IO and multiprotocol processing in highly demanding embedded architectures

IO and network management are faced with increasing speed requirements

Low physical I/O protocols or device management have always been handled by a hardware device, simply because line survey or reaction to a bus change need very short reaction time. It would require a huge amount of processing power in order to be fast enough to comply with the bus management physical and timing requirements.

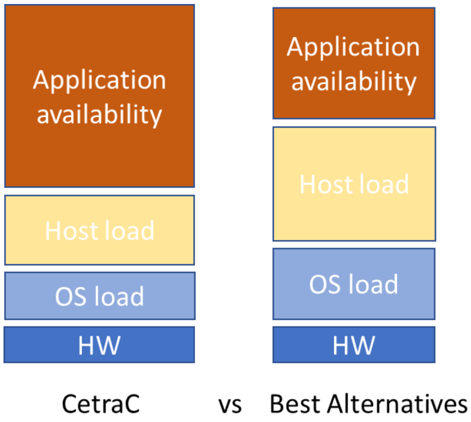
But once the physical layer has been handled by these components, it can still be up to a processor to undertake data handling and low-level protocol management.

Dedicated process and interruption management are the common answer in order to accomplish these tasks.

Hence high bandwidth protocols and large amount of data generate a large stress on processor load. Real time activities needed for this purpose generate a large amount of context switching, at the expense of calculation and data processing activities which should be the main objectives of the processing power.

If you add the data values surveillance and availability management and/or if you consider encryption/ decryption, the overhead of I/O management is hardly scalable.

In the case of Ethernet TCP/IP this difficulty has already been observed and off the shelf Ethernet boards can now be in charge of a larger amount of the protocol handling lowering the TCP/IP stack processing needs.



For instance, CRC, fragmentation, VLAN management can be, by configuration, put in charge of the Ethernet adapter TSO (TCP Segmentation Offload) for TCP or GSO (Generic Segmentation Offload)

Complex system handling

The more complex and more time dependent the protocol is, the more stress will be put on the CPUs, based on interruption handling - even with RT OS.

For example, ARINC664P7 and even more TSN expect a very precise timing for sending frames and for validity checking. Having a hardware layer in charge of these tasks not only reduces the processor(s) needs for fast interaction but also guarantees the precise timing and a low jitter value.

Synchronization needs are therefore essential for these protocols and IEEE 1588 or TSN 802.1AS have been designed to address this issue. As very fast and accurate calculation have to be achieved upon the arrival of the dedicated frames, having a direct hardware support for these functions is the guarantee of precise timing even on a heavy load situation.

In addition, synchronizing the applications is the key to the whole system performance. Data consumers and providers need a precise scheduling scheme to be able to have access to any data in the system, have their cycle of functional calculation, and be able to produce the output data in a timing that will allow the respect of the whole system cycle.

Having a free run timing for the applications is the best way to add cycles delays between the different systems that need to exchange data especially if these processes run on remote processors or microcontrollers.

Thus, a fast and reliable access to precise timing shared among the system is key and the IO processor hardware synchronization functions is the best possible common time source.

Even in a multiprocessor environment this hardware IO layer will be a useful part of the coordination, scheduling, and data segregation. The processor can be fully dedicated to its main purpose, e.g. analyze, calculate, and give results. If the OS or the supervisor stays in charge of practical processing segregation and scheduling, it will be much more efficient with the help of a hardware accelerator.

This virtualization of the data exchanges opens the door to a full Service Oriented Architecture (SOA) where processes are totally independent of the underlying network, protocols, and data management.

Like in modern avionic architectures, all complex embedded systems will hugely benefit from SOA techniques. Independent functional development tends to become a rule and having these codes totally unlinked to the deployment platform gives actual reuse possibilities and more a flexible system organization. Processor affectation for the processes as well as physical localization and protocols can be modified while keeping all these functional codes unchanged.

The multi-protocol hardware management will not only allow the integration of legacy devices and protocol in the evolution of the architectures but it will also make these evolutions easier by leaving a large part of the deployment unchanged when these legacy devices are replaced by a new device with a different communication path or even by an equivalent process in a pool of existing processes.

This hardware accelerator can also be in charge of data aggregation. Multiple sensor data coming from different protocols, or local calculation results coming from edge calculators can be joined by configuration in a unique data packet principally for the cyclic data exchanges.

This lowers the amount of small packet exchanges with the benefit of reducing the communication overhead, reducing the bandwidth usage, and freeing the processor of the handling of a high volume of interruptions generated by the exchanges of small amount of data. It can easily generate a reduction by a factor of 10 the interruption management load by a processing unit.

Safety and security

If network load and processing management are a huge concern in today’s architecture studies and deployment, safety and security are the new horizon.

With the increasing complexity of functions that are supported by the electronic architecture, failure is less and less an option and handling and maintaining a functional fall back mode will soon be the rule.

The use of the hardware IO and network management will natively give safety functionalities with

- The guarantee of data path segregation,
- Redundancy management integrated functions that allow the automatic duplication of sensitive data over disjoined paths and protocols communication link and still providing the data target a unique piece of information for action or process. This will help defining a data independent set of functions on the processors without having the application or the OS to care about these essential safety functions
- Data consistency (CRC calculation for instance) and timing coherency are also supported by the hardware with no concern of processing power and timing availability.

- A complete communication surveillance data path can be provided by the hardware layer. All needed abnormal incoming data / timing / physical protocol / bottleneck event can generate an error message linked to a dedicated function anywhere in the network in charge of the behavior decision in the current running context.

Regarding the cybersecurity challenge that is in front of us, it’s time to integrate support for it deep inside the architectures. The addition of a verification process somewhere in the system will still be needed but the help of embedded cybersecure communication layer will not be avoided.

Hardware communication companions are able to guarantee the segregation of the different data even on a unique communication link as mentioned before. It is a key of the insurance that no unauthorized access can have a way to spy or modify the content of an adjacent flow.

The hardware nature of these functions are also the guarantee that no malicious process can be introduced in any way to modify or spy the communication behavior as there are no processes!

By design these hardware functions are faster than the possible data rate of the links they are connected to. In case of an attack with a huge amount of data, not only it will be able to handle the incoming data, but by integrated filtering properties it will drop the unwanted traffic and avoid the rest of the system to be overflowed.

As we mentioned a communication monitoring system will be advised of these unwanted data arrivals, allowing protection procedures to be launched with no delay.

Integrated encryption / decryption functions cause only negligible delays (a few hundred of ns) and can be configured to protect sensitive communication paths or to add encryption functionality to devices that do not provide this possibility when sending or receiving data.

Conclusion.

Software does not address all issues in the most efficient way!

Let’s help the powerful and always more demanding embedded systems work better with an independent IO management and networking support. Hardware IO and networking assistance is the way of safe, secure, efficient and scalable electronic architectures.

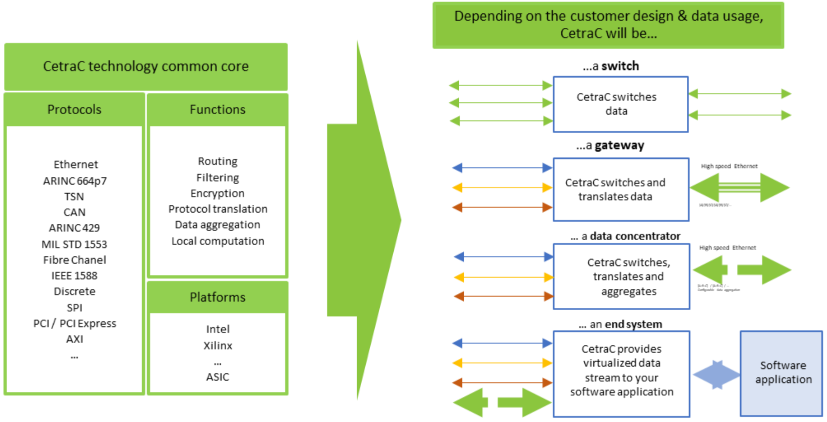
This is where CetraC full hardware and mature technology in an ideal solution. CetraC safety levels have been certified for aircraft, CetraC cybersecurity is qualified by the relevant institutions.

CetraC will help integrate the legacy parts of the automotive electronic architectures seamlessly into zonal and SOA architectures using high speed Ethernet and TSN, delivering instantly the level of safety and cybersecurity required for automated and autonomous operations.

Cetrac.io

Vincent Laporte, CTO

vincent@cetrac.io


CetraC technology common core
Protocols
Ethernet
ARINC 664p7
TSN
CAN
ARINC 429
MIL STD 1553
Fibre Chanel
IEEE 1588
Discrete
SPI
PCI / PCI Express
AXI
...
Functions
Routing
Filtering
Encryption
Protocol translation
Data aggregation
Local computation
Platforms
Intel
Xilinx
...
ASIC
Depending on the customer design & data usage, CetraC will be...
...a switch
CetraC switches data
...a gateway
CetraC switches and translates data
High speed Ethernet
... a data concentrator
CetraC switches, translates and aggregates
High speed Ethernet
Configurable data aggregation
... an end system
CetraC provides virtualized data stream to your software application
Software application