



## Advanced Interface Bus (AIB) Specification

---

*2021.3.182-3*

*Revision 2.1DRAFT10*

## Revision History

| Date      | Version | Summary Of Changes   |
|-----------|---------|--|
| 10/26/18  | 1.0     | Initial version  |
| 04/18/19  | 1.1     | Typographical fixes<br>Minor wording changes<br>Clarification of functionality<br>Additional bump array patterns for medium and high-density bump pitches  |
| 06/14/19  | 1.1.x   | Typographical fixes on Figure 39, clarification to Table 51<br>Clarification of I/O mismatch requirement in Section 2.1.9  |
| 09/27/19  | 1.2     | Clarification of latency specification in Section 2.1.7, Section 2.2, and Section 2.2.1<br>Updated Figure 41 and Figure 42 to remove dependencies of tx_transfer_en and rx_transfer_en from intermediate states  |
| 5/31/2020 | 2.0     | Issued closed in this revision (aib-phy-hardware numbering):<br>2,4 Electrical specs<br>11 Optional AUX block<br>13 Need to update *_sr_clkb doesn't operate in 2.0 mode<br>14 Need to update *_rcv_clk/clkb do not operate in 2.0 mode<br>15 Add follower signal list and bump map to Alternate Bump Maps to section 7<br>17 Need to update Table 6 MAC Interface with signals from the Usage Note<br>18 Add an AIB Plus 80 IOs Balanced example bump table<br>19 Table 40 lower left cell should be AIB0, lower right should be AIB1<br>20 Table 36 formulas should be $x=n-10$ , $y=2n+8$ . Table 39 formulas should be $x=n-12$ , $y=2n+20$<br>21 Add an AIB Plus 80 IOs Balanced example Bump Map to match new Bump Table<br>23 Summarize change between GEN1 and GEN2 as reference<br>24 Table 2 is confusing with min/typ Gen1/typ Gen2<br>26 Indicate unused pins (e.g. ns_rcv_clk) need to be driven low in Gen2<br>27 Update skew values for Gen2: Tds far 0.05UI, Tdcs far 0.025<br>28 Table 20 should list "Medium bump pitch" and use footnote for 45u, 36u examples<br>29 Update electrical spec for Gen2: tEW Near=0.65 UI, Trace=0.2 UI<br>30 Remove master & slave references, replace with leader and follower<br>31 Describe DBI and provision for 2 DBI bits for every 40 (38) data bits<br>35 Gen2 supports SDR only for the shift register<br>42 Bump tables need to explain what happens to signals not used in Gen2 like fs_sr_clkb<br>46 Clarify 3.2.1.4 Test Provision por_ovrd and device_detect_ovrd with example<br>47 Clarify in Interface Orientation that N, S, E, W can be dual-mode<br>48 Clarify 6.2 bump density ranges<br>49 Hyphenate high-density, medium-density and low-density |

Formatted Table

| Date             | Version          | Summary Of Changes   |
|------------------|------------------|--|
|                  |                  | 34 Channel information in electrical specs<br>41 Figure 18: The clock should appear stable and the skew is the distribution of data signals<br>43 Add standard loopback to help debug<br>50 2.0 minimum frequency should be 2Gbps. Change Table 3.<br>Issued closed in this revision (AIB-specification numbering):<br>2 Section 3.2 header format<br>3 Note to electrical specs about voltage supply<br>6 FIFOs added to 2.0 AIB Plus adapter<br>9 PHY/Application interface includes dual_mode_select signal<br>10 shift register clock signal name error<br>4 Loopback requirements<br>5 ESD specifications updated<br>12 Figure 32 error<br>13 Bump density description clarified<br>14 Added a recommended Word Mark bit location<br>16 Added m_wr_clk and m_rd_clk to MAC/PHY signal list<br>15 Added optional extension Test Pattern generator and checker<br>17 Only data and fwd_clk are low swing<br>18 Fixed SDR units in Table 3<br>19 Added Application to PHY signal m_gen2_mode for Gen1 or 2<br>20 m_power_on_reset changed to an input and an output<br>21 dual_mode_select must be stable at power-up<br>22 corrected m_rxifo_align_done name (See issue 24)<br>23 i_conf_done interface reset added to Application to PHY<br>24 Change FIFO to Phase Compensator<br>25 Revert Vcm specification for AIB Gen1 to Vos / 2 |
| <u>3/18/2021</u> | <u>2.1DRAFT1</u> | <u>30 Add standard packaging AIB-O support</u>   |

Formatted Table

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice.

Intel, the Intel logo and Stratix are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Other names and brands may be claimed as the property of others.

© Intel Corporation.

# Table of Contents

---

|  |    |
|--|----|
| Revision History .....   | 2  |
| Table of Figures .....   | 7  |
| Table of Tables .....  | 10 |
| Glossary and Acronyms .....                                    | 12 |
| Note on Language.....  | 15 |
| 1 Introduction .....   | 16 |
| 1.1 A Second Generation Die to Die Interconnect.....           | 16 |
| 1.2 Compliance Summary.....                                    | 17 |
| 1.3 Architecture .....   | 18 |
| 1.3.1 AIB Configurations.....                                  | 19 |
| 1.3.2 Near-Side and Far-Side Interfaces .....                  | 20 |
| 1.3.3 Leader, Follower, and Dual-Mode Interfaces.....          | 20 |
| 1.3.4 AIB Interface .....                                      | 22 |
| 1.3.5 AIB-to-MAC Interface.....                                | 27 |
| 1.3.6 AIB-to-Application Interface .....                       | 31 |
| 2 Functional Specification .....                               | 34 |
| 2.1 I/O Blocks.....  | 34 |
| 2.1.1 TX Block .....   | 34 |
| 2.1.2 RX Block.....  | 36 |
| 2.1.3 Data Exchange .....                                      | 37 |
| 2.1.4 Tristate.....  | 40 |
| 2.1.5 Weak Pull-Up and Pull-Down .....                         | 40 |
| 2.1.6 Data-clock operation.....                                | 40 |
| 2.1.7 Latency .....  | 46 |
| 2.1.8 Asynchronous mode .....                                  | 47 |
| 2.1.9 Mismatched Interfaces.....                               | 47 |
| 2.1.10 Unused Channels .....                                   | 48 |
| 2.2 AIB Adapter (AIB Plus only) .....                          | 48 |
| 2.2.1 Data-Retiming Register .....                             | 48 |
| 2.2.2 Adapter Phase Compensator (AIB Plus and Gen2 only) ..... | 49 |

|       |   |     |
|-------|---|-----|
| 2.2.3 | Sideband Control Signals .....                        | 54  |
| 2.2.4 | Data Bus Inversion .....                              | 62  |
| 3     | Reset and Initialization .....                        | 63  |
| 3.1   | Data-Transfer Ready .....                             | 63  |
| 3.1.1 | Standby Mode .....                                    | 63  |
| 3.1.2 | Data-Transfer Ready Signals .....                     | 63  |
| 3.1.3 | The Effects of De-asserting Data-Transfer Ready ..... | 63  |
| 3.2   | Initialization .....                                  | 64  |
| 3.2.1 | Power-on Reset Synchronization .....                  | 64  |
| 3.2.2 | Configuration .....                                   | 66  |
| 3.2.3 | Calibration (AIB Plus only) .....                     | 68  |
| 3.2.4 | AIB Link Ready .....                                  | 77  |
| 3.3   | Redundancy .....                                      | 77  |
| 3.3.1 | Active Redundancy .....                               | 77  |
| 3.3.2 | Passive Redundancy .....                              | 79  |
| 4     | Electrical Specification .....                        | 80  |
| 4.1   | Eye Diagram .....                                     | 80  |
| 4.2   | Overshoot and Undershoot .....                        | 81  |
| 4.3   | Electrostatic Discharge (ESD) Protection .....        | 82  |
| 5     | Design for Test .....                                 | 83  |
| 5.1   | JTAG .....  | 83  |
| 5.1.1 | JTAG I/Os .....                                       | 83  |
| 5.1.2 | JTAG cells .....                                      | 83  |
| 5.1.3 | AIB Private JTAG instructions .....                   | 83  |
| 5.2   | Loopback .....  | 84  |
| 6     | Physical Signal Arrangement .....                     | 86  |
| 6.1   | Interface Orientation .....                           | 86  |
| 6.2   | Bump Configuration .....                              | 88  |
| 6.3   | Bump Assignment Process .....                         | 89  |
| 6.3.1 | Data-Signal Bump Assignment .....                     | 91  |
| 6.3.2 | AIB AUX Signal Assignment .....                       | 107 |
| 6.3.3 | Example Bump Tables .....                             | 108 |
| 6.3.4 | Example Bump Maps .....                               | 112 |

|       |  |     |
|-------|--|-----|
| 6.4   | Stacking Channels into a Column .....  | 119 |
| 6.5   | Channel-Number Semantics .....   | 124 |
| 6.6   | Alternate Bump Maps.....   | 125 |
| 7     | AIB On Standard Packaging (AIB-O) .....  | 126 |
|       | AIB on Standard Packaging, or AIB-O, is an AIB 2.0 variant for applications that desire a high bandwidth parallel interface without using advanced high-density packaging technology. .... |     |
|       |  | 126 |
| 7.1   | AIB-O Interface .....  | 126 |
| 7.1.1 | AIB-O Channel .....  | 126 |
| 7.1.2 | AIB-O to MAC Interface.....  | 126 |
| 7.1.3 | AIB-O Reset and Initialization .....   | 127 |
| 7.2   | AIB-O Electrical Specification.....  | 127 |
| 7.3   | AIB-O Bump Configuration.....  | 128 |
| 7.4   | AIB-O Signal Bump Assignment .....   | 129 |
| 7.4.1 | AIB-O Example Bump Table.....  | 129 |
| 7.4.2 | AIB-O Example Bump Map.....  | 130 |
| 7.4.3 | Stacking AIB-O Channels into a Column .....  | 130 |
| 7.5   | Gen2 Features Not in AIB-O .....   | 131 |
| 7.6   | AIB-O Sideband Control Shift Register .....  | 131 |
| 7.6.1 | AIB-O Sideband Control Shift Register Timing .....   | 132 |
| 8     | Appendices .....   | 134 |
| 8.1   | Alternate Bump Maps.....   | 134 |
| 8.1.1 | Alternate (Leader) Bump Table .....  | 134 |
| 8.1.2 | Alternate (Leader) Channel Bump Map .....  | 135 |
| 8.1.3 | Alternate (Follower) Bump Table .....  | 137 |
| 8.1.4 | Alternate (Follower) Channel Bump Map.....   | 138 |
| 8.2   | Sideband-Control-Signal Shift Register Mapping (AIB Plus only) .....   | 140 |
| 8.3   | Test Pattern Extension.....  | 142 |
| 8.3.1 | Pattern Generator and Checker.....   | 142 |
| 8.3.2 | Pattern Data Paths .....   | 142 |

## Table of Figures

|   |    |
|---|----|
| Figure 1. AIB in the OSI Reference Model .....  | 19 |
| Figure 2. Near Side and Far Side.....   | 20 |
| Figure 3. Fixed Leader and Follower Interfaces .....  | 21 |
| Figure 4. Dual-Mode Interfaces.....   | 22 |
| Figure 5. AIB Signal Types.....   | 23 |
| Figure 6. Interconnecting Near-Side and Far-Side Signals .....                                    | 23 |
| Figure 7. An AIB Column.....  | 26 |
| Figure 8. AIB Data Signals .....  | 26 |
| Figure 9. MAC Interface .....   | 31 |
| Figure 10. SDR and DDR TX Blocks.....   | 34 |
| Figure 11. I/O Mapping: Transmit (Gen1 AIB Base).....   | 35 |
| Figure 12. I/O Mapping: Transmit (Gen1 AIB Plus, Gen2 AIB Base and AIB Plus use<br>DDR only)..... | 35 |
| Figure 13. SDR and DDR RX Blocks .....  | 36 |
| Figure 14. I/O Mapping: Receive (Gen1 AIB Base).....  | 36 |
| Figure 15. I/O Mapping: Receive (Gen1 AIB Plus, Gen2 AIB Base and AIB Plus use<br>DDR only).....  | 37 |
| Figure 16. SDR Data/Clock Timing .....  | 38 |
| Figure 17. DDR Data/Clock Timing .....  | 38 |
| Figure 18. Unit Intervals for SDR, DDR Clocks .....   | 39 |
| Figure 19. Skew Relationships.....  | 39 |
| Figure 20. Forwarded Clock – Transmit .....   | 41 |
| Figure 21. Forwarded Clock – Receive .....  | 42 |
| Figure 22. Receive-Domain Clock – Transmit (AIB Plus only, Gen1 mode only) .....                  | 43 |
| Figure 23. Receive-Domain Clock – Receive (AIB Plus only, Gen1 mode only) .....                   | 44 |
| Figure 24. Duty-Cycle Correction (AIB Plus only) .....  | 45 |
| Figure 25. Delay-Locked Loop .....  | 46 |
| Figure 26. Latency Measurement.....   | 47 |
| Figure 27. Asynchronous I/O Mode.....   | 47 |
| Figure 28. Mismatched Interfaces, One Channel .....   | 48 |
| Figure 29. Retiming Register.....   | 49 |

|  |     |
|--|-----|
| Figure 30. Adapter Phase Compensators (PC) .....                       | 50  |
| Figure 31. Word Marking Example, .....                                 | 52  |
| Figure 32. Word Assembly Example, .....                                | 53  |
| Figure 33. Sideband Control Shift Registers .....                      | 55  |
| Figure 34. Leader Sideband Control Shift Register .....                | 56  |
| Figure 35. Follower Sideband Control Shift Register .....              | 57  |
| Figure 36. Free-running Clock Generation Options .....                 | 58  |
| Figure 37. Sideband-Control Shift-Register Timing .....                | 59  |
| Figure 38. AIB Initialization .....                                    | 64  |
| Figure 39. Power-on reset synchronization .....                        | 65  |
| Figure 40. Power-on reset and device detect test provision .....       | 66  |
| Figure 41. Configuration completion signals. ....                      | 68  |
| Figure 42. Adapter Reset .....   | 69  |
| Figure 43. Free-running Clock Calibration State Machine .....          | 70  |
| Figure 44. Data-Path Calibration Architecture .....                    | 71  |
| Figure 45. Leader-to-Follower Datapath Calibration State Machine ..... | 74  |
| Figure 46. Follower-to-Leader Datapath Calibration State Machine ..... | 76  |
| Figure 47. Redundancy Routing .....                                    | 78  |
| Figure 48. Direction of Redundancy Signal Shift .....                  | 78  |
| Figure 49. Compliance Eye Mask .....                                   | 80  |
| Figure 50. Overshoot and Undershoot at the Receiver .....              | 82  |
| Figure 51. Near Side and Far Side Loopbacks .....                      | 85  |
| Figure 52. Interface Orientation .....                                 | 86  |
| Figure 53. Standard Chiplet-to-Chiplet Interconnection .....           | 87  |
| Figure 54. Rotated Chiplet-to-Chiplet Interconnection .....            | 88  |
| Figure 55. Bump Spacing for Microbump Array .....                      | 88  |
| Figure 56. Signal Relationships for AIB Base Configurations .....      | 90  |
| Figure 57. Signal Relationships for AIB Plus Configurations .....      | 91  |
| Figure 58. Bump Map Assignment Order .....                             | 106 |
| Figure 59. Bump Map Migration Between Different Bump Densities .....   | 107 |
| Figure 60. Low-Density Bump Map (AIB Base, 40 I/Os, Balanced) .....    | 113 |
| Figure 61. Low-Density Bump Map (AIB Base, 20 TX) .....                | 114 |
| Figure 62. Low-Density Bump Map (AIB Base, 20 RX) .....                | 114 |



|   |     |
|---|-----|
| Figure 63. Low-Density Bump Map (AIB Plus, 40 I/Os, Balanced).....                  | 115 |
| Figure 64. Low-Density Bump Map (AIB Plus, 20 TX).....                              | 116 |
| Figure 65. Low-Density Bump Map (AIB Plus, 20 RX) .....                             | 117 |
| Figure 66 Medium-Density Bump Map (AIB Plus, 40 I/Os, Balanced) .....               | 118 |
| Figure 67 Medium-Density Bump Map (AIB Plus, 80 I/Os, Balanced) .....               | 118 |
| Figure 68 High-Density Bump Map (AIB Plus, 40 I/Os, Balanced) .....                 | 119 |
| Figure 69. Low-Density Channel 0 and AUX: East Side.....                            | 119 |
| Figure 70. Low-Density Channel 0 and AUX: West Side.....                            | 120 |
| Figure 71. Low-Density Channel 0 and AUX: North Side .....                          | 121 |
| Figure 72. Low-Density Channel 0 and AUX: South Side .....                          | 122 |
| Figure 73. Channel Stacking: East Side.....   | 123 |
| Figure 74. Channel Stacking: West Side.....   | 123 |
| Figure 75. Channel Stacking: North Side .....                                       | 124 |
| Figure 76. Channel Stacking: South Side.....  | 124 |
| Figure 77. AIB-O Bump Spacing .....   | 128 |
| Figure 78. AIB-O Example Bump Map, 20 I/Os, Balanced.....                           | 130 |
| Figure 79. Stacking AIB-O channels with gap every six channels.....                 | 131 |
| Figure 80. AIB-O Sideband Control Shift Register Timing.....                        | 133 |
| Figure 81. Alternate Leader Bump Map .....  | 136 |
| Figure 82. Alternate Follower Bump Map .....  | 139 |
| Figure 83. Two die test case with independent pattern generators and checkers ..... | 143 |
| Figure 84. Two die test case using far side loopback.....                           | 143 |

## Table of Tables

---

|  |    |
|--|----|
| Table 1. AIB Gen1 and Gen2 .....                           | 17 |
| Table 2. Design Feature Summary.....                       | 18 |
| Table 3. AIB Data Rates.....                               | 20 |
| Table 4. AIB Interface Signals in AIB Channel .....        | 25 |
| Table 5. Number of Data Signals per Channel .....          | 25 |
| Table 6. Number of Channels per Column .....               | 25 |
| Table 7. MAC Interface .....                               | 29 |
| Table 8. Application Interface .....                       | 33 |
| Table 9. Skew Specifications .....                         | 38 |
| Table 10. Clock Duty-Cycle Requirements.....               | 45 |
| Table 11. Latency Specification .....                      | 46 |
| Table 12. PC PHY/MAC Interface Width Examples .....        | 51 |
| Table 13. Free-Running Clock Frequency.....                | 58 |
| Table 14. Sideband Control Signals .....                   | 61 |
| Table 15. Wired-AND Pull-Up Guidelines.....                | 68 |
| Table 16. Calibration Initiation Signals .....             | 72 |
| Table 17. Calibration Completion Signals .....             | 73 |
| Table 18. Leader-to-Follower Calibration Signals .....     | 75 |
| Table 19. Follower-to-Leader Calibration Signals .....     | 77 |
| Table 20. Electrical signal specifications .....           | 81 |
| Table 21. Overshoot and Undershoot Specifications .....    | 82 |
| Table 22. ESD Specifications.....                          | 82 |
| Table 23. Private JTAG instructions .....                  | 84 |
| Table 24. Bump Spacing Specifications for Bump Array ..... | 89 |
| Table 25. Bump Table: Starting.....                        | 92 |
| Table 26. Bump Table: Spare signals .....                  | 92 |
| Table 27. Bump Table: Inputs (AIB Base, Balanced) .....    | 92 |
| Table 28. Bump Table: Complete (AIB Base, Balanced).....   | 93 |
| Table 29. Bump Table: Inputs (AIB Base, All-TX) .....      | 94 |
| Table 30. Bump Table: Complete (AIB Base, All-TX).....     | 94 |
| Table 31. Bump Table: Inputs (AIB Base, All-RX).....       | 95 |

|   |            |
|---|------------|
| Table 32. Bump Table: Complete (AIB Base, All-RX) .....                       | 95         |
| Table 33. Bump Table: Inputs (AIB Plus, Balanced) .....                       | 96         |
| Table 34. Bump Table: Complete (AIB Plus, Balanced) .....                     | 97         |
| Table 35. Bump Table: Inputs (AIB Plus, All-TX) .....                         | 98         |
| Table 36. Bump Table: Complete (AIB Plus, All-TX) .....                       | 98         |
| Table 37. Bump Table: Inputs (AIB Plus, All-RX) .....                         | 99         |
| Table 38. Bump Table: Complete (AIB Plus, All-RX) .....                       | 100        |
| Table 39. Bump-Table Exemplar (AIB Base, Balanced) $x=n-10$ ; $y=2n+8$ .....  | 101        |
| Table 40. Bump-Table Exemplar (AIB Base, All-TX) $x=n-10$ .....               | 101        |
| Table 41. Bump-Table Exemplar (AIB Base, All-RX) $y=n+8$ .....                | 102        |
| Table 42. Bump-Table Exemplar (AIB Plus, Balanced) $x=n-12$ ; $y=2n+20$ ..... | 103        |
| Table 43. Bump-Table Exemplar (AIB Plus, All-TX) $x=n-12$ .....               | 104        |
| Table 44. Bump-Table Exemplar (AIB Plus, All-RX) $y=n+18$ .....               | 105        |
| Table 45. AIB AUX Signal Bump Table .....                                     | 108        |
| Table 46. Example Bump Table (AIB Base, 40 I/Os, Balanced) .....              | 108        |
| Table 47. Example Bump Table (AIB Base, 20 RX) .....                          | 109        |
| Table 48. Example Bump Table (AIB Base, 20 TX) .....                          | 109        |
| Table 49. Example Bump Table (AIB Plus, 40 I/Os, Balanced) .....              | 110        |
| Table 50. Example Bump Table (AIB Plus, 80 I/Os, Balanced) .....              | 111        |
| Table 51. Example Bump Table (AIB Plus, 20 TX) .....                          | 112        |
| Table 52. Example Bump Table (AIB Plus, 20 RX) .....                          | 112        |
| Table 53. AIB-O Number of Data Signals per Channel .....                      | 126        |
| Table 54. AIB-O Electrical signal specifications .....                        | 127        |
| Table 55. AIB-O Bump Spacing Specifications .....                             | 128        |
| <b>Table 56. AIB-O Example Bump Table, 20 I/Os, Balanced</b> .....            | <b>129</b> |
| Table 57. AIB-O Sideband Control Signals .....                                | 132        |
| Table 58. AIB-O Free-Running Clock Frequency .....                            | 133        |
| Table 59. Alternate Leader Channel Bump Table .....                           | 135        |
| Table 60. Alternate Follower Channel Bump Table .....                         | 138        |
| Table 61. Leader Sideband-Control Signals .....                               | 140        |
| Table 62. Follower Sideband-Control Signal .....                              | 141        |
| Table 63. Test Pattern Polynomials .....                                      | 142        |

## Glossary and Acronyms

---

This section defines phrases and acronyms that are not defined within the specification.

|                                  |  |
|----------------------------------|--|
| <b>assert</b>                    | To make a signal TRUE or active. For an active-high signal, this means asserting it HI; for an active-low signal, it means asserting it LO.  |
| <b>boustrophedon</b>             | Describes a back-and-forth motion. Literally means, "As the ox plows."   |
| <b>bump</b>                      | A die/package interconnect scheme that reflows solder bumps to make a connection from the die to the substrate to which it's being mounted. There may be a variety of sizes and means of implementing the bump. In this specification, bumps are implemented as microbumps. The terms "bump" and "microbump" are used interchangeably. |
| <b>CDM</b>                       | Charged-device model for ESD   |
| <b>chiplet</b>                   | A passivated bare die intended to be mounted on a substrate like an interposer.  |
| <b>compliance mask</b>           | Used to specify requirements on an eye diagram.  |
| <b>DCC</b>                       | Duty-cycle correction. Used to ensure that the duty cycle of a signal remains within specification.  |
| <b>DCD</b>                       | Duty-cycle distortion. Refers to effects that may result in a signal's duty cycle being out of specification.  |
| <b>DDR</b>                       | Double data rate. Data is captured on both edges of the clock.   |
| <b>de-assert</b>                 | To make a signal FALSE or inactive. For an active-high signal, this means de-asserting it LO; for an active-low signal, it means de-asserting it HI.   |
| <b>DLL</b>                       | Delay-locked loop. A circuit that can synthesize multiple frequencies and phase relationships from a single periodic signal. A digital version of a phase-locked loop (PLL).   |
| <b>quasi-differential signal</b> | A signal implemented over two wires having opposing polarities. Similar to analog differential signals, but created using a pair of digital signals.   |
| <b>ESD</b>                       | Electrostatic discharge  |

|                    |  |
|--------------------|--|
| <b>eye diagram</b> | Used to illustrate high-speed signal switching in a manner that allows specification and compliance testing of the signal characteristics.   |
| <b>handshake</b>   | A protocol that allows two sides of an interface to be connected, configured, and synchronized.  |
| <b>HI</b>          | A logic <i>high</i> level, equivalent to data value 1'b1. In the context of this specification, it is implemented with a high voltage value.   |
| <b>I/O</b>         | Input/output. In the context of this specification, refers to inputs and outputs collectively.   |
| <b>JTAG</b>        | Joint Test Action Group. Refers to a standard for the testing of chip-to-chip interconnections as well as select internal chip testing. The control and observability afforded by the standard may be used for more than just testing.                                     |
| <b>latency</b>     | The time required for a signal to traverse a given path. Equivalent to “delay.”  |
| <b>LO</b>          | A logic <i>low</i> level, equivalent to data value 1b0. In the context of this specification, it is implemented with a low voltage value.  |
| <b>LSB</b>         | Least-significant bit. In the context of a shift register, the bit with the lowest number.   |
| <b>MAC</b>         | Media Access Controller. The lowest-level of the Link Layer, which is the layer above the PHY layer in the OSI reference model.  |
| <b>microbump</b>   | A bump interconnect scheme with very small bumps. Suitable for mounting a die onto a substrate like an interposer, where fine lines can be created, but not for a printed circuit board. In this specification, the terms “bump” and “microbump” are used interchangeably. |
| <b>MSB</b>         | Most-significant bit. In the context of a shift register, the bit with the highest number.   |
| <b>OSI</b>         | Open Systems Interconnection. A project undertaken at the International Organization for Standardization (ISO), one outcome of which is the OSI reference model for networking.  |
| <b>overshoot</b>   | For the purposes of this specification, refers to a rising signal that exceeds the target static voltage before correcting itself  |

and ringing down to the target level.

|                          |   |
|--------------------------|---|
| <b>PHY</b>               | Physical layer. The bottom-most layer of the OSI reference model. It specifies electrical and basic signaling requirements.   |
| <b>redundancy</b>        | A technique for providing multiple instances of a resource for use in case of the failure of one of them. In the context of this specification, refers to techniques that can be implemented at or before testing and that are activated on power-up.               |
| <b>retiming</b>          | Refers to insertion or relocation of one or more flip-flops within a data path for the purpose of optimizing timing, power, and/or area.  |
| <b>RX</b>                | Receive. Applies to the inputs of a communications interface.   |
| <b>SDR</b>               | Single data rate. Data is captured on one edge of the clock.  |
| <b>sideband</b>          | Refers to signals that are not carried over the primary communication channels, but rather through auxiliary channels created specifically for those signals.   |
| <b>forwarded</b>         | In the context of a communications link, refers to a control signal – often the clock – sent from one side of the link to the other.  |
| <b>TX</b>                | Transmit. Applies to the outputs of a communications interface.   |
| <b>UI</b>                | Unit interval. The time, relative to the clock period, required to transmit a symbol. Allows timing specifications that are independent of a specific clock frequency.  |
| <b>undershoot</b>        | For the purposes of this specification, refers to a falling signal that goes below the target static voltage before correcting itself and ringing down to the target level.   |
| <b>weak pull-up/down</b> | An active or passive component connecting a signal to $V_{DD}$ /ground to ensure that the signal, when not driven, will not float. The signal will be pulled to a HI/LO value. The term “weak” indicates that the pull-up/down component will pass limited current. |

## Note on Language

---

In order to provide clarity on normative language as distinct from informative language, the following indicates the use of modal verbs:

- “Shall” indicates a requirement. Failure to meet the requirement will result in non-compliance.
- “May” indicates an option. Implementation of an option will result in compliance.
- “Should” indicates a strong suggestion for something outside the scope of the specification. Failure to implement the suggestion will not result in non-compliance.
- The lack of one of the above modal verbs indicates informative language.

# 1 Introduction

---

This specification describes the Advanced Interconnect Bus (AIB) architecture, interconnect attributes, signal management, and the configuration interface required to design and build systems and peripherals that are compliant with the AIB specification. The AIB is intended for interconnecting chiplets mounted within a package with signal distances of 10 millimeters or less (often referred to as Ultra-Short Reach). While no maximum interconnect distance is specified, AIB signal electrical requirements detailed in Section 4 shall be met. AIB is not intended for signals interconnecting or connecting to external package pins.

The specification is intended to provide interoperation between compliant chiplets. Choices made by the chiplet designer with respect to such elements as number and type of data signals, maximum supported clock speed, and any functionality that exceeds the minimum requirements established in the AIB specification should be documented in the chiplet datasheet.

AIB is a physical interconnect. Link, protocol and application layers are implemented on top of the AIB interface.

## 1.1 A Second Generation Die to Die Interconnect

The new high-level requirements of this second generation interconnect are:

- Higher bandwidth per pin to meet the increasing requirements of applications
- Reduced I/O voltage swing on data and data clocks for lower energy per bit to contain power and thermal impacts
- Compatibility with first generation AIB interfaces

Second generation AIB interfaces operate in either a Second Generation (Gen2) mode or a First Generation (Gen1) mode. Where Gen2 specifications are different than Gen1, the specifications are denoted with a Gen2 or Gen1 label.

Table 1 summarizes the new high-level requirements of Gen2.



| Requirement             | AIB Gen1                                      | AIB Gen2                   |
|-------------------------|---|----------------------------|
| Data Rate               | Up to 2Gbps                                   | Up to 6.4Gbps              |
| IO Voltage Output Swing | 0.9V  | 0.4V for data in Gen2 mode |
| Electrical and Timing   | See Table 20 Electrical Signal Specifications |                            |
| Backward Compatibility  | n/a   | Gen1                       |

**Table 1. AIB Gen1 and Gen2**

## 1.2 Compliance Summary

Table 2 summarizes the compliance points that shall be met in order to meet the AIB requirements. Each of the compliance points is discussed in the specification.

| Compliance Point                 | Required |          | Optional |
|----------------------------------|----------|----------|----------|
|                                  | AIB Base | AIB Plus |          |
| AC Parameters (Section 4)        | x        | x        |          |
| JTAG boundary scan (Section 5.1) | x        | x        |          |
| Redundancy (Section 3.2.4)       | x        | x        |          |
| Bump assignment (Section 6.3)    | x        | x        |          |
| AUX (Section 1.3.4.4)            | x        | x        |          |
| SDR (Section 2.1.3.1)            | x [1]    | x [1]    |          |
| DDR (Section 2.1.3.2)            | x [2]    | x        |          |
| Latency specification            | x        | x        |          |

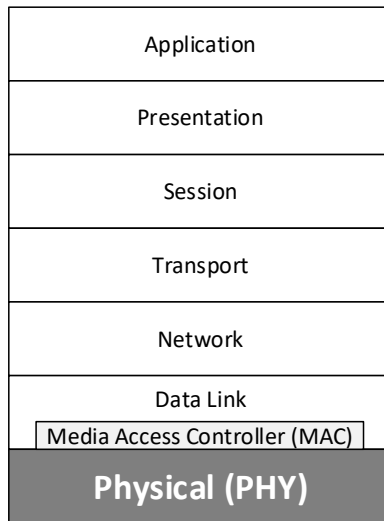
| Compliance Point                             | Required |          | Optional |
|--|----------|----------|----------|
|  | AIB Base | AIB Plus |          |
| (Section 2.1.7)                              |          |          |          |
| Free-running clock<br>(Section 2.2.3.2)      |          | x        |          |
| AIB Adapter<br>(Section 2.2)                 |          | x        |          |
| Data-transfer ready<br>(Section 3.1)         | x        | x        |          |
| Adapter reset<br>(Section 3.2.3.1.1)         |          | x        |          |
| DLL<br>(Section 2.1.6.7)                     |          |          | x        |
| DCC<br>(Section 2.1.6.6)                     |          |          | x        |
| Adapter retiming register<br>(Section 2.2.1) |          | x        |          |
| Conf_done<br>(Section 3.2.2.3.5)             | x        | x        |          |

1. In Gen2 mode, SDR is only supported for the Sideband Shift Register data (section 2.2.2).
2. Gen1 AIB Base supports only SDR data transfer. In Gen2 mode AIB Base and AIB Plus support only DDR data transfer.

**Table 2. Design Feature Summary**

## 1.3 Architecture

The AIB implements a physical-layer, or PHY, interconnect scheme, occupying Layer 1 on the OSI Reference Model.



**Figure 1. AIB in the OSI Reference Model**

### 1.3.1 AIB Configurations

There are two AIB configurations: AIB Base and AIB Plus. AIB Base is a simpler interface; AIB Plus provides features to achieve maximum performance.

AIB Base and AIB Plus configurations are not intended to be interconnected as two sides of a link due to signals required by AIB Plus configurations that are not present in AIB Base configurations.

There is no maximum data rate specified for AIB. In the interest of interoperability, minimum rates are specified in Table 3 along with typical rates to provide a range likely to be supported by multiple chiplets. The full range of the operating data rates should be documented in the chiplet data sheet.

#### 1.3.1.1 AIB Base

In Gen1 mode an AIB Base implementation shall transfer data using a single data rate (SDR) format (Section 2.1.3.1). In Gen2 mode an AIB Base implementation shall transfer data using a double data rate (DDR) format (Section 2.1.3.2).

#### 1.3.1.2 AIB Plus

In Gen1 mode an AIB Plus implementation shall transfer data using a DDR format and a SDR format. In Gen2 mode an AIB Plus implementation shall transfer data using a DDR format. AIB Plus implementations shall include an AIB Adapter block, specified in Section 2.2. A primary function of the AIB Adapter is to provide a standard means for initialization of a channel such as handshaking on DLL calibration.

| Data Format | Gen1 Supports Data Rates Up To This Minimum | Gen1 Typical Data Rates | Gen2 Supports Data Rates Up To This Minimum | Gen2 Typical Data Rates |
|-------------|---|-------------------------|---|-------------------------|
| SDR         | 500 Mbps                                    | 1Gbps                   | n/a   | n/a                     |
| DDR         | 1 Gbps                                      | 2Gbps                   | 2 Gbps                                      | 4Gbps, 6.4Gbps          |

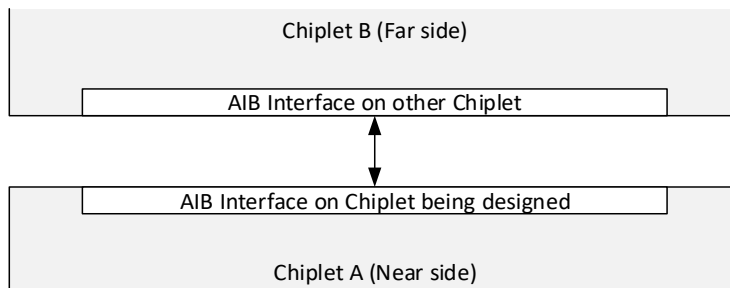
**Table 3. AIB Data Rates**

### 1.3.2 Near-Side and Far-Side Interfaces

AIB specifies an interface for one chiplet that can be connected to a compatible interface on a different chiplet. An AIB interface is one side of the connection. A chiplet designer may be creating only one of those AIB interfaces; the AIB interface to which it will connect is assumed to have been created by a different designer, putting it beyond the control of the designer.

In order to provide a clear distinction between behavior that the designer shall implement and behavior that the designer shall expect from the AIB interface to which it will connect, the AIB interface being created will be referred to as the Near-Side interface. The AIB interface to which the near-side interface will connect will be called the Far-Side interface. Signal names that reflect the signal origin will be prefixed with *ns\_* or *fs\_*, respectively.

In the example of Figure 2, Chiplet A is the subject chiplet under design. Chiplet A will eventually connect to Chiplet B. From the reference of Chiplet A, Chiplet A is the near side and Chiplet B is the far side.



**Figure 2. Near Side and Far Side**

### 1.3.3 Leader, Follower, and Dual-Mode Interfaces

An AIB interface pair has a leader side and a follower side. Leaders and followers shall have specific roles only during initialization. Specifically:

- The leader shall be responsible for providing a free-running clock signal (Section

2.2.3.2).

- The leader shall provide a *device\_detect* signal (Section 3.2.1).
- The follower shall provide a *power\_on\_reset* signal (Section 3.2.1)
- The leader and follower shall have differently sized Sideband Control Signal shift registers (Section 2.2.2).
- The AIB Adapter (AIB Plus only) shall behave differently for leaders and followers during initialization (Section 2.2).

An AIB interface configured as a leader shall be designed to connect to a follower; an AIB interface configured as a follower shall be designed to connect to a leader.

The leader/follower property of an interface is independent of the near-side/far-side property. A near-side interface can be either a leader or follower, as can a far-side interface.

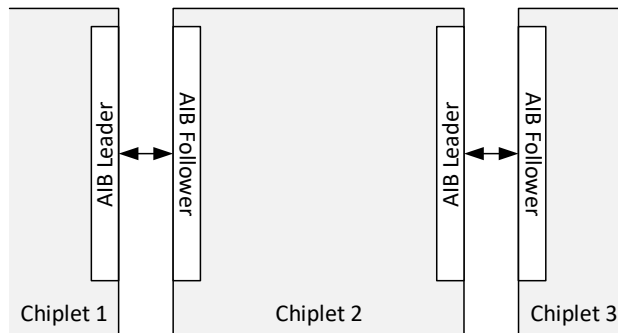
An AIB interface shall be configured as leader or follower by one of the following two means:

- By designing the interface as a leader or follower, referred to as a fixed interface.
- By implementing a dual-mode interface that can be configured as leader or follower on power-up.

#### 1.3.3.1 Fixed Interfaces

For fixed interfaces, the configuration of an interface as leader or follower shall be implemented by the chiplet designer and should be documented in the chiplet data sheet.

A chiplet may have one or more leaders, one or more followers, or a mixture of leaders and followers.

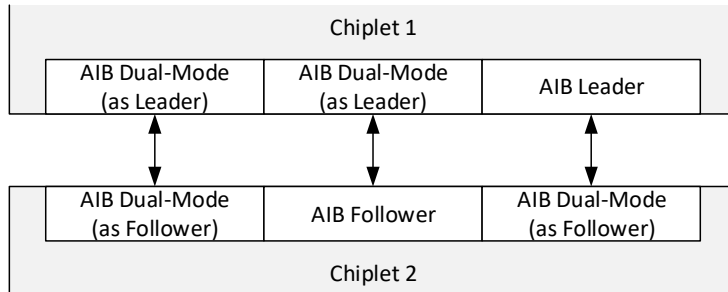


**Figure 3. Fixed Leader and Follower Interfaces**

#### 1.3.3.2 Dual-Mode Interfaces

A chiplet may implement both leader and follower capability on an AIB interface, with leader or follower configured prior to operation. This promotes greater interoperability, since the chiplet may connect to chiplets that have either leader or follower AIB interfaces. Such interfaces are referred to as *dual-mode* interfaces. Any actions

required to configure the dual-mode interface should be documented in the chiplet data sheet.



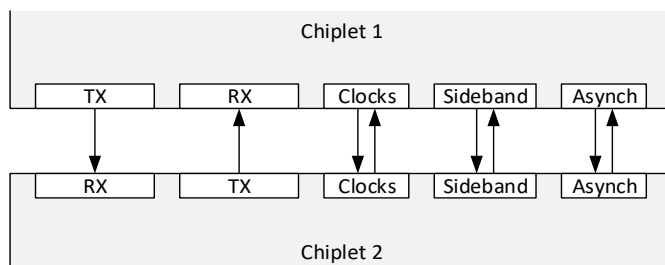
**Figure 4. Dual-Mode Interfaces**

Dual-mode interfaces shall implement a *dual\_mode\_select* signal that configures the interface either as a leader or as a follower. The interface shall be configured as a leader when the *dual\_mode\_select* signal is HI and as a follower when the *dual\_mode\_select* signal is LO. The selected mode shall be configured and stable before the end of the power-on reset phase (Section 3.2.1) and not during the configuration phase (Section 3.2.2).

### 1.3.4 AIB Interface

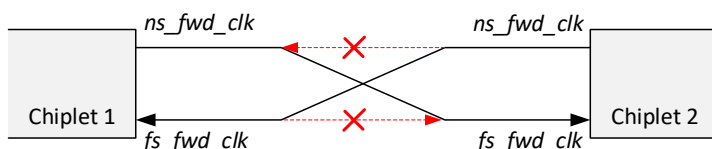
The AIB interface defines four signal types:

- Data signals
  - Inputs (RX): data input signals received by the interface
  - Outputs (TX): data output signals transmitted from the interface
- Clocks
  - Data clock out (*ns\_fwd\_clk*, *ns\_rcv\_clk*), sent by the near side chiplet to the far side chiplet
  - Data clock in (*fs\_fwd\_clk*, *fs\_rcv\_clk*): received by the near side chiplet from the far side chiplet
  - Free-running clock (AIB Plus only) (*ns\_sr\_clk*, *fs\_sr\_clk*): used by the sideband control signals
- Sideband control (AIB Plus only): used to implement calibration handshake.
- Asynchronous
  - *Power-on reset*: indicates whether a chiplet has completed power-on reset
  - *Device\_detect*: used to verify presence of the leader
  - *ns\_mac\_rdy*, *fs\_mac\_rdy*: used to communicate that the MAC is ready for data transmission
  - *ns\_adapter\_rstn*, *fs\_adapter\_rstn* (AIB Plus only): used to reset the AIB Adapter registers, the AIB I/O registers, and the calibration circuits.



**Figure 5. AIB Signal Types**

Two interconnected chiplets will have the same set of signals. Signals from the near and far sides will be connected, such as *ns\_fwd\_clk* and *fs\_fwd\_clk*. Near-side signals should not be interconnected (e.g., *ns\_fwd\_clk* from one chiplet should not be connected to *ns\_fwd\_clk* from the other chiplet); far-side signals should not be interconnected (e.g., *fs\_fwd\_clk* from one chiplet should not be connected to *fs\_fwd\_clk* from the other chiplet). This is indicated in Figure 6, but the crossed wires are for illustration only. Microbump locations (Section 6.3) ensure that connections will be straight lines.



**Figure 6. Interconnecting Near-Side and Far-Side Signals**

| Signal                        | Description   | Present in AIB Base | Present in AIB Plus |
|-------------------------------|---|---------------------|---------------------|
| <i>TX</i>                     | Synchronous data transmitted from the near side. (Section 2.1.1)  | X                   | X                   |
| <i>RX</i>                     | Synchronous data received from the far side. (Section 2.1.2)  | X                   | X                   |
| <i>ns_fwd_clk/ns_fwd_clkb</i> | Near-side transfer clock, forwarded from the near side to the far side for capturing received data. (Section 2.1.6.3)   | X                   | X                   |
| <i>fs_fwd_clk/fs_fwd_clkb</i> | Far-side transfer clock, forwarded from the far side to the near side for capturing received data. (Section 2.1.6.3)    | X                   | X                   |
| <i>ns_rcv_clk/ns_rcv_clkb</i> | Receive-domain clock forwarded from the near side to the far side for transmitting data from the far side. In Gen2 mode |                     | X                   |

| Signal                        | Description   | Present in AIB Base | Present in AIB Plus |
|-------------------------------|---|---------------------|---------------------|
|                               | these outputs are not used and shall be configured with a weak pull-down. (Section 2.1.6.4)   |                     |                     |
| <i>fs_rcv_clk/fs_rcv_clkb</i> | Receive-domain clock forwarded from the far side to the near side for transmitting data from the near side. In Gen2 mode these inputs are not used and shall be configured with a weak pull-down. (Section 2.1.6.4)     |                     | X                   |
| <i>ns_sr_clk/ns_sr_clkb</i>   | Forwarded serial shift register clock from near side to far side chiplet, driven by free running clock. In Gen2 mode the ns_sr_clkb output is not used and shall be configured with a weak pull-down. (Section 2.2.3.2) |                     | X                   |
| <i>fs_sr_clk/fs_sr_clkb</i>   | Forwarded serial shift register clock from far side to near side chiplet, driven by free running clock. In Gen2 mode the fs_sr_clkb input is not used and shall be configured with a weak pull-down. (Section 2.2.3.2)  |                     | X                   |
| <i>ns_sr_data</i>             | Time-multiplexed sideband-control data from near side to far side. (Section 2.2.3.5)  |                     | X                   |
| <i>fs_sr_data</i>             | Time-multiplexed sideband-control data from far side to near side. (Section 2.2.3.5)  |                     | X                   |
| <i>ns_sr_load</i>             | Sideband control load signal from near side to far side. (Section 2.2.3.3)  |                     | X                   |
| <i>fs_sr_load</i>             | Sideband control load control signal from far side to near side. (Section 2.2.3.3)  |                     | X                   |
| <i>ns_mac_rdy</i>             | Data-transfer-ready signal from near side to far side. (Section 3.1).   | X                   | X                   |
| <i>fs_mac_rdy</i>             | Data-transfer-ready signal from far side to near side. (Section 3.1).   | X                   | X                   |
| <i>ns_adapter_rstn</i>        | Asynchronous adapter reset signal from near side to far side (Section 3.2.3.3.1)  |                     | X                   |



| Signal                 | Description   | Present in AIB Base | Present in AIB Plus |
|------------------------|---|---------------------|---------------------|
| <i>fs_adapter_rstn</i> | Asynchronous adapter reset signal from far side to near side. (Section 3.2.3.3.1) |                     | X                   |

**Table 4. AIB Interface Signals in AIB Channel**

#### 1.3.4.1 AIB Channel

AIB signals shall be grouped into AIB Channels. An AIB channel shall have a number of data signals not to exceed a limit determined by the microbump pitch of the chiplet.

The data signals may consist of half TX and half RX signals ("balanced" interface), all TX signals ("all-TX" interface) or all RX signals ("all-RX" interface).

| Microbump pitch (μm) | Range of data signals per channel (Increments) |                |        |                |        |                |
|----------------------|--|----------------|--------|----------------|--------|----------------|
|                      | TX   |                |        | RX             |        |                |
|                      | Balanced                                       | All-TX         | All-RX | Balanced       | All-TX | All-RX         |
| ≤55                  | 20-80<br>(20)                                  | 20-160<br>(20) | 0      | 20-80<br>(20)  | 0      | 20-160<br>(20) |
| 10                   | 20-320<br>(20)                                 | 20-640<br>(20) | 0      | 20-320<br>(20) | 0      | 20-640<br>(20) |

**Table 5. Number of Data Signals per Channel**

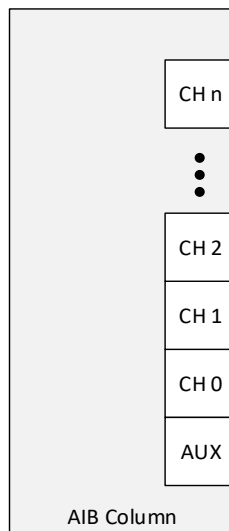
#### 1.3.4.2 AIB Column

AIB channels shall be grouped into an AIB column. All AIB channels within a column shall have the same configuration (Base or Plus) and the same number of data signals.

A column may include an AUX block adjacent to the first channel.

| Block type | Number allowed per column |
|------------|---------------------------|
| Channels   | 1, 2, 4, 8, 12, 16, 24    |
| AUX blocks | 1                         |

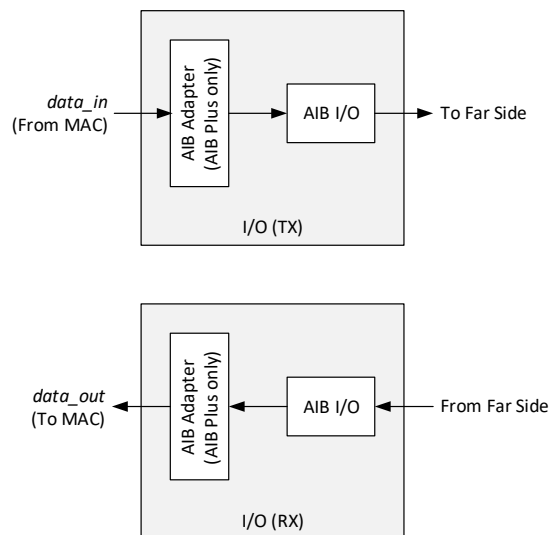
**Table 6. Number of Channels per Column**



**Figure 7. An AIB Column**

#### 1.3.4.3 AIB data paths

Each data path within a channel shall be composed of an I/O Block (Section 2.1) and, for AIB Plus implementations, an AIB Adapter (Section 2.2).



**Figure 8. AIB Data Signals**

#### **1.3.4.4 AUX Block**

The AUX block shall consist of two signals: *power\_on\_reset* and *device\_detect*. These signals shall be used during power-up initialization (Section 3.2.1). There shall be one AUX block per column.

The AUX block may use microbump IOs below the first channel or may use other IOs associated with the AIB interface.

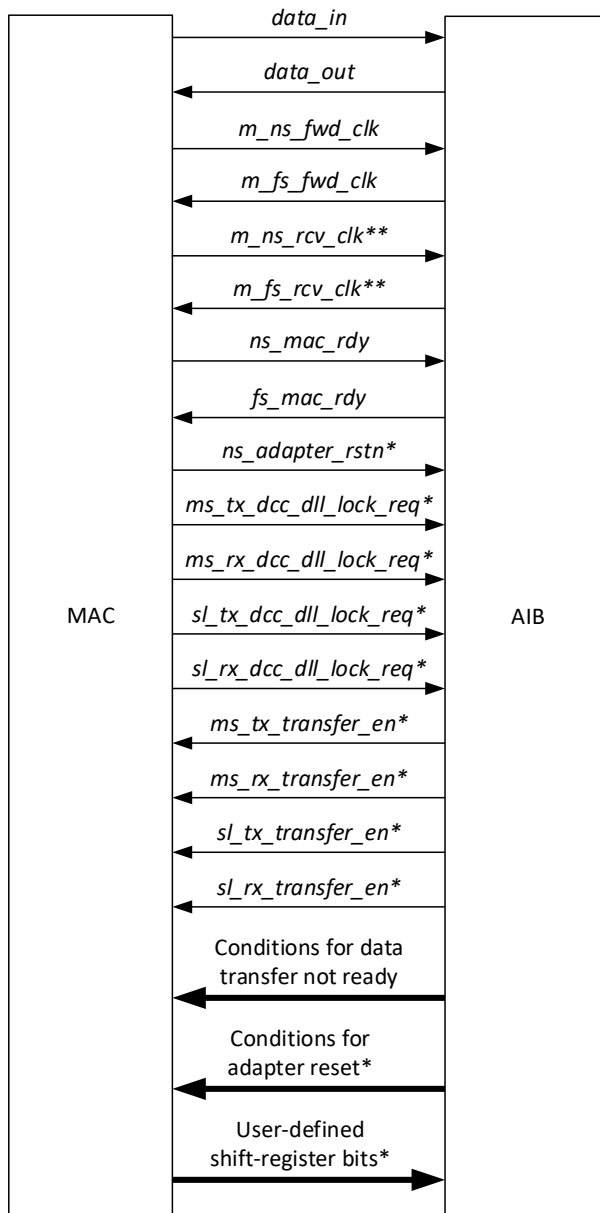
#### **1.3.5 AIB-to-MAC Interface**

The following per-AIB-channel signals shall constitute the interface to the MAC.

| Signals                                      | In (from MAC)<br>Out (to MAC) | Description   |
|--|-------------------------------|---|
| <i>data_in</i>                               | In                            | For transmitting across the AIB link (Section 2.1.1)  |
| <i>data_out</i>                              | Out                           | Received through the AIB link (Section 2.1.2)   |
| <i>data_in_f</i> (AIB Plus and Gen2 only)    | In                            | For transmitting across the AIB link using the AIB Adapter Phase Compensator (Section 2.2.2)  |
| <i>data_out_f</i> (AIB Plus and Gen2 only)   | Out                           | Received through the AIB link using the AIB Adapter Phase Compensator (Section 2.2.2)   |
| <i>m_wr_clk</i>                              | In                            | Clocks <i>data_in_f</i>   |
| <i>m_rd_clk</i>                              | In                            | Clocks <i>data_out_f</i>  |
| <i>m_ns_fwd_clk</i> (AIB Plus and Gen2 only) | In                            | For transmitting data from the near side to the far side (Section 2.1.6)  |
| <i>m_fs_fwd_clk</i>                          | Out                           | Received from the far side and converted from quasi-differential to single-ended (Section 2.1.6.3)  |
| <i>m_ns_rcv_clk</i> (AIB Plus only)          | In                            | Receive-domain clock forwarded from the near side to the far side for transmitting data from the far side (Section 2.1.6.4). In Gen2 mode this input is ignored by the PHY.               |
| <i>m_fs_rcv_clk</i> (AIB Plus only)          | Out                           | Received from the far side and converted from quasi-differential to single-ended (Section 2.1.6.4). In Gen2 mode this output should be ignored by the MAC.                                |
| <i>ns_mac_rdy</i>                            | In                            | For resetting near-side data transfers and communicating MAC readiness for calibration to the far side (Section 3.1). Logically equivalent to the <i>ns_mac_rdy</i> AIB interface signal. |

|  |     |  |
|--|-----|--|
| <i>fs_mac_rdy</i>  | Out | Indicates that the far-side MAC is ready to transmit data (Section 3.1). Logically equivalent to the <i>fs_mac_rdy</i> AIB interface signal. |
| <i>ns_adapter_rstn</i><br>(AIB Plus only)  | In  | Resets the AIB Adapter (section 3.1)   |
| <i>ms_rx_dcc_dll_lock_req</i><br><i>ms_tx_dcc_dll_lock_req</i><br>(AIB Plus only)                    | In  | Initiates calibration of transmit and receive paths for a leader interface (Section 3.2.3.3)   |
| <i>sl_rx_dcc_dll_lock_req</i><br><i>sl_tx_dcc_dll_lock_req</i><br>(AIB Plus only)                    | In  | Initiates calibration of transmit and receive paths for a follower interface (Section 3.2.3.3)   |
| <i>ms_tx_transfer_en</i><br><i>ms_rx_transfer_en</i><br>(AIB Plus only)                              | Out | Indicate that calibration on the leader is complete for transmit and receive paths (Section 3.2.3.3)   |
| <i>sl_tx_transfer_en</i><br><i>sl_rx_transfer_en</i><br>(AIB Plus only)                              | Out | Indicate that calibration on the follower is complete for transmit and receive paths (Section 3.2.3.3)                                       |
| <i>m_rx_align_done</i><br>(AIB Plus only, Gen2 only)   | Out | Indicates that the receiving AIB Adapter is aligned to incoming word marked data (Section 2.2.2.2)   |
| Signals indicating any conditions that may cause de-assertion of data-transfer ready                 | Out | Sent to MAC for possible data-transfer ready de-assertion by MAC (Section 3.1)   |
| Signals indicating any conditions that may cause AIB Adapter reset and recalibration (AIB Plus only) | Out | Sent to MAC for possible adapter reset by MAC (Section 3.2.3)  |
| User-defined shift-register bits<br>(AIB Plus only)  | In  | For transmitting application-defined bits to the far side (Section 2.2.3.5.2)  |
| <i>i_osc_clk</i>   | In  | Free running oscillator clock for a leader interface   |

**Table 7. MAC Interface**



\*AIB Plus only \*\*Gen1 AIB Plus only

**Figure 9. MAC Interface**

### **1.3.6 AIB-to-Application Interface**

The following per-AIB-interface signals shall constitute the interface to the application.

| Signals                     | In (from Application)<br>Out (to Application) | Description   |
|-----------------------------|---|---|
| <i>i_m_power_on_reset</i>   | Unused (Leader)<br>In (Follower)              | Follower: Controls the power_on_reset signal sent to the Leader. Must be stable at power-up.  |
| <i>o_m_power_on_reset</i>   | Out (Leader)<br>Unused (Follower)             | Leader: A copy of the power_on_reset signal from the Follower, qualified by override from m_por_ovrd.   |
| <i>m_por_ovrd</i>           | In (Leader)<br>Unused (Follower)              | Intended for standalone test without an AIB partner.<br>If LO, the Leader is not in reset.<br>If HI and the AIB interface signal power_on_reset input is 1, the Leader is held in reset. The Leader's power_on_reset input is required to have a weak pullup so that a no connect on that input will result in power_on_reset=HI. |
| <i>m_device_detect</i>      | Unused (Leader)<br>Out (Follower)             | A copy of the AIB interface signal device_detect from the Leader, qualified by m_device_detect_ovrd.  |
| <i>m_device_detect_ovrd</i> | Unused (Leader)<br>In (Follower)              | Intended for standalone test without an AIB partner.<br>If LO, the Follower uses the AIB interface device_detect input.<br>If HI, the Follower outputs m_device_detect=1.   |



|                         |    |  |
|-------------------------|----|--|
| <i>dual_mode_select</i> | In | Used with AIB interfaces that implement dual-mode (section 2.2.3.1.3). Must be stable at power up as this determines if the AIB interface receives power_on_reset (Leader) or drives power_on_reset (Follower). See section 3.2.1.1.<br><br>If LO the AIB interface is a Follower.<br>If HI the AIB interface is a Leader. |
| <i>m_gen2_mode</i>      | In | If LO when the AIB interface is released from reset, the AIB interface is in Gen1 mode.<br><br>If HI when the AIB interface is released from reset, the AIB interface is in Gen2 mode.   |
| <i>i_conf_done</i>      | In | Single control to reset all AIB channels' adapters in the interface. LO=reset, HI=out of reset. This input to the AIB PHY should be the same value as the chiplet's CONF_DONE pin. See section 3.2.2.3.5.  |

**Table 8. Application Interface**

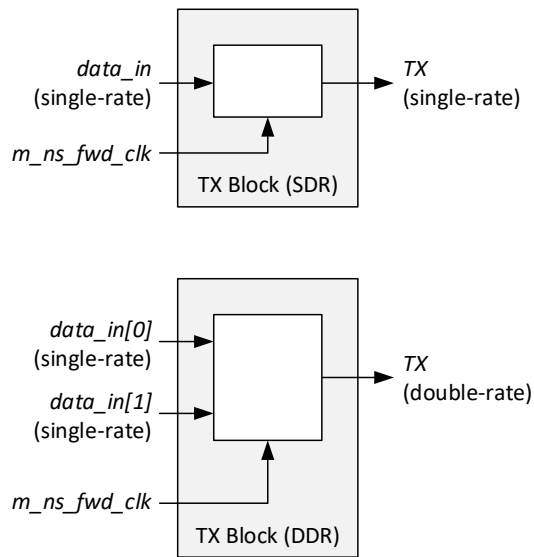
## 2 Functional Specification

### 2.1 I/O Blocks

I/O blocks are divided into one of two types: *RX* or *TX*.

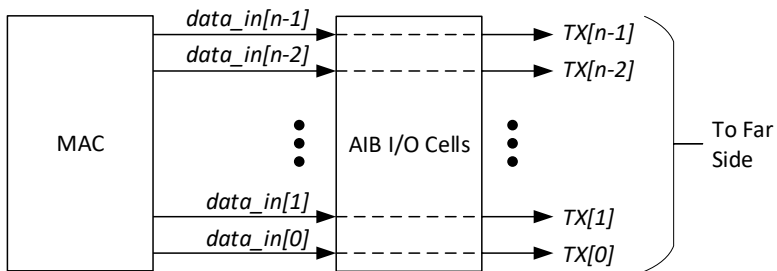
#### 2.1.1 TX Block

A TX block shall register data before transmission. A DDR output shall combine two single-rate data streams into one double-rate stream.

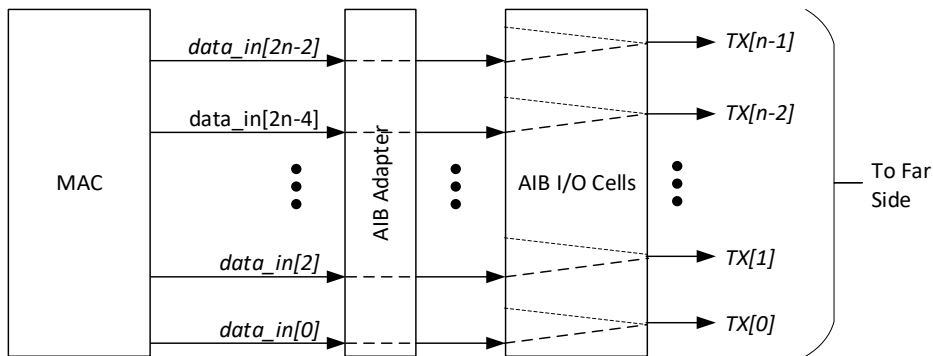


**Figure 10. SDR and DDR TX Blocks**

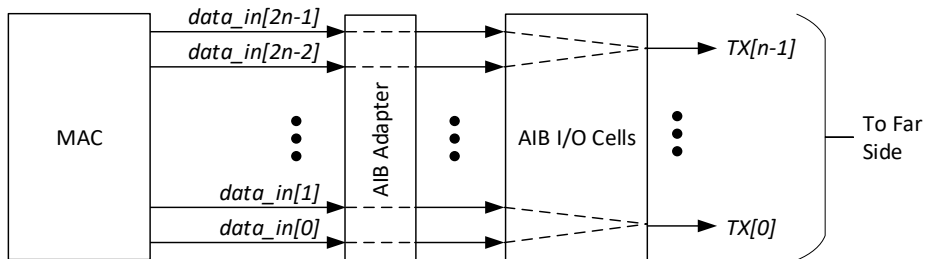
Inputs to the I/O block shall be mapped according to Figure 11 and Figure 12. For DDR signaling, two AIB I/O inputs are multiplexed onto one output (for example, inputs 0 and 1 are mapped to output 0). For a DDR-capable I/O, all DDR internal signals (double the number of *TX* signals) shall be implemented. If the I/O block is configured as SDR, then half of those internal signals shall not be used.



**Figure 11. I/O Mapping: Transmit  
(Gen1 AIB Base)**



**SDR**

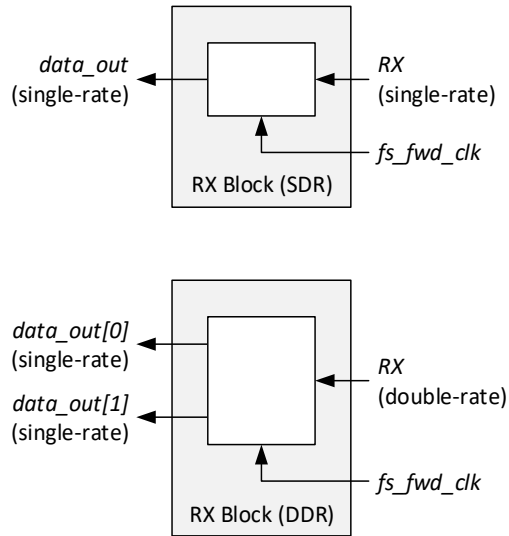


**DDR**

**Figure 12. I/O Mapping: Transmit  
(Gen1 AIB Plus, Gen2 AIB Base and AIB Plus use DDR only)**

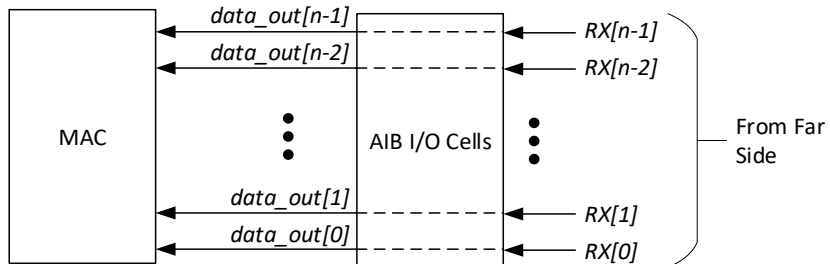
## 2.1.2 RX Block

An RX block shall register incoming data using the forwarded clock. For a double-data-rate stream data shall be clocked into one of two registers, one for each edge of the clock.

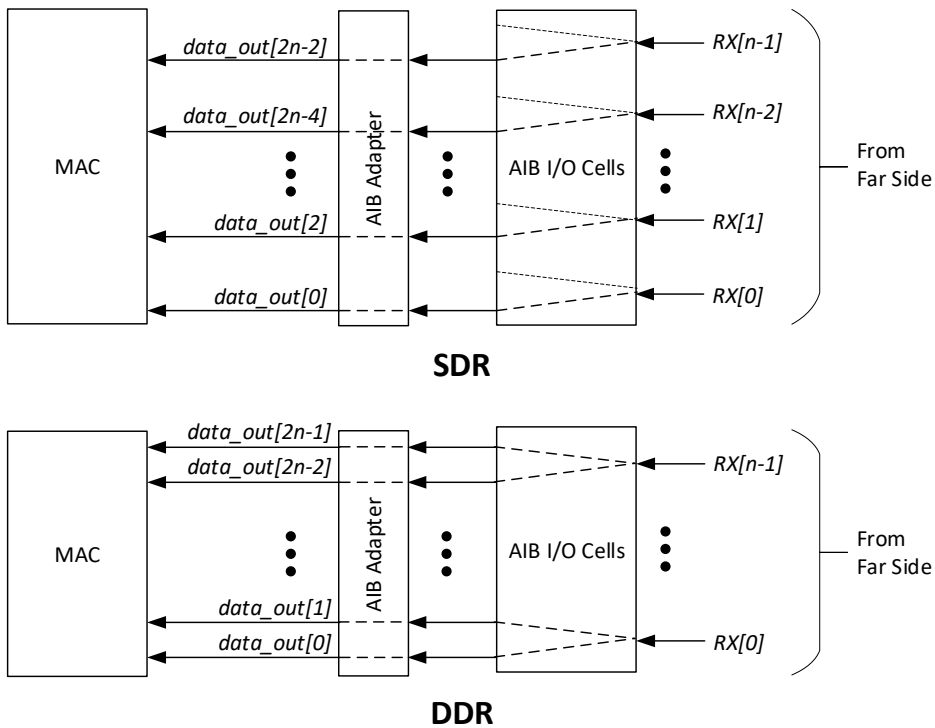


**Figure 13. SDR and DDR RX Blocks**

Outputs from the I/O block shall be mapped according to Figure 14 and Figure 15. For DDR signaling, one AIB I/O input is split into two AIB Adapter signals (for example, input 0 is mapped to AIB Adapter inputs 0 and 1). For a DDR-capable I/O (AIB Plus only), all DDR internal signals (double the number of RX signals) shall be implemented. If the I/O block is configured as SDR, then half of those internal signals shall not be used.



**Figure 14. I/O Mapping: Receive (Gen1 AIB Base)**



**Figure 15. I/O Mapping: Receive**  
(Gen1 AIB Plus, Gen2 AIB Base and AIB Plus use DDR only)

### 2.1.3 Data Exchange

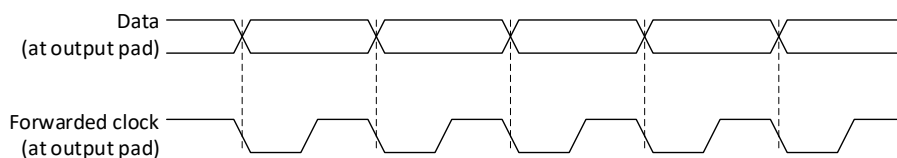
AIB data signals shall be exchanged either as SDR (Gen1) or DDR (Gen1 AIB Plus or Gen2).

All data signals within an AIB channel shall use the same data exchange format.

#### 2.1.3.1 SDR Data Exchange

Gen1 AIB Base interfaces shall exchange data using a SDR relationship between the clock and data. Data shall be transmitted on the falling edge of the clock.

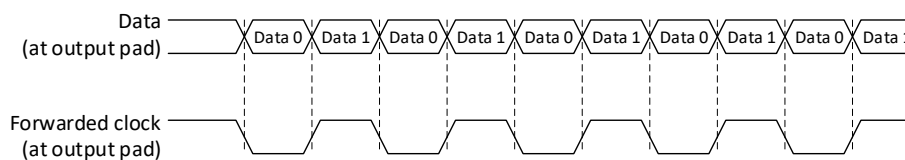
Gen1 AIB Plus implementations shall also implement SDR data signals.



**Figure 16. SDR Data/Clock Timing**

### 2.1.3.2 DDR Data Exchange

Gen1 AIB Plus and Gen2 interfaces shall be capable of exchanging data using a DDR relationship between the clock and data. When using DDR exchange, data shall be transmitted on both rising and falling edges of the clock.



**Figure 17. DDR Data/Clock Timing**

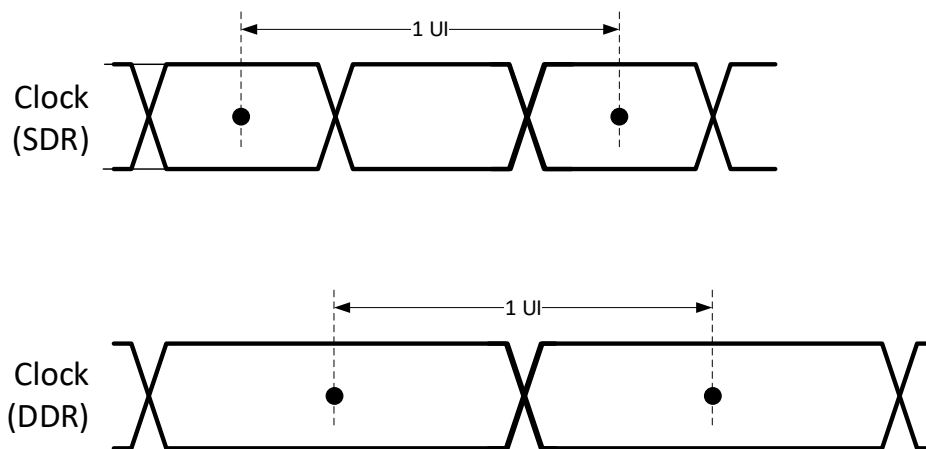
An AIB I/O shall be capable of both SDR and DDR data exchange regardless of which option is selected for a specific application.

### 2.1.3.3 Signal Skew

Skew between data edges within a channel and between clock and data edges within a channel shall meet the following specification, where a unit interval (UI) is illustrated in Figure 18.

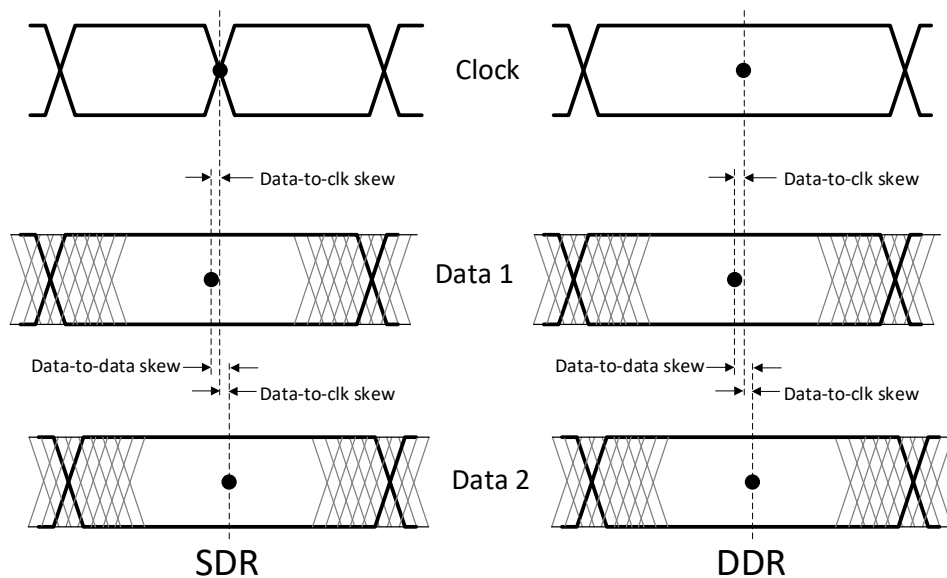
| Symbol    | Parameter                  | Gen1                         |                            | Gen2                         |                            |
|-----------|----------------------------|------------------------------|----------------------------|------------------------------|----------------------------|
|           |                            | Measured at near-side output | Measured at far-side input | Measured at near-side output | Measured at far-side input |
| $t_{DS}$  | Maximum data-to-data skew  | 0.04 UI                      | 0.06 UI                    | 0.04 UI                      | 0.05 UI                    |
| $t_{DCS}$ | Maximum data-to-clock skew | 0.02 UI                      | 0.03 UI                    | 0.02 UI                      | 0.025 UI                   |

**Table 9. Skew Specifications**



**Figure 18. Unit Intervals for SDR, DDR Clocks**

Skew relationships are illustrated in Figure 19. The relationships and specifications shall be met by *TX* signals, *ns\_fwd\_clk*, and *ns\_fwd\_clkb*, and by *ns\_sr\_clk*, *ns\_sr\_clkb* (*ns\_sr\_clkb* is Gen1 only), *ns\_sr\_data*, and *ns\_sr\_load*.



**Figure 19. Skew Relationships**

## 2.1.4 Tristate

All output signals shall be capable of being put into tristate.

## 2.1.5 Weak Pull-Up and Pull-Down

All inputs and outputs shall have an associated weak pull-up and weak pull-down with the equivalent driving strength of 10 – 20 kΩ. The inputs and outputs shall be configurable either with the pull-up, with the pull-down, or with neither the pull-up nor the pull-down.

## 2.1.6 Data-clock operation

### 2.1.6.1 Transmit Clock

The AIB layer shall receive a transmit clock, *m\_ns\_fwd\_clk*, from the MAC. That clock shall be used to transmit data and it shall be forwarded to the far side (Section 2.1.6.3). AIB Plus implementations may receive a receive-domain clock from the far side for use as a transmit clock source (Section 2.1.6.4).

### 2.1.6.2 Receive Clock (AIB Plus only, Gen1 mode only)

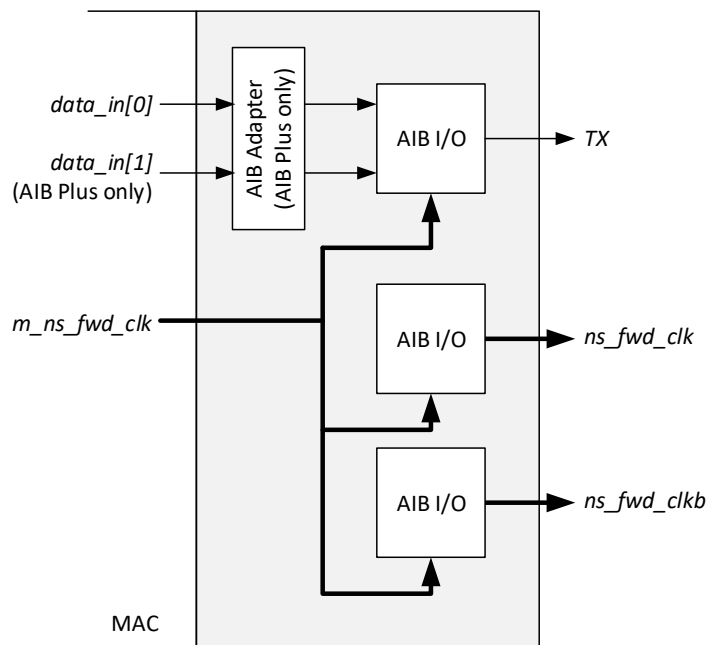
In AIB Plus implementations, the AIB layer shall receive a receive clock, *m\_ns\_rcv\_clk*, from the MAC. That clock may optionally be sent to the far side for its use in transmitting data to the near side (Section 2.1.6.4). In Gen2 mode the receive clock is not used.

### 2.1.6.3 Forwarded Transmit Clock

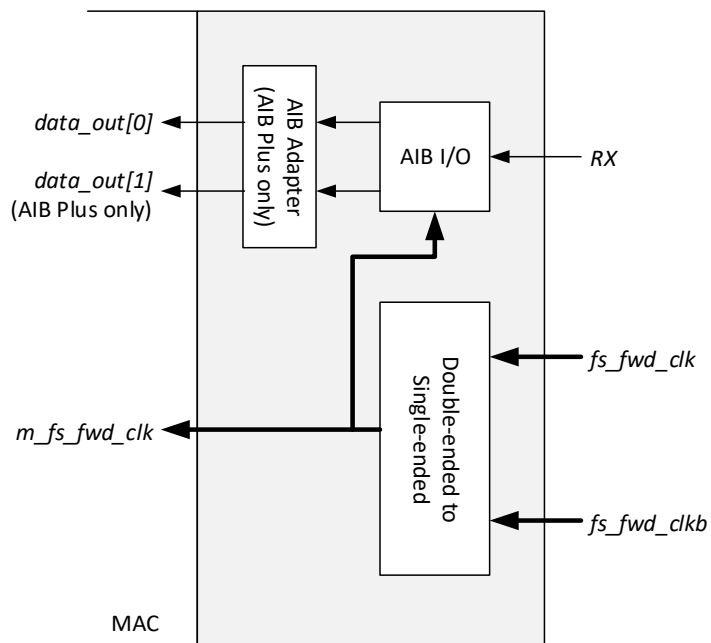
An AIB Base or AIB Plus channel shall forward its transmit clock to the far side. The clock signal shall be quasi-differential when moving from one chiplet to the other.

The near side shall receive the forwarded clock, *fs\_fwd\_clk*, from the far side. Once converted from quasi-differential to single-ended, the clock shall be made available to the MAC.





**Figure 20. Forwarded Clock – Transmit**



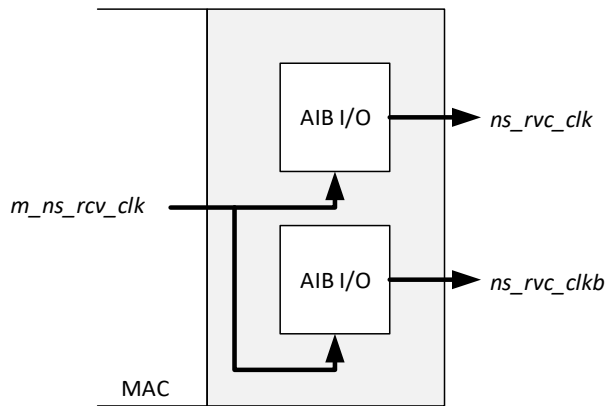
**Figure 21. Forwarded Clock – Receive**

#### **2.1.6.4 Receive-Domain Transmit Clock (AIB Plus only, Gen1 mode only)**

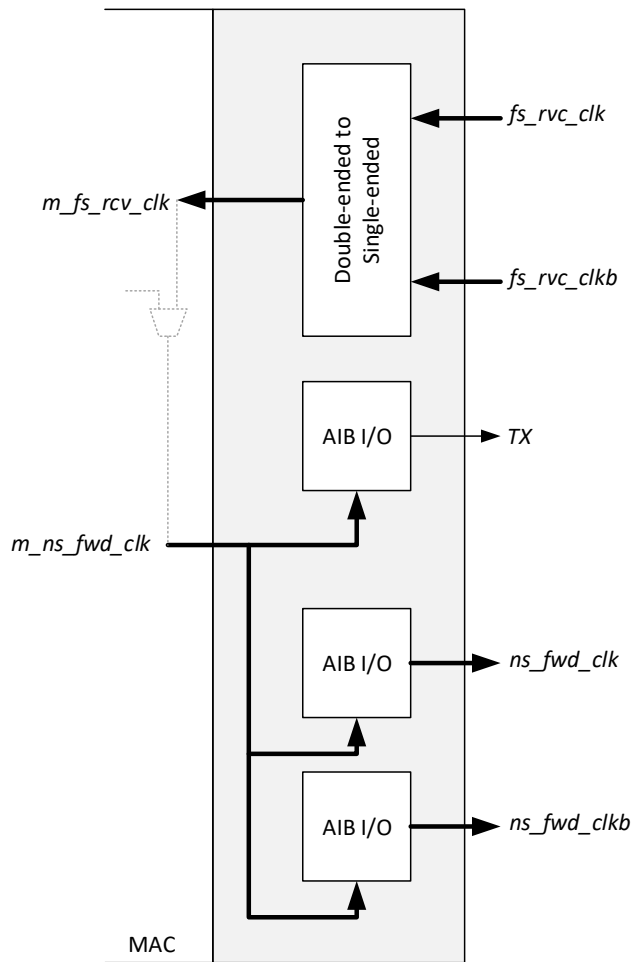
An AIB Plus channel may use a receive-domain clock from the far side as its transmit clock instead of the clock from the near-side MAC if it is desired to transmit data using the far-side receive-clock domain rather than the near-side clock domain. The receive-domain clock shall be quasi-differential when moving from one chiplet to the other.

The receive-domain clock, when received from the far side, shall be sent to the near-side MAC. The MAC may select the receive-domain clock as the transmit clock to be sent to the AIB layer. This MAC selection is suggested in Figure 23 in light gray for clarity only. The specific means of implementing the clock selection within the MAC is not specified in this document.

An AIB Plus interface shall generate either an active receive-domain clock or an inactive receive-domain clock. An active near-side receive-domain clock shall present the near-side receive-domain clock at the near-side receive-domain output clock bump. An inactive near-side receive-domain clock shall have a static value on the near-side receive-domain clock bump. If active, the data sheet should document the receive-domain clock frequency. If inactive, the data sheet should document that fact.



**Figure 22. Receive-Domain Clock – Transmit (AIB Plus only, Gen1 mode only)**



**Figure 23. Receive-Domain Clock – Receive (AIB Plus only, Gen1 mode only)**

#### 2.1.6.5 Clock Duty Cycle Requirements

The forwarded clock signal shall meet the following duty-cycle specification.

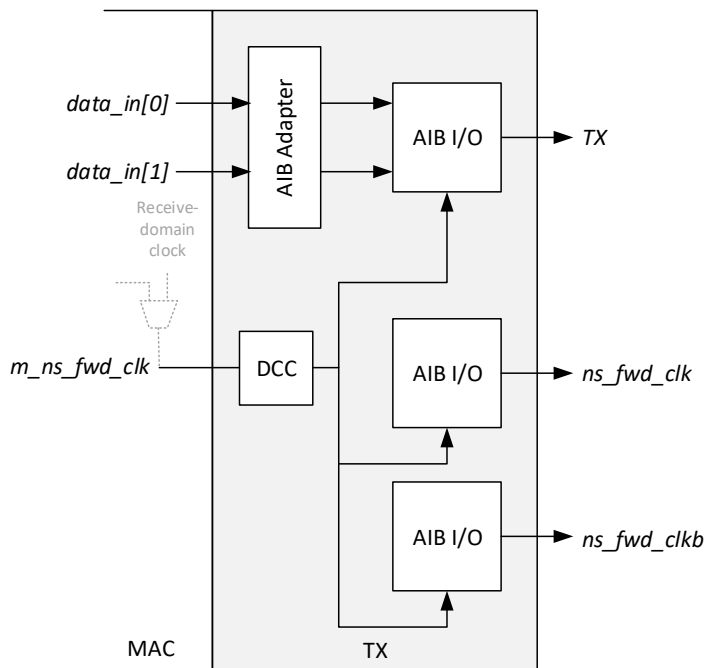
| Symbol            | Parameter                        |     | Near end   |
|-------------------|----------------------------------|-----|------------|
| $t_{\text{FDCD}}$ | Maximum forwarded-clock DCD      | SDR | $\pm 10\%$ |
|                   |                                  | DDR | $\pm 3\%$  |
| $t_{\text{RDCD}}$ | Maximum receive-domain clock DCD |     | $\pm 10\%$ |

**Table 10. Clock Duty-Cycle Requirements**

#### 2.1.6.6 Optional Forwarded-Clock Duty-Cycle Correction (AIB Plus only)

The transmit clock may pass through a duty-cycle correction (DCC) block prior to clocking the transmit register and prior to being forwarded if necessary for meeting the clock duty-cycle specification under all conditions of voltage and temperature (Section 2.1.6.5).

If the DCC is not present, calibration shall proceed as if it were (Section 3.2.3).



**Figure 24. Duty-Cycle Correction (AIB Plus only)**

#### 2.1.6.7 Optional Forwarded Clock Phase Selection (AIB Plus only)

The received forwarded clock may pass through a delay-locked loop (DLL) for phase

correction before clocking the capture register if necessary to ensure correct data sampling under all conditions of voltage and temperature.

If the DLL is not present, calibration shall proceed as if it were (Section 3.2.3).

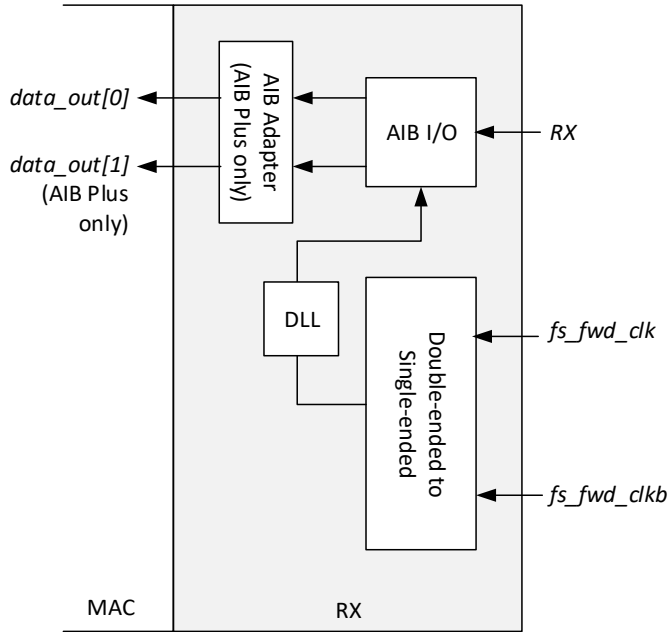


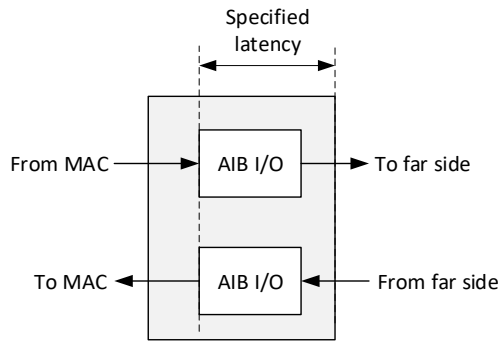
Figure 25. Delay-Locked Loop

2.1.7 Latency

The maximum latency for a signal from the input of a transmitting I/O block to the transmitted output, or from a received input to the output of the receiving I/O block, shall comply with Table 11 and Figure 26. For AIB Plus, the data sheet should indicate the total latency through the AIB Adapter and AIB I/O. The specified latency includes only the sequential delay and does not include analog delays through the transmitting output buffer, the receiving input buffer, or the interposer traces.

| Latency path | AIB Base, Gen1     | AIB Base, Gen2 and AIB Plus |
|--------------|--------------------|-----------------------------|
| Transmitting | 1 CLK cycle (1 UI) | 1.5 CLK cycle (3 UI)        |
| Receiving    | 1 CLK cycle (1 UI) | 1.5 CLK cycle (3 UI)        |

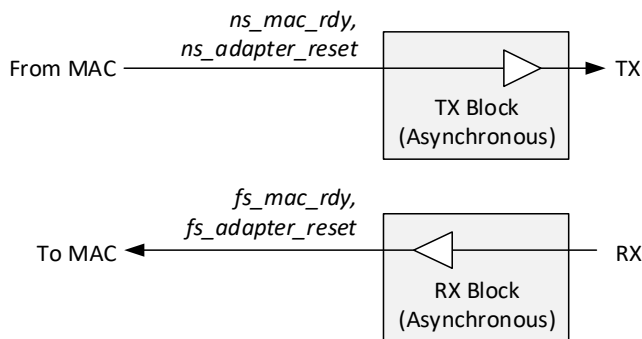
Table 11. Latency Specification



**Figure 26. Latency Measurement**

### 2.1.8 Asynchronous mode

I/O blocks that pass *xx\_mac\_rdy* (Gen1 mode only) and *xx\_adapter\_reset* signals shall operate in an asynchronous mode.



**Figure 27. Asynchronous I/O Mode.**

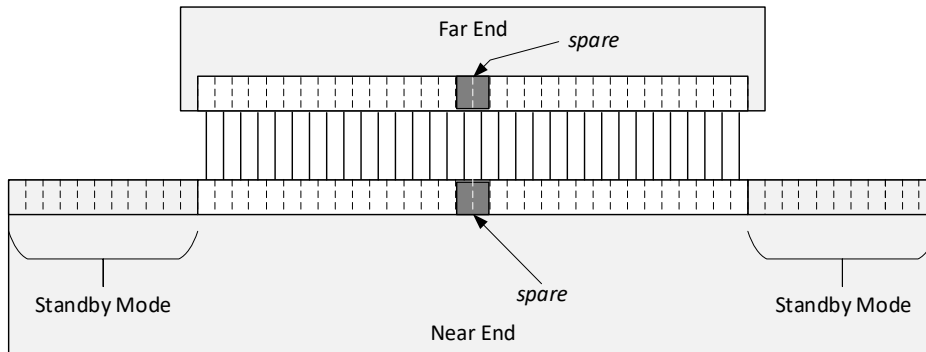
### 2.1.9 Mismatched Interfaces

The near side shall interoperate with a far side having a different number of I/Os than the near side provided:

- The two interfaces are of the same type (AIB Base or AIB Plus)
- The interfaces are aligned on the *spare* bumps.

If a near-side interface is connected to a far-side interface having fewer I/Os per channel than the near-side interface, then the unused I/Os in the near-side interface shall be placed into standby mode (Section 3.1.1). The near-side interface's MAC should be capable of handling the minimum number of I/Os from [Table 5Table-5](#) within the larger full number of bits in the near-side's AIB-to-MAC *data\_in* and *data\_out* signals. The near-side MAC should be configurable for operation with the minimum

number of I/Os or with the MAC's larger full number.



**Figure 28. Mismatched Interfaces, One Channel**

### 2.1.10 Unused Channels

Any unused channels shall have the same number of data signals as the used channels. The data signals in the unused channels shall be put into standby mode.

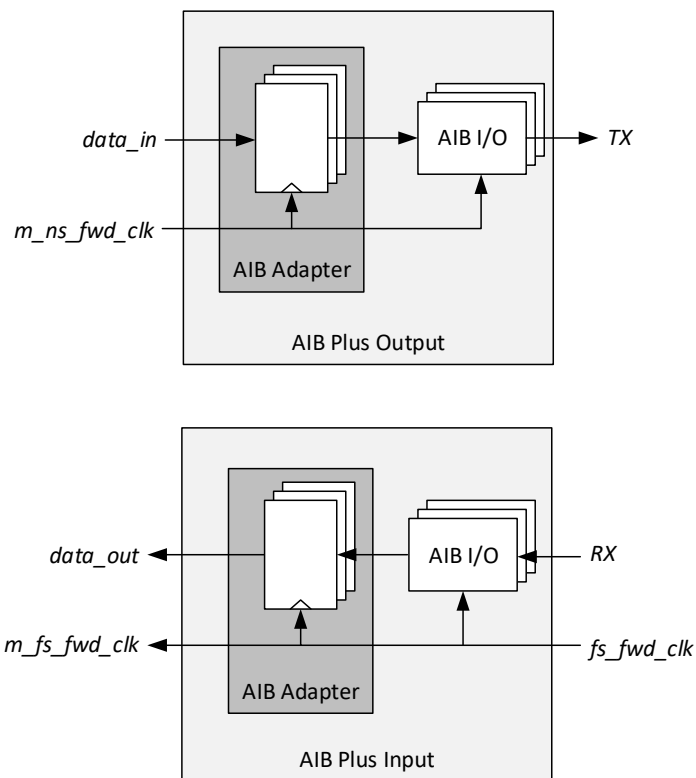
## 2.2 AIB Adapter (AIB Plus only)

The AIB Adapter shall provide data retiming, sideband control signals, and calibration control for DDR mode. There shall be one AIB Adapter per channel.

### 2.2.1 Data-Retiming Register

Within an AIB Plus interface, the AIB Adapter shall implement at least one retiming register in the data path on the transmit and receive sides. All data signals within a channel shall have the same retiming configuration. The data sheet should include total latency through all possible datapaths in the AIB Adapter.

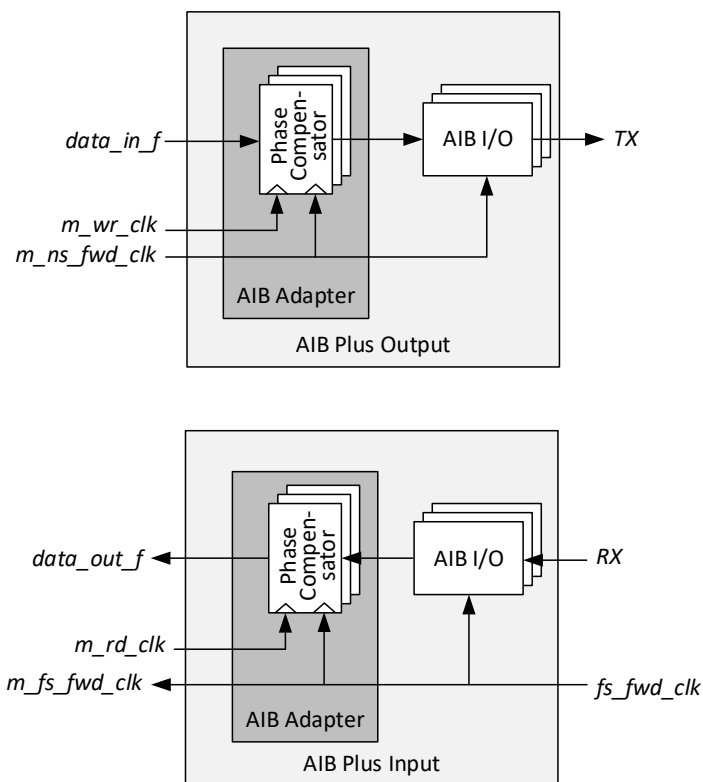




**Figure 29. Retiming Register**

### 2.2.2 Adapter Phase Compensator (AIB Plus and Gen2 only)

The AIB Adapter of a Gen2 AIB Plus interface shall implement a Phase Compensator (PC) for clock phase compensation between the AIB I/O and the MAC. The MAC supplied clocks shall be assumed by the AIB Adapter to be 0PPM different from the AIB I/O clocks, and the AIB Adapter shall compensate for the phase relation of the MAC-supplied clocks compared to the AIB I/O clocks. The phase compensation capability of the AIB adapter is implementation dependent. Figure 30 shows the adapter PC interfaces.



**Figure 30. Adapter Phase Compensators (PC)**

### 2.2.2.1 Phase Compensator Data Width and Rate at the PHY/MAC Interface

On the output path, the PC shall accept data from the MAC at full rate and same width as the AIB I/O's adapter interface, at half the rate and twice the width, and at one quarter the rate and four times the width. On the input path, the PC shall output data to the MAC at full rate and same width as the AIB I/O's adapter interface, at half the rate and twice the width, and at one quarter the rate and four times the width as the AIB I/O. The full rate, half rate or quarter rate mode shall be an AIB configuration setting. Note that the AIB I/O's adapter interface is twice as wide as RX or TX at the AIB I/O pins, since the AIB I/O performs SDR/DDR conversion.

The AIB Adapter shall assume that the *m\_wr\_clk* and *m\_rd\_clk* inputs correspond in frequency to the PC's full rate, half rate or quarter rate setting. For example, a 4Gbps AIB interface has *m\_fs\_fwd\_clk* at 2GHz. A quarter rate setting for the AIB Adapter PC has *m\_rd\_clk* at 500MHz, but still at 0PPM difference from *m\_fs\_fwd\_clk*.

Table 12 is an example of the width of *data\_out\_f* and *data\_in\_f* compared to RX and TX, with the PC PHY/MAC interface rate varied from full rate, to half rate and to quarter

rate.

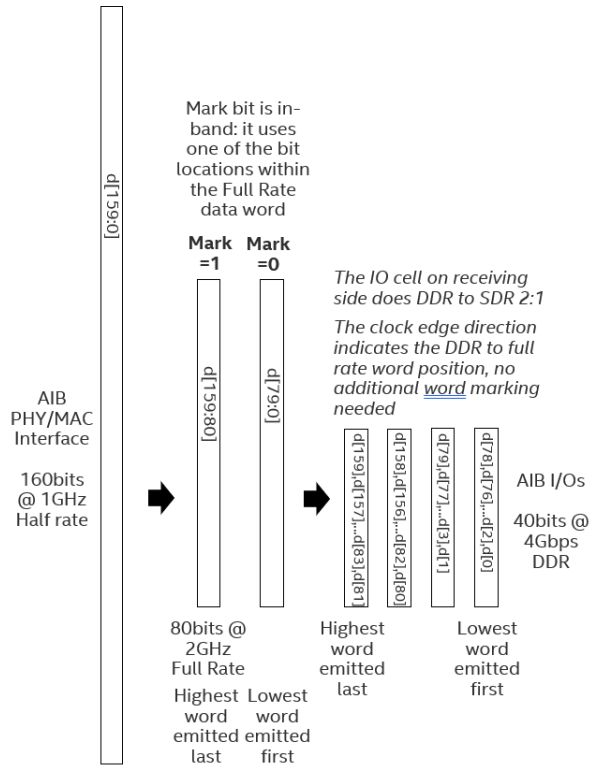
| AIB <i>RX</i> or <i>TX</i> width | Adapter PC Rate Configuration | Adapter PC PHY/MAC interface width of <i>data_out_f</i> or <i>data_in_f</i> |
|----------------------------------|-------------------------------|---|
| 40                               | Full Rate                     | 80  |
| 40                               | Half Rate                     | 160   |
| 40                               | Quarter Rate                  | 320   |

**Table 12. PC PHY/MAC Interface Width Examples**

#### **2.2.2.2 AIB Adapter Word Marking and Word Assembly**

Assembling a half rate or quarter rate word sent over an AIB channel requires information about the position of received full rate words within the original half rate or full rate word. A signal from the transmitting side to the receiving side is commonly used to indicate the full rate word's position. When word assembly by the AIB is enabled, the adapter uses a bit within each full rate data word to determine the full rate word's position.

The AIB Adapter shall have the capability to mark outgoing data to indicate which full rate data word was at the high position of a larger data word input from the MAC. Word marking is used in Adapter PC half rate and quarter rate configurations. A Mark bit is allocated in-band and used every full rate word. Figure 31 is an example word marking using half rate PC mode and 40Tx bits per channel. Note that the Mark bit is set to 1 only on the upper full rate word of the larger word received from the MAC. Other full rate words have their Mark bit set to 0.



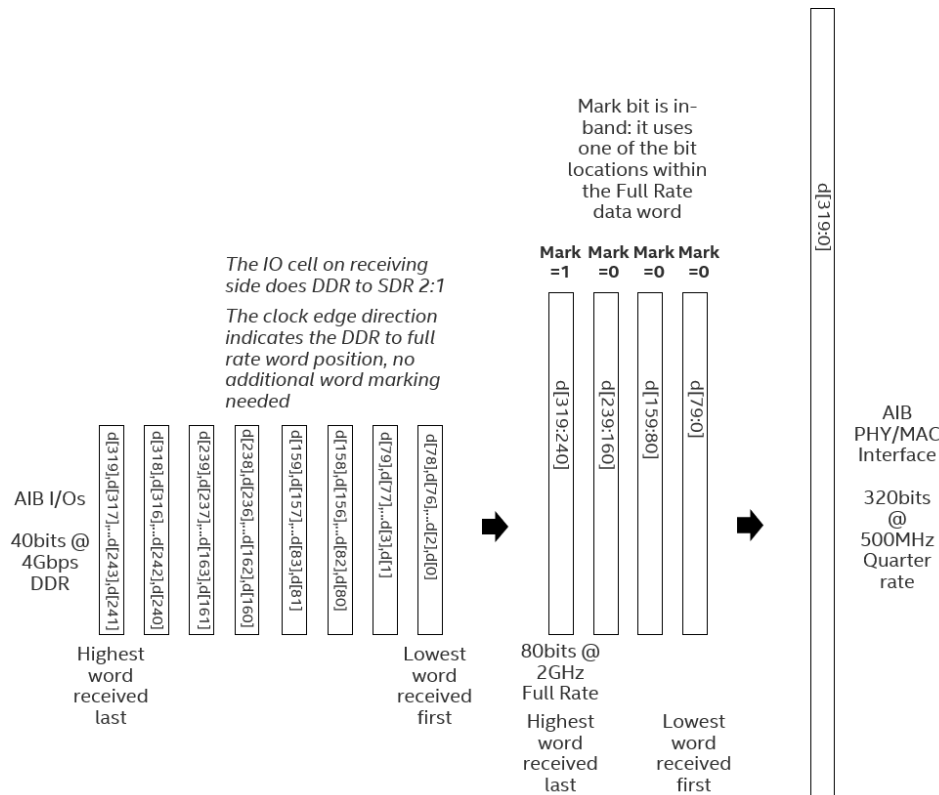
**Figure 31. Word Marking Example,  
Half Rate PC Mode, 40 Tx Bits per Channel**

The Mark bit position within the full rate word shall be configurable. The recommended bit position in the full rate word is the 2nd MSB, which avoids conflict with DBI (section 2.2.4). In the example of Figure 31, the 2<sup>nd</sup> MSB position puts the Mark bit at d[78] and d[158].

Word marking shall be programmable to one of two states:

- No marking (user data passes through at the Mark bit location)
- Mark is 0 or 1 according to the full rate word's position in the word at the MAC/PHY interface, with Mark=1 only in the highest full rate word

On the receiving side, the AIB Adapter shall reassemble the data from the AIB RX into a larger word if the adapter is configured into half rate or quarter rate mode. The adapter shall use the Mark bit to determine which is the high full rate word and assemble the words into their corresponding positions. The Mark bit position within the full rate word shall be configurable. Figure 32 is an example of quarter rate word assembly.



**Figure 32. Word Assembly Example,  
Quarter Rate PC Mode, 40 Rx Bits per Channel**

Once the receiving AIB Adapter finds the Mark bit and is assembling half rate or quarter rate words, the adapter is said to be aligned. Once aligned, the adapter shall set the PHY/MAC signal *m\_rx\_align\_done* HI and continue assembling half rate or quarter rate words from full rate words in the sequence received. Once aligned, the adapter does not need to locate the Mark bit again. This permits the transmit side to stop sending Mark bits and instead use that bit position for data.

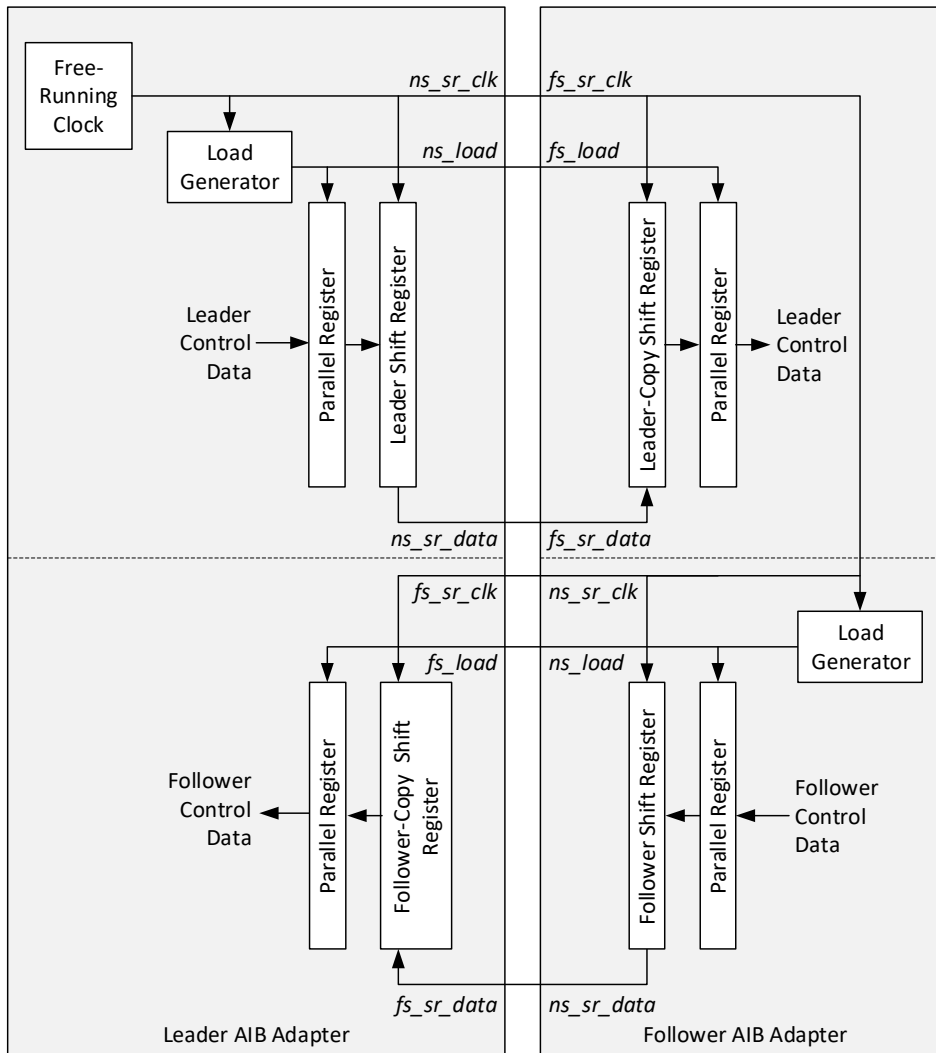
Once aligned, if the adapter sees an unexpected 0 or 1 in the Mark bit position, the adapter shall set *m\_rx\_align\_done* LO. The adapter shall still continue assembling half rate or quarter rate words in the sequence received. An interface reset (*power\_on\_reset*, *i\_conf\_done* or *i\_m\_power\_on\_reset*) or adapter reset (*ns\_adapter\_rstn* or *fs\_adapter\_rstn*) is required for the adapter to restart its alignment process.

Note that if a reset is applied, word marking must be enabled for the AIB Adapter to complete a new alignment process.

### 2.2.3 Sideband Control Signals

An AIB interface shall provide control signals for calibration and communication of control and status information between the near-side and far-side chiplets. These signals shall be time-domain multiplexed onto a single data line. Internal signals shall be first loaded into a parallel register before being shifted out serially, starting with the most-significant bit (MSB). Received serial signals shall be loaded into a parallel register upon assertion of the received *load* signal. A *load* signal (Section 2.2.3.3) shall coordinate the timing of the loading from and to parallel registers. The *load* signal shall be generated on the sending side and shall be forwarded from the sending chiplet to the receiving chiplet.

The sideband control shift register shall be clocked by a free-running clock (Section 2.2.3.2) such that the shift registers shall constantly send data. It shall not be possible to stop or restart the sending of sideband control signals.



**Figure 33. Sideband Control Shift Registers**

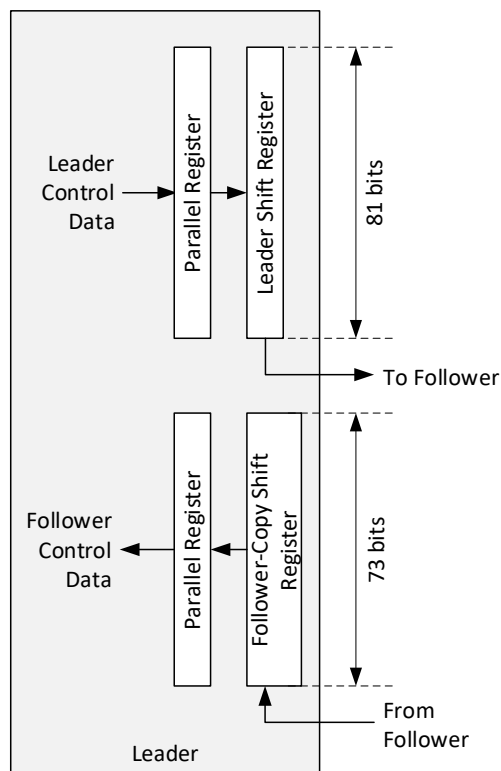
### 2.2.3.1 Shift-Register Configurations

The definition of the shift register and the position of signals shall depend upon whether the interface is a leader, follower, or dual-mode.

#### 2.2.3.1.1 Leader Interface

A leader interface shall include two shift registers: one for transmitting leader sideband

control signals to the follower (the leader shift register), and one for receiving sideband control signals from the follower (the follower-copy shift register). The leader shift register shall contain 81 bits. The follower-copy shift register shall contain 73 bits. All bits shall be implemented regardless of whether optional signals are implemented. Any signals not implemented shall permanently maintain their default values as defined in Table 61.

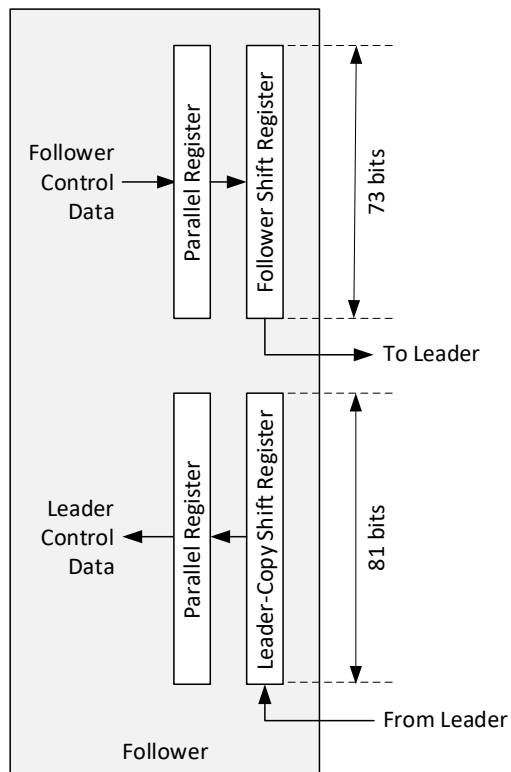


**Figure 34. Leader Sideband Control Shift Register**

#### 2.2.3.1.2 Follower Interface

A follower interface shall include two shift registers: one for transmitting follower sideband control signals to the leader (the follower register), and one for receiving sideband control signals from the leader (the leader-copy shift register). The follower shift register shall contain 73 bits. The leader-copy shift register shall contain 81 bits. All bits shall be implemented regardless of whether optional signals are implemented. Any signals not implemented shall permanently maintain their default values as defined in Table 62





**Figure 35. Follower Sideband Control Shift Register**

#### 2.2.3.1.3 Dual-Mode Interface

When configured as a leader, a dual-mode interface shall implement an output shift register configured as a leader and an input shift register configured as a follower. When configured as a follower, a dual-mode interface shall implement an output shift register configured as a follower and an input shift register configured as a leader.

The *dual\_mode\_select* signal (Section 1.3.3.2) shall select the shift-register configuration appropriate to the selected mode.

#### 2.2.3.2 Free-Running Clock

The shift registers shall be clocked by a free-running clock. The free-running clock shall be generated on the leader side and forwarded to the follower side.

The free-running clock shall be independent of the data clocks. It shall run continuously during operation. It shall have a frequency within the range specified in Table 13.

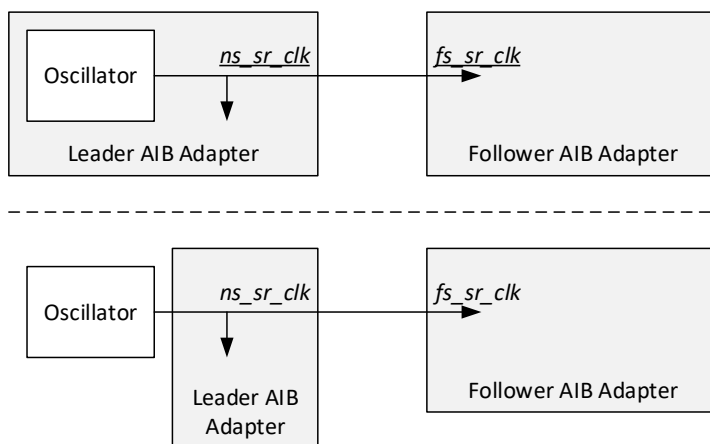
| Parameter   | Min     | Max   |
|---|---------|-------|
| Free-Running Clock Frequency (default)  | 600 MHz | 1 GHz |
| Free-Running Clock Frequency if user-defined signals (Section 2.2.3.5.2) are not used | 50 Mhz  | 1 Ghz |

**Table 13. Free-Running Clock Frequency**

When user defined signals are used, the higher minimum free-running clock frequency ensures that the far side will be able to see changes in the near side user signals. Since the shift registers are approximately 100 bits long, the user defined signals can change up to approximately a frequency that is 1/200<sup>th</sup> of the free-running clock frequency and the change can be detected on the far side.

#### 2.2.3.2.1 Free-Running Clock Generation

The leader interface chiplet shall be responsible for generating the free-running clock and distributing it to the follower interface via the *sr\_clk* signal (*ns\_sr\_clk* when sent from the near side; *fs\_sr\_clk* when received from the far side). The follower interface (especially dual-mode) chiplet can either use forwarded free running clock from leader interface chiplet or use its existing free running clock to support output operation of its sideband control signals. If an external oscillator is used to generate the free-running clock, then output of that oscillator shall be run through the leader interface before it is distributed to the follower interface via *sr\_clk* in order to ensure the correct phase relationships between the free-running clock and the *load* signal (Section 2.2.3.3).



**Figure 36. Free-running Clock Generation Options**

#### 2.2.3.3 Load Signal

An *sr\_load* signal (*ns\_sr\_load* when sent from the near side; *fs\_sr\_load* when received

from the far side) shall provide synchronous data transfer between parallel and serial shift registers (Figure 35). It shall be created from the free-running clock by a sideband-control state machine and transmitted as an SDR signal (Figure 16). The period of the *load* signal shall be a number of free-running clock cycles that is one more than the number of bits in the shift register (i.e., 1/74 or 1/82).

The *sr\_load* signal shall remain LO until asserted HI; it shall remain asserted HI for one clock cycle before being de-asserted LO.

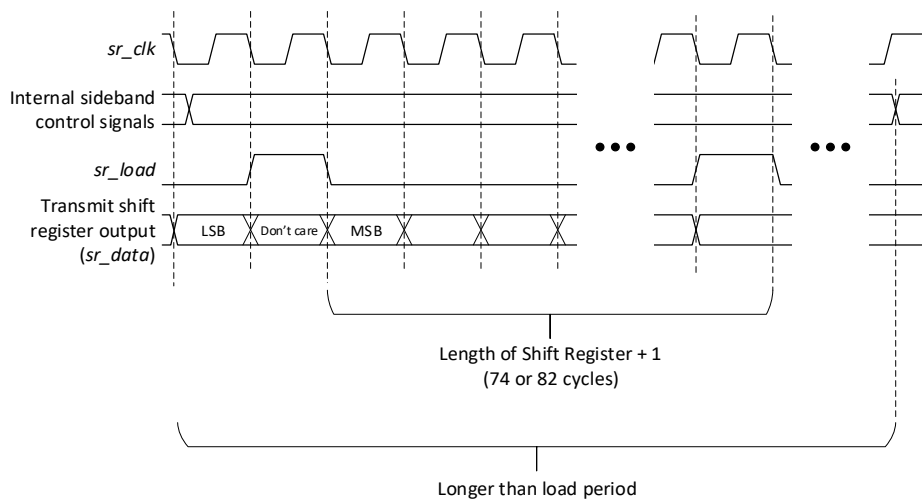
When transmitting sideband control signals, the control signal data shall be clocked into the parallel register when the *load* signal is asserted. When receiving sideband control signals, the received serial data shall be transferred into the parallel register when the *load* signal is asserted.

There shall be no valid control-bit data when the *load* signal is asserted. The MSB shall be shifted out on the falling edge of the *load* signal.

#### 2.2.3.4 Control Signal Timing

When loading a new sideband control signal value into the parallel register for shifting out, the value presented to the parallel load register shall remain valid for at least the maximum time between *load* assertions.

The first bit of the shift register to be shifted out when transmitting shall be the MSB (bit 72 or 80); the last bit to be shifted in when receiving shall be the LSB (bit 0).



**Figure 37. Sideband-Control Shift-Register Timing**

#### 2.2.3.5 Control Signals

Sideband control signals fall into one of the following categories:

- Calibration handshake
- Selective reset
- User-defined
- Reserved

The bits for any unused signals shall be maintained with default values for correct shift-register length.

The sideband control signals are summarized in Table 14 and are detailed in Table 62.

#### **2.2.3.5.1 Calibration Bits**

Calibration bits shall be generated by internal state machines as described in Section 3.2.3.

#### **2.2.3.5.2 User-Defined Bits**

User-defined bits are available for application use. Since both sides need to understand the function of user-defined bits, using these bits may limit chiplet interoperability. If implemented in an application, user-defined bits should be described in the chiplet data sheet.

#### **2.2.3.5.3 Shift-Register Signals**

Table 14 defines the sideband control signals for leader and follower chiplets. The table is organized by signal type; in-order signal tables with default values are provided in Section 8.1.3. *ms* prefixes refer to signals originating on the leader side; *sl* prefixes refer to signals originating on the follower side.

| Signal name  | Signal function                        | Bits                       | Signal origin (far-side or MAC) | Bit number                           |                            |
|--|--|----------------------------|---------------------------------|--------------------------------------|----------------------------|
|  |  |                            |                                 | Leader                               | Follower                   |
| Calibration (Section 3.2.3)                                    |  |                            |                                 |                                      |                            |
| <i>ms_osc_transfer_en</i><br><i>sl_osc_transfer_en</i>         | Oscillator calibration complete        | 1                          | FS                              | 80                                   | 72                         |
| <i>ms_tx_dcc_cal_done</i><br><i>sl_tx_dcc_cal_done</i>         | TX DCC calibration complete            | 1                          | FS                              | 68                                   | 31                         |
| <i>ms_rx_transfer_en</i><br><i>sl_rx_transfer_en</i>           | RX calibration complete                | 1                          | FS                              | 75                                   | 70                         |
| <i>ms_rx_dcc_dll_lock_req</i><br><i>sl_rx_dcc_dll_lock_req</i> | Start RX calibration                   | 1                          | MAC                             | NA                                   | 69                         |
| <i>ms_rx_dll_lock</i><br><i>sl_rx_dll_lock</i>                 | RX DLL locked                          | 1                          | FS                              | 74                                   | 68                         |
| <i>ms_tx_transfer_en</i><br><i>sl_tx_transfer_en</i>           | TX calibration complete                | 1                          | FS                              | 78                                   | 64                         |
| <i>ms_tx_dcc_dll_lock_req</i><br><i>sl_tx_dcc_dll_lock_req</i> | Start TX calibration                   | 1                          | MAC                             | NA                                   | 63                         |
| User-defined   |  |                            |                                 |                                      |                            |
| <i>external_cntl[]</i>   | Defined by protocol and/or application | Follower: 56<br>Leader: 63 | MAC or FS                       | 0-4<br>8-65                          | 32-57<br>28-30<br>0-26     |
| Other  |  |                            |                                 |                                      |                            |
| Reserved   |  |                            | NA                              | 79<br>76-77<br>69-73<br>66-67<br>5-7 | 71<br>65-67<br>58-62<br>27 |

**Table 14. Sideband Control Signals**

## 2.2.4 Data Bus Inversion

Data Bus Inversion (DBI) is intended to reduce the power delivery network load of an AIB PHY by limiting the number of AIB data bits that can switch between immediate data transfer cycles. In Gen2 mode (DDR data transfers), AIB shall support DBI-AC in groups of 20 RX and 20 TX data wires. Channels may have multiple DBI data groups, for example a channel with 40 RX and 40 TX has a total of 4 DBI data groups. DBI shall be configurable on or off. When DBI is configured on, the AIB adapter shall process both RX and TX data for DBI. With DBI off, the data inside RX and TX are unaffected.

With DBI on, DBI bits replace certain data bits. The example below shows a 40 bit TX channel. RX wires have DBI at the same bit locations. Channels with more IOs continue with same 19 data bits + 1 DBI bit increment.

DBI Off: TX[39:0] = data[39:0]

DBI On: TX[39:0] = {DBI[1], data[38:20], DBI[0], data[18:0]}

The MAC data\_in and data\_out (section 1.3.5) contains even and odd bits that are multiplexed at the DDR transmitter and demultiplexed at DDR receiver (sections 2.1.1, 2.1.2 and 2.1.3). The “data” in the above example is selected from the MAC data\_in, anticipating DDR transmission. With DBI On, the data bits at the DBI locations are not used.

### 2.2.4.1 TX DBI with DBI Enabled

Within a group of 19 data signals, the TX DBI logic calculates the DBI bit based on the number of data signals changing from their previous state on the AIB data bus.

The DBI logic below uses “+” to indicate arithmetic addition, “^” to indicate exclusive OR, and “?” as a ternary IF. “Current” refers to the new data word being prepared for sending on TX. “Prev” refers to the data immediately preceding the current data, that is the data issued onto the AIB bus before the current data.

$$\begin{aligned} \text{DBI}_{\text{current}} = & ((\text{data}[18]_{\text{current}} \wedge \text{data}[18]_{\text{prev}}) \\ & + (\text{data}[17]_{\text{current}} \wedge \text{data}[17]_{\text{prev}}) \\ & \dots \\ & + (\text{data}[1]_{\text{current}} \wedge \text{data}[1]_{\text{prev}}) \\ & + (\text{data}[0]_{\text{current}} \wedge \text{data}[0]_{\text{prev}})) > 9 ? 1 : 0; \end{aligned}$$

Within a group of 19 data signals, if the DBI bit=1 then the TX DBI logic inverts the data signals. Each calculated DBI bit replaces one data bit in TX as described previously.

### 2.2.4.2 RX DBI with DBI Enabled

Within a group of 20 RX signals, the RX DBI logic extracts the DBI bit. If DBI=1 then the RX DBI logic inverts the other RX bits.

## 3 Reset and Initialization

---

### 3.1 Data-Transfer Ready

A data-transfer ready signal shall be made available for control by the MAC layer. The data-transfer ready signal may be de-asserted due to application-driven changes, including but not limited to:

- An intentional change in clock frequency
- Receipt of bad data

De-asserting the data-transfer ready signal may also be necessary due to conditions within the AIB interface, which may include but are not limited to:

- Completion of configuration during power-up
- Initiation of reset by the far side
- Loss of DLL lock

Internal AIB conditions indicating the need for de-assertion of the data-transfer ready signal shall be sent to the MAC so that the MAC can de-assert the data-transfer ready signal.

For AIB Plus interfaces, once data-transfer ready has been re-asserted after having been de-asserted, the AIB Adapter shall be re-calibrated (Section 3.2.3). The reverse is not true: calibration may be initiated without de-asserting data-transfer ready first.

#### 3.1.1 Standby Mode

A signal shall be placed into standby mode by one of the following means:

- Driving the signal LO
- Putting the signal into tristate and enabling the weak pull-down.

During initialization, data outputs shall be placed into standby mode.

#### 3.1.2 Data-Transfer Ready Signals

Each channel shall have an *ns\_mac\_rdy* signal that is controlled by the near-side MAC. When the *ns\_mac\_rdy* signal is asserted HI by the MAC, it shall indicate that the near side is ready for calibration and data transfer. De-assertion of *ns\_mac\_rdy* shall affect only its own channel; other channels may continue transmitting data.

The *ns\_mac\_rdy* signal shall be forwarded to the far side, where it shall be received on the *fs\_mac\_rdy* input in order to inform the far-side MAC that the near-side MAC is or is not ready for calibration.

#### 3.1.3 The Effects of De-asserting Data-Transfer Ready

While the *ns\_mac\_rdy* signal is de-asserted:

- Data transmission shall halt

- Data outputs shall be placed into standby mode (Section 3.1.1)
- The clock output *ns\_fwd\_clk* shall go into standby mode.
- The reset signal *ns\_mac\_rdy* shall be sent to the far side of the interface in order to communicate that data transmission has halted and to allow for the far side to be reset.

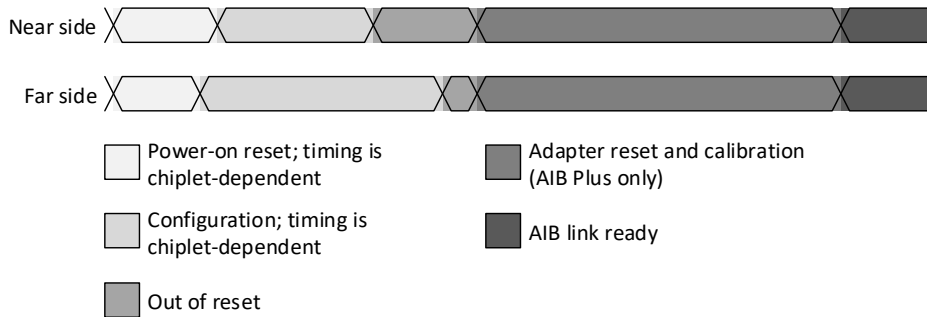
The contents of retiming registers (Section 2.2.1) shall be undefined following de-assertion of data-transfer ready.

De-assertion of the data-transfer ready signal shall not affect the free-running clock signals or the sideband-control signals.

## 3.2 Initialization

Initialization will consist of two or three steps in sequence:

- Power-on reset synchronization
- Configuration
- Calibration (AIB Plus only)



**Figure 38. AIB Initialization**

If there are multiple AIB Plus interfaces on a single chiplet, they shall all come out of configuration at the same time, but they may complete adapter reset and calibration at different times depending on implementation.

### 3.2.1 Power-on Reset Synchronization

Power-on reset, being the first step in initialization, shall not require any features enabled by configuration (in the manner that the SDR/DDR option is configured, for example), since configuration will not occur until after power-on reset.

#### 3.2.1.1 Power-on Reset Signals

Two signals shall participate in power-on reset: *power\_on\_reset* and *device\_detect*. Their function shall depend on whether the interface is a leader, a follower, or dual-mode (Section 1.3.3).

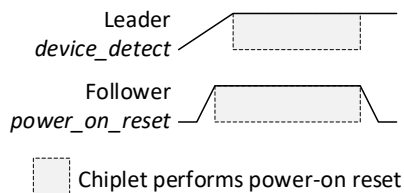


- For leader interfaces:
  - *power\_on\_reset* shall be implemented as an input.
  - *device\_detect* shall be implemented as an output.
- For follower interfaces:
  - *power\_on\_reset* shall be implemented as an output.
  - *device\_detect* shall be implemented as an input.
- For dual-mode interfaces, the *dual\_mode\_select* (Section 1.3.3.2) signal shall select the function of the *power\_on\_reset* and *device\_detect* signals as input/output (leader) or output/input (follower).

### 3.2.1.2 Power-on Reset Sequence

During power-on reset, all input and output signals shall be placed into standby mode (Section 3.1.1). The power-on reset sequence shall proceed as follows:

1. The leader interface shall assert its *device\_detect* signal HI to indicate its presence to follower interfaces on different chiplets. If no *device\_detect* signal is detected by the follower, then the follower may act both to ensure that it and its chiplet are in a safe state and to alert the MAC.
2. Each chiplet shall implement its own power-on reset routine. At the beginning of the routine, follower interfaces shall assert their *power\_on\_reset* signals HI.
3. When a chiplet completes its power-on reset sequence:
  - a. Leader interfaces shall begin the configuration stage.
  - b. Follower interfaces shall de-assert their *power\_on\_reset* signals LO and begin the configuration stage.



**Figure 39. Power-on reset synchronization**

### 3.2.1.3 Unused Interfaces

In order to ensure correct operation for chiplets with unused leader interfaces, the *power\_on\_reset* inputs for those unused interfaces shall have weak pull-ups.

In order to ensure correct operation for chiplets with unused follower interfaces, the *device\_detect* inputs for those unused interfaces shall have weak pull-downs.

### 3.2.1.4 Test Provision

In order to test the power-on reset sequence at the wafer level, two signals shall be provided for use by automated test equipment to override the *power\_on\_reset* and *device\_detect* signals when there is no leader/follower pair available. *por\_ovrd*



### 3.2.2.1 Output State During Configuration

All outputs, including data outputs, the free-running clock output, sideband-control output, reset outputs, and adapter reset outputs shall be in standby mode (Section 3.1.1) during configuration. The free-running clock output, sideband-control output, reset outputs, and adapter reset outputs must come out of standby mode upon completion of configuration.

### 3.2.2.2 Chiplet Configuration

Configuration of any non-AIB aspects of the chiplet is outside the scope of this specification.

### 3.2.2.3 AIB Interface Configuration

#### 3.2.2.3.1 Power-Up Configuration

All intended AIB features shall be configured at power-up.

#### 3.2.2.3.2 JTAG Configuration

The chiplet data sheet should document the configuration requirements that allow for successfully implementation of JTAG EXTEST and INTEST operations.

#### 3.2.2.3.3 Free-Running Clock Oscillator Stability (AIB Plus only)

Within a leader interface, the oscillator used for the free-running clock shall be stable before configuration is complete.

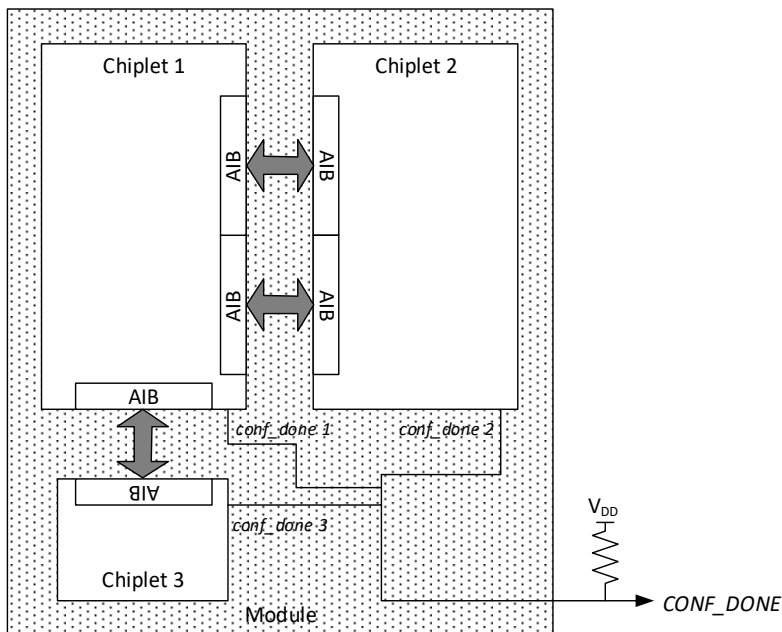
#### 3.2.2.3.4 Sideband Control Shift Register Readiness (AIB Plus only)

The sideband control shift register shall be operational once configuration is complete.

#### 3.2.2.3.5 Configuration Completion Signals

Each chiplet shall have a *conf\_done* signal. *conf\_done* shall be an open-drain output. It shall be asserted LO when configuring, and it shall be released when configuration of all interfaces on the chiplet is complete, the analog circuits are stable, and the free-running clock is stable. *conf\_done* shall indicate only that AIB configuration is complete. No other configuration completion (MAC, FPGA, etc.) shall be included in the generation of the *conf\_done* signal.

All *conf\_done* signals from all chiplets of a module should be connected in a wired-AND configuration to generate a module-level *CONF\_DONE* signal that shall be HI when all chiplets on the module have completed AIB configuration. The pull-up resistor used to implement the wired-AND function may reside on the module containing the chiplets with AIB interfaces, or it may reside off the module. The *CONF\_DONE* signal should be provided as an output of the module regardless of the resistor placement.



**Figure 41. Configuration completion signals.**

Data outputs shall remain in standby mode (Section 3.1.1) until *CONF\_DONE* is asserted, including in the case where *CONF\_DONE* is pulled low some time after being asserted high.

The resistance and  $V_{DD}$  values should comply with

| Parameter          | Value        |
|--------------------|--------------|
| Pull-up resistance | 1 k $\Omega$ |
| Pull-up $V_{DD}$   | 0.9 V        |

**Table 15. Wired-AND Pull-Up Guidelines**

### 3.2.3 Calibration (AIB Plus only)

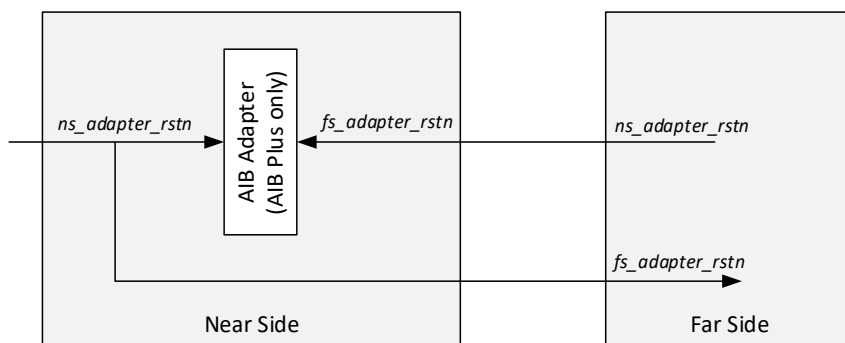
The calibration sequence shall proceed as follows:

- Adapter reset
- Free-running-clock synchronization
- Data path calibration

The adapter reset phase and free-running-clock synchronization phase may run concurrently.

### 3.2.3.1.1 Adapter Reset Signal

An AIB interface shall have an *ns\_adapter\_rstn* signal that is asserted by the MAC. It shall be forwarded to the far side of the interface, driving the far side's *fs\_adapter\_rstn* input. Likewise, the interface shall have an *fs\_adapter\_rstn* input that accepts the *ns\_adapter\_rstn* signal from the far side.

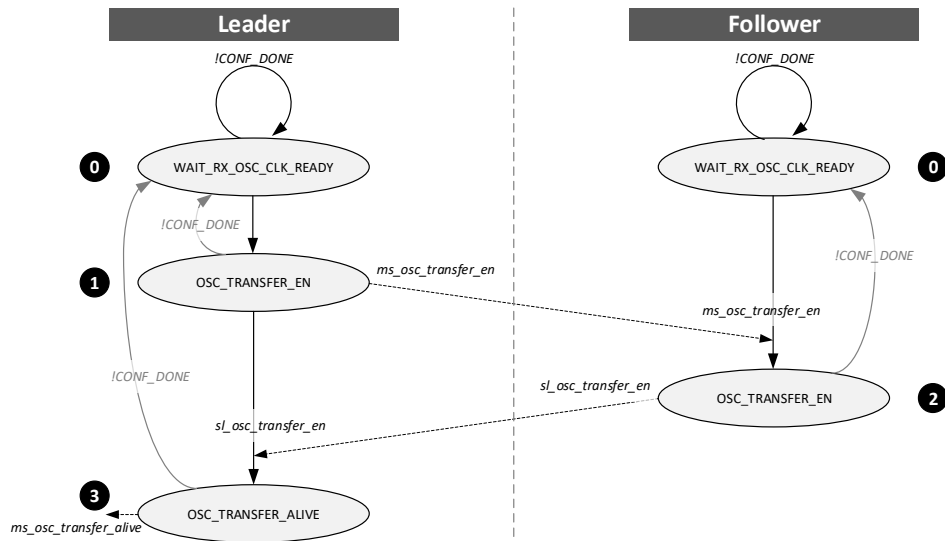


**Figure 42. Adapter Reset**

When either the near-side or far-side adapter reset signal is asserted LO, the adapter shall reset the calibration state machines. If adapter reset follows de-assertion of data-transfer ready, *ns\_mac\_rdy* must be asserted HI before *ns\_adapter\_rstn* is asserted HI.

### 3.2.3.2 Free-Running-Clock Synchronization

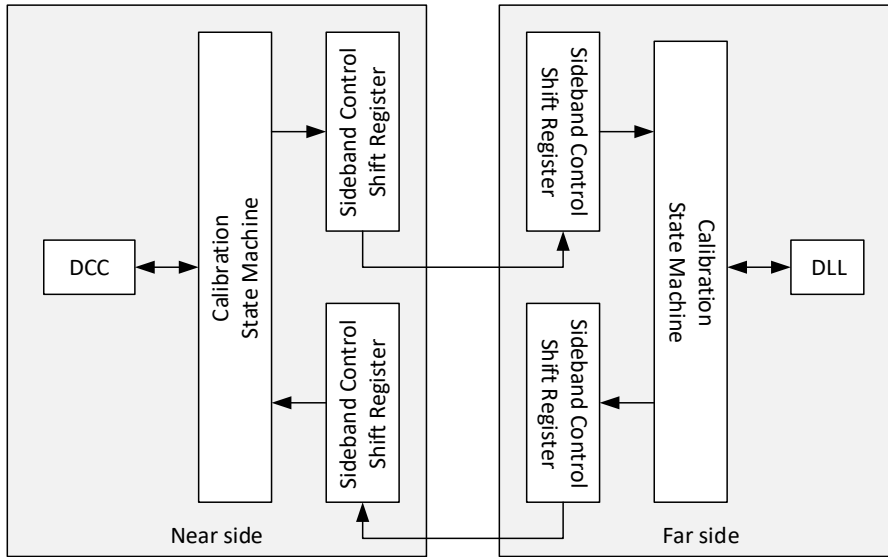
The free-running clocks are synchronized across the interface according to Figure 43. The numbers in black indicate the sequence of steps.



**Figure 43. Free-running Clock Calibration State Machine**

### 3.2.3.3 Data-Path Calibration

Data-path calibration shall be implemented via state machines on the leader and follower sides of the interface that intercommunicate via the sideband control signals.



**Figure 44. Data-Path Calibration Architecture**

Following the de-assertion of the adapter-reset signal(s), a calibration request shall be made by asserting a calibration request signal. Separate calibration sequences shall be used for a leader transmitting to a follower (Section 3.2.3.3.6) and a follower transmitting to a leader (Section 3.2.3.3.7). The two sequences may occur in any order or concurrently.

#### 3.2.3.3.1 Adapter Reset

Data-path calibration shall be initiated when the MAC layer asserts the *ns\_adapter\_rstn* signal LO or the far side asserts the *fs\_adapter\_rstn* signal LO. If the data-transfer ready signal was de-asserted prior to the start of calibration, then the *ns\_mac\_rdy* signal must be asserted HI prior to asserting the adapter-reset signals HI

The MAC must de-assert the adapter-reset signal prior to requesting calibration start.

#### 3.2.3.3.2 Calibration Request

Calibration can be requested by either the leader or the follower using the *ms\_xx\_dcc\_dll\_lock\_req* signal or the *sl\_xx\_dcc\_dll\_lock\_req* signal, respectively, where “xx” is *tx* or *rx* according to the direction of dataflow.

| Calibration initiator | Dataflow direction | Initiation signal             |
|-----------------------|--------------------|-------------------------------|
| Leader                | Leader to follower | <i>ms_tx_dcc_dll_lock_req</i> |
| Follower              | Leader to follower | <i>sl_rx_dcc_dll_lock_req</i> |
| Leader                | Follower to leader | <i>ms_rx_dcc_dll_lock_req</i> |
| Follower              | Follower to leader | <i>sl_tx_dcc_dll_lock_req</i> |

**Table 16. Calibration Initiation Signals**

Calibration for a dataflow direction shall commence when both leader and follower side have asserted their calibration request signals for that dataflow direction. Calibration request signals shall remain asserted until a new calibration is requested.

#### 3.2.3.3.3 DCC Calibration

Upon receipt of an *xx\_dcc\_dll\_lock\_req* signal, the DCC shall be calibrated. The means of calibration is not specified and is left to the designer. If the optional DCC is not present, then the state machines in Section 3.2.3.3 shall remain the same, with the DCC calibration state serving only to provide a signal indicating DCC calibration completion.

#### 3.2.3.3.4 DLL Calibration

Following DCC calibration, the receiving DLL shall be calibrated. The means of calibrating the DLL is not specified and is left to the designer.

If the optional DLL is not present, then the state machine in Section 3.2.3.3 shall remain the same, with the DLL lock state serving only to provide a signal indicating DLL lock completion.

#### 3.2.3.3.5 Calibration Completion

Calibration completion shall be indicated by the following signals. Full completion shall be indicated when all four signals are asserted HI. All four signals, once asserted, shall remain asserted until a new calibration sequence is requested.

| Calibration Completion Signal | Meaning  |
|-------------------------------|--|
| <i>ms_tx_transfer_en</i>      | Leader transmit block has completed calibration. |
| <i>ms_rx_transfer_en</i>      | Leader receive block has completed calibration   |
| <i>sl_tx_transfer_en</i>      | Follower transmit block has                      |

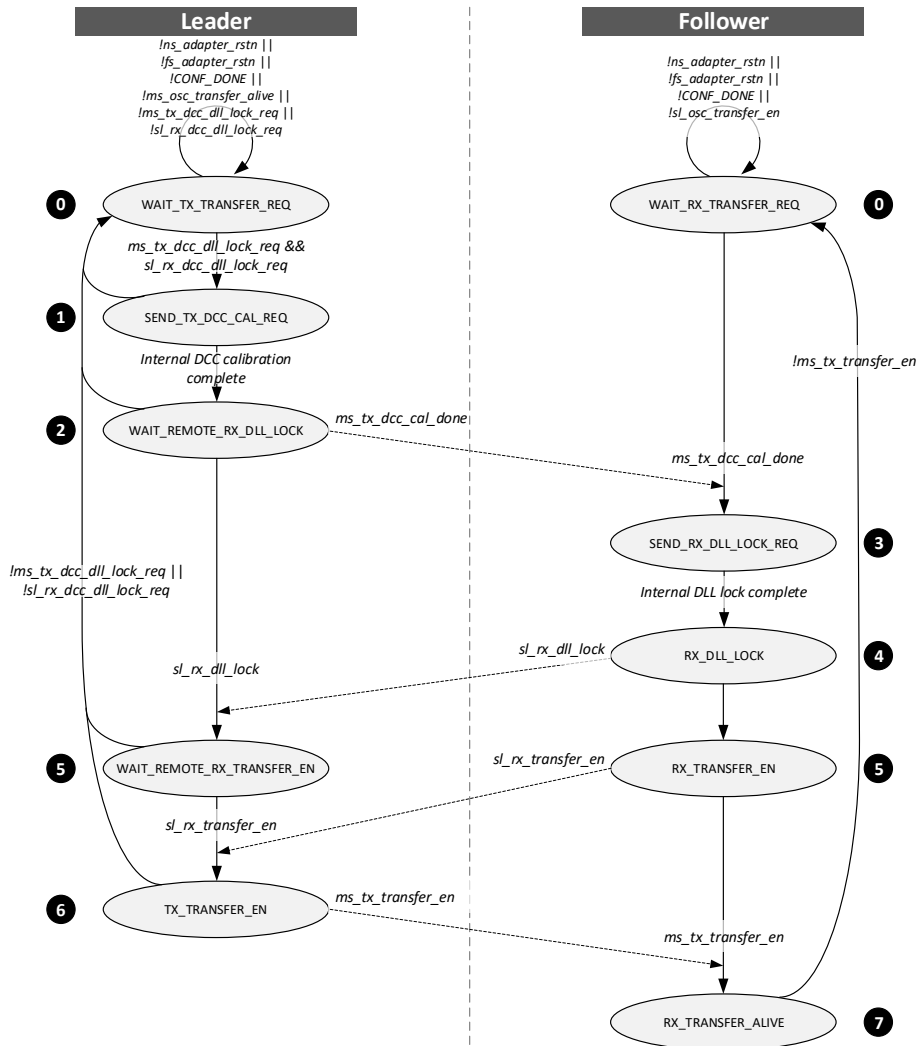


|                          |  |
|--------------------------|--|
|                          | completed calibration                            |
| <i>sl_rx_transfer_en</i> | Follower receive block has completed calibration |

**Table 17. Calibration Completion Signals**

#### **3.2.3.3.6 Calibration of Leader-to-Follower Datapath**

Leader-to-follower calibration shall comply with Figure 45. The numbers in black indicate the sequence of steps. The *ms\_osc\_transfer\_alive* signal shall come from the free-running clock synchronization state machine (Section 3.2.3.2).



**Figure 45. Leader-to-Follower Datapath Calibration State Machine**

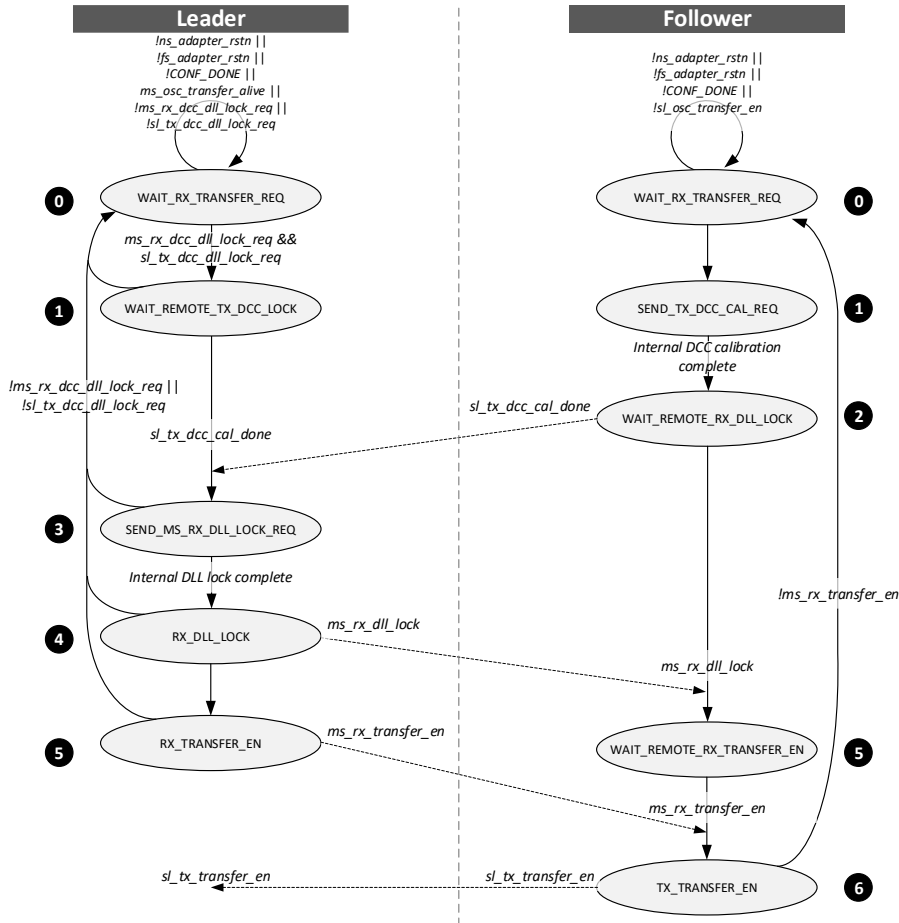
Signals used in the leader-to-follower calibration sequence are listed in Table 18.

| Signals                       | Description   |
|-------------------------------|---|
| <i>sl_rx_dcc_dll_lock_req</i> | Request from follower to start calibration. Once asserted, shall remain asserted until a new calibration is requested.  |
| <i>ms_tx_dcc_dll_lock_req</i> | Request from leader to start calibration. Once asserted, shall remain asserted until a new calibration is requested.  |
| <i>ms_tx_dcc_cal_done</i>     | Indicates that leader has completed its DCC calibration. Once asserted, shall remain asserted until a new calibration is requested.                             |
| <i>sl_rx_dll_lock</i>         | Indicates that follower has completed its DLL lock procedure. Once asserted, shall remain asserted until a new calibration is requested.                        |
| <i>sl_rx_transfer_en</i>      | Indicates that follower has completed its RX path calibration and is ready to receive data. Once asserted, shall remain asserted until calibration is complete. |
| <i>ms_tx_transfer_en</i>      | Indicates that leader has completed its TX path calibration and is ready to receive data.   |

**Table 18. Leader-to-Follower Calibration Signals**

#### 3.2.3.3.7 Calibration of Follower-to-Leader Datapath

Follower-to-leader calibration shall comply with Figure 46. The numbers in black indicate the sequence of steps. The *ms\_osc\_transfer\_alive* signal shall come from the free-running clock synchronization state machine (Section 3.2.3.2).



**Figure 46. Follower-to-Leader Datapath Calibration State Machine**

| Signals                       | Description  |
|-------------------------------|--|
| <i>sl_tx_dcc_dll_lock_req</i> | Request from follower to start calibration. Once asserted, shall remain asserted until a new calibration is requested. |
| <i>ms_rx_dcc_dll_lock_req</i> | Request from leader to start calibration. Once asserted, shall remain asserted until a new calibration is requested.   |

| Signals                   | Description   |
|---------------------------|---|
| <i>sl_tx_dcc_cal_done</i> | Indicates that follower has completed its DCC calibration. Once asserted, shall remain asserted until a new calibration is requested.                           |
| <i>ms_rx_dll_lock</i>     | Indicates that leader has completed its DLL lock procedure. Once asserted, shall remain asserted until a new calibration is requested.                          |
| <i>sl_tx_transfer_en</i>  | Indicates that follower has completed its TX path calibration and is ready to receive data. Once asserted, shall remain asserted until calibration is complete. |
| <i>ms_rx_transfer_en</i>  | Indicates that leader has completed its RX path calibration and is ready to receive data. Once asserted, shall remain asserted until calibration is complete.   |

**Table 19. Follower-to-Leader Calibration Signals**

### 3.2.4 AIB Link Ready

When both *sl\_tx\_transfer\_en* and *ms\_tx\_transfer\_en* are true, then the link shall be ready to transmit data.

## 3.3 Redundancy

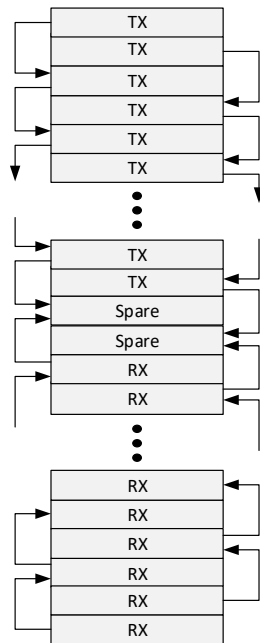
In order to improve interposer assembly yields, AIB interfaces shall implement a redundancy scheme. Data signals, clocks, and sideband control signals (AIB Plus only) shall implement an active redundancy scheme; *power\_on\_reset* and *device\_detect* signals shall implement a passive redundancy scheme.

### 3.3.1 Active Redundancy

Within each channel, two *spare* bumps shall be provided for implementation of active redundancy.

If a connection from one chiplet to another is either open or shorted, that signal shall be rerouted two signals away. All signals in the direction of the signal shift shall also have their signals rerouted by two signals. The shift shall terminate at the *spare* bumps. Signal shifting shall occur only between the faulty signal and the *spare* signal.

This signal shift shall be implemented by both interconnected chiplets.

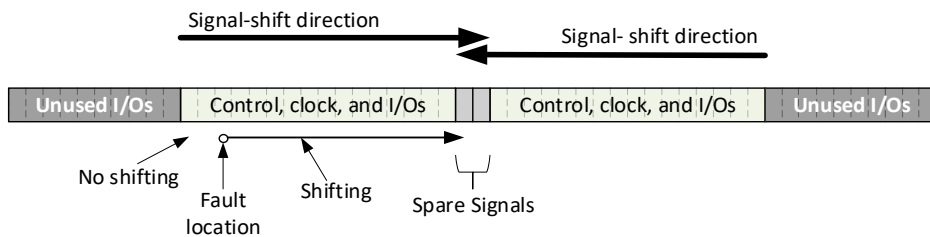


**Figure 47. Redundancy Routing**

When a faulty connection is detected and repaired, the output cell connected to the faulty connection shall be placed into standby mode.

For channels with unused data signals (Section 2.1.9), only used data signals shall be rerouted.

If two or more connections within one channel are faulty, then it shall not be possible to repair the connection.



**Figure 48. Direction of Redundancy Signal Shift**

### 3.3.1.1 Neighboring-Cell Requirements

Each I/O cell, in addition to implementing its primary function, shall be capable of

implementing the function of a neighboring rerouted signal. Specifically:

- The *spare* cells shall be configurable as output or input.
- Synchronous cells (outputs, inputs, clocks) that may carry rerouted asynchronous signals (control signals) shall be capable of asynchronous mode (Section 2.1.8).
- Asynchronous cells (control signals) that may carry rerouted synchronous signals (outputs, inputs, clocks) shall be capable of SDR synchronous mode (Section 2.1.3).
- In Gen2 mode a full voltage swing cell that may carry a rerouted low voltage swing signal (Table 20) shall be capable of operating in low swing mode if that cell is used in Gen2 mode (Table 4, Figure 47, Figure 56, Figure 57).

#### **3.3.1.2 Redundancy Storage**

During module test, if it is determined that a signal connection within a channel between two chiplets is faulty, the necessary redundancy activation shall be determined and stored in each chiplet.

#### **3.3.1.3 Redundancy Activation**

With each power-up event, the stored redundancy information shall be retrieved and applied.

#### **3.3.1.4 Redundancy Documentation**

Each chiplet should document both how to store redundancy and how to retrieve and activate redundancy in the data sheet.

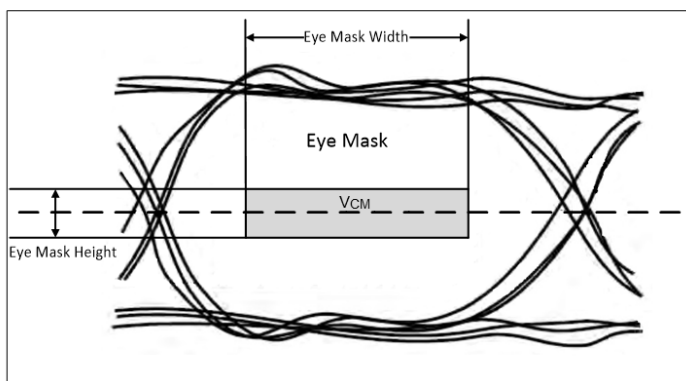
### **3.3.2 Passive Redundancy**

Two signals shall be active prior to configuration: *power\_on\_reset* and *device\_detect*. If the AUX Block is implemented using microbumps, in order to improve yields with respect to these signals, two bumps shall be provided for each signal and both bumps shall be soldered down to connect to two separate identical traces on the interposer. This passive redundancy scheme shall repair opens, but not shorts.

## 4 Electrical Specification

### 4.1 Eye Diagram

Compliance of data and clock signals shall be verified using a compliance mask on an eye diagram that specifies the minimum voltage swing ( $H_I - L_O$ ), the minimum duration during which the output voltage will be stable, and the maximum allowed over- and undershoot.



**Figure 49. Compliance Eye Mask**

Compliance is measured both at output bumps and at input bumps. It is applicable to both DDR/SDR modes for data, clocks or active control signals (such as *\*\_sr\_data*, and *\*\_sr\_load*). The connection between the near end and the far end is referred to as the *trace*.

For the receive-domain clock (Section 2.1.6.4), the near end is on the receiving side, since that is where that clock originates; the far end becomes the transmitting side. Timing specifications for the receive-domain clock are more stringent than those of the forwarded clock since the receive-domain clock traverses a trace from near end to far end before being forwarded across a second trace.

Signals (data, clock, and active control signals) shall meet the voltage and timing specifications provided in Table 20.

- Delays are specified in terms of *unit interval*, or *UI*. This refers to the minimum interval between data changes, and it is therefore a function of the clock frequency.
- For Gen1 the near-end timing specifications reflect microbumps with low-density pitch and a 0.5-pF capacitive load. For Gen2 the near-end timing specifications reflect microbumps with medium-density pitch and a 0.3pF capacitive load. Low, Medium and High-density bump pitches are defined in section 6.2. 55um is a low-density bump pitch. 45um and 36um are medium-density bump pitches.
- The “trace” timing specifications in Table 20 reference a channel that has 2dB



loss at 2GHz.

- Skew parameters are measured from the center of the eye.
- Jitter margin shall include all possible jitter contributors within the chiplet, including but not limited to reference clock jitter; intrinsic jitter contributed by any phase-locked loop, clock-data recovery, delay-lock loop, or the clock network; jitter caused by power-supply noise; and jitter caused by switching noise.

| Symbol           | Parameter   | Gen1 Mode and Gen2 Mode low-speed signals <sup>2</sup> |        |          | Gen2 Mode data and forwarded clock <sup>3</sup> |        |          |
|------------------|---|--|--------|----------|---|--------|----------|
|                  |   | Near end   | Trace  | Far end  | Near end  | Trace  | Far end  |
| V <sub>EH</sub>  | Minimum difference between LO and HI (eye diagram height)                         |  |        | +/-90 mV |   |        | +/-50 mV |
| V <sub>HI</sub>  | Minimum output voltage for HI state   | 0.7 V  |        |          | 0.4 V   |        |          |
| V <sub>LO</sub>  | Maximum output voltage for LO state   | 0 V  |        |          | 0 V   |        |          |
| V <sub>OS</sub>  | Typical output swing  | 0.9 V  |        |          | 0.4 V   |        |          |
| V <sub>CM</sub>  | Common mode voltage   | V <sub>OS</sub> / 2                                    |        |          | 0.2 V +/- 10mV <sup>1</sup>                     |        |          |
| t <sub>EW</sub>  | Minimum duration of valid output (eye diagram width) for data and forwarded clock | 0.56 UI  | 0.1 UI | 0.4 UI   | 0.65 UI   | 0.2 UI | 0.4 UI   |
| t <sub>BEW</sub> | Minimum duration of valid output (eye diagram width) for receive-domain clock     | 0.66 UI  | 0.1 UI | 0.56 UI  | n/a   | n/a    | n/a      |

1. V<sub>CM</sub> tolerance includes all board level effects, with power supply variation (e.g. regulator) within 5%.

2. "Gen2 Mode low-speed signals" includes all signals of [Table 4](#) except TX, RX, ns\_fwd\_clk, ns\_fwd\_clkb, fs\_fwd\_clk, fs\_fwd\_clkb.

3. "Gen2 Mode data and forward clock" are the signals TX, RX, ns\_fwd\_clk, ns\_fwd\_clkb, fs\_fwd\_clk, fs\_fwd\_clkb.

Formatted: Font: 9 pt

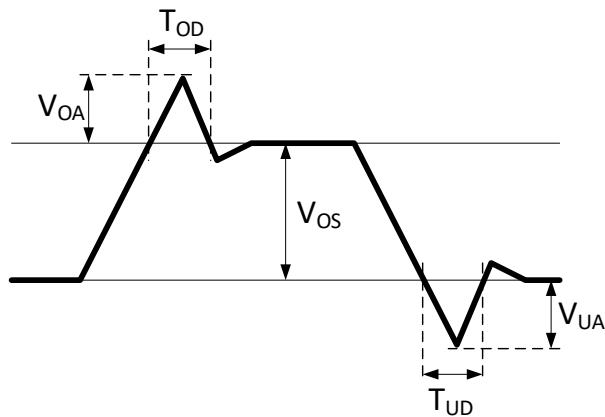
**Table 20. Electrical signal specifications**

## 4.2 Overshoot and Undershoot

Signal overshoot and undershoot at the receiver shall meet the specification in Table 21. Overshoot and Undershoot Specifications, where the symbols are illustrated in Figure 50. Overshoot and Undershoot.

| Symbol   | Parameter                         | Level                | Units |
|----------|-----------------------------------|----------------------|-------|
| $V_{OA}$ | Maximum peak overshoot amplitude  | $0.25 \cdot V_{OS}$  | V     |
| $V_{UA}$ | Maximum peak undershoot amplitude | $-0.25 \cdot V_{OS}$ | V     |
| $T_{OD}$ | Maximum overshoot duration        | 0.4                  | ns    |
| $T_{UD}$ | Maximum undershoot duration       | 0.4                  | ns    |

**Table 21. Overshoot and Undershoot Specifications**



**Figure 50. Overshoot and Undershoot at the Receiver**

### 4.3 Electrostatic Discharge (ESD) Protection

AIB IOs shall meet the ESD specifications in [Table 22](#).

| Parameter               | Minimum |
|-------------------------|---------|
| Discharge voltage (CDM) | 35 V    |
| Discharge current       | 300mA   |

**Table 22. ESD Specifications**

Formatted: Font: 12 pt, Font color: Auto

## 5 Design for Test

---

### 5.1 JTAG

#### 5.1.1 JTAG I/Os

All JTAG I/Os shall be fully compliant with the JTAG specification, IEEE 1149.1.

#### 5.1.2 JTAG cells

JTAG cells are simplified to eliminate the launch register. This means that:

- Data being shifted into the scan chain will be immediately applied during the shifting process without need for transferring the data from the chain into the launch registers. As data will be presented temporarily while being shifted, the data shall cause no undesired behavior while being shifted.
- It shall be possible to connect AIB JTAG cells into a single scan chain.
- It shall not be possible to create a single scan chain that mixes both standard JTAG cells and AIB JTAG cells.

The JTAG scan chain shall include all control and data I/Os, but not the *power\_on\_reset* or the *device\_detect* signals.

#### 5.1.3 AIB Private JTAG instructions

Manipulation of an AIB interface shall use the following private JTAG instructions. The operation codes should be documented in the chiplet data sheet.

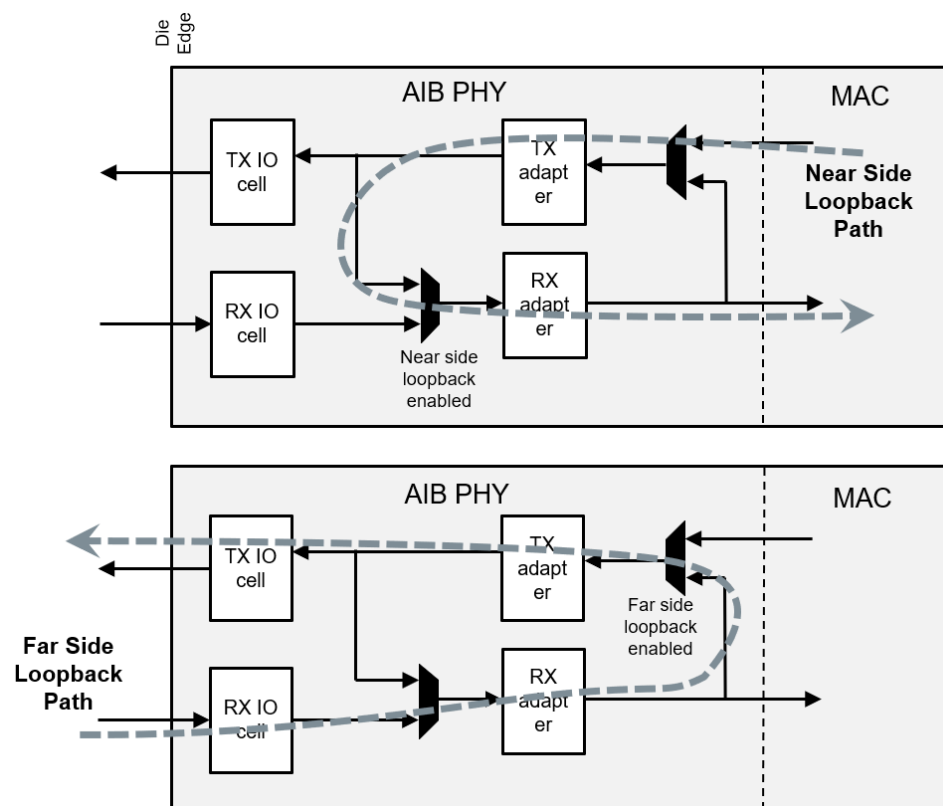
| Instruction Name   | Description   |
|--------------------|---|
| AIB_SHIFT_EN       | Enables boundary-scan register contents to be shifted out to TDO while test data is shifted into the boundary-scan registers via TDI.   |
| AIB_SHIFT_DIS      | Disables shifting of boundary-scan data out to TDO or in from TDI.  |
| AIB_TRANSMIT_EN    | Enables the forcing of a value onto output pins.  |
| AIB_TRANSMIT_DIS   | Disables the forcing of a value onto output pins  |
| AIB_RESET_EN       | Reset the registers in the input and output cells. Not effective until AIB_RESET_OVRD_EN is asserted to override the operational reset signal.  |
| AIB_RESET_DIS      | De-assert the reset signal from the registers in the input and output cells. Not effective until AIB_RESET_OVRD_EN is asserted to override the operational reset signal.                    |
| AIB_RESET_OVRD_EN  | Applies the reset state set by AIB_RESET_EN or AIB_RESET_DIS, overriding the operational reset signal.  |
| AIB_RESET_OVRD_DIS | Applies the reset state set by the operational reset signal.  |
| AIB_WEAKPU_EN      | Enable weak pull-up on all AIB IO blocks within a column.   |
| AIB_WEAKPU_DIS     | Disable weak pull-up on all AIB IO blocks within a column.  |
| AIB_WEAKPDN_EN     | Enable weak pull-down on all AIB IO blocks within a column.   |
| AIB_WEAKPDN_DIS    | Disable weak pull-down on all AIB IO cells within a column.   |
| AIB_INTEST_EN      | Enable testing of data path between the MAC layer (AIB Base) or the AIB Adapter (AIB Plus) and the I/O block.   |
| AIB_INTEST_DIS     | Disable testing of data path between the MAC layer (AIB Base) or the AIB Adapter (AIB Plus) and the I/O block.  |
| AIB_JTAG_CLKSEL    | Select the JTAG clock or the operational clock for the register in the I/O block. Takes an argument: if HI, then the JTAG clock is selected; if LO, then the operational clock is selected. |

**Table 23. Private JTAG instructions**

## 5.2 Loopback

An AIB Plus Gen2 interface shall have Near Side and Far Side data loopback modes

usable for test. Figure 51 shows a Near Side Loopback Path, used for loopback into the AIB PHY and back to the MAC. The Far Side Loopback Path of Figure 51 may be used between die, where the far side chiplet is sending data and receiving the looped-back data.



**Figure 51. Near Side and Far Side Loopbacks**

When in near side loopback mode, the AIB interface shall connect full rate TX data to the RX adapter input data and shall connect the *ns\_fwd\_clk* to the RX adapter and MAC/PHY interface. When in far side loopback mode, the AIB interface shall connect incoming RX data to the TX data adapter input and shall connect the incoming *fs\_fwd\_clk* to the outgoing *ns\_fwd\_clk*.

RX and TX data LSBs shall be mapped to each other, continuing for all bits up to RX MSB and TX MSB mapped to each other.

The interface shall have a control to turn on loopback mode, for example in a test configuration of the interface.

## 6 Physical Signal Arrangement

### 6.1 Interface Orientation

Die orientation shall be with respect to the die facing up with the alignment mark (or the [0,0] origin) at the lower left. Sides shall be referred to as East/West/North/South with respect to this orientation. Interface orientation should be documented in the chiplet data sheet.

- The west interface shall have Channel 0 at the bottom. This shall be a follower or dual-mode interface.
- The north interface shall have Channel 0 at the left. This shall be a follower or dual-mode interface.
- The east interface shall have Channel 0 at the bottom. This shall be a leader or dual-mode interface.
- The south interface shall have Channel 0 at the left. This shall be a leader or dual-mode interface.
- The AUX block, if implemented using microbumps, shall be placed at the end of a column next to Channel 0.

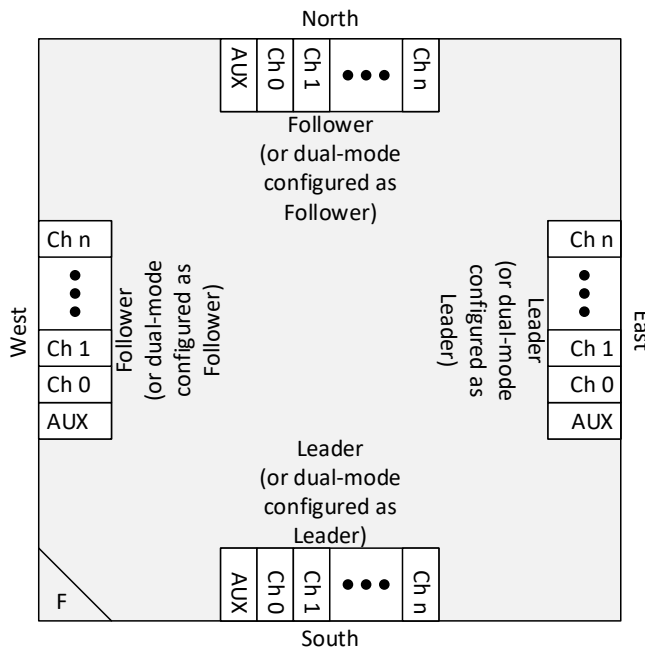
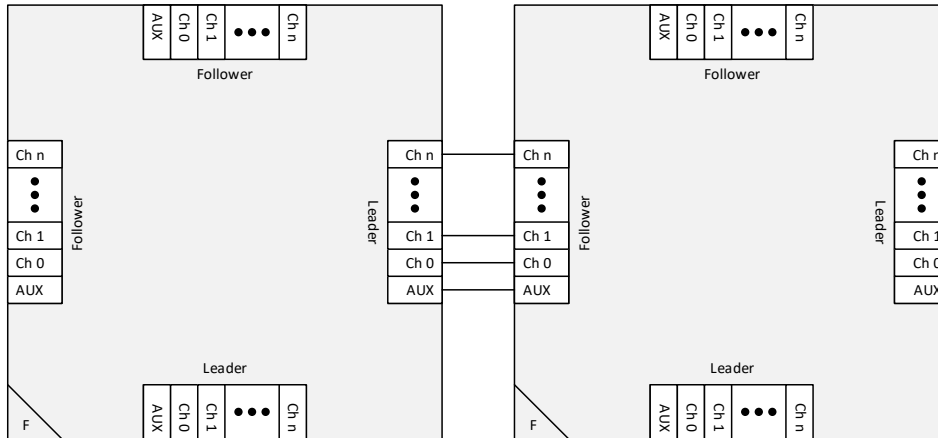


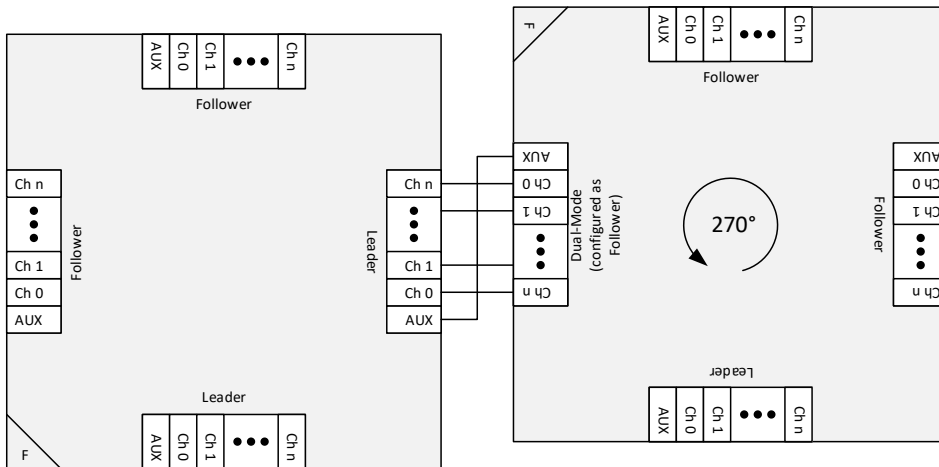
Figure 52. Interface Orientation

This convention facilitates interconnection of interfaces between chiplets in a way that ensures correct leader/follower and channel/AUX interconnect (Figure 53).



**Figure 53. Standard Chiplet-to-Chiplet Interconnection**

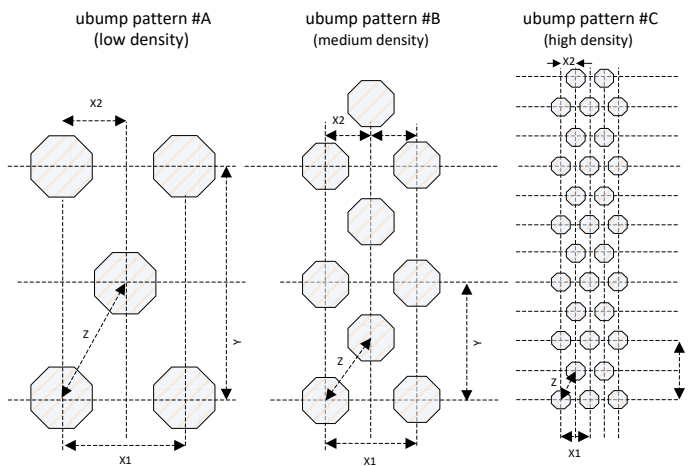
In the event of a chiplet being rotated from the standard orientation prior to being connected, the interface on the rotated chiplet must be dual-mode and configured to ensure a leader/follower relationship (example shown in Figure 54). This is intended to facilitate straight, balanced signal connections. The long AUX (Section 1.3.4.4) connections are acceptable, since these are static signals.



**Figure 54. Rotated Chiplet-to-Chiplet Interconnection**

## 6.2 Bump Configuration

Microbumps shall be configured in staggered arrays as shown in Figure 55.



**Figure 55. Bump Spacing for Microbump Array**

There are three supported bump arrays for low-density, medium-density and high-density microbumps.

- Low-density may be used with any microbump pitch up to 55  $\mu\text{m}$ .



- Medium-density may be used with any microbump pitch up to 52  $\mu\text{m}$ .
- High-density may be used with any microbump pitch up to 20  $\mu\text{m}$ .

The spacing of the microbump array shall meet the following specifications (Table 24) in  $\mu\text{m}$ . The X1 dimension may be reduced, but the Y dimension shall not be reduced. The channel height shall be fixed at 312.48  $\mu\text{m}$ . For medium and high-density arrays, depending on actual minimum bump pitch technology, either X1 or Z reaches minimum bump pitch first such that other dimension (Z or X1) will be set to non-minimum value.

| Symbol | Description                       | Low-density  | Medium-density   | High-density   |
|--------|-----------------------------------|--|--|--|
| X1     | Aligned-row bump-to-bump pitch    | 55 or $\geq$ minimum bump pitch such that Z minimum is met | 55 or $\geq$ minimum bump pitch such that Z minimum is met | 55 or $\geq$ minimum bump pitch such that Z minimum is met |
| X2     | Staggered-row bump-to-bump pitch  | $X1 \div 2$  | $X1 \div 2$  | $X1 \div 2$  |
| Z      | Diagonal bump-to-bump pitch       | $\geq$ minimum bump pitch                                  | $\geq$ minimum bump pitch                                  | $\geq$ minimum bump pitch                                  |
| Y      | Aligned-column bump-to-bump pitch | 104.16   | 52.08  | 26.04  |

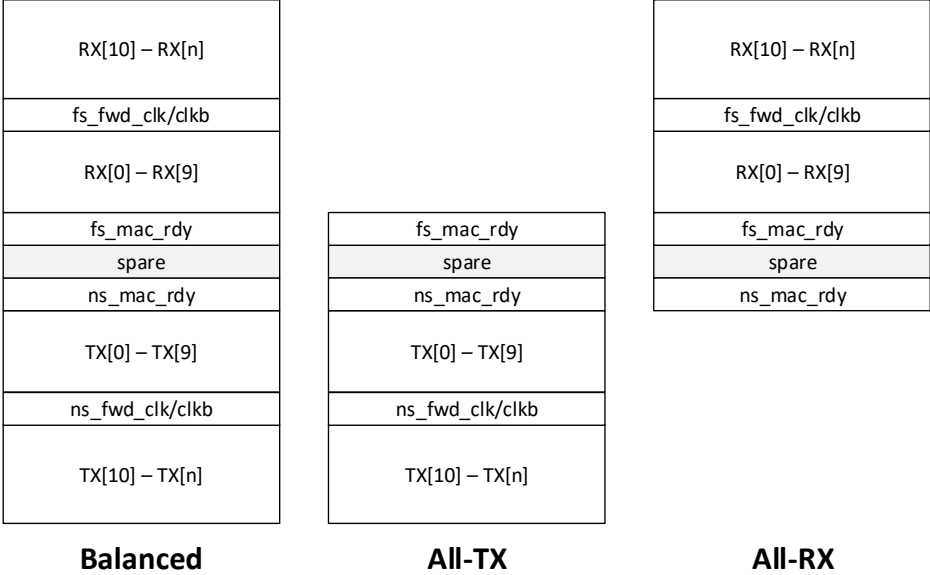
**Table 24. Bump Spacing Specifications for Bump Array**

## 6.3 Bump Assignment Process

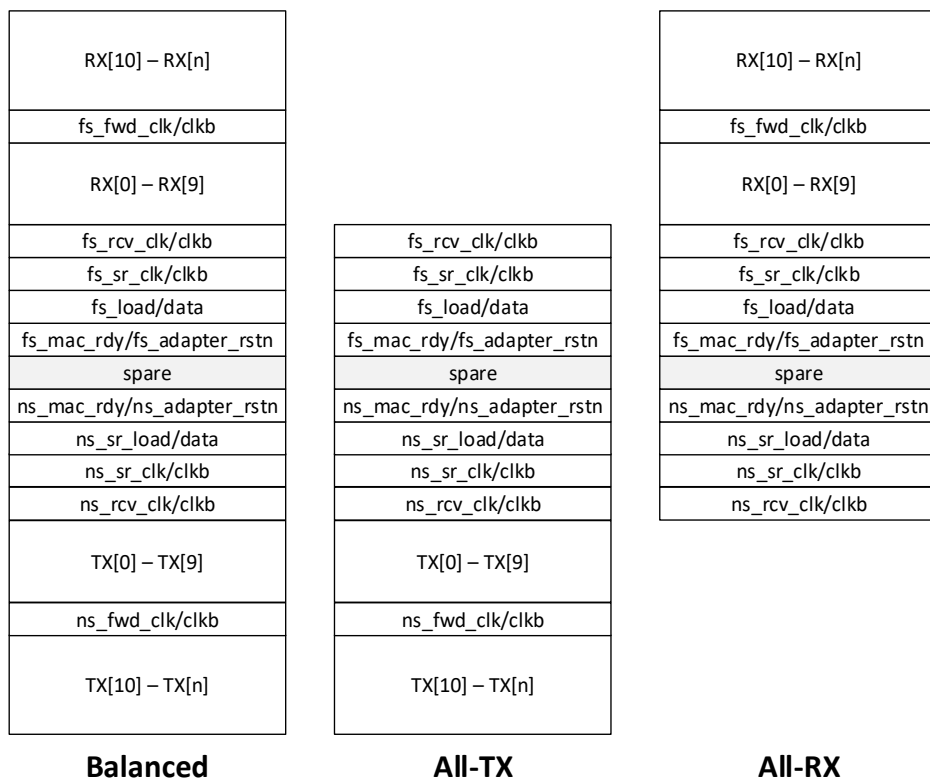
The low-density bump assignment process is based on the signal relationships for Balanced, All-TX, and All-RX configurations for both AIB Base and AIB Plus, as illustrated in Figure 56 and Figure 57. The goal of this placement process is to ensure the shortest, most direct connections between the near side and the far side. The connections between functional signals are described above. The *spare(0)* signal from the near side connects to the *spare(1)* signal on the far side; the *spare(1)* signal from the near side connects to the *spare(0)* signal on the far side.

The medium-density bump array is derived from the low-density bump array. The high-density bump array is derived from the medium-density bump array.

All bump arrays share the same bump table process.



**Figure 56. Signal Relationships for AIB Base Configurations**



**Figure 57. Signal Relationships for AIB Plus Configurations**

### 6.3.1 Data-Signal Bump Assignment

Because a channel may have a range of data signals as defined in [Table 5](#), specific bump assignments for every possible configuration are not practical. Instead, an algorithm determines how bumps shall be assigned for any legal data signal count.

Bumps shall be assigned as follows. Any bumps labeled as “(empty)” are available for any connection that doesn’t involved an active AIB signal.

#### 6.3.1.1 Gen1 and Gen2 Signals

As specified in Table 4, signals not used in Gen2 are still allocated in the bump tables for compatibility. See the notes in Table 4 for handling of those signals when in Gen2 mode.

#### 6.3.1.2 Bump Table

1. Create a *bump table* with pairs of signals. Odd-numbered signals are on the left, even-numbered signals are on the right.

| Bump ID | Bump Name | IO | Bump ID | Bump Name | IO |
|---------|-----------|----|---------|-----------|----|
|         |           |    |         |           |    |

**Table 25. Bump Table: Starting**

- The middle row of the table shall be for the *spare* signals (left and right).

| Bump ID | Bump Name | IO  | Bump ID | Bump Name | IO  |
|---------|-----------|-----|---------|-----------|-----|
|         |           |     |         |           |     |
|         | spare[0]  | I/O |         | spare[1]  | I/O |
|         |           |     |         |           |     |

**Table 26. Bump Table: Spare signals**

#### 6.3.1.2.1 AIB Base, Balanced

- Moving up from the middle, in order, the following signals, which are all inputs from the far side, shall be placed:
  - Empty on left (since no adapter reset); MAC ready on right
  - Five pairs of data inputs
  - The forwarded negative clock on left; the forwarded positive clock on right
  - The remaining pairs of used inputs
  - Any unused inputs

| Bump ID | Bump Name   | IO  | Bump ID | Bump Name  | IO  |
|---------|-------------|-----|---------|------------|-----|
|         | RX[n-1]     | in  |         | RX[n-2]    | in  |
|         | ...         |     |         | ...        |     |
|         | RX[11]      | in  |         | RX[10]     | in  |
|         | fs_fwd_clkb | in  |         | fs_fwd_clk | in  |
|         | RX[9]       | in  |         | RX[8]      | in  |
|         | RX[7]       | in  |         | RX[6]      | in  |
|         | RX[5]       | in  |         | RX[4]      | in  |
|         | RX[3]       | in  |         | RX[2]      | in  |
|         | RX[1]       | in  |         | RX[0]      | in  |
|         | (empty)     | in  |         | fs_mac_rdy | in  |
|         | spare[0]    | I/O |         | spare[1]   | I/O |
|         |             |     |         |            |     |

**Table 27. Bump Table: Inputs**

### (AIB Base, Balanced)

4. The “edge of chiplet” shall be directly above the topmost row.
5. Moving down from the *spare* signal, the following signals, which are all outputs to the far side, shall be placed:
  1. MAC ready on left; empty on right (since no adapter reset)
  2. Five pairs of data outputs
  3. The forwarded positive clock on left; the forwarded negative clock on right
  4. The remaining pairs of used outputs.
  5. Any unused outputs

| Bump ID | Bump Name   | IO  | Bump ID | Bump Name   | IO  |
|---------|-------------|-----|---------|-------------|-----|
|         | RX[n-1]     | in  |         | RX[n-2]     | in  |
|         | ...         |     |         | ...         |     |
|         | fs_fwd_clkb | in  |         | fs_fwd_clk  | in  |
|         | RX[9]       | in  |         | RX[8]       | in  |
|         | RX[7]       | in  |         | RX[6]       | in  |
|         | RX[5]       | in  |         | RX[4]       | in  |
|         | RX[3]       | in  |         | RX[2]       | in  |
|         | RX[1]       | in  |         | RX[0]       | in  |
|         | (empty)     | in  |         | fs_mac_rdy  | in  |
|         | spare[0]    | I/O |         | spare[1]    | I/O |
|         | ns_mac_rdy  | out |         | (empty)     | out |
|         | TX[0]       | out |         | TX[1]       | out |
|         | TX[2]       | out |         | TX[3]       | out |
|         | TX[4]       | out |         | TX[5]       | out |
|         | TX[6]       | out |         | TX[7]       | out |
|         | TX[8]       | out |         | TX[9]       | out |
|         | ns_fwd_clk  | out |         | ns_fwd_clkb | out |
|         | TX[10]      | out |         | TX[11]      | out |
|         | ...         |     |         | ...         |     |
|         | TX[n-2]     | out |         | TX[n-1]     | out |

**Table 28. Bump Table: Complete  
(AIB Base, Balanced)**

#### 6.3.1.2.2 AIB Base, All-TX

3. Moving up from the middle, in order, the following signals, which are all inputs from the far side, shall be placed:
  1. Empty on left (since no adapter reset); MAC ready on right

| Bump ID | Bump Name | IO  |
|---------|-----------|-----|
|         | (empty)   | in  |
|         | spare[0]  | I/O |
|         |           |     |

| Bump ID | Bump Name  | IO  |
|---------|------------|-----|
|         | fs_mac_rdy | in  |
|         | spare[1]   | I/O |
|         |            |     |

**Table 29. Bump Table: Inputs  
(AIB Base, All-TX)**

4. The “edge of chiplet” shall be directly above the topmost row.
5. Moving down from the *spare* signal, the following signals, which are all outputs to the far side, shall be placed:
  1. MAC ready on left; empty on right (since no adapter reset)
  2. Five pairs of data outputs
  3. The forwarded positive clock on left; the forwarded negative clock on right
  4. The remaining pairs of used outputs.
  5. Any unused outputs

| Bump ID | Bump Name  | IO  |
|---------|------------|-----|
|         | (empty)    | in  |
|         | spare[0]   | I/O |
|         | ns_mac_rdy | out |
|         | TX[0]      | out |
|         | TX[2]      | out |
|         | TX[4]      | out |
|         | TX[6]      | out |
|         | TX[8]      | out |
|         | ns_fwd_clk | out |
|         | TX[10]     | out |
|         | ...        |     |
|         | TX[n-2]    | out |

| Bump ID | Bump Name   | IO  |
|---------|-------------|-----|
|         | fs_mac_rdy  | in  |
|         | spare[1]    | I/O |
|         | (empty)     | out |
|         | TX[1]       | out |
|         | TX[3]       | out |
|         | TX[5]       | out |
|         | TX[7]       | out |
|         | TX[9]       | out |
|         | ns_fwd_clkb | out |
|         | TX[11]      | out |
|         | ...         |     |
|         | TX[n-1]     | out |

**Table 30. Bump Table: Complete  
(AIB Base, All-TX)**

#### 6.3.1.2.3 AIB Base, All-RX

3. Moving up from the middle, in order, the following signals, which are all inputs from the far side, shall be placed:
  1. Empty on left (since no adapter reset); MAC ready on right
  2. Five pairs of data inputs
  3. The forwarded negative clock on left; the forwarded positive clock on right
  4. The remaining pairs of used inputs
  5. Any unused inputs

| Bump ID | Bump Name   | IO  |
|---------|-------------|-----|
|         | RX[n-1]     | in  |
|         | ...         |     |
|         | RX[11]      | in  |
|         | fs_fwd_clkb | in  |
|         | RX[9]       | in  |
|         | RX[7]       | in  |
|         | RX[5]       | in  |
|         | RX[3]       | in  |
|         | RX[1]       | in  |
|         | (empty)     | in  |
|         | spare[0]    | I/O |
|         |             |     |

| Bump ID | Bump Name  | IO  |
|---------|------------|-----|
|         | RX[n-2]    | in  |
|         | ...        |     |
|         | RX[10]     | in  |
|         | fs_fwd_clk | in  |
|         | RX[8]      | in  |
|         | RX[6]      | in  |
|         | RX[4]      | in  |
|         | RX[2]      | in  |
|         | RX[0]      | in  |
|         | fs_mac_rdy | in  |
|         | spare[1]   | I/O |
|         |            |     |

**Table 31. Bump Table: Inputs  
(AIB Base, All-RX)**

4. The “edge of chiplet” shall be directly above the topmost row.
5. Moving down from the *spare* signal, the following signals, which are all outputs to the far side, shall be placed:
  1. MAC ready on left; empty on right (since no adapter reset)

| Bump ID | Bump Name   | IO  |
|---------|-------------|-----|
|         | RX[n-1]     | in  |
|         | ...         |     |
|         | fs_fwd_clkb | in  |
|         | RX[9]       | in  |
|         | RX[7]       | in  |
|         | RX[5]       | in  |
|         | RX[3]       | in  |
|         | RX[1]       | in  |
|         | (empty)     | in  |
|         | spare[0]    | I/O |
|         | ns_mac_rdy  | out |

| Bump ID | Bump Name  | IO  |
|---------|------------|-----|
|         | RX[n-2]    | in  |
|         | ...        |     |
|         | fs_fwd_clk | in  |
|         | RX[8]      | in  |
|         | RX[6]      | in  |
|         | RX[4]      | in  |
|         | RX[2]      | in  |
|         | RX[0]      | in  |
|         | fs_mac_rdy | in  |
|         | spare[1]   | I/O |
|         | (empty)    | out |

**Table 32. Bump Table: Complete  
(AIB Base, All-RX)**

#### 6.3.1.2.4 AIB Plus, Balanced

3. Moving up from the middle, in order, the following signals, which are all inputs from the far side, shall be placed:

1. Adapter reset on left; MAC ready on right
2. Shift-register load on left; shift-register data on right
3. Shift-register negative clock on left; shift-register positive clock on right
4. The receive-domain negative clock on left; the receive-domain positive clock on right
5. Five pairs of data inputs
6. The forwarded negative clock on left; the forwarded positive clock on right
7. The remaining pairs of used inputs
8. Any unused inputs

| Bump ID | Bump Name       | IO  | Bump ID | Bump Name  | IO  |
|---------|-----------------|-----|---------|------------|-----|
|         | RX[n-1]         | in  |         | RX[n-2]    | in  |
|         | ...             |     |         | ...        |     |
|         | RX[11]          | in  |         | RX[10]     | in  |
|         | fs_fwd_clkb     | in  |         | fs_fwd_clk | in  |
|         | RX[9]           | in  |         | RX[8]      | in  |
|         | RX[7]           | in  |         | RX[6]      | in  |
|         | RX[5]           | in  |         | RX[4]      | in  |
|         | RX[3]           | in  |         | RX[2]      | in  |
|         | RX[1]           | in  |         | RX[0]      | in  |
|         | fs_rcv_clkb     | in  |         | fs_rcv_clk | in  |
|         | fs_sr_clkb      | in  |         | fs_sr_clk  | in  |
|         | fs_sr_load      | in  |         | fs_sr_data | in  |
|         | fs_adapter_rstn | in  |         | fs_mac_rdy | in  |
|         | spare[0]        | I/O |         | spare[1]   | I/O |
|         |                 |     |         |            |     |

**Table 33. Bump Table: Inputs  
(AIB Plus, Balanced)**

4. The "edge of chiplet" shall be directly above the topmost row.
5. Moving down from the *spare* signal, the following signals, which are all outputs to the far side, shall be placed:
  1. MAC ready on left; adapter reset on right
  2. Shift-register data on left; shift-register load on right
  3. Shift-register positive clock on left; shift-register negative clock on right
  4. The receive-domain positive clock on left; the receive-domain negative clock on right
  5. Five pairs of data outputs
  6. The forwarded positive clock on left; the forwarded negative clock on right
  7. The remaining pairs of used outputs.
  8. Any unused outputs



| Bump ID | Bump Name       | IO  | Bump ID | Bump Name       | IO  |
|---------|-----------------|-----|---------|-----------------|-----|
|         | RX[n-1]         | in  |         | RX[n-2]         | in  |
|         | ...             |     |         | ...             |     |
|         | fs_fwd_clkb     | in  |         | fs_fwd_clk      | in  |
|         | RX[9]           | in  |         | RX[8]           | in  |
|         | RX[7]           | in  |         | RX[6]           | in  |
|         | RX[5]           | in  |         | RX[4]           | in  |
|         | RX[3]           | in  |         | RX[2]           | in  |
|         | RX[1]           | in  |         | RX[0]           | in  |
|         | fs_rcv_clkb     | in  |         | fs_rcv_clk      |     |
|         | fs_sr_clkb      | in  |         | fs_sr_clk       | in  |
|         | fs_sr_load      | in  |         | fs_sr_data      | in  |
|         | fs_adapter_rstn | in  |         | fs_mac_rdy      | in  |
|         | spare[0]        | I/O |         | spare[1]        | I/O |
|         | ns_mac_rdy      | out |         | ns_adapter_rstn | out |
|         | ns_sr_data      | out |         | ns_sr_load      | out |
|         | ns_sr_clk       | out |         | ns_sr_clkb      | out |
|         | ns_rcv_clk      | out |         | ns_rcv_clkb     | out |
|         | TX[0]           | out |         | TX[1]           | out |
|         | TX[2]           | out |         | TX[3]           | out |
|         | TX[4]           | out |         | TX[5]           | out |
|         | TX[6]           | out |         | TX[7]           | out |
|         | TX[8]           | out |         | TX[9]           | out |
|         | ns_fwd_clk      | out |         | ns_fwd_clkb     | out |
|         | TX[10]          | out |         | TX[11]          | out |
|         | ...             |     |         | ...             |     |
|         | TX[n-2]         | out | AIB1    | TX[n-1]         | out |

**Table 34. Bump Table: Complete  
(AIB Plus, Balanced)**

#### 6.3.1.2.5 AIB Plus, All-TX

3. Moving up from the middle, in order, the following signals, which are all inputs from the far side, shall be placed:
  1. Adapter reset on left; MAC ready on right
  2. Shift-register load on left; shift-register data on right
  3. Shift-register negative clock on left; shift-register positive clock on right
  4. The receive-domain negative clock on left; the receive-domain positive clock on right

| Bump ID | Bump Name       | IO  |
|---------|-----------------|-----|
|         | fs_rcv_clkb     | in  |
|         | fs_sr_clkb      | in  |
|         | fs_sr_load      | in  |
|         | fs_adapter_rstn | in  |
|         | spare[0]        | I/O |
|         |                 |     |

| Bump ID | Bump Name  | IO  |
|---------|------------|-----|
|         | fs_rcv_clk | in  |
|         | fs_sr_clk  | in  |
|         | fs_sr_data | in  |
|         | fs_mac_rdy | in  |
|         | spare[1]   | I/O |
|         |            |     |

**Table 35. Bump Table: Inputs  
(AIB Plus, AII-TX)**

4. The “edge of chiplet” shall be directly above the topmost row.
5. Moving down from the *spare* signal, the following signals, which are all outputs to the far side, shall be placed:
  1. MAC ready on left; adapter reset on right
  2. Shift-register data on left; shift-register load on right
  3. Shift-register positive clock on left; shift-register negative clock on right
  4. Empty bumps since there is no receive-domain clock output
  5. Five pairs of data outputs
  6. The forwarded positive clock on left; the forwarded negative clock on right
  7. The remaining pairs of used outputs.
  8. Any unused outputs

| Bump ID | Bump Name       | IO  |
|---------|-----------------|-----|
|         | fs_rcv_clkb     | in  |
|         | fs_sr_clkb      | in  |
|         | fs_sr_load      | in  |
|         | fs_adapter_rstn | in  |
|         | spare[0]        | I/O |
|         | ns_mac_rdy      | out |
|         | ns_sr_data      | out |
|         | ns_sr_clk       | out |
|         | (empty)         | out |
|         | TX[0]           | out |
|         | TX[2]           | out |
|         | TX[4]           | out |
|         | TX[6]           | out |
|         | TX[8]           | out |
|         | ns_fwd_clk      | out |
|         | TX[10]          | out |
|         | ...             |     |
|         | TX[n-2]         | out |

| Bump ID | Bump Name       | IO  |
|---------|-----------------|-----|
|         | fs_rcv_clk      |     |
|         | fs_sr_clk       | in  |
|         | fs_sr_data      | in  |
|         | fs_mac_rdy      | in  |
|         | spare[1]        | I/O |
|         | ns_adapter_rstn | out |
|         | ns_sr_load      | out |
|         | ns_sr_clkb      | out |
|         | (empty)         | out |
|         | TX[1]           | out |
|         | TX[3]           | out |
|         | TX[5]           | out |
|         | TX[7]           | out |
|         | TX[9]           | out |
|         | ns_fwd_clkb     | out |
|         | TX[11]          | out |
|         | ...             |     |
|         | TX[n-1]         | out |

**Table 36. Bump Table: Complete**

## (AIB Plus, AII-TX)

### 6.3.1.2.6 AIB Plus, AII-RX

3. Moving up from the middle, in order, the following signals, which are all inputs from the far side, shall be placed:
  1. Adapter reset on left; MAC ready on right
  2. Shift-register load on left; shift-register data on right
  3. Shift-register negative clock on left; shift-register positive clock on right
  4. Empty bumps since there is no receive-domain clock input
  5. Five pairs of data inputs
  6. The forwarded negative clock on left; the forwarded positive clock on right
  7. The remaining pairs of used inputs
  8. Any unused inputs

| Bump ID | Bump Name       | IO  | Bump ID | Bump Name  | IO  |
|---------|-----------------|-----|---------|------------|-----|
|         | RX[n-1]         | in  |         | RX[n-2]    | in  |
|         | ...             |     |         | ...        |     |
|         | RX[11]          | in  |         | RX[10]     | in  |
|         | fs_fwd_clkb     | in  |         | fs_fwd_clk | in  |
|         | RX[9]           | in  |         | RX[8]      | in  |
|         | RX[7]           | in  |         | RX[6]      | in  |
|         | RX[5]           | in  |         | RX[4]      | in  |
|         | RX[3]           | in  |         | RX[2]      | in  |
|         | RX[1]           | in  |         | RX[0]      | in  |
|         | (empty)         | in  |         | (empty)    | in  |
|         | fs_sr_clkb      | in  |         | fs_sr_clk  | in  |
|         | fs_sr_load      | in  |         | fs_sr_data | in  |
|         | fs_adapter_rstn | in  |         | fs_mac_rdy | in  |
|         | spare[0]        | I/O |         | spare[1]   | I/O |
|         |                 |     |         |            |     |

**Table 37. Bump Table: Inputs  
(AIB Plus, AII-RX)**

4. The “edge of chiplet” shall be directly above the topmost row.
5. Moving down from the *spare* signal, the following signals, which are all outputs to the far side, shall be placed:
  1. MAC ready on left; adapter reset on right
  2. Shift-register data on left; shift-register load on right
  3. Shift-register positive clock on left; shift-register negative clock on right
  4. The forwarded positive clock on left; the forwarded negative clock on right
  5. Five pairs of data outputs
  6. The borrowed positive clock on left; the borrowed negative clock on right

7. The remaining pairs of used outputs.
8. Any unused outputs

| Bump ID | Bump Name       | IO  | Bump ID | Bump Name       | IO  |
|---------|-----------------|-----|---------|-----------------|-----|
|         | RX[n-1]         | in  |         | RX[n-2]         | in  |
|         | ...             |     |         | ...             |     |
|         | fs_fwd_clkb     | in  |         | fs_fwd_clk      | in  |
|         | RX[9]           | in  |         | RX[8]           | in  |
|         | RX[7]           | in  |         | RX[6]           | in  |
|         | RX[5]           | in  |         | RX[4]           | in  |
|         | RX[3]           | in  |         | RX[2]           | in  |
|         | RX[1]           | in  |         | RX[0]           | in  |
|         | (empty)         | in  |         | (empty)         |     |
|         | fs_sr_clkb      | in  |         | fs_sr_clk       | in  |
|         | fs_sr_load      | in  |         | fs_sr_data      | in  |
|         | fs_adapter_rstn | in  |         | fs_mac_rdy      | in  |
|         | spare[0]        | I/O |         | spare[1]        | I/O |
|         | ns_mac_rdy      | out |         | ns_adapter_rstn | out |
|         | ns_sr_data      | out |         | ns_sr_load      | out |
|         | ns_sr_clk       | out |         | ns_sr_clkb      | out |
|         | ns_rcv_clk      | out |         | ns_rcv_clkb     | out |

**Table 38. Bump Table: Complete  
(AIB Plus, All-RX)**

6. Finally, assign bump ID starting at the bottom left with AIB0, bottom right with AIB1, and then moving up – even numbers on the left, odd numbers on the right.

The following tables illustrate the six versions of this algorithm for AIB Base and AIB Plus; Balanced, All-TX, and All-RX configurations. *n* represents the number of RX and/or TX signals.

| Bump ID | Bump Name   | IO  | Bump ID | Bump Name   | IO  |
|---------|-------------|-----|---------|-------------|-----|
| AIBy    | RX[n-1]     | in  | AIBy+1  | RX[n-2]     | in  |
|         | ...         |     |         | ...         |     |
| AIBx+28 | fs_fwd_clkb | in  | AIBx+29 | fs_fwd_clk  | in  |
| AIBx+26 | RX[9]       | in  | AIBx+27 | RX[8]       | in  |
| AIBx+24 | RX[7]       | in  | AIBx+25 | RX[6]       | in  |
| AIBx+22 | RX[5]       | in  | AIBx+23 | RX[4]       | in  |
| AIBx+20 | RX[3]       | in  | AIBx+21 | RX[2]       | in  |
| AIBx+18 | RX[1]       | in  | AIBx+19 | RX[0]       | in  |
| AIBx+16 | (empty)     | in  | AIBx+17 | fs_mac_rdy  | in  |
| AIBx+14 | spare[0]    | I/O | AIBx+15 | spare[1]    | I/O |
| AIBx+12 | ns_mac_rdy  | out | AIBx+13 | (empty)     | out |
| AIBx+10 | TX[0]       | out | AIBx+11 | TX[1]       | out |
| AIBx+8  | TX[2]       | out | AIBx+9  | TX[3]       | out |
| AIBx+6  | TX[4]       | out | AIBx+7  | TX[5]       | out |
| AIBx+4  | TX[6]       | out | AIBx+5  | TX[7]       | out |
| AIBx+2  | TX[8]       | out | AIBx+3  | TX[9]       | out |
| AIBx    | ns_fwd_clk  | out | AIBx+1  | ns_fwd_clkb | out |
|         | ...         |     |         | ...         |     |
| AIB0    | TX[n-2]     | out | AIB1    | TX[n-1]     | out |

**Table 39. Bump-Table Exemplar (AIB Base, Balanced)**  
 $x=n-10; y=2n+8$

| Bump ID | Bump Name  | IO  | Bump ID | Bump Name   | IO  |
|---------|------------|-----|---------|-------------|-----|
| AIBx+16 | (empty)    | in  | AIBx+17 | fs_mac_rdy  | in  |
| AIBx+14 | spare[0]   | I/O | AIBx+15 | spare[1]    | I/O |
| AIBx+12 | ns_mac_rdy | out | AIBx+13 | (empty)     | out |
| AIBx+10 | TX[0]      | out | AIBx+11 | TX[1]       | out |
| AIBx+8  | TX[2]      | out | AIBx+9  | TX[3]       | out |
| AIBx+6  | TX[4]      | out | AIBx+7  | TX[5]       | out |
| AIBx+4  | TX[6]      | out | AIBx+5  | TX[7]       | out |
| AIBx+2  | TX[8]      | out | AIBx+3  | TX[9]       | out |
| AIBx    | ns_fwd_clk | out | AIBx+1  | ns_fwd_clkb | out |
|         | ...        |     |         | ...         |     |
| AIB0    | TX[n-2]    | out | AIB1    | TX[n-1]     | out |

**Table 40. Bump-Table Exemplar (AIB Base, All-TX)**  
 $x=n-10$

| Bump ID | Bump Name   | IO  | Bump ID | Bump Name  | IO  |
|---------|-------------|-----|---------|------------|-----|
| AIBy    | RX[n-1]     | in  | AIBy+1  | RX[n-2]    | in  |
|         | ...         |     |         | ...        |     |
| AIB18   | fs_fwd_clkb | in  | AIB19   | fs_fwd_clk | in  |
| AIB16   | RX[9]       | in  | AIB17   | RX[8]      | in  |
| AIB14   | RX[7]       | in  | AIB15   | RX[6]      | in  |
| AIB12   | RX[5]       | in  | AIB13   | RX[4]      | in  |
| AIB10   | RX[3]       | in  | AIB11   | RX[2]      | in  |
| AIB8    | RX[1]       | in  | AIB9    | RX[0]      | in  |
| AIB6    | (empty)     | in  | AIB7    | fs_mac_rdy | in  |
| AIB4    | spare[0]    | I/O | AIB5    | spare[1]   | I/O |
| AIB2    | ns_mac_rdy  | out | AIB3    | (empty)    | out |

**Table 41. Bump-Table Exemplar (AIB Base, All-RX)**  
 $y=n+8$

| Bump ID | Bump Name       | IO  | Bump ID | Bump Name       | IO  |
|---------|-----------------|-----|---------|-----------------|-----|
| AlBy    | RX[n-1]         | in  | AlBy+1  | RX[n-2]         | in  |
|         | ...             |     |         | ...             |     |
| AlBx+44 | RX[11]          | in  | AlBx+45 | RX[10]          | in  |
| AlBx+42 | fs_fwd_clkb     | in  | AlBx+43 | fs_fwd_clk      | in  |
| AlBx+40 | RX[9]           | in  | AlBx+41 | RX[8]           | in  |
| AlBx+38 | RX[7]           | in  | AlBx+39 | RX[6]           | in  |
| AlBx+36 | RX[5]           | in  | AlBx+37 | RX[4]           | in  |
| AlBx+34 | RX[3]           | in  | AlBx+35 | RX[2]           | in  |
| AlBx+32 | RX[1]           | in  | AlBx+33 | RX[0]           | in  |
| AlBx+30 | fs_rcv_clkb     | in  | AlBx+31 | fs_rcv_clk      | in  |
| AlBx+28 | fs_sr_clkb      | in  | AlBx+29 | fs_sr_clk       | in  |
| AlBx+26 | fs_sr_load      | in  | AlBx+27 | fs_sr_data      | in  |
| AlBx+24 | fs_adapter_rstn | in  | AlBx+25 | fs_mac_rdy      | in  |
| AlBx+22 | spare[0]        | I/O | AlBx+23 | spare[1]        | I/O |
| AlBx+20 | ns_mac_rdy      | out | AlBx+21 | ns_adapter_rstn | out |
| AlBx+18 | ns_sr_data      | out | AlBx+19 | ns_sr_load      | out |
| AlBx+16 | ns_sr_clk       | out | AlBx+17 | ns_sr_clkb      | out |
| AlBx+14 | ns_rcv_clk      | out | AlBx+15 | ns_rcv_clkb     | out |
| AlBx+12 | TX[0]           | out | AlBx+13 | TX[1]           | out |
| AlBx+10 | TX[2]           | out | AlBx+11 | TX[3]           | out |
| AlBx+8  | TX[4]           | out | AlBx+9  | TX[5]           | out |
| AlBx+6  | TX[6]           | out | AlBx+7  | TX[7]           | out |
| AlBx+4  | TX[8]           | out | AlBx+5  | TX[9]           | out |
| AlBx+2  | ns_fwd_clk      | out | AlBx+3  | ns_fwd_clkb     | out |
| AlBx    | TX[10]          | out | AlBx+1  | TX[11]          | out |
|         | ...             |     |         | ...             |     |
| AlB0    | TX[n-2]         | out | AlB1    | TX[n-1]         | out |

**Table 42. Bump-Table Exemplar (AIB Plus, Balanced)**  
 $x=n-12$ ;  $y=2n+20$

| Bump ID | Bump Name       | IO  | Bump ID | Bump Name       | IO  |
|---------|-----------------|-----|---------|-----------------|-----|
| AIBx+30 | fs_rcv_clkb     | in  | AIBx+31 | fs_rcv_clk      | in  |
| AIBx+28 | fs_sr_clkb      | in  | AIBx+29 | fs_sr_clk       | in  |
| AIBx+26 | fs_sr_load      | in  | AIBx+27 | fs_sr_data      | in  |
| AIBx+24 | fs_adapter_rstn | in  | AIBx+25 | fs_mac_rdy      | in  |
| AIBx+22 | spare[0]        | I/O | AIBx+23 | spare[1]        | I/O |
| AIBx+20 | ns_mac_rdy      | out | AIBx+21 | ns_adapter_rstn | out |
| AIBx+18 | ns_sr_data      | out | AIBx+19 | ns_sr_load      | out |
| AIBx+16 | ns_sr_clk       | out | AIBx+17 | ns_sr_clkb      | out |
| AIBx+14 | (empty)         | out | AIBx+15 | (empty)         | out |
| AIBx+12 | TX[0]           | out | AIBx+13 | TX[1]           | out |
| AIBx+10 | TX[2]           | out | AIBx+11 | TX[3]           | out |
| AIBx+8  | TX[4]           | out | AIBx+9  | TX[5]           | out |
| AIBx+6  | TX[6]           | out | AIBx+7  | TX[7]           | out |
| AIBx+4  | TX[8]           | out | AIBx+5  | TX[9]           | out |
| AIBx+2  | ns_fwd_clk      | out | AIBx+3  | ns_fwd_clkb     | out |
| AIBx    | TX[10]          | out | AIBx+1  | TX[11]          | out |
|         | ...             |     |         | ...             |     |
| AIB0    | TX[n-2]         | out | AIB1    | TX[n-1]         | out |

**Table 43. Bump-Table Exemplar (AIB Plus, AII-TX)**  
*x=n-12*

| Bump ID | Bump Name       | IO  | Bump ID | Bump Name       | IO  |
|---------|-----------------|-----|---------|-----------------|-----|
| AIBy    | RX[n-1]         | in  | AIBy+1  | RX[n-2]         | in  |
|         | ...             |     |         | ...             |     |
| AIB32   | RX[13]          | in  | AIB33   | RX[12]          | in  |
| AIB30   | RX[11]          | in  | AIB31   | RX[10]          | in  |
| AIB28   | fs_fwd_clkb     | in  | AIB29   | fs_fwd_clk      | in  |
| AIB26   | RX[9]           | in  | AIB27   | RX[8]           | in  |
| AIB24   | RX[7]           | in  | AIB25   | RX[6]           | in  |
| AIB22   | RX[5]           | in  | AIB23   | RX[4]           | in  |
| AIB20   | RX[3]           | in  | AIB21   | RX[2]           | in  |
| AIB18   | RX[1]           | in  | AIB19   | RX[0]           | in  |
| AIB16   | (empty)         | in  | AIB17   | (empty)         | in  |
| AIB14   | fs_sr_clkb      | in  | AIB15   | fs_sr_clk       | in  |
| AIB12   | fs_sr_load      | in  | AIB13   | fs_sr_data      | in  |
| AIB10   | fs_adapter_rstn | in  | AIB11   | fs_mac_rdy      | in  |
| AIB8    | spare[0]        | I/O | AIB9    | spare[1]        | I/O |
| AIB6    | ns_mac_rdy      | out | AIB7    | ns_adapter_rstn | out |
| AIB4    | ns_sr_data      | out | AIB5    | ns_sr_load      | out |
| AIB2    | ns_sr_clk       | out | AIB3    | ns_sr_clkb      | out |
| AIB0    | ns_rcv_clk      | out | AIB1    | ns_rcv_clkb     | out |

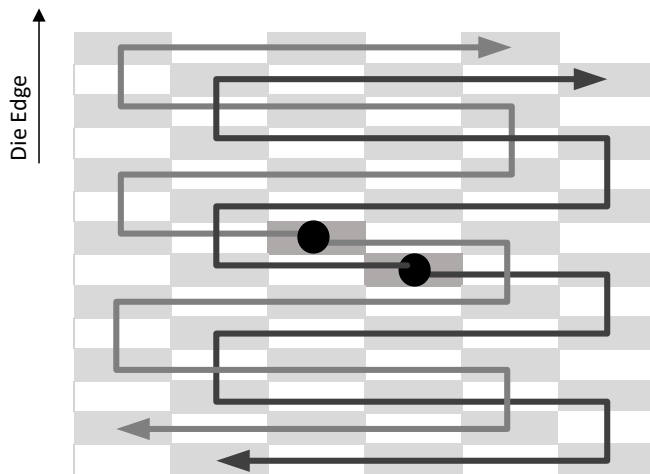


**Table 44. Bump-Table Exemplar (AIB Plus, All-RX)**  
 $y=n+18$

### 6.3.1.3 Bump Map

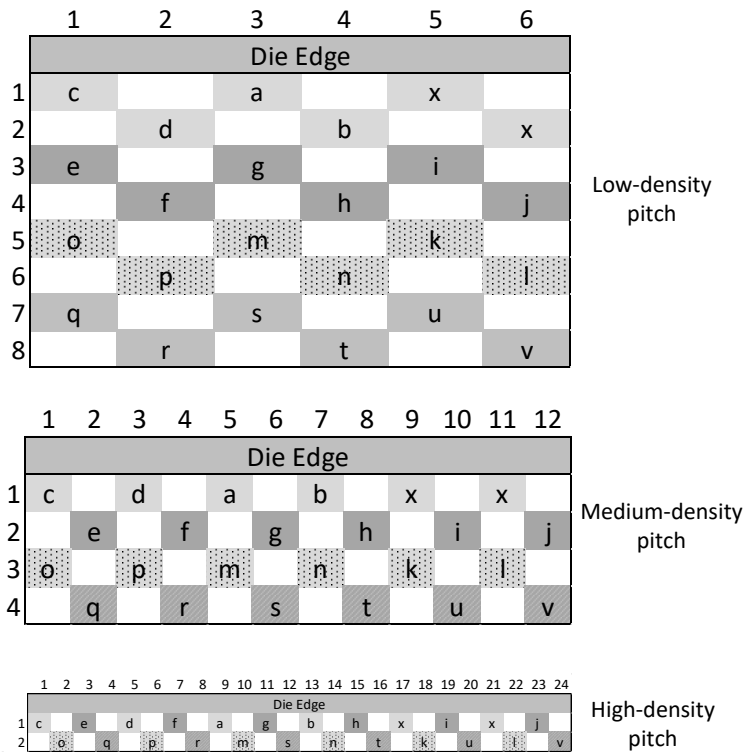
Create a *bump map* for low-density bump array as follows:

1. Allocate six columns and enough rows to accommodate all signals. “Row” and “Column” apply to a bump array with the side of the chiplet on the top.
2. Bumps are in a staggered array. Every other row starting with the first row is used for odd-numbered bumps. Every other row starting with the second row is used for even-numbered bumps.
3. If Balanced then:
  - 3.1. The middle two bumps (middle two rows, middle two columns) receive the *spare* signal bumps (odd and even).
4. If All-TX then
  - 4.1. If AIB Base then
    - 4.1.1. The two bumps in middle two columns, the row for the *spare* signals are calculated as follows, where  $n$  is the bump ID of the odd *spare* signal in the bump table:
      - 4.1.1.1.  $(n-1) \bmod 3 + 1$  for the odd-spare row, middle columns
      - 4.1.1.2. One row down for the even-spare row, middle columns
    - 4.2. If AIB Plus then
      - 4.2.1.1.  $(n-1) \bmod 3 + 1$  for the odd-spare row, middle columns
      - 4.2.1.2. One row down for the even-spare row, middle columns
5. If All-RX then:
  - 5.1. If AIB Base then:
    - 5.1.1. The middle columns, 3<sup>rd</sup> and 4<sup>th</sup> rows, receive the *spare* signal bumps (odd and even).
  - 5.2. If AIB Plus then:
    - 5.2.1. The middle columns, 5<sup>th</sup> and 6<sup>th</sup> rows, receive the *spare* signal bumps (odd and even).
6. For odd-numbered bumps, move first to the left from the *spare* bump and assign odd-numbered bumps in descending order until the left-most column is reached. From there, move up two rows and continue, this time to the right. When the rightmost column is reached, move up again two more rows and proceed back to the left. Continue in this winding fashion (boustrophedon) until all odd-numbered bumps have been assigned (figure). Any remaining bumps in the final row remain unassigned.
7. Moving back to the odd-numbered *spare* bump, move to the right and assign odd-numbered bumps in ascending order until the rightmost column is reached. From there, move down two rows and continue, this time to the left. Continue the boustrophedon movement until all remaining odd-numbered bumps have been assigned (figure), leaving any remaining bumps in the last row unassigned.
8. Repeat this process for even-numbered bumps starting from the even *spare* bump and using the even-bump rows (figure).
9. Repeat for all channels in the AIB interface.



**Figure 58. Bump Map Assignment Order**

The bump map for medium-density pitch shall be created by combining two adjacent low-density rows into a single row. As shown in Figure 59, rows 5 and 6 in the low-density map are combined to form row 3 in the medium-density map. The bump map for high-density pitch shall be created by combining two adjacent medium-density rows into a single row as shown in Figure 59.



**Figure 59. Bump Map Migration Between Different Bump Densities**

The IO physical placement shall not necessary be the same as the bump placement as a result of optimization of the die area, IO performance and power network. IO physical placement is not part of the AIB specification.

### 6.3.2 AIB AUX Signal Assignment

**AIB AUX Block using Microbumps.** The signals from the AIB AUX block (*power\_on\_reset*, *device\_detect*) shall be placed below channel 0 in a column, if the AUX block uses microbumps. Otherwise the signals may be placed in any suitable location. Leader chiplets shall have output bumps for *device\_detect* signals and input bumps for *power\_on\_reset*. Follower chiplets shall have input bumps for *device\_detect* signals and output bumps for *power\_on\_reset*. If the AUX block uses microbumps, there shall be two bumps per signal for passive redundancy (Section 3.3.2). The AUX bump table and map are shown below.

| Leader chiplet |                |     | Follower chiplet |                |     |
|----------------|----------------|-----|------------------|----------------|-----|
| Bump ID        | Bump Name      | IO  | Bump ID          | Bump Name      | IO  |
| AIBX3          | device_detect  | out | AIBX3            | device_detect  | in  |
| AIBX2          | device_detect  | out | AIBX2            | device_detect  | in  |
| AIBX1          | power_on_reset | in  | AIBX1            | power_on_reset | out |
| AIBX0          | power_on_reset | in  | AIBX0            | power_on_reset | out |

**Table 45. AIB AUX Signal Bump Table**

### 6.3.3 Example Bump Tables

| Bump ID | Bump Name   | IO  | Bump ID | Bump Name   | IO  |
|---------|-------------|-----|---------|-------------|-----|
| AIB48   | RX[19]      | in  | AIB49   | RX[18]      | in  |
| AIB46   | RX[17]      | in  | AIB47   | RX[16]      | in  |
| AIB44   | RX[15]      | in  | AIB45   | RX[14]      | in  |
| AIB42   | RX[13]      | in  | AIB43   | RX[12]      | in  |
| AIB40   | RX[11]      | in  | AIB41   | RX[10]      | in  |
| AIB38   | fs_fwd_clkb | in  | AIB39   | fs_fwd_clk  | in  |
| AIB36   | RX[9]       | in  | AIB37   | RX[8]       | in  |
| AIB34   | RX[7]       | in  | AIB35   | RX[6]       | in  |
| AIB32   | RX[5]       | in  | AIB33   | RX[4]       | in  |
| AIB30   | RX[3]       | in  | AIB31   | RX[2]       | in  |
| AIB28   | RX[1]       | in  | AIB29   | RX[0]       | in  |
| AIB26   | (empty)     | in  | AIB27   | fs_mac_rdy  | in  |
| AIB24   | spare[0]    | I/O | AIB25   | spare[1]    | I/O |
| AIB22   | ns_mac_rdy  | out | AIB23   | (empty)     | out |
| AIB20   | TX[0]       | out | AIB21   | TX[1]       | out |
| AIB18   | TX[2]       | out | AIB19   | TX[3]       | out |
| AIB16   | TX[4]       | out | AIB17   | TX[5]       | out |
| AIB14   | TX[6]       | out | AIB15   | TX[7]       | out |
| AIB12   | TX[8]       | out | AIB13   | TX[9]       | out |
| AIB10   | ns_fwd_clk  | out | AIB11   | ns_fwd_clkb | out |
| AIB8    | TX[10]      | out | AIB9    | TX[11]      | out |
| AIB6    | TX[12]      | out | AIB7    | TX[13]      | out |
| AIB4    | TX[14]      | out | AIB5    | TX[15]      | out |
| AIB2    | TX[16]      | out | AIB3    | TX[17]      | out |
| AIB0    | TX[18]      | out | AIB1    | TX[19]      | out |

**Table 46. Example Bump Table  
(AIB Base, 40 I/Os, Balanced)**

| Bump ID | Bump Name   | IO  | Bump ID | Bump Name   | IO  |
|---------|-------------|-----|---------|-------------|-----|
| AIB28   | RX[19]      | in  | AIB29   | RX[18]      | in  |
| AIB26   | RX[17]      | in  | AIB27   | RX[16]      | in  |
| AIB24   | RX[15]      | in  | AIB25   | RX[14]      | in  |
| AIB22   | RX[13]      | in  | AIB23   | RX[12]      | in  |
| AIB20   | RX[11]      | in  | AIB21   | RX[10]      | in  |
| AIB18   | fs_fwd_clkb | in  | AIB19   | fs_fwd_clk  | in  |
| AIB16   | RX[9]       | in  | AIB17   | RX[8]       | in  |
| AIB14   | RX[7]       | in  | AIB15   | RX[6]       | in  |
| AIB12   | RX[5]       | in  | AIB13   | RX[4]       | in  |
| AIB10   | RX[3]       | in  | AIB11   | RX[2]       | in  |
| AIB8    | RX[1]       | in  | AIB9    | RX[0]       | in  |
| AIB6    | (empty)     | in  | AIB7    | fs_mac_rdy  | in  |
| AIB4    | spare[0]    | I/O | AIB5    | spare[1]    | I/O |
| AIB2    | ns_mac_rdy  | out | AIB3    | (empty)     | out |
| AIB0    | ns_fwd_clk  | out | AIB1    | ns_fwd_clkb | out |

**Table 47. Example Bump Table  
(AIB Base, 20 RX)**

| Bump ID | Bump Name  | IO  | Bump ID | Bump Name   | IO  |
|---------|------------|-----|---------|-------------|-----|
| AIB26   | (empty)    | in  | AIB27   | fs_mac_rdy  | in  |
| AIB24   | spare[0]   | I/O | AIB25   | spare[1]    | I/O |
| AIB22   | ns_mac_rdy | out | AIB23   | (empty)     | out |
| AIB20   | TX[0]      | out | AIB21   | TX[1]       | out |
| AIB18   | TX[2]      | out | AIB19   | TX[3]       | out |
| AIB16   | TX[4]      | out | AIB17   | TX[5]       | out |
| AIB14   | TX[6]      | out | AIB15   | TX[7]       | out |
| AIB12   | TX[8]      | out | AIB13   | TX[9]       | out |
| AIB10   | ns_fwd_clk | out | AIB11   | ns_fwd_clkb | out |
| AIB8    | TX[10]     | out | AIB9    | TX[11]      | out |
| AIB6    | TX[12]     | out | AIB7    | TX[13]      | out |
| AIB4    | TX[14]     | out | AIB5    | TX[15]      | out |
| AIB2    | TX[16]     | out | AIB3    | TX[17]      | out |
| AIB0    | TX[18]     | out | AIB1    | TX[19]      | out |

**Table 48. Example Bump Table  
(AIB Base, 20 TX)**

| Bump ID | Bump Name       | IO  | Bump ID | Bump Name       | IO  |
|---------|-----------------|-----|---------|-----------------|-----|
| AIB60   | RX[19]          | in  | AIB61   | RX[18]          | in  |
| AIB58   | RX[17]          | in  | AIB59   | RX[16]          | in  |
| AIB56   | RX[15]          | in  | AIB57   | RX[14]          | in  |
| AIB54   | RX[13]          | in  | AIB55   | RX[12]          | in  |
| AIB52   | RX[11]          | in  | AIB53   | RX[10]          | in  |
| AIB50   | fs_fwd_clkb     | in  | AIB51   | fs_fwd_clk      | in  |
| AIB48   | RX[9]           | in  | AIB49   | RX[8]           | in  |
| AIB46   | RX[7]           | in  | AIB47   | RX[6]           | in  |
| AIB44   | RX[5]           | in  | AIB45   | RX[4]           | in  |
| AIB42   | RX[3]           | in  | AIB43   | RX[2]           | in  |
| AIB40   | RX[1]           | in  | AIB41   | RX[0]           | in  |
| AIB38   | fs_rcv_clkb     | in  | AIB39   | fs_rcv_clk      | in  |
| AIB36   | fs_sr_clkb      | in  | AIB37   | fs_sr_clk       | in  |
| AIB34   | fs_sr_load      | in  | AIB35   | fs_sr_data      | in  |
| AIB32   | fs_adapter_rstn | in  | AIB33   | fs_mac_rdy      | in  |
| AIB30   | spare[0]        | I/O | AIB31   | spare[1]        | I/O |
| AIB28   | ns_mac_rdy      | out | AIB29   | ns_adapter_rstn | out |
| AIB26   | ns_sr_data      | out | AIB27   | ns_sr_load      | out |
| AIB24   | ns_sr_clk       | out | AIB25   | ns_sr_clkb      | out |
| AIB22   | ns_rcv_clk      | out | AIB23   | ns_rcv_clkb     | out |
| AIB20   | TX[0]           | out | AIB21   | TX[1]           | out |
| AIB18   | TX[2]           | out | AIB19   | TX[3]           | out |
| AIB16   | TX[4]           | out | AIB17   | TX[5]           | out |
| AIB14   | TX[6]           | out | AIB15   | TX[7]           | out |
| AIB12   | TX[8]           | out | AIB13   | TX[9]           | out |
| AIB10   | ns_fwd_clk      | out | AIB11   | ns_fwd_clkb     | out |
| AIB8    | TX[10]          | out | AIB9    | TX[11]          | out |
| AIB6    | TX[12]          | out | AIB7    | TX[13]          | out |
| AIB4    | TX[14]          | out | AIB5    | TX[15]          | out |
| AIB2    | TX[16]          | out | AIB3    | TX[17]          | out |
| AIB0    | TX[18]          | out | AIB1    | TX[19]          | out |

**Table 49. Example Bump Table  
(AIB Plus, 40 I/Os, Balanced)**

| Bump ID | Bump Name       | IO  | Bump ID | Bump Name       | IO  |
|---------|-----------------|-----|---------|-----------------|-----|
| AIB100  | RX[39]          | in  | AIB101  | RX[38]          | in  |
| AIB98   | RX[37]          | in  | AIB99   | RX[36]          | in  |
| AIB96   | RX[35]          | in  | AIB97   | RX[34]          | in  |
| AIB94   | RX[33]          | in  | AIB95   | RX[32]          | in  |
| AIB92   | RX[31]          | in  | AIB93   | RX[30]          | in  |
| AIB90   | RX[29]          | in  | AIB91   | RX[28]          | in  |
| AIB88   | RX[27]          | in  | AIB89   | RX[26]          | in  |
| AIB86   | RX[25]          | in  | AIB87   | RX[24]          | in  |
| AIB84   | RX[23]          | in  | AIB85   | RX[22]          | in  |
| AIB82   | RX[21]          | in  | AIB83   | RX[20]          | in  |
| AIB80   | RX[19]          | in  | AIB81   | RX[18]          | in  |
| AIB78   | RX[17]          | in  | AIB79   | RX[16]          | in  |
| AIB76   | RX[15]          | in  | AIB77   | RX[14]          | in  |
| AIB74   | RX[13]          | in  | AIB75   | RX[12]          | in  |
| AIB72   | RX[11]          | in  | AIB73   | RX[10]          | in  |
| AIB70   | fs_fwd_clkb     | in  | AIB71   | fs_fwd_clk      | in  |
| AIB68   | RX[9]           | in  | AIB69   | RX[8]           | in  |
| AIB66   | RX[7]           | in  | AIB67   | RX[6]           | in  |
| AIB64   | RX[5]           | in  | AIB65   | RX[4]           | in  |
| AIB62   | RX[3]           | in  | AIB63   | RX[2]           | in  |
| AIB60   | RX[1]           | in  | AIB61   | RX[0]           | in  |
| AIB58   | fs_rcv_clkb     | in  | AIB59   | fs_rcv_clk      | in  |
| AIB56   | fs_sr_clkb      | in  | AIB57   | fs_sr_clk       | in  |
| AIB54   | fs_sr_load      | in  | AIB55   | fs_sr_data      | in  |
| AIB52   | fs_adapter_rstn | in  | AIB53   | fs_mac_rdy      | in  |
| AIB50   | spare[0]        | I/O | AIB51   | spare[1]        | I/O |
| AIB48   | ns_mac_rdy      | out | AIB49   | ns_adapter_rstn | out |
| AIB46   | ns_sr_data      | out | AIB47   | ns_sr_load      | out |
| AIB44   | ns_sr_clk       | out | AIB45   | ns_sr_clkb      | out |
| AIB42   | ns_rcv_clk      | out | AIB43   | ns_rcv_clkb     | out |
| AIB40   | TX[0]           | out | AIB41   | TX[1]           | out |
| AIB38   | TX[2]           | out | AIB39   | TX[3]           | out |
| AIB36   | TX[4]           | out | AIB37   | TX[5]           | out |
| AIB34   | TX[6]           | out | AIB35   | TX[7]           | out |
| AIB32   | TX[8]           | out | AIB33   | TX[9]           | out |
| AIB30   | ns_fwd_clk      | out | AIB31   | ns_fwd_clkb     | out |
| AIB28   | TX[10]          | out | AIB29   | TX[11]          | out |
| AIB26   | TX[12]          | out | AIB27   | TX[13]          | out |
| AIB24   | TX[14]          | out | AIB25   | TX[15]          | out |
| AIB22   | TX[16]          | out | AIB23   | TX[17]          | out |
| AIB20   | TX[18]          | out | AIB21   | TX[19]          | out |
| AIB18   | TX[20]          | out | AIB19   | TX[21]          | out |
| AIB16   | TX[22]          | out | AIB17   | TX[23]          | out |
| AIB14   | TX[24]          | out | AIB15   | TX[25]          | out |
| AIB12   | TX[26]          | out | AIB13   | TX[27]          | out |
| AIB10   | TX[28]          | out | AIB11   | TX[29]          | out |
| AIB8    | TX[30]          | out | AIB9    | TX[31]          | out |
| AIB6    | TX[32]          | out | AIB7    | TX[33]          | out |
| AIB4    | TX[34]          | out | AIB5    | TX[35]          | out |
| AIB2    | TX[36]          | out | AIB3    | TX[37]          | out |
| AIB0    | TX[38]          | out | AIB1    | TX[39]          | out |

**Table 50. Example Bump Table  
(AIB Plus, 80 I/Os, Balanced)**

| Bump ID | Bump Name       | IO  | Bump ID | Bump Name       | IO  |
|---------|-----------------|-----|---------|-----------------|-----|
| AIB38   | fs_rcv_clkb     | in  | AIB39   | fs_rcv_clk      | in  |
| AIB36   | fs_sr_clkb      | in  | AIB37   | fs_sr_clk       | in  |
| AIB34   | fs_sr_load      | in  | AIB35   | fs_sr_data      | in  |
| AIB32   | fs_adapter_rstn | in  | AIB33   | fs_mac_rdy      | in  |
| AIB30   | spare[0]        | I/O | AIB31   | spare[1]        | I/O |
| AIB28   | ns_mac_rdy      | out | AIB29   | ns_adapter_rstn | out |
| AIB26   | ns_sr_data      | out | AIB27   | ns_sr_load      | out |
| AIB24   | ns_sr_clk       | out | AIB25   | ns_sr_clkb      | out |
| AIB22   | (empty)         | out | AIB23   | (empty)         | out |
| AIB20   | TX[0]           | out | AIB21   | TX[1]           | out |
| AIB18   | TX[2]           | out | AIB19   | TX[3]           | out |
| AIB16   | TX[4]           | out | AIB17   | TX[5]           | out |
| AIB14   | TX[6]           | out | AIB15   | TX[7]           | out |
| AIB12   | TX[8]           | out | AIB13   | TX[9]           | out |
| AIB10   | ns_fwd_clk      | out | AIB11   | ns_fwd_clkb     | out |
| AIB8    | TX[10]          | out | AIB9    | TX[11]          | out |
| AIB6    | TX[12]          | out | AIB7    | TX[13]          | out |
| AIB4    | TX[14]          | out | AIB5    | TX[15]          | out |
| AIB2    | TX[16]          | out | AIB3    | TX[17]          | out |
| AIB0    | TX[18]          | out | AIB1    | TX[19]          | out |

**Table 51. Example Bump Table (AIB Plus, 20 TX)**

| Bump ID | Bump Name       | IO  | Bump ID | Bump Name       | IO  |
|---------|-----------------|-----|---------|-----------------|-----|
| AIB38   | RX[19]          | in  | AIB39   | RX[18]          | in  |
| AIB36   | RX[17]          | in  | AIB37   | RX[16]          | in  |
| AIB34   | RX[15]          | in  | AIB35   | RX[14]          | in  |
| AIB32   | RX[13]          | in  | AIB33   | RX[12]          | in  |
| AIB30   | RX[11]          | in  | AIB31   | RX[10]          | in  |
| AIB28   | fs_fwd_clkb     | in  | AIB29   | fs_fwd_clk      | in  |
| AIB26   | RX[9]           | in  | AIB27   | RX[8]           | in  |
| AIB24   | RX[7]           | in  | AIB25   | RX[6]           | in  |
| AIB22   | RX[5]           | in  | AIB23   | RX[4]           | in  |
| AIB20   | RX[3]           | in  | AIB21   | RX[2]           | in  |
| AIB18   | RX[1]           | in  | AIB19   | RX[0]           | in  |
| AIB16   | (empty)         | in  | AIB17   | (empty)         | in  |
| AIB14   | fs_sr_clkb      | in  | AIB15   | fs_sr_clk       | in  |
| AIB12   | fs_sr_load      | in  | AIB13   | fs_sr_data      | in  |
| AIB10   | fs_adapter_rstn | in  | AIB11   | fs_mac_rdy      | in  |
| AIB8    | spare[0]        | I/O | AIB9    | spare[1]        | I/O |
| AIB6    | ns_mac_rdy      | out | AIB7    | ns_adapter_rstn | out |
| AIB4    | ns_sr_data      | out | AIB5    | ns_sr_load      | out |
| AIB2    | ns_sr_clk       | out | AIB3    | ns_sr_clkb      | out |
| AIB0    | ns_rcv_clk      | out | AIB1    | ns_rcv_clkb     | out |

**Table 52. Example Bump Table (AIB Plus, 20 RX)**

### 6.3.4 Example Bump Maps

The following example bump maps represent the bump tables above as assigned to bumps. These examples include an AUX block using microbumps, in which case the



AUX block shall be placed next to Channel 0. If the AUX block does not use microbumps then the AUX assignments in these examples may be ignored.

Power microbumps are subject to chiplet requirements as established for different silicon processes, data rates, and configurations. Chiplets may implement a power-distribution network using C4 (core) bumps or microbumps to meet the power requirements. Power microbumps may be inserted between rows of I/O microbumps, in increments of six microbumps, in order to maintain signal order.

|   | 1               | 2      | 3     | 4     | 5      | 6      | 7     |
|---|-----------------|--------|-------|-------|--------|--------|-------|
|   | Edge of Chiplet |        |       |       |        |        | AUX   |
| A | Unused          |        | AIB1  |       | AIB3   |        | AIBX0 |
| B |                 | Unused |       | AIB0  |        | AIB2   |       |
| C | AIB9            |        | AIB7  |       | AIB5   |        | AIBX1 |
| D |                 | AIB8   |       | AIB6  |        | AIB4   |       |
| E | AIB11           |        | AIB13 |       | AIB15  |        | AIBX2 |
| F |                 | AIB10  |       | AIB12 |        | AIB14  |       |
| G | AIB21           |        | AIB19 |       | AIB17  |        | AIBX3 |
| H |                 | AIB20  |       | AIB18 |        | AIB16  |       |
| I | AIB23           |        | AIB25 |       | AIB27  |        |       |
| J |                 | AIB22  |       | AIB24 |        | AIB26  |       |
| K | AIB33           |        | AIB31 |       | AIB29  |        |       |
| L |                 | AIB32  |       | AIB30 |        | AIB28  |       |
| M | AIB35           |        | AIB37 |       | AIB39  |        |       |
| N |                 | AIB34  |       | AIB36 |        | AIB38  |       |
| O | AIB45           |        | AIB43 |       | AIB41  |        |       |
| P |                 | AIB44  |       | AIB42 |        | AIB40  |       |
| Q | AIB47           |        | AIB49 |       | Unused |        |       |
| R |                 | AIB46  |       | AIB48 |        | Unused |       |

**Figure 60. Low-Density Bump Map  
(AIB Base, 40 I/Os, Balanced)**

|   | 1               | 2     | 3      | 4      | 5      | 6      | 7     |
|---|-----------------|-------|--------|--------|--------|--------|-------|
|   | Edge of Chiplet |       |        |        |        |        | AUX   |
| A | AIB1            |       | Unused |        | Unused |        | AIBX0 |
| B |                 | AIB0  |        | Unused |        | Unused |       |
| C | AIB3            |       | AIB5   |        | AIB7   |        | AIBX1 |
| D |                 | AIB2  |        | AIB4   |        | AIB6   |       |
| E | AIB13           |       | AIB11  |        | AIB9   |        | AIBX2 |
| F |                 | AIB12 |        | AIB10  |        | AIB8   |       |
| G | AIB15           |       | AIB17  |        | AIB19  |        | AIBX3 |
| H |                 | AIB14 |        | AIB16  |        | AIB18  |       |
| I | AIB25           |       | AIB23  |        | AIB21  |        |       |
| J |                 | AIB24 |        | AIB22  |        | AIB20  |       |
| K | AIB27           |       | AIB29  |        | Unused |        |       |
| L |                 | AIB26 |        | AIB28  |        | Unused |       |

**Figure 61. Low-Density Bump Map  
(AIB Base, 20 TX)**

|   | 1               | 2      | 3      | 4      | 5     | 6     | 7     |
|---|-----------------|--------|--------|--------|-------|-------|-------|
|   | Edge of Chiplet |        |        |        |       |       | AUX   |
| A | Unused          |        | AIB1   |        | AIB3  |       | AIBX0 |
| B |                 | Unused |        | AIB0   |       | AIB2  |       |
| C | AIB9            |        | AIB7   |        | AIB5  |       | AIBX1 |
| D |                 | AIB8   |        | AIB6   |       | AIB4  |       |
| E | AIB11           |        | AIB13  |        | AIB15 |       | AIBX2 |
| F |                 | AIB10  |        | AIB12  |       | AIB14 |       |
| G | AIB21           |        | AIB19  |        | AIB17 |       | AIBX3 |
| H |                 | AIB20  |        | AIB18  |       | AIB16 |       |
| I | AIB23           |        | AIB25  |        | AIB27 |       |       |
| J |                 | AIB22  |        | AIB24  |       | AIB26 |       |
| K | Unused          |        | Unused |        | AIB29 |       |       |
| L |                 | Unused |        | Unused |       | AIB28 |       |

**Figure 62. Low-Density Bump Map  
(AIB Base, 20 RX)**

|   | 1               | 2      | 3     | 4     | 5      | 6      | 7     |
|---|-----------------|--------|-------|-------|--------|--------|-------|
|   | Edge of Chiplet |        |       |       |        |        | AUX   |
| A | AIB3            |        | AIB1  |       | Unused |        | AIBX0 |
| B |                 | AIB2   |       | AIB0  |        | Unused |       |
| C | AIB5            |        | AIB7  |       | AIB9   |        | AIBX1 |
| D |                 | AIB4   |       | AIB6  |        | AIB8   |       |
| E | AIB15           |        | AIB13 |       | AIB11  |        | AIBX2 |
| F |                 | AIB14  |       | AIB12 |        | AIB10  |       |
| G | AIB17           |        | AIB19 |       | AIB21  |        | AIBX3 |
| H |                 | AIB16  |       | AIB18 |        | AIB20  |       |
| I | AIB27           |        | AIB25 |       | AIB23  |        |       |
| J |                 | AIB26  |       | AIB24 |        | AIB22  |       |
| K | AIB29           |        | AIB31 |       | AIB33  |        |       |
| L |                 | AIB28  |       | AIB30 |        | AIB32  |       |
| M | AIB39           |        | AIB37 |       | AIB35  |        |       |
| N |                 | AIB38  |       | AIB36 |        | AIB34  |       |
| O | AIB41           |        | AIB43 |       | AIB45  |        |       |
| P |                 | AIB40  |       | AIB42 |        | AIB44  |       |
| Q | AIB51           |        | AIB49 |       | AIB47  |        |       |
| R |                 | AIB50  |       | AIB48 |        | AIB46  |       |
| S | AIB53           |        | AIB55 |       | AIB57  |        |       |
| T |                 | AIB52  |       | AIB54 |        | AIB56  |       |
| U | Unused          |        | AIB61 |       | AIB59  |        |       |
| V |                 | Unused |       | AIB60 |        | AIB58  |       |

**Figure 63. Low-Density Bump Map  
(AIB Plus, 40 I/Os, Balanced)**

|   | 1               | 2     | 3      | 4      | 5      | 6      | 7     |
|---|-----------------|-------|--------|--------|--------|--------|-------|
|   | Edge of Chiplet |       |        |        |        |        | AUX   |
| A | AIB3            |       | AIB1   |        | Unused |        | AIBX0 |
| B |                 | AIB2  |        | AIB0   |        | Unused |       |
| C | AIB5            |       | AIB7   |        | AIB9   |        | AIBX1 |
| D |                 | AIB4  |        | AIB6   |        | AIB8   |       |
| E | AIB15           |       | AIB13  |        | AIB11  |        | AIBX2 |
| F |                 | AIB14 |        | AIB12  |        | AIB10  |       |
| G | AIB17           |       | AIB19  |        | AIB21  |        | AIBX3 |
| H |                 | AIB16 |        | AIB18  |        | AIB20  |       |
| I | AIB27           |       | AIB25  |        | AIB23  |        |       |
| J |                 | AIB26 |        | AIB24  |        | AIB22  |       |
| K | AIB29           |       | AIB31  |        | AIB33  |        |       |
| L |                 | AIB28 |        | AIB30  |        | AIB32  |       |
| M | AIB39           |       | AIB37  |        | AIB35  |        |       |
| N |                 | AIB38 |        | AIB36  |        | AIB34  |       |
| O | AIB41           |       | Unused |        | Unused |        |       |
| P |                 | AIB40 |        | Unused |        | Unused |       |

**Figure 64. Low-Density Bump Map  
(AIB Plus, 20 TX)**

|   | 1               | 2      | 3      | 4      | 5     | 6     | 7     |
|---|-----------------|--------|--------|--------|-------|-------|-------|
|   | Edge of Chiplet |        |        |        |       |       | AUX   |
| A | Unused          |        | Unused |        | AIB1  |       | AIBX0 |
| B |                 | Unused |        | Unused |       | AIB0  |       |
| C | AIB7            |        | AIB5   |        | AIB3  |       | AIBX1 |
| D |                 | AIB6   |        | AIB4   |       | AIB2  |       |
| E | AIB9            |        | AIB11  |        | AIB13 |       | AIBX2 |
| F |                 | AIB8   |        | AIB10  |       | AIB12 |       |
| G | AIB19           |        | AIB17  |        | AIB15 |       | AIBX3 |
| H |                 | AIB18  |        | AIB16  |       | AIB14 |       |
| I | AIB21           |        | AIB23  |        | AIB25 |       |       |
| J |                 | AIB20  |        | AIB22  |       | AIB24 |       |
| K | AIB31           |        | AIB29  |        | AIB27 |       |       |
| L |                 | AIB30  |        | AIB28  |       | AIB26 |       |
| M | AIB33           |        | AIB35  |        | AIB37 |       |       |
| N |                 | AIB32  |        | AIB34  |       | AIB36 |       |
| O | Unused          |        | AIB41  |        | AIB39 |       |       |
| P |                 | Unused |        | AIB40  |       | AIB38 |       |

**Figure 65. Low-Density Bump Map  
(AIB Plus, 20 RX)**

|   | 1               | 2     | 3      | 4     | 5     | 6     | 7     | 8     | 9      | 10    | 11     | 12    |
|---|-----------------|-------|--------|-------|-------|-------|-------|-------|--------|-------|--------|-------|
|   | Edge of Chiplet |       |        |       |       |       |       |       |        |       |        | AUX   |
| A | AIB3            |       | AIB2   |       | AIB1  |       | AIB0  |       | Unused |       | Unused | AIBX0 |
| B |                 | AIB5  |        | AIB4  |       | AIB7  |       | AIB6  |        | AIB9  |        | AIB8  |
| C | AIB15           |       | AIB14  |       | AIB13 |       | AIB12 |       | AIB11  |       | AIB10  | AIBX1 |
| D |                 | AIB17 |        | AIB16 |       | AIB19 |       | AIB18 |        | AIB21 |        | AIB20 |
| E | AIB27           |       | AIB26  |       | AIB25 |       | AIB24 |       | AIB23  |       | AIB22  | AIBX2 |
| F |                 | AIB29 |        | AIB28 |       | AIB31 |       | AIB30 |        | AIB33 |        | AIB32 |
| G | AIB39           |       | AIB38  |       | AIB37 |       | AIB36 |       | AIB35  |       | AIB34  | AIBX3 |
| H |                 | AIB41 |        | AIB40 |       | AIB43 |       | AIB42 |        | AIB45 |        | AIB44 |
| I | AIB51           |       | AIB50  |       | AIB49 |       | AIB48 |       | AIB47  |       | AIB46  |       |
| J |                 | AIB53 |        | AIB52 |       | AIB55 |       | AIB54 |        | AIB57 |        | AIB56 |
| K | Unused          |       | Unused |       | AIB61 |       | AIB60 |       | AIB59  |       | AIB58  |       |

**Figure 66 Medium-Density Bump Map  
(AIB Plus, 40 I/Os, Balanced)**

|   | 1               | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9      | 10    | 11     | 12    |
|---|-----------------|-------|-------|-------|-------|-------|-------|-------|--------|-------|--------|-------|
|   | Edge of Chiplet |       |       |       |       |       |       |       |        |       |        |       |
| A | AIB1            |       | AIB0  |       | AIB3  |       | AIB2  |       | AIB5   |       | AIB4   |       |
| B |                 | AIB11 |       | AIB10 |       | AIB9  |       | AIB8  |        | AIB7  |        | AIB6  |
| C | AIB13           |       | AIB12 |       | AIB15 |       | AIB14 |       | AIB17  |       | AIB16  |       |
| D |                 | AIB23 |       | AIB22 |       | AIB21 |       | AIB20 |        | AIB19 |        | AIB18 |
| E | AIB25           |       | AIB24 |       | AIB27 |       | AIB26 |       | AIB29  |       | AIB28  |       |
| F |                 | AIB35 |       | AIB34 |       | AIB33 |       | AIB32 |        | AIB31 |        | AIB30 |
| G | AIB37           |       | AIB36 |       | AIB39 |       | AIB38 |       | AIB41  |       | AIB40  |       |
| H |                 | AIB47 |       | AIB46 |       | AIB45 |       | AIB44 |        | AIB43 |        | AIB42 |
| I | AIB49           |       | AIB48 |       | AIB51 |       | AIB50 |       | AIB53  |       | AIB52  |       |
| J |                 | AIB59 |       | AIB58 |       | AIB57 |       | AIB56 |        | AIB55 |        | AIB54 |
| K | AIB61           |       | AIB60 |       | AIB63 |       | AIB62 |       | AIB65  |       | AIB64  |       |
| L |                 | AIB71 |       | AIB70 |       | AIB69 |       | AIB68 |        | AIB67 |        | AIB66 |
| M | AIB73           |       | AIB72 |       | AIB75 |       | AIB74 |       | AIB77  |       | AIB76  |       |
| N |                 | AIB83 |       | AIB82 |       | AIB81 |       | AIB80 |        | AIB79 |        | AIB78 |
| O | AIB85           |       | AIB84 |       | AIB87 |       | AIB86 |       | AIB89  |       | AIB88  |       |
| P |                 | AIB95 |       | AIB94 |       | AIB93 |       | AIB92 |        | AIB91 |        | AIB90 |
| Q | AIB97           |       | AIB96 |       | AIB99 |       | AIB98 |       | AIB101 |       | AIB100 |       |

**Figure 67 Medium-Density Bump Map  
(AIB Plus, 80 I/Os, Balanced)**

|   | 1               | 2      | 3     | 4     | 5      | 6     | 7     | 8     | 9     | 10    | 11    | 12    | 13    | 14    | 15    | 16    | 17     | 18    | 19    | 20    | 21     | 22    | 23    | 24    |
|---|-----------------|--------|-------|-------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|-------|-------|-------|--------|-------|-------|-------|
|   | Edge of Chiplet |        |       |       |        |       |       |       |       |       |       |       |       |       |       |       |        |       |       |       |        |       |       | AUX   |
| A | AIB3            |        | AIB5  |       | AIB2   |       | AIB4  |       | AIB1  |       | AIB7  |       | AIB0  |       | AIB6  |       | Unused |       | AIB9  |       | Unused |       | AIB8  | AIBX0 |
| B |                 | AIB15  |       | AIB17 |        | AIB14 |       | AIB16 |       | AIB13 |       | AIB19 |       | AIB12 |       | AIB18 |        | AIB11 |       | AIB21 |        | AIB10 |       | AIB20 |
| C | AIB27           |        | AIB29 |       | AIB26  |       | AIB28 |       | AIB25 |       | AIB31 |       | AIB24 |       | AIB30 |       | AIB23  |       | AIB33 |       | AIB22  |       | AIB32 | AIBX1 |
| D |                 | AIB39  |       | AIB41 |        | AIB38 |       | AIB40 |       | AIB37 |       | AIB43 |       | AIB36 |       | AIB42 |        | AIB35 |       | AIB45 |        | AIB34 |       | AIB44 |
| E | AIB51           |        | AIB53 |       | AIB50  |       | AIB52 |       | AIB49 |       | AIB55 |       | AIB48 |       | AIB54 |       | AIB47  |       | AIB57 |       | AIB46  |       | AIB56 | AIBX2 |
| F |                 | Unused |       |       | Unused |       |       |       | AIB61 |       |       |       | AIB60 |       |       |       | AIB59  |       |       |       | AIB58  |       |       | AIBX3 |

**Figure 68 High-Density Bump Map  
(AIB Plus, 40 I/Os, Balanced)**

## 6.4 Stacking Channels into a Column

Channel 0 shall abut the AUX block bumps. The orientation shall depend on whether the interface is placed on the East/West sides or the North/South sides of the chiplet.

| Edge of Chiplet |        |       |       |        |        |        |        |       |  | AUX   |  |
|-----------------|--------|-------|-------|--------|--------|--------|--------|-------|--|-------|--|
| AIB3            |        | AIB1  |       | Unused |        | Unused |        | AIBX0 |  | AIBX1 |  |
| AIB5            | AIB2   |       | AIB0  |        | Unused |        | Unused | AIBX0 |  | AIBX1 |  |
| AIB15           | AIB4   | AIB7  | AIB6  | AIB9   |        |        | AIB8   | AIBX1 |  | AIBX2 |  |
|                 | AIB14  |       | AIB13 | AIB12  |        |        | AIB10  |       |  |       |  |
| AIB17           |        | AIB19 |       | AIB21  |        |        |        |       |  |       |  |
|                 | AIB16  |       | AIB18 |        | AIB23  |        | AIB22  |       |  |       |  |
| AIB27           | AIB26  | AIB25 | AIB24 | AIB23  |        | AIB22  |        |       |  |       |  |
|                 | AIB28  |       | AIB30 |        | AIB29  |        | AIB28  |       |  |       |  |
| AIB39           |        | AIB37 |       | AIB35  |        | AIB34  |        |       |  |       |  |
|                 | AIB38  |       | AIB36 |        | AIB35  |        | AIB34  |       |  |       |  |
| AIB41           | AIB40  | AIB43 | AIB42 | AIB45  |        | AIB44  |        |       |  |       |  |
|                 | AIB49  |       | AIB48 |        | AIB47  |        | AIB46  |       |  |       |  |
| AIB51           | AIB50  |       | AIB48 |        | AIB47  |        | AIB46  |       |  |       |  |
|                 | AIB52  |       | AIB55 |        | AIB57  |        | AIB56  |       |  |       |  |
| AIB53           |        | AIB55 |       | AIB57  |        | AIB56  |        |       |  |       |  |
|                 | AIB52  |       | AIB54 |        | AIB59  |        | AIB58  |       |  |       |  |
| Unused          |        | AIB61 |       | AIB59  |        | AIB58  |        |       |  |       |  |
|                 | Unused |       | AIB60 |        | AIB59  |        | AIB58  |       |  |       |  |

**Figure 69. Low-Density Channel 0 and AUX: East Side**

| Edge of Chiplet |        |       |       |        |       |  |  |
|-----------------|--------|-------|-------|--------|-------|--|--|
| AUX             | AIB3   | AIB1  | AIB0  | Unused |       |  |  |
| AIBX0           |        | AIB2  | AIB7  | AIB9   | AIB8  |  |  |
| AIBX1           |        | AIB4  | AIB6  | AIB11  | AIB10 |  |  |
| AIBX2           |        | AIB14 | AIB12 | AIB21  | AIB20 |  |  |
| AIB8            |        | AIB16 | AIB18 | AIB23  | AIB22 |  |  |
|                 |        | AIB25 | AIB24 | AIB33  | AIB32 |  |  |
|                 |        | AIB26 | AIB31 | AIB30  | AIB35 |  |  |
|                 | AIB29  |       | AIB37 | AIB36  | AIB45 |  |  |
|                 |        | AIB28 | AIB43 | AIB42  | AIB47 |  |  |
|                 | AIB39  |       | AIB38 | AIB49  | AIB57 |  |  |
|                 | AIB41  |       | AIB40 | AIB55  | AIB54 |  |  |
|                 |        |       | AIB50 | AIB61  | AIB60 |  |  |
|                 | AIB51  |       |       | Unused |       |  |  |
|                 |        |       |       |        |       |  |  |
|                 | AIB53  |       |       |        |       |  |  |
|                 |        |       |       |        |       |  |  |
|                 | Unused |       |       |        |       |  |  |

Figure 70. Low-Density Channel 0 and AUX: West Side



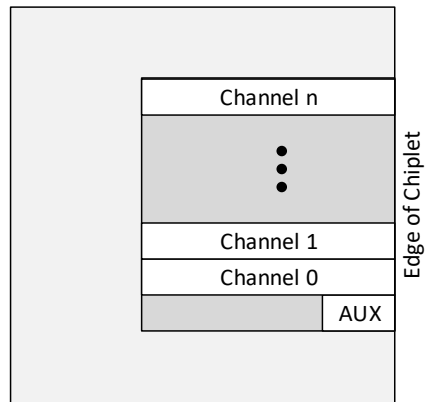
| Edge of Chiplet |        |        |       |       |        |        |
|-----------------|--------|--------|-------|-------|--------|--------|
| AUX             |        |        |       |       |        |        |
| AIBX0           | AIB3   |        | AIB1  |       | Unused |        |
|                 |        | AIB2   |       | AIB0  |        | Unused |
| AIBX1           | AIB5   |        | AIB7  |       | AIB9   |        |
|                 |        | AIB4   |       | AIB6  |        | AIB8   |
| AIBX2           | AIB15  |        | AIB13 |       | AIB11  |        |
|                 |        | AIB14  |       | AIB12 |        | AIB10  |
| AIB8            | AIB17  |        | AIB19 |       | AIB21  |        |
|                 |        | AIB16  |       | AIB18 |        | AIB20  |
|                 | AIB27  |        | AIB25 |       | AIB23  |        |
|                 |        | AIB26  |       | AIB24 |        | AIB22  |
|                 | AIB29  |        | AIB31 |       | AIB33  |        |
|                 |        | AIB28  |       | AIB30 |        | AIB32  |
|                 | AIB39  |        | AIB37 |       | AIB35  |        |
|                 |        | AIB38  |       | AIB36 |        | AIB34  |
|                 | AIB41  |        | AIB43 |       | AIB45  |        |
|                 |        | AIB40  |       | AIB42 |        | AIB44  |
|                 | AIB51  |        | AIB49 |       | AIB47  |        |
|                 |        | AIB50  |       | AIB48 |        | AIB46  |
|                 | AIB53  |        | AIB55 |       | AIB57  |        |
|                 |        | AIB52  |       | AIB54 |        | AIB56  |
|                 | Unused |        | AIB61 |       | AIB59  |        |
|                 |        | Unused |       | AIB60 |        | AIB58  |

**Figure 71. Low-Density Channel 0 and AUX: North Side**

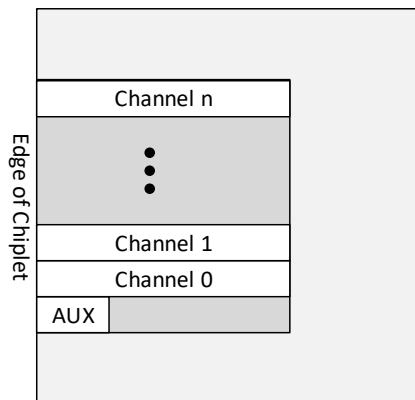
|       |          |          |              |              |        |        |
|-------|----------|----------|--------------|--------------|--------|--------|
|       | AIB58    |          | AIB60        |              | Unused |        |
|       |          | AIB59    |              | AIB61        |        | Unused |
|       | AIB56    |          | AIB54        |              | AIB52  |        |
|       |          | AIB57    |              | AIB55        |        | AIB53  |
|       | AIB46    |          | AIB48        |              | AIB50  |        |
|       |          | AIB47    |              | AIB49        |        | AIB51  |
|       | AIB44    |          | AIB42        |              | AIB40  |        |
|       |          | AIB45    |              | AIB43        |        | AIB41  |
|       | AIB34    |          | AIB36        |              | AIB38  |        |
|       |          | AIB35    |              | AIB37        |        | AIB39  |
|       | AIB32    |          | <b>AIB30</b> |              | AIB28  |        |
|       |          | AIB33    |              | <b>AIB31</b> |        | AIB29  |
|       | AIB22    |          | AIB24        |              | AIB26  |        |
|       |          | AIB23    |              | AIB25        |        | AIB27  |
|       | AIB20    |          | AIB18        |              | AIB16  |        |
| AIB8  |          | AIB21    |              | AIB19        |        | AIB17  |
|       | AIB10    |          | AIB12        |              | AIB14  |        |
| AIBX2 |          | AIB11    |              | AIB13        |        | AIB15  |
|       | AIB8     |          | AIB6         |              | AIB4   |        |
| AIBX1 |          | AIB9     |              | AIB7         |        | AIB5   |
|       | Unused d |          | AIB0         |              | AIB2   |        |
| AIBX0 |          | Unused d |              | AIB1         |        | AIB3   |
|       |          |          |              |              |        |        |

**Figure 72. Low-Density Channel 0 and AUX: South Side**

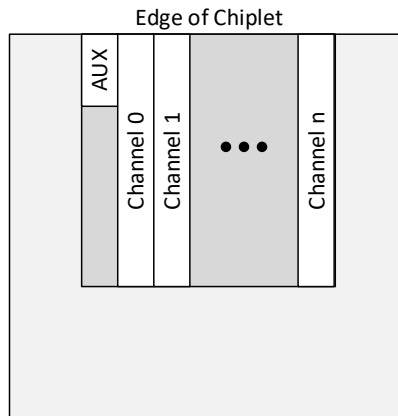
The remaining channels shall be stacked by abutting each channel against the prior channel. The order shall depend on whether the interface is placed on the East/West sides or the North/South sides of the chiplet.



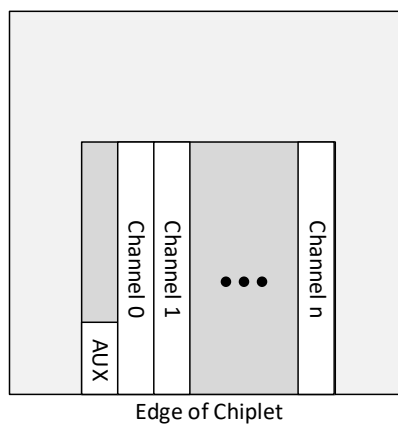
**Figure 73. Channel Stacking: East Side**



**Figure 74. Channel Stacking: West Side**



**Figure 75. Channel Stacking: North Side**



**Figure 76. Channel Stacking: South Side**

## 6.5 Channel-Number Semantics

The AIB Specification assumes no application-level semantics associated with channel numbers within a column. If an application implements channel-number semantics that impact how one interface connects to another, then it is the responsibility of the designer to resolve any channel-number conflicts.

## 6.6 Alternate Bump Maps

Alternate 40-data-signal bump tables could be used for devices intending to connect to an existing AIB interface devices. The tables and associated bump maps are shown in the appendix (Section 8.1).

If alternate implementation is to interface with an AIB Plus implementation, redundancy shall be disabled. In addition, one row of microbumps shall be inserted after each set of six channels when stacking channels into a column.

## 7 AIB On Standard Packaging (AIB-O)

AIB on Standard Packaging, or AIB-O, is an AIB 2.0 variant for applications that desire a high bandwidth parallel interface without using advanced high-density packaging technology.

AIB-O is intended for chip-to-chip interconnect within a package with typical signal distances of 5 millimeters or less. There is no maximum interconnect distance specified; instead the AIB-O electrical requirements must be met.

AIB-O shares all requirements of AIB Gen2 except as defined in this section.

### 7.1 AIB-O Interface

#### 7.1.1 AIB-O Channel

AIB-O uses a subset of the AIB channel signals:

- Data signals
  - Inputs (RX): data input signals received by the interface
  - Outputs (TX): data output signals transmitted from the interface
- Clocks
  - Data clock out (*ns\_fwd\_clk*), sent by the near side chiplet to the far side chiplet
  - Data clock in (*fs\_fwd\_clk*): received by the near side chiplet from the far side chiplet

The functions of other AIB Gen2 signals of Table 4 are performed by sideband control shift register as described in section 7.6.

An AIB-O channel shall have the signal counts as shown in [Table 53](#)~~Table 53~~.

| Bump pitch (μm) | Range of data signals per channel (Increments) |               |        |               |        |               |
|-----------------|--|---------------|--------|---------------|--------|---------------|
|                 | TX   |               |        | RX            |        |               |
|                 | Balanced                                       | All-TX        | All-RX | Balanced      | All-TX | All-RX        |
| ≤110            | 10-20<br>(10)                                  | 10-40<br>(10) | 0      | 10-20<br>(10) | 0      | 10-40<br>(10) |

**Table 53. AIB-O Number of Data Signals per Channel**

#### 7.1.2 AIB-O to MAC Interface

An AIB-O PHY has the same interface to the MAC as described in Table 7 for AIB Gen2.

### 7.1.3 AIB-O Reset and Initialization

AIB-O shall use the same reset and initialization as described in section 3, including the requirements for *CONF\_DONE*, *power\_on\_reset* and *device\_detect* signals.

The AIB-O Sideband of section 7.6 communicates the status between leader and follower for AIB-O per-channel reset, ready and channel calibration.

## 7.2 AIB-O Electrical Specification

Signals (data, clock, and active control signals) shall meet the voltage and timing specifications provided in Table 54Table 20.

- Delays are specified in terms of *unit interval*, or *UI*. This refers to the minimum interval between data changes, and it is therefore a function of the clock frequency.
- The near-end timing specifications are for a 0.63pF capacitive load.
- The “trace” timing specifications in Table 54 reference a channel that has 2dB loss at 2GHz.
- Skew parameters are measured from the center of the eye.
- Jitter margin shall include all possible jitter contributors within the chiplet, including but not limited to reference clock jitter; intrinsic jitter contributed by any phase-locked loop, clock-data recovery, delay-lock loop, or the clock network; jitter caused by power-supply noise; and jitter caused by switching noise.

| Symbol     | Parameter   | Near end                  | Trace | Far end   |
|------------|---|---------------------------|-------|-----------|
| $V_{EH}$   | Minimum difference between LO and HI (eye diagram height)                         |                           |       | +/-100 mV |
| $V_{OS}$   | Minimum output swing<br>Typical output swing<br>Maximum output swing              | 0.735 V<br>0.8 V<br>0.88V |       |           |
| $V_{CM}^1$ | Common mode voltage   | TBD                       |       |           |
| $t_{EW}$   | Minimum duration of valid output (eye diagram width) for data and forwarded clock |                           |       | 0.35 UI   |

1.  $V_{CM}$  tolerance includes all board level effects, with power supply variation (e.g. regulator) within 5%.

**Table 54. AIB-O Electrical signal specifications**

### 7.3 AIB-O Bump Configuration

The bump pattern for AIB-O is shown in Figure 77.

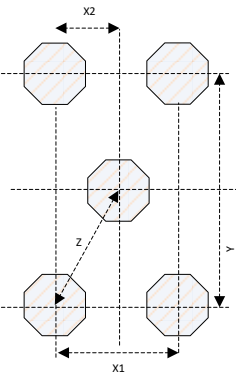


Figure 77. AIB-O Bump Spacing

The spacing of the bump array shall meet the Table 55 specifications. The X1 dimension may be reduced or expanded and may vary within the array. The Y dimension is parallel to the die edge. The AIB-O channel height shall be fixed at 312.48  $\mu\text{m}$ , the same as AIB.

| Symbol | Description                       | AIB-O   |
|--------|-----------------------------------|---|
| X1     | Aligned-row bump-to-bump pitch    | 156.24u or $\geq$ minimum bump pitch such that Z minimum is met |
| X2     | Staggered-row bump-to-bump pitch  | $X1 \div 2$ (typical)   |
| Z      | Diagonal bump-to-bump pitch       | $\geq$ minimum bump pitch                                       |
| Y      | Aligned-column bump-to-bump pitch | 156.24u   |

Table 55. AIB-O Bump Spacing Specifications



## 7.4 AIB-O Signal Bump Assignment

### 7.4.1 AIB-O Example Bump Table

| Bump ID | Bump Name   | IO  |
|---------|-------------|-----|
| A1      | RX[0]       | in  |
| A3      | RX[1]       | in  |
| B2      | RX[2]       | in  |
| B4      | RX[3]       | in  |
| C1      | RX[4]       | in  |
| C3      | GND         | n/a |
| D2      | RX[5]       | in  |
| D4      | RX[6]       | in  |
| E1      | RX[7]       | in  |
| E3      | fs_fwd_clkb | in  |
| F2      | RX[8]       | in  |
| F4      | RX[9]       | in  |
| G1      | VDD         | n/a |
| G3      | fs_fwd_clk  | in  |
| H2      | TX[9]       | out |
| H4      | VDD         | n/a |
| I1      | TX[7]       | out |
| I3      | ns_fwd_clk  | out |
| J2      | TX[6]       | out |
| J4      | TX[8]       | out |
| K1      | TX[4]       | out |
| K3      | ns_fwd_clkb | out |
| L2      | GND         | n/a |
| L4      | TX[5]       | out |
| M1      | TX[3]       | out |
| M3      | TX[2]       | out |
| N2      | TX[1]       | out |
| N4      | TX[0]       | out |

Table 56. AIB-O Example Bump Table, 20 I/Os, Balanced

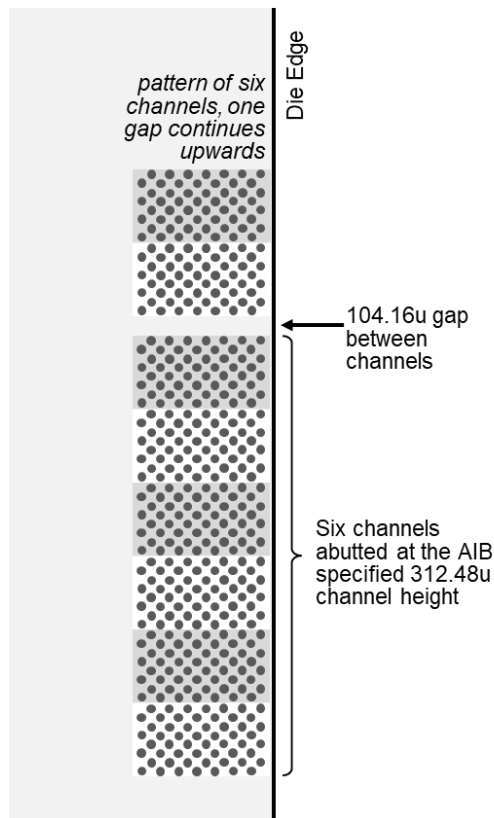
7.4.2 AIB-O Example Bump Map

|   | 1               | 2     | 3               | 4     |
|---|-----------------|-------|-----------------|-------|
|   | Edge of Chiplet |       |                 |       |
| A | RX[0]           |       | RX[1]           |       |
| B |                 | RX[2] |                 | RX[3] |
| C | RX[4]           |       | GND             |       |
| D |                 | RX[5] |                 | RX[6] |
| E | RX[7]           |       | fs_fwd_<br>clkb |       |
| F |                 | RX[8] |                 | RX[9] |
| G | VDD             |       | fs_fwd_<br>clk  |       |
| H |                 | TX[9] |                 | VDD   |
| I | TX[7]           |       | ns_fwd_<br>clk  |       |
| J |                 | TX[6] |                 | TX[8] |
| K | TX[4]           |       | ns_fwd_<br>clkb |       |
| L |                 | GND   |                 | TX[5] |
| M | TX[3]           |       | TX[2]           |       |
| N |                 | TX[1] |                 | TX[0] |

Figure 78. AIB-O Example Bump Map, 20 I/Os, Balanced

7.4.3 Stacking AIB-O Channels into a Column

AIB-O requires a 104.16u gap every six channels as shown in Figure 79. With AIB-O that gap is empty of bumps.



**Figure 79. Stacking AIB-O channels with gap every six channels**

## 7.5 Gen2 Features Not in AIB-O

Redundancy is not supported by AIB-O. As AIB-O uses standard packaging technology instead of microbumps, redundancy is unnecessary.

Data Bus Inversion (DBI) is not supported by AIB-O.

## 7.6 AIB-O Sideband Control Shift Register

Sideband communication is required to communicate channel reset, ready and calibration as described in 3.2.3. AIB-O uses one shift register in each direction for the entire AIB-O interface. The shift registers are transferred using per-interface signals:

*ns\_sr\_clk, sr\_sr\_data, ns\_sr\_load, fs\_sr\_clk, fs\_sr\_data, fs\_sr\_load.*

The signals *ns\_sr\_clk*, *sr\_sr\_data*, *ns\_sr\_load*, *fs\_sr\_clk*, *fs\_sr\_data*, *fs\_sr\_load* between an AIB-O interface pair shall have package trace lengths no longer than 10mm.

The reset, ready and calibration data for all channels in an interface are concatenated together and sent over *ns\_sr\_data* and *fs\_sr\_data* as shown in Table 57.

| Sideband signal per channel               | ns/fs_sr_data Bit Index |
|---|-------------------------|
| <i>ns_mac_rdy</i> <sup>1</sup>            | [channel index][0]      |
| <i>ns_adapter_rstn</i> <sup>1,2</sup>     | [channel index][1]      |
| <i>osc_transfer_en</i> <sup>3</sup>       | [channel index][2]      |
| <i>rx_dcc_dll_lock_req</i> <sup>2,3</sup> | [channel index][3]      |
| <i>tx_dcc_dll_lock_req</i> <sup>2,3</sup> | [channel index][4]      |
| <i>tx_dcc_cal_done</i> <sup>3</sup>       | [channel index][5]      |
| <i>rx_dll_lock</i> <sup>3</sup>           | [channel index][6]      |
| <i>rx_transfer_en</i> <sup>3</sup>        | [channel index][7]      |
| <i>tx_transfer_en</i> <sup>3</sup>        | [channel index][9]      |
| reserved                                  | [channel index][10-11]  |

1. "ns" (near side) is from the point of view of the source. The destination considers these as fs (far side) inputs.
2. By convention only the follower asserts these signals.
3. Coming from the leader, these signals have the same definition as "ms\_" + the signal name. Coming from the follower, these signals have the same definition as "sl\_" + the signal name.

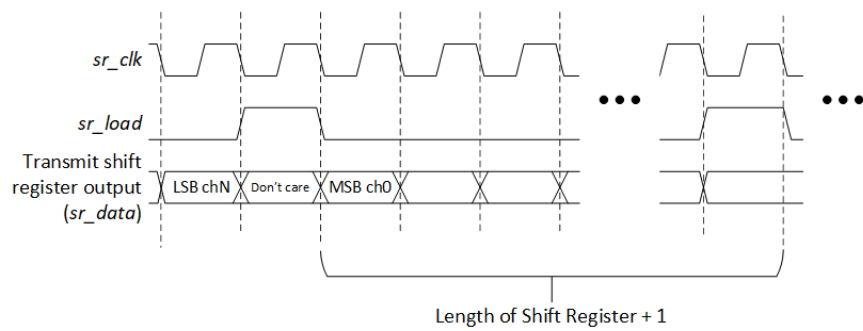
**Table 57. AIB-O Sideband Control Signals**

Data for channel index 0 is sent first, with the channel 0 MSB first. Starting with channel 0 permits AIB-O interfaces of different channel counts to interoperate using the channels in common. Any shift register data received on *fs\_sr\_data* in excess of an interface's channel count shall be ignored.

An interface that does not require calibration may fix *ns\_sr\_data*=HI.

### 7.6.1 AIB-O Sideband Control Shift Register Timing

The AIB-O *sr\_clk* signals (*ns\_sr\_clk* when sent from the near side; *fs\_sr\_clk* when received from the far side) shall be generated from the free running clock and distributed as in 2.2.3.2.1.



**Figure 80. AIB-O Sideband Control Shift Register Timing**

| Parameter                          | Min    | Max     |
|------------------------------------|--------|---------|
| AIB-O Free-Running Clock Frequency | 50 Mhz | 100 MHz |

**Table 58. AIB-O Free-Running Clock Frequency**

## 8 Appendices

### 8.1 Alternate Bump Maps

#### 8.1.1 Alternate (Leader) Bump Table

| Bump ID | Bump Name    | IO  | Bump ID | Bump Name       | IO  |
|---------|--------------|-----|---------|-----------------|-----|
| AIB61   | unused_AIB61 | tri | AIB50   | unused_AIB50    | tri |
| AIB72   | unused_AIB72 | tri | AIB73   | unused_AIB73    | tri |
| AIB75   | unused_AIB75 | tri | AIB74   | unused_AIB74    | tri |
| AIB91   | unused_AIB91 | tri | AIB90   | unused_AIB90    | tri |
| AIB95   | ns_sr_data   | out | AIB94   | ns_sr_load      | out |
| AIB85   | ns_sr_clk    | out | AIB84   | ns_sr_clkb      | out |
| AIB76   | unused_AIB76 | tri | AIB77   | unused_AIB77    | tri |
| AIB58   | unused_AIB58 | tri | AIB63   | unused_AIB63    | tri |
| AIB48   | unused_AIB48 | tri | AIB55   | unused_AIB55    | tri |
| AIB62   | unused_AIB62 | tri | AIB60   | unused_AIB60    | tri |
| AIB53   | unused_AIB53 | tri | AIB54   | unused_AIB54    | tri |
| AIB49   | ns_mac_rdy   | out | AIB56   | ns_adapter_rstn | out |
| AIB51   | unused_AIB51 | tri | AIB52   | unused_AIB52    | tri |
| AIB57   | fs_rcv_clk   | in  | AIB59   | fs_rcv_clkb     | in  |
| AIB64   | unused_AIB64 | tri | AIB65   | fs_adapter_rstn | in  |
| AIB80   | unused_AIB80 | tri | AIB81   | unused_AIB81    | tri |
| AIB78   | unused_AIB78 | tri | AIB79   | unused_AIB79    | tri |
| AIB87   | ns_rcv_clk   | out | AIB86   | ns_rcv_clkb     | out |
| AIB83   | fs_sr_clk    | in  | AIB82   | fs_sr_clkb      | in  |
| AIB89   | unused_AIB89 | tri | AIB88   | unused_AIB88    | tri |
| AIB93   | fs_sr_data   | in  | AIB92   | fs_sr_load      | in  |
| AIB71   | unused_AIB71 | tri | AIB70   | unused_AIB70    | tri |
| AIB68   | unused_AIB68 | tri | AIB69   | unused_AIB69    | tri |
| AIB66   | unused_AIB66 | tri | AIB67   | unused_AIB67    | tri |
| AIB20   | RX[0]        | in  | AIB21   | RX[1]           | in  |
| AIB22   | RX[2]        | in  | AIB23   | RX[3]           | in  |
| AIB24   | RX[4]        | in  | AIB25   | RX[5]           | in  |
| AIB26   | RX[6]        | in  | AIB27   | RX[7]           | in  |
| AIB28   | RX[8]        | in  | AIB29   | RX[9]           | in  |
| AIB43   | fs_fwd_clk   | in  | AIB42   | fs_fwd_clkb     | in  |
| AIB30   | RX[10]       | in  | AIB31   | RX[11]          | in  |
| AIB32   | RX[12]       | in  | AIB33   | RX[13]          | in  |

| Bump ID | Bump Name    | IO  | Bump ID | Bump Name    | IO  |
|---------|--------------|-----|---------|--------------|-----|
| AIB34   | RX[14]       | in  | AIB35   | RX[15]       | in  |
| AIB36   | RX[16]       | in  | AIB37   | RX[17]       | in  |
| AIB38   | RX[18]       | in  | AIB39   | RX[19]       | in  |
| AIB44   | fs_mac_rdy   | in  | AIB45   | unused_AIB45 | tri |
| AIB18   | TX[18]       | out | AIB19   | TX[19]       | out |
| AIB16   | TX[16]       | out | AIB17   | TX[17]       | out |
| AIB14   | TX[14]       | out | AIB15   | TX[15]       | out |
| AIB12   | TX[12]       | out | AIB13   | TX[13]       | out |
| AIB10   | TX[10]       | out | AIB11   | TX[11]       | out |
| AIB41   | ns_fwd_clk   | out | AIB40   | ns_fwd_clkb  | out |
| AIB8    | TX[8]        | out | AIB9    | TX[9]        | out |
| AIB6    | TX[6]        | out | AIB7    | TX[7]        | out |
| AIB4    | TX[4]        | out | AIB5    | TX[5]        | out |
| AIB2    | TX[2]        | out | AIB3    | TX[3]        | out |
| AIB0    | TX[0]        | out | AIB1    | TX[1]        | out |
| AIB46   | unused_AIB46 | tri | AIB47   | unused_AIB47 | tri |

**Table 59. Alternate Leader Channel Bump Table**

### 8.1.2 Alternate (Leader) Channel Bump Map

Figure 81 shows a full alternate bump map with allocation for power and ground bump. VCCIO bumps are dedicated for last stage of AIB I/O buffers. VCCD bumps are dedicated for digital circuits including AIB adapter. If the AIB chiplet is connected to Intel FPGA using Intel EMIB silicon interposer, VCCIO will be powered by FPGA and varies between 0.75V to 0.97V. The maximum VCCIO current consumption should not exceed 40mA per AIB channel.

|    | 1               | 2     | 3     | 4     | 5     | 6     |
|----|-----------------|-------|-------|-------|-------|-------|
|    | Edge of Chiplet |       |       |       |       |       |
| A  | VCCIO           |       | VCCIO |       | VCCIO |       |
| B  |                 | VCCIO |       | VCCIO |       | VCCIO |
| C  | VCCIO           |       | VCCIO |       | VCCIO |       |
| D  |                 | VCCIO |       | VCCIO |       | VCCIO |
| E  | VSS             |       | VSS   |       | VSS   |       |
| F  |                 | VSS   |       | VSS   |       | VSS   |
| G  | AIB16           |       | AIB41 |       | AIB2  |       |
| H  |                 | AIB12 |       | AIB8  |       | AIB46 |
| I  | AIB17           |       | AIB40 |       | AIB3  |       |
| J  |                 | AIB13 |       | AIB9  |       | AIB47 |
| K  | AIB18           |       | AIB10 |       | AIB4  |       |
| L  |                 | AIB14 |       | AIB6  |       | AIB0  |
| M  | AIB19           |       | AIB11 |       | AIB5  |       |
| N  |                 | AIB15 |       | AIB7  |       | AIB1  |
| O  | AIB44           |       | AIB30 |       | AIB24 |       |
| P  |                 | AIB34 |       | AIB43 |       | AIB20 |
| Q  | AIB45           |       | AIB31 |       | AIB25 |       |
| R  |                 | AIB35 |       | AIB42 |       | AIB21 |
| S  | AIB38           |       | AIB32 |       | AIB26 |       |
| T  |                 | AIB36 |       | AIB28 |       | AIB22 |
| U  | AIB39           |       | AIB33 |       | AIB27 |       |
| V  |                 | AIB37 |       | AIB29 |       | AIB23 |
| W  | VSS             |       | VSS   |       | VSS   |       |
| X  |                 | VSS   |       | VSS   |       | VSS   |
| Y  | VCCD            |       | VCCD  |       | VCCD  |       |
| Z  |                 | VCCD  |       | VCCD  |       | VCCD  |
| AA | AIB57           |       | AIB87 |       | AIB93 |       |
| AB |                 | AIB64 |       | AIB83 |       | AIB66 |
| AC | AIB59           |       | AIB86 |       | AIB92 |       |
| AD |                 | AIB65 |       | AIB82 |       | AIB67 |
| AE | AIB51           |       | AIB78 |       | AIB71 |       |
| AF |                 | AIB80 |       | AIB89 |       | AIB68 |
| AG | AIB52           |       | AIB79 |       | AIB70 |       |
| AH |                 | AIB81 |       | AIB88 |       | AIB69 |
| AI | AIB53           |       | AIB58 |       | AIB75 |       |
| AJ |                 | AIB62 |       | AIB95 |       | AIB61 |
| AK | AIB54           |       | AIB63 |       | AIB74 |       |
| AL |                 | AIB60 |       | AIB94 |       | AIB50 |
| AM | AIB49           |       | AIB76 |       | AIB91 |       |
| AN |                 | AIB48 |       | AIB85 |       | AIB72 |
| AO | AIB56           |       | AIB77 |       | AIB90 |       |
| AP |                 | AIB55 |       | AIB84 |       | AIB73 |
| AQ | VSS             |       | VSS   |       | VCCD  |       |
| AR |                 | VSS   |       | VCCD  |       | VCCD  |
| AS | VSS             |       | VSS   |       | VCCD  |       |
| AT |                 | VSS   |       | VCCD  |       | VCCD  |

**Figure 81. Alternate Leader Bump Map**



### 8.1.3 Alternate (Follower) Bump Table

| Bump ID | Bump Name                 | IO  |
|---------|---------------------------|-----|
| AIB61   | unused_AIB61              | out |
| AIB50   | unused_AIB50              | in  |
| AIB72   | unused_AIB72              | out |
| AIB73   | unused_AIB73              | out |
| AIB75   | unused_AIB75              | in  |
| AIB74   | unused_AIB74              | out |
| AIB91   | unused_AIB91              | in  |
| AIB90   | unused_AIB90              | in  |
| AIB95   | fs_sr_data                | in  |
| AIB94   | fs_sr_load                | in  |
| AIB85   | fs_sr_clk                 | in  |
| AIB84   | fs_sr_clkb                | in  |
| AIB76   | unused_AIB76              | in  |
| AIB77   | unused_AIB77              | in  |
| AIB58   | unused_AIB58              | out |
| AIB63   | unused_AIB63              | out |
| AIB48   | unused_AIB48 <sup>3</sup> | in  |
| AIB55   | unused_AIB55 <sup>3</sup> | in  |
| AIB62   | unused_AIB62              | in  |
| AIB60   | unused_AIB60              | in  |
| AIB53   | unused_AIB53 <sup>3</sup> | in  |
| AIB54   | unused_AIB54 <sup>3</sup> | in  |
| AIB49   | fs_mac_rdy                | in  |
| AIB56   | fs_adapter_rstn           | in  |
| AIB51   | unused_AIB51              | in  |
| AIB52   | unused_AIB52              | in  |
| AIB57   | ns_rcv_clk                | out |
| AIB59   | ns_rcv_clkb               | out |
| AIB64   | unused_AIB64              | out |
| AIB65   | ns_adapter_rstn           | out |
| AIB80   | unused_AIB80              | out |
| AIB81   | unused_AIB81              | out |
| AIB78   | unused_AIB78              | out |
| AIB79   | unused_AIB79              | out |

| Bump ID | Bump Name    | IO  |
|---------|--------------|-----|
| AIB87   | fs_rcv_clk   | in  |
| AIB86   | fs_rcv_clkb  | in  |
| AIB83   | ns_sr_clk    | out |
| AIB82   | ns_sr_clkb   | out |
| AIB89   | unused_AIB89 | out |
| AIB88   | unused_AIB88 | out |
| AIB93   | ns_sr_data   | out |
| AIB92   | ns_sr_load   | out |
| AIB71   | unused_AIB71 | in  |
| AIB70   | unused_AIB70 | in  |
| AIB68   | unused_AIB68 | in  |
| AIB69   | unused_AIB69 | in  |
| AIB66   | unused_AIB66 | in  |
| AIB67   | unused_AIB67 | out |
| AIB20   | tx[0]        | out |
| AIB21   | tx[1]        | out |
| AIB22   | tx[2]        | out |
| AIB23   | tx[3]        | out |
| AIB24   | tx[4]        | out |
| AIB25   | tx[5]        | out |
| AIB26   | tx[6]        | out |
| AIB27   | tx[7]        | out |
| AIB28   | tx[8]        | out |
| AIB29   | tx[9]        | out |
| AIB43   | ns_fwd_clk   | out |
| AIB42   | ns_fwd_clkb  | out |
| AIB30   | tx[10]       | out |
| AIB31   | tx[11]       | out |
| AIB32   | tx[12]       | out |
| AIB33   | tx[13]       | out |
| AIB34   | tx[14]       | out |
| AIB35   | tx[15]       | out |
| AIB36   | tx[16]       | out |
| AIB37   | tx[17]       | out |

| Bump ID | Bump Name    | IO  |
|---------|--------------|-----|
| AIB38   | tx[18]       | out |
| AIB39   | tx[19]       | out |
| AIB44   | ns_mac_rdy   | out |
| AIB45   | unused_AIB45 | out |
| AIB18   | rx[18]       | in  |
| AIB19   | rx[19]       | in  |
| AIB16   | rx[16]       | in  |
| AIB17   | rx[17]       | in  |
| AIB14   | rx[14]       | in  |
| AIB15   | rx[15]       | in  |
| AIB12   | rx[12]       | in  |
| AIB13   | rx[13]       | in  |
| AIB10   | rx[10]       | in  |
| AIB11   | rx[11]       | in  |

| Bump ID | Bump Name    | IO |
|---------|--------------|----|
| AIB41   | fs_fwd_clk   | in |
| AIB40   | fs_fwd_clkb  | in |
| AIB8    | rx[8]        | in |
| AIB9    | rx[9]        | in |
| AIB6    | rx[6]        | in |
| AIB7    | rx[7]        | in |
| AIB4    | rx[4]        | in |
| AIB5    | rx[5]        | in |
| AIB2    | rx[2]        | in |
| AIB3    | rx[3]        | in |
| AIB0    | rx[0]        | in |
| AIB1    | rx[1]        | in |
| AIB46   | unused_AIB46 | in |
| AIB47   | unused_AIB47 | in |

**Table 60. Alternate Follower Channel Bump Table**

#### 8.1.4 Alternate (Follower) Channel Bump Map

Figure 81 shows an alternate bump map for a Follower.

| 1               | 2     | 3     | 4     | 5     | 6     |
|-----------------|-------|-------|-------|-------|-------|
| Edge of Chiplet |       |       |       |       |       |
| VCCIO           |       | VCCIO |       | VSS   |       |
|                 | VCCIO |       | VSS   |       | VSS   |
| AIB1            |       | AIB7  |       | AIB15 |       |
|                 | AIB5  |       | AIB11 |       | AIB19 |
| AIB0            |       | AIB6  |       | AIB14 |       |
|                 | AIB4  |       | AIB10 |       | AIB18 |
| AIB47           |       | AIB9  |       | AIB13 |       |
|                 | AIB3  |       | AIB40 |       | AIB17 |
| AIB46           |       | AIB8  |       | AIB12 |       |
|                 | AIB2  |       | AIB41 |       | AIB16 |
| AIB23           |       | AIB29 |       | AIB37 |       |
|                 | AIB27 |       | AIB33 |       | AIB39 |
| AIB22           |       | AIB28 |       | AIB36 |       |
|                 | AIB26 |       | AIB32 |       | AIB38 |
| AIB21           |       | AIB42 |       | AIB35 |       |
|                 | AIB25 |       | AIB31 |       | AIB45 |
| AIB20           |       | AIB43 |       | AIB34 |       |
|                 | AIB24 |       | AIB30 |       | AIB44 |
| AIB69           |       | AIB88 |       | AIB81 |       |
|                 | AIB70 |       | AIB79 |       | AIB52 |
| AIB68           |       | AIB89 |       | AIB80 |       |
|                 | AIB71 |       | AIB78 |       | AIB51 |
| AIB67           |       | AIB82 |       | AIB65 |       |
|                 | AIB92 |       | AIB86 |       | AIB59 |
| AIB66           |       | AIB83 |       | AIB64 |       |
|                 | AIB93 |       | AIB87 |       | AIB57 |
| AIB73           |       | AIB84 |       | AIB55 |       |
|                 | AIB90 |       | AIB77 |       | AIB56 |
| AIB72           |       | AIB85 |       | AIB48 |       |
|                 | AIB91 |       | AIB76 |       | AIB49 |
| AIB50           |       | AIB94 |       | AIB60 |       |
|                 | AIB74 |       | AIB63 |       | AIB54 |
| AIB61           |       | AIB95 |       | AIB62 |       |
|                 | AIB75 |       | AIB58 |       | AIB53 |
| VCCIO           |       | VCCIO |       | VSS   |       |
|                 | VCCIO |       | VSS   |       | VSS   |
| VCCIO           |       | VCCIO |       | VSS   |       |
|                 | VCCIO |       | VSS   |       | VSS   |
| VCCIO           |       | VCCIO |       | VSS   |       |
|                 | VCCIO |       | VSS   |       | VSS   |
| VCCIO           |       | VCCIO |       | VSS   |       |
|                 | VCCIO |       | VSS   |       | VSS   |
| VCCIO           |       | VCCIO |       | VSS   |       |
|                 | VCCIO |       | VSS   |       | VSS   |

Figure 82. Alternate Follower Bump Map

## 8.2 Sideband-Control-Signal Shift Register Mapping (AIB Plus only)

| Bit Order | Side Band Control Signals from Leader to Follower | Bit Width | Default Value | Descriptions   |
|-----------|---|-----------|---------------|--|
| [80]      | ms_osc_transfer_en                                | 1         | 1             | MS output to SL to indicate MS OSC transfer has been enabled.      |
| [79]      | Reserved  | 1         | 1             | Reserved   |
| [78]      | ms_tx_transfer_en                                 | 1         | 1             | TX output to RX to indicate that TX OSC transfer has been enabled. |
| [77]      | Reserved  | 1         | 1             | Reserved   |
| [76]      | Reserved  | 1         | 1             | Reserved   |
| [75]      | ms_rx_transfer_en                                 | 1         | 1             | RX output to TX to indicate that RX is ready for data transfer.    |
| [74]      | ms_rx_dll_lock                                    | 1         | 1             | RX output to TX to indicate that RX DLL achieves lock.             |
| [73:71]   | Reserved  | 3         | 3'b111        | Reserved   |
| [70:69]   | Reserved  | 2         | 2'b11         | Reserved   |
| [68]      | ms_tx_dcc_cal_done                                | 1         | 1             | TX output to RX to indicate that DCC calibration is done           |
| [67]      | Reserved  | 1         | 0             | Reserved   |
| [66]      | Reserved  | 1         | 1             | Reserved   |
| [65:12]   | User defined                                      | 1         | 0             | For application use  |
| [11]      | User defined                                      | 1         | 0             | For application use  |
| [10]      | User defined                                      | 1         | 0             | For application use  |
| [9]       | User defined                                      | 1         | 0             | For application use  |
| [8]       | User defined                                      | 1         | 0             | For application use  |
| [7]       | Reserved  | 1         | 1             | Reserved   |
| [6]       | Reserved  | 1         | 0             | Reserved   |
| [5]       | Reserved  | 1         | 1             | Reserved   |
| [4:0]     | User defined                                      | 1         | 0             | For application use  |

**Table 61. Leader Sideband-Control Signals**

| Bit Order | Side Band Control Signals from Follower to Leader | Bit Width | Default Value | Descriptions  |
|-----------|---|-----------|---------------|---|
| [72]      | sl_osc_transfer_en                                | 1         | 1             | RX output to MS to indicate SL OSC transfer has been enabled. |

| Bit Order | Side Band Control Signals from Follower to Leader | Bit Width | Default Value | Descriptions   |
|-----------|---|-----------|---------------|--|
| [71]      | Reserved  | 1         | 0             | Reserved   |
| [70]      | sl_rx_transfer_en                                 | 1         | 1             | RX output to MS to indicate SL RX is ready to receive data   |
| [69]      | sl_rx_dcc_dll_lock_req                            | 1         | 1             | DLL/DCC calibration request from Follower RX to Leader TX AIB to start full DLL/DCC calibration.     |
| [68]      | sl_rx_dll_lock                                    | 1         | 1             | RX output to MS (adapter and PHY) to indicate SL DLL achieves lock.                                  |
| [67:65]   | Reserved  | 3         | 3'b000        | Reserved   |
| [64]      | sl_tx_transfer_en                                 | 1         | 1             | TX sends to RX (adapter and PHY) that it is ready for TX data transfer.                              |
| [63]      | sl_tx_dcc_dll_lock_req                            | 1         | 1             | PHY DLL/DCC calibration request from Follower TX to Leader RX AIB to start full DLL/DCC calibration. |
| [62]      | Reserved  | 1         | 0             | Reserved   |
| [61]      | Reserved  | 1         | 0             | Reserved   |
| [60]      | Reserved  | 1         | 1             | Reserved   |
| [59]      | Reserved  | 1         | 0             | Reserved   |
| [58]      | Reserved  | 1         | 1             | Reserved   |
| [57:32]   | User defined                                      | 1         | 0             | For application use  |
| [31]      | sl_tx_dcc_cal_done                                | 1         | 1             | TX AIB to notify MS that DCC calibration is complete.  |
| [30:28]   | User defined                                      | 1         | 0             | For application use  |
| [27]      | Reserved  | 1         | 0             | Reserved   |
| [26:17]   | User defined                                      | 1         | 0             | For application use  |
| [16]      | User defined                                      | 1         | 0             | For application use  |
| [15]      | User defined                                      | 1         | 0             | For application use  |
| [14]      | User defined                                      | 1         | 0             | For application use  |
| [13]      | User defined                                      | 1         | 0             | For application use  |
| [12:0]    | User defined                                      | 1         | 0             | For application use  |

**Table 62. Follower Sideband-Control Signal**

## 8.3 Test Pattern Extension

The ability for an AIB PHY to generate and check test patterns is useful for some application and test purposes. This optional test pattern extension describes the pattern generation and checking capabilities.

### 8.3.1 Pattern Generator and Checker

The test pattern function contains a pattern generator block and a pattern checker block for each AIB TX/RX signal pair. The pattern generator and checker may be programmed to use the PRBS polynomials of Table 63 and programmable patterns.

| Name   | Polynomial                     |
|--------|--------------------------------|
| PRBS40 | $X^{40} + X^{38} + X^{21} + 1$ |
| PRBS31 | $X^{31} + X^{28} + 1$          |
| PRBS23 | $X^{23} + X^{18} + 1$          |
| PRBS10 | $X^{10} + X^7 + 1$             |
| PRBS7  | $X^7 + X^6 + 1$                |

**Table 63. Test Pattern Polynomials**

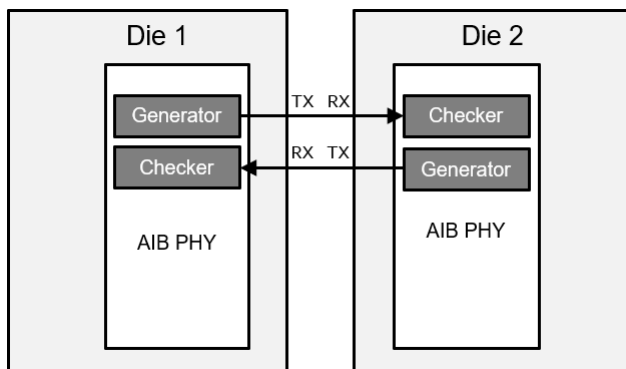
The pattern generator may be initialized (seeded) to an arbitrary value.

In addition to the PRBS patterns, the pattern generator is capable of storing a programmable pattern and pattern length. In programmable pattern mode, the generator repeats the pattern continuously. The programmable pattern generator is capable of producing all 0s, all 1s, alternating 0 and 1 and other patterns by storing the desired pattern into the pattern generator. The pattern checker is similarly capable of checking that the received RX data matches the pattern stored in the pattern checking block.

### 8.3.2 Pattern Data Paths

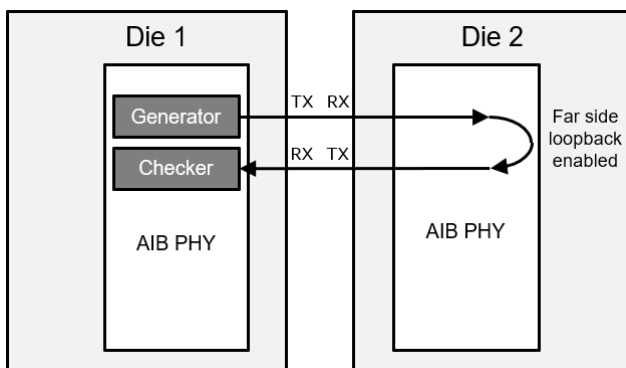
The AIB PHY may be programmed to select the pattern generation block as the source of AIB TX data in place of the MAC to PHY *data\_in* or *data\_in\_f* signals of Table 7.

Figure 83 is a supported two die test case where the test patterns in each direction are independent.



**Figure 83. Two die test case with independent pattern generators and checkers**

Figure 84 is a supported two die test case using the pattern generator and checker on one die and far side loopback (section 5.2) on the second die.



**Figure 84. Two die test case using far side loopback**