DaVinci™ Linux Drivers Datasheet

Platform Support Group (PSG)

ABSTRACT

The TI DaVinci™ Linux Drivers are incorporated in the MontaVista Linux Support Package (LSP) provided with MontaVista Linux Pro v4.0. The LSP consists of optimized multimedia peripheral device drivers for the TMS320DM644x Digital Media System-on-Chip (DMSoC), integrated with the MontaVista Linux 2.6.10 kernel to run on the ARM926 core. The drivers provide for rapid software development on the DM6446 EVM and are provided in source form to facilitate portability to production hardware platforms. This document provides an overview and performance data for each of the drivers.

Contents

1 Texas Instruments DaVinci Linux Drivers

1.1 Drivers Summary

The table below summarizes Linux driver support in the MontaVista Linux 2.6.10 LSP with PatchLevel 45 from TI (https://www-a.ti.com/downloads/sds_support/targetcontent/psp/index.html):

Table 1. DaVinci Peripheral Driver Support

Peripheral	Linux Driver Type	Use Case	See Section
Video Port Back End (VPBE) Display	FBDEV	NTSC or PAL Video display up to D1 resolution	3.1.2
Video Port Back End (VPBE) OSD graphics	FBDEV	Graphics on-screen display with alphablending	3.1.2
Video Port Front End (VPFE) Capture	V4L2	NTSC or PAL Video capture from TVP5146 Decoder.	3.1.3
Video Port Front End (VPFE) Capture – CCDC	V4L2	Video capture from CCD Controller	3.1.3
VPFE Previewer		Convert Bayer input to YCbCr format,	3.1.3.3
VPFE Resizer		Input image upscaling and downscaling.	3.1.3.4
VPFE Auto Exposure/Auto White Balance (AEW)		Performs auto exposure and auto white balance on CCDC input data.	3.1.3.5
VPFE Auto Focus (AF)		Performs auto focus on CCDC input.	3.1.3.6
Audio Serial Port (ASP)	OSS	Audio record and playback from AIC33	3.2
Ethernet *(10/100Mbps)*		Transmit/receive network data.	3.3
USB 2.0 OTG	Inventra USB driver stack	USB modes: USB full and high speed., Host, device and On the Go (OTG) roles. PIO and DMA modes supported.	3.4
USB 2.0 Mass Storage Class – NAND and HDD		USB pen drive or USB hard drive for data storage	3.4
USB 2.0 Hub		USB hubs for expansion of USB port	
USB 2.0 HID Class		USB human interface device such as keyboard or mouse.	3.4
USB Isochronous Class		USB ISO-Chronous Device driver. Supports Video and Audio Isochronous devices	
USB Network Class (CDC Slave)		USB Communication Device Class Slave Driver	
USB Network Class (RNDIS Slave)		USB Remote Network Driver Interface Specification.	
UART		Serial communication.	3.5
I2C		Interface to MSP430 for IR, RTC, bit I/O. Access to video & audio codec registers.	3.6
ATA / CF		Interface to IDE harddisk and compact flash storage media.	3.7
Asynchronous EMIF		External flash memory (NAND/NOR). (No support for big block, small block, SLC, MLC.)	3.8, 3.9
Multimedia Card (MMC) / Secure Digital (SD)		Interface to external flash cards. Both MMC and SD cards	3.10
PWM		Pulse Width Modulation waveform typically used for motor control.	3.11
Timers (Configured as 4 x 32bit)	MV High Resolution Timer	Linux tick. Also, MontaVista High Resolution Timer (HRT)	
Watchdog timer (64-bit)		Monitor runaway system condition.	
GPIO		Data I/O, and produce CPU interrupts and EDMA events.	

1.2 Limitations Summary

The table below summarizes the known feature limitations in the MontaVista Linux 2.6.10 LSP with PatchLevel 45 applied :

Table 2. Driver Limitations and DaVinci Peripherals Not Supported with Drivers

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).
SPI	No driver; no SPI peripher al on EVM.	Serial communication with other SPI devices.
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 .

1.3 Features

- The TI DaVinci Linux Drivers package supports the following device drivers:
    - Serial (UART, I2C)
    - Communication (CPMAC, USB (Target), USB (Host), USB (OTG))
    - Block (ATA, MMC, SD, NAND, NOR)
    - Audio Open Sound System (OSS) (ASP)
    - Video Processing Front End (VPFE)
        - CCD Controller Driver (V4L2)
        - Resizer
        - Preview Engine
        - Auto Exposure/Auto White Balance (AEW)
        - Auto Focus (AF)
    - Video Processing Back End (VPBE)
        - Video Display (FBDEV)
    - Watchdog Timer
    - GPIO
    - Pinmux
    - PWM
- Kernel API support for ASP and EDMA3
- U-Boot bootloader
- Supports JFFS2 filesystem for flash media drivers.
- Compiled with MontaVista Professional 4.0 arm_v5t_le toolchain, for Linux Kernel version 2.6.10.
- GNU/Linux ARM9 Application Binary Interface (ABI)
- For development on DaVinci EVM platform.
- Portable to customer DaVinci-based hardware platforms.
- Supported devices:
    - DM6446
    - DM6443
    - DM6441

1.4 Linux Base Port Overview

1.4.1 Toolchain and Version Information

The kernel and MontaVista Linux (MVL) user space application are compiled and linked with the MontaVista Professional 4.0 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) 3.2.0

- GNU libc (glibc) 2.3.3
- GNU Debugger (GDB) 6.3
- Binutils 2.15.94

1.4.2 Tested Modes

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

- No Forced Preemption (Server)
- Voluntary Kernel Preemption (Desktop)
- Preemptible Kernel (Low Latency Desktop)
- Complete Preemption (Real-Time)

1.4.3 Timers

There are three software-programmable general purpose timers on DaVinci. Two of the 64-bit timers are configured as four independent 32-bit timers as follows:

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

1.5 Version Information

This document applies to a driver set whose version is PatchLevel 45.

1.6 Documentation Support

The following documents are included in the /docs folder in the driver package:

Document Name	Contents
ApplicationNotes.pdf	Explains video applications usage and sample output.
AEWUserManual.pdf	Auto Expose/Auto White Balance (AEW) Driver User Manual
AFUserManual.pdf	Auto Focus (AF) Driver User Manual
CCDCUserManual.pdf	CCDC Driver User Manual
NOR_Flash_Driver_UsersManual.pdf	NOR Flash Driver User Manual
PreviewerUsersManu al.pdf	Previewer Driver User Manual
ResizerUsersManual.pdf	Resizer Driver User Manual
VPBEUsersManual.pdf	Video Processing Back End (VPBE) Driver User Manual
DaVinci_Custom_SDIO_Stack_UserManua l.pdf	SDIO User Manual

In addition, video applications and the NOR flash driver application are included in the /examples folder.

2 U-Boot Overview

2.1 Description

The U-Boot bootloader is executed at reset and loads the O/S kernel into DDR2 memory. U-Boot resides in flash memory, copies itself to DDR, and executes from DDR when the DaVinci EVM is powered on.

U-Boot performs the following functions:

- Initializes the DaVinci EVM hardware
- Places DSP in RESET
- Provides boot parameters to the Linux kernel
- Starts the Linux kernel
- Uploads new binary images to memory via serial port or Ethernet

	On New DVEVM Board "HD Boot"	After User Setup "NFS Boot"
U-Boot BootLoader	Flash	Flash
Linux Kernel	Flash	Tools PC
File System	DVEVM Hard Drive	Tools PC

Figure 1. U-Boot and Kernel Memory Locations

2.2 Features

- Default boot arguments are configured for the *initrd* image.
- The NAND bootloader initializes PLL1 at 594 MHz. 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.
- Supports switching of video standard on DVEVM board.
  - Set USER switch (10th switch) in S3: On = PAL, OFF = NTSC

2.3 Known Issues

- An EVM power cycle is required after environmental variables are set.
- *loadb* command fails with TeraTerm application.
- TFTP to NOR flash is not supported.

- Includes workaround for Switch Central Resource (SCR) lockup.
- Includes an Ethernet modification that sets a bit in the phy to boost signal strength. This is phy-specific.

2.4 Building the U-Boot ELF File

To build the U-Boot ELF file, which can be loaded from Code Composer Studio (CCS), edit the board/davinci/lowlevel_init.S file as follows:

```
_TEXT_BASE:
        .word   TEXT_BASE     /* SDRAM load addr from config.mk */

.global reset_cpu
reset_cpu:
        bl reset_processor

.globl lowlevel_init
lowlevel_init:
        /* mov     pc,     lr  */
```

Uncomment the last line above to prevent low-level initializations that will be done by the CCS GEL file. Build the U-Boot file. The U-Boot file is loadable from CCS.

2.5 Running U-Boot Examples

This section describes file changes required to run the U-Boot examples. Edit the u-boot/examples/Makefile as indicated below.

Change the LOAD_ADDR macro for ARM. This address should be in DDR2 memory.

```
ifeq ($(ARCH),i386)
LOAD_ADDR = 0x40000
endif

ifeq ($(ARCH),arm)
LOAD_ADDR = 0x86000000  ß Edit this line
endif
```

Build U-Boot and the examples:

```
# make distclean clean
# make davinci_config
# make  ß Builds u-boot
# make examples  ß Builds the examples
```

Load the hello_world.bin file in the examples directory on to the EVM, and run it:

```
DaVinci EVM # tftp 86000000 hello_world.bin
TFTP from server 172.24.133.168; our IP address is 172.24.133.16
Filename 'hello_world.bin'.
Load address: 0x86000000
Loading: #
done
Bytes transferred = 499 (1f3 hex)
DaVinci EVM # go 86000000
## Starting application at 0x86000000 ...
Example expects ABI version 2
Actual U-Boot ABI version 2
Hello World
argc = 1
argv[0] = "86000000"
argv[1] = "<NULL>"
Hit any key to exit ...
```

The example load address should match the address specified in the Makefile.

2.6 U-Boot Commands

setenv, saveenv, printenv

Example: setenv ipaddr 192.168.1.50

Common variables to configure the behavior of U-Boot:

Table 3. U-Boot Configuration Variables

Variable	Behavior
autoscr	Run script from memory
base	Print or set address offset
bdinfo	Print Board Info structure
boot	Boot default, i.e., run 'bootcmd'
bootd	Boot default, i.e., run 'bootcmd'
bootm	Boot application image from memory
bootp	Boot image via network using BootP/TFTP pr otocol
cmp	Memory compare
coninfo	Print console devices and information
cp	Memory copy
crc32	Checksum calculation
dhcp	Invoke DHCP client to obtain IP/boot params
echo	Echo args to console
erase	Erase FLASH memory
flinfo	Print FLASH memory informat ion
go	Start application at address 'addr'
help	Print online help
imimfo	Print header information for application image
imls	List all images found in flash
itest	Return true/false on integer compare
loadb	Load binary file over serial line (kermit mod e)
loads	Load S-Record file over serial line
loop	Infinite loop on address range
md	Memory display
mm	Memory modify (auto -incrementing)

mtest	Simple RAM test
mw	Memory write (fill)
nfs	Boot image via network using NFS protocol
nm	Memory modify (con stant address)
ping	Send ICMP ECHO_REQUEST to network host
printenv	Print environment variables
protect	Enable or disable FLASH write protection
rarpboot	Boot image via network using RARP/TFTP protocol
reset	Perform RESET of the CPU
run	Run commands in an environment variable
saveenv	Save environment variables to persistent storage
setenv	Set environment variables
sleep	Delay execution for some time
tftpboot	Boot image via network using TFTP protocol
version	Print U-Boot version

3 Linux Kernel Drivers

This section describes the DaVinci Linux device driver features, known issues and performance data.

3.1 Video Driver

3.1.1 Description

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

The video display driver for the DaVinci Video Processing Back End (VPBE) peripheral is a char driver, compliant with the frame buffer driver (FBDEV). The device nodes created for the VPBE driver are:

- /dev/fb/0 – OSD0 Window
- /dev/fb/1 – Vid0 Window
- /dev/fb/2 – OSD1 Window
- /dev/fb/3 – 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/v4l2/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.

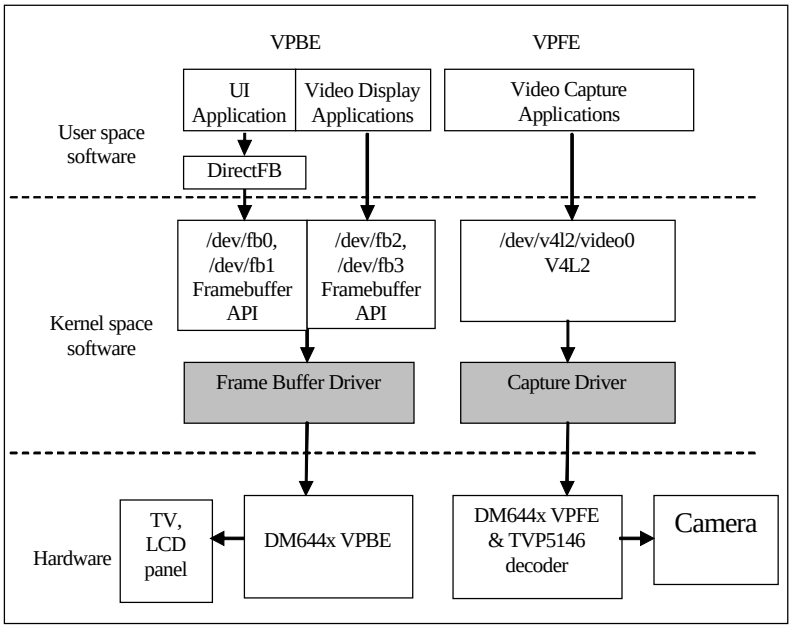


Figure 2. DM644x Video Driver Architecture

3.1.2 Video Display Driver

3.1.2.1 VPBE Driver Features

- Two video windows (VID0 and VID1) in YUV422 format. Both windows are 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.
  - Component video using three DACs.
- NTSC and PAL interlaced outputs using the internal video encoder.
- Flexibility to change window sizes and starting coordinates.
- Zoom feature to zoom the display to 1x, 2x, 4x.
- 480p & 576p formats.
- Digital output on RGB666 digital interface.
- Multiple video and OSD windows.
- RGB888 input mode for the video window.
- Palletized input (1/2/4/8 bit) mode for OSD winow.
- Runtime enable/disable of windows.
- Color-keying.
- Programmed flip-queueing.
- Maps output signals to the RCA jacks on the EVM based on boot time selection, as shown in the table below.

Table 4. 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

- Different combinations of interface, mode, vid0, vid1, osd0 and osd1.
- If the VPBE driver is static, the user must attach the following line to the U-Boot bootargs:

```
video=dm64xxfb:options,
```

where ‘options’ has the format:

```
format=s-video:output=ntsc:vid0=off
```

Other possible options are:

```
format=[s-video
|component|composite|rgb|ycc16|ycc8|prgb|srgb|epson|casio1g |
```

```
udisp|stn]

output= [ntsc|pal|525p|625p|lcd]

vid0=[off]
vid1=[off]
osd0=[off]
osd1=[off]
```

3.1.2.2 VPBE Driver Known Issues

- The driver does **NOT** support analog RBG output.
- The vid0 window cannot be used for interlaced video output.
- Dynamic removal of the module is not allowed when the FB console is in use.
- During dynamic loading of this driver, it sometimes displays the segmentation fault message because of a problem in the fb_show_logo function of fbmem.
- White patches at the top of VID windows are observed when using DVD players as input devices for capture.

3.1.2.3 Supported System Calls

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

3.1.2.4 Supported VPBE 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 bitma p 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_WI N	Sets OSD window 1 as attribute window or bitmap window.
FBIO_GET_BLINK	Gets the existing blinking interval of the attribute window and value of the vpbe_blink_enable flag.
FBIO_SET_BLINK	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_PARAM S	Gets the existing configuration of the bitmap (OSD0 / OSD1) window.
FBIO_SET_BITMAP_CONFIG_PARAM S	Sets the configuration of the BITMAP (OSD0 / OSD1) window.
FBIO_SET_DCLK	Sets the DCLK generated by the LCD c ontroller.
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 modelist using the vmode, xres, yres & 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 backgroun d color. The window should be disabled before using this command.
FBIO_ENABLE_DISPLAY	Enable or disable the display.
FBIOGET_VSCREENINFO	Gets the variable screen information of the framebuffer. This command can be used f or each framebuffer window.
FBIOPUT_VSCREENINFO	Sets variable screen paramete rs for framebuffer, 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 u sing the var_screeninfo offset of the buffer (out of number of buffers for the window: 3 for video & 2 for OSD) passed to the IOCTL. It calculates actual buffer location and sets it into the window register.
FBIO_SET_CURSOR	Configur es cursor parameters.
FBIO_GET_CON2FBMAP	Gets con2fbmap structure.
FBIO_SET_CON2FBMAP	Sets con2fbmap structure.
FBIO_WAITFORVSYNC	Synchronizes the buffered display by waiting for vsync before the call returns.
FBIO_SETATTRIBUTE	Program s the attribute w indow 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 4 x) for the corresponding display plane.

3.1.3 Video Capture Drivers

3.1.3.1 VPFE Driver 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.
- 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). The coordinates can be programmed to pick the image from the center of the capture scene, for example.
- 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.

3.1.3.2 VPFE Driver Known Issues

- The driver does **NOT** support the histogram feature.
- The driver does not support HD input.
- I2C read/write failure.
  - 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.
- The previewer will not work with the video port enabled in the CCDC driver.
- The previewer and H3A drivers cannot be used together.

3.1.3.3 Previewer Driver 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.

3.1.3.4 Resizer Driver Features

- Supports YUV422 color interleaved and 8-bit color separate data input formats.
- Input from CCDC, SDRAM or DDRAM.
- A standalone kernel module that can be used by multiple applications.
- Supports upscaling/downscaling multiplier from ¼ to 4x.

3.1.3.5 Auto Expose/Auto White Balance (AEW) Driver Features

- Supports image in Bayer pattern.

- Input from CCD controller.

3.1.3.6 AF Driver Features

- Supports image in Bayer pattern
- Input from CCD controller.

3.1.3.7 Supported System Calls

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

3.1.3.8 Supported VPFE 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 .

3.1.4 DVEVM Board Features

- The Video Processing Front End (VPFE) consists of the CCD controller (CCDC), preview engine, resizer, hardware 3A (H3A) statistic generator and histogram blocks.
- The Video Processing Back End (VPBE) consists of the on screen display (OSD), the video encoder (VENC) including the digital LCD (DLCD) and analog (i.e. DAC) interfaces.
- One video input port, supports composite or RGB .
- Four video DAC outputs: component, RGB, composite .

3.1.5 Menu Configuration: Video Driver

3.1.5.1 Video Capture (VPFE)

```
Device Drivers  --->
     Multimedia devices  --->
          <*> Video For Linux
          Video For Linux  --->
               --- Video Adapters
               <*> TVP5146 video decoder
               <*> Davinci Video Capture
```

3.1.5.2 Video Display (VPBE)

```
Device Drivers  --->
     Graphics support  --->
          [*] Support for frame buffer devices
          [*] Davinci Framebuffer support
          Console display driver support  --->
          <*> Framebuffer Console support
               Logo configuration  --->
                    [*] Bootup logo
                    [*]   Standard black and white      Linux logo
                    [*]   Standard 16-color Linux logo
                    [*]   Standard 224-color Linux logo
```

3.1.6 Performance and Benchmarks

Table 5. VPFE Memory Statistics

Configuration	Memory Statistics [1]				
	Program Memory	Data Memory			Total
		Initialized	Uninitialized	Stack	
davinci-vpfe.ko	11624	1636	0		13260
tvp5146.ko	3476	256	432		4164
v4l2-common.ko	2096	1336	0		3432
video-buf.ko	13888	280	280		14172
videodev.ko	5108	544	1024		6676

[1] All memory requirements are expressed in bytes for **full-featured** device driver usage.

Table 6. VPBE Memory Statistics

Configuration	Memory Statistics [1]				
	Program Memory	Data Memory			Total
		Initialized	Uninitialized	Stack	
davincifb.o (vpbe)	10032	1080	52		11164

[1] All memory requirements are expressed in bytes for **full-featured** device driver usage.

3.1.7 Notes

- VID0 Window is disabled due to a silicon bug.
- Video drivers are not available as modules.

3.2 Audio Driver

3.2.1 Description

The DaVinci audio driver is a char driver that supports the Audio Serial Port (ASP) peripheral. 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.

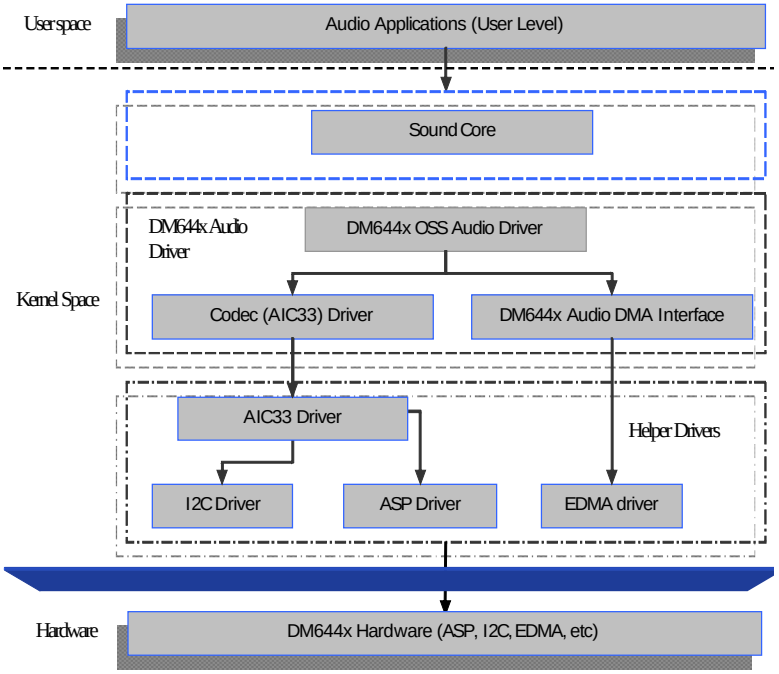


Figure 3. DM644x Audio Driver Architecture

3.2.2 Driver Features

- Open Sound System (OSS) version 1.11 compatible with /dev/sound/dsp and /dev/sound/mixer.
- Record and playback capability.
- Audio channel set to stereo mode by default.
- Supports AFMT_S16_LE (16 bit) audio format only.

3.2.3 Known Issues

- No support for other audio formats.
- Does not support DSP, right-justified, left-justified features.
- Does not support separate volume settings for left and right channels.
- Not available as a loadable module.
- Pausing and jitter noise are observed during playback when an *insmod* command is issued.

3.2.4 Supported System Calls

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

3.2.5 Supported IOCTLs

Constant	Description
OSS_GETVERSION	Identify the driver version .
SNDCTL_DSP_STEREO	Set stereo or mono c hannel.
SNDCTL_DSP_CHANNELS	Set the number of audio channels .
SNDCTL_DSP_SPEED	Set the sampling rate .
SNDCTL_DSP_SETFMT	Select the sample format .
SNDCTL_DSP_GETFMTS	Return a list of natively supported sample formats .
SNDCTL_DSP_GETBLKSIZE	Return the b lock 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 w aits 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_STEREO DEVS	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 .

3.2.6 DVEVM Board Features

- The ASP on the DVEVM is connected to a Texas Instruments TLV320AIC33 stereo codec for input and output of audio signals.
- There are FOUR audio interfaces on the DVEVM board:
  - LINE IN: Capture audio data from LINE

- LINE OUT: Play audio data
- MIC: Capture audio data from microphone
- HP OUT: Play audio data on headphone

3.2.7 Menu Configuration: Sound-OSS Driver.

```
Device Drivers  --->
     Sound  --->
     <*> Sound card support
          Advanced Linux Sound Architecture  --->
          Open Sound System  --->
               <*> Open Sound System (DEPRECATED)
               <*> DaVinci Sound Driver
               <*>   TLV320AIC33 Stereo Codec
```

3.2.8 Performance and Benchmarks

Table 7. Audio Driver Memory Statistics

Modules	Memory Statistics[1]				
	Program Memory	Data Memory			Total
		Initialized	Uninitialized	Stack	
davinci-audio.ko	7680	644	80		8404
davinci-audio-dma-intfc.ko	7128	296	28		7452
davinci-audio-aic33.ko	6484	704	36		7224

[1]All memory requirements are expressed in bytes for **full-featured** device driver usage.

3.3 Ethernet Driver

3.3.1 Driver Features

The Ethernet driver supports the Linux 2.6 NAPI Interface or TI proprietary interrupt pacing mechanism for increased performance. It supports packet processing in Interrupt vs. Tasklet mode for flexibility to suit different system requirements.

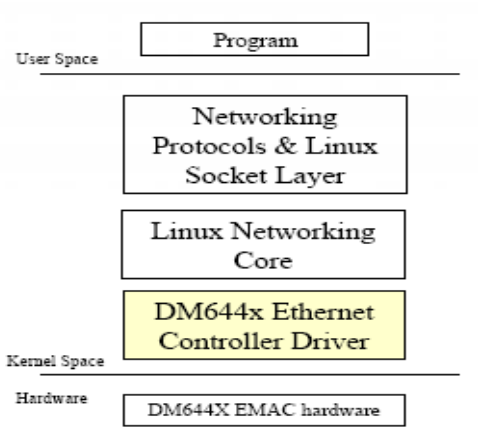


Figure 4. Linux Kernel Ethernet Driver

3.3.2 Constraints

The DaVinci EMAC 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.
- Receive Flow Control using RXnFREEBUFFER and RXnFLOWTHRESH registers

3.3.3 Known Issues

- When USB OTG is enabled on the modified gamma boards (hardware workaround for testing USB OTG), the Ethernet driver doesn't work as both share the same GPIO pins.

3.3.4 DVEVM Board Features

EMAC – 10/100 PHY connectivity

- Allows 802.3 connection to 10/100 PHY.
- EMAC and MDIO interface connects directly to the PHY.

3.3.5 Menu Configuration: Ethernet Driver

```
Device Drivers  --->
```

```
            Networking support  --->
                  [*] Networking support
                        Networking options  --->
                              <*> Packet socket
                              <*> Unix domain sockets
                              [*] Use xfrm policy fwd
                              [*] TCP/IP networking
                              [*]   IP: kernel level autoconfiguration
                              [*]     IP: DHCP support
                              <*> IP: TCP socket monitoring interface
                              <M> The IPv6 protocol
                              [*] Network packet filtering
                                        (replaces ipchains)  --->
                                    [*] Netpoll support for trapping
incoming
                                        packets
                                    [*] Netpoll traffic trapping
                                    [*] Network device support
                                    <M>   Universal TUN/TAP device driver
                                          support
                                          Ethernet (10 or 100Mbit)  --->
                                          [*] Ethernet (10 or 100Mbit)
                                          <*> TI DaVinci EMAC Support
                                    <M>   PPP (point-to-point protocol)
support
                                    <M>   PPP support for async serial ports
                                    <M>   PPP support for sync tty ports
                                    <M>   PPP Deflate compression
                                    <*>   Network console logging support
                                          (EXPERIMENTAL)
```

3.3.6 Performance and Benchmarks

Table 8. EMAC Driver Memory Statistics

Configuration	Memory Statistics [1]				
	Program Memory	Data Memory			Total
		Initialized	Uninitialized	Stack	
davinci_emac_driver.ko	49188	972	308		50468

[1] All memory requirements are expressed in bytes for **full-featured** device driver usage.

3.3.6.1 Performance: Desktop

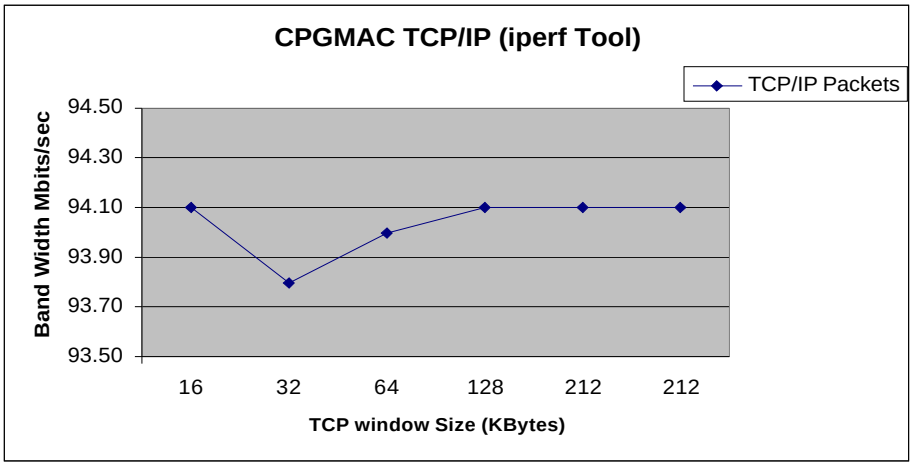


Figure 5. Runtime Performance Characteristics-IPERF_STRAIGHT_CABLE_Desktop

Table 9. Test Data

Test Setup Information	TCP window size in Kbytes	TCP/IP Packets		
		Bandwidth in Mbits/sec	Transfer size in Mbytes	Interval in sec
ARM Frequency = 283 MHZ	16	94.10	673	60
	32	93.80	671.00	60
	64	94.00	672.00	60
	128	94.10	673.00	60
	212	94.10	673.00	60
	212	94.10	673.00	60

3.3.6.2 Comments

- The CPGMAC is configured for auto negotiation, 100Mbps full duplex.
- The “iperf” tool is run on DUT1 in server mode and on DUT2 in client mode.
- Both the DUT and the PC were on the same switch.
- “iperf” version 1.7.0 was used for the tests.
- The data captured here is for “iperf” in client mode.
- The “iperf” tool is run on DUT1 and DUT2 with the NFS mounted.
- The Linux kernel is configured in ‘Desktop Mode’.

3.3.7 Performance:Low Latency

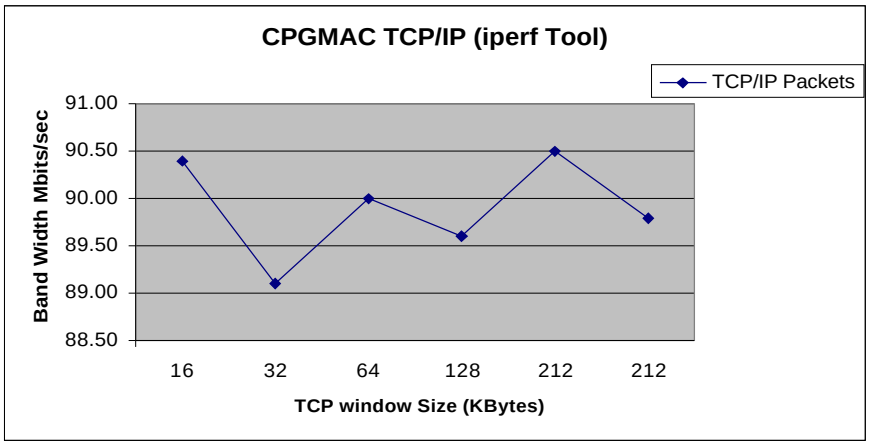


Figure 6. Runtime Performance Characteristics-IPERF_STRAIGHT_CABLE_LowLatency

Table 10. Test Data

Test Setup Information	TCP window size in kBytes	TCP/IP Packets		
		Bandwidth in Mbits/sec	Transfer size in MBytes	Interval in sec
ARM Frequency = 2 83 MHZ	16	90.40	647	60
	32	89.10	637.00	60
	64	90.00	643.00	60
	128	89.60	641.00	60
	212	90.50	647.00	60
	212	89.80	643.00	60

3.3.7.1 Comments

- The CPGMAC is configured in auto negotiation, 100Mbps full duplex.
- The "iperf" tool is run on DUT1 in server mode and on DUT2 in client mode.
- Both the DUT and the PC were on the same switch.
- “iperf” version 1.7.0 was used for the tests.
- The data captured here is for "iperf" in client mode.
- The "iperf" tool is run on DUT1 and DUT2 with the NFS mounted.
- The Linux kernel is configured in 'Low Latency Desktop Mode'.

3.3.8 Performance: Server Mode

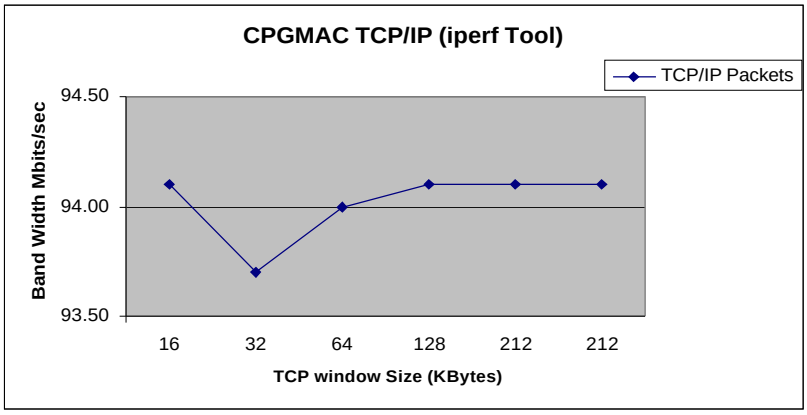


Figure 7. Runtime Performance Characteristics -IPERF_STRAIGHT_CABLE_Server

Table 11. Test Data

Test Setup Information	TCP window size in Kbytes	TCP/IP Packets		
		Bandwidth in Mbits/sec	Transfer size in Mbytes	Interval in sec
ARM Frequency = 283 MHZ	16	94.10	673	60
	32	93.70	670	60
	64	94.00	672	60
	128	94.10	673	60
	212	94.10	673	60
	212	94.10	673	60

3.3.8.1 Comments

- The CPGMAC is configured in auto negotiation, 100Mbps full duplex.
- The “iperf” tool is run on DUT1 in server mode and on DUT2 in client mode.
- Both the DUT and the PC are on same switch.
- iperf version 1.7.0 was used for the tests
- The data captured here is for “iperf” in client mode.
- The “iperf” tool is run on DUT1 and DUT2 with the NFS mounted.
- The Linux kernel is configured in ‘Server Mode’.

3.3.9 Performance: Desktop Mode

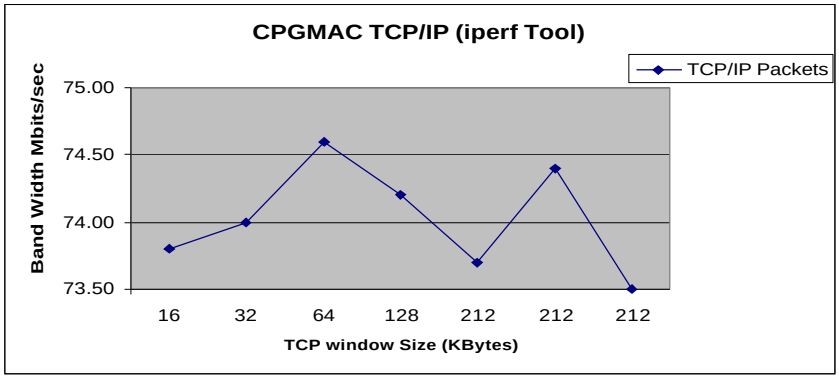


Figure 8. Runtime Performance Characteristics-IPERF_CROSS_CABLE_Desktop

Table 12. Test Data

Test Setup Information	TCP window size in KBytes	TCP/IP Packets		
		Bandwidth in Mbits/sec	Transfer size in MBytes	Interval in sec
ARM Frequency = 283 MHZ	16	73.80	528.00	60.00
	32	74.00	529.00	60.00
	64	74.60	533.00	60.00
	128	74.20	531.00	60.00
	212	73.70	527.00	60.00
	212	74.40	532.00	60.00
	212	73.50	526.00	60.00

3.3.9.1 Comments

- The CPGMAC is configured in auto negotiation, 100Mbps full duplex.
- The "iperf" tool is run on DUT1 in server mode and on DUT2 in client mode
- Both the DUT and the PC were on same switch
- “iperf” version 1.7.0 was used for the tests.
- The data captured here is for "iperf" in client mode.
- The "iperf" tool is run on DUT1 and DUT2 with the Ram disk ("initrd.image") mounted.
- The Linux kernel is configured in 'Desktop Mode'.

3.3.10 Performance: Low Latency Mode

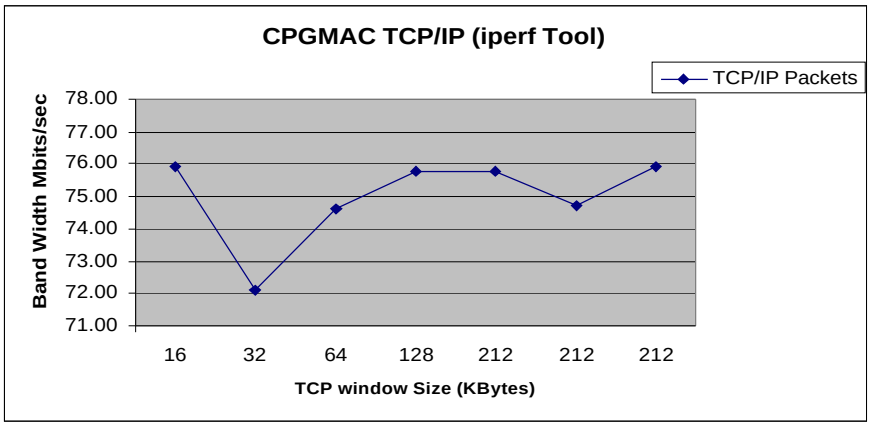


Figure 9. Runtime Performance Characteristics-IPERF_CROSS_CABLE_LowLatency

Table 13. Test Data

Test Setup Information	TCP window size in Kbytes	TCP/IP Packets		
		Bandwidth in Mbits/sec	Transfer size in Mbytes	Interval in sec
ARM Frequency = 283 MHZ	16	75.90	543.00	60.00
	32	72.10	516.00	60.00
	64	74.60	534.00	60.00
	128	75.80	542.00	60.00
	212	75.80	543.00	60.00
	212	74.70	534.00	60.00
	212	75.90	543.00	60.00

3.3.10.1 Comments

- The CPGMAC is configured in auto negotiation, 100Mbps full duplex.
- The “iperf” tool is run on DUT1 in server mode and on DUT2 in client mode
- Both the DUT and the PC were on same switch.
- “iperf” version 1.7.0 was used for the tests
- The data captured here is for “iperf” in client mode.
- The “iperf” tool is run on DUT1 and DUT2 with the Ram disk (“initrd.image”) mounted.
- The Linux kernel is configured in ‘Low Latency Desktop Mode’.

3.3.11 Performance: Server Mode

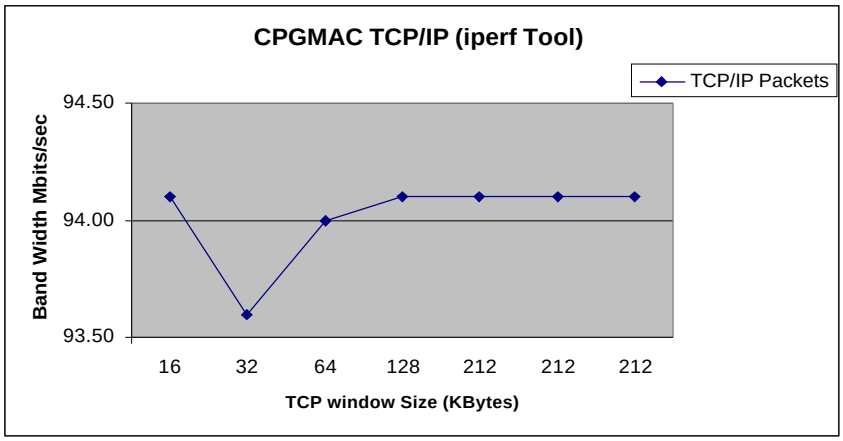


Figure 10. Runtime Performance Characteristics-IPERF_CROSS_CABLE_Server

Table 14. Test Data

Test Setup Information	TCP window size in KBytes	TCP/IP Packets		
		Bandwidth in Mbits/sec	Transfer size in MBytes	Interval in sec
ARM Frequency = 283 MHZ	16	94.10	673.00	60.00
	32	93.60	670.00	60.00
	64	94.00	673.00	60.00
	128	94.10	673.00	60.00
	212	94.10	673.00	60.00
	212	94.10	673.00	60.00
	212	94.10	673.00	60.00

3.3.11.1 Comments

- The CPGMAC is configured in auto negotiation, 100Mbps full duplex.
- The "iperf" tool is run on DUT1 in server mode and on DUT2 in client mode.
- Both the DUT and the PC were on same switch.
- “iperf” version 1.7.0 was used for the tests.
- The data captured here is for "iperf" in client mode.
- The "iperf" tool is run on DUT1 and DUT2 with the Ram disk ("initrd.image") mounted.
- The Linux kernel is configured in 'Server Mode'.

3.3.11.2 Example Network Applications

- *ping* is an echo client or echo server. It sends a ICMP echo request packet and waits for the ICMP echo reply.
- *traceroute* also uses ICMP packets. A different ICMP command will cause a bounce if it must be routed. This allows traceroute to track the route to the requested destination.

- *iperf* is a tool to measure network (TCP or UDP) performance. It will report bandwidth, jitter, and packet loss.
- Telnet is a protocol used to provide command line logins between hosts connected via TCP/IP.

3.4 USB Drivers

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

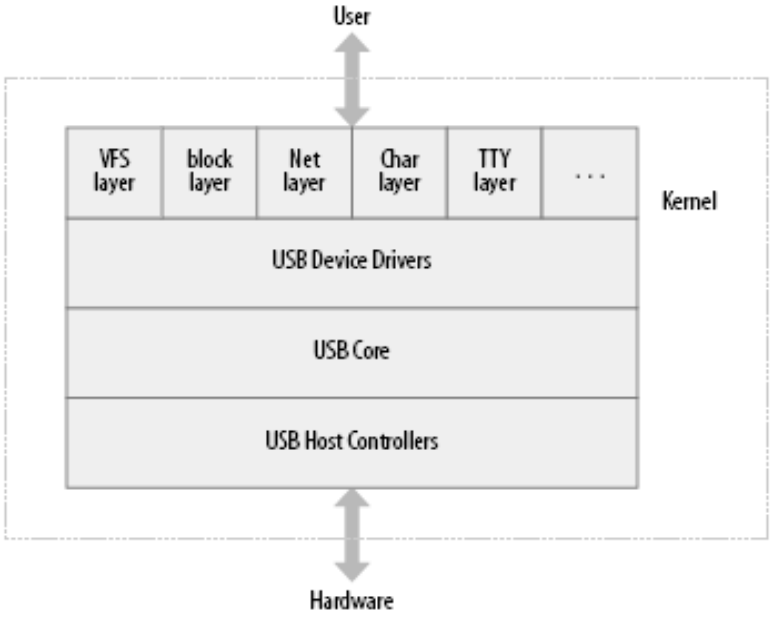


Figure 11. Linux Kernel USB Driver

3.4.1 USB Slave Driver

3.4.1.1 USB Slave Driver Features

- Supports data transfer in both PIO and DMA mode for all preemption modes.

3.4.1.2 USB Slave Driver Known Issues

- The SysCache test and Storage Data Verification Filter test failed.
- The module use count is not incremented for USB target (g_file_storage.ko).
- The module can be unloaded even while IO operation is in progress.
- During WHQL tests, the application failed to perform the test the first time.
- The "USB Serial Number" test fails with WHQL.

3.4.2 USB Host Driver

3.4.2.1 USB Host Driver Features

- Supports both PIO and DMA mode for all preemption modes.

3.4.2.2 USB Host Driver Known Issues

- The module can be unloaded/removed during IO operation.

3.4.3 USB OTG

3.4.3.1 USB OTG Features

- The OTG functionality is invoked by providing inputs from proc entry. This proc entry is created at *musb* controller driver initialization.
- In OTG mode, the VBus is off at initialization. To operate devices, the application needs to pass inputs to the driver.

3.4.3.2 USB OTG Known Issues

- USB OTG driver has been tested on modified gamma boards, which have a hardware workaround.
- USB OTG has not been tested using the OPT test suite.
- Following the hardware modification for USB OTG: EMAC and USB OTG will be mutually exclusive.

3.4.3.3 Commands

- i - Initiate Role identification, Start /Request session.
- e - End the Session on VBus
- s - Suspend USB bus

3.4.3.4 Operation sequence:

1. Start Session / Request Session and initiate role identification:

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

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.

3. Relinquish Host role from B-Device:

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

When OTG mode is selected, ensure that the bus is turned on by doing Step 1 "Start Session / Request Session". If this is not done, the connection will not be enumerated.

3.4.4 DVEVM Board Features

- The DaVinci DVEVM uses a 24 MHz crystal for the USB II clock generator. The USB controller is completely integrated in the DaVinci device.
- The J7 connector is a 3-position jumper located on the top side of the board and is used to select the termination for the USB host/client configuration on USB connector J10. Either +3.3V (1-2 position) or Ground (2-3 position) must be selected.

- Connector J10 is a USB connector. Three different connectors can be mounted at location J10. The default connector is USB host.

3.4.5 Menu Configuration: USB driver

3.4.5.1 USB Driver: Host Mode

Default configuration support for USB host mode.

```
Device Drivers  --->
            SCSI device support  --->
                  <*> SCSI device support
                  [*]   legacy /proc/scsi/support
                              ---   SCSI support type (disk, tape, CD-ROM)
                              <*>   SCSI disk support
USB support  --->
                  <*> Support for Host-side USB
                  ---   Miscellaneous USB options
                  [*]   USB device filesystem
                  --- USB Host Controller Drivers
                  <*> Inventra USB Highspeed Dual Role Controller Support
                  ---   DaVinci 644x USB support
                        Driver Mode (USB Host)  --->
                              (X) USB Host
                  --- USB Device Class drivers
                  <*> USB Mass Storage support
                  --- USB Input Devices
                  <*> USB Human Interface Device (full HID) support
                  [*]   HID input layer support
```

3.4.5.2 USB Driver: Gadget Mode.

```
Device Drivers  --->
      USB support  --->
            USB Gadget Support  --->
                  <*> Support for USB Gadgets
                  <*>   USB Gadget Drivers (Ethernet Gadget)  --->
                        (X) Ethernet Gadget
                  [*]       RNDIS support (EXPERIMENTAL) (NEW)
```

3.4.5.3 USB Driver: OTG

```
1. Enable options for Host mode.
2. Enable options for Gadget mode.
3. Enable following option:
Device Drivers  --->
      USB support  --->
            ---   DaVinci 644x USB support
            Driver Mode (USB Host)  --->
            (X) Both host and peripheral:  USB OTG (On the Go) Device
```

3.4.6 Performance and Benchmarks

Table 15. USB Memory Statistics

Configuration	Memory Statistics[1]				
	Program Memory	Data Memory			Total
		Initialized	Uninitialized	Stack	

usbcore.ko	53392	2008	1060		56460
musb_hdrc.ko	26320	664	0		26984
usb-storage.ko	24320	4700	0		29020
dummy_hcd.ko	14256	400	180		14836
g_ether.ko	24932	1092	140		26164
g_file_storage.ko	27200	872	148		28220
g_serial.ko	17134	868	332		18334
g_zero.ko	10272	680	112		11064
gadgetfs.ko	16160	1108	16		17284

[1] All memory requirements are expressed in bytes for **full-featured** device driver usage.

3.4.7 Performance: USB Host Write

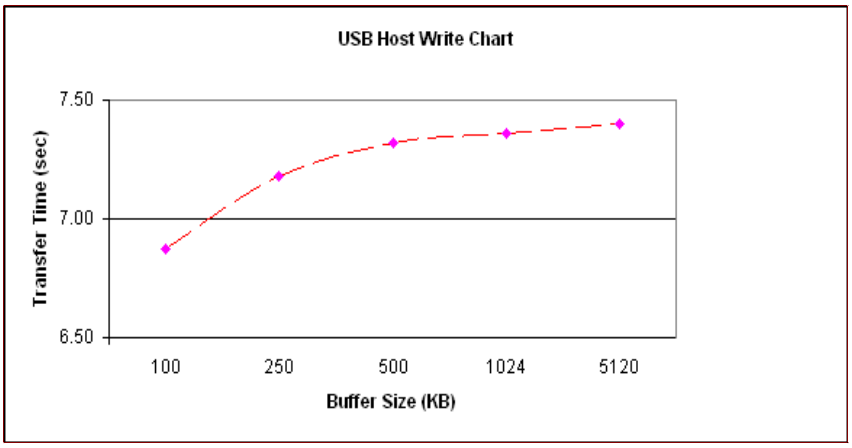


Figure 12. USB Host Write Performance Results

Table 16. Test Data

Test Setup Information	USB Host Write Results		
	Buffer size (KB)	Total Bytes Transferred (GB)	Transfer Time (sec)
ARM Frequency = 283 MHz	100	1	6.87
Mode = PIO	250	1	7.18
	500	1	7.32
	1024	1	7.36
	5120	1	7.40

3.4.7.1 Remarks

- The performance is measured using gettimeofday() function.
- Hard Disk - IOmega (40GB) is used for writing files.
- The accuracy of the performance time is limited because it involves loop time and time is taken from the gettimeofday() function.

3.4.8 Performance: USB Host Read

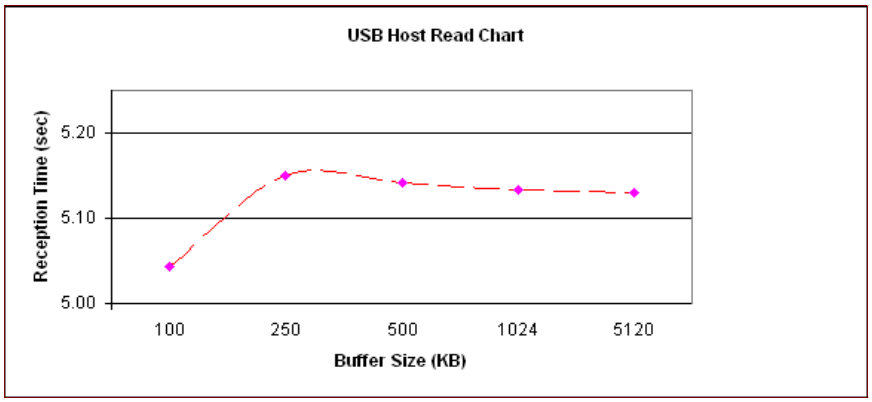


Figure 13. USB Host Read Performance Results

Table 17. Test Data

Test Setup Information	USB Host Read Result s		
	Buffer size (KB)	Total Bytes Received (GB)	Reception Time (sec)
ARM Frequency = 283 MHz	100	1	5.04
Mode = PIO	250	1	5.15
	500	1	5.14
	1024	1	5.13
	5120	1	5.13

3.4.8.1 Remarks

- The performance is measured using gettimeofday() function.
- Hard Disk - IOmega (40GB) is used for reading files.
- The accuracy of the performance time is limited since it involves loop time and time is taken from gettimeofday() function.

3.4.9 Performance: USB Slave Write

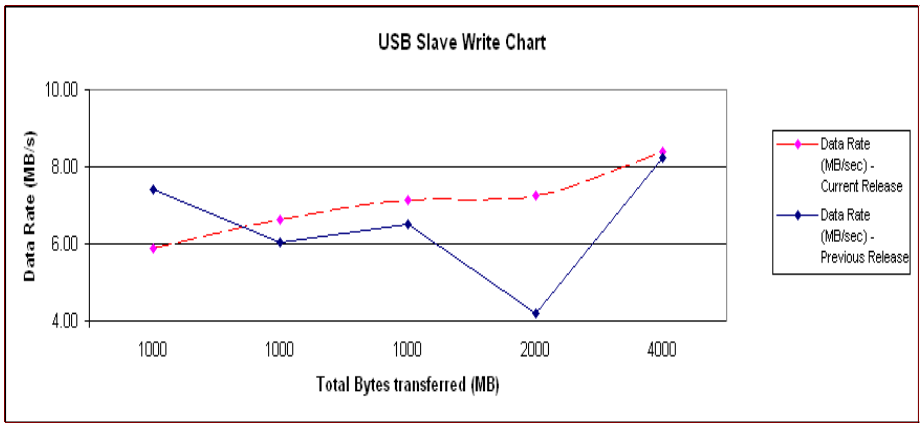


Figure 14. USB Slave Write Performance Results

Table 18. Test Data

Test Setup Information	USB Slave Write Results				
	Bytes Transferred (MB)	**Number of files**	**Total Bytes transferred (MB)**	**Data Rate (MB/sec) - Previous Release**	**Data Rate (MB/sec) - Current Release**
ARM Frequency = 283 MHz	50	20	1000	7.41	5.88
Mode = DMA	10	100	1000	6.02	6.62
	5	200	1000	6.49	7.14
	100	20	2000	4.20	7.27
	200	20	4000	8.23	8.40

3.4.9.1 Remarks

- The performance is measured using Windows XP PC.
- Hard Disk - Toshiba (40GB) is used for writing files.
- The accuracy of the performance time is limited since it involves loop time and time is taken from Windows XP PC.

3.4.10 Performance: USB Slave Read

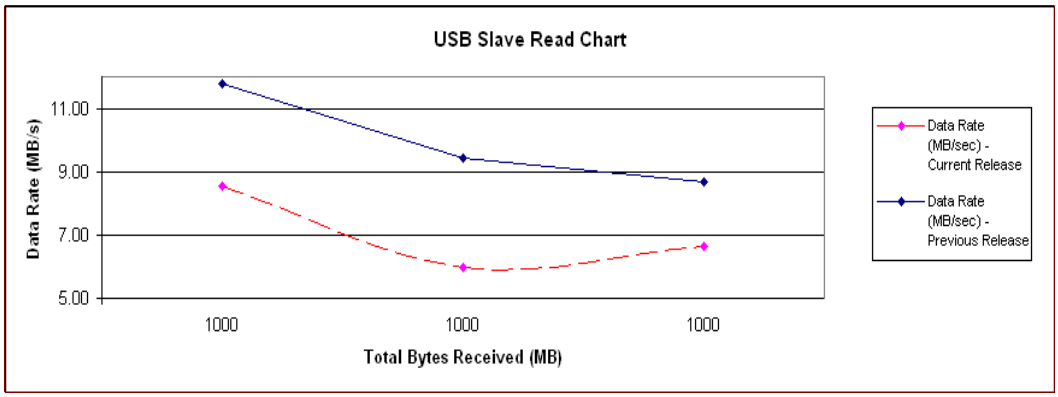


Figure 15. USB Slave Read Performance

Table 19. Test Data

Test Setup Information	USB Slave Read Results				
	Bytes Transferred (MB)	**Number of files**	**Total Bytes Received (MB)**	**Data Rate (MB/sec) - Previous Release**	**Data Rate (MB/sec) - Current Release**
ARM Frequency = 283 MHz	50	20	1000	11.76	8.54
Mode = DMA	10	100	1000	9.43	5.98
	5	200	1000	8.70	6.62

3.4.11 Remarks

- Performance was measured using a Windows XP PC.
- A Toshiba 40GN hard disk was used for reading files.
- The performance measurements include loop time overhead, and are taken from a Windows PC, so accuracy is limited.

3.5 UART Driver

3.5.1 Description

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.

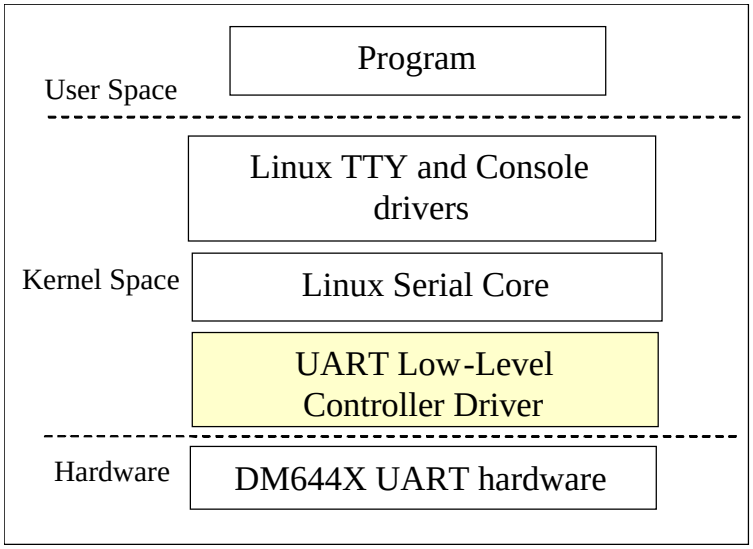


Figure 16. Linux UART Kernel Driver

3.5.2 Driver Features

- Supports UART0 only.

3.5.3 Known Issues

- UART data bit and parity setting fails.
- Continuous UART input overruns under moderate load.

3.5.4 Supported System Calls

open(), close(), read(),write(), Ioctl()

3.5.5 Supported IOCTLs

Table 20. Supported IOCTLs

Constant	Description
TIOCGSERIAL	Get device parameters from the UART (e.g. port type, port num, baud rate, base divisor, etc.)
TIOCSSERIAL	Set UART device parameters (e.g port type, port num, baud rate, base divisor, etc.)

3.5.6 DVEVM Board Feature

- The internal UART0 interface is level-shifted and routed to the RS-232 line drivers.

3.5.7 Menu Configuration: UART Driver

```
      Device Drivers  --->
           Character devices  --->
                Serial drivers  --->
                     <*> 8250/16550 and compatible serial support
                     [*]   Console on 8250/16550 and compatible serial
port
                     (2)   Maximum number of non-legacy 8250/16550
```

3.5.8 Performance and Benchmarks

Table 21. UART Memory Statistics

Configuration	Memory Statistics [1]				
	Program Memory	Data Memory			Total
		Initialized	Uninitialized	Stack	
Serial.o	28	124	0		152
Serial_core.o	16292	204	0		15496
8205.0	13556	320	640		14516

[1] All memory requirements are expressed in bytes for **full-featured** device driver usage.

3.5.9 Notes

- The current driver software supports only UART0.
- UART1 is multiplexed with ATA (CF) pins on the DaVinci DMSoC.
- UART2 is multiplexed with VPFE.

3.6 I2C Driver

3.6.1 Description

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.

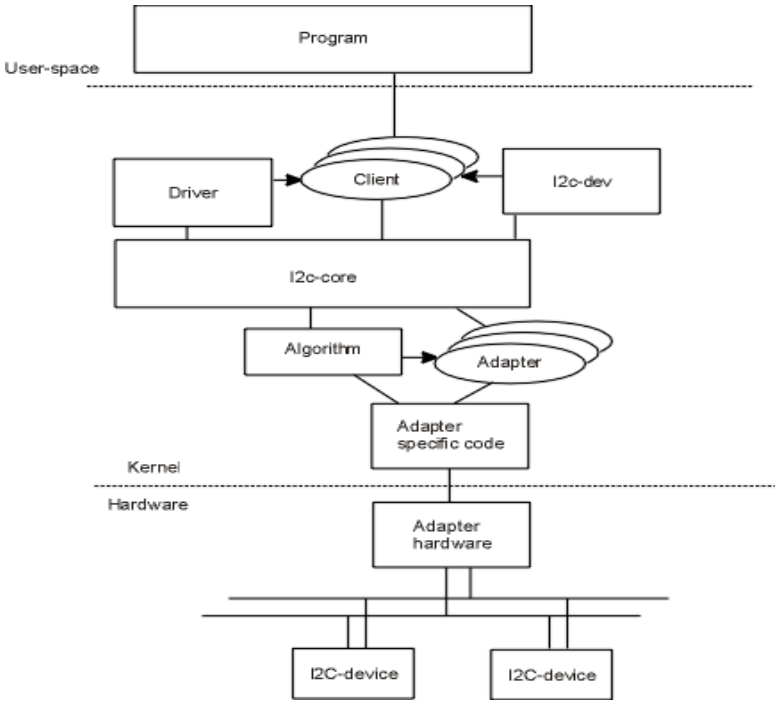


Figure 17. Linux Kernel I2C Driver

3.6.2 Driver Features

- 10-bit addressing is experimental in Linux, and is not supported.
- DMA mode is not supported.

3.6.3 Known Issues

- Due to hardware limitations of the MSP430, the I2C runs at 20KHz. This may cause intermittent failures of some I2C functions (e.g. I2C write).

3.6.4 Supported System Calls

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

3.6.5 Supported IOCTLs

Constant	Description
I2C_SLAVE_FORCE	Change slave address. Slave address is 7 or 10 bits. This changes the address,

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

3.6.6 DVEVM Board Features

- The DaVinci DVEVM is equipped with three PCF8574A IO expanders to handle various bit I/O functions. The I/O Expanders are at 0x38, 0x39 and 0x3A slave addresses. Each of the eight I/Os can be independently used as GPIO pins through the I2C interface.
- The DVEVM I2C bus also contains an I2C EEPROM.
- The DVEVM incorporates an MSP430 that supports infrared (IR), real-time clock (RTC) and bit I/O. The I2C interface on the DaVinci processor is used to communicate to the MSP430.

3.6.7 Menu Configuration: I2C Driver

```
Device Drivers  --->
     I2C support  --->
          <*> I2C support
          <*> I2C device interface
```

3.6.8 Performance and Benchmarks

Table 22. I2C Driver Memory Statistics

Configuration	Memory Statistics [1]				
	Program Memory	Data Memory			Total
		Initialized	Uninitialized	Stack	
I2c-davinci.o	3424	840	40		4304
I2c-core.o	11092	568	16		11676
i2c-dev.ko	5760	960	1024		7744

[1] All memory requirements are expressed in bytes for **full-featured** device driver usage.

3.6.9 Testing I2C with eeprog Utility

eeprog is a Linux C program that allows the user to read and write to 24Cxx EEPROM. The *eeprog* utility can be downloaded from the following link: http://codesink.org/download/eeprog-0.7.6.tar.gz. Unpack the source and type “make” to build the utility. Necessary changes should be done to the Makefile to compile the utility for DaVinci. eeprog uses SMBus commands to read/write to I2C EEPROM.

The Ethernet address present in the EEPROM can be read using the eeprog utility as follows:

```
# eeprog -16 –f /dev/i2c/0 0x50 –x –r 0x7f00:10
eeprog 0.7.6, a 24Cxx EEPROM reader/writer
Copyright (c) 2003-2004 by Stefano Barbato - All rights reserved.
  Bus: /dev/i2c/0, Address: 0x50, Mode: 16bit
  Reading 10 bytes from 0x7f00

7f00|  00 0e 99 02 53 48 ff ff   ff ff
```

The above command reads ten bytes from the EEPROM at address 0x50 on bus 0 starting at address 0x7f00 (hexadecimal) and prints it in hexadecimal format (-x). The “-16” flag is required to specify 16-bit address mode. The “-f” flag disables warnings and doesn’t ask for confirmation.

3.7 IDE ATA Driver

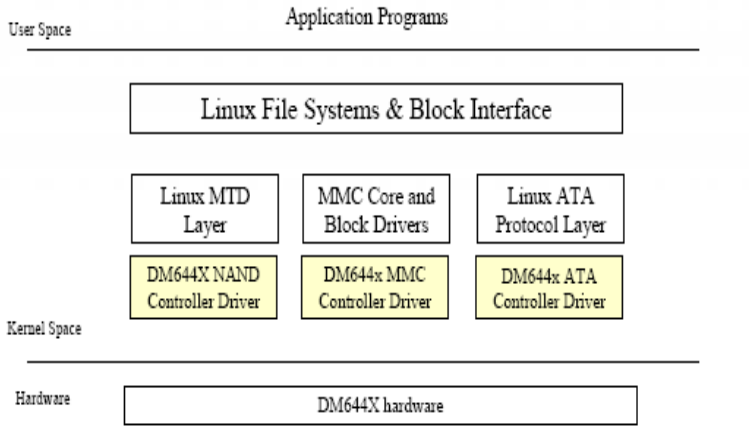


Figure 18. Linux Kernel IDE Driver

3.7.1 Description

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

3.7.2 Driver Features

- 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).

3.7.3 Known Issues

- NTFS file system is not supported.
- Performance not consistent with theoretical values.
- Removing IDE module with a mounted file system succeeds.
- Some drives (e.g. Maxtor) log error messages when switched to sleep mode.

3.7.4 Supported System Calls

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

3.7.5 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.
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.

3.7.6 DVEVM Board Features

- The hard drive provided on the evaluation board is a 40 GB IDE drive.
- The DaVinci ATA interface is multiplexed with the EMIF.

3.7.7 Menu Configuration: IDE Driver

```
Device Drivers --->
<*> ATA/ATAPI/MFM/RLL support-->
<*> Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support-->
          <*> Include IDE/ATA-2 DISK support
          [*]  IDE Taskfile Access
          <*> generic/default IDE chipset support
               <*> Davinci IDE interface support
```

3.7.8 Performance and Benchmarks

The EXT2 file system is used to read/write the file from/to the IDE. The runtime performance is described below.

Table 23. IDE Driver Memory Statistics

Configuration	Memory Statistics[1]				
	Program Memory	Data Memory			Total
		Initialized	Uninitialized	Stack	
palm_bk3710.ko	4200	696	76		4972
ide-generic.ko	300	256	0		556
ide-core.ko	69584	1672	7156		78412
ide-disk.ko	15572	560	0		16132

[1] All memory requirements are expressed in bytes for **full-featured** device driver usage.

3.7.9 Performance: Server Mode

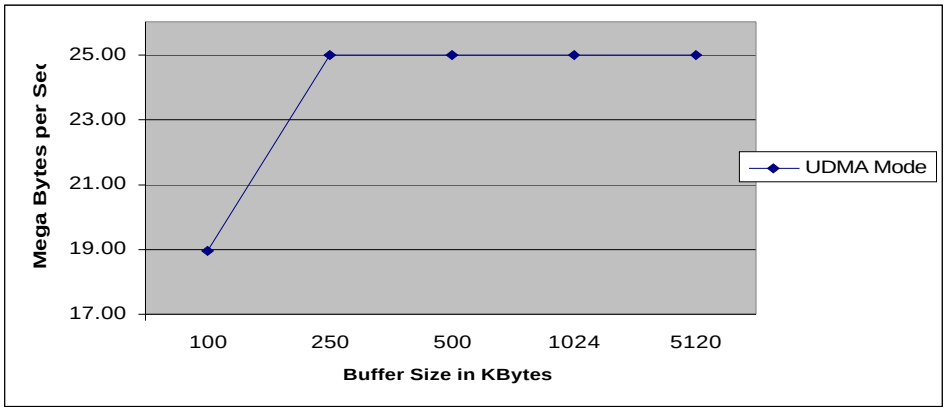


Figure 19. Runtime Performance Characteristics - ATA_FileWrite_283MHZ_ServerMode

Table 24. Test Data

Test Setup Information	Buffer Size in kBytes	UDMA Mode		
		Mega Bytes/Sec	File Size in MB	Duration in Sec
ARM Frequency = 283 MHZ	100	18.96	1024	54
	250	24.98	1024	41
	500	24.98	1024	41
	1024	24.98	1024	41
	5120	24.98	1024	41

3.7.9.1 Comments

- HDD used for testing: TOSHIBA MK4019GAX
- Mode tested: Auto Mode (UDMA 4)
- File system used: EXT3

3.7.9.2 Theoretical Performance Values

I/O data transfer rate (Mbytes/sec max) = 66.6Mbytes/sec (UDMA4)

3.7.10 Performance: Desktop Mode

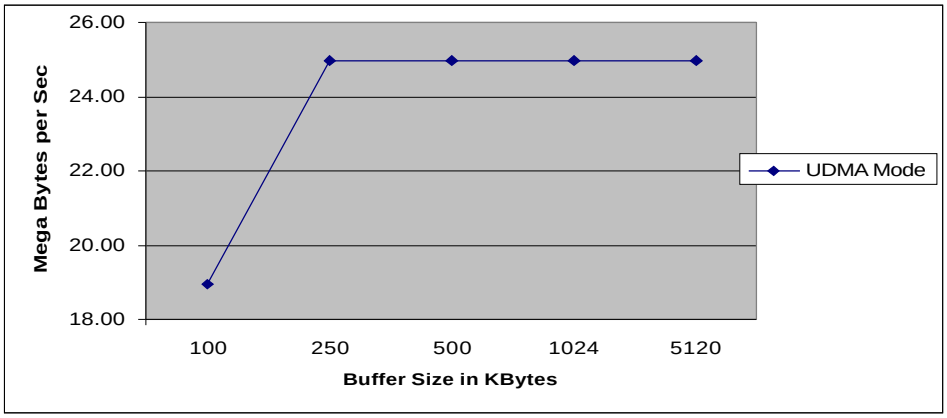


Figure 20. Runtime Performance Characteristics – ATA_FileWrite_283MHZ_DesktopMod

Table 25. Test Data

Test Setup Information	Buffer Size in Kbytes	UDMA Mode		
		Mega Bytes/Sec	File Size in MB	Duration in Sec
ARM Frequency = 283 MHZ	100	18.96	1024	54
	250	24.98	1024	41
	500	24.98	1024	41
	1024	24.98	1024	41
	5120	24.98	1024	41

3.7.10.1 Comments

- HDD used for testing: TOSHIBA MK4019GAX
- Mode tested: Auto Mode (UDMA 4)
- File system used: EXT2

3.7.10.2 Theoretical Performance Values

- I/O Data Transfer Rate (Mbytes/sec max) = 66.6Mbytes/sec (UDMA4)

3.7.11 Performance: Low Latency

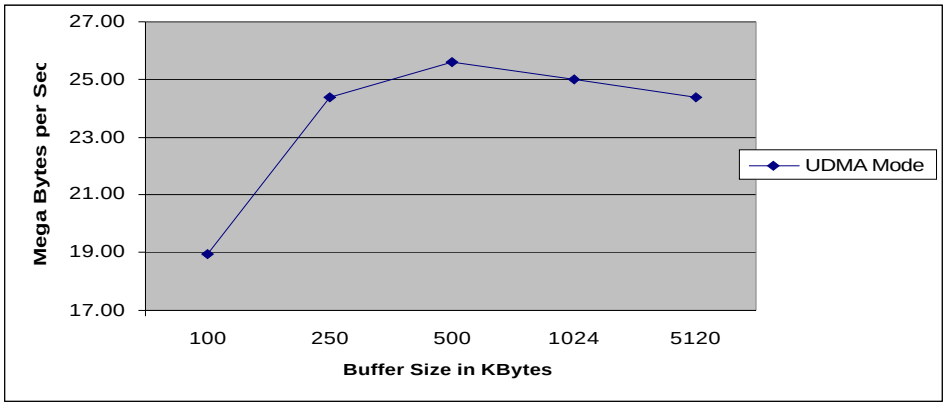


Figure 21. Runtime Performance Characteristics – ATA_FileWrite_283MHZ_LowLatency

Table 26. Test Data

Test Setup Information	Buffer Size in KBytes	UDMA Mode		
		Mega Bytes/Sec	File Size in MB	Duration in Sec
ARM Frequency = 283 MHZ	100	18.96	1024	54
	250	24.38	1024	42
	500	25.60	1024	40
	1024	24.98	1024	41
	5120	24.38	1024	42

3.7.11.1 Comments

- HDD used for testing: TOSHIBA MK4019GAX
- Mode tested: Auto Mode (UDMA 4)
- File system used: EXT2

3.7.11.2 Theoretical Performance Values

- I/O Data Transfer Rate (Mbytes/sec max) = 66.6Mbytes/sec (UDMA4)

3.7.12 Performance: Real-Time

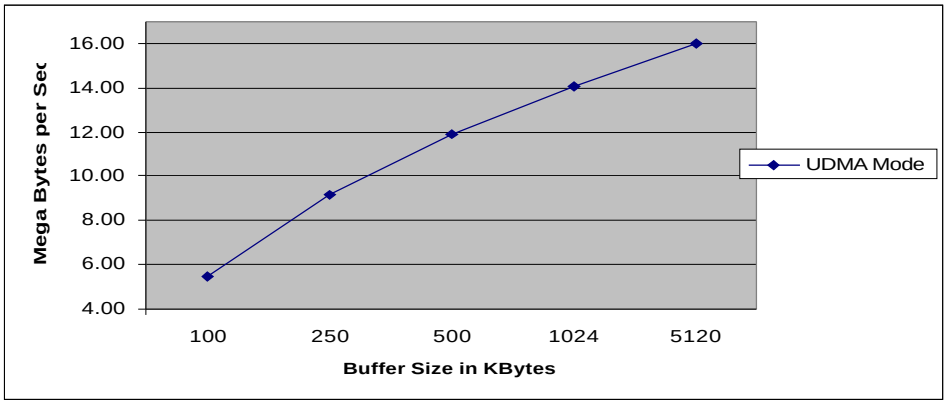


Figure 22. Runtime Performance Characteristics – ATA_FileWrite_283MHZ_RealTime

Table 27. Test Data

Test Setup Information	Buffer Size in KBytes	UDMA Mode		
		Mega Bytes/Sec	File Size in MB	Duration in Sec
ARM Frequency = 283 MHZ	100	5.45	1024	188
	250	9.14	1024	112
	500	11.91	1024	86
	1024	14.03	1024	73
	5120	16.00	1024	64

3.7.12.1 Comments

- HDD used for Testing: - TOSHIBA MK4019GAX
- Mode Tested: - Auto Mode (UDMA 4)
- File system Used: - EXT2 File System

3.7.12.2 Theoretical Performance Values

- I/O Data Transfer Rate (Mbytes/sec max) = 66.6Mbytes/sec (UDMA4)

3.7.13 Performance: Server Mode

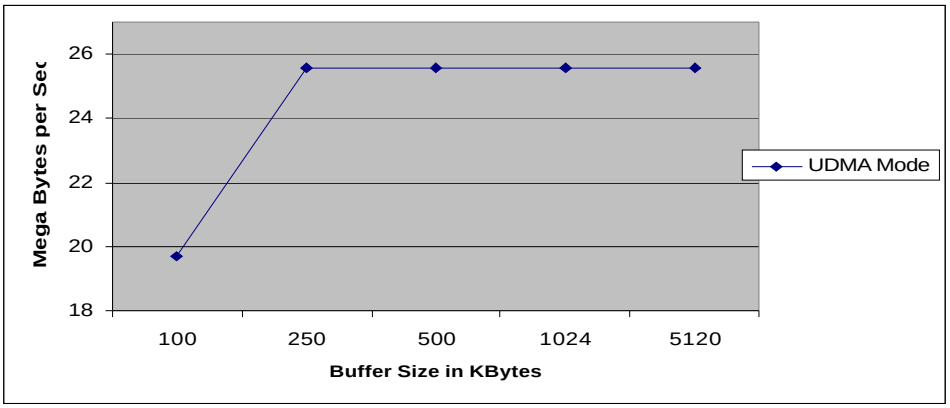


Figure 23. Runtime Performance Characteristics – ATA_FileRead_283MHZ_ServerMode

Table 28. Test Data

Test Setup Information	Buffer Size in Kbytes	UDMA Mode		
		Mega Bytes/Sec	File Size in MB	Duration in Sec
ARM Frequency = 283 MHZ	100	19.69	1024	52
	250	25.60	1024	40
	500	25.60	1024	40
	1024	25.60	1024	40
	5120	25.60	1024	40

3.7.13.1 Comments

- HDD used for Testing: - TOSHIBA MK4019GAX
- Mode Tested: - Auto Mode (UDMA 4)
- File system Used: - EXT2 File System

3.7.13.2 Theoretical Performance Values

- I/O Data Transfer Rate (Mbytes/sec max) = 66.6Mbytes/sec (UDMA4)

3.7.14 Performance: Desktop Mode

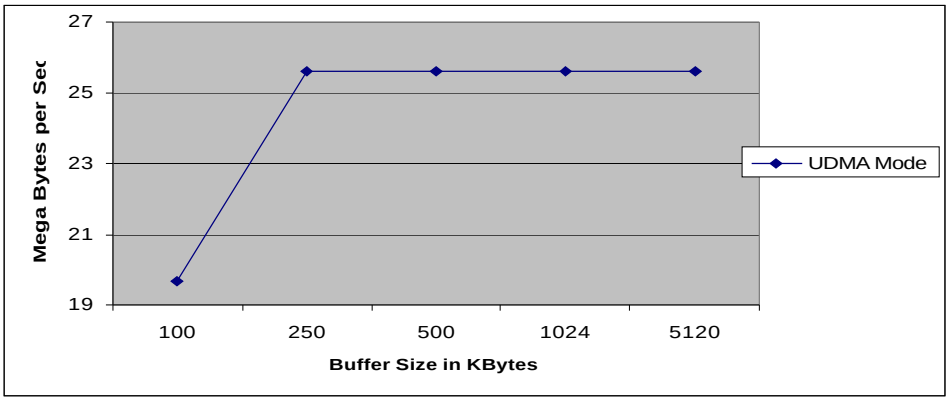


Figure 24. Runtime Performance Characteristics – ATA_FileRead_283MHZ_DesktopMode

Table 29. Test Data

Test Setup Information	Buffer Size in Kbytes	UDMA Mode		
		Mega Bytes/Sec	File Size in MB	Duration in Sec
ARM Frequency = 283 MHZ	100	19.69	1024	52
	250	25.60	1024	40
	500	25.60	1024	40
	1024	25.60	1024	40
	5120	25.60	1024	40

3.7.14.1 Comments

- HDD used for Testing: - TOSHIBA MK4019GAX
- Mode Tested: - Auto Mode (UDMA 4)
- File system Used: - EXT2 File System

3.7.14.2 Theoretical Performance Values

- I/O Data Transfer Rate (Mbytes/sec max) = 66.6Mbytes/sec (UDMA4)

3.7.15 Performance: LowLatency

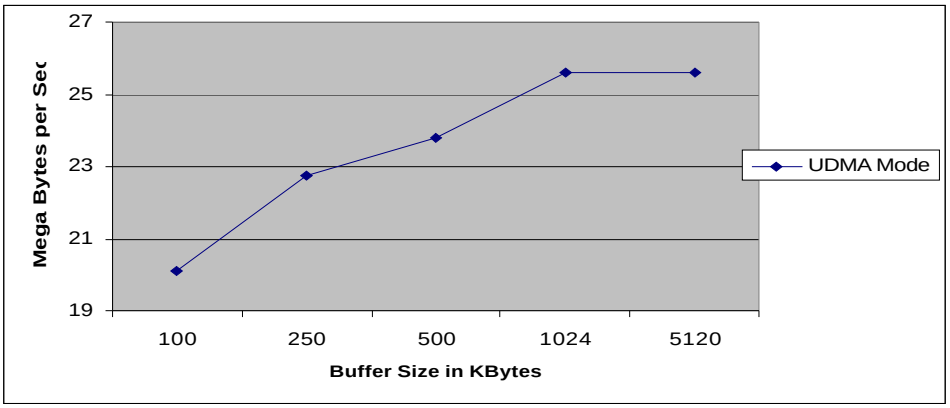


Figure 25. Runtime Performance Characteristics – ATA_FileRead_283MHZ_LowLatency

Table 30. Test Data

Test Setup Information	Buffer Size in Kbytes	UDMA Mode		
		Mega Bytes/Sec	File Size in MB	Duration in Sec
ARM Frequency = 283 MHZ	100	20.08	1024	51
	250	22.76	1024	45
	500	23.81	1024	43
	1024	25.60	1024	40
	5120	25.60	1024	40

3.7.15.1 Comments

- HDD used for Testing: - TOSHIBA MK4019GAX
- Mode Tested: - Auto Mode (UDMA 4)
- File system Used: - EXT2 File System

3.7.15.2 Theoretical Performance Values

- I/O Data Transfer Rate (Mbytes/sec max) = 66.6Mbytes/sec (UDMA4)

3.7.16 Performance: Real-Time

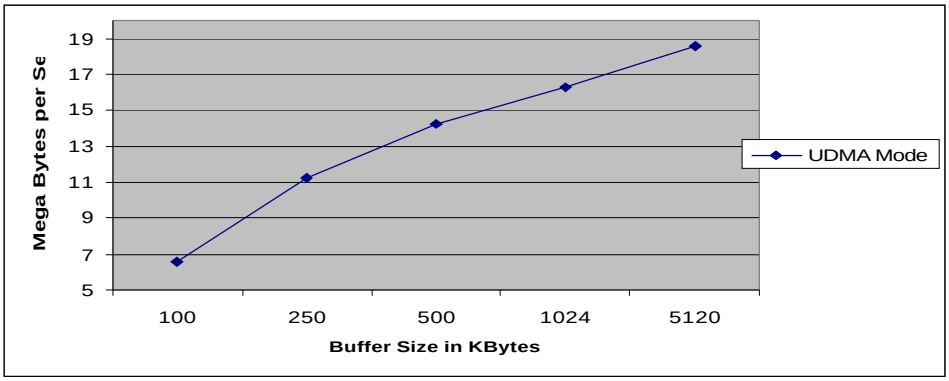


Figure 26. Runtime Performance Characteristics – ATA_FileRead_283MHZ_RealTime

Table 31. Test Data

Test Setup Information	Buffer Size in Kbytes	UDMA Mode		
		Mega Bytes/Sec	File Size in MB	Duration in Sec
ARM Frequency = 283 MHZ	100	6.61	1024	155
	250	11.25	1024	91
	500	14.22	1024	72
	1024	16.25	1024	63
	5120	18.62	1024	55

3.7.16.1 Comments

- HDD used for Testing: - TOSHIBA MK4019GAX
- Mode Tested: - Auto Mode (UDMA 4)
- File system Used: - EXT2 File System

3.7.16.2 Theoretical Performance Values

- I/O Data Transfer Rate (Mbytes/sec max) = 66.6Mbytes/sec (UDMA4)

3.7.17 Notes

- The ATA peripheral shares data lines and some control signals with EMIFA, GPIO and SPI.
- ATA DMA pins are multiplexed with UART1.
- When the hard drive is in use, other peripherals on the EM bus are not accessible, such as the Compact Flash interface and the XD/SM media cards.

3.8 NAND Driver

3.8.1 Description

NAND flash is accessible via the asynchronous external memory interface (EMIFA). When a chip select space is configured to operate in NAND Flash mode, the EMIF hardware can calculate the error correction code (ECC). ARM ROM supports booting of the DM6446 ARM processor from NAND flash.

The NAND driver is implemented as a block driver, compliant with the standard MTD driver. It supports various NAND Flash chips. (Refer to ”drivers/mtd/nand/nand_ids.h”.) The NAND driver creates the device nodes for user space access (/dev/mtdblock0, /dev/mtdblock1, etc.)

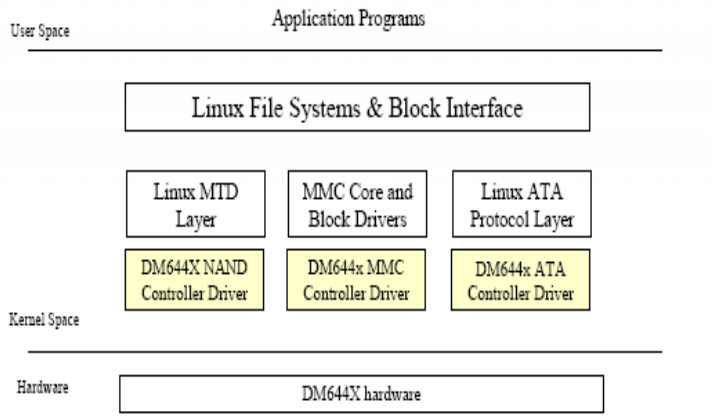


Figure 27. Linux Kernel NAND Driver

3.8.2 Driver Features

- NAND is tested with the JFFS2 file system.
- NAND has been divided into 4 partitions:
  - 256 KB read-only partition for U-Boot.
  - 128 KB read-only partition for environment variables.
  - 4 MB read/write partition for Linux.
  - Remainder for file system and others (read/write).

3.8.3 Known Issues

- The *rmmod* command to unlink the NAND module is not supported. If there is more than one partition, it fails with OOPS.
- Mount of a NAND partition fails after performing NAND write with the –j option.

3.8.4 Supported System Calls

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

3.8.5 Supported IOCTLs

Constant	Description
MEMGETREGIONCOUNT	Return number of era se 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.

3.8.6 DVEVM Board Feature

- The DVEVM is equipped with one 64MB NAND Flash. It is connected through the asynchronous EMIF (EMIFA).

3.8.7 Menu Configuration: NAND Driver: To enable the MTD device (NAND).

```
Device Drivers à
     Memory Technology Devices à
          <*> Memory Technology Device (MTD) Support
          <*> Direct char device access to MTD devices
          <*> Caching block device access to MTD devices
     NAND Flash Device Drivers à
               <*> NAND Device Support
               [ * ] Verify NAND page writes
               <*> NAND Flash device on DaVinci SoC
               [ * ] Hardware ECC Support on NAND Device for
DaVinci
```

3.8.8 Performance and Benchmarks

The JFFS2 file system is used to read/write from the NAND. The runtime performance is described below.

Table 32. NAND Memory Statistics

Configuration	Memory Statistics[1]				
	Program Memory	Data Memory			Total
		Initialized	Uninitialized	Stack	
nand_davinci.ko	3280	280	20		3580

[1] All memory requirements are expressed in bytes for **full-featured** device driver usage.

3.8.9 Performance: JFFS2 File Read

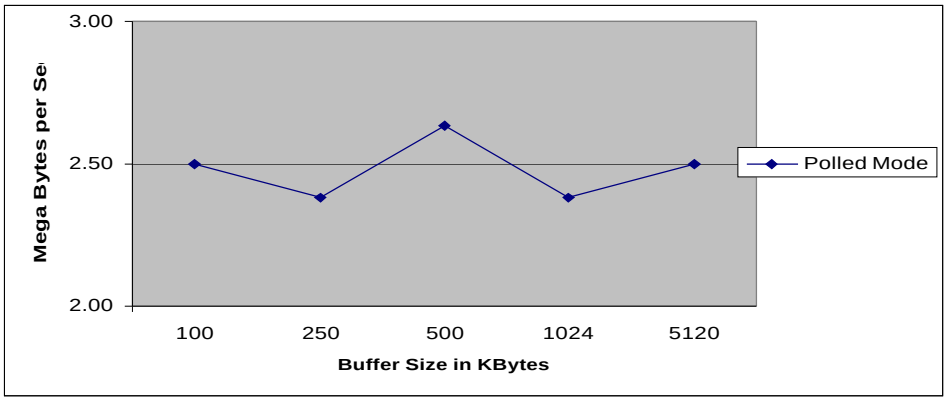


Figure 28. Runtime Performance Characteristics – JFFS2 File Read

Table 33. Test Data

Test Setup Information	Buffer Size in KBytes	Polled Mode		
		Mega Bytes/Sec	File Size in MB	Duration
ARM Frequency = 283 MHZ	100	2.50	50	20
	250	2.38	50	21
	500	2.63	50	19
	1024	2.38	50	21
	5120	2.50	50	20

3.8.9.1 Comments

- NAND Device Tested: Samsung K9K1208Q0C
- Mode Tested: Polled Mode

3.8.9.2 Theoretical Performance Values

- Read Speed: 16.1 Mbytes/sec (For Small Block NAND)

3.8.9.3 System Setup

- Read performance is measured by reading the file after a mount/unmount cycle.

3.8.10 Performance: JFFS2 File Write

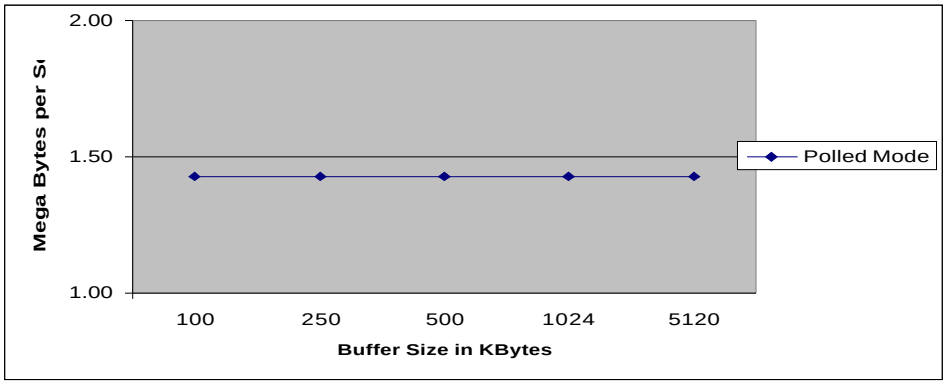


Figure 29. Runtime Performance Characteristics – JFFS2 File Write

Table 34. Test Data

Test Setup Information	Buffer Size in KBytes	Polled Mode		
		Mega Bytes/Sec	File Size in MB	Duration
ARM Frequency = 283 MHZ	100	1.43	50	35
	250	1.43	50	35
	500	1.43	50	35
	1024	1.43	50	35
	5120	1.43	50	35

3.8.10.1 Comments

- NAND device tested: SAMSUNG K9k1208Q0C
- Mode tested: Polled mode

3.8.10.2 Theoretical Performance Values

- Write speed: 6.9 Mbytes/sec (for small block NAND)

3.8.10.3 Notes

- As the pins required for EMIF (NAND) and ATA are multiplexed, thus NAND and ATA are mutually exclusive. Only either of the one can be used at a time.

3.8.11 Booting from NAND with a JFFS2 File System

Follow the steps listed below to boot out of JFFS2 file system on NAND:

4. The NAND flash layout is defined in arch/arm/mach-davinci/board-evm.c:

```
static struct mtd_partition nand_partitions[] = {
     /* bootloader (U-Boot, etc) in first sector */
     {
           .name             = "bootloader",
           .offset           = 0,
           .size             = SZ_256K,
           .mask_flags       = MTD_WRITEABLE, /* force read -only */
     },
     /* bootloader params in the next sector */
     {
           .name             = "params",
           .offset           = MTDPART_OFS_APPEND,
           .size             = SZ_128K,
           .mask_flags       = MTD_WRITEABLE,  /* force read-only */
     },
     /* kernel */
     {
           .name             = "kernel",
           .offset           = MTDPART_OFS_APPEND,
           .size             = SZ_4M,
           .mask_flags       = 0
     },
     /* file system */
     {
           .name             = "filesystem",
           .offset           = MTDPART_OFS_APPEND,
           .size             = MTDPART_SIZ_FULL,
           .mask_flags       = 0
     }
};
```

The four MTD partitions should look like this:

```
0x00000000-0x00040000 : "bootloader"
0x00040000-0x00060000 : "params"
0x00060000-0x00460000 : "kernel"
0x00460000-0x04000000 : "filesystem"
```

The JFFS2 file system will reside in the 4th partition.

5. Prepare the JFFS2 filesystem. (Perform these steps on the Linux Desktop.)
   a. Download the mtd utilities from http://www.linux-mtd.infradead.org/ and build it to generate the mkfs.jffs2 utility.
   b. Gather the rootfs to generate the filesystem. (Use the sample initrd.image.gz)
      1) gunzip initrd.image.gz
      2) mkdir myrootfs
      3) mount -o loop initrd.image myrootfs
   c. Create the JFFS2 filesystem
      4) mkfs.jffs2 -d myrootfs -o rootfs.jffs2 -e 0x4000
6. Erase the NAND flash using the flash_eraseall command and use the nandwrite utility to write the rootfs.jffs2 onto /dev/mtd3.
7. Modify the boot environment variables to boot to the new NAND JFFS2 file system.

```
Example: setenv bootargs mem=120M console=ttyS0,115200n8 root=/dev/mtdblock3 rw
rootfstype=jffs2 init=/bin/ash
```

8. Reboot the board to boot to the new JFFS2 file system on the NAND flash.

   While booting, the following messages can be seen on the HyperTerminal:

```
Empty flash at 0x00003ffc ends at 0x00004000 CLEANMARKER node found at 0x00004000
has totlen 0xc != normal 0x0  CLEANMARKER node found at 0x00008000 has totlen 0xc !=
normal 0x0
.............
.............
.............
.............
```

3.9 NOR Driver

3.9.1 Description

NOR is accessed via the asynchronous external memory interface (EMIFA). ARM ROM supports booting of the DM6446 ARM processor from NOR flash. The NOR driver is compliant with the standard MTD driver. The NOR driver creates the device nodes for user space access (/dev/mtdblock0, /dev/mtdblock1, etc.)

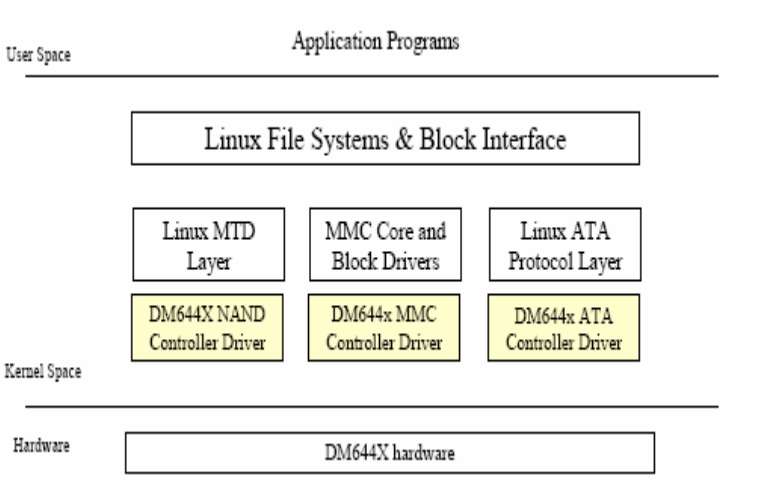


Figure 30. Linux Kernel NOR Driver

3.9.2 Driver Features

- Supports 16MB AMD AM29LV256M NOR Flash chip.
- Supports 16MB Intel I128P30T StrataFlash NOR Flash chip.
- Supports Read/Write/Erase from/to NOR Flash.
- NOR is tested with JFFS2 file system.
- NOR has been divided into 4 partitions:
    a. 128 KB Read/Write partition for U-Boot.
    b. 128 KB Read-only partition for environment variables.
    c. 4 MB Read/Write partition for Linux.
    d. Remaining space is used for file system and others (Read/Write).

3.9.3 Known Issues

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

3.9.4 Supported System Calls

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

3.9.5 DVEVM Board feature:

The DVEVM is equipped with one 6MB NOR Flash. It is connected through EMIF.

3.9.6 Performance

Refer to section 3.13.19 and 3.13.20 for NOR Driver performance data.

3.10 MMC/SD Driver

3.10.1 Description

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. It creates the device nodes for user space access (/dev/mmcblockp1, /dev/mmcblockp2, etc.).

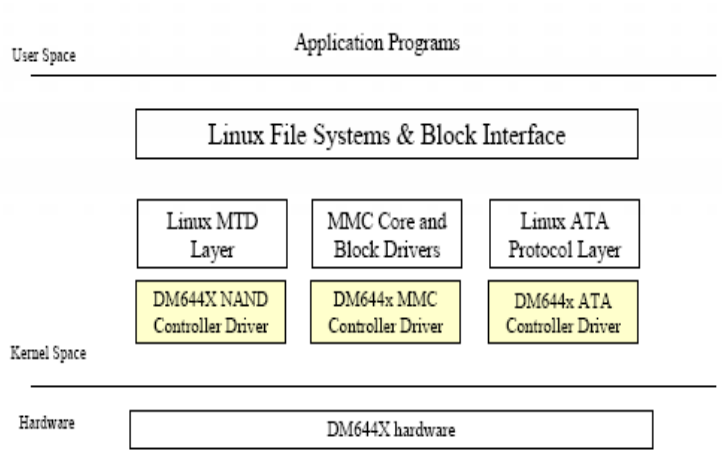


Figure 31. Linux Kernel MMC Driver

3.10.2 Driver Features

- DM644x MMC drivers are accessible as block device drivers
- MMC controller driver plugs into the Linux MMC core and block driver
- Linux filesystems and block devices layer abstract details of block devices (MMC).
- Mounting and unmounting filesystems, and associated filesystem creation/manipulation, is left to the user.
- Supports high-speed SD cards.

  NOTE: Some SD cards work fine at 50MHz. 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.
- 4-bit mode is supported.
- High-capacity SD support is not present in the driver.

3.10.3 Known Issues

- Due to MMC divisor restrictions, the init clock operates at 200KHz. At this frequency, some cards may not be recognized by Linux.
- Removing MMC module with a mounted file system succeeds.

- Removing SD module with a mounted file system succeeds.

3.10.4 Supported System Calls

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

3.10.5 DVEVM Board

- DVEVM supports MMC on the peripheral interface. There is a MMC connector on the baseboard.

3.10.6 Menu Configuration: To enable the MMC

```
Device Drivers à
          MMC/SD Card Support à
               <*> MMC support
               <*> MMC Block device driver
               <*> TI DAVINCI Multimedia Card Interface support
```

3.10.7 Benchmarks

The EXT2 file system is used to read a file from the MMC and write a file into MMC. The runtime performance is described below.

Table 35. MMC Memory Statistics

Configuration	Memory Statistics[1]				
	Program Memory	Data Memory			Total
		Initialized	Uninitialized	Stack	
davinci_mmc.ko	8644	616	212		9472
mmc_block.ko	5468	424	8		5900
mmc_core.ko	10844	732	4		11580

[1] All memory requirements are expressed in bytes for **full-featured** device driver usage.

3.10.8 Performance: MMC File Write

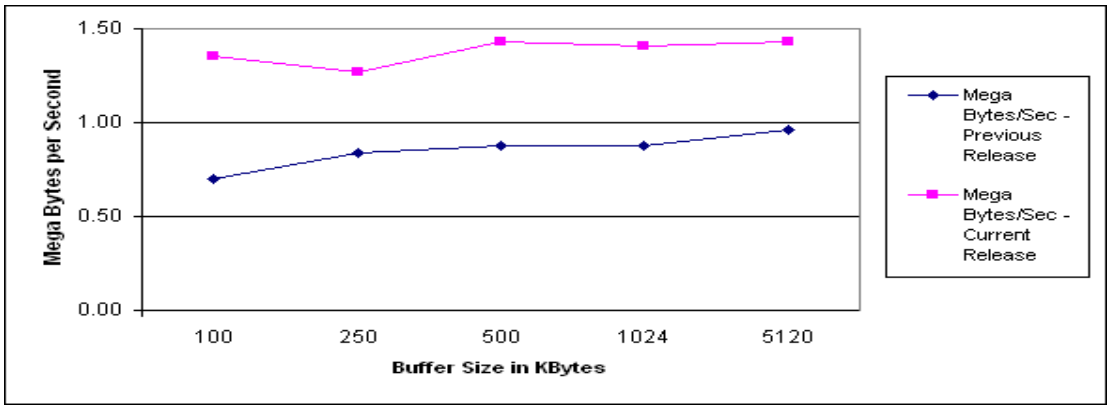


Figure 32. Runtime Performance Characteristics – MMC_FileWrite_EXT2_283MHZ

Table 36. Test Data

Test Setup Information	Buffer Size in KBytes	DMA Mode			
		Mega Bytes/Sec - Previous Release	Mega Bytes/Sec - Current Release	File Size in MB	Duration
ARM Frequency = 283 MHZ	100	0.70	1.35	100	74
	250	0.84	1.27	100	79
	500	0.88	1.43	100	70
	1024	0.88	1.41	100	71
	5120	0.96	1.43	100	70

3.10.8.1 Comments

- MMC Card used for Testing: Kingston
- Mode Tested: Default Mode (DMA mode)
- File system Used: EXT2 File System

3.10.8.2 Theoretical Performance Values

- Maximum Data Transfer Rate Supported = 2.5 Mbytes/sec

3.10.9 Performance: MMC File Read

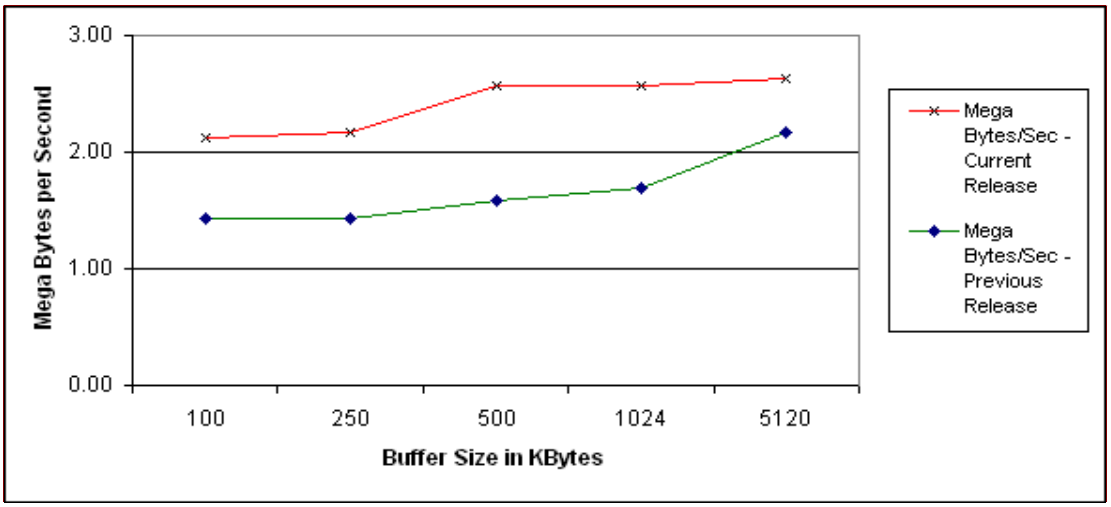


Figure 33. Runtime Performance Characteristics – MMC_FileRead_EXT2_283MHZ

Table 37. Test Data

Test Setup Information	Buffer Size in Kbytes	DMA Mode			
		Mega Bytes/Sec - Previous Release	Mega Bytes/Sec - Current Release	File Size in MB	Duration
ARM Frequency = 283 MHZ	100	1.43	2.13	100	47
	250	1.43	2.17	100	46
	500	1.59	2.56	100	39
	1024	1.69	2.56	100	39
	5120	2.17	2.63	100	38

3.10.9.1 Comments

- MMC Card used for Testing: Kingston 128MB
- Mode Tested: Default Mode (DMA Mode)
- File system Used: EXT2 File System

3.10.9.2 Theoretical Performance Values

- Maximum Data Transfer Rate Supported = 2.5 Mbytes/sec

3.10.9.3 Notes

- The MMC/SD peripheral does not support the SPI mode of operation.

3.11 PWM Driver

3.11.1 Description

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 PWM driver is implemented as a char driver. It can be accessed from user space as /dev/davinci_pwm0, /dev/davinci_pwm1 and /dev/davinci_pwm2.

3.11.2 Driver Features

- The PWM driver has been implemented as a char driver.
- The driver works only in interrupt mode. DMA mode is not supported.
- This driver supports all three instances of PWM.
- By default, the PWM driver has been configured in one-shot mode.

3.11.3 Known Issues

- 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.
- PWM output in one-shot mode is inconsistent.

NOTE: If PH1D value is 0, the first phase has zero time and the opposite of the first-phase output value is driven on PWM for the entire period. If PH1D is greater than or equal to (PER + 1), the first phase is 100% of the period and the P1OUT output level is sent for the duration of the period.

3.11.4 Menu Configuration: PWM Driver

```
Device Drivers à
Character Devices à
          …….
          …….
          …….
          < > DaVinci Resizer Driver
          < > DaVinci Previewer Driver Support
          <*> DaVinci PWM Driver Support
```

3.12 SDIO WLAN Driver

3.12.1 Description

Go through the below link for the details about the SDIO Linux stack Information.

http://downloads.sourceforge.net/sdio-linux/sdio-linux-documentation.tar.gz?modtime=1168190237&big_mirror=0

3.12.2 Driver Features

The driver supports the following features.

- 802.11b and 802.11g wireless standards.
- Infrastructure (Manage) and Ad-Hoc mode.
- Configuration of wireless parameters using iwconfig.
- Roaming between AP's.
- WEP Key encryption based communication.

3.12.3 Known issues

The driver doesn't support the following features.

- Statistics of different kinds of packets like multicast, broadcast and different kinds of error packets.
- iwlist command with parameters scan, event, access points and peers
- Joining networks with out specifying an explicit ESSID.
- Configurations of ready to send, fragmentation, transmit power and module power with iwconfig.

3.12.4 Constraints

- Wireless parameters can not be set after setting ESSID.

3.12.5 Supported System Calls

None

3.12.6 Supported IOCTLS

None

3.12.7 Performance

Mode	Up-Link	Downlink
Infrastructure- Mixed	10.25	6.63
Infrastructure- 802.11g only	10.67	7.95
Infrastructure- 802.11b only	2.65	2.77
Ad-Hoc	12.13	10.16

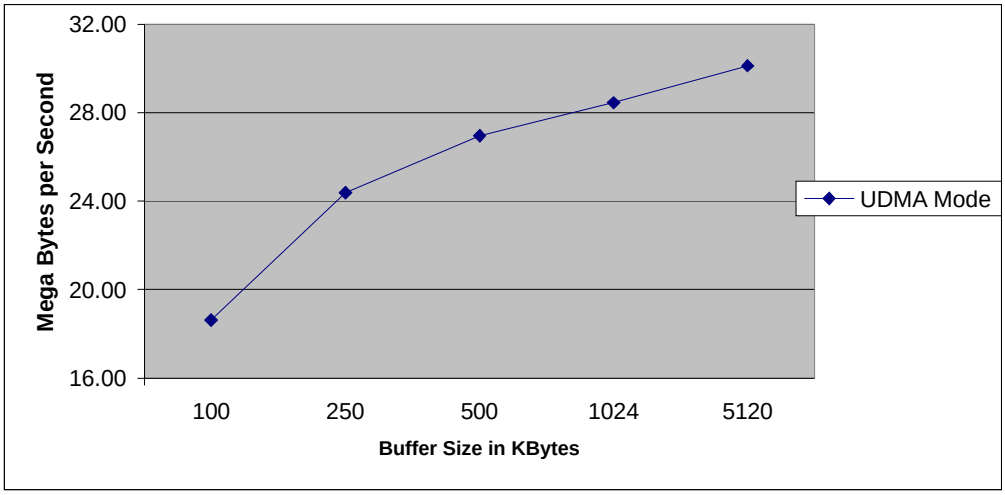
Table 38. WLAN Performance

3.12.7.1 Comments

- Infrastructure Mode: Linksys Wireless-G Access point model WAP54G. Channel 11 used for testing.
- Ad-Hoc Mode : Dlink Airplus G DWL-G510 Wireless PCI adapter plugged in desktop PC used as peer.
- All the tests were run using ttcp utility version 1.10 on Linux side and PCAUSA Test TCP Utility V2.01.01.08 running on WinXP side.

3.13 Additional Driver Performance Reports

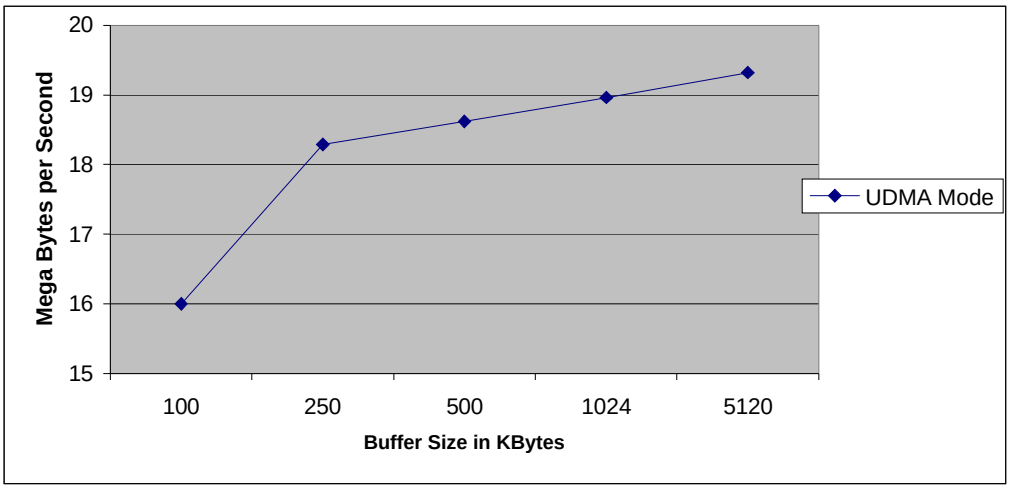
3.13.1 ATA – Write Performance - Patch Level 004 release



Test Setup Information	Buffer Size in KBytes	UDMA Mode		
		Mega Bytes/Sec	FileSize in MB	Duration in Sec
ARM Frequncy = 297 MHZ	100	18.62	1024	55
	250	24.38	1024	42
	500	26.95	1024	38
	1024	28.44	1024	36
	5120	30.12	1024	34

Comments:
HDD used for Testing :- TOSHIBA MK4019GAX
Mode Tested :- Auto Mode (UDMA 4)
Filesystem Used :-EXT2 FileSystem
Theoretical Performance Values:
I/O Data Transfer Rate(Mbytes/sec max) 66.6Mbytes/sec (UDMA4)

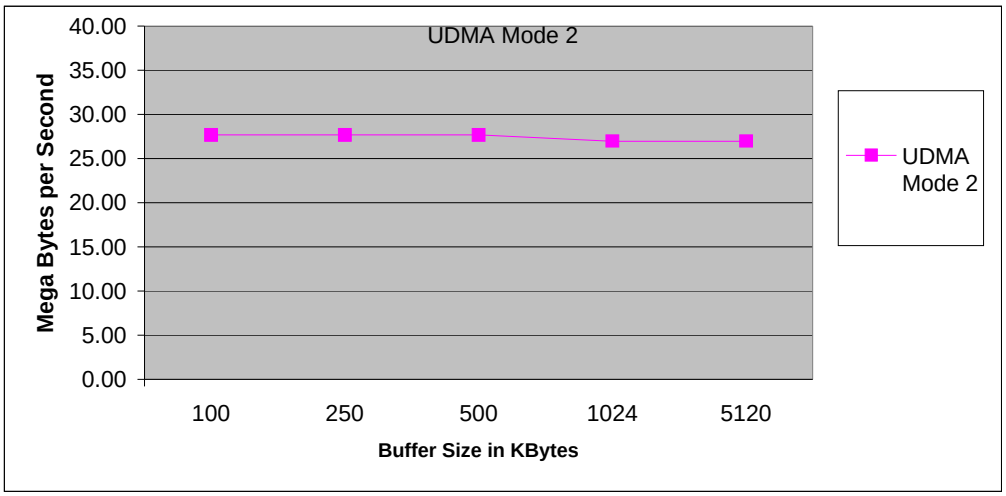
3.13.2 ATA – Read Performance - Patch Level 004 release



Test Setup Information	Buffer Size in Kbytes	UDMA Mode		
		Mega Bytes/Sec	FileSize in MB	Duration in Sec
ARM Frequncy = 297 MHZ	100	16.00	1024	64
	250	18.29	1024	56
	500	18.62	1024	55
	1024	18.96	1024	54
	5120	19.32	1024	53

Comments:
HDD used for Testing :- TOSHIBA MK4019GAX
Mode Tested :- Auto Mode (UDMA 4)
Filesystem Used :- EXT2 FileSystem
Theoretical Performance Values:
I/O Data Transfer Rate(Mbytes/sec max) 66.6Mbytes/sec (UDMA4)

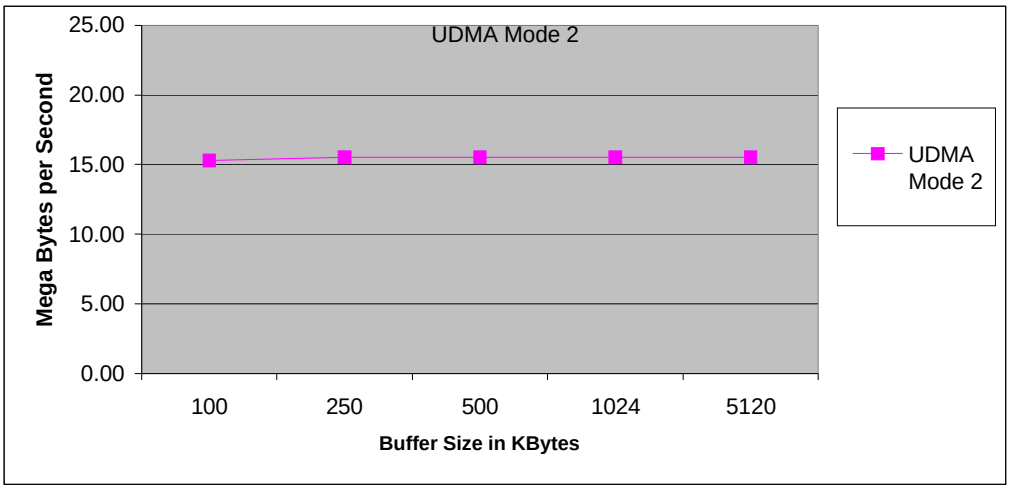
3.13.3 ATA – Write Performance - Patch Level 026 release



Test Setup Information	Buffer Size in KBytes	UDMA Mode 2		
		Mega Bytes/Sec	FileSize in MB	Duration in Sec
ARM Frequncy = 297 MHZ	100	27.68	1024	37
	250	27.68	1024	37
	500	27.68	1024	37
	1024	26.95	1024	38
	5120	26.95	1024	38

Comments:
HDD used for Testing :- TOSHIBA MK4019GAX
Filesystem Used :-EXT2 FileSystem

3.13.4 ATA – Read Performance - Patch Level 026 release



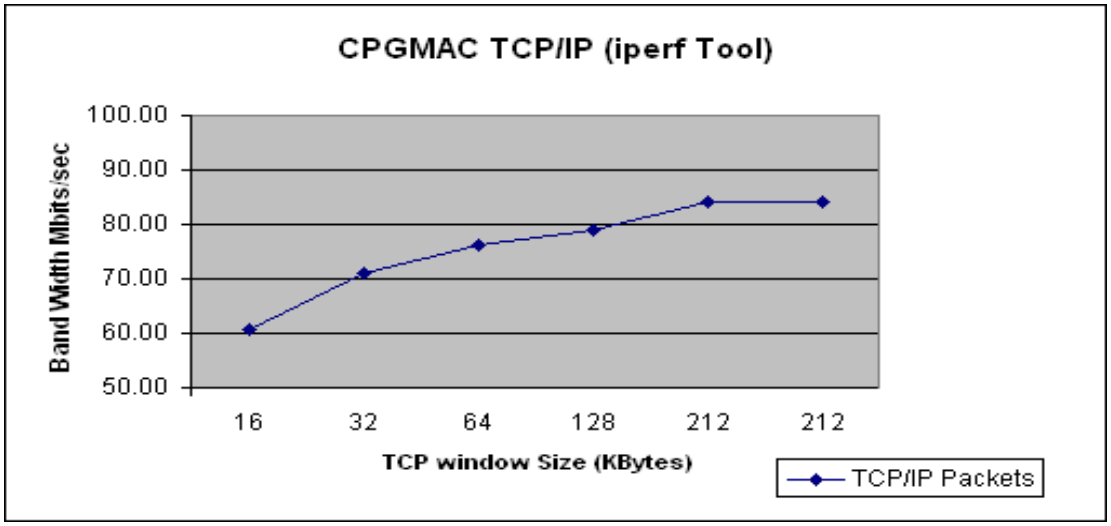
Test Setup Information	Buffer Size in Kbytes	UDMA Mode 2		
		Mega Bytes/Sec	FileSize in MB	Duration in Sec
ARM Frequncy = 297 MHZ	100	15.28	1024	67
	250	15.52	1024	66
	500	15.52	1024	66
	1024	15.52	1024	66
	5120	15.52	1024	66

Comments:

HDD used for Testing :- TOSHIBA MK4019GAX

Filesystem Used :- EXT2 FileSystem

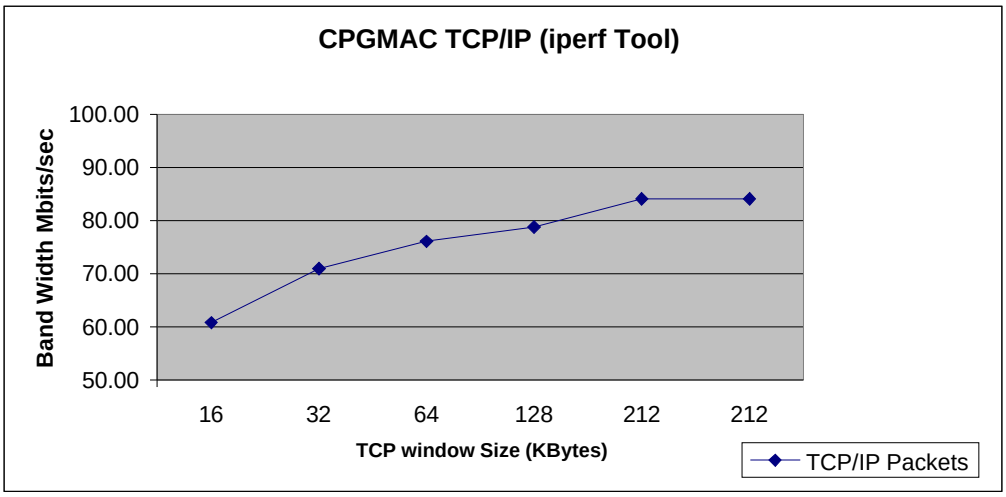
3.13.5 CPGMAC – IPERF_Client - Patch Level 004 Release



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	16	60.80	435	60
	32	71.00	508.00	60
	64	76.10	545.00	60
	128	78.80	564.00	60
	212	84.10	602.00	60
	212	84.10	602.00	60

Comments:
1. The CPGMAC is configured for Auto negotiation, Link is ->100Mbps Half duplex.
2. "iperf" tool is run on DUT1 in server mode and on DUT2 in Client mode
3. Both the DUT and the PC were on same switch
4. iperf version 1.7.0 was used for the tests
5. The Data captured here is for "iperf" in Client mode.
6. "iperf" tool is run on DUT1 and DUT2 with the NFS mounted.

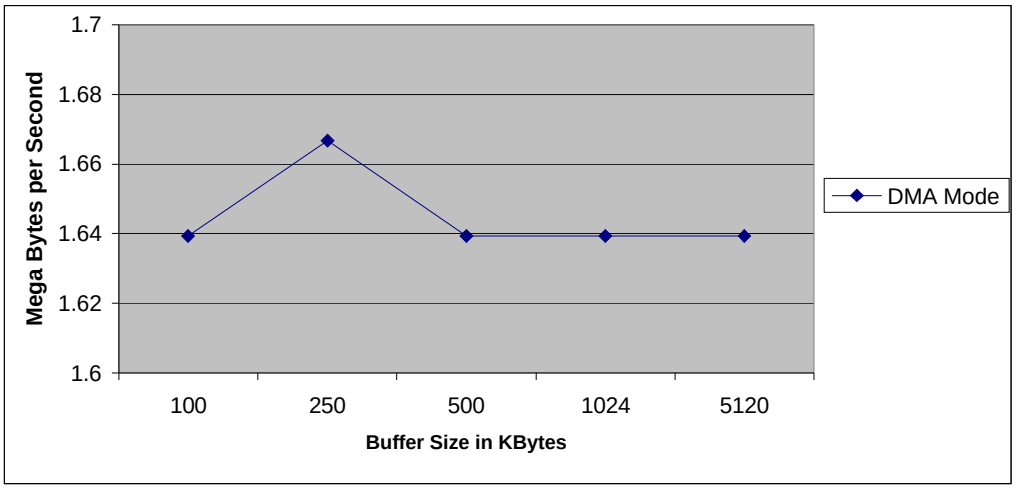
3.13.6 CPGMAC – IPERF_Server - Patch Level 004 Release



Test Setup Information	TCP window size in KBytes	TCP/IP Packets		
		BandwidthMbits/Sec	Transfer size in MBytes	Interval in sec
ARM Frequency = 297 MHZ	16	60.80	435	60
	32	71.00	508	60
	64	76.10	545	60
	128	78.80	564	60
	212	84.10	602	60
	212	84.10	602	60

Comments:
1. The CPGMAC is configured for Auto negotiation, Link is -> 100Mbps Half duplex.
2. "iperf" tool is run on DUT1 in server mode and on DUT2 in Client mode
3. Both the DUT and the PC were on same switch
4. iperf version 1.7.0 was used for the tests
5. The Data captured here is for "iperf" in Server mode.
6. "iperf" tool is run on DUT1 and DUT2 with the NFS mounted.

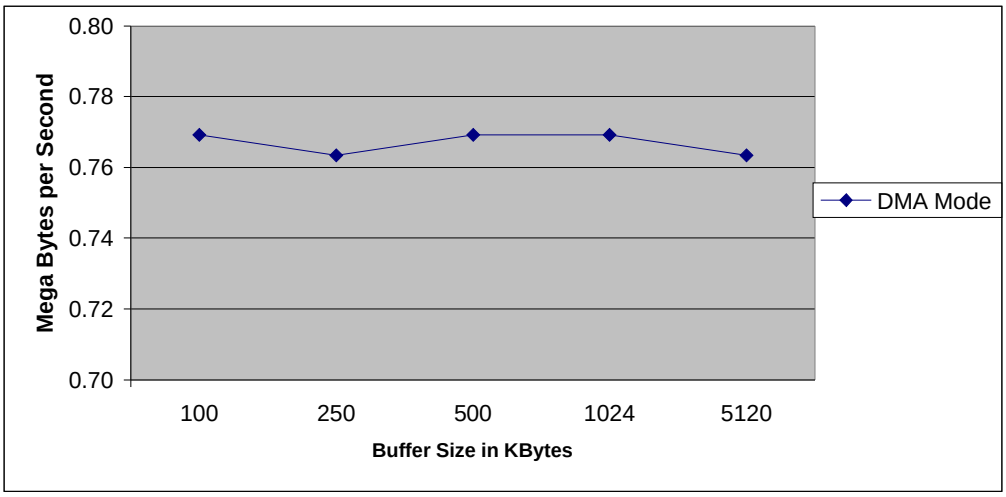
3.13.7 MMC – Read – Patch Level 004 Release



Test Setup Information	Buffer Size in Kbytes	DMA Mode		
		Mega Bytes/Sec	FileSize in MB	Duration
ARM Frequncy = 297 MHZ	100	1.64	100	61
	250	1.67	100	60
	500	1.64	100	61
	1024	1.64	100	61
	5120	1.64	100	61

Comments:
MMC Card used for Testing :- Kingston 128MB
Mode Tested :- Default Mode (DMA Mode)
Filesystem Used :- EXT2 FileSystem
Theoretical Performance Values:
Maximum Data Transfer Rate Supported 2.5 Mbytes/sec

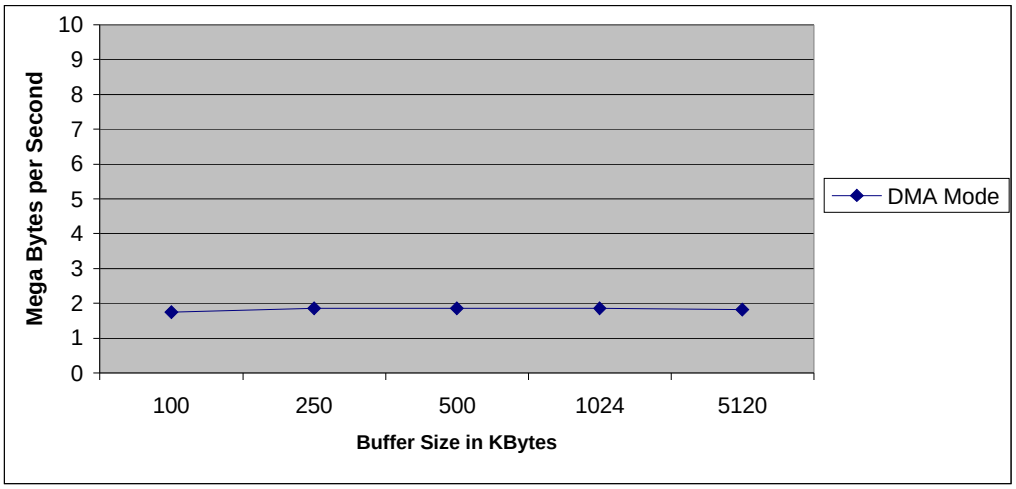
3.13.8 MMC – Write – Patch Level 004 Release



Test Setup Information	Buffer Size in KBytes	DMA Mode		
		Mega Bytes/Sec	FileSize in MB	Duration
ARM Frequncy = 297 MHZ	100	0.77	100	130
	250	0.76	100	131
	500	0.77	100	130
	1024	0.77	100	130
	5120	0.76	100	131

Comments:
MMC Card used for Testing :- Kingston 128MB
Mode Tested :- Default Mode (DMA mode)
Filesystem Used :- EXT2 FileSystem
Theoretical Performance Values:
Maximum Data Transfer Rate Supported 2.5 Mbytes/sec

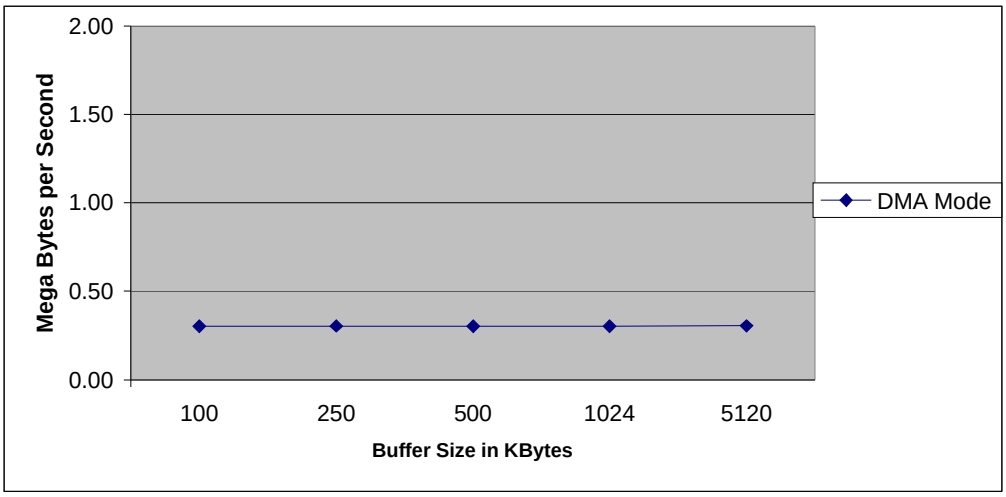
3.13.9 MMC – Read – Patch Level 026 Release



Test Setup Information	Buffer Size in Kbytes	DMA Mode		
		Mega Bytes/Sec	FileSize in MB	Duration
ARM Frequncy = 297 MHZ	100	1.75	100	57
	250	1.85	100	54
	500	1.85	100	54
	1024	1.85	100	54
	5120	1.82	100	55

Comments:
MMC Card used for Testing :- SanDisk MMC 1GB Card
Mode Tested :- Default Mode (DMA Mode)
Filesystem Used :- EXT2 FileSystem
Theoretical Performance Values:
Maximum Data Transfer Rate Supported 2.5 Mbytes/sec

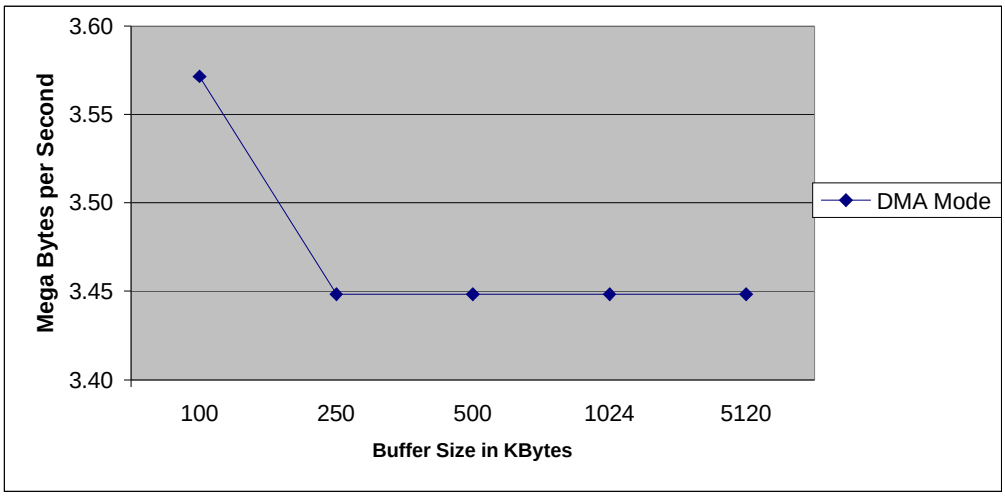
3.13.10 MMC – Write – Patch Level 026 Release



Test Setup Information	Buffer Size in KBytes	DMA Mode		
		Mega Bytes/Sec	FileSize in MB	Duration
ARM Frequncy = 297 MHZ	100	0.30	100	332
	250	0.30	100	329
	500	0.30	100	331
	1024	0.30	100	331
	5120	0.30	100	328

Comments:
MMC Card used for Testing :- SanDisk MMC 1GB Card
Mode Tested :- Default Mode (DMA mode)
Filesystem Used :- EXT2 FileSystem
Theoretical Performance Values:
Maximum Data Transfer Rate Supported 2.5 Mbytes/sec

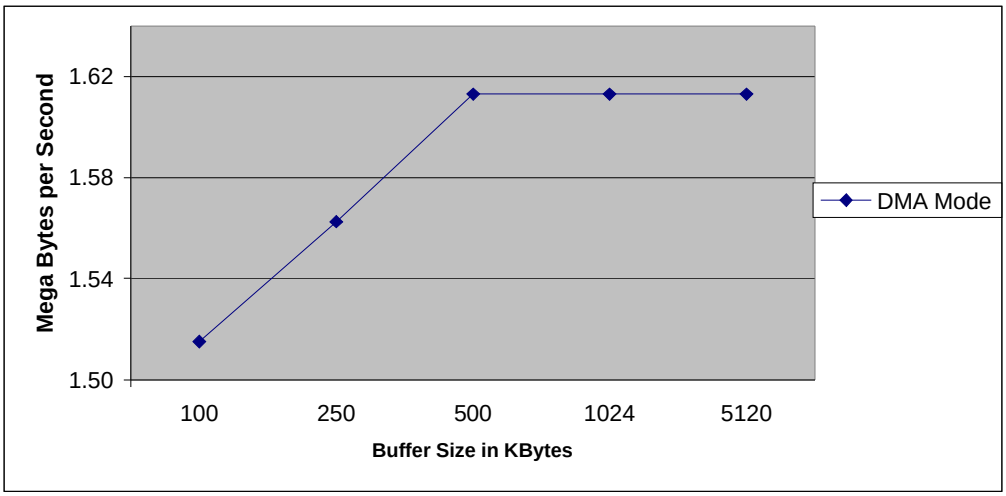
3.13.11 NAND – Write – Patch Level 004 Release



Test Setup Information	Buffer Size in KBytes	DMA Mode		
		Mega Bytes/Sec	FileSize in MB	Duration
ARM Frequncy = 297 MHZ	100	3.57	100	28
	250	3.45	100	29
	500	3.45	100	29
	1024	3.45	100	29
	5120	3.45	100	29

Comments:
NAND Device Tested:- Samsung K9K1208Q0C
Mode Tested :- DMA Mode
FileSystem used :- JFFS2
Theoretical Performance Values:
Write Speed : 6.9 Mbytes/sec (For Small Block NAND)

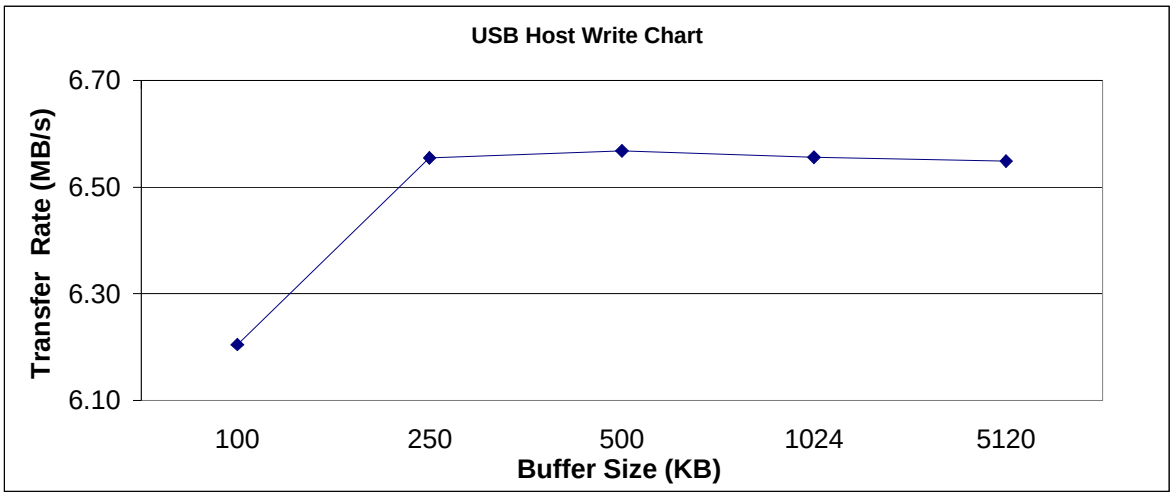
3.13.12 NAND – Read – Patch Level 004 Release



Test Setup Information	Buffer Size in KBytes	DMA Mode		
		Mega Bytes/Sec	FileSize in MB	Duration
ARM Frequncy = 297 MHZ	100	1.52	50	33
	250	1.56	50	32
	500	1.61	50	31
	1024	1.61	50	31
	5120	1.61	50	31

Comments:
NAND Device Tested :- Samsung K9K1208Q0C
Mode Tested :- DMA Mode
FileSystem used :- JFFS2
Theoretical Performance Values:
Read Speed : 16.1 Mbytes/sec (For Small Block NAND)

3.13.13 USB HOST – Write – Patch Level 004 Release

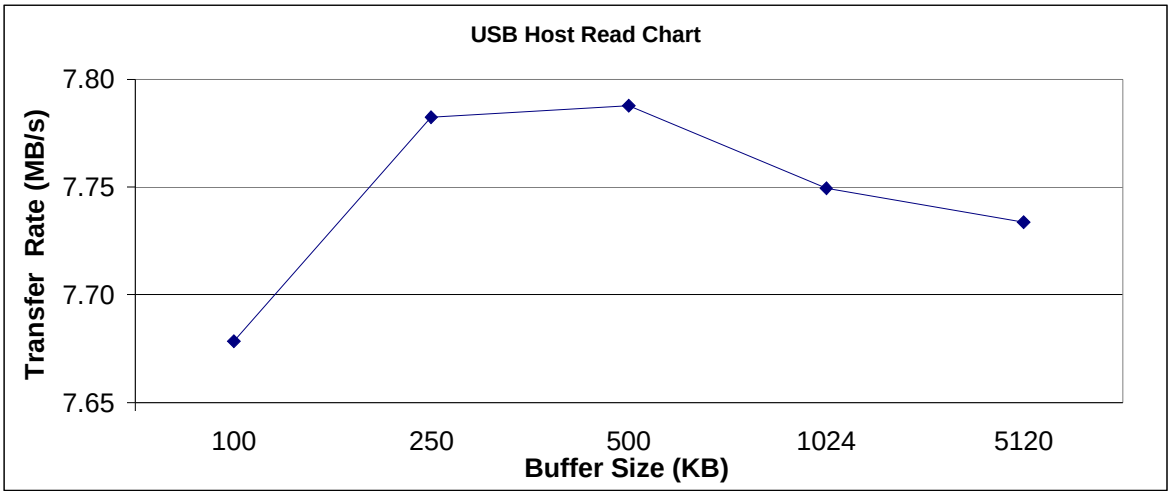


Test Setup Information	USB Host Write Results		
	Buffer size (KB)	Total Bytes Transferred (GB)	Current Transfer Rate (MB/s)
ARM Frequency = 297Mhz	100	1	6.20
Mode = DMA	250	1	6.56
Low Latency Pre-emption mode	500	1	6.57
FAT32 formatted Pen Drive	1024	1	6.56
Silicon: 1.3 (Beta EVM)	5120	1	6.55

Remarks:

1. The performance is measured using gettimeofday() function.
2. Pen drive - Sandisk (1GB) is used for writing files.
3. The performance time is not too accurate since it involves loop time and time is taken from gettimeofday() function.

3.13.14 USB HOST – Read – Patch Level 004 Release

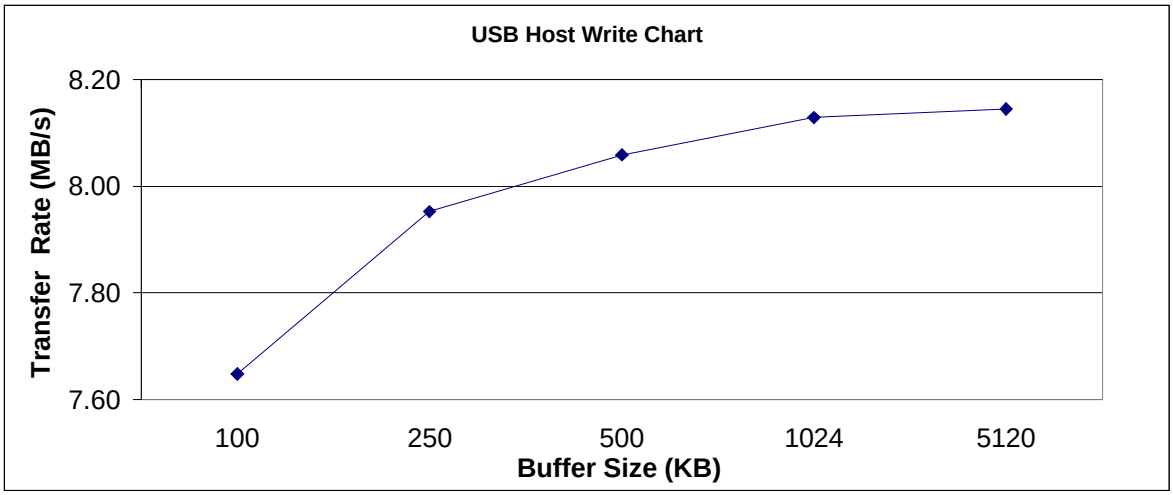


Test Setup Information	USB Host Read Results		
	Buffer size (KB)	Total Bytes Transferred (GB)	Current Transfer Rate (MB/s)
ARM Frequency = 297Mhz	100	1	7.68
Mode = DMA	250	1	7.78
Low Latency Pre-emption mode	500	1	7.79
FAT32 formatted Pen Drive	1024	1	7.75
Silicon: 1.3 (Beta EVM)	5120	1	7.73

Remarks:

1. The performance is measured using gettimeofday() function.
2. Pen Drive - Sandisk (1GB) is used for reading files.
3. The performance time is not too accurate since it involves loop time and time is taken from gettimeofday() function.

USB HOST – Write – Patch Level 014 Release

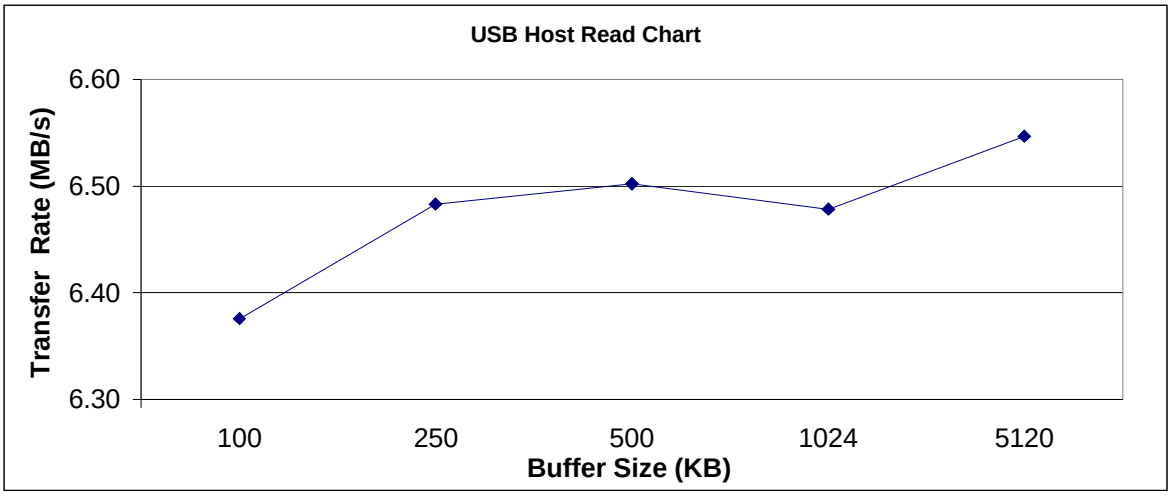


Test Setup Information	USB Host Write Results		
	Buffer size (KB)	Total Bytes Transferred (GB)	Current Transfer Rate (MB/s)
ARM Frequency = 297Mhz	100	1	7.65
Mode = DMA	250	1	7.95
Low Latency Pre-emption mode	500	1	8.06
FAT32 formatted Pen Drive	1024	1	8.13
Silicon: 1.3 (Delta EVM)	5120	1	8.14

Remarks:

1. The performance is measured using gettimeofday() function.
2. Pen drive - Lexar Jumbo drive (1GB) is used for writing files.
3. The performance time is not too accurate since it involves loop time and time is taken from gettimeofday() function.

3.13.15 USB HOST – Read – Patch Level 014 Release



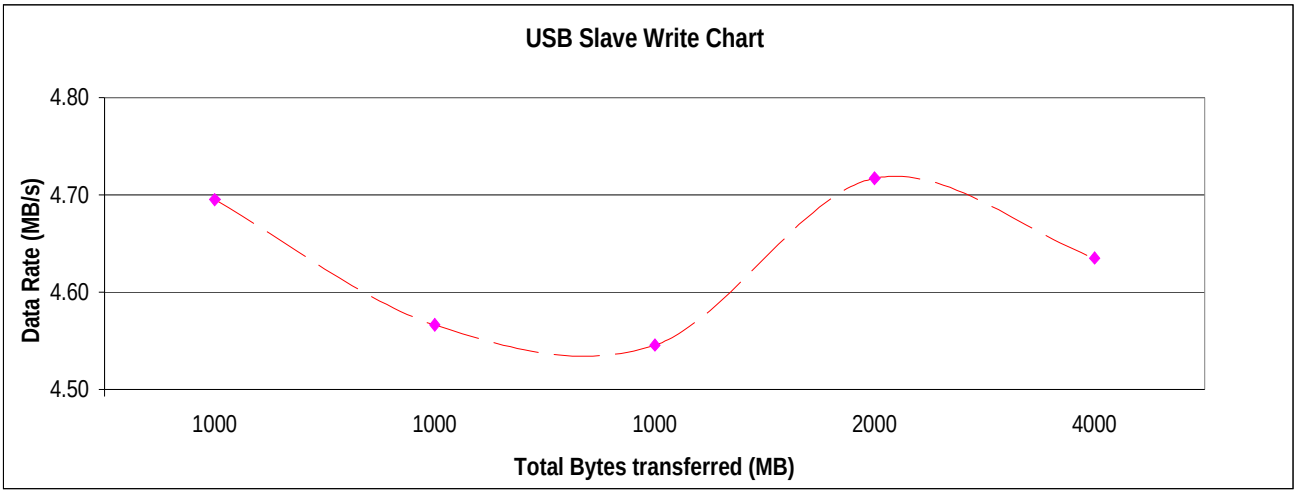
Test Setup Information	USB Host Read Results		
	Buffer size (KB)	Total Bytes Transferred (GB)	Current Transfer Rate (MB/s)
ARM Frequency = 297Mhz	100	1	6.38
Mode = DMA	250	1	6.48
Low Latency Pre-emption mode	500	1	6.50
FAT32 formatted Pen Drive	1024	1	6.48
Silicon: 1.3 (Delta EVM)	5120	1	6.55

Remarks:

1. The performance is measured using gettimeofday() function.
2. Pen Drive - Lexar Jumbo drive (1GB) is used for reading files.
3. The performance time is not too accurate since it involves loop time and time is taken from gettimeofday() function.

3.13.16 USB SLAVE – Write – Patch Level 004 Release

USB Slave Write Performance Results

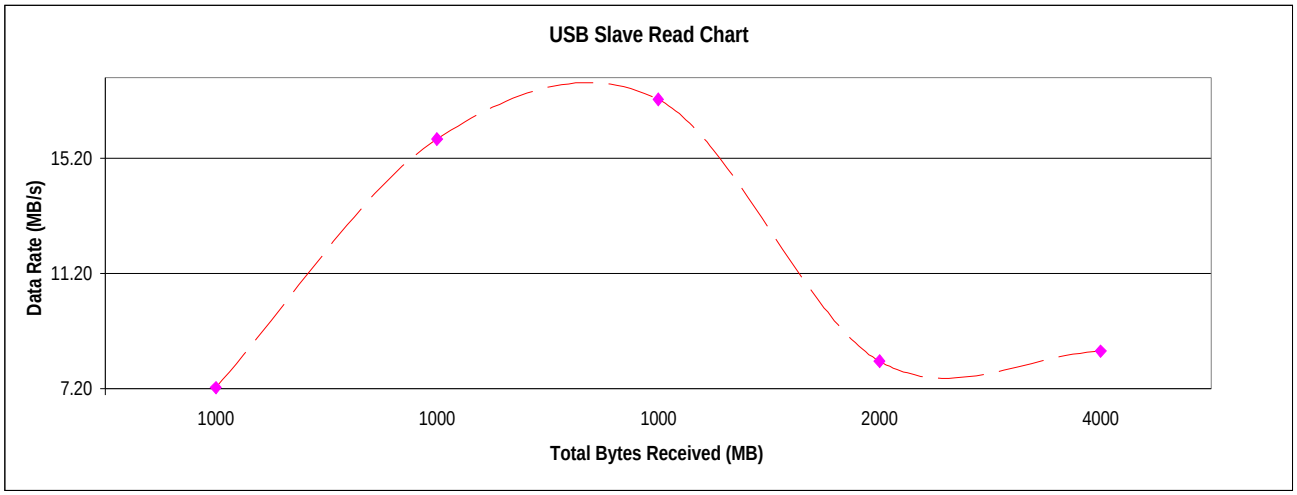


Test Setup Information	USB Slave Write Results			
	Bytes Transferred (MB)	Number of files	Total Bytes Transferred (MB)	Data Rate (MB/sec)
ARM Frequency = 297MHz	50	20	1000	4.69
Mode = DMA	10	100	1000	4.57
Low Latency Pre-emption mode	5	200	1000	4.55
CPU- 2.8GHz, 512MB RAM, WinXP (SP2)	100	20	2000	4.72
1.3 Silicon (Gamma Board)	200	20	4000	4.63

Remarks:

1. The performance is measured using Windows XP PC.

USB SLAVE – Read – Patch Level 004 Release

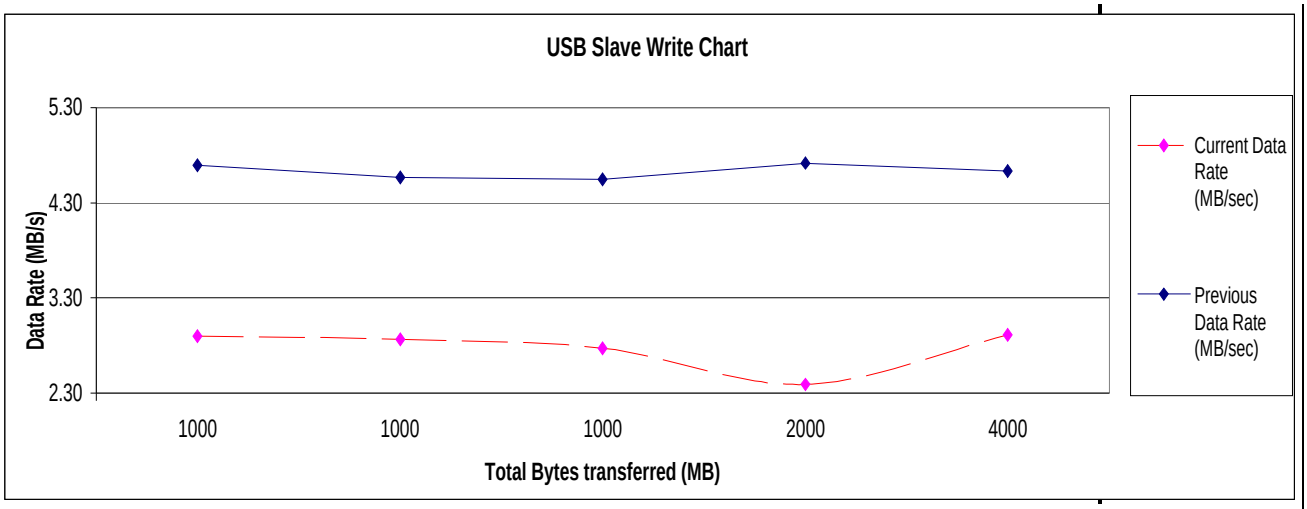


Test Setup Information	USB Slave Read Results			
	Bytes Transferred (MB)	Number of files	Total Bytes Received (MB)	Data Rate (MB/sec)
ARM Frequency = 297Mhz	50	20	1000	7.25
Mode = DMA	10	100	1000	15.87
Low latency Pre-emption mode	5	200	1000	17.24
CPU- 2.8GHz, 512MB RAM, WinXP (SP2)	100	20	2000	8.16
1.3 Silicon (Gamma Board)	200	20	4000	8.51

Remarks:

1. The performance is measured using Windows XP PC.
2. Hard Disk - Toshiba (40GB) is used for reading files.
3. The performance time is not too accurate since it involves loop time and time is taken from Windows XP PC.

3.13.17 USB SLAVE – Write – Patch Level 014 Release

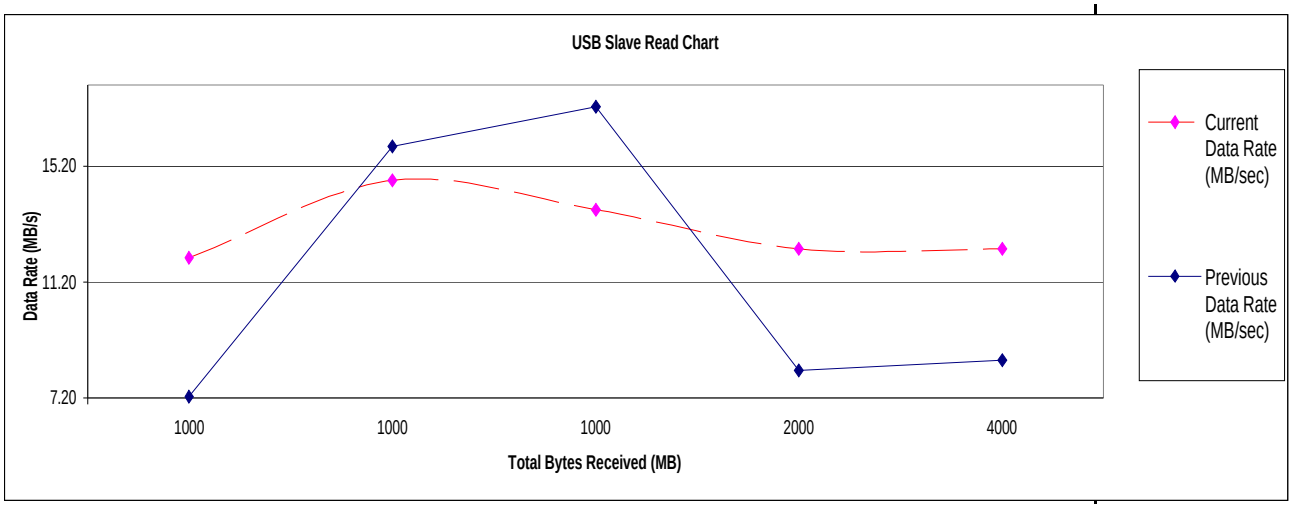


Test Setup Information	USB Slave Write Results				
	Bytes Transferred (MB)	Number of files	Total Bytes Transferred (MB)	Previous Data Rate (MB/sec)	Current Data Rate (MB/sec)
ARM Frequency = 297MHz	50	20	1000	4.69	2.90
Mode = DMA	10	100	1000	4.57	2.87
Low Latency Pre-emption mode	5	200	1000	4.55	2.77
	100	20	2000	4.72	2.39
1.3 Silicon (Beta Board)	200	20	4000	4.63	2.91

Remarks:

1. The performance is measured using Windows XP PC.
2. Hard Disk - Toshiba (40GB) is used for writing files.
3. Previous CPU Info: CPU - 2.8GHz, 512MB RAM, WinXP (SP2), Previous CPU Info: CPU - 3 GHz, 1GB RAM, WinXP (SP2)
4. The performance time is not too accurate since it involves loop time and time is taken from Windows XP PC.

3.13.18 USB SLAVE – Read – Patch Level 014 Release

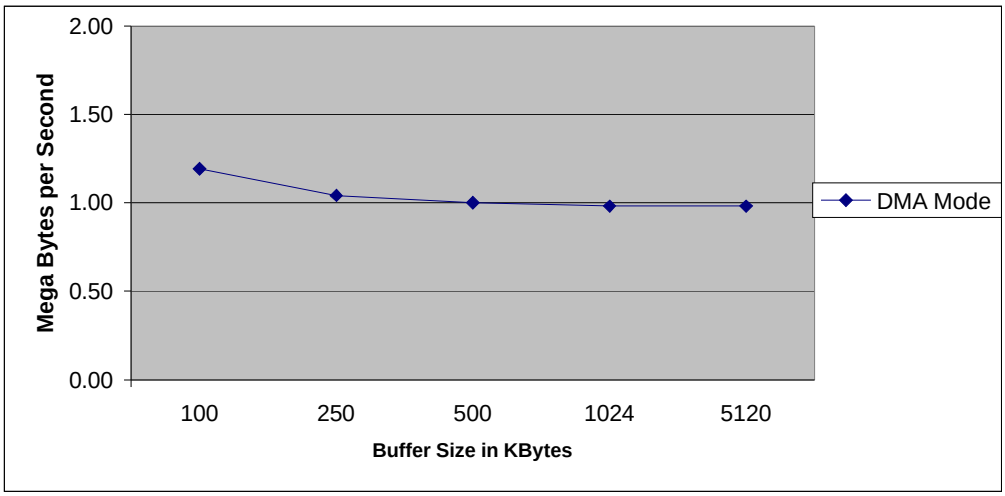


Test Setup Information	USB Slave Read Results				
	Bytes Transferred (MB)	Number of files	Total Bytes Received (MB)	Previous Data Rate (MB/sec)	Current Data Rate (MB/sec)
ARM Frequency = 297Mhz	50	20	1000	7.25	12.05
Mode = DMA	10	100	1000	15.87	14.71
Low latency Pre-emption mode	5	200	1000	17.24	13.70
	100	20	2000	8.16	12.35
1.3 Silicon (Beta Board)	200	20	4000	8.51	12.35

Remarks:

1. The performance is measured using Windows XP PC.
2. Hard Disk - Toshiba (40GB) is used for reading files.
3. Previous CPU Info: CPU - 2.8GHz, 512MB RAM, WinXP (SP2), Previous CPU Info: CPU - 3 GHz, 1GB RAM, WinXP (SP2)
4. The performance time is not too accurate since it involves loop time and time is taken from Windows XP PC.

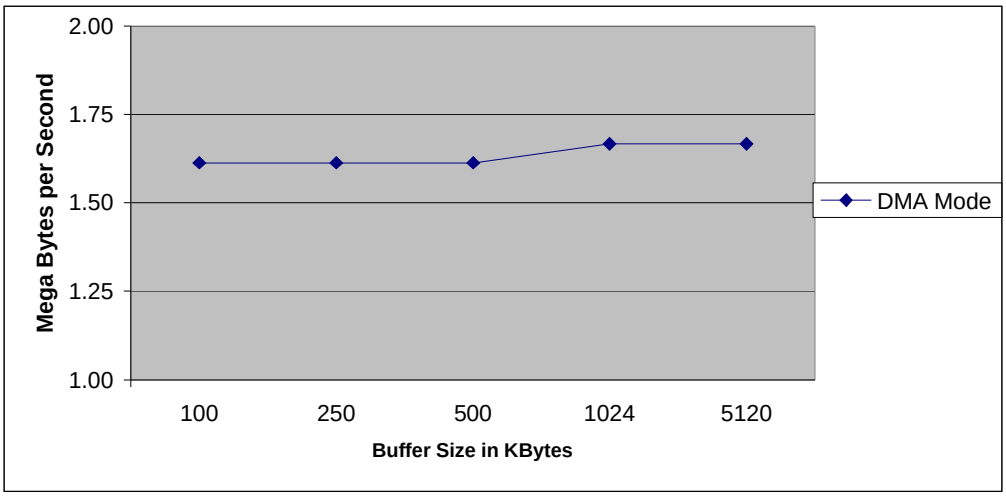
3.13.19 NOR – Write – Patch Level 023 Release



Test Setup Information	Buffer Size in KBytes	DMA Mode		
		Mega Bytes/Sec	FileSize in MB	Duration
ARM Frequncy = 297 MHZ	100	1.19	100	84
	250	1.04	100	96
	500	1.00	100	100
	1024	0.98	100	102
	5120	0.98	100	102

Comments:
NAND Device Tested:- AM29LV256M
FileSystem used :- JFFS2
Theoretical Performance Values:
Write Speed : 2.13 Mbytes/sec ()

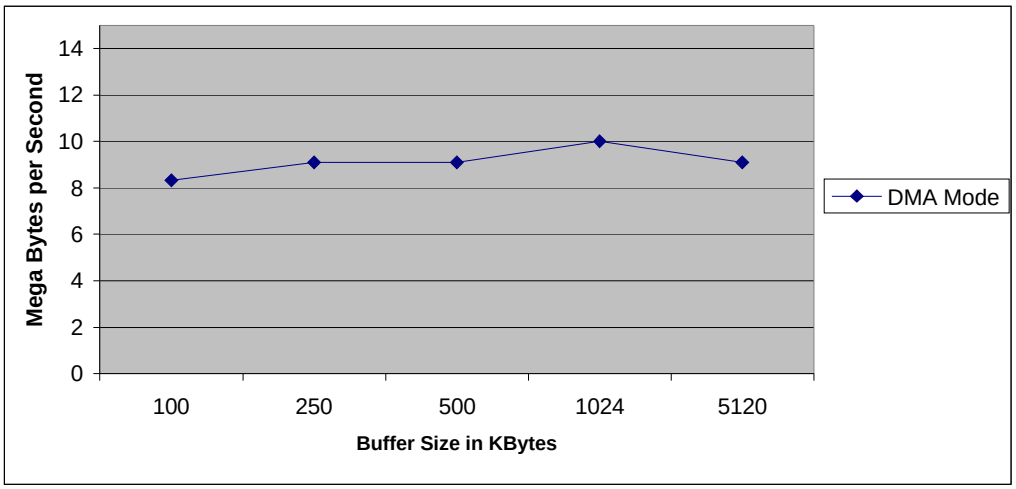
3.13.20 NOR – Read – Patch Level 023 Release



Test Setup Information	Buffer Size in KBytes	DMA Mode		
		Mega Bytes/Sec	FileSize in MB	Duration
ARM Frequncy = 297 MHZ	100	1.61	50	31
	250	1.61	50	31
	500	1.61	50	31
	1024	1.67	50	30
	5120	1.67	50	30

Comments:
NAND Device Tested :- AM29LV256M
FileSystem used :- JFFS2
Theoretical Performance Values:
Read Speed : 16.1 Mbytes/sec ()

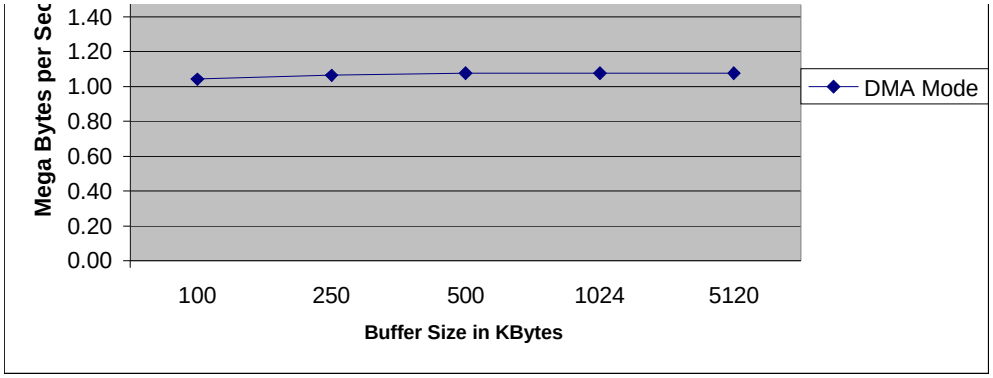
3.13.21 SD – Read – Patch Level 026 Release



Test Setup Information	Buffer Size in Kbytes	DMA Mode		
		Mega Bytes/Sec	FileSize in MB	Duration
ARM Frequncy = 297 MHZ	100	8.33	100	12
	250	9.09	100	11
	500	9.09	100	11
	1024	10.00	100	10
	5120	9.09	100	11

Comments:
SD Card used for Testing :- ExtremeIII 1GB Card
Mode Tested :- Default Mode (DMA Mode)
Filesystem Used :- EXT2 FileSystem

3.13.22 SD – Write – Patch Level 026 Release



Test Setup Information	Buffer Size in KBytes	DMA Mode		
		Mega Bytes/Sec	FileSize in MB	Duration
ARM Frequncy = 297 MHZ	100	1.04	100	96
	250	1.06	100	94
	500	1.08	100	93
	1024	1.08	100	93
	5120	1.08	100	93

Comments:
SD Card used for Testing: ExtremeIII 1GB Card
Mode Tested: Default Mode (DMA mode)
Filesystem Used: EXT2 FileSystem

3.13.23 MMC File Write- Low Latency - Patch Level 045 Release

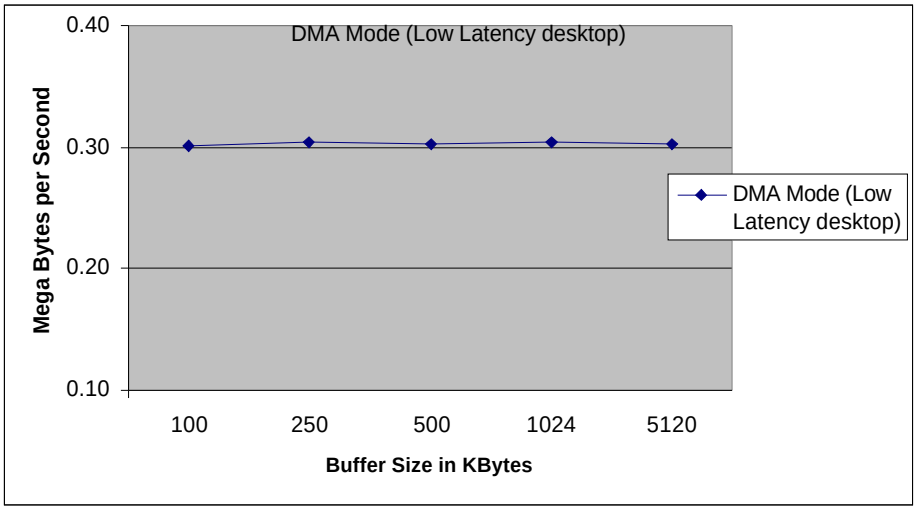


Figure 34. Runtime Performance Characteristics – MMC_File Write

Test Setup Information	Buffer Size in KBytes	DMA Mode		
		Mega Bytes/Sec	File Size in MB	Duration
ARM Frequency= 297 Mhz	100	0.30	100	331.92
	250	0.30	100	329.19
	500	0.30	100	330.47
	1024	0.30	100	329.59
	5120	0.30	100	330.36

Table 39. MMC File Write

3.13.23.1 Comments

- MMC Card used for Testing: San Disk RS-MMC 1GB Card
- File system Used: EXT2 File System
- Mode : DMA

3.13.24 MMC File Read-Low Latency - Patch Level 045 Release

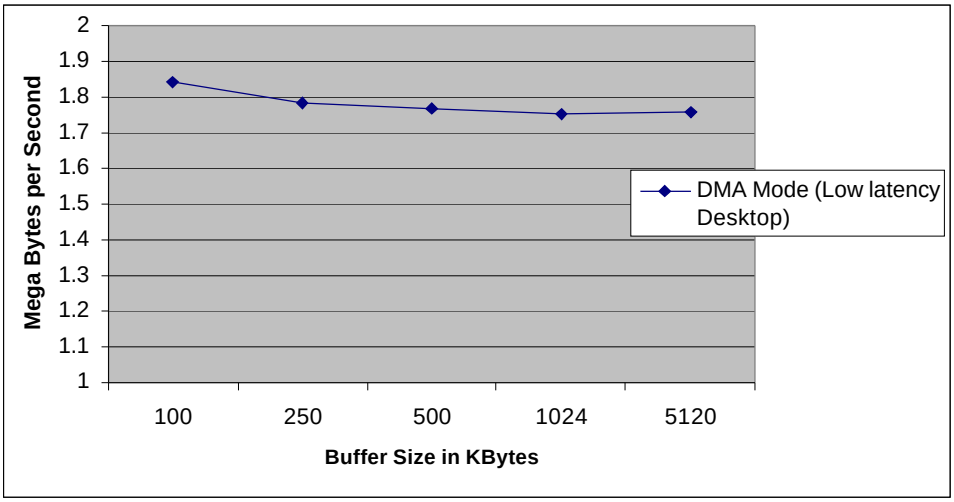


Figure 35. Runtime Performance Characteristics – MMC_File Read

Test Setup Information	Buffer Size in KBytes	DMA Mode		
		Mega Bytes/Sec	File Size in MB	Duration
ARM Frequency= 297 Mhz	100	1.84	100	54.31
	250	1.78	100	56.06
	500	1.77	100	56.49
	1024	1.75	100	56.99
	5120	1.76	100	56.85

Table 40. MMC File Read

3.13.24.1 Comments

- MMC Card used for Testing: San Disk RS-MMC 1GB Card
- File system Used: EXT2 File System
- Mode : DMA

3.13.25 MMC File Write-Real Time Pre-emption- Patch Level 045 Release

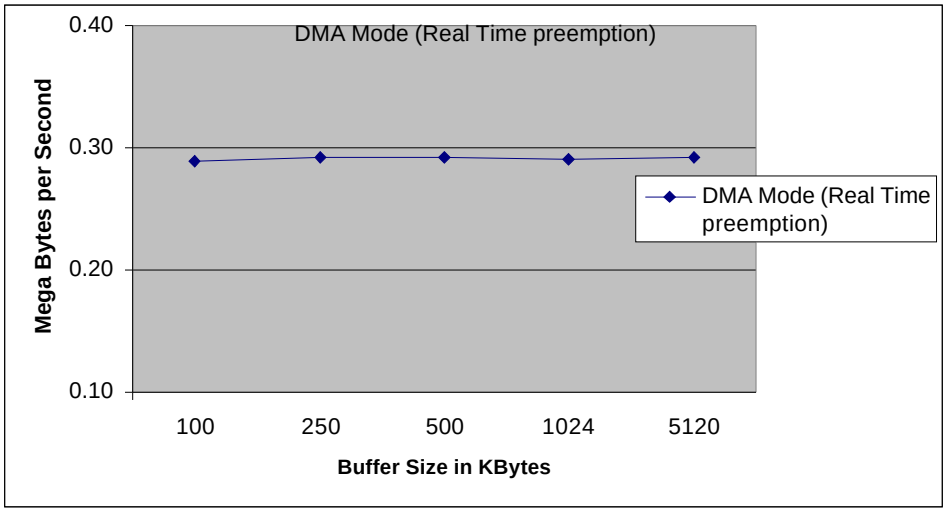


Figure 36. Runtime Performance Characteristics – MMC_File Write

Test Setup Information	Buffer Size in KBytes	DMA Mode		
		Mega Bytes/Sec	File Size in MB	Duration
ARM Frequency= 297 Mhz	100	0.29	100	345.06
	250	0.29	100	341.41
	500	0.29	100	342.59
	1024	0.29	100	344.05
	5120	0.29	100	341.71

Table 41. MMC File Write

3.13.25.1 Comments

- MMC Card used for Testing: San Disk RS-MMC 1GB Card
- File system Used: EXT2 File System
- Mode : DMA

3.13.26 MMC File Read-Real Time Pre-emption- Patch Level 045 Release

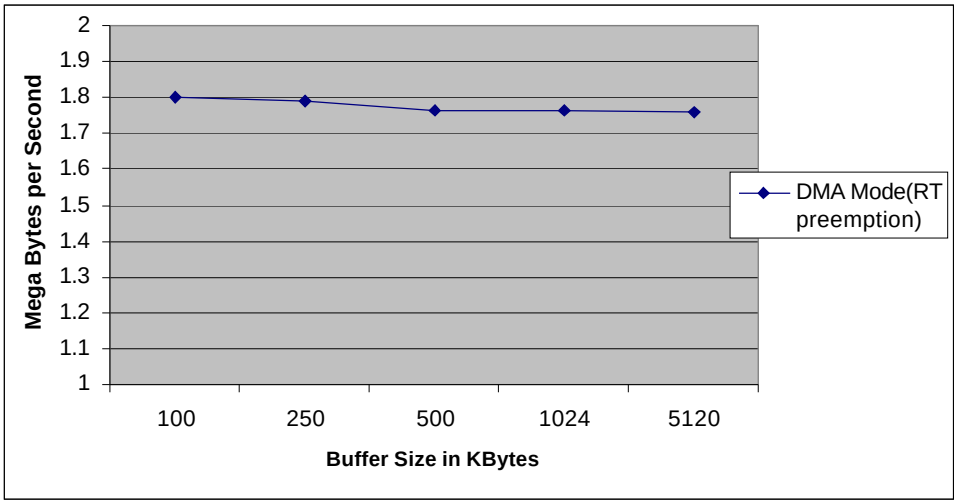


Figure 37. Runtime Performance Characteristics – MMC_File Read

Test Setup Information	Buffer Size in KBytes	DMA Mode		
		Mega Bytes/Sec	File Size in MB	Duration
ARM Frequency= 297 Mhz	100	1.80	100	55.55
	250	1.79	100	55.86
	500	1.77	100	56.63
	1024	1.77	100	56.62
	5120	1.76	100	56.9

Table 42. MMC File Read

3.13.26.1 Comments

- MMC Card used for Testing: San Disk RS-MMC 1GB Card
- File system Used: EXT2 File System
- Mode : DMA

3.13.27 SD File Write-Low Latency- Patch Level 045 Release

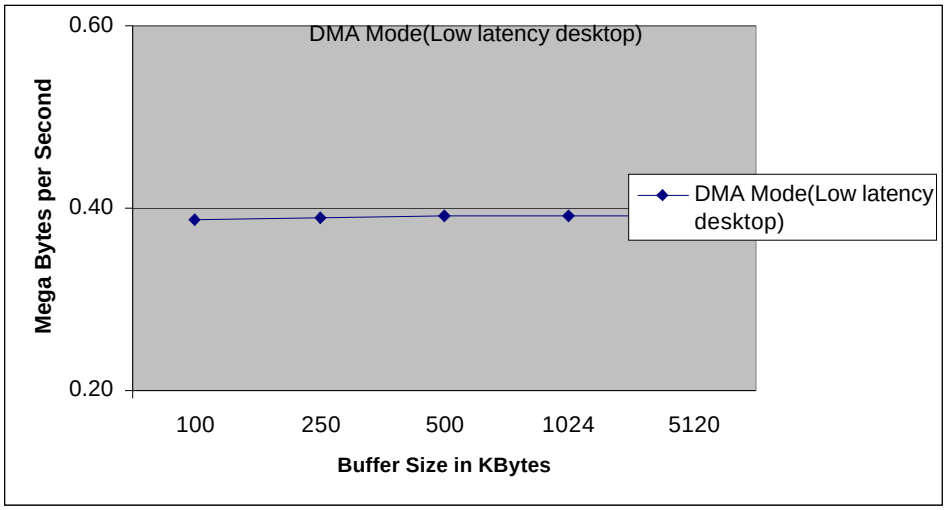


Figure 38. Runtime Performance Characteristics – SD_File Write

Test Setup Information	Buffer Size in KBytes	DMA Mode		
		Mega Bytes/Sec	File Size in MB	Duration
ARM Frequency= 297 Mhz	100	0.39	100	257.33
	250	0.39	100	256.25
	500	0.39	100	255.39
	1024	0.39	100	255.35
	5120	0.39	100	255.79

Table 43. SD File Write

3.13.27.1 Comments

- SD Card used for Testing: San Disk SD 1GB Card
- File system Used: EXT2 File System
- Mode : DMA

3.13.28 SD File Read-Low Latency- Patch Level 045 Release

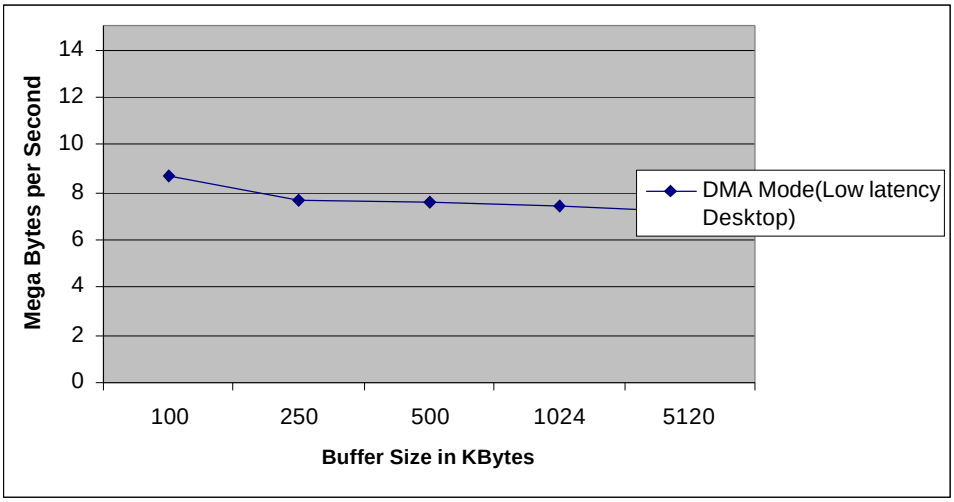


Figure 39. Runtime Performance Characteristics – SD_File Read

Test Setup Information	Buffer Size in KBytes	DMA Mode		
		Mega Bytes/Sec	File Size in MB	Duration
ARM Frequency= 297 Mhz	100	8.69	100	11.51
	250	7.66	100	13.05
	500	7.58	100	13.2
	1024	7.39	100	13.54
	5120	7.18	100	13.93

Table 44. SD File Read

3.13.28.1 Comments

- SD Card used for Testing: San Disk SD 1GB Card
- File system Used: EXT2 File System
- Mode : DMA

3.13.29 SD File Write-Real Time Pre-emption - Patch Level 045 Release

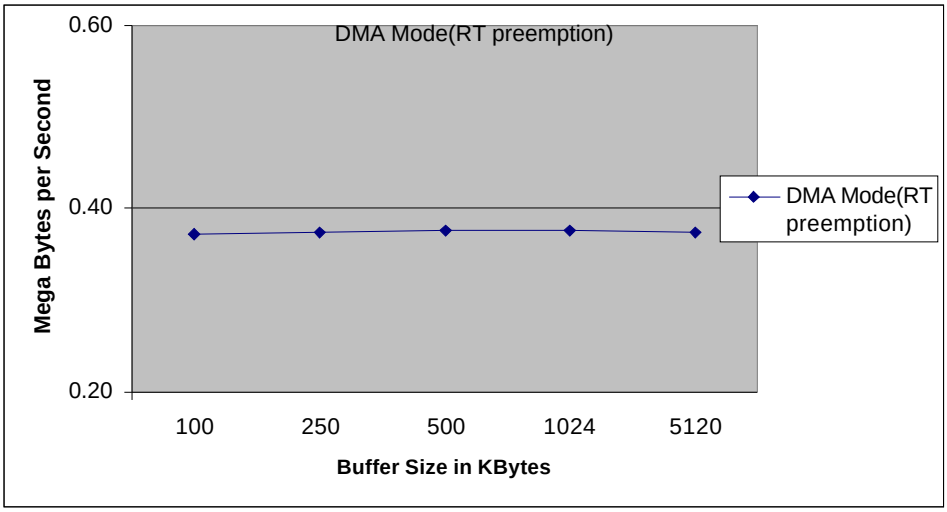


Figure 40. Runtime Performance Characteristics – SD_File Write

Test Setup Information	Buffer Size in KBytes	DMA Mode		
		Mega Bytes/Sec	File Size in MB	Duration
ARM Frequency= 297 Mhz	100	0.37	100	268.28
	250	0.37	100	267.74
	500	0.38	100	266.23
	1024	0.38	100	266.25
	5120	0.37	100	266.91

Table 45. SD File Write

3.13.29.1 Comments

- SD Card used for Testing: San Disk SD 1GB Card
- File system Used: EXT2 File System
- Mode : DMA

3.13.30 SD File Read-Real Time Pre-emption - Patch Level 045 Release

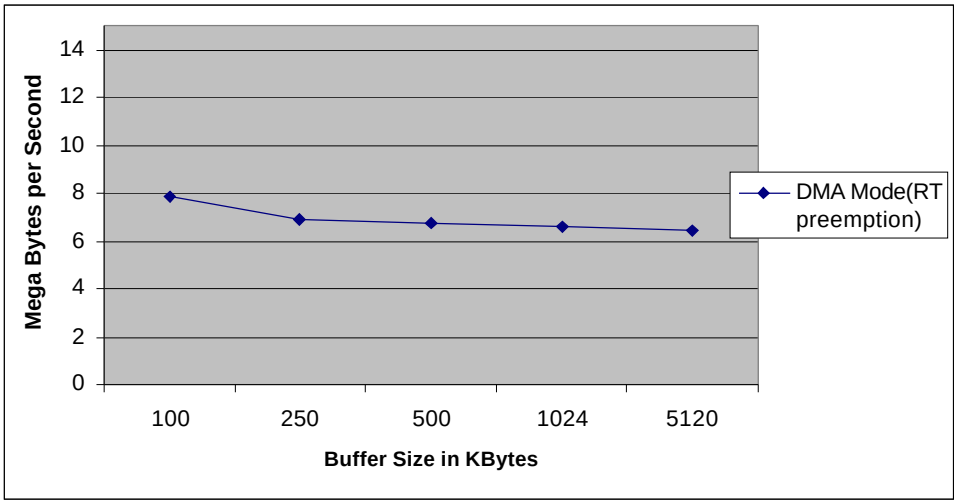


Figure 41. Runtime Performance Characteristics – SD_File Read

Test Setup Information	Buffer Size in KBytes	DMA Mode		
		Mega Bytes/Sec	File Size in MB	Duration
ARM Frequency= 297 Mhz	100	7.84	100	12.75
	250	6.93	100	14.43
	500	6.76	100	14.79
	1024	6.57	100	15.22
	5120	6.45	100	15.51

Table 46. SD File Read

3.13.30.1 Comments

- SD Card used for Testing: San Disk SD 1GB Card
- File system Used: EXT2 File System
- Mode : DMA

3.13.31 MMC File Write with Custom SDIO Stack-Low Latency- Patch Level 045

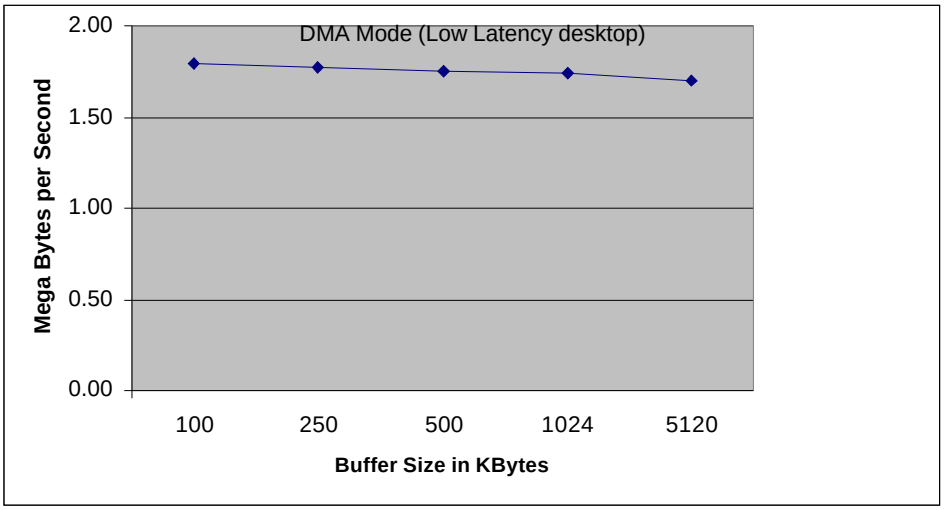


Figure 42. Runtime Performance Characteristics – MMC_File Write

Test Setup Information	Buffer Size in KBytes	DMA Mode		
		Mega Bytes/Sec	File Size in MB	Duration
ARM Frequency= 297 Mhz	100	1.79	100	55.88
	250	1.77	100	56.42
	500	1.76	100	56.96
	1024	1.74	100	57.52
	5120	1.70	100	58.91

Table 47. MMC File Write

3.13.31.1 Comments

- MMC Card used for Testing: San Disk RS-MMC 1GB Card
- File system Used: EXT2 File System
- Mode : DMA

3.13.32 MMC File Read with Custom SDIO Stack -Low Latency- Patch Level 045

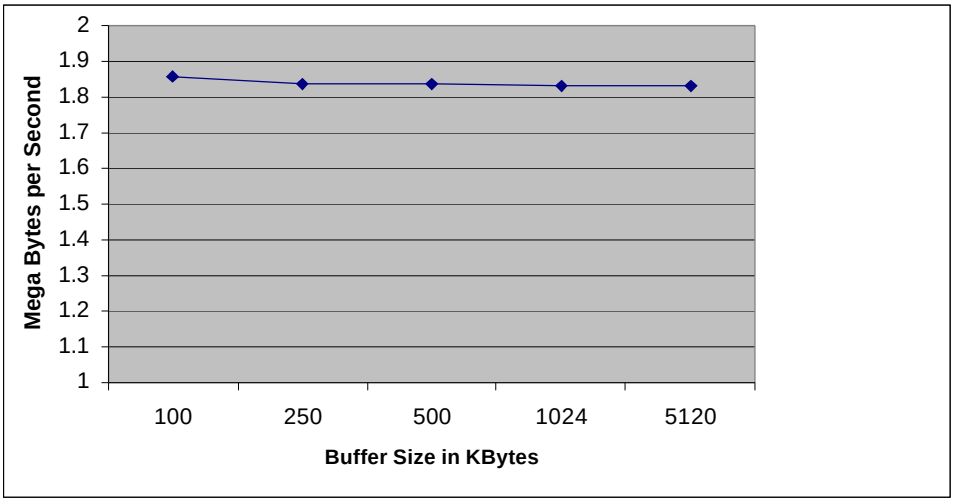


Figure 43. Runtime Performance Characteristics – MMC_File Read

Test Setup Information	Buffer Size in KBytes	DMA Mode		
		Mega Bytes/Sec	File Size in MB	Duration
ARM Frequency= 297 Mhz	100	1.86	100	53.86
	250	1.84	100	54.42
	500	1.84	100	54.48
	1024	1.83	100	54.52
	5120	1.83	100	54.63

Table 48. MMC File Read

3.13.32.1 Comments

- MMC Card used for Testing: San Disk RS-MMC 1GB Card
- File system Used: EXT2 File System
- Mode : DMA

3.13.33 MMC File Write with Custom SDIO Stack -Real Time - Patch Level 045

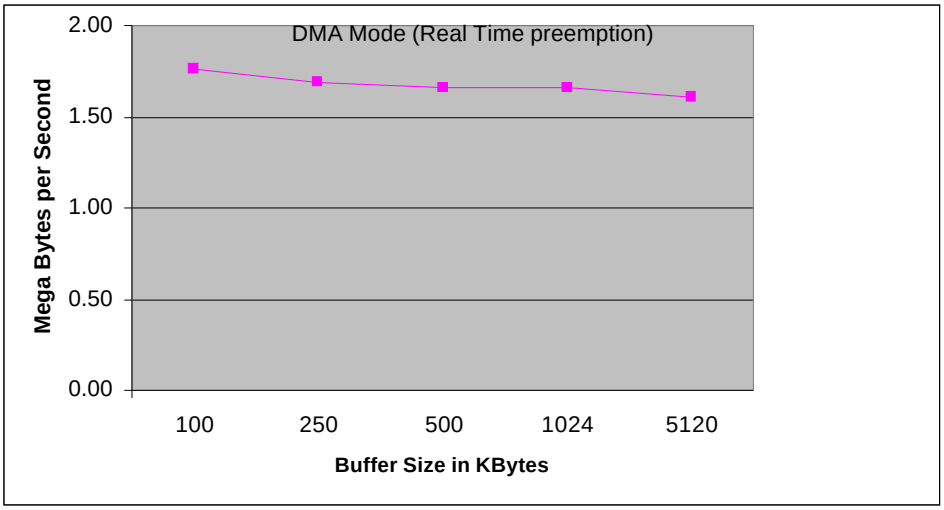


Figure 44. Runtime Performance Characteristics – MMC_File Write

Test Setup Information	Buffer Size in KBytes	DMA Mode		
		Mega Bytes/Sec	File Size in MB	Duration
ARM Frequency= 297 Mhz	100	1.77	100	56.6
	250	1.69	100	59.01
	500	1.66	100	60.14
	1024	1.66	100	60.38
	5120	1.61	100	61.99

Table 49. MMC File Write

3.13.33.1 Comments

- MMC Card used for Testing: San Disk RS-MMC 1GB Card
- File system Used: EXT2 File System
- Mode : DMA

3.13.34 MMC File Read with Custom SDIO Stack -Real Time -Patch Level 045

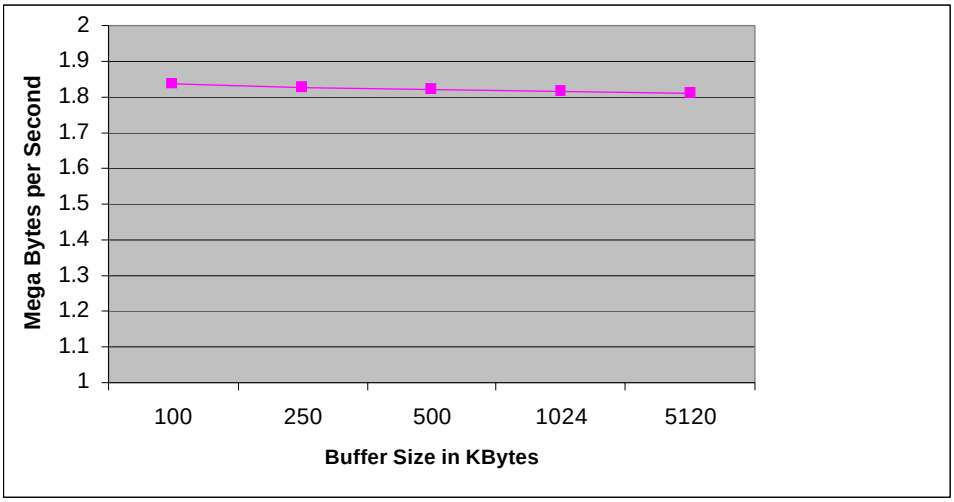


Figure 45. Runtime Performance Characteristics – MMC_File Read

Test Setup Information	Buffer Size in KBytes	DMA Mode		
		Mega Bytes/Sec	File Size in MB	Duration
ARM Frequency= 297 Mhz	100	1.83	100	54.51
	250	1.82	100	54.83
	500	1.82	100	54.93
	1024	1.81	100	55.12
	5120	1.81	100	55.24

Table 50. MMC File Read

3.13.34.1 Comments

- MMC Card used for Testing: San Disk RS-MMC 1GB Card
- File system Used: EXT2 File System
- Mode : DMA

3.13.35 SD File Write with Custom SDIO Stack -Low Latency - Patch Level 045

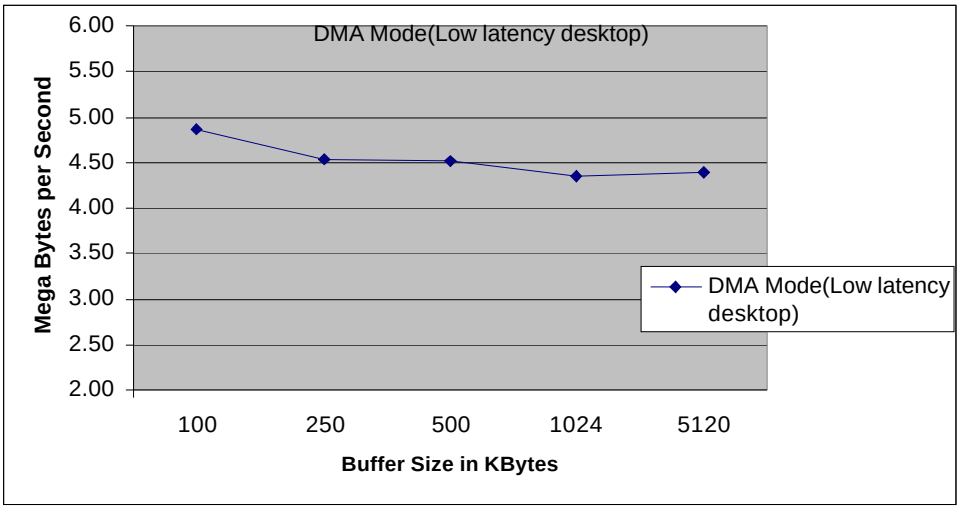


Figure 46. Runtime Performance Characteristics – SD_File Write

Test Setup Information	Buffer Size in KBytes	DMA Mode		
		Mega Bytes/Sec	File Size in MB	Duration
ARM Frequency= 297 Mhz	100	4.87	100	20.55
	250	4.54	100	22.04
	500	4.52	100	22.11
	1024	4.35	100	23
	5120	4.40	100	22.74

Table 51. SD File Write

3.13.35.1 Comments

- SD Card used for Testing: San Disk SD 1GB Card
- File system Used: EXT2 File System
- Mode : DMA

3.13.36 SD File Read with Custom SDIO Stack -Low Latency - Patch Level 045

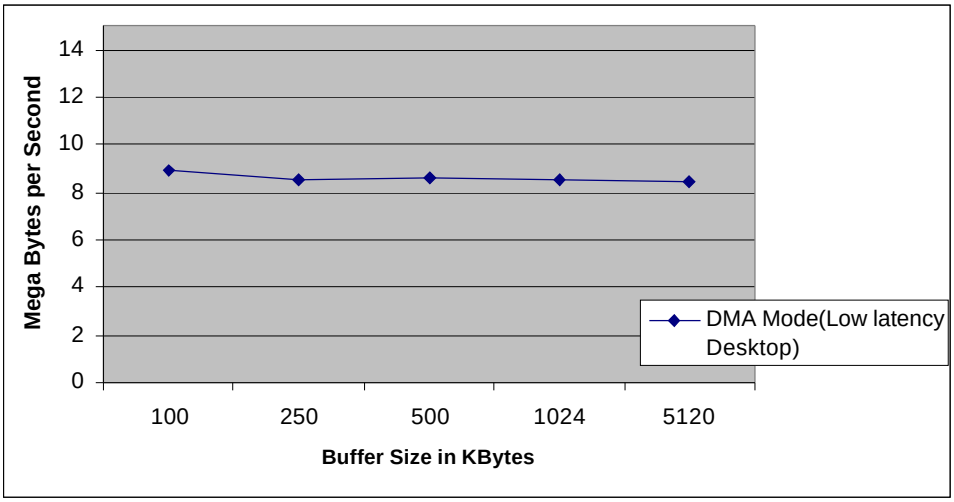


Figure 47. Runtime Performance Characteristics – SD_File Read

Test Setup Information	Buffer Size in KBytes	DMA Mode		
		Mega Bytes/Sec	File Size in MB	Duration
ARM Frequency= 297 Mhz	100	8.90	100	11.23
	250	8.54	100	11.71
	500	8.57	100	11.67
	1024	8.53	100	11.72
	5120	8.42	100	11.87

Table 52. SD File Read

3.13.36.1 Comments

- SD Card used for Testing: San Disk SD 1GB Card
- File system Used: EXT2 File System
- Mode : DMA

3.13.37 SD File Write with Custom SDIO Stack -Real Time - Patch Level 045

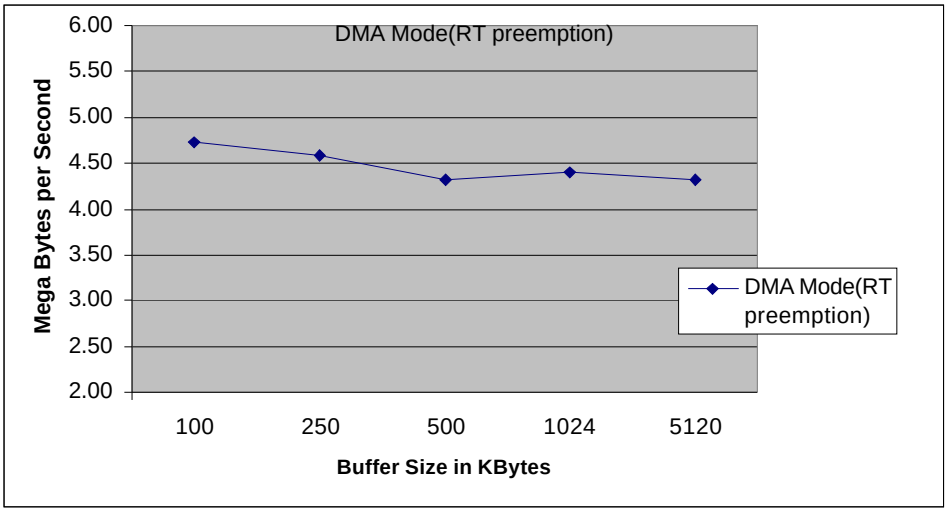


Figure 48. Runtime Performance Characteristics – SD_File Write

Test Setup Information	Buffer Size in KBytes	DMA Mode		
		Mega Bytes/Sec	File Size in MB	Duration
ARM Frequency= 297 Mhz	100	4.73	100	21.16
	250	4.59	100	21.81
	500	4.32	100	23.13
	1024	4.40	100	22.75
	5120	4.33	100	23.11

Table 53. SD File Write

3.13.37.1 Comments

- SD Card used for Testing: San Disk SD 1GB Card
- File system Used: EXT2 File System
- Mode : DMA

3.13.38 SD File Read with Custom SDIO Stack -Real Time - Patch Level 045

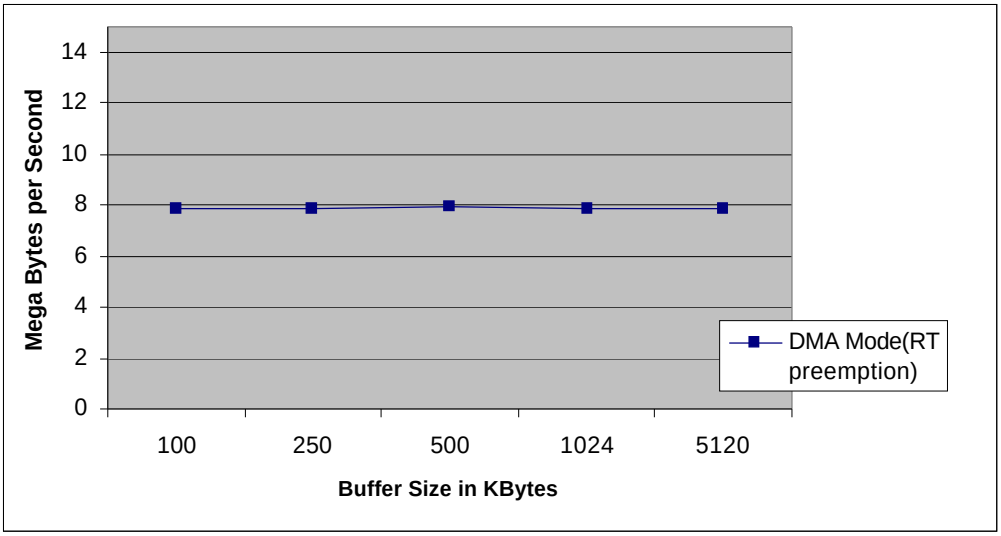


Figure 49. Runtime Performance Characteristics – SD_File Read

Test Setup Information	Buffer Size in KBytes	DMA Mode		
		Mega Bytes/Sec	File Size in MB	Duration
ARM Frequency= 297 Mhz	100	7.87	100	12.71
	250	7.87	100	12.7
	500	7.94	100	12.6
	1024	7.82	100	12.78
	5120	7.86	100	12.73

Table 54. SD File Read

3.13.38.1 Comments

- SD Card used for Testing: San Disk SD 1GB Card
- File system Used: EXT2 File System
- Mode : DMA