A Reconfigurable Asynchronous SERDES for Heterogenous Chiplet Interconnects
Jainaveen Sundaram, Srinivasan Gopal, Thomas P Thomas, Edward Burton, Erika Ramirez
Lozano & Tanay Karnik
Intel Corporation, Hillsboro, USA
jainaveen.sundaram@intel.com.

Abstract
With advancing packaging technologies, multi-die integration is gaining prominence among players in the semiconductor industry for its ability to create a system with heterogeneous functionalities (Digital Logic/High-speed I/O etc. at its most efficient process node) hence improving overall silicon yield. This work presents a source-synchronous die-to-die I/O using a self-timed loop (behaviorally analogous to a gated ring oscillator) that generates high-speed edges, eliminating any power-hungry PLL/DLLs found in traditional die-to-die I/O interfaces. In addition, the proposed asynchronous architecture enables easy reconfigurability of data rates under a small form factor and efficient design-reuse. The scheme achieves a maximum of 4.8Gbps/wire at 0.4pJ/b generating an effective shoreline bandwidth of 1.47Tbps/mm.

Keywords
Chiplets, 2.5D, 3D IC, asynchronous, interposer, high speed I/O, wireline communication.

1. Introduction
Multi chip module technologies have garnered popular attention for meeting the computational demands while extending Moore’s law [1]-[7]. As the development of systems-on-chip have increasingly become cost-prohibitive in ultra-deep sub-micron technologies, chiplet based design provides cost-effective and high-density integration. Fig. 1 (a) illustrates the existing problem landscape showing the underlying motivation for chip disaggregation [11]. Multi-functionality integration such as logic/memory, I/O, RF and mixed-signal systems have their respective challenges with regard to density, leakage, speed, voltage and passive structures limiting their optimum performance. Chip disaggregation leverages higher performance, lower cost and rapid prototyping. System design using chiplets is one the key enabler of 2.5D and 3D heterogenous integration using bridges or interposers that has been adopted in this work [8].

Fig. 1 (a) Problem landscape and it’s underlying motivation for chip disaggregation (b) Die-to-die communication direction and its architectural trade-offs

At the foresight of Dr. Gordon Moore in 1965, it is feasible to have economies of scale by building larger systems using smaller functions, which could be separately packaged and interconnected [2]. In this context, several industry leaders including Intel, Nvidia and Marvell have demonstrated chiplet-based heterogenous system designs [2]-[9]. Die-to-Die (D2D) I/O’s like Intel’s AIB [3], TSMC’s LIPINCON [12] are the current standards to connect chiplets within a System-in-Package (SiP). With a need to process exponentially growing data, heterogenous integration of chiplets offers a promising approach for quick time to market at reduced cost of development. Albeit existing challenges, the chiplet designs provide a promising direction in integrating multiple electrical functionalities such as analog mixed-signal and RF IPs, compute and memory to deliver optimum performance.

Chiplet design is an advanced packaging technology solution that offers a paradigm shift in the current ecosystem that supports scalable electrical interfaces [3]. As in a typical inter-chip environment, Figure-of-Merit for any inter-chiplet communication scheme are throughput offered (Gbps/mm) at the energy cost (pJ/b). The goal of for designing a die-to-die I/O is to maximize throughput (thus resulting in lesser bump count along the chip periphery) while minimizing its energy footprint.

One such popular interface would be to implement serial interface links using SERDES, such as the PCI express, which use a few serial high-speed wires, operating at 10s of Gb/s, to transport data. In a traditional implementation, these interfaces are more expensive to design and potentially experience higher latency and incur higher power. Specifically, traditional SERDES architectures use power-hungry PLL or DLL based clock recovery blocks to recover clock and data from the transmitted data at the receiver [3]. In this work, we develop an asynchronous SERDES
methodology using a self-timed loop to avoid high power clocking for short reach interconnects used in chiplet designs. Further, the design serial data rate is made reconfigurable for wide application space and trade power efficiency for data rate.

2. Motivation towards Asynchronous SERDES Design for Short Reach Chiplets

In order to obtain high density interconnects with high bandwidth, there is a need to increase bundled wire data rate (Gbps/mm) while minimizing energy expended (pJ/b), which can be derived from the problem landscape. Fig. 1 (b) presents the wire data rate and wire density trade-offs of the present architectural trends of high-speed serial links [2]. In the design space exploration of wire data rate and density, as one treads on the higher data rate spectrum, the power consumption increases exponentially. This architecture targets a moderate wire density under a low power regime. Furthermore, easy reconfigurability of data rate enables design re-use across applications, reducing the overall cost and improving time to market.

This paper presents a reconfigurable asynchronous SERDES architecture that targets medium data rates and higher shoreline bandwidth that increases the wire density under low power. The flagship contributions of this work include:

1. A self-timed loop to generate high speed pulses with delay calibration schemes that replaces the need of power-hungry blocks such as DLLs and Phase interpolators for a short reach channel.

2. A reconfigurable architecture that allows to program the amount of serialization and deserialization of data, which is useful in control the wire density as against data rate spanning wide application space.

The paper is divided into multiple sections as follows. The architectural and circuit design details are explained in

Fig. 3(a) and (b) Proposed asynchronous SERDES architecture for serial inter-chiplet link design

Section 3. Section 4 discusses the resulting simulation results and discussions. Section 5 concludes the paper with summary.

3. Architecture and Circuit Design Details

Fig. 2 shows a typical die in a System-in-Package (SiP) with die to die I/Os lined along the chiplet periphery. The TX and RX along the die boundary is made up of smaller modular self-contained TX and RX clusters. A TX and a RX Cluster together constitute the N-pump SERDES (NPS) described in the next few sub-sections.

Fig 3 shows the high-level block diagram of the N-Pump SERDES. On the Transmit die, data (synchronous to on-die clock, referred to as mesh clock) is split into N-bit busses and fed into the clusters. Each TX cluster is made of serializers (for data and strobe) that are used to create these N-bit chunks along with an asynchronous clock generated by the Clkgen. The output of the serializers is sent across the channel, using custom drivers designed and optimized for the interposer model. The
data is accompanied by source side mesh clock and DDR strobe generated off the asynchronous clock.

On the receive side, each incoming serial line is fed to a de-serializer, with a copy of strobe and mesh clock (from the TX side). The de-serialized data is then passed to a Clock-Domain-Crossing FIFO on the receive die (not shown in figure) and onto the fabric.

3.1. Transmit Cluster

Fig. 3(a) shows a die-boundary, with TX/RX Clusters laid out on the periphery. As shown to the right of Fig. 3(b), the TX Cluster is built of unit serializers, instantiated for both data serialization and strobe generation. The Clockgen (seen in Fig. 4) is responsible for generating asynchronous high-speed pulse train for every batch of valid data. Data serializers pick up chunks of N-bit parallel data from the fabric and work off this pulse train to serialize and send data using the TX driver cells.

3.1.1. Asynchronous Clockgen

As shown in Fig. 4, serializers are built using flops with MUX’es in between them. Each MUX chooses between incoming data (load phase) and data passed from preceding flop (shift phase). All serializers in the design (data, strobe and ones within Clockgen) are set to the same depth before coming off reset. The Strobe signal is generated using a separate serializer with the N-bit input hard coded to a 1-0-1-0... sequence. The Strobe serializer works off the negative edge of the pulse-train, aligning the strobe in quadrature with transmitted data for maximum sampling margin at RX.

3.1.2. Data and Strobe Serializers

As shown in Fig. 4, serializers are built using flops with MUX’es in between them. Each MUX chooses between incoming data (load phase) and data passed from preceding flop (shift phase). All serializers in the design (data, strobe and ones within Clockgen) are set to the same depth before coming off reset. The Strobe signal is generated using a separate serializer with the N-bit input hard coded to a 1-0-1-0... sequence. The Strobe serializer works off the negative edge of the pulse-train, aligning the strobe in quadrature with transmitted data for maximum sampling margin at RX.

3.1.3. TX Driver

For our design, the channel used is a 1200um trace drawn on a passive interposer connecting to the chiplet through micro-bumps. ESD diodes – a major bottleneck for high-speed low-power signaling is discarded. Sized up custom inverters was used to successfully transmit high speed data and strobe signal through the channel, with enough eye margin as discussed in the Results section.

3.2. Receive Cluster and De-Serializers

The receiver is made of unit de-serializers, each connected to a serial data line. Strobe and incoming mesh clock are passed through a clock tree before being passed into the de-serializers.

Fig 5 shows the circuit diagram of the de-serializer. Positive and negative edge of the Strobe is connected to cascaded flip-flops to capture and phase-shift the incoming
serial data. Mesh clock is used to sample the phase-shifted data to obtain parallel data passed onto the CDC FIFO.

3.3. Calibration and Redundancy lines

While the lack of PLLs/DLLs greatly decrease the energy and area footprint, it makes the design highly susceptible to Process/Voltage/Temperature (PVT) variations. Self-timed loop frequency, Clock-Data mismatches, driver strengths are some critical parameters needing to be monitored for successful operation.

Our design tackles all these issues by a robust and periodically run calibration algorithm. Counters identify frequency of the self-timed loop, while a standard BIST feature capable of adjusting simple delay lines (buffer chains attached to each signal at the RX) compensate for any delay mismatches. Redundancy lines are also laid out between the TX and RX, capable of re-routing traffic in case of faulty or broken lines post assembly.

4. Simulation Results and Discussions

4.1. Simulation Environment

The circuit schematics for the proposed N-pump SERDES is designed using Intel 10nm process with approximated layout parasitics [10]. A 130nm fab process was used for the physical design of the passive trace of length 1200um. This channel model was then extracted and plugged back into the schematic along with estimates for bump capacitance models. The Cadence Virtuoso Spectre ADE environment is used for co-simulation of two different environments to incorporate the channel model and the circuit schematics. The design was simulated for a typical process corner with 0.85V supply and at 27°C.

4.2. Simulation results

Mesh clock was set to 1.2Ghz frequency and the design was simulated with pump modes of 3/4/5 (by trimming serializer depth). Asynchronous shift clock generated by the Clockgen settled at 6.16Ghz (on average) which was passed to all serializers. Serialized data had an average 170ps UI accompanied by a strobe running at 2.45GHz. The effective layout parasitics include an effective RC routing of 250fF and 330 mOhms, followed by a bump capacitance on either sides of the channel interface.

The signal integrity simulations of the SERDES link is obtained by plotting the eye-diagram of the strobe path and the data paths respectively. The eye diagrams of the simulated results are plotted at the receiver end in Fig. 6. As shown in Fig. 6 (a), the strobe eye shows an eye opening of 0.75V eye height and an eye width of 0.77 UI. The data eye is also plotted in Fig. 6 (b), where the eye opening has a height of 0.66V and an eye width of 0.67 UI, respectively.

A Strobe signal is shared among all data lines within a cluster, with strobe power amortized across all parallel data lines. Energy/bit calculation across pump-modes and data-to-strobe grouping combination was plotted (as shown in Fig7).

Fig. 5 N-pump deserializer circuit

Fig. 6 Signal integrity simulations using eye diagram: (a) Strobe eye (b) Data eye

Fig. 7 Energy/bit for different modes of N-pump SERDES plotted against Data/Strobe groupings. Optimal configuration (highlighted in red) was picked to balance power efficiency, jitter budget and tuning range of delay lines in the RX.
We observed by grouping 16 data lines (shown in red box) to one strobe, we achieve optimal trade-off between power efficiency, jitter budget and correction range.

5. Conclusion

This work presents a PLL-light reconfigurable SERDES interface for obtaining high bandwidth wire density at low power. The architecture uses an asynchronous design by using high frequency clock pulses in a self-timed loop for maintaining edge sanctity. A delay calibration scheme is also presented for any edge variations without the use of any power hungry PLL/DLL. A reconfigurable N-pump scheme allows user the choice of trading high density by lowering data rate and the associated power consumption. The simulations show a promising 0.4 pJ/bit power efficiency and a highly scalable architecture achieving a shoreline bandwidth of 1.47 Tbps/mm.

6. References