Abstract- This work presents a low-complexity physical link micro architecture for a mesochronous on-chip communication insensitive to clock skew. This new link architecture, called LIME, integrates a low-latency flow control scheme; this feature may ease the set up of reliable Network-on-Chip infrastructures. The proposed architecture also supports virtual channels that are multiplexed on a single physical link. LIME can be integrated in a conventional digital design flow since it is implemented by means of standard cells only.

Keywords: Network on Chip, Physical link, Mesochronous, Flow Control, Synchronization

I. INTRODUCTION

A. The Network on Chip Paradigm

Market, application and nanoscale technology trends lead to new challenges for on-chip communication in complex Systems-on-chip (SoCs). The academic and industrial research is focusing on the Network-on-Chip (NoC) communication paradigm ([1],[1]) as an alternative to classical shared-bus schemes (e.g. AMBA AHB or AXI buses, STbus and so on). NoC promises to increase communication scalability, flexibility, reliability and system bandwidth exploitation while reducing power consumption. NoC is particularly suited for emerging Multi Processor-SoC (MPSOC) architectures. Moreover, NoCs provide SoC architects with a tool for designing on-chip communication systems quickly, thus increasing productivity and reducing time-to-market. NoCs are based on a packet-switched paradigm; thanks to the abstraction layers of the ISO/OSI stack (physical, data-link, network, transport, application) NoCs provide a methodology for designing an interconnect architecture independently from the attached Intellectual Property (IP) cores [2]. The main building blocks to arrange a NoC are: (i) the Router (R), (ii) the Network Interface (NI) which is in charge of protocol, bus size and frequency conversions and (iii) the point-to-point link. The latter implements the physical layer and actually lets physical data exchange between one NoC component to another.

B. Limits of Classical Synchronous Physical Links

The capability of on-chip physical link circuits to cope with the deep submicron technology issues is a key enabling factor for NoCs. Indeed, the physical synchronization issues impose a heavy scalability limitation also to NoC architectures; to fully enable the NoC deployment in the short term future, solutions at the architectural level ready to be industrialized must be provided. Nowadays most of NoC solutions are fully synchronous: all modules have the same clock and the frequency conversion is just allowed between the NoC and the end IP subsystems; frequency conversion is performed inside the NI at the entry/exit point of the network. Also most point-to-point links are synchronous; two connected routers run with the same clock, hence the basic flits of data are exchanged relying on the traditional synchronous assumption (a flit is the elementary data unit a packet is divided into and it is used for transport over the physical link).

The synchronous approach simplifies the design of the NoC link, also thanks to the wide support of CAD tools for synchronous designs, but the distribution of a single clock signal over the entire NoC structure is a big issue. In fact, the NoC is a distributed architecture physically spread over the chip to reduce the wiring length of the links; this is a challenge for clock distribution, especially considering the possible high NoC working frequencies. Therefore in a fully synchronous scenario the limits of the chip performance will be the timing constraints due to clock synchronization, clock tree distribution and power cost of the clock network. A single IP benefits from technology scaling thanks to the reduced size of transistors and length of internal wires. However, each new technology node integrates a greater number of IPs in the same die area thus the delay of global wires (e.g. clock) does not scale and becomes the actual bottleneck for system performance, even using a NOC communication architecture [3]. Keeping the clock skew at minimum takes a considerable time effort, which delays the timing closure process and worsens the area/power ratio due to the required clock buffers. Moreover NoC communication must be reliable and avoid link congestion; at the data-link layer, a hop-by-hop flow control mechanism has to be implemented on top of the synchronous link. Flits must be transferred on the hypothesis that the receiver can accept them, i.e. some protocol must be used to ensure the reliability of the transfer: acknowledged/non-acknowledged (ack/nack), on/off or credit-based are examples of hop-by-hop protocols that can be used in the NoC domain. A minimum amount of buffering is required to hide the link latency of the backward control information (the well-known “stop and go” issue); the amount of buffering depends on the specific protocol adopted.

To overcome the above issues this paper presents a mesochronous physical link architecture with integrated flow control facilities. The proposed architecture also provides support for virtual channels (VCs). VCs improve link
utilization and ease Quality-of-Service (QoS) management. The presented solution is optimized for low circuit complexity and low latency of NoC in deep submicron CMOS.

In the following, Section II introduces the mesochronous approach to synchronization and the Skew Insensitive Mesochronous Link (SIML) architecture. Section III shows an evolution of the SIML architecture, called LIME, which integrates flow control management and a parametric number of VCs; the presented case study deals with 2 virtual channels. Conclusions are drawn in Section IV.

II. THE MESOCHRONOUS SKEW-INSENSITIVE PHYSICAL LINK

A. The Mesochronous Approach

An alternative solution to synchronous links, presented in some NoC architectures, is to build asynchronous links [4],[5]. However, at present, asynchronous communication implies intolerable overhead in terms of area, latency and number of wires. Moreover, designing asynchronous circuits is not trivial as synchronization failure can occur at any time and current CAD tools and design flows do not support it.

Bisynchronous FIFOs are known in literature to deal with dual clock domains; write operations occur at a clock speed different from read ones. To avoid consistency issues (such as reading when FIFO is empty or writing when FIFO is full) empty/full generation circuitry has to be instantiated. This is based on the combination of cost expensive gray coding logic plus brute force synchronizer, which are not able to guarantee 100% robustness against all possible synchronization failures.

A trade-off between synchronous and asynchronous approaches is the mesochronous one. With respect to a fully synchronous paradigm, in a mesochronous scheme the clock signal distributed to the various macrocells is the same, but all the copies of the clock may each have an arbitrary amount of skew. In modern SoCs, this is a consistent scenario since the clock signal is generated from a single source and distributed across the chip floorplan with a space-dependent time-invariant phase offset, which is the clock skew.

B. The Skew-Insensitive Mesochronous Link (SIML)

SIML implements a mechanism that guarantees correct communication between a transmitter module and a receiver module also in case of skewed clock signals driving these modules. In this section, a brief review about its operating principle will be given; more details are available in our previous works [7] and [8].

SIML is composed of two units, SIML TX and SIML RX, respectively shown in Fig. 1 and Fig. 2. During steady-state operation, the tx synch module generates a signal \(strobe\) toggling at half the clock frequency (it is worth noting that at the receiver side the data will be sampled on both rising and falling clock edges and hence the communication bandwidth is the same despite the halved frequency of the tx \(strobe\) signal). The \(strobe\) signal is routed together with transmitted data and used in the Dual Stage Buffer module to synchronize them in the skewed clock domain.

SIML receiver module is composed of a rx synch block and a dual stage buffer. Like the corresponding tx synch block in the SIML transmitter, the rx synch block produces a \(strobe_{rd}\) signal, synchronous with the skewed clock on the receiver side, which toggles at half the clock frequency. The dual stage buffer in Fig. 2 is a 2-stage wagging FIFO: the first stage is sampled by the \(strobe\) signal (coming from the transmitter side), while the second stage is sampled by \(strobe_{rd}\) which is locally generated. If the sampling edges of \(strobe_{rd}\) occur enough time later than the corresponding sampling edges of \(strobe\), metastability is avoided during dual stage buffer operation. Since clock skew do not change during steady-state operation (under the mesochronous hypothesis), making the SIML receiver work without errors is just a matter of a proper initial reset phase. Such reset procedure is described in [8]. Once bootstrap synchronization is correctly accomplished, no other specific action is required during steady-state operation.

The latency introduced by SIML amounts to two clock cycles (due to the cascade of A/A’ and B/B’ registers in Fig. 2) and imposes no bottlenecks on the maximum achievable clock frequency. When synthesizing the SIML architecture on silicon an area cost of 50 \(\mu\)m\(^2\)/bit and a leakage power of 4 nW/bit are achieved. These data refer to synthesis on 65 nm CMOS standard-cells technology in worst case conditions, 125°C and 1.1 V supply voltage, target frequency of 1 GHz (met).

B. Limitations of SIML

A TX/RX SIML pair supports communication in only one direction. If a full-duplex link is needed, two SIML TX/RX pairs must be instantiated, one for each direction. However, a simple communication interface furnished of some flow control mechanism does have signals that travel on both directions. For example, an interface relying on a credit-based protocol has a valid signal that goes from the transmitter...
towards the receiver and a credit signal that goes in the opposite direction. Fig. 3 depicts this example and also shows the instantiation of two SIML links to provide synchronization in the two directions. In the figure an Input Buffer is also represented: buffering is often used in conjunction with flow control to provide efficient link utilization.

To be noted that in the architecture of Fig. 3 the dual stage wagging FIFO has to be sized to account not only for flow control signals (e.g. credit and val) but also for the entire payload bus (flit plus flit_id). The size of the flit plus flit_id bus (flit_id is the identifier used to determine the head/tail of the packet) ranges from tens to hundreds of bits, depending on the application. Moreover the architecture in Fig. 3 does not support the multiplexing of VCs on the link. Therefore a direct application of the SIML leads to a non efficient latency/buffering solution. To overcome the above issues a SIML enhancement (called LIME) with integrated low-latency flow control and VCs management is introduced in Section III.

III. LIME: A LOW-LATENCY MESOCHRONOUS LINK WITH INTEGRATED FLOW CONTROL

As shown in Fig. 4, LIME is applied to a unidirectional link from an upstream (us) module, i.e. the transmitter interface, to a downstream (ds) module, i.e. the receiver interface. Note that to support a unidirectional mesochronous link with flow control, LIME requires two SIML instantiations only for the flow control signals (dl_flow_ctrl_fwd and dl_flow_ctrl_bwd) while the flit plus flit_id enter directly into the input buffers.

LIME consists of a data bus (flit + flit_id) moving from upstream to downstream, while the hop-by-hop flow control uses two signals, dl_flow_ctrl_fwd (data link flow control forward) for transmitting data valid information and dl_flow_ctrl_bwd (data link flow control backward) for receiving the credit or ack/nack or on/off information. There will be a pair of flow control signals, forward + backward, for each additional required virtual channel.

The tx_trigger and rx_trigger signals between the upstream and downstream interfaces are required by the SIML interface to enable the synchronization for any skew condition $\delta$ between the upstream and downstream clocks. Note that the us_clk and ds_clk signals in Fig. 4 are actually the same clock except that they may have an arbitrary amount of skew $\delta$ as far as such skew is bound to 1 clock cycle.

As an example, two virtual channels are implemented in Fig. 4, thus showing the capability of LIME to have an integrated flow control support for a parametric number of virtual channels. A per-flit arbitration module (the “Transmit manager” block in the figure) is in charge of dynamically selecting on a per-flit basis the VC to be activated for transmission.

Inside the receiver interface, two bisynchronous input buffers (“input buffer _0” and “input buffer _1” in Fig. 4) facilitate data transport through the clock domain boundary. The additional signal tx_clk from transmitter to receiver is used to store data in the bisynchronous FIFOs. In both upstream and downstream interfaces proper logic is used to synchronize the flow control signals at the data-link level.

Minimum latency is introduced in the round trip path of the backward control signal, resulting in minimum size for the bisynchronous input buffers. With respect to the synchronous approach, the worst case round trip delay amounts to 2 clock cycles; to absorb this delay the bisynchronous input buffers in LIME have to be sized with two more locations vs. a pure synchronous scenario.

Thanks to the mesochronous synchronization of flow control signals in LIME there aren’t the reliability problems of bisynchronous FIFOs highlighted in Section II.B for synchronous links. As required by SIML to work [8], a few cycles are spent at reset to recover synchronization between the transmitter and receiver clocks. Of course, during this short phase, transmitter and receiver are disabled and the input buffers are disabled too (e.g. the empty signals of the FIFOs are reset); this is required only at start-up and during steady-state operation no synchronization procedures are needed.

In the downstream interface the synchronization of the dl_flow_ctrl_fwd signal versus the ds_clk occurs thanks to the special dual stage flip-flop mechanism driven by the “rx sync,” module, which defacto corresponds to the unit already described in Fig. 2. The latter generates the strobe_rd signal which is synchronous with ds_clk while the strobe signal (alias tx_trigger), synchronous with tx_clk (i.e. us_clk), is generated in the transmitter interface by the “tx sync,” whose architecture is the one in Fig. 1. Both strobe edges are used to sample the dl_flow_ctrl_fwd signal according to the SIML technique analyzed in Section II.B.

The dl_flow_ctrl_fwd signal received from the link, and synchronous with the us_clk, is directly used in the bisynchronous input buffers to enable the writing of a valid flit. On the other side, the synchronized dl_flow_ctrl_fwd signal is used in the “Empty generator” block to correctly control the input buffer empty signal, thus enabling a reliable FIFO read on behalf of the connected IP. The above mechanism avoids the use of classical gray code plus brute force synchronizer.

In the upstream interface, the synchronization of the dl_flow_ctrl_bwd signal versus the us_clk occurs thanks to the special dual stage flip-flop mechanism driven by the “tx sync,” module. The rx_trigger counterpart is the “tx sync,” block.
The architecture is mirrored w.r.t. the one described for the \texttt{dl_flow_ctrl_fwd} signal synchronization. The synchronized \texttt{dl_flow_ctrl_bwd} signal is used in the “Transmitter manager” module to enable/disable the send of the available flits at the output channel stage.

With respect to a classic synchronous link, whose limits have been highlighted in Section I.B, the area and power overheads of LIME are low. Indeed LIME simply requires:

- two more locations for the input buffers to absorb the round trip delay
- two 2-bit SIML structures for the mesochronous synchronization of the flow control signal (resulting in an area penalty of about 200 µm², i.e. roughly 100 logic gates, and a leakage power cost of 8 nW in 65 nm 1.1 V CMOS standard cells technology).

As example LIME has been synthesized in the 65 nm 1.1V CMOS technology in a test circuit connecting two 4-port routers with a flit size of 30, 2 virtual channels and an input buffer size of 2. The extra complexity due to LIME for the added input buffer locations and the added mesochronous flow control synchronization circuitry amounts to 7% (dominated by buffering). With respect to a straightforward extension of the SIML approach in Fig. 3, LIME saves power and area since the mesochronous scheme is applied only to the flow control signals instead of the entire data bus \((\text{flit + flit_id})\) as in Fig. 3.

IV. CONCLUSIONS

Compared to the state-of-the-art this paper has presented a new mesochronous physical link, called LIME, with integrated flow control. LIME improves design productivity and time-to-market allowing a distribution of a single NoC clock without worrying about skew constraints (up to 1 clock cycle). Under the hypothesis of mesochronous clocking this circuit, based on standard logic cells, is able to recover the skew and synchronize the communication. Besides the advantages of the mesochronous approach (such as relaxing back-end clock constraints), LIME also natively supports virtual channels. This new link has low complexity and minimum latency vs. a pure synchronous scenario.

ACKNOWLEDGMENT

This work has been partially supported by the EU 6th FP Project SHAPES (FP6-IST-26825).

REFERENCES