

# System Considerations for Wireless Capacitive Chip-to-Chip Signaling

Alex Chow, Philip Amberg, Michael Dayringer, Hesam Fathi Moghadam, Ron Ho, David Hopkins,  
Jon Lexau, Frankie Liu, Justin Schauer  
Oracle Labs, Oracle  
Redwood Shores, CA, USA

**Abstract**—Using capacitive-based chip-to-chip signaling in large-scale systems offers an interesting tradeoff between design and packaging complexity versus power consumption and performance. Placing chips together in close proximity offers low energy-per-bit costs and high I/O density, and therefore enables off-chip bandwidth levels far beyond those offered by traditional packaging and I/O technologies. Much of the previous published work on capacitive Proximity I/O has focused on mechanical methods for accurate chip alignment. In this paper we discuss some system design considerations unique to Proximity I/O. First, we compare and contrast circuit and layout techniques that optimize signal-to-noise ratio under expected chip misalignments. Next, we evaluate methods for establishing appropriate DC bias levels across a chip-to-chip capacitive link. Finally, we show a full Proximity I/O implementation to enumerate the required system overheads for clocking and misalignment compensation, and discuss how current trends in memory bandwidth and density are driving large-scale systems towards such solutions.

## I. INTRODUCTION

High-density multi-chip packaging solutions offer significant benefits for high-performance microelectronic systems. Multiple chips placed in close proximity within a single package can communicate across high-density I/O channels with low power and latency, and enable systems that bypass traditional off-chip communication bottlenecks [1]. Already, the push towards higher memory performance has prompted the development and adoption of novel, high-density stacked memories [2]. Continuing advances in multi-core processing and high performance computing will further drive systems towards these integration capabilities.

For such multi-chip systems, coupled data communication [3] provides efficient, high-performance I/O links between chips, or between chips and substrates, interposers, or other packaging carriers [4]. Compared to traditional 3D integration using through-silicon vias (TSVs), coupled interconnect allows communication without permanent soldered connections; this provides improved tolerance to mechanical stresses, and allows replacement of individual chips to improve system yield [5].

Using coupled communication links introduces additional complexities at the packaging, system, and circuit levels [6]. This paper explores design considerations to address some of these challenges specific to capacitively-coupled chip-to-chip I/O [7]. We present different options for I/O pad layouts and

arrangements, and compare their ability to reject noise under varying chip-to-chip alignment. Next, we describe solutions to establishing DC bias levels in a capacitively-coupled link, and explore their implementations and costs. Finally, we show how these system considerations apply to the design of a 40nm Proximity I/O link, and identify the overheads and complexities they impose.

## II. SIGNALING

Like traditional electrical links, capacitively-coupled channels are susceptible to crosstalk noise from adjacent neighbors. However, the possibility of chip-to-chip misalignment complicates and exacerbates the problem: signal and crosstalk vary with chip alignment; different neighbors attack differently; and with sufficient misalignment, combined crosstalk exceeds the desired signal. Effective crosstalk mitigation therefore requires analyzing the physical arrangements of two chips and their I/O pad locations.

Misalignment effects can be controlled through packaging solutions using self-alignment MEMS structures [8], as well as circuit techniques to dynamically measure [9] and compensate for misalignment [10]. However, these solutions add often significant costs in area, power, and speed; these overheads also rise with higher compensation precision. It is therefore worthwhile to study how signal and crosstalk coupling changes with alignment. This lets us determine the alignment tolerance required to achieve a desired signal-to-noise ratio, and implement correspondingly sufficient compensation schemes.

The choice of I/O pad arrangement and geometry directly affects the channel's ability to reject nearest neighbor crosstalk. In this paper, we consider four pad arrangements: Single Ended, Side Differential, Corner Differential, and Butterfly Differential (Fig. 1). Single Ended signaling uses only one pad per signal, but it offers no ability to reject crosstalk; in the worst case, a signal is opposed by opposing transitions on all four side and four corner neighbors. With Side Differential signaling, a differential pair of pads is placed side by side; in the worst case, a signal is attacked by four side neighbors, but sees no net crosstalk from the corners (under perfect alignment). Corner Differential signaling improves on this by placing a differential pair diagonally; two side

neighbors now become common-mode and inject zero net crosstalk; in the worst case, a signal sees crosstalk from two sides and four corners.

To achieve even better crosstalk rejection, we developed the Butterfly Differential signaling scheme, which rejects crosstalk from all four side neighbors [6]. Fig. 1(d) shows a differential channel (A+, A-) with its four neighbors (B, C, D, E). Channel A sees zero net noise from channels B and E, because transitions on B+ and E+ are canceled by opposite transitions on B- and E-, respectively. Channels C and D inject only common-mode noise since they couple equally to pads A+ and A-. In the worst case, under perfect alignment, each pad therefore sees noise from only the four corners.

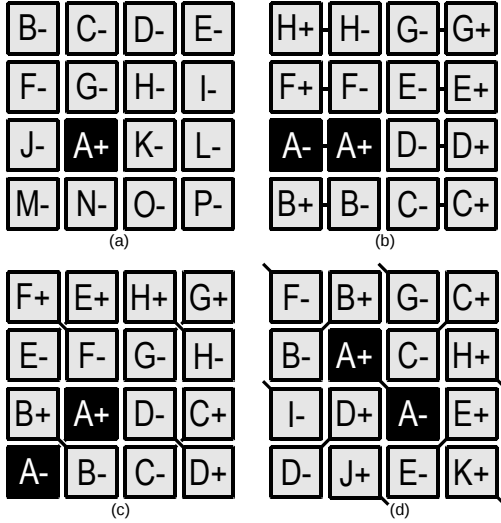


Fig. 1: Pad arrangements for (a) Single Ended, (b) Side Differential, (c) Corner Differential, and (d) Butterfly Differential signaling.

Butterfly Differential signaling is clearly the most robust, but it imposes an area overhead of one pad pitch around the perimeter of each array. It also introduces some routing complexity in layout. To compare the benefits and costs of these four pad arrangements, we quantify their noise rejection performance under varying alignment conditions. We use a 3D electromagnetic field solver [11] to compute the signal and crosstalk capacitances between the transmitting and receiving pads. We include the complete dielectric and metal stackup to model all coupling and parasitic capacitances.

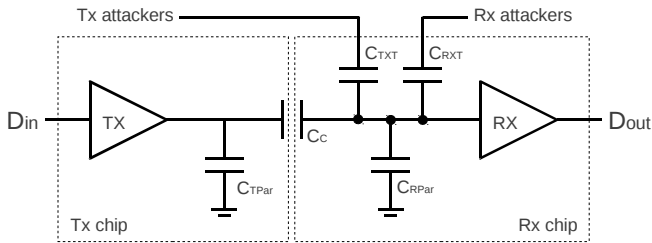


Fig. 2: Model of a chip-to-chip capacitively-coupled channel.

Fig. 2 is a simple channel model showing the coupling ( $C_c$ ), crosstalk ( $C_{TXT}$ ,  $C_{RXT}$ ), and parasitic ( $C_{TPar}$ ,  $C_{RPar}$ )

capacitances. The field solver computes these capacitances under various alignment conditions. We then determine the single-ended signal swing  $V_{sig}$  seen by the receiver RX as

$$V_{sig} = V_T \frac{C_c}{C_{RTot}} \quad (1)$$

where  $V_T$  is the voltage swing on the transmitting plate and  $C_{RTot}$  is the total capacitance at the receiver input. (This includes  $C_{RPar}$ , coupling capacitances, and receiver loading).

Each receiver sees crosstalk coupled directly from neighboring transmitting channels on the opposite chip ( $C_{TXT}$ ), and indirectly through neighboring receiving pads on the same chip ( $C_{RXT}$ ). Considering each of the eight nearest side and corner neighbors ( $i = 1, 2, \dots, 8$ ) and summing their contributions, the total crosstalk from the transmitting and receiving neighbors are, respectively,

$$V_{xT} = \frac{\sum_i b_i C_{TXTi} V_{Ti}}{C_{RTot}} \quad (2)$$

$$V_{xR} = \frac{C_{RXTi}}{C_{RTot}^2} \sum_i b_i C_{ci} V_{Ti} \quad (3)$$

where  $b_i$  is the worst-case bit value ( $\pm 1$ ) of the  $i^{\text{th}}$  attacking channel. For single-ended channels,  $b_i = -1$  for all  $i$ . For differential channels,  $b_i$  is chosen according to the layouts shown in Fig. 1. Excluding other bounded noise sources, a receiver observes a single-ended voltage swing of

$$V_{sx} = V_{sig} + V_{xT} + V_{xR}. \quad (4)$$

We consider a specific design with square I/O pads placed on a  $36 \mu\text{m}$  pitch, spaced  $1 \mu\text{m}$  apart. The pads are  $1 \mu\text{m}$  thick and covered by  $2 \mu\text{m}$  of passivation. We assume a  $200 \text{ fF}$  receiver input capacitance, and  $1 \text{ V}$  peak-to-peak signal transitions on the transmitting plates. To account for the area overhead of differential signaling, we used the area of two I/O pads per receiver for the Single Ended pad arrangement.

Fig. 3 shows the variation in the net differential received signal  $\Delta V_{sx}$  with chip-to-chip spacing ( $z$ ). As expected, Butterfly Differential signaling (BD) is the most effective. When the chips are close together, Corner Differential (CD) is preferable to Side Differential (SD) because CD rejects crosstalk from two side neighbors. Interestingly, however, SD outperforms CD at large separations because an SD pad sees zero net crosstalk from the corners; when the chips are far apart, differences in coupling between the side and corner neighbors narrow, and the proportionately larger attack from the corners degrade the performance of CD signaling.

Fig. 3 also shows the net differential received signal with in-plane misalignment, when one chip is simultaneously displaced along both in-plane dimensions. The rejection of common-mode and differential crosstalk is only effective if the coupling to differential transitions are symmetric. With misalignment, we see that CD signaling more closely matches BD signaling in noise cancellation performance, as coupling asymmetry removes some advantages of the BD layout.

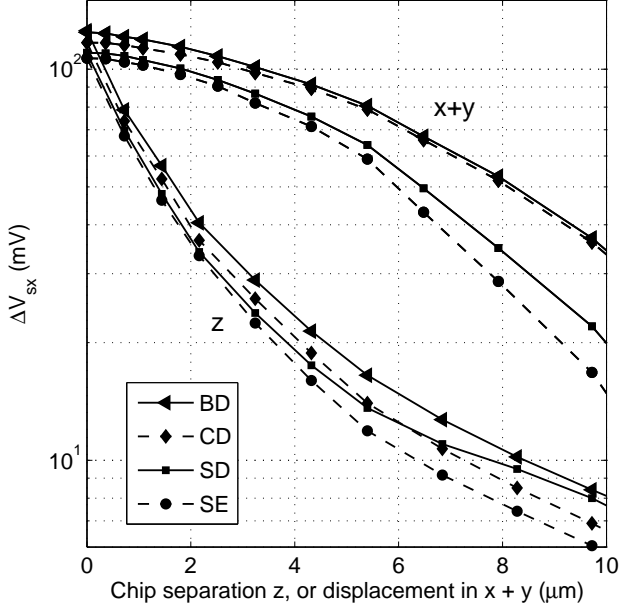


Fig. 3: Net differential received signal vs. chip-to-chip separation  $z$ , or simultaneous in-plane displacement along  $x$  and  $y$  (with  $z = 0$ ), for Butterfly Differential (BD), Corner Differential (CD), Side Differential (SD) and Single Ended (SE) signaling. For differential signaling,  $\Delta V_{sx} = V_{sx+} - V_{sx-}$ . For single ended signaling,  $\Delta V_{sx} = V_{sx}$ .

### III. DC BIAS CONSIDERATIONS

Since DC levels cannot be transmitted over a capacitively-coupled channel, we need methods of establishing an appropriate DC bias at the receiver input. Here, we briefly discuss three such techniques (Fig. 4).

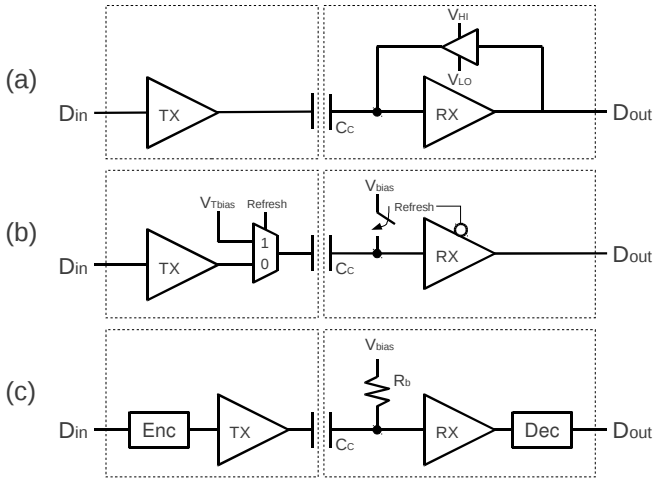


Fig. 4: Three methods of restoring DC bias across a capacitive channel.

#### Latching Receiver

We can use a feedback keeper around the receiver to maintain the DC bias voltage corresponding to the previously detected bit value (Fig. 4a). This method is conceptually simple, but the output voltage levels of the keeper must be

controlled such that coupled data transitions on the pad can overcome the bias introduced by the keeper. If the difference between  $V_{Hi}$ ,  $V_{Lo}$  and the receiver's midpoint bias is too large, incoming data transitions cannot trigger the receiver; if the difference is too small, noise margin degrades. Optimal choices of  $V_{Hi}$ ,  $V_{Lo}$  depend on the incoming signal swing, which changes with chip alignment and may vary over time. An effective implementation must therefore adapt  $V_{Hi}$  and  $V_{Lo}$  dynamically.

#### Periodic Refresh

An alternative approach is to use a refresh circuit that periodically drives the receiver input to the desired bias, while all transmitters simultaneously apply a common (usually midpoint) voltage (Fig. 4b). In the simplest implementation, all channels pause periodically for refresh, causing necessary interruptions to data flow. If this is not acceptable, we can instead build extra channels into the array and swap them with channels being refreshed, cycling and repeating through all channels.

#### Constant Biasing with DC Balanced Data

Finally, we may provide a constant bias to the receiver input through a slow (highly resistive) path (Fig. 4c). This technique is effective for DC balanced data that guarantees a minimum transition frequency, and encoding can enforce this property. A simple line code is the Manchester code (*e.g.* 1  $\rightarrow$  01, 0  $\rightarrow$  10); however, to sustain a given data rate, the channel must run at twice that rate, and consume up to twice the power. More sophisticated line codes (*e.g.* 8b/10b, 64/66b) impose lower bandwidth and power overheads, but introduce extra complexity and latency for encoding and decoding.

With this method, the bias voltage may drift depending on the data pattern. When the channel is idle, the receiver voltage is slowly pulled to  $V_{bias}$ , degrading noise margin and eventually destroying the previous bit value. Modeling the channel as a first-order RC circuit with time constant  $\tau = R_b (C_c + C_{RPar})$ , the bit period  $T$  of a random binary sequence should be short relative to  $\tau$  to prevent significant DC drift. In this case, the probability that the DC bias drifts outside a desired range  $\Delta V$ , for a signal swing  $V_T$ , is given by

$$P_e = \text{erfc} \left( 2 \frac{\Delta V}{V_T} \sqrt{\frac{\tau}{T}} \right). \quad (5)$$

For example, to achieve  $P_e < 10^{-26}$  for  $\Delta V/V_T = 0.1$ , we want  $\tau > 1431 T$ . For  $T = 300$  ps and typical capacitances of  $C_c = 1$  fF,  $C_{RPar} = 30$  fF, we design our biasing circuit such that  $R_b > 14$  M $\Omega$ . This assumes that the current through the biasing resistor dominates over other leakage currents (*e.g.* transistor gate leakage) at the receiver input node.

This analysis holds for active channels carrying a continuous data bitstream. For I/O channels characterized by bursty data, some combination of these methods may be more appropriate.

#### IV. DESIGN OF A CAPACITIVE CHIP-TO-CHIP LINK IN 40NM CMOS

We now consider the design of a 608-channel capacitively-coupled interface in 40nm 1.0 V CMOS (Fig. 5). Each channel is designed to operate at 3 Gbps to provide 3.6 Tbps of total bidirectional bandwidth. The target energy consumption of the communication circuits including clocking is  $< 1$  pJ/bit. Clock signals are recovered using DLLs. On-chip BIST circuits facilitate testing of each link and measurements of BER.

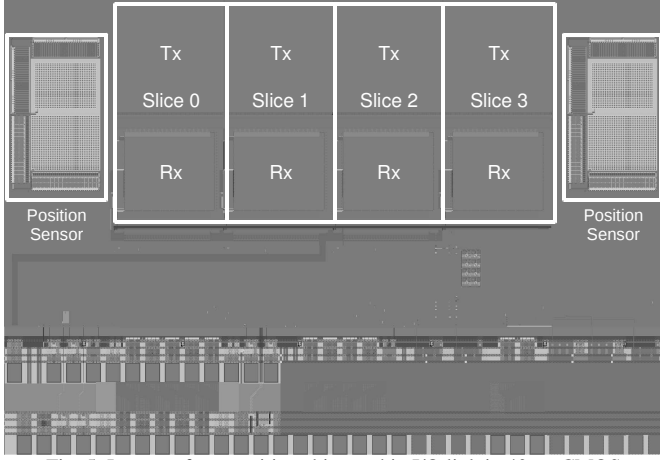


Fig. 5: Layout of a capacitive chip-to-chip I/O link in 40nm CMOS.

##### Physical Design

The I/O interface consists of four fitted “slices,” each with 152 transmit and 152 receive channels. Pads are placed at a pitch of  $24\ \mu\text{m}$ . For maximum crosstalk rejection, we use the Butterfly Differential pad arrangement. The diagonally alternating nature of this layout adds an area overhead of one pad pitch along each edge. The receive pad array in each slice is 17 pads (8 bits) tall by 20 pads (19 bits) wide.

To mitigate physical misalignment between chips, we use electronic alignment correction [7] which shifts the location of the transmitting channel such that it best aligns with the receiving pad. To achieve fine-grain correction, each transmitting pad is divided into a  $4 \times 4$  array of smaller micropads. Each bit may shift by up to half a pad pitch beyond the nominal configuration in each dimension. This accommodates misalignment of  $\pm 12\ \mu\text{m}$  in steps of  $6\ \mu\text{m}$ .

To deal with greater misalignment, we add extra channels in the arrays, and permute data bits in the datapath to achieve coarse shifting in steps of  $24\ \mu\text{m}$ . We allow shifts of  $\pm 4$  pad pitches ( $96\ \mu\text{m}$ ) vertically and  $\pm 2$  pad pitches ( $48\ \mu\text{m}$ ) horizontally. Fine-grain correction works on top of coarse shifting, allowing continuous fine-grain movement across the entire correction range.

The Butterfly pad layout and electronic alignment correction add a significant area overhead to the transmitter block. The active pad area used for signaling totals  $0.70\ \text{mm}^2$  across all four slices, but the total transmitter pad area is  $1.38\ \text{mm}^2$ ; overhead therefore doubles the transmitter size.

On-chip position sensors [9] measure the physical alignment between two chips, and the results are used to

configure the alignment correction circuitry. Each of the four slices may be configured separately to accommodate rotational misalignment.

##### Electrical Design

We implement continuous biasing at the receiver inputs, using transistors as biasing resistors [12]. For our application, Manchester encoding is too costly in power and bandwidth, and DC balanced codes introduce too much latency. Instead, we mix the input data stream with a  $2^8 - 1$  PRBS to create more probabilistic DC balanced data. With this approach, certain pathological data patterns (e.g. the PRBS itself) can cause the DC bias to drift outside the desired range.

The CMOS process used for this design does not provide high- $\kappa$  gate dielectrics, and leakage through the receiver transistor gate terminals is non-negligible compared to the current through the main biasing path. To limit gate leakage, we used 1.8 V thick-gate transistors to implement the devices directly connected to the receiving pads; unfortunately, these devices limit the speed of the receivers.

##### SUMMARY

This paper discusses some considerations particular to the design of capacitively-coupled chip-to-chip I/O links. We compared the crosstalk rejection properties of different I/O pad arrangements, and explored some solutions to establishing DC bias levels across a capacitive channel. Our 40nm design serves as a case study of how we resolve these issues in a modern deep submicron CMOS process.

##### REFERENCES

- [1] R. Drost, *et al.*, “Challenges in Building a Flat-Bandwidth Memory Hierarchy for a Large-Scale Computer with Proximity Communication,” Hot Interconnects 13, 17-19 Aug. 2005, pp. 13-22.
- [2] Micron Technology, Inc., “Hybrid Memory Cube: A Re-Architected DRAM Subsystem,” Hot Chips 23, 17-19 August, 2011.
- [3] R. Ho and R. Drost, Eds., *Coupled Data Communication Techniques for High-Performance and Low-Power Computing*, Springer, 2010.
- [4] S. Mick, *et al.*, “Buried Bump and AC Coupled Interconnection Technology,” Trans. on Adv. Packaging, Feb. 2004, pp. 121-125.
- [5] A. Chow, *et al.*, “Exploiting Capacitance in High-Performance Computer Systems,” VLSI-DAT, 23-25 Apr. 2008, pp. 55-58.
- [6] A. Chow, *et al.*, “Enabling Technologies for Multi-Chip Integration Using Proximity Communication,” VLSI-DAT, 28-30 Apr. 2009, 39-42.
- [7] D. Hopkins, *et al.*, “Circuit Techniques to Enable 430Gb/s/mm<sup>2</sup> Proximity Communication,” ISSCC, 11-15 Feb. 2007, pp. 368-369.
- [8] J. E. Cunningham, *et al.*, “Optical Proximity Communication in Packaged SiPhotonics,” Intl. Conf. on Group IV Photonics, 17-19 Sep. 2008, pp. 383-385.
- [9] A. Chow, *et al.*, “On-Chip CMOS Position Sensors Using Coherent Detection,” A-SSCC, 8-10 Nov. 2010, pp. 17-20.
- [10] R. Drost, *et al.*, “Electronic Alignment for Proximity Communication,” ISSCC, 15 – 19 Feb. 2004, pp. 144-145.
- [11] Maxwell Q3D Extractor, v.5, Ansoft Corporation, 2002.
- [12] J. Schauer, *et al.*, “Method and Apparatus for Biasing a Floating Node in an Integrated Circuit,” U.S. Patent 7750709, issued 6 Jul. 2010.