Jetson Xavier NX Fuse Specification

Application Note

Document History

DA-09876-001_v1.0

Version	Date	Description of Change
01	April 6, 2020	Initial Release

Table of Contents

List of Tables

Introduction

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

Jetson Xavier NX module 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 burned. After they are burned, they represent Logic 1.

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 block code ECC, based on a CRC, applied to a set of fuses. The ECC can correct one error in the set of protected fuses combined with very good error detection when more than one errors are present. This ECC has much less overhead than the OR-ECC but requires groups of bits to be burned together.

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

System Requirements

Jetson Xavier NX 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.

Note: The voltage supplied to the module should not be removed during burning.

Fuse Variables

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

An example of each of these is:

- ODM manufacturing fuses
  - Boot keys
  - Boot device
  - Product serial number
- ODM field programmable fuses
  - Date of first use
  - OTA information

Manufacturing Programmable Fuses

Jetson Xavier NX contains multiple manufacturing fuses that control different items for security and boot. These fuses should be burned during the manufacturing process. The ODM Production Mode fuse (also known as "Security Mode") should always be burned 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 the manufacturing fuses. Once burned it locks the values of the other manufacturing fuses. They cannot be burned once the ODM Production Mode fuse has been burned.

Table 1 summarizes available fuse settings and values for each.

> **Note**: All ODM fuses have the value of ZEROs when shipped to an ODM.

> ! CAUTION: Burning a fuse (changing the value of a fuse from 0 to 1) is non-reversible. Once a fuse bit is burned (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 burned to 1.
>
> The burning of fuses should be done without a system reset between different phases.
>
> The eMMC must be powered and pins associated with eMMC should not be driven externally during the fuse burning process if either of the following condition holds true:
>
> - It is a boot device.
> - RPMB provisioning is being done on this device along with fuse burning.

Table 1. Fuse Names and Descriptions

Fuse Name	Fuse Description	Bit Length	Notes
FUSE_SECURITY_MODE [0]	**ODM Production Mode** Also known as Security Mode. This fuse write-protects all manufacturing device fuses against any further fuse burning and also hides the SBK values. This fuse must be burned last.	1	
FUSE_ARM_JTAG_DIS [0]	**ARM JTAG Disable** Completely disables the external debug paths, including ARM JTAG path and USB SWD path. This field complements the ARM debug authentication field.	1	Note 3
FUSE_DEBUG_AUTHENTICATION [4:0]	**ARM Debug Authentication** Provides fine control of ARM debug capabilities Burning 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 Name	Fuse Description	Bit Length	Notes
FUSE_PRIVATE_KEY0 [31:0] /../ FUSE_PRIVATE_KEY3 [31:0]	**Secure Boot Key (SBK)** Stores an ODM-supplied secure boot key for each chip. Use of SBK and the authentication scheme are 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)** These eight consecutive registers encode a 256-bit hash of the ODM public key.	256	Note 3
FUSE_EK0 [31:0] /../ FUSE_EK7 [31:0]	**Endorsement Key** This key might be burned in encrypted form, with decryption performed by boot ROM.	256	Note 3
FUSE_RESERVED_SW [23:0]	**Reserved Bits for Software (read by Boot ROM)** Bit [2:0] Boot Device Select – Identifies the OS image boot device Bit [3] If set, will boot using the device selected in Bits 2:0 Bit [4] Reserved Bit [5] Enable Watchdog Bit [6] Reserved Bit [7] RCM SS Mode Enable – option to enable USB RCM to use SS transfer mode Bit [8] Reserved Bit [9] Reserved Bit [10] Enable Low Batt check and stall boot if SOC_GPIO02 is pulled low Bit [23] Disable entry into RCM mode	24	Note 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 [15:0]	**Boot Security Info** Bits interpreted by boot software with following mapping: Bits [1:0] mapped to Secure Boot Authentication Scheme, where 00b: SHA2 Hash 01b: 2048 bit RSA 10b: 3072 bit RSA 11b: ECC (Elliptic Curve, see also bit 7) Bit [2] secure boot encryption scheme, enables encryption using SBK when set to 1 Bit [3] ODM FEK usage enable Bits [6:4] ODM Fuse Encryption Key Select Bit [7] only used if bits 1:0 is set to 11b (elliptic) 0b = ECDSA with NIST P256 curve 1b = EdDSA (Ed25519) Bits not listed are reserved	16	Note 3
FUSE_SECURE_PROVISION_INFO [1:0]	**Factory Secure Provisioning** Allows the ODM to control secure provisioning features: [0] is hide bit; [1] is test_part bit. The hide bit ([0]) should be burned *before* burning SBK fuses if the Factory Secure Provisioning feature is being used.	2	Note 3
FUSE_CCPLEX_DFD_ACCESS_DISABLE[0]	**CCPLEX Low-Level DFD Access Disable** CCPLEX power management DFD Disable access (when fuse is burned, DFD access is 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.	384	Note 3

Fuse Name	Fuse Description	Bit Length	Notes
FUSE_ODM_INFO [15:0]	**ODM Info** No predefined use, free to use by ODM.	16	Note 3
FUSE_ODMID0 [31:0] FUSE_ODMID1 [31:0]	**ODM ID** These 2 consecutive registers encode a 64-bit ODM ID.	64	Note 3
FUSE_H2 [31:0]	**Hamming Code** Implement the ECC for the ODM manufacturing fuses. This fuse must be burned just before burning ODM Production Mode.	32	Notes 3, and 5
FUSE_FORCE_DEBUG_WITH_TEST_KEYS [0]	**Debug with Test Keys Enable** Enable the use of test keys when debug is allowed (to protect production key)	1	Note 3
FUSE_FLW2 [0]	**Force Large Weight** This bit is used as part of the ECC scheme, burn to 1 to ensure field H2 has large enough weight.	1	Note 3
FUSE_OPT_USB2NVJTAG_DISABLE [0]	Disable the USB debug path, note that this is independent of FUSE_ARM_JTAG_DIS	1	Note 3

Notes:

1. SBK will be used to decrypt the bootloader and Boot Config Table if encryption is enabled through boot_security_info fuse.
2. See the boot options fuse configuration table (Table 4) for the correct Boot settings for your platform.
3. Fuse burning of ODM manufacturing programmable fuses is disabled when ODM Production Mode fuse = 1.
4. After burning 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. Burning of these fuses will be done by NVIDIA software.

ODM Production Fuse

The ODM production fuse is a global lock of all the manufacturing fuses. During the manufacturing process, software should burn all other manufacturing fuses, then update the CRC ECC field (FUSE_H2, this may require to also update the FUSE_FLW2 field), then burn the ODM production fuse last. A reset is required for the lock to take into effect.

Debug Disable

There are three fuses which impact the ability to debug Jetson Xavier NX ARM processors.

ARM_JTAG_DISABLE

When burned, this fuse permanently prevents any JTAG access to the debug access port that occurs through the JTAG pins on the module. This prevents any JTAG access by external ARM debuggers during normal product lifetime. This also disables the USB SWD debug path.

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

CCPLEX_DFD_ACCESS_DISABLE

When this fused is burned, NVIDIA internal CCPLEX debug access is disabled on the chip. Burning 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

Signal Name	Description	Definition	Common Use Case
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 mapped over the Debug APB

Secure Boot Key

These fuses should be burned with the Secure Boot Key if SBK is being used. The SBK values are hidden once the ODM production mode fuse has been burned.

Public Key Hash

These fuses should be burned with the hash of the ODM public key.

Skip Boot Device Selection Straps

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

Jetson Xavier NX is supplied as configured to boot from straps. For production devices it is recommended that the fuses are used to select the boot device and this fuse should be burned. When this fuse is burned then the boot device is determined by the settings of the "Boot Device Selection" fuses.

Boot Device Selection

Jetson Xavier NX 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 Xavier NX uses eMMC for boot. These fuses should be burned to 0x3 if boot fuses are to be burned.

Table 4. Boot Device Configuration (FUSE_BOOT_DEVICE_INFO[7:0] eMMC Only)

Fuse Bits	Device	Description	Values (Default = 0x0)
7:0	eMMC	Data Rate: SDR Clk: 51 MHz MultiPage Support: READ_MULTIPLE	0x00
		Data Rate: SDR Clk: 25.5 MHz MultiPage Support: READ_MULTIPLE	0x01
		Data Rate: SDR Clk: 25.5 MHz MultiPage Support: READ_SINGLE	0x02
		Data Rate: DDR Clk: 51 MHz MultiPage Support: READ_MULTIPLE	0x03
		Data Rate: DDR Clk: 25.5 MHz MultiPage Support: READ_MULTIPLE	0x04

ODM Field Programmable Fuses

The following fuses are available for the system designer to use for burning during the product lifetime. If these fuses are to be altered, the module power supplies must be present throughout the fuse burning.

The RESERVED_ODM fuses are split into 12 banks of 32 bits. 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 burning of other bits in this bank.

RESERVED_ODM Banks 4-11 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_ODM11 [31:0])	The consecutive registers are reserved for the customer use, including ODM/software versioning. These fuses are field programmable, Reserved_ODM{0:3} can be individually locked against further burning using corresponding bits in ODM_Lock, bit [b] locks Reserved_ODM{b}.	384
ODM_lock (FUSE_ODM_LOCK [3:0])	ODM_lock[i] disables further changes 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 the *NVIDIA Tegra Linux Driver Package Development Guide* for information on how to burn the fuses.

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation ("NVIDIA") makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. 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 document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information 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 ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

Unless specifically agreed to in writing by NVIDIA, 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 this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform 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 document. NVIDIA accepts no 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 document 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 document. 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 document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND 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. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. 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 Terms 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, and Jetson Xavier 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

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
http://www.nvidia.com