A Scalable Network-on-Chip Microprocessor With 2.5D Integrated Memory and Accelerator

Sai Manoj P. D., *Member, IEEE*, Jie Lin, Shikai Zhu, *Student Member, IEEE*, Yingying Yin, Xu Liu, Xiwei Huang, Chongshen Song, Wenqi Zhang, Mei Yan, Zhiyi Yu, *Senior Member, IEEE*, and Hao Yu, *Senior Member, IEEE*

***Abstract*— This paper presents a 2.5D integrated microprocessor die, memory die, and accelerator die with 2.5D silicon interposer I/Os. The use of such 2.5D silicon interposer I/Os provide a scalable interconnection for core-core (up to 32 cores), core-memory (4× storage capacity) and core-accelerator (4.4× speedup in H.264 decoder). The 2.5D integrated chip was implemented in GF 65 nm process with multicore microprocessor operated at 500 MHz under 1.2 V supply with 1.08 W power dissipation. A pair of 8 Gbps 2.5D silicon interposer I/O is designed for each of 12 inter-die communication channels, achieving a bandwidth of 24 GBps with 7.5 pJ/bit energy efficiency. As a result, the specified applications such as H.264 video data analytics and AES encryption can achieve significant performance improvement of throughput and energy efficiency.**

***Index Terms*— 25D stacking, high bandwidth, multicore microprocessor, SerDes, silicon interposer.**

I. Introduction

With the advance technology scaling and system integration, a large number of processing cores can be progressively integrated for high throughput, but is difficult to maintain a low power density [1]–[9]. Heterogeneous multicore microprocessors are normally integrated with dedicated accelerators for application specified computing [10]–[12]. With further integration of memory, one can develop energy-efficient computing platform to support future data-oriented analytics for servers and also edge devices. As such, it becomes an emerging need to explore a novel multicore architecture with the advantage of scalability and configurability for the heterogeneous multicores integrated with accelerators and memories.

The traditional 2D integration suffers from low efficiency, delay, high power and also area as the I/O interconnect between logic and memory started to dominate the system performance [13], [14]. The 3D integration [15]–[18], performed by vertical stacking of dies using through-silicon vias (TSVs), enjoys the benefit of high efficiency of logic-memory integration. However, thermal dissipation becomes the major concern in the 3D integration [15], [17], [19]. On the other hand, 2.5D integration [18], [20]–[25], in which multiple dies are placed on one common substrate and are connected using silicon interposers, are designed as medium-distance transmission lines (T-lines) for logic and memory integration. A 2.5D Silicon interposer is used to connect inter-die: cores to cores, cores to accelerators and cores to memories in this work. As shown in Fig. 6, 2.5D silicon interposer comprises of through-silicon vias (TSVs) and a metal trace in substrate. Though the integration density of 2.5D ICs is lower compared to 3D ICs, the thermal dissipation concerns in 2.5D integration are much alleviated [26]. Many recent works [14], [27], [28] have shown the advantage to deploy 2.5D integration for future server and edge devices.

As explored in our preliminary result [1], the 2.5D integration greatly suits for exploring heterogeneous multicore network-on-chip architecture: multicore microprocessor die with 8 MIPS cores, memory die for data capacity expansion, and accelerator die for faster data such as H.264 video decoder. In this paper, we further fabricate the through silicon interposer as in [29] with all dies integrated. The entire chip was implemented in GF 65 nm process. The multicore microprocessor operates at 500 MHz under 1.2 V supply with 1.08 W power dissipation. The 2.5D silicon interposer based I/Os support 12-way full-duplex communication in parallel, bringing the bandwidth up to 24 GB/s with 7.5 pJ/bit energy efficiency. A functional level system partitioning strategy is followed here, targeting scalable future data-oriented computing systems. A functional level partitioning in such 2.5D or 3D systems yield significant benefits without large partitioning overhead.

The contributions of this work can be summarized as:

- This is the first demonstration from academia of 2.5D silicon interposer integrated multicore microprocessor with memory and accelerator, which has the benefits of system integration flexibility and energy efficiency.

Manuscript received June 29, 2016; revised November 16, 2016 and December 18, 2016; accepted December 19, 2016. Date of publication January 16, 2017; date of current version May 25, 2017. This work was supported in part by grant from Singapore MOE Tier-2 fund MOE2015-T2-2-013, MOE2010-T2-037, A*STAR PSF fund (11201202015), and support from Intel Lab and Samsung Corporation. This brief was recommended by Associate Editor F. J. Kurdahi.

S. M. P. D. is with the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore and also with the Institute of Computer Technology, TU Wien 1040, Austria.

J. Lin, S. Zhu, and Y. Yin are with the State Key Laboratory of ASIC & System, Fudan University, Shanghai 201203, China.

X. Liu, X. Huang, M. Yan, and H. Yu are with the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore 639798 (e-mail: haoyu@ntu.edu.sg).

C. Song and W. Zhang are with the National Center for Advanced Packaging Co. Ltd. (NCAP), Wuxi, China, and also with the Institute of Microelectronics of Chinese Academy of Sciences (IMECAS), Beijing, China.

Z. Yu is with Fudan University, and also with the SYSU-CMU Joint Institute of Engineering, School of Electronics and Information Technology, Sun Yat-sen University, China and also with SYSU-CMU Shunde International Joint Research Institute, Guangdong 510275, China (e-mail: yuzhiyi@mail.sysu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSI.2016.2647322

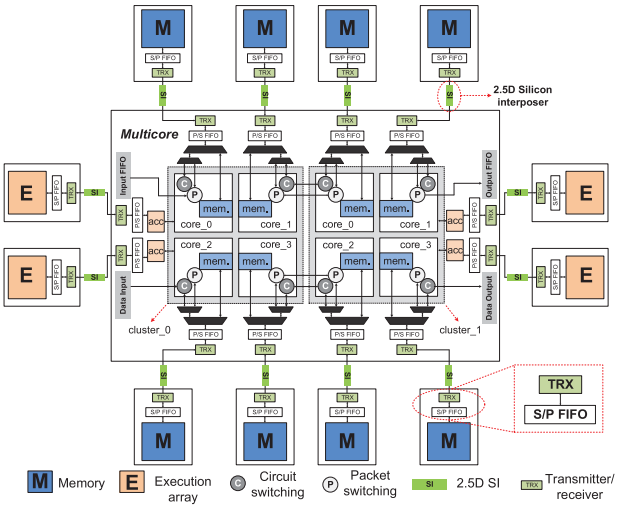


Fig. 1. 2.5D NoC architecture of heterogeneously integrated multicore, memory (SRAM), and accelerator by 2.5D silicon interposer I/Os.

- A new inter-die communication circuit is demonstrated with 2.5D silicon interposer I/O circuit as well as DMAs, presented in Sections IV and V.
- A new four level memory hierarchy is successfully implemented: 1) register files, 2) intra-cluster on-die memory, 3) intra-cluster off-die memory, and 4) inter-cluster memory, given in Section III.

Note that the 2.5D I/O design here is for "middle-distance" communication between cores, accelerator and memory in range $<$ cm, which we believe is the scale for future core, accelerator and memory integration. Compared to a "short-distance" communication at die level in range $<$mm, the 2.5D integration can integrate more components with better thermal dissipation compared to 3D integration. Moreover, compared to "long-distance" communication at PCB level in range $>$ few cm with SerDes I/O, the power efficiency of 2.5D silicon interposer based SerDes I/O would be much better due to the significantly smaller insertion loss, such that the data recovery circuit does not need over-designed with additional power.

The remaining of this paper is organized as follows. System architecture and the components of the system are described in Section II. Section III presents the extended memory architecture and the management. The interfaces between blocks are presented in Section IV and corresponding communication protocols are described in Section V. Mapping of applications to the hardware is explained in Section VI. Results are presented in Section VII with conclusion in Section VIII.

II. System Architecture and On-Chip Components

An overview of the proposed 2.5D multicore microprocessor system architecture is shown in Fig. 1. The system comprises of an 8-core MIPS microprocessor, works as the basic system, memory (M) and execution array (E). Inherited from our previous work in [4], the cores communicate through a high bandwidth and power efficient 2D mesh network-on-chip (NoC), which features packet-controlled circuit-switched double-layer routing. For the sake of data locality, the cores are further divided into 2 clusters, either of which has an individual

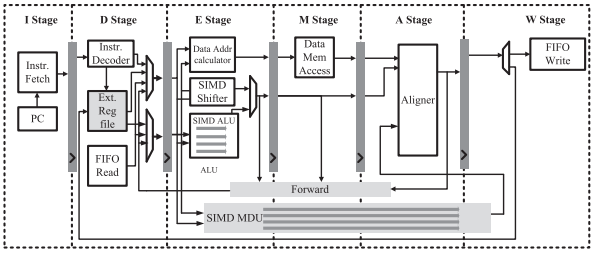


Fig. 2. MIPS core architecture with pipeline.

clock domain with fine-grained clock-gating. Each MIPS core is designed with private instruction memory (6 KB) and shared on-chip data memory with a small capacity of 4 KB. 32-bit input FIFOs and output FIFOs are the interfaces between primary I/Os and the microprocessors. They are dual-clock FIFOs supporting arbitrary different clock rates between FIFO read and FIFO write. The clock rate at the microprocessor's side is same as the microprocessor (about 500 MHz). Besides the input and output FIFOs, the microprocessor also has "P/S FIFOs" which is the interface between microprocessors and 2.5D I/Os. The 2.5D silicon interposer channel is indicated as '2.5D SI' in Fig. 1.

To meet the requirement of storage capacity in data intensive applications, the system is integrated with 8 identical off-die memory blocks longitudinally, denoted by 'M' in Fig. 1, with each having a capacity of 16 KB SRAM and can be accessed by pipeline or direct memory access (DMA) engine. The size of the SRAM can be easily changed (increased) depending on the need and application.

In order to process the kernels faster in multimedia and communication applications, this work further employs 4 identical expanded execution array units (denoted by 'E') in the transverse direction. Each execution array unit consists of accelerators, performing operations like entropy decoder for H.264 and 16-point FFT, matrix multiplier for LTE. It needs to be noted that, only execution array die needs to be redesigned and manufactured in case of altering applications.

A. Core

Core is one of the main blocks in the proposed multicore microprocessor design. We design a microprocessor without interlocked pipeline stages (MIPS) core, shown in Fig. 2. The design is a typical 6-stage pipeline structure of MIPS core and has enriched instruction set to support SIMD and direct-memory access (DMA) operations. The six-stage pipeline structure is presented in Fig. 2. The pipeline consists of instruction fetch, decode, execute, memory, align, and write back and commit stages [4]. Data-level parallelism (DLP), configurability and complexity are the major design principles. The single instruction multiple data (SIMD) instruction set architecture (ISA) is utilized for DLP. The SIMD supports data of different widths and computing modes: scalar-scalar, scalar-vector and vector-vector operations as in [4].

To improve the data locality and reduce power consumption, the register file of MIPS core is set as 64 words. This extended register file results in more available registers indicating more

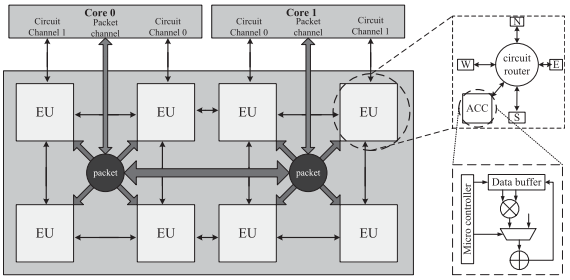


Fig. 3. Execution array of NoC multicore microprocessor.

capacity to allocate data used by SIMD instructions and improve performance.

B. Router

In a multicore microprocessor, the communication between cores or core and memory is important and is often the bottleneck. In our design, we make use of a NoC topology. The communication is carried out with the help of a router. The data from the core is sent to the router for the purpose of routing the data to the destination in our architecture. A five-port router is designed for the purpose of communication in a mesh topology. The central port of router is connected to the core and other four ports connect the adjacent routers or to the interface of off-die components i.e., memory/accelerator.

Each router consists of three components: input buffer to store the data with simple FIFO logic, an arbiter to determine the data transferring sequence and direction according to the routing algorithm, and a crossbar switch for the purpose of making switching based on the result from arbiter. Message-passing mechanism is widely utilized due to its better scalability [30]. However, we discuss a hybrid communication mechanism similar to that used in [6] to support both message passing and shared memory to achieve higher performance and energy efficiency.

In our implementation, a flit-based switching is implemented, which requires a small amount of channel buffer, and wormhole routing flow control is implemented for efficient buffer utilization. The size of flit indicates the width of the link connecting routers. In our proposed architecture, size of a flit is set to 67 bits with last 3 bits indicating the validity and flit type (i.e., head, body or tail flit). In head flit, the type of operation is denoted, with upper 64 bits include packet source, destination, operation type, memory access address for off-die memory or DMA address for off-die accelerator and so on. In body and tail flits, 64 bits denotes the kind of operation i.e., computation or store. An XY-ordered dimension order routing (DOR) algorithm is implemented by fixed logic, to generate deadlock-free routes in implemented mesh topology [30].

We have two kinds of routers in our design, one is packet-switched router, and the other is circuit-switched router. The NoC organization is inherited from our previous work [4].

C. Execution Array

Execution array (EA) is a mixed-granular spatio-temporal hardware reconfigurable framework. A framework of array of accelerators is shown in Fig. 3. Similar to cores, it consists of an array of execution units (EUs) composed of accelerator (ACC) and a router, shown in top zoomed-in part of Fig. 3. Architecture of accelerator is presented in bottom zoomed-in part of Fig. 3. Accelerator helps in speeding up the processing, but is limited to a particular application(s) for which it is designed. Accelerator communicates with external environment using a configurable hierarchical interconnect architecture. The target application to be mapped to the accelerator are mostly compute-intensive tasks. As the hardware supports multi-input multi-output tasks, more tasks can be executed on the accelerator in parallel.

In this work, accelerators can be divided into two classes: one is the basic accelerator and the other is a coarse-grained accelerator. Basic accelerators i.e., fine-grained accelerators such as adder, multiplier and shifter have small-scope granularity. To finish a complete task, they must be connected with other accelerators through interconnects. As such, these accelerators have high flexibility but complex configurations. They can be invoked simply but have lower flexibility. In this work, eight hardware accelerators are designed: four accelerators for H.264 application, two (FFT_8 and FFT_16) for FFT, one (Cordic) for triangular transformation and one for matrix multiplication. Accelerators such as Cordic, and FFT_8 are fine-grained, others are coarse-grained. A local data buffer is responsible for storing the temporary operands and results. Their programming models are similar with each other. They need to configure the data path, send data into right accelerator, and then receive computing result from the output port. The communication cost scales proportionately with the number of input data, due to the process of serial-to-parallel conversion between two dies. The number of accelerators and the type of accelerators is dependent on the application. The sequence of operations is controlled by a micro-controller referred to as schedule table. Most of the tasks cannot be accomplished in one clock cycle, as they need an accurate control of microcontroller and data buffer, as shown in the zoomed-in bottom part of Fig. 3.

III. Extended Memory System and Management

All on/off-die data memory blocks are shared by cores within the same cluster, and DMA operation is supported to transfer streaming data background between any-to-any on-die memory and off-die memory/accelerator, including across dies in core-core expansion mode.

A cache-free memory hierarchy is adopted in the proposed 2.5D multicore microprocessor architecture. According to the degree of coupling between pipeline and memory blocks, the extended memory system is divided into four logic hierarchy as: 1) register files, 2) the intra-cluster on-die memory, 3) the intra-cluster off-die memory, and 4) the inter-cluster memory. Register files can be considered as a part of the pipeline. MIPS core uses load/store architecture which only allows memory to be accessed by load and store operations. Operands of all other operations come from register files. Both on-die and off-die data memory in a cluster are distributed shared and can be accessed via load/store operations directly by any core in the cluster. On/off-die SRAMs are further split into 0.5 K word/bank, i.e., 2 K byte/bank, to improve the memory access

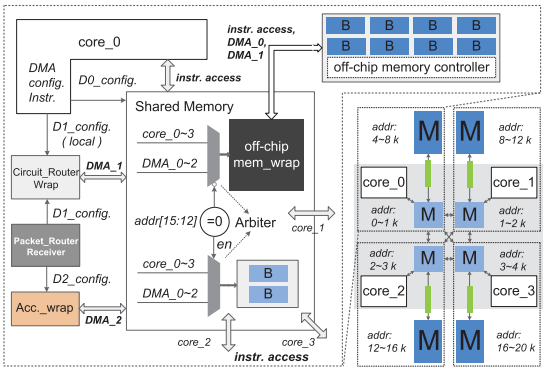


Fig. 4. Extended memory hierarchy with DMA control.

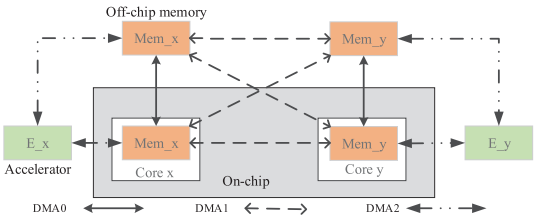


Fig. 5. Execution flow of DMA for memory access.

TABLE I
SUMMARY OF DIFFERENT COMMUNICATION PROTOCOLS

Transfer type		Access method	Latency (read)	Latency (write)
On-die	Local	Load/store	1	1
	Intra-cluster	Load/store	2	2
		DMA 1	No. of hops+ No. of data	No. of hops+ No. of data
	Inter-cluster	DMA 1	No. of hops+ No. of data	No. of hops+ No. of data
Off-die	Local	Load/store	In Fig. 19	In Fig. 19
	Intra-cluster	DMA0	In Fig. 19	In Fig. 19
		Load/store	In Fig. 19+2	In Fig. 19+2
		DMA 1	No. of hops+ In Fig. 19	No. of hops+ In Fig. 19
	Inter-cluster	DMA 1	No. of hops+ In Fig. 19	No. of hops+ In Fig. 19

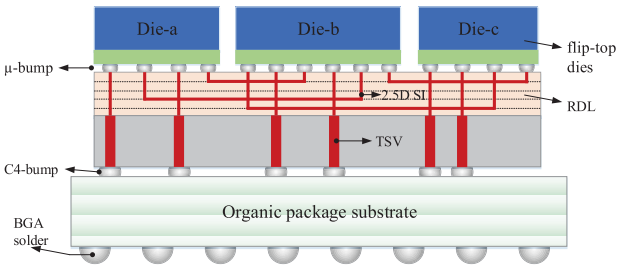


Fig. 6. 2.5D integration technology using 2.5D silicon interposer interconnect.

bandwidth and shorten the critical timing path. In each cluster, there is 4 K word on-chip data memory, each core 1 K word as local data memory with an access latency of 1 clock cycle and the rest of 3 K word as a remote data memory with 2 clock cycles access latency. The 32 K word off-die data memory is extended for each cluster. The access latency for off-die memory access is small and is shown in the later part of this paper. However, as the design is flexible i.e., the number of off-die connecting blocks can be increased or decreased, the amount of memory can as well be changed (increased or decreased). The memory details provided here are for the used application, whereas the designers are free to vary them in the design, depending on the application. For inter-cluster memory, DMA operations are supported to transfer streaming data background. As illustrated in Fig. 4, all access requests from cores and DMAs are served by an arbiter with round-robin algorithm to avoid deadlock.

The on-die memory is utilized to store run-time program variables and synchronization flags between cores, due to its low access latency (one clock cycle for local, and two for remote), while the off-die memory is for streaming data storage in embedded applications, whose large interface latency can be significantly hidden by DMA.

We designed a rich and flexible DMA mechanism to support background data transfer in various scenarios, which uses the software pre-fetching to hide the off-die access latency, improve the system performance significantly by reducing the performance cost of pipeline stall. As shown in Fig. 5, it supports the DMA transfer between any two memories, such as intra-core local transfer (DMA0): transfer between local on-die memory and local off-die memory; inter-core remote transfer (DMA1): transfer between local on-die memory and remote on-die memory, local on-die memory and remote off-die memory, local off-die memory and remote off-die memory. In addition, the DMA also supports the transfer between accelerator and memory, i.e., DMA2: transfer between local on-die memory and local off-die accelerator, local off-die memory and local off-die accelerator. Summary of different communication protocols is presented in Table I.

IV. INTERFACE CIRCUITRY BETWEEN DIES

The interface circuit is a critical part in the 2.5D integrated system, primarily dominated by the limitation of I/O resources for inter-die connection, since all the I/O cells (providing driver and ESD for package I/O) and transmitting and receiving (TRX) modules ought to be routed to the landing metal block on the top layer (as shown in Fig. 15) for micro-bump bonding. Involved I/O blocks are explained below.

A. 2.5D Silicon Interposer T-Line Interconnect

As shown in Fig. 6, the flip-top dies are mounted on silicon interposer side-by-side and connected with 2.5D silicon interposer T-lines i.e., with an array of TSVs, micro-bumps (μ-bump) and RDLs. Some of the μ-bumps are interconnected by the metal wire in RDLs (redistribution layers) for inter-die communication, which do not require large I/O pads, thus saving area, while others are linked to C4 micro-bumps through TSVs (for off-die communication), routing P/G and I/O signals for conventional ball grid array (BGA) package. The 2.5D silicon interposer based I/O has a custom-designed TRX (I/O link), which includes transmission line, VCO and a sampler connected to CDR, discussed in Section IV-C. The method to integrate chips makes 2.5D stacking better than 3D vertically stacking mainly in terms of thermal dissipation.

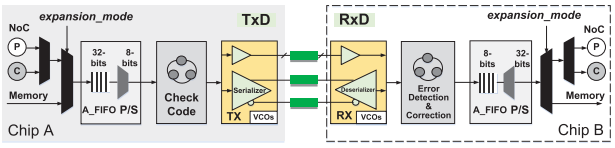


Fig. 7. 2.5D silicon interposer I/O interface circuits between dies.

The thermal runaway problem in many-core microprocessors is less pronounced in 2.5D integration compared to the 3D integration, and 2.5D silicon interposers are thermally resilient [26].

B. Digital Circuits

The interface circuitry between dies is illustrated in Fig. 7, in which the digital circuits comprises of asynchronous FIFO, digital Parallel-Serial and Serial-Parallel circuit, and error code detection and correction modules. At system boot phase, the configuration register of 'expansion mode' is initialized, together with 'coordinate of router'. The 32-bit packetized data from the double layer NoC (at core-core expansion mode) or off-die memory-access (at core-memory expansion mode) is selected by the multiplexer, and then stored in an asynchronous FIFO for digital-analog clock domains isolation. Parallel-Serial (32bit-8bit) and Serial-Parallel (8bit-32bit) logic circuits are inserted on the basis of the I/O quantity limitation. The 8-bit data is further transmitted through pair of 2.5D silicon interposer differential channels at 8 Gbps speed, handled by an analog SerDes. Along with the serialized data, 3 bits of the control signals are separately transmitted, namely w_en, w_full, w_index: the first two are for FIFO write protocol, and the last one is for particular notification decided by designer.

To solve the possible bit disorder problem at the output of Deserializer, a circuit is designed, which is activated during the boot phase and then outputs data with corrected bits' order by sending and detecting a serial of check code. This is solved as follows. Firstly, at the reset phase, under the control of finite-state-machine (FSM) in transmitter (TX), the digital circuit outputs "8'b10101010" to let the Serializer at Tx generate full toggle signals, from which CDR at the Deserializer will recover the clock. Once the reset is released, check code generation module at Tx will output 2 cycles of "8'b11111111" (for data synchronization) followed by "8'b01111111" (for checking); At the receiver end, once the error bit detection module captures first "8'b11111111", it checks for the next byte, if it's also all 1, no error bit happen, otherwise it will output the position index of the bit "1'b0" to the error correction module, which will re-organize the bits order and receive the corrected data.

C. TRX I/O Link

Another important component in the multicore memory-logic integrated system is the interconnect i.e., I/O link. We make use of the 2.5D silicon interposer I/Os to achieve higher speed with low power and better energy efficiency. Here we will discuss about the I/O utilized in our microprocessor design. The custom-designed TRX consists of 3 buffers driving low-speed control signals, as well as

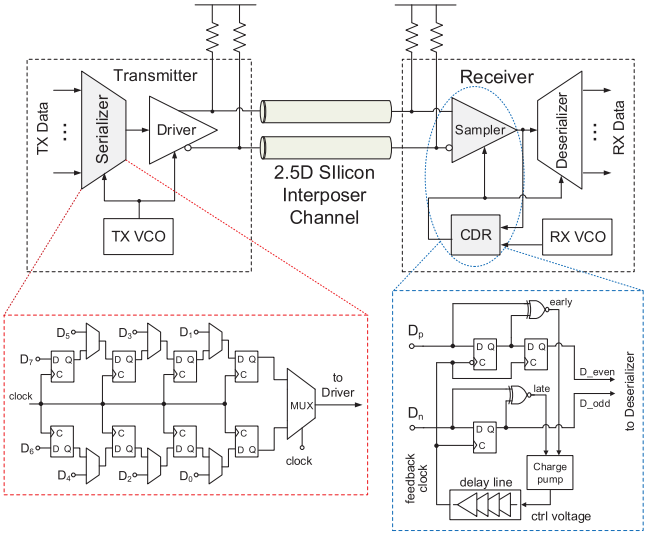


Fig. 8. 2.5D silicon interposer SerDes I/O circuits.

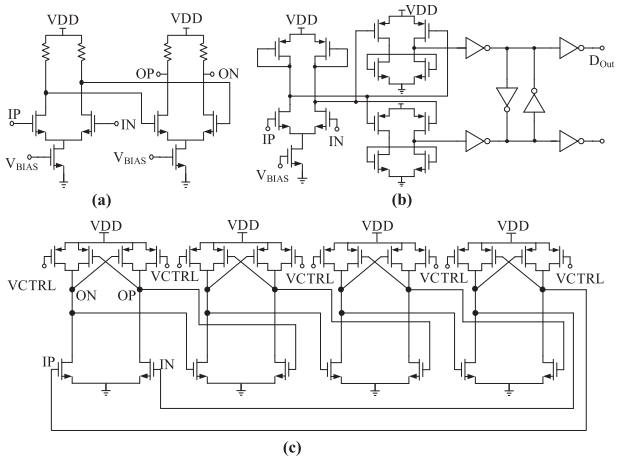


Fig. 9. (a) Driver at transmitter; (b) sampler at receiver; (c) synchronizer.

a serializer-deserializer (SerDes) transferring 8-bit data is depicted in Fig. 8.

The I/O design consists of transmitter (TX), channel and receiver (Rx) parts. The input data is first serialized with the help of serializer and further voltage-controlled oscillator (VCO) is utilized to maintain the clock frequency. A 8:1 serializer is employed at the transmitter to serialize the data. Further, to drive the 2.5D silicon interposer channel, a driver is implemented after the serializer.

1) Transmitter: Transmitter (TX) uses a 8:1 serializer to convert 8-bit parallel data into serial data, as shown in the left bottom zoomed-in box of Fig. 8. Four digital D flip-flops are implemented as a shift-register chain for each of the odd (D_1, D_3, D_5, D_7) and even (D_0, D_2, D_4, D_6) bits of data. This is followed by a 2:1 MUX to combine them altogether. This serializer is followed by a current-mode logic (CML) output driver to drive the T-line from the Tx to the Rx.

The driver at the transmitter utilized for 2.5D I/O is shown in Fig. 9(a). The driver circuit implemented at the transmitter of multicore I/O consists of two stage cascaded CML buffers. To alleviate the mismatching of impedance, 50Ω resistors

are utilized for the impedance matching of the transmission line.

2) Receiver: Compared to the traditional serial I/Os based on the backplane PCB trace [31], the 2.5D silicon interposer I/Os do not need the complex equalizer circuits at the receiver due to the small signal loss in the 2.5D silicon interposer T-line channel. At the receiver end, a sampler connected to clock-data recovery (CDR) is employed to convert the current-mode signals into digital CMOS level signals. The design of sampler with amplifier followed by two current mode comparators to convert the signal into differential logic digital signals is shown in Fig. 9(b). Processing of data in digital domain saves more power compared to analog de-multiplexer.

A delay-locked-loop (DLL) based CDR at the receiver is implemented to de-skew the sampling clocks, as shown in the right bottom zoomed-in box of Fig. 8. Two exclusive-or (XOR) gates in the right bottom zoomed-in box of Fig. 8 form a phase detector to judge the sampling clock position compared to input data and provide "early" pulse and "late" pulse. A charge-pump block converts these pulses into a variable voltage to control the DLL delay line, which can tune the delay phase of clocks and also provide feedback to the sampler. In this CDR design, a half-rate clock architecture is employed to decrease digital circuit working frequency and save power consumption. Further, these received samples are further converted back to parallel data with the help of 1:8 deserializer.

3) Voltage-Controlled Oscillator: For the voltage-controlled oscillator (VCO) design, a ring VCO is employed to get eight different phases for the whole system. Compared to LC tank oscillator, the ring VCO could get worse phase noise and low frequency, but has lower power consumption and area, which can be considered as a trade-off. Hence, we choose the ring VCO for this design. The VCO architecture and delay cell unit are shown in Fig. 9(c).

V. System Communication

The communication between cores is often the bottleneck in a multicore design, hence, it is important to have an efficient communication protocol to effectively utilize the bandwidth.

A. Core-to-Core Communication

First of all, we would like to clarify that communication between different components is handled differently in our work. Core-to-core communication is different from the core-to-memory communication. The communication between core-to-core happens through the routers. Router is provided with the input in form of a message from the core or other adjacent components. Router is partitioned into two layers i.e., packet layer and circuit layer. Depending on the kind and size of data to be transferred through the routers, the router's layer is utilized. Packet switched NoC is utilized to transfer small amounts of data and in a synchronized communication. Whereas, to transfer continuous and large data transfers like multi-media, big data applications, circuit-switched router (NoC) is utilized, which is flexible and configures to all combinational wire connections, resulting in less latency and better energy efficiency. This communication

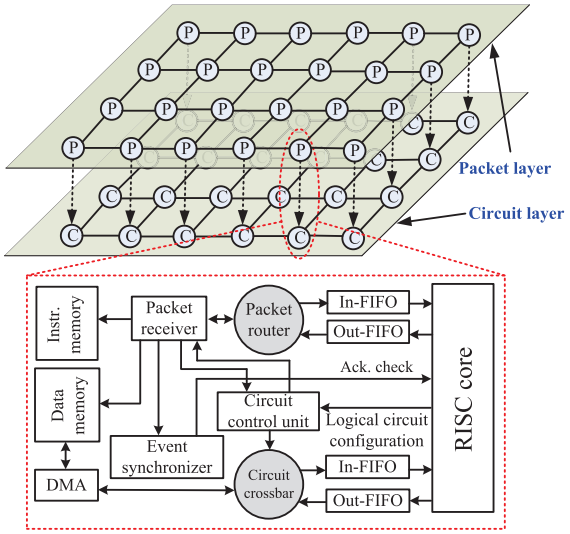


Fig. 10. Circuit-switched double-layer NoC with packet-control.

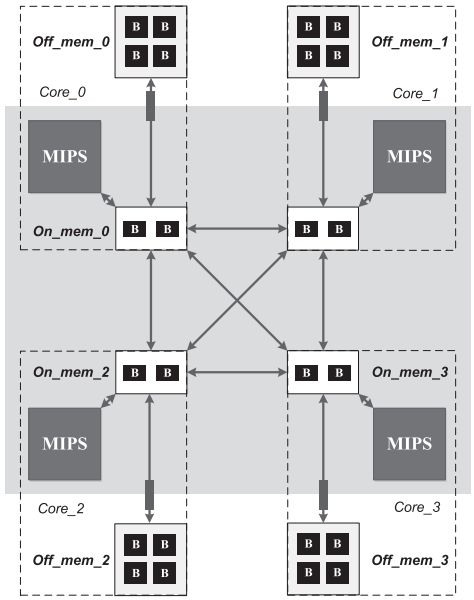


Fig. 11. Communication flow of core-memory.

is shown in Fig. 10. More details of this communication is presented in our previous work [4].

B. Core-to-Memory Communication

Core-to-memory communication happens through the dedicated pipeline memory access ports by hard wired connection, as shown in Fig. 11. Each core has 4 KB on-die data memory and 16 KB more for expansion in off-die SRAM. All memory is shared and can be accessed by pipeline of all cores within a cluster. Note that inter-die access has a grand delay in I/O bandwidth bottleneck leading to significant performance degradation. Hardware architecture provides abundant types of direct memory access (DMA) operation to hide off-die access latency, such as: on-off, off-off data DMA between memories; DMA between different multicore dies when at core-core expansion mode; and DMA between local memory and off-die accelerator. Arbiter helps to resolve the conflicts when

simultaneous communication from multiple cores to same port happens.

Core-memory and core-accelerator connections adopt the same interface circuitries, except that the former one employs additional multiplexer logic, making it support core-core connection by double-layer NoC expansion, thus obtaining multiple computational power and beneficial to the applications like big-data, neuroscience computing, and many others. The ports of packet/circuit switched routers located on the left/right sides are assigned to system I/O, which limits the core-core expansion transversely in this prototype.

C. Communication in Execution Array

The interconnection we utilize in the execution array is very similar to core-to-core communication, namely packet-controlled circuit-switched double-layer NoC, which enhances the scalability of execution array. The usage of NoC makes the execution array reconfigurable, parallel and sharable among multiple cores. As shown in Fig. 3, the packet router receives and parses the configuration packages from microprocessor, which contains the information of using circuit router and accelerator, and then the data path is set in one clock cycle. In the next step, the microprocessor sends the data to execution array through circuit channel, the data flows along the data path, into the accelerator. Finally, the output of the accelerator is feedback to the microprocessor. The communication between core and accelerator can be DMA or normal register instructions, the DMA has better transmission efficiency.

VI. Algorithm Mapping

The effectiveness of running an application depends on how well the mapping of hardware and software is performed. Hence, we discuss the mapping of software (algorithm) onto the hardware platform considering an example in this section.

The application of multicore microprocessors ranges from running independent tasks with essentially no communication to running parallel programs where threads must communicate to complete the task. Two important hurdles make parallel processing challenging. The first hurdle has to do with the limited parallelism available in programs, and the second arises from the relatively high cost of communication. The degree to which these hurdles are difficult or easy is determined both by the application and by the architecture.

When mapping applications to multicore microprocessor, both thread-level parallelism (TLP) and data-level parallelism (DLP) are available. We first analyze parallelism in the program, then select the appropriate parallel strategies according to the characteristics of parallel multicore microprocessor architecture. We explain the mapping using the H.264 decoder as a case study.

The H.264 standard has a high compression rate as well demands a large amount of computation. The H.264 decoder functional diagram is shown in Fig. 12. Entropy decoder (DEC) parses the coded video bit stream, generating transformed residual data for iTrans/iQuant (IT/IQ), predictive modes for Intra Prediction (INTRA) and motion vectors for Inter Prediction (INTER). IT/IQ demonstrates inverse-transform to restore residual, adding predictive pixels from

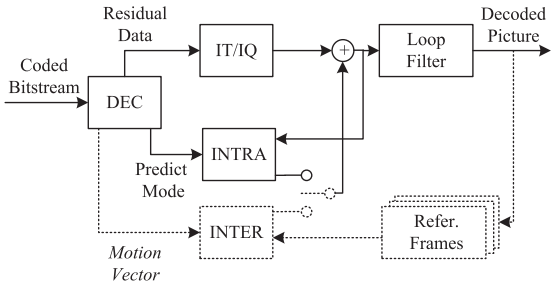


Fig. 12. Execution flow of H.264 decoder application (the solid-line portion is the block diagram of the intra decoder as baseline profile).

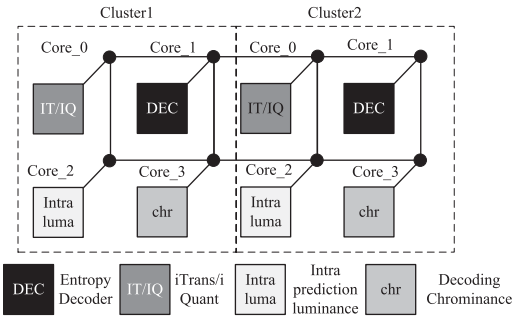


Fig. 13. Mapping of H.264 decoder application on 8-core.

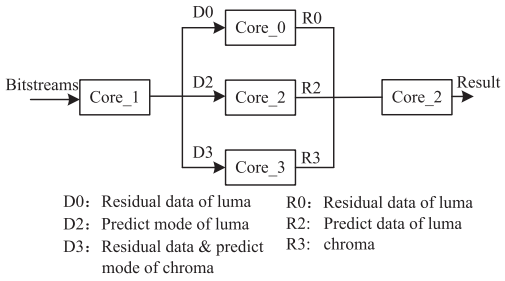


Fig. 14. Data flow of mapped H.264 decoder application on 8-core.

INTRA/INTER to finally reconstruct original frames (not considering loop filter). Decoded frames are stored as references for INTER.

For this particular applications, H.264 defines the profiles and levels specifying restrictions on bitstreams like some of the previous video standards. Seven profiles are defined to cover the various applications from the wireless networks to digital video streaming. Each profile specifies a subset of entire bitstream of syntax and limits that shall be supported by all decoders conforming to that Profile [32]. As shown by a solid line Fig. 13, a H.264 intra decoder of baseline profile, which profile is to be applicable to real-time conversational services such as video conferences and videophone application, is implemented on our 2.5D multicore microprocessor.

As depicted in Fig. 13, 8 cores are used to map two H.264 decoders, each cluster mapping one decoder. In each cluster, 4 cores are used for entropy decoder, IT/IQ, intra prediction and chroma decoding respectively.

The data flow of H.264 decoder implemented in our 2.5D multicore microprocessor platform is shown in Fig. 14, where Di ($i = 1, 2, 3$) indicates the input data of core i and Ri

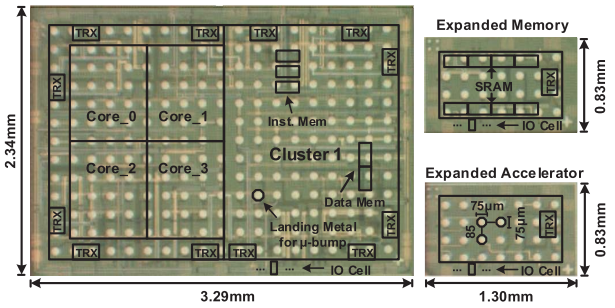


Fig. 15. Chip photo of fabricated reticule with multicore, accelerator, and memory dies with μ-bumps.

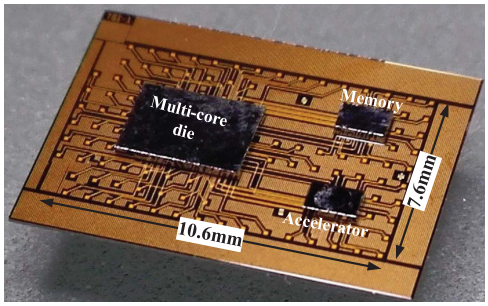


Fig. 16. Chip photo of 2.5D integration of dies.

indicates the output data of core i. As shown in Fig. 13, Core 1 is used as the entropy decoder which parses the input coded video bitstreams to get transform coefficients of residual data and predict modes. Core 0 reads the transformation coefficients of luminance residual data from the shared memory and finishes the inverse transform and inverse quantization. Core 2 reads the predict modes of luminance from shared memory and creates predict data of luminance. The addition of residual data and the predict data are also performed in Core 2. The decoding of chroma, including IT/IQ, prediction and the addition, is done in Core 3. At last, Core 2 sends both the chrominance data and luminance data through packet NoC to off-die.

Both spatial and temporal factors are considered in this mapping. From the spatial aspect, considering large amounts of data transferred within DEC, IT/IQ and INTRA, a reasonable solution is to map them in the same cluster to make full use of shared-memory communication. From the temporal aspect, to achieve targeted throughput and workload balance, a stage of pipeline is usually divided into several parallel threads. By iterative simulation we finally achieve core number allocation as mentioned.

Algorithm mapping and use of accelerators in case of H.264 decoding could be summarized as follows. We use hardware and software codesign to accomplish the decoding. Namely the core is responsible for the whole work. But during execution, it will call hardware accelerator to execute the compute-intensive works for which it is designed. After getting the results, the core will proceed with the next software task.

VII. Results

The multicore die with memory and accelerator is fabricated in 65 nm CMOS process. Fig. 15 presents the micrograph photo of the fabricated reticule which comprises of a multicore die, an accelerator die and a memory die. Gold stud bumps with a height of 70 μm are added on all pads for further flip-die bonding and 2.5D integration with silicon interposer. Fig. 16 provides the initial integration of multicore die with one accelerator die and one memory die using silicon interposer. The 10.6 mm $\times$7.7 mm silicon interposer chip is fabricated by authors with a procedure as in [29]. There are two RDL layers on the front side and one RDL layer on the backside of the interposer. Thickness of top and bottom RDL layers are 5 μm and 3 μm, respectively. The recommended RDL trace width and space is 10 μm. Micro-bumps with height, width and pitch of 70 μm, 75 μm, and 160 μm respectively are fabricated on the front side for multi-die (core, memory, and accelerator) bonding, while copper pillar bumps with a diameter of 250 μm and height of 80 μm were fabricated on the backside for interposer to substrate or chip-on-board bonding. Wet silicon recess etching process is used for TSV backside reveal, reducing the cost for fabrication and 2.5D multi-die integration [29].

Firstly, we show the comparison between presented 2.5D silicon interposer and the PCB trace, followed by evaluation of other presented components.

A. 2.5D Technology Performance Analysis

The traditional interconnection between microprocessors and memories is by printed circuit board (PCB) with backplane [33] containing sockets into which other boards can be plugged in (See Fig. 17(a)(i)). However, long trace ($\geq$ 25cm) and non-ideal vias are needed at PCB scale, hence there is a severe loss on the backplane, which requires current-starved circuits to reach the high data rate and equalizers to compensate the channel loss [33]. For the 2.5D silicon interposer [34], the microprocessors and memories dies are integrated on one common substrate by silicon interposer underneath (See Fig. 17(a)(ii)). Unlike traditional backplane based interconnects, 2.5D silicon interposer are much shorter with a few *mm* in length and are deployed underneath the substrate with less routing overhead. The channel loss vs. frequency is shown in Fig. 17(b) for PCB backplane I/O and 2.5D silicon interposer I/O, respectively. When comparing the loss at 5 GHz clock frequency, the PCB backplane with long (25 cm) trace has nearly 24 dB channel loss; and the 2.5D silicon interposer with small trace (10μm width, 3 mm length) has only 1 dB loss. The PCB technology used for comparison is IPC class II standard. Hence, the 2.5D silicon interposer based integration has much less loss with better performance for the memory-logic-integration.

B. Performance Evaluation

1) System Performance: To demonstrate the performance of this 2.5D integrated multicore microprocessor, an H.264 intra decoder is mapped to 8 cores. By calling entropy decoder in off-die execution array (accelerator die), which features coarse grained and loosely coupled with pipeline, the decoder performance achieves 720p@34fps at 500 MHz, which is 4.4$\times$ better than pure software solution, and also 1.7$\times$ better per

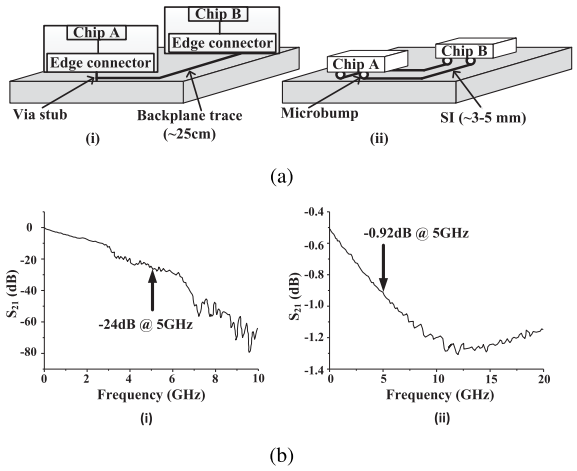


Fig. 17. (a) Interconnect by: (i) backplane PCB trace (ii) 2.5D silicon interposer T-line; (b) channel loss for: (i) backplane PCB trace; (ii) 2.5D silicon interposer T-line.

TABLE II
PERFORMANCE OF CORE

Technology	65nm LPE
Frequency	500MHz @ 1.2V
Typical power	1.08W
Energy efficiency (1 core)	20GOPS/W (51pj/OP)
# of μ-bumps	246
2.5D I/O speed	8Gbps (max)
Inter-die bandwidth	24 GB/s
Expansion mode	core-core, core-memory, core-accelerator

TABLE III
PERFORMANCE OF CORE AND COMPARISON

Metric	Proposed	[6]	Pentium [35]	[36]
Technology	65nm LPE	65nm	65nm	65nm
# of cores	16	16	1	4
Power dissipation	1.08W	498 mW	10W †	139mW
Performance	720p@34fps	720p@34fps	720p@34fps	720p@34fps
Energy efficiency	34.5nj/pix	13.5nj/pix	362nj/pix	5.0nj/pix
Integration	2.5D	2D	2D	2D

core on average than its predecessor [4]. The performance of a core for H.264 application is presented in Table II. Further, comparison of performance with other works performing running similar application is presented in Table III.

High-performance advanced encryption standard (AES) is also implemented on our proposed 2.5D architecture. Two parallel AES schemes are implemented, one is full software implementation and the another is software implementation with hardware accelerator [37]. Benefited from the high efficient inter-core communication obtained by shared-memory mechanism, the parallel software scheme shows a speedup of 2.46× on a 4-core implementation and 4.92× on an 8-core system compared with single-core microprocessor. The scheme of software implementation with hardware accelerator for mix columns can double the performance with a speedup of 9.78×, with 176.48 Mbps throughput. This shows the reconfigurability of the proposed 2.5D integrated architecture and its advantage, which can increase the system computing resources without additional tape-out costs.

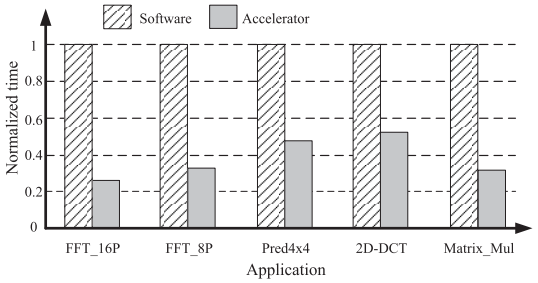


Fig. 18. Performance evaluation of accelerator against software for different tasks.

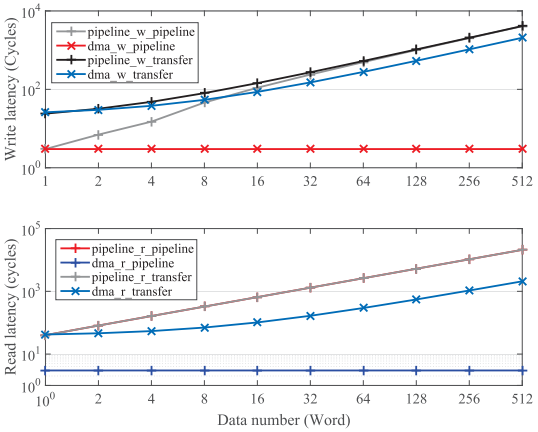


Fig. 19. Local access latency to off-die memory for write and read operations.

In addition to H.264 decoding and AES applications, a typical big data application - clustering algorithm is implemented by 4 cores basing on a 16000 point three-dimensional array of dataset, utilizing canopy method proposed in [38]. Simulation results show that after well-inserted DMA operations in accessing off-die memory, the overall computational time has been reduced 43.8% compared to instruction read/write. Besides, it achieves 3.8× performance improvement after mapped into 16 cores when expanding an 8-core microprocessor, which exhibits the flexibility and scalability of the proposed 2.5D integrated architecture.

Fig. 18 illustrates the speedup of some applications running on our multicore system with execution array. We can see that for different types of tasks, the speedup is different. For example, for a 16-point FFT application, the speedup is nearly 5×, whereas for the 2D-DCT application, the accelerator achieves only 2× speedup compared to the software. Considering the I/O delay between multiple chips in 2.5D system, not all applications are amenable to this acceleration framework, because the delay can degrade the performance largely. The applications studied are therefore categorized as:

- Applications which are compute-heavy and the input and output size is small benefits the most.
- Applications with low computation requirement and the input and output size is large benefit less.

2) Off-Die Access Latency: Fig. 19 illustrates the local access latency to the off-die memory, and it shows that

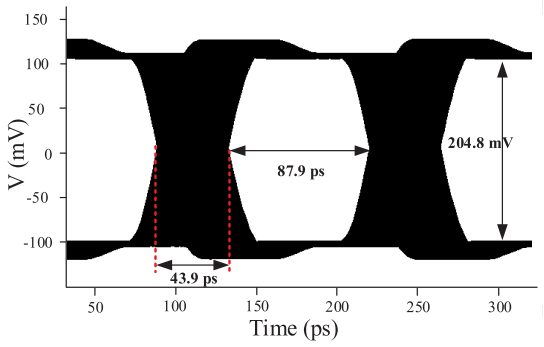


Fig. 20. The output at 2.5D silicon interposer SerDes I/O transmitter.

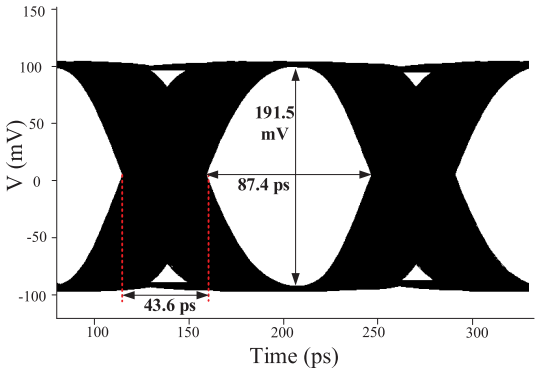


Fig. 21. The input at 2.5D silicon interposer SerDes I/O receiver.

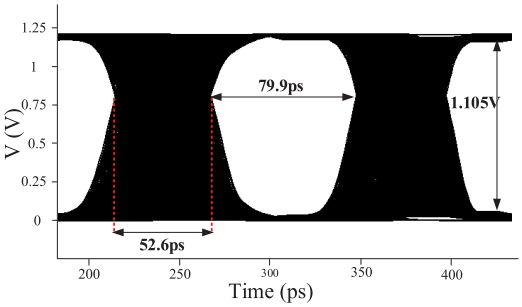


Fig. 22. The output at 2.5D silicon interposer SerDes I/O receiver.

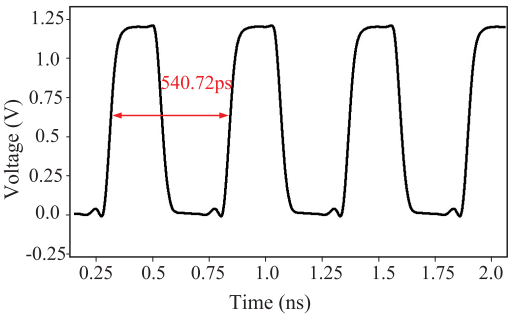


Fig. 23. The waveform of synchronizer VCO.

DMA does not occupy pipeline time since it transfers data background once the special configuration instructions are executed. Besides, DMA requires less packet head and thus consumes less data transfer time. The DMA write/read saves up to 49.8% and 90.3% transfer time (latency) respectively at 512 words compared with pure instruction write/read.

As the data characteristic of our specific application is mostly continuous and stream-like, the various DMA mechanisms act as hardware pre-fetching triggered by special instruction. This cache-free memory hierarchy could avoid coherence issues and maintain system simplicity [6].

C. I/O Component Performance

The system global clock is provided by VCO in I/O, whose frequency can be adjusted by control voltage The multicore, off-die memory and off-die accelerator use multi-clock and gated clock technology to reduce power consumption. Multiple clock domains are implemented, some lower frequency clocks are employed to initialize the multicore, and higher frequency utilized for normal operation. Many asynchronous reset signals are designed for each cluster (set of four cores), off-die memory and off-die accelerator respectively. For the convenience of testing, there are two sequential data input and output signals implemented in the multicore and bond-out to test board. The function of sequential data is similar with scan chain. One can set a sequential to input data and collect the result from output, and check the status of inner key register, initialization of instruction cache and so on.

With the main focus on I/O and VCO design, we study the signals at the receiver and transmitter. With the power supply V_{DD} of 1.2 V for I/O, the input and output signal at the I/Os are studied. The input signal to the transmitter is a 4 GHz signal. The corresponding differential output at the transmitter is shown in Fig. 20. The output swing (peak-to-peak) is 204.8 mV, with a cycle-to-cycle jitter is 43.9 ps. The bandwidth of the transmitter is 4 GHz (for individual I/O).

When the signal is transmitted from the transmitter through the designed 2.5D silicon interposer I/O, a minimal loss is experienced at the receiver input. Fig. 21 shows that the differential signals received after 2.5D silicon interposer transmission line, and the eye diagram of it. An amplitude of about 191.5 mV peak-to-peak voltage is observed at the input of receiver with a loss of nearly 13 mV over 3 mm T-line. The cycle-to-cycle jitter at the input of receiver is 43.6 ps. To reconstruct the signal, the receiver input is passed through the equalizer, which is presented in Fig. 22. The amplitude of the eye opening is 1.105 V, with an eye opening width of 79.9 ps and a cycle-to-cycle jitter of 52.6 ps. The power consumption of a single I/O is 29.9 mW with a bandwidth of 4 GHz, resulting in energy efficiency of 7.5 pJ/bit.

The characteristics of the VCO, which generates the clock for the operation of the system is presented here. The output frequency and amplitude of the VCO depend on the input supply voltage. Fig. 23 shows the VCO post layout simulation, With the voltage control (VCTRL) signal set to 830 mV, the frequency of VCO is nearly 1.9 GHz.

VIII. Conclusion

This paper presents a 2.5D integrated multicore network-on-chip, which consists of 3 heterogeneously integrated silicon dies: microprocessor die with 8 MIPS cores, memory die for

data capacity expansion, and an accelerator die for faster data processing such as H.264 video decoder. The 2.5D integrated chip was implemented in GF 65 nm process with multicore microprocessor operated at 500 MHz under 1.2 V supply with 1.08 W power dissipation. The 2.5D silicon interposer based I/Os support 12-way full-duplex communication in parallel, bringing the bandwidth up to 24 GB/s with 7.5 pJ/bit energy efficiency. As a result, the specified applications such as H.264 video data analytics and AES encryption have achieved significant performance improvement of throughput and energy efficiency. In addition, the demonstrated 2.5D integration is also scalable and configurable to be organized into various multi-chip systems to meet different data analytic application requirements, thus saving none recurring engineering (NRE). A more advanced 2.5D process facilitates higher scalability by accommodating more micro-bumps on dies for inter-die integration, higher communication bandwidth with low power.

ACKNOWLEDGEMENT

This work was supported in part by grant from Singapore MOE Tier-2 fund MOE2015-T2-2-013, MOE2010-T2-037, A*STAR PSF fund (11201202015), and support from Intel Lab and Samsung Corporation.

REFERENCES

[1] J. Lin, S. Zhu, Z. Yu, D. Xu, P. D. S. Manoj, and H. Yu, "A scalable and reconfigurable 2.5D integrated multicore processor on silicon interposer," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2015, pp. 1–4.

[2] M. Horowitz, "Computing's energy problem (and what we can do about it)," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 2014, pp. 10–14.

[3] Z. Yu *et al.*, "A 16-core processor with shared-memory and message-passing communications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 4, pp. 1081–1094, Apr. 2014.

[4] P. Ou *et al.*, "A 65nm 39GOPS/W 24-core processor with 11Tb/s/W packet-controlled circuit-switched double-layer network-on-chip and heterogeneous execution array," in *IEEE Int. Solid-State Circuits Conf. Dig.*, Feb. 2013, pp. 56–57.

[5] Z. Yu *et al.*, "AsAP: An asynchronous array of simple processors," *IEEE J. Solid-State Circuits*, vol. 43, no. 3, pp. 695–705, Mar. 2008.

[6] Z. Yu *et al.*, "An 800MHz 320mW 16-core processor with message-passing and shared-memory inter-core communication mechanisms," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 2012, pp. 64–66.

[7] D. N. Truong *et al.*, "A 167-processor computational platform in 65 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 44, no. 4, pp. 1130–1144, Apr. 2009.

[8] S. R. Vangal *et al.*, "An 80-tile sub-100-W teraflops processor in 65-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 43, no. 1, pp. 29–41, Jan. 2008.

[9] S. Bell *et al.*, "TILE64—Processor: A 64-core SoC with mesh interconnect," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 2008, pp. 588–598.

[10] M. Mantor, "AMD Radeon HD 7970 with graphics core next (GCN) architecture," in *Proc. IEEE Hot Chips Symp. (HCS)*, Aug. 2012, pp. 1–35.

[11] S. Che, J. Li, J. W. Sheaffer, K. Skadron, and J. Lach, "Accelerating compute-intensive applications with GPUs and FPGAs," in *Proc. Symp. Appl. Specific Process.*, Jun. 2008, pp. 101–107.

[12] Y.-W. Huang, B.-Y. Hsieh, T.-C. Chen, and L.-G. Chen, "Analysis, fast algorithm, and VLSI architecture design for H.264/AVC intra frame coder," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 3, pp. 378–401, Mar. 2005.

[13] M. A. Karim, P. D. Franzon, and A. Kumar, "Power comparison of 2D, 3D and 2.5D interconnect solutions and power optimization of interposer interconnects," in *Proc. IEEE Electron. Compon. Technol. Conf.*, May 2013, pp. 860–866.

[14] N. Kim, D. Wu, D. Kim, A. Rahman, and P. Wu, "Interposer design optimization for high frequency signal transmission in passive and active interposer using through silicon via (TSV)," in *Proc. IEEE Electron. Compon. Technol. Conf.*, Jun. 2011, pp. 1160–1167.

[15] M. Jung, T. Song, Y. Peng, and S. K. Lim, "Fine-grained 3-D IC partitioning study with a multicore processor," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 5, no. 10, pp. 1393–1401, Oct. 2015.

[16] D. Dutoit *et al.*, "A 0.9 pJ/bit, 12.8 GByte/s WideIO memory interface in a 3D-IC NoC-based MPSoC," in *Proc. IEEE Symp. VLSI Technol. (VLSI)*, Jan. 2013.

[17] M. P. D. Sai, H. Yu, Y. Shang, C. S. Tan, and S. K. Lim, "Reliable 3-D clock-tree synthesis considering nonlinear capacitive TSV model with electrical–thermal–mechanical coupling," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 32, no. 11, pp. 1734–1747, Nov. 2013.

[18] J. U. Knickerbocker *et al.*, "2.5D and 3D technology challenges and test vehicle demonstrations," in *Proc. IEEE Electron. Compon. Technol. Conf.*, Jun. 2012, pp. 1068–1076.

[19] H. Yu, J. Ho, and L. He, "Allocating power ground vias in 3D ICs for simultaneous power and thermal integrity," *ACM Trans. Design Autom. Electron. Syst.*, vol. 14, no. 3, 2009, Art. no. 41.

[20] M. Su, B. Black, Y. H. Hsiao, C. L. Changchien, C. C. Lee, and H. J. Chang, "2.5D IC micro-bump materials characterization and IMCs evolution under reliability stress conditions," in *Proc. IEEE Electron. Compon. Technol. Conf. (ECTC)*, May 2016, pp. 322–328.

[21] S. M. P. Dinakarrao, H. Yu, H. Huang, and D. Xu, "A Q-learning based self-adaptive I/O communication for 2.5D integrated many-core microprocessor and memory," *IEEE Trans. Comput.*, vol. 65, no. 4, pp. 1185–1196, Apr. 2016.

[22] D. Xu, N. Yu, P. D. S. Manoj, K. Wang, H. Yu, and M. Yu, "A 2.5 D memory-logic integration with data-pattern aware memory controller," *IEEE Des. Test*, vol. 32, no. 4, pp. 1–7, Aug. 2015.

[23] S. W. Ho *et al.*, "2.5D through silicon interposer package fabrication by chip-on-wafer (CoW) approach," in *Proc. IEEE Electron. Packag. Technol. Conf. (EPTC)*, Dec. 2014, pp. 679–683.

[24] J. Wang *et al.*, "High-speed and low-power 2.5D I/O circuits for memory-logic-integration by through-silicon interposer," in *Proc. IEEE Int. 3D Syst. Integr. Conf. (DIC)*, Oct. 2013, pp. 1–4.

[25] W. H. Ye and H. Li, "Design of Virtex-7 FPGA-based high-speed signal processor carrier board," in *Proc. Int. Conf. Control Eng. Commun. Technol.*, Jan. 2013, pp. 534–537.

[26] S.-S. Wu *et al.*, "A thermal resilient integration of many-core microprocessors and main memory by 2.5D TSI I/Os," in *Proc. ACM/IEEE DATE Conf.*, Mar. 2014, Art. no. 177.

[27] N. Sturcken *et al.*, "A 2.5D integrated voltage regulator using coupled-magnetic-core inductors on silicon interposer delivering 10.8 A/mm^2," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 2012, pp. 400–402.

[28] P.-T. Huang *et al.*, "2.5D heterogeneously integrated bio-sensing microsystem for multi-channel neural-sensing applications," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 2014, pp. 320–321.

[29] C. Song, L. Wang, Y. Yang, Z. Wang, W. Zhang, and L. Cao, "Robust and low cost TSV backside reveal for 2.5D multi-die integration," in *Proc. IEEE Electron. Compon. Technol. Conf.*, Jun. 2016, pp. 316–321.

[30] J. Howard *et al.*, "A 48-core IA-32 processor in 45 nm CMOS using on-die message-passing and DVFS for performance and power scaling," *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 173–183, Jan. 2011.

[31] S. Gondi and B. Razavi, "Equalization and clock and data recovery techniques for 10-Gb/s CMOS serial-link receivers," *IEEE J. Solid-State Circuits*, vol. 42, no. 9, pp. 1999–2011, Sep. 2007.

[32] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[33] J. F. Bulzacchelli *et al.*, "A 10-Gb/s 5-tap DFE/4-tap FFE transceiver in 90-nm CMOS technology," *IEEE J. Solid-State Circuits*, vol. 41, no. 12, pp. 2885–2900, Dec. 2006.

[34] J. R. Cubillo *et al.*, "Interconnect design and analysis for through silicon interposers (TSIs)," in *Proc. IEEE DIC*, Feb. 2012, pp. 1–6.

[35] V. Iverson, J. McVeigh, and B. Reese, "Real-time H.24-AVC codec on intel architectures," in *Proc. Int. Conf. Image Process.*, Oct. 2004, pp. 757–760.

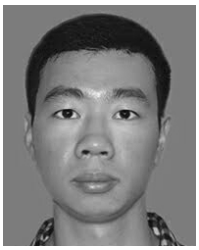
[36] T. Mori *et al.*, "A power, performance scalable eight-cores media processor for mobile multimedia applications," *IEEE J. Solid-State Circuits*, vol. 44, no. 11, pp. 2957–2965, Nov. 2009.

[37] J. Wang, W. Wang, J. Yang, Z. Yu, J. Han, and X. Zeng, "Parallel implementation of AES on 2.5D multicore platform with hardware and software co-design," in *Proc. IEEE Int. Conf. ASIC*, Nov. 2015, pp. 1–4.

[38] A. McCallum, K. Nigam, and L. H. Ungar, "Efficient clustering of high-dimensional data sets with application to reference matching," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2000, pp. 169–178.

Sai Manoj P. D. (S'13-M'17) received the M.S. degree from IIIT, Bangalore, India in 2012 and the Ph.D. degree from Nanayang Technological University, Singapore, in 2015. He is currently working in Vienna University of Technology (TU Wien) as a Postdoctoral Research Fellow. His research interests are SoC design, machine learning, Design of Self-aware ICs, 3D and 2.5D ICs, and low power system design. He is also recipient of A. Richard Newton Young Research Fellow Award in DAC 2013.

Jie Lin received the B.S. degree in electronic science and technology from Southeast University, Nanjing, China, in 2012, and the M.S. degree in microelectronics from Fudan University, Shanghai, China, in 2015. His research interests include high-performance and energy-efficient digital VLSI design with an emphasis on 2.5D integrated multi-core processors. He is currently a DSP Formal verification engineer in HiSilicon Technology (Shanghai) Ltd, China.



Shikai Zhu (S'15) received the B.E. degree in microelectronics from Fudan University, Shanghai, China, in 2015. He is currently working in Hisilicon, HUAWEI Technology Co., Ltd., Shanghai. His current research interests include multimedia and reconfigurable computing.

Yingying Yin received the B.S. degree in microelectronics from Fudan University, Shanghai, China, in 2013. She is currently working toward the M.A. degree in microelectronics at Fudan University. Her research interests include in-memory computing on 2.5D integrated multi-core processors and the implement of H.264 decoder on multi-core processors.

Xu Liu received the Ph.D. degree in electrical and computer engineering from National University of Singapore (NUS), Singapore, in 2015. He is currently a Research Fellow in Nanyang Technological University (NTU), Singapore. His research interests are biomedical IC design, biosensors, and mixed-signal IC design.

Xiwei Huang received the B.Eng. degree from Beijing Institute of Technology (BIT), China, in 2009, and the Ph.D. degree in electronic engineering from Nanyang Technological University (NTU), Singapore, in 2015. He is now an Associate Professor in the School of Electronics and Information, Hangzhou Dianzi University, China. His research interests include CMOS image sensor design, ISFET sensor design, and 3D/2.5D integration.

Chongshen Song, photograph and biography not available at the time of publication.

Wenqi Zhang, photograph and biography not available at the time of publication.

Mei Yan received the M.S. degree from Fudan University, China, in 1999, and the Ph.D. degree from University of New York at Stony Brook, NY, USA, in 2004. She was CMOS Image Sensor designer at Micron/Aptina, USA, from 2004 to 2010. From 2010 to 2015, she was a Research Fellow in VIRTUS IC design center of the EEE Department, Nanyang Technological University, Singapore. Currently, she is CMOS architect in Illumina San Francisco, CA, USA, working on next-generation CMOS-based DNA sequencers.

Zhiyi Yu (S'04-M'07-SM'16) received the B.S. and M.S. degrees in EE from Fudan University, China, in 2000 and 2003, respectively, and the Ph.D. degree in ECE from the University of California at Davis, CA, USA, in 2007. He was with IntellaSys Corporation, CA, USA, from 2007 to 2008. From 2009 to 2014, he was an associate professor in the Department of Microelectronics, Fudan University, China. Currently, he is an associate professor in the SYSU-CMU Joint Institute of Engineering, and a jointly appointed professor in the school of electronics and information technology, Sun Yat-sen University, China. He is also an adjunct professor of ECE department, Carnegie Mellon University, USA. His research interests include digital VLSI design and computer architecture. Dr. Yu serves on many conference committees, such as the ASSCC, VLSI-SOC, ISLPED, APSIPA, SASIMA.

Hao Yu (M'06–SM'14) received the B.S. degree from Fudan University, China, and the Ph.D. degree from the Electrical Engineering Department, University of California, Los Angeles, CA, USA. He is an Assistant Professor at the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His primary research interests are 3D-IC and RF-IC at nano-tera scale. He has 200 peer-reviewed IEEE/ACM publications and 4 books. Dr. Yu received Best Paper Award from the ACM TODAES'10, Best Paper Award nominations in DAC'06, ICCAD'06, ASP-DAC'12, Best Student Paper (advisor) Finalist in SiRF'13, RFIC'13, IMS'15, and Inventor Award'08 from semiconductor research cooperation. He is associate editor and technical program committee member for a number of journals and conferences.