WHITE PAPER

Chronos TileLink- Joining the RISC-V Revolution

By S. Giaconi, Chronos Tech

Introduction:

The progressive advancement in Machine Leaning (ML) and Artificial Intelligence (AI) on the edge requires significant computing power distributed in a wide range of different devices, creating the need for a non-proprietary standardized and efficient computational ecosystem. The open-source RISC-V architecture is catalyzing attention in the semiconductor industry as the solution to these new requirements. RISC-V has been designed from the beginning with the intent to be freely extensible and customizable to fit upcoming market segments and it has reached widespread support from chip and device makers in the field. Chronos Tech fully supports the standard interconnect used by RISC-V, TileLink, enabling a seamless replacement for the original pipelined fabric with a small footprint, low latency and secure alternative, simplifying timing margin and boosting performance in advanced heterogeneous multi-core SoCs.

RISC-V opensource architecture

RISC-V is a free and open Instruction Set Architecture (ISA) based on the established Reduced Instruction Set Computer (RISC) principles. Founded in 2015, the RISC-V Foundation today comprises over 250 members (including top industry players) building the first open, collaborative community of software and hardware developers. Born in 2010 at the University of California, Berkeley as a research project, the RISC-V ISA evolved into a global effort, delivering a new level of free, extensible software and hardware freedom on architecture, paving the way for the next 50 years of computing design and innovation.

Why RISC-V?

RISC-V is becoming more and more interesting for hardware developers, especially those building embedded IoT or edge devices. Like Arm processor counterparts, the architecture only requires a small amount of power when compared to full blown x86 devices from Intel or AMD. Some benchmarks are even indicating that RISC-V cores can be even more energy efficient than Arm ones, which have become the gold standard for mobile devices.

The major advantage of the new ISA over Arm, however, is that RISC-V is open source. OEMs can use the specification to design and manufacture chips without the need to pay royalties, which is a substantial portion of the price of bringing an Arm chip to market. On top of that RISC-V's "permissive" BSD licensing enables customized RISC-V chips to be made proprietary for IP protection, if desired, creating an economic viability for developers.

The fact that the ISA has been built from scratch, also avoids some of the problems that are embedded into older processor designs for back compatibility. The project is getting support and investment from major industry players, including Samsung, Google, IBM, Nvidia, and Qualcomm, among others.

Rocket Chip

Rocket Chip is Berkeley Architecture Research (BAR)'s parameterizable chip generator, and it serves as the basis for all the RISC-V implementations developed by the BAR. Rocket Chip can generate RTL RISC-V implementation that include virtual memory, a coherent multi-level cache hierarchy, IEEE-compliant floating-point units, and all the relevant infrastructure to be able to communicate with a running system.

The SoC can be configured with a single or multiple processor core, such as the in-order Rocket core or the out-of-order BOOM core. The architecture of the whole SoC, including the type/size/level of caches, the number of on-chip buses and the IP devices with specific buses, are all configurable at design time.

Figure 1 shows the one implementation of a Rocket Chip. The generator is written in Chisel (which is RTL embedded in Scala) enabling the full power of Scala for writing generators, including object oriented and functional programming as well as versatile parametrization.

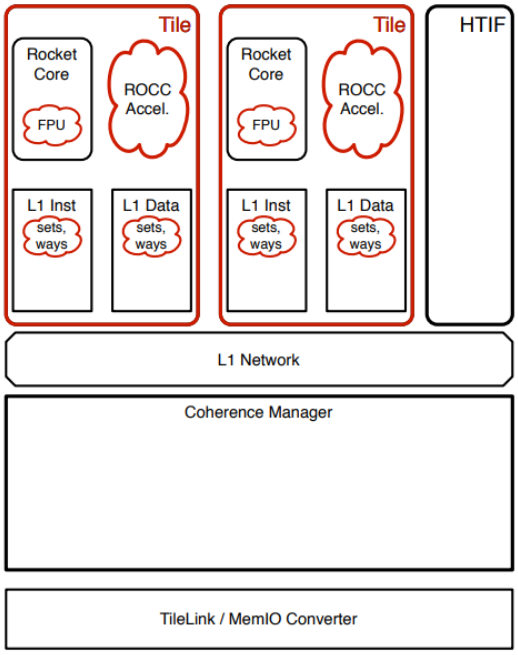

Figure 1 – Rocket Chip components.

The SoC generator helps tuning the design under different PPA constraints in different technology nodes, simplifying porting, and enabling an easy generation of fast simulation code, assertions, verification environment and HDL netlist. Parameters include number and type of cores, instantiation of floating-point units and vector units, cache sizes, associativity, number of TLB entries, cache-coherence protocol, number of floating-point pipeline stages, width of off-chip I/O, and more.

Rocket Chip generator takes care of building the whole SoC including the fabric connecting the different components. TileLink is a chip-scale interconnect standard, providing multiple masters with coherent memory mapped access to memory and other slave devices. TileLink is designed for use in a SoC to connect general-purpose multiprocessors, co-processors, accelerators, DMA engines, and simple or complex devices, using a fast and scalable interconnect providing both reduced latency and high-throughput transfers. The TileLink specification includes three conformance levels for attached agents, which indicates which subset of the protocol they must support. The simplest is TileLink Un-cached Lightweight (TL-UL), which supports only simple memory read and write (Get/Put) operations of single words. The next most complex is TileLink Un-cached Heavyweight (TL-UH), which adds various hints,

atomic operations, and burst accesses but without support for coherent caches. Finally, TileLink Cached (TL-C) is the complete protocol, which supports use of coherent caches.

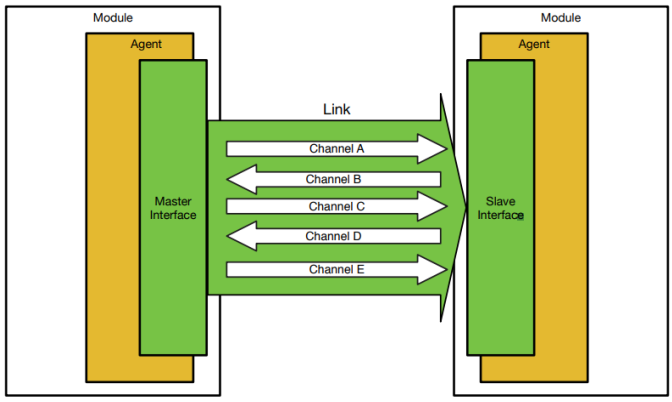

Figure 2 – The five channels of a TileLink.

TileLink protocol defines five logically independent channels over which messages can be sent by agents. Channels are directional, in that each passes messages either from master to slave interface or from slave to master interface. Figure 2 illustrates the directionality of the five channels. The two basic channels required to perform memory access operations are:

Channel A: Transmits a request that an operation be performed on a specified address range, accessing or caching the data.

Channel D: Transmits a data response or acknowledgement message to the original requestor.

The highest protocol conformance level (TL-C) adds three additional channels that provide the capability to manage permissions on cached blocks of data:

Channel B: Transmits a request that an operation be performed at an address cached by a master agent, accessing or writing back that cached data.

Channel C: Transmits a data or acknowledgment message in response to a Channel B request.

Channel E: Transmits a final acknowledgment of a cache block transfer from the original requestor, used for serialization.

Chronos Technology

Chronos technology was specifically designed with the goal of enabling the next generation of complex SoC in the latest FinFET nodes. It aims at deploying robust and secure on-chip and off-chip fabric, while drastically reducing interconnect latency and overheads. Such unique characteristics allow substantial area reduction, effortless integration of IPs, resilience to PVT and enhanced security; addressing the very source of the limitations in current SoC design.

Chronos solution is relying on the synergy of four independent technologies:

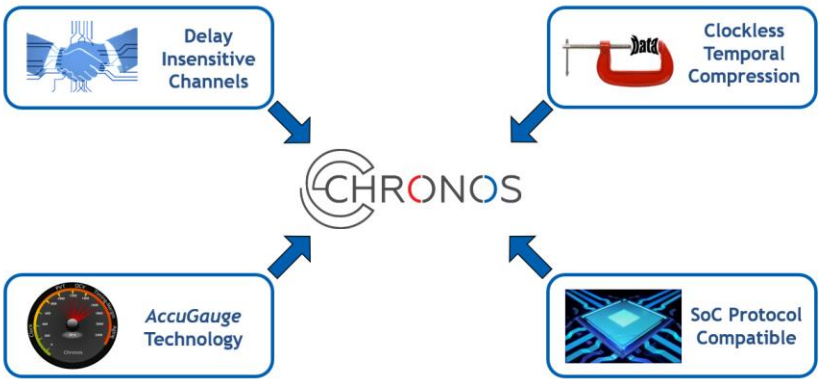

Figure 3 – Chronos Technology cardinals.

Delay-Insensitive (DI) channels move the validity of the data from the clock edge to the data itself using a choice of data encoding and "handshake" protocol. To practically leverage the benefits of DI channels in silicon we rely on the usage of Quasi-Delay-Insensitive (QDI) circuits. In a QDI circuit, there is no need to wait for a clock signal to start a computation. Consequently, the next computation can be initiated immediately when the result of the first computation is completed. The benefits of delay insensitivity have been long known to address resilience to PVT, and with that, simplification of timing closure as well as significant reduction in TTM. In the past, this set of benefits was usually offset by the significant increase in routing requirements and area, not to mention the resistance in using a non-standard design methodology (as well as set of tools) decreasing the engineering confidence level on a successful tape-out. Chronos Technology comes with a solution to all the above concerns, mitigating the risk factors while leveraging the benefits.

Chronos clock-less temporal compression mitigates the routing and area limitation previously discussed. It enables the serialization of the data based on a specific ratio without requiring any high-speed clock. This ratio can be adjusted at design time allowing to trade congestion vs link latency. Fine tuning of compression ratio and library type can be performed on a channel-by-channel basis.

SoC Protocol Compatibility: Chronos integrates directly with the most common modern interfaces (i.e. TileLink, AMBA-AXI, AMBA-CHI, OCP, etc.) as well as with any custom protocol used within dedicated IP and/or NOC routers which use flow control or credit based. Chronos is also able to interface with non-timing critical interfaces such as control registers, fuses, interrupts, etc., to greatly reduce routing congestion of thousands of miscellaneous signals at the top level.

AccuGauge probe enables the measurement of actual maximum performance of each Chronos channel on silicon at a specific PVT condition. This unique feature enables the testing and margining of each individual channel simplifying EV, qualification and DPM analysis. This unique probe can also be used to throttle on-the-fly voltage and speed while still guaranteeing functionality for each link enabling significant power saving compared to more traditional static AVFS table.

Chronos TileLink gasket

Chronos has developed a custom gasket to interface to a standard TileLink interface. Insertion of the "gasket" and creation of the new top level can be completely automated once the interfaces are properly described. The Chronos Front-End insertion flow makes sure to eliminate the original pipeline stages and clock distribution, insert the gaskets at the interfaces, connect them through Chronos repeaters, provide the new netlist and timing constraints. The Chronos Back-End flow takes care to close timing and implement the connection.

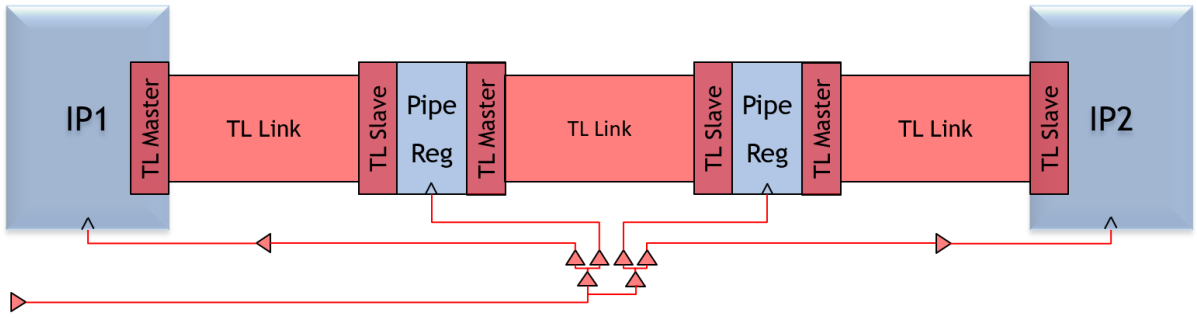

Figure 4 – Pipelined TileLink.

Figure 4 shows a pipelined TileLink connection between two IPs, a master (IP1) talking to a slave (IP2) with two retiming pipeline stages in the middle. It is important to notice that clock insertion step is necessary in order to guarantee a synchronous timing reference for each block and allow correct functionality.

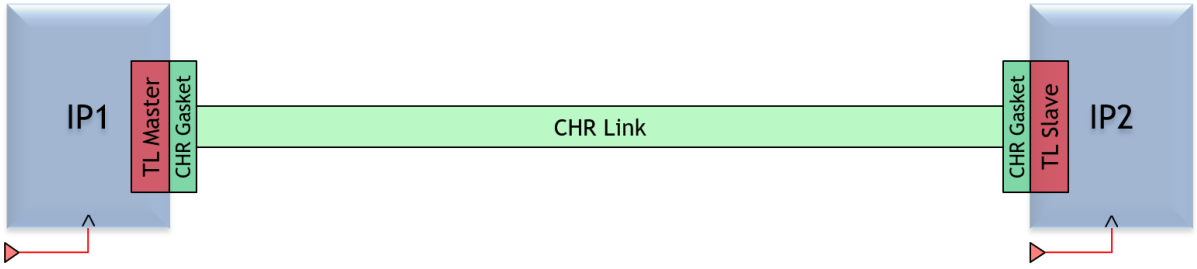

Figure 5 – Chronos TileLink.

Figure 5 shows the same connection but implemented with Chronos. The Chronos gasket connected to the master takes care of providing protocol conversion and data compression, while it is seen by the Master interface as a TileLink Slave interface. Data moves through the Chronos Link following handshake protocol between Chronos repeaters (no clock necessary). On the receiving end, the Chronos Gasket decompresses the data and converts the protocol back to TileLink to be received by the Slave interface on IP2. The Chronos gasket on the right end side is seen by the TL Slave gasket as a TileLink Master. Because of the Chronos patented compression technology, quite often the CHR link has a smaller width than the original one. The CHR link does not need a clock insertion step and the clocks feeding the two IPs can be either mesochronous or (if throughput is maintained) asynchronous. If the distance between the IPs is

substantial with respect to the original clock frequency (RC dominated connection) the latency of the Chronos link can be significantly shorter than the original one (no need to wait for a clock cycle to push data forward).

Rocket Chip with Chronos

To verify proper functionality of the Chronos TileLink we have created a modified version of the Rocket Chip with default settings. Two of the original TileLinks have been replaced with their Chronos versions.

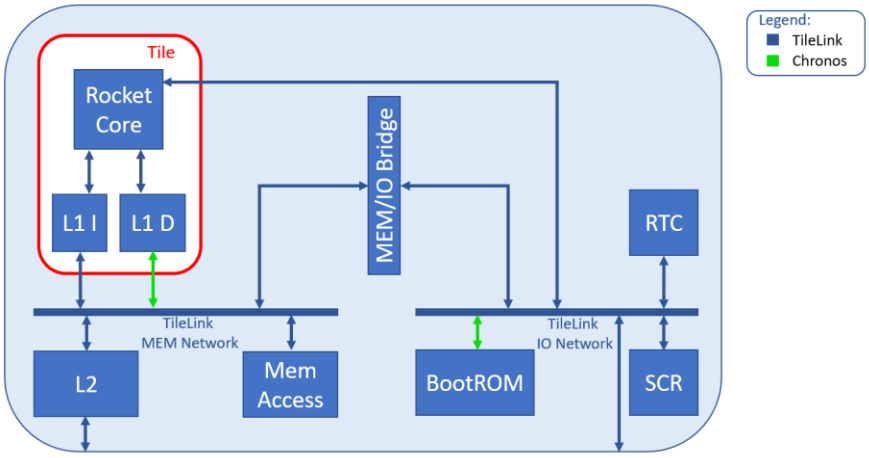

Figure 6 – Rocket Chip w/ Chronos.

The first one is the connection to the BootROM, which is a simpler TileLink Uncached Lightweight, and the second one is the connection between the Rocket Core and the Data Cache which is a full-blown TileLink Cached conformance level.

Figure 6 shows in green the link which have been replaced by Chronos Links.

A full regression (RISC-V assembly tests) has been performed confirming correct functionality for both the interfaces replaced. See Figure 7 for a snapshot of the regression log.

```
[ PASSED ] output/rv64ui-v-lwu.out      Completed after 212114 cycles
[ PASSED ] output/rv64ui-v-sd.out       Completed after 303742 cycles
[ PASSED ] output/rv64ui-v-slliw.out    Completed after 183126 cycles
[ PASSED ] output/rv64ui-v-sllw.out     Completed after 220706 cycles
[ PASSED ] output/rv64ui-v-sltiu.out    Completed after 182814 cycles
[ PASSED ] output/rv64ui-v-sltu.out     Completed after 198306 cycles
[ PASSED ] output/rv64ui-v-sraiw.out    Completed after 184818 cycles
[ PASSED ] output/rv64ui-v-sraw.out     Completed after 221558 cycles
[ PASSED ] output/rv64ui-v-srliw.out    Completed after 183462 cycles
[ PASSED ] output/rv64ui-v-srlw.out     Completed after 221186 cycles
[ PASSED ] output/rv64ui-v-subw.out     Completed after 197394 cycles
```

Figure 7 – RISC-V assembly tests log.

Figure 8 shows the signals at the BootROM interface after Chronos insertion confirming correct functionality

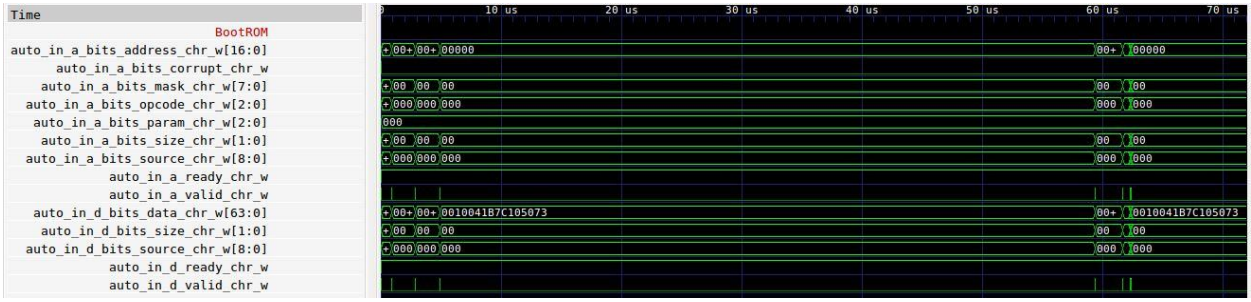

Figure 8 – BootROM interface signals w/ Chronos.

Figure 9 shows the signals at the Data Cache interface confirming correct behavior.

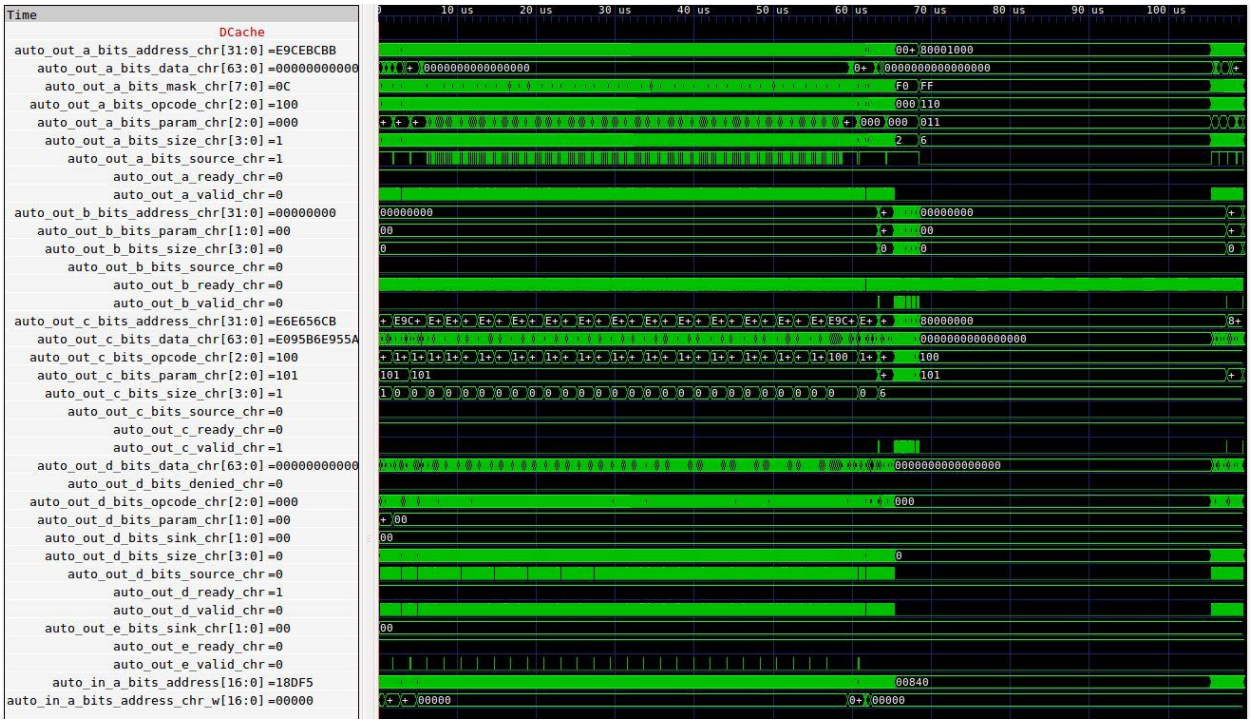

Figure 9 – Data Cache interface signals w/ Chronos.

Chronos Benefits

Chronos technology enables customization for each independent link, allowing accurate and easy PPA tradeoffs. Different scenarios can be investigated, using the *Xplorer* tool (GUI), before the final implementation is selected.

Let's look at the optimization opportunities for the two interfaces implemented with Chronos in the Rochet chip, assuming a recent FinFET technology as an example.

The BootROM usually runs at a relatively low frequency clock (i.e. 500MHz) and can be placed quite far (i.e. 6mm) from the memory bridge. Considering the functionality of the BootROM, the optimization goal would be to minimize the routing to reduce real estate area on chip. In the example we have seen the payload of the TileLink interface (channel A+D) connecting to the BootROM was 118bits wide. By using a compression ratio of 8, the Chronos implementation can reduce the payload routing to just 40bits (a saving of 66%) maintaining the same worst-case latency.

For the Data Cache link to the L2 Cache, instead, the scenario is completely different. Memory caches usually run a quite high speed (i.e. 1GHz), and in a multi-core SoC, the distance to the memory controller can be quite long (i.e. 5mm). In this scenario it is very important to minimize latency for those interconnects, because memory latency is affecting the performance of the whole system. In the example we have seen the original implementation was using 9 pipeline stages to cover the 5mm distance of the TileLink running at 1GHz resulting in a 10ns channel latency. The Chronos implementation removes the synchronous pipeline stages and produces a worst-case channel latency of just 5.11ns, de facto cutting the latency in a half.

Conclusion

Chronos Tech has developed a dedicated "gasket" tailored to the latest TileLink interface, which is becoming a key component in the widespread RISC-V architecture. It enables a seamless replacement for the original pipelined fabric with a revolutionary clock-less optimized version. We demonstrated the tunability of the Chronos channel, allowing low latency and small footprint implementations, all of them while simplifying timing margin and boosting performance and security in advanced heterogeneous multi-core SoCs.