RapidIO 10 Year Success: Stepping in when Moore's Law Can't Keep Up

By Devashish Paul,
Senior Product Manager, Integrated Device Technology

Back in 1969, Gordon Moore declared that semiconductor technology would double the density of transistors that could be put on a die every 18 months. To some extent, this set the expectations for an entire industry for the next 40 years. The problem with Moore's Law is that the number of transistors that can ever be put into a single piece of silicon exceeds the needs of the application, simply because the end user's expectations always exceed what the silicon provide today.

In a sense the needs of the applications are driven by what is currently available today. As an example, our expectations for what video conferencing should look like has advanced substantially from what was available 5, 10, 15 years ago, but at each step in time the user expects slightly more than what is "currently available". What is currently available is driven by how system designers can best partition an application over multiple discrete processing elements and build software on that platform.

In the 90's the amount of processing capacity available in leading processors made the deployment of real time synthetic aperture radar not very useable. Anything working in an airborne environment in real time was flying with low resolution, at low ground speed, with small frame sizes, with really low frame update rates, and consumed the limit of rack space, power consumption and cooling capacity that was available. Inherently, these systems needed multiple processing cards to conduct range and cross range processing and graphics display, creating images that were useable in real time by the operator.

In networking applications, line and air interfaces needed to support more users, with more complex services per card, with low power. Data needed to pass between multiple processing elements on a card for the implementation of a variety of functions both on the line/air interface side and on the network interface side.

Approximately 10 years ago, a number of experts from the embedded systems world met to come up with a better way for microprocessors, FPGAs, digital signal processors, ASICs, entire boards and entire chassis to speak to one another with any networking topology, with low latency, low power and an architecture than would simplify the design of application level software. For the reasons mentioned, above, it was clear back then (and continues to be clear today) that applications would very rarely be built in embedded systems with single processors only. Moore's law cannot possibly ever catch up with application needs. Or as the joke goes….what hardware gives, software takes. The moment software engineers get more computing resources; they find a way to deliver a better application level user experience.

Winding back 10 years, Ethernet offered a means of delivering peer to peer processor networks, be it chip to chip, board to board or between chassis. However, Ethernet evolves from the LAN and WAN networks and architects were trying to find an efficient way of using it in embedded system. Because of its background in LANs and WANs there is effectively an assumption that there will be a processor at each node to terminate the protocol stack. This is a reasonable assumption for LAN and WAN networks, but introduces too much latency and power consumption in real time embedded systems. There had to be a better way.

RapidIO 10 Year Success: Stepping in when Moore's Law Can't Keep Up

By Devashish Paul,
Senior Product Manager, Integrated Device Technology

PCI and the upcoming PCIe standards offered an alternative; however, they were really designed for monolithic single host processor systems, with the concept of a root complex. Scaling to multiple processors on line cards, over backplanes, with multiple hosts was going to be complex. With PCIe and non transparent bridging, the problem can be managed for a small number of endpoints, but the memory mapping becomes difficult very quickly

Because RapidIO was built from the ground up for multi processor peer to peer networks, it inherently comes with the following attributes

- Reliable transmission
- Sub micro second end to end packet delivery
- 100 ns cut through latency
- No processor overhead to terminate the protocol
- High performance messaging for transmitting large amounts of data
- Push architecture with the option of every processor in the system having its own memory subsystem

Early adopters in wireless and military markets realized that these attributes were exactly what some of the incumbent protocols (Ethernet and PCI) were lacking and started R&D or technology programs to RapidIO. These transitioned to some of the first production systems in a few years. As competitors noticed the performance capabilities that were being delivered into the market with RapidIO based systems, they realized that the performance "train" was leaving the station. It was time to either architect systems around RapidIO or be left behind. Soon standards bodies like VITA and PICMG drove RapidIO into their standards. We started seeing VITA 41 and 46 as well as ATCA and MicroTCA systems with RapidIO 1.3. Companies like Ericsson, Huawei, Mercury, Curtiss Wright and GE started talking publicly about their RapidIO based systems.

Today, over 150 distinct customers worldwide are using RapidIO. RapidIO Gen2 is being deployed in the market today with 4G Wireless, Open VPX military systems, 20 Gbaud ATCA and MicroTCA systems being built.

The vision 10 years ago by a small group of experts, seeing a gap in the "servicing of embedded interconnect needs" has brought us to the point where every 4G wireless system being built today is connected with RapidIO. It's not a surprise, because like any business, when you design something to service customer needs and add end customer value, the adoption happens.

In the next editions we'll delve into specific applications and show specifically how RapidIO has succeeded in 10 years in transforming how certain applications are architected and deployed in the field.