
# X-Wall MX-256C
# X-Wall MX-128C; X-Wall MX-256
# FIPS 140-2 certified SATA Full Disk Crypto Module
# Specification Rev. 2.2 Redacted

## Revision History

| Rev No. | Description | Author | Rev. Date |
|---|---|---|---|
| 1.0 | Initial release. | C. Y. Chiu, R. Wann, L. Lee | 01/23/2007 |
| 1.1 | Revised x58 & x59 API Command. | C. Y. Chiu | 05/24/2007 |
| 1.2 | Add key byte information related to API command; Modify API programming. | C. Y. Chiu, R. Wann | 06/06/2007 |
| 1.3 | Add 1.8V and 3.3V Power Consumption;<br>Add BOM of circuit layout;<br>General document editing. | C. Y. Chiu<br>L. Lee<br>R. Wann | 07/19/2007 |
| 1.3.1 | Revise cfgXWall on page 17 under CCR;<br>General document editing. | C. Y. Chiu<br>R. Wann | 08/07/2007 |
| 1.3.2 | Redefine Pin24; Pin43; Pin45, Pin46, Pin50<br>Update Typical Application Schematics | L. Lee<br>R. Wann | 08/30/2007 |
| 1.4 | Correction of AES key order convention; Reflect ECN 001 MX10092007; Sync2PHY connects to VDD3.3V | C. Y. Chiu<br>R. Wann | 01/25/2008 |
| 1.5 | Update MX Reference Schematics;<br>Revise definition of Pin 13;<br>Revise definition of Pin 37; | C. Y. Chiu<br>Butz Huang<br>R. Wann | 06/18/2008 |
| 1.5.1 | Update signal routing and typical BOM | Butz Huang | 07/01/2008 |
| 1.5.2 | Updating Packaging Information | R. Wann | 07/15/2008 |
| 1.6 | Document changes required at Pin#44 'pmMode' (page 7) to work with Silicon Image **SATALink SPIF 215A USB/SATA** bridge controller | B. Huang<br>C. Y. Chiu<br>R. Wann | 0723/2008 |
| 1.7 | Redefine Pin#17 Sync2Phy & Pin#34, 51;<br>Redefine Operating Temperature; | C. Y. Chiu<br>R. Wann | 02/06/2009 |
| 1.8 | Add SATA signal layout; Add PCB parameters with differential signals on page 13; Document changes required to work with Oxford **OXUF934DSB** bridge controller on page 8 & 9**;** | C. Y. Chiu<br>R. Wann | 06/24/2009 |
| 1.9 | Add patent, layout, and contact Information | R. Wann | 07/29/2010 |
| 2.0 | Update Pin#21 CfgHost definition on page 8 in RED. | C. Y. Chiu | 08/17/2010 |
| 2.1 | Update SEEPROM protocols on page 15; Select Atmel & Microchip 24LC02B on page 9; | R. Wann | 09/21/2010 |
| 2.2 | General Editing | R. Wann | 05/04/2015 |
| | Intentionally left blank | | |


***Asia Pacific***

***Enova Technology Corporation***
1st Floor, #11, Research & Development 2nd Rd.
Science-based Industrial Park, Hsin-Chu City
Taiwan 30076, Republic of China
P +886 3 577 2767  F +886 3 577 2770
www.enovatech.net; info@enovatech.net;

***North America***

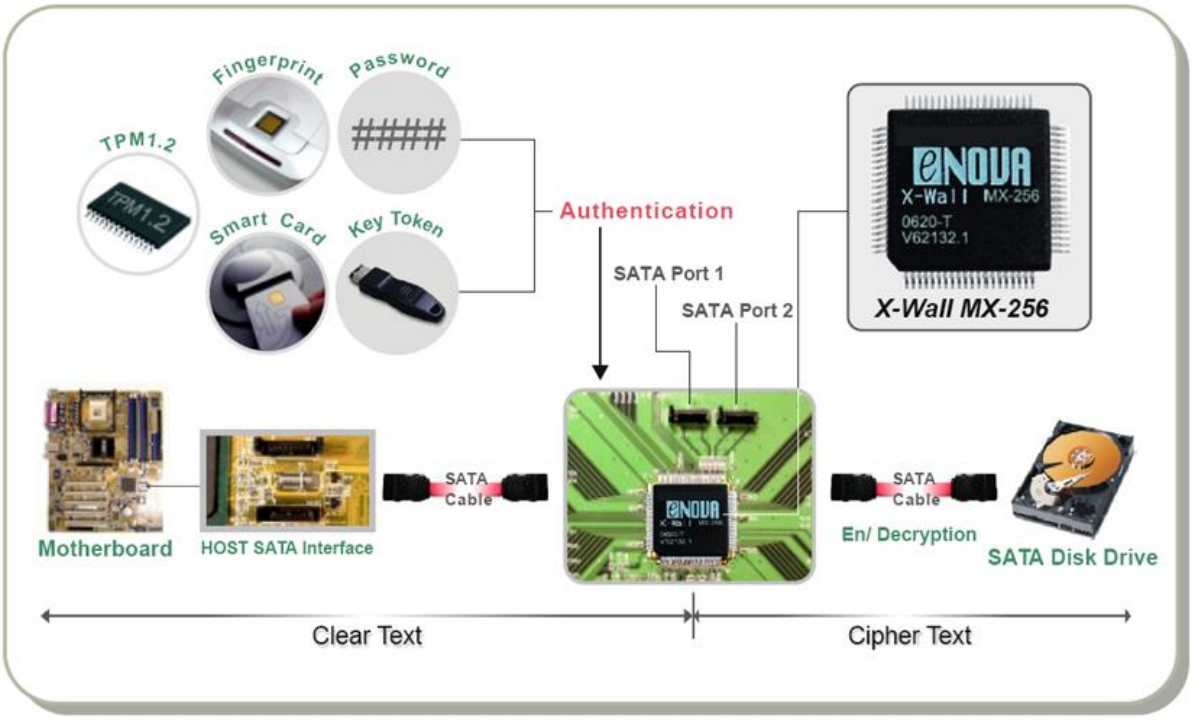
***Enova Technology***
1918 Junction Avenue
San Jose, California 95131, USA
P +1 .510.825.7900
http://www.enovatech.com
www.enovatech.com; info@enovatech.com

# Table of Content

# Introduction

The **patented**[1] ***X-Wall® MX*** family ASIC (Application Specific Integrated Circuit) is the 7th generation of ***X-Wall* real-time Full Disk Encryption technology**. It is engineered specifically to encrypt/decrypt entire SATA hard drive including boot sector and operating system without performance degradation. The cryptographic engine of the *X-Wall MX is NIST (National Institute of Standards and Technology)* and *CSE (Communications Security Establishment)* certified hardware *AES (Advanced Encryption Standard)* algorithm. **Certain *X-Wall MX* processors are currently pending review of an FIPS 140-2 certification process.**

## *How does it work?*




*X-Wall MX*, an SATA to SATA cryptographic bridge chip, sits between motherboard host SATA and the device SATA hard drive, encrypting entire SATA drive with wire speed performance while providing up to 256-bit AES hardware strength.

System performance with *X-Wall MX* engaged is unaffected. *X-Wall MX* can be operated with SATA 1.0a and SATA 2.6 compliant disk drives[2] with a **sustained** cryptographic throughput of 150MB/sec. The performance-optimized AES hardware engine performs all encryption and decryption. There are no extra software components, eliminating entirely the memory and interrupt overheads.

*X-Wall MX* requires no device driver and is independent from and invisible to all known Operating Systems including embedded OS. As long as the drive is SATA 1.0a and SATA 2.6 compliant, *X-Wall MX* will work in

[1] **US patents 7,136,995; 7,386,734; and Application 11/282,175. Taiwan & PR China patent 625110.**

[2] SATA 3Gbit drives can be operated on the *X-Wall MX* without performance degradation.

the system. Once authenticated, its operation is completely transparent to all users. There is no complex GUI involved therefore your regular computing behavior is unchanged.

## Key Management

*Key Management with X-Wall MX* can be versatile, which includes PIN/Password through Pre-boot authentication, TPM1.2, **CAC/PIV** Smartcard, Fingerprint, Single Sign On, or USB type external key token. One or more factor authentications are applicable through either built-in two-wire serial interface or API (Application Programming Interface) through SATA bus.

As the entire SATA hard drive is encrypted, there is no possibility of any secret being left unprotected on the drive, including password and "*Secret Key*." In an *X-Wall MX* protected system (drive), there is no simple way to read the data without the right "*Secret Key*." Only YOU have the right *Key* to unlock your data.

The *X-Wall MX* technology is compatible with all system designs incorporating SATA hard drive technologies.

## Key Benefits

- FIPS 140-2 certification review pending
- Offers wire speed performance at **sustained**[3] 150MB/sec on **all** cryptographic strengths
- Operating System independent
- Provides iron-clad security through hardware-based NIST & CSE certified cryptographic AES engine in both ECB and CBC mode of operation
- Simplify engineering design for security targets
- Configurable host and device SATA ports
- Multiple key load allows key swapping for drive deployment & repurposing

## Features

- Built-in Power-On-Self-Test (POST) ability to ensure product reliability
- POST includes cryptographic function tests
- Patented Crypto/Bypass mode switching capability[4]
- Versatile Key Management through either serial interface or built-in API (Application Programming Interface)
- 100% hardware AES (both ECB and CBC mode of operation) cryptographic engine producing sustained 150MB/sec real-time performance
- Very low power consumption (<400mW under continuous burst)
- 80-pin TQFP small form factor
- RoHS & Lead-free compliant
- 5 (Five) years warranty for defective part
- Military grade operating temperature from -45 to +90

[3] Please reference to test reports at http://www.enovatech.net/support/download/X-Wall%20MX%20Test%20Report_Part%202.pdf.

[4] Crypto/Bypass mode switching is an important technology that allows read/write of clear/cipher text data off the SATA drive at designer's discretion over unique application. Consult Enova Technology engineering support for this enhanced feature.

## System Requirement

- All Microsoft Windows Operating Systems
- Linux OS with SATA support
- All embedded OS with SATA support
- SATA 1.0a & SATA 2.6 compliant hard drives

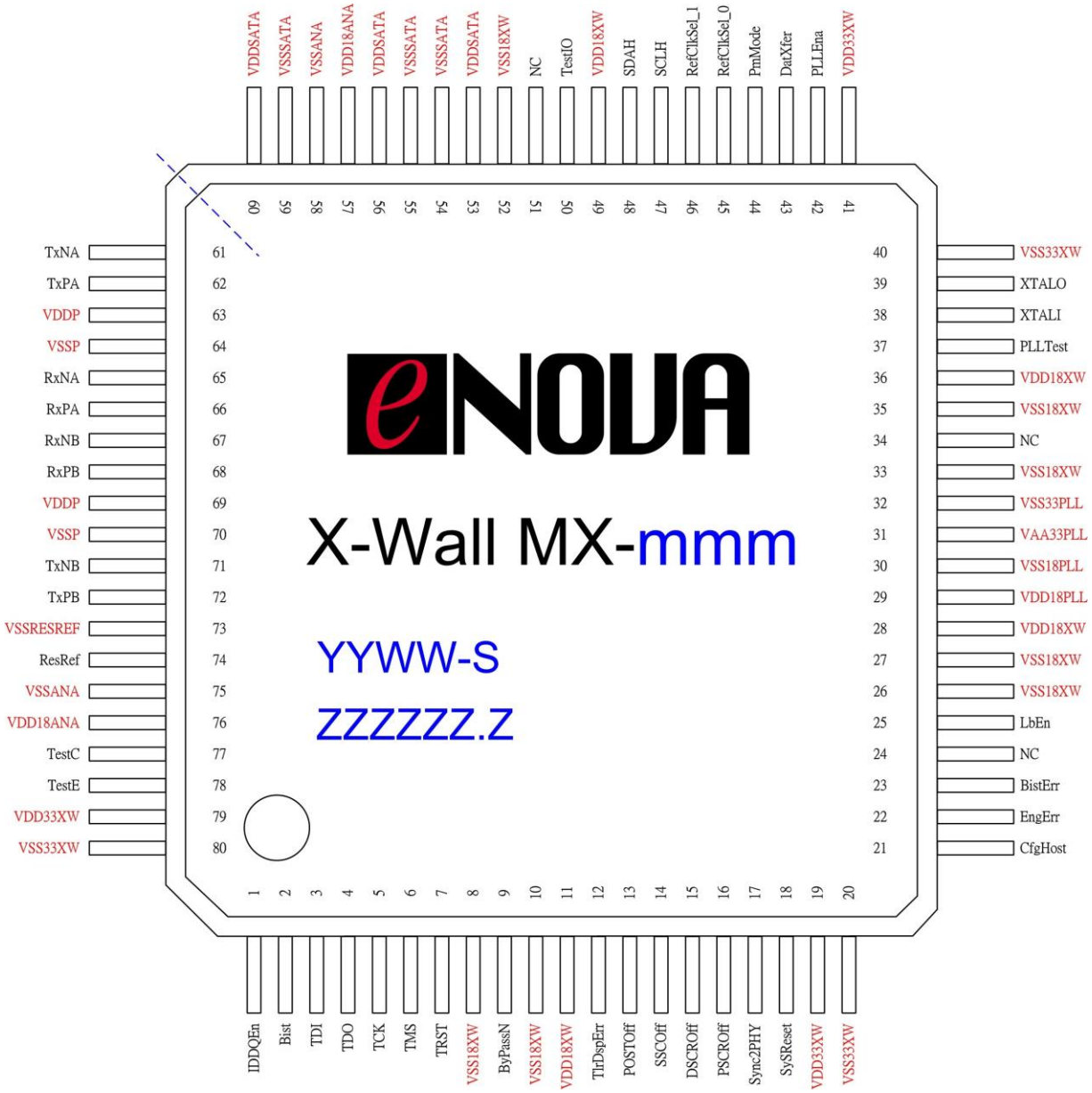
## Ordering Codes

| Stock Keeping Unit | Description | Mode of Operation (crypto mode) |
|---|---|---|
| X-Wall MX-256 | SATA real-time full disk encryption processor with AES 256-bit strength | ECB |
| X-Wall MX-128C | SATA real-time full disk encryption processor with AES 128-bit strength | CBC |
| X-Wall MX-256C | SATA real-time full disk encryption processor with AES 256-bit strength | CBC |

# *X-Wall MX* Pin Definitions

## *Pin Assignment*

All *X-Wall MX* family ASIC shares the same pin assignment and pin definition. Upgrading from a lower strength, for instance, from an *X-Wall MX-128* to an *X-Wall MX-256* is as easy as replacing the chip and the actual Secret Key.



## *Pin Definition and Description*

| PHY INTERFACE | | | | |
|---|---|---|---|---|
| **NAME** | **PIN*** | **DIR** | **TYPE** | **DESCRIPTION** |
| | | I | | Received differential input signals for channel A (channel #0). |
| | | O | | Differential serial output transmitted signals for channel A (channel #0). |
| | | I | | Received differential input signals for channel B (channel #1). |
| | | O | | Differential serial output transmitted signals for channel B (channel #1). |
| | | I/O | | Reference register, terminated to pin VSSREFREF through 2.7K±1% ohms. |
| Total | 9 | | | |
| | | | | |
| **CLOCK AND PLL CONTROL PINS** | | | | |
| **NAME** | **PIN** | **DIR** | **TYPE** | **DESCRIPTION** |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| Total | 6 | | | |
| | | | | |
| **FEATURE SETTING PINS** | | | | |
| | | | | |
| | | | | |
| | | I | | Built-in API command through Port Multiplier (PM) mode selection. When it is asserted HIGH, *X-Wall MX* will use control port address 0F (15) for API. If it is asserted LOW, then the *X-Wall MX* will use port address 0E (14) for API. Normally, disk access occupies port 0E (14). Don't care if command x58 and x59 are being utilized.<br><br>In the case that **Silicon Image SPIF215A USB/SATA bridge** controller is utilized to connect with the *X-Wall MX* chip, the SPIF215A will issue Software Reset command to port 0F (15) at initialization. As *X-Wall MX* by default, utilizes port 0F (15) for built-in API command thus it does not respond to the Software Reset command, causing a freezing condition. To solve that, grounding Pin#44 'pmMode' to select port 0E (14) for API command so that port 0F (15) of *X-Wall MX* will respond to SPIF215A 's Software Reset command and will be ready for regular disk access. Also reference to the ***X-Wall MX* reference schematics** for proper implementation.<br><br>In the case that **Oxford OXUF934DSB bridge controller** is utilized to operate with the *X-Wall MX*, the OXUF934DSB issues Software Reset command to port x0F (15) at initialization. As *X-Wall MX* by default, utilizes port x0F for built-in API command thus it does not respond to the Software Reset command, causing a freezing condition. To solve that, grounds Pin#44 'pmMode' (the default is NC) to select port x0E (14) for MX API command. More, set CfgHost of *X-Wall MX* to 0 (the default is 1). In addition, if the CfgHost=0, the OXUF934DSB disables HDD_POWER by GPIO#8. The firmware modification of the controller is required for proper power management, or manage power through different source. |
| Total | 2 | | | |
| | | | | |

**CONTROL AND INDICATOR SIGNALS**

| NAME | PIN | DIR | TYPE | DESCRIPTION |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| Total | 14 | | | |

**TWO-WIRE INTERFACE**

| NAME | PIN | DIR | TYPE | DESCRIPTION |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |

**JTAG TEST PINS**

| NAME | PIN | DIR | TYPE | DESCRIPTION |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | . |
| | | | | |
| | | | | |
| Total | 5 | | | |

**DEBUG INTERFACE**

| NAME | PIN | DIR | TYPE | DESCRIPTION |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| Total | 6 | | | |

**POWER GROUND**

| NAME | PIN | DIR | TYPE | DESCRIPTION |
|---|---|---|---|---|
| VDD18ANA | 57<br>76 | | | Analog 1.8V power supply for all PLLs of Serial ATA PHY. |
| VSSANA | 58<br>75 | | | Analog ground of VDD18ANA. |

| | | | | |
|---|---|---|---|---|
| VDDSATA | 60<br>56<br>53 | | | Digital 1.8V supply for Serial ATA PHY core. |
| VSSSATA | 59<br>55<br>54 | | | Digital ground of VDDSATA. |
| VSSRESREF | 73 | | | Analog ground returned for the external resistor reference. Connect a reference resistor between this pin and pin ResRef. |
| VDDP | 63<br>69 | | | 1.8V analog power supply for Serial ATA PHY high speed I/O |
| VSSP | 64<br>70 | | | Analog ground for Serial ATA PHY high speed I/O. |
| VDD33XW | 19<br>41<br>79 | | | Digital 3.3V supply for chip I/O. |
| VDD18XW | 11<br>28<br>36<br>49 | | | Digital 1.8V supply for chip core. |
| VSS33XW | 20<br>40<br>80 | | | Digital ground for chip I/O. |
| VSS18XW | 8<br>10<br>26<br>27<br>33<br>35<br>52 | | | Digital ground for chip core. |
| VDD18PLL | 29 | | | 1.8V digital power supply for PLL core. |
| VSS33PLL | 32 | | | Analog ground for PLL. |
| VAA33PLL | 31 | | | Analog 3.3V supply for PLL. |
| VSS18PLL | 30 | | | Digital ground for PLL core. |
| Total | 36 | | | |

# Electrical Characteristics

This section contains electrical specifications for the *X-Wall MX*. Please note, however, stressing conditions beyond the “Absolute Maximum Ratings” may cause permanent damage to the *X-Wall MX* device. Operating beyond the “operating conditions” is not recommended and extended exposure beyond “operating conditions” may adversely affect life and reliability of the *X-Wall MX* device.

## *Absolute Maximum Ratings*

| Symbol | Parameter | Value | | Unit |
|---|---|---|---|---|
| | | Min | Max | |
| **Ts** | **Storage Temperature** | **-55** | **+125** | **ºC** |
| **Ta** | **Operating Temperature** | **-45** | **+90** | **ºC** |

| | | | | |
|---|---|---|---|---|
| **VDD33** | **3.3V Digital Supply Voltage** | **-0.5** | **3.6** | **V** |
| **AVDD33** | **3.3V Analog Supply Voltage** | **-0.5** | **3.6** | **V** |
| **VDD18** | **1.8V Digital Supply Voltage** | **-0.5** | **1.93** | **V** |
| **AVDD18** | **1.8V Analog Supply Voltage** | **-0.5** | **1.93** | **V** |
| **VIN_IO33** | **Input Signal Voltage (Apply to 3.3V I/O pins)** | **-0.5** | **5** | **V** |
| **VO_IO33** | **Output Signal Voltage (Apply to 3.3V I/O pins)** | **-0.5** | **VDD33** | **V** |

### *DC Characteristics*

**Operating Conditions: VDD33=AVDD33=3.3V (±9.09%), VDD18=AVDD18=1.8V (±7.22%), GND=0V**

| Symbol | Parameter | Value | | Unit |
|---|---|---|---|---|
| | | Min | Max | |
| **VDD33** | **3.3V Digital Supply Voltage** | **3.0** | **3.6** | **V** |
| **AVDD33** | **3.3V Analog Supply Voltage** | **3.0** | **3.6** | **V** |
| **VDD18** | **1.8V Digital Supply Voltage** | **1.67** | **1.93** | **V** |
| **AVDD18** | **1.8V Analog Supply Voltage** | **1.67** | **1.93** | **V** |
| **IVDD33** | **3.3V Supply current (IVDD33 + IAVDD33)** | **12** | **12** | **mA** |
| **IVDD18** | **1.8V Supply current (IVDD18 + IAVDD18)** | **205** | **219** | **mA** |

## PCB Layout Guidelines

### *Typical Application Schematics*

In a typical add-on card design, the *X-Wall MX* is connected to two SATA connectors, a Mini-USB key interface, and LED indicators. For the detailed circuit layout files and Bill of Materials, please visit our website to download the latest revision. For special feature implementation such as built-in API source codes and related features, make a request with Enova Engineering at **info@enovatech.com**.

### *Typical Bill of Materials (BOM)*

| Item | Quantity |
|---|---|
| 1 | 1 |
| 2 | 26 |
| 3 | 9 |
| 4 | 11 |
| 5 | 2 |
| 6 | 8 |
| 7 | 2 |
| 8 | 2 |
| 9 | 1 |
| 10 | 1 |

| | |
|---|---|
| 11 | 1 |
| 12 | 6 |
| 13 | 2 |
| 14 | 1 |
| 15 | 1 |
| 16 | 1 |
| 17 | 1 |
| 18 | 1 |
| 19 | 1 |
| 20 | 1 |
| 21 | 1 |
| 22 | 2 |
| 23 | 2 |
| 24 | 1 |
| 25 | 1 |
| 26 | 1 |
| 27 | 2 |

## *Component Placement*

- For each power pin, add one bypass capacitor, which should be placed closely to the power pin.
- The DC blocking capacitors on SATA signal traces (C14, C15, C16, C17, C18, C19, C20 and C21) should be placed closely to *X-Wall MX*.
- R3 should be placed closely to *X-Wall MX*.
- The Crystal circuits should be placed closely to *X-Wall MX*.

## *PCB Trace Routing*

The *X-Wall MX* SATA signals routing can become really tricky thus deserves careful attention. The following bullets serve as a general guideline when the signal routing is attempted. Noted, however, this guideline does not cover the entire horizon of a complete design other than dealing with *X-Wall MX* specifically.

**SATA Signal Layout**

- To route the ResRef and VSSRESREF signals, **DO NOT** connect these signals to the ground and no other signals traces are routed near by these traces. Use 20mil trace width to route these traces and keep them as short as possible.
- The SATA TX signal pairs and SATA RX signal pairs **MUST HAVE** matching trace length. The difference of two line traces in either TX signal pairs or RX signal pairs should be restricted to below 150 mils.
- No other signals should be routed near by these SATA traces on **ANY** layers. There should be no more than one via on these traces; and there should be no stub on these traces.

Recommend to use DIP type SATA connector to avoid using more than one via. Keep these traces as short as possible. A solid ground plane should be placed directly underneath these traces to have better signal quality.

- The SATA TX signal pairs and SATA RX signal pairs **MUST HAVE** 100Ω differential impedance. To achieve aforementioned impedance value, Please refer to the paragraph of "PCB Parameters of Differential Signals."
- Do not route SATA traces underneath the crystal circuits or any other chips that employ high clocking.

**PCB Parameters of Differential Signals**

(Assume 1oz cooper density)

| Type | Material (dielectric Constant) | PCB thickness | Dielectric thickness | Trace width | Trace spacing |
|---|---|---|---|---|---|
| **2-layer**[5] | **FR4 (4.2)** | **1.6 mm** | **57 mil** | **SATA : 7mil<br>USB : 12mil** | **SATA : 5mil<br>USB : 5mil** |
| **4-layer** | **FR4 (4.2)** | **1.6 mm** | **4.3 mil** | **SATA : 5mil<br>USB : 6mil** | **SATA :10mil<br>USB : 8mil** |

[5] The layout engineer MUST follow this note precisely for a 2-layer PCB architecture is not a standard micro strip transmission line structure. There is a definite requirement to the spacing between the differential trace and the nearby cooper plane of the same layer. For PCB parameters specified above, this defined spacing is 8mil for SATA and is 9mil for USB.

# *X-Wall MX* Interface for Key Loading

There are two methods to deliver Secret Key to *X-Wall MX*. The first method is to deliver Secret Key through a built-in 2-wire serial interface. Both master and slave modes are supported. The second method is to deliver Secret Key via built-in API through SATA interface. The API protocol is further explained later in this design guide. Selection of various Key Loading methods is automatic, and is controlled by an embedded micro-controller.

If an external serial EEPROM device at bus address 1010000b was not found, *X-Wall MX* automatically switches its 2-wire serial interface to slave mode, awaiting commands and data from an external bus master. The designers can then use an external bus master to configure *X-Wall MX* and to deliver the *Secret Key*. Or designers can configure *X-Wall MX* and deliver *Secret Key* through SATA interface using API protocol. The following protocols reflect a correct key reading process while the *X-Wall MX* is at Master mode utilizing an SEEPROM (24LC02B) produced by either Atmel or Microchip at power on reset.

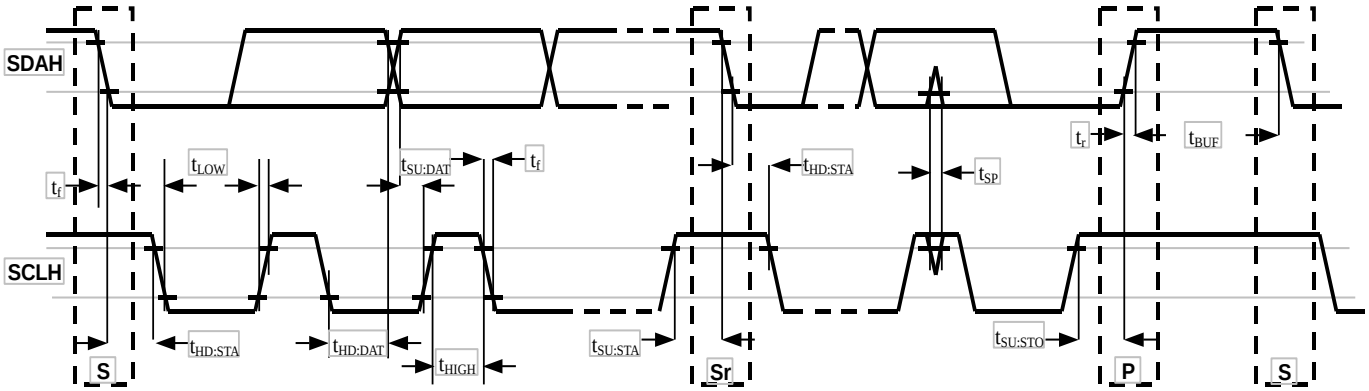
## *AES Key Ordering Convention*

Through out this document, the AES Key ordering will follow the convention stated in *FIPS-197*. That is, the least significant bit of a key sequence is the first input bit; the least significant byte is the first input byte, and so on. When denoting the key in symbols, the least significant bit is put to the left of the sequence. Therefore a 128-bit key sequence can be denoted as

$Key_{128} = \{ b_0, b_1, b_3, \ldots, b_{127} \}$,

## *X-Wall MX 2-wire Serial Interface Basic*

The bus interface has two bus wires. The first one, namely SDAH, is used for transmitting and receiving serial bit data. The second one, namely SCLH, is used for transmitting (master mode) and receiving (slave mode) clock pulses. By combining those two signals the START, repeated START, and STOP conditions are created, which are then used for constructing entire bus protocol. Listed below are signal-timing specification of SDAH and SCLH.




| PARAMETER | SYMBOL | MIN. | MAX. | UNIT |
|---|---|---|---|---|
| SCL clock frequency | $f_{SCL}$ | 0 | 400 | kHz |
| Hold time (repeated) START condition (**S**). After this period the first clock pulse is generated. | $t_{HD:STA}$ | 0.6 | - | µs |
| LOW period of the SCL clock | $t_{LOW}$ | 1.3 | - | µs |
| HIGH period of the SCL clock | $t_{HIGH}$ | 0.6 | - | µs |
| Set-up time for a repeated START condition (**Sr**) | $t_{SU:STA}$ | 0.6 | - | µs |
| Data hold time | $t_{HD:DAT}$ | 0 | 0.9 | µs |
| Data set-up time | $t_{SU:DAT}$ | 100 | - | ns |
| Rise time for both SDA and SCL signals | $t_r$ | $20+0.1C_b$ | 300 | ns |
| Fall time for both SDA and SCL signals | $t_f$ | $20+0.1C_b$ | 300 | ns |
| Setup time for STOP condition (**P**). | $t_{SU:STO}$ | 0.6 | - | µs |
| Bus free time between a STOP and a START condition. | $t_{BUF}$ | 1.3 | - | µs |
| Pulse width of spikes, which must be suppressed by the input filter. | $t_{SP}$ | 0 | 50 | ns |
| $C_b$: total capacitance of one bus line if pf. | | | | |

## X-Wall MX Configuration Protocol through 2-wire Serial Interface

Only slave mode protocol (that is, *X-Wall MX* 2-wire interface is configured as bus slave) will be described here since there is nothing designers need to do in master mode. First of all, we need to learn some of *X-Wall MX* internal register sets especially the following ones. The hexadecimal numbers preceding register names are addresses to the external 2-wire serial interface bus master.

### 0x80 Chip Command Register (CCR)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | |
| Type | | | | | | | | |
| Reset | | | | | | | | |

| Name | Code | Description |
|---|---|---|
| nop | 0000_0000 | |
| softRst | 0000_0001 | |
| cfgXWall | 0000_0100 | |
| selKeys | 0000_1ned | |

### 0x81 Command Parameter-1

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | para7 | para6 | para5 | para4 | para3 | para2 | para1 | Para0 |
| Type | WO | WO | WO | WO | WO | WO | WO | WO |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

### 0x82 Command Parameter-2

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | para7 | para6 | para5 | para4 | para3 | para2 | para1 | para0 |
| Type | WO | WO | WO | WO | WO | WO | WO | WO |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

### 0x83 Command Parameter-3

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | para7 | para6 | para5 | para4 | para3 | para2 | para1 | para0 |
| Type | WO | WO | WO | WO | WO | WO | WO | WO |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

### 0x84 Chip Status Register (CSR)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | |
| Type | | | | | | | | |
| Reset | | | | | | | | |

### *0x85 Cryptographic Configuration Register (CCFR)*

| ***Bit*** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ***Name*** | | | | | | | | |
| ***Type*** | | | | | | | | |
| ***Reset*** | | | | | | | | |

| **AS_1** | **AS_0** | **KS_2** | **KS_1** | **KS_0** | **Algorithm & Strength** |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 0 | |
| 1 | 0 | 0 | 1 | 1 | |
| 1 | 0 | 1 | 0 | 0 | |
| 0 | 1 | 0 | 1 | 0 | |
| 0 | 1 | 0 | 1 | 1 | |
| 0 | 1 | 1 | 0 | 0 | |
| 0 | 0 | - | - | - | |

### *0x86 Key Entry Mode/Status Register*

| ***Bit*** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ***Name*** | | | | | | | | |
| ***Type*** | | | | | | | | |
| ***Reset*** | | | | | | | | |

### *0x87 Cryptographic Operation Register (COR)*

| ***Bit*** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ***Name*** | | | | | | | | |
| ***Type*** | | | | | | | | |
| ***Reset*** | | | | | | | | |

| **N_KEY** | **E_KEY** | |
|---|---|---|

| | | |
|---|---|---|
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |

| N_KEY | D_KEY | Decryption Key Selected |
|---|---|---|
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |

| STEP | INSTRUCTIONS | COMMENTS |
|---|---|---|
| ***1*** | | |
| ***2*** | | |
| ***3*** | | |
| | | |
| ***4*** | | |
| ***5*** | | |

**Page Write Protocol**

[6] *X-Wall MX* is capable of reading a 2nd or more *Secret Key* sets within the same power on cycle. After the new *Secret Key* set is enabled, the previous *Secret Key* set is discarded. This feature can be best utilized over the drive re-purposing stage as pulling the *Secret Key* renders the entire disk content illegible.

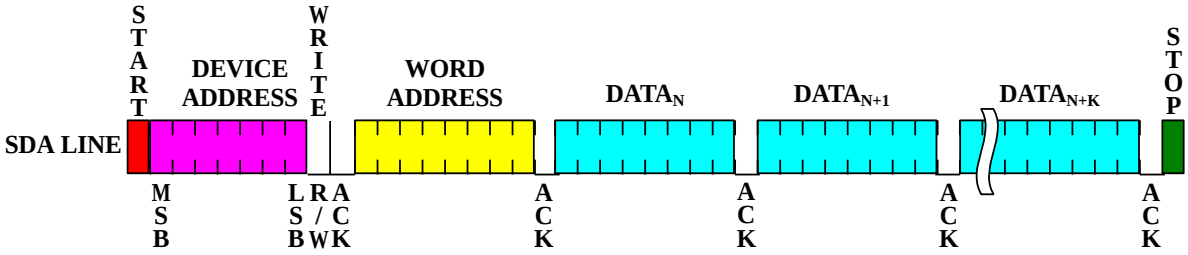



Figure 1. A page-write protocol: (page_write(device_address, start_address, data_byte#1, data_byte#2,…, data_byte#n);)

### Byte Write Protocol

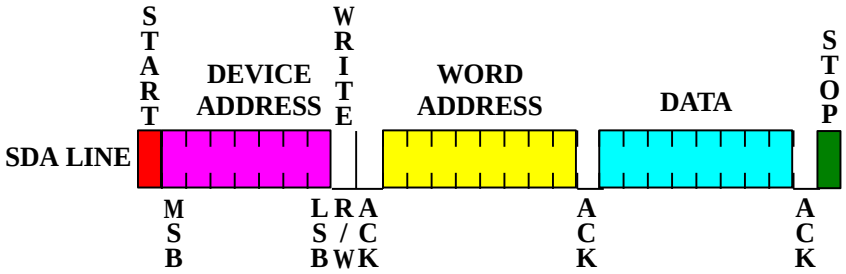



Figure 2. A byte-write protocol: (write_byte(device_address, word_address, data_byte);)

### Byte Read Protocol

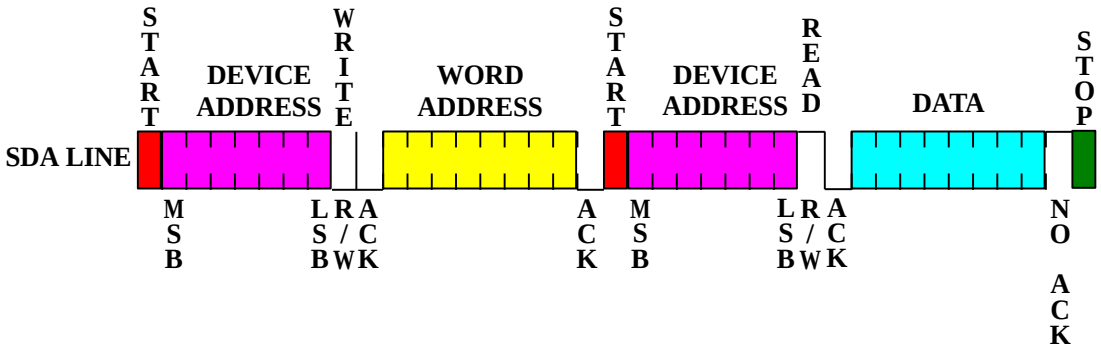



Figure 3. A byte-read protocol: (byte_read(device_address, word_address);)

## *X-Wall MX API Configuration Protocol through SATA interface*

*X-Wall MX* API mode is accomplished by two methods: 1.) act as a *Port Multiplier* with a single device port to allow registers access; and 2.) use two vendor specific commands.

Note that methods described above in #1 and #2 follow the same programming flow[7] but with different command opcodes. For better clarity in reading, 'pmMode' Read/Write commands (0xE4/0xE8) are specifically colored in **bold red** while 'Vendor Specific' Read/Write commands (0x58/0x59) are specifically colored in **bold blue**. Designers can choose to implement either 'pmMode' or 'Vendor Specific' commands but never mix them up.

### *X-Wall MX API Commands*

If a host adaptor wants to read register content from *X-Wall MX*, it should issue a '*Register – Host to Device* FIS' with corresponding fields having the following values:

| Register | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

where RegNum indicates the *double word* register address to be read, and PortNum has value 0xF. If the command is successfully received and executed, then *X-Wall MX* will issue a '*Register – Device to Host* FIS' with corresponding fields having the following values:

| Register | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | | | | | | | | |

[7] The programming guide is available upon specific request.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

Upon encountering an error, *X-Wall MX* shall send a '*Register – Device to Host* FIS' having the following values in corresponding fields:

| **Register** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

If a host adaptor wants to write to *X-Wall MX*'s internal registers, it must issue a '*Register – Host to Device* FIS' with corresponding fields having the following values:

| **Register** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

where RegNum indicates the double word register address to be written, PortNum has value 0xF, and "Value" carries bytes of the double word. If the command is successfully received and executed, then *X-Wall MX* will issue a '*Register – Device to Host* FIS' with corresponding fields having the following values:

| **Register** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

Upon encountering an error, *X-Wall MX* shall send a '*Register – Device to Host* FIS' having the following values in corresponding fields:

| **Register** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

### *X-Wall MX* API Configuration Protocol

*X-Wall MX* API uses the same register sets and storage buffer described in the 2-wire Serial Interface section. However, since both 'pmMode' and 'Vendor Specific' commands read or write four bytes (a double word) at a time, their addresses (RegNum) are different from those had been used in 2-wire Serial Interface Protocol. In general, the API register number can be derived from the two wire serial interface byte address using the equation below:

*X-Wall MX* supports multiple key loads which is also applicable with API programming.

| **STEP** | **INSTRUCTIONS** | **COMMENTS** |
|---|---|---|
| ***1*** | | |
| ***2*** | | |
| ***3*** | | |
| | | |
| | | |

| | | |
|---|---|---|
| ***4*** | | |
| ***Note*** | | |

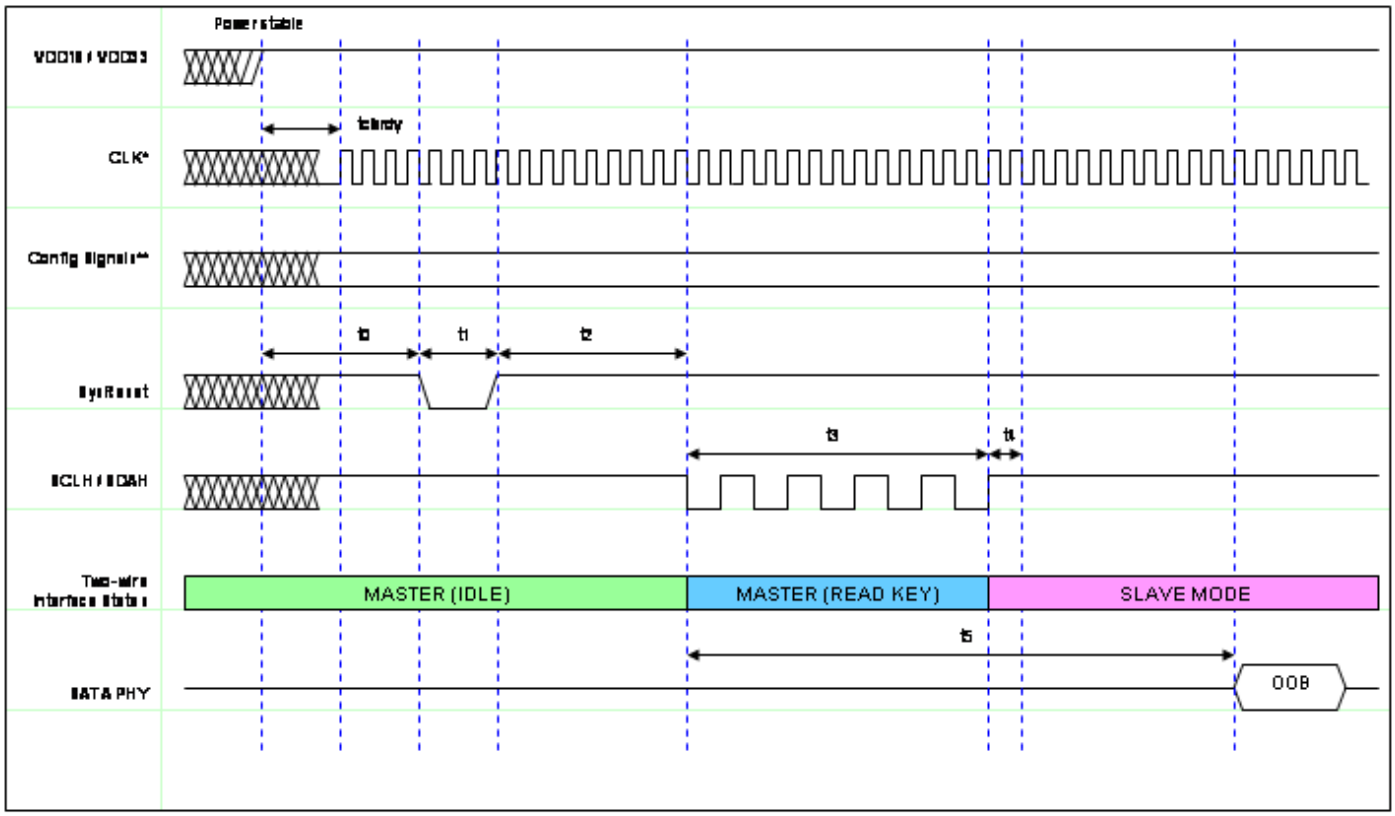
[8] Note that you **must** write this command to physically enable the *X-Wall MX* crypto mode operation. If the feature of repeated toggling between Crypto mode and Bypass mode is desirable, the switching back to Crypto mode from the previous Bypass mode will always require a WRITE command to the cfgXWall command to actually enable the Crypto function.

## Power-On Sequence

After power on (SysReset negated LOW and released HIGH), the Serial ATA PHY module requires about 6µs for internal PLLs to become fully functional. The PHY then will exchange OOBs on both channels. After the OOB sequence has completed with no error, the *X-Wall MX* is then ready for Serial ATA transactions.

After power on, *X-Wall MX* will perform Power-On-Self-Test and if an external *Secret Key* is available on the 2-wire Serial Interface, then *X-Wall MX* will load the key via the interface. After the *Secret Key* has been loaded and expanded, the *X-Wall MX* is ready for cryptographic operations. This sequence is completed far before the Serial ATA interface has finished exchanging OOBs. If the *Secret Key* were to be loaded via an external 2-wire Serial bus master or a Serial ATA host adaptor using built-in API, it is advised that the designers must make sure the key load sequence completes before any Serial ATA data transfer occurs for corrupted data might occur if SATA data transfer occurs before key load sequence is able to complete.

**X-Wall MX Power-On Sequence**




*CLK: the internal clock signal is generated by the built-in PLL of X-Wall MX.
**Config Signals must be ready before clock ready. See below signals.
(IDDQEn, BIST, ByPassN, TlrDspErr, POSTOff, DSCROff, PSCROff, Sync2PHY, LbEn, PLLTest, PLLEna, PmMode, RefClkSel0, RefClkSel1, TestIO, TestC, and TestE)

| Name | Description | Value | Comment |
|---|---|---|---|
| tclkrdy | Power stable to CLK(internal) ready | < 0.5 ms | The maximum time for PLL to output stable CLK |
| t0 | Power stable to SysReset valid (high) | > 1 ms | To ensure that SysReset will be valid only after internal CLK is ready |
| t1 | SysReset active (low) period | > 267 ns | |
| t2 | Power-On-Self-Test period | ~ 1.1 ms | |
| t3 | Read Key from external source | ~ 1.3 ms | For an AES 256-bits key value |
| t4 | *X-Wall MX* internal configuration time | ~ 30 us | |
| t5 | The First OOB | - | System dependent |

- The board design must meet t0 and t1 timing requirements.

# X-Wall MX Configuration Management

## Hardware Packaging

TQFP (Thin Quad Flat Package) provides low profile with 1.0mm body thickness, suitable for space concerned applications. Package size 10×10mm and lead-count 80 are offered for portable, lightweight and low profile applications. **All Enova *X-Wall MX* chips comply with RoHS and Lead-free specification.**

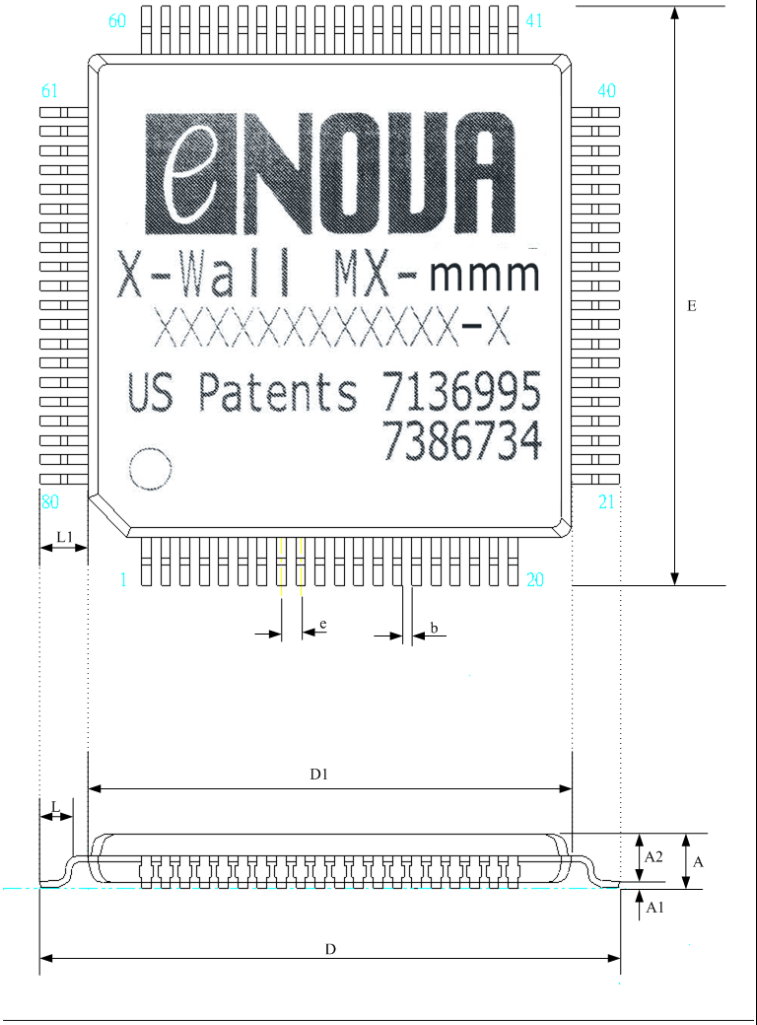
### Features

1. 10mm by 10mm body size with 80 lead-counts;
2. Copper lead-frame;
3. Low profile 1.0mm body thickness;
4. JEDEC MS-026/ACE standard outlines;

## Firmware Release

Hard coded version 1.1.0 released for ROM integration within a silicon.

## Hardware Version Control, Outline, and Dimension




| Symbol | Dimension [mm] | | |
|---|---|---|---|
| | min | NOR | MAX |
| e | 0.4 BSC | | |
| b | 0.13 | 0.18 | 0.23 |
| D1 | 10.00BSC | | |
| D , E | 12.00BSC | | |
| A | | | 1.20 |
| A1 | 0.05 | | 0.15 |
| A2 | 0.95 | 1.00 | 1.05 |
| L | 1.00(REF) | | |
| L1 | 0.45 | 0.60 | 0.75 |

**X-Wall MX** top marking:

**Enova** – Trademark

**X-Wall MX-mmm**, trademark and product SKU where mmm represents 3 to 4 digits as follows:

- 256, AES ECB 256-bit
- 256C, AES CBC 256-bit
- 192, AES ECB 192-bit
- 192C, AES CBC 192-bit
- 128, AES ECB 128-bit
- 128C, AES CBC 128-bit

**XXXXXXXXXXXX–S**

| 8 Lot No. | 4 date code | 2 version control|

8 digits for wafer lot number;

4 digits yyww (yy represents year and ww represents week) for manufacturing date code;

2 digits –S for version control where S represents serial mixed signal design;

**US Patent No.**: granted US patents listing.