USING SECOND- AND THIRD-GENERATION SERIAL RAPIDIO® DEVICES TO IMPROVE 3G+ BASEBAND PROCESSING

By Trevor Hiatt

Serial RapidIO® (sRIO®) has become the embedded interconnect of choice. Wireless telecommunications infrastructure—particularly baseband—has been at the forefront of system developments, with designs using sRIO specification 1.2 and 1.3 being produced by major OEMs around the world.

Although sRIO has served as the low-latency, high-bandwidth, high-reliability interconnect in embedded components, those devices continue to evolve with second- and third-generation switches and endpoints. The current device generation offers enhancements beyond the subset of required sRIO specifications—including the optional sRIO extension specifications as well as proprietary feature sets.

This white paper focuses on how enhancements in second- and third-generation sRIO devices can improve 3G+ baseband processing.

Setting the foundation—standard sRIO feature use in baseband

sRIO standardizes the switch-based, peer-to-peer network. Indeed, the fundamental requirements and functions of the switches themselves are standard per specification. sRIO-based switches allow for best-in-class latency and throughput, and flexible network topologies. The switches can provide modular, flexible backplane support in standards such as Advanced Telecom Computing Architecture (ATCA) or microTCA.

The sRIO standard allows for the synchronizing of all devices—especially endpoints—through the use of multicast event-control symbols. A host can generate the control symbol while the switch is required to forward to all destinations on its output ports with the lowest possible latency through the switch itself. Within baseband, this is very useful for synchronizing all components during frame synchronization.

All sRIO-based devices minimally offer receiver-controlled flow control. This guarantees that devices can handle congestion at input ports at the hardware's physical layer without dropping a packet. Additionally, every transaction is tracked with a transaction ID. Responses to packets sent are also tagged with this transaction ID, and backpressure and transaction completion is handled at the physical layer. In addition, response time during congestion conditions is minimized and packet delivery is guaranteed.

The doorbell packet is also standard in the specification. These packets act as interrupts to endpoints in an sRIO system. In baseband, digital signal processors (DSPs) can use these interrupts to indicate that a full block of In-Phase and Quadrature (IQ) data has been received and processing should begin. A host processor might use the doorbell to be notified of a given system event.

High-level 3G+ baseband considerations

ATCA and sRIO allow modularity. sRIO's switched architecture lends itself to hardware extensibility. OEMs can save cost and support multiple wireless standards by taking advantage of this flexibility. The architecture of choice that has emerged is multiples of baseband cards featuring four or more DSPs aggregated by a single switch board. In our example, a fabric interface chip (FIC) may be used to bridge RapidIO to an air interface protocol, such as Common Public Radio Interface (CPRI™), Open Base Station Architecture Initiative (OBSAI) or perhaps proprietary interface (Figure 1).

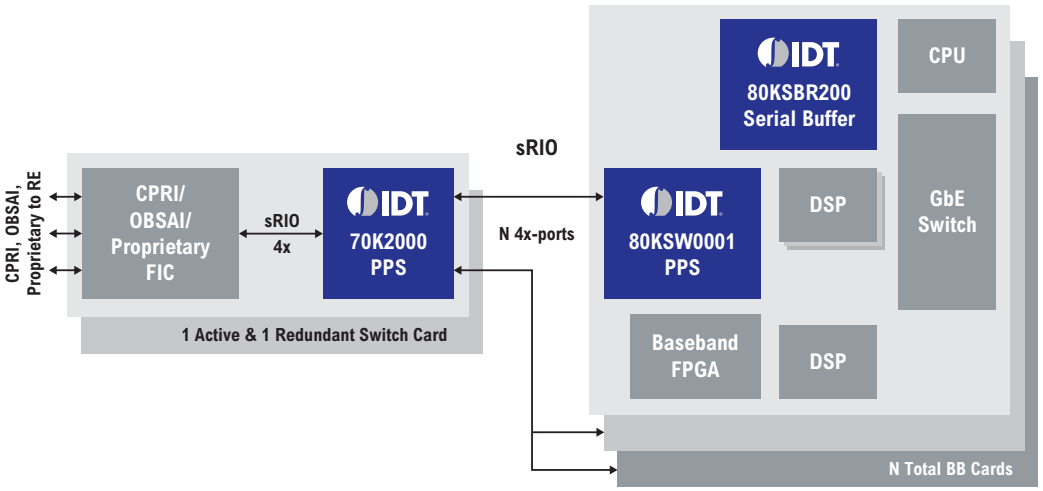

Figure 1 A generic baseband implementation with switch card and baseband cards, and sRIO used on the backplane.

Flagship DSPs with sRIO interconnects from multiple vendors are currently available. These DSPs utilize multiple high-performance direct memory access (DMA) engines to transfer data from internal memory to the sRIO port and vice-versa—often times maximizing the port throughput. Most devices are assigned multiple device IDs, so that they may be uniquely targeted by a "unicast ID" or be among several multicast recipients when multiple DSPs are configured with the same "multicast ID." Additionally, some DSPs offer "promiscuous modes" that will receive packets of any target ID. This flexibility is key in supporting control traffic targeting a particular DSP or supporting uplink data that is often multicast to a multiplicity of DSPs. In some systems with complex data path requirements, the promiscuous mode alleviates routing restrictions.

Field-programmable gate arrays (FPGAs) often offer baseband coprocessing in addition to the DSP farm. FPGAs allow not only highly parallel signal processing, but also are typically set up as look-aside devices with a single sRIO port. Occasionally, they employed in-line with the data path using two sRIO ports. They are not ordinarily used for switching due to the high cost of implementing multiple sRIO ports and switch fabric.

The FPGA offers a level of physical layer control to shape system traffic so that system performance may be optimized when implemented inline. This is critical in ensuring proper interpacket gap (IPG) timing is met at a receiving processor or preprocessor. Though sRIO supports the previously described backpressure mechanisms, in baseband, it is often helpful to have an FPGA device to further throttle traffic between endpoints such that the traffic spacing is consistent rather than bursty.

For example, the primary data transfer mechanism for a DSP is DMA, which tends to burst long lengths of packets at maximum possible speed. This "burstiness" can cause congestion at a receiving endpoint or switch, which can ultimately force a resend of traffic from the DSP. Providing consistent IPG can make the traffic behave better, and allow processing endpoints to avoid input buffer overflows and resultant packet retries. The IPG can be controlled with high timing resolution by providing idle clock cycles between packets to the FPGA's physical layer IP.

At least one host processor is implemented on the baseband card, which performs system operations, maintenance and may provide control information. To meet wireless infrastructure availability requirements, dual hosts are specified by sRIO with all appropriate arbitration functionality.

To meet frame delay requirements in the uplink, or even to act as a global storage memory, a large buffer supporting sRIO's sustained high throughput rate may be implemented. Figure 1 shows such a device on the baseband board. In support of multiple standards on a given platform, this optional buffer component may be made modular. Many OEMs are beginning to acknowledge the need for this separate buffer. System designers have come to realize implementations utilizing an endpoint's memory (for example, DSP memory) as a central storage space often leads to port congestion at the endpoint. This can ultimately impact the real value of the endpoint if congestion prevails. Offloading the memory requirement to a separate device alleviates this bottleneck. Careful system design consideration should be given the port bandwidth requirements at the endpoint in deciding whether it is appropriate to globally share endpoint memory.

Building onto baseband's sRIO foundation: enhancing the IQ data pipe with preprocessing

Switch enhancements

As shown in Figure 1, the switch is central to all endpoints. Thus, the switch stands at a topological "high ground." It seems intuitive then that any generic data transforms that must be applied on IQ data streams can be performed just one time in the switch. This is especially effective in the uplink where a low number of high-bandwidth streams must be reformatted according to baseband requirements and often multicast to multiple endpoints. A single switch board might multicast to multiple baseband boards, and similarly, those baseband board switches will likely multicast this data to multiple signal processors.

If the switch can perform this data formatting first, the receiving endpoints will not each have to perform those same transforms. Thus, precious DSP cycles and power may be saved by preprocessing the data in the switch, then multicasting this to the various endpoints.

Conversely, on the downlink, the switch serves as a central aggregation point on the baseband and the switch board(s). Architecturally, this is another key location to provide enhancements over standard sRIO switching with data manipulation capabilities.

Enhancing the data pipe with preprocessing capability

For our example implementation, let's consider packet payload processing functions in the uplink and downlink for a generic UMTS/WCDMA/TD-SCDMA system. We will assume there is a single IQ data stream supporting all potential Antenna Carriers (AxCs) in a given packet and any number of baseband boards. The available baseband boards will each support a unique subset of the potential AxCs, with possible overlap of AxCs among baseband boards. We will partition functions between switch and baseband boards, as shown in Figure 2. Upper and lower callouts highlight required preprocessing and frame delay functions on the uplink and downlink, respectively.

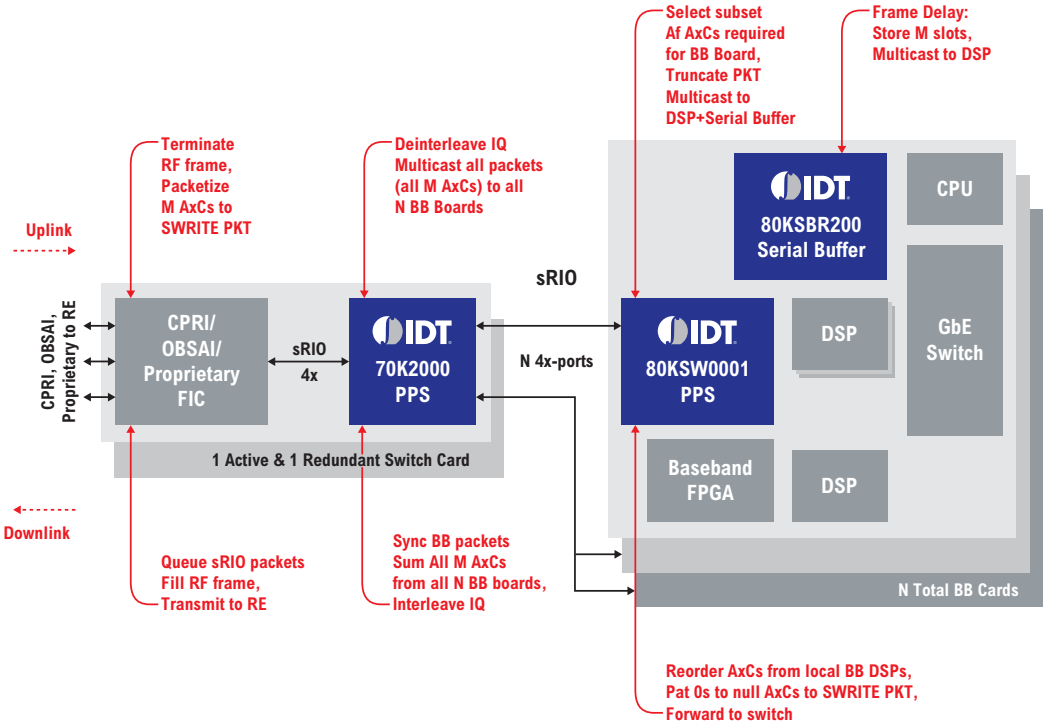

Figure 2 Our example with the WCDMA preprocessing requirements.

The example itself will highlight specific operations to be implemented. These functions will be a subset of all possible preprocessing capabilities. Figure 3 will provide actual system performance figures given this specific implementation. Where appropriate, other possible processing capabilities may be cited, but these will not be reflected in the final system performance figures.

Uplink: switch board

For our specific implementation on the uplink, IQ data formatting will occur on every packet and will include deinterleaving of I and Q sample data. The resultant IQ streams will then be broadcast (multicast) to all baseband boards. Native support should allow this to be applied to I-only or I and Q streams.

It should be noted that any control and timestamp information should be protected in a user-defined field within a portion of the packet payload so that it is not corrupted by data processing operations on the remainder of the packet payload.

Since the switch board is central to all baseband boards, it is a key location to provide global data formatting of the IQ streams. Alternative sample manipulation functions might include most significant bit/least significant bit reversal, IQ reordering, and/or sample resizing with or without sign extension and deletion.

An alternative implementation might allow the specific subset of AxCs supported by a given baseband board to be extracted from the complete IQ data stream. Thus, N unique preprocessing scenarios would be required in support of the N baseband boards. The resulting packet payload size will be reduced since it is a subset of the all AxCs from the originating payload. The system benefit is a reduction in bandwidth to the baseband cards. Ultimately, this can reduce the number or width of the sRIO ports to the baseband cards, reducing hardware cost and power consumption. For this example, we will instead take advantage of this AxC truncation on the baseband card (see next section).

Uplink: baseband board

As described as an optional implementation on the switch card, we will use the antenna carrier extraction capability to select just the subset of antenna carriers supported on the given baseband card. Since each baseband board will support its own unique subset of AxCs, the switch's pre-processor will be uniquely programmed on each baseband board to re-order just that baseband board's AxCs to the front of the payload, and then truncate the remainder of the packet (AxCs deletion). The resulting smaller packets will then be multicast to the DSPs at a fraction of the incoming data rate. Again, in doing so, the hardware cost and power consumption can be reduced.

Another system requirement is that the subset of carriers supported on a given baseband card may change dynamically over time. The preprocessor supports dynamic reprogramming of all processing requirements, including AxC selection/truncation, without dropping a single packet during the changeover.

sRIO natively supports the direct transfer of data into the target endpoint's memory. The packet header maintains the target address so that packets may be delivered directly to a given location in the target's memory. Preprocessing features enhance this by providing multiple output packet generation in tandem with target address generation. Thus, a given resultant packet may be partitioned into several smaller packets, each with their own unique start, stop and increment address values. This allows blocks of data within a packet's payload to be placed in several unique ranges of addresses within the DSP's memory. This enhanced DMA support provides a level of flexibility to the DSP software engineer.

Downlink: baseband board

Making the roundtrip back, we assume the baseband signal processors for a given baseband board will again be forwarding a subset of AxCs to the switch board. We will assume that the packet's antenna carrier arrangement may be out of order with respect to radio card requirements. We will also assume that AxCs will ultimately be summed for all baseband boards upon reaching the switch board.

The subset of AxCs supported on a given baseband card may be out of sequence, with some AxCs not supported at all. It is critical prior to summing at the switch card that AxCs are re-ordered in proper sequence, and null (unsupported) carriers are padded with sufficient zeros for the result of summing AxCs to be correct. Thus, the preprocessing requirements for a given baseband card are to re-order the AxCs within the packet payload and zero any null carriers. For example, the following sequence of AxCs from the DSP memory {AxC12, AxC15, AxC13} must become {0, 0, AxC12, AxC13, 0, AxC15, … , AxC1M}.

On the baseband card, control information may be embedded in the protected user-defined field within the payload as previously described. Again, the requirement to the preprocessor is to guarantee the user-defined field is protected during payload processing operations.

Downlink: switch board

At the switch board, packets from all N baseband boards are summed. AxC re-ordering and unused AxC zeroing ensures summation across antenna samples will be correct. For example, the data from baseband board one is

{0, 0, AxC12, AxC13, 0, AxC15, … , AxC1M},

and that from baseband board two is

(AxC20, 0, AxC22, 0, 0, AxC25, … , AxC2M),

then the sum result for the two will be

{AxC20, 0, (AxC12 + AxC22), AxC13, 0, (AxC15 + AxC25), … , (AxC1M + AxC2M)}.

The resultant sum may saturate, be truncated or be shifted bitwise in accordance with system requirements. Finally, IQ are interleaved, and the resultant packet is forwarded to the radio frequency card.

A unique challenge within packet-based systems is the need for synchronizing the packets prior to summation. There will inevitably be some finite skew among packets received from the separate baseband boards given the lack of explicit system synchronicity. Unavoidable skew may be incurred by standard sRIO physical layer effects, such as link synchronization or packet retries, synchronization skew among baseband boards and others. The preprocessor has separate buffer space for each input port and arrival windows to effectively synchronize (de-skew) packets from the baseband boards. Further, if a packet from a given baseband board is simply not received within the user-defined arrival period, the packet may optionally be replaced with zeros (or ones) and the summation will proceed without the packet to avoid stalling the data stream.

Lost packets such as this may be the result of several system issues. One possible cause might be related to bad signaling on the sRIO link or even a link that has failed and must be re-initialized. The preprocessor is able to identify and report a missing packet by sending a Port-Write maintenance packet with error information back to the host processor or offending DSP so that the situation can be corrected. If there are physical layer errors on the switch ports themselves, similar action may be taken.

3G+ baseband prototype system implementation

Off-the-shelf prototype hardware selection

For our example, we will utilize off-the-shelf components and hardware to create a prototype baseband system and review associated performance information. Several reasonable switches and associated evaluation boards can handle our switch board's pure sRIO port count "plumbing." To additionally meet the preprocessing requirements, we will use the EVC70K2000 Evaluation Carrier from IDT.

For the baseband board, a few suitable off-the-shelf advanced mezzanine card (AMC) cards can minimally meet prototyping requirements. Silicon Turnkey Express' (STx) AMC80KSW0001 board features four Texas Instruments' TMS320C6482 DSPs and the IDT80KSW0001 Preprocessing Switch device. This switch features the preprocessing requirements described for the baseband card.

The 10G serial buffer module provides UMTS frame delay on uplink

In our particular system, we elected to place a serial buffer local to the baseband board rather than the switch board. Since the preprocessor on each baseband board will truncate the packets, the bandwidth can be reduced and, conversely, greater incoming IQ data stream throughput may be supported.

Additionally, it is expected that, to optimize the system throughput, the switch board to baseband board links will run at about 60 percent of throughput just to support the real-time data stream. This allows a 40 percent margin for any eventualities of traffic congestion and subsequent packet retries. Had the serial buffer been placed on the switch board, this would effectively double the required switch board-baseboard link throughput to 120 percent (real time plus delayed traffic), which is not physically possible.

The serial buffer may store a full wireless UMTS frame or a variable Z number of slots within a frame, user-adjustable via configuration registers. As the delayed data is re-sent, the packets may have user-adjustable interpacket gap timing. The user may also enable the serial buffer to generate a doorbell interrupt to the receiving processor immediately after a user-defined number of packets (block of IQ data) have been transferred. This minimizes any delay in the processor between data receipt and the start of processing, and improves overall system performance.

The aforementioned AMC80KSW0001 baseband processing board offers an expansion port supporting STx's 80KSBR200 10G serial buffer module to meet the UMTS/CDMA frame delay buffer requirement.

Putting it all together: the final word is performance

Figure 3 reflects the component performance figures (power, latency and latency jitter) for our particular system example. The preprocessors were used exclusively on the uplink and downlink data pipe, and the data reflects the performance of the preprocessors for the data operations given in our example. The preprocessors are dedicated, provisioned paths, and there is no latency jitter in the preprocessing data path—the 35ns latency jitter is purely related to the standard sRIO physical layer effects. The preprocessor's best-in-class latency jitter saves overall system latency and thus the jitter budget to meet system synchronization requirements.

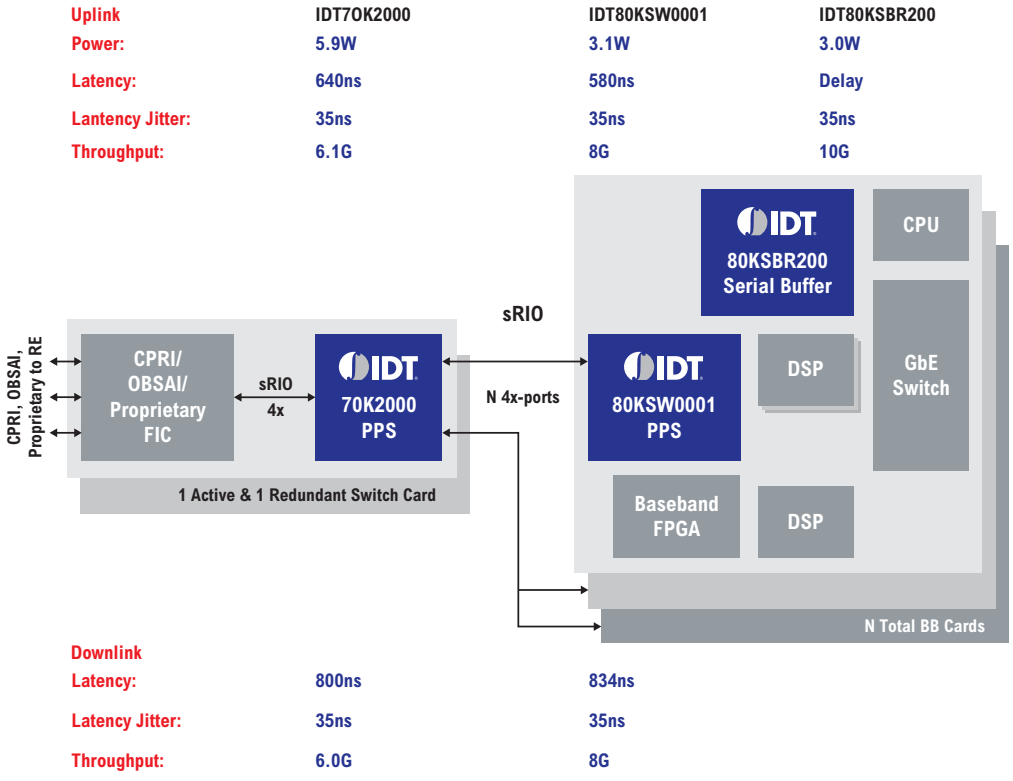

Figure 2 Our example with WCDMA preprocessing performance figures for the aforementioned processing requirements.

Though the data represents the uplink and downlink data paths to the baseband board, it should be expected that, during the primary baseband processing, depending on the algorithms required and associated partitioning, multiple hops among DSPs and the FPGA through the switch block itself will be required. It is critical then that switching latency is kept to a minimum because the total jitter will increase by multiples of the number of hops required. Typical latency for pure switching is 160ns and associated jitter is 35ns (assuming no blocking at the output port by other packets and priorities).

Device power is provided at top of Figure 3, while latency, latency jitter and throughput are provided for uplink (top) and downlink (bottom). The latency in the serial buffer is the user-defined frame/slot delay timing. The throughput data in Figure 3 does not reflect aggregate throughput for the device, but the maximum sustained throughput for the preprocessors for uplink and downlink processing requirements.

The preprocessors offload the DSP so that they may be used toward their primary algorithmic functions rather than data formatting requirements. This saves available cycles in the DSP. Consider, for example, that each preprocessor operation (de-interleaving) requires at least one processor cycle per sample (some operations require two or more). Consider that the DSP must perform this operation on each sample within the payload, real time on the data stream. For a typical 1GHz processor, depending on wireless system parameters, 20 percent of available processor cycles or more may be consumed by performing even a single function on all IQ samples.

Error handling

The completed system needs to be robust enough to detect, report and even correct systemic failures. Thus, error-handling capability cannot be taken lightly in a baseband infrastructure, which has stringent reliability and availability requirements, complicated by the distance of remotely deployed equipment. SRIO provides a robust set of error-reporting capabilities through the error management extension specification and software assist error recovery registers. Thus, many error conditions may be detected and even reported for those devices in which these are implemented. However, both of these are optional implementations. For example, if a field replaceable unit has a bad connection with the backplane, a classic symptom is loss of lane sync or bad characters seen at the receiver. Robust error handling support would identify this connectivity issue, log the error and report this back to a host processor where corrective action can be taken.

It is particularly critical for switches to implement robust error-handling capabilities since they connect with every endpoint in a system. System designers should expect current generation devices to offer sRIO-based features, and even enhancements beyond these. Hardware may be enhanced to detect—with resolution beyond the sRIO specifications—the types of errors seen not only on the sRIO ports themselves, but also on any other functional blocks. This includes proprietary functional blocks on the devices. They must not only provide error detection, but also hardware-based error responses. These responses should minimally include the ability to report errors back to a host processor. Ultimately, the action taken is best served by software interrupt service routines, allowing the most flexibility to the error management and controls software developer.

Maintaining system bit error rate

The sRIO physical layer specifications provide explicit requirements to maintain 10E-12 bit error rate (BER) or better. Component providers meet the requirements by providing transmitters with drive strength and pre-emphasis controls to meet sRIO-specified minimum transmission and receive requirements. Many vendors offer transmission capabilities beyond the minimums required by the sRIO specifications.

Enhancements to the standard include proprietary features that support measurement of system BER. This is especially useful in fine tuning transmission characteristics to optimize BER on prototype boards prior to production. Features may also support the "run time" measurement of BER of production boards so that the BER outside of system requirements may be reported, and transmission characteristics (drive strength and pre-emphasis) may be adjusted to resolve the issue.

Though sRIO does not currently specify these features, there are industry-standard SerDes built-in self-test capabilities, such as Pseudo Random Bit Sequence (PRBS) generators and corresponding error checkers, and loopback capabilities. Further enhancements include stressed eye capabilities to allow added confidence by testing at tighter margins. The system designer should take into account the enhanced proprietary feature sets of these devices to ensure BER may be tested, monitored and adjusted—especially between devices from different vendors, which may have unique implementations.

Hot swap support

The sRIO standard provides facilities to support hot-swapping of boards. This complements the ATCA management functions built into boards, such as the AMC70K2000 baseband prototype board. The aforementioned optional software assist error recovery registers allow resynchronizing transaction AckIDs in link partners when the link has been disrupted and a mismatch may exist between AckID values between ports on separate boards. This is particularly useful to support field replaceable switch and/or baseband boards as described in our example system.

Per sRIO specification, reset control symbols may be issued by a host to reset the entire device, and thus, the ports themselves. This will undoubtedly reset any AckIDs but may not be appropriate for a run-time system—since this will reset the device back to power-up values and reset route tables.

For run-time systems, configuration and routing information must stay intact or traffic will fail to be routed through the system. A non-sRIO standard override to the full device reset that will only reset the port that receives the reset control symbol is implemented to support this process. Thus, even if the link partner's have their AckID's out of sync and packets can no longer be transmitted on that link, one of the partners may still issue a reset control symbol at the physical layer to reset just that port—and not the entire device.

Configurable ports to support modular hardware

sRIO specification defines a standard 1x or 4x port. The 4x port can be downgraded to a 1x port on lane 0 or lane 2 for redundancy. This 1x/4x port mapping has been adopted for the subsidiary sRIO fabric specifications for ATCA and microTCA.

Although most baseband systems have explicit bandwidth and, therefore, port rate and width requirements, some are extensible to allow the same hardware to bridge femto, micro, pico or macro system borders.

To facilitate this, look for sRIO components, especially switches, to allow enhanced levels of configurability beyond the sRIO specification. Enhanced ports may be software reconfigured to support multiple 1x ports from a given 4x port. This optimizes pin count utilization and ultimately saves costs for the hardware designer.

Glossary

AckID	Transaction ID for link acknowledgements
AMC	Advanced mezzanine card
ATCA	Advanced Telecom Computing Architecture
AxC	All potential Antenna Carriers
BER	Bit error rate
CPRI	Common Public Radio Interface
DMA	Direct memory access
DSP	Digital signal processing
FIC	Fabric interface card
FPGA	Field-programmable gate array
Ghz	Gigahertz
IPG	Interpacket gap
IQ	In-Phase and Quadrature
ns	Nanosecond
OBSAI	Open Base Station Architecture Initiative
OEM	Original equipment manufacturer
PBRS	Pseudo Random Bit Sequence
SerDes	Serializer/deserializer
sRIO	Serial RapidIO
STx	Silicon Turnkey Express
TD-SCDMA	Time Division Synchronous Code Division Multiple Access
UMTS	Universal message transfer system
WCDMA	Wideband Code Division Multiple Access

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01 Jan 2024)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.