An Automatic Chip-Package Co-Design Flow for Multi-core Neuromorphic Computing SiPs

Jingjing Lan, Vishnu P. Nambiar, Rheeshaalaen Sabapathy, Rahul Dutta, Chai Tai Chong, Mihai Dragos Rotaru, Kuang Kuo Lin, Surya Bhattacharya, Kevin Tshun Chuan Chai and Anh Tuan Do
Institute of Microelectronics, Agency for Science, Technology and Research
2 Fusionopolis Way, #08-02 Innovis Tower, Singapore 138634
Lan_Jingjing@ime.a-star.edu.sg

2020 IEEE 22nd Electronics Packaging Technology Conference (EPTC) | 978-1-7281-8911-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/EPTC50525.2020.9315089

Abstract

The complexity and cost of system-on-chip (SoC) designs keep increasing every year, which has progressively led to more opportunities for 2.5D System-in-Package (SiP) design. While 2.5D integration technology offers advantages for heterogeneous chiplet-based systems, it also poses challenges of a more complex overall design flow with limited EDA tools support, physical design optimization issues, interposer floor-planning difficulties and complex system-level verification. Furthermore, accurate inter-chiplet connection modeling, chiplet characterization and top-level simulation activities are also needed for comprehensive verification of SiPs. To tackle these challenges, we share an automated chiplet-based co-design flow: built on the backbone with standard EDA design tools, an automatic SiP register transfer language (RTL) generator is developed for top-level SiP netlist generation; inter-chiplet connection routing with chip assembly router and parasitic extraction tools; interposer design and bumps placement with the foundry defined redistribution layer (RDL).

Introduction

Moore's law has provided the blueprint for the scaling of CMOS transistors in the last 50 years, driving CMOS technology to the limit, allowing designers to pack billions of transistors in cm^2 silicon area running computation at speed up to TOPs/sec [1]. However, Moore's law has come to the end of the road as the dimension of each transistor gate approaches the size of a single atom. From a cost perspective, a reduction in the cost per transistor has not kept the pace which also meant that it is costlier to develop in advanced technology. Other practical limits like Dennard's scaling has also driven the design focus away, more towards multi-core and domain-specific hardware to drive next-generation compute-intensive engines like AI hardware [2].

In recent years, advanced packaging is becoming a viable option to create custom System-in-Package (SiP) solution as opposed to system-on-chip (SoC) solution which helps to reduce cost, lower risk and creates the perfect opportunity to segment systems into different technologies (cost/performance perspective consideration) and integrate as "chiplets" using 2.5D integration technology [3]. In a 2.5D IC design, multi-chips are integrated at package level to provide all necessary system functions. One interposer-based 2.5D IC design is illustrated in Fig. 1. On top of the package, an interposer is employed as a separate layer for the connection between chips. Each single SoC is called a chiplet and mounted on a silicon interposer. High-speed communication between chiplets is facilitated through the interposer.

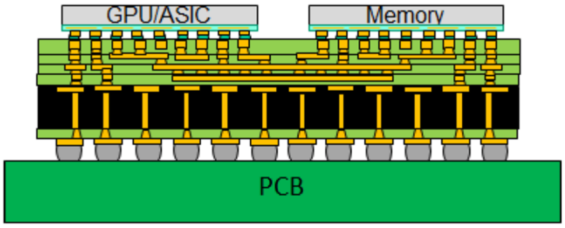

Fig. 1. Cross-section of a 2.5D system-in-package [4]

Comparing with 2D SoC design, 2.5D SiP design provides several advantages, including different technologies IP reuse and heterogeneous integration [5], the lower overall cost of development [6], and shorter time to market [7]. With 2.5D integration technology, different chiplets can be designed independently with various environments and schedules. The most suitable process technology can be used to implement each chiplet.

While 2.5D integration technology offers many advantages in heterogeneous system design, it also poses design challenges for the chiplet-based system, such as chiplet-based design flow, Electronic design automation (EDA) software tools support, chiplets floorplan on the interposer and chiplet-based system verification. Different approaches are proposed for 2.5D IC design methodology and physical implementation [5-8]. However, to the extent of our knowledge, there is no existing automatic design flow for many-core processing system with 2.5D integration technology. A flow with accurate inter-chiplet connection modeling, chiplets characterization and top-level analysis are desired to achieve comprehensive system verification.

In this paper, we present an automatic chiplet-based co-design flow for multi-core 2.5D SiPs. In this flow, the 2.5D SiP is designed using chiplets with the help of standard EDA design tools. An automatic SiP register transfer language (RTL) generator is developed for top-level SiP netlist generation. Manual connection for inter-chiplet wires is not necessary, which in turn reduces design and verification engineering effort. A multi-core 2.5D system comprising of multiple neuromorphic chiplets was built utilizing the proposed design flow. With this flow, shorter design time is required by the designers to implement the 2.5D system. The inter-chiplet connection with corresponding dimensions on the interposer is routing with chip assembly router and parasitic extraction tools. The bumps placement and interposer design is developed with the timing and power consumption information of the redistribution layer (RDL) defined by the foundry. With the electrical features, a more accurate analysis of inter-chiplet communication in the 2.5D system can be realized compared to previous works.

The rest of this paper is organized as follows. The proposed chiplet-based 2.5D design flow is introduced in the second Section. The third Section describes our automatic SiP netlist generator. The overall architecture of the neural computing 2.5D system is given in the fourth Section. The implementation results are presented in the fifth Section. Finally, the conclusions are summarized in the last Section.

Proposed Chiplet-Based Design Flow

Fig. 2 shows the proposed design flow for multi-core 2.5D systems. The system specifications and design environment are defined at the beginning of the flow. Then the whole system is partitioned into different chiplets to fulfill the system requirements.

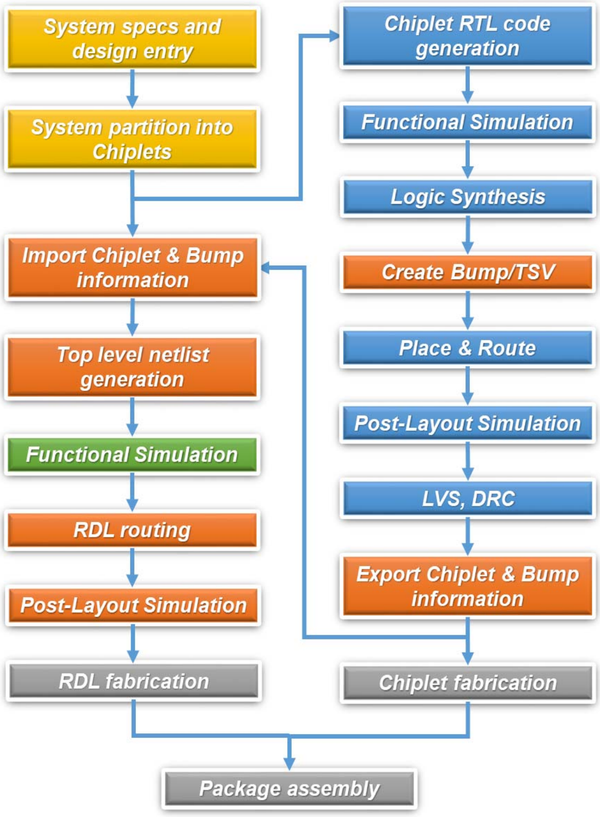

Fig. 2. Proposed design flow for multi-core 2.5D systems

The right side of Fig. 2 is the single-die chiplet design flow, which is similar to the traditional 2D design flow. The only difference is the information transfer of the chiplet and bumps.

After the chiplet RTL code is verified, the logic synthesis process is performed to generate a gate-level netlist. Before the specific chiplet physical implementation, the bump/ Through-Silicon Via (TSV) creation phase is required. Then the final chiplet physical layout is generated by the place and route EDA tools using this gate-level netlist. Once the chiplet layout passes the post-layout simulation and design rule checks (DRC), the final chiplet netlist and bumps information will be imported into the top-level SiP design.

The imported chiplet & bump information will be used to synchronize the interface between chiplets and interposer. The SiP netlist will be generated using our automated SiP RTL generator based on the number of chiplets to be integrated, which is needed to meet the end application requirements. The functional simulation of SiPs are quite similar to that of the traditional 2D systems, but executes at a higher hierarchy. The initial input of RDL routing includes SiP netlist, chiplet design files and interposer process design kit (PDK). Cadence EDA tools are employed to generate the interposer layout. Finally, the interposer and chiplets are sent to fabrication after the post-layout simulation.

Automatic SiP Netlist Generator

The proposed SiP netlist generator for a multi-core 2.5D system is described in this section. In traditional design flows, hierarchical chiplet schematics are represented by schematic symbols whereby all necessary inter-chiplet wires in top netlist are connected manually. In order to realize automatic chip-package co-design flow, one SiP RTL generator is built to reduce the extra effort needed for top-level netlist generation.

Algorithm 1: SiP Netlist Generating Algorithm

```
Load single chiplet core netlists
Load SiP schematic head and tail
Define interconnection for chiplets
Generate SiP netlist according to chiplet core number Chiplet
wireConnection = Interconnection wire between two Chiplets
tieHigh = Tie high signal for Chiplet
tieLow = Tie low signal for Chiplet
foreach c in Chiplet do
    foreach w in wireConnection do
        Route wire w in c
    foreach h in tieHigh do
        Tie high signal to power
    foreach l in tieLow do
        Tie low signal to ground
Create SiP netlist
Return Routing SiP netlist and design information
```

Algorithm 1 shows the complete process of the SiP netlist generation strategy. The single chiplet core is designed as a hard macro that contains I/O and power pins. With the schematic of the chiplet core, the top SiP schematic head file can be created. In the head file, the top package level I/O and power information are defined. The inter-chiplet wire information between two cores is given in the interconnection file. Finally, the connection between chiplet and top package level I/O is provided in the tail file. Once these files are ready, the many-core top SiP netlist can be generated automatically with our proposed SiP RTL generator.

First, the desired chiplet core number *Chiplet* and the design information is read into the SiP netlist generator. The head and tail files become the head and tail parts of the top SiP RTL netlist. Next, the tool will route all the interconnect wires *wireConnection* between chiplets. One search and connect strategy is employed to link the wires between chiplets in a loop. Only wires between two chiplet cores are routed at a time. Totally *w* wires are needed to connect all the pins between the two chiplets if there are *w* connections. After the wire connection, chiplet input signals which require a logic1 or logic0 value will be connected to the power or ground nets. As a result, the final SiP netlist is ready when the connection between the last two chiplets are routed.

Neural Computing 2.5D System

To demonstrate the proposed chiplet-based design flow, one multiple neural processing units (NPU) design is employed to build the multi-core 2.5D system [9]. The top-level diagram of the NPU system is provided in Fig. 3. As illustrated, the NPU consists of multiple neural computing cores (Neurocores) that are made up of specialized hardware blocks intended for accelerating neural network computations.

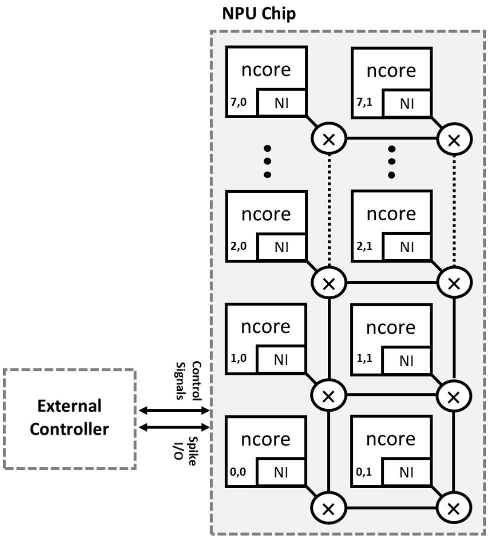

Fig. 3. NPU system top diagram

The top-level architecture of the multi-core NPU utilizing 2.5D SiP design is presented in Fig. 4. The interconnection between NPU chiplets is connected with wires on the RDL layer of the interposer. Each Neurocore of the NPU chip is accompanied by a dedicated router for communication, which is interconnected as a mesh [10]. The router port connection exits at the first and final Neurocore of the NPU for external control. They are designed to realize flexible one-to-one mapping of any logical neural network topology. Any Neurocore can be configured to run a spiking neural network operation, and then send its computation results to another Neurocore based on the mapping programmed into its routing lookup table (LUT).

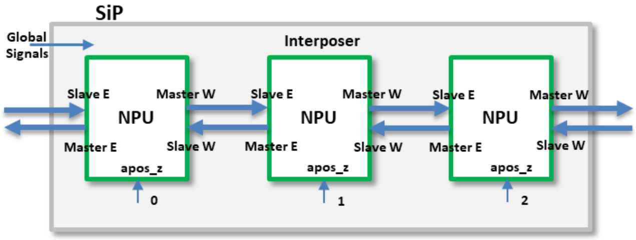

Fig. 4. Multicore NPU utilizing 2.5D SiP design

Implementation Results

The single NPU chiplet was designed using the traditional 2D flow. Then, the top-level SiP netlist header which consists of top-level signals (*clk*, *rst*, *master_e*, *slave_e*, *master_w*, *slave_w*) and power signals (*vdd*, *vss*) was created. The inter-chiplet wire information between NPUs (*master_w_0* and *slave_e_1*, *master_e_1* and *slave_w_0, master_w_1* and *slave_e_2*, *master_e_2* and *slave_w_1*) is provided in the interconnection file, which forms the body of the netlist. In the end of the netlist, the connection between chiplet and top package level I/O (*master_e_0*, *slave_e_0*, *master_w_2*, *slave_w_2*) is given. When all of these are combined alongside the NPU netlist, the multi-core top SiP netlist is generated automatically by our proposed SiP RTL generator. The simulation results of the generated 2.5D neural computing top netlist are present in Fig. 5.

In this flow, the top SiP design is routed using the chip assembly router tool with the PDK defined by the foundry. As shown in Fig. 6, the interposer has three RDL routing layers to ensure these connections are done seamlessly without any congestion. The electrical, signal and power integrity feature of the RDL are generated and characterized by the foundry. With these parametric data, the timing information of each design cell and the path is extracted and stored in a Standard Delay Format (SDF) file for post-layout simulation after the place and route of the layout. Top-level post-layout simulation with back-annotation is then performed utilizing the SDF file.

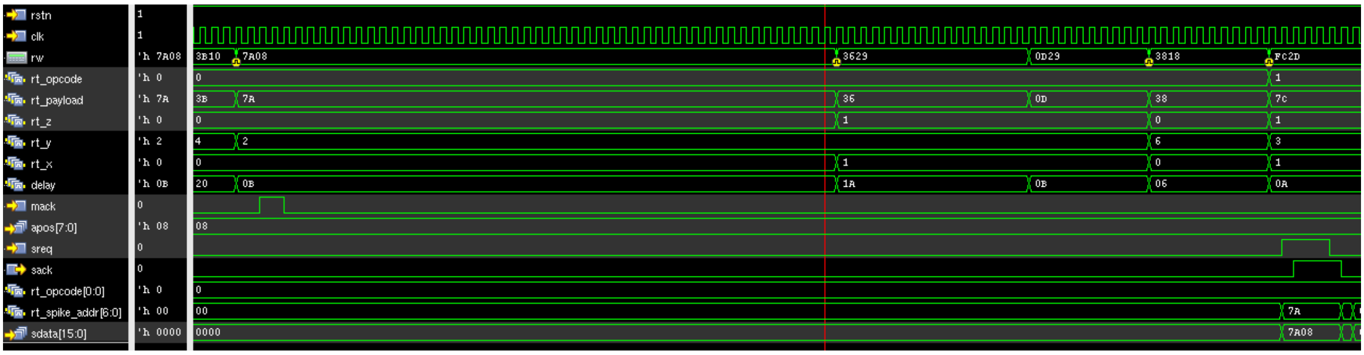
rstn 1
clk 1
rw 'h 7A08 3B10 7A08 3629 0D29 3818 FC2D
rt_opcode 'h 0 0 1
rt_payload 'h 7A 3B 7A 36 0D 38 7C
rt_z 'h 0 0 1 0 1
rt_y 'h 2 4 2 6 3
rt_x 'h 0 0 1 0 1
delay 'h 0B 20 0B 1A 0B 06 0A
mack 0
apos[7:0] 'h 08 08
sreq 0
sack 0
rt_opcode[0:0] 'h 0 0
rt_spike_addr[6:0] 'h 00 00 7A
sdata[15:0] 'h 0000 0000 7A08

Fig. 5. Simulation results of neural computing 2.5D design

No timing violations are reported by the simulator during the post-layout simulation of our design.

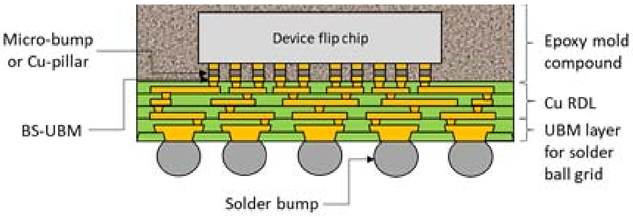

Fig. 6. Cross-section of RDL with three routing layers

In this design, all functional bumps are placed at the peripheral of the chiplets with a distance of 55 microns between each bump as illustrated in Fig. 7. The distance between the two bare dies is only 500 microns. Due to the increasing dimension compared with 2D chip design, the major design challenges encountered were physical verification issues, timing and DRC violations.

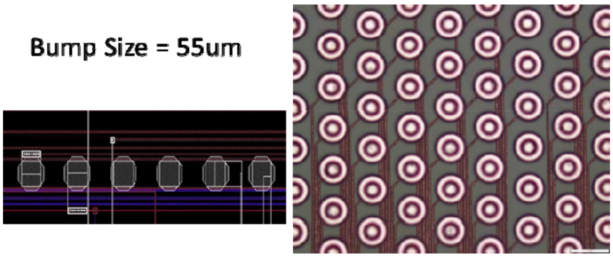

Fig. 7. Bump placement of neural computing 2.5D design

Fig. 8 shows the placement of the bumps and the routing wire for the interconnection respective to the neural computing chiplet bare dies. These wires describe the routing connections performed by the chip assembly router tools.

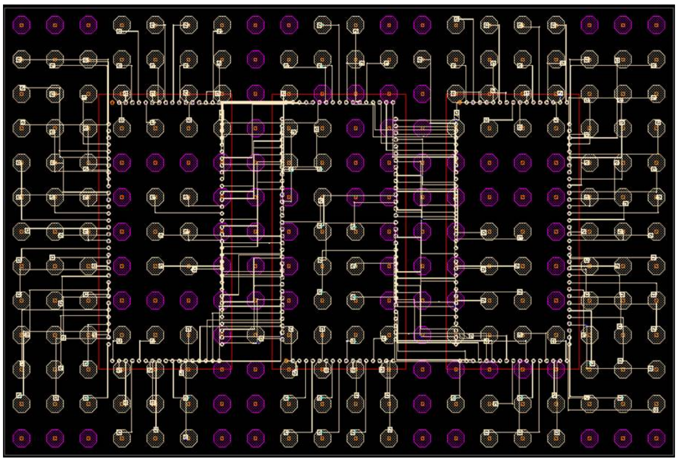

Fig. 8. Layout view of neural computing 2.5D design

Conclusions

In this paper, an automatic system-in-package co-design flow for a multi-core 2.5D system was presented. A multi-core 2.5D system comprising of multiple neuromorphic chiplets was built with the proposed design flow. An automatic SiP RTL generator is developed for top-level netlist generation, helping to reduce design time when implementing the resulting 2.5D SiP.

Acknowledgments

This work is the result of the Chiplet core project supported by the Institute of Microelectronics (IME), Agency for Science, Technology and Research (A*STAR), Singapore. The authors would like to appreciate the support and assistance of the project members as well as the Integrated Circuits and Design and CAD group of IME who had contributed and made this work successfully.

References

1. C. A. Mack, “Fifty years of Moores law”, *IEEE Transactions on Semiconductor Manufacturing*, Vol. 24, No. 2, pp. 202-207, May 2011.
2. R. H. Dennard, F. H. Gaensslen, H.-N. Yu, V. L. Rideout, E. Bassouse and A. R. LcBlanc, "Design of ion-implanted MOSFETs with very small physical dimensions", *IEEE Journal of Solid-State Circuits*, Vol. 9, No. 5, pp. 256-268, 1974.
3. M. LaPedus, “The Good And Bad Of Chiplets”, *Semiconductor Engineering*, May 27, 2020.
4. Hsiao H. Y., Ho S. W., S. B. Lim, S. C. Chong, S. Lim and Chai T. C., “Process Development of Fan-Out interposer with Multi-layer RDL for 2.5D System in Package”, *2018 IEEE 20th Electronics Packaging Technology Conference (EPTC)*, pp. 406-410, December 2018.
5. A. Coskun, F. Eris, A. Joshi, A. B. Kahng, Y. Ma and V. Srinivas, “A Cross-layer Methodology for Design and Optimization of Networks in 2.5D Systems”, *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, No. 101, pp. 1-8, November 2018.
6. A. Kannan, N. E. Jerger and G. H. Loh, “Enabling Interposer-based Disintegration of Multi-core Processors”, *Proc. 48th International Symposium on Microarchitecture (MICRO)*, pp. 546-558, December 2015.
7. D. Stow, I. Akgun, R. Barnes, P. Gu, and Y. Xie. “Cost analysis and cost-driven IP reuse methodology for SoC design based on 2.5D/3D integration”, *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1-6, November 2016.
8. A. Fontanelli, “System-in-Package technology: Opportunities and challenges”, *Proc. 9th International Symposium on Quality Electronic Design (ISQED)*, pp. 589-593, March 2008.
9. J. Pu, V. P. Nambiar, A. T. Do and W. L. Goh, “Block-based spiking neural network hardware with deme genetic algorithm”, *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1-5, May 2019
10. J. Pu, W. L. Goh, V. P. Nambiar and A. T. Do, “A Low Power and Low Area Router with Congestion-Aware Routing Algorithm for Spiking Neural Network Hardware Implementations”, *IEEE Transactions on Circuits and Systems II: Express Briefs*.