supports the following features using the DM355 IPIPE Hardware module:

- Previewing the capture input
- Resizing the capture input
- Various hardware configurations such as CFA interpolation, and DFC subtract.
- Accepts input from CCDC, SDRAM or DDRAM
- Converts Bayer pattern input image to YCbCr 4:2:2 format
- Can be loaded dynamically or built statically into the Linux kernel.
- Buffer exchange mechanism supports blocking mode. These supports mmap'ed exchange mechanism.
- YUV422 color interleaved and 8-bit color separate data input formats
- Standalone kernel module that can be used by multiple applications
- Upscaling/downscaling multiplier from 1/16 to 4x

The previewer-resizer driver does not support the DM6446 or DM6467 EVM. For the same previewing and resizing functionalities on DM6446/DM6467, previewer and resizer drivers should be used.

2.2.3.1 Support and Constraints

Features Not Supported	The driver does not support multichannels. When multiple open() calls are made, the driver returns an error.
Constraints	Video artifacts like jitter, non-sharp edges, chroma artifacts, horizontal lines in displayed image when no scaling is done.
Supported System Calls	open(), close(), ioctl(), mmap() and munmap().
Supported IOCTLs	The IPIPE driver supports the IOCTLs shown in Table 2-17.

Table 2-17. DM355 Previewer-Resizer Driver Supported IOCTLs

Constant	Description
PREV_ENUM_CAP	Enumerates the preview capabilities available to tune the image
PREV_REQBUF, RSZ_REQBUF	Requests either kernel-allocated buffer or user-allocated buffers to do I/O with previewer/resizer
PREV_QUERYBUF, RSZ_QUERY_BUF	Queries the buffer address for kernel-allocated buffers
PREV_S_CONFIG, PREV_G_CONFIG	Set or gets the input size, pixel format, and data source etc at the previewer
RSZ_S_CONFIG, RSZ_G_CONFIG	Sets or gets the input size, pixel format, and data source etc at the resizer
PREV_S_OPER_MODE, RSZ_S_OPER_MODE PREV_G_OPER_MODE, RSZ_G_OPER_MODE	Sets or gets the driver operation mode ("Continuous" or Single Shot mode)
PREV_S_PARAM	Sets a module parameter in the previewer to tune image
PREV_G_PARAM	Gets a module parameter in the previewer
PREV_S_CONTROL	Sets a control on a module. This is similar to PREV_S_PARAM except that it can be issued while streaming is on.
PREV_G_CONTROL	Gets a control parameter
PREV_PREVIEW, RSZ_RESIZE	Starts preview or Resize (or both) in Single shot mode

2.2.4 DM355 Auto Expose/Auto White Balance (AEW) Driver

The AEW Driver supports the following features:

- Image in Bayer pattern
- Input from CCD Controller
- Can be loaded dynamically or built statically into the Linux kernel.

2.2.4.1 Support and Constraints

Features Not Supported	None
Constraints	None
Supported System Calls	open(), close(), ioctl(), mmap() and munmap().
Supported IOCTLs	The AEW driver supports the IOCTLs shown in Table 2-18.

Table 2-18. DM355 AEW IOCTLs

Constant	Description
AEW_S_PARAM	Sets the following parameters/modules of the AEW hardware: • Window parameters • Black window parameters • Saturation check module • A-Law compression module
AEW_G_PARAM	Gets the AEW hardware settings
AEW_ENABLE	Enables AEW Engine with parameters set by AEW_S_PARAM
AEW_DISABLE	Disables AEW Engine

2.2.5 DM355 AF Driver

The AF Driver supports the following features:

- Image in Bayer pattern
- Input from CCD Controller
- Can be loaded dynamically or built statically into the Linux kernel.

2.2.5.1 Support and Constraints

Features Not Supported	None
Constraints	None
Supported System Calls	open(), close(), ioctl(), mmap() and munmap().
Supported IOCTLs	The AF driver supports the IOCTLs shown in Table 2-19.

Table 2-19. DM355 AF IOCTLs

Constant	Description
AF_S_PARAM	Sets the following parameters/modules of the AF hardware: • Paxel parameters • IIR Filter parameters • HMF Filter Parameter • A-Law compression module
AF_G_PARAM	Gets the AF hardware settings
AF_ENABLE	Enables AF Engine with parameters set by AF_S_PARAM
AF_DISABLE	Disables AF Engine

2.3 DM6467 Video Drivers

The video display driver for the DaVinci DM6467 VPIF peripheral is a char driver, compliant with the V4L2 framework. By default, the device nodes created for display driver are /dev/video2 (VID2 Channel) and /dev/video3 (VID3 Channel).

The video capture driver for the VPIF peripheral is a char driver, compliant with the V4L2 framework. By default, the device nodes created for the VPIF capture driver are /dev/video0 (VID0 Channel) and /dev/video1 (VID1 Channel).

The VDCE driver is a char driver, compliant with UDEV framework. The device node created for the VDCE driver is /dev/DavinciHD_vdce.

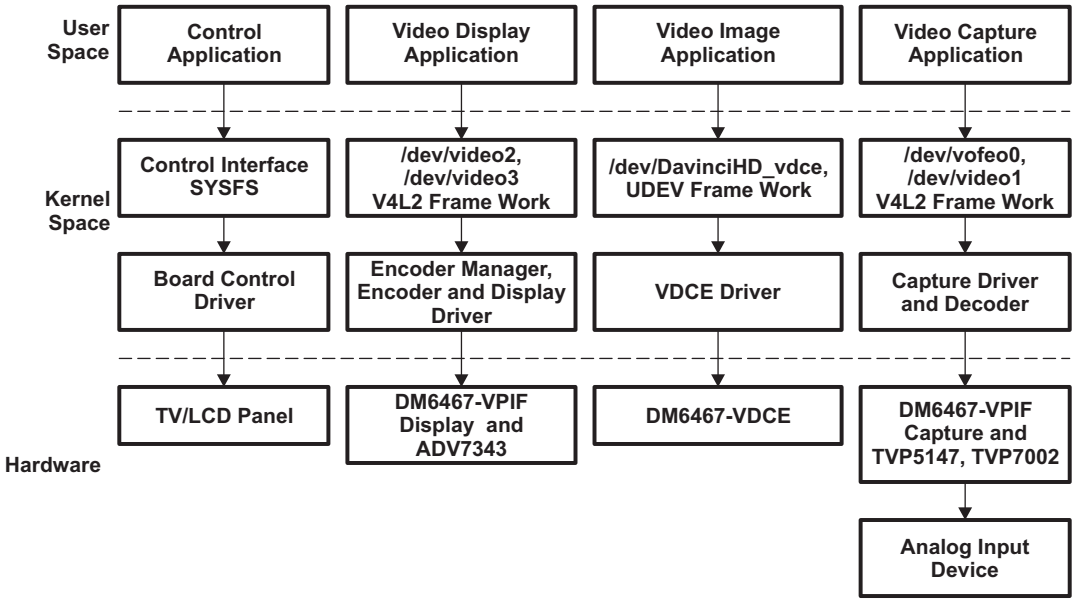

Figure 2-3. DM6467 Video Driver Architecture

2.3.1 DM6467 Video Display Driver

The Video Display Driver supports the following features:

- By default, two software channels of display are supported. Both the software channels will support Standard Definition (SD) display but only one (video2) channel will support High Definition (HD) display.
- Buffer exchange mechanism supports both blocking and non-blocking modes. These supports both mmap'ed and user pointer exchange mechanisms.
- Can be loaded dynamically or built statically into the Linux kernel.
- Different video resolutions are supported
  - SD display supports the following resolutions only:
    - NTSC 480i at 30 fps
    - PAL 576i at 25 fps
  - ED display supports the following resolutions only:
    - 480p at 60 fps
    - 576p at 50 fps
  - HD display supports the following resolutions only:
    - 720p at 60 fps
    - 720p at 50 fps

- 1080i at 30 fps
- 1080i at 25 fps

- Dynamic switching between various resolutions with some restriction.
- Dynamic switching of output interfaces with some restriction.
- The following VBI services:
  - CGMS and Closed Captioning on NTSC display
  - WSS on PAL display
- IOCTLs to support run-time change of the following parameters for ADV7343:
  - Brightness
  - Hue
  - Saturation

2.3.1.1 Support and Constraints

Features Not Supported All the remaining parameters for ADV7343 other than mentioned in Section 2.3.2.1.

Constraints

- Dynamic switching of resolution and output is not supported when streaming is on.
- VPIF input/output buffer addresses must be multiple of 8 bytes.
- VIDIOC_S_FMT IOCTL can be called only after VIDIOC_REQBUFS IOCTL for user pointer buffer exchange mechanism.
- If the user specifies less than three buffers while inserting the module, the driver allocates three buffers. If user has specified zero buffers, the driver assumes user pointer buffer exchange mechanism.
- If the user specifies a buffer size less than the required size for an NTSC image, the driver assumes the minimum buffer size for NTSC.

Supported System Calls open(), close(), ioctl(), mmap() and mumap()

Supported IOCTLs The Video Display driver supports the IOCTLs shown in Table 2-20

Table 2-20. DM6467 VPIF Display IOCTLs

Constant	Description
VIDIOC_ENUM_FMT	Enumerates supported formats by current decoder
VIDIOC_S_PRIORITY,VIDIOC_G_PRIORITY	Sets and gets the priority for file descriptor
VIDIOC_G_FMT, VIDIOC_S_FMT, VIDIOC_TRY_FMT	Gets or sets the data format, try a format
VIDIOC_QBUF, VIDIOC_DQBUF	Exchanges a buffer with the driver
VIDIOC_QUERYBUF	Queries the status of a buffer
VIDIOC_QUERYCAP	Queries device capabilities
VIDIOC_REQBUFS	Initiates memory mapping or user pointer I/O
VIDIOC_STREAMON, VIDIOC_STREAMOFF	Starts or stops streaming I/O
VIDIOC_CROPCAP	Gets cropping parameters

For more details refer http://v4l2spec.bytesex.org/v4l2spec/.

2.3.1.2 Performance Benchmarks

The performance numbers were captured using the following:

- Sony Bravia KLV-S26A10 TV was used for Display testing.
- SD Display benchmarks were captured using composite input/output video loop back.
- HD Display was tested using component input/output video loop back.

2.3.1.2.1 VPIF Display Low Latency Desktop Preemption

Table 2-21. DM6467 VPIF Display Performance Values – LLD

Video Resolution	Frame Rate (Frames Per Second)	CPU Occupancy (%)
NTSC -60 Hz	30.03 fps	1.26
PAL -50 Hz	25.05 fps	1.60
480p-60 Hz	60.06 fps	1.92
576p-50 Hz	50.10 fps	0.00

2.3.1.2.2 VPIF Display Real Time Preemption

Table 2-22. DM6467 VPIF Display Performance Values – RT

Video Resolution	Frame Rate (Frames Per Second)	CPU Occupancy (%)
NTSC -60 Hz	30.03 fps	1.55
PAL -50 Hz	25.05 fps	24.95
480p -60 Hz	60.06 fps	3.47
576p -50 Hz	50.10 fps	0.00

2.3.2 DM6467 Video Capture Drivers

The Video Capture Driver supports the following features:

- By default, two software channels of capture are supported. Both the software channels will support Standard Definition (SD) mode but only one (video0) channel will support High Definition (HD) capture. At an instant we can have two instances of SD operational or one instance of HD operational.
- Buffer exchange mechanism supports both blocking and non-blocking mechanisms. These support both mmap'ed and user pointer exchange mechanisms.
- Can be loaded dynamically or built statically into the Linux kernel.
- Two different video resolutions are supported - Standard Definition (SD), High definition (HD)
  - Standard Definition (SD) capture supports the following resolutions only:
    - NTSC 480i at 30 fps
    - PAL 576i at 25 fps
  - ED capture supports the following resolutions only:
    - 480p at 60 fps
    - 576p at 50 fps
  - High definition (HD) capture supports the following resolutions only:
    - 720p at 60 fps
    - 720p at 50 fps
    - 1080i at 30 fps
    - 1080i at 25 fps
- Dynamic switching between various resolutions with restrictions. See Section 2.3.2.1 for details.
- IOCTLs to support run-time change of the following parameters for TVP5147:
  - Brightness
  - Hue
  - Saturation

  - Contrast
  - Auto gain
- IOCTLs to support run-time change of the following parameters for TVP7002:
  - Brightness
  - Hue
  - Saturation
  - Contrast

2.3.2.1 Support and Constraints

Features Not Supported Raw Capture mode

Constraints

- Dynamic switching of resolution and output is not supported when streaming is on.
- VPIF input/output buffer addresses must be multiple of 8 bytes.
- VIDIOC_S_FMT IOCTL can be called only after VIDIOC_REQBUFS IOCTL for user pointer buffer exchange mechanism.
- If the user specifies less than three buffers while inserting the module, the driver allocates three buffers. If user has specified zero buffers, the driver assumes user pointer buffer exchange mechanism.
- If the user specifies a buffer size less than the required size for an NTSC image, the driver assumes the minimum buffer size for NTSC.

Supported System Calls open(), close(), ioctl(), mmap() and mumap()

Supported IOCTLs The Video Capture driver supports the IOCTLs shown in Table 2-23

Table 2-23. DM6467 VPIF Capture IOCTLs

Constant	Description
VIDIOC_ENUMINPUT	Enumerates video inputs
VIDIOC_ENUM_FMT	Enumerates supported formats by current decoder
VIDIOC_ENUMSTD	Enumerates supported video standards
VIDIOC_S_PRIORITY,VIDIOC_G_PRIORITY	Sets and gets the priority for file descriptor
VIDIOC_G_CTRL, VIDIOC_S_CTRL	Gets or sets the value of a control
VIDIOC_G_FMT, VIDIOC_S_FMT, VIDIOC_TRY_FMT	Gets or sets the data format, try a format
VIDIOC_G_INPUT, VIDIOC_S_INPUT	Queries or selects the current video input
VIDIOC_G_STD, VIDIOC_S_STD	Queries or selects the video standard of the current input
VIDIOC_QBUF, VIDIOC_DQBUF	Exchanges a buffer with the driver
VIDIOC_QUERYBUF	Queries the status of a buffer
VIDIOC_QUERYCAP	Queries device capabilities
VIDIOC_QUERYCTRL,	Enumerates controls items
VIDIOC_QUERYSTD	Senses the video standard received by the current input
VIDIOC_REQBUFS	Initiates memory mapping or user pointer I/O
VIDIOC_STREAMON, VIDIOC_STREAMOFF	Starts or stops streaming I/O
VIDIOC_CROPCAP	Gets cropping parameters
VIDIOC_G_SLICED_VBI_CAP	Gets the Sliced VBI parameters

For more details refer http://v4l2spec.bytesex.org/v4l2spec/.

2.3.2.2 Performance and Benchmarks

The performance numbers were captured using the following:

- Samsung Blu-Ray BD-P1000 player was used as an input source for testing.
- SD Display benchmarks were captured using composite input/output video loop back.
- HD Display was tested using component input/output video loop back.
- ARM clock frequency and the DDR frequency was 297 MHz.

2.3.2.2.1 VPIF Capture Low Latency Desktop Preemption

Table 2-24. DM6467 VPIF Capture Performance Values – LLD

Video Resolution	Frame Rate (Frames Per Second)	CPU Occupancy (%)
NTSC -60 Hz	30.03 fps	0.48
PAL -50 Hz	25.05 fps	0.00

2.3.2.2.2 VPIF Capture Real Time Preemption

Table 2-25. DM6467 VPIF Capture Performance Values – RT

Video Resolution	Frame Rate (Frames Per Second)	CPU Occupancy (%)
NTSC -60 Hz	30.03 fps	1.38
PAL -50 Hz	25.14 fps	0.55

2.3.3 DM6467 Video Data Conversion Engine Driver

The Video Data Conversion Engine Driver supports the following features:

- Video Conversion modes:
  - Precodec-mode
    - Resizing (optional)
    - Chroma conversion from YUV 422 to YUV 420
  - Postcodec-mode
    - Resizing (optional)
    - Chroma conversion from YUV 420 to YUV 422
    - Blending (optional)
    - Range-mapping-mode (optional)
  - Transcodec mode
    - Resizing (optional)
    - Chroma conversion (optional)
    - Blending (optional)
    - Range-mapping-mode (optional)
  - Edge Padding mode
- Interlaced and progressive formats
- Can be loaded dynamically or statically built into the Linux kernel.
- Processing of 16 channels simultaneously
- Blocking mode. Supports both mmap'ed and user pointer exchange mechanisms.
- Driver is compliant with the UDEV model.

2.3.3.1 Support and Constraints

Features Not Supported None

Constraints

- Sub-Picture mode is not supported due to hardware limitation.
- Horizontal and Vertical Phase Values for resizer are considered as zero.
- Simultaneous use of resizing and blending in post-codec or transcodec modes is restricted.
- The address of the buffer should be 8-byte aligned and the buffer size should be a multiple of 32 bytes.
- Pitch should be 8-byte aligned.
- In edge padding mode, the same buffer is used for input and output.

Supported System Calls open(), close(), ioctl(), mmap() and mumap()

Supported IOCTLs The Video Data Conversion Engine driver supports the IOCTLs shown in Table 2-26

Table 2-26. DM6467 VDCE IOCTLs

Constant	Description
VDCE_SET_PARAMS , VDCE_GET_PARAMS	Sets and gets the GE parameters
VDCE_START	Starts the VDCE processing
VDCE_REQBUF	Request to allocate buffers
VDCE_QUERYBUF	Queries the status of a buffer
VDCE_GET_DEFAULT	Gets the default parameters associated with VDCE processing mode.

2.4 DM365 Video Drivers

The DaVinci video driver is a set of video display, capture, statistics and processing components. Figure 2-4 shows the DM365 architecture.

The video display driver for the DaVinci video processing back end (VPBE) peripheral is a *char* driver, compliant with the frame buffer driver (FBDev) and V4L2 framework. The device nodes created for the FBDev VPBE driver are:

- /dev/fb/0 - OSD0 Window
- /dev/fb/1 or /dev/video1 - VID0 Window
- /dev/fb/2 - OSD1 Window
- /dev/fb/3 or /dev/video3- VID1 Window

The device nodes created for V4L2 Display driver are:

- /dev/video2 - VID0 Window
- /dev/video3 - VID1 Window

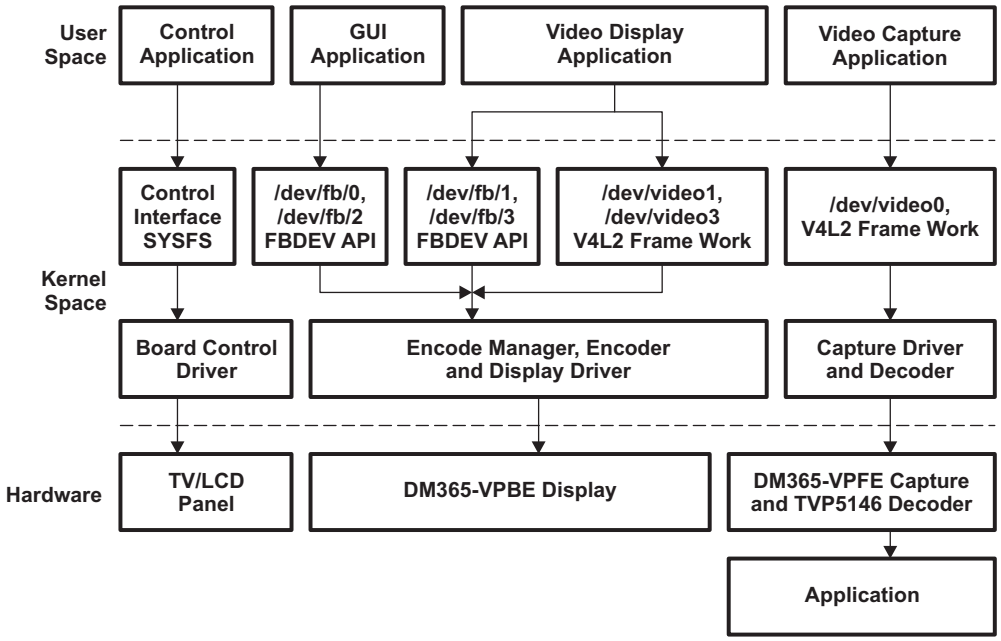

Figure 2-4. DM365 Video Driver Architecture

The video capture driver for the video processing front end (VPFE) peripheral is a *char* driver, compliant with the V4L2 specification. The device node created for the VPFE driver is /dev/video0.

The previewer, resizer, H3A (auto focus (AF), and auto expose/auto white balance (AEW)) drivers are front-end components that provide post-capture processing.

2.4.1 DM365 Video Display Driver

The VPBE display driver supports the following features:

- Two video windows (VID0 and VID1) in YUV422 format (UYVY). Both windows are triple buffered.
- OSD or bitmap window using OSD0 in RGB 565 format with double buffering
- One video window in YUV 420 Semi planar format (NV12)
- Buffer exchange mechanism supports both blocking and non-blocking modes. These supports mmap'ed exchange mechanisms.
- Can be loaded dynamically or built statically into the Linux kernel.
- Attribute window acts as a per-pixel alpha plane using the hardware plane OSD1. This plane is double-buffered.
- Analog outputs using the four internal DACs
  - Composite video using one DAC
  - S-Video using two DACs
  - Component video using three DACs
- NTSC and PAL interlaced outputs using the internal video encoder
- Flexibility to change window sizes and starting coordinates
- Zoom feature to zoom the display to 1x, 2x and 4x
- 480p and 576p formats
- Digital output on RGB666 digital interface
- RGB888 input mode for the video window one
- Palletized input (1/2/4/8 bit) mode for OSD window
- Run time enable/disable of windows
- Color keying
- Programmed buffer flipping
- OSD0 and OSD1 windows are controlled only by FBDEV interface
- VID0 and VID1 windows can be controlled by both V4L2 and FBDEV interface
- Output and mode are set using sysfs interface
- ED display supports the following resolutions:
  - 480p at 60 fps
  - 576p at 50 fps
- HD display supports following resolutions:
  - 720p at 60 fps
  - 1080i at 30 fps

2.4.1.1 Support and Constraints

Features Not Supported	• Analog RBG output • OSD windows are not supported by the V4L2 display driver
Constraints	• Dynamic removal of the module is not allowed when the FB console is in use. • Only VID0 will be usable in HD video modes (720p and 1080i).
Supported System Calls	open(), close(), ioctl(), mmap() and munmap()

2.4.1.2 Supported IOCTLs

The video display driver supports IOCTLs shown in Table 2-27.

Table 2-27. DM365 VPBE FBDEV IOCTLs Supported

Constant	Description
FBIO_GET_VIDEO_CONFIG_PARAMS	Gets the current video window configuration
FBIO_SET_VIDEO_CONFIG_PARAMS	Sets the configuration of the video window
FBIO_GET_BITMAP_CONFIG_PARAMS	Gets the existing configuration of the bitmap (OSD0 / OSD1) window
FBIO_SET_BITMAP_CONFIG_PARAMS	Sets the configuration of the BITMAP (OSD0 / OSD1) window
FBIO_SET_BACKG_COLOR	Sets the background color. The window should be disabled before using this command.
FBIOBLANK	Blank the display
FBIOGET_VSCREENINFO	Gets the variable screen information of the framebuffer. This command can be used for each framebuffer window.
FBIOPUT_VSCREENINFO	Sets variable screen parameters for framebuffer, including the window input format (resolution and bits per pixel)
FBIOGET_FSCREENINFO	Gets the fixed screen information of the framebuffer. This command can be used for each framebuffer window.
FBIOPUT_FSCREENINFO	Sets fixed screen parameters for framebuffer, including the window input format (resolution and bits per pixel)
FBIOPUTCMAP	Sets the pseudo palette
FBIOPAN_DISPLAY	Sets the display buffer for the window using the var_screeninfo offset of the buffer (out of number of buffers for the window: 3 for video and 2 for OSD) passed to the IOCTL. It calculates actual buffer location and sets it into the window register.
FBIO_SET_CURSOR	Configures cursor parameters
FBIO_WAITFORVSYNC	Synchronizes the buffered display by waiting for vsync before the call returns
FBIO_SETPOS	Changes the starting X and Y coordinate of the desired video or OSD plane
FBIO_SETPOSX	Changes the starting X coordinate of the desired video or OSD plane
FBIO_SETPOSY	Changes the starting Y coordinate of the desired video or OSD plane
FBIO_SETZOOM	Specifies the ZOOM parameters (identity, 2x or 4x) for the corresponding display plane

For more details refer http://www.linux-fbdev.org.

2.4.1.3 Performance and Benchmarks

The performance numbers were captured using the following:

- Sony Bravia KLV-S26A10 TV was used for Display testing
- SD Display benchmarks were captured using composite input/output video loop back
- ED Display was tested using composite input/component output video loop back

2.4.1.3.1 VPBE-FBDEV Low Latency Desktop Preemption

Table 2-28. DM365 VPBE-FBDEV Performance – LLD

Video Resolution	Frame Rate (Frames Per Second)	CPU occupancy (%)
480P-60	60.97 fps	0.72
576P-50	50.62	0
NTSC -60 Hz	30.08 fps	0.42
PAL -50 Hz	25.1 fps	0.1

2.4.1.3.2 VPBE-FBDEV Real Time Preemption

Table 2-29. DM365 VPBE-FBDEV Performance – RT

Video Resolution	Frame Rate (Frames Per Second)	CPU occupancy (%)
480P-60	64.02	0.96
576P-50	51.26	0.2
NTSC -60 Hz	30.09 fps	0.78
PAL -50 Hz	25.1 fps	0

2.4.2 DM365 Video Capture Driver

The video capture driver supports the following features:

- Decoded video input in YUV422 format (UYVY or YUYV) with the help of an external decoder
- Can be loaded dynamically or built statically into the Linux kernel
- Output video in YUV 420 semi planar format using previewer and resizer in Continuous/On the fly mode
- Buffer exchange mechanism supports blocking mode. This supports mmap'ed exchange mechanism.
- NTSC and PAL video input through an external decoder, including auto sensing capability
- Multi-buffered input with a minimum of three buffers, and support for more at run time
- Cropped input image with programmable start coordinates. This feature can be used to input an image at smaller sizes (e.g. QCIF). For example, the coordinates can be programmed to pick the image from the center of the capture scene
- Bayer pattern input, fault pixel correction, progressive input, LPF and time stamping
- The VPFE driver supports several IOCTLs. These IOCTLs are explained in the V4L2 specification.

2.4.2.1 Support and Constraints

Features Not Supported None

Constraints In Continuous/On the fly mode capture mode through previewer and resizer, the driver uses bottom field and scale it up vertically by 2x to get de-interlaced video. The interlace data cannot be captured in this mode.

Supported System Calls open(), close(), ioctl(), mmap() and munmap()

Supported IOCTLs The video capture driver supports IOCTLs shown in Table 2-30.

Table 2-30. DM365 VPFE-V4L2 IOCTLs

Constant	Description
VIDIOC_CROPCAP	Information about the video cropping and scaling abilities
VIDIOC_ENUMINPUT	Enumerates video inputs
VIDIOC_ENUMOUTPUT	Enumerates video outputs
VIDIOC_ENUMSTD	Enumerates supported video standards
VIDIOC_G_CROP, VIDIOC_S_CROP	Gets or sets the current cropping rectangle
VIDIOC_G_CTRL, VIDIOC_S_CTRL	Gets or sets the value of a control
VIDIOC_G_FMT, VIDIOC_S_FMT, VIDIOC_TRY_FMT	Gets or sets the data format, try a format
VIDIOC_G_INPUT, VIDIOC_S_INPUT	Queries or selects the current video input
VIDIOC_G_PARM, VIDIOC_S_PARM	Gets or sets streaming parameters
VIDIOC_G_STD, VIDIOC_S_STD	Queries or selects the video standard of the current input
VIDIOC_QBUF, VIDIOC_DQBUF	Exchanges a buffer with the driver
VIDIOC_QUERYBUF	Queries the status of a buffer
VIDIOC_QUERYCAP	Queries device capabilities
VIDIOC_QUERYCTRL, VIDIOC_QUERYMENU	Enumerates controls and menu control items
VIDIOC_QUERYSTD	Senses the video standard received by the current input

Table 2-30. DM365 VPFE-V4L2 IOCTLs (continued)

Constant	Description
VIDIOC_REQBUFS	Initiates memory mapping or user pointer I/O
VIDIOC_STREAMON, VIDIOC_STREAMOFF	Starts or stops streaming I/O
VPFE_CMD_S_MT9T001_PARAMS	Configures mt9t001 parameters
VPFE_CMD_G_MT9T001_PARAMS	Gets mt9t001 parameters
VPFE_CMD_S_CCDC_PARAMS	Sets CCDC parameters for Raw Bayer or YCbCr capture
VPFE_CMD_G_CCDC_PARAMS	Gets CCDC parameters for Raw Bayer or YCbCr capture

For more details refer to http://v4l2spec.bytesex.org/v4l2spec/.

2.4.2.2 Performance and Benchmarks

The performance numbers were captured using the following:

- Sony DVPNS-61P DVD player was used for capture testing
- SD Display benchmarks were captured using composite input/output video loop back

2.4.2.2.1 VPFE-V4L2 Low Latency Desktop Preemption

Table 2-31. DM365 VPFE-V4L2 Performance Values – LLD

Video Resolution	Frame Rate (Frames Per Second)	CPU Occupancy (%)
NTSC -60 Hz	25.05 fps	0
PAL -50 Hz	30.03 fps	0.3

2.4.2.2.2 VPFE-V4L2 Real Time Preemption

Table 2-32. DM365 VPFE-V4L2 Performance Values – RT

Video Resolution	Frame Rate (Frames Per Second)	CPU Occupancy (%)
NTSC -60 Hz	25.05 fps	7.13
PAL -50 Hz	30.03 fps	1.01

2.4.3 DM365 Previewer Driver

The Previewer Driver supports the following features:

- Input image in Bayer pattern
- Various hardware configurations such as CFA interpolation, DFC subtract, and so on.
- Accepts input from ISIF or SDRAM
- Converts Bayer pattern input image to YCbCr 4:2:2 format or YUV 420 semi planar format
- Can be loaded dynamically or built statically into the Linux kernel
- Buffer exchange mechanism supports blocking mode. These supports mmap'ed and user pointer buffer exchange mechanism.

2.4.3.1 Support and Constraints

Features Not Supported	None
Constraints	None
Supported System Calls	open(), close(), ioctl(), mmap() and munmap()
Supported IOCTLs	The Previewer Driver supports the IOCTLs shown in Table 2-33

Table 2-33. DM365 Preview Engine IOCTLs

Constant	Description
PREV_ENUM_CAP	Enumerates the preview capabilities available to tune the image
PREV_REQBUF	Requests either kernel-allocated buffer or user-allocated buffers to do I/O with previewer
PREV_QUERYBUF	Queries buffer address for kernel allocated buffers
PREV_S_CONFIG,	Sets the input size, pixel format, and data source etc at the previewer
PREV_G_CONFIG	Gets the input size, pixel format, and data source etc at the previewer
PREV_S_OPER_MODE	Sets the driver operation mode (“Continuous” or Single Shot mode)
PREV_G_OPER_MODE	Gets the driver operation mode (“Continuous” or Single Shot mode)
PREV_S_PARAM	Sets a module parameter in the previewer to tune image
PREV_G_PARAM	Gets a module parameter in the previewer
PREV_S_CONTROL	Sets a control on a module. This is similar to PREV_S_PARAM except that it can be issued while streaming is on.
PREV_G_CONTROL	Gets a control parameter
PREV_PREVIEW	Starts preview in Single shot mode

2.4.4 DM365 Resizer Driver

The Resizer Driver supports the following features:

- UYVY data input format
- UYVY or YUV 420 Semi planar output format
- Input from CCDC, SDRAM or DDRAM
- Upscaling/downscaling multiplier from 1/16x to 4x
- Can be loaded dynamically or built statically into the Linux kernel.
- Buffer exchange mechanism supports blocking mode. This supports mmap’ed and user pointer buffer exchange mechanism.
- Can be chained with previewer device to do preview and resize

2.4.4.1 Support and Constraints

Features Not Supported Only one channel supported

Constraints None

Supported System Calls open(), close(), ioctl(), mmap() and mumap()

Supported IOCTLs The resizer driver supports the IOCTLs shown in Table 2-34

Table 2-34. DM365 Resizer Driver Supported IOCTLs

Constant	Description
RSZ_REQBUF	Requests either kernel-allocated buffer or user-allocated buffers to do I/O with resizer
RSZ_QUERYBUF	Queries buffer address for kernel-allocated buffers
RSZ_S_CONFIG	Sets the input size, pixel format, and data source etc at the resizer
RSZ_G_CONFIG	Gets the input size, pixel format, and data source etc at the resizer
RSZ_S_OPER_MODE	Sets the driver operation mode ("Continuous" or Single Shot mode)
RSZ_G_OPER_MODE	Gets the driver operation mode ("Continuous" or Single Shot mode)
RSZ_RESIZE	Starts resize in single shot mode

2.4.5 DM365 Face Dectection Driver

The Face Detection Driver supports the following features:

- A loadable module
- Standalone kernel module so that it can be used by multiple applications
- Supports Y-only (luma) input data of 320x240 resolution
- Supports configurable parameters like minimum face size for detection, direction of detection

2.4.5.1 Support and Constraints

Features Not Supported	None
Constraints	The driver needs a work area buffer of size 13200x4 bytes.
Supported System Calls	open(), close(), ioctl(), mmap() and mumap()
Supported IOCTLs	The face detection driver supports the IOCTLs shown in Table 2-35

Table 2-35. DM365 Face Detection IOCTLs

Constant	Description
FACE_DETECT_SET_HW_PARAM	Initializes the Face Detection hardware engine
FACE_DETECT_GET_HW_PARAM	Gets the current hardware parameters for face detection hardware engine
FACE_DETECT_EXECUTE	Enables the Face Detection hardware engine
FACE_DETECT_SET_BUFFER	Initializes the input frame address and the work area buffer address for the face detection hardware.

2.4.6 DM365 Auto Expose/Auto White Balance (AEW) Driver

The Auto Expose/Auto White Balance (AEW) Driver supports the following features:

- Image in Bayer pattern
- Input from CCD controller
- Can be loaded dynamically or built statically into the Linux kernel.

2.4.6.1 Support and Constraints

Features Not Supported	None
Constraints	None
Supported System Calls	open(), close(), ioctl(), mmap() and mumap()
Supported IOCTLs	The AEW driver supports the IOCTLs shown in Table 2-36

Table 2-36. DM365 AEW IOCTLs

Constant	Description
AEW_S_PARAM	Sets the following parameters/modules of the AEW hardware: • Window parameters • Black window parameters • Saturation check module • A-Law compression module
AEW_G_PARAM	Gets the AEW hardware settings
AEW_ENABLE	Enables AEW Engine with parameters set by AEW_S_PARAM
AEW_DISABLE	Disables AEW Engine

2.4.7 DM365 AF Driver

The AF Driver supports the following features:

- Image in Bayer pattern
- Input from CCD controller
- Can be loaded dynamically or built statically into the Linux kernel.

2.4.7.1 Support and Constraints

Features Not Supported	None
Constraints	None
Supported System Calls	open(), close(), ioctl(), mmap() and mumap()
Supported IOCTLs	The AF driver supports the IOCTLs shown in Table 2-37

Table 2-37. DM365 AF IOCTLs

Constant	Description
AF_S_PARAM	Sets the following parameters/modules of the AF hardware: • Paxel parameters • IIR Filter parameters • HMF Filter Parameter • A-Law compression module
AF_G_PARAM	Gets the AF hardware settings
AF_ENABLE	Enables AF Engine with parameters set by AF_S_PARAM
AF_DISABLE	Disables AF Engine

2.5 Audio Driver

The DaVinci audio driver is a char driver that supports the multi-channel audio serial port (McASP) for DM6467, or multi-channel buffer serial port (McBSP) for DM644x, DM355, and DM365. It is compatible with advanced Linux sound architecture (ALSA) and open sound system (OSS) simulator interfaces (deprecated). The ALSA driver can be accessed from user space as plughw:0,0. The OSS driver can be accessed from user space as /dev/dsp and /dev/mixer. Though this kernel version (2.6.18) supports OSS through an emulation layer, it is expected that all OSS-based user applications are converted to use the ALSA libraries.

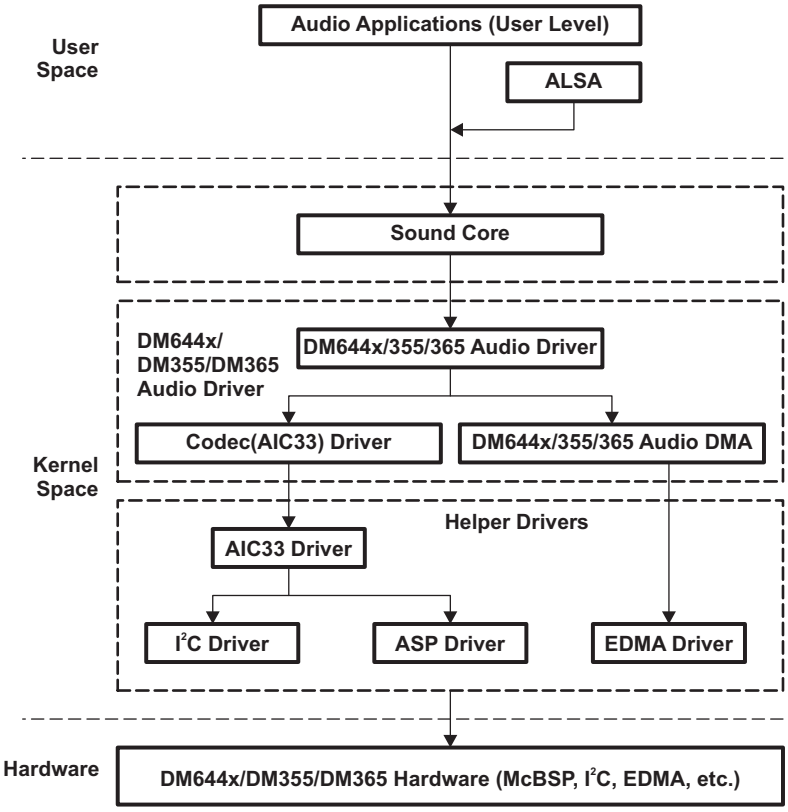

Figure 2-5. DM644x, DM355 and DM365 Audio Driver Architecture

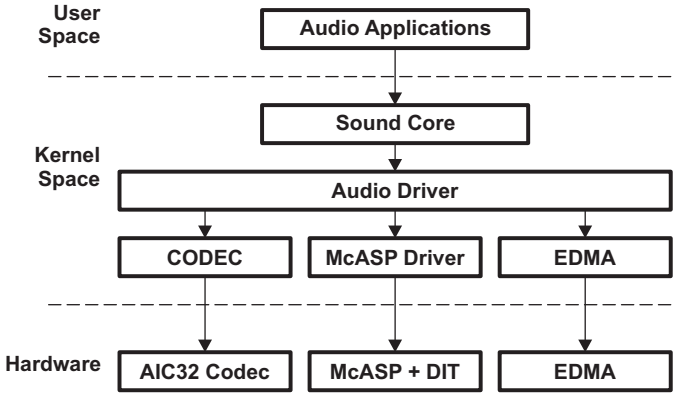

Figure 2-6. DM6467 Audio Driver Architecture

The DaVinci audio driver supports the following features:

- Record and playback capabilities
- Audio channel set to stereo mode by default
- Blocking mode
- 34 sample rates (from 7350 - 96000 Hz)
- DSP, I2S, and left/right justified formats
- Mixer interface
- Plays zeros using a loop job when no data is supplied by the application
- Supports separate volume settings for left and right channels
- DM6467:
  - Non-blocking mode
  - McASP works in slave mode as well as master mode. In the default configuration, AIC32 is master and McASP is slave.
  - McASP in DIT mode
  - S/PDIF supports only 44100Hz and 48000Hz sampling rates
- DM644x, DM355 and DM365:
  - McBSP works in slave mode as well as master mode. In the default configuration, AIC33 is master and McBSP is slave.

2.5.1 Support and Constraints

Features Not Supported

The audio driver does not support the following features:

- Non-Blocking mode (DM644x and DM355)
- Does not support DSP, right-justified, left-justified features (DM644x, DM355 and DM365)

Constraints

Supports AFMT_S16_LE (16 bit) audio format only.

Supported System Calls

open(), close(), read(), write(), ioctl(), select()

Supported OSS IOCTLs (deprecated)

Table 2-38 shows the ALSA IOCTLs supported

Table 2-38. Audio Driver Supported ALSA IOCTLs

Constant	Description
OSS_GETVERSION	Identifies the driver version
SNDCTL_DSP_STEREO	Sets stereo or mono channel
SNDCTL_DSP_CHANNELS	Sets the number of audio channels
SNDCTL_DSP_SPEED	Sets the sampling rate
SNDCTL_DSP_SETFMT	Selects the sample format
SNDCTL_DSP_GETFMTS	Returns a list of natively supported sample formats
SNDCTL_DSP_GETBLKSIZE	Returns the block size that the sound driver uses for data transfers
SNDCTL_DSP_GETCAPS	Returns a bitmask identifying the capabilities of a sound card DSP device.
SNDCTL_DSP_SETFRAGMENT	Requests a fragment size and number of fragments
SNDCTL_DSP_SYNC	Application waits until the last byte written to the device has been played
SNDCTL_DSP_SETDUPLEX	Turns ON the duplex mode
SNDCTL_DSP_POST	Notifies the driver that there may be a pause in the output
SNDCTL_DSP_GETTRIGGER	Returns the current trigger bits
SNDCTL_DSP_SETTRIGGER	Starts audio recording and/or playback in sync
SNDCTL_DSP_GETOPTR	Returns the current playback pointer
SNDCTL_DSP_GETIPTR	Returns the current recording pointer

Table 2-38. Audio Driver Supported ALSA IOCTLs (continued)

Constant	Description
SNDCTL_DSP_GETOSPACE	Returns the number of currently recorded bytes
SNDCTL_DSP_GETISPACE	Returns the number of currently recorded bytes
SNDCTL_DSP_NONBLOCK	Sets non-blocking mode
SNDCTL_DSP_RESET	Resets the device
SOUND_MIXER_VOLUME	Indicates master output level
SOUND_MIXER_LINE	Indicates line input
SOUND_MIXER_MIC	Indicates microphone input
SOUND_MIXER_RECSRC	Indicates the channels that are currently selected as the recording source
SOUND_MIXER_IGAIN	Indicates input gain level
SOUND_MIXER_OGAIN	Indicates output gain level
SOUND_MIXER_RECMASK	Returns a bitmask indicating the channels that can be used as a recording source
SOUND_MIXER_DEVMASK	Returns a bitmask indicating the channels are supported by the mixer
SOUND_MIXER_CAPS	Returns a bitmask indicating sound card capability
SOUND_MIXER_STEREODEVS	When set, indicates that a channel supports stereo mode. If cleared, it supports only one channel (mono).
SOUND_PCM_READ_RATE	Reads the sampling frequency
SOUND_PCM_READ_BITS	Returns a list of natively supported sample formats
SOUND_PCM_READ_CHANNELS	Returns the current number of read channels

Table 2-39. DM644x/DM355/DM365 Specific Audio IOCTLs

Constant	Description
SOUND_MIXER_INFO	Returns Mixer information

Table 2-40. DM6467 Specific Audio IOCTLs

Constant	Description
SOUND_MIXER_MICBIAS	Sets the microphone bias output voltage
SOUND_MIXER_BASS	Indicates bass tone control
SOUND_MIXER_TREBLE	Indicates treble tone control

The Advanced Linux Sound Architecture comes with a Kernel API and a library API. Application programmers should use the library API rather than the kernel API. The library offers 100% of the functionality of the kernel API, but adds major improvements in usability, making the application code simpler and better looking. The online-documentation for the same is available at: http://www.alsa-project.org.

2.5.2 DM644x Performance and Benchmarks

2.5.2.1 Audio ALSA Read and Write Performance Low Latency Desktop Preemption

The performance numbers were captured using the following:

- ARM / DSP frequency = 270 MHz
- DDR frequency = 162 MHz
- Application buffer size = 4096 bytes

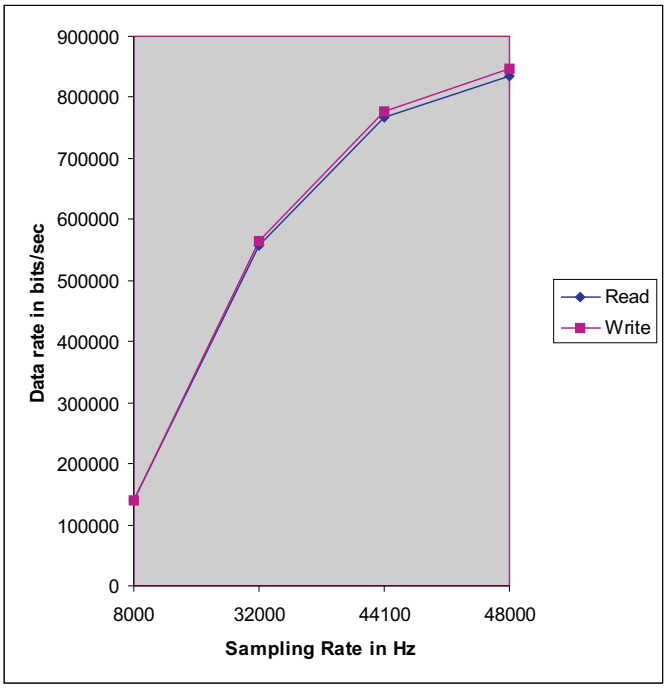

Figure 2-7. DM644x Audio ALSA Write and Read Performance – LLD

Table 2-41. DM644x Audio ALSA Read Performance – LLD

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of Bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	139215	5120	301.28	0.14
32000	16	2	556793	5120	75.33	0.35
44100	16	2	767292	5120	54.66	0.46
48000	16	2	835127	5120	50.22	0.44

Table 2-42. DM644x Audio ALSA Write Performance – LLD

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of Bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	140976	5120	297.52	0.15
32000	16	2	563901	5120	74.38	0.42
44100	16	2	777124	5120	53.97	0.56
48000	16	2	845850	5120	49.59	0.56

2.5.2.2 Audio ALSA Read and Write Performance Real Time Preemption

The performance numbers were captured using the following:

- ARM / DSP frequency = 297 MHz
- DDR frequency = 162 MHz
- Application buffer size = 4096 bytes

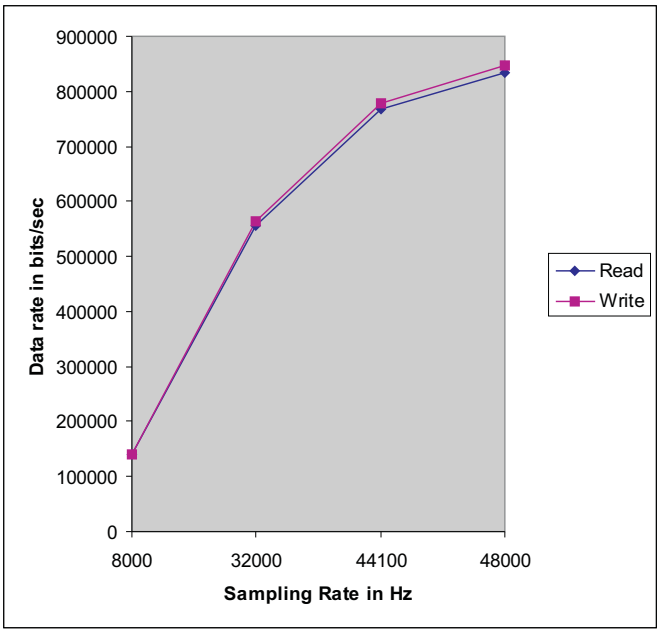

Figure 2-8. DM644x Audio ALSA Write and Read Performance – RT

Table 2-43. DM644x Audio ALSA Read Performance – RT

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	139215	5120	301.28	0.41
32000	16	2	556794	5120	75.33	0.85
44100	16	2	767296	5120	54.66	0.95
48000	16	2	835129	5120	50.22	0.99

Table 2-44. DM644x Audio ALSA Write Performance – RT

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	140976	5120	297.52	0.42
32000	16	2	563901	5120	74.38	0.84
44100	16	2	777123	5120	53.97	1.22
48000	16	2	845850	5120	49.59	1.04

2.5.2.3 Audio OSS Read and Write Performance Low Latency Desktop Preemption

The performance numbers were captured using the following:

- ARM / DSP frequency = 270 MHz
- DDR frequency = 162 MHz
- Application buffer size = 4096 bytes

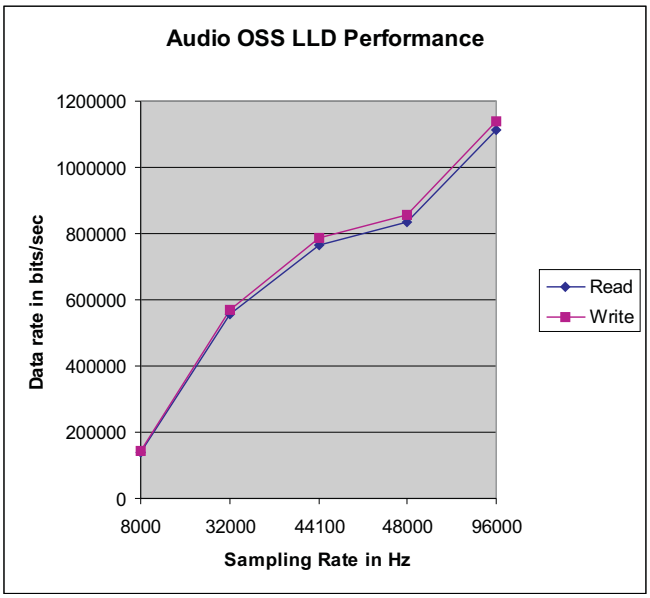

Figure 2-9. DM644x Audio OSS Write and Read Performance – LLD

Table 2-45. DM644x Audio OSS Read Performance – LLD

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of Bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	139214	5120	301.29	0.14
32000	16	2	556261	5120	75.40	0.31
44100	16	2	766509	5120	54.72	0.33
48000	16	2	834269	5120	50.28	0.50
96000	16	2	1112191	5120	37.71	0.64

Table 2-46. DM644x Audio OSS Write Performance – LLD

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of Bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	142769	5120	293.78	0.14
32000	16	2	570643	5120	73.50	0.29
44100	16	2	786399	5120	53.34	0.36
48000	16	2	855943	5120	49.00	0.41
96000	16	2	11414230	5120	36.75	0.52

2.5.2.4 Audio OSS Read and Write Performance Real Time Preemption

The performance numbers were captured using the following:

- ARM / DSP frequency = 270 MHz
- DDR frequency = 162 MHz
- Application buffer size = 4096 bytes

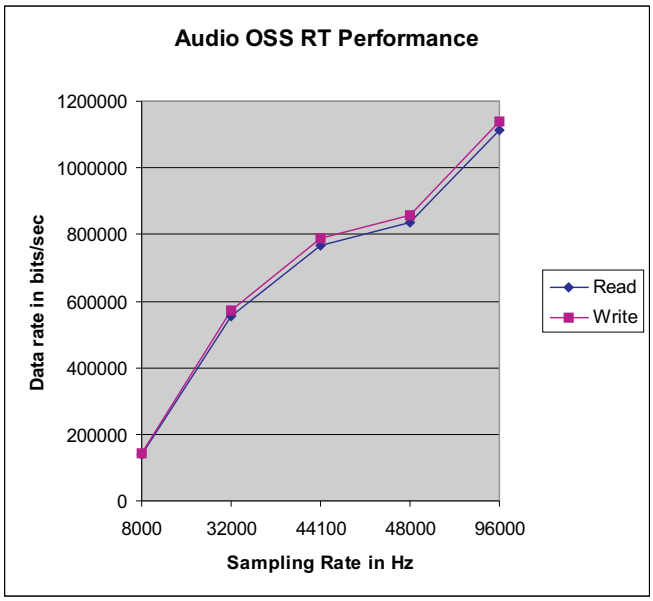

Figure 2-10. DM644x Audio OSS Write and Read Performance – RT

Table 2-47. DM644x Audio OSS Read Performance – RT

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	139214	5120	301.29	0.38
32000	16	2	556259	5120	75.40	0.73
44100	16	2	766508	5120	54.72	0.84
48000	16	2	834265	5120	50.28	0.79
96000	16	2	1112189	5120	37.71	1.06

Table 2-48. DM644x Audio OSS Write Performance – RT

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	142769	5120	293.78	0.44
32000	16	2	570642	5120	73.50	0.60
44100	16	2	786400	5120	53.34	0.84
48000	16	2	855943	5120	49.00	1.00
96000	16	2	1141228	5120	36.75	1.00

2.5.3 DM355 Performance and Benchmarks

2.5.3.1 Audio ALSA Read and Write Performance Low Latency Desktop Preemption

The performance numbers were captured using the following:

- ARM / DSP frequency = 270 MHz
- DDR frequency = 162 MHz
- Application buffer size = 4096 bytes

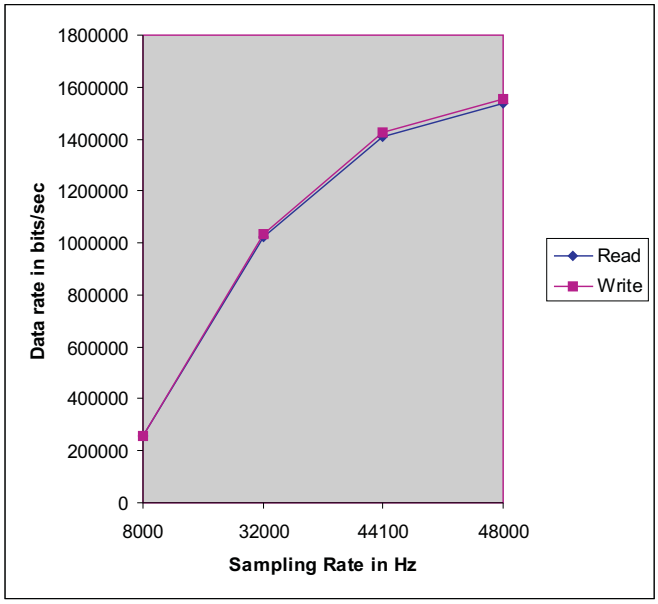

Figure 2-11. DM355 Audio Write and Read Performance – LLD

Table 2-49. DM355 Audio Read Performance – LLD

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	255798	5120	163.97	0.1
32000	16	2	1023121	5120	41.00	0.17
44100	16	2	1409887	5120	29.75	1.58
48000	16	2	1534629	5120	27.33	0.18

Table 2-50. DM355 Audio Write Performance – LLD

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	259034	5120	161.92	0.96
32000	16	2	1036125	5120	40.48	0.27
44100	16	2	1427891	5120	29.37	1.77
48000	16	2	1554177	5120	26.99	0.3

2.5.3.2 Audio Read and Write Performance Real Time Preemption

The performance numbers were captured using the following:

- ARM / DSP frequency = 270 MHz
- DDR frequency = 162 MHz
- Application buffer size = 4096 bytes

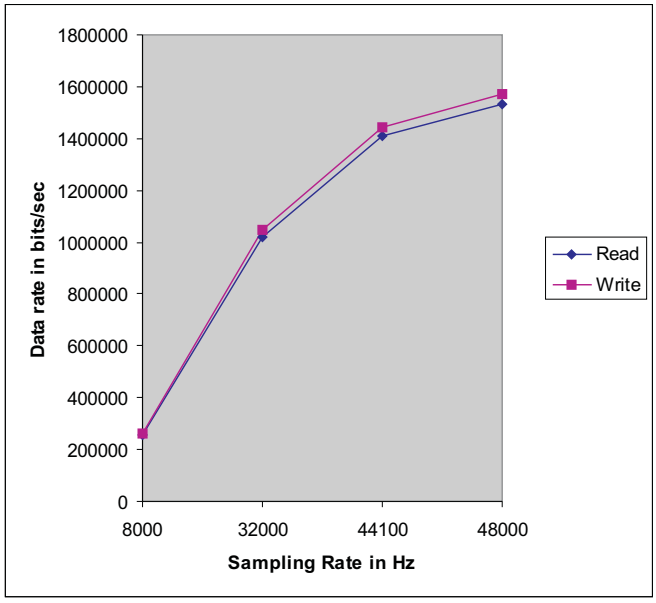

Figure 2-12. DM355 Audio Write and Read Performance – RT

Table 2-51. DM355 Audio Read Performance – RT

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	255798	5120	163.97	0.39
32000	16	2	1021943	5120	41.04	0.61
44100	16	2	1408107	5120	29.79	2
48000	16	2	1532579	5120	27.37	2.11

Table 2-52. DM355 Audio Write Performance – RT

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	262300	5120	159.90	0.83
32000	16	2	1048432	5120	40.01	3.72
44100	16	2	1444772	5120	29.03	1.54
48000	16	2	1572536	5120	26.67	1.27

2.5.3.3 Audio OSS Read and Write Performance Low Latency Desktop Preemption

The performance numbers were captured using the following:

- ARM / DSP frequency = 270 MHz
- DDR frequency = 162 MHz
- Application buffer size = 4096 bytes

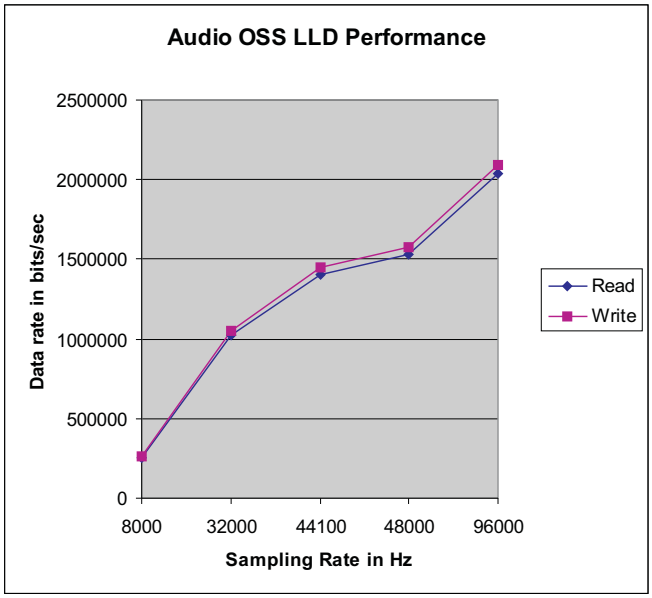

Figure 2-13. DM355 Audio OSS Write and Read Performance – LLD

Table 2-53. DM355 Audio OSS Read Performance – LLD

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	255789	5120	163.97	0.13
32000	16	2	1021751	5120	41.05	0.29
44100	16	2	1407734	5120	29.79	1.68
48000	16	2	1532141	5120	27.38	1.13
96000	16	2	2042184	5120	50.54	4.82

Table 2-54. DM355 Audio OSS Write Performance – LLD

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	262298	5120	159.91	0.23
32000	16	2	1048416	5120	40.01	2.72
44100	16	2	1444753	5120	29.03	1.34
48000	16	2	1572505	5120	26.67	1.8
96000	16	2	2096507	5120	20.01	5.9

2.5.3.4 Audio OSS Read and Write Performance Real Time Preemption

The performance numbers were captured using the following:

- ARM / DSP frequency = 270 MHz
- DDR frequency = 162 MHz
- Application buffer size = 4096 bytes

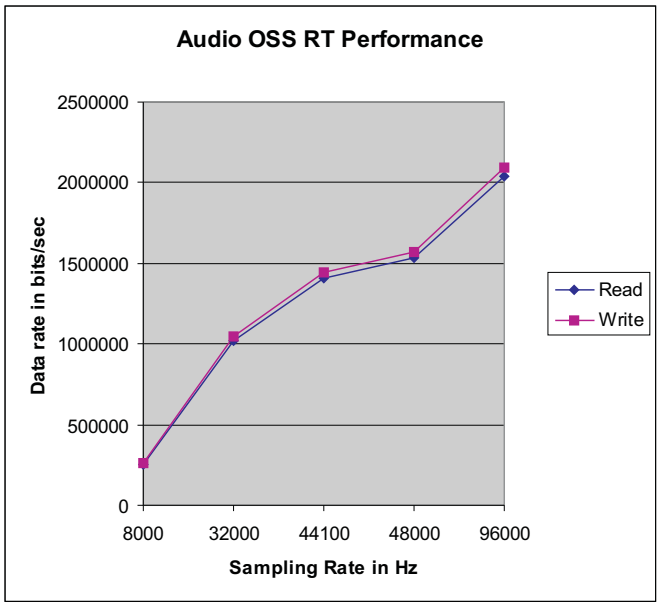

Figure 2-14. DM355 Audio OSS Write and Read Performance – RT

Table 2-55. DM355 Audio OSS Read Performance – RT

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	255790	5120	169.97	1.26
32000	16	2	1021736	5120	41.05	0.56
44100	16	2	1407739	5120	29.79	2.5
48000	16	2	1532097	5120	27.38	1.97
96000	16	2	2042184	5120	20.54	7.12

Table 2-56. DM355 Audio OSS Write Performance – RT

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	262297	5120	159.91	1.22
32000	16	2	1048405	5120	40.01	0.57
44100	16	2	1444729	5120	29.03	2.26
48000	16	2	1572486	5120	26.67	2.17
96000	16	2	2096473	5120	20.01	1.49

2.5.4 DM6467 Performance and Benchmarks

2.5.4.1 Audio ALSA Read and Write Performance Low Latency Desktop Preemption

The performance numbers were captured using the following:

- ARM / DSP frequency = 297 MHz
- DDR frequency = 297 MHz
- Application buffer size = 4096 bytes

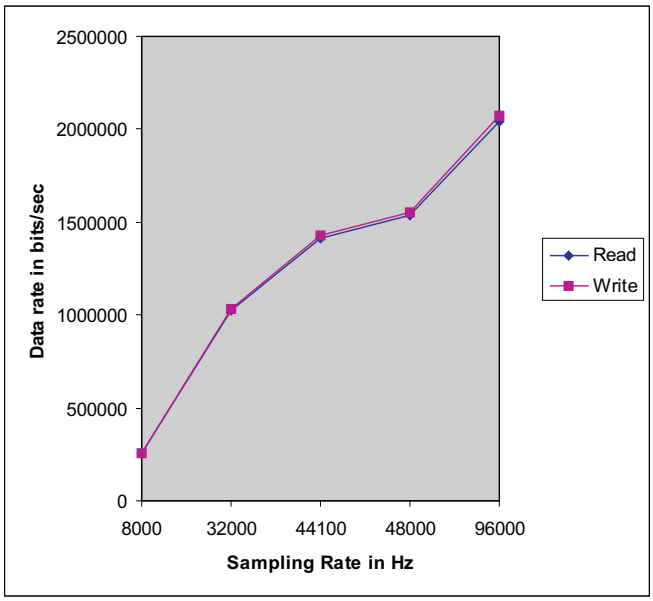

Figure 2-15. DM6467 Audio ALSA Write and Read Performance – LLD

Table 2-57. DM6467 Audio ALSA Read Performance – LLD

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	255780	5120	163.98	0.13
32000	16	2	1022807	5120	41.01	0.27
44100	16	2	1409357	5120	29.76	0.57
48000	16	2	1533982	5120	27.34	0.73
96000	16	2	2044938	5120	20.51	1.02

Table 2-58. DM6467 Audio ALSA Write Performance – LLD

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	259015	5120	161.93	0.1
32000	16	2	1036056	5120	40.48	0.47
44100	16	2	1427801	5120	29.38	0.58
48000	16	2	1554082	5120	26.99	0.59
96000	16	2	2072103	5120	20.24	1.04

2.5.4.2 Audio ALSA Read and Write Performance Real Time Preemption

The performance numbers were captured using the following

- ARM frequency = 297 MHz
- DDR frequency = 297 MHz
- Application buffer size = 4096 bytes

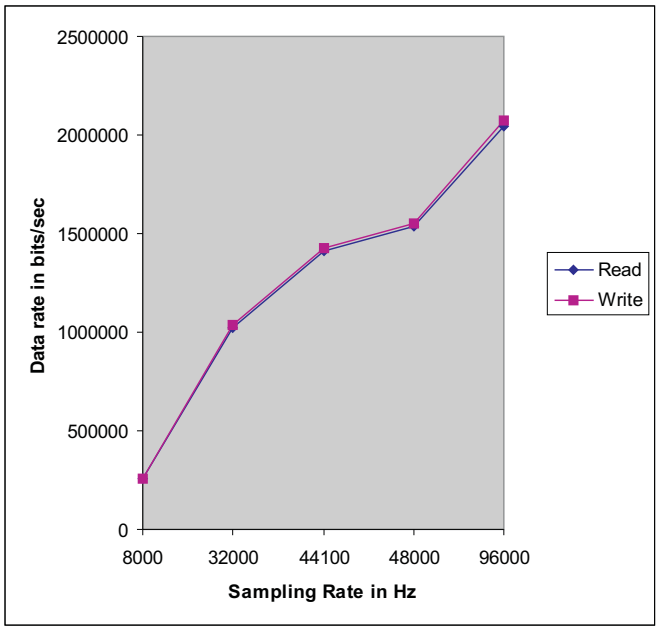

Figure 2-16. DM6467 Audio ALSA Write and Read Performance – RT

Table 2-59. DM6467 Audio ALSA Read Performance – RT

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	255779	5120	163.98	0.18
32000	16	2	1022815	5120	41.01	0.78
44100	16	2	1409365	5120	29.76	0.81
48000	16	2	1533991	5120	27.34	0.88
96000	16	2	2044953	5120	20.51	1.7

Table 2-60. DM6467 Audio ALSA Write Performance – RT

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	259015	5120	161.93	0.22
32000	16	2	1036056	5120	40.48	0.81
44100	16	2	1427800	5120	29.38	0.95
48000	16	2	1554079	5120	26.99	0.81
96000	16	2	2072100	5120	20.24	1.18

2.5.4.3 Audio OSS Read and Write Performance Low Latency Desktop Preemption

The performance numbers were captured using the following:

- ARM / DSP frequency = 297 MHz
- DDR frequency = 297 MHz
- Application buffer size = 4096 bytes

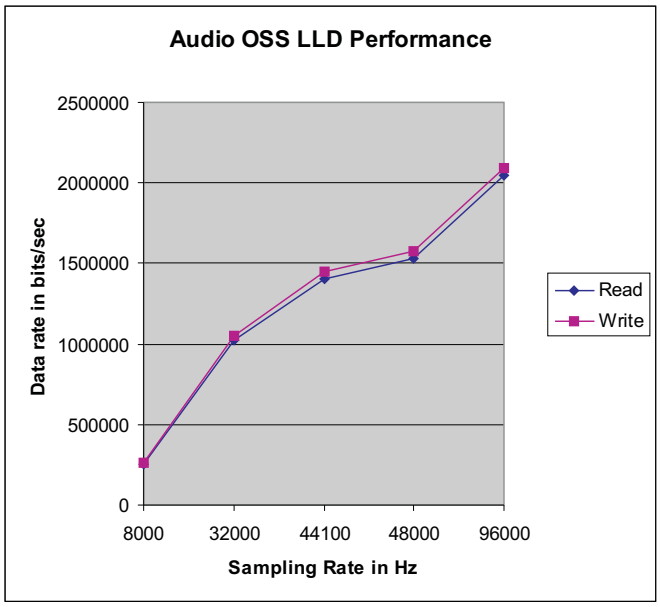

Figure 2-17. DM6467 Audio OSS Write and Read Performance – LLD

Table 2-61. DM6467 Audio OSS Read Performance – LLD

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	255780	5120	163.98	0.13
32000	16	2	1022807	5120	41.01	0.27
44100	16	2	1409357	5120	29.76	0.57
48000	16	2	1533982	5120	27.34	0.73
96000	16	2	2044938	5120	20.51	1.02

Table 2-62. DM6467 Audio OSS Write Performance – LLD

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	259015	5120	161.93	0.1
32000	16	2	1036056	5120	40.48	0.47
44100	16	2	1427801	5120	29.38	0.58
48000	16	2	1554082	5120	26.99	0.59
96000	16	2	2072103	5120	20.24	1.04

2.5.4.4 Audio OSS Read and Write Performance Real Time Preemption

The performance numbers were captured using the following

- ARM frequency = 297 MHz
- DDR frequency = 297 MHz
- Application buffer size = 4096 bytes

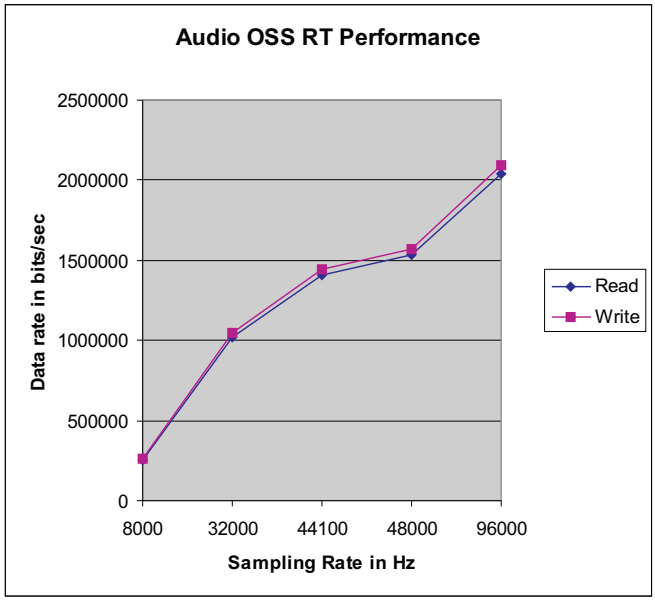

Figure 2-18. DM6467 Audio OSS Write and Read Performance – RT

Table 2-63. DM6467 Audio OSS Read Performance – RT

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	255779	5120	163.98	0.18
32000	16	2	1022815	5120	41.01	0.78
44100	16	2	1409365	5120	29.76	0.81
48000	16	2	1533991	5120	27.34	0.88
96000	16	2	2044953	5120	20.51	1.7

Table 2-64. DM6467 Audio OSS Write Performance – RT

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	259015	5120	161.93	0.22
32000	16	2	1036056	5120	40.48	0.81
44100	16	2	1427800	5120	29.38	0.95
48000	16	2	1554079	5120	26.99	0.81
96000	16	2	2072100	5120	20.24	1.18

2.5.5 DM365 Performance and Benchmarks

2.5.5.1 Audio ALSA Read and Write Performance Low Latency Desktop Preemption

The performance numbers were captured using the following:

- ARM / DSP frequency = 297 MHz
- DDR frequency = 243 MHz
- Application buffer size = 4096 bytes

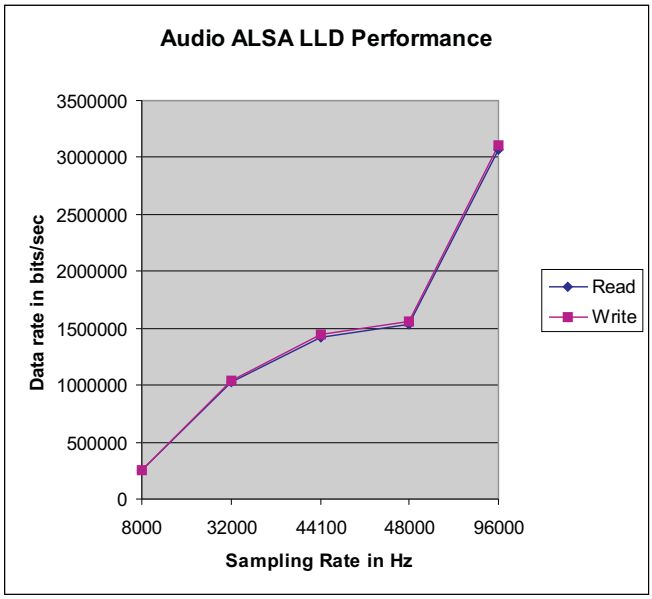

Figure 2-19. DM365 Audio ALSA Read and Write Performance – LLD

Table 2-65. DM365 Audio ALSA Read Performance – LLD

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of Bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	256053	5120	163.81	0.18
32000	16	2	1022741	5120	41.01	0.73
44100	16	2	1422012	5120	29.50	0.71
48000	16	2	1533937	5120	27.34	0.88
96000	16	2	3066595	5120	16.68	1.83

Table 2-66. DM365 Audio ALSA Write Performance – LLD

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of Bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	258980	5120	161.96	0.22
32000	16	2	1035911	5120	40.49	0.59
44100	16	2	1443791	5120	29.06	0.83
48000	16	2	1555736	5120	26.96	0.93
96000	16	2	3111430	5120	13.48	1.78

2.5.5.2 Audio ALSA Read and Write Performance Real Time Preemption

The performance numbers were captured using the following:

- ARM / DSP frequency = 297 MHz
- DDR frequency = 243 MHz
- Application buffer size = 4096 bytes

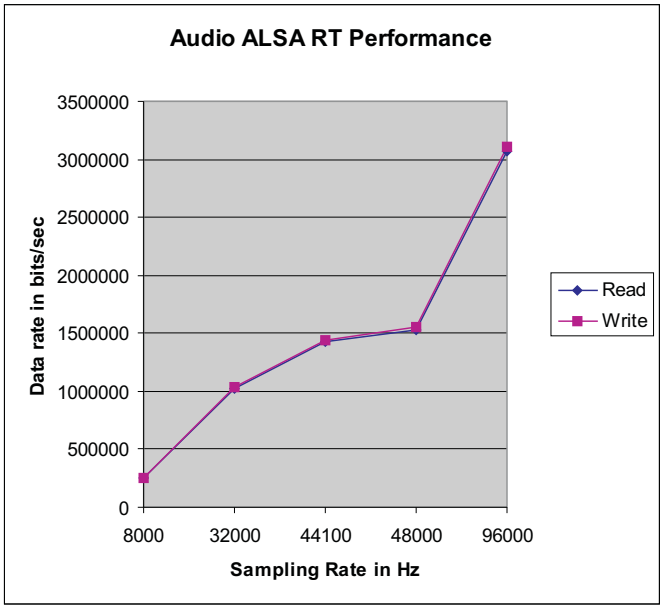

Figure 2-20. DM365 Audio ALSA Write and Read Performance – RT

Table 2-67. DM365 Audio ALSA Read Performance – RT

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	255745	5120	163.00	0.5
32000	16	2	1022750	5120	41.01	1.43
44100	16	2	1422019	5120	29.50	1.82
48000	16	2	1533945	5120	27.34	1.89
96000	16	2	3066654	5120	13.68	2

Table 2-68. DM365 Audio ALSA Write Performance – RT

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	258980	5120	161.96	0.54
32000	16	2	1035910	5120	40.49	1.16
44100	16	2	1434490	5120	29.24	1.67
48000	16	2	1553853	5120	26.99	1.95
96000	16	2	3107672	5120	13.50	3.08

2.5.5.3 Audio OSS Read and Write Performance Low Latency Desktop Preemption

The performance numbers were captured using the following:

- ARM / DSP frequency = 297 MHz
- DDR frequency = 243 MHz
- Application buffer size = 4096 bytes

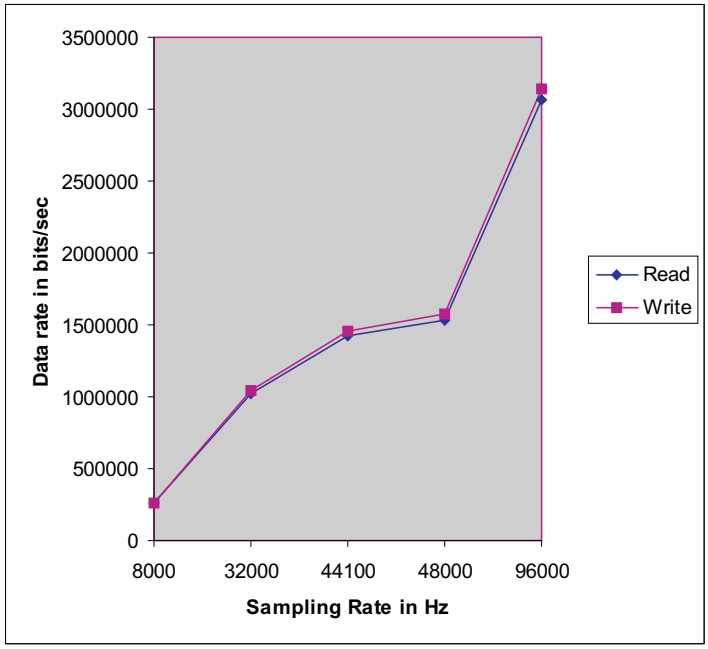

Figure 2-21. DM365 Audio OSS Read and Write Performance – LLD

Table 2-69. DM365 Audio OSS Read Performance – LLD

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	255740	5120	164.01	0.19
32000	16	2	1021602	5120	41.06	0.61
44100	16	2	1420284	5120	29.53	0.88
48000	16	2	1531966	5120	27.38	0.88
96000	16	2	3061327	5120	13.70	1.53

Table 2-70. DM365 Audio OSS Write Performance – LLD

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	262252	5120	159.93	0.18
32000	16	2	1048252	5120	40.01	0.65
44100	16	2	1457611	5120	28.78	0.45
48000	16	2	1572309	5120	26.68	0.75
96000	16	2	3144189	5120	13.34	1.65

2.5.5.4 Audio OSS Read and Write Performance Real Time Preemption

The performance numbers were captured using the following:

- ARM / DSP frequency = 297 MHz
- DDR frequency = 243 MHz
- Application buffer size = 4096 bytes

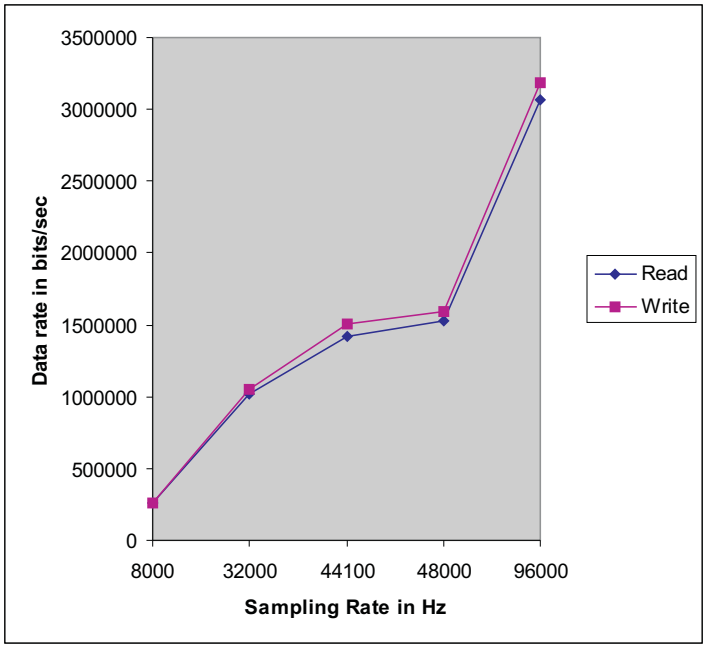

Figure 2-22. DM365 Audio OSS Write and Read Performance – RT

Table 2-71. DM365 Audio OSS Read Performance – RT

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	259003	5120	161.94	0.53
32000	16	2	1021600	5120	41.06	0.95
44100	16	2	1420282	5120	29.53	1.18
48000	16	2	1531962	5120	27.38	1.17
96000	16	2	3061260	5120	13.70	2.76

Table 2-72. DM365 Audio OSS Write Performance – RT

Sampling Rate in Hz	Word Length in bits	No. of Channels per Sample	No. of bits/Sec	Buffer Size in KB	Duration in Sec	%CPU Occupancy
8000	16	2	262252	5120	159.93	0.52
32000	16	2	1048253	5120	40.01	1.37
44100	16	2	1501615	5120	27.93	1.18
48000	16	2	1592364	5120	26.34	1.29
96000	16	2	3184272	5120	13.17	2.27

2.6 Ethernet Driver

The Ethernet driver supports the Linux 2.6 NAPI Interface. The driver communicates to the PHY, which is connected to the MAC layer using the MDIO interface.

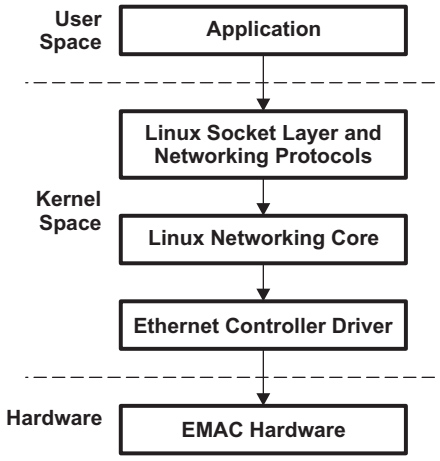

Figure 2-23. Linux Kernel Ethernet Driver

Following features are supported by the driver:

- 10/100 Mbps Speed
- Auto Negotiation
- Multicast and Broadcast
- Promiscuous mode
- Full and Half Duplex modes
- DM6467
  - 1000 Mbps speed

2.6.1 Support and Constraints

Features Not Supported None

Constraints

- Interrupt pacing feature
- Multiple receive and transmit channel are not supported.
- QoS support with VLAN tag discrimination using multiple RX channels.
- Receive Flow Control using RXnFREEBUFFER and RXnFLOWTHRESH registers.
- Currently Gigabit Ethernet facility is not available on DM6467.
- Hard-coding of PHY_MASK to 0x3 (up to 0x3) may not work on another board design (different from TI EVMs).

2.6.2 DM644x Performance and Benchmarks

2.6.2.1 Cross Cable Low Latency Desktop Preemption

The performance numbers were captured using the following:

- ARM frequency = 270 MHz
- CPMAC is configured in auto negotiation, 100Mbps full duplex.
- The iperf tool is run on DUT1 in server mode and DUT2 in client mode (DM644x EVM).
- Server side cmd: "iperf -s "
- Client side cmd: "iperf -c <server ip> -w<window size/2> -d -t60"
- Iperf version 2.0.2 (with patch-iperf-linux-2.6.21 bugfix) was used for the tests.
- The data captured here is for iperf in client mode.
- The DM644x EVM has been booted through hard disk boot.

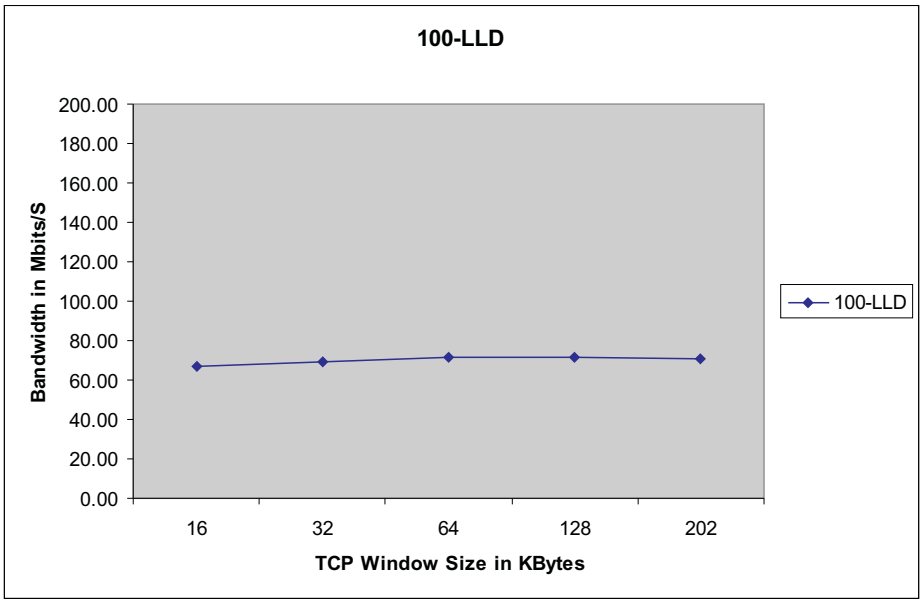

Figure 2-24. DM644x CPMAC 100 Performance – LLD

Table 2-73. DM644x CPMAC 100 Performance Values – LLD

TCP Window Size in KBytes	Bandwidth Mbits/Sec(Tx+Rx)	Transfer Size in MBytes	Interval in Sec
16	67.09	479.2	60
32	69.4	496	60
64	71.3	510	60
128	71.2	509	60
202	71.1	509	60

2.6.2.2 Cross Cable Real Time Preemption

The performance numbers were captured using the following:

- ARM frequency = 270 MHz
- CPMAC is configured in auto negotiation, 100Mbps full duplex.
- The iperf tool is run on DUT1 in server mode and DUT2 in client mode (DM644x EVM).
- Server side cmd: "iperf -s "
- Client side cmd: "iperf -c <server ip> -w<window size/2> -d -t60"
- Iperf version 2.0.2 (with patch-iperf-linux-2.6.21 bugfix) was used for the tests.
- The data captured here is for iperf in client mode.
- The DM644x EVMs has been booted through hard disk boot.

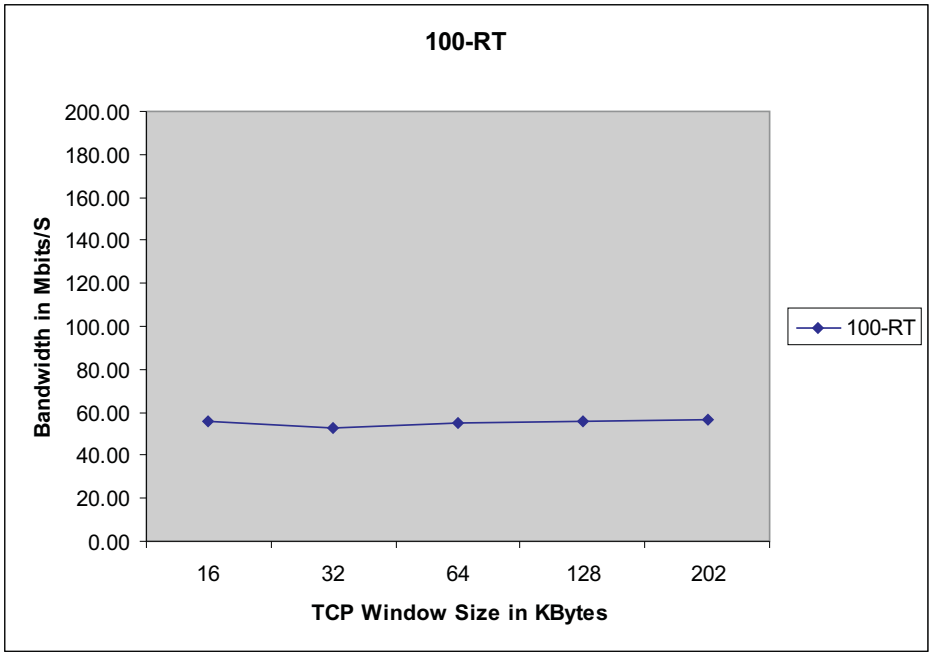

Figure 2-25. DM644x CPMAC 100 Performance – RT

Table 2-74. DM644x CPMAC 100 Performance Values – RT

TCP Window Size in KBytes	Bandwidth Mbits/Sec(Tx+Rx)	Transfer Size in MBytes	Interval in Sec
16	56.17	402.6	60
32	52.9	378	60
64	54.9	392	60
128	56.2	402	60
212	56.7	405	60

2.6.3 DM355 Performance and Benchmarks

2.6.3.1 Straight Cable Low Latency Desktop Preemption

The performance numbers were captured using the following:

- ARM frequency = 270 MHz
- The iperf tool is run on Linux PC on network in server mode and DUT (DM355 EVM) in client mode.
- Server side cmd: "iperf -s "
- Client side cmd: "iperf -c <server ip> -w<window size/2> -d -t60"
- Iperf version 2.0.4 (with patch-iperf-linux-2.6.21 bugfix) was used for the tests.
- The data captured here is for iperf in server mode.
- Speed is set to 100Mbps

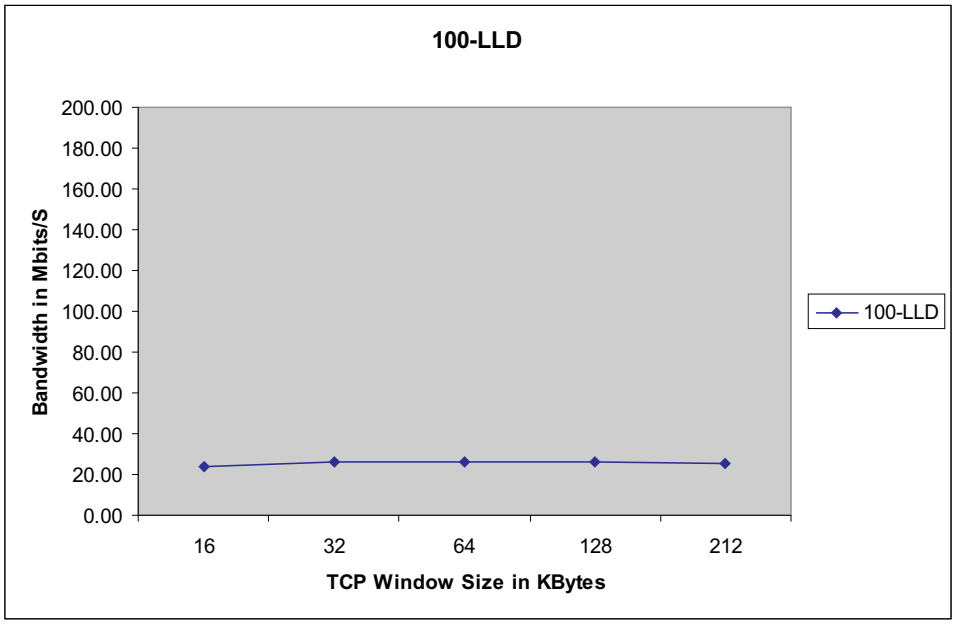

Figure 2-26. DM355 CPGMAC 100 Performance – LLD

Table 2-75. DM355 CPGMAC 100 Performance Values – LLD

TCP Window Size in KBytes	Bandwidth Mbits/Sec	Transfer Size in MBytes	Interval in Sec
16	24.20	172.5	60
32	25.80	184.6	60
64	25.80	184.2	60
128	25.80	185	60
212	25.50	182.7	60

2.6.3.2 Straight Cable Real Time Preemption

The performance numbers were captured using the following:

- ARM frequency = 270 MHz
- The iperf tool is run on Linux PC on network in server mode and DUT (DM355 EVM) in client mode.
- Server side cmd: "iperf -s "
- Client side cmd: "iperf -c <server ip> -w<window size/2> -d -t60"
- Iperf version 2.0.4 (with patch-iperf-linux-2.6.21 bugfix) was used for the tests.
- The data captured here is for iperf in server mode.
- Speed is set to 100Mbps

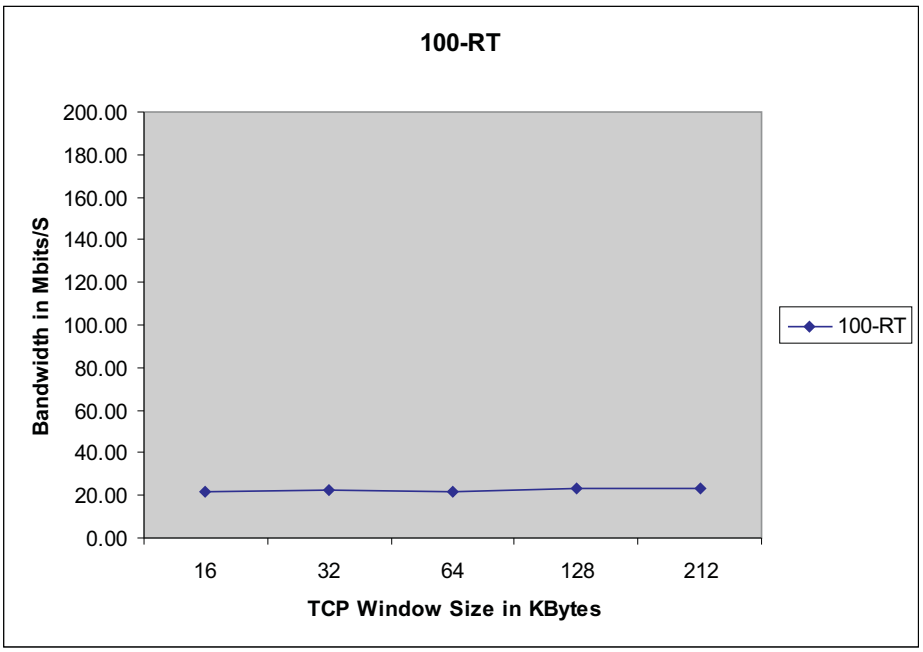

Figure 2-27. DM355 CPGMAC 100 Performance – RT

Table 2-76. DM355 CPGMAC 100 Performance Values – RT

TCP Window Size in KBytes	Bandwidth Mbits/Sec	Transfer Size in MBytes	Interval in Sec
16	21.40	151.99	60
32	22.24	159.3	60
64	21.86	156.2	60
128	23.20	166.2	60
212	23.20	166.2	60

2.6.4 DM365 Performance and Benchmarks

2.6.4.1 Cross Cable Low Latency Desktop Preemption

The performance numbers were captured using the following:

- ARM frequency = 297 MHz
- CPMAC is configured in auto negotiation, full duplex.
- The iperf tool is run on DUT1 in server mode and DUT2 in client mode (DM365 EVM).
- Server side cmd: "iperf -s "
- Client side cmd: "iperf -c <server ip> -w<window size/2> -d -t60"
- Iperf version 2.0.2 (with patch-iperf-linux-2.6.21 bugfix) was used for the tests.
- The data captured here is for iperf in client mode.

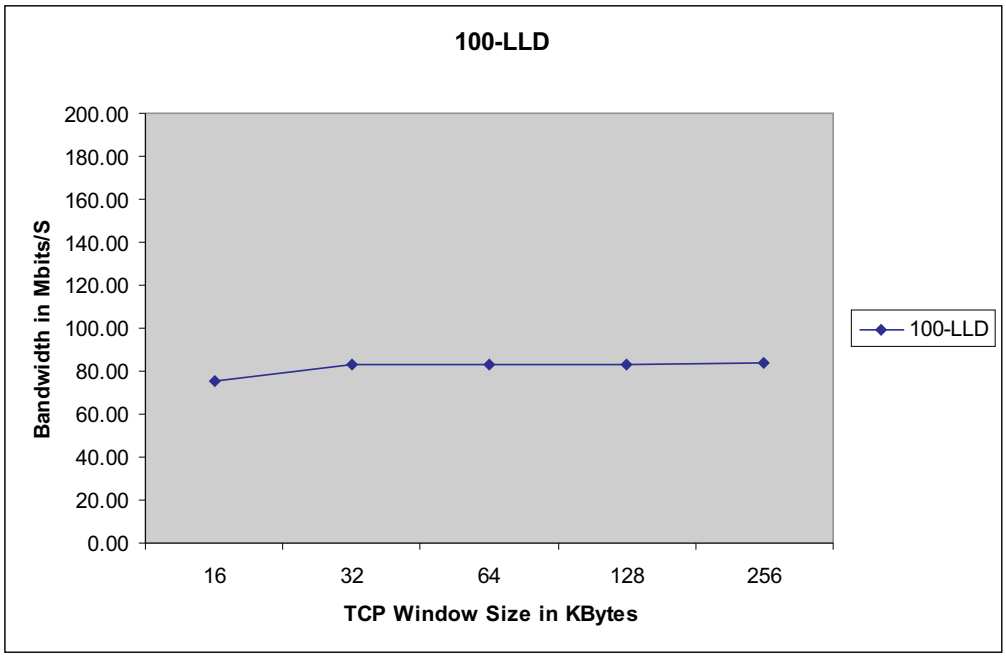

Figure 2-28. DM365 TCP Performance – LLD

Table 2-77. DM365 TCP Performance Values – LLD

TCP Window Size in KBytes	Bandwidth Mbits/Sec(Tx+Rx)	Transfer Size in MBytes	Interval in Sec
16	75.020	536.50	66
32	83.100	594.00	60
64	83.400	596.00	60
128	83.400	597.00	60
256	83.600	598.00	60

2.6.4.2 Cross Cable Real Time Preemption

The performance numbers were captured using the following:

- CPMAC is configured in auto negotiation, 100Mbps full duplex.
- The iperf tool is run on DUT1 in server mode and DUT2 in client mode.
- Server side cmd: "iperf -s "
- Client side cmd: "iperf -c <server ip> -w<window size/2> -d -t60"
- Iperf version 2.0.2 (with patch-iperf-linux-2.6.21 bugfix) was used for the tests.
- The data captured here is for iperf in client mode.

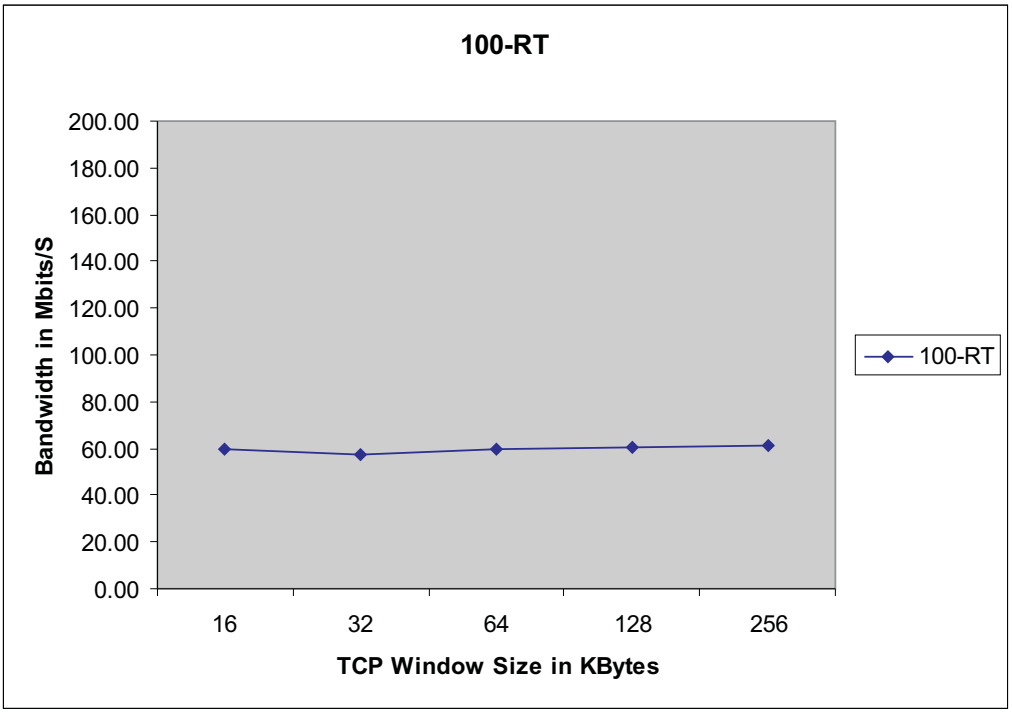

Figure 2-29. DM365 TCP Performance – RT

Table 2-78. DM365 TCP Performance Values – RT

TCP Window Size in KBytes	Bandwidth Mbits/Sec(Tx+Rx)	Transfer Size in MBytes	Interval in Sec
16	59.8	428	60
32	57.2	409	60
64	59.4	425	60
128	60.8	435	60
256	61.5	440	60

2.7 USB Drivers

The USB driver is implemented as a communication driver and supports MSC, CDC, RNDIS, HID, Audio/Video ISO classes.

On the DM644x and DM365, CPPI DMA is used for both host and device operations.

On the DM6467, internal CPPI DMA engine is used for both host and device operations.

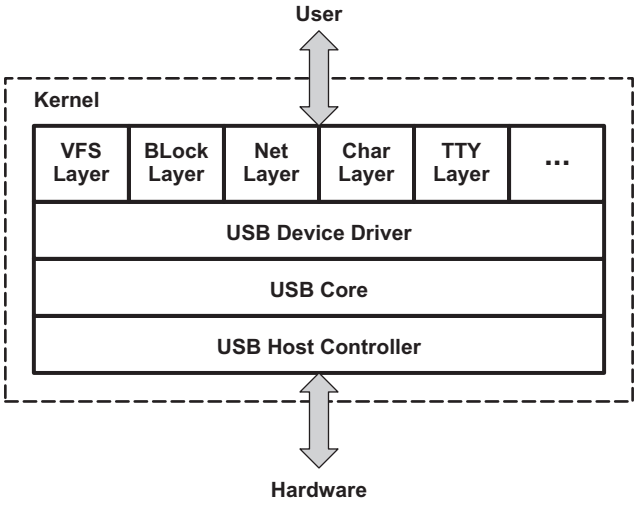

Figure 2-30. Linux Kernel USB Driver

Following features are supported by the driver:

- Data transfer in both PIO and DMA mode for all preemption modes
- Internal CPPI DMA is used for both host and device operations.
- Interrupt Endpoint scheduling feature to overcome the limitations of availability of number of Interrupt endpoints.
- Provision to reserve bulk or Isochronous endpoints according to the necessity.
- MSC Host and Slave support
- USB HID Host support with hiddev and evdev interfaces
- CDC Slave support
- RNDIS Slave support
- USB-ISO Audio/Video Host support

2.7.1 Support and Constraints

Features Not Supported NTFS file system for USB MSC Class

Constraints

- A/V tools (Mplayer/madplay) utilize the DaVinci ARM CPU 100%. Therefore, when these tools are used, only support of 1 USB ISO device is considered.
- MSC device performance in presence of ISO devices may be reduced. The primary reason is the lack of ARM CPU bandwidth.
- Implementation provisions the option of reserving an EP (Rx,Tx) combination for handling ISO devices. If this option is not chosen, then there could be situations were ISO devices are not supported due to lack of Rx EPs in the controller.

- Following Hardware (DaVinci USB Controller) issues exists, which may impact ISO support:
  - CPPI Rx ISO DMA stalls randomly. This leads to ISO capture timing out. Work around is to start some un-related USB activity with some other USB devices such as connecting/disconnecting, USB mouse movement, USB keyboard stroke and so on. This restarts the DMA and Rx ISO operates correctly.
  - MSC device enumeration/IO (Rx) fails in presence of ISO (TX) traffic on the same EP number. Work around is implemented as part of option for reserving ISO EP in a separate EP from the EP reserved for BULK.
  - High bandwidth ISO device are not supported as DaVinci USB IP does not support High Bandwidth EPs.
- Hang is observed during playback using USB ISO Speaker is in DMA mode.
- Hangs are observed for USB ISO audio playback at different sampling rates for PIO mode in RT preemption mode when application is restarted more than once.
- snd_pcm_readi() does not read the number of frames given as input parameter if it is more than period size with USB ISO Audio.
- Removal of musb_hdrc.ko is successful when an IO is in progress with USB ISO Audio and Video applications.
- Removal of usbhid.ko is successful when an input is in progress through keyboard/mouse.
- USB Slave Write operation fails for DMA/PIO transfers in Low Latency Desktop preemption modes on Windows 2000 PC (DM644x).
- The ping between host and slave is not successful, if the USB cable is connected before the EVM boots up (DM644x).
- Driver hang is observed while measuring USB-CDC/RNDIS performance using iperf tool.(DM6467).

Supported System Calls open(), close(), read(), write(), ioctl()

Supported IOCTLS None

2.7.2 DM644x Performance and Benchmarks

2.7.2.1 USB CDC / RNDIS DMA Low Latency Desktop and Real Time Preemption

The performance numbers were captured using the following:

- Server side cmd: “iperf -s”
- Client side cmd: “iperf -c <server ip> -w <window size /2> -d –t60
- RNDIS Setup: iperf version 1.7.0 DUT Slave connected to WinXP SP2 (WinXP Dell GX280 P4, 1GB RAM) Host
- CDC Setup: iperf version 2.0.2 (with patch-iperf-linux-2.6.21 bugfix) DUT Slave connected to 2.6.9 Linux (Dell GX270 P4, 512 RAM) Host
- Kernel is configured in DMA mode
- The data captured is for “iperf” in Client mode

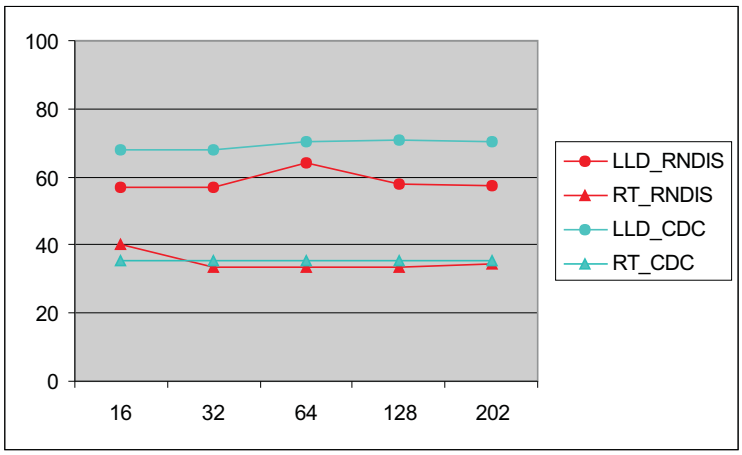

Figure 2-31. DM644x USB CDC/RNDIS DMA Performance

Table 2-79. DM644x USB CDC/RNDIS DMA Performance Values

TCP Window Size in Kbytes	Bandwidth Mbits/Sec			
	LLD_RNDIS	RT_RNDIS	LLD_CDC	RT_CDC
16	57.05	39.96	67.87	35.20
32	56.90	33.55	68.00	35.30
64	64.20	33.31	70.40	35.20
128	57.90	33.58	70.80	35.20
202	57.50	34.60	70.40	35.40

2.7.2.2 USB ISO Video Low Latency Desktop Preemption

The performance numbers were captured using the following:

- Open source Philips Webcam pwc driver was used (version 10.0.12-rc1)
- The readings were taken using the Logitech Quickcam Notebook Pro USB

Table 2-80. DM644x USB ISO Video Performance Values – LLD

Webcam Used	Mode	Capture Frame Rate (fps)	% CPU Load During Capture
Logitech Quickcam Notebook Pro USB	DMA	10.40	18.06

2.7.2.3 USB ISO Video Real Time Preemption

The performance numbers were captured using the following:

- Open source Philips Webcam pwc driver was used (version 10.0.12-rc1)
- The readings were taken using the Logitech Quickcam Notebook Pro USB

Table 2-81. DM644x USB ISO Video Performance Values – RT

Webcam Used	Mode	Capture Frame Rate (fps)	% CPU Load During Capture
Logitech Quickcam Notebook Pro USB	DMA	10.57	19.47

2.7.2.4 USB MSC Host DMA Low Latency Desktop Preemption

The performance numbers were captured using the following:

- iomega LDHD500-U USB HD 500GB 3.5" drive
- ARM frequency = 216 MHz
- File format = EXT3

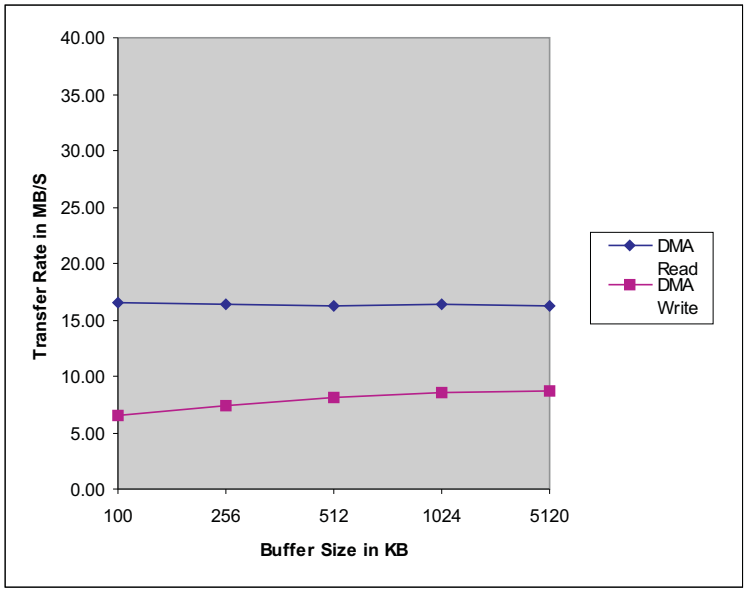

Figure 2-32. DM644x USB MSC Host DMA Performance – LLD

Table 2-82. DM644x USB MSC Host DMA Read Performance Values – LLD

Buffer Size (KB)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)
100	100	16.50
256	100	16.37
512	100	16.30
1024	100	16.34
5120	100	16.25

Table 2-83. DM644x USB MSC Host DMA Write Performance Values – LLD

Buffer Size (KB)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)
100	100	6.45
256	100	7.46
512	100	8.13

Table 2-83. DM644x USB MSC Host DMA Write Performance Values – LLD (continued)

Buffer Size (KB)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)
1024	100	8.50
5120	100	8.67

2.7.2.5 USB MSC Host DMA Real Time Preemption

The performance numbers were captured using the following:

- iomega LDHD500-U USB HD 500GB 3.5" drive
- ARM frequency = 216 MHz
- File format = EXT3

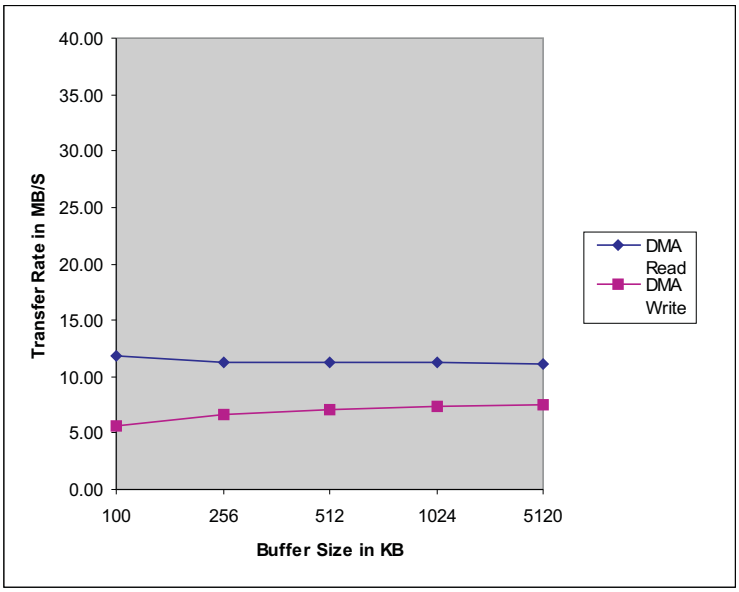

Figure 2-33. DM644x USB MSC Host DMA Performance – RT

Table 2-84. DM644x USB MSC Host DMA Read Performance Values – RT

Buffer Size (KB)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)
100	100	11.81
256	100	11.23
512	100	11.29
1024	100	11.27
5120	100	11.18

Table 2-85. DM644x USB MSC Host DMA Write Performance Values – RT

Buffer Size (KB)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)
100	100	5.68
256	100	6.61
512	100	7.10
1024	100	7.35
5120	100	7.45

2.7.3 DM355 Performance and Benchmarks

2.7.3.1 USB CDC/RNDIS DMA Low Latency Desktop and Real Time Preemption

The performance numbers were captured using the following:

- Server side cmd: “iperf –s”
- Client side cmd: “iperf -c <server ip> -w <window size /2> -d –t60
- RNDIS Setup: iperf version 1.7.0 DUT Slave connected to WinXP SP2 (WinXP Dell GX280 P4, 1GB RAM) Host
- CDC Setup: iperf version 2.0.4 on DUT Slave connected to 2.6.9 Linux (Dell GX270 P4, 512 RAM) Host
- Kernel is configured in DMA (for USB) mode
- The data captured is for “iperf” in Client mode

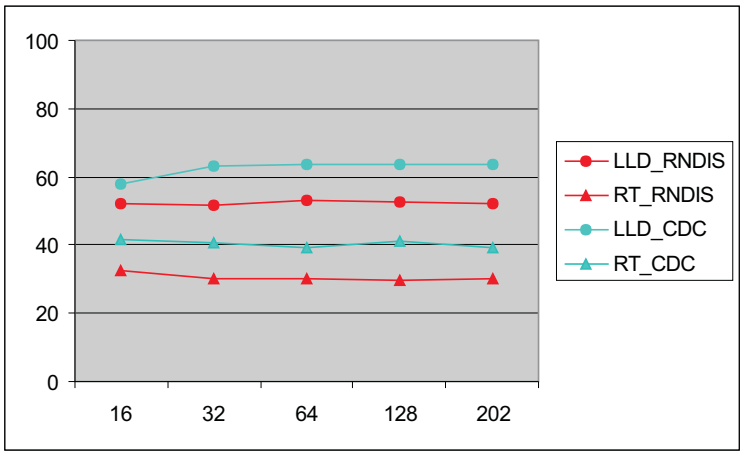

Figure 2-34. DM355 USB CDC/RNDIS DMA Performance

Table 2-86. DM355 USB CDC/RNDIS DMA Performance Values

TCP Window Size in Kbytes	Bandwidth Mbits/Sec			
	LLD_RNDIS	RT_RNDIS	LLD_CDC	RT_CDC
16	39.95	25.05	47.61	26.94
32	37.90	21.84	45.60	32.95
64	38.40	24.27	45.60	33.27
128	38.90	26.00	45.80	33.93
202	39.20	29.06	46.00	32.59

2.7.3.2 USB ISO Video Low Latency Desktop Preemption

The performance numbers were captured using the following:

- Open source Philips Webcam pwc driver was used (version 10.0.12-rc1)
- The readings were taken using the Logitech Quickcam Notebook Pro USB

Table 2-87. DM355 USB ISO Video Performance Values – LLD

Webcam Used	Mode	Capture Frame Rate (fps)	% CPU Load During Capture
Logitech Quickcam Notebook Pro USB	DMA	10.20	13.17

2.7.3.3 USB ISO Video Real Time Preemption

The performance numbers were captured using the following:

- Open source Philips Webcam pwc driver was used (version 10.0.12-rc1)
- The readings were taken using the Logitech Quickcam Notebook Pro USB

Table 2-88. DM355 USB ISO Video Performance Values – RT

Webcam Used	Mode	Capture Frame Rate (fps)	% CPU Load During Capture
Logitech Quickcam Notebook Pro USB	DMA	10.46	11.76

2.7.3.4 USB MSC Host DMA Low Latency Desktop Preemption

The performance numbers were captured using the following:

- iomega LDHD500-U USB HD 500GB 3.5" drive
- ARM frequency = 216 MHz
- File format = EXT3

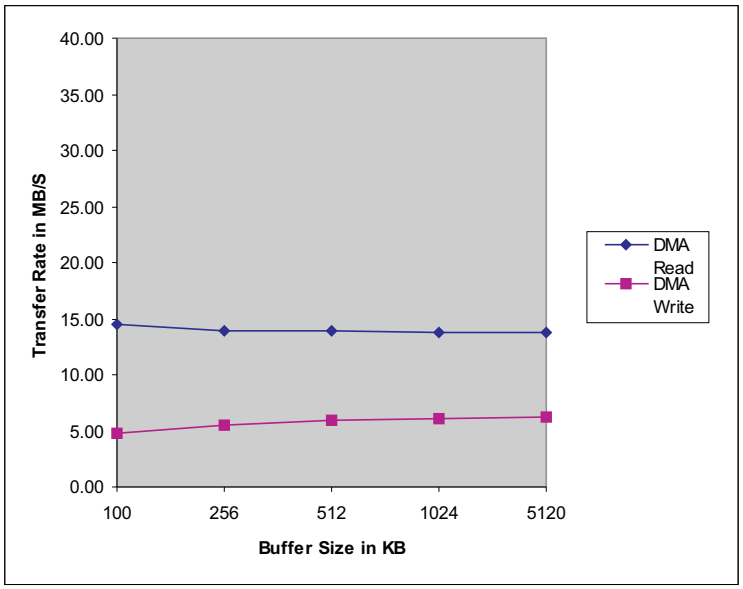

Figure 2-35. DM355 USB MSC Host DMA Performance – LLD

Table 2-89. DM355 USB MSC Host DMA Read Performance Values – LLD

Buffer Size (KB)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)
100	100	14.45
256	100	13.95
512	100	13.96
1024	100	13.83
5120	100	13.72

Table 2-90. DM355 USB MSC Host DMA Write Performance Values – LLD

Buffer Size (KB)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)
100	100	4.72
256	100	5.44
512	100	5.91
1024	100	6.12
5120	100	6.22

2.7.3.5 USB MSC Host DMA Real Time Preemption

The performance numbers were captured using the following:

- iomega LDHD500-U USB HD 500GB 3.5" drive
- ARM frequency = 216 MHz
- File format = EXT3

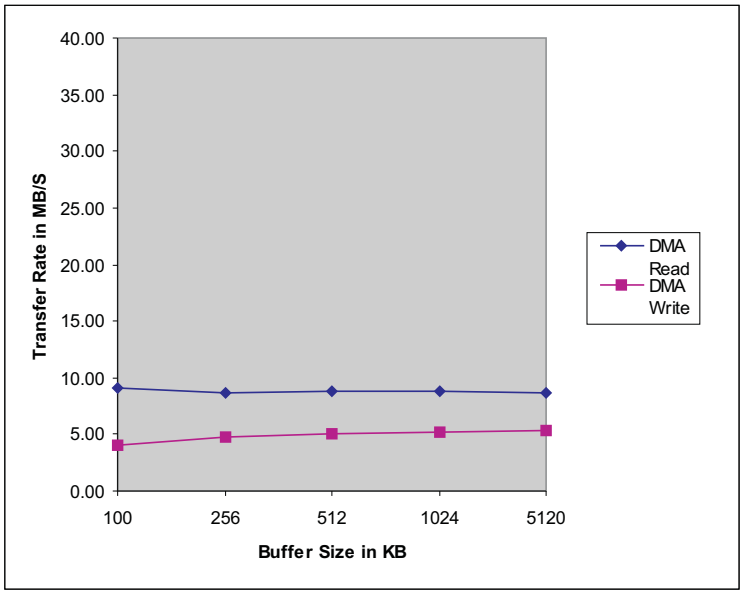

Figure 2-36. DM355 USB MSC Host DMA Performance – RT

Table 2-91. DM355 USB MSC Host DMA Read Performance Values – RT

Buffer Size (KB)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)
100	100	9.13
256	100	8.73
512	100	8.77
1024	100	8.74
5120	100	8.67

Table 2-92. DM355 USB MSC Host DMA Write Performance Values – RT

Buffer Size (KB)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)
100	100	4.03
256	100	4.72
512	100	5.07
1024	100	5.18
5120	100	5.35

2.7.4 DM6467 Performance and Benchmarks

2.7.4.1 USB CDC/RNDIS DMA Low Latency Desktop and Real Time Preemption

The performance numbers were captured using the following:

- Server side cmd: “iperf –s”
- Client side cmd: “iperf -c <server ip> -w <window size /2> -d –t60
- RNDIS Setup: iperf version 1.7.0 DUT Slave connected to WinXP SP2 (WinXP Dell GX280 P4, 1GB RAM) Host
- CDC Setup: iperf version 2.0.2 (with patch-iperf-bugfix) on DUT Slave connected to 2.6.9 Linux (Dell GX270 P4, 512 RAM) Host
- Kernel is configured in DMA (for USB) mode
- The data captured is for “iperf” in Client mode

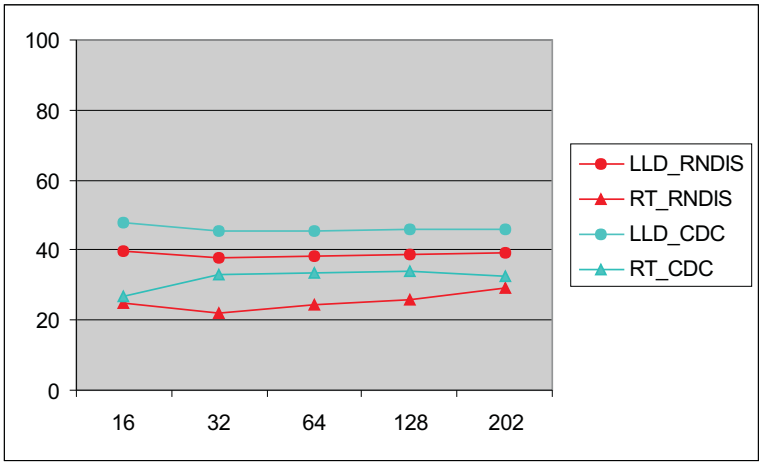

Figure 2-37. DM6467 USB CDC/RNDIS DMA Performance

Table 2-93. DM6467 USB CDC/RNDIS DMA Performance Values

TCP Window Size in Kbytes	Bandwidth Mbits/Sec			
	LLD_RNDIS	RT_RNDIS	LLD_CDC	RT_CDC
16	39.95	25.05	47.61	26.94
32	37.90	21.84	45.60	32.95
64	38.40	24.27	45.60	33.27
128	38.90	26.00	45.80	33.93
202	39.20	29.06	46.00	32.59

2.7.4.2 USB ISO Video Low Latency Desktop Preemption

The performance numbers were captured using the following:

- Open source Philips Webcam pwc driver was used (version 10.0.12-rc1)
- The readings were taken using the Logitech Quickcam Notebook Pro USB

Table 2-94. DM6467 USB ISO Video Performance Values – LLD

Webcam Used	Mode	Capture Frame Rate (fps)	% CPU Load During Capture
Logitech Quickcam Notebook Pro USB	DMA	10.69	19.08

2.7.4.3 USB ISO Video Real Time Preemption

The performance numbers were captured using the following:

- Open source Philips Webcam pwc driver was used (version 10.0.12-rc1)
- The readings were taken using the Logitech Quickcam Notebook Pro USB

Table 2-95. DM6467 USB ISO Video Performance Values – RT

Webcam Used	Mode	Capture Frame Rate (fps)	% CPU Load During Capture
Logitech Quickcam Notebook Pro USB	DMA	10.82	28.30

2.7.4.4 USB MSC Host DMA Low Latency Desktop Preemption

The performance numbers were captured using the following:

- Maxtor USB Device
- ARM frequency = 297 MHz
- File format = EXT3

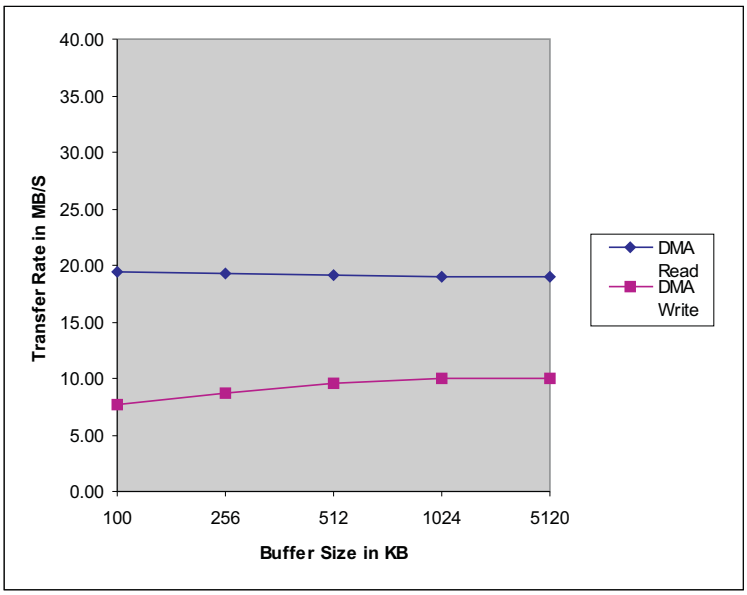

Figure 2-38. DM6467 USB MSC Host DMA Performance – LLD

Table 2-96. DM6467 USB MSC Host DMA Read Performance Values – LLD

Buffer Size (KB)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)
100	100	19.36
256	100	19.31
512	100	19.11
1024	100	19.04
5120	100	18.93

Table 2-97. DM6467 USB MSC Host DMA Write Performance Values – LLD

Buffer Size (KB)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)
100	100	7.61
256	100	8.68
512	100	9.54
1024	100	9.97
5120	100	10.05

2.7.4.5 USB MSC Host DMA Real Time Preemption

The performance numbers were captured using the following:

- Maxtor USB Device
- ARM frequency = 297 MHz
- File format = EXT3

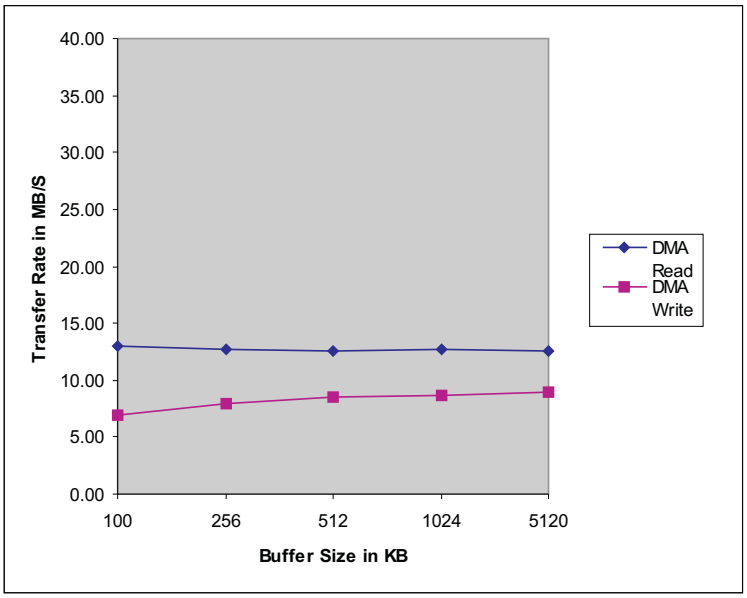

Figure 2-39. DM6467 USB MSC Host DMA Performance – RT

Table 2-98. DM6467 USB MSC Host DMA Read Performance Values – RT

Buffer Size (KB)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)
100	100	13.01
256	100	12.64
512	100	12.61
1024	100	12.71
5120	100	12.54

Table 2-99. DM6467 USB MSC Host DMA Write Performance Values – RT

Buffer Size (KB)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)
100	100	6.91
256	100	7.93
512	100	8.46
1024	100	8.70
5120	100	8.93

2.7.5 DM365 Performance and Benchmarks

2.7.5.1 USB CDC/RNDIS DMA Low Latency Desktop and Real Time Preemption

The performance numbers were captured using the following:

- Server side cmd: “iperf –s”
- Client side cmd: “iperf -c <server ip> -w <window size /2> -d –t60
- RNDIS Setup: iperf version 1.7.0 DUT Slave connected to WinXP SP2 (WinXP Dell GX280 P4, 1GB RAM) Host
- CDC Setup: iperf version 2.0.2 (with patch-iperf-linux-2.6.21 bugfix) DUT Slave connected to 2.6.9 Linux (Dell GX270 P4, 512 RAM) Host
- Kernel is configured in DMA mode
- The data captured is for “iperf” in Client mode

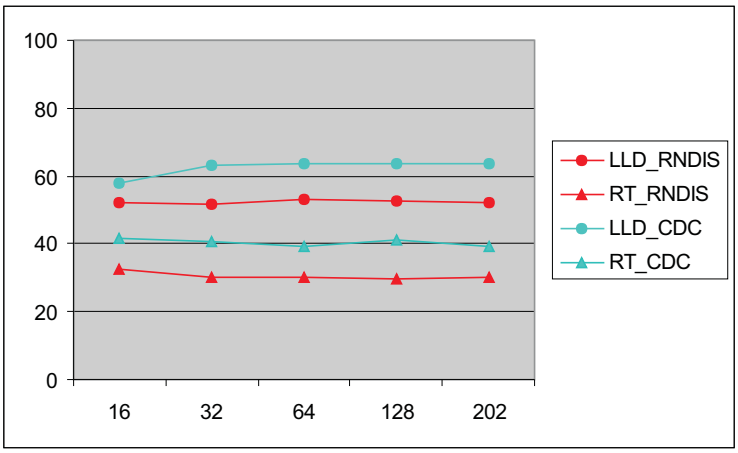

Figure 2-40. DM365 USB CDC/RNDIS Performance

Table 2-100. DM365 USB CDC/RNDIS DMA Performance Values

TCP Window Size in Kbytes	Bandwidth Mbits/Sec			
	LLD_RNDIS	RT_RNDIS	LLD_CDC	RT_CDC
16	52.12	32.44	58.10	41.40
32	51.80	30.05	63.30	40.60
64	53.10	30.01	63.60	39.07
128	52.60	29.49	63.60	41.30
202	52.30	30.37	63.70	39.27

2.7.5.2 USB ISO Video Low Latency Desktop Preemption

The performance numbers were captured using the following:

- Open source Philips Webcam pwc driver was used (version 10.0.12-rc1)
- The readings were taken using the Logitech Quickcam Notebook Pro USB Webcam
- Performance values were taken in both DMA and PIO modes

Table 2-101. DM365 USB ISO Video Performance Values – LLD

Webcam Used	Mode	Capture Frame Rate (fps)	% CPU Load During Capture
Logitech Quickcam Notebook Pro USB	DMA	10.18	10.72

2.7.5.3 USB ISO Video Real Time Preemption

The performance numbers were captured using the following:

- Open source Philips Webcam pwc driver was used (version 10.0.12-rc1)
- The readings were taken using the Logitech Quickcam Notebook Pro USB Webcam

Table 2-102. DM365 USB ISO Video Performance Values – RT

Webcam Used	Mode	Capture Frame Rate (fps)	% CPU Load During Capture
Logitech Quickcam Notebook Pro USB	DMA	10.55	57.85

2.7.5.4 USB MSC Host DMA Low Latency Desktop Preemption

The performance numbers were captured using the following:

- Seagate FreeAgent USB Pro 500GB 3.5" drive
- ARM frequency = 297 MHz
- File format = EXT3

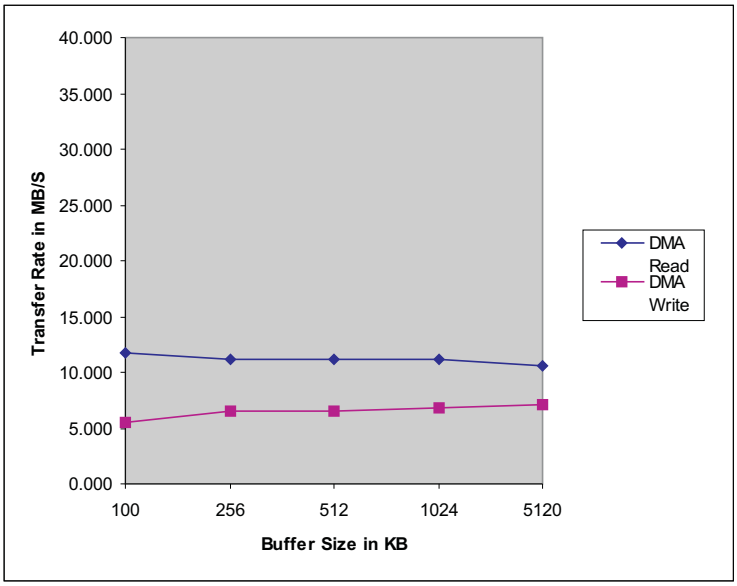

Figure 2-41. DM365 USB MSC Host DMA Performance – LLD

Table 2-103. DM365 USB MSC Host DMA Read Performance Values – LLD

Buffer Size (KB)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)
100	100	11.71
256	100	11.19
512	100	11.15
1024	100	11.11
5120	100	10.51

Table 2-104. DM365 USB MSC Host DMA Write Performance Values – LLD

Buffer Size (KB)	Total Bytes Transferred (MB)	Total Bytes Rate (MB/s)
100	100	5.52
256	100	6.46
512	100	6.46

Table 2-104. DM365 USB MSC Host DMA Write Performance Values – LLD (continued)

Buffer Size (KB)	Total Bytes Transferred (MB)	Total Bytes Rate (MB/s)
1024	100	6.85
5120	100	7.09

2.7.5.5 USB MSC Host DMA Real Time Preemption

The performance numbers were captured using the following:

- Seagate FreeAgent USB Pro 500GB 3.5" drive
- ARM frequency = 297 MHz
- File format = EXT3

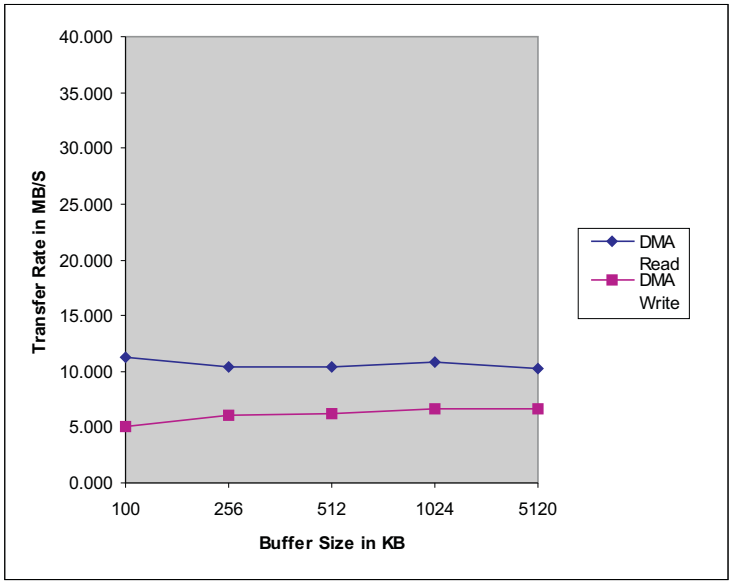

Figure 2-42. DM365 USB MSC Host DMA Performance – RT

Table 2-105. DM365 USB MSC Host DMA Read Performance Values – RT

Buffer Size (KB)	Total Bytes Transferred (MB)	Total Bytes Rate (MB/s)
100	100	11.27
256	100	10.41
512	100	10.37
1024	100	10.84
5120	100	10.30

Table 2-106. DM365 USB MSC Host DMA Write Performance Values – RT

Buffer Size (KB)	Total Bytes Transferred (MB)	Total Bytes Rate (MB/s)
100	100	5.12
256	100	6.08
512	100	6.21
1024	100	6.70
5120	100	6.67

2.8 IDE ATA Driver

The ATA controller provides a glueless interface to storage media. The IDE driver is implemented as a block driver. It supports ATA standards and creates the device nodes /dev/hda1, /dev/hda2, and so on for user space access.

On the DM644x and DM365, the IDE driver supports up to UDMA4 (UDMA0, UDMA1, UDMA2, UDMA3, and UDMA4).

On the DM6467, the IDE driver supports up to UDMA4 (UDMA0, UDMA1, UDMA2, UDMA3, and UDMA4), mDMA2 (mDMA0, mDMA1, and mDMA2) and PIO4 (PIO0, PIO1, PIO2, PIO3, and PIO4).

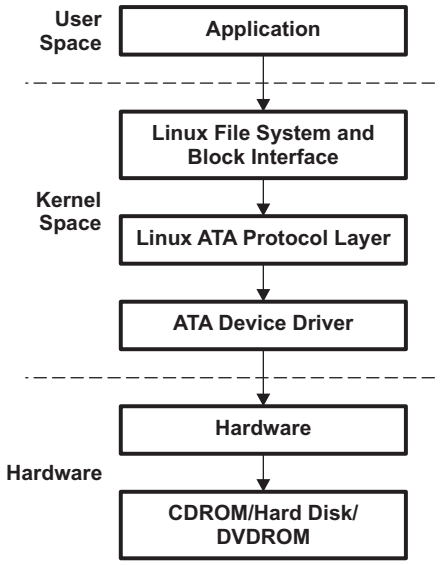

Figure 2-43. Linux Kernel IDE Driver

The IDE ATA driver supports the following features:

- The IDE ATA driver is accessible as a block device driver.
- The ATA controller driver plugs into the ATA protocol layer and block driver.
- The Linux file systems and block device layer abstract the details of the ATA block device.
- Mounting and dismounting file systems, associated file system creation/manipulation, can be performed by the user.

2.8.1 Support and Constraints

Features not supported NTFS file system is not supported

Constraints

- Performance not consistent with theoretical values
- The IDE device is not enabled by default in the menuconfig.
- Currently, MDMA or UDMA modes cannot be set.

Supported System Calls open(), close(), read(), write(), ioctl()

Supported IOCTLs See Table 2-107.

Table 2-107. Supported ATA IOCTLs

Constant	Description
HDIO_GETGEO	Gets device geometry
HDIO_GET_IDENTITY	Gets IDE identification information
HDIO_GET_NICE	Gets nice flags
HDIO_DRIVE_TASKFILE	Executes raw task file
HDIO_DRIVE_CMD	Executes a special drive command
HDIO_DRIVE_TASK	Executes task and special drive command
HDIO_SCAN_HWIF	Registers and (re) scans interface
HDIO_UNREGISTER_HWIF	Un-registers interface
HDIO_SET_NICE	Sets nice flags
HDIO_DRIVE_RESET	Executes a device reset
HDIO_GET_BUSSTATE	Gets the bus state of the HWIF
HDIO_SET_BUSSTATE	Sets the bus state of the HWIF

2.9 MMC/SD Driver

The MMC controller provides an interface to external MMC cards using MMC specification V3.31. Communication between the MMC controller and MMC card(s) is performed by the MMC/SD protocol. The ARM and EDMA controller can read/write the data in the card by accessing the registers in the MMC/SD controller. The MMC driver is implemented as a block driver shown in Figure 2-44. It creates the device nodes for user space access (/dev/mmcblkp1, /dev/mmcblkp2, etc.).

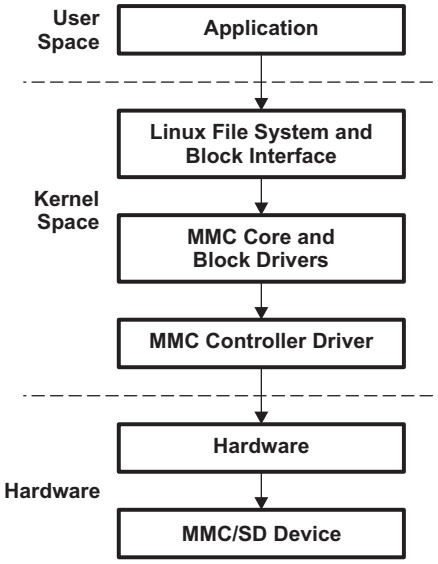

Figure 2-44. Linux Kernel MMC Driver Architecture

The MMC Device Driver supports the following features:

- The MMC drivers are accessible as block device drivers.
- DMA transfer mode
- The MMC controller driver plugs into the Linux MMC core and block driver.
- Linux file systems and block devices layer abstract details of block devices (MMC).
- Mounting and unmounting file systems, and associated file system creation/manipulation, is left to the user.
- The driver supports high-speed SD cards.
- Supports 4-bit mode.
- On the DM644x, DM355 and DM365, interrupt mode is supported.

2.9.1 Support and Constraints

Features not supported High-capacity SD support is not present in the driver.

Constraints

- Removing SD module with a mounted file system succeeds.
- Due to MMC divisor restrictions, the init clock operates at 200 KHz. At this frequency, some cards may not be recognized by Linux.
- Removing MMC module with a mounted file system succeeds.
- Some SD cards work fine at 50 MHz. Some SD cards fail with Response CRC errors, but can be used by ignoring them. Some SD cards fail with DATA CRC errors for all data transfer commands and become unusable.
- Certain warnings are observed when the MMC module built in PIO mode is removed.

Supported System Calls open(), close(), write() and read()

Supported IOCTLs None

2.9.2 DM644x Performance and Benchmarks

2.9.2.1 MMC DMA Low Latency Desktop Preemption

The performance numbers were captured using the following:

- MMCSD Card used for testing = SanDisk 2GB, 120x
- File system used = EXT2
- ARM clock frequency = 297 MHz

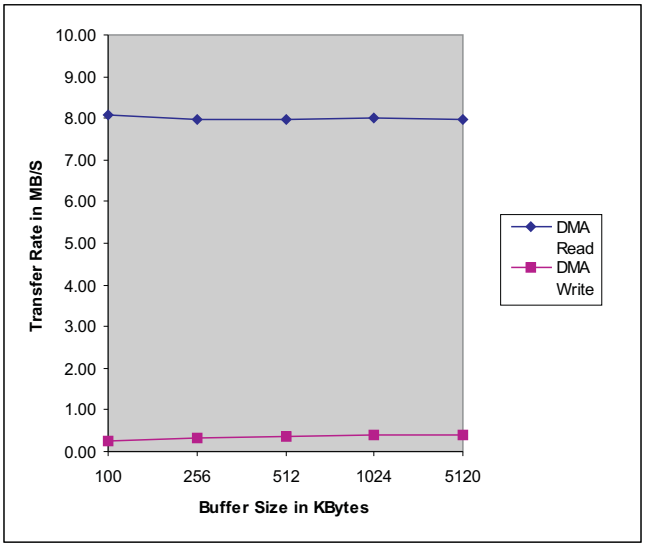

Figure 2-45. DM644x MMC DMA Performance – LLD

Table 2-108. DM644x MMC DMA Read Performance Values – LLD

Buffer Size In KB	Megabytes/sec	File Size in MB
100	8.06	100
256	7.98	100
512	7.96	100
1024	8.00	100
5120	7.96	100

Table 2-109. DM644x MMC DMA Write Performance Values – LLD

Buffer Size In KB	Megabytes/sec	File Size in MB
100	0.27	100
256	0.34	100
512	0.36	100
1024	0.38	100
5120	0.39	100

2.9.2.2 MMC DMA Real Time Preemption

The performance numbers were captured using the following:

- MMC Card used for testing = SanDisk 2GB, 120x
- File system used = EXT2
- ARM clock frequency and DDR frequency = 270 MHz

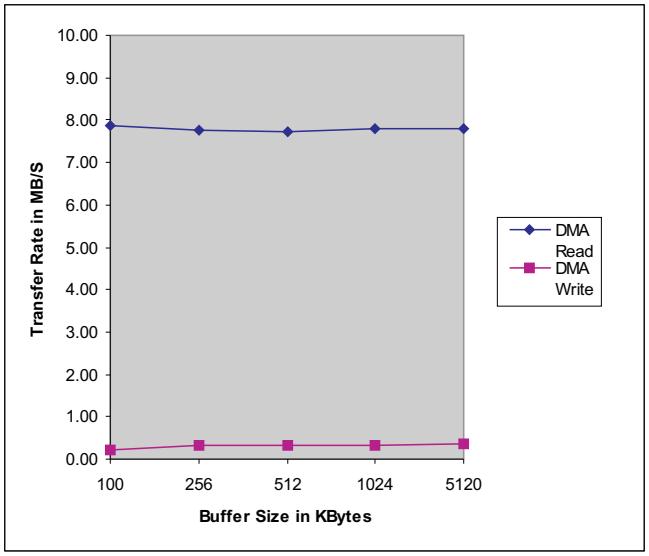

Figure 2-46. DM644x MMC DMA Performance – RT

Table 2-110. DM644x MMC DMA Read Performance Values – RT

Buffer Size In KB	Megabytes/sec	File Size in MB
100	7.88	100
256	7.76	100
512	7.73	100
1024	7.80	100
5120	7.78	100

Table 2-111. DM644x MMC DMA Write Performance Values – RT

Buffer Size In KB	Megabytes/sec	File Size in MB
100	0.23	100
256	0.32	100
512	0.33	100
1024	0.34	100
5120	0.36	100

2.9.3 DM355 Performance and Benchmarks

2.9.3.1 MMC DMA Low Latency Desktop Preemption

The performance numbers were captured using the following:

- MMCSD Card used for testing = Kingston 2 GB, 120x
- File system used = EXT2
- ARM clock frequency = 297 MHz

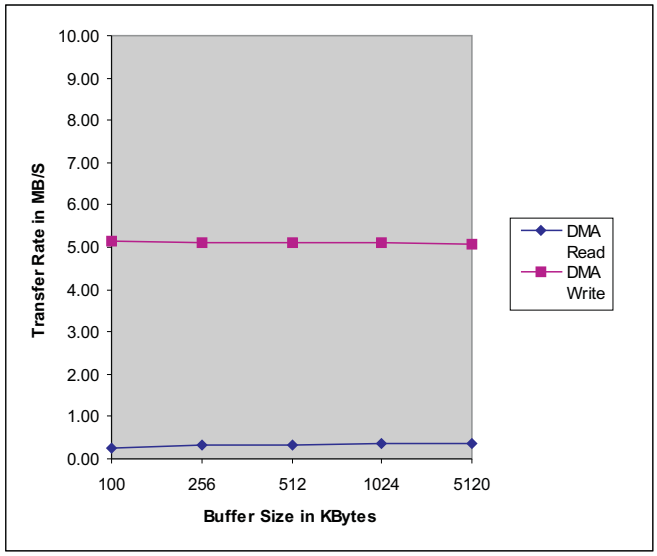

Figure 2-47. DM355 MMC DMA Performance – LLD

Table 2-112. DM355 MMC DMA Read Performance Values – LLD

Buffer Size In KB	Megabytes/sec	File Size in MB
100	5.13	100
256	5.11	100
512	5.09	100
1024	5.12	100
5120	5.09	100

Table 2-113. DM355 MMC DMA Write Performance Values – LLD

Buffer Size In KB	Megabytes/sec	File Size in MB
100	0.25	100
256	0.31	100
512	0.34	100
1024	0.35	100
5120	0.36	100

2.9.3.2 MMC DMA Real Time Preemption

The performance numbers were captured using the following:

- MMCSD Card used for testing = Kingston 2 GB, 120x
- File system used = EXT2
- ARM clock frequency and DDR frequency = 270 MHz

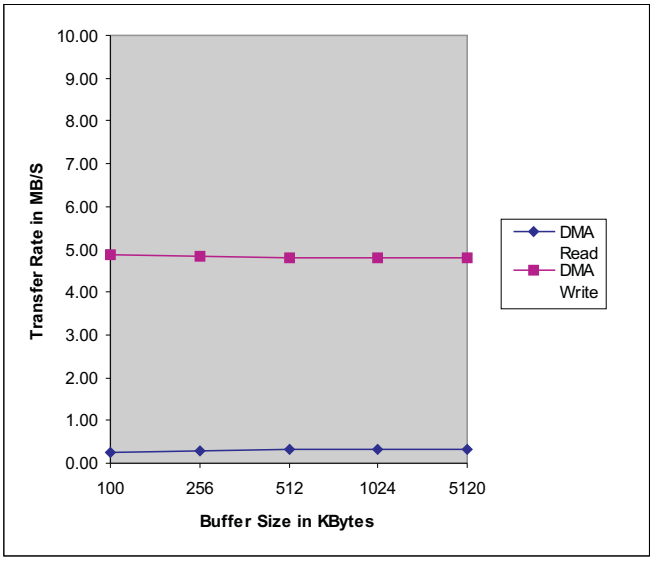

Figure 2-48. DM355 MMC DMA Performance – RT

Table 2-114. DM355 MMC DMA Read Performance Values – RT

Buffer Size In KB	Megabytes/sec	File Size in MB
100	4.86	100
256	4.82	100
512	4.80	100
1024	4.82	100
5120	4.80	100

Table 2-115. DM355 MMC DMA Write Performance Values – RT

Buffer Size In KB	Megabytes/sec	File Size in MB
100	0.24	100
256	0.29	100
512	0.31	100
1024	0.33	100
5120	0.33	100

2.9.4 DM6467 Performance and Benchmarks

2.9.4.1 MMC DMA Low Latency Desktop Preemption

The performance numbers were captured using the following:

- MMCSD Card used for testing = SanDisk 2 GB, 120x
- File system used = EXT2
- ARM clock frequency = 297 MHz

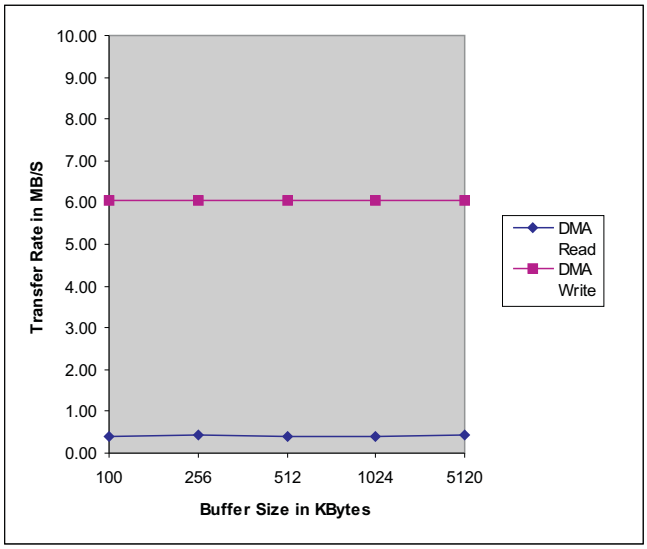

Figure 2-49. DM6467 MMC DMA Performance – LLD

Table 2-116. DM6467 MMC DMA Read Performance Values – LLD

Buffer Size In KB	Megabytes/sec	File Size in MB
100	6.06	100
256	6.06	100
512	6.05	100
1024	6.05	100
5120	6.05	100

Table 2-117. DM6467 MMC DMA Write Performance Values – LLD

Buffer Size In KB	Megabytes/sec	File Size in MB
100	0.42	100
256	0.42	100
512	0.42	100
1024	0.42	100
5120	0.42	100

2.9.4.2 MMC DMA Real Time Preemption

The performance numbers were captured using the following:

- MMCSD Card used for testing = SanDisk 2GB, 120x
- File system used = EXT2
- ARM clock frequency and DDR frequency = 297 MHz

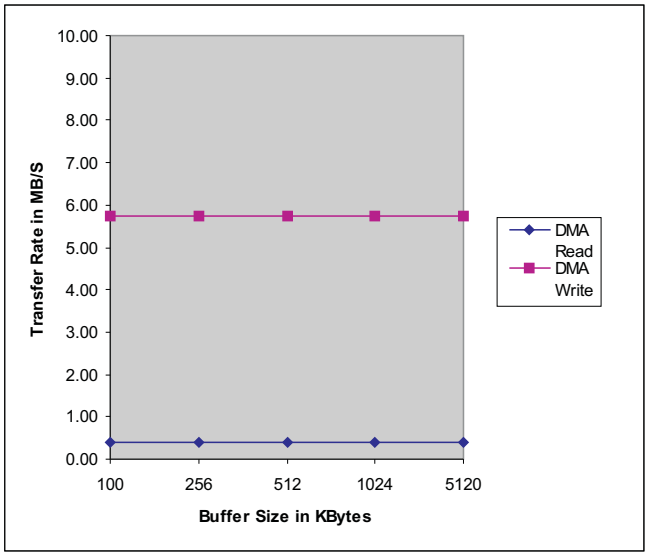

Figure 2-50. DM6467 MMC DMA Performance – RT

Table 2-118. DM6467 MMC DMA Read Performance Values – RT

Buffer Size In KB	Megabytes/sec	File Size in MB
100	5.75	100
256	5.74	100
512	5.74	100
1024	5.73	100
5120	5.74	100

Table 2-119. DM6467 MMC DMA Write Performance Values – RT

Buffer Size In KB	Megabytes/sec	File Size in MB
100	0.40	100
256	0.40	100
512	0.40	100
1024	0.40	100
5120	0.40	100

2.9.5 DM365 Performance and Benchmarks

2.9.5.1 MMC DMA Low Latency Desktop Preemption

The performance numbers were captured using the following:

- MMCSD Card used for testing = Kingston 2 GB, 120x
- File system used = EXT2
- ARM clock frequency and DDR frequency = 297 MHz

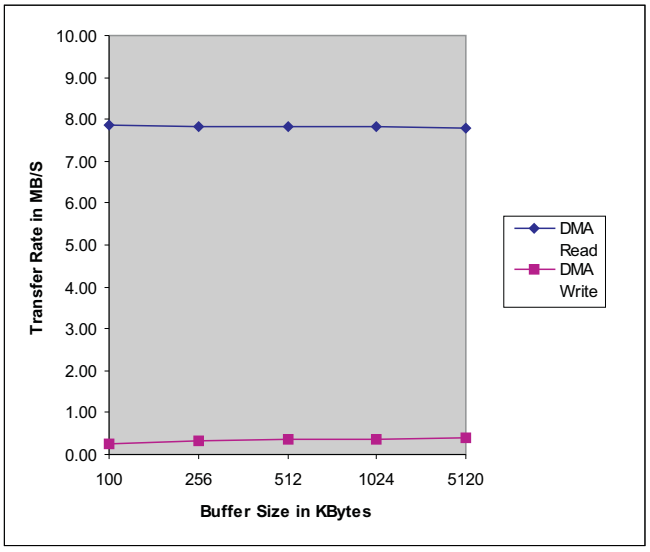

Figure 2-51. DM365 MMC DMA Performance – LLD

Table 2-120. DM365 MMC DMA Read Performance Values – LLD

Buffer Size In KB	Megabytes/sec	File Size in MB
100	7.86	100
256	7.84	100
512	7.81	100
1024	7.83	100
5120	7.79	100

Table 2-121. DM365 MMC DMA Write Performance Values – LLD

Buffer Size In KB	Megabytes/sec	File Size in MB
100	0.27	100
256	0.34	100
512	0.37	100
1024	0.38	100
5120	0.39	100

2.9.5.2 MMC DMA Real Time Preemption

The performance numbers were captured using the following:

- MMCSD Card used for testing = SanDisk 2GB, 120x
- File system used = EXT2 File System
- ARM clock frequency and DDR frequency = 297 MHz

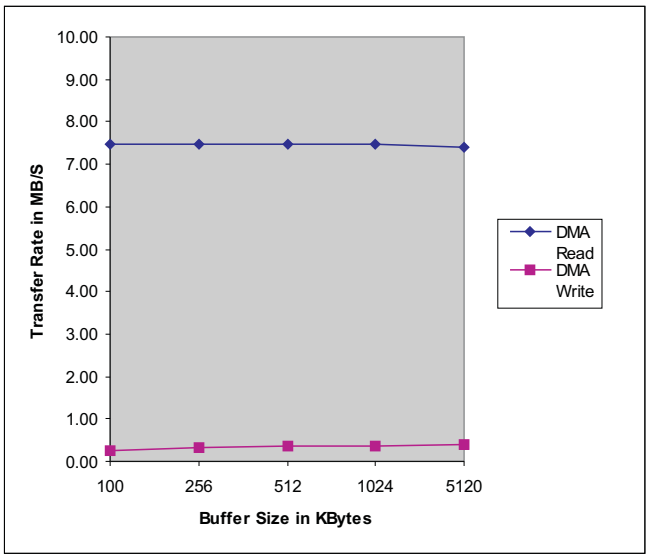

Figure 2-52. DM365 MMC DMA Performance – RT

Table 2-122. DM365 MMC DMA Read Performance Values – RT

Buffer Size In KB	Megabytes/sec	File Size in MB
100	7.49	100
256	7.47	100
512	7.48	100
1024	7.46	100
5120	7.41	100

Table 2-123. DM365 MMC DMA Write Performance Values – RT

Buffer Size In KB	Megabytes/sec	File Size in MB
100	0.26	100
256	0.32	100
512	0.35	100
1024	0.36	100
5120	0.38	100

2.10 NAND Driver

The NAND driver is implemented as a block driver, compliant with the standard MTD driver. Figure 2-53 shows the driver architecture. It supports various NAND Flash chips. (See drivers/mtd/nand/nand_ids.h.) The NAND driver is encapsulated by the MTD block/char driver. It creates the device MTDBLOCK nodes (/dev/mtdblock0, /dev/mtdblock1, etc.) and MTDCHAR nodes (/dev/mtd0, /dev/mtd1, etc.) for user space access.

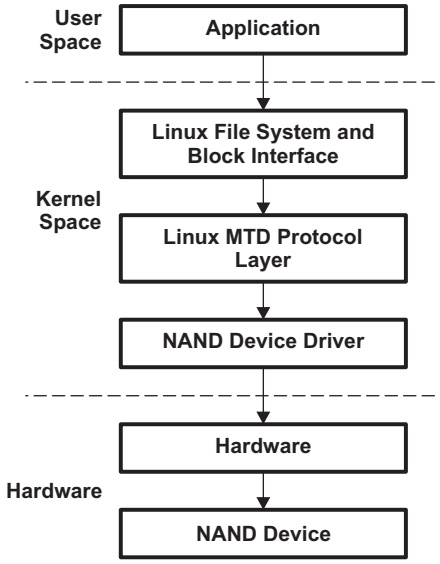

Figure 2-53. Linux Kernel NAND Driver Architecture

2.10.1 Support and Constraints

Features not supported The rmmod command to unlink the NAND module

Constraints

- Mount of a NAND partition fails after performing NAND write with the –j option.
- Read performance seems to be very low when compared with LSP1.3 (DM644x only)

Supported system calls open(), close(), write(), read(), fwrite(), fread(), ioctl()

Supported IOCTLs See Table 2-124.

Table 2-124. Supported NAND Driver IOCTLs

Constant	Description
MEMGETREGIONCOUNT	Return number of erase block regions.
MEMGETINFO	Get layout and capabilities.
MEMERASE	Erase flash blocks.
MEMWRITEOOB	Write out-of-band (OOB) info (ECC).
MEMREADOOB	Read out-of-band (OOB) info (ECC).
MEMLOCK	Lock flash blocks to disallow changes.
MEMUNLOCK	Unlock flash to allow changes.
MEMSETOOBSEL	Set default OOB info.
MEMGETOOBSEL	Get OOB info.
MEMGETBADBLOCK	Check whether the specified block is bad.
MEMSETBADBLOCK	Set block as bad.

2.10.2 DM644x Performance and Benchmarks

2.10.2.1 NAND Polled Low Latency Desktop Preemption

The performance numbers were captured using the following:

- The NAND device tested = Samsung 64MB (Small Block)
- ARM clock frequency = 297 MHz
- Linux block level read/write (/dev/mtdblock3)
- File system = YAFFS2

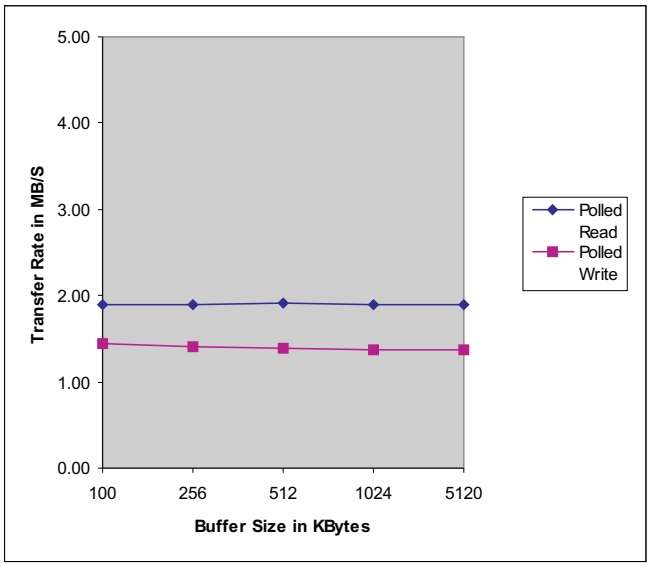

Figure 2-54. DM644x NAND Polled Performance – LLD

Table 2-125. DM644x NAND Polled Read Performance Values – LLD

Buffer Size In KB	Polled Mode	
	Megabytes/sec	File Size in MB
100	1.90	10
256	1.90	10
512	1.91	10
1024	1.90	10
5120	1.89	10

Table 2-126. DM644x NAND Polled Write Performance Values – LLD

Buffer Size In KB	Polled Mode	
	Megabytes/sec	File Size in MB
100	1.44	10
256	1.42	10
512	1.39	10
1024	1.38	10
5120	1.37	10

2.10.2.2 NAND Polled Real Time Preemption

The performance numbers were captured using the following:

- The NAND device tested = Samsung 64MB (Small Block)
- ARM clock frequency = 297 MHz
- Linux block level read/write (/dev/mtdblock3)
- File system = YAFFS2

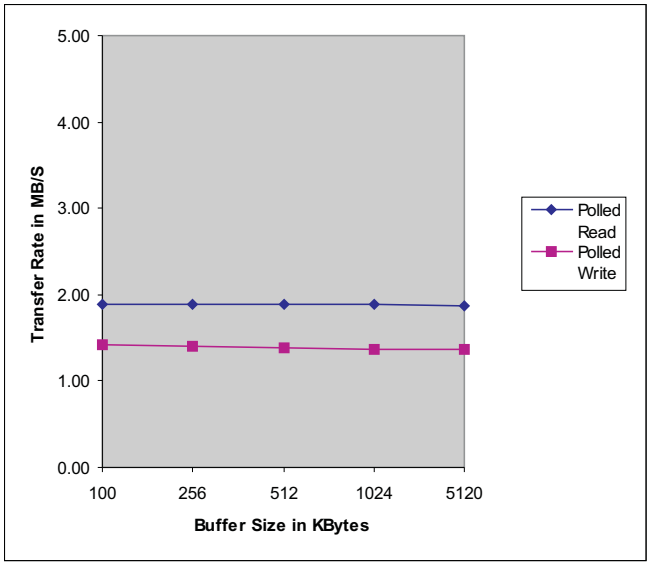

Figure 2-55. DM644x NAND Polled Performance – RT

Table 2-127. DM644x NAND Polled Read Performance Values – RT

Buffer Size In KB	Polled Mode	
	Megabytes/sec	File Size in MB
100	1.88	10
256	1.88	10
512	1.88	10
1024	1.88	10
5120	1.87	10

Table 2-128. DM644x NAND Polled Write Performance Values – RT

Buffer Size In KB	Polled Mode	
	Megabytes/sec	File Size in MB
100	1.43	10
256	1.40	10
512	1.38	10
1024	1.37	10
5120	1.36	10

2.10.3 DM355 Performance and Benchmarks

2.10.3.1 NAND Polled Low Latency Desktop Preemption

The performance numbers were captured using the following:

- The NAND device tested = Micron 512MB (Big Block)
- ARM clock frequency = 216 MHz
- Linux block level read/write (/dev/mtdblock3)
- File system = YAFFS2

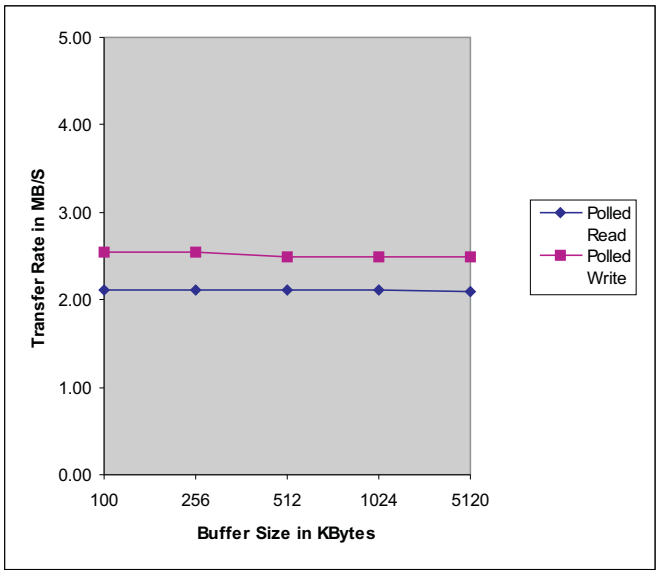

Figure 2-56. DM355 NAND Polled Performance – LLD

Table 2-129. DM355 NAND Polled Read Performance Values – LLD

Buffer Size In KB	Polled Mode	
	Megabytes/sec	File Size in MB
100	2.11	100
256	2.10	100
512	2.10	100
1024	2.11	100
5120	2.10	100

Table 2-130. DM355 NAND Polled Write Performance Values – LLD

Buffer Size In KB	Polled Mode	
	Megabytes/sec	File Size in MB
100	2.55	100
256	1.54	100
512	2.49	100
1024	2.50	100
5120	2.50	100

2.10.3.2 NAND Polled Real Time Preemption

The performance numbers were captured using the following:

- The NAND device tested = Micron 512MB (Big Block)
- ARM clock frequency = 216 MHz
- Linux block level read/write (/dev/mtdblock3)
- File system = YAFFS2

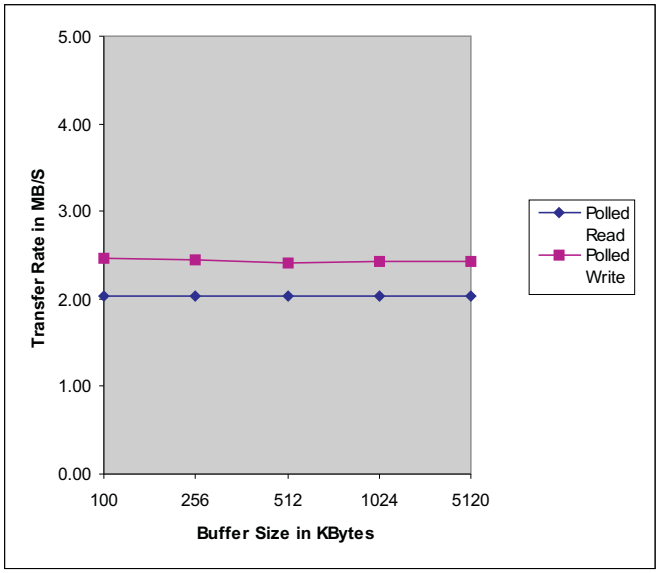

Figure 2-57. DM355 NAND Polled Performance – RT

Table 2-131. DM355 NAND Polled Read Performance Values – RT

Buffer Size In KB	Polled Mode	
	Megabytes/sec	File Size in MB
100	2.04	100
256	2.03	100
512	2.03	100
1024	2.03	100
5120	2.03	100

Table 2-132. DM355 NAND Polled Write Performance Values – RT

Buffer Size In KB	Polled Mode	
	Megabytes/sec	File Size in MB
100	2.46	100
256	2.44	100
512	2.41	100
1024	2.42	100
5120	2.42	100

2.10.4 DM6467 Performance and Benchmarks

2.10.4.1 NAND Polled Low Latency Desktop Preemption

The performance numbers were captured using the following:

- The NAND device tested = ST NAND01GB
- ARM clock frequency = 297 MHz
- Linux block level read/write (/dev/mtdblock3)
- File system = YAFFS2

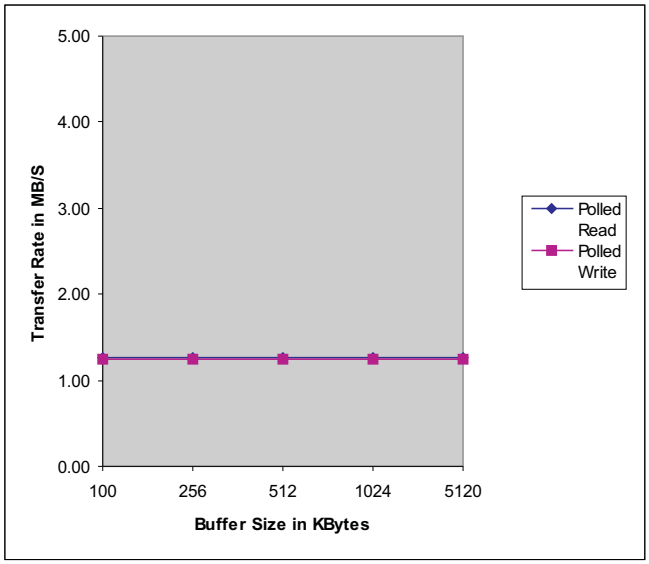

Figure 2-58. DM6467 NAND Polled Performance – LLD

Table 2-133. DM6467 NAND Polled Read Performance Values – LLD

Buffer Size In KB	Polled Mode	
	Megabytes/sec	File Size in MB
100	1.26	10
256	1.26	10
512	1.26	10
1024	1.26	10
5120	1.26	10

Table 2-134. DM6467 NAND Polled Write Performance Values – LLD

Buffer Size In KB	Polled Mode	
	Megabytes/sec	File Size in MB
100	1.25	10
256	1.25	10
512	1.25	10
1024	1.25	10
5120	1.25	10

2.10.4.2 NAND Polled Real Time Preemption

The performance numbers were captured using the following:

- The NAND device tested = Micron 512MB (Big Block)
- ARM clock frequency = 297 MHz
- Linux block level read/write (/dev/mtdblock3)
- File system = YAFFS2

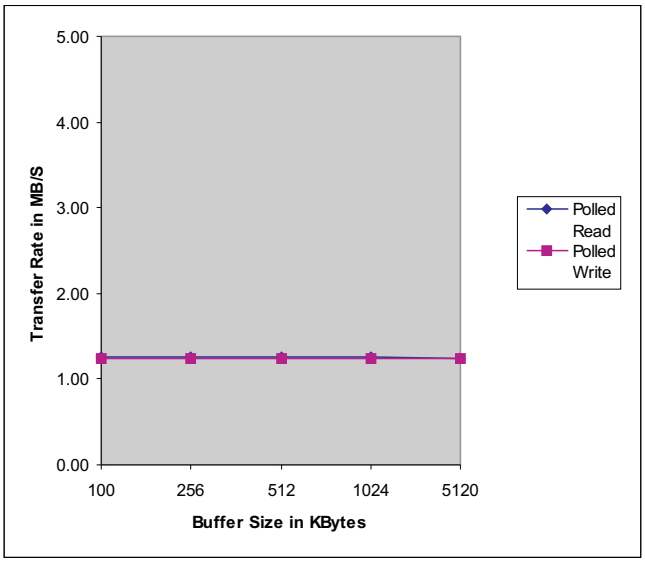

Figure 2-59. DM6467 NAND Polled Performance – RT

Table 2-135. DM6467 NAND Polled Read Performance Values – RT

Buffer Size In KB	Polled Mode	
	Megabytes/sec	File Size in MB
100	1.25	10
256	1.25	10
512	1.25	10
1024	1.25	10
5120	1.25	10

Table 2-136. DM6467 NAND Polled Write Performance Values – RT

Buffer Size In KB	Polled Mode	
	Megabytes/sec	File Size in MB
100	1.24	10
256	1.25	10
512	1.25	10
1024	1.25	10
5120	1.24	10

2.10.5 DM365 Performance and Benchmarks

2.10.5.1 NAND Polled Low Latency Desktop Preemption

The performance numbers were captured using the following:

- The NAND device tested = MICRON 2 GB SLC NAND
- ARM clock frequency = 297 MHz
- Linux block level read/write (/dev/mtd3)
- File system = YAFFS2

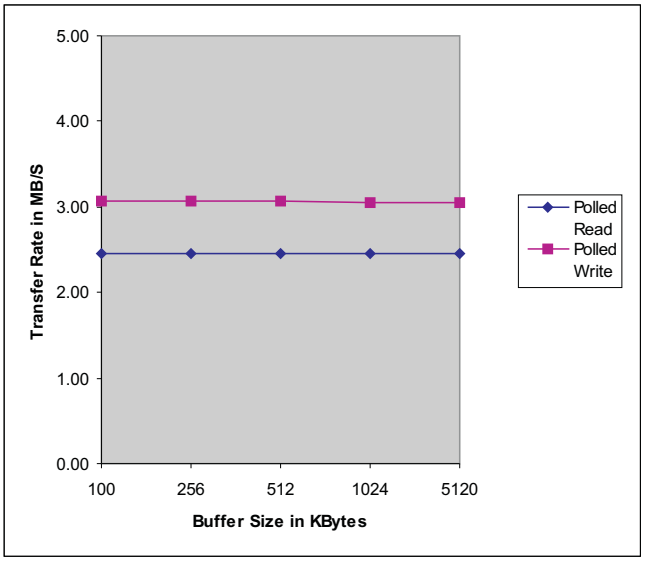

Figure 2-60. DM365 NAND Polled Performance – LLD

Table 2-137. DM365 NAND Polled Read Performance Values – LLD

Buffer Size In KB	Polled Mode	
	Megabytes/sec	File Size in MB
100	2.46	100
256	2.45	100
512	2.46	100
1024	2.45	100
5120	2.45	100

Table 2-138. DM365 NAND Polled Write Performance Values – LLD

Buffer Size In KB	Polled Mode	
	Megabytes/sec	File Size in MB
100	3.07	100
256	3.07	100
512	3.06	100
1024	3.06	100
5120	3.05	100

2.10.5.2 NAND Polled Real Time Preemption

The performance numbers were captured using the following:

- The NAND device tested = MICRON 2 GB SLC NAND
- ARM clock frequency = 297 MHz
- Linux block level read/write (/dev/mtd3)
- File system = YAFFS2

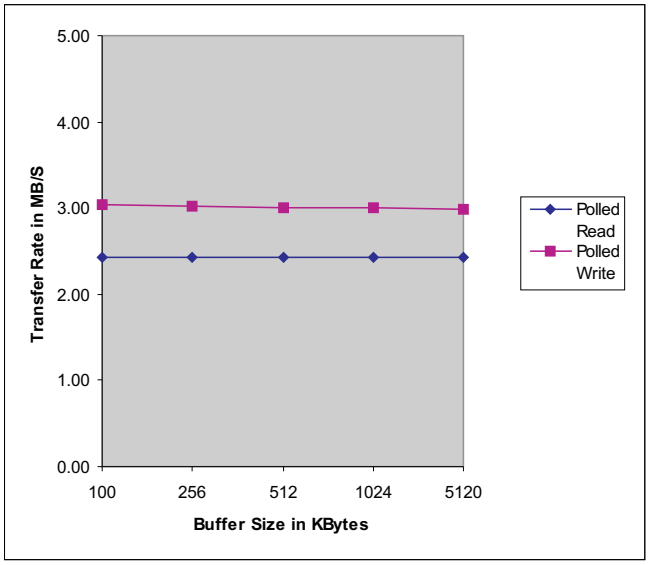

Figure 2-61. DM365 NAND Polled Performance – RT

Table 2-139. DM365 NAND Polled Read Performance Values – RT

Buffer Size In KB	Polled Mode	
	Megabytes/sec	File Size in MB
100	2.43	100
256	2.42	100
512	2.42	100
1024	2.42	100
5120	2.42	100

Table 2-140. DM365 NAND Polled Write Performance Values – RT

Buffer Size In KB	Polled Mode	
	Megabytes/sec	File Size in MB
100	3.04	100
256	3.01	100
512	3.01	100
1024	3.00	100
5120	2.99	100

2.11 NOR Driver

The NOR is accessed through the Asynchronous External Memory Interface (EMIFA). ARM ROM supports booting of the DM644x ARM processor from NOR flash. The NOR driver is compliant with the standard MTD driver. The NOR driver creates the device nodes for user space access (/dev/mtdblock0, /dev/mtdblock1,/dev/mtd0,/dev/mtd1 and so on.)

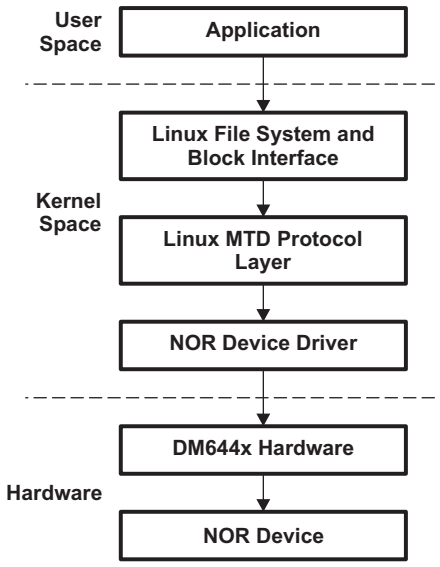

Figure 2-62. Linux Kernel NOR Driver Architecture

The NOR Driver is not applicable to the DM6467, because the EVMs do not have NOR support.

On the DM6446, the NOR Driver supports the following features:

- Supports 16MB AMD AM29LV256M NOR Flash and Intel I128P30T StrataFlash NOR Flash chips.
- Supports Read/Write/Erase from/to NOR Flash
- Polled mode of transfer
- NOR is tested with JFFS2 file system
- NOR has been divided into four partitions:
  1. 128 KB Read/Write partition for U-Boot
  2. 128 KB Read-only partition for environment variables
  3. 4 MB Read/Write partition for Linux
  4. Remaining space is used for file system and others (Read/Write)

Features Not Supported	The rmmod command to unlink the NOR module.
Constraints	None
Supported System Calls	open(), close(), write(), read(), pwrite(), pread(), ioctl()
Supported IOCTLs	None

2.11.1 DM644x Performance and Benchmarks

2.11.1.1 NOR Low Latency Desktop Preemption

The performance numbers were captured using the following:

- NOR device tested = Intel 128P30T (16MB)
- ARM clock frequency = 297 MHz
- Linux block level read/write (/dev/mtdblock3)
- File system = JFFS2

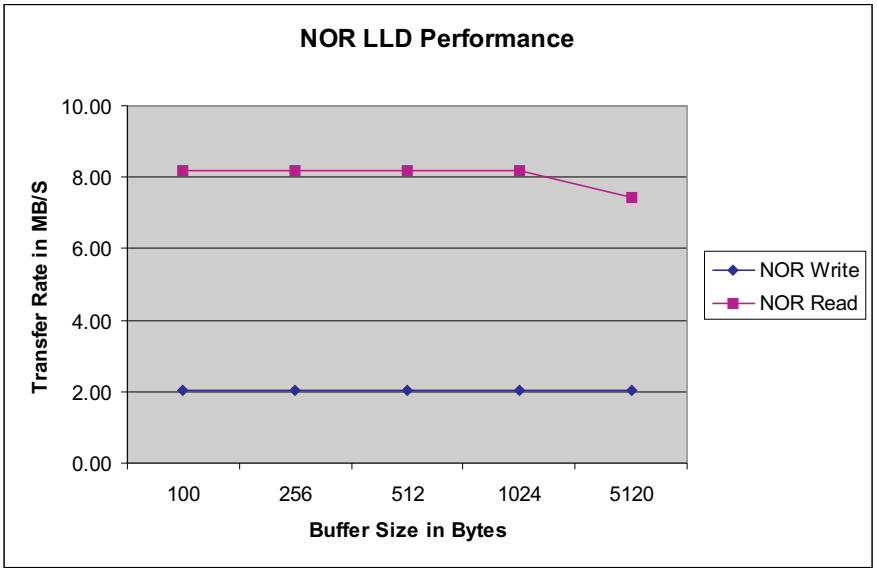

Figure 2-63. DM644x NOR Polled Performance – LLD

Table 2-141. DM644x NOR Polled Read Performance Values – LLD

Buffer Size In KB	Polled Mode	
	Megabytes/sec	File Size in MB
100	8.18	10
256	8.18	10
512	8.18	10
1024	8.17	10
5120	7.43	10

Table 2-142. DM644x NOR Polled Write Performance Values – LLD

Buffer Size In KB	Polled Mode	
	Megabytes/sec	File Size in MB
100	2.03	10
256	2.04	10
512	2.03	10
1024	2.04	10
5120	2.03	10

2.11.1.2 NOR Real Time Preemption

The performance numbers were captured using the following:

- NOR device tested = Intel 128P30T (16MB)
- ARM clock frequency = 297 MHz
- Linux block level read/write (/dev/mtdblock3)
- File system = JFFS2

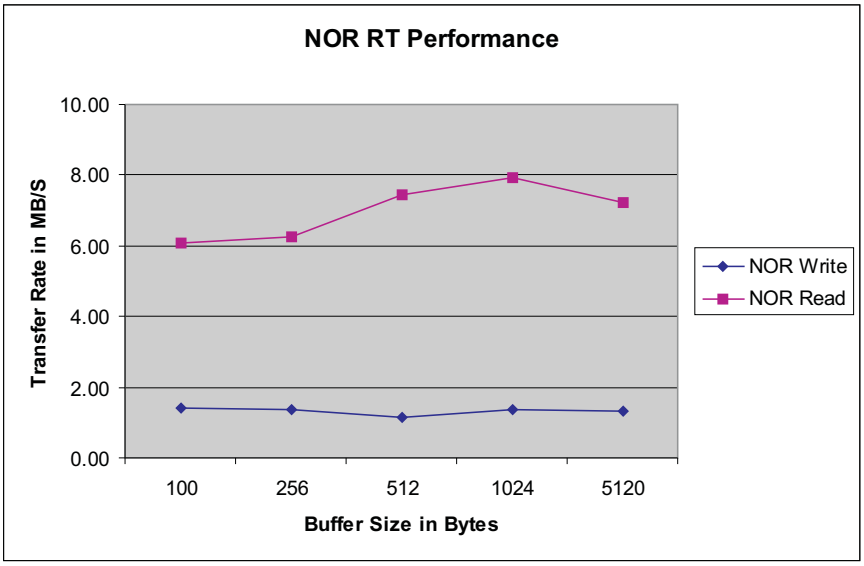

Figure 2-64. DM644x NOR Polled Performance – RT

Table 2-143. DM644x NOR Polled Read Performance Values – RT

Buffer Size In KB	Polled Mode	
	Megabytes/sec	File Size in MB
100	6.10	10
256	6.26	10
512	7.46	10
1024	7.91	10
5120	7.22	10

Table 2-144. DM644x NOR Polled Write Performance Values – RT

Buffer Size In KB	Polled Mode	
	Megabytes/sec	File Size in MB
100	1.43	10
256	1.35	10
512	1.13	10
1024	1.36	10
5120	1.34	10

2.12 UART Driver

The UART peripheral is based on the industry standard TL16C550 asynchronous communications element, which is a functional upgrade of the TL16C450. The UART driver is implemented as a serial driver, and can be accessed from user space as /dev/ttyS0 device.

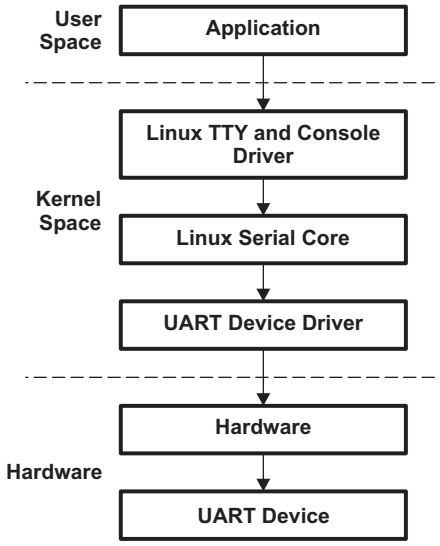

Figure 2-65. Linux Kernel UART Driver Architecture

The UART Driver supports the following features:

- Supports UART0
- Supports UART1 for CIR on DM6467

2.12.1 Support and Constraints

Features Not Supported	None
Constraints	• UART data bit and parity setting fails • Continuous UART input overruns under moderate load
Supported System Calls	open(), close(), read(),write(), ioctl()
Supported IOCTLs	See Table 2-145

Table 2-145. UART Supported IOCTLs

CONSTANT	DESCRIPTION
TIOCGSERIAL	Get device parameters from the UART (for example: port type, port num, baud rate, base divisor, etc.)
TIOCSSERIAL	Set UART device parameters (for example: port type, port num, baud rate, base divisor, etc.)

2.13 I²C Driver

The I²C peripheral is compliant with the Philips Semiconductor I²C-bus specification version 2.1. The I²C driver is implemented as a serial driver. The I²C driver can be accessed from the user space as /dev/i2c/0.

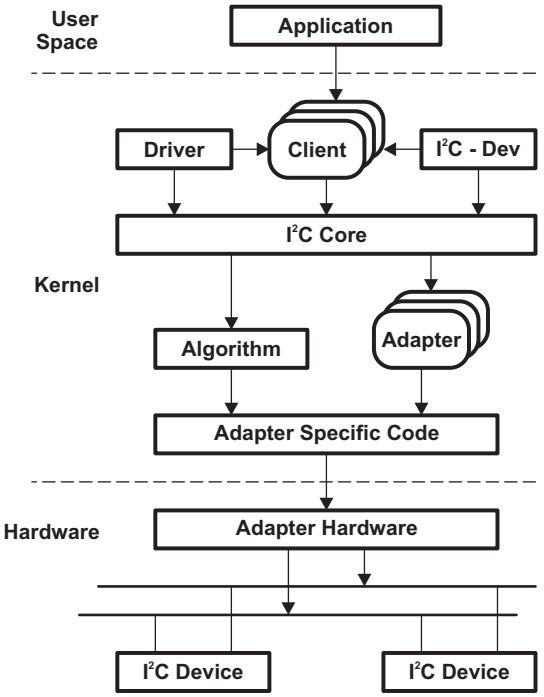

Figure 2-66. Linux Kernel I²C Driver Architecture

The I²C Driver supports the following features:

- 7-Bit Addressing
- Fast mode
- Interrupt mode

2.13.1 Support and Constraints

Features Not Supported

- 10-Bit Addressing
- DMA mode

Constraints

- DM644x and DM355 - Due to hardware limitations of MSP430, I²C runs at 20 KHz. This may cause intermittent failures of some I²C functions (example, I²C write).
- DM6467 - Default I²C frequency is 100 KHz. 400 KHz is not supported as there are failures observed with Video encoders/decoders write operation.

Supported System Calls open(), close(), write(), read(), ioctl()

Supported IOCTLs The I²C Driver supports the IOCTLs listed in Table 2-146.

Table 2-146. I²C Driver Supported IOCTLs

CONSTANT	DESCRIPTION
I2C_SLAVE_FORCE	Change slave address. Slave address is 7 or 10 bits. This changes the address, even if it is already taken.
I2C_TENBIT	7- or 10-bit address. (value = 0 for 7 bits; value != 0 for 10 bits.)
I2C_FUNCS	Get the adapter functionality.
I2C_RDWR	Combined R/W transfer (one stop only).

2.13.2 DM644x Performance and Benchmarks

2.13.2.1 I²C Low Latency Desktop Preemption

The performance numbers were captured using the following:

- Slave used: EEPROM (ATMEL 642 25640A)
- Bus frequency = 20 KHz
- ARM frequency = 297 MHz

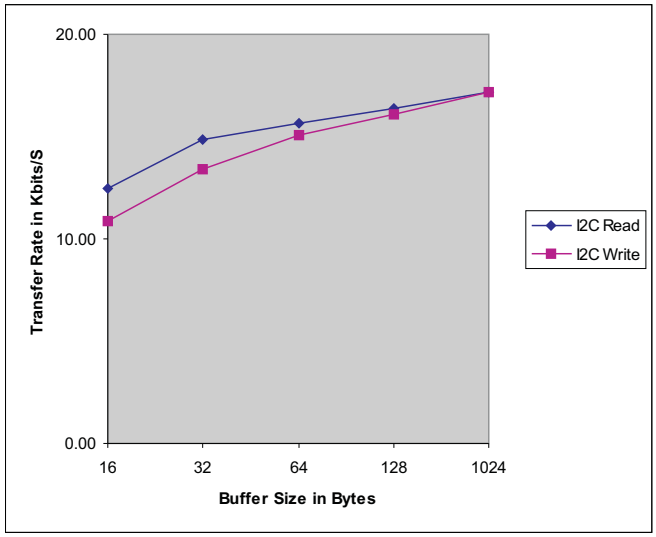

Figure 2-67. DM644x I²C Performance – LLD

Table 2-147. DM644x I²C Read Performance – LLD

Buffer Size In Bytes	Data Rate Kbits/sec	File Size in Bytes
16	12.50	1024
32	14.84	1024
64	15.63	1024
128	16.41	1024
1024	17.19	1024

Table 2-148. DM644x I²C Write Performance – LLD

Buffer Size In Bytes	Data Rate Kbits/sec	File Size in Bytes
16	10.90	1024
32	14.84	1024
64	15.63	1024
128	16.41	1024
1024	17.19	1024

2.13.2.2 I²C Real Time Preemption

The performance numbers were captured using the following:

- Bus frequency = 20 KHz
- ARM frequency = 297 MHz

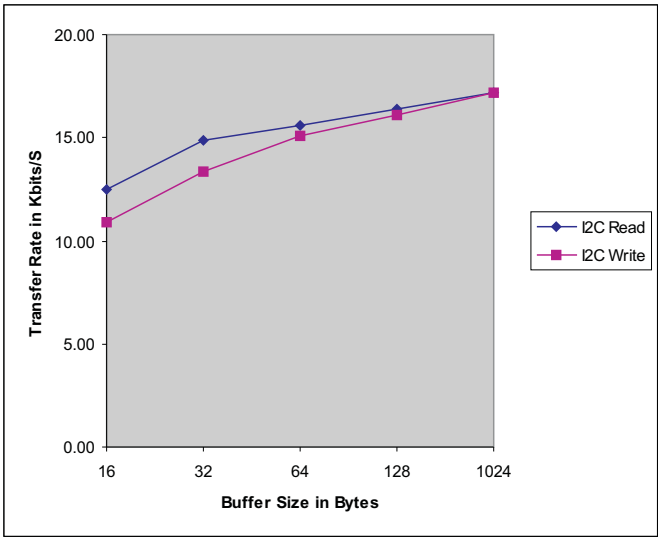

Figure 2-68. DM644x I²C Performance – RT

Table 2-149. DM644x I²C Read Performance – RT

Buffer Size In Bytes	Data Rate Kbits/sec	File Size in Bytes
16	10.88	1024
32	13.36	1024
64	15.08	1024
128	16.12	1024
1024	17.15	1024

Table 2-150. DM644x I²C Write Performance – RT

Buffer Size In Bytes	Data Rate Kbits/sec	File Size in Bytes
16	10.88	1024
32	13.36	1024
64	15.08	1024
128	16.12	1024
1024	17.15	1024

2.13.3 DM355 Performance and Benchmarks

2.13.3.1 I²C Low Latency Desktop Preemption

The performance numbers were captured using the following:

- Slave used: Video Codec (TVP5146)
- Bus frequency = 20 KHz
- ARM frequency = 216 MHz

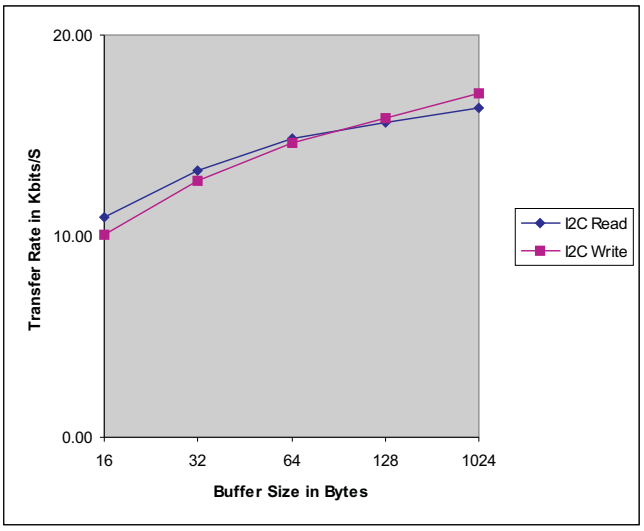

Figure 2-69. DM355 I²C Performance – LLD

Table 2-151. DM355 I²C Read Performance – LLD

Buffer Size In Bytes	Data Rate Kbits/sec	File Size in Bytes
16	10.94	1024
32	13.34	1024
64	14.84	1024
128	15.63	1024
1024	16.41	1024

Table 2-152. DM355 I²C Write Performance – LLD

Buffer Size In Bytes	Data Rate Kbits/sec	File Size in Bytes
16	10.09	1024
32	12.73	1024
64	14.65	1024
128	15.85	1024
1024	17.07	1024

2.13.3.2 I²C Real Time Preemption

The performance numbers were captured using the following:

- Slave used: Video Codec (TVP5146)
- Bus frequency = 20 KHz
- ARM frequency = 216 MHz

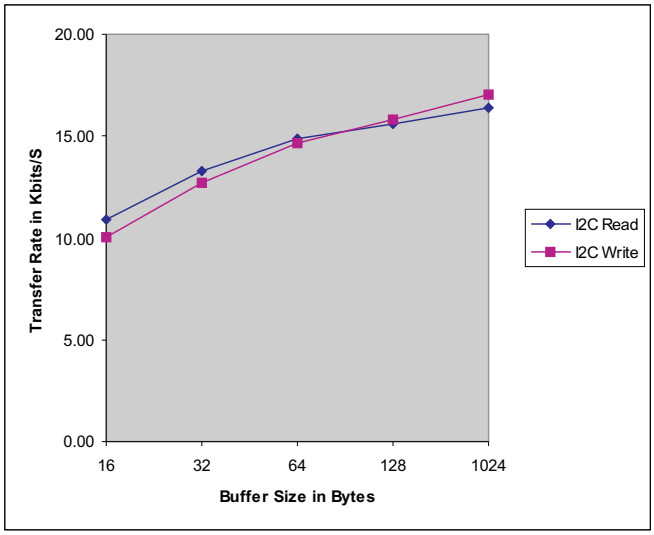

Figure 2-70. DM355 I²C Performance – RT

Table 2-153. DM355 I²C Read Performance – RT

Buffer Size In Bytes	Data Rate Kbits/sec	File Size in Bytes
16	10.94	1024
32	13.28	1024
64	14.84	1024
128	15.63	1024
1024	16.41	1024

Table 2-154. DM355 I²C Write Performance – RT

Buffer Size In Bytes	Data Rate Kbits/sec	File Size in Bytes
16	10.05	1024
32	12.70	1024
64	14.63	1024
128	15.84	1024
1024	17.06	1024

2.13.4 DM6467 Performance and Benchmarks

2.13.4.1 I²C Low Latency Desktop Preemption

The performance numbers were captured using the following:

- Slave used: EEPROM (ATMEL 642 25640A)
- Bus frequency = 100 KHz
- ARM frequency = 297 MHz

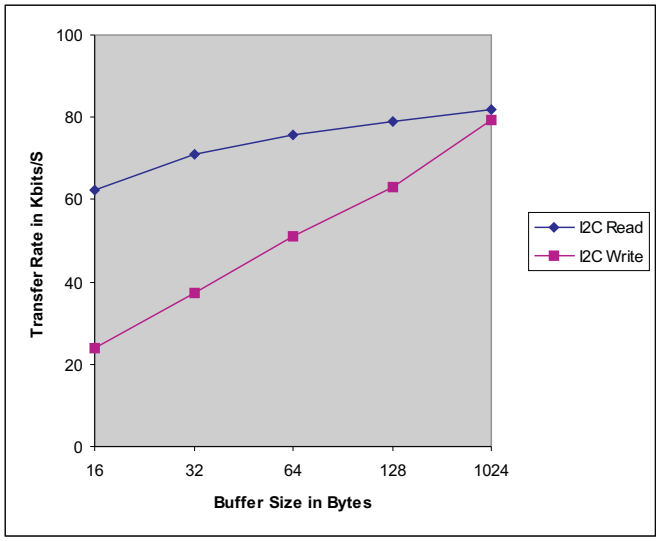

Figure 2-71. DM6467 I²C Performance – LLD

Table 2-155. DM6467 I²C Read Performance – LLD

Buffer Size In Bytes	Data Rate Kbits/sec	File Size in Bytes
16	62.50	1024
32	71.09	1024
64	75.78	1024
128	78.91	1024
1024	82.03	1024

Table 2-156. DM6467 I²C Write Performance – LLD

Buffer Size In Bytes	Data Rate Kbits/sec	File Size in Bytes
16	24.00	1024
32	37.15	1024
64	51.18	1024
128	63.13	1024
1024	79.35	1024

2.13.4.2 I²C Real Time Preemption

The performance numbers were captured using the following:

- Slave used: EEPROM (ATMEL 642 25640A)
- Bus frequency = 100 KHz
- ARM frequency = 297 MHz

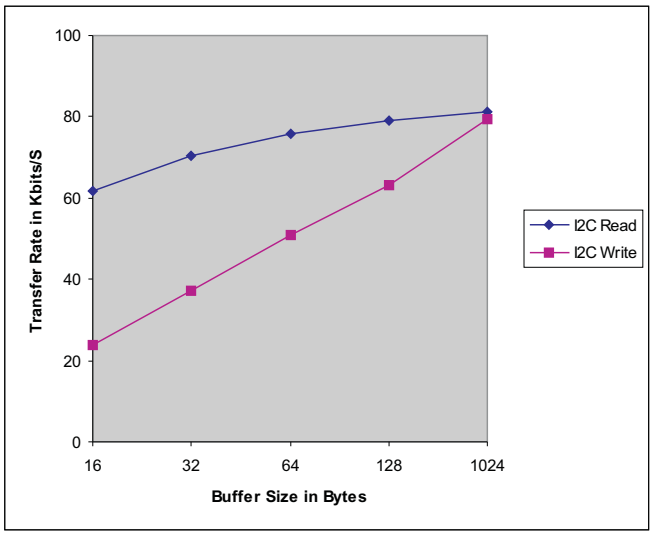

Figure 2-72. DM6467 I²C Performance – RT

Table 2-157. DM6467 I²C Read Performance – RT

Buffer Size In Bytes	Data Rate Kbits/sec	File Size in Bytes
16	61.72	1024
32	70.31	1024
64	75.78	1024
128	78.91	1024
1024	81.25	1024

Table 2-158. DM6467 I²C Write Performance – RT

Buffer Size In Bytes	Data Rate Kbits/sec	File Size in Bytes
16	23.89	1024
32	37.02	1024
64	51.08	1024
128	63.04	1024
1024	79.34	1024

2.13.5 DM365 Performance and Benchmarks

2.13.5.1 I²C Low Latency Desktop Preemption

The performance numbers were captured using the following:

- Slave used: EEPROM (ATMEL 642 25640A)
- Bus frequency = 20 KHz
- ARM frequency = 297 MHz

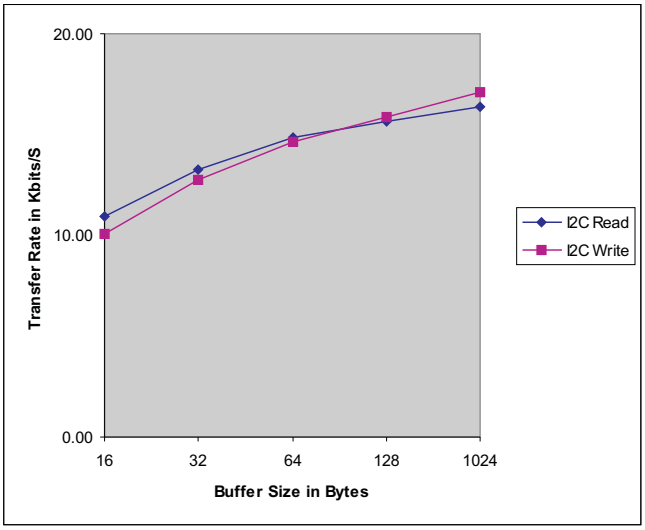

Figure 2-73. DM365 I²C Performance – LLD

Table 2-159. DM365 I²C Read Performance – LLD

Buffer Size In Bytes	Data Rate Kbits/sec	File Size in Bytes
16	10.94	1024
32	13.28	1024
64	14.84	1024
128	15.63	1024
1024	16.41	1024

Table 2-160. DM365 I²C Write Performance – LLD

Buffer Size In Bytes	Data Rate Kbits/sec	File Size in Bytes
16	10.09	1024
32	12.73	1024
64	14.65	1024
128	15.85	1024
1024	71.07	1024

2.13.5.2 I²C Real Time Preemption

The performance numbers were captured using the following:

- Bus frequency = 20 KHz
- ARM frequency = 297 MHz

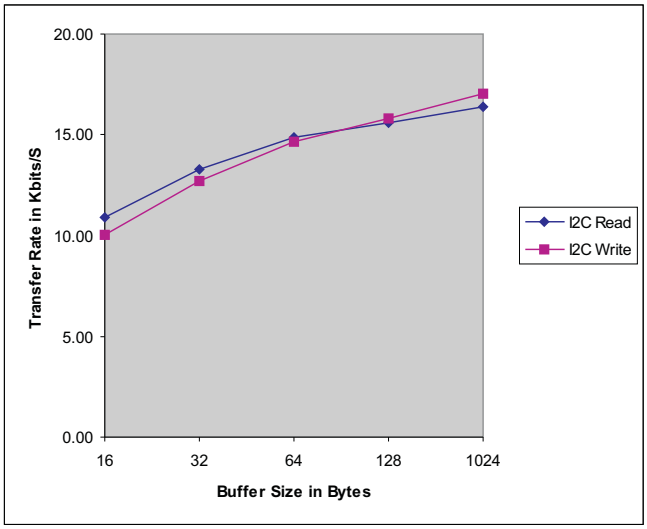

Figure 2-74. DM365 I²C Performance – RT

Table 2-161. DM365 I²C Read Performance – RT

Buffer Size In Bytes	Data Rate Kbits/sec	File Size in Bytes
16	10.94	1024
32	13.28	1024
64	14.84	1024
128	15.63	1024
1024	16.41	1024

Table 2-162. DM365 I²C Write Performance – RT

Buffer Size In Bytes	Data Rate Kbits/sec	File Size in Bytes
16	10.05	1024
32	12.70	1024
64	14.63	1024
128	15.84	1024
1024	17.06	1024

2.14 SPI Driver

SPI driver is implemented as block driver and compliant with standard MTD driver. It supports various EEPROM devices. The SPI driver creates device node for user space access (example, /dev/mtd0). The SPI control driver supports serial communication with other SPI devices in a 3-pin or 4-pin mode.

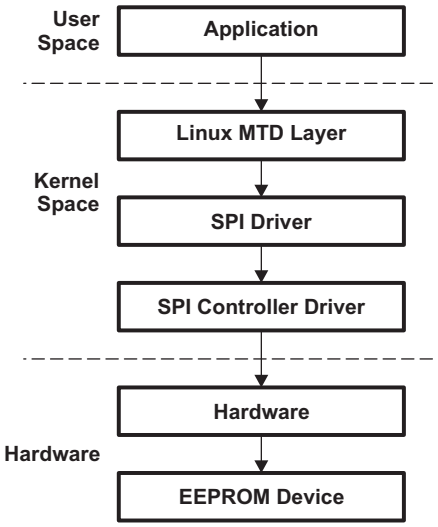

Figure 2-75. Linux Kernel SPI Driver Architecture

The SPI Driver supports the following features:

- DMA and PIO (bit-bang) modes are supported

2.14.1 Support and Constraints

Features Not Supported	/dev/spi entry is not present.
Constraints	In order to access SPI driver, you need to access /dev/mtd0 on DM6467 and /dev/mtd5 on DM355 and DM365
Supported System Calls	open(), close(), write() and read()
Supported IOCTLs	None

2.14.2 DM355 Performance and Benchmarks

2.14.2.1 SPI Driver Read/Write Low Latency Desktop Preemption

The performance numbers were captured using the following:

- SPI frequency = 2 MHz
- SPI mode = Polled
- Slave used = EEPROM

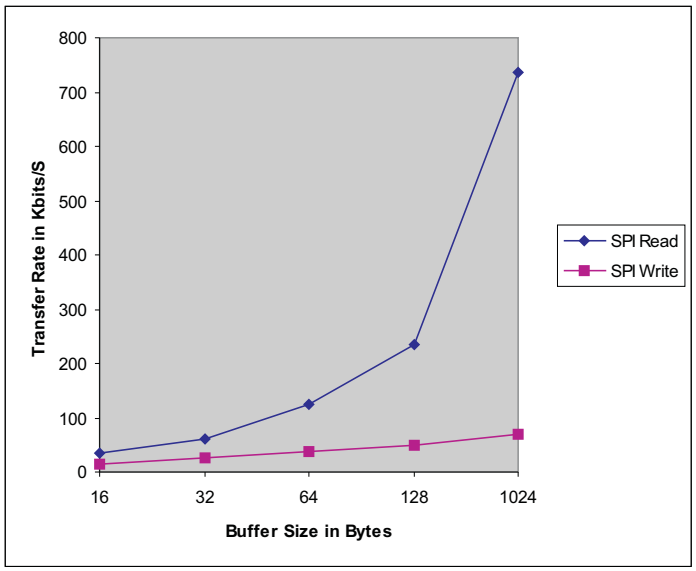

Figure 2-76. DM355 SPI Read / Write Performance – LLD

Table 2-163. DM355 SPI Read/Write Performance Values – LLD

Appplication Buffer Size	Write		Read	
	Kbits/Sec	Total Buffer Size	Kbits/Sec	Total Buffer Size
16	15.58	1024	34.39	1024
32	25.73	1024	60.40	1024
64	37.96	1024	126.05	1024
128	50.35	1024	235.16	1024
1024	70.59	1024	737.46	1024

2.14.2.2 SPI Read/Write Real Time Preemption

The performance numbers were captured using the following:

- SPI frequency = 2 MHz
- SPI mode = Polled
- Slave used = EEPROM

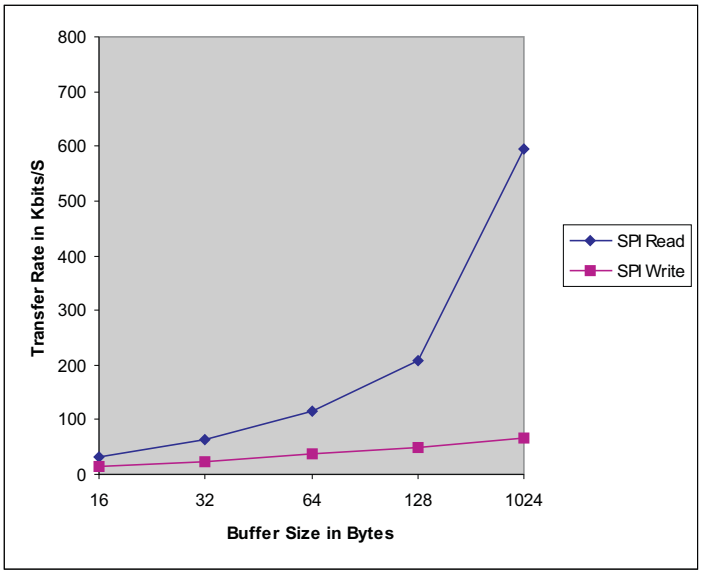

Figure 2-77. DM355 SPI Read/Write Performance – RT

Table 2-164. DM355 SPI Read/Write Performance Values – RT

Appplication Buffer Size	Write		Read	
	Kbits/Sec	Total Buffer Size	Kbits/Sec	Total Buffer Size
16	14.71	1024	31.73	1024
32	24.40	1024	62.18	1024
64	36.13	1024	114.95	1024
128	47.77	1024	207.91	1024
1082	67.20	1024	595.59	1024

2.14.3 DM6467 Performance and Benchmarks

2.14.3.1 SPI Driver Read/Write Low Latency Desktop Preemption

The performance numbers were captured using the following:

- SPI frequency = 2 MHz
- SPI mode = Polled
- Slave used = EEPROM

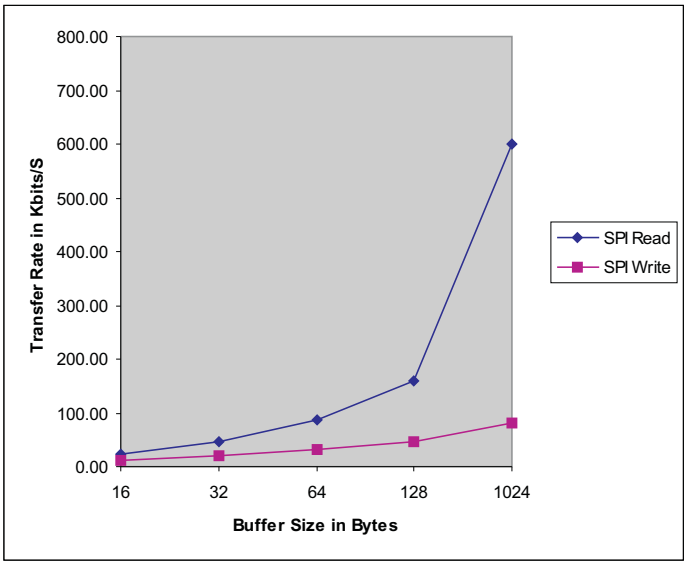

Figure 2-78. DM6467 SPI Read/Write Performance – LLD

Table 2-165. DM6467 SPI Read/Write Performance Values – LLD

Application Buffer Size	Write		Read	
	Kbits/Sec	Total Buffer Size	Kbits/Sec	Total Buffer Size
16	10.64	1024	22.19	1024
32	19.19	1024	45.09	1024
64	31.30	1024	86.31	1024
128	46.66	1024	159.15	1024
1082	81.54	1024	599.07	1024

2.14.3.2 SPI Read/Write Real Time Preemption

The performance numbers were captured using the following:

- SPI frequency = 2 MHz
- SPI mode = Polled
- Slave used = EEPROM

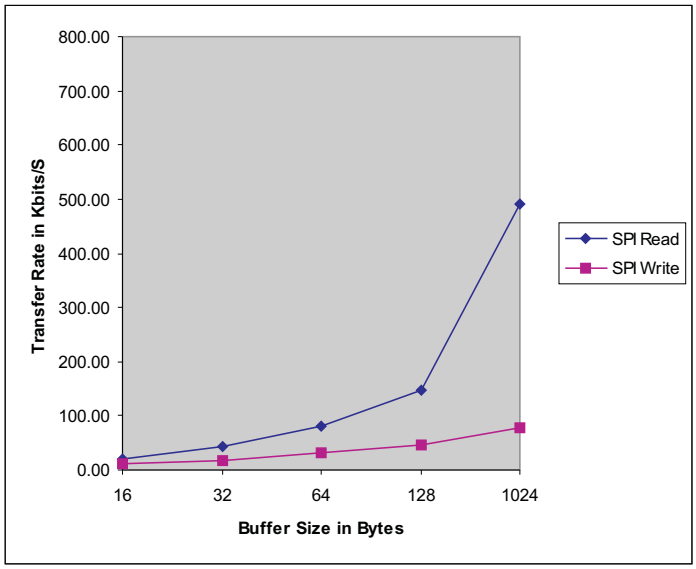

Figure 2-79. DM6467 SPI Read/Write Performance – RT

Table 2-166. DM6467 SPI Read/Write Performance Values – RT

Appplication Buffer Size	Write		Read	
	Kbits/Sec	Total Buffer Size	Kbits/Sec	Total Buffer Size
16	10.27	1024	21.15	1024
32	18.48	1024	42.98	1024
64	30.34	1024	80.74	1024
128	45.23	1024	145.98	1024
1082	79.25	1024	490.67	1024

2.14.4 DM365 Performance and Benchmarks

2.14.4.1 SPI Driver Read/Write Low Latency Desktop Preemption

The performance numbers were captured using the following:

- SPI frequency = 2 MHz
- SPI mode = Polled
- Slave used = EEPROM

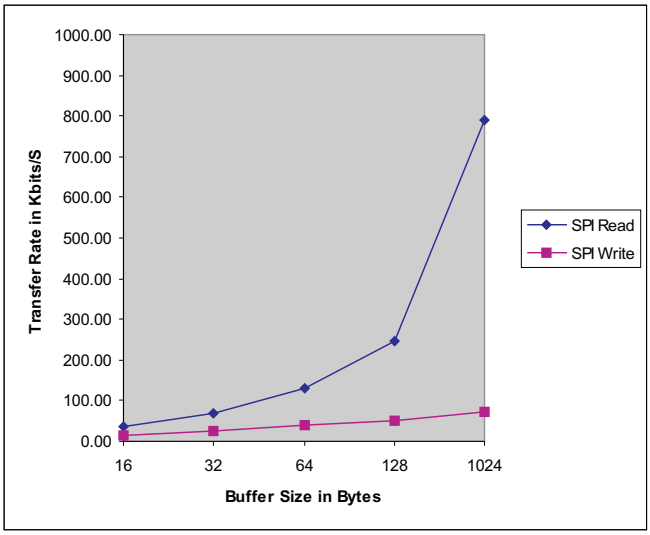

Figure 2-80. DM365 SPI Read / Write Performance – LLD

Table 2-167. DM365 SPI Read/Write Performance Values – LLD

Appplication Buffer Size	Write		Read	
	Kbits/Sec	Total Buffer Size	Kbits/Sec	Total Buffer Size
16	16.02	1024	35.23	1024
32	26.23	1024	68.93	1024
64	38.67	1024	130.03	1024
128	51.20	1024	244.89	1024
1024	71.73	1024	789.27	1024

2.14.4.2 SPI Read/Write Real Time Preemption

The performance numbers were captured using the following:

- SPI frequency = 2 MHz
- SPI mode = Polled
- Slave used = EEPROM

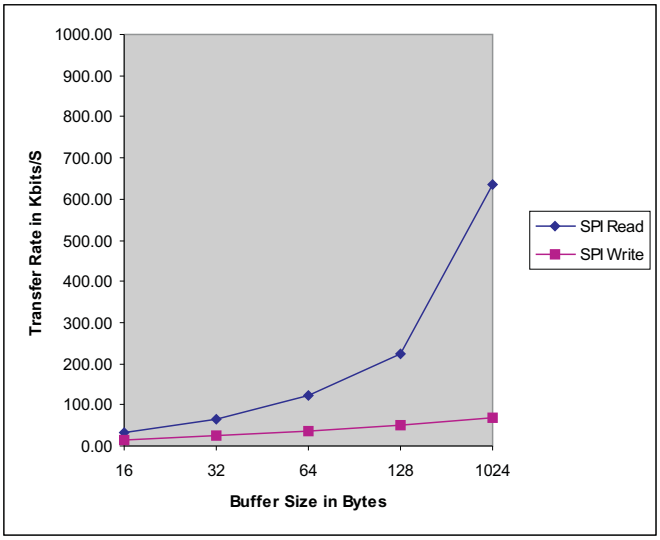

Figure 2-81. DM365 SPI Read/Write Performance – RT

Table 2-168. DM365 SPI Read/Write Performance Values – RT

Appplication Buffer Size	Write		Read	
	Kbits/Sec	Total Buffer Size	Kbits/Sec	Total Buffer Size
16	15.23	1024	33.35	1024
32	25.20	1024	65.19	1024
64	37.10	1024	121.12	1024
128	49.25	1024	222.05	1024
1024	68.93	1024	635.93	1024

2.15 TSIF Driver

The TSIF driver is implemented as a char driver. Transport Stream InterFace (TSIF) driver enables the user to send and receive MPEG-TS packet. Transport stream (MPEG-TS) is a communication protocol for audio, video, and data which is specified in MPEG-2 standard. The driver supports both parallel and serial interface for input and output streams.

The TSIF Driver supports the following features:

- Can be loaded dynamically or built statically into the Linux kernel.
- Two instances of TSIF (TSIF0,TSIF1) are present on DM6467 EVM.
- TSIF1 supports only serial mode.
- TSIF0 supports serial and parallel mode.
- Creation of 12 device nodes for each instance of TSIF.
- rx_filter0-7 for 7 PID filters and bypass mode
- tx_control – For transmit control
- transmit – For transmit data
- ATS – For ATS read/write
- Zero copy buffers
- Both 1-bit Serial and 8-bit Parallel I/F with synchronous and asynchronous mode
- Bypass mode.
- Four modes of PID filtering
- Full Manual Mode
- Semi-automatic mode - A
- Semi-automatic mode - B
- Full automatic mode
- Consequential mode
- Transmission and reception at either 13.5 MHz or 16.875 MHz in Parallel mode
- Transmission and reception at either 27 MHz, 54 MHz or 81 MHz in Serial mode
- Boundary sensing interrupts and GOP interrupts
- Specific packet detection

2.15.1 Support and Constraints

Features Not Supported

None

Constraints

- Serial async mode cannot be tested due to EVM constraint.
- Some of the Video and UART pins are muxed with TSIF. Currently the driver supports using one SD Display along with TSIF0 in parallel mode and TSIF1 in serial mode.
- When TSIF0 is configured in serial mode, some of the TSIF pins are muxed with UART0 and UART1. These muxed UART pins are not brought on the DM6467 TSIF daughter card connector provided by Spectrum Digital. Configuring TSIF0 in serial mode will have data integrity issues. To test serial mode use TSIF1 or short the resistors R406 and R413 on the daughter card (so that TSIF0 can be used in Serial mode at higher transmit frequencies).

- The Parallel async mode for TSIF0 has been tested only with the DC_P2 connector.
- Currently both the instances of TSIF (TSIF0 and TSIF1) use the same clock source. Because of this constraint, if user wants to use both instances of TSIF, the transmit clock frequency has to be same in both cases.
- Consequential interface has been tested by storing the source stream in SDRAM. Source was not taken from HPI or PCI modules.

Supported System Calls open(), close(), mmap(), ioctl()

Supported IOCTLs See Table 2-169.

Table 2-169. TSIF Driver-Supported IOCTLs

Constant	Description
TSIF_SET_PID_FILTER_CONFIG	Sets up the hardware filter according to the parameters provided.
TSIF_GET_PID_FILTER_CONFIG	Gets the hardware filter configuration
TSIF_SET_RX_CONFIG	Sets up the receiver configuration according to the parameters provided.
TSIF_GET_RX_CONFIG	Gets the receiver configuration
TSIF_SET_TX_CONFIG	Sets up the transmitter configuration according to the parameters provided.
TSIF_GET_TX_CONFIG	Gets the transmitter configuration
TSIF_START_RX	Starts the TSIF receiver
TSIF_STOP_RX	Stops the TSIF receiver
TSIF_START_TX	Starts the TSIF transmitter
TSIF_STOP_TX	Stops the TSIF transmitter
TSIF_GET_RX_BUF_STATUS	Gets the status of receive buffer
TSIF_GET_TX_BUF_STATUS	Gets the status of transmit buffer
TSIF_CONFIG_PAT	Configures the TSIF module to detect PAT
TSIF_GET_PAT_PKT	Copies the new PAT packet to the user passed buffer
TSIF_GET_PAT_CONFIG	Gets the flag which says whether PAT detection is enabled or not
TSIF_CONFIG_PMT	Configures the TSIF module to detect PMT
TSIF_GET_PMT_PKT	Copies the new PMT packet to the user passed buffer
TSIF_GET_PMT_CONFIG	Gets the flag which says whether PMT detection is enabled or not
TSIF_ENABLE_PCR	Enables PCR detection
TSIF_DISABLE_PCR	Disables PCR detection
TSIF_SET_SPCPKT_CONFIG	Enables the user to set the configuration for detecting specific packet
TSIF_GET_SPCPKT_CONFIG	Enables the user to get the configuration for detecting specific packet
TSIF_GET_READ_AVAIL	Gets the length of available data that can be read in the ring buffer
TSIF_SET_READ_AVAIL	Moves the Read Pointer of Receiver by the specified amount
TSIF_GET_WRITE_AVAIL	Gets the length of available free space in the ring buffer
TSIF_SET_WRITE_AVAIL	Moves the Write Pointer of the Transmitter by the specified amount
TSIF_REQ_TX_BUF	Allocates the memory for TX buffer
TSIF_QUERY_TX_BUF	Obtain the address of the Transmitter Ring buffer that has to be passed to the mmap call
TSIF_REQ_RX_BUF	Allocates the memory for RX buffer
TSIF_QUERY_RX_BUF	Obtain the address of the Receiver Ring buffer that has to be passed to the mmap call
TSIF_RX_RING_BUF_CONFIG	Configure the Receiver Ring buffer
TSIF_TX_RING_BUF_CONFIG	Configure the Transmitter Ring buffer
TSIF_SET_ATS	Enables the transmit ATS by filling the initial ATS value
TSIF_GET_ATS	Reads the current ATS value from TX_ATS_MON register

Table 2-169. TSIF Driver-Supported IOCTLs (continued)

Constant	Description
TSIF_BYPASS_ENABLE	Enables the Bypass mode
TSIF_SET_CONSEQUENTIAL_MODE	Enables Consequential mode
TSIF_WAIT_FOR_RX_COMPLETE	Waits for the receiver to complete
TSIF_ENABLE_GOP_DETECT	Enables/disables GOP detection.
TSIF_GET_SPCPKT	Reads the contents of the specific packet from the boundary sensing packet store address.
TSIF_WAIT_FOR_TX_COMPLETE	Waits for the transmit to complete
TSIF_SET_ENDIAN_CTL	Selects the SDRAM data storage format.

2.16 VLYNQ Driver

VLYNQ is serial (that is, low pin count) communications interface that enables the extension of an internal bus segment to one or more external physical devices. VLYNQ accomplishes this function by serializing bus transactions in one device, transferring the serialized transaction between devices through a VLYNQ port, and de-serializing the transaction in the external device.

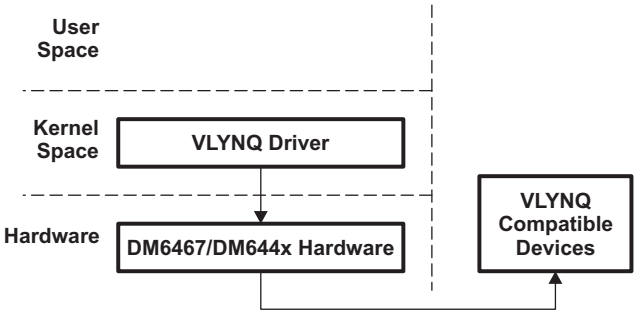

Figure 2-82. VLYNQ Diagram

The following features are supported by the driver:

- Configuration of the master and slave
- Configuration of clock direction and divisor values
- Configuration of VLYNQ Interrupts

2.16.1 Support and Constraints

Features Not Supported	None
Constraints	Currently, link between Vlynq Local(Master) and Vlynq Peer(Slave) is not detected.
Supported System Calls	None
Supported IOCTLs	None

2.17 PWM Driver

The pulse width modulator (PWM) generates a pulse periodic waveform for motor control, or can act as a digital-to-analog converter with some external components. This PWM peripheral is a timer with a period counter and a first-phase duration comparator. Bit width of the period and first-phase duration are both programmable.

The DaVinci PWM Driver supports the following features:

- The PWM driver is implemented as a char driver.
- Supports four instances of PWM on the DM355 and DM365 EVMs - It can be accessed from user space as /dev/davinci_pwm0, /dev/davinci_pwm1, /dev/davinci_pwm2, and /dev/davinci_pwm3
- Supports three instances of PWM on the DM6446 - It can be accessed from user space as /dev/davinci_pwm0, /dev/davinci_pwm1, and /dev/davinci_pwm2
- Supports Interrupt mode
- Supports continuous and one-shot (default) modes

2.17.1 Support and Constraints

Features not supported	DMA mode is not supported.
Constraints	In continuous mode, the minimum period for the PWM module is 8 cycles. Periods smaller than 8 cycles do not allow the interrupt capability to work properly.
Supported System Calls	open(), close(), ioctl()
Supported IOCTLs	See Table 2-170

Table 2-170. PWM Driver-Supported IOCTLs

Constant	Description
PWMIOC_SET_MODE	Set the mode of operation: continuous or one-shot
PWMIOC_SET_PERIOD	Set the PWM period
PWMIOC_SET_DURATION	Set the PWM First-Phase duration
PWMIOC_SET_RPT_VAL	Set the Repeat Count Value
PWMIOC_SET_FIRST_PHASE_STATE	Set the First Phase output level
PWMIOC_SET_INACT_OUT_STATE	Set the Inactive output level
PWMIOC_START	Start the PWM
PWMIOC_STOP	Reset the PWM settings

2.18 EDMA Driver

The EDMA controller handles all data transfers between the level-two (L2) cache/memory controller and the device peripherals. The EDMA supports up to 64-event channels and 8 QDMA channels. The EDMA consists of a scalable Parameter RAM (PaRAM) that supports flexible ping-pong, circular buffering, channel-chaining, auto-reloading, and memory protection. The EDMA allows movement of data to/from any addressable memory spaces, including internal memory (L2 SRAM), peripherals, and external memory.

The EDMA driver exposes only the kernel level APIs. This driver is used as a utility by other drivers for data transfer. Table 2-171 gives the details about the EDMA3 resource allocation for DM644x, DM355 and DM365. Table 2-172 gives such details for DM6467.

Table 2-171. DM644x and DM355 EDMA3 Resource Details

EDMA Resource	ARM	DSP
DMA Channel	0-63	None
QDMA Channel	4	0-3, 5-7
TCC	0-63	None
PaRAM Set	0-127	None

Table 2-172. DM646x EDMA3 Resource Details

EDMA Resource	ARM	DSP
DMA Channel	4-12, 16-23, 28, 29, 32-42, 54, 55	0-3, 13-15, 24-27, 30, 31, 43-53, 56-63
QDMA Channel	7	0-6
TCC	4-12, 16-23, 28, 29, 32-42, 54, 55	0-3, 13-15, 24-27, 30, 31, 43-53, 56-63
PaRAM Set	0-127	128-511

Table 2-173. DM365 EDMA3 Resource Details

EDMA Resource	ARM	DSP
DMA Channel	0-63	None
QDMA Channel	0-7	None
TCC	0-63	None
PaRAM Set	0-256	None

The EDMA driver supports the following features (Kernel APIs):

- Request and Free DMA channel
- Programs DMA channel
- Start and Synchronize with DMA transfers
- Provides DMA transaction completion callback to applications
- Multiple instances of EDMA driver on a single processor
- Read/Write a specific CC register
- Polled Mode DMA Transfers
- APIs can return error values

Features not supported	FIFO addressing mode is not supported.
Constraints	None
Supported System Calls	None
Supported IOCTLs	None

2.18.1 DM644x Performance and Benefits

2.18.1.1 EDMA Memory to Memory Low Latency Desktop Preemption

The performance numbers were captured using the following:

- Bytes transferred = 65535 bytes

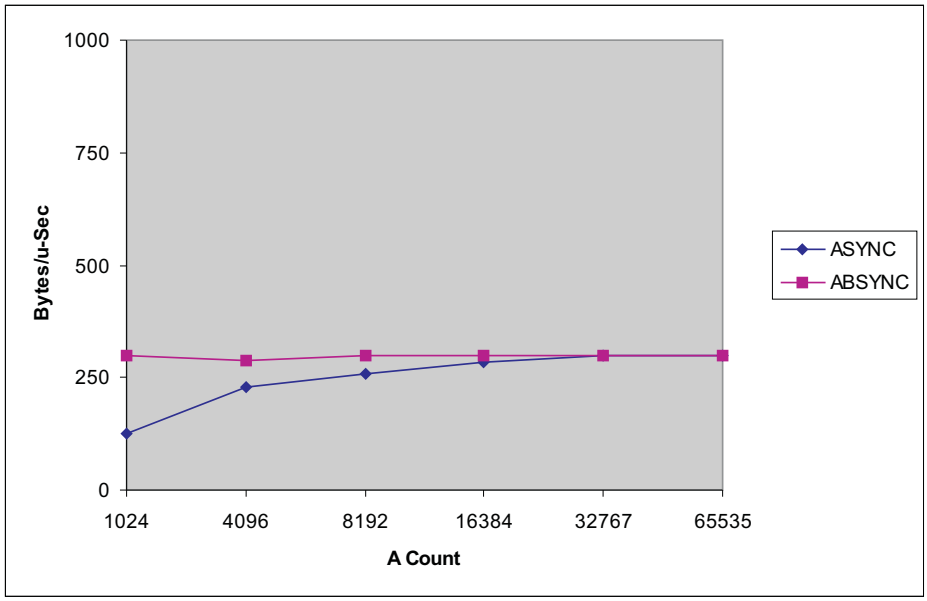

Figure 2-83. DM644x EDMA Memory to Memory Performance – LLD

Table 2-174. DM644x A-Sync EDMA Memory to Memory Performance Values – LLD

A Count	B Count	C Count	Time (u_Sec)	Bytes/u-Sec
1024	64	1	517	126.76
4096	16	1	287	228.35
8192	8	1	253	259.04
16384	4	1	231	283.71
32768	2	1	218	300.61
65535	1	1	219	299.25

Table 2-175. DM644x AB-Sync EDMA Memory to Memory Performance Values – LLD

A Count	B Count	C Count	Time (u_Sec)	Bytes/u-Sec
1024	64	1	219	299.25
4096	16	1	227	288.7
8192	8	1	219	299.25
16384	4	1	218	299.96
32768	2	1	218	300.61
65535	1	1	218	300.62

2.18.1.2 EDMA Memory to Memory Real Time Preemption

The performance numbers were captured using the following:

- Bytes transferred = 65535 bytes

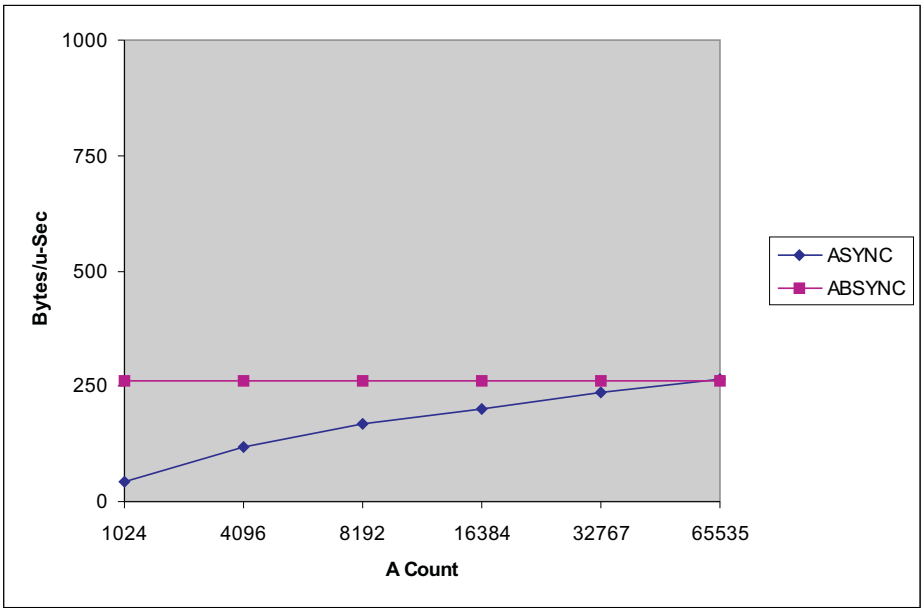

Figure 2-84. DM644x EDMA Memory to Memory Performance – RT

Table 2-176. DM644x A-Sync EDMA Memory to Memory Performance Values – RT

A Count	B Count	C Count	Time (u-Sec)	Bytes/u-Sec
1024	64	1	1471	44.55
4096	16	1	555	118.08
8192	8	1	392	167.18
16384	4	1	325	201.65
32768	2	1	277	236.58
65535	1	1	248	264.25

Table 2-177. DM644x AB-Sync EDMA Memory to Memory Performance Values – RT

A Count	B Count	C Count	Time (u-Sec)	Bytes/u-Sec
1024	64	1	250	262.14
4096	16	1	249	263.2
8192	8	1	251	261.1
16384	4	1	249	262.62
32768	2	1	252	260.06
65535	1	1	250	262.14

2.18.2 DM355 Performance and Benefits

2.18.2.1 EDMA Memory to Memory Low Latency Desktop Preemption

The performance numbers were captured using the following:

- Bytes transferred = 65535 bytes

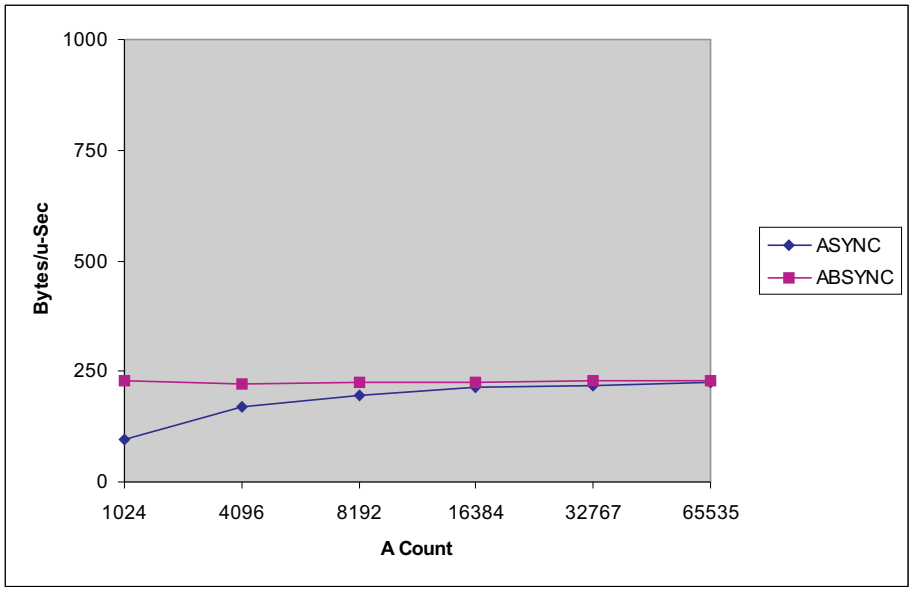

Figure 2-85. DM355 EDMA Memory to Memory Performance – LLD

Table 2-178. DM355 A-Sync EDMA Memory to Memory Performance Values – LLD

A Count	B Count	C Count	Time (u-Sec)	Bytes/u-Sec
1024	64	1	672	97.52
4096	16	1	384	170.67
8192	8	1	337	194.47
16384	4	1	308	212.78
32768	2	1	303	216.28
65535	1	1	289	226.76

Table 2-179. DM355 AB-Sync EDMA Memory to Memory Performance Values – LLD

A Count	B Count	C Count	Time (u-Sec)	Bytes/u-Sec
1024	64	1	287	228.35
4096	16	1	295	222.16
8192	8	1	289	226.77
16384	4	1	289	226.27
32768	2	1	286	229.14
65535	1	1	288	227.55

2.18.2.2 EDMA-Memory to Memory Real Time Preemption

The performance numbers were captured using the following:

- Bytes transferred = 65535 bytes

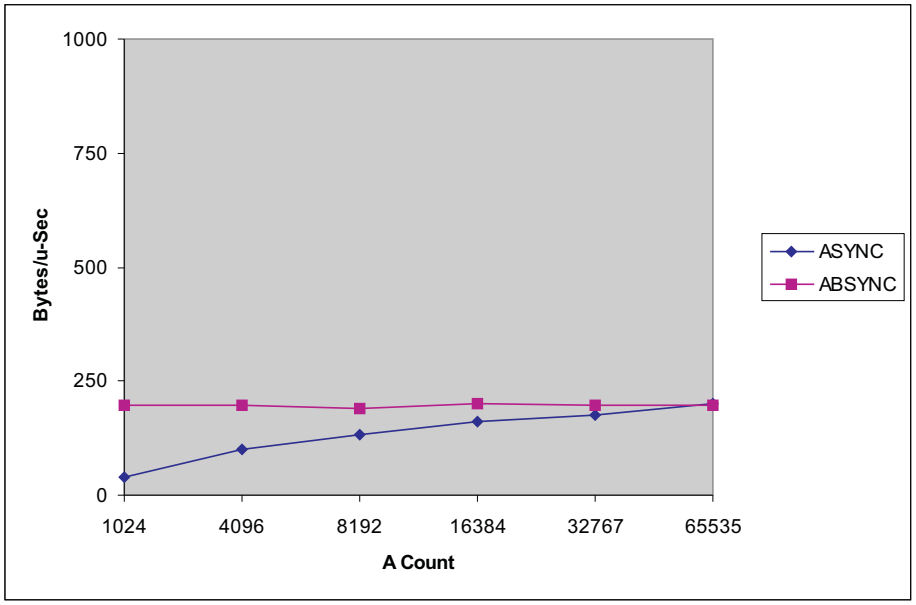

Figure 2-86. DM355 EDMA Memory to Memory Performance – RT

Table 2-180. DM355 A-Sync EDMA Memory to Memory Performance Values – RT

A Count	B Count	C Count	Time (u-Sec)	Bytes/u-Sec
1024	64	1	1708	38.37
4096	16	1	658	99.6
8192	8	1	491	199.47
16384	4	1	404	162.22
32768	2	1	373	175.69
65535	1	1	327	200.41

Table 2-181. DM355 AB-Sync EDMA Memory to Memory Performance Values – RT

A Count	B Count	C Count	Time (u-Sec)	Bytes/u-Sec
1024	64	1	330	198.59
4096	16	1	331	197.99
8192	8	1	342	191.63
16384	4	1	326	200.59
32768	2	1	331	197.99
65535	1	1	331	197.99

2.18.3 DM6467 Performance and Benefits

2.18.3.1 EDMA Memory to Memory Low Latency Desktop Preemption

The performance numbers were captured using the following:

- Bytes transferred = 65536 bytes

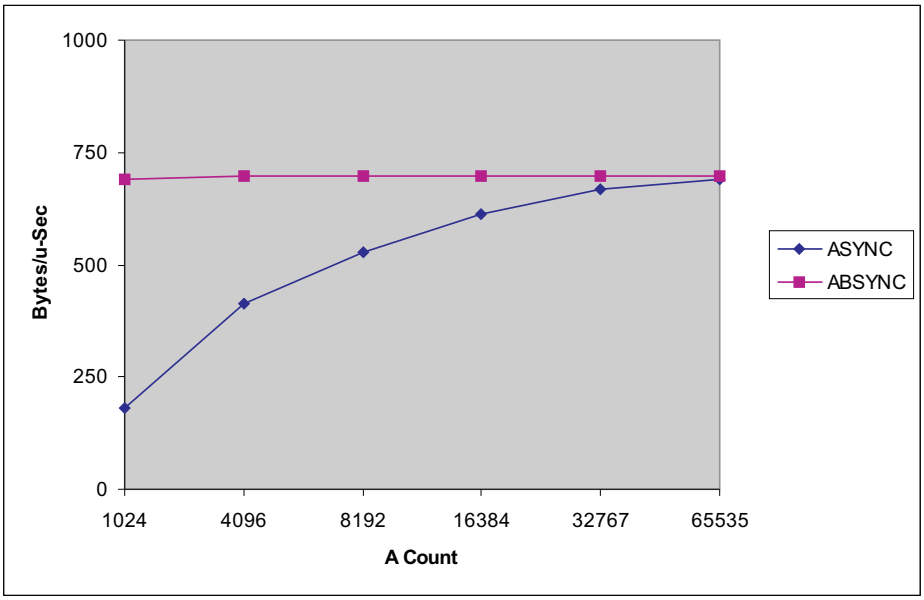

Figure 2-87. DM6467 EDMA Memory to Memory Performance – LLD

Table 2-182. DM6467 A-Sync EDMA Memory to Memory Performance Values – LLD

A Count	B Count	C Count	Time (u-Sec)	Bytes/u-Sec
1024	64	1	365	179.55
4096	16	1	159	412.18
8192	8	1	124	528.52
16384	4	1	107	612.49
32768	2	1	98	668.71
65535	1	1	95	689.84

Table 2-183. DM6467 AB-Sync EDMA Memory to Memory Performance Values – LLD

A Count	B Count	C Count	Time (u-Sec)	Bytes/u-Sec
1024	64	1	95	689.85
4096	16	1	94	697.19
8192	8	1	94	697.19
16384	4	1	94	695.66
32768	2	1	94	697.17
65535	1	1	94	697.18

2.18.3.2 EDMA-Memory to Memory Real Time Preemption

The performance numbers were captured using the following:

- Bytes transferred = 65536 bytes

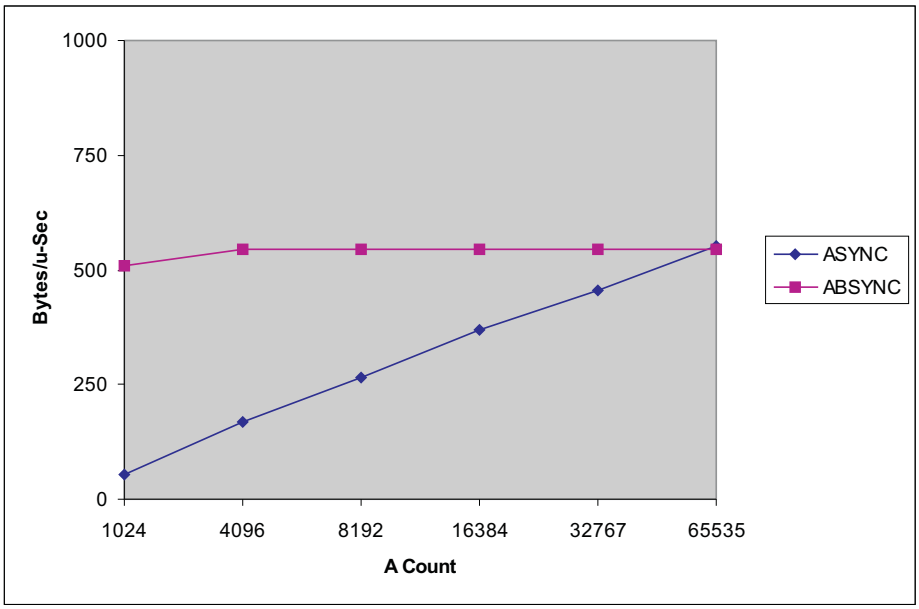

Figure 2-88. DM6467 EDMA Memory to Memory Performance – RT

Table 2-184. DM6467 A-Sync EDMA Memory to Memory Performance Values – RT

A Count	B Count	C Count	Time (u-Sec)	Bytes/u-Sec
1024	64	1	1211	54.12
4096	16	1	391	167.67
8192	8	1	247	265.33
16384	4	1	178	368.18
32768	2	1	144	455.1
65535	1	1	119	550.71

Table 2-185. DM6467 AB-Sync EDMA Memory to Memory Performance Values – RT

A Count	B Count	C Count	Time (u-Sec)	Bytes/u-Sec
1024	64	1	129	508.03
4096	16	1	120	546.13
8192	8	1	120	546.13
16384	4	1	120	544.93
32768	2	1	120	546.12
65535	1	1	120	546.13

2.18.4 DM365 Performance and Benefits

2.18.4.1 EDMA Memory to Memory Low Latency Desktop Preemption

The performance numbers were captured using the following:

- Bytes transferred = 65535 bytes

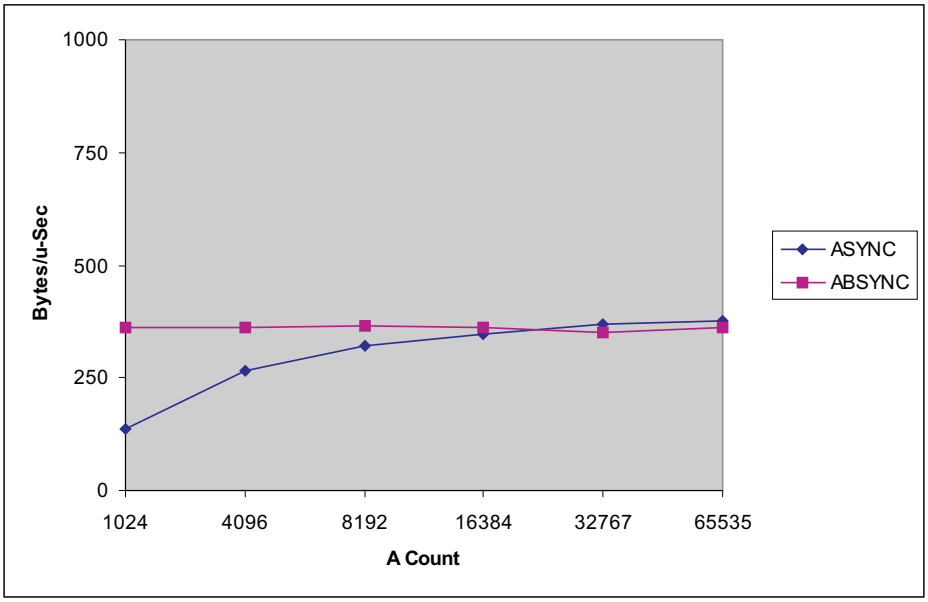

Figure 2-89. DM365 EDMA Memory to Memory Performance – LLD

Table 2-186. DM365 A-Sync EDMA Memory to Memory Performance Values – LLD

A Count	B Count	C Count	Time (microsec)	Bytes/u-Sec
1024	64	1	480	136.53
4096	16	1	245	267.49
8192	8	1	205	317.69
16384	4	1	189	346.75
32767	2	1	178	368.17
65535	1	1	174	376.64

Table 2-187. DM365 AB-Sync EDMA Memory to Memory Performance Values – LLD

A Count	B Count	C Count	Time (microsec)	Bytes/u-Sec
1024	64	1	182	360.09
4096	16	1	182	360.09
8192	8	1	180	364.09
16384	4	1	181	361.28
32767	2	1	186	352.33
65535	1	1	182	360.08

2.18.4.2 EDMA-Memory to Memory Real Time Preemption

The performance numbers were captured using the following:

- Bytes transferred = 65536 bytes

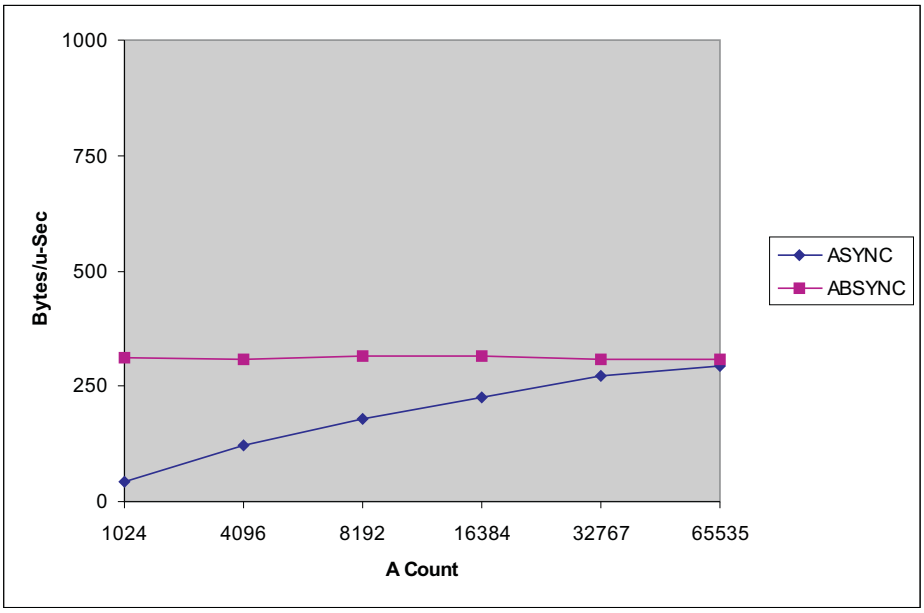

Figure 2-90. DM365 EDMA Memory to Memory Performance – RT

Table 2-188. DM365 A-Sync EDMA Memory to Memory Performance Values – RT

A Count	B Count	C Count	Time (microsec)	Bytes/u-Sec
1024	64	1	1521	43.09
4096	16	1	542	120.92
8192	8	1	368	178.09
16384	4	1	292	224.44
32767	2	1	241	271.93
65535	1	1	224	292.57

Table 2-189. DM365 AB-Sync EDMA Memory to Memory Performance Values – RT

A Count	B Count	C Count	Time (microsec)	Bytes/u-Sec
1024	64	1	209	313.57
4096	16	1	212	309.13
8192	8	1	208	315.08
16384	4	1	208	314.38
32767	2	1	212	309.12
65535	1	1	212	309.13

2.19 PCI Slave Driver

PCI Slave driver is implemented for the Linux operating system running on the DM6467 board interfacing with the host machine over the PCI interface. The driver is developed as a simple character driver exposing a minimal set of file operation structure namely release, mmap, ioctl, and fasync.

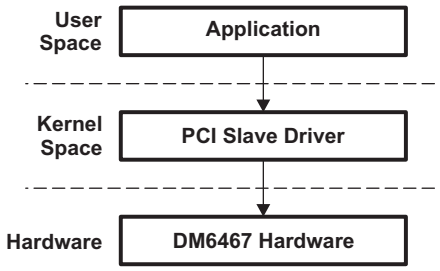

Figure 2-91. PCI Slave Driver Diagram

The PCI Slave driver supports the following functions:

- Generates and Handle interrupts received on the PCI interface.

Features not supported	None
Constraints	None
Supported System Calls	open(), close(), release(), ioctl(), mmap()
Supported IOCTLs	See Table 2-190.

Table 2-190. PCI Slave Driver-Supported IOCTLs

Constant	Description
PCIMODULE_CMD_GENNOTIFY	To generate a software interrupt
PCIMODULE_CMD_GETBARLOCN	To read the mapping

3 U-Boot Overview

The U-Boot bootloader in this release is based on the open-source version U-Boot 1.3.4. U-Boot 1.3.4 executes at reset and loads the O/S kernel into DDR2 memory.

U-Boot for NAND: In this case, U-Boot resides in NAND memory. At power reset, UBL is executed, which is responsible for loading the U-Boot from NAND to DDR2 memory. U-Boot is then executed from DDR2.

3.1 Functions

The U-Boot is responsible for the following functions:

- Initializes the DM355, DM365, DM6467, and DaVinci DM6446 EVM hardware
- Places DSP in RESET on the DaVinci DM6446 EVM
- Provides boot parameters to the Linux kernel
- Starts the Linux kernel
- Uploads new binary images to memory via serial port or Ethernet

Table 3-1. U-Boot, Kernel and File System Locations

	On DaVinci DM644x Board HD Boot	On DM355 board NAND Boot	On DM365 board NAND Boot	DM644x/DM6467/DM355/DM365 Boards NFS Boot
U-Boot Bootloader	NAND	NAND	NAND	Flash/NAND
Linux Kernel	NAND	NAND	NAND	Tools PC
File System	Hard Drive	NAND	NAND	Tools PC

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information 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.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DLP® Products	www.dlp.com	Broadband	www.ti.com/broadband
DSP	dsp.ti.com	Digital Control	www.ti.com/digitalcontrol
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Military	www.ti.com/military
Logic	logic.ti.com	Optical Networking	www.ti.com/opticalnetwork
Power Mgmt	power.ti.com	Security	www.ti.com/security
Microcontrollers	microcontroller.ti.com	Telephony	www.ti.com/telephony
RFID	www.ti-rfid.com	Video & Imaging	www.ti.com/video
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265