

## Intelligent Network Interface Controller for 50 Mbit/s Automotive Networks

### Features

- Complete 50 Mbit/s synchronous network interface
- Embedded network management functions
  - Network protection mode
  - Hardware & application watchdog timer
  - Intelligent muting
  - Diagnostics
  - Fallback Operation
- IEEE MAC addressing and Ethernet channel
- Media Local Bus (MediaLB<sup>®</sup>) Port (OS81212)
  - Eases inter-chip communication and streaming
  - MediaLB 3-pin interface at speeds up to 1024xFs
- I<sup>2</sup>C™ Control Port inter-chip message exchange
- Streaming Port supports synchronous, fixed latency data exchange for a variety of serial audio formats including time-division multiplex (TDM) and pulse density modulation (PDM)
- SPI Port supports asynchronous and control packets (OS81212/4)
- General Purpose I/O (GPIO) Port
- Remote control and configuration for operation without a local External Host Controller.
  - I<sup>2</sup>C (master) message tunneling
  - GPIO port control
- Operating voltages 3.3 V/1.8 V
- Available in 56-pin (OS81212) and 48-pin (OS81214/6) QFN packages with exposed pad
- -40 to +125 °C junction temperature

### Conformity

This document applies to hardware revision B2B

### Applications

- Automotive infotainment network nodes including instrument cluster, amplifier, headrest speakers, microphones, acoustic processing units, and rear seat entertainment.

### General Description

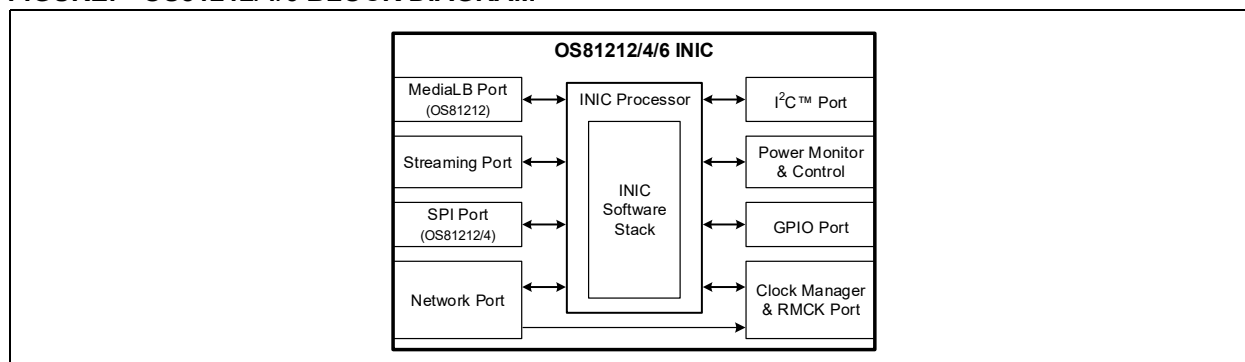
The OS81212/4/6 is a highly integrated *Intelligent Network Interface Controller* (INIC) for 50 Mbit/s INICnet-based automotive networks with a transformer-less balanced media physical layer (bPHY) optimized for unshielded twisted pair (UTP) copper wire.

The INIC provides encapsulation of all low-level functions necessary to develop a network-compliant device, significantly simplifying network implementation in a node. Integration of the *INIC Software Stack* into the INIC provides network-compliant real-time behavior. The *INIC Software Stack* significantly relieves the External Host Controller (EHC) from real-time processing tasks. Supervision of the application is also provided, including a protection mode that is entered when an application is not present (i.e. start-up) or the EHC malfunctions. This protection mode prevents application malfunctions from influencing the integrity of the network and the system.

When an EHC is engaged, a message-based interface, as opposed to a register-based interface, is available for communication with INIC. A unified and centralized network management software stack (UNICENS) is available for the EHC to build a complete, lean, system solution.

The INIC conforms to the ISO 21806 standard developed by the International Organization for Standardization (ISO<sup>®</sup>).

**FIGURE: OS81212/4/6 BLOCK DIAGRAM**



## TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at [docerrors@microchip.com](mailto:docerrors@microchip.com). We welcome your feedback.

### Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

<http://www.microchip.com>

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000000A is version A of document DS30000000).

### Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; <http://www.microchip.com>
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

### Customer Notification System

Register on our web site at [www.microchip.com](http://www.microchip.com) to receive the most current information on all of our products.

## TABLE OF CONTENTS

|            |  |     |
|------------|--|-----|
| 1          | OS81212/4/6 PINOUT .....                       | 5   |
| 2          | EQUIVALENT SCHEMATICS FOR PINS .....           | 9   |
| 3          | OVERVIEW .....                                 | 11  |
| 4          | INIC PROCESSOR .....                           | 22  |
| 5          | I <sup>2</sup> C PORT .....                    | 23  |
| 6          | MediaLB PORT .....                             | 28  |
| 7          | STREAMING PORT .....                           | 35  |
| 8          | SPI PORT .....                                 | 40  |
| 9          | EXTERNAL POWER MANAGEMENT .....                | 59  |
| 10         | CLOCK MANAGER .....                            | 61  |
| 11         | RMCK PORT .....                                | 62  |
| 12         | GENERAL PURPOSE INPUT/OUTPUT (GPIO) PORT ..... | 63  |
| 13         | JTAG PORT .....                                | 66  |
| 14         | ELECTRICAL CHARACTERISTICS .....               | 69  |
| 15         | APPLICATION INFORMATION .....                  | 90  |
| 16         | PACKAGING INFORMATION .....                    | 99  |
| APPENDIX A | FEATURE MATRIX .....                           | 107 |
| APPENDIX B | REFERENCES .....                               | 108 |
| APPENDIX C | REVISION HISTORY .....                         | 110 |
| APPENDIX D | LIST OF ACRONYMS .....                         | 111 |
| APPENDIX E | LIST OF TABLES .....                           | 112 |
| APPENDIX F | LIST OF FIGURES .....                          | 113 |

## Conventions

The following abbreviations and symbols are used to improve readability.

| Example                 | Description  |
|-------------------------|--|
| <b>BIT</b>              | Name of a single bit within a field  |
| <b>FIELD.BIT</b>        | Name of a single bit (BIT) in FIELD  |
| x...y                   | Range from x to y, inclusive   |
| <b>BITS[m:n]</b>        | Groups of bits from m to n, inclusive  |
| <b>PIN</b>              | Pin Name   |
| <b>SIGNAL</b>           | Signal Name  |
| msb, lsb                | Most significant bit, least significant bit  |
| MSB, LSB                | Most significant byte, least significant byte  |
| zzzzb                   | Binary number (value zzzz)   |
| 0xzzz                   | Hexadecimal number (value zzz)   |
| zzh                     | Hexadecimal number (value zz)  |
| rsvd                    | Reserved memory location. Must write 0, read value indeterminate   |
| code                    | Instruction code, or API function or parameter   |
| <i>Multi Word Name</i>  | Used for multiple words that are considered a single unit, such as:<br><i>Resource Allocate</i> message, or <i>Connection Label</i> , or <i>Decrement Stack Pointer</i> instruction. |
| <i>Section Name</i>     | Emphasis, Reference, Section or Document name.   |
| $\overline{\text{VAL}}$ | Over-bar indicates active low pin or register bit  |
| x                       | Don't care   |
| <Parameter>             | <> indicate a Parameter is optional or is only used under some conditions  |
| {,Parameter}            | Braces indicate Parameter(s) that repeat one or more times.  |
| [Parameter]             | Brackets indicate a nested Parameter. This Parameter is not real and actually decodes into one or more real parameters.  |

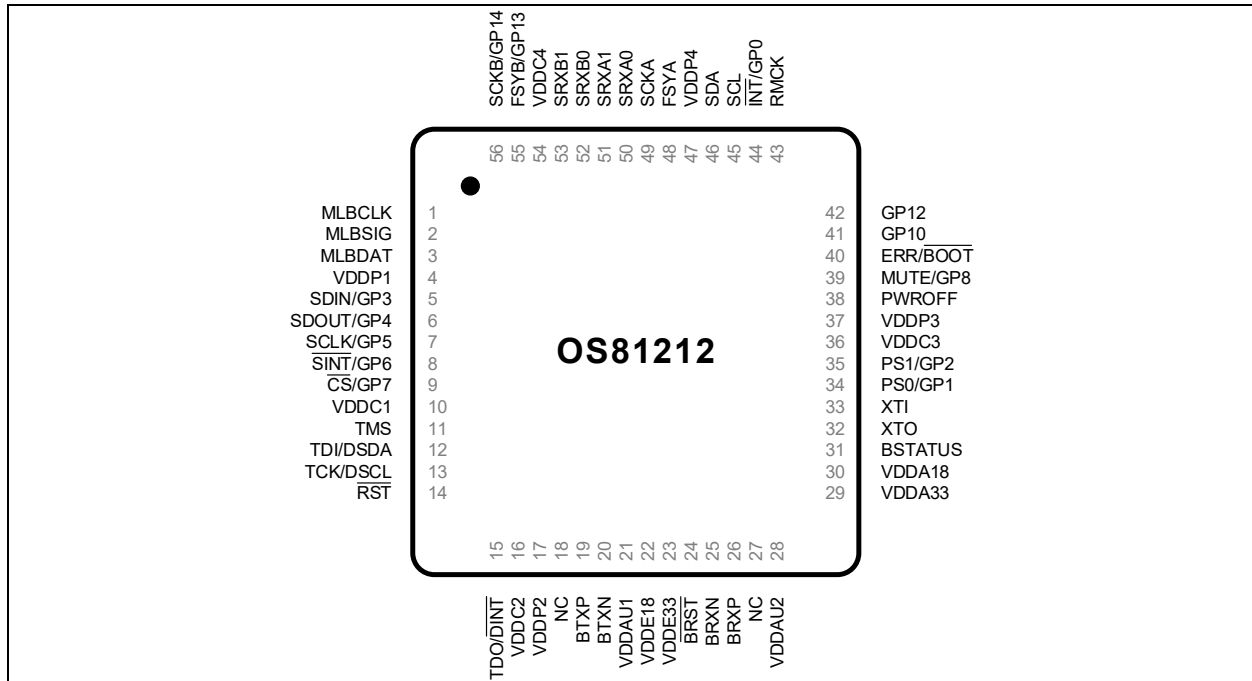
## 1.0 OS81212/4/6 PINOUT

Input pins must not be left floating; therefore, they must be driven, have pull-ups or pull-downs, or connected directly to a power pin or ground.

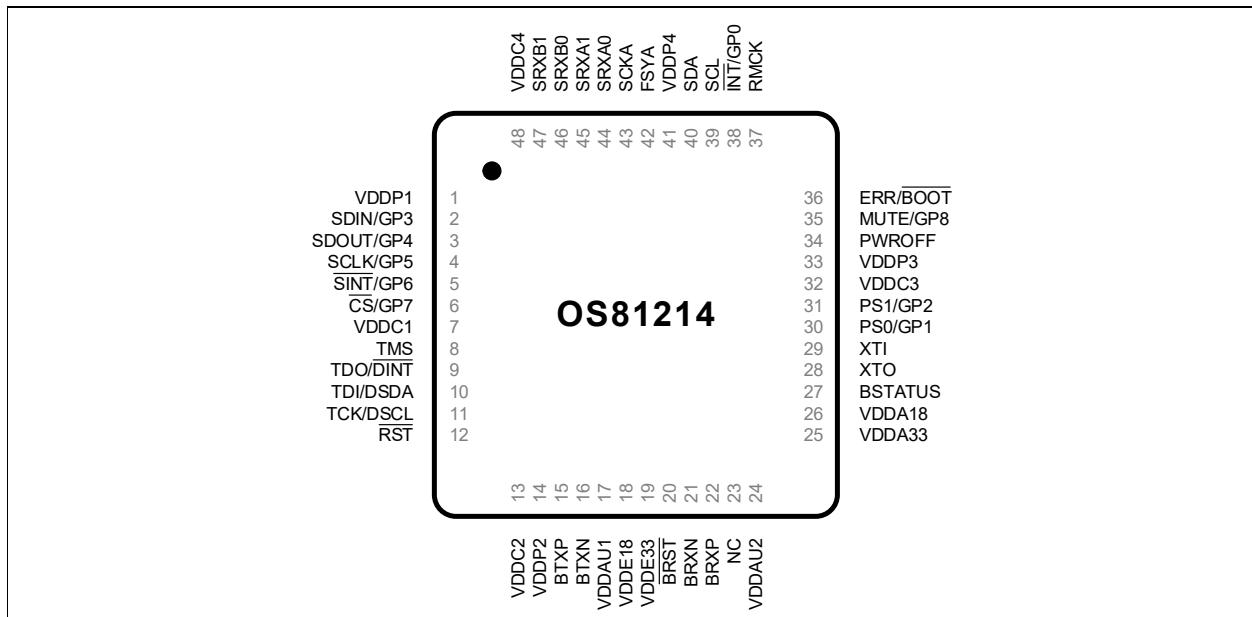
Digital pins that can be configured as outputs (i.e., pin types D<sub>OUT</sub>, D<sub>OUTD</sub>, D<sub>OUTZ</sub>, D<sub>I/O</sub>, and D<sub>I/OD</sub>) are high impedance during power-up/reset. The pin types shown in [Table 1-1](#) are the values after power-up/reset.

**Note:** Hardware port pins other than the I<sup>2</sup>C Port and JTAG Port pins are high impedance until opened/enabled. This includes the **RMCK** pin, which remains high impedance until the output clock is specifically enabled.

**FIGURE 1-1: OS81212 PIN DIAGRAM**



**FIGURE 1-2: OS81214 PIN DIAGRAM**



# OS81212/4/6

FIGURE 1-3: OS81216 PIN DIAGRAM

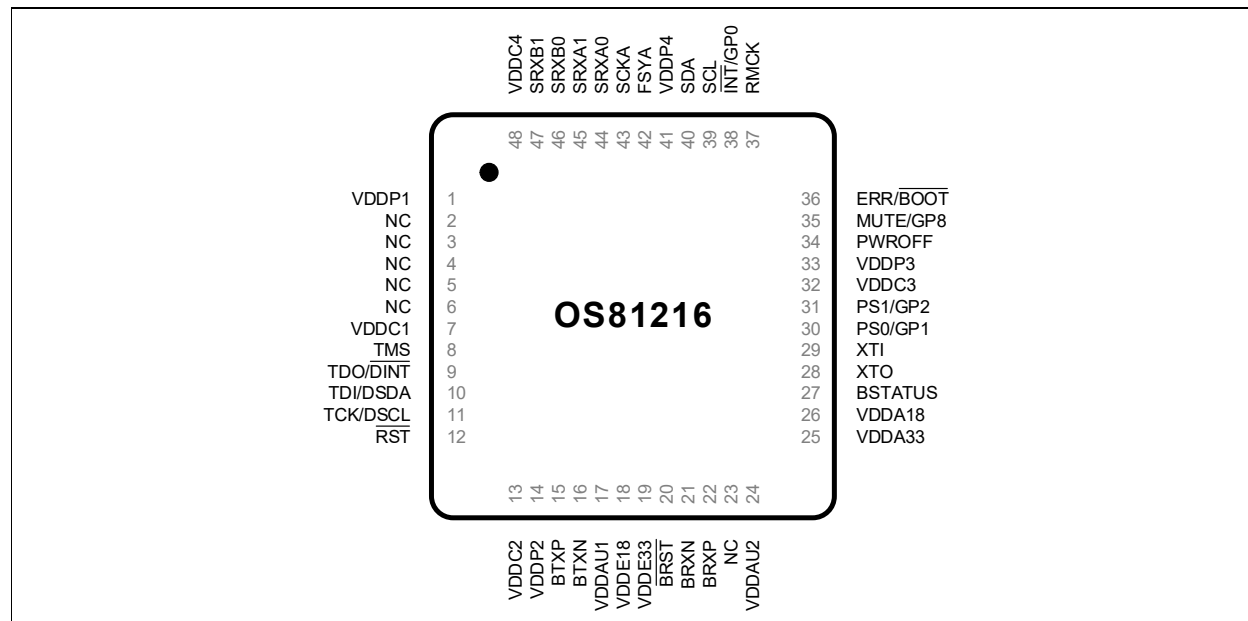


TABLE 1-1: OS81212/4/6 PIN ALLOCATION TABLE

| OS81212 Pin | OS81214 Pin | OS81216 Pin | Name                | Type              | HW Port | Description   |
|-------------|-------------|-------------|---------------------|-------------------|---------|---|
| 1           | -           | -           | MLBCLK <sup>2</sup> | D <sub>OUT</sub>  | MediaLB | Singled-ended Clock line for MediaLB 3-pin Interface  |
| 2           | -           | -           | MLBSIG <sup>2</sup> | D <sub>I/O</sub>  | MediaLB | Singled-ended Signal line for MediaLB 3-pin Interface |
| 3           | -           | -           | MLBDAT <sup>2</sup> | D <sub>I/O</sub>  | MediaLB | Singled-ended Data line for MediaLB 3-pin Interface   |
| 4           | 1           | 1           | VDDP1               |                   |         | 3.3 V periphery power supply (digital)                |
| 5           | 2           | -           | SDIN                | D <sub>IN</sub>   | SPI     | Data In (MOSI - Master Out, Slave In)                 |
|             |             |             | GP3                 | D <sub>I/O</sub>  | GPIO    | General Purpose Input/Output 3                        |
| -           | -           | 2           | NC                  |                   |         | No Connect. This pin must be left open and floating.  |
| 6           | 3           | -           | SDOUT               | D <sub>OUT</sub>  | SPI     | Data Out (MISO - Master In, Slave Out)                |
|             |             |             | GP4                 | D <sub>I/O</sub>  | GPIO    | General Purpose Input/Output 4                        |
| -           | -           | 3           | NC                  |                   |         | No Connect. This pin must be left open and floating.  |
| 7           | 4           | -           | SCLK                | D <sub>IN</sub>   | SPI     | Clock   |
|             |             |             | GP5                 | D <sub>I/O</sub>  | GPIO    | General Purpose Input/Output 5                        |
| -           | -           | 4           | NC                  |                   |         | No Connect. This pin must be left open and floating.  |
| 8           | 5           | -           | SINT                | D <sub>OUT</sub>  | SPI     | Interrupt (active low)                                |
|             |             |             | GP6                 | D <sub>I/O</sub>  | GPIO    | General Purpose Input/Output 6                        |
| -           | -           | 5           | NC                  |                   |         | No Connect. This pin must be left open and floating.  |
| 9           | 6           | -           | CS                  | D <sub>IN</sub>   | SPI     | Chip Select (active low)                              |
|             |             |             | GP7                 | D <sub>I/O</sub>  | GPIO    | General Purpose Input/Output 7                        |
| -           | -           | 6           | NC                  |                   |         | No Connect. This pin must be left open and floating.  |
| 10          | 7           | 7           | VDDC1               |                   |         | 1.8 V core power supply (digital)                     |
| 11          | 8           | 8           | TMS <sup>1</sup>    | D <sub>IN</sub>   | JTAG    | Test Mode Select                                      |
| -           | 9           | 9           | TDO <sup>1</sup>    | D <sub>OUTZ</sub> | JTAG    | Test Data Output                                      |
|             |             |             | DINT <sup>1</sup>   | D <sub>OUTD</sub> |         | Debug Interrupt (active low)                          |

**Note 1:** Pull-up resistor required.

**2:** Pull-down resistor required.

**TABLE 1-1: OS81212/4/6 PIN ALLOCATION TABLE (CONTINUED)**

| OS81212 Pin | OS81214 Pin | OS81216 Pin | Name                                  | Type              | HW Port | Description  |
|-------------|-------------|-------------|---------------------------------------|-------------------|---------|--|
| 12          | 10          | 10          | TDI <sup>1</sup>                      | D <sub>IN</sub>   | JTAG    | Test Data Input  |
|             |             |             | DSDA <sup>1</sup>                     | D <sub>I/OD</sub> |         | Debug Data   |
| 13          | 11          | 11          | TCK <sup>1</sup>                      | D <sub>IN</sub>   | JTAG    | Test Clock Input   |
|             |             |             | DSCL <sup>1</sup>                     | D <sub>I/OD</sub> |         | Debug Clock  |
| 14          | 12          | 12          | $\overline{\text{RST}}$               | D <sub>IN</sub>   |         | Hardware Reset Input (active low). (Pull-up resistor to <b>VDDP<sub>n</sub></b> supply should be used when not driven high by an external device. A series resistor should be used in lieu of the pull-up when always driven by an external device.) |
| 15          | -           | -           | TDO <sup>1</sup>                      | D <sub>OUTZ</sub> | JTAG    | Test Data Output   |
|             |             |             | $\overline{\text{DINT}}$ <sup>1</sup> | D <sub>OUTD</sub> |         | Debug Interrupt (active low)   |
| 16          | 13          | 13          | VDDC2                                 |                   |         | 1.8 V core power supply (digital)  |
| 17          | 14          | 14          | VDDP2                                 |                   |         | 3.3 V periphery power supply (digital)   |
| 18          | -           | -           | NC                                    |                   |         | No Connect. This pin must be left open and floating.   |
| 19          | 15          | 15          | BTXP                                  | A <sub>I/O</sub>  | Network | Positive (differential) bPHY network transmitter output  |
| 20          | 16          | 16          | BTXN                                  | A <sub>I/O</sub>  | Network | Negative (differential) bPHY network transmitter output  |
| 21          | 17          | 17          | VDDAU1                                |                   |         | 3.3 V continuous power supply (analog)   |
| 22          | 18          | 18          | VDDE18                                |                   |         | 1.8 V bPHY power supply (analog)   |
| 23          | 19          | 19          | VDDE33                                |                   |         | 3.3 V bPHY power supply (analog)   |
| 24          | 20          | 20          | $\overline{\text{BRST}}$ <sup>1</sup> | A <sub>I/O</sub>  | Network | Hardware Reset Input (active low) for the Balanced Media Physical Layer. When asserted, the transmitter output is disabled. A pull-up resistor to <b>VDDP<sub>n</sub></b> is required.   |
| 25          | 21          | 21          | BRXN                                  | A <sub>I/O</sub>  | Network | Negative (differential) bPHY network receiver input  |
| 26          | 22          | 22          | BRXP                                  | A <sub>I/O</sub>  | Network | Positive (differential) bPHY network receiver input  |
| 27          | 23          | 23          | NC                                    |                   |         | No Connect. This pin must be left open and floating.   |
| 28          | 24          | 24          | VDDAU2                                |                   |         | 3.3 V continuous power supply (analog)   |
| 29          | 25          | 25          | VDDA33                                |                   |         | 3.3 V power supply (analog)  |
| 30          | 26          | 26          | VDDA18                                |                   |         | 1.8 V power supply (analog)  |
| 31          | 27          | 27          | BSTATUS                               | D <sub>OUT</sub>  | Network | bPHY Network Activity Status Output used during wake-up:<br><ul style="list-style-type: none"> <li>- Driven low when a valid signal is detected</li> <li>- Driven high to <b>VDDAU<sub>n</sub></b> when a qualified signal is not present</li> </ul> |
| 32          | 28          | 28          | XTO                                   | A <sub>I/O</sub>  |         | Crystal Oscillator Output  |
| 33          | 29          | 29          | XTI                                   | A <sub>I/O</sub>  |         | Crystal Oscillator Input or External CMOS Clock Input  |
| 34          | 30          | 30          | PS0                                   | D <sub>IN</sub>   |         | External Power Management Status Bit 0   |
|             |             |             | GP1                                   | D <sub>I/O</sub>  | GPIO    | General Purpose Input/Output 1   |
| 35          | 31          | 31          | PS1                                   | D <sub>IN</sub>   |         | External Power Management Status Bit 1   |
|             |             |             | GP2                                   | D <sub>I/O</sub>  | GPIO    | General Purpose Input/Output 2   |
| 36          | 32          | 32          | VDDC3                                 |                   |         | 1.8 V core power supply (digital)  |
| 37          | 33          | 33          | VDDP3                                 |                   |         | 3.3 V periphery power supply (digital)   |

**Note 1:** Pull-up resistor required.

**2:** Pull-down resistor required.

# OS81212/4/6

**TABLE 1-1: OS81212/4/6 PIN ALLOCATION TABLE (CONTINUED)**

| OS81212 Pin | OS81214 Pin | OS81216 Pin | Name                                  | Type              | HW Port          | Description  |
|-------------|-------------|-------------|---------------------------------------|-------------------|------------------|--|
| 38          | 34          | 34          | PWROFF <sup>1</sup>                   | D <sub>OUTD</sub> |                  | External Power Management Power-Down Indicator. This pin is driven low by INIC after initialization. When high, indicates that the INIC Processor is ready to be shut down. A pull-up resistor is required when used. If not used, this pin may be left unconnected. |
| 39          | 35          | 35          | MUTE <sup>1</sup>                     | D <sub>OUTD</sub> |                  | Mute Indicator Output. A pull-up resistor is required when used. If not used, this pin may be left unconnected.  |
|             |             |             | GP8                                   | D <sub>I/O</sub>  | GPIO             | General Purpose Input/Output 8   |
| 40          | 36          | 36          | ERR                                   | D <sub>OUT</sub>  |                  | Error Indicator Output   |
|             |             |             | $\overline{\text{BOOT}}$ <sup>1</sup> | D <sub>IN</sub>   |                  | Configuration Pin. This pin is attached to the configuration/debug header and used by the Microchip <a href="#">INICkit Tool [3]</a> to load initial configuration data into INIC. May also be connected to the EHC to allow in-system configuration of the INIC.    |
| 41          | -           | -           | GP10                                  | D <sub>I/O</sub>  | GPIO             | General Purpose Input/Output 10  |
| 42          | -           | -           | GP12                                  | D <sub>I/O</sub>  | GPIO             | General Purpose Input/Output 12  |
| 43          | 37          | 37          | RMCK                                  | D <sub>OUT</sub>  | RMCK             | Recovered Master Clock Output  |
| 44          | 38          | 38          | $\overline{\text{INT}}$ <sup>1</sup>  | D <sub>OUTD</sub> | I <sup>2</sup> C | Interrupt (active low). Indicates a service request from the EHC when the Control Port is operating as an I <sup>2</sup> C slave.  |
|             |             |             | GP0                                   | D <sub>I/O</sub>  | GPIO             | General Purpose Input/Output 0   |
| 45          | 39          | 39          | SCL <sup>1</sup>                      | D <sub>I/OD</sub> | I <sup>2</sup> C | Clock  |
| 46          | 40          | 40          | SDA <sup>1</sup>                      | D <sub>I/OD</sub> | I <sup>2</sup> C | Data   |
| 47          | 41          | 41          | VDDP4                                 |                   |                  | 3.3 V periphery power supply (digital)   |
| 48          | 42          | 42          | FSYA                                  | D <sub>I/O</sub>  | Streaming        | Frame Sync for Streaming Port A  |
| 49          | 43          | 43          | SCKA                                  | D <sub>I/O</sub>  | Streaming        | Bit Clock for Streaming Port A   |
| 50          | 44          | 44          | SRXA0                                 | D <sub>I/O</sub>  | Streaming        | Data I/O Signal 0 for Streaming Port A   |
| 51          | 45          | 45          | SRXA1                                 | D <sub>I/O</sub>  | Streaming        | Data I/O Signal 1 for Streaming Port A   |
| 52          | 46          | 46          | SRXB0                                 | D <sub>I/O</sub>  | Streaming        | Data I/O Signal 0 for Streaming Port B   |
| 53          | 47          | 47          | SRXB1                                 | D <sub>I/O</sub>  | Streaming        | Data I/O Signal 1 for Streaming Port B   |
| 54          | 48          | 48          | VDDC4                                 |                   |                  | 1.8 V core power supply (digital)  |
| 55          | -           | -           | FSYB                                  | D <sub>I/O</sub>  | Streaming        | Frame Sync for Streaming Port B  |
|             | -           | -           | GP13                                  | D <sub>I/O</sub>  | GPIO             | General Purpose Input/Output 13  |
| 56          | -           | -           | SCKB                                  | D <sub>I/O</sub>  | Streaming        | Bit Clock for Streaming Port B   |
|             | -           | -           | GP14                                  | D <sub>I/O</sub>  | GPIO             | General Purpose Input/Output 14  |
| ePAD        | ePAD        | ePAD        | GND                                   |                   |                  | The exposed paddle on the bottom side of the QFN package is the primary ground for the OS81212/4/6 and must be connected to ground on the PCB for proper operation.  |

**Note 1:** Pull-up resistor required.

**2:** Pull-down resistor required.



## 2.0 EQUIVALENT SCHEMATICS FOR PINS

FIGURE 2-1: PIN-EQUIVALENT FOR ANALOG I/O PIN -  $A_{I/O}$

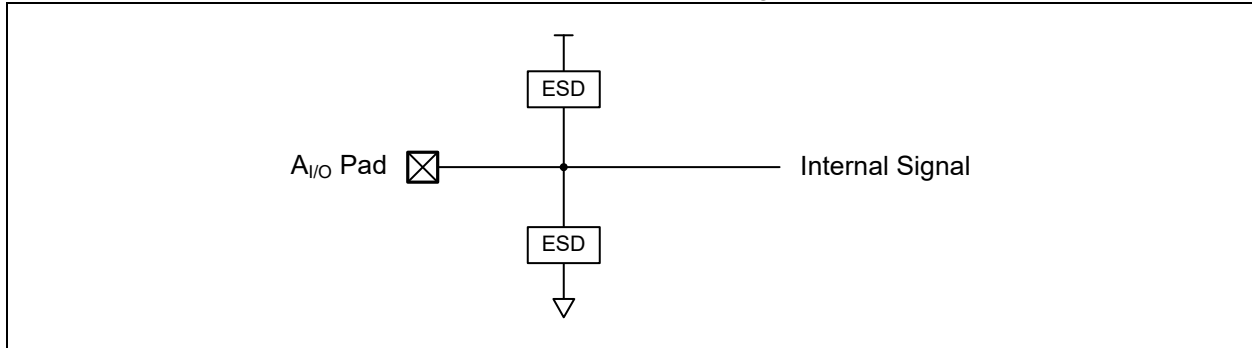


FIGURE 2-2: PIN-EQUIVALENT FOR DIGITAL INPUT PIN -  $D_{IN}$

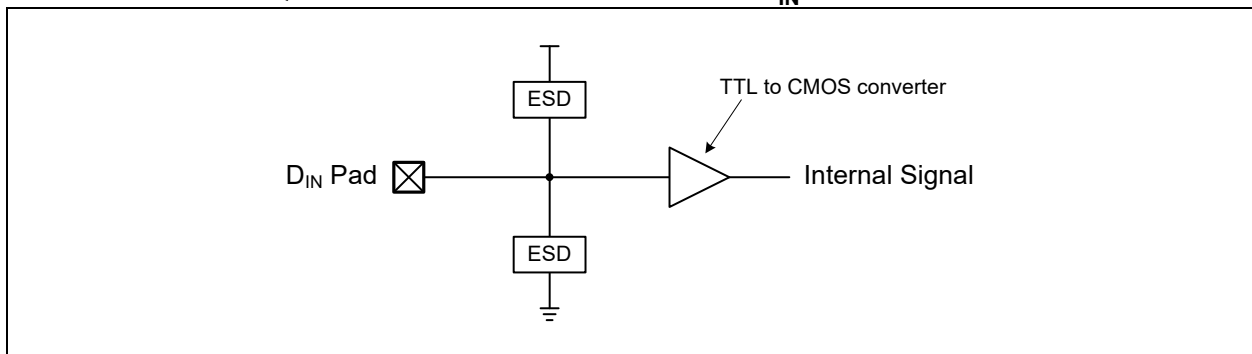


FIGURE 2-3: PIN-EQUIVALENT FOR DIGITAL OUTPUT PIN -  $D_{OUT}$

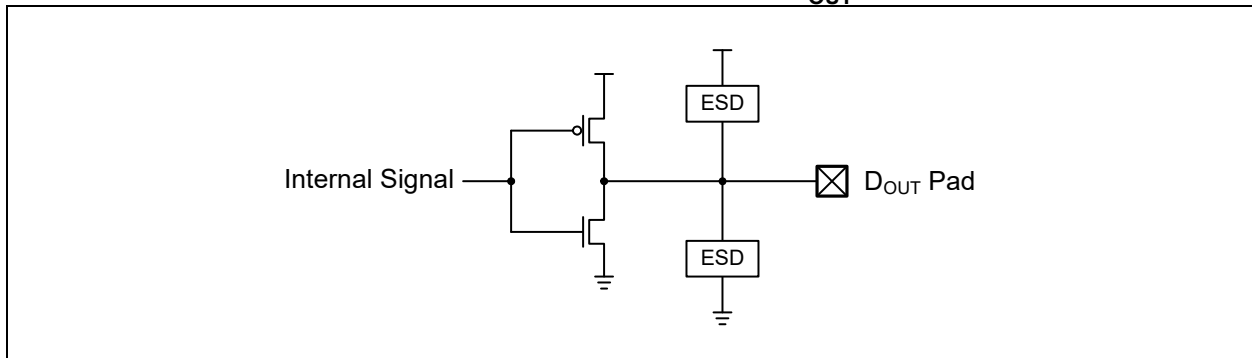
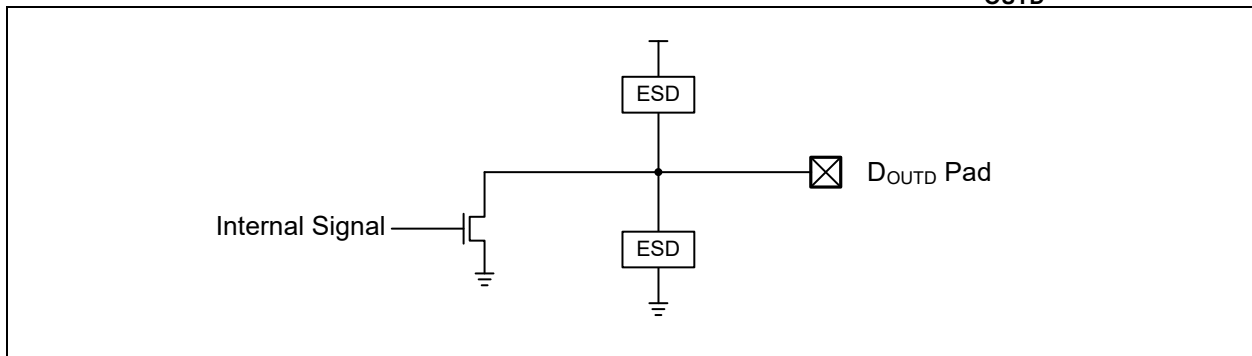
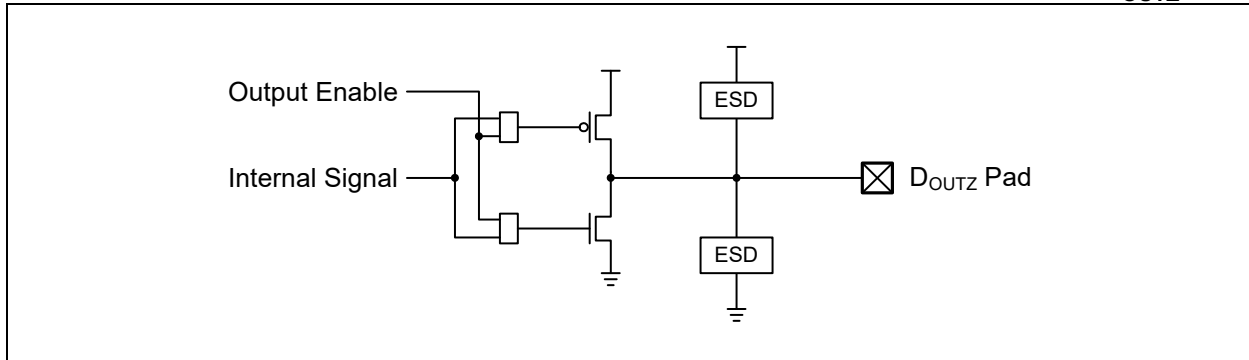


FIGURE 2-4: PIN-EQUIVALENT FOR OPEN-DRAIN DIGITAL OUTPUT PIN -  $D_{OUTD}$

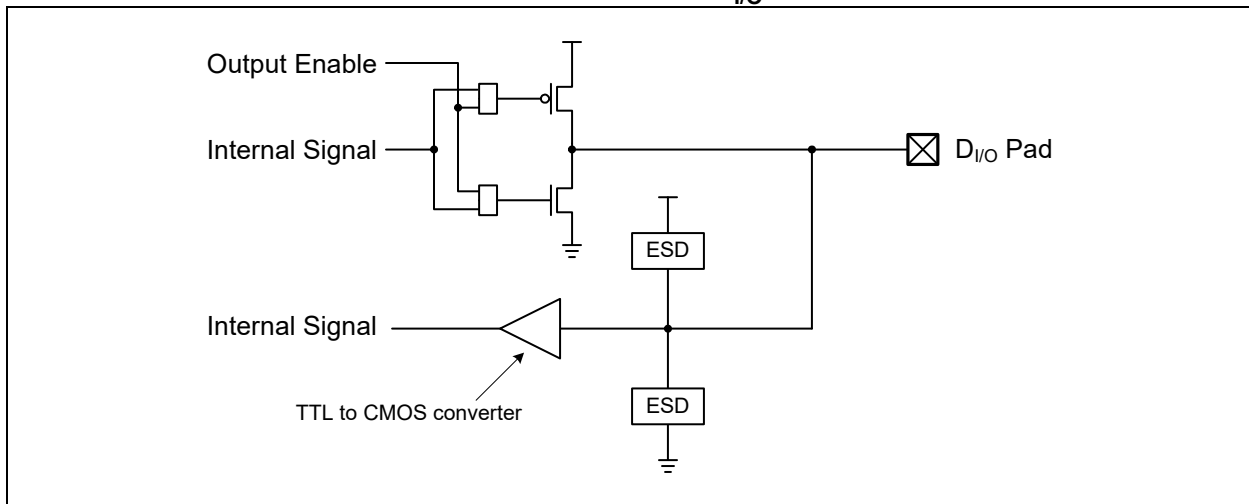


# OS81212/4/6

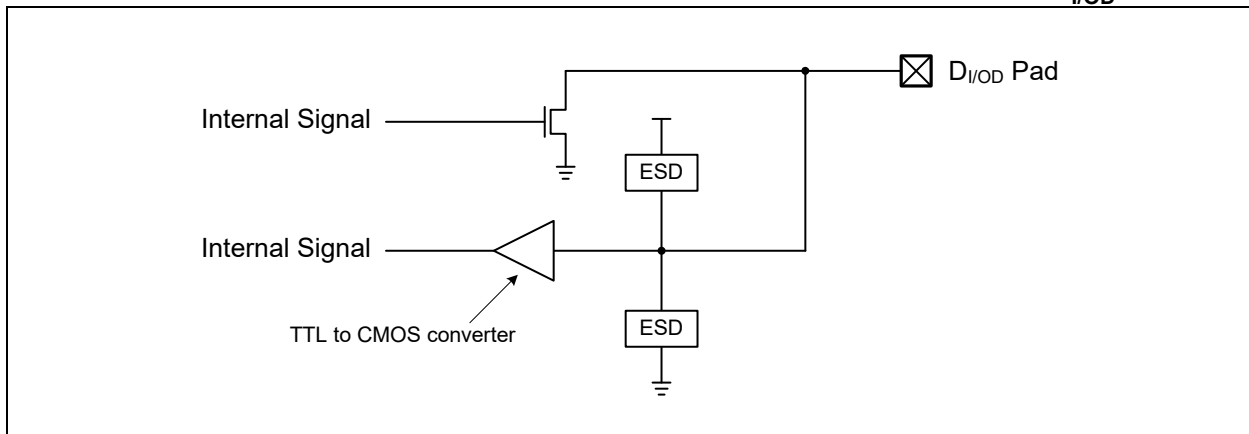
**FIGURE 2-5: PIN-EQUIVALENT FOR DIGITAL OUTPUT PIN WITH HIGH-Z CONTROL - D<sub>OUTZ</sub>**



**FIGURE 2-6: PIN-EQUIVALENT FOR DIGITAL I/O PIN - D<sub>I/O</sub>**



**FIGURE 2-7: PIN-EQUIVALENT FOR DIGITAL INPUT/OPEN-DRAIN OUTPUT PIN - D<sub>I/OD</sub>**



## 3.0 OVERVIEW

**Note:** This Data Sheet is designed to be used in conjunction with the [INIC Interface Specification \[4\]](#) to provide a complete reference for the operation and use of the OS81212/4/6.

The OS81212/4/6 are network transceiver devices belonging to the Microchip *Intelligent Network Interface Controller* (INIC) family. The OS81212/4/6 INICs create a time division multiplexed synchronous network by transmitting 128 bytes frames at a 48 kHz frame rate (Fs) yielding a 49.2 Mbit/s data rate. A network of *Electronic Control Units* (ECUs) based on INICs creates an INICnet network. INICnet can be configured and managed at the application level by the provided UNICENS software stack. Alternatively, the MOST NetServices software stack can be used to create a MOST compliant network.

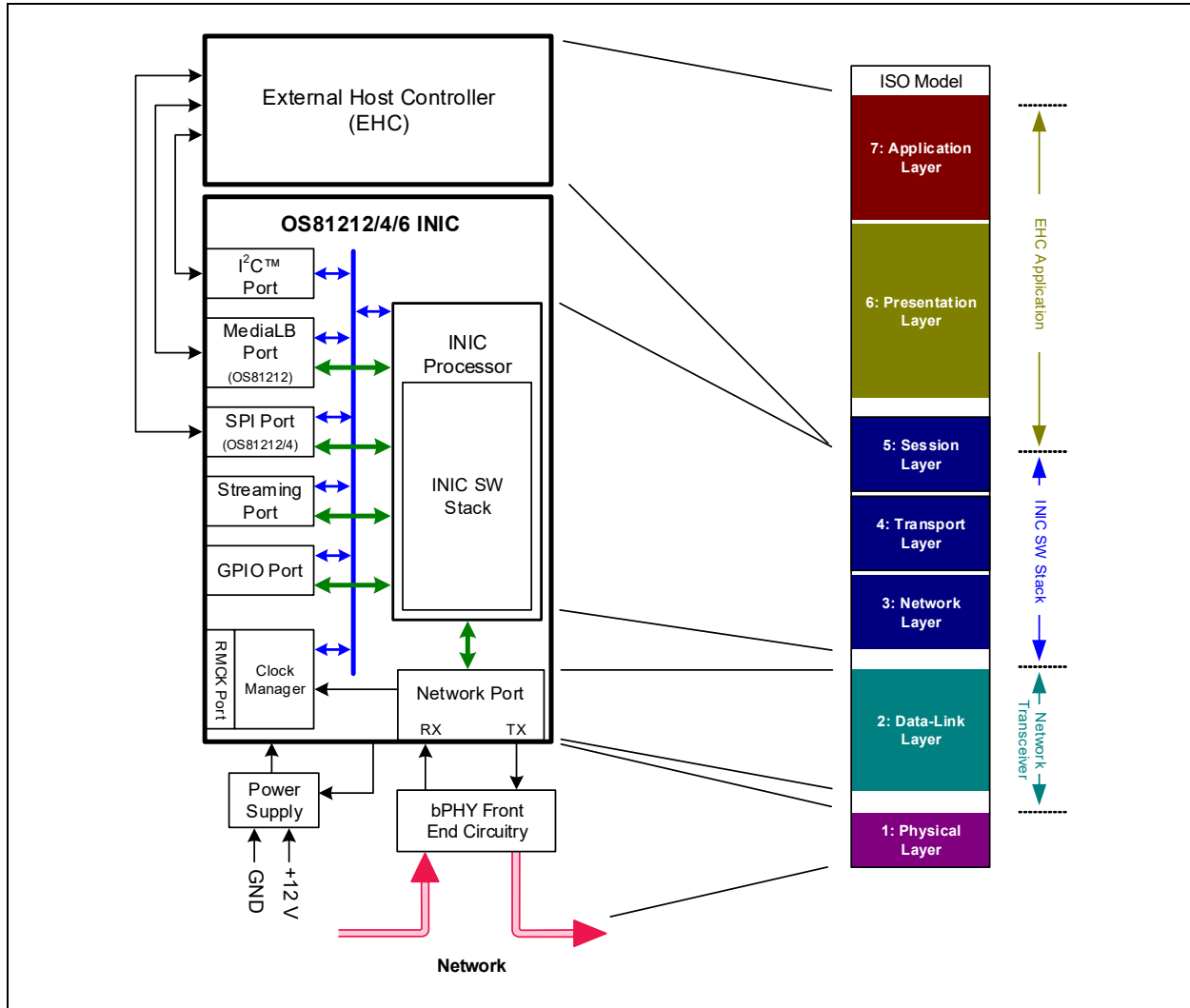
There are two major differences between a MOST compliant network and a UNICENS based network, besides meeting the requirements of the MOST Specification. First, the configuration of the network and routing of streaming data on a UNICENS network is done by one central node. In a MOST network, the configuration is distributed among the nodes, with each node being responsible for connecting to a network resource under the direction of the local EHC. Since UNICENS does this configuration under a single node (*Root Node*), it enables a new class of nodes, called Slim Nodes (*Remote Nodes*) that do not require a local processor. Second, the MOST Specification describes a dedicated control channel and a specific MOST message format for messages between applications. Whereas in a UNICENS network, it is assumed that application messages between nodes will be transmitted over the Ethernet channel in the form of IP messages using an OEM defined protocol such as SOME/IP, MQTT, or any other OEM defined protocol.

All relevant network management functions are handled on-chip, providing a complete system interface to the physical layer components. Minimal additional components are required due to the high-level of integration. An on-chip PLL with ultra-low jitter guarantees accurate audio and video transmission and clock recovery over a wide frequency range.

A typical network node consists of the physical layer devices connected to the network, the OS81212/4/6 to handle the low-level protocols, and an *External Host Controller* (EHC) for the mid- to high-level functions. This architecture eliminates the need for the user to implement the lower protocol levels required by the [ISO 21806 \(Parts 1-7\): 2020 – Road Vehicles – Media Oriented Systems Transport \(MOST\) Specifications \[1\]](#), the [ISO 21806 \(Parts 14-15\): 2021 – Road Vehicles – Media Oriented Systems Transport \(MOST\) Lean Application Layer Specification \[2\]](#), or the UNICENS stack, thereby drastically shrinking development time. Network management functions are off-loaded from the programmer, allowing full concentration on the application being developed. Depending on the embedded firmware, the functions and API of the low-level protocol can vary. Refer to the [INIC Interface Specification \[4\]](#) for more information. [Figure 3-1](#) illustrates the OS81212/4/6 with the protocol stack implementation.

# OS81212/4/6

FIGURE 3-1: INIC HARDWARE/SYSTEM OVERVIEW



## 3.1 Network

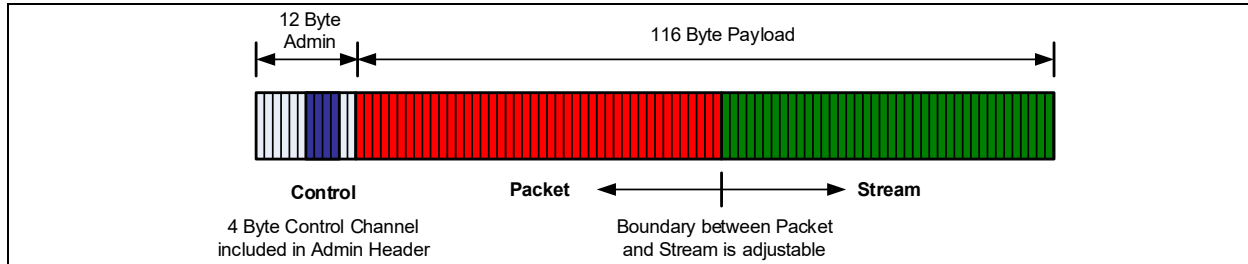
To minimize costs, the network supports a peer-to-peer methodology, eliminating excessive hardware overhead such as a hub (although hub-based architectures can also be implemented). In addition to handling network interface and communication management functions, the OS81212/4/6 INICs also handle all of the important low-level network management functions such as node position sensing (Plug-and-Play), start up, and shut down. Other features include error reporting, fail-safe operation, and channel allocation. Placing these features in hardware off-loads the EHC, allowing it to focus on higher-level network functions.

The network implements multiple simultaneous data transport methods operating across a single medium. Once bandwidth is allocated, each data transport method operates independently (not affecting the others), providing a robust, dependable, and deterministic system architecture. The OS81212/4/6 INICs support the following network data transport methods:

- Control - consists of a single *control channel*
- Asynchronous (Packet) - consists of a single *packet channel* (OS81212/4 only)
- Synchronous Streaming - consists of one or more *synchronous streaming channels*
- Isochronous Streaming - consists of one or more *isochronous channels*

The network frame's channel boundaries are shown in [Figure 3-2](#).

FIGURE 3-2: NETWORK FRAME CHANNELS



### 3.1.1 CONTROL

The INIC has a unique 4 byte (1.5 Mbit/s data rate) control channel shared by all network nodes. Applications use this channel to transmit control messages between nodes. These control messages from the EHC application are referred to as *Application Control Frames*. *Application Control Frames* are transported as ICMs, RCMs, or MCMs between an EHC driver and a INIC peripheral port (i.e. I<sup>2</sup>C, MediaLB, or SPI). *Application Control Frames* are then transported as *Control Frames* on the network's control channel. Any network node that needs to transmit a *Control Frame* on the network must arbitrate for the control channel. The arbitration is handled by INIC. UNICENS and NetServices libraries provide API commands to send and receive *Application Control Frames* which abstracts the internal INIC software stack from the application.

The INIC automatically monitors the control channel on the Network Port and uses information in the header to determine whether the node is the intended receiver of the *Control Frame*. When a *Control Frame* is to be received (e.g. node address match), it is buffered within the INIC and made available to the application via one of the OS81212/4/6 peripheral hardware ports that supports RCM or MCM traffic (see [Table 3-1](#)).

### 3.1.2 ASYNCHRONOUS (PACKET)

The INIC has a single asynchronous packet channel with an adjustable bandwidth that is shared by all network nodes. Applications use this channel to transmit large data packets (e.g. navigation maps) in bursts to other network nodes. These data packets from the EHC application are referred to as *MHP Frames* or *Ethernet Frames*. *MHP Frames* or *Ethernet Frames* are transported as MEPs (Meta Ethernet Packets) or MDPs (*Meta Data Packets*) from the EHC driver to the INIC peripheral port (MediaLB or SPI) where the message is then converted into a *Packet Frame* or *Ethernet Data Frame* to be sent across the network to the intended node. Any network node that needs to transmit a *Packet Frame* or *Ethernet Data Frame* can arbitrate for the packet channel. The arbitration is handled by INIC.

The INIC automatically monitors the packet channel on the Network Port and uses information in the *MHP Frame* or *Ethernet Frame* header to determine whether the node is the intended receiver (e.g. node address match). When a frame on the packet channel is to be received, the appropriate *MHP Frame* or *Ethernet Frame* is buffered within the INIC and made available as an MDP or MEP on one of the OS81212/4 peripheral ports that support MDP or MEP traffic (see [Table 3-1](#)).

### 3.1.3 SYNCHRONOUS STREAMING

The *Synchronous Streaming transport method* can consist of multiple synchronous streaming channels, each transporting raw, real-time synchronous data (e.g. audio, video). The data on a synchronous streaming channel is sourced by a single transmitting node that has been granted channel bandwidth. Synchronous streaming channels are broadcast, making the data available for reception by one or more sink nodes.

For high-speed synchronous data, INIC acts like a cross-point switch to connect synchronous streaming channels to sources/sinks attached to OS81212/4/6 hardware ports that support synchronous data (see [Table 3-1](#)).

### 3.1.4 ISOCHRONOUS STREAMING

On the OS81212, the *Isochronous Streaming transport method* can consist of multiple isochronous channels, each transporting data that is sourced by a single transmitting node (no arbitration) that has been granted channel bandwidth. Isochronous channels are broadcast, making the data available for reception by one or more sink nodes.

Streaming data is transported across isochronous channels, albeit in a *structured* format. Various streaming data structures can be transmitted across an isochronous channel, including:

- DiscreteFrame Isochronous (DFI) Streaming Phase
- A/V Packetized (AVP) Isochronous Streaming Packets
- Quality of Service (QoS) IP Streaming Packets

## 3.1.4.1 DiscreteFrame Isochronous Phase

A *DiscreteFrame Isochronous* (DFI) phase channel transports the *clock phase component* (also referred to as DFI phase), which includes the phase information necessary to regenerate a specific clock frequency, asynchronous to the network, on the sink node. DFI phase is typically used in association with A/V Packetized (AVP) isochronous to transfer video clock data from the source encoder to the sink decoder. OS81212 hardware ports that support the exchange of DFI phase are shown in [Table 3-1](#).

## 3.1.4.2 A/V Packetized (AVP) Isochronous Streaming Packets

*A/V Packetized* (AVP) isochronous streaming packets are used to transmit streaming data in an application-specific format (e.g. MPEG2-TS 188 bytes). Sink nodes access the data using a locally generated clock that is independent of the network frame rate, as well as the frame rate used by the node sourcing the AVP packets. OS81212 hardware ports that support the exchange of AVP packets are shown in [Table 3-1](#).

## 3.1.4.3 Quality of Service (QoS) IP Streaming Packets

An isochronous channel can be used to transport *Ethernet Data Frames* over *dedicated* network bandwidth. In such a system, the *Ethernet Data Frames* are sourced by a single transmitting node that has been granted channel bandwidth. Although the same *Ethernet Data Frames* could alternatively be transmitted across the packet channel, arbitration prior to each packet transmission would be required. *Ethernet Data Frames* exchanged across an isochronous channel are referred to as *QoS Frames*, and are intended for Internet Protocol (IP) message applications.

*QoS Frames* are transmitted in a streaming, broadcast fashion, which makes the *QoS Frame* stream available for reception by one or more sink nodes. However, unlike sink nodes receiving messages on the asynchronous packet channel, nodes that need to sink *QoS Frames* must first be properly configured by the application for reception on a specific channel.

Once the INIC is attached to a network channel that transmits *QoS Frames*, the destination address in the packet headers is ignored and the *QoS Frame* is always received.

Due to the streaming nature of this data transport, received packets are not buffered in their entirety within INIC; rather, an internal circular buffer is used to route the stream of *QoS Frames* to the OS81212 hardware port that supports and is configured for *QoS Frame* reception (see [Table 3-1](#)).

## 3.2 INIC Processor

The INIC Processor manages the transfer of data between the network and other hardware ports. The Network Port connects the INIC Processor to the network using the OS81212/4/6 network transmit and receive pins (**BTXP/BTXN** and **BRXP/BRXN**) to exchange data over the bPHY interface. The bPHY interface of the OS81212/4/6 requires external *Analog Front-End (AFE)* circuitry to connect the INIC Processor to the balanced media network and increase system immunity to EMI.

Working in conjunction with the Clock Manager, the Network Port recovers the network clock, decodes received data, and passes the data to the INIC Processor. The INIC Processor then routes data to the appropriate destinations on and off the chip.

The network frame to be transmitted by the OS81212/4/6 is constructed by the INIC Processor by combining incoming network data from the Network Port with outgoing data from other on-chip resources. The Network Port then encodes the data for network transmission.

The *INIC Software Stack* runs on the INIC Processor and configures the Network Port at power-up, which allows the network to become operational without any external protocol stack. This has been accomplished by placing all the lower layers of the ISO protocol stack inside of the device. [Table 3-1](#) illustrates the position of the *INIC Software Stack* within the ISO software model.

Each network node contains a general set of functions to manage the network. These functions are split with the INIC managing the low-level functions and the EHC managing the higher-level functions.

The INIC also contains functions used to configure the chip. This includes the setup of hardware ports and the usage of network bandwidth for transporting streaming data.

The upper levels of the network protocol must be implemented by an EHC that is capable of processing the user application with the minimal overhead imposed by the INIC API. During real-time operation, the EHC and the *INIC Software Stack* within the INIC Processor communicate using control messages via the I<sup>2</sup>C, SPI, or MediaLB Ports.

## 3.3 Hardware Ports

Each OS81212/4/6 hardware port transports different network data types. [Table 3-1](#) shows the data types that can be transported across each hardware interface.

**TABLE 3-1: HARDWARE PORT DATA TRANSPORT**

| <i>Peripheral Hardware Ports:</i> |         |          |          |                  |             |             |             |
|-----------------------------------|---------|----------|----------|------------------|-------------|-------------|-------------|
|                                   | ICM     | RCM, MCM | MEP, MDP | Synchronous Data | DFI Phase   | AVP Packets | QoS (MEP)   |
| MediaLB <sup>1</sup>              | X       | X        | X        | X                | X           | X           | X           |
| Streaming                         |         |          |          | X                |             |             |             |
| RMCK                              |         |          |          | X                |             |             |             |
| SPI <sup>2</sup>                  | X       | X        | X        |                  |             |             |             |
| I <sup>2</sup> C Port             | X       | X        |          |                  |             |             |             |
| <i>Hardware Port:</i>             |         |          |          |                  |             |             |             |
|                                   | Control | Control  | Packet   | Synchronous      | Isochronous | Isochronous | Isochronous |
| Network                           |         | X        | X        | X                | X           | X           | X           |

**Note 1:** OS81212 only.

**2:** OS81212/4 only.

Using the OS81212/4/6 API, a *socket* is created for each data stream transported across a hardware port. The specific data type to be exchanged is specified when creating the socket. Data routing between hardware ports is setup when sockets of like data types are linked together. For more information on sockets, refer to the [INIC Interface Specification \[4\]](#).

### 3.3.1 NETWORK PORT

The Network Port receiver is over-sampled at a high-frequency and data is recovered by a digital state machine. When the port is configured as a timing-slave, the OS81212/4/6 recovers the network clock, which the Clock Manager then uses to generate other internal clocks. When the port is configured as the timing-master, the internal clocks and the network clock are generated by the Clock Manager based on an external crystal (XTI/XTO). The Network Port transmitter works in conjunction with the INIC Processor to encode data for network transmission.

After reset, the OS81212/4/6 automatically interacts with the network, performing all necessary low-level network management functions. This enables the EHC to configure its interface to the OS81212/4/6 when the application is ready.

### 3.3.2 MediaLB PORT

**Note:** The MediaLB Port is only available on the OS81212.

The MediaLB Port supports communication between the EHC and the OS81212 INIC by means of the *Media Local Bus* (MediaLB) protocol. MediaLB provides a low cost, easy to implement hardware interface that standardizes and simplifies network application development. MediaLB has one bus master, referred to as the *MediaLB Controller* (or simply *Controller*), which is always the OS81212. All other connected components (including the EHC) are known as *MediaLB Devices* (or simply *Devices*). MediaLB Device functionality is a subset of Controller functionality; the Controller (OS81212) functions as a Device when it is receiving data from other connected components.

The OS81212 supports the MediaLB protocol with a single-ended MediaLB 3-pin interface. This interface supports the exchange of network data types, including: control messages, asynchronous packets, synchronous data, DFI phase, QoS packets, and AVP packets. The transmitting Device sends data on a dedicated data line, while information about the data is simultaneously transported on an independent signal information line.

# OS81212/4/6

---

When using the MediaLB 3-pin interface, the OS81212 MediaLB Port output clock is configurable as  $256 \times Fs$  (12.288 MHz at 48 kHz),  $512 \times Fs$  (24.576 MHz at 48 kHz), or  $1024 \times Fs$  (49.152 MHz at 48 kHz). When configured for  $256 \times Fs$ , the OS81212 can transmit and receive up to 28 bytes (7 quadlets) per frame of data. When configured for  $1024 \times Fs$ , the OS81212 can transmit and receive up to 124 bytes (31 quadlets) per frame of data.

MediaLB is a token-passing bus, where the Controller manages the token and generates the data clock. The Controller passes the token by sending out a *ChannelAddress* on the signal information line. The *ChannelAddresses* are pre-assigned by the board integrator and are associated with a specific transmitting Device and receiving Device. Once the transmitting Device associated with the *ChannelAddress* is granted bus access, it can send one quadlet of data on the MediaLB data line after a defined time delay. The Device provides information about the data being transmitted by simultaneously sending out a command on the signal information line. After receiving the *ChannelAddress*, the receiving Device must decide whether to receive the associated message or reject it. A status byte from the receiving Device is placed on the signal information line, one byte after the command from the sending Device is received. Once per network frame, the Controller generates a unique pattern (*FRAMESYNC*) on the signal information line. This defines the MediaLB frame edge, as well as the byte boundary of signal and data lines for all Devices.

Since MediaLB is a high-speed bus, connections to external devices should be implemented in hardware and not *bit-banged* on generic ports. A MediaLB Device interface, in the form of VHDL code, is available from Microchip. For more information about the Media Local Bus, refer to the [MediaLB Specification \[5\]](#).

### 3.3.3 STREAMING PORT

The Streaming Port provides a gateway for synchronous data exchange between the OS81212/4/6 and external legacy devices. The OS81212/4/6 Streaming Port supports four serial data pins sharing a common synchronization signal and bit clock. The data formats supported by the Streaming Port are compatible with industry-standard serial interfaces found on many ADCs, DACs and other devices.

### 3.3.4 RMCK PORT

The RMCK Port is a hardware interface controlled by the internal Clock Manager. This port provides a single clock output (**RMCK**) that is synchronous to the network frame rate ( $Fs$ ). The RMCK Port is typically used in conjunction with the Streaming Port to exchange synchronous data over the network. The **RMCK** output is used to synchronize external application devices (e.g. ADC or DAC) to the common network timebase.

### 3.3.5 SPI PORT

The *Serial Peripheral Interface* (SPI) Port supports the transmission of asynchronous and control packets over an interface that is compatible with many microprocessors, data converters and other devices. When the SPI Port is enabled, it operates as an SPI bus slave.

### 3.3.6 I<sup>2</sup>C PORT

Using the default *INIC Configuration String*, the I<sup>2</sup>C Port operates as a bus slave and supports control message (such as ICMS, RCMs, and MCMs) exchange. An interrupt pin (**INT**) is used to notify the EHC when the I<sup>2</sup>C Port requires service (e.g. control message received). While the EHC can communicate with the OS81212/4/6 over the I<sup>2</sup>C Port alone, the I<sup>2</sup>C Port data rate limits the throughput on the Application Layer. Application performance can be maximized by using either the MediaLB Port or SPI Port which also supports the exchange of control messages.

The OS81212/4/6 I<sup>2</sup>C Port may be configured as an I<sup>2</sup>C master by changing the settings in the *INIC Configuration String*. In master mode, the OS81212/4/6 performs read and write operations with on-board I<sup>2</sup>C slave devices based on remote commands received from the network. Slave interrupt signals may be connected to any available INIC GPIO pin. The *Root Node* application controlling the remote slave can then sense a change in the GPIO when the slave asserts its interrupt.

### 3.3.7 GPIO PORT

Depending on the *INIC Configuration String* settings and firmware loaded in OS81212/4/6, the *General Purpose Input/Output* (GPIO) Port may be available to support various external operations. Some of the OS81212/4/6 hardware ports may not be available when the GPIO Port is used.

### 3.3.8 PORT EXPANSION

Devices belonging to the Microchip *I/O Companion* (IOC) family can be used as companions to INIC to provide additional features and hardware ports to the application. When an IOC is used as a companion to the OS81212 INIC, the MediaLB bus is typically the medium used to exchange data between the two devices. Data exchanged over MediaLB is routed to IOC hardware interfaces on which external devices may reside (e.g. ADC, DSP, video displays, etc.).



The OS85654 IOC, as depicted in [Figure 3-3](#), is a high-performance routing engine that supports I/O expansion and provides a *Digital Transmission Content Protection* (DTCP) coprocessor. The primary function of the OS85654 is the routing of data streams between various hardware ports, while the integrated DTCP coprocessor provides optional encryption/decryption of data streams routed through the IOC device. Various services are available to support full *Authentication and Key Exchange* (AKE) implementation. For applications that do not require DTCP, the OS85656 IOC provides the same functionality as the OS85654, albeit without the DTCP coprocessor.

Applications supporting video benefit from the OS85621 IOC with its integrated low-latency H.264 CODEC, industry standard video I/O part, and DTCP coprocessor. See [Figure 3-4](#). Applications not requiring content protection may instead utilize the OS85623, which provides the same functionality as the OS85621 but does not include the DTCP coprocessor.

Contact Microchip for more information about the I/O Companion family of devices.

**FIGURE 3-3: INIC-IOC DTCP APPLICATION EXAMPLE**

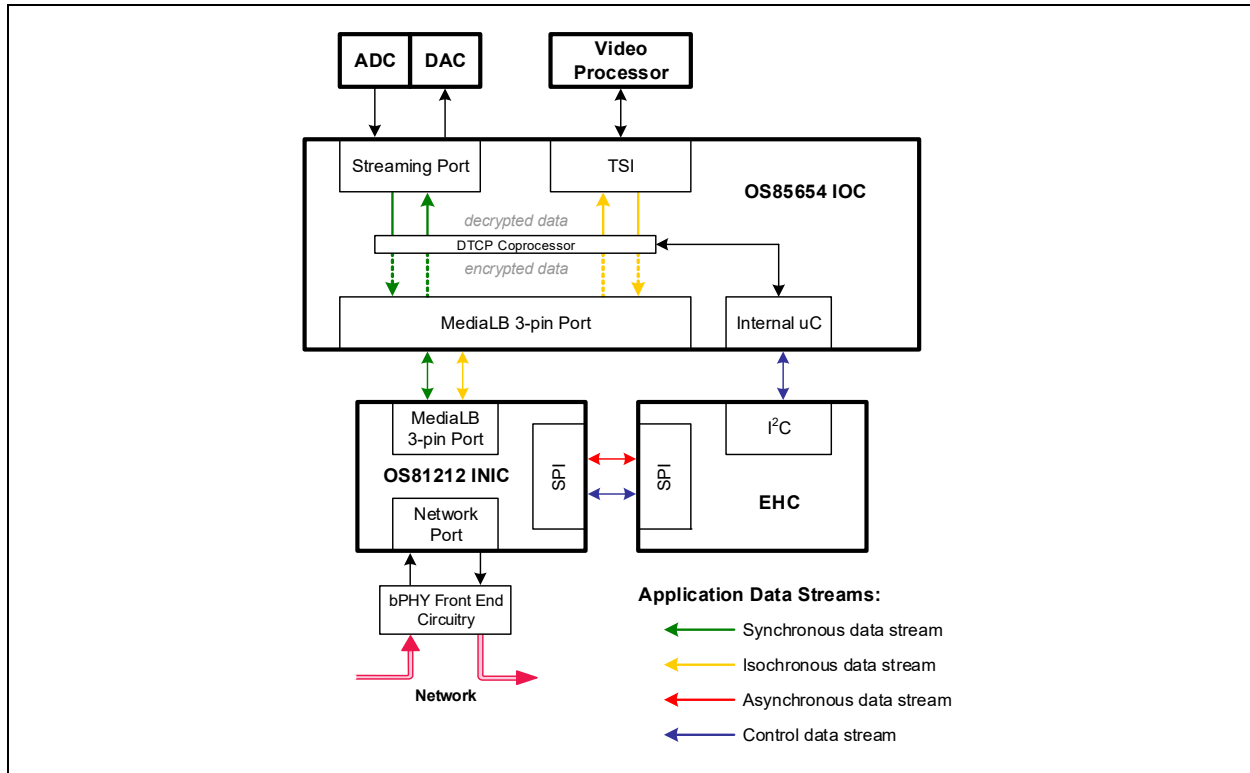
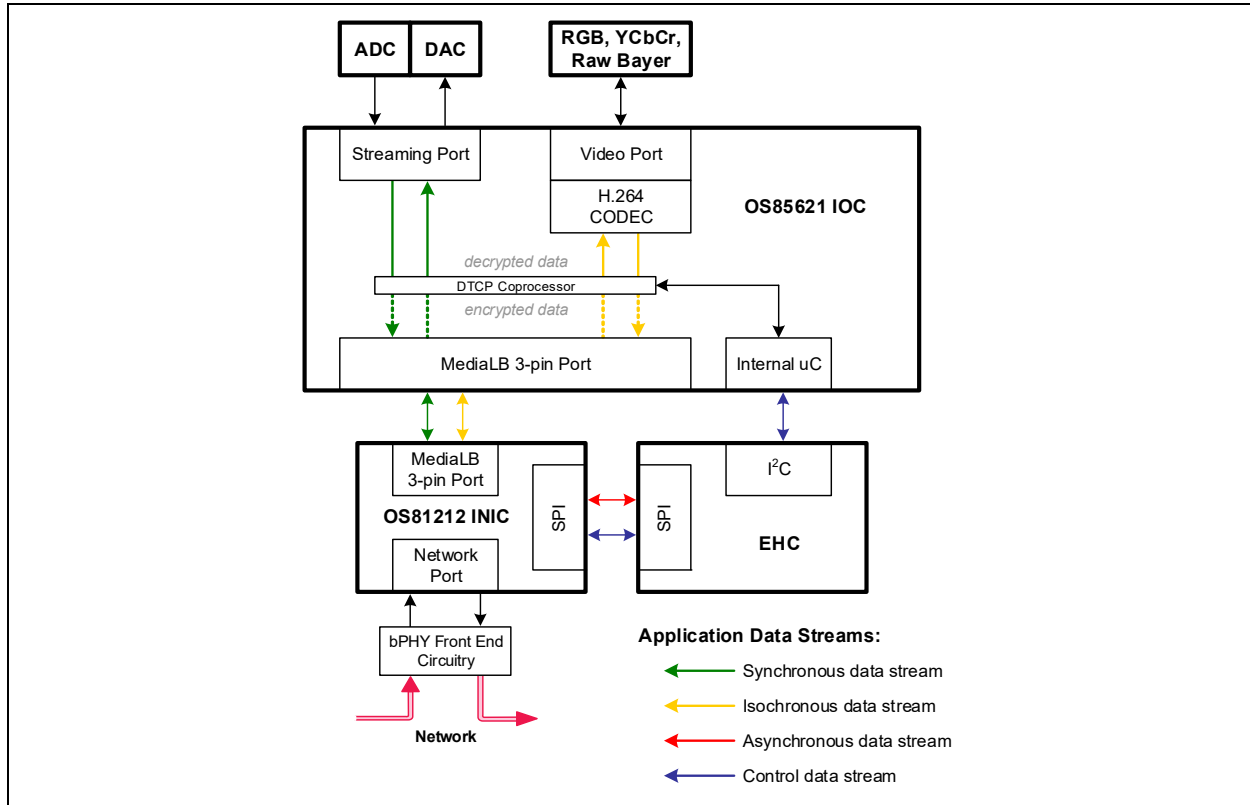


FIGURE 3-4: INIC-IOC VIDEO APPLICATION EXAMPLE



### 3.4 Centralized Network Diagnostics and Fallback Operation

The OS81212/4/6 network interface includes an embedded feature allowing the data path to reverse the direction of communication (receive on the **BTXP/BTXN** pins and transmit on the **BRXP/BRXN** pin). This reversal of direction enables two special modes of operation: isolation of cable faults if the normal ring is broken, and a special simplex mode of operation called Fallback Operation that can be used for fault-tolerant network operation.

#### 3.4.1 CENTRALIZED NETWORK DIAGNOSTICS

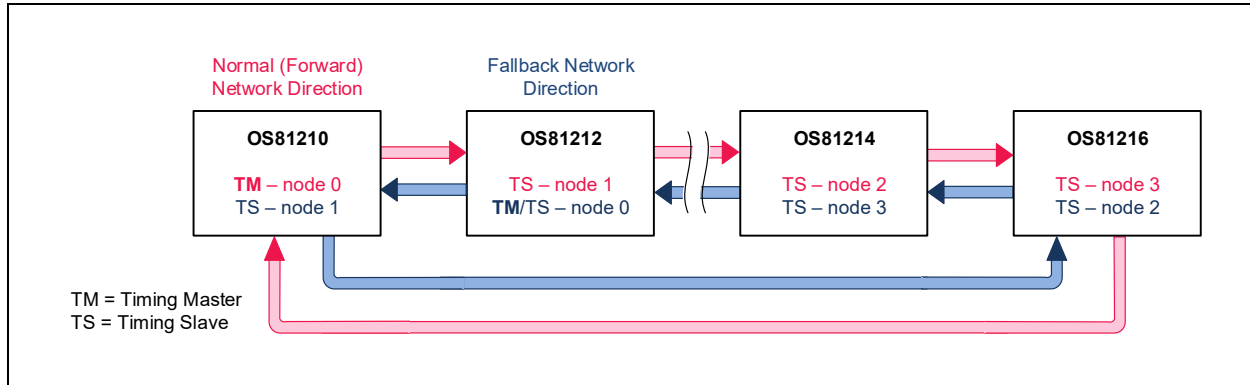
Centralized Network Diagnostics are used to collect diagnostic information from nodes in the network such as cable link connections in case normal operation is not possible due to a break in the ring architecture. These diagnostics are executed by the Timing Master (*Root Node*) over a control channel therefore requiring no additional cable wiring such as an electrical control line. All other nodes are *Remote Nodes* which do not require any additional hardware (EHC) or logic. For more information on triggering Central Network Diagnostics, please refer to the [INIC Interface Specification \[4\]](#).

#### 3.4.2 FALLBACK OPERATION

The Fallback Operation mode allows a segment of the network to operate in the reverse direction even if the ring is damaged. This feature can be used to implement an emergency call (eCall) feature (e.g. react to a network physical interruption) or to implement a simplex daisy chain network topology for simple microphone applications. This mode is intended for applications required to sustain synchronous streaming data and limited control communication during a fault state. In this mode, the direction of communication is reversed and a new node will negotiate to become the Timing Master. Nodes that are required to operate in this fault-tolerant mode must enable Fallback Operation in the *INIC Configuration String*. If enabled, further parameters must be set allowing nodes permission to become the new timing master or a timing slave, or react only as a timing slave during Fallback Operation. Additionally, these nodes can be configured to automatically stream synchronous data on pre-defined channels when Fallback Operation begins. This enables the transfer of streaming data with no application firmware required.

A normal forward network direction example is shown in [Figure 3-5](#). If the network is broken between nodes 1 and 2, Fallback Operation can be initiated and ultimately the furthestmost node upstream of the break allowed to become a Timing Master will take over (Forward network node 1 would become Fallback node 0). For more information on Fallback Operation, please see the [INIC Interface Specification \[4\]](#).

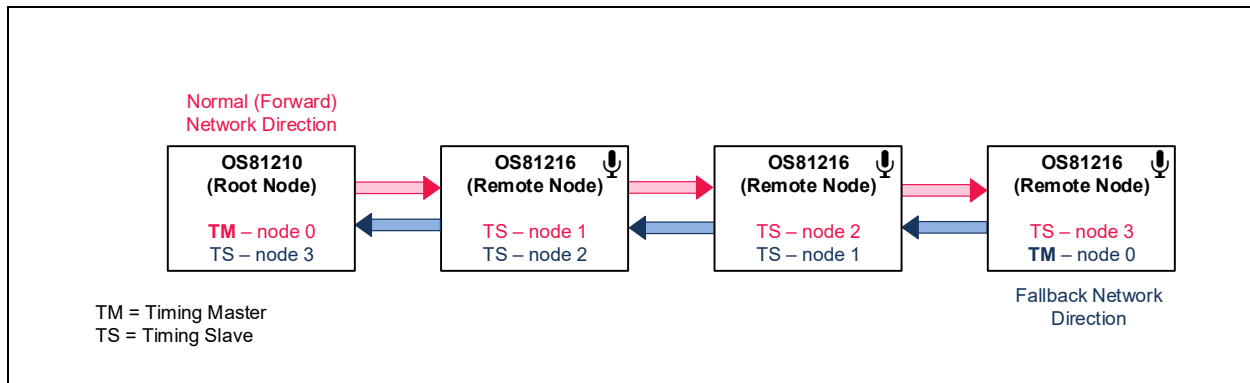
**FIGURE 3-5: eCall EXAMPLE**



Fallback Operation can also be used to implement a limited simplex daisy chain network which can automatically stream synchronous data in one direction. This feature is useful to create a simple microphone network. Only streaming data can be transmitted during the Fallback Operation mode. Each node must be statically configured via the *INIC Configuration String* to operate in Fallback Operation and use a predetermined channel to transmit streaming data. In [Figure 3-6](#), the *Root Node* (forward network direction timing master node 0) will start the network and begin the negotiation phase. The network direction is reversed, and the previous forward direction timing slave node 3 becomes the timing master node 0 in Fallback Operation. Once the negotiation timer has expired, streaming data will begin transmitting over each microphone's channel in the Fallback Network direction to the *Root Node*. This limited simplex daisy chain mode enables the creation of a completely statically defined network that operates without further configuration following a single startup command.

For more information on Fallback Operation, please see the [INIC Interface Specification \[4\]](#).

**FIGURE 3-6: SIMPLEX DAISY CHAIN MICROPHONE NETWORK EXAMPLE**



## 3.5 MOST NetServices

In addition to a generic automotive network, the OS81212/4/6 can be used to implement a MOST network. To accelerate development of MOST network applications that use the OS81212/4/6, Microchip offers the MOST NetServices API, which provides a seamless software interface between INIC and the EHC. The division of services between the OS81212/4/6 INIC and the EHC in a MOST network is illustrated in [Figure 3-1](#). A software library is available to provide all services that are relevant for:

- exchanging application data on a MOST network,
- managing hardware port data connections (e.g. sockets),
- managing high-level system tasks, and
- using control messages to access INIC for specific operations.

The MOST NetServices code incorporated into the EHC is divided into two distinct components: *Basic Services* and *Applications Socket*.

The *Basic Services* component includes message management, configuration management, and state control and supervision. It provides the facilities that enable direct communication with INIC through an interface to a low-level driver that is independent of the hardware port used. Formatted messages are sent and received by the low-level driver across the interface between the EHC and INIC.

The *Applications Socket* component resides on top of *Basic Services* and provides the functions required to implement the full FBlock `NetBlock` on the node. This component also contains a command interpreter, which provides a simple API for developing application FBlocks (e.g. FBlock `AudioAmp`).

MOST NetServices code is modular, allowing it to be customized for a particular application. Implemented in ANSI C, the MOST NetServices API can be adapted for individual requirements through configuration files.

With respect to the ISO communications model (shown in [Figure 3-1](#)):

- The OS81212/4/6 is connected to the bPHY *balanced media physical layer* with external passive front-end components.
- The OS81212/4/6 supports the *Data Link Layer* up to a portion of the *Session Layer*.
- The EHC must provide the remaining portion of the *Session Layer*, the *Presentation Layer*, and the *Application Layer*. When the MOST NetServices code is integrated:
  - the *Basic Services* component provides the remaining portion of the *Session Layer*, and
  - the *Applications Socket* component provides the *Presentation Layer* and a portion of the *Application Layer*.

The [INIC Interface Specification \[4\]](#) defines the FBlocks (and all associated functions) that are supported by the OS81212/4/6.

## 3.6 Unified Centralized Network Stack (UNICENS)

The *Unified Centralized Network Stack* (UNICENS) is an alternative to NetServices simplifying the handling of a complex, heterogeneous infotainment network. While NetServices is required to be implemented and running on every node of the system, UNICENS only needs to be implemented on a single *Root Node*. UNICENS provides centralized control and management for the entire network.

The network may be either statically or dynamically configured by UNICENS. In a static configuration, UNICENS scans the network on startup to discover the nodes, remotely configuring and controlling each node via *Control Frames*. A Remote Host Controller (RHC) may be implemented at a *Remote Node*, but is not required thereby significantly reducing hardware cost. This allows the network software to be centralized into the *Root Node* EHC, reducing system complexity and development cycles. UNICENS controls hardware at the *Remote Nodes* through remote I<sup>2</sup>C messages and GPIO control.

The OS81210 is ideal for implementation in the *Root Node* with its integrated USB Port. The OS81212, OS81214, and OS81216 are optimized for use in *Remote Nodes* with or without a Remote Host Controller (RHC).

More information regarding UNICENS may be obtained upon request from Microchip.

## 3.7 Operating System Drivers

Drivers are available which aid integration of the INIC with operating systems being used in many complex systems. For example, Microchip has made an INIC device driver available within the Linux<sup>®</sup>1 mainline kernel.

The Linux driver for INIC implements each of the network data types presenting them to the application in standard programming interfaces. Synchronous streaming channels are implemented as Advanced Linux Sound Architecture (ALSA) hardware and appear as standard audio sources and sinks. A/V Packetized data channels are accessed using the Video4Linux2 (V4L2) application framework. The network Ethernet channel appears as a standard Ethernet interface. Control channels are accessed as a standard character device. The standard presentation of network data channels greatly simplifies application programming effort.

Contact Microchip for more information regarding the INIC driver, including the Linux kernel driver for INIC.

---

1. Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the U.S. and other countries.

## 4.0 INIC PROCESSOR

The OS81212/4/6 is a valid network node even without an attached EHC, due to the on-chip INIC Processor. The *INIC Software Stack*, running on the INIC Processor, provides a message-based interface that is easily implemented in a high-level language. This message-based interface not only simplifies the interface between the application and the network, but allows the EHC software to be completely driven by message events. Using the INIC API, as described in the [INIC Interface Specification \[4\]](#), the EHC has access to the same functionality that would be available on a register-based interface, but the programmer is spared the effort of making software align with the chip architecture. Much of the INIC application interface is common across INIC chips, minimizing the effort of upgrading the software for different application platforms. The power-up default settings of the INIC Processor, such as timing-master/timing-slave functionality and other configuration options, can be changed through the *Customer Configuration Interface* (see [Section 15.6 “Configuration and Debug”](#)).

Partitioning the node architecture between INIC and the EHC results in an efficient implementation, allowing INIC to respond to many network events without intervention by the EHC. Network initialization is optimized, not requiring the network transceiver at each node to wait for the Application Layer to respond. Application software runs efficiently since the number of interrupts to the EHC is minimized. Interaction between the EHC and INIC can be further reduced if the EHC enables notification for INIC properties.

### 4.1 Data Exchange

General information on data exchange with the OS81212/4/6 is given below. Refer to the [INIC Interface Specification \[4\]](#) for more detailed information. The INIC Message Interface consists of two virtual paths - the Configuration Interface which handles message to control INIC itself and the Application Interface which is used to pass application specific messages between nodes.

#### 4.1.1 CONFIGURATION DATA

The OS81212/4/6 Configuration Interface is used for *message-based* data exchange between INIC and the EHC.

These messages transferred between the local INIC and the EHC via the MediaLB Port, SPI Port, or I<sup>2</sup>C Port are called *Application Control Frames*. *Application Control Frames* can be routed to the local INIC for control and configuration, or through the INIC and across the network for controlling and configuring remote INIC nodes.

*Application Control Frames* exchanged between the local EHC and the internal *INIC Software Stack* are called *INIC Control Messages* (ICMs). *Remote Control Messages* (RCMs) are similar to ICMs with the exception that they are routed through the local INIC to remote INIC nodes for configuration and network management purposes.

All *Application Control Frames* transferred up or down the *INIC Software Stack* are exchanged using the second generation Port Message Protocol (PMPv2). The message format (defined as the *Port Message*) is the same, regardless of the physical hardware port used. For more information on PMPv2, refer to the [Port Message Protocol Version 2 User's Guide \[6\]](#).

All control messages require a status message response from the receiving device. This status message indicates the OS81212/4/6 received the message successfully. Similarly, when the OS81212/4/6 sends a control message to the EHC (from the network or as a response to a previous request), the EHC is required to respond with a status message indicating message reception was successful. INIC provides several mechanisms (e.g. idle, buffer timer, watchdog, etc.) to monitor the *Port Message* transfer during a predefined time frame. Refer to the [INIC Interface Specification \[4\]](#) for more information on configuring these mechanisms.

#### 4.1.2 APPLICATION DATA

The OS81212/4 INIC supports multiple data formats and channels to send messages between applications on different nodes (e.g. a volume command between a head unit and amp module). *Ethernet Data Frames* can be used over the packet channel to transmit high bandwidth data between nodes using standard Ethernet formats such as TCP/IP or UDP. This allows the use of standardized protocols such as SOME/IP or MQTT to be used at the application level. The Ethernet frames are transferred into INIC via the MediaLB or SPI Port.

If the high bandwidth or standardized Ethernet protocols provided by the packet channel is not needed, it is possible to use the Application Interface to send user defined application specific control messages between nodes via the I<sup>2</sup>C, MediaLB or SPI Port. These messages transported over the INIC control channel are called *Meta Control Messages* (MCMs).

*Quality of Service* (QoS) packets are handled like real-time data on a synchronous streaming network channel. QoS packet headers are ignored by the INIC and all channel data is directly routed in a continuous fashion to the appropriate INIC hardware port, bypassing the INIC Message Interface.

## 5.0 I<sup>2</sup>C PORT

The I<sup>2</sup>C Port supports the I<sup>2</sup>C protocol and is processed in bytes by the internal firmware of the OS81212/4/6. The SCL pin clocks data in and out; SDA is a bi-directional data pin in which data is transferred MSB-first. When the OS81212/4/6 I<sup>2</sup>C Port is configured as an I<sup>2</sup>C slave device, it must be managed externally. The I<sup>2</sup>C Port is configured as an I<sup>2</sup>C slave in the default *INIC Configuration String*, however, the power-up defaults can be changed through the *Customer Configuration Interface* (see Section 15.6 “Configuration and Debug”). If the I<sup>2</sup>C Port is not configured as an I<sup>2</sup>C master by default at power-up (or reset), it can also be configured as an I<sup>2</sup>C master through the INIC API.

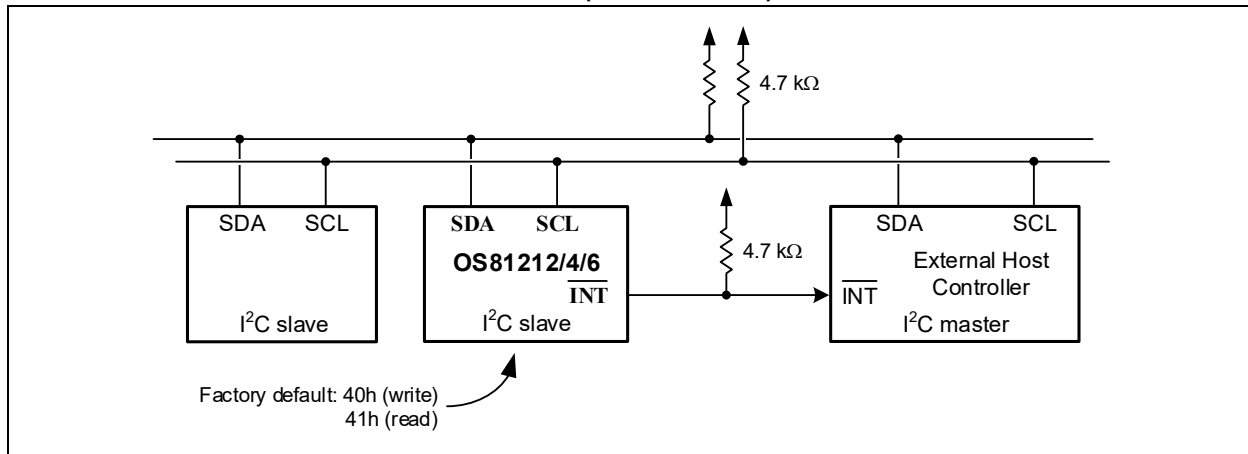
**Note:** When used in I<sup>2</sup>C master mode, the I<sup>2</sup>C Port is unavailable for use as an I<sup>2</sup>C slave device and cannot be used for Port Message exchange with an EHC.

### 5.1 Slave Operation

As a slave, the I<sup>2</sup>C Port supports communication of control messages between the OS81212/4/6 and the EHC. The control messages may be used to configure and control INIC, target local INIC FBlocks (ICMs), or be routed through the INIC between the EHC and the network control channel (RCMs and MCMs). The control messages are encapsulated within *Port Messages* that are transferred over the I<sup>2</sup>C bus. The first two bytes of the *Port Message* contains the *Port Message Length* (PML) which is the number of bytes remaining in the message. Refer to the [INIC Interface Specification \[4\]](#) for more detailed information.

Figure 5-1 illustrates the OS81212/4/6 as a slave in an I<sup>2</sup>C environment.

**FIGURE 5-1: I<sup>2</sup>C PORT PIN CONNECTIONS (SLAVE MODE)**



When the I<sup>2</sup>C Port is enabled as a slave, the  $\overline{\text{INT}}$  pin is used by the INIC to signal the communication status to the EHC I<sup>2</sup>C bus master, eliminating the need for a polling by the EHC. The  $\overline{\text{INT}}$  pin is an open-drain, active low output that can be wire-OR'ed with the interrupt outputs of other devices into the EHC's interrupt input. INIC drives  $\overline{\text{INT}}$  low to inform the EHC when service through the port is required (e.g. message available for reading). Following reset, the EHC should wait for the first falling edge of  $\overline{\text{INT}}$ , which occurs when initialization is complete. The first *Port Message* available for reading through the I<sup>2</sup>C Port indicates INIC is ready to receive messages and the EHC should initiate synchronization of the OS81212/4/6 as described in the [INIC Interface Specification \[4\]](#).

The first byte of an I<sup>2</sup>C transaction is the bus address plus a read/write bit ( $\overline{\text{R/W}}$ ) that determines whether the EHC is reading or writing *Port Messages*. The OS81212/4/6 factory default I<sup>2</sup>C address is 41h for read, and 40h for write; however, using the *Customer Configuration Interface* (Section 15.6 “Configuration and Debug”), the default I<sup>2</sup>C address can be changed in the *INIC Configuration String*.

The INIC processing of I<sup>2</sup>C transactions varies based on the overall loading of internal tasks. Clock stretching is a handshaking mechanism implemented in the I<sup>2</sup>C Port as it provides an appropriate mechanism to adapt the data transfer rate dynamically, thereby maximizing the transfer rate at any given time. If the OS81212/4/6 cannot keep up with a message, it stretches the clock following the eighth data bit or the acknowledge bit; therefore, the I<sup>2</sup>C bus master must monitor SCL when communicating with INIC.

# OS81212/4/6

If the OS81212/4/6 is busy supporting higher priority tasks, it will NACK the I<sup>2</sup>C address, indicating that the master should retry the message at a later time. I<sup>2</sup>C NACK responses occur at the address byte boundary (e.g. following the R/W bit).

The I<sup>2</sup>C Port does not support repeated START conditions in slave mode. Therefore, the master must terminate each transaction with a STOP condition before the next transaction can be started.

For more information on the I<sup>2</sup>C protocol, refer to the [I2C-Bus Specification \[7\]](#).

## 5.1.1 WRITING TO THE I<sup>2</sup>C PORT

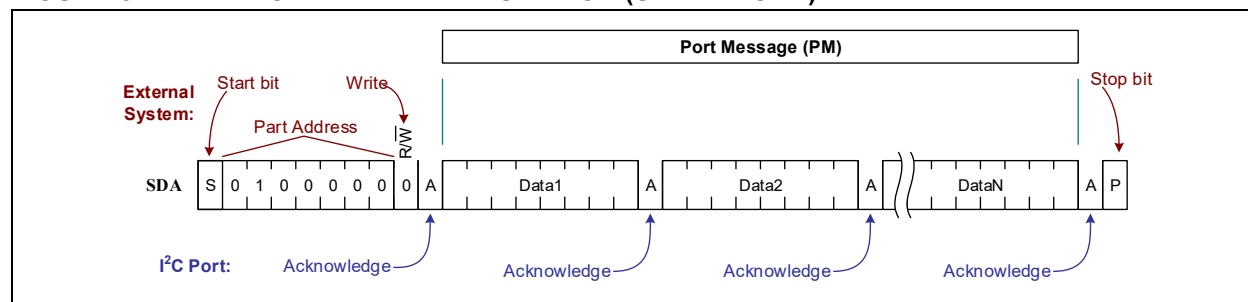
The transmission of a *Port Message* to the I<sup>2</sup>C Port is accomplished by performing an I<sup>2</sup>C write transaction. The beginning of a write transaction is marked by a START condition. The first byte transmitted specifies the chip address, as well as whether the transaction is a read or a write. The OS81212/4/6 address (factory default of 0100000b) occupies the upper seven bits of the first byte; the R/W bit is the LSB.

When the I<sup>2</sup>C Port receives an address byte of 40h (factory default OS81212/4/6 address and R/W bit clear), it acknowledges reception of the byte through an acknowledge bit and the *Port Message* is then written into the I<sup>2</sup>C Port. If the I<sup>2</sup>C Port does not acknowledge the address byte, the master should halt the write transaction and retry the *Port Message* later. The *Port Message* format is described in the [INIC Interface Specification \[4\]](#).

Sending messages to the INIC does not involve the  $\overline{\text{INT}}$  pin.

[Figure 5-2](#) illustrates the I<sup>2</sup>C write transaction. The characters “S” and “P” represent the start and stop conditions for messages.

**FIGURE 5-2: I<sup>2</sup>C PORT WRITE TRANSACTION (SLAVE MODE)**



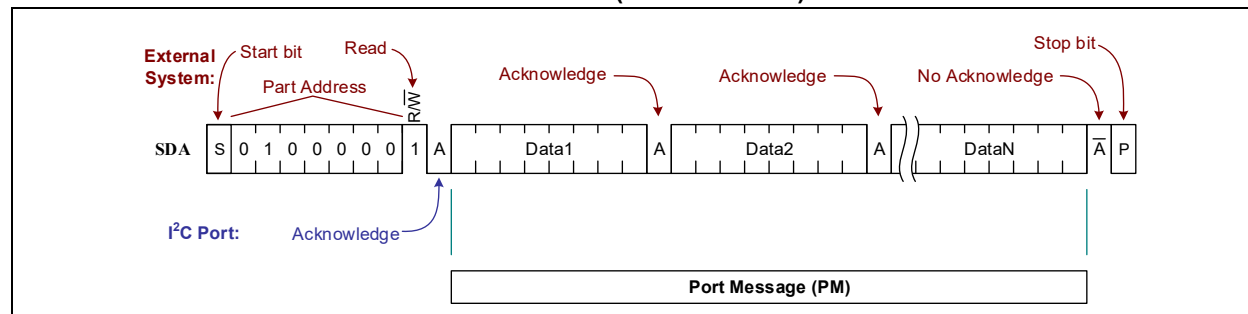
## 5.1.2 READING FROM THE I<sup>2</sup>C PORT

Reception of a *Port Message* from the I<sup>2</sup>C Port is accomplished by performing an I<sup>2</sup>C read transaction. Each read transaction is marked by a START condition. The first byte specifies the chip address, as well as whether the transaction is a read or a write. The OS81212/4/6 address (factory default of 0100000b) occupies the upper seven bits of the first byte; the R/W bit is the LSB.

When the I<sup>2</sup>C Port receives the address byte of 41h (factory default OS81212/4/6 address and R/W bit set), it acknowledges reception of the byte through an acknowledge bit and the *Port Message* is read from the I<sup>2</sup>C Port. If the I<sup>2</sup>C Port does not acknowledge the address byte, the master should halt the read transaction and retry the transaction later. The *Port Message* format is described in the [INIC Interface Specification \[4\]](#)

[Figure 5-3](#) illustrates the I<sup>2</sup>C transmission read transaction. The characters “S” and “P” represent the start and stop conditions for messages.

**FIGURE 5-3: I<sup>2</sup>C PORT READ TRANSACTION (SLAVE MODE)**





The  $\overline{\text{INT}}$  pin being asserted low indicates that at least one *Port Message* is available for reading. Once a message starts being read, the  $\overline{\text{INT}}$  pin is de-asserted. If additional messages are pending (waiting to be read), the  $\overline{\text{INT}}$  pin is asserted low again. This pending interrupt can occur as soon as the last byte of the current message is read.

Message availability can also be polled by the EHC without using the  $\overline{\text{INT}}$  pin interrupt mechanism. The *Port Message Length* (PML) value inside the first two bytes of the *Port Message* indicates if a message is pending. The EHC initiates a read transaction and reads the PML value. If the read PML value is zero, no message is available to be read from the INIC. However, such a polling mechanism will keep the bus busy, hence preventing other transactions from being processed.

In each read transaction, only one message can be fetched from the I<sup>2</sup>C Port. Only after the entire message is read will the INIC remove the message from its queue. If a message is not fully read (due to a STOP or repeated START condition received before the end of the message),  $\overline{\text{INT}}$  is reasserted low and the next read command starts reading from the beginning of the disrupted message, not the position where the read was previously terminated. In addition, if more bytes are read than the message length specified by the *Port Message Length* field or if no messages are queues, the INIC will return zero (00h) values in the read result.

Ideally, the EHC reads the entire length of the *Port Message* in a single I<sup>2</sup>C Port read transaction. To accomplish this, the EHC I<sup>2</sup>C driver must be capable of pausing the transaction after reading the PML value. During this time, the EHC can allocate the necessary memory and proceed with reading the remaining bytes of the message. For EHC I<sup>2</sup>C drivers that are not able to decipher the message length during a single read transaction, other methods of reading the *Port Message* are feasible, including the following:

- Example 1: The EHC can issue two consecutive I<sup>2</sup>C Port read transactions. The first transaction reads the length field of the *Port Message*. Based on the *Port Message Length* obtained in the first read transaction, a second transaction reads the entire *Port Message*.
- Example 2: The EHC can always read the maximum length of a *Port Message* in a single read transaction, regardless of the actual message length. Once the *Port Message* is read, application software can later determine the relevant portion of the read data. The INIC returns 00h data when a read transaction extends beyond the actual length of the *Port Message*. However, this will result in very high bus traffic when reading the remaining 00h data following the end of the *Port Message*. This method is inefficient and therefore should only be chosen if the other methods cannot be used.

## 5.2 Master Operation

The OS81212/4/6 I<sup>2</sup>C Port may be configured as a bus master to support applications requiring remote control of local I<sup>2</sup>C slaves through the network. As an I<sup>2</sup>C master, the OS81212/4/6 performs read and write operations with local I<sup>2</sup>C slave devices based on INIC API commands.

When operating as an I<sup>2</sup>C bus master, two settings for I<sup>2</sup>C transaction speed are available. The SCL clock frequency can be set to either 100 kHz (Standard-mode) or 400 kHz (Fast-mode). Faster speed modes as defined in the [I2C-Bus Specification \[7\]](#) (e.g. Fast-mode plus, High-speed mode) are not supported by the OS81212/4/6. The speed setting is selected when the port is created.

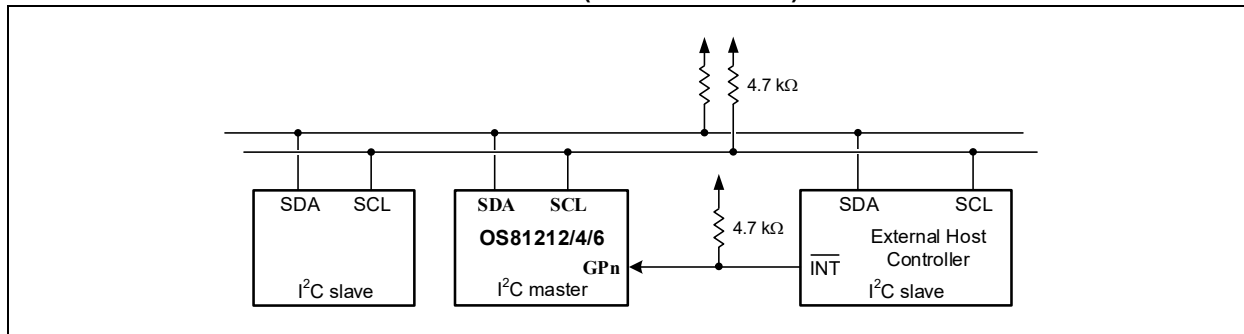
The OS81212/4/6 supports both the standard 7-bit device addressing and enhanced 10-bit I<sup>2</sup>C addressing modes. Standard 7-bit device addressing is most commonly used by slaves. The 10-bit addressing mode expands the number of possible device addresses while maintaining backwards-compatibility with devices that only support 7-bit addressing. Per the [I2C-Bus Specification \[7\]](#), a '11110' following a START condition indicates the use of enhanced 10-bit addressing. For 10-bit addressing, the first byte contains the two most significant bits of the 10-bit device address as well as the R/W flag. The remaining eight bits of the address are sent in the second byte. The OS81212/4/6 supports full interoperability for systems with mixed 7-bit and 10-bit slave device addresses.

As described in the [I2C-Bus Specification \[7\]](#), multi-master configurations implement an arbitration mechanism to allow more than one master to control the bus at the same time without corrupting data exchange. The OS81212/4/6 supports multi-master I<sup>2</sup>C configurations and implements the necessary bus-busy-state detection and bus arbitration logic to prevent data collision during read/write transfers inside a multi-master environment.

[Figure 5-4](#) illustrates the OS81212/4/6 as a master in an I<sup>2</sup>C environment.

# OS81212/4/6

**FIGURE 5-4: I<sup>2</sup>C PORT PIN CONNECTIONS (MASTER MODE)**



Clock stretching is a handshaking mechanism used by slave devices to adapt the data transfer rate dynamically. A slave device is permitted to hold SCL low after receiving (or sending) a bit to indicate it is not yet ready to process additional data. To ensure compatibility with all slave devices, the OS81212/4/6 master supports clock stretching at the bit and byte boundary.

Some I<sup>2</sup>C slave devices implement an interrupt pin to indicate to the I<sup>2</sup>C master that service is needed. The INIC GPIO Port can be configured and remotely monitored to support devices that implement interrupt functionality. Refer to [Chapter 12.0 “General Purpose Input/Output \(GPIO\) Port”](#) for more information.

Once master mode is enabled, remote control messages (*Control Frames*) are used by a remote node on the MOST network to initiate I<sup>2</sup>C read and write transactions on the local bus. The read/write methods specify the transfer mode, the slave target address, and the number of bytes to be read/written. The maximum block size for a read/write transaction is 32 bytes of data. A successful read transaction returns the device address, the number of bytes read, and the list of data bytes. When a write transaction successfully finishes, the device address and number of bytes written is returned.

Refer to the [INIC Interface Specification \[4\]](#) for the function reference and more information on using the OS81212/4/6 I<sup>2</sup>C Port in master mode.

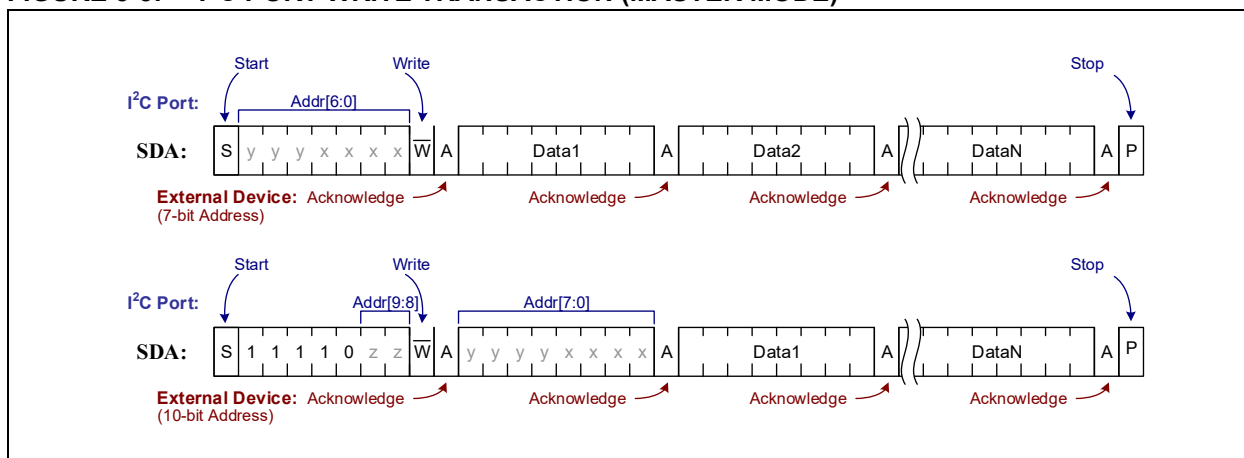
## 5.2.1 WRITING TO AN I<sup>2</sup>C SLAVE DEVICE

Per the [I<sup>2</sup>C-Bus Specification \[7\]](#) a '11110' following a START condition indicates use of the 10-bit addressing scheme. For a write operation, the first byte contains the two most significant bits of the 10-bit device address plus the R/W bit clear. Following acknowledgment from the external device, the remaining eight address bits are sent in the second byte. Message payload (*Data1* through *DataN*) begins in the third byte of the message for 10-bit addressing.

When using 10-bit addressing, the second address byte is actually sent as the first data byte. Therefore, if the message payload contains *n* data bytes, the total length of the write transaction must be set to *n+1*.

Figure 5-5 illustrates a write transaction for both 7-bit and 10-bit addressing modes.

**FIGURE 5-5: I<sup>2</sup>C PORT WRITE TRANSACTION (MASTER MODE)**

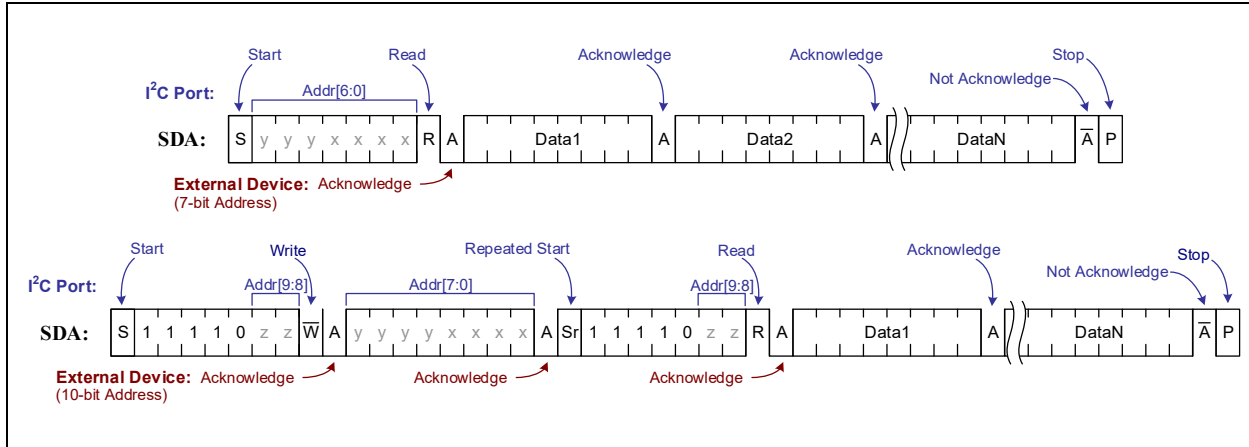


## 5.2.2 READING FROM AN I<sup>2</sup>C SLAVE DEVICE

Read transactions on slave devices with 10-bit addresses require a Repeated START condition. Refer to the [I<sup>2</sup>C-Bus Specification \[7\]](#) for more information. The master first fully addresses the 10-bit slave device with a write transaction (R/W bit clear) and then issues a Repeated START condition. After the Repeated START, the direction is changed and the master completes the read transaction with an abbreviated version (only the two most significant bits) of the address and the R/W bit set.

Figure 5-6 illustrates a read transaction for both 7-bit and 10-bit addressing modes.

**FIGURE 5-6: I<sup>2</sup>C PORT READ TRANSACTION (MASTER MODE)**

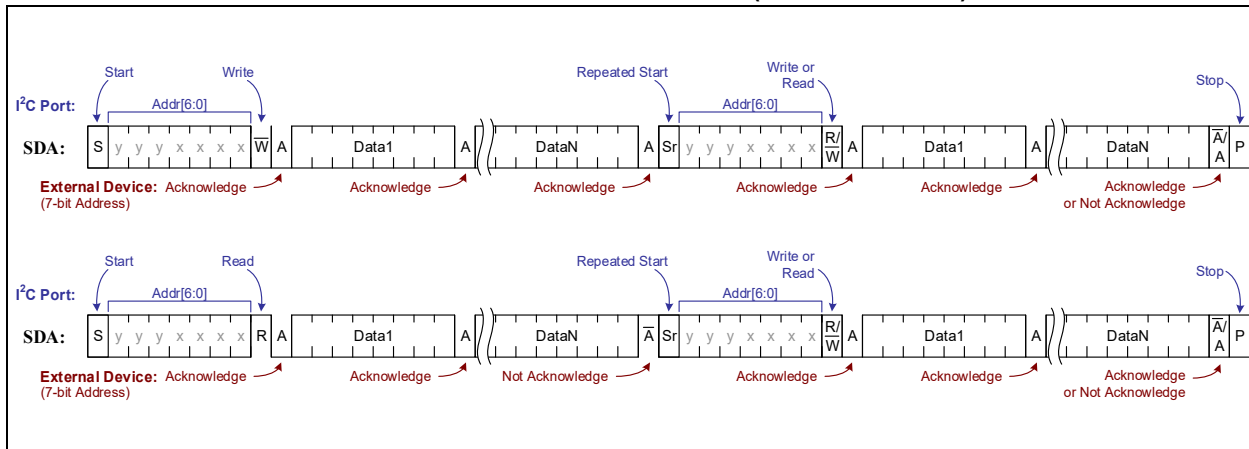


## 5.2.3 REPEATED STARTS ON AN I<sup>2</sup>C SLAVE DEVICE

A repeated START mode is available for both Read and Write transactions to a single I<sup>2</sup>C slave in both 7-bit and 10-bit addressing. The I<sup>2</sup>C master can issue any combination of subsequent writes or reads to an I<sup>2</sup>C slave using a repeated START condition in place of the previous transaction's STOP condition.

Figure 5-7 illustrates a repeated START following a write or a read transaction in 7-bit addressing mode.

**FIGURE 5-7: I<sup>2</sup>C PORT REPEATED WRITES AND READS (MASTER MODE)**



## 6.0 MediaLB PORT

**Note:** The MediaLB Port is only available on the OS81212.

A *Media Local Bus* (MediaLB) Port is available on the OS81212, supporting a MediaLB 3-pin interface. When enabled, the MediaLB Port provides the EHC access to network data types including: control messages, asynchronous packets, synchronous data, DFI phase, Quality of Service (QoS) IP streaming packets, and A/V Packetized (AVP) isochronous streaming packets.

The MediaLB Port supports communication of control messages between the OS81212 and the EHC. The control messages may be used to configure and control INIC (ICMs), target the local INIC FBlocks (ICMs), or be routed by INIC between the EHC and the network control channel (RCMs and MCMs). Refer to the [INIC Interface Specification \[4\]](#) for more detailed information.

The MediaLB Port also allows access to the internal functions of the OS81212, eliminating the need for I<sup>2</sup>C Port access. With MediaLB, only one interface is needed for data and control. When combined with the MediaLB software API library, the system designer is insulated from much of the complexity of managing the interface between the application software and the network. When MediaLB is configured as the configuration or application interface, the first PMPv2 status message sent to the EHC after power-up/reset indicates that the OS81212 is initialized and ready to communicate with the EHC. In common applications, this message is used to instruct the EHC to synchronize and initialize communication. Additionally, the EHC may initiate synchronization of the OS81212, as defined in the [INIC Interface Specification \[4\]](#). This ensures that OS81212 and the EHC can synchronize communication under various circumstances.

The MediaLB Port is disabled in the default *INIC Configuration String*, however, the power-up defaults can be changed through the *Customer Configuration Interface* (see [Section 15.6 “Configuration and Debug”](#)). If the MediaLB Port is not enabled by default at power-up (or reset), it can be enabled and configured through the local INIC FBlocks.

Data is transmitted onto the network most significant byte (MSB) first. To comply with the [ISO 21806 \(Parts 1-7\): 2020 – Road Vehicles – Media Oriented Systems Transport \(MOST\) Specifications \[1\]](#) and the [ISO 21806 \(Parts 14-15\): 2021 – Road Vehicles – Media Oriented Systems Transport \(MOST\) Lean Application Layer Specification \[2\]](#), when transmitting multi-byte words, the most significant byte should be transmitted first. In addition, when transmitting stereo data, left should be sent before right.

### 6.1 MediaLB 3-pin Physical Layer

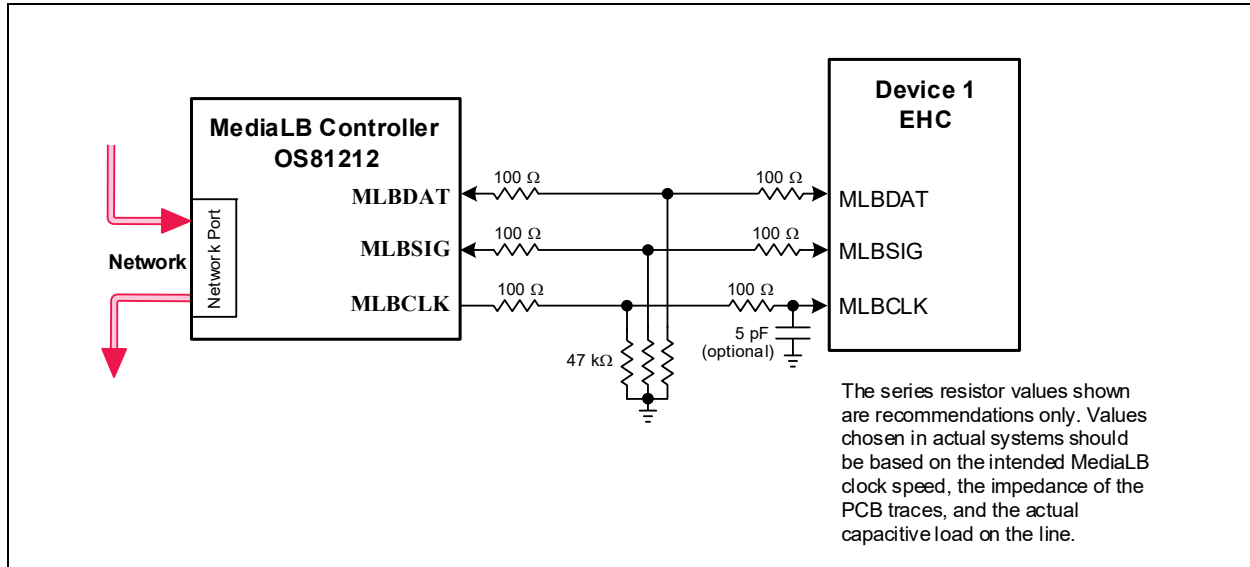
The MediaLB 3-pin physical layer is a single-ended bus consisting of:

- **MLBCLK** - MediaLB Controller output signal that clocks data in and out of the MediaLB Port
- **MLBSIG** - bi-directional signal that transports signal information to and from the MediaLB Port
- **MLBDAT** - bi-directional signal that transports data to and from the MediaLB Port

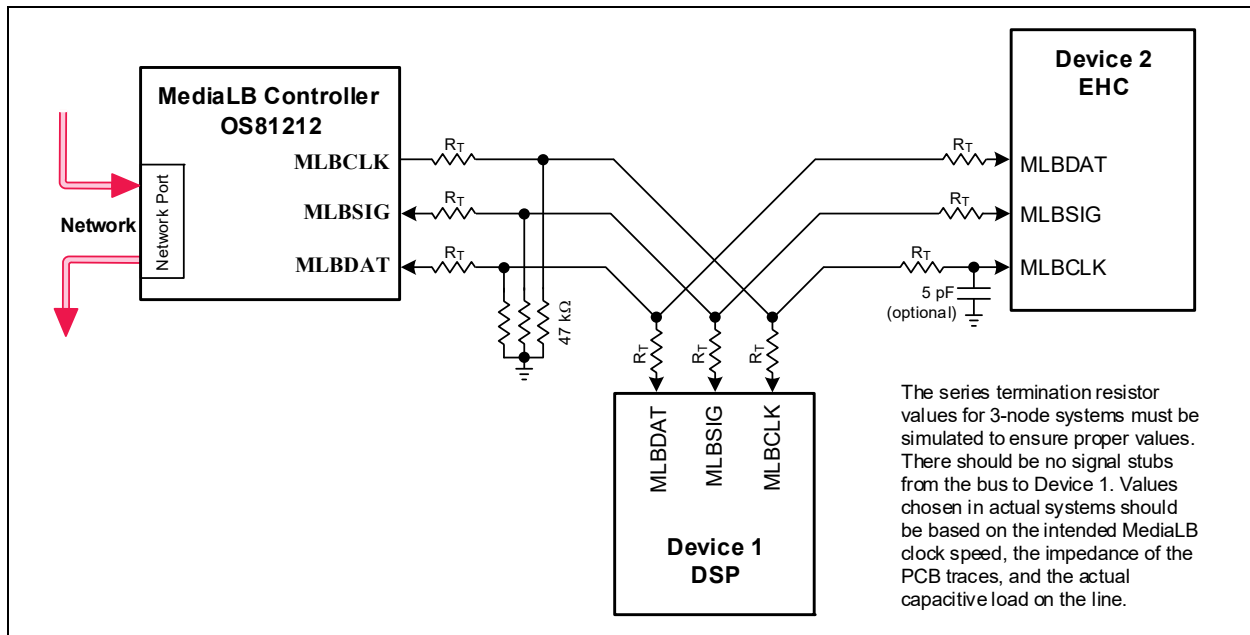
The MediaLB Controller (OS81212) uses **MLBSIG** to grant bus access to a MediaLB Device (e.g. EHC). That Device can then send data to other Devices (including the Controller). The OS81212 **MLBDAT** pin receives data for configuration or transmission onto the network, and transmits data to other Devices from the network receiver. Data and signal information is shifted into Devices by the MLBC bus clock, which is driven by the OS81212 **MLBCLK** pin. A complete description of the Media Local Bus can be found in the [MediaLB Specification \[5\]](#).

The MediaLB 3-pin connection diagrams are shown in [Figure 6-1](#) and [Figure 6-2](#). Each MediaLB line has a weak pull-down resistor to keep the bus in a known state when no device is driving the line. MediaLB devices also have a series resistor near the device for termination and rise/fall time control. The clock line may optionally have AC-parallel termination at the farthest point from the Controller (clock source) to further control rise and fall times. In a three node system, the routing to the device in the middle of the bus should be done in a pass-through manner so that stubs on the bus are eliminated. The MediaLB 3-pin bus should be routed as 50  $\Omega$  single-ended traces on a single layer over a continuous reference plane. As applications will vary, the MediaLB 3-pin bus should be simulated to fine-tune performance, minimize reflections, and optimize the series termination resistor values. Please see [Section 15.9.2 “MediaLB Port”](#) and the [MediaLB Specification \[5\]](#) for layout guidance.

**FIGURE 6-1: MediaLB 3-PIN INTERFACE CONNECTION DIAGRAM 2-NODE SYSTEM**



**FIGURE 6-2: MediaLB 3-PIN INTERFACE CONNECTION DIAGRAM 3-NODE SYSTEM**



## 6.2 MediaLB 3-pin Data Link Layer

The MediaLB protocol supports a set of physical channels for sending data over the Media Local Bus. The physical channels (PCn) are four bytes in length (one quadlet) and can be grouped into logical channels. Each logical channel represents a unidirectional data path (one or more quadlets in length) and is defined by a unique *ChannelAddress*.

On each logical channel, the source (transmitter) is a unique Device, with a destination (receiver) of one or more Devices. In a system, the *ChannelAddresses* are generally static and defined at the development stage; however, the MediaLB Controller can dynamically define additional *ChannelAddresses* during operation to communicate with other Devices.

The MediaLB Controller initiates communication over the Media Local Bus by sending out a *ChannelAddress* on **MLBSIG**. This indicates to the addressed Device that it can begin transmitting on the logical channel associated with that *ChannelAddress*. The structure of the *ChannelAddress* is described in the [MediaLB Specification \[5\], Link Layer Section](#).

# OS81212/4/6

The *ChannelAddress* range for the MediaLB 3-pin interface is 0x0002-0x003E; however, the number of *ChannelAddresses* supported is based on the MediaLB Port speed and the size of the logical channels associated with the *ChannelAddresses*. The total number of physical channels available at a given MediaLB Port 3-pin speed is indicated in [Table 6-1](#). The mapping of specific physical channels into a logical channel is arbitrary and managed by the MediaLB Controller.

**TABLE 6-1: AVAILABLE MediaLB BANDWIDTH**

| MediaLB 3-pin Speed  | Quadlets per MediaLB Frame | Physical Channels | Available Physical Channels <sup>1</sup> | Bytes Available per Frame |
|----------------------|----------------------------|-------------------|--|---------------------------|
| 256×Fs               | 8                          | PC0 - PC7         | 7  | 28                        |
| 512×Fs               | 16                         | PC0 - PC15        | 15                                       | 60                        |
| 1024×Fs <sup>2</sup> | 32                         | PC0 - PC31        | 31                                       | 124                       |

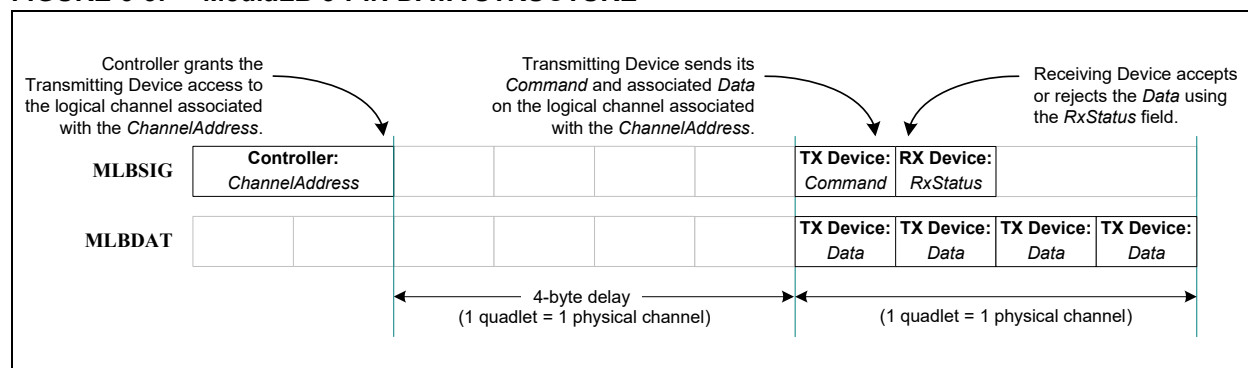
**Note 1:** PC0 (first physical channel of the MediaLB frame) is always used as the *SystemChannel*; therefore, the bytes of this physical channel are not available for application data.

**2:** The MediaLB 3-pin 1024×Fs speed should only be used if the bandwidth is required. Timing constraints at 1024×Fs limit the bus to a Controller and one closely spaced Device. The design should be thoroughly simulated to ensure feasibility and the layout must be optimized to ensure a successful implementation. Please consult the [MediaLB Specification \[5\]](#) and contact Microchip support if 1024×Fs is to be used.

Once per network frame, the MediaLB Controller generates a unique 16-bit *FRAMESYNC* pattern on the signal information line (**MLBSIG**). The end of the *FRAMESYNC* pattern defines byte and physical channel boundaries, and indicates that the MediaLB frame boundary occurs one quadlet later. Physical Channel 0 (PC0), defined as the first physical channel of the MediaLB frame, is always used as the *SystemChannel*. The *FRAMESYNC* pattern represents a special *ChannelAddress*, which grants the MediaLB Controller PC0 for system administration.

As illustrated in [Figure 6-3](#), one quadlet after the *ChannelAddress* is sent by the Controller, the Device associated with that *ChannelAddress* sends out a MediaLB *Command* on **MLBSIG**, while simultaneously sending the corresponding data on **MLBDAT**. All MediaLB Devices, including the OS81212 Controller, compare the *ChannelAddress* to their internal Channel Address table to determine whether they are the intended receiver.

**FIGURE 6-3: MediaLB 3-PIN DATA STRUCTURE**



When the OS81212 Controller is a receiving Device, it places the *RxStatus* response on **MLBSIG** one byte after the MediaLB *Command* is sent by the transmitting Device. To accept the data, an *RxStatus* response of *ReceiverReady* is sent. To reject the data, an *RxStatus* response of *ReceiverBusy* is sent.

**Note:** In accordance with the [MediaLB Specification \[5\]](#), the OS81212 does not acknowledge synchronous data transmissions (*RxStatus* default response of *ReceiverReady*).

When the OS81212 is the receiving Device for a logical channel and recognizes a reception error, it responds with an *RxStatus* of *ReceiverProtocolError* in the next physical channel of that logical channel. The following conditions cause INIC to generate *ReceiverProtocolError*, when receiving control messages and asynchronous packets:

- Inappropriate MediaLB *Command* for a particular channel (e.g. a logical channel is setup for control messages when a synchronous command is received).

- *ControlStart*, *AsyncStart*, *ControlEnd*, or *AsyncEnd* is received while in the middle of a packet reception.
- *ControlEnd* or *AsyncEnd* is not received when the end of the buffer is expected, based on *Port Message Length* (PML).
- The PML indicates a size that is greater than allowed for a control message.

Figure 6-4 illustrates the MediaLB interface, as various Devices drive the bus (**MLBSIG/MLBDAT**) during a given network frame. This figure depicts the 256×Fs MediaLB speed; however, the logic also applies to the 512×Fs and 1024×Fs speeds (with more PCn channels).

The Controller drives the *ChannelAddresses* (*ChannelAddress* B through H in this example), granting the corresponding Devices bus access. The transmitting Device for the *ChannelAddress* drives the *Command* and *Data*, and the receiving Device responds with reception status (*RxStatus*). In the second to last physical channel of the frame (PC6 at 256×Fs), the Controller drives the *FRAMESYNC* pattern onto the signal information line (**MLBSIG**) in place of a *ChannelAddress*, for synchronization to the network frame. The *Command* and *Data* associated with the *FRAMESYNC* (also known as the *SystemChannel*) is sent by the Controller one quadlet following the *FRAMESYNC* pattern, in the first physical channel of the Network frame (PC0).

Depending on the number of physical channels grouped into logical channels, fewer unique *ChannelAddresses* may be seen in the frame. In Figure 6-4, each logical channel is one quadlet (one physical channel), mapping to seven *ChannelAddresses* (B through H). If one of these logical channel consisted of two quadlets and another consisted of three quadlets, then only four unique *ChannelAddresses* would be seen on the bus (B through E).





## 6.2.1 MediaLB COMMANDS AND RESPONSES

Network data can be transported across MediaLB using various MediaLB *Commands* and *RxStatus* responses. The commands/responses used for a logical channel are dependent on the data type transported. Table 6-2 shows the MediaLB *Commands* and *RxStatus* responses associated with each network data type.

**TABLE 6-2: MediaLB COMMANDS & RESPONSES SUPPORTED**

| Network Data                             | Supported Commands   | Supported Responses   |
|--|--|---|
| Synchronous Data                         | <i>SyncData</i><br><i>NoData</i> *   | -   |
| Asynchronous Packets                     | <i>AsyncStart</i><br><i>AsyncContinue</i><br><i>AsyncEnd</i><br><i>AsyncBreak</i><br><i>NoData</i> *   | <i>ReceiverReady</i><br><i>ReceiverBusy</i><br><i>ReceiverBreak</i><br><i>ReceiverProtocolError</i> |
| Control Messages                         | <i>ControlStart</i><br><i>ControlContinue</i><br><i>ControlEnd</i><br><i>ControlBreak</i><br><i>NoData</i> *   | <i>ReceiverReady</i><br><i>ReceiverBusy</i><br><i>ReceiverBreak</i><br><i>ReceiverProtocolError</i> |
| Discrete Frame Isochronous (DFI) Phase   | <i>IsoNoData</i><br><i>Iso1Byte</i><br><i>Iso2Bytes</i><br><i>Iso3Bytes</i><br><i>Iso4Bytes</i><br><i>IsoSync1Byte</i><br><i>IsoSync2Bytes</i><br><i>IsoSync3Bytes</i><br><i>IsoSync4Bytes</i> | -   |
| A/V Packetized Isochronous (AVP) Packets | <i>IsoNoData</i><br><i>Iso1Byte</i><br><i>Iso2Bytes</i><br><i>Iso3Bytes</i><br><i>Iso4Bytes</i><br><i>IsoSync1Byte</i><br><i>IsoSync2Bytes</i><br><i>IsoSync3Bytes</i><br><i>IsoSync4Bytes</i> | -   |
| QoS Packets†                             | <i>AsyncStart</i><br><i>AsyncContinue</i><br><i>AsyncEnd</i>   | <i>ReceiverReady</i><br><i>ReceiverBusy</i>   |

\* For synchronous logical channels, the *NoData* command indicates that the Tx Device assigned to that *ChannelAddress* has not setup the channel yet. For asynchronous and control logical channels, *NoData* is used during packet data transfer where there is no data available to transmit.

† The *AsyncBreak Command* is not supported on logical channels exchanging QoS packets. As a MediaLB receiver, the OS81212 never responds with an *RxStatus* of *ReceiverProtocolError* or *ReceiverBreak*. The only supported responses for QoS data packets are *ReceiverReady* and *ReceiverBusy*.



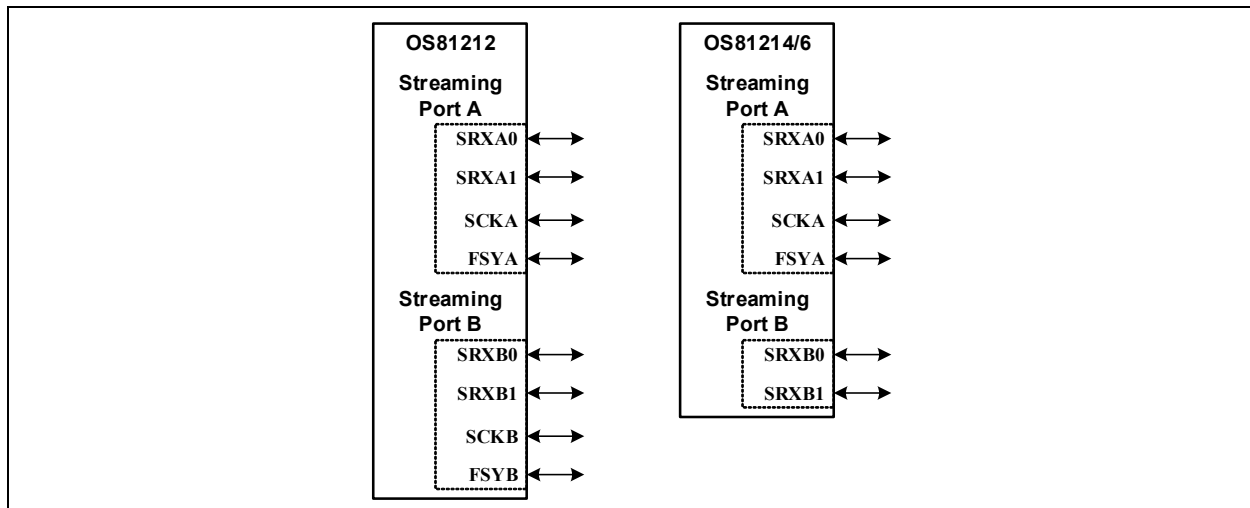
## 7.0 STREAMING PORT

**Note:** The Streaming Port B SCKB and FSYB pins are only available on the OS81212.

The Streaming Port is used by external devices (e.g. ADCs, DACs, microphones, etc.) to source and sink streaming data with low overhead. The Streaming Port can exchange synchronous data in formats which are compatible with industry-standard serial interfaces including left-justified, right-justified, I<sup>2</sup>S-compatible, time-division multiplex (TDM), or sequential formats. Mono microphones with a pulse density modulation (PDM) output are supported using the sequential format.

The OS81212/4/6 supports two Streaming Ports (Streaming Port A and Streaming Port B). Each Streaming Port provides two serial data pins: **SRXA0** and **SRXA1** for Streaming Port A, and **SRXB0** and **SRXB1** for Streaming Port B. In addition to the two serial data pins, Streaming Port A provides a bit clock (**SCKA**) and frame synchronization signal (**FSYA**). Streaming Port B is typically linked to Streaming Port A in which the **SCKA** and **FSYA** clock signals are shared by all the data pins of the two ports. On the OS81212, Streaming Port B additionally provides a bit clock (**SCKB**) and frame synchronization signal (**FSYB**) allowing Streaming Port B data pins to be clocked independently from Streaming Port A. The direction of the Streaming Port A and Streaming Port B data pins can be configured independently, as shown in [Table 7-1](#) and [Table 7-2](#).

**FIGURE 7-1: STREAMING PORT PIN CONFIGURATIONS**



**TABLE 7-1: STREAMING PORT A CONFIGURATION OPTIONS**

| Streaming Port A Mode | I/O Options                      | SRXA0         | SRXA1         | Comments   |
|-----------------------|----------------------------------|---------------|---------------|--|
| Generic Streaming     | Two Input Pins                   | Input (SRA0)  | Input (SRA1)  | All data pins share FSYA/SCKA. Data pins of Streaming Port B may also be available based on configuration. |
|                       | Two Output Pins                  | Output (SXA0) | Output (SXA1) |  |
|                       | One Input Pin/<br>One Output Pin | Input (SRA0)  | Output (SXA1) |  |

**TABLE 7-2: STREAMING PORT B CONFIGURATION OPTIONS**

| Streaming Port B Mode | I/O Options                      | SRXB0         | SRXB1         | Comments  |
|-----------------------|----------------------------------|---------------|---------------|---|
| Generic Streaming     | Two Input Pins                   | Input (SRB0)  | Input (SRB1)  | All data pins share FSYA/SCKA of Streaming Port A, or on the OS81212, the data pin can be clocked independently by FSYB/SCKB. |
|                       | Two Output Pins                  | Output (SXB0) | Output (SXB1) |   |
|                       | One Input Pin/<br>One Output Pin | Input (SRB0)  | Output (SXB1) |   |

**Note:** Streaming Port A must be created before Streaming Port B, as Streaming Port B cannot be the sole Streaming Port. With both Streaming Port A and Streaming Port B in Generic Streaming mode, Streaming Port B can be linked to Streaming Port A in which all the data pins (**SRXA0**, **SRXA1**, **SRXB0**, **SRXB1**) utilize the Streaming Port A clocking signals (**FSYA** and **SCKA**). Additionally on the OS81212, Streaming Port B data pins (**SRXB0** and **SRXB1**) can be configured independently using Streaming Port B clocks (**FSYB** and **SCKB**).

Refer to the [INIC Interface Specification \[4\]](#) for more information on configuring the Streaming Port options and data formats.

## 7.1 Generic Streaming Data Exchange

All data pins on the Streaming Port share a common bit clock **SCKA/SCKB** and a common start-of-frame synchronization signal **FSYA/FSYB** which is often also used to delineate left/right channel boundaries. The bit clock (**SCKA/SCKB**) operates at  $64 \times F_s$ ,  $128 \times F_s$ ,  $256 \times F_s$ , or  $512 \times F_s$  where  $F_s$  is the frame rate of **FSYA/FSYB**.

With both Streaming Port A and Streaming Port B in Generic Streaming mode, Streaming Port B can be linked to Streaming Port A in which all the data pins (**SRXA0**, **SRXA1**, **SRXB0**, **SRXB1**) share the Streaming Port A clocking signals (**FSYA** and **SCKA**). On the OS81212, Streaming Port B clocks (**FSYB** and **SCKB**) can be configured independently from Streaming Port A clocks (**FSYA** and **SCKA**) allowing the Streaming Port B data pins (**SRXB0**, **SRXB1**) to be clocked separately.

When synchronous data is exchanged on the serial data pins, the **FSYA/FSYB** and **SCKA/SCKB** pins can be configured as inputs (Slave Mode) or outputs (Master Mode). As inputs, **FSYA/FSYB** and **SCKA/SCKB** must be synchronous to the network (derived from **RMCK**). In this configuration, the external **FSYA/FSYB** source only needs to meet a minimum high/low time (see [Section 14.7](#)). As outputs, the **FSYA/FSYB** and **SCKA/SCKB** pins are driven based on the network frame rate. When configured as an output, the **FSYA/FSYB** pin is driven by the INIC at a 50% duty cycle, or the INIC can drive **FSYA/FSYB** when using the TDM format. When using Master Mode, there is a clock mode where INIC can drive **FSYA/FSYB** phase locked to the network frame. In this mode, the latency between an audio source to the network or the network to an audio sink is always 2 frames periods ( $41.6 \mu s$  at 48 kHz). When **FSYA/FSYB** is not phase locked (regardless of Slave or Master mode), there is an arbitrary relation between **FSYA/FSYB** and the network frame; therefore, the latency can vary between 1 to 4 frame periods.

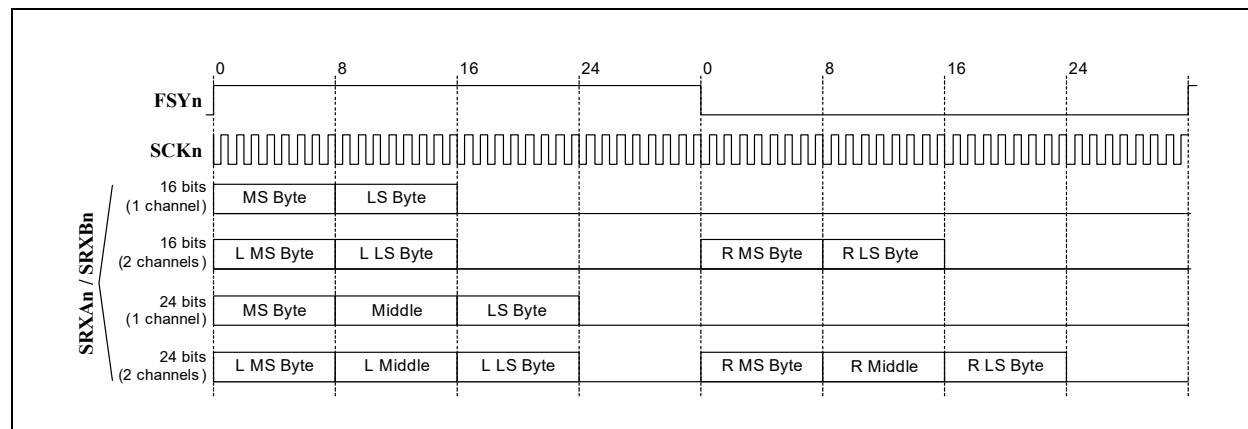
Refer to the [INIC Interface Specification \[4\]](#) for more information on configuring the Streaming Port options and data formats.

### 7.1.1 LEFT-JUSTIFIED ALIGNMENT

When using the left-justified alignment format, the first data bit occurs on the first clock after a transition on **FSYA/FSYB**. The data and **FSYA/FSYB** signals change on the falling edges of **SCKA/SCKB** and are valid on the rising edges. Data is arranged MSB first, high byte first. The start of the frame is indicated by the rising edge of **FSYA/FSYB**.

[Figure 7-2](#) illustrates the data alignment and timing when the left-justified format is selected. For higher data bit rates, the channel data remains aligned with the edges of **FSYA/FSYB** with more unused data bytes between channels.

**FIGURE 7-2: LEFT-JUSTIFIED STREAMING DATA FORMAT**

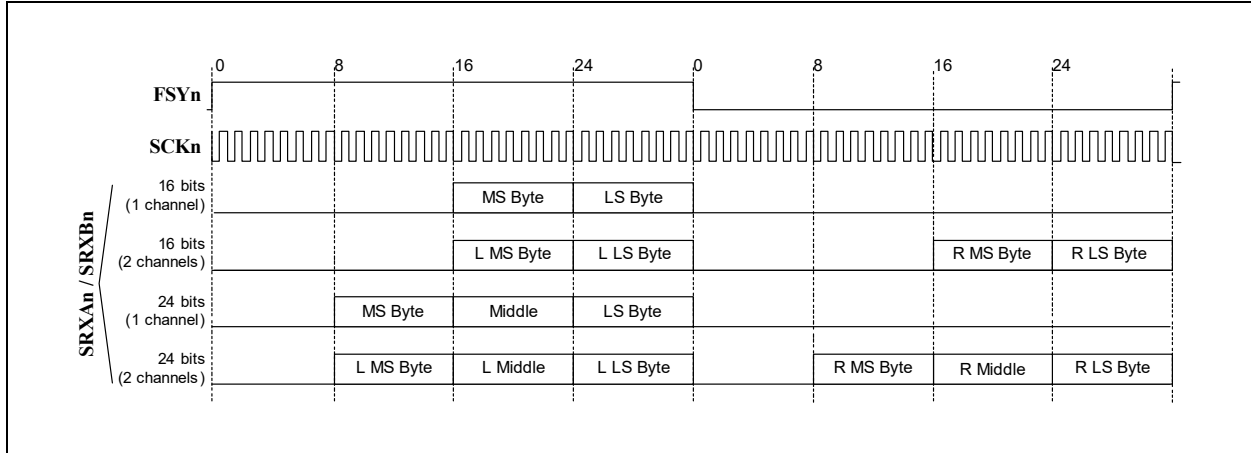


## 7.1.2 RIGHT-JUSTIFIED ALIGNMENT

When using the right-justified alignment format, the last data bit occurs on the last clock before a transition on **FSYA/FSYB**. The data and **FSYA/FSYB** signals change on the falling edges of **SCKA/SCKB** and are valid on the rising edges. Data is arranged MSB first, high byte first. The start of the frame is indicated by the rising edge of **FSYA/FSYB**.

Figure 7-3 illustrates the data alignment and timing when the right-justified format is selected. For higher data bit rates, the channel data remains aligned with the edges of **FSYA/FSYB** with more unused data bytes between channels.

**FIGURE 7-3: RIGHT-JUSTIFIED STREAMING DATA FORMAT**

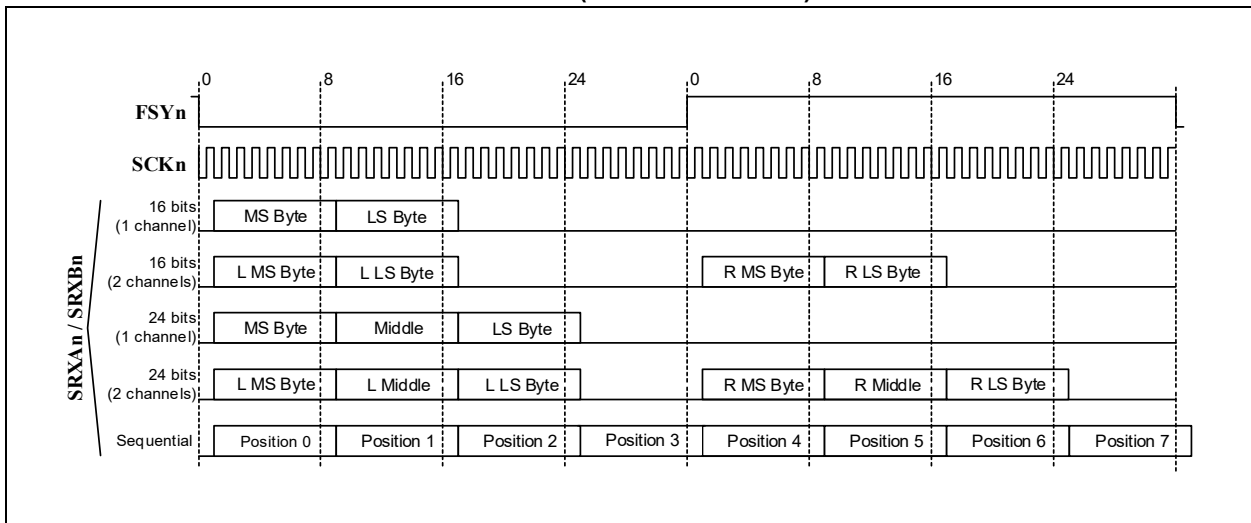


## 7.1.3 DELAYED-BIT ALIGNMENT

When using the delayed-bit alignment format, which is  $I^2S$ -compatible, there is a single **SCKA/SCKB** clock delay between the start of frame (falling edge of **FSYA/FSYB**) and the start of the frame data on **SRXn**. **FSYA/FSYB** and **SRXn** change on the falling edges of **SCKA/SCKB** and are valid on the rising edges. Data is arranged MSB first, high byte first.

Figure 7-4 illustrates the data alignment and timing when the delayed-bit format is selected. For higher data bit rates, the channel data remains bit-delayed from the edges of **FSYA/FSYB** with more unused data bytes between channels.

**FIGURE 7-4: DELAYED-BIT DATA FORMAT ( $I^2S$  COMPATIBLE)**



For a single (mono) channel using delayed-bit alignment format at higher **SCKA/SCKB** rates, a sequential, delayed-bit alignment format is available (as shown in Figure 7-4). When using this format, byte positions in a frame remain bit-delayed and fixed; however, the actual number of byte positions *used* in the frame is flexible. This allows sockets of various sizes to be connected.

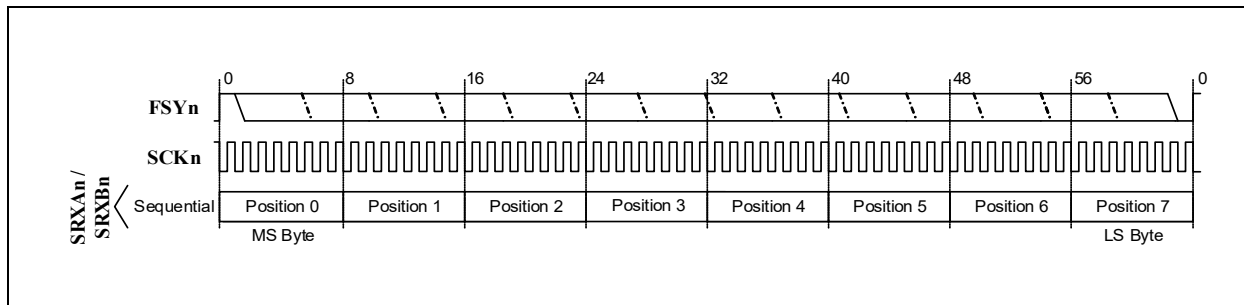
# OS81212/4/6

## 7.1.4 SEQUENTIAL ALIGNMENT

When using the sequential (non-delayed) alignment format, the data and **FSYA/FSYB** signals change on the falling edges of **SCKA/SCKB** and are valid on the rising edges. Data is arranged MSB first, high byte first. This format allows data bytes to exist at every byte position in the frame; however, the actual number of byte positions *used* is flexible. This allows a socket of custom size to be connected.

Figure 7-5 illustrates the byte positions and timing for the sequential (non-delayed) alignment format when **SCKA/SCKB** is  $64 \times F_s$ . For higher data bit rates, more byte positions are available.

**FIGURE 7-5: SEQUENTIAL DATA FORMAT**



Because the sequential alignment format does not place restrictions on socket size, this format can support a single (mono) channel. For example, with **SCKA/SCKB** at  $128 \times F_s$ , a single 16-bit data channel could be sent in the following byte positions:

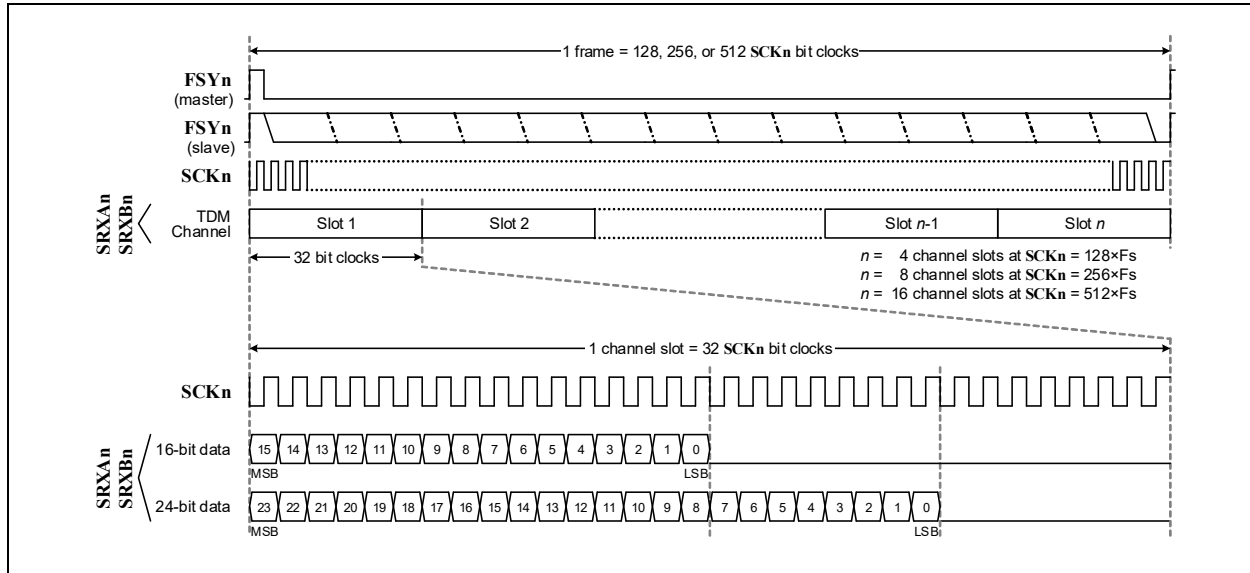
- Positions 0 and 1 (for left-justified alignment)
- Positions 6 and 7 (for right-justified alignment)

The sequential alignment format also supports pulse density modulation (PDM) mono microphone applications. Only **SCKA/SCKB** is required to provide the timebase for PDM microphones. The microphone's continuous bit stream enters a single Streaming Port data pin. When using this feature, **SCKA/SCKB** is typically configured at  $64 \times F_s$  and all data byte positions of the frame will be filled by the microphone. Stereo PDM is not supported.

## 7.1.5 TIME-DIVISION MULTIPLEX

The time-division multiplex (TDM) format enables the routing of multiple data streams on a single data pin. The data and **FSYA/FSYB** signals change on the falling edges **SCKA/SCKB** and are valid on the rising edges. Data is transferred MSB first. Each frame of 128, 256, or 512 bits is composed of multiple 32-bit fixed-size *channel slots*. Figure 7-6 illustrates the TDM (non-delayed) alignment format. Identified in Figure 7-6 by the variable  $n$ , the number of channel slots per frame is dependent upon the **SCKA/SCKB** bit clock rate. Four 32-bit channel slots are available per data pin at a clock rate of  $128 \times F_s$ , eight channel slots are available at  $256 \times F_s$ , and 16 channel slots are available at  $512 \times F_s$ . All data is routed within each channel slot most-significant bit first and left-justified. Either 16-bit or 24-bit data may be transferred within the TDM channel slots and all channel slots for a pin are configured to transfer the same number of data bits per channel slot; i.e., it is not possible to mix routing of 16-bit and 24-bit data within channel slots of a single pin. However, not all channel slots within the frame must be used to route data. All unrouted transmit channel slots and channel slot bytes are transmitted as zeros.

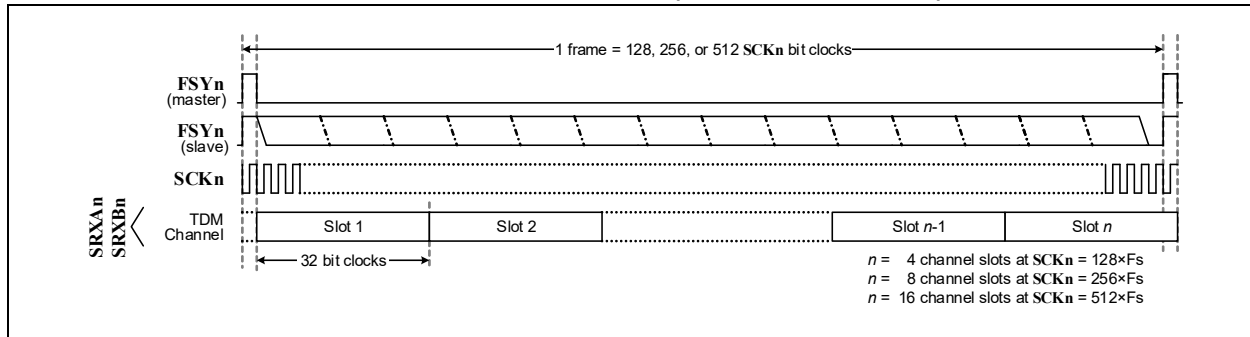
**FIGURE 7-6: TIME-DIVISION MULTIPLEX FORMAT**



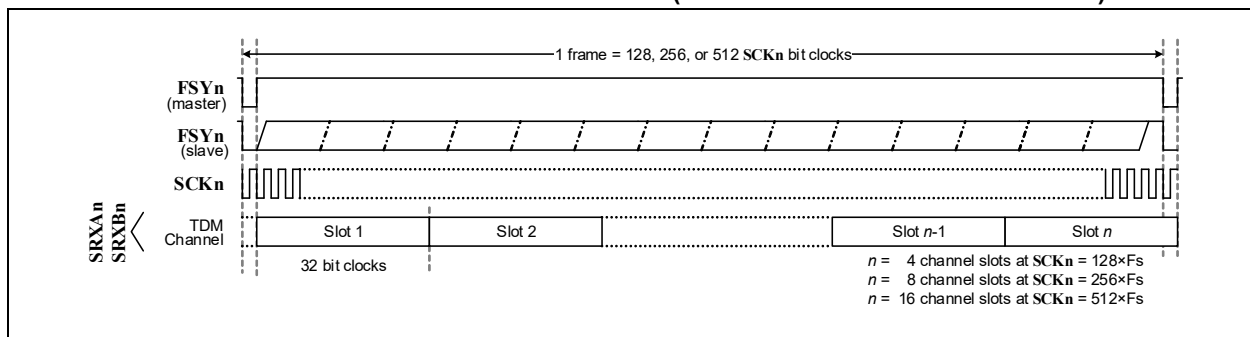
The TDM alignment format may be configured for operation in one of three clock formats:

- TDM Non-Delayed - The first channel slot is aligned with the start of the frame identified by the rising edge of FSYA/FSYB (Figure 7-6).
- TDM Delayed-Bit (bit-delayed only) - The first channel slot is delayed from the start of frame by one bit clock and a rising edge of FSYA/FSYB signifies the start of the frame (Figure 7-7).
- TDM Delayed-Bit (I<sup>2</sup>S compatible) - The first channel slot is delayed by one bit clock from the start of the frame coincident with a falling edge on FSYA/FSYB (Figure 7-8).

**FIGURE 7-7: TIME-DIVISION MULTIPLEX FORMAT (DELAYED-BIT ONLY)**



**FIGURE 7-8: TIME-DIVISION MULTIPLEX FORMAT (DELAYED-BIT - I<sup>2</sup>S COMPATIBLE)**



# OS81212/4/6

## 8.0 SPI PORT

**Note:** The SPI Port is only available on the OS81212/4.

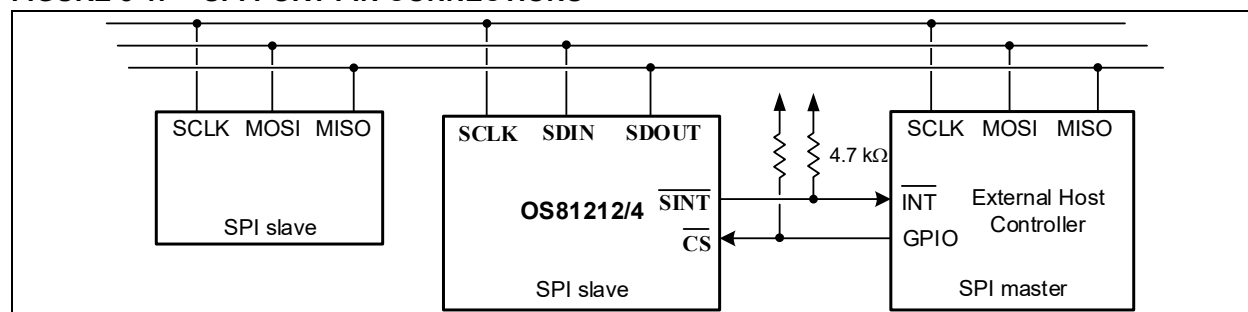
The *Serial Peripheral Interface (SPI)* Port on the OS81212/4 is designed to interface with SPI Port available on many industry devices. In addition to the four standard SPI signals ( $\overline{\text{CS}}$ , SCLK, SDIN, and SDOUT), the SPI Port uses an additional interrupt line (SINT) which is used for signaling to an External Host Controller (EHC). When the SPI Port is enabled, it operates as an SPI bus slave and can exchange data packets at a rate up to 25 Mbit/s (e.g. SCLK=512×Fs at Fs=48 kHz). The SPI Port supports the exchange of asynchronous and control packets (*MHP Frames/ Ethernet Frames and Application Control Frames*) at data rates higher than possible with I<sup>2</sup>C.

Table 8-1 shows the 5-pin SPI Port, including the four standard signals and the additional interrupt. Typical connections for the SPI Port are shown in Figure 8-1. External pull-up resistors are recommended on SINT and  $\overline{\text{CS}}$  to maintain these signals in a known state while the interface is inactive (e.g. power-up/reset).

**TABLE 8-1: SERIAL PERIPHERAL INTERFACE**

| Pin Name               | Description              | Direction (Slave Mode) |
|------------------------|--------------------------|------------------------|
| $\overline{\text{CS}}$ | Chip Select (active low) | input                  |
| SDIN                   | Serial Data In (MOSI)    | input                  |
| SDOUT                  | Serial Data Out (MISO)   | output                 |
| SCLK                   | Clock                    | input                  |
| SINT                   | Interrupt (active low)   | output                 |

**FIGURE 8-1: SPI PORT PIN CONNECTIONS**



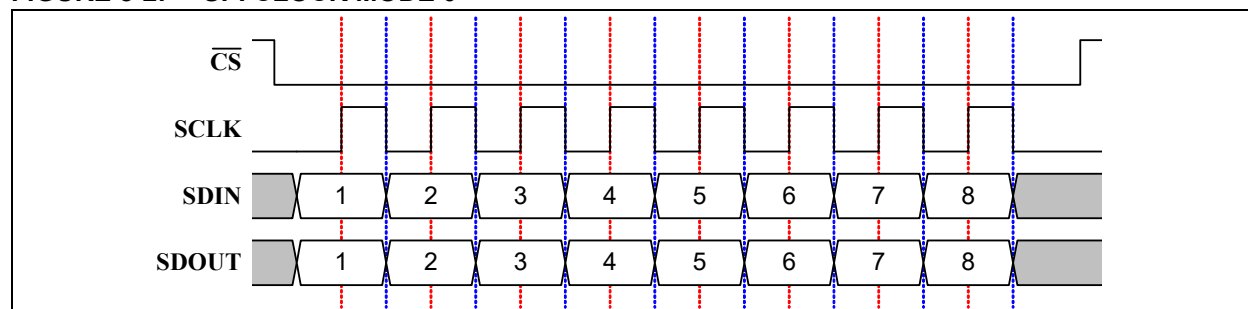
### 8.1 Clock Modes

Four different clock modes are available on the SPI Port. Refer to the [INIC Interface Specification \[4\]](#) for more information on configuring the SPI Port.

#### 8.1.1 MODE 0

In Clock Mode 0, SCLK is *low* when the bus is idle. Data transitions on the *falling* clock edge and is latched on the *rising* clock edge. This mode is defined as (CPOL=0, CPHA=0) by the [SPI Block Guide Version 4.01 \[10\]](#).

**FIGURE 8-2: SPI CLOCK MODE 0**

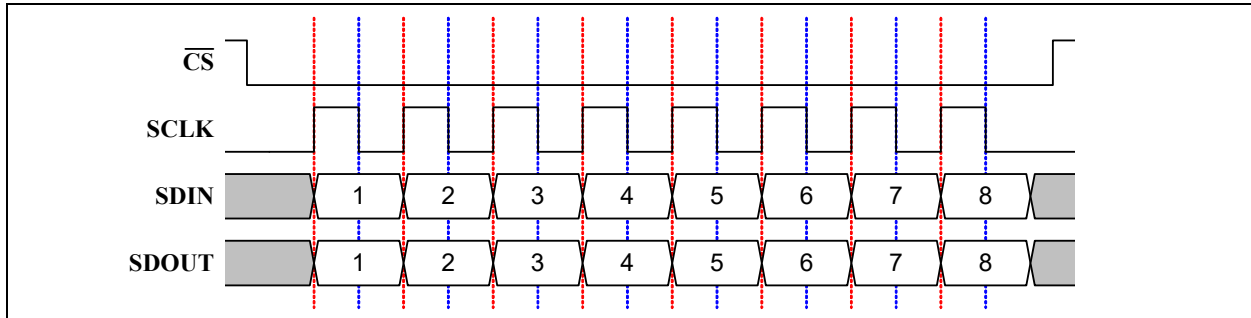




## 8.1.2 MODE 1

In Clock Mode 1, SCLK is *low* when the bus is idle. Data transitions on the *rising* clock edge and is latched on the *falling* clock edge. This mode is defined as (CPOL=0, CPHA=1) by the [SPI Block Guide Version 4.01 \[10\]](#).

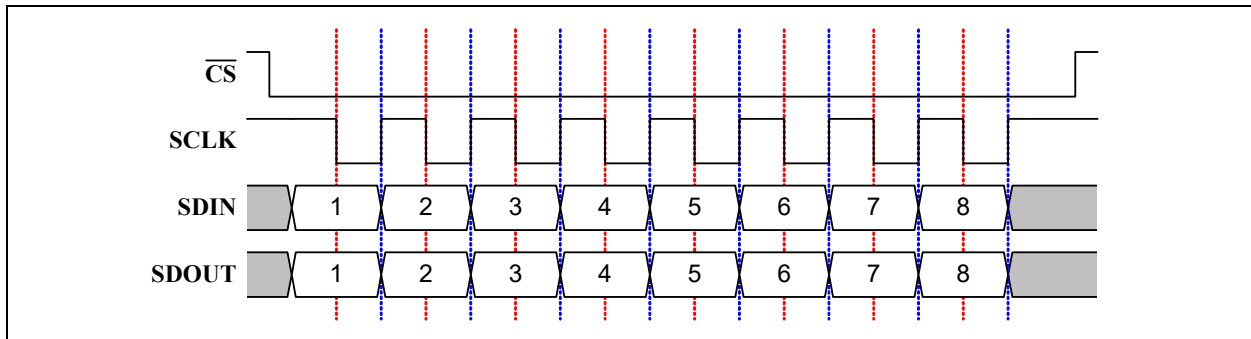
**FIGURE 8-3: SPI CLOCK MODE 1**



## 8.1.3 MODE 2

In Clock Mode 2, SCLK is *high* when the bus is idle. Data transitions on the *rising* clock edge and is latched on the *falling* clock edge. This mode is defined as (CPOL=1, CPHA=0) by the [SPI Block Guide Version 4.01 \[10\]](#).

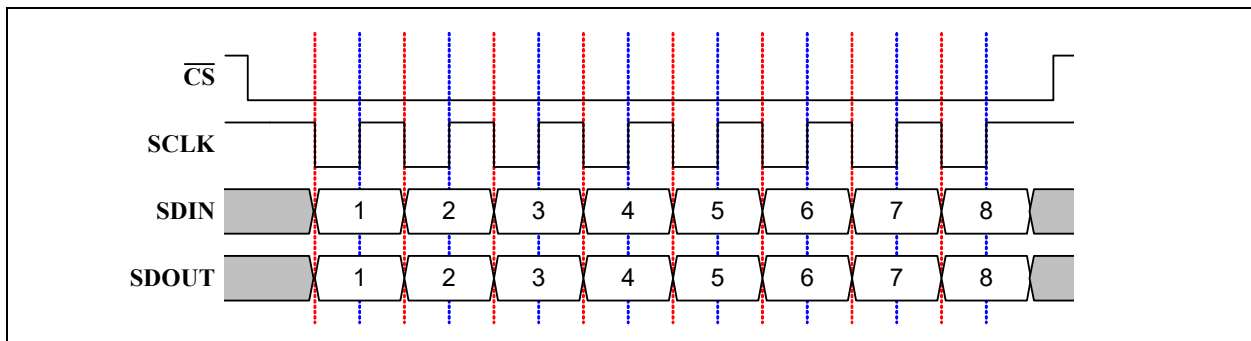
**FIGURE 8-4: SPI CLOCK MODE 2**



## 8.1.4 MODE 3

In Clock Mode 3, SCLK is *high* when the bus is idle. Data transitions on the *falling* clock edge and is latched on the *rising* clock edge. This mode is defined as (CPOL=1, CPHA=1) by the [SPI Block Guide Version 4.01 \[10\]](#).

**FIGURE 8-5: SPI CLOCK MODE 3**



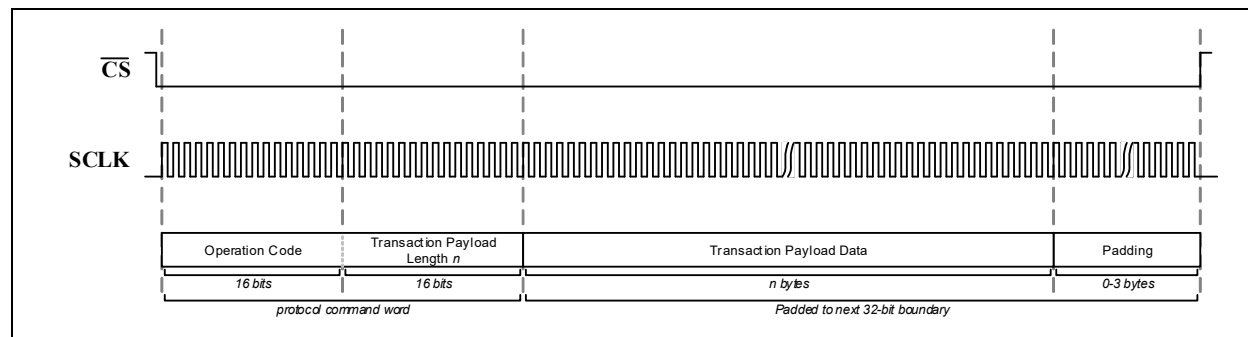
## 8.2 SPI Protocol

As illustrated in Figure 8-6, the SPI bus master (also referred to as the EHC) initiates communication with the SPI Port by driving  $\overline{CS}$  low. A *protocol command word* is the first four bytes sent by the EHC on **SDIN** after the falling edge of  $\overline{CS}$ . This 32-bit command word defines the direction of the data transfer (e.g., read or write operation), the resource being accessed (registers or packet data channels), and the length of the *transaction payload data* that follows. The remaining bytes of the transaction contain the *transaction payload data* being read from or written to the SPI Port. Multiple transactions are not allowed while  $\overline{CS}$  is asserted; i.e., only one transaction is allowed for each assertion of  $\overline{CS}$ . The  $\overline{CS}$  pin must remain asserted through the entire transaction, only going high following the last bit clock of the transaction.

The SPI Port supports 32-bit word aligned communication. All SPI Port transactions must therefore be a multiple of 4 bytes in length. This is accomplished by adding one to three padding bytes of data at the end of the transaction, if necessary.

Data is always shifted most significant bit first into **SDIN** and out of **SDOUT**.

**FIGURE 8-6: SPI TRANSACTION**



**Note:** During the time that the EHC is writing the 32-bit command word to the SPI Port on **SDIN**, there is no data for the INIC to output on **SDOUT**. The **SDOUT** pin is therefore driven low for the first 32-bits of every transaction. Similarly, when the EHC is performing a write transaction to the SPI Port, **SDOUT** remains driven low during the entire write transaction. The INIC only drives data onto the **SDOUT** pin following the *protocol command word* during read transactions.

### 8.2.1 PROTOCOL COMMAND WORD

A *protocol command word* is the first four bytes sent by the EHC on the **SDIN** data line after the falling edge of  $\overline{CS}$ . The 32-bit *protocol command word* consists of a 16-bit *operation code* (opcode) followed by a 16-bit *transaction payload length*. As shown in Table 8-2, the opcode defines the direction of the transaction data transfer, the SPI Port resource being accessed, and the resource address. The opcode read-not-write (**RNW**) bit indicates if the transaction is a read from or a write to the SPI port. The channel-not-register resource selector bit (**CNR**) selects the type of resource (control/status registers or logical packet channels) within the SPI to be accessed. When the **CNR** bit is clear, the transaction will access a set of SPI configuration and status registers. These registers are described in detail in Section 8.7. With **CNR** set, the transaction will access one of the packet communication channels. The opcode address field (**ADDR**) specifies the address of the SPI Port resource being accessed (register or packet communication channel address). The 16-bit *transaction payload length* field specifies the number of data bytes in the *transaction payload data* that follows, excluding any necessary padding.

**Note:** The *transaction payload length* field must never be zero, otherwise an SPI protocol error will result.

Table 8-2 shows the structure of the 16-bit opcode portion of the *protocol command word*.

**TABLE 8-2: SPI TRANSACTION OPCODE**

|     |    |    |     |           |    |   |   |
|-----|----|----|-----|-----------|----|---|---|
| RNW | —  | —  | CNR | ADDR<3:0> |    |   |   |
| 15  | 14 | 13 | 12  | 11        | 10 | 9 | 8 |
| —   | —  | —  | —   | —         | —  | — | — |
| 7   | 6  | 5  | 4   | 3         | 2  | 1 | 0 |

bit 15      **RNW:** SPI transaction direction (Read-Not-Write)

0 = Write Transaction

1 = Read Transaction

bit 14-13    **Unimplemented:** Write as '0'

bit 12      **CNR:** Channel-Not-Register resource selection bit.

0 = Address SPI control and status registers

1 = Address packet communication channel resources

bit 11-8    **ADDR<3:0>:** SPI resource address

When **CNR** is '0', the **ADDR** field addresses the control and status registers. See [Section 8.7](#).

When **CNR** is '1', the **ADDR** field selects the packet communication channels:

0000 = Driver Control Interface (DCI) Channel access

0010 = Asynchronous Channel access (*MHP Frames/Ethernet Frames*)

0100 = Control Channel access (*Application Control Frames*)

All other encoding values are reserved and should not be used.

bit 7-0      **Unimplemented:** Write as '0'

## 8.3 Control / Status Register Transactions

The SPI Port implements a register interface which the EHC uses to control, configure, and obtain status of the SPI hardware. These control and status registers are described in detail in [Section 8.7](#).

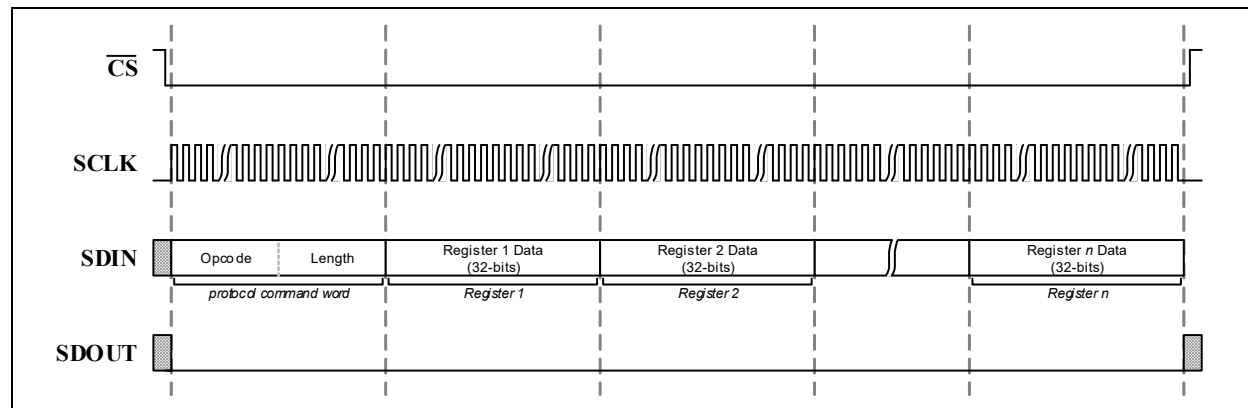
### 8.3.1 CONTROL / STATUS REGISTER WRITE TRANSACTION

When writing to SPI registers, the EHC constructs the *protocol command word* with the opcode **RNW** bit clear to indicate a write transaction, and the **CNR** bit clear to select addressing registers. The opcode address field, **ADDR**, selects the first register address to be written. For example, writing to register address 0x1 would result in a *protocol command word* opcode of 0x0100.

The transaction payload contains the data to be written to the registers. Each register is 32-bits in size, so four bytes of data are required in the transaction payload for each register written. Multiple registers with consecutive addresses may be written using a single write transaction as shown in [Figure 8-7](#). A maximum of 8 consecutive registers may be written in a single transaction. Attempting to write more than 8 registers will result in the additional data being discarded.

The transaction payload must be a multiple of 4 bytes in length. The number of registers that will be written is equal to the size of the transaction payload (in bytes) divided by 4. As such, the *transaction payload length* field within the *protocol command word* must always be an integer multiple of four, therefore padding bytes are not necessary.

**FIGURE 8-7: SPI REGISTER WRITE TRANSACTION**



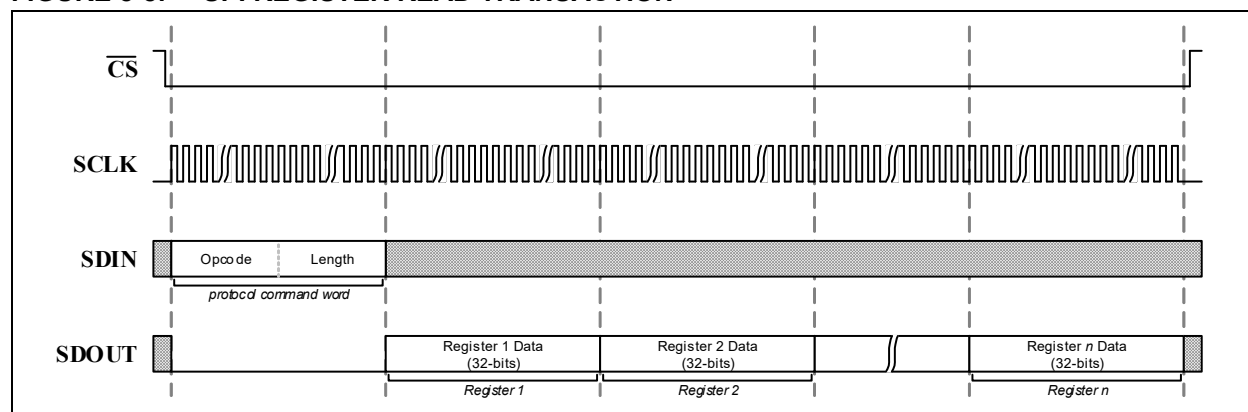
### 8.3.2 CONTROL / STATUS REGISTER READ TRANSACTION

Reading from SPI registers requires the EHC to construct the *protocol command word* with the opcode **RNW** bit set to indicate a read transaction, and the **CNR** bit clear to select addressing registers. The opcode address field, **ADDR**, selects the first register address to be read. For example, reading from register address 0x1 would result in a *protocol command word* opcode of 0x8100.

Following the *protocol command word* sent by the EHC on **SDIN**, the INIC will transmit the transaction payload onto **SDOUT**. The transaction payload contains the data read from the registers. Each register is 32-bits in size, so four bytes of data are transmitted in the transaction payload for each register read. Multiple registers with consecutive addresses may be read using a single read transaction as shown in [Figure 8-8](#). A maximum of 8 registers may be read in a single transaction. Attempting to read more than 8 registers will result in zeros being read.

The transaction payload must be a multiple of 4 bytes in length. The number of registers that will be read is equal to the size of the transaction payload (in bytes) divided by 4. As such, the *transaction payload length* field within the *protocol command word* must always be an integer multiple of four.

**FIGURE 8-8: SPI REGISTER READ TRANSACTION**



## 8.4 Packet Transactions

The SPI Port supports the transfer of *MHP Frames/Ethernet Frames* and *Application Control Frames*. The transfer of packet messages through the SPI Port is accomplished by reading and writing one of the packet channel addresses. The EHC writes the *protocol command word* opcode **CNR** bit set as '1' to select addressing the packet channels. The address **ADDR** field is used to select the specific packet channel which will be read from or written to.

*MHP Frames/Ethernet Frames* and *Application Control Frames* are encapsulated within *Port Messages* which are transferred within the SPI transaction payload. The first two bytes of the *Port Message* contain the *Port Message Length* (PML). The PML contains the number of bytes remaining in the *Port Message*. Refer to the [INIC Interface Specification \[4\]](#) for more detailed information. When multiple packets are transferred in a single SPI transaction, the PML is used to locate the end of the current message and the beginning of the next message (and next PML).

## 8.4.1 PACKET WRITE TRANSACTION

The SPI Port is able to receive packets from the EHC when the packet receive status bit (**ARIS**, **CRIS**) is set within the CHSTS register. The receive status bit is set to '1' when the SPI is able to receive at least one packet, and the total amount of buffer space available for receiving a packet is greater than the receive threshold (**ARTHR**, **CRTHR**). The receive threshold is typically configured to be equal to, or greater than the maximum size of packet the EHC may write to the SPI Port. The maximum size of a *MHP Frame* or *Ethernet Frame* is 1510 bytes, while an *Application Control Frame* is a maximum of 64 bytes in size. By configuring the receive threshold to the maximum packet size, the EHC is guaranteed to be able to write one maximal length packet each time the receive interrupt status bit is set.

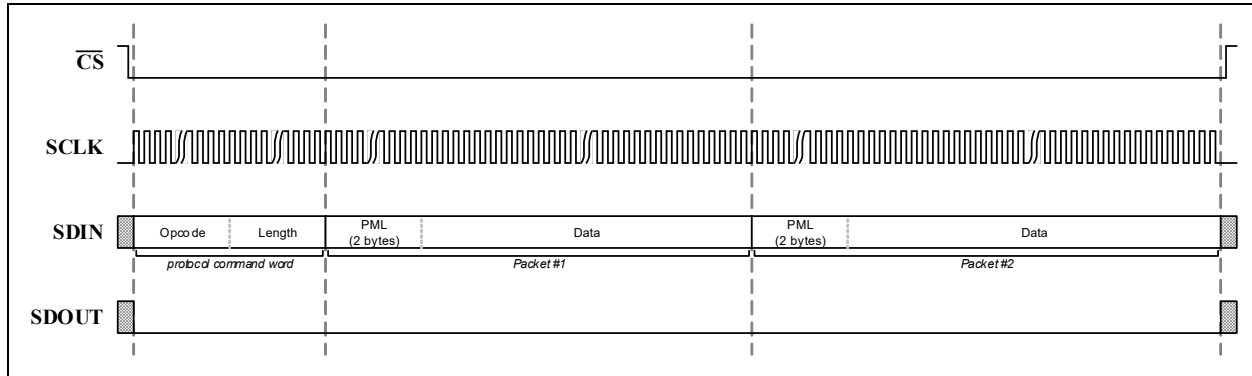
Prior to writing packets to the SPI Port, the EHC must determine how many packets and the total number of bytes that may be received by the SPI Port. This information is available by reading the receive buffer status register (ARXB, CRXB). The **RXPCNT** field specifies the maximum number of packets that the SPI Port is able to receive and the **RXFREE** field contains the maximum number of bytes that may be written by the EHC.

**Note:** The total number of bytes written, *plus one extra byte per packet*, must not exceed the number of bytes specified by **RXFREE** field.

The EHC writes packets to the SPI Port by constructing a *protocol command word* with the opcode **RNW** bit clear to indicate a write transaction and the **CNR** bit set to '1' to select writing to the packet channels. The opcode address field, **ADDR**, selects the packet data channel (asynchronous or control) to which the packets will be written. For example, writing a *MHP Frame* or *Ethernet Frame* to the asynchronous channel requires a *protocol command word* opcode of 0x1200, whereas an opcode of 0x1400 results in writing a *Application Control Frame* to the control channel.

The transaction payload contains the packet data to be written to the packet channel. Multiple packets may be written using a single write transaction as shown in Figure 8-9. Only whole packets may be written; writing of partial (i.e., incomplete) packets is not permitted. When multiple packets are being written in a single transaction, padding is only applied at the end of the *transaction payload data*, not between packets.

**FIGURE 8-9: SPI PACKET WRITE TRANSACTION**



## 8.4.2 PACKET READ TRANSACTION

The SPI Port has data packets available for the EHC to read when the packet transmit status (**ATIS**, **CTIS**) bit is set within the CHSTS register. The transmit status bit is set automatically when the INIC has at least one packet available for reading. Once the EHC has determined that data packets are available for reading from the SPI Port, the transmit buffer status register (ATXB, CTXB) status register is read to determine the number of packets and total number of bytes the SPI Port has ready for the EHC to read. The **TXPCNT** field contains the number of packets available for the EHC to read. The **TXAVAIL** field reflects the total number of packet bytes required to read the packets.

Reading packets from the SPI Port requires the EHC to construct a *protocol command word* with the opcode **RNW** bit set to indicate a read transaction and the **CNR** bit set to '1' to select reading from the packet channels. The opcode address field, **ADDR**, selects the packet data channel (asynchronous or control) from which the packets will be read. For example, a *protocol command word* opcode of 0x9200 results in the reading of *MHP Frame* or *Ethernet Frame* from the asynchronous channel and an opcode of 0x9400 results in reading a *Application Control Frame* from the control channel.

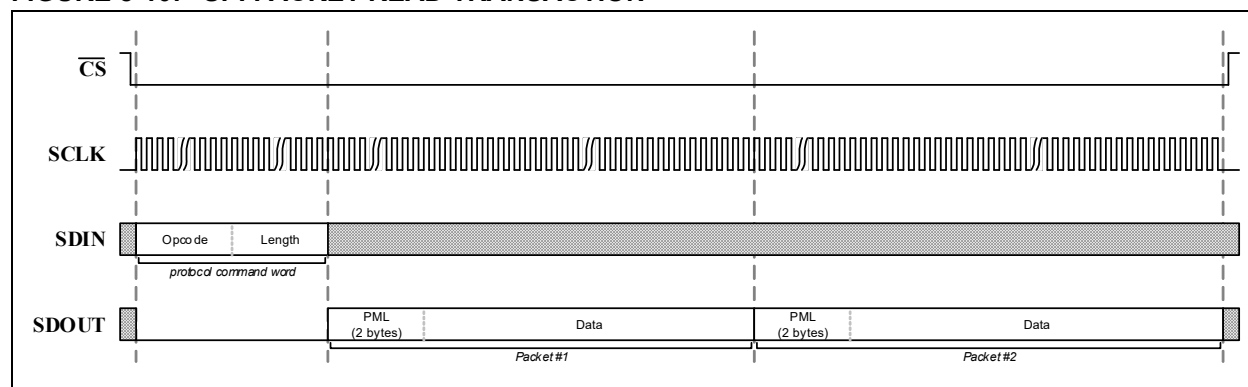
# OS81212/4/6

Following the write of the *protocol command word* by the EHC on **SDIN**, INIC transmits the transaction payload onto **SDOUT**. The transaction payload contains the packet data read from the logical packet data channel. Multiple packets may be read using a single read transaction as shown in [Figure 8-10](#). Reading of partial (incomplete) packets is not permitted.

**Note:** The *transaction payload length* field must never be zero, otherwise an SPI protocol error will result, possibly resulting in the loss of data packets.

**Note:** As padding is only applied at the end of the *transaction payload data*. When multiple packets are being read in a single transaction, padding will not be inserted between packets. Therefore, successive packets in the transaction may not begin on a 32-bit aligned boundary.

**FIGURE 8-10: SPI PACKET READ TRANSACTION**



## 8.5 Driver Control Interface (DCI)

The EHC normally accesses all INIC configuration and status through the normal Control Message interface (e.g., MediaLB, I<sup>2</sup>C Control Port). However, the Control Message interface may not always be available due to an interface error or device reset. In these conditions, the Driver Control Interface (DCI) enables the EHC driver to directly access low-level driver related configuration and status of the INIC when no other communication interface is available. The low-level configuration and status are implemented as a virtual register which the EHC accesses by writing *DCI Command* packets to the DCI channel. Each *DCI Command* either writes data to the DCI virtual registers, or requests data to be read from the registers and returned in a *DCI Response* packet.

Low-level INIC configuration and status accessible to the EHC via the DCI include:

- Retrieval and configuration of INIC status such as Node Address, Packet EUI-48 Address, Network Availability, etc.
- Packet channel synchronization in the event of a fatal channel error
- Device synchronization in the event the EHC or INIC resets or encounters a fatal error
- Bidirectional notification between the EHC and INIC on status changes or events including resource availability, channel and device synchronization, fatal errors, etc.

The DCI also implements an interface to communicate events between the INIC and EHC. The EHC may signal to the INIC that a *DCI Command* has been written is ready for processing by the INIC (**CMDREQ**), or that the EHC driver requires synchronization with the INIC (**SYNCREQ**). Three status flags are used by the INIC to signal events to the EHC. The **CMDDN** status indicates that the INIC has completed processing of a *DCI Command* and a *DCI Response* is ready for the EHC to read. The INIC uses the **NTF** status bit to notify the EHC that a DCI virtual register status has changed. Should the INIC detect an error condition, it will be signaled with the **ERR** status bit. The DCI may be configured to assert the **DEIS** bit and the **SINT** pin on the occurrence of any or all of the three INIC generated event status flags being asserted (see [Figure 8-13](#)).

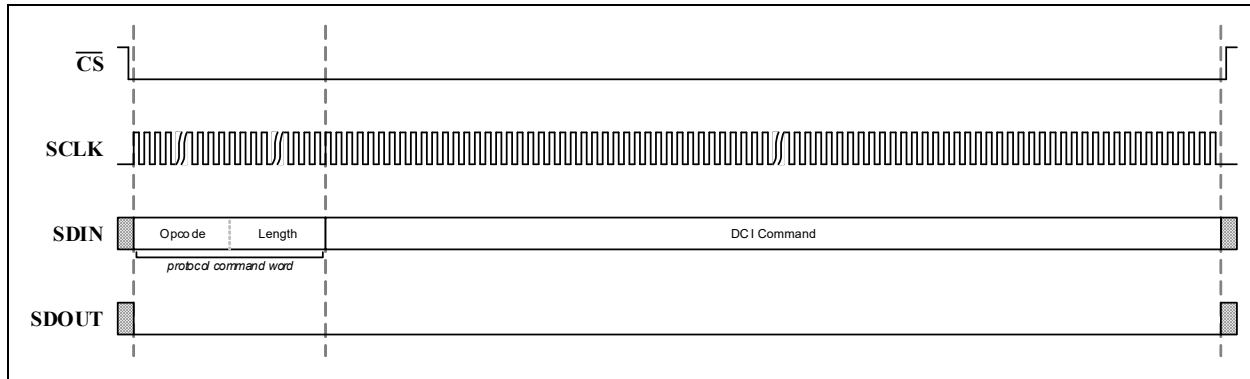
Refer to the [INIC Interface Specification \[4\]](#) for detailed information regarding the *DCI Command* and *DCI Response* packet formats as well as the virtual register mapping.

## 8.5.1 DCI COMMAND WRITE

The writing of *DCI Command* packets is performed in a manner similar to the writing of packet messages to the control or asynchronous channels. The INIC is ready to accept a *DCI Command* packet when the **CMDREQ**, **CMDDN**, and **SYNCREQ** bits in the DCCTL register are all '0'. The EHC writes a *DCI Command* packet to the SPI Port by constructing a *protocol command word* with the opcode **RNW** bit clear to indicate a write transaction and the **CNR** bit set to '1' to select writing to the packet channels. The opcode address field, **ADDR**, is set accordingly to select the DCI data channel for writing. For example, writing a *DCI Command* packet requires a *protocol command word* opcode of 0x1000.

The transaction payload follows the *protocol command word* and contains the *DCI Command* packet to be written to the DCI channel as shown in Figure 8-11. The *DCI Command* is always an integer multiple of 4 bytes in length, therefore no padding bytes are necessary. Once the EHC has written the *DCI Command*, the INIC is notified of the pending command by setting **CMDREQ** to '1' at which time the INIC will execute the posted *DCI Command* and clear **CMDREQ**.

**FIGURE 8-11: DCI COMMAND PACKET WRITE TRANSACTION**



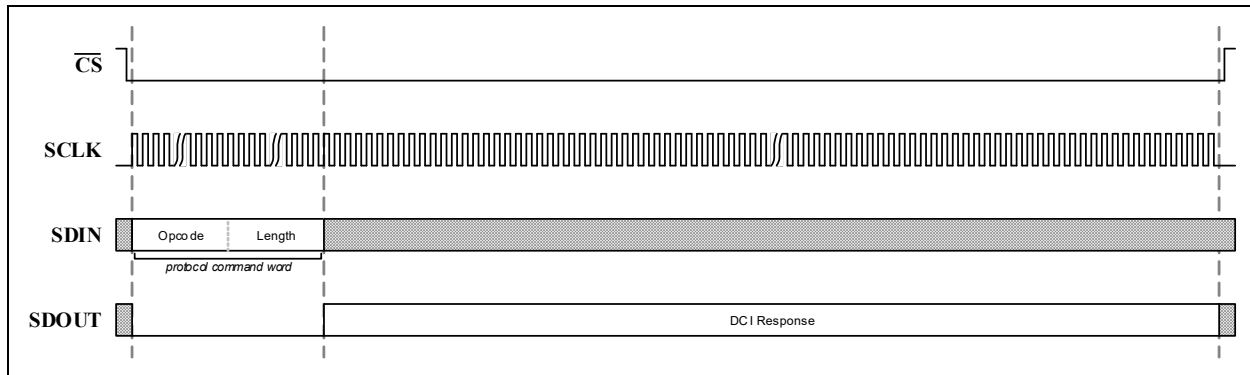
## 8.5.2 DCI RESPONSE READ

Once the INIC has executed a *DCI Command*, it will clear the **CMDREQ** bit. When INIC has responded to the command with a *DCI Response*, it will set the **CMDDN** bit to '1' to alert the EHC that a response packet is to be read. If unmasked, the **DEIS** bit in the CHSTS register will also be set when a *DCI Response* packet is available for reading.

Reading a *DCI Response* packet from the SPI Port requires the EHC to construct a *protocol command word* with the opcode **RNW** bit set to indicate a read transaction and the **CNR** bit set to '1' to select reading from the packet channels. The opcode address field, **ADDR**, is set to select reading from the DCI data channel. For example, a *protocol command word* opcode of 0x9000 results in the reading of *DCI Response* packets from the DCI data channel.

Following the write of the *protocol command word* by the EHC on **SDIN**, INIC transmits the transaction payload containing the *DCI Response* onto **SDOUT** (see Figure 8-12). The *DCI Response* is always an integer multiple of 4 bytes in length, therefore no padding bytes are transmitted. Once the EHC has read the *DCI Response*, the **CMDDN** bit is cleared by writing a '1'.

**FIGURE 8-12: DCI RESPONSE PACKET READ TRANSACTION**



## 8.6 Interrupt Functionality

As an alternative to polling the **CHSTS** register to determine a change if the SPI Port state, the EHC can instead configure and monitor **SINT**. The **SINT** pin is a shared signal to indicate availability for packet read and write transactions on the asynchronous and control packet channels, and pending DCI events the EHC should react to.

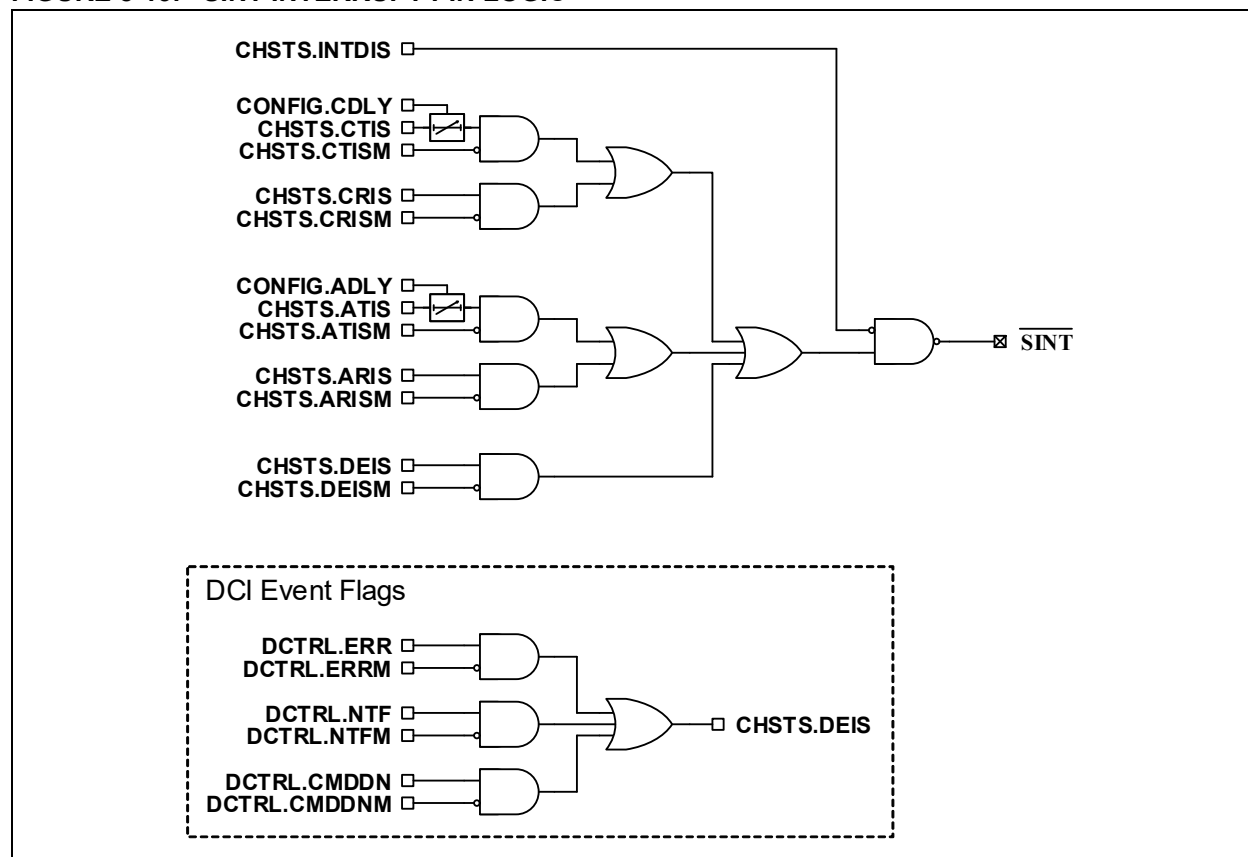
When an *Application Control Frame* or *MHP Frame/Ethernet Frame* becomes ready for reading or writing, the corresponding interrupt channel status bit within the **CHSTS** register will be set causing the **SINT** pin to be asserted. The interrupt channel status bit will remain pending until the event that caused the status to be set is resolved. When any of the interrupt channel status bits within the **CHSTS** register are set, the **SINT** pin is asserted to notify the EHC that the SPI Port is available for packet communication. In addition, the **SINT** pin may be used to indicate the assertion of the DCI event status, **DEIS**, indicating a need for the EHC to read the DCI virtual registers to determine the source and nature of the event.

Each interrupt source status bit that can generate an assertion of the **SINT** pin has a corresponding interrupt mask bit. When the interrupt mask bit is set, the associated interrupt channel status bit will not cause an assertion of the **SINT** pin. Additionally, assertion of the **SINT** pin may be disabled globally by setting the **CHSTS.INTDIS** bit. When disabled, **SINT** will remain driven high.

The logic for asserting the **SINT** pin is illustrated in [Figure 8-13](#).

After it is asserted, **SINT** remains low until the SPI transaction starts. After the transaction completes and **CS** is de-asserted, the INIC will then check if there are new interrupts pending (from channel sources or DCI).

**FIGURE 8-13: SINT INTERRUPT PIN LOGIC**





## 8.6.1 INTERRUPT DELAY

When a burst of small *Packet Frames/Ethernet Data Frames* or *Control Frames* (referred to as *MHP Frames/Ethernet Frames* or *Application Control Frames* at the EHC driver level) is received from the network for transmission to the EHC, the SINT pin may repeatedly become asserted in quick succession, possibly adding an increased load on the EHC to process the interrupts by reading each packet individually. An alternative is to limit the assertion rate of the SINT pin when asynchronous packets are ready for transmit by the INIC.

When the asynchronous or control packet transmit delay mode is enabled by setting **ADLY**, or **CDLY**, respectively, an internal timer is started when the first packet is ready for transmit to the EHC and the packet transmit interrupt status bit (**ATIS**, or **CTIS**) is set. Only after the timer has expired will the SINT pin assert. This delay allows the EHC to read and process multiple packets at once. The amount of delay is configured in **INTDLY**.

While the ability to delay assertion of SINT until multiple, small packets are ready for reading is useful, it is also desirable for the EHC to immediately read and process a large *MHP Frame* or *Ethernet Frame* from the asynchronous channel. Therefore, when the asynchronous transmit interrupt delay mode is enabled, the SINT pin may be asserted early before the delay expiration if the total number of bytes pending read by the EHC is greater than the asynchronous packet transmit threshold, **ATTHR**. The interrupt delay feature is exclusive to the asynchronous channel and is not extended to the control channel or DCI.

# OS81212/4/6

## 8.7 Control / Status Registers

A register interface is provided to permit the EHC software driver to configure the SPI Port and access port status. These registers are accessed using register read and write transactions as described in [Section 8.3](#).

**TABLE 8-3: SPI REGISTER SUMMARY**

| Address | Name   | Bit<br>31/23/15/7 | Bit<br>30/22/14/6 | Bit<br>29/21/13/5 | Bit<br>28/20/12/4 | Bit<br>27/19/11/3 | Bit<br>26/18/10/2 | Bit<br>25/17/9/1 | Bit<br>24/16/8/0 |         |  |
|---------|--------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|---------|--|
| 0x0     | CONFIG | 31:24             | CRTHR<3:0>        |                   |                   |                   | ARTHR<3:0>        |                  |                  |         |  |
|         |        | 23:16             | —                 | —                 | —                 | —                 | ATTHR<3:0>        |                  |                  |         |  |
|         |        | 15:8              | —                 | —                 | —                 | —                 | —                 | —                | CDLY             | ADLY    |  |
|         |        | 7:0               | INTDLY<7:0>       |                   |                   |                   |                   |                  |                  |         |  |
| 0x1     | CHSTS  | 31:24             | INTDIS            | —                 | —                 | DEISM             | CTISM             | CRISM            | ATISM            | ARISM   |  |
|         |        | 23:16             | —                 | —                 | —                 | DEIS              | CTIS              | CRIS             | ATIS             | ARIS    |  |
|         |        | 15:8              | PCE               | DCIES             | —                 | —                 | —                 | —                | —                | —       |  |
|         |        | 7:0               | CRES              | CRCA              | CTES              | CTCA              | ARES              | ARCA             | ATES             | ATCA    |  |
| 0x2     | DBC    | 31:24             | LEN<7:0>          |                   |                   |                   |                   |                  |                  |         |  |
|         |        | 23:16             | BS<7:0>           |                   |                   |                   |                   |                  |                  |         |  |
|         |        | 15:8              | —                 | —                 | —                 | —                 | —                 | —                | —                | —       |  |
|         |        | 7:0               | —                 | —                 | —                 | —                 | —                 | —                | —                | —       |  |
| 0x3     | DCTRL  | 31:24             | —                 | —                 | —                 | —                 | —                 | ERRM             | NTFM             | CMDDNM  |  |
|         |        | 23:16             | —                 | —                 | —                 | —                 | —                 | ERR              | NTF              | CMDDN   |  |
|         |        | 15:8              | —                 | —                 | —                 | —                 | —                 | —                | —                | —       |  |
|         |        | 7:0               | —                 | —                 | —                 | —                 | —                 | —                | CMDREQ           | SYNCREQ |  |
| 0x4     | ARXB   | 31:24             | RXPCNT<15:8>      |                   |                   |                   |                   |                  |                  |         |  |
|         |        | 23:16             | RXPCNT<7:0>       |                   |                   |                   |                   |                  |                  |         |  |
|         |        | 15:8              | RXFREE<15:8>      |                   |                   |                   |                   |                  |                  |         |  |
|         |        | 7:0               | RXFREE<7:0>       |                   |                   |                   |                   |                  |                  |         |  |
| 0x5     | ATXB   | 31:24             | TXPCNT<15:8>      |                   |                   |                   |                   |                  |                  |         |  |
|         |        | 23:16             | TXPCNT<7:0>       |                   |                   |                   |                   |                  |                  |         |  |
|         |        | 15:8              | TXAVAIL<15:8>     |                   |                   |                   |                   |                  |                  |         |  |
|         |        | 7:0               | TXAVAIL<7:0>      |                   |                   |                   |                   |                  |                  |         |  |
| 0x6     | CRXB   | 31:24             | RXPCNT<15:8>      |                   |                   |                   |                   |                  |                  |         |  |
|         |        | 23:16             | RXPCNT<7:0>       |                   |                   |                   |                   |                  |                  |         |  |
|         |        | 15:8              | RXFREE<15:8>      |                   |                   |                   |                   |                  |                  |         |  |
|         |        | 7:0               | RXFREE<7:0>       |                   |                   |                   |                   |                  |                  |         |  |
| 0x7     | CTXB   | 31:24             | TXPCNT<15:8>      |                   |                   |                   |                   |                  |                  |         |  |
|         |        | 23:16             | TXPCNT<7:0>       |                   |                   |                   |                   |                  |                  |         |  |
|         |        | 15:8              | TXAVAIL<15:8>     |                   |                   |                   |                   |                  |                  |         |  |
|         |        | 7:0               | TXAVAIL<7:0>      |                   |                   |                   |                   |                  |                  |         |  |

## REGISTER 8-1: CONIG: SPI CONFIGURATION

|             |       |       |       |            |       |       |       |
|-------------|-------|-------|-------|------------|-------|-------|-------|
| R/W-0       | R/W-0 | R/W-0 | R/W-0 | R/W-0      | R/W-0 | R/W-0 | R/W-0 |
| CRTHR<3:0>  |       |       |       | ARTHR<3:0> |       |       |       |
| 31          | 30    | 29    | 28    | 27         | 26    | 25    | 24    |
| U-0         | U-0   | U-0   | U-0   | R/W-0      | R/W-0 | R/W-0 | R/W-0 |
| —           | —     | —     | —     | ATTHR<3:0> |       |       |       |
| 23          | 22    | 21    | 20    | 19         | 18    | 17    | 16    |
| U-0         | U-0   | U-0   | U-0   | U-0        | U-0   | R/W-0 | R/W-0 |
| —           | —     | —     | —     | —          | —     | CDLY  | ADLY  |
| 15          | 14    | 13    | 12    | 11         | 10    | 9     | 8     |
| R/W-0       | R/W-0 | R/W-0 | R/W-0 | R/W-0      | R/W-0 | R/W-0 | R/W-0 |
| INTDLY<7:0> |       |       |       |            |       |       |       |
| 7           | 6     | 5     | 4     | 3          | 2     | 1     | 0     |

|                       |  |                      |                  |
|-----------------------|--|----------------------|------------------|
| Legend:               |  |                      |                  |
| R = Readable bit      | W = Writable bit                               | P = Programmable bit | r = Reserved bit |
| U = Unimplemented bit | -n = Bit Value at POR: ('0', '1', x = Unknown) |                      |                  |

bit 27-24     **CRTHR<3:0>**: Control Packet Receive Buffer Free Threshold  
 Specifies the minimum number of free bytes available in the SPI control packet buffer which the EHC may write before the **CRIS** status bit is set pending.

- 0000 = 8 bytes
- 0001 = 16 bytes
- 0010 = 24 bytes
- 0011 = 32 bytes
- 0100 = 48 bytes
- 0101 = 64 bytes
- 0110 = 80 bytes
- 0111 = 96 bytes
- 1000 = 128 bytes
- 1001 = 256 bytes
- 1010 = 512 bytes

All other encoding values are reserved and should not be used.

bit 27-24     **ARTHR<3:0>**: Asynchronous Packet Receive Buffer Free Threshold  
 Specifies the minimum number of free bytes available in the SPI asynchronous packet buffer which the EHC may write before the **ARIS** status bit is set pending.

- 0000 = 64 bytes
- 0001 = 96 bytes
- 0010 = 128 bytes
- 0011 = 192 bytes
- 0100 = 256 bytes
- 0101 = 384 bytes
- 0110 = 512 bytes
- 0111 = 768 bytes
- 1000 = 1024 bytes
- 1001 = 1536 bytes
- 1010 = 2048 bytes
- 1011 = 3072 bytes

All other encoding values are reserved and should not be used.

bit 23-20     **Unimplemented**: Write as '0'

- bit 19-16    **ATTHR<3:0>**: Asynchronous Packet Transmit Buffer Available Threshold  
Specifies the minimum number of bytes of asynchronous packet data available in the SPI asynchronous packet buffer available for the EHC to read.  
This field is only valid if the asynchronous packet transmit interrupt delay (**ADLY**) is enabled.  
0000 = 64 bytes  
0001 = 96 bytes  
0010 = 128 bytes  
0011 = 192 bytes  
0100 = 256 bytes  
0101 = 384 bytes  
0110 = 512 bytes  
0111 = 768 bytes  
1000 = 1024 bytes  
1001 = 1536 bytes  
1010 = 2048 bytes  
1011 = 3072 bytes
- All other encoding values are reserved and should not be used.
- bit 15-10    **Unimplemented**: Write as '0'
- bit 9        **CDLY**: Interrupt delay mode for control packet transmit  
1 = If enabled and not masked, assertion of the **SINT** pin will be delayed from when the control transmit interrupt status (**CTIS**) is set.  
0 = If enabled and not masked, the **SINT** pin will be immediately asserted when the control transmit interrupt status (**CTIS**) is set.
- bit 8        **ADLY**: Interrupt delay mode for asynchronous packet transmit  
1 = If enabled and not masked, assertion of the **SINT** pin will be delayed from when the asynchronous transmit interrupt status (**ATIS**) is set.  
0 = If enabled and not masked, the **SINT** pin will be immediately asserted when the asynchronous transmit interrupt status (**ATIS**) is set.
- bit 7-0     **INTDLY<7:0>**: Interrupt Delay Count. Specifies a period of time which to delay packet channel transmit ready interrupts. The delay is specified in number network frames.  
00000000 = 1/Fs seconds (20.8  $\mu$ s at Fs = 48 KHz)  
00000001 = 2/Fs seconds (41.7  $\mu$ s at Fs = 48 KHz)  
00000010 = 3/Fs seconds (62.5  $\mu$ s at Fs = 48 KHz)  
...  
01111111 = 128/Fs seconds (2.67 ms at Fs = 48 KHz)  
...  
11111111 = 256/Fs seconds (5.33 ms at Fs = 48 KHz)
- The delay counter is derived from an internal frame marker. Therefore, if a status interrupt bit becomes pending immediately prior to the next frame marker, the effective delay before **SINT** is asserted may be one frame period less than specified (20.8  $\mu$ s at Fs = 48 KHz).

## REGISTER 8-4: CHSTS: CHANNEL STATUS

|        |       |       |       |       |       |       |       |
|--------|-------|-------|-------|-------|-------|-------|-------|
| R/W-0  | U-0   | U-0   | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| INTDIS | —     | —     | DEISM | CTISM | CRISM | ATISM | ARISM |
| 31     | 30    | 29    | 28    | 27    | 26    | 25    | 24    |
| U-0    | U-0   | U-0   | R-0   | R-0   | R-0   | R-0   | R-0   |
| —      | —     | —     | DEIS  | CTIS  | CRIS  | ATIS  | ARIS  |
| 23     | 22    | 21    | 20    | 19    | 18    | 17    | 16    |
| R/W-0  | R/W-0 | U-0   | U-0   | U-0   | U-0   | U-0   | U-0   |
| PCE    | DCIES | —     | —     | —     | —     | —     | —     |
| 15     | 14    | 13    | 12    | 11    | 10    | 9     | 8     |
| R/W-0  | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| CRES   | CRCA  | CTES  | CTCA  | ARES  | ARCA  | ATES  | ATCA  |
| 7      | 6     | 5     | 4     | 3     | 2     | 1     | 0     |

|                       |  |                      |                  |
|-----------------------|--|----------------------|------------------|
| Legend:               |  |                      |                  |
| R = Readable bit      | W =Writable bit                                | P = Programmable bit | r = Reserved bit |
| U = Unimplemented bit | -n = Bit Value at POR: ('0', '1', x = Unknown) |                      |                  |

- bit 31      **INTDIS:** Global Interrupt Enable/Disable bit  
 0 = The SINT pin is enabled and controlled by the unmasked interrupt status bits.  
 1 = The SINT pin is disabled and driven high.
- bit 30-29    **Unimplemented:** Write as '0'
- bit 28      **DEISM:** DCI Event Interrupt Status Mask  
 0 = The SINT pin will be asserted when the DCI Event Interrupt Status (**DEIS**) bit is set.  
 1 = The DCI Event Interrupt Status (**DEIS**) bit is masked and prevented from asserting SINT when set.
- bit 27      **CTISM:** Control Channel Transmit Interrupt Status Mask  
 0 = The SINT pin will be asserted when the Control Channel Transmit Status (**CTIS**) bit is set.  
 1 = The Control Channel Transmit Status (**CTIS**) bit is masked and prevented from asserting SINT when set.
- bit 26      **CRISM:** Control Channel Receive Interrupt Status Mask  
 0 = The SINT pin will be asserted when the Control Channel Receive Status (**CRIS**) bit is set.  
 1 = The Control Channel Receive Status (**CRIS**) bit is masked and prevented from asserting SINT when set.
- bit 25      **ATISM:** Asynchronous Channel Transmit Interrupt Status Mask  
 0 = The SINT pin will be asserted when the Asynchronous Channel Transmit Status (**ATIS**) bit is set.  
 1 = The Asynchronous Channel Transmit Status (**ATIS**) bit is masked and prevented from asserting SINT when set.
- bit 24      **ARISM:** Asynchronous Channel Receive Interrupt Status Mask  
 0 = The SINT pin will be asserted when the Asynchronous Channel Receive Status (**ARIS**) bit is set.  
 1 = The Asynchronous Channel Receive Status (**ARIS**) bit is masked and prevented from asserting SINT when set.
- bit 23-21    **Unimplemented:** Write as '0'
- bit 20      **DEIS:** DCI Event Interrupt Status  
 0 = The DCI has no status event pending for the EHC to read.  
 1 = The DCI has a status event pending for the EHC to read.
- bit 19      **CTIS:** Control Channel Transmit Interrupt Status  
 0 = The control packet channel has no packets pending for the EHC to read.  
 1 = Packets are available for the EHC to read from the control packet channel. The number of packets and total number of bytes available for the EHC to read are specified in the CTXB register.

|          |   |
|----------|---|
| bit 18   | <b>CRIS:</b> Control Channel Receive Interrupt Status<br>0 = The control packet channel cannot receive packets written by the EHC.<br>1 = Packets may be written by the EHC to the control packet channel. The number of packets and total number of bytes that may be written by the EHC are specified in the CRXB register.                           |
| bit 17   | <b>ATIS:</b> Asynchronous Channel Transmit Interrupt Status<br>0 = The asynchronous packet channel has no packets pending for the EHC to read.<br>1 = Packets are available for the EHC to read from the asynchronous packet channel. The number of packets and total number of bytes available for the EHC to read are specified in the ATXB register. |
| bit 16   | <b>ARIS:</b> Asynchronous Channel Receive Interrupt Status<br>0 = The asynchronous packet channel cannot receive packets written by the EHC.<br>1 = Packets may be written by the EHC to the asynchronous packet channel. The number of packets and total number of bytes that may be written by the EHC are specified in the ARXB register.            |
| bit 15   | <b>PCE:</b> SPI Protocol Command Error<br>0 = No errors detected.<br>1 = An SPI protocol error was encountered. This bit is sticky when set and the EHC must write a '1' to clear the error state.  |
| bit 14   | <b>DCIES:</b> DCI Error Status<br>0 = No errors detected.<br>1 = An error was detected while the EHC was writing a <i>DCI Command</i> or reading a <i>DCI Response</i> . Data loss may have occurred. This bit is sticky when set and the EHC must write a '1' to clear the error state.  |
| bit 13-8 | <b>Unimplemented:</b> Write as '0'  |
| bit 7    | <b>CRES:</b> Control Receive Error Status<br>0 = No errors detected.<br>1 = An error was detected while the EHC was writing a packet to the control packet channel. Data loss may have occurred. This bit is sticky when set and the EHC must write a '1' to clear the error state.   |
| bit 6    | <b>CRCA:</b> Control Receive Channel Availability<br>0 = The control packet channel is not available for the EHC to write packet data.<br>1 = The control packet channel has been enabled and is available to receive packet data from the EHC.   |
| bit 5    | <b>CTES:</b> Control Transmit Error Status<br>0 = No errors detected.<br>1 = An error was detected while the EHC was reading a packet from the control packet channel. Data loss may have occurred. This bit is sticky when set and the EHC must write a '1' to clear the error state.  |
| bit 4    | <b>CTCA:</b> Control Transmit Channel Availability<br>0 = The control packet channel is not available for the EHC to receive packet data.<br>1 = The control packet channel has been enabled and is available to transmit packet data to the EHC.   |
| bit 3    | <b>ARES:</b> Asynchronous Receive Error Status<br>0 = No errors detected.<br>1 = An error was detected while the EHC was writing a packet to the asynchronous packet channel. Data loss may have occurred. This bit is sticky when set and the EHC must write a '1' to clear the error state.   |
| bit 2    | <b>ARCA:</b> Asynchronous Receive Channel Availability<br>0 = The asynchronous packet channel is not available for the EHC to write packet data.<br>1 = The asynchronous packet channel has been enabled and is available to receive packet data from the EHC.  |
| bit 1    | <b>ATES:</b> Asynchronous Transmit Error Status<br>0 = No errors detected.<br>1 = An error was detected while the EHC was reading a packet from the asynchronous packet channel. Data loss may have occurred. This bit is sticky when set and the EHC must write a '1' to clear the error state.  |
| bit 0    | <b>ATCA:</b> Asynchronous Transmit Channel Availability<br>0 = The asynchronous packet channel is not available for the EHC to receive packet data.<br>1 = The asynchronous packet channel has been enabled and is available to transmit packet data to the EHC.  |

**Note:** Do not attempt to write to the DBC register.

## REGISTER 8-5: DBC: DCI BUFFER CONFIGURATION

|          |     |     |     |     |     |     |     |
|----------|-----|-----|-----|-----|-----|-----|-----|
| R-0      | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| LEN<7:0> |     |     |     |     |     |     |     |
| 31       | 30  | 29  | 28  | 27  | 26  | 25  | 24  |
| R-0      | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| BS<7:0>  |     |     |     |     |     |     |     |
| 23       | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
| r-0      | r-0 | r-0 | r-0 | r-0 | r-0 | r-0 | r-0 |
| —        | —   | —   | —   | —   | —   | —   | —   |
| 15       | 14  | 13  | 12  | 11  | 10  | 9   | 8   |
| r-0      | r-0 | r-0 | r-0 | r-0 | r-0 | r-0 | r-0 |
| —        | —   | —   | —   | —   | —   | —   | —   |
| 7        | 6   | 5   | 4   | 3   | 2   | 1   | 0   |

**Legend:**

R = Readable bit      W = Writable bit      P = Programmable bit      r = Reserved bit  
 U = Unimplemented bit      -n = Bit Value at POR: ('0', '1', x = Unknown)

bit 31-24      **LEN<7:0>**: Length of *DCI Command* written or pending *DCI Response*

bit 23-16      **BS<15:0>**: DCI Buffer Size

bit 15-0      **Reserved**: Reserved

## REGISTER 8-6: DCTRL: DCI CONTROL

|     |     |     |     |     |       |        |         |
|-----|-----|-----|-----|-----|-------|--------|---------|
| U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 | R/W-0  | R/W-0   |
| —   | —   | —   | —   | —   | ERRM  | NTFM   | CMDDNM  |
| 31  | 30  | 29  | 28  | 27  | 26    | 25     | 24      |
| U-0 | U-0 | U-0 | U-0 | U-0 | R-0   | R-0    | R-0     |
| —   | —   | —   | —   | —   | ERR   | NTF    | CMDDN   |
| 23  | 22  | 21  | 20  | 19  | 18    | 17     | 16      |
| U-0 | U-0 | U-0 | U-0 | U-0 | U-0   | U-0    | U-0     |
| —   | —   | —   | —   | —   | —     | —      | —       |
| 15  | 14  | 13  | 12  | 11  | 10    | 9      | 8       |
| U-0 | U-0 | U-0 | U-0 | U-0 | U-0   | R/W-0  | R/W-0   |
| —   | —   | —   | —   | —   | —     | CMDREQ | SYNCREQ |
| 7   | 6   | 5   | 4   | 3   | 2     | 1      | 0       |

**Legend:**

R = Readable bit      W = Writable bit      P = Programmable bit      r = Reserved bit  
 U = Unimplemented bit      -n = Bit Value at POR: ('0', '1', x = Unknown)

# OS81212/4/6

- bit 31-27 **Unimplemented:** Write as '0'
- bit 26 **ERRM:** DCI Error Interrupt Status Mask  
 0 = The **SINT** pin will be asserted when the DCI Error (**ERR**) status bit is set.  
 1 = The DCI Error (**ERR**) status bit is masked and prevented from asserting **SINT** when set.
- bit 25 **NTFM:** DCI Notification Interrupt Status Mask  
 0 = The **SINT** pin will be asserted when the DCI Notification (**NTF**) status bit is set.  
 1 = The DCI Notification (**NTF**) status bit is masked and prevented from asserting **SINT** when set.
- bit 24 **CMDDNM:** DCI Command Done Interrupt Status Mask  
 0 = The **SINT** pin will be asserted when the DCI Command Done (**CMDCN**) status bit is set.  
 1 = The DCI Command Done (**CMDDN**) status bit is masked and prevented from asserting **SINT** when set.
- bit 23-19 **Unimplemented:** Write as '0'
- bit 18 **ERR:** DCI Error Status  
 0 = No error detected.  
 1 = A fatal error was detected on the DCI by the INIC. The EHC must write a '1' to clear.
- bit 17 **NTF:** DCI Notification Status  
 0 = The EHC clears this bit by writing a '1' to acknowledge the status change notification.  
 1 = The INIC sets this bit to notify the EHC of a DCI status change. The EHC should respond by reading the DCI virtual registers.
- bit 16 **CMDDN:** DCI Command Done Status  
 0 = The EHC clears this bit by writing a '1' after reading the *DCI Response*.  
 1 = The INIC sets this bit to indicate that a *DCI Response* packet is available for reading from the DCI channel by the EHC.
- bit 15-2 **Unimplemented:** Write as '0'
- bit 0 **CMDREQ:** Command Request  
 0 = No command. The INIC clears this bit when it has executed the *DCI Command*.  
 1 = The EHC writes a '1' setting this bit to request the INIC to execute a *DCI Command* packet that has been written to the DCI channel.
- bit 0 **SYNCREQ:** Driver Synchronization Request  
 0 = No synchronization requested. The INIC clears this bit.  
 1 = The EHC writes a '1' setting this bit to request DCI synchronization.

## REGISTER 8-7: ARXB: ASYNCHRONOUS PACKET RECEIVE BUFFER STATUS

|              |     |     |     |     |     |     |     |
|--------------|-----|-----|-----|-----|-----|-----|-----|
| R-0          | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| RXPCNT<15:8> |     |     |     |     |     |     |     |
| 31           | 30  | 29  | 28  | 27  | 26  | 25  | 24  |
| R-0          | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| RXPCNT<7:0>  |     |     |     |     |     |     |     |
| 23           | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
| R-0          | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| RXFREE<15:8> |     |     |     |     |     |     |     |
| 15           | 14  | 13  | 12  | 11  | 10  | 9   | 8   |
| R-0          | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| RXFREE<7:0>  |     |     |     |     |     |     |     |
| 7            | 6   | 5   | 4   | 3   | 2   | 1   | 0   |

|                       |  |                      |                  |
|-----------------------|--|----------------------|------------------|
| Legend:               |  |                      |                  |
| R = Readable bit      | W = Writable bit                               | P = Programmable bit | r = Reserved bit |
| U = Unimplemented bit | -n = Bit Value at POR: ('0', '1', x = Unknown) |                      |                  |



bit 31-16    **RXPCNT<15:0>**: Number of asynchronous packets available to be written by EHC  
bit 15-0     **RXFREE<15:0>**: Number of free buffer bytes available to be written by EHC

## REGISTER 8-8:    **ATXB: ASYNCHRONOUS PACKET TRANSMIT BUFFER STATUS**

|               |     |     |     |     |     |     |     |
|---------------|-----|-----|-----|-----|-----|-----|-----|
| R-0           | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| TXPCNT<15:8>  |     |     |     |     |     |     |     |
| 31            | 30  | 29  | 28  | 27  | 26  | 25  | 24  |
| R-0           | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| TXPCNT<7:0>   |     |     |     |     |     |     |     |
| 23            | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
| R-0           | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| TXAVAIL<15:8> |     |     |     |     |     |     |     |
| 15            | 14  | 13  | 12  | 11  | 10  | 9   | 8   |
| R-0           | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| TXAVAIL<7:0>  |     |     |     |     |     |     |     |
| 7             | 6   | 5   | 4   | 3   | 2   | 1   | 0   |

**Legend:**

R = Readable bit                      W = Writable bit                      P = Programmable bit                      r = Reserved bit  
U = Unimplemented bit                -n = Bit Value at POR: ('0', '1', x = Unknown)

bit 31-16    **TXPCNT<15:0>**: Number of asynchronous packets available to be read by EHC  
bit 15-0     **TXAVAIL<15:0>**: Number of bytes available to be read by EHC

## REGISTER 8-9:    **CRXB: CONTROL PACKET RECEIVE BUFFER STATUS**

|              |     |     |     |     |     |     |     |
|--------------|-----|-----|-----|-----|-----|-----|-----|
| R-0          | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| RXPCNT<15:8> |     |     |     |     |     |     |     |
| 31           | 30  | 29  | 28  | 27  | 26  | 25  | 24  |
| R-0          | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| RXPCNT<7:0>  |     |     |     |     |     |     |     |
| 23           | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
| R-0          | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| RXFREE<15:8> |     |     |     |     |     |     |     |
| 15           | 14  | 13  | 12  | 11  | 10  | 9   | 8   |
| R-0          | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| RXFREE<7:0>  |     |     |     |     |     |     |     |
| 7            | 6   | 5   | 4   | 3   | 2   | 1   | 0   |

**Legend:**

R = Readable bit                      W = Writable bit                      P = Programmable bit                      r = Reserved bit  
U = Unimplemented bit                -n = Bit Value at POR: ('0', '1', x = Unknown)

bit 31-16    **RXPCNT<15:0>**: Number of control packets available to be written by EHC  
bit 15-0     **RXFREE<15:0>**: Number of free buffer bytes available to be written by EHC

## REGISTER 8-10: CTXB: CONTROL PACKET TRANSMIT BUFFER STATUS

|               |     |     |     |     |     |     |     |
|---------------|-----|-----|-----|-----|-----|-----|-----|
| R-0           | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| TXPCNT<15:8>  |     |     |     |     |     |     |     |
| 31            | 30  | 29  | 28  | 27  | 26  | 25  | 24  |
| R-0           | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| TXPCNT<7:0>   |     |     |     |     |     |     |     |
| 23            | 22  | 21  | 20  | 19  | 18  | 17  | 16  |
| R-0           | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| TXAVAIL<15:8> |     |     |     |     |     |     |     |
| 15            | 14  | 13  | 12  | 11  | 10  | 9   | 8   |
| R-0           | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| TXAVAIL<7:0>  |     |     |     |     |     |     |     |
| 7             | 6   | 5   | 4   | 3   | 2   | 1   | 0   |

|                       |                 |  |                  |
|-----------------------|-----------------|--|------------------|
| Legend:               |                 |  |                  |
| R = Readable bit      | W =Writable bit | P = Programmable bit                           | r = Reserved bit |
| U = Unimplemented bit |                 | -n = Bit Value at POR: ('0', '1', x = Unknown) |                  |

bit 31-16     **TXPCNT<15:0>**: Number of control packets available to be read by EHC  
bit 15-0     **TXAVAIL<15:0>**: Number of bytes available to be read by EHC

## 9.0 EXTERNAL POWER MANAGEMENT

The OS81212/4/6 INIC supports external power management logic for controlling Electronic Control Unit (ECU) power by providing two inputs (**PS0** and **PS1**) and one output (**PWROFF**). The **PS0** and **PS1** inputs communicate power status from the ECU external power management logic to the INIC. When a power management event occurs, it should be communicated to the INIC through **PS0** and **PS1** by the external power management circuitry. The functional device behavior that occurs in response to an appropriate external power management status event is described in the [INIC Interface Specification \[4\]](#) and can be configured depending on the application requirements (e.g. OEM specifications). The **PWROFF** output pin is driven low during normal operating conditions and released when the OS81212/4/6 INIC is ready to be powered-off by the external power management logic.

If external power management is not used, **PS0** and **PS1** should be pulled to ground (to indicate  $U_{Normal}$ ) or power management should be disabled within the *INIC Configuration String* (see the [INIC Interface Specification \[4\]](#)). When power management is disabled, the **PS0** and **PS1** pins are available for use as GPIO signals.

[Table 9-1](#) describes the power management states defined by the [ISO 21806 \(Parts 1-7\): 2020 – Road Vehicles – Media Oriented Systems Transport \(MOST\) Specifications \[1\]](#) and [ISO 21806 \(Parts 14-15\): 2021 – Road Vehicles – Media Oriented Systems Transport \(MOST\) Lean Application Layer Specification \[2\]](#) should be communicated to the INIC through **PS0** and **PS1**.

**TABLE 9-1: EXTERNAL POWER MANAGEMENT STATES**

| PS1 | PS0 | Status                       |
|-----|-----|------------------------------|
| 0   | 0   | $U_{Normal}$                 |
| 0   | 1   | Switch-To-Power (STP)        |
| 1   | 0   | $U_{Critical}$ , $U_{Super}$ |
| 1   | 1   | $U_{Low}$                    |

The INIC recognizes an external power management state when the **PS0** and **PS1** remain at a valid logic level for a minimum time,  $t_{pshold}$  (see [Section 14.4.2 "Reset and Configuration"](#)); however, the *INIC Software Stack* may use longer hold times for **PS0** and **PS1** states before responding to an external power management event. In addition, the *INIC Software Stack* controls how INIC responds to external power management events (e.g. driving **PWROFF** and/or forcing network shutdown).

A Switch-To-Power (STP) event is recognized as the INIC samples the **PS0** and **PS1** pins at power-up (or reset) and on the *transition* to **PS[1:0] = 01**. If these pins continue to indicate an STP status (**PS[1:0] = 01**) after an STP event has already been recognized, then the  $U_{Normal}$  state is assumed.

[Figure 9-1](#) illustrates a typical application in which the external power management circuitry is powered by a continuous supply and manages the switched power to the rest of the node. This circuitry also monitors the activity signal (**BSTATUS**) coming from the OS81212/4/6 network activity detector. When network activity is detected, the circuitry enables the switched supply and drives the **PS[1:0]** pins to the proper state, as depicted in [Figure 14-1](#). When switched power is applied to INIC, it could take up to  $t_{stp}$  time for INIC to recognize the **PS[1:0]** state and **PWROFF** to become valid. When INIC is ready to be shut down (such as when  $U_{Low}$  state is recognized), it releases the **PWROFF** pin (pulled high through an external resistor).

Additionally, **ERR** is driven high and the network output is switched off when the  $U_{Low}$  state is recognized. As long as the  $U_{Low}$  state is indicated on **PS[1:0]**, the network port outputs remain disabled, regardless of activity on the network port inputs.

If the EHC requires time after power-up (or reset) for initialization, it could utilize a **PWRHOLD** line that the power management circuitry detects. The use of **PWRHOLD** would suspend node power-down until the EHC is ready.

The [INIC Interface Specification \[4\]](#) contains more detailed information on configuring various parameters of the External Power Management function, such as the behavior of **PWROFF**.

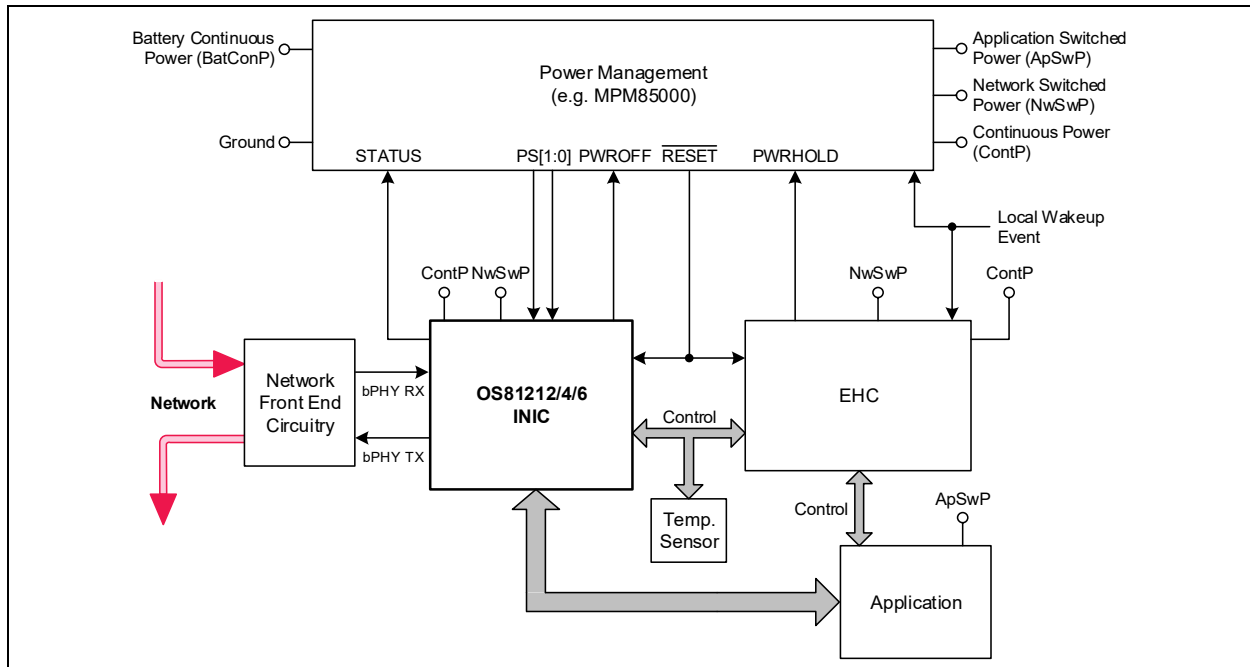
INIC supports implementation of a low-power mode for an ECU that requires power conservation while keeping INIC powered and active on the network. A message to enter a low-power mode is initiated from the network and passed by INIC to the EHC. When the EHC receives this message, it may power down certain areas of the ECU and, if supported, the EHC may then enter into a low-power or sleep mode. INIC will wake the EHC by driving **INT** low when any message initiated from the network targets the EHC. This EHC wake feature is only supported when the I<sup>2</sup>C port is configured as a slave device and selected as the configuration interface.

# OS81212/4/6

The Microchip *Automotive Power Management Device* (MPM85000) performs all power management functionality required of network ECUs including wake detection, power supply monitoring and status (**PS[1:0]**), reset generation, and optional *Switch To Power* (STP) pulse detection. The MPM85000 also supplies a 3.3 V continuous power supply output for the OS81212/4/6 **VDDAU<sub>n</sub>** pins. Integrating both the OS81212/4/6 and MPM85000 on a network node minimizes application circuitry and software complexity, allowing the application designer to focus on the application. Contact Microchip for more information on the MPM85000.

Figure 9-1 illustrates the fundamental hardware architecture of a network ECU.

**FIGURE 9-1: TYPICAL POWER MANAGEMENT CIRCUITRY**



## 10.0 CLOCK MANAGER

The Clock Manager generates internal clocks and optionally drives an external clock (**RMCK**) on the RMCK Port. Refer to [Chapter 11.0 "RMCK Port"](#) for more information on the functionality and configuration of **RMCK**.

### 10.1 PLL Lock Status

After power-up, the internal analog PLL locks to an external clock reference (**XTI/XTO**) and provides the primary time-base for the OS81212/4/6. A digital PLL performs clock and data recovery at the Network Port. For proper data and clock recovery of the incoming network bit stream, a valid and continuous reference clock must always be available.

When network lock is initially achieved, it takes three frames to synchronize streaming data. Therefore, streaming data will not be transferred properly between the network and any open source or sink ports for three frames when going from the unlock to lock state. While the PLL is unlocked, only control messages targeting the local INIC can be communicated with the EHC.

**Note:** A valid and continuous external reference clock (**XTI/XTO**) must be present for normal operation. If the reference clock is disabled during operation, INIC may become inoperable and/or the state of the **ERR** pin may become invalid. A reset and valid reference clock is required to recover from this condition.

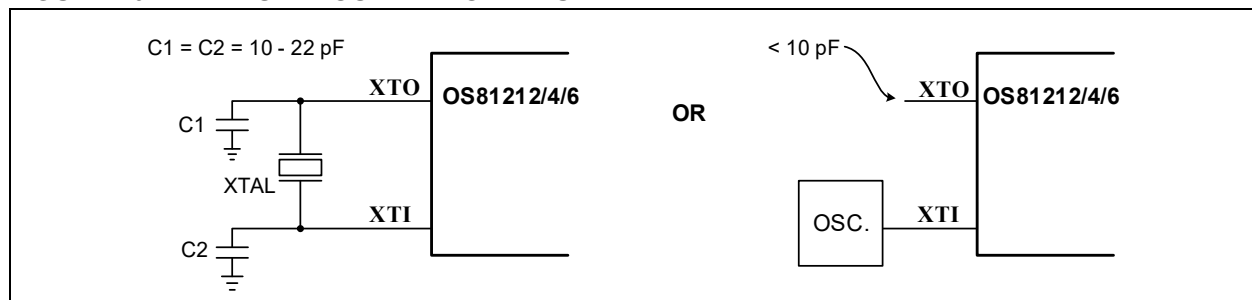
### 10.2 Crystal Pins (**XTI/XTO**)

The crystal oscillator should be in a fundamental, parallel resonant mode. [Figure 10-1](#) depicts the external circuitry connected to the OS81212/4/6 oscillator circuit. Since the internal inverter/amplifier is operated in its linear region, external series resistors should not be used as they will lower the gain and could cause start-up problems. Several factors must be considered when selecting a crystal including load capacitance, oscillator margin, cut, and operating temperature. For more information on crystals, see [Section 15.8 "Crystal Oscillator Selection"](#). The crystal frequency must be 384 times the desired operational network frequency ( $F_s$ ).

If an external clock is used in lieu of a crystal oscillator, it must be connected to **XTI**. The clock must be stable prior to the rising edge of **RST** and remain stable for proper operation. In addition, **XTO** should float and have minimal capacitance (see [Figure 10-1](#)).

**Note:** In timing-master applications, any jitter present at **XTI/XTO** is transferred through the Clock Manager to the various OS81212/4/6 clock outputs (e.g. **RMCK**, **MLBCLK**, **BTXP/BTXN**, etc.). To ensure MOST network physical layer compliance at SP1 and optimal operation of all external application interfaces, jitter at **XTI** must be minimized. A local crystal oscillator (and not an external clock source) is strongly recommended.

**FIGURE 10-1: CRYSTAL OSCILLATOR INPUT**



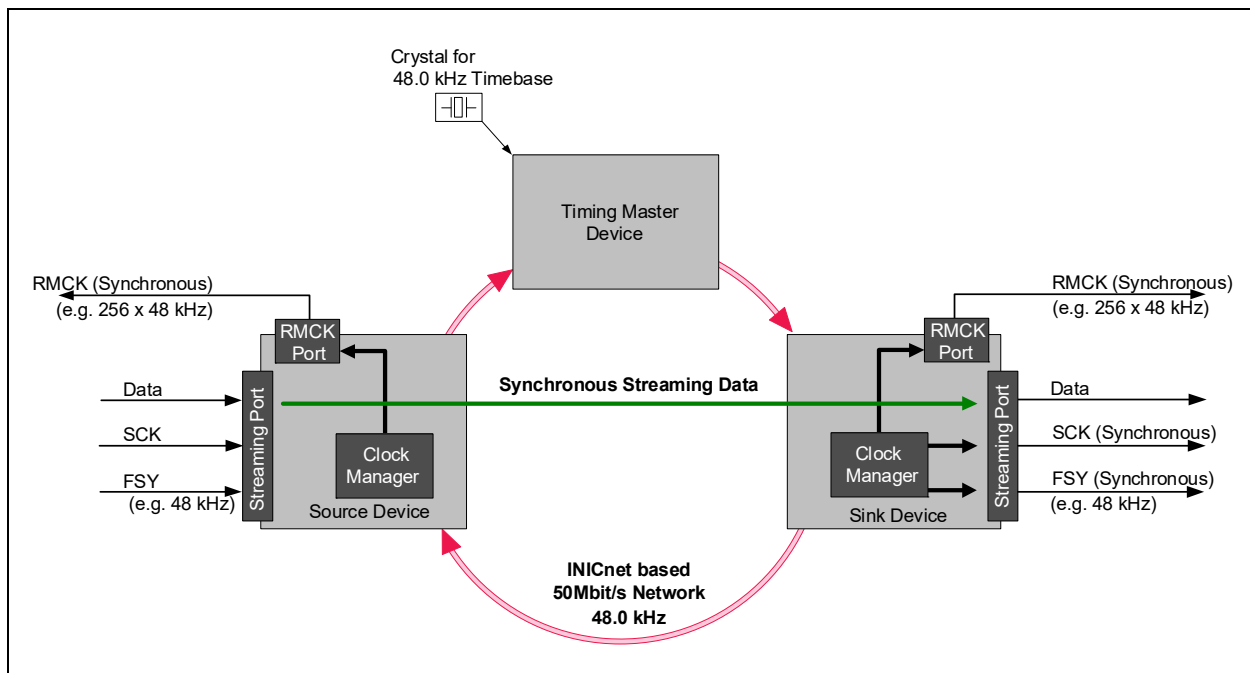
## 11.0 RMCK PORT

The external interface of the RMCK Port consists of a single pin (**RMCK**) which can be configured as a clock output to support synchronization of external devices to a clock multiple synchronous to the network frame rate. Refer to the [INIC Interface Specification \[4\]](#) for more information on configuring the RMCK Port.

### 11.1 Synchronous Output Clock

The **RMCK** pin output is derived by dividing an internal  $3072 \times F_s$  clock that is phase locked to the network frame rate and therefore synchronous to the network. For synchronous streaming data exchanged at the network frequency, the Streaming Port and RMCK Port can be connected to an external device, such as an ADC or DAC, as shown in [Figure 11-1](#).

**FIGURE 11-1: RMCK PORT WITH SYNCHRONOUS OUTPUT CLOCK - EXAMPLE**



## 12.0 GENERAL PURPOSE INPUT/OUTPUT (GPIO) PORT

The OS81212/4/6 allows certain pins to be reprogrammed from their default functionality to support general purpose input/output (GPIO) functionality. Each **GP<sub>n</sub>** and its default function when using standard factory firmware is listed in [Table 12-1](#), [Table 12-2](#) and [Table 12-3](#). The user may override the default function via the *INIC Configuration String* and/ or custom firmware. See the [INIC Interface Specification \[4\]](#) for more details on the alternate configuration options.

**TABLE 12-1: OS81212 GPIO PIN AVAILABILITY**

| Pin | GPIO | Default Function         | Notes   |
|-----|------|--------------------------|---|
| 44  | GP0  | $\overline{\text{INT}}$  | The I <sup>2</sup> C Port is not available in Slave Mode when <b>GP0</b> is used.                                   |
| 34  | GP1  | PS0                      | The External Power Management functionality is not available when either <b>GP1</b> or <b>GP2</b> are used.         |
| 35  | GP2  | PS1                      |   |
| 5   | GP3  | SDIN                     | The SPI Port is not available when any of <b>GP3</b> , <b>GP4</b> , <b>GP5</b> , <b>GP6</b> or <b>GP7</b> are used. |
| 6   | GP4  | SDOUT                    |   |
| 7   | GP5  | SCLK                     |   |
| 8   | GP6  | $\overline{\text{SINT}}$ |   |
| 9   | GP7  | $\overline{\text{CS}}$   |   |
| 39  | GP8  | MUTE                     | Mute indication for the external application is not available when <b>GP8</b> is used.                              |
| 41  | GP10 | GP10                     | Pin dedicated to GP10   |
| 42  | GP12 | GP12                     | Pin dedicated to GP12   |
| 55  | GP13 | FSYB                     | Streaming Port B must be linked to Streaming Port A (SCKA/FSYA) when either <b>GP13</b> or <b>GP14</b> is used.     |
| 56  | GP14 | SCKB                     |   |

**TABLE 12-2: OS81214 GPIO PIN AVAILABILITY**

| Pin | GPIO | Default Function         | Notes   |
|-----|------|--------------------------|---|
| 39  | GP0  | $\overline{\text{INT}}$  | The I <sup>2</sup> C Port is not available in Slave Mode when <b>GP0</b> is used.                                   |
| 34  | GP1  | PS0                      | The External Power Management functionality is not available when either <b>GP1</b> or <b>GP2</b> are used.         |
| 35  | GP2  | PS1                      |   |
| 2   | GP3  | SDIN                     | The SPI Port is not available when any of <b>GP3</b> , <b>GP4</b> , <b>GP5</b> , <b>GP6</b> or <b>GP7</b> are used. |
| 3   | GP4  | SDOUT                    |   |
| 4   | GP5  | SCLK                     |   |
| 5   | GP6  | $\overline{\text{SINT}}$ |   |
| 6   | GP7  | $\overline{\text{CS}}$   |   |
| 35  | GP8  | MUTE                     | Mute indication for the external application is not available when <b>GP8</b> is used.                              |

**TABLE 12-3: OS81216 GPIO PIN AVAILABILITY**

| Pin | GPIO | Default Function        | Notes   |
|-----|------|-------------------------|---|
| 39  | GP0  | $\overline{\text{INT}}$ | The I <sup>2</sup> C Port is not available in Slave Mode when <b>GP0</b> is used.                           |
| 34  | GP1  | PS0                     | The External Power Management functionality is not available when either <b>GP1</b> or <b>GP2</b> are used. |
| 35  | GP2  | PS1                     |   |
| 35  | GP8  | MUTE                    | Mute indication for the external application is not available when <b>GP8</b> is used.                      |

**Note:** The OS81212/4/6  $\overline{\text{INT}}$  pin is also used as part of the communication interface during the device update process. During *Boot Mode*, this pin is configured as an open-drain output and may be driven low for flow control signaling. Therefore, any external circuitry used for GPIO functionality must be able to withstand INIC actively driving **GP0** low.

## 12.1 GPIO Classes

GPIO pins are highly configurable which provides the user with the flexibility to support various applications. The following GPIO pin modes are available:

- *Output (Push-Pull)* - High Level or Low Level
- *Output (Open-Drain)* - High Impedance (Hi-Z) or driven low. Trigger: Rising Edge, Falling Edge, High Level, Low Level
- *Input* - Trigger: Rising Edge, Falling Edge, High Level, Low Level
- *Sticky Input* - Trigger: Rising Edge, Falling Edge
- *Debounced Input* - Trigger: Rising Edge, Falling Edge, High Level, Low Level

At OS81212/4/6 power-up/reset, all GPIO pins are high-impedance inputs. During GPIO initialization, the user specifies the pin configuration for each GPIO pin that is used and configures any other relevant details, such as debounce settings. Refer to Chapter 14.0 for GPIO details such as voltage levels and timing. Further details for each GPIO mode are included in the following subsections and the [INIC Interface Specification \[4\]](#).

### 12.1.1 PUSH-PULL OUTPUT

A *Push-Pull Output* provides a user-controlled output signal that is actively driven both high and low. The default output state of the pin can be configured as either high or low level.

All GPIO pins are high-impedance inputs following power-up and reset; therefore, an external pull-up resistor is required to hold the pin at a logic high level before the GPIO is configured as a *Push-Pull Output*.

### 12.1.2 OPEN-DRAIN OUTPUT

An *Open-Drain Output* provides a user-controlled output signal that is actively driven low and set to high-impedance during the logic high state. An external pull-up resistor is required to establish the logic high level when INIC releases the pin. This GPIO mode is useful for ganged signals for which more than one device can drive the line active (low).

Event notification (triggers) can also be configured on outputs similarly to inputs. For *Open-Drain Outputs*, this functionality is also used to notify the higher-level application that another device has driven the line active low.

All GPIO pins are high-impedance inputs following power-up and reset, therefore an external pull-up resistor is required to hold the pin at a logic high level before the GPIO is configured as an *Open-Drain Output*.

### 12.1.3 INPUT

A GPIO pin configured as an input can generate a notification (trigger) when a logic level (high-level or low-level) is detected or when a signal transition (edge) is detected.

### 12.1.4 STICKY INPUT

A *Sticky Input* allows the user to capture very brief pulses that may occur using dedicated hardware mechanisms and is appropriate for high speed inputs where active conditions may only exist for brief periods of time. The captured value is referred to as the sticky state and is independent of the actual current state of the hardware pin. Once the active condition has been captured, the input is 'stuck' and INIC does not report any further event conditions. Therefore, the captured state must be manually cleared by a higher-level application to re-enable the reporting of event conditions. The main advantage of a *Sticky Input* is the ability to capture narrow pulses that cannot be detected when using other GPIO modes.

Edge-based trigger events on a *Sticky Input* pin remain valid as long as the input signal remains at the active level. Once the event is cleared, INIC will immediately report another event if the active level remains present on the GPIO pin. This behavior is similar to that of *Level-Sensitive Input* GPIO pins; however, in the case of *Sticky Inputs*, this action may result in additional edge-based events being reported when no further edges have actually occurred.

Some considerations must be made with the use of *Sticky Input* pins. One is that the captured state must be manually cleared by a higher-level application in order to re-enable the reporting of subsequent event notifications. Additionally, there are complexities (described above) associated with the handling of an input event that is maintained at the active level rather than appearing as a narrow pulse.



## 12.1.5 DEBOUNCED INPUT

A *Debounced Input* is an *Input* that does not immediately report an event. Rather, the value on the pin is sampled again after a period of time and compared to the initially sampled value. If the values match, the signal is assumed to be stable (valid), the value is captured, and an event is reported. If the values do not match, the original edge event is considered non-qualified and is ignored.

A *Debounced Input* GPIO is useful for capturing events generated by mechanical switches that normally suffer from contact bounce. The time delay allows the signal to settle before reporting the event thereby only reporting a single, clean event to the monitoring application.

Refer to the [INIC Interface Specification \[4\]](#) for more information on configuring a *Debounced Input*.

## 12.2 GPIO Triggers

### 12.2.1 LEVEL-SENSITIVE TRIGGERS

A *Level-Sensitive Trigger* monitors the state of a GPIO pin and generates an event notification when either a high-level or low-level logic is detected. The event notification remains valid as long as the signal stays active (e.g., high or low, depending on configuration).

The *Level-Sensitive Trigger* is appropriate for signals where active levels are maintained until serviced. An advantage of this trigger is that subsequent event triggers are only signaled once the controlling device re-enables event notifications, provided the active level has been maintained. This behavior is desirable in application scenarios where:

- a peripheral attached to the OS81212/4/6 GPIO pin does not de-assert its hardware interrupt between events, or
- interrupt lines from multiple hardware peripherals are ganged together (e.g. using open-drain logic) onto a single OS81212/4/6 GPIO pin.

When using a *Level-Sensitive Trigger*, a higher-level application must manually clear events in order to re-enable the reporting of subsequent event notifications. Details for re-enabling of subsequent event notifications is found within the [INIC Interface Specification \[4\]](#).

### 12.2.2 EDGE-SENSITIVE TRIGGERS

An *Edge-Sensitive Trigger* generates an event notification when signal transitions are detected. This type of trigger is appropriate for detecting debounced transition events on an input signal or open-drain output signal in which subsequent signal transitions are immediately reported to the controlling application.

## 13.0 JTAG PORT

The OS81212/4/6 implements a dedicated, 4-pin Test Access Port (TAP) interface (**TDI**, **TDO**, **TCK** and **TMS**), as defined by the IEEE 1149.1 Test Access Port standard. For more information on the IEEE 1149.1 Test Access Port standard, see [IEEE Standard Test Access Port and Boundary-Scan Architecture \[11\]](#).

The test-data input (**TDI**) and test-data output (**TDO**) pins provide the means of shifting data into and out of the OS81212/4/6 boundary-scan register (BSR). The TAP interface is controlled by the test clock (**TCK**) and the test-mode select (**TMS**) pins. When **TMS** is low, the OS81212/4/6 TAP interface is enabled and the rising edge of **TCK** clocks in data from **TMS** and **TDI**. The TAP state machine is asynchronously reset at power-up and when **TMS** is held high for five rising edges of **TCK**. After reset, the **TDO** pin is high impedance. **TDO** is driven by the TAP controller only when in the *Shift-IR* or *Shift-DR* state.

For proper operation, all OS81212/4/6 TAP interface pins should be pulled high through resistors. A 4.7 k $\Omega$  pull-up resistor is recommended for **TCK**, **TDI**, and **TDO** per the IEEE 1149.1 standard. A 47 k $\Omega$  resistor is recommended for **TMS**.

The OS81212/4/6 TAP interface supports all the mandatory boundary-scan instructions, as specified in the IEEE 1149.1 standard. Supported instructions include:

- **BYPASS** - Selects the bypass register (BYR) to be connected for serial access between **TDI** and **TDO**.
- **EXTEST** - Selects the boundary-scan register (BSR) to be connected for serial access between **TDI** and **TDO**.
- **SAMPLE/PRELOAD** - Captures a snapshot of the data at the digital pins during normal operation, and loads a data pattern into the boundary-scan register (BSR) prior to a new boundary-scan operation.
- **IDCODE** - (optional IEEE 1149.1 instruction) Used to read the Manufacturer ID from the identification register (IDR).

The OS81212/4/6 must be placed in *Boot Mode* before accessing the JTAG Port. In *Boot Mode*, the *INIC Software Stack* is disabled and the OS81212/4/6 waits for manual access through the JTAG Port. In this mode, standard communication channels (e.g. I<sup>2</sup>C Port, MediaLB Port) are unavailable. The following sequence should be followed when using the OS81212/4/6 JTAG Port:

- Place the OS81212/4/6 in *Boot Mode*:
  - Hold the OS81212/4/6 in reset (drive  $\overline{\text{RST}}$  pin low),
  - Hold the  $\overline{\text{BOOT}}$  pin low,
  - Release the OS81212/4/6 from reset ( $\overline{\text{RST}}$  pin high),
- Perform JTAG operations,
- Release the  $\overline{\text{BOOT}}$  pin, and
- Reset the OS81212/4/6 to resume a normal mode of operation.

**Note:** The  $\overline{\text{BOOT}}$  pin may be used in the boundary scan after the configuration pin hold time ( $t_{\text{cphrs}}$ ) given in [Section 14.4.2 "Reset and Configuration"](#) is met.

The Boundary Scan Description Language (BSDL) file for the OS81212/4/6 is available from Microchip upon request.

## 13.1 Instruction Register (IR)

The JTAG Instruction Register (IR) is accessed when the TAP controller is in the *Select-IR-scan*, *Capture-IR*, *Shift-IR*, *Exit1-IR*, *Pause-IR*, *Exit2-IR*, or *Update-IR* state.

### REGISTER 13-1: IR: JTAG INSTRUCTION REGISTER

| R/W-0 | R/W-0 | R/W-0 | R/W-1 |
|-------|-------|-------|-------|
| IR[3] | IR[2] | IR[1] | IR[0] |
| 3     | 2     | 1     | 0     |

**Legend:**

R = Readable bit      W = Writable bit      P = Programmable bit      r = Reserved bit  
 U = Unimplemented bit      -n = Bit Value at POR: ('0', '1', x = Unknown)

bit 3-0      **IR<3:0>**: Instruction Code. Default value = 0001. Supported instructions include:

- 0000 = EXTEST      Enables EXTEST operation
- 0001 = SAMPLE/PRELOAD      Enables SAMPLE or PRELOAD operation
- 0010 = IDCODE      Enables shifting out of the Manufacturer ID
- 1111 = BYPASS      Enables BYPASS
- All other codes      Not supported

# OS81212/4/6

## 13.2 Identification Register (IDR)

The JTAG Identification Register (IDR) is accessed when the TAP controller is in the *Select-DR-scan*, *Capture-DR*, *Shift-DR*, *Exit1-DR*, *Pause-DR*, *Exit2-DR*, or *Update-DR* state.

### REGISTER 13-2: IDR: JTAG IDENTIFICATION REGISTER

|        |        |        |        |         |         |         |         |
|--------|--------|--------|--------|---------|---------|---------|---------|
| R      | R      | R      | R      | R-0     | R-0     | R-0     | R-0     |
| REV[0] | REV[1] | REV[2] | REV[3] | FEAT[0] | FEAT[1] | FEAT[2] | FEAT[3] |
| 31     | 30     | 29     | 28     | 27      | 26      | 25      | 24      |
| R      | R      | R      | R      | R       | R       | R       | R       |
| PID[0] | PID[1] | PID[2] | PID[3] | PID[4]  | PID[5]  | PID[6]  | PID[7]  |
| 23     | 22     | 21     | 20     | 19      | 18      | 17      | 16      |
| r-0    | r-0    | r-0    | r-0    | R-0     | R-1     | R-0     | R-0     |
| —      | —      | —      | —      | MID[0]  | MID[1]  | MID[2]  | MID[3]  |
| 15     | 14     | 13     | 12     | 11      | 10      | 9       | 8       |
| R-0    | R-1    | R-0    | R-0    | R-0     | R-1     | R-0     | R-1     |
| MID[4] | MID[5] | MID[6] | MID[7] | MID[8]  | MID[9]  | MID[10] | 1       |
| 7      | 6      | 5      | 4      | 3       | 2       | 1       | 0       |

|                       |  |                      |                  |  |  |  |  |
|-----------------------|--|----------------------|------------------|--|--|--|--|
| Legend:               |  |                      |                  |  |  |  |  |
| R = Readable bit      | W = Writable bit                               | P = Programmable bit | r = Reserved bit |  |  |  |  |
| U = Unimplemented bit | -n = Bit Value at POR: ('0', '1', x = Unknown) |                      |                  |  |  |  |  |

The 32-bit JTAG IDR register is read out least significant bit first, bit 0 to bit 31. The bits below are listed in reverse order from the way they are read out.

bit 31-28 **REV<3:0>**: Hardware Revision. Default value depends on actual hardware revision.

- 0001 = Revision A0A
- 0010 = Revision A1B
- 0100 = Revision B1A
- 0101 = Revision B2B – The data sheet is applicable to this hardware revision.

bit 27-24 **FEAT<3:0>**: Feature Identifier. Default value depends on device grade.

- 0000 = Grade A (standard)

bit 23-16 **PID<15:0>**: Part Identifier.

- 32h = OS81212
- 34h = OS81214
- 33h = OS81216

bit 15-12 **Reserved**: Read as '0000'

bit 11-1 **MID<10:0>**: 11-bit Manufacturer ID.

- 222h = Microchip Manufacturer ID

bit 0 Least significant bit (must be set per IEEE 1149.1 specification). Default value = 1.

## 14.0 ELECTRICAL CHARACTERISTICS

Specifications are subject to change without notice.

### 14.1 Absolute Maximum Ratings

**TABLE 14-1: ABSOLUTE MAXIMUM RATINGS**

| Parameter <sup>1</sup>                        | Min   | Max                     | Unit |
|---|-------|-------------------------|------|
| Storage Temperature                           | -65   | 150                     | °C   |
| Junction Temperature Under Bias               |       | 150                     | °C   |
| Power Supply Voltage: <sup>2</sup>            |       |                         |      |
| Core (VDDCn)                                  | -0.2  | 2.5                     | V    |
| Analog (VDDA18)                               | -0.2  | 2.5                     | V    |
| Peripheral (VDDPn)                            | -0.2  | 3.9                     | V    |
| Analog (VDDA33)                               | -0.2  | 3.9                     | V    |
| Analog Continuous Power (VDDAU <sub>n</sub> ) | -0.2  | 3.9                     | V    |
| bPHY Analog (VDDE33)                          | -0.2  | 3.9                     | V    |
| bPHY Analog (VDDE18)                          | -0.2  | 2.5                     | V    |
| Power Supply Ramp Rate:                       |       |                         |      |
| Analog Continuous Power (VDDAU <sub>n</sub> ) | 16.67 | 6000                    | μs/V |
| Peripheral (VDDPn)                            | 16.67 | 6000                    | μs/V |
| Power Dissipation: <sup>3</sup>               |       |                         |      |
| OS81212                                       |       | 985                     | mW   |
| OS81214                                       |       | 944                     | mW   |
| OS81216                                       |       | 944                     | mW   |
| DC Current to any Pin Except Power            |       | ±10                     | mA   |
| Maximum Input Voltage:                        |       |                         |      |
| BRXN, BRXP                                    | -0.5  | VDDAU <sub>n</sub> +0.3 | V    |
| XTI   | -0.5  | VDDA33+0.3              | V    |
| Other digital pins                            | -0.5  | VDDPn+0.3               | V    |

**Note 1:** Stresses above those listed in this table may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions, above those indicated in the operation listings of this specification, is not implied. Exposure to maximum rating conditions may affect device reliability.

**2:** The OS81212/4/6 power supplies must be sequenced in a way such that the lower voltage supplies never exceed the higher voltage supplies. Specifically, VDDCn must never be more than 0.5 V above VDDPn; and VDDA33, VDDE33, VDDA18, and VDDE18 must never be more than 0.5 V above VDDAU<sub>n</sub>. These requirements apply to power-up, power-down, and normal operation.

**3:** Maximum power dissipation is calculated using the maximum active mode power supply currents (listed in Section 14.3 “DC Characteristics (other than MediaLB Port)”) and the maximum power supply voltages under operating conditions (listed in Section 14.2 “Operating Conditions”).

# OS81212/4/6

## 14.2 Operating Conditions

TABLE 14-2: OPERATING CONDITIONS

| Parameter                                     | Symbol         | Min   | Max                | Unit |
|---|----------------|-------|--------------------|------|
| Junction Temperature                          | $T_J$          | -40   | 125                | °C   |
| Power Supply Voltage:                         |                |       |                    |      |
| Core (VDDCn)                                  |                | 1.71  | 1.89               | V    |
| Analog (VDDA18)                               |                | 1.71  | 1.89               | V    |
| Peripheral (VDDPn)                            |                | 3.135 | 3.465              | V    |
| Analog (VDDA33)                               |                | 3.135 | 3.465              | V    |
| Analog Continuous Power (VDDAU <sub>n</sub> ) |                | 3.135 | 3.465              | V    |
| bPHY Analog (VDDE33)                          |                | 3.135 | 3.465              | V    |
| bPHY Analog (VDDE18)                          |                | 1.71  | 1.89               | V    |
| Voltage applied to pins:                      |                |       |                    |      |
| BRXN, BRXP                                    |                | 0     | VDDAU <sub>n</sub> | V    |
| XTI   |                | 0     | VDDA33             | V    |
| Other digital pins                            |                | 0     | VDDPn              | V    |
| Operating Network Frame Rate <sup>1</sup>     | F <sub>s</sub> | 47.9  | 48.1               | kHz  |

**Note 1:** The targeted operating network frame rate (F<sub>s</sub>) is 48 kHz.

## 14.3 DC Characteristics (other than MediaLB Port)

$T_J = -40$  to  $125$  °C;  $V_{DDCn}, V_{DDA18}, V_{DDE18} = 1.8$  V  $\pm 5$  %;  $V_{DDPn}, V_{DDA33}, V_{DDE33}, V_{DDAU_n} = 3.3$  V  $\pm 5$  %; ePAD GND = 0.0 V; Load Capacitance = 30 pF; PLL locked at 48 kHz; unless otherwise noted.

**TABLE 14-3: DC CHARACTERISTICS (OTHER THAN MediaLB PORT)**

| Parameter   | Symbol  | Min            | Typ | Max                             | Unit                             | Comments                |
|---|---|----------------|-----|---------------------------------|----------------------------------|-------------------------|
| Low-Level Input Voltage:<br>XTI <sup>1</sup><br>All other digital pins  | $V_{IL}$  |                |     | 0.75<br>0.8                     | V<br>V                           |                         |
| High-Level Input Voltage:<br>XTI <sup>1</sup><br>All other digital pins   | $V_{IH}$  | 2.0<br>2.0     |     |                                 | V<br>V                           |                         |
| Input Leakage Current   | $I_L$   |                |     | $\pm 10$                        | $\mu$ A                          | $0 < V_{in} < V_{DDPn}$ |
| Input Hysteresis  | $\Delta V_{hyst}$   | 200            |     |                                 | mV                               |                         |
| Low-Level Output Voltage:<br>All other digital pins   | $V_{OL}$  |                |     | 0.4                             | V                                | $I_{OL} = 4$ mA         |
| High-Level Output Voltage:<br>All other digital pins  | $V_{OH}$  | $V_{DDPn}-0.4$ |     |                                 | V                                | $I_{OH} = -4$ mA        |
| Digital Input Pin Capacitance   |   |                |     | 10                              | pF                               |                         |
| <b>Active Mode Current: 2,3</b>   |   |                |     |                                 |                                  |                         |
| 1.8 V Supply Current:<br>Core ( $V_{DDCn}$ )<br>Analog ( $V_{DDA18}$ )<br>bPHY Analog ( $V_{DDE18}$ )   | $I_{DDC18}$<br>$I_{ANA18}$<br>$I_{ANAE18}$                            |                |     | 185<br>82<br>1                  | mA<br>mA<br>mA                   |                         |
| 3.3 V Supply Current:<br>Peripheral ( $V_{DDPn}$ )<br>OS81212<br>OS81214<br>OS81216<br>Analog ( $V_{DDA33}$ )<br>bPHY Analog ( $V_{DDE33}$ )<br>Analog Continuous ( $V_{DDAU_n}$ ) <sup>4</sup> | $I_{DDP33}$<br><br><br><br>$I_{ANA33}$<br>$I_{ANAE33}$<br>$I_{VDDAU}$ |                |     | 37<br>25<br>25<br>40<br>6<br>55 | mA<br>mA<br>mA<br>mA<br>mA<br>mA |                         |
| <b>Sleep Mode Current: 5</b>  |   |                |     |                                 |                                  |                         |
| 3.3 V Supply Current:<br>Analog Continuous ( $V_{DDAU_n}$ )   | $I_{VDDAU-SL}$  |                |     | 65                              | $\mu$ A                          |                         |

**Note 1:** Applicable only when an external clock driver is used.

- 2:** The supply current specified for  $V_{DDCn}$ ,  $V_{DDPn}$ , and  $V_{DDE33}$  is the sum total of all power pins for the power supply.
- 3:** The maximum current is calculated by loading each pin to the corresponding maximum loading ( $I_{OL}$ ) and each INIC port is enabled at the following clock speed (MediaLB  $MLBCLK = 1024 \times F_s$ , Streaming Port  $SCKA/SCKB = 512 \times F_s$  with all data pins configured as outputs, SPI  $SCLK = 25$  MHz, I<sup>2</sup>C  $SCL = 400$  kHz,  $RMCK = 512 \times F_s$ ).
- 4:** During a bulk current injection (BCI) event with a common mode swing of  $0.9 V_{PK-PK}$  on the **BRXP** and **BRXN** pins,  $I_{VDDAU}$  can increase by 6.4 mA (RMS).
- 5:** Sleep Mode is when  $V_{DDAU_n} = 3.3$  V and all other power supplies are at 0 V. In this mode, the bPHY physical interface is suspended in a low power state and remains idle until valid network activity is detected.

# OS81212/4/6

## 14.4 AC Characteristics (other than MediaLB Port)

### 14.4.1 GENERAL SIGNAL AND CLOCKS

$T_J = -40$  to  $125$  °C;  $VDDCn, VDDA18 = 1.8\text{ V} \pm 5\%$ ;  $VDDPn, VDDA33, VDDAUn = 3.3\text{ V} \pm 5\%$ ; ePAD GND = 0.0 V; Load Capacitance = 30 pF; PLL locked at 48 kHz; unless otherwise noted.

**TABLE 14-4: AC CHARACTERISTICS (OTHER THAN MediaLB PORT)**

| Parameter   | Symbol           | Min    | Typ              | Max                                 | Unit                                   | Comments                 |
|---|------------------|--------|------------------|-------------------------------------|--|--------------------------|
| Output rise time:<br>ERR, TDO<br>RMCK, SRXAn, SRXBn, FSYn, SCKn, MUTE<br>SDOUT, SINT<br>BSTATUS<br>GPn  | $t_{hdr}$        |        |                  | 12<br>5<br>5<br>12<br>5             | ns<br>ns<br>ns<br>ns<br>ns             | ( $V_{OL}$ to $V_{OH}$ ) |
| Output fall time:<br>ERR, PWROFF<br>SDA, SCL, INT<br>DSDA, DSCL, DINT, TDO<br>RMCK, SRXAn, SRXBn, FSYn, SCKn, MUTE<br>SDOUT, SINT<br>BSTATUS<br>GPn | $t_{hdf}$        |        |                  | 12<br>12<br>12<br>5<br>5<br>12<br>5 | ns<br>ns<br>ns<br>ns<br>ns<br>ns<br>ns | ( $V_{OH}$ to $V_{OL}$ ) |
| GPIO Debounce Time Delay  | $t_{dbdelay}$    | 1      |                  | 65,535                              | ms                                     |                          |
| GPIO Debounce Time Resolution   | $t_{dbres}$      |        | 1                |                                     | ms                                     |                          |
| Minimum Pulse Width (Normal Input)  | $t_{minpwnml}$   | 200    |                  |                                     | $\mu\text{s}$                          |                          |
| Minimum Pulse Width (Sticky Input)  | $t_{minpwstky}$  | 60     |                  |                                     | ns                                     |                          |
| Maximum MUTE pin delay from event <sup>1</sup>  | $t_{muteon}$     |        |                  | 2                                   | ms                                     |                          |
| <b>Clocks:</b>  |                  |        |                  |                                     |  |                          |
| RMCK Recovered Master Clock Output  | $f_{rmck}$       | 18,000 |                  | $1536 \times F_s$                   | Hz                                     | when enabled             |
| External clock/crystal frequency <sup>2</sup>   | $f_{osc}$        |        | $384 \times F_s$ |                                     | Hz                                     |                          |
| External clock/crystal deviation <sup>2</sup>   | $\Delta f_{osc}$ |        |                  | $\pm 200$                           | ppm                                    |                          |

**Note 1:** The  $t_{muteon}$  parameter is the maximum time from an event occurring that can corrupt data to the MUTE pin being asserted high.

**2:** For a 50 Mbit/s network, the external clock/crystal must meet the physical layer specifications for Time Base Deviation  $\Delta F_s$  and network bit rate tolerance within  $\pm 200$  ppm over all conditions.



## 14.4.2 RESET AND CONFIGURATION

$T_J = -40$  to  $125$  °C;  $V_{DDCn}, V_{DDA18} = 1.8$  V  $\pm 5$  %;  $V_{DDPn}, V_{DDA33}, V_{DDAUn} = 3.3$  V  $\pm 5$  %; ePAD GND = 0.0 V; Load Capacitance = 30 pF; PLL locked at 48 kHz; unless otherwise noted.

**TABLE 14-5: RESET AND CONFIGURATION AC CHARACTERISTICS**

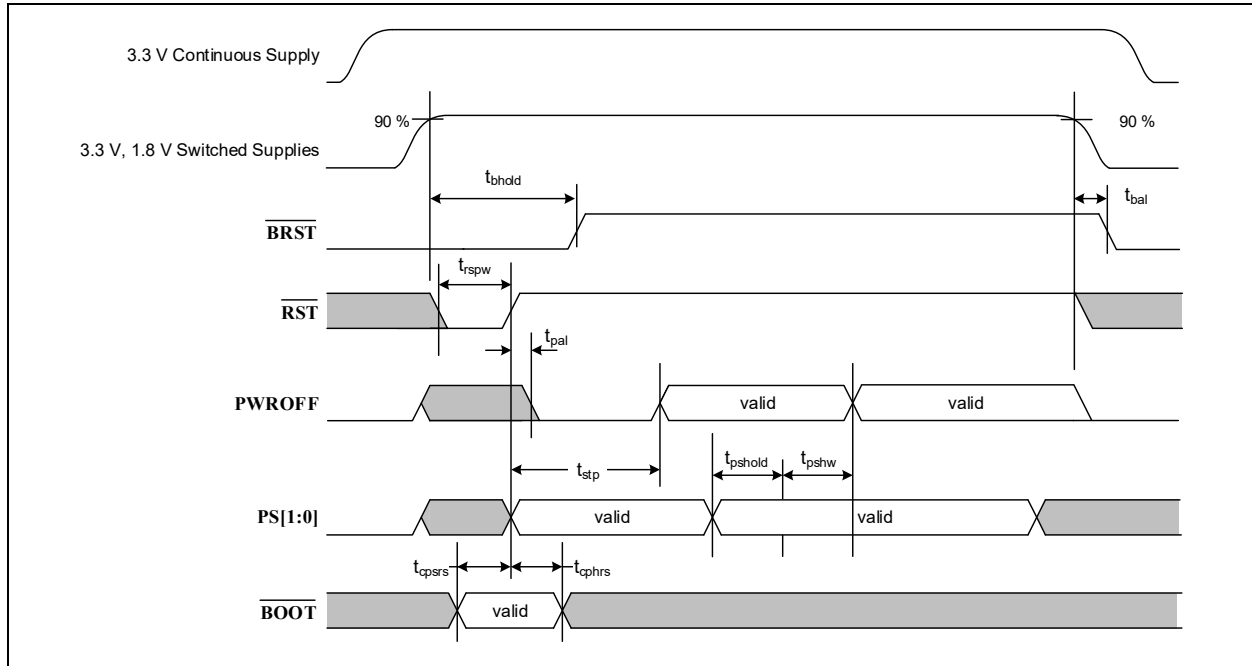
| Parameter   | Symbol       | Min | Typ | Max | Unit    | Comments |
|---|--------------|-----|-----|-----|---------|----------|
| PS[1:0] hold time for event detection <sup>1,2,3</sup>    | $t_{pshold}$ | 2.2 |     |     | ms      |          |
| PS[1:0] event detection to hardware response <sup>2</sup> | $t_{pshw}$   |     | 2   | 30  | ms      |          |
| RST rising to PS[1:0] state recognition <sup>3</sup>      | $t_{stp}$    |     | 20  | 100 | ms      |          |
| RST pulse width <sup>4,6</sup>                            | $t_{rspw}$   | 150 |     |     | ns      |          |
| BOOT setup to RST rising <sup>4,6</sup>                   | $t_{cpsrs}$  | 0   |     |     | ns      |          |
| BOOT hold time from RST rising <sup>4,6</sup>             | $t_{cphrs}$  | 2.5 |     |     | ms      |          |
| RST rising to PWROFF asserted low                         | $t_{pal}$    |     | 10  | 100 | $\mu$ s |          |
| BRST hold time after $\geq 90$ % VDD <sup>5</sup>         | $t_{bhold}$  | 1   |     |     | ms      |          |
| BRST asserted from $\leq 90$ % VDD <sup>5</sup>           | $t_{bal}$    | 0   |     | 100 | $\mu$ s |          |

**Note 1:**  $t_{pshold}$  is the minimum time required for the INIC hardware to detect an external power management state change at the PS[1:0] pins. Events that do not meet this qualification time will be ignored.

- 2:** Once a power state change is detected and qualified (see  $t_{pshold}$ ), any hardware responses (e.g. disabling the network outputs and releasing the PWROFF pin when  $U_{LOW}$  is recognized) occur within  $t_{pshw}$ . Subsequent power state changes may override pending hardware responses. Therefore, to guarantee that a particular hardware response occurs,  $t_{pshold}$  should be at least  $t_{pshw(max)}$ . Refer to [Chapter 9.0 "External Power Management"](#) for more information. The *INIC Software Stack* may take a longer time to report an external power management event to the EHC.
- 3:** Immediately following power-up (or reset), INIC detects and responds to PS[1:0] within  $t_{stp}$ . Therefore  $t_{pshold}$  be at least  $t_{stp(max)}$  after power-up (or reset) to guarantee the initial power state detected.
- 4:** After the 3.3 V and 1.8 V power supplies have stabilized, INIC must be reset for normal operation. A proper reset is when all supplies are  $\geq 90$  % VDD.
- 5:** In order to prevent glitches in the network transmit signals during startup and shutdown, the BRST pin must be held low for a minimum hold time of  $t_{bhold}$  once all supplies are  $\geq 90$  % VDD, and must be asserted low within  $t_{bal}$  after all supplies are  $\leq 90$  % VDD.
- 6:** The RST pulse width indicates the minimum time required to reset the part; however, the BOOT pin can take longer to settle to its default state, based on the trace capacitance and size of the on-board pull-up or pull-down resistor. Therefore, the RST pulse width must be long enough to allow the external BOOT pin to achieve its default state.

# OS81212/4/6

FIGURE 14-1: POWER-UP AND RESET TIMING



## 14.4.3 BALANCED MEDIA PHYSICAL LAYER

$T_J = -40$  to  $125$  °C;  $V_{DDCn}, V_{DDA18}, V_{DDE18} = 1.8$  V  $\pm 5$  %;  $V_{DDPn}, V_{DDA33}, V_{DDE33}, V_{DDAU} = 3.3$  V  $\pm 5$  %; ePAD GND = 0.0 V; Load Capacitance = 30 pF; PLL locked at 48 kHz; unless otherwise noted.

**TABLE 14-6: ELECTRICAL PHYSICAL LAYER AC CHARACTERISTICS**

| Parameter  | Symbol            | Min | Typ | Max | Unit    | Comment  |
|--|-------------------|-----|-----|-----|---------|--|
| $\overline{BRST}$ rising to <i>bPHY Tx</i> output enabled                                    | $t_{bphyon}$      |     |     | 100 | $\mu$ s |  |
| $\overline{BRST}$ rising to <i>bPHY Tx</i> output disabled                                   | $t_{bphyoff}$     |     |     | 2   | $\mu$ s |  |
| $\overline{RST}$ rising to <i>bPHY Tx</i> output active <sup>1</sup>                         | $t_{inicinit}$    | 50  |     | 60  | ms      |  |
| Activity on <i>bPHY Rx</i> to <i>bPHY Tx</i> output enabled <sup>2</sup>                     | $t_{NtwStartup}$  |     |     | 6   | ms      | $\overline{RST} = 3.3$ V,<br>$\overline{BRST} = 3.3$ V |
| Loss of activity on <i>bPHY Rx</i> to <i>bPHY Tx</i> output disabled <sup>3</sup>            | $t_{NtwShutdown}$ |     |     | 15  | ms      | $\overline{RST} = 3.3$ V,<br>$\overline{BRST} = 3.3$ V |
| Activity on <i>bPHY Rx</i> to <i>BSTATUS</i> falling   | $t_{staf}$        | 50  |     | 700 | $\mu$ s |  |
| Loss of activity on <i>bPHY Rx</i> to <i>BSTATUS</i> rising - normal shutdown                | $t_{statr}$       |     |     | 100 | $\mu$ s |  |
| Loss of activity on <i>bPHY Rx</i> to <i>BSTATUS</i> rising - Sudden Signal Off <sup>4</sup> | $t_{stattrso}$    |     | 180 |     | ms      |  |

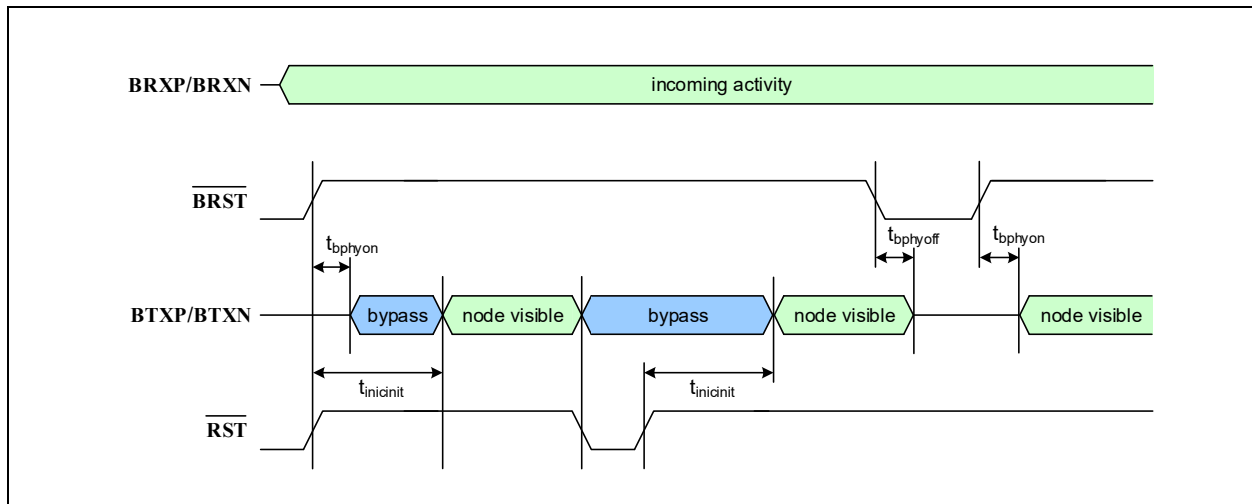
**Note 1:** When INIC's *BTXP/BTXN* output is enabled, the INIC transceiver becomes *visible* and INIC enters the network (e.g. obtains a node address).

**2:** The  $t_{NtwStartup}$  parameter is only valid if  $t_{inicinit}$  has elapsed, otherwise the time from activity on *bPHY Rx* until *bPHY Tx* enabled is determined by  $t_{inicinit}$ .  $t_{NtwStartup}$  also assumes activity has been absent for at least the minimum  $t_{Restart}$  time. The  $t_{NtwStartup}$  parameter represents the INIC component of the [ISO 21806 \(Parts 1-7\): 2020 – Road Vehicles – Media Oriented Systems Transport \(MOST\) Specifications \[1\]](#) and [ISO 21806 \(Parts 14-15\): 2021 – Road Vehicles – Media Oriented Systems Transport \(MOST\) Lean Application Layer Specification \[2\]](#)  $t_{WakeUp}$  parameter.

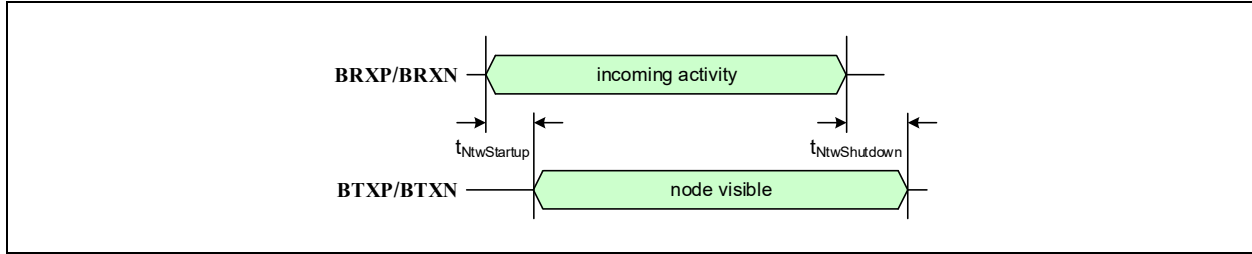
**3:** The  $t_{NtwShutdown}$  parameter assumes  $t_{inicinit}$  has previously expired.  $t_{NtwShutdown}$  represents the INIC component of the [ISO 21806 \(Parts 1-7\): 2020 – Road Vehicles – Media Oriented Systems Transport \(MOST\) Specifications \[1\]](#) and [ISO 21806 \(Parts 14-15\): 2021 – Road Vehicles – Media Oriented Systems Transport \(MOST\) Lean Application Layer Specification \[2\]](#)  $t_{ShutDown}$  parameter.

**4:** During an unexpected loss of activity on the *bPHY Rx* (such as a sudden-signal-off (SSO) event), the *BSTATUS* reaction time will increase.

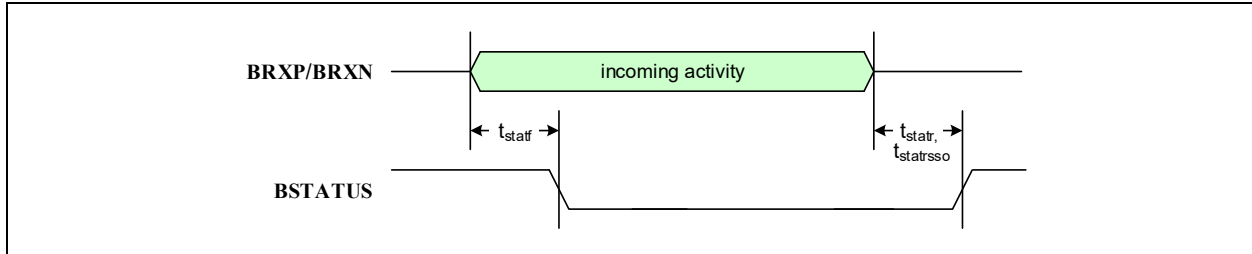
**FIGURE 14-2: BALANCED MEDIA PHYSICAL LAYER RESET TIMING**



**FIGURE 14-3: BALANCED MEDIA PHYSICAL LAYER STARTUP AND SHUTDOWN TIMING**



**FIGURE 14-4: BALANCED MEDIA PHYSICAL LAYER STATUS TIMING**



## 14.5 Network Characteristics

### 14.5.1 BALANCED MEDIA PHYSICAL LAYER

$T_J = -40$  to  $125$  °C;  $V_{DDCn}, V_{DDA18}, V_{DDE18} = 1.8$  V  $\pm 5$  %;  $V_{DDPn}, V_{DDA33}, V_{DDE33}, V_{DDAU_n} = 3.3$  V  $\pm 5$  %; ePAD GND = 0.0 V; Load Capacitance = 30 pF; PLL locked at 48 kHz; unless otherwise noted.

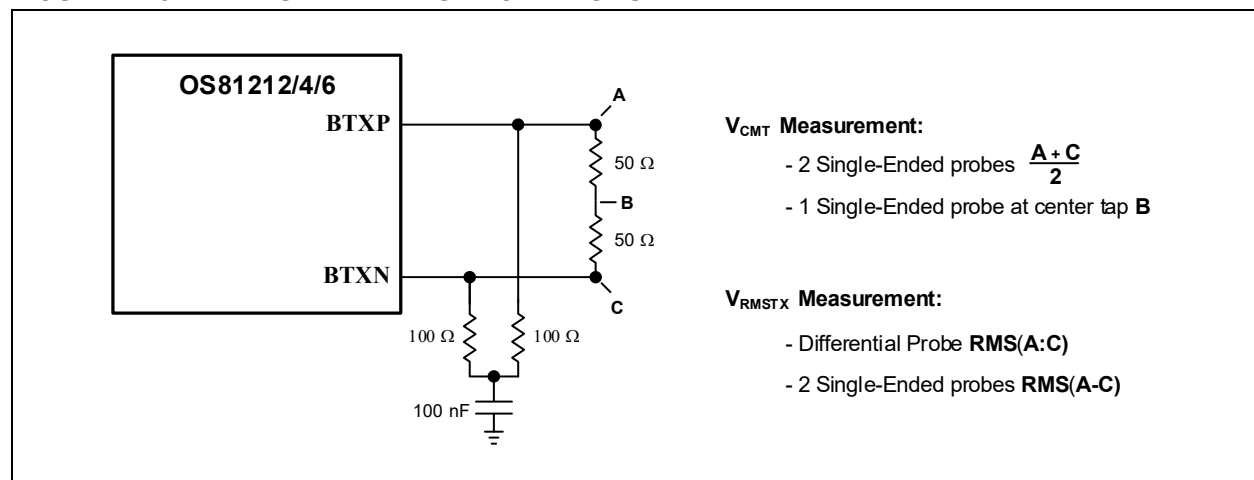
**TABLE 14-7: ELECTRICAL PHYSICAL LAYER ELECTRICAL CHARACTERISTICS**

| Parameter                         | Symbol             | Min | Typ  | Max | Unit     | Test Conditions |
|-----------------------------------|--------------------|-----|------|-----|----------|-----------------|
| <b>Transmitter:</b>               |                    |     |      |     |          |                 |
| BTX Common-Mode output voltage    | $V_{CMT}$          |     | 1.65 |     | V        | See Figure 14-5 |
| Rise time, fall time <sup>1</sup> | $t_{rtx}, t_{ftx}$ | 4.6 |      | 6.9 | ns       |                 |
| Transferred jitter <sup>2</sup>   | $t_{jtx}$          |     |      | 50  | ps (RMS) |                 |
| RMS signal amplitude              | $V_{RMSTX}$        | 533 |      | 921 | mV (RMS) | See Figure 14-5 |
| Eye-mask <sup>3</sup>             | See Figure 14-6    |     |      |     |          |                 |
| <b>Receiver</b>                   |                    |     |      |     |          |                 |
| BRX Common-Mode input voltage     | $V_{CMR}$          |     | 1.65 |     | V        |                 |
| Input RMS amplitude for ON State  | $V_{RMSRXON}$      | 110 |      |     | mV (RMS) | See Note 4      |
| Input RMS amplitude for OFF State | $V_{RMSRXOFF}$     |     |      | 35  | mV (RMS) | See Note 5      |
| Eye-mask <sup>3</sup>             | See Figure 14-7    |     |      |     |          |                 |

**Note 1:** Rise and fall time parameters are based upon 10% - 90% signal amplitude limits.

- 2:** Using the jitter filter specified in ISO 21806 (Parts 12-13): 2021 – Road Vehicles – Media Oriented Systems Transport (MOST) 50 Mbit/s Balanced Media Physical Layer Specification [12].
- 3:** Using Golden PLL specified in ISO 21806 (Parts 12-13): 2021 – Road Vehicles – Media Oriented Systems Transport (MOST) 50 Mbit/s Balanced Media Physical Layer Specification [12].
- 4:** Using the  $V_{RMSRXON}$  frequency range (12.288 MHz - 24.576 MHz) specified in ISO 21806 (Parts 12-13): 2021 – Road Vehicles – Media Oriented Systems Transport (MOST) 50 Mbit/s Balanced Media Physical Layer Specification [12].
- 5:** Using the  $V_{RMSRXOFF}$  frequency range (0 kHz - 10 kHz) specified in ISO 21806 (Parts 12-13): 2021 – Road Vehicles – Media Oriented Systems Transport (MOST) 50 Mbit/s Balanced Media Physical Layer Specification [12].

**FIGURE 14-5: TRANSMITTER TEST CONDITIONS**

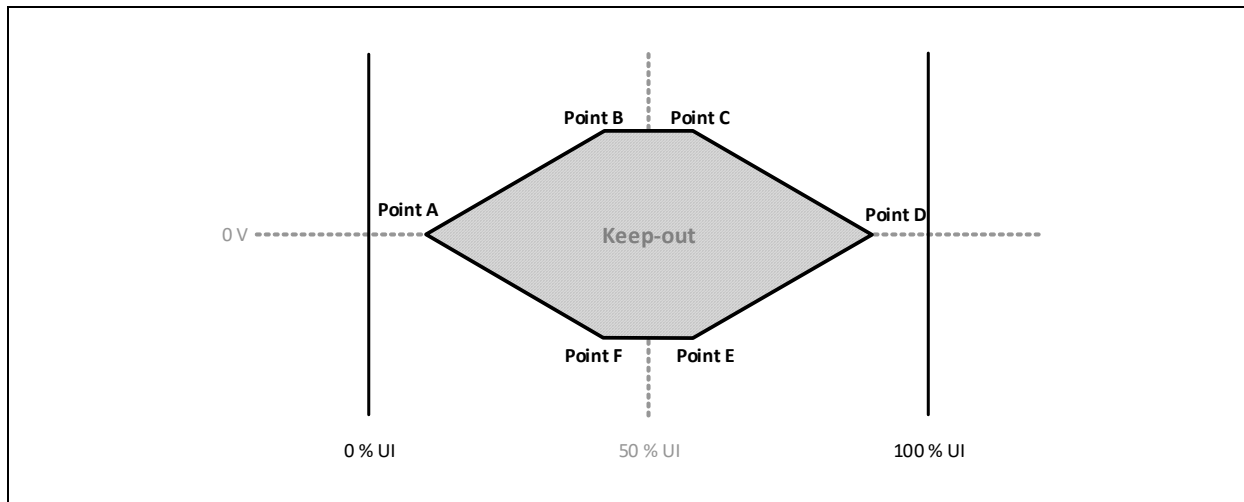


# OS81212/4/6

## 14.5.1.1 BTXP/BTXN - Transmit Eye Mask Pattern

The eye mask pattern in [Figure 14-6](#) defines the minimum transmit eye opening area for BTXP/BTXN. The OS81212/4/6 BTXP/BTXN differential signal that is transmitted does not intrude on any of the keep-out areas defined for this eye mask.

**FIGURE 14-6: BTXP/BTXN TRANSMIT EYE MASK PATTERN**



[Table 14-8](#) defines the boundaries of the BTXP/BTXN eye mask shown in [Figure 14-6](#).

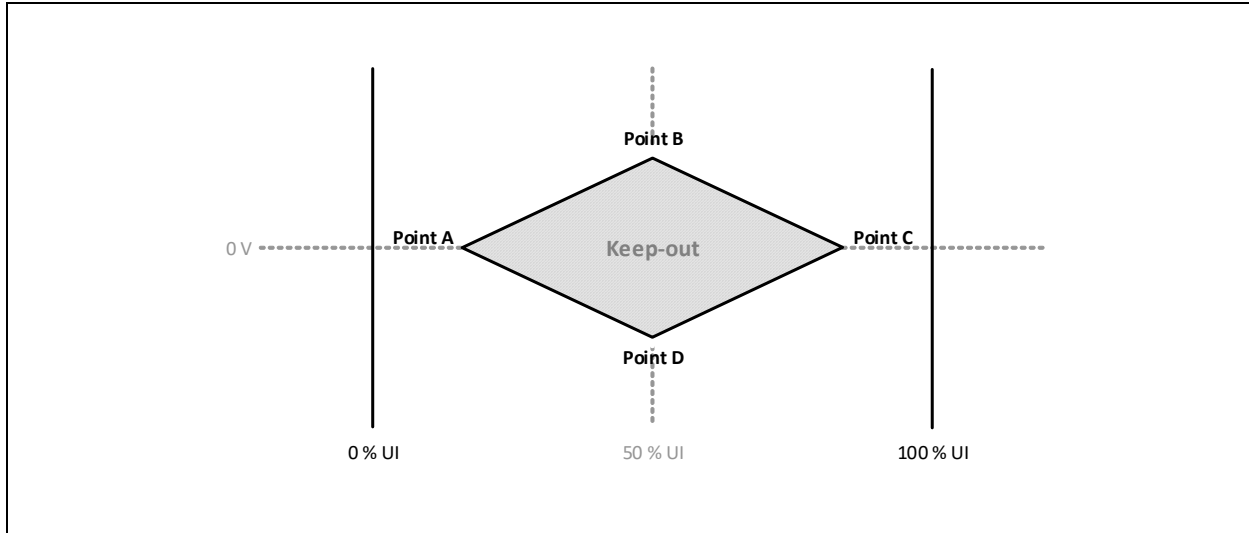
**TABLE 14-8: BTXP/BTXN TRANSMIT EYE MASK BOUNDARIES**

| Parameter | Voltage | Time      |
|-----------|---------|-----------|
| Point A   | 0.00V   | 7.5 % UI  |
| Point B   | +0.50V  | 45.0 % UI |
| Point C   | +0.50V  | 55.0 % UI |
| Point D   | 0.00V   | 92.5 % UI |
| Point E   | -0.50V  | 55.0 % UI |
| Point F   | -0.50V  | 45.0 % UI |

## 14.5.1.2 BRXP/BRXN - Receive Eye Mask Pattern

The OS81212/4/6 BRXP/BRXN receiver will tolerate an input signal that runs up to the keep-out boundaries of the eye mask pattern in [Figure 14-7](#). A differential signal received at BRXP/BRXN that fills the eye mask (up to the keep-out boundary) represents the worst-case signal that is tolerable by the OS81212/4/6 receiver.

**FIGURE 14-7: BRXP/BRXN RECEIVE EYE MASK PATTERN**



[Table 14-9](#) defines the boundaries of the BRXP/BRXN eye mask shown in [Figure 14-7](#).

**TABLE 14-9: BRXP/BRXN RECEIVE EYE MASK BOUNDARIES**

| Parameter | Voltage | Time      |
|-----------|---------|-----------|
| Point A   | 0.00V   | 30.0 % UI |
| Point B   | +0.050V | 50.0 % UI |
| Point C   | 0.00V   | 70.0 % UI |
| Point D   | -0.050V | 50 % UI   |

# OS81212/4/6

## 14.6 MediaLB Port (3-pin)

**Note:** Refer to the [MediaLB Specification \[5\]](#) for full system-level specifications defining the interaction between the MediaLB Controller (i.e. OS81212) and attached MediaLB Devices.

$T_J = -40$  to  $125$  °C;  $V_{DDCn}, V_{DDA18} = 1.8$  V  $\pm 5$  %;  $V_{DDPn}, V_{DDA33}, V_{DDAUn} = 3.3$  V  $\pm 5$  %; ePAD GND = 0.0 V; Load Capacitance = 60 pF; PLL locked at 48 kHz; unless otherwise noted.

**TABLE 14-10: MediaLB PORT (3-PIN 256×FS OR 512×FS)**

| Parameter  | Symbol       | Min        | Typ    | Max                | Unit       | Comments                       |
|--|--------------|------------|--------|--------------------|------------|--------------------------------|
| Low-level input voltage  | $V_{IL}$     |            |        | 0.7                | V          |                                |
| High-level input voltage   | $V_{IH}$     | 1.8        |        |                    | V          |                                |
| Input leakage current  | $I_L$        |            |        | $\pm 1$            | $\mu$ A    | $0 < V_{in} < V_{DDPn}$        |
| Low-level output voltage   | $V_{OL}$     |            |        | 0.4                | V          | $I_{OL} = 8$ mA                |
| High-level output voltage  | $V_{OH}$     | 2.4        |        |                    | V          | $I_{OH} = -8$ mA               |
| MLBCLK operating frequency <sup>1,2</sup>                        | $f_{mck}$    | 12.2624    | 12.288 | 12.3136<br>12.4672 | MHz<br>MHz | 256×Fs<br>256×Fs, PLL unlocked |
|  |              | 24.5248    | 24.576 | 24.6272<br>24.9344 | MHz<br>MHz | 512×Fs<br>512×Fs, PLL unlocked |
| MLBCLK rise time   | $t_{mckr}$   |            |        | 3                  | ns         | $V_{IL}$ to $V_{IH}$           |
| MLBCLK fall time   | $t_{mckf}$   |            |        | 3                  | ns         | $V_{IH}$ to $V_{IL}$           |
| MLBCLK cycle time  | $t_{mckc}$   |            | 81     |                    | ns         | 256×Fs                         |
|  |              |            | 40     |                    | ns         | 512×Fs                         |
| MLBCLK low time  | $t_{mckl}$   | 30         | 35.5   |                    | ns         | 256×Fs                         |
|  |              | 14         | 16.5   |                    | ns         | 512×Fs                         |
| MLBCLK high time   | $t_{mckh}$   | 30         | 36.5   |                    | ns         | 256×Fs                         |
|  |              | 14         | 16.5   |                    | ns         | 512×Fs                         |
| MLBSIG/MLBDAT receiver input valid to MLBCLK falling             | $t_{ds mcf}$ | 1          |        |                    | ns         |                                |
| MLBSIG/MLBDAT receiver input hold from MLBCLK low                | $t_{dh mcf}$ | $t_{mdzh}$ |        |                    | ns         |                                |
| MLBSIG/MLBDAT output high impedance from MLBCLK low <sup>3</sup> | $t_{mcf dz}$ | 0          |        | $t_{mckl}$         | ns         |                                |
| Bus hold from MLBCLK low <sup>3</sup>                            | $t_{mdzh}$   | 4          |        |                    | ns         |                                |
| MLBSIG/MLBDAT output valid from MLBCLK high                      | $t_{mcr dv}$ |            |        | 8                  | ns         |                                |

**Note 1:** The OS81212 can shut off MLBCLK to place MediaLB in a low-power state.

**2:** When the PLL is locked, MLBCLK frequencies are based on the operating network frame rate (Fs) specification given in [Section 14.2 "Operating Conditions"](#). During PLL unlock, Fs may drift up to 48.7 kHz.

**3:** The MediaLB driver can release MLBSIG/MLBDAT (e.g. high impedance) as soon as MLBCLK is low; however, the logic state of the final driven bit on the line must remain on the bus for  $t_{mdzh}$ . Therefore, coupling must be minimized while meeting the maximum capacitive load listed.



$T_J = -40$  to  $125$  °C;  $V_{DDCn}, V_{DDA18} = 1.8$  V  $\pm 5$  %;  $V_{DDPn}, V_{DDA33}, V_{DDAUn} = 3.3$  V  $\pm 5$  %; ePAD GND = 0.0 V; Load Capacitance = 40 pF; PLL locked at 48 kHz; unless otherwise noted.

**TABLE 14-11: MediaLB PORT (3-PIN 1024xFS)**

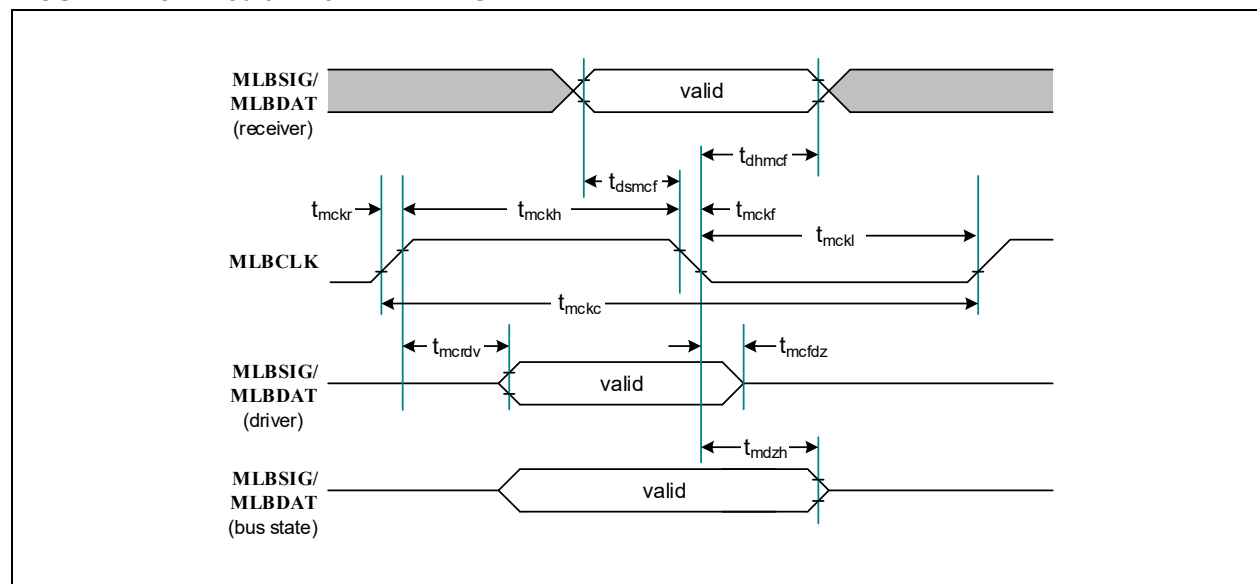
| Parameter  | Symbol      | Min        | Typ    | Max                | Unit       | Comments                         |
|--|-------------|------------|--------|--------------------|------------|----------------------------------|
| Low-level input voltage  | $V_{IL}$    |            |        | 0.7                | V          |                                  |
| High-level input voltage   | $V_{IH}$    | 1.8        |        |                    | V          |                                  |
| Input leakage current  | $I_L$       |            |        | $\pm 1$            | $\mu$ A    | $0 < V_{in} < V_{DDPn}$          |
| Low-level output voltage   | $V_{OL}$    |            |        | 0.4                | V          | $I_{OL} = 8$ mA                  |
| High-level output voltage  | $V_{OH}$    | 2.4        |        |                    | V          | $I_{OH} = -8$ mA                 |
| MLBCLK operating frequency <sup>1,2</sup>                        | $f_{mck}$   | 49.0496    | 49.152 | 49.2544<br>49.8688 | MHz<br>MHz | 1024xFs<br>1024xFs, PLL unlocked |
| MLBCLK rise time   | $t_{mckr}$  |            |        | 1                  | ns         | $V_{IL}$ to $V_{IH}$             |
| MLBCLK fall time   | $t_{mckf}$  |            |        | 1                  | ns         | $V_{IH}$ to $V_{IL}$             |
| MLBCLK cycle time  | $t_{mckc}$  |            | 20.3   |                    | ns         |                                  |
| MLBCLK low time  | $t_{mckl}$  | 6.1        | 7.3    |                    | ns         |                                  |
| MLBCLK high time   | $t_{mckh}$  | 9.3        | 10.2   |                    | ns         |                                  |
| MLBSIG/MLBDAT receiver input valid to MLBCLK falling             | $t_{dsmcf}$ | 1          |        |                    | ns         |                                  |
| MLBSIG/MLBDAT receiver input hold from MLBCLK low                | $t_{dhmcf}$ | $t_{mdzh}$ |        |                    | ns         |                                  |
| MLBSIG/MLBDAT output high impedance from MLBCLK low <sup>3</sup> | $t_{mcfdz}$ | 0          |        | $t_{mckl}$         | ns         |                                  |
| Bus hold from MLBCLK low <sup>3</sup>                            | $t_{mdzh}$  | 2          |        |                    | ns         |                                  |
| MLBSIG/MLBDAT output valid from MLBCLK high                      | $t_{mcrdv}$ |            |        | 7                  | ns         |                                  |

**Note 1:** The OS81212 can shut off MLBCLK to place MediaLB in a low-power state.

**2:** When the PLL is locked, MLBCLK frequencies are based on the operating network frame rate (Fs) specification given in Section 14.2 “Operating Conditions”. During PLL unlock, Fs may drift up to 48.7 kHz.

**3:** The MediaLB driver can release MLBSIG/MLBDAT (e.g. high impedance) as soon as MLBCLK is low; however, the logic state of the final driven bit on the line must remain on the bus for  $t_{mdzh}$ . Therefore, coupling must be minimized while meeting the maximum capacitive load listed.

**FIGURE 14-8: MediaLB 3-PIN TIMING**



# OS81212/4/6

## 14.7 Streaming Port

$T_J = -40$  to  $125$  °C;  $V_{DDCn}, V_{DDA18} = 1.8$  V  $\pm 5$  %;  $V_{DDPn}, V_{DDA33}, V_{DDAU_n} = 3.3$  V  $\pm 5$  %; ePAD GND = 0.0 V; Load Capacitance = 30 pF ( $64 \times F_s$  to  $256 \times F_s$ ); Load Capacitance = 20 pF ( $512 \times F_s$ ); PLL locked at 48 kHz; unless otherwise noted.

**TABLE 14-12: STREAMING PORT**

| Parameter   | Symbol     | Min             | Typ | Max              | Unit | Comments         |
|---|------------|-----------------|-----|------------------|------|------------------|
| FSYA/FSYB frequency <sup>1</sup>                        | $f_{fsy}$  | 47.9            | 48  | 48.1             | kHz  |                  |
| SCKA/SCKB frequency <sup>1</sup>                        | $f_{sck}$  | $64 \times F_s$ |     | $512 \times F_s$ | Hz   |                  |
| SRXAn/SRXBn valid to SCKA/SCKB rising <sup>2,4</sup>    | $t_{srs}$  | 2               |     |                  | ns   |                  |
| SRXAn/SRXBn hold from SCKA/SCKB rising <sup>2,4</sup>   | $t_{srh}$  | 5               |     |                  | ns   |                  |
| <b>SCK and FSY Outputs (Master Mode):</b>               |            |                 |     |                  |      |                  |
| SCKA/SCKB low time                                      | $t_{sckl}$ | 50              |     |                  | ns   | $64 \times F_s$  |
|   |            | 50              |     |                  | ns   | $128 \times F_s$ |
|   |            | 35              |     |                  | ns   | $256 \times F_s$ |
|   |            | 18.5            |     |                  | ns   | $512 \times F_s$ |
| SCKA/SCKB high time                                     | $t_{sckh}$ | 50              |     |                  | ns   | $64 \times F_s$  |
|   |            | 50              |     |                  | ns   | $128 \times F_s$ |
|   |            | 35              |     |                  | ns   | $256 \times F_s$ |
|   |            | 17.5            |     |                  | ns   | $512 \times F_s$ |
| SCKA/SCKB falling to FSYA/FSYB valid <sup>2,3</sup>     | $t_{fsyv}$ | -5              |     | 5                | ns   |                  |
| SRXAn/SRXBn valid from SCKA/SCKB falling <sup>2,4</sup> | $t_{sxv}$  | -2              |     | 9                | ns   |                  |
| <b>SCK and FSY Inputs (Slave Mode):</b>                 |            |                 |     |                  |      |                  |
| SCKA/SCKB low time                                      | $t_{sckl}$ | 15              |     |                  | ns   |                  |
| SCKA/SCKB high time                                     | $t_{sckh}$ | 15              |     |                  | ns   |                  |
| FSYA/FSYB valid to SCKA/SCKB rising <sup>2,3</sup>      | $t_{fsys}$ | 5               |     |                  | ns   |                  |
| FSYA/FSYB hold from SCKA/SCKB rising <sup>2,3</sup>     | $t_{fsyh}$ | 5               |     |                  | ns   |                  |
| SRXAn/SRXBn valid from SCKA/SCKB falling <sup>2,4</sup> | $t_{sxv}$  | 4               |     | 12.5             | ns   |                  |

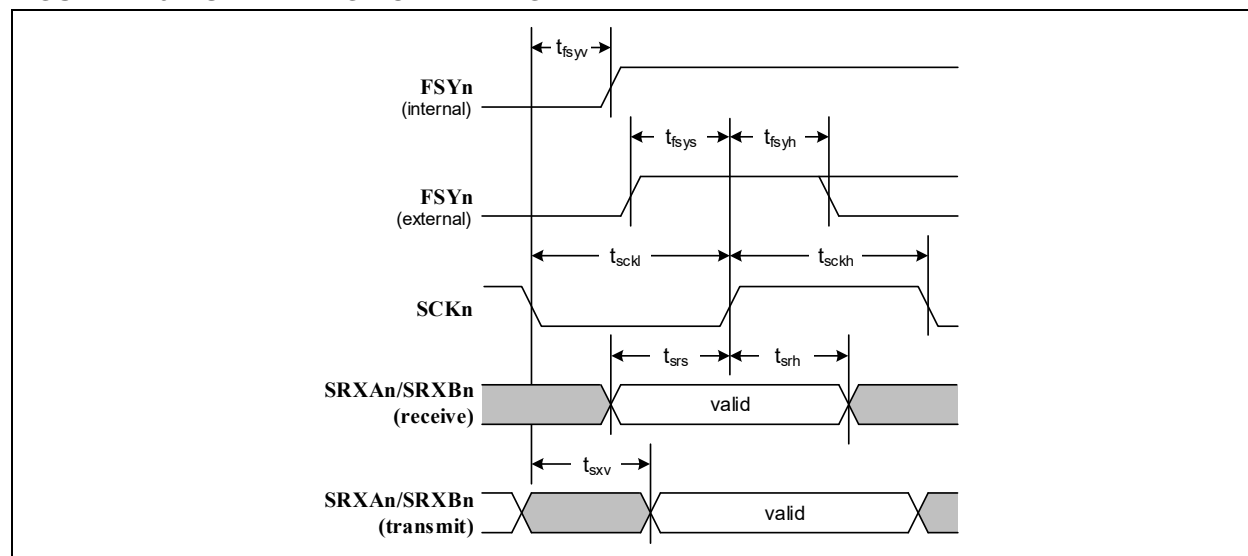
**Note 1:** If SCKA/SCKB and FSYA/FSYB are inputs, they must be frequency locked to the master clock (RMCK output clock).

**2:** The SCKA/SCKB edge at which data is stable and not changing is the rising edge for all supported formats.

**3:** FSYA/FSYB polarity depends on the data format selected when the port is opened.

**4:** The MSB of SRXAn/SRXBn is the first or the second bit after FSYA/FSYB changes, based on the format selected.

**FIGURE 14-9: STREAMING PORT TIMING**



## 14.8 SPI Port

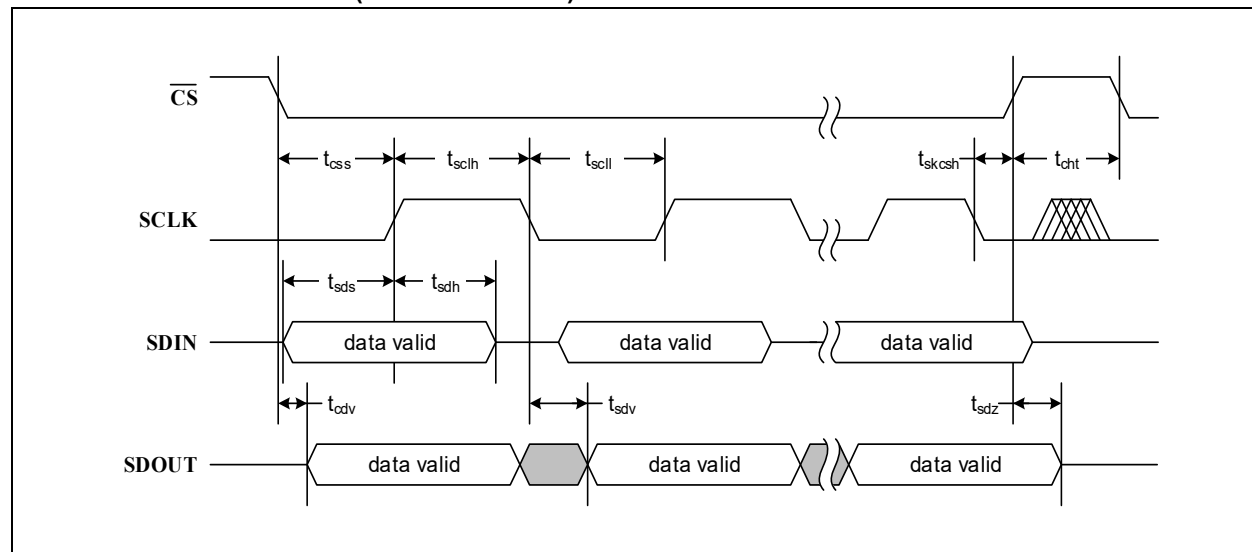
### 14.8.1 CLOCK MODE 0

$T_J = -40$  to  $125$  °C;  $V_{DDCn}, V_{DDA18} = 1.8$  V  $\pm 5$  %;  $V_{DDPn}, V_{DDA33}, V_{DDAUn} = 3.3$  V  $\pm 5$  %; ePAD GND = 0.0 V; Load Capacitance = 30 pF; PLL locked at 48 kHz; unless otherwise noted.

**TABLE 14-13: SPI PORT (CLOCK MODE 0)**

| Parameter                          | SPI Slave   |     |     |         |
|------------------------------------|-------------|-----|-----|---------|
|                                    | Symbol      | Min | Max | Unit    |
| SCLK frequency                     | $f_{scl}$   |     | 25  | MHz     |
| SCLK low time                      | $t_{scll}$  | 15  |     | ns      |
| SCLK high time                     | $t_{sclh}$  | 15  |     | ns      |
| SCLK low to $\overline{CS}$ high   | $t_{skcsh}$ | 5   |     | ns      |
| $\overline{CS}$ low to SCLK rising | $t_{css}$   | 8   |     | ns      |
| $\overline{CS}$ low to SDOUT valid | $t_{cdv}$   |     | 12  | ns      |
| $\overline{CS}$ high time          | $t_{cht}$   | 1   |     | $\mu$ s |
| SDIN valid to SCLK rising          | $t_{sds}$   | 2   |     | ns      |
| SDIN hold from SCLK rising         | $t_{sdh}$   | 5   |     | ns      |
| SCLK falling to SDOUT valid        | $t_{sdv}$   |     | 11  | ns      |
| $\overline{CS}$ high to SDOUT Hi-Z | $t_{sdz}$   |     | 8   | ns      |

**FIGURE 14-10: SPI TIMING (CLOCK MODE 0)**



# OS81212/4/6

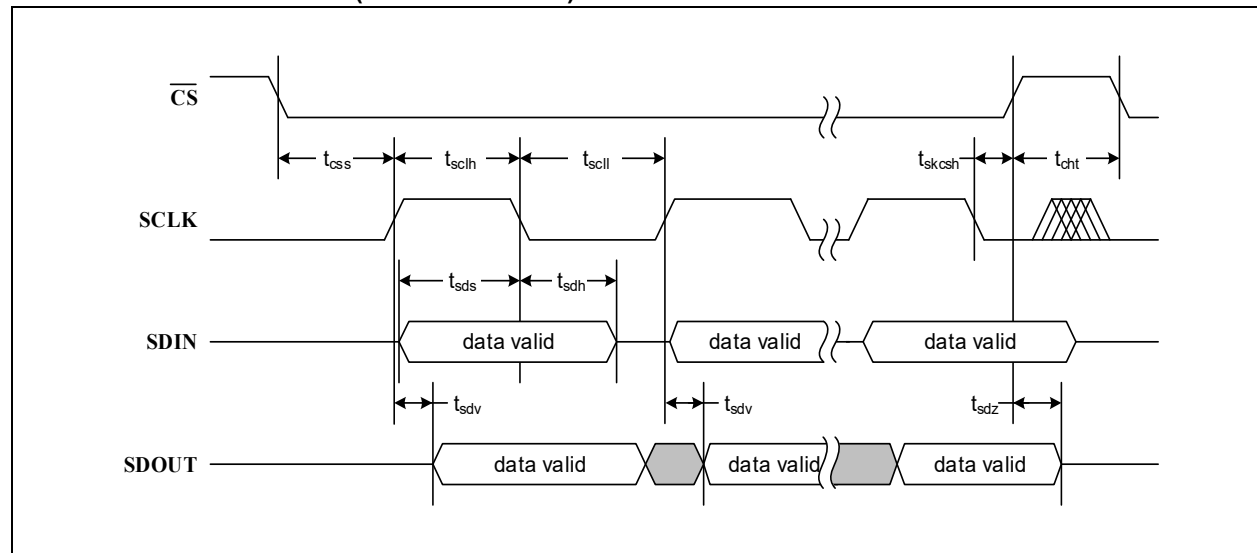
## 14.8.2 CLOCK MODE 1

$T_J = -40$  to  $125$  °C;  $V_{DDCn}, V_{DDA18} = 1.8$  V  $\pm 5$  %;  $V_{DDPn}, V_{DDA33}, V_{DDAUn} = 3.3$  V  $\pm 5$  %; ePAD GND = 0.0 V; Load Capacitance = 30 pF; PLL locked at 48 kHz; unless otherwise noted.

**TABLE 14-14: SPI PORT (CLOCK MODE 1)**

| Parameter                          | SPI Slave   |     |     |         |
|------------------------------------|-------------|-----|-----|---------|
|                                    | Symbol      | Min | Max | Unit    |
| SCLK frequency                     | $f_{scl}$   |     | 25  | MHz     |
| SCLK low time                      | $t_{scll}$  | 15  |     | ns      |
| SCLK high time                     | $t_{sclh}$  | 15  |     | ns      |
| SCLK low to $\overline{CS}$ high   | $t_{skcsh}$ | 5   |     | ns      |
| $\overline{CS}$ low to SCLK rising | $t_{css}$   | 8   |     | ns      |
| $\overline{CS}$ high time          | $t_{cht}$   | 1   |     | $\mu$ s |
| SDIN valid to SCLK falling         | $t_{sds}$   | 2   |     | ns      |
| SDIN hold from SCLK falling        | $t_{sdh}$   | 5   |     | ns      |
| SCLK rising to SDOUT valid         | $t_{sdv}$   |     | 11  | ns      |
| $\overline{CS}$ high to SDOUT Hi-Z | $t_{sdz}$   |     | 8   | ns      |

**FIGURE 14-11: SPI TIMING (CLOCK MODE 1)**



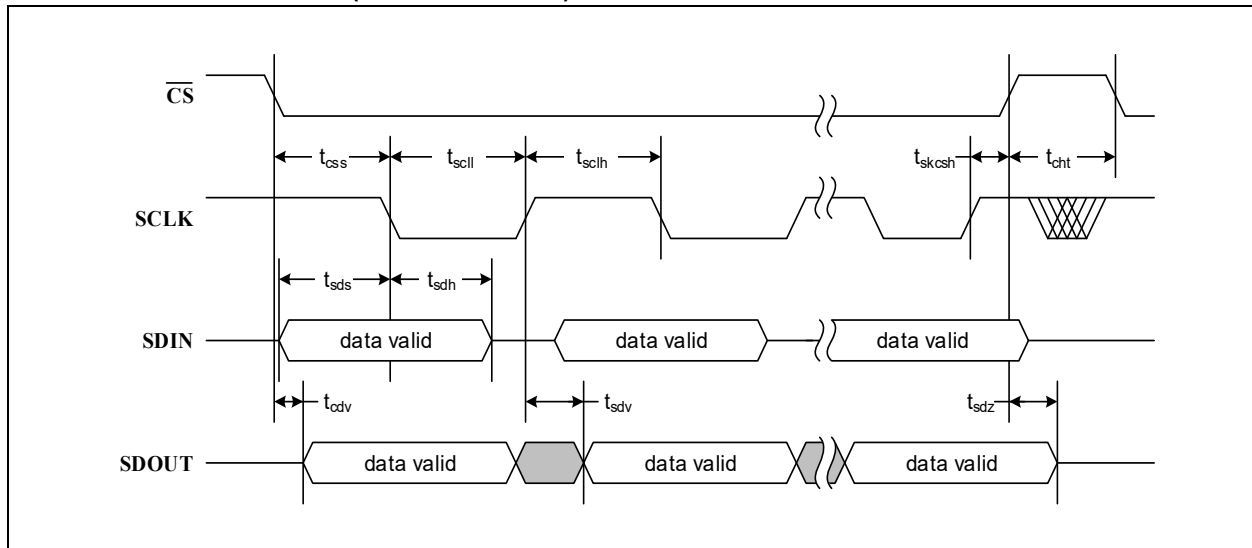
## 14.8.3 CLOCK MODE 2

$T_J = -40$  to  $125$  °C;  $V_{DDCn}, V_{DDA18} = 1.8$  V  $\pm 5$  %;  $V_{DDPn}, V_{DDA33}, V_{DDAUn} = 3.3$  V  $\pm 5$  %; ePAD GND = 0.0 V; Load Capacitance = 30 pF; PLL locked at 48 kHz; unless otherwise noted.

**TABLE 14-15: SPI PORT (CLOCK MODE 2)**

| Parameter                           | SPI Slave   |     |     |         |
|-------------------------------------|-------------|-----|-----|---------|
|                                     | Symbol      | Min | Max | Unit    |
| SCLK frequency                      | $f_{scl}$   |     | 25  | MHz     |
| SCLK low time                       | $t_{scll}$  | 15  |     | ns      |
| SCLK high time                      | $t_{sclh}$  | 15  |     | ns      |
| SCLK high to $\overline{CS}$ high   | $t_{skcsh}$ | 5   |     | ns      |
| $\overline{CS}$ low to SCLK falling | $t_{css}$   | 8   |     | ns      |
| $\overline{CS}$ low to SDOUT valid  | $t_{cdv}$   |     | 12  | ns      |
| $\overline{CS}$ high time           | $t_{cht}$   | 1   |     | $\mu$ s |
| SDIN valid to SCLK falling          | $t_{sds}$   | 2   |     | ns      |
| SDIN hold from SCLK falling         | $t_{sdh}$   | 5   |     | ns      |
| SCLK rising to SDOUT valid          | $t_{sdv}$   |     | 11  | ns      |
| $\overline{CS}$ high to SDOUT Hi-Z  | $t_{sdz}$   |     | 8   | ns      |

**FIGURE 14-12: SPI TIMING (CLOCK MODE 2)**



# OS81212/4/6

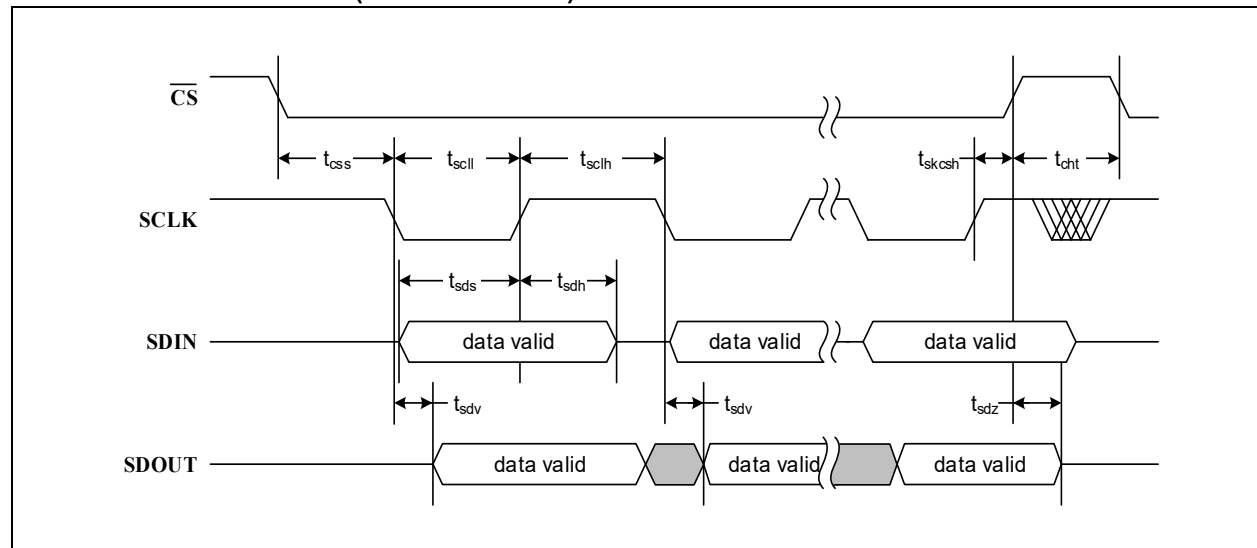
## 14.8.4 CLOCK MODE 3

$T_J = -40$  to  $125$  °C;  $V_{DDCn}, V_{DDA18} = 1.8$  V  $\pm 5$  %;  $V_{DDPn}, V_{DDA33}, V_{DDAUn} = 3.3$  V  $\pm 5$  %; ePAD GND = 0.0 V; Load Capacitance = 30 pF; PLL locked at 48 kHz; unless otherwise noted.

**TABLE 14-16: SPI PORT (CLOCK MODE 3)**

| Parameter                           | SPI Slave   |     |     |         |
|-------------------------------------|-------------|-----|-----|---------|
|                                     | Symbol      | Min | Max | Unit    |
| SCLK frequency                      | $f_{scl}$   |     | 25  | MHz     |
| SCLK low time                       | $t_{scll}$  | 15  |     | ns      |
| SCLK high time                      | $t_{sclh}$  | 15  |     | ns      |
| SCLK high to $\overline{CS}$ high   | $t_{skcsh}$ | 5   |     | ns      |
| $\overline{CS}$ low to SCLK falling | $t_{css}$   | 8   |     | ns      |
| $\overline{CS}$ high time           | $t_{cht}$   | 1   |     | $\mu$ s |
| SDIN valid to SCLK rising           | $t_{sds}$   | 2   |     | ns      |
| SDIN hold from SCLK rising          | $t_{sdh}$   | 5   |     | ns      |
| SCLK falling to SDOUT valid         | $t_{sdv}$   |     | 11  | ns      |
| $\overline{CS}$ high to SDOUT Hi-Z  | $t_{sdz}$   |     | 8   | ns      |

**FIGURE 14-13: SPI TIMING (CLOCK MODE 3)**



## 14.9 I<sup>2</sup>C Port

T<sub>J</sub> = -40 to 125 °C; VDDCn,VDDA18 = 1.8 V ±5 %; VDDPn,VDDA33,VDDAUn = 3.3 V ±5 %; ePAD GND = 0.0 V; Load Capacitance = 30 pF; PLL locked at 48 kHz; unless otherwise noted.

**TABLE 14-17: I<sup>2</sup>C PORT (SLAVE MODE)**

| Parameter   | Symbol            | Min | Max | Unit |
|---|-------------------|-----|-----|------|
| <b>Slave Mode:</b>  |                   |     |     |      |
| SCL frequency   | f <sub>scl</sub>  |     | 400 | kHz  |
| Bus free between transmissions (SDA high time between stop and start) | t <sub>buf</sub>  | 1.3 |     | µs   |
| Start condition hold time (SDA falling to SCL falling)                | t <sub>stah</sub> | 0.6 |     | µs   |
| SCL low   | t <sub>scll</sub> | 1.3 |     | µs   |
| SCL high  | t <sub>sclh</sub> | 0.6 |     | µs   |
| SDA input hold from SCL falling                                       | t <sub>sdah</sub> | 0   | 900 | ns   |
| SDA output valid from SCL falling                                     | t <sub>ov</sub>   |     | 900 | ns   |
| SDA input valid to SCL rising   | t <sub>sdas</sub> | 100 |     | ns   |
| (repeated) start condition setup time                                 | t <sub>stas</sub> | 0.6 |     | µs   |
| SDA and SCL rise time <sup>1</sup>                                    | t <sub>r</sub>    |     | 300 | ns   |
| SDA and SCL fall time <sup>1</sup>                                    | t <sub>f</sub>    |     | 300 | ns   |
| Stop condition setup time (SCL rising to SDA rising)                  | t <sub>stps</sub> | 0.6 |     | µs   |

**Note 1:** Rise and fall time specifications are based on the V<sub>IL</sub> and V<sub>IH</sub> values given in the [I2C-Bus Specification \[7\]](#) and not the V<sub>IL</sub> and V<sub>IH</sub> specifications given in [Table 14-3](#).

**TABLE 14-18: I<sup>2</sup>C PORT (MASTER MODE)**

| Parameter   | Symbol            | Min | Max  | Unit |
|---|-------------------|-----|------|------|
| <b>Master Mode (Standard-mode):</b>                                   |                   |     |      |      |
| SCL frequency   | f <sub>scl</sub>  |     | 100  | kHz  |
| Bus free between transmissions (SDA high time between stop and start) | t <sub>buf</sub>  | 4.7 |      | µs   |
| Start condition hold time (SDA falling to SCL falling)                | t <sub>stah</sub> | 4.0 |      | µs   |
| SCL low   | t <sub>scll</sub> | 4.7 |      | µs   |
| SCL high  | t <sub>sclh</sub> | 4.0 |      | µs   |
| SDA input hold from SCL falling                                       | t <sub>sdah</sub> | 0   | 3.45 | µs   |
| SDA output valid from SCL falling                                     | t <sub>ov</sub>   |     | 3.45 | µs   |
| SDA input valid to SCL rising   | t <sub>sdas</sub> | 250 |      | ns   |
| (repeated) start condition setup time                                 | t <sub>stas</sub> | 4.7 |      | µs   |
| SDA and SCL rise time <sup>1</sup>                                    | t <sub>r</sub>    |     | 1000 | ns   |
| SDA and SCL fall time <sup>1</sup>                                    | t <sub>f</sub>    |     | 300  | ns   |
| Stop condition setup time (SCL rising to SDA rising)                  | t <sub>stps</sub> | 4.0 |      | µs   |
| <b>Master Mode (Fast-mode):</b>                                       |                   |     |      |      |
| SCL frequency   | f <sub>scl</sub>  |     | 400  | kHz  |
| Bus free between transmissions (SDA high time between stop and start) | t <sub>buf</sub>  | 1.3 |      | µs   |
| Start condition hold time (SDA falling to SCL falling)                | t <sub>stah</sub> | 0.6 |      | µs   |
| SCL low   | t <sub>scll</sub> | 1.3 |      | µs   |

**Note 1:** Rise and fall time specifications are based on the V<sub>IL</sub> and V<sub>IH</sub> values given in the [I2C-Bus Specification \[7\]](#) and not the V<sub>IL</sub> and V<sub>IH</sub> specifications given in [Table 14-3](#).

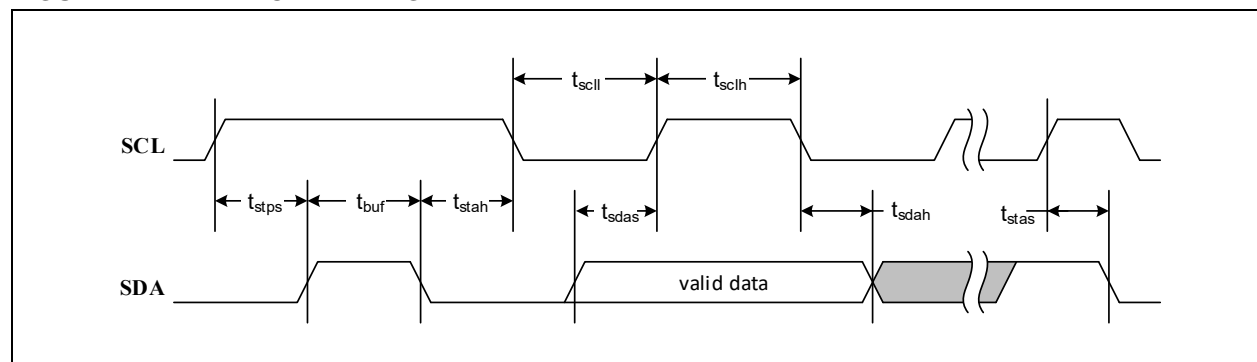
# OS81212/4/6

**TABLE 14-18: I<sup>2</sup>C PORT (MASTER MODE) (CONTINUED)**

| Parameter  | Symbol     | Min | Max | Unit    |
|--|------------|-----|-----|---------|
| SCL high   | $t_{sclh}$ | 0.6 |     | $\mu$ s |
| SDA input hold from SCL falling                      | $t_{sdah}$ | 0   | 900 | ns      |
| SDA output valid from SCL falling                    | $t_{ov}$   |     | 900 | ns      |
| SDA input valid to SCL rising                        | $t_{sdas}$ | 100 |     | ns      |
| (repeated) start condition setup time                | $t_{stas}$ | 0.6 |     | $\mu$ s |
| SDA and SCL rise time <sup>1</sup>                   | $t_r$      |     | 300 | ns      |
| SDA and SCL fall time <sup>1</sup>                   | $t_f$      |     | 300 | ns      |
| Stop condition setup time (SCL rising to SDA rising) | $t_{stps}$ | 0.6 |     | $\mu$ s |

**Note 1:** Rise and fall time specifications are based on the  $V_{IL}$  and  $V_{IH}$  values given in the [I2C-Bus Specification \[7\]](#) and not the  $V_{IL}$  and  $V_{IH}$  specifications given in [Table 14-3](#).

**FIGURE 14-14: I<sup>2</sup>C PORT TIMING**





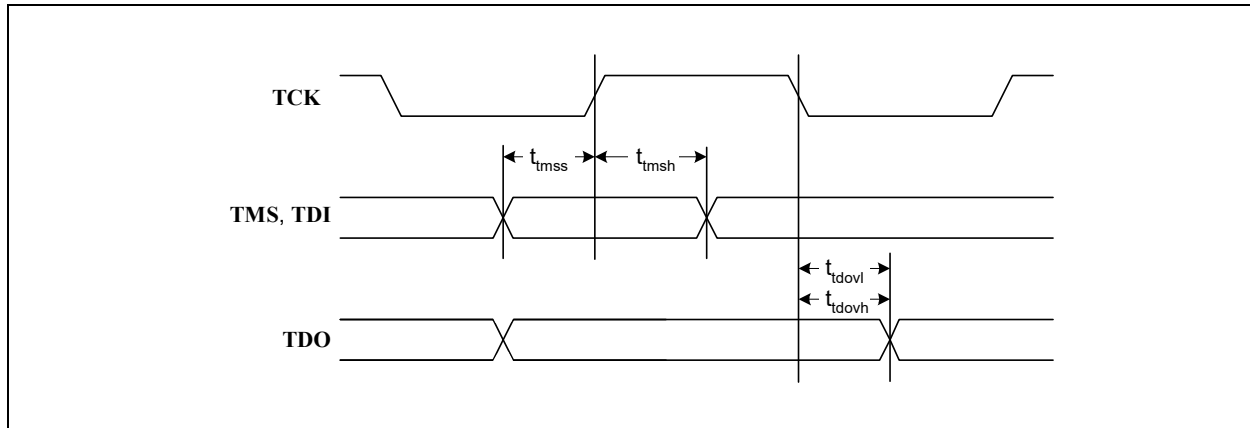
## 14.10 JTAG Port

$T_J = -40$  to  $125$  °C;  $V_{DDCn}, V_{DDA18} = 1.8\text{ V} \pm 5\%$ ;  $V_{DDPn}, V_{DDA33}, V_{DDAUn} = 3.3\text{ V} \pm 5\%$ ; ePAD GND = 0.0 V; Load Capacitance = 30 pF; PLL locked at 48 kHz; unless otherwise noted.

**TABLE 14-19: JTAG PORT**

| Parameter                                   | Symbol      | Min | Max | Unit |
|---|-------------|-----|-----|------|
| Maximum TCK Frequency                       | $f_{tck}$   |     | 10  | MHz  |
| TDO output valid high from TCK falling edge | $t_{tdovl}$ |     | 25  | ns   |
| TDO output valid low from TCK falling edge  | $t_{tdovh}$ |     | 25  | ns   |
| TMS and TDI input setup to TCK rising edge  | $t_{tmss}$  | 5   |     | ns   |
| TMS and TDI input hold from TCK rising edge | $t_{tmsh}$  | 5   |     | ns   |

**FIGURE 14-15: JTAG TIMING**



# OS81212/4/6

## 15.0 APPLICATION INFORMATION

### 15.1 Power

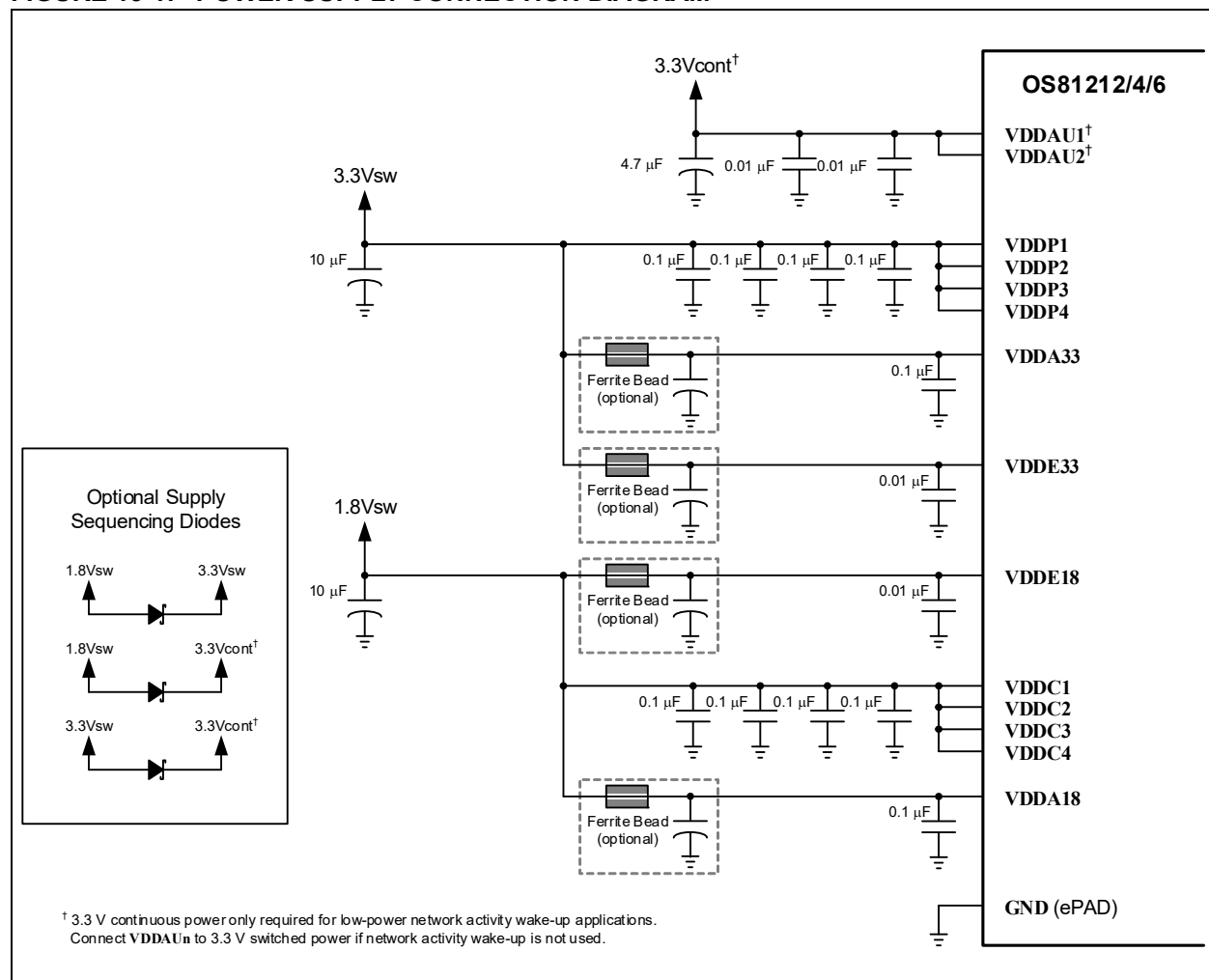
The OS81212/4/6 requires 3.3 V power for the **VDDP<sub>n</sub>**, **VDDA33**, and **VDDE33** pins and 1.8 V power for the **VDDC<sub>n</sub>**, **VDDA18**, and **VDDE18** pins. Additionally, for INIC to support network activity detection while in a low power mode, a continuous 3.3 V power supply should be connected to **VDDAU<sub>n</sub>** to power the internal bPHY activity detector.

Typical network ECUs support a *Sleep Power State* where continuous power is always connected to the ECU to allow the device to wake the ECU due to network activity. For networks requiring network activity detection, **VDDAU<sub>n</sub>** should be connected to 3.3 V continuous power to support the *Sleep Power State*, as shown in Figure 15-1. When using the OS81212/4/6 in a network ECU that does not implement a *Sleep Power State* and wake from network activity, **VDDAU<sub>n</sub>** should be connected to 3.3 V switched power. In this case, the INIC does not require 3.3 V continuous power.

The 1.8 V switched power supply must never exceed the 3.3 V switched supply by more than 0.5 V. When a continuous 3.3 V supply is used to implement a *Sleep Power State*, the 1.8 V and 3.3 V supplies must never exceed the 3.3 V continuous power supply by more than 0.5 V. Furthermore, digital input pins must never be driven to more than the **VDDP<sub>n</sub>** supply, and analog input pins should never be driven to more than the **VDDAU<sub>n</sub>** supply. These requirements are applicable to power-up and power-down, as well as normal operational conditions.

Figure 15-1 illustrates a typical power arrangement for the OS81212/4/6 with the power supply architecture and decoupling required to minimize jitter effects.

**FIGURE 15-1: POWER SUPPLY CONNECTION DIAGRAM**



## 15.2 Optional Components

Figure 15-1 shows Schottky diodes connected between the various power supply rails and ferrite beads on the analog power supply pins. The use of these components is optional and should be considered on a case-by-case basis depending on application requirements and limitations.

As noted in Section 15.1 “Power”, the 1.8 V switched supply must never exceed the 3.3 V switched supply by more than 0.5 V. The 1.8 V and 3.3 V switched supplies must also never exceed the 3.3 V continuous supply by more than 0.5 V in applications implementing a *Sleep Power State*. One approach to satisfying these power sequencing requirements is to connect a Schottky diode between the supplies to prevent the lower voltage supply from exceeding the higher supply by more than a forward diode drop. See Figure 15-1. If used, these diodes must be sized appropriately; the forward voltage drop must be less than 0.5 V when the diode conducts current to protect the OS81212/4/6 in a power supply fault condition.

Ferrite beads may optionally be added to **VDDA18**, **VDDA33**, **VDDE18**, and **VDDE33** for increased noise immunity in EMI-sensitive applications. Depending on the properties of the ferrite bead, their combination with the decoupling capacitors may cause a resonance peaking at lower frequencies leading to an undesired amplification of low-frequency noise in the system resulting in increased jitter and electromagnetic radiation. Since the ferrite bead selection is highly dependent on the noise present in the system, which varies from design to design, the addition of a large bulk capacitor, typically 10  $\mu$ F, is recommended to be placed on the INIC side of the ferrite bead. See Figure 15-1. During the prototype phase, it is recommended to include the option for the bulk capacitors at the ferrite beads should the need arise to populate them to reduce the effects of resonance peaking in the circuit.

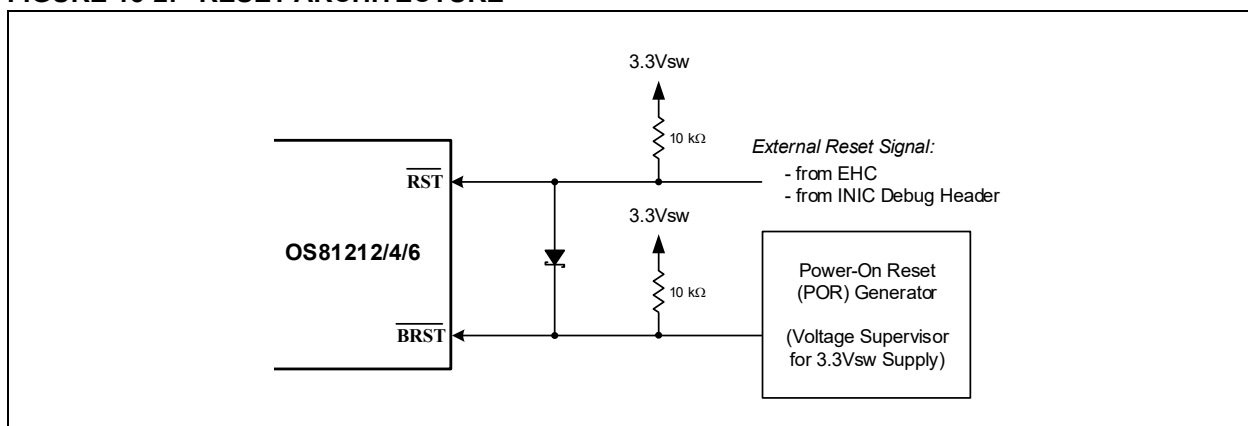
## 15.3 Reset

Once all power supplies have stabilized, the OS81212/4/6 must be reset ( $\overline{\text{RST}}$  low) for normal operation. A proper reset is when all supplies are  $\geq 90\%$  of their target voltage; however, electrical specifications are applicable only when supplies are within  $\pm 5\%$  of their target voltage. Once initialized, INIC is typically reset by the EHC only when a fatal error occurs, or when loading the *INIC Configuration String* (see the [INIC Interface Specification \[4\]](#)).

If the external Power-On-Reset (POR) circuit does not drive the  $\overline{\text{RST}}$  line high (e.g. open-drain output), an external pull-up resistor to 3.3 V should be used. If the external POR circuit drives the  $\overline{\text{RST}}$  line high and low (e.g. push-pull output), a series resistor should be used between the POR circuit and the OS81212/4/6  $\overline{\text{RST}}$  pin in lieu of the pull-up. This series resistor allows INIC to be reset by something other than the POR circuit, such as the EHC or the [INICkit Tool \[3\]](#). The pull-up or series resistor should be at least 10 k $\Omega$  and the  $\overline{\text{RST}}$  pulse width should be long enough to allow the **BOOT** configuration pin to achieve its default state (see Section 14.4.2 “Reset and Configuration”).

Diode isolation is recommended on the OS81212/4/6  $\overline{\text{RST}}$  line. An example reset circuit is shown in Figure 15-2. The POR from the reset generator influences the INIC  $\overline{\text{RST}}$  pin; however, the Schottky diode ensures that an application reset (i.e. from the EHC) does not disturb the physical layer. The physical layer reset signal acts as a transmitter enable/disable signal that ensures glitch-free operation at the physical layer output during power supply ramp up/down. The physical layer reset signal must be connected to the OS81212/4/6  $\overline{\text{BRST}}$  pin.

FIGURE 15-2: RESET ARCHITECTURE



## 15.4 Physical Layer

The OS81212/4/6 INIC is used in 50 Mbit/s balanced media physical layer (bPHY) network applications. The integrated INIC bPHY Network Port supports the electrical transmission of a network bit-stream between two or more compatible network devices using low-cost unshielded twisted pair (UTP) wire.

The bPHY interface consists of:

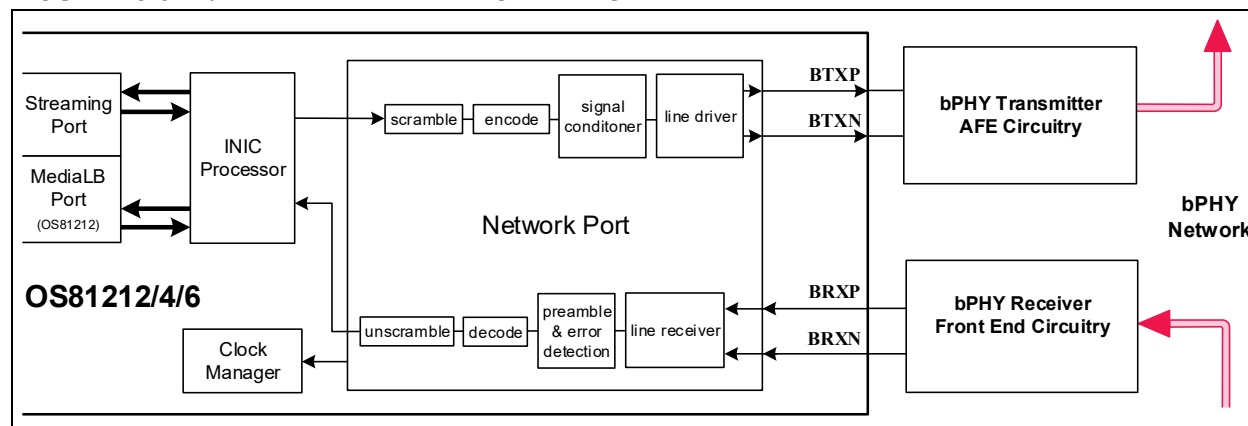
- **BTXP** - positive transmitter output
- **BTXN** - negative transmitter output
- **BRXP** - positive receiver input
- **BRXN** - negative receiver input
- **BRST** - network reset input
- **BSTATUS** - network activity status output

Figure 15-3 illustrates the connection between the INIC Network Port and the bPHY network. A large portion of the transmitter and receiver front end circuitry is used to reduce electromagnetic emissions and increase the overall system immunity to EMI. In order for data transmission to be reliable over a wide operating range, all signals along the transmission path need to be relatively noise free, with minimal jitter and pulse width distortion. As a result, both the transmitter and receiver front end circuits are designed for reducing electrical emissions, increasing system immunity to EMI, and minimizing reflections.

When **BRST** is driven low, the transmitter output is disabled. **BRST** should only be asserted by the power-on-reset generator and should not be connected directly to the INIC **RST** pin. Refer to Figure 15-2 for the recommended reset architecture. **BSTATUS** will be driven low when a valid network signal is detected and driven high when a qualified signal is not present.

**Note:** Contact Microchip for the [OS8121x Physical Layer Reference Design Application Note \[13\]](#) which describes the front end circuitry implementation in detail.

**FIGURE 15-3: bPHY INTERFACE - BLOCK DIAGRAM**



When transmitting data onto the network, the Network Port accepts data bytes from the INIC Processor, scrambles the data to reduce data-dependent jitter, conditions the signal for transmission over UTP, and transmits the encoded bit-stream onto **BTXP/BTXN**. These pins are driven by an internal digital-to-analog converter, which drives a low-pass filtered version of the bit-stream by sourcing current into external load resistors. The **BTXP/BTXN** pins interface with the transmitter front end circuitry which provides an analog low-pass filter.

When the OS81212/4/6 receives the serial network bit-stream at **BRXP/BRXN**, it converts the differential input to a single-ended signal. The Network Port detects the preamble in the received bit-stream, detects transmission errors, and works in conjunction with the Clock Manager to recover the clock from the bit-stream. The Network Port then decodes and unscrambles the bit-stream, and sends the network data to the INIC Processor as data bytes.

When the OS81212/4/6 is configured as a timing-slave, the Network Port is synchronized to the bit-stream recovered clock. When configured as the timing-master, the Network Port controls network timing, based on the external clock input. The transmitted network frames are aligned to this clock.

## 15.5 Other Application Pins

For normal operation, the **BOOT** pin should be pulled to 3.3 V. To support the *Customer Configuration Interface*, the on-board pull-up resistor on **BOOT** must not be less than 10 k $\Omega$ ; however, the actual resistor value should be chosen based on the input leakage of the INIC pin and on the other devices connected to the line. The JTAG Port pins **TMS**, **TCK**, **TDI**, and **TDO** require pull-up resistors to 3.3 V, regardless of whether or not the JTAG interface is used.

During the INIC initialization time,  $t_{inicinit}$ , the **ERR** pin becomes active. The **ERR** pin is driven high to indicate a Network Unlock, which occurs when two consecutive frame preambles are not received. At all other times, the **ERR** pin is driven low to indicate Network Lock. Network Lock occurs when three consecutive preambles are received with the correct timing. In a typical application, the **ERR** pin is used to control an LED to indicate Network Lock.

The **MUTE** pin is an optional output that should be pulled high by an on-board resistor (10 k $\Omega$  recommended) when used. If an event occurs that can corrupt the data for any registered synchronous streaming sink connection, the **MUTE** pin is driven low to inform external devices when muting is required. Refer to the [INIC Interface Specification \[4\]](#) for more information.

When an external crystal is used, the **XTI/XTO** pins should be connected to the crystal with no series resistor placed between the pins. The crystal used must be 384×Fs and comply with the specifications outlined in [Section 15.8 “Crystal Oscillator Selection”](#). If an external clock is used in lieu of a crystal oscillator, it must be connected to **XTI** and support the drive levels listed in [Chapter 14.0 “Electrical Characteristics”](#). In this scenario, **XTO** should float and have a stub capacitance less than 10 pF.

Jitter at **XTI** must be minimized in all applications. Driving an external clock signal is more prone to noise coupling events (e.g. crosstalk); therefore, a local crystal oscillator is the preferred method of providing the INIC clock reference. This directive is especially relevant when INIC operates as the network timing-master.

When external power management is used, the **PWROFF** pin should be pulled high through an external resistor. After initialization, INIC drives **PWROFF** low during normal operating conditions and releases **PWROFF** when the INIC Processor is ready to be shutdown (due to network inactivity or  $PS[1:0] = U_{Low}$ ). Refer to [Chapter 9.0 “External Power Management”](#) for more information.

## 15.6 Configuration and Debug

### 15.6.1 CONFIGURATION STRING

The OS81212/4/6 provides *One-Time Programmable* (OTP) memory that is used for storing power-up configuration settings. OTP memory is divided into two sections, which allows configuration settings to be written twice. When OTP memory is written once, the first section stores the configuration data and the second section is unused. When OTP memory is written again, the second section stores the configuration data and all data previously written into the first section is ignored. After power-up/reset, configuration settings are loaded from the appropriate OTP section into RAM.

The initial power-up configuration settings stored in OTP memory are collectively referred to as the *INIC Configuration String*. The *INIC Configuration String* defines the port(s) which will be automatically opened and includes initial hardware configuration information such as:

- I<sup>2</sup>C Port address and mode,
- Default port for INIC-EHC communication,
- MediaLB Port speed (OS81212),
- **RMCK** configuration,
- GPIO usage, and
- Network device mode (timing-master/timing-slave).

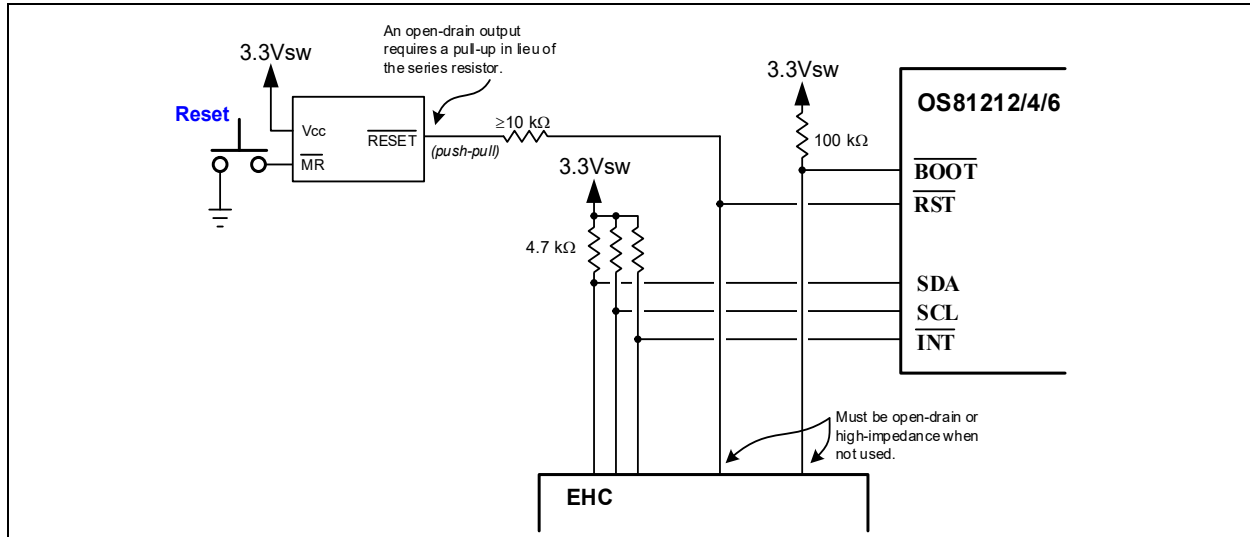
If OTP memory is not initialized, factory default settings for the *INIC Configuration String* included in the INIC firmware are used in its place. The OTP memory only needs to be programmed when the factory defaults are not adequate for the application. Refer to the [INIC Interface Specification \[4\]](#) for more information on the factory default values and customization of the *INIC Configuration String*.

### 15.6.2 EHC VIA I<sup>2</sup>C PORT

The I<sup>2</sup>C Port can be used by the EHC to program the *INIC Configuration String*. This approach is suitable for production-level applications. To support this feature, the EHC must be connected to the INIC I<sup>2</sup>C Port (**SDA**, **SCL**, and **INT**) and be capable of controlling the **BOOT** and **RST** pins. The required connections are shown in [Figure 15-4](#).

# OS81212/4/6

FIGURE 15-4: CONNECTION DIAGRAM FOR I<sup>2</sup>C PORT INTERFACE



The sequence required of the EHC to program the OS81212/4/6 INIC is as follows:

- Hold the OS81212/4/6 in reset (drive  $\overline{\text{RST}}$  pin low),
- Hold the  $\overline{\text{BOOT}}$  pin low,
- Release the OS81212/4/6 from reset ( $\overline{\text{RST}}$  pin high),
- Erase and program OS81212/4/6 INIC, as described in the [Device Update Process User's Guide \[14\]](#),
- Release the  $\overline{\text{BOOT}}$  pin, and
- Reset the OS81212/4/6 to resume a normal mode of operation.

### 15.6.3 INICkit VIA JTAG PORT

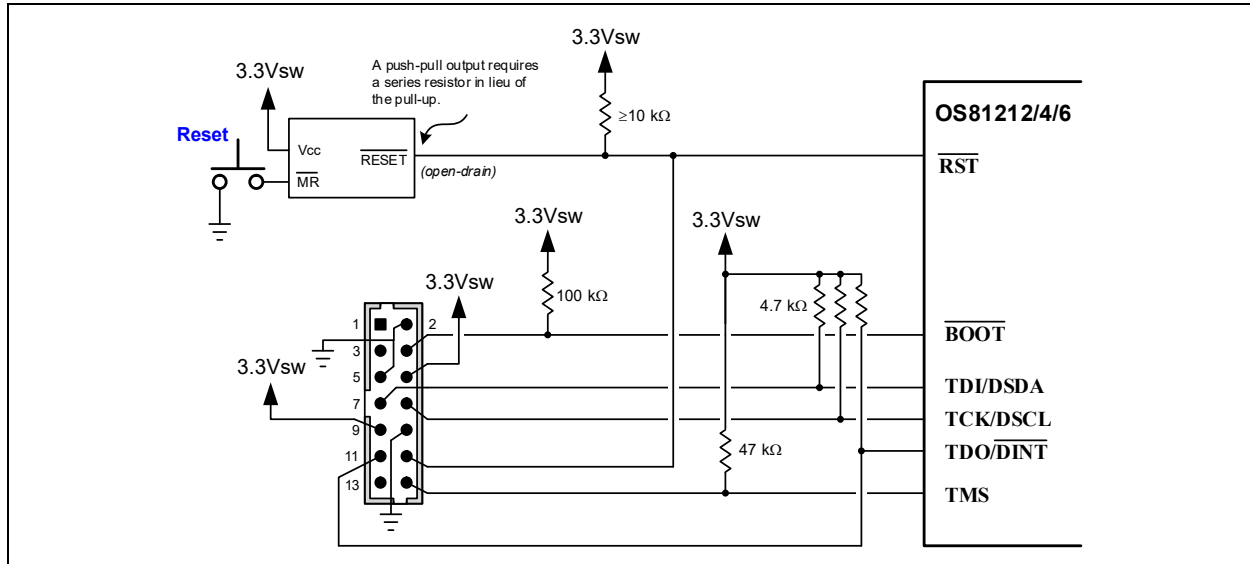
The OS81212/4/6 JTAG Port can be converted into a configuration interface for use with the Microchip [INICkit Tool \[3\]](#). *INICkit* requires that the application PCB provide a standard 14-pin (2x7) 2 mm header (such as Molex 87332-1420 or equivalent). The interface between the *INICkit tool* and the OS81212/4/6 is also referred to as the *Customer Configuration Interface*.

In addition to writing the *INIC Configuration String* into OTP memory, *INICkit* provides debugging capabilities by allowing users access to INIC (via PC software) for:

- Exploring internal INIC properties and states while it is operating in the target platform,
- Viewing internal INIC states and routing configurations graphically, and
- Creating an INIC data memory snapshot as a file dump.

[Figure 15-5](#) illustrates the necessary connections between the OS81212/4/6 and the configuration/debug header that accommodates *INICkit*. To allow the header to drive the  $\overline{\text{BOOT}}$  and  $\overline{\text{RST}}$  pins during configuration, the pull-up resistors on the lines should be greater than 10 k $\Omega$ . If the  $\overline{\text{BOOT}}$  and  $\overline{\text{RST}}$  lines are also connected to the EHC (as described in [Section 15.6.2 "EHC via I2C Port"](#)), the EHC must use open-drain logic or set its outputs as high-impedance when the *INICkit tool* connects to the OS81212/4/6 via the configuration/debug header. It is strongly recommended to include the footprint for the configuration/debug header even if the header is not populated.

**FIGURE 15-5: CONFIGURATION/DEBUG HEADER**



Contact Microchip for more information on the [INICKit Tool \[3\]](#).

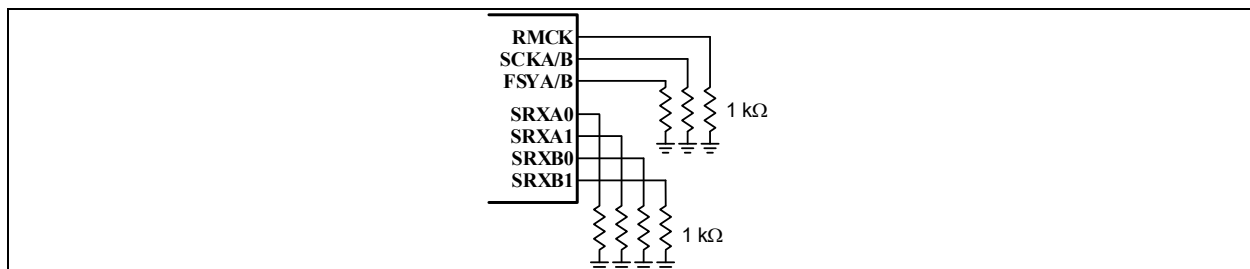
## 15.7 Termination

The OS81212/4/6 input pins should not be left floating in the application. Unused input pins can be connected directly to the power supply or to ground, unless otherwise noted. Alternatively, pull-ups or pull-downs can be used (in lieu of a direct connection to power or ground) for protection during erroneous configuration as an output as shown in the following figures.

### 15.7.1 STREAMING PORT

Figure 15-6 provides a recommendation for unused pin termination on the OS81212/4/6 Streaming Port.

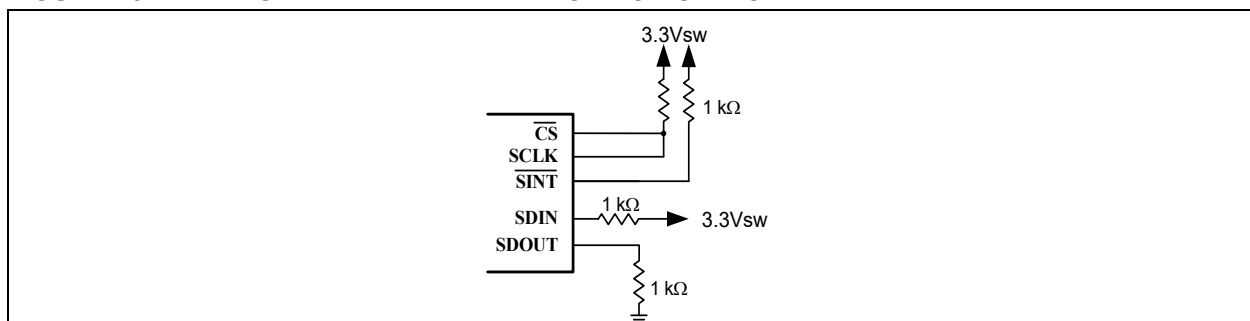
**FIGURE 15-6: RECOMMENDED TERMINATION FOR STREAMING PORT**



### 15.7.2 SPI PORT

Figure 15-7 provides a recommendation for unused pin termination on the OS81212/4 SPI Port.

**FIGURE 15-7: RECOMMENDED TERMINATION FOR SPI PORT**

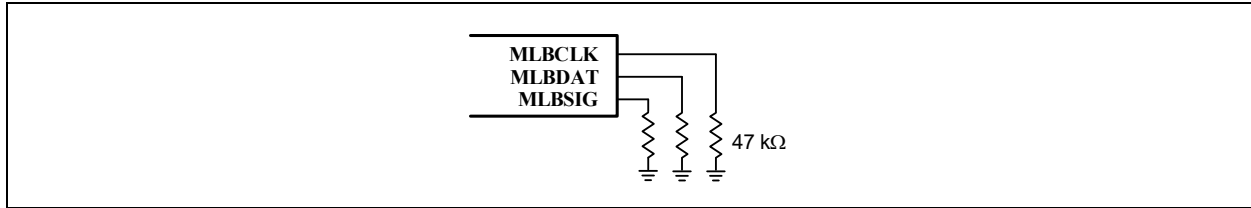


# OS81212/4/6

## 15.7.3 MEDIALB PORT

Figure 15-8 provides a recommendation for unused pin termination on the OS81212 MediaLB Port.

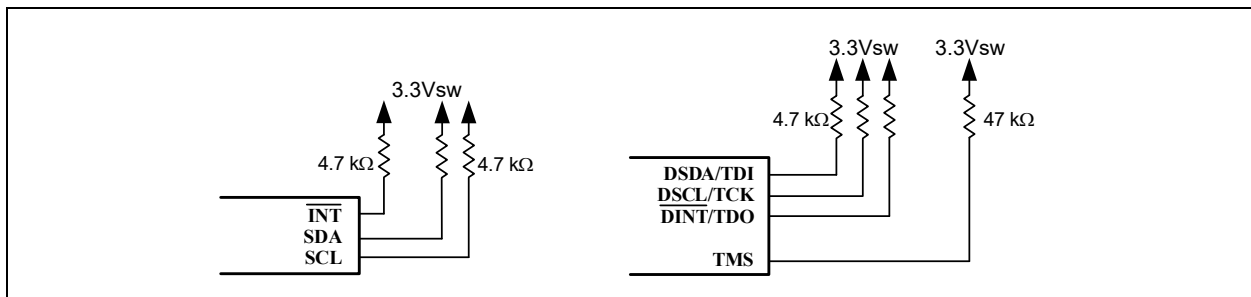
**FIGURE 15-8: RECOMMENDED TERMINATION FOR MediaLB PORT**



## 15.7.4 I<sup>2</sup>C AND JTAG PORTS

Figure 15-9 provides a recommendation for unused pin termination on the OS81212/4/6 I<sup>2</sup>C Port and JTAG Port.

**FIGURE 15-9: RECOMMENDED TERMINATION FOR I<sup>2</sup>C AND JTAG PORTS**



**Note:** Pull-ups are required on I<sup>2</sup>C buses, regardless of whether the ports are used.



## 15.8 Crystal Oscillator Selection

Several factors must be considered when selecting a crystal. These include load capacitance, oscillator margin, cut, and operating temperature.

Oscillator margin is a measure of the stability of an oscillator circuit, and is defined as the ratio of the oscillator's negative resistance ( $R_{NEG}$ ) to the crystal's ESR ( $R_{ESR}$ ), or:

### EQUATION 15-1: OSCILLATOR MARGIN MEASUREMENT

$$\text{Margin} = \frac{|R_{NEG}|}{R_{ESR}} = \frac{|R_{VAR}| + R_{ESR}}{R_{ESR}}$$

The negative resistance can be measured by placing a variable resistor ( $R_{VAR}$ ) in series with the crystal and finding the largest resistor value where the crystal still starts up properly. This point would be just below where the oscillator does not start-up or where the start-up time is excessively long. Ideally, oscillator margin should be greater than 10, and should be at least 5. Smaller oscillator margin can affect the ability of the oscillator to start up.

The load capacitance, specified when ordering the crystal, is the series combination of the capacitance on each leg of the crystal. This capacitance includes not only the added capacitors, but also PCB trace (shunt) capacitance and chip pin capacitance. Larger capacitors also have a negative effect on oscillator margin. External capacitors on each leg (C1 and C2 in [Figure 10-1](#)) are typically in the 10 to 22 pF range. The transconductance ( $g_m$ ) of the internal INIC inverting amplifier is nominally 4.56 mS.

**TABLE 15-1: CRYSTAL OSCILLATOR SPECIFICATIONS**

| Name        | Value             | Description   |
|-------------|-------------------|---|
| Frequency   | 384×Fs            | Frequency relative to operating network frame rate (Fs) |
| Correlation | Parallel Resonant | Mode of oscillation                                     |
| Osc. Mode   | Fundamental       | Oscillation mode or operation mode                      |
| $C_L$       | 10-22 pF          | Load capacitance (typical)                              |
| $C_O$       | 7 pF              | Recommended maximum shunt (parallel) capacitance        |
| ESR         | 100 $\Omega$      | Recommended maximum equivalent series resistance        |
| Drive Level | 50 $\mu$ W        | Drive level (typical)                                   |
| Cut         | AT                | AT cut produces the best temperature stability          |
| Tolerance   | $\pm$ 200 ppm     | Frequency tolerance over all conditions                 |

The crystal cut and tolerance value listed in [Table 15-1](#) are *typical* values and may be changed to suit differing system requirements. Higher ESR values (than those listed in the [Table 15-1](#)) run the risk of having start-up problems and should be thoroughly tested before being used. Contact the crystal manufacturer for more information.

## 15.9 Layout Guidelines

INIC board designs should follow the basic guidelines outlined throughout this chapter, with special consideration given to component selection and placement.

### 15.9.1 POWER AND DECOUPLING

A dedicated decoupling capacitor should be used at each OS81212/4/6 power pin for localized decoupling. Typical values are shown in [Figure 15-1](#). For EMI-sensitive applications C0G capacitors are strongly recommended for their superior performance at high frequencies.

To reduce trace impedance and keep the loop areas small, decoupling capacitors should be positioned as close as possible to their respective OS81212/4/6 power pins. The recommended placement of decoupling capacitors is on the same side of the PCB as the OS81212/4/6. However, decoupling capacitors may be mounted on the opposite side of the PCB to reduce the distance between the capacitor and the pin, since inductance from vias is less harmful than long traces that may allow external interference. Two vias to the ground plane should be used for each decoupling capacitor to minimize inductance.

# OS81212/4/6

---

In situations where both large (e.g. electrolytic or tantalum) and small (e.g. ceramic) capacitors are used, the smaller capacitor should take priority for placement and layout optimization.

As in all low-noise circuits, two vias should be used from power and ground traces to the respective power and ground planes to reduce parasitic inductance, whenever possible. Additionally, power traces should be as wide as practical. The exposed paddle of the QFN package is the primary ground connection for the OS81212/4/6 and must be adequately connected to the PCB ground plane.

Ground-plane fill is used under the part to minimize the impedance between ground connections, which also helps minimize EMI. If a ground-plane fill must be broken by a trace, multiple stitching vias should connect each section of the fill to a reference ground plane on another layer. Vias that connect high frequency decoupling capacitors to the ground plane should be as close as possible to the OS81212/4/6 to minimize loop area back to the package exposed paddle ground.

## 15.9.2 MediaLB PORT

The OS81212 *Media Local Bus* (MediaLB) 3-pin interface is provided to facilitate high-speed data exchange between INIC and the EHC. MediaLB is intended for use on a single PCB. For effective operation, especially at higher bus speeds, signal routing and component placement is extremely important. Refer to the [MediaLB Specification \[5\]](#) for application and implementation guidelines including trace length requirements and layout examples.

## 15.9.3 PHYSICAL LAYER

Proper hardware implementation (e.g. component placement, signal routing and shield/chassis/digital grounding) is essential for achieving stringent automotive performance. For optimum performance, attention should be given to the routing and placement of physical layer components. When possible, the signals should be routed on a single layer and the use of vias should be minimized. Additionally, traces should be routed over a continuous reference plane, preferably a ground plane. To minimize reflections, two 45 degree turns should be used rather than a single 90 degree turn. Crosstalk should be avoided by not running other signals in parallel to the transmit and receive lines for long distances.

|  |
|--|
| <p><b>Note:</b> For Physical Layer layout recommendations, please refer to the latest <a href="#">OS8121x Physical Layer Reference Design Application Note [13]</a> available from Microchip Support upon request.</p> |
|--|

## 15.9.4 CRYSTAL OSCILLATOR

When using an external crystal, the distance between the INIC XTI/XTO pins and the external crystal should be minimized. As mentioned in [Chapter 10.0 “Clock Manager”](#), jitter on XTI must also be minimized. When an externally generated clock signal is used in lieu of a crystal oscillator, the clock signal must be routed as a high-priority trace with adequate countermeasures against jitter and crosstalk.

## 15.9.5 THERMAL CONSIDERATIONS

The exposed paddle of the QFN package provides the ground connection for the OS81212/4/6 and must be adequately connected to the PCB ground plane. The paddle of the package must be soldered to the thermal land (thermal pad) for efficient heat dissipation and proper grounding. The thermal pad should be at least as large as the exposed portion of the package paddle and be connected to the ground plane of the board using heat tubes (plated hole vias without thermal reliefs). At least nine heat tubes are recommended.

The OS81212 56-pin QFN land pattern is shown in [Figure 16-5](#), and the land pattern for the OS81214/6 48-pin QFN is shown in [Figure 16-8](#).

## 15.9.6 MISCELLANEOUS

Floating areas of copper fill near the INIC must be avoided due to the risk of noise coupling into the system. If needed, a keep-out can be used around the OS81212/4/6 and its immediate peripheral circuitry (e.g. decoupling capacitors, crystal, termination components, etc.) to ensure isolated regions of floating copper are not inadvertently created when copper fill is poured. If a keep-out is not used, multiple stitching vias should connect any floating copper fills to an appropriate reference on other power or ground layers.

## 16.0 PACKAGING INFORMATION

### 16.1 Package Marking

FIGURE 16-1: OS81212 TOP MARKING

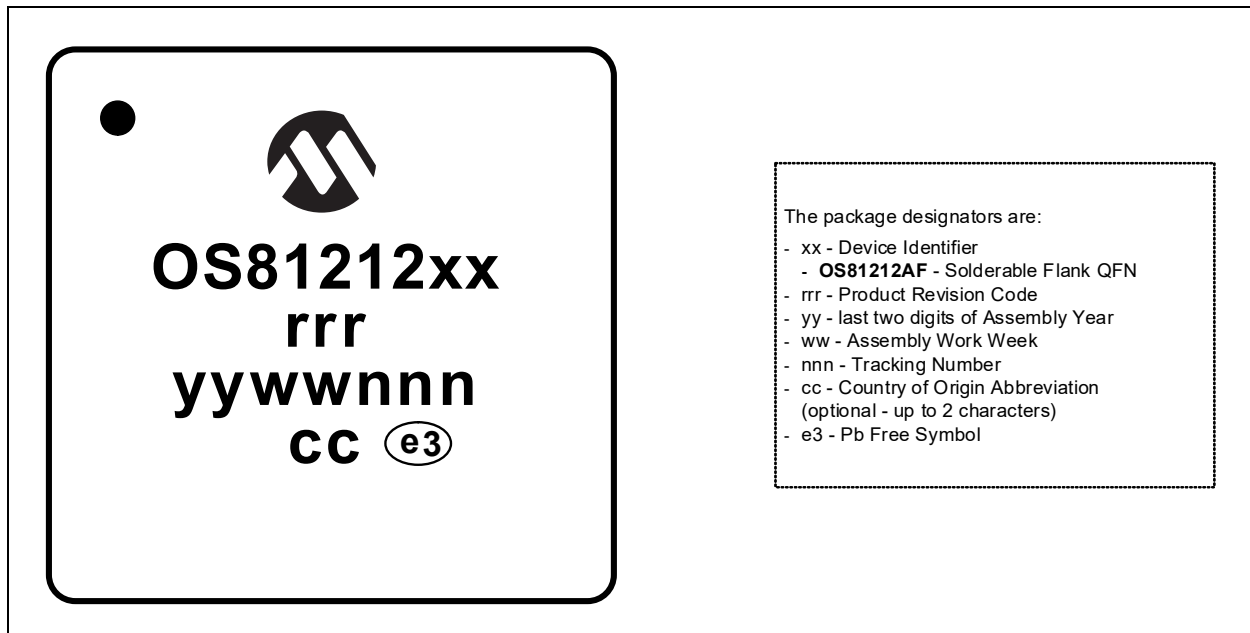
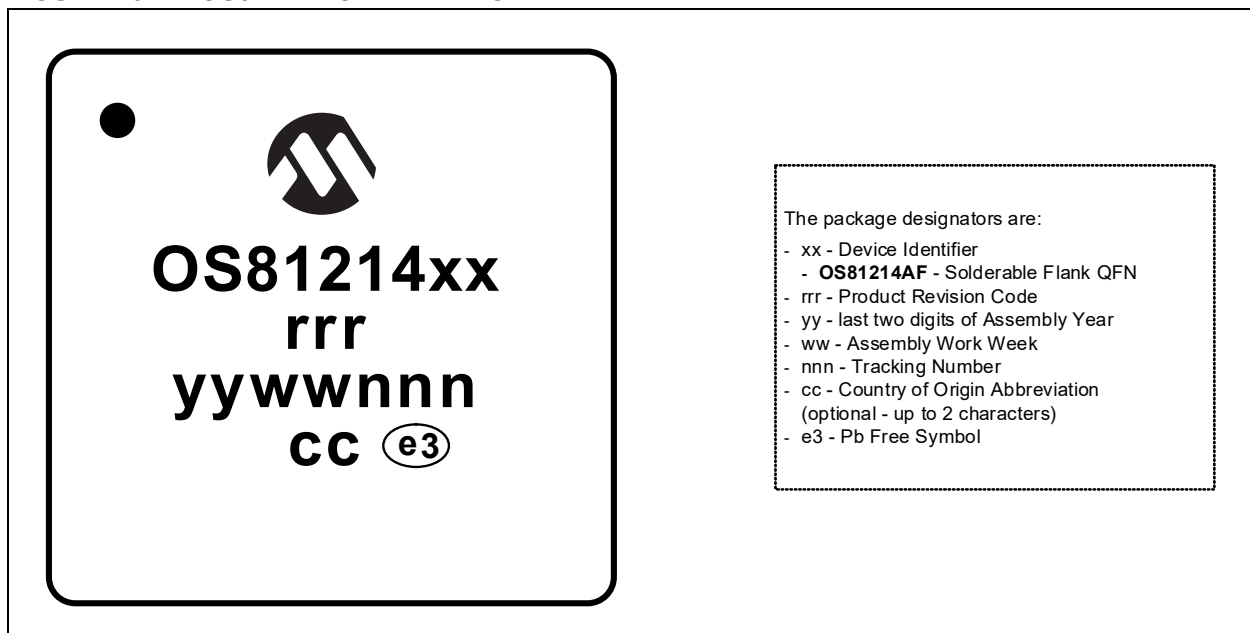
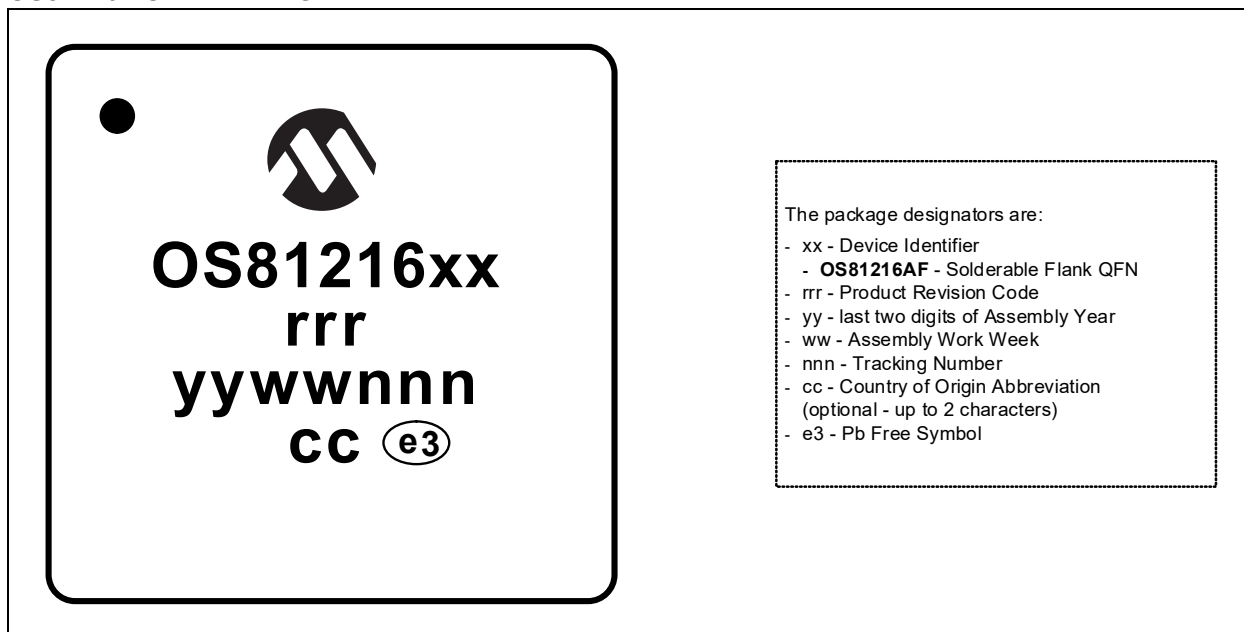


FIGURE 16-2: OS81214 TOP MARKING



# OS81212/4/6

## OS81216 TOP MARKING



## 16.2 Package Thermal Characteristics

**TABLE 16-1: OS81212 56-PIN SOLDERABLE FLANK QFN THERMAL CHARACTERISTICS**

| Parameter                   | Symbol        | Value | Unit |
|-----------------------------|---------------|-------|------|
| Typical Junction to Package | $\Psi_{JT}$   | 0.2   | °C/W |
| Typical Junction to Ambient | $\theta_{JA}$ | 25    | °C/W |

**TABLE 16-2: OS81214/6 48-PIN SOLDERABLE FLANK QFN THERMAL CHARACTERISTICS**

| Parameter                   | Symbol        | Value | Unit |
|-----------------------------|---------------|-------|------|
| Typical Junction to Package | $\Psi_{JT}$   | 0.2   | °C/W |
| Typical Junction to Ambient | $\theta_{JA}$ | 27    | °C/W |

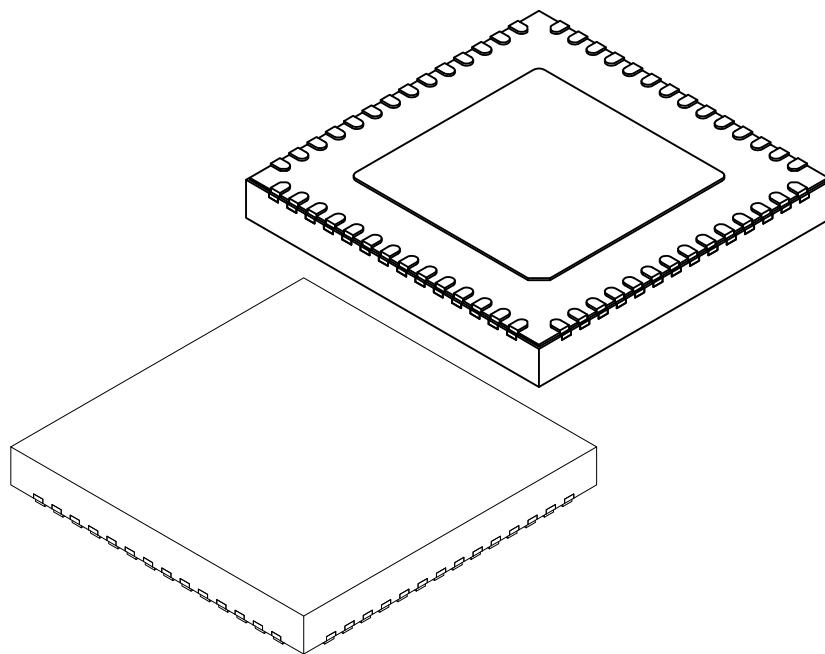


# OS81212/4/6

FIGURE 16-4: OS81212 56-PIN SOLDERABLE FLANK QFN PACKAGE DRAWING (2 OF 3)

## 56-Lead Very Thin Plastic Quad Flat, No Lead Package (NDC) - 8x8x1.0 mm Body [VQFN] With 5.15 mm Exposed Pad and Stepped Wettable Flanks

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



| Dimension Limits           | Units | MILLIMETERS |      |       |
|----------------------------|-------|-------------|------|-------|
|                            |       | MIN         | NOM  | MAX   |
| Number of Terminals        | N     | 56          |      |       |
| Pitch                      | e     | 0.50 BSC    |      |       |
| Overall Height             | A     | 0.80        | 0.90 | 1.00  |
| Standoff                   | A1    | 0.00        | 0.02 | 0.05  |
| Terminal Thickness         | A3    | 0.20 REF    |      |       |
| Overall Length             | D     | 8.00 BSC    |      |       |
| Exposed Pad Length         | D2    | 5.05        | 5.15 | 5.25  |
| Overall Width              | E     | 8.00 BSC    |      |       |
| Exposed Pad Width          | E2    | 5.05        | 5.15 | 5.25  |
| Terminal Width             | b     | 0.18        | 0.25 | 0.30  |
| Terminal Length            | L     | 0.35        | 0.40 | 0.45  |
| Terminal-to-Exposed-Pad    | K     | 1.03 REF    |      |       |
| Wettable Flank Step Length | D3    | -           | -    | 0.085 |
| Wettable Flank Step Height | A4    | 0.10        | -    | 0.19  |

**Notes:**

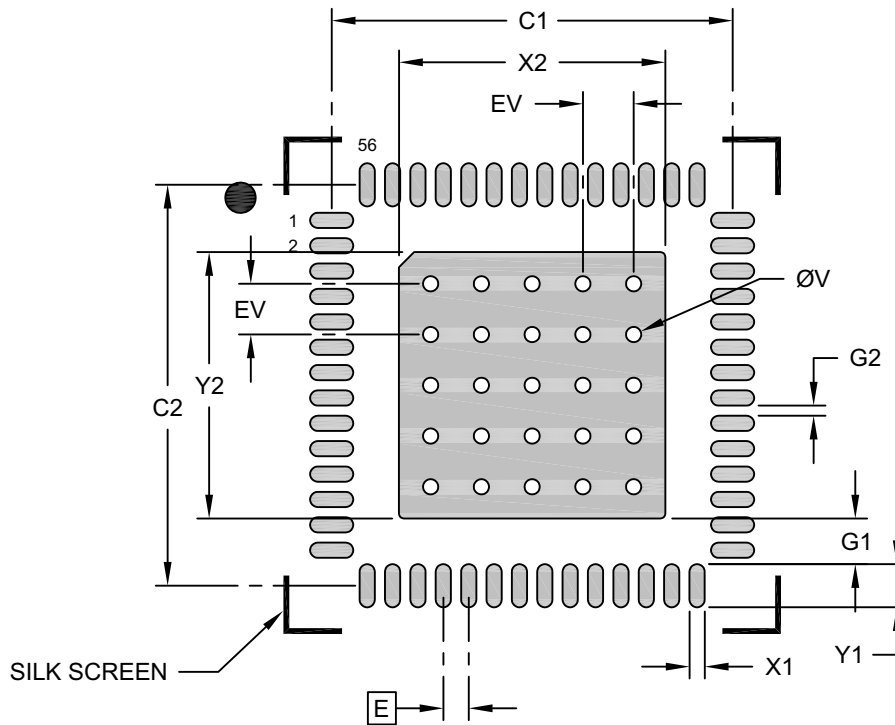
- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Package is saw singulated
- Dimensioning and tolerancing per ASME Y14.5M  
BSC: Basic Dimension. Theoretically exact value shown without tolerances.  
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-25510 Rev A Sheet 2 of 2

**FIGURE 16-5: OS81212 56-PIN SOLDERABLE FLANK QFN PACKAGE DRAWING (3 OF 3)**

**56-Lead Very Thin Plastic Quad Flat, No Lead Package (NDC) - 8x8x1.0 mm Body [VQFN]  
With 5.158 mm Exposed Pad and Stepped Wettable Flanks**

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



**RECOMMENDED LAND PATTERN**

| Dimension Limits                 | Units | MILLIMETERS |      |      |
|----------------------------------|-------|-------------|------|------|
|                                  |       | MIN         | NOM  | MAX  |
| Contact Pitch                    | E     | 0.50 BSC    |      |      |
| Optional Center Pad Width        | X2    |             |      | 5.25 |
| Optional Center Pad Length       | Y2    |             |      | 5.25 |
| Contact Pad Spacing              | C1    |             | 7.90 |      |
| Contact Pad Spacing              | C2    |             | 7.90 |      |
| Contact Pad Width (X56)          | X1    |             |      | 0.30 |
| Contact Pad Length (X56)         | Y1    |             |      | 0.85 |
| Contact Pad to Center Pad (X56)  | G1    | 0.85        |      |      |
| Contact Pad to Contact Pad (X52) | G2    | 0.20        |      |      |
| Thermal Via Diameter             | V     |             | 0.30 |      |
| Thermal Via Pitch                | EV    |             | 1.00 |      |

**Notes:**

1. Dimensioning and tolerancing per ASME Y14.5M  
BSC: Basic Dimension. Theoretically exact value shown without tolerances.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

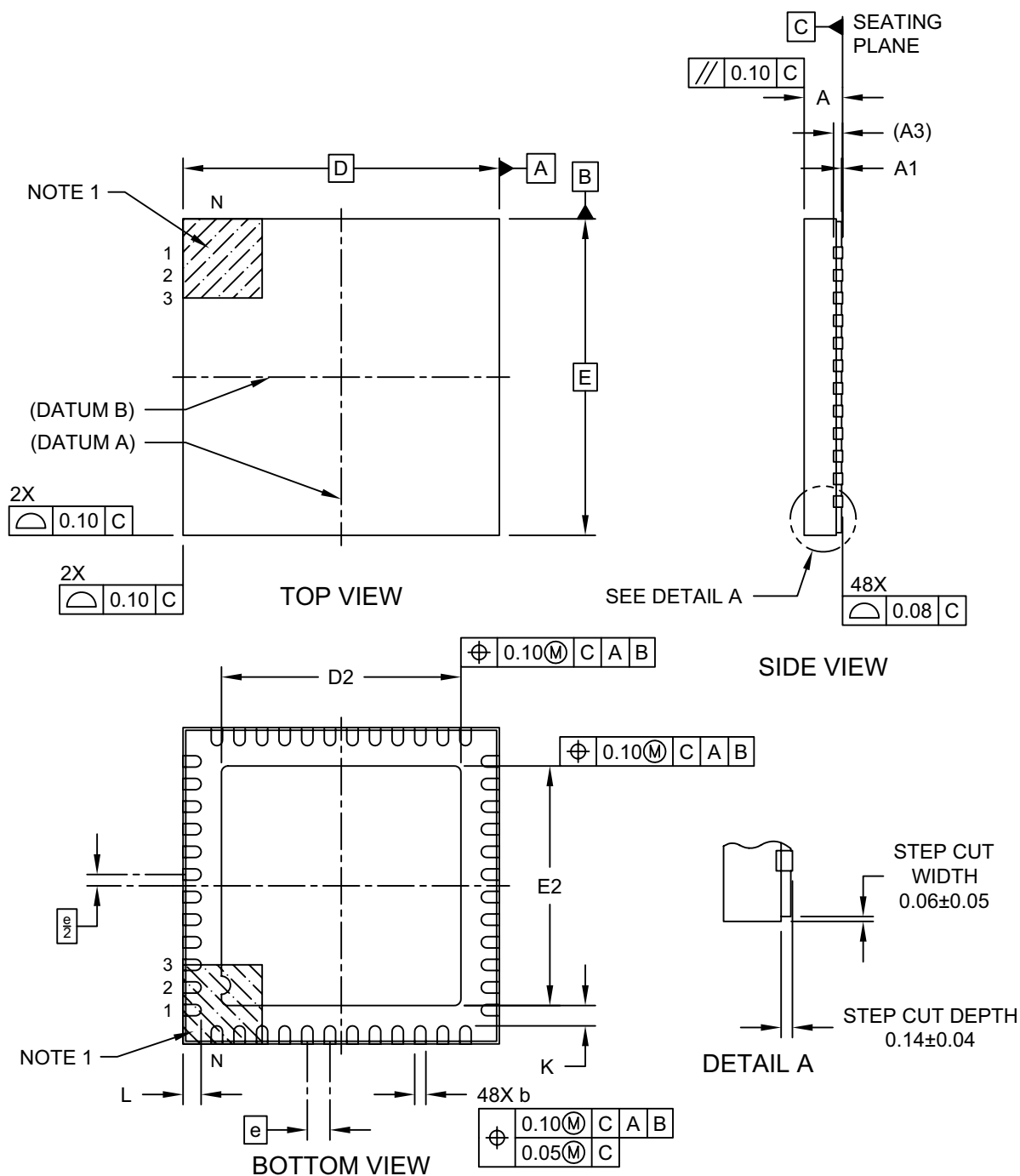
Microchip Technology Drawing C04-C04-27510 Rev A

# OS81212/4/6

FIGURE 16-6: OS81214/6 48-PIN SOLDERABLE FLANK QFN PACKAGE DRAWING (1 OF 3)

## 48-Lead Very Thin Plastic Quad Flat, No Lead Package (KFX) - 7x7 mm Body [VQFN] With Step Cut Wettable Flanks

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



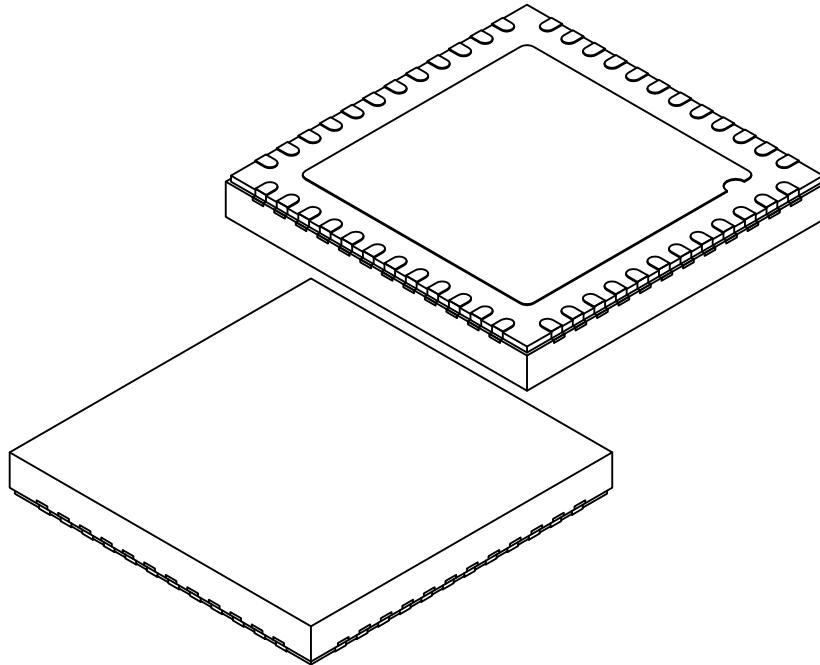
Microchip Technology Drawing C04-0447A Sheet 1 of 2



FIGURE 16-7: OS81214/6 48-PIN SOLDERABLE FLANK QFN PACKAGE DRAWING (2 OF 3)

## 48-Lead Very Thin Plastic Quad Flat, No Lead Package (KFX) - 7x7 mm Body [VQFN] With Step Cut Wettable Flanks

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



|                         |    | MILLIMETERS |      |      |
|-------------------------|----|-------------|------|------|
| Units                   |    |             |      |      |
| Dimension Limits        |    | MIN         | NOM  | MAX  |
| Number of Terminals     | N  | 48          |      |      |
| Pitch                   | e  | 0.50 BSC    |      |      |
| Overall Height          | A  | 0.80        | 0.85 | 0.90 |
| Standoff                | A1 | 0.00        | 0.02 | 0.05 |
| Terminal Thickness      | A3 | 0.20 REF    |      |      |
| Overall Length          | D  | 7.00 BSC    |      |      |
| Exposed Pad Length      | D2 | 5.20        | 5.30 | 5.40 |
| Overall Width           | E  | 7.00 BSC    |      |      |
| Exposed Pad Width       | E2 | 5.20        | 5.30 | 5.40 |
| Terminal Width          | b  | 0.18        | 0.25 | 0.30 |
| Terminal Length         | L  | 0.35        | 0.40 | 0.45 |
| Terminal-to-Exposed-Pad | K  | 0.45 REF    |      |      |

**Notes:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Package is saw singulated
- Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

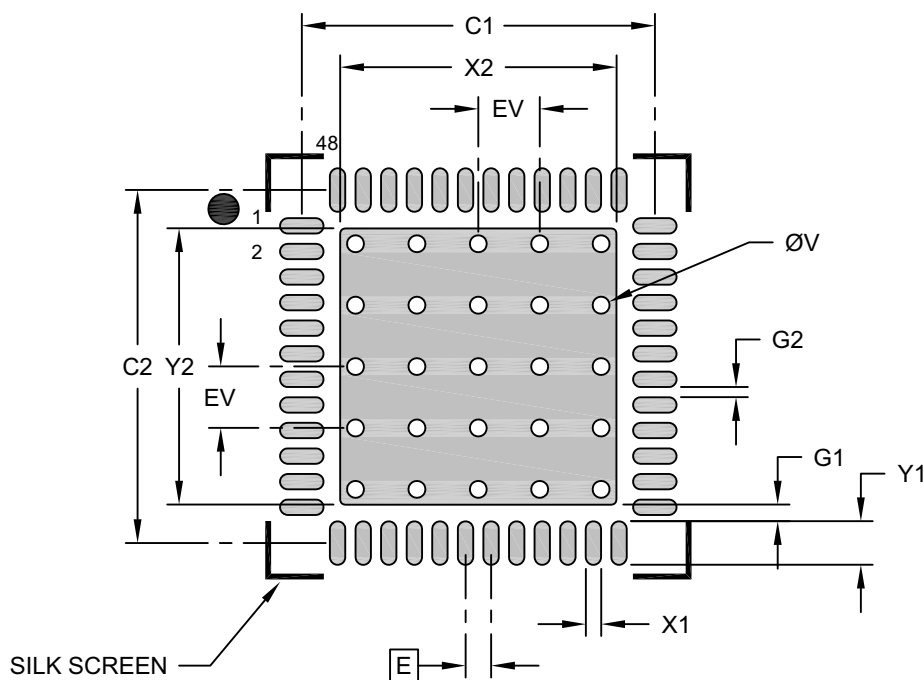
Microchip Technology Drawing C04-0447A Sheet 2 of 2

# OS81212/4/6

FIGURE 16-8: OS81214/6 48-PIN SOLDERABLE FLANK QFN PACKAGE DRAWING (3 OF 3)

## 48-Lead Very Thin Plastic Quad Flat, No Lead Package (KFX) - 7x7 mm Body [VQFN] With Step Cut Wettable Flanks

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



RECOMMENDED LAND PATTERN

|                                  | Units | MILLIMETERS |      |      |
|----------------------------------|-------|-------------|------|------|
|                                  |       | MIN         | NOM  | MAX  |
| Dimension Limits                 |       |             |      |      |
| Contact Pitch                    | E     | 0.50 BSC    |      |      |
| Optional Center Pad Width        | X2    |             |      | 5.40 |
| Optional Center Pad Length       | Y2    |             |      | 5.40 |
| Contact Pad Spacing              | C1    |             | 6.90 |      |
| Contact Pad Spacing              | C2    |             | 6.90 |      |
| Contact Pad Width (X48)          | X1    |             |      | 0.30 |
| Contact Pad Length (X48)         | Y1    |             |      | 0.85 |
| Contact Pad to Center Pad (X48)  | G1    | 0.33        |      |      |
| Contact Pad to Contact Pad (X44) | G2    | 0.20        |      |      |
| Thermal Via Diameter             | V     |             | 0.33 |      |
| Thermal Via Pitch                | EV    |             | 1.20 |      |

**Notes:**

- Dimensioning and tolerancing per ASME Y14.5M  
BSC: Basic Dimension. Theoretically exact value shown without tolerances.
- For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-2447A

## APPENDIX A: FEATURE MATRIX

Some hardware ports are only available with specific INIC devices. A list of available features is provided in [Table A-1](#).

**TABLE A-1: OS81212/4/6 FEATURE MATRIX**

| Device Feature                        | Devices |         |         |
|---------------------------------------|---------|---------|---------|
|                                       | OS81212 | OS81214 | OS81216 |
| Control Port                          | x       | x       | x       |
| MediaLB Port                          | x       |         |         |
| Streaming Port A clocks SCKA and FSYA | x       | x       | x       |
| Streaming Port B clocks SCKB and FSYB | x       |         |         |
| SPI Port                              | x       | x       |         |
| External Power Management             | x       | x       | x       |
| Clock Manager                         | x       | x       | x       |
| RMCK Port                             | x       | x       | x       |
| GPIO Port                             | x       | x       | x       |
| JTAG Port                             | x       | x       | x       |
| bPHY Interface                        | x       | x       | x       |
| All other features                    | x       | x       | x       |

## APPENDIX B: REFERENCES

Documents listed below and referenced within this publication are current as of the release of this publication and may have been reissued with more current information. To obtain information on Microchip automotive documentation please submit a case through [www.microchip.com/support](http://www.microchip.com/support).

All non-Microchip documentation should be retrieved from the applicable website locations listed below. Microchip is not responsible for the update, maintenance or distribution of non-Microchip documentation.

Because the Internet is a constantly changing environment, all Internet links mentioned below and throughout this document are subject to change without notice.

- [1] ISO 21806 (Parts 1-7): 2020 – Road Vehicles – Media Oriented Systems Transport (MOST) Specifications  
International Standards Organization:  
Part 1: General Information and Definitions: [www.iso.org/standard/71833.html](http://www.iso.org/standard/71833.html)  
Part 2: Application Layer: [www.iso.org/standard/71834.html](http://www.iso.org/standard/71834.html)  
Part 3: Application Layer Conformance Test Plan: [www.iso.org/standard/71835.html](http://www.iso.org/standard/71835.html)  
Part 4: Transport Layer and Network Layer: [www.iso.org/standard/71836.html](http://www.iso.org/standard/71836.html)  
Part 5: Transport Layer and Network Layer Test Plan: [www.iso.org/standard/71837.html](http://www.iso.org/standard/71837.html)  
Part 6: Data Link Layer: [www.iso.org/standard/71838.html](http://www.iso.org/standard/71838.html)  
Part 7: Data Link Layer Test Plan: [www.iso.org/standard/71839.html](http://www.iso.org/standard/71839.html)
- [2] ISO 21806 (Parts 14-15): 2021 – Road Vehicles – Media Oriented Systems Transport (MOST) Lean Application Layer Specification  
International Standards Organization:  
Part 14: Lean Application Layer: [www.iso.org/standard/77930.html](http://www.iso.org/standard/77930.html)  
Part 15: Lean Application Layer Conformance Test Plan: [www.iso.org/standard/77931.html](http://www.iso.org/standard/77931.html)
- [3] INICKit Tool  
Microchip. [www.k2l.de](http://www.k2l.de).
- [4] INIC Interface Specification  
DS60001464. Microchip. [www.microchip.com/support](http://www.microchip.com/support).
- [5] MediaLB Specification  
TB0400AN4V2. Microchip. [www.microchip.com/support](http://www.microchip.com/support).
- [6] Port Message Protocol Version 2 User's Guide  
Microchip. [www.microchip.com/support](http://www.microchip.com/support).
- [7] I<sup>2</sup>C-Bus Specification  
UM10204\_3. NXP Semiconductors (formerly a division of Philips). [www.nxp.com](http://www.nxp.com).
- [8] MediaLB Monitor Adapter User's Manual  
Microchip. [www.k2l.de](http://www.k2l.de).
- [9] MediaLB Analyzer User's Manual  
Microchip. [www.k2l.de](http://www.k2l.de).
- [10] SPI Block Guide Version 4.01  
S12SPIV4. NXP Semiconductors (formerly Freescale Semiconductor). [www.nxp.com](http://www.nxp.com).

- [11] IEEE Standard Test Access Port and Boundary-Scan Architecture  
IEEE Std. 1149.1. Institute of Electrical and Electronics Engineers (IEEE). [www.ieee.org](http://www.ieee.org).
- [12] ISO 21806 (Parts 12-13): 2021 – Road Vehicles – Media Oriented Systems Transport (MOST)  
50 Mbit/s Balanced Media Physical Layer Specification  
International Standards Organization:  
Part 12: 50 Mbit/s Balanced Media Physical Layer: [www.iso.org/standard/77927.html](http://www.iso.org/standard/77927.html)  
Part 13: 50 Mbit/s Balanced Media Physical Layer Conformance Test Plan: [www.iso.org/standard/77928.html](http://www.iso.org/standard/77928.html)
- [13] OS8121x Physical Layer Reference Design Application Note  
DS60001452. Microchip. [www.microchip.com/support](http://www.microchip.com/support).
- [14] Device Update Process User's Guide  
DS60001494. Microchip. [www.microchip.com/support](http://www.microchip.com/support).

## APPENDIX C: REVISION HISTORY

### Revision H (DS60001449H March 2023)

- Removal of “Confidential” status

### Revision G (DS60001449G January 2022)

- OS81212 Package change

### Revision F (DS60001449F July 2021)

- Updated BRXN, BRXP Pin Voltage Reference
- Updated Receive Eye Mask Voltage Reference
- Updated Reference to ISO MOST Specifications

### Revision E (DS60001449E March 2020)

- Updated for Release to Production
- Electrical Characteristics
  - Updated max current, power, and network characteristics for B2B hardware

### Revision D (DS60001449D January 2019)

- General
  - Updated pin names to reference the B1A hardware update
  - Introduced OS81216 INIC
- Overview
  - Introduced Centralized Network Diagnostics and Fallback Operation
- MediaLB Port
  - Updated MediaLB 3-pin diagrams to better demonstrate bus topology
- Streaming Port
  - Introduced Streaming Port pulse density modulation (PDM) format
  - Introduced Streaming Port B clock pins for the OS81212
- Electrical Characteristics
  - Introduced maximum MUTE pin delay from event ( $t_{muteon}$ )
  - Updated max current, power, and network characteristics for B1A hardware

### Revision C (DS60001449C August 2017)

- Updated for OS81212/4 A1B

### Revision B (DS60001449B February 2017)

- Corrected power supply connection diagram.

### Revision A (DS60001449A February 2017)

- Initial document release.

## APPENDIX D: LIST OF ACRONYMS

The following is a list of commonly used acronyms found throughout this document.

ADC – Analog to Digital Converter  
AFE – Analog Front-End  
AKE – Authentication and Key Exchange  
API – Application Programming Interface  
AVP – Audio / Video Packetized  
CKE – Content Key Exchange  
DAC – Digital to Analog Converter  
DCI – Driver Control Interface  
DFI – Discrete Frame Isochronous  
DTCP – Digital Transmission Content Protection  
ECU – Electronic Control Unit  
EHC – External Host Controller  
FPH – FIFO Protocol Header  
GPIO – General Purpose I/O  
ICM – INIC Control Message  
INIC – Intelligent Network Interface Controller  
IOC – I/O Companion  
IPC – Inter-Processor Communication  
JTAG – Joint Test Action Group  
MCM – Meta Control Message  
MEP – Meta Ethernet Package  
MDP – Meta Data Packet  
MHP – MOST High Protocol  
MOST – Media Oriented Systems Transport  
PDM – Pulse Density Modulation  
PMHL – Port Message Header Length  
PML – Port Message Length  
PMP – Port Message Protocol  
QFN – Quad Flat No-leads package  
QoS – Quality of Service  
RCM – Remote Control Message  
ICM – Remote Host Controller  
RMCK – Remote Master Clock  
SPI – Serial Peripheral Interface  
STP – Switch To Power  
STP – Shielded Twisted Pair  
TDM – Time Division Multiplexing  
UNICENS – Unified Centralized Network Stack  
UTP – Unshielded Twisted Pair  
VIOC – Video I/O Companion  
VQFN – Very thin Quad Flat No-leads package

## APPENDIX E: LIST OF TABLES

|              |   |     |
|--------------|---|-----|
| TABLE 1-1:   | OS81212/4/6 Pin Allocation Table .....                              | 6   |
| TABLE 3-1:   | Hardware Port Data Transport .....                                  | 15  |
| TABLE 6-1:   | Available MediaLB Bandwidth .....                                   | 30  |
| TABLE 6-2:   | MediaLB Commands & Responses Supported .....                        | 33  |
| TABLE 7-1:   | Streaming Port A Configuration Options .....                        | 35  |
| TABLE 7-2:   | Streaming Port B Configuration Options .....                        | 35  |
| TABLE 8-1:   | Serial Peripheral Interface .....                                   | 40  |
| TABLE 8-2:   | SPI Transaction Opcode .....  | 43  |
| TABLE 8-3:   | SPI Register Summary .....  | 50  |
| TABLE 9-1:   | External Power Management States .....                              | 59  |
| TABLE 12-1:  | OS81212 GPIO Pin Availability .....                                 | 63  |
| TABLE 12-2:  | OS81214 GPIO Pin Availability .....                                 | 63  |
| TABLE 12-3:  | OS81216 GPIO Pin Availability .....                                 | 63  |
| TABLE 14-1:  | Absolute Maximum Ratings .....                                      | 69  |
| TABLE 14-2:  | Operating Conditions .....  | 70  |
| TABLE 14-3:  | DC Characteristics (other than MediaLB Port) .....                  | 71  |
| TABLE 14-4:  | AC Characteristics (other than MediaLB Port) .....                  | 72  |
| TABLE 14-5:  | Reset and Configuration AC Characteristics .....                    | 73  |
| TABLE 14-6:  | Electrical Physical Layer AC Characteristics .....                  | 75  |
| TABLE 14-7:  | Electrical Physical Layer Electrical Characteristics .....          | 77  |
| TABLE 14-8:  | BTXP/BTXN Transmit Eye Mask Boundaries .....                        | 78  |
| TABLE 14-9:  | BRXP/BRXN Receive Eye Mask Boundaries .....                         | 79  |
| TABLE 14-10: | MediaLB Port (3-Pin 256×Fs or 512×Fs) .....                         | 80  |
| TABLE 14-11: | MediaLB Port (3-Pin 1024×Fs) .....                                  | 81  |
| TABLE 14-12: | Streaming Port .....  | 82  |
| TABLE 14-13: | SPI Port (Clock Mode 0) .....                                       | 83  |
| TABLE 14-14: | SPI Port (Clock Mode 1) .....                                       | 84  |
| TABLE 14-15: | SPI Port (Clock Mode 2) .....                                       | 85  |
| TABLE 14-16: | SPI Port (Clock Mode 3) .....                                       | 86  |
| TABLE 14-17: | I <sup>2</sup> C Port (Slave Mode) .....                            | 87  |
| TABLE 14-18: | I <sup>2</sup> C Port (Master Mode) .....                           | 87  |
| TABLE 14-19: | JTAG Port .....   | 89  |
| TABLE 15-1:  | Crystal Oscillator Specifications .....                             | 97  |
| TABLE 16-1:  | OS81212 56-Pin Solderable Flank QFN Thermal Characteristics .....   | 100 |
| TABLE 16-2:  | OS81214/6 48-Pin Solderable Flank QFN Thermal Characteristics ..... | 100 |
| TABLE A-1:   | OS81212/4/6 Feature Matrix .....                                    | 107 |



## APPENDIX F: LIST OF FIGURES

|              |  |    |
|--------------|--|----|
| FIGURE 1-1:  | OS81212 Pin Diagram .....  | 5  |
| FIGURE 1-2:  | OS81214 Pin Diagram .....  | 5  |
| FIGURE 1-3:  | OS81216 Pin Diagram .....  | 6  |
| FIGURE 2-1:  | Pin-equivalent for Analog I/O pin - AI/O .....                                   | 9  |
| FIGURE 2-2:  | Pin-equivalent for Digital Input pin - DIN .....                                 | 9  |
| FIGURE 2-3:  | Pin-equivalent for Digital Output pin - DOUT .....                               | 9  |
| FIGURE 2-4:  | Pin-equivalent for Open-Drain Digital Output pin - DOOUT .....                   | 9  |
| FIGURE 2-5:  | Pin-equivalent for Digital Output pin with High-Z control - DOUTZ .....          | 10 |
| FIGURE 2-6:  | Pin-equivalent for Digital I/O pin - DI/O .....                                  | 10 |
| FIGURE 2-7:  | Pin-equivalent for Digital Input/Open-Drain Output pin - DI/OD .....             | 10 |
| FIGURE 3-1:  | INIC Hardware/System Overview .....  | 12 |
| FIGURE 3-2:  | Network Frame Channels .....   | 13 |
| FIGURE 3-3:  | INIC-IOC DTCP Application Example .....  | 17 |
| FIGURE 3-4:  | INIC-IOC Video Application Example .....   | 18 |
| FIGURE 3-5:  | eCall Example .....  | 19 |
| FIGURE 3-6:  | Simplex Daisy Chain Microphone Network Example .....                             | 19 |
| FIGURE 5-1:  | I <sup>2</sup> C Port Pin Connections (Slave mode) .....                         | 23 |
| FIGURE 5-2:  | I <sup>2</sup> C Port Write Transaction (Slave Mode) .....                       | 24 |
| FIGURE 5-3:  | I <sup>2</sup> C Port Read Transaction (Slave Mode) .....                        | 24 |
| FIGURE 5-4:  | I <sup>2</sup> C Port Pin Connections (Master Mode) .....                        | 26 |
| FIGURE 5-5:  | I <sup>2</sup> C Port Write Transaction (Master Mode) .....                      | 26 |
| FIGURE 5-6:  | I <sup>2</sup> C Port Read Transaction (Master Mode) .....                       | 27 |
| FIGURE 5-7:  | I <sup>2</sup> C Port Repeated Writes and Reads (Master Mode) .....              | 27 |
| FIGURE 6-1:  | MediaLB 3-pin Interface Connection Diagram 2-Node System .....                   | 29 |
| FIGURE 6-2:  | MediaLB 3-pin Interface Connection Diagram 3-Node System .....                   | 29 |
| FIGURE 6-3:  | MediaLB 3-pin Data Structure .....   | 30 |
| FIGURE 6-4:  | MediaLB 3-pin Interface Example (at 256×Fs) .....                                | 32 |
| FIGURE 6-5:  | MediaLB 3-Pin Mode to Parallel Interface .....                                   | 34 |
| FIGURE 7-1:  | Streaming Port Pin Configurations .....  | 35 |
| FIGURE 7-2:  | Left-Justified Streaming Data Format .....                                       | 36 |
| FIGURE 7-3:  | Right-Justified Streaming Data Format .....                                      | 37 |
| FIGURE 7-4:  | Delayed-Bit Data Format (I <sup>2</sup> S Compatible) .....                      | 37 |
| FIGURE 7-5:  | Sequential Data Format .....   | 38 |
| FIGURE 7-6:  | Time-Division Multiplex Format .....   | 39 |
| FIGURE 7-7:  | Time-Division Multiplex Format (Delayed-Bit Only) .....                          | 39 |
| FIGURE 7-8:  | Time-Division Multiplex Format (Delayed-bit - I <sup>2</sup> S Compatible) ..... | 39 |
| FIGURE 8-1:  | SPI Port Pin Connections .....   | 40 |
| FIGURE 8-2:  | SPI Clock Mode 0 .....   | 40 |
| FIGURE 8-3:  | SPI Clock Mode 1 .....   | 41 |
| FIGURE 8-4:  | SPI Clock Mode 2 .....   | 41 |
| FIGURE 8-5:  | SPI Clock Mode 3 .....   | 41 |
| FIGURE 8-6:  | SPI Transaction .....  | 42 |
| FIGURE 8-7:  | SPI Register Write Transaction .....   | 44 |
| FIGURE 8-8:  | SPI Register Read Transaction .....  | 44 |
| FIGURE 8-9:  | SPI Packet Write Transaction .....   | 45 |
| FIGURE 8-10: | SPI Packet Read Transaction .....  | 46 |
| FIGURE 8-11: | DCI Command Packet Write Transaction .....                                       | 47 |
| FIGURE 8-12: | DCI Response Packet Read Transaction .....                                       | 47 |
| FIGURE 8-13: | SINT Interrupt Pin Logic .....   | 48 |
| FIGURE 9-1:  | Typical Power Management Circuitry .....   | 60 |
| FIGURE 10-1: | Crystal Oscillator Input .....   | 61 |
| FIGURE 11-1: | RMCK Port with Synchronous Output Clock - Example .....                          | 62 |
| FIGURE 14-1: | Power-Up and Reset Timing .....  | 74 |

# OS81212/4/6

---

|               |  |     |
|---------------|--|-----|
| FIGURE 14-2:  | Balanced Media Physical Layer Reset Timing.....                      | 75  |
| FIGURE 14-3:  | Balanced Media Physical Layer Startup and shutdown Timing .....      | 76  |
| FIGURE 14-4:  | Balanced media Physical Layer STATUS Timing.....                     | 76  |
| FIGURE 14-5:  | Transmitter Test Conditions.....                                     | 77  |
| FIGURE 14-6:  | BTXP/BTXN Transmit Eye Mask Pattern.....                             | 78  |
| FIGURE 14-7:  | BRXP/BRXN Receive Eye Mask Pattern.....                              | 79  |
| FIGURE 14-8:  | MediaLB 3-pin Timing .....   | 81  |
| FIGURE 14-9:  | Streaming Port Timing .....  | 82  |
| FIGURE 14-10: | SPI Timing (Clock Mode 0).....                                       | 83  |
| FIGURE 14-11: | SPI Timing (Clock Mode 1).....                                       | 84  |
| FIGURE 14-12: | SPI Timing (Clock Mode 2).....                                       | 85  |
| FIGURE 14-13: | SPI Timing (Clock Mode 3).....                                       | 86  |
| FIGURE 14-14: | I <sup>2</sup> C Port Timing .....                                   | 88  |
| FIGURE 14-15: | JTAG Timing .....  | 89  |
| FIGURE 15-1:  | Power Supply Connection Diagram .....                                | 90  |
| FIGURE 15-2:  | Reset Architecture .....   | 91  |
| FIGURE 15-3:  | bPHY Interface - Block Diagram .....                                 | 92  |
| FIGURE 15-4:  | Connection Diagram for I <sup>2</sup> C Port Interface .....         | 94  |
| FIGURE 15-5:  | Configuration/Debug Header .....                                     | 95  |
| FIGURE 15-6:  | Recommended Termination for Streaming Port .....                     | 95  |
| FIGURE 15-7:  | Recommended Termination for SPI Port .....                           | 95  |
| FIGURE 15-8:  | Recommended Termination for MediaLB Port.....                        | 96  |
| FIGURE 15-9:  | Recommended Termination for I <sup>2</sup> C and JTAG Ports.....     | 96  |
| FIGURE 16-1:  | OS81212 Top Marking.....   | 99  |
| FIGURE 16-2:  | OS81214 Top Marking.....   | 99  |
| FIGURE 16-3:  | OS81212 56-Pin Solderable Flank QFN Package Drawing (1 of 3) .....   | 101 |
| FIGURE 16-4:  | OS81212 56-Pin Solderable Flank QFN Package Drawing (2 of 3) .....   | 102 |
| FIGURE 16-5:  | OS81212 56-Pin Solderable Flank QFN Package Drawing (3 of 3) .....   | 103 |
| FIGURE 16-6:  | OS81214/6 48-Pin Solderable Flank QFN Package Drawing (1 of 3) ..... | 104 |
| FIGURE 16-7:  | OS81214/6 48-Pin Solderable Flank QFN Package Drawing (2 of 3) ..... | 105 |
| FIGURE 16-8:  | OS81214/6 48-Pin Solderable Flank QFN Package Drawing (3 of 3) ..... | 106 |

## THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at [www.microchip.com](http://www.microchip.com). This website is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the website contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

## CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip website at [www.microchip.com](http://www.microchip.com). Under "Support", click on "Customer Change Notification" and follow the registration instructions.

## CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

**Technical support is available through the website at:** <http://microchip.com/support>

## AUTOMOTIVE PRODUCT SUPPORT

Customers using Microchip automotive products can receive additional help through the [Automotive Support Contacts](#) page of the Microchip web site.

# OS81212/4/6

## PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

| <u>PART NO.</u>                     | <u>X</u> | <u>X</u>     | <u>[X]</u>         | - | <u>rrr</u>  | - | <u>vvvvv</u>           | - | <u>[ss]</u>              | - | <u>[xxx]</u>         |
|-------------------------------------|----------|--------------|--------------------|---|---|---|------------------------|---|--------------------------|---|----------------------|
| Device                              | Grade    | Package Type | Tape and Reel Flag |   | Product Revision Code   |   | Firmware Revision Code |   | Firmware Service Release |   | Special Feature Code |
| Device                              |          | OS81212      |                    | = | 50 Mbit/s Automotive Intelligent Network Interface Controller with MediaLB      |   |                        |   |                          |   |                      |
|                                     |          | OS81214      |                    | = | 50 Mbit/s Automotive Intelligent Network Interface Controller (no MediaLB)      |   |                        |   |                          |   |                      |
|                                     |          | OS81216      |                    | = | 50 Mbit/s Automotive Intelligent Network Interface Controller (no MediaLB, SPI) |   |                        |   |                          |   |                      |
| Grade                               | A        |              |                    | = | All Features  |   |                        |   |                          |   |                      |
| Package Type                        | F        |              |                    | = | QFN with solderable terminals   |   |                        |   |                          |   |                      |
| Tape and Reel Flag (optional)       | Blank    |              |                    | = | Standard Packaging (Tube/Tray)  |   |                        |   |                          |   |                      |
|                                     | R        |              |                    | = | Tape and Reel   |   |                        |   |                          |   |                      |
| Product Revision Code               | rrr      |              |                    | = | 3 character code specifying product revision                                    |   |                        |   |                          |   |                      |
| Firmware Revision Code              | vvvvv    |              |                    | = | 6 character code specifying firmware revision                                   |   |                        |   |                          |   |                      |
| Firmware Service Release (optional) | ss       |              |                    | = | 2 character code specifying service release                                     |   |                        |   |                          |   |                      |
| Special Feature Code (optional)     | xxx      |              |                    | = | 3 character code for special requirements                                       |   |                        |   |                          |   |                      |

### Examples:

- a) OS81212AF-rrr-vvvvvv-xxx  
56-pin solderable terminal QFN package
- b) OS81212AFR-rrr-vvvvvv-xxx  
56-pin solderable terminal QFN package, Tape and Reel
- c) OS81214AF-rrr-vvvvvv-xxx  
48-pin solderable terminal QFN package
- d) OS81214AFR-rrr-vvvvvv-xxx  
48-pin solderable terminal QFN package, Tape and Reel
- e) OS81216AF-rrr-vvvvvv-xxx  
48-pin solderable terminal QFN package
- f) OS81216AFR-rrr-vvvvvv-xxx  
48-pin solderable terminal QFN package, Tape and Reel

---

**Note the following details of the code protection feature on Microchip products:**

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "unbreakable" Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products

---

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at <https://www.microchip.com/en-us/support/design-help/client-support-services>.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

For information regarding Microchip's Quality Management Systems, please visit [www.microchip.com/quality](http://www.microchip.com/quality).

**Trademarks**

The Microchip name and logo, the Microchip logo, Adaptec, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, CryptoMemory, CryptoRF, dsPIC, flexPWR, HELDO, IGLoo, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, Flashtec, Hyper Speed Control, HyperLight Load, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, TrueTime, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, Clockstudio, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, GridTime, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, IntelliMOS, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, KoD, maxCrypto, maxView, memBrain, MINDI, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-ICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SmartHLS, SMART-I.S., storClad, SQL, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, Trusted Time, TSHARC, USBCheck, VariSense, VectorBlox, VeriPHY, ViewSpan, WiperLock, XpressConnect, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2023, Microchip Technology Incorporated,

All Rights Reserved.

ISBN: 978-1-5224-9630-4



## Worldwide Sales and Service

### AMERICAS

**Corporate Office**  
2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 480-792-7200  
Fax: 480-792-7277  
Technical Support:  
<http://www.microchip.com/support>  
Web Address:  
[www.microchip.com](http://www.microchip.com)

**Atlanta**  
Duluth, GA  
Tel: 678-957-9614  
Fax: 678-957-1455

**Austin, TX**  
Tel: 512-257-3370

**Boston**  
Westborough, MA  
Tel: 774-760-0087  
Fax: 774-760-0088

**Chicago**  
Itasca, IL  
Tel: 630-285-0071  
Fax: 630-285-0075

**Dallas**  
Addison, TX  
Tel: 972-818-7423  
Fax: 972-818-2924

**Detroit**  
Novi, MI  
Tel: 248-848-4000

**Houston, TX**  
Tel: 281-894-5983

**Indianapolis**  
Noblesville, IN  
Tel: 317-773-8323  
Fax: 317-773-5453  
Tel: 317-536-2380

**Los Angeles**  
Mission Viejo, CA  
Tel: 949-462-9523  
Fax: 949-462-9608  
Tel: 951-273-7800

**Raleigh, NC**  
Tel: 919-844-7510

**New York, NY**  
Tel: 631-435-6000

**San Jose, CA**  
Tel: 408-735-9110  
Tel: 408-436-4270

**Canada - Toronto**  
Tel: 905-695-1980  
Fax: 905-695-2078

### ASIA/PACIFIC

**Australia - Sydney**  
Tel: 61-2-9868-6733

**China - Beijing**  
Tel: 86-10-8569-7000

**China - Chengdu**  
Tel: 86-28-8665-5511

**China - Chongqing**  
Tel: 86-23-8980-9588

**China - Dongguan**  
Tel: 86-769-8702-9880

**China - Guangzhou**  
Tel: 86-20-8755-8029

**China - Hangzhou**  
Tel: 86-571-8792-8115

**China - Hong Kong SAR**  
Tel: 852-2943-5100

**China - Nanjing**  
Tel: 86-25-8473-2460

**China - Qingdao**  
Tel: 86-532-8502-7355

**China - Shanghai**  
Tel: 86-21-3326-8000

**China - Shenyang**  
Tel: 86-24-2334-2829

**China - Shenzhen**  
Tel: 86-755-8864-2200

**China - Suzhou**  
Tel: 86-186-6233-1526

**China - Wuhan**  
Tel: 86-27-5980-5300

**China - Xian**  
Tel: 86-29-8833-7252

**China - Xiamen**  
Tel: 86-592-2388138

**China - Zhuhai**  
Tel: 86-756-3210040

### ASIA/PACIFIC

**India - Bangalore**  
Tel: 91-80-3090-4444

**India - New Delhi**  
Tel: 91-11-4160-8631

**India - Pune**  
Tel: 91-20-4121-0141

**Japan - Osaka**  
Tel: 81-6-6152-7160

**Japan - Tokyo**  
Tel: 81-3-6880-3770

**Korea - Daegu**  
Tel: 82-53-744-4301

**Korea - Seoul**  
Tel: 82-2-554-7200

**Malaysia - Kuala Lumpur**  
Tel: 60-3-7651-7906

**Malaysia - Penang**  
Tel: 60-4-227-8870

**Philippines - Manila**  
Tel: 63-2-634-9065

**Singapore**  
Tel: 65-6334-8870

**Taiwan - Hsin Chu**  
Tel: 886-3-577-8366

**Taiwan - Kaohsiung**  
Tel: 886-7-213-7830

**Taiwan - Taipei**  
Tel: 886-2-2508-8600

**Thailand - Bangkok**  
Tel: 66-2-694-1351

**Vietnam - Ho Chi Minh**  
Tel: 84-28-5448-2100

### EUROPE

**Austria - Wels**  
Tel: 43-7242-2244-39  
Fax: 43-7242-2244-393

**Denmark - Copenhagen**  
Tel: 45-4485-5910  
Fax: 45-4485-2829

**Finland - Espoo**  
Tel: 358-9-4520-820

**France - Paris**  
Tel: 33-1-69-53-63-20  
Fax: 33-1-69-30-90-79

**Germany - Garching**  
Tel: 49-8931-9700

**Germany - Haan**  
Tel: 49-2129-3766400

**Germany - Heilbronn**  
Tel: 49-7131-72400

**Germany - Karlsruhe**  
Tel: 49-721-625370

**Germany - Munich**  
Tel: 49-89-627-144-0  
Fax: 49-89-627-144-44

**Germany - Rosenheim**  
Tel: 49-8031-354-560

**Israel - Ra'anana**  
Tel: 972-9-744-7705

**Italy - Milan**  
Tel: 39-0331-742611  
Fax: 39-0331-466781

**Italy - Padova**  
Tel: 39-049-7625286

**Netherlands - Drunen**  
Tel: 31-416-690399  
Fax: 31-416-690340

**Norway - Trondheim**  
Tel: 47-7288-4388

**Poland - Warsaw**  
Tel: 48-22-3325737

**Romania - Bucharest**  
Tel: 40-21-407-87-50

**Spain - Madrid**  
Tel: 34-91-708-08-90  
Fax: 34-91-708-08-91

**Sweden - Gothenberg**  
Tel: 46-31-704-60-40

**Sweden - Stockholm**  
Tel: 46-8-5090-4654

**UK - Wokingham**  
Tel: 44-118-921-5800  
Fax: 44-118-921-5820