The following document contains information on Cypress products. The document has the series name, product name, and ordering part numbering with the prefix “MB”. However, Cypress will offer these products to new and existing customers with the series name, product name, and ordering part number with the prefix “CY”.

**How to Check the Ordering Part Number**

1. Go to www.cypress.com/pcn.
2. Enter the keyword (for example, ordering part number) in the **SEARCH PCNS** field and click **Apply**.
3. Click the corresponding title from the search results.
4. Download the Affected Parts List file, which has details of all changes

**For More Information**

Please contact your local sales office for additional information about Cypress products and solutions.

**About Cypress**

Cypress is the leader in advanced embedded system solutions for the world's most innovative automotive, industrial, smart home appliances, consumer electronics and medical products. Cypress' microcontrollers, analog ICs, wireless and USB-based connectivity solutions and reliable, high-performance memories help engineers design differentiated products and get them to market first. Cypress is committed to providing customers with the best support and development resources on the planet enabling them to disrupt markets by creating new product categories in record time. To learn more, go to www.cypress.com.

AN205261

# FR, MB91460, Reset

This application note describes the reset flow on MB91460.Depending on reset cause (Power ON, external reset pin input, Watchdog reset or software reset) the MCUs of MB91460 Series run through two different reset states: INIT reset state (Settings Initialization Reset) and RST reset state (Operation Initialization Reset).

## Contents

1 Introduction....1
2 Reset Flow and Reset Causes....2
2.1 Reset Flow....2
3 INIT Reset State....3
3.1 How to enter INIT Reset State....3
3.2 Releasing the INIT Reset State request....3
3.3 Operations performed in INIT Reset State (INIT reset cancellation sequence)....3
4 RST Reset State....4
4.1 How to enter RST Reset State....4
4.2 Releasing the RST Reset State request....4
4.3 Operations performed in RST Reset State (RST reset cancellation sequence)....4
4.4 RST Reset Operation Modes....5
5 Evaluation of the Reset Cause....6
5.1 Reset Cause Register RSRR....6
5.2 Hardware Watchdog Control Register HWWD....7
6 Hardware Standby input HSTX....7
6.1 Influence of HSTX/INITX on Oscillation Stabilization Wait Time....7
7 Register initialisation of different Resets type....8
7.1 Register not initialised by Reset type....8
8 Additional Information....8
Document History....9

## 1 Introduction

This application note describes the reset flow on MB91460.

Depending on reset cause (Power ON, external reset pin input, Watchdog reset or software reset) the MCUs of MB91460 Series run through two different reset states: INIT reset state (Settings Initialization Reset) and RST reset state (Operation Initialization Reset).

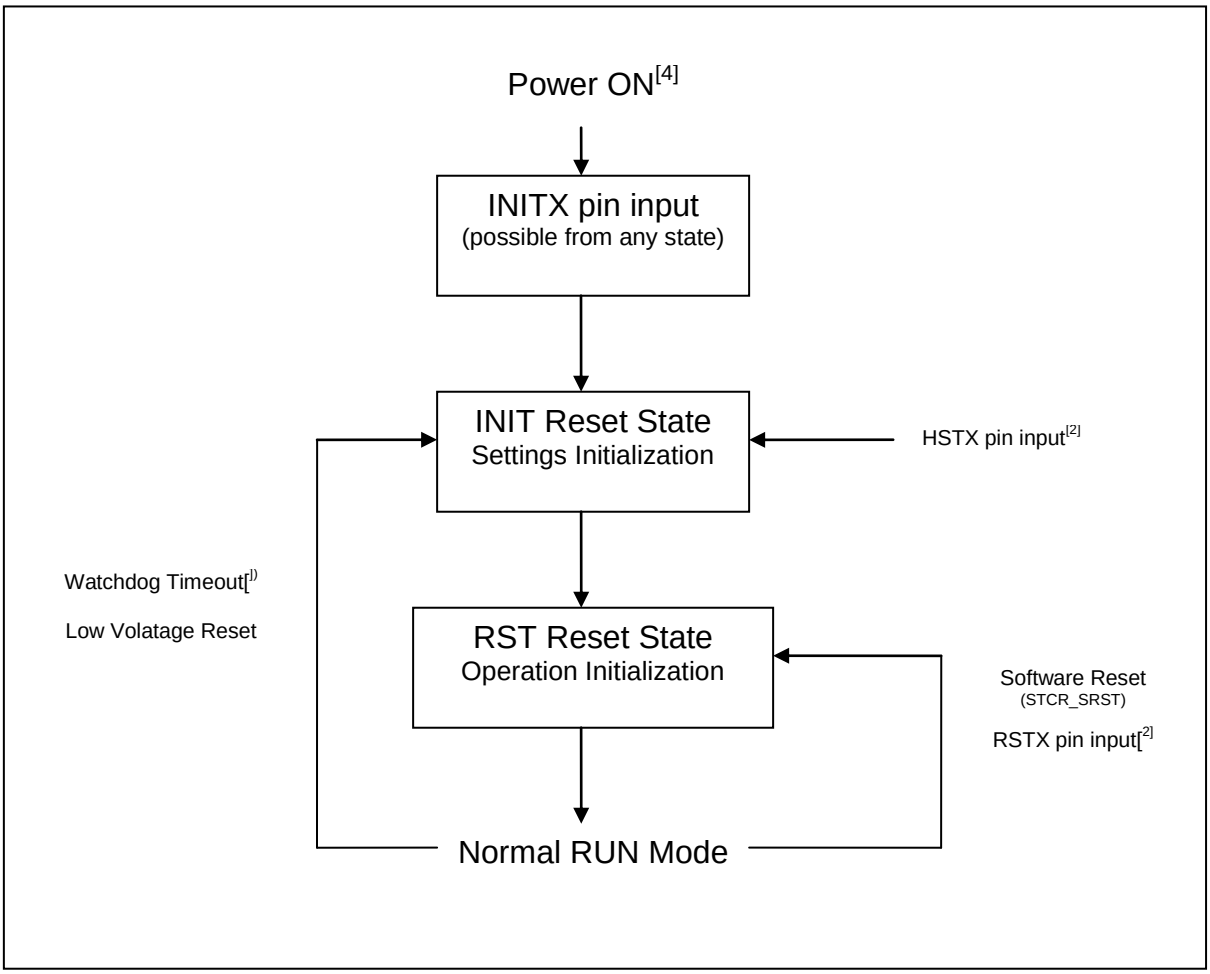
In the following the operations performed in the two above mentioned reset states are described and the flow through these states is explained. In addition information about how to determine the reset cause will be given.

# 2 Reset Flow and Reset Causes

This Chapter gives an overview about the flow through the different reset states and how these states can be invoked.

## 2.1 Reset Flow

Figure 1. Reset Flow




1. When INIT Reset State was entered due to Watchdog Timeout the setting of the oscillation stabilization time (STCR_OS[1:0]) are not changed
2. External RSTX pin is not supported by some of the MB91460 devices
3. External Hardware Standby input pin HSTX is not supported by some of the MB91460 devices. On devices supporting HSTX the setting of the oscillation stabilization wait time (STCR_OS[1:0]) depends on the order of releasing HSTX and INITX pin (for details please see of this document).
4. After power on the INITX input pin has to be kept low until the oscillation of the external oscillator has stabilized.

# 3 INIT Reset State

INIT Reset is used to initialize settings of the controller. In the following a description of the initialization steps is given.

## 3.1 How to enter INIT Reset State

The INIT Reset state is entered by:

- Pulling INTIX to low level a)
- Pulling HSTX to low level (when supported by the specific MB91460 Series MCU)
- Low Voltage Reset (for details about the setup of the Low Voltage Reset/Interrupt please refer to the Hardware Manual of MB91460 Series)
- Watchdog Reset generated by Hardware Watchdog (RC oscillator driven) or by Software Watchdog (Timebase Timer driven) (for details about the two Watchdog types please refer the Hardware Manual of MB91460 Series)

a) At Power on the INITX pin must be kept at low level until the external oscillator is started completely.

b) The oscillator start up time depends on the used oscillator, the circuit (e.g. the load capacities) and the PCB layout. The start up time has to be evaluated by the customer and the customer has to guarantee that the low level is supplied to the INITX input pin during this oscillator start up time.

## 3.2 Releasing the INIT Reset State request

The INIT Reset state request is released:

- Supply high level to INITX input pin c)
- Supply high level to HSTX input pin (when supported by the specific MB91460 Series MCU)

c) In case of Power On please keep the oscillator start up time in mind when releasing the INITX input pin to high level (please refer to the above remark added for the signal supplied to INITX while oscillator start up at Power On).

## 3.3 Operations performed in INIT Reset State (INIT reset cancellation sequence)

- Configuration of clock settings to default settings (clock source is Main Clock, PLL disabled)
- Oscillation Stabilization time is set to default value (STCR_OS[1:0] = b'00) d)
- Wait for oscillation stabilization wait time as specified by STCR_OS[1:0]
- Initialize Reset Cause Register RSRR e)
- Enter RST Reset State

d) Configuration of the oscillation stabilization wait time (STCR_OS[1:0]) to default settings is not done when INIT Reset State was entered due to Watchdog reset

e) A Watchdog Timeout will not initialize the Reset Cause Register.

# 4 RST Reset State

RST Reset is used to initialize the operation of the controller. In the following a description of the initialization steps is given.

## 4.1 How to enter RST Reset State

- Entered from previous INIT Reset State
- Apply low level to RSTX input pin
- Invoking a Software Reset by writing 0 to the SRST flag in the STCR register (for details about the Standby Control Register STCR please refer to the Hardware Manual of MB91460 Series)

## 4.2 Releasing the RST Reset State request

The RST Reset State request is released when the RST Reset state is entered.

## 4.3 Operations performed in RST Reset State (RST reset cancellation sequence)

- All MCU peripheral registers will be initialized to default values as described in the Hardware Manual of MB91460 Series. The clock setting registers will not be changed.
- Configuration of IO Ports to default values like all IO ports set to input, global port enable flag is cleared, … (for details refer to the Hardware Manual of MB91460)
- Evaluation of the setting of the Mode Pins MD[2:0] and setting up the device operating mode (bus mode and external bus width) correspondingly. Please find an overview of the device operation modes in the Hardware Manual of MB91460 Series.
- Set up of the CS0 Area to 0x00000000 – 0xFFFFFFFF (entire address range)
- Change to normal operation (RUN)
- Read Mode Vector from address 0x000FFFF8 (depending on the Mode Pin setting and the used MB91460 Series MCU the Mode Vector address might be in the internal ROM area or in the external ROM area)
- Write Mode Vector to Mode Register (MODR)
- Read Reset Vector from address 0x000FFFFC (depending on the Mode Pin setting and the used MB91460 Series MCU the Mode Vector address might be in the internal ROM area or in the external ROM area). The bus width which is used for the access to the Reset Vector is defined by the previously read Mode Vector.
- Write Reset Vector to PC (Program Counter) and start program execution

## 4.4 RST Reset Operation Modes

For the RST Reset operation two modes are possible: Normal (asynchronous) Reset Operation or Synchronous Reset Operation.

The RST Reset Operation mode is selected by the SYNCR-flag in the Timebase Timer Control Register TBCR.

Figure 2. Register TBCR

- **TBCR: Address 0482h (Access: Byte)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | bit |
|---|---|---|---|---|---|---|---|---|
| TBIF | TBIE | TBC2 | TBC1 | TBC0 | --- | SYNCR | SYNCS | |
| 0 | 0 | X | X | X | X | 0 | 0 | Initial value (INITX pin, watchdog) |
| 0 | 0 | X | X | X | X | X | X | Initial value (Software reset) |
| R(RM1),W | R/W | R/W | R1,W | R/W | RX/WX | RX/WX | R/W | Attribute |

For details on the Timebase Timer Control Register TBCR please refer to the Hardware manual of the MB91460 Series.

### 4.4.1 Normal (Asynchronous) RST Reset Operation

Normal reset operation refers to the mode when the device goes to the operation reset (RST) state immediately after an operation reset (RST) request occurs. For a normal reset, the device changes to the reset (RST) state immediately after a reset (RST) request is received regardless of the current state of internal bus access.

In normal reset mode, the result on any bus operation that is in progress at the time the device changes state is not guaranteed. However, acceptance of the operation reset (RST) request is guaranteed.

### 4.4.2 Synchronous RST Reset Operation

Synchronous RST reset operation refers to the mode when the device does not go to the operation reset (RST) state after an operation reset (RST) request until all bus access has halted.

In synchronous RST reset mode, the device does not go to the RST reset state when a RST reset request is received if internal bus access is still in progress. When such a RST reset request is received, a sleep request is issued to the internal bus. The device does not change to the operation reset RST state until all buses have shutdown operation and changed to SLEEP state.

In synchronous reset mode, the results of bus operations are guaranteed because the device does not change state until all bus access has halted. However, if bus access should not halt for some reason, no requests can be received while bus operation continues. In such case, the settings initialization reset (INIT) remains available at any time.

The following lists cases in which bus access may not stop:

- If bus wait is enabled due to continuous input of RDY (ready request) to the external expansion bus interface.

# 5 Evaluation of the Reset Cause

The Reset Cause Register can be used by an application to keep track of the cause of last reset.

## 5.1 Reset Cause Register RSRR

Figure 3. Register RSRR

- **RSRR: Address 0480h (Access: Byte, Half-word)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | bit |
|---|---|---|---|---|---|---|---|---|
| INIT | HSTB | WDOG | ERST | SRST | LINIT | WT1 | WT0 | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Initial value (INITX pin input) |
| - | - | - | X | X | - | 0 | 0 | Initial value (Watchdog reset) |
| X | X | X | - | - | X | 0 | 0 | Initial value (Software reset) |
| R/WX | R/WX | R/WX | R/WX | R/WX | R/WX | R/W | R/W | Attribute |

INIT: INIT Reset occurred due to INITX pin input or Hardware Watchdog Timeout[A)].

HSTB: INIT Reset occurred due to HSTX pin input.

WDOG: INIT Reset occurred due to Software Watchdog Timeout.

ERST: RST Reset occurred due to RSTX pin input.

SRST: RST Reset occurred due to Software Reset (STCR_SRST).

LINIT: INIT Reset occurred due to Low Voltage Reset.

A) When the INIT-flag in the RSRR register is set it is necessary to check the Hardware Watchdog control register HWWD also to distinguish between INITX pin input reset and Hardware Watchdog Timeout. For details on HWWD please refer to point 0 of this document and to the hardware manual of MB91460 Series.

The RSRR register is cleared after the first read. Since the RSRR is already evaluated by the BootROM when the MB91460 MCU is started in Single Chip Mode (pins MD[2:0]=b'000) the BootROM will provide the original content of RSRR in the CPU register R4 at application start and a copy of the original RSRR content can be found at a device specific RAM address. For the RAM address at which the RSRR content is stored please refer to the datasheet of the used MB91460 device.

For details on the Reset Cause Register RSRR please refer to the Hardware manual of the MB91460 Series

## 5.2 Hardware Watchdog Control Register HWWD

Figure 4. Register RSRR

- **HWWD: Address 04C7h (Access: Byte)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | bit |
|---|---|---|---|---|---|---|---|---|
| RESV0 | RESV0 | RESV0 | RESV1 | CL | RESV0 | RESV0 | CPUF | |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | Initial value (INITX pin input, watchdog reset) |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | X | Initial value (Software reset) |
| R/W0 | R/W0 | R/W0 | R/W1 | W | R/W0 | R/W0 | R/W | Attribute |

CPUF: When the last reset occurred due to a Hardware Watchdog Timeout the CPUF-flag in the Hardware Watchdog control register is set while in the RSRR register (please refer to point 5.1) only the INIT-flag is set. The CPUF-flag has to be cleared by the application.

For details on the Hardware Watchdog control register HWWD please refer to the Hardware manual of the MB91460 Series.

# 6 Hardware Standby input HSTX

This chapter describes the function of the Hardware Standby input pin HSTX.

Hardware Standby is intended to halt the complete MCU including also the external oscillators.

## 6.1 Influence of HSTX/INITX on Oscillation Stabilization Wait Time

Releasing the Hardware Standby Mode also the signal input to the oscillation stabilization wait time STCR_OS[1:0] is configured depending on the state of the INTX input pin:

- OS[1:0]=b'00 when INITX input pin is at low level while releasing HSTX from low to high
- OS[1:0]=b'11 when INITX input pin is at high level while releasing HSTX from low to high

Please see the below table for the resulting oscillation stabilization wait times depending on the setting of STCR_OS[1:0]

Table 1. STCR: OS bits

| OS[1:0] | The oscillation stabilization wait time after a reset (INIT) or on recovering from STOP state. | | |
|---|---|---|---|
| | Oscillation stabilization wait time | When using Main clock (For a 4.0MHz Main clock) | When using Sub clock (For a 32.768kHz Sub clock) |
| 00 | $\Phi \times 2^{1}$ | 1.00µs | 61µs |
| 01 | $\Phi \times 2^{11}$ | 1.0ms | 62.5ms |
| 10 | $\Phi \times 2^{16}$ | 32ms | 2.0s |
| 11 | $\Phi \times 2^{22}$ | 2s | 128s |

The Hardware Standby input pin HSTX is not supported by some of the MB91460 devices.

# 7 Register initialisation of different Resets type

This chapter describes the register initialisation of different Reset type.

Information about reset initialisation of the different types can be found also in MB91460 series Hardware Manual, Chapter 8 Device State Transition and Chapter 9 Reset.

## 7.1 Register not initialised by Reset type

Depending of Reset type occurrence some registers / bits are not initialised to default value.

The following table lists the register with are different initialled by reset types, refer to Hardware manual for more details.

Table 2. Register not initialised by different Reset types

| Reset type | Register not set to default value |
|---|---|
| INITX (pin) Reset | -- |
| Hardware Watchdog Reset (HWWD) | STCR.OS[1:0]<br>CSVCR, CSCFG<br>HWWDE, HWWD |
| Clock Supervisor Reset | CSVCR, CSCFG |
| Software Reset (RST) | CLKR<br>DIVR0/1<br>CSFG<br>OSCCR<br>STCR.OSCD[2:1], STCR.OS[1:0]<br>TBCR.SYNCR, .SYNCS<br>PLLDIVM<br>PLLDIVN<br>PLLDIVG<br>PLLMULG<br>PLLCTRL<br>CANPRE<br>RSRR<br>HWWDE, HWWD<br>OSCRH<br>WTCR.ST<br>LVDET<br>REGCTR<br>FSCR0/1<br>CSVCR, CSCFG |

# 8 Additional Information

Information about Cypress Microcontrollers can be found on the following Internet page:

http://www.cypress.com/cypress-microcontrollers

## Document History

Document Title: AN205261 - FR, MB91460, Reset

Document Number: 002-05261

| Revision | ECN | Orig. of Change | Submission Date | Description of Change |
|---|---|---|---|---|
| ** | - | NOFL | 11/02/2007 | V1.0 RSchum<br>First draft |
| | | | 06/24/2008 | V1.1 HPi, Corrected formatting |
| | | | 08/26/2010 | V1.2 MSt, Added chapter 7, Changed information to new Company name, exchanged Disclaimer |
| *A | 5084073 | NOFL | 01/16/2016 | Converted Spansion Application Note “MCU-AN-300052-E-V12” to Cypress format |
| *B | 5873197 | AESATMP9 | 09/05/2017 | Updated logo and copyright. |

## Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

## Products

| | |
|---|---|
| ARM® Cortex® Microcontrollers | cypress.com/arm |
| Automotive | cypress.com/automotive |
| Clocks & Buffers | cypress.com/clocks |
| Interface | cypress.com/interface |
| Internet of Things | cypress.com/iot |
| Memory | cypress.com/memory |
| Microcontrollers | cypress.com/mcu |
| PSoC | cypress.com/psoc |
| Power Management ICs | cypress.com/pmic |
| Touch Sensing | cypress.com/touch |
| USB Controllers | cypress.com/usb |
| Wireless Connectivity | cypress.com/wireless |

## PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

## Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

## Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2007-2017. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.