NVIDIA JETSON TX2 FUSE SPECIFICATION

DA-08415-001_v1.3 | September 2017

Application Note

DOCUMENT CHANGE HISTORY

DA-08415-001_v1.3

Version	Date	Description of Change
1.0	May 3, 2017	Initial Release
1.1	August 4, 2017	•Updated Table 1 •Removed "ODM Field Programmable Fuses Used by NVIDIA Software" section •Removed "Fuse Programming" section •Added note regarding state of eMMC/UFS pins during the fuse programming process
1.2	August 25, 2017	Updated FUSE_BOOT_SECURITY_INFO [5:0] in Table 1
1.3	September 8, 2017	Updated "Arm Debug Authentication Signals" section

TABLE OF CONTENTS

LIST OF TABLES

INTRODUCTION

This application note provides a technical overview of the issues and considerations related to the NVIDIA® Jetson™ TX2 Fuse Specification.

NVIDIA Jetson TX2 includes customer/Original Device Manufacturer (ODM)-programmable fuses which are used to store security keys and ODM system design configuration options. Fuses are divided into 2 distinct areas:

- Manufacturing Fuses (for example, security keys, boot options, etc.)
- ODM Field Fuses (for example, defined by ODM software for rollback protection, IDs, etc.)

All fuses default values are Logic 0 when not programmed. After they are programmed they represent Logic 1.

Note: Jetson TX2 utilizes the NVIDIA® Tegra® X2 which is a Parker series SoC.

SYSTEM REQUIREMENTS

Jetson TX2 contains all the power and logic to program the onboard fuses. The system designer does not have to make any provision on their own system design.

FUSE VARIABLES

Jetson TX2 contains 2 types of fuses for ODM use. Those that configure the device and should be programmed during the manufacturing process before the product is released to the end user, and those that may be programmed during the lifetime of the product by the ODM for software to use.

An example of each of these is:

- Manufacturing Fuses: Boot Keys, Boot device
- ODM Fuses: Product serial number, date of first use

MANUFACTURING PROGRAMMABLE FUSES

Jetson TX2 contains multiple manufacturing fuses that control different items for security and boot. These fuses should be programmed during the manufacturing process. The ODM Production Mode fuse (also known as “Security Mode”) should always be programmed by the ODM on the manufacturing line before the product is shipped to the end user. This fuse acts as a master lock for all of the manufacturing fuses. Once programmed it locks the values of the other manufacturing fuses. They cannot be programmed once the ODM Production Mode fuse has been programmed.

Table 1 summarizes available fuse settings and values for each.

Note: All ODM fuses have the value of ZEROs when shipped to a customer.

CAUTION: Programming a fuse (changing the value of a fuse from 0 to 1) is non-reversible. Once a fuse bit is programmed (set to 1), you cannot change the fuse value from 1 to 0. For example: A value of 1 (0x0001) can be changed to 3 (0x0011) or 7 (0x0111). It cannot however be changed to a value of 4 (0x0100) since bit zero is already programmed to 1.

The burning of fuses should be done without a system reset between different phases.

The eMMC/UFS must be powered and pins associated with eMMC/UFS should not be driven externally during the fuse programming process if either of the following conditions holds true:

1. It is a boot device.
2. RPMB provisioning is done on this device along with fuse burning.

Table 1. Fuse Name and Description

Fuse Name	Fuse Description	Bit Length	Notes
FUSE_SECURITY_MODE [0]	**ODM Production Mode** Also known as ODM Security Mode. This fuse write-protects all manufacturing device fuses against any further fuse programming and also hides the SBK values. **This fuse must be programmed last**.	1	
FUSE_ARM_JTAG_DIS [0]	**ARM JTAG Disable** Disables future use of ARM JTAG debug port. When this fuse is programmed, access to the ARM JTAG debug port is permanently disabled.	1	Note 3

Fuse Name	Fuse Description	Bit Length	Notes
FUSE_DEBUG_AUTHENTICATION [4:0]	**ARM Debug Authentication** Provides fine control of ARM debug capabilities Programming one of these fuses permanently disables the equivalent debug capability: •Bit 0 forces dbgen to 0 •Bit 1 forces niden to 0 •Bit 2 forces spiden to 0 •Bit 3 forces spniden to 0 •Bit 4 forces deviceen to 0	5	Note 3
FUSE_PRIVATE_KEY0 [31:0] /../ FUSE_PRIVATE_KEY3 [31:0]	**Secure Boot Key (SBK)** Stores an ODM-supplied secure boot key for each chip. Used of SBK is dependent on the authentication scheme selected via fuse_boot_security_info. Example: “0xABCDEF” input value will be represented as “0x00000000000000000000000000ABCDEF”	128	Note 1, 3, 4
FUSE_PUBLIC_KEY0 [31:0] /../ FUSE_PUBLIC_KEY7 [31:0]	**Public Key Hash (PKC)** Stores the hash of a public key provided by the ODM. Storing the hash allows to authenticate the full key.	256	Note 3
FUSE_RESERVED_SW [3]	**Skip Boot Device Selection Straps** Ignores the device selection straps and chooses the secondary boot device from the fuses when set.	1	Note 3
FUSE_RESERVED_SW [2:0]	**Boot Device Selection** Identifies the OS image boot device. Enumerated value read by the internal boot ROM.	3	Note 2, 3
FUSE_BOOT_DEVICE_INFO [23:0]	**Boot Device Configuration** Identifies the OS image boot device configuration. Used in conjunction with the Boot Device Selection to provide its configuration.	24	Note 2, 3

Fuse Name	Fuse Description	Bit Length	Notes
FUSE_BOOT_SECURITY_INFO [5:0]	**Boot Security Info** Bits interpreted with the following mapping [1:0] mapped to Secure Boot Authentication Scheme, where 00b: AES-CMAC using SBK 01b: AES-CMAC using SBK 10b: 2048 bit RSA 11b: NIST P-256 Curve ECC [2] enables encryption using SBK (all firmware images will be encrypted with SBK) [5:3] reserved	6	Note 3
FUSE_RESERVED_SW [5]	**Watchdog Enable** Used to enable watchdog	1	Note 3
FUSE_CCPLEX_DFD_ACCESS_DISABLE [0]	**CCPLEX Low-Level DFD ACCESS DISABLE** When fuse is programmed, low-level hardware debugging for NVIDIA internal diagnostics is totally disabled.	1	Note 3
FUSE_KEK00 [31:0] FUSE_KEK01 [31:0] FUSE_KEK02 [31:0] FUSE_KEK03 [31:0] FUSE_KEK10 [31:0] FUSE_KEK11 [31:0] FUSE_KEK12 [31:0] FUSE_KEK13 [31:0] FUSE_KEK20 [31:0] FUSE_KEK21 [31:0] FUSE_KEK22 [31:0] FUSE_KEK23 [31:0]	**Key Encryption Key or Key Seed** These 12 consecutive registers can be used to encode some Key Encryption Key and/or some Key Seed, with different combinations of width. Software interprets them as the following: KEK0 (128), KEK1 (128) and KEK2 (128). KEK256 (256-bit key) can be used as a 256-bit key-encryption key; or, as a 128-bit key-encryption key (KEK0) and a 128-bit key generation key (KEK1)	384	Note 3
FUSE_ODM_INFO [15:0]	**ODM Info** 8 LSB contain the 8 LSB of the USB PID for an USB device used for dead battery boot compliance. Bit 14 and Bit 15 are reserved for use by NVIDIA. Remaining bits are reserved for use by ODM.	16	Note 3
FUSE_ODM_CRC [8:0]	**ODM CRC** **T**he 8 LSB are a CRC used to check integrity of a subset of ODM programmed information, the MSB is a present/valid bit (cannot rely on CRC being not zero to indicate CRC is present).The use of a CRC is optional.	9	Note 3, 5

Fuse Name	Fuse Description	Bit Length	Notes
FUSE_ODMID0 [31:0] FUSE_ODMID1 [31:0]	**ODM ID** These 2 consecutive registers encode a 64 bit ODM ID.	64	Note 3
FUSE_SATA_MPHY_ODM_CALIB [3:0]	**FUSE SATA/MPHY Calibration** Calibration for the IO brick used for SATA or MPHY. [1:0] Varies SATA/MPHY pad TX_AMP and _PEAK to compensate for different trace lengths. [3:2] Reserved for other board dependent calibration	4	Note 3, 6
FUSE_H2	**Hamming Code** Implement the ECC for the ODM manufacturing fuses. **This fuse must be programmed just before burning ODM Production Mode**.	14	Note 3, 5, 6

Notes:

1. The SBK is not active to encrypt objects such as the boot loader, CFG, etc. until the ODM Production Mode fuse is programmed. Even if these entries are non-zero, the value is valid and can be read back (for example, used for SSK calculation). After ODM production fuse is programmed and a subsequent reset, the SBK value cannot be read back.
2. See the boot options fuse configuration table for the correct Boot settings for your platform.
3. Fuse programming is disabled when ODM Production Mode fuse = 1.
4. After programming the value and rebooting the chip, the value is an input to the SSK calculation regardless of whether the ODM Production Mode fuse has been set.
5. Programming of these fuses will be done by NVIDIA software.
6. Check secure boot software package for fuse burning operation.

ODM Production Fuse

The ODM production fuse is a global lock of all the manufacturing fuses. During the manufacturing process, software should program all other manufacturing fuses, then update the Hamming ECC field (**FUSE_H2**), then program the ODM production fuse last.

Debug Disable

There are two fuses which impact the ability to debug Tegra X2 ARM processors.

ARM_JTAG_DISABLE

When programmed, this fuse permanently prevents any JTAG access to the debug access port that occurs through the JTAG pins on Tegra X2. This prevents any JTAG access by external ARM debuggers during normal product lifetime.

Note: Boundary Scan is still possible through the JTAG pins irrespective of this fuse state.

CCPLEX_DFD_ACCESS_DISABLE

When this fuse is programmed (to a 1), NVIDIA internal CCPLEX debug access is disabled on the chip. Programming this fuse will prevent NVIDIA from performing any hardware level debug on the CCPLEX, should it be required.

ARM Debug Authentication Signals

These fuses control the standard ARM debug authentication signals; each fuse forces the corresponding signal to 0 (disabled). Table 2 describes the ARM debug authentication signals.

Table 2. ARM Debug Authentication Signals

Signal Name	Description	Definition	Common Use Case
DBGEN	**Debug Enable** When asserted, enables invasive and non-invasive debug of non-secure state. Note that when DBGEN is not asserted access to debug components is generally still permitted, but those components are disabled.	NonSecure Invasive Debug Enable	CPUs to halt AXIAP to make system accesses ETR to stream trace to DRAM
NIDEN	**Non-Invasive Debug Enable** When asserted, enables non-invasive debug operations, such as trace, of non-secure state. NIDEN can be asserted independently of DBGEN.	NonSecure Non Invasive Debug Enable	PTM trace from CPUs
SPIDEN	**Secure Privileged Invasive Debug Enable** When asserted along with DBGEN, enables invasive and non-invasive debug of Secure state.	Secure Invasive Debug Enable	AXI_AP to make secure accesses into the system ETR to write to Secure DRAM
SPNIDEN	**Secure Privileged Non-Invasive Debug Enable** When asserted along with NIDEN, enables non-invasive debug of Secure state.	Secure Non Invasive Debug Enable	Accessing Secure registers in PMU and CPUs over the Debug APB
DEVICEEN	**Device Debug Enabled** Enables the external debug tools connection to the device. This signal also drives the DBGSWENABLE which is an enable input signal of the CoreSight Components and Cortex-A Series processor.	Device Enable	Accessing any registers on mapped over the Debug APB

Secure Boot Key

These fuses should be programmed with the Secure Boot key if SBK is being used. The SBK only takes effect once the ODM production Mode fuse has been programmed.

Public Key Hash

These fuses should be programmed with the hash of the ODM public key. It only takes effect once the ODM production Mode fuse has been programmed.

Skip Boot Device Selection Straps

This fuse determines if the boot device selection is determined by the straps or by the fuse settings.

Jetson TX2 is supplied as configured to boot from Straps. It is recommended that for Production devices the fuses are used to select the boot device and this fuse should be programmed. When this fuse is programmed then the boot device is determined by the setting of the Boot Device Selection fuses.

Boot Device Selection

Jetson TX2 uses eMMC for boot. These fuses should remain at their default (0x0 = eMMC).

Table 3. Boot Selection (FUSE_RESERVED_SW[2:0])

Register	Description	Values
FUSE_RESERVED_SW [2:0]	Boot Device Select	0x0 = eMMC

Boot Device Information

These fuses determine parameters for the boot device. Jetson TX2 uses eMMC for boot. These fuses should be programmed to 0x0020 (No DDR, Query Voltage, Boot Mode off, 25.5 MHz, 1.8V, 512 Byte Page size) if boot fuses are to be programmed.

Table 4. Boot Device Configuration (eMMC Only)

Device	Fuse Bits	Description	Values (Default = 0x0)
eMMC	23:6	Reserved	Ignored; set to 0x0
	5	MultiPage support	0x0 = default Multi page read (page size determined by se length and

Device	Fuse Bits	Description	Values (Default = 0x0)
			DMA capability, target memory/buffer size/limits)
			0x1 = Single page read (512 Byte)
	4:3	Clock Divider PLLP clock at 408 Mhz	0x0 = default clock divider 16 (clock at 25.5 MHz)
			0x1 = clock divider 8 (clock at 51 MHz)
			0x2 = Reserved
			0x3 = Reserved
	2	Disable Boot Mode	0x0 = Boot mode Off
			0x1 = Boot mode On
	1	Voltage Range	0x0 = Query Voltage
			0x1 = Low Voltage
	0	DDR Mode Selection	0x0 = Normal
			0x1 = DDR

ECC

Individual fuses can fail with very low probability and the fuse logic corrects these failures by using redundancy techniques:

- An OR-ECC, where two fuses are ORed together to get the corrected value. This code is unidirectional and protects against a 1b becoming a 0b.
- A Hamming ECC applied to a set of fuses and able to correct one error in the set of protected fuses. The Hamming code has much less overhead than the OR-ECC, but requires groups of bits to be programmed together.

Both ECC methods are transparent to software when using fuse option registers to get access to fuse information, but requires some care when programming fuses.

ODM FIELD PROGRAMMABLE FUSES

The following fuses are available for the system designer to use for programming during the product lifetime. If these fuses are to be altered, then the fuse programming voltage **VPP_FUSE** must be present in the system at the time the fuses are to be programmed.

The **RESERVED_ODM** fuses are split into 8 banks of 32bits. The first of these 4 banks (0-3) can be locked out by setting the corresponding bit in the ODM Lock fuse (for example, to lock **RESERVED_ODM** Bank 1, then ODM LOCK Bit [1] should be set). This will prevent any unintentional programming of other bits in this bank.

RESERVED_ODM Banks 4-7 do not have this lock feature.

Table 5. Field Programmable Fuses

Fuse Name	Fuse Description	Bit Length
Reserved ODM (FUSE_RESERVED_ODM0 [31:0]) /../ (FUSE_RESERVED_ODM7 [31:0])	Customer programmable fuses. One anticipated application of the Reserved ODM fuses is software version revocation, although their use is solely at the discretion of the customer. The Reserved ODM fuses remain programmable after the ODM Production Mode fuse has been programmed. Default value is set to all zeros for no Reserved ODM fuses programmed.	256
ODM_lock (FUSE_ODM_LOCK [3:0])	ODM_lock[i] disables further change to the i-th 32 bits subset of the reserved ODM field. Applicable to the first four subsets only. FUSE_ODM_LOCK [0] = Reserved ODM[0] FUSE_ODM_LOCK [1] = Reserved ODM[1] FUSE_ODM_LOCK [2] = Reserved ODM[2] FUSE_ODM_LOCK [3] = Reserved ODM[3]	4

Note: Refer to *Jetson Device Secure Boot and Fuse Burning README and Tools* for information on how to program the fuses.

Notice

The information provided in this specification is believed to be accurate and reliable as of the date provided. However, NVIDIA Corporation ("NVIDIA") does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This publication supersedes and replaces all other specifications for the product that may have been previously supplied.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and other changes to this specification, at any time and/or to discontinue any product or service without notice. Customer should obtain the latest relevant specification before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer. NVIDIA hereby expressly objects to applying any customer general terms and conditions with regard to the purchase of the NVIDIA product referenced in this specification.

NVIDIA products are not designed, authorized or warranted to be suitable for use in medical, military, aircraft, space or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer's own risk.

NVIDIA makes no representation or warranty that products based on these specifications will be suitable for any specified use without further testing or modification. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to ensure the product is suitable and fit for the application planned by customer and to do the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this specification. NVIDIA does not accept any liability related to any default, damage, costs or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this specification, or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this specification. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA. Reproduction of information in this specification is permissible only if reproduction is approved by NVIDIA in writing, is reproduced without alteration, and is accompanied by all associated conditions, limitations, and notices.

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the NVIDIA terms and conditions of sale for the product.

ARM

ARM, AMBA and ARM Powered are registered trademarks of ARM Limited. Cortex, MPCore and Mali are trademarks of ARM Limited. All other brands or product names are the property of their respective holders. "ARM" is used to represent ARM Holdings plc; its operating company ARM Limited; and the regional subsidiaries ARM Inc.; ARM KK; ARM Korea Limited.; ARM Taiwan Limited; ARM France SAS; ARM Consulting (Shanghai) Co. Ltd.; ARM Germany GmbH; ARM Embedded Technologies Pvt. Ltd.; ARM Norway, AS and ARM Sweden AB.

Trademarks

NVIDIA, the NVIDIA logo, Jetson, and Tegra are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright