

TSV-Virtualization for Multi-Protocol-Interconnect in 3D-ICs

Felix Miller, Thomas Wild, Andreas Herkersdorf
Institute for Integrated Systems
Technische Universität München
Munich, Germany
Email: {felix.miller, thomas.wild, herkersdorf}@tum.de

Abstract—Through Silicon Vias (TSVs) are the method of choice to realize vertical connections between different chip layers in three dimensional Integrated Circuits (3D-ICs). These TSVs offer a fast connection and due to their short wire length, only a small capacitive load to the driving circuitry. On the other hand TSVs consume a relative large amount of chip area and as TSV-count increases the overall yield generally drops due to TSV manufacturing difficulties. As a result of the low capacitance, TSVs can be clocked much higher than conventional intra-layer links. To fully utilize the TSV-based vertical bandwidth we propose using them in a multiplexed manner and share them between several virtual links. On top of that we propose using TSVs to stretch state-of-the-art interconnects like busses, crossbars or NoCs to other silicon layers in the 3D stack. This reduces TSV count and gives designers the opportunity to easily migrate from 2D to 3D designs and to largely benefit from reuse of existing IP blocks and interconnection schemes.

I. INTRODUCTION

Years ago, when much lower clock frequencies were applied in digital circuits, signals that cross a entire chip within one clock cycle where possible by using global wires [1]. In current SoCs however, with much higher frequencies on the one hand and increased numbers of communicating elements on the other, shared buses that span the whole chip are not feasible anymore. This so-called interconnect bottleneck has led to the emergence of new interconnection schemes like bridged busses [2], ring topologies or Network-on-Chip [3]. These approaches however do not grab the root of the interconnect bottleneck as they do not reduce the wire lengths but allow signals to take multiple clock cycles for reaching their destination.

A promising solution to overcome the interconnect bottleneck is using ICs consisting of several active silicon layers, so called 3D-ICs. If the vertical distance between the layers is short and the connections can be realized as direct vias, punching through the silicon, the average length of global wires is obviously reduced. In 3D-ICs the length of the longest interconnect is reduced by a factor of \sqrt{N} where N is the number of active layers [4].

The costs of TSVs are high in terms of two aspects: (1) Area consumption: Although TSVs are very short connections they have large diameters in the dimension of several (tens) of μm . Wide parallel vertical interconnects built with TSV arrays

therefore consume a significant amount of chip area. (2) Yield drop: With larger number of TSVs in a 3D-IC the probability for that IC to fail dramatically increases. This impact on yield makes 3D-ICs with a high density of TSVs expensive.

With the approach presented in this paper we are pursuing two objectives:

- 1) Reduce TSV count while keeping interconnect performance
- 2) Give designers a technique to ease the transition from 2D to 3D design and allow design reuse

The rest of the paper is organized as follows: In section II we give a general motivation for our approach. Section III shows related work in the field of 3D interconnect. Section IV presents our concept of virtual vertical links, section V gives information on the implementation of building blocks used in our architecture and section VI gives simulation results. Section VII concludes this paper.

II. MOTIVATION

TSVs have diameters in the range of multiples of even the widest intra-layer wires on the upper metal layer. In contrast to global wires, they are very short (in the region of some tens of μm). This allows applying a much higher clock frequency to TSV arrays, compared to long intra-layer wires.

Synchronous global clock distribution networks like three dimensional extension of H-trees are hard to realize and will very unlikely be a suitable solution for chip wide clock distribution in future 3D-ICs [5]. Thus, a fully synchronous 3D-IC is not a realistic scenario. This further stimulates using interconnect schemes with inherent support for clock domain crossings and support a GALS (Globally Asynchronous Locally Synchronous) based design methodology.

As the CPU industry is moving towards multicore and manycore architectures, On-Chip-Networks (NoCs) are applied. Architectures with tens or even hundreds of processing units are on the horizon. First prototypes of such Chip-Multi-Processors (CMPs) like the Intel SCC [6] or the Tiler [7] processors are already available. Such architectures require a high-bandwidth NoC as interconnect backbone. Such a NoC is typically built as mesh structure where each router is connected to one (or more) compute tile(s) and four neighboring routers. 3D-Integration offers large benefits for such systems.

The presented work was partly done under the scope of the NEEDS project which is supported by the German Federal Ministry of Education and Research, funding label 01M3090.

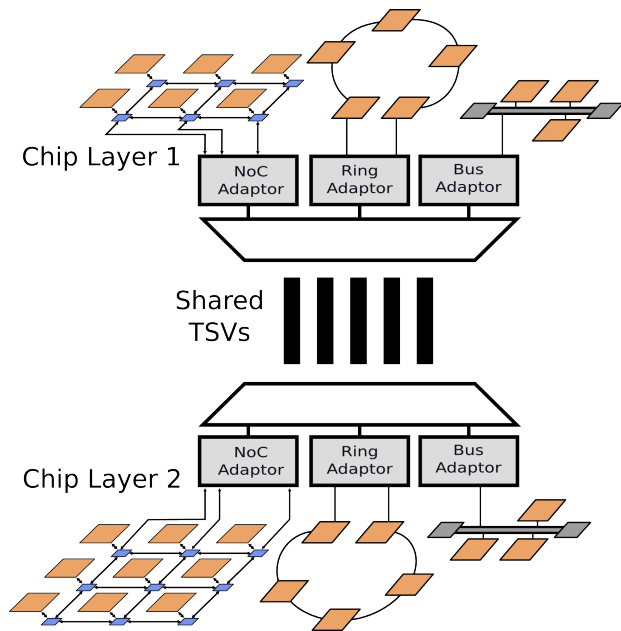


Fig. 1. Vertical continuation of different interconnect topologies

On the one hand, the processing elements are brought closer together (each processing tile has more neighbors resulting in shorter wires) and on the other hand a huge die can be split into several smaller dies. This is beneficial, as multiple smaller dies generally show a higher yield, than one large die of the size of the total area.

To realize a 3D-Mesh-NoC, many TSVs are required, resulting in a large TSV induced area consumption [8]. Besides the large area consumption, also the overall yield drops significantly if many TSVs are used in a design [9]. Our solution to this problem is the aggregation of multiple vertical links and transportation over one shared TSV array, that is clocked with a higher frequency than the intra-layer interconnects.

While such CMPs are very homogenous structures, also heterogeneous 3D systems benefit from our approach. It is expected that most of the upcoming heterogeneous 3D-ICs will not be full new designs from scratch, but rather be a mixture of both conventional and newly developed 3D optimized circuit blocks. Thus it would be a big advantage if several on-chip and off-chip interconnection structures could be reused and integrated into the chip stack. To offer designers a solution for reusing legacy protocols we suggest a flexible architecture that is capable of stretching multiple interconnects realized with different on-chip-interconnect protocols.

To the best of our knowledge, this is the first work, where a concept is shown that allows stretching multiple and different on-chip-interconnects between different silicon layers of a 3D-IC using a common TSV array.

III. RELATED WORK

The idea of using serialization schemes to reduce the number of TSVs used for inter-layer communication has been discussed in several publications [10]–[13].

In [11] the approach of using asynchronous timeless QDI protocols (e.g. 1-of-m signaling) for TSV-based communication is presented. Such asynchronous protocols have the advantage that no global clock distribution is required and that they are tolerant against variances in delay on different TSVs. However, they require at least twice as many TSVs as a conventional synchronous link. Also their specific protocol (use of All-Zero-Wavefronts to keep apart consecutive data words) can slow down the throughput.

In [10] a serialization scheme using a synchronous TSV link is presented. However, only the serialization of one link is covered. There is no multiplexing of multiple links or statistical multiplexing of TSV bandwidth.

In [13] the authors present a serialization scheme based on the ideas of conventional On-Chip-Serialization with shift-register based pulsed transmission.

None of the works listed above covers multiplexing different links on one TSV array.

Our research is closely related to the topic of 3D-Network-On-Chip (3D-NoC) ([14] gives an overview). Several topologies for 3D-NoCs have been proposed that extend conventional 2D NoCs to the third dimension [15]–[17]. This extension is achieved by adding additional ports to 2D intra-layer routers for connecting vertical links. Such structures suffer from high interconnect area due to high number of TSVs and are not optimized for TSV saving. 3D optimized routers, that stretch over more than one layer have been presented in [18], [19]. In [18] a 3D-crossbar is used, and in [19] a stacked router is presented, where the NoC links and nodes themselves are multilayered and therefore such a router requires that the IP-Cores connected to the NoC, are designed in a multilayered fashion as well. Both implementations target to minimize the logic area by avoiding costly 7-port crossbars. However, they still require a large number of vertical interconnects and a rather optimistic TSV induced area consumption is assumed. In [20] the authors propose a scheme where multiple routers share a common TSV array for intra layer NoC links. Access to the TSV array is granted by an arbiter. However, no serialization is applied here and the TSV array is clocked with the same frequency as the intra-layer links.

In [12] a hierarchical NoC-Router using serialization is presented.

The works of Rahmani et al. [21]–[25] focus on power efficient 3D NoC architectures, but also target to reduce TSV count. Here for the vertical interconnects, a TSV array is operated in a bidirectional manner and frequency upscaling is applied to compensate for the throughput loss compared to unidirectional links. In [23]–[25] a NoC architecture using vertical busses for inter-layer communication is proposed.

Although naturally a NoC can offer virtual channels, all these channels are homogenous. We, in contrast, focus on stretching multiple heterogeneous interconnects by using virtual links. These links can have different bandwidth requirements, data widths and run different protocols on different clock frequencies. The goal of our work is to reduce the number of TSVs, and by that improve yield and reduce area consumption.

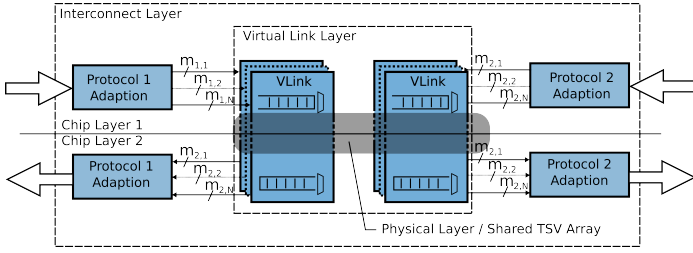


Fig. 2. Abstraction layers in 3D on-chip-interconnect

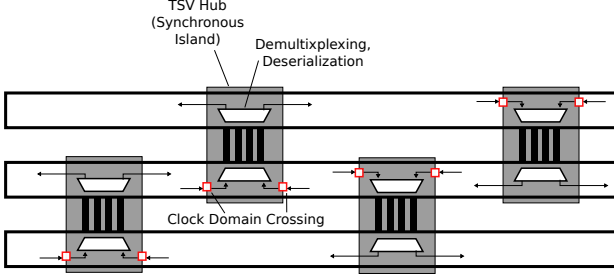


Fig. 3. TSV-Hubs as synchronous islands in a GALS-System

IV. VERTICAL LINK ARCHITECTURE

Several models to divide on-chip communication in SoC designs into multiple levels of abstractions have been proposed in literature [26]. For 3D-ICs the situation is different in the sense, that a new layer can be introduced between two of the lowest layers. With the presence of TSVs it is possible and beneficial to use virtual links for realizing inter-layer interconnects. Multiple virtual links can share a TSV array, by using TDMA and serialization.

To characterize inter-layer communication we propose structuring the communication hierarchy into a stack of three abstraction layers, which we call as follows:

- Physical/TSV Layer
- Virtual Link Layer
- Interconnect Layer

Figure 2 shows the 3 abstraction layer and forms what we call a TSV-Hub. Such a TSV-Hub has the ability of transparently continuing several (possibly different) interconnect protocols to other chip layers. A protocol is adapted on the interconnect layer by a specific adaptor. Such an adaptor in turn, uses one or more generic virtual links (VLinks).

We focus on GALS as a general clocking paradigm for the overall 3D-IC but we operate the TSV-Arrays in a synchronous fashion. In terms of a GALS system a TSV-Hub forms an synchronous island (as shown in Figure 3), spanning two chip layers. TSV arrays can be built in a very regular fashion with little deviation of physical parameters like TSV capacitance. Therefore a TSV array is predestinated for being used as a high-speed synchronous parallel link.

The following paragraphs further explain the different abstraction layers.

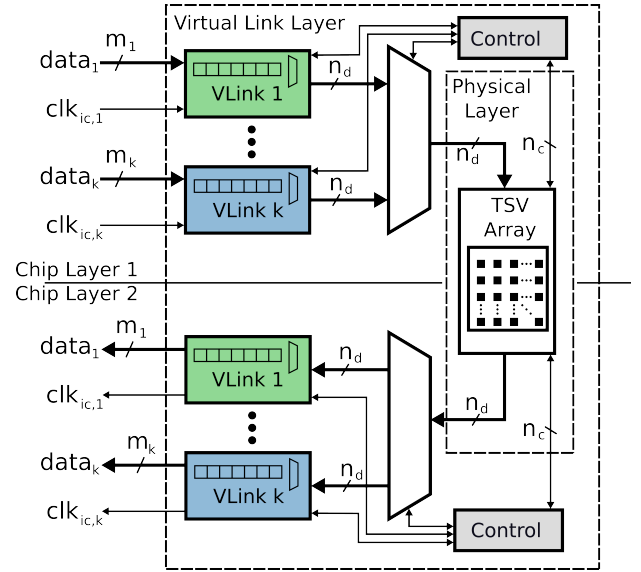


Fig. 4. Vertical Continuation of different interconnect

1) *Physical/TSV Layer*: The physical layer builds the electrical inter-layer connection, realized by an array of TSVs. The parameters that characterize the physical link to the next level of abstraction are the maximum clock frequency f_{tsv} and the number of TSVs in the array (n). Hence, the raw capacity offered by the TSV array equals: $B_{tsv} = f_{tsv} \cdot n$.

The maximum clock frequency can be calculated by means of a lumped RC-based TSV model. In [27] it is shown that a T-shaped lumped RC-model is sufficient in general for calculating the delay and deviates only little from the results, gained by measurements or simulations.

The quantities of the elements in the lumped model can be derived from several physical properties, like TSV-height and -diameter, TSV oxide thickness, TSV material, level of doping of the surrounding substrate and the length of the connecting horizontal wires.

The actual area consumption of a TSV array is determined by several factors: the diameter of the TSVs, the diameter of the TSV landing pads (larger pads reduce alignment difficulties), and the TSV pitch (higher pitch generally results in a higher yield [28]).

An overview on different TSV technologies and their dimensions is provided in [28].

2) *Virtual Link Layer*: To use a physical TSV array in a flexible manner we propose a Virtual Link Layer, where generic virtual links (we call them VLinks) are provided. All these VLinks (within one TSV-Hub) are transported over the same TSV array. The terminations of such VLinks come in different flavors and can be used as building blocks to construct a vertical hub that is tailored to the specific needs of the protocols that are transported, and to the properties of the design, like the clock distribution and level of synchronicity of different synchronous islands.

The VLinks provided by the Virtual Link Layer are trans-

ported over a TSV array with a total number of n TSVs, that can be split in data TSVs (n_d) and control TSVs (n_c) used for signaling and flow control (Figure 4).

The VLink terminations provide 3 functionalities: (1) Serialization/deserialization of the incoming/outgoing data stream to adjust it to the number of TSVs provided by the physical TSV array. (2) Temporal buffering of a specific number of data words, (3) Synchronization between interconnect and TSV clock domain.

A VLink is configurable regarding several aspects: (1) the number of protocol side data bits (m), (2) the number of physical side data bits / number of data TSVs (n_d), (3) the relation of the read and write clocks (synchronous, mesochronous, asynchronous), (4) the quality-of-Service level (guaranteed service or best effort), (5) the buffer depth of the link at both input and output side.

A first analytical view on data serialization for TSV-based interconnects has been presented in [12], where one link is serialized. We extend it for the scenario where multiple links are first serialized and then multiplexed on one TSV array.

The serialization rate S_i for a specific link i equals $S_i = m_i/n_d$.

The relation of the interconnect clock and TSV clock, gives the link speedup $L = f_{TSV}/f_{Link}$

If multiple links are present, that show different data sizes (m_i) and different interconnect frequencies ($f_{ic,i}$) the requirement to keep the link performance is:

$$\sum_{i=0}^{N-1} f_{ic,i} \cdot m_i \geq f_{tsv} \cdot n_d \text{ or } \sum_{i=0}^{N-1} \frac{S_i}{L_i} \geq 1$$

Depending on the expected traffic patterns it can be useful to oversubscribe the TSV array in terms of bandwidth. This leads to a performance degregation if the load is 100% on all links and the inequalities given above are not fulfilled anymore.

The virtual links are multiplexed on one physical link (TSV array) in a TDMA fashion. The used arbiter is modular. Depending on the services that are used, it implements two arbitration schemes:

A pure TDMA arbiter is used, for links requiring a guaranteed bandwidth. Here, timeslots can be assigned directly to a specific VLink to allocate a guaranteed share of the TSV bandwidth. In addition to the fixed timeslots from the TDMA arbiter, dynamic timeslots can be added by running a dynamic TDMA (dTDMA) schedule (based on the ideas of [29] and [30]). With dTDMA, timeslots in the cycle are only used if there is data available in the corresponding queue. In other words, the number of timeslots can grow and shrink from round to round with the number of queues containing data.

The following building blocks are available for the termination of VLinks:

- Asynchronous register with handshake
- Asynchronous FIFO
- Mesochronous register handshake
- Mesochronous FIFO
- Synchronous/Ratiochronous register with or without handshake

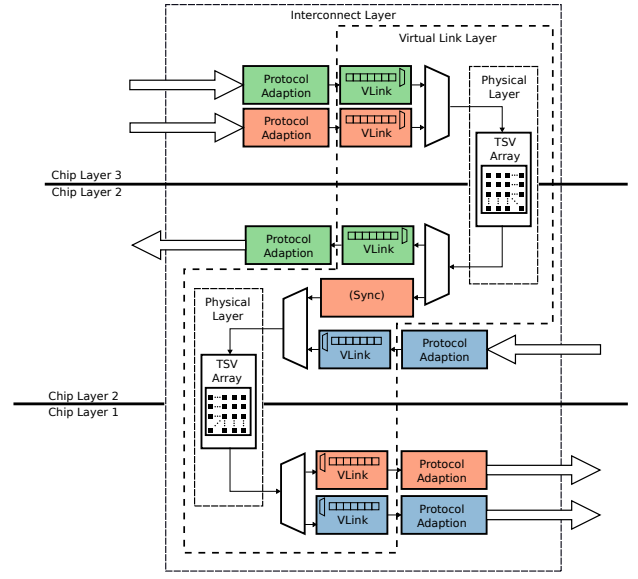


Fig. 5. multiple TSV-Hubs in series

- Synchronous/Ratiochronous FIFO

The asynchronous blocks offer the highest flexibility, but also exhibit the highest cost in terms of area and latency. Read and write clock can be fully independent.

The mesochronous blocks are used if there is a mesochronous relation between the TSV clock and the interconnect clock. That is the case if TSV and interconnect clock share one clock source. Mesochronous synchronizers can be realized with less chip area and show a smaller delay. The synchronizers are built according to [5].

If a link is established not to a direct neighbor but to a chip layer farther away in the chip stack, multiple TSV-Hubs can be traversed in series (as depicted in Figure 5). If a link is not leaving at a certain chip layer, the Virtual Link Layer is not left in this chip layer and no VLink termination is required. Only a synchronizer is needed to allow clock domain crossing between different TSV clocks.

3) *Interconnect Layer*: The Interconnect Layer forms the adaption to one or more interconnect protocols like busses, crossbar interfaces or NoC links. Depending on the protocol requirements, such adaptors can be more or less complex. For bus protocols where very strict inter-cycle dependencies exist, such a adaptor is rather complex requiring state machines on both link ends. For interconnect protocols that follow a "Fire-and-Forget" approach, the adaptor can be of very low complexity and may boil down to an adaption of handshake signaling.

Each protocol adaptor can make use of one or more VLinks. For a NoC interface, the adaptor generally requires only one VLink, as long as no virtual channels are used. If virtual channels are used, a VLink is required for each virtual channel. Dedicated VLinks for virtual channels can be omitted, if the NoC is designed such that the destination can deassert its ready signal early (almost full), at a time where at least as many words fit in the destination's input buffer as the maximal

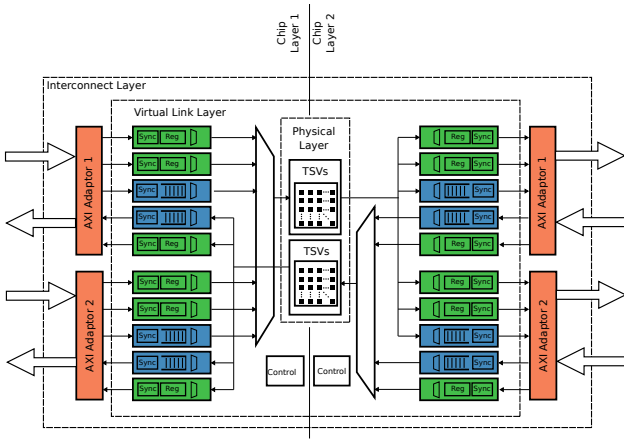


Fig. 6. Continuation of two AXI links

latency of the TSV-Hub (in cycles).

V. IMPLEMENTATION

An implementation of the building blocks described above on RTL level has been realized. The RTL descriptions have been synthesized for a 65 nm and a 40 nm standard cell CMOS process.

To model the TSVs, parameters of the TSV process described in [28] are used.

The TSV-Hubs require a high speed clock, however the clock skew between different TSV-Hubs is nonrelevant. Therefore, the clock network does not have to be a sophisticated H-tree, or even a “3D-H-tree”. Another option is generating the clock internally by means of a PLL circuit. Within one TSV-Hub, the TSV clock is delivered through a dedicated clock TSV to the other chip layer. Our simulations show, that for the chosen standard cell process, and with the physical TSV parameters from [28] a TSV clock of $f_{tsv} = 2$ GHz is feasible.

To demonstrate our concept, we composed a TSV-Hub capable of transporting two independent AXI interconnects as it is shown in figure 6. The two AXI links are fully independent. Of course, it would also be possible to continue different protocols by using appropriate adaptors. We decided however, for the sake of clarity, two use only links of one protocol in this case study.

The AMBA AXI protocol is a well-known and widely used protocol for on-chip-interconnects [31]. The AXI specification, however, only defines the link interface and handshake protocols and not the actual architecture of an interconnect fabric. An AXI link is composed of five independent channels (Write Address, Write Data, Read Address, Read Data and Write Response), where each channel is unidirectional except for one wire used as flow control (ready signal). The size of the data word within the data channels is configurable, we use 64 and 32 bit wide data signals in our implementation. The address size is 32 bit. Besides the actual address or data information each AXI channel transports further control information and handshake signals.

TABLE I
WIDTHS OF AXI CHANNELS IN BITS

Channel	32 Bit	64 Bit
Read Address (W_{raddr})	49	49
Write Address (W_{waddr})	55	55
Write Data (W_{wdata})	42	74
Read Data (W_{rdata})	40	72
Write Response (W_{wresp})	9	9
Sum	195	259

Table I gives an overview on the AXI channel sizes for a 32 and a 64 bit implementation. To establish an AXI-Link to another chip layer without use of serialization and multiplexing on the TSVs, 195 or 259 TSVs would be necessary for a 32 bit or 64 bit implementation, respectively.

The actual bandwidth requirement for a AXI link can be calculated as follows for the downstream/write (master to slave) direction:

$$B_{axi,down} = f_{ic} \left(W_{wdata} + \sum_{i=1}^{N_w} \frac{W_{waddr}}{i} r_{w,i} + \sum_{i=1}^N \frac{W_{raddr}}{i} r_{r,i} \right)$$

and as follows for the upstream/read (slave to master) direction:

$$B_{axi,up} = f_{ic} \left(W_{rdata} + \sum_{i=1}^{N_r} \frac{W_{wresp}}{i} \cdot r_{w,i} \right)$$

With the interconnect freequency f_{ic} and the maximum occuring burst sizes N_w and N_r . $r_{w,i}$ being the probability that a write burst is of length i , and $r_{r,i}$ being the probability that a read burst is of length i .

If the TSV count is reduced, such that the bandwidth offered by the TSV array B_{tsv} is smaller than the bandwidth that is required for transporting an AXI interconnect, the performance degrades linearly as the TSV count decreases.

This is shown in the figures 8(a) to 8(d) as dashed lines for different burst sizes and single word transfers. However to reach the ideal (dashed) line in an implementation it would be required to realize non-integer serialization rates (eg. 3.22 to map an AXI read address channel to an array of 23 TSVs). Especially if there is no common divisor in m (width of the VLink) and n (number of data TSVs), a full $m \times n$ crossbar is required at the serializer. Such a crossbar however consumes a large chip area, that generally cannot be justified by the further TSV savings. If such a crossbar is not present, a multiplexer maps $\lceil m/n_d \rceil$ different portions of the m -wide-word to the data TSVs. The required bandwidth of the VLink demands from the TSV array is then raised to:

$$B_{i,eff} = f_{ic} \cdot n_d \cdot \lceil S_i \rceil$$

The interconnect layer of our demonstrating implementation is formed by 4 AXI-Adaptors to support two independent AXI interconnects as depicted in figure 6.

Each AXI-Adaptor connects to 5 VLinks. This enables an independent flow in each of the AXI channels of one AXI interconnect.

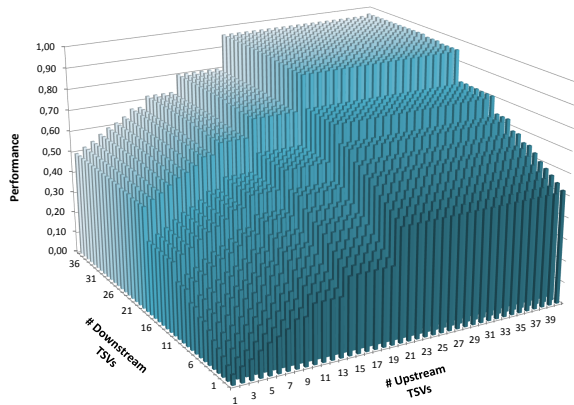


Fig. 7. Performance for different numbers of up- and downstream TSVs

Due to the mesochronous clock distribution, the VLinks of the mesochronous type are applied. To allow a continuous data flow it is clear, that FIFOs have to be used for the data channels. The depth of the FIFOs can be kept small. To prevent data loss it has to be ensured, that the round trip delay through the mesochronous synchronizer (actual data and flow control in reverse direction) is covered. We use 4 word deep FIFOs in our implementation.

For the VLinks used by the address channels we only use mesochronous registers with handshaking. The utilization of the address channels crucially depends on the length of the burst transfers that occur in the present traffic pattern. For long bursts the address channels are idle most of the time and a slight delay on these channels due to the handshaking process has only little influence on the overall performance. As a result of the pipelined nature of the AXI protocol, such delays are completely hidden for consecutive burst transfers. For single word transfers, however, where each data word is accompanied by a corresponding address on the address channel, the performance significantly degrades if delays occur on the address channels.

We target an interconnect clock of $f_{ic} = 400$ MHz and a TSV clock of $f_{tsv} = 1.6$ GHz. This means, that with a serialization rate of $S = f_{tsv}/f_{ic} = 4$ and a reduction of the TSV count by 4 compared to the number of wires needed for the intra-layer-link, the TSV link can keep pace with the bandwidth requirement of the VLinks. However, the occurrence of burst transfers favors oversubscribing the TSV link, as the actual bandwidth requirement of the address channel is much smaller than its pure word size suggests.

At the physical layer, the downstream TSV-Array is built out of 21 data TSVs and the upstream array contains 20 data TSVs. These two numbers form a pareto-optimal point as shown in section VI.

VI. RESULTS

To evaluate the performance of the TSV-Hub we carried out simulations for 4 different architectural compositions (one and two AXI-links, 32 and 64 bit data width), different burst sizes

TABLE II
AREA CONSUMPTION OF THE MESOCHRONOUS FIFOs IN μm^2

Channel	65 nm	40 nm
Read Data	3473	1778
Write Data	3570	1828

TABLE III
AREA CONSUMPTION OF THE MESOCHRONOUS SYNCHRONIZERS IN μm^2

Channel	65 nm	40 nm
Read Address	1528	852
Write Address	1711	964
Write Response	490	250

TABLE IV
AREA CONSUMPTION OF COMPLETE TSV-HUB (IN μm^2)

	Logic Area		TSV area (pitch 20 μm)	TSV + Logic Area		Reference Area
	65 nm	40 nm		65 nm	40 nm	
1x 32 bit	21876	11548	14000	35876	25548	78800
2x 32 bit	40662	21489	23200	63862	44689	156000
1x 64 bit	34369	18050	18800	53169	36850	103600
2x 64 bit	64383	34002	36000	100383	70002	207200

TABLE V
PERCENTAGE OF ORIGINAL TSV INDUCED AREA

	65 nm	40 nm
1x 32 bit	45 %	32 %
2x 32 bit	41 %	29 %
1x 64 bit	51 %	36 %
2x 64 bit	48 %	34 %

and different TSV counts. The simulations were performed on RTL level, however realistic clock frequencies for both TSVs and conventional logic have been chosen. We compared the achieved throughput with the throughput of a conventional AXI interconnect that is not continued over a TSV-hub and normalized to this reference throughput.

The figures 8(a) to 8(d) show the results of these simulations. The dashed lines represent the throughput that is achievable in case of a perfect serializer. A perfect serializer means that any serialization rate is possible. In case of a non-perfect serializer, for a non-integer serialization rate, the serializer generates for each m sized word $\lceil m/n \rceil$ n -sized words, where the last word is not completely filled up with information. The simulation results are shown by the solid lines in figure 8.

If we take the the case for the 64 bit wide double link and 32 word long burst transfers, we see that with 37 TSVs we still achieve 97% throughput, to achieve 98% at least 55 TSVs would be necessary and 74 TSVs for 100%.

Figure 7 shows the mean value of up- and downstream throughput for different numbers of upstream and downstream TSVs (two AXI links / 32 bit). Here the point with 21 downstream and 20 upstream TSVs results in a throughput of 95%. For 100% throughput 40 upstream and 42 downstream TSVs would be required.

These TSV counts show pareto optimal points, where we

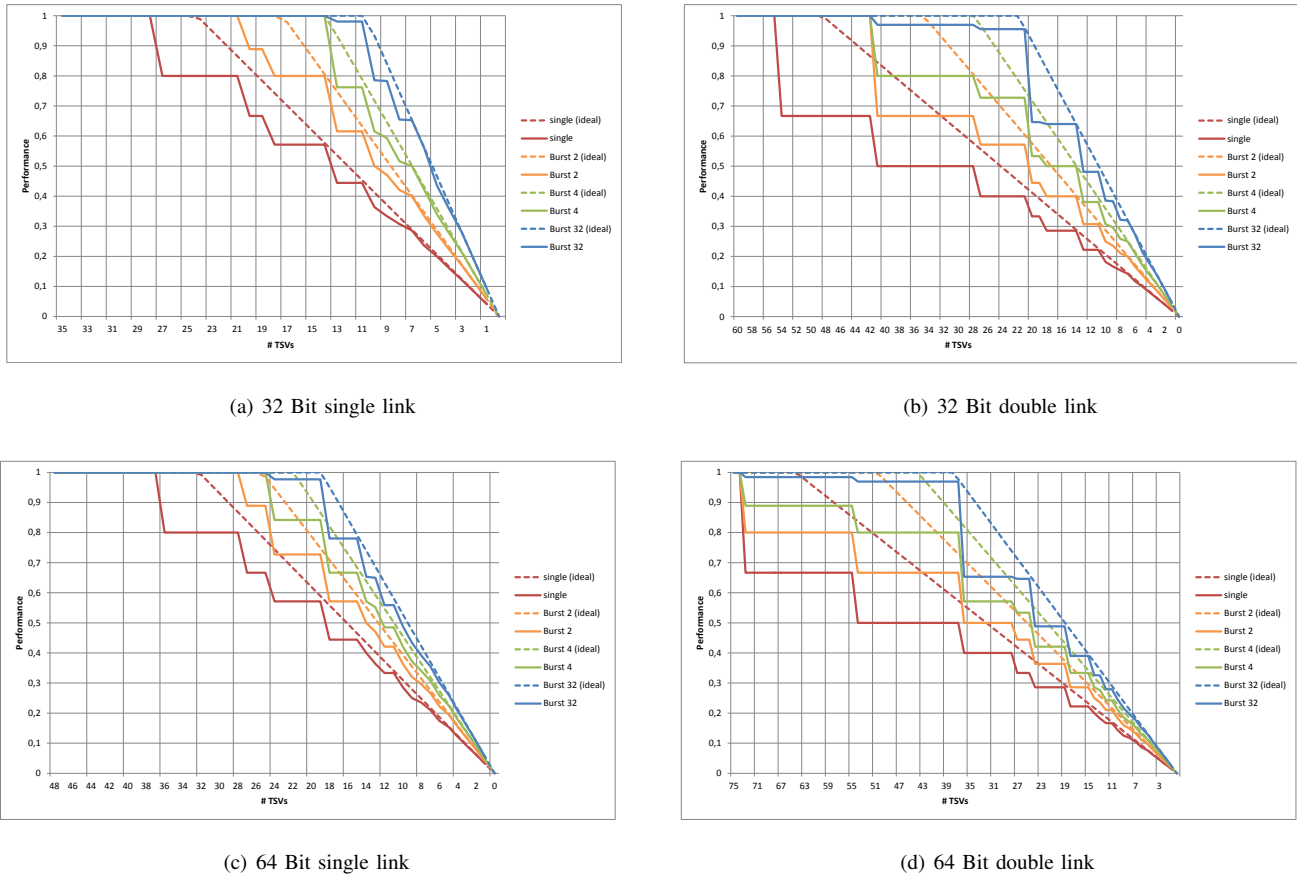


Fig. 8. Performance Degradation for downstream axi link

accept a certain (but small) loss of performance, but have a good utilization of the present TSVs.

Table IV shows the area occupation of both logic and TSV induced area for the different scenarios. The total area is compared to a reference area that is required if no multiplexing and serialization is applied. In a design containing only two 32-bit AXI links that span more than one active layer, around 390 TSVs are necessary if each wire has its own dedicated TSV. If a TSV pitch of $20\ \mu\text{m}$ is assumed, such an array would consume $0.156\ \text{mm}^2$ of chip area. With serialization and multiplexing we can reduce the number of required TSVs to 47 resulting in a TSV area of only $0.023\ \text{mm}^2$. With the additional area needed for serializing and multiplexing ($0.041\ \text{mm}^2$) we achieve a total area of $0.064\ \text{mm}^2$ (41% of the original TSV area). For a 40 nm process, the same calculation gives 29% of the original area. The saved area is just one benefit. In addition to that, the TSV introduced yield drop is significantly lower as the number of TSVs has been reduced by 80%.

Obviously with smaller TSV pitches, the area saving achieved decreases. For the the implementations shown in IV, the critical value is around $7\ \mu\text{m}$ for 65 nm process and $5\ \mu\text{m}$ for the 40 nm process. For smaller TSV pitches no area saving is achieved by serialization/multiplexing. However, the TSV

introduced yield drop is still reduced.

Naturally also the feature size of the integration process for conventional logic influence the achieved area saving. Smaller feature sizes reduce the logic overhead caused by multiplexing and serialization and thus support our concept.

If we look at current trends, technology scaling is still continuing as we see by the advent of the 22 nm process technologies. We assumed a TSV pitch of $20\ \mu\text{m}$ in our analysis, as this is a realistic number for current prototype 3D-ICs. The first commercially, by foundry services available TSVs processes, however, will most likely have a much higher TSV pitch in order to achieve a acceptable yield. This trend can be seen for example at programs like CEA-Leti's "Open 3D" initiative [32], where a minimum TSV pitch of $80\ \mu\text{m}$ will be available. This results in a much higher TSV induced areas, as the TSVs area scales quadratically with the TSV pitch.

VII. CONCLUSION

We presented a new, protocol aware, concept of better utilizing TSV's transfer capacity. The concept makes use of virtual vertical links that share a common TSV array. By using a higher clock speed for TSV arrays and serializing the data stream of individual virtual links, a reduction of TSV count is achieved. If the load on the virtual links is not always

at 100 % a further reduction of TSV count is achieved by oversubscription and multiplexing of several virtual links onto the same TSV array.

We presented a case study for a vertical continuation of two independent AXI interconnects, that share a common TSV array and showed that TSV induced area can be reduced such that, the area occupied by the logic overhead for serialization and multiplexing, together with the actual TSV area is only 29 to 51 % of the original TSV area (when serialization and multiplexing is not applied). One promising application scenario for our concept is to combine multiple vertical links in a 3D mesh NoC. But also for more heterogenous topologies the concept is beneficial. Not all upcoming 3D-SoCs will require a fully-fledged regular NoC (like a stacked mesh). Heterogenous 3D-SoCs will make us of different application specific point to point protocols to connect to on-chip-sensors, hardware accelerators and peripherie. We showed in this paper, that it is possible to stretch such protocols over multiple layers in 3D-ICs with economical use of TSVs.

REFERENCES

- [1] R. Ho, K. Mai, and M. Horowitz, "The future of wires," *Proceedings of the IEEE*, vol. 89, no. 4, pp. 490–504, apr 2001.
- [2] Y. Zhang, W. Ye, and M. Irwin, "An alternative architecture for on-chip global interconnect: segmented bus power modeling," in *Signals, Systems Computers, 1998. Conference Record of the Thirty-Second Asilomar Conference on*, vol. 2, nov 1998, pp. 1062–1065 vol.2.
- [3] L. Benini and G. De Micheli, "Networks on chips: a new soc paradigm," *Computer*, vol. 35, no. 1, pp. 70–78, jan 2002.
- [4] V. Pavlidis and E. Friedman, "Interconnect-based design methodologies for three-dimensional integrated circuits," *Proceedings of the IEEE*, vol. 97, no. 1, pp. 123–140, jan. 2009.
- [5] I. Loi, F. Angiolini, and L. Benini, "Developing mesochronous synchronizers to enable 3d nocs," in *Proceedings of the conference on Design, automation and test in Europe*, ser. DATE '08. New York, NY, USA: ACM, 2008, pp. 1414–1419. [Online]. Available: <http://doi.acm.org/10.1145/1403375.1403717>
- [6] S. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, A. Singh, T. Jacob, S. Jain, V. Erraguntla, C. Roberts, Y. Hoskote, N. Borkar, and S. Borkar, "An 80-tile sub-100-w teraflops processor in 65-nm cmos," vol. 43, no. 1, jan. 2008, pp. 29–41.
- [7] D. Wentzlaff, P. Griffin, H. Hoffmann, L. Bao, B. Edwards, C. Ramey, M. Mattina, C.-C. Miao, J. Brown, and A. Agarwal, "On-chip interconnection architecture of the tile processor," *Micro, IEEE*, vol. 27, no. 5, pp. 15–31, sept.-oct. 2007.
- [8] P. Vivet, D. Dutoit, Y. Thonnart, and F. Clermidy, "3d noc using through silicon via: An asynchronous implementation," in *VLSI and System-on-Chip (VLSI-SoC), 2011 IEEE/IFIP 19th International Conference on*, oct. 2011, pp. 232–237.
- [9] I. Loi, S. Mitra, T. H. Lee, S. Fujita, and L. Benini, "A low-overhead fault tolerance scheme for tsv-based 3d network on chip links," in *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design*, ser. ICCAD '08. Piscataway, NJ, USA: IEEE Press, 2008, pp. 598–602. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1509456.1509589>
- [10] F. Sun, A. Cevrero, P. Athanasopoulos, and Y. Leblebici, "Design and feasibility of multi-gb/s quasi-serial vertical interconnects based on tsvs for 3d ics," in *VLSI System on Chip Conference (VLSI-SoC), 2010 18th IEEE/IFIP*, sept. 2010, pp. 149–154.
- [11] V. Darve, "An asynchronous serial link for a 3d network-on-chip," in *Friday Workshop on 3D Integration - Date Conference 2010*, 2010.
- [12] F. Darve, A. Sheibanyrad, P. Vivet, and F. Petrot, "Physical implementation of an asynchronous 3d-noc router using serial vertical links," in *VLSI (ISVLSI), 2011 IEEE Computer Society Annual Symposium on*, july 2011, pp. 25–30.
- [13] S. Pasricha, "Exploring serial vertical interconnects for 3d ics," in *Design Automation Conference, 2009. DAC '09. 46th ACM/IEEE*, july 2009, pp. 581–586.
- [14] A.-M. Rahmani, K. Latif, P. Liljeberg, J. Plosila, and H. Tenhunen, "Research and practices on 3d networks-on-chip architectures," in *NORCHIP, 2010*, nov. 2010, pp. 1–6.
- [15] V. Pavlidis and E. Friedman, "3-d topologies for networks-on-chip," in *SOC Conference, 2006 IEEE International*, Sept. 2006, pp. 285–288.
- [16] B. Feero and P. P. Pande, "Performance evaluation for three-dimensional networks-on-chip," in *ISVLSI '07: Proceedings of the IEEE Computer Society Annual Symposium on VLSI*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 305–310.
- [17] B. S. Feero and P. P. Pande, "Networks-on-chip in a three-dimensional environment: A performance evaluation," *IEEE Trans. Comput.*, vol. 58, no. 1, pp. 32–45, 2009.
- [18] J. Kim, C. Nicopoulos, D. Park, R. Das, Y. Xie, V. Narayanan, M. S. Yousif, and C. R. Das, "A novel dimensionally-decomposed router for on-chip communication in 3d architectures," *SIGARCH Comput. Archit. News*, vol. 35, pp. 138–149, 2007.
- [19] D. Park, S. Eachempati, R. Das, A. Mishra, Y. Xie, N. Vijaykrishnan, and C. Das, "Mira: A multi-layered on-chip interconnect router architecture," in *Computer Architecture, 2008. ISCA '08. 35th International Symposium on*, June 2008, pp. 251–261.
- [20] C. Liu, L. Zhang, Y. Han, and X. Li, "Vertical interconnects squeezing in symmetric 3d mesh network-on-chip," in *Design Automation Conference (ASP-DAC), 2011 16th Asia and South Pacific*, jan. 2011, pp. 357–362.
- [21] A. Rahmani, P. Liljeberg, J. Plosila, and H. Tenhunen, "Exploring a low-cost inter-layer communication scheme for 3d networks-on-chip," in *Computer Architecture and Digital Systems (CADS), 2010 15th CSI International Symposium on*, sept. 2010, pp. 163–166.
- [22] A.-M. Rahmani, K. Vaddina, P. Liljeberg, J. Plosila, and H. Tenhunen, "Power and area optimization of 3d networks-on-chip using smart and efficient vertical channels," in *Integrated Circuit and System Design. Power and Timing Modeling, Optimization, and Simulation*, ser. Lecture Notes in Computer Science, J. Ayala, B. Garcia-Camara, M. Prieto, M. Ruggiero, and G. Sicard, Eds. Springer Berlin / Heidelberg, 2011, vol. 6951, pp. 278–287.
- [23] A.-M. Rahmani, P. Liljeberg, J. Plosila, and H. Tenhunen, "Lastz: An ultra optimized 3d networks-on-chip architecture," in *Digital System Design (DSD), 2011 14th Euromicro Conference on*, 31 2011-sept. 2 2011, pp. 173–180.
- [24] A.-M. Rahmani, K. Latif, K. Vaddina, P. Liljeberg, J. Plosila, and H. Tenhunen, "Congestion aware, fault tolerant, and thermally efficient inter-layer communication scheme for hybrid noc-bus 3d architectures," in *Networks on Chip (NoCS), 2011 Fifth IEEE/ACM International Symposium on*, may 2011, pp. 65–72.
- [25] A.-M. Rahmani, K. Latif, P. Liljeberg, J. Plosila, and H. Tenhunen, "A stacked mesh 3d noc architecture enabling congestion-aware and reliable inter-layer communication," in *Parallel, Distributed and Network-Based Processing (PDP), 2011 19th Euromicro International Conference on*, feb. 2011, pp. 423–430.
- [26] S. Pasricha and N. Dutt, *On-chip communication architectures : system on chip interconnect*. Amsterdam; Boston: Elsevier / Morgan Kaufmann Publishers, 2008.
- [27] G. Katti, M. Stucchi, K. De Meyer, and W. Dehaene, "Electrical modeling and characterization of through silicon via for three-dimensional ics," *Electron Devices, IEEE Transactions on*, vol. 57, no. 1, pp. 256–262, jan. 2010.
- [28] G. Van der Plas, P. Limaye, I. Loi, A. Mercha, H. Oprins, C. Torregiani, S. Thijs, D. Linten, M. Stucchi, G. Katti, D. Velenis, V. Cherman, B. Vandeveld, V. Simons, I. De Wolf, R. Labie, D. Perry, S. Bronckers, N. Minas, M. Cupac, W. Ruythooren, J. Van Olmen, A. Phommahaxay, M. de Potter de ten Broeck, A. Opdebeeck, M. Rakowski, B. De Wachter, M. Dehan, M. Nelis, R. Agarwal, A. Pullini, F. Angiolini, L. Benini, W. Dehaene, Y. Travalay, E. Beyne, and P. Marchal, "Design issues and considerations for low-cost 3-d tsv ic technology," *Solid-State Circuits, IEEE Journal of*, vol. 46, no. 1, pp. 293–307, jan. 2011.
- [29] T. D. Richardson, C. Nicopoulos, D. Park, V. Narayanan, Y. Xie, C. Das, and V. Degalalal, "A hybrid soc interconnect with dynamic tdma-based transaction-less buses and on-chip networks," in *Proceedings of the 19th International Conference on VLSI Design held jointly with 5th International Conference on Embedded Systems Design*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 657–664. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1112244.1112670>
- [30] F. Li, C. Nicopoulos, T. Richardson, Y. Xie, V. Narayanan, and M. Kandemir, "Design and management of 3d chip multiprocessors using network-in-memory," *SIGARCH Comput. Archit. News*, vol. 34, no. 2, pp. 130–141, 2006.
- [31] ARM Ltd. AXI specification. Available at www.arm.com.
- [32] CEA-LETI. Open 3D Initiative. [Http://www-leti.cea.fr](http://www-leti.cea.fr).