WHITE PAPER

Chronos Integration – Next Generation SoC

By S. Giaconi & G. Rinaldi, Chronos Tech

Introduction:

System on Chip (SoC) complexity is evolving at an accelerated pace. Design houses keep integrating more diversified Intellectual Properties (IPs) to outpace competition and meet new market demand. Significantly different IPs interacting with each other require a more dynamic fabric to take advantage of their unique characteristics. Neuromorphic engines are the latest differentiator in Machine Learning (ML) enabled SoCs, but their integration with traditional digital architecture used within SoC has been proven to be quite challenging. Addressing current shortcomings can provide a significant competitive advantage in the next few years. Chronos technology can easily overcome those challenges enabling the next generation of SoCs.

Architectures:

From a structural point of view, we can think of an SoC as a system composed of a plurality of IPs that provide the required functionalities, and which are connected by one or more interconnect fabrics thereby enabling effective and efficient dataflow between IPs.

IPs:

They represent the building blocks of the system and up to few years ago they were divided into two main categories: digital and analog; with the first one representing clocked hard macro or synthesizable soft macros, and the second one covering the entire spectrum of pure analog, RF, mixed-signal semi-custom or custom implementations. New customer requirements, as well as technology evolution, have brought the need to better match hardware implementations to the design nature of an IP to maximize Performance Power and Area (PPA) metrics.

Let's examine some examples of IPs used within a modern SoC:

- **High performance CPU:** This is the perfect example of an IP where a lot has been invested over time in terms of architecture, optimization and test. In this case traditional digital synchronous implementation probably represents the best fit to leverage existing methodology flows, enable re-use and minimize security risks.
- **Crypto Engine:** This IP has special security requirements; it needs to protect the private keys and perform undetectable cryptography algorithms to prevent Differential Power Analysis (DPA) attacks. In this case an asynchronous implementation will spread the electromagnetic interference (EMI) in time and if Quasi Delay Insensitive (QDI) logic is used, it will make all the data transition symmetric, shielding against DPA attacks.
- **Neuromorphic Engine:** This engine tries to mimic the behavior of a biological brain, making use of multilevel neural network of artificial neurons connected by virtual synapses. This IP is particularly fast and efficient at quickly inferring decisions, once the training phase is completed, outperforming any other IP in tasks such as computer vision, speech recognition and autonomous decision making. The most natural implementation for this type of IP is asynchronous (mimicking

the behavioral of the brain) and if Bundled Data (BD) architecture is used the total area and power of the IP can also be optimized.

- **Viterbi Decoder:** the Viterbi algorithm provides an efficient implementation of Forward Error Correction (FEC) in order to improve channel reliability. Today, it is used in many digital communications systems, in applications as diverse as LTE Physical Downlink Control Channel (PDCCH), CDMA and GSM digital cellular, 802.11 wireless LANs, dial-up modems, and satellite. The recursive nature of its algorithm enables a concise asynchronous implementation and by using a BD architecture area and power can also be reduced.

Fabrics:

Fabrics enable the exchange of information among IP blocks. High levels of integration are causing on-chip communications and transaction handling to become a major constraint, de-facto limiting the achievable SoC performance. It is important to notice that with so many IP blocks shared across different SoC developers, the fabric is becoming one of the most important differentiators in SoC PPA metric.

Let's review the most common chip interconnect topologies following a chronological development order:

- **Bus:** It represents a simple way of communication between blocks; they can be shared (for low throughput to save area and routing) or can be dedicated (for custom high-speed point-to-point data transfer). Over time the introduction of flow-control and credit-based handshake protocols has enabled easy insertion of pipeline stages between two end-points significantly extending the range of this interconnect.
- **Crossbar:** With the growing number of integrated IPs, crossbars emerged, enabling simultaneous connections between initiators and targets following a matrix multiplexing approach. As the number of initiators and targets increased as well as the data bus width, simple multiplexing started to become unfeasible. Hierarchical crossbar architectures started to dominate at the cost of extra area and latency.
- **Network on Chip (NoC):** NoC introduced the concept of packetized information at hardware level. The technology emerged as a solution to the wiring problem, enabling a separation between the physical layer, the transport layer and the transaction layer of the communication. This concept allowed simple communication (with minimal wiring) within the nodes composing the NoC, while distributing the routing information among them. Packetization of the information enabled address, control, and data to share the same wires, while maintaining the required Quality of Service (QoS) for each transmission. The distributed nature of the router within a NoC has also allowed an easier floorplanning of complex SoC, enabling utilization of the area between IP to implement the interconnect. A significant latency penalty is traded in exchange for reduced routing.
- **Asynchronous QDI NoC:** An Asynchronous QDI NoC has all the benefits of the synchronous counterpart, with the added advantage of reduced latency (data travels at the maximum speed allowed by the technology, without the need to wait for the clock), and simplified timing closure (PVT insensitivity). The trade-off is area usage, which in the case of a "distributed NoC" is quite small if compared to the total area of the SoC.

SoC evolution:

Following the introduction of Application Specific Integrated Circuit (ASIC) chips in the market, SoCs differentiated themselves by including one or more Central Processing Units (CPU) within the silicon die. Initial versions of SoCs had quite a few IPs, but the trend of adding IPs to the system quickly led to complex monolithic chips; this resembles the functionality of previous entire discrete computer systems, including dynamic memory, input/output ports, peripherals and secondary storage systems. Integration of Analog and Radio Frequency (RF) IPs created what nowadays we can call the first modern version of SoC which in the example of Figure 1 it is named SoC Gen1.

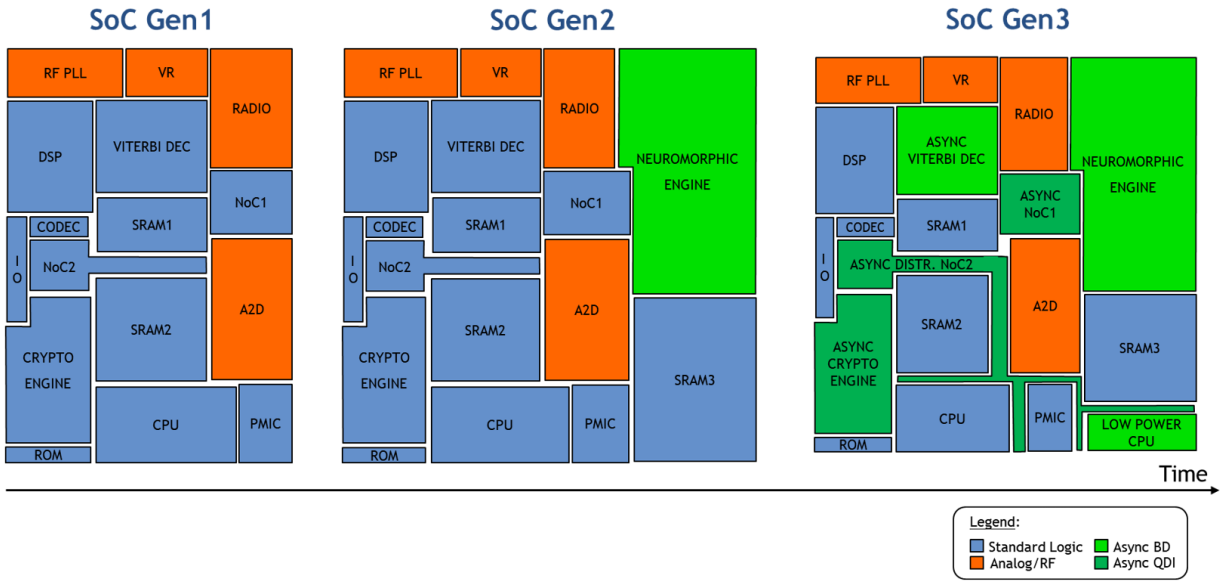

Figure 1: SoC evolution

The second modern generation of SoC (Gen2 in Figure 1) is what several big semiconductor companies are currently targeting with the proliferation of distributed Artificial Intelligence (AI) and Machine Learning (ML). It introduces a Neuromorphic engine as a stand-alone IP to help speed up real time decisions and accelerate computer vision and/or speech recognition. This IP has been introduced in Gen1 as a synchronous model of a more natural asynchronous IP (often utilizing a GPU IPs acting as neural networks) but the real advantage in terms of performance and power will be achieved with the integration of an event driven asynchronous BD neuromorphic engine.

Figure 2 shows the difficulties that SoC architects are facing in integrating asynchronous IPs within traditionally synchronous SoC architectures. How to reconcile the dataflow? How to take care of the protocol? What about test? Etc.

The vision for the future is even more complex. To take full advantage of different IPs and fabric implementation, the SoC architect should be able to pick and choose each implementation and seamlessly integrate it within the design to allow for the best PPA optimization possible.

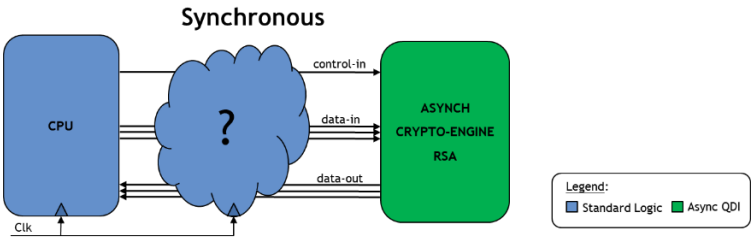

Figure 2: Challenges in integrating Asynchronous IP within an SoC

SoC Gen3 in Figure 1 depicts the future of SoC where the best implementation is chosen for each IP and a plurality of fabrics make sure to efficiently connect them in order to achieve the best result.

Chronos advantage:

The introduction of Chronos enables a quick integration of different IP implementations within the same SoC. Chronos not only enables connection between synchronous IP with reduced routing and often optimized latency, but it can easily integrate any flow-control or credit-based synchronous protocol to any asynchronous channel, providing also a conversion of different channels styles if needed. It can also be used to connect two different asynchronous IPs with reduced routing, making it the ideal candidate for heterogeneous IP integration.

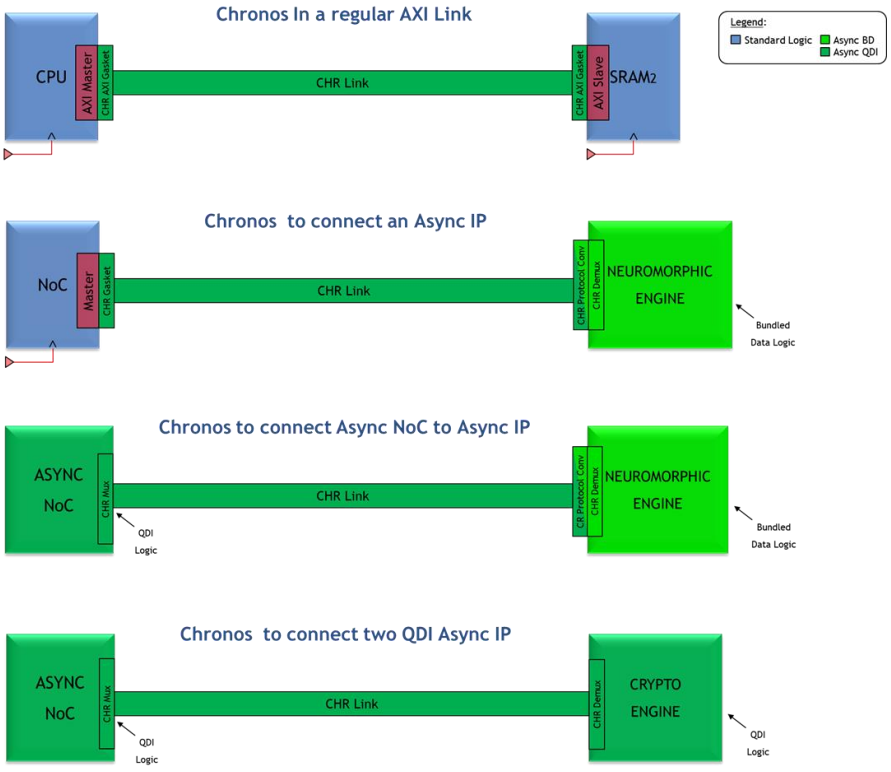

Figure 3: Integrating Asynchronous IP with Chronos

Figure 3 shows how Chronos can easily connect different type of IPs. The first drawing in Figure 3 represents the replacement of an AXI link between 2 standard synchronous IPs with a Chronos link, leading to smaller routing and better latency performance. The second drawing shows the connection of a neuromorphic engine (designed in asynchronous BD) to a traditional synchronous NoC through a Chronos Channel and Chronos protocol converter. The third drawing, instead depicts the implementation of an asynchronous QDI distributed NoC (to improve resilience to PVT and decrease latency among IPs), connected through a Chronos channel to a BD asynchronous neuromorphic engine. The last drawings in Figure 3 shows the connection through a Chronos channel of an Asynchronous distributed NoC with a QDI crypto engine.

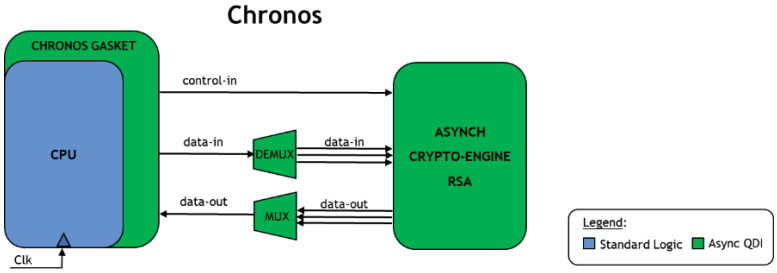

Figure 4: Integrating Asynchronous IP with Chronos

Reviewing again the issue of integrating an asynchronous IP to a standard synchronous design (depicted in Figure 2), Figure 4 shows the simple solution utilizing Chronos. A Chronos gasket is inserted to convert the synchronous signaling to a compressed Chronos signaling and right before reaching the asynchronous IP the signals are de-muxed / muxed to be able to be manipulated by the IP.

Summary:

In summary, System on Chip (SoC) requirements are becoming more complex. Chip architects are facing new challenges requiring complex IP integration and aggressive PPA optimizations. New generation of IPs are becoming available, providing better security, power and performance at the cost of a more challenging integration. Chronos technology can solve these challenges providing the perfect fabric for seamlessly integrating all the new IPs within existing architectures, leveraging a combination of well-known and reliable standard IPs with new event-based counterparts, facilitating a smooth transition and integration with the future.