

System Software and Power Management in a System-on-Chip for Portable Audio Players

Fitzgerald B Archibald, Michael O. Polley, Stephen H. Li, Jason D. Kridner

Portable devices include audio playback, radio, audio record, movie playback, and image viewer applications. System-on-Chip (SoC) for portable audio players demands low-power and small form-factors differentiated by a wide array of audio effects along with video and image capabilities. This paper describes the power management, software and hardware architecture of SoC for portable audio players.

Index Terms: power management in handheld devices, SoC for portable audio player, systems software for handheld devices.

Contents

1	Introduction	1
2	Multimedia System Overview	2
3	Power Management	3
4	Top-Level System	5
5	Conclusion	7
6	Appendix	8
7	Acknowledgment.....	8
8	References	8

List of Figures

1	SoC Block Diagram.....	2
2	Portable Audio Player Application	2
3	HDD Duty Cycle Diagram.....	4
4	Application Start-Up State Diagram	5
5	Application Program Memory Overlay	6
6	Software Architecture	7

1 Introduction

The audio system for the portable market is different from that for home entertainment, car audio, and professional audio systems. The mobile devices require long battery life, small form factor and the ability to operate under vibrations. The audio quality requirement on a portable audio might not be as stringent as in home or professional equipment. For example, 16-bit PCM samples are output on a portable audio player, whereas 24-bit PCM samples are output to a D/A converter (DAC) in DTV [2]. The portable audio segment takes many forms as in portable audio players, portable media players, voice recorders and cell phones [8].

This paper explores the hardware, software, and systems aspects as well as power optimization techniques of the portable audio player SoC.

2 Multimedia System Overview

The software system for portable players consists of control software like a graphic user interface (GUI), file system management, license acquisition (part of DRM), connectivity to the PC for download services, time update, and alarm/calendar functionality. The control software runs efficiently on a general-purpose processor (GPP) like ARM. The audio, image, video codec and effects would require specialized accelerators or co-processors, or a DSP. Some of the audio functionalities could be possible with the use of ARM core with DSP extensions [10].

Figure 1. SoC Block Diagram

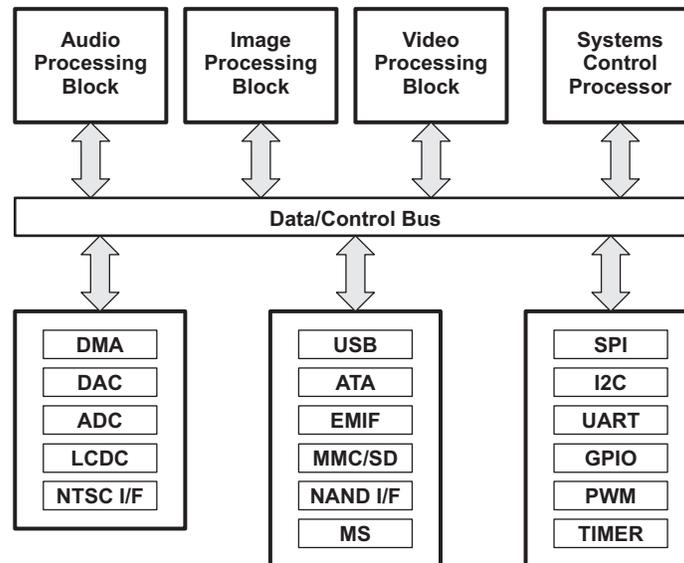
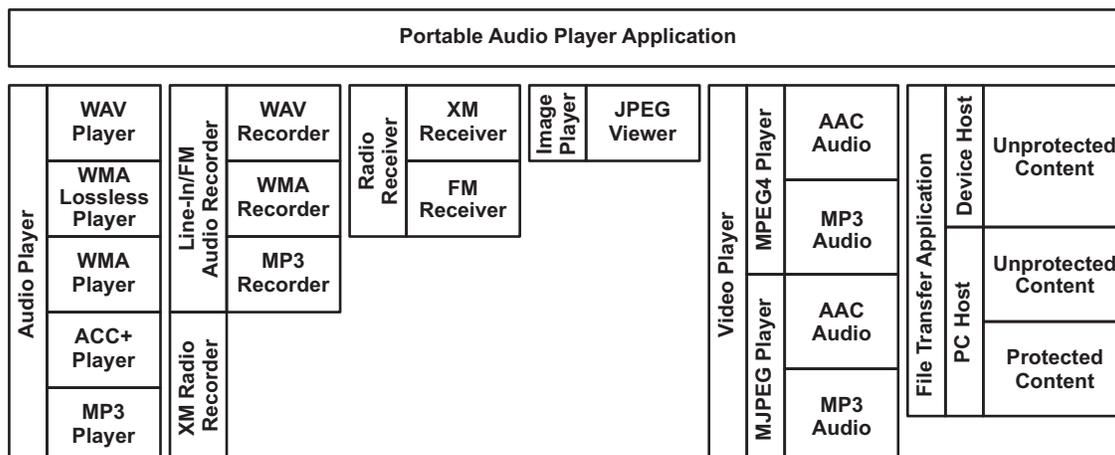


Figure 1 illustrates a SoC with essential hardware IP blocks to enable the portable audio player system realization. The audio, image, and video processing blocks can be realized using a DSP coupled with accelerators or a GPP coupled with co-processors. The ARM core is typically used for systems control functionality to achieve low power for control applications [11].

Figure 2 provides various applications that are needed to realize audio, image, and video requirements of a portable audio player.

Figure 2. Portable Audio Player Application



The audio system on a portable device has to handle the digital rights management (DRM) for media files to protect copyrights, allow connectivity to a personal computer (PC) for download and/or upload of files, connectivity between two devices, compressed audio (MP3, AAC, ...) file playback, FM and/or satellite radio reception, and recording of microphone and/or radio content in compressed form.

Portable audio players provide additional features like photo viewer, video playback, and gaming for added value to customers. These features allow easy view of digital photos (in JPEG format) as well as serve as storage/transfer media. The video playback capability on a portable audio player might be limited with respect to frame rate, frame size, codec type, etc. as compared to that of a portable media player (PMP). Image and video systems are possible because of the availability of Digital Signal Processor (DSP) and / or image/video accelerators and co-processors [12].

3 Power Management

The system is incomplete without satisfying the desired battery life. Various parameters like voltage, frequency, and IDLE modes control the battery life or power consumption by a device. Audio playback time is benchmarked by running the system with only input data streaming, audio decoder (MP3), and DAC circuitry connected to a headphone jack.

3.1 Clock Frequency and Voltage

Clock frequency and voltage would determine dynamic power consumption for a given gate count and process technology [6]. Higher frequency operation of cores would require higher operating voltage. The voltage increase has a square effect on power, whereas a frequency change affects power linearly as can be seen from (1), for CMOS circuits [2].

$$P = \alpha C_L V_{dd}^2 f \quad (1)$$

In 2, f is the system clock frequency, V_{dd} is the supply voltage, C_L is the load capacitance, and α is the switching activity.

Dynamic clock change capability would optimize power consumption by clocking the circuitry based on the processing speed required by the application. Dynamic clock change for cores could be implemented by monitoring the time spent by the cores in the IDLE task in a real-time operating system (RTOS)-based system [7]. Dynamic clock change could result in real-time misses if adequate care is not taken while designing the system. The most preferred choice is selecting the clock frequency from a clock table based on the application state. The table is pre-determined at development time. Dynamic clock change values could be used to validate the clock table and vice-versa.

Voltage and frequency changes are in incremental steps. For a given voltage, several frequency settings are possible using PLL, within the allowed frequency range.

The SoC is split into multiple power and clock domains to allow precise control of the frequency and voltage of one domain without affecting the operating voltage/frequency of another power/clock domain.

3.2 Idle and Power Down

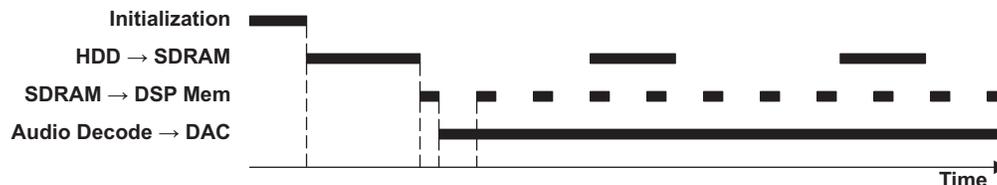
The modules that are not in use can be transitioned to IDLE state (clock is stopped) so that the modules do not consume dynamic power. The IDLE sequence or procedure will be followed for the ACTIVE to IDLE state transition. For example, in the case of an image-only application, audio DAC (analog and digital) and I2S interface hardware blocks can be in IDLE state. The Wake-up sequence will be followed for waking up modules in IDLE mode. Only leakage current will be consumed by modules in IDLE mode.

The modules having power on/off control will be powered-down, when not in use to save power (for example, internal audio DAC can be powered-down when the player is in file transfer mode). The power-down and power-up will follow a necessary sequence to allow the proper functioning of devices. In portable devices with security/encryption features, real-time clock (RTC) can never be powered-down. The power-down of analog interfaces can save significant power. PLL can be turned off for applications that can operate from direct crystal clock (PLL bypass mode) to save energy.

3.3 Track Cache

In hard-disk drive (HDD)-based players a significant amount of power is consumed by the hard disk if the disk is powered-on [9]. Based on SDRAM memory availability, a sufficient amount of media is transferred from the hard disk to external memory (SDRAM) for playback. The external memory is treated as track cache (for hard-disk access). Once the track cache is full, the hard disk is powered-off. Again, when the track cache reaches a lower threshold, the HDD device is turned on (adjust threshold based on HDD power-on/spin-up time). Thus, instead of constant power consumption by the hard disk, a duty cycle is applied to prolong battery life, as in Figure 3. The associated ATA driver can be powered-off, when the hard-disk drive is off.

Figure 3. HDD Duty Cycle Diagram



3.4 SDRAM Operation Mode

SDRAM memory can be placed in self-refresh mode to reduce power consumption. SDRAM could automatically come out of self-refresh mode when there is access and enter self-refresh mode if there is no access [5].

SDRAM access could be minimized on an ARM system by placing the stack and frequently used variables in ARM internal memory. This might not be feasible if the internal memory is limited for complex applications.

A DSP system accesses SDRAM for fetching data from track cache, for storing/retrieving PCM samples (buffering), and for loading/saving programs from SDRAM. The track cache memory access can be optimized by providing large enough DSP internal memory to hold compressed data for decoder input. The track buffer in internal memory is refilled when the buffer reaches the lower refill threshold, thus minimizing SDRAM access. The buffering and program load/save accesses can be minimized by tuning application overlay parameters by the use of lower and upper thresholds for buffering.

3.5 Sample Rate Control Using DAC

Software sample rate control (SRC) has the problem of consuming CPU clock cycles and thus power. The alternative is to set the sample rate register in DAC to the desired sample rate, eliminating the need for software SRC [8].

3.6 Board Design

A choice of hardware components like low power DAC/ADC, mSDRAM would enhance playback time. Board layout, unused components, power management unit (PMU), and pin termination are vital for system power efficiency. Adequate care needs to be taken while designing a board to get maximum battery life and audio quality.

4 Top-Level System

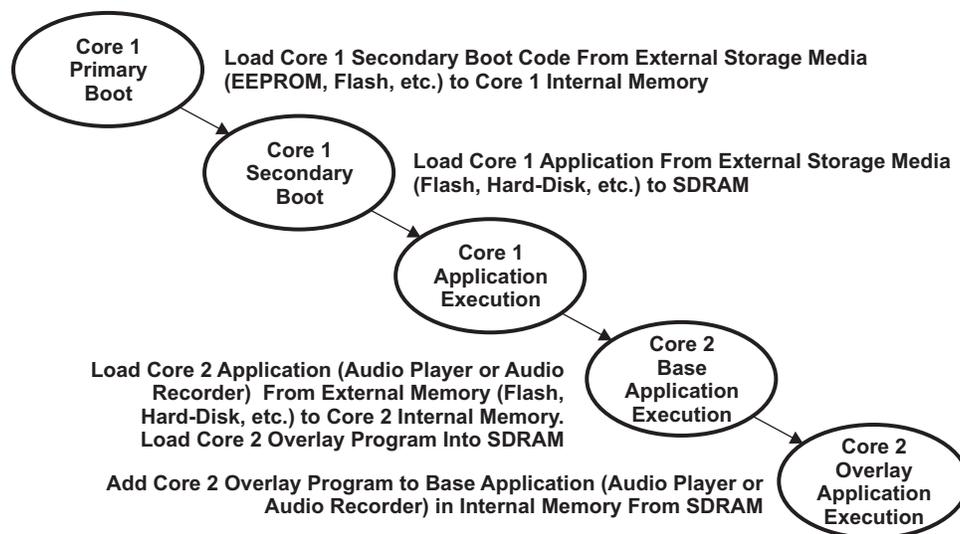
4.1 System Boot

For SoC having multiple cores, one of the cores (typically GPP) acts as master. The SoC internal ROM contains the primary system boot code. Primary boot code is executed by ARM from its location, which would load secondary boot code from external ROM or flash memory into ARM internal memory (RAM). The primary boot gives control to the secondary boot, which in turn loads the ARM application from the flash or hard disk to the ARM internal/external memory. The boot sequence is shown in [Figure 4](#).

Typically all the programs (boot-loader and application) are encrypted to ensure security. The decryption could be performed by the ARM core. When the boot-up time needs to be minimized, dedicated hardware decryption logic can be used (a boot-up sequence along with other system parameters like power-up sequences, system software would determine starting the delay of playback).

In case of dual-core architecture, the primary core loads the program onto the secondary core while holding the core in reset. The program for the secondary core could be in flash or hard-disk memory. The program is loaded onto SDRAM memory by the primary core. The DMA is used for loading a program to the secondary core memory from SDRAM to minimize loading time and to free up the primary core for parallel operations. Once the secondary core program and data load is completed, the primary core releases the secondary core from reset, allowing the secondary core to execute a loaded program.

Figure 4. Application Start-Up State Diagram



4.2 ROM vs RAM

ROM takes less area compared to RAM. Stable and well-tested software modules (both data and program) can be placed in ROM instead of RAM.

Placing basic building elements like RTOS, drivers, and application framework in ROM would enhance ROM usage.

Care should be taken while placing modules in ROM for correcting defective code or data in ROM. Defective code or data in ROM can be replaced with alternative code or data, if the ROM content is accessed via pointers resident in RAM.

ROM content can be encrypted for security. On-chip ROM is a key element in secure boot systems.

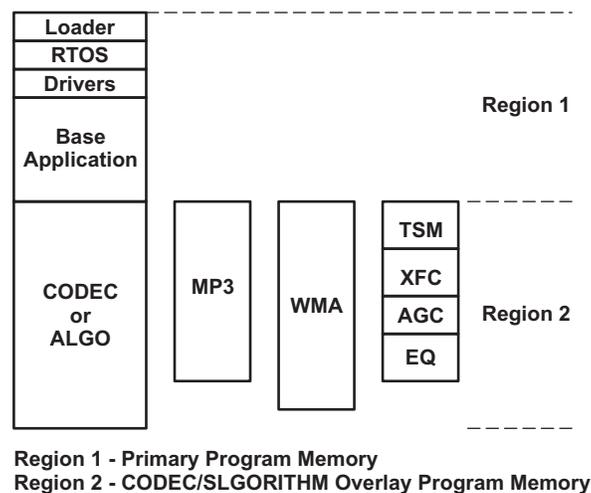
4.3 Application Load/Overlay

When there is limited RAM for program applications, memory overlay is used. The program and data memory is loaded dynamically at run time. The secondary core program could load the necessary decoder code, from a fixed SDRAM address, before invoking the decoder module.

In [Figure 5](#) dynamic overlay is illustrated wherein the code and data is swapped on a continuous basis to achieve real-time playback. This scheme is essential for applications requiring large memory (audio player with complex post-processing, video player, etc), when available internal memory space is limited. The audio effects program and audio codec reside in same physical memory space. To achieve this overlay, the codec and post-processing programs states need to be saved and reloaded. An overlay scheduler loads and unloads each application on a pseudo-periodic fashion. To reduce the frequency of program swap, SDRAM buffers are used to hold larger number of samples of codec output and post-process program output. Increasing the buffer size beyond a certain limit would introduce delay when the player state transitions are initiated.

For video player applications, the audio program in memory is replaced by a video program after buffering enough audio data. The audio application has higher priority for the overlay scheduler to avoid audio breaks.

Figure 5. Application Program Memory Overlay



4.4 Buffering Consideration

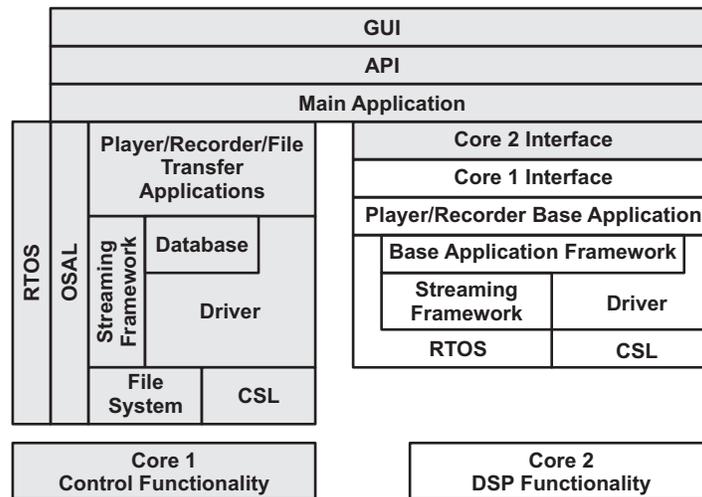
Data buffering plays a crucial role in real-time playback, and power, cycles, and memory optimization. The system should have adequate buffering along with clock cycles to avoid real-time misses. SDRAM accesses for program swap can be optimized by tuning buffer sizes and the overlay scheduler. The amount of data buffers in internal memory is one of the factors affecting context switching overhead. Buffering and data format compatibility are essential when interfacing modules.

Data converters might be needed to match the buffer size and data format of the different modules. Memory copy involved with data converters consumes CPU cycles. Data copy could be performed with DMA if enough channels are available, to save CPU cycles and power. DMA support for 1D and 2D data transfers would optimize CPU cycles for audio and image/video applications.

4.5 Software Architecture

The software architecture follows a layered approach as shown in Figure 6. The chip support library (CSL) enables the masking of device-specific address mapping from the driver layer. The OS abstraction layer (OSAL) protects the application and driver layers from RTOS API changes and RTOS changes. Core1 performs computation intensive part of the application, whereas core 2 performs control intensive part of the application. Application-specific implementation changes are masked from the upper layers using well defined APIs.

Figure 6. Software Architecture



4.6 DRM

The decrypt cipher is computation intensive. Hence, the decrypt cipher could be implemented on a DSP or decryption hardware block. License acquisition and handling is mainly a control-oriented task, which can be performed using the ARM [8].

The DRM module is closely coupled with the database and file transfer application. The real-time-clock (RTC) is needed on SoC for ensuring DRM effectiveness against clock rollback [4]. This would require that RTC alone be powered-on in the SoC.

5 Conclusion

Dual-core SoC architecture allows for better operating frequency, and voltage control along with clock gating to achieve low power without degrading system performance. System design will minimize inter-process communication for applications to improve efficiency of processor utilization.

The DSP and accelerators, and/or co-processors aid in video playback and image display in portable audio players. GPP is optimal for control functionality associated with SoC. Programmable solutions provide the ability to add algorithms and custom interfaces to differentiate products using the same SoC platforms, and thus reducing the time to market.

The integration of IP like PMU, ADC/DAC and image/video co-processors into SoC or providing chipsets aid in reducing the bill of material (BOM), cost, power optimization, and board size reduction. Peripheral clock gating and power domain control allow power consumption reduction.

6 Appendix

Table 1 illustrates power consumption for an AAC-LC player on a DA295 EVM (with headphone). The AAC stream used is AAC-LC stream at 128 kbps bit-rate and 44.1 KHz sample rate. Battery used for measurement is 630 mAh. The measured battery life is 35 hours, which can be improved further by board design changes.

Table 1. AAC PLAYER BATTERY LIFE ON DA295 EVM

Player State	Measured Power	Duty Cycle	Average Power
HDD active <ul style="list-style-type: none"> ARM@ 30 MHz EMIF@ 30 MHz 	1086 mW	0.0047 <ul style="list-style-type: none"> 24 MB SDRAM track buffer 6 MB/sec transfer 2.3 sec init/spin-up, 0.5 sec spin-down 	5.1 mW
HDD not active <ul style="list-style-type: none"> ARM@ 12 MHz EMIF@ 3 MHz 	54.6 mW <ul style="list-style-type: none"> 10.95 mW - Hibari digital 13.28 mW - Hibari DAC/Amp 6.45 mW - EMIF I/O + SDRAM 23.94 mW - Headphone, supply loss, other 	0.9953	54.3 mW

7 Acknowledgment

The authors thank Alec Robinson, R&D, TI for his inputs on software. The authors acknowledge valuable comments on system architecture, power management and audio DAC from Bruce Bonnet, Architecture Team, TI. Authors thank Keven Coates, and Greg Hewes of TI for their inputs on system and board.

8 References

1. *Power Optimization of Variable-Voltage Core-Based Systems* by I Hong, D Kirovski, G Qu, M Potkonjak, and M B Srivastava; IEEE Transactions On Computer-Aided Design Of Integrated Circuits And Systems, Vol. 18, No. 12, December 1999
2. *A Digital Television Receiver Constructed Using A Media Processor* by C Peplinski, T Fink; 106th AES Convention, Munich, May 1999
3. *A Programmable DSP Solution To A Solid State Audio Player* by J Hayes, S Handley, J Kridner, and M Nadeski; Consumer Electronics, IEEE Transactions on Volume 45, Issue 3, Aug. 1999, pp. 975 – 979.
4. *Windows Media DRM Architecture*, Microsoft Corporation
5. Micron 64 MB mSDRAM datasheet. Available: http://download.micron.com/pdf/datasheets/dram/mobile/Y25L_64Mb.pdf
6. *Power Management in Complex SoC Design* by J Flynn. Available: <http://www.synopsys.com/sps>
7. *Dynamic Power Management for Embedded Systems*, IBM, MontaVistaSoftware. Available: http://www.research.ibm.com/arl/publications/papers/DPM_V1.1.pdf
8. *Audio System for Portable Market* by F J Archibald, 121st AES Convention, San Francisco, Oct 2006
9. Seagate ST1.3 HDD datasheet. Available: http://www.seagate.com/docs/pdf/datasheet/disc/ds_st1_3.pdf
10. *ARM DSP-Enhanced Extensions* by Francis, May 2001. Available: <http://www.arm.com/pdfs/ARM-DSP.pdf>
11. *OMAP5912 Multimedia Processor Device Overview and Architecture Reference Guide (SPRU748)*, Texas Instruments
12. *TMS320C55x Hardware Extensions for Image/Video (SPRU098)*, Texas Instruments

Fitzgerald Archibald received B.E (Electronics and Communication Engineering) from PSG College of Technology, Coimbatore, India in 1996. He worked on onboard control systems software development for geo-synchronous satellites from 1996 to 1999 in ISRO Satellite Centre, Bangalore, India. In 2001-2002, he worked on speech decoder, real-time kernel, and audio algorithms for DVD audio team in Sony Electronics, San Jose, USA. While in Philips Semiconductors (Bangalore, India, and Sunnyvale, USA) in 1999-2001 and 2002-2004, he worked on audio algorithms and systems for STB, DTV, and internet audio. He is part of the Imaging and Audio Group in Texas Instruments Inc, Bangalore, India from 2004-till date working on audio systems and algorithm development, and video algorithm development.

Mike Polley received his B.S., M.S., and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology. In 1996 Mike joined Texas Instruments to help TI influence the DSL standards bodies and enter the ADSL market. Mike led the ADSL research group developing DSP based solutions for evolving DSL standards. Mike then came unwired - he helped TI establish a fixed-wireless access chipset business and then led the way developing multiple-antenna wireless LAN technology and implementations in advance of the IEEE 802.11n standard. Mike currently manages the architecture team developing camera and video chips for cell phones and he is a Distinguished Member of Technical Staff.

Stephen Li earned his BS in Electrical Engineering at University of Texas at Arlington, MS in Biomedical Engineering at UT Arlington/UT Southwestern Medical Center, and MS in Applied Mathematics at Southern Methodist University. Joined Texas Instruments in 1984, Stephen has worked, in different capacities, on the design, modeling, processing, application, and architecture definition of VLSI IC. Stephen is the author of several technical papers and journal articles covering VLSI architecture and design. Stephen owns over 25 U.S. patents in the same area, as well as A/V algorithm development.

Jason Kridner is a Senior Member of Technical Staff in the DSP Systems Portable Audio and Video (PAV) group working with other team members to define core technologies for portable media enabled products. Jason has work experience as a draftsman and C programmer for test equipment before joining TI in 1992 as a co-op in DSP Product Engineering where he primarily automated data collection and generation of characterization reports. In 1994, Jason moved into DSP Applications working on board and ASIC design for audio and eventually DSL products. In 1999, he wrote much of the system software for TI's entry into the MP3 portable player chipset market. In 2002, the digital still camera and portable audio businesses merged and Jason participated in video codec development, systems architecture, and support for imaging, video, and audio applications in portable media players and mobile phones. In 2007, he accepted the role of chief technologist for PAV to guide in setting the strategic direction for the group. He joined the IEEE in 1992.

IMPORTANT NOTICE

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

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

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

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

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

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

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

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

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

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

Products

Amplifiers	amplifier.ti.com
Data Converters	dataconverter.ti.com
DSP	dsp.ti.com
Clocks and Timers	www.ti.com/clocks
Interface	interface.ti.com
Logic	logic.ti.com
Power Mgmt	power.ti.com
Microcontrollers	microcontroller.ti.com
RFID	www.ti-rfid.com
RF/IF and ZigBee® Solutions	www.ti.com/lprf

Applications

Audio	www.ti.com/audio
Automotive	www.ti.com/automotive
Broadband	www.ti.com/broadband
Digital Control	www.ti.com/digitalcontrol
Medical	www.ti.com/medical
Military	www.ti.com/military
Optical Networking	www.ti.com/opticalnetwork
Security	www.ti.com/security
Telephony	www.ti.com/telephony
Video & Imaging	www.ti.com/video
Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2008, Texas Instruments Incorporated