

**Edition 2016-02**

**Published by  
Infineon Technologies AG  
81726 Munich, Germany**

**© 2016 Infineon Technologies AG  
All Rights Reserved.**

#### **Legal Disclaimer**

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics. With respect to any examples or hints given herein, any typical values stated herein and/or any information regarding the application of the device, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation, warranties of non-infringement of intellectual property rights of any third party.

#### **Information**

For further information on technology, delivery terms and conditions and prices, please contact the nearest Infineon Technologies Office ([www.infineon.com](http://www.infineon.com)).

#### **Warnings**

Due to technical requirements, components may contain dangerous substances. For information on the types in question, please contact the nearest Infineon Technologies Office.

Infineon Technologies components may be used in life-support devices or systems only with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

# XMC1100 / XMC1200 / XMC1300

Fixed Flash Wait States

XMC1000 Family

ARM<sup>®</sup> Cortex<sup>®</sup>-M0  
32-bit processor core

Data Sheet Addendum

V1.0 2016-02

---

**XMC1100 / XMC1200 / XMC1300 Data Sheet Addendum**

**Revision History: V1.0 2016-02**

---

Previous Version: none

---

Page	Subjects
	Initial version.

---

**Trademarks**

C166™, TriCore™, XMC™ and DAVE™ are trademarks of Infineon Technologies AG.  
ARM®, ARM Powered®, Cortex®, Thumb® and AMBA® are registered trademarks of ARM, Limited.

CoreSight™, ETM™, Embedded Trace Macrocell™ and Embedded Trace Buffer™ are trademarks of ARM, Limited.

**We Listen to Your Comments**

Is there any information in this document that you feel is wrong, unclear or missing?  
Your feedback will help us to continuously improve the quality of this document.  
Please send your proposal (including a reference to this document) to:

[mcdocu.comments@infineon.com](mailto:mcdocu.comments@infineon.com)



## **Table of Contents**

<b>1</b>	<b>Fixed Flash Wait States</b> . . . . .	<b>6</b>
1.1	Flash read access with fixed wait states . . . . .	6
1.2	NVM Registers . . . . .	7
1.3	Electrical Parameters . . . . .	10
1.3.1	Flash Memory Parameters . . . . .	10

## 1 Fixed Flash Wait States

The parameter limits defined in this addendum extend the electrical parameters defined in the XMC1100 / XMC1200 / XMC1300 Data Sheet stated below.

- Data Sheet AA-Step, V1.4, 2014-05
- Data Sheet AB-Step, V1.6, 2015-04

### 1.1 Flash read access with fixed wait states

Per default the XMC1100 / XMC1200 / XMC1300 devices use a configuration with adaptive wait states for read accesses to the flash memory, dynamically adapting to the system frequency and flash access timing without user software interaction.

Alternatively, it is possible to configure the XMC1100 / XMC1200 / XMC1300 devices to apply fixed wait states to each flash read access, improving determinism of program execution from flash. The required number of wait states depends on the system frequency  $f_{MCLK}$ , as defined in the parameter  $N_{FWSFLASH}$ . The number of wait states can be configured with the bit `NVM_NVMMCONF.WS`, the selection of adaptive or fixed wait states is done with the bit `NVM_CONFIG1.WS`.

**Attention: Any write operation to the register `NVM_CONFIG1` to switch between adaptive and fixed wait states configuration must only modify the bit `NVM_CONFIG1.FIXWS`. Changing other bits in `NVM_CONFIG1` can lead to unpredictable results.**

**Attention: Before and after the fixed wait states configuration or the system frequency  $f_{MCLK}$  is changed, the number of selected wait states must always comply to the parameter  $N_{FWSFLASH}$ .**

Below is a code snippet defining the register addresses, configuring one wait state and then switching to operation with fixed wait states.

#### Example

```
// Headers and variables to fix number of wait states to "1"
#define ADDR1 0x40050008 //Address of NVM_NVMMCONF
uint32_t * NVM_NVMMCONF = (uint32_t *) ADDR1;
#define ADDR2 0x40050048 //Address of NVM_CONFIG1
uint32_t * NVM_CONFIG1 = (uint32_t *) ADDR2;

// init sequence to fix number of wait states to "1"
*NVM_NVMMCONF = *NVM_NVMMCONF | 0x1000; //Set .WS bit => 1WS
*NVM_CONFIG1 = *NVM_CONFIG1 | 0x0800; //Set .FIXWS bit => fixed
WS scheme
```

## 1.2 NVM Registers

### NVM Configuration Register

The definition of bit NVMCONF.12 changes to NVMCONF.WS.

#### NVM\_NVMCONF

**NVM Configuration Register (4005 0008<sub>H</sub>)** **Reset Value: 9000<sub>H</sub>**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>NVM_ON</b>	<b>INT_ON</b>	<b>0</b>	<b>WS</b>	<b>SECPROT</b>								<b>0</b>	<b>HRLEV</b>		<b>0</b>
rw	rw	rw	rw	rw								r	rw		rw

Field	Bits	Type	Description
<b>NVM_ON</b>	15	rw	<p><b>NVM On</b></p> <p>When cleared, no software code can be executed anymore from the NVM, until it is set again. I.e., already the software code that initiates the change in NVM_ON itself may not reside in the NVM, otherwise the software is stalled forever.</p> <p>0<sub>B</sub> <b>SLEEP</b>, NVM is switched to or stays in sleep mode.            1<sub>B</sub> <b>NORM</b>, NVM is switched to or stays in normal mode.</p>
<b>INT_ON</b>	14	rw	<p><b>Interrupt On</b></p> <p>When enabled the completion of a sequence started by setting NVMPROG.ACTION (write or erase sequence) will be indicated by NVM interrupt. The same is true for the wake-up sequence.</p> <p>0<sub>B</sub> <b>INTOFF</b>, No NVM ready interrupts are generated.            1<sub>B</sub> <b>INTON</b>, NVM ready interrupts are generated.</p>
<b>0</b>	13	rw	<p><b>Reserved for Future Use</b></p> <p>Must be written with 0 to allow correct operation.</p>
<b>WS</b>	12	rw	<p><b>Number of fixed Wait States</b></p> <p>Defines the number of fixed wait states when NVM_CONFIG1.FIXWS = 1<sub>B</sub>.</p> <p>0<sub>B</sub> 0 fixed wait states.            1<sub>B</sub> 1 fixed wait state.</p>
<b>SECPROT</b>	11:4	rw	<p><b>Sector Protection<sup>1)</sup></b></p> <p>This field defines the number of write, erase, verify protected sectors, starting with physical sector 0.</p>

**Fixed Flash Wait States**

Field	Bits	Type	Description
<b>0</b>	3	r	<b>Reserved</b> Read as 0; should be written with 0.
<b>HRLEV</b>	2:1	rw	<b>Hardread Level<sup>2)</sup></b> Defines single hardread level for verification with <b>NVMPROG.ACTION.VERIFY = 11<sub>B</sub></b> : 00 <sub>B</sub> <b>NR</b> , Normal read 01 <sub>B</sub> <b>HRW</b> , Hardread written 10 <sub>B</sub> <b>HRE</b> , Hardread erased 11 <sub>B</sub> <b>RFU</b> , Reserved for Future Use
<b>0</b>	0	rw	<b>Reserved for Future Use</b> Must be written with 0 to allow correct operation.

- 1) For SECPROT > 0, SECPROT defines the number of protected sectors. The sectors 0 to SECPROT-1 cannot be written, erased, or verified. All writes that target the protected sectors are accepted, but are internally ignored.
- 2) HRLEV defines the hardread level for a stand-alone verification sequence started with NVMPROG.ACTION.VERIFY = 11<sub>B</sub>. This hardread level is used until the end of the verification sequence. HRLEV may not be changed in between.

### Configuration 1 Register

The bit NVM\_CONFIG1.FIXWS allows to switch between adaptive and fixed wait state configuration.

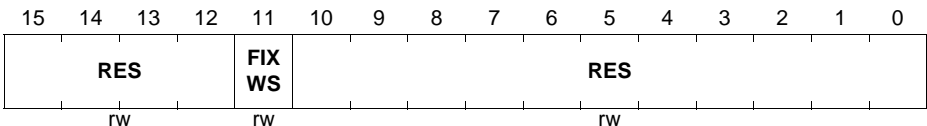
**Attention: Any write operation to the register *NVM\_CONFIG1* to switch between adaptive and fixed wait states configuration must only modify the bit *NVM\_CONFIG1.FIXWS*. Changing other bits in *NVM\_CONFIG1* can lead to unpredictable results.**

### NVM\_CONFIG1

Configuration 1 Register

(4005 0048<sub>H</sub>)

Reset Value: XXXX<sub>H</sub>



Field	Bits	Type	Description
<b>RES</b>	15:12	rw	<b>Reserved</b> Must not be changed when programming the NVM_CONFIG1 register.

**Fixed Flash Wait States**

<b>Field</b>	<b>Bits</b>	<b>Type</b>	<b>Description</b>
<b>FIXWS</b>	11	rw	<b>Wait States Scheme</b> Defines the scheme by which flash wait states are generated. With fixed wait states NVM_NVMCONF.WS defines the number of wait states. 0 <sub>B</sub> adaptive wait states. 1 <sub>B</sub> fixed wait states.
<b>RES</b>	10:0	rw	<b>Reserved</b> Must not be changed when programming the NVM_CONFIG1 register.



### 1.3 Electrical Parameters

#### 1.3.1 Flash Memory Parameters

This definition expands the flash wait states definition by parameters for the configuration with fixed wait states.

**Table 1 Flash Memory Parameters**

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Fixed Flash Wait States configured in bit NVM_NVMCONF.WS	$N_{FWSFLASH}$ SR	0	0	1		NVM_CONFIG1. FIXWS = 1 <sub>B</sub> , $f_{MCLK} \leq 16$ MHz
		1	1	1		NVM_CONFIG1. FIXWS = 1 <sub>B</sub> , $16 \text{ MHz} < f_{MCLK} \leq 32$ MHz

[www.infineon.com](http://www.infineon.com)

Published by Infineon Technologies AG

# Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

## Infineon:

[XMC1301Q040F0016ABXUMA1](#) [XMC1302Q040X0064ABXUMA1](#) [XMC1201T038F0200ABXUMA1](#)  
[XMC1301Q040F0032ABXUMA1](#) [XMC1201T038F0032ABXUMA1](#) [XMC1302Q040X0128ABXUMA1](#)  
[XMC1302T038X0016ABXUMA1](#) [XMC1302T016X0008ABXUMA1](#) [XMC1302Q024F0064ABXUMA1](#)  
[XMC1100Q040F0064ABXUMA1](#) [XMC1302Q024X0016ABXUMA1](#) [XMC1302Q024F0016ABXUMA1](#)  
[XMC1100Q040F0032ABXUMA1](#) [XMC1100Q040F0016ABXUMA1](#) [XMC1302T038X0064ABXUMA2](#)  
[XMC1301T016F0008ABXUMA1](#) [XMC1302Q040X0016ABXUMA1](#) [XMC1301T038F0032ABXUMA1](#)  
[XMC1202Q040X0016ABXUMA1](#) [XMC1201Q040F0064ABXUMA1](#) [XMC1100T016X0016ABXUMA1](#)  
[XMC1201Q040F0016ABXUMA1](#) [XMC1301T016F0016ABXUMA1](#) [XMC1301T016X0016ABXUMA1](#)  
[XMC1302T016X0032ABXUMA1](#) [XMC1100T038F0032ABXUMA1](#) [XMC1100T016X0032ABXUMA1](#)  
[XMC1100T016F0032ABXUMA1](#) [XMC1201T028F0016ABXUMA1](#) [XMC1302T038X0032ABXUMA1](#)  
[XMC1202T016X0064ABXUMA1](#) [XMC1302T038X0128ABXUMA1](#) [XMC1202Q024X0016ABXUMA1](#)  
[XMC1302T016X0016ABXUMA1](#) [XMC1100Q024F0064ABXUMA1](#) [XMC1202Q040X0032ABXUMA1](#)  
[XMC1301Q040F0008ABXUMA1](#) [XMC1202T016X0032ABXUMA1](#) [XMC1201T038F0128ABXUMA1](#)  
[XMC1301Q024F0016ABXUMA1](#) [XMC1202T028X0016ABXUMA1](#) [XMC1202T028X0032ABXUMA1](#)  
[XMC1301T038F0016ABXUMA1](#) [XMC1301Q024F0008ABXUMA1](#) [XMC1202T028X0064ABXUMA1](#)  
[XMC1201Q040F0200ABXUMA1](#) [XMC1202Q024X0032ABXUMA1](#) [XMC1100T038F0064ABXUMA1](#)  
[XMC1100Q024F0016ABXUMA1](#) [XMC1100T038X0064ABXUMA1](#) [XMC1302Q040X0200ABXUMA1](#)  
[XMC1201T038F0016ABXUMA1](#) [XMC1201Q040F0128ABXUMA1](#) [XMC1302Q040X0032ABXUMA1](#)  
[XMC1301T016F0032ABXUMA1](#) [XMC1301T038F0008ABXUMA1](#) [XMC1302T038X0064ABXUMA1](#)  
[XMC1201T038F0064ABXUMA1](#) [XMC1301T038X0064ABXUMA2](#) [XMC1100Q024X0016ABXUMA1](#)