



Session 5A: Hardware & Software Architectures



Real time embedded software system on a heterogeneous Digital Signal Processor and RISC processor architecture

By: Manfred Christ
Telematics System Engineer - SMTS
Texas Instruments Germany
Haggertystrasse 1
85356 Freising
Germany

Tel: +49-8161-804294 - Fax: +49-8161-804523

Email: m-christ@ti.com

Biographical details:

Manfred Christ is employed at Texas Instruments since 1978; he is Senior Member Technical Staff. He worked in various positions as product engineer and since 1986 as Field Application Engineer in DSP and analog application groups. He joined the Texas Instrument's Telematics team in 2002 and focuses in the role as Telematics System Engineer on multimedia and navigation applications and develops system solutions for the automotive market.

Title of paper: Real time embedded software system on a heterogeneous Digital Signal Processor and RISC processor architecture





Introduction

This paper discusses a generic telematics systems based on the OMAP™ (open multimedia application platform) heterogeneous ARM™/DSP multiprocessor system. These 2 processors are integrated as SOC (system on chip) with a peripheral mix dedicated to the automotive requirements as a one chip solution. These processors are able to run parallel various real time applications optimized either for the DSP or the ARM-RISC processor or shared between both processors.

Automotive radio and navigation systems require high integration and flexibility through programmability. High integration provides cost and space optimized solutions. Programmability allows adaptation of the software for new developments in algorithms and standards. Low power consumption is another important requirement and concern of system architects. Application optimized processors and high integration is important to fulfill all these requirements.

Telematics applications have to deal with several audio sources, have to drive several audio amplifiers and require an easy interface to the user, usually the driver of the car. Several real time applications have to run in parallel, like the HMI (human machine interface), either based on traditional radio-button based interfaces or on voice recognition.

The OMAP™ dual core processor addresses all these issues with various optimized interfaces for analog, audio and digital data sources and sinks. The ARM™ processor, driven from validated automotive operational systems controls the system, and dynamically loads the required algorithms to the DSP. The DSP is optimized for audio and codec functions and able to react as real time processor to the various data streams.

Heterogeneous DSP/RISC solutions, with application optimized peripherals built as SOC, will be key for many applications. These solutions combine both high performance DSPs and standard RISC processors. Standard operational systems can be used. This allows fast time-to-market as many 3rd party software products can be easily integrated.

A simple definition of Telematics

In modern cars it is required that the driver information is integrated in the environment of the car. Telematics is the name for such solutions. It involves the convergence of consumer electronics, computing, and communication in the vehicle. Navigation, digital radio, Internet, multimedia entertainment, reconfigurable displays, wireless access, and other applications are driving the need for advanced electronic components in the vehicle.

Considerations for a Telematic system

A Telematic system is the integration of radio, navigation and communication into 1 system. The navigation system is based on a GPS (global positioning system) module providing the location information. In the future the new European Galileo system will provide even better and more precise location information. The coordinates provided from the navigation system is correlated with mapping data from a CD or DVD. This allows calculating the best route based on the location of the car and the available data. The problem with this solution is that the data on the CD is usually not in an actual status. It cannot take into account the traffic situation, like density of traffic on the various routes, accidents and traffic jams can quickly change the situation. A dynamic system has to take these circumstances into account.

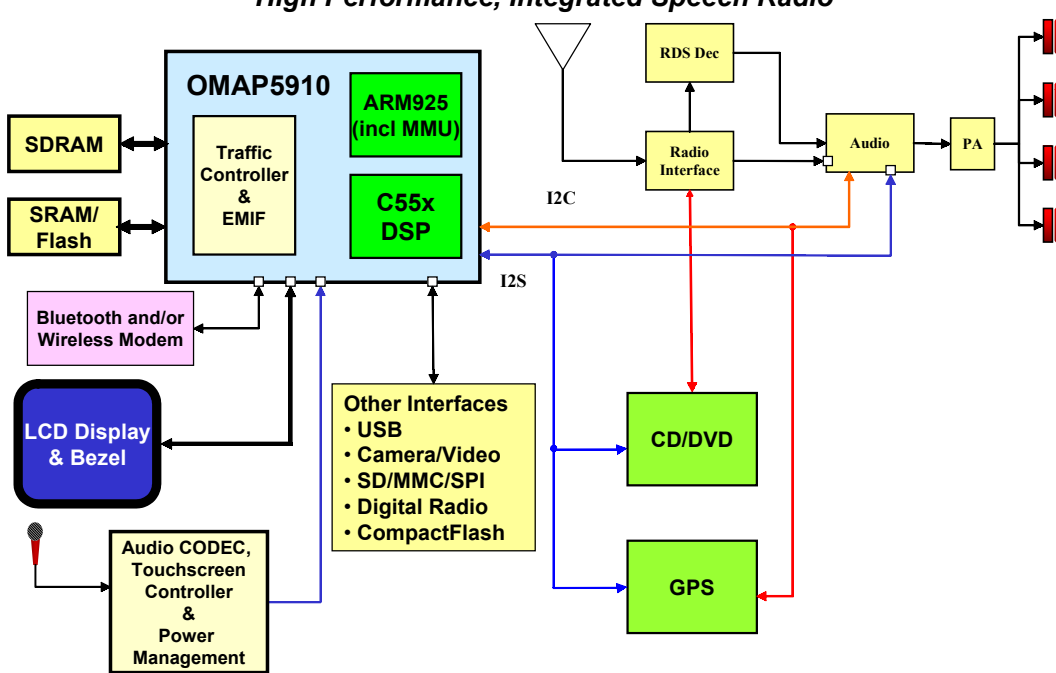
Dynamic data for continuous update can come from various sources. Today analog radio provides regular traffic information, but there is no link to the navigation system, except that the driver can actively take the traffic information into account. Another disadvantage is that the traffic information is broadcasted for a large area, therefore the necessary data comes quite often not precise enough and without recommendation for other routes. Digital Radio will change this in the future. Data can be



transmitted to the car very precisely and directly to the navigation computer. This will help to quickly develop alternative routes to recommend to the driver.

OMAP5910 System Diagram

High Performance, Integrated Speech Radio



Generic information about road constructions on certain routes and updated road information in general can come via a mobile phone as an update to a CD on regular basis.

A modern Telematic system has to integrate all these aspects. The GPS based navigation system provides the location information. This data is correlated with the mapping information stored on a CD or DVD and updated with actual information coming via digital radio and a mobile phone data channel. Based on this data and the planned destination of the journey the navigation computer can calculate to optimal route. If new data coming in suggest a change due to traffic jam, an alternative route can be calculated and suggested.

As the driver should not work with the navigation system while driving the interface between the navigation computer and the driver has to be done via speech. A modern Telematics system has therefore to include good speech synthesis software with speech recognition. Such a system can establish a dialog between the driver and the navigation computer in order to find the best route under actual traffic situations to the planned destination.

Other applications for a Telematics system

The Telematics system has to be more than a navigation computer. As already discussed it has an interface to the digital radio and the GSM phone. This includes of course that the necessary audio stream to the speaker in case of the radio, or the microphone and a speaker for the mobile phone will be routed through the navigation computer or Telematics system. This required that the HMI (human machine interface) for all radio, mobile phone and navigation based functions have to be done from the same system.

Audio streams sources are CD-DVD player, flash memory cards, radio tuner, or the microphone for the mobile phone. These sources have to be routed to the speaker.



In future systems also video streams have to be routed through the same system to provide either video from a DVD player to the rear seats in the car. In this case the audio data might be different in the front and rear of the car, so that 2 stereo audio streams are required.

OMAP, a programmable solution for adaptable applications

The previously described system requires that a very flexible and configurable system. It is impossible, that a dedicated hardware solution or an embedded processor runs all these applications in parallel. Additionally it has to be taken into account that new codecs might be defined and the system has to be flexible enough to adapt these new algorithms. This is only possible with a software programmable solution.

OMAP (open multimedia application platform) is an answer to these requirements. It contains 2 processors which are optimized for different applications. Common interfaces to the outside world allow the different input data and audio streams to be loaded into the processors. After processing the data various data sinks route the data to the speaker or display.

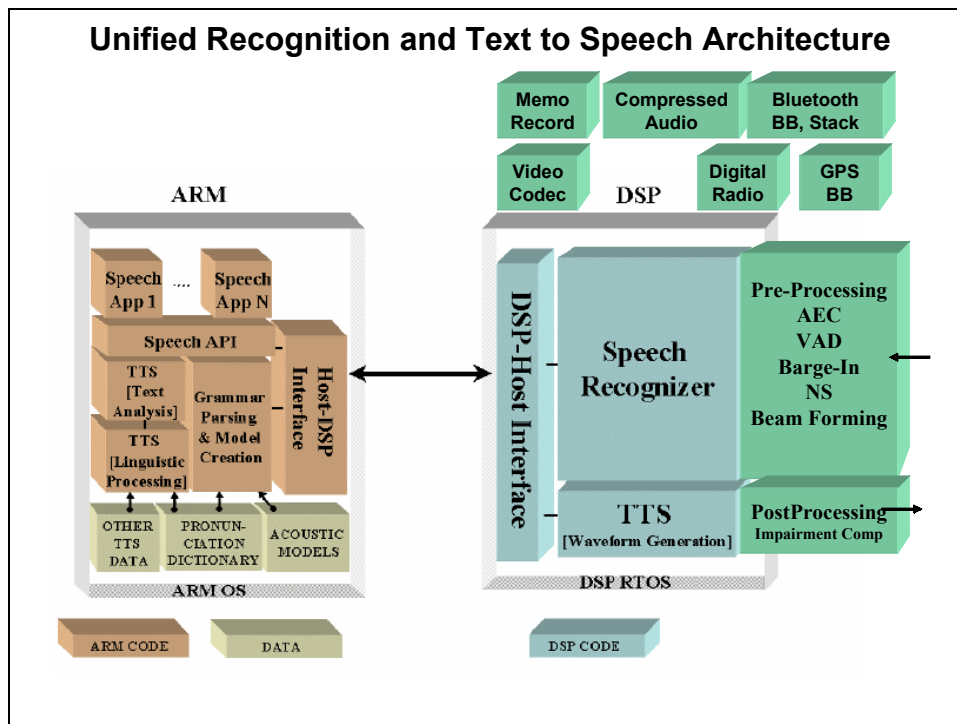
Of course also the OMAP processor system cannot run all the applications concurrently. Therefore the actually required algorithm is loaded to the right processor, if possible into the internal memory or cache.

OMAP is originally developed for the mobile market. Also here different algorithms and applications are required. For HMI and data management functions, which usually have to run on a standard operational system, a RISC processor, like the ARM926 is the optimized processor. Many operational systems, like Windows® CE .NET™, Linux®, Nucleus™ and VxWorks™, Symbian™, QNX™, and many more exist for this processor platform or are in development.

Audio streams, calculation intensive algorithms, e.g. in GPS, codecs with various decoders like MP3 and MP3Pro, MPEG I+II Layer 2, MPEG II + MPEG IV AAC, WMA, AC3, ADPCM are best suited for a digital signal processor. Therefore a TMS320C55x core has been integrated into OMAP.

OMAP™ Application Partitioning Example

The following example illustrates the partitioning of an OMAP processor for speech applications as HMI application as it is required in Telematics systems.





Information from the system is provided via Text to Speech while commands from the user are taken via speech recognition. The DSP does the pre- and post-processing. These require typical digital signal processor algorithms, like filtering to reduce the noise etc. In multitasking the DSP continues the processing of the audio streams, the radio and GPS functions. The ARM processor does the grammar parsing and acoustic models and is also responsible to search in the large external vocabularies to find the right match. On the ARM a typical operating system is executed while on the DSP the DSP/BIOS operating system is used. The reference frame work provides the necessary functionality to exchange data between both processor cores, and of course the run the required applications on the DSP, controlled from the ARM processor.

Both processors provide the right mix of features to allow optimized execution of the various applications. Together they form a heterogeneous system with with following portioning:

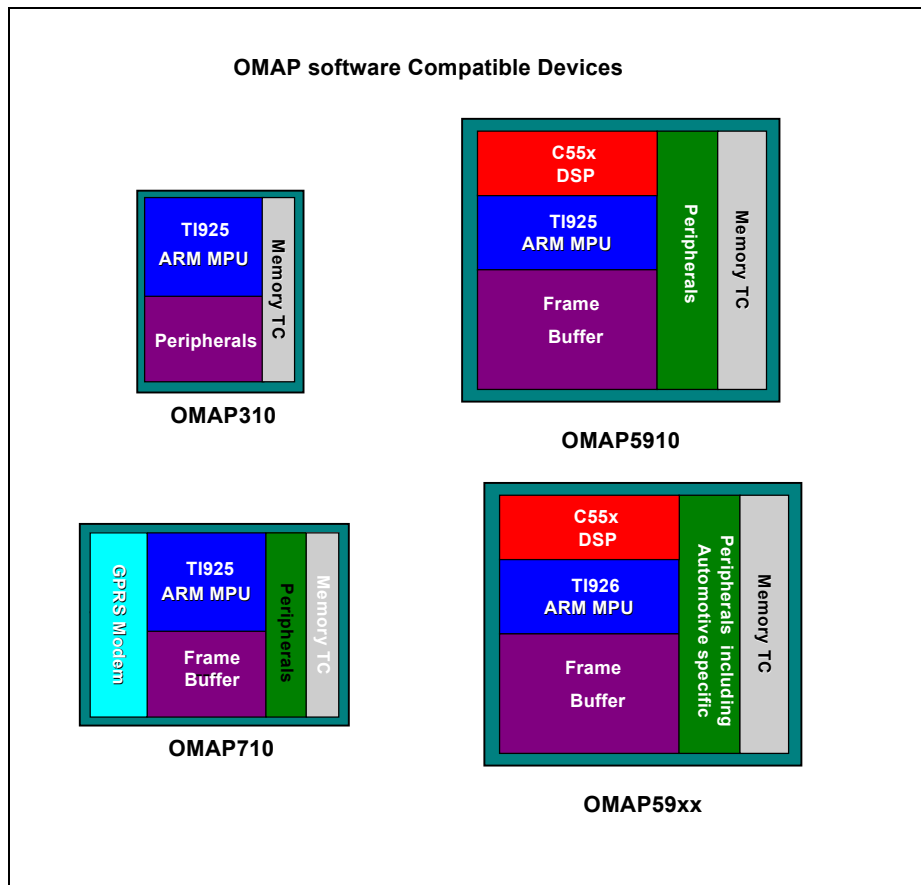
- ARM is system master and controls the system resources:
 - Shared and shareable memory (Not internal DSP memory)
 - Access to shared peripherals
 - DSP MIPS (ARM controls distribution of computing tasks to the DSP)
 - ARM peripherals (even when the end target of the I/O is the DSP)
 - ARM - DSP data streams
 - Controls of all DSP within the system
- DSP connected via a DSP software bridge to the ARM:
 - DSP controls its own internal memory and peripherals
 - DSP does NOT assume control of other system resources
 - BIOS and eXpress compliant algos

Evolution of the OMAP architecture

The OMAP architecture has its origins in TI's 2G wireless baseband devices, originally based on an ARM7 MPU and C54x DSP as a power-efficient and adaptable solution, interfacing to GSM and various other mobile phone standards throughout the world. The new wireless standards 2.5G and 3G

demand more processing performance, in order to offer capabilities such as 384 Kbps data transfers. In addition, wireless technologies like BlueTooth and 802.11 offer alternative and sometimes complimentary means of wireless data exchanges.





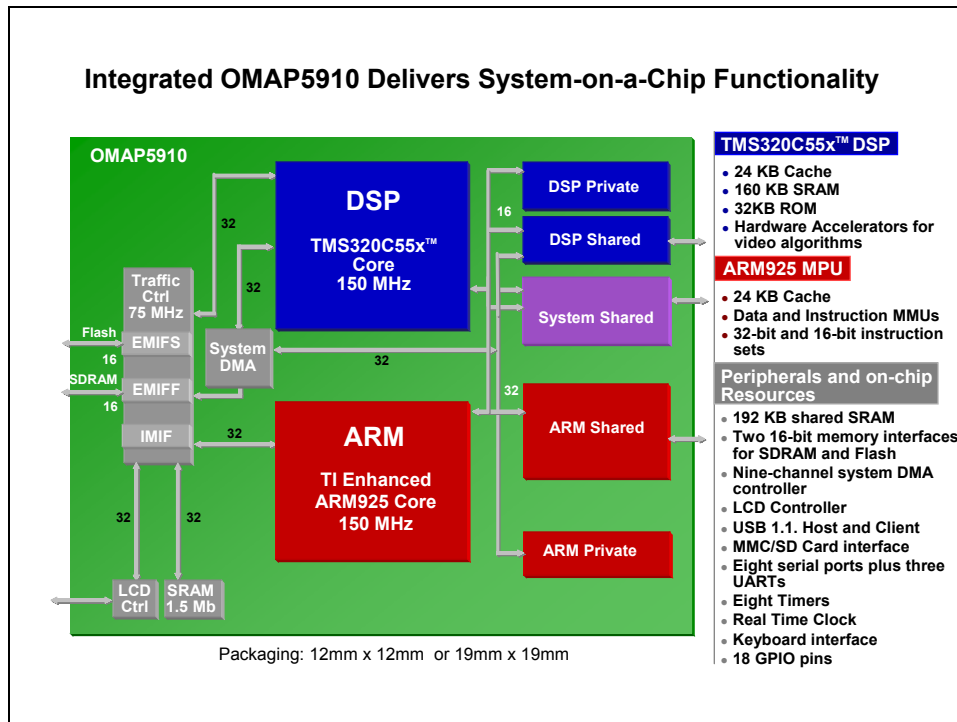
The family of devices offered by the OMAP platform provides a common TI-enhanced ARM microprocessor and a collection of peripherals and memory interfaces. Additional multimedia performance is provided by a C55x DSP.

The latest development is the OMAP5910, which is with some modifications the basis for other, application specific solutions. The next chip, the OMAP59xx will have dedicated automotive peripherals, e.g. a CAN interface. It will be based on an ARM926 RISC and c55x DSP processor kernel. This combination of computation power with the right mix of peripherals will provide the necessary performance for speech interface based radio navigation system as previously defined as a typical Telematics system.

OMAP Architecture Details

A general-purpose processor (GPP or MPU) runs applications under an operating system such as WinCE, Symbian, QNX and many others. Support for a variety of industry-standard operating systems provide the sophisticated development environments and allow reuse of software and experience for software teams. Standard programming interfaces (APIs) are called to perform a variety of multimedia and navigation operations. Depending on the complexity of the operation and available system resources, the API may perform the operation on the spot, offload it to a digital signal processor (DSP), direct a system direct memory access resource (DMA) to perform it, or make use of a custom peripheral such as security module. DSPs are ideal for performing mathematically-intensive real-time tasks such as MP3, AAC, and many others or even video decoding. DMAs automate the transfers of data from larger external memory to fast internal memory (and back), or between peripherals and memory, without requiring MPU or DSP intervention. Custom peripherals perform relatively simple, well-defined tasks extremely efficiently.

By delegating work to those most efficient at performing it, overall performance is enhanced, and power reduced by idling those resources not in active use. It preserves software compatibility while allowing for flexible tradeoffs between performance, power-consumption and system-cost. The use of a programmable DSP ensures adaptability to ever-evolving protocols, formats, and standards such as digital music.



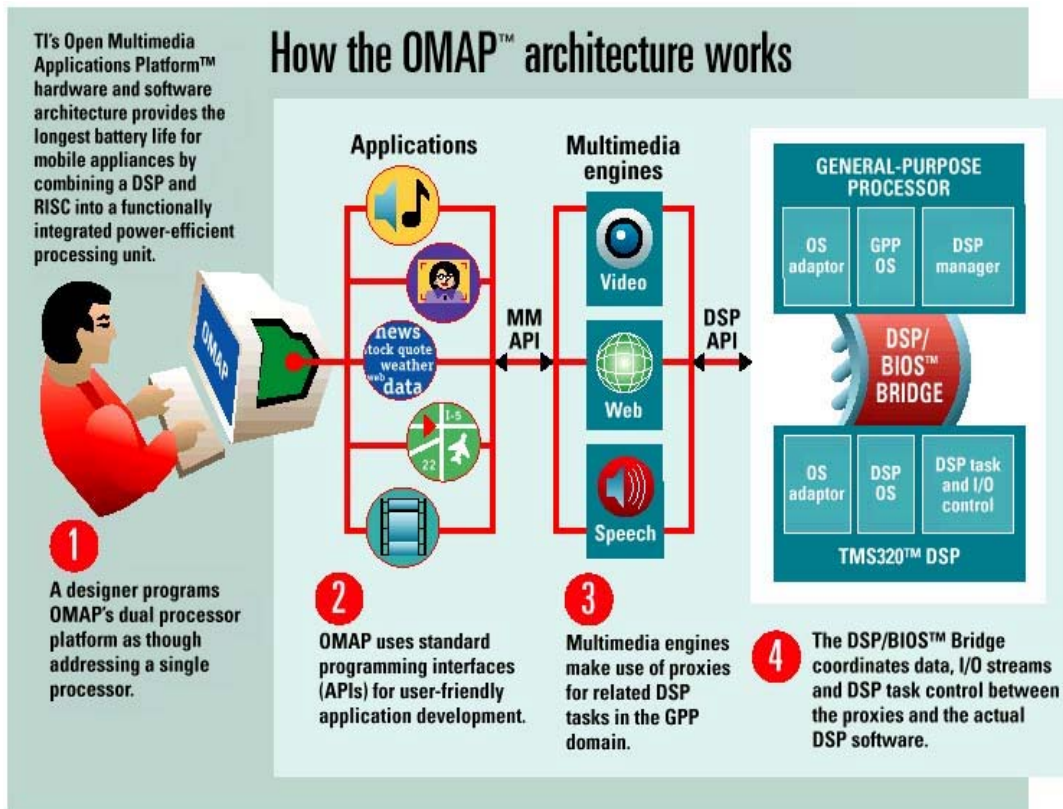
The GPP, DSP, and DMA could each be attempting to access memory, such as external SDRAM / Flash or internal video frame buffers, or the memory-mapped registers of peripherals. The Traffic Controller (TC) will arbitrate between these requests and direct simultaneous accesses on different buses to maximize data throughput for the system.

To manage the offloading of tasks from the GPP to the DSP, software components such as multimedia engines utilize DSP/BIOS link, an TI-provided API designed to manage multiprocessor communications.

Examples of media engines are either available or in development include:

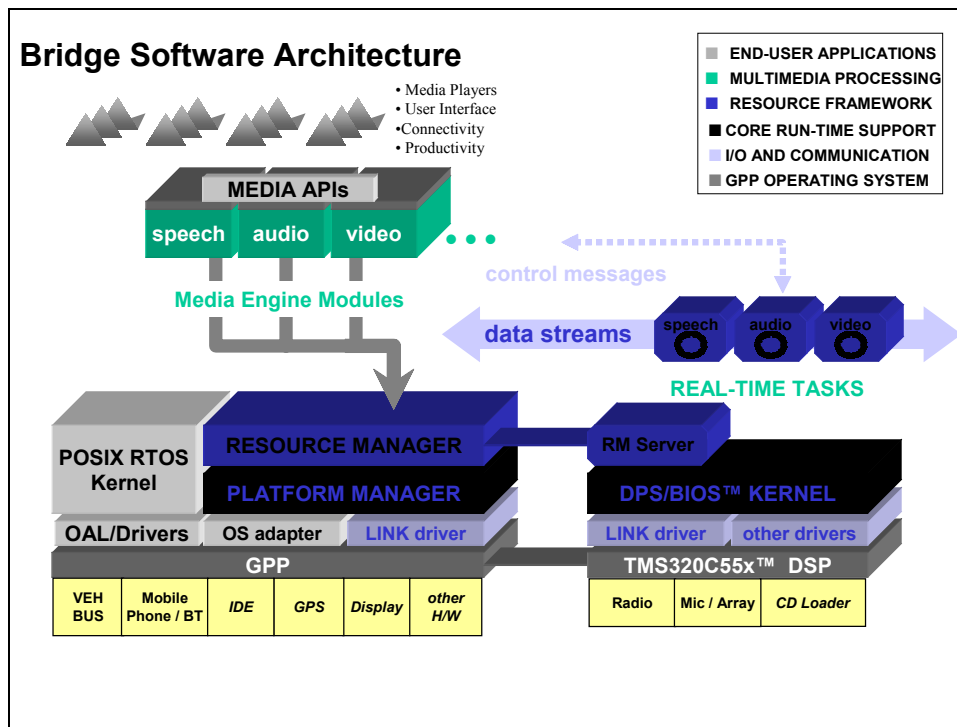
- Audio: MP3, AC, WMA, RealAudio, and MIDI
- Video: JPEG, MPEG-4/263, GIF, JPEG2000, and RealVideo
- Speech Recognition: Speaker independent or dependent, noise suppression, echo cancellation.
- Security
- Location-Based Services
- Vocoders: AMR

Because of the programmable and open nature of the OMAP architecture, new media engines can be developed by anyone, and DSP/BIOS Link helps ensure that multiple media engines can coexist.



Software development for Radio Navigation Systems on OMAP

A developer would create an application to run under an embedded operating system, such as WinCE or Symbian Quartz. When function calls are made to an API such as a subset of OpenGL, the ARM MPU would collect the data and pass these buffers to the DSP engine, using DSP/Bridge to handle the details of allocating and managing the DSP resources used for this task. While the DSP takes care of the numerically intensive graphics algorithms, the MPU resumes executing the remainder of the game processing. This dual-processing model is similar to that found in a PC with a 3D graphics accelerator card. A variation on this approach is to have the ARM processor perform the geometry portion of the 3D pipeline, while the DSP performs the rasterization portion.



Code Composer Studio for OMAP Debugging

The OMAP Code Composer Studio (CCStudio) Integrated Development Environment (IDE) provides debug support for the OMAP platform via heterogeneous debugging of the ARM926 and TMS320C55x-based DSP subsystem cores, which are connected on the same JTAG scan path within the device. Simultaneous debug of two or more CPUs, sometimes referred to as co-emulation, allows the user to coordinate debugging between both of the processors. The Parallel Debug Manager (PDM) control within Code Composer Studio IDE can be used to execute the targets in parallel, i.e., stepping or running. It can also be used to configure global breakpoints. This feature allows you to designate both processors to halt when either processor halts. Global breakpoints and co-emulation are very useful when trying to debug code that provides communication or signalling between processors.

This application report describes how to setup Code Composer Studio using the heterogeneous device driver, and how to start debugging an OMAP platform using the Parallel Debug Manager. Although some of the values will be different for other OMAP platforms, the process contained in this application report can be used as a framework for those devices.

Summary

TI's OMAP architecture provides the programmable solution necessary to address the ever-changing landscape in Telematics Radio Navigation systems. As the requirements continuously increase (better graphics, sound, performance), software will need to run on a range of platforms at different price points, and preserve the software development investment across several generations of products. The OMAP architecture provides a framework where flexible software components can be used across different Oses and devices, providing additional features and performance in a backwards- and forwards-compatible manner.



References:

Wireless Games & Entertainment, Hervé Maréchal, David Matt, Texas Instruments

<http://www.omap.com>

<http://dspvillage.ti.com>

<http://www.arm.com>

