LSP 1.20 DaVinci Linux Drivers

Data Manual

Literature Number: SPRS496A
April 2008–Revised August 2008

Contents

1 LSP 1.20 DaVinci™ Linux Drivers 11
1.1 Features 11
1.2 Description 12
1.3 Limitations Summary 13
1.4 Toolchain and Version Information 13
1.5 Tested Modes 13
1.6 Timers 13
1.7 LSP/PSP Version Information 14
1.8 Documentation Support 14
2 Linux Kernel Device Drivers 15
2.1 Video Drivers 15
2.1.1 Video Display Driver - V4L2 16
2.1.1.1 Support and Constraints 17
2.1.1.2 Performance and Benchmarks 17
2.1.2 Video Display Driver - Fbdev 19
2.1.2.1 Support and Constraints 19
2.1.2.2 Performance and Benchmarks 20
2.1.3 Video Capture Driver 21
2.1.3.1 Features Not Supported 22
2.1.3.2 Constraints 22
2.1.3.3 Supported System Calls 22
2.1.3.4 Supported IOCTLs 22
2.1.3.5 Performance and Benchmarks 22
2.1.4 IPIPE Driver 24
2.1.4.1 Support and Constraints 24
2.1.4.2 Performance and Benchmarks 24
2.1.5 Previewer Driver 25
2.1.5.1 Support and Constraints 25
2.1.5.2 DM6446 EVM Performance and Benchmarks 25
2.1.6 Resizer Driver 25
2.1.6.1 Support and Constraints 26
2.1.6.2 Performance and Benchmarks 26
2.1.7 H3A Driver 26
2.1.7.1 Support and Constraints 26
2.1.7.2 Performance and Benchmarks 27
2.2 Audio Driver 27
2.2.1 Driver Features 28
2.2.2 Support and Constraints 28
2.2.3 Performance and Benchmarks 29
2.3 Ethernet Drivers 34
2.3.1 DM9000A Ethernet Driver for DM355 34
2.3.1.1 Support and Constraints 35
2.3.1.2 Performance and Benchmarks 35
2.3.2 CPMAC Ethernet Driver for DM6446 39
2.3.2.1 Driver Features 39
2.3.2.2 Support and Constraints 39
2.3.2.3 Performance and Benchmarks 39
2.4 USB Drivers 44
2.4.1 Driver Features 44
2.4.2 Support and Constraints 45
2.4.3 Performance and Benchmarks 45

2.4.3.1 DM355 EVM 45
2.4.3.2 DM6446 EVM 57
2.5 IDE ATA Driver 69
2.5.1 Driver Features 69
2.5.2 Support and Constraints 69
2.5.3 Performance and Benchmarks 70
2.5.3.1 DM6446 EVM 70
2.6 MMC/SD Driver 78
2.6.1 Driver Features 78
2.6.2 Support and Constraints 78
2.6.3 Performance and Benchmarks With Default MontaVista MMC/SD Stack 79
2.6.3.1 DM355 EVM 79
2.6.3.2 DM6446 EVM 87
2.6.4 Performance and Benchmarks With Custom SDIO Stack 93
2.6.4.1 DM355 EVM 93
2.6.4.2 DM6446 EVM 97
2.7 NAND Driver 102
2.7.1 Support and Constraints 102
2.7.2 Performance and Benchmarks 103
2.7.2.1 DM355 EVM 103
2.7.2.2 DM6446 EVM 117
2.8 NOR Driver 123
2.8.1 Performance and Benchmarks 125
2.8.1.1 DM6446 EVM 125
2.9 UART Driver 139
2.9.1 Driver Features 139
2.9.2 Support and Constraints 139
2.10 I2C Driver 140
2.10.1 Support and Constraints 140
2.11 SPI Driver 141
2.11.1 Support and Constraints 141
2.11.2 DM355 Performance and Benchmarks 141
2.11.2.1 SPI Driver - Read/Write Performance: Test Setup 142
2.11.2.2 SPI Driver- Read/Write Performance: Low Latency Desktop Mode 143
2.11.2.3 SPI Driver- Read/Write Performance: Server Mode 143
2.11.2.4 SPI Driver- Read/Write Performance: Real-Time Preemption Mode 143
2.12 PWM Driver 144
2.12.1 Support and Constraints 144
2.13 Watchdog Driver 145
2.14 GPIO Module 146
2.15 EDMA 146
3 U-Boot Overview 147
3.1 Functions 147
3.2 Features 147
3.3 U-Boot Commands 147

List of Figures

2-1 DM355/DM644x Video Driver Architecture 16
2-2 DaVinci Audio Driver Architecture 28
2-3 Audio Driver Runtime Performance Characteristics (LLD) - DM355 30
2-4 Audio Driver Runtime Performance Characteristics (Server) - DM355 30
2-5 Audio Driver Runtime Performance Characteristics (Real-Time) - DM355 31
2-6 Audio Driver Runtime Performance Characteristics (LLD) - DM6446 32
2-7 Audio Driver Runtime Performance Characteristics (Server) - DM6446 33
2-8 Audio Driver Runtime Performance Characteristics (Real-Time) - DM6446 33
2-9 Linux Kernel DM9000A Ethernet Driver Architecture 34
2-10 DM9000A Driver Runtime Performance Characteristics (LLD) 36
2-11 DM9000A Driver Runtime Performance Characteristics (Server) 37
2-12 DM9000A Driver Runtime Performance Characteristics (Real-Time) 38
2-13 Linux Kernel Ethernet Driver Architecture 39
2-14 CPMAC Ethernet Driver Runtime Performance Characteristics (LLD) 41
2-15 CPMAC Ethernet Driver Runtime Performance Characteristics (Server) 42
2-16 CPMAC Ethernet Driver Runtime Performance Characteristics (Real-Time) 43
2-17 Linux Kernel USB Driver Architecture 44
2-18 USB Driver - Host Write Performance Results (LLD) - DM355 47
2-19 USB Driver - Host Read Performance Results (LLD) - DM355 48
2-20 USB Driver - Host Write Performance: Server Mode 49
2-21 USB Driver - Host Read Performance Results (Server) — DM355 50
2-22 USB Driver - Host Write Performance Results (Real-Time) — DM355 51
2-23 USB Driver - Host Read Performance Results (Real-Time) - DM355 52
2-24 USB Driver - RNDIS Performance Results (LLD) - DM355 54
2-25 USB RNDIS Performance Results (Server) - DM355 55
2-26 USB Driver - RNDIS Performance Results (Real-Time) - DM355 56
2-27 USB Driver - Host Write Performance Results (LLD) - DM6446 59
2-28 USB Driver - Host Read Performance Results (LLD) - DM6446 60
2-29 USB Driver - Host Write Performance Results (Server) - DM6446 61
2-30 USB Driver - Host Read Performance Results (Server) - DM6446 62
2-31 USB Driver - Host Write Performance Results (Real-Time) - DM6446 63
2-32 USB Driver - Host Read Performance Results (Real-Time) - DM6446 64
2-33 USB Driver - RNDIS Performance Results (LLD) - DM6446 65
2-34 USB Driver - RNDIS Performance Results (Server) - DM6446 67
2-35 USB Driver - RNDIS Performance Results (Real-Time) - DM6446 68
2-36 Linux Kernel IDE Driver Architecture 69
2-37 IDE Driver - Write Performance Results (LLD) - DM6446 72
2-38 IDE Driver - Read Performance Results (LLD) - DM6446 73
2-39 IDE Driver - Write Performance Results (Server) - DM6446 74

2-40 IDE Driver Read - Performance Results (Server) - DM6446 75
2-41 IDE Driver - Write Performance Results (Real-Time) - DM6446 76
2-42 IDE Driver - Read Performance Results (Real-Time) - DM6446 77
2-43 Linux Kernel MMC Driver Architecture 78
2-44 MMC/SD Driver - SD Write Performance Results (LLD) - DM355 81
2-45 MMC/SD Driver - SD Read Performance Results (LLD) - DM355 82
2-46 MMC/SD Driver - SD Write Performance Results (Server) - DM355 83
2-47 MMC/SD Driver - SD Read Performance Results (Server) - DM355 84
2-48 MMC/SD Driver - SD Write Performance Results (Real-Time) - DM355 85
2-49 MMC/SD Driver - SD Read Performance Results (Real-Time) - DM355 86
2-50 MMC/SD Driver - SD Write Performance Results (LLD) - DM6446 88
2-51 MMC/SD Driver - SD Read Performance Results (LLD) - DM6446 89
2-52 MMC/SD Driver - SD Write Performance Results (Server) - DM6446 90
2-53 MMC/SD Driver - SD Read Performance Results (Server) - DM6446 91
2-54 MMC/SD Driver - SD Write Performance Results (Real-Time) - DM6446 92
2-55 MMC/SD Driver - SD Read Performance Results (Real-Time) - DM6446 93
2-56 Custom SDIO Stack - SD Write Performace Results (LLD) - DM355 94
2-57 Custom SDIO Stack - SD Read Performace Results (LLD) - DM355 94
2-58 Custom SDIO Stack - SD Write Performace Results (Server) - DM355 95
2-59 Custom SDIO Stack - SD Write Performace Results (Server) - DM355 96
2-60 Custom SDIO Stack - SD Write Performace Results (Real-Time) - DM355 96
2-61 Custom SDIO Stack - SD Write Performace Results (Real-Time) - DM355 97
2-62 Custom SDIO Stack - SD Write Performance Results (LLD) – DM6446 98
2-63 Custom SDIO Stack - SD Write Performance Results (LLD) - DM6446 98
2-64 Custom SDIO Stack - SD Write Performance Results (Server) - DM6446 99
2-65 Custom SDIO Stack - SD Write Performance Results (Server) - DM6446 100
2-66 Custom SDIO Stack - SD Write Performance Results (Real-Time) - DM6446 100
2-67 Custom SDIO Stack - SD Write Performance Results (Real-Time) - DM6446 101
2-68 Linux Kernel NAND Driver Architecture 102
2-69 SLC Write Performance: Low Latency Desktop Mode 105
2-70 NAND Driver - SLC Read (LLD) 106
2-71 NAND Driver - SLC Write (Server) 107
2-72 NAND Driver - SLC Read (Server) 108
2-73 NAND Driver - SLC Write (RT) 109
2-74 NAND Driver - SLC Read (RT) 110
2-75 NAND Driver - MLC Write (LLD) 111
2-76 NAND Driver - MLC Read (LLD) 112
2-77 NAND Driver - MLC Write (Server) 113
2-78 NAND Driver - MLC Read (Server) 114
2-79 NAND Driver - MLC Write (RT) 115
2-80 NAND Driver - MLC Read (RT) 116

2-81 NAND Driver - Write (LLD) 118
2-82 NAND Driver - Read (LLD) 119
2-83 NAND Driver - Write (Server) 120
2-84 NAND Driver - Read (Server) 121
2-85 NAND Driver - Write (RT) 122
2-86 NAND Driver - Read (RT) 123
2-87 Linux Kernel NOR Driver Architecture 124
2-88 NOR Driver - AMD Write Performance Results (LLD) - DM6446 126
2-89 NOR Driver - AMD Read Performance Results (LLD) – DM6446 127
2-90 NOR Driver - AMD Write Performance Results (Server) – DM6446 128
2-91 NOR Driver - AMD Read Performance Results (Server) – DM6446 129
2-92 NOR Driver - AMD Write Performance Results (Real-Time) – DM6446 130
2-93 NOR Driver - AMD Read Performance Results (Real-Time) – DM6446 131
2-94 NOR Driver - Intel Write (LLD) 132
2-95 NOR Driver - Intel Read (LLD) 133
2-96 NOR Driver - Intel Read (LLD) 134
2-97 NOR Driver - Intel Write (Server) 135
2-98 NOR Driver - Intel Read (Server) 136
2-99 NOR Driver - Intel Write (RT) 137
2-100 NOR Driver - Intel Read (RT) 138
2-101 Linux Kernel UART Driver Architecture 139
2-102 Linux Kernel I2C Driver Architecture 140
2-103 Linux Kernel SPI Driver Architecture 141

List of Tables

1-1 Device Drivers Summary 12
1-2 Limitations Summary 13
2-1 Output Mapping 16
2-2 IOCTLs Supported 17
2-3 Memory Requirements 17
2-4 V4L2 Display Driver Performance Data (LLD) - DM355 18
2-5 V4L2 Display Driver Performance Data (Server) - DM355 18
2-6 V4L2 Display Driver Performance Data (Real-Time) - DM355 18
2-7 Memory Requirements 18
2-8 V4L2 Display Driver Performance Data (LLD) - DM6446 18
2-9 V4L2 Display Driver Performance Data (Server) - DM6446 18
2-10 V4L2 Display Driver Performance Data (Real-Time) - DM6446 18
2-11 Fbdev Driver - EVM Output Signal Mapping 19
2-12 Fbdev Display Driver Supported IOCTLs 20
2-13 Fbdev Display Driver Performance Data (LLD) - DM355 21
2-14 Fbdev Display Driver Performance Data (Server) - DM355 21
2-15 Fbdev Display Driver Performance (Real-Time) data - DM355 21
2-16 Fbdev Display Driver Performance Data (LLD) - DM6446 21
2-17 Fbdev Display Driver Performance Data (Server) - DM6446 21
2-18 Fbdev Display Driver Performance (Real-Time) data - DM6446 21
2-19 Video Capture Driver Supported IOCTLs 22
2-20 V4L2 Capture Driver Memory Statistics - DM355 23
2-21 V4L2 Capture Driver Performance Data (LLD) - DM355 23
2-22 V4L2 Capture Driver Performance Data (Server) - DM355 23
2-23 V4L2 Capture Driver Performance Data (Real-Time) - DM355 23
2-24 V4L2 Capture Driver Memory Statistics - DM6446 23
2-25 V4L2 Capture Driver Performance Data (LLD) - DM6446 23
2-26 V4L2 Capture Driver Performance Data (Server) - DM6446 24
2-27 V4L2 Capture Driver Performance Data (Real-Time) - DM6446 24
2-28 IPIPE Driver Supported IOCTLs 24
2-29 IPIPE Driver Memory Statistics - DM355 24
2-30 Previewer Driver Supported IOCTLs 25
2-31 Previewer Driver Memory Statistics - DM6446 25
2-32 Resizer Driver Supported IOCTLs 26
2-33 Resizer Driver Memory Statistics - DM6446 26
2-34 H3A Driver Supported IOCTLs 27
2-35 H3A Driver Memory Statistics - DM355 27
2-36 H3A Driver Memory Statistics - DM6446 27
2-37 DaVinci Audio Driver Supported IOCTLs 28

2-38 Audio Driver Memory Statistics - DM355 29
2-39 Audio Driver Runtime Performance Data (LLD) - DM355 30
2-40 Audio Driver Runtime Performance Data (Server) - DM355 31
2-41 Audio Driver Runtime Performance Data (Real-Time) - DM355 31
2-42 Audio Driver Memory Statistics - DM6446 32
2-43 Audio Driver Runtime Performance Data (LLD) - DM6446 32
2-44 Audio Driver Runtime Performance Data (Server) - DM6446 33
2-45 Audio Driver Runtime Performance Data (Real-Time) - DM6446 34
2-46 DM9000A Driver Memory Statistics 35
2-47 DM9000A Driver Performance Using IPerf (LLD) 36
2-48 DM9000A Driver Performance Using IPerf (Server) 37
2-49 DM9000A Driver Performance Using IPerf (Real-Time) 38
2-50 CPMAC Ethernet Driver Memory Statistics 40
2-51 CPMAC Ethernet Driver Performance Using IPerf (LLD) 41
2-52 CPMAC Ethernet Driver Performance Using IPerf (Server) 42
2-53 CPMAC Ethernet Driver Performance Using IPerf (Real-Time) 43
2-54 USB Driver Memory Statistics - DM355 45
2-55 USB Driver - Host Write Performance Data (LLD) - DM355 47
2-56 USB Driver - Host Read Performance Data (LLD) - DM355 48
2-57 USB Driver - Host Write Performance Data (Server) - DM355 49
2-58 USB Driver - Host Read Performance Data (Server) - DM355 50
2-59 USB Driver - Host Write Performance Data (Real-Time) - DM355 51
2-60 USB Driver - Host Read Performance Data (Real-Time) - DM355 52
2-61 USB Driver - RNDIS Performance Data Using IPerf (LLD) - DM355 54
2-62 USB RNDIS Performance Data Using IPerf (Server) - DM355 55
2-63 USB Driver - RNDIS Performance Data Using IPerf (Real-Time) - DM355 56
2-64 USB Driver Memory Statistics - DM6446 57
2-65 USB Driver - Host Write Performance Data (LLD) - DM6446 59
2-66 USB Driver - Host Read Performance Data (LLD) - DM6446 60
2-67 USB Driver - Host Write Performance Data (Server) - DM6446 61
2-68 USB Driver - Host Read Performance Data (Server) - DM6446 62
2-69 USB Driver - Host Write Performance Data (Real-Time) - DM6446 63
2-70 USB Driver - Host Read Performance Data (Real-Time) - DM6446 64
2-71 USB Driver - RNDIS Performance Data Using IPerf (LLD) - DM6446 65
2-72 USB Driver - RNDIS Performance Data Using IPerf (Server) - DM6446 67
2-73 USB Driver - RNDIS Performance Data Using IPerf (Real-Time) - DM6446 68
2-74 IDE Driver Supported IOCTLs 69
2-75 IDE ATA Driver Memory Statistics - DM6446 70
2-76 IDE Driver - Write Performance Data (LLD) - DM6446 72
2-77 IDE Driver - Read Performance Data (LLD) - DM6446 73
2-78 IDE Driver - Write Performance Data (Server) - DM6446 74

2-79 IDE Driver - Read Performance Data (Server) - DM6446 75
2-80 IDE Driver - Write Performance Data (Real-Time) - DM6446 76
2-81 IDE Driver - Read Performance Data (Real-Time) - DM6446 77
2-82 MMC Driver Memory Statistics - DM355 79
2-83 MMC/SD Driver - SD Write Performance Data (LLD) - DM355 81
2-84 MMC/SD Driver - SD Read Performance Data (LLD) - DM355 82
2-85 MMC/SD Driver - SD Write Performance Data (Server) - DM355 83
2-86 MMC/SD Driver - SD Read Performance Data (Server) - DM355 84
2-87 MMC/SD Driver - SD Write Performance Data (Real-Time) - DM355 85
2-88 MMC/SD Driver - SD Read Performance Data (Real-Time) - DM355 86
2-89 MMC Driver Memory Statistics - DM6446 87
2-90 MMC/SD Driver - SD Write Performance Data (LLD) - DM6446 88
2-91 MMC/SD Driver - SD Read Performance Data (LLD) - DM6446 89
2-92 MMC/SD Driver - SD Write Performance Data (Server) - DM6446 90
2-93 MMC/SD Driver - SD Read Performance Data (Server) - DM6446 91
2-94 MMC/SD Driver - SD Write Performance Data (Real-Time) - DM6446 92
2-95 MMC/SD Driver - SD Read Performance Data (Real-Time) - DM6446 93
2-96 Custom SDIO Stack - SD Write Performace Data (LLD) - DM355 94
2-97 Custom SDIO Stack - SD Read Performace Results (LLD) - DM355 95
2-98 Custom SDIO Stack - SD Write Performace Data (Server) - DM355 95
2-99 Custom SDIO Stack - SD Write Performace Data (Server) - DM355 96
2-100 Custom SDIO Stack - SD Write Performace Results (Real-Time) - DM355 97
2-101 Custom SDIO Stack - SD Write Performace Data (Real-Time) - DM355 97
2-102 Custom SDIO Stack - SD Write Performance Data (LLD) - DM6446 98
2-103 Custom SDIO Stack - SD Write Performance Data (LLD) - DM6446 99
2-104 Custom SDIO Stack - SD Write Performance Data (Server) - DM6446 99
2-105 Custom SDIO Stack - SD Write Performance Data (Server) - DM6446 100
2-106 Custom SDIO Stack - SD Write Performance Data (Real-Time) - DM6446 101
2-107 Custom SDIO Stack - SD Write Performance Data (Real-Time) - DM6446 101
2-108 Supported IOCTLs 102
2-109 NAND Driver Memory Statistics - DM355 103
2-110 NAND Driver - SLC Write (LLD) 105
2-111 NAND Driver - SLC Read (LLD) 106
2-112 NAND Driver - SLC Write (Server) 107
2-113 NAND Driver - SLC Read (Server) 108
2-114 NAND Driver - SLC Write (RT) 109
2-115 NAND Driver - SLC Read (RT) 110
2-116 NAND Driver - MLC Write (LLD) 111
2-117 NAND Driver - MLC Read (LLD) 112
2-118 NAND Driver - MLC Write (Server) 113
2-119 NAND Driver - MLC Read (Server) 114

2-120 NAND Driver - MLC Write (RT) 115
2-121 NAND Driver - MLC Read (RT) 116
2-122 NAND Driver Memory Statistics - DM6446 117
2-123 NAND Driver - Write (LLD) 118
2-124 NAND Driver - Read (LLD) 119
2-125 NAND Driver - Write (Server) 120
2-126 NAND Driver - Read (Server) 121
2-127 NAND Driver - Write (RT) 122
2-128 NAND Driver - Read (RT) 123
2-129 NOR Supported IOCTLs 124
2-130 NOR Driver Memory Statistics - DM6446 125
2-131 NOR Driver - AMD Write Performance Data (LLD) - DM6446 126
2-132 NOR Driver - AMD Read Performance Data (LLD) - DM6446 127
2-133 NOR Driver - AMD Write Performance Data (Server) - DM6446 128
2-134 NOR Driver - AMD Read Performance Data (Server) - DM6446 129
2-135 NOR Driver - AMD Write Performance Data (Real-Time) - DM6446 130
2-136 NOR Driver - AMD Read Performance Data (Real-Time) - DM6446 131
2-137 NOR Driver - Intel Write (LLD) 132
2-138 NOR Driver - Intel Read (LLD) 133
2-139 NOR Driver - Intel Read (LLD) 134
2-140 NOR Driver - Intel Write (Server) 135
2-141 NOR Driver - Intel Read (Server) 136
2-142 NOR Driver - Intel Write (RT) 137
2-143 NOR Driver - Intel Read (RT) 138
2-144 UART Supported IOCTLs 139
2-145 I2C Driver Supported IOCTLs 140
2-146 SPI Driver Memory Statistics - DM355 141
2-147 SPI Driver - Read/Write Performance Data (Low Latency Desktop) - DM355 143
2-148 SPI Driver - Read/Write Performance Data (Server) - DM355 143
2-149 SPI Driver - Read/Write Performance Data (Real-Time) - DM355 143
2-150 PWM Driver-supported IOCTLs 144
2-151 WDT Driver Supported IOCTLs 145
3-1 U-Boot, Kernel and File System Locations 147
3-2 U-Boot Command Set 147

1 LSP 1.20 DaVinci™ Linux Drivers

1.1 Features

- **Video Processing Subsystem Front End (VPFE)**
  - **Supports NTSC/PAL Video Capture From TVP5146 Decoder, using V4L2-based Capture Driver**
  - **Supports Raw Capture From Micron Image Sensors (MT9T001 and MT9T031)**
  - **Supports Previewer Engine to Convert Bayer Input Into YCbCr Format using Previewer Driver (DM644x only)**
  - **Supports Input Image Upscaling and Downscaling using Resizer Driver (DM644x only)**
  - **Supports Auto Exposure and Auto White balance (AEW) on CCDC Input Data**
  - **Supports Auto Focus (AF) on CCDC Input**
  - **Supports Conversion of Bayer Input Into YCbCr Format (Previewer) as Well as Upscaling and Downscaling (Resizer) using IPIPE Driver (DM355 only)**
- **Video Processing Back End (VPBE)**
  - **Supports NTSC/PAL and 480p/576p (DM644x only) Video Display up to D1 Resolution Using Fbdev and V4L2-based Video Display Drivers (Mutually Exclusive)**
  - **Supports Graphic On-screen Display (OSD) With Alphablending using Fbdev Driver**
  - **Supports VGA Output Using LogicPD LCD Daughter Card**
  - **Supports HD (720p/1080i) Video Display using THS8200 Daughter Card**
- **Audio - Open Sound System (OSS)**
  - **Supports Audio Record and Playback From AIC33**
- **Communication (CPMAC, DM9000A, USB)**
  - **Supports CPMAC (internal EMAC) Driver for transmitting and receiving network data (DM644x only)**
  - **Supports DM9000A (External Ethernet Controller) Driver for transmitting and receiving network data (DM355 only)**
  - **Supports USB Host, Gadget and OTG (partial) Modes Based on Inventra USB Driver Stack, With Support for USB 2.0 Full and High Speed Devices, Mass Storage Class (NAND and HDD), Hub, HID, and ISO Class**
- **Storage (ATA, MMC/SD, NAND, NOR)**
  - **Supports MMC/SD Interface to External Flash Cards - MMC and SD**
  - **Supports Big Block NAND (MLC and SLC) With 4-Bit HW ECC Capability**
  - **Supports ATA Interface to IDE Hard Disk (DM644x only)**
  - **Supports NOR Flash Memory for Storage (DM644x only)**
- **Supports Serial Communication With SPI Devices (DM355 only)**
- **Serial (UART, I2C)**
  - **Supports Serial Communication Using UART**
  - **Supports I2C Interface to MSP430 Controller as well as other Audio/Video Encoder/Decoders**
- **Supports Kernel APIs for General Purpose Input/Output (GPIO) Configuration**
- **Supports Pulse Width Modulation Driver Module**
- **Supports up to 4 Timers (configured as 4x32-bit), Including MV High Resolution Timer**
- **Supports a 64-bit Watchdog Timer for Monitoring System Conditions**
- **Supports EDMA3 Kernel APIs**
- **Supports RamDisk, EXT2/EXT3, YAFFS/YAFFS2 on NAND and JFFS2 on NOR File Systems**
- **Compiled with MontaVista Professional 4.0.1 arm_v5t_le Toolchain**
- **Supports GNU/Linux ARM9 Application Binary Interface (ABI)**
- **Development on DaVinci ™ DM6446 EVM and DM355 EVM Platforms**
- **Portable to Customer DaVinci DM644x and DM355 Hardware Platforms**
- **Supported Devices:**
  - **TMS320DM355**
  - **TMS320DM644x**

1.2 Description

The baseline kernel in the Production Release plus Patch Level 14 of LSP 1.20 is derived from TI-DaVinci LSP 1.10 (MV 4.0.1 Kernel 2.6.10) with Patch level 49 applied. The patches are applied to MV 4.0.1 Kernel 2.6.10, as-is, or with modifications or with a replacement community/MV patch. The merged patches from community/MV utilize the latest community framework and enable better interoperability with standard product and community code.

Table 1-1. Device Drivers Summary

Peripheral	Description	Linux Driver Type	Device Support
Video Processing Front End (VPFE)	Enables Video Capture for SD standard resolutions (NTSC, PAL)	V4L2	DM644x, DM355
Video Processing Back End (VPBE)	Enables Video Display for SD standard resolutions (NTSC, PAL)	V4L2	DM644x, DM355
Video Processing Back End (VPBE)	Enables Video Display for SD standard resolutions (NTSC, PAL)	FBDEV	DM644x, DM355
Video Processing Back End (VPBE) OSD graphics	Graphics on-screen display with alphablending	FBDEV	DM644x, DM355
IPIPE	Combined Previewer Engine and Resizer modules - Convert Bayer input to YCbCr format, Input image up-scaling up to 4x and downscaling up to 1/4x		DM355
Preview Engine	Convert Bayer input to YCbCr format		DM644x
Resizer	Input image up-scaling up to 4x and downscaling up to 1/4x		DM644x
Auto Exposure/Auto White Balance (AEW)	Auto Exposure /Auto White Balance for statistical collection of video data		DM644x, DM355
Auto Focus (AF)	Auto Focus for statistical collection of video data		DM644x, DM355
Audio (McBSP)	Audio Record and Playback	OSS	DM644x, DM355
Ethernet	Transmit/receive network data. Supports Auto negotiation with 10/100 Mbps link speed.	Network Driver	DM644x
DM9000A Ethernet	Transmit/receive network data. Supports Auto negotiation with 10/100 Mbps link speed.	Network Driver	DM355
USB 2.0 MSC Host	USB Mass Storage Class Host Driver		DM644x, DM355
USB 2.0 MSC Slave	USB Mass Storage Class Slave Driver		DM644x, DM355
USB 2.0 CDC Host	USB Communication Device Class Host Driver		DM644x, DM355
USB 2.0 CDC Slave	USB Communication Device Class Slave Driver		DM644x, DM355
USB 2.0 HID Host	USB Human Interface Device Host Driver		DM644x, DM355
USB 2.0 ISO	USB Isochronous Device Driver for Video and Audio Devices		DM644x, DM355
USB 2.0 OTG	USB On The Go. Supports HNP and SRP.		DM644x, DM355
ATA/ATAPI	Interface to ATA hard disk/CDROM		DM644x
NAND	Flash Storage system		DM644x, DM355
NOR	Flash Storage system		DM644x
MMC/SD	Interface to Multi Media /Secure digital cards		DM644x, DM355
UART	Serial Communication Interface		DM644x, DM355
I2C	Inter-IC Communication		DM644x, DM355
SPI	Serial Peripheral Interface		DM355
PWM	Pulse Width Modulation typically used for motor control		DM644x, DM355
Timers	General Purpose Timers and OS ticks		DM644x, DM355
Watchdog Timer (64-bit)	Monitor system condition		DM644x, DM355
GPIO	General Purpose IO		DM644x, DM355
PINMUX	Device Pin Configuration		DM644x, DM355
EDMA3	Enhanced DMA Engine		DM644x, DM355

1.3 Limitations Summary

Table 1-2 summarizes the known feature limitations in the MontaVista Linux 2.6.10 LSP 1.20.

Table 1-2. Limitations Summary

PERIPHERAL	STATUS	USE CASE
VLYNQ	No driver; no VLYNQ peripheral on EVM	Interface to FPGA or VLYNQ peripheral
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

1.4 Toolchain and Version Information

The kernel and MontaVista Linux (MVL) user space application are compiled and linked with the MontaVista Professional 4.0.1 arm_v5t_le toolchain. The Application Binary Interface (ABI) is GNU/Linux ARM9. The Linux kernel version in MontaVista Linux Professional 4.0.1 is 2.6.10.

The toolchain components consist of:

- GNU Compiler Collection (GCC) 3.4.3
- GNU libc (glibc) 2.8.3
- GNU Debugger (GDB) 6.3
- Binutils 2.15.94

1.5 Tested Modes

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

- No forced preemption (server)
- Preemptible kernel (low latency desktop)
- Complete preemption (real-time)

1.6 Timers

DM355 EVM

The DM355 has four software-programmable general purpose timers. Two of the 64-bit timers are configured as four independent 32-bit timers:

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

DM6446 EVM

The DaVinci DM6446 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
   a. Timer 0 Low (1:2) - Free-running counter, used for cycle counter
   b. 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.

1.7 LSP/PSP Version Information

This document applies to the driver set from the Production Release plus Patch Level 14 of LSP 1.20.

1.8 Documentation Support

The following documents are available for download:

SPRAAU0 *LSP 1.20 Frame Buffer Device Driver Migration From LSP 1.10 Application Report*
describes how to migrate from the LSP 1.10 to the LSP 1.20 frame buffer device driver.

SPRUEK9 *LSP 1.20 DaVinci FBDev Display User's Guide*
describes how to install and use the FBDev display driver.

SPRUEL3 *LSP 1.20 DaVinci V4L2 Display User's Guide*
describes how to install and use the V4L2 display driver.

SPRUEL6 *LSP 1.20 DaVinci Video Sysfs User's Guide*
describes how to install and use the Video Sysfs driver.

SPRUEP5 *LSP 1.20 DaVinci Linux AF Driver User's Guide*
describes how to install and use the AF Driver.

SPRUEP6 *LSP 1.20 DaVinci AEW Device Driver User's Guide*
describes how to install and use the auto expose/auto white balance (AEW) driver.

SPRUEP7 *LSP 1.20 DaVinci CCDC Device Driver User's Guide*
describes how to install and use the CCDC driver.

SPRUF10 *LSP 1.20 DaVinci NOR Flash Device Driver User's Guide*
describes how to install and use the NOR Flash Driver.

SPRUFE9 *LSP 1.20 DaVinci Previewer Device Driver User's Guide*
describes how to install and use the previewer driver.

SPRUFF0 *LSP 1.20 DaVinci Resizer Device Driver User's Guide*
describes how to install and use the resizer driver.

SPRUFG0 *LSP 1.20 DaVinci Linux EVM Installation User's Guide*
describes how to install the Linux EVM.

SPRUFG1 *LSP 1.20 DaVinci Linux IPIPE Driver User's Guide*
describes how to install and use the IPIPE Driver.

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 Video Drivers

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

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

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

The device nodes created for V4L2 Display driver are:

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

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), auto expose/auto white balance (AEW)) and histogram drivers are front-end components that provide post-capture processing.

The following add-on daughter cards have been used to test HD capture/display and VGA resolutions:

Image Sensors:

- Micron MT9T031P12STCH ES (3 Megapixel)
- A ribbon cable is required to connect the Micron sensor headboard to the DM355 EVM. The ribbon cable can be ordered at www.samtec.com (SAMTEC product #: IDSD-13-D-04.00-T-G)
- Tested with DM355 devices

LCDs:

- LogicPD Touch-screen LCD
- LCD kits can be procured through distribution – Avnet, Arrow or Digikey; Part Number – LCD-6.4-VGA-10R
- Adaptor cards needed to connect this LCD to the EVM can be ordered directly from LogicPD
- Touch-screen features are not enabled with the EVM; Only the LCD screen has been tested with the EVM.

THS8200 Daughter cards:

- THS8200 cards can be ordered from Spectrum Digital

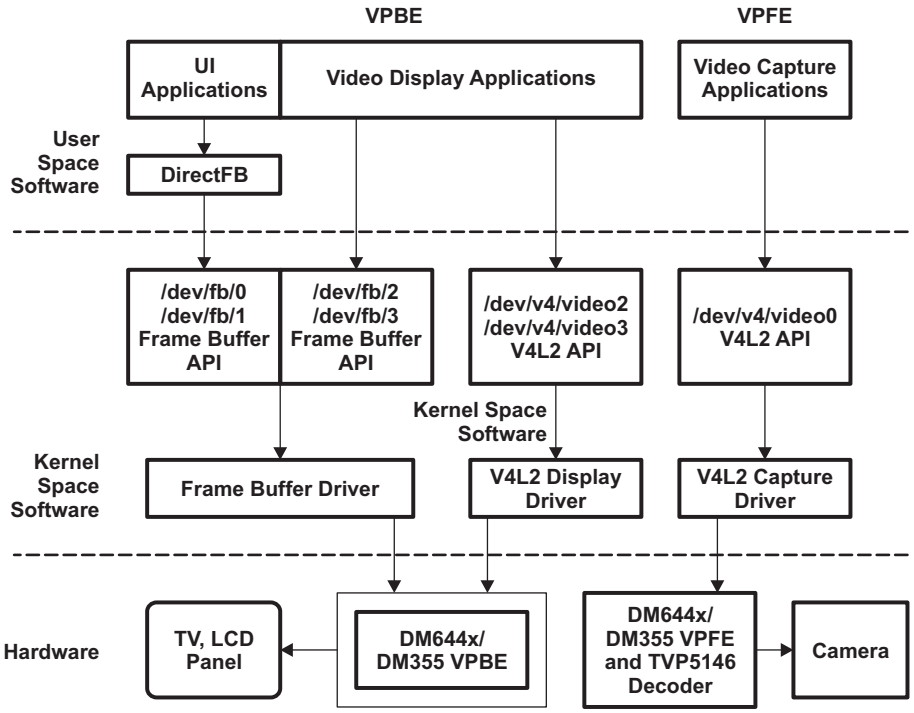

Figure 2-1. DM355/DM644x Video Driver Architecture

2.1.1 Video Display Driver - V4L2

The V4L2 display driver supports the following features:

- Two video windows (VID0 and VID1) in YUV422 format (UYVY)
- Co-exists with Fbdev driver. V4L2 drives the video windows and Fbdev drives the OSD windows
- Number of buffers is configurable, with a default of three buffers
- Requires sysfs to do output or standard switching. The related IOCTLs are removed from the V4L2 driver.
- Analog outputs using the four internal DACs
  - Composite video using one DAC
  - S-Video using two DACs (Only in DM6446)
  - Component video using three DACs (Only in DM6446)
- NTSC and PAL interlaced outputs using the internal video encoder
- VGA output (640 × 480, 640 × 400 and 640 × 350 resolutions) through LogicPD daughter card
- Support for HD video (720p and 1080i resolutions) on VID0 window through THS8200 daughter card
- Flexibility to change window sizes and starting coordinates
- Runtime enable/disable of windows
- Maps output signals to the RCA jacks on the EVM based on boot time selection, as shown in Table 2-1 for DM6446: (For DM355, only one DAC is available and it maps to composite output)

Table 2-1. Output Mapping

Output Selection	DAC A	DAC B	DAC C
Composite	Composite	Composite	Composite
Component	Y	Pb	Pr
S-Video	Composite	Y	C

2.1.1.1 Support and Constraints

Features Not Supported The OSD Windows are not supported by V4L2 Video Display Driver

Constraints

- The vid0 and vid1 windows cannot be used simultaneously in DM6446 (PG < 2.1); however, this issue is not applicable in DM355.
- 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()

Supported IOCTLs See Table 2-2

Table 2-2. IOCTLs Supported

Constant	Description
VIDIOC_REQBUFS	Allocate the memory
VIDIOC_QUERYBUF	Get Physical address of the allocated buffer
VIDIOC_QUERYCAP	Querying capabilities of driver
VIDIOC_S_OUTPUT	Set the video output type
VIDIOC_G_OUTPUT	Get the video output type
VIDIOC_ENUMSTD	Enumerate the information of available standards, for current output of current encoder
VIDIOC_QUERYSTD	Detect the current standard set in current encoder
VIDIOC_S_STD	Set the video standard
VIDIOC_G_STD	Get the video standard
VIDIOC_ENUM_FMT	Enumerate the information of format those are supported by current encoder
VIDIOC_S_FMT	Set the format for current encoder
VIDIOC_G_FMT	Get the format for current encoder
VIDIOC_TRY_FMT	Validate the format for current encoder
VIDIOC_QBUF	Enqueue the buffer in buffer queue
VIDIOC_DQBUF	Dequeue the buffer in buffer queue
VIDIOC_STREAMON	Start video Display functionality
VIDIOC_STREAMOFF	Stop video Display functionality
VIDIOC_CROPCAP	Query crop capabilities of driver
VIDIOC_G_CROP	Get current crop parameters
VIDIOC_S_CROP	Set current crop parameters

2.1.1.2 Performance and Benchmarks

DM355 EVM:

Table 2-3 shows the memory requirements for the V4L2 Display Driver module built for the DM355 device.

Table 2-3. Memory Requirements

Configuration	Memory Statistics (in Bytes)			
	Program Memory	Data Memory		Total
		Initialized	Uninitialized	
davinci_display.ko	16608	1264	104	17976

Table 2-4. V4L2 Display Driver Performance Data (LLD) - DM355

Video Format	Frames per Second
NTSC	29.972
PAL	24.986
LCD	64.762

Table 2-5. V4L2 Display Driver Performance Data (Server) - DM355

Video Format	Frames per Second
NTSC	29.973
PAL	24.984
LCD	64.777

Table 2-6. V4L2 Display Driver Performance Data (Real-Time) - DM355

Video Format	Frames per Second
NTSC	29.958
PAL	25.003
LCD	64.802

DM6446 EVM:

Table 2-7 shows the memory requirements for the V4L2 Display Driver module built for DM6446.

Table 2-7. Memory Requirements

Configuration	Memory Statistics (in Bytes)			
	Program Memory	Data Memory		Total
		Initialized	Uninitialized	
davinci_display.ko	16616	1240	104	17960

Table 2-8. V4L2 Display Driver Performance Data (LLD) - DM6446

Video Format	Frames per Second
NTSC	29.969
PAL	24.986
LCD	64.767

Table 2-9. V4L2 Display Driver Performance Data (Server) - DM6446

Video Format	Frames per Second
NTSC	29.956
PAL	24.984
LCD	64.781

Table 2-10. V4L2 Display Driver Performance Data (Real-Time) - DM6446

Video Format	Frames per Second
NTSC	29.975
PAL	24.996
LCD	64.776

2.1.2 Video Display Driver - Fbdev

The Fbdev Display Driver supports the following features:

- Two video windows (VID0 and VID1) in YUV422 format. Both windows are double-buffered or triple-buffered.
- Blinking in the On Screen Display (OSD) attribute field
- An OSD or bitmap window using OSD0 in RGB 565 format with double buffering
- An attribute window to act 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 (only in DM644x)
  - Component video using three DACs (only in DM644x)
- NTSC and PAL interlaced outputs using the internal video encoder
- HD video outputs (720p and 1080i resolutions) through THS8200 daughter card
- Flexibility to change window sizes and starting coordinates
- Zoom feature to zoom the display to 1x, 2x, 4x
- 480p and 576p formats
- VGA output (640 × 480, 640 × 400 and 640 × 350 resolutions) through LogicPD daughter card
- Digital output on RGB666 digital interface
- Multiple video and OSD windows
- RGB888 input mode for the video window (only for DM644x)
- Palletized input (1/2/4/8 bit) mode for OSD window
- Runtime enable/disable of windows
- Color-keying
- Programmed flip-queuing
- Maps output signals to the RCA jacks on the EVM based on boot time selection, as shown in Table 2-11 for DM6446 (For DM355, only one DAC is available and it maps to composite output.)
- Different combinations of interface, mode, vid0, vid1, osd0 and osd1

Table 2-11. Fbdev Driver - EVM Output Signal Mapping

Output Selection	DAC A	DAC B	DAC C	DAC D
Composite	Composite	Composite	Composite	Composite
Component	Y	Pb	Pr	Composite
S-Video	Composite	Y	C	Composite

2.1.2.1 Support and Constraints

Features Not Supported	The driver does NOT support analog RGB output (applicable only for DM355)
Constraints	Dynamic removal of the module is not allowed when the FB console is in use
Supported System Calls	open(), close(), ioctl(), mmap() and munmap()
Supported IOCTLs	The Fbdev Display Driver supports the IOCTLs shown in Table 2-12

Table 2-12. Fbdev Display Driver Supported IOCTLs

Constant	Description
FBIO_ENABLE_DISABLE_WIN	Allocates memory for the specified window and enables it. Memory allocated for the window is: xres * yres * bits_per_pixel *NUMBUFS. xres, yres and bits_per_pixel are variables set in fb_var_screeninfo, where NUMBUFS is 3 for VID windows and 2 for bitmap windows.
FBIO_SET_BITMAP_BLEND_FACTOR	Sets the blend factor for bitmap window. Amount of blending (i.e. relative amount of video data vs. bitmap data) at each pixel is determined by the blending factor.
FBIO_SET_BITMAP_WIN_RAM_CLUT	Sets up the color look up table in RAM per user specifications
FBIO_ENABLE_DISABLE_ATTR_WIN	Sets OSD window 1 as attribute window or bitmap window.
FBIO_GET_BLINK_INTERVAL	Gets the existing blinking interval of the attribute window and value of the vpbe_blink_enable flag.
FBIO_SET_BLINK_INTERVAL	Sets the blinking of the attribute window
	Blinking interval can be set in the structure.
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_INTERFACE	Sets the display interface configuration to the default mode
FBIO_GET_INTERFACE	Gets the current display interface
FBIO_QUERY_MODE	Query whether the passed interface and mode type are supported by the driver
FBIO_SET_MODE	Sets mode. If standard mode is selected, timing parameters are not applicable. Driver will find suitable mode from mode list using the vmode, xres, yres and fps parameters of the passed structure.
FBIO_GET_MODE	Get the current hardware status
FBIO_SET_VENC_CLK_SOURCE	Sets the clock source for the VENC module
FBIO_SET_BACKG_COLOR	Sets the background color. The window should be disabled before using this command.
FBIOGET_VSCREENINFO	Gets the variable screen information of the frame buffer. This command can be used for each frame buffer window.
FBIOPUT_VSCREENINFO	Sets variable screen parameters for frame buffer, including the window input format (resolution and bits per pixel).
FBIO_PUT_CMAP	Sets up pseudo palette
FBIO_GET_CMAP	Return the CMAP to the application
FBIO_PAN_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_GET_CON2FBMAP	Gets con2fbmap structure
FBIO_SET_CON2FBMAP	Sets con2fbmap structure
FBIOBLANK	Used to disable the entire display window
FBIO_WAITFORVSYNC	Synchronizes the buffered display by waiting for vsync before the call returns.
FBIO_SETATTRIBUTE	Programs the attribute window OSD1. The arguments take in rectangle coordinates and fill it with the required transparency value.
FBIO_SETPOS	Changes the starting X and Y coordinate of the desired video or OSD plane
FBIO_SETZOOM	Specifies the ZOOM parameters (identity, 2x or 4x) for the corresponding display plane
FBIO_SETPOSX	Used to program the start position of the planes in X direction
FBIO_SETPOSY	Used to program the start position of the planes in Y direction

2.1.2.2 Performance and Benchmarks

Memory requirements for the Fbdev Display Driver are not shown, as it is not recommended to build Fbdev as a dynamic kernel module.

DM355 EVM

Table 2-13. Fbdev Display Driver Performance Data (LLD) - DM355

Video Format	Frames per Second
TVP5146 (NTSC)	29.997
TVP5146 (PAL)	25
LCD (LogicPD)	64.704

Table 2-14. Fbdev Display Driver Performance Data (Server) - DM355

Video Format	Frames per Second
TVP5146 (NTSC)	29.991
TVP5146 (PAL)	24.995
LCD (LogicPD)	64.749

Table 2-15. Fbdev Display Driver Performance (Real-Time) data - DM355

Video Format	Frames per Second
TVP5146 (NTSC)	29.99
TVP5146 (PAL)	25.019
LCD (LogicPD)	64.572

DM6446 EVM

Table 2-16. Fbdev Display Driver Performance Data (LLD) - DM6446

Input Video Format	Frames per Second
TVP5146 (NTSC)	29.986
TVP5146 (PAL)	25.029
LCD (LogicPD)	64.818

Table 2-17. Fbdev Display Driver Performance Data (Server) - DM6446

Input Video Format	Frames per Second
TVP5146 (NTSC)	29.977
TVP5146 (PAL)	25
LCD (LogicPD)	64.832

Table 2-18. Fbdev Display Driver Performance (Real-Time) data - DM6446

Input Video Format	Frames per Second
TVP5146 (NTSC)	29.95
TVP5146 (PAL)	25.03
LCD (LogicPD)	64.775

2.1.3 Video Capture Driver

The video capture driver supports the following features:

- Supports decoded video input in YUV422 format (UYVY or YUYV) with the help of an external decoder
- NTSC and PAL video input through an external decoder, including auto sensing capability
- Raw data input using MT9T001 or MT9T031 (3 Megapixel) image sensors. Supports VGA (640 x 480), SD (NTSC/PAL/480p/576p) and HD (720p) resolutions
- Multi-buffered input with a minimum of three buffers, and support for more at runtime
- Supports 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

2.1.3.1 Features Not Supported

The video capture driver does not support the following features:

- The driver does NOT support the histogram feature

2.1.3.2 Constraints

The following are the constraints of the video capture driver:

- The driver exhibits intermittent I2C read/write failures
  - An I2C read/write may fail with MT9T001. In the MT9T001 driver, a temporary workaround is added to reduce I2C write failure frequency
  - Captured data may be incorrect if MT9T001 registers are not configured properly
  - I2C read back values may be incorrect

2.1.3.3 Supported System Calls

The video capture driver supports the following system calls: open(), close(), ioctl(), mmap() and munmap().

2.1.3.4 Supported IOCTLs

The video capture driver supports IOCTLs shown in Table 2-19.

Table 2-19. Video Capture Driver Supported IOCTLs

Constant	Description
VIDIOC_CROPCAP	Information about the video cropping and scaling abilities.
VIDIOC_ENUMOUTPUT	Enumerate video inputs.
VIDIOC_ENUMOUTPUT	Enumerate video outputs.
VIDIOC_ENUMSTD	Enumerate 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.

2.1.3.5 Performance and Benchmarks

DM355 EVM

Table 2-20 shows the memory requirements for the V4L2 capture driver modules built for the DM355 device.

Table 2-20. V4L2 Capture Driver Memory Statistics - DM355

Configuration	Memory Statistics (in Bytes)			
	Program Memory	Data Memory		
		Initialized	Uninitialized	Total
davinci_capture.ko	27232	7496	536	35264
tvp5146.ko	4120	256	464	4840
mt9t001.ko	9756	256	472	10484

Table 2-21. V4L2 Capture Driver Performance Data (LLD) - DM355

Input Video Format	Frames per Second
TVP5146 (NTSC)	29.983
TVP5146 (PAL)	25.015
MT9T031 (480p)	29.922
MT9T031 (576p)	24.958

Table 2-22. V4L2 Capture Driver Performance Data (Server) - DM355

Input Video Format	Frames per Second
TVP5146 (NTSC)	29.992
TVP5146 (PAL)	25.002
MT9T031 (480p)	29.926
MT9T031 (576p)	24.956

Table 2-23. V4L2 Capture Driver Performance Data (Real-Time) - DM355

Input Video Format	Frames per Second
TVP5146 (NTSC)	30.001
TVP5146 (PAL)	25.022
MT9T031 (480p)	29.925
MT9T031 (576p)	24.964

DM6446 EVM

Table 2-24 shows the memory requirements for the V4L2 capture driver modules built for the DM6446 device.

Table 2-24. V4L2 Capture Driver Memory Statistics - DM6446

Configuration	Memory Statistics (in Bytes)			
	Program Memory	Data Memory		
		Initialized	Uninitialized	Total
davinci_capture.ko	19572	2384	536	22492
tvp5146.ko	4120	256	440	4840
mt9t001.ko	9756	256	448	10460

Table 2-25. V4L2 Capture Driver Performance Data (LLD) - DM6446

Input Video Format	Frames per Second
TVP5146 (NTSC)	29.986
TVP5146 (PAL)	25.012

Table 2-26. V4L2 Capture Driver Performance Data (Server) - DM6446

Input Video Format	Frames per Second
TVP5146 (NTSC)	30.001
TVP5146 (PAL)	25.027

Table 2-27. V4L2 Capture Driver Performance Data (Real-Time) - DM6446

Input Video Format	Frames per Second
TVP5146 (NTSC)	29.988
TVP5146 (PAL)	25.015

2.1.4 IPIPE Driver

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

- Previewing the capture input
- Resizing the capture input
- Slicing of an image
- Supports buffer access mechanism through memory mapping

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

2.1.4.1 Support and Constraints

Features Not Supported — The driver does not support multichannels. When multiple open() calls are made, the driver returns an error.

Constraints — Defect pixel correction (DFC) feature is disabled as a workaround for the IPIPE freeze issue when the application does not wait long enough between IPIPE_SET_PARAM and IPIPE_START IOCTLs.

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

Supported IOCTLs — See Table 2-28

Table 2-28. IPIPE Driver Supported IOCTLs

Constant	Description
IPIPE_REQBUF	Request frame buffers to be allocated by the IPIPE module
IPIPE_QUERYBUF	Query the status of a particular frame buffer
IPIPE_SET_PARAM	Set the IPIPE hardware parameters
IPIPE_GET_PARAM	Get the IPIPE hardware parameters
IPIPE_START	Submit a IPIPE processing request using parameters set by IPIPE_SET_PARAM

2.1.4.2 Performance and Benchmarks

DM355 EVM

Table 2-29 shows the memory requirements for the IPIPE Driver module built for DM355.

Table 2-29. IPIPE Driver Memory Statistics - DM355

Configuration	Memory Statistics (in Bytes)			
	Program Memory	Data Memory		
		Initialized	Uninitialized	Total
dm355_ipipe_driver.ko	21392	1368	192	22952

2.1.5 Previewer Driver

The previewer driver does not support DM355 EVM. For the same previewing and resizing functionalities on DM355, the IPIPE driver should be used.

The DM6446 EVM previewer driver supports the following features:

- Input image in Bayer pattern
- Supports various hardware configurations such as CFA interpolation, DFC subtract, etc
- Accepts input from SDRAM or DDRAM
- Converts Bayer pattern input image to YCbCr 4:2:2 format
- Supports buffer access mechanism through memory mapping

2.1.5.1 Support and Constraints

Features Not Supported	None
Constraints	• The previewer driver will not work with the video port enabled in the CCDC driver • The previewer driver and H3A drivers cannot be used together
Supported System Calls	open(), close(), ioctl(), mmap() and mumap()
Supported IOCTLs	The previewer driver supports the IOCTLs shown in Table 2-30

Table 2-30. Previewer Driver Supported IOCTLs

Constant	Description
PREV_REQBUF	Request frame buffers to be allocated by the PREV module
PREV_QUERYBUF	Request physical address of buffers allocated by the PREV_REQBUF ioctl.
PREV_SET_PARAM	Set the Previewer parameters
PREV_GET_PARAM	Get the Previewer hardware parameters
PREV_GET_STATUS	Get the current status of the hardware
PREV_PREVIEW	Submits a previewing task to the hardware
	Returns the size reduction in the output image compared to input image in
PREV_GET_CROPSIZE	Terms of number of pixels per line and number of lines depending on features enabled
PREV_SET_EXP	Set allowable delay between consecutive read requests from Previewer module

2.1.5.2 DM6446 EVM Performance and Benchmarks

Table 2-31 shows the memory requirements for the previewer driver module built for DM6446.

Table 2-31. Previewer Driver Memory Statistics - DM6446

Configuration	Memory Statistics (in Bytes)			
	Program Memory	Data Memory		
		Initialized	Uninitialized	Total
davinci_previewer_driver.ko	13712	648	188	14548

2.1.6 Resizer Driver

The Resizer Driver does not support DM355 EVM. For the same previewing and resizing functionalities on DM355, IPIPE Driver should be used.

The Resizer Driver supports the following features:

- Supports YUV422 color interleaved and 8-bit color separate data input formats
- Input from CCDC, SDRAM or DDRAM
- The driver can be used by multiple applications

- Supports upscaling/downscaling multiplier from one-fourth to 4x

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 resizer driver supports the IOCTLs shown in Table 2-32

Table 2-32. Resizer Driver Supported IOCTLs

Constant	Description
RSZ_REQBUF	Request frame buffers to be allocated by the RSZ module
RSZ_QUERYBUF	Request physical address of buffers allocated by the RSZ_REQBUF
RSZ_S_PRIORITY	Set the current priority setting of the logic channel identified by fd.
RSZ_S_PARAM	Set the Resizer hardware parameters associated with this logic channel
RSZ_G_PARAM	Get the Resizer hardware parameters associated with this logic channel
RSZ_G_STATUS	Get the current status of the hardware
RSZ_RESIZE	submits a resizing task to the logic channel associated with fd
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	This ioctl is used to set allowable delay between consecutive read requests from Resizer module

2.1.6.2 Performance and Benchmarks

DM6446 EVM

Table 2-33 shows the memory requirements for the resizer driver module built for DM6446.

Table 2-33. Resizer Driver Memory Statistics - DM6446

Configuration	Memory Statistics (in Bytes)			
	Program Memory	Data Memory		
		Initialized	Uninitialized	Total
davinci_rsz_driver.ko	11052	648	204	11904

2.1.7 H3A Driver

The H3A - Auto expose/auto white balance (aew) and auto focus (af) drivers support the following features:

- Supports image in Bayer pattern
- Supports Input from CCD controller
- Tested with CCDC Micron Sensors - MT9T031

The H3A driver is not tested on DM6446 EVM.

2.1.7.1 Support and Constraints

Features Not Supported	None
Constraints	None
Supported System Calls	open(), close(), read() and ioctl()
Supported IOCTLs	The H3A Driver supports IOCTLs shown in Table 2-34.

Table 2-34. H3A Driver Supported IOCTLs

Constant	Description
AEW_S_PARAM	This ioctl is used to set the following parameters/modules of the AEW hardware: • Window parameters • Black window parameters • Saturation check module • A-Law compression module
AEW_G_PARAM	This ioctl is used to get the AEW hardware settings.
AEW_ENABLE	This ioctl is used to enable AEW Engine with parameters set by AEW_S_PARAM.
AEW_DISABLE	This ioctl is used to disable AEW Engine.

2.1.7.2 Performance and Benchmarks

DM355 EVM

Table 2-35 shows the memory requirements for the H3A driver modules built for DM355.

Table 2-35. H3A Driver Memory Statistics - DM355

Configuration	Memory Statistics (in Bytes)			
	Program Memory	Data Memory		
		Initialized	Uninitialized	Total
dm355_aew_driver.ko	10564	672	92	11328
dm355_af_driver.ko	11216	672	92	11980

DM6446 EVM

Table 2-36 shows the memory requirements for the H3A driver modules built for DM6446.

Table 2-36. H3A Driver Memory Statistics - DM6446

Configuration	Memory Statistics (in Bytes)			
	Program Memory	Data Memory		
		Initialized	Uninitialized	Total
davinci_aew_driver.ko	10356	648	92	11096
davinci_af_driver.ko	10592	648	92	11332

2.2 Audio Driver

The DaVinci audio driver is a char driver that supports the Audio Serial Port (ASP) peripheral. Figure 2-2 shows the driver architecture. It is compatible with open sound system (OSS) interface version 1.11. The OSS driver can be accessed from user space as /dev/dsp and /dev/mixer.

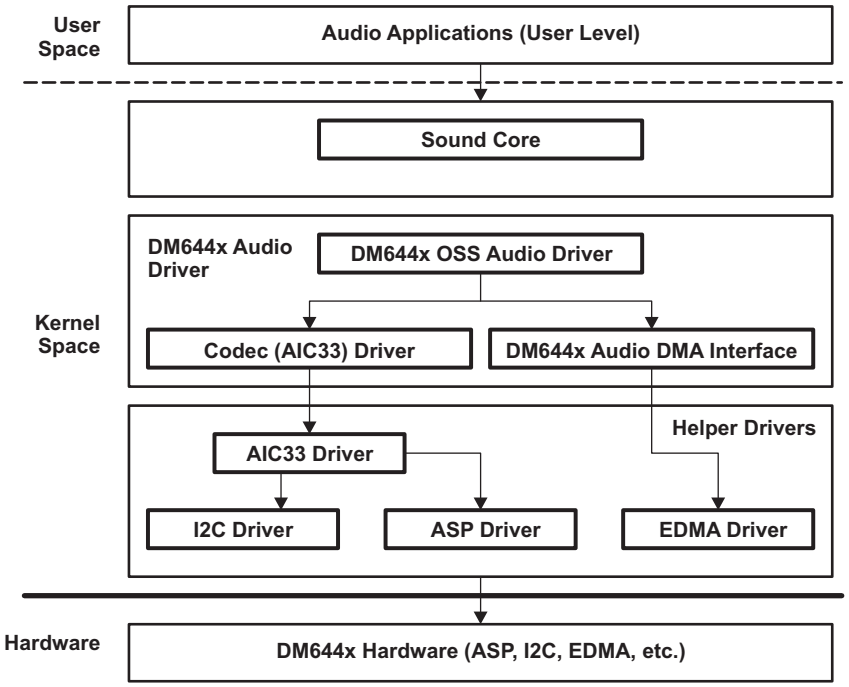

Figure 2-2. DaVinci Audio Driver Architecture

2.2.1 Driver Features

The DaVinci audio driver supports the following features:

- Supports record and playback capabilities
- Supports both DM355 and DM6446 EVMs
- Open sound system (OSS) version 1.11 compatible with /dev/sound/dsp and /dev/sound/mixer
- Audio channel set to stereo mode by default
- Supports AFMT_S16_LE (16 bit) audio format only

2.2.2 Support and Constraints

Features Not Supported

The audio driver does not support the following features:

- The audio driver does not support DSP, right-justified, left-justified features
- The audio driver does not support other audio formats.

Constraints

The left and right channels are reversed at a sample rate of 8 kHz.

Supported System Calls

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

Supported IOCTLs

Table 2-37 shows the IOCTLs supported by the DaVinci audio driver.

Table 2-37. DaVinci Audio Driver Supported IOCTLs

Constant	Description
OSS_GETVERSION	Identify the driver version
SNDCTL_DSP_STEREO	Set stereo or mo no channel
SNDCTL_DSP_CHANNELS	Set the number of audio channels
SNDCTL_DSP_SPEED	Set the sampling rate
SNDCTL_DSP_SETFMT	Select the sample format

Table 2-37. DaVinci Audio Driver Supported IOCTLs (continued)

Constant	Description
SNDCTL_DSP_GETFMTS	Return a list of natively supported sample formats
SNDCTL_DSP_GETBLKSIZE	Return the block size that the sound driver uses for data transfers
SNDCTL_DSP_GETCAPS	Return a bitmask identifying the capabilities of a sound card DSP device
SNDCTL_DSP_SETFRAGMENT	Request 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	Notify the driver that there is likely to be a pause in the output
SNDCTL_DSP_GETTRIGGER	Return the current trigger bits
SNDCTL_DSP_SETTRIGGER	Start audio recording and/or playback in sync
SNDCTL_DSP_GETOPTR	Return the current playback pointer
SNDCTL_DSP_GETIPTR	Return the current recording pointer
SNDCTL_DSP_GETOSPACE	Return the number of currently recorded bytes
SNDCTL_DSP_GETISPACE	Return the number of currently recorded bytes
SNDCTL_DSP_NONBLOCK	Sets non-blocking mode
SNDCTL_DSP_RESET	Device reset
SOUND_MIXER_VOLUME	Master output level
SOUND_MIXER_LINE	Line input
SOUND_MIXER_MIC	Microphone input
SOUND_MIXER_RECSRC	Indicates which channels are currently selected as the recording source
SOUND_MIXER_BASS	Bass tone control
SOUND_MIXER_TREBLE	Treble tone control
SOUND_MIXER_IGAIN	Input gain level
SOUND_MIXER_OGAIN	Output gain level
SOUND_MIXER_RECMASK	Return a bitmask indicating channels that can be used as a recording source
SOUND_MIXER_DEVMASK	Return a bitmask indicating which channels are supported by the mixer
SOUND_MIXER_CAPS	Return 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	Read the sampling frequency
SOUND_PCM_READ_BITS	Return a list of natively supported sample formats

2.2.3 Performance and Benchmarks

DM355 EVM

Table 2-38 shows the memory requirements for the audio driver module built for DM355.

Table 2-38. Audio Driver Memory Statistics - DM355

Configuration	Memory Statistics (in Bytes)			
	Program Memory	Data Memory		
		Initialized	Uninitialized	Total
dm644x_audio_driver.ko	21684	1200	1176	24060

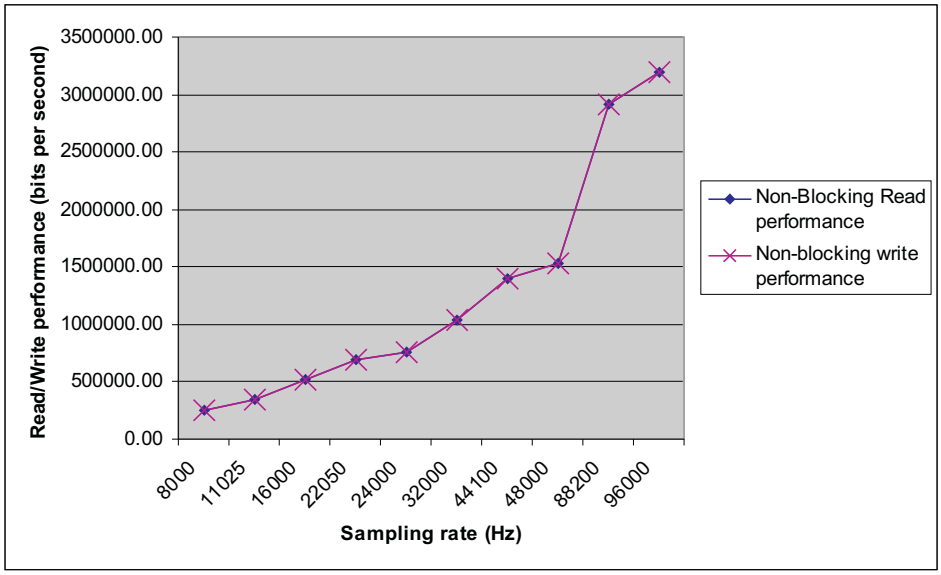

Figure 2-3. Audio Driver Runtime Performance Characteristics (LLD) - DM355

Table 2-39. Audio Driver Runtime Performance Data (LLD) - DM355

Test Setup Information	Sampling Rate (Hz)	Read Rate (bits/sec)	Write Rate (bits/sec)
ARM Frequency = 216 MHz DDR Frequency = 171 MHz	8000	256000.00	256000.00
	11025	351648.35	351648.35
	16000	516129.03	516129.03
	22050	711111.11	711111.11
	24000	761905.76	761905.76
	32000	1000000.00	1000000.00
	44100	1454545.45	1454545.45
	48000	1523809.52	1523809.52
	88200	2909090.90	2909090.90
	96000	3200000.00	3200000.00

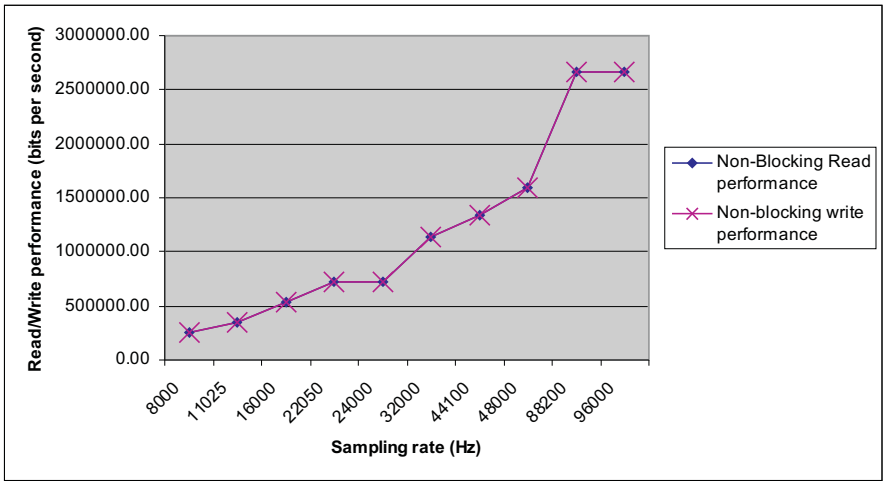

Figure 2-4. Audio Driver Runtime Performance Characteristics (Server) - DM355

Table 2-40. Audio Driver Runtime Performance Data (Server) - DM355

Test setup Information	Sampling Rate (Hz)	Read Rate (bits/sec)	Write Rate (bits/sec)
ARM Frequency = 216 MHz DDR Frequency = 171 MHz	8000	258064.52	258064.52
	11025	347826.10	347826.10
	16000	533333.33	533333.33
	22050	727272.72	727272.72
	24000	727272.72	727272.72
	32000	1142857.14	1142857.14
	44100	1333333.33	1333333.33
	48000	1600000.00	1600000.00
	88200	2666666.67	2666666.67
	96000	2666666.67	2666666.67

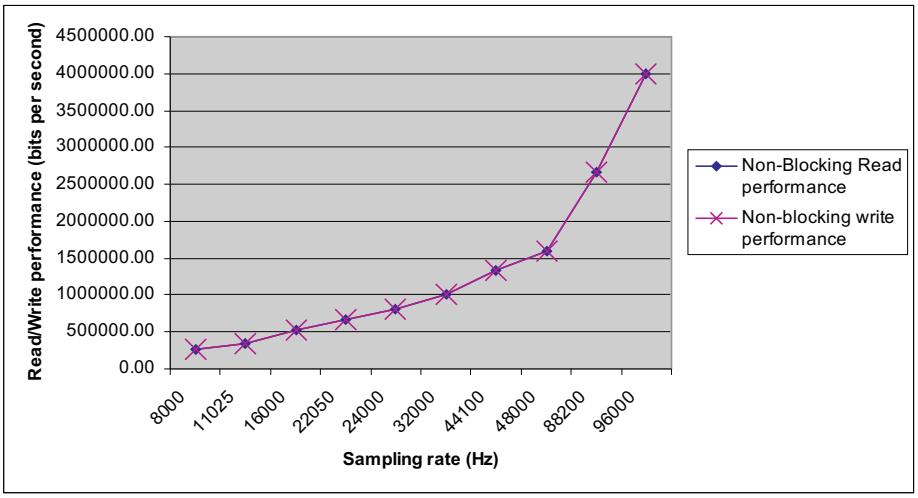

Figure 2-5. Audio Driver Runtime Performance Characteristics (Real-Time) - DM355

Table 2-41. Audio Driver Runtime Performance Data (Real-Time) - DM355

Test setup Information	Sampling Rate (Hz)	Read Rate (bits/sec)	Write Rate (bits/sec)
ARM Frequency = 216 MHz DDR Frequency = 171MHz	8000.00	258064.52	258064.52
	11025.00	347826.10	347826.10
	16000.00	533333.33	533333.33
	22050.00	666666.67	666666.67
	24000.00	800000.00	800000.00
	32000.00	1000000.00	1000000.00
	44100.00	1333333.33	1333333.33
	48000.00	1600000.00	1600000.00
	88200.00	2666666.67	2666666.67
	96000.00	4000000.00	4000000.00

DM6446 EVM

Table 2-42 shows the memory requirements for the audio driver module built for DM6446.

Table 2-42. Audio Driver Memory Statistics - DM6446

Configuration	Memory Statistics (in Bytes)			
	Program Memory	Data Memory		
		Initialized	Uninitialized	Total
dm644x_audio_driver.ko	21520	1176	1176	23872

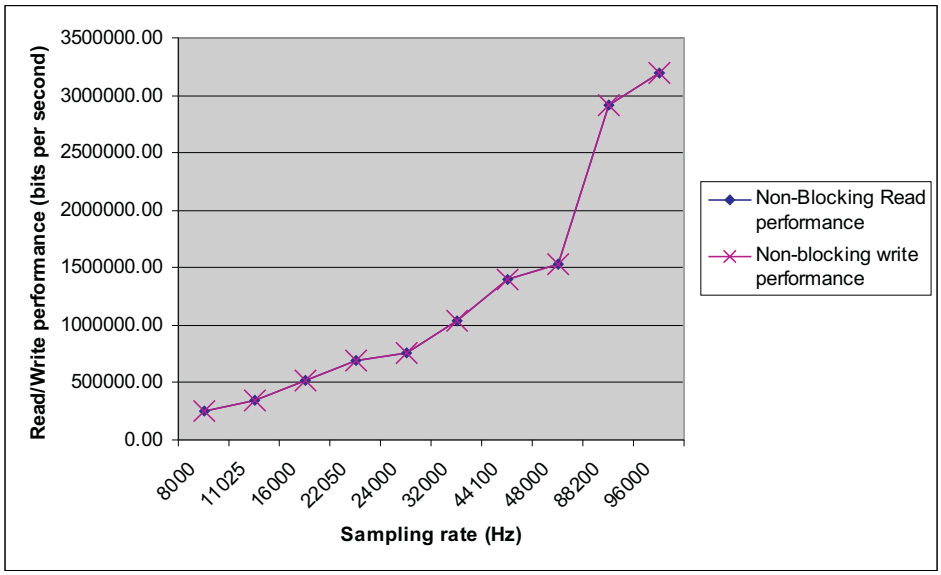

Figure 2-6. Audio Driver Runtime Performance Characteristics (LLD) - DM6446

Table 2-43. Audio Driver Runtime Performance Data (LLD) - DM6446

Test setup Information	Sampling Rate (Hz)	Read Rate (bits/sec)	Write Rate (bits/sec)
ARM Frequency = 297 MHz DDR Frequency = 162 MHz	8000.00	256000.00	256000.00
	11025.00	351648.35	351648.35
	16000.00	516129.03	516129.03
	22050.00	695652.17	695652.17
	24000.00	761904.76	761904.76
	32000.00	1032258.06	1032258.06
	44100.00	1391304.34	1391304.34
	48000.00	1523809.52	1523809.52
	88200.00	2909090.90	2909090.90
	96000.00	3200000.00	3200000.00

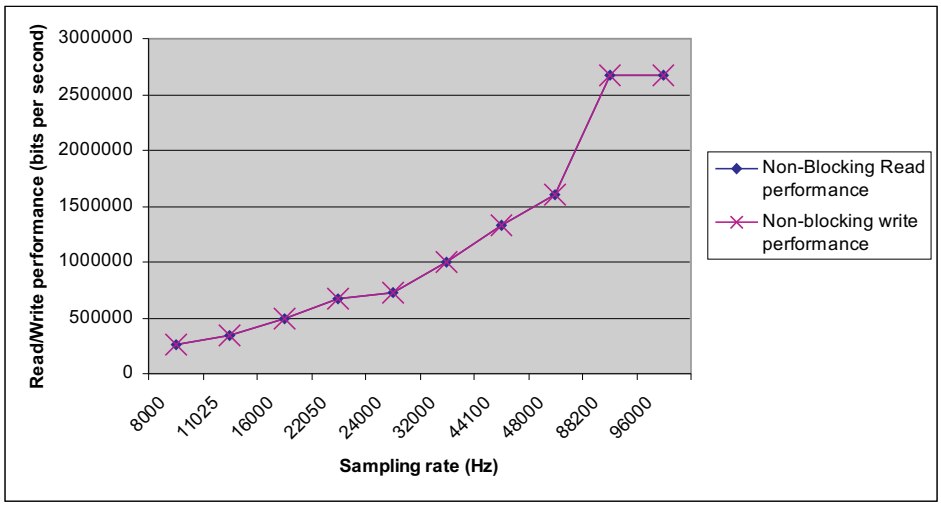

Figure 2-7. Audio Driver Runtime Performance Characteristics (Server) - DM6446

Table 2-44. Audio Driver Runtime Performance Data (Server) - DM6446

Test setup Information	Sampling Rate (Hz)	Read Rate (bits/sec)	Write Rate (bits/sec)
ARM Frequency = 297 MHz DDR Frequency = 162 MHz	8000.00	258064.52	258064.52
	11025.00	347826.10	347826.10
	16000.00	500000.00	500000.00
	22050.00	666666.67	666666.67
	24000.00	727272.72	727272.72
	32000.00	1000000.00	1000000.00
	44100.00	1333333.33	1333333.33
	48000.00	1600000.00	1600000.00
	88200.00	2666666.67	2666666.67
	96000.00	2666666.67	2666666.67

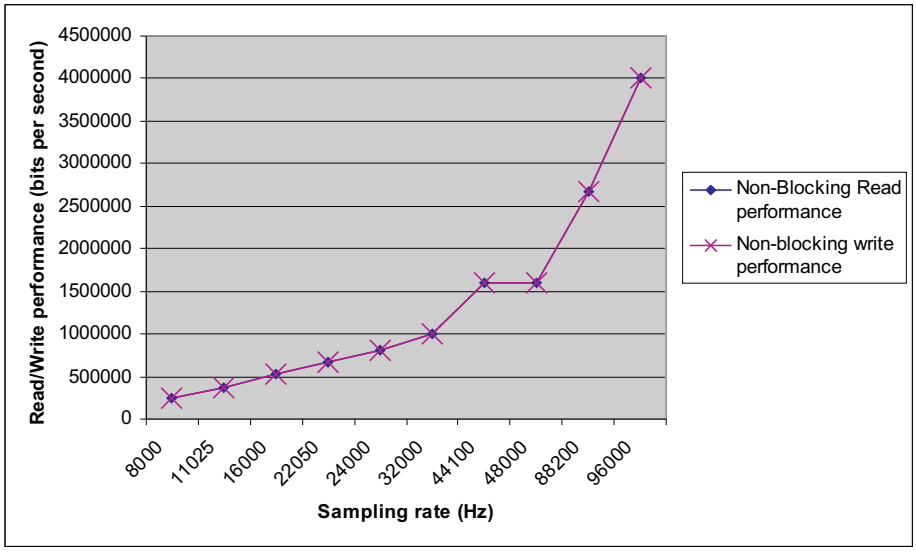

Figure 2-8. Audio Driver Runtime Performance Characteristics (Real-Time) - DM6446

Table 2-45. Audio Driver Runtime Performance Data (Real-Time) - DM6446

Test Setup Information	Sampling Rate (Hz)	Read Rate (bits/sec)	Write Rate (bits/sec)
ARM Frequency = 297 MHz DDR Frequency = 162 MHz	8000.00	250000.00	250000.00
	11025.00	363636.36	363636.36
	16000.00	533333.33	533333.33
	22050.00	666666.67	666666.67
	24000.00	800000.00	800000.00
	32000.00	1000000.00	1000000.00
	44100.00	1600000.00	1600000.00
	48000.00	1600000.00	1600000.00
	88200.00	2666666.67	2666666.67
	96000.00	4000000.00	4000000.00

2.3 Ethernet Drivers

This section includes DM9000A Ethernet Driver for DM355 and the CPMAC Ethernet Driver for DM6336.

2.3.1 DM9000A Ethernet Driver for DM355

The DM355 EVM uses a Davicom DM9000A 10/100M Ethernet chip hardware (see Figure 2-9). It is a 16/8 bit Ethernet controller with general processor interface and an internal SRAM of 16 KB size for transmit and receive FIFO buffer. The chip uses 16-bit data bus to DM355 using DM355 EMIF, providing higher data throughput. It uses standard control interface to the ARM processor (INT, CS, IOR, IOW) in DM355.

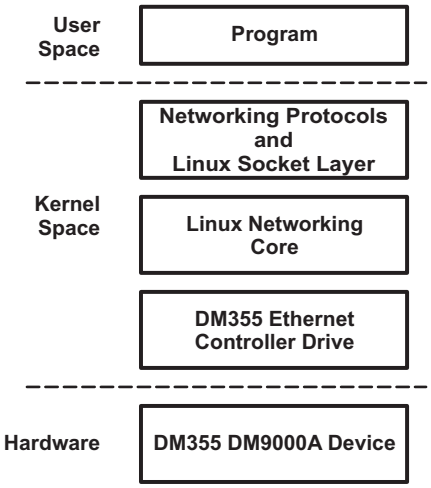

Figure 2-9. Linux Kernel DM9000A Ethernet Driver Architecture

The DM9000A Ethernet driver supports the following features:

- 10/100 Mbps Speed
- Auto Negotiation
- Multicast and Broadcast
- Promiscuous mode
- Full and Half Duplex modes
- MAC Address Storage on a Dedicated EEPROM, controlled by DM9000A Controller

2.3.1.1 Support and Constraints

Features Not Supported

The DM9000A Ethernet Driver does not support following features:

- The Driver does not support 1Gbps speed.
- The Driver does not support 2.6 NAPI Interface.

Constraints

DM9000A Performance numbers are low compared to the numbers from CPMAC driver on DM6446. The performance numbers are expected to be lower when compared to the performance on DM6446, due to the following facts: The CPU speed on DM355 is lower than that of the DM6446. The DM9000A Ethernet Controller is on EMIF, with other peripherals such as NAND, while CPMAC on DM6446 is connected to the PHY.

Supported System Calls

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

Supported IOCTLs

None

2.3.1.2 Performance and Benchmarks

Table 2-46 shows the memory requirements for the DM9000A Ethernet Driver module.

Table 2-46. DM9000A Driver Memory Statistics

Configuration	Memory Statistics (in Bytes)			
	Program Memory	Data Memory		
		Initialized	Uninitialized	Total
dm9000.ko	10184	524	0	10708

2.3.1.2.1 DM9000A Driver Performance: Test setup

- **Hardware used:** DM355 EVM, Windows XP/SP2 PC, 10/100 Ethernet Switch, CAT 5E Ethernet cables
- **Root Filesystem used:** YAFFS2 (/dev/mtdblock3)
- **Mode tested:** Low Latency Desktop, Server and Real-Time preemption modes
- **Test Commands:**
  - Server side command: "iperf -s"
  - Client side command : "iperf -c <server ip> -w<window size> -t60"
  - iperf version 1.70 is used
  - DM9000A Ethernet driver is configured for Auto Negotiation
- **Test sequence:**
  1. Connect the Windows XP PC to a 10/100 switch
  2. Connect the EVM to the same switch
  3. Configure each interface with an appropriate IP Address
  4. Start “iperf”, using the server side command, on the EVM
  5. Start “iperf”, using the client side command, on the Windows PC
  6. Run this test 5 times, each time changing the window size to a different size (16, 32, 64, 128, and 212)
  7. The Bandwidth data is collected from the “iperf”, in the server mode, running on the EVM

2.3.1.2.2 DM9000A Driver Performance: Low Latency Desktop Mode

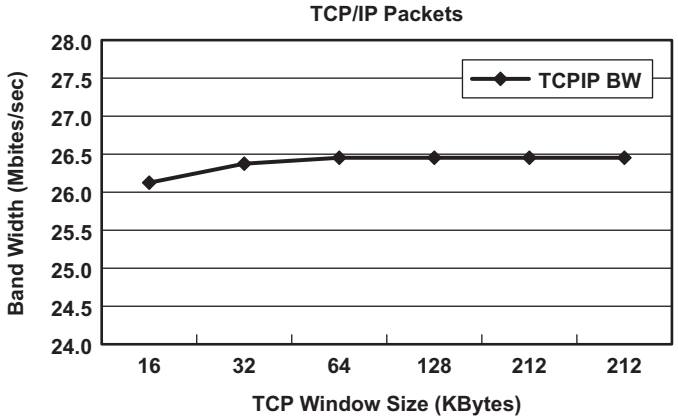

Figure 2-10. DM9000A Driver Runtime Performance Characteristics (LLD)

Table 2-47. DM9000A Driver Performance Using IPerf (LLD)

Test Setup Information	TCP Window Size in KBytes	TCP/IP Packets		
		Bandwidth in Mbits/sec	Transfer Size in MBytes	Interval in Seconds
ARM Frequency = 216 MHz DDR Frequency = 171 MHz	16	26.1	186	60
	32	26.3	188	60
	64	26.4	189	60
	128	26.4	189	60
	212	26.4	189	60
	212	26.4	189	60

Test Setup:

- See the Section 2.3.1.2.1, DM9000A Driver Performance: Test setup.
- Rebuild the Linux Kernel for low latency desktop preemption mode.

2.3.1.2.3 DM9000A Driver Performance: Server Mode

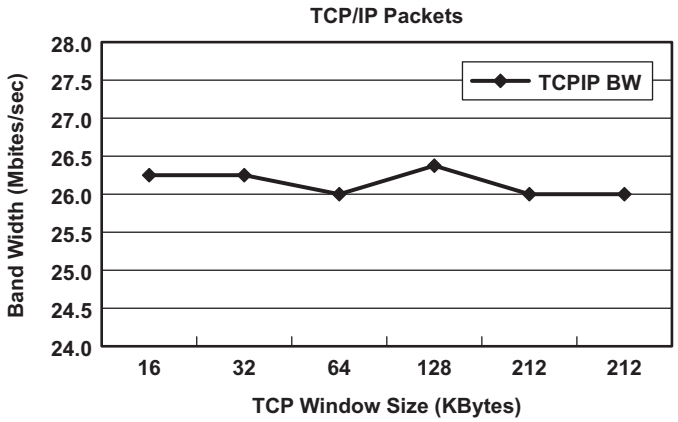

Figure 2-11. DM9000A Driver Runtime Performance Characteristics (Server)

Table 2-48. DM9000A Driver Performance Using IPerf (Server)

Test Setup Information	TCP Window Size in KBytes	TCP/IP Packets		
		Bandwidth in Mbits/sec	Transfer size in MBytes	Interval in Seconds
ARM Frequency = 216 MHz DDR Frequency = 171 MHz	16	26.2	187	60
	32	26.2	188	60
	64	25.9	186	60
	128	26.3	188	60
	212	26	186	60
	212	26	186	60

Test Setup:

- See Section 2.3.1.2.1, DM9000A Driver Performance: Test setup.
- Rebuild the linux kernel for server preemption mode.

2.3.1.2.4 DM9000A Driver Performance: Real-Time Preemption Mode

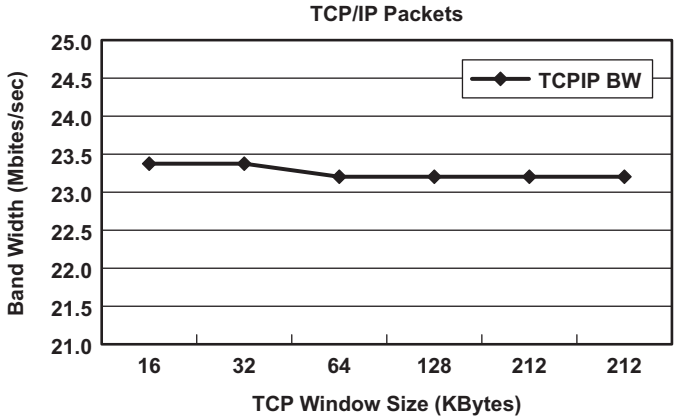

Figure 2-12. DM9000A Driver Runtime Performance Characteristics (Real-Time)

Table 2-49. DM9000A Driver Performance Using IPerf (Real-Time)

Test Setup Information	TCP Window Size in KBytes	TCP/IP Packets		
		Bandwidth in Mbits/sec	Transfer Size in MBytes	Interval in Seconds
ARM Frequency = 216 MHz DDR Frequency = 171 MHz	16	23.3	166	60
	32	23.3	167	60
	64	23.2	166	60
	128	23.2	166	60
	212	23.2	166	60
	212	23.2	166	60

Test Setup:

- See Section 2.3.1.2.1, DM9000A Driver Performance: Test setup.
- Rebuild the Linux Kernel for Real-Time preemption mode.

2.3.2 CPMAC Ethernet Driver for DM6446

The Ethernet driver supports the Linux 2.6 NAPI Interface (see Figure 2-13). The driver talks to the PHY connected to the MAC layer using the MDIO interface.

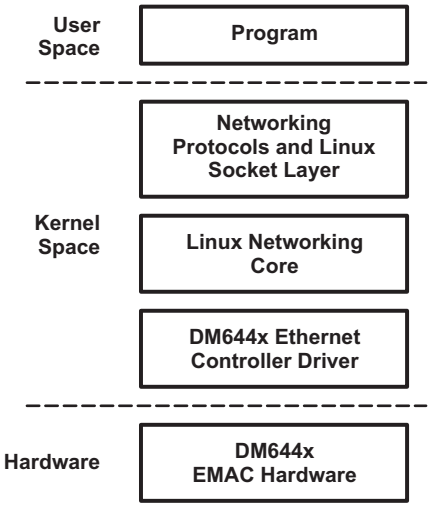

Figure 2-13. Linux Kernel Ethernet Driver Architecture

2.3.2.1 Driver Features

The CPMAC Ethernet driver supports the following features:

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

2.3.2.2 Support and Constraints

Features Not Supported	The CPMAC Ethernet driver does not support 1 Gbps speed.
Constraints	The DaVinci CPMAC Ethernet driver does NOT support the following hardware features: • Multiple receive and transmit channels; the driver supports only one receive and one transmit channel • QoS support with VLAN tag discrimination using multiple RX channels • Receive Flow Control using RXnFREEBUFFER and RXnFLOWTHRESH registers • Interrupt pacing feature
Supported System Calls	open(), close(), read(), write(), and ioctl()
Supported IOCTLs	None

2.3.2.3 Performance and Benchmarks

Table 2-50 shows the memory requirements for the CPMAC Ethernet Driver Module.

Table 2-50. CPMAC Ethernet Driver Memory Statistics

Configuration	Memory Statistics (in Bytes)			
	Program Memory	Data Memory		
		Initialized	Uninitialized	Total
davinci_emac_driver.ko	47204	972	108	48284

2.3.2.3.1 CPMAC Ethernet Driver Performance: Test Setup

- **Hardware used:** DM6446 EVM, Windows XP/SP2 PC, 10/100 Ethernet Switch, CAT 5E Ethernet cables
- **Root Filesystem used:** ATA – EXT3 (/dev/hda1)
- **Mode tested:** Low Latency Desktop, Server and Real-Time preemption modes
- **Test Commands:**
  - Server side command: "iperf -s"
  - Client side command : "iperf -c <server ip> -w<window size> -t60"
  - iperf version 1.70 is used
  - DM9000A Ethernet driver is configured for Auto Negotiation
- **Test sequence:**
  1. Connect the Windows XP PC to a 10/100 switch
  2. Connect the EVM to the same switch
  3. Configure each interface with an appropriate IP Address
  4. Start “iperf”, using the server side command, on the EVM
  5. Start “iperf”, using the client side command, on the Windows PC
  6. Run this test 5 times, each time changing the window size to a different size (16, 32, 64, 128, and 212)
  7. The Bandwidth data is collected from the “iperf”, in the server mode, running on the EVM

2.3.2.3.2 CPMAC Ethernet Driver Performance: Low Latency Desktom Preemption Mode

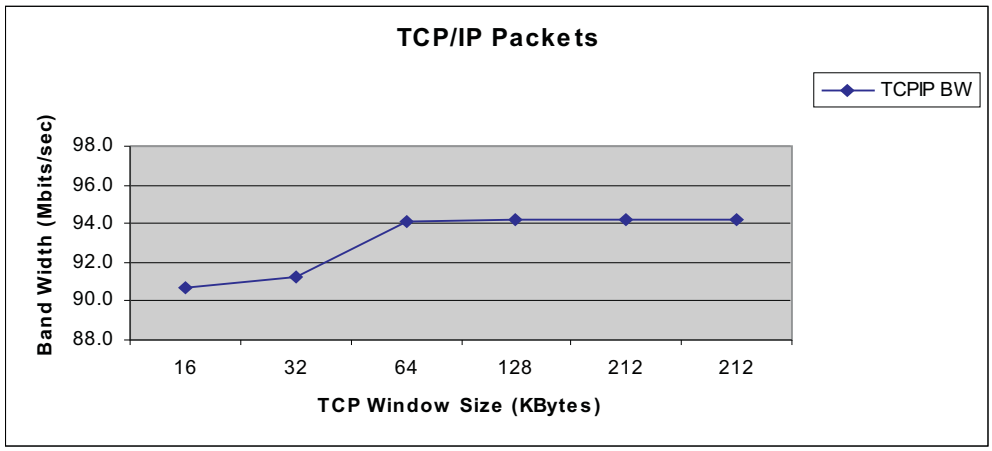

Figure 2-14. CPMAC Ethernet Driver Runtime Performance Characteristics (LLD)

Table 2-51. CPMAC Ethernet Driver Performance Using IPerf (LLD)

Test Setup Information	TCP Window Size in KBytes	TCP/IP Packets		
		Bandwidth in Mbits/sec	Transfer size in MBytes	Interval in Seconds
ARM Frequency = 297 MHz DDR Clock Frequency = 162 MHz	16	90.7	649	60
	32	91.2	653	60
	64	94.1	674	60
	128	94.2	674	60
	212	94.2	674	60
	212	94.2	674	60

Test Setup:

- See Section 2.3.2.3.1, CPMAC Ethernet Driver Performance: Test setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.3.2.3.3 CPMAC Ethernet Driver Performance: Server Preemption Mode

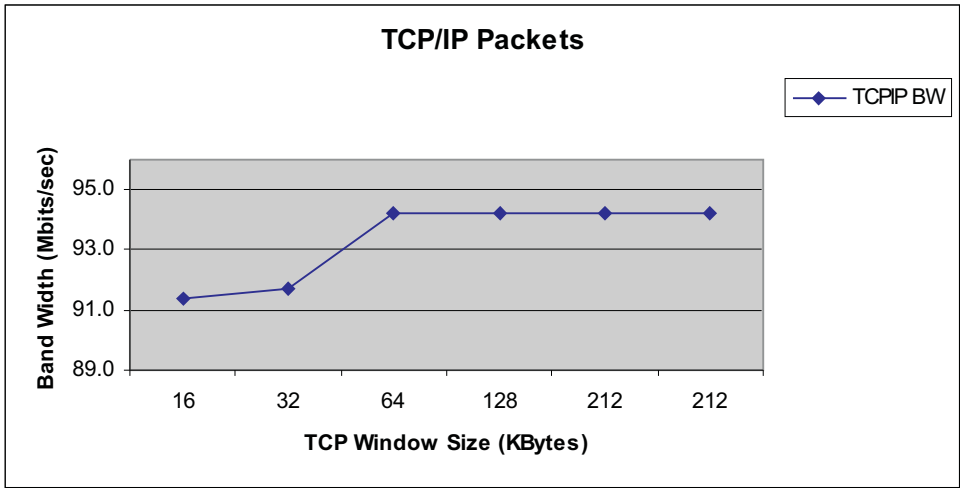

Figure 2-15. CPMAC Ethernet Driver Runtime Performance Characteristics (Server)

Table 2-52. CPMAC Ethernet Driver Performance Using IPerf (Server)

Test Setup Information	TCP Window Size in KBytes	TCP/IP Packets		
		Bandwidth in Mbits/sec	Transfer size in MBytes	Interval in Seconds
ARM Frequency = 297 MHz DDR Clock Frequency = 162 MHz	16	91.4	653	60
	32	91.7	654	60
	64	94.2	674	60
	128	94.2	674	60
	212	94.2	674	60
	212	94.2	674	60

Test Setup:

- See Section 2.3.2.3.1, CPMAC Ethernet Driver Performance: Test setup.
- Rebuild the Linux Kernel for Server preemption mode.

2.3.2.3.4 CPMAC Ethernet Driver Performance: Real-Time Preemption Mode

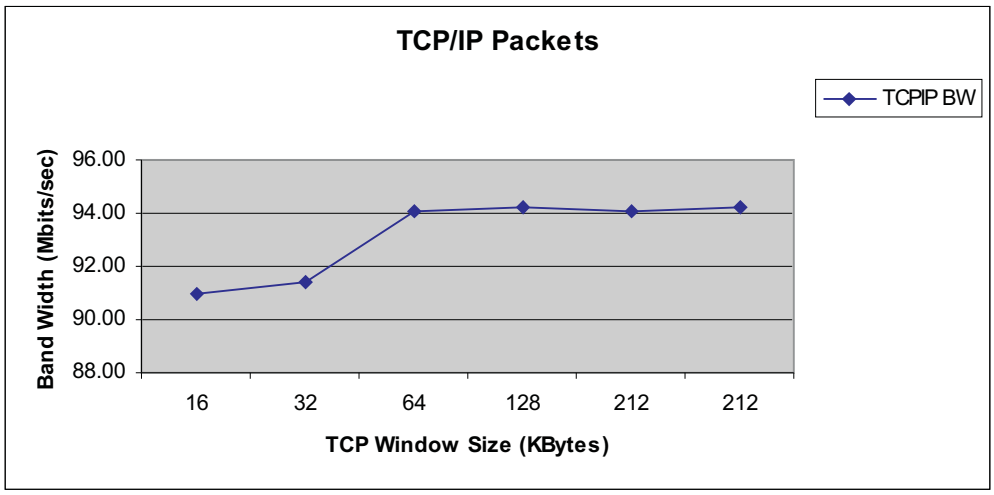

Figure 2-16. CPMAC Ethernet Driver Runtime Performance Characteristics (Real-Time)

Table 2-53. CPMAC Ethernet Driver Performance Using IPerf (Real-Time)

Test Setup Information	TCP Window Size in KBytes	TCP/IP Packets		
		Bandwidth in Mbits/sec	Transfer size in MBytes	Interval in Seconds
ARM Frequency = 297 MHz DDR Clock Frequency = 162 MHz	16	91	651	60
	32	91.4	644	60
	64	94.1	654	60
	128	94.2	674	60
	212	94.1	674	60
	212	94.2	674	60

Test Setup:

- See Section 2.3.2.3.1 CPMAC Ethernet Driver Performance: Test setup.
- Rebuild the Linux Kernel for Real-Time preemption mode.

2.4 USB Drivers

The USB peripheral provides support for USB 2.0 including HID, HDD, CD and mass storage classes. Figure 2-17 shows the driver architecture. The USB driver is implemented as a communication driver. EDMA is used for both host and device operations. USB host and slave drivers are described in the following sections.

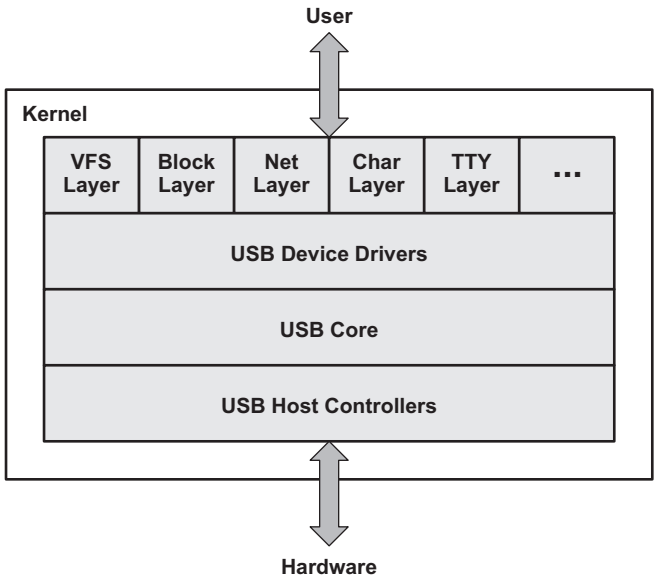

Figure 2-17. Linux Kernel USB Driver Architecture

2.4.1 Driver Features

The USB Driver supports the following features:

- Supports data transfer in both PIO and DMA mode for all kernel preemption modes
- Supports host and slave operations
- Supports class driver for HID and Mass storage.
- Supports USB ISO Class
- Supports partial USB OTG for HNP and SRP. USB OTG functionality has been tested on the DM355 devices by connecting two EVMs back to back using a mini-AB USB cable.
- OTG functionality is currently operated/invoked by providing inputs to the proc entry.

List of commands:

- i – Initiate Role Identification, Start/Request session.
- e – End the Session on VBUS
- s – Suspend USB Bus

Operation Sequence:

1. Start/Request Session

   Initiate role identification by passing the following command

   ```
   echo "i" > /proc/driver/otg
   ```

   This will do role identification and do SRP/provide session on the Bus.

2. Start HNP

   From A-Device, pass the following command to suspend Bus

   ```
   echo "s" > /proc/driver/otg
   ```

   This will result in HNP role switching.

To relinquish Host role from B-Device, pass the following command:

```
echo "e" > /proc/driver/otg
```

Also, when OTG mode is selected, be sure that the bus is turned on by performing step 1 (Start/Request Session). If this is not done, the connected device will not be enumerated.

The OTG specification requires that the driver (USB Linux Kernel stack plus MUSB Controller Driver) communicate several events to the Application Layer. MV 2.6.10 kernel lacks the infrastructure to communicate OTG events to the Application Layer.

The Pro v4.0 MV 2.6.10 kernel lacks an interface for the Application Layer to request USB bus access. This is required in OTG mode to initiate a Device/Host mode from the Application Layer and to perform role switching.

The MUSB Controller driver uses a /proc interface to work around the kernel limitation, to expose an interface to the user for requesting USB bus access and to perform role switching. The MUSB driver is tested for OTG functionalities using this interface on the DaVinci platform.

The above mentioned points are the limitations from the USB stack perspective in supporting full-fledged OTG functionality on the Davinci platform.

2.4.2 Support and Constraints

Features Not Supported	• NTFS file system is not supported. • USB isochronous, printer, and wireless classes are not supported.
Constraints	Implementation provisions the option of reserving an end point (EP - Rx, Tx) for handling ISO devices.
Supported System Calls	open(), close(), read(), write() and ioctl()
Supported IOCTLs	None

2.4.3 Performance and Benchmarks

2.4.3.1 DM355 EVM

Table 2-54 shows the memory requirements for the USB Driver modules built for DM355.

Table 2-54. USB Driver Memory Statistics - DM355

Configuration	Memory Statistics (in Bytes)			
	Program Memory	Data Memory		Total
		Initialized	Uninitialized	
usbcore.ko	78516	2800	1116	82432
musb_hdrc.ko	27116	628	0	27744
usb-storage.ko	24324	4700	0	29024
usbhid.ko	20928	1428	0	22356
g_ether.ko	21596	1080	136	22812
g_file_storage.ko	24616	800	148	25564
g_serial.ko	17442	868	332	18642
g_zero.ko	10416	680	112	11208
gadgetfs.ko	16348	1108	16	17472

2.4.3.1.1 USB Driver - Host Read/Write Performance: Test Setup

- **Hardware used:** DM355 Platform, Seagate FreeAgent Pro 500GB 3.5" Hard Drive – 7200 (Mfg Part #: ST305004FPA1E3-RK)
- **Root Filesystem used:** YAFFS2 (/dev/mtdblock3)
- **Test Filesystem used on USB Hard Disk:** VFAT

- **Mode tested:** Low Latency Desktop, Server and Real-Time preemption modes
- **Test Commands:**
  - test_perf_write.sh

```
#!/bin/sh
if [ $# -ne 3 ]; then
       echo "Usage: $0 <mount dir> <buffer-size> <count>" 1>&2
exit 127
fi
dd if=/dev/zero of=$1/test.out bs=$2 count=$3
umount $1
NOTE:  This is to flush file system cache
```

  - test_perf_read.sh

```
#!/bin/sh
if [ $# -ne 3 ]; then
       echo "Usage: $0 <mount dir> <buffer-size> <count>" 1>&2
exit 127
fi
dd if=$1/test.out of=/dev/null bs=$2 count=$3
umount $1
```

- **Test sequence:**
  1. mkdir /mnt/usb
  2. mount -t vfat /dev/sda1 /mnt/usb
  3. time ./test_perf_write.sh /mnt/usb <buffer-size> <count>
  4. mount –t vfat /dev/sda1 /mnt/usb
  5. time ./test_perf_read.sh /mnt/usb <buffer-size> <count>
  6. mount –t vfat /dev/sda1 /mnt/usb
  7. rm –f /mnt/usb/test.out
  8. umount /mnt/usb
  9. Repeat the steps 2 – 8, for each of the different buffer sizes – 4K, 64K, 128K, 256K, 512K and 1024K
  10. For each iteration, the count is also calculated for a constant file size (count * buffer-size) of 256M

2.4.3.1.2 USB Driver - Host Write Performance: Low Latency Desktop Mode

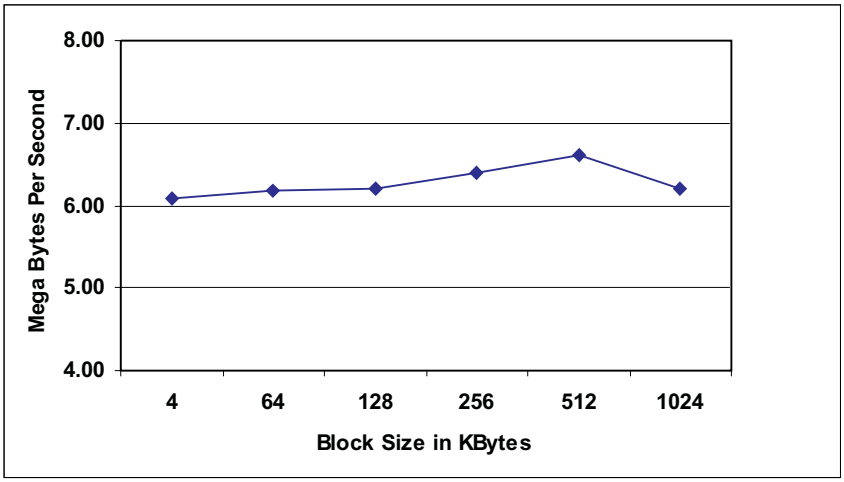

Figure 2-18. USB Driver - Host Write Performance Results (LLD) - DM355

Table 2-55. USB Driver - Host Write Performance Data (LLD) - DM355

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 216 MHZ	4	256	42.02	6.09
	64	256	41.37	6.19
	128	256	41.23	6.21
	256	256	40.05	6.39
	512	256	38.75	6.61
	1024	256	41.28	6.20

Test Setup:

- See Section 2.4.3.1.1, USB Driver Host Read/Writer Performance: Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.4.3.1.3 USB Driver - Host Read Performance: Low Latency Desktop Mode

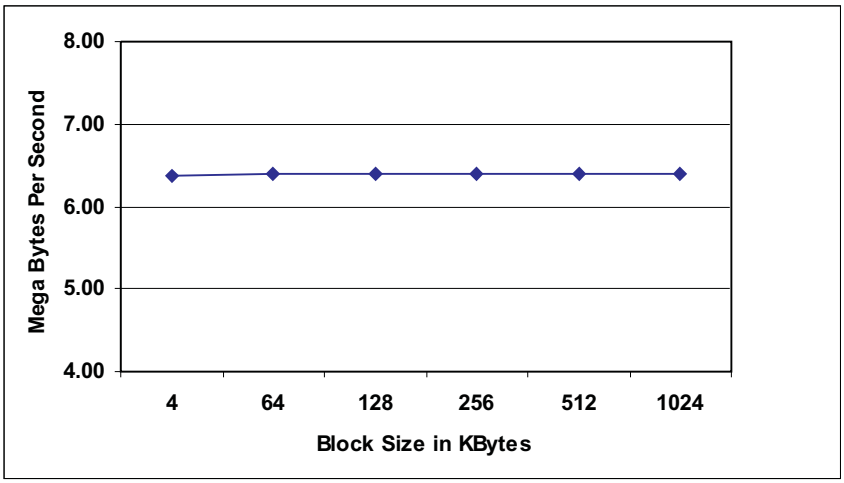

Figure 2-19. USB Driver - Host Read Performance Results (LLD) - DM355

Table 2-56. USB Driver - Host Read Performance Data (LLD) - DM355

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 216 MHZ	4	256	40.13	6.38
	64	256	40.00	6.40
	128	256	40.04	6.39
	256	256	40.02	6.40
	512	256	40.06	6.39
	1024	256	40.07	6.39

Test Setup:

- See Section 2.4.3.1.1, USB Driver Host Read/Writer Performance: Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.4.3.1.4 USB Driver - Host Write Performance: Server Mode

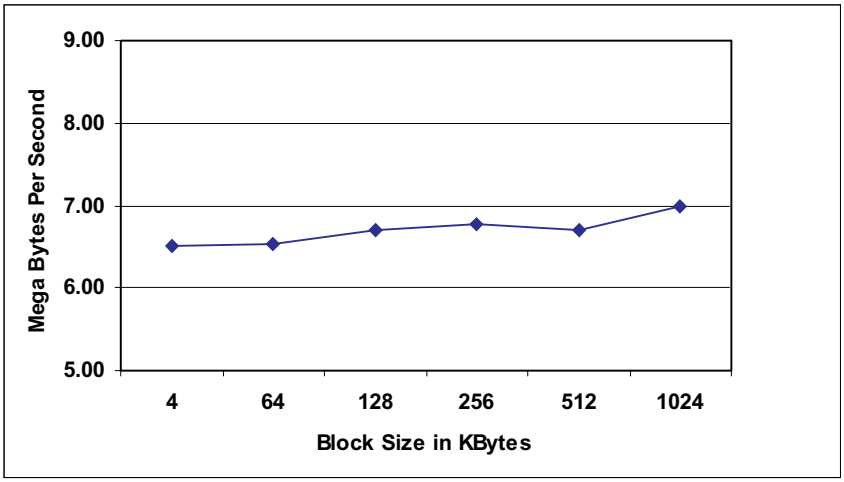

Figure 2-20. USB Driver - Host Write Performance: Server Mode

Table 2-57. USB Driver - Host Write Performance Data (Server) - DM355

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 216 MHZ	4	256	39.33	6.51
	64	256	39.25	6.52
	128	256	38.25	6.69
	256	256	37.86	6.76
	512	256	38.18	6.71
	1024	256	36.60	6.99

Test Setup:

- See Section 2.4.3.1.1, USB Driver Host Read/Writer Performance: Test Setup.
- Rebuild the Linux Kernel for Server preemption mode.

2.4.3.1.5 USB Driver - Host Read Performance: Server Mode

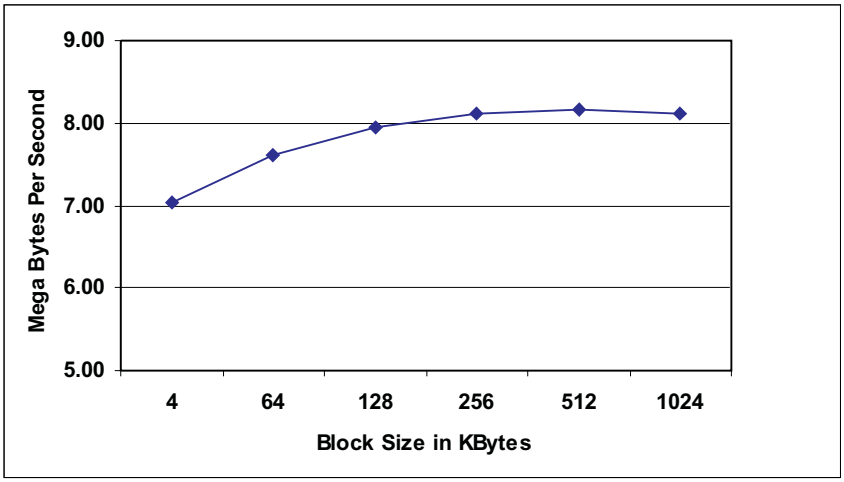

Figure 2-21. USB Driver - Host Read Performance Results (Server) — DM355

Table 2-58. USB Driver - Host Read Performance Data (Server) - DM355

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 216 MHZ	4	256	36.44	7.03
	64	256	33.60	7.62
	128	256	32.20	7.95
	256	256	31.53	8.12
	512	256	31.40	8.15
	1024	256	31.52	8.12

Test Setup:

- See Section 2.4.3.1.1, USB Driver Host Read/Writer Performance: Test Setup.
- Rebuild the Linux Kernel for Server preemption mode.

2.4.3.1.6 USB Driver - Host Write Performance: Real-Time Preemption Mode

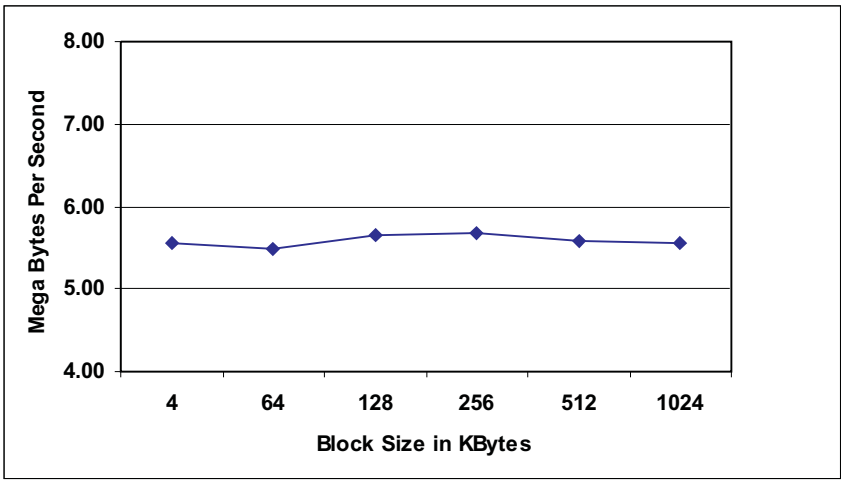

Figure 2-22. USB Driver - Host Write Performance Results (Real-Time) — DM355

Table 2-59. USB Driver - Host Write Performance Data (Real-Time) - DM355

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 216 MHZ	4	256	46.02	5.56
	64	256	46.72	5.48
	128	256	45.31	5.65
	256	256	45.11	5.68
	512	256	45.83	5.59
	1024	256	46.09	5.55

Test Setup:

- See Section 2.4.3.1.1, USB Driver Host Read/Writer Performance: Test Setup.
- Rebuild the Linux Kernel for Real-Time preemption mode.

2.4.3.1.7 USB Driver - Host Read Performance: Real-Time Preemption Mode

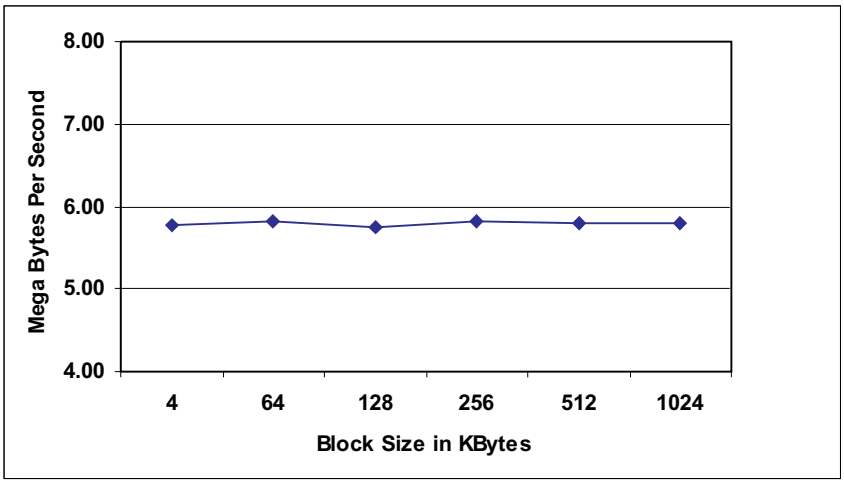

Figure 2-23. USB Driver - Host Read Performance Results (Real-Time) - DM355

Table 2-60. USB Driver - Host Read Performance Data (Real-Time) - DM355

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 216 MHZ	4	256	44.30	5.78
	64	256	44.04	5.81
	128	256	44.47	5.76
	256	256	43.98	5.82
	512	256	44.11	5.80
	1024	256	44.20	5.79

Test Setup:

- See Section 2.4.3.1.1, USB Driver Host Read/Writer Performance: Test Setup.
- Rebuild the Linux Kernel for Real-Time preemption mode.

2.4.3.1.8 USB Driver - RNDIS Performance: Test Setup

- **Hardware used:** DM355 EVM, Windows XP/SP2 PC, USB 2.0 Cable
- **Root Filesystem used:** YAFFS2 (/dev/mtdblock3)
- **Mode tested:** Low Latency Desktop, Server and Real-Time preemption modes
- **Test Commands:**
  - Server side command: "iperf -s"
  - Client side command : "iperf -c <server ip> -w<window size> -t60"
  - iperf version 1.70 is used
- **Test sequence:**
  1. Connect the Windows XP PC to the EVM using a USB 2.0 cable
  2. Configure each interface with an appropriate IP Address
  3. Start “iperf”, using the server side command, on the EVM
  4. Start “iperf”, using the client side command, on the Windows PC
  5. Run this test 5 times, each time changing the window size to a different size (16, 32, 64, 128, and 212)
  6. The Bandwidth data is collected from the “iperf”, in the server mode, running on the EVM

2.4.3.1.9 USB Driver - RNDIS Performance: Low Latency Desktop Mode

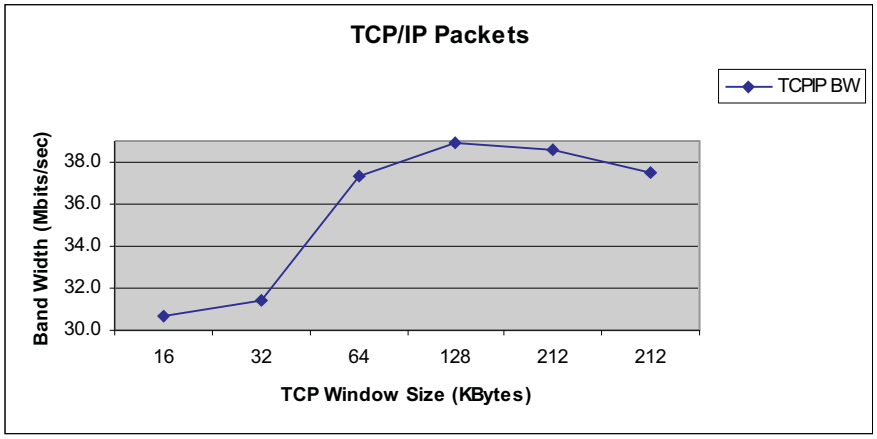

Figure 2-24. USB Driver - RNDIS Performance Results (LLD) - DM355

Table 2-61. USB Driver - RNDIS Performance Data Using IPerf (LLD) - DM355

Test Setup Information	TCP Window Size in Kbytes	TCP/IP Packets		
		Bandwidth in Mbits/sec	Transfer size in Mbytes	Interval in sec.
ARM Frequency = 216 MHz DDR Clock = 171 MHz Mode = DMA	16	30.7	219	60
	32	31.4	225	60
	64	37.3	267	60
	128	38.9	278	60
	212	38.6	278	60
	212	37.5	268	60

Test Setup:

- See Section 2.4.3.1.8, USB Driver RNDIS Performance: Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.4.3.1.10 USB Driver - RNDIS Performance: Server Mode

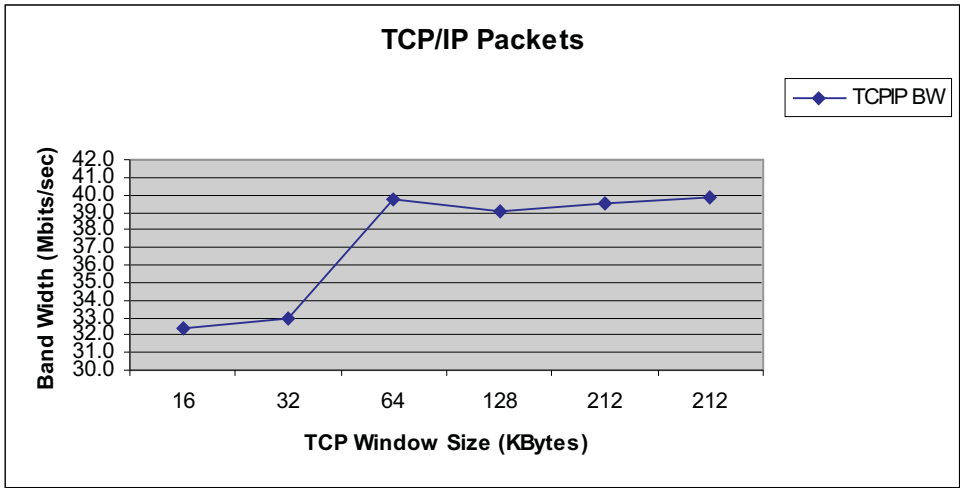

Figure 2-25. USB RNDIS Performance Results (Server) - DM355

Table 2-62. USB RNDIS Performance Data Using IPerf (Server) - DM355

Test Setup Information	TCP Window Size in Kbytes	TCP/IP Packets		
		Bandwidth in Mbits/sec	Transfer size in Mbytes	Interval in sec.
ARM Frequency = 216 MHz DDR Clock = 171 MHz Mode = DMA	16	32.4	232	60
	32	33	236	60
	64	39.7	284	60
	128	39.1	280	60
	212	39.5	203	60
	212	39.8	285	60

Test Setup:

- See Section 2.4.3.1.8, USB Driver RNDIS Performance: Test Setup.
- Rebuild the linux kernel for server preemption mode.

2.4.3.1.11 USB Driver - RNDIS Performance: Real-Time Preemption Mode

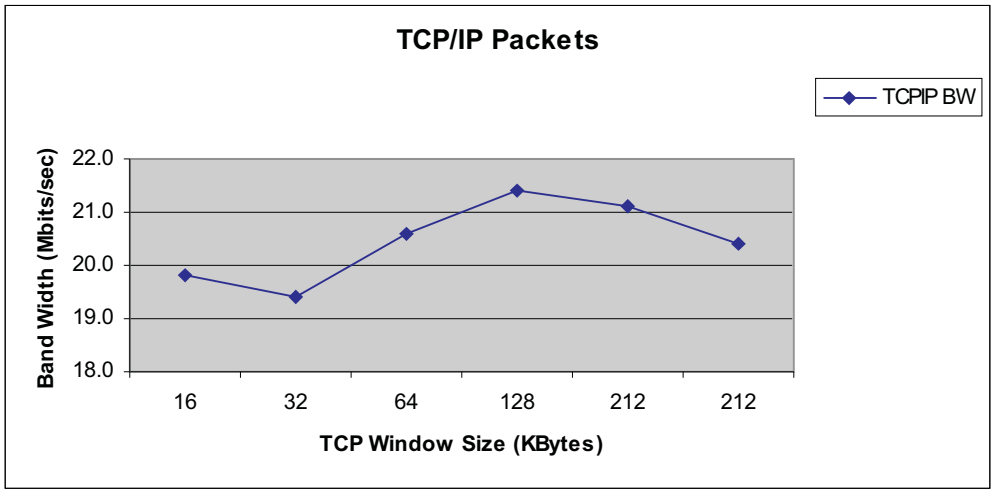

Figure 2-26. USB Driver - RNDIS Performance Results (Real-Time) - DM355

Table 2-63. USB Driver - RNDIS Performance Data Using IPerf (Real-Time) - DM355

Test Setup Information	TCP Window Size in Kbytes	TCP/IP Packets		
		Bandwidth in Mbits/sec	Transfer size in Mbytes	Interval in sec.
ARM Frequency = 216 MHz DDR Clock = 171 MHz Mode = DMA	16	19.8	142	60
	32	19.4	139	60
	64	20.6	147	60
	128	21.4	153	60
	212	21.1	151	60
	212	20.4	151	60

Test Setup:

- See Section 2.4.3.1.8, USB Driver RNDIS Performance: Test Setup.
- Rebuild the Linux Kernel for Real-Time preemption mode.

2.4.3.2 DM6446 EVM

Table 2-64 shows the memory requirements for the USB Driver modules built for DM6446.

Table 2-64. USB Driver Memory Statistics - DM6446

Configuration	Memory Statistics (in Bytes)			
	Program Memory	Data Memory		Total
		Initialized	Uninitialized	
usbcore.ko	78428	2800	1116	82344
musb_hdrc.ko	27004	688	0	27692
usb-storage.ko	24248	4700	0	28948
usbhid.ko	20932	1428	0	22360
g_ether.ko	21792	1080	136	23008
g_file_storage.ko	24528	800	148	25476
g_serial.ko	17442	868	332	18642
g_zero.ko	10416	680	112	11208
gadgetfs.ko	16348	1108	16	17472

2.4.3.2.1 USB Driver - Host Read/Write Performance: Test Setup

- **Hardware used:** DM6446 Platform, Seagate FreeAgent Pro 500GB 3.5" Hard Drive – 7200 (Mfg Part #: ST305004FPA1E3-RK)
- **Root Filesystem used:** ATA – EXT3 (/dev/hda1)
- **Test Filesystem used on USB Hard Disk:** VFAT
- **Mode tested:** Low Latency Desktop, Server and Real-Time preemption modes
- **Test Commands:**
  - test_perf_write.sh

```
#!/bin/sh
if [ $# -ne 3 ]; then
       echo "Usage: $0 <mount dir> <buffer-size> <count>" 1>&2
exit 127
fi
dd if=/dev/zero of=$1/test.out bs=$2 count=$3
umount $1
NOTE:  This is to flush file system cache
```

  - test_perf_read.sh

```
#!/bin/sh
if [ $# -ne 3 ]; then
       echo "Usage: $0 <mount dir> <buffer-size> <count>" 1>&2
exit 127
fi
dd if=$1/test.out of=/dev/null bs=$2 count=$3
umount $1
```

- **Test sequence:**
  1. mkdir /mnt/usb
  2. mount -t vfat /dev/sda1 /mnt/usb
  3. time ./test_perf_write.sh /mnt/usb <buffer-size> <count>
  4. mount –t vfat /dev/sda1 /mnt/usb
  5. time ./test_perf_read.sh /mnt/usb <buffer-size> <count>
  6. mount –t vfat /dev/sda1 /mnt/usb
  7. rm –f /mnt/usb/test.out
  8. umount /mnt/usb
  9. Repeat the steps 2 – 8, for each of the different buffer sizes – 4K, 64K, 128K, 256K, 512K and 1024K

10. For each iteration, the count is also calculated for a constant file size (count * buffer-size) of 256M

2.4.3.2.2 USB Driver - Host Write Performance: Low Latency Desktop Mode

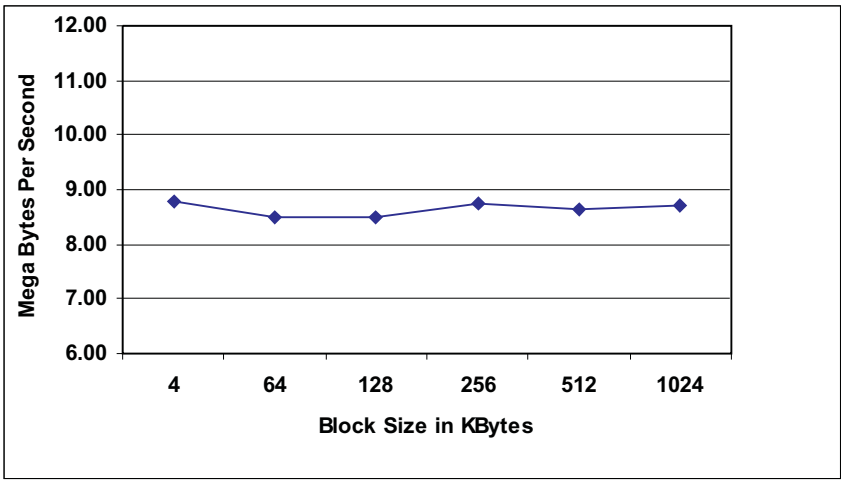

Figure 2-27. USB Driver - Host Write Performance Results (LLD) - DM6446

Table 2-65. USB Driver - Host Write Performance Data (LLD) - DM6446

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ DM6446	4	256	29.10	8.80
	64	256	30.19	8.48
	128	256	30.10	8.50
	256	256	29.26	8.75
	512	256	29.64	8.64
	1024	256	29.40	8.71

Test Setup:

- See Section 2.4.3.2.1, USB Driver Host Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.4.3.2.3 USB Driver - Host Read Performance: Low Latency Desktop Mode

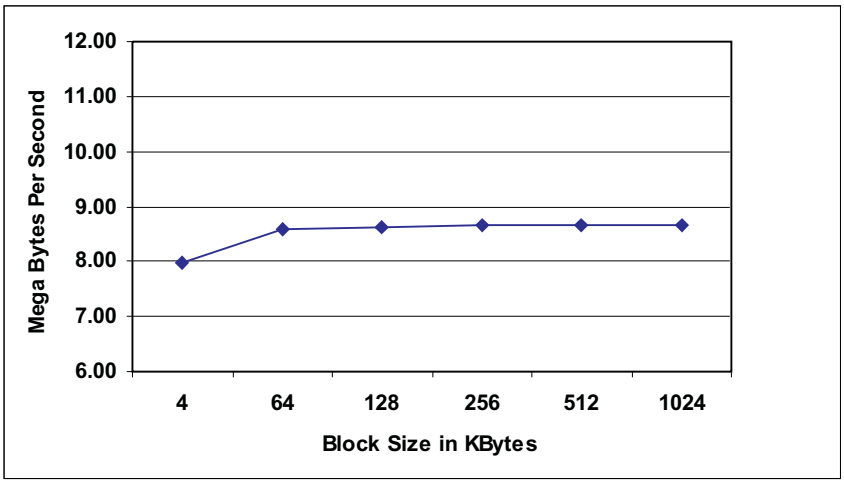

Figure 2-28. USB Driver - Host Read Performance Results (LLD) - DM6446

Table 2-66. USB Driver - Host Read Performance Data (LLD) - DM6446

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ DM6446	4	256	32.03	7.99
	64	256	29.80	8.59
	128	256	29.70	8.62
	256	256	29.59	8.65
	512	256	29.59	8.65
	1024	256	29.56	8.66

Test Setup:

- See Section 2.4.3.2.1, USB Driver Host Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.4.3.2.4 USB Driver - Host Write Performance: Server Mode

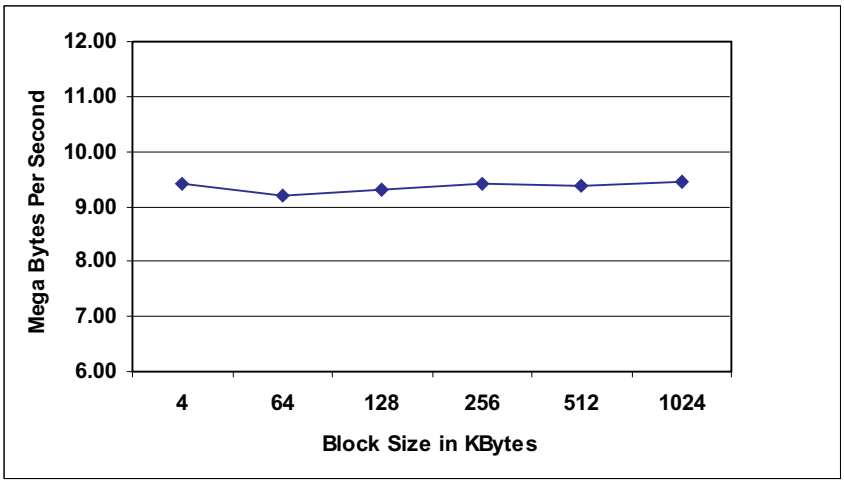

Figure 2-29. USB Driver - Host Write Performance Results (Server) - DM6446

Table 2-67. USB Driver - Host Write Performance Data (Server) - DM6446

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ DM6446	4	256	27.15	9.43
	64	256	27.86	9.19
	128	256	27.52	9.30
	256	256	27.15	9.43
	512	256	27.32	9.37
	1024	256	27.12	9.44

Test Setup:

- See Section 2.4.3.2.1, USB Driver Host Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Server preemption mode.

2.4.3.2.5 USB Driver - Host Read Performance: Server Mode

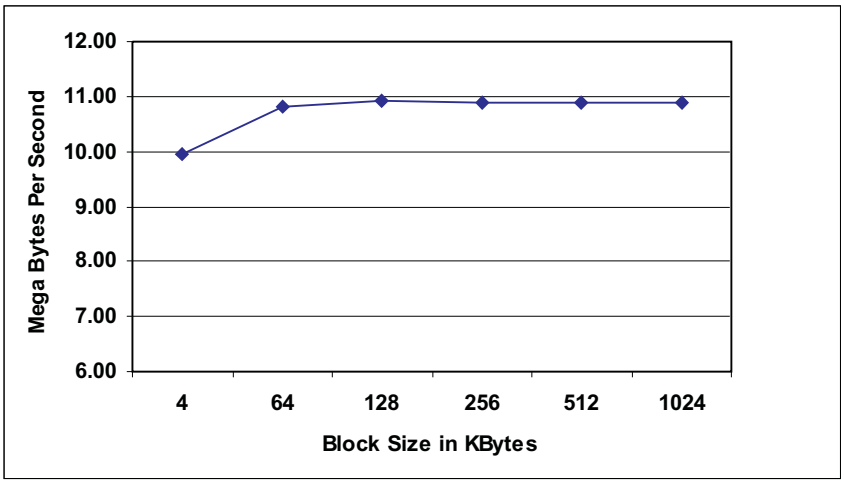

Figure 2-30. USB Driver - Host Read Performance Results (Server) - DM6446

Table 2-68. USB Driver - Host Read Performance Data (Server) - DM6446

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ dm6446	4	256	25.73	9.95
	64	256	23.65	10.82
	128	256	23.47	10.91
	256	256	23.50	10.89
	512	256	23.53	10.88
	1024	256	23.48	10.90

Test Setup:

- See Section 2.4.3.2.1, USB Driver Host Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Server preemption mode.

2.4.3.2.6 USB Driver - Host Write Performance: Real-Time Preemption Mode

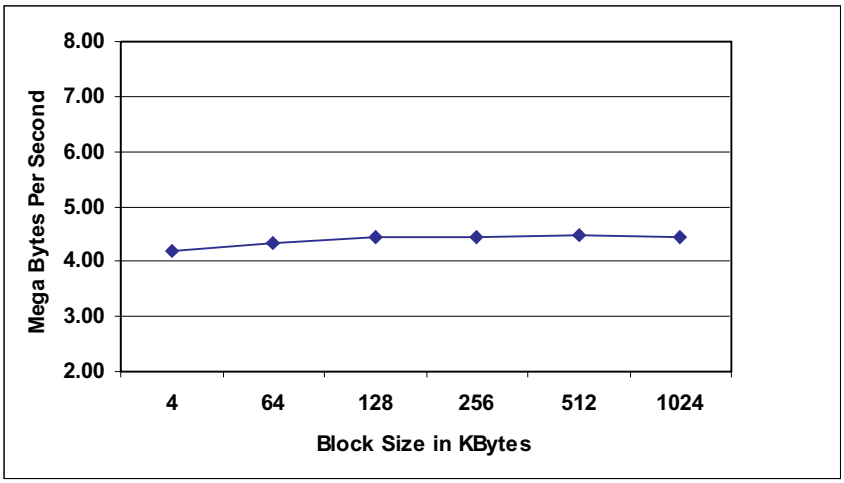

Figure 2-31. USB Driver - Host Write Performance Results (Real-Time) - DM6446

Table 2-69. USB Driver - Host Write Performance Data (Real-Time) - DM6446

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ DM6446	4	256	61.21	4.18
	64	256	58.88	4.35
	128	256	57.42	4.46
	256	256	57.53	4.45
	512	256	57.22	4.47
	1024	256	57.67	4.44

Test Setup:

- See Section 2.4.3.2.1, USB Driver Host Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for real-time preemption mode.

2.4.3.2.7 USB Driver - Host Read Performance: Real-Time Preemption Mode

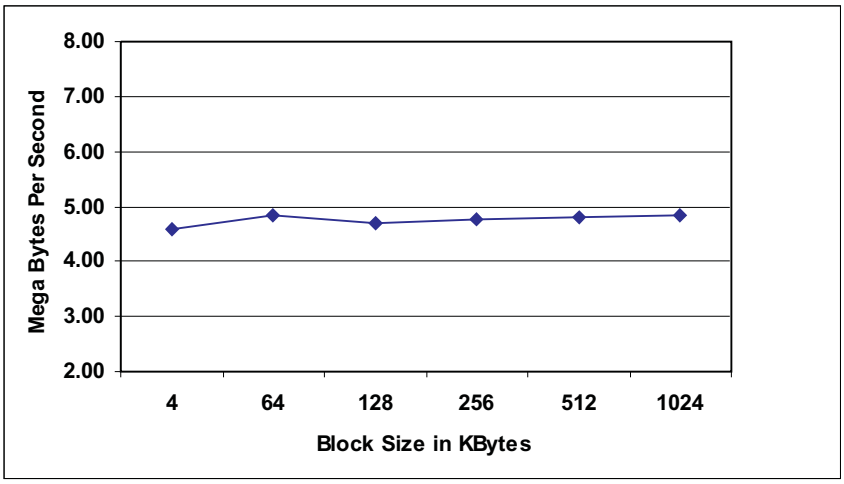

Figure 2-32. USB Driver - Host Read Performance Results (Real-Time) - DM6446

Table 2-70. USB Driver - Host Read Performance Data (Real-Time) - DM6446

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ DM646	4	256	55.66	4.60
	64	256	53.04	4.83
	128	256	54.71	4.68
	256	256	53.66	4.77
	512	256	53.35	4.80
	1024	256	53.02	4.83

Test Setup:

- See Section 2.4.3.2.1, USB Driver Host Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Real-Time preemption mode.

2.4.3.2.8 USB Driver - RNDIS Performance - Test Setup

- **Hardware used:** DM6446 EVM, Windows XP/SP2 PC, USB 2.0 Cable
- **Root Filesystem used:** ATA – EXT3 (/dev/hda1)
- **Mode tested:** Low Latency Desktop, Server and Real-Time preemption modes
- **Test Commands:**
  - Server side command: "iperf -s"
  - Client side command : "iperf -c <server ip> -w<window size> -t60"
  - iperf version 1.70 is used
- **Test sequence:**
  1. Connect the Windows XP PC to the EVM using a USB 2.0 cable
  2. Configure each interface with an appropriate IP Address
  3. Start “iperf”, using the server side command, on the EVM
  4. Start “iperf”, using the client side command, on the Windows PC
  5. Run this test 5 times, each time changing the window size to a different size (16, 32, 64, 128, and 212)
  6. The Bandwidth data is collected from the “iperf”, in the server mode, running on the EVM

2.4.3.2.9 USB Driver - RNDIS Performance: Low Latency Desktop Mode

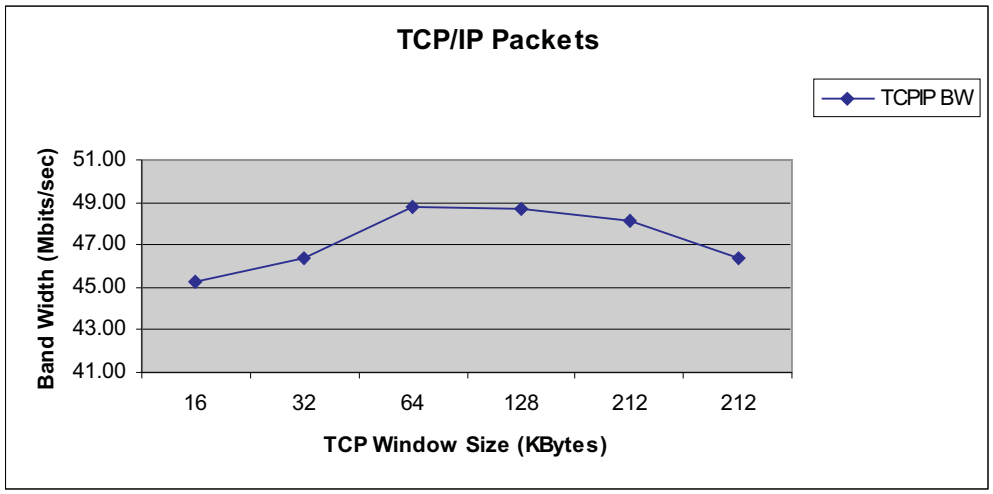

Figure 2-33. USB Driver - RNDIS Performance Results (LLD) - DM6446

Table 2-71. USB Driver - RNDIS Performance Data Using IPerf (LLD) - DM6446

Runtime Performance Characteristics - Low Latency Mode				
Test Setup Information	TCP Window Size in KBytes	TCP/IP Packets		
		Bandwidth in Mbits/sec	Transfer Size in MBytes	Interval in sec.
ARM Frequency = 297 MHz DDR Frequency = 162 MHz Mode = DMA	16	45.3	324	60
	32	46.4	332	60
	64	48.8	349	60
	128	48.7	349	60
	212	48.1	344	60
	212	46.4	332	60

Test Setup:

- See Section 2.4.3.2.8, USB Driver RNDIS Performance: Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.4.3.2.10 USB Driver - RNDIS Performance: Server Mode

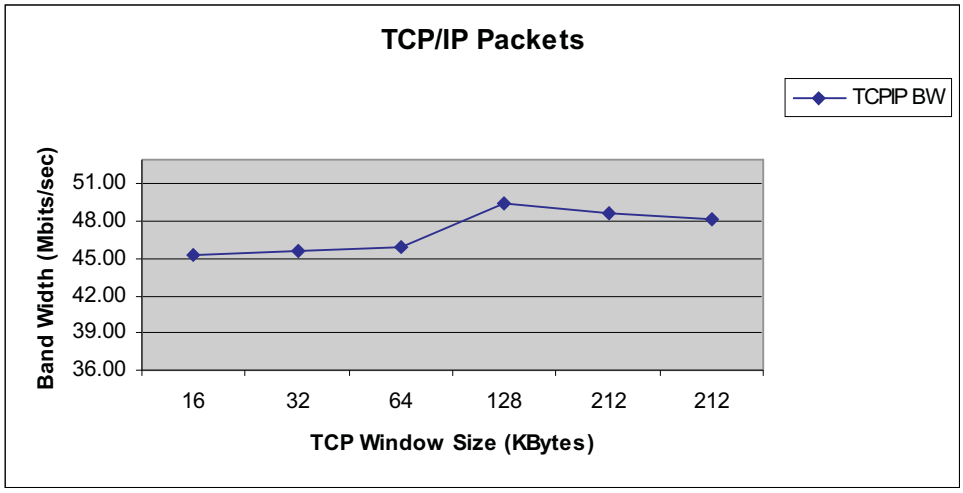

Figure 2-34. USB Driver - RNDIS Performance Results (Server) - DM6446

Table 2-72. USB Driver - RNDIS Performance Data Using IPerf (Server) - DM6446

Test Setup Information	TCP Window Size in KBytes	TCP/IP Packets		
		Bandwidth in Mbits/sec	Transfer size in MBytes	Interval in sec.
ARM Frequency = 297 MHz DDR Frequency = 162 MHz Mode = DMA	16	45.3	324	60
	32	45.6	327	60
	64	46	329	60
	128	49.4	353	60
	212	48.7	349	60
	212	48.2	345	60

Test Setup:

- See Section 2.4.3.2.8, USB Driver RNDIS Performance: Test Setup.
- Rebuild the Linux Kernel for Server preemption mode.

2.4.3.2.11 USB Driver - RNDIS Performance: Real-Time Preemption Mode

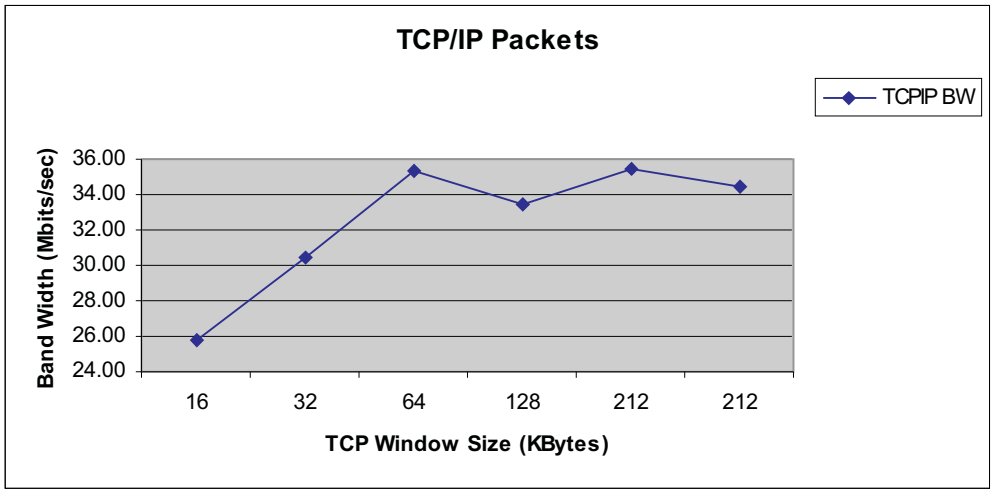

Figure 2-35. USB Driver - RNDIS Performance Results (Real-Time) - DM6446

Table 2-73. USB Driver - RNDIS Performance Data Using IPerf (Real-Time) - DM6446

Test Setup Information	TCP window size in KBytes	TCP/IP Packets Bandwidth in Mbits/sec	Transfer size in MBytes	Interval in sec.
ARM Frequency = 297 MHz DDR Frequency = 162 MHz Mode = DMA	16	25.8	185	60
	32	30.5	219	60
	64	35.3	253	60
	128	33.4	239	60
	212	35.5	254	60
	212	34.5	247	60

Test Setup:

- See Section 2.4.3.2.8, USB Driver RNDIS Performance: Test Setup.
- Rebuild the Linux Kernel for Real-Time preemption mode.

2.5 IDE ATA Driver

The ATA controller provides glueless interface to storage media. The IDE driver is implemented as a block driver (see Figure 2-36). It supports ATA standards, and creates the device nodes /dev/hda1, /dev/hda2, etc. for user space access.

NOTE
The device nodes /dev/hda1 and /dev/hda2 are created because the disk has partitions. If using a generic disk or a disk without partitions, then the generic dev node is /dev/hd[a-z].

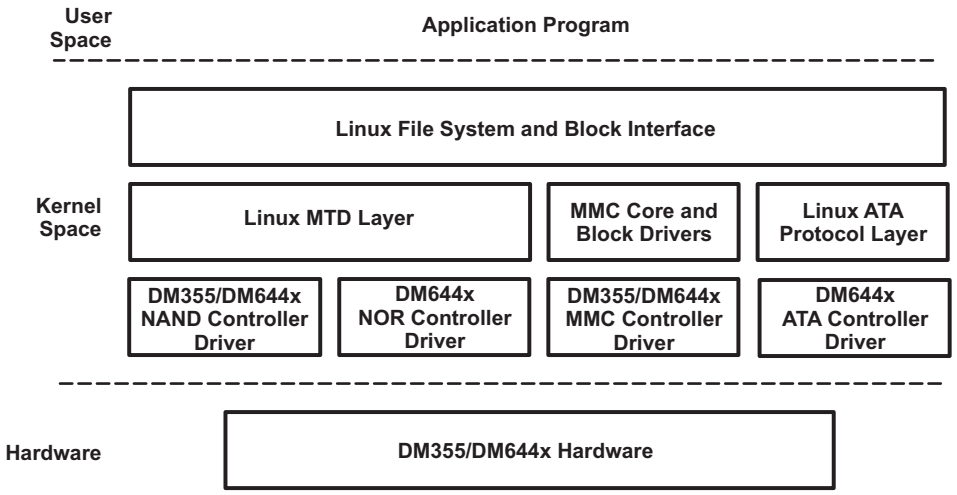

Figure 2-36. Linux Kernel IDE Driver Architecture

2.5.1 Driver Features

The IDE Driver is not applicable on DM355, since the DM355 EVM does not have IDE ATA support.

The IDE Driver supports following features on the DM6446:

- The DM644x 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 unmounting file systems, and associated file system creation/manipulation, are left to the user
- IDE driver supports up to UDMA4 (UDMA0, UDMA1, UDMA2, UDMA3 and UDMA4)

2.5.2 Support and Constraints

Features Not Supported	NTFS file system is not supported
Constraints	Performance not consistent with theoretical values
Supported System Calls	open(), close(), read(), write(), ioctl()
Supported IOCTLs	See Table 2-74

Table 2-74. IDE Driver Supported IOCTLs

Constant	Description
HDIO_GETGEO	Get device geometry
HDIO_GET_IDENTITY	Get IDE identification information
HDIO_GET_NICE	Get nice flags
HDIO_DRIVE_TASKFILE	Execute raw task file

Table 2-74. IDE Driver Supported IOCTLs (continued)

Constant	Description
HDIO_DRIVE_CMD	Execute a special drive command
HDIO_DRIVE_TASK	Execute task and special drive command
HDIO_SCAN_HWIF	Register and (re)scan interface
HDIO_UNREGISTER_HWIF	Un-register interface
HDIO_SET_NICE	Set nice flags
HDIO_DRIVE_RESET	Execute a device reset
HDIO_GET_BUSSTATE	Get the bus state of the HWIF
HDIO_SET_BUSSTATE	Set the bus state of the HWIF

2.5.3 Performance and Benchmarks

2.5.3.1 DM6446 EVM

Table 2-75 shows the memory requirements for the IDE ATA Driver modules built for DM6446.

Table 2-75. IDE ATA Driver Memory Statistics - DM6446

Configuration	Memory Statistics (in Bytes)			
	Program Memory	Data Memory		Total
		Initialized	Uninitialized	
ide-core.ko	69420	1672	7164	78256
ide-disk.ko	15476	560	0	16036
ide-generic.ko	292	256	0	548
palm_bk3710.ko	4428	652	72	5152

2.5.3.1.1 IDE Driver - Read/Write Performance: Test Setup

- **Hardware used:** DM6446 Platform, TOSHIBA MK4032GAX Hard Disk (40 GB)
- **Root Filesystem used:** NFS
- **Test Filesystem used on USB Hard Disk:** EXT3
- **Mode tested:** Low Latency Desktop, Server and Real-Time preemption modes
- **Test Commands:**
  - test_perf_write.sh

```
#!/bin/sh
if [ $# -ne 3 ]; then
       echo "Usage: $0 <mount dir> <buffer-size> <count>" 1>&2
exit 127
fi
dd if=/dev/zero of=$1/test.out bs=$2 count=$3
umount $1
NOTE:  This is to flush file system cache
```

  - test_perf_read.sh

```
#!/bin/sh
if [ $# -ne 3 ]; then
       echo "Usage: $0 <mount dir> <buffer-size> <count>" 1>&2
exit 127
fi
dd if=$1/test.out of=/dev/null bs=$2 count=$3
umount $1
```

- **Test sequence:**
  1. mkdir /mnt/ata
  2. mount -t ext3 /dev/hda1 /mnt/ata -o data=ordered
  3. time ./test_perf_write.sh /mnt/ata <buffer-size> <count>
  4. mount -t ext3 /dev/hda1 /mnt/ata -o data=ordered

5. time ./test_perf_read.sh /mnt/ata <buffer-size> <count>
6. mount -t ext3 /dev/hda1 /mnt/ata -o data=ordered
7. rm –f /mnt/ata/test.out
8. umount /mnt/ata
9. Repeat the steps 2 – 8, for each of the different buffer sizes – 4K, 64K, 128K, 256K, 512K and 1024K
10. For each iteration, the count is also calculated for a constant file size (count * buffer-size) of 256M

2.5.3.1.2 IDE Driver - Write Performance: Low Latency Desktop Mode

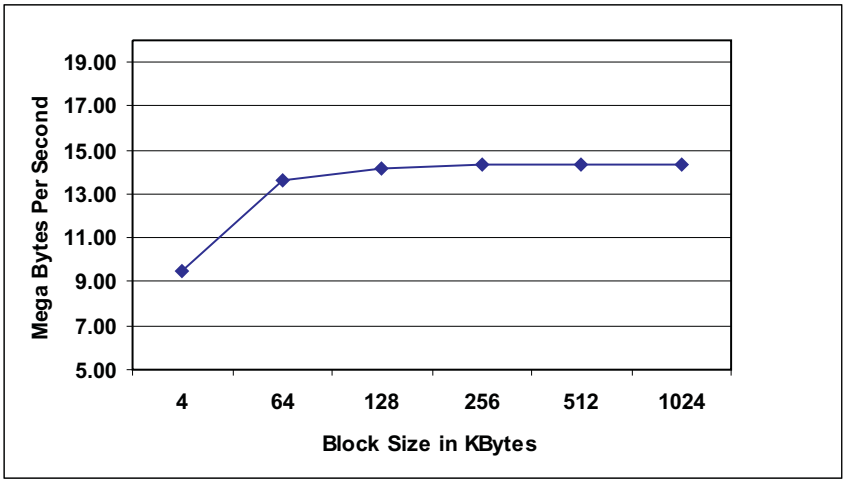

Figure 2-37. IDE Driver - Write Performance Results (LLD) - DM6446

Table 2-76. IDE Driver - Write Performance Data (LLD) - DM6446

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration Seconds	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	256	28.37	9.46
	64	256	19.65	13.66
	128	256	18.96	14.16
	256	256	18.71	14.35
	512	256	18.72	14.34
	1024	256	18.75	14.32

Test Setup:

- See Section 2.5.3.1.1, IDE Driver Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.5.3.1.3 IDE Driver - Read Performance: Low Latency Desktop Mode

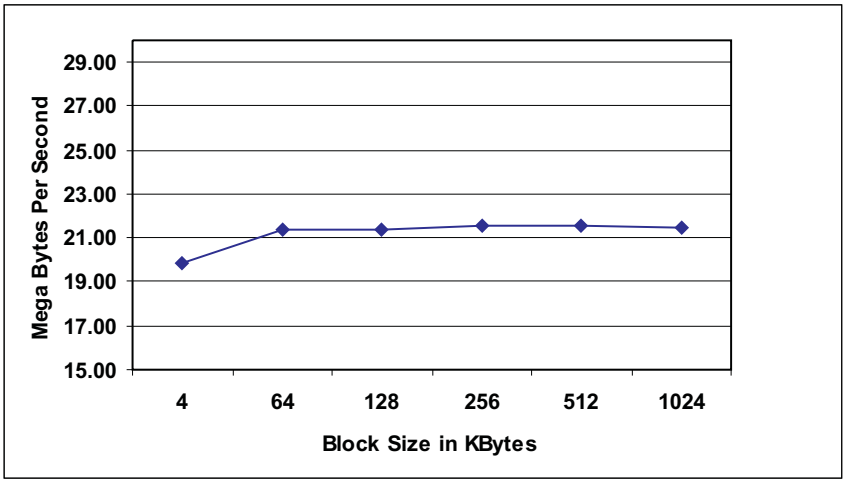

Figure 2-38. IDE Driver - Read Performance Results (LLD) - DM6446

Table 2-77. IDE Driver - Read Performance Data (LLD) - DM6446

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration Seconds	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	256	13.52	19.85
	64	256	12.55	21.39
	128	256	12.58	21.34
	256	256	12.47	21.53
	512	256	12.44	21.58
	1024	256	12.51	21.46

Test Setup:

- See Section 2.5.3.1.1, IDE Driver Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.5.3.1.4 IDE Driver - Write Performance: Server Mode

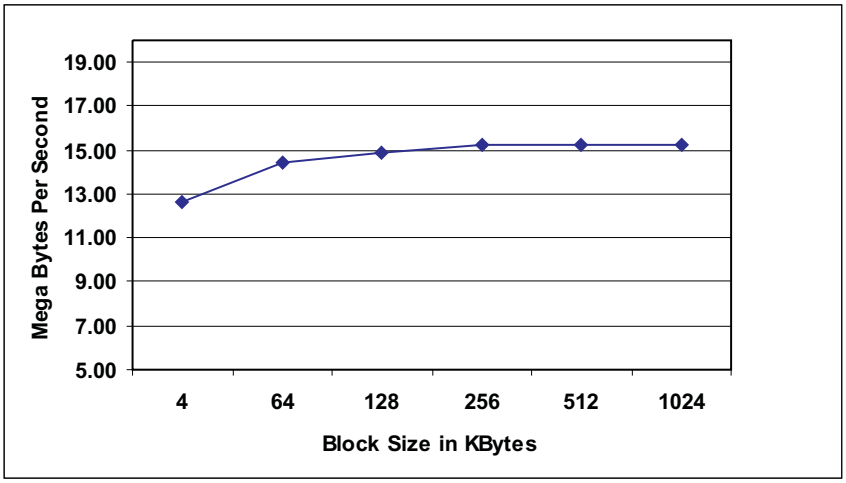

Figure 2-39. IDE Driver - Write Performance Results (Server) - DM6446

Table 2-78. IDE Driver - Write Performance Data (Server) - DM6446

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration Seconds	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	256	21.21	12.66
	64	256	18.55	14.47
	128	256	18.03	14.89
	256	256	17.66	15.20
	512	256	17.62	15.23
	1024	256	17.66	15.20

Test Setup:

- See Section 2.5.3.1.1, IDE Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Server preemption mode.

2.5.3.1.5 IDE Driver - Read Performance: Server Mode

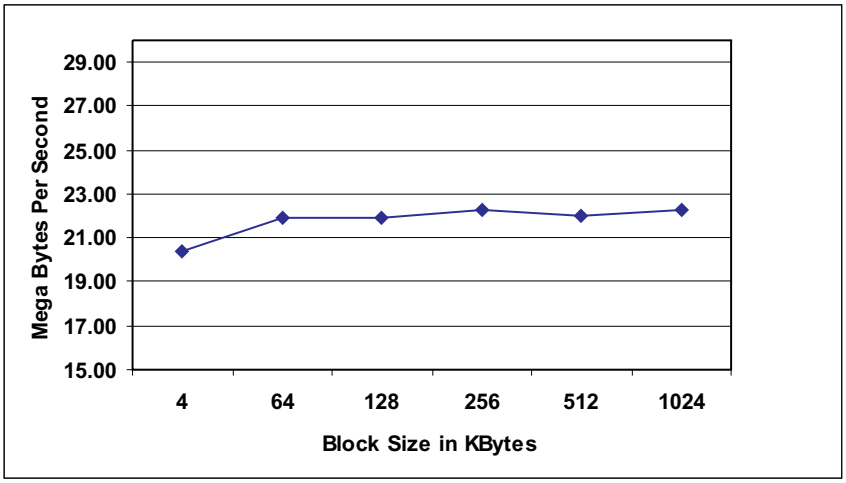

Figure 2-40. IDE Driver Read - Performance Results (Server) - DM6446

Table 2-79. IDE Driver - Read Performance Data (Server) - DM6446

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	256	13.15	20.41
	64	256	12.26	21.90
	128	256	12.23	21.95
	256	256	12.06	22.26
	512	256	12.18	22.04
	1024	256	12.03	22.31

Test Setup:

- See Section 2.5.3.1.1, IDE Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Server preemption mode.

2.5.3.1.6 IDE Driver - Write Performance: Real-Time Preemption Mode

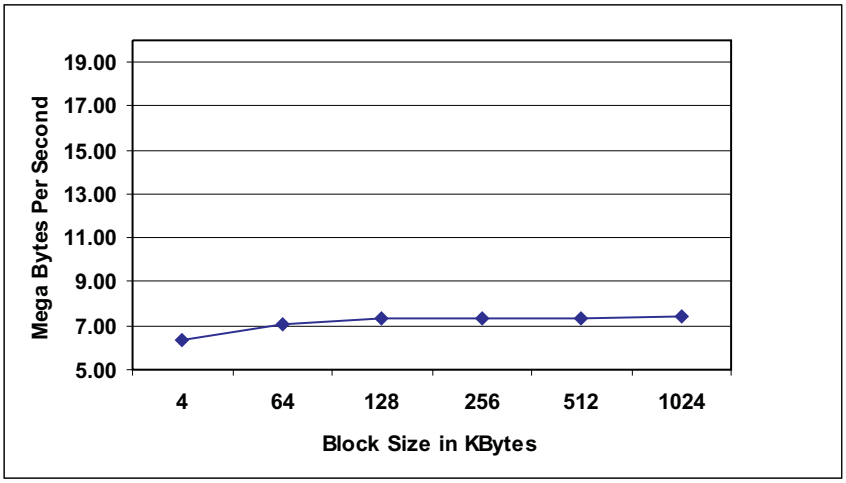

Figure 2-41. IDE Driver - Write Performance Results (Real-Time) - DM6446

Table 2-80. IDE Driver - Write Performance Data (Real-Time) - DM6446

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	256	42.01	6.39
	64	256	37.89	7.08
	128	256	36.69	7.32
	256	256	36.58	7.34
	512	256	36.40	7.37
	1024	256	36.22	7.41

Test Setup:

- See Section 2.5.3.1.1, IDE Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Real-Time preemption mode.

2.5.3.1.7 IDE Driver - Read Performance: Real-Time Preemption Mode

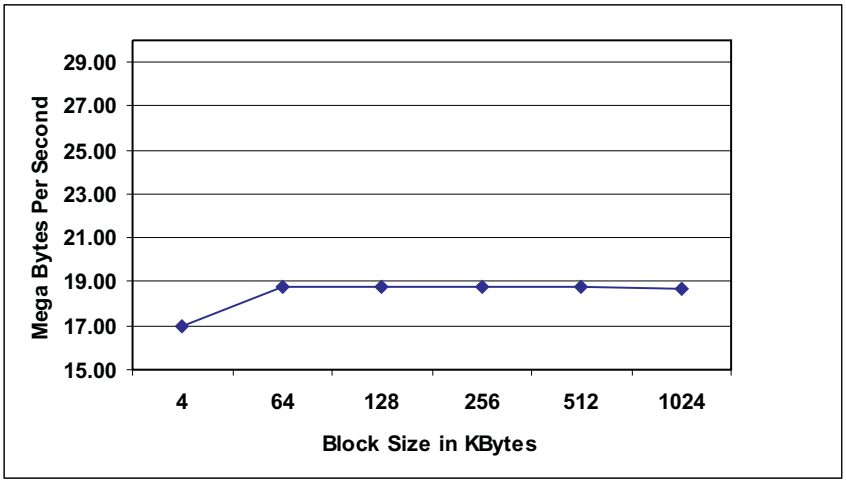

Figure 2-42. IDE Driver - Read Performance Results (Real-Time) - DM6446

Table 2-81. IDE Driver - Read Performance Data (Real-Time) - DM6446

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	256	15.80	16.99
	64	256	14.31	18.76
	128	256	14.31	18.76
	256	256	14.28	18.80
	512	256	14.30	18.77
	1024	256	14.35	18.71

Test Setup:

- See Section 2.5.3.1.1, IDE Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for real-time preemption mode.

2.6 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 2-43. It creates the device nodes for user space access (/dev/mmcblkp1, /dev/mmcblkp2, etc.).

NOTE

The device nodes /dev/mmcblkp1 and /dev/mmcblkp2 are created because the disk has partitions. If using a generic disk or a disk without partitions then the generic dev node is /dev/mmcblk[0,1,2…]

The DaVinci DM6446 and DM355 EVMs support MMC on the peripheral interface. There is one MMC/SD connector on the DM6446 baseboard, while there are two MMC/SD connectors on the DM355 baseboard.

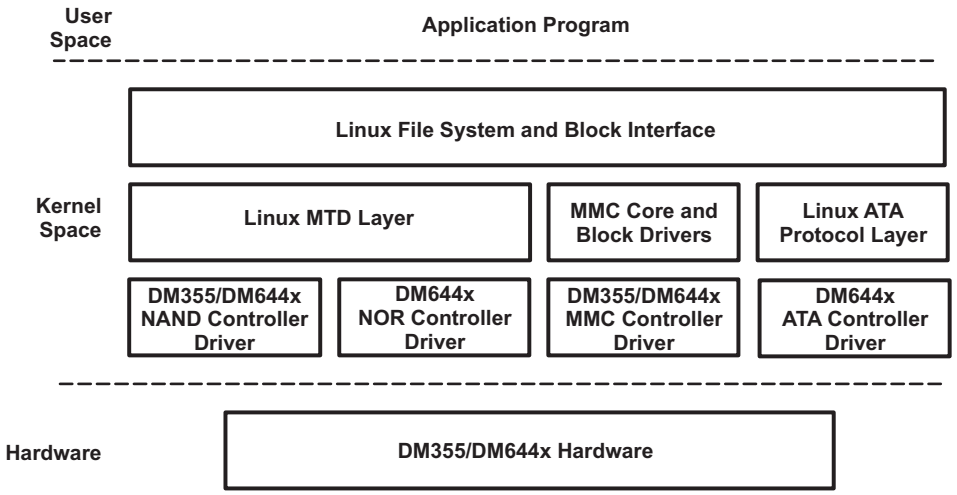

Figure 2-43. Linux Kernel MMC Driver Architecture

2.6.1 Driver Features

The DaVinci MMC Device Driver supports the following features:

- The DaVinci MMC Device driver for DM6446 and DM355 are accessible as block device drivers.
- 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.
- Supports DMA and PIO modes of operation.
- Supports high-capacity SD cards through custom SDIO stack.

2.6.2 Support and Constraints

Features not supported	None.
Constraints	On DM6446 EVM, some SD cards work at 50 MHz. Some SD cards fail with response CRC errors but can still be used by ignoring them. Some SD cards fail with DATA CRC errors for all data transfer commands and become unusable.

Supported System Calls	open(), close(), write() and read()
Supported IOCTLs	None

2.6.3 Performance and Benchmarks With Default MontaVista MMC/SD Stack

2.6.3.1 DM355 EVM

Table 2-82 shows the memory requirements for the MMC/SD Driver modules built for DM355.

Table 2-82. MMC Driver Memory Statistics - DM355

Configuration	Memory Statistics (in Bytes)			
	Program Memory	Data Memory		Total
		Initialized	Uninitialized	
davinci_mmc.ko	11676	372	0	12048
mmc_block.ko	5832	424	8	6264
mmc_core.ko	14376	880	16	15272

2.6.3.1.1 MMC/SD Driver - SD Read/Write Performance: Test Setup

- **Hardware used:** DM355 Platform, SanDisk 2GB Secure Digital (SD) memory card (Model: SDSDBR-2048-A10)
  - Performance can vary depending on the brand and type of SD card used
- **Root Filesystem used:** YAFFS2 (/dev/mtdblock3)
- **Test Filesystem used on the SD card :** VFAT
- **Mode tested:** Low Latency Desktop, Server and Real-Time preemption modes
- **Test Commands:**
  - test_perf_write.sh

```
#!/bin/sh
if [ $# -ne 3 ]; then
       echo "Usage: $0 <mount dir> <buffer-size> <count>" 1>&2
exit 127
fi
dd if=/dev/zero of=$1/test.out bs=$2 count=$3
umount $1
NOTE:  This is to flush file system cache
```

  - test_perf_read.sh

```
#!/bin/sh
if [ $# -ne 3 ]; then
       echo "Usage: $0 <mount dir> <buffer-size> <count>" 1>&2
exit 127
fi
dd if=$1/test.out of=/dev/null bs=$2 count=$3
umount $1
```

- **Test sequence:**
  1. mkdir /mnt/mmc
  2. mount -t vfat /dev/mmcblk0p1 /mnt/mmc NOTE: /dev/mmcblk0 could be used, if there are no partitions
  3. time ./test_perf_write.sh /mnt/mmc <buffer-size> <count>
  4. mount -t vfat /dev/mmcblk0p1 /mnt/mmc
  5. time ./test_perf_read.sh /mnt/mmc <buffer-size> <count>
  6. mount -t vfat /dev/mmcblk0p1 /mnt/mmc
  7. rm –f /mnt/mmc/test.out
  8. umount /mnt/mmc
  9. Repeat the steps 2 – 8, for each of the different buffer sizes – 4K, 64K, 128K, 256K, 512K and

1024K

10. For each iteration, the count is also calculated for a constant file size (count * buffer-size) of 256M

2.6.3.1.2 MMC/SD Driver - SD Write Performance: Low Latency Desktop Mode

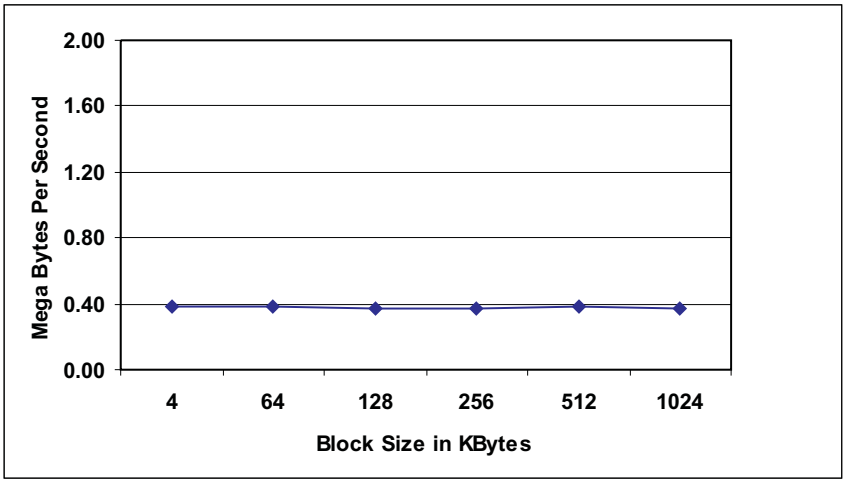

Figure 2-44. MMC/SD Driver - SD Write Performance Results (LLD) - DM355

Table 2-83. MMC/SD Driver - SD Write Performance Data (LLD) - DM355

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 217 MHZ	4	256	705.75	0.38
	64	256	708.10	0.38
	128	256	712.67	0.38
	256	256	712.90	0.38
	512	256	708.27	0.38
	1024	256	711.63	0.38

Test Setup:

- See Section 2.6.3.1.1, MMC/SD Driver - SD Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.6.3.1.3 MMC/SD Driver - SD Read Performance: Low Latency Desktop Mode

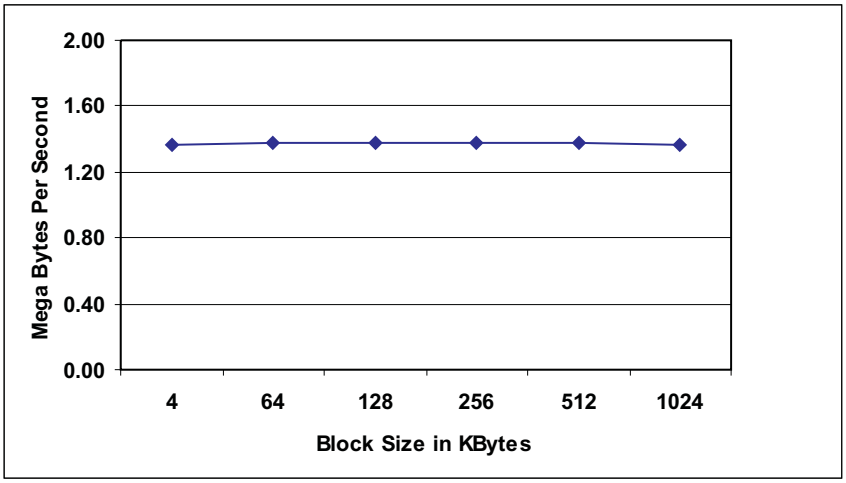

Figure 2-45. MMC/SD Driver - SD Read Performance Results (LLD) - DM355

Table 2-84. MMC/SD Driver - SD Read Performance Data (LLD) - DM355

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 217 MHZ	4	256	197.01	1.36
	64	256	195.42	1.37
	128	256	195.16	1.38
	256	256	195.22	1.38
	512	256	195.21	1.38
	1024	256	195.93	1.37

Test Setup:

- See Section 2.6.3.1.1, MMC/SD Driver - SD Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.6.3.1.4 MMC/SD Driver - SD Write Performance: Server Mode

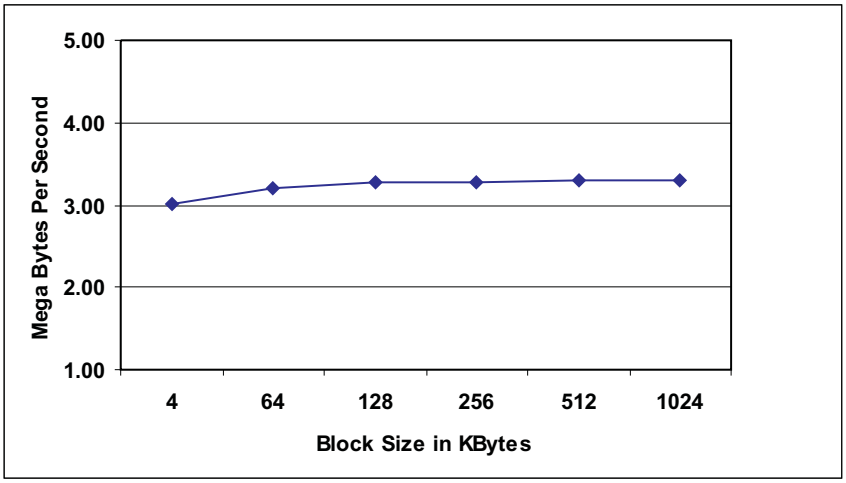

Figure 2-46. MMC/SD Driver - SD Write Performance Results (Server) - DM355

Table 2-85. MMC/SD Driver - SD Write Performance Data (Server) - DM355

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 217 MHZ	4	256	700.79	0.38
	64	256	692.92	0.39
	128	256	693.03	0.39
	256	256	697.86	0.38
	512	256	693.11	0.39
	1024	256	695.48	0.39

Test Setup:

- See Section 2.6.3.1.1, MMC/SD Driver - SD Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for server preemption mode.

2.6.3.1.5 MMC/SD Driver - SD Read Performance: Server Mode

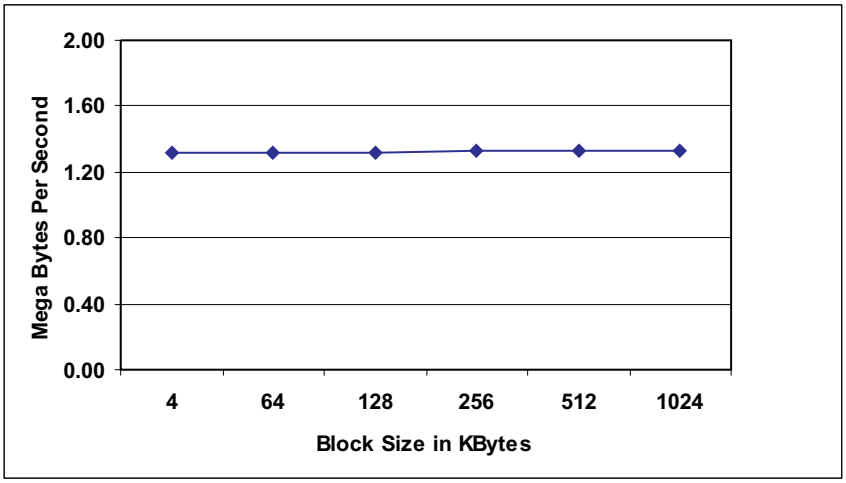

Figure 2-47. MMC/SD Driver - SD Read Performance Results (Server) - DM355

Table 2-86. MMC/SD Driver - SD Read Performance Data (Server) - DM355

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 217 MHZ	4	256	204.21	1.31
	64	256	202.91	1.32
	128	256	202.92	1.32
	256	256	202.60	1.32
	512	256	202.03	1.33
	1024	256	202.61	1.32

Test Setup:

- See Section 2.6.3.1.1, MMC/SD Driver - SD Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Server preemption mode.

2.6.3.1.6 MMC/SD Driver - SD Write Performance: Real-Time Preemption Mode

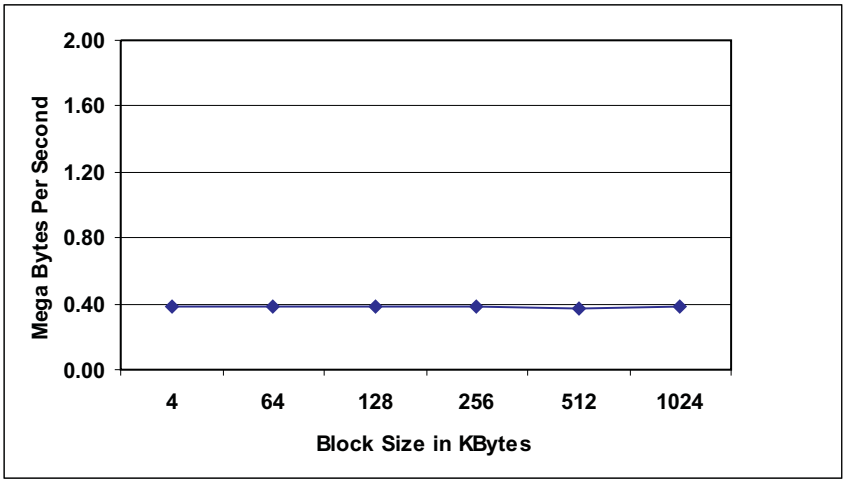

Figure 2-48. MMC/SD Driver - SD Write Performance Results (Real-Time) - DM355

Table 2-87. MMC/SD Driver - SD Write Performance Data (Real-Time) - DM355

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 217 MHZ	4	256	791.97	0.34
	64	256	791.58	0.34
	128	256	789.60	0.34
	256	256	788.99	0.34
	512	256	788.03	0.34
	1024	256	786.24	0.34

Test Setup:

- See Section 2.6.3.1.1, MMC/SD Driver - SD Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Real-Time preemption mode.

2.6.3.1.7 MMC/SD Driver - SD Read Performance: Real-Time Preemption Mode

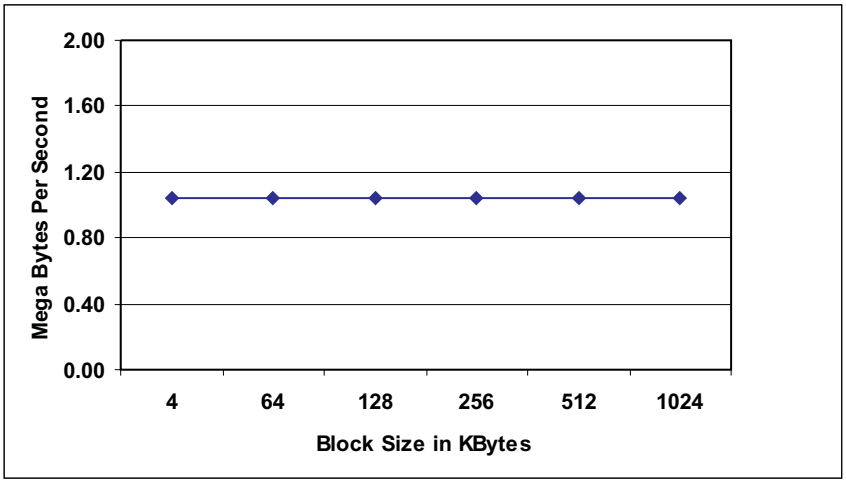

Figure 2-49. MMC/SD Driver - SD Read Performance Results (Real-Time) - DM355

Table 2-88. MMC/SD Driver - SD Read Performance Data (Real-Time) - DM355

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 217 MHZ	4	256	258.48	1.04
	64	256	256.62	1.05
	128	256	256.27	1.05
	256	256	256.41	1.05
	512	256	257.62	1.04
	1024	256	257.43	1.04

Test Setup:

- See Section 2.6.3.1.1, MMC/SD Driver - SD Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Real-Time preemption mode.

2.6.3.2 DM6446 EVM

The Table below shows the memory requirements for the MMC/SD Driver modules built for DM6446.

Table 2-89. MMC Driver Memory Statistics - DM6446

Configuration	Memory Statistics (in Bytes)			
	Program Memory	Data Memory		Total
		Initialized	Uninitialized	
davinci_mmc.ko	11420	372	0	11792
mmc_block.ko	5540	424	8	5972
mmc_core.ko	14196	880	16	15092

2.6.3.2.1 MMC/SD Driver - SD Read/Write Performance - Test Setup

- **Hardware used:** DM6446 Platform, 2GB 120X secure digital (SD) memory card (Model: SD2GBU)
- **Root Filesystem used:** ATA - EXT3 (/dev/hda1)
- **Test Filesystem used on the SD card:** VFAT
- **Mode tested:** Low Latency Desktop, Server and Real-Time preemption modes
- **Test Commands:**
  - test_perf_write.sh

```
#!/bin/sh
if [ $# -ne 3 ]; then
       echo "Usage: $0 <mount dir> <buffer-size> <count>" 1>&2
exit 127
fi
dd if=/dev/zero of=$1/test.out bs=$2 count=$3
umount $1
NOTE:  This is to flush file system cache
```

  - test_perf_read.sh

```
#!/bin/sh
if [ $# -ne 3 ]; then
       echo "Usage: $0 <mount dir> <buffer-size> <count>" 1>&2
exit 127
fi
dd if=$1/test.out of=/dev/null bs=$2 count=$3
umount $1
```

- **Test sequence:**
  1. mkdir /mnt/mmc
  2. mount -t vfat /dev/mmcblk0p1 /mnt/mmc NOTE: /dev/mmcblk0 could be used, if there are no partitions
  3. time ./test_perf_write.sh /mnt/mmc <buffer-size> <count>
  4. mount -t vfat /dev/mmcblk0p1 /mnt/mmc
  5. time ./test_perf_read.sh /mnt/mmc <buffer-size> <count>
  6. mount -t vfat /dev/mmcblk0p1 /mnt/mmc
  7. rm –f /mnt/mmc/test.out
  8. umount /mnt/mmc
  9. Repeat the steps 2 – 8, for each of the different buffer sizes – 4K, 64K, 128K, 256K, 512K and 1024K
  10. For each iteration, the count is also calculated for a constant file size (count * buffer-size) of 256M

2.6.3.2.2 MMC/SD Driver - SD Write Performance: Low Latency Desktop Mode

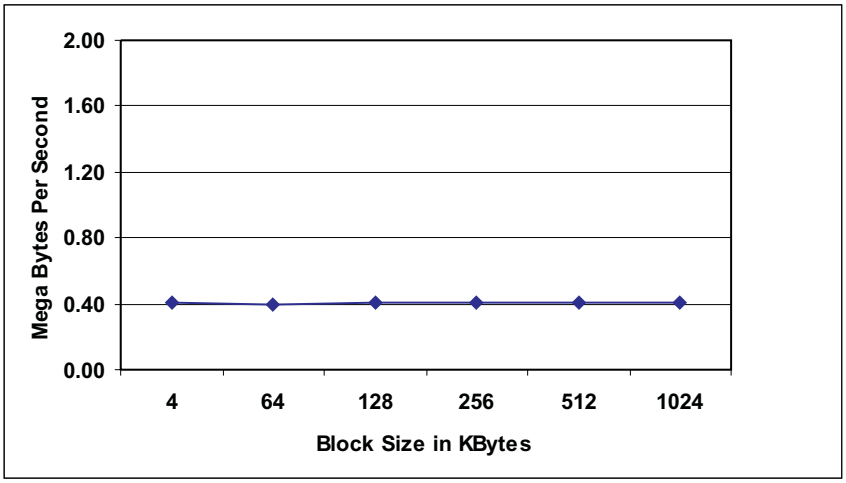

Figure 2-50. MMC/SD Driver - SD Write Performance Results (LLD) - DM6446

Table 2-90. MMC/SD Driver - SD Write Performance Data (LLD) - DM6446

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	256	664.75	0.40
	64	256	669.57	0.40
	128	256	663.73	0.40
	256	256	667.75	0.40
	512	256	660.89	0.41
	1024	256	660.98	0.41

Test Setup:

- See Section 2.6.3.2.1 , MMC/SD Driver - SD Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.6.3.2.3 MMC/SD Driver - SD Read Performance: Low Latency Desktop Mode

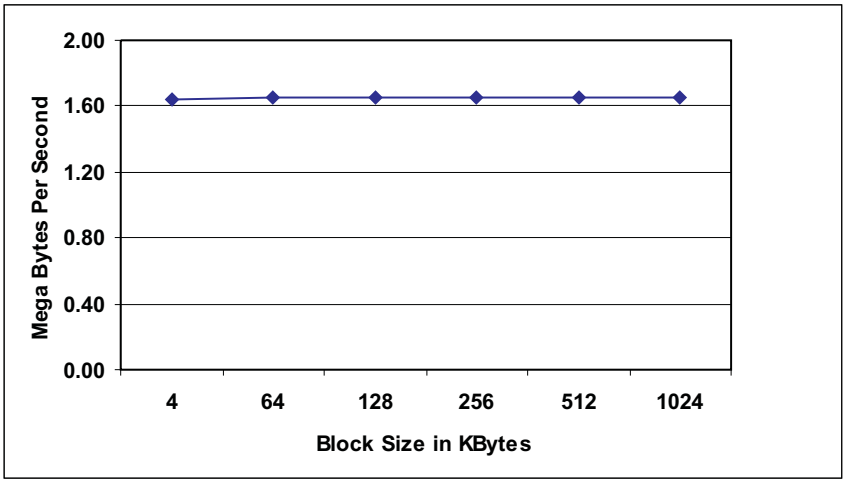

Figure 2-51. MMC/SD Driver - SD Read Performance Results (LLD) - DM6446

Table 2-91. MMC/SD Driver - SD Read Performance Data (LLD) - DM6446

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	256	163.60	1.64
	64	256	162.83	1.65
	128	256	162.59	1.65
	256	256	162.62	1.65
	512	256	162.60	1.65
	1024	256	162.80	1.65

Test Setup:

- See Section 2.6.3.2.1, MMC/SD Driver - SD Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.6.3.2.4 MMC/SD Driver - SD Write Performance: Server Mode

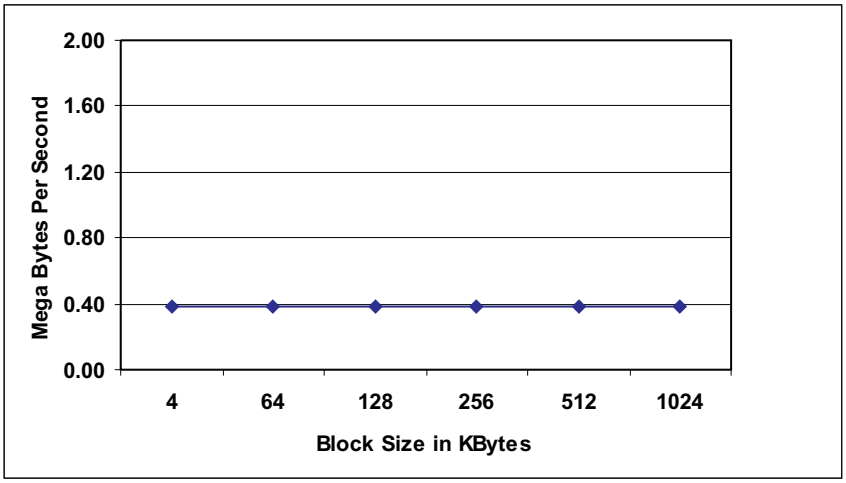

Figure 2-52. MMC/SD Driver - SD Write Performance Results (Server) - DM6446

Table 2-92. MMC/SD Driver - SD Write Performance Data (Server) - DM6446

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	256	707.87	0.38
	64	256	705.40	0.38
	128	256	708.03	0.38
	256	256	705.32	0.38
	512	256	710.79	0.38
	1024	256	706.04	0.38

Test Setup:

- See Section 2.6.3.2.1, MMC/SD Driver - SD Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for server preemption mode.

2.6.3.2.5 MMC/SD Driver - SD Read Performance: Server Mode

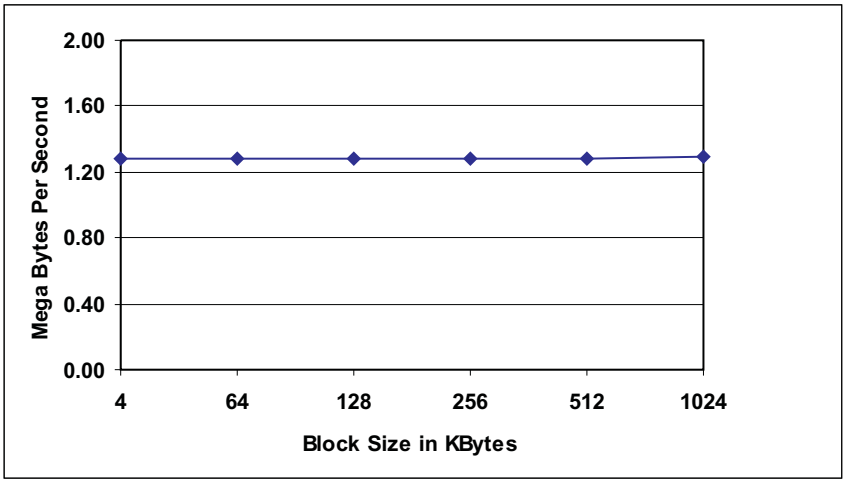

Figure 2-53. MMC/SD Driver - SD Read Performance Results (Server) - DM6446

Table 2-93. MMC/SD Driver - SD Read Performance Data (Server) - DM6446

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	256	209.92	1.28
	64	256	208.70	1.29
	128	256	208.52	1.29
	256	256	208.52	1.29
	512	256	209.43	1.28
	1024	256	208.26	1.29

Test Setup:

- See Section 2.6.3.2.1, MMC/SD Driver - SD Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Server preemption mode.

2.6.3.2.6 MMC/SD Driver - SD Write Performance: Real-Time Preemption Mode

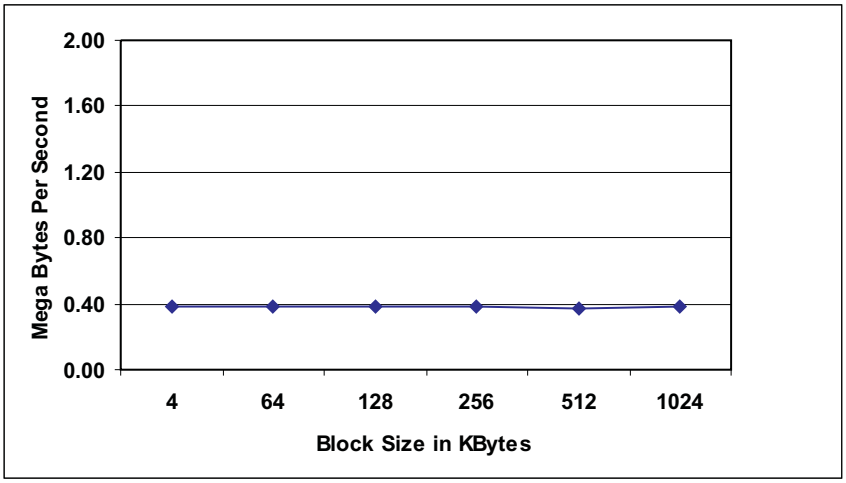

Figure 2-54. MMC/SD Driver - SD Write Performance Results (Real-Time) - DM6446

Table 2-94. MMC/SD Driver - SD Write Performance Data (Real-Time) - DM6446

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	256	709.12	0.38
	64	256	709.52	0.38
	128	256	710.53	0.38
	256	256	709.13	0.38
	512	256	715.91	0.37
	1024	256	706.76	0.38

Test Setup:

- See Section 2.6.3.2.1, MMC/SD Driver - SD Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Real-Time preemption mode.

2.6.3.2.7 MMC/SD Driver - SD Read Performance: Real-Time Preemption Mode

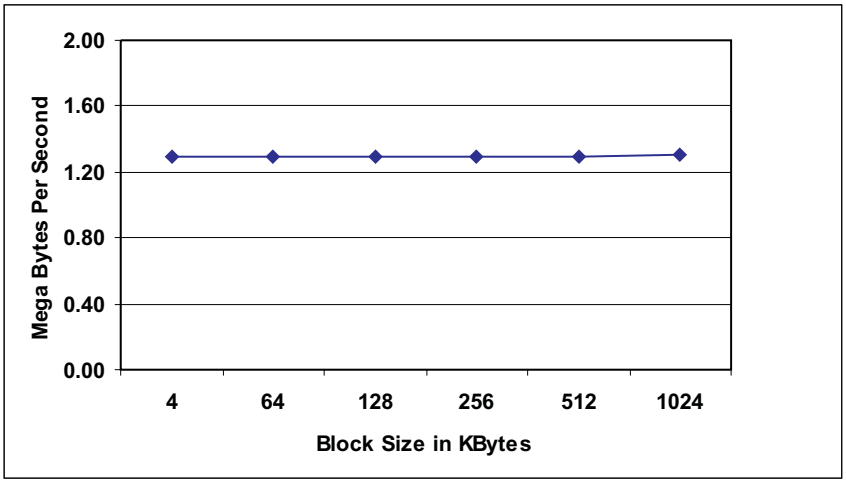

Figure 2-55. MMC/SD Driver - SD Read Performance Results (Real-Time) - DM6446

Table 2-95. MMC/SD Driver - SD Read Performance Data (Real-Time) - DM6446

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	256	208.46	1.29
	64	256	206.88	1.30
	128	256	207.17	1.30
	256	256	206.83	1.30
	512	256	207.09	1.30
	1024	256	206.25	1.30

Test Setup:

- See Section 2.6.3.2.1, MMC/SD Driver - SD Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Real-Time preemption mode.

2.6.4 Performance and Benchmarks With Custom SDIO Stack

2.6.4.1 DM355 EVM

2.6.4.1.1 Custom SDIO Stack - SD Write Performance: Low Latency Desktop (LLD) Mode

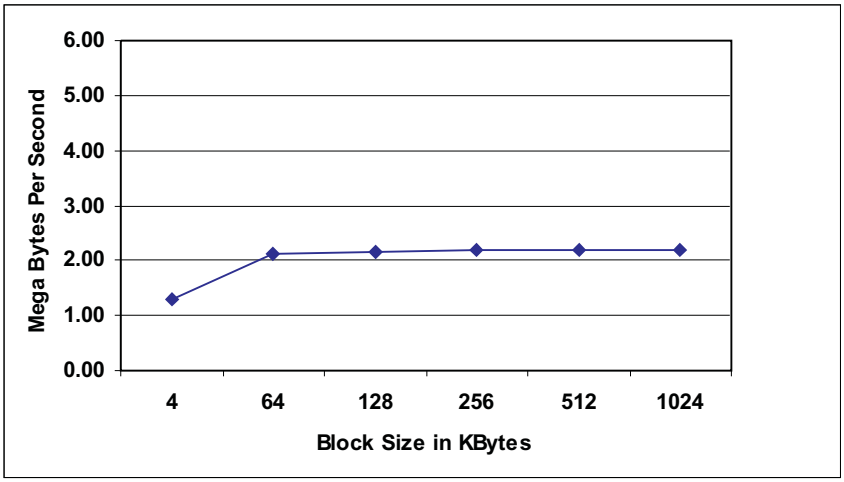

Figure 2-56. Custom SDIO Stack - SD Write Performace Results (LLD) - DM355

Table 2-96. Custom SDIO Stack - SD Write Performace Data (LLD) - DM355

Test Setup Information	Block Size in Kbytes	File Size in MB	Duration (Seconds)	Mega Bytes/sec
ARM Frequency = 217 MHZ	4	256	200	1.28
	64	256	121	2.12
	128	256	118	2.17
	256	256	117	2.19
	512	256	116	2.21
	1024	256	116	2.21

Test Setup:

- See Section 2.6.3.2.1, MMC/SD Driver - SD Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.6.4.1.2 Custom SDIO Stack - SD Read Performance: Low Latency Desktop (LLD) Mode

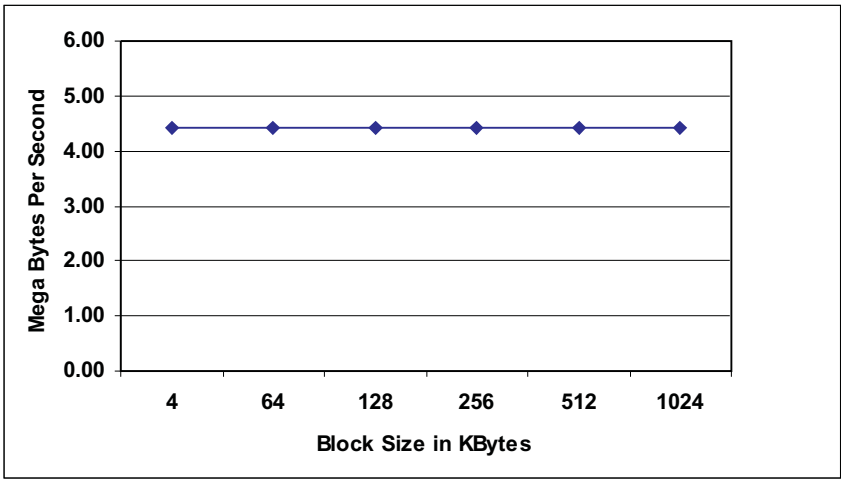

Figure 2-57. Custom SDIO Stack - SD Read Performace Results (LLD) - DM355

Table 2-97. Custom SDIO Stack - SD Read Performace Results (LLD) - DM355

Test Setup Information	Block Size in Kbytes	File Size in MB	Duration (Seconds)	Mega Bytes/sec
ARM Frequency = 217 MHZ	4	256	58	4.41
	64	256	58	4.41
	128	256	58	4.41
	256	256	58	4.41
	512	256	58	4.41
	1024	256	58	4.41

Test Setup:

- See Section 2.6.3.2.1, MMC/SD Driver - SD Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.6.4.1.3 Custom SDIO Stack - SD Write Performance: Server Mode

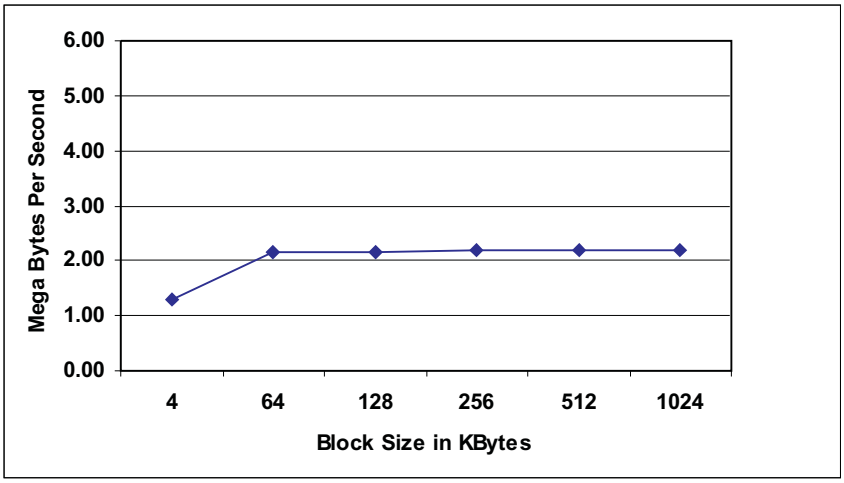

Figure 2-58. Custom SDIO Stack - SD Write Performace Results (Server) - DM355

Table 2-98. Custom SDIO Stack - SD Write Performace Data (Server) - DM355

Test Setup Information	Block Size in Kbytes	File Size in MB	Duration (Seconds)	Mega Bytes/sec
ARM Frequency = 217 MHZ	4	256	196	1.31
	64	256	119	2.15
	128	256	119	2.15
	256	256	117	2.19
	512	256	117	2.19
	1024	256	117	2.19

Test Setup:

- See Section 2.6.3.2.1, MMC/SD Driver - SD Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.6.4.1.4 Custom SDIO Stack - SD Read Performance: Server Mode

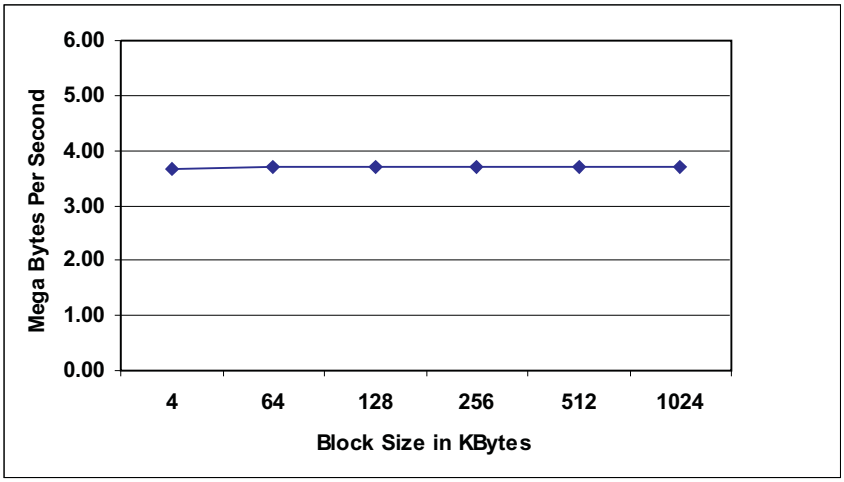

Figure 2-59. Custom SDIO Stack - SD Write Performace Results (Server) - DM355

Table 2-99. Custom SDIO Stack - SD Write Performace Data (Server) - DM355

Test Setup Information	Block Size in Kbytes	File Size in MB	Duration (Seconds)	Mega Bytes/sec
ARM Frequency = 217 MHZ	4	256	70	3.66
	64	256	69	3.71
	128	256	69	3.71
	256	256	69	3.71
	512	256	69	3.71
	1024	256	69	3.71

Test Setup:

- See Section 2.6.3.2.1, MMC/SD Driver - SD Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Server preemption mode.

2.6.4.1.5 Custom SDIO Stack - SD Write Performance: Real-Time Preemption Mode

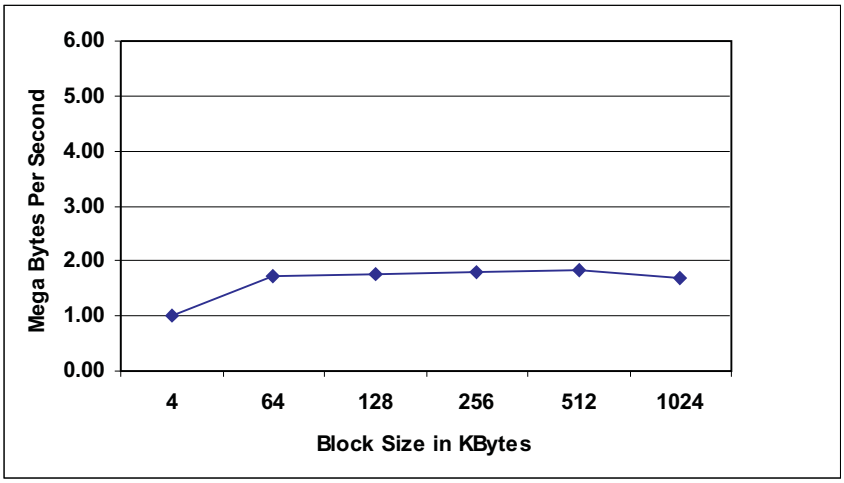

Figure 2-60. Custom SDIO Stack - SD Write Performace Results (Real-Time) - DM355

Table 2-100. Custom SDIO Stack - SD Write Performace Results (Real-Time) - DM355

Test Setup Information	Block Size in Kbytes	File Size in MB	Duration (Seconds)	Mega Bytes/sec
ARM Frequency = 217 MHZ	4	256	252	1.02
	64	256	150	1.71
	128	256	145	1.77
	256	256	142	1.80
	512	256	141	1.82
	1024	256	152	1.68

Test Setup:

- See Section 2.6.3.2.1, MMC/SD Driver - SD Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Real-Time preemption mode.

2.6.4.1.6 Custom SDIO Stack - SD Read Performance: Real-Time Preemption Mode

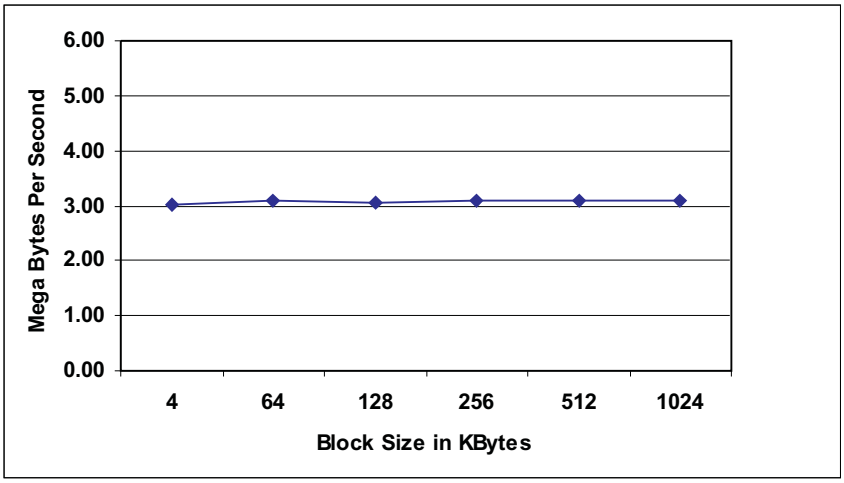

Figure 2-61. Custom SDIO Stack - SD Write Performace Results (Real-Time) - DM355

Table 2-101. Custom SDIO Stack - SD Write Performace Data (Real-Time) - DM355

Test Setup Information	Block Size in Kbytes	File Size in MB	Duration (Seconds)	Mega Bytes/sec
ARM Frequency = 217 MHZ	4	256	85	3.01
	64	256	83	3.08
	128	256	84	3.05
	256	256	83	3.08
	512	256	83	3.08
	1024	256	83	3.08

Test Setup:

- See Section 2.6.3.2.1, MMC/SD Driver - SD Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Real-Time preemption mode.

2.6.4.2 DM6446 EVM

2.6.4.2.1 Custom SDIO Stack - SD Write Performance: Low Latency Desktop Mode

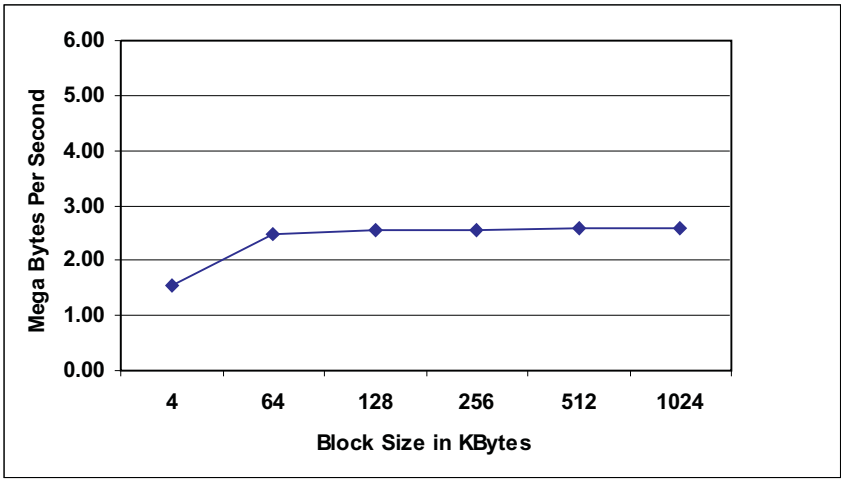

Figure 2-62. Custom SDIO Stack - SD Write Performance Results (LLD) – DM6446

Table 2-102. Custom SDIO Stack - SD Write Performance Data (LLD) - DM6446

Test Setup Information	Block Size in Kbytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequency = 297 MHz	4	256	165	1.55
	64	256	103	2.49
	128	256	101	2.53
	256	256	100	2.56
	512	256	99	2.59
	1024	256	99	2.59

Test Setup:

- See Section 2.6.3.2.1, MMC/SD Driver - SD Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.6.4.2.2 Custom SDIO Stack - SD Read Performance: Low Latency Desktop Mode

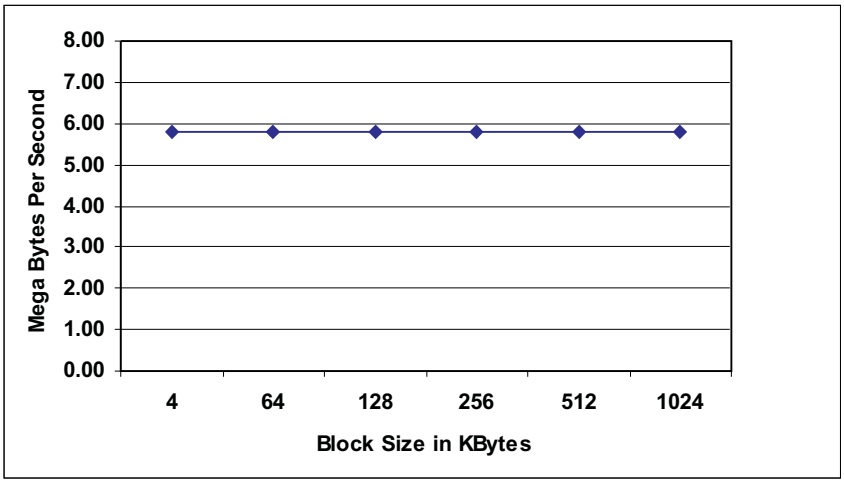

Figure 2-63. Custom SDIO Stack - SD Write Performance Results (LLD) - DM6446

Table 2-103. Custom SDIO Stack - SD Write Performance Data (LLD) - DM6446

Test Setup Information	Block Size in Kbytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequency = 297 MHz	4	256	44	5.82
	64	256	44	5.82
	128	256	44	5.82
	256	256	44	5.82
	512	256	44	5.82
	1024	256	44	5.82

Test Setup:

- See Section 2.6.3.2.1, MMC/SD Driver - SD Read/Write Performance: Test Setup
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode

2.6.4.2.3 Custom SDIO Stack - SD Write Performance: Server Mode

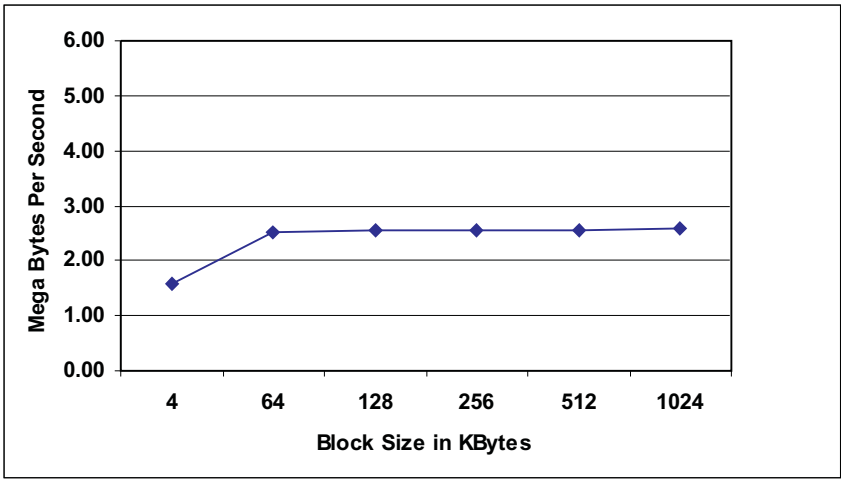

Figure 2-64. Custom SDIO Stack - SD Write Performance Results (Server) - DM6446

Table 2-104. Custom SDIO Stack - SD Write Performance Data (Server) - DM6446

Test Setup Information	Block Size in Kbytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequency = 297 MHz	4	256	162	1.58
	64	256	102	2.51
	128	256	101	2.53
	256	256	100	2.56
	512	256	100	2.56
	1024	256	99	2.59

Test Setup:

- See Section 2.6.3.2.1, MMC/SD Driver - SD Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.6.4.2.4 Custom SDIO Stack - SD Read Performance: Server Mode

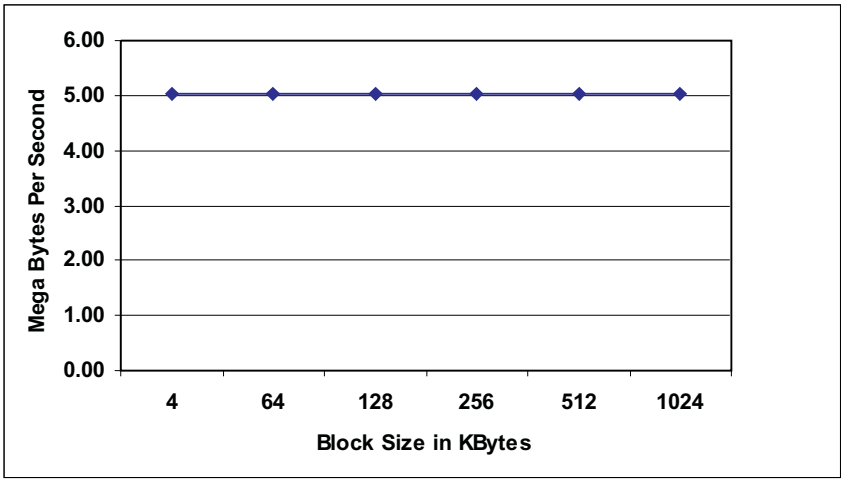

Figure 2-65. Custom SDIO Stack - SD Write Performance Results (Server) - DM6446

Table 2-105. Custom SDIO Stack - SD Write Performance Data (Server) - DM6446

Test Setup Information	Block Size in Kbytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequency = 297 MHz	4	256	51	5.02
	64	256	51	5.02
	128	256	51	5.02
	256	256	51	5.02
	512	256	51	5.02
	1024	256	51	5.02

Test Setup:

- See Section 2.6.3.2.1, MMC/SD Driver - SD Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Server preemption mode.

2.6.4.2.5 Custom SDIO Stack - SD Write Performance: Real-Time Preemption Mode

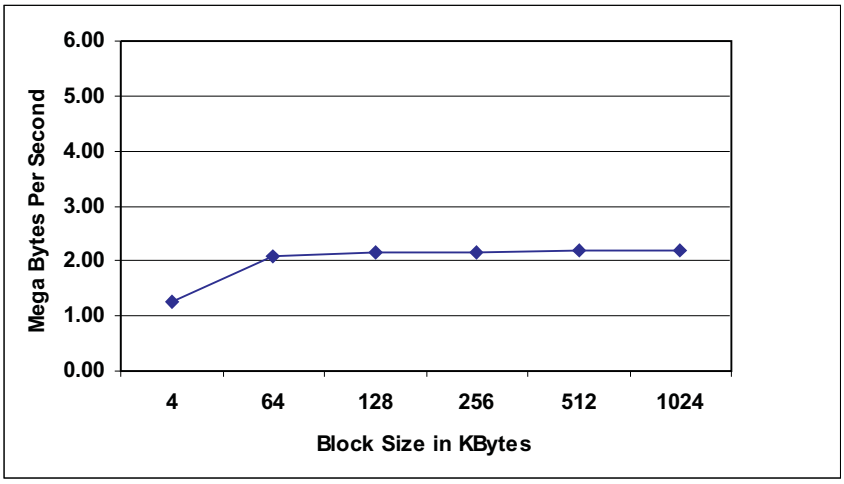

Figure 2-66. Custom SDIO Stack - SD Write Performance Results (Real-Time) - DM6446

Table 2-106. Custom SDIO Stack - SD Write Performance Data (Real-Time) - DM6446

Test Setup Information	Block Size in Kbytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequency = 297 MHz	4	256	202	1.27
	64	256	123	2.08
	128	256	119	2.15
	256	256	118	2.17
	512	256	117	2.19
	1024	256	116	2.21

Test Setup:

- See Section 2.6.3.2.1, MMC/SD Driver - SD Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Real-Time preemption mode.

2.6.4.2.6 Custom SDIO Stack - SD Read Performance: Real-Time Preemption Mode

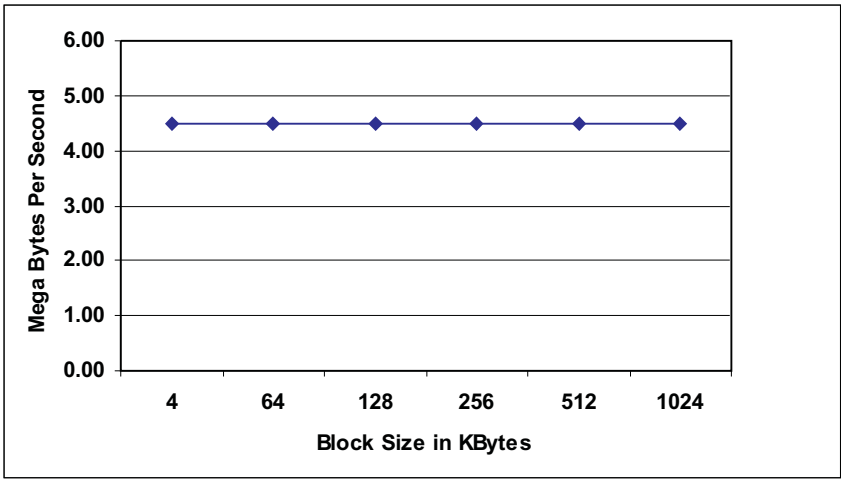

Figure 2-67. Custom SDIO Stack - SD Write Performance Results (Real-Time) - DM6446

Table 2-107. Custom SDIO Stack - SD Write Performance Data (Real-Time) - DM6446

Test Setup Information	Block Size in Kbytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequency = 297 MHz	4	256	57	4.49
	64	256	57	4.49
	128	256	57	4.49
	256	256	57	4.49
	512	256	57	4.49
	1024	256	57	4.49

Test Setup:

- See Section 2.6.3.2.1, MMC/SD Driver - SD Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Real-Time preemption mode.

2.7 NAND Driver

The NAND driver is implemented as a block driver, compliant with the standard MTD driver. Figure 2-68 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.

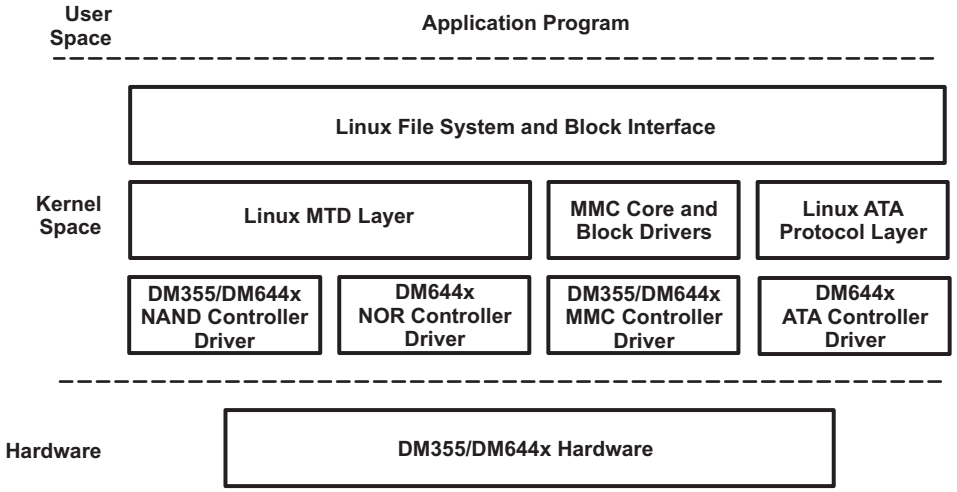

Figure 2-68. Linux Kernel NAND Driver Architecture

The NAND driver supports following features:

- The driver supports Big Block NAND access
- The driver supports 4-bit hardware ECC
- The driver supports YAFFS (on DM6446), YAFFS2 (on DM355) and JFFS2 file systems
- The NAND flash space is divided into the following partitions:
  - Read-only bootloader partition for UBL and U-Boot (3.75MB on DM355 and 2MB on DM6446)
  - Read-only partition for Environment variables (256KB on DM355 and 128KB on DM6446)
  - 4 MB read-write partition for the Kernel image
  - Read-write partition for the reduced target YAFFS2 file system (512MB, only on DM355)
  - Remaining (≈1.45GB on DM355 and ≈55MB on DM6446) read-write partition for the MV target file system

2.7.1 Support and Constraints

Features not supported	none
Constraints	The rmmod command to unlink the NAND module is not supported. If there is more than one partition, it fails with a "Device Busy" message
Supported system calls	open(), close(), write(), read(), pwrite(), pread(), ioctl()
Supported IOCTLs	see Table 2-108

Table 2-108. Supported 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).

Table 2-108. Supported IOCTLs (continued)

Constant	Description
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.

2.7.2 Performance and Benchmarks

2.7.2.1 DM355 EVM

Table 2-109 shows the memory requirements for the NAND Driver modules built for DM355.

Table 2-109. NAND Driver Memory Statistics - DM355

Configuration	Memory Statistics (in Bytes)			
	Program Memory	Data Memory		Total
		Initialized	Uninitialized	
nand.ko	27536	1352	2052	30940
nand_ecc.ko	1228	256	0	1484
nand_ids.ko	1380	1768	0	3148
nand_davinci.ko	6580	600	68	7248

2.7.2.1.1 NAND Driver - Read/Write Performance: Test Setup

- **Hardware used:** DM355 Platform, MT29F16G08FAA: 2GB – SLC, MT29F16G08QAA: 2GB – MLC
- **Root Filesystem used:** YAFFS2 (/dev/mtdblock3)
- **Test Filesystem used on the NAND Flash:** YAFFS2 (/dev/mtdblock4)
- **Mode tested:** Low Latency Desktop, Server and Real-Time preemption modes
- **Test Commands:**
  - test_perf_write.sh

```
#!/bin/sh
if [ $# -ne 3 ]; then
       echo "Usage: $0 <mount dir> <buffer-size> <count>" 1>&2
exit 127
fi
dd if=/dev/zero of=$1/test.out bs=$2 count=$3
umount $1
NOTE:  This is to flush file system cache
```

  - test_perf_read.sh

```
#!/bin/sh
if [ $# -ne 3 ]; then
       echo "Usage: $0 <mount dir> <buffer-size> <count>" 1>&2
exit 127
fi
dd if=$1/test.out of=/dev/null bs=$2 count=$3
umount $1
```

- **Test sequence:**
  1. mkdir /mnt/nand
  2. mount –t yaffs2 /dev/mtdblock4 /mnt/nand
  3. time ./test_perf_write.sh /mnt/nand <buffer-size> <count>
  4. mount –t yaffs2 /dev/mtdblock4 /mnt/nand
  5. time ./test_perf_read.sh /mnt/nand <buffer-size> <count>

6. mount –t yaffs2 /dev/mtdblock4 /mnt/nand
7. rm –f /mnt/nand/test.out
8. umount /mnt/nand
9. Repeat the steps 2 – 8, for each of the different buffer sizes – 4K, 64K, 128K, 256K, 512K and 1024K
10. For each iteration, the count is also calculated for a constant file size (count * buffer-size) of 256M

2.7.2.1.2 NAND Driver - SLC Write Performance: Low Latency Desktop Mode

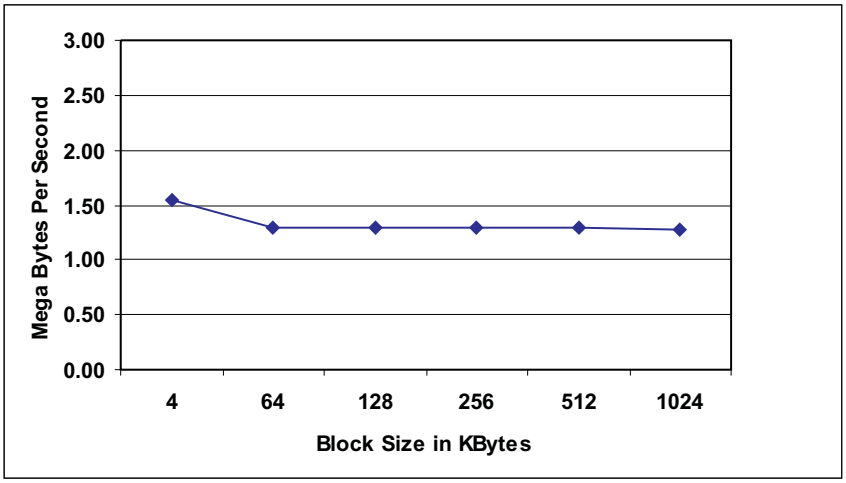

Figure 2-69. SLC Write Performance: Low Latency Desktop Mode

Table 2-110. NAND Driver - SLC Write (LLD)

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 217 MHZ	4	256	174.21	1.54
	64	256	208.70	1.29
	128	256	207.00	1.30
	256	256	207.54	1.29
	512	256	208.91	1.28
	1024	256	209.18	1.28

Test Setup:

- See Section 2.7.2.1.1, NAND Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.7.2.1.3 NAND Driver - SLC Read Performance: Low Latency Desktop Mode

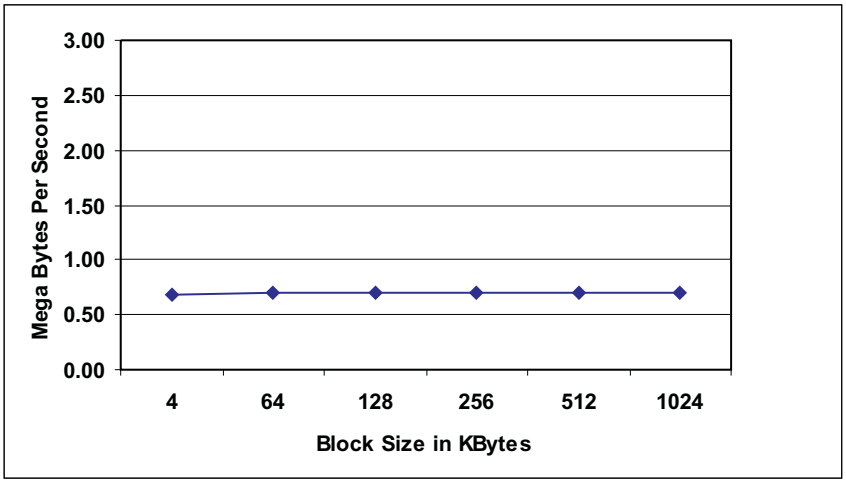

Figure 2-70. NAND Driver - SLC Read (LLD)

Table 2-111. NAND Driver - SLC Read (LLD)

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 217 MHZ	4	256	388.81	0.69
	64	256	387.53	0.69
	128	256	387.43	0.69
	256	256	387.54	0.69
	512	256	386.52	0.69
	1024	256	387.77	0.69

Test Setup:

- See Section 2.7.2.1.1, NAND Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.7.2.1.4 NAND Driver - SLC Write Performance: Server Mode

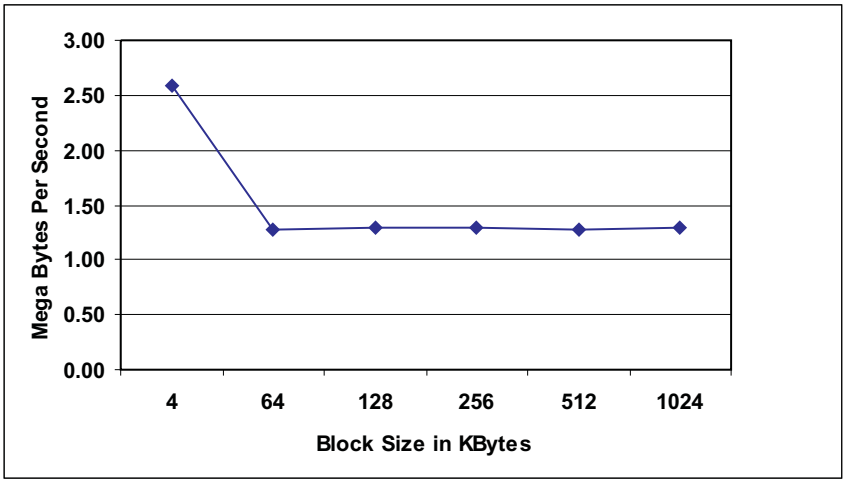

Figure 2-71. NAND Driver - SLC Write (Server)

Table 2-112. NAND Driver - SLC Write (Server)

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 217 MHZ	4	256	104.11	2.58
	64	256	210.69	1.27
	128	256	206.67	1.30
	256	256	206.67	1.30
	512	256	209.23	1.28
	1024	256	208.79	1.29

Test Setup:

- See Section 2.7.2.1.1, NAND Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Server preemption mode.

2.7.2.1.5 NAND Driver - SLC Read Performance: Server Mode

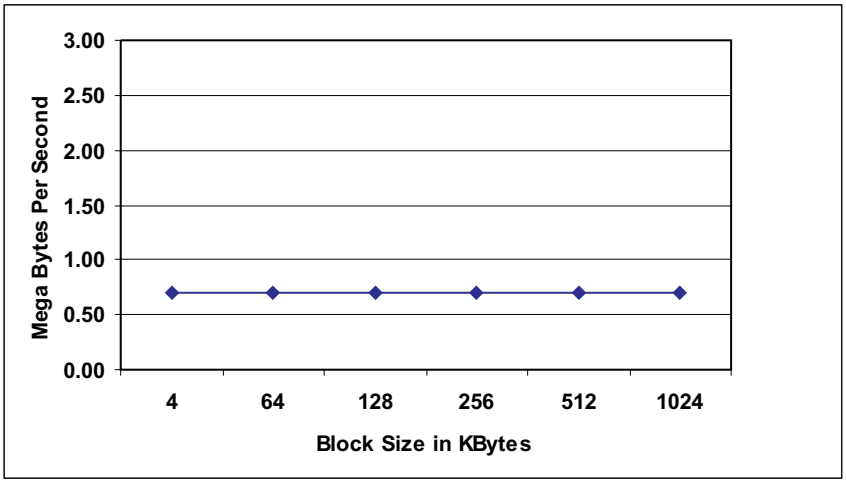

Figure 2-72. NAND Driver - SLC Read (Server)

Table 2-113. NAND Driver - SLC Read (Server)

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 217 MHZ	4	256	387.92	0.69
	64	256	386.44	0.69
	128	256	386.47	0.69
	256	256	386.41	0.69
	512	256	386.61	0.69
	1024	256	386.61	0.69

Test Setup:

- See Section 2.7.2.1.1, NAND Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Server preemption mode.

2.7.2.1.6 NAND Driver - SLC Write Performance: Real-Time Preemption Mode

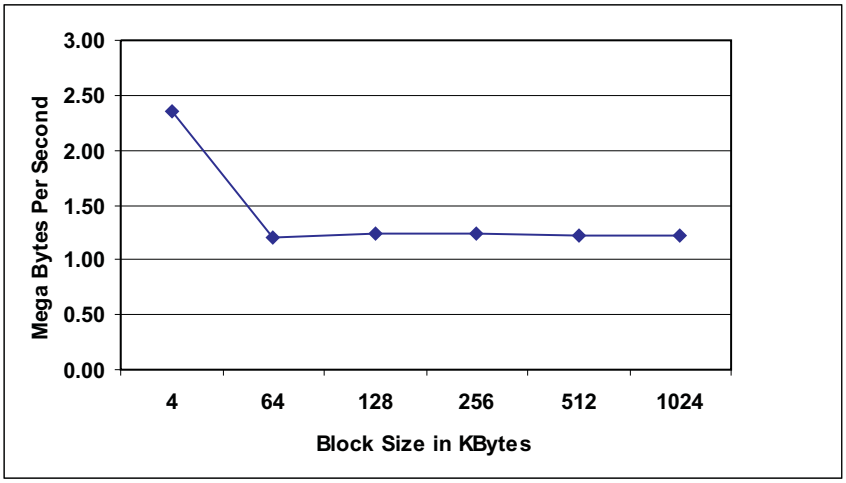

Figure 2-73. NAND Driver - SLC Write (RT)

Table 2-114. NAND Driver - SLC Write (RT)

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 217 MHZ	4	256	113.85	2.36
	64	256	221.87	1.21
	128	256	218.12	1.23
	256	256	217.17	1.24
	512	256	220.04	1.22
	1024	256	219.93	1.22

Test Setup:

- See Section 2.7.2.1.1, NAND Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Real-Time preemption mode.

2.7.2.1.7 NAND Driver - SLC Read Performance: Real-Time Preemption Mode

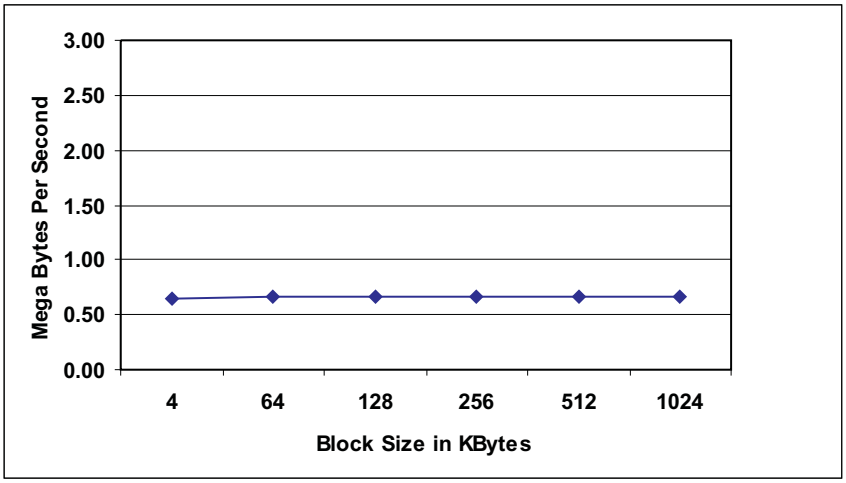

Figure 2-74. NAND Driver - SLC Read (RT)

Table 2-115. NAND Driver - SLC Read (RT)

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 217 MHZ	4	256	410.86	0.65
	64	256	408.42	0.66
	128	256	408.41	0.66
	256	256	408.39	0.66
	512	256	408.54	0.66
	1024	256	408.42	0.66

Test Setup:

- See Section 2.7.2.1.1, NAND Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Real-Time preemption mode.

2.7.2.1.8 NAND Driver - MLC Write Performance: Low Latency Desktop Mode

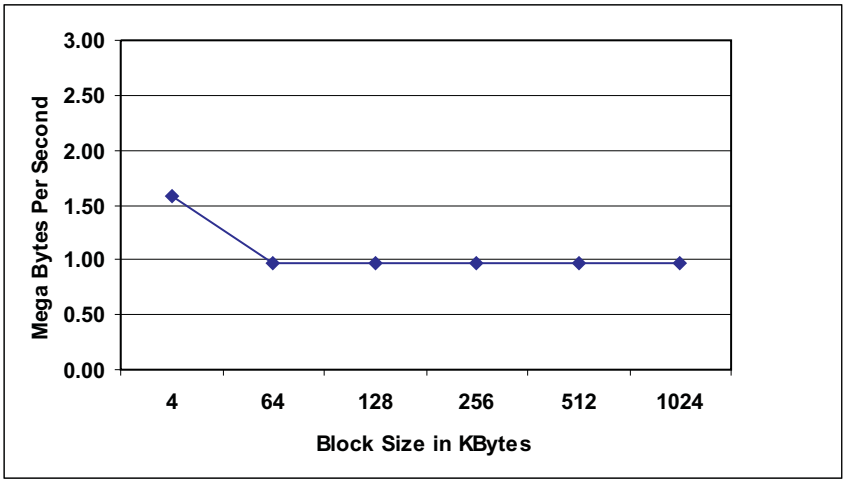

Figure 2-75. NAND Driver - MLC Write (LLD)

Table 2-116. NAND Driver - MLC Write (LLD)

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 217 MHZ	4	256	169.78	1.58
	64	256	277.71	0.97
	128	256	278.04	0.97
	256	256	276.10	0.97
	512	256	276.47	0.97
	1024	256	276.74	0.97

Test Setup:

- See Section 2.7.2.1.1, NAND Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.7.2.1.9 NAND Driver - MLC Read Performance: Low Latency Desktop Mode

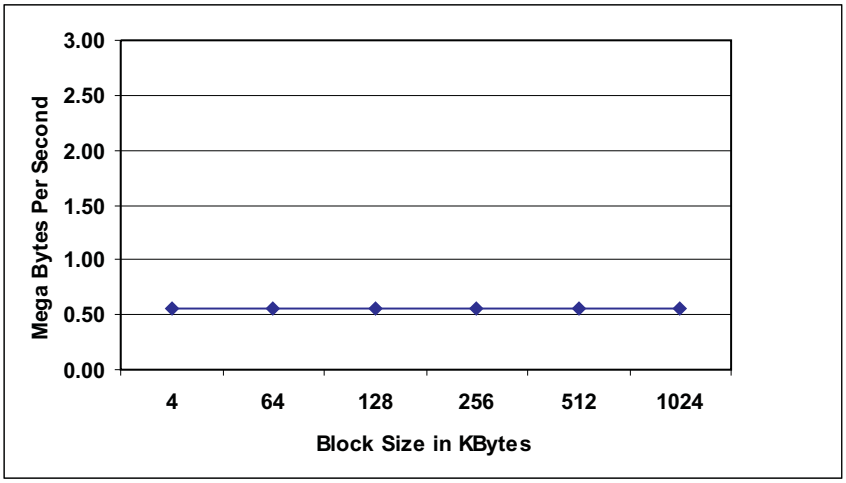

Figure 2-76. NAND Driver - MLC Read (LLD)

Table 2-117. NAND Driver - MLC Read (LLD)

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 217 MHZ	4	256	480.91	0.56
	64	256	479.81	0.56
	128	256	481.77	0.56
	256	256	481.52	0.56
	512	256	481.72	0.56
	1024	256	481.51	0.56

Test Setup:

- See Section 2.7.2.1.1, NAND Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.7.2.1.10 NAND Driver - MLC Write Performance: Server Mode

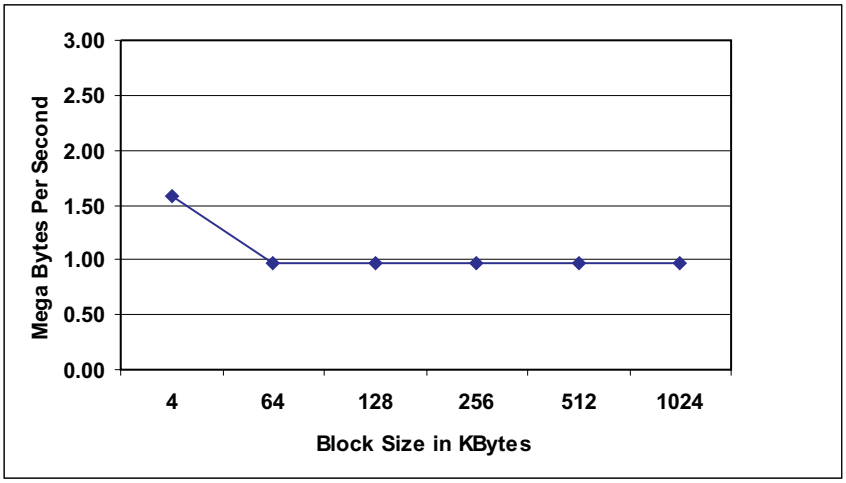

Figure 2-77. NAND Driver - MLC Write (Server)

Table 2-118. NAND Driver - MLC Write (Server)

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 217 MHZ	4	256	168.94	1.59
	64	256	277.39	0.97
	128	256	277.82	0.97
	256	256	276.14	0.97
	512	256	276.23	0.97
	1024	256	276.38	0.97

Test Setup:

- See Section 2.7.2.1.1, NAND Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Server preemption mode.

2.7.2.1.11 NAND Driver - MLC Read Performance: Server Mode

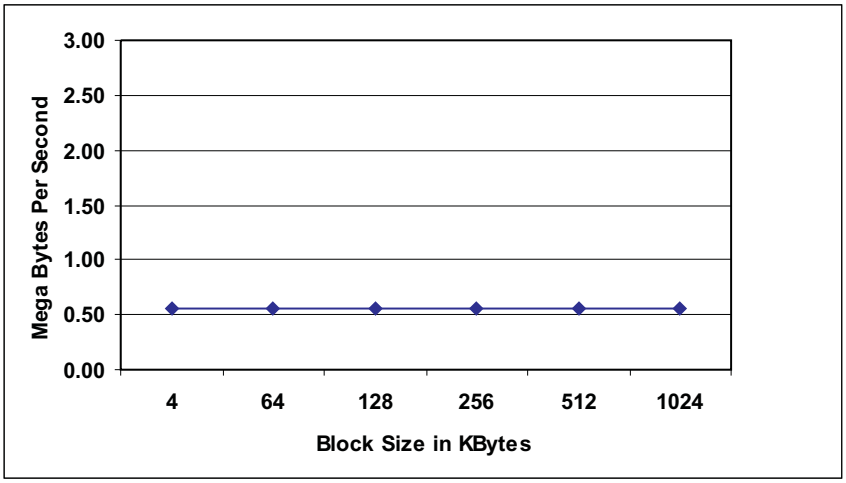

Figure 2-78. NAND Driver - MLC Read (Server)

Table 2-119. NAND Driver - MLC Read (Server)

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 217 MHZ	4	256	481.24	0.56
	64	256	479.26	0.56
	128	256	481.08	0.56
	256	256	480.83	0.56
	512	256	480.51	0.56
	1024	256	480.51	0.56

Test Setup:

- See Section 2.7.2.1.1, NAND Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Server preemption mode.

2.7.2.1.12 NAND Driver - MLC Write Performance: Real-Time Preemption Mode

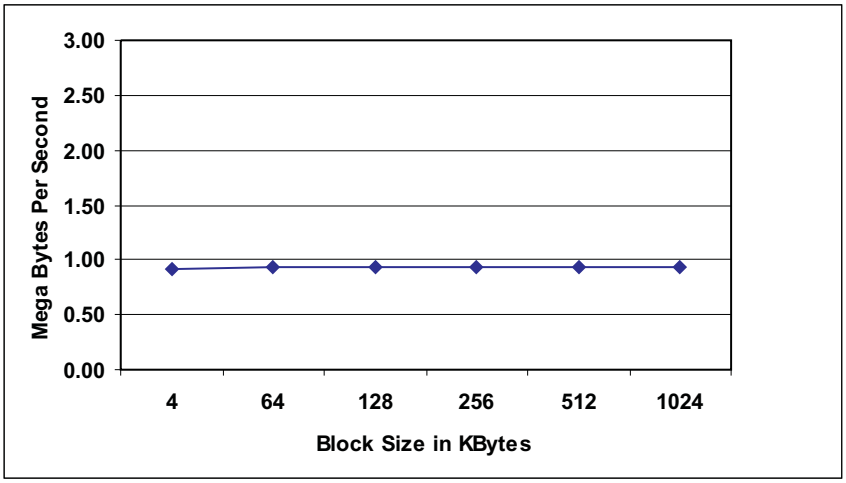

Figure 2-79. NAND Driver - MLC Write (RT)

Table 2-120. NAND Driver - MLC Write (RT)

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 217 MHZ	4	256	295.57	0.91
	64	256	289.02	0.93
	128	256	288.69	0.93
	256	256	287.00	0.94
	512	256	287.63	0.93
	1024	256	287.45	0.93

Test Setup:

- See Section 2.7.2.1.1, NAND Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Real-Time preemption mode.

2.7.2.1.13 NAND Driver - MLC Read Performance: Real-Time Preemption Mode

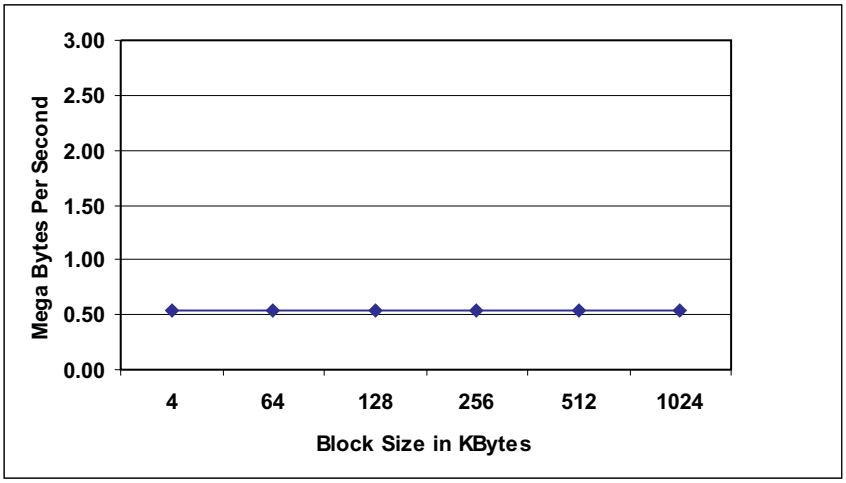

Figure 2-80. NAND Driver - MLC Read (RT)

Table 2-121. NAND Driver - MLC Read (RT)

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 217 MHZ	4	256	504.48	0.53
	64	256	502.86	0.53
	128	256	504.36	0.53
	256	256	504.21	0.53
	512	256	504.13	0.53
	1024	256	504.07	0.53

Test Setup:

- See Section 2.7.2.1.1, NAND Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Real-Time preemption mode.

2.7.2.2 DM6446 EVM

Table 2-122 shows the memory requirements for the MMC/SD Driver modules built for DM6446.

Table 2-122. NAND Driver Memory Statistics - DM6446

Configuration	Memory Statistics (in Bytes)			
	Program Memory	Data Memory		Total
		Initialized	Uninitialized	
nand.ko	27552	1352	2052	30956
nand_ecc.ko	1228	256	0	1484
nand_ids.ko	1380	1768	0	3148
nand_davinci.ko	6584	600	68	7252

2.7.2.2.1 NAND Driver - Read/Write Performance: Test Setup

- **Hardware used:** DM6446 Platform, Samsung 60MB SEC 613 F170
- **Root Filesystem used:** NFS
- **Test Filesystem used on the NAND Flash:** JFFS2 (/dev/mtdblock3)
- **Mode tested:** Low Latency Desktop, Server and Real-Time preemption modes
- **Test Commands:**
  - test_perf_write.sh

```
#!/bin/sh
if [ $# -ne 3 ]; then
       echo "Usage: $0 <mount dir> <buffer-size> <count>" 1>&2
exit 127
fi
dd if=/dev/zero of=$1/test.out bs=$2 count=$3
umount $1
NOTE:  This is to flush file system cache
```

  - test_perf_read.sh

```
#!/bin/sh
if [ $# -ne 3 ]; then
       echo "Usage: $0 <mount dir> <buffer-size> <count>" 1>&2
exit 127
fi
dd if=$1/test.out of=/dev/null bs=$2 count=$3
umount $1
```

- **Test sequence:**
  1. mkdir /mnt/nand
  2. mount –t jffs2 /dev/mtdblock3 /mnt/nand
  3. time ./test_perf_write.sh /mnt/nand <buffer-size> <count>
  4. mount –t jffs2 /dev/mtdblock3 /mnt/nand
  5. time ./test_perf_read.sh /mnt/nand <buffer-size> <count>
  6. mount –t jffs2 /dev/mtdblock3 /mnt/nand
  7. rm –f /mnt/nand/test.out
  8. umount /mnt/nand
  9. Repeat the steps 2 – 8, for each of the different buffer sizes – 4K, 64K, 128K, 256K, 512K and 1024K
  10. For each iteration, the count is also calculated for a constant file size (count * buffer-size) of 16M

2.7.2.2.2 NAND Driver - Write Performance: Low Latency Desktop Mode

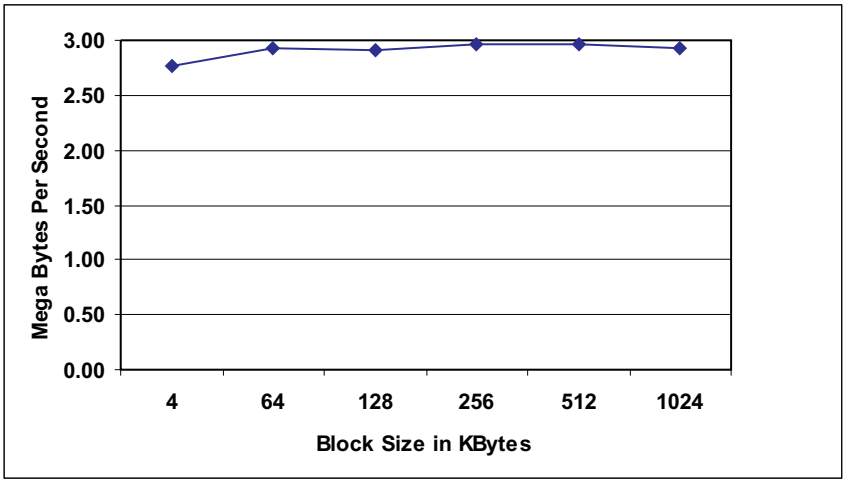

Figure 2-81. NAND Driver - Write (LLD)

Table 2-123. NAND Driver - Write (LLD)

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	16	6.08	2.76
	64	16	5.73	2.93
	128	16	5.75	2.92
	256	16	5.67	2.96
	512	16	5.66	2.96
	1024	16	5.72	2.93

Test Setup:

- See Section 2.7.2.2.1, NAND Driver - Read/Write Performance - Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.7.2.2.3 NAND Driver - Read Performance: Low Latency Desktop Mode

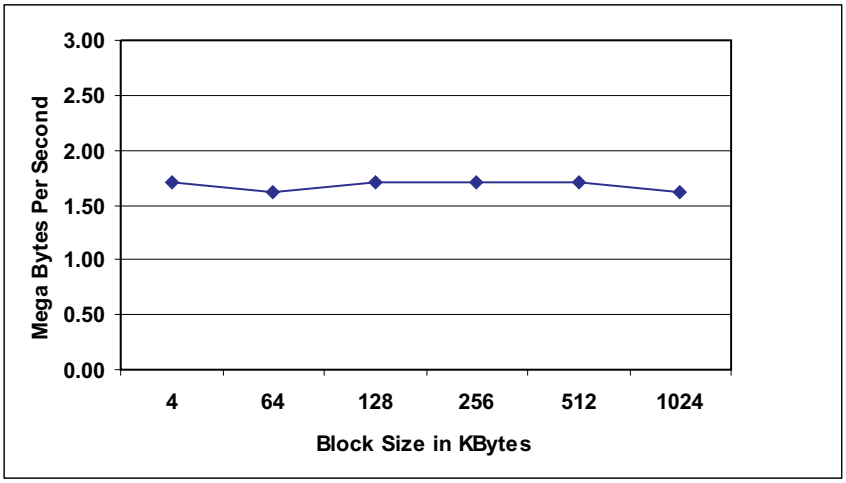

Figure 2-82. NAND Driver - Read (LLD)

Table 2-124. NAND Driver - Read (LLD)

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	16	9.87	1.70
	64	16	10.38	1.62
	128	16	9.80	1.71
	256	16	9.79	1.71
	512	16	9.78	1.72
	1024	16	10.33	1.62

Test Setup:

- See Section 2.7.2.2.1, NAND Driver - Read/Write Performance - Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.7.2.2.4 NAND Driver - Write Performance: Server Mode

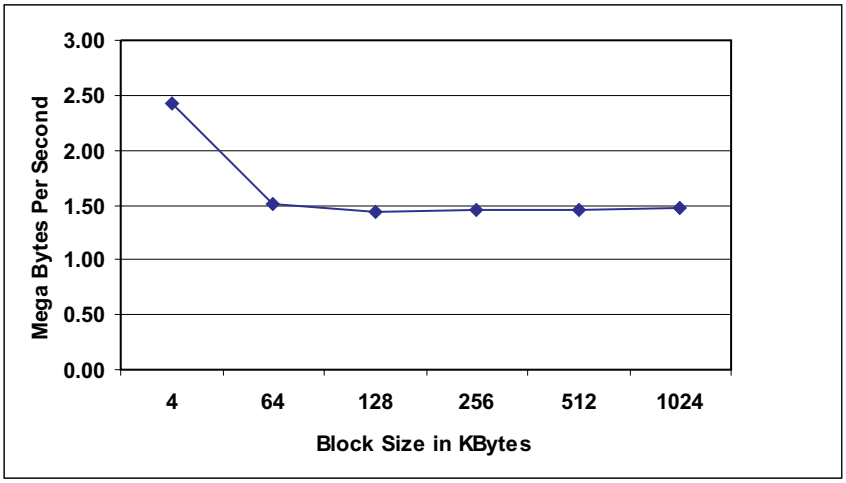

Figure 2-83. NAND Driver - Write (Server)

Table 2-125. NAND Driver - Write (Server)

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHz	4	16	6.90	2.43
	64	16	11.12	1.51
	128	16	11.68	1.44
	256	16	11.47	1.46
	512	16	11.53	1.46
	1024	16	11.41	1.47

Test Setup:

- See Section 2.7.2.2.1, NAND Driver - Read/Write Performance - Test Setup.
- Rebuild the Linux Kernel for Server preemption mode.

2.7.2.2.5 NAND Driver - Read Performance: Server Mode

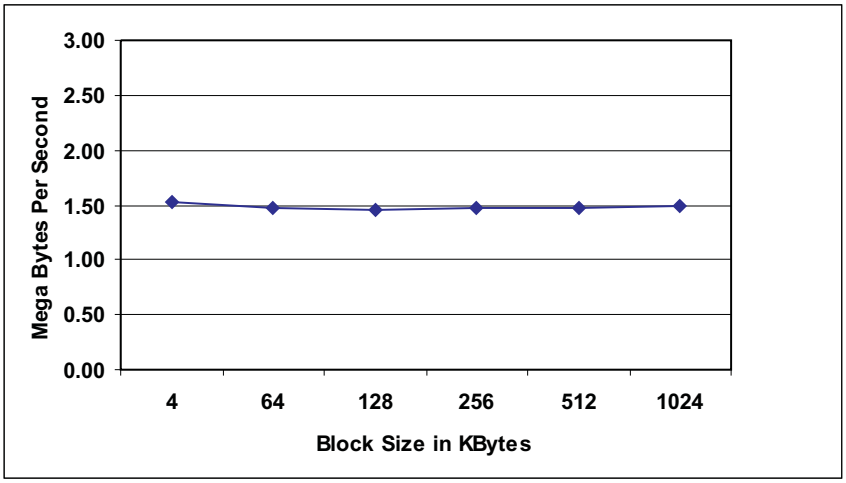

Figure 2-84. NAND Driver - Read (Server)

Table 2-126. NAND Driver - Read (Server)

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	16	11.03	1.52
	64	16	11.40	1.47
	128	16	11.51	1.46
	256	16	11.41	1.47
	512	16	11.41	1.47
	1024	16	11.26	1.49

Test Setup:

- See Section 2.7.2.2.1, NAND Driver - Read/Write Performance - Test Setup.
- Rebuild the Linux Kernel for Server preemption mode.

2.7.2.2.6 NAND Driver - Write Performance: Real-Time Preemption Mode

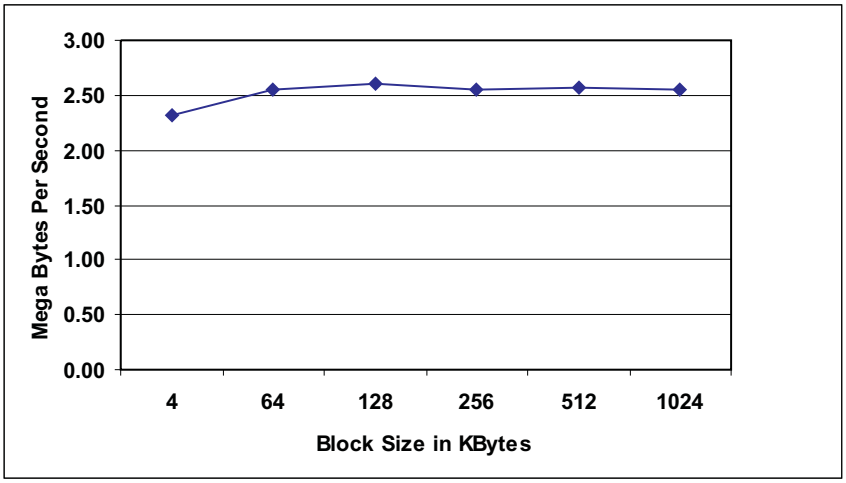

Figure 2-85. NAND Driver - Write (RT)

Table 2-127. NAND Driver - Write (RT)

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	16	7.24	2.32
	64	16	6.57	2.55
	128	16	6.44	2.61
	256	16	6.60	2.54
	512	16	6.52	2.57
	1024	16	6.57	2.55

Test Setup:

- See Section 2.7.2.2.1, NAND Driver - Read/Write Performance - Test Setup.
- Rebuild the Linux Kernel for Real-Time preemption mode.

2.7.2.2.7 NAND Driver - Read Performance: Real-Time Preemption Mode

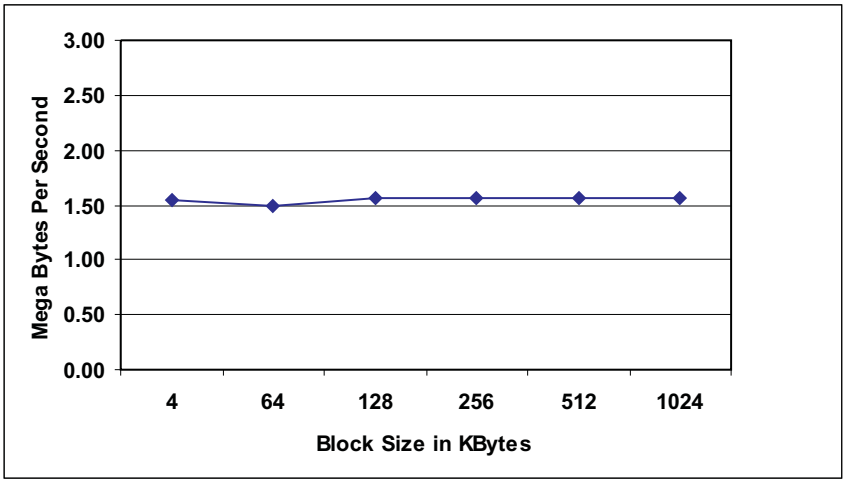

Figure 2-86. NAND Driver - Read (RT)

Table 2-128. NAND Driver - Read (RT)

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	16	10.84	1.55
	64	16	11.31	1.48
	128	16	10.76	1.56
	256	16	10.76	1.56
	512	16	10.70	1.57
	1024	16	10.73	1.56

Test Setup:

- See Section 2.7.2.2.1, NAND Driver - Read/Write Performance - Test Setup.
- Rebuild the Linux Kernel for Real-Time preemption mode.

2.8 NOR Driver

The NOR driver complies with the standard MTD driver. The NOR 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.

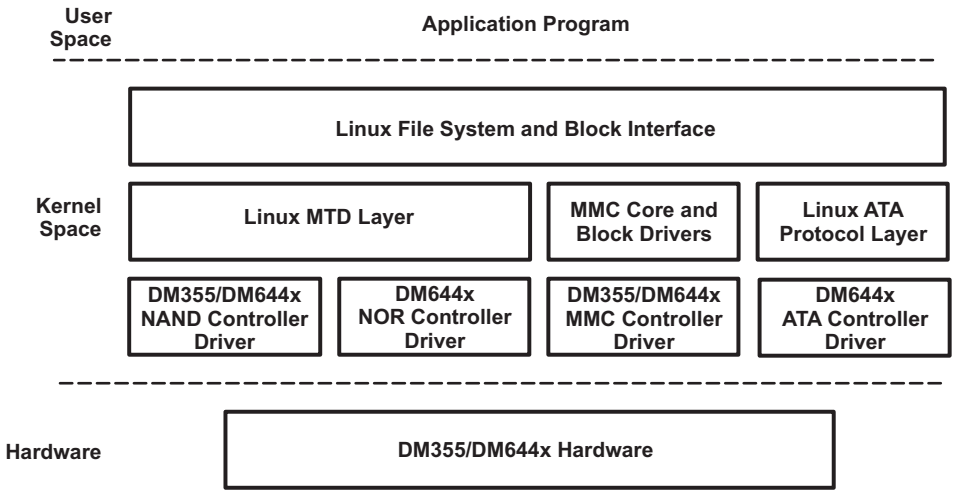

Figure 2-87. Linux Kernel NOR Driver Architecture

The NOR Driver is not applicable to the DM355, because the DM355 EVM does not have NOR support.

On the DM6446, the NOR Driver supports following features:

- Supports 16MB AMD AM29LV256M NOR Flash and Intel I128P30T StrataFlash NOR Flash chips.
- Supports Read/Write/Erase from/to NOR Flash
- Supports 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 None

Constraints The rmmod command to unlink the NOR module is not supported. If there is more than one partition, it fails with OOPS

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

Supported IOCTLs See Table 2-129

Table 2-129. NOR Supported 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

2.8.1 Performance and Benchmarks

2.8.1.1 DM6446 EVM

Table 2-130 shows the memory requirements for the NOR Driver modules built for DM6446.

Table 2-130. NOR Driver Memory Statistics - DM6446

Configuration	Memory Statistics (in Bytes)			
	Program Memory	Data Memory		Total
		Initialized	Uninitialized	
amd_flash.ko	15440	280	0	15720
davinci-nor-flash.ko	2676	416	0	3092

2.8.1.1.1 NOR Driver - Read/Write Performance: Test Setup

- **Hardware used:** DM6446 Platform, AM29LV256M Flash (size: 32MB), Intel 128P30T Flash (size: 16M)
- **Root Filesystem used:** NFS
- **Test Filesystem used on the NOR Flash:** JFFS2 (/dev/mtdblock3)
- **Mode tested:** Low Latency Desktop, Server and Real-Time preemption modes
- **Test Commands:**
  - test_perf_write.sh

```
#!/bin/sh
if [ $# -ne 3 ]; then
       echo "Usage: $0 <mount dir> <buffer-size> <count>" 1>&2
exit 127
fi
dd if=/dev/zero of=$1/test.out bs=$2 count=$3
umount $1
NOTE:  This is to flush file system cache
```

  - test_perf_read.sh

```
#!/bin/sh
if [ $# -ne 3 ]; then
       echo "Usage: $0 <mount dir> <buffer-size> <count>" 1>&2
exit 127
fi
dd if=$1/test.out of=/dev/null bs=$2 count=$3
umount $1
```

- **Test sequence:**
  1. mkdir /mnt/nand
  2. mount –t jffs2 /dev/mtdblock3 /mnt/nand
  3. time ./test_perf_write.sh /mnt/nand <buffer-size> <count>
  4. mount –t jffs2 /dev/mtdblock3 /mnt/nand
  5. time ./test_perf_read.sh /mnt/nand <buffer-size> <count>
  6. mount –t jffs2 /dev/mtdblock3 /mnt/nand
  7. rm –f /mnt/nand/test.out
  8. umount /mnt/nand
  9. Repeat the steps 2 – 8, for each of the different buffer sizes – 4K, 64K, 128K, 256K, 512K and 1024K
  10. For each iteration, the count is also calculated for a constant file size (count * buffer-size) of 16M

2.8.1.1.2 NOR Driver - AMD Write Performance: Low Latency Desktop Mode

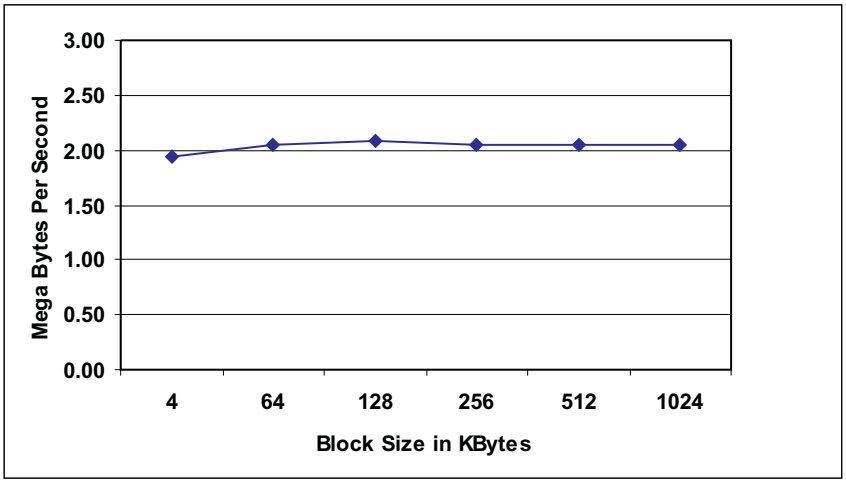

Figure 2-88. NOR Driver - AMD Write Performance Results (LLD) - DM6446

Table 2-131. NOR Driver - AMD Write Performance Data (LLD) - DM6446

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	16	8.61	1.95
	64	16	8.22	2.04
	128	16	8.05	2.08
	256	16	8.16	2.06
	512	16	8.22	2.04
	1024	16	8.19	2.05

Test Setup:

- See Section 2.8.1.1.1, NOR Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.8.1.1.3 NOR Driver - AMD Read Performance: Low Latency Desktop Mode

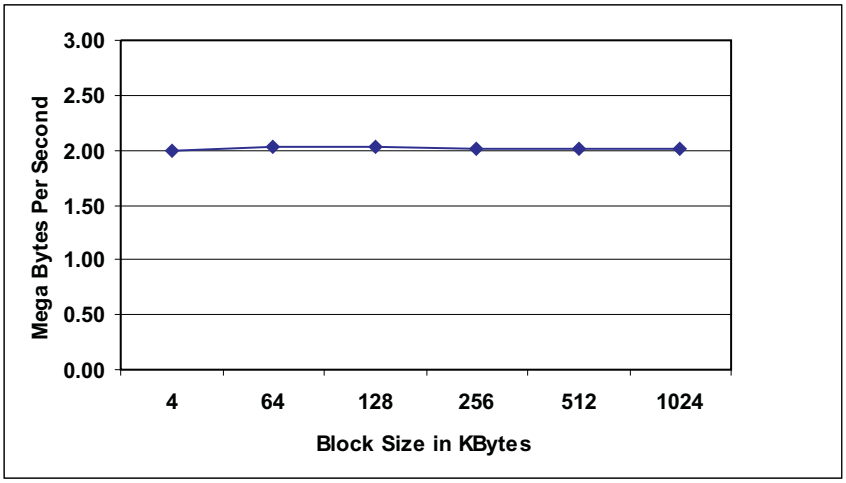

Figure 2-89. NOR Driver - AMD Read Performance Results (LLD) – DM6446

Table 2-132. NOR Driver - AMD Read Performance Data (LLD) - DM6446

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	16	8.43	1.99
	64	16	8.29	2.02
	128	16	8.28	2.03
	256	16	8.37	2.00
	512	16	8.34	2.01
	1024	16	8.35	2.01

Test Setup:

- See Section 2.8.1.1.1, NOR Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.8.1.1.4 NOR Driver - AMD Write Performance: Server Mode

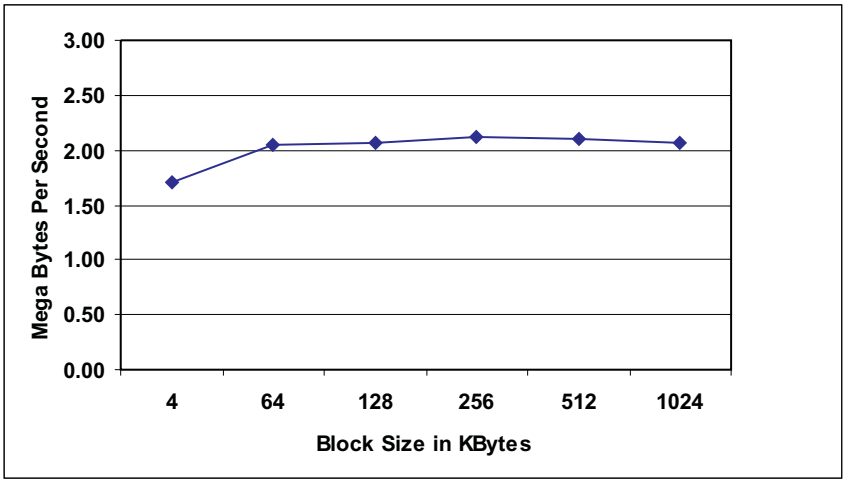

Figure 2-90. NOR Driver - AMD Write Performance Results (Server) – DM6446

Table 2-133. NOR Driver - AMD Write Performance Data (Server) - DM6446

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	16	9.79	1.71
	64	16	8.17	2.05
	128	16	8.09	2.07
	256	16	7.94	2.11
	512	16	7.98	2.10
	1024	16	8.12	2.07

Test Setup:

- See Section 2.8.1.1.1, NOR Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Server preemption mode.

2.8.1.1.5 NOR Driver - AMD Read Performance: Server Mode

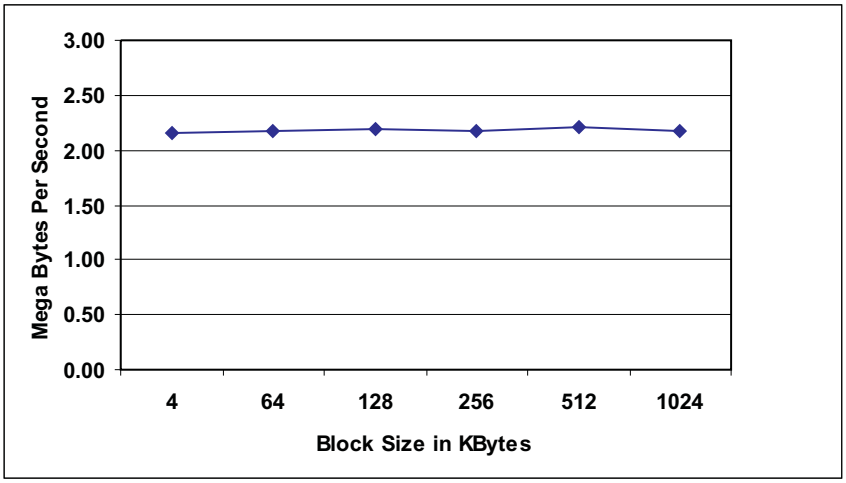

Figure 2-91. NOR Driver - AMD Read Performance Results (Server) – DM6446

Table 2-134. NOR Driver - AMD Read Performance Data (Server) - DM6446

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	16	7.80	2.15
	64	16	7.73	2.17
	128	16	7.63	2.20
	256	16	7.72	2.17
	512	16	7.61	2.20
	1024	16	7.72	2.17

Test Setup:

- See Section 2.8.1.1.1, NOR Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Server preemption mode.

2.8.1.1.6 NOR Driver - AMD Write Performance: Real-Time Preemption Mode

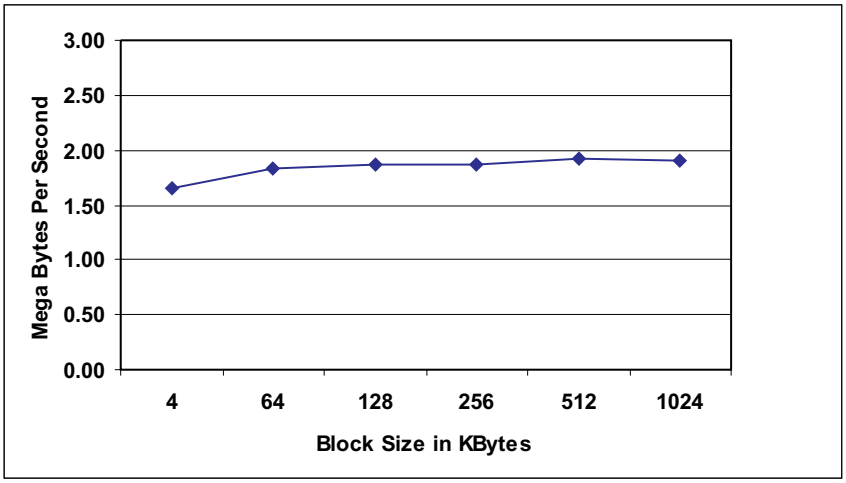

Figure 2-92. NOR Driver - AMD Write Performance Results (Real-Time) – DM6446

Table 2-135. NOR Driver - AMD Write Performance Data (Real-Time) - DM6446

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	16	10.18	1.65
	64	16	9.19	1.83
	128	16	8.97	1.87
	256	16	8.98	1.87
	512	16	8.75	1.92
	1024	16	8.85	1.90

Test Setup:

- See Section 2.8.1.1.1, NOR Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for real-time preemption mode.

2.8.1.1.7 NOR Driver - AMD Read Performance: Real-Time Preemption Mode

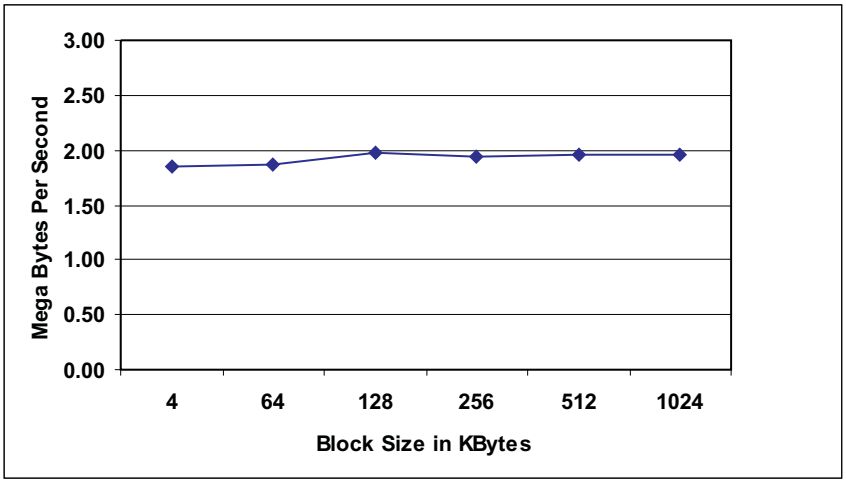

Figure 2-93. NOR Driver - AMD Read Performance Results (Real-Time) – DM6446

Table 2-136. NOR Driver - AMD Read Performance Data (Real-Time) - DM6446

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	16	9.05	1.85
	64	16	8.96	1.87
	128	16	8.50	1.97
	256	16	8.62	1.95
	512	16	8.60	1.95
	1024	16	8.56	1.96

Test Setup:

- See Section 2.8.1.1.1, NOR Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Real-Time preemption mode.

2.8.1.1.8 NOR Driver - Intel Write Performance: Low Latency Desktop Mode

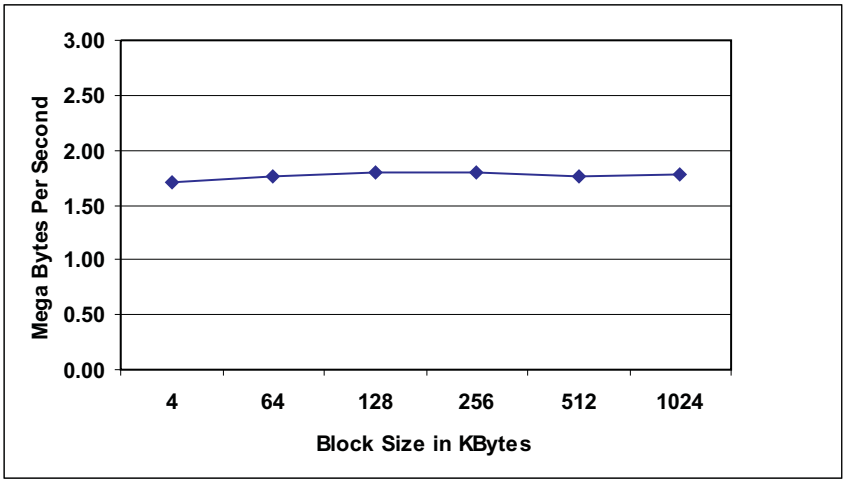

Figure 2-94. NOR Driver - Intel Write (LLD)

Table 2-137. NOR Driver - Intel Write (LLD)

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	16	9.78	1.72
	64	16	9.50	1.77
	128	16	9.35	1.79
	256	16	9.37	1.79
	512	16	9.56	1.75
	1024	16	9.40	1.78

Test Setup:

- See Section 2.8.1.1.1, NOR Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.8.1.1.9 NOR Driver - Intel Read Performance: Low Latency Desktop Mode

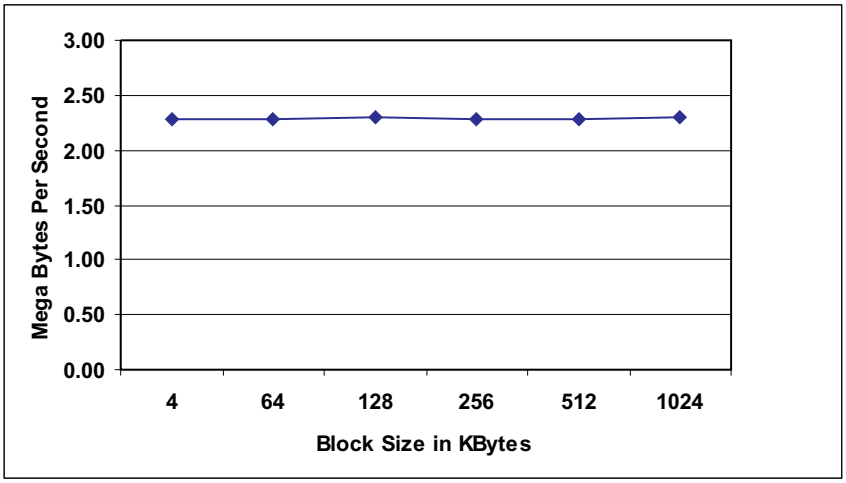

Figure 2-95. NOR Driver - Intel Read (LLD)

Table 2-138. NOR Driver - Intel Read (LLD)

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	16	7.38	2.27
	64	16	7.33	2.29
	128	16	7.31	2.30
	256	16	7.34	2.29
	512	16	7.34	2.29
	1024	16	7.31	2.30

Test Setup:

- See Section 2.8.1.1.1, NOR Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.8.1.1.10 NOR Driver - Intel Write Performance: Server Mode

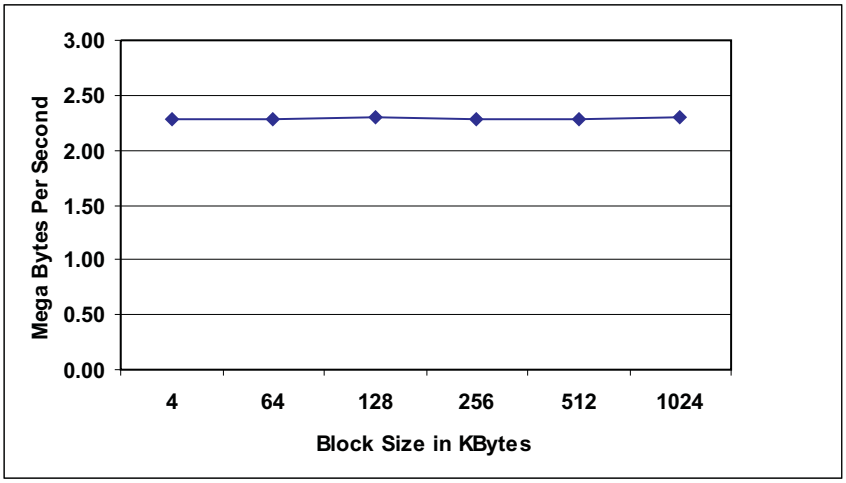

Figure 2-96. NOR Driver - Intel Read (LLD)

Table 2-139. NOR Driver - Intel Read (LLD)

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	16	7.38	2.27
	64	16	7.33	2.29
	128	16	7.31	2.30
	256	16	7.34	2.29
	512	16	7.34	2.29
	1024	16	7.31	2.30

Test Setup:

- See Section 2.8.1.1.1, NOR Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Server preemption mode.

2.8.1.1.11 NOR Driver - Intel Write Performance: Server Mode

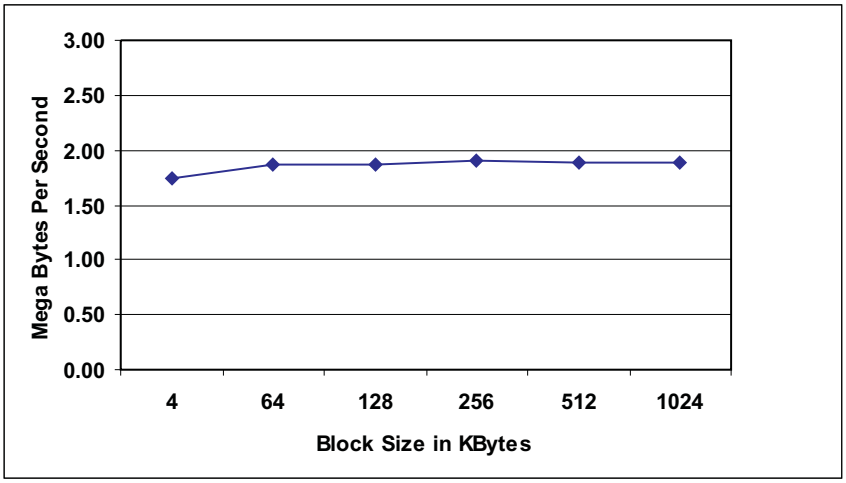

Figure 2-97. NOR Driver - Intel Write (Server)

Table 2-140. NOR Driver - Intel Write (Server)

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	16	9.60	1.75
	64	16	8.99	1.87
	128	16	8.94	1.88
	256	16	8.84	1.90
	512	16	8.89	1.89
	1024	16	8.88	1.89

Test Setup:

- See Section 2.8.1.1.1, NOR Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Server preemption mode.

2.8.1.1.12 NOR Driver - Intel Read Performance: Server Mode

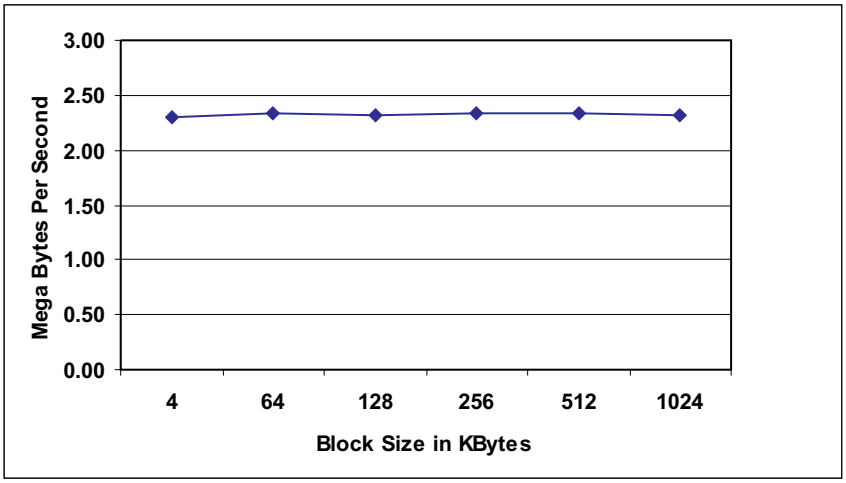

Figure 2-98. NOR Driver - Intel Read (Server)

Table 2-141. NOR Driver - Intel Read (Server)

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	16	7.29	2.30
	64	16	7.20	2.33
	128	16	7.23	2.32
	256	16	7.17	2.34
	512	16	7.17	2.34
	1024	16	7.23	2.32

Test Setup:

- See Section 2.8.1.1.1, NOR Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Server preemption mode.

2.8.1.1.13 NOR Driver - Intel Write Performance: Real-Time Preemption Mode

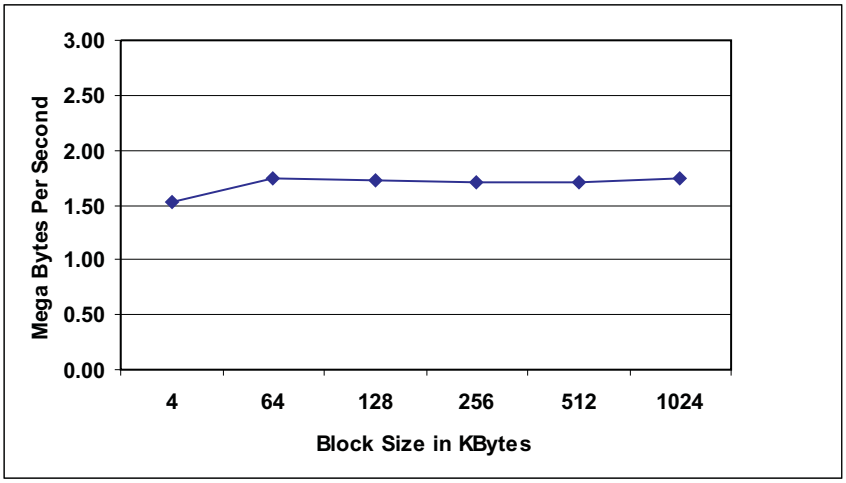

Figure 2-99. NOR Driver - Intel Write (RT)

Table 2-142. NOR Driver - Intel Write (RT)

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	16	10.99	1.53
	64	16	9.66	1.74
	128	16	9.75	1.72
	256	16	9.78	1.72
	512	16	9.78	1.72
	1024	16	9.67	1.73

Test Setup:

- See Section 2.8.1.1.1, NOR Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Real-Time preemption mode.

2.8.1.1.14 NOR Driver - Intel Read Performance: Real-Time Preemption Mode

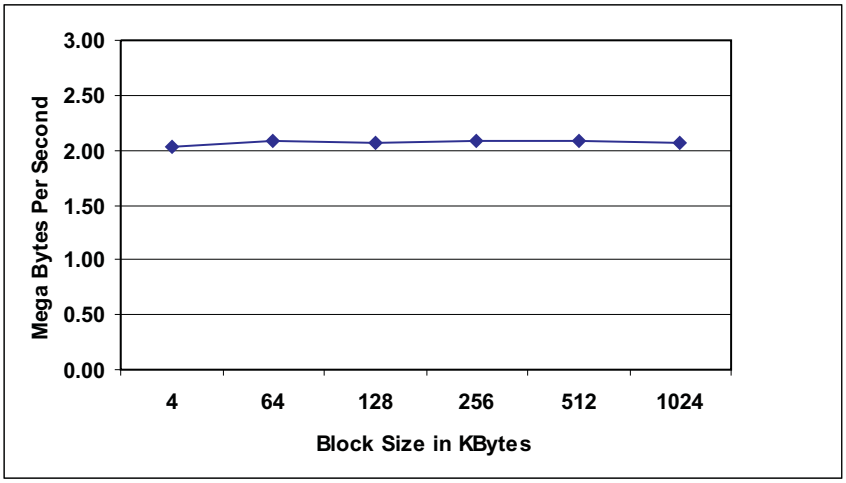

Figure 2-100. NOR Driver - Intel Read (RT)

Table 2-143. NOR Driver - Intel Read (RT)

Test Setup Information	Block Size in KBytes	FileSize in MB	Duration (Seconds)	Mega Bytes/Sec
ARM Frequncy = 297 MHZ	4	16	8.23	2.04
	64	16	8.02	2.09
	128	16	8.11	2.07
	256	16	8.03	2.09
	512	16	8.03	2.09
	1024	16	8.13	2.06

Test Setup:

- See Section 2.8.1.1.1, NOR Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Real-Time preemption mode.

2.9 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.

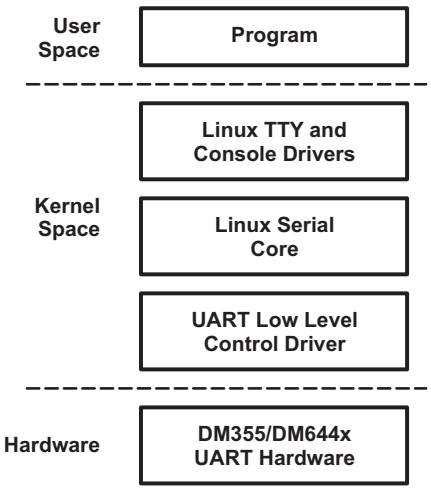

Figure 2-101. Linux Kernel UART Driver Architecture

2.9.1 Driver Features

The UART Driver supports following features:

- Supports UART0
- Supports UART1 (on the DM355 EVM only)

2.9.2 Support and Constraints

Features Not Supported	None
Constraints	None
Supported System Calls	open(), close(), read(),write(), ioctl()
Supported IOCTLs	See Table 2-144

Table 2-144. 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.)

2.10 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.

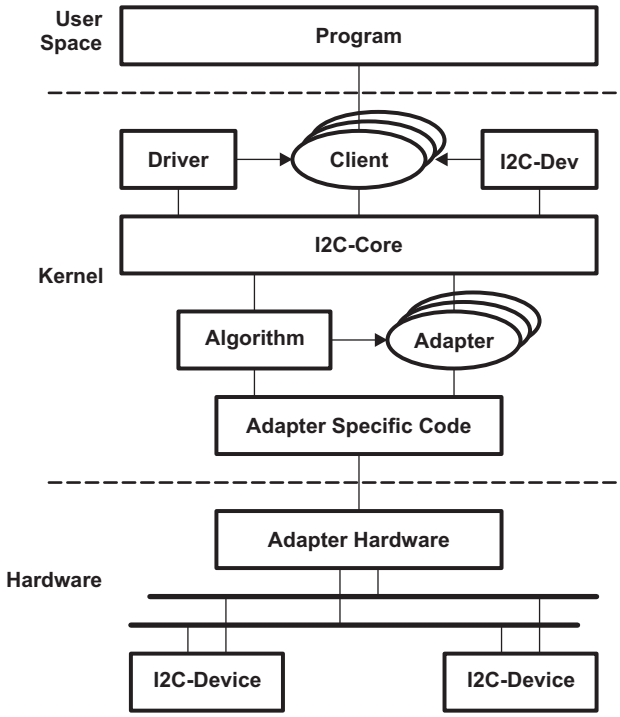

Figure 2-102. Linux Kernel I2C Driver Architecture

The I2C Driver supports the following features:

- 7-Bit Addressing
- Interrupt mode of operation

2.10.1 Support and Constraints

Features Not Supported

The I2C Driver does not support following features:

- 10-Bit Addressing is not supported (It is experimental in Linux).
- DMA mode is not supported.

Constraints

None

Supported System Calls

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

Supported IOCTLs

The I2C Driver supports the IOCTLs listed in Table 2-145.

Table 2-145. 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 = 1for 10 bits.)
I2C_FUNCS	Get the adapter functionality.
I2C_RDWR	Combined R/W transfer (one stop only).

2.11 SPI Driver

SPI-EEPROM is accessible via the serial peripheral interface (SPI). SPI-EEPROM driver is implemented as block driver and compliant with standard MTD driver. It supports various EEPROM devices. The SPI-EEPROM driver creates device node for user space access (e.g.: /dev/mtd3). The SPI control driver supports serial communication with other SPI devices in a 3-bin or 4-bin mode. It supports multi-chip-select operation for up to two SPI slave devices and operates as a master SPI device only.

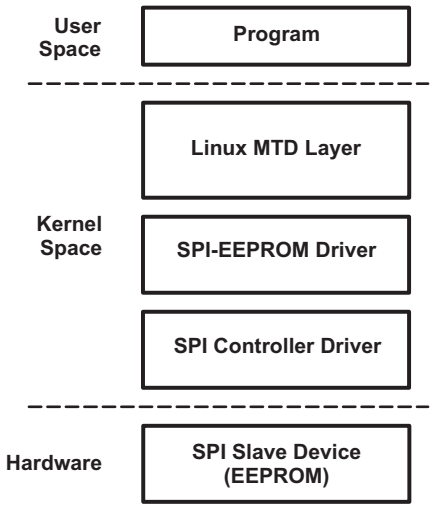

Figure 2-103. Linux Kernel SPI Driver Architecture

The DM355 EVM SPI Driver supports the following features:

- SPI Controller Driver controls the SPI master device for the serial communication to the SPI slave devices
- The SPI EEPROM driver controls SPI EEPROM device. SPI EEPROM is configured as a MTD device. SPI EEPROM driver can be accessed from the user space as /dev/mtd4
- Supports both Interrupt mode and Polling (default) modes of operation

The SPI Driver is not tested on the DM6446 EVM.

2.11.1 Support and Constraints

Features Not Supported	none
Constraints	none
Supported System Calls	write() and read()
Supported IOCTLs	none

2.11.2 DM355 Performance and Benchmarks

Table 2-146 shows the memory requirements for the SPI Driver modules built for DM355.

Table 2-146. SPI Driver Memory Statistics - DM355

Configuration	Memory Statistics (in Bytes)			
	Program Memory	Data Memory		Total
		Initialized	Uninitialized	
davinci_spi_master.ko	7892	380	56	8328
davinci_spi_bitbang.ko	3524	256	0	3780

2.11.2.1 SPI Driver - Read/Write Performance: Test Setup

- **Hardware used:** DM355 Platform, ATMEL25640A (size: 8K)
- **Root Filesystem used:** YAFFS2 (/dev/mtdblock3)
- **Test Filesystem used on the ATMEL EEPROM:** /dev/mtdblock5, /dev/mtd5, Raw Read/Write
- **Mode tested:** Low Latency Desktop, Server and Real-Time preemption modes
- **Test Commands:**
  - The performance is measured using gettimeofday() function
  - The SPI Driver is in the default mode of operation – Polling

2.11.2.2 SPI Driver- Read/Write Performance: Low Latency Desktop Mode

Table 2-147. SPI Driver - Read/Write Performance Data (Low Latency Desktop) - DM355

Test Setup Information	Data Size in KBytes	Write Rate in KBytes/sec	Read Rate in KBytes/sec
ARM Frequency = 216 MHz DDR Frequency = 171 MHz	8	9.5	74.6

Test Setup:

- See Section 2.11.2.1, SPI Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Low Latency Desktop preemption mode.

2.11.2.3 SPI Driver- Read/Write Performance: Server Mode

Table 2-148. SPI Driver - Read/Write Performance Data (Server) - DM355

Test Setup Information	Data Size in KBytes	Write Rate in KBytes/sec	Read Rate in KBytes/sec
ARM Frequency = 216 MHz DDR Frequency = 171 MHz	8	9.6	77.4

Test Setup:

- See Section 2.11.2.1, SPI Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Server preemption mode.

2.11.2.4 SPI Driver- Read/Write Performance: Real-Time Preemption Mode

Table 2-149. SPI Driver - Read/Write Performance Data (Real-Time) - DM355

Test Setup Information	Data Size in KBytes	Write Rate in KBytes/sec	Read Rate in KBytes/sec
ARM Frequency = 216 MHz DDR Frequency = 171 MHz	8	8.7	52.2

Test Setup:

- See Section 2.11.2.1, SPI Driver - Read/Write Performance: Test Setup.
- Rebuild the Linux Kernel for Real-Time preemption mode.

2.12 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 following features:

- The PWM driver is implemented as a char driver.
- Supports four instances of PWM on the DM355 EVM - 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

2.12.1 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() and ioctl()
Supported IOCTLs	See Table 2-150

Table 2-150. 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

2.13 Watchdog Driver

The Watchdog Timer is required in the system to prevent it from lockup in case of the software being trapped in loops with no controlled exit. Both DM355 and DM6446 EVMs use 64-Bit Timer 2, as user-configurable 64-bit watchdog timer.

Driver Features	The WDT driver is implemented as a char driver.
Features Not Supported	None
Constraints	None
Supported System Calls	open(), close(), write(), read() and ioctl()
Supported IOCTLs	See Table 2-151
Performance and Benchmarks	N/A

Table 2-151. WDT Driver Supported IOCTLs

Constant	Description
WDIOC_GETSUPPORT	Get Watchdog information like identity, options and firmware_version
WDIOC_KEEPALIVE	Keep pinging to make UUT keep alive after WDT is opened
WDIOC_GETTIMEOUT	Get timeout value
WDIOC_GETSTATUS	Get current status of Watchdog Timer

2.14 GPIO Module

The GPIO module connects to external devices. While it is possible that the firmware implements some standard connectivity protocol over GPIO (I2C for example), the GPIO module itself is not required to compliant with any such standards.

The GPIO driver exposes only the kernel level API's. This driver is used as a utility by other drivers for configuring the general purpose input/output pins.

All of the features are supported.

The driver supports following features (Kernel APIs):

- Set input/output direction
- Set GPIO values for GPIOs in the output direction
- Get GPIO values for GPIOs in the input direction
- Interrupt support: To configure GPIOs for interrupts.

2.15 EDMA

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.

The 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

3 U-Boot Overview

The U-Boot bootloader in this release is based on the open-source version U-Boot 1.2.0. U-Boot 1.2.0 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, along with NAND UBL. At power reset, NAND UBL executes, which is responsible for loading the U-Boot from NAND to DDR2 memory. U-Boot then executes from DDR2.

U-Boot for NOR (only on DM6446): In this case, U-Boot resides in NOR flash memory. At power reset, U-Boot copies itself into DDR2 memory, and executes from DDR2.

3.1 Functions

The U-Boot is responsible for the following functions:

- Initializes the DM355 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 DM6446 Board HD Boot	On DM355 board NAND Boot	DM6446/DM355 Boards NFS Boot
U-Boot Bootloader	Flash	NAND	Flash
Linux Kernel	Flash	NAND	Tools PC
File System	DVEVM Hard Drive	NAND - YAFFS2	Tools PC

3.2 Features

The following is a list of features supported by U-Boot 1.2.0 for DM355 and DaVinci DM6446:

- Default boot arguments are configured for the *initrd* image
- The NAND bootloader initializes PLL1 at 594 MHz on DaVinci DM6446 EVM, while it initializes PLL1 at 216 MHz on DM355 EVM. In this mode, the user boot loader (UBL) gets booted by the ROM boot loader (RBL), which loads the U-Boot NAND bootloader
- NOR boot mode initializes the PLL1 to 594 MHz on DaVinci DM6446 EVM
- Supports switching of video standard on DVEVM board.
  - DM355 EVM: Set connector in J1: 2-3 = PAL, 1-2 = NTSC
  - DM6446 EVM: Set USER switch (10th switch) in S3: On = PAL, OFF = NTSC
  - The Bootloader sets the videostd environment variable that can be fed to bootargs.

3.3 U-Boot Commands

Table 3-2 provides details on the commands supported by U-Boot 1.2.0.

Table 3-2. U-Boot Command Set

Command	Behavior	Example
askenv	Get environment variables from stdin	askenv serverip Please enter 'serverip':192.168.1.1
autoscr	Run script from memory	autoscr 80700000 ## Executing script at 80700000
base	Print or set address offset for memory commands	base 80700000 Base Address: 0x80700000
bdinfo	Print Board Info structure	

Table 3-2. U-Boot Command Set (continued)

Command	Behavior	Example
boot	Boot default, i.e., run 'bootcmd'	
bootd	Boot default, i.e., run 'bootcmd'	
bootm	Boot application image from memory	bootm 80700000
bootp	Boot image via network using BootP/TFTP protocol	bootp 80700000 uImage
cmp	Memory compare	cmp.l 80700000 82000000 16
coninfo	Print console device and information	
cp	Memory copy	cp.l 80700000 82000000 16
crc32	Checksum calculation	crc32 80700000 100 82000000
dhcp	Invoke DHCP client to obtain IP/boot params	
erase	Erase FLASH memory (on DM6446)	
flinfo	Print FLASH memory information (on DM6446)	
go	Start application at address addr	go 80700000 ## Starting application at 0x80700000 ...
help	Print online help	
iminfo	Print header information for application image	
imls	List all images found in flash (on DM6446)	
loadb	Load binary file over serial line (kermit mode)	loadb 80700000
loads	Load S-Record file over serial line	Loads 80700000
loop	Infinite loop on address range	loop.l 80700000 1
md	Memory display	md.l 80700000 16
mm	Memory modify (auto-incrementing)	mm.l 80700000 80700000: bbbbbbbb ? cccccccc 80700004: aaaaaaaa ?
mtest	Simple RAM test	mtest 80700000 80700100 ff
mw	Memory write (fill)	mw.l 80700000 aaaaaaaa 100
nm	Memory modify (constant address)	nm.l 80700000 80700000: aaaaaaaa ? bbbbbbbb 80700000: bbbbbbbb ?
ping	Send ICMP ECHO_REQUEST to network host	ping 192.168.1.100 host 192.168.1.100 is alive
printenv	Print environment variables	
protect	Enable or disable FLASH write protection (on DM6446)	
rarpboot	Boot image via network using RARP/TFTP protocol	rarpboot 80700000 uImage
reset	Perform RESET of the CPU	
run	Run commands in an environment variable	
saveenv	Save environment variables to persistent storage	
setenv	Set environment variables	setenv serverip 192.168.1.100
sleep	Delay execution for some time	
tftpboot	Boot image via network using TFTP protocol	tftp 80700000 uImage
version	Print U-Boot version	

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
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Clocks and Timers	www.ti.com/clocks	Digital Control	www.ti.com/digitalcontrol
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Telephony	www.ti.com/telephony
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

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