



White Paper

Integrating ARM Microprocessors into Advanced Digital Products

Bluewater Systems Limited
PO Box 13 889
Christchurch
New Zealand

Tel. +64 3 377 9127
Fax. +64 3 377 9135
solutions@bluewatersys.com

www.bluewatersys.com

Abstract

Designing electronic products is becoming more complex as markets demand more features, higher integration and short design cycles. The traditional monolithic approach to design (which long ago ran out of steam in silicon design) is become more expensive, risky and inflexible. This paper describes the problems with the current approach to design and how a new modular method can provide significant benefits.

The Market

We are familiar with what seems like an explosion of features in electronic devices. Perhaps cellphones are leading the charge, with build-in high resolution digital cameras, Bluetooth, MP3 player, PDA, etc. But other markets are little different. Even in the custom embedded space it is common now for customers to demand ethernet, GPRS modem, WiFi or large amounts of storage.

Many of these designs are based around ARM, a low power high performance technology which is used in the iPod and Playstation Portable, for example. Around 1.3 billion ARM chips shipped in 2004, many in products with an enormous array of features.

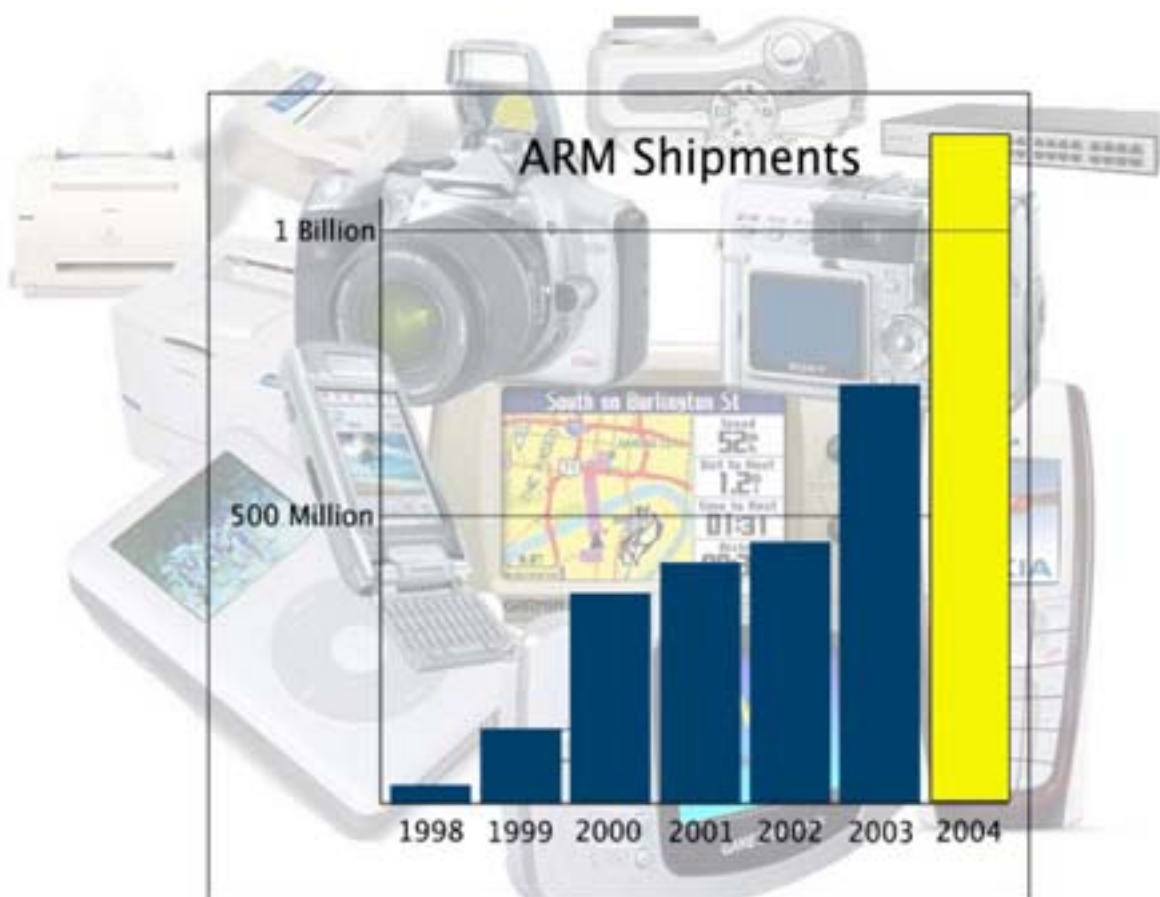


Illustration 1: The explosion of features and unit volumes in modern ARM-based consumer electronics



Illustration 2: A large number of companies are licensed to ship ARM chips

Design Challenges

But if you have not heard of ARM you may be forgiven. Markets are focussing on the peripherals, not the core. Unlike the much smaller PC market, the speed or type of the processor inside is of little concern to the average consumer.

Yet the processor and memory system are not trivial parts of the design. If anything they are more complex. While it may take a few months for an engineer to come up with a new (and correct!) processor/SDRAM/flash/PSU design, a GPRS module can be hooked up to an available UART in a few hours.

In fact the design time involved in the 'core' of a new product is rapidly becoming a significant distraction. Electronic design engineers are not cheap, and the core design time extends the cost and time of development. This can result in a loss of revenue as it increases time to market. There is also the risk that a fundamental error (perhaps in the SDRAM interface) may require an expensive, time consuming board respin, or leave the product not fully functional as intended. New product features can involve a similar risk, particularly if they affect board layout.

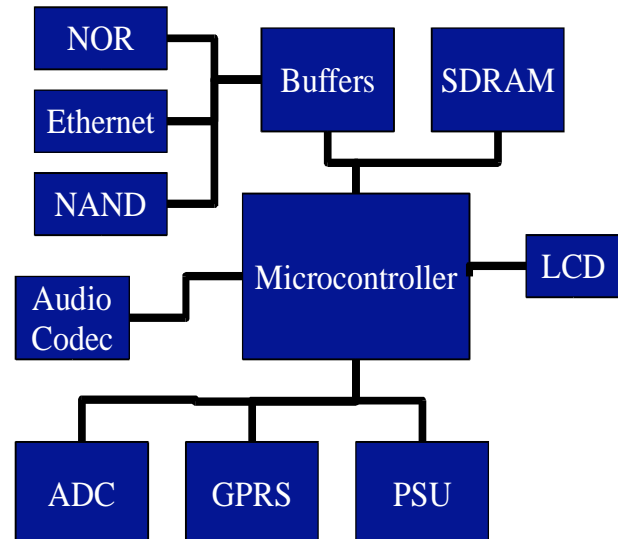


Illustration 3: Block diagram of a typical embedded system

Those companies who struggle with a number of product variants in the market know how hard it is to upgrade the processor in all of them at once! Making variants to fill every market niche is fraught with difficulty. Despite the fact that the core circuit may be very similar, each needs a different full

test rig, software build to cope with minor circuit changes, support and maintenance and an ongoing roadmap. But sharing the same board across many designs is also problematic. Customers may complain when they can see the ethernet is tracked out in the low end boards, but not available for want of a \$6 part.

In addition, production costs must be considered. It is tricky to fully test a high speed CPU circuit in production, and it certainly adds to the cost and complexity of test rigs and test software. Latent issues can cause intermittent problems in the field, and can be very difficult to track down.

Modern processors may also have complex power supply requirements. An Intel PXA270, for example, simultaneously requires 1.1V, 1.3V, 1.8V or 2.5V, 3.0V, 3.3V supplies as well as a variable 0.65-1.55V supply! Even just designing the power supply for this is obviously going to be a lot of work and fraught with risk.

Making a revision to a monolithic board may involve re-laying out carefully hand-crafted areas of the board to fit in a new component. With an 8-12 layer board, the cost involved in this can be high. As BGA parts become more common the technology needed to build reliable boards is becoming more advanced. It isn't always possible to use 4 layer through-hole (PTL) or stick to TQFP packages. Often it is the processor and memory chips which force the use of BGA, requiring a more expensive board just to cope with these items.

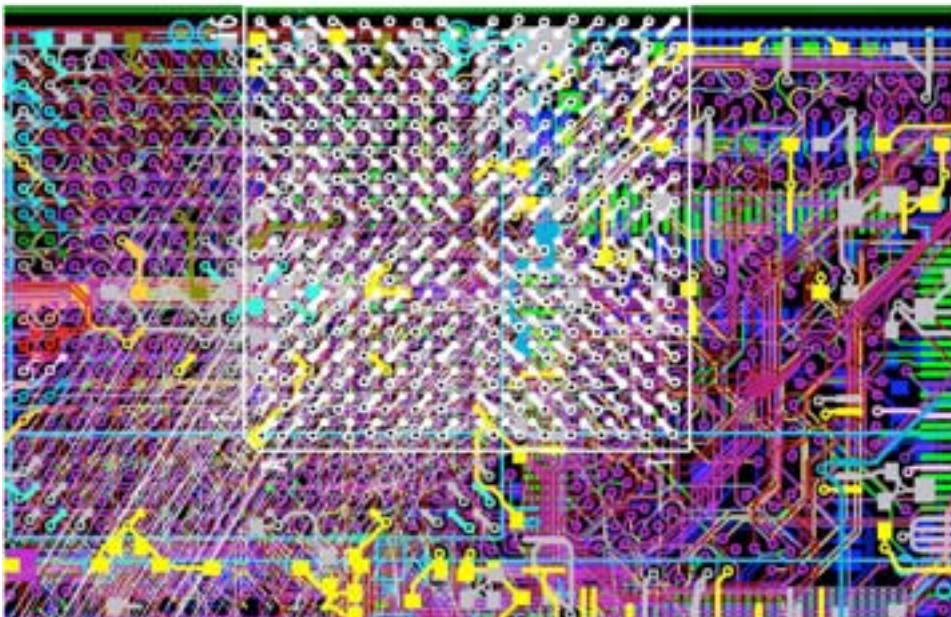


Illustration 4: Multi-layer boards can create enormous problems when components must be moved. In this 12 layer, a chip has to be moved to make way for a connector, thus requiring more than a week of layout work

Yield problems can increase costs and push the timeframe out just when the product is about to hit the market. These problems are much more difficult and expensive to manage, and much more damaging, when they involve BGAs, high layer count PCBs and tiny surface mount components which are hard enough to see, let alone solder. Yet without these advanced components, it may be hard to meet the processor design guidelines, or the form factor requirements of your market.

Our new Requirements

In summary design engineers are spending so much time, effort and money worrying about the processor/memory part of the design, they are bound to be distracted from their real purpose: efficiently and quickly putting together a product which integrates the required features.

So what can be done to alleviate these problems? After all they are not unique to the PCB design industry. In chip design, OEMs long ago gave up designing everything from scratch each time. For example, the chip inside iPod includes two ARM cores, a MP3 audio decoder, LCD controller, USB and firewire device ports among other items: none of these was designed from scratch for the iPod. All were made available as modules. Such Intellectual Property (IP) can be purchased, verified in an FPGA system such as ARM's Integrator or Versatile platform, and then sent to TSMC and the like for fabrication. Often these chips (which are at least an order of magnitude more complex than today's PCB designs) come back and work first time!

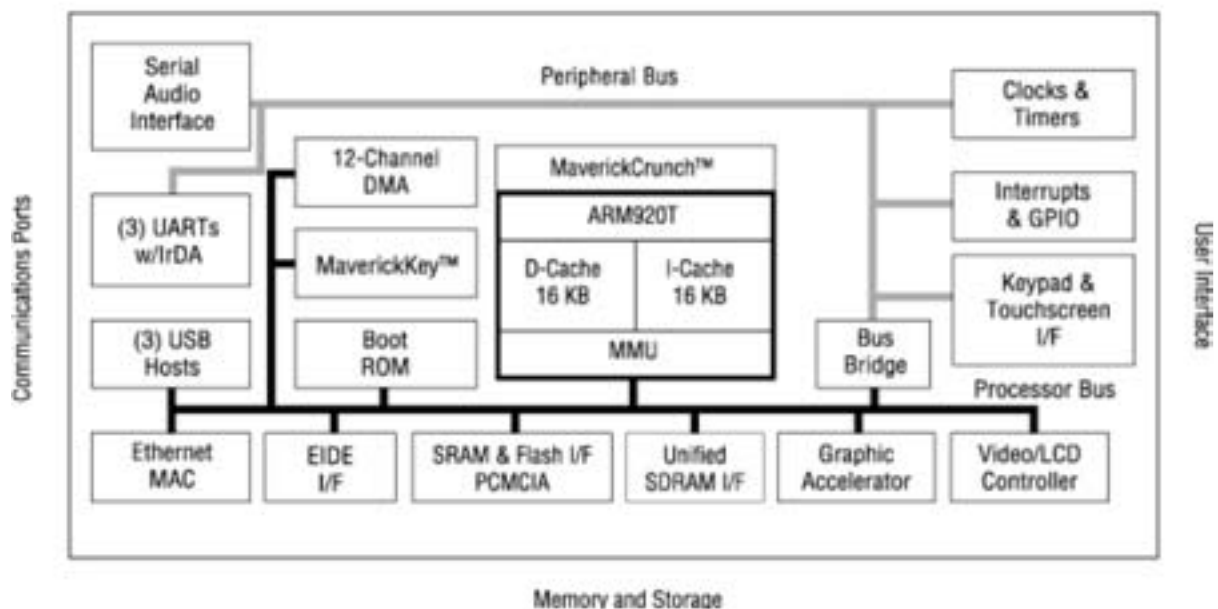


Illustration 5: Chip design relies on IP blocks as this Cirrus Logic example shows

We need a way of packaging up the 'common' parts of a design so that they become invariant. This would result in faster product revisions and faster design time, since we wouldn't have to worry about these each time. Production test would be much less of a headache, since the common part could be pretested. We could use 4 layer boards with simple surface mount technology, thus lowering production costs and expanding options as to which PCB manufacturers we could use.

Ideally our product should be just a connector board, containing connectors and interfacing for the peripherals as dictated by the application requirements. The 'common' parts of the design could then plug into that board. These 'common' parts of the design would come with ready made software so the bring-up can focus on the peripherals on the real value-added parts of the design.

Obviously, the smaller the 'common' parts can be made, the better, since we don't want to have to expand the product just to fit these in. Height also matters, in terms of clearance around other components and the product casing.

The System Module

Enter the Single Board Computer Module, or simply System Module. This consists of a small plug-in board which provides the invariant circuitry: processor, SDRAM, NAND/NOR flash, basic peripherals, exotic power supplies, etc. Software such as demonstration code or (for higher end boards) a full-blown Linux kernel is typically supplied on CD ready to go. This software would normally run essentially unmodified on your connector board with the module plugged in.

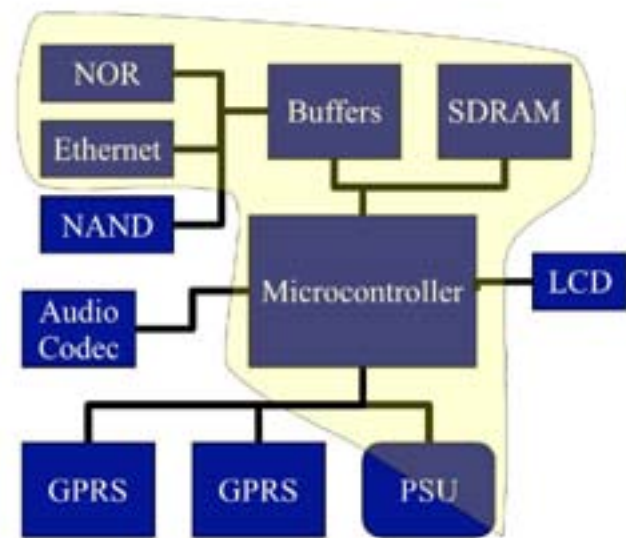


Illustration 6: Typical embedded system design based around a System Module. The coloured area is incorporated in the module, thus significantly speeding up the design process



Illustration 7: A System Module (connectors on the reverse side)

The System Module provides a great many benefits. It can reduce development costs (verses full custom design) by 50-80%. It provides an effective 100% production yield on the most complex part of the product. It is low cost because (although it is perhaps a 10 or 12 layer board) it is very small.

The System Module also takes care of all or most of the BGA and tiny surface mount parts, so may allow your connector board to use less exotic technology. It may even provide an easy upgrade as new modules become available, provided they are reasonably backwards compatible.

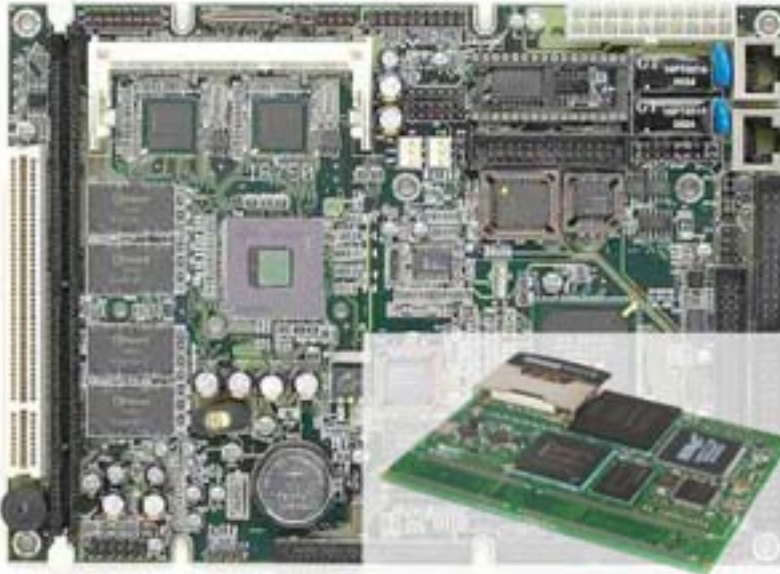


Illustration 8: Comparing a Single Board Computer (SBC) to a System Module, the System Module is significantly smaller and does not have the outside connectors. However, the systems module is more flexible with a much larger range of I/O options.

But isn't a System Module just a cut-down Single Board Computer (SBC)?

No, it is much more flexible. With an SBC you may be limited to certain I/O options (parallel and serial, perhaps a few buffered GPIOs). With a System Module, full access to the processor I/O is available, so there is no need to bodge your product up with intelligent external peripherals. Even the system bus may be exposed, so that extra memory peripherals (such as more memory or a USB chip) can be added. The form factor is smaller, allowing it to be used even in tiny products.

Is there some loss of control by using a fixed circuit designed outside the company? What if an engineers wants to implement something slightly differently in order to achieve a different effect? Well it is true that System Modules may somewhat constrain the design (for example by making decisions as to how certain dual-use processor pins are to be used), but this is seldom a problem in practice.

In fact, for today's high-integrated mid-volume products, it is no exaggeration to say that the System Module is the new way of designing embedded systems.

Available Modules

Over the past few years System Modules have become more popular. Suppliers such as ADS, Cogent, Compulab, Esfia, Gumstix and Strategic Test push their wares, each with different aims and advantages. Most consist of a processor, some SDRAM and boot flash, and some connectors to interface to your connector board.



Illustration 9: System Modules come in many shapes and sizes

Our company, Bluewater Systems has developed the 'Snapper' range of System Modules. Uniquely in the market, Snapper 255 includes an FPGA which supports custom logic, and a removable flash stick for expandable storage memory. It also has buffers so that high speed signals need not propagate to your carrier board and complicate the design. Another product, Snapper 9315 is aimed at the lower end, but still providing a 200MHz processor with high resolution LCD support, PCMCIA and IDE. Both are pin compatible to a large degree, allowing a common carrier board to support both. Snapper is screwed to the carrier board making it immune to vibration.

Bluewater Systems also allows customers to integrate the Snapper design into a single board for higher volume designs. Depending on the application, this becomes cost-effective for runs of 2000-5000 boards.



Illustration 10: Snapper can be integrated into a PCB very quickly using advanced PCB tools

How to use a System Module

Assuming that you have located a suitable System Module, how should it be used to accelerate your project? How do the milestones change and what are the challenges?

In the early part of the project you may not be sure of the processor you want to use. However, these days this is largely determined by the on-chip peripherals and performance that your design requires. You may be unsure of some of these at an early stage (“will my software be able to service all the interrupts and still have time to update the display?”)

The project would normally start with an off-the-shelf development board which uses a suitable System Module. This will provide connectors for the peripherals you require, and perhaps a number that you don't. The board should come with software to help you get started right away, and the supplier can either recommend or provide a suitable development environment. A good development board will support a range of LCDs, ethernet (for efficient software download/development if nothing else), JTAG, plenty of memory and easy access to expansion headers.

You can get started on feasibility with this board, trying out software (perhaps porting existing software) and testing some of the interfaces. Development time/cost and unit costs can be estimated at this stage, particularly if much of the core functionality is provided by the System Module. The supplier may assist with time estimates based on their experience.

Once the project is properly kicked off, software work can start, using the development board as a base. In parallel, the hardware people can get started on developing the carrier board for your particular application. Perhaps the supplier of the System Module can offer example schematics for connecting LCDs, ethernet, USB, modems, analogue chips and other peripherals. If so this can

accelerate design further. This should in most cases be a straightforward 4-layer PCB, so your engineers can spend more time making a really good design which is easy to manufacture and has a low BOM cost!

Once the carrier board design is complete, it must be fabricated and then tested. This is called the bring-up phase and is often fraught with difficulty. However, since the module (where much of the complexity resides) is known to work, bring-up is immensely simplified.

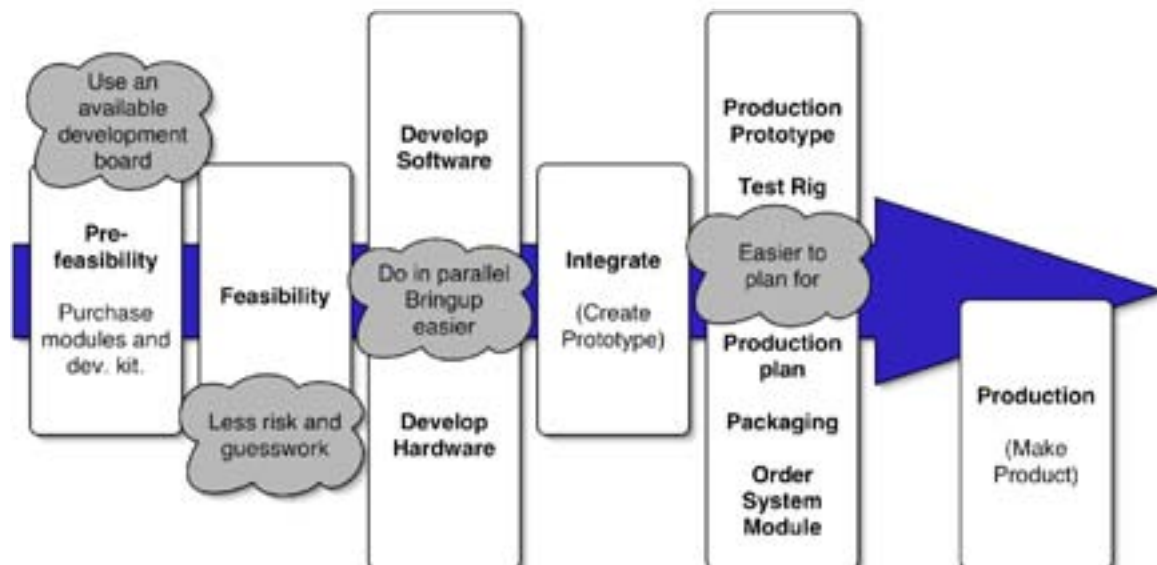


Illustration 11: System Modules enable you to develop faster, cheaper and smarter

Once a prototype board is working, the software will be migrated from the off-the-shelf development board to your new board. This will be a major milestone and should provide a readily demonstrable platform.

At this point (if not earlier), the actual production of the carrier will need to be attended to. System Modules must be ordered. A suitable test rig needs to be designed, but again there is less need to test every part of the System Module since it has already been tested by the supplier. This should simplify test and reduce manufacturing cost.

The prototype may need one or perhaps two respins to obtain the required degree of perfection. Meanwhile packaging and documentation can be finalised.

Overall, the use of the System Module should save time in feasibility and hardware and test design. With the final product it should reduce unit cost and reduce risk of failure in the field. It allows software development to start very early on a reasonable approximation of final hardware. Finally it allows for more accurate estimation of time and cost throughout the project.

Case Study Overview

So what are the actual benefits of using a System Module on your next project. Here, Bluewater Systems can shed some light.

A 2004 project undertaken on behalf of NEC required Bluewater Systems to develop a custom product for interfacing between four different generations of NEAX telephone exchanges, and storage

data on xD cards in a high availability environment. The project included a significant amount of reverse-engineering to understand the intricacies of each protocol and interface. NEC also required access to prototype hardware for several months to verify operation and advise of possibly problems. The time frame from start of feasibility to delivery of completed, boxed units was 9 months.

Without the Snapper module, it is fair to say that these timescales would have been impossible at any reasonable cost. However, with Snapper it was a different story. The 115 FPGA I/Os were put to use talking to the NEAX and driving the three xD cards at high speed. The 400 MHz ARM/XScale processor was easily up to the job of controlling the system. The LCD and ethernet controllers enabled a simple user interface for local operations as well as comprehensive remote control. With power consumption a little over a watt, the product required no fan and was able to operate without fault in a hot environment. Snapper included a CD with software tools, boot loader and a linux kernel with full driver support

In fact, largely due to the early decision to use the Snapper System Module, the product was not only delivered on time, but it has been so successful in this part of the world that NEC plans to add more features and sell it worldwide.



Illustration 12: The DSU was developed based around the Snapper System Module. It is shown against the tape unit it replaces

Conclusion

Engineers naturally want to control all parts of their system, and some will distrust outside technology. However, with the design complexity of modern embedded systems this can be a very expensive and risky policy. This paper has described a new approach to design, making use of pre-verified System Modules for the invariant core of the system. This approach assists with many of the problems experienced by engineers, both during design and manufacture.

It is clear that System Modules have much to contribute to the embedded systems market.