

## Efficient Architecture for Telephony Applications using multiple TMS320C5x cores

Robin Getz

Senior Field Applications Engineer  
National Semiconductor Corporation

[rgetz@nsc.com](mailto:rgetz@nsc.com)

**ABSTRACT:** *Today's embedded systems designs often incorporate more than one microprocessor / DSP combination. These are generally very tightly coupled and run in parallel, handling diverse tasks simultaneously. The need for a full industry standard DSP working in concert with a processor engine has been required for a multitude of applications. A brief overview of a test chip that validates the multi-processor model will be presented, which combines the industry standard Emerald (TMS320C5x compatible) DSP core with a RISC engine, a optimized solution for communication systems.*

An example application of a tightly coupled microprocessor / DSP combination is the multi purpose printer / scanner / fax machine. The main microprocessor controls the scan / copy functions and drives the printer or scanner engine, while a DSP handles the fax modem tasks. These increasingly complex systems are able to accomplish digital signal processing, general computing and control tasks in parallel. This is usually done at the expense of development time and device cost.

In order to reduce costs, there have been two strategies to resolve the performance requirements. One is to integrate DSP operations into general-purpose processors and the other is to put a DSP side by side with a general-purpose controller.

Various semiconductor companies have tried to answer this requirement with MAC operations within the pipelines in their RISC

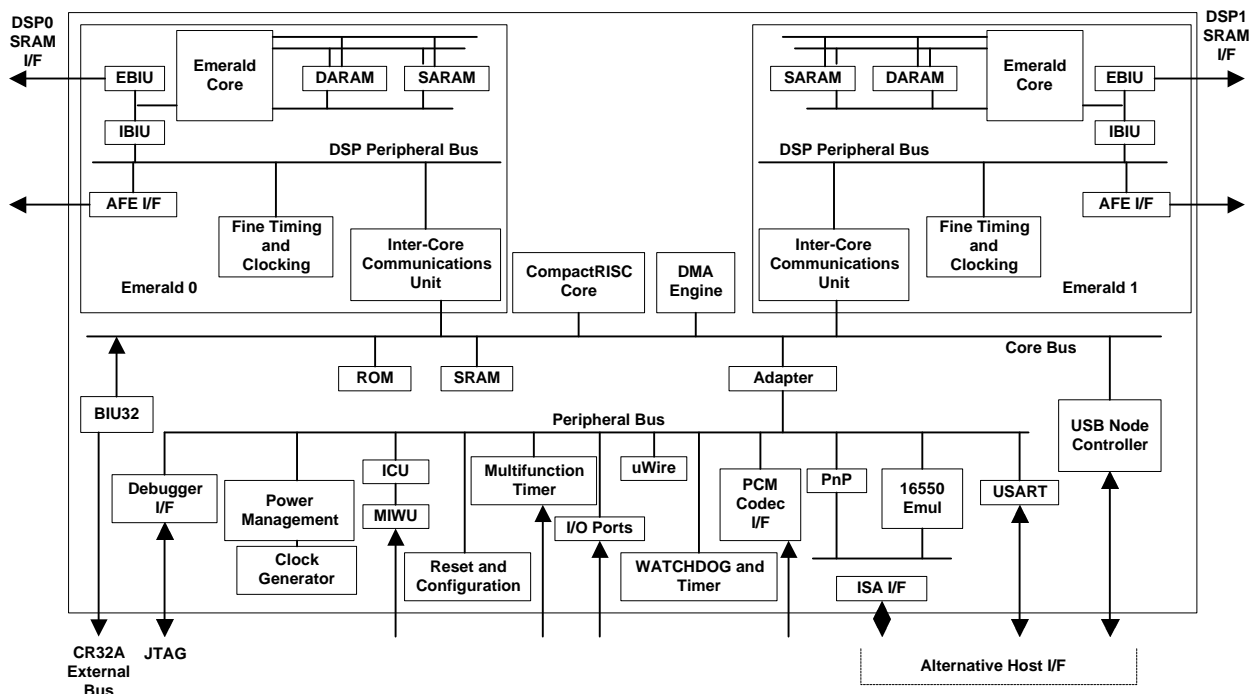
cores (ADI's Shark VIS and NSC's NS32GX320) or adding a separate pipeline for the MAC operation (ARM Piccolo, Pentium™ MMX Processor and Hitachi SH-DSP) usually with limited success. Both solutions make the processor more expensive, and are lagging behind the requirements of modern DSP processors since they can not perform a MAC in parallel with complex address calculations, two loads, a store, saturate and shifting. Also, the non-standard programming model prevents leveraging existing DSP software.

Other architectures have placed non-standard DSPs beside RISC cores, requiring software engineers to re-write all of their hand-tweaked assembly code, delaying products to market. Lack of support for a parallel programming model makes software development extremely difficult.

### ***Normandy Overview***

In order to validate this idea and have a multi-core platform targeted at communication applications, National created a device code named Normandy. With this chip National took the approach of using standard processor cores and integrating them into a software friendly parallel system. This integration included two elements:

- an inter-core communications unit (ICCU) that provides a unified address space for the processors and interrupt driven signaling and semaphores; and
- a multi core debugging environment.



*Normandy Block Diagram  
Figure 1*

The unified debug environment supports the debug of each core by itself without impacting the operation of the other cores, or concurrent debug via means such as breaking all the cores at the same time.

Normandy solves the requirement of coupling DSP and microprocessor without slowing down software development. Normandy has two full Emerald DSPs (TMS320C5x compatible) and a CR32, CompactRISC™ 32-bit RISC core. This processing power is supported by an Inter Core Communications Unit (ICCU), local memories for each processor, USB node controller, ISA host interface with full PnP support, AFE interface, power management functions, interrupt control unit, 16 channels of DMA, various multi-purpose timers, WATCHDOG™ and a common JTAG based debugging port.

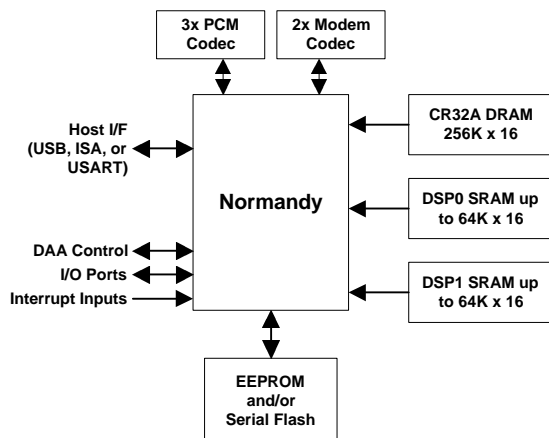
The industry standard Emerald DSP cores on Normandy allow software engineers to keep their software investment in hand-crafted DSP assembly and put their non-critical control or general computation code in a high level language like C in order to run it on the

CompactRISC core. This not only speeds their product to market, but also allows for greater flexibility and maintenance on their code base in the future.

Each of the processors has a local bus with local resources (memory and peripherals). This localizes the processor's memory traffic allowing full parallel operation, maximizing the overall system performance. This ICCU bridges between the processors allowing access to the other processors' local bus and resources. This enables flexibility in the allocation of system resources for the different tasks, thus simplifying the programming model.

A Normandy-based system can have up to five external codec devices. Two of these can be high-quality modem codecs. The remaining three codecs are PCM codecs. These PCM codecs are used for PCM quality voice I/O. Code and data are stored in either a DRAM connected to the CompactRISC external bus, or two SRAM devices connected to the DSP external bus. An EEPROM or a serial flash

device is used to hold configuration information. A typical block diagram is shown below.



*Normandy System Block Diagram*  
Figure 2

The memory and I/O devices are directly mapped into the 4 Gbyte CompactRISC address. This includes code, data and peripherals for both Emeralds and the CompactRISC bus.

### ***Emerald Compatibility***

The Emerald is binary compatible with TMS320C5x - Texas Instruments popular family of DSP processors. Thus it supports the complete instruction set of the TMS320C5x and is cycle-by-cycle compatible in terms of instruction execution. The Emerald core was enhanced with a few extra features:

- **BIU:** the Emerald bus has been enhanced to support a glue-less interface to memory devices off chip for both the Emerald and the CompactRISC cores
- **Address Map:** The Emerald core is designed to enable customization of the on-chip memories to the need of the application in each implementation. Special support was added to enable easy mapping of on-chip devices and communication to the CompactRISC bus.
- **Peripherals:** The Emerald core includes the legacy timer and interrupt control unit. Other peripherals are added as required by the specific application. In the Normandy, the AFE and Timing Units have been added

to support the communication application needs.

- **On-Chip Debug Support:** The Emerald supports a software monitor (EMON) that resides in memory and also communicates with the debugger via a JTAG based port.

### ***Inter-Core Communications***

The ICCU connects the CompactRISC core bus to one Emerald bus. The ICCU allows the cores to share memory, transfer messages, and keep coherency of shared data. Messages are transferred using a mailbox mechanism and coherency is maintained with a set of hardware semaphores.

Where each respective core usually accesses devices that are directly connected to its local bus (core, peripheral, and external buses) the ICCU allows connection to a non-direct connect device. The Emerald can access devices connected to the CompactRISC buses by performing fetch, read, write or I/O operations from the communications window. In such cases, the ICCU serves as a bridge to the CompactRISC bus. When the CompactRISC core accesses Emerald address space, the ICCU translates the address and performs the requested action.

The ICCU also contains three sets of 4-bit general use semaphores. These semaphores support test-and-set and test-and-clear operations between the Emerald and the CompactRISC core or the other Emerald.

### ***Development Systems***

Developing with Normandy can seem like a daunting task, not only because of system complexity, but because debuggers have typically not been coupled in order to debug multiple DSPs or CPUs. Typically, debugging designs such as this means that you must use separate debuggers and pay close attention to deal with failures and exceptions to ensure you know the status of each processor. Without that knowledge, failures induced by actions taken by other processors can be elusive and consume a great deal of product development time.

Normandy was developed with the software development and debug effort in mind. The task of debugging three tightly coupled processor cores all running independently has been made easier by implementing a multi-core debugging methodology in the chip.

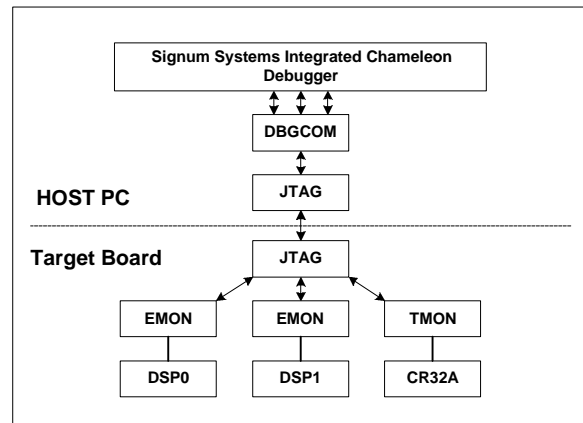
National's multi-core based Debugger Communications Interface (DBGCOM) enables multiple cross-debuggers, running on a single host PC with Windows 95 or NT, to interact with multiple cores integrates on a single chip.

Rather than the debugger connecting across a physical interface between the host PC and the processors used in the target system, the DBGCOM resides between the debugger and the physical communication interface, thus isolating and providing the transport layer in the communications model.

The processors run a monitor residing on the target board (TMON or EMON) to communicate with the debugger through a pre-defined API which enables debugging. When used to debug with Normandy, DBGCOM allows the connection of several multi-threaded virtual communication channels, one to each of the processors in the chip. DBGCOM is reentrant so that you can invoke the program multiple times simultaneously. Each one is transparent to the other caller of the program. This allows several debuggers to be active, communicating with DBGCOM at the same time.

Signum Systems <http://www.signum.com> offers a debugger compatible with DBGCOM and Normandy. With this debugger a single window can display the status and configuration information about the multiple cores at the same time. You can run emulation and even single step through instructions in a coupled debugging environment.

When debugging, breakpoints set for one processor are interactive, allowing an error condition in one processor to trigger a breakpoint in another processors debugging routine. The resulting closely coupled debugging environment allows development and testing of multiple processors in virtually any application.



*Normandy Debugger Communications Interface  
Figure 3*

### ***Example Application using Normandy***

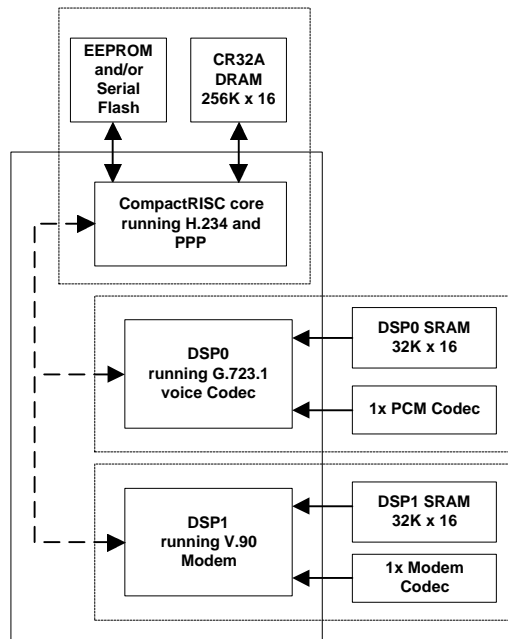
A typical application that would be well suited for Normandy is a residential Voice Over IP (VoIP) telephone, sometimes called an "Internet Phone".

With a VoIP telephone, when the user dials a long distance number instead of connecting to the public carrier network, the phone automatically connects to an Internet Service Provider (ISP). After connecting to the ISP, the voice conversation is packetized and transmitted via TCP/IP using the ITU standard protocols across the internet where the receiving phone rings. These long distance voice (or fax calls) could easily add up to expensive monthly costs for an individual.

The ITU voice codec used is G.723.1 which provides compressed data rates of 6.3 and 5.3 Kbps and VAD/CNG "silence" compression which can further reduce the average bit rate. The G.723.1 protocol is available for the TMS320C5x Family of DSPs from The DSP Group <http://www.dspg.com>

The system partition would have a DSP run the voice codec, the other DSP run a standard V.90 modem, and the CompactRISC core run the H.324 and PPP protocols in order to communicate with the ISP. A block diagram of the proposed system is shown in Figure 4. Because the solution is using industry standard DSPs, with readily available software (both V.90 and G.723.1) the solution will take

minimal engineering and debug effort and should be on the retail shelves soon after beginning product development.



*Proposed Voice over IP solution based on Normandy  
Figure 4*

## Conclusion

The combination of standard cores and peripherals makes a Normandy derivative device well suited for applications targeted around multi-media, high speed communications or numerous other applications. The ability to run more than one DSP on a single chip, and have coupled debug environment is a large step in the requirements for today's system engineers.

Normandy is proof of concept from National with the Emerald DSP core on it, and provides an alternative for systems on a chip solutions that many OEMs are requiring.

*Robin Getz is a Senior Field Applications Engineer located in Portland Oregon, who worked on the Normandy team. He has been with National for the past 5 years, previously working as a designer. He can be reached at [rgetz@nsc.com](mailto:rgetz@nsc.com).*

## References

Falik, Ohad, [Ohad.Falik@nsc.com](mailto:Ohad.Falik@nsc.com); *Normandy Specification* National Semiconductor; Internal Document (NDA required)

Kedem, Rafi, [Rafi.Kedem@nsc.com](mailto:Rafi.Kedem@nsc.com); *National Provides Debugging Solution to Tightly-Coupled Multi-Processor Applications*; EETimes, [http://www.national.com/appinfo/compactrisc/eetimes\\_092997.html](http://www.national.com/appinfo/compactrisc/eetimes_092997.html)

*Products and company names mentioned herein may be the trademarks of their respective owners.*