

CSPA : A Communicating Signal Processing Architecture that is scalable, heterogeneous and modular for Eonic's Atlas Universal Digital Signal Computer.

by Eric Verhulst, Marketing Director Eonic Solutions

Introduction

The breakthrough CSPA approach applied in the Atlas Universal Digital Signal Computer from Eonic Solutions, <http://www.eonic.com>, has shifted the system design approach from hardware to software. This is enabled by the use of reprogrammable FPGA as the key components. Around a central, but flexible "high speed backbone" the resulting CSPA architecture features I/O channels and general purpose DSPs. Reprogrammable glue logic as well as fast pre-and post processing in the datastream forms the foundation of an embedded DSP system design methodology. This flexible, modular and highly scalable design drives economic benefits that drastically reduce the development time and technology risk to a cost-efficient level.

Embedded DSP pushing the limits

The current wave of the electronics technology revolution being fueled by Digital Signal Processing, it is no wonder that semiconductor companies have been supplying the market with an increasing number of processing blocks targeted at more specific segments. And while in Digital Signal Processing compute power is an essential element, depending on the application domain, other factors like power consumption, cost and e.g. required precision and I/O are also essential. As a result the DSP offer ranges from low cost, low power fixed point 16bit and 32bit over 32bit floating point to reprogrammable FPGAs with million of gates and offering 100's of GigaOps.

A natural evolution for advanced applications is that they are developed on flexible high-end systems first. This is to prove the algorithms and the feasibility. When the algorithms are stable, developers can start optimizing and when the new offerings on the market have reached the next higher performance level, the algorithms can reach the market in higher volume applications.

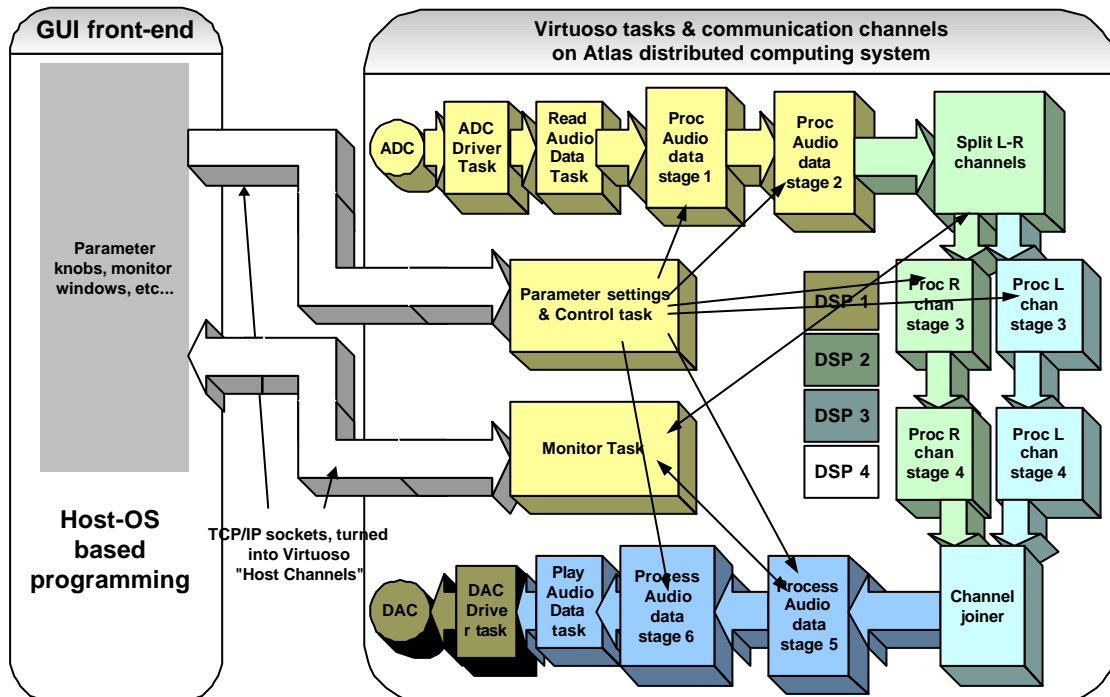
A serious problem with the currently often-used design practice is that the next step in the development process requires a new implementation and new hardware design. Given that the software architecture can be kept fairly independent of the hardware, Eonic Solutions developed a new approach for the hardware platform that is software driven. Making use of the Virtual Single Processor model of the Virtuoso RTOS, the resulting architecture is called **CSPA**. This stands for **Communicating Signal Processing Architecture**, making

the link with the CSP model (Communicating Sequential Processes) on which Virtuoso VSP was based. CSPA is the basic architecture for the Eonic's Atlas product family allowing the scalable use of FPGA and DSP compute processors as well as of I/O channels. As the approach is based on an open architecture, also other software development environments are supported.

The Virtual Single Processor model in Virtuoso VSP

The original design goals of the Virtuoso RTOS were to provide an isolation layer for the real-time embedded developer between the increasingly complex DSP hardware. As high-end systems often use multiple DSPs (up to several 100 to 1000's), Virtuoso VSP not only isolates the application program from the target processor but also from the underlying network topology. Hence, the Virtuoso OS services exhibit so-called "***distributed semantics***", which is implemented by way of a separate system kernel and a communication protocol in the OS. A major benefit of the approach is total scalability without the need to change the application source code of a program. A second benefit is that the combination of the distributed semantics and multi-tasking result in very modular programs allowing to remap the tasks as modules on different target processors. If the hardware architecture supports this model, this results in a very efficient but very flexible system design methodology. For completeness, the Virtuoso VSP technology was originally developed by Eonic Systems, who sold it to Wind River Systems in 2001.

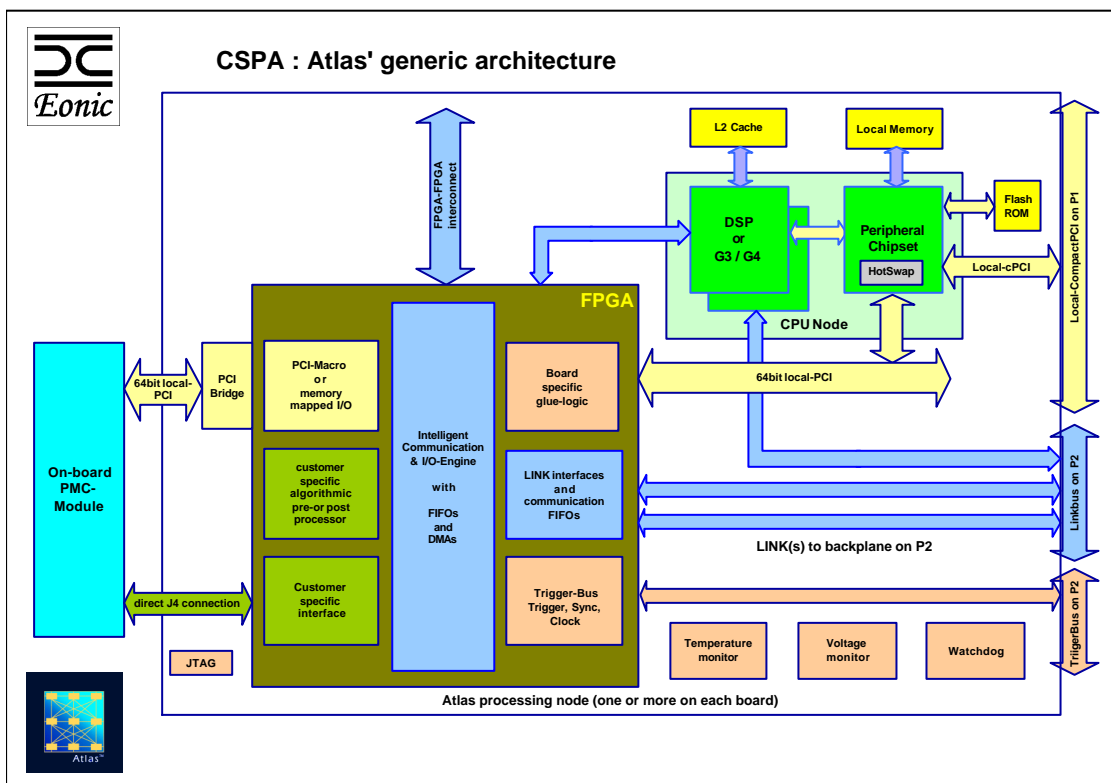
Virtual Single Processor model, independent of target processor and number of processors



CSPA : the Atlas architecture

The above principle being the guidelines for the design of the Atlas product family, a successful result is due to the details and awareness of the boundary conditions under which the system is intended to be used. However, in the development cycle most flexibility is needed at the beginning. This often means that the system should allow for changes to be made quickly without to much concern about the overhead. Therefore Atlas uses high-end mostly floating point DSPs, large high-end FPGAs and a communication backbone that scales its bandwidth while adding or removing processing nodes. Another option was the selection of industry standards whenever possible when they allow to meet the requirements. Hence Atlas uses a CompactPCI architecture and provides for open standard I/O modules like IP (Industry Pack) and PMC (PCI Mezzanine Card). This means that unless the application has very specific I/O requirements, the developer can select from a wide range of I/O modules, available from a wide range of manufacturers. This flexibility within the generic CSPA architecture has given Atlas the attribute of “**Universal Digital Signal Computer**”.

In figure 1., one can see the generic system architecture. Input and output data channels are acquired through IP or PMC modules with the data being passed on to the on-board processor or its memory through the on-board FPGA. This allows for direct processing by the DSP or pre-or post processing in the FPGA.



A system means communication

A single processor, being a von Neumann Machine, can only execute one thing at a time. Even with a multi-tasking RTOS, communication between the tasks is part of the context switch provided by the kernel. Preemptive multi-tasking is then most of all providing modularity and real-time support. When real-time I/O streams and other processors are present in the system, then communication of data becomes an essential part as well. Communication requires set-up times with throughput being limited by the available bandwidth. In DSP systems computations are done on real-time on often high bandwidth datastreams and this can require a large amount of the processor resources. The better DSPs support this by having a prioritized interrupt structure, multiple DMA and multiple independent busses. Hence, a multi-DSP systems is composed of mainly three building blocks, besides memory : DSP compute processors, DMA engines and a communication backbone. These resources are to be shared by all tasks in the application. To minimize the resulting loss of CPU cycles, the communication should be done in parallel with the computation and with minimum delay. The ideal solution is a point-to-point communication network between all the processors. This avoids the decreasing bandwidth and inherent access delay of shared busses. In the Atlas architecture, all interprocessor communication is implemented via so-called "links" . Either they are part of the DSP architecture (the case for Analog Devices SHARC), either they are implemented in the on-board FPGA (the case for Texas Instruments C6202/3 and Motorola PowerPC). Some of the links are directly connected on-board, but by using the P2 connector and high bandwidth linkcables on the backplane, large systems can be assembled with 100's of boards without introducing bottlenecks. The use of FPGA has major advantages. E.g. one can implement processing functions in the datastream with very little overhead, off-loading the DSPs, or implement a broadcasting link at very high speed. One can also build systems composed of different processor types with no additional software development and one can use the links as a standard interconnect to I/O devices. This **decentralized communication backbone** is one of the key features of the CSPA concept. It allows to scale the system in compute power and bandwidth without any extra development and makes maximum use of the available resources. It should be stressed that the gain also comes from a much reduced communication delay. In a multi-processor system, communication delays mean that tasks are kept waiting for data, and hence this can result in idling. This also shows that multi-tasking is often a must to further mask out the effects of the communication delays. In the CSPA architecture, this effect is further reduced by having the communication being executed in parallel (using DMA) with the processing.

A small note on the FPGA resources

Today's FPGAs are no longer just a pool of reprogrammable gates. FPGA chips are now populated with "soft" and "hard" macros, covering functions like specialized memory elements, frequency generation, PLL, PCI interfaces, LVDS I/O ports and even respectable RISC engines. Now reaching over a million of gates, FPGA have really become small systems on-a-chip, but with one major distinctive feature : they are individually reprogrammable. For a board designer, this offers major advantages : e.g. if most of the logic is concentrated in the FPGA, boards need not to be redesigned when a change is made. The change might be needed to increase the system performance (e.g. by putting the filter function between the I/O module and the DSP), to adapt an I/O interface for a new Mezzanine module or often as well to coop with a silicon bug in one of the complex chips on the board that was encountered during stress testing. With no FPGA, such changes often require a costly redesign and months of delay. Of course, the use of FPGA has also some drawbacks. The price of a component is fairly high (but its use eliminates other components on the board), routing all the I/O pins is a challenge and programming the devices requires some expertise. But the main reason to use FPGA today is besides the potential processing power, the flexibility it provides to coop with the increasing complexity and shorter lifecycles of high-end semiconductor components. At Eonic this allowed us to design just a few basic but modular boards that can be reprogrammed for each application and almost unlimited scalability in processing power and communication bandwidth. Hence the central components on the CSPA based Atlas are the FPGA. Most signals on the board are connected to it, allows an enormous flexibility in the way the hardware can be changed by changing the FPGA's program.

I/O engine

In the CSPA model, I/O and interprocessor communication are handled in a similar way. This results in very low overhead for the main DSPs. E.g. data coming from a PMC A/D converter are passed to a buffering FIFO. Using a DMA engine in the FPGA, data is then moved to buffer memory and upon completion an interrupt is generated to the main DSP. As a result, the DSP will not waste cycles on handling the I/O and can start processing on the data as soon as it is available. Further optimization steps can also put some of the lower-level pre-processing steps (e.g. a filter or datashuffling operation) inside the FPGA. As FPGAs allow the operations to be pipelined, this results in much smaller delays. With the Atlas system, the developer can start developing in C on the DSPs. When the application runs correctly, profiling can determine what parts can be put in the FPGA. Access to its resources is through a driver that emulates a standard link (with a FIFO and DMA as essential building blocks). Hence, no redesign is needed and very few program changes. The result can be

that less processor boards are needed, reducing the overall system cost. Similarly, as communication is has now become the central paradigm, in the CSPA based Atlas, processing becomes a peripheral activity of the communication backbone. Hence data no longer needs to pass through the DSP to be communicated, reducing further the overhead.

Trigger-bus

In a DSP system, especially when multiple DSPs and I/O channels are involved, system wide synchronization and programmable clocks are a must to keep the incoming and outgoing data coherent. Often, the accuracy needed will be in the same rage or well below of the duration of a single clock pulse. Such an accuracy cannot be achieved in software trough e.g. the shared PCI bus. Therefore, Atlas features a user programmable trigger bus on some pins of the P2 connector. As these signals are generated inside the FPGA, they allow a great flexibility and functionality that is similar to the PXI approach but without the need for an extra board.

Where did the PCI bus go ?

As Atlas is based on the CompactPCI specification, the reader might wonder where the PCI bus has been left. It is clear that the PCI bus (in fact any other bus) was never designed to scale up systems from 1 to 1000's of processors and the increasing gap between processor speed, memory and bus standards is only making it worse. Although CompactPCI was selected for its technical superiority and increasing availability, in the Atlas system the PCI bus mainly acts as the electrical and mechanical backplane (e.g. for power), generic system control (e.g. hotswap) and generic access to a host processor. However, the PCI bus can also be used as a local I/O bus when e.g. the designer has no other option as to use a CompactPCI based I/O board. Hence the PCI bus main communication function will be to communicate between the DSP processing nodes and the host for e.g. file access, GUI, etc. As a host-OS is often slower - real-time wise and the PCI bus has no real prioritization, this matches well in terms with the host functionality.

A software driven hardware means a reduced development cycle

A major benefit of the Atlas approach is a major reduction of the development cycle. Once the system specifications are collected, a development system can quickly be put together by using off-the-shelf modules. If specific modules require custom engineering or are not immediately available, their function can be simulated by a simple task, allowing to develop the application logic. As

an Atlas system comes delivered with software installed, development can start immediately. The Virtuoso model also allows to start developing on a different target processor with a final selection of the hardware when the application has been profiled. The on-board FPGA allowing to change the hardware without a redesign, major timesavings are also achieved. Current high-end designs at e.g. several 100's of MHz require an expert designer and can easily result in a development and debugging time of 6 to 12 months. Major risks, due to unknown silicon deficiencies are largely eliminated because the Atlas board is used as a proven but flexible component. As the bulk of the glue-logic is implemented in the FPGA, silicon bug fixes are often possible by a simple uploading of a new program.

Reliability issues

As the Atlas boards were designed as basic building blocks of a system from the ground up, major attention was placed on assuring high reliability. As this is done at design time, the impact on the cost is minor. First of all the form factors (3U and 6U CompactPCI, PMC and IP) are robust and easily cooled by way of their mechanical designs. In an adequate housing, they can withstand vibrations, heat and electrical disturbances. Hotswap support was added for testing convenience but also to allow board replacement without having to reboot the whole system. Each board features watchdog timers, several temperature and voltage monitors. Links can be made to operate with time-outs and error detection. As most production problems are solder related, each production run is tested during 48 hours undergoing thermal and voltage cycling, while running stress test programs. Such tests generate mechanical stress in the boards, allowing to detect e.g. weak solder points. Hence shipped systems will seldom fail. In addition, careful attention was paid in the selection of the components. Whenever possible, components were selected on the basis of availability of extended temperature range versions. This allows to deliver extended temperature systems, without the need to redesign the hardware.

Scalability : a matched pair between I/O and processing power

If one looks on the market, several vendors have come up with multiprocessor architectures. The distinguishing factor however is scalability. The ideal scalable system must scale seamlessly from 1 to 1000's of processors and from 1 to several 1000's of I/O channels. Important is to have no scaling boundaries and complete homogeneity when adding boards. E.g. PCI bridges can work well at the board level (if not pipelined), but pose a scaling bottleneck when adding boards. Similarly, memory switches can be fast locally, but can impose tree like structures when adding processing nodes. These problems are less pronounced in the CSPA Atlas architecture. Whenever a processor board is

added, an equal amount of bandwidth in I/O and interprocessor communication is added. Of course, at some point topology restrictions (often connector related) will emerge. But this will often only be so at the rack level, allowing seamless scalability up to a few 100 of processors. Most embedded DSP applications further show clustering behavior, which means that the more system grows, the less global communication is needed.

Application domains

The flexibility and scalability of the Atlas CSPA architecture allows it to be used for a wide range of applications. Early applications cover : ASIC prototyping, Formula 1 engine control, high speed visual inspection, high speed laser beam control, sonar, phase array radars, 3G base stations, and other. The systems cover from just one processor to close to a 100 processors. In some cases, the algorithms were running after just a few days or weeks at real-time speed. Sometimes, development was started on one processor type and ported later on mostly by recompiling to a faster processor architecture.

The scalability is well illustrated by an application that required four systems, each sampling the RF signal at 200 Msps, having D/A outputs at 200 MHz and with the main processing going to calculating FFTs on the signals. This required in total 4x6 450 MHz G4. With no redesign, the "*linkbus*" was reconfigured to work in parallel and demonstrated continuous data rates in excess of 200 Mbytes/sec/channel, while handling the FFTs on the G4. Total processing power for one such system is about 36 GFlops and total bandwidth is about 40 Gbytes/sec, still below the maximum the Atlas system can handle.

Communication LINKs, FIFOs, LVDS, Rapid-I/O and other switched fabric architectures

While the CompactPCI world is now increasingly moving towards so-called switched fabrics on the backplane to overcome the bus bottleneck, it is a field of many competing offerings that are often pushed by particular vendors for particular application domains. To name just a few : Infiniband, RapidIO, NGIO, StarGen, etc. Most of these new schemes are adopting LVDS signaling schemes. LVDS is also what is found inside the high-end FPGA reaching (announced) bit rates of several Gbit/sec. LVDS links are simple, reliable and allow to bridge longer distances with less wires and less power. The use of FPGA again offers advantages. For pure DSP applications straightforward links between the processors are sufficient and require not many resources. If however links have to be made with e.g. third party components, macro cells inside the FPGA allow to build a matching interface. Awaiting widely adopted standards (PICMG 2.17 is a likely candidate), the FPGA solution in Atlas allows such architectures today.



Lower cost, higher volume deployment

The reader is undoubtedly left with the question if the approach taken is affordable for higher volume deployment. In order to answer this question, it is important to understand the development process and the options that were taken by the Atlas approach. The bottom line reasoning is that the embedded systems designer is today squeezed between two major constraints : the first is the ever decreasing time to market, while the second one is the decreasing life cycle of new technologies that at the same time become more complex to design with because of the high frequencies and small component sizes. Hence the notion of an Atlas Development System. Such a system can be put together mostly by using off the shelf Atlas boards and Mezzanine modules. The system will be "over-configured" for maximum flexibility. E.g. more memory that is likely needed and the more expensive versions of the FPGA with more build-in macro functions. This guarantees that during development no resource walls are encountered. Once the application is running (in real-time and in real-use conditions), the software can be optimized and profiled to reduce the hardware resources, e.g. by putting some functions inside the FPGA and by using assembler. After this stage, the Atlas board can be produced with pin

compatible but less expensive components or for larger volumes, a complete new PCB can be developed. Note that recently the FPGA vendors have come up with schemes to order the FPGA with a fixed hardware functionality, by eliminating some of the masks and hereby approaching the power and costprice characteristics of ASICs.

Conclusion

The CSPA approach applied in the Atlas Universal Digital Signal Computer has shifted the system design approach from hardware to software. This is enabled by the use of reprogrammable FPGA as the key components. The resulting architecture is one where I/O channels and general purpose DSPs are attached to a flexible “backbone” that provides for a high speed network, reprogrammable glue logic as well as fast pre-and post processing in the datastream. The result is an embedded DSP system design methodology that is flexible, modular and highly scalable. The economic benefits are a drastic reduction in development time and technology risk reduction at a cost-efficient level.

For more information : www.eonic.com

Eric Verhulst, Marketing Director and founder of Eonic can be reached at eric.verhulst@eonic.com