

### From the President

The competition in the local loop is still perceived to be a threat to RBOCs, even though some of the marginal CLECs have gone out of business. Is there a silver lining in this dark cloud? The answer is “yes”. Aggressive deployment of broadband technologies in the form of DSL should help sustain local revenue growth for the RBOCs in the near future.

From the available statistics, it is clear that the RBOCs have only scratched the surface in terms of broadband deployment. Assuming \$25/month revenue per

*Continued on page 2*

### Communications Host Interfacing: From Bit-Pushing to System-on-Chip

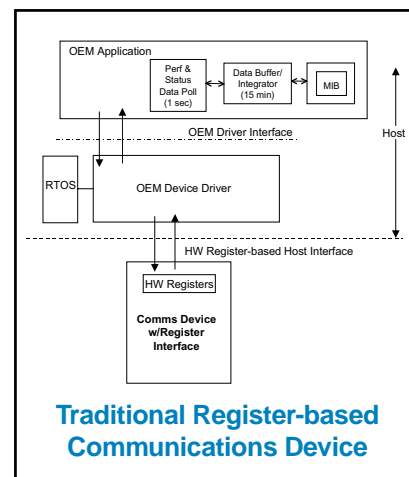
By Brian Stroehlein  
(bstroehl@txc.com)  
Product Marketing Manager,  
TranSwitch Corporation

#### Evolution of the System-on-Chip

For decades, communications (comms) OEMs developed systems using register-oriented hardware and software interfaces because large device geometries and the limitations of system architectures dictated that every usable square millimeter be devoted to the user (data) plane. Hardware engineers made the decisions about what components to include in the user plane. Software engineers normally did not participate in these decisions because, since all devices were challenging and time-consuming for the software engineer to program, there was little opportunity to optimize time, cost, or quality based on comms component selection.

Software engineers, working from the device data sheet and a separate register description document, wrote the system's software starting with hybrid assembler/C handlers, worked their way up to a full device driver,

and—three to six months later—they integrated application code with low-level code and began debug.



**Traditional Register-based Communications Device**

Communications device vendors provided handlers and drivers, but the quality of code supplied was highly variable. As devices became more complex, lack of adequate software support caused endless nightmares for vendors. Vendors complained of unreliable, unusable pseudo-Application Programming Interfaces (APIs); sometimes they had to start over from scratch after giving up on vendor-supplied code, even after hardware design

*Continued on page 2*

### *In this issue*

**Communications Host Interfacing:  
From Bit-Pushing to SoC**

**President's Report.....Continued from page 1**

DSL subscriber, Bell South, which has approximately 26M access lines, can generate \$1B in additional revenue per year with roughly 3M DSL subscribers. This represents only about 11% market penetration in the U.S. In South Korea, DSL penetration is more than 50%.

While this incremental gain in DSL revenue may be less than their potential gain from long-distance entry, the RBOCs are well aware that x-DSL penetration is very strategic for them. They must employ it to halt the progress being made by cable operators in gaining control of residential subscriber accounts with high-speed Internet access deployment using cable modems. In the current depressed economy, control of accounts creates a unique competitive advantage.

We at TranSwitch believe that, as the market recovers, there will be a healthy opportunity for us in the access space as RBOCs and cable operators battle for account control. We already have a number of stellar products for application in access equipment with more in the development pipeline. The EtherMap-3 mapper, which we recently introduced, is one such trailblazing offering, in addition to our CUBIT and ASPEN families of products.

*Santanu Das*

**Bit-Pushing.....Continued from page 1**

had already taped out. Register description documents were often 700 or more pages long, which made writing new code difficult and time-consuming and therefore costly.

**SoCs have changed everything**

Advanced process technologies have resulted in dramatically shrinking geometries, which in turn has made possible the development of system-on-chip (SoC) device architectures, so that user, control, and management planes now can all be included on the same piece of silicon using integrated software and firmware. Conventional communications blocks share a chip with microcontroller, memory, and message interface blocks, all interconnected via an on-chip bus. Resident firmware includes block drivers, link maintenance, alarms, message interface, performance and configuration Management Information Base (MIB) objects, and link signaling, among other functions.

The combination of device firmware and higher-level service-oriented host API software in SoC-based system architectures greatly simplifies device-to-host interfacing and system development. Comms device vendors now are able to integrate control and management planes as well as user planes into their products, and OEMs benefit from shortened development times and lower project costs compared to

traditional multichip methods.

This paper explores and quantifies, in business terms, the value of the SoC approach compared to the traditional method and how the device vendor's product philosophy is affected by the SoC approach. The TranSwitch TEPro SoC is used to demonstrate the benefits of the SoC approach.

**How SoCs Benefit the OEM**

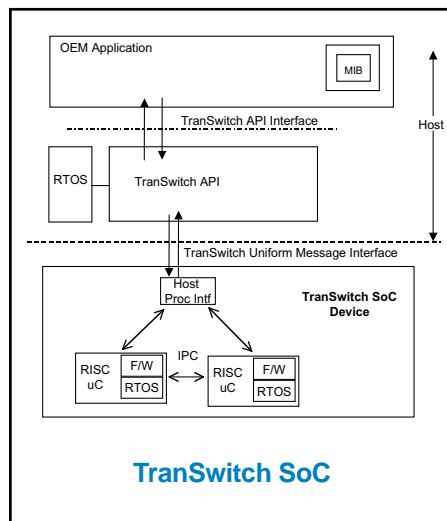
**Abstraction, encapsulation, and device virtualization**

SoC architectures make much more efficient use of abstraction, encapsulation and device virtualization than is possible with traditional multichip designs. Abstraction is the process of extracting essential elements of a device's operation so that they may be modeled or acted upon, while ignoring those elements that are not needed. Encapsulation is a technique to keep together data structures and the procedures that act upon them so they can be presented to the user through a well-defined interface that hides the internal workings. The encapsulated data and functions become, in effect, a distinct software "component". Device virtualization allows application software development to begin before target hardware is available.

**Abstraction**

SoC software and hardware architectures enable abstraction of the register-level, bit-pushing details of a device's operation: the application tells the host API what it wants done, the host API constructs and queues messages for the device, and the device firmware contains the instructions for executing the application's request.

Abstraction means that the underlying operational details such as register maps and feature functionality can vary considerably from device to device within a family, but will appear to change little or not at all from the application's perspective.



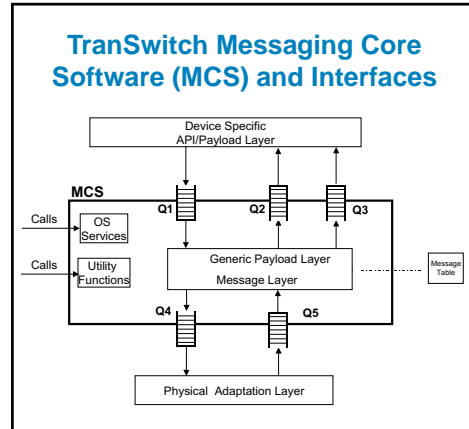
**TranSwitch SoC**

This also makes for excellent application code migration.

A common look and feel to the API among device families further eases the application development task. For example, TranSwitch is adopting a consistent set of base functions and features across its SoC families which:

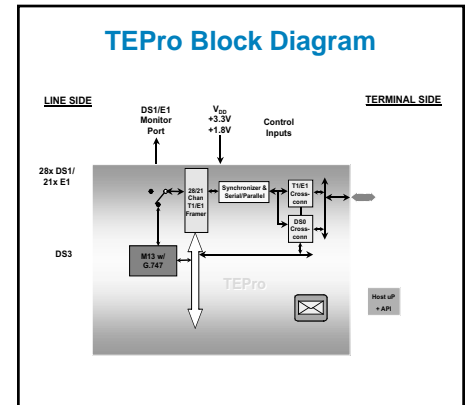
- Provide total abstraction of hardware
- Enable applications to be written using the API spec only
- Enable setting of default configurations with a single API call
- Enable hot spare and non-disruptive switch-over
- Enable non-disruptive host re-boot
- Use message-based interfaces to devices
- Incorporate reusable Messaging Core Software (MCS)
- Support multiple device drivers with a single MCS instantiation
- Support up to 16 different device drivers from the single MCS, each managing up to 255 devices from single host

software that makes it into a component. For example, all communication (except hardware interrupts) between TranSwitch host API processes and TranSwitch device firmware processes takes place via a common Messaging Core Software.



For hardware, encapsulation means enclosing direct register manipulation within the device and enabling service-oriented host communication via a message-based interface. This makes for very robust performance: once a software object or hardware block has been proven and encapsulated, its behavior is predictable and immune to unin-

generation of communications devices is in full bloom. As a first-mover with a true SoC in the comms space, TranSwitch offers an excellent portfolio of field-proven, high-quality software. TEPro is a channelized DS3 device which incorporates several device drivers for running on-chip blocks that were discrete devices in the previous generation.



**Device virtualization**

Through device virtualization, the host APIs for TranSwitch SoCs have compile switches that enable application software development to begin before target hardware is available.

To code differentiable, value-added features for comms devices employing the traditional (non-SoC) approach, software engineers must digest a 700-page register description document, they must have prototype hardware, and they must develop/debug a performance/configuration MIB objects software module before they can begin.

**The SoC-embedded firmware advantage**

For the TEPro SoC, the on-chip Tensilica RISC processor (Link Manager) and TranSwitch-supplied firmware (DD-AMPS™) run:

- Drivers for all functional blocks within TEPro
- DS3 and DS1/E1 facility and path-maintenance Data link messaging
- DS3 and DS1/E1 Alarms per ANSI T1/E1.231

**Generic TranSwitch Driver API Function Calls**

Layer Management	Config/Control	Status/Performance Management
Baseline driver create/delete	Set	Retrieve status
Driver create/delete	Get	Alarms
Device create/delete		
Performance monitoring		
Device attach/detach		
Device reset		
Device default configuration		

**Encapsulation**

SoC software and hardware architectures also enable encapsulation. For software, this means enclosing low-level services and related data within software objects, hidden from view by other objects. Encapsulation is an attribute of a piece of

tended interactions with other objects or blocks. Again, this makes for excellent application code migration.

Of course, SoCs are only as good as the software that runs them, and device vendors with a weak software portfolio and/or poor software development practices will be at a distinct disadvantage when the next

**Bit-Pushing.....Continued from page 3**

- Host Message interface
- Performance and configuration MIB object maintenance per IETF RFCs 2494, 2495, and 2496
- CAS/RBS, SLC-96, ISDN-PRI Layer 1 & 2, and SS7/CCS7 Layer 1 & 2 (incl. FISUs) Signaling insertion and extraction

TEPro's embedded firmware capabilities enable the following features:

• **Device Drivers**

The bit-oriented device drivers for the functional blocks in TEPro reside and run in the Link Manager block, a RISC core with integrated instruction and data RAM. Note that the SoC interface has encapsulated and abstracted the management of more than 90,000 register bits.

• **Message Interface**

The SoC's message interface enables hardware encapsulation – that is, the enclosing of direct register manipulation within the device, which ensures robust performance. All communication to and from the device happens via this message-based interface. Low-level real-time operations not requiring the host's attention are handled within the SoC by its internal Link Manager RISC microcontroller, RTOS, and firmware.

• **Data Link**

The resident firmware in TEPro manages T3/E1/T1 embedded links for remote in-service monitoring and diagnostics as well as signaling and control forwarding. Data link messages are inserted and extracted via the host mailbox message interface.

• **Alarms**

The resident firmware in TEPro handles alarms per ANSI T1/E1.231.

Alarm indications such as OOF (red), RAIS (yellow), AIS (blue), and LOS are passed to the host via the host mailbox message interface.

• **Performance Monitoring**

IETF RFCs 2494, 2495, and 2496 are universally-accepted standards for describing line quality and for configuring lines for service or testing. This is what's expected by the OEM's customer, and there's no real opportunity for the OEM to create new, product-differentiating features here. The TEPro SoC's RISC firmware includes code which provides ready-to-use MIB objects to the host. The OEM software engineers are not required to develop/debug their own performance/configuration MIB objects software module before they can begin coding for the real differentiating features of their system.

• **Signaling**

The TEPro SoC can be programmed for insertion/extraction of CAS/RBS and message-based (SS7, CCS7, ISDN-PRI) signaling via the message interface or terminal-side TDM buses.

Operations Research field during the "Quality Revolution" of the 1980s demonstrated that these three parameters are interrelated, and that true optimization improves all three simultaneously without having to make tradeoffs. All relevant business metrics (except perhaps safety) derive from the simultaneous improvement of these three parameters.

**Make vs. buy is a crucial decision**

In the past, software engineers programmed for whatever component(s) the hardware design engineer selected. Whatever local optimization opportunity there was at the programming level had little effect on global project optimization.

However, for channelized DS3 devices (and soon for many other device types), what once was a component-selection decision has now become a make-or-buy decision, which can have a significant impact on the three key parameters. The components chosen by the hardware design engineer determine whether the project's software engineers spend significant time (and the OEM's money) learning register maps and pushing bits to develop all of their software from the ground up or whether they simply design in a device that incorporates all of the low-level details into a modern SoC interface architecture.

A process flow diagram for a typical component selection process is shown below.

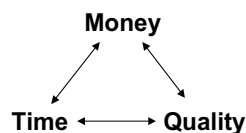
**Communications device vendor product philosophy**

The comms device vendor's product philosophy determines the vendor's choice of hardware and software architecture (traditional or SoC) and their approach to satisfying

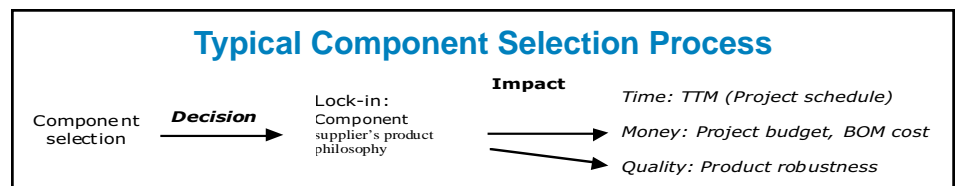
**Design Project Optimization**

**Money, time and quality are inter-dependent**

Normally, projects are optimized based on three key business parameters:



The process optimization work of Eliyahu Goldratt and others in the



OEM project requirements with high-quality, service-oriented software. Product philosophy also determines to what degree standards of software excellence are applied, or if the vendor even perceives software as a critical part of the product.

Since an OEM gets locked into a vendor's product philosophy at the point of component selection, the selection can mean either months of bit-pushing or developing value-added code right away. Thus the component supplier's product philosophy can have an enormous impact on a project's time, money, and quality metrics.

Optimization of time, cost, and quality in a communications OEM project requires:

- Recognizing the hidden "make-or-buy" decision inherent in component selection
- Identifying a comms device vendor whose product philosophy recognizes that the OEM wants to complete projects, not push bits

## Quantifiable Benefits of SoCs for OEMs

Clearly, the component supplier's product philosophy can have an enormous impact on a project's time, money, and quality metrics. These benefits can be quantified as follows:

### Project cost savings

A rough estimate of the value to the OEM of the software TranSwitch provides free with TEPro shows the following savings, based on numbers gathered from TranSwitch's Engineering Group and published salary surveys:

- Avg. software engineer's (MTS) salary: \$87,500/yr
- Burden (@ 40%): \$35,000/yr
- Fully-burdened Software Engineer: \$122,500/yr
- TEPro software "man-years": 6.25 yrs.
- Estimated software value: \$765,625

The software offered with competitive state-of-the-art, channelized DS3 devices employing the traditional multichip approach is of minimal value to the OEM. A survey of OEMs indicates most have opted instead to develop all of their own code from the ground up. When an OEM chooses an SoC device like TEPro, they may gain in excess of \$750,000 in project budget savings alone.

The cost savings for an OEM on a per-device basis amounts to:

3,000 pcs:	\$255.21/pc
10,000 pcs:	\$76.56/pc
30,000 pcs:	\$25.52/pc
100,000 pcs:	\$7.65/pc

Similar savings apply to other TranSwitch SoC devices.

*Note:* Up to 20 percent of the 6.25 man-years' work cited above may be attributable to TranSwitch-specific work that an OEM might not do. However, even if the estimate is adjusted by double or even triple that amount, the project savings are still compelling.

### Speedy time-to-market results in additional savings

Product development experts have developed the concept of "Cost of Delay," which quantifies profit dollars lost per unit time schedule slip. An excellent introduction to this subject can be found at: <http://www.newproductdynamics.com/JPIM5-99/JPIM5-99.pdf>

The article includes advice on calculating the Cost of Delay for a specific project. It also cautions that "...there are no universal values, such as 'six months of delay equals one-third of a product's lifetime profit.'" It should be obvious, however, that schedule slips for next-generation comms systems products could easily cost OEMs hundreds of thousands to millions of dollars a month.

Experience supporting customers designing with our register-oriented Application-Specific Standard Products (ASSPs) and software indicates that a medium-sized OEM hosting interface project typically

takes 12 to 18 months from component selection to initial production. But using our T3BwP SoC (TEPro's predecessor), it took one of our beta customers only five months to go from component selection to production on a new dual DS3 interface/switch card for a product family extension.

As an example, six months off a new product development schedule might be worth:

<u>Cost of Delay</u>	<u>Total Benefit</u>
\$100,000/month	\$600,000
\$300,000/month	\$1,800,000
\$1,000,000/month	\$6,000,000

### SoCs will always be cheaper

Rapidly shrinking device geometries resulting from today's advanced process technologies mean that comms SoC devices can be priced competitively against traditional devices offering similar user-plane densities. In the future, comms SoCs will offload ever-larger portions of the host processor's burden, allowing designers to choose cheaper host processors and smaller memories, thus lowering the total Bill Of Material (BOM) cost.

### Software is the key to a quality solution

Vendors of next-generation comms component products will have to provide much more than just an SoC. Software must be considered an integral part of any SoC offering. Well-designed software is just as important as the hardware in any SoC, and any device vendor with a weak software portfolio will be at an extreme disadvantage in the emerging market. As a first-mover with a true SoC in the non-NPU communications space, TranSwitch offers a portfolio of field-proven, high-quality software. TEPro, our first SoC product, incorporates several proven device drivers for running blocks which were discrete devices in the previous generation.

*Continued on page 6*

**Bit-Pushing.....Continued from page 5**

Device encapsulation and abstraction ensure highly robust performance by enclosing, simplifying, and commonizing details of device operation. This simplifies the device-to-host interface, pushes system development projects forward, and allows engineers to quickly turn their attention to the value-added portions of product development.

**Conclusion**

By using SoCs for communications host interfacing, OEMs realize multiple benefits. Primary among these is the system robustness that results from the use of abstraction, device virtualization, and encapsulation techniques. In TranSwitch's SoCs, these techniques, plus TranSwitch's firmware support of standards, spare the OEM's engineers from having to code low-level details of system operation and allow them to start coding value-added features immediately, with a dramatically reduced learning curve. The result is drastically reduced time-to-market as well as significant savings in project costs.

Bit-pushing belongs in the last century, not in the new millennium!

**Appendix A****TranSwitch has raised the industry standard:**

<b>From:</b>	<b>To:</b>
Software developed to meet the minimum customer expectation	Software developed with the system in mind; by developers for developers
Register-oriented programming interface	Device encapsulation and abstraction
Consuming host MIPS	SoC (System-on-Chip)
Developer writes code per RFCs	SoC (Standards-on-Chip)
Need hardware for application development	Device virtualization capability
Software as an afterthought/ nuisance necessary to generate chip sales	Software as an integral part of the product offering
Arm's length connection to software: contracted-out to 3rd party	Ownership: Software developed and maintained internally
Limited/no hot-swap capability	Software (and hardware) support for hot-swap
Spotty adherence to software excellence principles for interrupts, buffer management, error handling, and semaphores; high probability of unintended module interactions	Software modules are totally self-contained

