Chiplets: How Small is too Small?

Alexander Graening, Saptadeep Pal, Puneet Gupta
Department of Electrical and Computer Engineering
University of California, Los Angeles
agraening@ucla.edu, saptadeep@ucla.edu, puneet@ee.ucla.edu

***Abstract*—As chiplet systems increase in popularity, it is important to revisit the tradeoffs for converting a monolithic design to a chiplet system. Chip yield, reusability, performance binning, and floorplanning push us toward smaller chiplets. Meanwhile, inter-chiplet interconnect and assembly overheads push us toward larger chips both in terms of power and cost. This work explores the impacts of these considerations on the *minimum* chiplet size that makes sense. We examine the case of a large design that could be built as a single monolithic system on chip (SoC) or as a system of chiplets and show that optimal chiplet size depends on a wide range of parameters. Our analysis indicates that the smallest chiplet sizes that are viable cost-wise depends both on technology node and on type of logic. The optimal point appears to be 50-150mm^2 in 40nm and 40-80mm^2 in 7nm for microprocessor type logic. For random logic, the optimal point increases beyond 200mm^2 in both cases. This makes the case for chipletization weaker in all but the largest SoCs.**

I. Introduction

Chiplet systems have become popular for systems that would otherwise have unacceptably low yield due to their large total area. This trend runs counter to the previous trend of increasing integration by packing more into a single monolithic chip.

For large designs, splitting into smaller chips has several advantages. Small chips have higher yield than large chips. Chiplet reuse in the form of including a single chiplet design in multiple systems allows sharing of design, manufacturing, and testing costs across systems. If a chiplet contains too much functionality, it can become specialized to a specific design, and the opportunities for reuse are limited[1][2].

There are also drawbacks to splitting a design into chiplets. Assembly and packaging for a design that has been partitioned into chiplets are more costly than they are for a monolithic SoC design. In addition to the costs related to the integration substrate (e.g. silicon interposer) and greatly increased assembly time, yield in the assembly and packaging stage suffer due to the increased number of fine-pitch bonds that must be made and number of individual chiplets that must be assembled. Also, the total silicon area would go up in order to accommodate the additional IO cells needed for inter-chiplet communication. This inter-chiplet communication also increases the overall power consumption of the design.

In this work, we build an analytical framework to answer the question: what is the right chiplet size when the multitude of manufacturing related factors are considered? We consider a case of a design that is small enough to be manufactured as a monolithic SoC, but is large enough to benefit significantly from splitting into chiplets to improve yield. Our analysis shows that building systems out of tiny chiplets or chiplets substantially smaller than 40mm^2 would likely not be cost optimal unless advancements to the assembly process are made.

Section II models the cost benefits/overheads of breaking an SoC into chiplets. Section III discusses a case study of a large system built using chiplets and analyzes the sensitivity of system cost to factors such as defect density, assembly cost, IO size etc. In Section IV, we discuss the other factors (which are often architecture dependent and difficult to quantify in general) that would affect the choice of chiplet size. Finally, Section V concludes the work.

We try to cover a range of parameters to keep our analysis general, but since the conclusions and analyses in this paper are somewhat design dependent, we are releasing our model at https://github.com/nanocad-lab/cost_model_chiplets.git to assist in further studies.

II. Cost Implications of Chipletization

In this section, we quantify and model some of the most important factors that affect the cost of manufacturing a system using chiplets.

A. IO Cells

Splitting a design into chiplets introduces the requirement for inter-chiplet communication. This adds overhead in terms of area and power requirements for interconnect wires and IO cells to provide ESD protection and drive relatively long connections between chiplets [3].

1) ESD Requirements: The amount of ESD protection necessary for 2.5D designs is not a settled question, but there is some consensus that the "external" IO cells that will be connected to external pins on the finished module will need higher levels of protection than "internal" IO cells that will not be connected to external pins, drive only inter-chiplet wires and will only be exposed to ESD in the manufacturing process. For external IO cells, JEDEC recommendations suggest meeting 250V Charged Device Model (CDM) protection [4]. On the other hand, a TSMC discussion of a Lite-IO cell design for inter-chiplet communication only uses 10V CDM protection [5]. Whether ESD protection is necessary and how much is necessary for inter-chiplet connections depends on process specifications, so we look at a range of levels of protection from 0V to 500V CDM.

2) Area Impact of ESD: Due to ESD protection, the total design area will increase due to the IO cell area for inter-chiplet connections in addition to the overhead due to minimum separation distance between chiplets. In order to determine the IO cell size necessary to meet different levels of ESD protection, we ran SPICE simulations using the circuit shown in Figure 1. The clamp is assumed to be shared among many IO cells, so this is not included in IO cell area calculations.

This work was supported in part by CHIMES, one of the seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA, ASML and the CDEN center. We would also like to thank Youngtag Woo and Pieter Woltgens from ASML for technical discussions which guided the direction of this work.

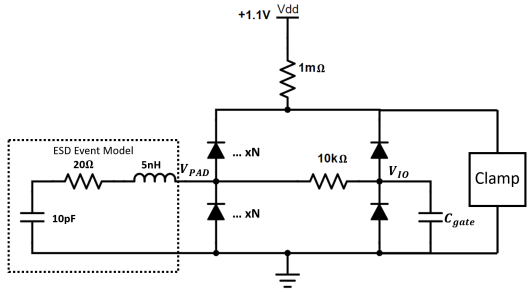

Fig. 1. ESD Simulation Structure. The number of diode pairs connected V_{PAD} are swept to find the number necessary to keep V_{IO} below the breakdown voltage of the gate dielectric. The ESD event model is from [8].

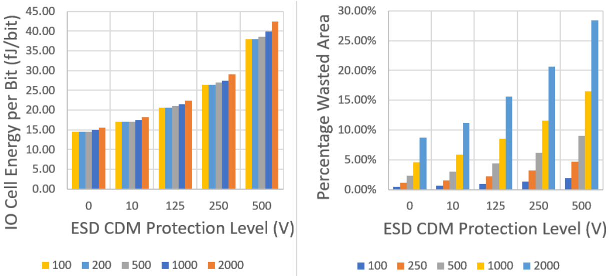

Fig. 2. Left: Extra Energy per Bit for Inter-Chiplet Communication for Different IO Densities. Right: Percentage of Area Consumed by IO Cells for Different IO Densities. Both: IO densities range from 100 IOs/mm^2 to 2000 IOs/mm^2. For reference, this corresponds to maximum IO pad pitch ranging from 100um to about 22um. A VDD value of 1.1V was used for computing power.

By setting different initial voltages for the capacitor in the above model, we tested different levels of CDM protection. For the purposes of this study, the device was considered to pass if the voltage never passed the voltage breakdown level of the gate oxide. Other types of incremental damage can also occur as a result of ESD events [6], but were not considered here as we wanted a simple metric of pass or fail. Buffer sizes were taken from [7]. Results are shown in Table I.

As a design is partitioned into more and more chiplets, the area requirement for IO cells increases substantially since the increasing number of inter-chiplet interconnects need ESD protection and larger driver circuits.

TABLE I

AREA REQUIREMENTS FOR IO CELLS BY ESD PROTECTION LEVEL IN 40NM. NOTE THAT THESE IO CELL SIZES ARE FOR SIMPLE IO CELLS DESIGNED FOR INTER-CHIPLET CONNECTIONS. EXTERNAL IOS WOULD USE MUCH LARGER IO CELLS, E.G. GPIO CELLS IN 28NM OF 3,250μM^2 [9] AND IN 12NM THAT ARE 1,500μM^2 [10] ARE ROUTINELY USED IN PRODUCTS.

		Area (μm^2)			
ESD(V)	Diode Pairs	ESD	Buffer	IO Cell	ESD Cap(fF)
0	0	0.00	47.63	47.63	0.00
10	2	15.02	47.63	62.65	8.02
125	6	45.07	47.63	92.70	18.97
250	11	82.62	47.63	130.25	37.65
500	20	150.22	47.63	197.85	74.97

To compute the impact of IO cells on area, IO densities are scanned from 100-2000 IOs/mm^2 across multiple levels of ESD protection. Results are shown in Figure 2.

Inter-chiplet IOs also cost extra energy in order to drive the inter-chiplet interconnect wire and the additional capacitance added by the ESD diodes. To compute energy per bit, we used wire parasitics from a commercial 40nm PDK and the ESD diode capacitance. The interconnect wirelength was estimated from the inter-die separation (assumed to be 300μm) and area of the IO cells assuming multiple rows of boundary placed IO cells, similar to the methodology in [3]. We assume added wirelength from the logic to the boundary-placed IO cell and back to the pads which are assumed to be placed in a grid pattern across the full chip area. The cost in energy per bit is shown in Figure 2.

B. Chiplet Cost

Individual chiplet cost is dependent on the individual chiplet yield. Taking k_{die} as the cost per untested die, the actual cost per die is given by Equation 1.

$$C_{die} = \frac{k_{die}}{Y_{die}} \tag{1}$$

The cost per untested die (k_{die}) is dependent both on the area of the die and on how well the die fits into the reticle size. The impact of utilization of the reticle size is negligible for small chiplets, but can have a larger effect for chiplets that are relatively large compared to the reticle. This cost is the result of needing an increased number of lithographic exposures to manufacture dies that do not evenly divide into the reticle field [11] see Figure 3. The cost per untested die is given below in Equation 2.

$$k_{die} = k_{silicon} A_{chip} + \frac{k_{exposures}}{\left\lfloor \frac{A_{reticle}}{A_{chip}} \right\rfloor} \tag{2}$$

Where $k_{silicon}$ is the cost per unit area independent of exposure cost, and $k_{exposures}$ is the cost per exposure. We assume the lithography cost is 34% of the total wafer cost [12].

Die yield can be given by Equation 3. This yield model is taken from [13], but split into two components: die yield and assembly yield.

$$Y_{die} = Y_{wp} \left(1 + \frac{A_c D_0}{\alpha}\right)^{-\alpha} \tag{3}$$

Where Y_{wp} is the wafer process yield assumed to be 94% [13]. A_c is the critical area. For the purposes of this study, we use the core area (excluding IO area) as critical area. For a more accurate analysis of a real design, the actual critical area of the design should be computed and used here instead of the core area. D_0 is the defect density. We use a value of 0.004/mm^2 from [14] and also look at defect densities of 0.002/mm^2 and 0.0007/mm^2 to get a range of values. For reference, 0.004/mm^2, 0.002/mm^2, and 0.0007/mm^2 correspond to yields of approximately 70%, 83%, and 93% respectively. α is the clustering factor for defects and we use a value of 2 [14][15].

In Figure 3, one can see that the cost per mm^2 has somewhat of a stepwise behavior depending on how well the chip fits in the reticle size. For this plot, we did not make any assumptions about the aspect ratio of the chips and just considered how well the chip size divides into the reticle. The general trend is due to yield decreasing as chip size increases and the steps are due to the chip size increasing above a number that divides evenly into the 858mm^2 (26x33mm) reticle.

C. Assembly Cost

Assembly time and cost will increase with the number of chiplets. How this scales is dependent on the bonding process. There are two main models we will look at here. In one case, the chips are placed and bonded individually (e.g. in the case of copper pillar thermal compression bonding) giving an assembly time as follows.

$$T_{assembly} = N(T_{place} + T_{bonding}) \tag{4}$$

On the other hand, it is possible that multiple chips will be placed and tacked individually on the interconnect substrate, but finally bonded at the same time as is the case for solder reflow.

$$T_{assembly} = NT_{place} + T_{bonding} \tag{5}$$

In the second case, bonding scales significantly better than in the first case, but the time for pick and place still scales with the number of chips. Interposer cost is discussed in [16]. The cost of assembly can be modeled by looking at materials cost and machine operating cost as shown below.

$$C_{assembly} = C_{interposer} + k_{machine} T_{assembly} \tag{6}$$

The machine operating cost of assembly for the machine or machines used consists of the amortized cost of the equipment and maintenance over the equipment lifetime plus the cost of electricity and technician salaries. To compute values for $k_{machine}$, we assume machine costs of $200k-$2M depreciated over 5 years with full-time personnel costs of $200k per year and 90% uptime with a single bonding head.

Bonding time for Cu-Cu pillar bonding is 20 seconds [17] and hybrid bonding is shorter at 10 seconds [18]. Since hybrid bonding and Cu-Cu pillar bonding offer low pitch interconnects (down to 10um [19][17]), we do not consider other types of bonding for this study. Pick and place time is assumed to be somewhere from 2-10 seconds per die depending on the required precision of placement [20], although this will vary based on required bonding precision. In Figure 4, we assume Cu-Cu pillar bonding and 10 seconds for pick and place, but the trend holds for other types of bonding as well. The interposer cost is taken from [16].

The values for assembly time and machine costs vary linearly with respect to the number of chiplets. Assembly yield is another consideration for the cost of assembly. The assembly yield scales with the number of interconnects and the number of chiplets.

$$Y_{assembly} = Y_{alignment}^{N_{die}} \times Y_{pins}^{N_{pins}} \tag{7}$$

In Equation 7, the first term is the yield of die alignment and the second is the pin bonding yield. The more dies there are in the multi-chiplet system, the higher the risk of misalignment and the more pins that need to be bonded, the higher the risk that a pin will not bond correctly. Since we are not making assumptions about IO density yet, Figure 4 does not include yield although we include assembly yield in our analysis in the next section. We assume values of 99.9% for $Y_{alignment}$ and 99.9999% for $Y_{bonding}$ [21].

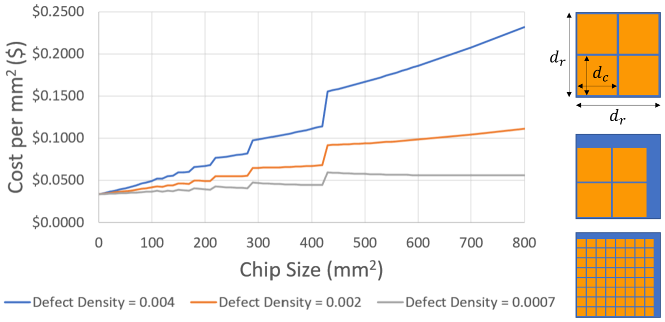

Fig. 3. Silicon Cost for Different Chiplet Sizes and Defect Densities for 858mm^2 (26x33mm) Reticle and Reticle Fit Examples. The three examples on the right (top to bottom): an example of a large chiplet design that fits evenly into the reticle size; an example of large chiplets that fit poorly into the reticle size; and an example of a poorly fitting small chiplet. Overhead for a bad fit is lower for small chiplets.

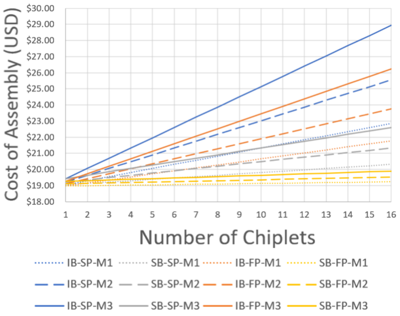

Fig. 4. Assembly Machine and Personnel Cost by Number of Chiplets. IB stands for individual bonding, SB stands for simultaneous bonding, M1 refers to $200,000 machine, M2 refers to $1,000,000 machine, and M3 refers to $2,000,000 machine. FP stands for 2-second pick and place, SP indicates 10 seconds. This plot uses Cu-Cu pillar bonding time (20 seconds) in all cases.

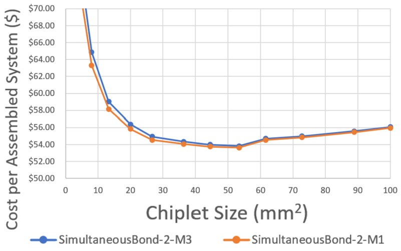

Fig. 5. Simultaneous Bonding for 2 Second Pick and Place Time

III. Results

In this section, we discuss the results of applying the above analysis to an example case where we take an 800mm^2 chip and split it into evenly sized chiplets.

The cost may be found using Equation 8.

$$C = \frac{1}{Y_{assembly}} \left[\left(\sum_{i=1}^{N_c} C_{die} \right) + C_{assembly} \right] \tag{8}$$

Where C is the total cost and both $Y_{assembly}$ and C_{die} are given in the previous sections. There is one more piece we need to be able to apply this analysis and that is the number of IOs per chiplet. To do this, we use Rent's Rule as shown in Equation 9 [22].

$$N_p = kN_g^a \tag{9}$$

Here N_p is the number of pins or IOs, N_g is the number of gates in the design, and k and a are constants that depend on the architecture. Rent's Rule gives us a way of estimating how the number of IOs scales with different sized chips assuming the chips are the same types of design. Although the Rent's Rule constants are design specific, this allows us to draw some useful conclusions.

To estimate the number of gates in the design we divided the area by the size of an AND gate in 40nm. We use two sets of constants, the primary set of constants we used is given in [23] as values of the constants for a microprocessor architecture and we compare to the values given in the same paper for random logic in the comparison for levels of ESD protection below. We chose these values as the random logic case is a pessimistic case of splitting the design and the microprocessor estimates are a somewhat optimistic case. Splitting a design at IP boundaries that are more interconnect intensive than a microprocessor will likely result in values somewhere between these two cases.

Unless otherwise stated, the following plots us medium defect density (0.002/mm^2), 125V CDM ESD protection, microprocessor Rent's Rule constants, and M2 ($1M machine) with 30 second individual bonding of chiplets in accordance with Equation 4.

A. Dependence on Defect Density

Defect density affects the rate at which yield decreases when chip size increases. In Figure 6, it can be seen that there is a relatively smooth trend in cost for each defect density level. Wafer cost is from [24]. For 40nm, the chiplet size that minimizes cost ranges between 50mm^2 and 160mm^2. Note that for the low defect density, large chiplets only result in a moderate cost increase over the optimal point. In all cases, the assembly cost begins to dominate for very small (and therefor numerous) chiplets. *Improving die yield allows manufacturing of both larger monolithic designs and larger chiplets. Conversely, low yield technologies can justify smaller chiplet.*

B. Dependence on Assembly Cost

Assembly cost has a substantial impact on the small, numerous chiplet side of this study. To examine the impact on minimum chiplet size, we held ESD and defect density constant and changed the machine cost and bonding time. We used high defect density M1 and M3 refer to the same $200k and $2M machines described earlier. Individual bonding was assumed for this analysis, but the combined time required for the pick and place and bonding ranges from 10 seconds to 30 seconds. In this plot, the minimum cost values range between chiplet size of 50mm^2 and 100mm^2. The trend seen in Figure 7 is that *faster and cheaper bonding allows for a larger number of smaller chiplets*.

If we can do simultaneous bonding for high density interconnect pins and fast pick and place, this will improve the assembly costs and reduce the overhead of assembly. If this is 20 seconds of bonding one time and 2 seconds of pick and place for each chiplet, we get the results in Figure 5. For faster and cheaper assembly, smaller (more numerous) chiplets become feasible.

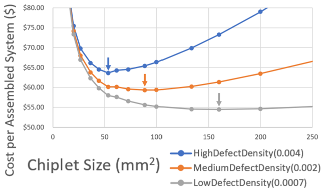

Fig. 6. Cost for Different Defect Densities at 40nm. Defect densities marked in the legend are in units of defects/mm^2. This uses the Rent's Rule constants for microprocessor logic. The minimum points are marked with arrows.

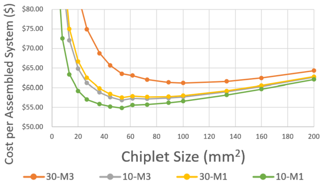

Fig. 7. Costs for Different Bonding/Pick and Place Times and Machines. M3 is a $2,000,000 machine and M1 is a $200,000 machine. The number represents the number of seconds for individually placing and bonding each chiplet.

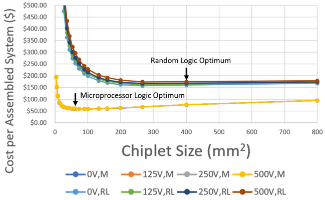

Fig. 8. Cost for Different Levels of ESD Protection and Rent's Rule Constants. M refers to the Rent's Rule constants for microprocessor circuits, and RL refers to Rent's rule constants for random logic [23].

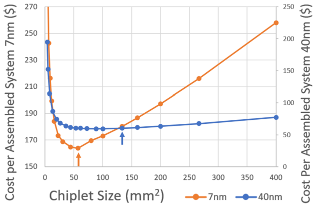

Fig. 9. Cost for Assembled System in 40nm and 7nm. Optimal points are marked with arrows.

C. Dependence on IO Size and IO Density

The cost of IO cells can substantially affect the optimal point for chiplet size and the overall system cost. This necessarily is dependent on level of ESD protection and the IO density. In Figure 8, we look at different levels of ESD protection for two different sets of Rent's Rule constants: one for microprocessor logic and one for random logic. For the first set of Rent's Rule constants, the difference between different levels of ESD protection is negligible, but for the second set of Rent's Rule constants, the higher IO density means ESD and assembly yield has a larger impact. *Inter-chiplet interconnect increases area requirements and lowers assembly yield, so good chiplet designs should prioritize splitting at logical boundaries that minimize inter-chiplet IO.*

D. Comparison with Different Technology Node

Most of this analysis has been done with numbers from the 40nm node. To show how this scales for more advanced nodes, we compared to 7nm. In Figure 9, you can see that the minimum point for 7nm is smaller than for 40nm and 40nm has less penalty for a nonoptimal chiplet size than 7nm does. Note that since ESD does not scale well between nodes, we used the same ESD area for 7nm and 40nm. The buffer size was reduced for 7nm to reflect increased drive strength of transistors in 7nm.

E. Inter-chiplet Communication Power

Increased power consumption is an additional cost of chipletization that does not neatly fit into the cost metric described above. Figure 10 shows the additional power consumed due to the capacitance added at chiplet boundaries in the form of ESD protection and top level wires. As can be seen, *for highly connected designs (the random logic case), small chiplets (<100mm^2) may be energy-wise unaffordable.*

Harsher ESD requirements exacerbate the problem. Good chiplet partitioning approaches which minimize inter-chiplet connectivity can help alleviate the energy overheads.

IV. Other Chiplet Size Tradeoffs

A. Floorplanning Overhead

In a design consisting solely of "hard" IP blocks (blocks with a fixed layout), there may be some wasted area after packing the IP blocks into a rectangular chip. If we consider SoC floorplanning as a rectangle packing problem [25], packing all these IP blocks into a single monolithic design will result in less wasted area than packing the IP blocks into several smaller chiplets that each contain fewer IP blocks. For reference, the optimal solutions to the Consecutive Squares benchmark for 2-10 squares contain between 2.86% and 16.7% empty space while the optimal solutions for more than 20 blocks all contain less than 1% empty area. This indicates that the packing problem will result in more inefficiency for smaller chiplet sizes when using "hard" IP blocks.

This effect is less pronounced when using "soft" IP blocks or a mix of "soft" and "hard" IP blocks, since this allows more flexibility in floorplanning and more efficient packing of IP blocks. This case still benefits from larger groups of IPs since smaller groups will increase the likelihood of chiplets with significantly different sizes that can be difficult to fit together efficiently without wasted area on the interposer.

Since this effect is difficult to quantify for real designs we did not directly consider it in the above analysis. It is important to keep in mind however that wasted area is minimized by including everything in a single SoC as this gives the greatest flexibility of floorplanning and eliminates minimum chiplet separation distances.

B. Test Cost

The testing process is impacted by splitting a monolithic design into multiple chiplets. On one hand, smaller chips provide the opportunity for more fine-grained tests. This can improve the quality of the final product [23]. Although the chips will be better tested, testing time goes up since chips will be tested individually before being assembled on the interposer and likely again after assembly to ensure assembly did not introduce any errors.

C. Non-Recurring Engineering Cost

Non-Recurring Engineering (NRE) costs include costs such as masks and tooling that occur regardless of manufacturing volume. In this analysis, we have assumed a high manufacturing volume so these costs can be small. For low volume manufacturing, NRE costs can become a significant factor making smaller chiplets less cost-effective (due to a greater number of masks) unless chiplets can be designed to be reusable across a wide range of designs. For a discussion on mitigating NRE costs for low-volume chips by using general purpose reusable chiplets that can be produced in high volume for use in many different designs see [2].

D. Reuse of Chiplets

As discussed in [26] and [1], chiplet reuse scales with the size of chiplets and is impacted by the type of chiplets. A heterogeneous chiplet system can be well-suited to chiplet IP reuse across separate designs. Ultimately, the development time and cost of future designs can be reduced if chiplets are thoughtfully designed as functional blocks that are useful across many systems. Predicting and modeling reuse patterns is difficult and is out of scope for this work. For more discussion on the benefits of reuse, see [2].

E. Heterogeneous Chiplets

One possible application of chiplet systems is the possibility of mixing different technology nodes by integrating different technology chiplets on the same interposer. This could have cost and performance advantages since it would allow producing performance-critical chiplets in an advanced node and non-performance-critical chiplets with low switching activity in an older node. This could potentially allow the non-performance-critical chiplet to be less expensive, more power efficient due to reduced leakage, and easier to reuse across different versions of the system to reduce design cost. This is a very broad design-space to explore and is a good direction for future work.

F. Cost-Aware Partitioning

The models in this paper can be used to estimate costs for specific chiplet parameters and could be used to help inform cost-aware partitioning in a CAD tool in the future. An important consideration here is that inter-chiplet interconnect requirements can become expensive for poorly chosen partitioning (random logic in Figure 8), so a smart cost-aware partitioning algorithm would likely result in partitions that are closer to the microprocessor interconnect density since the microprocessor Rent's Rule constants assume a well-chosen chip boundary.

G. Delay-Aware Chiplet Yield

In many-core homogenous systems, the entire system must run faster than a certain minimum frequency or be considered defective. This means that for a monolithic system, the entire chip will be considered defective if a single core falls below this threshold. Testing chiplets and only using known good dies in the assembly improves yield. Yield can drop off quickly for chiplets containing many cores depending on the maximum delay distribution as shown in Figure 11, left. This study comes with several caveats as performance binning, adaptive voltage scaling, multiple clock domains, etc. all can help mitigate this performance-limited yield issue (albeit with other power or cost overheads). Here, we assume a Gaussian distribution of maximum delay [27].

Figure 11, right shows a similar plot to those shown in Section III. The default parameters are the same as described in that section, but the performance yield is added in. Note that this assumes a homogeneous many-core system that contains 1mm^2 cores, so an 800mm^2 chip contains 800 cores. If this is split into 4 chiplets, each 200mm^2 chiplet contains 200 cores.

V. Conclusion

Our 800mm^2 design study seems to indicate that the best size for chiplets is somewhere between 50mm^2 and 150mm^2 for microprocessor logic and above 200mm^2 for random logic. Splitting a large design into chiplets improves yield substantially at first, but eventually, the cost and yield loss due to assembly begin to dominate. Fast, low-cost, high-yielding assembly methods may help make many-chiplet systems practical just as high-yield die manufacturing processes can help make larger chips more practical. Although these two factors seem to have the greatest impact on the ideal chiplet size, it is also important to consider

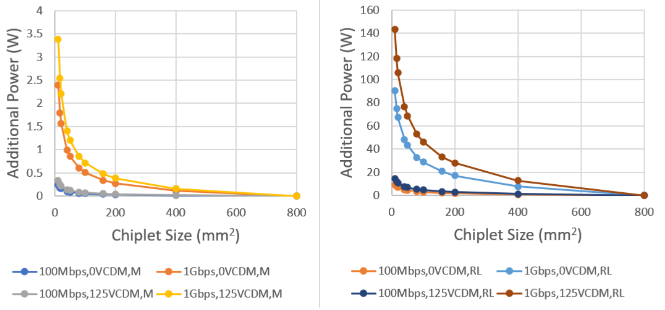

Fig. 10. The Additional Power Consumed by Added Wire and ESD Capacitance. Since this measures additional power due to inter-chiplet interconnect, at chiplet size of 800, there are no internal wires and additional power is 0. The plot starts at chiplet size of 10mm^2. M stands for microprocessor and RL for random logic Rent's Rule constants. We analyze two cases of inter-chiplet communication intensity: 100Mbps per link and 1Gbps per link. For 4GHz inter-chiplet communication link, 1GBps amounts to a link activity factor of 25%.

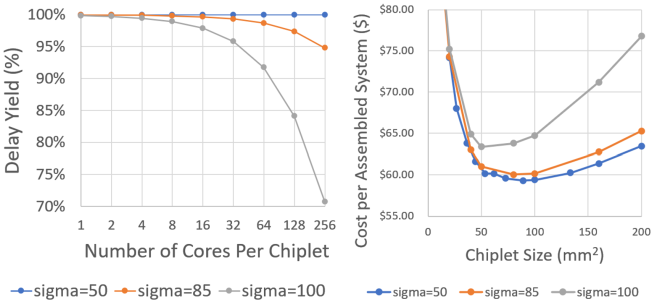

Fig. 11. Left: Yield with Number of Cores Per Chiplet. The delay distribution for cores is assumed to be Gaussian with a mean of 1000ps and sigma as marked in the legend. This plot considers delay values above the threshold to fail. Right: Cost of Assembled System with Performance-Aware Yield for 40nm.

inter-chiplet IO requirements as splitting a design introduces the need for additional drivers for longer connecting wires along with ESD protection that increase both area and power usage that do not scale well with technology node. This impact can be minimized by assuring that manufacturing processes use ESD controls to reduce or eliminate the need for ESD on inter-chiplet IO cells.

For brevity, the case studies in this paper are limited and the models are general. Larger or smaller SoC sizes, mixed-size chiplet partitioning, models of NRE costs, and models of chiplet reuse are interesting future directions. Large "minimum economically viable" chiplet sizes have implications for a future chiplet-based design ecosystem. 50+mm^2 is a large real estate in advanced technology nodes making reusable chiplet IPs challenging. Our ongoing work is investigating these chiplet ecosystem challenges.

References

[1] S. Pal, D. Petrisko, R. Kumar *et al.*, "Design space exploration for chiplet-assembly-based processors," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 4, pp. 1062–1073, 2020.

[2] P. Ehrett, T. Austin, and V. Bertacco, "Chopin: Composing cost-effective custom chips with algorithmic chiplets," in *2021 IEEE 39th International Conference on Computer Design (ICCD)*, 2021, pp. 395–399.

[3] S. Pal and P. Gupta, "Pathfinding for 2.5d interconnect technologies," in *2020 ACM/IEEE International Workshop on System Level Interconnect Prediction (SLIP)*. IEEE, 2020, pp. 1–8.

[4] *Recommended ESD-CDM Target Levels, JEDEC, JEP157A*, JEDEC JEP157A, 2022.

[5] Y.-K. Cheng, F. Lee, M.-F. Chen *et al.*, "Next-generation design and technology co-optimization (dtco) of system on integrated chip (soic) for mobile and hpc applications," in *2020 IEEE International Electron Devices Meeting (IEDM)*, 2020, pp. 41.3.1–41.3.4.

[6] A. Amerasekera, W. van den Abeelen, L. van Roozendaal *et al.*, "Esd failure modes: characteristics mechanisms, and process influences," *IEEE Transactions on Electron Devices*, vol. 39, no. 2, pp. 430–436, 1992.

[7] S. Pal, I. Alam, K. Sahoo *et al.*, "I/o architecture, substrate design, and bonding process for a heterogeneous dielet-assembly based waferscale processor," in *2021 IEEE 71st Electronic Components and Technology Conference (ECTC)*, 2021, pp. 298–303.

[8] B. Baker, "Fundamentals of hbm, mm, and cdm tests," 2019. [Online]. Available: https://embeddedcomputing.com/technology/analog-and-power/fundamentals-of-hbm-mm-and-cdm-tests

[9] "28nm i/o libraries." [Online]. Available: https://certus-semi.com/tsmc-28nm/

[10] "12nm i/o libraries." [Online]. Available: https://certus-semi.com/tsmc-16-12nm/

[11] K. Jeong, A. B. Kahng, and C. J. Progler, "New yield-aware mask strategies," in *Photomask and Next-Generation Lithography Mask Technology XVIII*, T. Konishi, Ed., vol. 8081, International Society for Optics and Photonics. SPIE, 2011, p. 80810P. [Online]. Available: https://doi.org/10.1117/12.899295

[12] D. Patel, "Die size and reticle conundrum - cost model with lithography scanner throughput," 2018. [Online]. Available: https://semianalysis.substack.com/p/die-size-and-reticle-conundrum-cost

[13] W. Kuo and T. Kim, "An overview of manufacturing yield and reliability modeling for semiconductor products," *Proceedings of the IEEE*, vol. 87, no. 8, pp. 1329–1344, 1999.

[14] Y. Chen, D. Niu, Y. Xie *et al.*, "Cost-effective integration of three-dimensional (3d) ics emphasizing testing cost analysis," in *2010 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2010, pp. 471–476.

[15] "International technology roadmap for semiconductors 2007 edition, yield enhancement." [Online]. Available: https://www.semiconductors.org/wp-content/uploads/2018/08/2007Yield.pdf

[16] H. Li, G. Katti, L. Ding *et al.*, "The cost study of 300mm through silicon interposer (tsi) with beol interconnect," in *2013 IEEE 15th Electronics Packaging Technology Conference (EPTC 2013)*. IEEE, 2013, pp. 664–668.

[17] A. A. Bajwa, S. Jangam, S. Pal *et al.*, "Heterogeneous integration at fine pitch (10 µm) using thermal compression bonding," in *2017 IEEE 67th Electronic Components and Technology Conference (ECTC)*, 2017, pp. 1276–1284.

[18] M. Ohyama, M. Nimura, J. Mizuno *et al.*, "Hybrid bonding of cu/sn microbump and adhesive with silica filler for 3d interconnection of single micron pitch," in *2015 IEEE 65th Electronic Components and Technology Conference (ECTC)*. IEEE, 2015, pp. 325–330.

[19] J. H. Lau, "Recent advances and trends in advanced packaging," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 12, no. 2, pp. 228–252, 2022.

[20] "Flexible manufacturing platform for high volume tcb and high density fowlp, https://ewh.ieee.org/soc/cpmt/presentations/cpmt1602b.pdf." [Online]. Available: https://ewh.ieee.org/soc/cpmt/presentations/cpmt1602b.pdf

[21] G. Gao, T. Workman, L. Mirkarimi *et al.*, "Chip to wafer hybrid bonding with cu interconnect: High volume manufacturing process compatibility study," in *2019 International Wafer Level Packaging Conference (IWLPC)*, 2019, pp. 1–9.

[22] D. Stroobandt, "Recent advances in system-level interconnect prediction," *IEEE Circuits and Systems Newsletter*, vol. 19, no. 9, pp. 4–20, 2000.

[23] P. Singh and D. Landis, "Optimal chip sizing for multi-chip modules," *IEEE Transactions on Components, Packaging, and Manufacturing Technology: Part B*, vol. 17, no. 3, pp. 369–375, 1994.

[24] A. M. Saif M. Khan, "Ai chips: What they are and why they matter, cset.georgetown.edu/research/ai-chips-what-they-are-and-why-they-matter/, https://doi.org/10.51593/20190014," 2020. [Online]. Available: cset.georgetown.edu/research/ai-chips-what-they-are-and-why-they-matter/

[25] E. Huang and R. E. Korf, "Optimal rectangle packing: An absolute placement approach," *Journal of Artificial Intelligence Research*, vol. 46, pp. 47–87, 2013.

[26] S. S. Iyer, S. Jangam, and B. Vaisband, "Silicon interconnect fabric: A versatile heterogeneous integration platform for ai systems," *IBM Journal of Research and Development*, vol. 63, no. 6, pp. 5:1–5:16, 2019.

[27] J. Sartori, A. Pant, R. Kumar *et al.*, "Variation-aware speed binning of multi-core processors," in *2010 11th International Symposium on Quality Electronic Design (ISQED)*, 2010, pp. 307–314.