

White Paper: Ethernet in the Embedded Space

A primer on Ethernet concepts from determinism to TCP offload engines for high-speed data processing



NETWORKING FROM THE START

Networking originated from the need to share information. Many of us accomplish such a thing on a daily basis through conversation. For example, your typical office framework: you work side-by-side with your colleagues but also have a manager who will check on the work being produced periodically. You have both peer-to-peer and supervisory communication taking place.

When it comes to Ethernet, different kinds of equipment are needed, yet the goal of communication stays the same. Keep it simple and, especially in terms of hardware and software, keep it inexpensive. One of the major factors that affect those goals is timeliness – responses should be received in a reasonable period of time after inquiry. Keeping the process predictable creates a deterministic system. Some early arrangements of deterministic networks took the form of the token ring and the token bus.

TOKEN RING

The token ring was established quite some time ago in an office environment to allow a complex grid of many terminals to each have some allotted time to get work done. The media access unit (MAU) diagram (fig.1) visually shows the procession of information. Consider that there is one token, and this token allows one terminal node to broadcast and receive. The token must be passed between the nodes, giving each its turn. Typically, there would be a token rotation and a token hold time. The rotation is the given order of terminal nodes to which the token is passed, while hold time is how long each node gets to do its requested job.

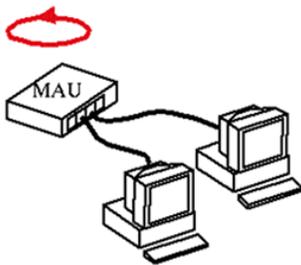


Fig.1

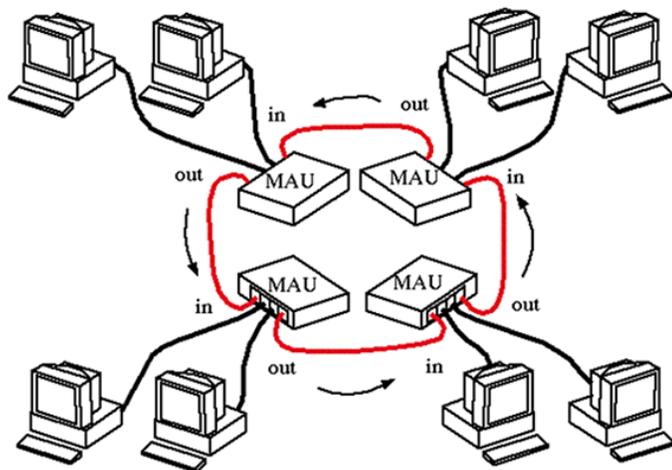


Fig.2

In a more complex office environment (fig.2), where you have several of these MAUs passing the token around their ring, there may very well be several terminal nodes connected to any one MAU. The terminal nodes will need to share its time with the MAU.

TOKEN BUS

The token bus is very similar to the token ring, in that only one terminal node has the token at any point in time and every node gets the token at a predetermined time. The rotation order and hold time are usually preconfigured when the token bus is set up by the network manager. In the example (fig.3), the rotation is continuously counterclockwise but that may not always be the case. Terminal D may pass to C, back to D, and then continue on to E; any order is possible.

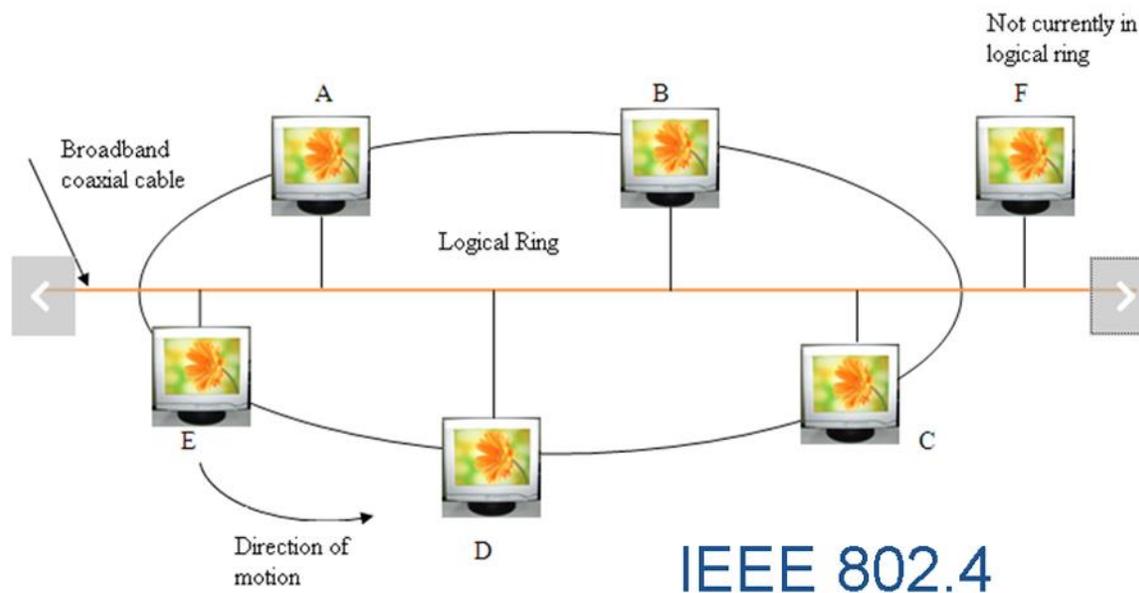


Fig.3

The network manager that sets these arrangements up is connected to the same token bus. It's primary function is to set each node's token rotation and time sequence during network initialization, and to continuously monitor network traffic as a diagnosis tool. The IEEE 802.4 standard, also known as Manufacturing Automation Protocol (MAP), was a popular type of communications networking standard installed in many factories where deterministic traffic could be predicted and placed on a network.

EMBEDDED APPLICATIONS

Networking in the embedded space is used:

1. To replace legacy serial communication
2. To connect subsystems (peer-to-peer)
3. To connect the subordinates to the supervisory
4. To deliver captured information to storage
5. To enable timely interrogation of stored information
6. To create seamless information boundaries between systems

THE RISE OF ETHERNET

Ethernet arrived on the scene in 1980 and became fully standardized in 1985. It quickly became popular as a wonderful, low-cost standard. Why was it so inexpensive? It was used heavily with many terminals in the office environment, and the sheer number of these terminals drove the price down. It was based on a multi-drop technology; running one long cable and allowing nodes to be appended fairly easily. Using the non-deterministic Carrier Sense Multi-Access/Collision Detect (CSMA/CD) protocol, performance varied between “well-behaved nodes” and “bandwidth hogs”. Well-behaved nodes knew enough to broadcast on the cable and then detach to allow others a chance to transmit. Bandwidth hogs would get a hold of a cable and stay on, preventing other broadcasts and reception. CSMA/CD protocol is the reason these two possibilities exist.

Carrier sense multi-access means multiple nodes are allowed on a cable, all can access it, and they can all listen to find out if the cable is clear for transmission. When a node is ready to Transmit or Broadcast, it must first listen or detect whether the network cable is quiet, that is, no other node is using the network cable for transmission. If the network cable is quiet, it would then attempt transmission while at the same time listening to the cable network. If the transmitting node hears the echo of its message, it then knows that its transmission will be successful and it has control of the network. If two or more nodes start broadcasting at the same time, the echo would contain the combined transmission of the two or more nodes attempting transmission; this is a collision of transmissions. Each respective node then, not hearing its own transmission, would back-off from further transmitting for some wait period and start all over again...listening...and attempting to transmit...and listening for its own transmission. This is why the Ethernet CSMA/CD protocol is not deterministic. When boundary conditions on the data packet size are enforced, it is possible to use Ethernet in some instances for control applications since nodes are forced to transmit less information but more frequently.

Media typically used for Ethernet is fiber or RJ45 Copper (CATx) cable. Fiber can go for longer distances and is noise-resistant but also very costly. On the other hand, copper wire is reasonably priced but may have a slight bit more electromagnetic interference. Overall, copper is very popular and is used in most home and embedded systems.

802.3 COMPLIANCE EVOLUTION

Ethernet evolved over time, using different cables at varying lengths and node counts, starting at 10Base5 (fig.4). When 10GBase-T was developed, it used a full duplex point-to-point mode of transmission between only two nodes. This mode is very high speed, with no interference or determinism issues. Likewise, with 40GBase-T, transmission is also full duplex point-to-point but with the distances starting to shorten a bit. Our focus will be on the 10Gb Ethernet.

	Cables	Speed	Max. Length	Max. Nodes
10Base5	Thick coax	10Mbps	500 m	100 @2.5 meter intervals
10Base2	Thin coax	10Mbps	5-4-3 Rule: 5 segments of 185 m 4 repeaters Only 3 segments could have nodes, with a maximum of 30 stations per segment	
10Base-T	CAT3/4/5	10Mbps	100 m	1024 nodes
10Base-F	Fiber	10Mbps	2 km	
100Base-T	CATx	100Mbps	100 m	
1000Base-T	CAT5	1Gbps	100 m	
10GBase-T	CAT6 Class F/Category 7	10Gbps	100 m	
40GBase-T	CAT6A/7		50 m	
100GBase-T				

Fig.4

SYSTEM INTEGRATION AND STANDARDIZATION

When looking at system integration goals, there are a variety of issues one can face, but the most important one is standardization. The Open Systems Interconnect (OSI) standard was designed so that multiple parties could participate, communicate, and share information by implementing a specific combination of hardware and software. The hardware is the physical connection to the medium, while the software has to execute and manage the software packet exchange. The ultimate objective is reliable connectivity to get the job done. The availability of the network, or performance, varies along with its speed. The CSMA/CD protocol proved that there are some performance issues. The question that needs asking is, “is the performance sufficient to get my job done?” There are pros and cons to every structure.

STANDARDIZED OPEN SYSTEMS INTERCONNECT

OSI MODEL	
7	 Application Layer Type of communication: E-mail, file transfer, client/server.
6	 Presentation Layer Encryption, data conversion: ASCII to EBCDIC, BCD to binary, etc.
5	 Session Layer Starts, stops session. Maintains order.
4	 Transport Layer Ensures delivery of entire file or message.
3	 Network Layer Routes data to different LANs and WANs based on network address.
2	 Data Link (MAC) Layer Transmits packets from node to node based on station address.
1	 Physical Layer Electrical signals and cabling.

The Open Systems Interconnect (OSI) TCP/IP stack (fig.5) is made up of 7 layers. The lowest layer is the Physical Layer of fiber or copper, possibly wireless nowadays. This interface is the means by which a node would communicate on the medium. The next layer is the Data Link (MAC) Layer. This is where information pertaining to the station address is used to link information to pass from one node to another. The third layer is the Network Layer which works with multiple bridges and multiple cell networks. After that, the Transport Layer ensures that information is sent and delivered between a station address on one network to a station address on another network. Next, the Session Layer separates the environment for each particular application or user. Following is the Presentation Layer which ensures that the information coming from the Session Layer is put into the proper format for the Application Layer to use. Lastly, the Application Layer is where the work is done; whether you are sending emails, controlling machinery, collecting information, etc.

Fig.5

RESOURCE REQUIREMENTS

Resource requirements on the OSI model depend on the goal each layer is trying to achieve. Layers 1 and 2 don't require nearly as much as 3, 4, 5, or 6. A tremendous amount of logic needs to be executed in the upper layers, and that can chew up a lot of CPU time and memory depending on system architecture and bus speeds. The faster the computer and related data buses, the more seamless the information transfer will be; moving data is where you will see the majority of your resources being used.

IMPLEMENTATION SPECIFICS

There are many implementation scenarios combining the hardware interface and intensive OSI TCP/IP software protocol. The following are the four most popular options and how they may be employed.

CPU EXECUTION OF TCP/IP STACK

A logical first step is using the CPU to drive the OSI stack. Today, computers come in single and multi-core architectures. In a single core computer, the logic to execute the TCP/IP stack may be sitting in an executable part of that computer. Merely sitting there will occupy some memory resources, and actually consumes a lot of CPU resources to drive this stack. The faster the stack needs to execute, the less time is available to execute the application in that computer. When a multi-core CPU is used, one of the simplest things to do is to move that OSI TCP/IP stack software to a core all by itself. In this way, the dedicated core will execute the communications protocol on its own, leaving the additional cores to do any other work that needs to be done at that node.

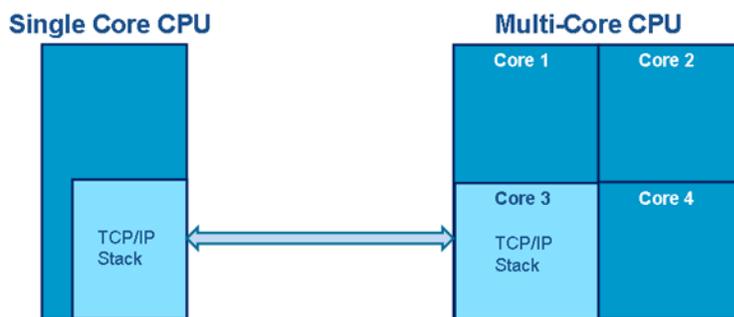


Fig.6

MORE THROUGHPUT AND THE ENHANCED PERFORMANCE ARCHITECTURE

Increasing the performance of your existing Computer System architecture is another popular option. Increasing CPU memory is always a good start. Another popular option is to use the OSI enhanced performance architecture. If you look at the OSI architecture (fig.7), you have the Physical Layer to which no changes can be made. Next there is the Data Link layer which must be present in some form to resolve station-to-station addressing. However, by streamlining the communication between the MAC Layer and the Application Layer, basically consolidating the logic normally executing at the Presentation, Session, Transport, and Networking Layers, a tremendous reduction of code execution can be achieved. This reduces the size of the stack logic, the load on the CPU, and the associated memory requirements; this means faster execution. The cost of this type of setup is sacrificing some protocols and other implementations that might be supported in the layers 3-6.

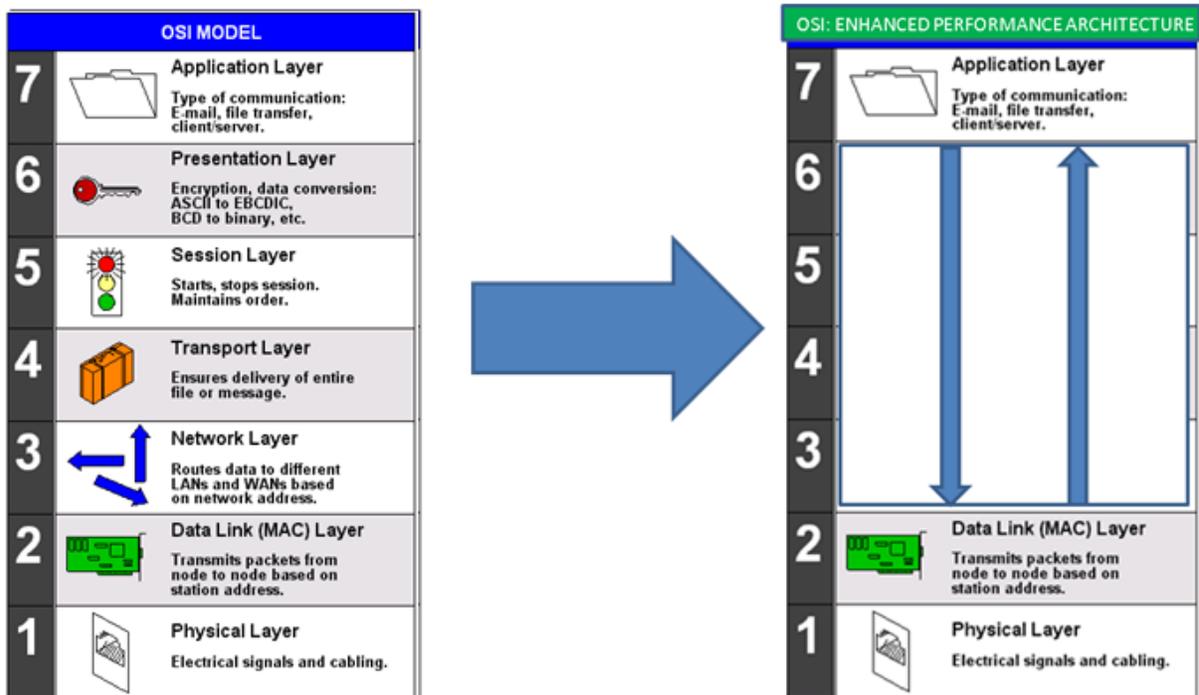


Fig.7

THE SILICON STACK

Use of a silicon stack to offload work is another way of increasing performance. A silicon stack is basically an auxiliary CPU; its sole purpose is to process communications. Silicon stacks provide additional capabilities such as: IPV4/IPV6, iWARP RDMA, iSCSI, FCoE, TCP DDP, and full TCP Offload. The elegance of the Silicon Stack is that the entire OSI TCP/IP Stack Plus More can be implemented without impacting Application Logic performance.

SILICON STACK PERFORMANCE

Below is a performance comparison between the silicon stack and the software stack (fig.8). Examining Throughput per port, it can be seen that the Software Stack is limited to roughly 40MB per second whereas the Silicon Stack can sustain 250MBps on 1GbE and 2500MBps on 10GbE. The Host CPU overhead for the silicon stack implementation is extremely low, as the silicon stack in essence is a parallel engine to the CPU; the Host CPU overhead for the software stack, on the other hand, is comparatively high since the Software Stack competes for CPU Resources at an increasing level as the communication speed increases. Latency is the time that it takes for the transmission to start after you've preconfigured all of the parameters for the transmission; here too it is obvious that the silicon stack exceeds in performance over the software stack as CPU resources are used only minimally for the silicon stack implementation. Determinism is the variation on the latency for sending and receiving transmission packets; again the silicon stack wins due to its limited CPU resource impact. As for reliability under load, the silicon stack experiences no noticeable change in performance while the software stack will be impacted as resources are shared with any executing applications.

	Silicon Stack		Software Stack	
	1GbE	10GbE	1GbE	10GbE
Throughput per port sustained Mbyte/s	250	2500	40 host CPU limited	
Host Overhead	Very low		Very high	
Latency	15 μ sec	10 μ sec	250 μ sec	
Determinism (Typical)	\pm 2 μ sec		\pm 200 μ sec	
Reliability Under Load	Excellent (any load conditions)		Poor (under heavy load)	

Fig.8

HARDWARE: TCP OFF-LOAD ENGINE

How is a TCP off-load engine (TOE) integrated into an embedded system? Many single board computers (SBC) today have XMC sites that can be used to plug in an XMC form factored TCP off-load engine which can potentially support up to four 10Gb Ethernet ports. The SBC will most likely have one or more 1Gb Ethernet ports as well, and those will be driven by the software stack executing on the SBC itself. In the context of the overall system, the TCP Off-Load Engine can provide up to four extremely high speed ports executing in parallel with the 1Gb Ethernet port(s) on the SBC itself.

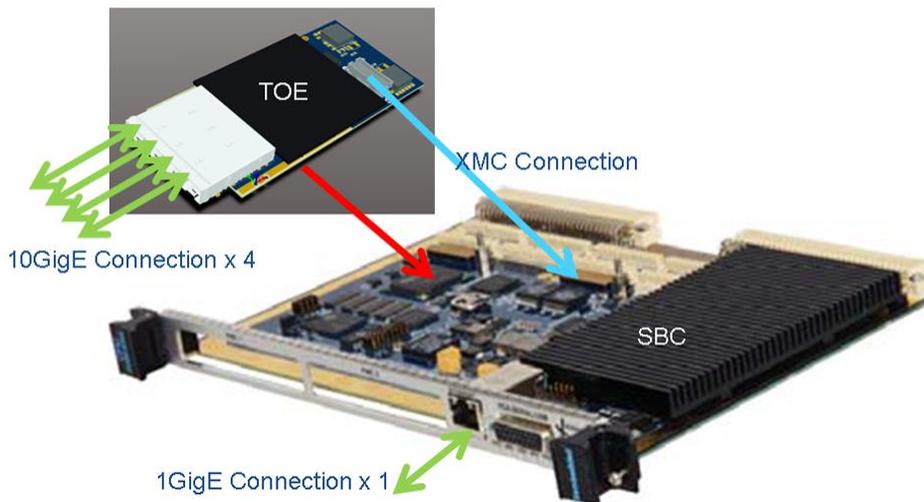


Fig.9

If we look at this in block diagram form, you have your single board computer (SBC) and you have your bus on the SBC. You have the ability of the SBC to process packet information to do all kinds of wonderful functions for you. Then you have this auxiliary piece of hardware which is called the TCP off-load Engine (TOE). It supports up to four 10Gb Ethernet ports connected by an XMC connector back to the SBC. You can use this TOE card as a switch if you choose, and information can be routed from one network to another. It could be routed from one 10Gb port through the SBC for some packet work and modification there and then shipped off across the bus. It could possibly be used with information coming in for processing and then going out on the 1Gb Ethernet port. You have a whole plethora of options available. The capability you have by adding a TOE is being able to access 10Gb Ethernet ports on an SBC and not impact the processing power of the SBC.

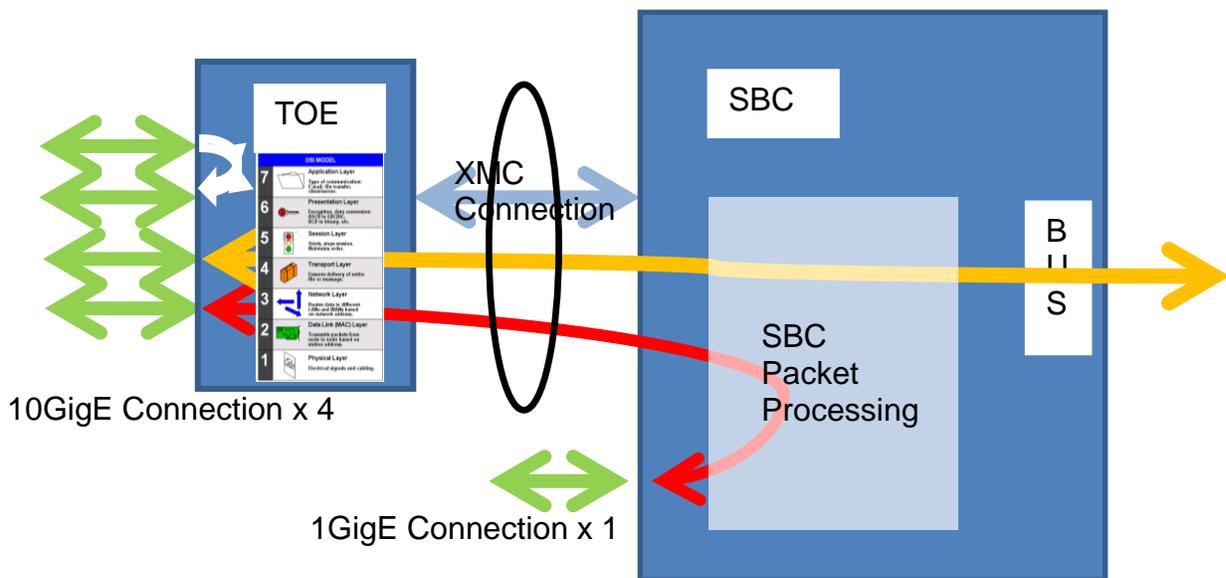


Fig. 10

SPECIAL FPGA PACKET PROCESSING

Your process may require special information packeting or manipulating special packets. Using an FPGA in one of the XMC sites, this work could be off-loaded so that not all of the processing was being done on the SBC.

For instance, images are being captured and coming in on gigabit Ethernet. You may have to take two images and overlay them. This overlay could be done right within the FPGA, then sent back to the CPU if there is any additional processing, and then back across the bus to some other location. The FPGA modules that Acromag recently released have a Virtex-6 FPGA and two 1Gb Ethernet ports (either fiber or copper), giving you more of an opportunity for off-load processing.

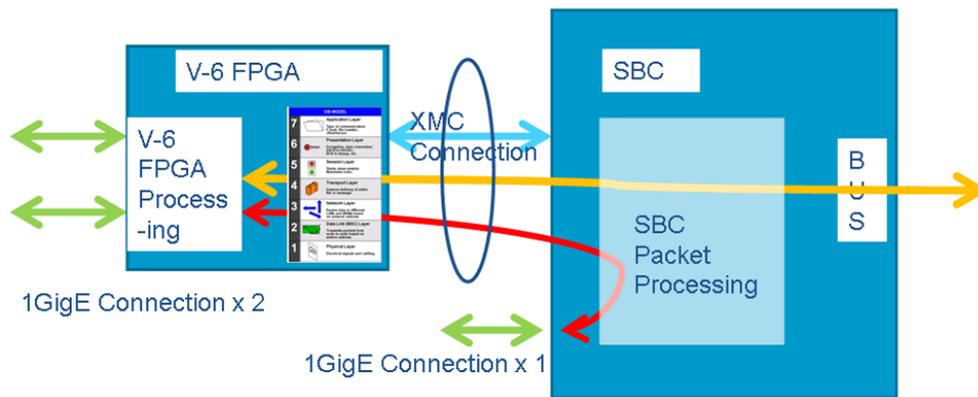


Fig. 11

HEIGHTENED EXPECTATIONS FOR TECHNOLOGY TODAY

The biggest trend today is sensors. There are more options and better quality sensors for use in almost every embedded application. They are being designed so that you get better situational awareness. To obtain a better understand what is going on in the world, it is necessary to actually gather and analyze more data. More data will spill over into increased storage requirements and all of the systems architecture issues that go along with it; whether you are processing that information or moving it from place to place.

No one wants to hear excuses that the data cannot be processed. Expectations have been amplified in terms of better results, sophisticated algorithms, and faster CPUs. Today, Acromag is helping by designing FPGAs for use in this field along with GPUs to help with increasing amounts of data being collected.

The data transport infrastructure in any system is most important. It needs to assist with data collection, shifting that data to analytical engines in a timely fashion, and getting the data to storage devices. This is often where you find many of the Gigabit and 10Gb Ethernet networking solutions making that possible. Even so, networking expectations are very high when discussing time limits for the delivery of information and the deterministic nature when it comes to the delivery of that information. Technologies are continuing to move forward and engineers concentrate on using the latest technologies. Communications Technology needs to meet the requirements of today, as well making it possible to keep ahead of new specification requirements as the future may dictate.

ABOUT ACROMAG

Acromag has designed and manufactured measurement and control products for more than 50 years. They are an AS9100 and ISO 9001-certified international corporation with a world headquarters near Detroit, Michigan and a global network of sales representatives and distributors. Acromag offers a complete line of [embedded computing](#) and [embedded I/O](#) products including bus boards, mezzanine modules, wiring accessories, and software. Industries served include military, aerospace, manufacturing, transportation, utilities, and scientific research laboratories.

For more information about Acromag products, call the Inside Sales Department at (248) 295-0310, FAX (248) 624-9234. E-mail solutions@acromag.com or write Acromag at 30765 South Wixom Road, P.O. Box 437, Wixom, MI 48393-7037 USA. The web site is www.acromag.com.