Intrinsic Area Array ICs: What, Why, and How

Peyman Dehkordi, Chandra Tan, and Donald Bouldin *
Department of Electrical Engineering
The University of Tennessee
Knoxville, TN 37996–2100
{dehkordi,chandra,bouldin@microsys1.engr.utk.edu}

Abstract

Area-array bonding technology (i.e. Flip-chip, C4) was pioneered by IBM in the late 1960's as an alternative to periphery bonding technology (i.e. wire-bond). In recent years, several commercial companies have started offering bumping and flip-chip services. Flip-chip technology is expected to grow at at compound annual growth rate of 38% through the year 2001 [1]. The purpose of this paper is to address the IC design issues and alternatives that are presently being used for area-array bonding technology and show the impact of these design issues at the system level.

1. Introduction

Area-array solder interconnections can be made using the flip–chip assembly technique. Known also as the C4 process (Controlled Collapse Chip Connection), flip-chip bonding was developed by IBM in 1964 and has been used for over 30 years to assemble IBM's mainframe computer modules [16]. Flip-chip gives the highest chip density of any packaging method to support pad-limited IC designs. Instead of placing the chips in space-wasting individual packages, flip chip places them face down onto matching connections on a substrate or board by means of solder pads or bumps. Since the connections are under the chip, no additional space is required for bond wires. Connections may be made anywhere on the surface of the chip, using the whole chip area for connections, instead of just around the perimeter. The solder bumps provide shorter connections avoiding the performance penalties of conventional bond wires.

Some studies have been conducted recently to show the impact of area–array flip–chip bonding on today's VLSI designs. The study conducted in [7] showed the reduction of die size and the increase in I/O count when peripheral wire–bond technology was replaced by area–array flip–chip technology. A conceptual trade–off analysis between peripheral and area–array bonding in MCMs is presented in [14]. The impact of partitioning an ultra-large single die into multiple smaller dies housed on a MCM is shown in [8]. The result of this study indicated a dramatic reduction of cost if the CPU studied were divided into three chips bonded using area–array technology and interconnected on a MCM-D substrate.

In recent years, several commercial companies have started offering bumping and flip-chip services. Flip-chip technology is expected to grow at a compound annual growth rate of 38% through the year 2001 [1]. Flip-chip technology does have a major drawback in that not every IC design is built using this process, so a decision to use flip chip for all the devices in multichip modules or boards may require postprocessing the die to put solder bumps on the area-array pads. This postprocessing step involves the placement and routing of the area-array pads on the existing top metal layer of the die or on an additional distribution metal layer. The purpose of this paper is to address the IC design issues and alternatives that are presently being used for area-array bonding technology and show the impact of these design issues at the system level.

2. Present Approaches for Designing An Area–Array IC

Recently, most military and commercial electronic products are evolving toward lower cost, smaller form factor, lower weight and higher performance. All the improvements can best be realized at the chip level by eliminating the IC packages and transitioning from perimeter to area I/O. With the exception of IBM and a few semiconductor manufacturers who are producing area–array ICs, most manufacturers are reluctant to add bumps to their dies or redesign for area-array unless large quantities of dies are procured.

*The authors gratefully acknowledge the support of ARPA grant DAAH04-94-G0004

Presently, there are two major approaches for designing an area-array IC: **Intrinsic** and **extrinsic**. An intrinsic area-array IC is one specially designed and laid out for area-array bonding. Whereas, an extrinsic area-array IC is one originally designed and laid out for periphery bonding but has been converted for area–array bonding by means of a redistribution layer. As shown in Figure 1, both types of ICs take advantage of the superior electrical performance of solder bumps and a smaller foot print as compared to wire-bond ICs. Our previous work quantitatively showed additional benefits can be gained by using intrinsic area–array ICs rather than using the extrinsic counterparts at the IC level [6].

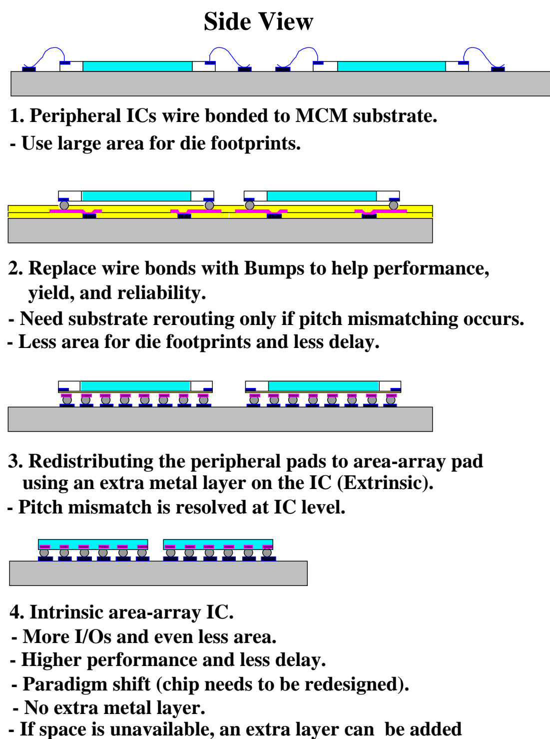

Figure 1. Why Intrinsic Area-Array IC.

An intrinsic approach, pioneered by IBM [3, 13, 15], is to place and route the I/O buffers anywhere closer to the center of the die and connect these I/O buffers to their corresponding pad sites placed in an array on the surface of the die. Using this approach, it is possible to design chips around circuit constraints rather than pad constraints; that is, it is possible to place buffer circuits where they make the most sense from a circuit design standpoint rather than where they are required for periphery bond purposes. Although the area–array IC resulting from this approach is fully optimized for silicon performance, this approach requires many additional considerations to conventional IC physical design processes. The extrinsic approach is the semi–optimized approach which is a process for transforming fine pitch peripheral IC pads to area array pads using a redistribution layer. In this approach, any die with periphery pads can be converted to an area-array IC by employing this redistribution process. This is best done at the wafer stage where a low-cost, batch photolithographic process can be used. Several semiconductor manufacturers including Motorola have designed some of their chips with area–array flip–chip assembly using this process [10, 11]. Some algorithms have been developed to perform the routing from the peripheral bonding pads to the area array C4 pads [4, 5, 17].

3. Intrinsic Approach

3.1. Problem Definition

At the present time, there is no commercial IC layout tool available to the IC design community to perform intrinsic area–array pad placement and routing. We are currently finishing the development of an area–array pad router which automates the placement and routing of the area–array pads on the IC. This is a post–processing tool which allows IC layouts generated by any other tool to be processed for area–array I/O pad placement and routing. This tool is different from a previously reported tool [5] since ours will place and route the pad using existing layers on the IC (if possible) before adding a new redistribution layer. Figure 2 shows some of the design steps for the intrinsic area–array IC layout problem. Unlike the physical design of a periphery

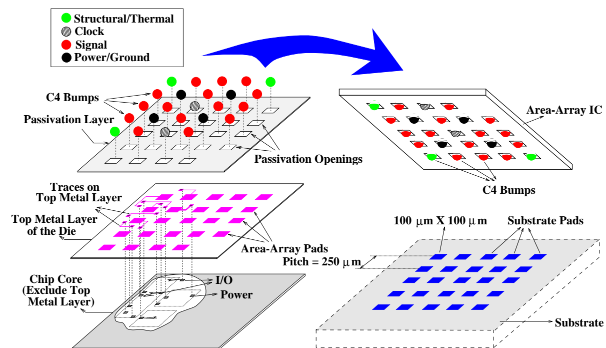

Figure 2. Intrinsic Area-Array IC Layout Problem Definition.

bonded IC which places and routes the I/O cells (pads and I/O buffers) around the die after the active components of the chip have been laid out, the I/O buffers for area-array pads and the active components are laid out at the same time.

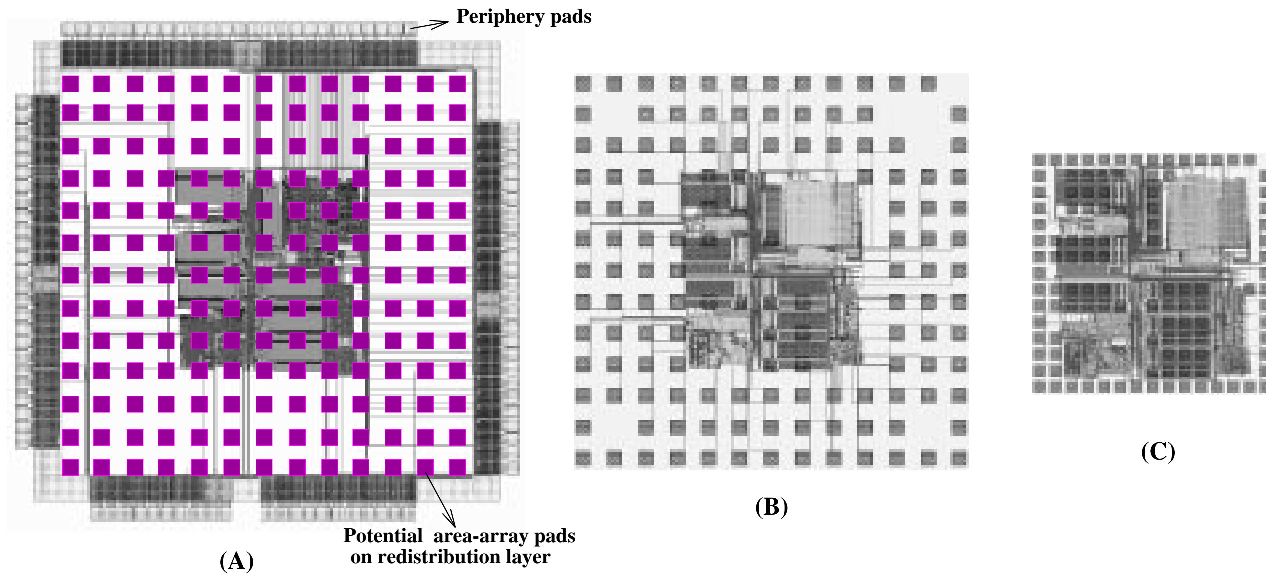

Figure 3. Extrinsic Vs Intrinsic Area-Array IC.

The area-array bonding pads will then be placed on the top metal layer of the die and routed to their corresponding I/O buffers.

The layout problem of an intrinsic area-array IC can be stated as: Given a "core" portion of the chip which already contains the I/O buffers, place the possible uniformly spaced area-array pads on the top metal layer of the design which may contain some prerouted wires and keepout areas and route these pads to the I/O ports of the chip using a limited number of available connection layers, such that design rules are obeyed and some other objective functions are satisfied.

3.2. Implementation

The framework for intrinsic area-array IC placement and routing has been implemented in C and run on a SUN Sparcworkstation. Given the mask geometries of the chip core stored in CIF (Caltech Intermediate Form), our tool will generate the area-array padframe of the IC. The pads are large areas of metal that are left unprotected by the overglass layer so they can be soldered to the bumps using the C4 process. Pad size is defined usually by the minimum size to which the solder bumps can be attached. The spacing of the pads is defined by the minimum pitch of the substrate. Both of these parameters are specified by the designer when running our tool. Before the tool proceeds to pad placement and pad routing steps, the chip core layout is flattened to extract the geometric information on the top few layers (the number of layers is specified by the designer), the pad keepout areas, the locations of I/O terminals, and the boundary of the core circuits. Utilizing this extracted information, the pad placement algorithm calculates the center of the chip core and then using this center as a reference point, generates a set of the potential area–array bonding pads on the top metal layer satisfying the user–specified bond pitch. The power/ground and clock I/O pads are routed first before the signal I/Os. The routing technique used is based on the general area routing algorithm [2, 9, 12].

4. Results and Discussions

In this paper, we present the results our area–array layout tool produced for two examples to show the impact of the intrinsic area–array IC on the IC and system level. The first example is the "Quantizer" taken from the tutorial example of the Epoch physical design tool. Figure 3A shows the "Quantizer" die with peripheral bonding pads. The size of the die is 3.88 $\times$ 3.87 mm. The number of peripheral I/O pads is 96 with size of 100 μm and pitch of 110 μm. The core of this die is 1.48 $\times$ 1.64 mm. Area–array pads have been superimposed on the top surface of the die to illustrate how the area–array version of the die could be done using the extrinsic approach to redistribute the I/Os from the periphery to an area–array. The size of each of these pads is 100 μm and the spacing between pads is 250 μm. Figure 3B shows the area–array version of the die using the intrinsic approach with the same pad size and pitch used in extrinsic approach. The area pads were placed on the existing top metal layer and the pad routing was done on the top two metal layers. The die size is reduced (2.85 $\times$ 2.85 mm) and the number

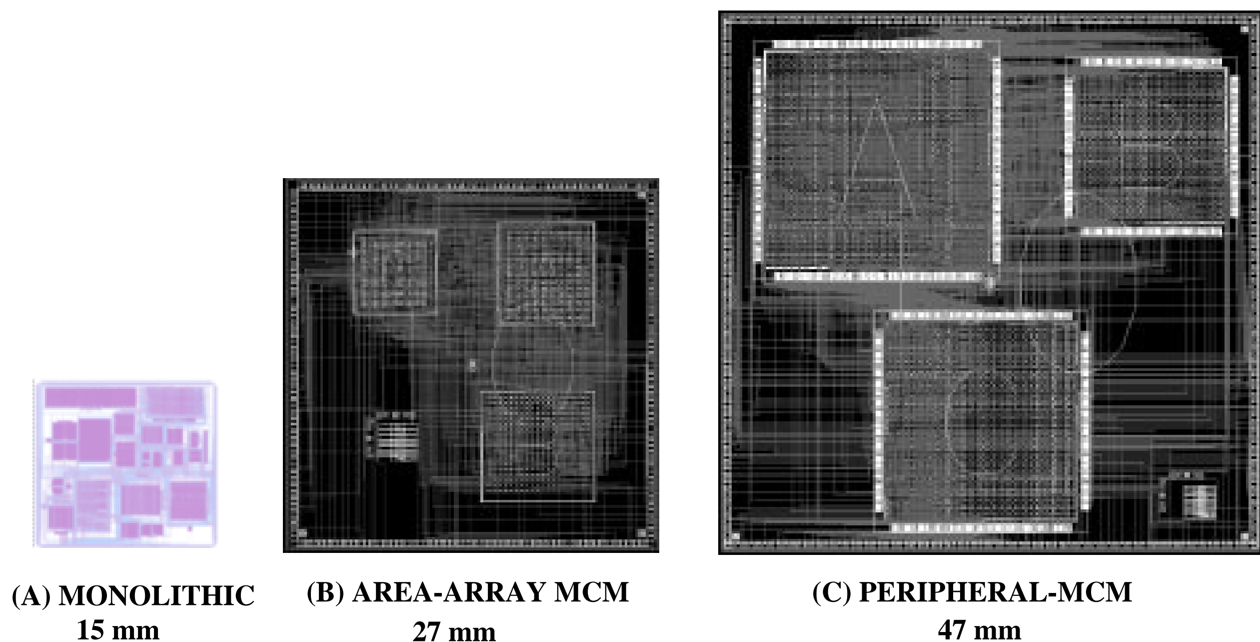

Figure 4. System Level Comparison (Monolithic, Intrinsic, Wire-bond.)

of I/O pads is increased (118). The extra I/O pads were used to connect more power and ground to the die to improve the signal integrity. With the advances of flip–chip technology, smaller bonding pad size and pitch are possible in the future. The design with area pad size of 80 μm and pitch of 120 μm is shown in Figure 3C. The reduction of the die size is more profound. If we use extrinsic approach with the same pad size and pitch, there will be no reduction of die size. In the above examples, we have shown the impact of intrinsic approach on the IC level.

In the next example, we will show the impact of the intrinsic area–array IC on the system level. We used the SUN MicroSparc CPU as a representative of a large design where MCM implementation may be required. The MCM implementation for a 3–chip design is shown in Figure 4 along with the monolithic solution (drawing to scale). The substrate is based on the Micro Module Systems MCM-D process and the ICs are Hewlett-Packard 0.5-micron technology as offered through the MIDAS and MOSIS services. Figures 4B and 4C are the 3-chip MCM system based on the intrinsic and periphery ICs. The results illustrate a clear advantage of the intrinsic approach. The study conducted in [8] indicated in more detail the impact of area–array technology at the system level.

5. Conclusion

We have presented an intrinsic approach of designing an area–array IC. We described the major reasons why we needed this approach and showed how this approach worked. The preliminary results on small and large designs have shown the impact of this intrinsic approach on the IC and system level.

6. Acknowledgement

The Epoch Physical Design tool by Cascade Design Automation and the MCM Station software by Mentor Graphics have been extensively used throughout this work. The authors gratefully acknowledge the donation of these two tools to the University of Tennessee.

References

[1] In *World Wide Web home page of FlipChip Technologies*, http://www.flipchip.com, August 1996.

[2] M. H. Arnold and W. S. Scott. An Interactive Maze Router with Hints. In *Proc. 25th Design Automation Conf.*, 1988.

[3] A. J. Blodgett and D. R. Barbour. Thermal Conduction Module: A High-Performance Ceramic Package. *IBM Journal of Research and Development*, 26:30–36, 1982.

[4] J.-D. Cho. *High-Performance Physical Design in Multilayer Packages*. PhD thesis, Northwestern University, Evanston, Illinois, June 1993.

[5] J. Darnauer and W. W. Dai. Fast Pad Redistibution for Periphery-IO to Area-IO. In *Proc. Multichip Module Conf.*, pages 38–43, Santa Cruz, CA, March 1994.

[6] P. Dehkordi and D. Bouldin. Design for Packageability: Early Consideration of Packaging from a VLSI Designer's Viewpoint. *Computer*, 1:76–81, Apr. 1993.

[7] P. Dehkordi and D. Bouldin. Design for Packageability: The Impact of Bonding Technology on the Size and Layout of VLSI Dies. In *Proc. IEEE Multichip Module Conf.*, pages 153–159, 1993.

[8] P. Dehkordi, K. Ramamurthi, D. Bouldin, H. Davidson, and P. Sandborn. Impact of Packaging Technology on System Partitioning: A Case Study. In *Proc. Multichip Module Conf.*, pages 144–149, Santa Cruz, CA, January 1995.

[9] G. T. Hamachi. An Obstacle-Avoiding Router for Custom VLSI. Technical Report UCB/CSD 86/291, Univ. of California, Berkeley, CA, April 1986.

[10] W. Huang and J. Castro. CBGA Package Design for C4 PowerPC Microprocessor Chips: Trade-off between Substrate Routability and Performance. In *IEEE 44th Electronic Components & Technology Conf.*, pages 88–93, Washington, D. C., May 1994.

[11] G. Kromann, D. Gerke, and W. Huang. A Hi–Density C4/CBGA Interconnect Technology for a CMOS Microprocessor. In *Proc. IEEE 44th Electronic Components & Technology Conf.*, pages 22–28, Washington, D. C., May 1994.

[12] C. Y. Lee. An Algorithm for Path Connections and Its Applications. *IRE Trans. Electronic Computers*, pages 246–265, September 1961.

[13] K. C. Norris and A. H. Landzberg. Reliability of Controlled Collapse Interconnections. *IBM Journal of Research and Development*, 13:266–271, 1969.

[14] P. Sandborn, M. Abadir, and C. Murphy. The Tradeoff Between Peripheral and Area Array Bonding of Components in Multichip Modules. *IEEE Trans. on Components, Packaging and Mfg. Tech. - Part A*, 17(2):249–256, June 1994.

[15] R. R. Tummala and J. Knickerbocker. Advanced Cofired Multichip Technology at IBM. In *Proc. IEPS.*, San Diego, CA, September 1991.

[16] R. R. Tummala and E. J. Rymaszewski, editors. *Microelectronic Packaging Handbook*. Van Nostrand Reinhold, 1989.

[17] M.-F. Yu and W. W.-M. Dai. Single-Layer Fanout Routing and Routability Analysis for Ball Grid Arrays. Technical Report UCSL–CRL–95–18, Univ. of California, Santa Cruz, CA., April 1995.