**Datasheet**

# IMPINJ MONZA® X-8K DURA

## TAG CHIP DATASHEET

# OVERVIEW

Monza® X-8K Dura is a UHF Gen2 RFID IC product with 8192 bits of user Non-Volatile Memory (NVM) and an I2C interface.

As an I2C device Monza X-8K Dura operates as a standard I2C EEPROM. The contents of this EEPROM can also be accessed wirelessly via the UHF Gen2 RFID Protocol.

## Features

- EPCglobal and ISO 18000-63 compliant, Gen2V2 compliant.

- 8192 bits of user NVM

- 16 One Time Programmable (OTP) blocks (3583 bits) via BlockPermalock feature supported by both I2C and EPC Gen2 interface

- QT for read control and data privacy on RF link

- I2C slave interface with NVM read and write and four user configurable I2C slave addresses

- −19.1 dBm typical read sensitivity when using a single RF antenna port & dipole tag attached, -26.1 dBm with DC input & dipole tag attached.

- −21.6dBm typical read sensitivity when using dual RF antenna ports and dipole tag attached.

- −14.1 dBm typical write sensitivity when using a single RF antenna port and dipole tag attached.

- I2C control of RF access

- Write wakeup mode, which allows Monza X-8K Dura to wake up a device.

- FastID™ inventory mode, a Gen2 compliant, patent-pending method for EPC+TID based inventory that is 2-3 times faster than previous methods

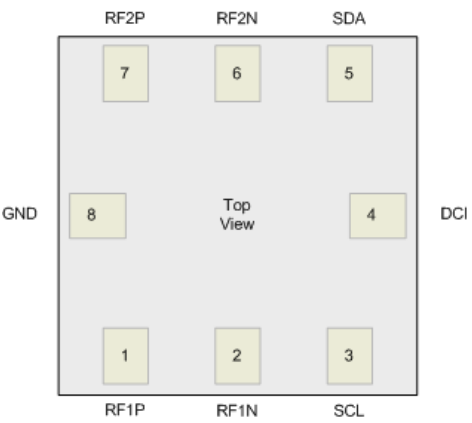## Table 1: Impinj Monza X-8K Dura Input Port Overview

| NAME | DESCRIPTION | CHARACTERISTIC | CHIP VIEW |
|------|-------------|----------------|-----------|
| RF1_P | Differential RF Input Port 1 | 1.6kΩ, 1pF<br>−17 dBm single-port sensitivity<br>−19.5 dBm True3D sensitivity |  |
| RF1_N | | | |
| RF2_P | Differential RF Input Port 2 | | |
| RF2_N | | | |
| DCI | DC Input | 1.6–3.6V | |
| SCL | I2C Clock Input | VIH/L=70% / 30% DCI<br>IOL=6mA @ 0.4V | |
| SDA | I2C Data Input | | |
| GND | Ground | | |

# TABLE OF CONTENTS

# 1  INTRODUCTION

## 1.1  Scope

This datasheet defines the physical and logical specifications for Gen 2-compliant Monza X-8K Dura tag chip, a reader-talks-first, radio frequency identification (RFID) component operating in the UHF frequency range.

## 1.2  Reference Documents

The following reference documents were used to compile this datasheet:

- EPC™ Radio Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz – 960 MHz (Gen 2 Specification)[1].
  - The conventions used in the Gen 2 Specification (normative references, terms and definitions, symbols, abbreviated terms, and notation) were adopted in the drafting of this Monza X-8K Dura Tag Chip Datasheet. Users of this datasheet should familiarize themselves with the Gen 2 Specification.
- EPC™ Tag Data Standards Specification[2].
- EPCglobal Interoperability Test System for EPC Compliant Class-1 Generation-2 UHF RFID[3].
  - Monza X-8K Dura tag chips are compliant with this Gen 2 interoperability standard.
- I2C-bus Specification Rev. 03, June 19 2007, NXP Doc UM10204[4]

You may consult these documents for more information about compliance standards and specifications.

# 2  FUNCTIONAL  DESCRIPTION

Monza X-8K Dura enables users to communicate wirelessly with the processor inside electronic devices using standard Gen 2 RFID readers, which unlocks many new benefits for consumer electronics manufacturers, retailers and end users. Monza X-8K Dura connects to the processor of an electronic device through a standard I2C bus. This enables the processor to read and write the Monza X chip memory with information that is accessible to UHF Gen 2 RFID readers even when the electronic device is powered off. By enabling electronic devices to communicate with RFID readers, Monza X chips deliver a wide range of extended capabilities such as theft deterrence in the supply chain and device configuration/upgrades at point of sale and beyond. By default, applying voltage to the DCI pin will disable RF communications. This can be reconfigured by the user (see Gen2/I2C Arbitration).

**Figure 1: Monza X-8K Dura – Electronic Diagram**
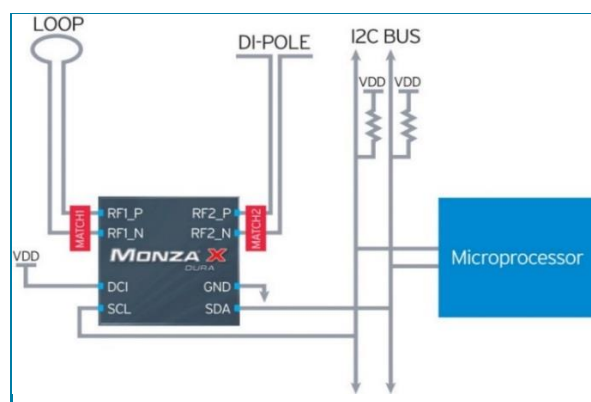
## 2.1   Reader Communications (Gen2/RF Commands)

A reader communicates with Monza X-8K Dura using standard Gen2 RFID commands. Please see the EPCglobal Class-1 Generation-2 UHF RFID Air-Interface Protocol V1.2.0 for details.

The Gen 2 memory map is shown in Table 2. Fields in blue text are read only from a Gen2 reader.  Reserved memory bank words 4-10 are read only.

**Table 2: Gen2 Interface Memory Map**

| GEN2 BANK NAME | I2C BIT ADDR | BIT ADDRESS | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| User (11$_2$) | 1FF0$_h$-1FFF$_h$ | USER [ 15 : 0 ] | | | | | | | | | | | | | | | |
| | … | … | | | | | | | | | | | | | | | |
| | 00$_h$-0F$_h$ | USER [ 8191 : 8176 ] | | | | | | | | | | | | | | | |
| TID (10$_2$) | B0$_h$-BF$_h$ | EPC_PUBLIC [ 15 : 0 ] | | | | | | | | | | | | | | | |
| | … | … | | | | | | | | | | | | | | | |
| | 60$_h$-6F$_h$ | EPC_PUBLIC [ 95 : 80 ] | | | | | | | | | | | | | | | |
| | 50$_h$-5F$_h$ | TID_SERIAL [ 15 : 0 ] | | | | | | | | | | | | | | | |
| | 40$_h$-4F$_h$ | TID_SERIAL [ 31 : 16 ] | | | | | | | | | | | | | | | |
| | 30$_h$-3F$_h$ | TID_SERIAL [ 47 : 32 ] | | | | | | | | | | | | | | | |
| | 20$_h$-2F$_h$ | TDTS [ 15 : 0 ] = 0x2000 | | | | | | | | | | | | | | | |
| | 10$_h$-1F$_h$ | TID_DESIGNER [ 3 : 0 ] = 0001$_2$ | | | | TID_MODEL [11 : 0 ] = 150$_h$ | | | | | | | | | | | |
| | 00$_h$-0F$_h$ | CLASS_ID [ 7 : 0 ] = 11100010$_2$ | | | | | | | | XTID = 1 | | TID_DESIGNER [ 10 : 4 ] = 0000000$_2$ (Gen2 mask designer is address 08$_h$ to 13$_h$) | | | | | | |
| EPC (01$_2$) | 90h-9Fh | EPC_PRIVATE [ 15 : 0 ] | | | | | | | | | | | | | | | |
| | … | … | | | | | | | | | | | | | | | |
| | 20h-2Fh | EPC_PRIVATE [ 127 : 112 ] | | | | | | | | | | | | | | | |
| | 10h-1Fh | EPC_LENGTH [ 4 : 0 ] | | | | | UMI | XI = 0 | NSI [ 8 : 0 ] (Numbering System Identifier, default 000000000$_2$) | | | | | | | | |
| | 00h-0Fh | EPC_CRC [ 15 : 0 ] | | | | | | | | | | | | | | | |
| Reserved (00$_2$) | A0$_h$-AF$_h$ | RFU = 0 | | | | | | | | | | WWU = 0 | BPL_EN = 0 | QT_SR = 0 | QT_MEM=0 | DCI_RF_EN = 0 | RF2_DIS = 0 | RF1_DIS = 0 |
| | 90$_h$-9F$_h$ | BLOCK_PERMALOCK [ 0 : 15 ] | | | | | | | | | | | | | | | |
| | 80$_h$-8F$_h$ | RESERVED | | | | | | | | | | | | | | | |
| | 70$_h$-7F$_h$ | RESERVED | | | | | | | | | | | | | | | |
| | 60$_h$-6F$_h$ | RESERVED | | | | | | | | | | | | | | | |
| | 50$_h$-5F$_h$ | RESERVED | | | | | | | | | | | | | | | |
| | 40$_h$-4F$_h$ | LOCK_KILL [ 1 : 0 ] | | LOCK_ACCESS [ 1 : 0 ] | | LOCK_EPC [ 1 : 0 ] | | LOCK_USER [ 1 : 0 ] | | LOCK_DA | RFU = 0 | | | | | KILL | I2C_ADDR [ 1 : 0 ] = 11$_2$ |
| | 30$_h$-3F$_h$ | ACCESS_PASSWORD [ 15 : 0 ] | | | | | | | | | | | | | | | |
| | 20$_h$-2F$_h$ | ACCESS_PASSWORD [ 31 : 16 ] | | | | | | | | | | | | | | | |
| | 10$_h$-1F$_h$ | KILL_PASSWORD [ 15 : 0 ] | | | | | | | | | | | | | | | |
| | 00$_h$-0F$_h$ | KILL_PASSWORD [ 31 : 16 ] | | | | | | | | | | | | | | | |

*Read-Only memory is highlighted in blue. See legend below for memory descriptions.*

- "**LOCK_DA**" = Permalock bit for I2C device address
- "**SKU**" = Shop Keeping Units (Used to indicate die type)
- "**DCI_RF_EN**" = Enables RF when DCI is present

- **"QT_MEM"** = Selects memory map profile
  - 0: Tag uses private memory map
  - 1: Tag uses public memory map
- **"I2C_ADDR"** = NVM configurable bits of I2C device address $1101XX0_2$, where $XX=11_2$ by default
- **"WWU"** = Write Wake Up
- **"QT_SR"** = Short Range in open and secured states
- **"RF_DIS [ 1: 0 ]"** = RF disable selectable by port

## 2.2   Advanced Monza Inventory Features

Monza tag chips support two unique, patent-pending features designed to boost inventory performance for traditional EPC and TID-based applications:

- TagFocus™ mode minimizes redundant reads of strong tags, allowing the reader to focus on weak tags that are typically the last to be found. Using TagFocus™, readers can suppress previously read tags by indefinitely refreshing their S1 B state.
- FastID™ mode makes TID-based applications such as authentication practical by boosting TID-based inventory speeds by 2 to 3 times. Readers can inventory both the EPC and the TID without having to perform an access command. Setting the EPC word length to zero enables TID-only serialization.

Monza X-8K Dura tag chips will stay in the TagFocus or FastID states once set, until the tags lose power from RF and from DC input, also known as Battery Assisted Passive (BAP) mode.

## 2.3   Support for Optional Gen 2 Commands

Following optional Gen 2 commands are supported:

**Table 3: Optional Gen 2 Commands**

| COMMAND | CODE | LENGTH | DETAILS |
|---|---|---|---|
| **Access** | 11000110 | 56 | |
| **BlockWrite** | 11000111 | >57 | • Accepts valid one-word commands<br>• Accepts valid two-word commands if pointer is an even value<br>• Returns error code (000000002) if it receives a valid two-word command with an odd value pointer<br>• Returns error code (000000002) if it receives a command for more than two words<br>• Does not respond to block write commands of zero words |
| **BlockPermalock** | 11001001 | >66 | • Sixteen blocks<br>• Four, 512 bits in size<br>• Twelve 128 bits in size<br>• Command can be disabled through I2C |

## 2.4 I2C Interface (SDA, SCL, DCI Pins)

I2C is a standard two-wire interface (clock and data) that supports multiple addressable chips on a bus. Monza X-8K Dura only supports slave capability. Monza X-8K Dura's I2C features are compatible with the industry-standard I2C bus. Specifically, Monza X-8K Dura is compatible with I2C specification (I2C Rev 0.03, June 19 2007, NXP Doc UM10204). Monza X-8K Dura implements the following I2C capabilities:

- I2C slave
- I2C Start Condition
- I2C Repeated Start Condition
- I2C Stop Condition
- I2C Acknowledge
- I2C 7-bit slave address with two NVM programmable bits ($1101XX0_2$, where $XX=11_2$ by default)
- Fast mode transfer rates of 0-400kbits/second
- The DCI voltage provides I2C bus VOH/VOL reference and power.

When an I2C master addresses Monza X-8K Dura it must format its write transactions as described here. In addition to the I2C device address Monza X-8K Dura has a 16-bit, two byte, memory address that a master writes on every write transaction. The memory address specifies which memory byte the master is addressing. Only the lower 11 bits of the memory address are used. The master should set the upper bits of the memory address A15-A11 to zero.

The memory address stored in Monza X-8K Dura is only updated explicitly during a write transaction (R/W == 0). A master only writes a memory address, and future read transactions use the previously written address. A diagram of a transaction that writes the memory address is shown in Figure 2. All bit positions are explicitly shown so the boundary between the I2C device address and the Monza X-8K Dura memory address is clear. Subsequent diagrams do not explicitly show these address bits.

### Figure 2: Addressing the Device and Setting the Memory Address
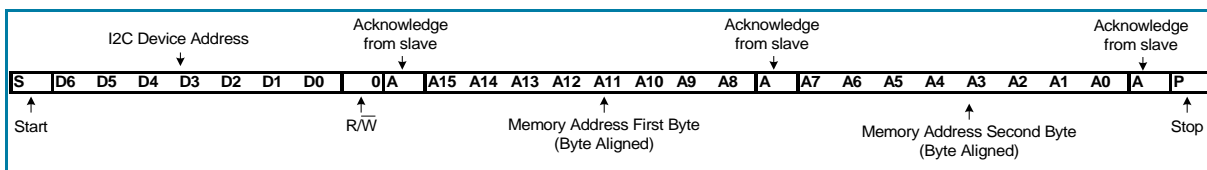


When performing an NVM write a master transmits data after the memory address. Monza X-8K Dura's NVM is organized as 16-bit words. Writes must align on word boundaries. The NVM allows one- or two-word writes (equivalent to two- or four-byte writes). When executing a one-word write Monza X-8K Dura ignores the LSB (A0) of the memory address. When executing a two-word write Monza X-8K Dura ignores the two LSBs (A1, A0) of the memory address. If the write transaction is valid then Monza X-8K Dura begins the NVM write after receiving a stop from the I2C master. Monza X-8K Dura will not respond to subsequent I2C transactions for the duration of the NVM write operation. The write time for one- and two-word write operations is the same. A one-word NVM write transaction is shown in Figure 3.

Monza X-8K Dura may observe several types of invalid NVM-write transactions. If a master sends one or three data bytes then Monza X-8K Dura will not perform the write (recall that Monza X-8K Dura writes 16-bit words). If a master sends more than two words then Monza X-8K Dura will not perform the write. Monza X-8K Dura also checks the memory address and will not perform a write if the address is invalid (but note that Monza X-8K Dura updates its memory address even if the address is invalid).

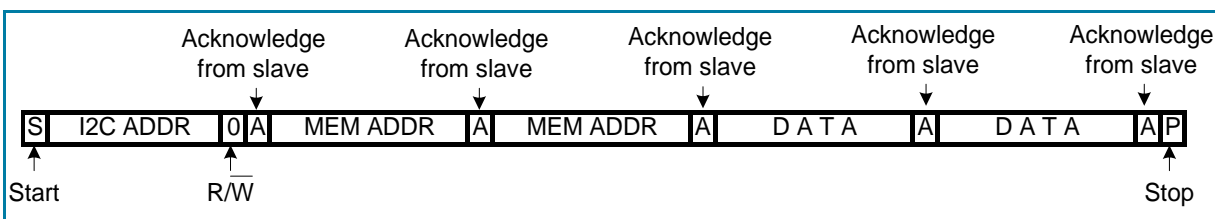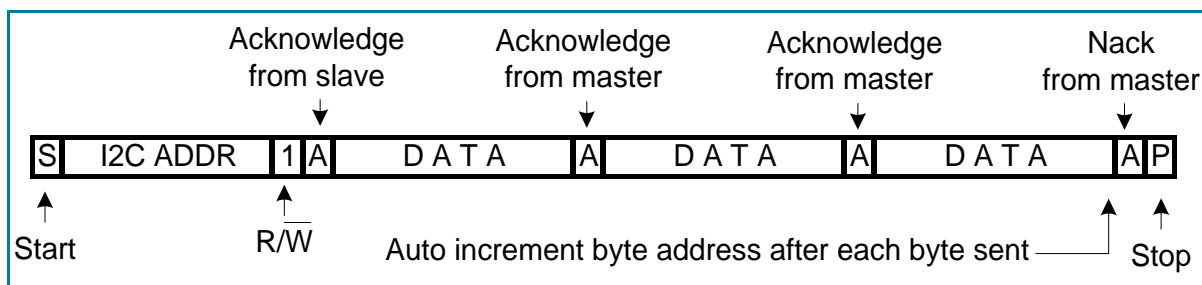**Figure 3: One-Word Monza X-8K Dura Write Transaction**



Figure 4 shows a read transaction. The read starts from the stored address. Monza X-8K Dura increments the address as it sends each data byte.

**Figure 4: Monza X-8K Dura Read Transaction**



Reads start from the stored address and continue to the end of memory, at which point Monza X-8K Dura will cease exchanging data over I2C. Monza X-8K Dura will send all 1's if the master continues to read beyond the end of the memory. To read from a new location the master must send a new address. The master may halt the read at a byte boundary and later initiate a new read transaction starting from that byte. For completeness the combined write transaction then read transaction is shown in Figure 5.

**Figure 5: Write Transaction to Set Address Followed by Repeated Start and Read Transaction**
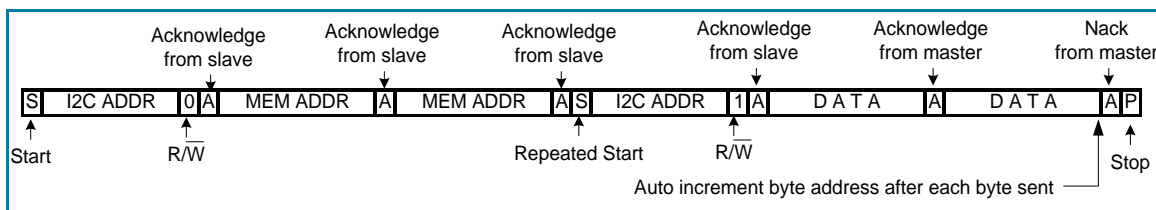


Monza X-8K Dura ignores all Gen2 Lock, Kill permissions when reading / writing over I2C. The I2C port has read access to the entire NVM. The I2C port has write access to most, but not all, of the NVM. Monza X-8K Dura precludes a master from writing its manufacturing calibration fields (shown as Reserved in the I2C memory map of Table 4); these locations are read-only.

## 2.5   I2C Memory Map

Gen 2 and I2C have different views on how a memory map is organized. In I2C everything is done according to bytes. One uses byte addressing, byte writing, and byte reading. In Gen2 things are done in terms of bits or 16-bit words.

Monza X-8K Dura is a hybrid of these two approaches. It forces I2C to do one word or two word writes, but allows for byte wise reading and addressing. When reading via I2C the first bit read is always bit seven within the byte. The next byte read is at the next higher I2C byte address. The I2C memory map in byte wise format is shown in Table 4. An additional memory map that shows bit addressing from I2C in a word wise format is shown in Table 5.

### Table 4: I2C Interface Memory Map in a Byte Wise Format

| GEN2 BANK NAME | I2C BYTE ADDR (DEC) | BIT ADDRESS IN BYTE | | | | | | | | "I2C PERM" |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| USER ($11_2$) | 1087 | USER | | | | | | | | R/W |
| | 1086 | USER | | | | | | | | R/W |
| | . . . | USER | | | | | | | | R/W |
| | 65 | USER | | | | | | | | R/W |
| | 64 | USER | | | | | | | | R/W |
| TID ($10_2$) | 63 | EPC_PUBLIC | | | | | | | | R/W |
| | 62 | EPC_PUBLIC | | | | | | | | R/W |
| | . . . | EPC_PUBLIC | | | | | | | | R/W |
| | 52 | EPC_PUBLIC | | | | | | | | R/W |
| | 51 | TID_SERIAL (Byte 5) | | | | | | | | R |
| | 50 | TID_SERIAL (Byte 4) | | | | | | | | R |
| | 49 | TID_SERIAL (Byte 3) | | | | | | | | R |
| | 48 | TID_SERIAL (Byte 2) | | | | | | | | R |
| | 47 | TID_SERIAL (Byte 1) | | | | | | | | R |
| | 46 | TID_SERIAL (Byte 0) | | | | | | | | R |
| | 45 | 0x00 | | | | | | | | R |
| | 44 | 0x20 | | | | | | | | R |
| | 43 | TID_MODEL [ 7 : 0 ] = $01010000_2$ | | | | | | | | R |
| | 42 | TID_DESIGNER [ 3 : 0 ] = $0001_2$ | | | | TID_MODEL [ 11 : 8 ] = $0001_2$ | | | | R |
| | 41 | XTID = 1 | TID_DESIGNER [ 10 : 4 ] = $0000000_2$ (Gen2 mask designer is address $08_h$ to $13_h$) | | | | | | | R |
| | 40 | CLASS_ID [ 7 : 0 ] = $11100010_2$ | | | | | | | | R |
| EPC ($01_2$) | 39 | EPC_PRIVATE | | | | | | | | R/W |
| | 38 | EPC_PRIVATE | | | | | | | | R/W |
| | . . . | EPC_PRIVATE | | | | | | | | R/W |
| | 24 | EPC_PRIVATE | | | | | | | | R/W |
| | 23 | NSI [ 7 : 0 ] | | | | | | | | R/W |
| | 22 | EPC_LENGTH [ 4 : 0 ] | | | | | UMI | XI (NVM) | NSI [ 8 ] | R/W |
| Reserved ($00_2$) | 21 | RFU = 0 | WWU = 0 | BPL_EN = 0 | QT_SR = 0 | QT_MEM = 0 | DCI_RF_EN = 0 | RF2_DIS = 0 | RF1_DIS = 0 | R/W |
| | 20 | RFU (WRITE as $00000000_2$) | | | | | | | | R/W |
| | 19 | BLOCK_PERMALOCK [ 8 : 15 ] | | | | | | | | R/W |
| | 18 | BLOCK_PERMALOCK [ 0 : 7 ] | | | | | | | | R/W |
| | 17 | RESERVED | | | | | | | | R |
| | . . . | RESERVED | | | | | | | | R |
| | 10 | RESERVED | | | | | | | | R |
| | 9 | LOCK_DA | RFU = $0000_2$ | | | | KILL | I2C_ADDR [ 1 : 0 ] = $11_2$ | | R/W |
| | 8 | LOCK_KILL [ 1 : 0 ] | LOCK_ACCESS [ 1 : 0 ] | | LOCK_EPC [ 1 : 0 ] | | LOCK_USER [ 1 : 0 ] | | | R/W |
| | 7 | ACCESS_PASSWORD | | | | | | | | R/W |
| | 6 | ACCESS_PASSWORD | | | | | | | | R/W |
| | 5 | ACCESS_PASSWORD | | | | | | | | R/W |
| | 4 | ACCESS_PASSWORD | | | | | | | | R/W |
| | 3 | KILL_PASSWORD | | | | | | | | R/W |
| | 2 | KILL_PASSWORD | | | | | | | | R/W |
| | 1 | KILL_PASSWORD | | | | | | | | R/W |
| | 0 | KILL_PASSWORD | | | | | | | | R/W |

*Read-Only memory is highlighted in blue. See legend below for memory descriptions.*

- **"I2C_ADDR"** = NVM configurable bits of I2C device address $1101XX0_2$, where $XX=11_2$ by default
- **"DCI_RF_EN"** = Enables RF when DCI is present
- **"QT_MEM"** = Selects memory map profile
  - 0: Tag uses private memory map
  - 1: Tag uses public memory map
- **"LOCK_DA"** = Permalock bit for I2C Device Address
- **"QT_SR"** = Short Range in open and secured states
- **"RF_DIS [ 1: 0 ]"** = RF disable selectable by port

### Table 5: I2C Interface Memory Map in a Word Wise Format

| GEN2 BANK NAME | I2C BIT ADDR | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | I2C PERM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| User (11₂) | 21F0h-21FFh | USER [15:0] | | | | | | | | | | | | | | | | W |
| | … | … | | | | | | | | | | | | | | | | W |
| | 200h-20Fh | USER [8191:8176] | | | | | | | | | | | | | | | | W |
| TID (10₂) | 1F0h-1FFh | EPC_PUBLIC [15:0] | | | | | | | | | | | | | | | | W |
| | … | … | | | | | | | | | | | | | | | | W |
| | 1A0h-1AFh | EPC_PUBLIC [95:80] | | | | | | | | | | | | | | | | W |
| | 190h-19Fh | TID_SERIAL [15:0] | | | | | | | | | | | | | | | | R |
| | 180h-18Fh | TID_SERIAL [31:16] | | | | | | | | | | | | | | | | R |
| | 170h-17Fh | TID_SERIAL [47:32] | | | | | | | | | | | | | | | | R |
| | 160h-16Fh | TDTS [15:0] = 0x2000 | | | | | | | | | | | | | | | | R |
| | 150h-15Fh | TID_DESIGNER [3:0] = 0001₂ | | | | TID_MODEL [11:0] = 150h | | | | | | | | | | | | R |
| | 140h-14Fh | CLASS_ID [7:0] = 11100010₂ | | | | | | | | XTID = 1 | TID_DESIGNER [10:4] = 0000000₂ (Gen2 mask designer is address 08h to 13h) | | | | | | | R |
| EPC (01₂) | 130h-13Fh | EPC_PRIVATE [15:0] | | | | | | | | | | | | | | | | W |
| | … | … | | | | | | | | | | | | | | | | W |
| | C0h-CFh | EPC_PRIVATE [127:112] | | | | | | | | | | | | | | | | W |
| | B0h-BFh | EPC_LENGTH [4:0] | | | | | UMI | XI = 0 | NSI [8:0] (Numbering System Identifier, default 000000000₂) | | | | | | | | | W |
| Reserved (00₂) | A0h-AFh | RFU = 0 | | | | | | | | | WWU | BPL_EN | QT_SR | QT_MEM | DCI_RF_EN | RF2_DIS | RF1_DIS | W |
| | 90h-9Fh | BLOCK_PERMALOCK [0:15] | | | | | | | | | | | | | | | | W* |
| | 80h-8Fh | RESERVED | | | | | | | | | | | | | | | | R |
| | 70h-7Fh | RESERVED | | | | | | | | | | | | | | | | R |
| | 60h-6Fh | RESERVED | | | | | | | | | | | | | | | | R |
| | 50h-5Fh | RESERVED | | | | | | | | | | | | | | | | R |
| | 40h-4Fh | LOCK_KILL [1:0] | | LOCK_ACCESS [1:0] | | LOCK_EPC [1:0] | | LOCK_USER [1:0] | | LOCK_DA | RU = 0 | | | | KILL | I2C_ADDR [1:0] | | W |
| | 30h-3Fh | ACCESS_PASSWORD [15:0] | | | | | | | | | | | | | | | | W |
| | 20h-2Fh | ACCESS_PASSWORD [31:16] | | | | | | | | | | | | | | | | W |
| | 10h-1Fh | KILL_PASSWORD [15:0] | | | | | | | | | | | | | | | | W |
| | 00h-0Fh | KILL_PASSWORD [31:16] | | | | | | | | | | | | | | | | W |

*Read-Only memory is highlighted in blue. See legend below for memory descriptions.*

- **"W*"** = Bits are writable from I2C but some are sticky. See Monza X-8K Dura I2C and Gen2 BlockPermalock for details.

- "**BLOCK_PERMALOCK[0:15]**" = BlockPermalock bits
- "**DCI_RF_EN**" = Enables RF when DCI is present
- "**QT_MEM**" = Selects memory map profile
  - 0: Tag uses private memory map
  - 1: Tag uses public memory map
- "**I2C_ADDR**" = NVM configurable bits of I2C device ID
- "**QT_SR**" = Short Range in open and secured states
- "**RF_DIS [ 1: 0 ]**" = RF disable selectable by port

## 2.6  I2C Control of Monza X-8K Dura Behavior

The I2C interface can control Monza X-8K Dura behavior by writing to bytes 8 or 9, 18 or 19, 20 or 21 and 22 or 23. The following sections describe how control bits in these words change the behavior.

## 2.7  Monza X-8K Dura I2C and Gen2 Lock Bits

The lock bits for the kill password (LOCK_KILL[1:0]), the access password LOCK_ACCESS[1:0]), the EPC memory bank (LOCK_EPC[1:0]), and the USER  memory bank (LOCK_ USER[1:0]) are in byte eight of memory. In each of these lock bit pairs bit 1 corresponds to pwd-write or  pwd-read/write and bit 0 corresponds to the permalock bit. Note that the I2C can always change the state of these bits and that their permissions only apply to the RF Gen2 interface.

**Table 6: Lock Bit-Field Functionality for EPC and User Memory Banks**

| PWD-WRITE | PERMALOCK | DESCRIPTION |
|:---:|:---:|---|
| 0 | 0 | Associated memory bank is writeable from either the **open** or **secured** states. |
| 0 | 1 | Associated memory bank is permanently writeable from either the **open** or **secured** states and may never be locked. |
| 1 | 0 | Associated memory bank is writeable from the **secured** state but not from the **open** state. |
| 1 | 1 | Associated memory bank is not writeable from any state. |

**Table 7: *Lock* Bit-Field Functionality for Access and Kill Passwords**

| PWD-READ/WRITE | PERMALOCK | DESCRIPTION |
|:---:|:---:|:---|
| **0** | 0 | Associated password location is readable and writeable from either the **open** or **secured** states. |
| **0** | 1 | Associated password location is permanently readable and writeable from either the **open** or **secured** states and may never be locked. |
| **1** | 0 | Associated password location is readable and writeable from the **secured** state but not from the **open** state. |
| **1** | 1 | Associated password location is not readable or writeable from any state. |

## 2.8   Monza X-8K Dura I2C and Gen2 BlockPermalock

Monza X-8K Dura segments user memory into 16 blocks. Blocks zero through 15 may be blockpermalocked from either the Gen2 interface or the I2C interface. A blockpermalocked block allows reads but not writes to the block. Blockpermalocking is permanent for blocks one through 15 and may not be unlocked from either interface. The blockpermalock may be undone for block zero from the I2C interface and I2C ignores the blockpermalock permission for block zero.

The 16 blocks as seen from the I2C interface are shown in Table 8. The figure includes the Gen 2 User memory bank bit addresses of the blocks and the I2C byte addresses of the blocks. Note that a large portion of the User memory bank has no permalock blocks. Please see the Gen2 specification for details on how a reader may lock the memory via *BlockPermalock* command.

## Table 8: BlockPermalock Blocks as Seen from the I2C Interface

| USER MEMORY BANK BIT RANGE | BLOCKS | I2C BLOCK PERMA LOCKABLE | GEN2 BLOCK PERMA LOCKABLE |
|---|---|---|---|
| 3584 - 8191 | Rest of User Memory (No BlockPermalock Blocks) | No | No |
| 3456 - 3583 | BLOCK 15 (128 bits) | Yes | Yes |
| 3328 - 3455 | BLOCK 14 (128 bits) | Yes | Yes |
| 3200 - 3327 | BLOCK 13 (128 bits) | Yes | Yes |
| 3072 - 3199 | BLOCK 12 (128 bits) | Yes | Yes |
| 2944 - 3071 | BLOCK 11 (128 bits) | Yes | Yes |
| 2816 - 2943 | BLOCK 10 (128 bits) | Yes | Yes |
| 2688 - 2815 | BLOCK 9 (128 bits) | Yes | Yes |
| 2560 - 2687 | BLOCK 8 (128 bits) | Yes | Yes |
| 2432 - 2559 | BLOCK 7 (128 bits) | Yes | Yes |
| 2304 - 2431 | BLOCK 6 (128 bits) | Yes | Yes |
| 2176 - 2303 | BLOCK 5 (128 bits) | Yes | Yes |
| 2048 - 2175 | BLOCK 4 (128 bits) | Yes | Yes |
| 1536 - 2047 | BLOCK 3 (512 bits) | Yes | Yes |
| 1024 - 1535 | BLOCK 2 (512 bits) | Yes | Yes |
| 512 - 1023 | BLOCK 1 (512 bits) | Yes | Yes |
| 0 - 511 | BLOCK 0 (512 bits) | No | Yes |

The mechanism for a microprocessor permalocking over I2C is as follows: Execute a one-word (2 byte) write to bytes 18 and 19 (word address nine). There are sixteen blockpermalock bits in bytes 18 and 19 that control the write permission to the 16 user-memory blocks. Monza X-8K Dura will bitwise OR each of the current permalock bits with the 15 bits corresponding to blocks one through 15 and write the updated word into NVM. Block zero may be unlocked via the I2C interface. Monza X-8K Dura does not allow unlocking of blockpermalocked memory in blocks one through 15 via either the Gen2 interface or I2C interface.

To control the Gen2 interface access to the *BlockPermalock* command the I2C interface will have a *BlockPermalock* command enable bit that only it can write to. When the bit BPL_EN is set to 1, Monza X-8K Dura will execute valid *BlockPermalock* commands and when it is cleared it will ignore all *BlockPermalock* commands. The location of the BPL_EN bit is in bit five of byte 21 (default value is 0).

## 2.9  Monza X-8K Dura I2C Control of LOCK_DA and I2C_ADDR[1:0] Bits

The I2C_ADDR[1:0] bits in byte 9 contain the I2C device address bits D2 and D1 respectively. These bits are meant to be reconfigurable in order to avoid address conflicts on the I2C bus, allowing four different address possibilities. The factory default value for these bits is $11_b$, corresponding to a default I2C slave

address of 1101**11**0$_b$. The LOCK_DA bit is a permalock bit that should be set once the I2C device address is selected.  The I2C_ADDR[1:0] and LOCK_DA bits are configurable only via I2C, not RF. Once the LOCK_DA bit is set to 1 the value of the LOCK_DA bit and I2C_ADDR[1:0] are preserved by the chip during all writes to bytes 8 and 9.

## 2.10 Monza X-8K Dura Control of the QT Function

The QT_SR and QT_MEM bits control in byte 21 control the QT functionality of Monza X-8K Dura. They have no effect on I2C operation and only change RF Gen2 behavior.  The two bits operate independently from each other. The bits may be changed via the QT command over RF or by writing to the bits via I2C (default values are both 0). The QT_SR bit turns on Monza X-8K Dura's short-range mode when it is set. When Monza X-8K Dura is in short range, operations in OPEN or SECURED states are required to be close to the reader.  Note, however, that if the Monza X-8K Dura antenna has a gain < -9dBi, there will be no OPEN or SECURED access through the RF port when the QT_SR bit is set.  The QT_MEM bit controls how Monza X-8K Dura's memory appears to the Gen2 interface.  When the bit is set Monza X-8K Dura is in public mode: user memory bank is hidden, TID serialization is hidden, and uses its EPC_PUBLIC in the EPC bank.  When the bit is cleared Monza X-8K Dura is in private mode and all of its memory is exposed. The memory map in Figure 6 shows the Monza X-8K Dura memory in both private and public data profiles.

**Figure 6: Monza X-8K Dura QT Memory Profiles**
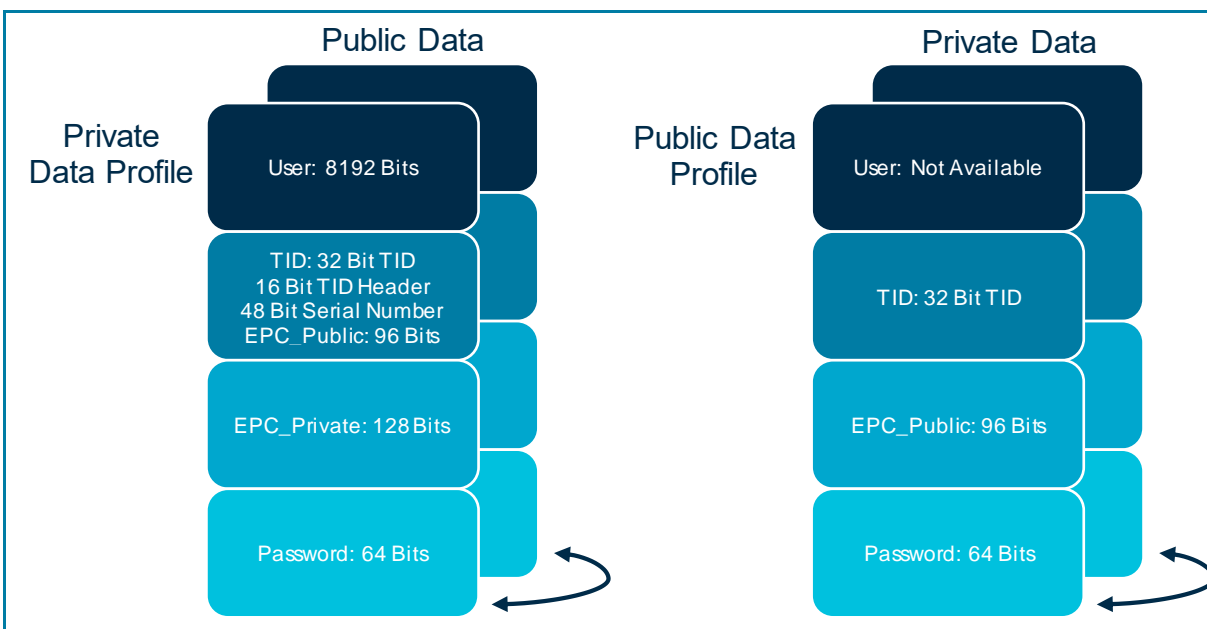


*These are the memory profiles that can be configured using the QT command. In the Private Data Profile, all the tag chip memory is available. In the Public Data Profile, only Public EPC is available, along with the TID header.*

## Table 9: QT Command Code

| COMMAND | CODE | LENGTH (BITS) | DETAILS |
|---------|------|---------------|---------|
| **QT** | 1110000000000000 | 68 | • The QT command controls the switching between the Private and Public profiles<br>• The tag must be in the SECURED state to transition to the memory indicated by the command<br>• If a tag receives a QT command with an invalid handle, it ignores that command<br>• The tag responds with the Insufficient Power error code if the power check fails on write<br>• The tag responds with the Other error code if the write times out |

## Table 10: QT Command Details

| QT COMMAND | CODE | READ/WRITE | PERSISTENCE | RFU | PAYLOAD | RN | CRC-16 |
|------------|------|------------|-------------|-----|---------|-----|--------|
| **#bits** | 16 | 1 | 1 | 2 | 16 | 16 | 16 |
| **Details** | 1110000000000000 | 0: Read<br>1: Write | 0: Temporary<br>1: Permanent | $00_b$ | QT Control | handle | |

**Table 11: QT Command Field Descriptions**

| FIELD | DESCRIPTION |
|---|---|
| **Read/Write** | • The Read/Write field indicates whether the tag reads or writes QT control data.<br>• Read/Write=0 means read the QT control bits in cache.<br>• Read/Write=1 means write the QT control bits |
| **Persistence** | • If Read/Write=1, the Persistence field indicates whether the QT control is written to nonvolatile (NVM) or volatile memory.<br>• Persistence=0 means write to volatile memory.<br>• Persistence=1 means write to NVM memory |
| **RFU** | • These bits are reserved for future use and will be ignored. |
| **Payload (QT Control)** | • This field controls the QT functionality. These bits are ignored when the Read/Write field equals<br>• Bit 15 (MSB) is first transmitted bit of the payload field.<br><br><table><tr><th>BIT #</th><th>NAME</th><th>DESCRIPTION</th></tr><tr><td>15</td><td>QT_SR</td><td>1: Tag reduces range if in or about to be in OPEN or SECURED state<br>0: Tag does not reduce range</td></tr><tr><td>14</td><td>QT_MEM</td><td>1: Tag uses Public Memory Map (see Figure 6)<br>0: Tag uses Private Memory Map (see Figure 6)</td></tr><tr><td>13:0</td><td></td><td>Reserved for future use. Tag will return these bits as zero.</td></tr></table> |
| **RN** | • The tag will ignore any QT command received with an invalid handle |

The tag response to the QT Command with Read/Write = 0 uses the preamble specified by the Text value in the Query command that initiated the round. See Figure 6 for read response details.

**Table 12: QT Command Details**

|  | HEADER | DATA | RN | CRC-16 |
|---|---|---|---|---|
| **#bits** | 1 | 16 | 16 | 16 |
| **Description** | 0 | QT Control | handle | |

## 2.11 Monza X-8K Dura I2C Control of Gen2 Response to Ack Command

The length field in byte 22 may be written from I2C. The length field specifies the number of words backscattered in response to a Gen2 *Ack* command. Byte 22 also contains an NVM space for the UMI bit which may be read or written from I2C. For the Monza X-8K, the Gen2 UMI bit is fixed at 0. The I2C master

may also do this to indicate that there is data in the user memory bank for the reader to read. The XI bit is always backscattered as a zero to the Gen2 reader even though the I2C master may read the underlying NVM bit.

## 2.12 RF Access Control

Monza X-8K Dura provides three levels of control over RF access as follows:

1. Setting either or both the RF2_DIS or RF1_DIS bits in byte 21 of the NVM disables RF access on the corresponding RF port. These bits are configurable only via I2C, not RF. The factory defaults are 0, enabling RF1 and RF2.

2. Setting the DCI_RF_EN bit to 0 in byte 21 of the NVM inhibits all RF access when DCI voltage is present. This takes precedence over the state of the RF1_DIS, RF2_DIS bit in #1 above. This bit is also only configurable from I2C, not RF. This bit is set by factory default to 0. Thus, by default, there is RF access to RF1, RF2 when DCI voltage is NOT present and no RF access when DCI voltage is present. The specification $V_{RFON}$ determines the DCI voltage that inhibits RF.

3. Setting the KILL bit 2 in byte 9 of the NVM will inhibit all RF access. This bit can be set from RF using a Gen2 KILL command or by writing from I2C. This is the normal mechanism for a reader to disable a chip. This KILL bit takes precedence over both #1 and #2 above. I2C can re-write this bit back to 0 and thus reverse and RF KILL command. The factory default for KILL is 0.

The factories defaults are set so Monza X-8K Dura operates like any other RFID tag when DCI voltage is not present. When DCI voltage is present the default behavior, through mechanism #2 above, is to inhibit all RF access. The KILL bit is always set to 0 at the factory by Gen2 definition. In conventional RFID chips once this bit is set to 1 a chip is dead and can never be resurrected. Monza X-8K Dura, having a hard wired I2C interface, allows un-doing the KILL operation from I2C.

## 2.13 Gen2/I2C Arbitration

Monza X-8K Dura can communicate via RF and via I2C. If Monza X-8K Dura is connected to an antenna and I2C Master with high DCI voltage, Monza X-8K Dura requires arbitration to determine how to respond to commands. Monza X-8K Dura can be configured to enable or disable RF communication when DCI voltage is high by setting the bit DCI_RF_EN to 0 or 1. If the DCI_RF_EN bit is set to 1, then Monza X-8K Dura can operate in three different states as shown in Figure 7. The states are "*Internal Control*", "*I2C Control*", and "*Idle or RF Receive*". If the DCI_RF_EN bit is set to 0 then RF will be disabled when DCI voltage is high allowing only I2C communication with the chip. Monza X-8K Dura won't respond to RF commands when in the *Idle or RF Receive* state under these conditions.

*Internal Control*: Monza X-8K Dura is in *Internal Control* when (1) executing an initialization sequence, (2) writing the NVM or (3) backscattering a response to an RF command. When in *Internal Control* Monza X-8K Dura ignores I2C transactions or RF commands.

*I2C Control*: Monza X-8K Dura is in *I2C Control* when a master is issuing commands to Monza X-8K Dura over the I2C bus. I2C Control starts when Monza X-8K Dura detects a matching device ID and is not under *Internal Control*. The I2C bus master releases control of Monza X-8K Dura either by ending a transaction with a stop bit or by issuing a subsequent start with a non-matching device ID. If Monza X-8K Dura was commanded to perform an NVM write then it moves to *Internal Control*, otherwise it returns to idle. When in *I2C Control* Monza X-8K Dura ignores all RF commands. Note that the master may stall the I2C bus (by holding SCL low) in the middle of a transaction and prevent RF access until releasing the bus.

*Idle or RF Receive*: Monza X-8K Dura is in *Idle or RF Receive* when receiving an RF command or when idle. After receiving a command Monza X-8K Dura transitions to *Internal Control* to execute the command. Executing a command may cause Monza X-8K Dura to (1) backscatter a reply (2) write to NVM or (3) change internal states. An I2C transaction may interrupt Monza X-8K Dura in *Idle or RF Receive*– by this means the I2C port exercises priority over the RF port and may not be locked out. Note that I2C is locked out when Monza X-8K Dura transitions to *Internal Control* to execute the command.

In certain operating states and under certain conditions Monza X-8K Dura may appear unresponsive to an I2C master for up to 20 milliseconds (during a slow Gen2 backscatter). This datasheet recommends that an I2C master have a retry algorithm that can accommodate Monza X-8K Dura being busy.

**Figure 7: Monza X-8K Dura Operating States**
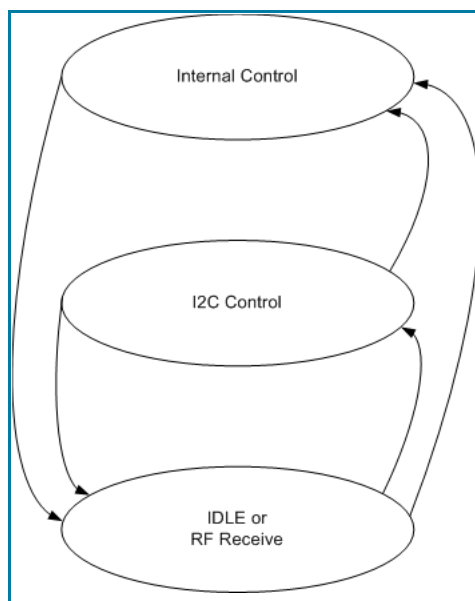


*This diagram shows the operating states when Monza X-8K Dura is connected to both an antenna and an I2C master with high DCI voltage.*

## 2.14 Write Wakeup Mode

Monza X-8K Dura has a wake-up feature that is tied to writes being performed over the Gen2 interface. This feature allows Monza X-8K Dura to trigger an event that may be used to wake up a device. To enable this feature, the I2C master must set the WWU bit (bit 6 of byte 21) to 1 (default value is 0). Then the master must set the Monza X-8K Dura's DCI pin to 0V (i.e. sleep mode). The SCL and SDA lines must remain high, but will draw no current. A reader may continue to interact with Monza X-8K Dura on the RF ports. If a reader performs a write operation, and the wake-up bit is set, Monza X-8K Dura will assert the SCL I/O pulling the SCL line low for the duration of the write operation, approximately 4ms. This transition is detectable by the sleeping master and may be used to wake up the system.

**Figure 8: Monza X-8K Dura Write Wakeup Mode Schematic and Timing Diagram**
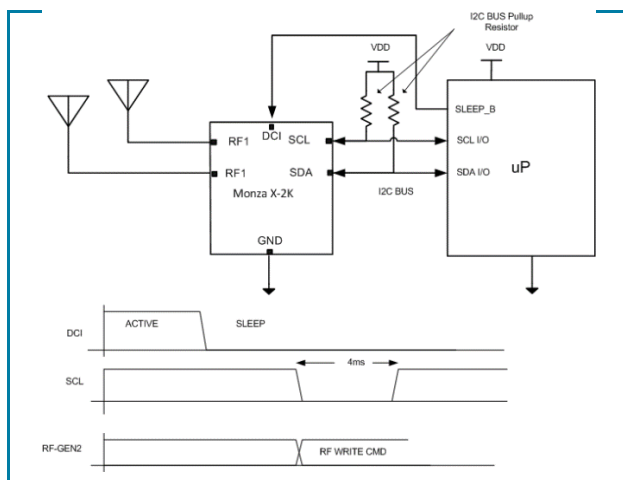
# 3   CHIP CHARACTERISTICS

## 3.1   Physical Characteristics

**Table 13: Packaging Information**

| PARAMETER | DESCRIPTION | CONDITION | MIN | NOM | MAX | UNITS | COMMENTS |
|---|---|---|---|---|---|---|---|
| **IC package** | Chip package | All | | | | XQFN 8L 2.00×2.00×0.35mm | |
| **Pin count** | Package pins | All | | 8 | | pins | 2 – Port1 RF+/–<br>2 – Port2 RF+/–<br>2 – DCI/gnd<br>2 – I2C(SDA/SCL) |

**Figure 9: Mechanical Dimensions\***
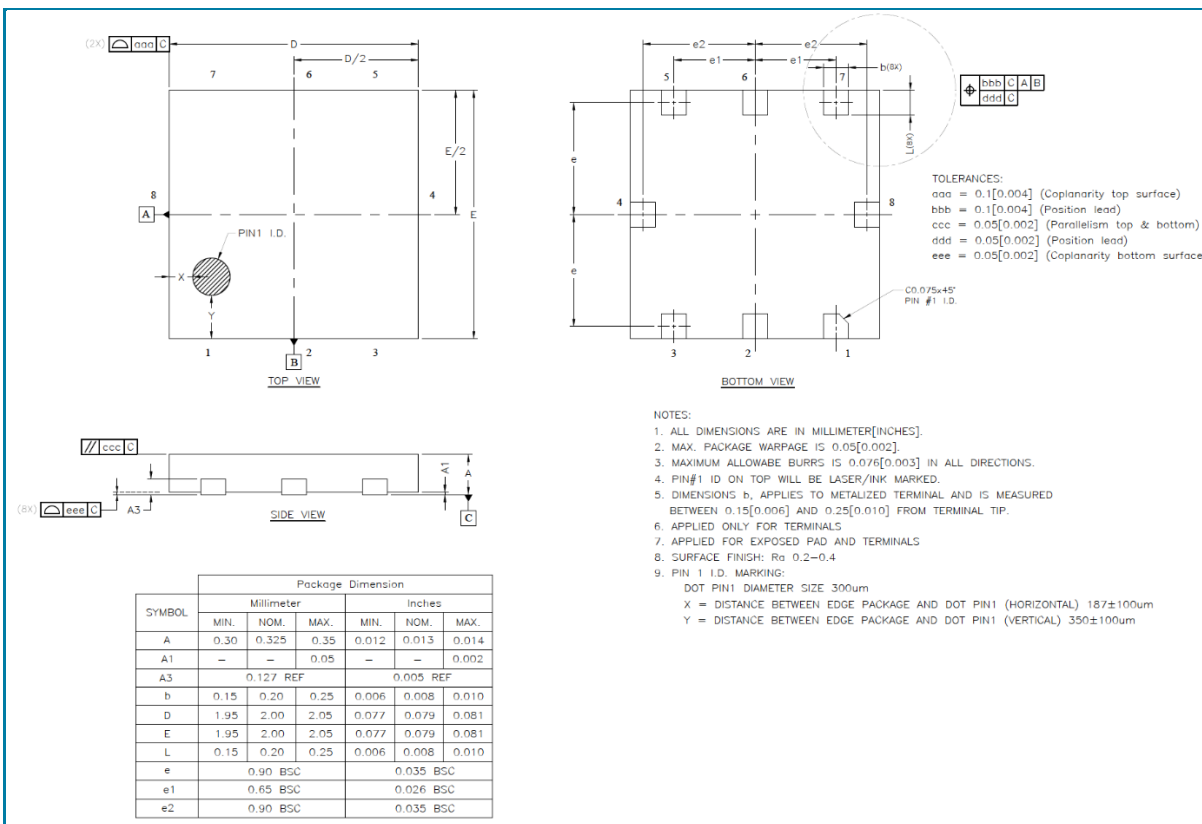


*\*The position of the Pin1 I.D. marking dot in the Mechanical Dimensions is specified for Monza X-8K Dura chips manufactured on or after date code "1807" as shown on the package box label, or "807" as shown on the product markings (see Product Delivery Specifications). Parts produced before this date code have a Pin1 I.D. marking positioned 100um closer to the vertical center line (B) of the package.*

## 3.2 Absolute Maximum Ratings

**Table 14: Absolute Maximum Ratings – Passive Mode**

| PARAMETER | DESCRIPTION | CONDITION | MIN | NOM | MAX | UNITS | COMMENTS |
|---|---|---|---|---|---|---|---|
| Absolute maximum pin voltage | Absolute maximum voltage on any chip pin | All except DCI | −0.3 | | 4.1 | V | From the I2C spec, the max DC voltage is 3.3V+20% (max operating voltage) + 0.5V for survivability |
| ESD | | HBM | 2 | | | kV | |
| | | CDM | 500 | | | V | |
| Operating Temperature | Temperature for full specified performance | | -25 | | +95* | °C | See Read/Write Sensitivity for temperature ranges in Section 3.4 |
| Persistence Temperature | Temperature for Gen2 flag persistence | | -25 | | +40 | °C | As per the Gen2 v.1.2.0 specification for flag persistence |
| Storage temperature | Temperature for 10-year NVM retention | | −40 | | +95* | °C | See Impinj's NVM usage model |
| Assembly survival temp | Temperature for reflow soldering / assembly | | | | +260 | °C | Peak temp of JEDEC-MO255 for lead free soldering |
| Moisture Sensitivity Level | Moisture/Reflow Sensitivity Classification | | | MSL1 | | | According to IPC/JEDEC's J-STD-20 |

*Maximum Operating/Storage temperature of +95 °C is specified for Monza X-8K Dura chips manufactured on the last week of September 2017 or later, which is date code "1739" or newer on the package box label, and "739" or newer on the product markings (see Product Delivery Specifications). Parts produced before this date are rated to a maximum Operating/Storage temperature of +85°C.

**Table 15: Absolute Maximum Ratings – Battery Assisted Passive (BAP) Mode**

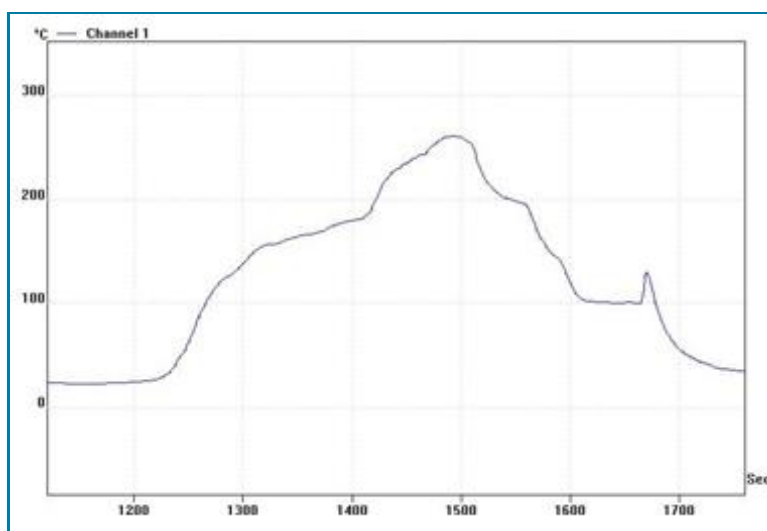| PARAMETER | DESCRIPTION | CONDITION | MIN | NOM | MAX | UNITS | COMMENTS |
|---|---|---|---|---|---|---|---|
| Absolute maximum pin voltage | Absolute maximum voltage on any chip pin | All except DCI | −0.3 | | 4.1 | V | From the I2C spec, the max DC voltage is 3.3V+20% (max operating voltage) + 0.5V for survivability |
| ESD | | HBM | 2 | | | kV | |
| | | CDM | 500 | | | V | |
| Operating Temperature | Temperature for full specified performance | | 0 | | +95* | °C | See Read/Write Sensitivity for temperature ranges in Section 3.4 |
| Persistence Temperature | Temperature for Gen2 flag persistence | | 0 | | +40 | °C | As per the Gen2 v.1.2.0 specification for flag persistence |
| Storage temperature | Temperature for 10-year NVM retention | | −40 | | +95* | °C | See Impinj's NVM usage model |
| Assembly survival temp | Temperature for reflow soldering / assembly | | | | +260 | °C | Peak temp of JEDEC-MO255 for lead free soldering |
| Moisture Sensitivity Level | Moisture/Reflow Sensitivity Classification | | | MSL1 | | | According to IPC/JEDEC's J-STD-20 |

*Maximum Operating/Storage temperature of +95 ℃ is specified for Monza X-8K Dura chips manufactured on the last week of September 2017 or later, which is date code "1739" or newer on the package box label, and "739" or newer on the product markings (see Product Delivery Specifications). Parts produced before this date are rated to a maximum Operating/Storage temperature of +85 ℃.*

## 3.3 Reflow Temperature Profile

**Figure 10: Reflow Temperature Profile**

## 3.4 Electrical Characteristics

**Table 16: Electrical Characteristics – RF Performance, Passive**

| PARAMETER | DESCRIPTION | CONDITION | MIN | NOM | MAX | UNITS | COMMENTS |
|---|---|---|---|---|---|---|---|
| $S_{READ}$ | Matched RF Input Read Sensitivity DRM, M=4 | Passive | | -17 | | dBm | Using DC Input, Monza X-8K Dura can be used in Battery Assisted Passive (BAP) mode to increase read/write range |
| $S_{WRITE}$ | Matched RF Input Write Sensitivity DRM, M=4 | Passive | | −12 | | dBm | |
| $R_p$ | Parallel Equivalent Real Input Impedance | At Sensitivity | | 1600 | | Ohms | |
| $C_p$ | Parallel Equivalent RF Input Capacitance | | | 1 | | pF | |

**Table 17: Electrical Characteristics – RF Performance, BAP**

| PARAMETER | DESCRIPTION | CONDITION | MIN | NOM | MAX | UNITS | COMMENTS |
|---|---|---|---|---|---|---|---|
| $S_{READ}$ | Matched RF Input Read Sensitivity DRM, M=4 | BAP | | -24 | | dBm | Using DC Input, Monza X-8K Dura can be used in Battery Assisted Passive (BAP) mode to increase read/write range |
| $S_{WRITE}$ | Matched RF Input Write Sensitivity DRM, M=4 | BAP | | -24 | | dBm | |
| $R_p$ | Parallel Equivalent Real Input Impedance | At Sensitivity | | 1600 | | Ohms | |
| $C_p$ | Parallel Equivalent RF Input Capacitance | | | 1 | | pF | |

## Table 18: Electrical Characteristics – Power

| PARAMETER | DESCRIPTION | CONDITION | MIN | NOM | MAX | UNITS | COMMENTS |
|---|---|---|---|---|---|---|---|
| $V_{DCI}$ | DCI Input Voltage/I2C Reference | 25 C to 95 C | 1.4* | | 3.6 | V | |
| | | -40 C to 25 C | 1.6 | | 3.6 | | |
| $I_{DCW}$ | Current drawn by chip during write | $1.4<V_{DCI}<2.0$ | | 100 | 200 | µA | |
| | | $2.0<V_{DCI}<3.6$ | | 140 | 220 | | |
| $I_{DCI}$ | Current drawn by chip during read or idle | $1.4<V_{DCI}<2.0$ | | 15 | 30 | µA | |
| | | $2.0<V_{DCI}<3.6$ | | 20 | 40 | | |
| $T_{PU}$ | Power Up Time. Time from VDCI applied until I2C accepts transactions. | VDCI=1.6V | | | 2 | ms | NOTE: I2C will not interrupt a write operation. This could delay I2C access up to 20ms if RF is writing. |
| $V_{RF\_EN}$ | Max Vdd for which RF will always be enabled | | | | 0.25 | V | Applies if the DCI_RF_EN bit is set to 0. |
| $V_{RF\_DIS}$ | Min Vdd for which RF will always be disabled | | 1.35 | | | V | |

*DCI voltage minimum of 1.4V is specified for Monza X-8K Dura chips manufactured on the last week of September 2017 (work week 39) or later, which is date code "1739" or newer on the package box label, and "739" or newer on the product markings (see Product Delivery Specifications). Parts produced before this date are rated to a minimum DCI voltage of 1.6V.*

## Table 19: Electrical Characteristics – I2C

| PARAMETER | DESCRIPTION | CONDITION | MIN | NOM | MAX | UNITS | COMMENTS |
|---|---|---|---|---|---|---|---|
| $V_{IH}$ | HIGH-level input voltage | All | 70% | | | %$V_{DCI}$ | |
| $V_{IL}$ | LOW-level input voltage | All | | | 30% | %$V_{DCI}$ | From the section 6 of the I2C specification |
| $V_{HYS}$ | Input hysteresis | All | 0.1 | | | V | |
| $V_{OL1}$ | Low Level output voltage 1 | 3mA sink current, VDCI > 2V | 0 | | 0.4 | V | (open-drain or open-collector) at 3mA sink current |
| $V_{OL2}$ | Low Level output voltage 2 | 2mA sink current, VDCI ≤ 2V | 0 | | 0.2 × VDCI | V | (open-drain or open-collector) at 2mA sink current |
| $T_{OF}$ | Output Fall Time | Bus C= 40-400pf | 20 | | 250 | ns | |
| $C_I$ | Pin Capacitance | | | | 10 | pF | Total capacitive load on the SDA/SCL pins |
| $I_{IL}$ | SCL/SDA Input Leakage Current | Vin=3.7V  0V<$V_{DCI}$<3.7V | | 1 | 100 | nA | Within I2C spec limit of 10uA |

## 3.5  Memory Characteristics

## Table 20: Memory Characteristics

| PARAMETER | DESCRIPTION | CONDITION | MIN | NOM | MAX | UNITS | COMMENTS |
|---|---|---|---|---|---|---|---|
| EPC Memory | EPC NVM | In Private Mode Only | | 128 | | Bits | User writeable. This memory is hidden over RF when QT is enabled. |
| User Memory | Total User NVM | In Private Mode Only | | 8192 | | Bits | User defined memory space. This memory is hidden over RF when QT is enabled |
| QT alternative EPC | Alternative EPC presented during RF singulation | In public mode only | | 96 | | bits | A user can switch the tag's RF QUERY–ACK response from EPC to alternative EPC using the QT command |
| Kill/Access Passwords | Password NVM | Access required | | 64 | | bits | Standard 32-bit Gen2 access and kill passwords |
| TID mfg#/serial# | TID ROM | In private mode only | | 96 | | bits | TID serial number is hidden over RF when QT is enabled |
| $T_{write}$ | Memory write time, 16 or 32 bits | | | 4.7 | 5 | Ms | |

## 3.6 RF Functionality

**Table 21: RF Functionality**

| PARAMETER | DESCRIPTION | CONDITION | MIN | NOM | MAX | UNITS | COMMENTS |
|---|---|---|---|---|---|---|---|
| **Air protocol** | Gen2 V1.2.0 | All | | | | | No recommissioning; no BlockErase |
| **RF ports** | Number of RF ports | All | | 2 | | | Dual-differential RF ports |
| **RF Port Disable** | NVM Settable bit per port | | | | | | The operation of one or both RF ports may be disabled by setting NVM bits through the I2Cport |
| **DC Blocks RF** | NVM Settable bit | | | | | | Option to allow the presence of DC to disable both RF ports |

## 3.7 I2C Characteristics

**Table 22: I2C Characteristics**

| PARAMETER | DESCRIPTION | CONDITION | MIN | NOM | MAX | UNITS | COMMENTS |
|---|---|---|---|---|---|---|---|
| **I2C port** | Number of ports | All | | | 1 | | Slave I2C (SCL/SDA) |
| **I2C functionality** | Compatible with I2C-bus specification and user manual Rev. 03 – 19 June 2007 | All | | R/W | | | An external device can R/W memory |
| **Supported I2C Bus Protocol Features** | Start condition | Supported | | | | | Mandatory I2C features for a slave device are supported |
| | Stop condition | Supported | | | | | |
| | Acknowledge | Supported | | | | | |
| | 7-bit slave address | Supported | | | | | |
| **I2C write size** | Word size for I2C write | All | 16 | | 32 | bits | Writes are on word addresses and not byte addresses |
| **I2C read size** | Word size for I2C read | All | | N×8 | | bits | May read data 8bits at a time, where N is limited by start address and banksize |
| **I2C memory arbitration** | RF/I2C port priority | All | | 1st | | | RF/I2C arbitrate for NVM access |
| **I2C Address** | I2C Device Address | All | | 1101XX0 | | | XX is I2C_ADDR[1:0] |
| **Transfer rates** | I2C transfer data rates | All | 0 | | 400 | kbps | I2C fast mode |

## 3.8   NVM Usage Model

**Table 23: NVM Usage Model**

| MAXIMUM CONDITION | | RETENTION (YEARS) |
|---|---|---|
| **WRITES PER ROW** | **TOTAL WRITES** | |
| **10** | 100 | 50 |
| **1k** | 10k | 10 |
| **10k** | 100k | 1 |

*Rows are 32 bits long on even word boundaries (i.e., words 0 and 1 are row 0 for user memory).*

## 3.9   Environmental Compliance

**Table 24: Environmental Compliance**

| REQUIREMENT | COMMENTS |
|---|---|
| **RoHS** | Monza X-8K Dura is RoHS compliant. It meets the directive 2002/95/EC (RoHS). RoHS declaration letter is available upon request. |
| **REACH** | Monza X-8K Dura does not, to our current knowledge, contain substances above the legal threshold that are on the Candidate List of Substances of Very High Concern (SVHC). Our company's intention is that all products sold to our EU and EEA customers by our legal entities in Europe are compliant with REACH regulatory requirements. REACH declaration letter is available upon request. |

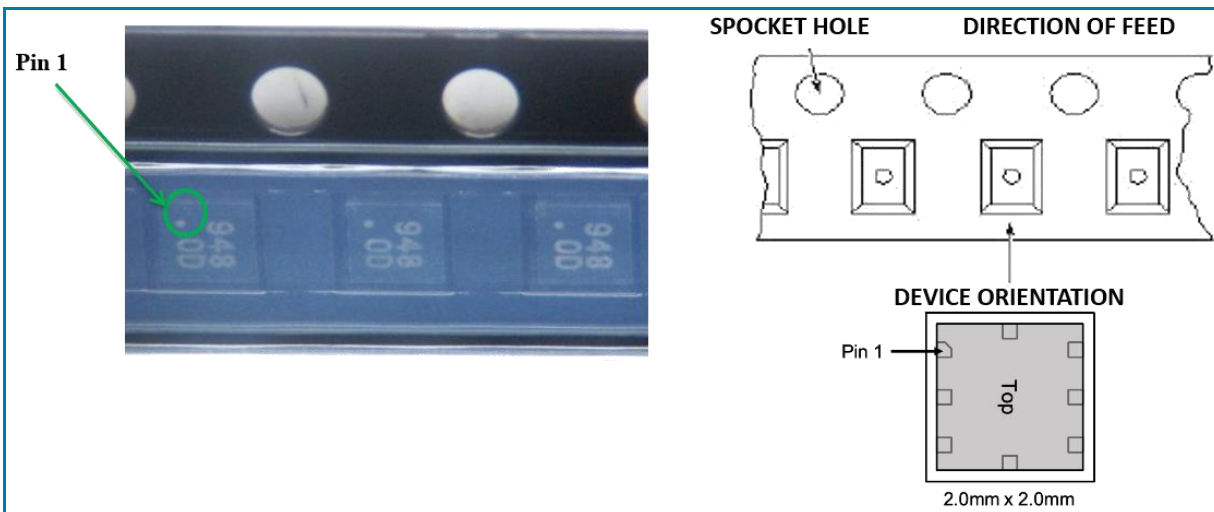# 4   PRODUCT DELIVERY SPECIFICATIONS

## 4.1   Marking Specification

- Y= Year of production (1 = 2011, 2 = 2012 …)
- WW = Work Week of production
- X8 = Product Code (Monza X-8K Dura)

**Figure 11: Marking Specifications**



Y W W

X 8

Pin # 1 dot
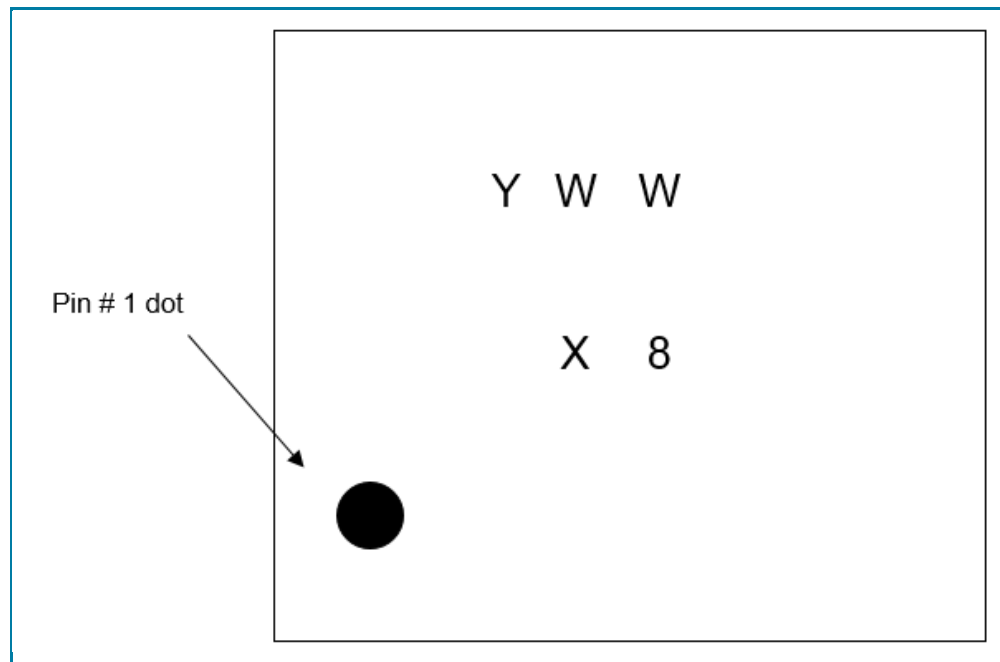
*This is the Product Marking as it appears on the Monza X-8K Dura chip. Note that YWW will be replaced with the production year and work week. For example: "739" would represent year 2017 and work week 39.*

**Figure 12: Device Orientation on Tape Carrier**



Pin 1

SPOCKET HOLE    DIRECTION OF FEED

DEVICE ORIENTATION

Pin 1    Top

2.0mm x 2.0mm

## 4.2  Tape and Reel Specification

### Figure 13: Tape and Reel Specification



*Parts per reel / Minimum order quantity: 3000*

### Figure 14: Reel Dimensions

# 5 ERRATA

The following table lists the known issues in Monza X-8K Dura.

**Table 25: Errata**

| ISSUE NUMBER | DESCRIPTION |
|---|---|
| 1 | When issuing a Select command to Monza X-8K, the tag will not respond correctly if the last bit of User memory, bit 8191, is used as part of the mask. If this bit is used, the compare against the mask and this bit will not be able to succeed. |

# 6 FOOTPRINT COMPATIBILITY WITH IMPINJ MONZA® X-2K DURA

Monza X-8K Dura (Part Order#: IPJ-P6005-X2AT) is a higher memory capacity version of Monza® X-2K Dura (Part Order#: IPJ-P6001-Q2AT).

Monza X-2K Dura is designed to have 2176 bits of user NVM, enabling more OTP blocks. Its package dimensions are 1.6 x 1.6 x 0.35 mm. It is designed to be a drop-in replacement for Monza X-8K Dura if the layout footprint recommended below is used. For more details about the Monza X-2K Dura including product availability, please contact Impinj.

**Figure 15: Recommended Common Layout Footprint – Monza X-8K Dura and Monza X-2K Dura**
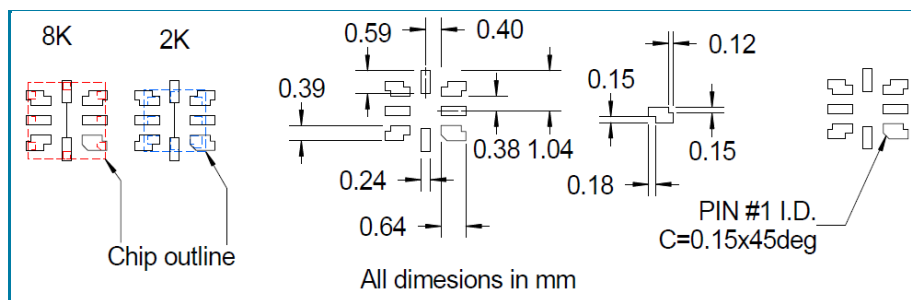


# 7 ORDERING INFORMATION

Contact sales@impinj.com for ordering support.

**Table 26: Ordering Information**

| MODEL | PART NUMBER | USER MEMORY | PACKAGE SIZE |
|---|---|---|---|
| Monza X-8K Dura | IPJ-P6005-X2AT | 8,192 bits | 2.0 x 2.0 x 0.35 mm |

# 8  EXTERNAL REFERENCE

[1] Third Party Link: *EPC™ Radio Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz – 960 MHz, Version 1.2.0 (Gen 2 Specification)* (https://www.gs1.org/sites/default/files/docs/epc/uhfc1g2_1_2_0-standard-20080511.pdf)

[2] Third Party Link: *GS1's EPC Tag Data Standard* (https://www.gs1.org/sites/default/files/docs/epc/GS1_EPC_TDS_i1_11.pdf)

[3] Third Party Link*: EPCglobal Interoperability Test System for EPC Compliant Class-1 Generation-2 UHF RFID Devices v.1.2.4, August 4, 2006* (https://www.gs1.org/docs/epc/uhfg2_1_0_9-TestMethod-Interoperability_1_2_4-20060627.pdf)

[4] Third Party Link: *I2C Specification Rev. 03, June 19 2007*, *NXP Doc UM10204* (Latest Version: https://www.nxp.com/docs/en/user-guide/UM10204.pdf)

# 9  NOTICES

Copyright © 2019, Impinj, Inc. All rights reserved.

Impinj gives no representation or warranty, express or implied, for accuracy or reliability of information in this document.  Impinj reserves the right to change its products and services and this information at any time without notice.

EXCEPT AS PROVIDED IN IMPINJ'S TERMS AND CONDITIONS OF SALE (OR AS OTHERWISE AGREED IN A VALID WRITTEN INDIVIDUAL AGREEMENT WITH IMPINJ), IMPINJ ASSUMES NO LIABILITY WHATSOEVER AND IMPINJ DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATED TO SALE AND/OR USE OF IMPINJ PRODUCTS INCLUDING  LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT.

NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY PATENT, COPYRIGHT, MASKWORK RIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT IS GRANTED BY THIS DOCUMENT.

Impinj assumes no liability for applications assistance or customer product design. Customers should provide adequate design and operating safeguards to minimize risks.

Impinj products are not designed, warranted or authorized for use in any product or application where a malfunction may reasonably be expected to cause personal injury or death, or property or environmental damage ("hazardous uses"), including but not limited to military applications; life-support systems; aircraft control, navigation or communication; air-traffic management; or in the design, construction, operation, or maintenance of a nuclear facility. Customers must indemnify Impinj against any damages arising out of the use of Impinj products in any hazardous uses

Impinj, and Impinj products and features are trademarks or registered trademarks of Impinj, Inc. For a  complete list of Impinj Trademarks, visit www.impinj.com/trademarks. All other product or service names may be trademarks of their respective companies.

The products referenced in this document may be covered by one or more U.S. patents. See www.impinj.com/patents for details.