System Design for 3D Multi-FPGA Packaging

Thorlindur Thorolfsson, Paul D. Franzon
Department of Electrical and Computer Engineering,
North Carolina State University, Raleigh, NC, 27695
{trthorol, paulf}@ncsu.edu, 919.515.7351

***Abstract*—3D stacking and integration can provide tremendous advantages to electronic systems. This paper explores the system-level considerations such as layout, routing and IO in the design of 3D Multi-FPGA Packaging, along with their architectural implications.**

***Index Terms*—3D IC, CAD, VLSI**

1. Introduction

Numerous technologies for creating 3D systems are becoming available. These exciting technologies offer potential for great performance. At the same time they present challenging problems. The purpose of this paper is not to explore 3D technology (see [1, 2] for a summary) but to illustrate some of the system-level design considerations that arise by presenting examples from our experience in designing a 3D multi-FPGA package using Irvine Sensors' 3D MINT process [3], along with the architectural implications that they bring about.

2. Design of 3D Multi-FPGA Package

In this section we explain the details of the multi-FPGA package we designed. First, in section 2.1 we give an overview of the 3D MINT process that was used to build the package. Second, in section 2.2 we explain the design details, including layout, routing and IO considerations.

2.1 3D MINT Technology

The Irvine Sensors' 3D MINT process works in the following way. First, holes are cut into silicon or alumina substrates. Embedded elements, which include: active dice, passives, and copper plugs (shown as hashed stipple in Figure 2) are then placed in the holes. Epoxy resin is used to hold the embedded elements in place in the substrate. Once the embedded elements have been inserted into the substrate, a conventional integrated circuit process that provides up to six layers of metal is used to interconnect the pads on the active circuits, the passives and the copper plugs. The substrate layers are then interspersed between thermal management layers and stacked vertically. The thermal management layers are very important to draw the heat away from the active circuits to the outside. The resulting stack is the packaged in traditional manner, in our case using a ball grid array. Figure 1 shows a close-up of the interface between the substrate and one side of an integrated circuit. Figure 2 shows a cross sectional drawing of a 3D MINT system with four active dice, four thermal management layers.

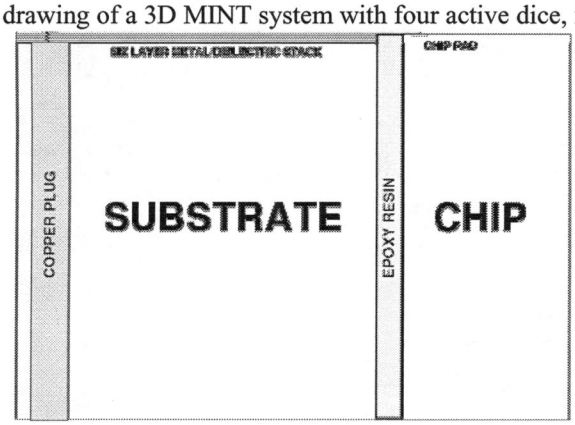

Figure 1. Irvine Sensors 3D Mint Process - cross section blowup of chip-substrate interface (not to scale)

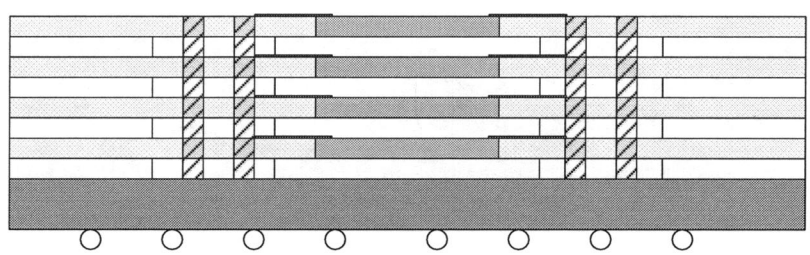

Figure 2. Irvine Sensors 3D Mint Process - cross sectional

It is important to note how tier-to-tier connectivity is achieved through copper plugs as shown on the left side of Figure 2. The plugs traverse the entire thickness of the substrate and can optionally be contacted by vias below the lowest substrate metal layer. This effectively provides blind via capability, which allows stacking of a via from layer 3 to layer 4 right on top of a via between layers 1 and 2 without connecting the nets associated with those vias.

2.2 Design of Multi-FPGA 3D System

This section explains the details of the 3D multi-FPGA system prototype that we designed using Irvine Sensors' 3D MINT technology. The system is a four-tier stack-up where each tier is comprised of a Xilinx VP20 FPGA, two Micron 256 MB DDR2 SDRAM chips, and six embedded passive decoupling stealth capacitors [4] to provide power and ground integrity. The total size of the module (excluding the cooling fan) is 27mm x 37 mm x 4 mm. The main issue in laying out the tiers of the module is heat. The Xilinx VP20 FPGA can operate at a temperature up to 125 °C where as the Micron SDRAM can only operate up to a temperature of 85 °C. As a result of this, the system was designed so that no memory die was ever directly above any of the FPGA dice.

The actual routing was done using Skill code in the Cadence tool suite. In order to do routing several issues had to be considered. First, the silicon substrate only has six metal layers to route with on each tier. Of those six metal layers, the first four metal layers on each tier have to be mostly devoted to power planes. This is necessary because the FPGA chips require a lot of supplies. The remaining wires from the first four metal layers, that are left over after routing power are used to support routing between the FPGA and the memory chips on that tier. The fifth metal layer is used to do all of the FPGA tier-to-tier routing. The reason that it is possible to do all the tier-to-tier routing in just one metal layer is that the FPGA chips have extreme flexibility in their internal connectivity. As a matter of fact, the programmability of the FPGA chips provides yet another level of routing if necessary. The sixth metal layer is used to support the tier-to-tier connectivity through the copper plugs. Overall, this routing provides very good tier-tier connectivity (summarized in Table 1) and massive amounts of IO bandwidth, both regular and RocketIO. The regular IO bandwidth comes from the large number of channels, while the RocketIO bandwidth comes from the raw speed of the RocketIO, which is 3.125 Gbps. The clock and programming nets are routed directly through the BGA. This approach was chosen because it ensures that all other tiers will still be usable even if one or more of the tiers is damaged. In case the BGA is damaged, all the control and clock signals are also routed to the top of the stack where they can possibly be accessed by bonding or probing. Additionally, provisions have been made for upward expansion to a fifth tier, although this tier would have limited IO bandwidth to all but the tier directly underneath it.

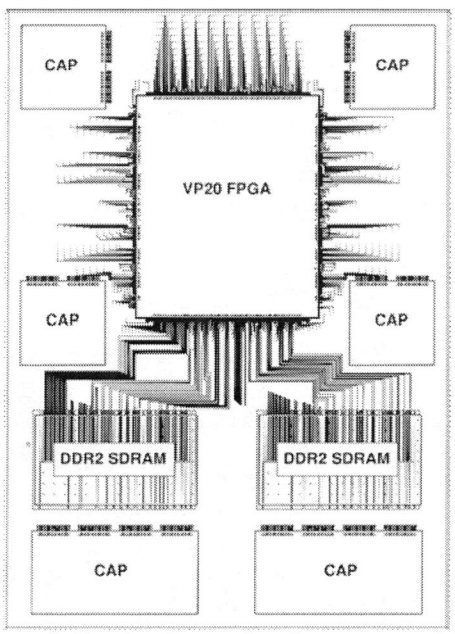

Figure 3. Routing for one tier of prototype module

Tier	BGA	Tier 1	Tier 2	Tier 3	Tier 4
BGA		163/2	36/2	41/2	29/2
Tier 1	163/2		161/2	0/2	0/2
Tier 2	36/2	162/2		135/2	
Tier3	36/2	0/2	135/2		161/2
Tier4	29/2	0/2		161/2	

Table 1: Regular IO / Rocket IO connectivity between tiers.. Each number represents the number of channels available for that type of IO.

3 System Synthetic Aperture Radar (SAR) Implementation

We plan to use the 3D multi-FPGA system to explore architectures that benefit from 3D integration. Our main focus so far has been on synthetic aperture radar. SAR is a natural fit for the 3D multi-FPGA system because it requires high performance, high memory bandwidth, and a small footprint for use in aircraft such as unmanned aerial vehicles. We also plan to explore other architectures, including computer vision and ray-tracing.

3.1 Brief Introduction to SAR

Synthetic aperture radar or SAR is a type of radar that unlike conventional radar is primarily used for imaging. SAR uses the motion of the radar platform along with extensive use of digital signal processing to effectively produce a very narrow beam that synthetically increases the size of the imaging aperture. SAR imaging has a wide variety of uses ranging from remote sensing applications such as cartography and oceanography to military uses such as reconnaissance, surveillance and battle damage assessment. For the right applications, SAR has three big advantages over optical imaging. The first advantage is that SAR carries its own illumination. This means that the SAR images are formed only from the waves transmitted by the radar, making the imaging independent on external lighting conditions such as time of day. The second advantage of SAR is that the frequencies commonly used for SAR can pass through clouds and other weather artifacts with very little attenuation. This means that SAR systems can be used under almost any weather conditions. The third and final advantage is that many materials scatter microwave frequencies differently than they do visible ones. This means that they can provide information that is not present in visible spectrum pictures such as ice composition of glaciers.

3.2 Implementation Issues in 3D packaging, a Case Study on SAR

The algorithm that we use for our SAR architecture is derived from the one used in the RASSP [5], project, with minor modifications such as increased resolution. Our algorithm operates on a grid, where the two axes are range (perpendicular to platform motion) and azimuth (parallel to platform motion). Furthermore, the algorithm consists of the following five steps, listed below preceded by either range or azimuth depending on along which axis the step operates: 1) Range Low Pass Filtering, 2) Range Fast Fourier Transform, 3) Azimuth Fast Fourier Transform, 4) Azimuth Complex Multiply 5) Azimuth Inverse Fast Fourier Transform. All these steps can be accomplished using a combination of four processing units: 1) Low Pass Filter Unit, 2) Fast Fourier Transform Butterfly Unit, 3) Memory Reordering Unit, 4) Complex Multiplication Unit.

Overall, the algorithm needs a lot of memory bandwidth. The key to supplying this bandwidth is to partition the memory into small fast memories that are close to the logic that uses them. These partitions are a natural fit for the tiers in the 3D multi-FPGA package. Earlier we explained how the different steps operate on different axes; these differentiations are important for the partitioning of the memory. We have determined, through simulation, that of the five steps, all the steps except the Range Fast Fourier Transform can operate completely independently of range. Operating completely independently of range means that the step requires only the data stored in the memory of one tier to complete its calculation. In the Range Fast Fourier Transform step, the only range dependent step, just 23% of the operations in the step are actually range dependent. This leads naturally to partitioning the memory based on range rows. In this scheme we put every fourth range row on its own tier. In this scheme every range row ending in b‘00 in binary goes on tier0, b’01 on tier1, b’10 on tier2 and b’11 on tier3. This approach to memory partitioning can be extended for any power of two. Figure 4 shows how the architecture is arranged on the four tiers, with the two units the require access to memory on more than one tier shown to occupy all four tiers.

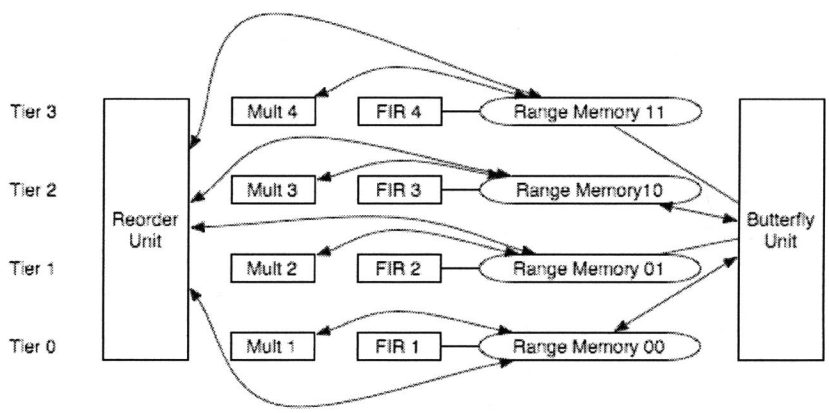

Figure 4. SAR architecture partitioned onto four tiers

We envision the 3D multi-FPGA packaging to be extended to more than four tiers to supply even more memory bandwidth. This extra bandwidth would provide for an increase in the SAR resolution. The memory bandwidth requirements for SAR increase drastically as the resolution increases as shown in Figure 5 below. We also envision similar partitioning schemes to provide the benefits of 3D packaging to other architectures.

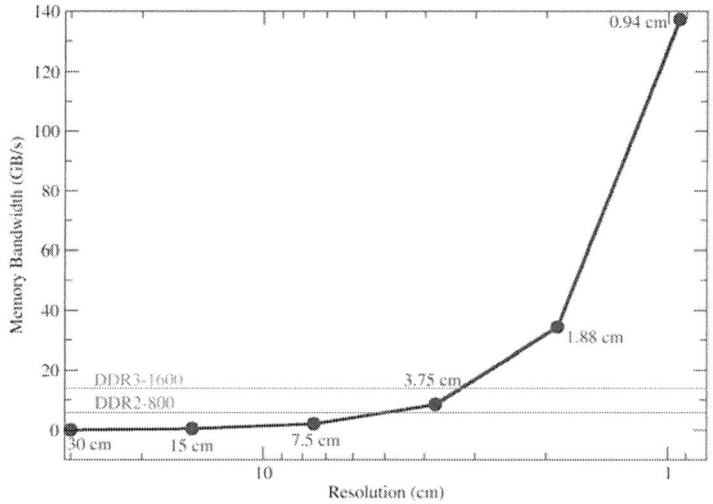

Figure 5. Memory bandwidth requirement vs. target resolution capability.

4. Conclusions

Numerous technologies for 3D integration are becoming available, including 3D packaging. The availability of these technologies presents, new challenges, in terms layout, routing and IO. We have discussed several of these challenges and how to approach them in our discussion of the 3D multi-FPGA package. We have also discussed the architectural issues of 3D packaging in our discussion of the SAR architecture. It is clear that the 3D packaging is the future, bringing the benefits of high memory bandwidth and reduced interconnect length, to numerous architectures.

References

[1] K. Banerjee, S. Souri, P. Kapur, K. Saraswat, "3-D ICs: A Novel Chip Design For Improving Deep-Submicrometer Interconnect Performance and Systems-on-Chip Integration," Proc. IEEE, Vol. 89, No. 5, 2001.

[2] W. R. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A. M. Sule, M. Steer, and P. D. Franzon, "Demystifying 3D ICs: The Pros and Cons of Going Vertical," IEEE Design & Test of Computers, vol. 22, no. 6, pp. 498-510, Nov.-Dec. 2005.

[3] J. Stern, V. Ozguz, J. Yamaguchi, P. Franzon, S. Lipa, S. Mick, L. Schaper, A. Malshe, M. Glover, M. Kelley, A. Glezer, Y. Joshi, "A Heterogeneous System-in-a-Stack Technology Incorporating Area-Array Interconnects, Thermal Management and Integrated Passives," IMAPS Int. Conf. & Exhibition on Device Packaging, 2006

[4] L. Schaper, R. Ulrich, D. Nelms, E. Porter, T. Lenihan, and C. Wan, "The 'stealth' decoupling capacitor," in Proc. 47th Electron. Comp. Technol. Conf., May 1997, pp. 724–729.

[5] B. Zuerndorfer, G. A. Shaw, "SAR Processing for RASSP Application" Proceedings 1st Annual RASSP Conference, Arlington, VA, August, 1994