Application Note

IMPINJ M730 AND M750

ENCODING RECOMMENDATIONS

Version 1.0

www.impinj.com

TABLE OF CONTENTS

1 INTRODUCTION

The Impinj M730 and M750 RAIN RFID tag chips provide high performance, fast inventory capability, and advanced features for next-generation, universal RAIN RFID tags. This document highlights important information and outlines general guidance for encoding Impinj M730 and M750 tag chips.

For additional information about these tag chips, please refer to the Impinj M730 & M750 Datasheet on the Impinj Support Portal at https://www.impinj.com/support.

Users may use this document alongside the whitepaper "Ensuring Encoding Quality - Service Bureau Best Practices for Monza Chips," also available through the Impinj Support Portal.

2 ENCODING CONSIDERATIONS

2.1 Encoding Overview

Note the following details when encoding Impinj M730 and M750 tag chips:

- The kill and access passwords are always the same value and have the same lock status
- The user can put the tag into short range in one of two ways, shown below. Both options require the tag to be in the secured state with a non-zero access password. The Impinj M730 & M750 Datasheet covers these methods in more detail.
  - Write the SR bit to 1 in Reserved memory word 4
  - Use the Gen2v2 *Untraceable* command
- The memory write speed for Impinj M700 series is 3.2 ms per *Write*, *BlockWrite*, *Lock, Kill*, or *Untraceble* operation, for writing up to 32 bits

2.2 Tag Chip Memory Overview

The primary difference between the Impinj M730 and M750 tag chips is the tag chip memory organization on each chip. Each chip may also be identified by the tag model number in the TID memory. Table 1 summarizes the memory configuration and the TID identification.

Table 1: Impinj M700 Series Memory Organization

PRODUCT	FIRST 32 BITS OF TID (HEX)	FACTORY DEFAULT PC WORD (HEX)	EPC MEMORY SIZE (BITS)	USER MEMORY SIZE (BITS)	SHARED ACCESS/KILL PWD
Impinj M730	E280 1191	3000	128	0	Yes
Impinj M750	E280 1190	3400	96	32	Yes

Table 2 shows the Reserved memory map, which is identical for both tag chips.

Table 2: Impinj M730 and M750 Reserved Memory Map

MEMORY BANK NUMBER	MEMORY BANK NAME	MEMORY BANK BIT ADDRESS	BIT ADDRESS															
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00_2	RESERVED (NVM)	140_h-$14F_h$	RFU[12:0]=000_h													ATV[2:0]		
		70_h-$7F_h$	Factory Calibration C[15:0]															
		60_h-$6F_h$	Factory Calibration B[15:0]															
		50_h-$5F_h$	Factory Calibration A[15:0]															
		40_h-$4F_h$	Internal Configuration[15:5]											SR	Internal Configuration[3:1]			A
		30_h-$3F_h$	Shared Access Password[15:0]															
		20_h-$2F_h$	Shared Access Password[31:16]															
		10_h-$1F_h$	Shared Kill Password[15:0]															
		00_h-$0F_h$	Shared Kill Password[31:16]															

Note: The Impinj M730 and M750 tag chips have a single 32-bit password; the Access and Kill passwords are shared and aliased over one another.

2.2.1 Encoding the Shared Password

For encoding the shared password to the tag, Impinj recommends encoding the access password only at Reserved memory words 2-3. This will set the kill password to the same value automatically.

2.2.2 Locking Tag Memory

Impinj M700 series tag chips must have the access and kill passwords locked in the same way. The table below lists specific examples of valid payloads for locking the *Access* and *Kill* passwords. Users can lock additional memory along with passwords as well. The tag chip supports additional payloads unless the access and kill password lock settings conflict. If the payload for the *Lock* command is not valid, the tag chip will respond back with an error code “Not supported” (00000001_b). For example, if the user issues a payload to lock or permalock User memory for an Impinj M730 tag, which has no User memory, the tag will respond with the error code “Not supported” (00000001_b) if it is in the secured state and receives a *Lock* command with such a payload.

Locking the access/kill passwords make the passwords read/write protected. Locking EPC or User memory makes it password write protected. Permalocking the access/kill passwords will make them permanently read/write protected and unchangeable. Permalocking the EPC and User memory makes these memory banks permanently write protected and unchangeable. Users may read EPC and User memory regardless of lock status. For full details about the *Lock* command, refer to the Gen2v2 specification.

Table 3: Supported *Lock* Command Payloads for Locking Passwords

LOCK COMMAND PAYLOAD (HEX)	*LOCK* COMMAND PAYLOAD (BINARY)	*LOCK* OPERATION ACTION				TAG CHIP SUPPORTS LOCK SEQUENCE?	
		KILL PWD	ACCESS PWD	EPC MEMORY	USER MEMORY	IMPINJ M730	IMPINJ M750
F03C0	1111 0000 0011 1100 0000	Perma-locked	Perma-locked	No change	No change	Yes	Yes
A0280	1010 0000 0010 1000 0000	Locked	Locked	No change	No change	Yes	Yes
A0000	1010 0000 0000 0000 0000	Unlocked	Unlocked	No change	No change	Yes	Yes
F0140	1111 0000 0001 0100 0000	Perma-unlocked	Perma-unlocked	No change	No change	Yes	Yes
AC2B0	1010 1100 0010 1011 0000	Locked	Locked	Perma-locked	No change	Yes	Yes
A82A0	1010 1000 0010 1010 0000	Locked	Locked	Locked	No change	Yes	Yes
F83E0	1111 1000 0011 1110 0000	Perma-locked	Perma-locked	Locked	No change	Yes	Yes
FCFF3	1111 1100 1111 1111 0011	Perma-locked	Perma-locked	Perma-locked	Perma-locked	No	Yes
F8BE2	1111 1000 1011 1110 0010	Perma-locked	Perma-locked	Locked	Locked	No	Yes
A8AA2	1010 1000 1010 1010 0010	Locked	Locked	Locked	Locked	No	Yes
A8000	1010 1000 0000 0000 0000	Unlocked	Unlocked	Unlocked	Unlocked	Yes	Yes

Note: Users can only issue a Lock command to a tag in the ***secured*** *state. Additional valid configuration options are not listed above. If one of the fields was previously permalocked or permaunlocked, that lock status cannot be changed, and the Lock command will fail.*

2.2.2.1 Locking Tag Memory Examples

Below are examples of how to lock Impinj M730 or M750 tags using payloads shown in Table 3.

Example 1: Encode an Impinj M730/M750 tag with a unique access password, unique EPC, lock passwords and permalock EPC memory.

- Write non-zero access password to the tag at Reserved memory words 2
- Write EPC to desired value
- Lock tag with *Lock* command with payload of $AC2B0_h$[1]

Access and kill passwords are locked and are readable or writable from the **secured** state but not from the **open** state. EPC memory is permanently locked and is not writable from **open** or **secured** state.

Example 2: Encode an Impinj M730/M750 tag with a unique access password, unique EPC, permalock passwords and lock EPC memory.

- Write non-zero access password to the tag at Reserved memory words 2
- Write EPC to desired value
- Lock tag with *Lock* command with payload of $F83E0_h$

Access and kill passwords are permanently locked and are not readable or writable from **open** or **secured** state. EPC memory is locked and is writable from the **secured** state but not from the **open** state.

---

[1] From the Gen2v2 specification, after tags are inventoried, they will transition to the **secured** state if the tag has a zero-value access password. A factory default tag with a zero-value access password may have the password written and tag locked without use of the *Access* command if the tag is not re-inventoried after the password is written. If the tag is inventoried after it has a non-zero-value access password, the *Access* command sequence must be used to transition the tag from the **open** to **secured** state. The tag must be in the **secured** state to lock the tag with the *Lock* command, write a locked EPC, read a locked access or kill password.

3 NOTICES