Product Bulletin

January 2017 **PB1381**

MachXO2/MachXO3/LPTM21 WISHBONE Flash Corruption Avoidance

Lattice is issuing this Product Bulletin to inform its customers of certain circuit observations and to supply avoidance strategies.

Background: Lattice introduced the MachXO2 device family in 2010, followed by the MachXO3 device family in 2014. The products are very popular and among the most successful CPLDs in the product line. Over 100 million devices have shipped to date.

Observation: Lattice has recently discovered a corner use case in which, if a strict sequence of WISHBONE commands is issued, an unintended and disruptive corruption of the internal Flash memory can occur. While the probably of a typical user design encountering this issue is very low, the impact could be significant. Both MachXO2 and MachXO3 devices are effected, as well as the LPTM21 Platform Manager 2 product, which is based on the MachXO2 1200 device.

Lattice is aware of one customer-related instance of the issue described below. This instance was found during the customer's product development. Lattice is providing this document as a guide to determine exposure to this risk, and to recommend sure methods to guarantee avoidance of the issue.

Issue Summary:

When certain User Flash Memory or Configuration Memory Read commands issued over WISHBONE are set to read out 13 or more contiguous Flash pages, the user may be at risk of Flash corruption. The corruption may impact all flash sectors, including User, Configuration or Feature Row.

Figure 1: Risk examples:

Command	*Hex Sequence*
Read UFM, 13 pages	`0xCA 10 00 0E`
Read Flash, 197 pages	`0x73 10 00 C6`

A user is NOT AT RISK if:

1- The user does not use internal WISHBONE port, or
2- The user does not use internal WISHBONE port to access UFM or Configuration Flash, or
3- When accessing UFM or Configuration Flash, the user has designed a WISHBONE master circuit with deterministic behavior and always reads 12 flash pages or less with a single read command, or

4- The user has incorporated Lattice Reference design or Demonstration code, such as RD1126 “RAM-Type Interface for Embedded User Flash Memory Reference Design”, RD1129 “MachXO2 I2C Embedded Programming Access Firmware” or UG57 ” MachXO2 Programming Via WISHBONE Interface”. These designs utilize read commands with 1 page operands, or
5- For LPTM21 or MachXO2 + ASC designs: The user pattern is generated using Platform Designer tool, without imported HDL.

Figure 2: Safe examples:

Command	*Hex Sequence*
Read UFM, 1 page	0xCA 10 00 01
Read UFM, 1 page	0xCA 10 00 00
Read Flash, 4 pages	0x73 10 00 05

Detailed Issue Description:

Generically, the command processor receives command strings via a sysConfig port (JTAG, SSPI, I2C or WISHBONE) which are placed into a buffer. Command strings are composed of a sequence of Command bytes, operand bytes, and data bytes. Once a command operation is completed (e.g. finishes reading the specified last page of a flash read command) the next command execution is commenced using the port clock. The command interpreter ceases interpreting commands and is reset when the port receives a protocol STOP symbol (e.g. chip-select negation in SSPI, STOP in I2C, or ‘connection enable’ de-assertion in WISHBONE).

As opposed to the SSPI, I2C or JTAG ports, the WISHBONE interface port clock (wb_clk_i) is typically free-running. Several WISHBONE clock cycles typically occur prior to the de-assertion of the WISHBONE ‘connection enable’ bit. Additionally, even when the command buffer is ‘empty’, the last data or operand byte of the previous command persists in the buffer (this is typical in many FIFO buffer implementations). This combination of factors allows the command processor to begin acting upon the last resident byte in the command buffer upon completion of the previous command string. The command processor will ‘re-arm’ and begin processing the next command upon completion of the previous command.

When the last resident byte matches a valid command it can become an unexpected command. If the processor begins to execute the unexpected command, this can lead to placing the device into an invalid state. While executing the unexpected command, subsequent actual WISHBONE commands may be ignored, or serve to further corrupt the device. The actual magnitude of device disruption is therefore data, sequence and timing dependent. Most possible last resident byte values do not match valid commands, and not all valid commands are ‘potent’ commands, that is capable of disrupting or corrupting the device operation.

The analysis of the Lattice engineering team as determined that exposure to the issue is dependent upon the several necessary factors:

1) WISHBONE port use.

    a. A free-running clock must exist to allow the controller to process the last resident byte. For the wishbone port, the 'wb_clk_i' port is typically free-running. The SSPI, I2C and JTAG protocol sysConfig ports use non-free running (gapped) clocks, thus are not susceptible.

2) Specific sysConfig commands. Only read-type commands with auto-increment are susceptible. The following two documented commands are at risk

sysConfig Command	*Command (Hex)*	*Operation*
LSC_READ_INCR_NV;	*0x73*	*Read Configuration Flash*
LSC_READ_TAG;	*0xCA*	*Read UFM*

3) The specific sysConfig command's last operand byte must match a 'potent sysConfig command'. For the commands listed in 2), above, the last operand byte is the last byte of the 'PPPP' num_pages value. A 'potent sysConfig command' can perform a write or erase operation, change device state or alter a control registers. Therefore, setting the last byte of 'PPPP' num_pages to the same value as a 'potent sysConfig command' can lead to device disruption or corruption.

    A partial list of 'potent sysConfig commands' includes:

Potent sysConfig Command	*Command (Hex)*
ISC_ERASE	*0x0E*
LSC_PROG_CTRL0	*0x22*
ISC_DISABLE	*0x26*
LSC_INIT_ADDR_UFM	*0x47*
ISC_ENABLE_X	*0x74*
ISC_ENABLE	*0xC6*
LSC_PROG_TAG	*0xC9*
LSC_ERASE_TAG	*0xCB*

    A complete list of 'potent sysConfig commands' appears in the Appendix A.

4) The specific sysConfig command must complete its current operation to 're-arm' the command processor. This occurs, for example, when the last specified page is read from the UFM read operation.

Many user designs utilize a limited set of predictable and deterministic 'PPPP' num_pages counts when reading User or Configuration Flash. If the deterministic 'PPPP' num_pages values are exclusive of the list of 'Potent sysConfig commands' in factor 3) the issue is avoided and the probability of failure is zero.

Other user designs access a variable number of UFM pages. The probability of a practical design utilizing all necessary factors is low, but non-zero. Of these factors, factor 4) can be easily manipulated to guarantee avoidance of the issue and reduce the probability of failure to zero.

Issue Avoidance:

When the user WISHBONE Master circuit is used to read 13 or more pages of flash memory in a single command, Lattice recommends a straight-forward modification of the 'PPPP' operand which guarantees circumvention of the issue and removes any risk of inadvertent Flash corruption. The corrective action avoids the disruptive condition by altering the command operand to prevent the formal completion of the read command. This, in turn, prevents the command processor from 're-arming' and erroneously processing the buffer's last resident byte as a unexpected command. The corrective action (terminating a read command early) has no negative side effect.

> **Recommendation**: Modify the sysConfig read-type command 'PPPP' num_pages operand to be a constant 0x3FFF value.

Figure 5: Examples of recommended operand use:

Command	*Old Hex Sequence*	*New Hex Sequence*
Read UFM, 13 pages	`0xCA 10 00 0E`	`0xCA 10 3F FF`
Read Flash, 197 pages	`0x73 10 00 C6`	`0x73 10 3F FF`
Read UFM, any number of pages	`0xCA 10 xx xx`	`0xCA 10 3F FF`
Read Flash, any number of pages	`0x73 10 xx xx`	`0x73 10 3F FF`

Note that the lowest code 'Potent sysConfig Command' is 0x0E. Therefore, any customer design which is deterministic which guarantees UFM or Config accesses of 12 pages or less will not be exposed to a disrupted or corrupted device as a result of the issue described above.

Updated Documentation:

The following technical notes have been updated:

- TN1204 v3.8 Dec-2016 "MachXO2 Programming and Configuration Usage Guide": http://www.latticesemi.com/view_document?document_id=39085
- TN1246 v2.4 Nov-2016 "Using User Flash Memory and Hardened Control Functions in MachXO2 Devices Reference Guide": http://www.latticesemi.com/view_document?document_id=46300
- TN1279 v2.0 Dec-2016 "MachXO3 Programming and Configuration Usage Guide": http://www.latticesemi.com/view_document?document_id=50123
- TN1294 v1.7 Dec-2016 "Using Hardened Control Functions in MachXO3 Devices Reference Guide": http://www.latticesemi.com/view_document?document_id=50512

Customers are advised to utilize the above strategies to prevent possibility of field failures.

CONTACT

If you have any questions or require additional information, please contact techsupport@latticesemi.com

Sincerely,

Lattice Semiconductor MachXO2 Product Line Management

Revision History

Date	Version	Description of Revisions
12/20/2016	1.0	Initial Release
1/3/2017	1.1	Fixed Typo in footer